

## INTRODUÇÃO

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

O espaço em bits ou bytes que estão disponíveis para representar um endereço do sistema

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?

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

*a. byte-addressable*

Cada endereço de contém 1 byte

*b. bit-addressable*

Cada endereço de contém 1 bit

*c. word-addressable*

Cada endereço de contém 1 word (4 bytes)

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

*a. do Data Bus*

Data bus é a linha onde os dados caminham quer venham estes do CPU ou para o CPU

*b. do Address Bus*

Address Bus é a linha que identifica qual o endereço ao qual o CPU quer aceder

*c. do Control Bus*

Control Bus é a linha que indica qual a operação ou instrução que está/quer que seja feita

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.

Control Bus

## MICROCONTOLADORES (EMBEDED 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?

É o compilador que traduz código produzido de host e traduz para código que o target consiga reconhecer (normalmente código máquina)

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

Bootloader é um programa que reside sempre no microcontrolador e que apenas disponibiliza funções básicas de transferência e execução de programas

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

A arquitetura de um microcontrolador é muito menos complexa do que a arquitetura de um sistema computacional

- b. quanto à sua frequência de trabalho

A frequência de trabalho de um microcontrolador é bastante inferior ao da frequência de um CPU de um sistema computacional

- c. quanto à disponibilização de periféricos

A disponibilização de periféricos de um microcontrolador é bastante maior do que de um sistema computacional

- d. quanto ao custo

O custo de um microcontrolador é bastante inferior de um sistema computacional

- e. quanto à energia consumida

O consumo de um microcontrolador é bastante inferior de um sistema computacional

- f. quanto aos seus campos de aplicação

Os campos de aplicação são igualmente vastos??

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

É um sistema computacional especializado que executa apenas tarefas pré-definidas e com recursos limitados disponíveis. Tem um custo muito inferior a um sistema computacional no entanto tem uma área de aplicação muito vasta

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

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

b. LAT Usado para escrever valor num porto de saída

c. PORT Usado para ler os valores de um porto de entrada

(1)

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?

Sim

b. continua a ser possível ler o valor que se encontra armazenado no registo LAT?

Sim

c. se sim, qual o sinal que permite realizar essa leitura?

PORTx (RD\_PORT no esquema)

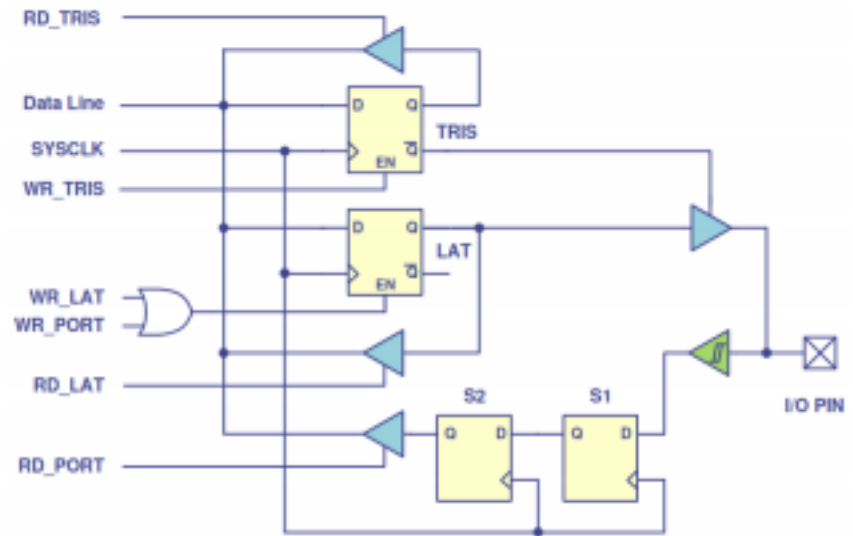
14. Num porto de I/O do PIC32 (esquema abaixo), se eu não souber qual a configuração do mesmo (saída ou entrada),

a. será possível saber essa informação programaticamente?

Sim

b. se sim, qual o sinal de controlo que permite essa operação?

O sinal TRISx onde x corresponde à letra do porto (RD\_TRIS no esquema)



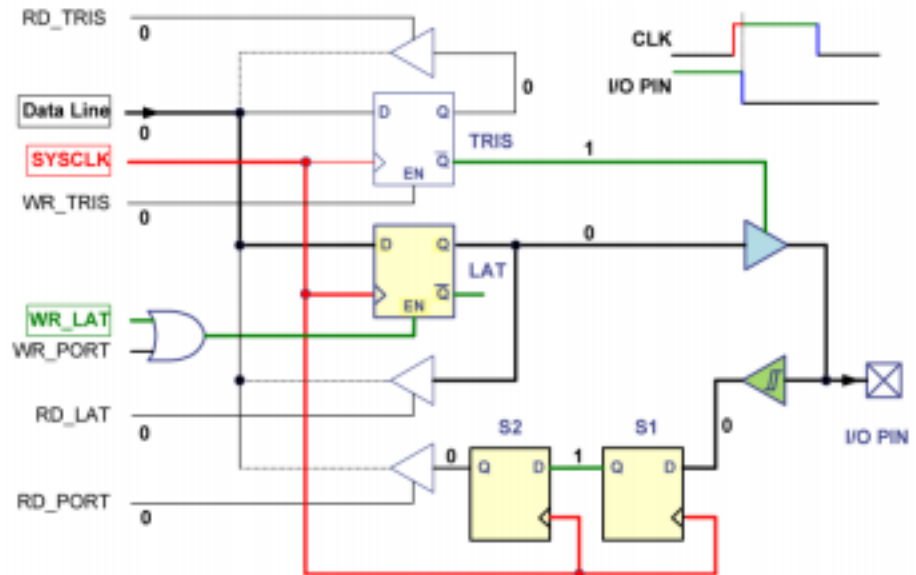
15. Considere que está a decorrer uma operação sobre um porto de I/O de um PIC32 (ver esquema abaixo – considere o instante assinalado no diagrama temporal).

a. o porto encontra-se configurado como entrada ou saída?

Como saída porque o TRIS está a 0

b. identifique e descreva qual a operação que está em curso.

Escrita do valor do valor “0” no porto de saída



(2)

16. Repita o exercício anterior para o esquema apresentado abaixo

Porto configurado como entrada

Operação de leitura do valor de entrada no porto



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?

É designada por I/O Module

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?

Polling: O CPU está constantemente a ver se o periférico está pronto para a transferência de informação

Interrupção: O CPU está a fazer as suas tarefas até que recebe a sinalização do periférico a dizer que está pronto para a troca de informação, assim que o CPU estiver pronto ele assume o controlo desta troca

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:

Polling

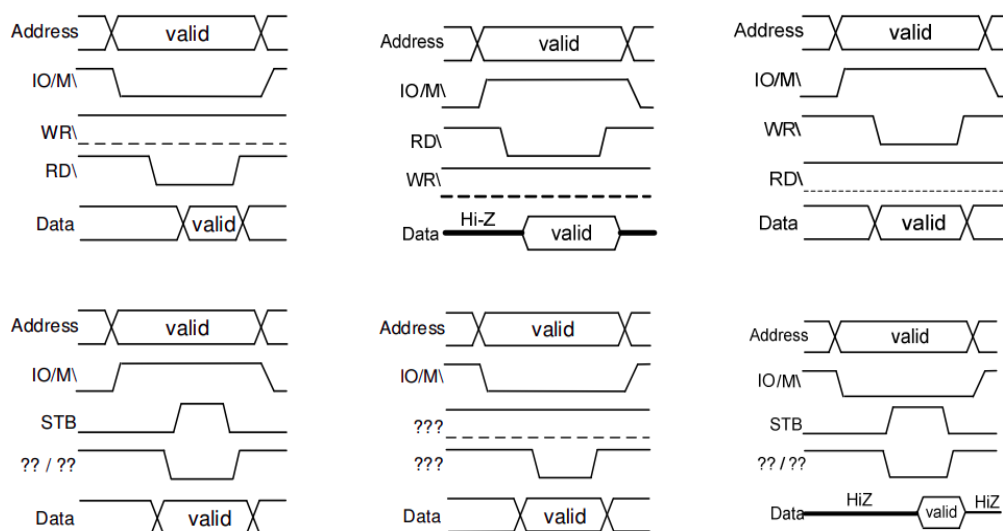
23. Quando nos referimos a um “Módulo de I/O”, estamos a referir-nos especificamente a que parte do periférico de que este módulo faz parte?

A parte que interage entre o CPU e o periférico.

