



Universidade Federal da Fronteira Sul
Curso de Ciência da Computação
Campus Chapecó

Finite State Machines

Máquinas de Estados

Finitos

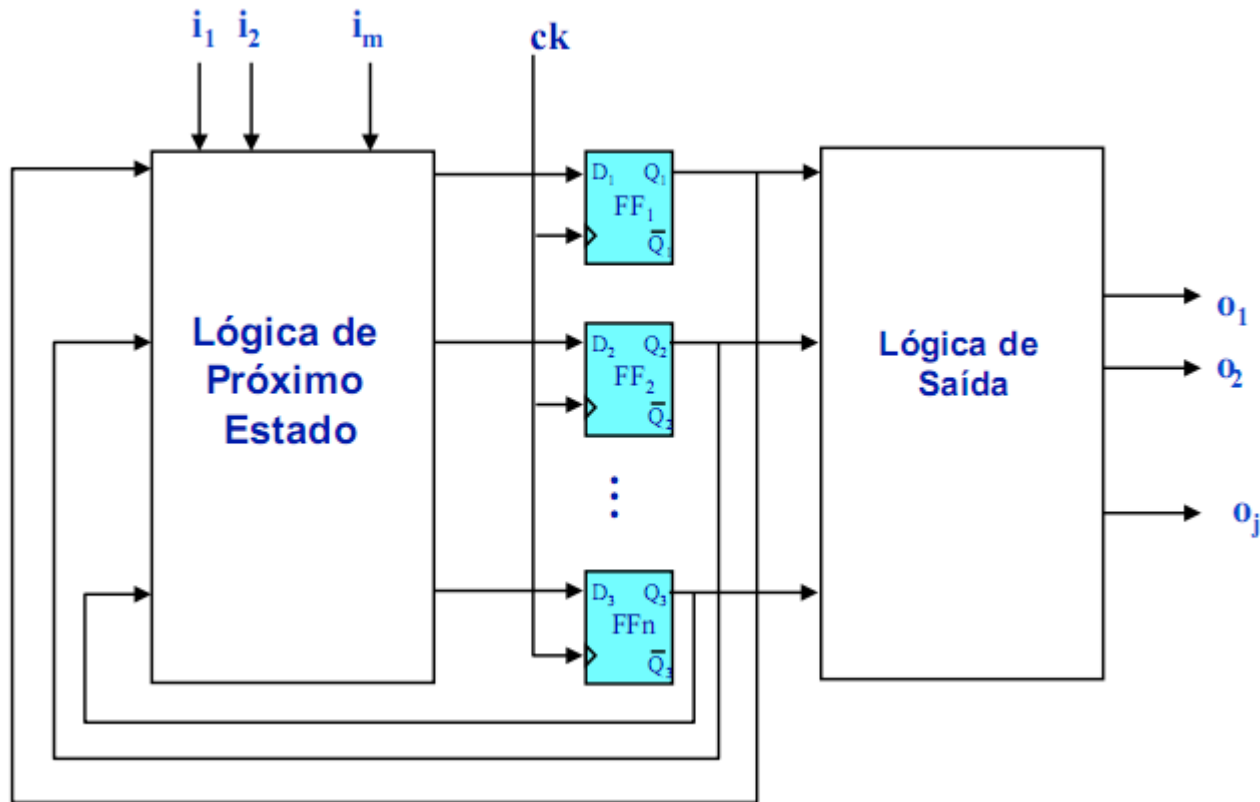
Prof. Luciano L. Caimi
lcaimi@uffs.edu.br

► Máquinas de Estado Finitos

- ✓ Podem ser síncronas (cadenciadas por um sinal monótono chamado relógio ou clock) ou assíncronas (sem relógio).
- ✓ Máquina Sequenciais Síncronas são mais utilizadas porque:
 - São mais fáceis de projetar e de validar.
 - Têm operação mais segura, são mais robustas.
- ✓ Há dois modelos: **Moore** e **Mealy**.
- ✓ Registradores podem ser vistos como Máquina Sequenciais Síncronas.

Máquinas Sequenciais

► Formato geral



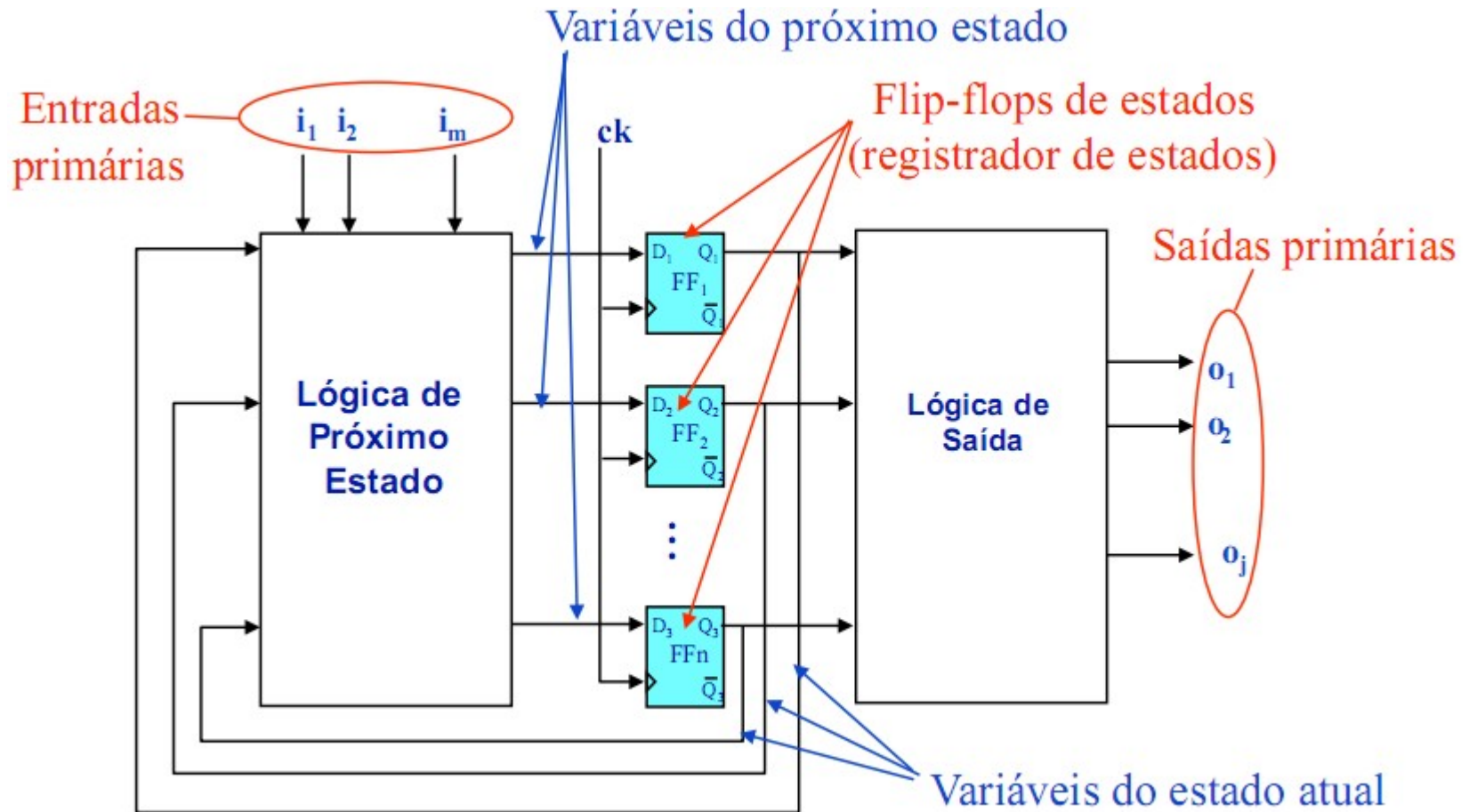
► Máquinas de Estado Finitos

- ✓ Nome dado ao modelo genérico (abstrato) de circuitos sequenciais.
- ✓ Em inglês FSMs: Finite State Machines.
- ✓ O comportamento depende das entradas e do estado em que o circuito se encontra (“estado corrente” ou “estado atual”).
- ✓ O estado corresponde ao valor de um conjunto de variáveis binárias denominadas variáveis de estado.
- ✓ As variáveis de estado ficam armazenadas no registrador de estado.
- ✓ Dado um estado atual e uma combinação de valores de entrada (vetor de entrada), a FSM calcula os valores das saídas (vetor de saída) e o próximo estado.

