

INTRODUÇÃO

1. Num sistema computacional, como definiria com as suas próprias palavras o conceito de espaço de endereçamento?
São endereços (endereços) na memória, cada espaço tem a sua única identificação (etiqueta)
 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*
 4. Identifique e descreva pelas suas palavras qual o papel, na arquitetura de um sistema computacional:
 - a. do *Data Bus* → *Canal onde circula a data entre os componentes*
 - b. do *Address Bus* → *Transfere a informação da localização dos dados que o CPU quer ler ou escrever.*
 - c. do *Control Bus* → *Transfere os sinais de controlo, que têm como objectivo controlar as operações do sistema.*
 5. Na arquitetura de um sistema computacional, como designa o barramento que permite identificar o registo, na memória ou num periférico, do qual ou para o qual vai ocorrer uma transferência de informação. *Address Bus*
 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 (Control Bus)*

MICROCONTOLADORES (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? *Programa criado na plataforma A e executado na plataforma B*

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

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

 - quanto aos principais aspectos da sua arquitetura interna → + simples
 - quanto à sua frequência de trabalho → menor
 - quanto à disponibilização de periféricos → menor
 - quanto ao custo → + barato
 - quanto à energia consumida → - energia consumida
 - quanto aos seus campos de aplicação → campos + simples

distanciamento funções básicas de transição e execução de um programa

Microprocessada ≠ Microcontrolador

↑
+ complexo ↑
+ simples

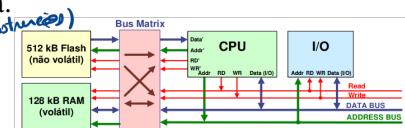
10. Como descreveria as principais características de um sistema embebido?

Realiza uma tarefa em específico (programável), baseado num microcontrolador.

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.

MÓDULOS DE I/O

Arquitetura Harvard → 2 estágios de endereçamento de memória



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

 - a. TRIS → configuração dos portos input & output
 - b. LAT → manipular o estado dos portos output
 - c. PORT → ler o estado dos portos input

- o de 4k bytes, na gama de endereços 0x0000 a 0x7FFF*

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

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

 - $LATB = (LATB \& 0xFF3F) | (1 \ll 7);$
 - $LATB = (LATB \& 0xFFFFB) | (0 \ll 6);$
 - $LATB = (LATB | 0x0080) \& (0 \ll 6);$
 - $LATB = (LATB \& 0x0000) | (1 \ll 7);$

$$\begin{array}{ccccccccc}
 x_{15} & x_{14} & x_{13} & x_{12} & x_{11} & x_{10} & x_9 & x_8 & 0 \\
 \underbrace{\quad\quad\quad}_{F} & \underbrace{\quad\quad\quad}_{F} & \quad & \quad & \quad & \quad & \quad & \quad & \quad
 \end{array}
 \quad
 \begin{array}{ccccccccc}
 0 & 0 & \alpha_5 & \alpha_4 & \alpha_3 & \alpha_2 & \alpha_1 & \alpha_0 \\
 \quad & \quad & \underbrace{\quad\quad\quad}_{B} & \quad & \quad & \quad & \quad & \quad
 \end{array}$$

$\alpha_1 \alpha_5 \alpha_4 \alpha_3 \alpha_2 \alpha_1 \alpha_0$

1 0 ✓

Shift 7 pages as section "1"

$$B - 1011 - 11$$

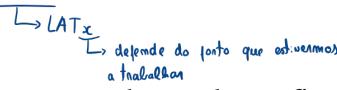
$\begin{matrix} 2 \\ 1 \end{matrix}$ $\begin{matrix} 2 \\ 3 \end{matrix}$ $\begin{matrix} 2 \\ 2 \end{matrix}$

Amd

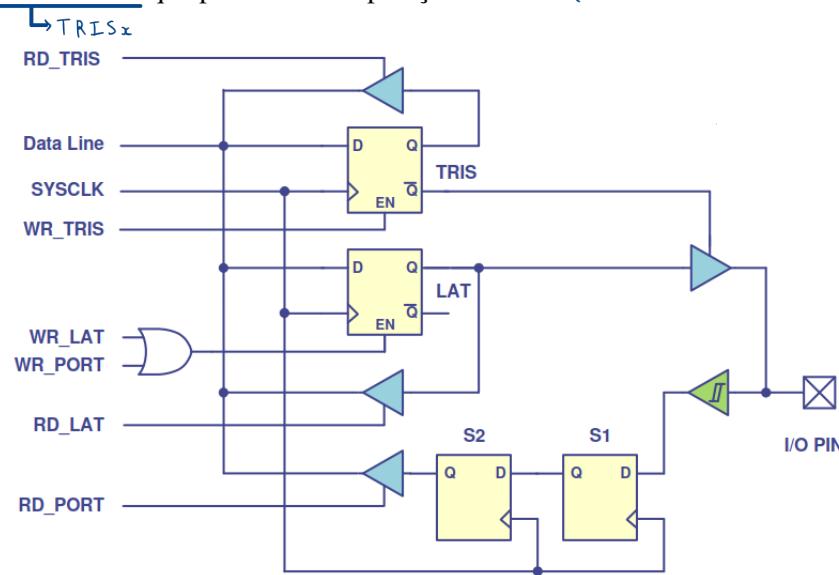
1 << 2

Dā shift \Rightarrow x as value logic!

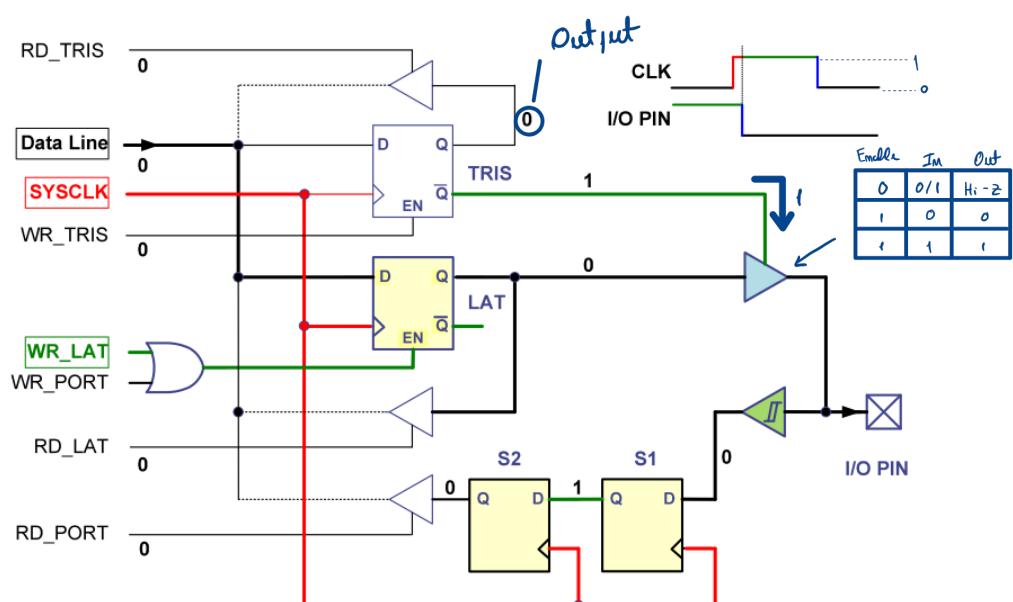
13. Num porto de I/O do PIC32 (esquema abaixo), quando o porto se encontra configurado como porto de entrada,
- continua a ser possível escrever no registo de saída?
 - continua a ser possível ler o valor que se encontra armazenado no registo LAT?
 - se sim, qual o sinal que permite realizar essa leitura?



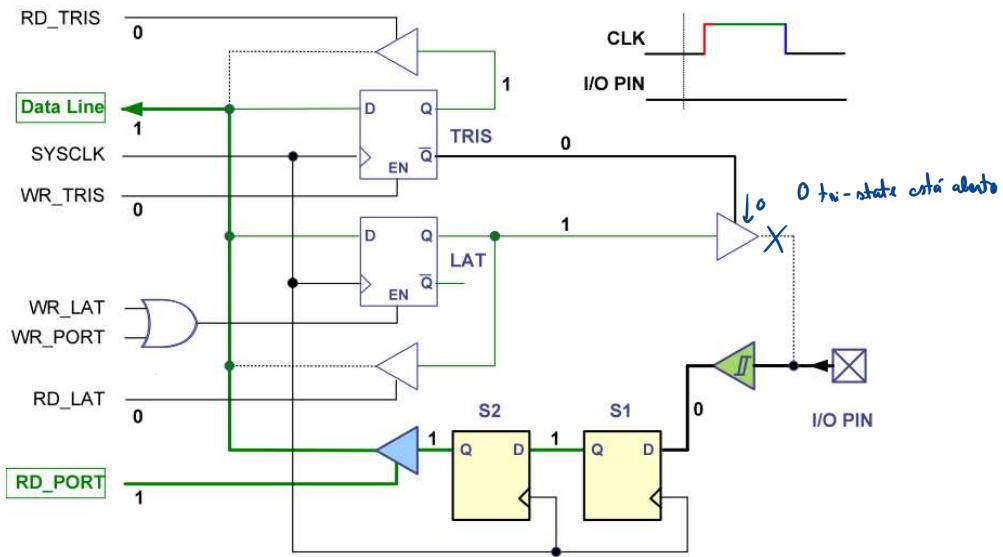
14. Num porto de I/O do PIC32 (esquema abaixo), se eu não souber qual a configuração do mesmo (saída ou entrada),
- será possível saber essa informação programaticamente?
 - se sim, qual o sinal de controlo que permite essa operação?



