

Introdução à Arquitetura de Computadores

Aula 4

Os Blocos Combinatórios Básicos

Circuitos Sequenciais e Máquinas de Estados

Pedro M. Lavrador

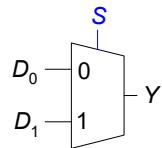
Departamento de Electrónica, Telecomunicações e Informática
Universidade de Aveiro
plavrador@ua.pt

Índice

- Os Blocos Combinatórios Básicos.
 - *Multiplexers*
 - Descodificadores
 - Somadores
- Algumas considerações sobre tempo
- Circuitos Sequenciais
 - Latch R-S e Flip-Flops tipo D
 - Registros e Memórias
 - Máquinas e diagramas de estado

Multiplexer

- Um Multiplexer (multiplexador) ou Mux é um circuito com N entradas de dados e 1 saída, e com $\log_2 N$ entradas de seleção que especificam qual das entradas liga à saída.
- Um Mux 2:1 tem 2 entradas de dados, uma entrada de seleção e uma saída.



S	Y
0	D_0
1	D_1

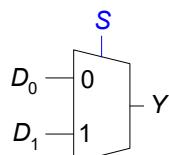
09/03/2018

PML – IAC - 2018

3

Multiplexer

- Exercício:**
 - Escrever a tabela de verdade do Mux 2:1, simplificá-la e representar uma implementação do circuito usando portas lógicas.



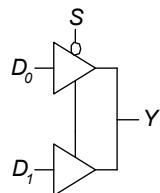
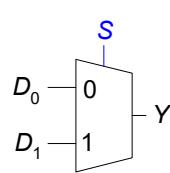
09/03/2018

PML – IAC - 2018

4

Multiplexer

- O multiplexer pode também ser implementado usando portas *tristate*.
- Uma porta *tristate*, pode ser descrita de modo simplista como uma porta que admite um terceiro estado isto é ela pode ter à saída o estado lógico '0', '1' ou 'alta impedância'.



09/03/2018

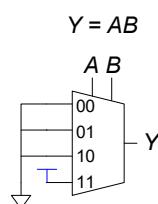
PML – IAC - 2018

5

Funções Lógicas com Multiplexers

- O multiplexer pode ser usado para implementar funções lógicas, servindo como implementação direta de uma tabela de verdade:
 - As variáveis lógicas são usadas como entradas de seleção;
 - As entradas do multiplexer são as constantes '0' ou '1' que implementam a função pretendida.
- Por exemplo: Implementação de uma AND com mux.

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



09/03/2018

PML – IAC - 2018

6

Funções Lógicas com Multiplexers

- Exercício:

- Usar um multiplexer 8:1 para implementar a seguinte função:
- $Y = A \cdot \bar{B} + \bar{B} \cdot \bar{C} + \bar{A} \cdot B \cdot C$
- É possível implementar a função anterior usando um multiplexer 4:1?

09/03/2018

PML – IAC - 2018

7

Índice

- Os Blocos Combinatórios Básicos.
 - Multiplexers
 - Descodificadores
 - Somadores
- Algumas considerações sobre tempo
- Circuitos Sequênciais
 - Latch R-S e FlipFlops tipo D
 - Registros e Memórias
 - Máquinas e diagramas de estado

09/03/2018

PML – IAC - 2018

8

Descodificador

- Um descodificador é um circuito com N entradas e 2^N saídas.
- Em cada instante está ativa a saída correspondente ao número que está representado na entrada.
 - Só está ativa uma saída em cada instante;
- O descodificador é um circuito necessário por exemplo para aceder a memórias:
 - permite selecionar a célula a partir do seu endereço.

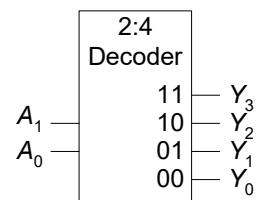
09/03/2018

PML – IAC - 2018

9

Descodificador

- Um descodificador de 2 entradas tem 4 saídas.



A_1	A_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

$$\begin{aligned}
 Y_0 &= \overline{A_1} \cdot \overline{A_0} = m_0 \\
 Y_1 &= \overline{A_1} \cdot A_0 = m_1 \\
 Y_2 &= A_1 \cdot \overline{A_0} = m_2 \\
 Y_3 &= A_1 \cdot A_0 = m_3
 \end{aligned}$$

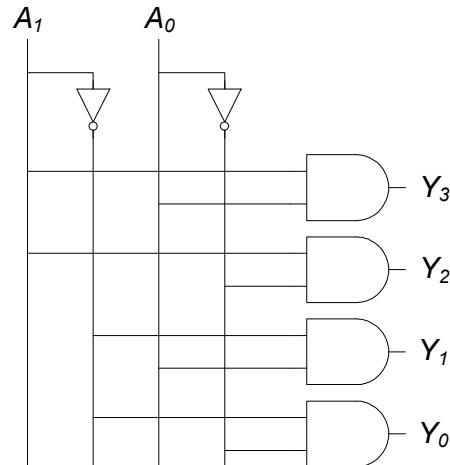
09/03/2018

PML – IAC - 2018

10

Descodificador

- A equação da saída Y_x corresponde ao mintermo x .



