

MEMÓRIAS

Sistemas Digitais II

Prof. Marcelo Wendling

Set/10

Definição

São blocos que armazenam informações codificadas digitalmente – números, letras, caracteres quaisquer, comandos de operações, endereços ou qualquer tipo de dado. Dividem-se basicamente em dois grupos: **Memórias de escrita e leitura** e **Memórias apenas de leitura** (IDOETA).



Bits, Nibbles e Bytes

As memórias armazenam dados em unidades chamadas de **bits**, que é a menor unidade de dados binários.

Um conjunto de 4 bits forma **1 nibble** e o conjunto de 8 bits forma **1 byte**.

Endereço de Memória

A localização de uma unidade de dado num arranjo de memória é denominada **endereço**. (FLOYD)

Ou seja, um conjunto de bits chamados de **bits de endereço** indexam uma posição única de memória que contém um determinado **dado** armazenado.

Classificação

São classificadas basicamente por:

- 1) Acesso
- 2) Volatilidade
- 3) Troca de dados
- 4) Tipo de Armazenamento

1) Acesso

Os locais onde as informações estão armazenadas são denominadas **localidade de memória**, onde cada localidade possui um endereço que permite o seu acesso.
(Conjunto de bits)

Podemos ter acesso a uma dada localidade de duas maneiras:

1.1) Acesso Seqüencial

1.2) Acesso Aleatório

1.1) Acesso Seqüencial

Para acessar o endereço desejado passa por todos os outros endereços (localidades) intermediárias.

O tempo de acesso depende do lugar onde a informação está armazenada.

1.2) Acesso Aleatório

Acessam o endereço desejado diretamente sem passar pelas localidades intermediárias.

Principais memórias: **RAM** (Random-Access-Memory).

Possuem tempo de acesso pequeno e igual para qualquer localidade da memória.

2) Volatilidade

Podem ser **Voláteis** ou **Não-Voláteis**:

2.1) Voláteis

Perdem os dados armazenados quando é cortada a **alimentação**. Ex: RAM

Feitas geralmente a partir de elementos semicondutores, como Flip-Flops por exemplo.

2.2) Não-voláteis

Não perdem os dados mesmo sem alimentação. Ex: ROM, PROM, EPROM.

3) Troca de dados

Podem ser de **Escrita / Leitura** ou somente **Leitura**.

3.1) Escrita / Leitura

Permitem acesso a uma localidade qualquer para armazenar a informação desejada ou para a leitura do dado na localidade.

Ex: RAM

3) Troca de dados

3.2) Leitura

Como o próprio nome diz, são aquelas em que a informação é fixa, só podendo efetuar-se a leitura.

São conhecidas também como **ROM** (Read-Only Memory).

4) Tipo de Armazenamento

Classificam-se em **Estáticas** ou **Dinâmicas**.

4.1) Armazenamento Estático

São aquelas em que uma vez inserido o dado, este lá permanece.

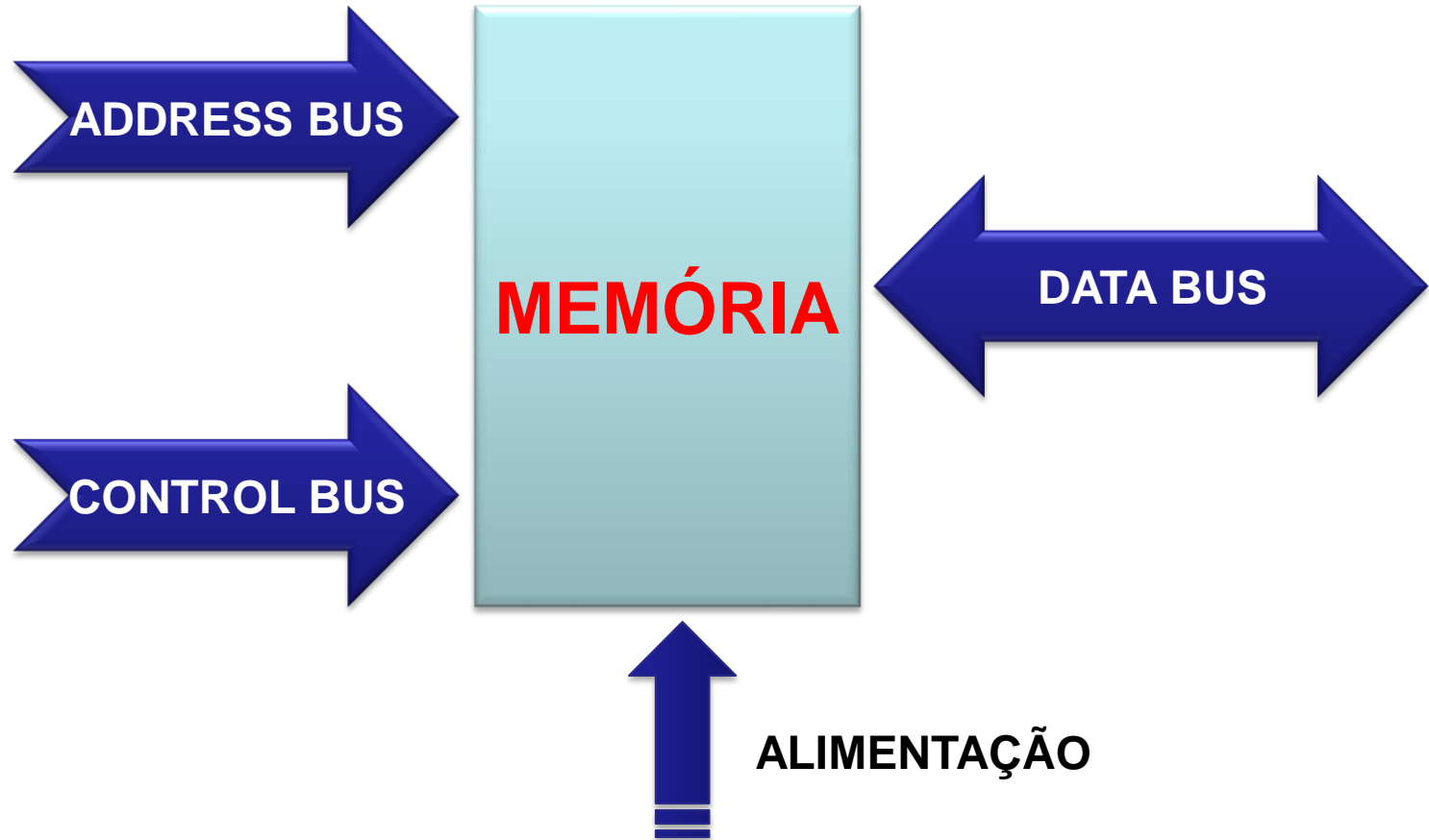
4.2) Armazenamento Dinâmico

São aquelas em que necessitamos inserir a informação de tempos em tempos.

Classificação - Resumo



Estrutura Geral



Organização

No que se refere à quantidade de dados armazenados, utiliza-se a notação

$N \times m$

Onde **N** é a quantidade de localidades da memória e **m** é o número de bits que é armazenado por localidade.

Ex: 32x8
128x8
64kx8

O número de localidades é sempre múltiplo de 2^n , fato derivado da possibilidade total de endereçamento por um determinado número de fios, que também é múltiplo de 2.

Algumas considerações

- 1) A designação k (quilo) significa 2^{10} (1024 bits).
A designação M (mega) significa 2^{20} (1048576 bits).

Portanto quando utilizamos a denominação 64kx8, dizemos que a memória possui 64 x 1024, portanto 65536, localidades com 1byte (8 bits) cada.

- 2) **Capacidade de Memória**, que significa o número total de bits que podem ser armazenados em uma memória. É determinada pelo produto $N \times m$.