Máquinas Sequenciais

► Modelo de Moore

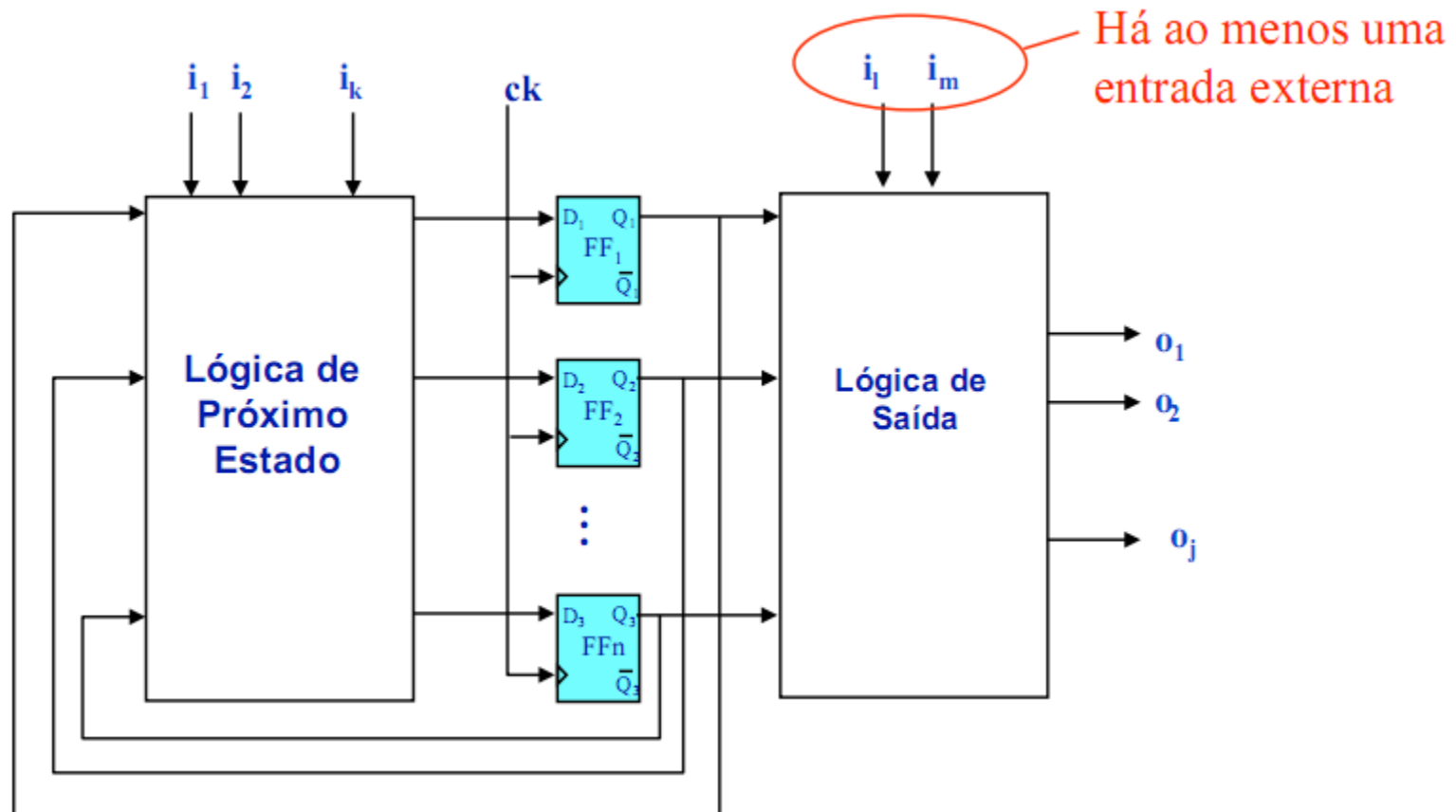
- ✓ Característica principal: as saídas dependem apenas do estado atual.



Máquinas Sequenciais

► Modelo de Mealy

- ✓ Característica principal: as saídas dependem do estado atual e de entrada(s) primária(s).



► Síntese de Circuitos Sequenciais

Roteiro para síntese (= projeto)

1. Determinar quantos estados são necessários (e o nro de variáveis de estado)
2. Construir o diagrama de estados, observando com cuidado o comportamento solicitado para a FSM e adotando um modelo de FSM (Moore ou Mealy):
 - Determinar as transições entre estados necessárias
 - Escolher um estado inicial
3. Construir a tabela de próximo estado e a tabela das saídas
4. Sintetizar (projetar) os circuitos combinacionais: lógica de próximo estado e lógica de saída.

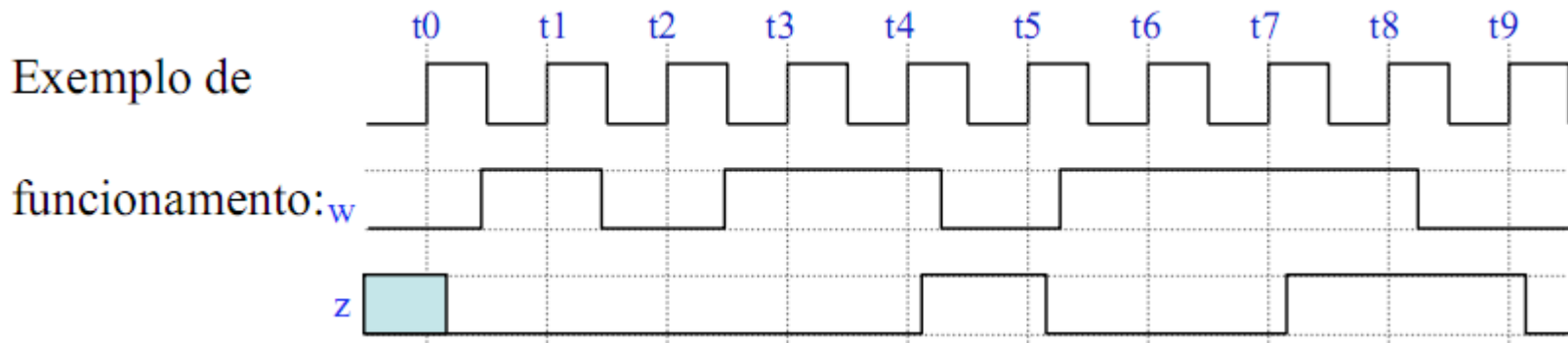
Máquinas Sequenciais



► Síntese de Circuitos Sequenciais

Exemplo: 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”. 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”. As mudanças de z estão sincronizadas com a borda de relógio ativa.

