

DEPARTAMENTO DE ENGENHARIA  
ELÉTRICA

CIRCUITOS DIGITAIS I

---

# Contadores Síncronos

---

*Autor:*  
Caio NIGRO VIANNA

*Orientador:*  
Dr. Ricardo TOKIO  
HIGUTI

15 de agosto de 2019



UNIVERSIDADE ESTADUAL PAULISTA  
"JÚLIO DE MESQUITA FILHO"  
Campus de Ilha Solteira

## Sumário

<b>1</b>	<b>Introdução</b>	<b>2</b>
<b>2</b>	<b>A tabela verdade</b>	<b>3</b>
2.1	Estado atual e próximo estado . . . . .	3
2.2	Reinterpretando a tabela verdade dos flip-flops . . . . .	4
2.2.1	Flip-flop tipo D . . . . .	4
2.2.2	Flip-flop tipo T . . . . .	5
2.2.3	Flip-flop tipo JK . . . . .	5
2.3	Montando a tabela final . . . . .	7
2.4	Mapas de Karnaugh . . . . .	8
2.5	Revisão . . . . .	11
<b>3</b>	<b>Estados fora da contagem principal</b>	<b>11</b>
3.1	Diagrama de estados . . . . .	11
3.2	A solução mais rápida . . . . .	12
3.3	A solução mais barata . . . . .	15
3.4	Revisão . . . . .	20
<b>4</b>	<b>Exercício resolvido</b>	<b>20</b>

# 1 Introdução

Contador síncrono é um circuito digital formado por flip-flops em paralelos, tal que todas as entradas clocks estejam conectados na mesma fonte de clock. Graças a isso, a análise e o projeto de um contador síncrono é bem diferente da análise de um contador assíncrono; enquanto um exercício de um contador assíncrono geralmente inclui a análise de uma forma de onda para verificar quando e como cada flip-flop reage aos clocks, além do estudo de estado temporários e pulso espúrios, um exercício de um contador síncrono consiste em montar um contador, seguindo uma tabela verdade e projetando as entradas dos flip-flops.

O projeto de um contador síncrono que segue uma ordem crescente utiliza mais portas lógicas que um contador assíncrono, porém, nunca é necessário utilizar as funções preset e clear do contador síncrono, graças a isso, esse tipo de contador é preferível para ser utilizados em circuitos de alta frequência como a memória de um computador, onde o delay proporcionado pelos clocks em cascata e o estado temporário do contador assíncrono é altamente indesejável, podendo causar corrupção de dados caso um estado que não faz parte da sequência seja lido acidentalmente, enquanto um contador assíncrono pode ser preferível em um circuito alimentador de LEDs, onde qualquer estado errado nem seria visto por um olho humano e os atrasos podem ser ignorados ou são até desejados para criar um efeito cascata.

Além disso, um contador que deve seguir uma ordem que não é somente crescimento ou decréscimo unitário provavelmente será projetado como um contador síncrono, montar um contador assíncrono que segue outra sequência requer uma lógica específica para cada montagem, enquanto o projeto de um contador síncrono segue o mesmo padrão para qualquer sequência e contudo é preferível por muitos.

## 2 A tabela verdade

### 2.1 Estado atual e próximo estado

Em um contador síncrono, todos os estados possíveis devem produzir um próximo estado específico, determinado pela sequência que o contador vai seguir. Portanto, na tabela verdade do contador, é necessário colocar esse próximo estado do lado do estado atual.

**Exercício do capítulo:** Montar um contador síncrono que passa pela sequência aleatória 4 2 0 3 7 1 5 6. O objetivo final de um exercício de contador síncrono é determinar quais as entradas dos flip-flops que compõem o contador

A sequência contém oito números, entre 0 e 7. Estamos trabalhando com um contador de três estágios, variando entre números dentro da faixa  $000_2$  e  $111_2$ . Vamos começar a tabela verdade apenas com o Estado Atual preenchido:

Estado atual	Próximo estado
$Q_C Q_B Q_A$	$Q_C Q_B Q_A$
000	
001	
010	
011	
100	
101	
110	
111	

Colocar o estado atual em ordem crescente é sempre o primeiro passo. Um erro comum é colocar o estado atual seguindo a ordem da sequência do contador, como o estado atual é a primeira coluna e portanto a entrada da tabela verdade, ela deve começar em 0 e terminar no último valor possível pela quantidade de bits.

