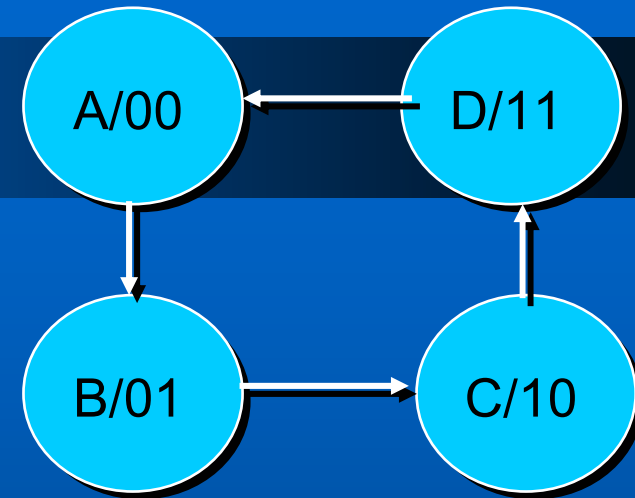
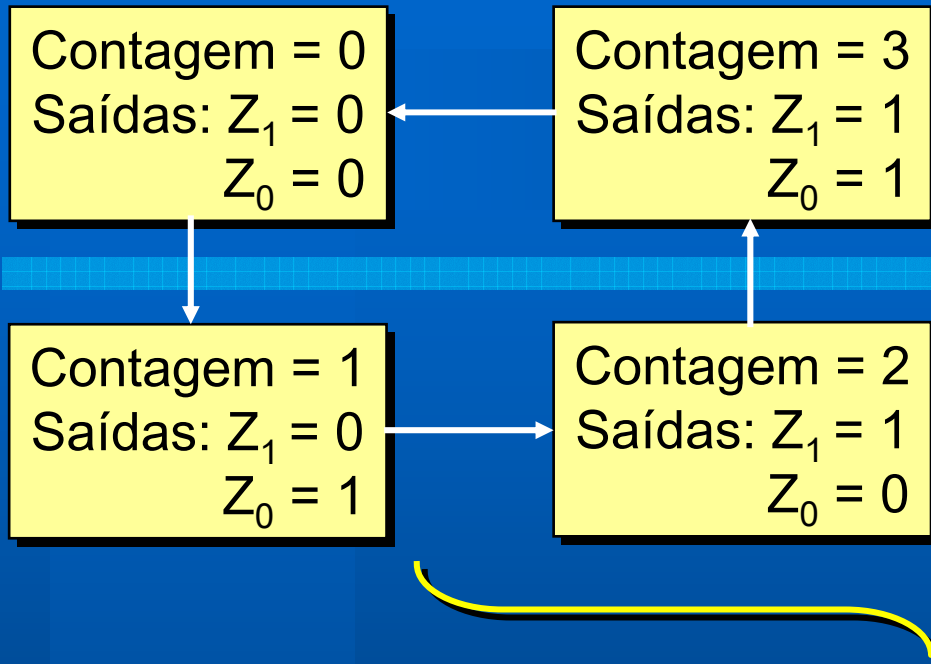


# SÍNTESE DE SIST. SEQUENCIAIS SÍNCRONOS

**Sel 414 - Sistemas Digitais**

**Prof. Homero Schiabel**

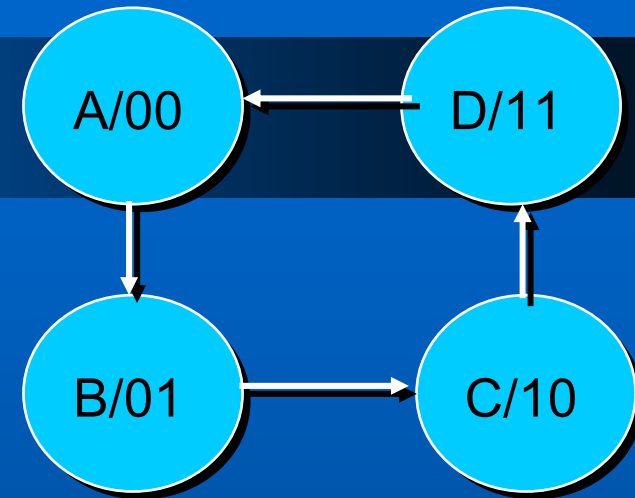
*Síntese Sist. Síncronos*



Estado Presente	Saída Presente ( $Z_1Z_0$ )	Estado Futuro
A	00	B (01)
B	01	C (10)
C	10	D (11)
D	11	A (00)

**Tabela de Estado**

*Como construir esse contador, considerando os diversos tipos de FF?*



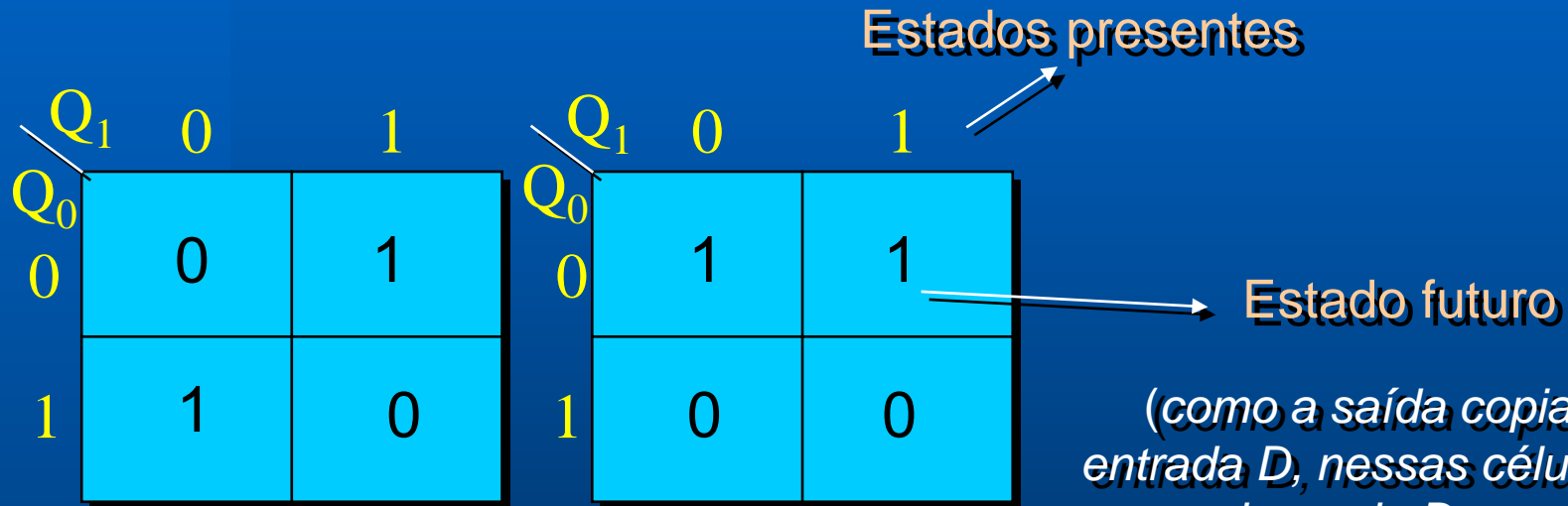
Estado Presente	Saída Presente ( $Z_1Z_0$ )	Estado Futuro
A	00	B (01)
B	01	C (10)
C	10	D (11)
D	11	A (00)

Tabela de Estado

## Escolha do FF a usar

- Suponhamos FF tipo **D**

- 4 estados  $\rightarrow$  2 FF
- não tem entradas externas



(como a saída copiará a entrada  $D$ , nessas células vão os valores de  $D$  que serão copiados para produzir os est. futuros de  $Q$ )