15. Considere que está a decorrer uma operação sobre um porto de I/O de um PIC32 (ver esquema abaixo – considere o instante assinalado no diagrama temporal).
- o porto encontra-se configurado como entrada ou saída? *Output*
 - identifique e descreva qual a operação que está em curso. *A escrever num porto de saída*



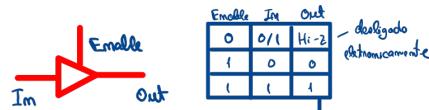
16. Repita o exercício anterior para o esquema apresentado abaixo



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

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? **Modelo de programação**
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? **Inicia - Le ou escreve - monitoriza**
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?
Interface de saída e entrada, que permite a comunicação entre o periférico e o CPU.



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

- que tipo de dispositivos lógicos devem ser usados na ligação ao barramento de dados?
- por que razão é fundamental usar esses dispositivos?

Bufffer Tri-State

INTERRUPÇÕES

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

- descreva sucintamente os dois modos de funcionamento do sistema de interrupções
- no caso do sistema de interrupções feita por hardware, como descreveria sucintamente o seu funcionamento
- 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

E/S Programada

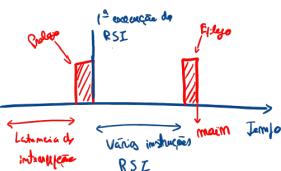
- Polling
- Verifica se não está ocupado
- Bloqueante

E/S Interrupção

- RSI
- O leitor é que executa quando está pronto
- Pode pegar outras tarefas

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

- "prólogo"? — Salvaguardar a informação antes da RSI
- "epílogo"? — Reposição da informação depois da RSI



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

São as negas que o CPU foge tarefas que não estão relacionadas com a transferência de informação

28. Considere um sistema baseado num CPU a funcionar a uma frequência de 10 MHz com uma taxa de execução de 5 MIPS (5×10^6 instruções por segundo, CPI = 2) que processa por interrupção eventos externos periódicos. Se o overhead total do atendimento 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.

29. Considere um sistema baseado num CPU a funcionar a uma frequência de 40 MHz com uma taxa de execução de 16 MIPS (16×10^6 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.

30. 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 (33.3×10^6 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.

31. Considere um sistema baseado num CPU a funcionar a uma frequência de 10 MHz com uma taxa de execução de 5 MIPS (5×10^6 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.

32. Considere agora um sistema baseado num CPU a funcionar a uma frequência de 80 MHz com uma taxa de execução de 40 MIPS (40×10^6 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.

33. Descreva sucintamente, para o caso de um sistema de interrupções vetorizadas com prioridade estabelecida por "daisy chain":

- como é estabelecida a prioridade de resposta a interrupções simultâneas
- como se designa o sinal por hardware que permite estabelecer e assegurar o funcionamento do sistema em "daisy chain"

1. Frequência do processador (Clock Frequency)

- Normalmente medido em MHz ou GHz.

ex: 10 MHz - realiza 10 milhões de operações de ciclos de clock por segundo

2. MIPS (Million Instructions Per Second)

$$\text{MIPS} = \frac{\text{Clock Frequency (MHz)}}{\text{CPI}}$$

\ Cycles Per Instruction - N° médio de ciclos de CLK para executar uma instrução

3. Overhead

- Refere-se ao custo adicional de tratar uma interrupção

4. Frequência Máxima das Instruções

- Número máximo de vezes por segundo que o sistema pode lidar com uma interrupção sem perder eventos

$$f_{\max} = \frac{f_{\text{clock}}}{\frac{\text{Ciclos Totais}}{\text{por interrupção}}} - \text{Overhead}$$

\ frequência do clock
Overhead + Ciclos da rotina de serviço

5. Número máximo de Instruções da Rotina de Serviço

$$\text{Ciclos disponiveis} = \frac{f_{\text{clock}}}{f_{\text{interrupção}}} - \text{Overhead}$$

$$\text{Instruções máximas} = \frac{\text{Ciclos disponiveis}}{\text{CPI}}$$

6. Mais fórmulas:

- Ciclos Totais Disponíveis

$$\text{Ciclos totais} = \frac{f_{\text{clock}}}{f_{\text{interrupção}}}$$

- Número Máximo de Instruções

$$\text{Instruções} = \frac{\text{Ciclos Disponíveis}}{\text{CPI}}$$

- Frequência Máxima de Interrupções

$$f_{\max} = \frac{f_{\text{clock}}}{\text{Overhead} + \text{Ciclos da Rotina}}$$

(28)

$$\text{Ciclos totais} = 20 + 40 = 60 \text{ ciclos}$$

$$f_{\max} = \frac{f_{\text{clock}}}{\text{C. Totais}} = \frac{10 \times 10^6}{60} = 166,66 \text{ Hz} \\ = 166,66 \text{ kHz}$$

$10 \text{ MHz} - 1000 \text{ Hz}$

34. Descreva, sucintamente, as fases temporais de atendimento a uma interrupção num sistema de interrupções vetorizadas.
35. 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.
36. Descreva, sucintamente, o funcionamento de um sistema de interrupções baseado em "identificação da fonte por software"

DMA

37. 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. Pedido (Bus Request) 3. Configuração 5. Fim da transferência
 2. Concessão (Bus Grant) 4. Transferência
38. Qual a operação que, tipicamente, um controlador de DMA executa quando conclui um processo de transferência de informação enquanto *bus master*. *DMA Interrupt, com o objetivo do DMAC avisar o CPU que já acabou a transferência.*
39. Descreva, sucintamente, qual a finalidade do sinal busgrant, num sistema que suporte transferência de dados por DMA, quem gera esse sinal e em que circunstâncias.
confirmação que o barramento está livre.
40. 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.
 1. Pedido do DMA 2. CPU concede breve controlo ao DMAC 3. DMAC faz a transferência mensárga 4. DMAC transfere e libera
volta ao inicio
41. Descreva, sucintamente, qual a diferença entre os modos de operação “bloco” e “burst” de um controlador de DMA.
O barramento é dividido entre o CPU e o DMAC
transfaz tudo de uma vez de forma contínua (não parando), dando assim
uma vez de forma contínua (não parando), dando assim
42. 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:
- controlador de 32 bits, frequência de funcionamento do DMAC de 500MHz, bloco de 512 words de 32 bits
 - controlador de 16 bits, frequência de funcionamento do DMAC de 1GHz, bloco de 512 words de 32 bits
 - controlador de 16 bits, frequência de funcionamento do DMAC de 1GHz, bloco de 512 words de 16 bits
 - controlador de 16 bits, frequência de funcionamento do DMAC de 500MHz, bloco de 2Kwords de 16 bits
43. Volte a resolver o problema anterior considerando agora que um *bus cycle* é realizado em 2 ciclos de relógio e para as seguintes condições:
- controlador de 32 bits, frequência de funcionamento do DMAC de 1GHz, bloco de 1K words de 32 bits
 - controlador de 16 bits, frequência de funcionamento do DMAC de 500MHz, bloco de 2K words de 32 bits
 - controlador de 16 bits, frequência de funcionamento do DMAC de 1GHz, bloco de 256 words de 32 bits
 - controlador de 16 bits, frequência de funcionamento do DMAC de 500MHz, bloco de 2Kwords de 16 bits
44. Resolva as duas primeiras alíneas do problema anterior considerando agora que o controlador é dedicado.
45. 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 mínimo necessário para efetuar a transferência de um bloco de dados para as seguintes condições:
- controlador de 32 bits, frequência de funcionamento do DMAC de 250 MHz, bloco de 512 words de 32 bits
 - controlador de 16 bits, frequência de funcionamento do DMAC de 1GHz, bloco de 512 words de 32 bits
 - controlador de 16 bits, frequência de funcionamento do DMAC de 500MHz, bloco de 2Kwords de 16 bits

42. 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:

 - controlador de 32 bits, frequência de funcionamento do DMAC de 500MHz, bloco de 512 words de 32 bits
 - controlador de 16 bits, frequência de funcionamento do DMAC de 1GHz, bloco de 512 words de 32 bits
 - controlador de 16 bits, frequência de funcionamento do DMAC de 1GHz, bloco de 512 words de 16 bits
 - controlador de 16 bits, frequência de funcionamento do DMAC de 500MHz, bloco de 2Kwords de 16 bits

$$\text{Fórmula: } T = \frac{\text{Nº de words} \times \text{Largura do word's (bits)}}{\text{Largura do barramento (bits)} \times \text{Frequência (Hz)}}$$

$$a) T = \frac{512 \cdot 32}{32 \cdot 500 \cdot 10^6} = 1024 \cdot 10^{-6} = 1,024 \mu\text{s}$$