Agora vamos completar a tabela. A primeira linha é o 0, olhando para a sequência, o número que vem depois é o 3. Então escrevemos 011 na tabela. Na próxima linha temos o 1, na sequência o que vem depois é o 5. Então colocamos 101 na tabela. Lembre-se: na linha do número 6, o último elemento, nós colocamos o próximo estado como 100, porque o 4 é o primeiro elemento. Com essa lógica, é possível finalizar a tabela:

Estado atual	Próximo estado
$Q_C Q_B Q_A$	$Q_C Q_B Q_A$
000	011
001	101
010	000
011	111
100	010
101	110
110	100
111	001

## 2.2 Reinterpretando a tabela verdade dos flip-flops

Como dito anteriormente, um contador essencialmente é composto por flip-flops, e ainda não discutimos qual tipo será utilizado para montar esse contador. Existem três tipos de flip-flops que são comumente utilizados em contadores síncronos: o tipo D, o tipo T e o tipo JK. Graças ao fato que já temos o próximo estado de cada flip-flop na tabela completa e queremos determinar a entrada dos flip-flops, nós devemos inverter a tabela verdade que normalmente é apresentada, ou seja, colocar a entrada em função da saída.

### 2.2.1 Flip-flop tipo D

O próximo estado do flip-flop tipo D só depende da entrada D, o estado atual não tem impacto na saída do próximo estado. Graças a isso, o flip-flop tipo D tende a ser considerado mais simples que os outros. Vamos reservar uma coluna para o estado atual na tabela verdade do flip-flop D mesmo assim:

D	$Q_{\text{atual}}$	$Q_{\text{próximo}}$
0	0	0
0	1	0
1	0	1
1	1	1

Tabela 1: Tabela verdade do flip-flop tipo D.

Como já dito, o plano é deixar a entrada em função da saída. D e  $Q_{\text{próximo}}$  são sempre iguais, portanto reescrever a tabela com o  $Q_{\text{próximo}}$  na esquerda e o D na direita não requer nenhuma outra alteração:

Estado atual	Próximo estado	D
0	0	0
0	1	1
1	0	0
1	1	1

Tabela 2: Nova tabela verdade do flip-flop tipo D.

### 2.2.2 Flip-flop tipo T

A tabela do flip-flop tipo T depende dos valores de ambos os estados, se assemelhando a função XOR:

T	Q <sub>atual</sub>	Q <sub>próximo</sub>
0	0	0
0	1	1
1	0	1
1	1	0

Tabela 3: Tabela verdade do flip-flop tipo T.

T é a função *toggle*, ela altera o valor do estado se valer 1 ao invés de forçar um valor específico na saída. Portanto, T deve valer 0 quando os estados atual e próximo já são iguais e deve valer 1 caso haja mudança de valor, a tabela nova acaba seguindo a mesma ordem da sua apresentação regular:

Estado atual	Próximo estado	T
0	0	0
0	1	1
1	0	1
1	1	0

Tabela 4: Nova tabela verdade do flip-flop tipo T.

### 2.2.3 Flip-flop tipo JK

A tabela verdade do flip-flop tipo JK contém três bits de entrada e um único bit de saída:

J	K	$Q_{\text{atual}}$	$Q_{\text{próximo}}$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Tabela 5: Tabela verdade do flip-flop tipo JK

Nós estamos procurando uma tabela que contém apenas o Estado Atual e o Próximo Estado na entrada, com J e K na saída. Para transformarmos uma tabela de oito linhas em uma tabela de quatro linhas, nós precisaremos utilizar o *don't care* na saída, ou seja, encontrar casos onde uma das saídas produzidas pela combinação das entradas pode valer 0 ou 1 e conseguir manter o mesmo funcionamento no circuito. Isso é possível no flip-flop JK:

- Se  $Q_{\text{atual}}$  vale 0 e  $Q_{\text{próximo}}$  deve valer 0, a entrada pode ser a função manter ( $JK=00_2$ ) ou a função reset ( $JK=01_2$ );
- Se  $Q_{\text{atual}}$  vale 0 e  $Q_{\text{próximo}}$  deve valer 1, a entrada pode ser a função set ( $JK=10_2$ ) ou a função toggle ( $JK=11_2$ );
- Se  $Q_{\text{atual}}$  vale 1 e  $Q_{\text{próximo}}$  deve valer 0, a entrada pode ser a função reset ( $JK=01_2$ ) ou a função toggle ( $JK=11_2$ );
- Se  $Q_{\text{atual}}$  vale 1 e  $Q_{\text{próximo}}$  deve valer 1, a entrada pode ser a função manter ( $JK=00_2$ ) ou a função set ( $JK=10_2$ ).

