

Aula 18

Memórias

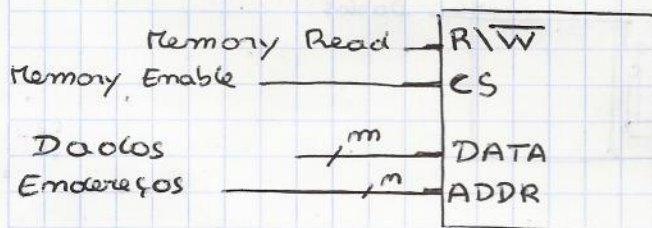
Tipos de Memória

- RAM (Random Access Memory)
- ROM (Read Only Memory)
 - PROM (Programmable ROM)
 - EPROM (Erasable PROM)
 - EEPROM (Electric EPROM)

RANDOM

↳ O Tempo de Acesso é "aleatório" ou seja, independente da localização de que se quer acessar

Circuito de Memória

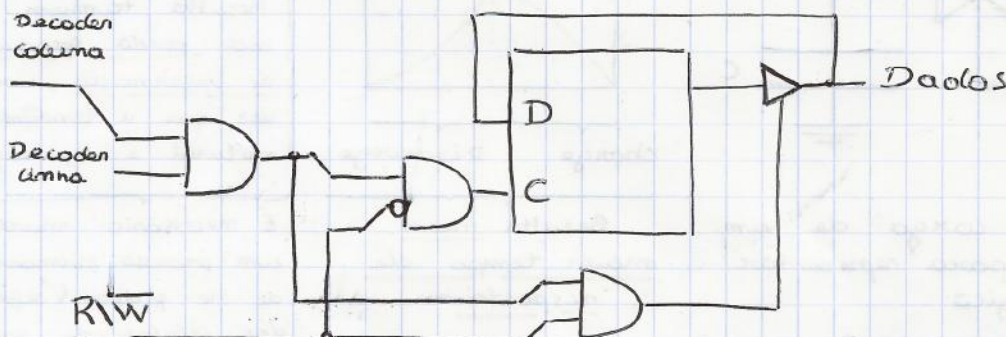


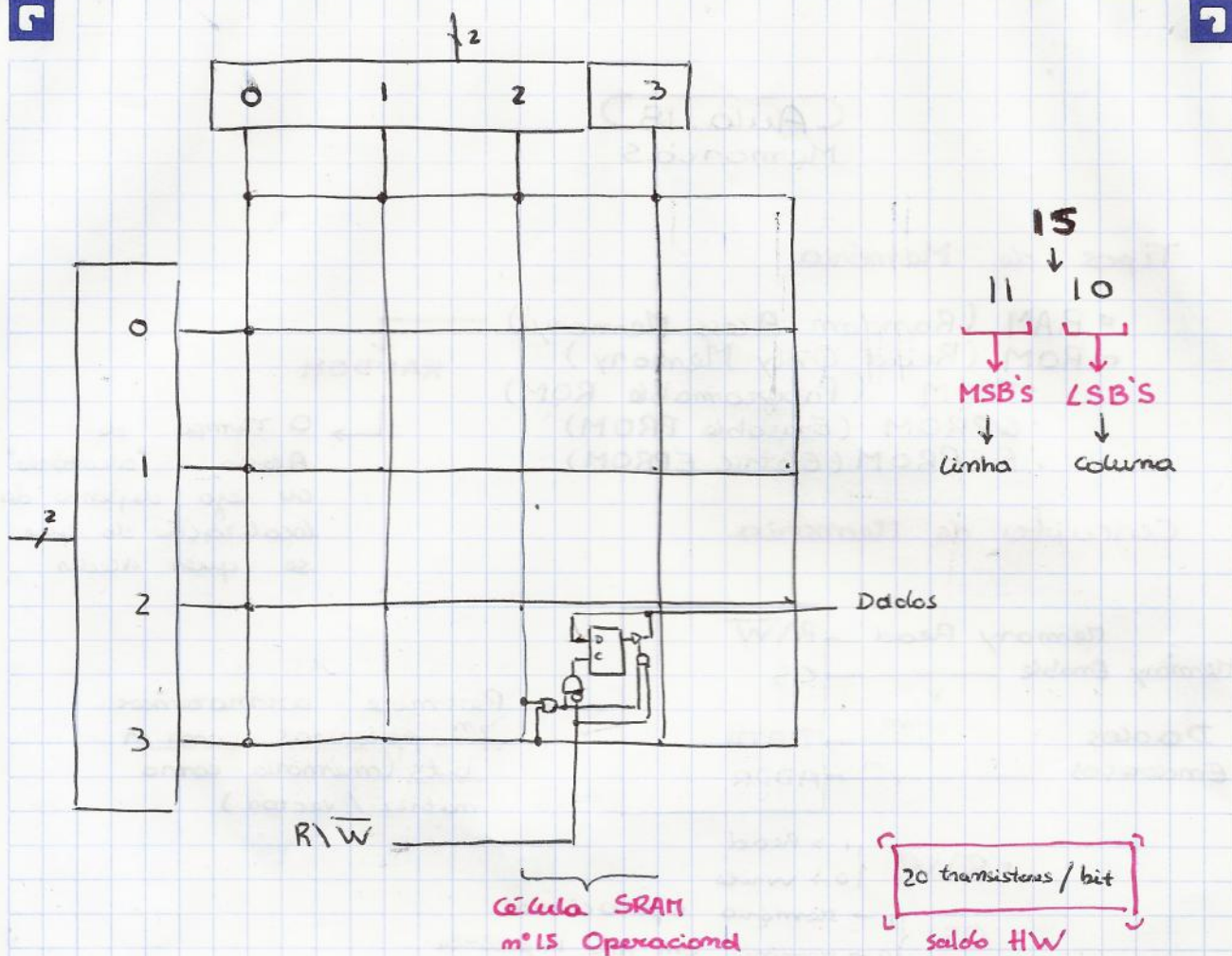
Permite armazenar 2^m palavras com n bits (memória como matriz / vector)