Algumas considerações

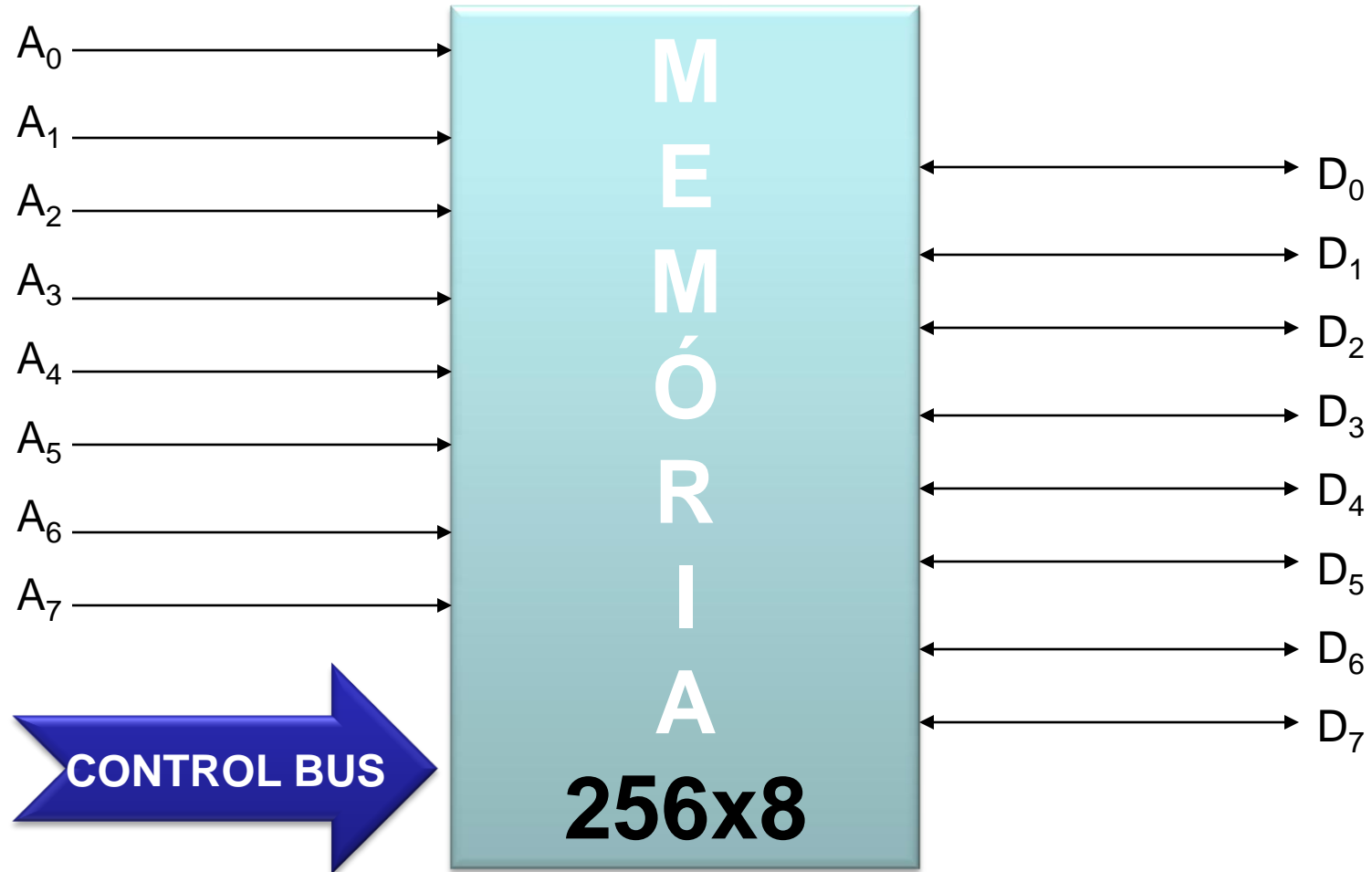
3) Palavra de endereço é o conjunto de níveis lógicos ou bits necessários para o endereçamento de uma determinada localidade de memória.

Mapeamento de memória

Consideremos uma memória genérica 256x8, que possui:

- 256 localidades
- 8 bits para endereçamento ($2^8 = 256$)
identificados de A_7 até A_0 , sendo o endereço da localidade inicial é 00_h e da localidade final é FF_h
- 1byte de dados por localidade, de D_7 até D_0

Mapeamento de memória



Mapeamento de memória

Address Bus (binário)	Address Bus (hexa)	Localidade	Conteúdo
$A_7 A_6 A_5 A_4 A_3 A_2 A_1 A_0$			
0 0 0 0 0 0 0 0	0 0 _h	L ₀	I ₀
0 0 0 0 0 0 0 1	0 1 _h	L ₁	I ₁
0 0 0 0 0 0 1 0	0 2 _h	L ₂	I ₂
0 0 0 0 0 0 1 1	0 3 _h	L ₃	I ₃
0 0 0 0 0 1 0 0	0 4 _h	L ₄	I ₄
...

Mapeamento de memória

Address Bus (binário)	Address Bus (hexa)	Localidade	Conteúdo
...
1 0 1 0 0 1 1 1	A 7 _h	L ₁₆₇	I ₁₆₇
1 0 1 0 1 0 0 0	A 8 _h	L ₁₆₈	I ₁₆₈
...
1 1 1 1 1 1 0 1	F D _h	L ₂₅₃	I ₂₅₃
1 1 1 1 1 1 1 0	F E _h	L ₂₅₄	I ₂₅₄
1 1 1 1 1 1 1 1	F F _h	L ₂₅₅	I ₂₅₅

Determinando o Address Bus

O número de bits que compõe o **barramento de endereços** é determinado a partir do número de localidades que a memória possui:

Ex: Uma memória 256x8 possui 256 localidades de memória, portanto o barramento de dados deve seguir a expressão abaixo:



Memória ROM

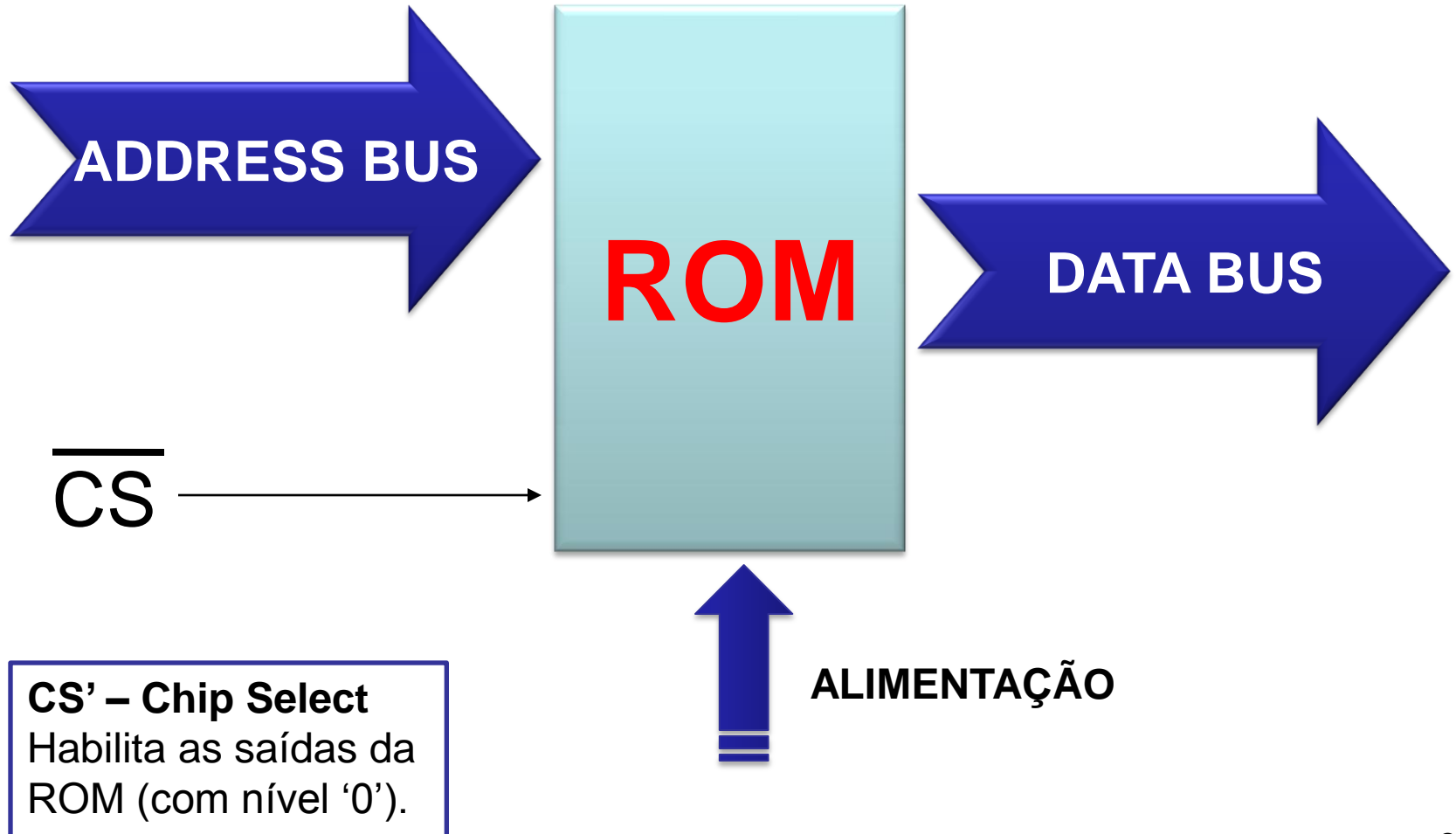
Características:

Permitem somente a **Leitura** dos dados gravados nela.

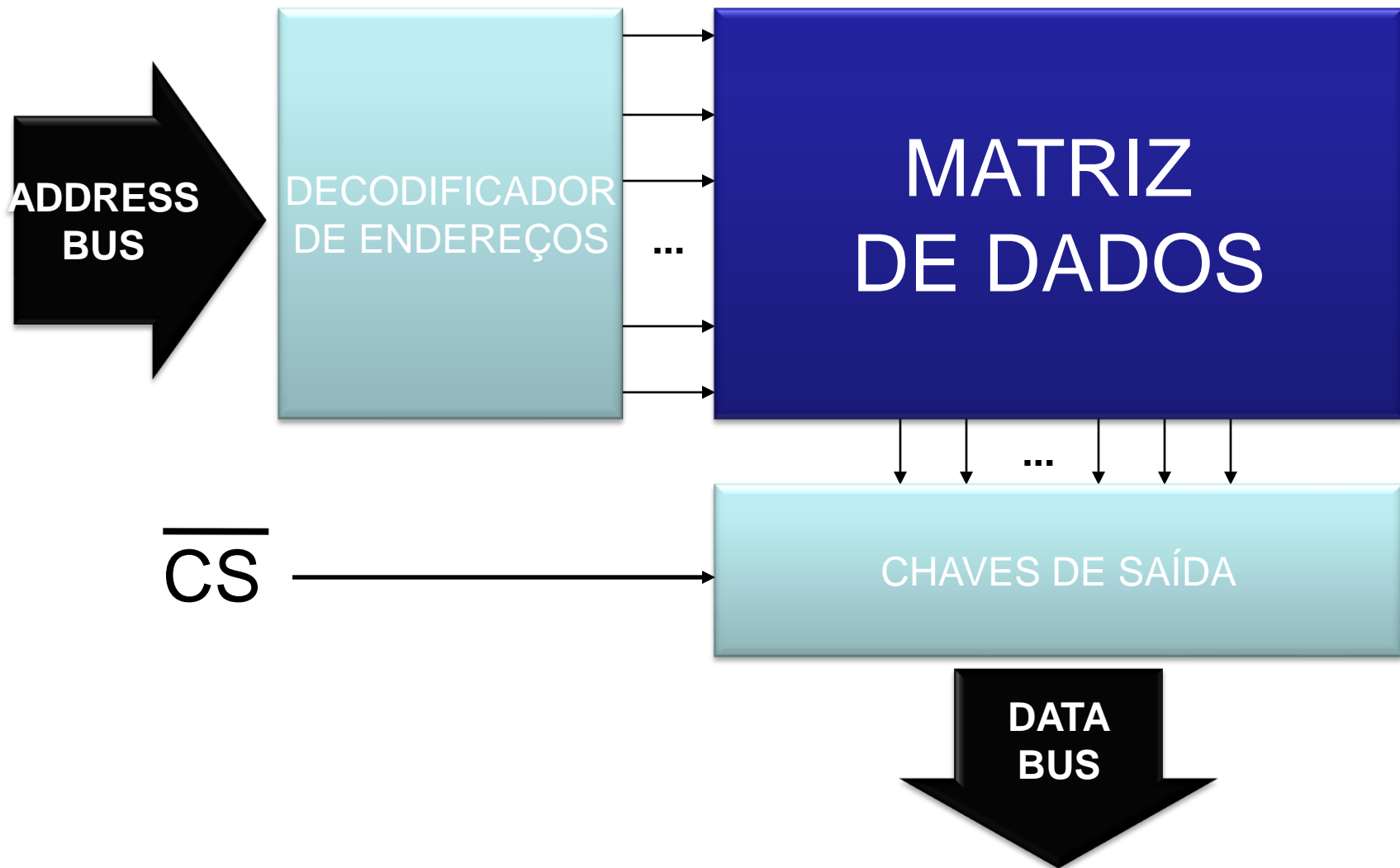
Possuem **acesso aleatório** e memorização **não-volátil**.

Podem ser consideradas como circuitos combinacionais, pois apresentam as saídas de dados em função das combinações entre as variáveis de entrada (endereçamento).