O que se pode tirar disso é que o J manda na saída de uma maneira direta quando o estado atual é 0, porque ele pode manter essa saída quando ele for 0 ou forçar a saída para 1 se ele valer 1; o K é o seu complementar e manda na saída da mesma maneira quando o seu estado atual é 1, porque ele mantém a saída quando ele for 0 ou força a saída para 0 se valer 1. Ou seja, quando o seu estado atual é 0, a entrada K é irrelevante para o sistema: é o J quem vai ou não trocar o seu valor, seguindo a mesma lógica, se o seu estado atual é 1, a entrada J não tem mais função no sistema, apenas K determina o próximo estado. Com isso, a nova tabela verdade do flip-flop JK tem esse formato:

Estado atual	Próximo estado	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Tabela 6: Nova tabela verdade do flip-flop tipo JK

Pode-se dizer que o flip-flop JK funciona como uma junção dos flip-flops tipo D e tipo T, porque ele tem as opções de forçar um valor específico mas também de manter ou inverter

o valor atual. Graças a isso, nós continuaremos o exercício do capítulo (e todo o resto do material) apenas pensando no flip-flop tipo JK.

### 2.3 Montando a tabela final

Para terminar de montar a tabela verdade do nosso contador síncrono, nós precisamos colocar um par de entrada JK para cada estágio dos estados, porque a quantidade de flip-flops é igual a quantidade de bits.

Estado atual	Próximo estado	Entrada necessária								
		$Q_C$	$Q_B$	$Q_A$	$J_C$	$K_C$	$J_B$	$K_B$	$J_A$	$K_A$
000	011									
001	101									
010	000									
011	111									
100	010									
101	110									
110	100									
111	001									

Tabela 7: Modelo padrão de uma tabela de contadores síncronos.

Portanto, um contador de três estágios requer projetar seis entradas de 8 bits. Um contador de quatro estágios teria duas entradas a mais e o dobro de bits, ou seja, oito entradas e 16 bits. A quantidade de estados no seu contador pode aumentar ou diminuir drasticamente o tamanho do projeto.

Agora precisamos colocar o valor de cada entrada na tabela verdade. Vamos olhar para a primeira linha da Tabela 7:

Estado atual	Próximo estado	Entrada necessária								
		$Q_C$	$Q_B$	$Q_A$	$J_C$	$K_C$	$J_B$	$K_B$	$J_A$	$K_A$
000	011									

Temos que completar três pares. O valor  $Q_C$  está indo de 0 para 0. Olhando para a Tabela 6, as entradas devem ter valores  $J = 0$  e  $K = X$  nesse caso.

Estado atual	Próximo estado	Entrada necessária								
		$Q_C$	$Q_B$	$Q_A$	$J_C$	$K_C$	$J_B$	$K_B$	$J_A$	$K_A$
000	011				0	X				

Os valores de  $Q_B$  e  $Q_A$  ambos foram de 0 para 1. Na Tabela 6, essa ação gera  $J = 1$  e  $K = X$ .

Estado atual	Próximo estado	Entrada necessária					
$Q_C Q_B Q_A$	$Q_C Q_B Q_A$	$J_C$	$K_C$	$J_B$	$K_B$	$J_A$	$K_A$
000	011	0	X	1	X	1	X

Agora olhando para a linha 2:

Estado atual	Próximo estado	Entrada necessária					
$Q_C Q_B Q_A$	$Q_C Q_B Q_A$	$J_C$	$K_C$	$J_B$	$K_B$	$J_A$	$K_A$
001	101						

$Q_C$  está indo de 0 para 1. Completamos com  $J = 1$  e  $K = X$ .  $Q_B$  está indo de 0 para 0, então  $J = 0$  e  $K = X$ .  $Q_A$  vai de 1 para 1, isso nos dá  $J = X$  e  $K = 0$ .

Estado atual	Próximo estado	Entrada necessária					
$Q_C Q_B Q_A$	$Q_C Q_B Q_A$	$J_C$	$K_C$	$J_B$	$K_B$	$J_A$	$K_A$
001	101	1	X	0	X	X	0

Seguindo essa lógica, nós podemos completar todas as linhas da tabela:

Estado atual	Próximo estado	Entrada necessária					
$Q_C Q_B Q_A$	$Q_C Q_B Q_A$	$J_C$	$K_C$	$J_B$	$K_B$	$J_A$	$K_A$
000	011	0	X	1	X	1	X
001	101	1	X	0	X	X	0
010	000	0	X	X	1	0	X
011	111	1	X	X	0	X	0
100	010	X	1	1	X	0	X
101	110	X	0	1	X	X	1
110	100	X	0	X	1	0	X
111	001	X	1	X	1	X	0

Tabela 8: Tabela final do exercício do capítulo.

## 2.4 Mapas de Karnaugh

Lembre-se do objetivo final do exercício: determinar quais saídas devem ser ligadas nas entradas dos flip-flops. Ou seja, estamos atrás de uma função para  $J_C$ ,  $K_C$ ,  $J_B$ ,  $K_B$ ,  $J_A$  e para  $K_A$ , todas dependendo de  $Q_C$ ,  $Q_B$  e  $Q_A$ . Como temos a tabela verdade, o único passo que falta para encontrar tais funções é colocar as tabelas em mapas de Karnaugh.

