



IC-UNICAMP

MC 613

IC/Unicamp

Prof Guido Araújo

Prof Mario Côrtes

Prof Sandro Rigo

Memória

Tópicos

- Tipos de memórias
- Organização
- Decodificação de endereço
- Memórias em VHDL
- Usando memórias na DE1

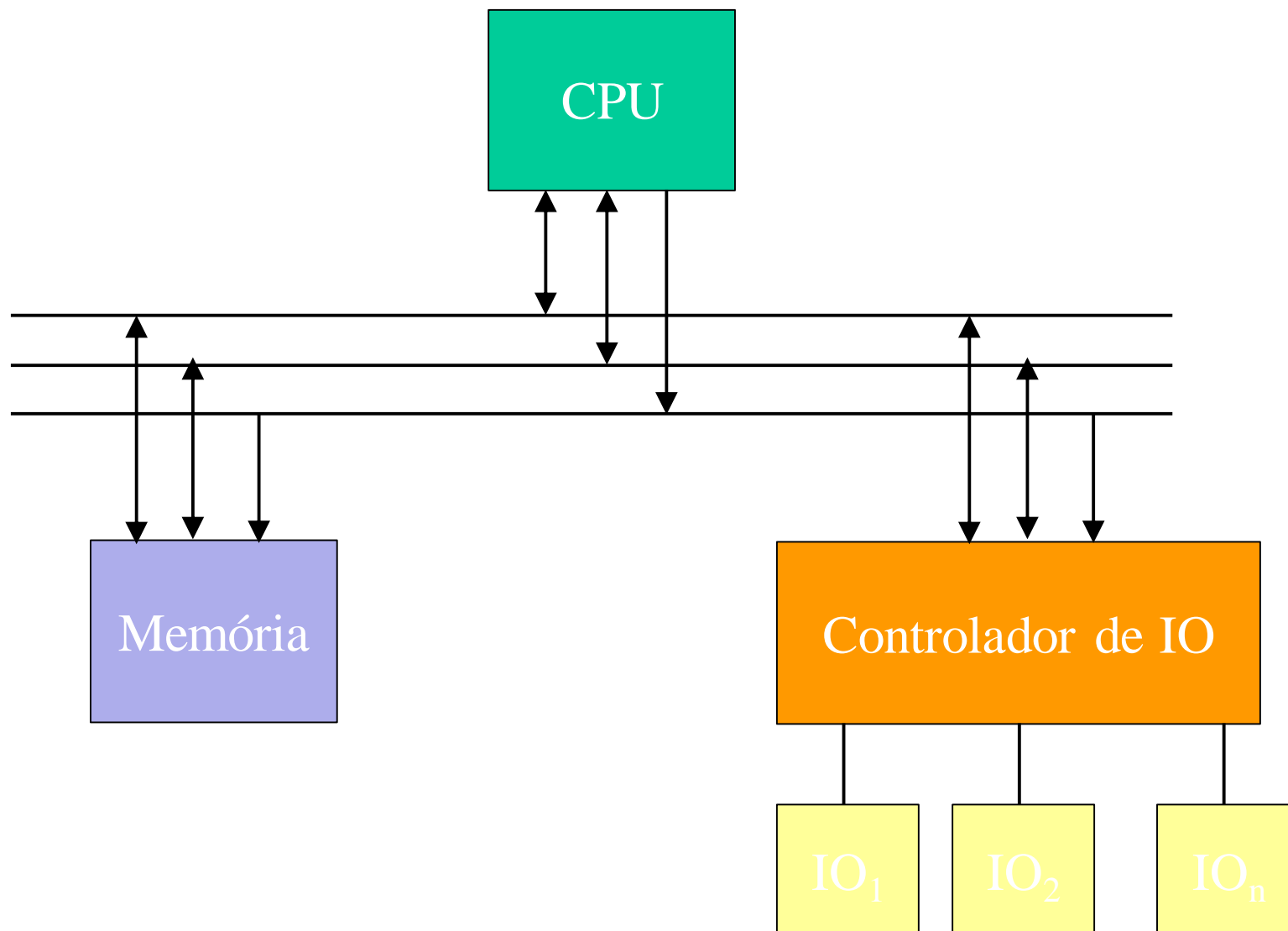
Introdução



IC-UNICAMP

- Memória: dispositivos capazes de armazenar eficientemente grande quantidade de dados
- Organização: semelhante a uma tabela de dados
 - n linhas, com m bits cada
- Operações: leitura e escrita

Sistema de memória: uso típico



Organização e dimensões

Conceitualmente: uma tabela com linhas de dados
Organizadas como uma matriz (array) de duas
dimensões de células de bits

Cada célula armazena um bit

No exemplo

16 linhas de dados
palavras de 8 bits

16
linhas

8 bits

0	0	1	0	1	1	0	0
1	0	1	1	0	0	1	0
1	0	1	1	0	0	0	1
1	1	0	0	1	0	1	0
0	0	1	0	1	1	0	0
1	0	1	1	0	0	1	0
1	0	1	1	0	0	0	1
1	1	0	0	1	0	1	0
0	0	1	0	1	1	0	0
1	0	1	1	0	0	1	0
1	0	1	1	0	0	0	1
1	1	0	0	1	0	1	0
0	0	1	0	1	1	0	0
1	0	1	1	0	0	1	0
1	0	1	1	0	0	0	1
1	1	0	0	1	0	1	0