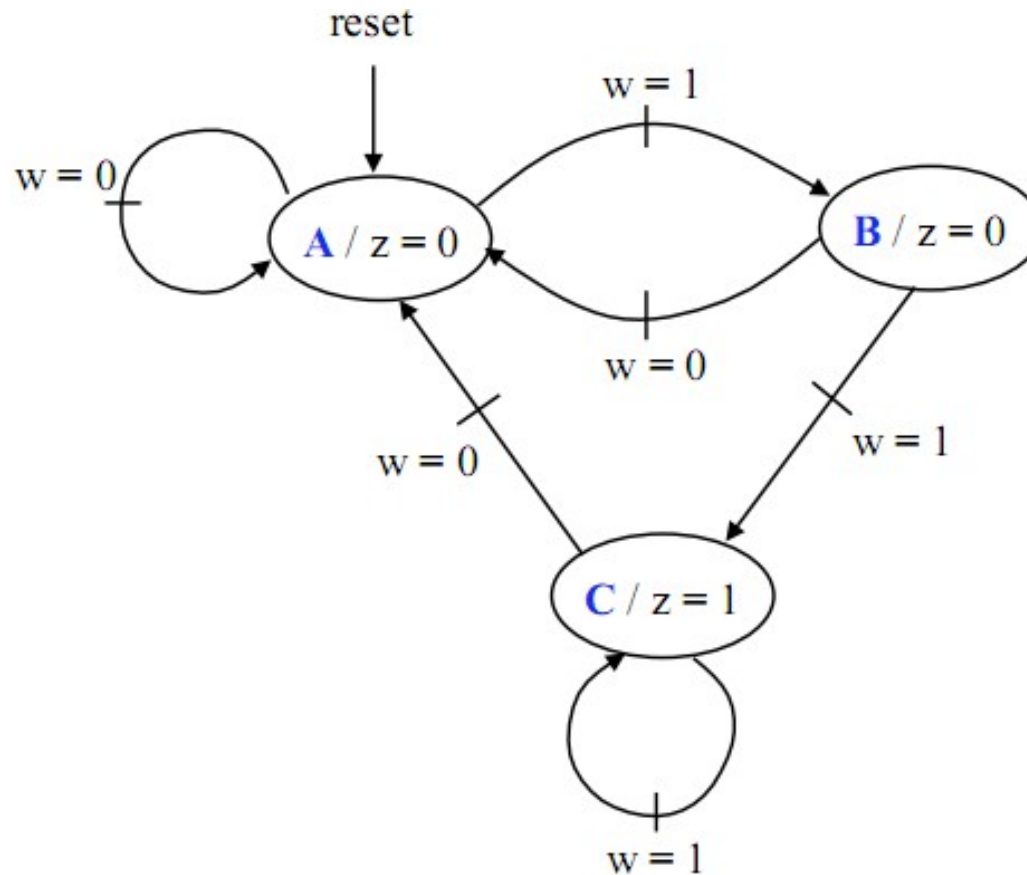


Máquinas Sequenciais



► Síntese de Circuitos Sequenciais

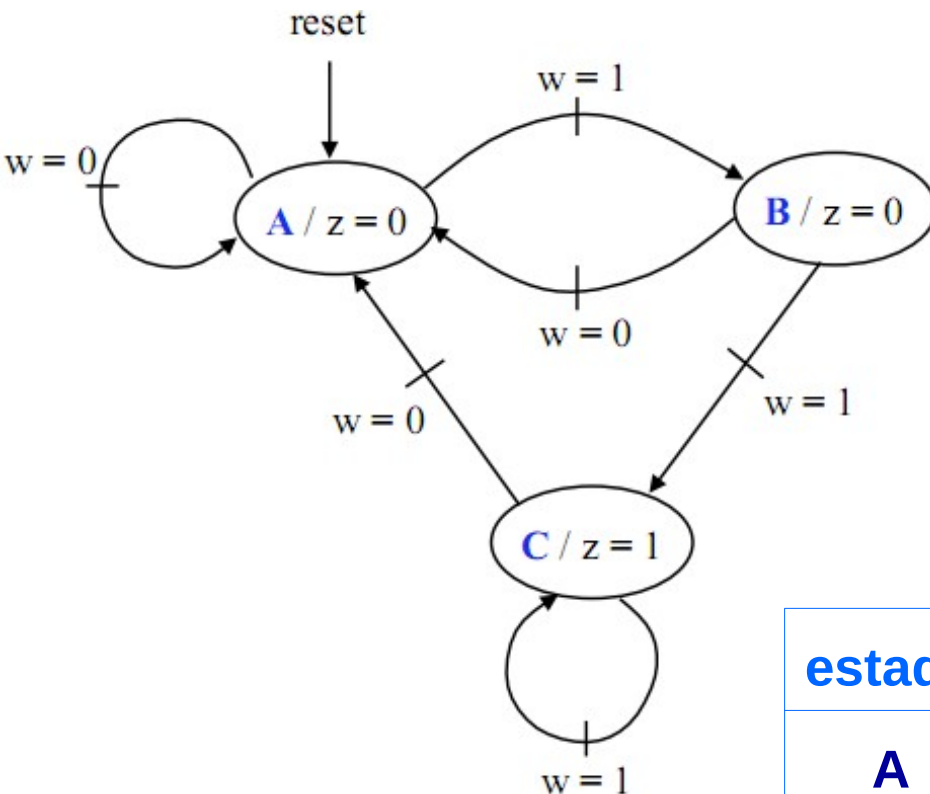
Diagrama de estados



Máquinas Sequenciais

► Síntese de Circuitos Sequenciais

Tabelas de transição de estado e de saída



estado atual	W	próximo estado
A	0	A
A	1	B
B	0	A
B	1	C
C	0	A
C	1	C

estado	z
A	0
B	0
C	1

Máquinas Sequenciais



► Síntese de Circuitos Sequenciais

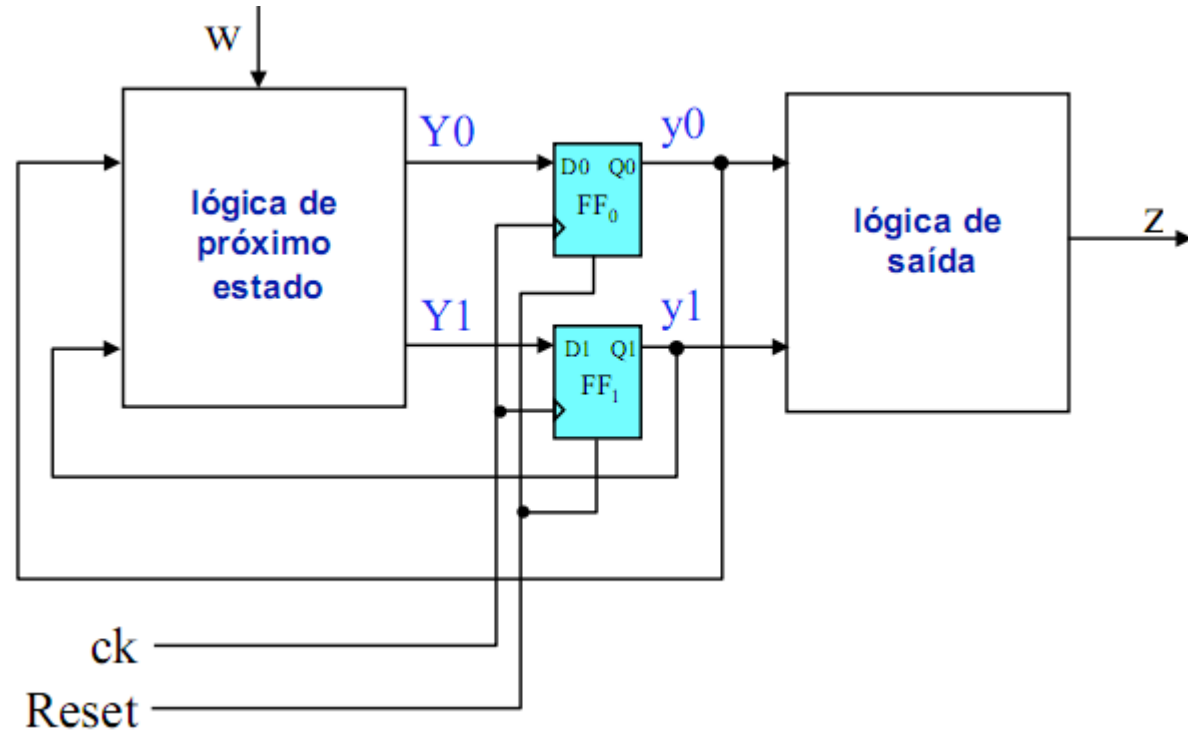
Diagrama de Blocos

Sinais de entrada: w

Sinais de saída: z

Número de estados: 3

⇒ Logo, são necessário
2 FF para armazenar
as variáveis de estado



Máquinas Sequenciais

► Síntese de Circuitos Sequenciais

Codificação de Estados

Supondo a seguinte codificação: 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	W	próximo estado	
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	-

Máquinas Sequenciais

► Síntese de Circuitos Sequenciais

Projetando a lógica de próximo estado

y_1y_0	w	Y_1Y_0
00	0	00
00	1	01
01	0	00
01	1	10
10	0	00
10	1	10
11	0	XX
11	1	XX

$Y_1)$		$\overline{y_0}$		y_0	
		$\overline{y_1}$			
	$\overline{y_1}$	0	0	1	0
	y_1	0	1	x	x
		\overline{w}	w	\overline{w}	
$Y_0)$		$\overline{y_0}$		y_0	
		$\overline{y_1}$			
	$\overline{y_1}$	0	1	0	0
	y_1	0	0	x	x
		\overline{w}	w	\overline{w}	

Máquinas Sequenciais



► Síntese de Circuitos Sequenciais

Projetando a lógica de saída

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

$$z = y_1$$

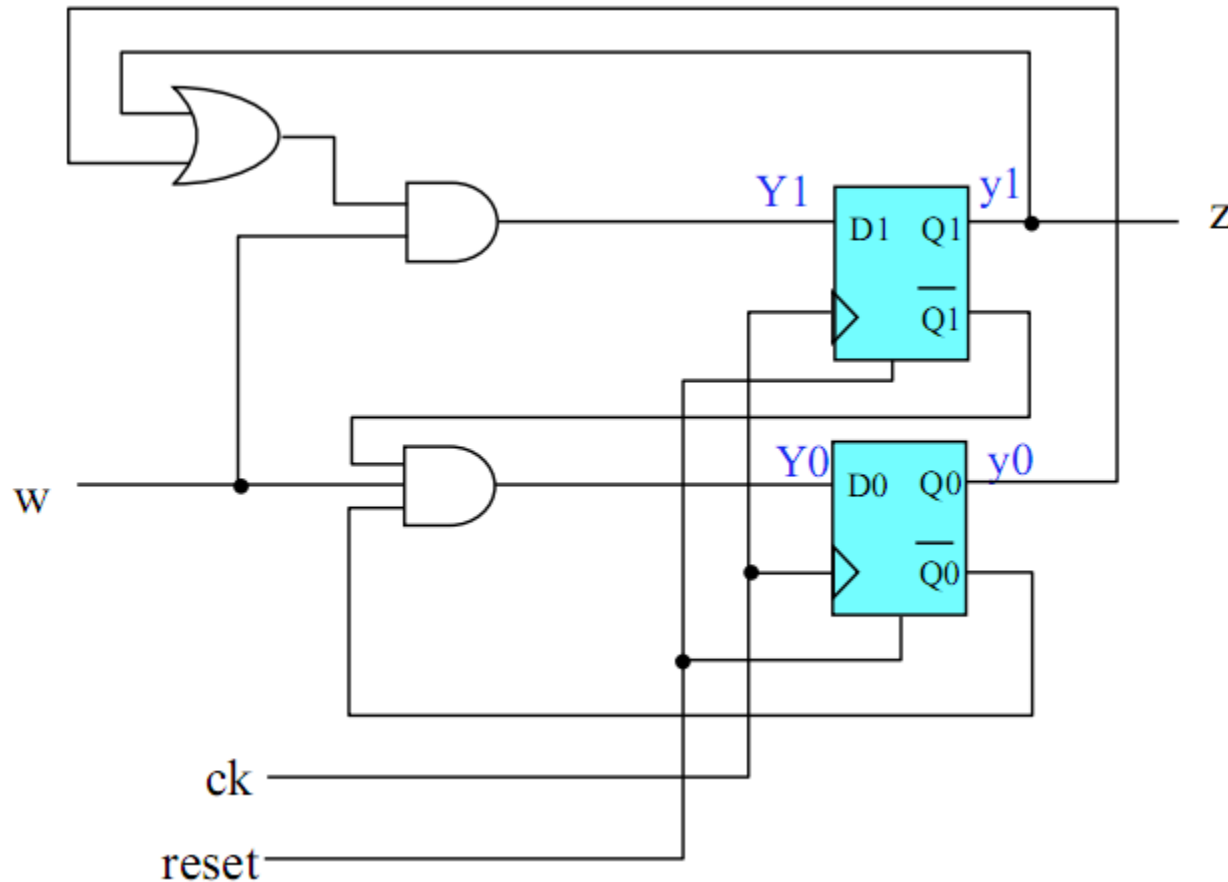
z)

