

# MC542

## Organização de Computadores Teoria e Prática

2006

Prof. Paulo Cesar Centoducatte

[ducatte@ic.unicamp.br](mailto:ducatte@ic.unicamp.br)

[www.ic.unicamp.br/~ducatte](http://www.ic.unicamp.br/~ducatte)

**MC542**

## **Circuitos Lógicos**

### **Circuitos Seqüenciais Síncronos**

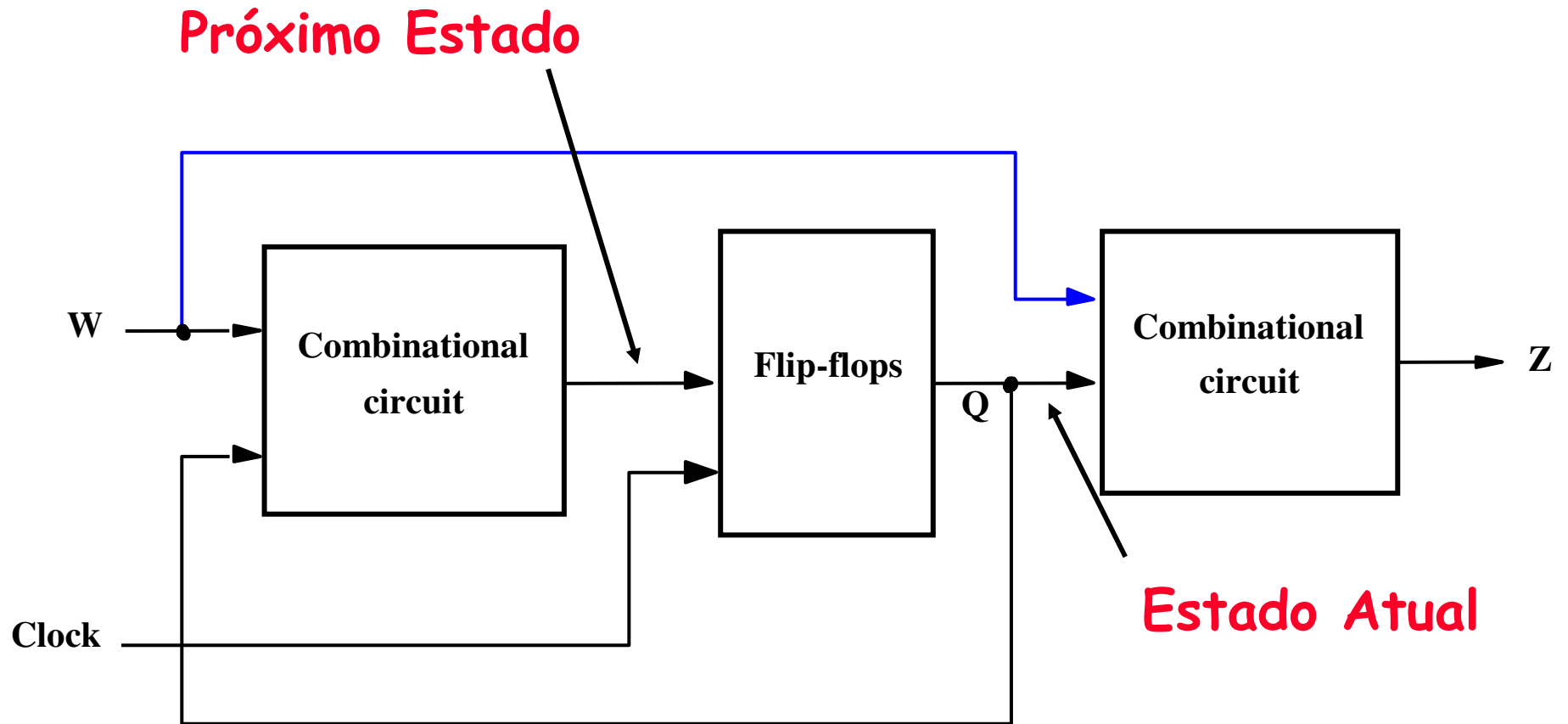
**“Fundamentals of Digital Logic with VHDL  
Design” - (Capítulo 8)**

## Título do Capítulo Abordado

### Sumário

- Diagramas de Estados
- Tabela de Estados
- Atribuição de Estados
- Escolha dos Flip-Flops
- Derivação do Próximo Estado e Saída
- Exemplos

# Circuito Seqüencial



Máquina de Moore

Máquina de Mealy

# Circuitos Seqüenciais Síncronos

- O circuito possui uma entrada **w** e uma saída **s**
- Toda mudança no circuito ocorre na borda do **clock**

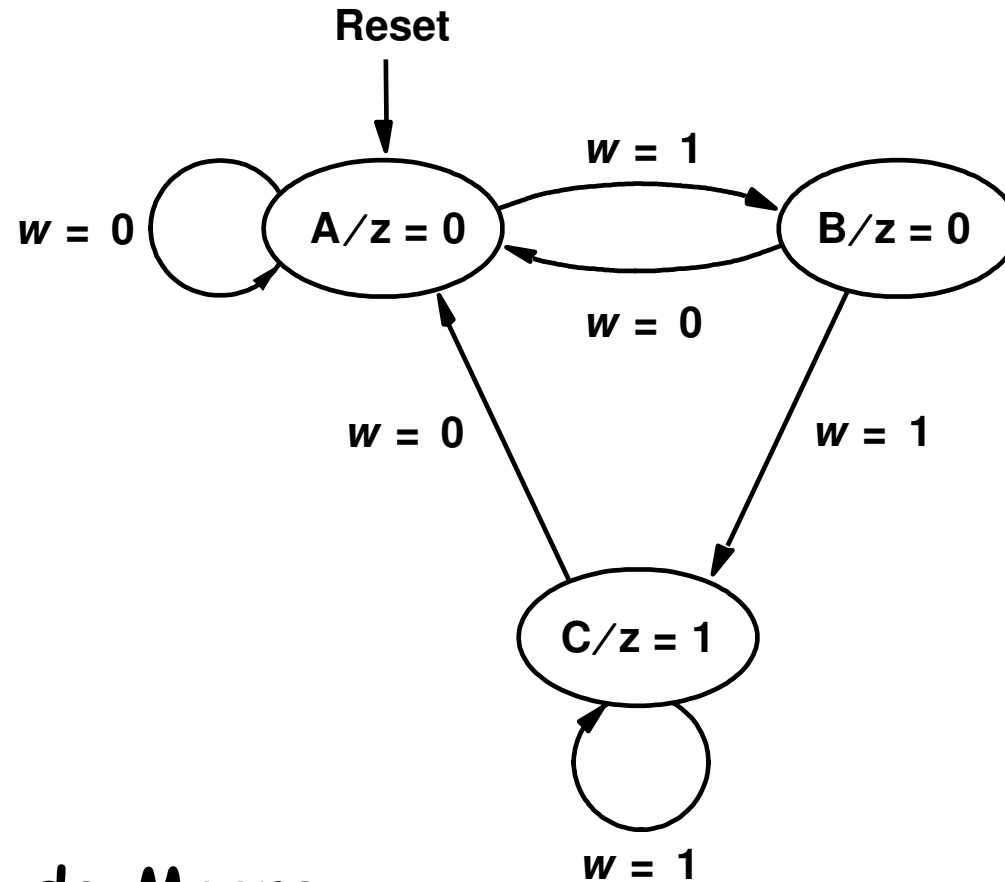
**Exemplo:** Projetar um circuito que possui uma entrada **w** de 1 bit e uma saída **z** também de 1 bit e  $z = 1$  se  $w = 1$  durante os dois ciclos de clock precedentes e  $z = 0$  caso contrário. O circuito deve responder à borda de subida do clock.