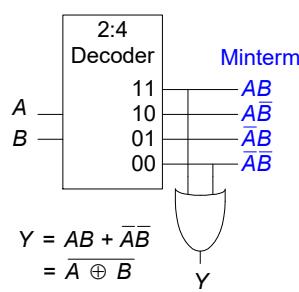
09/03/2018

PML – IAC - 2018

11

Descodificador

- O descodificador pode ser visto como um “gerador de mintermos e portanto usado como uma base para a implementação de equações lógicas escritas na forma de soma de produtos.
- Por exemplo o operador XNOR ($Y = AB + \bar{A}\bar{B}$) pode ser implementado usando um descodificador como:



09/03/2018

PML – IAC - 2018

12

Índice

- Os Blocos Combinatórios Básicos.
 - *Multiplexers*
 - Descodificadores
 - Somadores
- Algumas considerações sobre tempo
- Circuitos Sequênciais
 - Latch R-S e FlipFlops tipo D
 - Registos
 - Máquinas e diagramas de estado

09/03/2018

PML – IAC - 2018

13

Blocos Combinatórios Básicos

Somador Completo de um bit

- Quando fazemos uma adição em binário, de dois números: A mais B:

$$\begin{array}{r}
 1110 \\
 0010 \\
 +1111 \\
 \hline
 10001
 \end{array}$$

- Na coluna i temos que considerar os bits A_i , B_i , o $Carry In_i$ e calcular as saídas Y_i e $Carry Out_i$.

09/03/2018

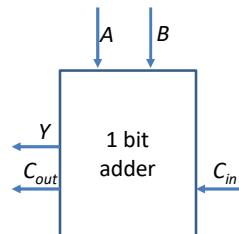
PML – IAC - 2018

14

Somador Completo de um bit

- Exercício:

- Escreva a tabela de verdade do somador completo, considerando:
 - como entradas A , B e C_{in}
 - como saídas Y e C_{out}
- Represente uma implementação possível do somador completo.



09/03/2018

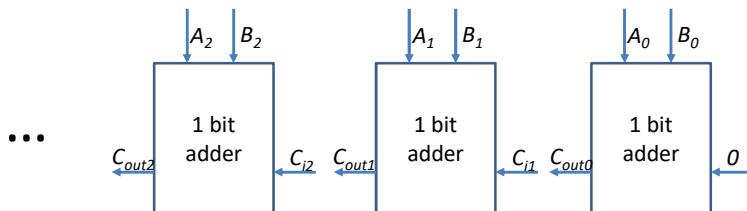
PML – IAC - 2018

15

Somador Completo

- Somador de N bits:

- Uma possível implementação de um somador de N bits é cascatear vários somadores



- Esta solução tem um tempo de atraso muito grande porque a soma do último bit depende do resultado do primeiro.
 - Os tempos de atraso de cada somador propagam-se.
 - Existem melhores implementações.

09/03/2018

PML – IAC - 2018

16

Outros Circuitos Básicos

- Além dos Multiplexers, Descodificadores e Somadores são ainda blocos fundamentais:
 - Comparadores
 - Unidades aritméticas e lógicas
 - Realizam uma de várias operações possíveis.
 - A saída é escolhida com um multiplexer.
 - e *Shifters*

09/03/2018

PML – IAC - 2018

17

Índice

- Os Blocos Combinatórios Básicos.
 - *Multiplexers*
 - Descodificadores
 - Somadores
- **Algumas considerações sobre tempo**
- Circuitos Sequenciais
 - Latch R-S e FlipFlops tipo D
 - Registros e Memórias
 - Máquinas e diagramas de estado

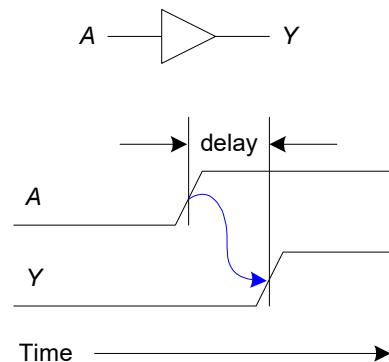
09/03/2018

PML – IAC - 2018

18

Algumas considerações sobre tempo

- Uma alteração no sinal de entrada demora algum tempo a produzir alteração da saída (a propagação não é instantânea).



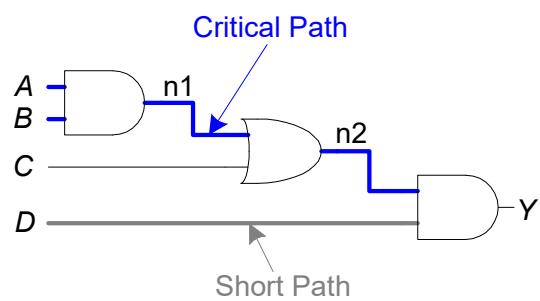
09/03/2018

PML – IAC - 2018

19

Algumas considerações sobre tempo

- Caminho curto e caminho longo.



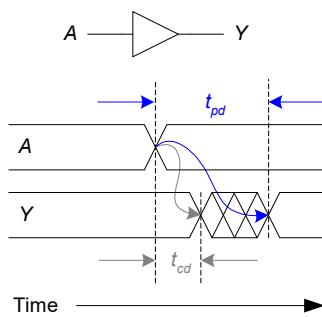
09/03/2018

PML – IAC - 2018

20

Algumas considerações sobre tempo

- Define-se **tempo de propagação (t_{pd} delay)** como o tempo máximo que o sinal demora a propagar-se da entrada para a saída.
- Define-se **tempo de contaminação (t_{cd})** como o tempo mínimo que o sinal demora a propagar-se da entrada para a saída.