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? Porta Tristate
- b. por que razão é fundamental usar esses dispositivos? Serve para diminuir o ruído

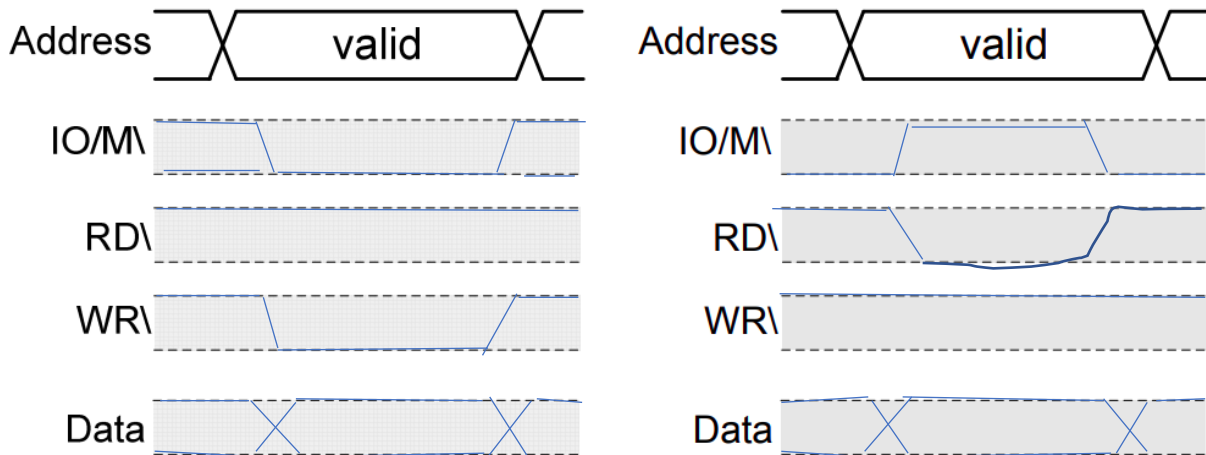
25. Os diagramas temporais que se seguem, nas várias figuras, representam operação de transferência de informação (leitura ou escrita) de/para dispositivos que podem estar mapeados em espaços de endereçamento de I/O ou de memória. Identifique, para cada diagrama qual o tipo de operação e espaço de endereçamento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



1- Leitura de Memória, 2- Leitura de Periférico, 3- Escrita na memória, 4 – Escrita no periférico, 5 -

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

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



(4)

## INTERRUPÇÕES

27. 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

Podemos ter interrupções geradas por software onde o CPU usa polling para saber quando é que a interrupção é gerada, ou a interrupção é feita por hardware

- do 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

A interrupção por hardware gera um sinal que um dos periféricos gera para sinalizar que existe uma interrupção auxiliado também por um vetor que identifica qual o periférico que gerou essa exceção. Cada periférico tem associado a si uma prioridade, este atributo é usado para quando 2 ou mais periféricos geram interrupções ao mesmo tempo

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

- "prólogo"? Prólogo é o conjunto de instruções que salvam os registos necessários antes de entrar na função de atendimento à interrupção
- "epílogo"? Epílogo é o conjunto de instruções de que repõem os valores corretos dos registos guardados antes de ser iniciado atendimento à interrupção

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

O overhead é o número de ciclos de relógio que o CPU gasta a fazer o epílogo e o prólogo de cada rotina de serviço à interrupção. Existe para saber quantas intrusões é que RSI tem verdadeiramente para atender e tratar da interrupção

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 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.

Overhead = 10 instruções, ISR = 40 instruções, Rotina = 40+10 = 50 instruções

$5M / 50 = 10^5$  rotinas,  $f = 100\text{kHz}$

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.

16M de instruções possíveis em 1 segundo

Overhead = 75 ciclos = 30 instruções

$(ISR + 30) \times 200k = 16M$  ( $\Rightarrow$ )  $200k \text{ ISR} + 6M = 16M$  ( $\Rightarrow$ )  $200k \text{ ISR} = 10M$  ( $\Rightarrow$ )  $ISR = 10M/200k = 50 \text{ inst}$

32. Recalcule a solução para o problema anterior admitindo agora que o CPU 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.

Números com virgulas duvido que apareça no exame

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.

$(70 + \text{Overhead}) \times 50k = 5M$  ( $\Rightarrow$ )  $70 + \text{Overhead} = 100$  ( $\Rightarrow$ )  $\text{Overhead} = 30 \text{ instruções}$  ( $\Rightarrow$ )  $\text{Overhead} = 60 \text{ ciclos}$

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 = 20 inst, ISR = 20 inst, Rotina = 40 inst

Podem ocorrer  $40M/40 = 1M$  rotinas por segundo,  $f = 1MHz$

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

Os periféricos ficam ligados em cadeia, cada um tem um Interrupt Acknowledge IN e um Interrupt Acknowledge OUT, o OUT está ligado ao IN do próximo periférico, este apenas fica ativo de o periférico em questão não tiver gerado nenhum Interrupt. O primeiro a receber um Interrupt Acknowledge que tenha feito requerido a interrupção vai responder com o seu vetor, sendo esta a interrupção que vai ser executada. (ou seja, a de maior prioridade)

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

Interrupt Acknowledge

(5)

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

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**

38. Descreva, sucintamente, o funcionamento de um sistema de interrupções baseado em "identificação da fonte por software" **O CPU vai ler o registo status de cada um dos periféricos para determinar qual dos periféricos requisitou a 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 CPU envia um comando ao controlador do disco (DiskCtrl): leitura de um dado sector, número de palavras
2. O CPU programa o DMA: endereço inicial da zona de dados a transferir (Controlador do disco), endereço inicial da zona destino (Memória), número de palavras a transferir
3. O CPU pode continuar com outras tarefas
4. Quando o DiskCtrl tiver lido a informação pedida para a sua zona de memória interna, ativa o sinal Dreq do DMA (sinalizando dessa forma o DMA de que a informação está pronta para ser transferida)
5. O DMA ativa o sinal BusReq, pedindo autorização ao CPU para ser bus master, e fica à espera...

6. Logo que possa, o CPU coloca os seus barramentos em alta impedância e ativa o sinal BusGrant (o que significa que o DMA passou a ser o bus master)
7. O DMA ativa o sinal Dack e o DiskCtrl, em resposta, desativa o sinal Dreq
8. O DMA efetua a transferência
9. Quando o DMA termina a transferência
10. O CPU quando deteta a desativação do BusReq desativa também o sinal BusGrant e pode novamente usar os barramentos
11. Logo que possa, o CPU atende a interrupção gerada pelo DMA
40. Qual a operação que, tipicamente, um controlador de DMA executa quando conclui um processo de transferência de informação enquanto *bus master*. **Gerar uma interrupção para ser atendida pelo CPU**
41. 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 **O busgrant é um sinal que tem a finalidade de alertar ao DMA que o CPU já acabou as suas tarefas e que o DMA pode assumir o controlo do Bus Master**
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. **No modo cycle stealing o DMA aproveita os ciclos de relógio em que o CPU não necessita do acesso ao Bus Master para ir transferindo parcialmente os seus dados**
43. Descreva, sucintamente, qual a diferença entre os modos de operação “bloco” e “burst” de um controlador de DMA.

**Bloco - O DMA assume o controlo dos barramentos até todos os dados terem sido transferidos**

**Burst - O DMA transfere até atingir o número de palavras pré-programado ou até o periférico não ter mais informação pronta para ser transferida.**

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  **$T = 2\text{ns}$ , tempo necessário =  $512 * 2 * T * 1(32 \text{ para } 32\text{bits}) = 2048 \text{ ns}$**

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

**$T = 1\text{ns}$ , Palavras = 512, precisa de 2 ciclos para transferir 32bits**

**Tempo =  $512 * 2 * T * 2(16 \text{ para } 32 \text{ bits}) = 2048 \text{ ns}$**

c. controlador de 16 bits, frequência de funcionamento do DMA de 1GHz, bloco de 512 words de 16 bits

**$T = 1\text{ns}$ , Tempo =  $512 * 2 * T * 1(16 \text{ para } 16 \text{ bits}) = 1024\text{ns}$**

d. controlador de 16 bits, frequência de funcionamento do DMA de 500MHz, bloco de 2Kwords de 16 bits  **$T = 2\text{ns}$ , 16bits para 16 bits,**

$$\text{Tempo} = 2k * 2 * T * 1 = 2048 * 4 = 8196 \text{ ns}$$

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:

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

$$T = 1\text{ns}, \text{Tempo} = 1024 * 2 * 2T = 4096 \text{ ns}$$

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

$$T = 2\text{ns}, \text{Tempo} = 2048 * 2 * 2T * 2(16 \text{ para } 32\text{bits}) = 32\,768 \text{ ns}$$