# Circuitos Seqüenciais Síncronos

- Exemplo do comportamento do circuito a ser projetado

<b>Clock cycle:</b>	$t_0$	$t_1$	$t_2$	$t_3$	$t_4$	$t_5$	$t_6$	$t_7$	$t_8$	$t_9$	$t_{10}$
$w:$	0	1	0	1	1	0	1	1	1	0	1
$z:$	0	0	0	0	0	1	0	0	1	1	0

# Diagrama de Estados



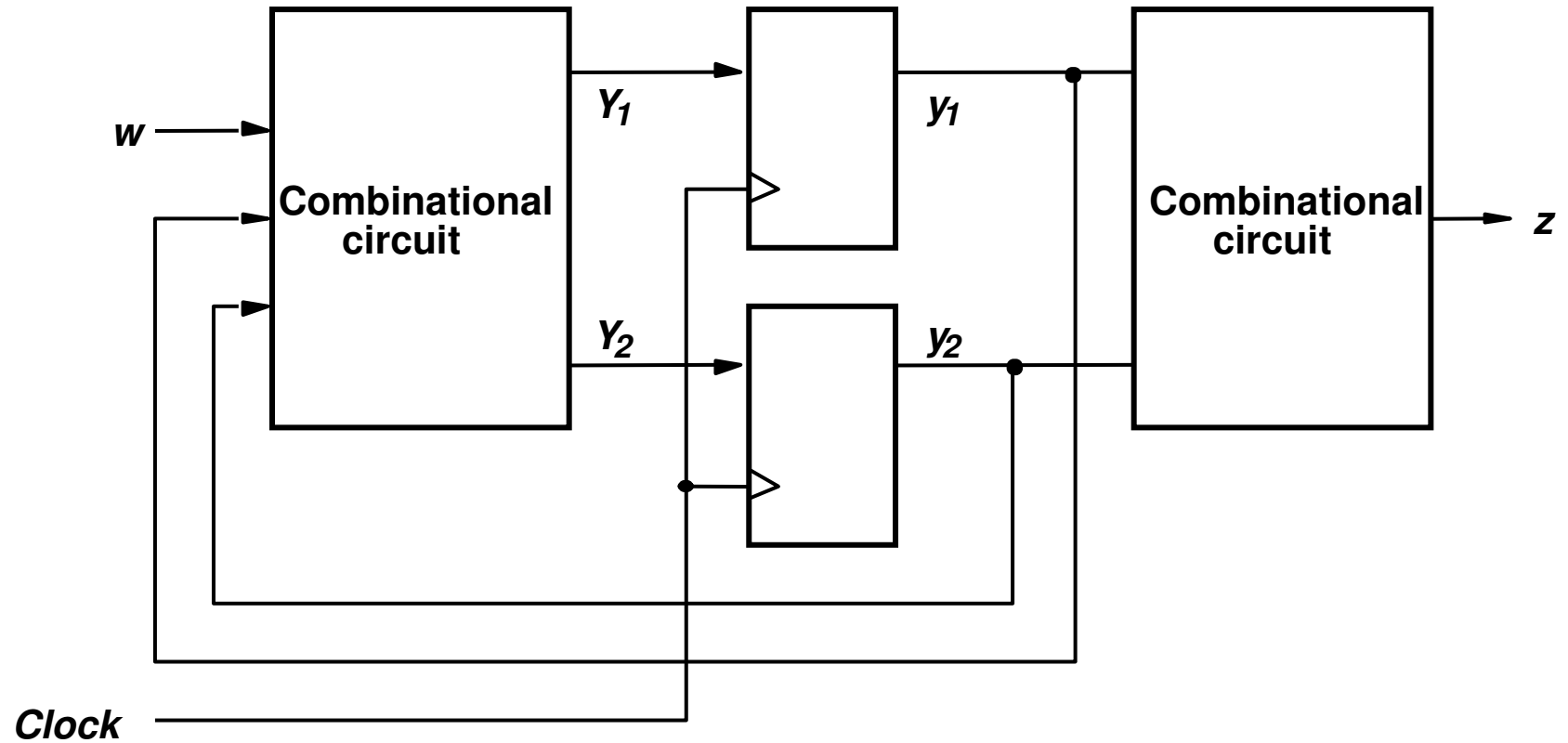
Máquina de Moore

# Tabela de Estados

Present state	Next state		Output $z$
	$w = 0$	$w = 1$	
A	A	B	0
B	A	C	0
C	A	C	1



# Atribuição de Estados



# Atribuição de Estados

	Present state $y_2 y_1$	Next state		Output $z$
		$w = 0$	$w = 1$	
		$Y_2 Y_1$	$Y_2 Y_1$	
A	00	00	01	0
B	01	00	10	0
C	10	00	10	1
	11	<i>dd</i>	<i>dd</i>	<i>d</i>

