Arquitectura de Computadores II

(Exercícios Resolvidos)

Para a resposta às 5 questões seguintes considere o trecho de código Assembly x86 e o valor dos registos internos que se apresentam de seguida:

DS = 0x12B0 **ES** = 0x3F50 **CS** = 0xA15C **SS** = 0xF000 **IP** = 0x378A **SP** = 0x7F00
AX = 0x1234 **BX** = 0x150A **CX** = 0x01F4 **DX** = 0x713A **SI** = 0x0000 **DI** = 0xFFFF

Endereço	**Mnemónica**
A15C:378A	MOV AL, [BX]
A15C:378C	PUSH AX
A15C:378D	CALL 0x5678
A15C:3790	MOV DX, 0x5A63
A15C:3793	OUT DX, AL
A15C:3795	SUB AX, BX

1) A próxima instrução a ser executada é:
a) MOV AL,[BX]
b) PUSH AX
c) CALL 0x5678
d) MOV DX,0x5A63

O CS:IP indica a próxima instrução, CS:IP = A15C:378A → MOV AL,[BX]

// resposta **a)**

2) O endereço físico de memória referenciado pela instrução "MOV AL, [BX]"
a) 0xF150A
b) 0xA2AC0
c) 0x1400A
d) 0x40A0A

Endereço linear = Segmento * 16 + offset

Na instrução vai aceder-se à memória para ir buscar dados, logo o segmento será o "Data Segment" (DS) = 0x12B0 e o offset será o o BX = 150A

12B0 * 16 + 150A = 12B00 + 150A = 1400A

// resposta **c)**

3) O valor do registo SP logo após a execução da instrução "PUSH AX" é:
a) 0x7EFF
b) 0x7EFE
c) 0x7F02
d) 0x1234

A Stack cresce no sentido dos endereços mais baixos e SP é automaticamente pré-decrementado de 2 quando é colocada nova informação (instrução "push")

SP = SP – 2 = 7F00 – 2 = 7EFE

// resposta **b)**

4) O conteúdo no topo da stack logo após a instrução "CALL 0x5678" é:

a) 0x3790
b) 0x378D
c) 0x378C
d) 0x5678

Na Stack é guardado automaticamente o endereço de retorno, como o endereço da instrução seguinte é A15C:3790, na Stack ficará 3790

// resposta **a)**

5) Após a execução da instrução "SUB AX, BX", as flags associadas à ALU tomam os valores:

a) ZERO=0; CARRY=1; SIGNAL=1; OVERFLOW=0
b) ZERO=0; CARRY=0; SIGNAL=1; OVERFLOW=0
c) ZERO=1; CARRY=1; SIGNAL=1; OVERFLOW=0
d) ZERO=0; CARRY=1; SIGNAL=0; OVERFLOW=1

Na ALU vai ser feita a operação: AX – BX

AX: 1234
\- BX: - 150A
FD2A

- O resultado não deu zero, logo ZERO=0
- 1-1-1=-1 que implica ir buscar 1 à casa da esquerda, logo CARRY=1
- O bit mais significativo é 1, logo SIGNAL=1
- O número está correctamente representado, logo OVERFLOW=0

// resposta **a)**

Para responder às 5 questões seguintes considere o trecho de codigo assembly Intel x86 e o valor dos registos internos que se apresentam de seguida:

DS=0x0400 **ES**=0x0600 **CS**=0x0200 **SS**=0x0C00 **IP**=0x000D **SP**=0x7F06
AX=0x5678 **BX**=0x0037 **CX**=0x9ABC **DX**=0x0000 **SI**=0x F234 **DI**=0xF234

Endereço	**Mnemónica**
0200:000A	MOV DX, 0xC000
0200:000D	OUT DX, AL
0200:000E	CMP SI, DI
0200:0010	JNE 0x1C
0200:0012	ADD [BX], CX
0200:0014	POP AX
0200:0015	CALL 0x0094
0200:0018	ADD SP, 2

6) A proxima instrucao a ser executada é:

a) MOV DX, 0xC000
b) OUT DX, AL
c) CMP SI, DI
d) JNE 0x1C

O CS:IP indica a próxima instrução, CS:IP = 0200:000D → OUT DX, AL

// resposta **b)**

7) O endereço fisico de memoria (com 20 bits) referenciado pela instruçao “ADD [BX], CX” é:

a) 0x00037
b) 0x00370
c) 0x04037
d) 0x00437

Endereço linear = Segmento * 16 + offset

Na instrução vai aceder-se à memória para por lá os dados (valor de CX), logo o segmento será o “Data Segment” (DS) = 0x0400 e o offset será o o BX = 0x0037
0400 * 16 + 0037 = 4000 + 0037 = 4037

// resposta **c)**

8) O valor do registo SP logo após a execucao da instrucao “POP AX” é:

a) 0x7F08
b) 0x5678
c) 0x7F00
d) 0x0014

A Stack cresce no sentido dos endereços mais baixos e SP é automaticamente pós-incrementado de 2 quando é retirada informação (instrução "pop")
SP = SP + 2 = 7F06 + 2 = 7F08

// resposta **a)**

9) O conteúdo do topo da stack apos a execucao da instrucao “CALL 0x0094” é:

a) 0x0015
b) 0x0094
c) 0x0200
d) 0x0018

Na Stack é guardado automaticamente o endereço de retorno, como o endereço da instrução seguinte é 0200:0018, na Stack ficará 0018

// resposta **d)**

10) Após a execução da instrução “CMP SI, DI”, as flags associadas à ALU tomam os valores:

a) ZERO=0; CARRY=1; SIGNAL=1; OVERFLOW=0
b) ZERO=1; CARRY=1; SIGNAL=1; OVERFLOW=1
c) ZERO=1; CARRY=0; SIGNAL=0; OVERFLOW=0
d) ZERO=0; CARRY=1; SIGNAL=0; OVERFLOW=1

Na ALU vai ser feita a operação: SI – DI

```
  SI:    F234
- DI:  - F234
       ------
        0000
```

- O resultado deu zero, logo ZERO=1
- F-F=0, não foi buscar nada a uma casa à esquerda, logo CARRY=0
- O bit mais significativo é zero, logo SIGNAL=0
- O número está correctamente representado, logo OVERFLOW=0

// resposta **C)**

11) O 80188 é:
a) Um microcontrolador
b) Um microprocessador
c) Um CPU híbrido
d) Nenhuma das anteriores está correcta.

O Intel 80188 é um microprocessador.

// resposta **b)**

12) Os microprocessadores da família X86 são:
a) RISCs
b) CISCs
c) DISCs
d) Nenhuma das anteriores está correcta.

Complex Instruction Set Computer

// resposta **b)**

13) Na arquitectura intel x86 podem ser usados os seguintes registos nas instruções de acesso à memoria externa através de endereçamento indirecto:
a) BX, BP, SI e DI
b) SP, BP, SI e DI
c) AX, BX, CX e DX
d) AX, BX, SI e DI

Nas instruções de acesso à memória por endereçamento indirecto não se pode usar SP, logo a b) não é; não se pode usar AX,CX e DX, logo a c) e d) não são.

// resposta **a)**

14) Na arquitectura intel x86 a instrução "POP AX" realiza a seguintes operações:
a) AX=[SS:SP]; SP=SP-2
b) SP=SP-2; AX=[SS:SP]
c) SP=SP+2; AX=[SS:SP]
d) AX=[SS:SP]; SP=SP+2

A Stack cresce no sentido dos endereços mais baixos e SP é automaticamente pós-incrementado de 2 quando é retirada informação (instrução "pop")

Portanto em 1º lugar retira-se a informação da Stack e coloca-se em AX, ou seja:
AX=[SS:SP]
e depois o ponteiro SP é incrementado, ou seja:
SP=SP+2

// resposta **d)**

15) Num processador da família Intel X86, no processo de chamada de uma subrotina:

a) São colocados no stack automaticamente o endereço de retorno e o conteúdo dos registos a salvaguardar.
b) O segmento e offset do endereço de retorno são armazenados automaticamente no stack.
c) É necessário garantir que se utilizam instruções de retorno especiais.
d) Nenhuma das anteriores está correcta.