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

$$T = 1\text{ns}, \text{Tempo} = 256 * 2 * 2T * 2(16 \text{ para } 32\text{bits}) = 2048\text{ns}$$

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

$$T = 2\text{ns}, \text{Tempo} = 2048 * 2 * 2T * 1(16 \text{ para } 16\text{bits}) = 16\,384\text{ns}$$

(6)

46. Resolva as duas primeiras alíneas do problema anterior considerando agora que o controlador é dedicado. **No thanks basicamente é só dividir por 2 todos os resultados**

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 mínimo 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  $T = 4\text{ns}$ , (32 para 32bits)

$$\text{Tempo} = 512 * (2T + T + 2T + T) = 122884 \text{ ns} = 12,28 \text{ us}$$

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

$$T = 1\text{ns}, (16 \text{ para } 32\text{bits})$$

$$\text{Tempo} = 512 * 2 * (2T + T + 2T + T) = 6144\text{ns}$$

c. controlador de 16 bits, frequência de funcionamento do DMA de 500MHz, bloco de 2K words de 16 bits  $T = 2\text{ns}$ , (16 para 16bits)

$$\text{Tempo} = 2k * (2T + T + 2T + T) = 24\text{us}$$

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  $T = 2\text{ns}$ , (32 para 32bits)

$$\text{Tempo} = 512 * T = 1024 \text{ ns}$$

b. controlador de 16 bits, frequência de funcionamento do DMA de 1GHz, bloco de 4K words de 32 bits

$$T = 1\text{ns}, (16 \text{ para } 32\text{bits})$$

$$\text{Tempo} = 4\text{k} * T * 2 = 8000 \text{ ns}$$

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

$$T = 1\text{ns}, (16 \text{ para } 32\text{bits})$$

$$\text{Tempo} = 512 * T * 2 = 1024 \text{ ns}$$

d. controlador de 16 bits, frequência de funcionamento do DMA de 500MHz, bloco de 1Kwords de 16 bits

$$T = 2\text{ns}, (16 \text{ para } 16\text{bits})$$

$$\text{Tempo} = 1\text{k} * T = 2048 \text{ ns}$$

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 (equivalente a  $2 * T_{\text{BC}}$ ) para efetuar uma operação de leitura ou escrita. desse DMA. O tempo mínimo entre operações elementares deverá ser de  $1T_{\text{BC}}$

$$T = 1/100\text{Mhz} = 0,01 * 10^{-6} = 10 * 10^{-9} = 10\text{ns}; 1T_{\text{BC}} = 1\text{Ciclo}$$

$$\text{Na transferência de 4Bytes} \rightarrow 2T_{\text{BC}} + 1c + 2T_{\text{BC}} + 1c = 60\text{ns}$$

$$1\text{Byte} = 15\text{ns}; F = 66.6 \text{ MHz}$$

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

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

$$T = 8.3\text{ns}$$

$$4\text{Bytes} \rightarrow 2T + 4T + 2T + 4T \sim 100 \text{ ns}$$

$$1\text{Byte} \rightarrow 100/4 = 25\text{ns}; F = 40\text{MHz}$$

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

$$T = 1/80\text{MHz} = 12.5 \text{ ns}$$

$$4\text{Bytes} \rightarrow 2T + 6T + 2T + 6T = 200\text{ns}$$

$$1\text{Byte} \rightarrow 200/4 = 50\text{ns}; F = 20\text{MHz}$$

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

$$T = 8.3\text{ns}$$

$$2\text{Bytes} \rightarrow 2T + 4T + 2T + 4T \sim 100\text{ns}$$

$$1\text{Byte} \rightarrow 100/2 = 50\text{ns}; F = 25\text{MHz}$$

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

$$T = 5\text{ns}$$

$$2\text{Bytes} \rightarrow T + T + T + T \sim 20\text{ns}$$

$$1\text{Byte} \rightarrow 20/2 = 10\text{ns}; F = 100\text{MHz}$$

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.

$$T = 1/100\text{MHz} = 10\text{ns}$$

$$4\text{ Bytes} \rightarrow 2T = 20\text{ ns}$$

$$1\text{Byte} \rightarrow 20/4 = 5\text{ns}; F = 200\text{MHz}$$

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",  $1T$  mínimo, "deposit",  $1T$  mínimo).

$$T = 1/100\text{MHz} = 10\text{ns}$$

$$4\text{ Bytes} \rightarrow T + T + T + T = 40\text{ ns}$$

$$1\text{Byte} \rightarrow 40/4 = 10\text{ ns}; F = 100\text{MHz}$$

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

NÃO

## 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 ( $f_{in}$ ) e a frequência à saída do segundo ( $f_{out}$ ).

$$f_{in}/k_1 = f_{out1}$$

$$f_{out1} = f_{in2}; f_{in2}/k_2 = f_{out2}$$

$$f_{in} = k_1 * k_2 f_{out2}$$

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.

O duty cycle é a razão entre o tempo em que um sinal está ativo e o tempo total do sinal. O duty cycle pode ser usado praticamente para diminuir intensidades de por exemplo luzes e rotações de uma ventoinha

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:  $F_{in}/F_{out} = k+1 \Rightarrow F_{out} = F_{in}/k+1$

a. frequência de entrada do timer for 20MHz e  $k = 1999$   $F_{out} = 20\text{MHz}/2000 = 10\text{kHz} \rightarrow T = 1/10\text{kHz} = 100 \mu\text{s}$

b. frequência de entrada do timer for 40MHz e  $k = 1249$   $F_{out} = 40\text{MHz}/1250 = 32\text{kHz} \rightarrow T = 31.25\mu\text{s}$

c. frequência de entrada do timer for 80MHz e  $k = 32767$   $F_{out} = 80\text{MHz}/32768 = 2441\text{Hz} \rightarrow T = 409\mu\text{s}$

d. frequência de entrada do timer for 2MHz e  $k = 1023$   $F_{out} = 2\text{MHz}/1024 = 1953\text{Hz} \rightarrow T = 539\mu\text{s}$

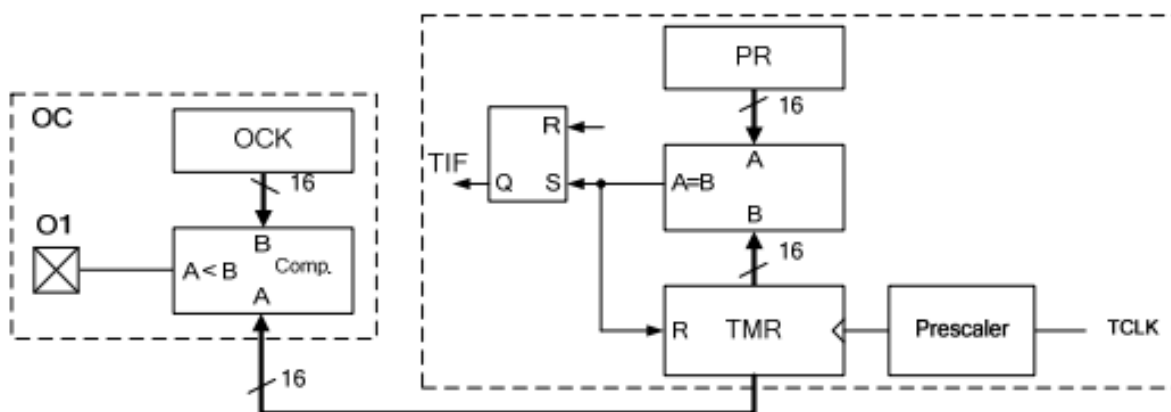
55. Alguns dos timers que estudou e utilizou têm, como último andar do temporizador, um divisor por dois. Descreva, sucintamente, qual a razão e finalidade desse divisor por dois.