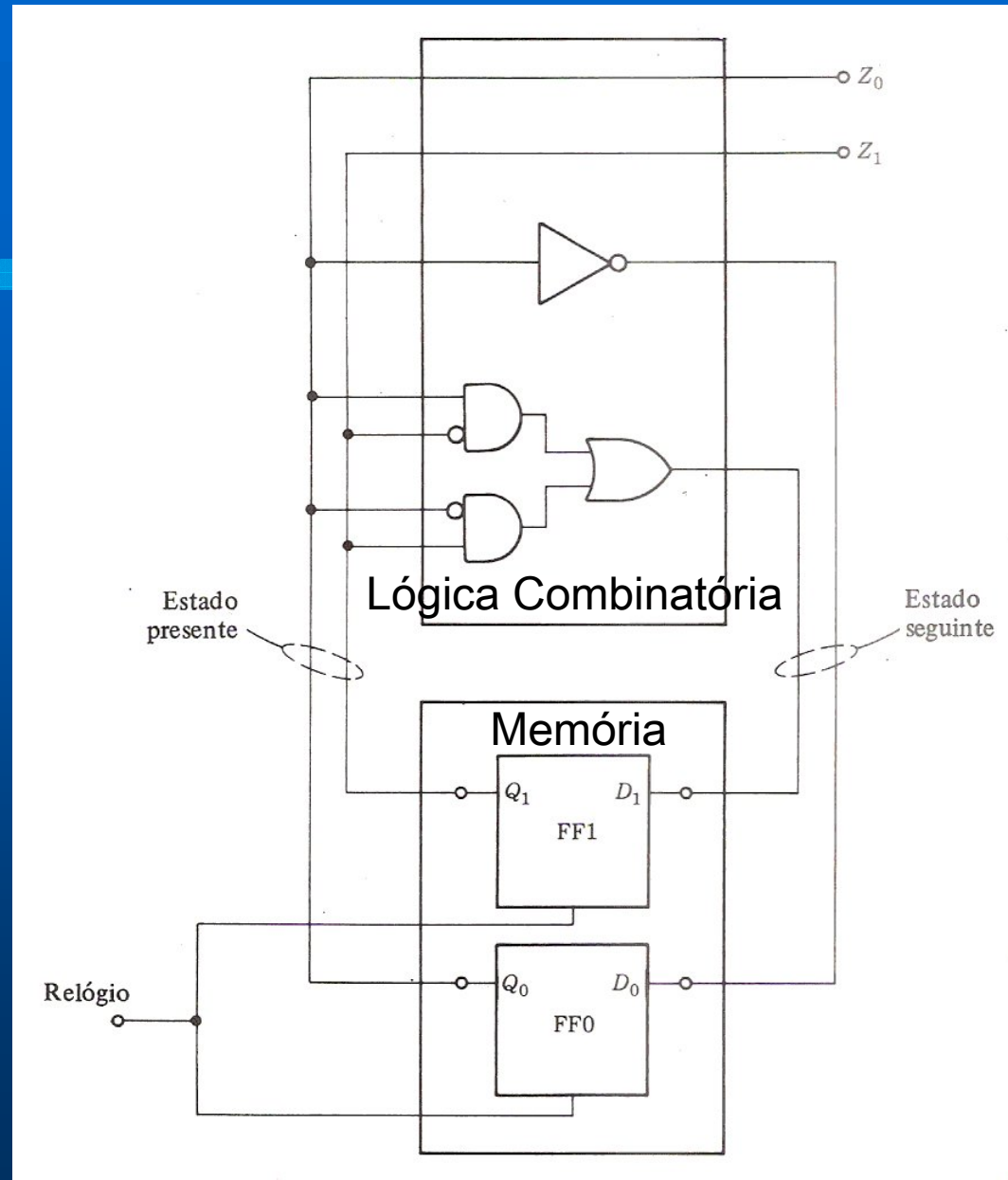
$D_1$

$$D_1 = Q_1 \oplus Q_0$$

$D_0$

$$D_0 = \overline{Q_0}$$

Eq. de excitação



Sintetizar um Circuito Sequencial significa obter o seu Diagrama Lógico a partir de uma descrição do problema, ou a partir de seu Diagrama de Estado.

## Procedimentos para a Síntese:

### Etapa 1

A partir de uma descrição do problema, derivar a Tabela de Estado.

### Etapa 2

Usar técnicas de redução de estados para encontrar a Tabela de Estado de um circuito equivalente de Mínimo-Estado.

### Etapa 3

Escolher a Atribuição de Estados e gerar as Tabelas de Transição de Saída.

## Procedimentos para a Síntese:

### Etapa 4

Determinar os dispositivos de memória ou FF a serem usados e encontrar seus Mapas de Excitação.

### Etapa 5

Dos Mapas de Excitação, produzir as Equações Lógicas de Chaveamento. Gerar os Mapas de Saída e determinar as Equações Lógicas de Saída.

### Etapa 6

Desenhar o Diagrama Lógico do Circuito Sequencial usando as Equações Lógicas e os dispositivos de memória escolhidos.

## Tabelas de Entrada e Equações Características dos principais Flip-Flops:

Transição de Estados		Entradas
Q	Q*	D
0	0	0
0	1	1
1	0	0
1	1	1

$$Q^* = D$$

Transição de Estados		Entradas	
Q	Q*	S	R
0	0	0	d
0	1	1	0
1	0	0	1
1	1	d	0

$$Q^* = S + \bar{R}.Q$$

Transição de Estados		Entradas
Q	Q*	T
0	0	0
0	1	1
1	0	1
1	1	0

$$Q^* = T.\bar{Q} + \bar{T}.Q$$

Transição de Estados		Entradas	
Q	Q*	J	K
0	0	0	d
0	1	1	d
1	0	d	1
1	1	d	0

$$Q^* = \bar{K}.Q + J.\bar{Q}$$



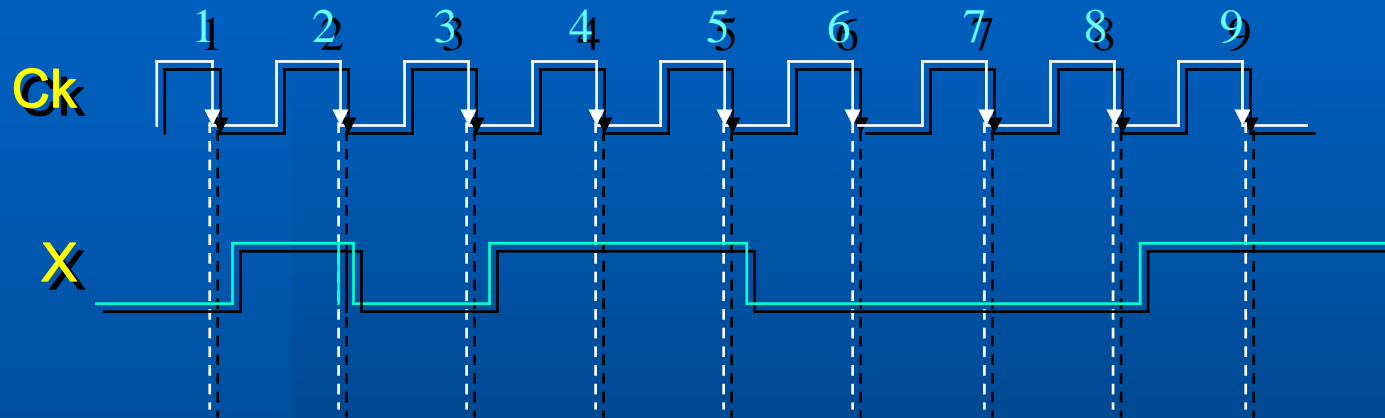
# Exemplo 1:

Detector de seqüências:



Modelo **MOORE**

Suponhamos a entrada  $X = 010110001$ \*



*\* Se não houvesse pulsos do Ck como controle, leríamos  $X = 010101$ , pois não seria levado em conta o tempo de duração do nível lógico*