É automaticamente colocado na Stack o offset do endereço de retorno.

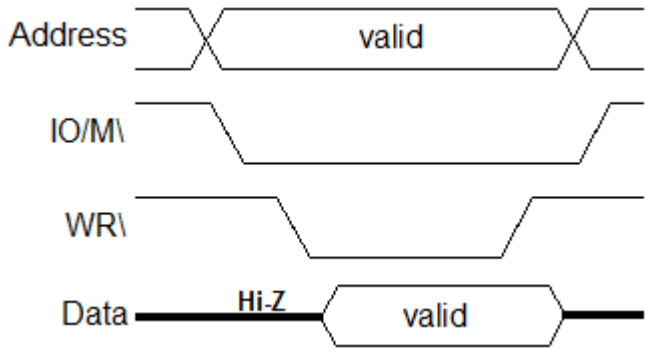
// resposta **d)**

16) Num processador que detenha sinais específicos para implementar Isolated I/O, o acesso aos periféricos:

a) Pode apenas ser efectuado por instruções específicas para I/O.
b) Pode apenas ser efectuado pelas mesmas instruções de acesso à memória.
c) Pode ser efectuado por ambos os tipos de instruções, dependendo da forma como esteja projectado o hardware.
d) Nenhuma das anteriores está correcta.

No Memory-mapped I/O, como memória e periféricos estão no mesmo espaço de endereçamento, usam as mesmas instruções. No caso do Isolated I/O, como estão em espaços de endereçamentos diferentes, tem de se usar instruções específicas.

// resposta **a)**



(relativo à questão 17)

17) O diagrama temporal da figura representa um ciclo de:

a) Leitura de um dispositivo mapeado no espaço de endereçamento de memória
b) Escrita num dispositivo mapeado no espaço de endereçamento de I/O
c) Escrita num dispositivo mapeado no espaço de endereçamento de memória
d) Leitura de um dispositivo mapeado no espaço de endereçamento de I/O

Quando o Address é válido tem-se que WR\ = 0, como WR\ é activo a 0, o diagrama representa um ciclo de escrita. Como IO/M\ = 0, está a selecionar-se o endereçamento de memória.

// resposta **c)**

19) Num espaço de endereçamento de 16 bits, um descodificador implementado através da expressão lógica "CE\ = A15 + A14\ + A12\", descodifica a(s) seguinte(s) gama(s) de endereço(s):

a) 0x8000 a 0x8FFF, 0xA000 a 0xAFFF
b) 0x8000 a 0x8FFF
c) 0x5000 a 0x5FFF, 0x7000 a 0x7FFF
d) 0x5000 a 0x5FFF

A15	A14	A13	A12
0	1	X	1

Para A13 = 0, implica endereços da forma: 5XXXh, que implica a gama: 5000h a 5FFFh
Para A13 = 1, implica endereços da forma: 7XXXh, que implica a gama: 7000h a 7FFFh

// resposta **c)**

20) Num espaço de endereçamento de 16 bits, um descodificador implementado através da gama "CE\ = A15 + A13\ + A11", descodifica a(s) seguinte(s) gama(s) de endereço(s):

a) 0x2000 a 0x37FF, 0x6000 a 0x77FF
b) 0x2000 a 0x27FF, 0x3000 a 0x37FF, 0x6000 a 0x67FF, 0x7000 a 0x77FF
c) 0x8800 a 0x8FFF, 0x9800 a 0x9FFF, 0xC800 a 0xCFFF, 0xD800 a 0xDFFF
d) Nenhuma das anteriores

A15	A14	A13	A12	A11
0	X	1	X	0

Seja Y = 0XXXb, que implica Y variar entre 0000 = 0h e 0111 = 7h

Para A13 = 0, e A12 = 0, implica endereços da forma: 2YXX, que implica a gama: 2000 a 27FF
Para A13 = 0, e A12 = 1, implica endereços da forma: 3YXX, que implica a gama: 3000 a 37FF
Para A13 = 1, e A12 = 0, implica endereços da forma: 6YXX, que implica a gama: 6000 a 67FF
Para A13 = 1, e A12 = 1, implica endereços da forma: 7YXX, que implica a gama: 7000 a 77FF

// resposta **b)**

21) A equação lógica CE\ = A18\ + A17 + A16\ + A15 + X em que X = IO/M* selecciona um bloco de memória de:

a) 128K
b) 32K
c) 16K
d) Nenhuma das anteriores está correcta.

De A0 a A14 os bits podem tomar qualquer valor. São ao todo 15 bits, logo a equação selecciona um bloco de memória de $2^{15} = 2^5 * 2^{10} = 32K$. Depois tem-se ainda o A19 a poder tomar qualquer valor, logo $2^1 = 2$, significa que existe 2 blocos de 32K.

// resposta **b)**

22) O endereço base do bloco da questão anterior é:

a) F0000H
b) A000H
c) 5000H
d) Nenhuma das anteriores está correcta.

A19	A18	A17	A16	A15
X	1	0	1	0

O endereço base, que é o endereço mais baixo, implica A19 = 0
0101b = 5h

A15 = 0, e todos os restantes bits também (para se obter o endereço mais baixo), implica o endereço base: 50000h

// resposta **d)**

23) A equação lógica CE\ = A16 + A15\ + A14 + A13\ + A12 + X em que X = IO/M* selecciona um bloco de memória de:

a) 32K
b) 8K
c) 2K
d) Nenhuma das anteriores está correcta.

De A0 a A11 os bits podem tomar qualquer valor. São ao todo 12 bits, logo a equação selecciona um bloco de memória de $2^{12} = 2^2 * 2^{10}$=4K. Depois tem-se ainda o A19,A18 e A17 a poder tomar qualquer valor, logo 2^3=8, significa que existe 8 blocos de 4K.

// resposta **d)**

24) O endereço base do bloco da questão anterior é:

a) 0C000H
b) 0A000H
c) 05000H
d) Nenhuma das anteriores está correcta.

A19	A18	A17	A16	A15	A14	A13	A12
X	X	X	0	1	0	1	0

O endereço base, que é o endereço mais baixo, implica A19,A18,A17 = 0
Tendo A11, ..., A0 = 0, o endereço base será: 0A000

// resposta **b)**

25) Num porto de entrada constituido por buffers tri-state, o sinal de activação activo baixo (EN\) é obtido a partir dos sinais CE\ e RD\ de acordo com a expressao lógica:

a) En\ = (CE\ + RD\)\
b) En\ = (CE\ . RD\)\
c) En\= CE\ . RD\
d) En\ = CE\ + RD\

O CPU vai ler de um periférico quando este periférico for selecionado pelo descodificador de endereços (CE=1) e apenas quando o CPU quiser ler (RD=1), portanto En=CE.RD

En=CE.RD ⇔ En\=(CE.RD)\ ⇔ En\=CE\+RD\ *(aplicando as Leis de DeMorgan)*

// resposta **d)**

26) Pretende desenvolver-se um protocolo para interligar dois dispositivos por meio de um barramento paralelo. É necessário suportar operações de escrita e de leitura. Deve utilizar-se:

a) Uma linha sinalizar a operação de leitura e outra linha para sinalizar a operação de escrita
b) Uma única linha que codifica leitura ("1") ou escrita ("0"), acompanhada de um sinal de strobe
c) Uma única linha que codifica leitura ("0") ou escrita ("1"), acompanhada de um sinal de strobe
d) Todas as opções acima indicadas são possíveis

Pretende-se que haja leitura, escrita ou que não faça nada. São 3 acontecimentos diferente, como $\log_2(3)=1{,}58...$ são precisos no mínimo 2 fios. Não importa que sinais tenha cada operação, o que importa é que é possivel realizar as 3 operações diferentes.

