



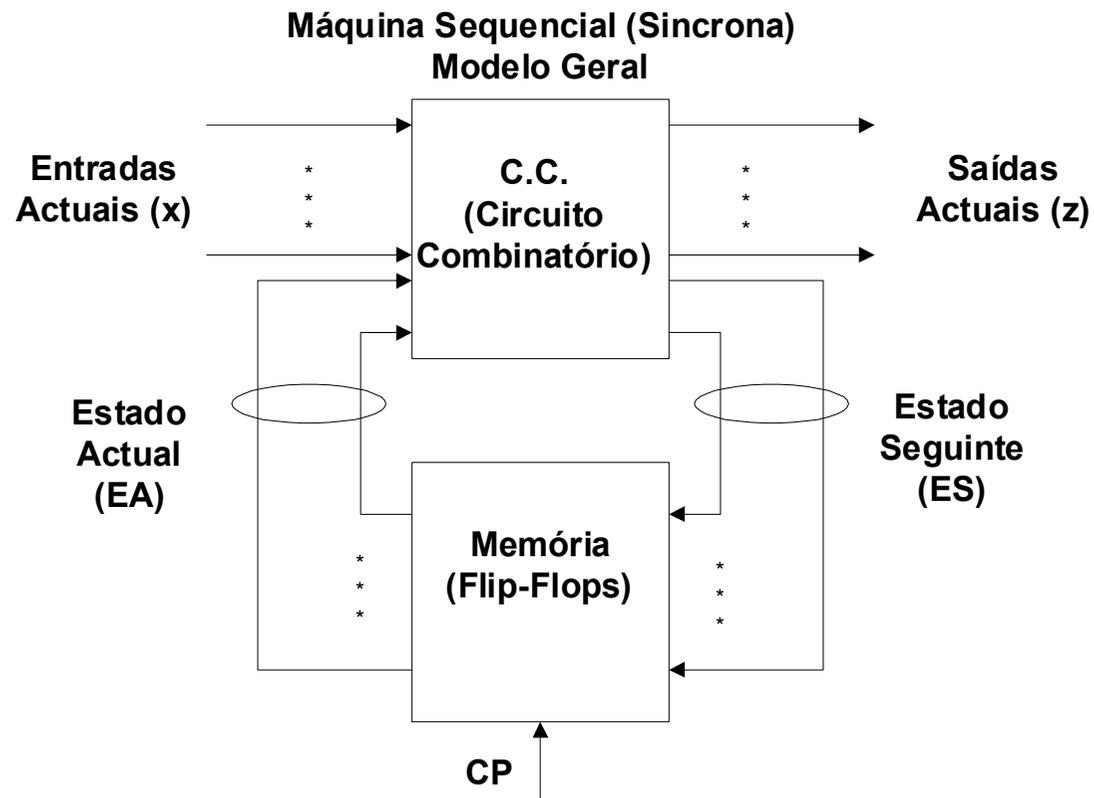
# CIRCUITOS SEQUENCIAIS SÍNCRONOS

- **SUMÁRIO:**
  - CIRCUITOS DE MOORE E MEALY
    - CARACTERIZAÇÃO
    - ESPECIFICAÇÃO
    - SÍNTESE
    - EXEMPLOS
  - PROJECTOS ALTERNATIVOS
    - 1 FLIP-FLOP / ESTADO
    - UTILIZAÇÃO DE CONTADORES

## CIRCUITOS SEQUENCIAIS SÍNCRONOS - 3

### ● CIRCUITOS DE MOORE E MEALY - CARACTERIZAÇÃO

#### MODELO GERAL

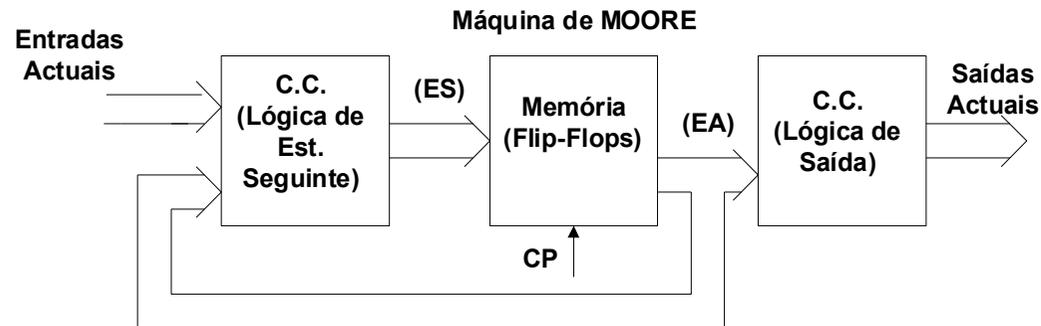


## CIRCUITOS SEQUENCIAIS SÍNCRONOS - 4

### ● CIRCUITOS DE MOORE E MEALY - CARACTERIZAÇÃO

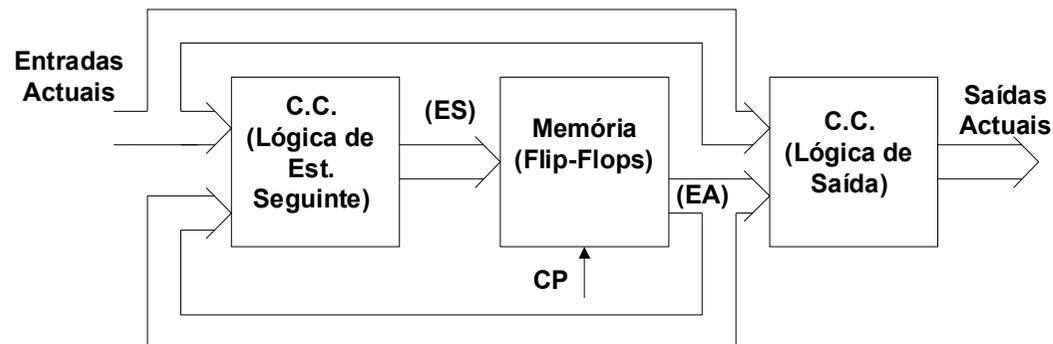
#### Modelo de Moore

As saídas são função directa do estado.



#### Modelo de Mealy

As saídas são função do estado e das entradas.



**Nota:** Em geral, os circuitos de Moore são mais simples (na geração das saídas), mas os circuitos de Mealy têm um número de estados menor ou igual, o que permite reduzir, nalguns casos, o número de FFs necessários.

## CIRCUITOS SEQUENCIAIS SÍNCRONOS - 5

### ● MÁQUINA DE MOORE - ESPECIFICAÇÃO

**Exemplo:** Projecto de um detector de sequência.

O projecto a realizar tem como objectivo identificar uma **sequência de três '1's na entrada X**.

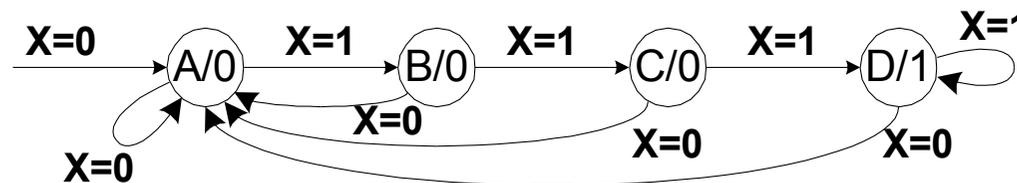
#### Diagrama de estados

Cada **estado** é identificado através de um círculo com uma referência única e (habitualmente) com os valores que as saídas têm nesse estado.

Ex: no estado A a saída toma o valor 0; no estado D a saída toma o valor 1.

Cada **transição** entre estados é descrita através de um vector ao qual está associado o valor das entradas que conduzem a essa transição.

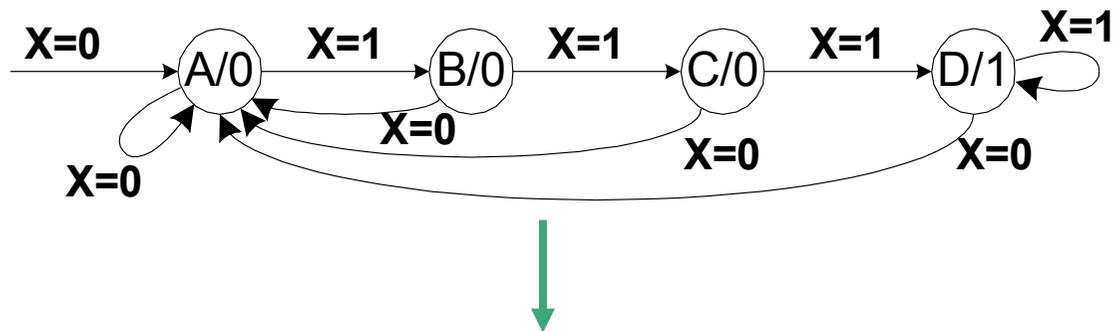
Ex: se estiver no estado A e a entrada for 1, a máquina irá passar para o estado B; se estiver no estado A e a entrada for 0, a máquina manter-se-á no estado A;



## CIRCUITOS SEQUENCIAIS SÍNCRONOS - 6

### ● MÁQUINA DE MOORE - SÍNTESE

A **tabela de transição de estados** explicita as saídas para cada estado (Moore) e os estados seguintes em função das várias entradas.



Estado Presente	Saída (Presente) Z	Estado Seguinte	
		X=0	X=1
A	0	A	B
B	0	A	C
C	0	A	D
D	1	A	D

### ● CODIFICAÇÃO DE ESTADOS

A **codificação de estados** substitui a representação simbólica dos estados por códigos binários.

Para representar  $M$  estados o código tem que ter pelo menos  $N$  bits, tal que  $M \leq 2^N$ .

Em geral, usa-se o número mínimo de bits necessário (número mínimo de flip-flops).

No entanto, veremos adiante que a utilização de um número de bits de estado superior ao mínimo pode ser vantajosa nalguns casos.

Por facilidade, as codificações mais utilizadas são as que correspondem à ordenação segundo o código binário ou segundo o código de Gray. No entanto, qualquer codificação que atribua códigos diferentes a estados diferentes conduz a implementações correctas.

A selecção do código mais apropriado para uma dada implementação (p.ex. o código que conduz ao circuito mais simples) não é trivial. Existem algoritmos mais ou menos complexos para tentar obter a **melhor** codificação, mas o seu estudo está fora do âmbito desta cadeira (uma heurística simples é, por exemplo, escolher o código de modo a minimizar as mudanças de bits em estados consecutivos da máquina).

## CIRCUITOS SEQUENCIAIS SÍNCRONOS - 8

### ● MÁQUINA DE MOORE - SÍNTESE

O processo de síntese inicia-se com a atribuição de uma codificação aos estados e com a selecção dos FFs a utilizar (conforme descrito na síntese de contadores).

**Tabela de  
Codificação de Estados**

Estado	$Q_1Q_0$
A	00
B	01
C	10
D	11



Estado Presente	Saída Presente Z	Estado Seguinte	
		X=0	X=1
A	0	A	B
B	0	A	C
C	0	A	D
D	1	A	D

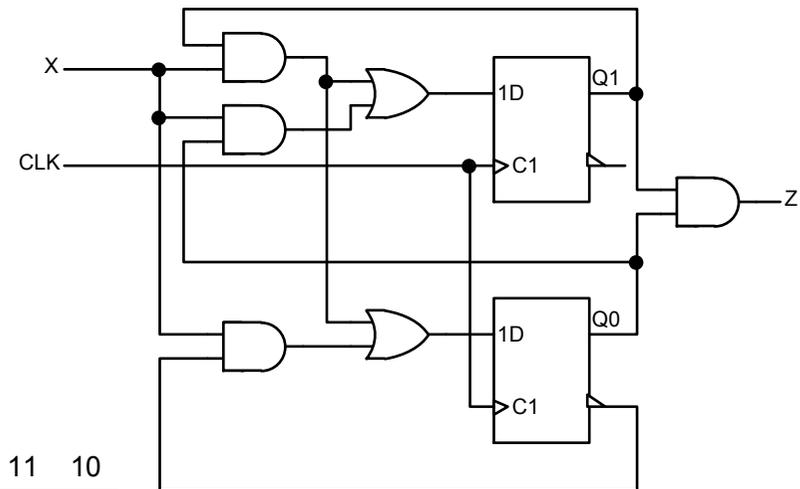
Estado Presente $Q_1Q_0$ em (n)	Saída Presente Z	Estado Seguinte $Q_1Q_0$ em (n+1)	
		X=0	X=1
00	0	00	01
01	0	00	10
10	0	00	11
11	1	00	11

## CIRCUITOS SEQUENCIAIS SÍNCRONOS - 9

### ● MÁQUINA DE MOORE – SÍNTESE COM FF D

O processo de síntese prossegue com a determinação, p.ex. através de mapas de Karnaugh, das entradas dos FFs como função das entradas do circuito e do estado anterior.

Estado Presente $Q_1Q_0$ em (n)	Saída Presente Z	Estado Seguinte $Q_1Q_0$ em (n+1)	
		X=0	X=1
0 0	0	0 0	0 1
0 1	0	0 0	1 0
1 0	0	0 0	1 1
1 1	1	0 0	1 1



$Q_1 \backslash Q_0$	0	1
0	0	0
1	0	1

$$Z = Q_1 \cdot Q_0$$

$Q_1 \backslash Q_0$	00	01	11	10
X=0	0	0	0	0
X=1	0	1	1	1

$$D_1 = X \cdot Q_1 + X \cdot Q_0$$

$Q_1 \backslash Q_0$	00	01	11	10
X=0	0	0	0	0
X=1	1	0	1	1

$$D_0 = X \cdot Q_1 + X \cdot \bar{Q}_0$$

## CIRCUITOS SEQUENCIAIS SÍNCRONOS - 10

### ● MÁQUINA DE MOORE – FF D – CODIFICAÇÃO ALTERNATIVA

O que muda se a codificação de estados for diferente?