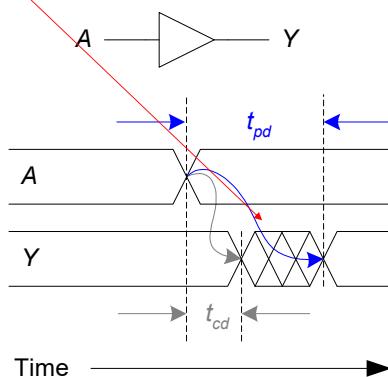
09/03/2018

PML – IAC - 2018

21

Algumas considerações sobre tempo

- Glitch:** quando apenas uma mudança da entrada produz mais do que uma mudança na saída (variações rápidas não previstas)



09/03/2018

PML – IAC - 2018

22

Índice

- Os Blocos Combinatórios Básicos.
 - *Multiplexers*
 - Descodificadores
 - Somadores
- Algumas considerações sobre tempo
- Circuitos Sequenciais
 - Latch R-S e FlipFlops tipo D
 - Registos e Memórias
 - Máquinas e diagramas de estado

09/03/2018

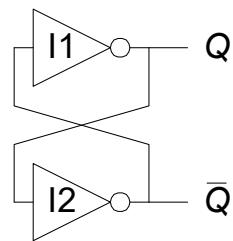
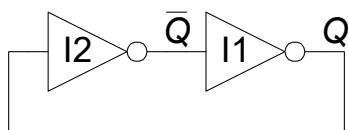
PML – IAC - 2018

23

Circuitos Sequenciais

Circuitos Sequenciais

- Circuitos sequenciais são aqueles em que a saída depende do valor atual e do valor passado das entradas.
- Mas, como pode um circuito manter informação sobre o valor passado das entradas?
- O circuito Bi-estável:



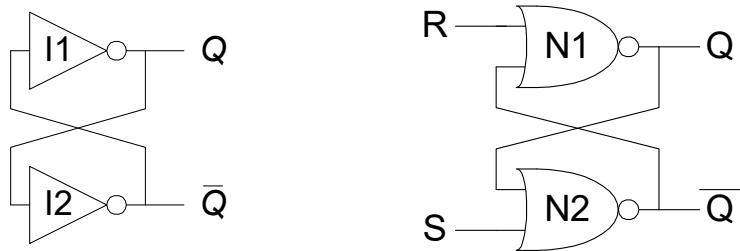
09/03/2018

PML – IAC - 2018

24

A latch SR

- O circuito bi-estável guarda um bit, mas não tem entrada que permita controlar o seu estado.
- Substituindo os NOT's por NOR's criamos duas entradas no circuito:
 - O SET e o RESET.



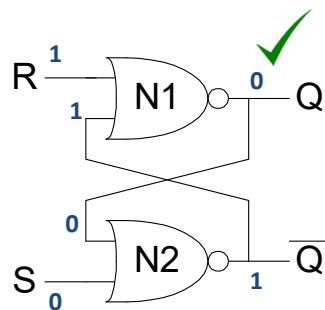
09/03/2018

PML – IAC - 2018

25

A latch SR

- Comportamento quando $R=1$ e $S=0$



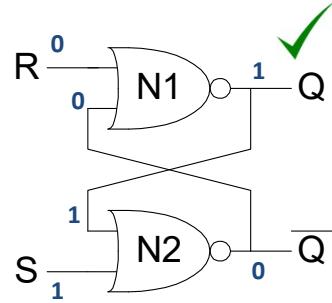
09/03/2018

PML – IAC - 2018

26

A latch SR

- Comportamento quando R=0 e S=1



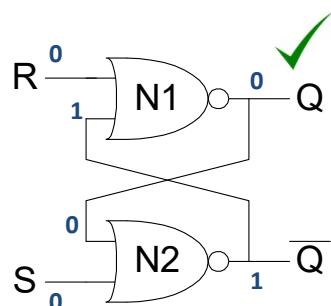
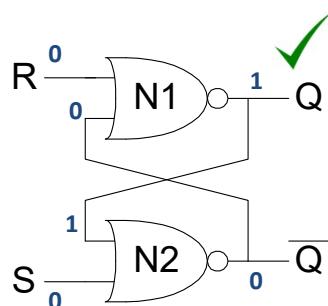
09/03/2018

PML – IAC - 2018

27

A latch SR

- Comportamento quando R=0 e S=0
 - Mantem o estado anterior!



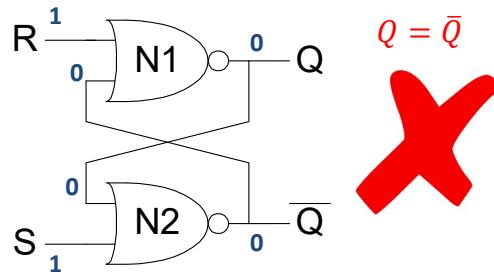
09/03/2018

PML – IAC - 2018

28

A latch SR

- Comportamento quando R=1 e S=1



09/03/2018

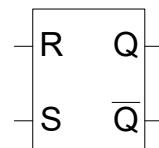
PML – IAC - 2018

29

A latch SR

- A latch SR permite guardar um bit de estado (Q)
- O valor armazenado pode ser controlado usando as entradas Set ou Reset
- **Tem um estado inválido. (S=1 e R=1)**
- Precisa de ser melhorada para evitar esse estado proibido.

