



**Universidade Federal de Santa Catarina**  
**Centro Tecnológico**  
Departamento de Informática e Estatística  
Curso de Graduação em Ciências da Computação



# **Lógica Programável**

**INE 5348**

## **Aula 6**

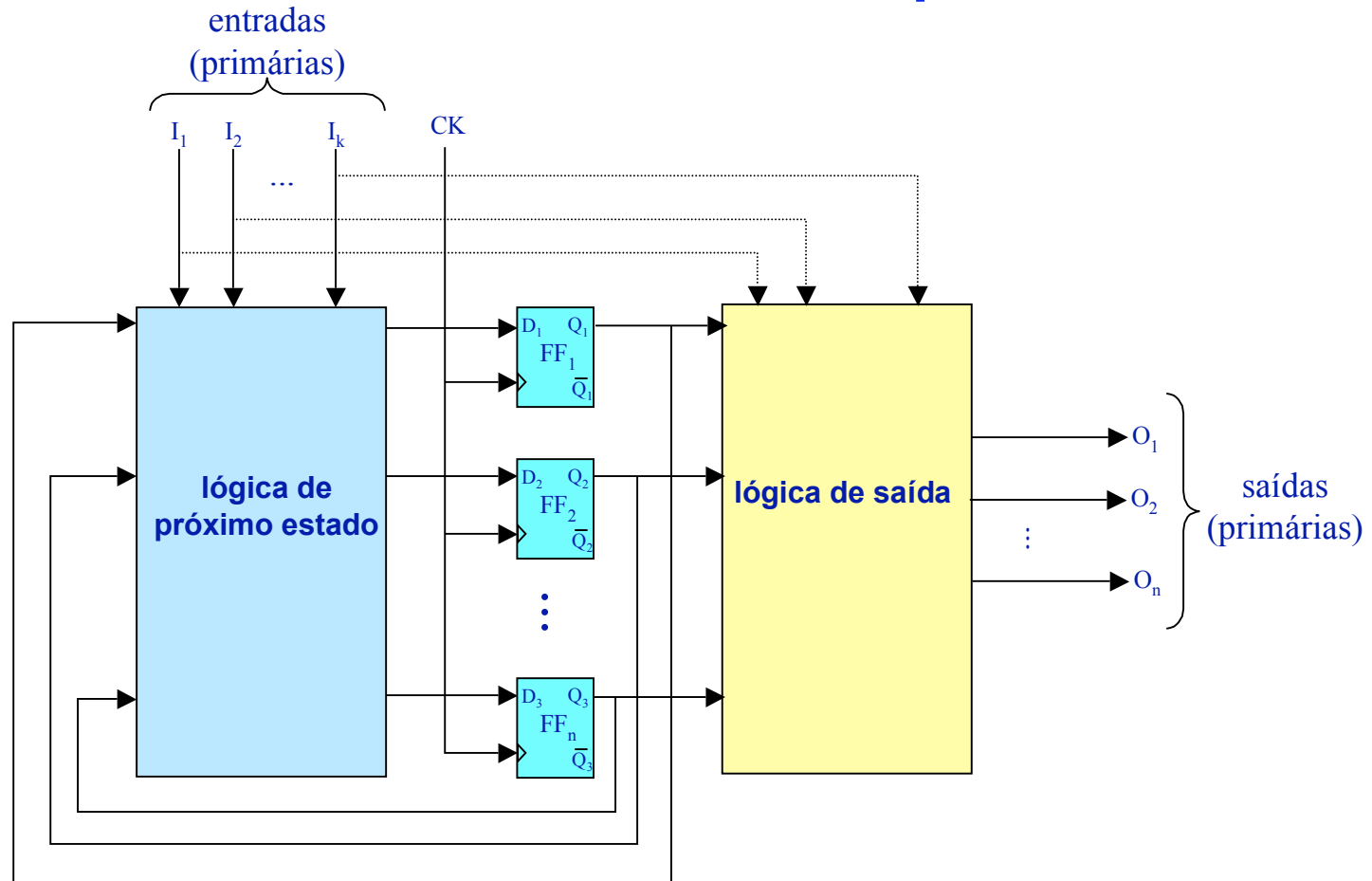
**Máquinas Seqüências Síncronas: Síntese de circuitos seqüenciais. Modelos de Moore e de Mealy. Minimização e codificação de estados. Descrição de circuitos seqüenciais em VHDL, síntese e simulação.**

**Prof. José Luís Güntzel**  
**[guntzel@inf.ufsc.br](mailto:guntzel@inf.ufsc.br)**

**[www.inf.ufsc.br/~guntzel/ine5348/ine5348.html](http://www.inf.ufsc.br/~guntzel/ine5348/ine5348.html)**

# Máquinas Seqüenciais Síncronas

## ► Diagrama de Blocos de um Circuito Seqüencial Síncrono



# Máquinas Seqüenciais Síncronas

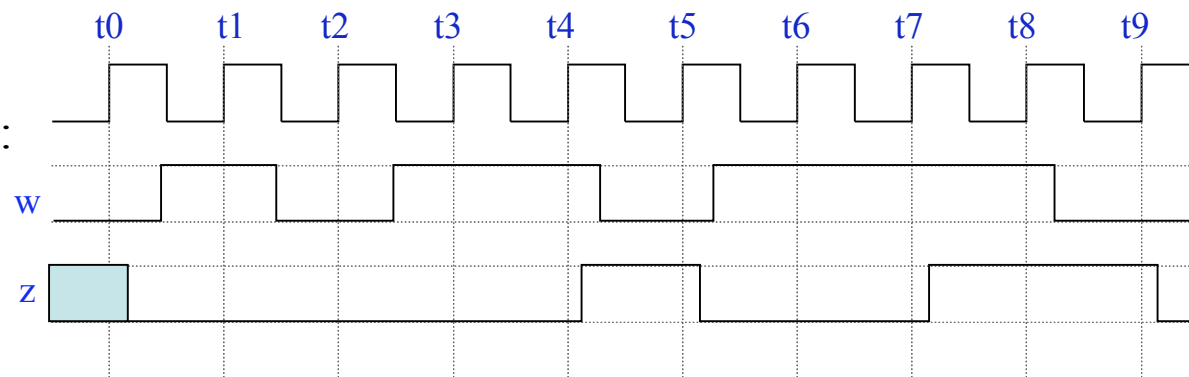
## ► Síntese de Circuitos Seqüenciais

### Exemplo 2:

Projete um circuito que satisfaça às seguintes especificações:

1. O circuito possui uma entrada, **w**, e uma saída, **z**.
2. Todas as mudanças de valores no circuito ocorrem na borda de subida do sinal de relógio.
3. Quando o circuito detectar que a entrada **w** vale “0”, a saída **z** deve valer “0” no ciclo de relógio seguinte. Porém, quando o circuito detectar que a entrada **w** vale “1” durante duas bordas de relógio consecutivas, a saída **z** deve passar a valer “1” no ciclo de relógio seguinte à segunda ocorrência do valor “1”. As mudanças de **z** estão sincronizadas com a borda de relógio ativa.

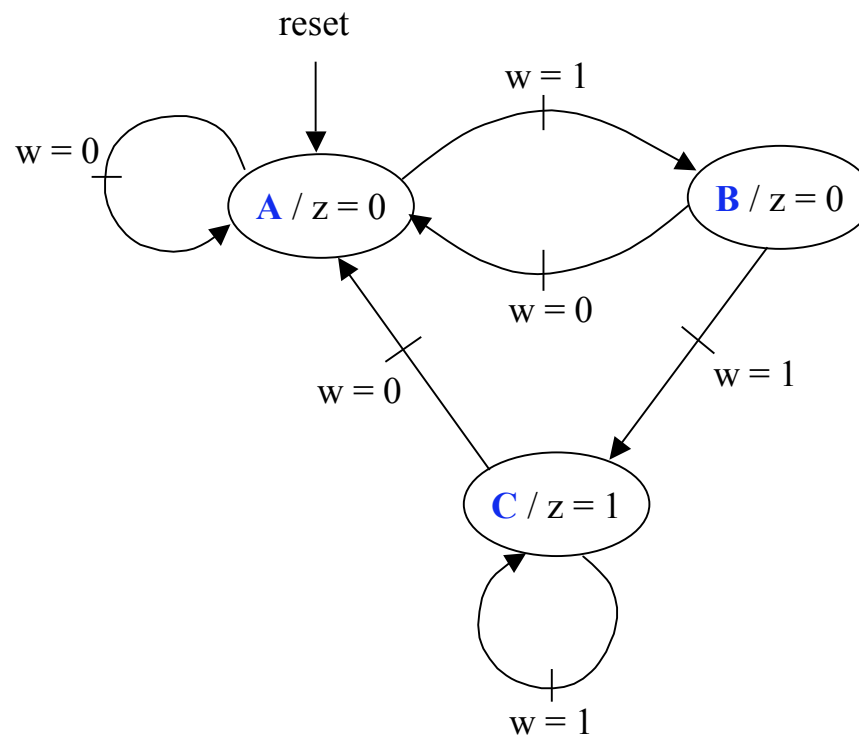
Exemplo de funcionamento:



# Máquinas Seqüenciais Síncronas

## ► Síntese de Circuitos Seqüenciais

### Exemplo 2: Diagrama de Estados

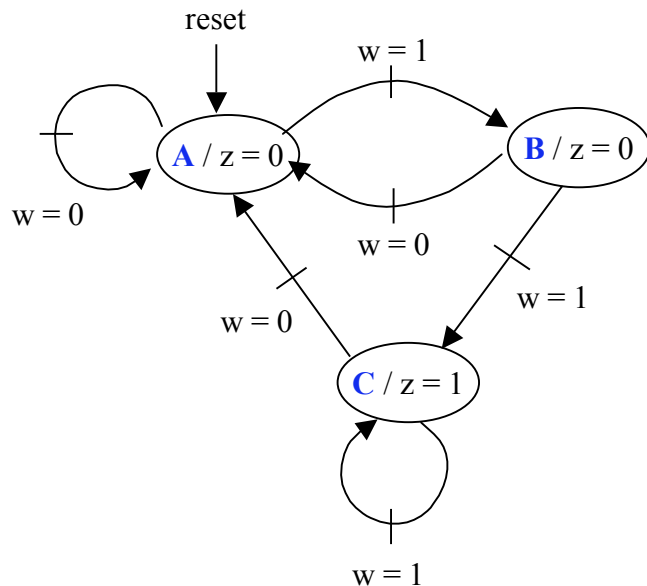


# Máquinas Seqüenciais Síncronas

## ► Síntese de Circuitos Seqüenciais

### Exemplo 2: Tabelas de (Transição de) Estados e de Saída

Tabela de (transição de) estados



Estado atual	w	Próximo estado
A	0	A
A	1	B
B	0	A
B	1	C
C	0	A
C	1	C

Tabela de saída

Estado	z
A	0
B	0
C	1

# Máquinas Seqüenciais Síncronas

## ► Síntese de Circuitos Seqüenciais

### Exemplo 2: Diagrama de Blocos

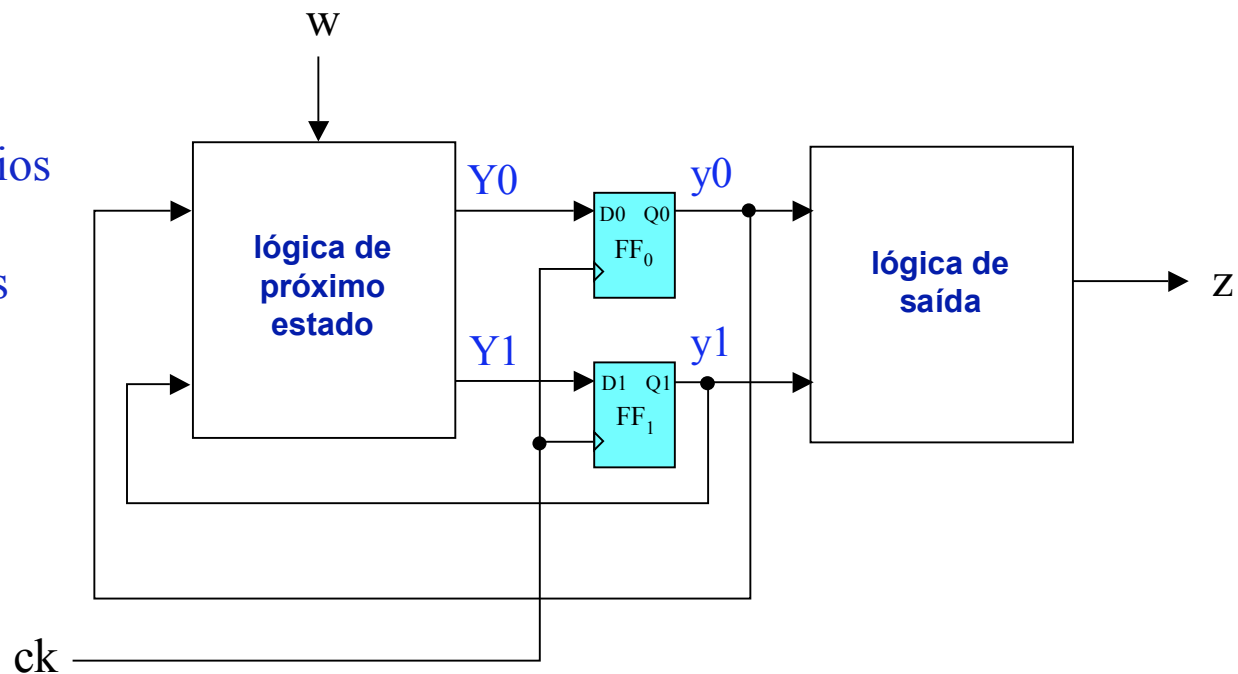
Sinais de entrada:  $w$

Sinais de saída:  $z$

Número de estados: 3

⇒ Logo, são necessários

2 flip-flops para  
armazenar as variáveis  
de estado



# Máquinas Seqüenciais Síncronas

## ► Síntese de Circuitos Seqüenciais

### Exemplo 2: Assinalamento de Estados

Supondo o seguinte assinalamento: A=00, B=01, C=10

Estado atual	w	Próximo estado
A	0	A
A	1	B
B	0	A
B	1	C
C	0	A
C	1	C



	Estado atual y1y0	w	Próximo estado Y1Y0	
A	00	0	00	A
A	00	1	01	B
B	01	0	00	A
B	01	1	10	C
C	10	0	00	A
C	10	1	10	C
-	11	0	XX	-
-	11	1	XX	-