Codificação de Estados

A	0	0
B	0	1
C	1	1
D	1	0

	$Q_0$	0	1
$Q_1$	0	0	0
1	1	0	0

$$Z = Q_1 \cdot \bar{Q}_0$$

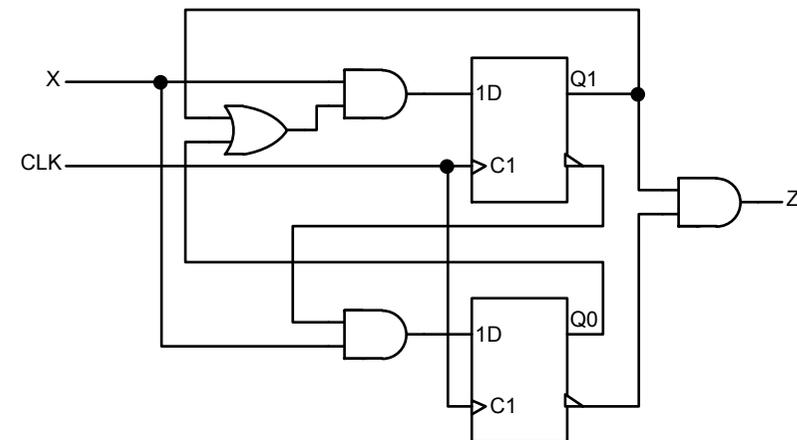
	$Q_1, Q_0$	00	01	11	10
X	0	0	0	0	0
1	0	1	1	1	0

$$D_1 = X \cdot Q_1 + X \cdot Q_0$$

$$= X \cdot (Q_1 + Q_0)$$

	$Q_1, Q_0$	00	01	11	10
X	0	0	0	0	0
1	1	1	0	0	0

$$D_0 = X \cdot \bar{Q}_1$$



# CIRCUITOS SEQUENCIAIS SÍNCRONOS - 11

## ● MÁQUINA DE MOORE – SÍNTESE COM FF JK

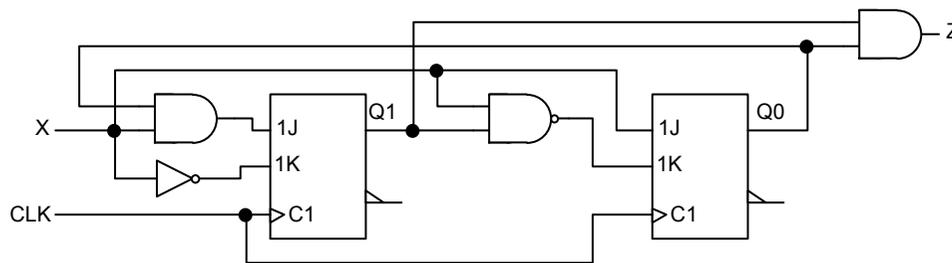
Estado Presente	Saída Presente	Estado Seguinte $Q_1Q_0$ em (n+1)			
Estado Presente $Q_1Q_0$ em (n)	Saída Presente Z	Valores a colocar nas entradas dos FF			
		X=0		X=1	
		J1K1	J0K0	J1K1	J0K0
00	0	0X 0X	0X 1X		
01	0	0X X1	1X X1		
10	0	X1 0X	X0 1X		
11	1	X1 X1	X0 X0		

Estado Presente $Q_1Q_0$ em (n)	Saída Presente Z	Valores a colocar nas entradas dos FF			
		X=0		X=1	
		J1K1	J0K0	J1K1	J0K0
00	0	0X 0X	0X 1X		
01	0	0X X1	1X X1		
10	0	X1 0X	X0 1X		
11	1	X1 X1	X0 X0		

Estado Presente $Q_1Q_0$ em (n)	Saída Presente Z	Valores a colocar nas entradas dos FF			
		X=0		X=1	
		J1K1	J0K0	J1K1	J0K0
00	0	0X 0X	0X 1X		
01	0	0X X1	1X X1		
10	0	X1 0X	X0 1X		
11	1	X1 X1	X0 X0		



$$J_1 = XQ_0$$

$$J_0 = X$$

$$Z = Q_1Q_0$$

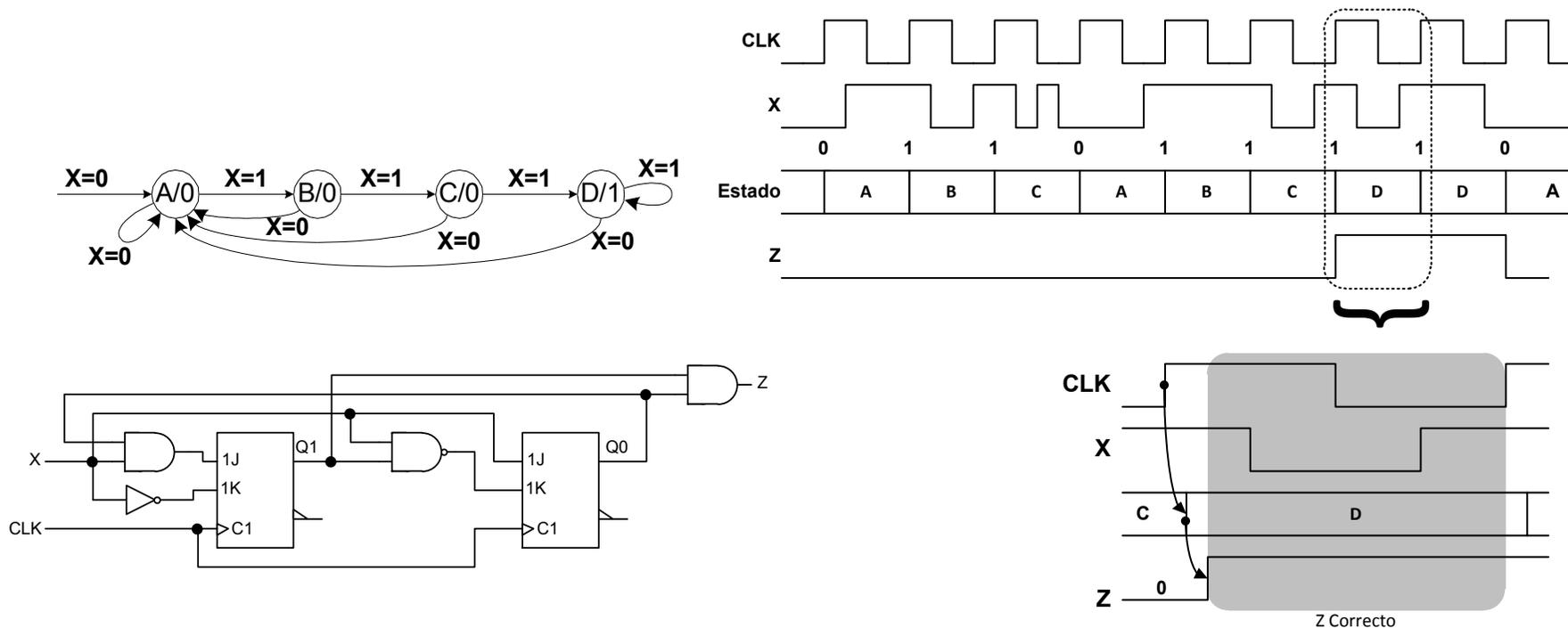
$$K_1 = \bar{X}$$

$$K_0 = \bar{X} + \bar{Q}_1 = \overline{XQ_1}$$

## CIRCUITOS SEQUENCIAIS SÍNCRONOS - 12

### ● MÁQUINA DE MOORE – DIAGRAMA TEMPORAL

Na máquina de Moore a saída depende apenas do estado: a saída varia na sequência da mudança de estado, portanto muda na sequência da transição de relógio e mantém-se constante durante o resto do ciclo de relógio. Exemplo:

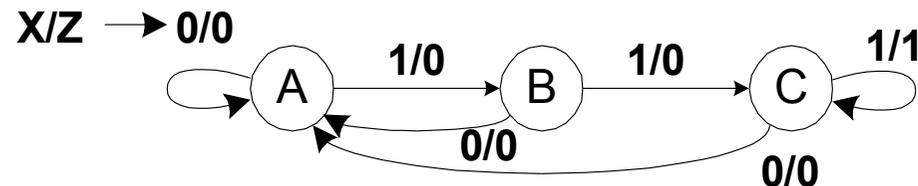


### ● MÁQUINA DE MEALY - ESPECIFICAÇÃO

Na máquina de Mealy a saída depende do estado **e das entradas**.

O diagrama de estados **reflete**, portanto, a influência das entradas nas **saídas**. Esta influência é habitualmente representada nos **vectores correspondentes às transições entre estados**.

Exemplo: O mesmo projecto de um detector de sequência de três '1's na entrada X.

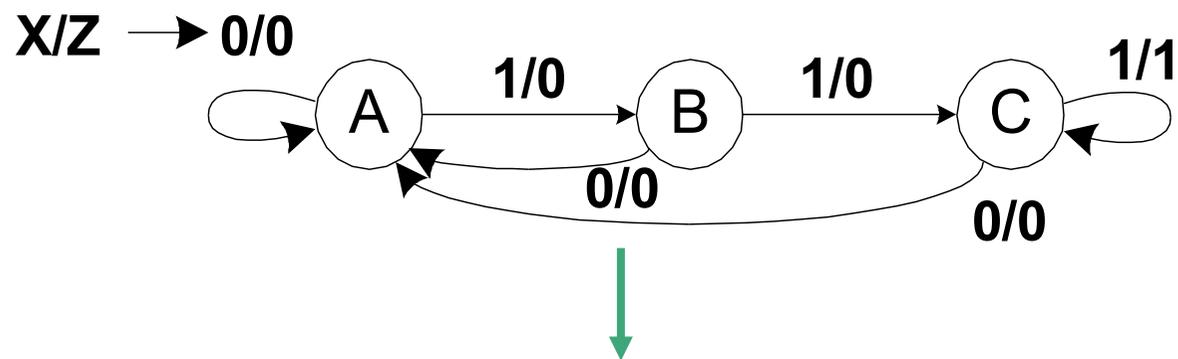


Neste caso particular o comportamento do circuito pode ser modelado com um **número de estados inferior** ao modelo de Moore. Contudo, não há redução do número de FFs, apenas a existência de indiferenças suplementares nos mapas de Karnaugh que podem conduzir a uma maior simplificação da lógica associada à geração das entradas. Em contrapartida, a geração da saída será mais complexa.

## CIRCUITOS SEQUENCIAIS SÍNCRONOS - 14

### ● MÁQUINA DE MEALY – SÍNTESE

A estrutura da **tabela de transição de estados** é alterada de modo a reflectir a diferente opção na geração das saídas.



Estado Presente	Estado Seguinte e Saída Presente (Z)	
	X=0	X=1
A	A/0	B/0
B	A/0	C/0
C	A/0	C/1

## CIRCUITOS SEQUENCIAIS SÍNCRONOS - 15

### ● MÁQUINA DE MEALY – SÍNTESE

**Tabela de  
Codificação de Estados**

Estado	$Q_1Q_0$
A	00
B	01
C	10

Estado Presente	Estado Seguinte e Saída Presente (Z)	
	X=0	X=1
A	A/0	B/0
B	A/0	C/0
C	A/0	C/1

Estado Presente $Q_1Q_0$	Estado Seguinte e Saída Presente (Z)	
	X=0	X=1
00	00/0	01/0
01	00/0	10/0
10	00/0	10/1

## CIRCUITOS SEQUENCIAIS SÍNCRONOS - 16

### ● MÁQUINA DE MEALY – SÍNTESE COM FF D – codificação 1

Estado Presente $Q_1Q_0$	Estado Seguinte e Saída Presente (Z)	
	X=0	X=1
00	00/0	01/0
01	00/0	10/0
10	00/0	10/1

$Q_1Q_0$	X			
X	00	01	11	10
0	0	0	X	0
1	0	1	X	1

$Q_1Q_0$	X			
X	00	01	11	10
0	0	0	X	0
1	1	0	X	0

$Q_1Q_0$	X			
X	00	01	11	10
0	0	0	X	0
1	0	0	X	1

$$D_1 = X \cdot Q_1 + X \cdot Q_0$$

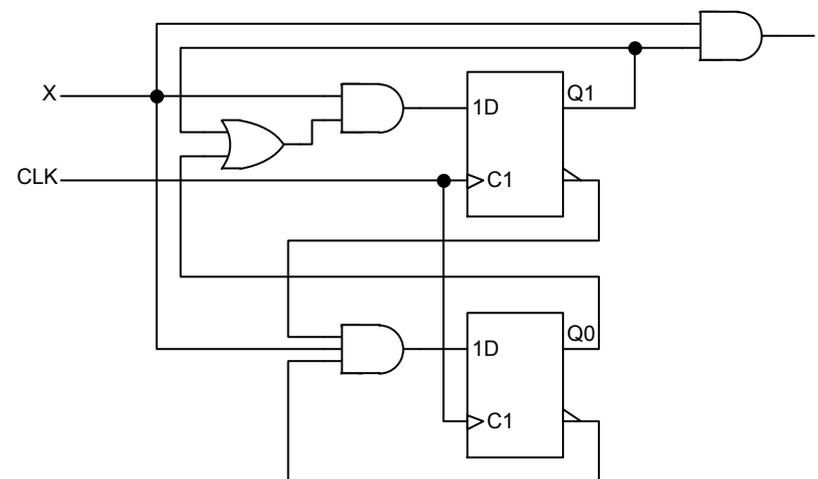
$$= X \cdot (Q_1 + Q_0)$$

$$D_0 = X \cdot \bar{Q}_1 \cdot \bar{Q}_0$$

$$Z = X \cdot Q_1$$

#### Codificação de Estados

A	0	0
B	0	1
C	1	0



## CIRCUITOS SEQUENCIAIS SÍNCRONOS - 17

### ● MÁQUINA DE MEALY – SÍNTESE COM FF D – codificação 2

Estado Presente $Q_1Q_0$	Estado Seguinte e Saída Presente (Z)	
	X=0	X=1
00	00/0	01/0
01	00/0	11/0
11	00/0	11/1