- $\overline{R/W}$ $\left\{ \begin{array}{l} 1 \rightarrow \text{Read} \\ 0 \rightarrow \text{Write} \end{array} \right.$
- \overline{CS} $\left\{ \begin{array}{l} 1 \rightarrow \text{memória operacional} \\ 0 \rightarrow \text{memória em Alta Impedância} \end{array} \right.$

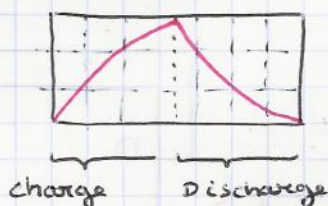
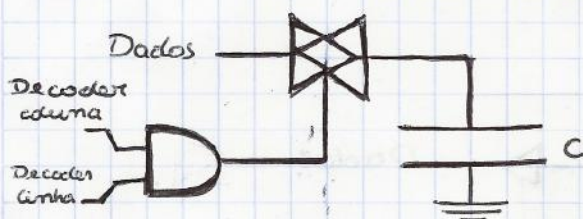
Memórias estáticas e Memórias Dinâmicas

SRAM - STATIC RAM





DRAM - Dinamic RAM



Resulta num maior tempo de resposta

Resulta também na perda gradual de informação, uma vez que a tendência natural é descarregar

- Utilização da carga de um condensador para representar um valor lógico

É necessário manter um processo permanente de re-escrita ("refreshamento") das células de forma a evitar as potenciais perdas de informação

Prós e contras - DRAM

- leitura destrutiva (perda de carga na operação de leitura)