Bloco representativo - ROM



Arquitetura interna - ROM

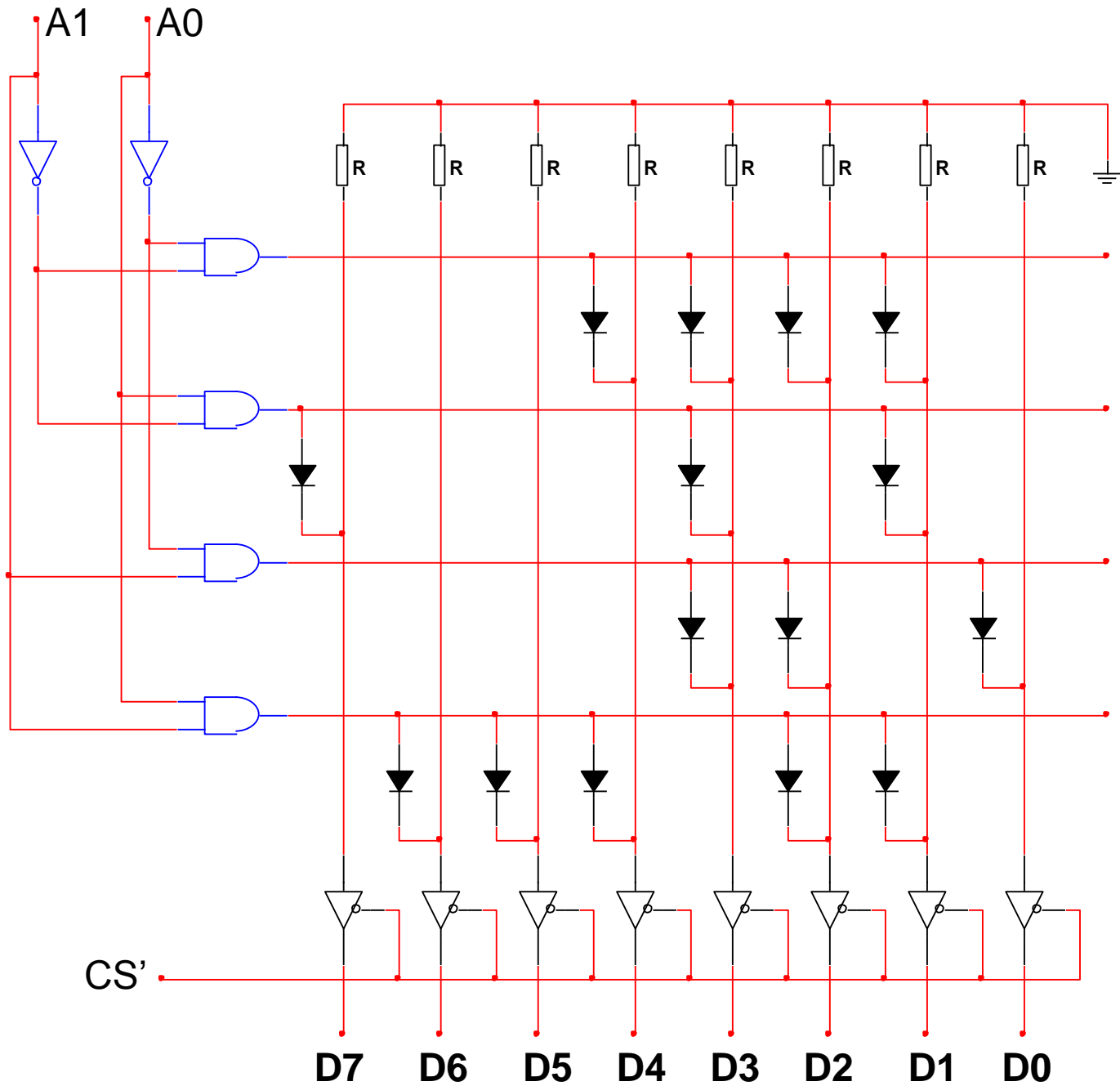


Exemplo

Utilizamos como exemplo uma memória ROM 4x8 com os seguintes dados gravados:

ADDRESS	DATA	
	HEX	BIN
0 0	1 E	0 0 0 1 1 1 1 0
0 1	8 A	1 0 0 0 1 0 1 0
1 0	0 D	0 0 0 0 1 1 0 1
1 1	7 6	0 1 1 1 0 1 1 0

Circuito equivalente



Considerações finais - ROM

São programadas através de gabaritos fotográficos e fabricadas sob encomendas de grande quantidade para aplicações específicas.

Podem ser programadas apenas uma vez.

Memória PROM

(Programmable Read-Only Memory)

Permitem o armazenamento dos dados pelo próprio usuário, porém feito de modo definitivo.

Após a programação, a memória se torna uma ROM.

Memória PROM

O princípio de programação é o rompimento das pequenas ligações dos semicondutores internos, onde se quer armazenar os dados.

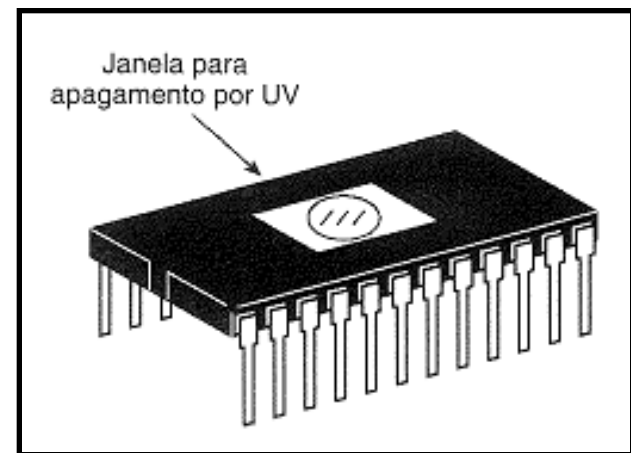
Memória do tipo não-volátil, acesso aleatório e de apenas de leitura.

Memória EPROM

(Erasable Programmable Read-Only Mem.)

Podem ser apagadas mediante a exposição à luz ultravioleta, por uma janela existente em seu encapsulamento e, ainda, ser reprogramadas.

São conhecidas também como UVPRM (Ultraviolet PROM)



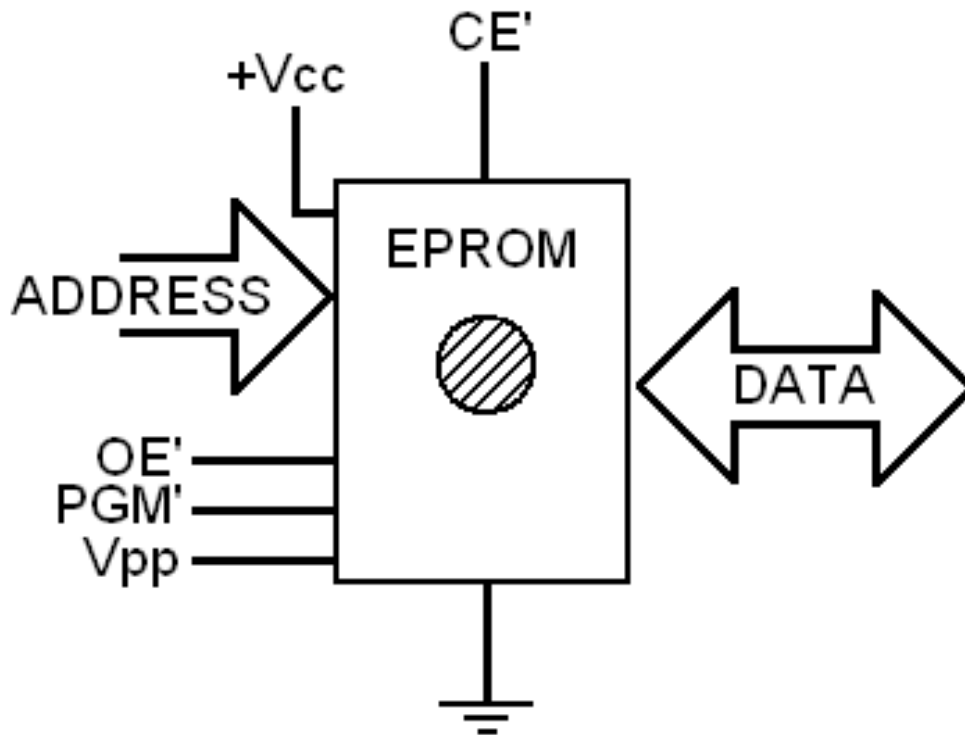
Memória EPROM

Após a programação se tornam memórias PROM.

Os dados são apagados de maneira universal, portanto, após o reset, toda a memória deve ser reprogramada.

Memória EPROM

Diagrama de blocos básico:



ADDRESS: barramento de endereço
DATA: barramento de dados
CE': chip enable
OE': output enable
PGM': habilitação da programação
Vpp: tensão de programação