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

- a. controlador de 32 bits, frequência de funcionamento do DMAC de 1GHz, bloco de 1K words de 32 bits
 - b. controlador de 16 bits, frequência de funcionamento do DMAC de 500MHz, bloco de 2K words de 32 bits
 - c. controlador de 16 bits, frequência de funcionamento do DMAC de 1GHz, bloco de 256 words de 32 bits
 - d. controlador de 16 bits, frequência de funcionamento do DMAC de 500MHz, bloco de 2Kwords de 16 bits



$$\text{Fórmula: } T = \frac{\underset{\text{Transferencia}}{N^o \text{ de words} \times \text{Largura do word's (bits)}}}{\underset{\text{Largura do barramento (b.ts) \times Freqüencia (Hz)}}{} \times \underset{\text{U^o de ciclos}}{}}$$

$$a) T = \frac{0,24 \cdot 32}{32 \cdot 1 \cdot 10^9} \times 2 = 2,048 \cdot 10^{-6} = 2,048 \mu s$$

46. 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 DMAC de 500 MHz, bloco de 512 words de 32 bits
 - b. controlador de 16 bits, frequência de funcionamento do DMAC de 1GHz, bloco de 4K words de 32 bits
 - c. controlador de 16 bits, frequência de funcionamento do DMAC de 1GHz, bloco de 512 words de 32 bits
 - d. controlador de 16 bits, frequência de funcionamento do DMAC de 500MHz, bloco de 1Kwords de 16 bits

$$\text{Fórmula : } N = \frac{\text{Número total de bits a transferir}}{\text{Largura do enunciado (bits)}} = \frac{N^o \text{ de words} \times \text{Largura do word (bits)}}{\text{Largura do enunciado (bits)}}$$

$$a) N = \frac{512 \cdot 32}{32} = \frac{16384}{32} = 512 \text{ Bus cycles}$$

47. Determine a taxa de transferência de pico (expressa em Bytes/s) de um DMAC 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 (equivalente a $1*T_{BC}$) para efetuar uma operação de leitura ou escrita por esse DMAC. O tempo mínimo entre operações elementares deverá ser de $1T_{BC}$

$$\text{Fórmulas: } T_{BC} = \frac{l}{g} \text{ giroscópio}$$

$$\text{Taxa transference Price} \downarrow \\ R_{\text{transf}} = \frac{1}{T_{\text{transf}}}$$

$$T_{transfere\acute{c}ia} = \underbrace{2T_{BC}}_{\text{Letura}} + \underbrace{1T_{BC}}_{\text{tempo de im\^tensivo entre as operac\~oes}} + \underbrace{2T_{BC}}_{\text{Escrita}}$$

$$\text{Taxa transferencia} = R_{\text{transf}} \times \text{Tamamlo de dados (bytes)}$$

$$\therefore 32 \text{ bits} - 4 \text{ bytes}$$
$$\therefore 8 \text{ bits} = 1 \text{ byte}$$

- ✓
46. 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:
- controlador de 32 bits, frequência de funcionamento do DMAC de 500 MHz, bloco de 512 words de 32 bits
 - controlador de 16 bits, frequência de funcionamento do DMAC de 1GHz, bloco de 4K words de 32 bits
 - controlador de 16 bits, frequência de funcionamento do DMAC de 1GHz, bloco de 512 words de 32 bits
 - controlador de 16 bits, frequência de funcionamento do DMAC de 500MHz, bloco de 1Kwords de 16 bits
- ✓
47. Determine a taxa de transferência de pico (expressa em Bytes/s) de um DMAC 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 (equivalente a $1*T_{BC}$) para efetuar uma operação de leitura ou escrita por esse DMAC. O tempo mínimo entre operações elementares deverá ser de $1T_{BC}$
48. Resolva o problema anterior, mas considerando agora as seguintes condições: X
- controlador de 32 bits, frequência do DMAC de 120 MHz, $1T_{BC} = 2$ ciclos de relógio, tempo mínimo entre operações elementares de $2 T_{BC}$
 - controlador de 32 bits, frequência do DMAC de 80 MHz, $1T_{BC} = 2$ ciclos de relógio, tempo mínimo entre operações elementares de $3 T_{BC}$
 - controlador de 16 bits, frequência do DMAC de 120 MHz, $1T_{BC} = 2$ ciclos de relógio, tempo mínimo entre operações elementares de $2 T_{BC}$
 - controlador de 16 bits, frequência do DMAC de 200 MHz, $1T_{BC} = 1$ ciclos de relógio, tempo mínimo entre operações elementares de $1 T_{BC}$
49. Suponha um DMAC 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).
- determine a taxa de transferência desse DMAC (expressa em Bytes/s), supondo um funcionamento em modo bloco.
 - determine a taxa de transferência de pico desse DMAC (expressa em Bytes/s), supondo um funcionamento em modo "cycle-stealing" e um tempo mínimo entre operações elementares de 1 ciclo de relógio ("fetch", $1T$ mínimo, "deposit", $1T$ mínimo).
 - Repita as alíneas anteriores supondo um DMAC dedicado com as características referidas anteriormente.

TIMERS

$$\rightarrow f_{out_1} = \frac{f_{in}}{k_1} \Rightarrow f_{out_2} = \frac{f_{out_1}}{k_2} \Leftrightarrow f_{out_2} = \frac{f_{in}}{k_1 k_2} \Leftrightarrow f_{out_2} = \frac{f_{in}}{K_1 K_2}$$

50. 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_{in}) e a frequência à saída do segundo (f_{out}).
51. 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 usada em aplicações práticas. → Percebam que nem todo sinal é digital logico e durante um ciclo completo do sinal
52. 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:
- frequência de entrada do *timer* for 20MHz e $k = 1999$
 - frequência de entrada do *timer* for 40MHz e $k = 1249$
 - frequência de entrada do *timer* for 80MHz e $k = 32767$
 - frequência de entrada do *timer* for 2MHz e $k = 1023$
53. 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.

$$f_{out} = \frac{20\ 000\ 000}{1999 + 1} = \frac{20\ 000\ 000}{20\ 000} = 1\ 000\ Hz$$

$$T_{out} = \frac{1}{10\ 000} = 100\ \mu s$$

Timers e Divisões de Frequência

• Para que servem? Gerar atrasos, medir tempo e criar sinalas periódicas

• Divisão de Frequência - podem ter 2 timers ligados em cascata com fatores K_1, K_2

$$f_{\text{out}} = \frac{f_{\text{in}}}{K_1 \cdot K_2}, \text{ o período de saída vai ser } T_{\text{out}} = \frac{1}{f_{\text{out}}}$$

• Duty Cycle e PWM (Pulse Width Modulation)

$$D = \frac{T_{\text{on}}}{T_{\text{período}}} \cdot 100\% \quad (\text{Percentagem de um sinal digital que está High durante um ciclo completo})$$

• Prescalers

• Divide a f_{clk} para ajustar a temporização do timer

• O período do timer (PR):

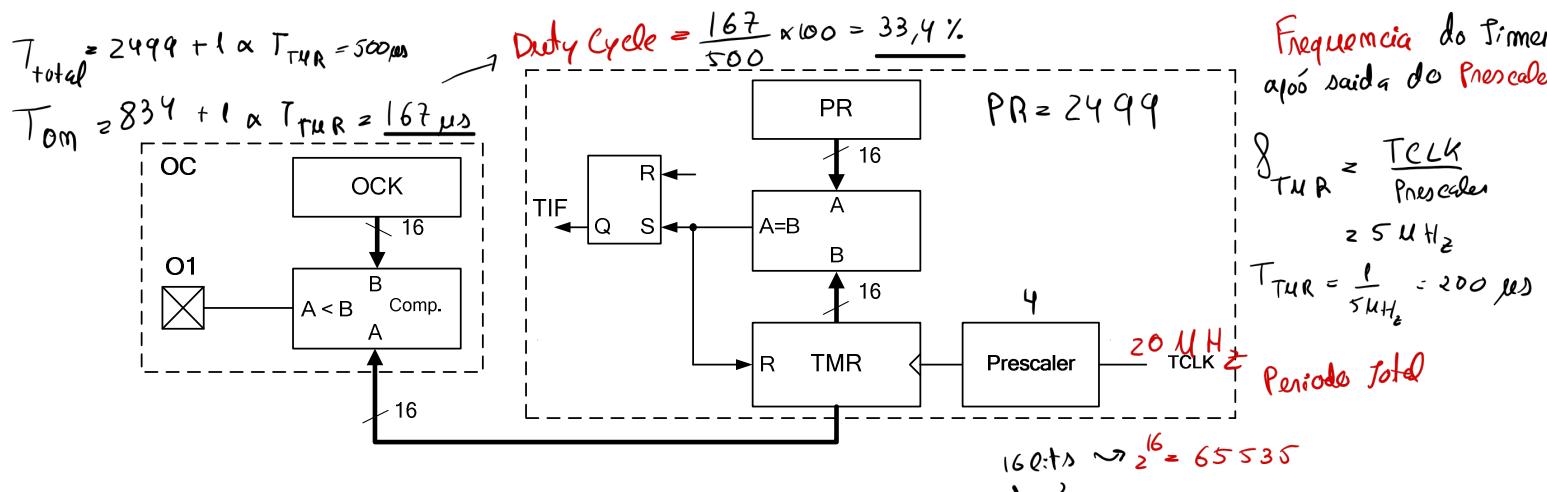
$$PR = \frac{f_{\text{clk}}}{f_{\text{designada}} \times \text{Prescaler}} - 1$$

• Watchdog Timer (WDT)

• O WDT reinicia o microcontrolador se não for resetado dentro de um tempo limite

$$T_{\text{max}} = \frac{2^N}{f_{\text{clk}}}, N - \text{número de bits do contador}$$

54. 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 respetivo *duty cycle*.



