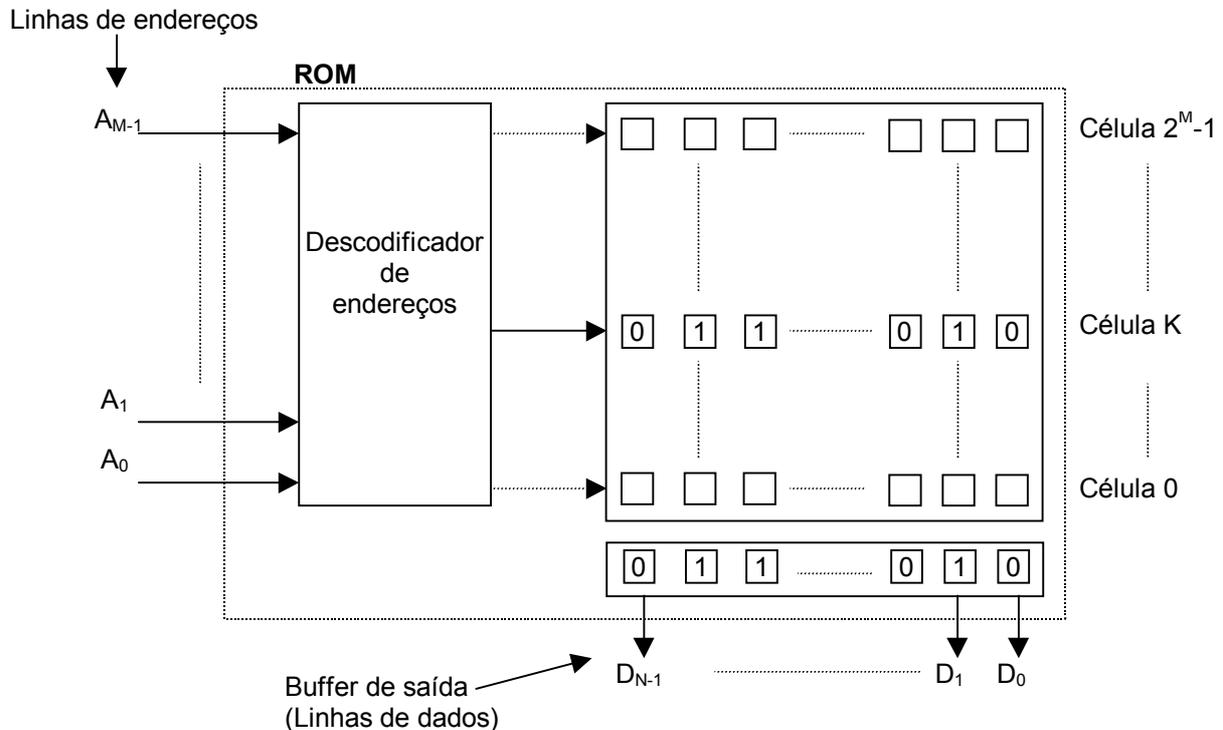


## Memórias ROM (“Read-Only Memory”)

As memórias ROM, também designadas por memórias mortas, são constituídas por uma matriz de dispositivos com capacidade para armazenar um bit de informação. Esta matriz é organizada por células/palavras constituídas por um número determinado de bits. São memórias não voláteis, isto é, não perdem a informação que armazenam depois de retirada a alimentação. Depois de gravadas/programadas (armazenamento da informação digital binária), estas memórias apenas permitem operações de leitura, resultando desta característica a designação ROM (“Read-Only Memory”).

### Estrutura interna



O descodificador de endereços activa a célula de memória K correspondente ao valor binário colocado nas linhas de endereço  $A_{M-1}..A_0$ . O valor da célula seleccionada é colocado, através de um buffer de saída, nas linhas de dados  $D_{N-1}..D_0$ .

A capacidade total de uma memória ROM depende do número de linhas de endereço (M) e do número de linhas de dados (N). Por exemplo, uma ROM com 13 linhas de endereços e 8 linhas de dados, disponibiliza  $2^M=2^{13}$  células/palavras de N=8 bits ( $2^{13} \times 8 \text{ bits} = 8K \times 8 \text{ bits} = 64Kb$ ).