# Máquinas Seqüenciais Síncronas

## ► Síntese de Circuitos Seqüenciais

### Exemplo 2: Projetando a lógica de próximo estado

y1y0	w	Y1Y0
00	0	00
00	1	01
01	0	00
01	1	10
10	0	00
10	1	10
11	0	XX
11	1	XX

Y1	$\bar{y}1\bar{y}0$	$\bar{y}1y0$	$y1y0$	$y1\bar{y}0$
$\bar{w}$	0	0	X	0
w	0	1	X	1

$w \cdot y0$  (points to the '1' in row w, column  $\bar{y}1y0$ )  
 $w \cdot y1$  (points to the '1' in row w, column  $y1\bar{y}0$ )

$$Y1 = w \cdot y1 + w \cdot y0$$

$$= w (y1 + y0)$$

Y0	$\bar{y}1\bar{y}0$	$\bar{y}1y0$	$y1y0$	$y1\bar{y}0$
$\bar{w}$	0	0	X	0
w	1	0	X	0

$w \cdot \bar{y}1 \cdot \bar{y}0$  (points to the '1' in row w, column  $\bar{y}1\bar{y}0$ )

$$Y0 = w \cdot \bar{y}1 \cdot \bar{y}0$$



# Máquinas Seqüenciais Síncronas

## ► Síntese de Circuitos Seqüenciais

### Exemplo 2: Projetando a lógica de saída

	Estado $y_1y_0$	$z$
A	00	0
B	01	0
C	10	1
	11	X

$z$	$\bar{y}_0$	$y_0$
$\bar{y}_1$	0	0
$y_1$	1	X

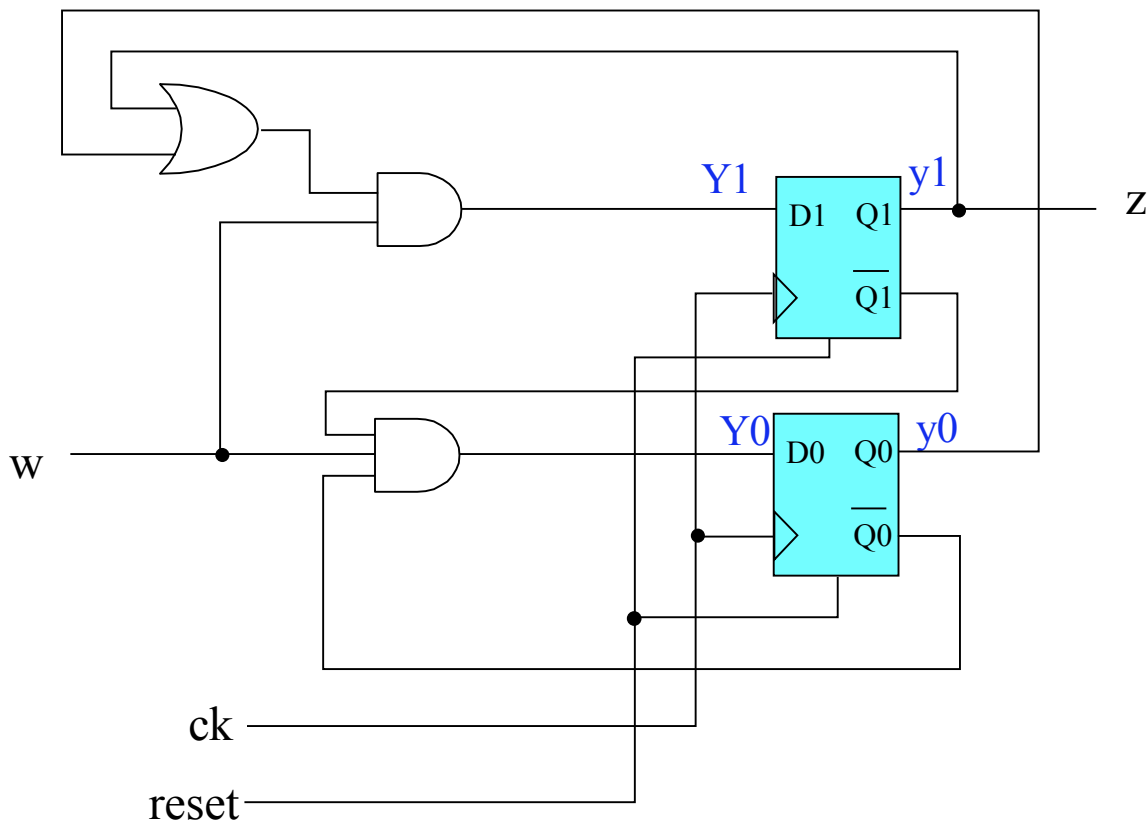
A red box highlights the cell containing '1' at the intersection of  $y_1$  and  $\bar{y}_0$ . A red line points from this cell to the label  $y_1$  below the table.

$$z = y_1$$

# Máquinas Seqüenciais Síncronas

## ► Síntese de Circuitos Seqüenciais

### Exemplo 2: circuito final

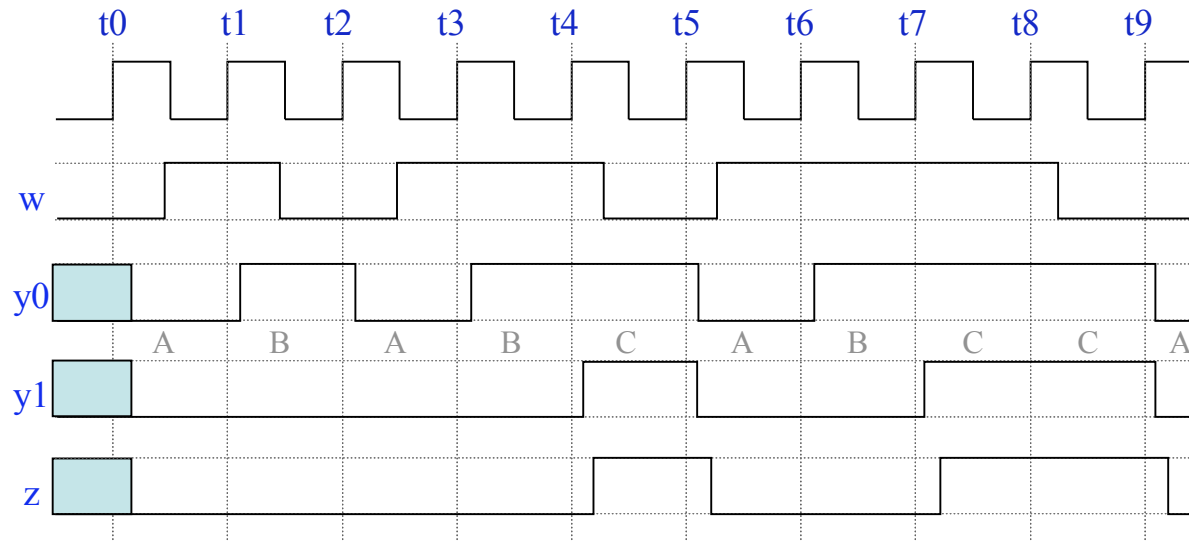


$$\begin{aligned} Y1 &= w \cdot y1 + w \cdot y0 \\ &= w (y1 + y0) \\ Y0 &= w \cdot \overline{y1} \cdot \overline{y0} \\ z &= y1 \end{aligned}$$

# Máquinas Seqüenciais Síncronas

## ► Síntese de Circuitos Seqüenciais

### Exemplo 2: Funcionamento dinâmico

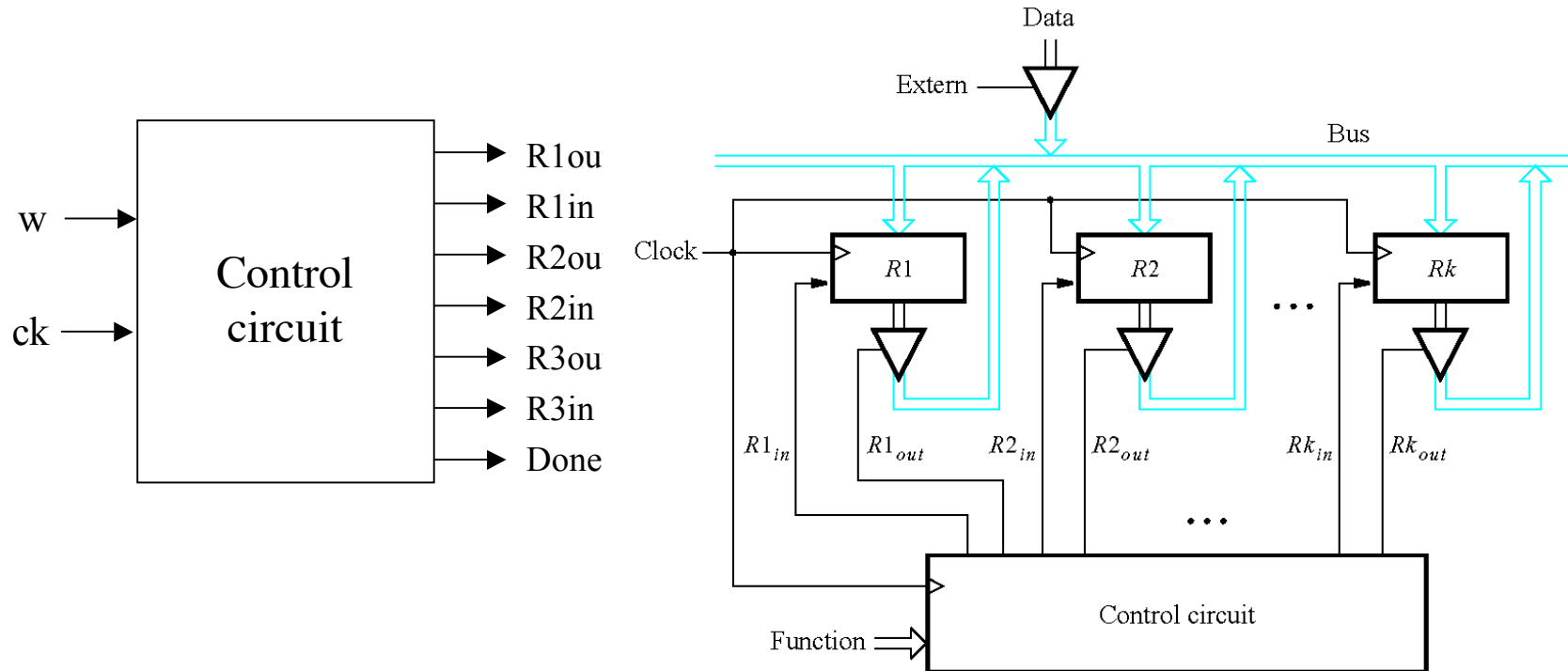


# Máquinas Seqüenciais Síncronas

## ► Síntese de Circuitos Seqüenciais

### Exemplo 3:

Projetar o circuito de controle que permita realizar um *swap* entre R1 e R2, utilizando R3 como temporário. (Desconsiderar outras possíveis operações.)