Memória EEPROM

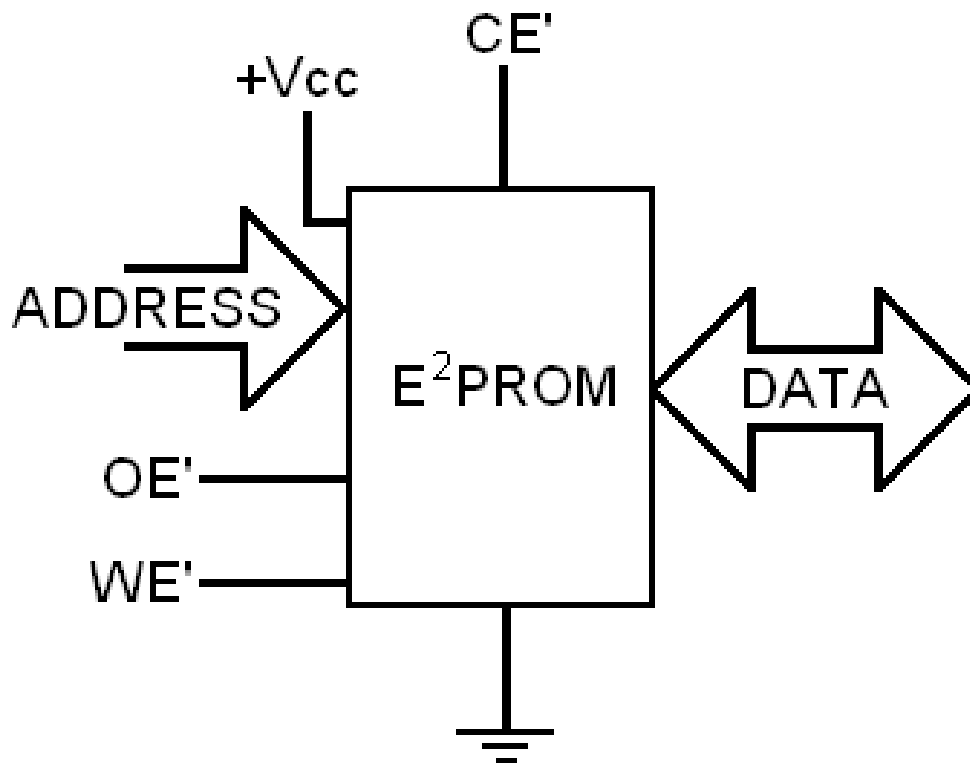
EEPROM ou E²PROM

(Electrically Erasable Programmable Read-Only Memory).

São um avanço das memórias EPROM, pois podem ser apagadas eletricamente e ainda isoladamente por palavra de dados, sem a necessidade de programação total.

Memória EEPROM

Diagrama de blocos básico:



ADDRESS: barramento de endereço

DATA: barramento de dados

CE': chip enable

OE': **output** enable

WE': **write** enable

Memória RAM

Memórias de escrita e leitura.

Voláteis.

Acesso aleatório.

Possuem tempo de acesso bastante reduzido.

Memória RAM

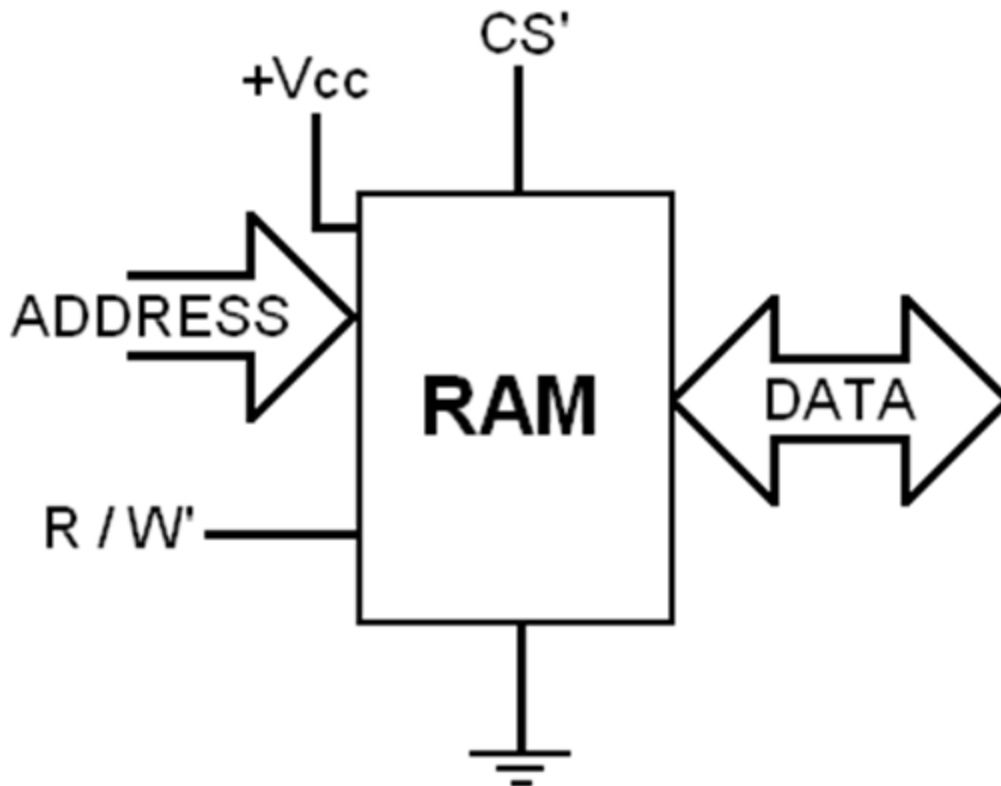
São encontradas de dois tipos:

SRAM (Static RAM): arquitetura básica flip-flops.

DRAM (Dynamic RAM): arquitetura interna microprocessada.

Memória RAM

Diagrama de blocos básico:



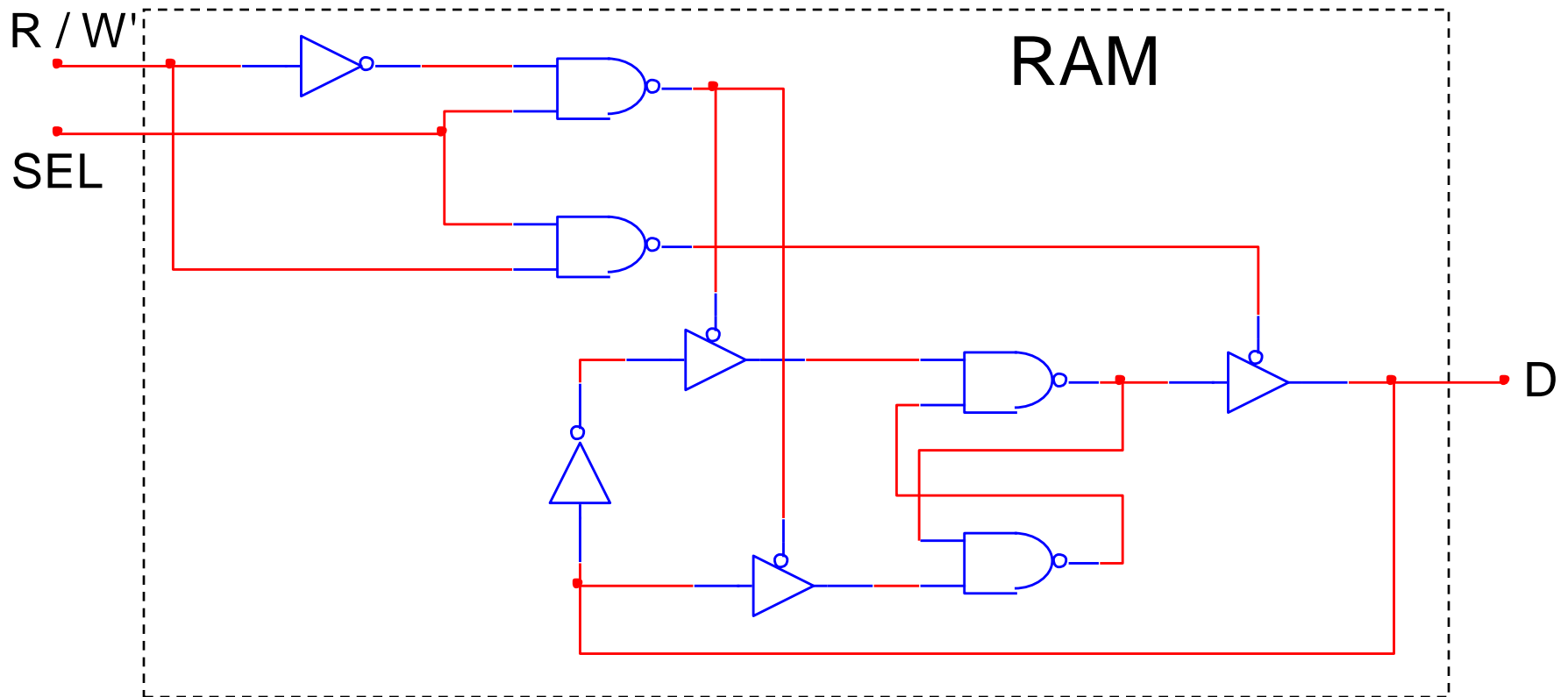
ADDRESS: barramento de endereço

DATA: barramento de dados

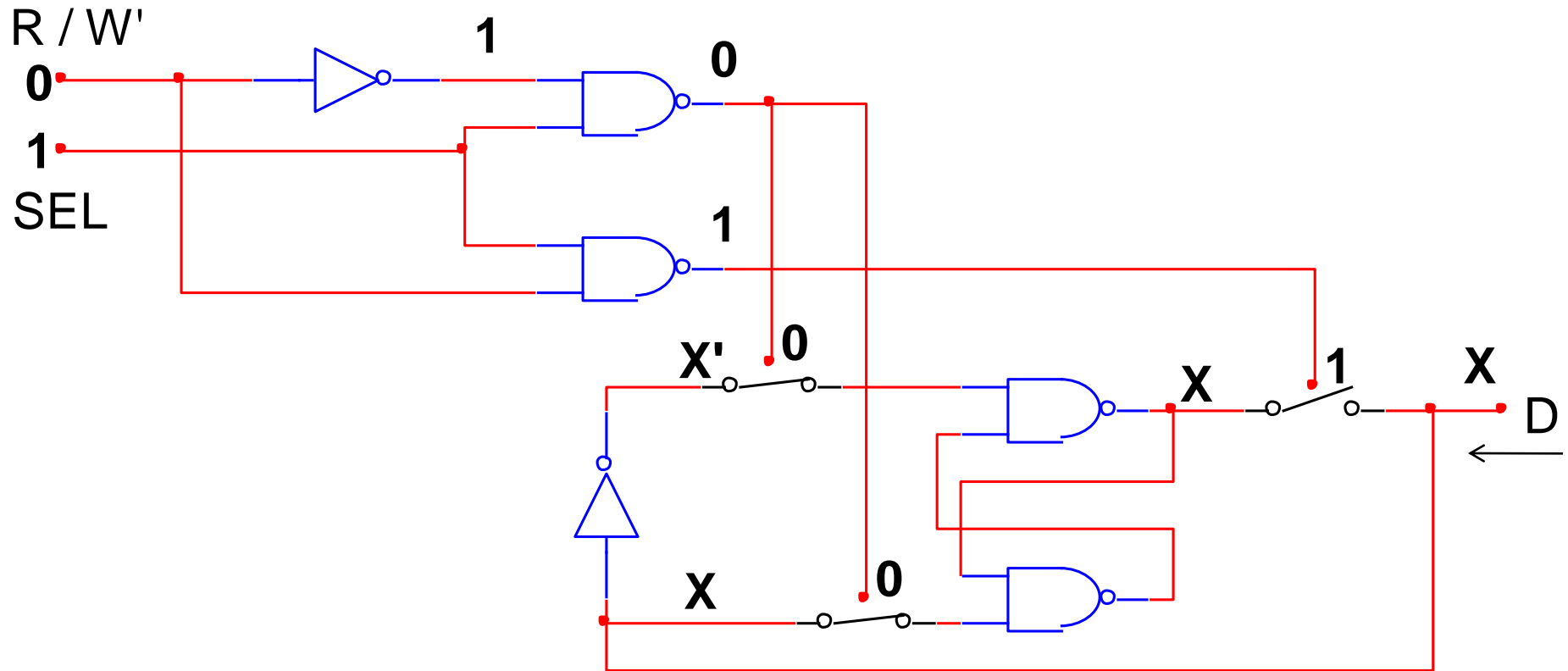
CE': chip enable

R / W': Read – 1; Write – 0;

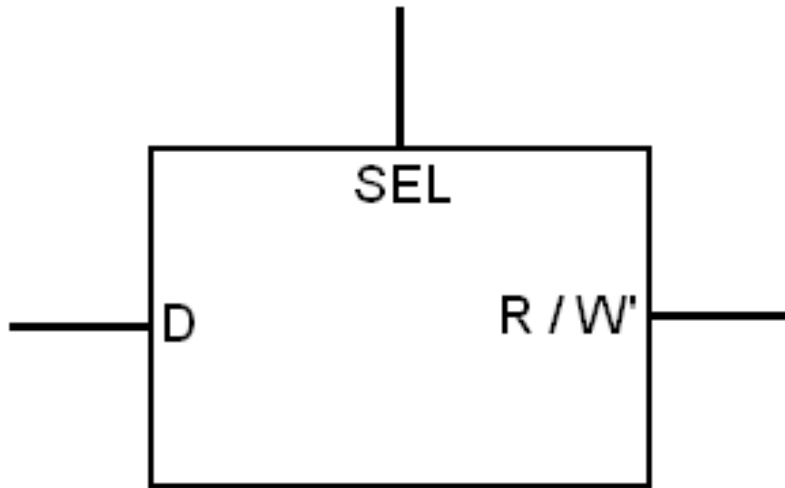
RAM – Célula Básica



RAM – Funcionamento (R/W'=0)