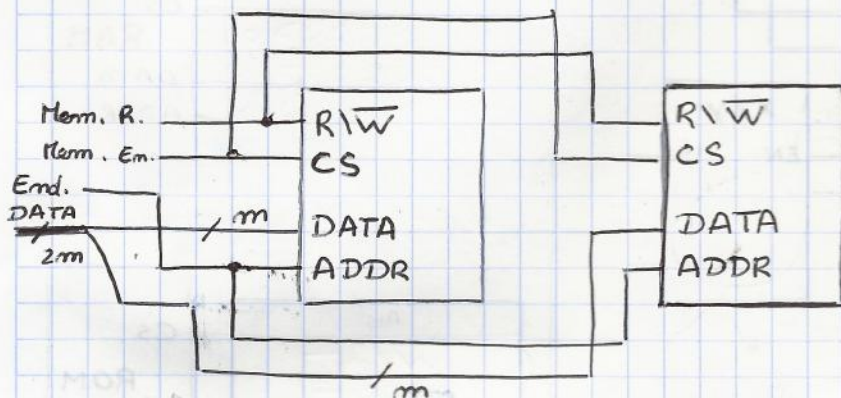
+ Muito baratas

5 transistores/bit
saldo HW

Planos de Memória

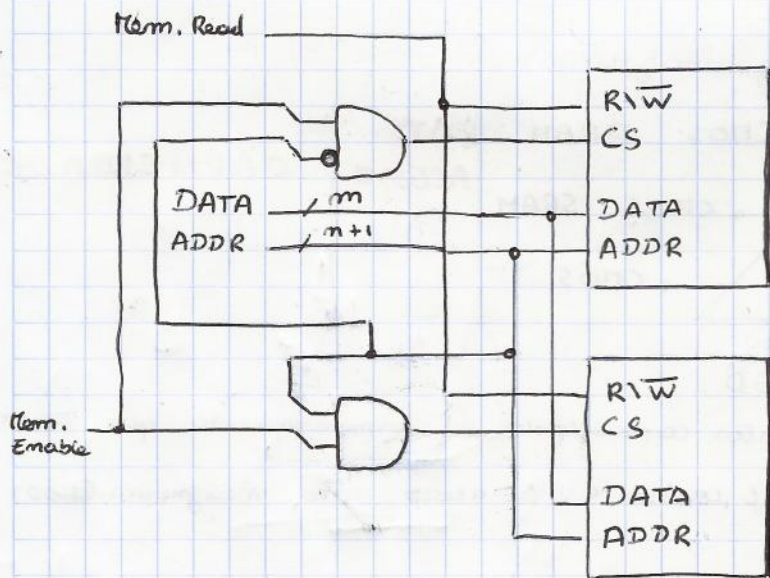
Expansão da largura

Exemplo: $2^m \times m$ bits $\rightarrow 2^m \times 2m$ bits



Expansão dos Endereços

Exemplo: $2^m \times m$ bits $\rightarrow 2^{m+1} \times m$ bits



→ Opera endereços:

0 (...) 0 0 0 0
|
0 (...) 1 1 1 1

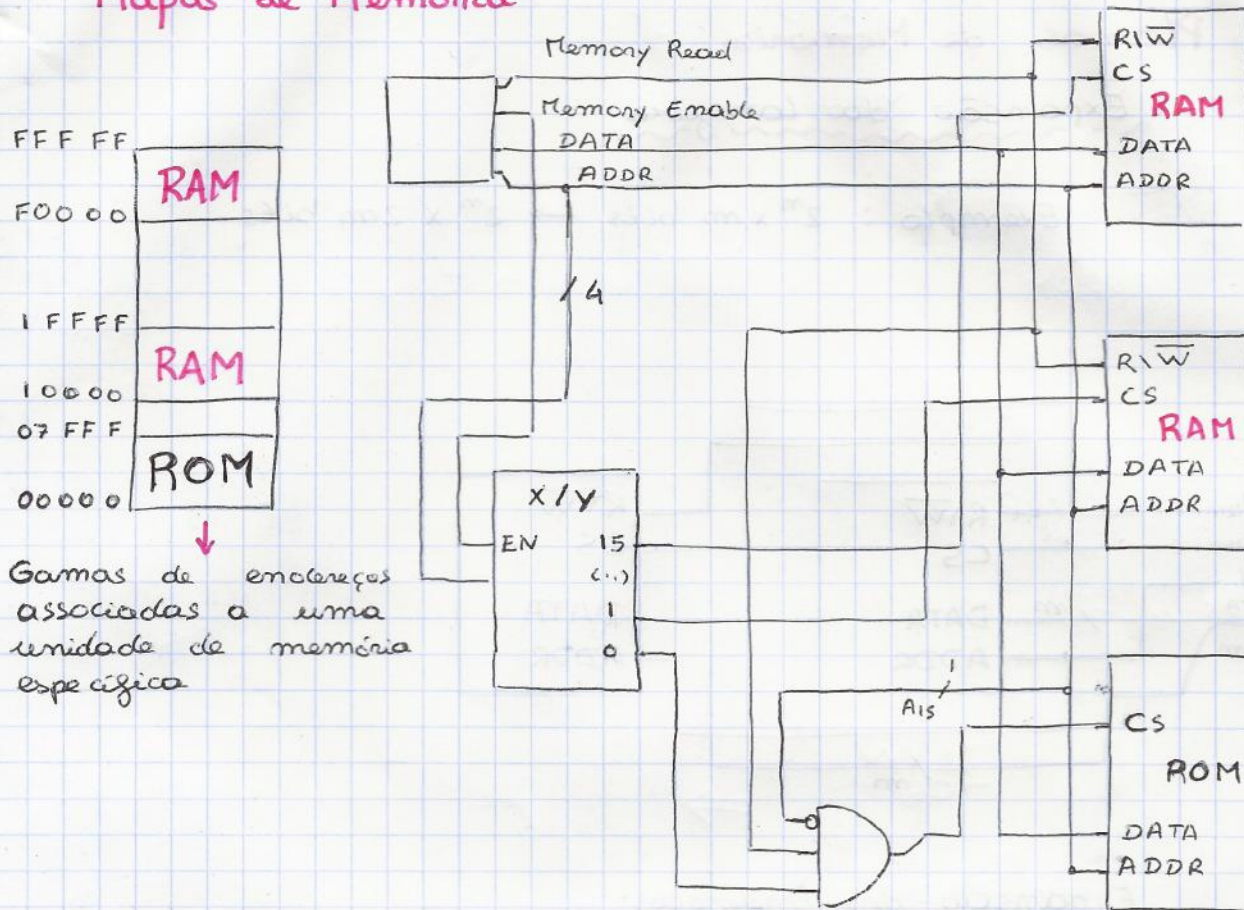
0 (...) $\phi \phi \phi \phi$

→ Opera endereços:

1 (...) 0 0 0 0
|
1 (...) $\phi \phi \phi \phi$

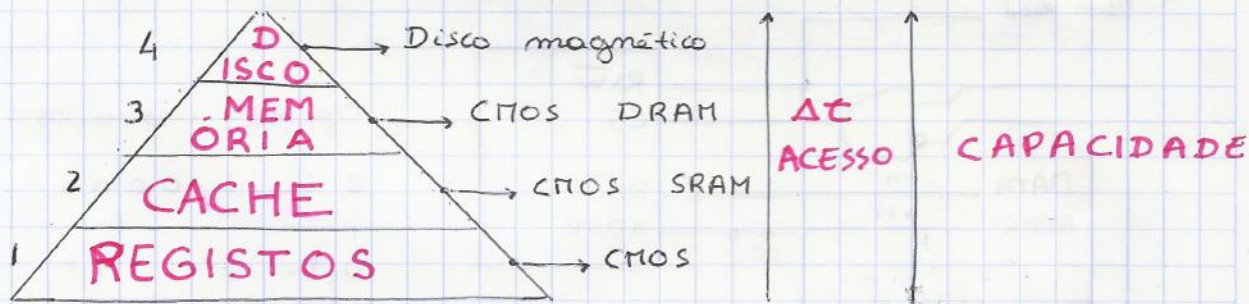
1 (...) $\phi \phi \phi \phi$

Mapas de Memória



↓
Gammas de endereços associadas a uma unidade de memória específica

Hierarquia de Memória



Nível 3.5 → Disco SSD

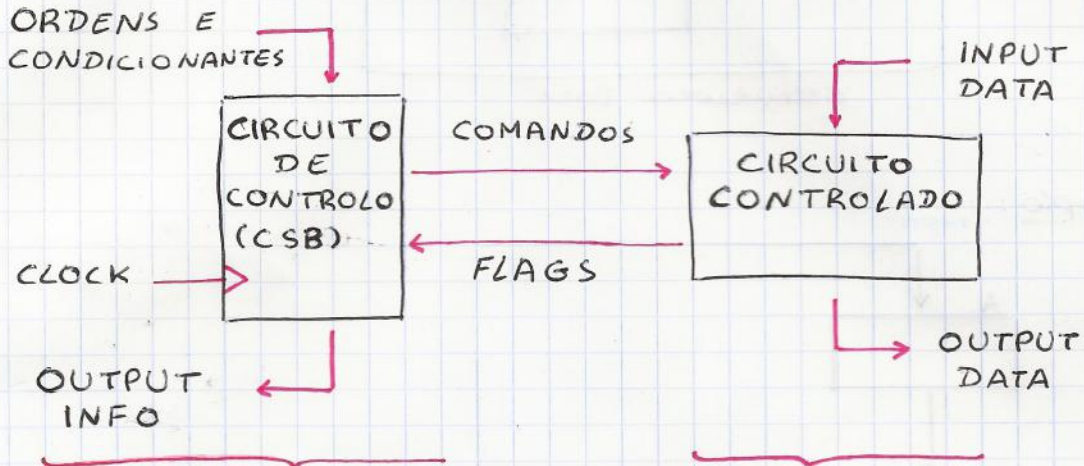
↳ Maior capacidade de armazenamento que DRAM's
↳ Δt acesso < Δt acesso disco magnético (HDD)

Aula 19

MEMp.

Circuitos de Dados

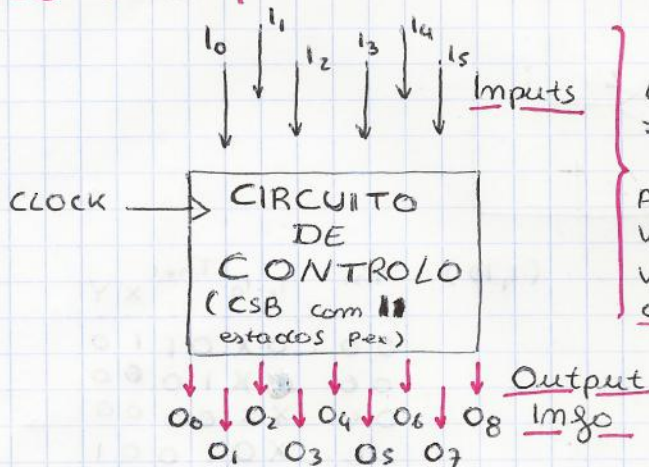
Circuitos de Controlo e Circuitos de Dados



Obrigatoriamente
um circuito Sequencial
(CSB)

Sequencial
ou
Combinatório

Condensação de circuitos controladores complexos

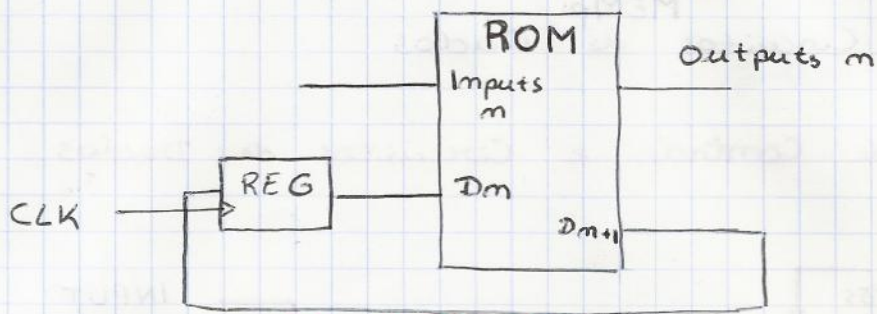


6 entradas + 11 estados =
= 17 variáveis

A resolução com mapas de Karnaugh ou tabelas de verdade é altamente dispendiosa de tempo

