

**MEMÓRIAS E
CIRCUITOS DE
LÓGICA PROGRAMÁVEL**

- **SUMÁRIO:**

- MEMÓRIAS INTEGRADAS

- RAM

- ROM

- PROM

- EPROM

- LÓGICA PROGRAMÁVEL

- PLAs

- PALs

- FPGAs

● MEMÓRIAS

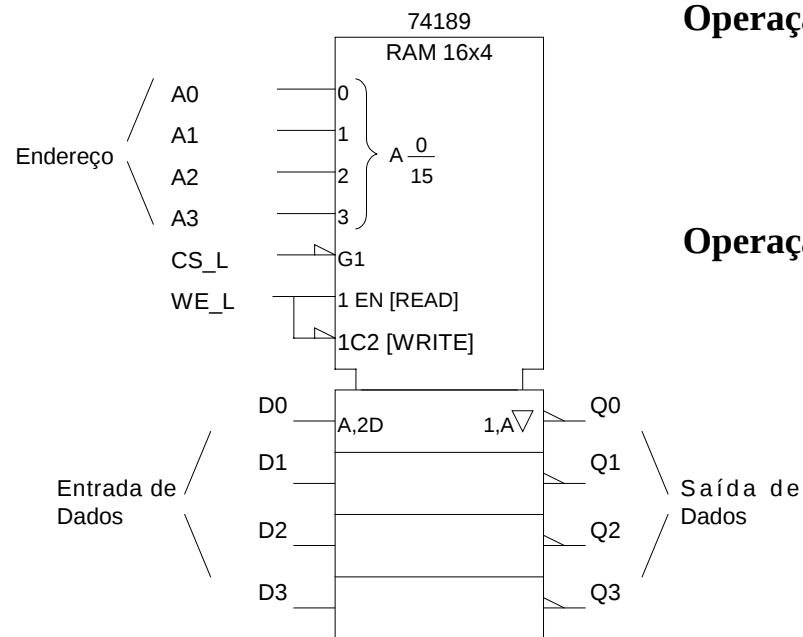
Na sequência do estudo de **FFs**, como elementos básicos de memória de 1 bit, e de **registos** como elementos capazes de armazenar 1 palavra de N bits, surge, naturalmente, a necessidade de elementos que permitam o armazenamento conjuntos de K palavras de N bits. Esses elementos designam-se por **memórias**.

Na presente abordagem, referem-se em especial os aspectos funcionais e as estruturas lógicas destes circuitos, deixando-se os detalhes da electrónica para cadeiras subsequentes.

MEMÓRIAS - 4

● RAM - MODELO SIMPLIFICADO

RAM – “Random Access Memory”



Operação de Escrita:

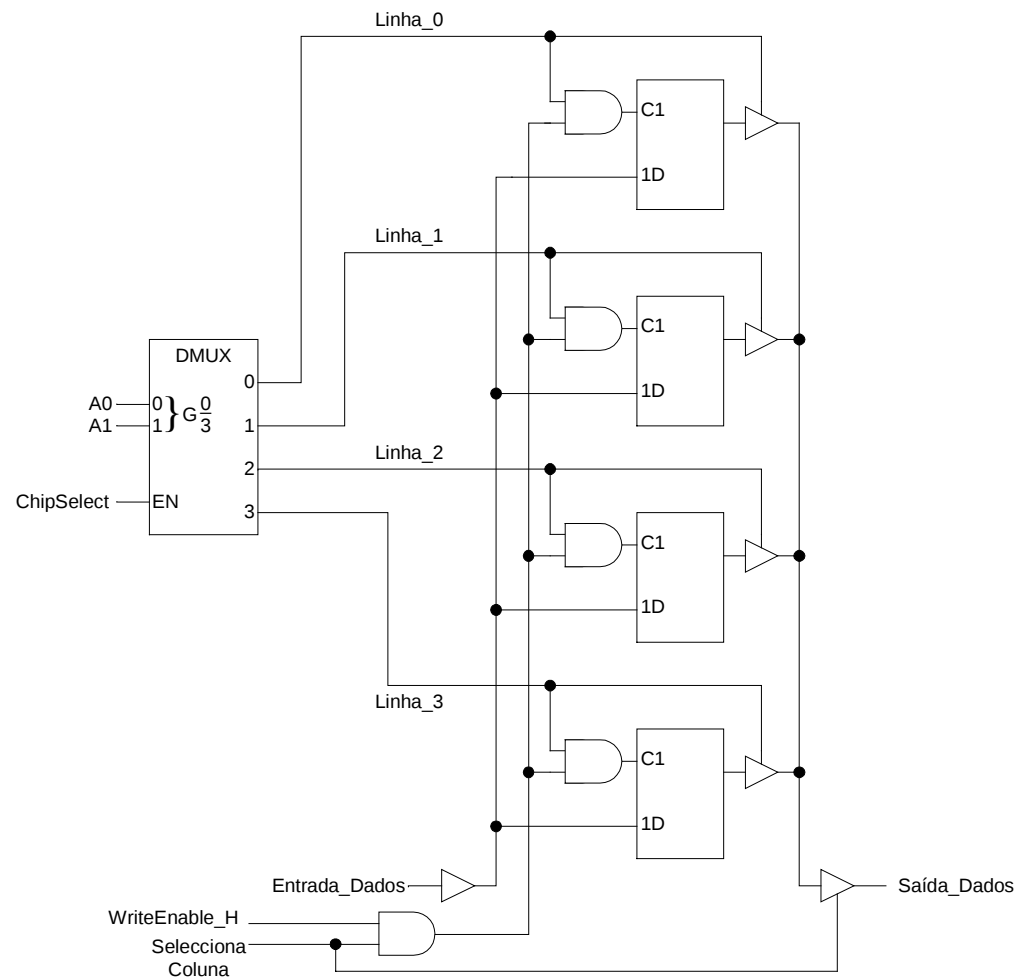
- (1) Colocar **endereço** nas linhas de endereço;
- (2) Colocar dados nas linhas de entrada de dados (Ds);
- (3) Activar CS_L e WE_L (L - Escrita).