# Escolha dos Flip-Flops e Derivação das Equações de Excitação e de Saída

## FF tipo D

		$y_2 y_1$			
$w$		00	01	11	10
0		0	0	d	0
1		1	0	d	0

Ignoring don't cares

$$Y_1 = w\bar{y}_1\bar{y}_2$$

Using don't cares

$$Y_1 = w\bar{y}_1\bar{y}_2$$

		$y_2 y_1$			
$w$		00	01	11	10
0		0	0	d	0
1		0	1	d	1

$$Y_2 = wy_1\bar{y}_2 + w\bar{y}_1y_2$$

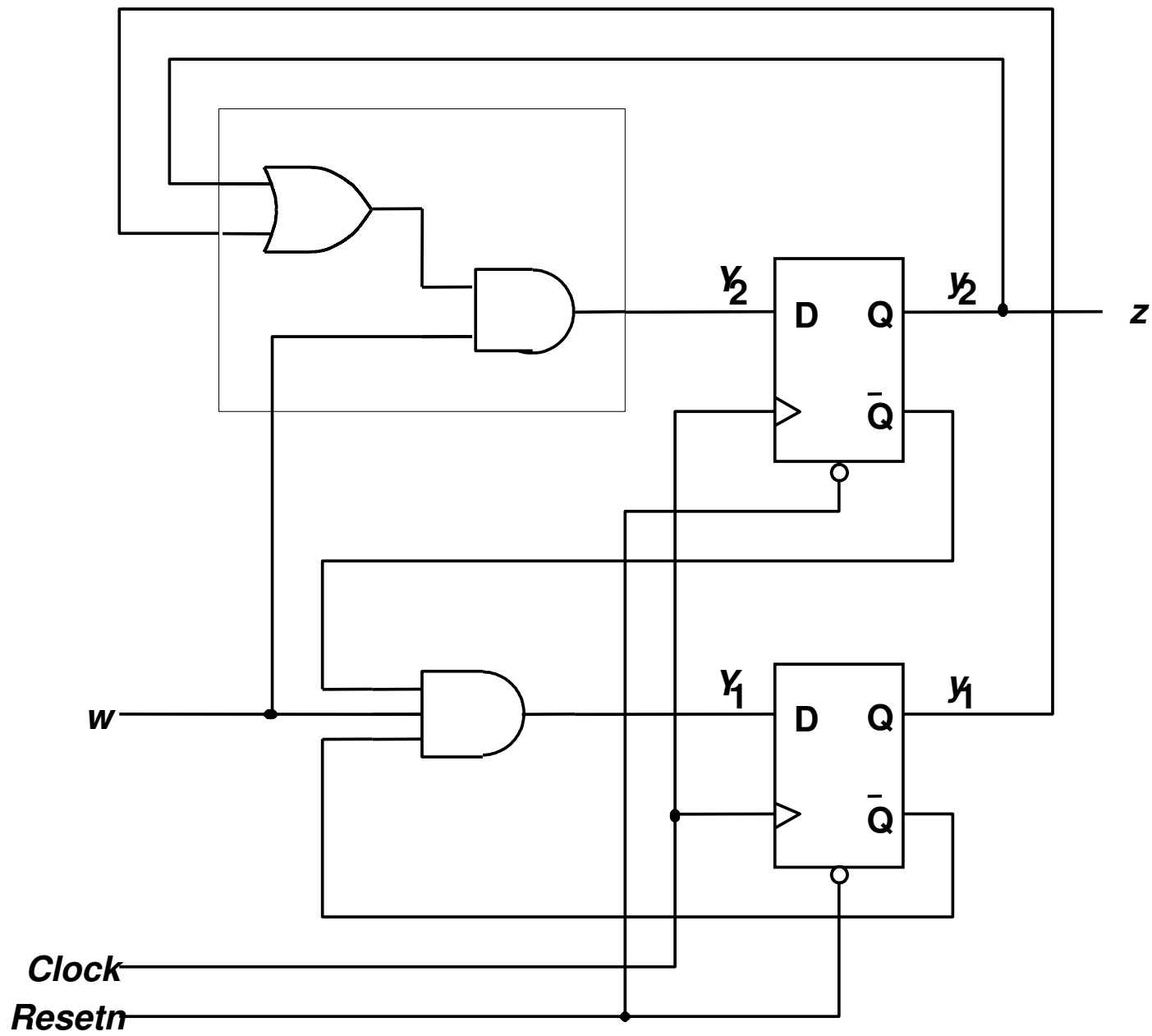
$$Y_2 = wy_1 + wy_2$$

$$= w(y_1 + y_2)$$

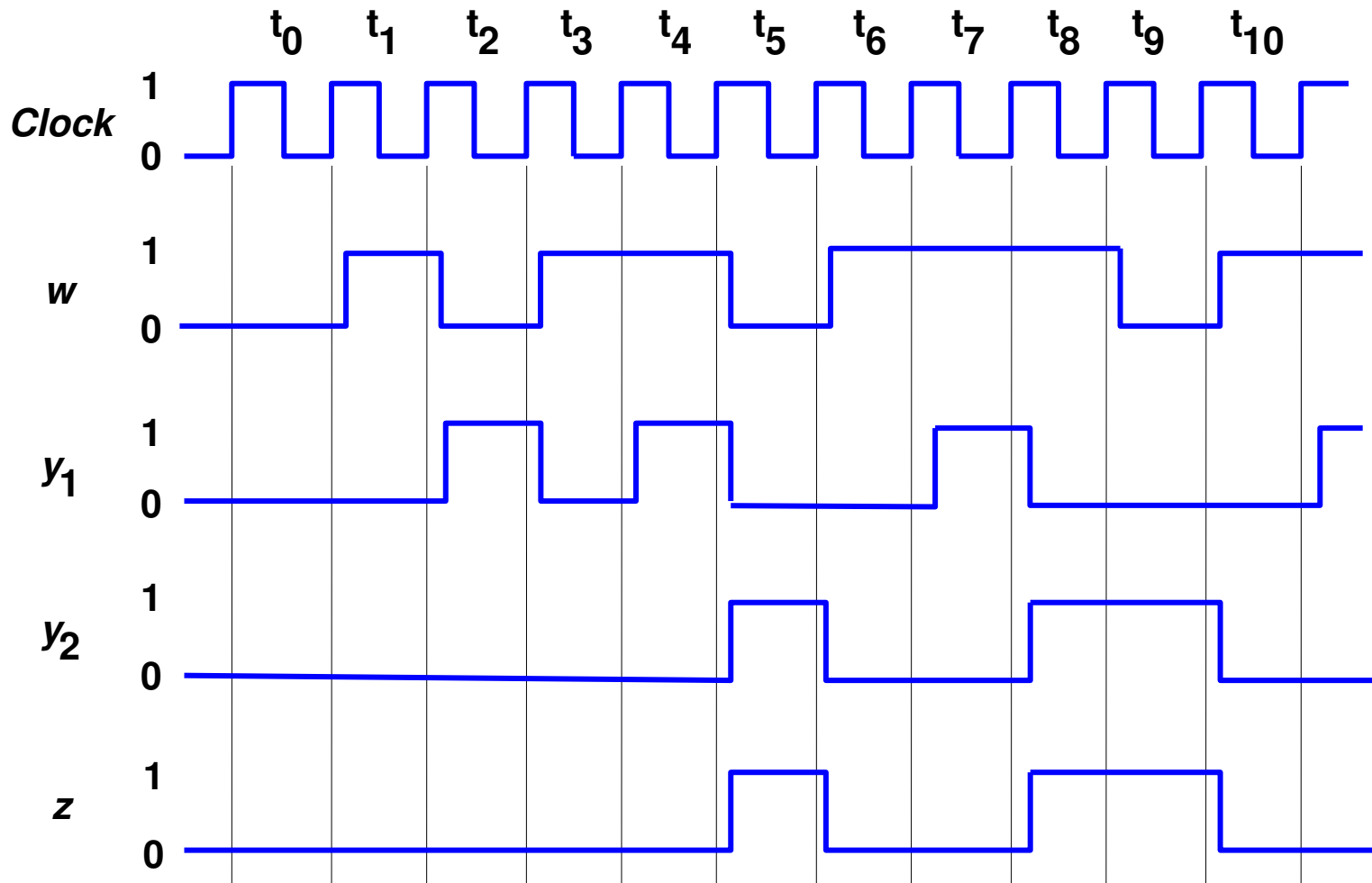
		$y_1$	
	$y_2$	0	1
0		0	0
1		1	d

$$z = \bar{y}_1y_2$$

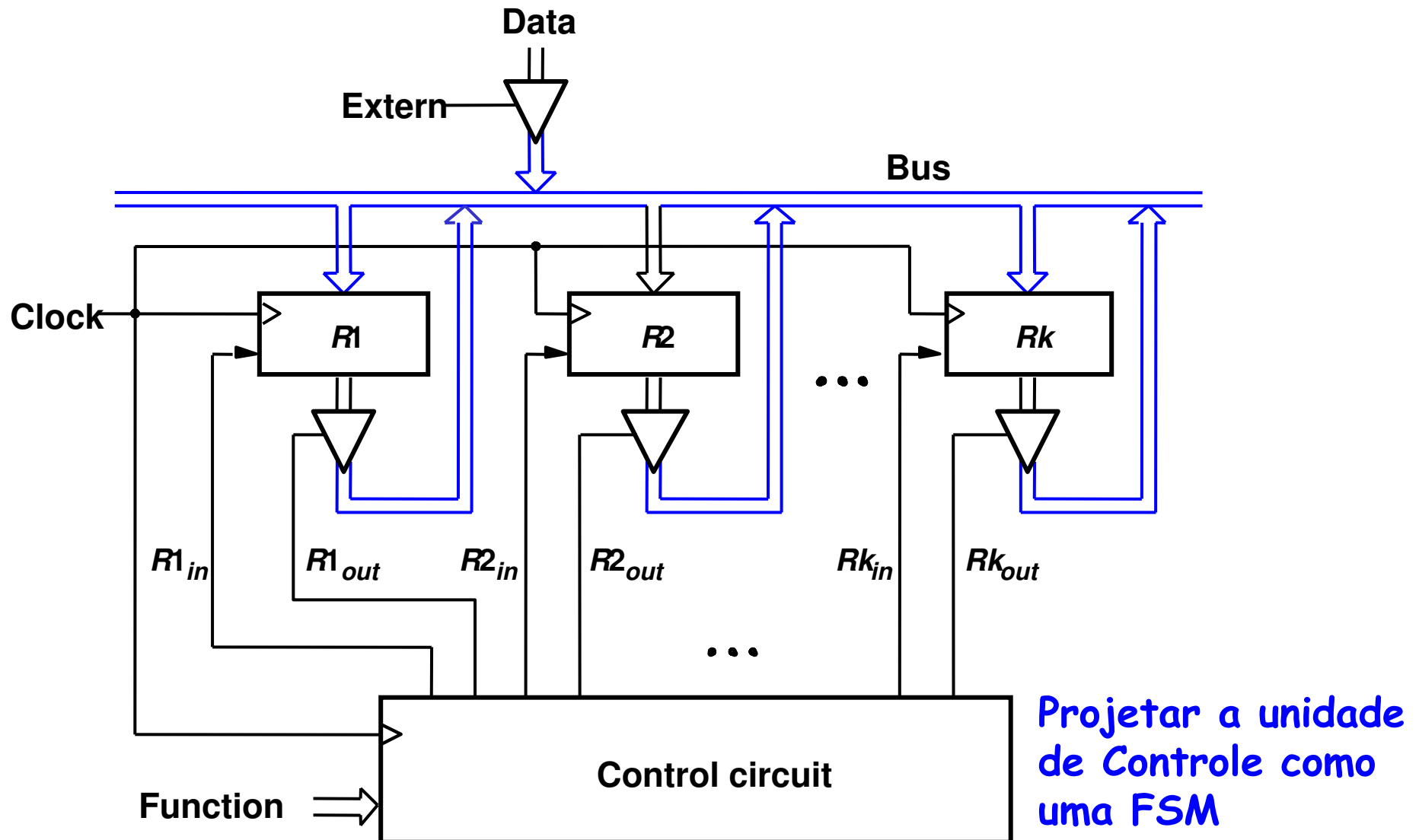
$$z = y_2$$



# Diagrama de Tempo

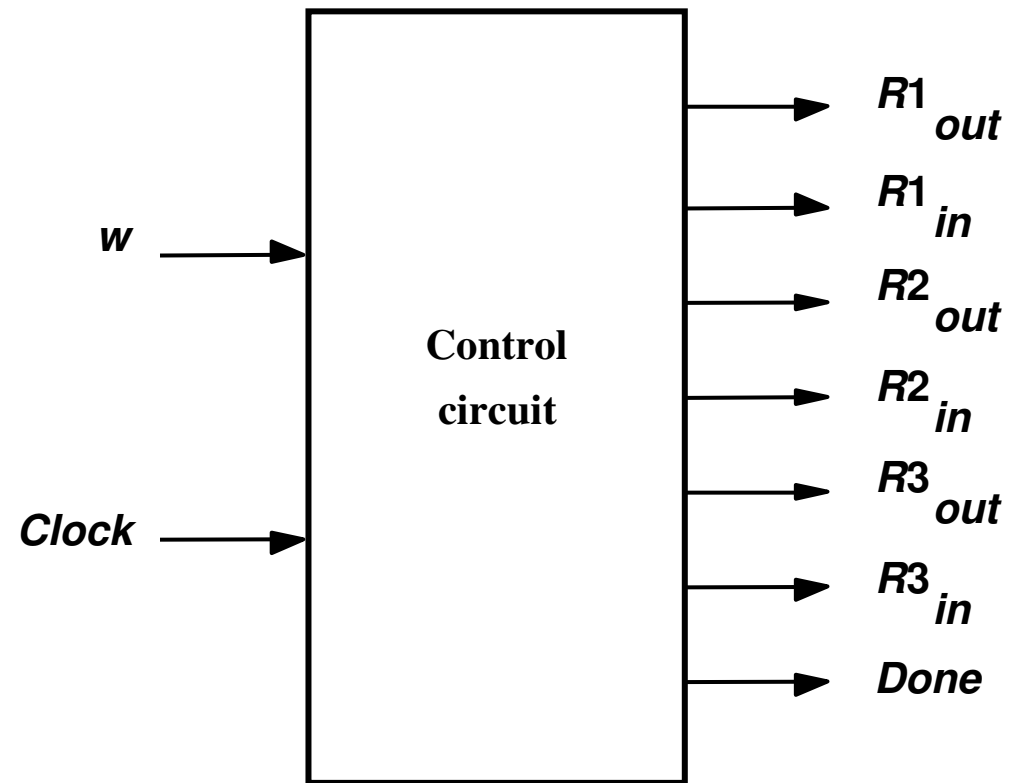


