TÉCNICO
LISBOA

MEMÓRIAS E CIRCUITOS DE LÓGICA PROGRAMÁVEL

SUMÁRIO:

- MEMÓRIAS INTEGRADAS
  - RAMs
  - ROMs
- LÓGICA PROGRAMÁVEL
  - PROMs
  - PLAs
  - PALs
  - FPGAs
- IMPLEMENTAÇÃO DE MÁQUINAS DE ESTADO UTILIZANDO LÓGICA PROGRAMÁVEL

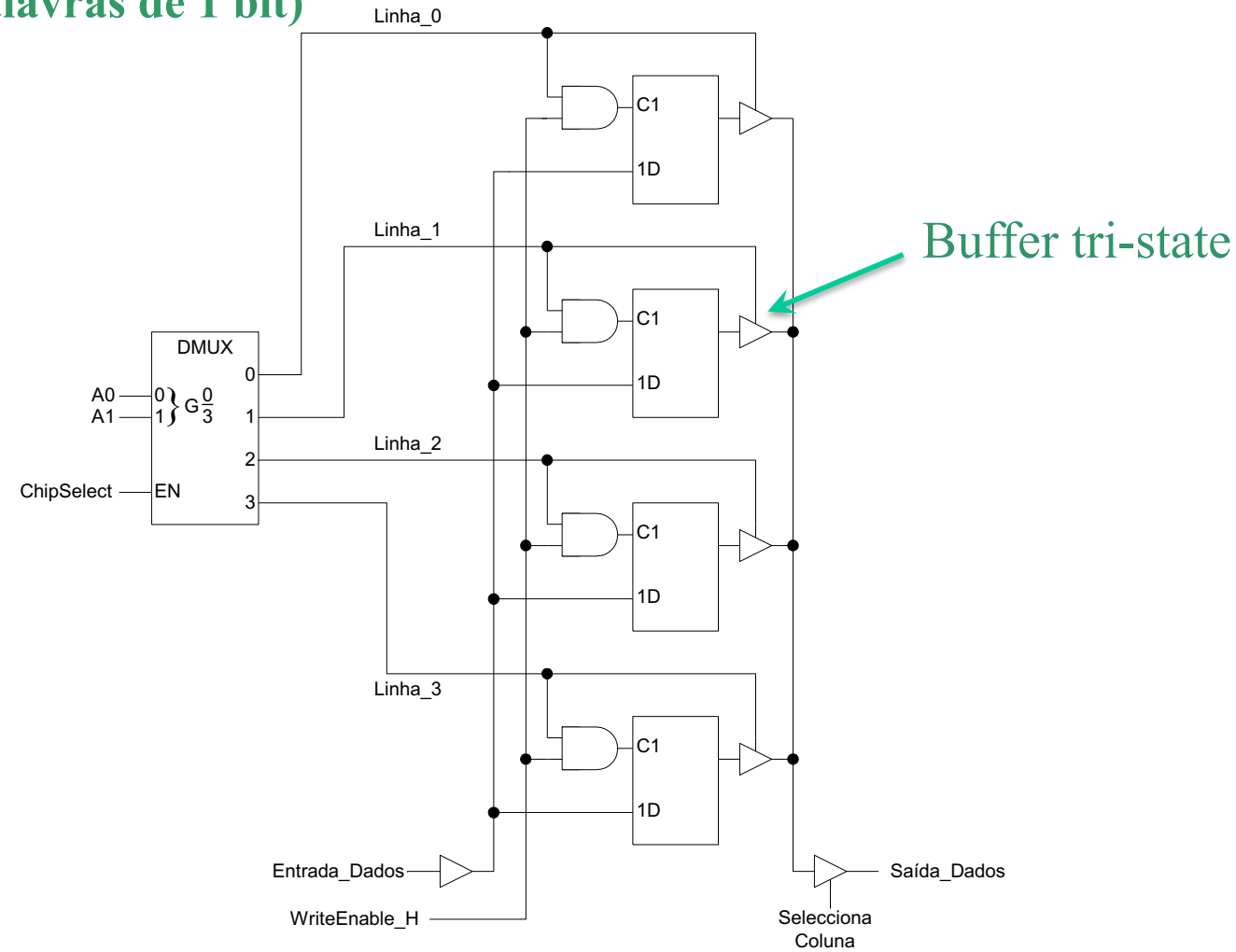
MEMÓRIAS

- **Flip-Flops**: Elementos básicos de memória; **1 bit**.
- **Registos**: Armazenamento de **1 palavra de N bits**;
- **Memórias**: Armazenamento de **K palavras de N bits**;

