

CIRCUITOS SEQUENCIAIS SÍNCRONOS

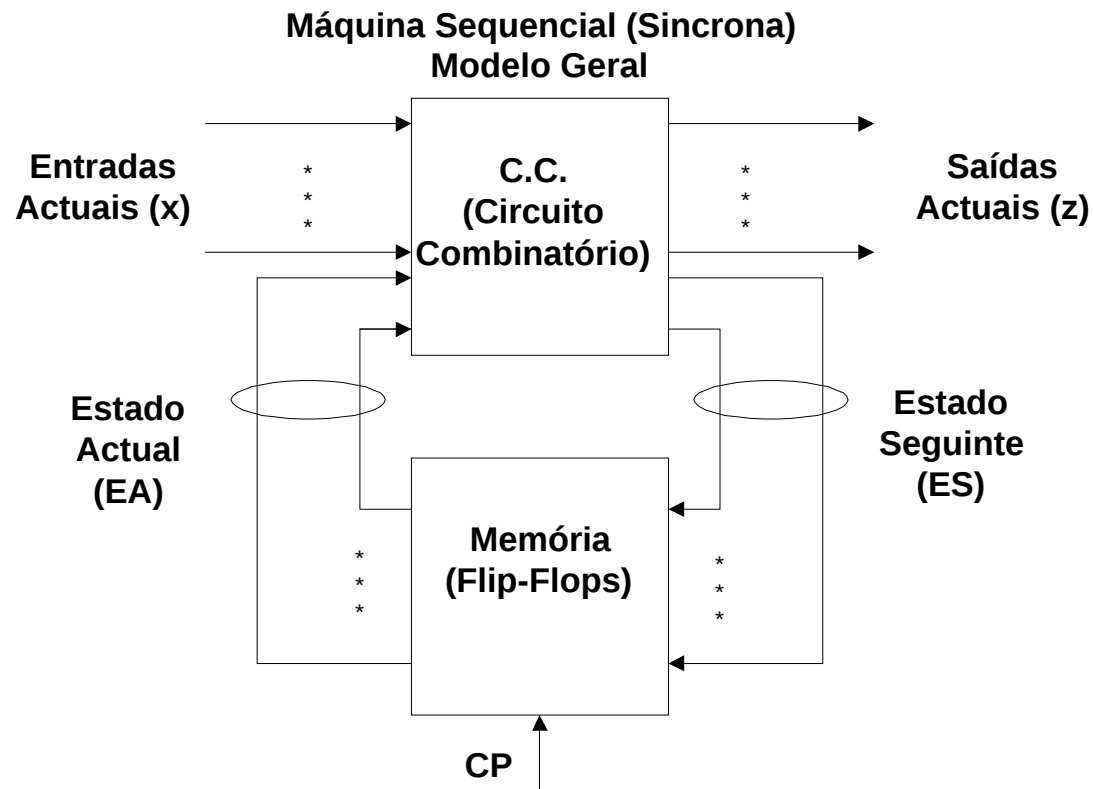
● SUMÁRIO:

- CIRCUITOS DE MOORE E MEALY
 - CARACTERIZAÇÃO
 - ESPECIFICAÇÃO
 - SÍNTESE
 - CONVERSÃO
 - EXEMPLOS
- PROJECTOS ALTERNATIVOS
 - 1 FLIP-FLOP / ESTADO
 - UTILIZAÇÃO DE CONTADORES
- SIMPLIFICAÇÃO DE MÁQUINAS DE ESTADOS

CIRCUITOS SEQUENCIAIS SÍNCRONOS - 3

● CIRCUITOS DE MOORE E MEALY (CARACTERIZAÇÃO)

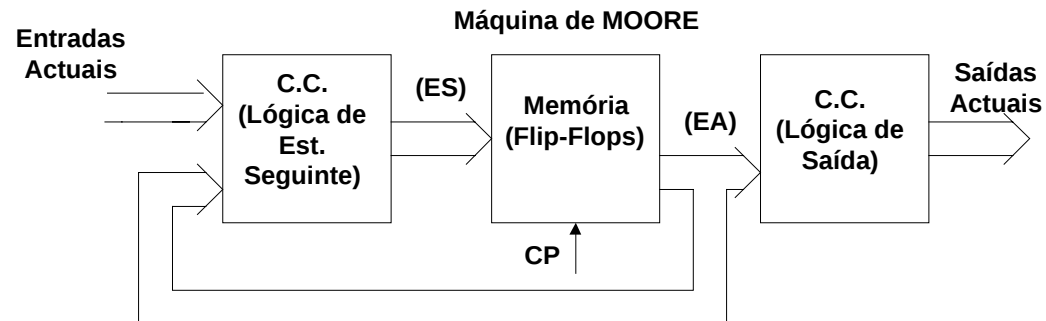
MODELO GERAL



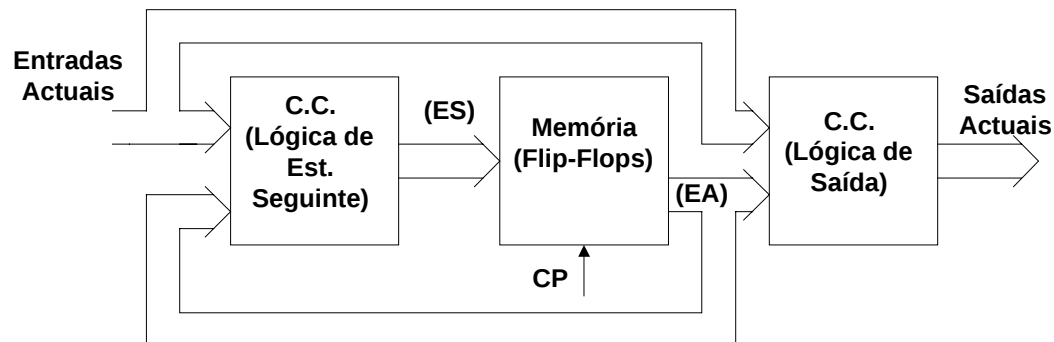
CIRCUITOS SEQUENCIAIS SÍNCRONOS - 4

● CIRCUITOS DE MOORE E MEALY (CARACTERIZAÇÃO)

Modelo de Moore: Circuito no qual as saídas são função directa do estado.



Modelo de Mealy: Circuito no qual as saídas são função do estado e das entradas.



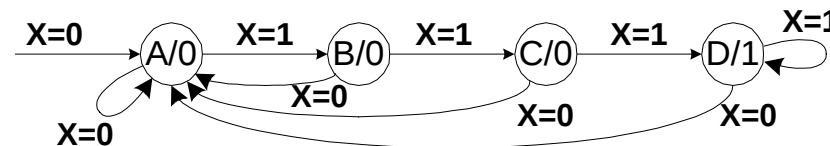
Nota: Em geral, os circuitos de Moore apresentam uma maior simplicidade na geração das saídas, enquanto os circuitos de Mealy conduzem a um menor número de estados e à eventual redução do número de FFs necessários.

CIRCUITOS SEQUENCIAIS SÍNCRONOS - 5

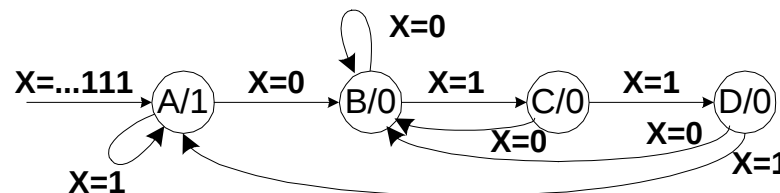
● MÁQUINA DE MOORE (ESPECIFICAÇÃO)

Para análise da especificação e síntese de máquinas de estados segundo os modelos de Moore e de Mealy, utilizar-se-á um **exemplo do projecto de um detector de sequência**. Em particular, o projecto a realizar terá por objectivo determinar 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 as saídas que lhe estão associadas (máquina de Moore). Cada **transição** entre estados é descrita através de um vector ao qual está associado o valor das entradas que conduzem a essa transição.



OU

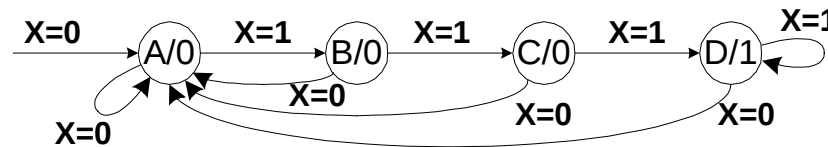


CIRCUITOS SEQUENCIAIS SÍNCRONOS - 6

● MÁQUINA DE MOORE (SÍNTESE)

Tabelas de Transição de Estados e Tabelas de Excitação

Por fim, a especificação da máquina de estados pode ainda ser realizada através da construção de uma **tabela de transição de estados**, onde se explicitam 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

Na síntese da máquina de estados é necessário substituir a representação simbólica dos estados por códigos binários – **codificação de estados**.

Se a máquina tem 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 (corresponde ao número mínimo de flip-flops). No entanto, veremos adiante que a utilização de um número de bits de estado superior ao mínimo pode ser, nalguns casos, vantajosa.

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

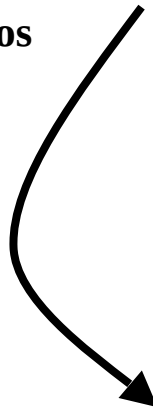
● MÁQUINA DE MOORE (SÍNTESE)

Tabelas de Transição de Estados e Tabelas de Excitação (cont)

Após a especificação da máquina pretendida o processo de síntese inicia-se com a atribuição de uma codificação aos estados e a selecção dos FFs a utilizar (conforme descrito na síntese de contadores).

