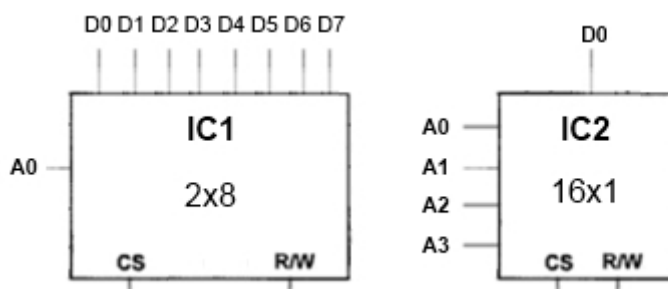


Parte Teórica

- 1** Considere uma máquina com um espaço de endereçamento de 16 bits, uma cache de 4KB, mapeamento directo e blocos de 1 Byte.
- a) Quantos bits são necessários para a *tag*? Justifique. **(2 Val)**
- b) Qual a capacidade total desta cache, contando com os bits da *tag* mais os *valid bits*? **(1 Val)**

- 2** Considere os dois circuitos integrados de memória RAM representados na figura, onde A_3, A_2, A_1, A_0 representam linhas de endereço, $D_7 \dots D_0$ representam linhas de dados, R/W representa a linha de leitura/escrita e CS a linha de *Chip Selection*.



Faça um esboço associando múltiplos circuitos integrados iguais de forma a obter uma memória RAM com 16 endereços com 8 bits cada. Deverá ser indicada explicitamente a linha de CS (*Chip Selection*) da memória resultante:

- a) Utilizando apenas circuitos de memória do tipo IC1. **(1 Val)**
- b) Utilizando apenas circuitos de memória do tipo IC2. **(1 Val)**
- 3** A gravação perpendicular e a gravação longitudinal utilizam formas distintas de representação da informação. Descreva resumidamente as diferenças substanciais entre estas duas formas de gravação magnética. **(2 Val)**

Equivalente ao teste de 2 valores (responder numa folha separada)

1. Converta o número $110011,111_2$ para decimal.
2. Converta o número -53_{10} para um número binário de 8 bits segundo a representação de Complementos de 2.
3. Represente o número decimal $-3,75$ no formato IEEE 754 de precisão simples. Apresente o resultado em Hexadecimal.
4. A janela abaixo representa a extração de parte do segmento de dados de um programa, no codeview. A sua declaração foi feita da forma apresentada abaixo:

```
dseg segment para 'data'
        num1 dword ?
        num2 real4 ?
        num3 fword ?
dseg ends
```

0E5C:0000	B8	5C	0D	8E	D8	B4	4C	CD	21	00	00	00	00	00	00	00
0E5C:0010	5C	06	00	00	48	C0	4E	42	4E	42	30	38	68	02	00	00
0E5C:0020	B2	00	D0	00	01	00	43	56	02	00	00	00	EF	0F	A0	B0
0E5C:0030	00	AA	00	43	01	0D	08	EA	B2	00	00	00	00	00	00	00
0E5C:0040	00	00	AA	00	01	00	43	56	02	00	00	00	42	30	38	00
0E5C:0050	00	00	00	00	01	00	00	00	00	43	56	02	00	00	00	00
0E5C:0060	D8	B4	4C	CD	21	00	43	06	00	00	00	56	02	00	00	00

Indique, através da análise dos dados em memória (conteúdo do rectângulo) qual o valor em hexadecimal correspondente a cada uma das variáveis num1, num2 e num3. Assuma que o registo DS contém o valor 0E5Eh.

Parte Prática (Realize cada uma das perguntas em Folhas separadas)

- 1** Desenvolva um programa em Assembly que converta um número decimal (base 10) para binário (base 2). O valor a converter deve ser compreendido entre 0 e 65535, não sendo necessário validar se o mesmo se encontra na gama definida no enunciado. Os dígitos que resultam da conversão devem ser armazenados num vetor pela ordem correta (dígito mais significativo no índice 0 do vetor e assim sucessivamente até que o dígito menos significativo ocupe o maior índice). O resultado da conversão terá sempre 16 dígitos. **(3,5 Val)**

Exemplos:

Número 30654

VetorConversao 0111011110111110

- 2** Realize um programa em Assembly que, recorrendo à memória de vídeo, imprima um histograma horizontal correspondente aos elementos constituintes de um vetor VETOR1. O vetor tem no máximo 13 elementos compreendidos entre 1 e 9. Cada barra do histograma deverá ficar separada por uma linha em branco, e será preenchida com um carácter obtido a partir de um outro vetor VETOR2 com a mesma dimensão de VETOR1. A 1ª barra do histograma será preenchida com o 1º elemento do vetor VETOR2, a 2ª com o 2º e assim sucessivamente. No final de cada barra deverá ser impresso um número correspondente ao número de elementos dessa barra. Assim, assumindo por exemplo as seguintes declarações

```
VETOR1 db 9, 8, 5, 6, 7
VETOR2 db '*', '+', '%', '&
```

O ecran teria o seguinte aspeto.

(3,5 Val)

```
***** 9
+++++++ 8
%%%%%%%% 5
&&&&&&& 6
$$$$$$$ 7
...
```

NOTAS:

- A memória de vídeo, no caso de sistemas policromáticos, tem início na localização B800h:0000h.
- Implementações com recurso a interrupções não serão avaliadas.

Template Base

```
.8086
.model small
.stack 2048

dseg      segment para public 'data'
    ...
dseg      ends

cseg      segment para public 'code'
    assume cs:cseg, ds:dseg, ss:stack

Main      proc

    mov     ax, dseg
    mov     ds, ax

    ...

    mov     ah, 4CH
    int     21H
Main      endp

cseg      ends
end        main
```