	$\overline{y_0}$	y_0
$\overline{y_1}$	0	0
y_1	1	X

Máquinas Sequenciais

► Síntese de Circuitos Sequenciais

Circuito Final



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

► Síntese de Circuitos Sequenciais

Roteiro **Revisitado** para a Síntese (= Projeto)

1. Determinar quantos estados são necessários (e o nro de variáveis de estado)
2. Construir o diagrama de estados, observando com cuidado o comportamento solicitado para a FSM e adotando um modelo de FSM (Moore ou Mealy):
 - Determinar as transições entre estados necessárias
 - Selecionar um estado para servir como estado inicial
3. Construir a tabela de próximo estado e das saídas
4. Escolher uma codificação para os estados e definir o tipo de flip-flops para compor o registrador de estados.
5. Sintetizar (projetar) os circuitos combinacionais: lógica de próximo estado e lógica de saída.

Máquinas Sequenciais



► Exemplo de FSM em VHDL

```
1  LIBRARY ieee ;
2  USE ieee.std_logic_1164.all ;
3  ENTITY contabits1 IS
4      PORT ( Clock, Reset, w : IN STD_LOGIC ;
5              z : OUT STD_LOGIC ) ;
6  END contabits1 ;
7  ARCHITECTURE Behavior OF contabits1 IS
8      TYPE Tipo_estado IS (A, B, C) ;
9      SIGNAL y : Tipo_estado ;
10 BEGIN
11     PROCESS ( Reset, Clock )
12     BEGIN
13         IF Reset = '1' 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 ;
23             END CASE ;
24         END IF ;
25     END PROCESS ;
26 END ARCHITECTURE Behavior OF contabits1 ;
```

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 simbólicos: A, B, C

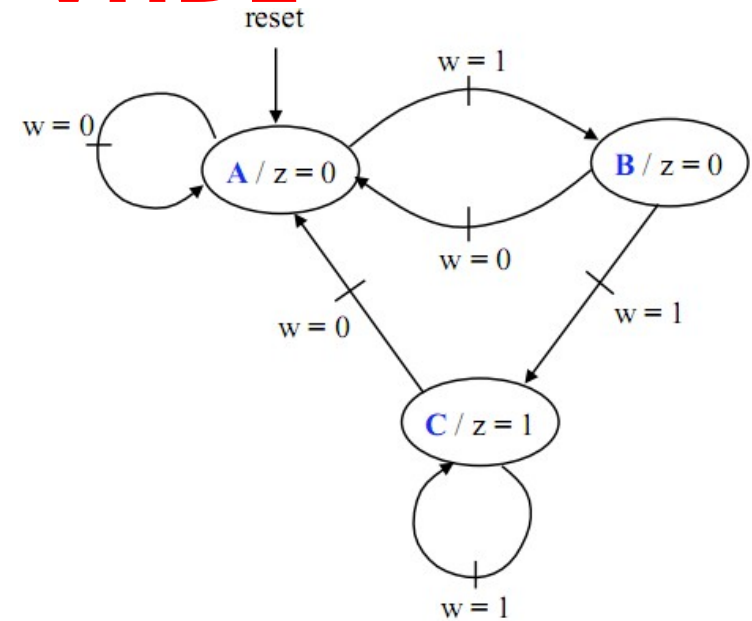
O sinal y representa as saídas dos flip-flops que armazenam os estados desta FSM.

Máquinas Sequenciais



► Exemplo de FSM em VHDL

```
17      WHEN A =>
...      .....
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 ;
```



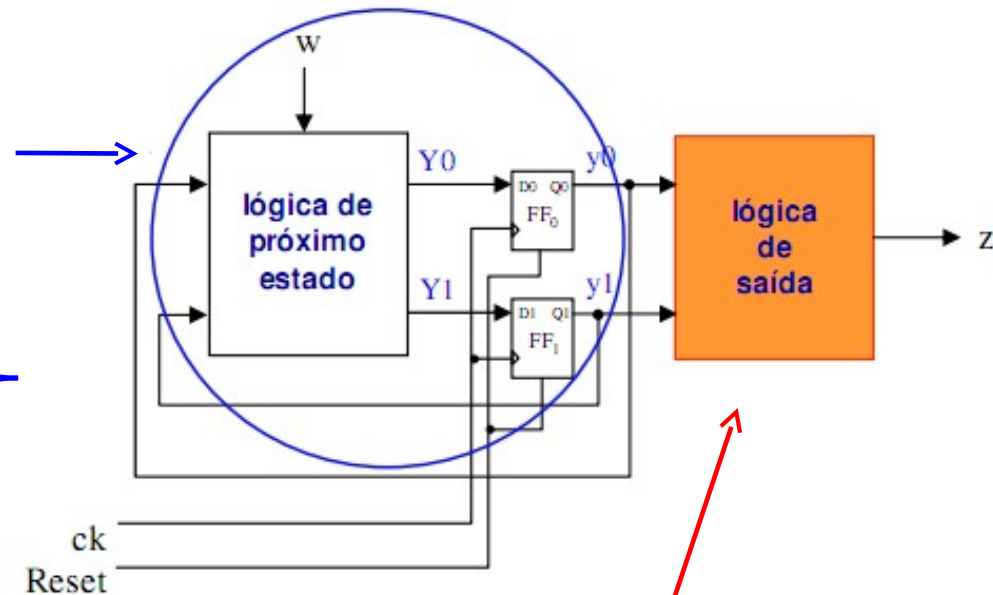
Pode-se utilizar um processo apenas para implementar a lógica de saída

```
PROCESS ( y )
BEGIN
    z <= '1' WHEN y = C ELSE '0' ;
END PROCESS;
```

Máquinas Sequenciais

► Exemplo de FSM em VHDL

```
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 ;
```



Máquinas Sequenciais

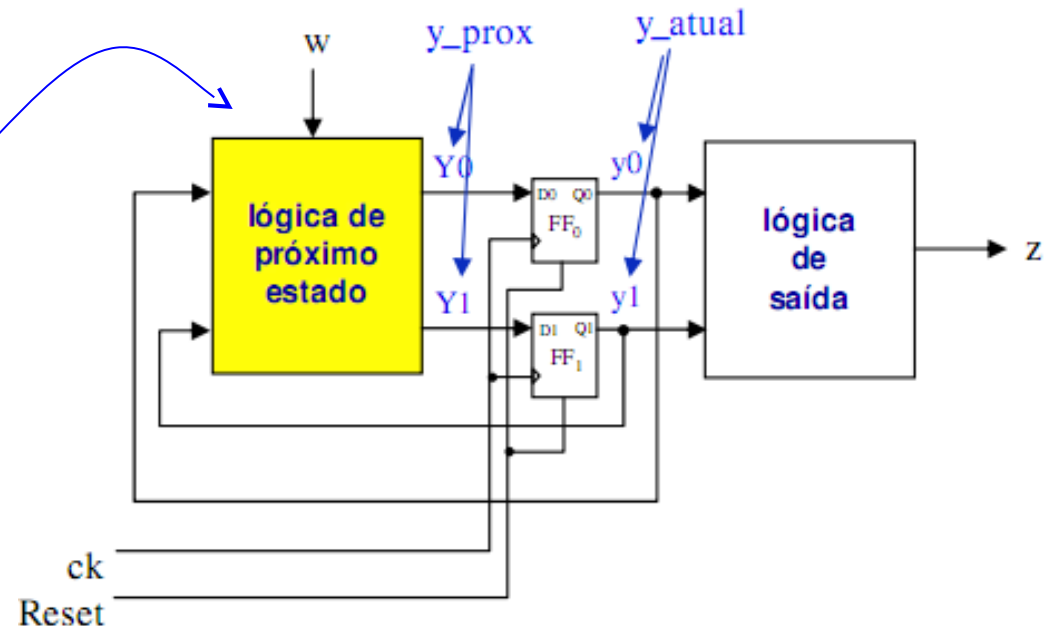
```
ARCHITECTURE Behavior OF contabits1 IS
  TYPE Tipo_estado IS (A, B, C) ;
  SIGNAL y_atual, y_prox : Tipo_estado ;
BEGIN
  PROCESS ( w, y_atual )
  BEGIN
    CASE y_atual IS
      WHEN A =>
        IF w = '0' THEN
          y_prox <= A ;
        ELSE
          y_prox <= B ;
        END IF ;
      WHEN B =>
        IF w = '0' THEN
          y_prox <= A ;
        ELSE
          y_prox <= C ;
        END IF ;
      WHEN C =>
        IF w = '0' THEN
          y_prox <= A ;
        ELSE
          y_prox <= C ;
        END IF ;
    END CASE ;
  END PROCESS ;
```

