

Parte Teórica
Correção

Cotações:

1– 2; 2a– 0,5, 2b– 0,5, 2c– 1, 2d– 1,5 ; 3– 2; 4a– 2 4b– 2; 5a– 0,5 5b– 2 5c– 3; 6a– 1, 6b– 1, 6c–1

1. Considere o interface básico com o sistema de computação de um módulo de entrada/saída. Que registos inclui e qual a respetiva função?

- **Data Register** – registo onde o processador coloca o dado a ser escrito no periférico (*output*) e onde o periférico coloca o dado a ser lido pelo processador (*input*)
- **Status Register** (*read-only*) – bits que indicam o estado do periférico
- **Control Register** (*write-only*) – registo onde o processador escreve os comandos/modo de operação para o periférico

2. O USB é atualmente um dos mais populares protocolos de comunicação, suportando diferentes taxas de transferência de dados.

a. Qual a taxa de transferência do USB *Fast Mode*?

480 Mbits/s

b. Que tipo de pacotes existem no USB?

Token Packets – o controlador USB endereça o dispositivo

Data Packets – contêm os dados

Handshake Packets – faz o *handshake*

c. Supondo que um pacote isócrono inclui no máximo 1kBytes, e que um dispositivo pode enviar apenas um desses pacotes por *micro frame* qual é a máxima taxa de transferência do dispositivo?

Microframe: uma cada 125µs

Máxima taxa de transferência = $2^{10}/(125 \times 10^{-6}) = 8 \text{ Mbist/s}$

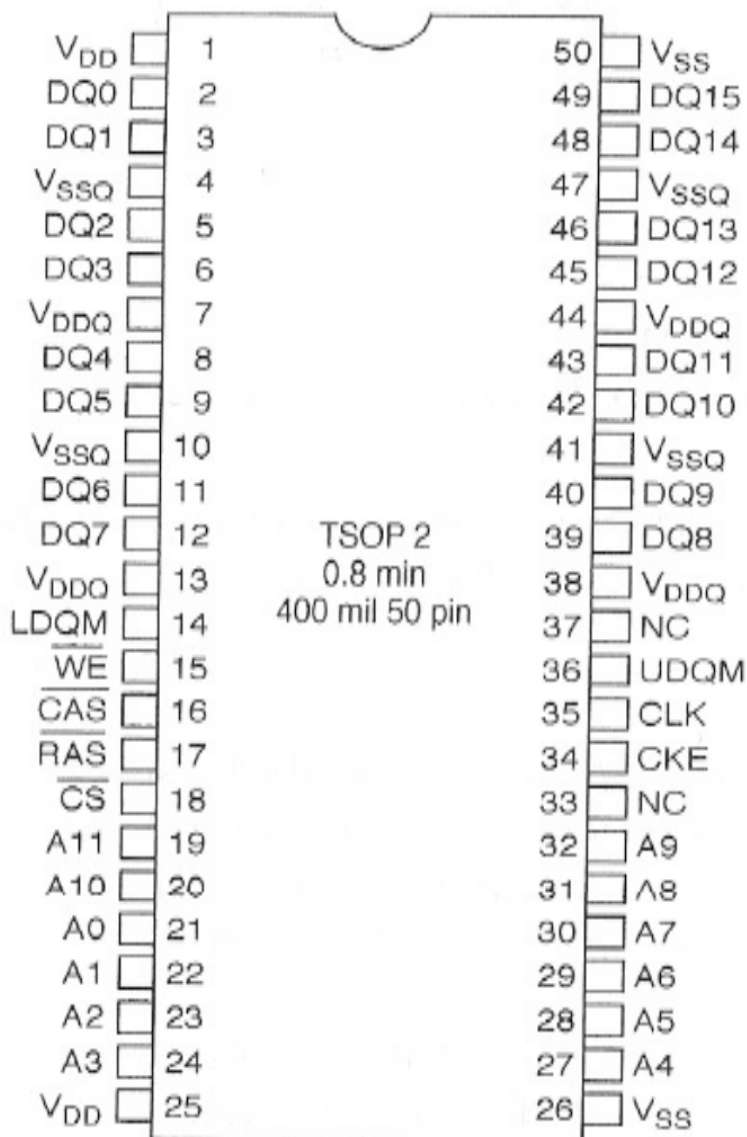
d. Qual o código usado na transmissão? Porque é usado *bit stuffing*?

NRZI (Non Return to Zero Inverted), em que a transmissão de um zero é representada por uma transição (de 1 para 0 ou de 0 para 1) e a transmissão de um 1 por ausência de transição (valor fixo, 1 ou 0).

Como o sinal de relógio não é transmitido, são as transições do valor do sinal na linha que são usadas para manter o sincronismo do relógio do recetor. Quando são transmitidos muitos 1 seguidos, é acrescentado (“*stuffed*”) um 0 para assegurar que o sincronismo do relógio do recetor se mantém.

3. A figura representa uma SDRAM.

Parte Teórica
Correção



Qual a organização da SDRAM (quantas palavras de quantos bits pode armazenar)?