Vamos fazer isso para  $J_C$ . Nós temos os seus valores na tabela verdade:

$J_C$
0
1
0
1
X
X
X
X

Então colocamos isso em um mapa de Karnaugh de três variáveis:

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

E então resolvemos o mapa para determinar  $J_C$ :

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

$$J_C = Q_A$$

Fazendo isso para todas as outras entradas:

$K_C$ :

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

$$K_C = \overline{Q_B} \cdot \overline{Q_A} + Q_B \cdot Q_A = \overline{Q_B \oplus Q_A}$$

 $J_B:$ 

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

$$J_B = Q_C + \overline{Q_A}$$

 $K_B:$ 

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

$$K_B = Q_C + \overline{Q_A}$$

 $J_A:$ 

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

$$J_A = \overline{Q_C} \cdot \overline{Q_B}$$

 $K_A:$ 

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

$$K_A = Q_C \cdot \overline{Q_B}$$

Com isso, projetamos o contador síncrono que segue a ordem 4 2 0 3 7 1 5 6. Para fisicamente montar esse contador, basta pegar alguns circuitos integrados de flip-flops tipo JK, CIs de portas lógicas (lembrando que nunca é necessária a utilização de portas inversoras, os flip-flops contém saídas  $\overline{Q_c} \cdot \overline{Q_b} \cdot \overline{Q_a}$ ), ligar um gerador de ondas como o clock e encaixar as saídas nas entradas seguindo o que foi projetado.

## 2.5 Revisão

Nesse capítulo, você aprendeu a montar um contador síncrono que passa por exatamente  $2^n$  estados utilizando  $n$  flip-flops tipo JK.

1. Começamos escrevendo a tabela verdade, a saída atual está na coluna de entrada e portanto vai em ordem crescente.
2. Olhamos para a sequência e montamos a coluna dos próximos estados.
3. Comparamos o estado atual com o próximo estado seguindo a Tabela 6. Terminamos a tabela inteira.
4. Mapas de Karnaugh são utilizados para determinar entradas.

## 3 Estados fora da contagem principal

Na seção anterior, nós montamos um contador síncrono de três estágios que passa por oito estados, a quantidade máxima possível. Nessa seção, vamos aprender o que deve ser feito caso estamos projetando um contador que tem alguns estados fora da contagem

**Exercício do capítulo:** Montar um contador síncrono que passa pela sequência 1 3 5 7, números ímpares crescente. O objetivo desse exercício é encontrar expressões para as entradas  $J$  e  $K$  assim como o capítulo anterior, mas também vamos precisar escolher valores para essas entradas e garantir que o contador não entra em um loop indesejado. Antes, vamos aplicar o diagrama de estados em contadores síncronos.

### 3.1 Diagrama de estados

Resumidamente, o diagrama de estados é um desenho que esquematiza a sequência de um contador, mostrando todos os possíveis estados e seus respectivos próximos estados. Na seção anterior, nós trabalhamos com a sequência 4 2 0 3 7 1 5 6. O seu diagrama de estados performa um loop perfeito:

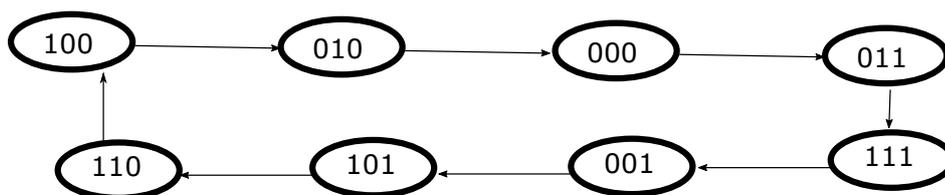


Figura 1: Diagrama de estados do exemplo anterior.

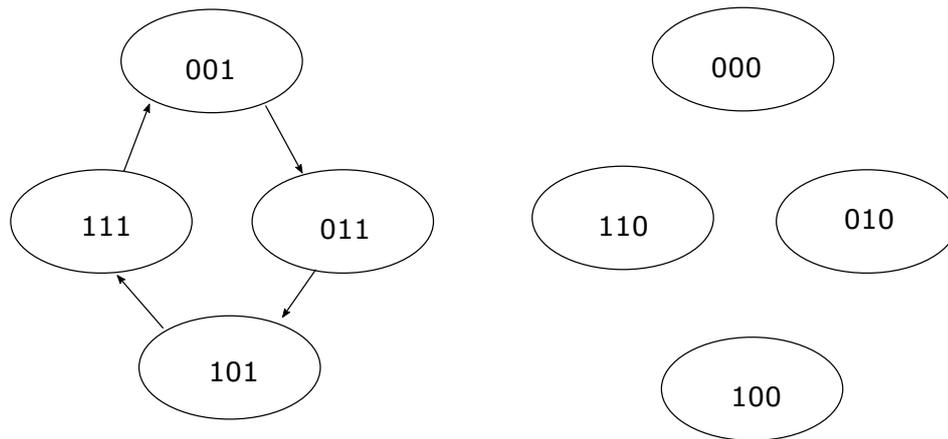


Figura 2: Diagrama incompleto da nossa seqüência.