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

Em termos da notação que usamos:

$y_atual \rightarrow y$ (estado atual)

$y_prox \rightarrow Y$ (próximo estado)



Máquinas Sequenciais

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

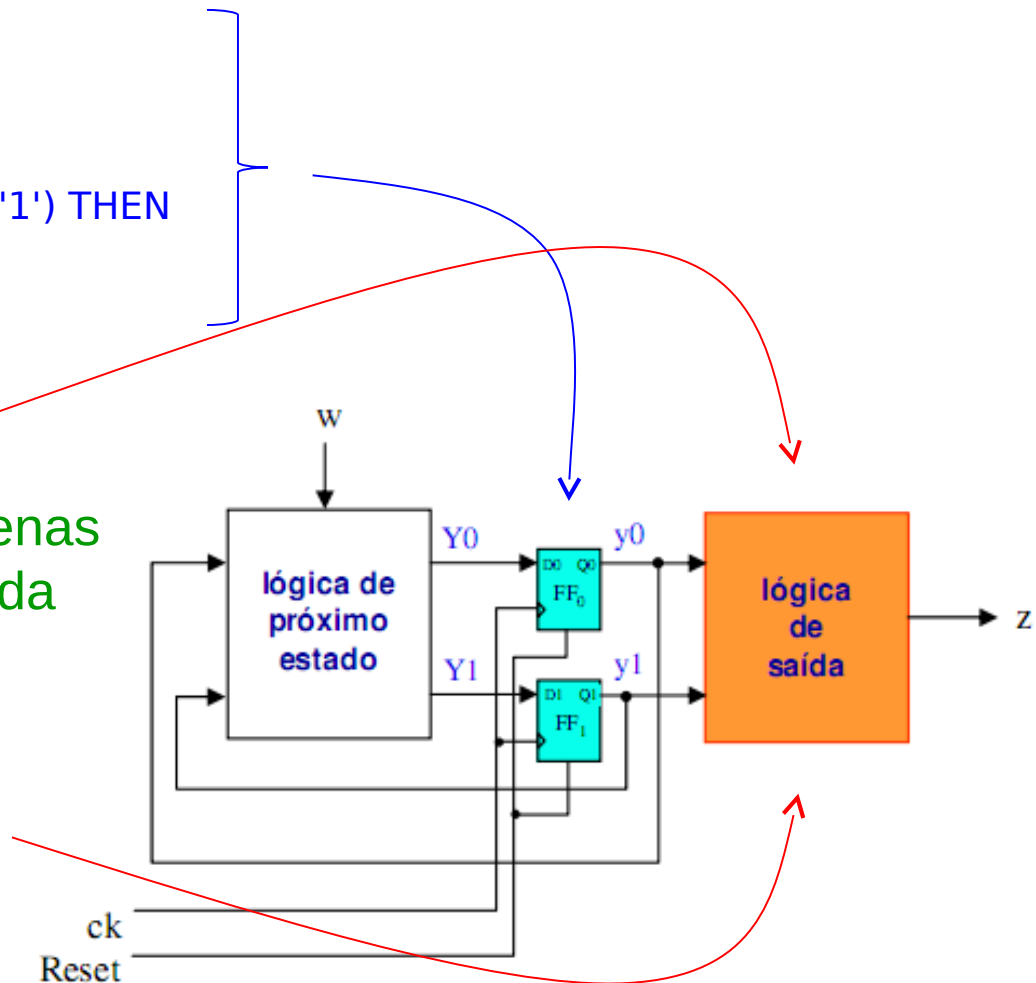
```
PROCESS (Clock, Reset)
BEGIN
  IF Reset = '1' THEN
    y_atual <= A ;
  ELSIF (Clock'EVENT AND Clock = '1') THEN
    y_atual <= y_prox ;
  END IF ;
END PROCESS ;
```

```
z <= '1' WHEN y_atual = C ELSE '0' ;
```

```
END Behavior ;
```

Pode-se utilizar um processo apenas para implementar a lógica de saída

```
PROCESS (y)
BEGIN
  if y = C then
    z <= '1';
  else
    z <= '0';
  end if;
END PROCESS;
```



► Exercício

Construir uma FSM que implementa um contador crescente/decrescente de módulo 5. O circuito possui uma entrada S (sentido) que indica se a contagem é crescente ($S=0$) ou decrescente ($S=1$). As saídas são 3 bits com o valor da contagem.

► Síntese de Circuitos Sequenciais

Exemplo: 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. 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

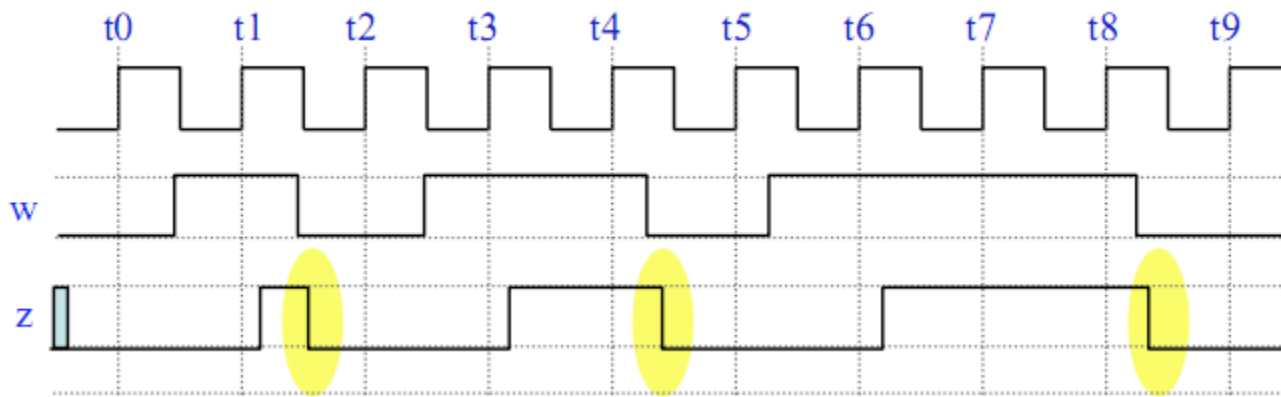
Máquinas Sequenciais



► Exemplo (mealy)

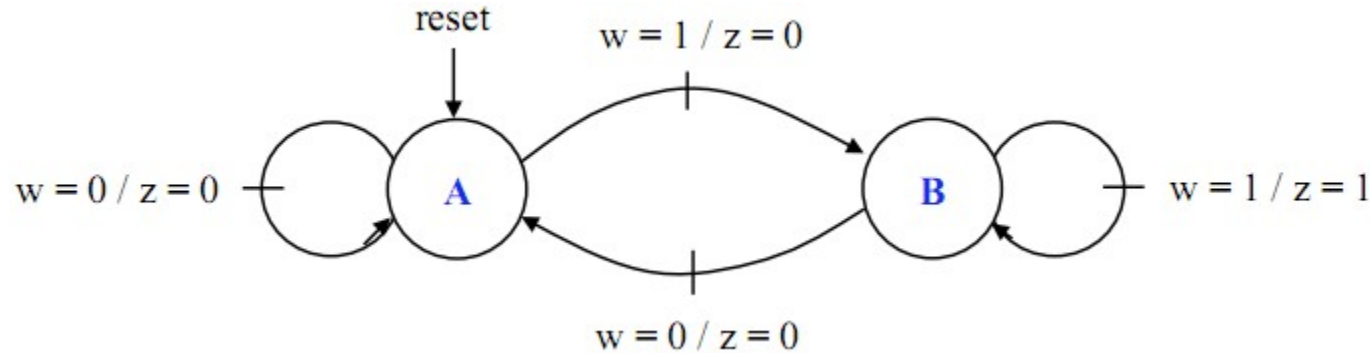
Considere a seguinte modificação da especificação acima:

- ✓ 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 Sequenciais

► Exemplo (mealy)



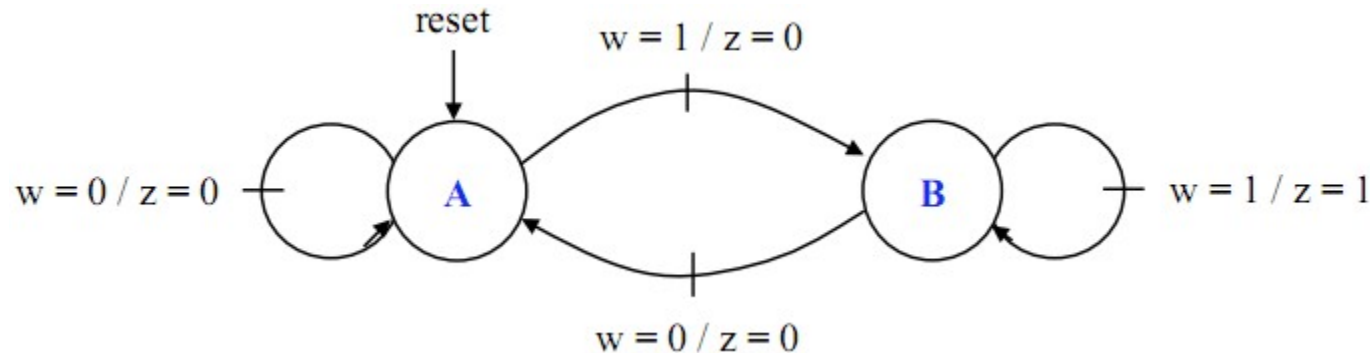
Interpretação do Diagrama de Estados:

- ✓ 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 Sequenciais

► Exemplo (mealy)

Tabelas de transição de estado e de saída



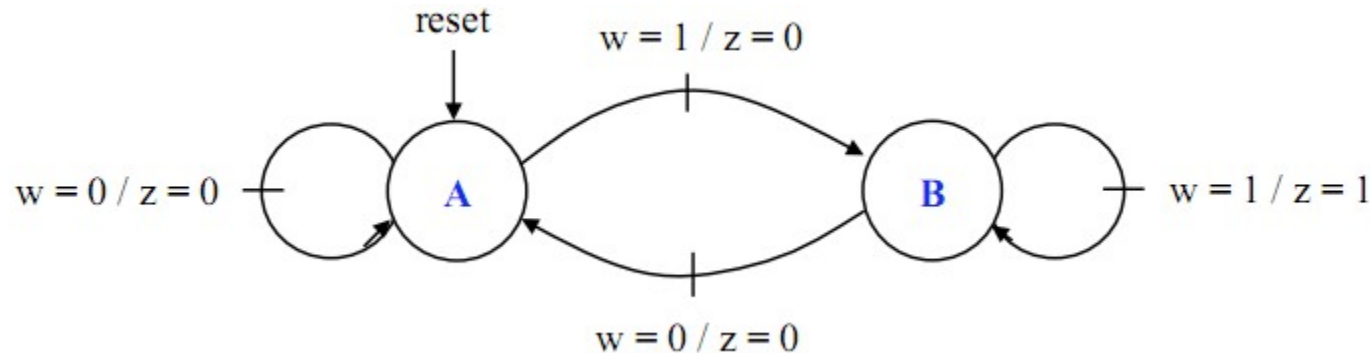
estado atual	W	próximo estado
A	0	A
A	1	B
B	0	A
B	1	B

estado	W	z
A	0	0
A	1	0
B	0	0
B	1	1

Máquinas Sequenciais

► Exemplo (mealy)

Utilizando um bit para codificar o estado



estado atual	y	w	Y	próximo estado
A	0	0	0	A
A	0	1	1	B
B	1	0	0	A
B	1	1	1	B

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

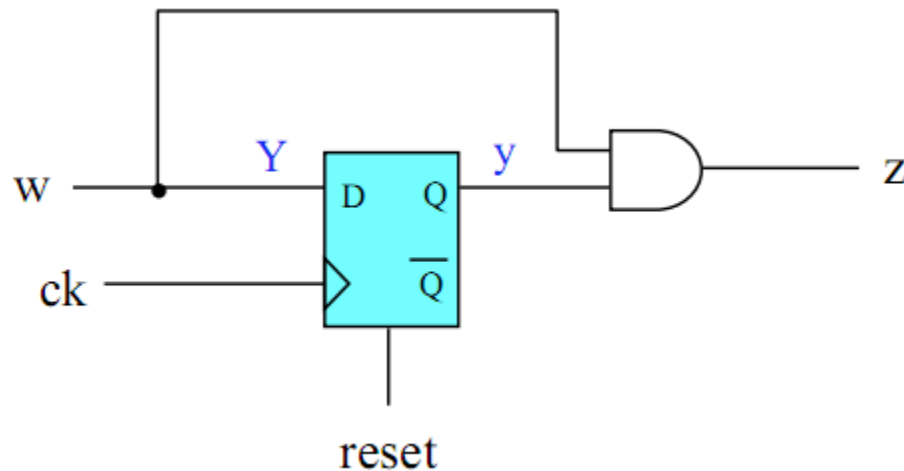
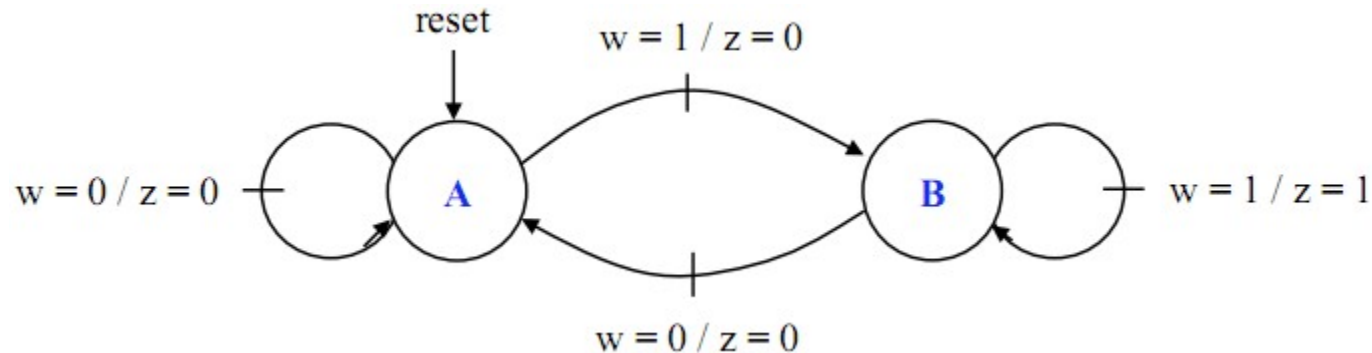
estado	y	w	z
A	0	0	0
A	0	1	0
B	1	0	0
B	1	1	1

$$z = y \cdot w$$

Máquinas Sequenciais

► Exemplo (mealy)

Circuito Final



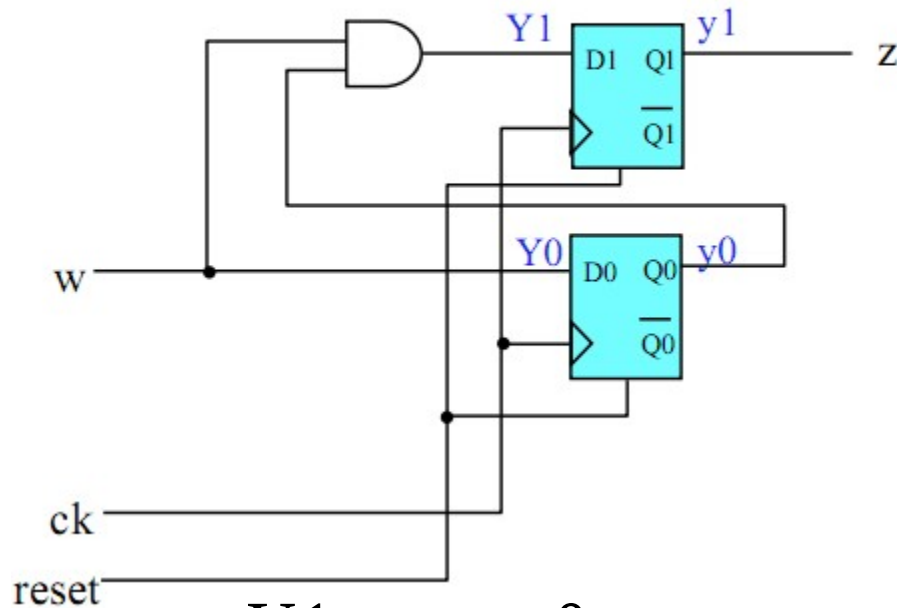
$$y = \overline{y} \cdot w + y \cdot w = w$$