SR Latch
Symbol



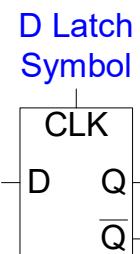
09/03/2018

PML – IAC - 2018

30

A latch D

- A latch D tem duas entradas D e CLK
 - D são os dados a escrever na latch
 - CLK determina QUANDO os dados são escritos
- Funcionamento:
 - Quando CLK está a 1 é transparente para D ($Q=D$)
 - Quando CLK está a 0 guarda o valor anterior (é opaco)
- Evita o estado inválido.

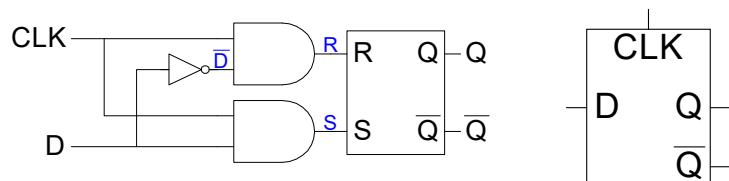


09/03/2018

PML – IAC - 2018

31

A latch D: implementação



CLK	D	\bar{D}	S	R	Q	\bar{Q}
0	X	\bar{X}	0	0	Q_{prev}	\bar{Q}_{prev}
1	0	1	0	1	0	1
1	1	0	1	0	1	0

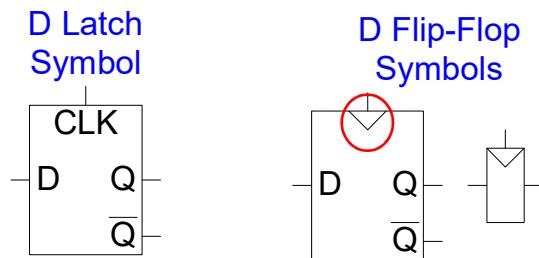
09/03/2018

PML – IAC - 2018

32

O Flip-Flop D

- O Flip-Flop D funciona como a Latch D exceto que a escrita ocorre apenas quando o relógio transita de 0 para 1.
- O flip-flop é ativo na transição (*edge-triggered*)
 - Há apenas um instante em que a escrita é efetuada.

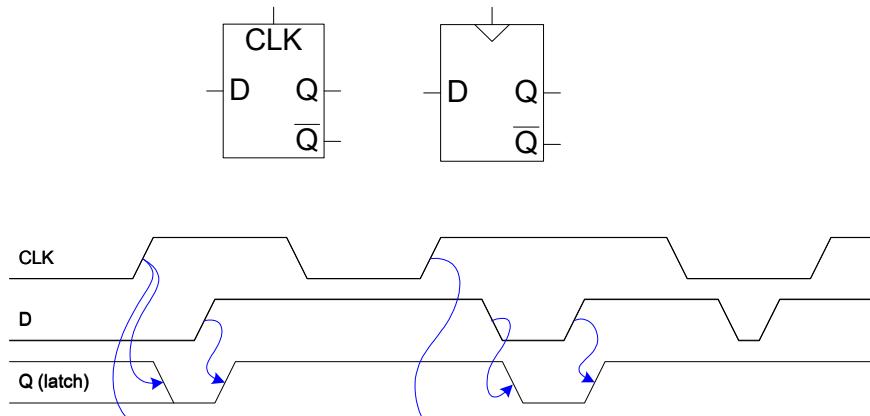


09/03/2018

PML – IAC - 2018

33

O funcionamento do Flip-Flop D vs Latch D



09/03/2018

PML – IAC - 2018

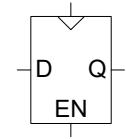
34

O Flip-Flop D

- Variantes do Flip-Flop D:

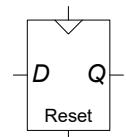
- Com enable

- controla quando são escritos dados
 - EN=0, não há escrita (mesmo que haja CLK).



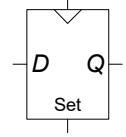
- Com reset

- Reset = 1, faz Q = 0, independentemente de D.



- Com Set

- Set = 1, faz Q = 1, independentemente de D.

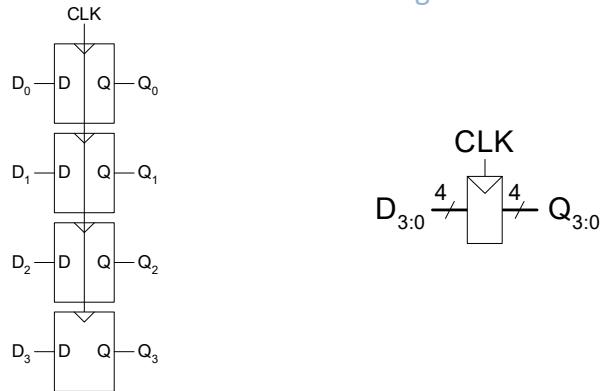


Índice

- Os Blocos Combinatórios Básicos.
 - Multiplexers
 - Descodificadores
 - Somadores
- Algumas considerações sobre tempo
- Circuitos Sequenciais
 - Latch R-S e FlipFlops tipo D
 - Registros e Memórias
 - Máquinas e diagramas de estado

Registros

- Um Flip-Flop D, consegue armazenar um bit.
- Para armazenar N bits podemos agrupar N flip-flops D.
 - Chamamos a essa estrutura um registo de N bits.