$Q_1Q_0$	00	01	11	10
X	0	0	0	X
0	0	0	0	X
1	0	1	1	X

$$D_1 = X \cdot Q_0$$

$Q_1Q_0$	00	01	11	10
X	0	0	0	X
0	0	0	0	X
1	1	1	1	X

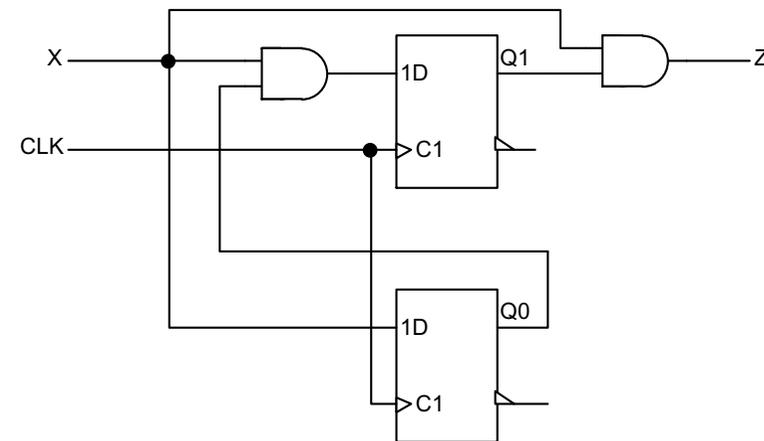
$$D_0 = X$$

$Q_1Q_0$	00	01	11	10
X	0	0	0	X
0	0	0	0	X
1	0	0	1	X

$$Z = X \cdot Q_1$$

#### Codificação de Estados

A	0	0
B	0	1
C	1	1



## CIRCUITOS SEQUENCIAIS SÍNCRONOS - 18

### ● MÁQUINA DE MEALY – SÍNTESE COM FF JK

Estado Presente $Q_1Q_0$	Estado Seguinte e Saída Presente (Z)	
	X=0	X=1
00	00/0	01/0
01	00/0	10/0
10	00/0	10/1

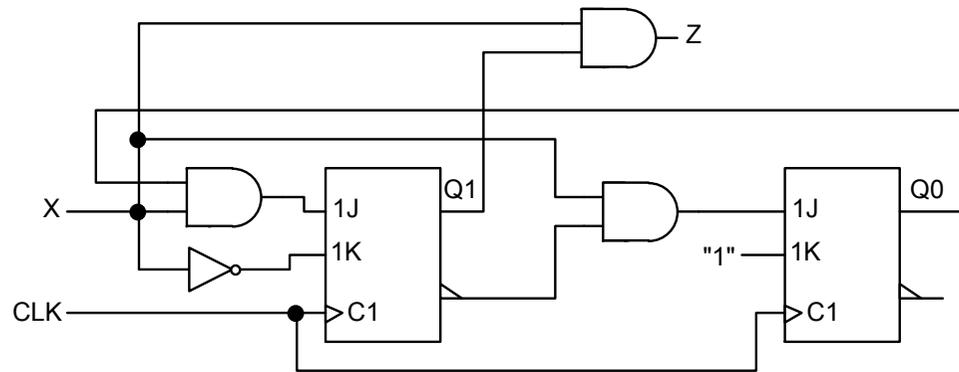
Q-Transição		JK
de	para	
0	0	0X
0	1	1X
1	0	X1
1	1	X0

Estado Presente $Q_1Q_0$	Valores a colocar nas entradas dos FF / Saída Presente (Z)	
	X=0	X=1
	J1K1 J0K0 /Z	J1K1 J0K0 /Z
00	0X 0X /0	0X 1X /0
01	0X X1 /0	1X X1 /0
10	X1 0X /0	X0 0X /1

# CIRCUITOS SEQUENCIAIS SÍNCRONOS - 19

## ● MÁQUINA DE MEALY – SÍNTESE COM FF JK

Estado Presente $Q_1Q_0$	Valores a colocar nas entradas dos FF / Saída Presente (Z)	
	X=0	X=1
	J1K1 J0K0 / Z	J1K1 J0K0 / Z
00	0X 0X / 0	0X 1X / 0
01	0X X1 / 0	1X X1 / 0
10	X1 0X / 0	X0 0X / 1



$$J_1 = XQ_0$$

$$J_0 = X\bar{Q}_1$$

$$K_1 = \bar{X}$$

$$K_0 = 1$$

$$Z = XQ_1$$

$Q_1Q_0$	00	01	11	10
x				
0	0	0	x	x
1	0	1	x	x

**J<sub>1</sub>**

$Q_1Q_0$	00	01	11	10
x				
0	x	x	x	1
1	x	x	x	0

**K<sub>1</sub>**

$Q_1Q_0$	00	01	11	10
x				
0	0	x	x	0
1	1	x	x	0

**J<sub>0</sub>**

$Q_1Q_0$	00	01	11	10
x				
0	x	1	x	x
1	x	1	x	x

**K<sub>0</sub>**

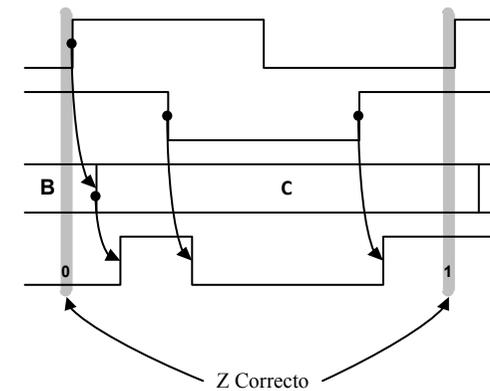
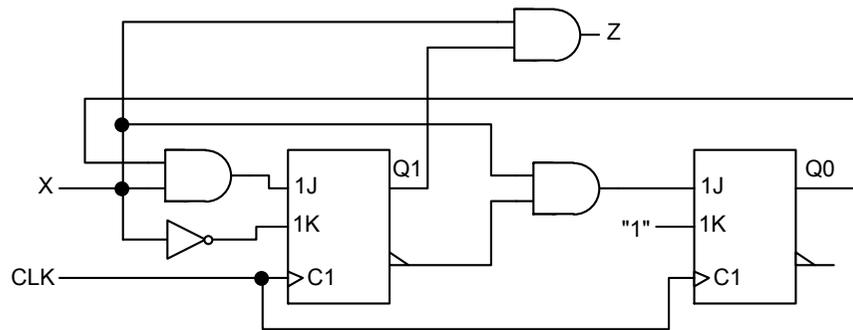
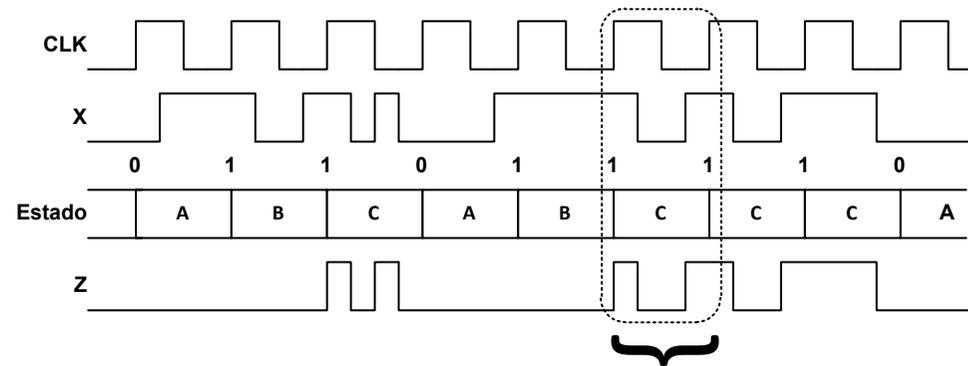
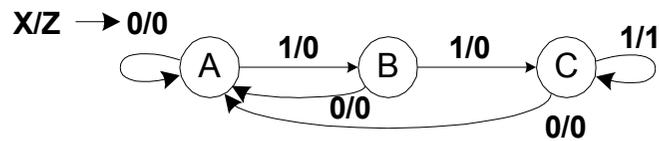
$Q_1Q_0$	00	01	11	10
x				
0	0	0	x	0
1	0	0	x	1

**Z**

## CIRCUITOS SEQUENCIAIS SÍNCRONOS - 20

### ● MÁQUINA DE MEALY – DIAGRAMA TEMPORAL

Na máquina de Mealy a saída pode ser diretamente afectada por uma variação na entrada, mesmo que não haja mudança de estado. A saída tenta “antecipar” o valor que só é, de facto, correto imediatamente antes da transição de relógio.



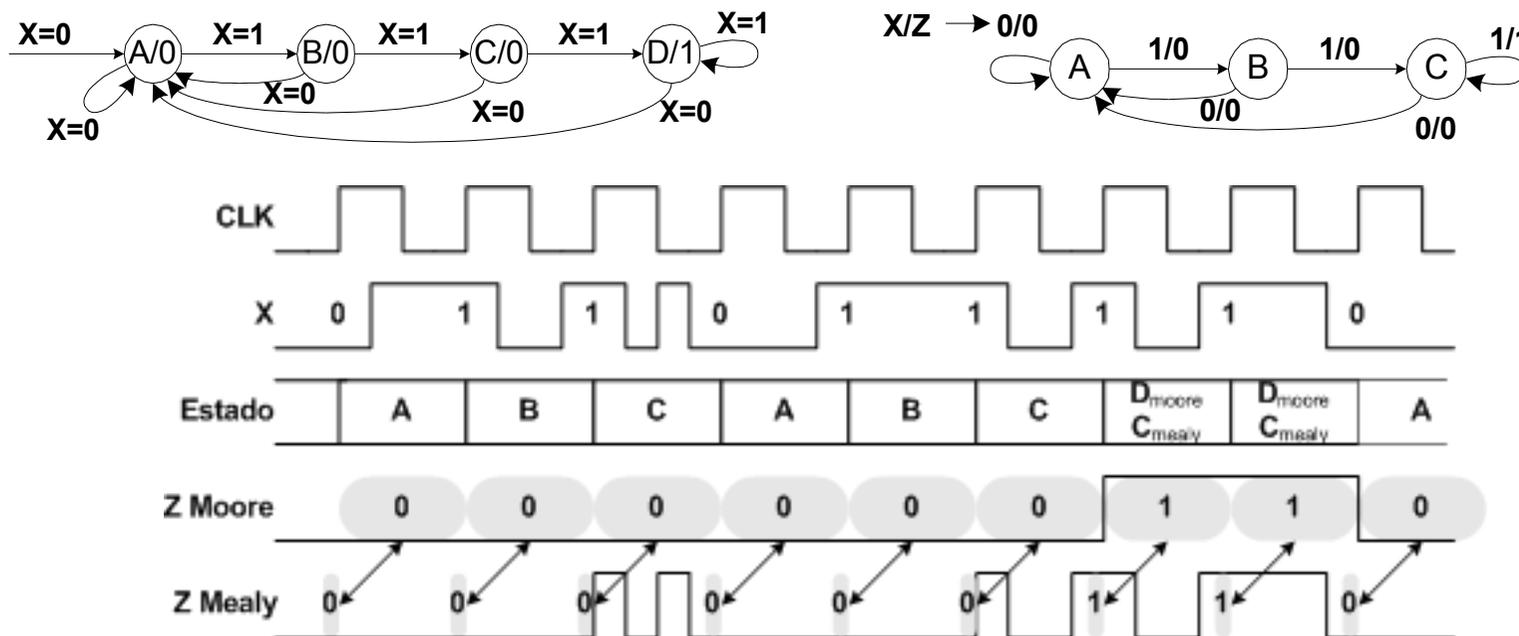
## CIRCUITOS SEQUENCIAIS SÍNCRONOS - 21

### ● MÁQUINA DE MOORE VS. MÁQUINA DE MEALY

As saídas em ambas as máquinas são equivalentes, mas não idênticas.

A máquina de Mealy disponibiliza as saídas *imediatamente antes* do flanco de relógio; estas só são garantidamente válidas exatamente nesse instante.

Em contrapartida, na máquina de Moore as saídas são válidas durante todo o ciclo de relógio *após* o flanco de relógio (menos o tempo inicial de comutação dos FF/portas).

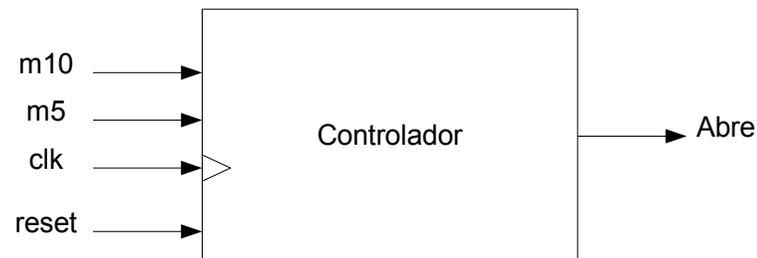


## CIRCUITOS SEQUENCIAIS SÍNCRONOS - 22

### ● EXEMPLO – Máquina de Distribuição de Pastilhas (MDP)

Projecto de um controlador de uma máquina de distribuição de pastilhas:

- A máquina entrega 1 pastilha quando recebe 15 cêntimos em moedas;
- Recebe 1 moeda de cada vez;
- Aceita moedas de 5 e de 10 cêntimos;
- Um sensor mecânico indica se foi recebida 1 moeda de 5 cêntimos ou se foi recebida uma moeda de 10 cêntimos;
- O circuito tem uma única saída que, quando ativa, abre a gaveta e entrega a pastilha ao cliente;
- Não é entregue troco se forem recebidas moedas a mais.