**PROPOSIÇÃO:**  $Z = 1$  somente quando  $X = 1$  por três ou mais ciclos consecutivos do Ck

<b>Ck =</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>12</b>	<b>13</b>	<b>14</b>
<b>X =</b>	0	1	1	0	1	1	1	0	1	1	1	1	1	-
<b>Z = ?</b>	0	0	0	0	0	0	0	1	0	0	0	1	1	1

## Exemplo 1:

Detector de sequências:



Modelo **MOORE**

- ➔ Não há conexão direta entre  $X$  e  $Z$
- ➔ Se o sistema está num estado fixo – como durante o ciclo do  $CK$  – variações em  $X$  não afetam a saída  $Z$

I. PROPOSIÇÃO:  $Z = 1$  somente quando  $X = 1$  por três ou mais ciclos consecutivos do  $Ck$

$Ck = 1$	2	3	4	5	6	7	8	9	10	11	12	13	14
$X = 0$	1	1	0	1	1	1	0	1	1	1	1	1	-
$Z = ?$	0	0	0	0	0	0	1	0	0	0	1	1	1

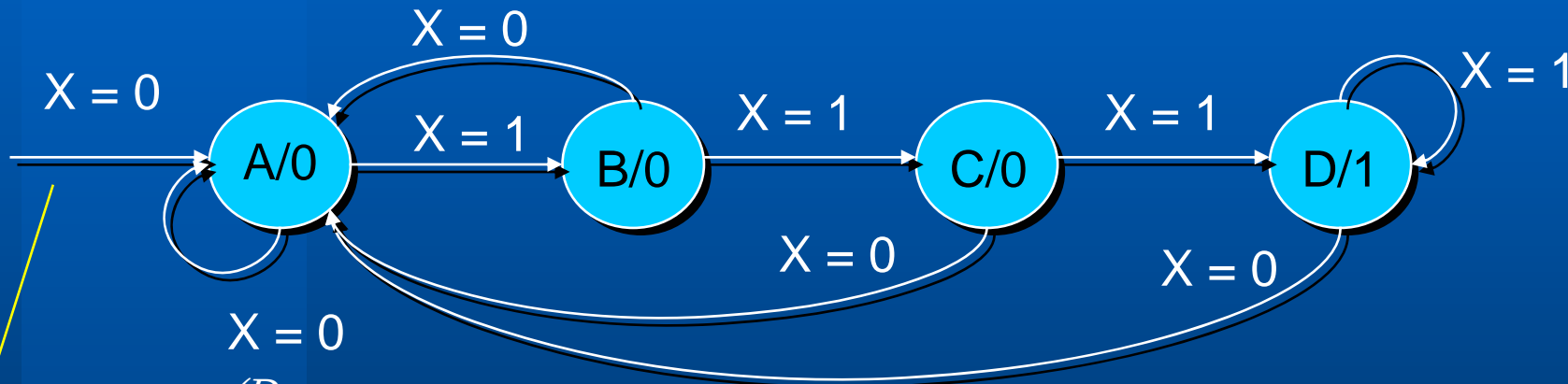
# Exemplo 1:

Detector de sequências:



Modelo **MOORE**

## II. Diagrama de estados



$X = 0$

*(Deve armazenar em A a informação – que não mudou)*

*Se  $X=1$ , Z no estado A seria desconhecido, pois seria necessário conhecer X nos dois ciclos anteriores ( $X=1 \rightarrow Z=1$ , caso contrário,  $Z=0$ ). Portanto, essa definição do estado A aqui é arbitrária.*

# Exemplo 1:

Detector de sequências:



Modelo **MOORE**

## III. Escolha do FF ➔ usando FF JK

$Q^n \rightarrow Q^{n+1}$	J	K
0 → 0	0	d
0 → 1	1	d
1 → 0	d	1
1 → 1	d	0

TABELA DE ESTADOS a partir da atribuição de estado

Est. Presente ( $Q_1Q_0$ )	Saída Presente (Z)	Est. Futuro	
		X = 0	X = 1
A (= 00)	0	A (=00)	B (=01)
B (= 01)	0	A (=00)	C (=10)
C (= 10)	0	A (=00)	D (=11)
D (= 11)	1	A (=00)	D (=11)

8 combinações

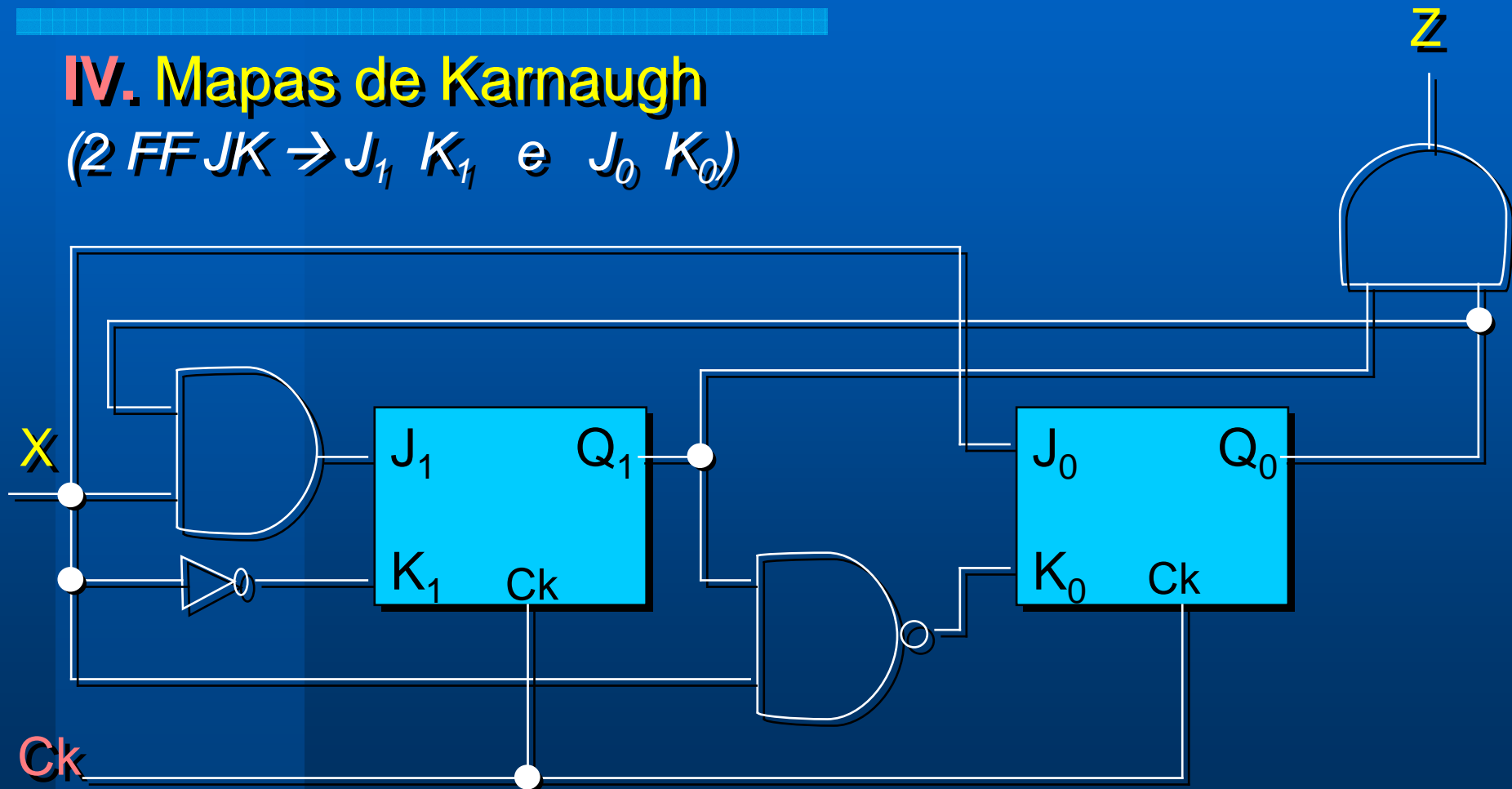
# Exemplo 1:

Detector de sequências:



Modelo **MOORE**

**IV. Mapas de Karnaugh**  
(2 FF JK  $\rightarrow$   $J_1, K_1$  e  $J_0, K_0$ )



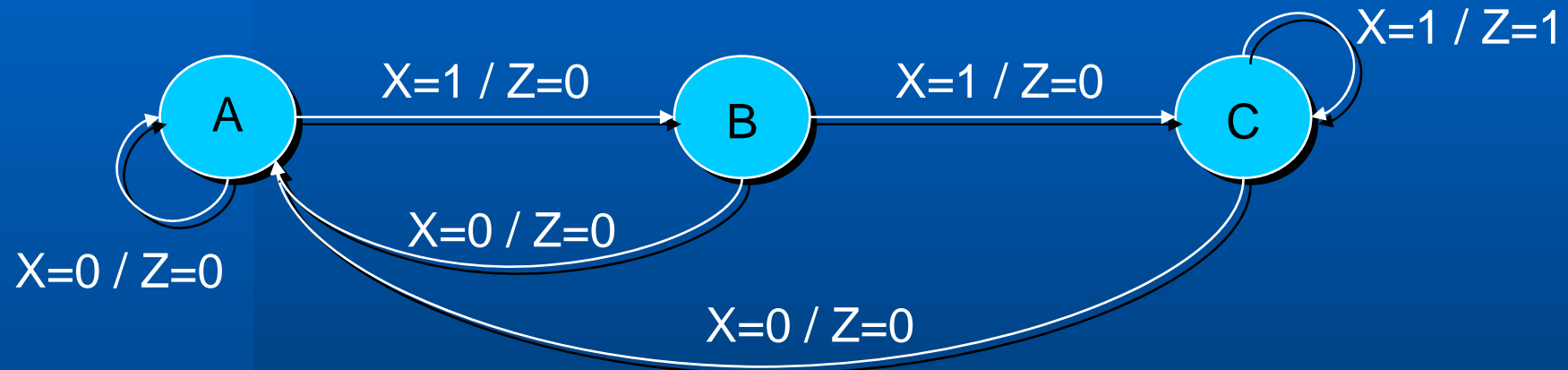
## Exemplo 2:

Detector de sequências:



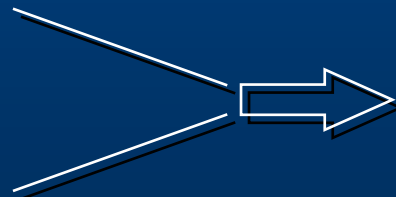
Modelo **MEALY**

### I. Diagrama de estados



### II. Atribuição de estados

A = 00  
B = 01  
C = 10



*Ainda são necessários  
2 FF (3 estados)*

## Exemplo 2:

Detector de sequências:



Modelo **MEALY**

### III. TABELA DE ESTADOS a partir da atribuição de estado

Est. Presente ( $Q_1Q_0$ )	Est. Futuro / Saída Presente	
	X = 0	X = 1
A (= 00)	A / 0	B / 0
B (= 01)	A / 0	C / 0
C (= 10)	A / 0	C / 1

# Exemplo 1:

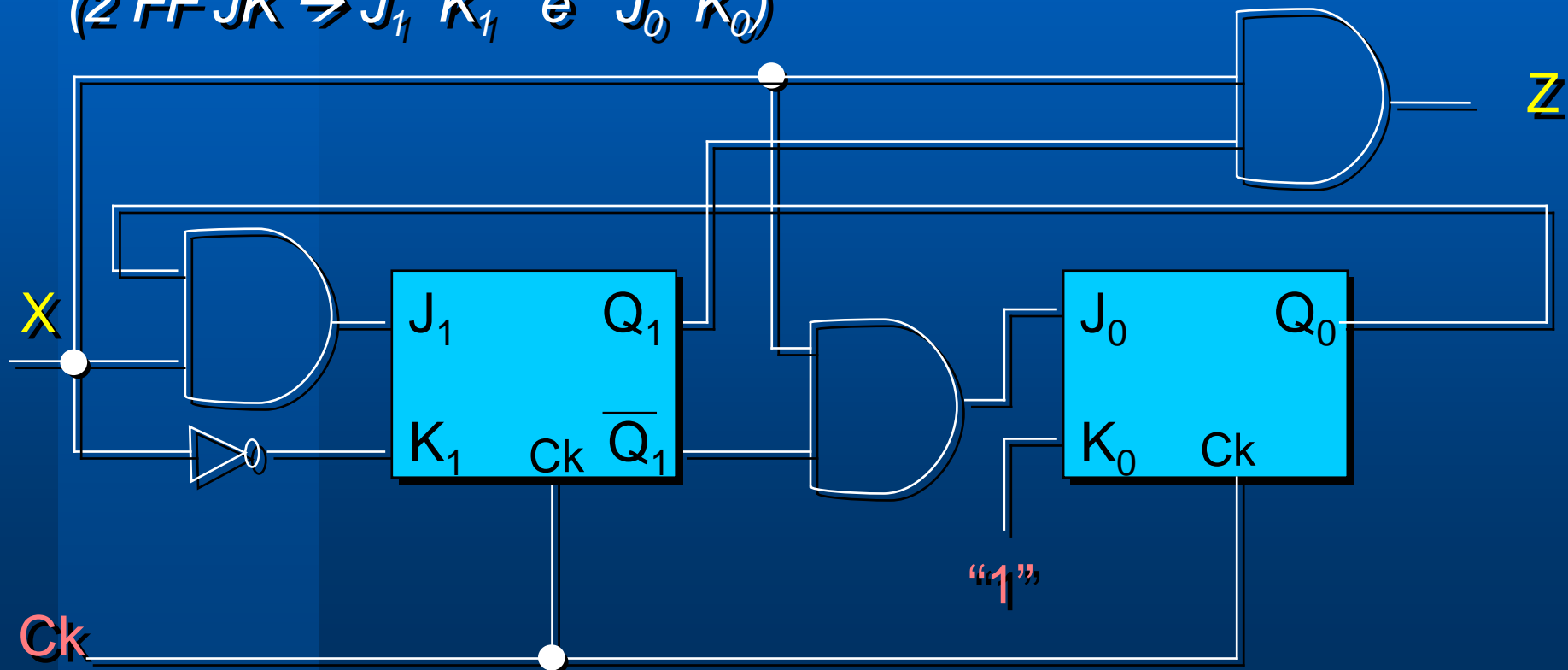
Detector de sequências:



Modelo **MEALY**

## IV. Mapas de Karnaugh

(2 FF JK  $\rightarrow$   $J_1, K_1$  e  $J_0, K_0$ )





## Exemplo 3:

Sintetizar o Circuito Sequencial Síncrono dado pela Tabela de Estado.

A) Com FF tipo D.

B) Com FF JK.

### Tabela de Estado

	x	
	0	1
A	A/0	B/0
B	A/0	C/1
C	B/0	D/0
D	C/1	D/0

### Atribuição de Estados

Estados	y <sub>1</sub>	y <sub>2</sub>
A	0	0
B	0	1
C	1	1
D	1	0

y <sub>1</sub> y <sub>2</sub>	x	
	0	1
00	00/0	01/0
01	00/0	11/1
11	01/0	10/0
10	11/1	10/0

Tabela de Transição ou Tabela de Estado Binária.