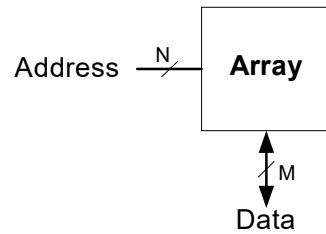
09/03/2018

PML – IAC - 2018

37

Memórias

- Armazemam de modo eficiente grandes quantidades de dados.
- Os três tipos mais comuns são:
 - DRAM: Dynamic Random Access Memory
 - SRAM: Static Random Access Memory
 - ROM: Read Only Memory
- As memórias RAM são **voláteis**.
- As memórias ROM são **não voláteis**.



09/03/2018

PML – IAC - 2018

38

Tipos de Memórias

- RAM: Random Access Memory
 - Volátil: perde a informação quando é desligada.
 - Pode ser lida e escrita rapidamente.
 - Chama-se Random Access Memory, porque se pode aceder com igual facilidade a qualquer posição de memória, ao contrário das memórias de acesso sequencial como as cassetes de fita.
- ROM: Read Only Memory
 - Não volátil: mantém a informação mesmo quando é desligada.
 - O acesso para leitura é rápido.
 - Acesso para escrita é impossível ou lento.
 - Chama-se Read Only Memory, porque as primeiras memórias de facto eram programadas ainda em fábrica ou escritas por um processo destrutivo.
 - Atualmente isto já não é verdade com a tecnologia Flash Memory.

09/03/2018

PML – IAC - 2018

39

Um visionário: Robert Dennard

- Inventou a DRAM em 1966 na IBM
- Enfrentou o ceticismo dos colegas que não acreditavam que o processo funcionaria.
- Por volta de 1975 a DRAM estava em “todos” os computadores.



09/03/2018

PML – IAC - 2018

40

Outro: Fujio Masuoka

- Trabalhou na Toshiba em circuitos rápidos e memórias.
- De forma não autorizada conduziu um projeto à noite e aos fins de semana no final da década de 70.
- O processo de limpar a memória pareceu-lhe semelhante ao flash de uma máquina.
- A Toshiba demorou a comercializar.
- A Intel lançou a memória flash no mercado em 1988.
- Atualmente é um negócio de 25 mil milhões de dólares por ano.



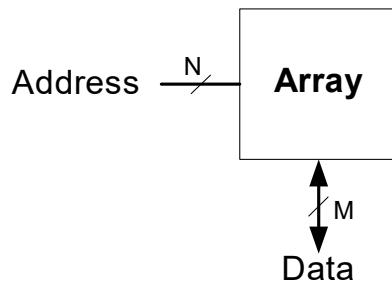
09/03/2018

PML – IAC - 2018

41

Arrays de Memória

- A memória organiza-se como um *array* bidimensional de células de 1 bit.
- Com N bits de endereço e M bits de dados, a memória tem:
 - Número de Palavras (Depth): 2^N linhas
 - Número de bits de cada palavra (Width): M colunas
 - Tamanho: Depth x Width = $2^N \times M$



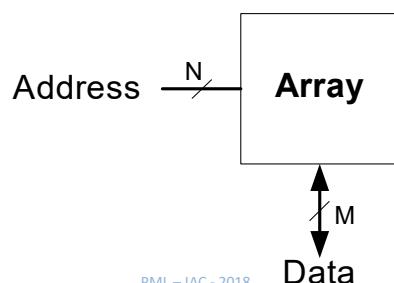
09/03/2018

PML – IAC - 2018

42

Arrays de Memória

- Espaço de endereçamento vs Endereçabilidade
- Espaço de endereçamento
 - Número de endereços possíveis = 2^N = número de palavras.
- Endereçabilidade
 - Número de bits em cada Endereço = M = tamanho da palavra.



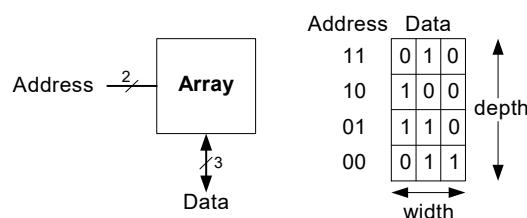
09/03/2018

PML – IAC - 2018

43

Exemplo de um Array de memória

- Um *array* de memória com 2 bits de endereço e 3 bits de dados.
 - Número de Palavras (Depth): $2^2 = 4$
 - Tamanho da palavra: 3 bits
 - É um array de $2^2 \times 3$ bits.
- A palavra armazenada no endereço 01 é 110.



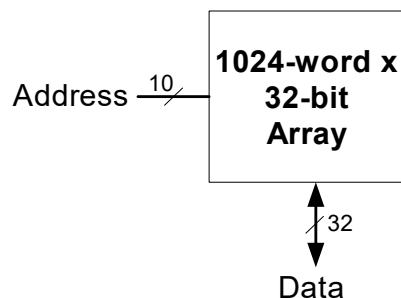
09/03/2018

PML – IAC - 2018

44

Exemplo de um Array de memória

- Um array de memória com 10 bits de endereço e 32 bits de dados.
 - Número de Palavras (Depth): $2^{10} = 1024$
 - Tamanho da palavra: 32 bits
 - É um array de 1024×32 bits = 1k x 32.