Organização e dimensões

nº bits

Largura (width):

nº de colunas no array
 = nº de bits na linha de dados
 = word size

Profundidade (Depth):

número de linhas do array

**nº
linhas**

Tamanho do array

largura x profundidade
 = (nº de linhas) * (bits/linha)
 = (nº de linhas) * (word size)

0	0	1	0	1	1	0	0
1	0	1	1	0	0	1	0
1	0	1	1	0	0	0	1
1	1	0	0	1	0	1	0
0	0	1	0	1	1	0	0
1	0	1	1	0	0	1	0
1	0	1	1	0	0	0	1
1	1	0	0	1	0	1	0
0	0	1	0	1	1	0	0
1	0	1	1	0	0	1	0
1	0	1	1	0	0	0	1
1	1	0	0	1	0	1	0
0	0	1	0	1	1	0	0
1	0	1	1	0	0	1	0
1	0	1	1	0	0	0	1
1	1	0	0	1	0	1	0

Organização, entradas, saídas

Entradas

Endereço: n bits selecionam 2^n linhas

Dados (bidirecional): m bits de dados de escrita ou leitura

Controle: WR, RD, OutputEnable

Tamanho da memória

$2^n * m$ bits

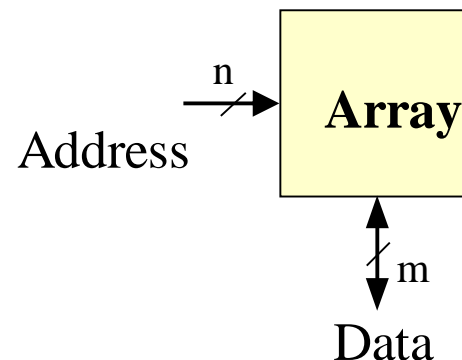
Exemplo: se $m = 8$ (1 Byte) e $n = 10$

1024 linhas (1K) e 8 colunas

tamanho da memória = 1 KB

ou 1K x 1B

ou 8 Kb



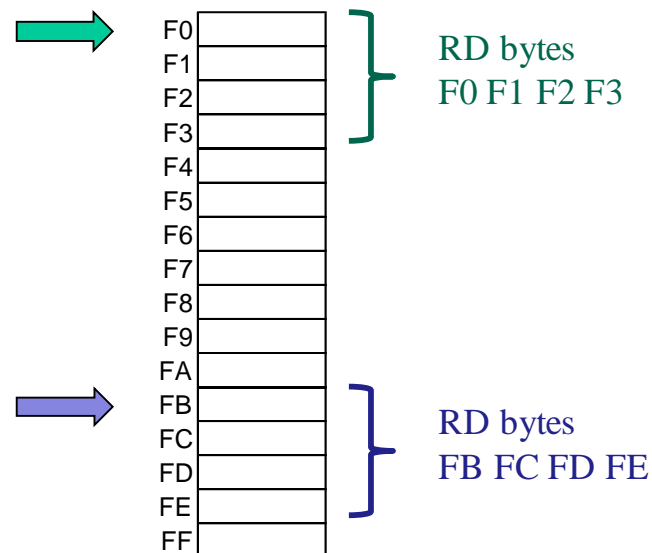
Endereçamento

Primeiro caso: endereçamento acessa uma linha somente

Exmpl1: endereçamento → byte. Endereço aponta para byte. Largura da memória = 1 Byte

Exmpl2: endereçamento → palavra de 32 bits. Largura da memória = 4 Bytes

Segundo caso:
endereçamento a byte mas Read tem output de 4 bytes
Aplicação: dados e instruções



Memórias

Principais tipos memórias:

Memória somente de leitura – Read only memory (ROM)

Memórias de leitura e escrita – Random Access Memory (RAM)

Memórias dinâmicas – Dynamic random access memory (DRAM)

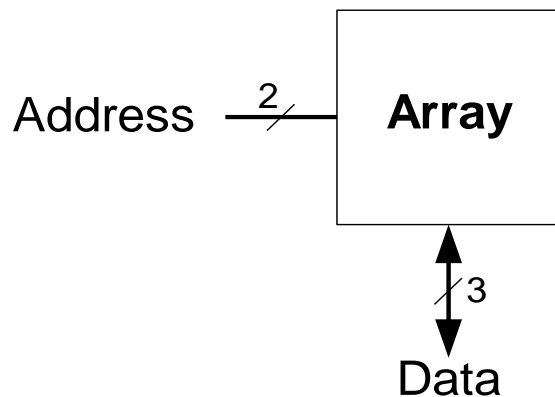
Memórias estáticas – Static random access memory (SRAM)

Um dado de valor de M-bit pode ser lido ou escrito por vez em um endereço de N-bit.

Memória : Exemplo

Array de $2^2 \times 3$ -bit

Word size de 3-bits



Address	Data			
11	0	1	0	<div style="display: flex; align-items: center;"> <div style="flex: 1; border-left: 1px solid black; border-right: 1px solid black; height: 100px; margin: 0 5px;"></div> <div style="flex: 1; text-align: center;"> depth ↑ ↓ </div> </div>
10	1	0	0	
01	1	1	0	
00	0	1	1	
	<div style="display: flex; align-items: center;"> <div style="flex: 1; border-top: 1px solid black; border-bottom: 1px solid black; height: 10px; margin: 0 5px;"></div> <div style="flex: 1; text-align: center;"> width </div> </div>			