TIPOS de MEMÓRIAS

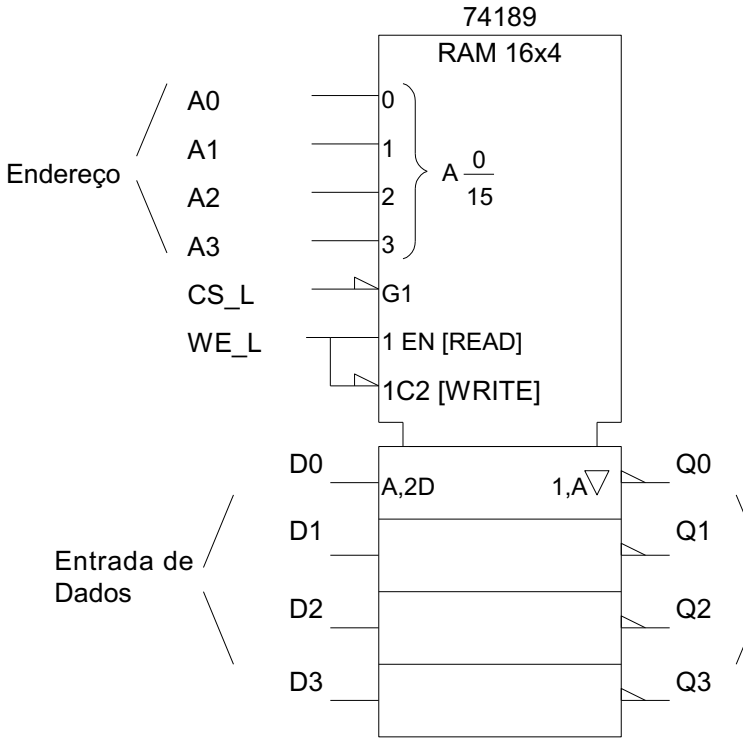
- **RAMs – Random Access Memory** (memória volátil)
- **ROMs** (não volátil)
  - ROMs (Read Only Memory): Memória apenas de leitura, sendo o conteúdo definido durante o fabrico.
  - PROM (Programable Read Only Memory): Memória programável pelo utilizador uma única vez.
  - EPROMs (Erasable Programable ROM): ROM com capacidade de reprogramação; a memória pode ser apagada através de radiação ultravioleta intensa.
  - EEPROMs (Electrically Erasable Programable ROM): ROM reprogramável que pode ser apagada electricamente (funciona a nível de blocos relativamente pequenos, e.g. bytes) .
- **Flash Memory** (tipo NAND e NOR, não volátil)
  - Programada e apagada eletricamente. Trabalha com blocos grandes e por isso é de escrita mais rápida que as EEPROM (mas muito mais lenta que RAMs). A leitura é mais lenta que RAMs e ROMs.
  - Usada para armazenamento geral em telemóveis, máquinas fotográficas, SSD, etc.

RAM (Random Access Memory) - MODELO LÓGICO DA ESTRUTURA BASE (4 palavras de 1 bit)



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) Ativar WE_L (L – Escrita);
(4) Ativar CS_L.

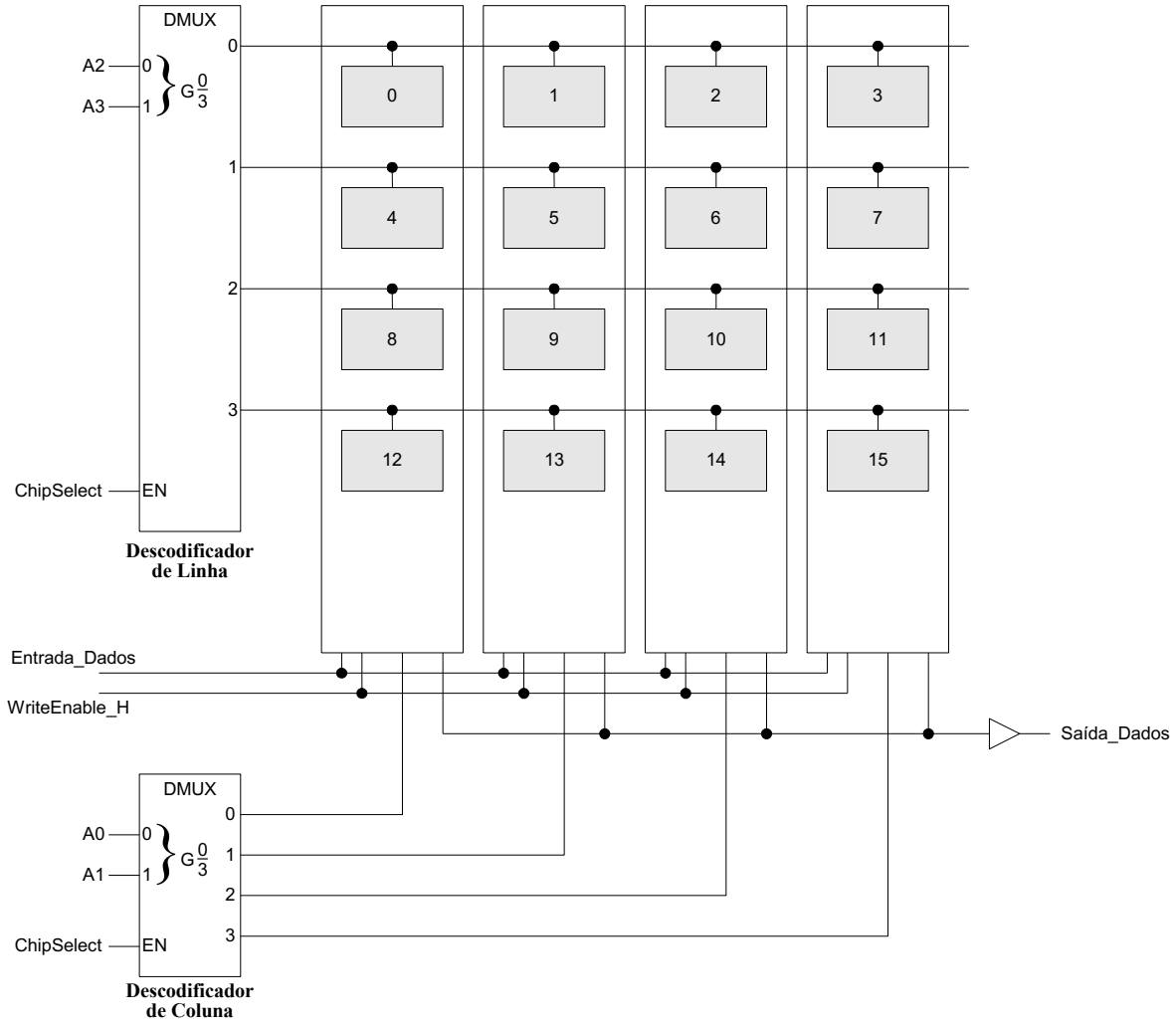
Operação de Leitura:

(1) Colocar **endereço** nas linhas de endereço;
(2) Desativar WE_L (H – Leitura);
(3) Ativar CS_L;
(4) Ler dados pretendidos nas linhas de saída (Os).

RAM - ESTRUTURA TIPO

EXEMPLO:

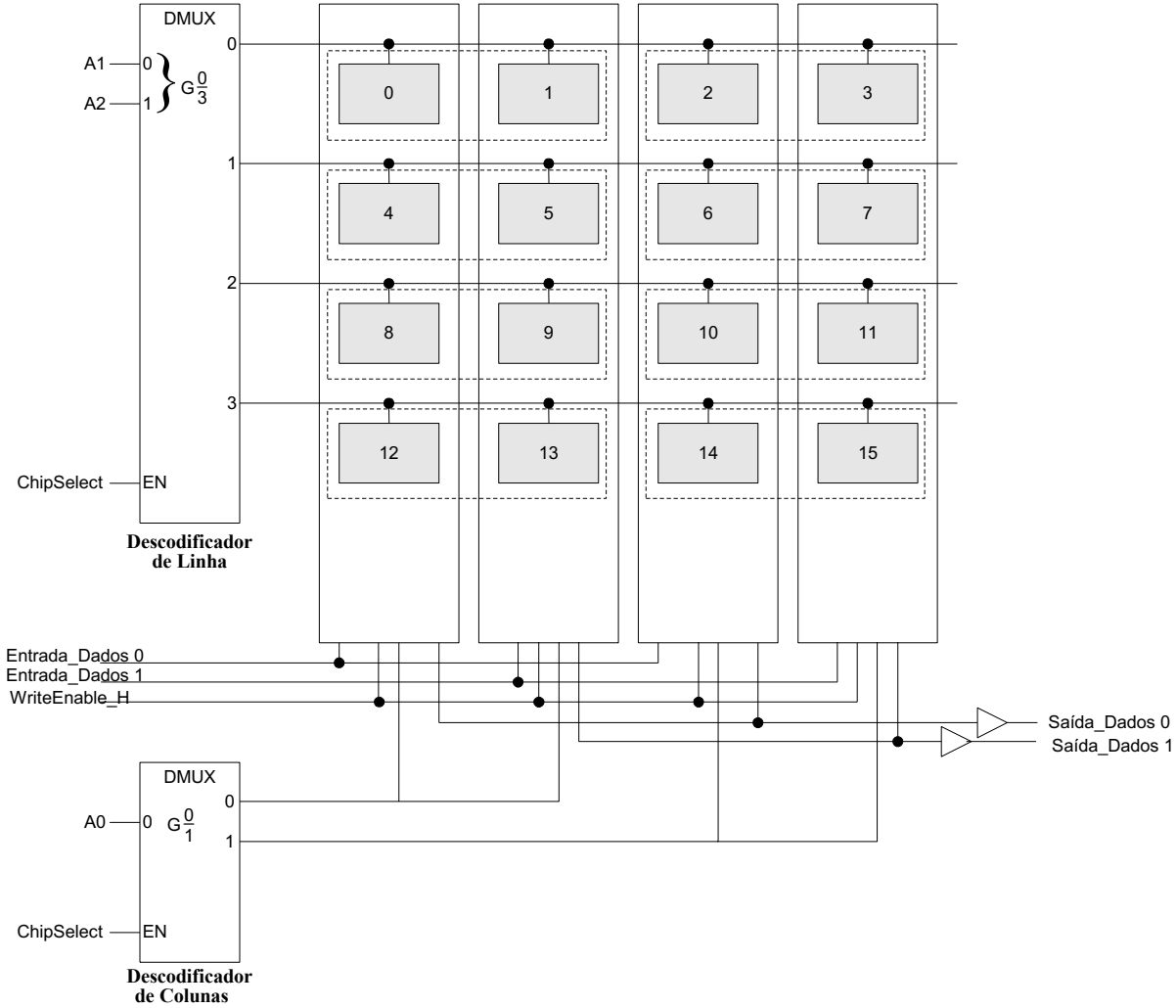
RAM 16x1 (16 linhas de 1 bit cada), organizada segundo uma matriz de 4x4 bits.