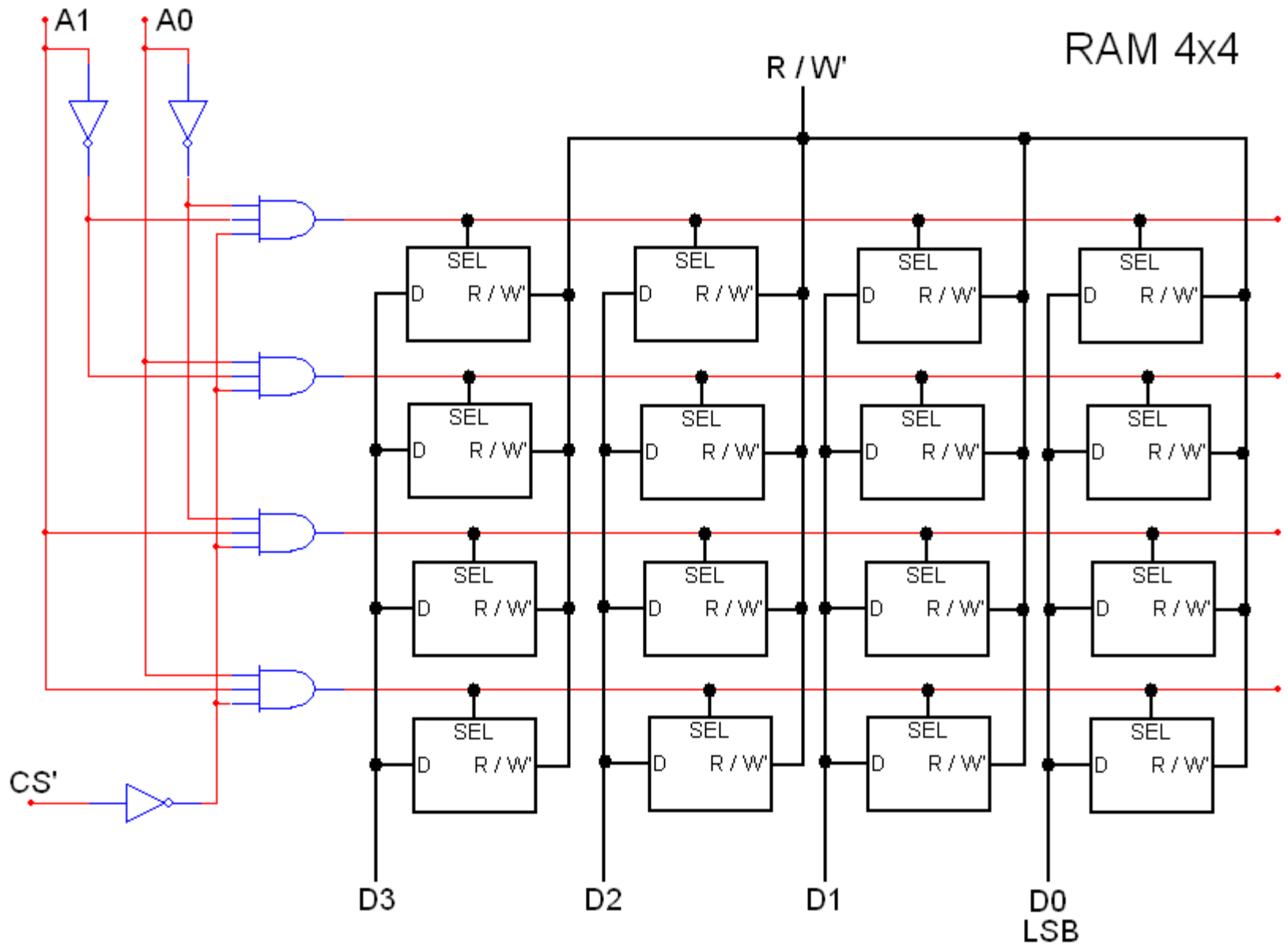


RAM – Célula Genérica



SEL	R / W'	D
0	X	Z
1	0	ESCRITA
	1	LEITURA

Arquitetura Interna



Memória Flash

Diferentemente da memória RAM e também das SRAM, a memória Flash permite armazenar dados por longos períodos, sem precisar de alimentação elétrica. Graças a isso, a memória Flash se tornou rapidamente a tecnologia dominante em cartões de memória, pendrives, memória de armazenamento em câmeras, celulares e palmtops e assim por diante.

Memória Flash

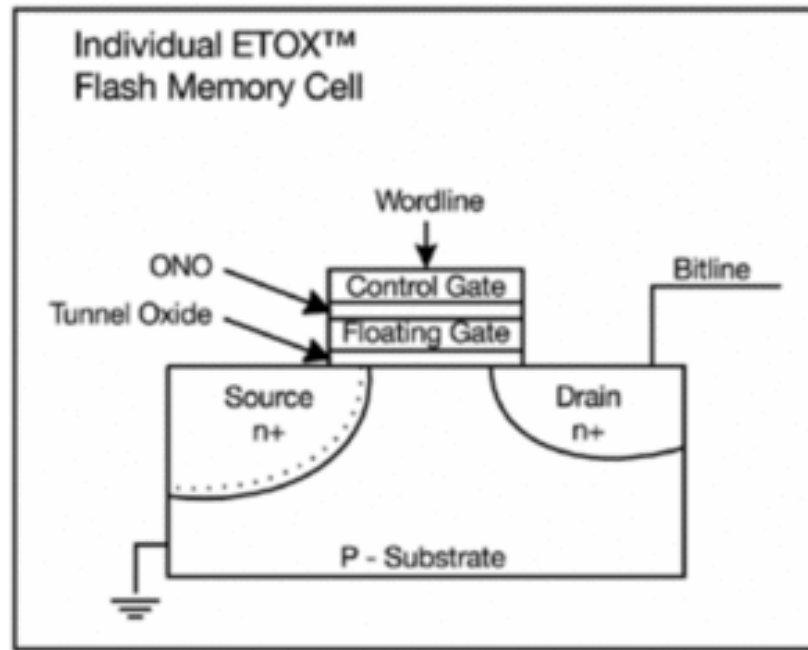
Constituem um tipo específico de memória E²PROM, em que os dados são gravados ou apagados em blocos de muitos bytes. Essa característica confere às flash mais rapidez que as E²PROM tradicionais.

Existem basicamente dois tipos de memória flash - NAND e NOR -, que diferem no método de leitura e gravação de dados. A primeira permite apenas acesso seqüencial aos dados. Nela, o endereçamento é mais simples e a entrada, mais rápida. A NOR, geralmente empregada nos chips de BIOS dos microcomputadores e nos telefones celulares, possibilita acesso aleatório porém gravação mais lenta.

Memória Flash

Num dispositivo flash, há milhares de células distribuídas numa matriz de linhas e colunas acessíveis isoladamente ou em blocos. Cada uma consiste em dois transistores (porta de controle e porta flutuante) separados por uma camada fina de óxido. A porta de controle é ligada à linha de bytes da matriz. A porta flutuante se conecta à linha de bits (por meio de um terminal chamado dreno) e à fonte de elétrons (terminal fonte ou terra).

Memória Flash



Mesmo que o dispositivo flash esteja completamente desligado, os elétrons permanecem "presos" na porta flutuante em cada célula.

Para acessar a informação (bit) presente na célula, é preciso aplicar uma tensão específica na porta de controle. A corrente resultante vai depender da quantidade e do arranjo dos elétrons na porta flutuante, caracterizando o dado armazenado.

Expansão de memória

É a necessidade de se expandir pelo menos uma das características de uma memória, isso quando a memória utilizada não atende às necessidade de uso.

Para isso faz-se **associação** específicas de suas características.

Expansão de memória

Quando falamos de expansão de capacidade, utilizamos dois tipos de expansão:

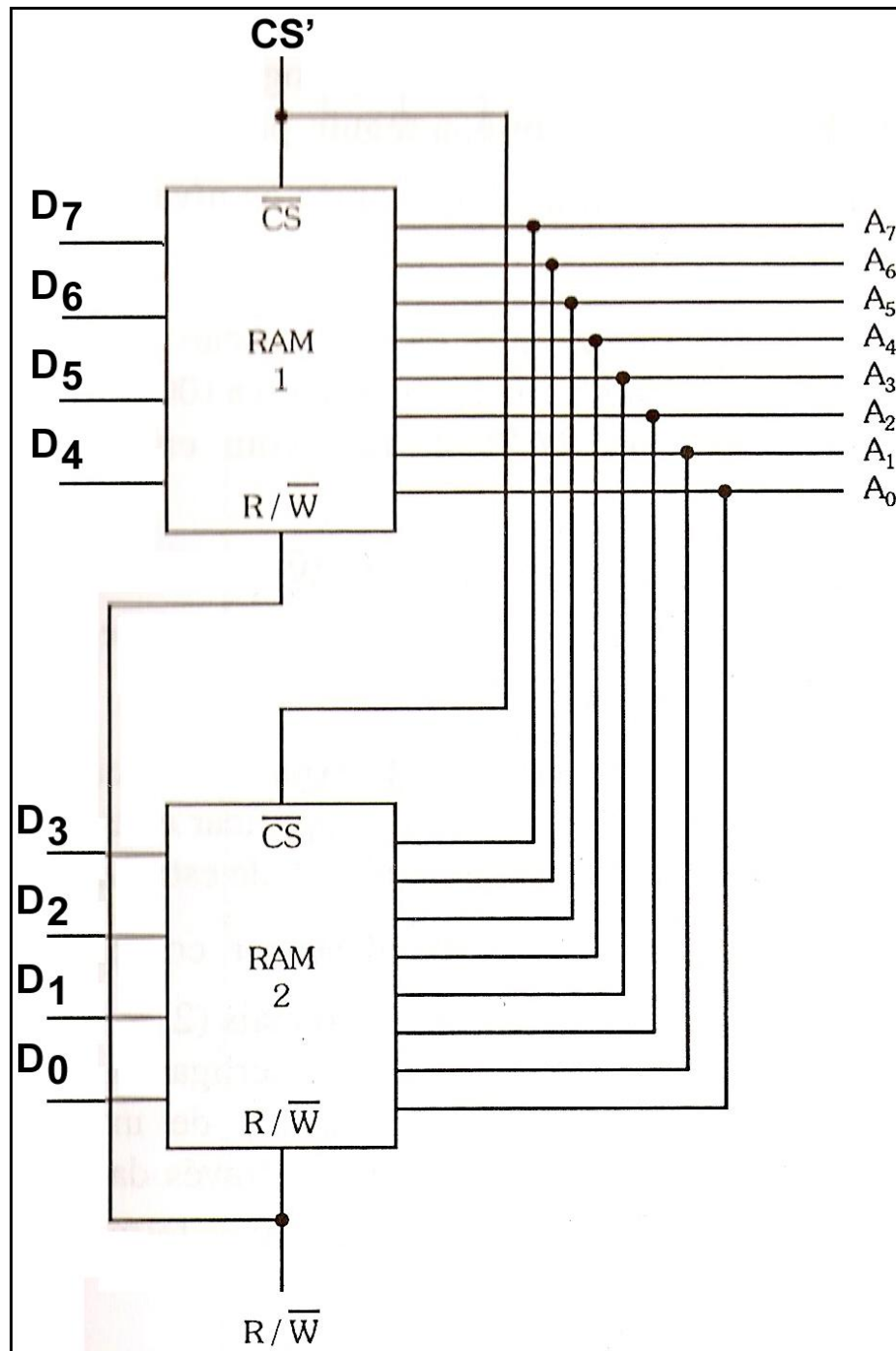
- **Expansão da capacidade da palavra de dados.**
- **Expansão da capacidade de memória (ou localidades de memória).**