$Y_1 Y_2 / z$

# A) Com Flip-flops tipo D

y <sub>1</sub> y <sub>2</sub>	x	
	0	1
00	00/0	01/0
01	00/0	11/1
11	01/0	10/0
10	11/1	10/0

**y<sub>1</sub>y<sub>2</sub>/z**

**K-maps:**

y <sub>1</sub> y <sub>2</sub>	x	
	0	1
00	0	0
01	0	1
11	0	0
10	1	0

**z**

**D<sub>1</sub> = y<sub>1</sub>**

y <sub>1</sub> y <sub>2</sub>	x	
	0	1
00	0	0
01	0	1
11	0	1
10	1	1

**y<sub>1</sub>**

**D<sub>2</sub> = y<sub>2</sub>**

y <sub>1</sub> y <sub>2</sub>	x	
	0	1
00	0	1
01	0	1
11	1	0
10	1	0

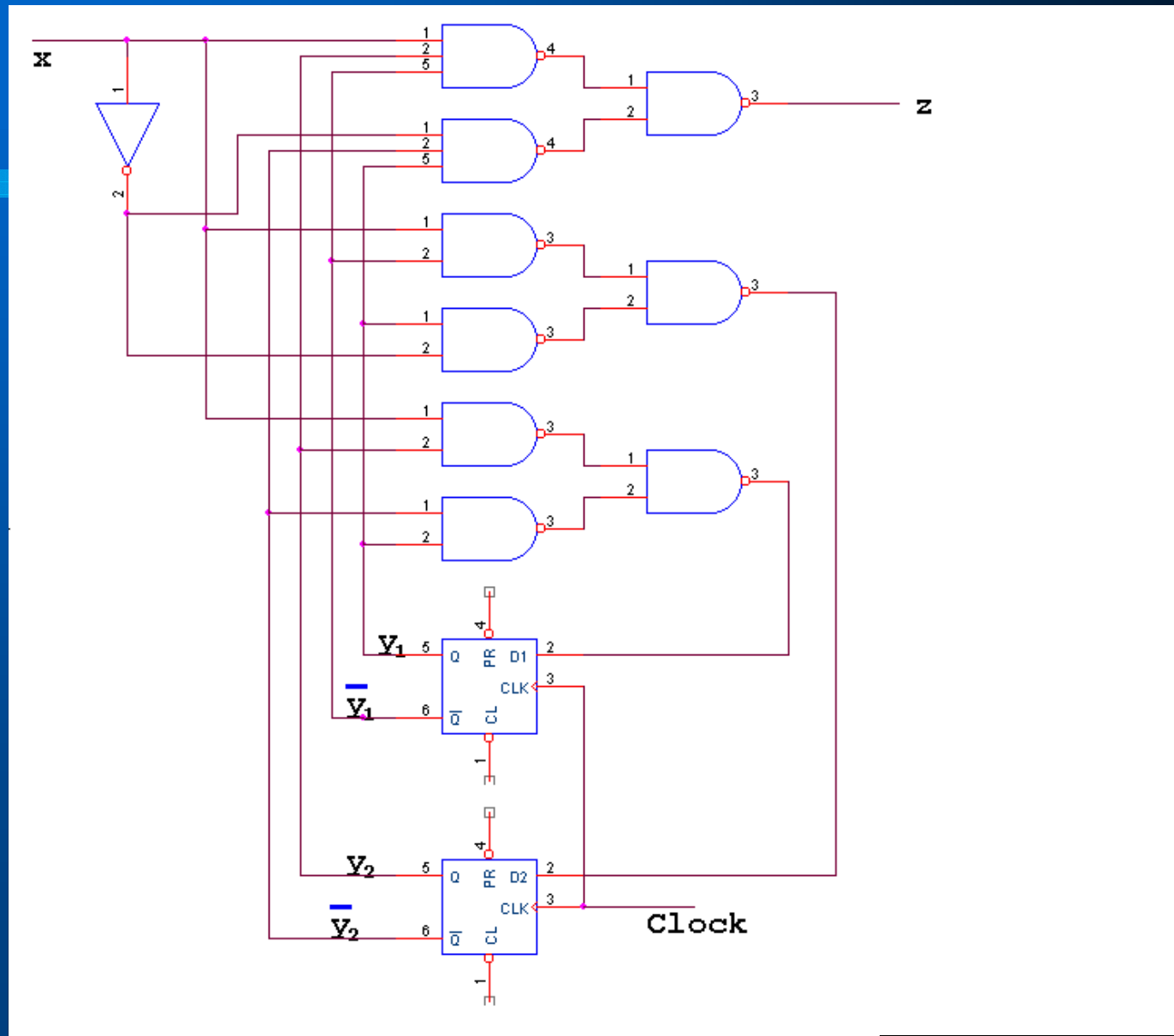
**y<sub>2</sub>**

$$z = x \cdot \bar{y}_1 \cdot y_2 + \bar{x} \cdot y_1 \cdot \bar{y}_2$$

$$D_1 = y_1 \cdot \bar{y}_2 + x \cdot y_2$$

$$D_2 = \bar{x} \cdot y_1 + x \cdot \bar{y}_1 = x \oplus y_1$$

# Diagrama Lógico:



## B) Com Flip-flops JK

Transição de Estados		Entradas	
Q	Q*	J	K
0	0	0	d
0	1	1	d
1	0	d	1
1	1	d	0

Q\* = 0

Q = 1

y <sub>1</sub> y <sub>2</sub>	x	
	0	1
00	00/0	01/0
01	00/0	11/1
11	01/0	10/0
10	11/1	10/0