### Tipos de memórias ROM

- ◆ ROM – o conteúdo das células é gravado/programado durante o processo de fabrico. Não podem ser apagadas, não sendo possível a sua reutilização.
- ◆ PROM (“Programmable ROM”) – o conteúdo das células é gravado/programado pelo utilizador recorrendo a um programador (Programador de PROM’s). Apenas podem ser gravadas uma única vez, não podendo ser reprogramadas tendo em vista a sua reutilização.
- ◆ EPROM (“Erasable Programmable ROM”) – o conteúdo das células é gravado/programado pelo utilizador recorrendo a um programador (Programador de EPROM’s). Podem ser apagadas através da exposição a raios ultra-violetas durante um período de tempo (5 a 20 minutos). Neste processo todos os bits da memória são colocados a 1, sendo possível a sua reutilização.
- ◆ EEPROM (“Electrically Erasable Programmable ROM”) – idêntico à EPROM mas com a possibilidade de ser reprogramada através de sinais eléctricos no próprio sistema de destino.

## Utilização de memórias ROM na implementação de circuitos

As memórias ROM constituem o primeiro exemplo de entre vários tipos de dispositivos de lógica programável (PLD-“Programmable Logic Devices”).

### Implementação de circuitos combinacionais

Qualquer função combinacional é directamente realizável desde que se disponha de um circuito integrado ROM cujo número de linhas de endereços é igual ou superior ao número de variáveis da função. A utilização de ROM's na implementação de circuitos combinacionais é um processo bastante simples. De facto, uma ROM com M linhas de endereços e N linhas de dados corresponde à implementação directa de uma tabela de verdade com M variáveis de entrada e N funções de saída. Neste tipo de solução não é necessário efectuar-se qualquer processo de simplificação das funções de saída.

Exemplo:

Circuito conversor BCD-7 segmentos

Variáveis de entrada: DCBA (activas a 1)

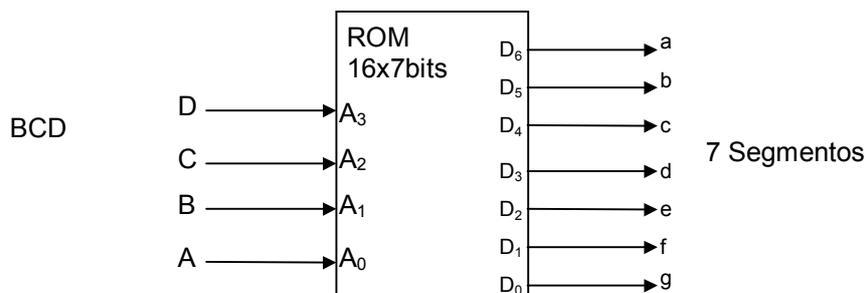
Variáveis de saída: segmentos a, b, c, d, e, f, g (activos a 0)

Para a implementação deste circuito combinacional é necessária uma memória ROM com 4 linhas de endereços ( $2^4=16$  células de memória) e 7 linhas de dados (células de memória constituídas por 7 bits), a que corresponde uma capacidade total de 16x7 bits.