Palavra de dados

Este tipo de expansão é utilizado quando o **número de bits de saída** necessitado por um circuito é maior que o fornecido pela memória.

Como solução **associa-se a saída** de duas ou mais memórias, fazendo com que a quantidade de bits de saída seja maior.

Expansão de palavras de dados



Os terminais CS' e R / W' são interligados, pois essas operações são comuns aos dois blocos.

A palavra de dados é composta pela associação dos dois blocos, fazendo com que duas memórias com 4 bits de saída se associem formando uma memória com 8 bits de saída, como exemplifica a figura ao lado.

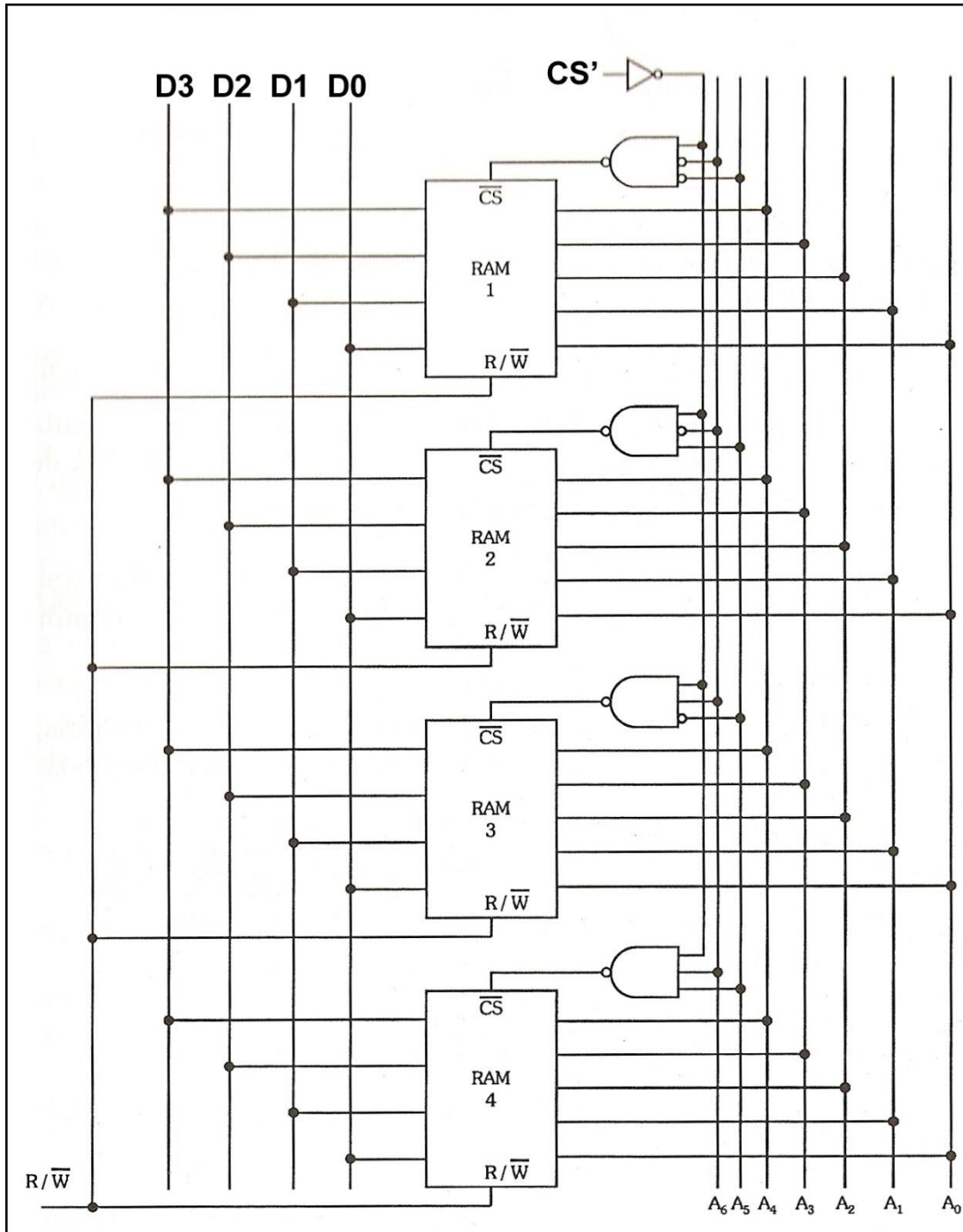
No caso ao lado, duas memórias RAM 256x4, se associaram em uma RAM 256x8.

Capacidade de memória

Este tipo de expansão é utilizado quando o **número de localidade de memória** necessários é maior que o fornecido pela memória.

Como solução **associa-se os terminais de endereço** de duas ou mais memórias, fazendo com que a quantidade de bits de saída seja maior.

Expansão de capacidade de memória



Os terminais de saída de dados são curto-circuitados, pois as saídas de cada bloco são comuns ao mesmo barramento de dados.

O que se implementa nesse tipo de associação é o endereçamento pelo terminal CS', que passa a ser um dos terminais de endereço, possibilitando o acesso a cada bloco separadamente.

Expansão de capacidade de memória

A capacidade de endereçamento foi expandida, ou seja, pode-se acessar mais localidades de memória

No exemplo anterior possuímos 4 memórias RAM 32x4 foram associadas em uma RAM 128x4.

Abaixo possuímos uma tabela com a estrutura de endereçamento dessa memória:

MEM.	A6	A5	A4	A3	A2	A1	A0	HEX
RAM 1	0	0	0	0	0	0	0	00
	0	0	1	1	1	1	1	1F
RAM 2	0	1	0	0	0	0	0	20
	0	1	1	1	1	1	1	3F
RAM 3	1	0	0	0	0	0	0	40
	1	0	1	1	1	1	1	5F
RAM 4	1	1	0	0	0	0	0	60
	1	1	1	1	1	1	1	7F

Expansão de memória

Uma outra possibilidade de expansão seria a combinação dos dois métodos anteriores, ou seja, a ampliação da palavra de dados e o número de localidades.

Ex: Utilizando blocos de memória RAM 128x4, forme uma de 256x8.

Expansão de memória

Esta expansão consiste na ampliação tanto no número de localidade quanto também da palavra de dados.

Para executá-la, vamos determinar inicialmente o número de terminais de endereço da nova memória:

$$N_{\text{LOCALIDADES}} = 2^n \text{ onde } n \text{ é o número de terminais de endereço.}$$
$$256 = 2^8 \text{ portanto são } \mathbf{8 \text{ bits}} \text{ de endereçamento.}$$

Para o novo **endereçamento**, vamos utilizar os 7 terminais de cada bloco ($128 = 2^7$) e mais um terminal ligado a um circuito seletivo.

Para compor a nova **palavra de dados**, associa-se um outro sistema semelhante para a duplicação.

Expansão de memória