Memória : Exemplo

Nº de linhas = $2^{10} = 1024 = 1K$

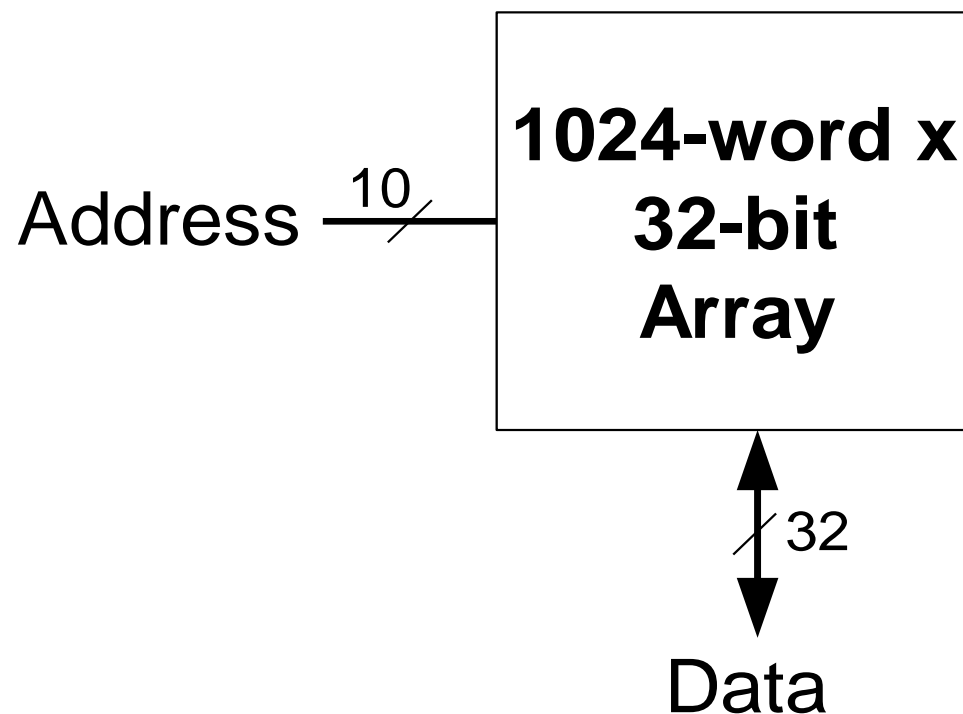
Nº de colunas = word size = 32 bits = 4B

Tamanho

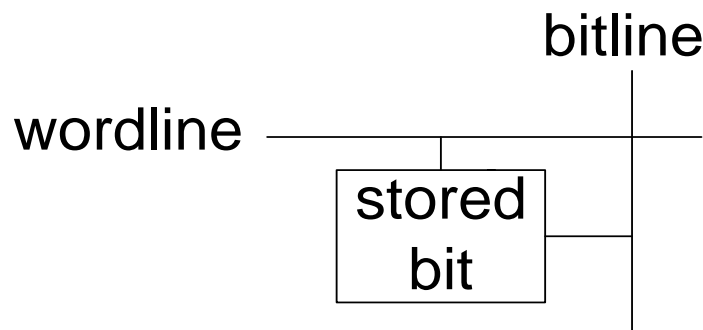
1K x 4B

ou 4KB

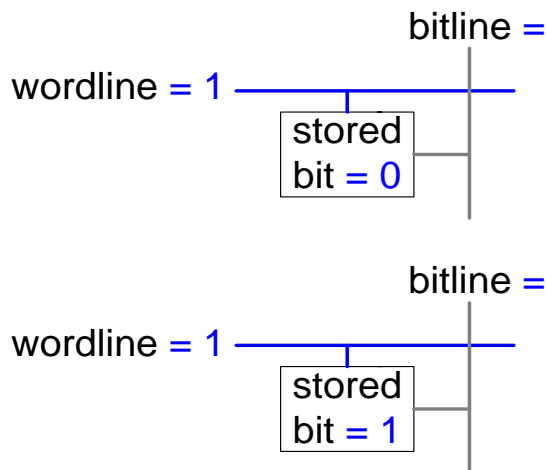
ou 32 Kb



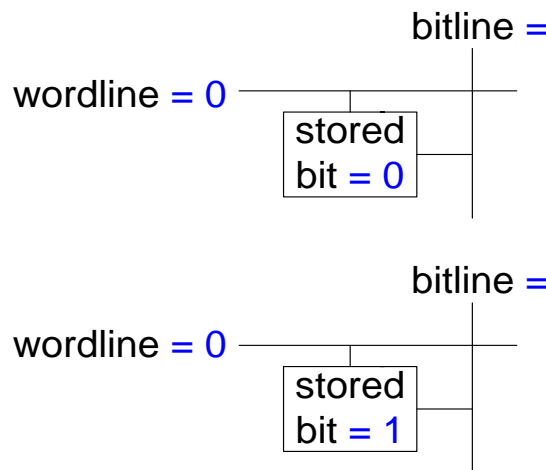
Memória: Célula de bit



Exemplo



(a)



(b)

Procedimento para leitura

Endereço seleciona
(decodificador)
1 linha (1 wordline)

Cada célula selecionada na
wordline aciona o bitline,
levando o valor para a
saída