Tabela de verdade /Conteúdo das células de memória

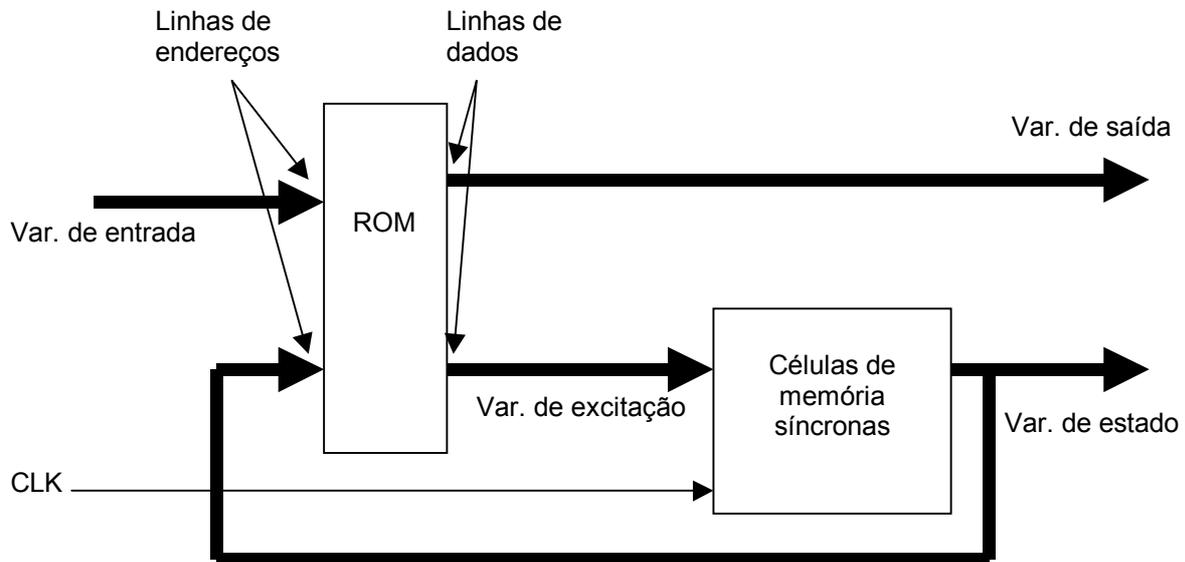
| BCD      |                |                |                |                | 7 Segmentos    |                |                |                |                |                |                |
|----------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| Nº       | D              | C              | B              | A              | a              | b              | c              | d              | e              | f              | g              |
| Endereço |                |                |                |                | Conteúdo       |                |                |                |                |                |                |
|          | A <sub>3</sub> | A <sub>2</sub> | A <sub>1</sub> | A <sub>0</sub> | D <sub>6</sub> | D <sub>5</sub> | D <sub>4</sub> | D <sub>3</sub> | D <sub>2</sub> | D <sub>1</sub> | D <sub>0</sub> |
| 0        | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 1              |
| 1        | 0              | 0              | 0              | 1              | 1              | 0              | 0              | 1              | 1              | 1              | 1              |
| 2        | 0              | 0              | 1              | 0              | 0              | 0              | 1              | 0              | 0              | 1              | 0              |
| 3        | 0              | 0              | 1              | 1              | 0              | 0              | 0              | 0              | 1              | 1              | 0              |
| 4        | 0              | 1              | 0              | 0              | 1              | 0              | 0              | 1              | 1              | 0              | 0              |
| 5        | 0              | 1              | 0              | 1              | 0              | 1              | 0              | 0              | 1              | 0              | 0              |
| 6        | 0              | 1              | 1              | 0              | 1              | 1              | 0              | 0              | 0              | 0              | 0              |
| 7        | 0              | 1              | 1              | 1              | 0              | 0              | 0              | 1              | 1              | 1              | 1              |
| 8        | 1              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              |
| 9        | 1              | 0              | 0              | 1              | 0              | 0              | 0              | 1              | 1              | 0              | 0              |
|          | 1              | 0              | 1              | 0              | x              | x              | x              | x              | x              | x              | x              |
|          | 1              | 0              | 1              | 1              | x              | x              | x              | x              | x              | x              | x              |
|          | 1              | 1              | 0              | 0              | x              | x              | x              | x              | x              | x              | x              |
|          | 1              | 1              | 0              | 1              | x              | x              | x              | x              | x              | x              | x              |
|          | 1              | 1              | 1              | 0              | x              | x              | x              | x              | x              | x              | x              |
|          | 1              | 1              | 1              | 1              | x              | x              | x              | x              | x              | x              | x              |

Diagrama lógico



## Implementação de circuitos sequenciais síncronos

As memórias ROM podem ser utilizadas na síntese dos circuitos combinacionais que constituem o decodificador de estado seguinte e o decodificador de saída. O bloco de células de memória dos circuitos sequenciais síncronos é implementado externamente à ROM de acordo com a estrutura geral da figura.