RAM - ESTRUTURA TIPO

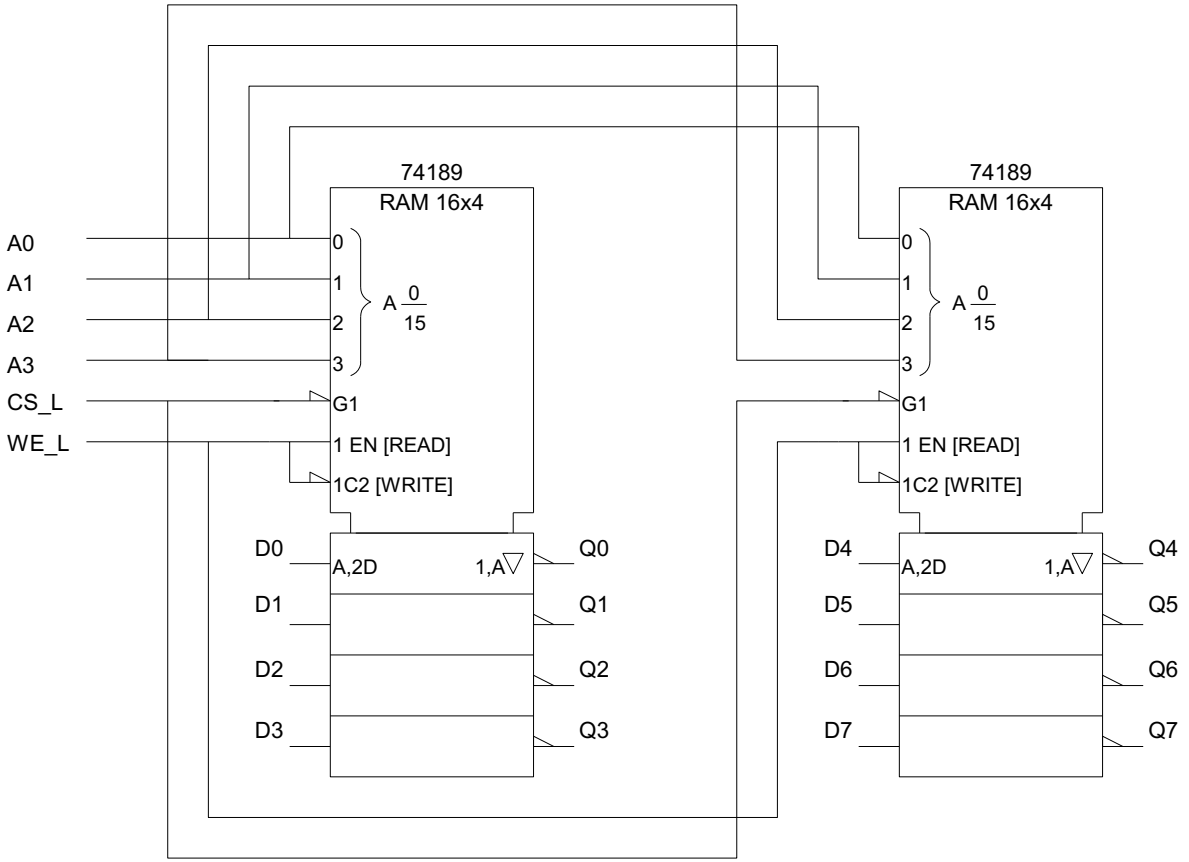
EXEMPLO:

Uma RAM 8x2 (8 linhas de 2 bits cada), organizada segundo a mesma matriz de 4x4 bits.