// resposta **d)**

27) Uma ligação ponto a ponto dispõe dos seguintes sinais: STROBE, ACKNOWLEDGE, R/W*. Em princípio deverá estar a utilizar um protocolo:

a) Síncrono
b) Semi-sncrono
c) Handshaken
d) Nenhuma das anteriores

Os sinais STROBE e R/W* podem ser comuns a todos. O ACKNOWLEDGE é usado para transferência múltipla assíncrona (handshaken).

// resposta **c)**

28) Os protocolos síncronos são vantajosos quando:

a) Todos os dispositivos presentes no sistema são rápidos
b) Todos os dispositivos presentes no sistema são lentos
c) Os dispositivos presentes no sistema apresentam uma velocidade homogénea
d) Os dispositivos presentes no sistema apresentam uma velocidade heterogénea

A velocidade depende da velocidade da unidade mais lenta. Este protocolo é vantajoso em relação aos outros, quando as unidades têm todas a mesma velocidade.

// resposta **c)**

29) Numa transferência do tipo síncrono:

a) É necessário um sinal de Acknowledge gerado pelo periférico.
b) A transferência efectua-se apenas nas transições do sinal de relógio.
c) O periférico sinaliza o fim da transferência através de um sinal de Wait.
d) Nenhuma das anteriores está correcta.

Nos protocolos assíncronos é que se utiliza os sinal acknowledge, nos protocolos semi-síncronos é que se utiliza o sinal wait. Nos protocolos síncronos a transferência faz-se enquanto o sinal valid está activo, e não apenas nas transições do sinal de relógio.

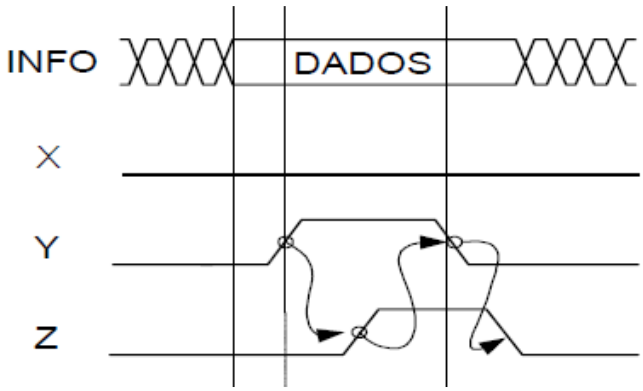
// resposta **d)**

30) Numa tansferência assíncrona:

a) Assume-se que o dispositivo externo responde à velocidade do CPU e, consequentemente, não existem sinais de protocolo envolvidos na transacção
b) O CPU prolonga o ciclo de leitura/escrita até que o dispositivo externo sinalize que a operação pretendida foi completada
c) O CPU prolonga o ciclo de leitura/escrita por um ou mais ciclos de relógio, em função de um sinal de protocolo gerado pelo dispositivo externo
d) Nenhuma das anteriores

Nos protocolos síncronos a velocidade depende da unidade mais lenta e é sempre constante, assumindo sem verificar que a informação chega ao receptor. A hipótese a) não é pois é um protocolo síncrono. Nos protocolos semi-sincronos utiliza-se um funcionamento síncrono, mas o receptor pode activar o sinal "wait" e realizar-se mais ciclos de relógio enquanto o "wait" estiver activo, tornando-se assincrono. A hipótese c) não é pois é um protocolo semi-síncrono. Nos protocolos assíncronos a fonte tem um sinal a dizer se a informação é válida, e se for válida, é enviada para o receptor enquanto este a quiser. A hipótese b) é verdadeira.

// resposta **b)**



(relativo às questões 31,32 e 33)

31) A figura representa um ciclo:

a) de leitura
b) de escrita
c) de arbitragem
d) nenhuma das opções anteriores está correcta

Como os dados aparecem primeiro, é um ciclo de escrita.

// resposta **b)**

32) O ciclo representado na figura tem handshake do tipo:

a) Interlocked
b) Completamente interlocked (fully interlocked)
c) Síncrono
d) Merged

É assíncrono.

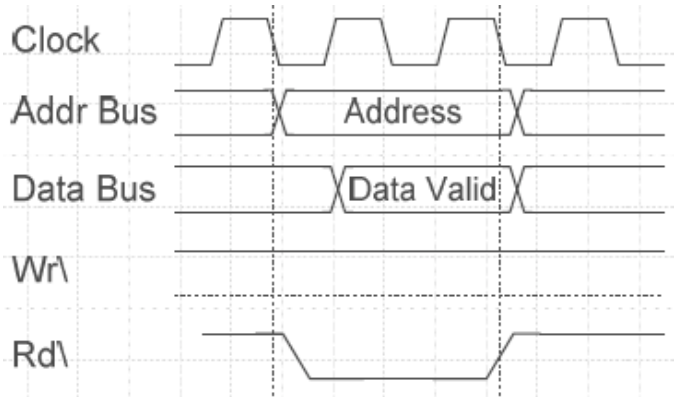
// resposta **b)**

33) Os sinais representados na figura pelos rótulos {X, Y , Z} denominam-se, respectivamente:

a) Read; Write ; Strobe
b) Write; Read; Valid
c) Read; Write ; Acknowledge
d) Write; Read; Acknowledge

Como é um ciclo de escrita, Read = 0 (sempre), logo é o X. Num ciclo de escrita, é necessária dar a ordem que se pode escrever, logo Y é Write. Depois há a autorização que se pode escrever, logo Y é Acknowledge.

// resposta **c)**



(relativo à questão 34)

34) Considere-se um CPU que funciona a uma frequência de 500 MHz e o ciclo de leitura corresponde ao da figura. Pretende-se ligar a este CPU uma memória RAM com um tempo de acesso de 15 ns. Sendo uma transferência de tipo semi-síncrono, qual o número de wait-states que é necessário introduzir para que a operação decorra com sucesso?

a) 2
b) 4
c) 6
d) 8

Como f = 500 MHz, então T = 1/f := 2ns

O tempo de acesso é 15 ns. Como 15 / 2 = 7.5, para que a operação ocorra com sucesso, são necessários 8 ciclos de relógio. Na figura pode ver-se que o ciclo de leitura dura 2 ciclos de relógio, então serão precisos mais 6 ciclos para que se obtenha os 8 ciclos pretendidos.

// resposta **c)**

35) Diz-se que um barramento é do tipo micro-ciclo quando:

a) a fase de endereçamento engloba a transferência de dados
b) a fase de transferência de dados engloba o endereçamento
c) o fim do ciclo de endereçamento depende do fim prévio do ciclo de dados
d) as fases de endereçamento e transferência de dados são tratadas como operações autónomas

Quando a fase de endereçamento engloba a transferência de dados é Merged, logo a a) e c) não são. A b) é irreal.

// resposta **d)**

36) Quando o endereçamento é do tipo micro-ciclo pode afirmar-se que as transferências do tipo bloco (block transfer) :

a) Permitem um maior throughput pois com uma única fase de endereçamento conseguem-se realizar várias transferências de dados
b) Implicam uma maior complexidade na implementação dos slaves
c) Podem ter limitações em relação ao tamanho máximo do bloco que pode ser transferido numa única operação
d) Todas as opções anteriores estão correctas

37) Admita que possui um sistema com um barramento multiplexado para dados e endereços, com largura de 8bits. Admita que é usada uma estratégia baseada em qualifiers codificados. Qual o número mínimo de qualifiers que são necessários no caso de o endereçamento requerer 24 bits e o comprimento dos dados ser 8 bits?

a) 1
b) 2
c) 3
d) 4

Tem-se que 24/8=3 e 8/8=1, portanto tem-se 4 “pacotes” de 8 bits, sendo 8 bits a largura do barramento de dados e endereços. $4 = 2^2$.