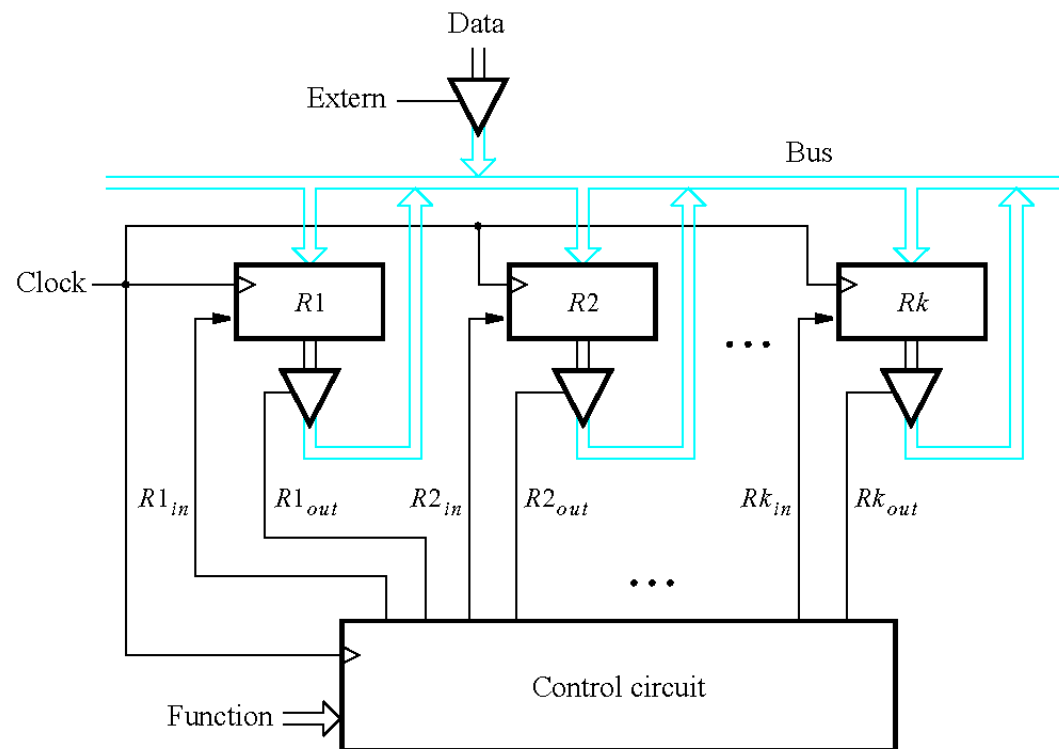
# Máquinas Seqüenciais Síncronas

## ► Síntese de Circuitos Seqüenciais

### Exemplo 3:

Passos:	Sinais de controle
$R3 \leftarrow R2;$	$R2_{out} = 1; R3_{in} = 1;$
$R2 \leftarrow R1;$	$R1_{out} = 1; R2_{in} = 1;$
$R1 \leftarrow R3;$	$R3_{out} = 1; R1_{in} = 1;$ $Done = 1;$

Obs: os sinais não citados na tabela devem valer “0”

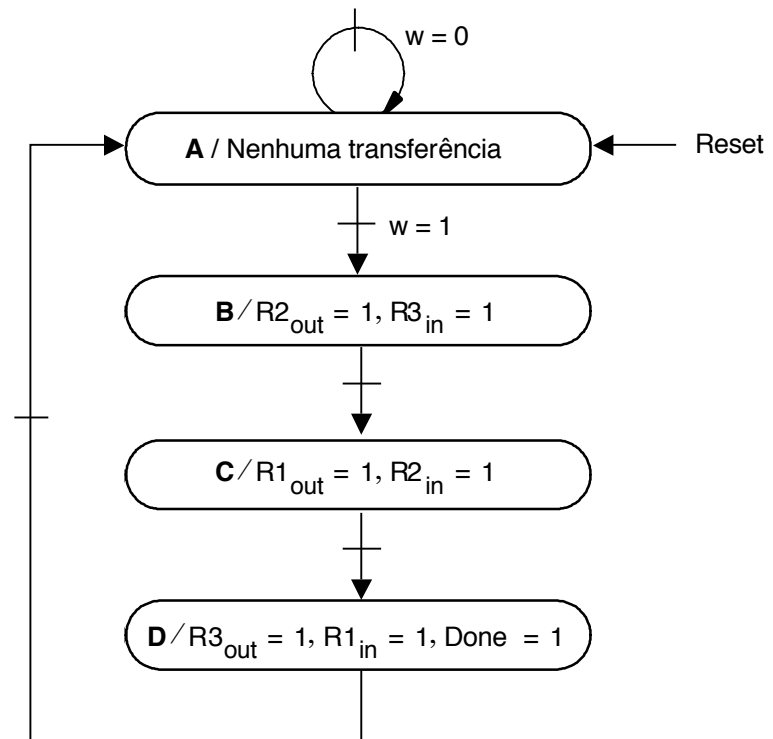


# Máquinas Seqüenciais Síncronas

## ► Síntese de Circuitos Seqüenciais

### Exemplo 3: Diagrama de estados

Passos:	Sinais de controle
$R3 \leftarrow R2;$	$R2_{out} = 1; R3_{in} = 1;$
$R2 \leftarrow R1;$	$R1_{out} = 1; R2_{in} = 1;$
$R1 \leftarrow R3;$	$R3_{out} = 1; R1_{in} = 1;$ $Done = 1;$



# Máquinas Seqüenciais Síncronas

## ► Síntese de Circuitos Seqüenciais

### Exemplo 3: Tabelas de (Transição de) Estados e de Saída

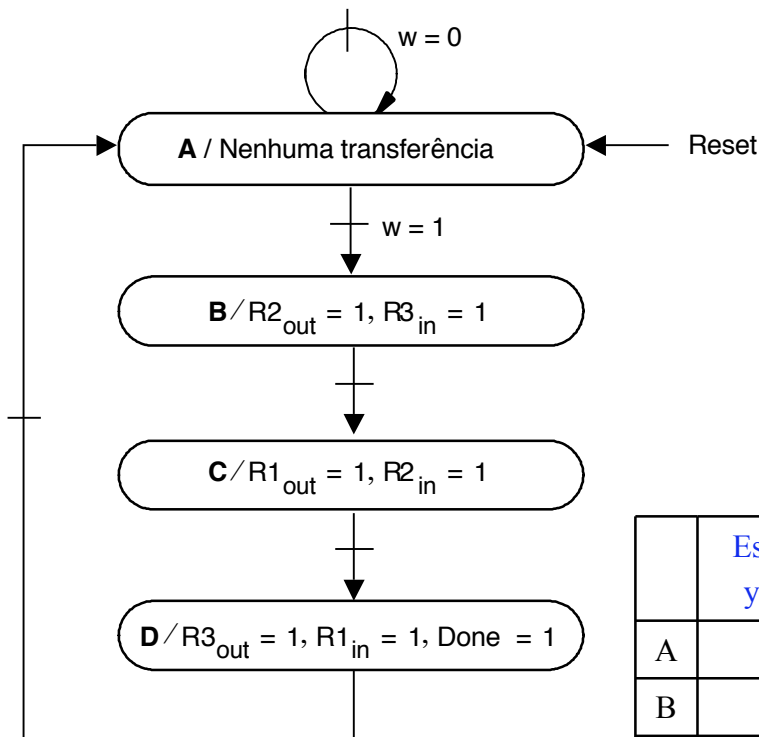


Tabela de  
(transição de)  
estados

Estado atual	w	Próximo estado
A	0	A
A	1	B
B	X	C
C	X	D
D	X	A

Tabela de saída

	Estado y1y0	R1out	R1in	R2out	R2in	R3out	R3in	Done
A	00	0	0	0	0	0	0	0
B	01	0	0	1	0	0	1	0
C	10	1	0	0	1	0	0	0
D	11	0	1	0	0	1	0	1

# Máquinas Seqüenciais Síncronas

## ► Síntese de Circuitos Seqüenciais

### Exemplo 3: Diagrama de Blocos

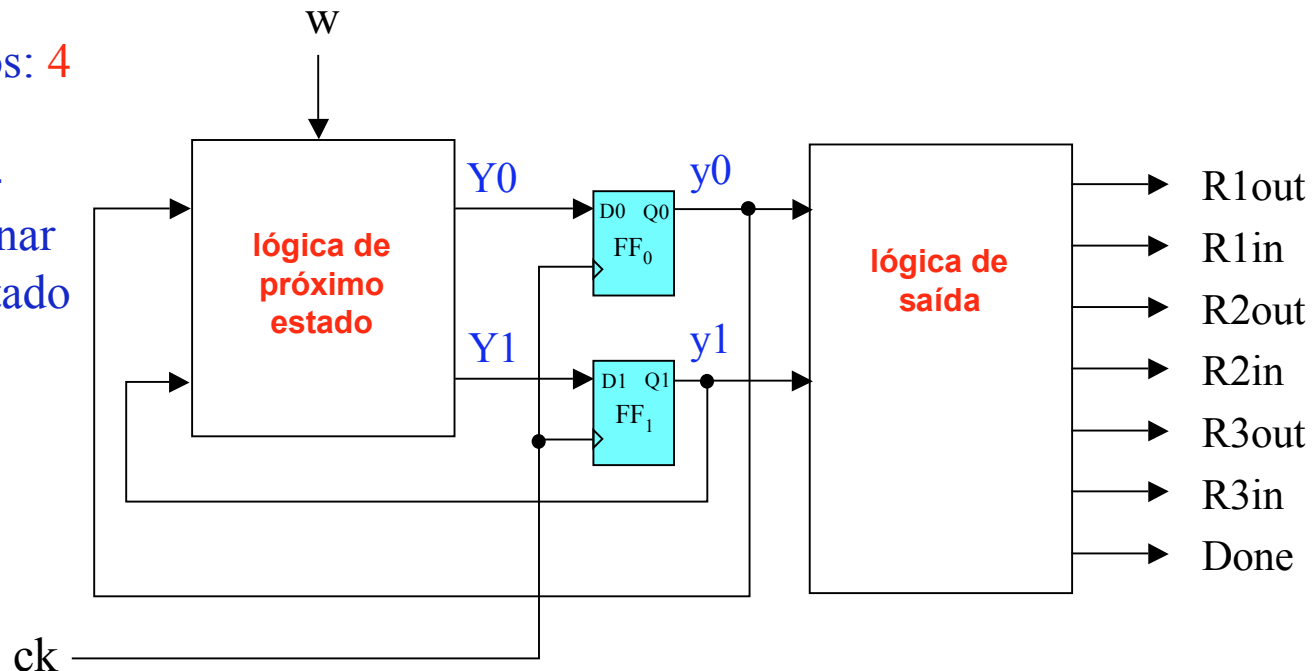
Sinais de entrada:  $w$

Sinais de saída:  $z$

Número de estados: 4

⇒ Logo, são

necessários **2** flip-flops para armazenar as variáveis de estado





# Máquinas Seqüenciais Síncronas

## ► Síntese de Circuitos Seqüenciais

### Exemplo 3: Assinalamento de Estados

Supondo o seguinte assinalamento: A=00, B=01, C=10, D=11

Estado atual	w	Próximo estado
A	0	A
A	1	B
B	X	C
C	X	D
D	X	A



	Estado atual y1y0	w	Próximo estado Y1Y0	
A	00	0	00	A
A	00	1	01	B
B	01	X	10	C
C	10	X	11	D
D	11	X	00	A

# Máquinas Seqüenciais Síncronas

## ► Síntese de Circuitos Seqüenciais

### Exemplo 3: Projetando a lógica de próximo estado

	Estado atual $y_1y_0$	w	Próximo estado $Y_1Y_0$	
A	00	0	00	A
A	00	1	01	B
B	01	X	10	C
C	10	X	11	D
D	11	X	00	A

$Y_1$	$\bar{y}_1\bar{y}_0$	$\bar{y}_1y_0$	$y_1y_0$	$y_1\bar{y}_0$
$\bar{w}$	0	1	0	1
w	0	1	0	1

$y_1 \cdot \bar{y}_0$  (points to  $y_1\bar{y}_0$  in row  $\bar{w}$ )  
 $\bar{y}_1 \cdot y_0$  (points to  $\bar{y}_1y_0$  in row w)

$$Y_1 = \bar{y}_1 \cdot y_0 + y_1 \cdot \bar{y}_0 = y_1 \oplus y_0$$

$Y_0$	$\bar{y}_1\bar{y}_0$	$\bar{y}_1y_0$	$y_1y_0$	$y_1\bar{y}_0$
$\bar{w}$	0	0	0	1
w	1	0	0	1

$y_1 \cdot \bar{y}_0$  (points to  $y_1\bar{y}_0$  in row  $\bar{w}$ )  
 $w \cdot \bar{y}_0$  (points to  $\bar{y}_1\bar{y}_0$  in row w and  $y_1\bar{y}_0$  in row w)

$$Y_0 = w \cdot \bar{y}_0 + y_1 \cdot \bar{y}_0$$

# Máquinas Seqüenciais Síncronas

## ► Síntese de Circuitos Seqüenciais

### Exemplo 3: Projetando a lógica de saída

	Estado y1y0	R1out	R1in	R2out	R2in	R3out	R3in	Done
A	00	0	0	0	0	0	0	0
B	01	0	0	1	0	0	1	0
C	10	1	0	0	1	0	0	0
D	11	0	1	0	0	1	0	1