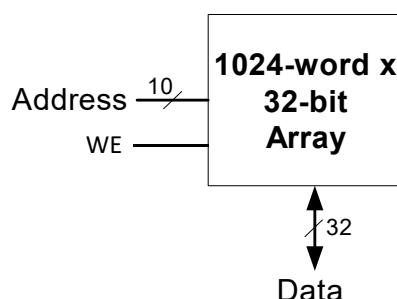
09/03/2018

PML – IAC - 2018

45

Array de memória

- Operações de leitura vs Operações de Escrita.
- Como distinguir?
 - WE = Write Enable
- Duas operações possíveis:
 - WE = 1 -> Escrita
 - O valor das linhas Data é guardado no endereço Address
 - WE = 0 -> Leitura
 - O valor guardado no endereço Address é colocado nas linhas Data.



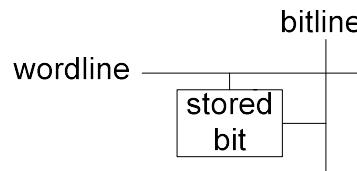
09/03/2018

PML – IAC - 2018

46

Células de memória

- Uma célula de memória é a unidade básica capaz de armazenar um bit
 - A célula de memória tem pelo menos dois interfaces
 - Um para saber quando é endereçada
 - Um para receber/transmitir a informação
 - Como a memória está organizada em “palavras” o endereçamento é comum a todos os bits de cada palavra: **wordline**.
 - **wordline** funciona como um enable, i.e., seleciona a linha a ser acedida.
 - **Só pode haver uma wordline ativa de cada vez.**



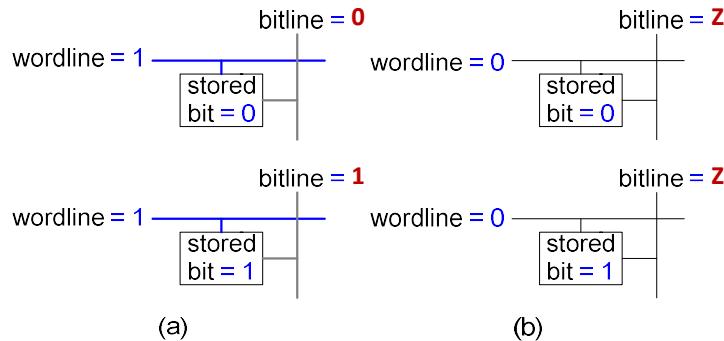
09/03/2018

PML – IAC - 2018

47

Funcionamento da Célula de memória

- Quando a *wordline* está ativa o conteúdo da célula passa para a *bitline*.
 - Quando a *wordline* está inativa a célula não interfere com a *bitline*.
 - Está em alta impedância ou *tristate*.



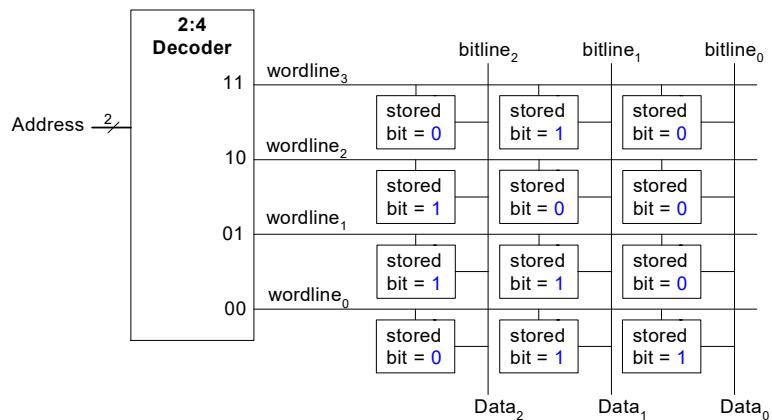
09/03/2018

PML – IAC - 2018

48

Organização da memória

- Um array de memória com dois bits de endereço e 3 bits de dados, pode esquematizar-se como:



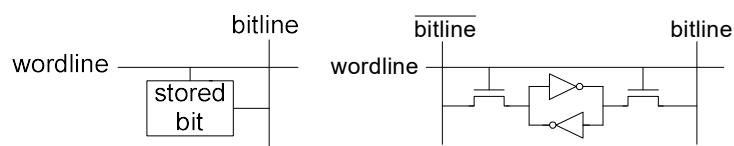
09/03/2018

PML – IAC - 2018

49

RAM estática vs Dinâmica

- SRAM: Static RAM**
 - O bit é guardado em dois inversores acoplados.
 - O valor é mantido enquanto o circuito estiver alimentado, por isso se chama **estática**.
 - Cada célula de memória precisa de um circuito complexo.