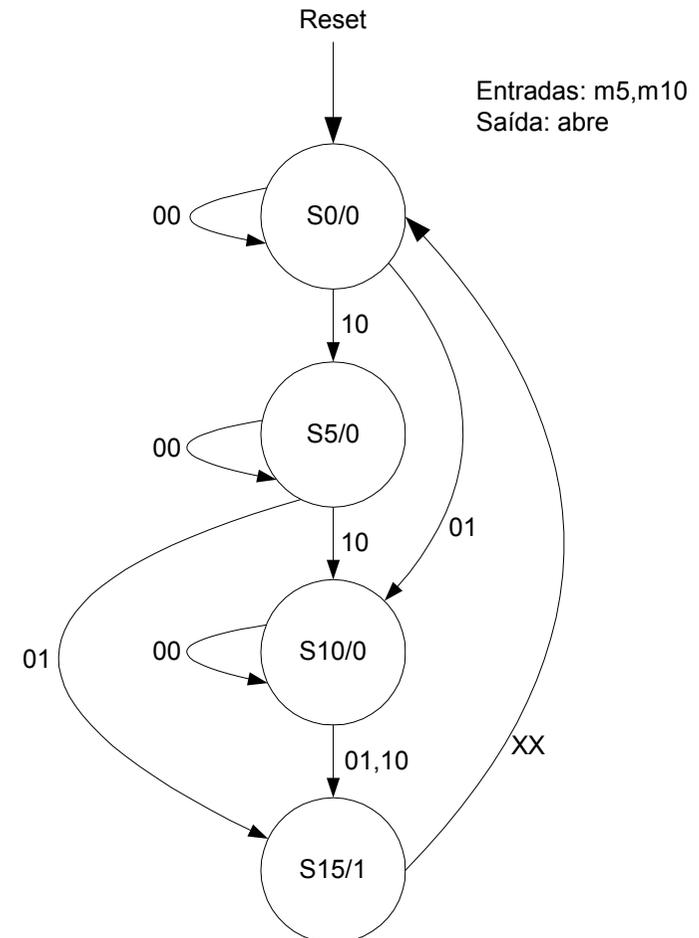


## CIRCUITOS SEQUENCIAIS SÍNCRONOS - 23

### ● MDP – DIAGRAMA DE ESTADOS

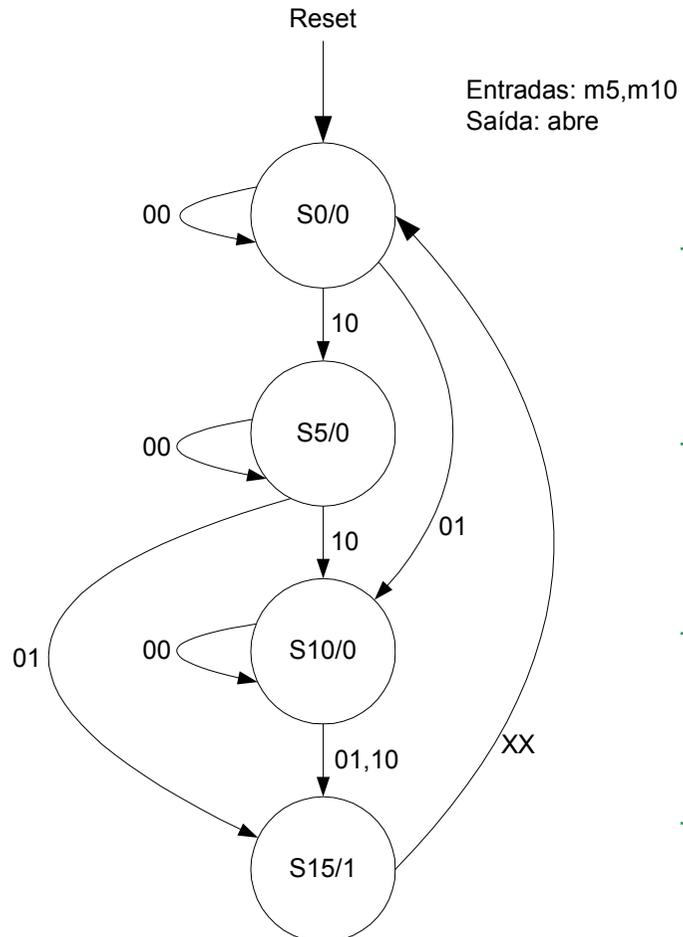
O controlador pode ser realizado com uma máquina de 4 estados:

- S0 – estado inicial, não foi introduzida moeda.
- S5 – foram introduzidos 5 cêntimos.
- S10 – foram introduzidos 10 cêntimos (é indiferente se 1 moeda de 10c ou 2 de 5c).
- S15 – foram introduzidos 15 cêntimos (ou mais).
- A saída é activada apenas quando é atingido o estado S15.



## CIRCUITOS SEQUENCIAIS SÍNCRONOS - 24

### ● MDP – TABELA DE TRANSIÇÃO DE ESTADOS



Estado Actual	Entradas		Estado Seguinte (D1 <sub>n+1</sub> , D0 <sub>n+1</sub> )			Saída		
	Q1	Q0	m5	m10	Q1	Q0	ABRE	
S0C	0	0	0	0	S0C	0	0	0
			0	1	S10C	1	0	0
			1	0	S5C	0	1	0
			1	1	X	X	X	0
S5C	0	1	0	0	S5C	0	1	0
			0	1	S15C	1	1	0
			1	0	S10C	1	0	0
			1	1	X	X	X	0
S10C	1	0	0	0	S10C	1	0	0
			0	1	S15C	1	1	0
			1	0	S15C	1	1	0
			1	1	X	X	X	0
S15C	1	1	0	0	S0C	0	0	1
			0	1	S0C	0	0	1
			1	0	S0C	0	0	1
			1	1	X	X	X	1

## CIRCUITOS SEQUENCIAIS SÍNCRONOS - 25

### ● MDP – PROJECTO DA LÓGICA COMBINATÓRIA

		m5 m10			
		00	01	11	10
Q <sub>1</sub> Q <sub>0</sub>	00	0	1	X	0
	01	0	1	X	1
	11	0	0	X	0
	10	1	1	X	1

**K-mapa para D1**

$$D_1 = \overline{Q_1} m10 + \overline{Q_1} Q_0 m5 + Q_1 \overline{Q_0}$$

		Q <sub>0</sub>	
		0	1
Q <sub>1</sub>	0	0	0
	1	0	1

**K-mapa para Saída**

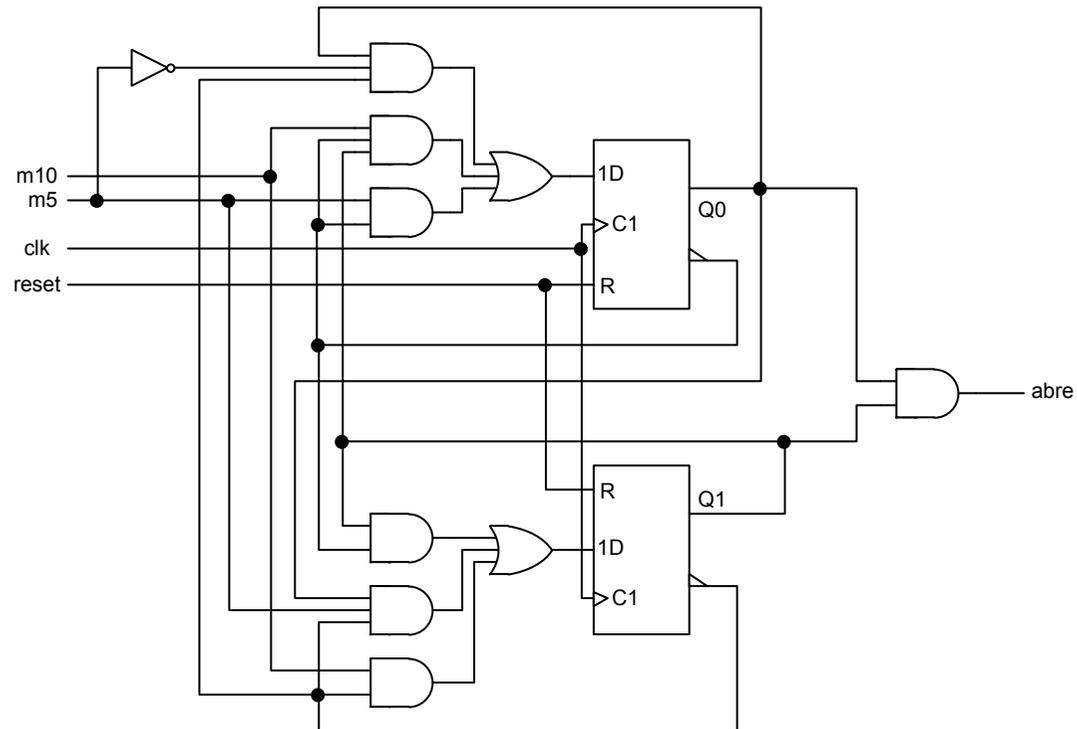
$$Abre = Q_1 Q_0$$

		m5 m10			
		00	01	11	10
Q <sub>1</sub> Q <sub>0</sub>	00	0	0	X	1
	01	1	1	X	0
	11	0	0	X	0
	10	0	1	X	1

**K-mapa para D0**

$$D_0 = \overline{Q_0} m5 + \overline{Q_1} Q_0 \overline{m5} + Q_1 \overline{Q_0} m10$$

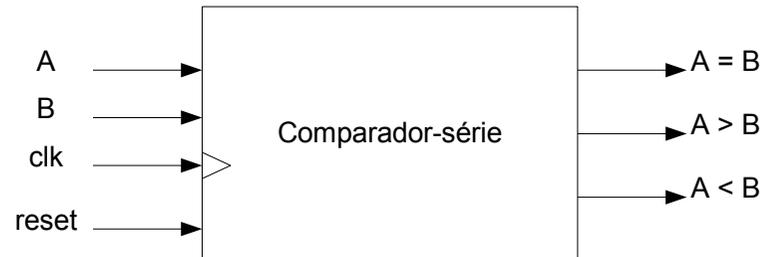
### ● MDP – LOGIGRAMA DO CIRCUITO



### ● EXEMPLO – Comparador Série (CMP)

Pretende-se projectar um comparador série, que compara 2 números binários A e B.

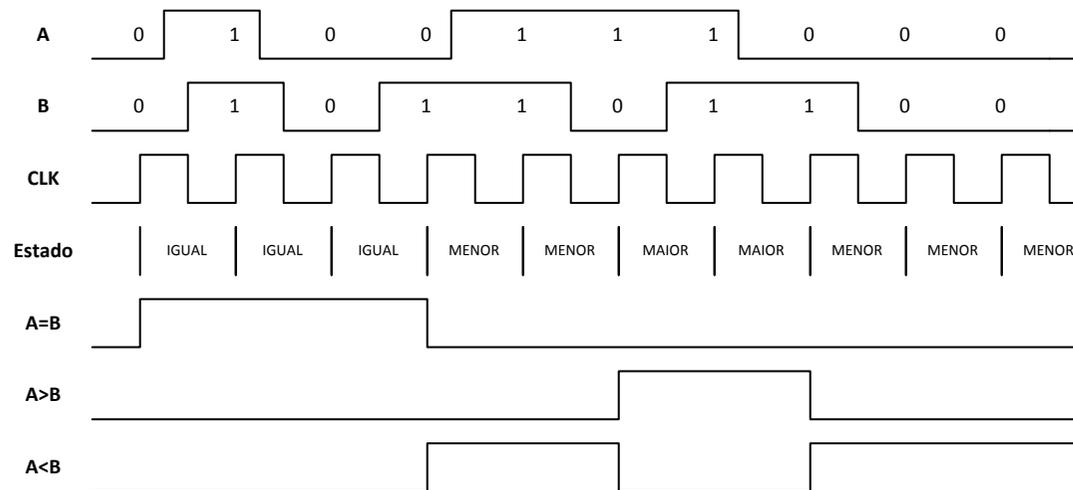
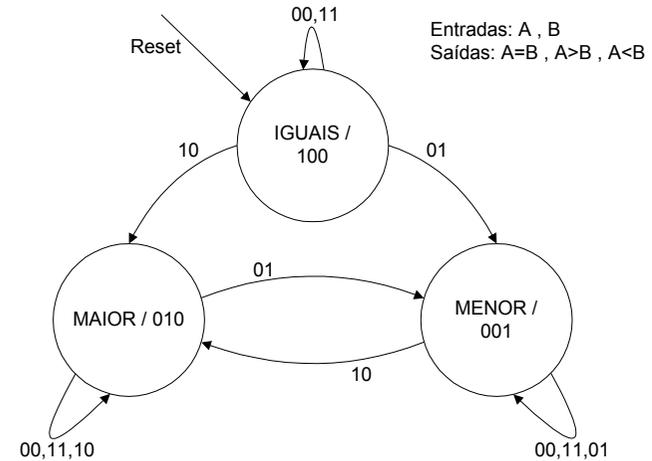
Os números são recebidos em série (bit a bit), com o bit de menor peso a ser recebido primeiro.



## CIRCUITOS SEQUENCIAIS SÍNCRONOS - 28

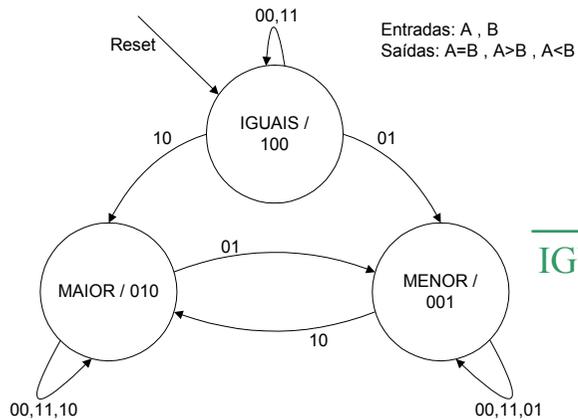
### ● CMP1 – DIAGRAMA DE ESTADOS

O controlador pode ser realizado como uma máquina de Moore de 3 estados.