Operação de Leitura:

- (1) Colocar **endereço** nas linhas de endereço;
- (2) Activar CS_L e Desactivar WE_L (H – Leitura) ;
- (3) Ler dados pretendidos nas linhas de saída (Os).

MEMÓRIAS - 5

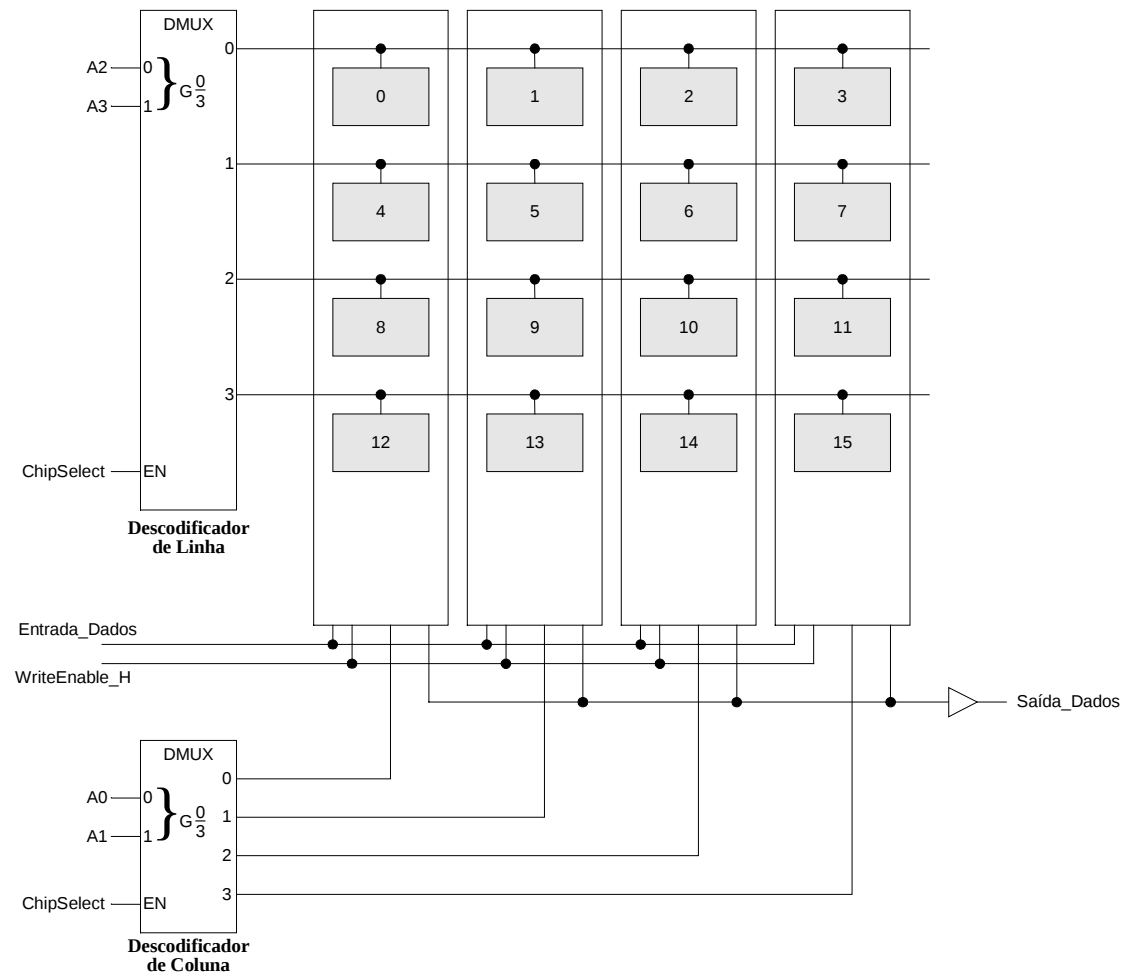
● RAM - MODELO LÓGICO DA ESTRUTURA BASE



MEMÓRIAS - 6

● RAM - ESTRUTURA TIPO

EXEMPLO:
RAM 16x1,
organizada
segundo uma
matriz de 4x4 bits.