**Tabelas 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 JK)

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

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

J₁

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

K₁

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

J₀

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

K₀

$$J_1 = XQ_0$$

$$K_1 = \bar{X}$$

$$J_0 = X$$

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

$$Z = Q_1Q_0$$

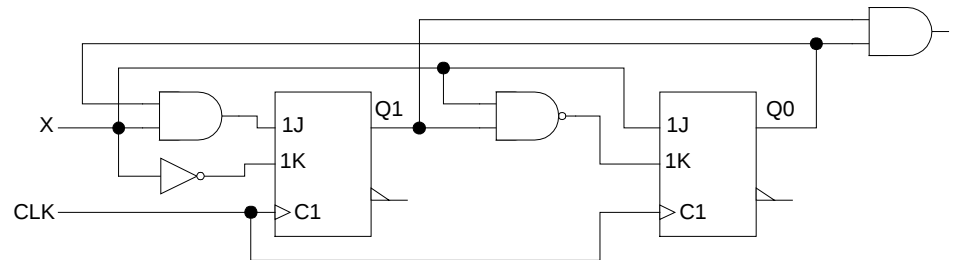
Q_1	0	1
0	0	0
1	0	1

Z

CIRCUITOS SEQUENCIAIS SÍNCRONOS - 10

● MÁQUINA DE MOORE (DIAGRAMA LÓGICO COM FF JK)

Por fim, obtém-se o seguinte diagrama lógico, correspondente ao detector de sequência pretendido:

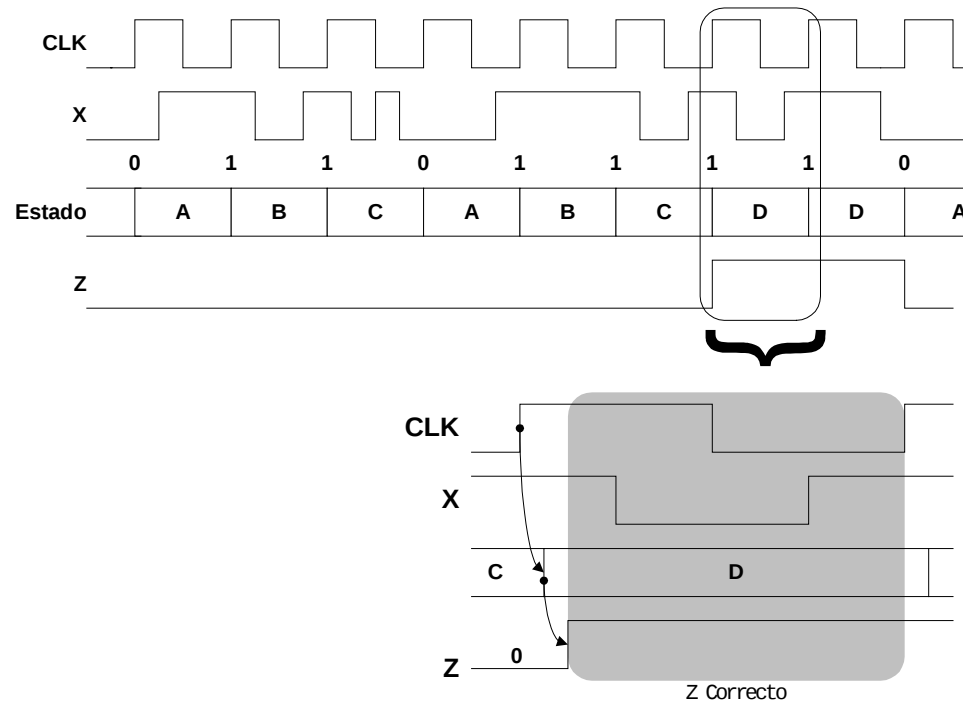


Sugestão: determine a frequência máxima com que este circuito funciona correctamente.

CIRCUITOS SEQUENCIAIS SÍNCRONOS - 11

● 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.



CIRCUITOS SEQUENCIAIS SÍNCRONOS - 12

● MÁQUINA DE MOORE – FF D – codificação 1

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

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

Q_0	0	1
Q_1		
0	0	0
1	0	1

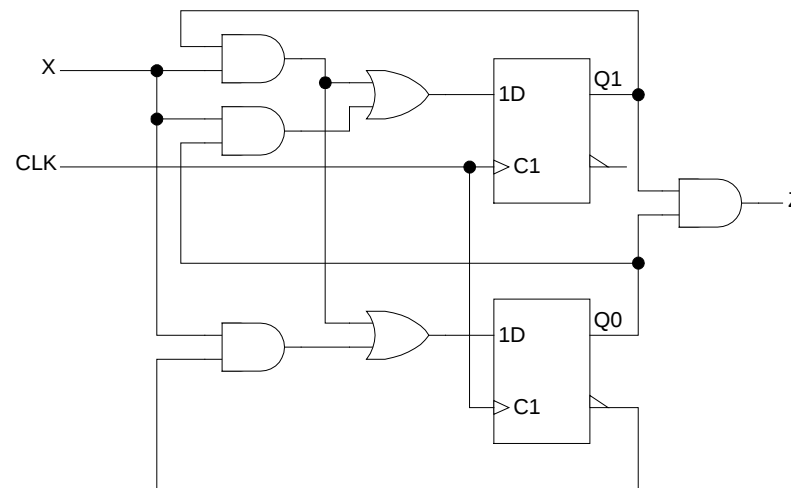
Codificação
de Estados

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

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

$$D_0 = X \cdot Q_1 + X \cdot \overline{Q_0}$$

$$Z = Q_1 \cdot Q_0$$



CIRCUITOS SEQUENCIAIS SÍNCRONOS - 13

● MÁQUINA DE MOORE – FF D – codificação 2

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

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

Q_0		0	1
Q_1	0	0	0
	1	1	0

Codificação
de Estados

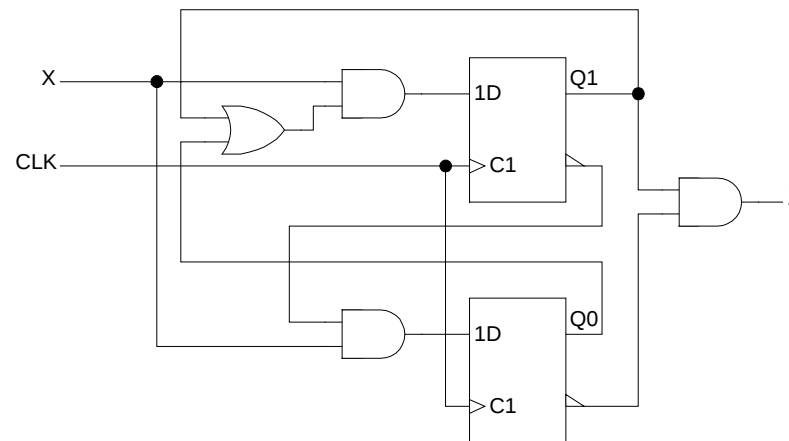
A	0	0
B	0	1
C	1	1
D	1	0

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

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

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

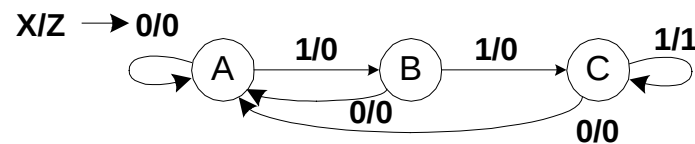
$$Z = Q_1 \cdot \bar{Q}_0$$



● MÁQUINA DE MEALY (ESPECIFICAÇÃO)

Para o estudo das **máquinas de Mealy** considera-se, tal como no caso das máquinas de Moore, o projecto de um detector de sequência de três '1's na entrada X.

Neste caso o **diagrama de estados** altera-se de acordo com a filosofia da máquina de Mealy, ou seja, a saída deixa de estar exclusivamente dependente do estado e daí a sua representação surgir, habitualmente, nos **vectores correspondentes às transições entre estados** onde se **reflecte**, também, a influência das entradas nessas **saídas**.

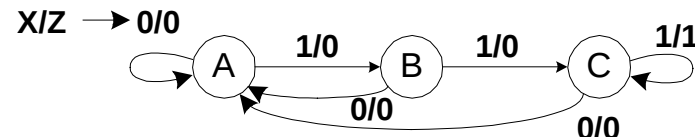


Conforme se pode observar, neste caso o comportamento do circuito pode ser modelado com um **número de estados inferior** ao modelo de Moore. Contudo, neste caso particular não há redução do número de FFs, apenas a existência de indiferenças suplementares nos mapas de Karnaugh que podem conduzir a uma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

CIRCUITOS SEQUENCIAIS SÍNCRONOS - 15

● 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.



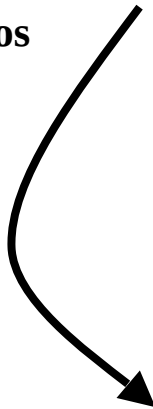
Estado Presente	Estado Seguinte e Saída Presente (Z)	
	X=0	X=1
A	A/0	B/0
B	A/0	C/0
C	A/0	C/1

● MÁQUINA DE MEALY (SÍNTESE)

A codificação de estados indicada, conduz a:

**Tabelas 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 - 17

● MÁQUINA DE MEALY (SÍNTESE COM FF JK)

Conforme referido atrás, a geração das entradas é simplificada pela existência de indiferenças adicionais nos mapas de Karnaugh, enquanto o mapa para a geração das saídas é mais complexo devido à dependência das entradas.

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

J_1

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

K_1

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

J_0

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

K_0

$$J_1 = XQ_0 \quad K_1 = \bar{X}$$

$$J_0 = X\bar{Q}_1 \quad K_0 = 1$$

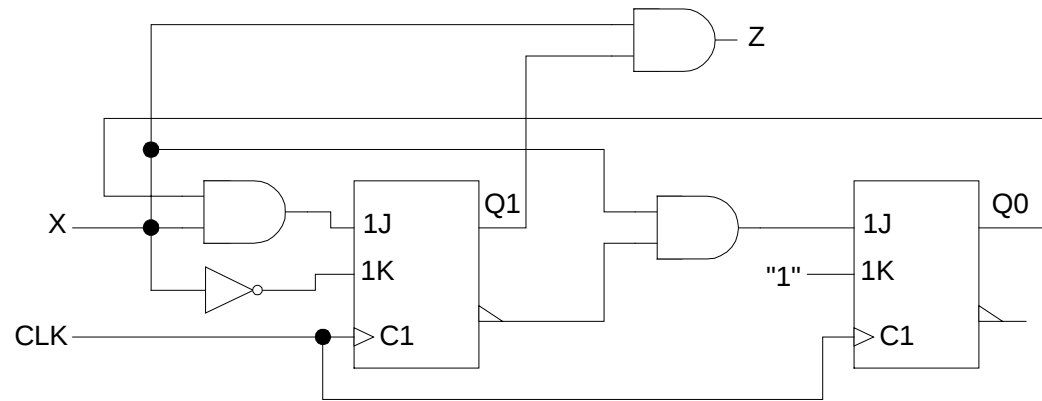
$$Z = XQ_1$$

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

Z

CIRCUITOS SEQUENCIAIS SÍNCRONOS - 18

● MÁQUINA DE MEALY (DIAGRAMA LÓGICO COM FF JK)



Sugestão: determine a frequência máxima deste circuito e compare a solução obtida com o caso da máquina de Moore.