## CIRCUITOS SEQUENCIAIS SÍNCRONOS - 29

### ● CMP1 – TABELA DE TRANSIÇÃO DE ESTADOS



	Estado Actual		Entradas		Estado Seguinte		Saídas (Estado Actual)			
	Q1	Q0	A	B	Q1	Q0	A=B	A>B	A<B	
IGUAIS	0	0	0	0	IGUAIS	0	0	1	0	0
			0	1	MENOR	0	1	1	0	0
			1	0	MAIOR	1	0	1	0	0
			1	1	IGUAIS	0	0	1	0	0
MENOR	0	1	0	0	MENOR	0	1	0	0	1
			0	1	MENOR	0	1	0	0	1
			1	0	MAIOR	1	0	0	0	1
			1	1	MENOR	0	1	0	0	1
MAIOR	1	0	0	0	MAIOR	1	0	0	1	0
			0	1	MENOR	0	1	0	1	0
			1	0	MAIOR	1	0	0	1	0
			1	1	MAIOR	1	0	0	1	0

## CIRCUITOS SEQUENCIAIS SÍNCRONOS - 30

### ● CMP1 – PROJECTO DA LÓGICA COMBINATÓRIA

		A B			
		00	01	11	10
Q <sub>1</sub> Q <sub>0</sub>	00	0	0	0	1
	01	0	0	0	1
	11	X	X	X	X
	10	1	0	1	1

**K-mapa para D1**

$$D_1 = Q_1 \bar{B} + A \bar{B} + Q_1 A$$

		A B			
		00	01	11	10
Q <sub>1</sub> Q <sub>0</sub>	00	0	1	0	0
	01	1	1	1	0
	11	X	X	X	X
	10	0	1	0	0

**K-mapa para D0**

$$D_0 = \bar{A} B + Q_0 \bar{A} + Q_0 B$$

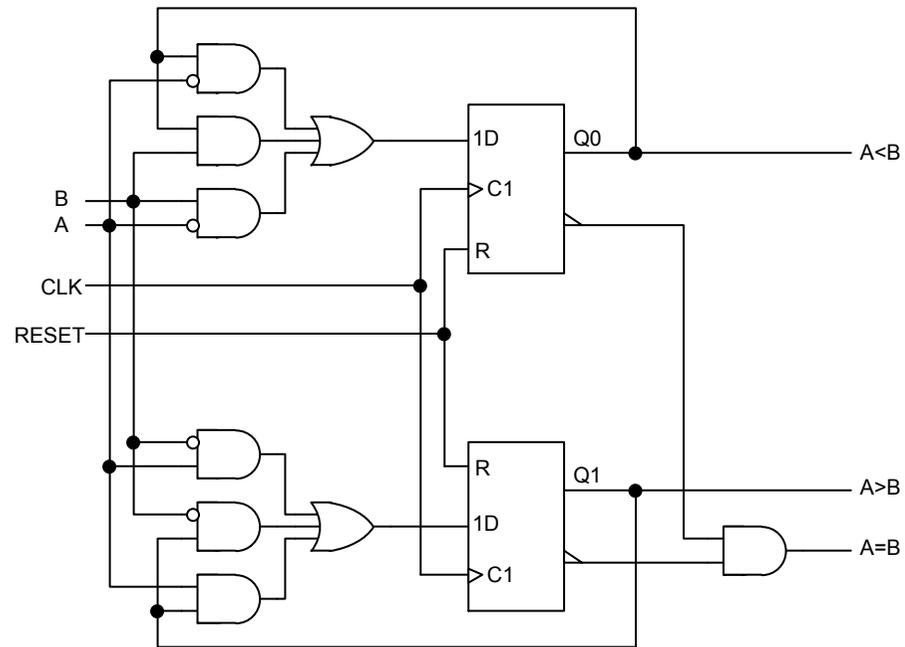
#### K-mapas para Saídas

		Q <sub>0</sub>		
		0	1	
Q <sub>1</sub>	0	1	0	(A = B) = $\bar{Q}_1 \bar{Q}_0$
	1	0	X	

		Q <sub>0</sub>		
		0	1	
Q <sub>1</sub>	0	0	0	(A > B) = Q <sub>1</sub>
	1	1	X	

		Q <sub>0</sub>		
		0	1	
Q <sub>1</sub>	0	0	1	(A < B) = Q <sub>0</sub>
	1	0	X	

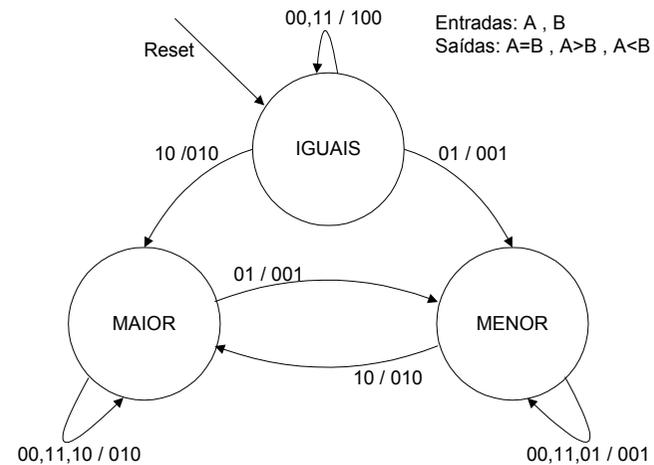
### ● CMP1 – LOGIGRAMA DO CIRCUITO



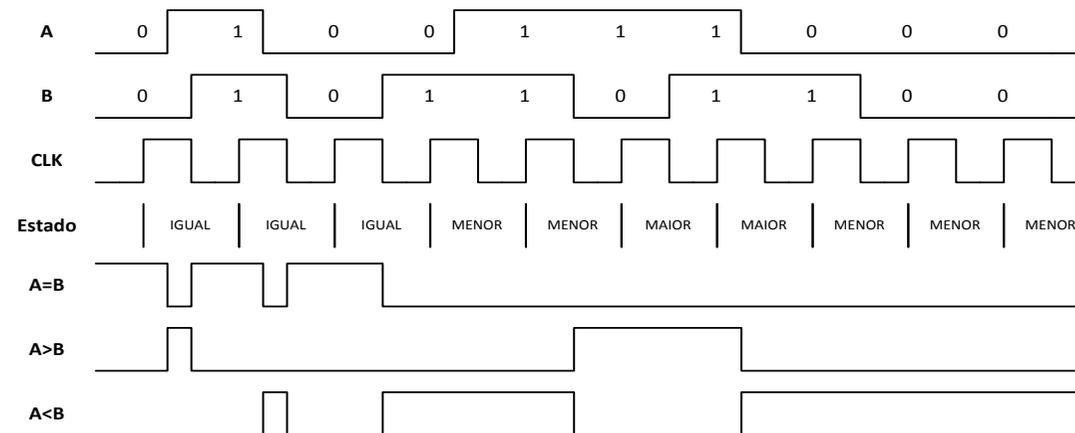
## CIRCUITOS SEQUENCIAIS SÍNCRONOS - 32

### ● CMP2 – DIAGRAMA DE ESTADOS

O controlador pode também ser realizado como uma máquina de Mealy de 3 estados.

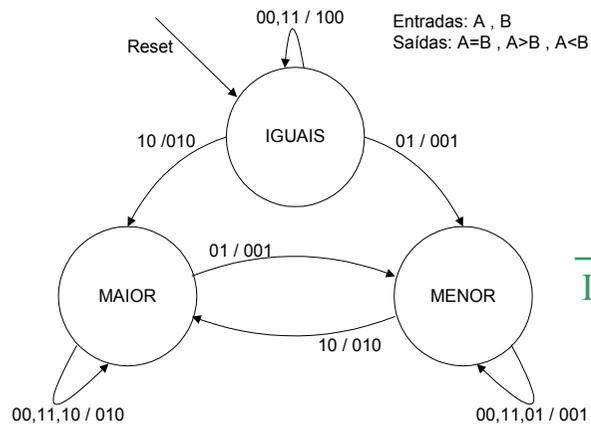


### ► Exemplo



## CIRCUITOS SEQUENCIAIS SÍNCRONOS - 33

### ● CMP2 – TABELA DE TRANSIÇÃO DE ESTADOS



Estado Actual	Entradas		Estado Seguinte		Saídas					
	Q1	Q0	A	B	Q1	Q0	A=B	A>B	A<B	
IGUAIS	0	0	0	0	IGUAIS	0	0	1	0	0
			0	1	MENOR	0	1	0	0	1
			1	0	MAIOR	1	0	0	1	0
			1	1	IGUAIS	0	0	1	0	0
MENOR	0	1	0	0	MENOR	0	1	0	0	1
			0	1	MENOR	0	1	0	0	1
			1	0	MAIOR	1	0	0	1	0
			1	1	MENOR	0	1	0	0	1
MAIOR	1	0	0	0	MAIOR	1	0	0	1	0
			0	1	MENOR	0	1	0	0	1
			1	0	MAIOR	1	0	0	1	0
			1	1	MAIOR	1	0	0	1	0

## ● CMP2 – PROJECTO DA LÓGICA COMBINATÓRIA

		A B			
		00	01	11	10
Q <sub>1</sub> Q <sub>0</sub>	00	0	0	0	1
	01	0	0	0	1
	11	X	X	X	X
	10	1	0	1	1

$$D_1 = Q_1 \bar{B} + A \bar{B} + Q_1 A$$

		A B			
		00	01	11	10
Q <sub>1</sub> Q <sub>0</sub>	00	0	1	0	0
	01	1	1	1	0
	11	X	X	X	X
	10	0	1	0	0

$$D_0 = \bar{A} B + Q_0 \bar{A} + Q_0 B$$

		A B			
		00	01	11	10
Q <sub>1</sub> Q <sub>0</sub>	00	1	0	1	0
	01	0	0	0	0
	11	X	X	X	X
	10	0	0	0	0

$$\begin{aligned} (A = B) &= \bar{Q}_1 \bar{Q}_0 \bar{A} \bar{B} + \bar{Q}_1 \bar{Q}_0 A B \\ &= \bar{Q}_1 \bar{Q}_0 (A \oplus B) \end{aligned}$$

		A B			
		00	01	11	10
Q <sub>1</sub> Q <sub>0</sub>	00	0	0	0	1
	01	0	0	0	1
	11	X	X	X	X
	10	1	0	1	1

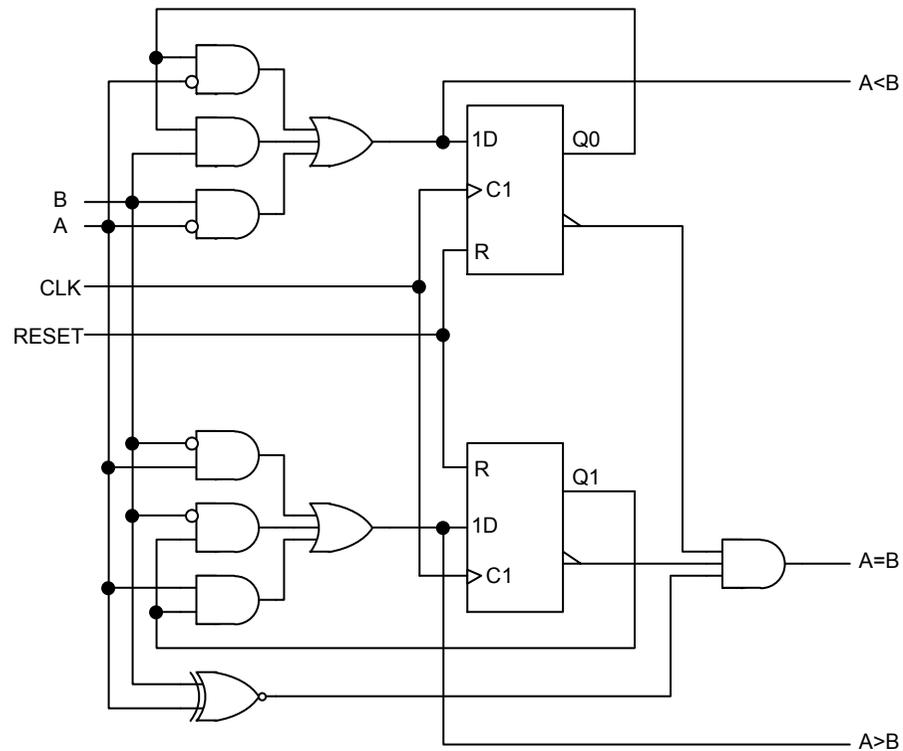
$$(A > B) = Q_1 \bar{B} + A \bar{B} + Q_1 A$$

		A B			
		00	01	11	10
Q <sub>1</sub> Q <sub>0</sub>	00	0	1	0	0
	01	1	1	1	0
	11	X	X	X	X
	10	0	1	0	0

$$(A < B) = \bar{A} B + Q_0 \bar{A} + Q_0 B$$

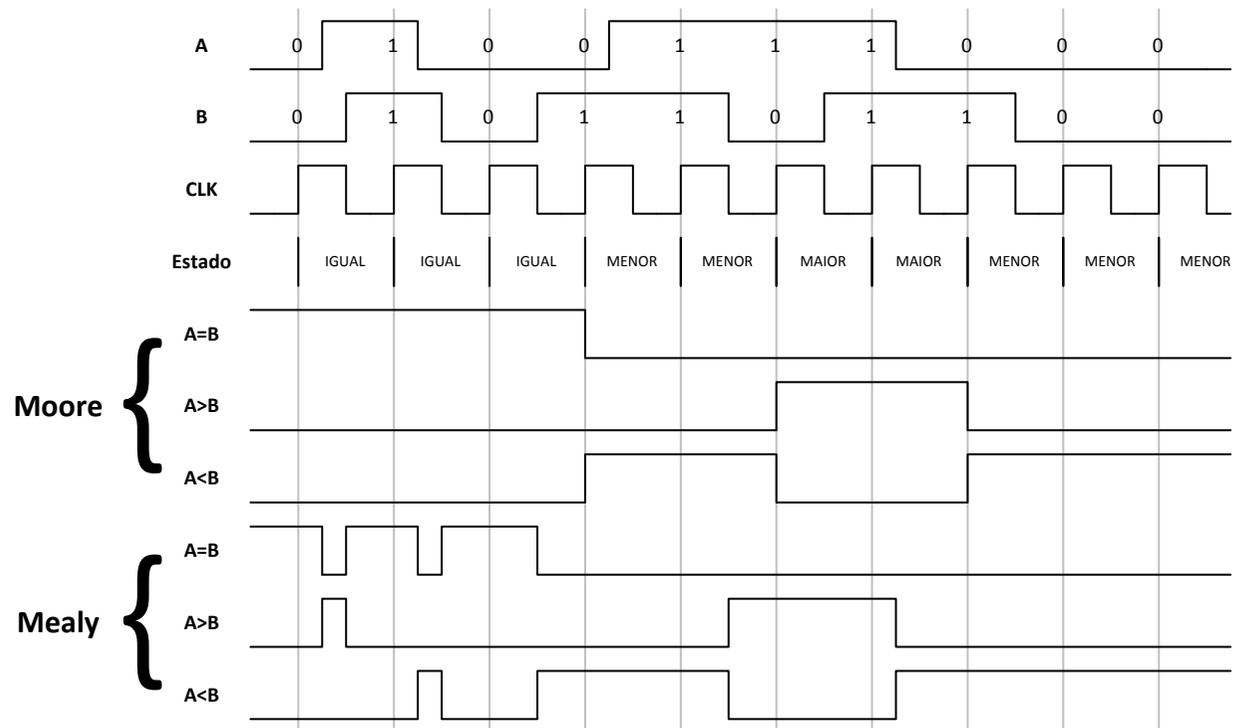
## CIRCUITOS SEQUENCIAIS SÍNCRONOS - 35

### ● CMP2 – LOGIGRAMA DO CIRCUITO



## CIRCUITOS SEQUENCIAIS SÍNCRONOS - 36

### ● CMP – FORMAS DE ONDA – Moore vs. Mealy



### ● PROJECTOS ALTERNATIVOS

A síntese de circuitos sequenciais síncronos foi abordada segundo o método clássico cujo objectivo é o de minimizar o número de flip-flops do circuito.