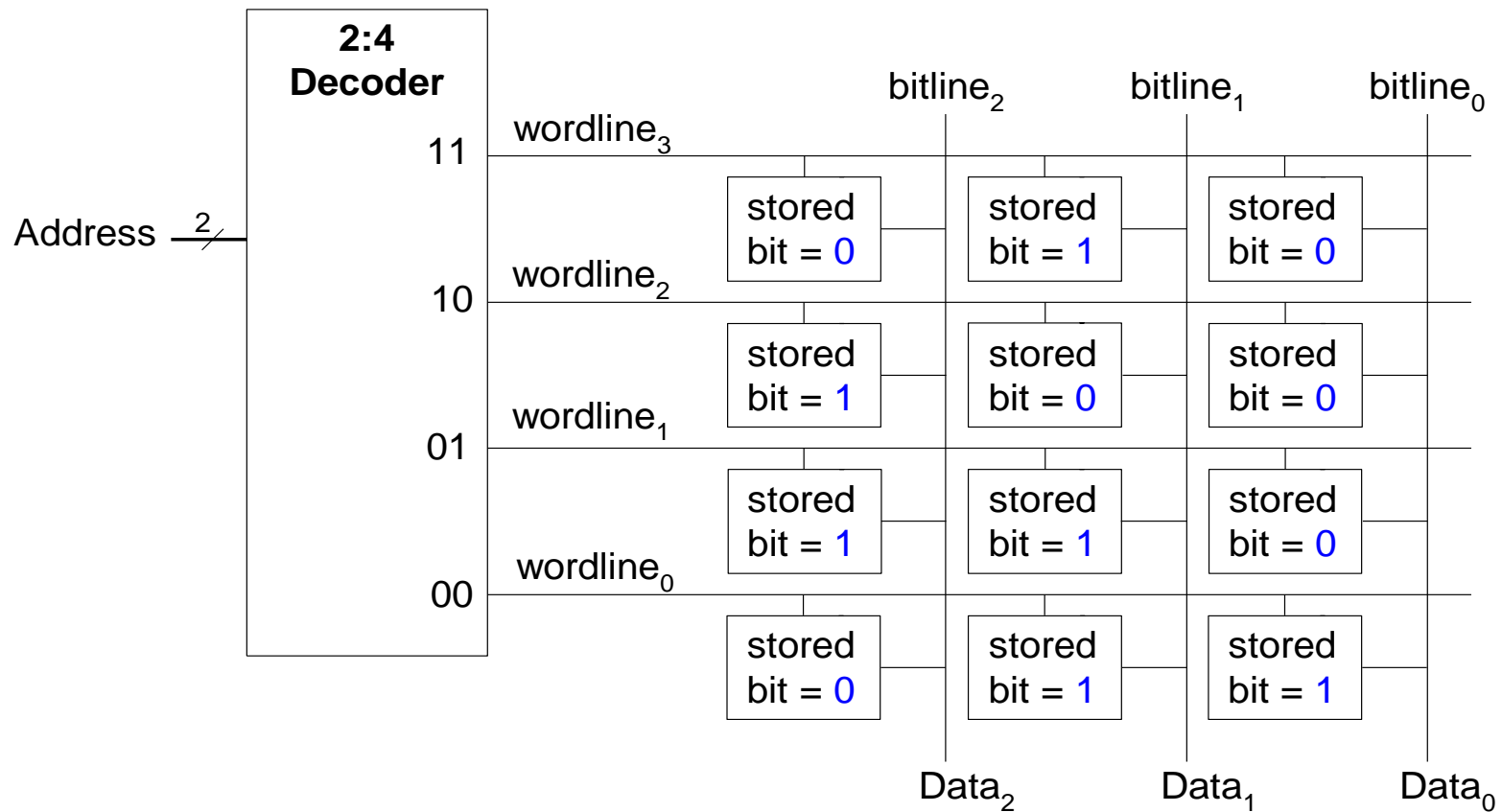
Procedimento para escrita

Endereço seleciona
(decodificador)
1 linha (1 wordline)

Valor a ser escrito colocado
na bitline (bidirecional)

Sinal de controle WR ativa a
escrita do valor do bitline
na célula

Memória: 4x3



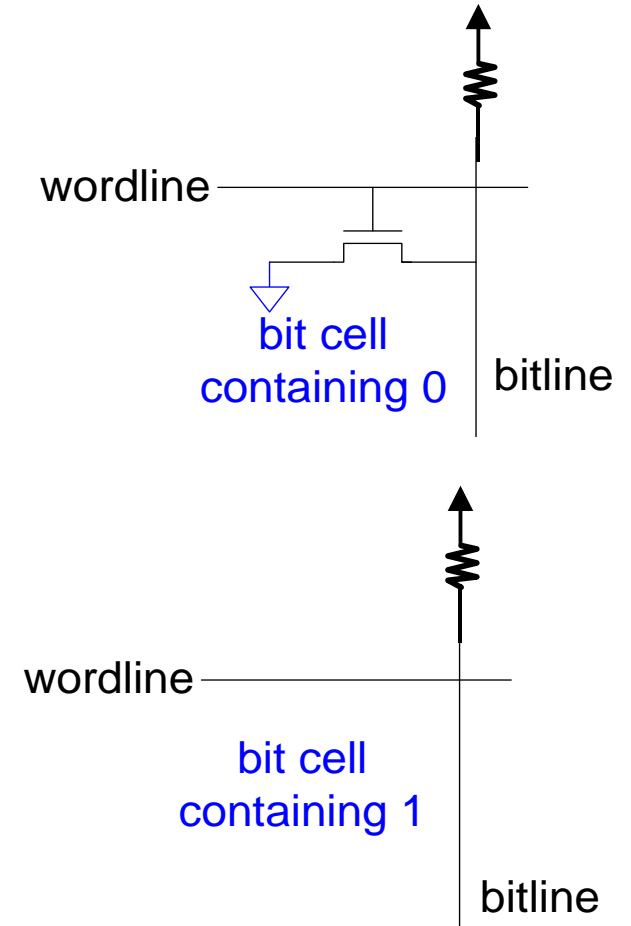
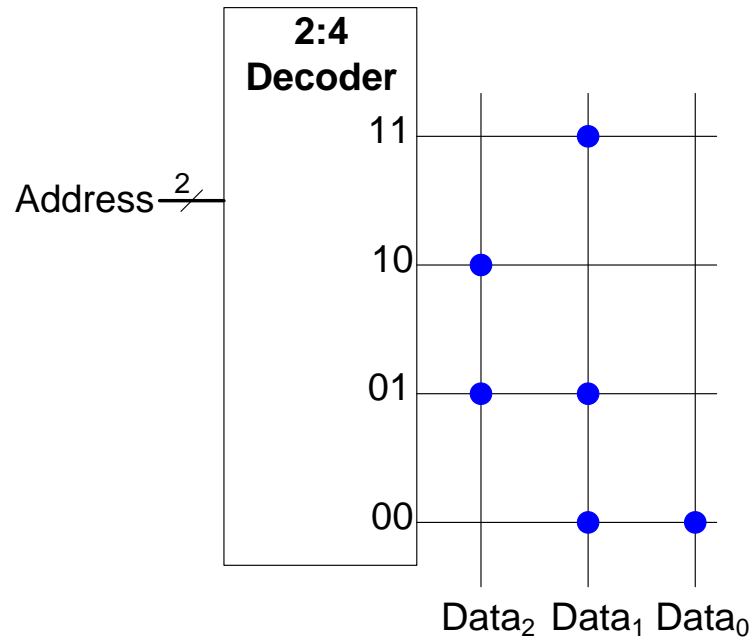
Tipos de Memórias