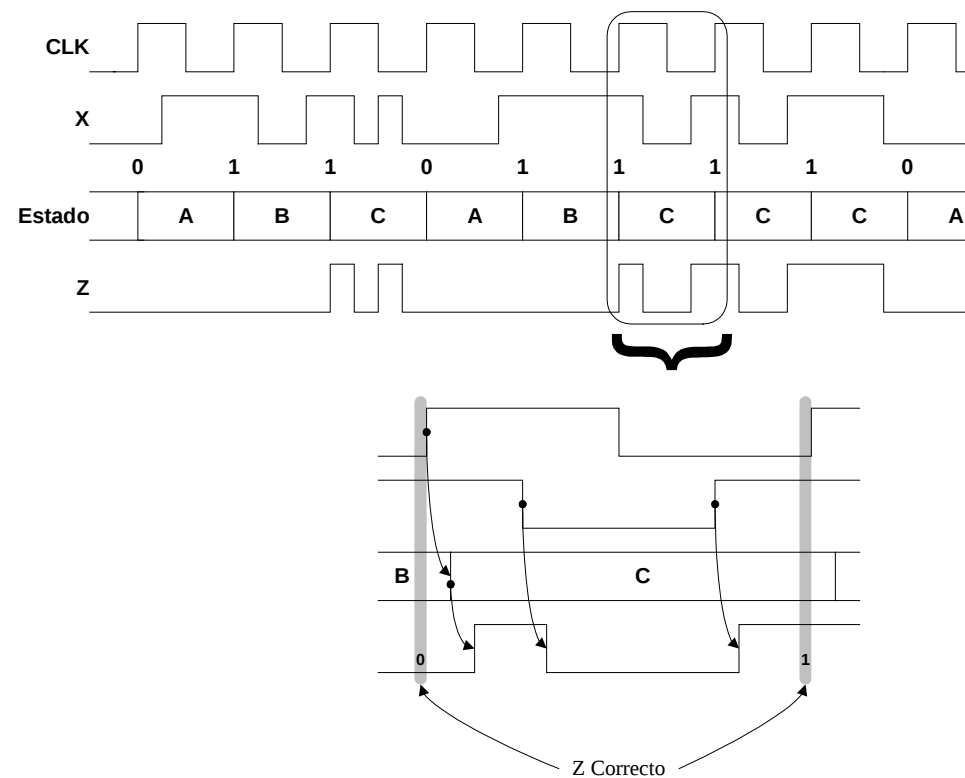
Exemplo 1: Determine os **diagramas de estado** para um detector da sequência 0101 (com e sem sobreposição) para as máquinas de Moore e Mealy.

Exemplo 2: Determine os **diagramas de estado** (Moore e Mealy) para um gerador de sequência onde se aguarda que a entrada seja '1' para gerar a sequência '1011'. A saída mantém-se a '1' até a entrada ser 0, nesta situação recomeça-se o ciclo.

CIRCUITOS SEQUENCIAIS SÍNCRONOS - 19

● MÁQUINA DE MEALY (DIAGRAMA TEMPORAL)

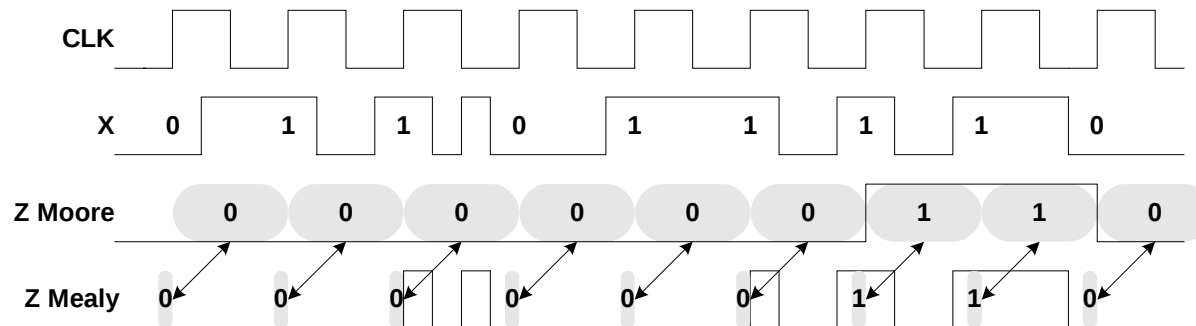
Na máquina de Mealy a saída pode ser directamente afectada por uma variação na entrada, mesmo que não haja mudança de estado. A saída tenta “antecipar” o valor que só é, de facto, correcto imediatamente antes da transição de relógio.



● MÁQUINA DE MOORE VS. MÁQUINA DE MEALY

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

A máquina de Mealy “antecipa” as saídas, mas em contrapartida tem um comportamento mais complexo.



CIRCUITOS SEQUENCIAIS SÍNCRONOS - 21

● MÁQUINA DE MEALY – FF D – codificação 1

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

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

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

Codificação
de Estados

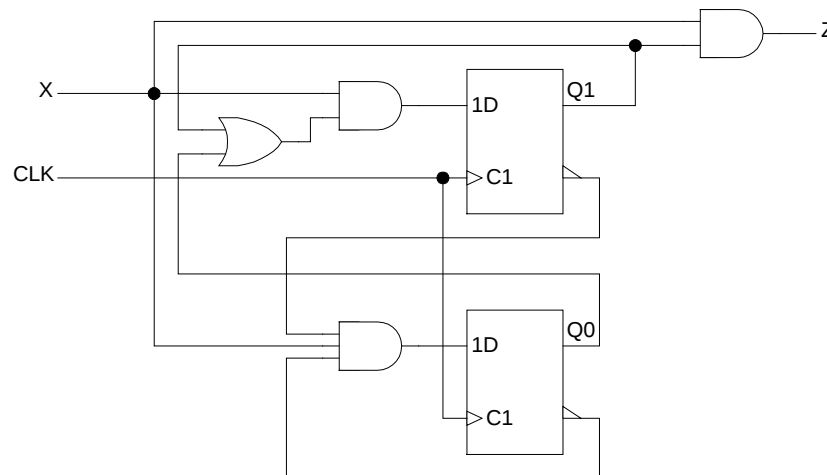
A	0	0
B	0	1
C	1	0

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

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

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

$$Z = X \cdot Q_1$$



CIRCUITOS SEQUENCIAIS SÍNCRONOS - 22

● MÁQUINA DE MEALY – FF D – codificação 2

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

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

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

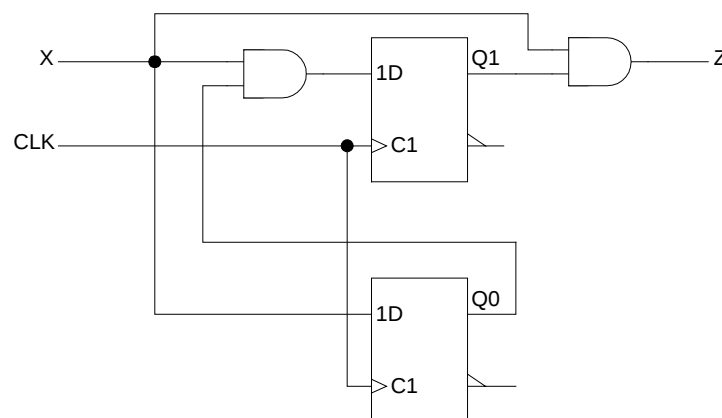
Codificação
de Estados

	0	0
A	0	0
B	0	1
C	1	1

$$D_1 = X \cdot Q_0$$

$$D_0 = X$$

$$Z = X \cdot Q_1$$



● CONVERSÃO DE MOORE PARA MEALY

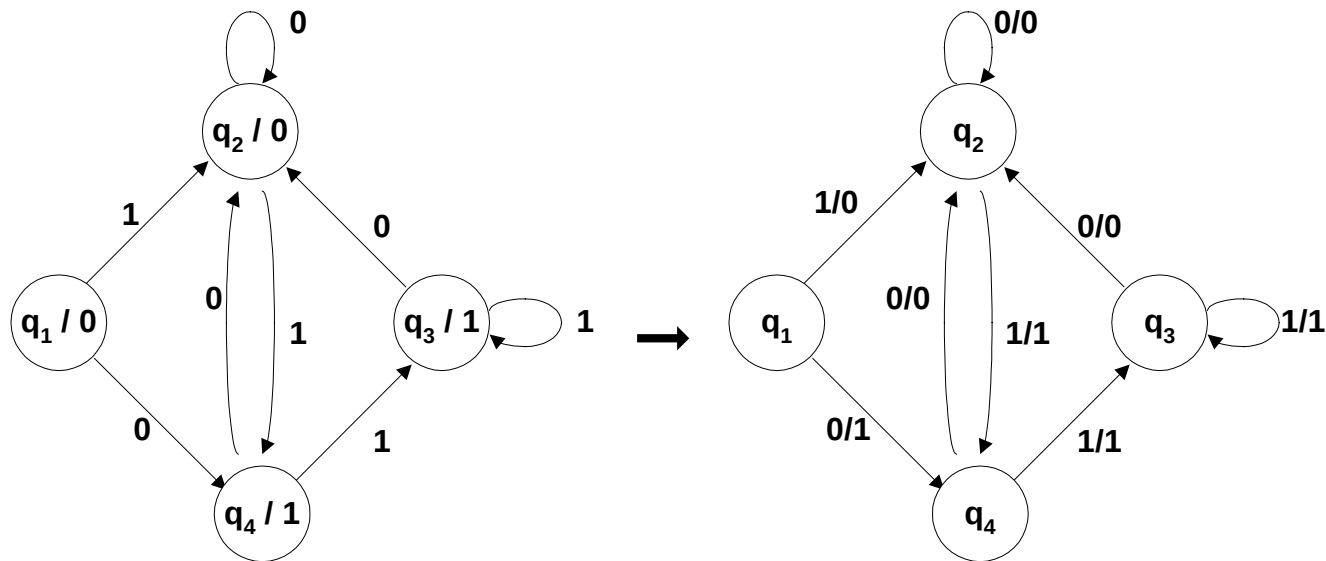
No modelo de Moore uma entrada define um caminho para um estado e só aí a saída sofre alteração, enquanto no modelo de Mealy a saída é alterada assim que a entrada varia.

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



● CONVERSÃO DE MOORE PARA MEALY (cont.)