# Exemplo: FSM para Controle de Transferência de Dados Entre Registradores em um Barramento

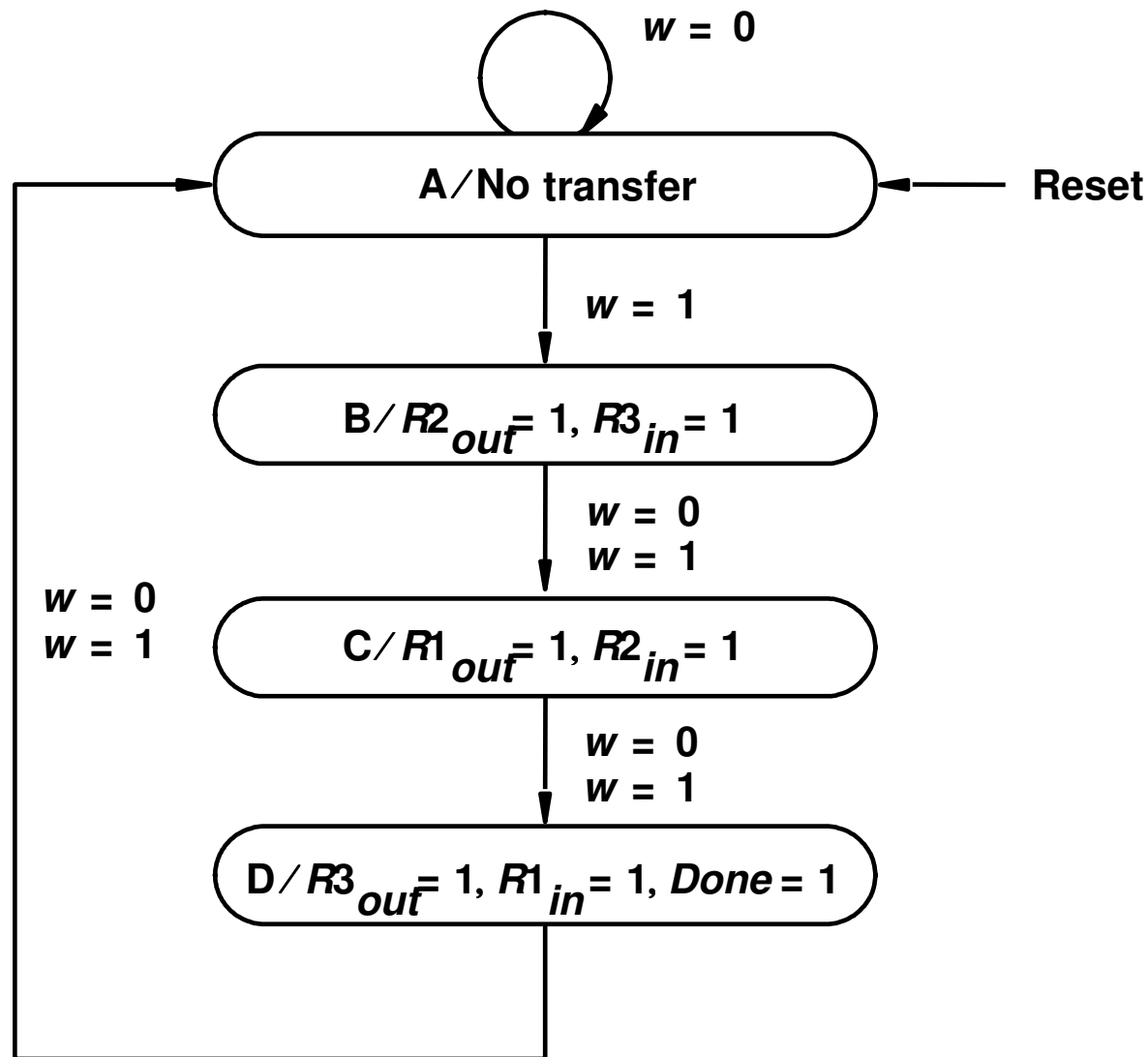


# Exemplo (Cont.)

Controle para realizar swap entre R1 e R2, usando R3 como auxiliar



# Exemplo (Cont.) Diagrama de Estados





## Exemplo (Cont.) Tabela de Estados

Present state	Next state		Outputs						
	$w = 0$	$w = 1$	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	$Done$
A	A	B	0	0	0	0	0	0	0
B	C	C	0	0	1	0	0	1	0
C	D	D	1	0	0	1	0	0	0
D	A	A	0	1	0	0	1	0	1

## Exemplo (Cont.)

### Tabela de Atribuição de Estados

	Present state	Nextstate		Outputs						
		$w = 0$	$w = 1$							
	$y_2y_1$	$Y_2Y_1$	$Y_2Y_1$	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	$Done$
A	00	00	0 1	0	0	0	0	0	0	0