- Read only memory (ROM): não volátil
- Random access memory (RAM): volátil

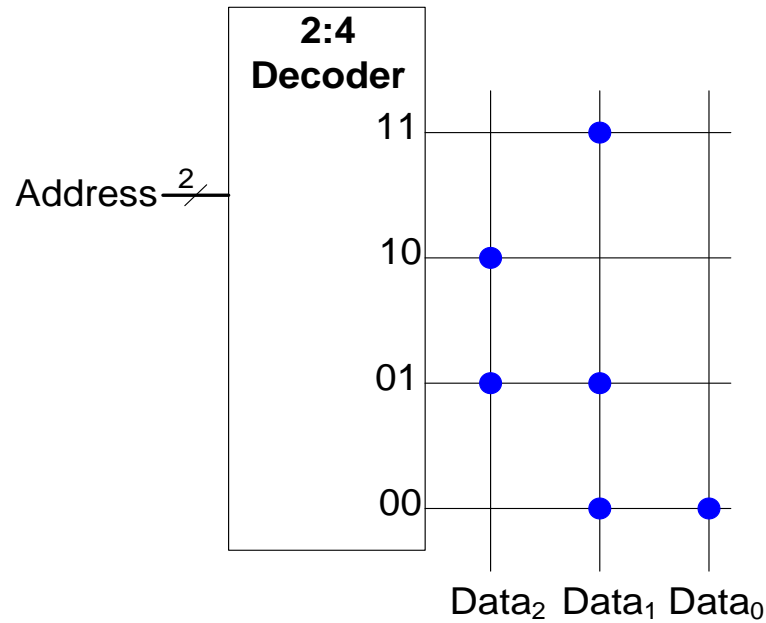
ROM

- Read only memory (ROM)
 - Não volátil: não perdem seus dados quando a alimentação é desligada
 - Pode ser lida rapidamente, porém a escrita é lenta (no caso das ROMs reprogramáveis)
 - Memórias em câmeras digitais, pen drives são ROMs
 - Historicamente denominadas de *read only memory* porque as primeiras ROMs eram fabricadas já com os dados ou escritas posteriormente queimando-se fusíveis → somente leitura

ROM



ROM



Address	Data			depth
11	0	1	0	
10	1	0	0	
01	1	1	0	
00	0	1	1	
width				

Detalhes da ROM

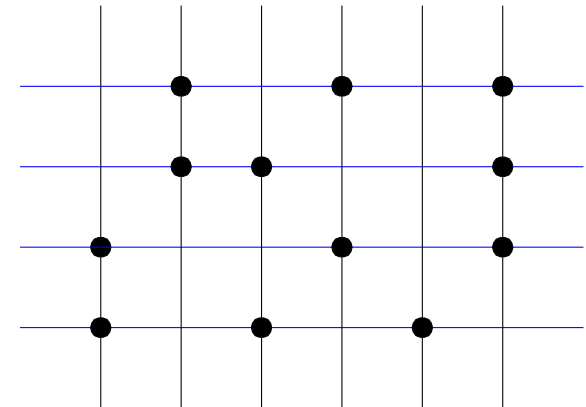
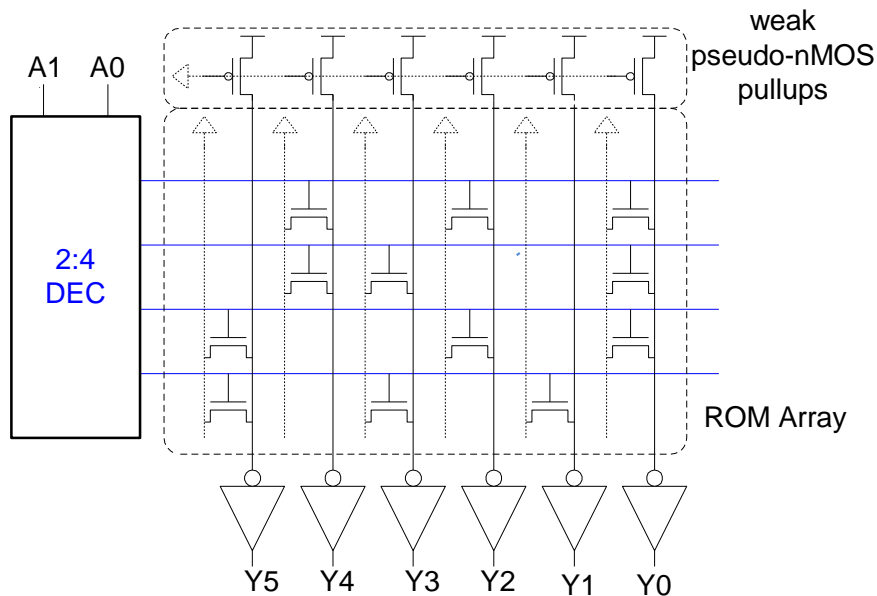
- 4-word x 6-bit ROM
 - Representada por diagrama de pontos
 - Pontos indicam 1's na ROM