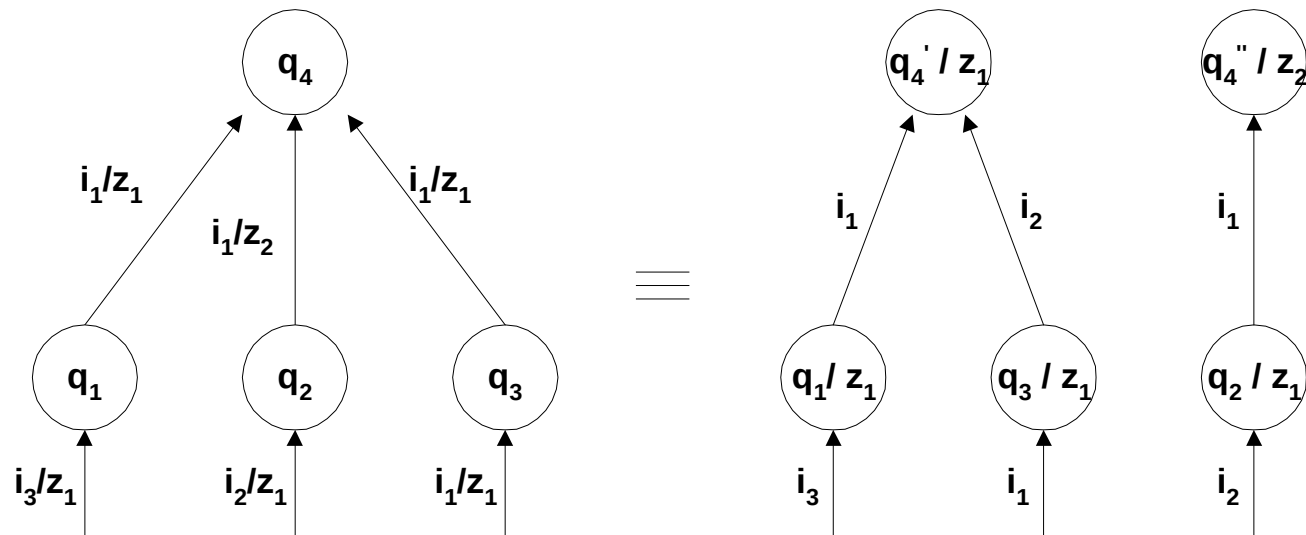
Exemplo:



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).

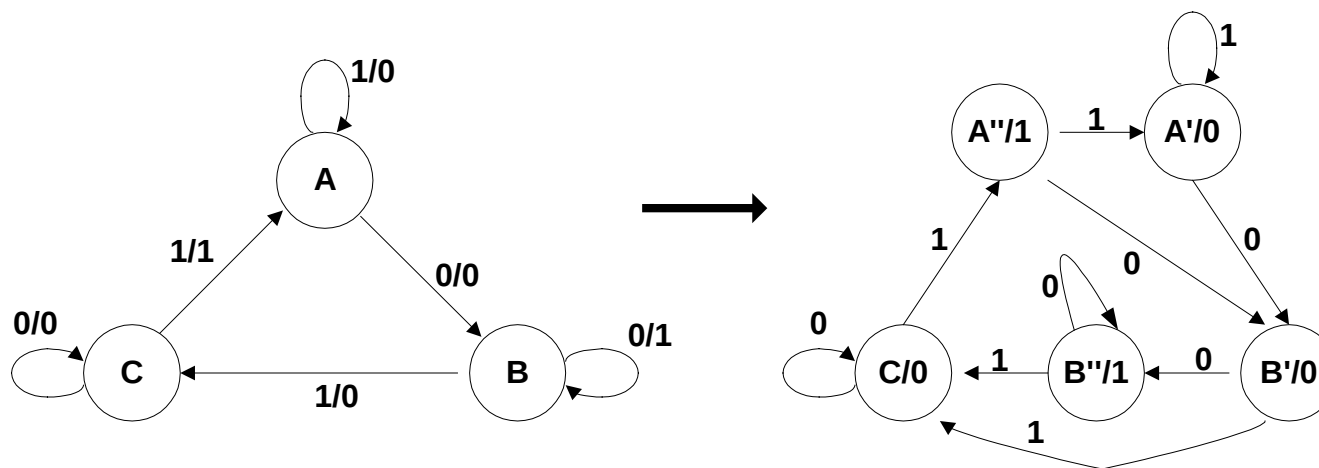
● CONVERSÃO DE MEALY PARA MOORE

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:



● 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;

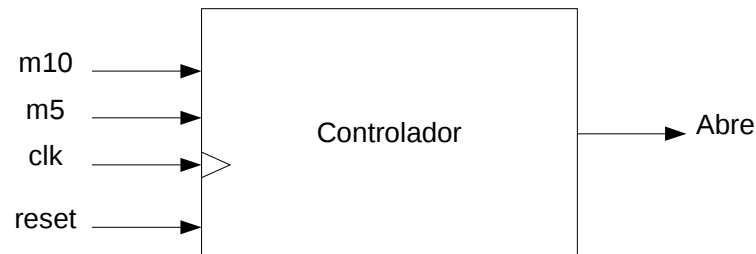
recebe 1 moeda de cada vez;

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 activa, abre a gaveta e entrega a pastilha ao cliente;

não é entregue troco se forem recebidas moedas a mais.



● MDP – DIAGRAMA DE ESTADOS

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

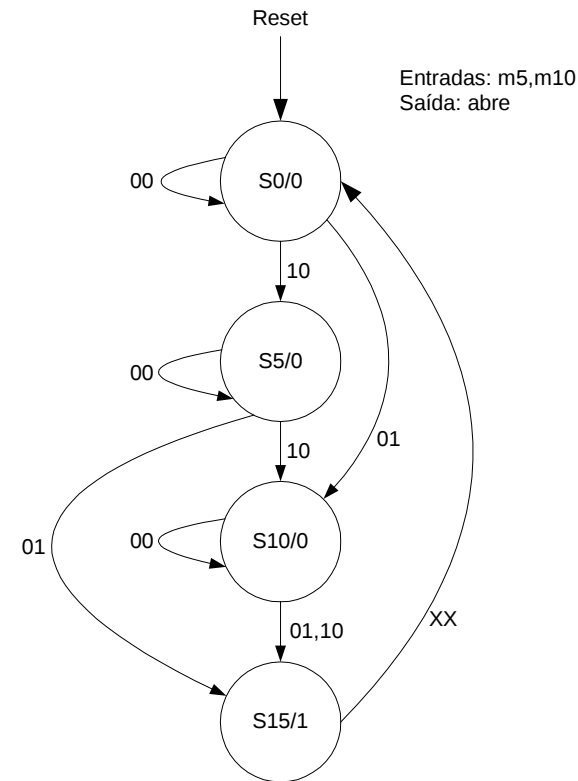
S0 – estado inicial, não foi introduzida moeda.

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 - 29

● MDP – TABELA DE TRANSIÇÃO DE ESTADOS

Estado Actual			Entradas		Estado Seguinte			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

● MDP – PROJECTO DA LÓGICA COMBINATÓRIA

		m5 m10			
$Q_1 Q_0$		00	01	11	10
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_0	
Q_1		0	1
0		0	0
1		0	1

K-mapa para Saída

$$Abre = Q_1 Q_0$$

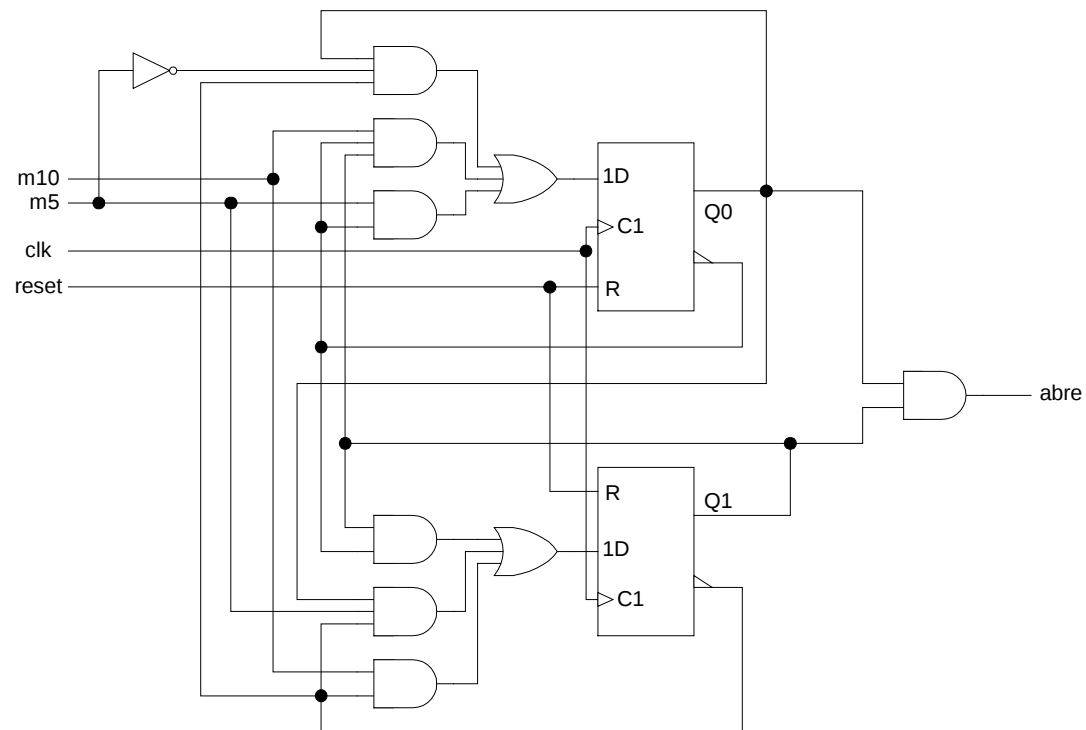
		m5 m10			
$Q_1 Q_0$		00	01	11	10
00		0	0	X	1
01		1	1	X	0
11		0	0	X	0
10		0	1	X	1

K-mapa para D0

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

CIRCUITOS SEQUENCIAIS SÍNCRONOS - 31

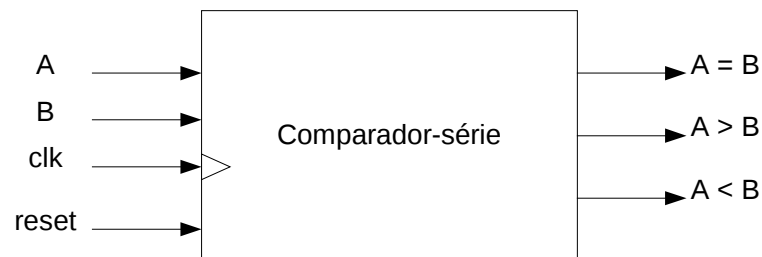
● MDP – LOGIGRAMA DO CIRCUITO



● EXEMPLO – Comparador Série (CMP)

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

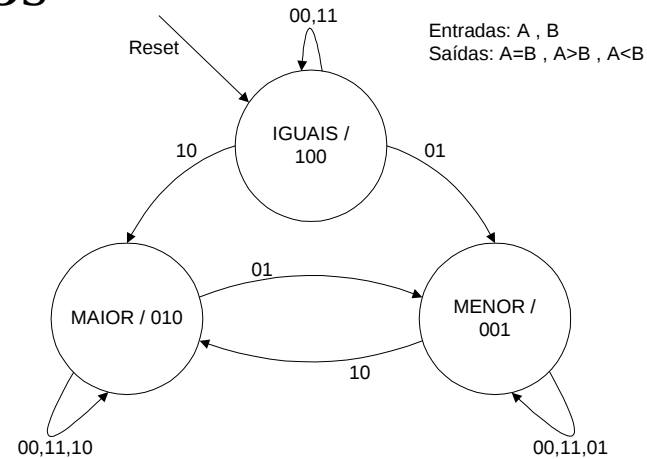
Os números são recebidos em série (bit a bit), com o bit de menor peso a ser recebido primeiro.