De seguida, referir-se-ão alguns métodos alternativos que, nalguns casos, podem ser mais vantajosos.

No capítulo seguinte será abordada ainda a implementação de máquinas de estados com recurso a memórias.

### ● PROJECTOS ALTERNATIVOS – 1 FLIP-FLOP POR ESTADO

“*One-hot encoding*” - metodologia de projecto alternativa que consiste em utilizar um número de flip-flops igual ao número de estados e codificar os estados de modo a existir um e um só FF activo por estado.

O projecto tem as seguintes características:

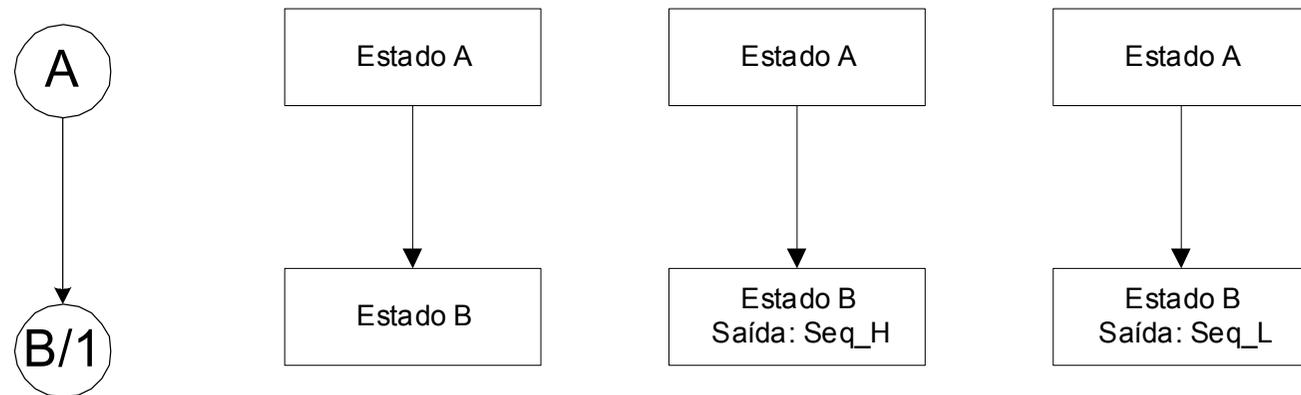
- Cada estado corresponde à activação de 1 e só um FF;
  - e.g., 0100000 - correcto; 01100000 - incorrecto.
- A inicialização do circuito corresponde à activação do FF correspondente ao estado inicial e à desactivação de todos os outros;
  - e.g., 10000000 (para oito estados).
- O projecto fica bastante simplificado porque é possível obter as equações e/ou o diagrama lógico do circuito directamente a partir da especificação da máquina. (No entanto, o projecto pode ser realizado também segundo o procedimento habitual)

### ● UM FLIP-FLOP / ESTADO

No caso de 1 FF por estado o diagrama lógico pode ser determinado **sem** recurso a tabelas de estado ou a mapas de Karnaugh.

A determinação do diagrama lógico é conseguida através da implementação directa de sub-estruturas do diagrama de estados ou de um fluxograma.

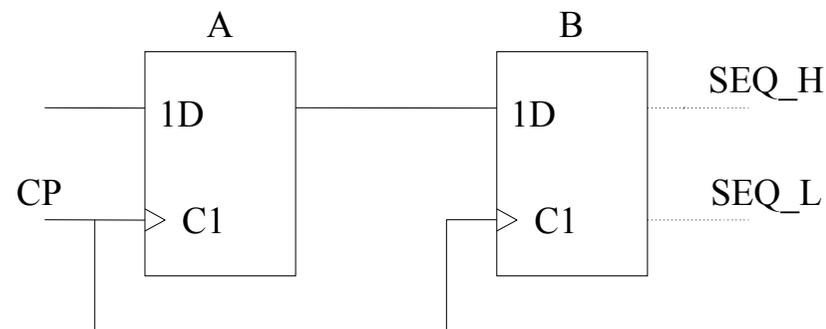
(1) Transição **Incondicional** entre Estados:



**Equações:**

$$D_B = Q_A$$

$$SEQ\_H = Q_B$$

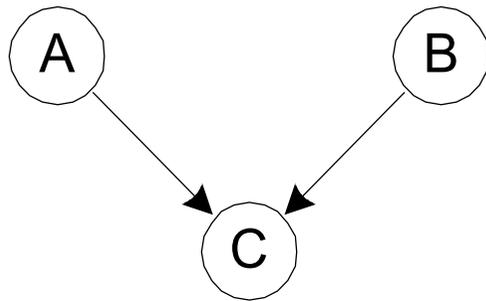




## CIRCUITOS SEQUENCIAIS SÍNCRONOS - 41

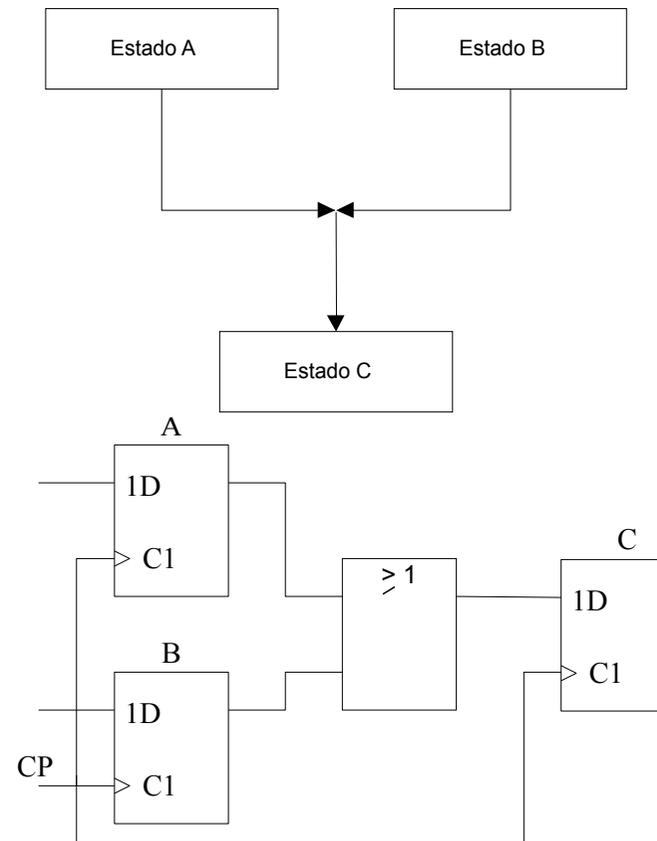
### ● UM FLIP-FLOP / ESTADO (cont.)

(3) Convergência de Ramos correspondentes a Transições entre Estados:

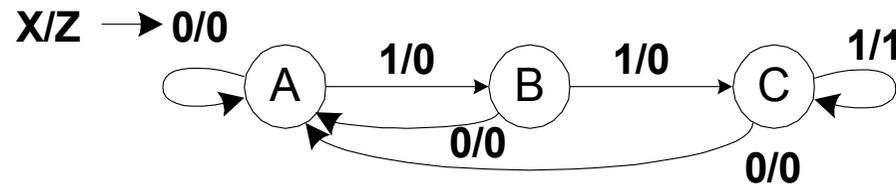


Equação:

$$D_C = Q_A + Q_B$$



## ● UM FLIP-FLOP / ESTADO – EXEMPLO SEQ-MEALY



Estado Presente	Estado Seguinte e Saída Presente (Z)	
	X=0	X=1
A	A/0	B/0
B	A/0	C/0
C	A/0	C/1

Derivação “directa” das equações:

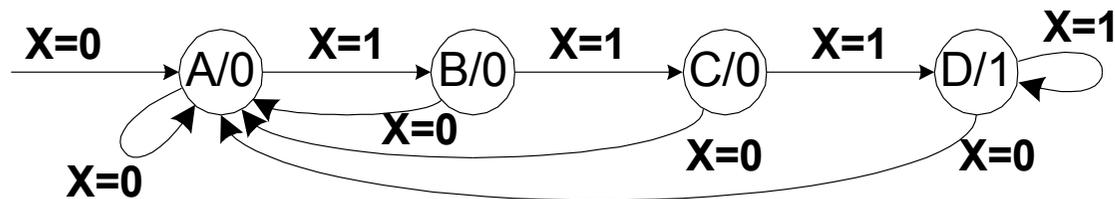
$$\begin{aligned}
 D_A &= \bar{X} \cdot Q_A + \bar{X} \cdot Q_B + \bar{X} \cdot Q_C \\
 &= \bar{X} \cdot (Q_A + Q_B + Q_C) \\
 &= \bar{X}
 \end{aligned}$$

$$D_B = X \cdot Q_A$$

$$D_C = X \cdot Q_B + X \cdot Q_C$$

$$Z = X \cdot Q_C$$

- **UM FLIP-FLOP / ESTADO – EXEMPLO SEQ-MOORE**



**Equações:**

$$D_A = \bar{X} \cdot Q_A + \bar{X} \cdot Q_B + \bar{X} \cdot Q_C + \bar{X} \cdot Q_D = \bar{X} \cdot (Q_A + Q_B + Q_C + Q_D)$$

$$D_B = X \cdot Q_A$$

$$D_C = X \cdot Q_B$$

$$D_D = X \cdot Q_C + X \cdot Q_D$$

$$Z = Q_D$$

### ● UM FLIP-FLOP / ESTADO – EXEMPLO (CONT.)

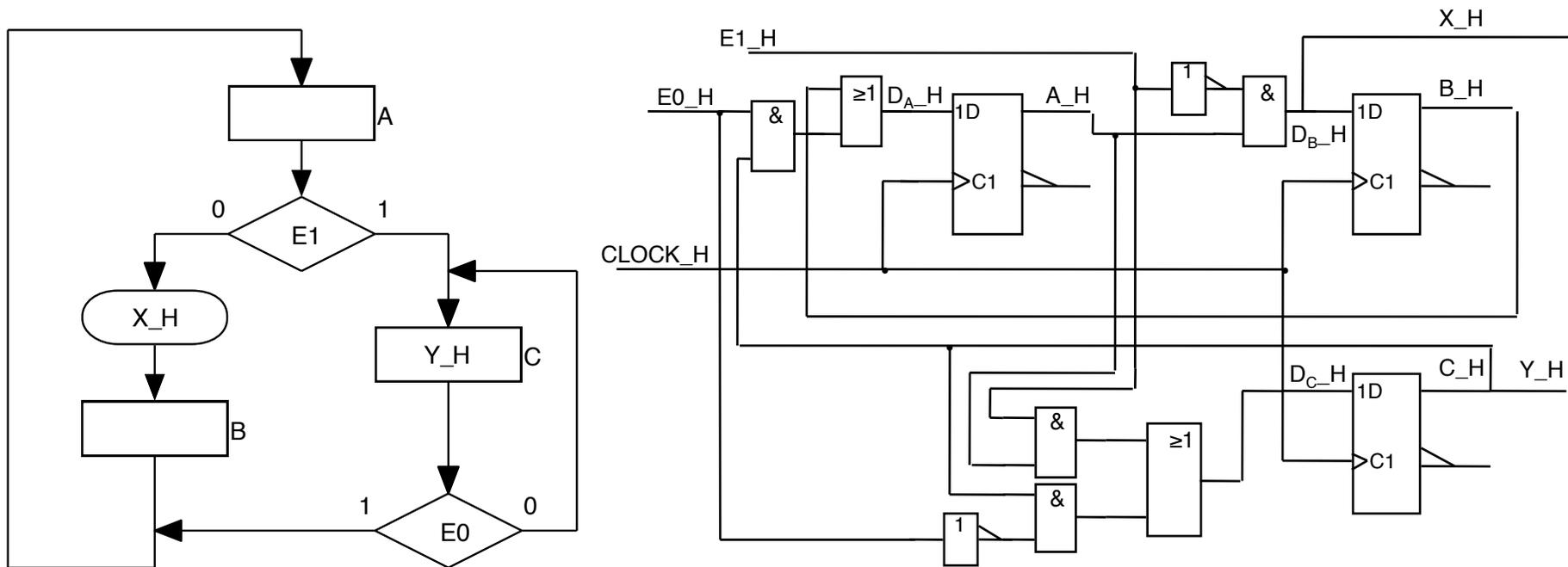
No presente caso, devido ao reduzido número de estados a síntese clássica permite uma maior simplicidade na solução sem grande trabalho adicional ao nível da síntese.