$$R1out = R2in = y1 \cdot \overline{y0}$$

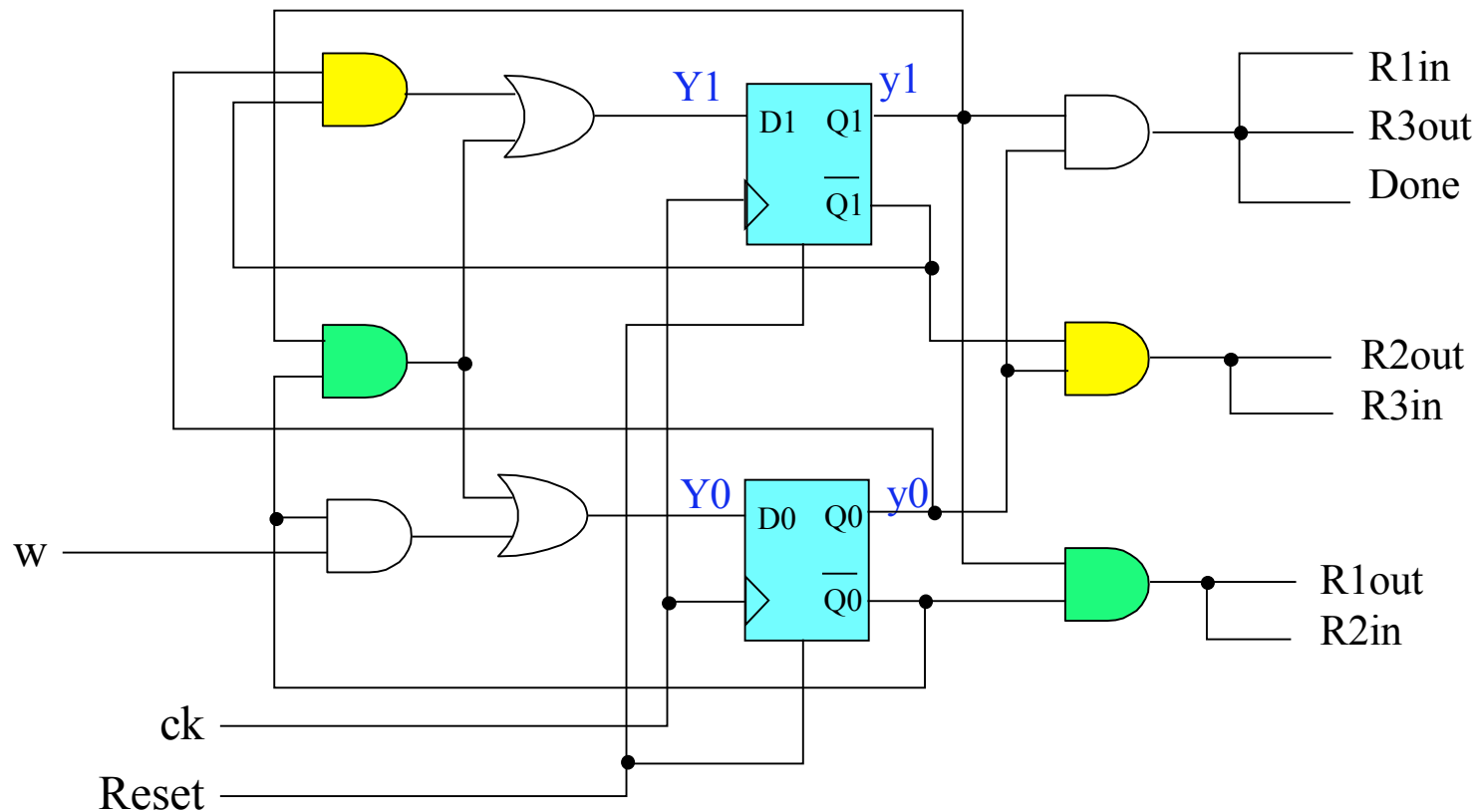
$$R1in = R3out = Done = y1 \cdot y0$$

$$R2out = R3in = \overline{y1} \cdot y0$$

# Máquinas Seqüenciais Síncronas

## ► Síntese de Circuitos Seqüenciais

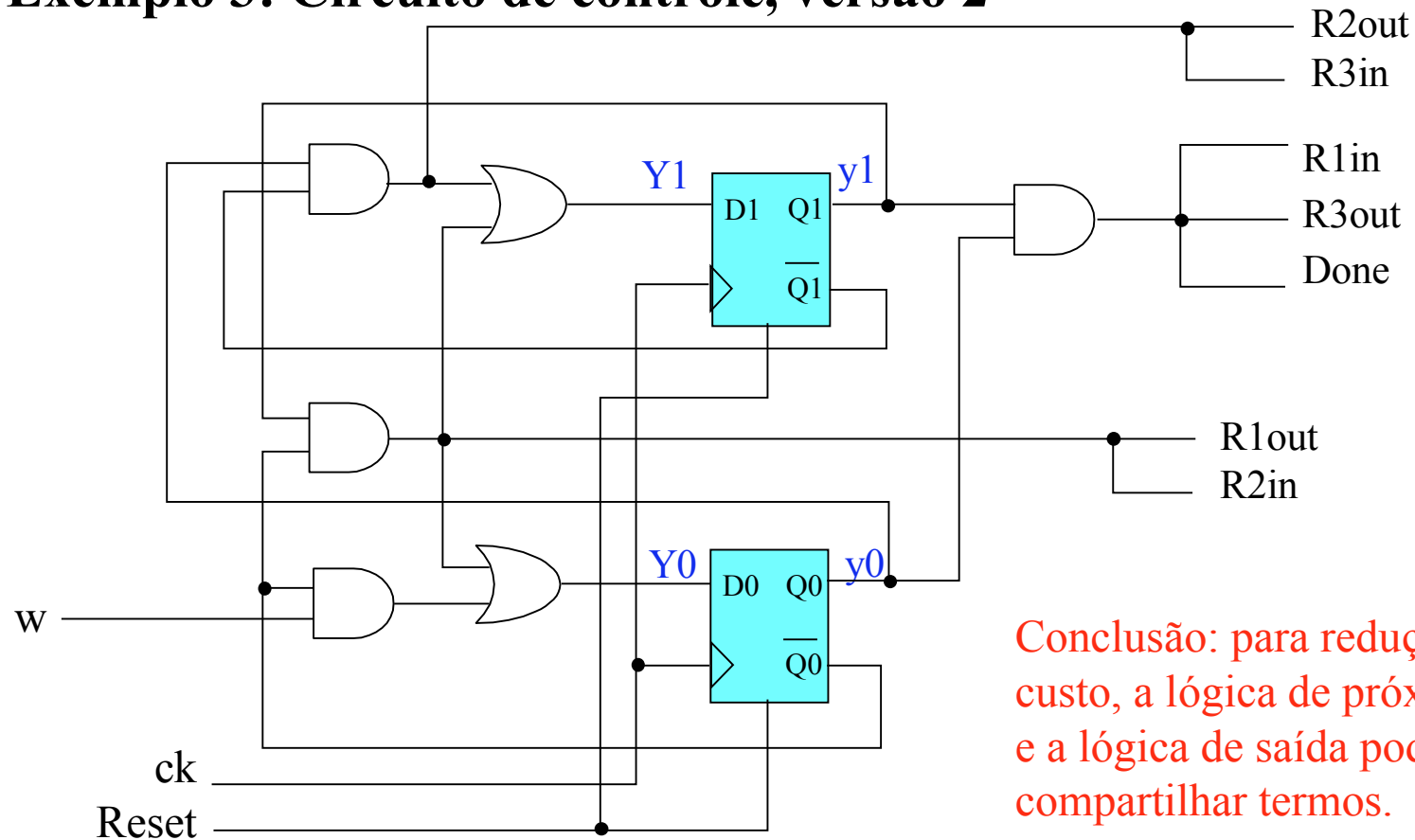
### Exemplo 3: Circuito de controle, versão 1



# Máquinas Seqüenciais Síncronas

## ► Síntese de Circuitos Seqüenciais

### Exemplo 3: Circuito de controle, versão 2



# Máquinas Seqüenciais Síncronas

## ► Minimização e Codificação de Estados

Considerando o exemplo 2, porém com o assinalamento de estados A=00, B=01, C=11

	Estado atual y1y0	w	Próximo estado Y1Y0	
A	00	0	00	A
A	00	1	01	B
B	01	0	00	A
B	01	1	11	C
-	10	0	XX	-
-	10	1	XX	-
C	11	0	00	A
C	11	1	11	C

	Estado y1y0	z
A	00	0
B	01	0
-	10	X
C	11	1

# Máquinas Seqüenciais Síncronas

## ► Minimização e Codificação de Estados

Considerando o exemplo 2, porém com o assinalamento de estados  $A=00$ ,  $B=01$ ,  $C=11$

	Estado atual $y_1y_0$	$w$	Próximo estado $Y_1Y_0$	
A	00	0	00	A
A	00	1	01	B
B	01	0	00	A
B	01	1	11	C
-	10	0	XX	-
-	10	1	XX	-
C	11	0	00	A
C	11	1	11	C

$Y_1$	$\bar{y}_1\bar{y}_0$	$\bar{y}_1y_0$	$y_1\bar{y}_0$	$y_1y_0$
$\bar{w}$	0	0	0	X
$w$	0	1	1	X

$$w \cdot y_0$$

$$Y_1 = w \cdot y_0$$

$Y_0$	$\bar{y}_1\bar{y}_0$	$\bar{y}_1y_0$	$y_1\bar{y}_0$	$y_1y_0$
$\bar{w}$	0	0	0	X
$w$	1	1	1	X

$$Y_0 = w$$

$w$

# Máquinas Seqüenciais Síncronas

## ► Minimização e Codificação de Estados

Considerando o exemplo 2, porém com o assinalamento de estados  $A=00$ ,  $B=01$ ,  $C=11$

$y_1y_0$	$z$
00	0
01	0
10	X
11	1

$z$	$\bar{y}_0$	$y_0$
$\bar{y}_1$	0	0
$y_1$	X	1

$y_1$

$$z = y_1$$

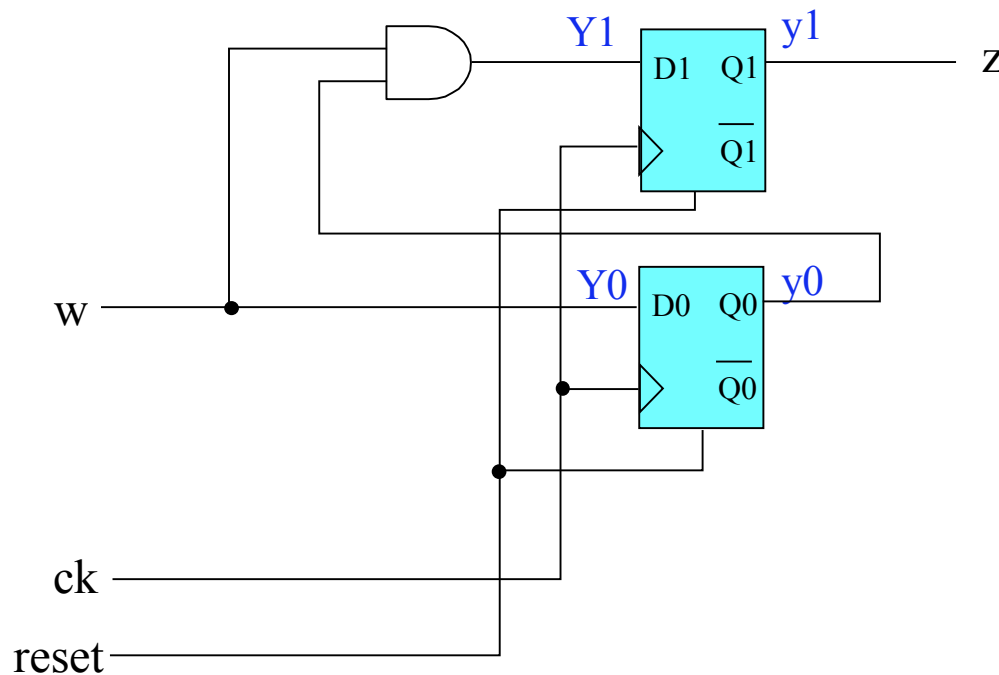
**Coincidentemente**, a equação de saída não mudou.



# Máquinas Seqüenciais Síncronas

## ► Minimização e Codificação de Estados

Considerando o exemplo 2, porém com o assinalamento de estados  
 $A=00$ ,  $B=01$ ,  $C=11$