// resposta **b)**

38) Pretende-se fazer a transferência de informação (leitura/escrita) tipo microciclo num barramento de 24 bits no qual se quer trabalhar com 24 bits de endereços e 16 bits de dados. Considerando que se utiliza um strobe específico para os endereços e outro para os dados, quantos qualifiers são necessários?

a) 1
b) 2
c) 3
d) Nenhuma das anteriores

Os strobes já dizem se são endereços ou dados. Como o barramento é de 24 bits, consegue suportar, sem a necessidade de recorrer a qualifiers, os 24 bits de endereços, e os 16 bits de dados. Não são precisos qualifiers.

// resposta **d)**

39) Considere um barramento paralelo multiplexado, constituído por 24 linhas de informação. Sobre este barramento pretende implementar-se um protocolo de comunicação que apresenta um espaço de endereçamento de 24 bits e 16 bits de dados. A multiplexagem do barramento de informação requer no mínimo:

a) três linhas de strobe independentes
b) uma linha de strobe e uma linha do tipo infotype
c) uma linha de strobe e duas linhas do tipo qualifier
d) uma linha do tipo Read Pulse e uma linha do tipo Write Pulse

40) Na transferência de informação entre memória e I/O, define-se latência como:

a) O tempo que decorre desde o pedido de informação até à disponibilização do último byte.
b) O tempo que decorre desde o pedido de informação até à disponibilização do primeiro byte.
c) O nº máximo de bytes/seg transferidos após o início da transferência.
d) Nenhuma das anteriores está correcta.

41) Num sistema computacional onde se usa transferência de informação sob a forma de E/S programada:

a) A informação é trocada directamente entre a memória e o periférico.
b) O CPU inicia a transferência após o periférico ter sinalizado que está pronto para trocar informação.
c) O CPU inicia e controla a transferência de informação.
d) Nenhuma das anteriores está correcta.

42) No modelo de entrada/saída programada:

a) O CPU tem que esperar que o periférico esteja disponível para a troca de informação.
b) Quando o periférico está pronto para disponibilizar/receber informação sinaliza o CPU.
c) A transferência de informação faz-se por DMA.
d) Nenhuma das anteriores está correcta.

43) Quando é usada a técnica de entrada/saída de dados por interrupção:

a) O periferico faz um pedido de interrupção ao CPU quando estiver pronto para transferir os dados
b) O periferico faz um pedido de interrupção ao CPU após a conclusão de transferência de dados
c) O CPU verifica através de um ciclo de polling se o periférico está pronto para transferir os dados
d) Nenhuma das anteriores

Na técnica Programmed I/O o CPU toma a iniciativa e controla a transferência, ficando à espera num ciclo de polling até o periférico estar disponivel. Este é o caso da hipótese c), logo não é a técnica de interrupção. Na técnica de interrupção, o periférico toma a iniciativa indicando que está pronto para a transferência e depois o CPU controla a mesma. É o caso da hipótese a). Na hipótese b) não faz sentido sequer primeiro transferir os dados e só depois interromper o CPU.

// resposta **a)**

44) Na família x86, uma interrupção dá origem a um ciclo de acknowledge durante o qual:

a) É transferido no data bus o endereço da ISR.
b) É transferido no data bus o endereço de uma posição de uma tabela na qual deve estar o endereço da ISR.
c) É apenas indicado que a interrupção foi aceite já que a ISR correspondente tem de estar numa posição de memória pré-definida para cada interrupção.
d) Nenhuma das anteriores está correcta.

45) Quando uma série de periféricos estão organizados numa cadeia daisy chain:

a) É necessária uma linha de interrupção específica para cada periférico.
b) É necessário fazer polling sobre os periféricos para detectar quem fez a interrupção.
c) O periférico com prioridade mais elevada consegue fornecer ao CPU o vector de interrupção.
d) Nenhuma das anteriores está correcta.

Em daisy chain, o CPU recebe a informação que houve uma interrupção e vais verificar sequencialmente qual o periférico responsável. Os vectores serão fornecidos ao CPU mediante a posição dos periféricos. O periférico mais próximo fornece primeiro, o segundo em segundo lugar, e assim sucessivamente. A posição dos periféricos define uma prioridade.

// resposta **c)**

46) Num sistema com interrupções vectorizadas:

a) Os periféricos podem estar agrupados numa cadeia daisy-chain
b) A identificação da fonte é realizada por hardware
c) A cada periférico é atribuido um vector único
d) todas as anteriores

47) Num sistema baseado num processador da arquitectura intel x86, a memória apresenta, a partir do endereco 0000:0004E, o seguinte conteúdo: 3B 45 12 89 7C 9F 8D 6B. O endereço da rotina de serviço à interrupção aí programado e o correspondente vector são:

a) Endereço = 3B45:1289, Vector =19
b) Endereço = 9F7C:8912, Vector = 20
c) Endereço = 1289:7C9F, Vector = 20
d) Endereço = 8D6B:7C9F, Vector = 21

Memória:

Endereço	Conteúdo	
0000:00054 →	6B	
	8D	
0000:00052 →	9F	Segment
	7C	
0000:00050 →	89	Offset
	12	
0000:0004E →	45	Segment
	3B	
0000:0004C →	?	Offset
	?	
(. . .)		
0000:00000 →	?	Offset
	?	

Vai sendo guardado na memória o Segment e o Offset da RSI. Ao todo, é ocupado um bloco de 4 endereços para cada conjunto de Segment:Offset. O primeiro Offset ficou em 0000:00000, o segundo em 0000:00004, e assim sucessivamento, em múltiplos de 4. Olhando para o conteúdo da memória, o endereço 0000:0004E não pode ter o endereço da RSI, pois só se tem o Segment, mas a seguir tem-se 0000:00050 com o Offset e o Segment correspondente em 0000:00052. Portanto a RSI tem o endereço 9F7C:8912. Como a RSI apareceu no endereço 0000:00050, e como se sabe que estes endereços da RSI são guardados no endereçamente de memória em múltiplos de 4, tem-se que o vector é 20, pois foram guardados 20 endereços de RSI's.
(0x50 = 80 em decimal, então 80/4 = 20)

// resposta **b)**

48) Num sistema de interrupções vectorizadas, a sequência de operações efectuadas pelo CPU na fase de atendimento a uma interrupção é, pela ordem indicada, a seguinte:

a) Identificação da fonte, determinação do endereço da RSI, salvaguarda do endereço de retorno, salto para RSI
b) Salto para a RSI, identificação da fonte, determinação do endereco da RSI, salvaguarda do endereço de retorno
c) Determinação do endereco da RSI, identificação da fonte, salvaguarda do endereço de retorno, salto para a RSI
d) Nenhuma das anteriores

49) Na arquitectura Intel x86, no atendimento a uma interrupção (em modo real), o CPU efectua, entre outras, as seguintes operações:

a) Salvaguarda na stack o registo flags e o endereço /(segmento e offset) da rotina de serviço à interrupção e desactiva as interrupções
b) Salvaguarda na stack os registos flags, CS, IP e o endereço (segmento e offset) da rotina de serviço a interrupção e desactiva as interrupções
c) Salvaguarda na stack os registos flags, CS e IP, e desactiva as interrupções
d) Salvaguarda na stack o registo flags e desactiva as interrupções; os restantes registos não são automaticamente salvaguardados, sendo da responsabilidade do programador fazê-lo

50) Um watchdog é um circuito que permite:

a) Fazer reset ao CPU após uma certa temporização.
b) Receber linhas de interrupção de periféricos e controlar a sua ligação ao CPU.
c) Que um controlador de DMA fique indefinidamente a controlar o barramento após ele lhe ser atribuído.
d) É evidente que é um animal doméstico.