Contudo, este exemplo serve para evidenciar a facilidade com que se pode passar de um diagrama de estados ou fluxograma directamente para um diagrama lógico.

**Nota 1:** por simplicidade não se representam os sinais de preset e clear dos FFs que permitem impôr o estado inicial no circuito – neste tipo de codificação é sempre necessário inicializar a máquina! (porquê?)

**Nota 2:** em tecnologias em que o custo de um flip-flop é o mesmo de uma porta lógica básica, este tipo de codificação é muitas vezes o mais adequado, pois o acréscimo de FFs é compensado pela simplicidade do projecto e, algumas vezes, também por uma redução na lógica combinatória.

## ● UM FLIP-FLOP / ESTADO – EXEMPLO FLUXOGRAMA



$$D_A = B + C E0$$

$$D_B = A \overline{E1}$$

$$D_C = A E1 + C \overline{E0}$$

$$X = A \overline{E1}$$

$$Y = C$$

## CIRCUITOS SEQUENCIAIS SÍNCRONOS - 46

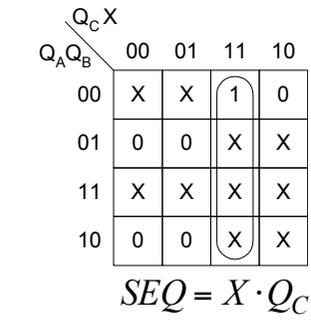
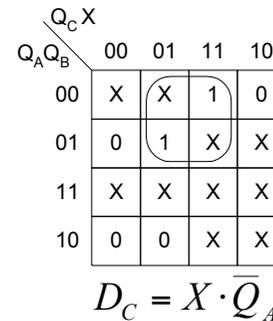
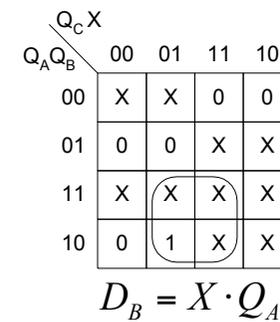
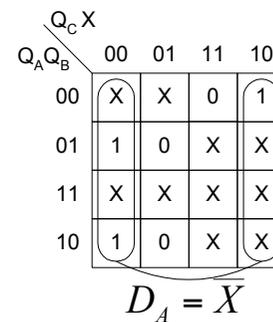
### ● UM FLIP-FLOP / ESTADO

Projecto segundo a síntese clássica (detecção da sequência “111” – máq. Mealy):

#### Codificação de Estados

	$Q_A$	$Q_B$	$Q_C$
A	1	0	0
B	0	1	0
C	0	0	1

Estado Presente	Estado Seguinte e Saída Presente (SEQ)	
	X=0	X=1
A	A/0	B/0
B	A/0	C/0
C	A/0	C/1



**ATENÇÃO:** Um projecto de 1 FF por estado NUNCA deve ser feito desta forma ineficiente (e que está limitada pela dimensão dos mapas de Karnaugh)

### ● PROJECTOS ALTERNATIVOS – CONTADORES

Uma outra alternativa em termos de projecto corresponde à utilização de contadores integrados na implementação das máquinas de estados.

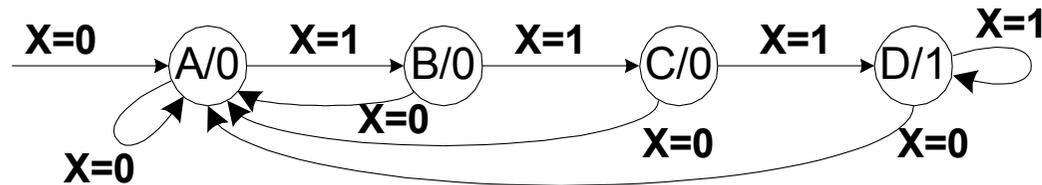
Esta solução é especialmente útil no caso de existir um elevado número de transições incondicionais entre estados, o que reduz consideravelmente a necessidade de carregamento paralelo.

Deste modo a utilização de contadores na implementação de máquinas de estado requer o projecto da:

- lógica combinatória que gera o sinal de carregamento paralelo (LOAD)
- lógica combinatória que gera as entradas de carregamento paralelo (dada pelo estado seguinte quando há quebra de sequência)

### ● PROJECTOS ALTERNATIVOS – CONTADORES

**EXEMPLO:** Detector da sequência “111”



Naturalmente, este exemplo apenas requer a utilização de um contador de módulo 4. Admitindo que só está disponível um contador de módulo 16, só se utilizarão os 2 bits de menor peso.

Considerando a seguinte codificação de estados:

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

teremos a seguinte sequência de estados de contagem:

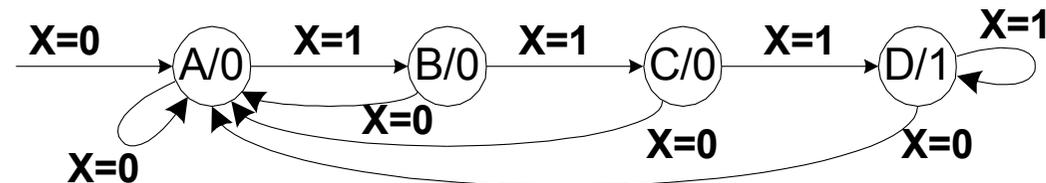
A	00
B	01
C	10
D	11

### ● PROJECTOS ALTERNATIVOS – CONTADORES

#### EXEMPLO (cont)

... e os seguintes casos em que a máquina de estados quebra a sequência de contagem:

E.A.	Entrada	E.S.
00	X=0	00
01	X=0	00
10	X=0	00
11	X=0	00
11	X=1	11



De seguida devem construir-se os mapas de Karnaugh de 3 entradas (2 bits de E.A. e X) para as várias funções pretendidas: sinal de carregamento paralelo (LOAD\_L); entradas de carregamento paralelo para os 2 bits de menor peso do contador.

## CIRCUITOS SEQUENCIAIS SÍNCRONOS - 50

### ● PROJECTOS ALTERNATIVOS – CONTADORES

#### EXEMPLO (cont)

#### 1) Sinal de Activação do Carregamento Paralelo:

Q1Q0 x	00	10	11	01
0	0	0	0	0
1	1	1	0	1

$LOAD\_L = x \cdot \overline{Q_1} \overline{Q_0}$

#### 2) Determinação da entrada D0:

Q1Q0 x	00	10	11	01
0	0	0	0	0
1	x	x	1	x

$$D0 = x$$

#### 3) Determinação da entrada D1:

Q1Q0 x	00	10	11	01
0	0	0	0	0
1	x	x	1	x

$$D1 = x$$

## ● PROJECTOS ALTERNATIVOS – CONTADORES

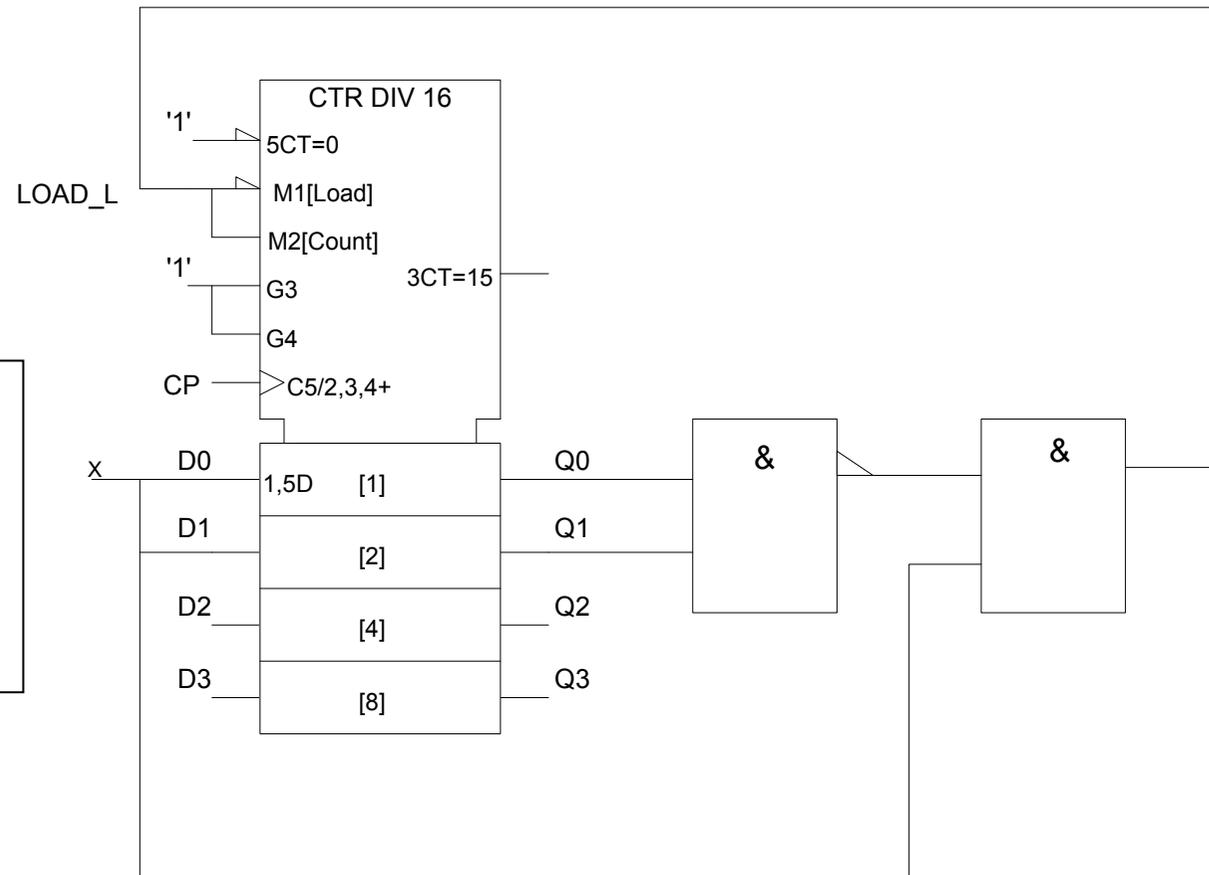
### EXEMPLO (cont):

**Nota:** para situações mais complexas faria sentido a utilização de um decodificador para detectar os estados que conduzem a carregamentos paralelos

$$D_0 = X$$

$$D_1 = X$$

$$LOAD\_L = X\overline{Q_1}Q_0$$



**(NOTA: A matéria constante nos slides seguintes não foi dada nas aulas teóricas)**

### ● SIMPLIFICAÇÃO POR ELIMINAÇÃO DE ESTADOS REDUNDANTES

Embora as máquinas de Moore e Mealy sejam conceptualmente elementares a sua utilização na resolução de alguns problemas de controlo, mais complexos do que simples detectores de sequência, pode facilmente conduzir a um elevado número de estados e transições. Deste modo, pode tornar-se extremamente complexo obter a melhor solução directamente.

Como é óbvio, a utilização de uma representação com o número mínimo de estados tem vantagens claras tanto na síntese, como na implementação.

2 estados são equivalentes se, para as mesmas entradas, transitam para os mesmos estados e têm as mesmas saídas:

Estado Presente	Estado Seguinte e Saída Presente (Z)	
	X=0	X=1
...	...	...
P	R/0	S/1
...	...	...
Q	R/0	S/1
...	...	...

**Nota:** As técnicas descritas são válidas tanto para Mealy como Moore e com um número arbitrário de estados, entradas e saídas.

### ● ELIMINAÇÃO POR INSPECÇÃO

Estado Presente	Estado Seguinte e Saída Presente (Z)	
	X=0	X=1
A	B/0	C/1
<b>B</b>	<b>C/0</b>	<b>A/1</b>
C	D/1	B/0
<b>D</b>	<b>C/0</b>	<b>A/1</b>
E	D/0	C/1

→

Estado Presente	Estado Seguinte e Saída Presente (Z)	
	X=0	X=1
A	B/0	C/1
B	C/0	A/1
C	<b>B/1</b>	B/0
E	<b>B/0</b>	C/1

→

Estado Presente	Estado Seguinte e Saída Presente (Z)	
	X=0	X=1
A	B/0	C/1
B	C/0	A/1
C	B/1	B/0

## CIRCUITOS SEQUENCIAIS SÍNCRONOS - 54

### ● ELIMINAÇÃO POR PARTIÇÃO

No caso anterior, a existência de estados redundantes era evidente pela explicitação de *estados seguintes* e saídas idênticas para diferentes *estados presentes*.

No entanto, 2 estados são equivalentes se, para as mesmas entradas, transitam para **estados equivalentes** e têm as mesmas saídas.

O **método das partições** permite identificar estados redundantes também quando os estados seguintes não sejam inicialmente os mesmos.