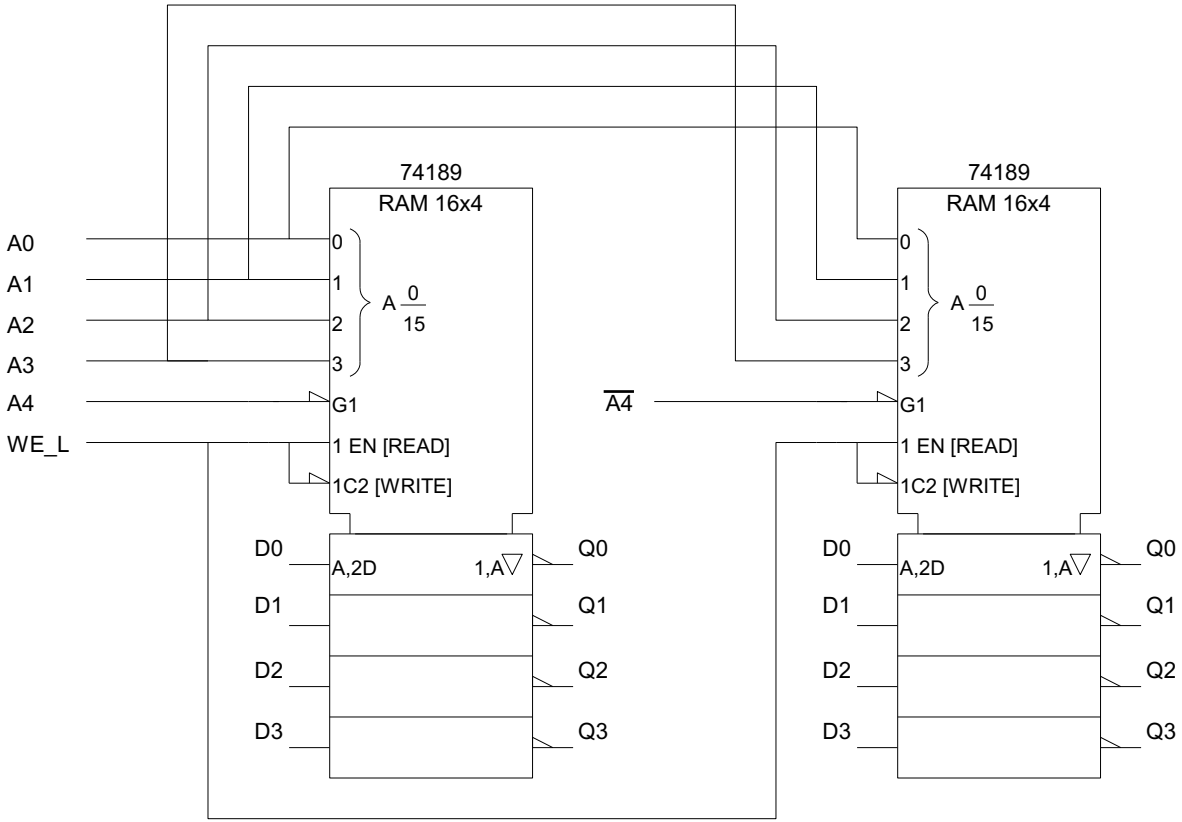
RAM - ASSOCIAÇÃO

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



RAM - ASSOCIAÇÃO

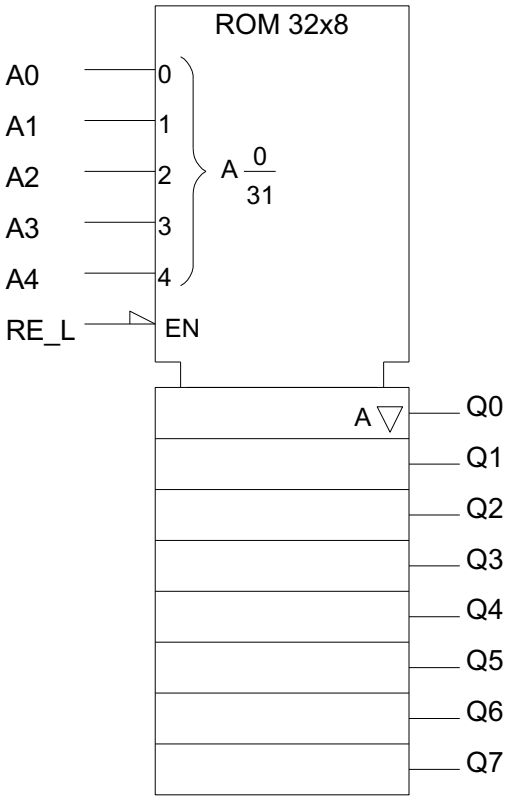
Aumento do Número de Palavras (de 16 para 32 palavras):



Notas: (1) Por simplicitade omitiram-se as ligações das entradas D e das saídas Q das duas memórias ao BUS de Dados;

(2) Para fazer associações maiores (4 memórias, 8 memórias, etc.), utilizam-se decoders.