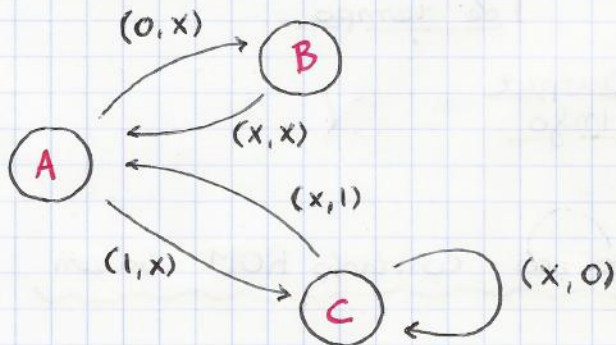
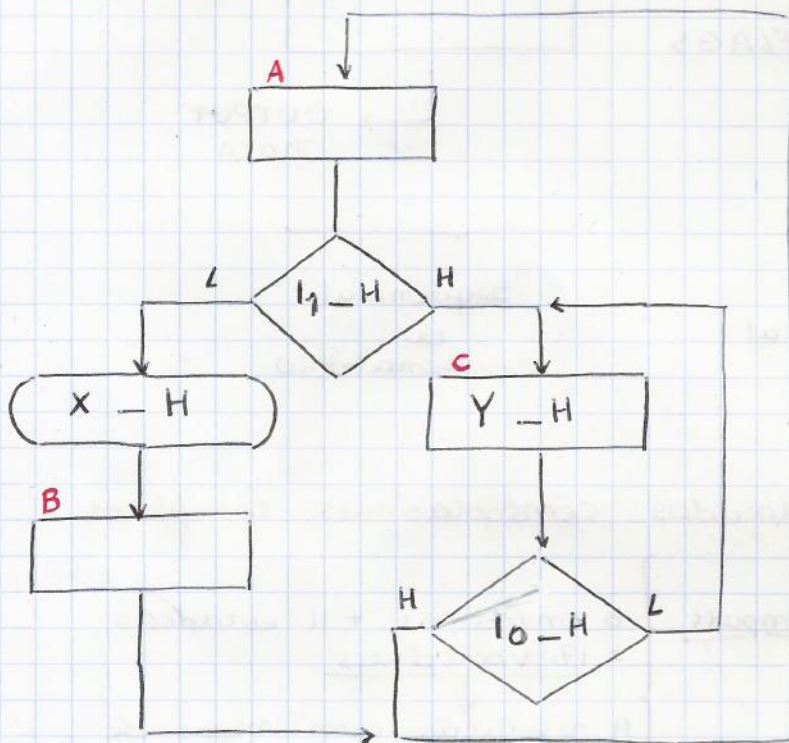
Solução : Circuito de controlo ROM-driven

Estrutura básica de controle por ROM



Estrutura Base

Exemplo:



(1, 10)

T_m	I_1	I_0	T_{m+1}	X	Y
00	0	X	0	1	0
00	1	X	1	0	0
01	X	X	0	0	0
10	X	0	1	0	1
10	X	1	0	0	1

Tabela de estados Base

T _m	I ₁	I ₀	T _{m+1}	X	Y	ADDR
A ₃ A ₂	A ₁	A ₀				
00	0	0	01	1	0	0
00	0	1	01	1	0	1
00	1	0	10	0	0	2
00	1	1	10	0	0	3
01	0	0	00	0	0	4
01	0	1	00	0	0	5
01	1	0	00	0	0	6
01	1	1	00	0	0	7
10	0	0	10	0	1	8
10	0	1	00	0	1	9
10	1	0	10	0	1	10
10	1	1	00	0	1	11
11	X	X	XX	X	X	12-15

→ Permite estados de erro.

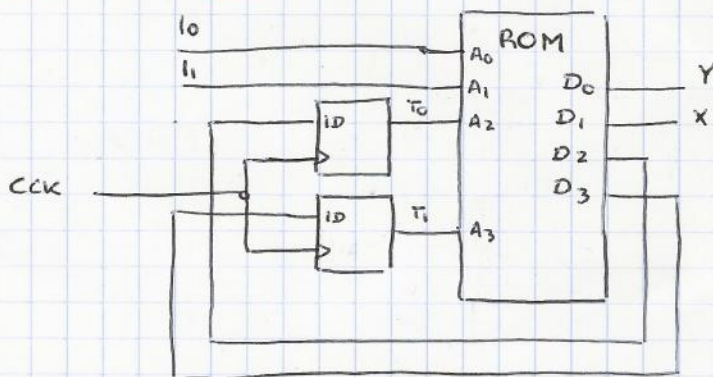
Tabela de transição de estados com endereços