$2^{24} \times 16$ – 16M palavras de 16 bits (16Mx16)

Capacidade total = $2^{24} \times 2^4 = 256\text{Mbits}$

4. Numa posição de um sistema de memória SECDED (Single Error Correction, Double Error Detection), byte-addressable, está armazenado o seguinte valor: **0110011000111**
- A informação armazenada está correta? Justifique a sua resposta
 - No caso de estar errada, é possível corrigir o erro? Justifique a sua resposta e indique o valor corrigido da informação armazenada no caso da resposta ser afirmativa.

M8 M7 M6 M5 C8 M4 M3 M2 C4 M1 C2 C1 P
0 1 1 0 0 1 1 0 0 0 1 1 1

$$C8_{\text{calc}} = M8 \text{ XOR } M7 \text{ XOR } M6 \text{ XOR } M5 = 0 \text{ XOR } 1 \text{ XOR } 1 \text{ XOR } 0 = 0$$

$$C4_{\text{calc}} = M8 \text{ XOR } M4 \text{ XOR } M3 \text{ XOR } M2 = 0 \text{ XOR } 1 \text{ XOR } 1 \text{ XOR } 0 = 0$$

$$C2_{\text{calc}} = M7 \text{ XOR } M6 \text{ XOR } M4 \text{ XOR } M3 \text{ XOR } M1 = 1 \text{ XOR } 1 \text{ XOR } 1 \text{ XOR } 1 \text{ XOR } 0 = 0$$

$$C1_{\text{calc}} = M7 \text{ XOR } M5 \text{ XOR } M4 \text{ XOR } M2 \text{ XOR } M1 = 1 \text{ XOR } 0 \text{ XOR } 1 \text{ XOR } 0 \text{ XOR } 0 = 0$$

$$P_{\text{calc}} = 0$$

Parte Teórica
Correção

$P_{\text{calc}} \text{ XOR } P_{\text{arm}} = 1$ **Erro num numero impar de bits**

$8_{\text{calc}} \text{ XOR } 8_{\text{arm}} = 0 \text{ XOR } 0 = 0$

$C4_{\text{calc}} \text{ XOR } C4_{\text{arm}} = 0 \text{ XOR } 0 = 0$

$C2_{\text{calc}} \text{ XOR } C2_{\text{arm}} = 1 \text{ XOR } 0 = 1$

$C1_{\text{calc}} \text{ XOR } C1_{\text{arm}} = 1 \text{ XOR } 0 = 1$ **Erro em M1 (o bit na posição 3)**

Dado corrigido: 01101101

5. Um sistema de 32-bits com a memória byte-addressable tem uma cache direct-mapped de 64kBytes com blocos (linhas) de 16 palavras de 32 bits.

- a. Quantas linhas tem a cache?

Linha: 16x4 Bytes = 64 Bytes

Nº linhas = $64 \times 2^{10} / 64 = 1024$

- b. Quais os campos em que estão divididos os endereços da memória, qual o número de bits de cada um deles e que informação contêm?

31	16 15	6 5	0
Tag	Index	Byte Offset	

Tag – os 16 bits mais significativos do endereço das posições de memória armazenadas na linha

Index – Nº da linha da cache

Byte Offset – posição do byte na linha; os bits 2 a 5 indicam a posição da palavra na linha

- c. Sendo $(\$s0) = 0x200CA004$, é executada a instrução **lw \$t0, 100(\$s0)**. Assuma que a linha onde se mapeia o endereço efetivo de memória gerado por esta instrução está vazia. Em que linha da cache e em que posição na linha é colocada a palavra acedida? Quais os endereços das outras palavras que são transferidas para essa linha da cache? Qual o conteúdo do campo **Tag** correspondente a essa linha?

$100 = 0x64$ $200CA004 + 00000064 = 200CA068$

31	16 15	6 5	0
0010 0000 0000 1100	10 10000001		101000	

Tag = 0x200C

Nº Linha = $1010000001_2 = 641_{10}$

Posição da palavra na linha: Word Offset = $1010_2 = 10$

Endereços das outras palavras na linha:

0x200CA040, 0x200CA044, 0x200CA048, 0x200CA04C, 0x200CA050, 0x200CA054, 0x200CA058, 0x200CA05C, 0x200CA060, 0x200CA064, 0x200CA068, 0x200CA06C, 0x200CA070, 0x200CA074, 0x200CA078, 0x200CA07C

6. Os SSD (*Solid State Disk*) usam memória *flash*.

- a. Que outros tipos de memória não volátil existem?

Memórias de semicondutores não voláteis: ROM, PROM, EPROM, EEPROM

- b. Que tipos de memória *flash* conhece? Qual desses tipos é usado nos SSD?

NOR Flash e NAND Flash. SSDs usam NAND Flash

Parte Teórica
Correção

- c. Quais as principais vantagens dos SSD relativamente aos discos magnéticos (HDD)? E quais as vantagens que os HDD ainda têm?

Principais vantagens dos SSD:

- **Mais rápidos (mais de uma ordem de grandeza mais rápidos)**
- **Robustez – menos suscetíveis a choques e vibrações**
- **Menor consumo de energia ($\approx 2W$ por drive)**
- **Mais pequenos e menos barulhentos**

Vantagem dos discos magnéticos:

- **Menor custo por bit**
- **Unidades com maior capacidade**