Um watchdog timer tem como função monitorizar a operação do microprocessador e, em certas condições, gerar um sinal de Reset.

// resposta **a)**

51) Considere um timer de 8 bits que funciona, em modo, alternado, com duas constantes de divisão KA e KB. Utilizando o timer como divisor de frequência e supondo que a frequência de entrada é 2 MHz, a mínima frequência de saída que é possível obter é, aproximadamente:

a) 2 KHz
b) 4 KHz
c) 8 KHz
d) 125 KHz

$$f_{out} = \frac{f_{in}}{k} \Rightarrow f_{out} = \frac{2 * 2^{20}}{2^8} = \frac{2^{21}}{2^8} = 2^{13} = 2^3 * 2^{10} = 8\ KHz$$

// resposta **c)**

52) Considere um watchdog timer, com uma frequência de entrada de 1 MHz, construido a partir de um contador de 8 bits que, sempre que a contagem atinge o valor máximo, força o reset do processador que esta a monitorizar. O programa a correr nesse processador tem que, periodicamente, colocar o valor do contador a 0. De modo a impedir o reset do processador, o período de actuação no watchdog timer não pode ser superior a, aproximadamente:

a) 0,25 ms
b) 0,5 ms
c) 1,0 ms
d) 2 ms

$$f_{out} = \frac{f_{in}}{k} \Rightarrow f_{out} = \frac{1 * 2^{20}}{2^8} = 1 * 2^{12} = 1 * 2^2 * 2^{10} \approx 4 * 10^3 Hz$$

$$T = \frac{1}{f} \Rightarrow T = \frac{1}{4 * 10^3} = 0.25 * 10^{-3}\ s$$

// resposta **a)**

53) Considere um watchdog timer, com uma frequência de entrada de 100KHz, construído a partir de um contador de 8 bits que, sempre que a contagem atinge o valor máximo, força o reset do processador cuja operação está monitorizar. O programa a correr nesse processador, tem que periodicamente colocar o contador a 0. De modo a impedir o reset do processador, o periodo de actuação do watchdog deverá ser superior a, aproximadamente:

a) 80 μs
b) 39 ns
c) 2,5 ms
d) 1,2 μs

$$f_{out} = \frac{f_{in}}{k} \Rightarrow f_{out} = \frac{100 * 2^{10}}{2^8} = 100 * 2^2 = 4 * 100 = 0.4 * 10^3 Hz$$

$$T = \frac{1}{f} \Rightarrow T = \frac{1}{0.4 * 10^3} = 2.5 * 10^{-3}\ s$$

// resposta **c)**

54) O trecho de código assembly Intel x86 seguinte envia 2000 caracteres para um periférico

```
send:   mov  bx, 0x2800
        mov  dx, 0x1000
        mov  cx, 2000
s1:     in  al, dx
        and  al, 0x06
        jz  s1
        mov  al, [bx]
        out  dx, al
        inc  bx
        dec  cx
        jnz  s1
        ret
```

Admitindo que este código e executado num processador de 10 MIPS (executa 10 x 10^6 instruções/seg) e que o ciclo de polling é efectuado em média 5 vezes, a taxa de transferência média que se obtém é, aproximadamente:

a) 500 KB/s
b) 1 MB/s
c) 1,25 MB/s
d) 10 MB/s

O ciclo de repetição {s1: ... jnz s1} vai repetir-se muitas vezes, já que é necessário enviar 2000 caracteres. Por isso, para efeito de cálculo, pode desprezar-se as 3 primeiras instruções, que só acontecem uma vez, ao contrário das restantes.

s1: in al, dx and al, 0x06 jz s1	Este é o ciclo de polling. Tem 3 instruções, como se repete em média 5 vezes, dá um total de 3*5 = 15 instruções
mov al, [bx] out dx, al inc bx dec cx jnz s1	As restantes instruções são 5

Total de instruções = 15 + 5 = 20 instruções

$$Taxa\ de\ trânferência = \frac{nr.instruções/segundo}{nr.instruções} = \frac{10\ MIPS}{20} = \frac{10 * 10^6}{20} = 500 * 10^3$$

// resposta **a)**

55) O trecho de código Assembly x86 seguinte envia 20000 caracteres para um periférico.

```
send:   mov bx, 0x2800
        mov  dx, 0x1000
        mov  cx, 20000
s1:     in  al, dx
        and  al, 0x06
        cmp  al, 0x06
        je  s1
        mov  al, [bx]
        out  dx, al
        inc  bx
        dec  cx
        jnz  s1
        ret
```

Admitindo que este código é executado num processador de 20 MIPS (executa $2*10^7$) ciclo de polling é efectuado em média 5 vezes, a taxa de transferência no processador é de aproximadamente:

a) 1,6 KB/s
b) 20 MB/s
c) 4 MB/s
d) 800 KB/s

O ciclo de repetição {s1: ... jnz s1} vai repetir-se muitas vezes, já que é necessário enviar 20000 caracteres. Por isso, para efeito de cálculo, pode desprezar-se as 3 primeiras instruções, que só acontecem uma vez, ao contrário das restantes.

s1: in al, dx and al, 0x06 cmp al, 0x06 je s1	Este é o ciclo de polling. Tem 4 instruções, como se repete em média 5 vezes, dá um total de 4*5 = 20 instruções
mov al, [bx] out dx, al inc bx dec cx jnz s1	As restantes instruções são 5

Total de instruções = 20 + 5 = 25 instruções

$$Taxa\ de\ trânferência = \frac{nr.instruções/segundo}{nr.instruções} = \frac{20\ MIPS}{25} = \frac{20 * 10^6}{25} = 800 * 10^3$$

// resposta **d)**

56) Numa transferência por DMA, quando o controlador de DMA pretende dar início à transferência:

a) Gera uma interrupção ao CPU que é interpretada como um pedido para cedência dos barramentos; a transferência tem início quando o DMA receber a confirmação, através do sinal busgrant, de que os barramentos foram libertados
b) Informa o CPU, através da linha busreq, que vai dar inicio à transferência e inicia-a de imediato. Quando o CPU necessitar de aceder à memoria activa o sinal busgrant e o DMA suspende, temporariamente, a transferência
c) Gera uma interrupção ao CPU sinalizando-o, dessa forma, que vai dar início a transferência
d) Requisita ao CPU o controlo dos barramentos, através do sinal busreq, iniciando a transferência logo que se torne no bus master

57) Numa transferência de DMA em modo cycle-stealing:

a) É necessário que a memória esteja pronta para iniciar a transferência.
b) O controlador de DMA só pode controlar o barramento durante um número pré-determinado de ciclos.
c) O controlador de DMA e a memória acordam num número de bytes e transferem-nos todos seguidos.
d) Nenhuma das anteriores está correcta.