Porém, no nosso exemplo, alguns estados estão fora da contagem principal. O diagrama de estados sempre precisa apresentar *todos* os estados possíveis da contagem.

Para projetar um contador que não tem todos os estados na sua contagem principal, nós precisamos determinar o que acontece caso o contador entre em um desses estados, ou seja, como explicado anteriormente, vamos escolher valores para  $J$  e  $K$  e ver como que o próximo estado reage. Nós vamos ver duas maneira de se completar um projeto que contém esses estados fora do loop.

### 3.2 A solução mais rápida

Estamos projetando um contador com oito estados. Nós aprendemos na seção anterior como é o formato da tabela verdade. Vamos aplicar o que já sabemos nos estados que fazem parte da sequencia e deixar apenas um \_ nos próximos estados que não conhecemos para montar a tabela verdade:

Estado atual	Próximo estado	Entrada necessária					
$Q_C Q_B Q_A$	$Q_C Q_B Q_A$	$J_C$	$K_C$	$J_B$	$K_B$	$J_A$	$K_A$
000	$\overline{\quad}$	$\overline{0}$	$\overline{X}$	$\overline{1}$	$\overline{X}$	$\overline{X}$	$\overline{0}$
001	011	0	X	1	X	X	0
010	$\overline{\quad}$	$\overline{1}$	$\overline{X}$	$\overline{X}$	$\overline{1}$	$\overline{X}$	$\overline{0}$
011	101	1	X	X	1	X	0
100	$\overline{\quad}$	$\overline{X}$	$\overline{0}$	$\overline{1}$	$\overline{X}$	$\overline{X}$	$\overline{0}$
101	111	X	0	1	X	X	0
110	$\overline{\quad}$	$\overline{X}$	$\overline{1}$	$\overline{X}$	$\overline{1}$	$\overline{X}$	$\overline{0}$
111	001	X	1	X	1	X	0

A solução mais rápida para um contador assim é forçar a contagem a ir para o primeiro estado da seqüência caso o contador seja ligado fora dela. Nesse caso, vamos colocar 001 como o próximo estado em todos os estados que não fazem parte do loop que é relevante:

Estado atual	Próximo estado	Entrada necessária								
		$Q_C$	$Q_B$	$Q_A$	$J_C$	$K_C$	$J_B$	$K_B$	$J_A$	$K_A$
000	<u>0</u> <u>0</u> <u>1</u>	$\bar{0}$	$\bar{0}$	$\bar{1}$	$\bar{0}$	$\bar{1}$	$\bar{0}$	$\bar{0}$	$\bar{1}$	$\bar{0}$
001	011	0	X	1	X	1	X	X	0	0
010	<u>0</u> <u>0</u> <u>1</u>	$\bar{0}$	$\bar{0}$	$\bar{1}$	$\bar{0}$	$\bar{1}$	$\bar{0}$	$\bar{1}$	$\bar{0}$	$\bar{0}$
011	101	1	X	X	1	X	1	X	0	0
100	<u>0</u> <u>0</u> <u>1</u>	$\bar{0}$	$\bar{0}$	$\bar{1}$	$\bar{0}$	$\bar{1}$	$\bar{0}$	$\bar{1}$	$\bar{0}$	$\bar{0}$
101	111	X	0	1	X	1	X	X	0	0
110	<u>0</u> <u>0</u> <u>1</u>	$\bar{0}$	$\bar{0}$	$\bar{1}$	$\bar{0}$	$\bar{1}$	$\bar{0}$	$\bar{1}$	$\bar{0}$	$\bar{0}$
111	001	X	1	X	1	X	1	X	0	0

Como agora nós temos o estado atual e o próximo estado em todas as linhas, podemos completar a tabela e resolver o exercício igual o que fizemos na seção anterior, utilizando a Tabela 6