$$Z = y \cdot w$$

Máquinas Sequenciais

► Comparando Moore e Mealy

Moore

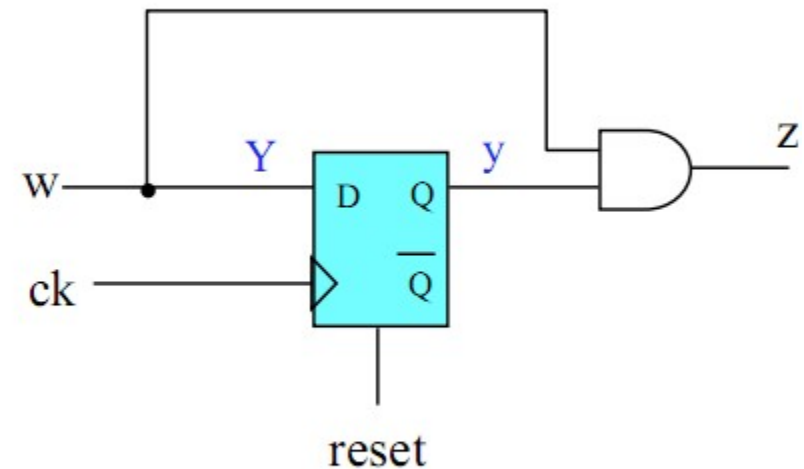


$$Y1 = w + y0$$

$$Y0 = w$$

$$z = y1$$

Mealy



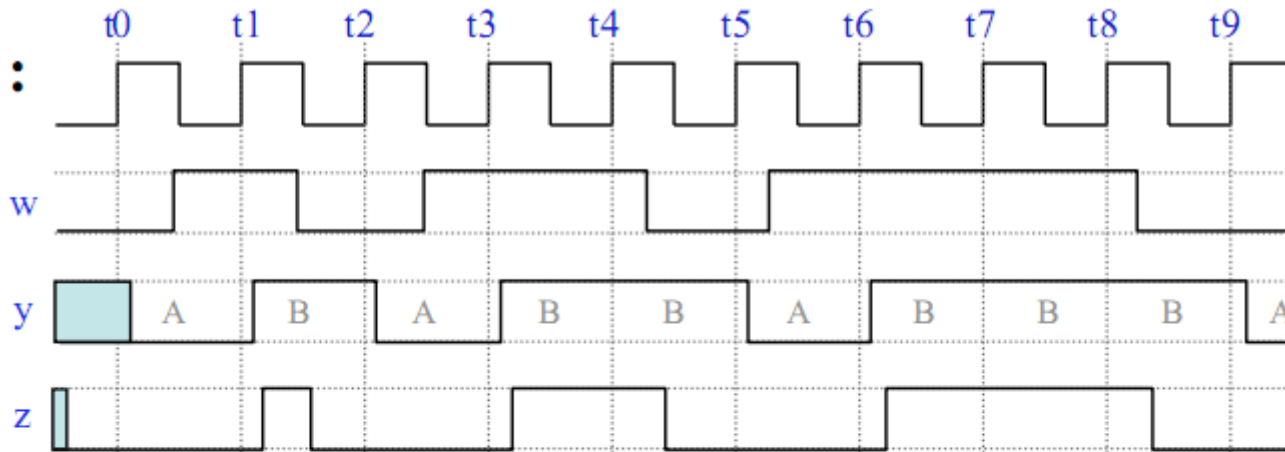
$$Y = w$$

$$z = y \cdot w$$

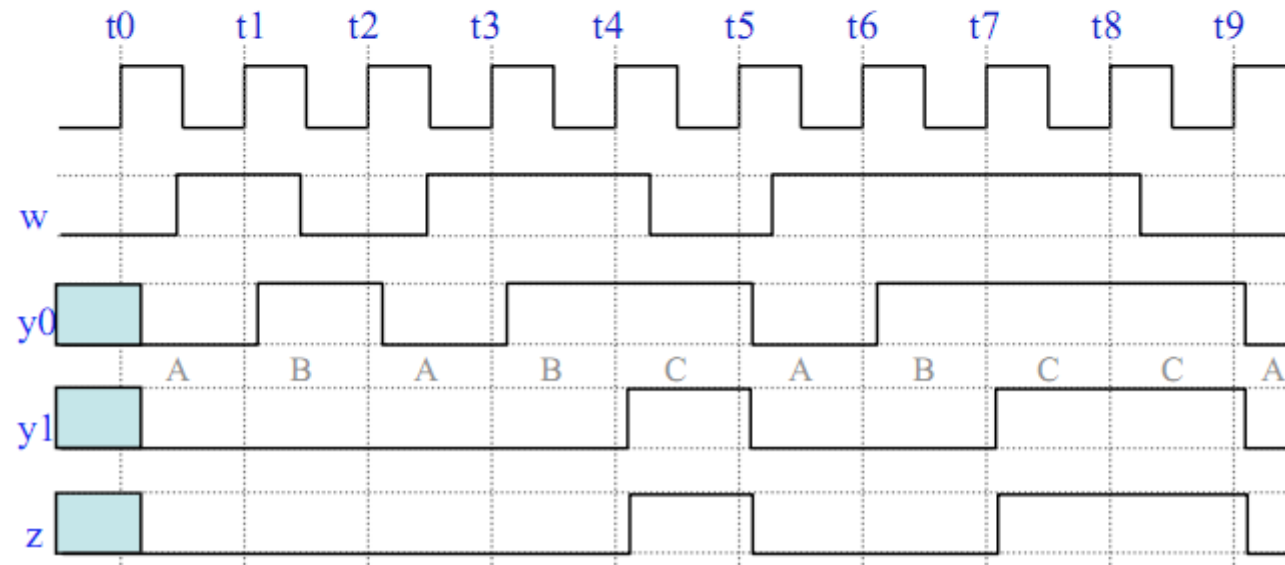
Máquinas Sequenciais

► Comparando Moore e Mealy

Mealy

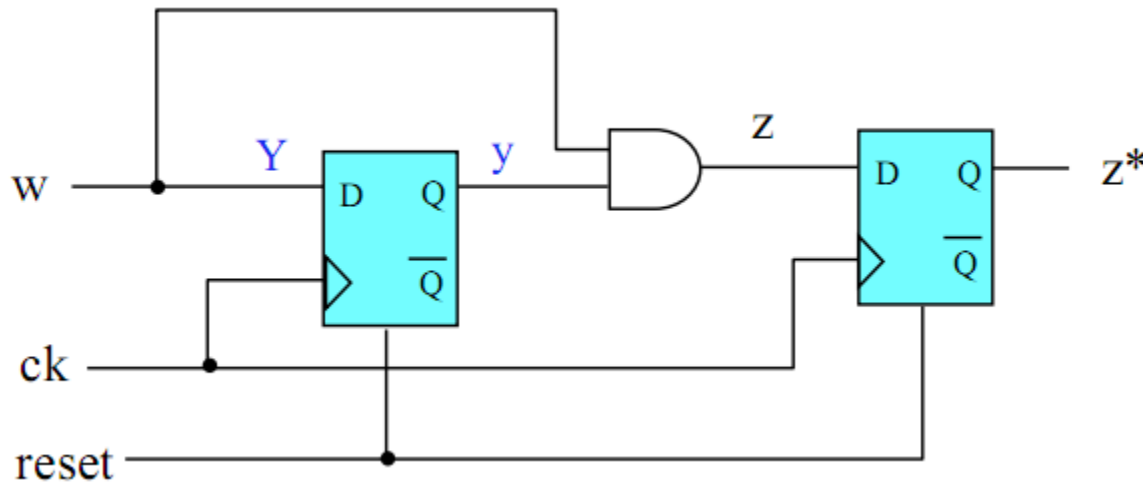


Moore



Máquinas Sequenciais

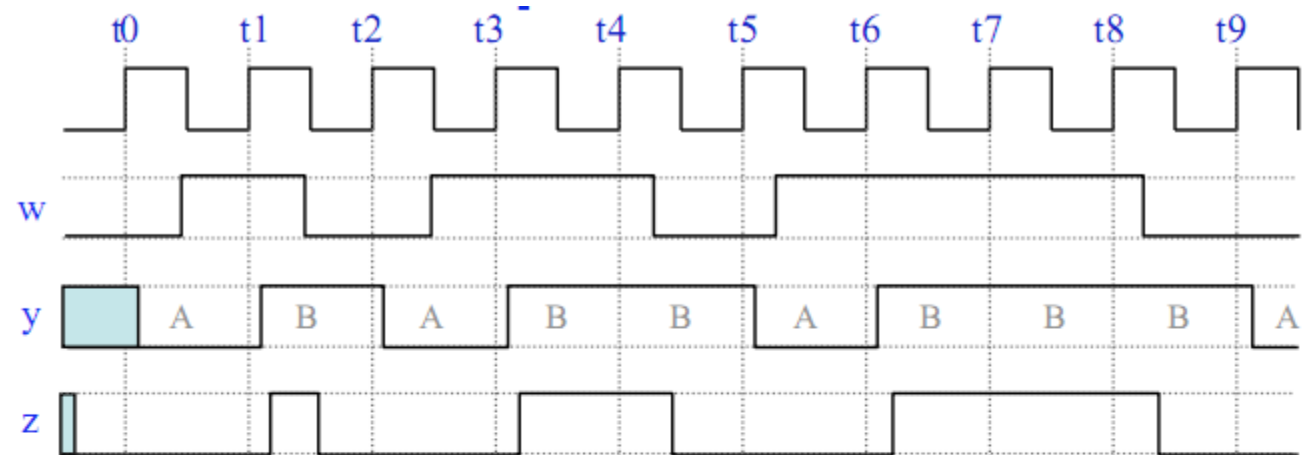
Porém, se passarmos a saída z por um segundo flip-flop, filtraremos o comportamento assíncrono. De fato, estaremos transformando o circuito para o Modelo de Moore...