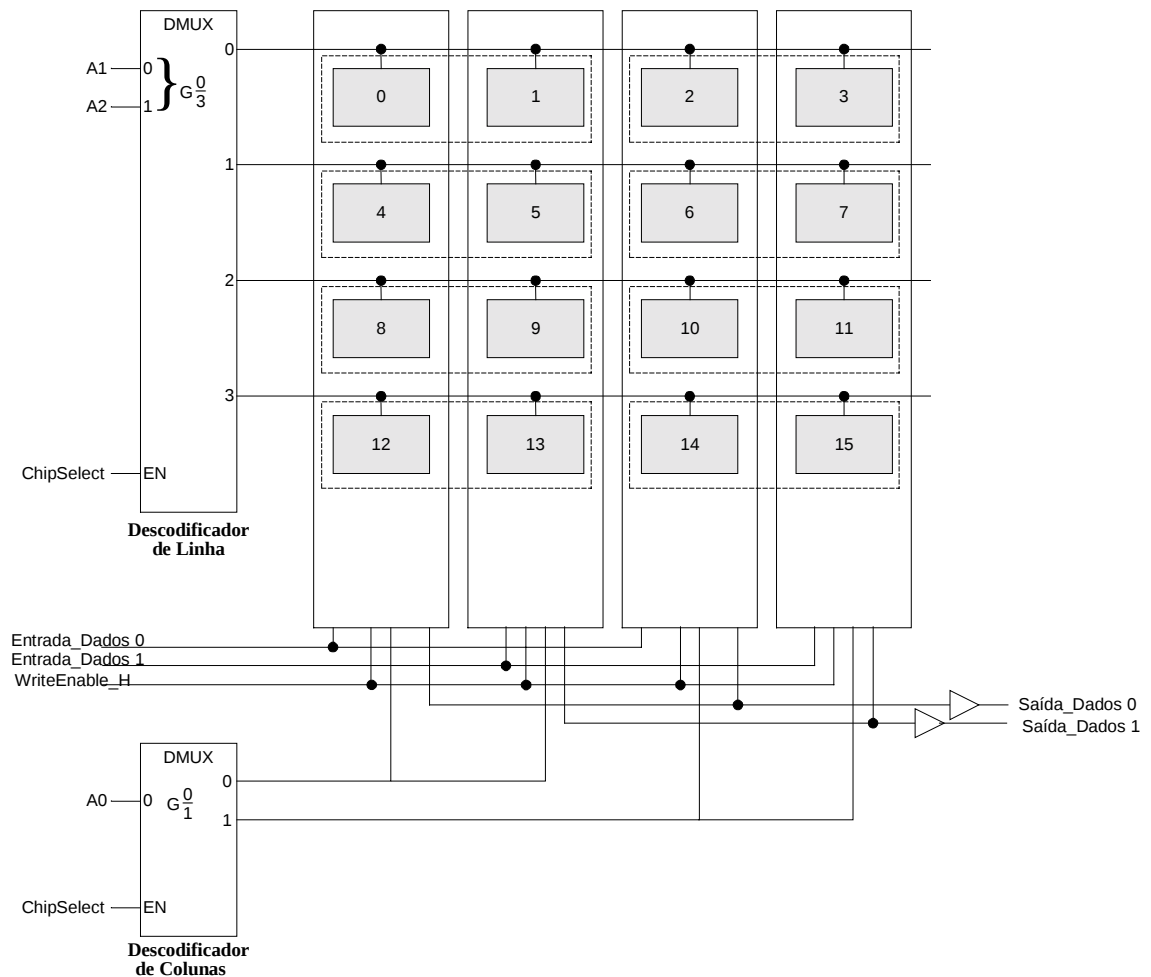


MEMÓRIAS - 7

● RAM - ESTRUTURA TIPO

EXEMPLO:

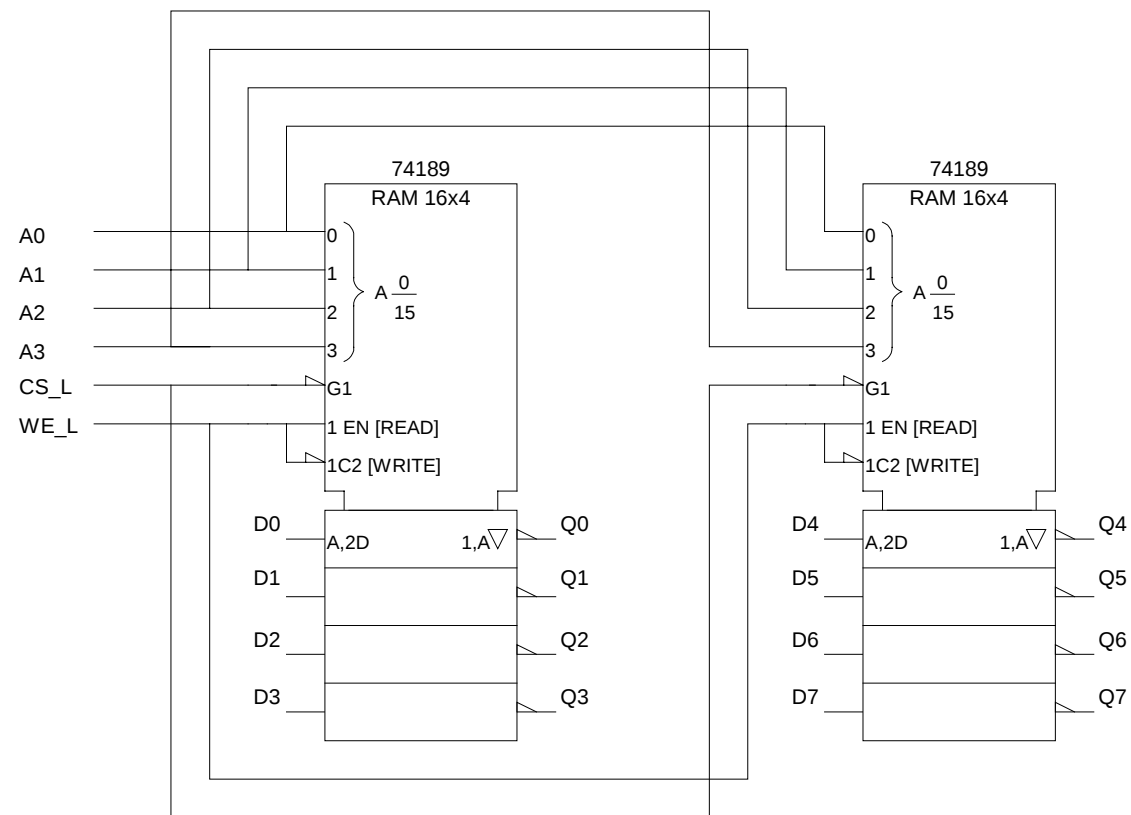
Uma RAM 8x2 é organizada segundo a mesma matriz de 4x4 bits, que é agora, de facto, uma matriz de 4x2 palavras de 2 bits cada.



MEMÓRIAS - 8

● RAM - ASSOCIAÇÃO

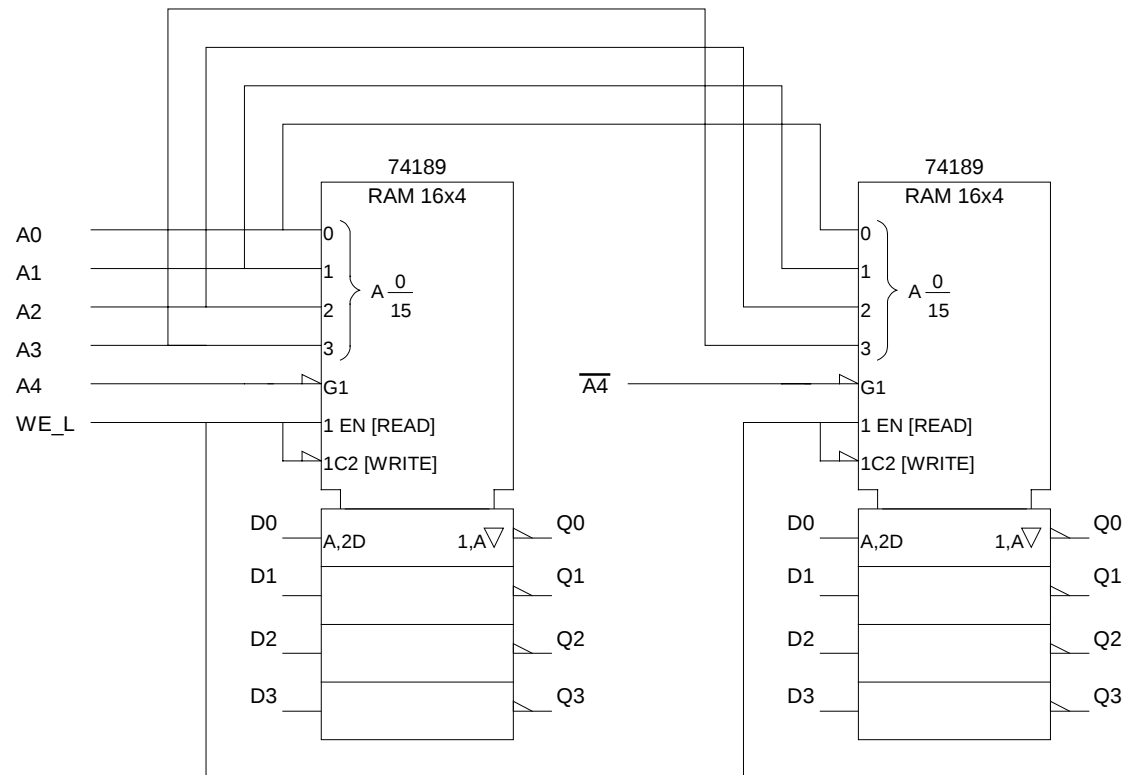
Aumento da Dimensão das Palavras (de 4 para 8 bits):



MEMÓRIAS - 9

● RAM - ASSOCIAÇÃO

Aumento do Número de Palavras (de 16 para 32 palavras):



Nota: Por simplicidade omitiram-se as ligações das entradas D e das saídas Q das duas memórias ao BUS de Dados

● ROM - DEFINIÇÕES

ROM – “Read Only Memory” – Memória apenas de leitura, sendo o conteúdo definido durante a fabricação.