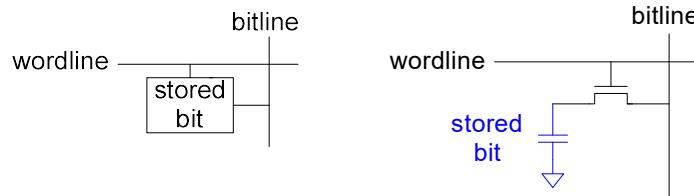
09/03/2018

PML – IAC - 2018

50

RAM estática vs Dinâmica

- DRAM: Dynamic RAM
 - O bit é guardado como uma carga num condensador.
 - O condensador tem perdas de carga (ao fim de algum tempo é impossível distinguir um 0 de um 1);
 - A leitura da memória é destrutiva.
 - Chama-se **dinâmica** porque o valor precisa de ser re-escrito periodicamente e também sempre que é lido.



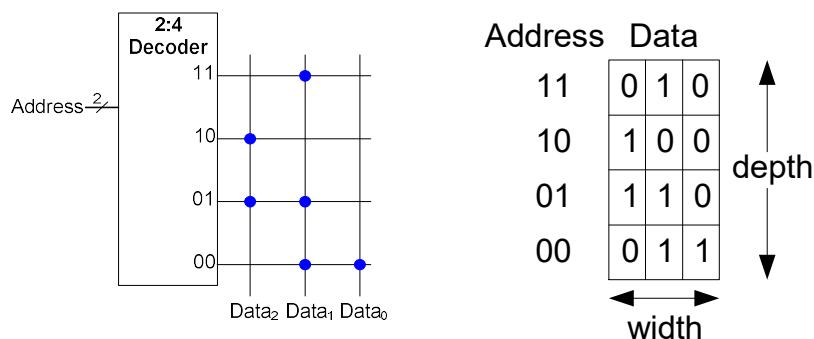
09/03/2018

PML – IAC - 2018

51

Memória ROM

- Notação Pontual:
 - O descodificador opera como um gerador de mintermos.
 - Cada coluna faz o OR dos mintermos selecionados assinalando-os com os pontos.



09/03/2018

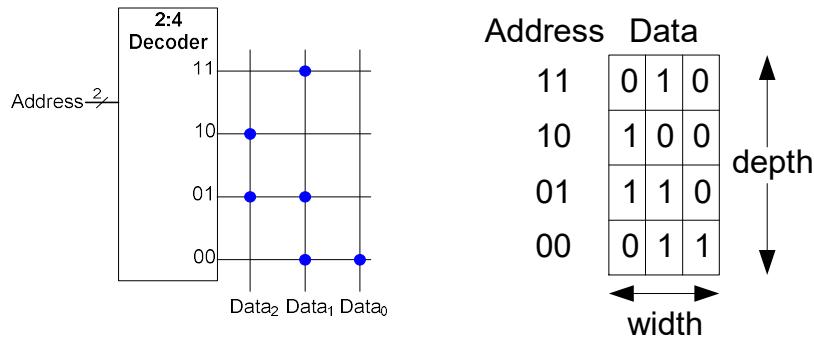
PML – IAC - 2018

52

Implementações Lógicas com ROM

- As saídas Data₂...Data₀ implementam as seguintes funções lógicas:

$$\begin{aligned}
 - Data_2 &= (A_1 \cdot \overline{A}_0 + \overline{A}_1 \cdot A_0) = A_1 \oplus A_0 \\
 - Data_1 &= (A_1 A_0 + \overline{A}_1 A_0 + \overline{A}_1 \cdot \overline{A}_0) = \overline{A}_1 + A_0 \\
 - Data_0 &= \overline{A}_1 \cdot \overline{A}_0
 \end{aligned}$$



09/03/2018

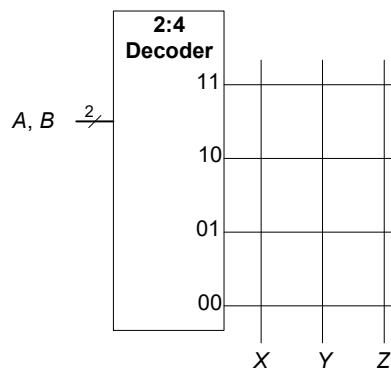
PML – IAC - 2018

53

Implementações Lógicas com ROM

- Exercício:
- Implementar as funções lógicas seguintes usando um array de memória 2²x3:

$$\begin{aligned}
 - X &= AB \\
 - Y &= A + B \\
 - Z &= A\bar{B}
 \end{aligned}$$



09/03/2018

PML – IAC - 2018

54

Índice

- Os Blocos Combinatórios Básicos.
 - *Multiplexers*
 - Descodificadores
 - Somadores
- Algumas considerações sobre tempo
- Circuitos Sequenciais
 - Latch R-S e FlipFlops tipo D
 - Registos e Memórias
 - Máquinas e diagramas de estado

09/03/2018

PML – IAC - 2018

55

Circuitos Sequenciais: Máquinas e Diagramas de Estado

Máquinas e Diagramas de Estado

- Uma máquina de estados é um circuito que:
 - Tem vários estados possíveis;
 - O estado atual é armazenado num registo (conjunto de flip-flops);
 - O estado muda na transição do relógio;
 - O sistema é sincronizado pelo relógio
- Regras de composição de circuitos sequenciais:
 - Cada elemento do circuito é um registo ou um circuito combinatório
 - Pelo menos um elemento do circuito é um registo
 - Todos os registos partilham o mesmo relógio
 - Todos os caminhos cíclicos têm de conter pelo menos um registo