B	01	10	1 0	0	0	1	0	0	1	0
C	10	11	1 1	1	0	0	1	0	0	0
D	11	00	0 0	0	1	0	0	1	0	1

# Derivação das Equações de Excitação e de Saída

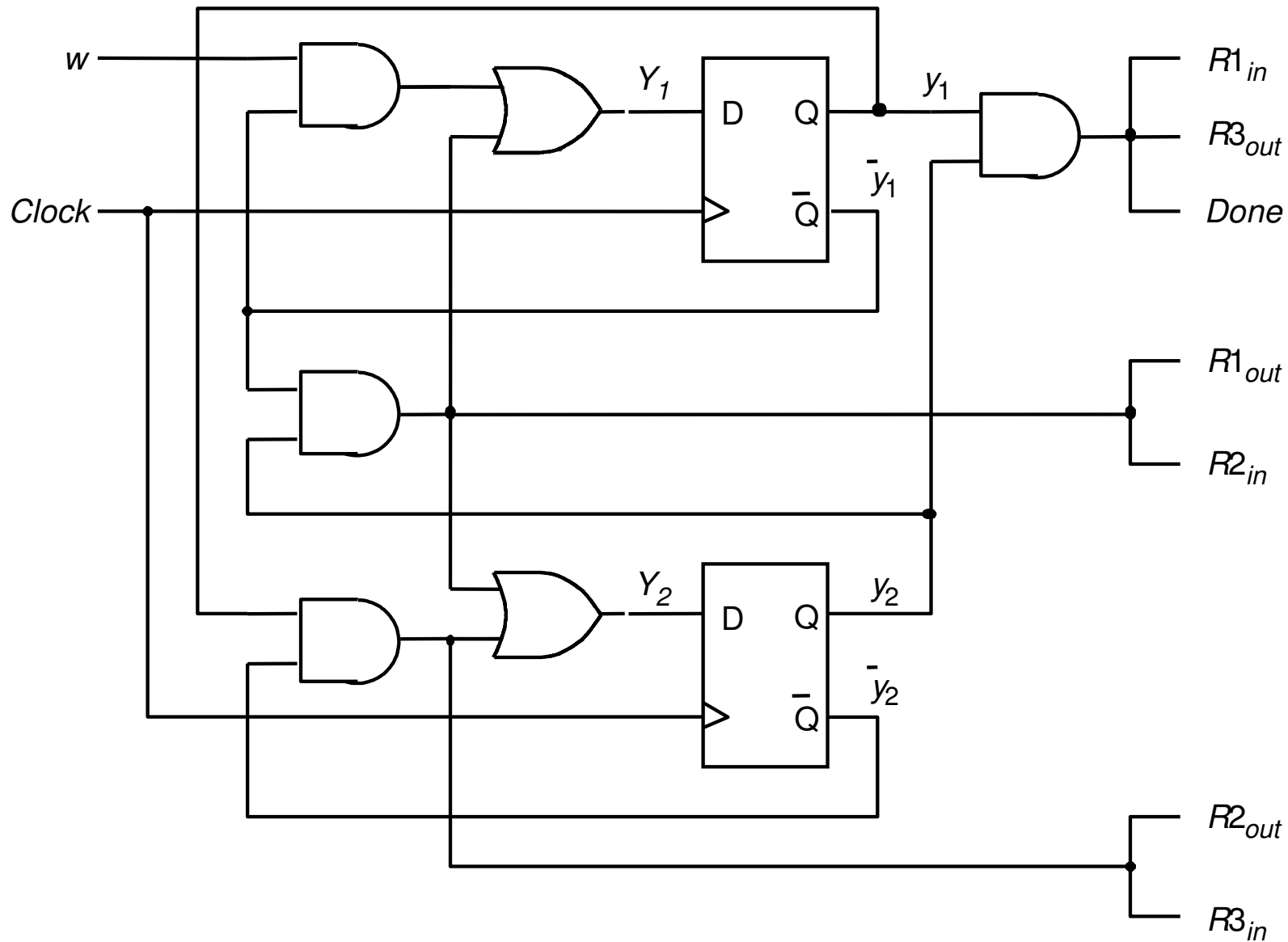
$w$	$y_2y_1$	00	01	11	10
0					1
1		1			1

$$Y_1 = wy\bar{y}_1 + \bar{y}_1y_2$$

$w$	$y_2y_1$	00	01	11	10
0			1		1
1			1		1

$$Y_2 = y_1\bar{y}_2 + \bar{y}_1y_2$$

# Derivação das Equações de Excitação e de Saída



# Atribuição de Estados

	Present state $y_2 y_1$	Next state		Output $z$
		$w = 0$	$w = 1$	
		$Y_2 Y_1$	$Y_2 Y_1$	
<b>A</b>	<b>00</b>	<b>00</b>	<b>01</b>	<b>0</b>
<b>B</b>	<b>01</b>	<b>00</b>	<b>10</b>	<b>0</b>
<b>C</b>	<b>10</b>	<b>00</b>	<b>10</b>	<b>1</b>
	<b>11</b>	<i>dd</i>	<i>dd</i>	<i>d</i>