ROMs PROGRAMÁVEIS - Memória apenas de leitura, sendo o conteúdo definido pelo utilizador.

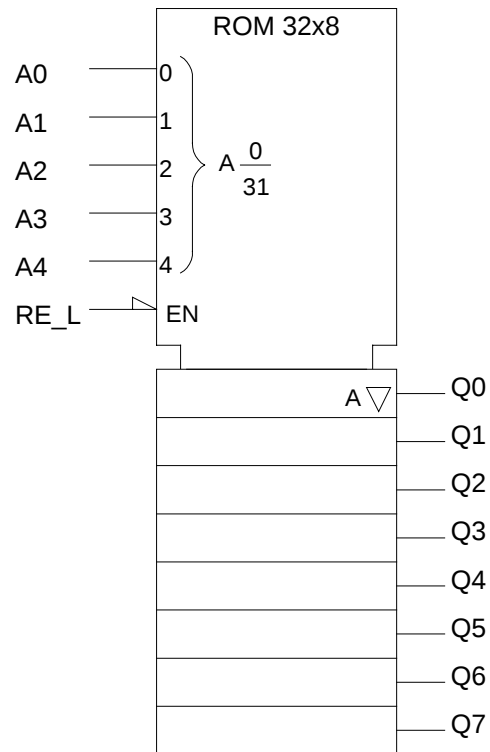
PROM - ROM Programável (uma única vez).

EPROM - ROM Programável com capacidade de reprogramação.
Neste caso, a memória pode ser apagada através da aplicação prolongada a radiação ultravioleta intensa.

Nota: existem outras classes de memórias não abordadas neste estudo.

MEMÓRIAS - 11

● ROM - SIMBOLOGIA

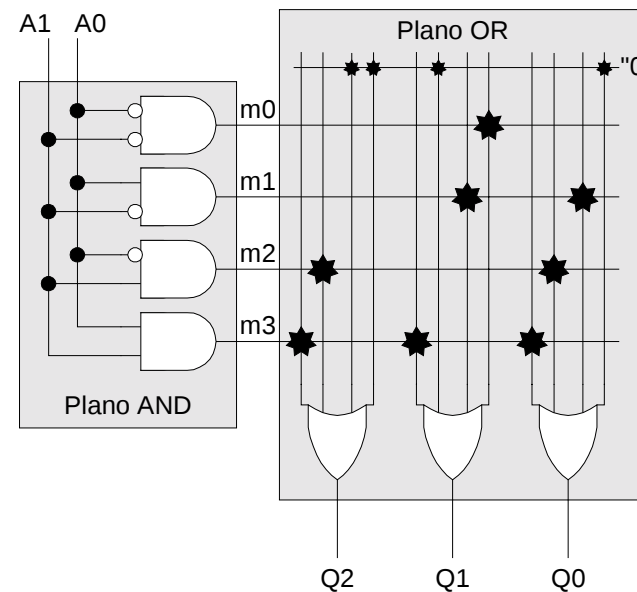


MEMÓRIAS - 12

● ROM – ESTRUTURA TIPO

EXEMPLO: ROM 4x3 que armazena os 4 primeiros números primos:

A_1	A_0	Q_2	Q_1	Q_0	
0	0	0	1	0	2
0	1	0	1	1	3
1	0	1	0	1	5
1	1	1	1	1	7



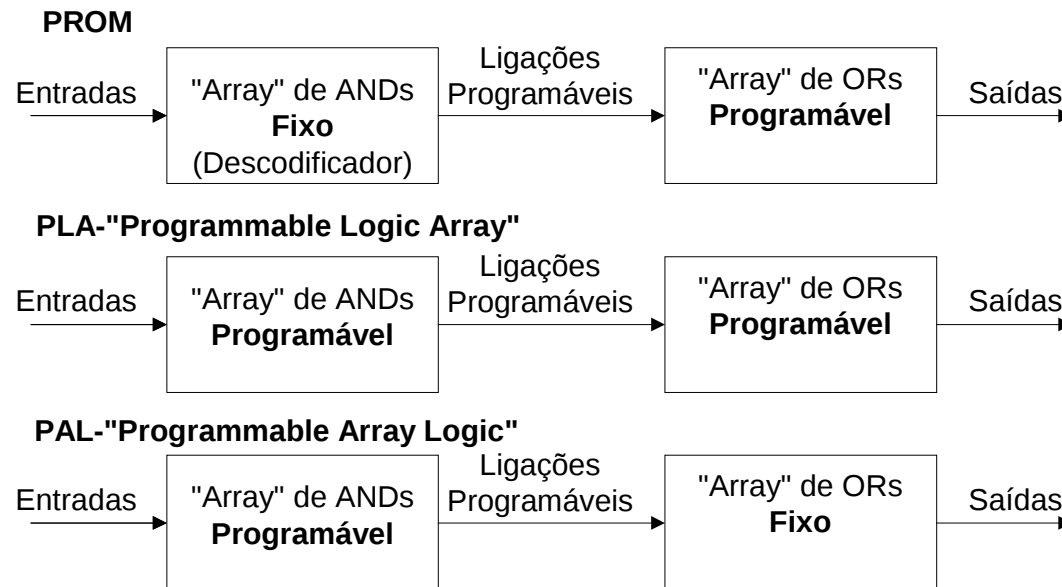
A programação da ROM é concretizada pela efectivação das ligações entre as saídas das ANDs e as entradas das ORs.