Para se obter um relógio com duty cycle de 50%

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 respetivo *duty cycle*.

$$OC = (\text{duty cycle} * (PR+1))/100$$

$$834 = (\text{duty} * 2500)/100 \Rightarrow 83400 = \text{duty} * 2500 \Rightarrow \text{duty} = 83400/2500 \sim 33\%$$



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

$$K = \text{ceil}(50\text{Mhz}/(65536 * f_{out})) = \text{ceil}(50\text{Mhz}/5\text{M}) \sim 10 \Rightarrow K = 16$$

- b. calcule o valor da constante PR2

$$PR2 = (50\text{Mhz}/16/85) - 1 = 36\,740, \dots \sim 36741$$

58. Repita o exercício anterior, supondo que se está a usar o Timer T1  $K = 64$ ,  $PR1 = 9191$

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

$$K = \text{ceil}(40\text{Mhz}/(65536 * f_{out})) = \text{ceil}(40\text{Mhz}/6,5\text{M}) \sim 6, \dots \Rightarrow K = 8$$

- b. determine o valor das constantes PR3 e OC5RS

$$PR3 = (40\text{Mhz}/8/100) - 1 = (5\text{M}/100) - 1 = 50\text{k} - 1 = 49999$$

$$OC5RS = (25 * 50000)/100 = 25 * 500 = 12500$$

- c. determine a resolução do sinal de PWM obtido

$$\log_2(5\text{M}/100) = \log_2(50\text{k}) \sim 15\text{bits}$$

60. Considere ainda um timer como o da figura acima (semelhante aos dos PIC32) com a sua saída ligada a 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%:

$$PR = (20\text{Mhz}/8/200) - 1 = (100\text{k}/8) - 1 = 12500 - 1 = 12499$$

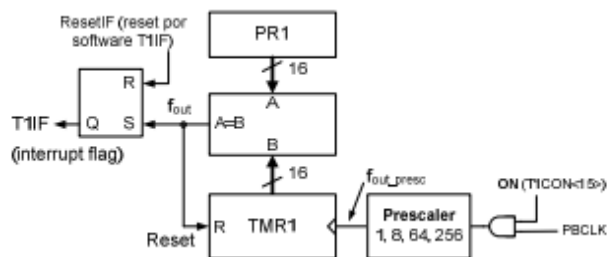
$$OCK = (25 * 12500)/100 = 3125$$

61. 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  $f_{out}$  seja de 15ms, com a melhor precisão possível:

$$F = 1/15\text{ms} = 66\text{Hz}$$

$$K = \text{ceil}(20\text{M}/(65536 * 66)) = \text{ceil}(20\text{M}/4\text{M}) \sim 5 \Rightarrow K = 8$$

$$PR1 = (20\text{M}/8/66) - 1 = 37878$$



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:  $F_{in} / F_{out} = k+1$

- a. frequência de entrada do timer for 20MHz e período do sinal à saída de 5ms

$$F_{out} = 1/5\text{ms} = 200\text{Hz}$$

$$K = (20M / 200) - 1 = 100k - 1 = 99999$$

b. frequência de entrada do timer for 25MHz e período do sinal à saída de 1ms

$$F_{out} = 1/1ms = 1kHz$$

$$K = (25M/1k) - 1 = 25k - 1 = 24999$$

c. frequência de entrada do timer for 40MHz e período do sinal à saída de 250ms

$$F_{out} = 1/250ms = 4Hz$$

$$K = (40M/4) - 1 = 1M - 1 = 999999$$

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  $T = 10\mu s; 10 * 2^{16} \mu s = 655360\mu s$

b. frequência de entrada de 20 kHz, contador crescente de 10 bits  $T = 50\mu s; 50 * 2^{10} \mu s = 51200 \mu s$

c. frequência de entrada de 50MHz, contador crescente de 24 bits  $T = 20\mu s; 20 * 2^{24} \mu s = 335544320 \mu s$

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.

$$T = 1/100k = 10\mu s$$

$$10 * 10^{-6} * 2^N = 170 * 10^{-3} \Rightarrow N = 13, \dots \Rightarrow N = 14$$

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] **PASSO**

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.

$$T * 2^{16} > 150 * 10^{-3} \Rightarrow T > 150ms / 65536 \Rightarrow T > 2,3 \mu s$$

$$F = 1/2,3\mu s = 435 kHz$$

## NOÇÕES BÁSICAS DE BARRAMENTOS / DESCODIFICAÇÃO DE ENDEREÇOS

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

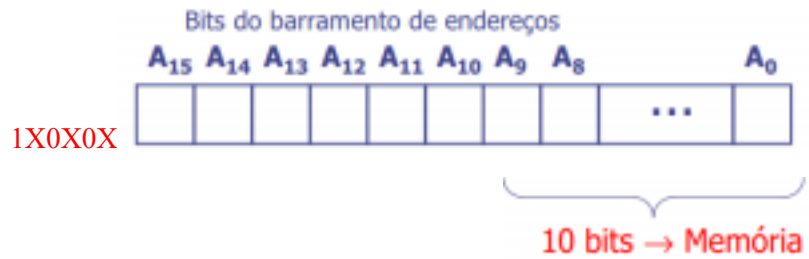
a. apresente a expressão lógica que implementa este decodificador:

i. em lógica positiva  $A15 \cdot A13 \cdot A11$



ii. em lógica negativa  $A_{15} \setminus + A_{13} + A_{11}$

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



Temos:  $A_{14}$ ,  $A_{12}$ ,  $A_{10}$  como don't cares, logo temos  $2^3$  replicas

1000 00      1000 01      1100 00      1100 01      ...

[0x8000, 0x83FF] [0x8400, 0x87FF] [0xC000, 0xC3FF] [0xC400, 0xC7FF]

68. 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  $A_{14}$  e  $A_{12}$  do bloco dos 4 bits mais significativos, resultando na expressão  $CS \setminus = A_{15} + A_{13} \setminus$



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

1000      1001      1100      1101

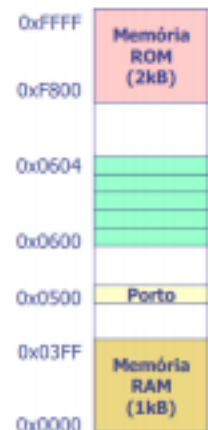
[0x8000, 0x8FFF] [0x9000, 0x9FFF] [0xC000, 0xCFFF] [0xD000, 0xDFFF]

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

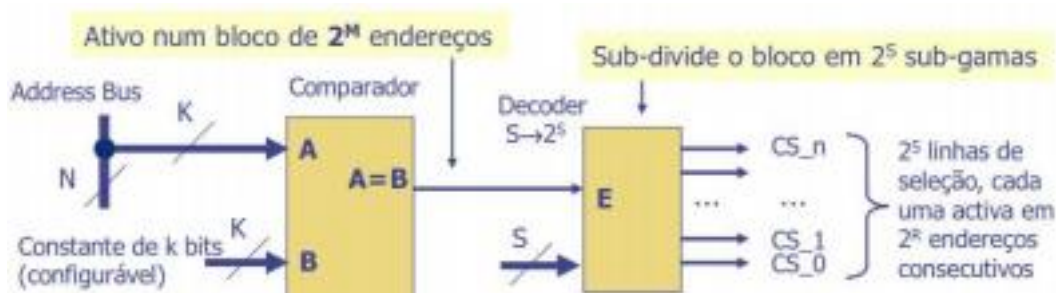
0x800E e 0x900E e 0xC00E e 0xD00E

69. 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.

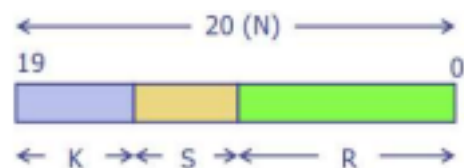
- 1ª [0x0000, 0x03FF]    0000 0000 CS= A15\ . A14\ . A13\ . A12\ . A11\ . A10\  
 2ª [0x0500]            0000 0101 CS= A15\ . A14\ . A13\ . A12\ . A11\ . A10 . A9\ . A8\  
 3ª [0x0600, 0x0604]    0000 0110  
 4ª [0xF800, 0xFFFF]    1111 1000



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=32bits.



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

$$S = 2, R = 13, K = 5$$

b. esboce o circuito digital que implementa este descodificador

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

Supondo que a combinação de K bits é 00000

Primeira gama: S = 00

0000 000 0 0000 0000 0000 = 0x00000

0000 000 1 1111 1111 1111 = 0x01FFF

[0x00000, 0x01FFF]

Segunda gama: S = 01

[0x02000, 0x03FFF]

Última gama: S = 11

[0x06000, 0x07FFF]

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

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

0x3AC45 = 0011 1010 1100 0100 0101, K = 111, S = 01

Endereço inicial e final da gama (S = 01) : [0x3A000, 0x3BFFF]

Endereço inicial e final: [0x38C45, 0x38D45]

(10)

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. **Skip**

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

**S tem 2bits, R tem 11 bits, K tem 7 bits**

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 **Skip**

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 (seleccionado 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.