As linhas de endereços são alimentadas com as variáveis de estado e as variáveis de entrada. As linhas de dados correspondem às variáveis de excitação (entradas do módulo de células de memória) e às variáveis de saída.

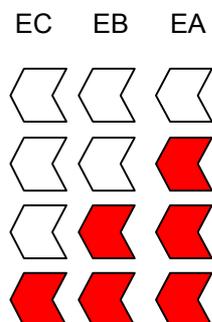
Exemplo:

Pretende-se controlar um sistema de sinalização constituído por seis indicadores luminosos (EC, EB, EA, DA, DB, DC). Estes indicadores luminosos constituíram, em tempos, parte integrante do sistema de sinalização de alguns veículos automóveis.

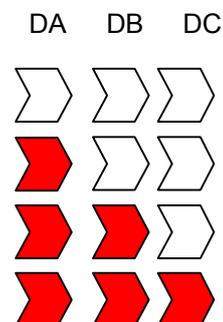


A indicação da mudança de direcção é obtida pelo acender, em sequência, dos indicadores luminosos tal como se descreve na figura seguinte. A sinalização de perigo ("quatro-piscas ligados") é obtida pelo ligar e desligar dos seis indicadores em simultâneo.

Mudança de direcção  
para a esquerda



Mudança de direcção  
para a direita



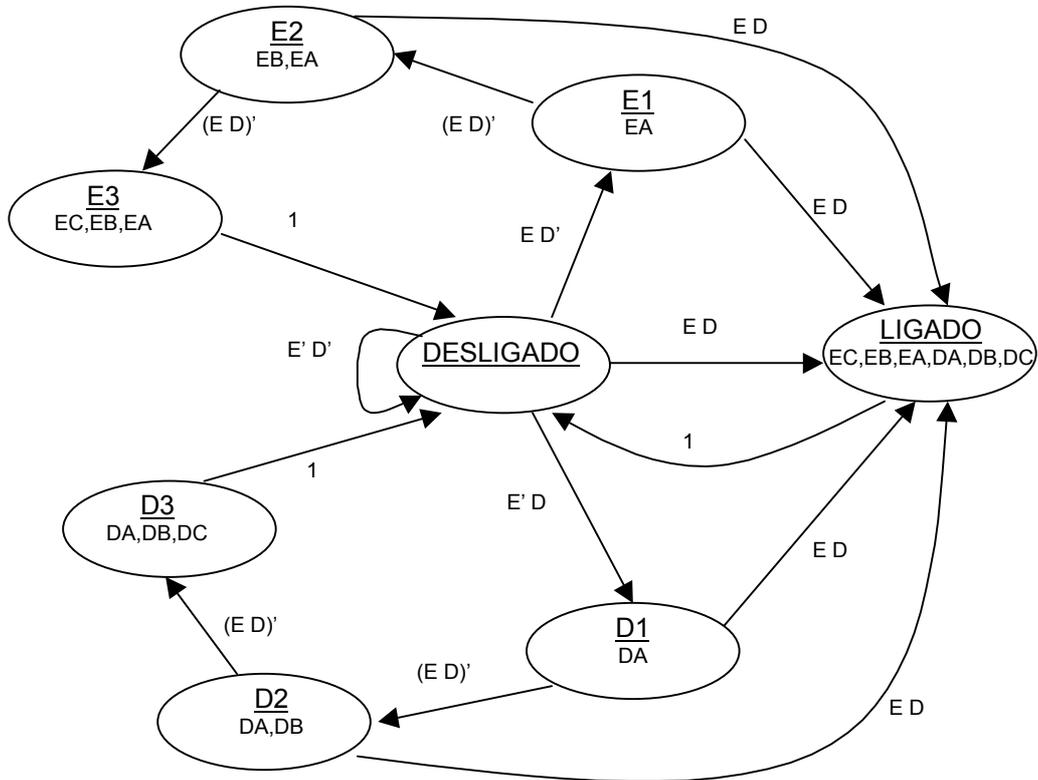


### Diagrama de estados

De acordo com as especificações funcionais, obtém-se o diagrama de estado da figura. Ao contrário do que é habitual, cada transição é acompanhada da expressão lógica (equação de transição) que condiciona a transição para o estado seguinte. Por exemplo:

