

## I

Para a resposta às 4 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

<u>Endereço</u>	<u>Mnemónica</u>
A15C:378A	MOV AL, [BX]
A15C:378C	PUSH AX
A15C:378D	CALL 0x5678
A15C:3790	MOV DX, 0x5A63
A15C:3793	OUT DX, AL

1) A próxima instrução a ser executada é:

- a) MOV AL, [BX]
- b) PUSH AX
- c) CALL 0x5678
- d) MOV DX, 0x5A63

**R:**

A próxima instrução a ser executada é a apontada por CS:IP.  
Logo,  
A15C:378A → MOV AL, [BX]

2) O endereço físico de memória referenciado pela instrução “MOV AL, [BX]”

- a) 0xF150A
- b) 0xA2AC0
- c) 0x1400A
- d) 0x40A0A

**R:**

DS = 0x12B0 (neste caso é o DS por ser end. físico de memória)

End. Linear = segmento \*16 + *offset*

seg. = 12B0;                      *offset* = 150A

end. = 12B00 + 150A = 1400A

3) O valor do registo SP após a execução da instrução “PUSH AX” é:

- a) ...
- b) ...
- c) ...
- d) ...

4) O conteúdo do topo da *stack* da instrução “CALL 0x5678” é:

- a) 0x3790
- b) ...
- c) ...
- d) ...

6) Num sistema com interrupções vectorizadas:

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

**R:**

- A identificação da fonte de interrupção é feita por h/w num processo genericamente designado por "Interrupt acknowledge cycle"! Logo a alínea a) é verdadeira.
- Periféricos organizados numa estrutura *daisy chain*! Logo a alínea b) é verdadeira.
- Cada periférico possui um identificador único, designado por vector! Logo a alínea c) é verdadeira.

7) Numa memória estática SRAM:

- a) As células precisam de refrescamento regularmente
- b) O tempo de acesso é independente da posição (...)
- c) O barramento de endereços é multiplexado no (..)
- d) Todas as anteriores

**R:**

Por exclusão de partes, a escolha é a alínea b) porque:

- A memória DRAM é que precisa de refrescamento, logo se a opção a) está mal, o que implica que a d) também já não pode ser.
- É também na DRAM que há multiplexagem, por isso só sobra a resposta b).

8) Numa memória dinâmica de 64Mx8 o número de transistores que constitui a área de armazenamento é aproximadamente:

- a)  $537 \times 10^6$
- b)  $67 \times 10^6$
- c)  $403 \times 10^6$
- d)  $3220 \times 10^6$

**R:**

No caso da DRAM  $\rightarrow$  (1 transistor + 1 condensador) / célula

$$64\text{M} \times 8 \Rightarrow (64 \times 2^{20}) \times 8 \Rightarrow (2^6 \times 2^{20}) \times 2^3 = 2^{29}$$

$$2^{29} \cong 537 \times 10^6$$

*Nota: Se fosse SRAM, e porque tem 6 transistores / célula, o resultado seria*

$$2^{29} \times 6 \cong 537 \times 10^6 \times 6 \cong 3220 \times 10^6$$

9) 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

**R:**

$$16\text{M} = 2^4 \times 2^{20} = 2^{24} \rightarrow 24 \text{ linhas de endereço}$$

32 bits por word

## II

10) Num espaço de endereçamento de 16 bits, um decodificador 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

R:

A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	
0	X	1	X	0	X	X	X	X	X	X	X	X	X	X	X	
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0x2000
0	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	0x27FF
0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0x3000
0	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	0x37FF
0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0x6000
0	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	0x67FF
0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0x7000
0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	0x77FF

11) Na arquitectura Intel x86, no atendimento a uma interrupção (em modo geral), o CPU efectua (...) as seguintes operações:

- a) Salvaguarda na *stack* o registo de *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 (...) interrupção e desactiva as interrupções
- c) Salvaguarda na *stack* o registo *flags* e desactiva as interrupções; os restantes (...) são automaticamente salvaguardados, e é da responsabilidade do programador (...)
- d) (...)

14) Considere um *watch dog 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 período de actuação do *watch dog* deverá ser superior a, aproximadamente:

- a) 80 μs
- b) 39 ns
- c) 2,5 ms
- d) 1,2 μs

**R:**

$$F_{out} = \frac{F_{in}}{K} \quad \text{e} \quad F_{in} = \frac{1}{T_{in}}$$

$$T_{in} = \frac{1}{F_{in}} = \frac{1}{100k} = 10\mu s$$

$$F_{out} = K \times T_{in} = 256 \times 10\mu s \cong 2,5ms$$

**15)** 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) Requisita ao CPU o controlo dos barramentos, através do sinal *busreq*, iniciando a transferência logo que se torne no *bus master*
- c) Informa o CPU, através da linha *busreq*, que vai dar início à transferência e inicia-a de imediato. Quando o CPU necessitar de aceder à memória activa o sinal *busgrant* e o DMA suspende temporariamente, a transferência.
- d) Gera uma interrupção ao CPU sinalizando-o, dessa forma, que vai dar inicio ao processo de transferência.