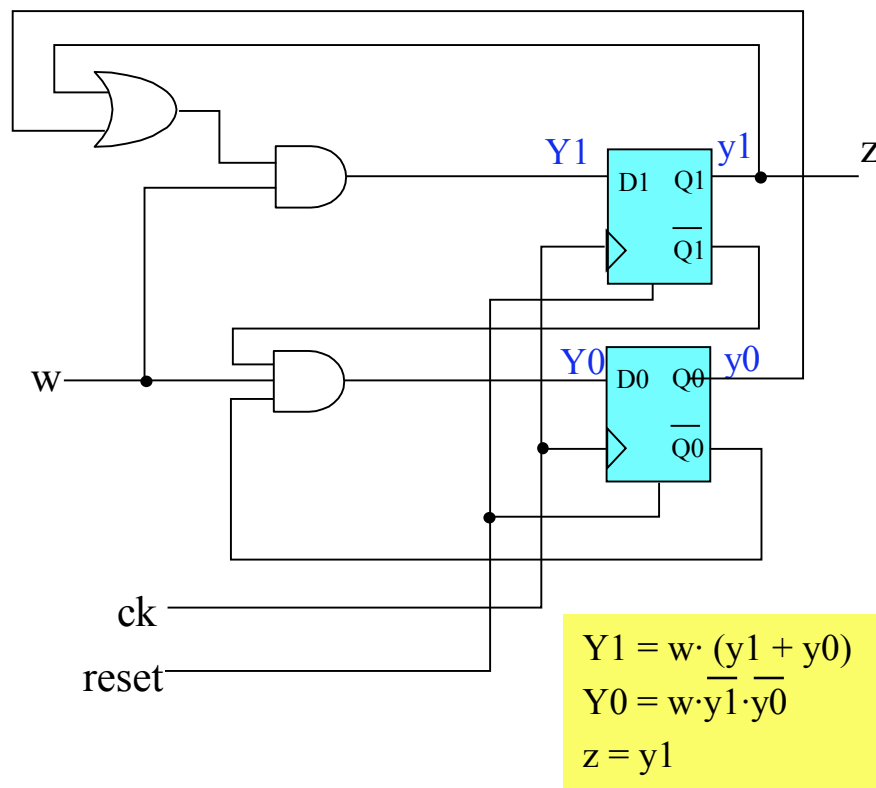


$$\begin{aligned} Y1 &= w \cdot y0 \\ Y0 &= w \\ z &= y1 \end{aligned}$$

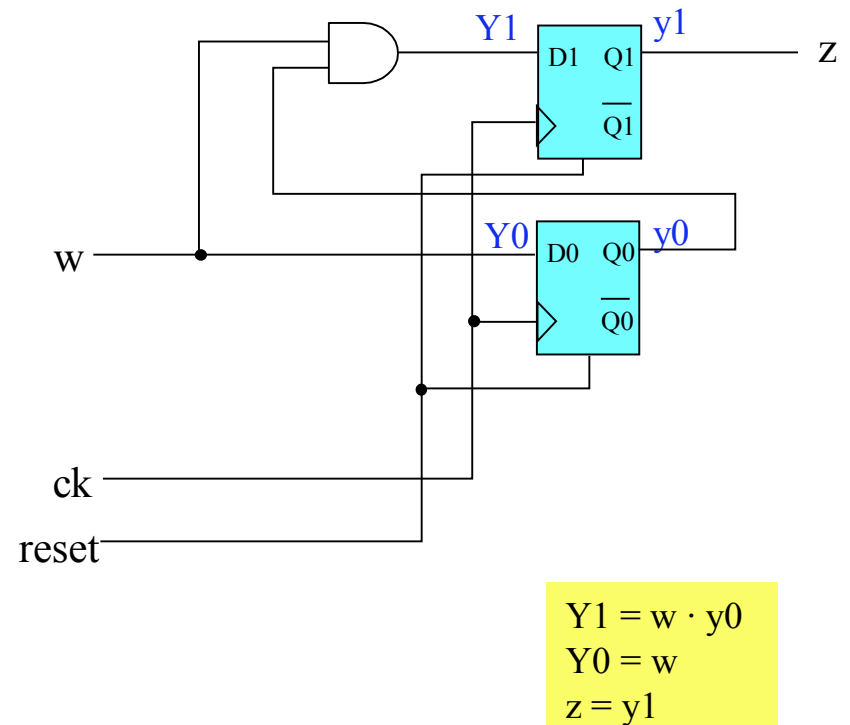
# Máquinas Seqüenciais Síncronas

## ► Minimização e Codificação de Estados

Assinalamento A=00, B=01, C=10



Assinalamento A=00, B=01, C=11



# Máquinas Seqüenciais Síncronas

## ► Minimização e Codificação de Estados

Considerando o exemplo 3, porém com o assinalamento de estados  
A=00, B=01, C=11, D=10

Estado atual	w	Próximo estado
A	0	A
A	1	B
B	X	C
C	X	D
D	X	A



	Estado atual y1y0	w	Próximo estado Y1Y0	
A	00	0	00	A
A	00	1	01	B
B	01	X	11	C
D	10	X	00	A
C	11	X	10	D

# Máquinas Seqüenciais Síncronas

## ► Minimização e Codificação de Estados

Considerando o exemplo 3, porém com o assinalamento de estados  
**A=00, B=01, C=11, D=10**

	Estado atual $y_1y_0$	w	Próximo estado $Y_1Y_0$	
A	00	0	00	A
A	00	1	01	B
B	01	X	11	C
D	10	X	00	A
C	11	X	10	D

$Y_1$	$\bar{y}_1\bar{y}_0$	$\bar{y}_1y_0$	$y_1y_0$	$y_1\bar{y}_0$
$\bar{w}$	0	1	1	0
w	0	1	1	0

$$Y_1 = y_0$$

$Y_0$	$\bar{y}_1\bar{y}_0$	$\bar{y}_1y_0$	$y_1y_0$	$y_1\bar{y}_0$
$\bar{w}$	0	1	0	0
w	1	1	0	0

$$w \cdot \bar{y}_1$$

$$\bar{y}_1 \cdot y_0$$

$$Y_0 = w \cdot \bar{y}_1 + \bar{y}_1 \cdot y_0$$

# Máquinas Seqüenciais Síncronas

## ► Minimização e Codificação de Estados

Considerando o exemplo 3, porém com o assinalamento de estados  
**A=00, B=01, C=11, D=10**

	Estado y1y0	R1out	R1in	R2out	R2in	R3out	R3in	Done
A	00	0	0	0	0	0	0	0
B	01	0	0	1	0	0	1	0
D	10	0	1	0	0	1	0	1
C	11	1	0	0	1	0	0	0

$$R1out = R2in = y1 \cdot y0$$

$$R1in = R3out = Done = y1 \cdot \overline{y0}$$

$$R2out = R3in = \overline{y1} \cdot y0$$

# Máquinas Seqüenciais Síncronas

## ► Minimização e Codificação de Estados

Assinalamento A=00, B=01, C=11, D=10

$$Y1 = \overline{y1} \cdot y0 + y1 \cdot \overline{y0}$$

$$Y0 = w \cdot \overline{y0} + y1 \cdot \overline{y0}$$

$$R1_{out} = R2_{in} = y1 \cdot y0$$

$$R1_{in} = R3_{out} = Done = y1 \cdot y0$$

$$R2_{out} = R3_{in} = y1 \cdot y0$$

Assinalamento A=00, B=01, C=10, D=11

$$Y1 = y0$$

$$Y0 = w \cdot \overline{y1} + \overline{y1} \cdot y0$$

$$R1_{out} = R2_{in} = y1 \cdot y0$$

$$R1_{in} = R3_{out} = Done = y1 \cdot \overline{y0}$$

$$R2_{out} = R3_{in} = \overline{y1} \cdot y0$$

# Máquinas Seqüenciais Síncronas

## ► Codificação de Estados

### Codificação “One-Hot”

- **Consiste em usar tantas variáveis de estado quantos forem os estados**
- **Cada estado é codificado de modo que somente uma das variáveis de estado vale “1” e todas as demais valem “0”**

# Máquinas Seqüenciais Síncronas

## ► Codificação de Estados

**Aplicando a Codificação “One-Hot” ao Exemplo 2:**

**A=001, B=010, C=100**

Estado atual	w	Próximo estado
A	0	A
A	1	B
B	0	A
B	1	C
C	0	A
C	1	C



	Estado atual y2 y1 y0	w	Próximo estado Y2 Y1 Y0	
A	001	0	001	A
A	001	1	010	B
B	010	0	001	A
B	010	1	100	C
C	100	0	001	A
C	100	1	100	C



As combinações de entrada não citadas têm como próximo estado XXX



# Máquinas Seqüenciais Síncronas

## ► Codificação de Estados

Aplicando a Codificação “One-Hot” ao Exemplo 2:

A=001, B=010, C=100

	Estado atual y2 y1 y0	w	Próximo estado Y2 Y1 Y0	
A	001	0	001	A
A	001	1	010	B
B	010	0	001	A
B	010	1	100	C
C	100	0	001	A
C	100	1	100	C

y2	$\bar{y}_2\bar{y}_1$	$\bar{y}_2y_1$	$y_2y_1$	$y_2\bar{y}_1$
$\bar{y}_0\bar{w}$	X	0	X	0
$\bar{y}_0w$	X	1	X	1
$y_0w$	0	X	X	X
$y_0\bar{w}$	0	X	X	X

$\bar{y}_0 \cdot w$

# Máquinas Seqüenciais Síncronas

## ► Codificação de Estados

Aplicando a Codificação “One-Hot” ao Exemplo 2:  
A=001, B=010, C=100

	Estado atual y2 y1 y0	w	Próximo estado Y2 Y1 Y0	
A	001	0	001	A
A	001	1	010	B
B	010	0	001	A
B	010	1	100	C
C	100	0	001	A
C	100	1	100	C

Y1	$\bar{y}_2\bar{y}_1$	$\bar{y}_2y_1$	$y_2y_1$	$y_2\bar{y}_1$
$\bar{y}_0\bar{w}$	X	0	X	0
$\bar{y}_0w$	X	0	X	0
$y_0w$	1	X	X	X
$y_0\bar{w}$	0	X	X	X

*Note: A red box highlights the row for  $y_0w$ , and an arrow points to the cell containing '1' with the label  $y_0 \cdot w$ .*

# Máquinas Seqüenciais Síncronas

## ► Codificação de Estados

Aplicando a Codificação “One-Hot” ao Exemplo 2:  
A=001, B=010, C=100

	Estado atual y2 y1 y0	w	Próximo estado Y2 Y1 Y0	
A	001	0	001	A
A	001	1	010	B
B	010	0	001	A
B	010	1	100	C
C	100	0	001	A
C	100	1	100	C

y0	$\bar{y}_2\bar{y}_1$	$\bar{y}_2y_1$	$y_2y_1$	$y_2\bar{y}_1$
$\bar{y}_0\bar{w}$	X	1	X	1
$\bar{y}_0w$	X	0	X	0
$y_0w$	0	X	X	X
$y_0\bar{w}$	1	X	X	X

# Máquinas Seqüenciais Síncronas

## ► Codificação de Estados

Aplicando a Codificação “One-Hot” ao Exemplo 2:

A=001, B=010, C=100

	Estado $y_2 y_1 y_0$	z
A	001	0
B	010	0
C	100	1



As combinações de entrada  
não citadas têm como saída X

z	$\bar{y}_2\bar{y}_1$	$\bar{y}_2y_1$	$y_2y_1$	$y_2\bar{y}_1$
$\bar{y}_0$	X	0	X	1
$y_0$	0	X	X	X

$y_2$

$$z = y_2$$

# Máquinas Seqüenciais Síncronas

## ► Minimização e Codificação de Estados

Assinalamento A=00, B=01, C=10

$$\begin{aligned}Y1 &= w \cdot (y1 + y0) \\ Y0 &= w \cdot \overline{y1} \cdot \overline{y0} \\ z &= y1\end{aligned}$$

Assinalamento A=00, B=01, C=11

$$\begin{aligned}Y1 &= w \cdot y0 \\ Y0 &= w \\ z &= y1\end{aligned}$$

Codificação “One Hot”

$$\begin{aligned}Y2 &= \overline{y0} \cdot w \\ Y1 &= y0 \cdot w \\ Y0 &= \overline{w} \\ z &= y2\end{aligned}$$

# Máquinas Seqüenciais Síncronas

## ► Síntese de Circuitos Seqüenciais: Modelo de Mealy

### Exemplo 2:

Projete um circuito que satisfaça às seguintes especificações:

1. O circuito possui uma entrada, **w**, e uma saída, **z**.
2. Todas as mudanças de valores no circuito ocorrem na borda de subida do sinal de relógio.
3. Quando o circuito detectar que a entrada **w** vale “0”, a saída **z** deve valer “0” no ciclo de relógio seguinte. Porém, quando o circuito detectar que a entrada **w** vale “1” durante duas bordas de relógio consecutivas, a saída **z** deve passar a valer “1” no ciclo de relógio seguinte à segunda ocorrência do valor “1”. As mudanças de **z** estão sincronizadas com a borda de relógio ativa.

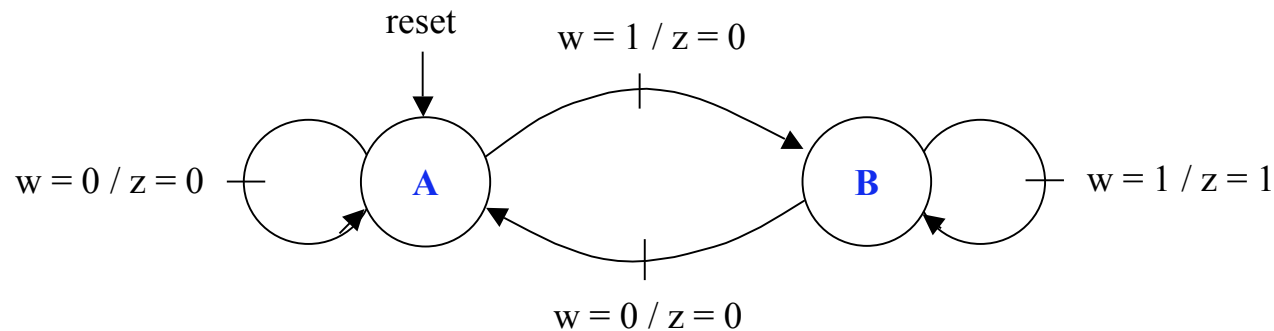
### Considere a seguinte modificação da especificação acima:

- O sinal de saída **z** não precisa esperar que um segundo valor igual a “1” seja amostrado da entrada **w**.
- Porém, se **z** = 1 e **w** muda de “1” para “0”, **z** deve também mudar para “0”, independentemente da borda ativa do relógio

# Máquinas Seqüenciais Síncronas

## ► Síntese de Circuitos Seqüenciais: Modelo de Mealy

### Exemplo 2: Diagrama de Estados



Interpretação do Diagrama de Estados:

- Durante o ciclo de relógio atual, o valor da saída  $z$  corresponde ao rótulo assinalado em alguma das arestas que partem do estado atual.
- No caso do estado B, por exemplo,  $z$  pode valer “0” ou valer “1”, conforme for o valor de  $w$ . Isto implica que  $z$  pode mudar de valor antes que a máquina de estados mude de estado.