Nota: As linhas sem ligação são, por defeito, ligadas a “0” (elemento neutro da OR).

● LÓGICA PROGRAMÁVEL

Nesta secção referem-se algumas estruturas, derivadas da estrutura básica de uma ROM, cujo objectivo está centrado na sua programação para implementação de funções combinatórias e não na sua utilização como memória.

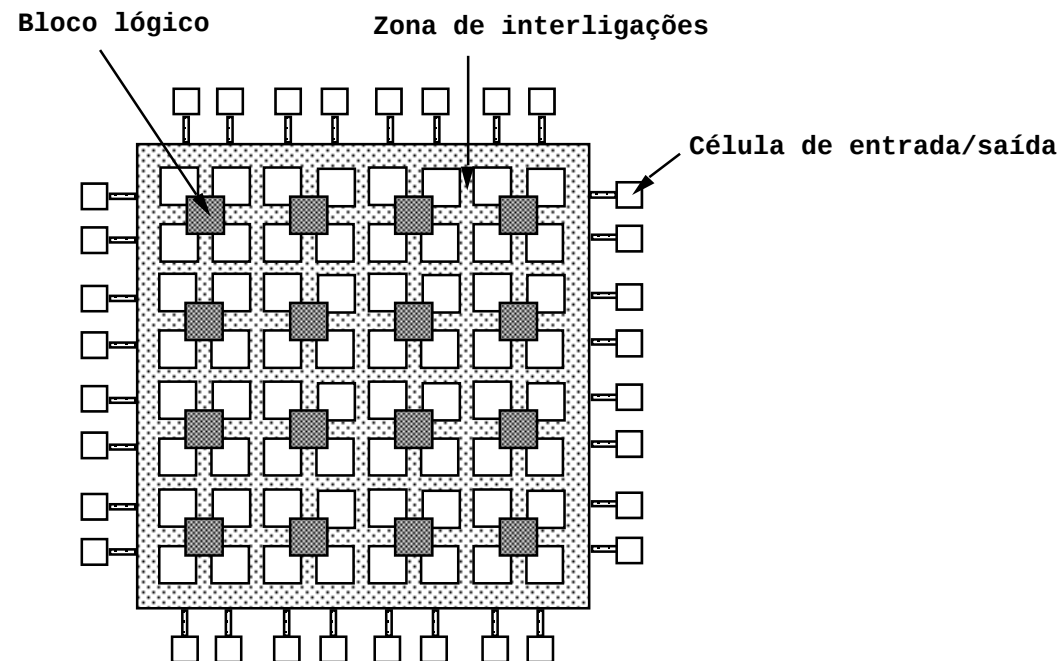
PLDs - Programmable Logic Devices



● FPGAs – “FIELD PROGRAMMABLE GATE-ARRAYS”

As FPGAs são circuitos programáveis que permitem realizar circuitos de complexidade equivalente até algumas centenas de milhar de portas lógicas.

O tipo de programação usado nas FPGAs mais populares baseia-se na utilização de memórias RAM distribuídas para realizar as funções lógicas e controlar as interligações do circuito.



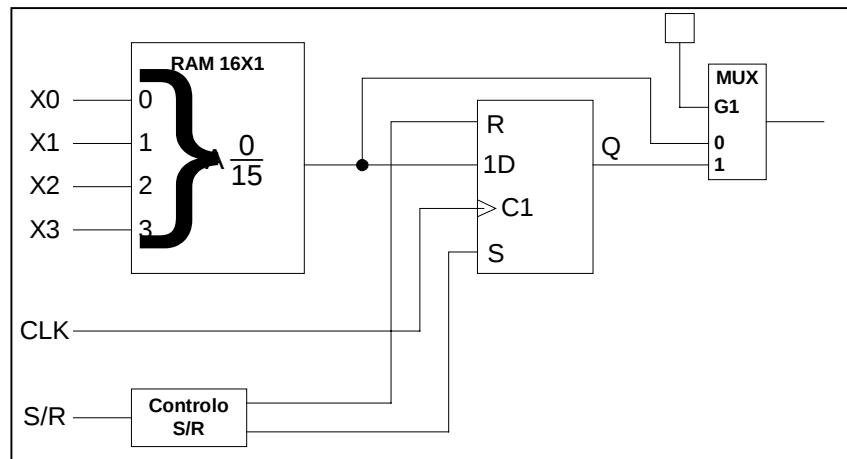
● BLOCO LÓGICO BÁSICO DE UMA FPGA

O bloco lógico básico de uma FPGA é constituído tipicamente por uma LUT – “Look-Up Table” e um FF D.

A LUT é simplesmente uma pequena RAM (são habitualmente utilizadas LUTs de 4 entradas = RAM de 16 bits) que, para efeitos de utilização do circuito, funciona apenas em modo de leitura (a escrita corresponde à programação da função lógica).