Y<sub>1</sub>Y<sub>2</sub>/z

y <sub>1</sub> y <sub>2</sub>	x	
	0	1
00	0d	0d
01	0d	1d
11	d1	d0
10	d0	d0

J<sub>1</sub>K<sub>1</sub>

y <sub>1</sub> y <sub>2</sub>	x	
	0	1
00	0d	1d
01	d1	d0
11	d0	d1
10	1d	0d

J<sub>2</sub>K<sub>2</sub>

# K-maps:

y <sub>1</sub> y <sub>2</sub>	x	
	0	1
00	0	0
01	0	1
11	d	d
10	d	d

J<sub>1</sub>

$$J_1 = x \cdot y_2$$

y <sub>1</sub> y <sub>2</sub>	x	
	0	1
00	d	d
01	d	d
11	1	0
10	0	0

K<sub>1</sub>

$$K_1 = \overline{x} \cdot y_2$$

y <sub>1</sub> y <sub>2</sub>	x	
	0	1
00	0	1
01	d	d
11	d	d
10	1	0

J<sub>2</sub>

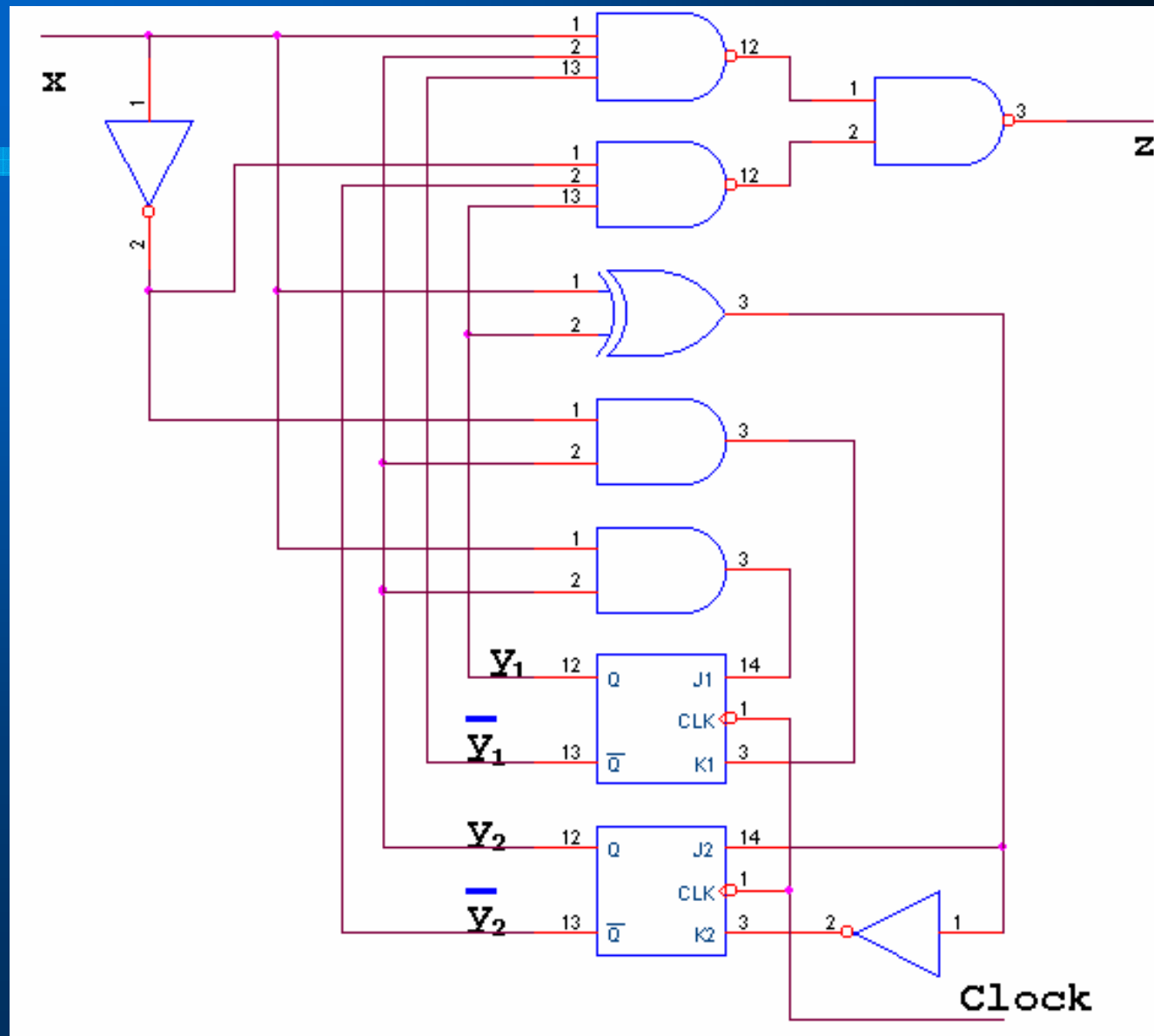
$$J_2 = x \oplus y_1$$

y <sub>1</sub> y <sub>2</sub>	x	
	0	1
00	d	d
01	1	0
11	0	1
10	d	d

K<sub>2</sub>

$$K_2 = \overline{x \oplus y_1} = \overline{J_2}$$

# Diagrama Lógico:



## Exemplo 4:

Projetar um Circuito Sequencial Síncrono que reconheça uma sequência de entrada consistindo de exatamente dois zeros seguidos de 10 (permitindo superposição).

$x = 001001000010010$

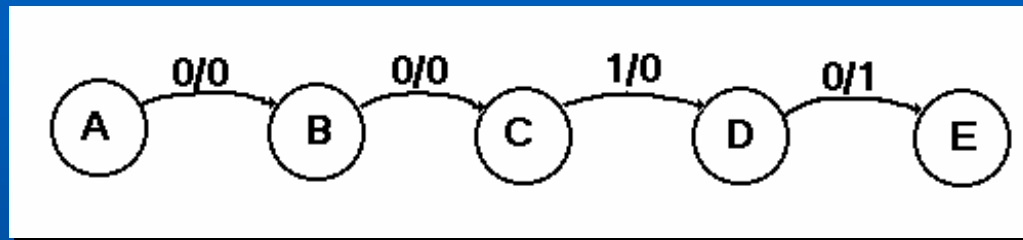
$z = 000100100000001$

The diagram illustrates the mapping from the input sequence  $x = 001001000010010$  to the output sequence  $z = 000100100000001$ . Brackets and arrows show the following connections:

- A red bracket under the first two '0's of  $x$  points to the first '0' of  $z$ .
- A blue bracket under the '1' and the following '0' of  $x$  points to the second '0' of  $z$ .
- A purple bracket under the last two '0's of  $x$  points to the final '1' of  $z$ .

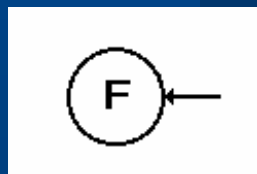
## I. Gerando o Diagrama de Estado:

Considerando **A** o Estado Inicial, a sequência reconhecida deverá ter o seguinte fluxo:

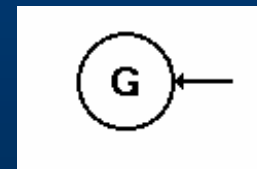


Se algum erro acontecer, ou seja, se alguma entrada diferente da esperada ocorrer, o circuito deverá ir para um estado de erro.

Assim, atribuímos dois novos estados de erro:



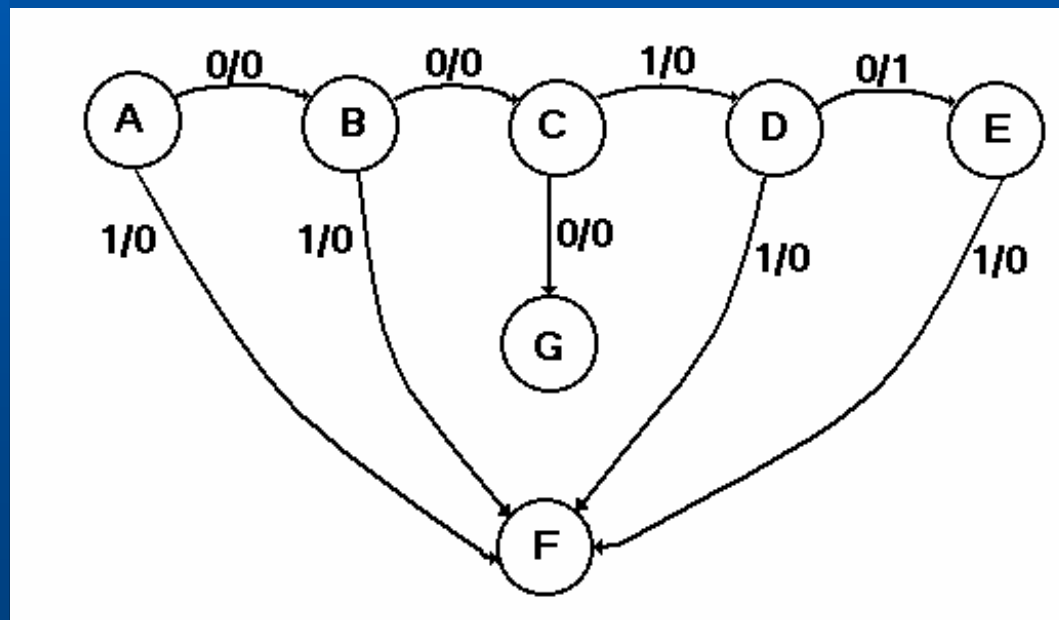
Entrada incorreta  
 $x=1$



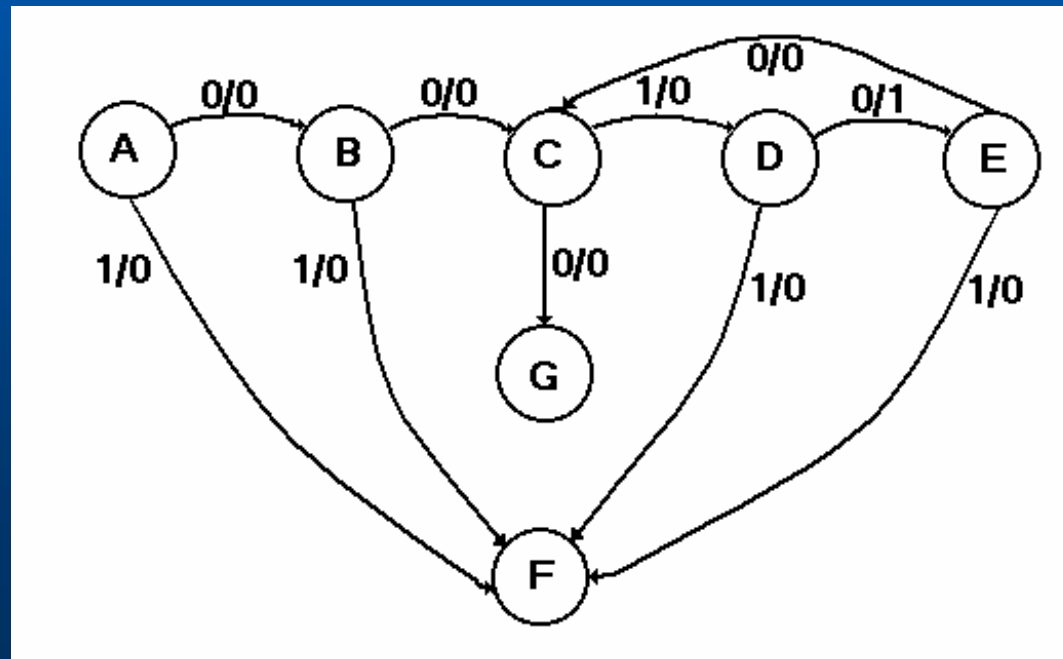
Entrada incorreta  
 $x=0$



Se ocorrer qualquer entrada diferente da sequência a ser reconhecida, o circuito deverá ir para um dos estados de erro.