Estado atual	Próximo estado	Entrada necessária								
		$Q_C$	$Q_B$	$Q_A$	$J_C$	$K_C$	$J_B$	$K_B$	$J_A$	$K_A$
000	<u>0</u> <u>0</u> <u>1</u>	<u>0</u>	<u>X</u>	<u>0</u>	<u>X</u>	<u>1</u>	<u>X</u>	<u>1</u>	<u>X</u>	<u>X</u>
001	011	0	X	1	X	1	X	X	0	0
010	<u>0</u> <u>0</u> <u>1</u>	<u>0</u>	<u>X</u>	<u>X</u>	<u>1</u>	<u>1</u>	<u>X</u>	<u>1</u>	<u>X</u>	<u>X</u>
011	101	1	X	X	1	X	1	X	0	0
100	<u>0</u> <u>0</u> <u>1</u>	<u>X</u>	<u>1</u>	<u>0</u>	<u>X</u>	<u>1</u>	<u>X</u>	<u>1</u>	<u>X</u>	<u>X</u>
101	111	X	0	1	X	1	X	X	0	0
110	<u>0</u> <u>0</u> <u>1</u>	<u>X</u>	<u>1</u>	<u>X</u>	<u>1</u>	<u>1</u>	<u>X</u>	<u>1</u>	<u>X</u>	<u>X</u>
111	001	X	1	X	1	X	1	X	0	0

$J_C$ :

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

$$J_C = Q_B \cdot Q_A$$

$K_C$ :

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

$$K_C = Q_B + \overline{Q_A}$$

$J_B$ :

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

$K_B$ :

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

$J_B = Q_A$

$K_B = 1$

$J_C$ :

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

$K_C$ :

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

$J_A = 1$

$K_A = 0$

Com tudo isso, nós projetamos um contador que só passa pela sequência 1 3 5 7 que também garante que, caso o contador seja ligado em outro estado, imediatamente se conserta para a sequência correta. O seu diagrama de estados pode ser mostrado assim:

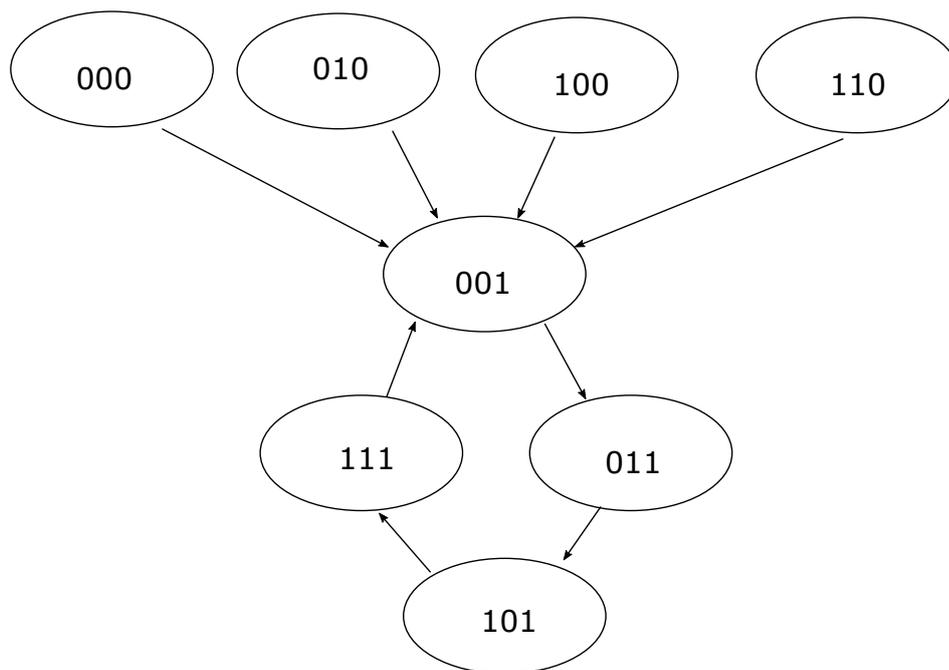


Figura 3: Diagrama completo da nossa sequência para a primeira resolução.

### 3.3 A solução mais barata

Existe uma outra solução que tenta diminuir a quantidade de portas lógicas no circuito. Essa solução permite a possibilidade do contador ser ligado em um estado irrelevante e continuar a contagem em estados irrelevantes por um tempo limitado, mas ela também garante que não haja a possibilidade de um loop errado no diagrama de estados. Esse tipo de resolução do exercício é o mais apropriado.

Se olharmos para a tabela final de qualquer projeto de contador síncrono, percebe-se um padrão na posição dos *don't care*. É possível completar a localização de todos os *don't care* mesmo antes de pensar em como vamos completar o próximo estado. Repare que, quando o estado atual for 0, o valor de  $K$  é  $X$  independente do próximo estado, similarmente, quando o estado atual vale 1, o valor de  $J$  é  $X$ , sempre. Então, nós já podemos escrever uma parte da nossa tabela verdade:

Estado atual	Próximo estado	Entrada necessária						
		$Q_C Q_B Q_A$	$Q_C Q_B Q_A$	$J_C$	$K_C$	$J_B$	$K_B$	$J_A$
000	—	—	X	—	X	—	X	
001	011	0	X	1	X	X	0	
010	—	—	X	X	—	—	X	
011	101	1	X	X	1	X	0	
100	—	X	—	—	X	—	X	
101	111	X	0	1	X	X	0	
110	—	X	—	X	—	—	X	
111	001	X	1	X	1	X	0	

Tabela 9: *Don't care* já aplicados.