58) Para a transferência de 1024 words (de 16 bits) um controlador de DMA de 8 bits, não dedicado, a funcionar em modo bloco, demora:

a) 8192 bus cycles
b) 4096 bus cycles
c) 2048 bus cycles
d) 1024 bus cycles

59) A fetch e o deposit duram 2 ciclos cada. Considerando uma frequência de 5MHz, 1024 palavras para transferir de 32 bits cada e um barramento de dados de 8 bits, a duração da transferência se o controlador de DMA for dedicado funcionando em bloco e a duração se for em modo cycle-stealing, é respectivamente:

a) 3,3 ms e 4,9 ms
b) 4,9 ms e 3,3 ms
c) 1,6 ms e 4,9 ms
d) 1,6 ms e 3,3 ms

DMC_C dedicado	Faz o fetch ou o deposit
DMC_C não dedicado	Faz o fecth e o deposit
DMC_C em modo Cycle-Stealing	Faz o fecth, espera 1 ciclo, faz o deposit e espera mais um ciclo

T=1/f => T=200 ns
bytes para transferência = 1024*32/8 = 4096 bytes

- DMC_ dedicado => 2T
  Para 4096 bytes tem-se que t = $2*200*10^{-9}*4096 \approx 1,6$ ms
- Cycle-Stealing => 2T + T + 2T + T = 6T
  Para 4096 bytes tem-se que t = $6*200*10^{-9}*4096 \approx 4,9$ ms

// resposta **c)**

60) Em barramentos multi-master um dos módulos que compõem o circuito de arbitragem denomina-se "sincronização de saída" e destina-se a:

a) Garantir que as saídas dos masters se mantêm estáveis durante o processo de arbitragem
b) Seleccionar qual dos masters irá utilizar o barramento em seguida
c) Garantir que as linhas do barramento não ficam flutuantes
d) Nenhuma das opções anteriores está correcta

61) Em barramentos multi-master os circuitos de arbitragem do tipo ripple apresentam, face aos circuitos de arbitragem tipo look-ahead, as seguintes vantagens:

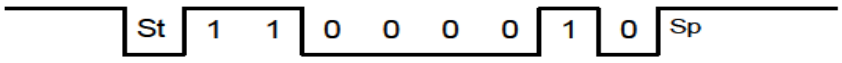
a) Maior simplicidade e rapidez
b) Maior simplicidade e homogeneidade
c) Maior rapidez e homogeneidade
d) Pelo menos um dos tipos de circuito acima referidos não é utilizado para realizar circuitos de arbitragem

62) Há diversas técnicas para realizar a sincronização de relógio em barramentos série, entre elas:

a) Relógio codificado
b) Relógio explícito auto-sincronizado
c) Relógio explícito do transmissor
d) Todas as opções anteriores são verdadeiras

63) Em barramentos série a transmissão de dados é organizada em tramas, as quais contêm campos com diferentes funções. Tipicamente as tramas contêm os seguintes campos:

a) Sincronização, Identificação, Arbitragem, Detecção de erros e Dados
b) Sincronização, Arbitragem, Detecção de erros e Dados
c) Identificação, Arbitragem, Detecção de erros e Dados
d) Todas as estruturas acima definidas são possíveis



(relativo à questão 63)

64) A figura representa uma trama RS-232 em que:

a) é transmitido o caracter 43H, sendo usada uma configuração de 7 bits de dados e paridade par
b) é transmitido o caracter 43H, sendo usada uma configuração de 7 bits de dados e paridade impar
c) é transmitido o caracter C2H, sendo usada uma configuração de 8 bits de dados sem paridade
d) é transmitido o caracter 61H, sendo usada uma configuração de 7 bits de dados com paridade ímpar

65) O standard RS232 prevê a possibilidade de realizar-se handshaking por hardware entre dois dispositivos do tipo DTE com base nos sinais "RTS" e "CTS". Quando o receptor deseja interromper o envio de dados:

a) Desactiva a sua saída "RTS", a qual se encontra ligada à entrada "CTS" do transmissor
b) Desactiva a sua saída "CTS", a qual se encontra ligada à entrada "RTS" do transmissor
c) Desactiva a sua saída "RTS", a qual se encontra ligada à entrada "RTS" do transmissor
d) Desactiva a sua saída "CTS", a qual se encontra ligada à entrada "CTS" do transmissor

66) Numa interface RS232 a utilização de XON / XOFF corresponde a:
a) Handshaking hardware
b) Handshaking através das linhas CTS / RTS
c) Estabelecer o início da ligação
d) Nenhuma das anteriores

67) O protocolo série RS232 apresenta uma técnica de sincronização de relógio denominada "relógio implícito", segundo a qual:
a) o relógio do receptor é sincronizado com o do transmissor por meio da transmissão de um stop bit em cada trama.
b) o relógio é enviado, em forma codificada, conjuntamente com os dados
c) o relógio é enviado numa linha dedicada
d) nenhuma das opções anteriores está correcta

68) Numa interface RS232 transferiram-se 2.048 bytes de informação em 178 mseg. O baudrate seria provavelmente:
a) 115.200
b) 9.600
c) 57.600
d) Nenhuma das anteriores

69) Em termos de arquitecturas de interligação, o protocolo SPI permite:
a) A existência de vários slaves, sendo a selecção efectuada por meio de Chip Selects independentes
b) A existência de vários slaves, sendo a selecção efectuada por meio de um campo de identificação embebido na trama
c) Ligar apenas dois dispositivos, um master e um slave
d) Todas as opções anteriores estão erradas

70) No que é vantajoso o protocolo SPI?
a) Tem uma comunicação full duplex
b) É fácil de implementar, por hardware ou por software
c) Não precisa de transceivers
d) Todas as anteriores

71) No protocolo SPI o relógio é:
a) Explícito, gerado pelo Master.
b) Explícito, gerado pelo Slave
c) Implícito
d) Nenhuma das anteriores

72) Em termos de arquitecturas de interligação, o protocolo I2C permite:
a) A existência de vários slaves, sendo a selecção efectuada por meio de Chip Selects independentes
b) A existência de vários slaves, sendo a selecção efectuada por meio de um campo de identificação embebido na trama
c) Ligar apenas dois dispositivos, um master e um slave
d) Todas as opções anteriores estão erradas

73) O barramento I2C é multimaster, decorrendo o processo de arbitragem:
a) Através de um par de linhas Req*/Gnt* dedicadas, em paralelo com a transacção corrente (hidden bus arbitration)
b) Através da técnica de bit dominante/bit recessivo na linha SCL, perdendo um master a arbitragem quando lê um bit dominante tendo escrito um bit recessivo
c) Através da técnica de bit dominante/bit recessivo na linha SDA, perdendo um master a arbitragem quando lê um bit recessivo tendo escrito um bit dominante
d) Através da técnica de bit dominante/bit recessivo na linha SDA, perdendo um master a arbitragem quando lê um bit dominante tendo escrito um bit recessivo

74) No protocolo I2C a arbitragem entre Masters é feita:
a) Por um árbitro de barramento residente num dos Masters.
b) Pela largura dos relógios de cada Master.
c) Pelo endereço da trama quando vários Masters iniciam uma transmissão em simultâneo.
d) Nenhuma das anteriores

75) A ligação de periféricos em USB é feita com uma topologia de.
a) Mesh
b) Tiered Star
c) Barramento
d) Qualquer das anteriores

76) Para interligar dispositivos USB utiliza-se um cabo com as seguintes características:
a) 4 condutores, 1 para transmissão de dados (TxD), 1 para recepção de dados (RxD) e dois para fornecimento de energia (VBUS e GND)
b) 4 condutores, sendo um par dedicado à transmissão de dados (Tx+ e Tx-) e o outro par à recepção de dados (Rx+ e Rx-), ambos em modo diferencial
c) 4 condutores, 2 para transmissão de dados em modo diferencial (D+ e D-) e dois para fornecimento de energia (VBUS e GND)
d) Nenhuma das opções anteriores está correcta

77) A especificação USB suporta diversos tipos de transferências, entre elas:
a) Controlo
b) Isócronas
c) Bulk
d) Todas as anteriores

78) A transferência de dados periódicos em tempo real (largura de banda e latência garantidas) designa-se:
a) Control Transfers
b) Isochronous Transfers
c) Interrupt Transfers
d) Bulk Transfers

79) A técnica de bit stuffing é utilizada:
a) Associada à codificação Manchester
b) Para garantir que se consegue identificar o início da trama
c) Para garantir que a linha não se mantém por demasiado tempo no mesmo nível
d) Nenhuma das anteriores