- do estado DESLIGADO, o estado do sistema mantém-se se  $E=D=0$  (eq. transição =  $E' D'$ ), transita-se para o estado E1 se  $E=1$  e  $D=0$  (eq. transição =  $E D'$ ), transita-se para o estado D1 se  $E=0$  e  $D=1$  (eq. transição =  $E' D$ ), transita-se para o estado LIGADO se  $E=D=1$  (eq. transição =  $E D$ );
- do estado E2, transita-se para o estado LIGADO se  $E=D=1$  (eq. transição =  $E D$ ), transita-se para o estado E3 se  $E=0$  ou  $D=0$  (eq. transição =  $(E D)'$ ).

Na representação do estado, apenas se faz a referência às variáveis de saída que se encontram activas. Por exemplo, no estado DESLIGADO, nenhuma das saídas está activa, no estado D2, estão activas as saídas DA e DB.



**Tabela de estados**

| Estado presente<br>( $Q_C, Q_B, Q_A$ ) | E | D | Var. excitação/Estado seguinte<br>( $D_C, D_B, D_A$ ) / ( $Q_C, Q_B, Q_A$ ) | EC | EB | EA | DA | DB | DC |
|--|---|---|---|----|----|----|----|----|----|
| DESLIGADO(000)                         | 0 | 0 | DESLIGADO(000)  | 0  | 0  | 0  | 0  | 0  | 0  |
|  | 0 | 1 | D1(100)   |    |    |    |    |    |    |
|  | 1 | 0 | E1(001)   |    |    |    |    |    |    |
|  | 1 | 1 | LIGADO(111)   |    |    |    |    |    |    |
| E1(001)                                | 0 | 0 | E2(010)   | 0  | 0  | 1  | 0  | 0  | 0  |
|  | 0 | 1 | E2(010)   |    |    |    |    |    |    |
|  | 1 | 0 | E2(010)   |    |    |    |    |    |    |
|  | 1 | 1 | LIGADO(111)   |    |    |    |    |    |    |
| E2(010)                                | 0 | 0 | E3(011)   | 0  | 1  | 1  | 0  | 0  | 0  |
|  | 0 | 1 | E3(011)   |    |    |    |    |    |    |
|  | 1 | 0 | E3(011)   |    |    |    |    |    |    |
|  | 1 | 1 | LIGADO(111)   |    |    |    |    |    |    |
| E3(011)                                | X | X | DESLIGADO(000)  | 1  | 1  | 1  | 0  | 0  | 0  |
| D1(100)                                | 0 | 0 | D2(101)   | 0  | 0  | 0  | 1  | 0  | 0  |
|  | 0 | 1 | D2(101)   |    |    |    |    |    |    |
|  | 1 | 0 | D2(101)   |    |    |    |    |    |    |
|  | 1 | 1 | LIGADO(111)   |    |    |    |    |    |    |
| D2(101)                                | 0 | 0 | D3(110)   | 0  | 0  | 0  | 1  | 1  | 0  |
|  | 0 | 1 | D3(110)   |    |    |    |    |    |    |
|  | 1 | 0 | D3(110)   |    |    |    |    |    |    |
|  | 1 | 1 | LIGADO(111)   |    |    |    |    |    |    |
| D3(110)                                | X | X | DESLIGADO(000)  | 0  | 0  | 0  | 1  | 1  | 1  |
| LIGADO(111)                            | X | X | DESLIGADO(000)  | 1  | 1  | 1  | 1  | 1  | 1  |



## Diagrama lógico/implementação

Depois de programada, a EPROM é colocada no circuito em modo de leitura. De acordo com as especificações apresentadas pelo fabricante, as entradas CE# e OE# são colocadas a  $V_{IL}$  (0V) e as entradas  $V_{PP}$  e PGM# são colocadas a  $V_{IH}$  (+5V).

Dos oito flip-flops D do registo de dados 74LS374, apenas são utilizados os três primeiros, correspondente às entradas  $D_2..D_0$  e respectivas saídas  $O_2..O_0$ . Sendo as saídas em três estados, a entrada OE' é colocada a zero.