# Máquinas Seqüenciais Síncronas

## ► Síntese de Circuitos Seqüenciais: Modelo de Mealy

### Exemplo 2: Tabelas de (Transição de) Estados e de Saída

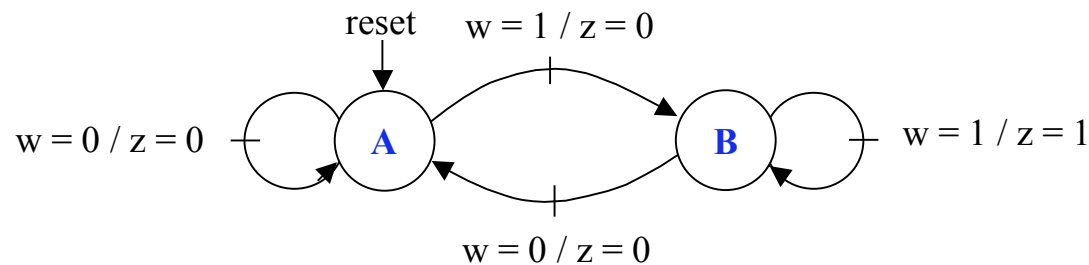


Tabela de (transição de) estados

Estado atual	w	Próximo estado
A	0	A
A	1	B
B	0	A
B	1	B

Tabela de saída

Estado	w	z
A	0	0
A	1	0
B	0	0
B	1	1



# Máquinas Seqüenciais Síncronas

## ► Síntese de Circuitos Seqüenciais: Modelo de Mealy

### Exemplo 2: Projetando a lógica de próximo estado e a lógica de saída

- Como há somente dois estados, iremos utilizar somente uma variável de estado.
- Assumindo o seguinte assinalamento de estados: A=0, B=1

Lógica de Próximo Estado

	y	w	Y	
A	0	0	0	A
A	0	1	1	B
B	1	0	0	A
B	1	1	1	B

$$Y = \overline{y} \cdot w + y \cdot w = w$$

Lógica de Saída

y	w	z
0	0	0
0	1	0
1	0	0
1	1	1

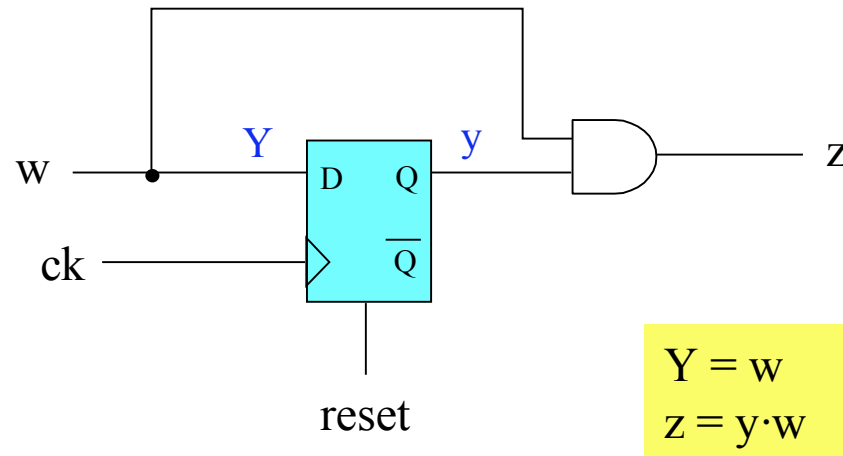


$$z = y \cdot w$$

# Máquinas Seqüenciais Síncronas

## ► Síntese de Circuitos Seqüenciais: Modelo de Mealy

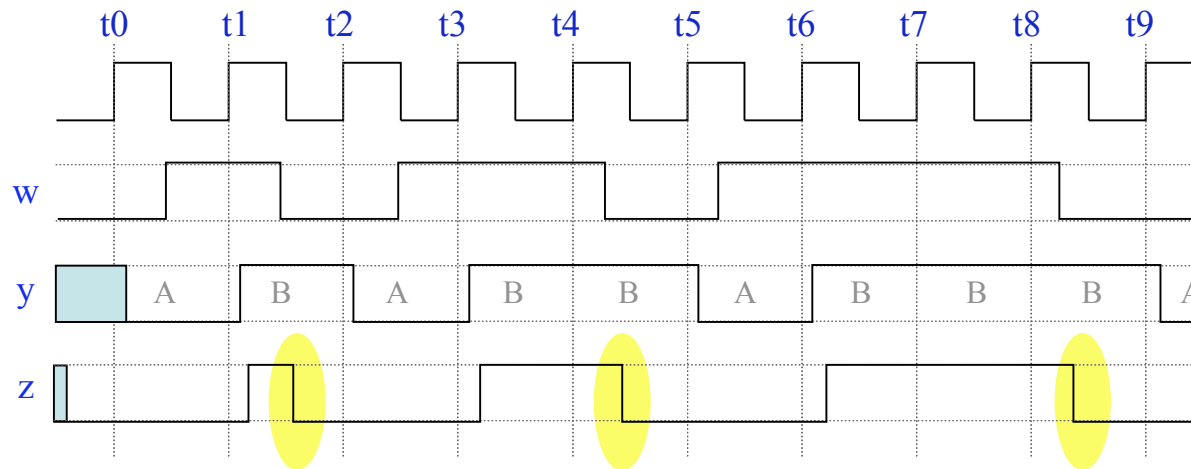
### Exemplo 2: circuito final



# Máquinas Seqüenciais Síncronas

## ► Síntese de Circuitos Seqüenciais: Modelo de Mealy

**Exemplo 2: exemplo de funcionamento dinâmico**

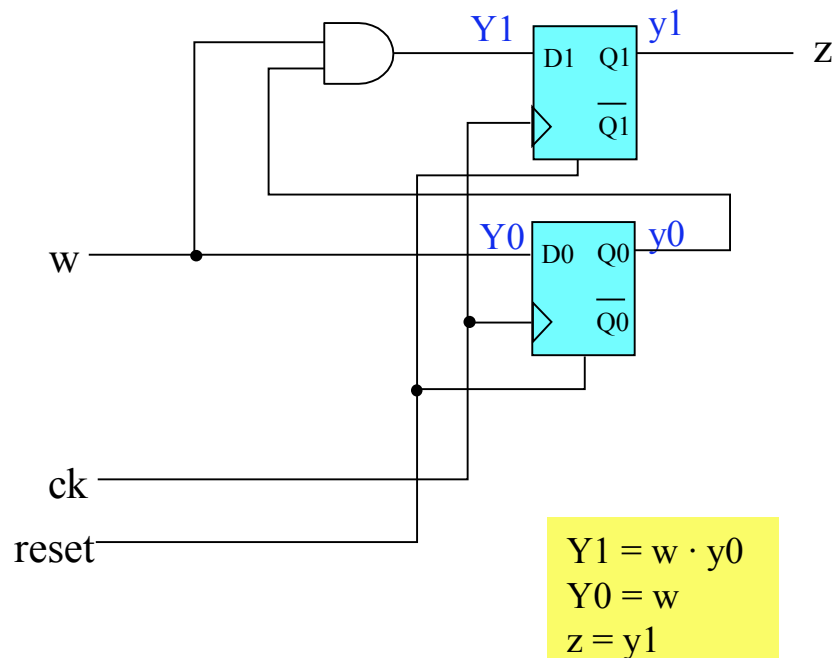


# Máquinas Seqüenciais Síncronas

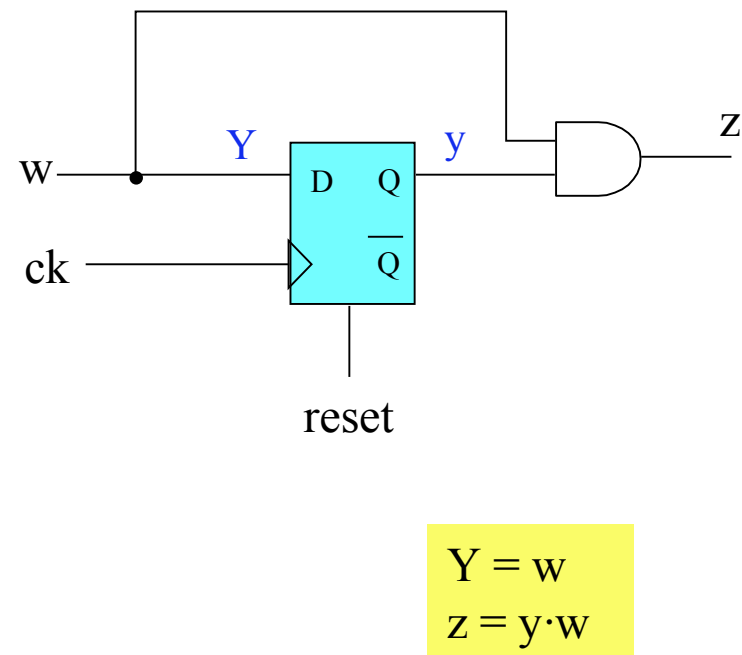
## ► Síntese de Circuitos Seqüenciais

### Exemplo 2: Comparando Moore e Mealy

#### Moore



#### Mealy

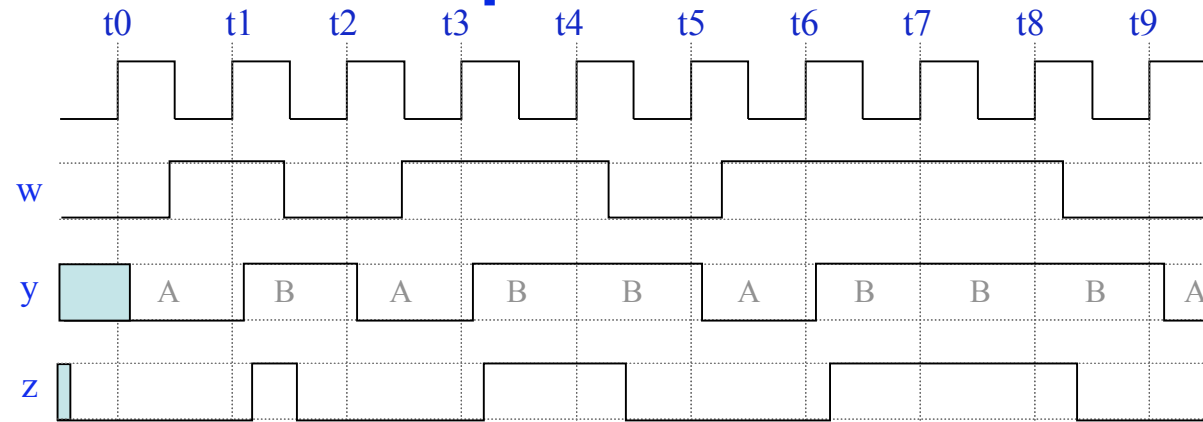


# Máquinas Seqüenciais Síncronas

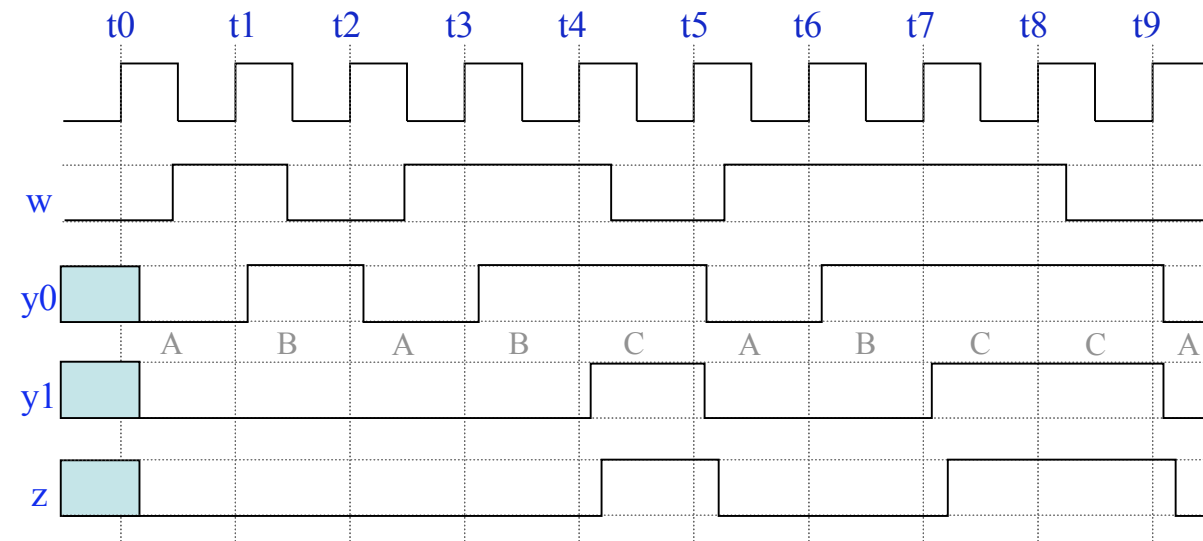
## ► Síntese de Circuitos Seqüenciais

Exemplo 2:

Mealy



Moore

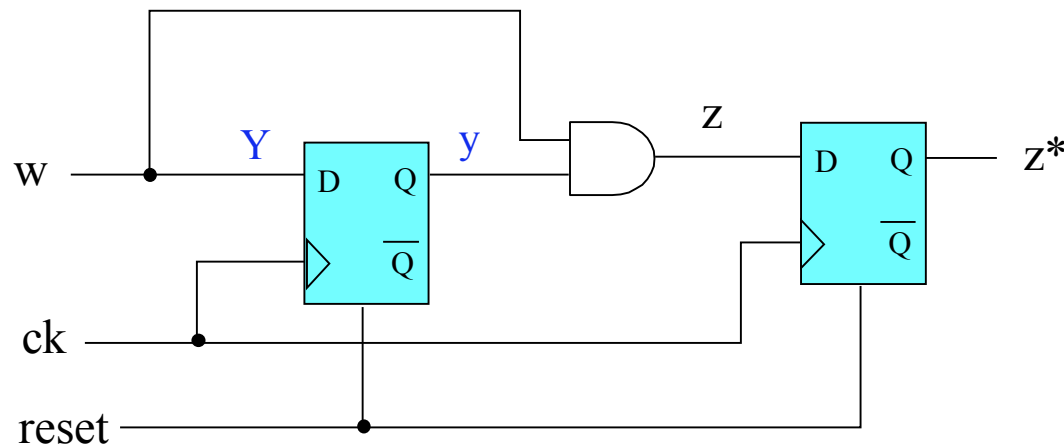


# Máquinas Seqüenciais Síncronas

## ► Síntese de Circuitos Seqüenciais

### Exemplo 2:

Porém, se passarmos a saída  $z$  por um segundo flip-flop, filtraremos o comportamento assíncrono. De fato, estaremos transformando o circuito para o Modelo de Moore...