A programação da lógica consiste no armazenamento na LUT da tabela de verdade da função lógica a realizar.

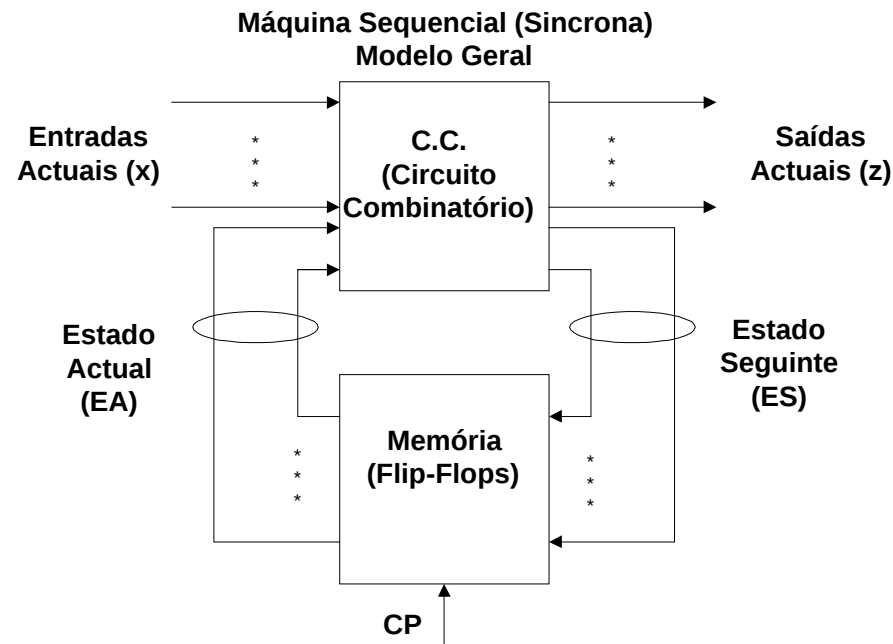
O controlo (programável) do MUX permite incluir/excluir o FF no/do circuito.



Bloco Básico – Modelo simplificado

● MÁQUINAS DE ESTADO E MEMÓRIAS

A estrutura de uma máquina de estados permite que se utilizem memórias (para implementar o circuito combinatório) conferindo a possibilidade tanto de tornar os sistemas de controlo programáveis, portanto mais versáteis, como, ainda, o desenvolvimento de sistemas funcionalmente mais complexos sem o aumento correspondente da complexidade do hardware. Qual o preço a pagar por esta solução em relação à implementação clássica?

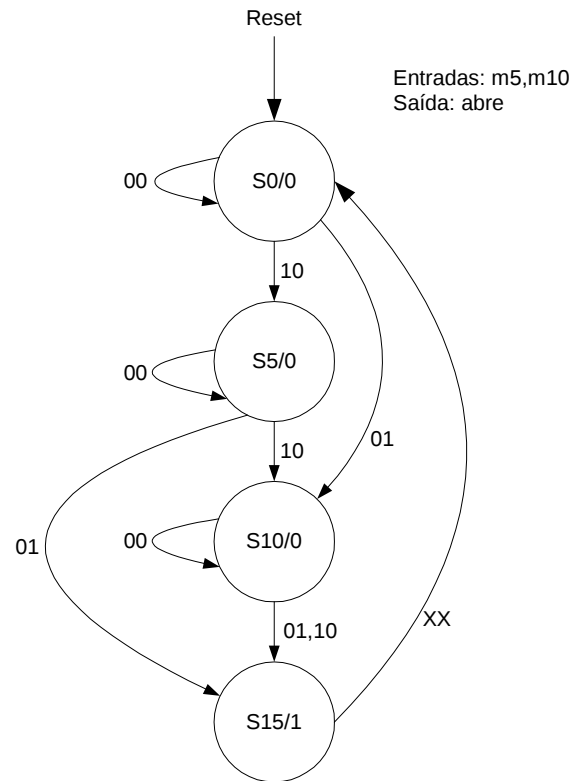


MEMÓRIAS - 17

● MÁQUINAS DE ESTADO E MEMÓRIAS (cont.)

EXEMPLO: Máquina de Distribuição de Pastilhas

(conforme analisada no capítulo dos Circuitos Sequenciais Síncronos)



	Estado Actual		Entradas		Estado Seguinte		Saída ABRE
	Q1	Q0	m5	m10	Q1	Q0	
S0C	0	0	0	0	S0C	0	0
			0	1	S10C	1	0
			1	0	S5C	0	0
			1	1	X	X	0
S5C	0	1	0	0	S5C	0	0
			0	1	S15C	1	0
			1	0	S10C	1	0
			1	1	X	X	0
S10C	1	0	0	0	S10C	1	0
			0	1	S15C	1	0
			1	0	S15C	1	0
			1	1	X	X	0
S15C	1	1	0	0	S0C	0	1
			0	1	S0C	0	1
			1	0	S0C	0	1
			1	1	X	X	1

● MÁQUINAS DE ESTADO E MEMÓRIAS (cont.)