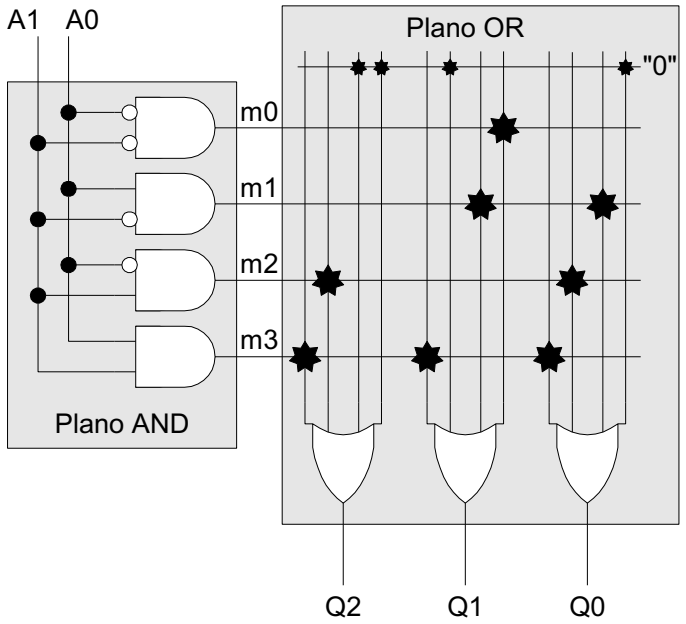
ROM - SIMBOLOGIA



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**

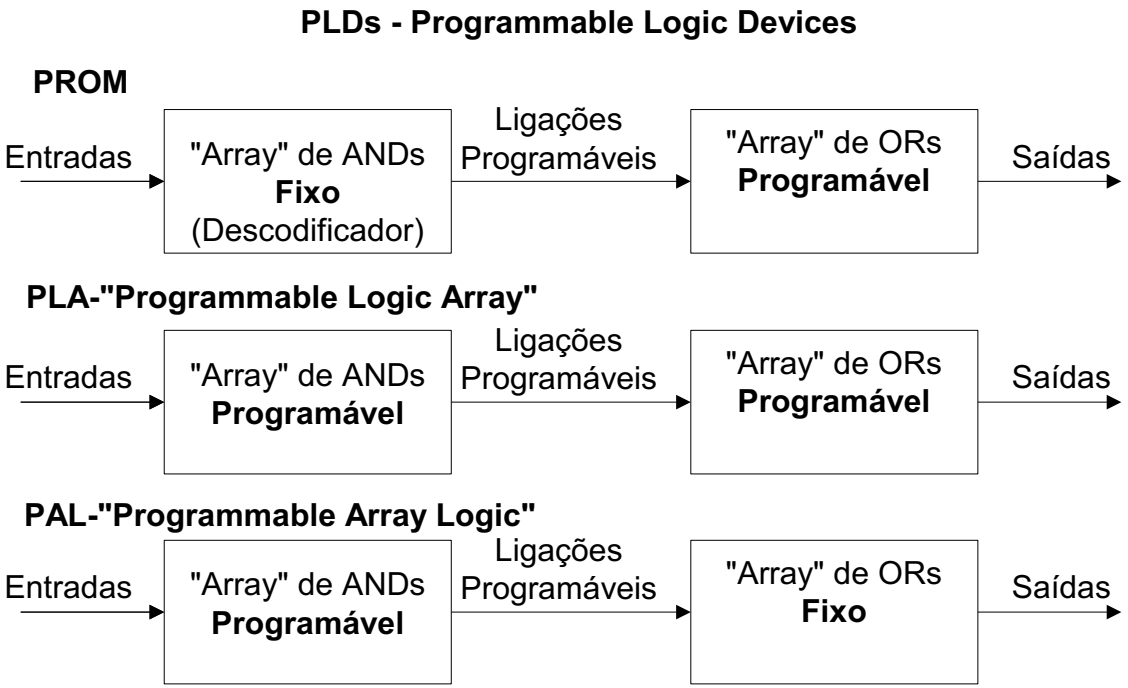


A programação da ROM é concretizada pela efetivação das ligações entre as saídas das ANDs e as entradas das ORs.

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

LÓGICA PROGRAMÁVEL

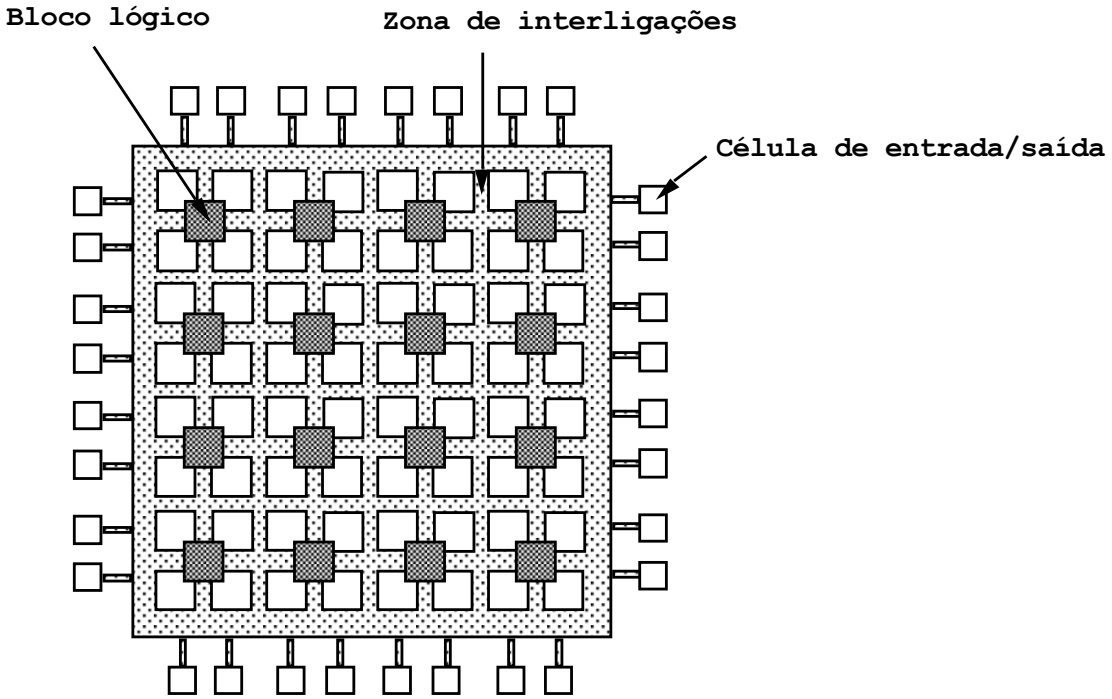
Nesta secção referem-se algumas estruturas**, derivadas da estrutura básica de uma ROM, cujo objectivo está centrado na sua programação** para implementação de funções combinatórias **e não na sua utilização como memória.**



FPGAs – "FIELD PROGRAMMABLE GATE-ARRAYS"

As FPGAs são circuitos programáveis que permitem realizar circuitos de complexidade equivalente a 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.



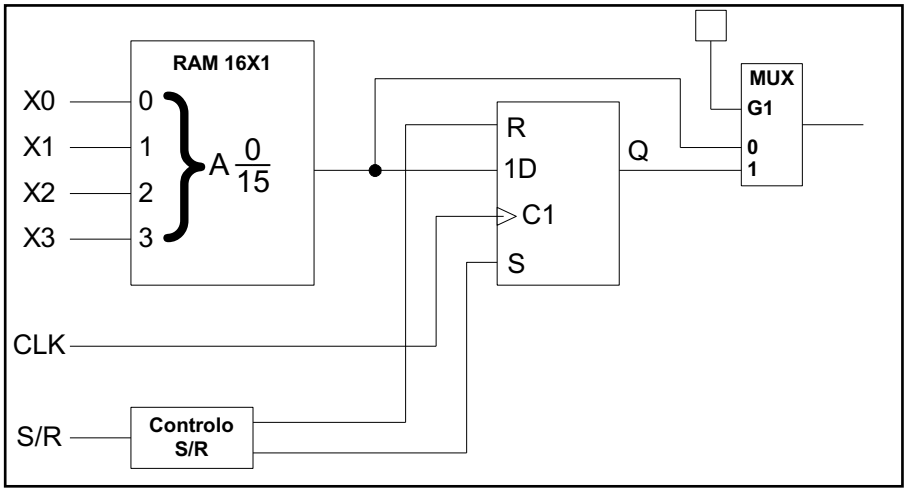
BLOCO LÓGICO BÁSICO DE UMA FPGA

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 16x1 bit) 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.