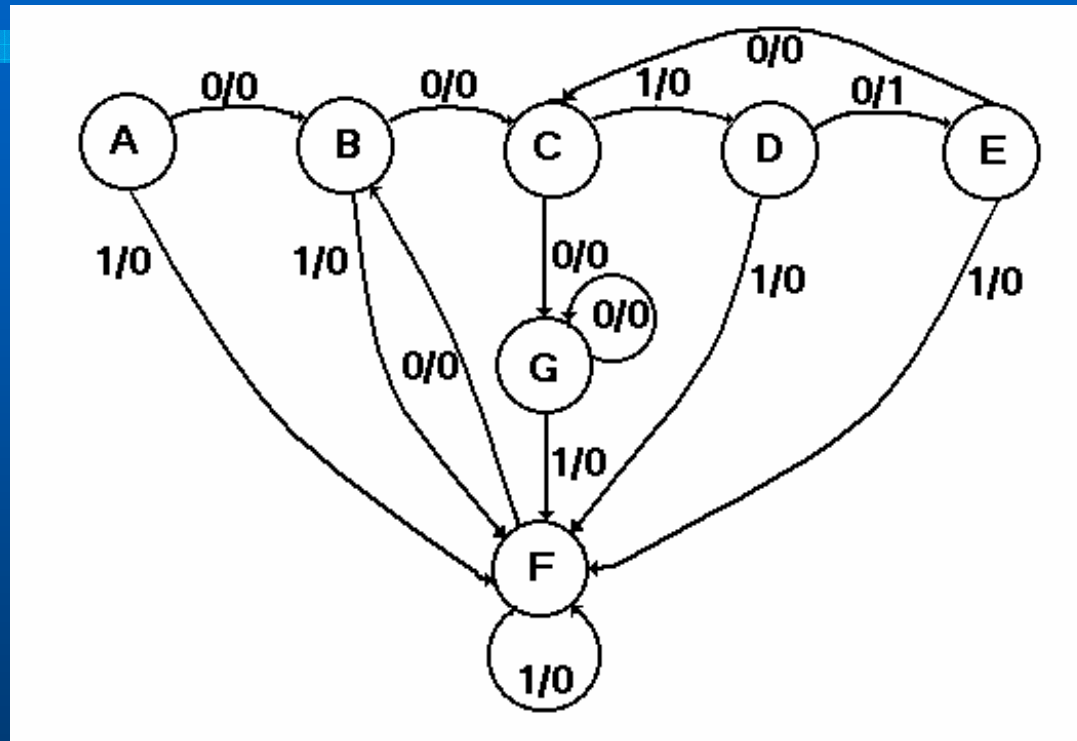
Se, no último estado da sequência correta (E), ocorrer uma entrada 0, o circuito deverá ir para o estado C, permitindo a superposição.



x = 0010010

z = 0001001

Completando as transições para os Estados de Erro, tem-se o Diagrama de Estados abaixo:



Note que, se o circuito está no estado **F** e recebe uma entrada **x=0**, o mesmo deve voltar ao estado **B**, pois pode ser início de uma sequência válida, após **x=1** (chegada em **F** de qualquer estado).

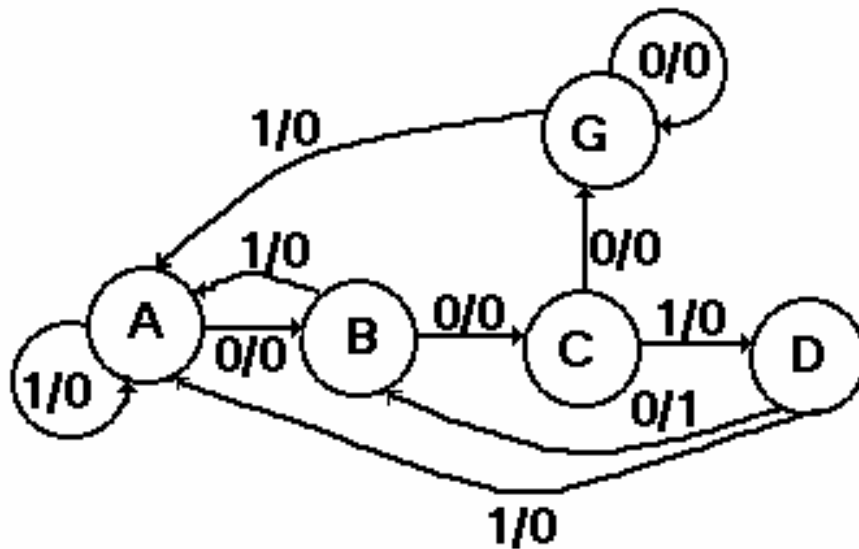
## II. Tabela de Estados:

	x	
	0	1
A	B/0	F/0
B	C/0	F/0
C	G/0	D/0
D	E/1	F/0
E	C/0	F/0
F	B/0	F/0
G	G/0	F/0

Os Estados *A* e *F*, bem como os Estados *B* e *E* são iguais, ou equivalentes. Podemos então, eliminar as linhas *E* e *F*.

## Tabela de Estados Reduzida:

	x	
	0	1
A	B/0	A/0
B	C/0	A/0
C	G/0	D/0
D	B/1	A/0
G	G/0	A/0



## Diagrama de Estados Reduzido

**Exemplo 5:** Projetar um Somador Serial Síncrono Binário, que compute a soma de dois números binários de  $n$  bits, 1 bit por vez, começando pelo LSB.

Supondo:

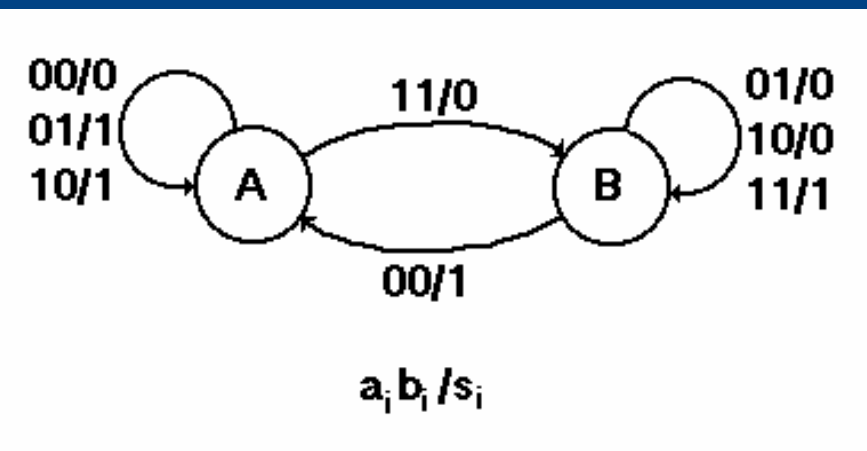
Estado A  $\Rightarrow$  Carry ( $c$ ) = 0