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

 - calcule o valor mínimo da constante de divisão a aplicar ao *prescaler* e indique qual o valor efetivo dessa constante
 - calcule o valor da constante PB2

$T_{out} = \frac{1}{85} \approx 11,76 \text{ ms}$

constante : 50000000 / 85 = 588235

$$T_{\text{out}} = \frac{1}{85} \approx 11,26 \text{ ms}$$

constante :

588230 ♂
K Pfeuefles

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

$$PRZ = \left(\frac{8 \text{ PBCU}}{x - 8 \text{ dem. dia.}} \right) - 1 \approx 36764 \quad \text{Número de ciclos necessários} = T_{out} \cdot 8 \text{ PBCU}$$

72 588.235 ciclos

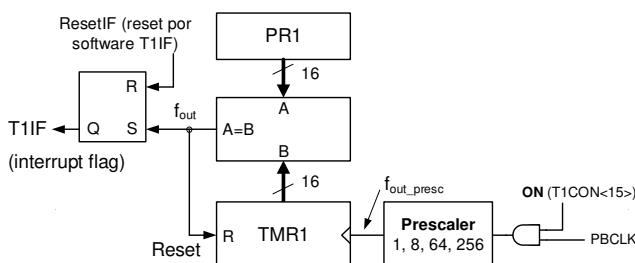
$\Leftrightarrow k_{\text{Prexclen}} \approx 8,92$

57. 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:

 - determine o valor efetivo da constante de *prescaler* que maximiza a resolução do sinal PWM
 - determine o valor das constantes PR3 e OC5RS
 - determine a resolução do sinal de PWM obtido

58. 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 OCK para obter na saída O1 um sinal periódico com uma frequência de 200Hz e um *duty cycle* de 25%.

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



60. 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:

 - frequência de entrada do *timer* for 20MHz e período do sinal à saída de 5ms
 - frequência de entrada do *timer* for 25MHz e período do sinal à saída de 1ms
 - frequência de entrada do *timer* for 40MHz e período do sinal à saída de 250ms
$$f_{\text{out}} = \frac{f_{\text{in}}}{k+1}$$

$$g_{out} = \frac{g}{k+1}$$

$$g_{\text{eff}} = \frac{1}{20000000} = 0,0000000500$$

~~2. 2. 6. 1~~ = 1 min

$$0,001 = 1 \text{ ms}$$

$$0,000001 = 1 \mu s$$

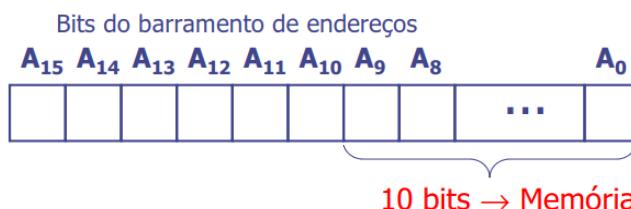
(7)

- TP1
61. 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 50MHz, contador crescente de 24 bits
 62. 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 por forma a que, na situação descrita, o *watchdog timer* não gere um reset ao processador.
 63. Repita o problema anterior admitindo agora que a frequência de entrada 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]
 64. 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 timer* 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 qua a máxima frequência de relógio aplicada na entrada do *watchdog timer*.

NOÇÕES BÁSICAS DE BARRAMENTOS / DESCODIFICAÇÃO DE ENDEREÇOS

65. Para um barramento de endereço como o indicado abaixo, que seleciona blocos de memória com 1Kbyte, suponha que no descodificador apenas se consideram os bits A15, A13 e A11, com os valores 1, 0 e 0, respectivamente.
 - a. apresente a expressão lógica que implementa este descodificador:
 - i. em lógica positiva $Y = A_{15} \cdot \bar{A}_{13} \cdot \bar{A}_{11}$
 - ii. em lógica negativa $\bar{Y} = \bar{A}_{15} \cdot A_{13} \cdot A_{11}$
 - b. indique os endereços inicial e final da gama-base descodificada e de todas as réplicas

$A_{15} \ A_9 \ A_{13} \ A_2 \ A_{11} \ A_{10} - A_0$
 1 1 0 1 0 11111111

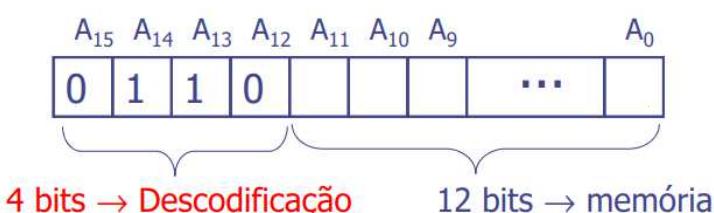


1024 blocos

Resto são bits dont care (1)

Quais são os blocos que são selados ($A_{15}=1, A_8=0, \dots$)

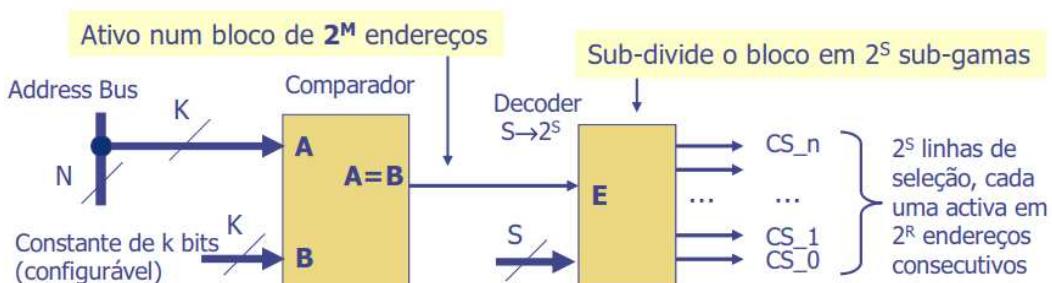
66. Considere o exemplo de um espaço de endereçamento indicado na figura abaixo, em que os blocos de memória têm uma dimensão de 4Kbyte. Admita agora não vamos descodificar os bits A14 e A12 do bloco dos 4 bits mais significativos, resultando na expressão $CS = A_{15} + A_{13}$
 - a. determine as gamas do espaço de endereçamento de 16 bits ocupadas pela memória.
 - b. determine os endereços possíveis para aceder à 15ª posição da memória.



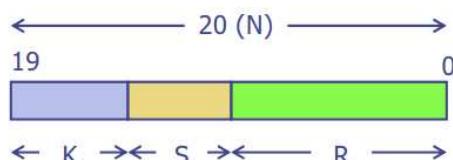
67. Escreva as equações lógicas dos 4 descodificadores necessários para a geração dos sinais de seleção para cada um dos dispositivos identificados na figura ao lado.



68. Para o exemplo da figura abaixo, determine a gama de endereços em que cada uma das linhas CS_x está ativa, com a constante de comparação 0x00102 e admitindo que o valor de S é composto por 3 bits e N=32bits.



69. 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 descodificador
 - c. indique os endereços inicial e final para a primeira, segunda e última gamas de 32k endereços, que é possível descodificar
 - d. para a última gama de endereços, indique os endereços inicial e final atribuídos a cada uma das 4 memórias de 8kbyte
 - 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 8KByte à qual está atribuído este endereço

70. 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.

- 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;
- especifique a dimensão de todos os barramentos e quais os bits que são usados;
- 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.

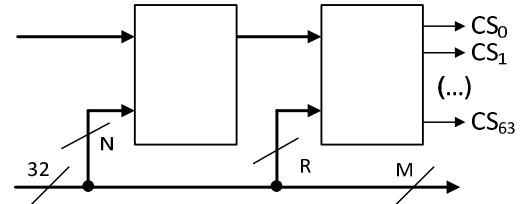
71. O periférico com 5 registos, do exercício anterior, tem um barramento de endereços com três bits. Suponha que esses bits estão ligados aos bits A0, A1 e A2 do barramento de endereços do CPU.

- usando o descodificador 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);
- 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.

72. Admita que, num espaço de endereçamento de 16 bits, um descodificador é implementado através da expressão lógica "**CE\ = A15 + A14 + A12**". Determine qual/quais a(s) gama(s) de endereço(s) que este módulo descodifica.