# Escolha dos Flip-Flops e Derivação das Equações de Excitação e de Saída

## FF tipo D

		$y_2 y_1$			
$w$		00	01	11	10
0		0	0	d	0
1		1	0	d	0

Ignoring don't cares

$$Y_1 = w\bar{y}_1\bar{y}_2$$

Using don't cares

$$Y_1 = w\bar{y}_1\bar{y}_2$$

		$y_2 y_1$			
$w$		00	01	11	10
0		0	0	d	0
1		0	1	d	1

$$Y_2 = wy_1\bar{y}_2 + w\bar{y}_1y_2$$

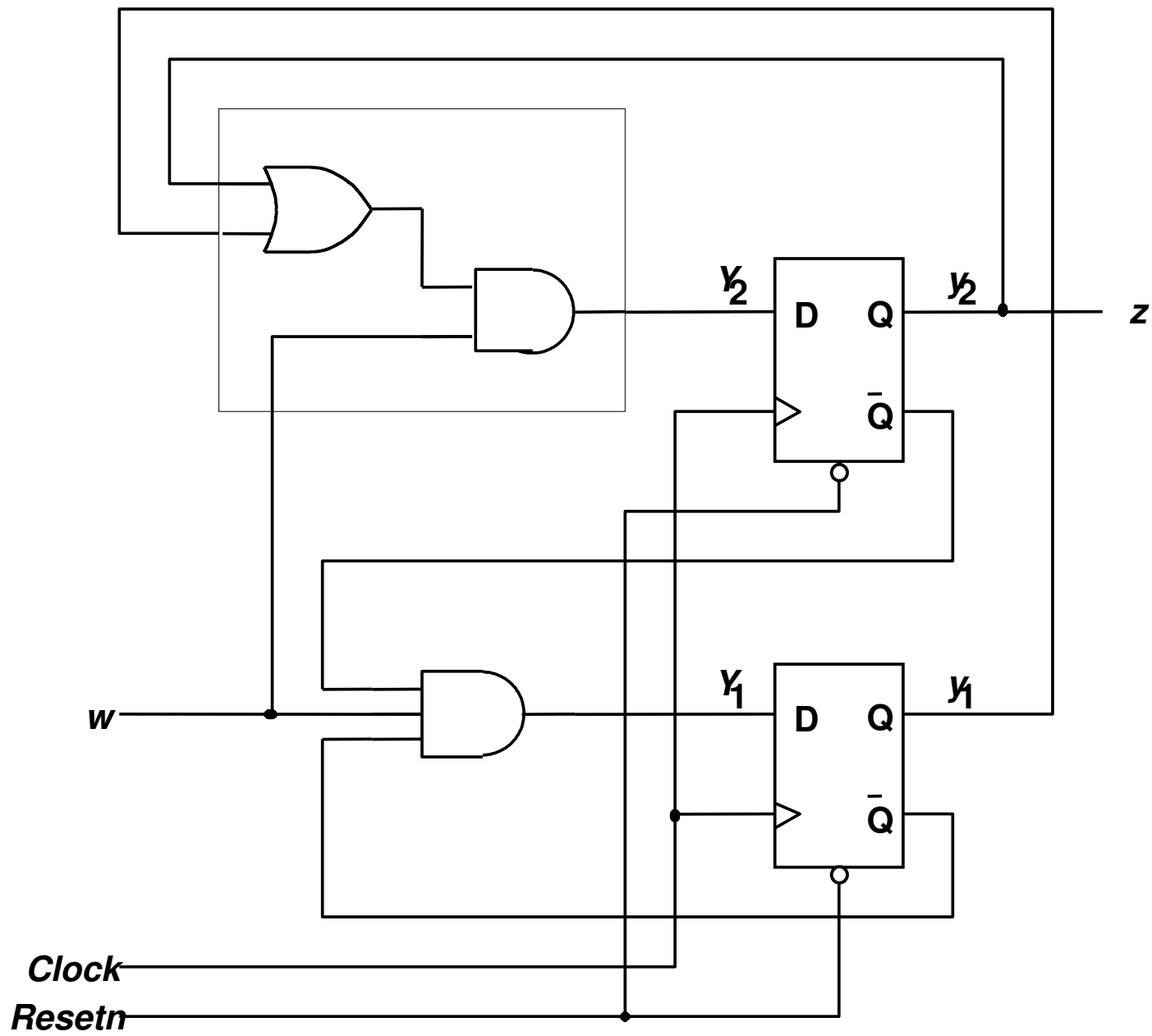
$$Y_2 = wy_1 + wy_2$$

$$= w(y_1 + y_2)$$

		$y_1$	
	$y_2$	0	1
0		0	0
1		1	d

$$z = \bar{y}_1y_2$$

$$z = y_2$$



# Atribuição de Estados

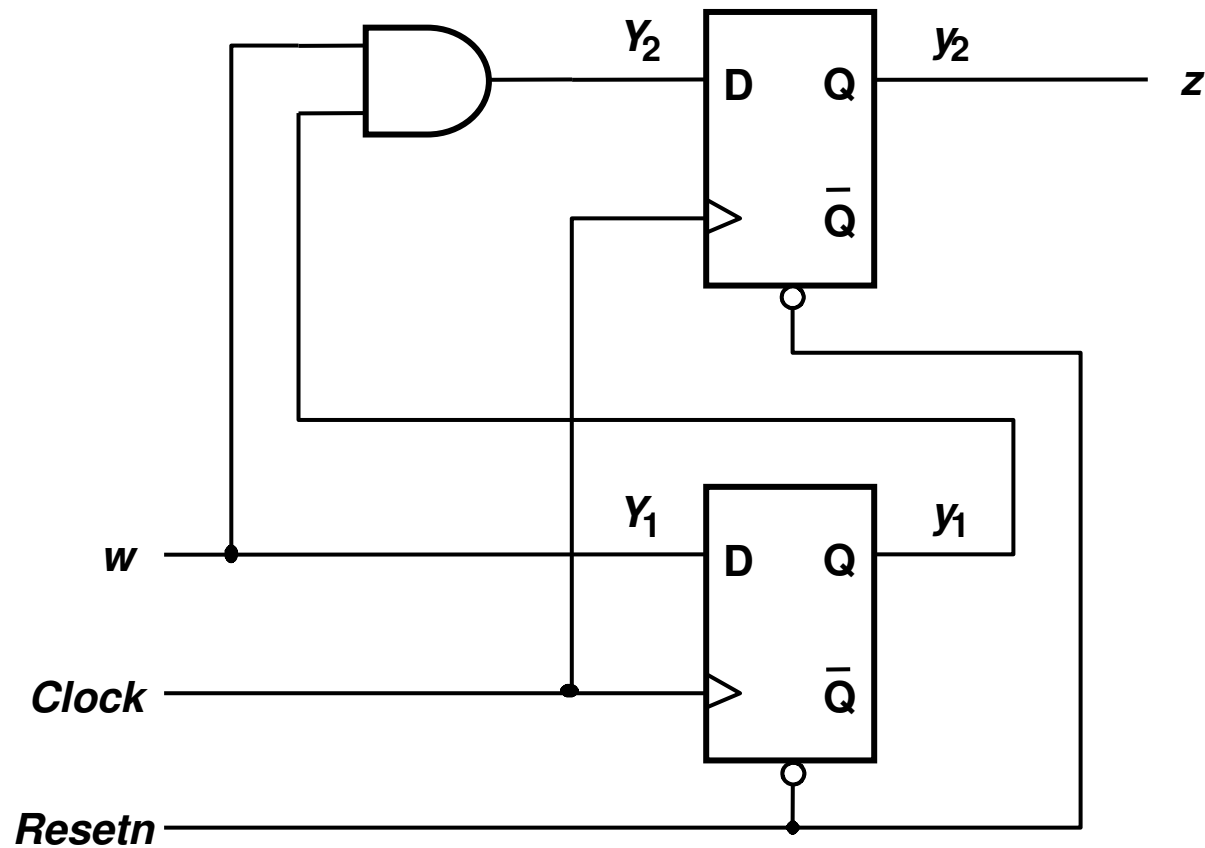
## Existe uma Solução Melhor?