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:
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.
- 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
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.
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:
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?
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 decodificador que utiliza os bits  $[A_{13}..A_{10}]$ :
- determine quantas réplicas da memória resultam desta decodificação
  - identifique, justificando, se os espaços de endereçamento são contíguos ou não contíguos

(11)

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

*Sel*

= +

$\sum$

lógica do tipo

$A_{i11}$

$i$

=

12

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

a. **"CE\ = A15 + A14 + A12\"**

b. **"CE\ = A15 + A13\ + A12\"**

c. = + +

. Admita neste caso que o espaço de endereçamento é de 24 bits d. = + + . Admita neste caso que o espaço de endereçamento é de 32 bits

## 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:
- prioridades fixas **A prioridade é atribuída à priori e esta hierarquia é depois usada para decisão**
  - critério *Last-Come/First-Served* **O último a gerar interrupção é o primeiro a aceder ao barramento**
  - critério *First-Come/First-Served* **O primeiro a gerar interrupção é o primeiro a aceder ao barramento**
  - round-robin* **O acesso ao barramento é atribuído rotativamente**
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. **Semi síncrono**
84. Identifique que elementos caracterizam um barramento paralelo de tipo síncrono e que tipos de transferência de dados são suportados pelo mesmo. **Necessitam de uma linha Bus Clock**

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*. Os elementos que caracterizam uma transferência assíncrona são o sinal *valid*, *acknowledge*. As vantagens são acentuadas quando há uma transferência entre vários dispositivos com frequência diferente

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:

- o espaço de endereçamento é de 32 bits os dados estão organizados em palavras de 32 bits
- o espaço de endereçamento é de 16 bits os dados estão organizados em palavras de 32 bit
- o espaço de endereçamento é de 32 bits os dados estão organizados em palavras de 16 bits
- o espaço de endereçamento é de 48 bits os dados estão organizados em palavras de 16 bits

87. Em barramentos *multi-master*, o que entende por situações de "starvation" São situações onde um dos dispositivos quer ser master mas tem uma *priority* baixa, quando isto acontece basta haver pelo menos um dispositivo a querer aceder ao barramento e vai "negar" o acesso ao dispositivo de prioridade baixa

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.

$$T = 1/200\text{Mhz} = 5\text{ns} \quad e \quad 10\text{inst} = 25 \text{ ciclos} = 125 \text{ ns}$$

$$4\text{Bytes} \rightarrow 125\text{ns}, 1\text{Byte} \rightarrow 31,25\text{ns}; F = 32 \text{ MB/s}$$

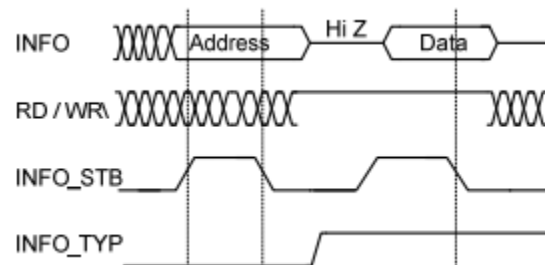
89. Considerando o diagrama temporal da figura ao lado podemos afirmar que estamos perante um ciclo de:

- identifique a natureza da operação

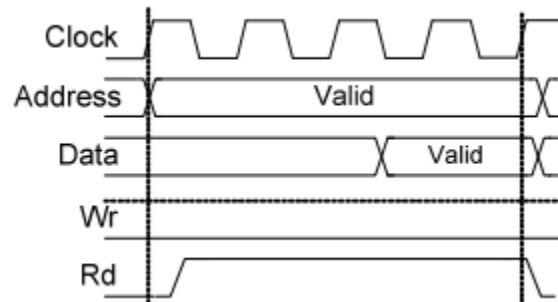
Operação de Leitura

- identifique qual o tipo de transferência que está ali representada **Transferencia de dados**

- Identifique qual a configurações da operação de transferência de dados **Microciclo**



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

$$T = 1/500\text{Mhz} = 2\text{ns}$$

$$\text{Tempo total} = T_{\text{mem}} + T_{\text{descodificacao}} + T = 12 \text{ ns} + 2,5\text{ns} + 2\text{ns} = 16,5\text{ns}$$

$$\text{O tempo da operação é } 2\text{ns} * 4\text{ciclos} = 8\text{ns}$$

$$16,5 - 8 = 8.5\text{ns}$$

$$8,5/2 = 4,5 \sim 5 \text{ ciclos de wait states}$$

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

$$T = 1/200\text{Mhz} = 5\text{ns}$$

$$\text{Tempo total} = T_{\text{mem}} + T_{\text{descodificacao}} + T = 35 \text{ ns} + 7\text{ns} + 5\text{ns} = 47\text{ns}$$

$$\text{O tempo da operação é } 5\text{ns} * 4\text{ciclos} = 20\text{ns}$$

$$47 - 20 = 27\text{ns}$$

$$27/5 = 5,4 \sim 6 \text{ ciclos de wait states}$$

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

a. *merged* Existe uma coexistencia temporal entre o Address e o Data, o que significa enquanto o Data for válido o Address também é.

b. *microciclo* A address e os dados coexistem no mesmo barramento por isso não é possível obter os dois ao mesmo tempo. Para isto temos dois sinais de strobe (strobe\_address e strobe\_data) que ficam com o nível logico a 1 quando a linha de informação contem uma address ou data respetivamente

c. *read-modify-write*

Possibilita a modificação parcial de bits

Acesso duplo com um só endereço

Primeiro a leitura depois a escrita.

d. *read\_after-write*

Acesso duplo com um só endereçamento.

Primeiro escrita depois leitura para verificação

e. *block*

Acesso múltiplo incremental, iniciado por um endereçamento.

Escreve um bloco de informação.

Um endereço é gerado e de seguida é enviada a informação, o slave incrementa o endereço automaticamente sempre que existe um pulso no write.

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

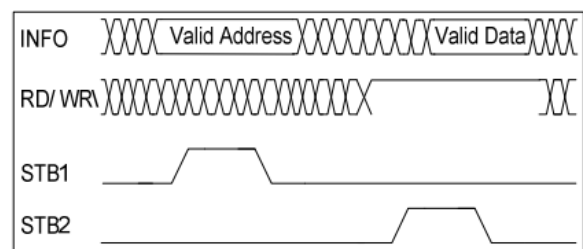
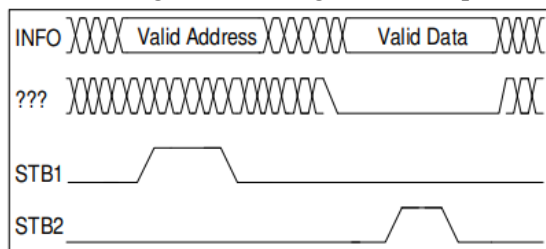
a. como caracterizaria, quanto à natureza da operação uma operação cada uma delas

1 - Escrita, 2 - Leitura

b. face à observação dos diagramas, considera que estamos perante uma operação síncrona, semi-síncrona ou assíncrona **Síncrono porque não tem sinal wait nem acknowledge**

c. que tipo de multiplexagem é possível observar no barramento INFO?

d. como designaria a configuração da operação realizada? **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. **Implementar uma solução round robin ou FIFO**

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? **Irá atribuir o barramento ao processo com mais 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 **Implementacao mais simples**

b. ao nível da cablagem de suporte (em barramentos com fios) **Existe menos um fio porque os dados so podem seguir numa direcao**

c. ao nível do custo **Menor porque é mais simples e consome menos recursos**



d. ao nível da distância de transmissão **É possível ser transmitido a longas distancias**

e. ao nível do débito de transmissão **É elevado**

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

a. síncrono **O sinal de relógio é transmitido de forma explícita através de um sinal adicional**

b. assíncrono

**Não é usado um clock na transmissão. Neste modo é necessário acrescentar bits para sinalizar o início e o fim da transmissão.**

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? **Ter uma linha extra onde há um sinal de clock explícito ou misturar o clock com a linha de dados**

98. Nos barramentos série com comunicação síncrona, o que entende por codificação Manchester? **Passa por codificar o clock do transmissor na linha de data**

99. Qual a diferença entre um protocolo de comunicação série full-duplex e um protocolo de comunicação série half duplex? **O protocolo full-duplex admite transferência de dados nas duas direcoes ao mesmo tempo (vistos que está em linhas separadas) quando o half-duplex apenas aceita numa direcao de cada vez**

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

a. protocolos full-duplex **SPI, USB 3.0, RS232**

b. protocolos half-duplex **I2C e CAN, USB2.0**

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.