80) Para transferir dados directamente entre dois periféricos USB:
a) É preciso ligar ambos os periféricos ao mesmo hub USB.
b) Um dos periféricos tem de estar ligado directamente ao Host.
c) É necessário que ambos obedeçam à versão 2.x da norma.
d) Nenhuma das anteriores

81) Um device driver é:
a) Um tipo de periférico para armazenar dados (ex. disquete, pen,...)
b) Um programa que permite a outro programa interagir com um dado dispositivo de hardware
c) Um dispositivo físico que permite melhorar o desempenho do computador
d) Nenhuma das anteriores

82) No kit DET188, a efectiva transferência de dados entre a USART e os buffers de transmissão/recepção ocorre por:
a) DMA
b) Programação
c) Interrupção
d) Qualquer uma das anteriores

83) A USART gera interrupções:
a) Só quando recebe uma frame (RxRdy)
b) Só quando o buffer de transmissão está disponível (TxRdy)
c) Quando recebe uma frame (RxRdy) ou quando o buffer de transmissão está disponível (TxRdy)
d) Nenhuma das anteriores

84) Ligaram-se memórias de 16Kx4 a um processador com um barramento de endereços com 18 bits e barramento de dados de 8 bits. Sabendo que se está a utilizar apenas metade do espaço de memória disponível, utiliza-se, para seleccionar as memórias, um descodificador com o seguinte número de saídas:
a) 16
b) 8
c) 6
d) Nenhuma das anteriores está correcta

Colocando duas memórias 16Kx4 em paralelo, tem-se uma memória 16Kx8. Obteve-se 8 bits para o barramento de dados, sem usar nenhum descodificador.

$16Kx8 = 2^{14}x8$

O barramento de endereços tem 18 bits, como só se utiliza metade da memória, pode considerar-se apenas 17.

17-14 = 3. Portanto faltam 3 bits que serão colocados à entrada de um descodificador. O número de saídas desse descodificador será $2^3 = 8$.

// resposta **b)**

85) Uma memória estática (SRAM) tem tempos de acesso da ordem de:
a) milisegundos
b) microsegundos
c) nanosegundos
d) Nenhuma das anteriores está correcta

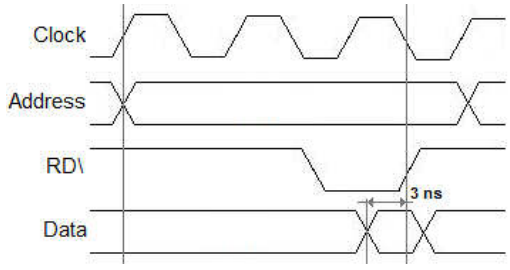
86) Uma memoria SRAM (RAM estatica) possui relativamente a uma DRAM (RAM dinâmica) a vantagem de:
a) Ser mais rápida
b) Possuir um menor custo por bit
c) Permitir densidades mais elevadas
d) Nenhuma das anteriores

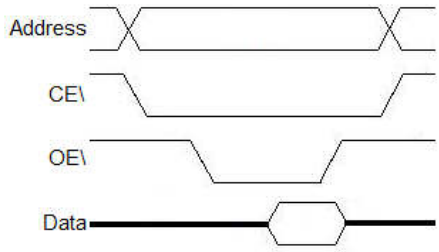
87) Numa memória estática (SRAM) de 64Mx8 o número de transistores que constitui a área de armazenamento é aproximadamente:
a) 537x10^6
b) 67 x10^6
c) 403 x10^6
d) 3220 x10^6

$64Mx8 = 64 \times 2^{20} \times 8 = 2^6 \times 2^{20} \times 2^3 = 2^{29} = 2^{30}/2 \approx 10^9/2 = 500 \times 10^6$
(é uma boa aproximação, quando não se pode usar calculadora)

Como na SRAM há 6 transistores/célula, tem-se que:
Nr transistores = $(500 \times 10^6) \times 6 = 3000 \times 10^6$ transistores

// resposta **d)**





(relativo à questão 88)

88) Considere um sistema constituido por um microprocessador com um barramento de dados de 8 bits a funcionar a 10 MHz e ligado a uma memoria SRAM de 128x8. A figura apresenta os diagramas temporais relativos a um ciclo de leitura da memória pelo microprocessador. Suponha que o atraso introduzido pelo circuito de descodificação da memória e de 10 ns, que no barramento de dados entre a memória e o CPU existe um buffer que introduz um atraso de propagação de 7 ns e que o tempo de setup do processador é de 3 ns. Para que o sistema funcione correctamente, o limite superior para o tempo de acesso da memória devera ser:
a) 80 ns
b) 233 ns
c) 230 ns
d) 100 ns

No diagrama da esquerda observa-se a duração de 2,5T. Como f=10MHz e T=1/f, então T=100ns. Assim sendo, 2,5T = 250ns.

$T_{access} = 250 - 10 - 7 - 3 = 230ns$

// resposta **c)**

89) Numa memória dinâmica (DRAM):
a) As células necessitam de refrescamento regular
b) O barramento de endereços é multiplexado no tempo
c) O tempo de acesso é independente da posição de memória acedida
d) Todas as anteriores

90) Numa memória dinâmica (DRAM):
a) É necessário aceder periodicamente à memória numa operação de escrita para que a informação não se perca.
b) A informação mantém-se incondicionalmente desde que a memória esteja com alimentação.
c) O controlador de linha é o responsável pela gestão do sentido do barramento de dados.
d) Nenhuma das anteriores está correcta.

Faz-se uma operação de leitura para haver refresh. Na operação escrita podia estragar-se os dados que lá estão. Na DRAM as céculas têm condesadores, se apenas de mantiver a alimentação, então a tensão do condensador tende para zero, pois a tensão de um condensador depende da variação de corrente, e não da corrente propriamente. O controlador de linha controla a linha, não se a operação é de escrita ou leitura.

// resposta **d)**

91) Uma memória dinâmica com 7 linhas de endereço pode conter:
a) 64K posições
b) 32K posições
c) 16K posições
d) 8K posições

Matriz quadrada →linhas = colunas = 7
O adress tem 7+7 = 14 bits

2^{14} = 16K

// resposta **c)**

92) Numa memória dinâmica (DRAM) de 8Mx16 o número de transistores que constitui a área de armazenamento é, aproximadamente:
a) 805x10^6
b) 9x10^6
c) 50x10^6
d) 134x10^6

93) O número de bits dos barramentos de endereços e de dados de uma memória dinâmica (DRAM) de 8Mx8 é, respectivamente:
a) 11 e 16
b) 12 e 16
c) 16 e 8
d) 23 e 16

94) O número de bits dos barramentos de endereços e de dados de uma memória dinâmica de 16Mx32 é respectivamente:

a) 24 e 32
b) 32 e 24
c) 12 e 32
d) 16 e 32

DRAM => Capacidade = 2^{2m} x n, sendo m o número de bits do endereço e n o número de bits dos dados

16M x 32 = $16*2^{20}$ x 32 = 2^4*2^{20} x 32 = 2^{24} x 32 = 2^{2*12} x 32 => m=12; n=32

// resposta **c)**

95) Os parâmetros relativos a um ciclo de leitura de uma memória dinâmica (DRAM) de 16Mx8 são os seguintes: Access time from RAS=20 ns; Access time from CAS=10 ns; RAS width=40 ns; Cycle time=50 ns; Precharge time=10 ns. Utilizando este ciclo de leitura, a taxa de transferência máxima que é possível obter é:

a) 7 MB/s
b) 10MB/s
c) 20 MB/s
d) 40 MB/s

Cycle time = RAS width + Precharge time = 50 ns

$f_{max} = \frac{1}{T_{RC}} = \frac{1}{50*10^{-9}} = 20 * 10^6$ MB/s

// resposta **c)**

96) Na memória da questão anterior (16Mx8), os parâmetros relativos a um ciclo de refrescamento são os seguintes: RAS width=40 ns; Cyclo time=50 ns; Precharge time=10 ns. O tempo necessario para efectuar um refrescamento completo à memoria é, aproximadamente:

a) 50 ns
b) 204 µs
c) 408 µs
d) 838 ms

Cycle time = RAS width + Precharge time = 50 ns

16Mx8 = 2^{24}x8

Sendo a matriz quadrada, tem-se 2^{12} linhas e 2^{12} colunas. Todas as células de uma linham sofrem refresh simultaniamente no mesmo ciclo. Como são 2^{12} linhas, o ciclo vai-se repetir 2^{12} vezes. (2^{12} = 4k)