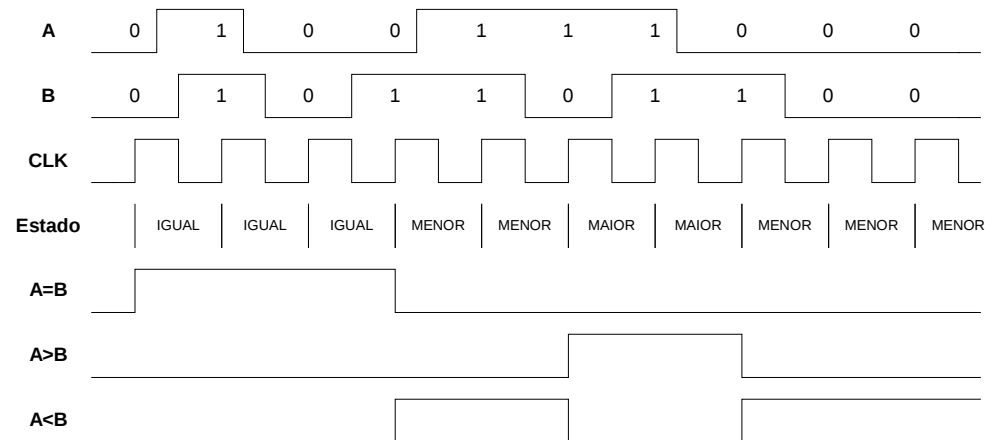
CIRCUITOS SEQUENCIAIS SÍNCRONOS - 33

● CMP1 – DIAGRAMA DE ESTADOS

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



► Exemplo



CIRCUITOS SEQUENCIAIS SÍNCRONOS - 34

● CMP1 – 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	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 - 35

● CMP1 – PROJECTO DA LÓGICA COMBINATÓRIA

		A B			
		00	01	11	10
Q ₁ Q ₀	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 ₁ Q ₀	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 ₀	
		0	1
Q ₁	0	1	0
	1	0	X

$(A = B) = \bar{Q}_1 \bar{Q}_0$

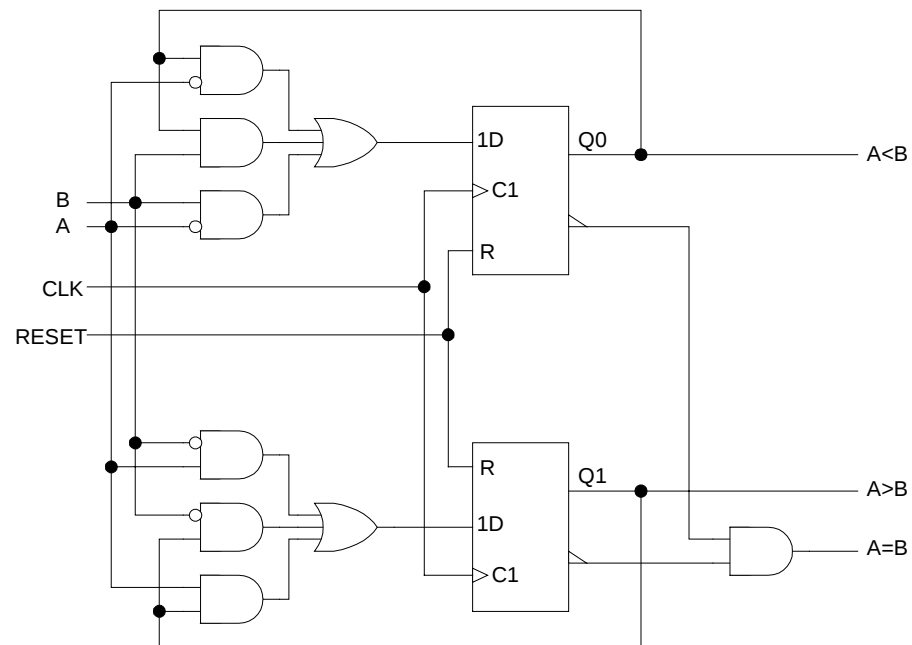
		Q ₀	
		0	1
Q ₁	0	0	0
	1	1	X

$(A > B) = Q_1$

		Q ₀	
		0	1
Q ₁	0	0	1
	1	0	X

$(A < B) = Q_0$

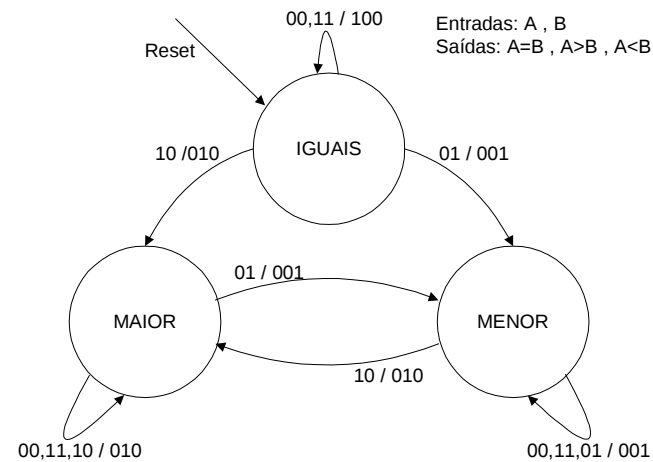
● CMP1 – LOGIGRAMA DO CIRCUITO



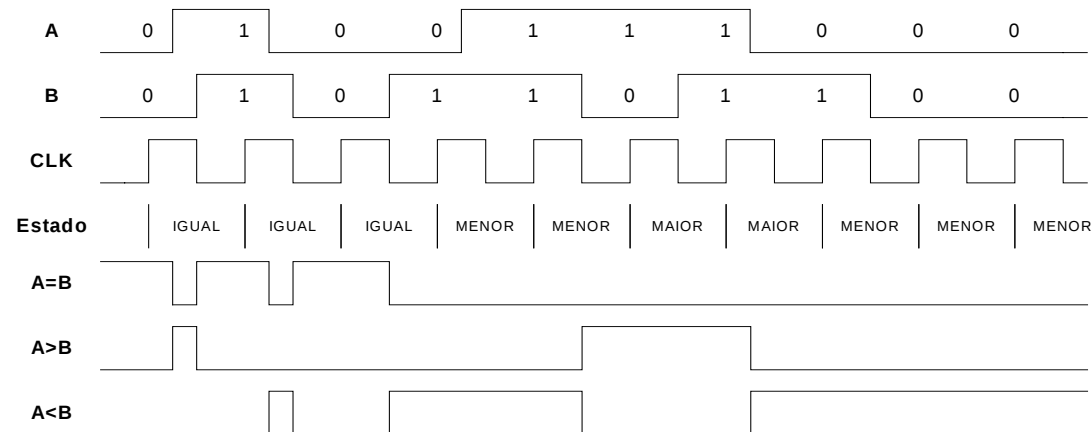
CIRCUITOS SEQUENCIAIS SÍNCRONOS - 37

● CMP2 – DIAGRAMA DE ESTADOS

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



► Exemplo



CIRCUITOS SEQUENCIAIS SÍNCRONOS - 38

● 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

CIRCUITOS SEQUENCIAIS SÍNCRONOS - 39

● CMP2 – PROJECTO DA LÓGICA COMBINATÓRIA

Q ₁ Q ₀	A B			
	00	01	11	10
00	0	0	0	1
01	0	0	0	1
11	X	X	X	X
10	1	0	1	1

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

Q ₁ Q ₀	A B			
	00	01	11	10
00	0	1	0	0
01	1	1	1	0
11	X	X	X	X
10	0	1	0	0

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

Q ₁ Q ₀	A B			
	00	01	11	10
00	1	0	1	0
01	0	0	0	0
11	X	X	X	X
10	0	0	0	0

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

Q ₁ Q ₀	A B			
	00	01	11	10
00	0	0	0	1
01	0	0	0	1
11	X	X	X	X
10	1	0	1	1

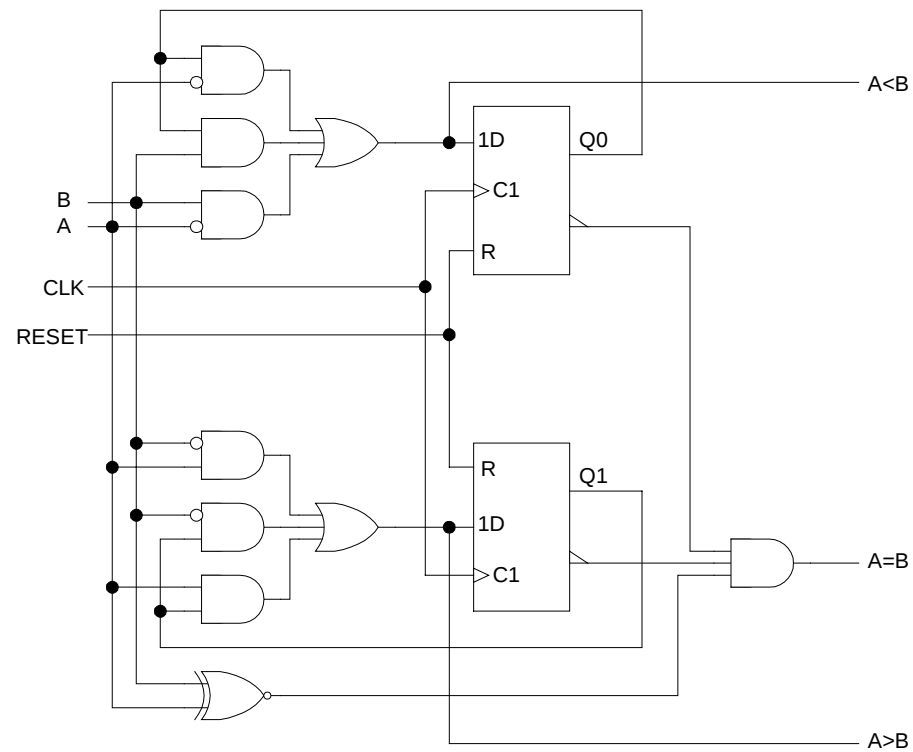
$$(A > B) = Q_1 \bar{B} + A \bar{B} + Q_1 A$$