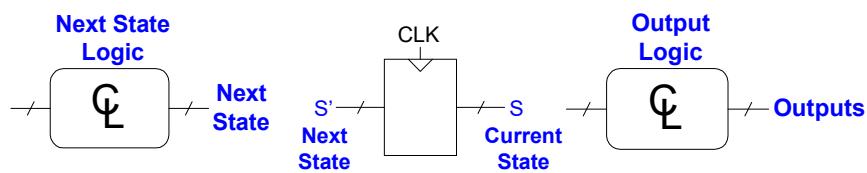
09/03/2018

PML – IAC - 2018

56

Máquinas e Diagramas de Estado

- Um registo de estado:
 - Guarda o estado atual
 - Carrega o próximo estado na transição do relógio
- Lógica combinatória:
 - Calcula o próximo estado
 - Calcula as saídas



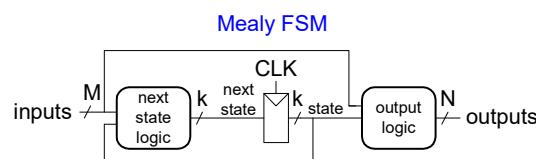
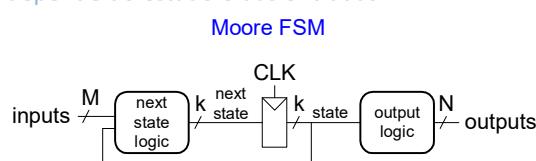
09/03/2018

PML – IAC - 2018

57

Máquinas e Diagramas de Estado

- Máquinas de estado síncronas (Máquinas de Moore):
 - A saída depende apenas do estado atual.
 - i.e., só há mudanças na saída quando há transições de estado.
- Máquinas de estado assíncronas (Máquinas de Mealy):
 - A saída depende do estado e das entradas.



09/03/2018

PML – IAC - 2018

58

Máquinas e Diagramas de Estado

- Procedimento de projeto de Máquinas de Estados (Moore)
 1. Identificar entradas e saídas.
 2. Desenhar o diagrama de estados e transições.
 3. Escrever a tabela de transição de estados.
- Em função do estado atual e das entradas.
 4. Escrever a tabela das saídas (em função do estado).
 5. Escrever (e simplificar) as equações booleanas:
- da lógica do próximo estado
- da lógica de saída.
 6. Desenhar o esquema do circuito.

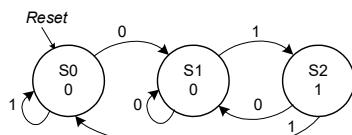
09/03/2018

PML – IAC - 2018

59

Exercício: Detetor de sequências

- Exercício:
- A Alice quer implementar um detetor de sequências que recebe como entrada 0's e 1's e ativa a saída sempre que detetar a sequência 01.
- 1º identificar entradas e saídas:
 - Entradas: o bit da sequência
 - Saída: 1 se as entradas anteriores foram 01, 0 nos outros casos.
- 2º desenhar o diagrama de estados e transições



State	Encoding
S0	00
S1	01
S2	10

09/03/2018

PML – IAC - 2018

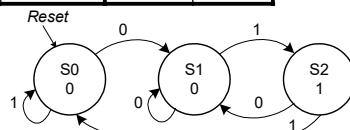
60

Exercício: Detetor de sequências

- 3º escrever a tabela de transições de estados:

Current State		Inputs A	Next State	
S_1	S_0		S'_1	S'_0
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		

State	Encoding
S0	00
S1	01
S2	10



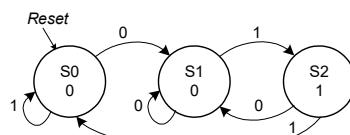
09/03/2018

61

Exercício: Detetor de sequências

- 4º escrever a tabela das saídas em função do estado:

Current State		Output Y
S_1	S_0	
0	0	
0	1	
1	0	



09/03/2018

PML – IAC - 2018

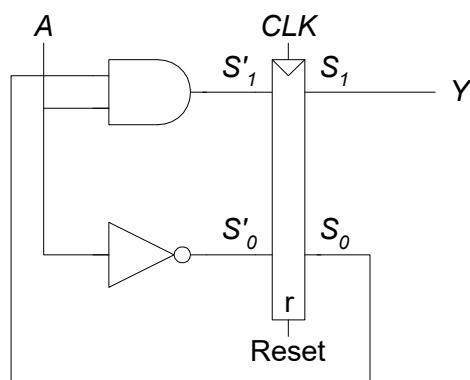
62

Exercício: Detetor de sequências

- 5º escrever as equações booleanas da lógica de saída e da lógica do próximo estado:

- $S'_0 = \dots$
- $S'_1 = \dots$
- $Y = \dots$

- 6º desenhar o circuito



09/03/2018

PML – IAC - 2018

63

Máquinas e Diagramas de Estado

- Exercício:
- Projetar uma máquina de estados que implemente um contador módulo 4.
- A sequência de contagem é 0, 1, 2, 3, 0, 1, 2, 3, 0, ...
- Esta máquina de estados tem duas particularidades:
 - A saída atual é igual ao estado atual
 - O próximo estado apenas depende do estado atual

09/03/2018

PML – IAC - 2018

64