**Bit a Bit: CAN, RS232, USB**

**Byte: SPI, I2C**

## **PROTOCOLO SPI**

102. Como caracterizaria o barramento SPI no que respeita:

a. ao tipo de ligação entre dispositivos **Ligacao ponto a ponto**

b. ao tipo de sincronização entre dispositivos **Síncrono**

c. à natureza da transferência de dados (bi-direcional, unidirecional) **bidirecional**

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 100KHz e a resolução dos sensores é de 16 bits

$$T = 1/100\text{KHz} = 10\mu\text{s}$$

$$16 * T * 20 = 16 * 10\mu\text{s} * 20 = 3200\mu\text{s}$$

b. o número de sensores é 8, a frequência de relógio é de 20KHz e a resolução dos sensores é de 8 bits

$$T = 1/20\text{KHz} = 50\mu\text{s}$$

$$8 * T * 8 = 8 * 50\mu\text{s} * 8 = 3200\mu\text{s}$$

c. o número de sensores é 10, a frequência de relógio é de 100KHz e a resolução dos sensores é de 8 bits

$$T = 1/100\text{KHz} = 10\mu\text{s}$$

$$8 * T * 10 = 8 * 10\mu\text{s} * 10 = 800\mu\text{s}$$

d. o número de sensores é 30, a frequência de relógio é de 50KHz e a resolução dos sensores é de 8 bits

$$T = 1/50\text{KHz} = 20\mu\text{s}$$

$$8 * T * 30 = 8 * 20\mu\text{s} * 30 = 4800\mu\text{s}$$

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.

?

106. 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*.

O master pode transferir 3 \* 8 bits ligado ao primeiro slave como MOSI

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.

O master deve configurar o clock para uma frequência igual ou inferior à suportada pelo slave.

Ativar a linha SS\ do slave com o qual vai comunicar

Em cada ciclo de relógio o master coloca na linha MOSI um bit de informação que é lido pelo slave na transição de relógio oposta seguinte, e o slave coloca na linha MISO um bit de informação que é lido pelo master na transição oposta seguinte.

No final o master e o slave trocaram o conteúdo dos seus shift-registers

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*

O protocolo I2C é um protocolo multi-master, ponto a ponto. A geração de sinal de relógio é da responsabilidade do master e a sincronização é porque a frequência usada tem que ser suportada pelo 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.

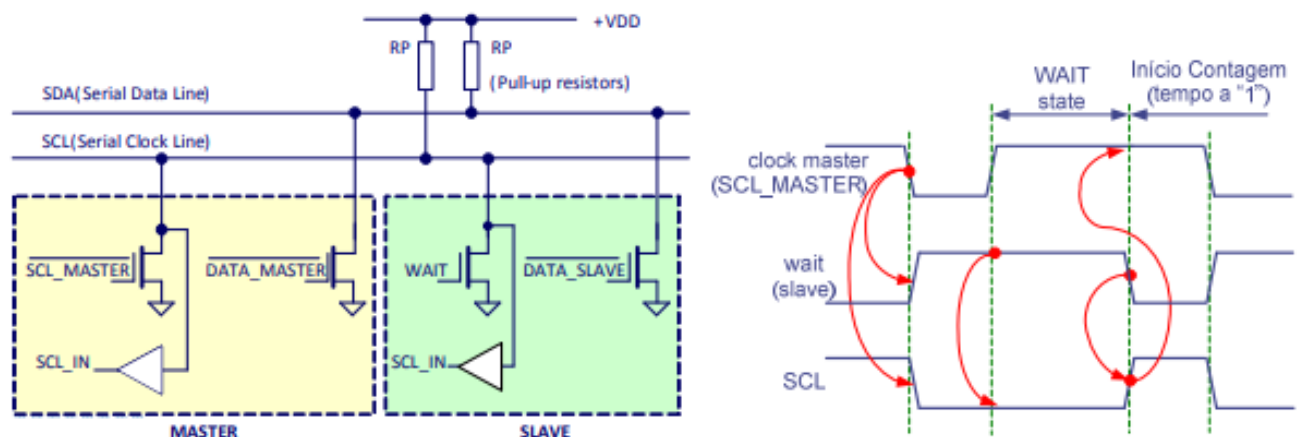
A arbitragem é feita pelo bit dominante na SCL, todos os masters têm acesso a esta linha por isso se o valor que enviou não coincide com o valor recebido o master liberta-se da linha SCL e comuta para modo slave. Quando o barramento tiver disponível (STOP bit) este poderá voltar a tentar

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.

Existem os bits START e STOP, o Start ocorre quando SCL é 1 e SDA passa para 0, e o STOP é quando SCL esta a 1 e SDA passa para 1. O tempo entre o STOP e o START é o tempo em que o barramento se encontra livre

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 esta método pode ser particularmente interessante numa dada arquitetura.

Quando o slave gera um sinal wait (bit recessivo a 0 na SCL) o master sabe que a frequência à qual está a trabalhar é demasiado elevada, por isso ele espera até que o sinal wait seja removida e recomeça a contagem de modo a que consiga gerar uma frequência à qual o slave consegue responder. Este processo é interessante porque dá a oportunidade ao Master comunicar com Slaves mais lentos de modo síncrono



## RS232

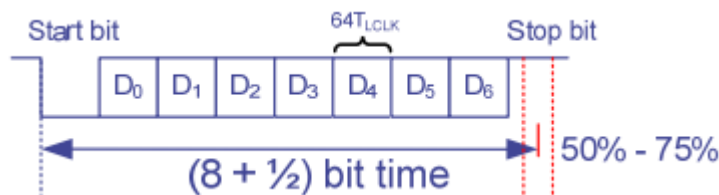
116. Como classificaria a interface RS-232 relativamente:

- a. topologia da ligação (ponto a ponto, multi-drop, multi-master, ...) e número de ligações físicas do barramento **ponto a ponto, master-slave. 3 ligacoes, 2 para data 1 para ground**
- b. direccionalidade das comunicações (half duplex, full-duplex, ...) **full duplex**
- c. natureza da sincronização (síncrona com relógio explícito, síncrona com relógio implícito, assíncrona, ...) **assíncrona**
- d. formatação da informação (byte oriented, bit 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. **O erro que pode acontecer é o bit que está a ser lido não ser correto porque o tempo de sobamostragem acumular. Erro de paridade deteta esses 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. 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 SDA (Serial Data Line) SCL (Serial Clock Line) (Pull-up resistors) 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

120. 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.

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

**10 – 19200 e 7 - x então  $x = 19200 * 7 / 10 = 13440 \text{bps}$**

- b. *baudrate* de 115200 bps, 8 bits de dados, sem bit de paridade e 2 stop bits

$$11 - 115200 \text{ e } 8 - x \text{ então } x = 115200 * 8 / 11 = 83\,781 \text{bps}$$

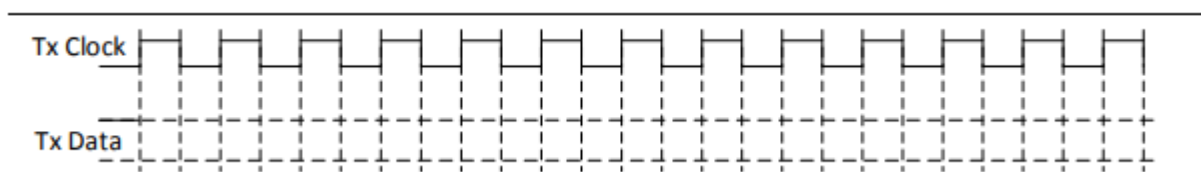
c. *baudrate* de 9600 bps, 8 bits de dados, 1 bit de paridade e 2 stop bits

$$12 - 9600 \text{ e } 8 - x \text{ então } x = 9600 * 8 / 12 = 6\,400 \text{bps}$$

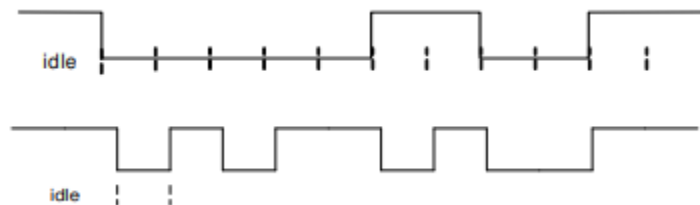
d. *baudrate* de 1200 bps, 7 bits de dados, sem bit de paridade e 1 stop bit

$$9 - 1200 \text{ e } 7 - x \text{ então } x = 1200 * 7 / 9 = 933 \text{bps}$$

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, de stop e de paridade.



123. Um dispositivo com interface RS-232, configurado para transmitir com 7 bits de dados, paridade par

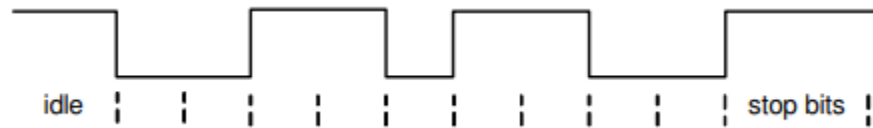


e 2 stop bits, produz as duas tramas representadas nos diagramas seguintes que são recebidas por outro dispositivo RS-232 incorretamente configurado para 8 bits de dados, sem paridade e 1 stop bit mas com o mesmo baud rate. Nestas circunstâncias, determine se no recetor vais ser detetado algum erro e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

Sim, vai ser detetado um erro de Paridade. Isto acontece porque no 9º bit (bit de paridade) o valor lido não é o correto.