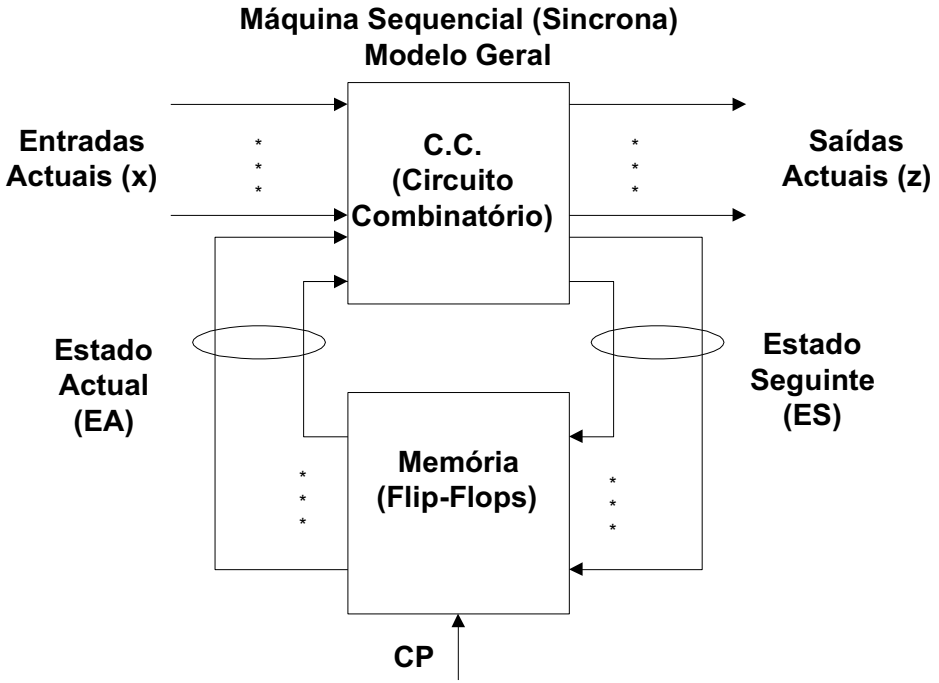


Bloco Básico – Modelo simplificado

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 de tornar os sistemas de controlo programáveis, ou seja, mais versáteis, e permitindo o desenvolvimento de sistemas funcionalmente mais complexos sem o aumento correspondente da complexidade do hardware.

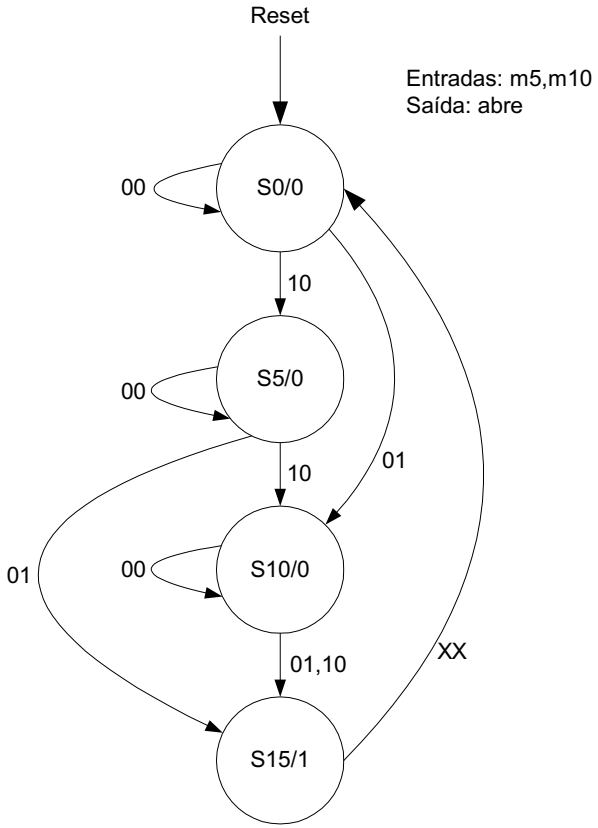
Qual o preço a pagar por esta solução em relação à implementação clássica?



MÁQUINAS DE ESTADO E MEMÓRIAS (cont.)

EXEMPLO: Máquina de Distribuição de Pastilhas

(conforme analisada no capítulo dos Circuitos Sequenciais Síncronos)



Estado Actual EA	$Q1_n$	$Q0_n$	Entradas m5	Entradas m10	Estado Seguinte ES	$Q1_{n+1}$	$Q0_{n+1}$	Saída EA ABRE
S0C	0	0	0	0	S0C	0	0	0
			0	1	S10C	1	0	0
			1	0	S5C	0	1	0
			1	1	X	X	X	0
S5C	0	1	0	0	S5C	0	1	0
			0	1	S15C	1	1	0
			1	0	S10C	1	0	0
			1	1	X	X	X	0
S10C	1	0	0	0	S10C	1	0	0
			0	1	S15C	1	1	0
			1	0	S15C	1	1	0
			1	1	X	X	X	0
S15C	1	1	0	0	S0C	0	0	1
			0	1	S0C	0	0	1
			1	0	S0C	0	0	1
			1	1	X	X	X	1

MÁQUINAS DE ESTADO E MEMÓRIAS (cont.)