	<b>Present state</b>	<b>Next state</b>		<b>Output</b>
		$w = 0$	$w = 1$	
		$y_2 y_1$	$Y_2 Y_1$	
<b>A</b>	<b>00</b>	<b>00</b>	<b>01</b>	<b>0</b>
<b>B</b>	<b>01</b>	<b>00</b>	<b>11</b>	<b>0</b>
<b>C</b>	<b>11</b>	<b>00</b>	<b>11</b>	<b>1</b>
	<b>10</b>	<i>dd</i>	<i>dd</i>	<i>d</i>



# Atribuição de Estados

## Existe uma Solução Melhor?



# Atribuição de Estados

- Para Circuitos grandes, diferentes Atribuições de Estados, tem um impacto considerável sobre o custo
- Na prática é impossível determinar a melhor atribuição de estados para circuitos grandes.
- Soluções com abordagem baseada em busca exaustiva são impraticáveis.
- Ferramentas de CAD usam, em geral, técnicas baseadas em heurísticas para realizarem a atribuição de estados e os detalhes não são publicados.

## Atribuição de Estados One-Hot Encoding

- Uso de tantas variáveis de Estados quantos forem os Estados.
- Um estado é representado com uma variável igual a 1 e todas as outras em 0
  - A variável igual a 1 é chamada de "hot"

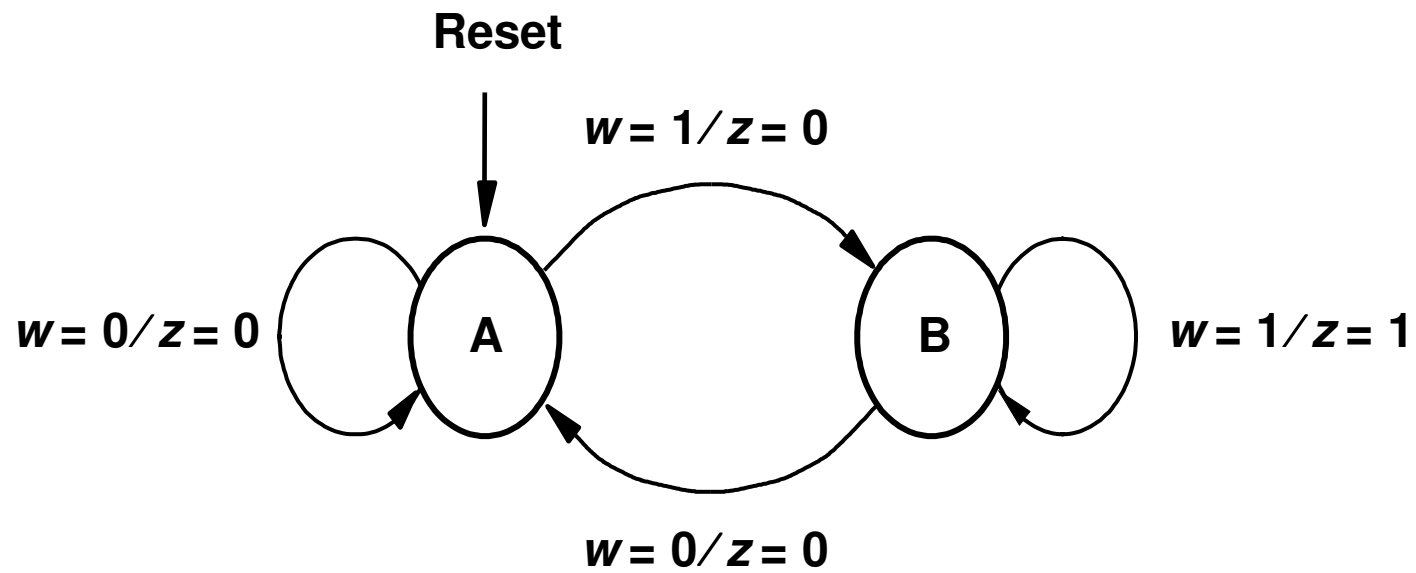
### Exemplo:

	Present state	Nextstate		Output  z
		w = 0	w = 1	
	y <sub>3</sub> y <sub>2</sub> y <sub>1</sub>	Y <sub>3</sub> Y <sub>2</sub> Y <sub>1</sub>	Y <sub>3</sub> Y <sub>2</sub> Y <sub>1</sub>	
A	001	001	010	0
B	010	001	100	0
C	100	001	100	1

# FSM de Mealy

<b>Clock cycle:</b>	<b><math>t_0</math></b>	<b><math>t_1</math></b>	<b><math>t_2</math></b>	<b><math>t_3</math></b>	<b><math>t_4</math></b>	<b><math>t_5</math></b>	<b><math>t_6</math></b>	<b><math>t_7</math></b>	<b><math>t_8</math></b>	<b><math>t_9</math></b>	<b><math>t_{10}</math></b>
<b><math>w</math>:</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>1</b>
<b><math>z</math>:</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>0</b>