EXEMPLO: Máquina de Distribuição de Pastilhas (cont.)

Tabela de Memória

Tendo por base a Tabela de Transição de Estados e a memória RAM apresentada anteriormente utiliza-se esta memória para implementar a função do circuito combinatório, i.e., determinar o Estado Seguinte e a Saída.

O Endereço de Memória (A3 A2 A1 A0) será especificado, respectivamente, pelas variáveis de estado Q1 Q0 e pelas entradas M5 M10.

A Palavra de Memória deverá conter a informação do Estado Seguinte Q1 Q0 em (n+1) e a Saída Z nos 3 bits menos significativos.

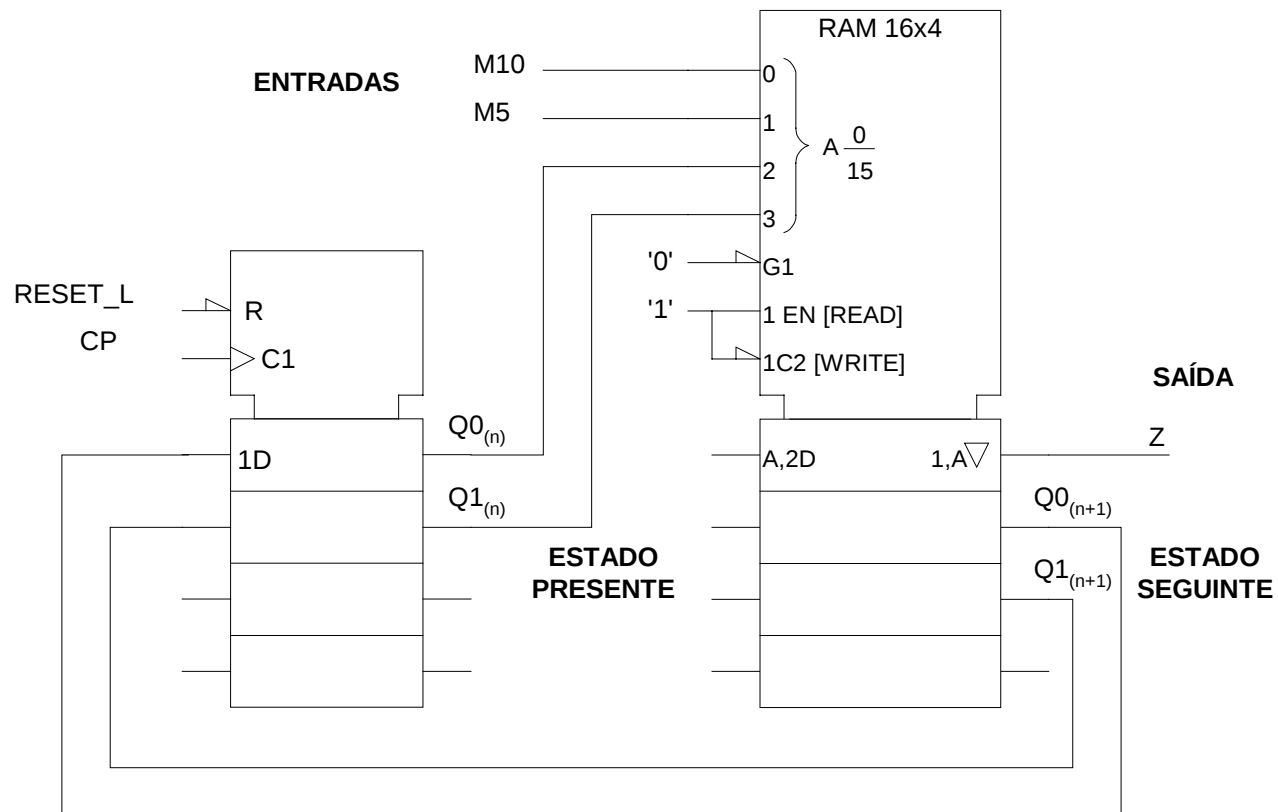
Nota: A cinzento indicam-se as entradas (endereços) não utilizadas e as saídas não consideradas (bit mais significativo).

Endereço				Conteúdo			
A3	A2	A1	A0	Q3	Q2	Q1	Q0
0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0
0	0	1	0	0	0	1	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	1	0
0	1	0	1	0	1	1	0
0	1	1	0	0	1	0	0
0	1	1	1	0	0	0	0
1	0	0	0	0	1	0	0
1	0	0	1	0	1	1	0
1	0	1	0	0	1	1	0
1	0	1	1	0	0	0	0
1	1	0	0	0	0	0	1
1	1	0	1	0	0	0	1
1	1	1	0	0	0	0	1
1	1	1	1	0	0	0	1

MEMÓRIAS - 19

● MÁQUINAS DE ESTADO E MEMÓRIAS (cont.)

EXEMPLO: Máquina de Distribuição de Pastilhas (Implementação)



BIBLIOGRAFIA

[1] M. Morris Mano, Charles R. Kime, “Logic and Computer Design Fundamentals”, Prentice-Hall International, Inc. (Capítulo 6, Secções 6.1-6.9)