Word 0: **010101**

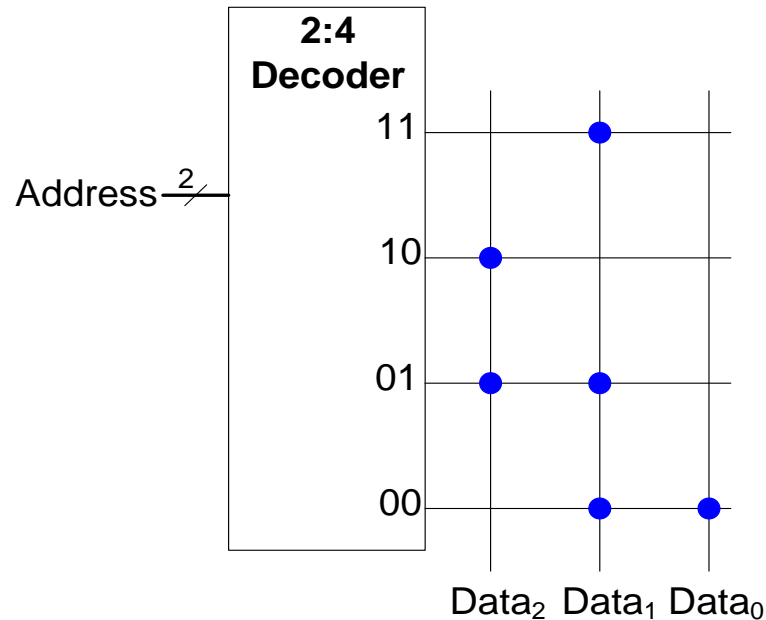
Word 1: **011001**

Word 2: **100101**

Word 3: **101010**



Lógica com ROM



$$Data_2 = A_1 \oplus A_0$$

$$Data_1 = \sim(A_1 \cdot \sim A_0)$$

$$Data_0 = \sim A_1 \cdot \sim A_0$$

RAM

- Random access memory
 - Volátil: perde o dado quando a alimentação é desligada
 - Pode ser lida ou escrita rapidamente
 - A memória principal do seu computador é RAM (specificamente, DRAM)
 - Historicamente denominada de *random access memory* porque qualquer palavra de dado pode ser acessada como qualquer outra (em contraste com *sequential access memories* como fita magnética).

Tipos de RAM

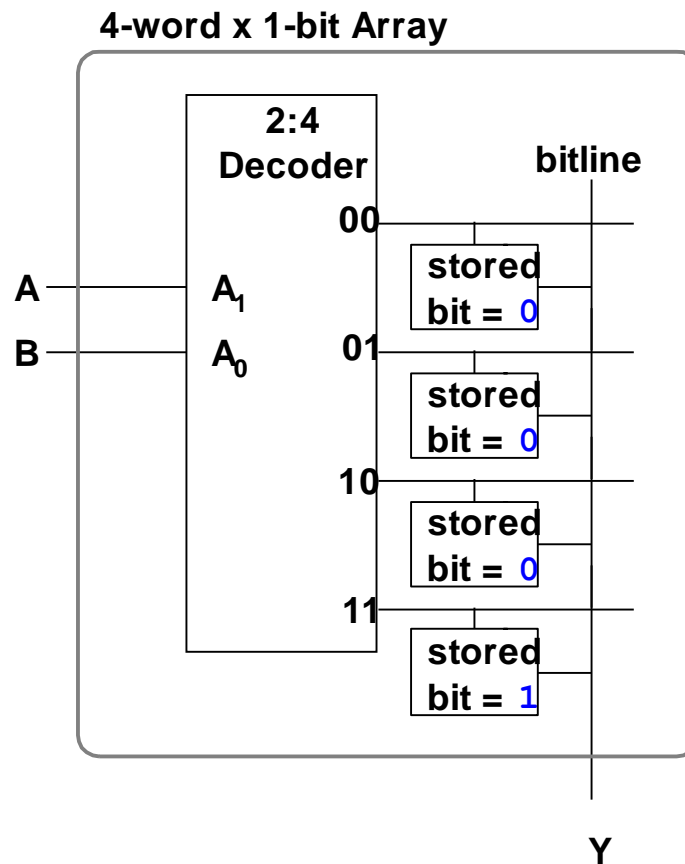
- Os dois tipos de RAM são:
 - Dynamic random access memory (DRAM)
 - Static random access memory (SRAM)
- A diferença é como armazenam os dados:
 - DRAM usa um capacitor
 - SRAM usa **cross-coupled inverters** (“latch”)

Lógica com Memória

- A memória usada para executar funções lógicas é denominada *lookup tables* (LUT).
- O usuário tem o valor de saída para cada combinação das entradas (address).