# FSM de Mealy



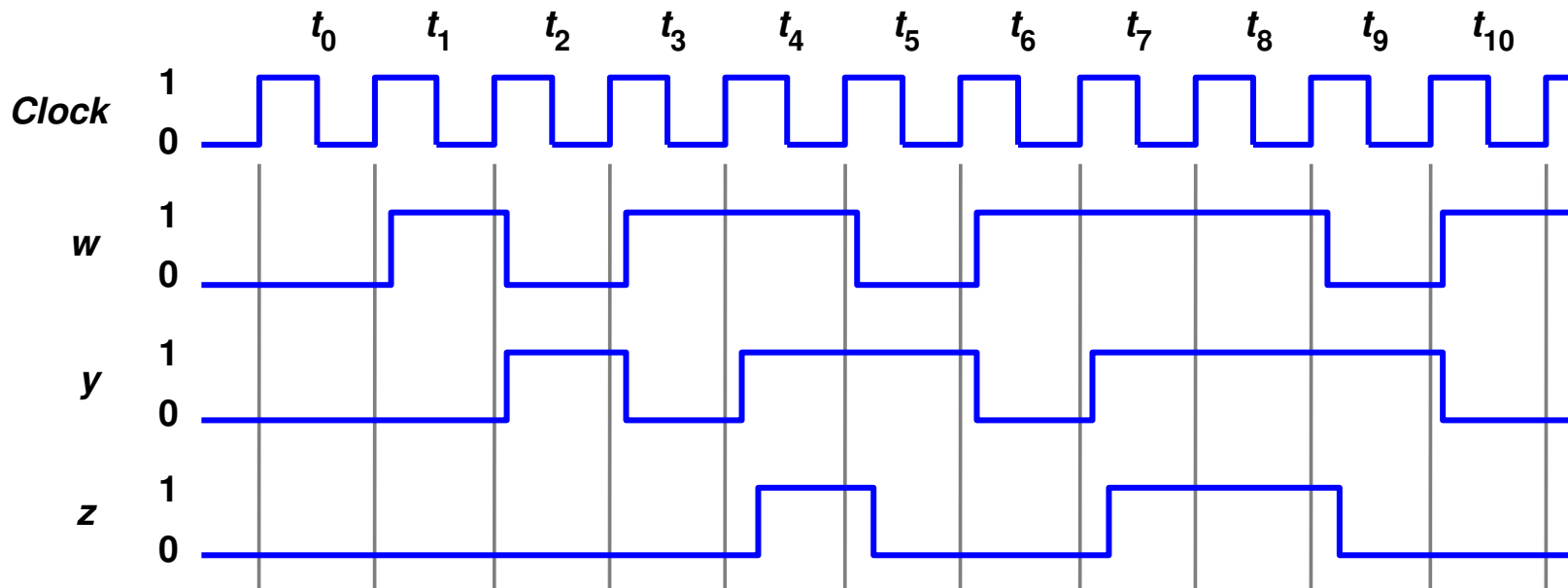
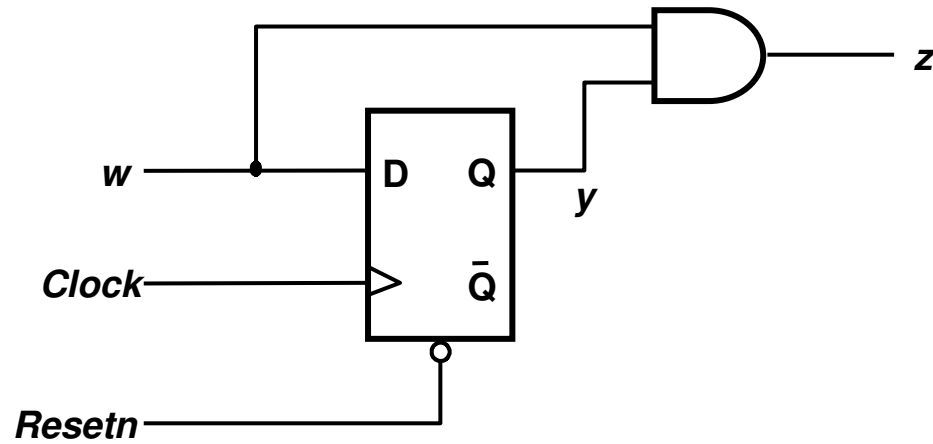
# FSM de Mealy

Present state	Next state		Output $z$	
	$w = 0$	$w = 1$	$w = 0$	$w = 1$
A	A	B	0	0
B	A	B	0	1

# FSM de Mealy

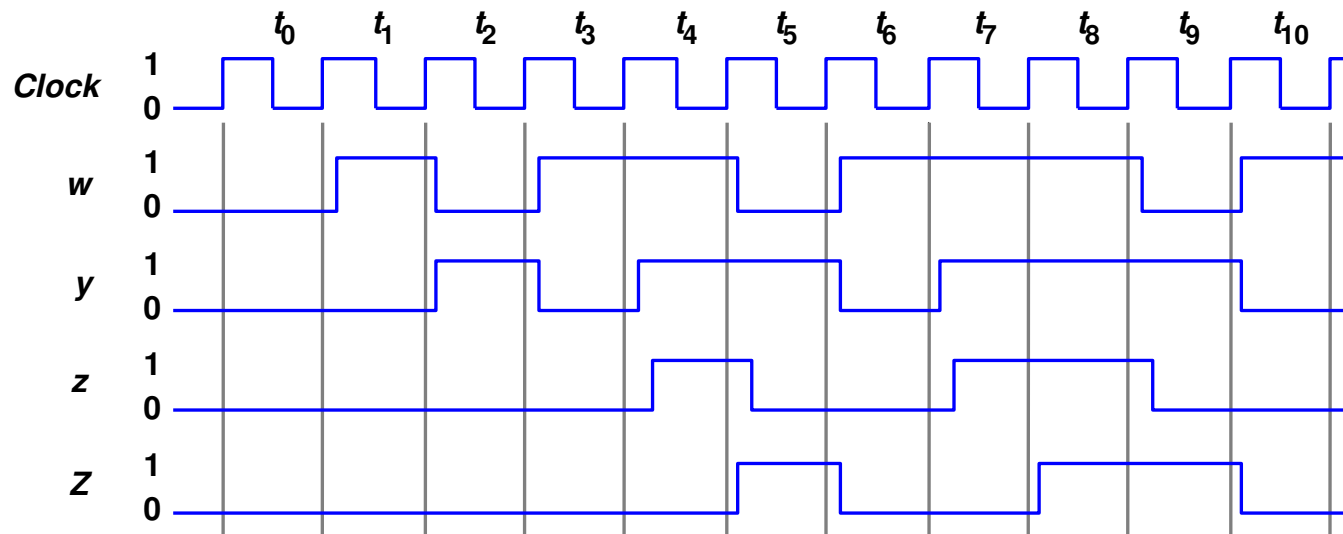
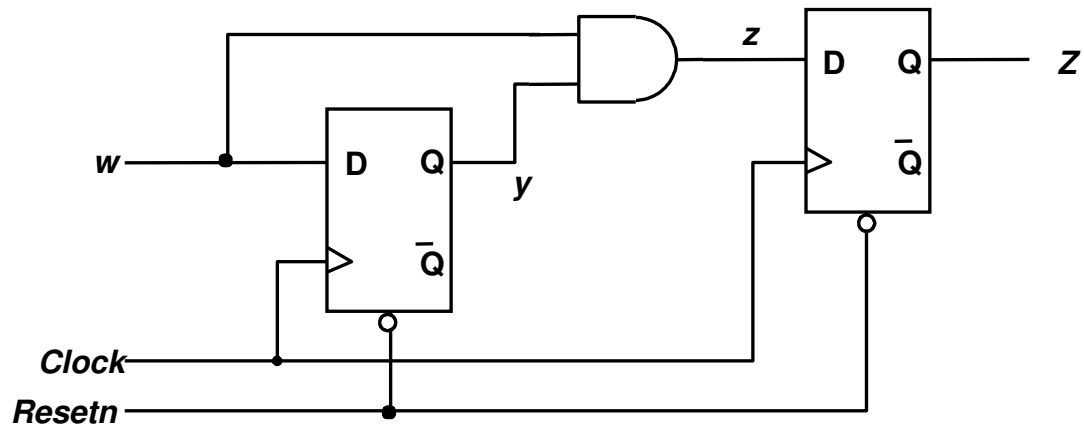
	Present state	Next state		Output	
		$w = 0$	$w = 1$	$w = 0$	$w = 1$
	$y$	$Y$	$Y$	$z$	$z$
<b>A</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>
<b>B</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>1</b>

# FSM de Mealy





# FSM de Mealy Para a Especificação Original



# FSM - Exercícios

- Projetar um contador binário que conte da seguinte forma: 1, 3, 5, 7, 9, 11, 13, 15, 0, 2, 4, 6, 8, 10, 12, 14, 1
  - Projete usando FF JK
  - Projete usando FF T
  - Projete usando FF RS

## Preenchimento do MK para FF JK e RS

	K		J	
0 -> 0	1		0	
	0	X	0	0
0 -> 1	0		1	
	1	X	1	1
1 -> 0	1		0	
	1	1	1	X
1 -> 1	0		1	
	0	0	0	X

	R		S	
0 -> 0	1		0	
	0	X	0	0
0 -> 1	0	0	1	1
	1	1	0	0
1 -> 1	0		1	
	0	0	0	X