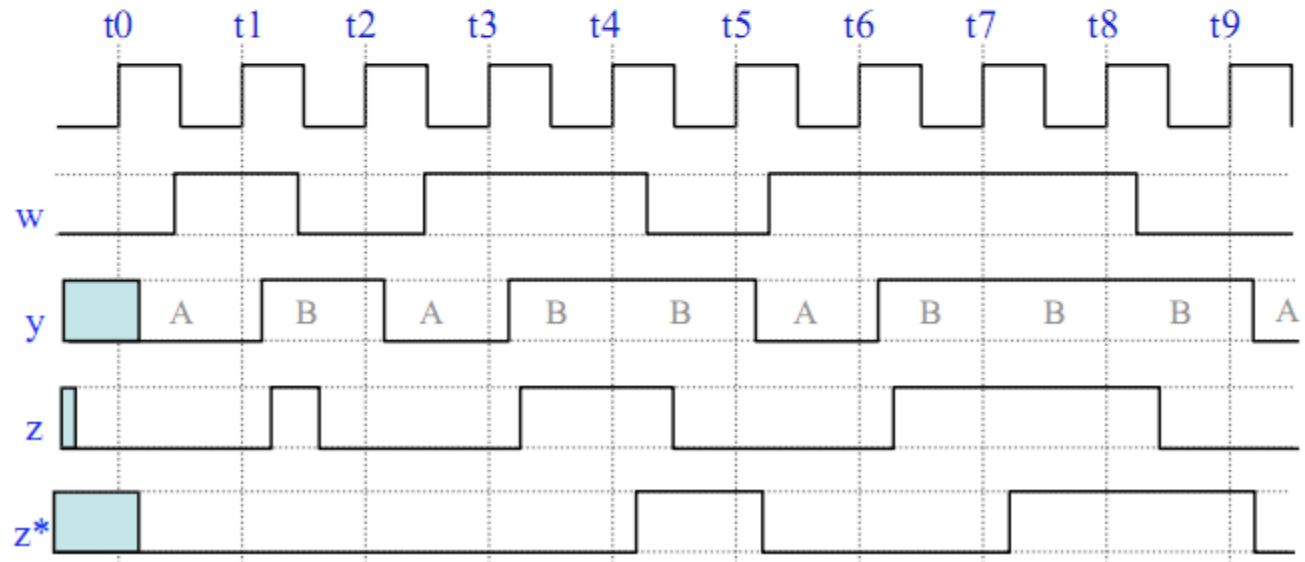
Máquinas Sequenciais

► Comparando Moore e Mealy

Mealy



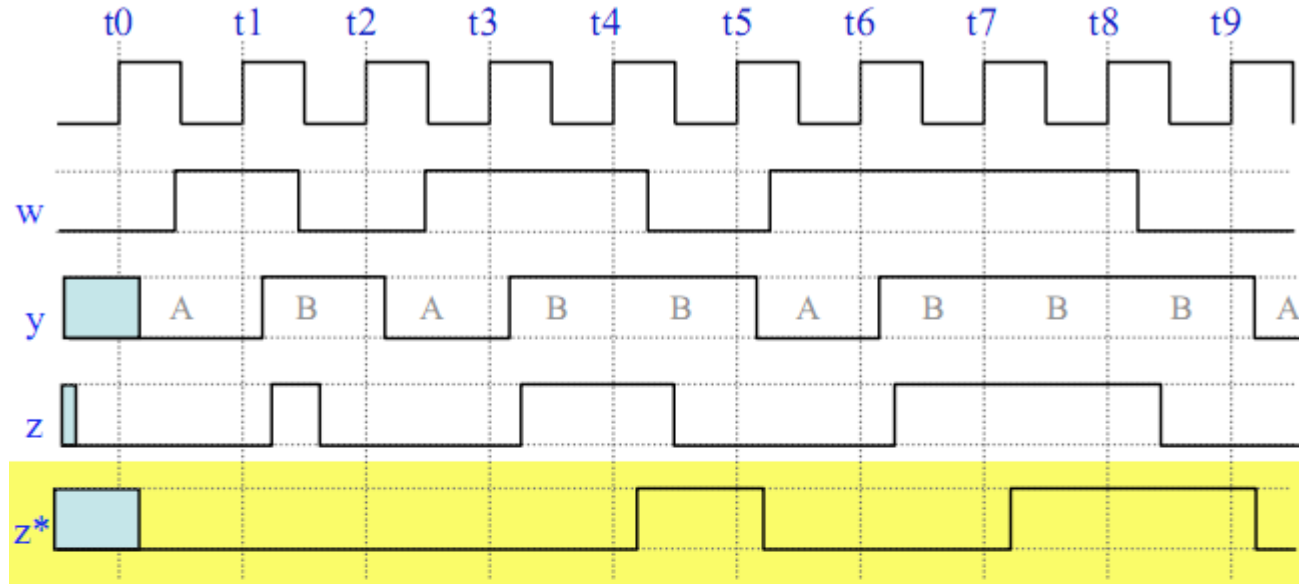
**Mealy transf.
em Moore**



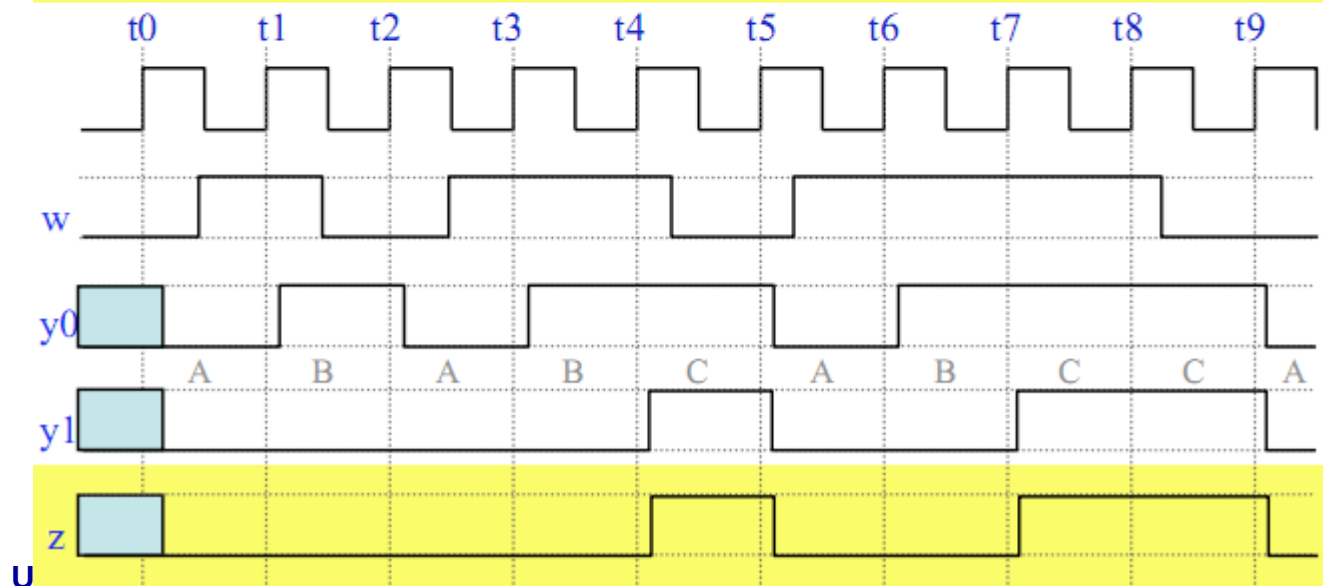
Máquinas Sequenciais

► Comparando Moore e Mealy

Mealy transf.
em Moore



Moore



Máquinas Sequenciais



ENTITY contabits2 IS

PORT (Clock, Reset, w : IN STD_LOGIC ;
z : OUT STD_LOGIC) ;

END contabits2 ;

ARCHITECTURE Behavior OF contabits2 IS

TYPE Tipo_estado IS (A, B) ;

SIGNAL y : Tipo_estado;

BEGIN

PROCESS (Reset, Clock)

BEGIN

IF Reset = '1' 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 ;

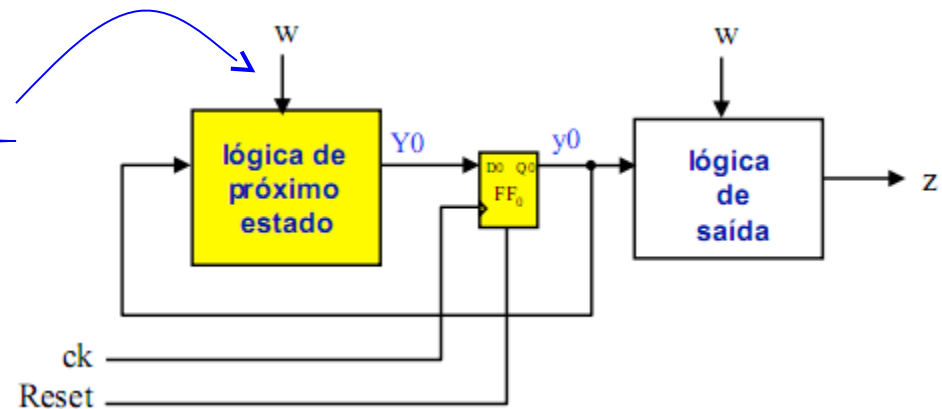
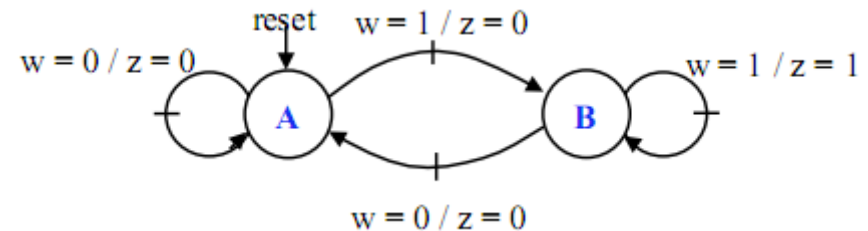
END IF ;

END CASE ;

END IF ;

END PROCESS ;

FSM descrita segundo o Modelo de Mealy (2 processos)



Máquinas Sequenciais

FSM descrita segundo o Modelo de Mealy (2 processos)

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

A saída “z” precisa estar em um processo diferente porque suas mudanças de valor não estão sujeitas ao relógio (pois trata-se de uma FSM de Mealy)