**R:**

O controlador de DMA torna-se um *bus master* activando o sinal *BUS\_REQUEST* e esperando pela activação do sinal *BUS\_GRANT* (CPU activa *BUS\_GRANT* quando está em condições de libertar os barramentos)

**17)** ... são os seguintes: *Access time from RAS* = 50 ns; *Access time from CAS* = 20ns; *RAS width* = 75 ns; *Cycle Time* = 100 ns; *Precharge Time* = 25ns. Utilizando um ciclo de leitura, a taxa de transferência máxima que é possível obter é:

- a) 10MB/s
- b) 20MB/s
- c) 40MB/s
- d) 13,3MB/s

**R:**

**18)** Na memória da questão anterior (64Mx8), os parâmetros relativos a um ciclo de refrescamento são os seguintes: *RAS width* = 75ns; *Cycle time* = 100ns; *Precharge time* = 25ns. O tempo necessário para efectuar um refrescamento completo à memória é, aproximadamente:

- a) 1,6  $\mu$ s
- b) 0,8 ms
- c) 6,4  $\mu$ s
- d) 100 ns

**R:**

**19)** 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

**R:**

**20)** 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<sub>10</sub>) pode ocupar a posição:

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

**R:**

A cache parcialmente associativa é semelhante à cache com mapeamento directo, mas a primeira permite que mais que um bloco de um mesmo grupo possa estar na cache.

Uma cache com associatividade de 2 permite que 2 blocos do mesmo grupo possam estar simultaneamente na cache.

Cache com mapeamento directo:

O endereço do bloco é obtido por:

- Endereço do bloco = endereço / tamanho do bloco

A posição da cache associada a um dado endereço é dada por:

- Pos = endereço do bloco % número de blocos da cache

O tamanho do bloco e o número de blocos da cache são potências de 2

$$\text{Nº de blocos da cache} = \frac{64\text{bytes}}{4\text{bytes}} = 16 \text{ blocos}$$

$$\text{End. bloco} = \frac{\text{endereço}}{\text{tamanho do bloco}} = \frac{0x8D}{4\text{bytes}} = 0x2$$

$$\text{Pos.} = \text{endereço do bloco} \% \text{nº de blocos da cache} = 0x2 \% 16 = 0$$

**21)** 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  $2 \cdot 10^7$ ) 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

**R:**

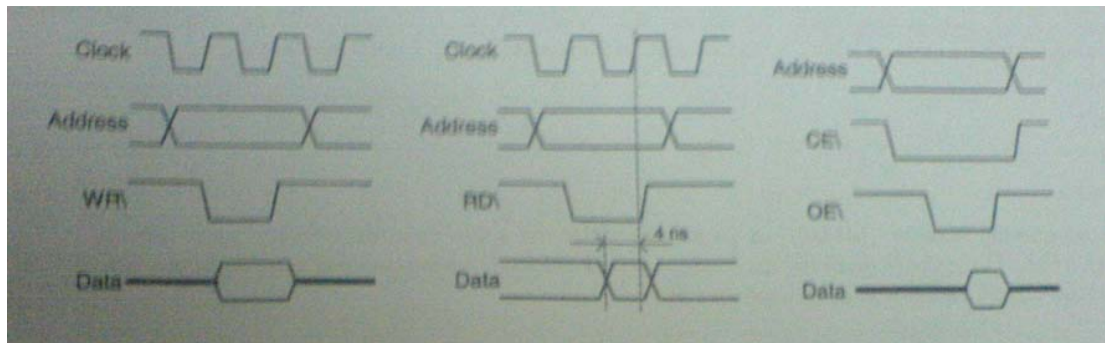
$$20 \text{ MIPS} = 2 \times 10^7$$

O código tem 25 instruções:

As 3 primeiras instruções não contam e como há *polling* às 4 inst. Seguintes, multiplica-se por 5 e soma-se as 5 instruções que faltam.

$$\text{Taxa de transferência} = \frac{20 \text{ MIPS}}{\text{n}^\circ \text{ de instruções}} = \frac{20 \times 10^6}{25} = 800 \text{ KB/s}$$





- 1) Se a frequência de relógio do CPU for 40 MHz, a transferência de informação da memória com o CPU só é possível introduzindo *wait states* no ciclo de leitura. Determine o número de *wait states* necessário para que o sistema funcione correctamente. Justifique os seus cálculos.

- 2) O sistema inclui um controlador de DMA não dedicado que funciona com uma frequência de relógio de 20 MHz e em que os ciclos de leitura e escrita são os apresentados anteriormente para o microprocessador. Determine a máxima taxa de transferência (bytes/seg) que é possível obter com este controlador, admitindo um funcionamento em modo bloco (note que o barramento de dados é de 32 bits). Justifique os seus cálculos.

- 3) Suponha que a memória SRAM de 256Kx32 é um módulo construído a partir de circuitos integrados de 64Kx32. Apresenta o diagrama lógico detalhado da organização desse módulo de memória, bem como o bloco lógico resultante e respectivos sinais de interface (utilize todos os sinais de interface necessários e **indique claramente a dimensão e composição de todos os barramentos que utilizar**)