Estado B  $\Rightarrow$  Carry ( $c$ ) = 1

A soma entre cada dígito binário deverá ser:

$$s_i = a_i + b_i + c_i$$

**Diagrama de Estado:**



## Tabela de Estado:

	$a_i b_i$			
$y^k$	00	01	11	10
A	A/0	A/1	B/0	A/1
B	A/1	B/0	B/1	B/0

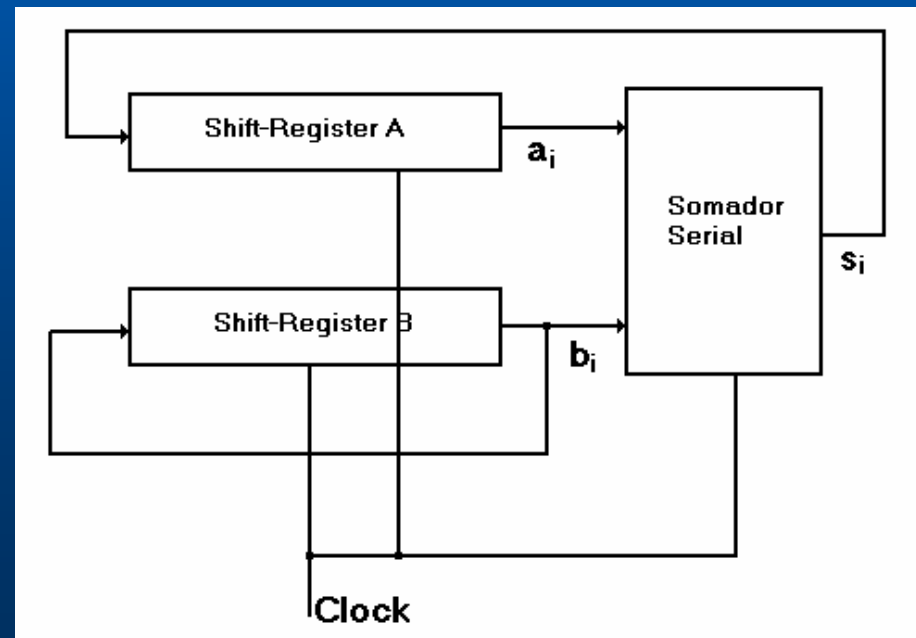
$y^{k+1}/s_i$

Sendo  $y^k = c_i \Rightarrow$  Estado Presente, ou seja, o valor do Carry a ser somado aos bits de entrada gerando  $s_i$  e  $c_{i+1} = y^{k+1}$

## Tabela Verdade do Somador Serial

$a_i$	$b_i$	$c_i$	$c_{i+1}$	$s_i$
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Circuito completo para os  $n$  bits.



## Implementando com FF tipo D:

	$a_i b_i$			
$y^k$	00	01	11	10
0	0	0	1	0
1	0	1	1	1

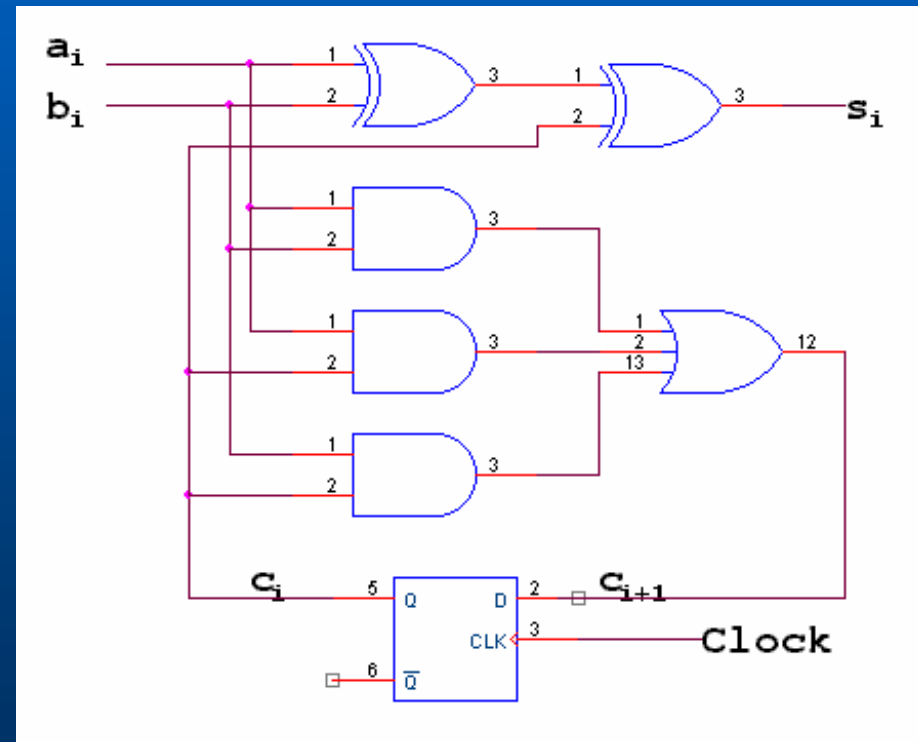
$y^{k+1}$

$$y^{k+1} = a_i b_i + a_i y^k + b_i y^k$$

$$c_{i+1} = a_i b_i + a_i c_i + b_i c_i$$

	$a_i b_i$			
$y^k$	00	01	11	10
0	0	1	0	1
1	1	0	1	0

$s_i$



$$s_i = a_i \oplus b_i \oplus c_i$$



## Síntese pelo Método da Equação de Aplicação (Somente para FF JK)

$$Q^* = J\bar{Q} + \bar{K}Q$$

- Etapa 1:** Gerar a Tabela de Estado Binária comum ao procedimento de síntese de Circuito Sequencial Síncrono.
- Etapa 2:** Ao invés de gerar as Tabelas de Excitação e os K-maps para FF JK, gerar os K-maps para FF tipo D.

## Síntese pelo Método da Equação de Aplicação (Somente para FF JK)

$$Q^* = J\bar{Q} + \bar{K}Q$$

**Etapa 3:** Para cada variável Próximo Estado  $Y_i$ , dividir seu K-map em duas metades, uma associada a  $y_i$  e outra associada a  $\bar{y}_i$ .

**Etapa 4:** Minimizar a função  $Y_i$ , mas não atravessar a fronteira que divide o mapa em duas metades. Isto gerará uma função na forma:  
$$Y_i = (J_i)\bar{y}_i + (\bar{K}_i)y_i$$

**Etapa 5:** Escrever as expressões para  $J_i$  e  $K_i$  diretamente da função da Etapa 4.

**Ex.:** Síntese do Circuito Sequencial Síncrono dado através da Tabela de Estado do exemplo anterior

### Tabela de Estado

y <sub>1</sub> y <sub>2</sub>	x	
	0	1
00	00/0	01/0
01	00/0	11/1
11	01/0	10/0
10	11/1	10/0

$$Y_i = (J_i)\bar{y}_i + (\bar{K}_i)y_i$$

y <sub>1</sub> y <sub>2</sub>	x	
	0	1
00	0	0
01	0	1
11	0	1
10	1	1

Y<sub>1</sub>

y <sub>1</sub> y <sub>2</sub>	x	
	0	1
00	0	1
01	0	1
11	1	0
10	1	0

Y<sub>2</sub>

$$Y_1 = (xy_2)\bar{y}_1 + (x+\bar{y}_2)y_1$$

$$J_1 = xy_2$$

$$K_1 = \overline{(x+\bar{y}_2)} = \bar{x}y_2$$

$$Y_2 = (x\bar{y}_1 + \bar{x}y_1)\bar{y}_2 + (x\bar{y}_1 + \bar{x}y_1)y_2 = (x \oplus y_1)\bar{y}_2 + (x \oplus y_1)y_2$$

$$J_2 = (x \oplus y_1)$$

$$K_2 = \overline{x \oplus y_1}$$