73. 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 das ativa em 16k endereços consecutivos, num espaço de endereçamento de 32 bits.

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



74. 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.

75. 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:

76. 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?

77. 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]:

- determine quantas réplicas da memória resultam desta descodificação;
- identifique, justificando, se os espaços de endereçamento são contíguos ou não contíguos.

78. 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 $\overline{Sel} = \overline{A_{11}} + \sum_{i=12}^{19} A_i$

- obtenha a expressão do sinal "Sel" (ativo baixo) um porto mapeado na gama de endereços **0x00400...0x007FF**, num espaço de endereçamento de 20 bits;
- 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.
79. 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
- "CE\ = A15 + A14 + A12\"**
 - "CE\ = A15 + A13\ + A12\"**
 - $\overline{CE} = A23 + A21 + \overline{A20}$** . Admita neste caso que o espaço de endereçamento é de 24 bits
 - $\overline{CE} = A31 + \overline{A29} + A28$** . Admita neste caso que o espaço de endereçamento é de 32 bits

BARRAMENTOS DE COMUNICAÇÃO SÉRIE

80. Classifique as vantagens dos barramentos série (ao nível físico) quando comparados com barramentos paralelo:
- ao nível da implementação + **simples**
 - ao nível da cablagem de suporte (em barramentos com fios) - **fios**
 - ao nível do custo + **baixo**
 - ao nível da distância de transmissão + **desperdício**
 - ao nível do débito de transmissão - **velocidade**
81. O que caracteriza topologicamente um barramento de comunicação série para podermos afirmar que este é um barramento:
- síncrono - **tem um relógio implícito e define o ritmo (simetria a informação)**
 - assíncrono - **não tem um relógio dedicado (sincronização feita por bits de start and stop)**
82. 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?
- Relógio Implicito**
 - Relógio implícito individualmente sincronizado**
 - Relógio Codificado**
83. Nos barramentos série com comunicação síncrona, o que entende por codificação Manchester?
3. **Relógio Codificado** transmissão → Bit a transmissão **XOR clock** \rightsquigarrow **CK** ↑↓↑↓↑↓ | **Data** **101010**
84. Qual a diferença entre um protocolo de comunicação série *full-duplex* e um protocolo de comunicação série *half-duplex*? **ambas são bidirecionais**, mas a *full-duplex* permite enviar e receber ao mesmo tempo numa vez que a *half-duplex* só permite fazer uma de cada vez
85. Dos protocolos de comunicação série que estudou nas aulas teóricas dê exemplos de:
- protocolos *full-duplex* - **SPI**
 - protocolos *half-duplex* - **I²C, CAN**
86. 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. **Bit** → **CAN** → **enviam bit a bit**

PROTOCOLO SPI

Byte → **I²C** → **enviam blocos de 8 bits (1 byte)**

87. Como caracterizaria o barramento SPI no que respeita:

- ao tipo de ligação entre dispositivos — **única**
- ao tipo de sincronização entre dispositivos — **Relógio Externo** (**máster que gera o CLK**)
- à natureza da transferência de dados (**bi-direcional, unidirecional**)

full-duplex

$$T_{bit} = \frac{1}{100000} = 0,00001 \times 10^6 = 10 \mu s$$

$$T_{tempo} = 10 \cdot 16 = 160 \mu s$$

$$T_{total} = 160 \cdot 20 = 3200 \mu s = 3,2 ms$$

88. 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:

- o número de sensores é 20, a frequência de relógio é de 100KHz e a resolução dos sensores é de 16 bits
- o número de sensores é 8, a frequência de relógio é de 20KHz e a resolução dos sensores é de 8 bits
- o número de sensores é 10, a frequência de relógio é de 100KHz e a resolução dos sensores é de 8 bits
- o número de sensores é 30, a frequência de relógio é de 50KHz e a resolução dos sensores é de 8 bits

89. Como caracterizaria um sistema SPI entre as seguintes opções: multi-master assíncrono; multi-master síncrono; ponto a ponto assíncrono: ponto a ponto síncrono. ✓
 só existe 1 master
 é síncrono!

90. Diria que o protocolo SPI é adequado para ligação entre dispositivos a longas distâncias? Justifique adequadamente a sua resposta. Não é, é mau por causa do ruído, problemas c/ sincronização, dem determinado do endereço

91. Numa arquitetura em que um master SPI de 8bits 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.

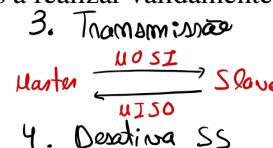
$3 \cdot 8 = 24$ bits, 3 bytes por transação SPI

$I_{MOSI}, I_{MSO}, SCLK, SS$
L Slave Select

92. 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.

I2C

2. Slave desejado



93. Considere o diagrama temporal representado abaixo. Admita que representa a comunicação I2C entre um master (μC) e um slave (ADC de 10 bits).

- a. qual o endereço do elemento slave (ADC)? $0110101 = 0x35$

- b. estamos perante uma operação de escrita ou de leitura?

- c. quantos ACKs são gerados pelo slave? 1

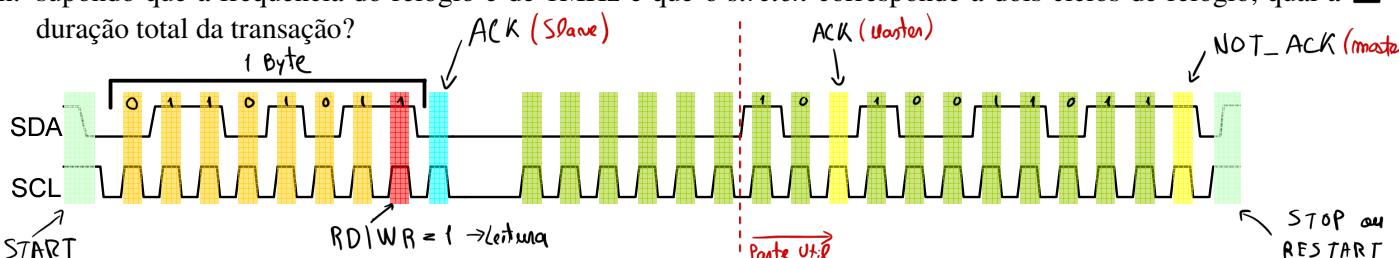
- d. quantos ACKs são gerados pelo master? 1

- e. quantos NACKs são gerados? Por quem? 1 - master

- f. qual o valor (expresso em hexadecimal) que foi fornecido pela ADC ao μC , sabendo que este começa sempre pelo MSBit? $1010011011 = 0x29B$

- g. quantas situações de clock stretch são gerados nesta transação? Por quem? 1 - master

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



94. 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. 7 bits no primeiro Byte

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

Têm 2 linhas (SDA, SCL). SDA - transporta dados SCL - sincronização pelo master

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

Quando SDA e SCL estão high ao mesmo tempo, visto que quando comece a nova transferência o SDA vai a 0 enquanto SCL = 1

97. Descreva sucintamente, no protocolo I2C, quem é responsável pela geração do sinal de relogio e como é possível assegurar a sincronização do mesmo entre master e slave - É pelo master e por causa do stretch

É o slave que manda a linha SCL de estar lento

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

Bit dominante 0

Se master coloca um 0 e o outro coloca um 1, o que medeis com 0 ganha

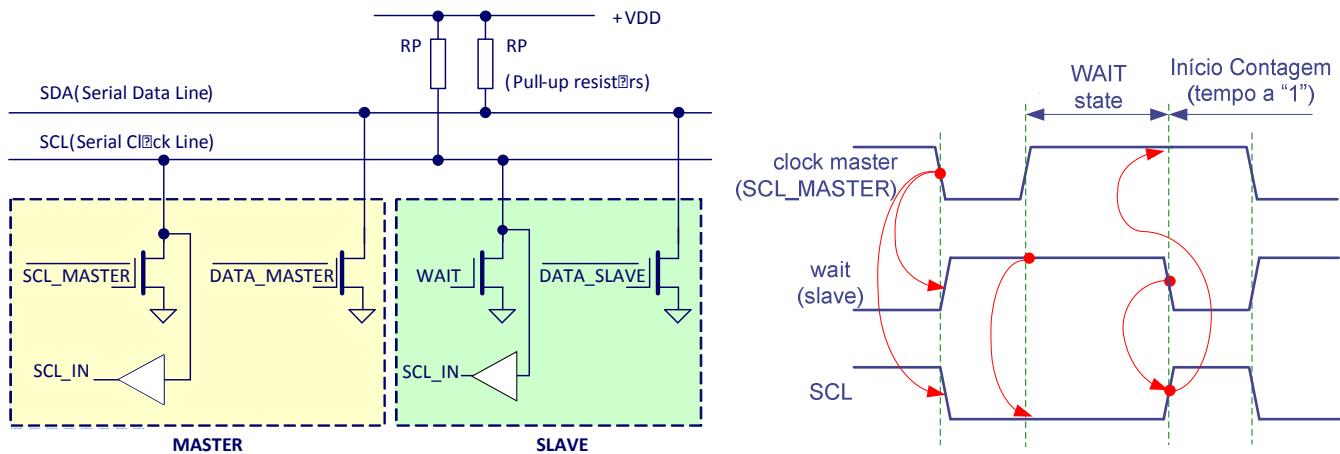
(12)

na arbitragem → Compatibilidade bit a bit

No I²C, os bits usam lógica de coletor aberto. O "0" lógico é迫使 para baixo pelo dispositivo e o "1" é obtido através das resistências de pull-up. Isto permite o funcionamento de múltiplos dispositivos no barramento sem conflitos, sendo essencial para a arbitragem e clock stretching.