O que nós vamos fazer agora é levar esses termos desconhecidos para os mapas de Karnaugh. Lá, vamos analisar quais valores para cada um deles fica melhor na resolução do próprio mapa, anotar, e ver como que o próximo estado reage na nossa contagem. O funcionamento é bem parecido com o *don't care*, mas nós precisamos ter o cuidado de anotar qual valor que estamos assumindo para cada possibilidade, porque não podemos fazer um loop que está fora da contagem.

Vamos fazer isso para a entrada  $J_C$ :

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

Se completarmos o termo zero com 0 e o termo dois com 1, poderemos resolver o mapa de Karnaugh com um grupo de quatro entradas. É a melhor coisa a se fazer nesse

caso. Atenção: sempre mantenha sublinhado as entradas, porque depois poderemos ter que mudar alguma coisa para impedir um loop indesejado.

		$Q_B Q_A$			
		00	01	11	10
$Q_C$	0	<u>0</u>	0	1	<u>1</u>
	1	X	X	X	X

$$J_C = Q_B$$

Voltando isso para a tabela:

Estado atual	Próximo estado	Entrada necessária							
		$Q_C$	$Q_B$	$Q_A$	$J_C$	$K_C$	$J_B$	$K_B$	$J_A$
000	<u>0</u>	0	X	-	X	-	X	-	X
001	011	0	X	1	X	X	X	X	0
010	<u>1</u>	1	X	X	-	-	-	-	X
011	101	1	X	X	1	X	1	X	0
100	<u>X</u>	X	-	-	X	-	X	-	X
101	111	X	0	1	X	X	X	X	0
110	<u>X</u>	X	-	X	-	-	-	-	X
111	001	X	1	X	1	X	1	X	0

Tendo estado atual e a entrada, completamos o próximo estado:

Estado atual	Próximo estado	Entrada necessária							
		$Q_C$	$Q_B$	$Q_A$	$J_C$	$K_C$	$J_B$	$K_B$	$J_A$
000	<u>0</u>	0	X	-	X	-	X	-	X
001	011	0	X	1	X	X	X	X	0
010	<u>1</u>	1	X	X	-	-	-	-	X
011	101	1	X	X	1	X	1	X	0
100	<u>X</u>	X	-	-	X	-	X	-	X
101	111	X	0	1	X	X	X	X	0
110	<u>X</u>	X	-	X	-	-	-	-	X
111	001	X	1	X	1	X	1	X	0

Fazendo isso para as outras cinco entradas dos flip-flops...  $K_C$ :

		$Q_B Q_A$			
		00	01	11	10
$Q_C$	0	X	X	X	X
	1	<u>0</u>	0	1	<u>1</u>

$$K_C = Q_B$$

$J_B$ :

		$Q_B Q_A$			
		00	01	11	10
$Q_C$	0	<u>1</u>	1	X	X
	1	<u>1</u>	1	X	X

$K_B$ :

		$Q_B Q_A$			
		00	01	11	10
$Q_C$	0	X	X	1	<u>1</u>
	1	X	X	1	<u>1</u>

$$J_B = 1$$

$$K_B = 1$$

$J_A$ :

		$Q_B Q_A$			
		00	01	11	10
$Q_C$	0	<u>0</u>	X	X	<u>0</u>
	1	<u>0</u>	X	X	<u>0</u>

$K_A$ :

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

$J_A$  é um caso um pouco curioso, porque preencher com 0 ou com 1 satisfaz a resolução. Escolhemos 0.

$$K_A = 0$$

Apesar que determinados todas as entradas, é imperativo verificar o que acontece com o próximo estado.