EXEMPLO: Máquina de Distribuição de Pastilhas (cont.)

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.

Nas variáveis de endereço de Memória (A3 A2 A1 A0) representam-se as variáveis de estado (Q1 Q0) e as entradas (M5 M10). Assim, as linhas de endereço deverão conter todas as combinações possíveis de Estado Presente e entradas possíveis nesse estado.

A Palavra de Memória de cada endereço deverá conter a informação do Estado Seguinte, Q1 Q0 (n+1) e Saída, Z (n).

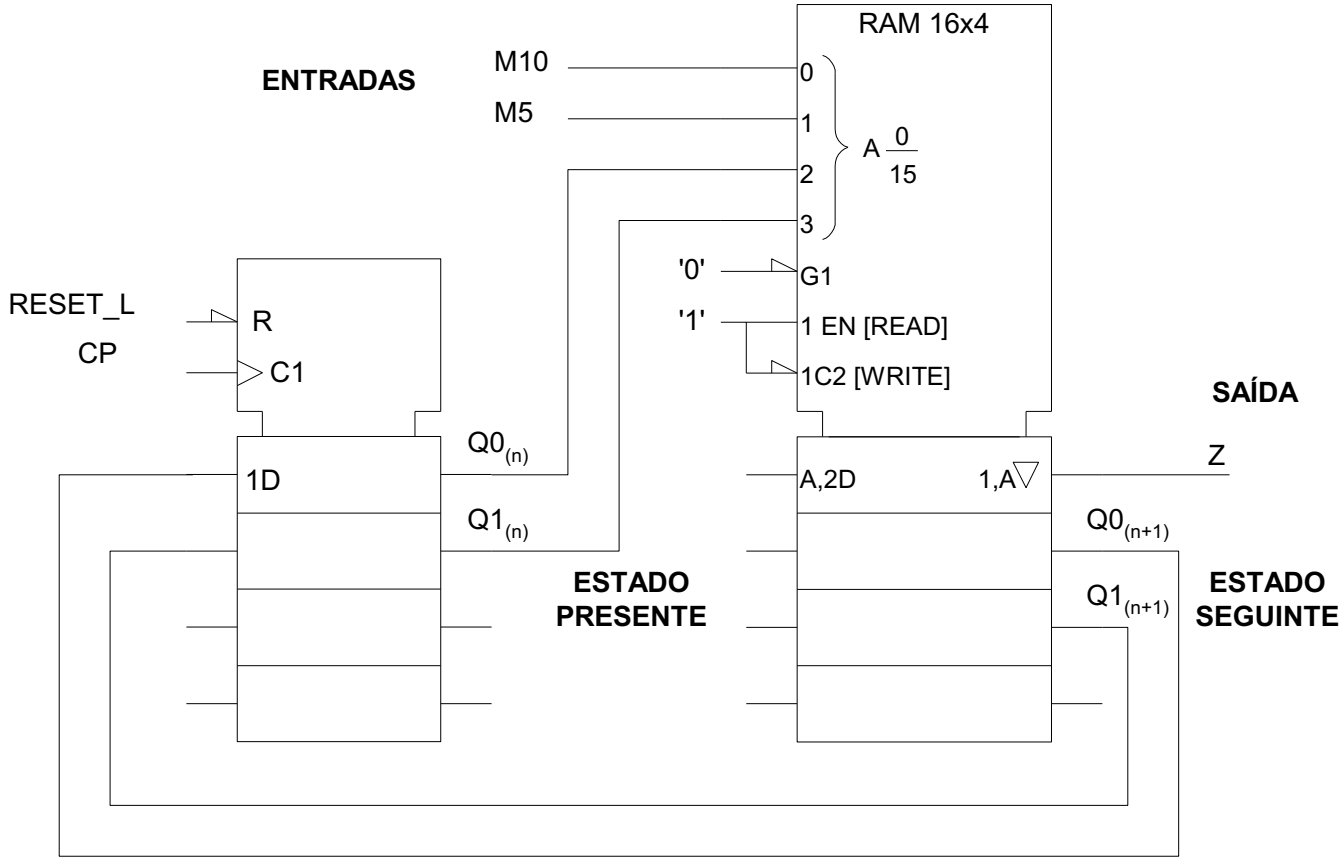
Nota: A cinzento indicam-se os endereços não utilizados e as saídas não consideradas (bit mais significativo).

Tabela de Memória

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

MÁQUINAS DE ESTADO E MEMÓRIAS (cont.)

EXEMPLO: Máquina de Distribuição de Pastilhas (cont.)



Nota: Existem outras formas alternativas para este tipo de implementação "microprogramada", mais complexas, mas mais eficientes.

BIBLIOGRAFIA

[1] Arroz,G., Monteiro,J.C., Oliveira,A., "Arquitectura de Computadores, dos Sistemas Digitais aos Microprocessadores", Secções 6.8 e 7.5.3, 2ª Edição, 2009

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