Q ₁ Q ₀	A B			
	00	01	11	10
00	0	1	0	0
01	1	1	1	0
11	X	X	X	X
10	0	1	0	0

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

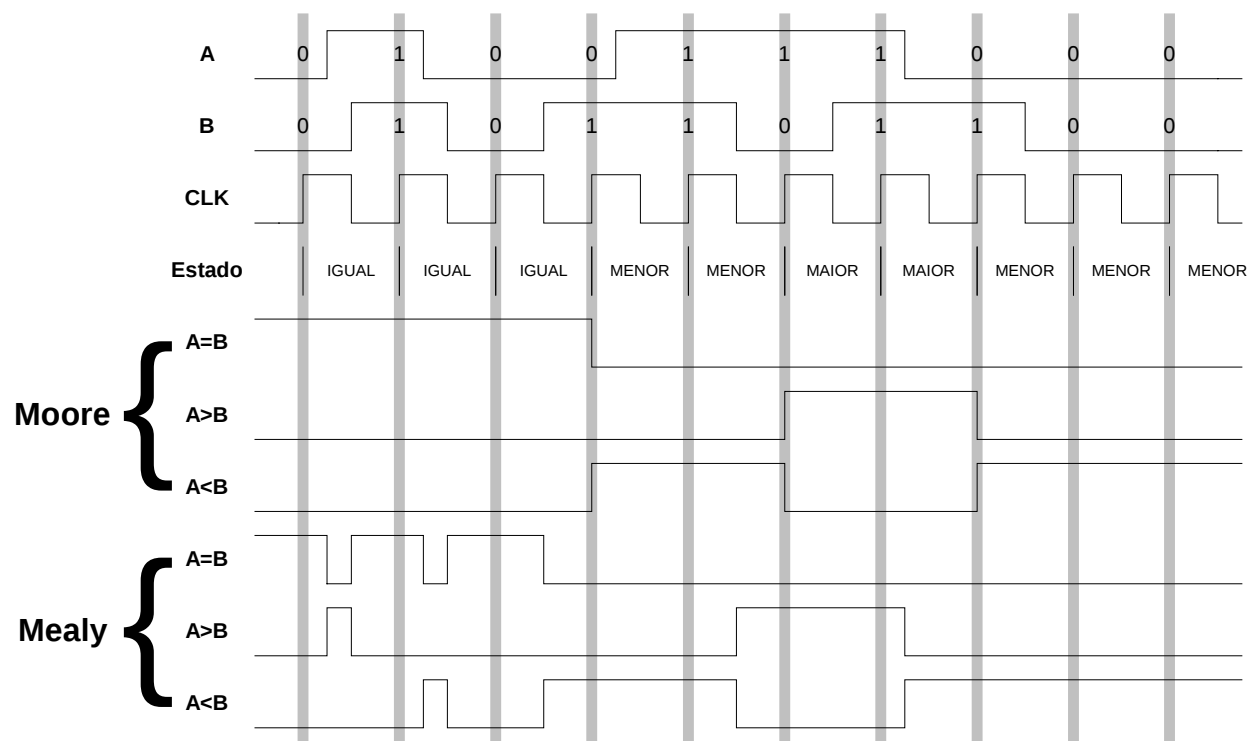
CIRCUITOS SEQUENCIAIS SÍNCRONOS - 40

● CMP2 – LOGIGRAMA DO CIRCUITO



CIRCUITOS SEQUENCIAIS SÍNCRONOS - 41

● 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, apresenta-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 / ESTADO)

Uma metodologia de projecto alternativa 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 (“*One-hot encoding*”).

Neste caso, o projecto tem as seguintes características:

- Cada estado corresponde à activação de 1 e só um FF, e.g., 0100000 (correcto) e 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

Projecto segundo o procedimento habitual (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 (Z)	
	X=0	X=1
A	A/0	B/0
B	A/0	C/0
C	A/0	C/1

		$Q_C X$			
$Q_A Q_B$		00	01	11	10
00	X	X	0	1	
01	1	0	X	X	
11	X	X	X	X	
10	1	0	X	X	

$D_A = \overline{X}$

$Q_A Q_B \backslash Q_C X$		$Q_C X$			
		00	01	11	10
$Q_A Q_B$	00	X	X	0	0
	01	0	0	X	X
	11	X	X	X	X
	10	0	1	X	X

$D_B = X \cdot Q_A$

$Q_A Q_B \backslash Q_C X$		$Q_C X$			
		00	01	11	10
00	X	X	1	0	
01	0	1	X	X	
11	X	X	X	X	
10	0	0	X	X	

$D_C = X \cdot \overline{Q_A}$

$Q_A Q_B$		$Q_C X$			
		00	01	11	10
00	X	X	1	0	
01	0	0	X	X	
11	X	X	X	X	
10	0	0	X	X	

$Z = X \cdot Q_C$

Veremos, no entanto, que o facto de 1 estado ser directamente identificado pela existência de um “1” no FF respectivo, permite uma derivação directa das equações/logigrama do circuito.

● UM FLIP-FLOP / ESTADO

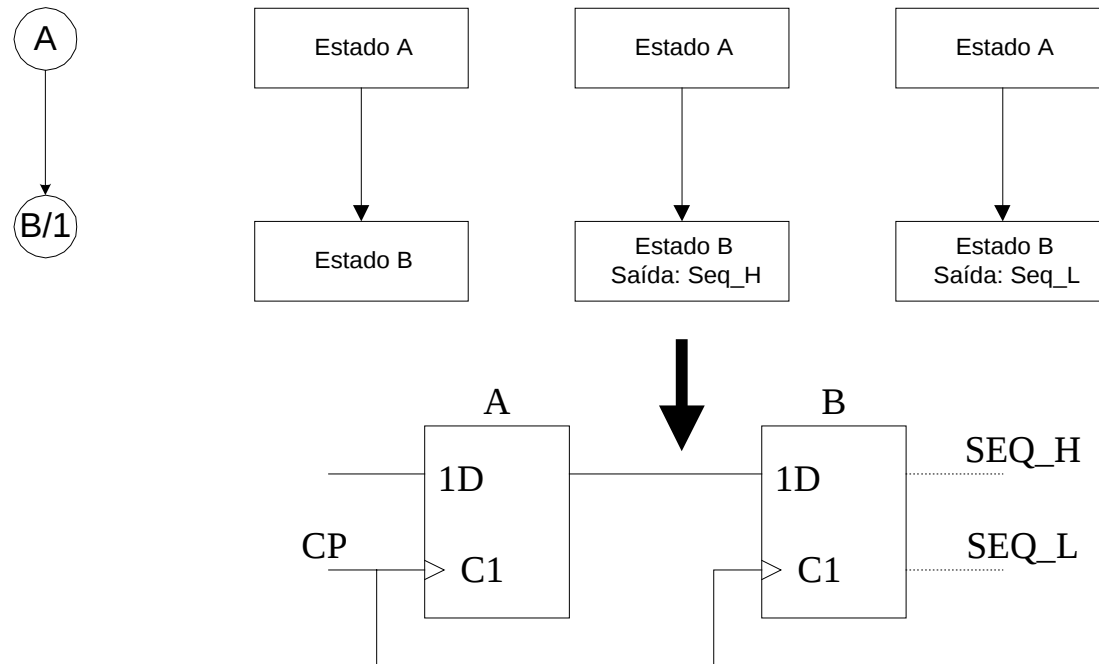
No caso de 1 FF por estado o diagrama lógico determina-se **sem** recurso a tabelas de estado ou a mapas de Karnaugh. A determinação do diagrama lógico é, assim, conseguida através da implementação directa de sub-estruturas do diagrama de estados.

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

Equações:

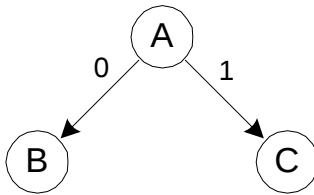
$$D_B = Q_A$$

$$SEQ_H = Q_B$$



● UM FLIP-FLOP / ESTADO (cont.)

(2) Transição **Condicional** entre Estados:

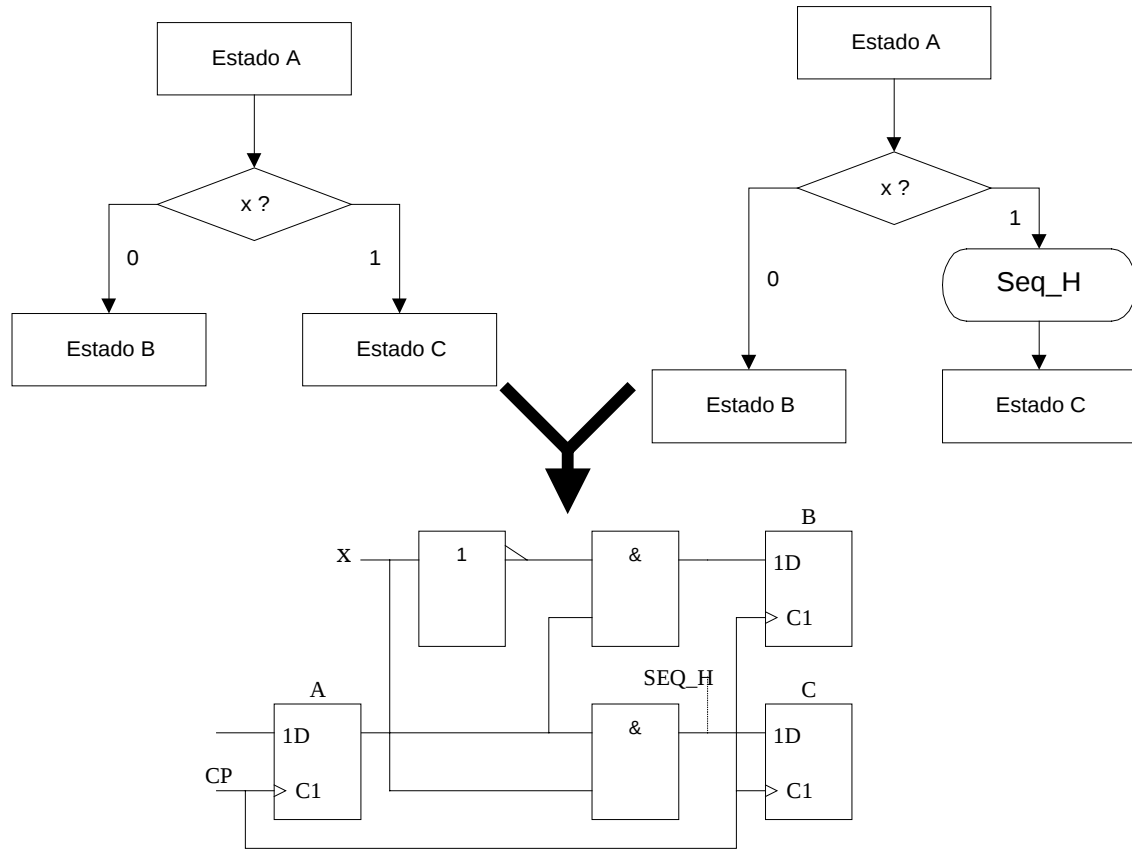


Equações:

$$D_B = \bar{X} \cdot Q_A$$

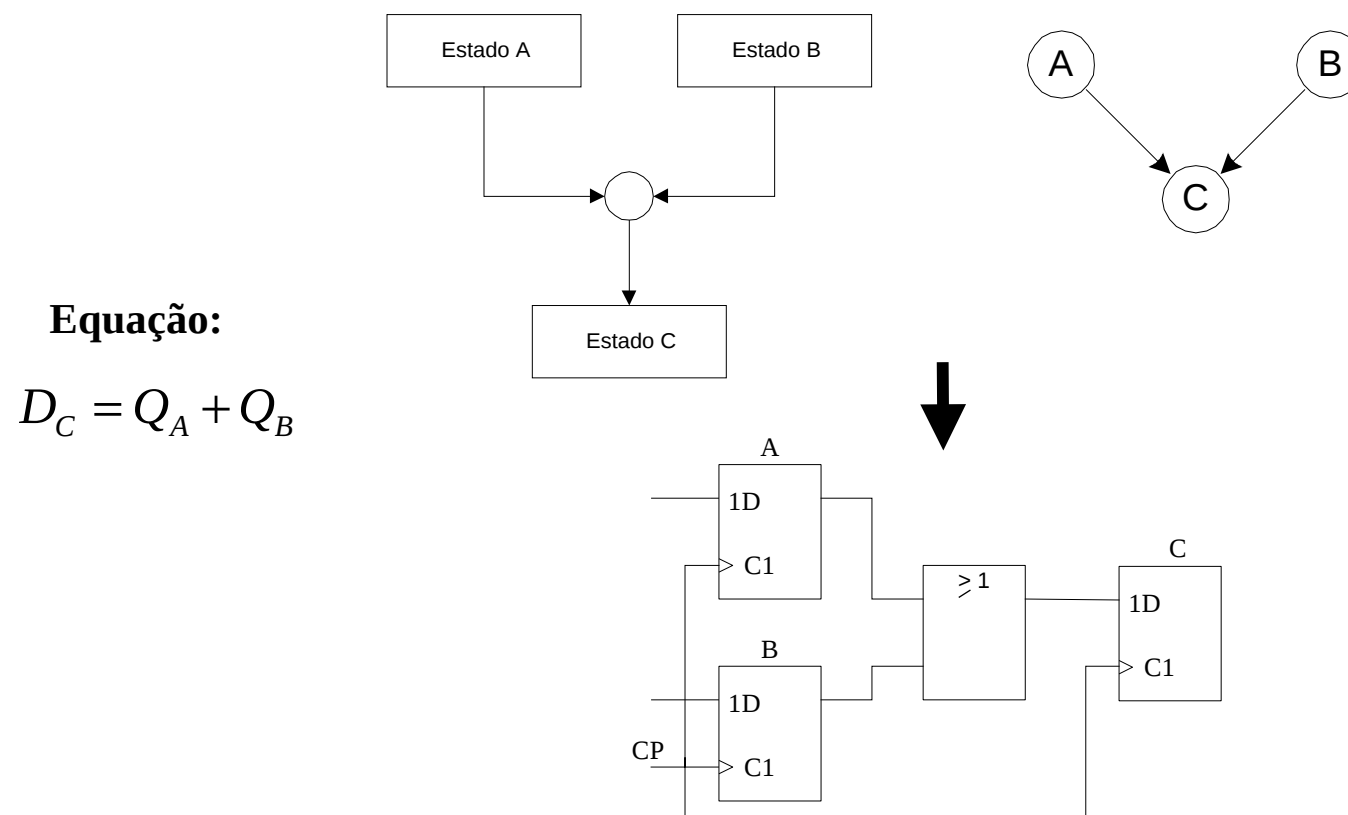
$$D_C = X \cdot Q_A$$

$$SEQ_H = X \cdot Q_A$$

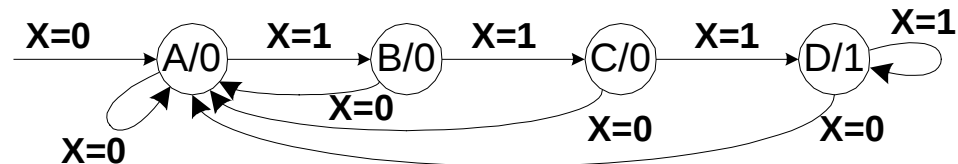


● UM FLIP-FLOP / ESTADO (cont.)

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



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



Equações:

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

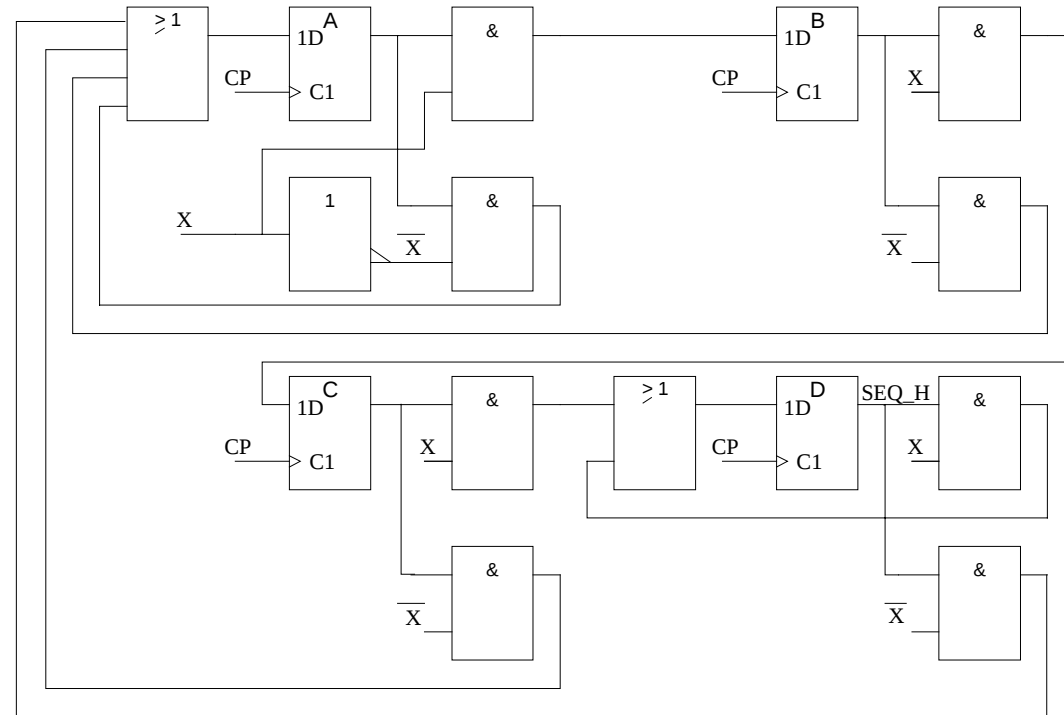
$$D_B = X \cdot Q_A$$

$$D_C = X \cdot Q_B$$

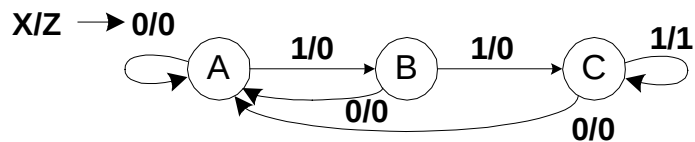
$$D_D = X \cdot Q_C + X \cdot Q_D$$

$$SEQ_H = Q_D$$

Nota: A equação de D_A pode ser trivialmente simplificada. Como?



● 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:

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

$$D_B = X \cdot Q_A$$

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

$$Z = X \cdot Q_C$$

Compare com os resultados obtidos no slide 44.

● 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 directamente para um diagrama lógico.

Nota: 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: 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.

● PROJECTOS ALTERNATIVOS – CONTADORES

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

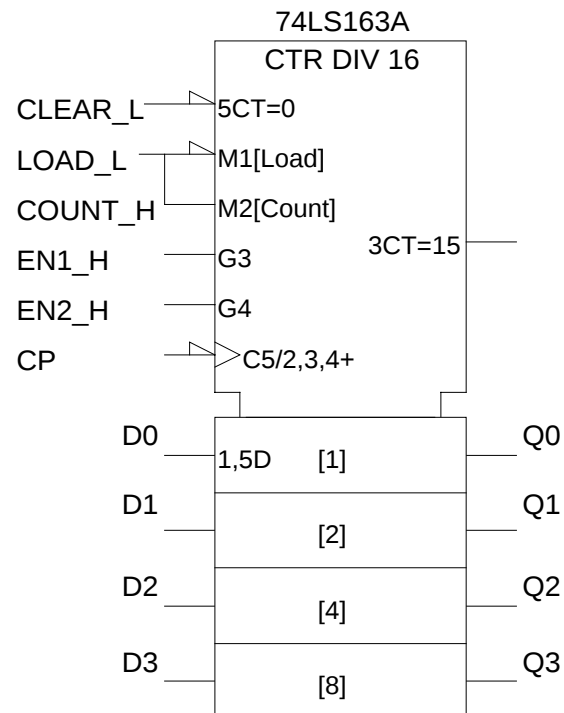
Esta solução é especialmente útil no caso de existir um elevado número de transições incondicionais entre estados, o que reduz consideravelmente a necessidade de carregamento paralelo.