## DEVICE DRIVERS

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

a. do Sistema Operativo ou das aplicações

Um programa que cria uma camada de abstração que permite o acesso do dispositivo de forma independente da sua implementação.

b. do lado da interface com o dispositivo periférico

É um modulo de software que possibilita um conjunto de funções

127. Quando se interliga um periférico com um sistema de processamento, quem é responsável por fornecer o software do *device driver*? Porquê? **O fornecedor porque esse é que tem conhecimento de como o periférico funciona**

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

a. O que entende por buffer circular? Uma lista ligada em que a tail está ligada à head

b. Quais são as principais variáveis necessárias para gerir um buffer circular? **Head, size e tail**

c. 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? **O count e este deve ser atualizado sempre que há um operação vistos que o device driver mais usar essa variavel**

129. Descreva por palavras suas o que se entende por secção crítica num trecho de código? **Uma secção de código na qual não é favorável acontecer uma interrupção**

**CAN**

130. Como classificaria o protocolo CAN relativamente:

a. à topologia da ligação (ponto a ponto, multi-drop, multi-master, ...) e número de ligações físicas do barramento **multi-master e multi-drop**

b. direccionalidade das comunicações (half duplex, full-duplex, ...) **half duplex**

c. natureza da sincronização (síncrona com relógio explícito, síncrona com relógio implícito, assíncrona, ...)

d. formatação da informação (byte oriented, bit oriented, ...) **bit oriented**

131. 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. **Consiste em introduzir um bit de polaridade inversa quando 5 bits seguidos tem a mesma polaridade. Usado para efeitos de sincronização**

132. 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. **Identifica o tipo de informação a ser enviada**
133. 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 qualquer um pode desempenhar o papel de master mas so pode haver apenas 1 porque há arbitragem feita pelo ID da mensagem**
134. 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.
135. Quando, num barramento CAN, um *master* envia uma trama de dados, quantos dispositivos irão receber essa mensagem? Porquê? **Todos os dispositivos recebem a mensagem porque não há nenhum método de endereçamento estamos presentes de uma situação (broadcast)**
136. O protocolo adotado pelos barramentos CAN apresenta uma muito elevada capacidade de detetar erros do lado dos recetores (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.

## CRC

### Bit Stuffing

137. 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

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

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



139. No barramento CAN existem quatro tipos de tramas: “*Data Frame*”, “*Remote Transmission Request Frame*”, “*Error Frame*” e “*Overload Frame*”. Descreva de forma sucinta qual a finalidade de cada um destes tipos de tramas.

## USB

140. O protocolo USB suporta quatro tipos de transferência de dados. Descreva, para cada uma delas, as suas principais características no que se refere à sua periodicidade, largura de banda, latência e garantia de entrega. Dê exemplos de cenários de aplicação em que cada um destes tipos de transferência é ou pode ser usado:
- a. transferências de controlo
  - b. transferências “*bulk*”
  - c. transferências isócronas
  - d. transferências de interrupção
141. Descreva sucintamente a topologia das ligações físicas, no protocolo USB, nomeadamente quanto à sua configuração, número de níveis e número máximo de dispositivos suportados.
142. Como classificaria o protocolo USB (2.0) relativamente:
- a. à topologia da ligação (ponto a ponto, *multi-drop*, *multi-master*, ...)
  - b. direccionalidade das comunicações (*half duplex*, *full-duplex*, ...)
  - c. codificação e transmissão de dados
  - d. formatação da informação (*byte oriented*, *bit oriented*, ...)
143. No protocolo USB cada dispositivo ligado ao *host-controller* é genericamente designado por “*function*”. Uma *function* pode assumir um de dois papéis. Identifique esses papéis e as suas principais características.
144. Descreva sucintamente, no protocolo USB, os objetivos do processo normalmente designado por enumeração.
145. No protocolo USB, o sistema de endereçamento é composto por três campos. Descreva sucintamente o objetivo de cada um desses campos e os valores mínimo e máximo que cada um pode ter.

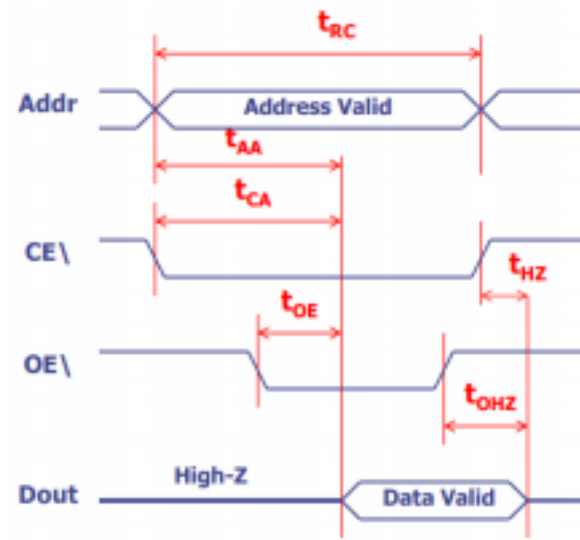
(19)

146. O mecanismo de comunicação um dispositivo USB e o “*host controller*” é feito através de canais virtuais. Qual a designação desses canais e o que interligam? Esses canais permitem comunicação bi-direcional? Porquê?

## TECNOLOGIA, ORGANIZAÇÃO E FUNCIONAMENTO DE RAMs

147. 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 e a versão de uma célula DRAM de um bit.
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 demonstre essa vantagem.

149. Quando falamos em  $t_{RC}$  (*Read Cycle Time*) de uma memória nas operações de leitura estamos a



referir-nos especificamente a que tempo?

150. Descreva sucintamente os conceitos de:

a. *Access Time*

b. taxa de transferência

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  $(128/32) * (8/1) = 4 * 8 = 32$  circuitos
- b. circuitos de 32k x 4 bits  $(128/32) * (8/4) = 4 * 2 = 8$  circuitos
- c. circuitos de 16k x 8 bits  $(128/16) * (8/8) = 8 * 1 = 8$  circuitos
- d. circuitos de 64k x 8 bits  $(128/64) * (8/8) = 2 * 1 = 2$  circuitos
- e. circuitos de 128k x 1 bit  $(128/128) * (8/1) = 1 * 8 = 8$  circuitos

152. Admita que dispõe de uma memória estática SRAM de  $256k \times 8$  (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.

$$265k = 2^{18}$$

$$8 \text{ matrizes de } (2^9 * 2^9) = (256 \times 256)$$

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

(20)

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

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

$$\text{linhas} = 2^{21} / 2^{11} = 2^{10} \text{ linhas} = 1024 \text{ linhas}$$

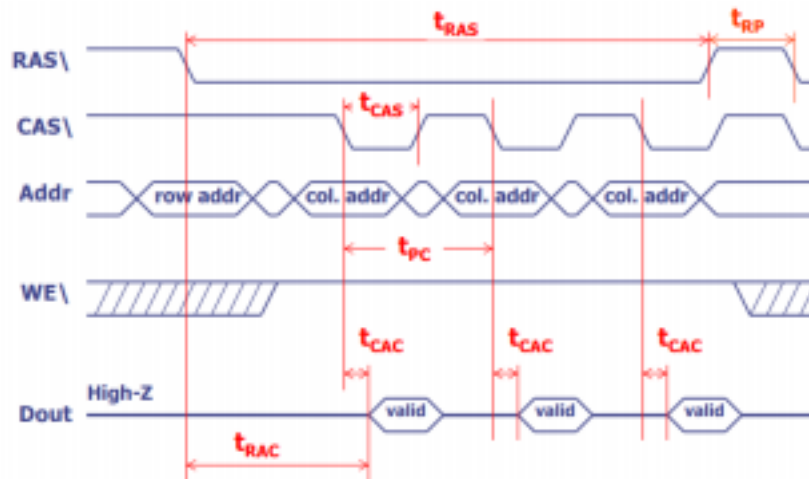
- a. RAS *width*=50 ns; Precharge time=25 ns:  $1024 * (50 + 25) = 76\,800\text{ns}$
- b. RAS *width*=40 ns; Precharge time=15 ns:  $1024 * (40 + 15) = 56\,320\text{ns}$
- c. RAS *width*=65 ns; Precharge time=30 ns:  $1024 * (65 + 30) = 97\,280\text{ns}$

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.