99. 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.

100. 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 esta método pode ser particularmente interessante numa dada arquitetura.



RS232

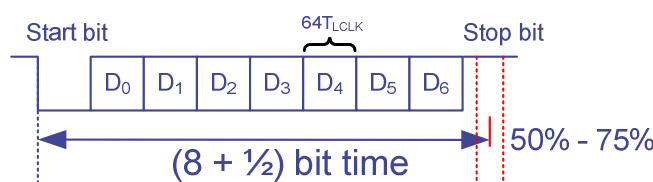
101. Como classificaria a interface RS-232 relativamente:

- a. topologia da ligação (ponto a ponto, *multi-drop*, *multi-master*, ...) e número de ligações físicas do barramento
- b. direccionalidade das comunicações (*half duplex*, *full-duplex*, ...) — **Full Duplex**
- c. natureza da sincronização (síncrona com relógio explícito, síncrona com relógio implícito, assíncrona, ...)
- d. formatação da informação (*byte oriented*, *bit oriented*, ...) *Sincronizam-se apesar do START!* / **Relógios independentes**

102. 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. *Erro de Paridade e de Synchronizing*

103. 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.

104. 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 +-37.5% do tempo de bit). Para isso calcule sucessivamente o seguinte:



- a. intervalo de validação em períodos do sinal de relógio (T_{LCLK}) 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.

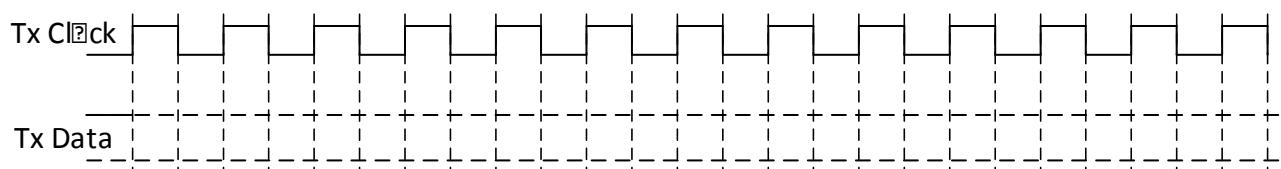
105. Repita o exercício anterior admitindo agora que o *baudrate* é 57600 bps, a trama é composta por 8 bits com paridade par e 2 stop bits, sendo o fator de sobre amostragem de 16.

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

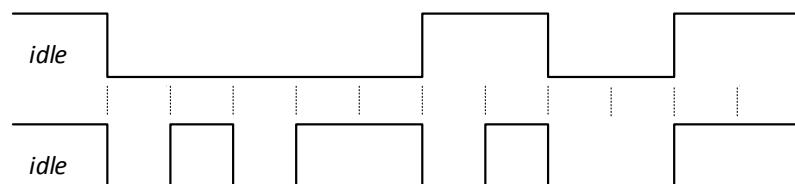
- baudrate* de 19200 bps, 7 bits de dados, 1 bit de paridade e 1 stop bit.
- baudrate* de 115200 bps, 8 bits de dados, sem bit de paridade e 2 stop bits.
- baudrate* de 9600 bps, 8 bits de dados, 1 bit de paridade e 2 stop bits.
- baudrate* de 1200 bps, 7 bits de dados, sem bit de paridade e 1 stop bit.

$$\begin{aligned} \text{Taxa Líquida} &= \text{baudrate} \cdot \left(\frac{\text{nº de Bits de dados}}{\text{nº total de bits}} \right) \\ &= 19200 \cdot \left(\frac{7}{10} \right) = 13440 \text{ bps} \end{aligned}$$

107. 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, de stop e de paridade.

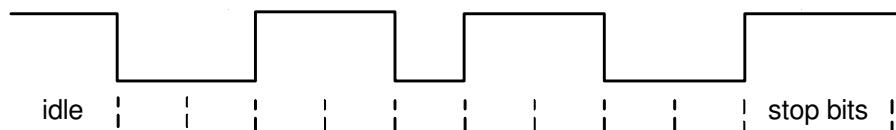


108. Um dispositivo com interface RS-232, configurado para transmitir com 7 bits de dados, paridade par e 2 stop bits, produz as duas tramas representadas nos diagramas seguintes que são recebidas por outro dispositivo RS-232 incorretamente configurado para 8 bits de dados, sem paridade e 1 stop bit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



109. 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 detetado algum error e porquê. Caso não seja detetado nenhum erro, determine, em hexadecimal, qual o valor recebido.

110. 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 detetado algum error e porquê. Caso não seja detetado nenhum erro, determine, em hexadecimal, qual o valor recebido.



DEVICE DRIVERS

111. Se tivesse que identificar as principais características/objetivos de um *device driver* como o descreveria na perspetiva:

- do Sistema Operativo ou das aplicações *camada de abstracção*
- do lado da interface com o dispositivo periférico *Tradutor*

112. Quando se interliga um periférico com um sistema de processamento, quem é responsável por fornecer o software do *device driver*? Porquê? *Fabricante do dispositivo, Igreja manda que minguem*

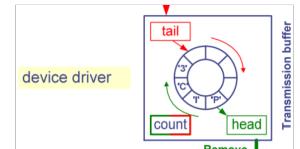
113. Nas aulas teórico práticas foi apresentado um *device driver* para um dispositivo UART (RS-232C). Esse *device driver* usava, para a transmissão e para a receção, dois buffers circulares.

- O que entende por buffer circular? *Conjunto / Estrutura de dados, que permite a leitura ou escrita de informação como se fosse um ciclo*
- Quais são as principais variáveis necessárias para gerir um buffer circular? *tail, head, count*
- Destas variáveis há uma cuja gestão é gerida pela aplicação e pelo *device driver*. Qual é essa variável e que cuidados há a ter com ela quando é atualizada do lado da aplicação? *Count*

114. Descreva por palavras suas o que se entende por secção crítica num trecho de código?

CAN

Secção do código em que as interrupções estão desativadas



115. Como classificaria o protocolo CAN relativamente:

- à topologia da ligação (ponto a ponto, multi-drop, multi-master, ...)
- direccionalidade das comunicações (half duplex, full duplex, ...)
- natureza da sincronização (síncrona com relógio explícito, síncrona com relógio implícito, assíncrona, ...)
- formatação da informação (byte oriented, bit oriented, ...)

relógios independentes

116. No barramento CAN a codificação das tramas de dados utiliza a técnica de "bit stuffing". Descreva sucintamente o motivo por que esta técnica é aplicada e em que consiste. *Colocar um bit de polaridade diferente depois de 5 bits iguais para a sincronização ocorre quando ocorre alguma transição (1→0, 0→1)*

117. No barramento CAN, na composição de uma trama de dados existe um campo de 11 bits designado por "identifier". Descreva sucintamente qual a finalidade deste campo. *Identificador da mensagem. Função: arbitragem entre masters (ID mais baixo → maior prioridade) !*

118. É ou não é verdade que, no barramento CAN, qualquer nó da rede pode desempenhar o papel de master. Se sim, será possível que dois masters enviem mensagens em simultâneo? Porquê? *Sim, tam de ledin (Multi-master)*

Não!

119. Descreva sucintamente o mecanismo através do qual, no barramento CAN, é realizado o controlo de acesso ao meio (arbitragem) quando mais do que um *master* tentam enviar mensagens em simultâneo. *ID mais Baixo → + prioridade Bit Dominante → Ø*

120. Quando, num barramento CAN, um *master* envia uma trama de dados, quantos dispositivos irão receber essa mensagem? Porquê? *Todos, Igreja a transmissão é broadcast*

121. O protocolo adotado pelos barramentos CAN apresenta uma muito elevada capacidade de detetar erros do lado dos receptores (probabilidade de não detetar um erro inferior a $4,7 \times 10^{-11}$). Para tal, este protocolo recorre a um total de cinco técnicas complementares para detetar erros. Nomeie e descreva sucintamente cada uma dessas técnicas.

1. CRC Error
2. ACK Error
3. Bit error
4. Stuffing Error

122. Admita que, numa aplicação a usar CAN 2.0A (trama com identificador standard), o mecanismo de aceitação de mensagens do controlador CAN foi configurado com os seguintes valores: máscara=0x7FA, filtro=0x5C0. Determine, nesta situação, quais os identificadores de mensagens que são aceites e passadas ao software que está a usar o controlador.