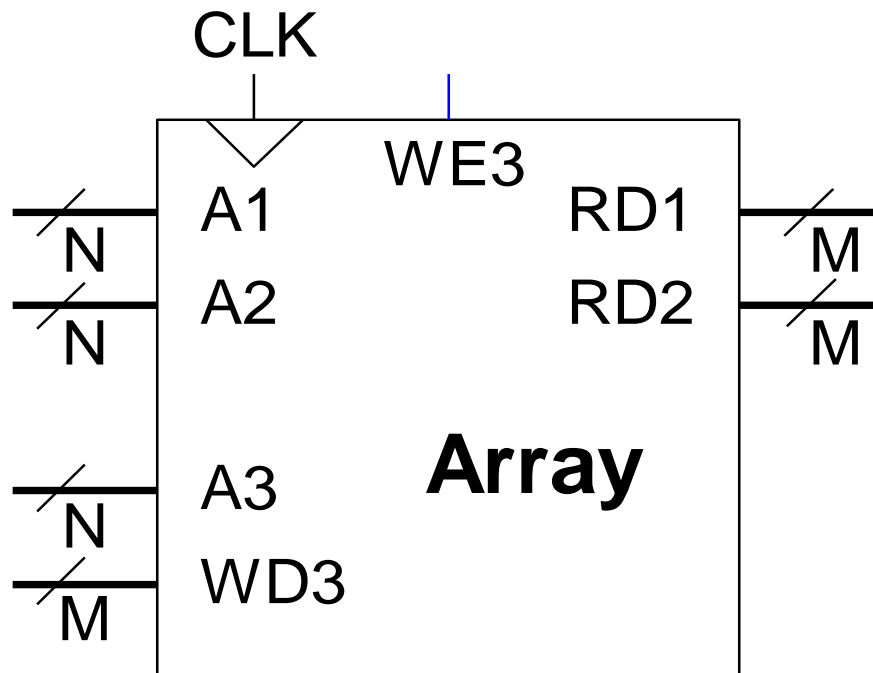
Truth Table

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

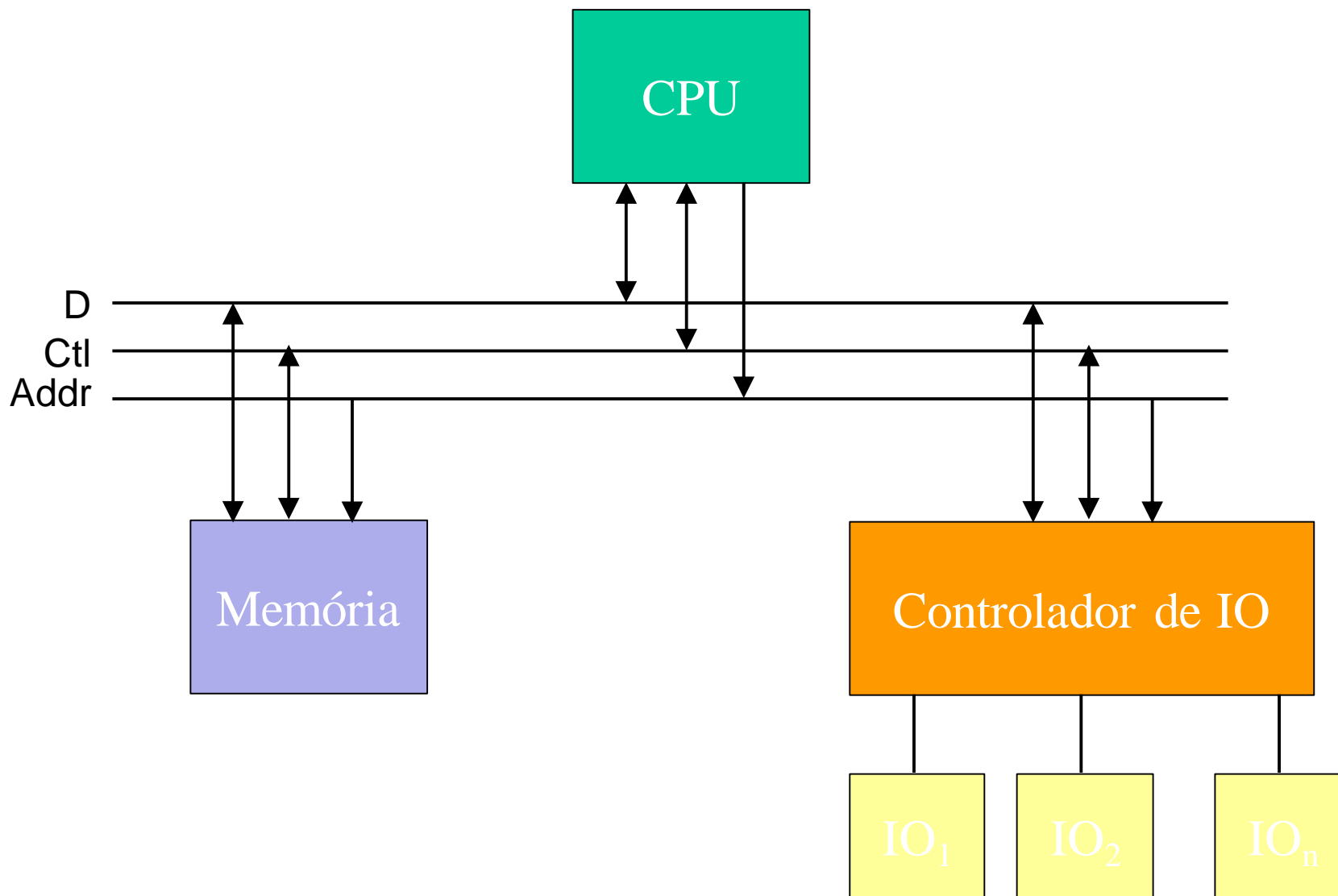


Memórias Multi-Portas

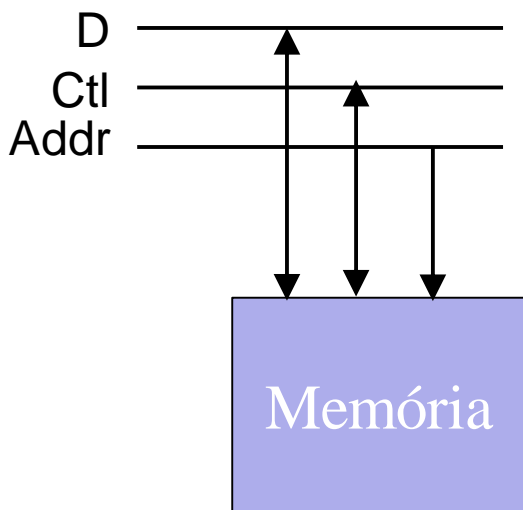
- Porta: par endereço/dado (address/data)
- Memória 3-portas
 - 2 portas de leitura (A1/RD1, A2/RD2)
 - 1 porta de escrita (A3/WD3, WE3 enables writing)



Sistema de memória: uso típico



Dispositivo de memória: interfaces



- Dados:
 - bidirecional: dados a serem escritos ou lidos
- Address:
 - input apenas: endereço da posição de memória onde os dados serão escritos ou de onde serão lidos
- Control
 - Inputs:
 - RD, WR → indicam a operação a ser executada
 - OE: output enable (saída Z)
 - Outputs: não é usual
 - poderia ser status = ready, por exemplo

Constituição de um sistema de memória com componentes

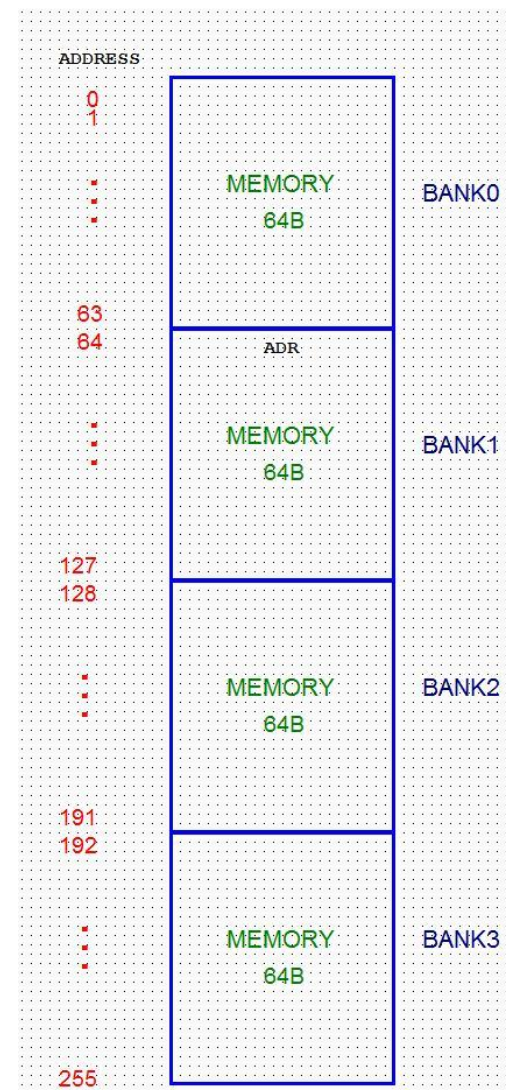
- Um sistema de memória: ex. 8GB
 - endereçamento a bytes: necessários 33 bits
 - controle: RD e WR
 - dados: 8 bits
- Pode ser constituído por 8 chips de memória de 1 GB cada. Cada chip:
 - endereçamento: necessários 30 bits → usar os 30 LSB dos 33 bits do sistema
 - controle: RD e WR comuns a todos os chips
 - dados: 8 bits comuns a todos os chips

Decodificação de endereço

- Em geral, sistema de memória
 - n módulos (chips) de tamanho fixo
 - barramentos grandes, para permitir expansão
 - decodificação de endereço
- Exemplo em seguida

Mapa da Memória

- Descreve como bancos de memória (chips) podem ser ligados para formar a memória global
- Supor sistema de memória com os seguintes sinais:
 - adr: endereços (8-bits)
 - memdata: leitura de dados (8-bits)
 - writedata: escrita de dados (8-bits)
 - clk: clock
 - memwrite: habilita escrita
- Como conectar os pinos dos vários chips de memória com os sinais do processador?



Decodificação de Endereços

- Exemplo

Chips (bancos) de memória são de 64B

Memória total desejável é de 256B

Barramento de endereços e dados com 8 bits