Tabela de conteúdo da ROM

ADDR DATA

0	01 10
1	01 10
2	10 00
3	10 00
4	00 00
5	00 00
6	00 00
7	00 00
8	10 01
9	00 01
10	10 01
11	00 01
12-15	00 00 → Inibição dos erros

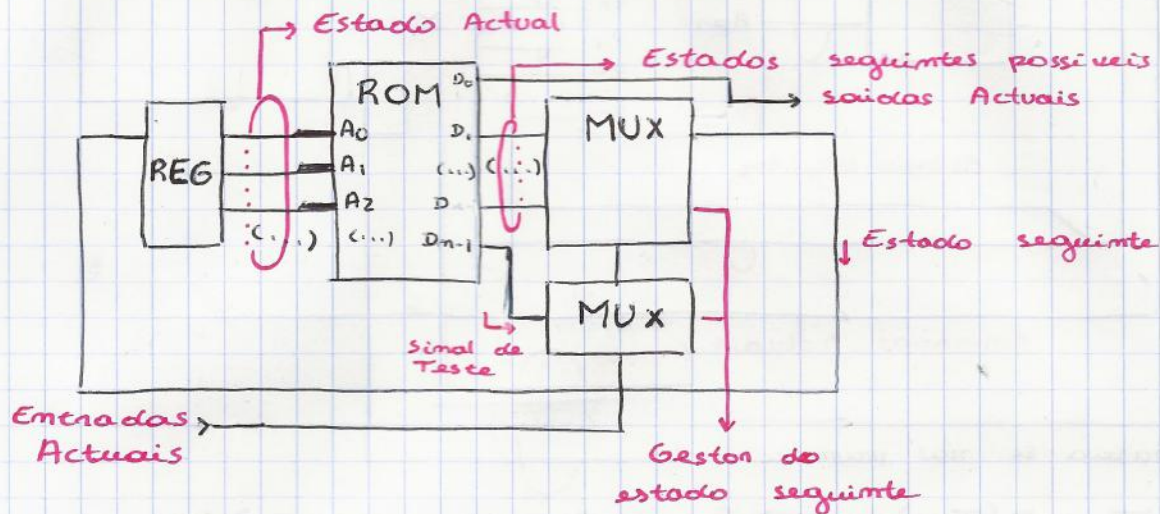
Implementação



Aula 20

Endereçamento Implícito e Explícito

Controlo por ROM com endereçamento Explícito

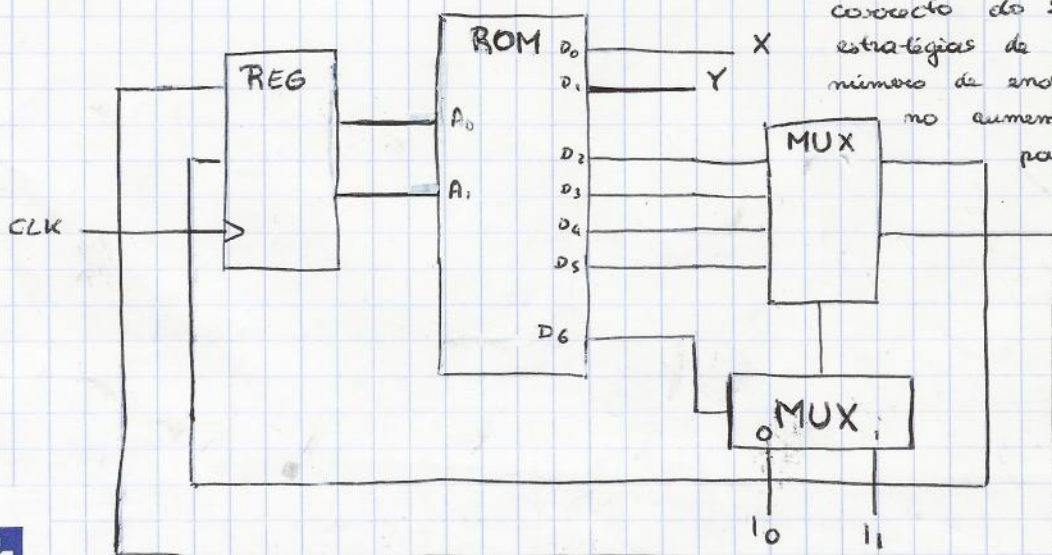


Utilizando o exemplo da Aula 19:

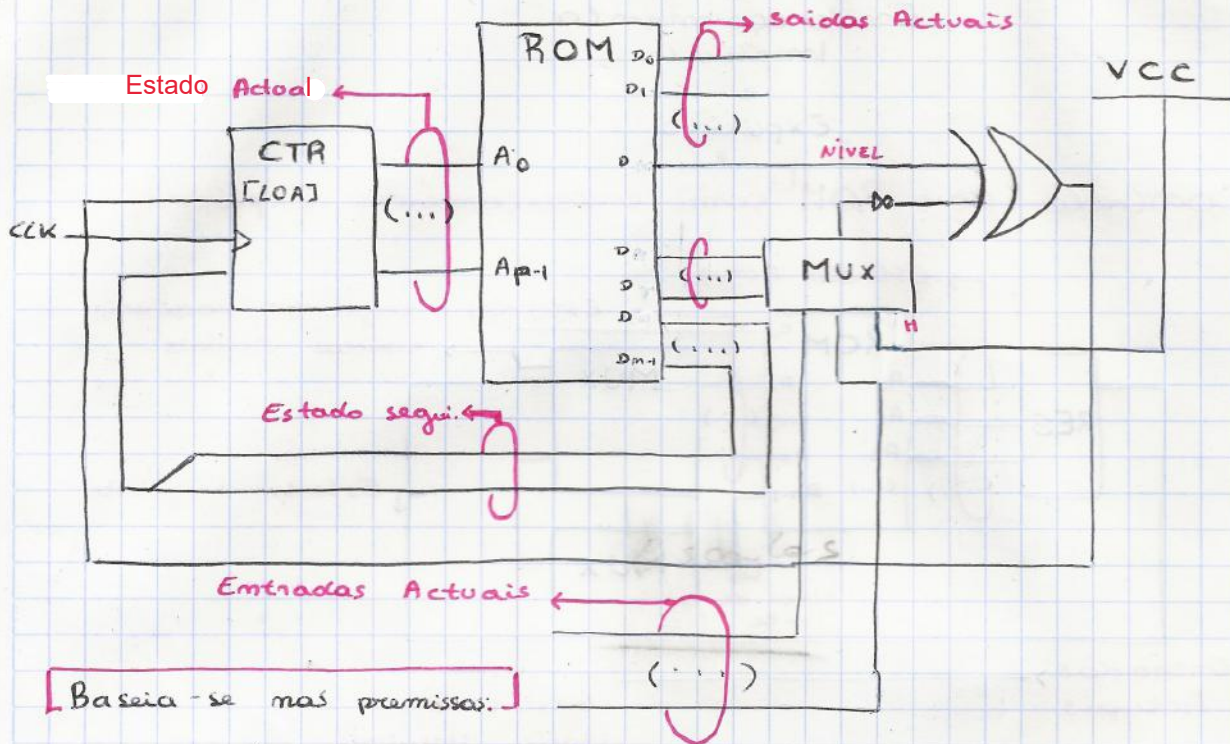
T_m	T_{st}	T_{m+1}	Y	X
$A_1 A_0$	D_6	$D_5 D_4 D_3 D_2$	D_1	D_0
A 00	1	1 0 0 1	0	0
B 01	X	0 0 0 0	0	1
C 10	0	0 0 1 0	1	0
1 1	X	X X X X	X	X

ADDR	DATA
0	1100100
1	0000001
2	0001010
3	0000000

Conclui-se que, logicamente para manter o funcionamento correcto do sistema as estratégias de diminuição do número de endereços resultam no aumento de bits da palavra



Controlo por ROM com endereçamento implícito



Baseia-se nas premissas:

- $T_{m+1} = (T_m) + 1$ → Função [INC] no Contador
- ou
- $T_{m+1} \neq (T_m) + 1$ → Função [LOAD] no contador
 - [LOAD] condicional: Depende do valor de uma variável de entrada
 - [LOAD] Incondicional: Seleção da entrada $H(VCC)$, Nível a 1