Mask bit	Filter bit	Message Id bit	Res. bit
0	X	X	Accept
1	0	0	Accept
1	0	1	Reject
1	1	0	Reject
1	1	1	Accept

123. Resolva novamente o problema anterior admitindo que o mecanismo de aceitação de mensagens do controlador CAN foi configurado com os seguintes valores:

- máscara=0x4CC, filtro=0x088.
- máscara=0x7FF, filtro=0x253.
- máscara=0x7F0, filtro=0x0A0.

0x01111111010 - MASCARA

0x0101110000000 - FILTRO

010111000000

- Os *zeros* tem de ser assim
- Os *verdes* são 0, Igreja a máscara correspondente está a 0

(15)

Comparação entre SPI, I²C, CAN e RS-232

Característica	SPI Ponto-a-ponto	I ² C Multimídia	CAN Multimídia	RS-232 Ponto-a-ponto
Tipo de comunicação	Full-Duplex	Half-Duplex (multi-mestre)	Half-Duplex (multi-mestre)	Full-Duplex (ponto-a-ponto)
Número de fios	4 (MOSI, MISO, SCLK, SS)	2 (SDA, SCL)	2 (CAN_H, CAN_L)	3 (TX, RX, GND)
Distância típica	Curta (cm a metros)	Curta (poucos metros)	Longa (centenas de metros)	Longa (até 15 m ou mais)
Velocidade típica	Alta (dezenas de Mbps)	Média (~1 Mbps)	Média (~1 Mbps)	Baixa (~115200 bps típico)
Topologia	Mestre-Escravo	Multi-mestre, multi-escravo	Multi-mestre, multi-escravo (bus)	Ponto-a-ponto
Sincronismo	Síncrono (clock dedicado)	Síncrono (no barramento)	Assíncrono com arbitragem	Assíncrono
Endereçamento	Não (via pinos SS/CS)	Sim (7 ou 10 bits)	Sim (IDs nas mensagens)	Não (ligação direta)
Nº de dispositivos	Limitado (pelos pinos CS)	Até 127	Centenas	2
Deteção de colisão	Não	Sim (clock stretching)	Sim (arbitragem)	Não
Aplicações típicas	Sensores, memórias	Sensores, EEPROMs, RTCs	Automóvel, indústria	PCs, modems

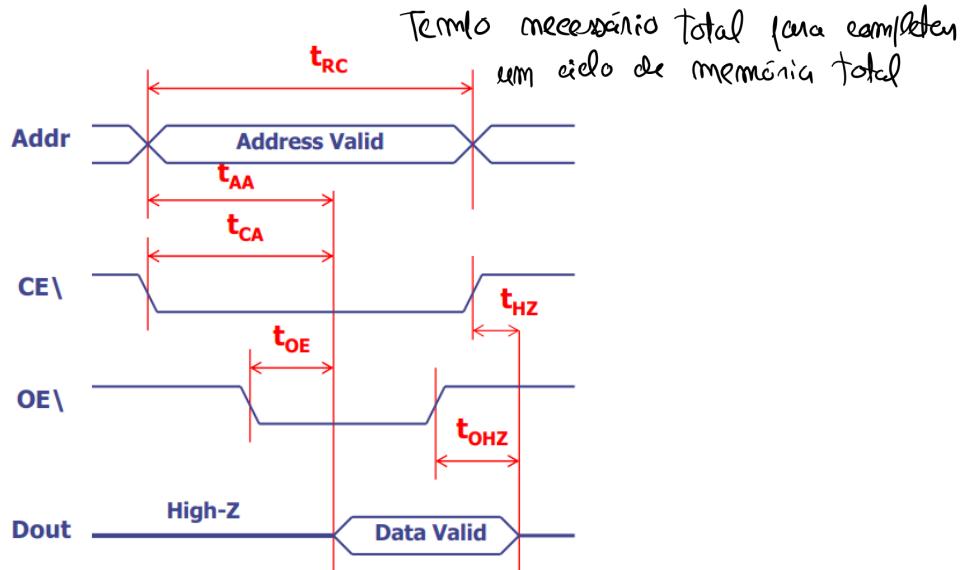
TECNOLOGIA, ORGANIZAÇÃO E FUNCIONAMENTO DE RAMs

124. Numa memória estática SRAM, uma célula de um bit é composta por seis transístores. Descreva sucintamente as vantagens e desvantagens entre a solução SRAM quando comparada com a versão de uma célula DRAM de um bit. + rápida, + fiável

125. 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 demonstre essa vantagem.

- Número de ligações

126. Quando falamos em t_{RC} (Read Cycle Time) de uma memória nas operações de leitura estamos a referir-nos especificamente a que tempo?



127. Descreva sucintamente os conceitos de:

- Access Time – Tempo entre o pedido de leitura e o momento que ele fica disponível na saída de memória
- taxa de transferência – Quantidade de dados por intervalo de tempo (MB/s)

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

- circuitos de 32k x 1 bit
- circuitos de 32k x 4 bits
- circuitos de 16k x 8 bits
- circuitos de 64k x 8 bits
- circuitos de 128k x 1 bit

$$N_{\text{endereços}} = \frac{\text{Total de endereços necessários}}{\text{Endereços por circuito}}$$

$$N_{\text{largura}} = \frac{\text{Número de bits por palavra}}{\text{Bits por circuito}}$$

$$N_{\text{Total}} = N_{\text{endereços}} \times N_{\text{largura}}$$

129. 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 compõe cada matriz.

$$N_{\text{TOTAL CÉLULAS}} = N_{\text{PALAVRAS}} \times N_{\text{BITS POR PALAVRA}}$$

$$N_{\text{MATRIZES}} = \frac{\text{Número de bits por palavra}}{\text{Bits por matriz}}$$

$$N_{\text{células}} = \frac{N_{\text{TOTAL}}}{N_{\text{matrizes}}}$$

130. Suponha que dispõe de 16 circuitos de memória de 1Mx4. Usando todos estes circuitos, determine qual a capacidade de armazenamento da memória quando:

- a largura da palavra é 4bits
- a largura da palavra é 8bits
- a largura da palavra é 32bits
- a memória tem 2M endereços
- a memória tem 8M endereços

$$N_{\text{LINHAS}} \times N_{\text{COLUNAS}} = N_{\text{células}}$$

$$\Leftrightarrow N_{\text{linhas}} \approx N_{\text{COLUNAS}} \approx \sqrt{N_{\text{células}}}$$

(Normalmente)

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

- 512M x 8 bits, implementada com uma matriz de células com 16k linhas
- 256M x 1 bit, implementada com uma matriz de células com 16k linhas
- 4G x 4 bit, implementada com uma matriz quadrada
- 1G x 1 bit, implementada com uma matriz de células com 32k linhas
- 2G x 8 bit, implementada com uma matriz de células com 64k linhas
- 256M x 1 bit, implementada com uma matriz de células com 8K linhas

132. Considere uma memória DRAM de 2Mx16, implementada com matrizes de armazenamento de 2048 colunas. Determine aproximadamente o tempo necessário para efetuar um refrescamento completo dessa memória se os seus parâmetros relativos a um ciclo de refrescamento, do tipo *RAS Only*, forem os seguintes:

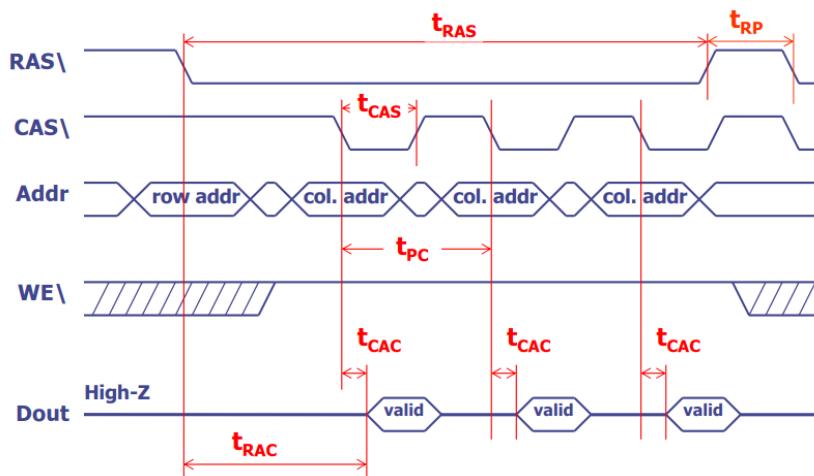
- RAS width=50 ns; Precharge time=25 ns;
- RAS width=40 ns; Precharge time=15 ns;
- RAS width=65 ns; Precharge time=30 ns;

133. Repita o exercício anterior admitindo agora:

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

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

135. 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.



