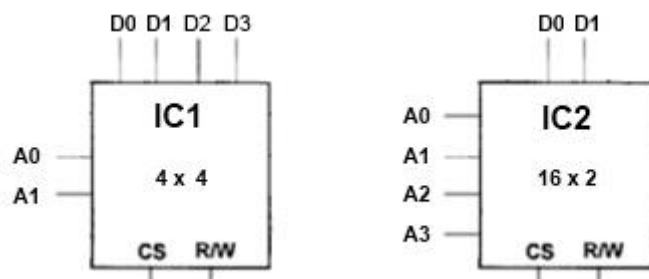


**Parte Teórica**

- 1 Justifique a necessidade de refrescamento de memória, indicando em que tipo de tecnologias é utilizada, como se processa e que implicações tem na velocidade de acesso. **(2 Val.)**
- 2 É notório o esforço que os fabricantes de microprocessadores fazem para aumentar a performance dos processadores sempre que apresentam uma nova geração. Técnicas como **multiprocessamento virtual**, **multiprocessamento real**, **“Hyper-Threading”** e **múltiplos núcleos de execução**, foram implementadas com o objectivo de aumentar o desempenho dos processadores. Descreva cada uma destas técnicas e indique quais delas poderão ser utilizadas simultaneamente. **(2 Val.)**
- 3 Considere os dois circuitos integrados de memória RAM representados na figura, onde  $A_3$ ,  $A_2$ ,  $A_1$ ,  $A_0$  representam linhas de endereço,  $D_1$ ,  $D_0$  representam linhas de dados, R/W representa a linha de leitura/escrita e CS a linha de *Chip Selection*. **(2 Val.)**



Faça um esboço associando múltiplos circuitos integrados iguais de forma a obter uma memória RAM com 32 endereços com 4 bits cada. Deverá ser indicada explicitamente a linha de CS (*Chip Selection*) da memória resultante, bem como as linhas de dados e endereços:

- a) Utilizando apenas circuitos de memória do tipo IC1. **(2 Val)**
- b) Utilizando apenas circuitos de memória do tipo IC2. **(1 Val)**

## Parte Prática

1. Desenvolva um programa em Assembly que verifique para cada elemento do vetor v1 se estes são múltiplos dos elementos na mesma posição relativa do vetor v2. Ambos os vetores possuem o mesmo tamanho, são compostos por elementos na gama [0..255] e terminam com o caracter "\$". Se um elemento de v1 numa determinada posição for múltiplo de v2 na mesma posição, deve ser colocado num terceiro vetor v3, também nessa mesma posição, o valor 0. Caso contrário, deve ser colocado nessa mesma posição de v3, o menor múltiplo de v2 maior que o elemento em v1. **(2,5 Val.)**

Exemplo:

V1: 15, 18, 19, 20, 21,..., 25

V2: 3, 5, 4, 5, 6,..., 5

V3: 0, 20, 20, 0, 24,..., 0

2. Realize um programa em Assembly que, a partir de um vetor VECT1, com n números de 8 bits contendo em cada posição valores que correspondem a atributos de cor de determinados caracteres presentes no ecrã, altere os atributos de todos os caracteres presentes no ecrã com atributos iguais aos presentes nesse vetor, substituindo-os pelos atributos que constam num outro vetor VECT2 também de tamanho n. Os atributos a alterar encontram-se na mesma posição dos dois vetores. Assim, se por exemplo, os atributos de um determinado caracter correspondem ao valor presente na posição y do VECT1 esses atributos serão substituídos pelo valor presente na posição y de VECT2. O valor que indica o tamanho n dos vetores encontra-se na variável TAM. **(2,5 Val.)**

Nota: A memória de vídeo, no caso de sistemas policromáticos, tem início na localização B800h:0000h.

```
.8086
.model small
.stack 2048

DATA_HERE    SEGMENT
...
DATA_HERE    ENDS
CODE_HERE    SEGMENT
    ASSUME CS:CODE_HERE, DS:DATA_HERE
START:mov     ax, DATA_HERE
    mov     ds, ax
    mov     bx, 0b800h
    mov     es, bx
    ...
    mov     ah, 4ch
    int     21h
CODE_HERE    ENDS
END START
```

BOA SORTE! ☺