50n * 4k = 200 µs

// resposta **b)**

97) Pretende-se implementar um módulo de memória DRAM de 512Mx8 a partir de circuitos de memória de 32Mx1. O número de circuitos de memória de 32Mx1 que é necessario organizar é:

a) 8
b) 16
c) 24
d) 128

Aumentar o comprimento da palavra: 32Mx1 → 32Mx8 são necessárias 8 memórias em paralelo do tipo 32Mx1

Para aumentar o número de posições de memória: 32Mx1 → 512Mx1 ⇔ 2^5Mx1 → 2^9Mx1, é necessário um descodificador de 9-5=4 entradas, que implica 2^4=16 saídas, estando uma memória do tipo 32Mx1 por cada saída.

16*8 = 128

// resposta **d)**

98) Numa memória cache, caso os dados pretendidos não se encontrarem num bloco de nível primário ocorre:

a) Um hit
b) Um miss
c) O envio de um sinal para o CPU a dizer que a informação está inválida
d) Nenhuma das anteriores

O nível primário contém os blocos de memória mais recentemente utilizados. Os pedidos de informação são sempre dirigidos ao nível primário, sendo o nível secundário envolvido apenas quando a informação pretendida não está nesse nível. Se os dados pretendidos se encontram num bloco do nível primário então existe um hit, caso contrário ocorre um miss. Na ocorrência de um miss é efectuado o acesso ao nível secundário para obter os dados (transferindo o bloco que contém a informação pretendida).

// resposta **b)**

99) Uma memória cache com mapeamento associativo (fully associative) e N linhas permite que um dado bloco de memória externa seja colocado na cache:

a) Apenas numa linha predefinida
b) Em qualquer linha
c) Numa de N/2 linhas possiveis
d) Nenhuma das anteriores

100) Considere um processador com um espaço de endereçamento de 32 bits e uma memória cache de mapeamento directo com 512 linhas de 128 bytes cada uma. A dimensão, em bits, dos campos tag, group e byte é:

a) Tag: 7; Group: 16; Byte: 9
b) Tag: 9; Group: 7; Byte: 16
c) Tag: 16; Group: 9; Byte: 7
d) Nenhuma das anteriores

Número de linhas: $512 = 2^9$ => O Group tem 9 bits
Dimensão dos blocos: $128 = 2^7$ => Com 7 bits, pode-se indicar qual o Byte pretendido
Número de bits restantes: 32 – (9+7) = 16 => A tag tem 16 bits

// resposta **c)**

101) Considere um processador com um espaço de endereçamento de 32 bits e uma memória cache de mapeamento directo com 1024 blocos de 64 bytes cada um. A dimensão, em bits, dos campos ”tag”, ”group” e ”byte” é:

a) Tag: 10; Group: 16; Byte: 6
b) Tag: 16; Group: 6; Byte: 10
c) Tag: 16; Group: 10; Byte: 6
d) Nenhuma das anteriores

Número de blocos: $1024 = 2^{10}$ => O Group tem 10 bits
Dimensão dos blocos: $64 = 2^6$ => Com 6 bits, pode-se indicar qual o Byte pretendido
Número de bits restantes: 32 – (10+6) = 16 => A tag tem 16 bits

// resposta **c)**

102) Considere uma cache com associatividade de 2 de 256 bytes em que a dimensão de cada bloco é de 16 bytes. Numa cache com estas características o bloco que contém o endereço de memória 0x9C (156 em decimal) pode ser colocado na linha:

a) 1
b) 0
c) 6
d) 9

Dimensão dos blocos: $16 = 2^4$ => Com 4 bits, pode-se indicar qual o Byte pretendido

O cache tem 256 bytes, e cada bloco tem 16 bytes, então a cache tem 256/16 = 16 blocos. Como a associatividade é de 2, há 2 blocos por linha, logo há 16/2 = 8 linhas. Como $8 = 2^3$, então o Set *(= linha)* representa-se em 3 bits.

Tem-se que:

Tag (?? bits)	Set (3 bits)	Byte (4 bits)

0x9C => 0...01 001 1100

O Set = Group = Linha = Posição = $1_2 = 1_{10}$

// resposta **a)**

103) Considere uma cache com associatividade de 2 de 64 bytes em que a dimensão de cada bloco é de 4 bytes. Numa cache com estas características o bloco que contém o endereço de memória 0x8D (141 em decimal) pode ocupar a posição:

a) 6
b) 9
c) 0
d) 3

Dimensão dos blocos: $4 = 2^2$ => Com 2 bits, pode-se indicar qual o Byte pretendido

O cache tem 64 bytes, e cada bloco tem 4 bytes, então a cache tem 64/4 = 16 blocos. Como a associatividade é de 2, há 2 blocos por linha, logo há 16/2 = 8 linhas. Como $8 = 2^3$, então o Set *(= linha)* representa-se em 3 bits.

Tem-se que:

Tag (?? bits)	Set (3 bits)	Byte (2 bits)

0x8D => 0...0100 011 01

O Set = Group = Linha = Posição = $11_2 = 3_{10}$

// resposta **d)**

104) O que é uma Memória Virtual?

a) Um tipo de memória, onde os seus endereços são directamente apontados pelo CPU
b) Um protocolo que separa endereços e dados de memória física
c) Uma técnica que permite aumentar a dimensão aparente da memória física do sistema
d) Nenhuma das anteriores

Uma memória virtual é uma técnica que permite a utilização de armazenamento secundário (discos) para aumentar a dimensão aparente da memória física do sistema.

// resposta **c)**

105) Se uma página que o CPU pretenda aceder não está em memória:

a) É gerado “page fault”
b) O CPU gera uma excepção e a informação é transferida do disco para a memória
c) Elabora uma tarefa que pode durar milhões de ciclos de relógio
d) Todas as anteriores

Se a página que o CPU pretende aceder não está em memória, é gerado um "page fault“. Então o CPU gera uma excepção e o handler respectivo (parte do Sistema Operaativo) desencadeia a transferência de informação do disco para a memória (tarefa que pode demorar milhões de ciclos de relógio...). Quando a página tiver sido transferida para a memória o processo anterior é retomado.

// resposta **d)**

106) Um endereço virtual é representado em 32 bits. Sabendo que a dimensão de cada página é de 4 KB, qual o número de páginas que a memória virtual suporta?

a) 8K
b) 1M
c) 4M
d) 8M

Dimensão de cada página: 4KB = $2^2 * 2^{10} = 2^{12}$ => utilizam 12 bits do endereço virtual
Bits para indentificar a página: 32 – 12 = 20 => existem 2^{20} páginas

// resposta **b)**

107) Um endereço virtual é representado em 32 bits e um endereço físico é representado por 30 bits. Sabendo que a dimensão de cada página é de 4 KB, quantos dos 30 bits presentes no endereço físico são resultado de um "adress translation", no processo de conversão do endereço virtual para físico?

a) 16
b) 18
c) 20
d) 22

Dimensão de cada página: 4KB = $2^2 * 2^{10} = 2^{12}$ => utilizam 12 bits do endereço virtual
Bits para indentificar a página física: 30 – 12 = 18 *(que são resultado do "adress translation")*

// resposta **b)**