MEMÓRIA CACHE

! $1KB = 1024 \text{ Bytes}$

Total
/ Bloco

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

- 1 comparador $\left\{ \begin{array}{l} \text{a. de mapeamento direto, dimensão de 16 kB, blocos de 64 bytes e 256 linhas} \\ \text{b. de mapeamento direto, dimensão de 32 kB, blocos de 128 bytes e 256 linhas} \\ \text{c. parcialmente associativa, dimensão de 64 kB, com associatividade de 4 e blocos de 64 bytes} \\ \text{d. parcialmente associativa, dimensão de 128 kB, com associatividade de 8 e blocos de 64 bytes} \\ \text{e. totalmente associativa, dimensão de 256 kB e blocos de 256 bytes} \end{array} \right.$
- $N_{comparadores} = N_{linhas}$ ou $N_{comparadores} = N_{associatividade}$ $\left\{ \begin{array}{l} \text{Número de vias} = \frac{N_{envelopes}}{N_{conjunto}} \\ \text{L} = \frac{\text{Tamanho Total Cache}}{\text{Tamanho Bloco}} = \frac{256 \cdot 1024}{256} = 1024 \text{ comparadores} \end{array} \right.$

137. 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.

$Offset = \log_2 (\text{tamanho do bloco em bytes})$

$Indice = \log_2 (\text{Número conjuntos})$

$TAG = 32 - (Indice + Offset)$

(17)

- Número taxa de colisão
 / Mais complexidade
 Mais tempo de acesso
138. Descreva, sucintamente, as vantagens e desvantagens de uma *cache* totalmente associativa quando comparada com uma *cache* de mapeamento direto.
139. Descreva, sucintamente, as várias técnicas de substituição no caso de ocorrência de um *miss* e de não haver blocos livres na *cache*.
140. 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*).
141. Explique, sucintamente, qual a finalidade do *dirty bit* e em que condições este bit é necessário.
Indica se o bloco foi modificado em relação ao original
142. Considere uma memória *cache* parcialmente associativa. Determine o número de bits do campo "group/set" do endereço quando:
 $\text{Bit}_{\text{set}} = \log_2 (S) \leftarrow S = \frac{L}{N} = \frac{L}{4} \quad L = \frac{32.1024}{32}$
- 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.
143. Explique, sucintamente, qual a finalidade do *valid bit* e quantos destes bits são necessários para uma determinada *cache* com N linhas. → *Indica se o conteúdo da linha é válido ou não. 1 - válido. 0 - inválido.*
144. Explique, sucintamente, o conceito de princípio da localidade, incluindo em quantas formas este se apresenta e dando exemplos que justificam a sua existência.
 $TMA = (Hit Ratio \times T_{Cache}) + (Miss Ratio \times (T_{Cache} + T_{Memória}))$
145. 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
146. Determine o tempo médio de acesso (em ciclos de relógio) a uma memória hierárquica sabendo que:
Hit Ratio = 90% . 60 + 10% . 100 = 54 + 10 = 64 ciclos de relógio
- 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
147. 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**
148. 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**
149. Numa *cache* com mapeamento direto, ao dividir-se o endereço real pela dimensão do bloco obtém-se o quê?

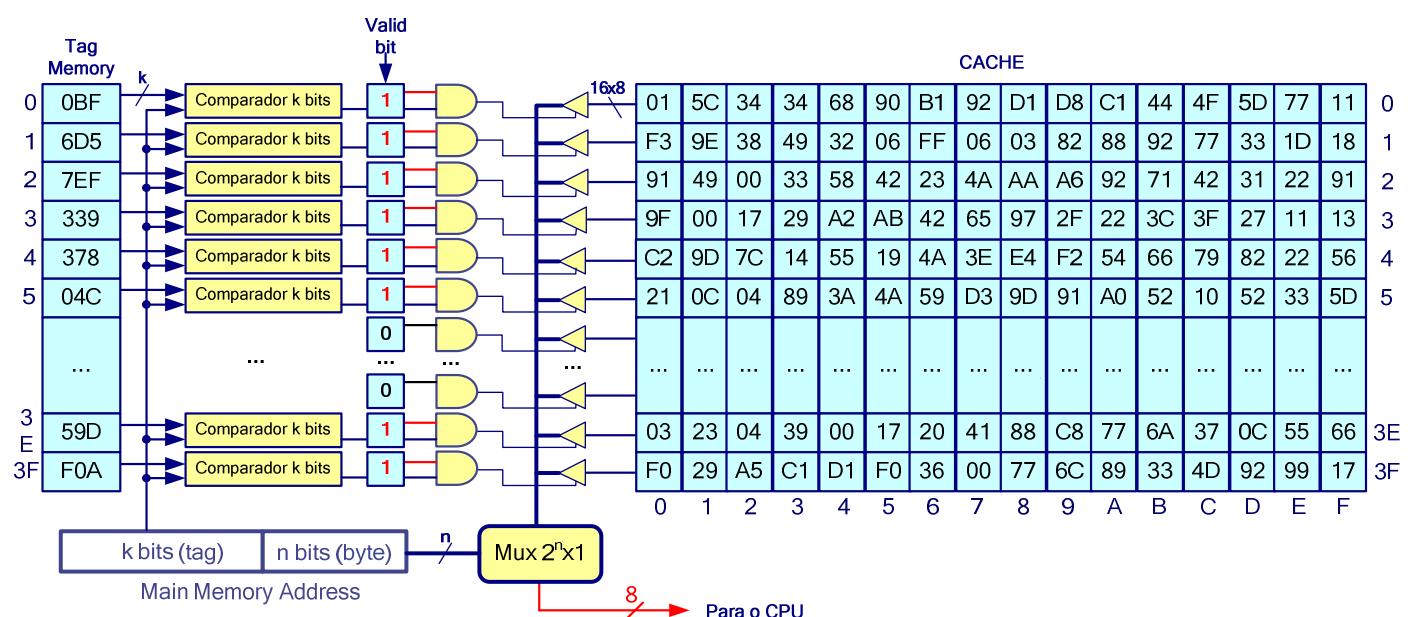
150. Indique como pode ser obtido aritmeticamente o endereço do “grupo” (linha da cache), numa *cache* com mapeamento direto.

151. Considere um espaço de endereçamento de 30 bits, em que se encontra implementada uma *cache* parcialmente associativa. Determine o número de bits respetivamente da *Tag address*, do *Group address* e do *Byte address*, admitindo que:

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

152. 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.

- determine os valores de **k** e **n**, e ainda a dimensão da *cache* em bytes
- 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



MEMÓRIA VIRTUAL

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

- as páginas têm 8 kBytes, e a *page table* de cada processo tem 256k entradas
- as páginas têm 4 kBytes, e a *page table* de cada processo tem 1M entradas
- as páginas têm 4 kBytes, e a *page table* de cada processo tem 256k entradas
- as páginas têm 16 kBytes, e a *page table* de cada processo tem 128k entradas

154. 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*.

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

- TLB miss, *page table miss*, *cache hit*
- TLB miss, *page table hit*, *cache miss*
- TLB hit, *page table miss*, *cache miss*
- TLB hit, *page table miss*, *cache hit*
- TLB hit, *page table hit*, *cache hit*

156. Na técnica normalmente designada por "memória virtual" a que é igual o número de entradas da *page table*?
157. Descreva sucintamente no que consiste a tradução de endereços virtuais em endereços físicos.
158. 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 512 k
b. o espaço de endereçamento virtual é 48 bits e o número de entradas da *page table* é 2 M
c. o espaço de endereçamento virtual é 30 bits e o número de entradas da *page table* é 128 k
159. Na técnica normalmente designada por "memória virtual" o número de entradas da *page table* é sempre igual a quê?
160. Qual o significado do *valid bit* de uma entrada da *page table*, quando este se encontra ativo?
161. Para aumentar a performance de uma TLB (*Translation-lookaside buffer*) esta adota uma organização particular. Descreva-a sucintamente.
162. 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.
163. 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
164. Descreva sucintamente o procedimento adotado quando ocorre um *Page Fault*, evidenciando os vários passos que são adotados para resolver o problema.
165. 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.
166. 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.
167. 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?
168. 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*"?

169. 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
	$2^? = 64\text{ K}$	
		0xFFFFFFFF
	$2^? = 256\text{ T}$	
64		

170. 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	8 KB	
32	4 KB	
32	8 KB	
48	4 KB	

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

- a. 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						
2 KB						
4 KB						
8 KB						

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

172. 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*:

31	27 26	NN KK	0
Valid, Dirty, Read, Write, Execute	Bits não usados	PPN	

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):

Endereço	Valor
...	
0x03C14328	0xA8002743
0x03C14324	0xF0011500
0x03C14320	0x81030308
0x03C1431C	0x28007A39
0x03C14318	0xF0008F0C
0x03C14314	0x70006004
0x03C14310	0xB0003207
0x03C1430C	0xA0000105
0x03C14308	0xA000007E
...	

- a. Considerando a informação fornecida, determine o valor de **NN** e **KK**:
173. 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
174. Admita que ao ler uma instrução, o CPU gera o endereço **0x000079C0**. Indique, descrevendo sucintamente, o que acontece neste caso.
175. 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.
176. Num acesso do CPU ao endereço virtual **0x0000620C**, identifique se, e quais são, as instruções *assembly* do MIPS que poderiam estar em execução.
177. 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.
178. Repita os exercícios 172 a 177, admitindo agora que a dimensão das páginas é de **64KBytes**.