Estado Presente	Estado Seguinte e Saída Presente (Z)	
	X=0	X=1
A <sub>1</sub>	B/0	C/0
B <sub>1</sub>	D/0	E/0
C <sub>1</sub>	G/0	E/0
D <sub>1</sub>	H/0	F/0
E <sub>1</sub>	G/0	A/0
F <sub>1</sub>	G/1	A/0
G <sub>1</sub>	D/0	C/0
H <sub>1</sub>	H/0	A/0

→

Estado Presente	Estado Seguinte e Saída Presente (Z)	
	X=0	X=1
A <sub>1</sub>	B <sub>1</sub>	C <sub>1</sub>
B <sub>1</sub>	D <sub>1</sub>	E <sub>1</sub>
C <sub>1</sub>	G <sub>1</sub>	E <sub>1</sub>
D <sub>1</sub>	H <sub>1</sub>	F <sub>2</sub>
E <sub>1</sub>	G <sub>1</sub>	A <sub>1</sub>
F <sub>2</sub>	G <sub>1</sub>	A <sub>1</sub>
G <sub>1</sub>	D <sub>1</sub>	C <sub>1</sub>
H <sub>1</sub>	H <sub>1</sub>	A <sub>1</sub>

→

Estado Presente	Estado Seguinte e Saída Presente (Z)	
	X=0	X=1
A <sub>1</sub>	B <sub>1</sub>	C <sub>1</sub>
B <sub>1</sub>	D <sub>3</sub>	E <sub>1</sub>
C <sub>1</sub>	G <sub>1</sub>	E <sub>1</sub>
D <sub>3</sub>	H <sub>1</sub>	F <sub>2</sub>
E <sub>1</sub>	G <sub>1</sub>	A <sub>1</sub>
F <sub>2</sub>	G <sub>1</sub>	A <sub>1</sub>
G <sub>1</sub>	D <sub>3</sub>	C <sub>1</sub>
H <sub>1</sub>	H <sub>1</sub>	A <sub>1</sub>

## CIRCUITOS SEQUENCIAIS SÍNCRONOS - 55

### ● ELIMINAÇÃO POR PARTIÇÃO (cont.)

Estado Presente	Estado Seguinte e Saída Presente (Z)	
	X=0	X=1
A <sub>1</sub>	B <sub>1</sub>	C <sub>1</sub>
B <sub>1</sub>	D <sub>3</sub>	E <sub>1</sub>
C <sub>1</sub>	G <sub>1</sub>	E <sub>1</sub>
D <sub>3</sub>	H <sub>1</sub>	F <sub>2</sub>
E <sub>1</sub>	G <sub>1</sub>	A <sub>1</sub>
F <sub>2</sub>	G <sub>1</sub>	A <sub>1</sub>
G <sub>1</sub>	D <sub>3</sub>	C <sub>1</sub>
H <sub>1</sub>	H <sub>1</sub>	A <sub>1</sub>

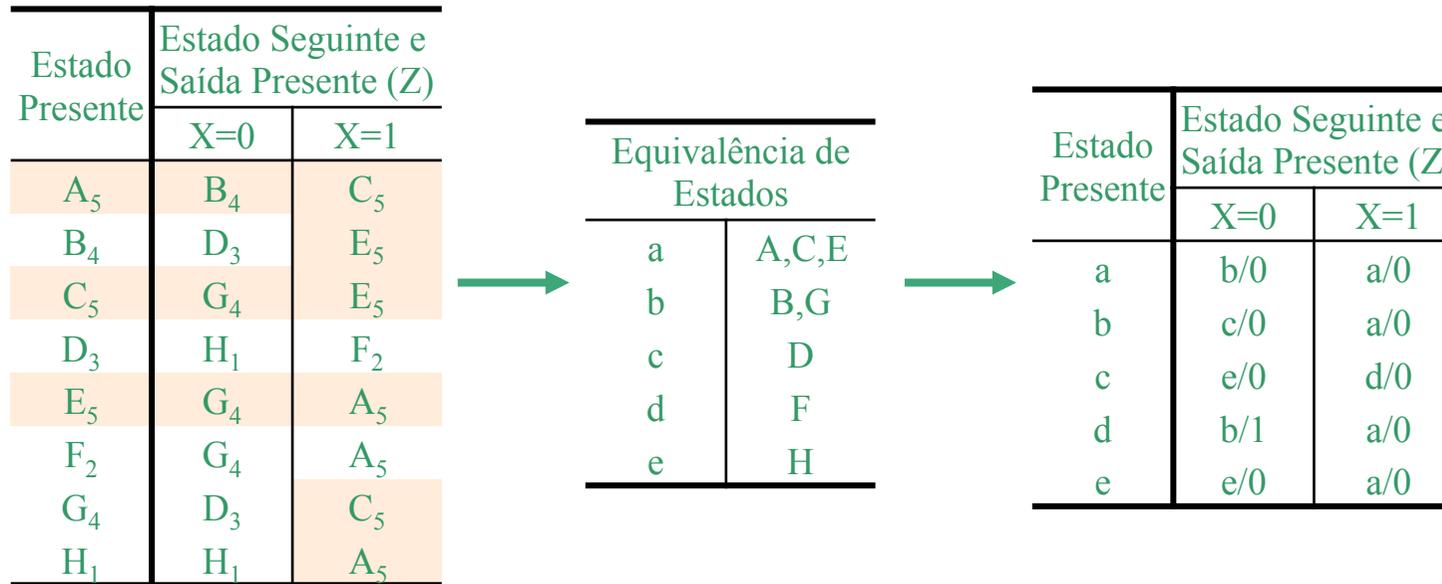
→

Estado Presente	Estado Seguinte e Saída Presente (Z)	
	X=0	X=1
A <sub>1</sub>	B <sub>4</sub>	C <sub>1</sub>
B <sub>4</sub>	D <sub>3</sub>	E <sub>1</sub>
C <sub>1</sub>	G <sub>4</sub>	E <sub>1</sub>
D <sub>3</sub>	H <sub>1</sub>	F <sub>2</sub>
E <sub>1</sub>	G <sub>4</sub>	A <sub>1</sub>
F <sub>2</sub>	G <sub>4</sub>	A <sub>1</sub>
G <sub>4</sub>	D <sub>3</sub>	C <sub>1</sub>
H <sub>1</sub>	H <sub>1</sub>	A <sub>1</sub>

→

Estado Presente	Estado Seguinte e Saída Presente (Z)	
	X=0	X=1
A <sub>5</sub>	B <sub>4</sub>	C <sub>5</sub>
B <sub>4</sub>	D <sub>3</sub>	E <sub>5</sub>
C <sub>5</sub>	G <sub>4</sub>	E <sub>5</sub>
D <sub>3</sub>	H <sub>1</sub>	F <sub>2</sub>
E <sub>5</sub>	G <sub>4</sub>	A <sub>5</sub>
F <sub>2</sub>	G <sub>4</sub>	A <sub>5</sub>
G <sub>4</sub>	D <sub>3</sub>	C <sub>5</sub>
H <sub>1</sub>	H <sub>1</sub>	A <sub>5</sub>

### ● ELIMINAÇÃO POR PARTIÇÃO (cont.)



### ● CONVERSÃO DE MOORE PARA MEALY

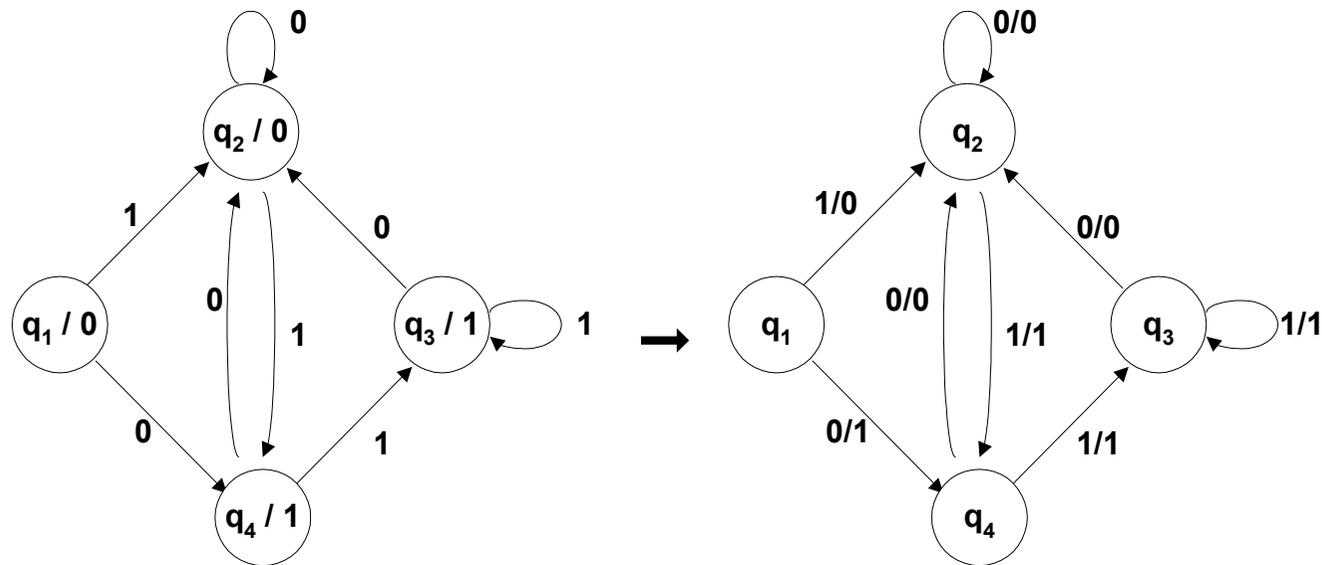
No modelo de Moore uma entrada define um caminho para um estado e só aí a saída sofre alteração, enquanto no modelo de Mealy a saída é alterada assim que a entrada varia.

A conversão de Moore para Mealy consiste, portanto, na antecipação das saídas do seguinte modo:



### ● CONVERSÃO DE MOORE PARA MEALY (cont.)

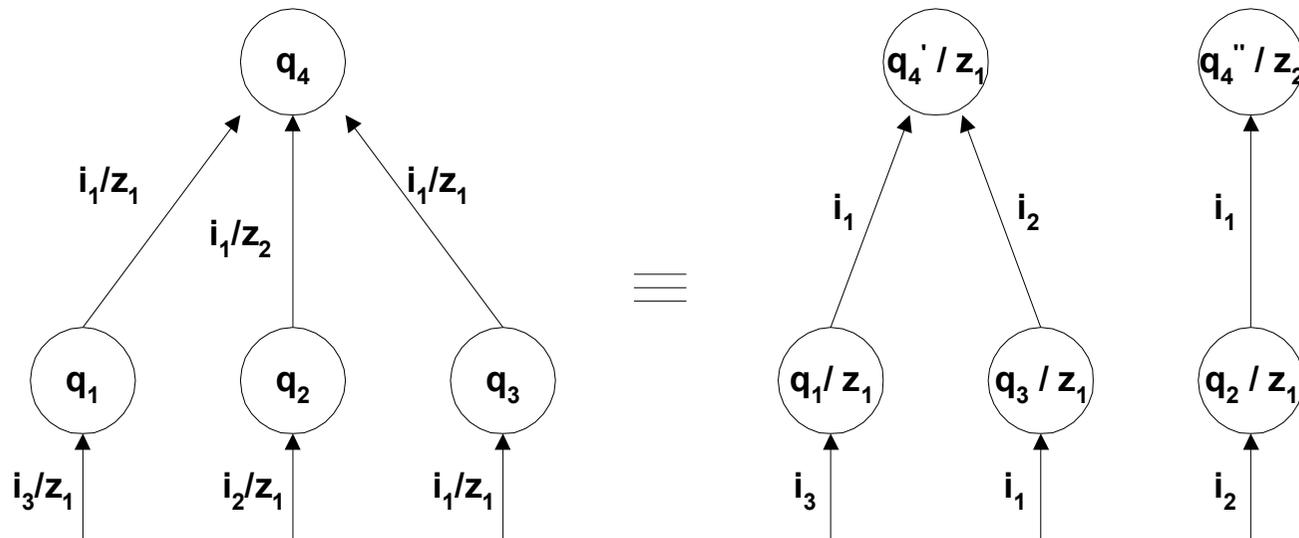
Exemplo:



**Nota:** a conversão descrita não minimiza o número de estados da máquina de Mealy, e.g., obtém-se um diagrama de Mealy com 4 em vez de 3 estados. A solução é identificar os estados redundantes ou recorrer a um processo sistemático de simplificação (a estudar mais à frente).

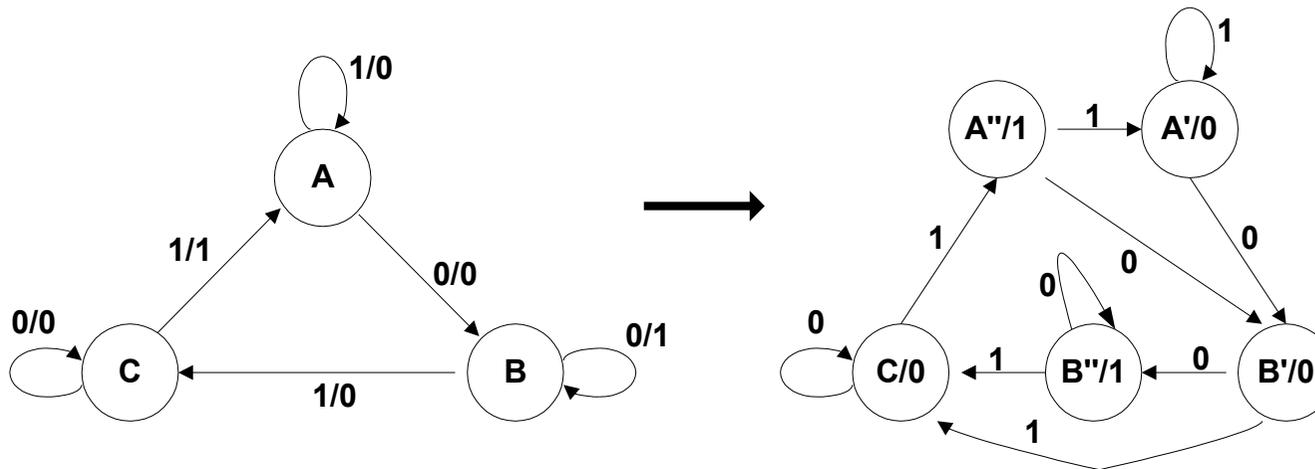
### ● CONVERSÃO DE MEALY PARA MOORE

Neste caso, a existência de ramos com saídas diferentes no modelo de Mealy a convergirem para um mesmo estado obriga à inserção de novos estados para construir o diagrama de Moore.



### ● CONVERSÃO DE MEALY PARA MOORE (cont.)

Exemplo:



### BIBLIOGRAFIA

- [1] Arroz,G., Monteiro,J.C., Oliveira,A., “Arquitectura de Computadores, dos Sistemas Digitais aos Microprocessadores”, cap.7, 2ª Edição, 2009
- [2] M. Morris Mano, Charles R. Kime, “Logic and Computer Design Fundamentals”, Prentice-Hall International, Inc. (Capítulo 4, Secções 4.4 a 4.7)