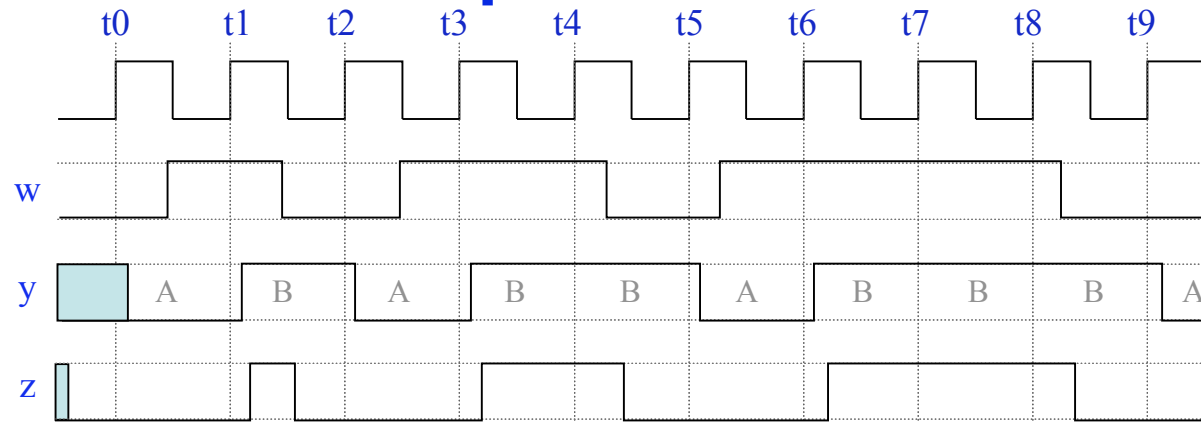


# Máquinas Seqüenciais Síncronas

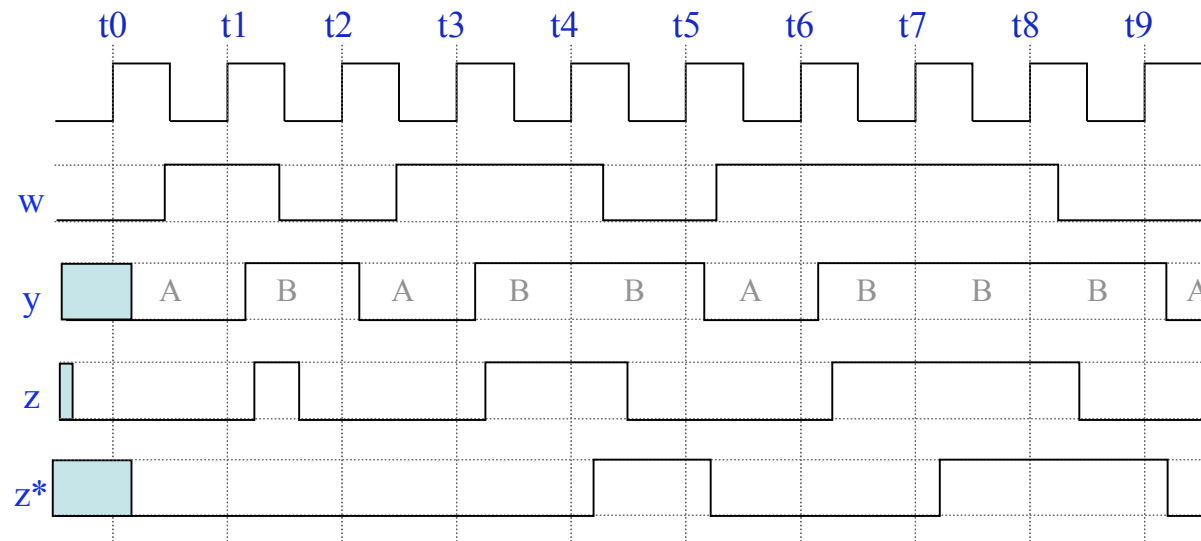
## ► Síntese de Circuitos Seqüenciais

Exemplo 2:

Mealy

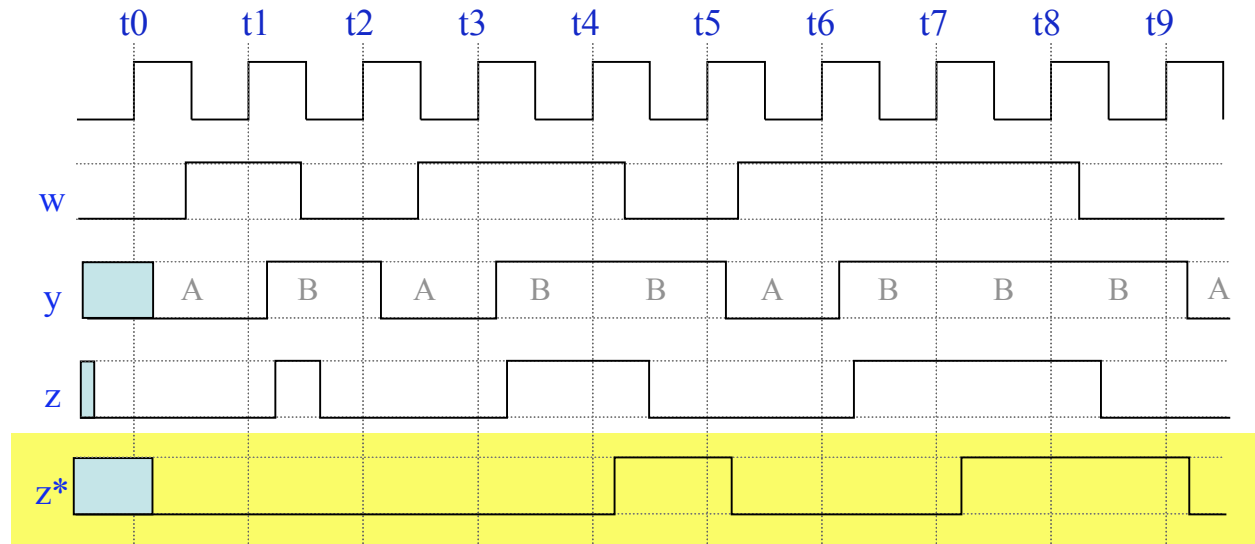


Mealy  
transform.  
em Moore

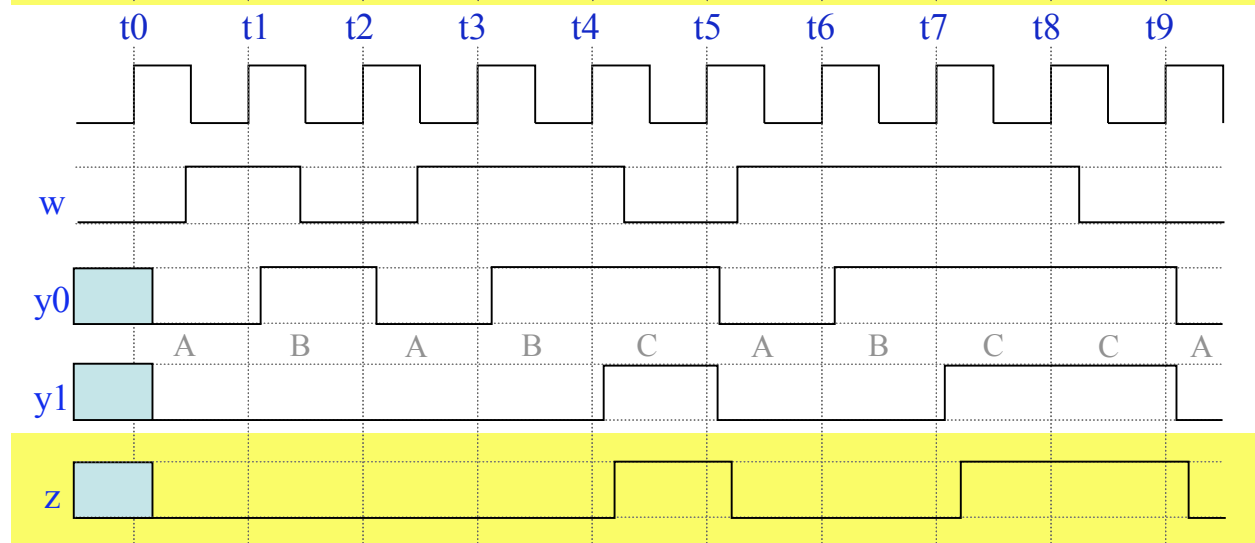


# Máquinas Seqüenciais Síncronas

**Mealy  
transform.  
em Moore**



**Moore**





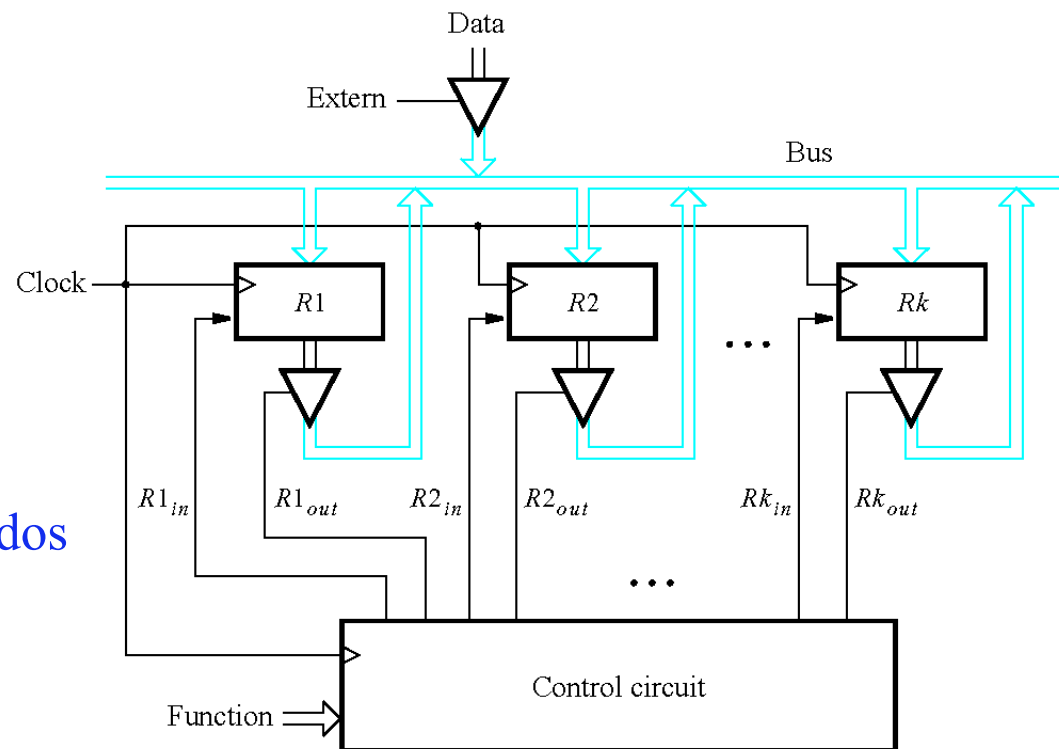
# Máquinas Seqüenciais Síncronas

## ► Síntese de Circuitos Seqüenciais: Modelo de Mealy

### Exemplo 3:

Passos:	Sinais de controle
$R3 \leftarrow R2;$	$R2_{out} = 1; R3_{in} = 1;$
$R2 \leftarrow R1;$	$R1_{out} = 1; R2_{in} = 1;$
$R1 \leftarrow R3;$	$R3_{out} = 1; R1_{in} = 1;$ $Done = 1;$

Obs: os sinais de saída não citados na tabela devem valer “0”