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

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.



## 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
- (21)
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.
161. Descreva, sucintamente, as vantagens e desvantagens de uma cache totalmente associativa quando comparada com uma cache de mapeamento direto.
162. Descreva, sucintamente, as várias técnicas de substituição no caso de ocorrência de um *miss* e de não haver blocos livres na *cache*.
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*).
164. Explique, sucintamente, qual a finalidade do *dirty bit* e em que condições este bit é necessário.
165. Considere uma memória *cache* parcialmente associativa. Determine o número de bits do campo "*group/set*" do endereço quando:
- 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.
  - 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.
  - 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.

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

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

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

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

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**

(22)

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**

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

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 respetivamente da *Tag address*, do *Group address* e do *Byte address*, admitindo que:

a. a cache é de 64Kbytes, a associatividade é 4 e o bloco é composto por 16 bytes

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

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

## 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  $2^x = 8\text{kbytes} (=) x = 13\text{bits}$ ,  $2^y = 256\text{kbytes} (=) y = 18\text{bits}$ .  $N = x + y = 31\text{bits}$
- b. as páginas têm 4 kBytes, e a *page table* de cada processo tem 1M entradas  $4\text{kbytes} \rightarrow 12\text{bits}$ ,  $1\text{MB} \rightarrow 20\text{bits}$ ,  $N = 32\text{ bits}$
- c. as páginas têm 4 kBytes, e a *page table* de cada processo tem 256k entradas  $2^x = 4\text{kbytes} (=) x = 12\text{bits}$ ,  $2^y = 256\text{kbytes} (=) y = 18\text{bits}$ .  $N = x + y = 30\text{bits}$
- d. as páginas têm 16 kBytes, e a *page table* de cada processo tem 128k entradas  $2^x = 16\text{kbytes} (=) x = 14\text{bits}$ ,  $2^y = 128\text{kbytes} (=) y = 17\text{bits}$ .  $N = x + y = 31\text{bits}$

(23)

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*. **O sistema TLB usa uma tecnologia à qual permite identificar endereços que são utilizados recentemente, a dimensão da TLB é menor do que a page table visto que todos os endereços na TLB estão na page table mas nem todos os endereços da page table estão na TLB**

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 **Não**
- b. TLB miss, *page table* hit, *cache* miss **Sim**
- c. TLB hit, *page table* miss, *cache* miss **Não**
- d. TLB hit, *page table* miss, *cache* hit **Não**
- e. TLB hit, *page table* hit, *cache* hit **Sim**

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

180. Descreva sucintamente no que consiste a tradução de endereços virtuais em endereços físicos. A tradução consistem em manter os N bits de offset que são usados para escolher o endereço a ser acedido e converter os bits da virtual page num endereço physical page que coincide com um endereço que existe na memória

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 512 k

512kb -> 29bits,

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

182. Na técnica normalmente designada por "memória virtual" o número de entradas da *page table* é sempre igual a quê? Igual ao 179

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

O valid bit sinaliza se os dados estão presentes na memória principal (RAM) quando este está a "1" caso o valor seja "0" significa que os dados estão guardados numa memória secundária (Disco)

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

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. Algumas vantagens são: não haver erros de memória fragmentada, segurança porque cada programa tem o seu próprio endereço reservado para si, e a partilhar de memória entre programas

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

a. eficiência na utilização da memória A memória física pode ser libertada ou realocada dependendo das necessidades do programa.

b. segurança Endereços usados pelos processos não são endereços da memória física

c. transparência Cada processo tem o seu próprio espaço de endereçamento

d. partilha de memória entre processos Com o mecanismo de proteção (escrita e/ou leitura) de torna-se possível de partilhar dados entre processos

187. Descreva sucintamente o procedimento adotado quando ocorre um *Page Fault*, evidenciando os vários passos que são adotados para resolver o problema. Quando um page fault ocorre os dados são transferidos da memória secundária para a memória principal, neste pode ocorrer problemas como não haver mais memória física.

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. Quando isto acontece é necessário libertar memória física para abrir "espaço" mas antes é preciso verificar se esta memória foi alterada

e se esta foi alterada vamos ter que copiar o seu conteúdo primeiro para a memória secundária de modo a manter a informação que lá estava.

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.

O dirty bit indica se esta memória foi alterada recentemente, de modo a que indique a necessidade de salvar o conteúdo desse registo

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"?  $26 - 9 = 17\text{bits}$ ;  $2^{17}$  entradas

b. qual a dimensão do espaço de endereçamento físico?  $\text{PPN} = 16 - 4 = 12\text{bits}$

Memória física.  $12\text{bits} + 9 = 21\text{ bits}$

c. em quantas páginas está organizado o espaço de endereçamento físico?  $2^{12} = 4096$  paginas

(24)

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"?

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).

Virtual address size (n)	# Virtual addresses (N)	Maior endereço virtual (hexadecimal)
8	$2^8 = 256$	0xFF
16	$2^7 = 64\text{ K}$	0xFFFF
32	$2^{32} = 4\text{M}$	0xFFFFFFFF
58?	$2^7 = 256\text{ T}$	0xFFFFFFFFFFFFFFFF?
64	$2^{64} = 16\text{ Hexa?}$	0xFFFFFFFFFFFFFFFF



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):

n	P	# PTEs
16	4 KB 12bits	$2^4$
16	8 KB 13bits	$2^3$
32	4 KB 12bits	$2^{20}$
32	8 KB 13bits	$2^{19}$
48	4 KB 12bits	$2^{36}$

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

P	VPN	VPO	PPN	PPO	# virtual pages	# physical pages
1 KB $2^{10}$	22	10	14	10	4M	16k
2 KB $2^{11}$	21	11	13	11	2M	8k
4 KB $2^{12}$	20	12	12	12	1M	4k
8 KB $2^{14}$	19	13	11	13	512k	2k

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*:

Valid, Dirty, Read, Write, Execute	Bits não usados	PPN
------------------------------------	-----------------	-----

Considere ainda que o espaço de endereçamento virtual é de **4 GBytes (32bits)**, as páginas de memória são de **4 kBytes (12bits)** e o espaço de endereçamento físico é de **1 GByte (30bits)**. 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
	...

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

**KK – 11** porque PPN tem 12 bits

**NN - 12**

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

Tudo junto:

**0x000031F8 => VPN(20MSB) = 0x0003, OFFSET(12LSB) = 0x1F8**

**Memoria: VPN\*4 = 0xC + 0x03C14300 = 0x03C1430C**

**Valor da memoria: 0xA0000105 => PPN(12LSB) = 0x105**

**Memoria Fisica: PPN . OFFSET = 0x01051F8**

$0x000031FC \Rightarrow VPN(20MSB) = 0x0003, OFFSET(12LSB) = 0x1FC$

Memoria:  $VPN*4 = 0xC + 0x03C14300 = 0x03C1430C$

Valor da memoria:  $0xA0000105 \Rightarrow PPN(12LSB) = 0x105$

Memoria Fisica:  $PPN . OFFSET = 0x01051FC$

$0x000062C0 \Rightarrow VPN(20MSB) = 0x0006, OFFSET(12LSB) = 0x2C0$

Memoria:  $VPN*4 = 0x18 + 0x03C14300 = 0x03C14318$

Valor da memoria:  $0xF0008F0C \Rightarrow PPN(12LSB) = 0xF0C$

Memoria Fisica:  $PPN . OFFSET = 0xF0C2C0$

$0x00009FFC \Rightarrow VPN(20MSB) = 0x0009, OFFSET(12LSB) = 0xFFC$

Memoria:  $VPN*4 = 0x24 + 0x03C14300 = 0x03C14324$

Valor da memoria:  $0xF0011500 \Rightarrow PPN(12LSB) = 0x500$

Memoria Fisica:  $PPN . OFFSET = 0x500FFC$

$0x000091E4 \Rightarrow VPN(20MSB) = 0x0009, OFFSET(12LSB) = 0x1E4$

Memoria:  $VPN*4 = 0x24 + 0x03C14300 = 0x03C14324$

Valor da memoria:  $0xF0011500 \Rightarrow PPN(12LSB) = 0x500$

Memoria Fisica:  $PPN . OFFSET = 0x5001E4$

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

$0x000079C0 \Rightarrow VPN(20MSB) = 0x00007; OFFSET(12LSB) = 0x9C0$

Memoria:  $VPN*4 = 0x1C + 0x03C14300 = 0x03C1431C$

Valor da memoria:  $0x28007A39$

$5MSB = 0010\ 1; Valid \ \& \ Dirty = 0, Read = 1, Write = 0, Execute = 1$

O que vai acontecer é que o processo vai poder ler ou executar