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
- lógica combinatória que gera as entradas de carregamento paralelo (estado seguinte quando há quebra de sequência)

● PROJECTOS ALTERNATIVOS – CONTADORES

A título de exemplo considerar-se-á o detector de sequência referido anteriormente, como forma de comparar as anteriores soluções, e o contador integrado 74LS163A (ver secção de contadores).



● PROJECTOS ALTERNATIVOS – CONTADORES

EXEMPLO: Detector da sequência “111”

Naturalmente, este exemplo apenas requer a utilização de um contador de módulo 4 (4 estados Moore ou 3 estados Mealy).

Tendo presente a sub-utilização do contador disponível, prossegue-se com a síntese das lógica combinatória.

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 com as 3 entradas (E.A. e X) para as várias funções pretendidas (sinal de carregamento paralelo, duas entradas menos significativas do carregamento paralelo).

● 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.

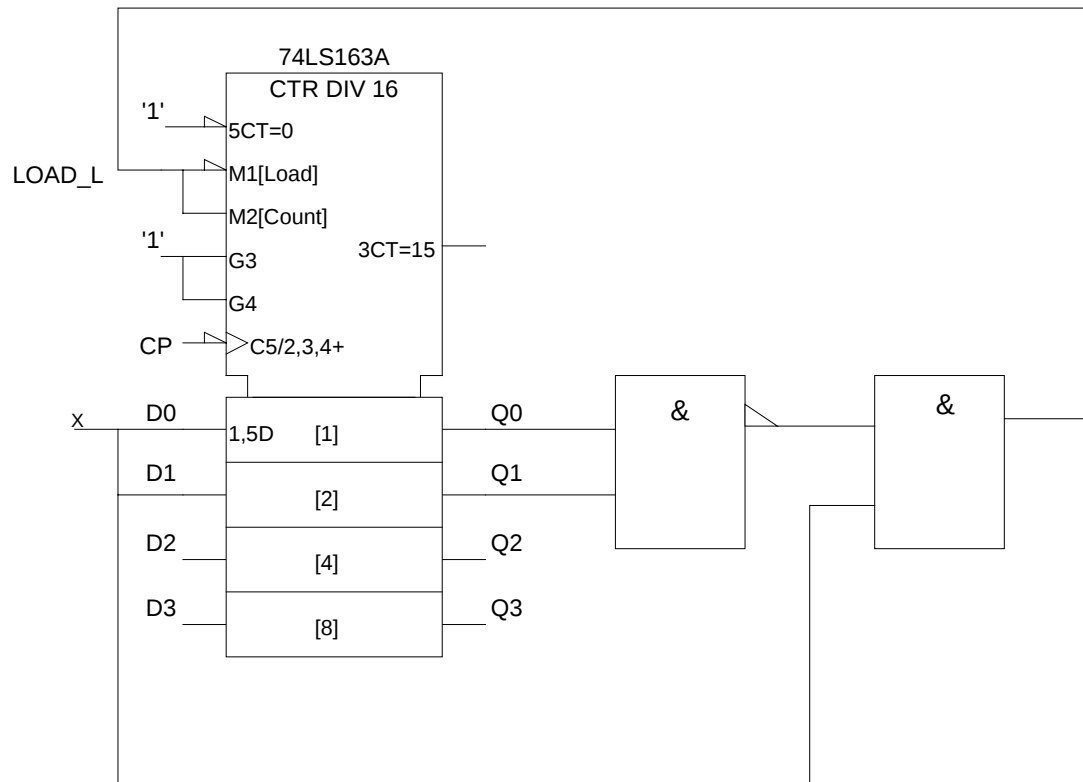
Q1Q0 x	00	10	11	01
0	0	0	0	0
1	x	x	1	x

$D1 = x$

● 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



● 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.

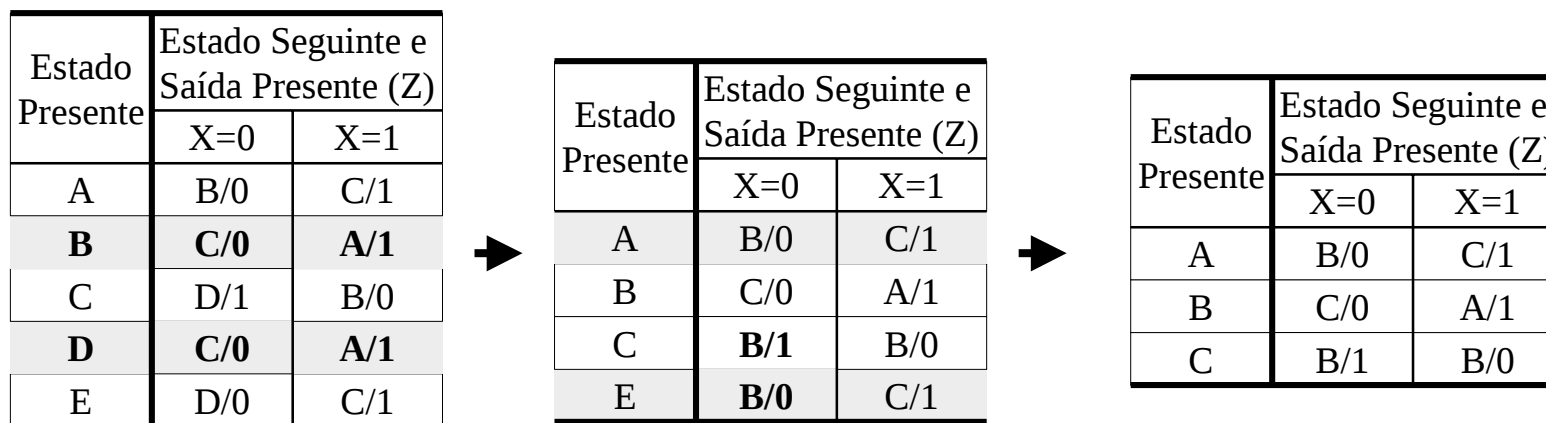
Como é óbvio, a utilização de uma representação com o número mínimo de estados tem vantagens claras tanto na síntese, como na implementação.

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



● ELIMINAÇÃO POR PARTIÇÃO

No caso anterior, a existência de estados redundante 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)			Estado Presente	Estado Seguinte e Saída Presente (Z)			Estado Presente	Estado Seguinte e Saída Presente (Z)	
	X=0	X=1			X=0	X=1			X=0	X=1
A ₁	B/0	C/0	→	A ₁	B ₁	C ₁	→	A ₁	B ₁	C ₁
B ₁	D/0	E/0		B ₁	D ₁	E ₁		B ₁	D ₃	E ₁
C ₁	G/0	E/0		C ₁	G ₁	E ₁		C ₁	G ₁	E ₁
D ₁	H/0	F/0		D ₁	H ₁	F ₂		D ₃	H ₁	F ₂
E ₁	G/0	A/0		E ₁	G ₁	A ₁		E ₁	G ₁	A ₁
F ₁	G/1	A/0		F ₂	G ₁	A ₁		F ₂	G ₁	A ₁
G ₁	D/0	C/0		G ₁	D ₁	C ₁		G ₁	D ₃	C ₁
H ₁	H/0	A/0		H ₁	H ₁	A ₁		H ₁	H ₁	A ₁

● ELIMINAÇÃO POR PARTIÇÃO (cont.)

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

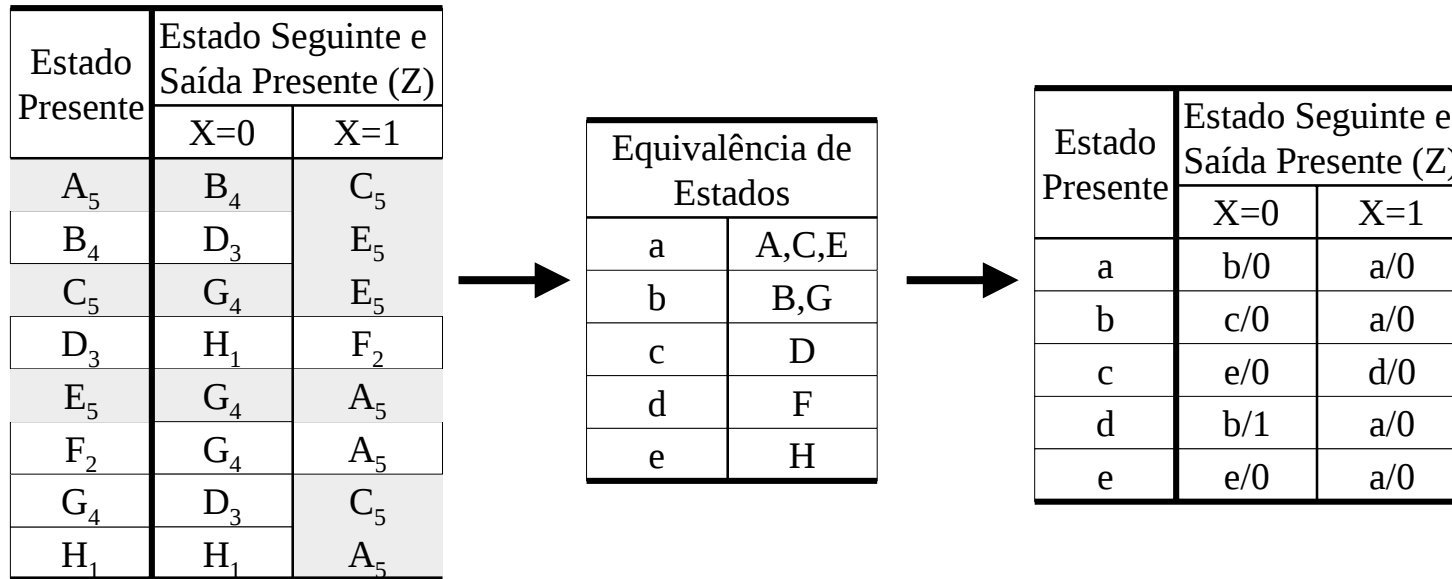
→

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

→

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

● ELIMINAÇÃO POR PARTIÇÃO (cont.)



BIBLIOGRAFIA

[1] M. Morris Mano, Charles R. Kime, “Logic and Computer Design Fundamentals”, Prentice-Hall International, Inc. (Capítulo 4, Secções 4.4 a 4.7)