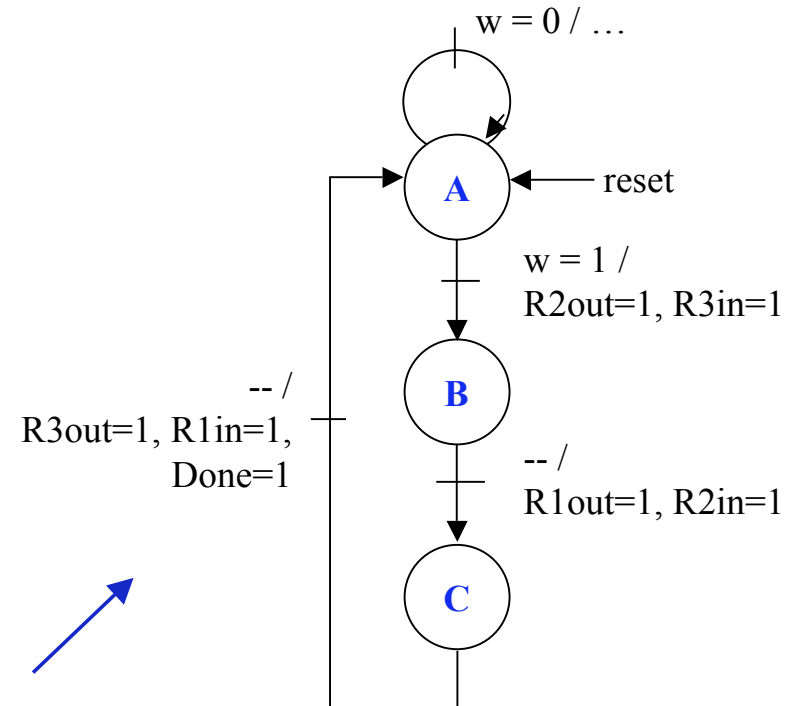


# Máquinas Seqüenciais Síncronas

## ► Síntese de Circuitos Seqüenciais: Modelo de Mealy

### Exemplo 3:

Passos:	Sinais de controle
$R3 \leftarrow R2;$	$R2out = 1; R3in = 1;$
$R2 \leftarrow R1;$	$R1out = 1; R2in = 1;$
$R1 \leftarrow R3;$	$R3out = 1; R1in = 1;$ $Done = 1;$



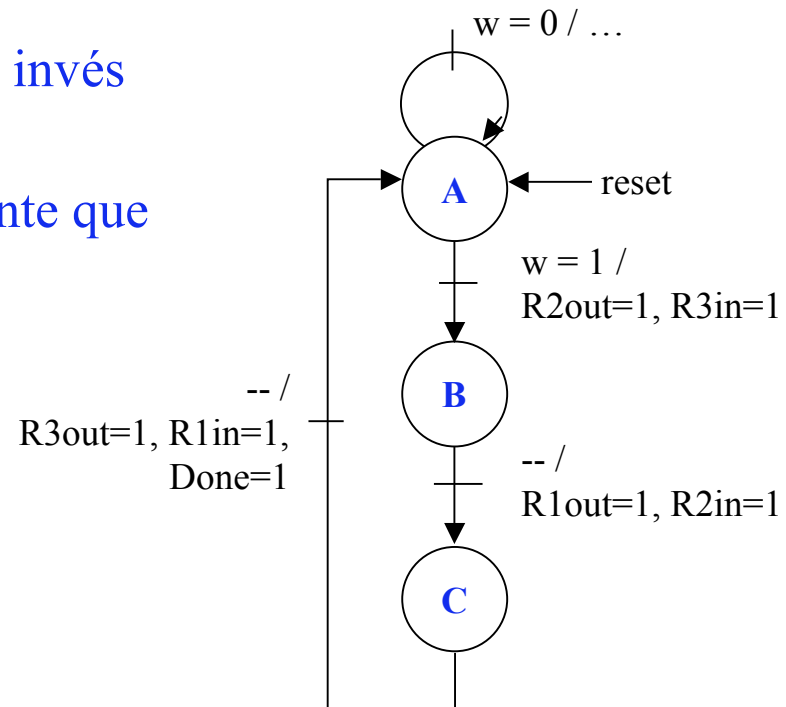
Obs: os sinais de saída não citados na tabela devem valer “0”

# Máquinas Seqüenciais Síncronas

## ► Síntese de Circuitos Seqüenciais: Modelo de Mealy

### Exemplo 3:

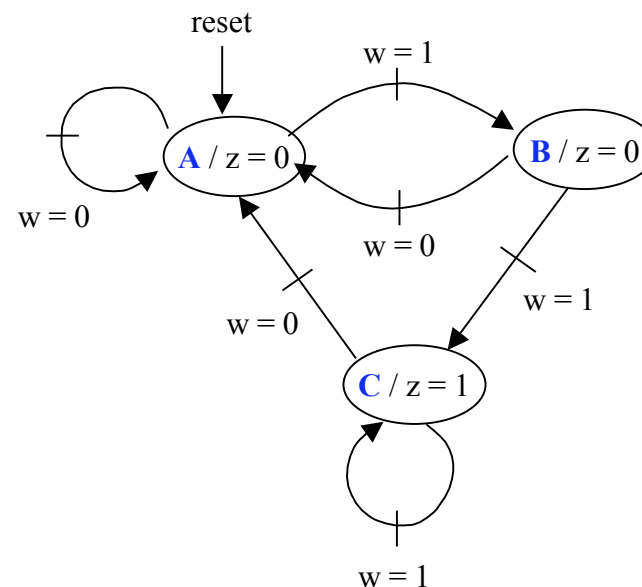
- A versão Mealy requer três estados (ao invés de quatro da versão Moore)
- Porém, isto não significa necessariamente que o circuito será menor, pois ainda são necessários dois flip-flops...
- A versão Mealy para o exemplo 3 gera os sinais de controle um ciclo de relógio mais cedo que a versão Moore
- Logo, para realizar o swap entre dois registradores a versão Mealy necessita de 3 ciclos de relógio, enquanto a versão Morre necessita de 4 ciclos .



# Máquinas Seqüenciais Síncronas

## ► Máquinas de Estados

- VHDL não define padrão para a descrição de máquinas de estados finitos
- Existe mais de uma maneira de se descrever uma dada FSM
- Considere a FSM com o seguinte diagrama de estados
- Esta FSM segue o modelo de Moore



# Máquinas Seqüenciais Síncronas

## ▶ Máquinas de Estados

```
1  LIBRARY ieee ;
2  USE ieee.std_logic_1164.all ;

3  ENTITY simple IS
4      PORT ( Clock, Resetn, w      : IN  STD_LOGIC ;
           z      : OUT STD_LOGIC ) ;
5  END simple ;

7  ARCHITECTURE Behavior OF simple IS
8      TYPE State_type IS (A, B, C) ;
9      SIGNAL y : State_type ;
10 BEGIN
11     PROCESS ( Resetn, Clock )
12     BEGIN
13         IF Resetn = '0' THEN
14             y <= A ;
15         ELSIF (Clock'EVENT AND Clock = '1') THEN
16             CASE y IS
17                 WHEN A =>
18                     IF w = '0' THEN
19                         y <= A ;
20                     ELSE
21                         y <= B ;
22                     END IF ;
```

**FSM descrita segundo o Modelo de Moore, Versão 1 (somente 1 processo)**

“TYPE” permite criar um tipo de sinal definido pelo usuário.  
Neste caso, se está definindo um dado chamado State\_type que pode assumir um entre 3 valores: A, B, C

# Máquinas Seqüenciais Síncronas

## ▶ Máquinas de Estados

```
15      ELSIF (Clock'EVENT AND Clock = '1') THEN
16          CASE y IS
17              WHEN A =>
18                  IF w = '0' THEN
19                      y <= A ;
20                  ELSE
21                      y <= B ;
22                  END IF ;
23              WHEN B =>
24                  IF w = '0' THEN
25                      y <= A ;
26                  ELSE
27                      y <= C ;
28                  END IF ;
29              WHEN C =>
30                  IF w = '0' THEN
31                      y <= A ;
32                  ELSE
33                      y <= C ;
34                  END IF ;
35          END CASE ;
36      END IF ;
37  END PROCESS ;
38  z <= '1' WHEN y = C ELSE '0' ;
39 END Behavior ;
```

**FSM descrita segundo o Modelo  
de Moore, Versão 1  
(continuação)**

# Máquinas Seqüenciais Síncronas

## ▶ Máquinas de Estados

```
ARCHITECTURE Behavior OF simple IS
  TYPE State_type IS (A, B, C) ;
  SIGNAL y_present, y_next : State_type ;
BEGIN
  PROCESS ( w, y_present )
  BEGIN
    CASE y_present IS
      WHEN A =>
        IF w = '0' THEN
          y_next <= A ;
        ELSE
          y_next <= B ;
        END IF ;
      WHEN B =>
        IF w = '0' THEN
          y_next <= A ;
        ELSE
          y_next <= C ;
        END IF ;
      WHEN C =>
        IF w = '0' THEN
          y_next <= A ;
        ELSE
          y_next <= C ;
        END IF ;
    END CASE ;
  END PROCESS ;
```

**FSM descrita segundo o Modelo de Moore, Versão 2 (2 processos)**

Um processo para o bloco de próxima estado ...

# Máquinas Seqüenciais Síncronas

## ► Máquinas de Estados

FSM descrita segundo o Modelo de Moore, Versão 2 (2 processos)

```
PROCESS (Clock, Resetn)
BEGIN
  IF Resetn = '0' THEN
    y_present <= A ;
  ELSIF (Clock'EVENT AND Clock = '1') THEN
    y_present <= y_next ;
  END IF ;
END PROCESS ;

z <= '1' WHEN y_present = C ELSE '0' ;
END Behavior ;
```

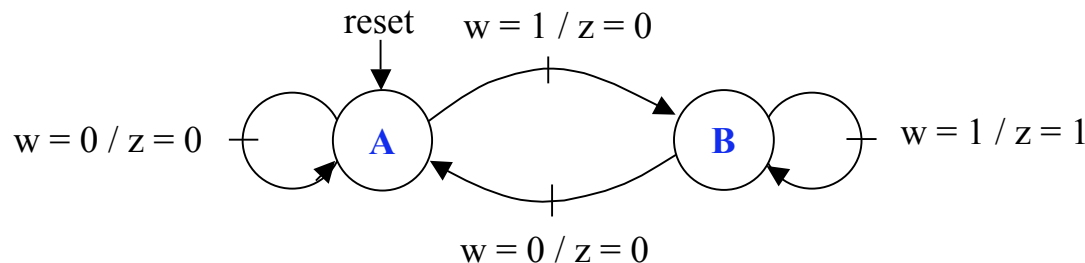
... e outro processo para o registrador de estado



# Máquinas Seqüenciais Síncronas

## ► Máquinas de Estados

- Considere a FSM descrita pelo diagrama de estados que segue, a qual segue o **modelo de Mealy**



# Máquinas Seqüenciais Síncronas

## ▶ Máquinas de Estados

```
LIBRARY ieee ;
USE ieee.std_logic_1164.all ;
ENTITY mealy IS
    PORT ( Clock, Resetn, w      : IN    STD_LOGIC ;
          z                      : OUT   STD_LOGIC ) ;
END mealy ;
```

```
ARCHITECTURE Behavior OF mealy IS
    TYPE State_type IS (A, B) ;
    SIGNAL y : State_type ;
BEGIN
    PROCESS ( Resetn, Clock )
    BEGIN
        IF Resetn = '0' THEN
            y <= A ;
        ELSIF (Clock'EVENT AND Clock = '1') THEN
            CASE y IS
                WHEN A =>
                    IF w = '0' THEN y <= A ;
                    ELSE y <= B ;
                    END IF ;
                WHEN B =>
                    IF w = '0' THEN y <= A ;
                    ELSE y <= B ;
                    END IF ;
            END CASE ;
        END IF ;
    END PROCESS ;
```

**FSM descrita segundo o  
Modelo de Mealy  
(2 processos)**

# Máquinas Seqüenciais Síncronas

## ► Máquinas de Estados

```
PROCESS ( y, w )
BEGIN
  CASE y IS
    WHEN A =>
      z <= '0' ;
    WHEN B =>
      z <= w ;
  END CASE ;
END PROCESS ;
END Behavior ;
```

**FSM descrita segundo o  
Modelo de Mealy  
(2 processos)  
Continuação**

# Máquinas Seqüenciais Síncronas

## ▶ Máquinas de Estados

### Outra FSM

```
Use ieee.std_logic_1164.all;
```

```
ENTITY Cont IS
```

```
  PORT(clk, reset, flag1 : IN STD_LOGIC;  
        S1, S2, S3 : OUT STD_LOGIC);
```

```
END Cont;
```

```
ARCHITECTURE comportamento OF Cont IS
```

```
SIGNAL state: STD_LOGIC_VECTOR(1 DOWNT0 0);
```

```
BEGIN
```

```
PROCESS (clk, reset)
```

```
BEGIN
```

```
  IF (reset='0') THEN  
    state <= "00";
```

← Definição das  
variáveis de  
estado

# Máquinas Seqüenciais Síncronas

## ► Máquinas de Estados (continuação)

### Outra FSM

```
ELSIF (clk'EVENT AND clk = '1') THEN
```

```
  CASE state IS
```

```
    WHEN "00" => state <= "01";  S1 <= '0';  S2 <= '1';  S3 <= '1';
```

```
    WHEN "01" => IF flag1 = '1' THEN state <= "10";
```

```
                  ELSE state <= "11"; END IF;
```

```
                  S1 <= '1';  S2 <= '1';  S3 <= '0';
```

```
    WHEN "10" => state <= "11"; S1 <= '1';  S2 <= '1';  S3 <= '1';
```

```
    WHEN "11" => state <= "00"; S1 <= '1';  S2 <= '0';  S3 <= '1';
```

```
    WHEN OTHERS => state <= "00";
```

```
  END CASE;
```

```
END IF;
```

```
END PROCESS;
```

```
END comportamento;
```