Estado atual	Próximo estado	Entrada necessária					
$Q_C Q_B Q_A$	$Q_C Q_B Q_A$	$J_C$	$K_C$	$J_B$	$K_B$	$J_A$	$K_A$
000	<u>0</u>	<u>0</u>	X	<u>1</u>	X	<u>0</u>	X
001	011	0	X	1	X	X	0
010	<u>1</u>	<u>1</u>	X	X	<u>1</u>	<u>0</u>	X
011	101	1	X	X	1	X	0
100	—	X	<u>0</u>	<u>1</u>	X	<u>0</u>	X
101	111	X	0	1	X	X	0
110	—	X	<u>1</u>	X	<u>1</u>	<u>0</u>	X
111	001	X	1	X	1	X	0

Estado atual	Próximo estado	Entrada necessária					
$Q_C Q_B Q_A$	$Q_C Q_B Q_A$	$J_C$	$K_C$	$J_B$	$K_B$	$J_A$	$K_A$
000	<u>0 1 0</u>	<u>0</u>	X	<u>1</u>	X	<u>0</u>	X
001	011	0	X	1	X	X	0
010	<u>1 0 0</u>	<u>1</u>	X	X	<u>1</u>	<u>0</u>	X
011	101	1	X	X	1	X	0
100	<u>1 1 0</u>	X	<u>0</u>	<u>1</u>	X	<u>0</u>	X
101	111	X	0	1	X	X	0
110	<u>0 0 0</u>	X	<u>1</u>	X	<u>1</u>	<u>0</u>	X
111	001	X	1	X	1	X	0

Tabela 10: Exercício ainda não resolvido

Se analisarmos o que acontece quando o contador é ligado, por exemplo, no estado 000, perceberemos que há um problema. O diagrama de estados do contador é apresentado assim:

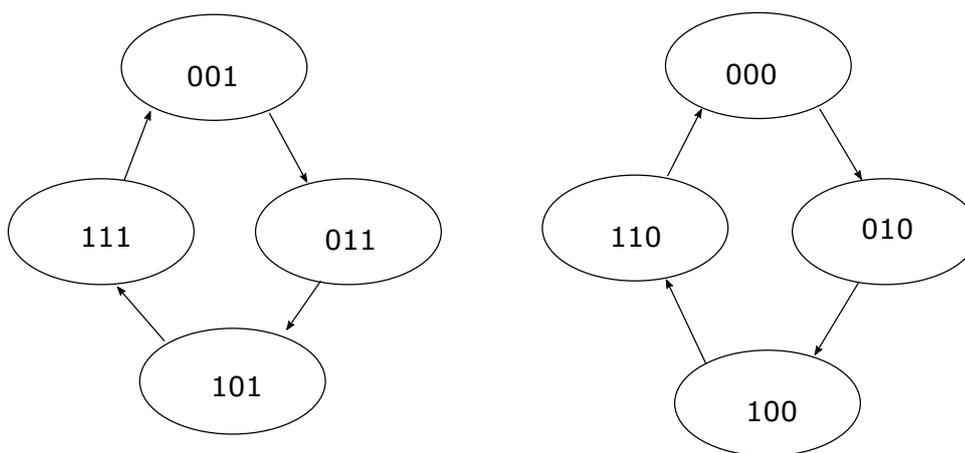


Figura 4: Diagrama da resolução atual.

Na tentativa de projetar um contador que conta apenas números ímpares, projetamos um contador que pode também contar números pares, caso ele ligue nesse estado. Isso é um defeito grave, o contador nunca vai começar a funcionar como esperado nessa situação. É também por isso que nós sempre estudamos como que o contador reage às

entradas que estamos colocando nos flip-flops até nos números fora da sequência, se tivéssemos colocado *don't care* em todos os estados que são irrelevantes, eventualmente nós projetaríamos esse contador defeituoso sem condições de perceber o seu problema.

Para solucionar isso, nós podemos escolher qualquer um dos estados da contagem paralela e forçar o seu próximo estado a ir para um valor ímpar, parecido com a solução mais rápida. Uma outra possibilidade, bem específica para esse exemplo, é trocar o valor de  $J_A$ . Ele é inteiramente composto de *don't care* e de valores escolhidos por nós. Originalmente usamos  $J_A = 0$ , vamos agora tentar  $J_A = 1$ .

Estado atual	Próximo estado	Entrada necessária							
		$Q_C Q_B Q_A$	$Q_C Q_B Q_A$	$J_C$	$K_C$	$J_B$	$K_B$	$J_A$	$K_A$
000	<u>0</u> <u>1</u> <u>1</u>	<u>0</u>	X	<u>1</u>	X	<u>1</u>	X	<u>1</u>	X
001	0 <u>1</u> <u>1</u>	0	X	1	X	X	0	X	0
010	<u>1</u> <u>0</u> <u>1</u>	<u>1</u>	X	X	<u>1</u>	<u>1</u>	X	<u>1</u>	X
011	1 <u>0</u> <u>1</u>	1	X	X	1	X	0	X	0
100	<u>1</u> <u>1</u> <u>1</u>	X	<u>0</u>	<u>1</u>	X	<u>1</u>	X	<u>1</u>	X
101	1 <u>1</u> <u>1</u>	X	0	1	X	X	0	X	0
110	<u>0</u> <u>0</u> <u>1</u>	X	<u>1</u>	X	<u>1</u>	<u>1</u>	X	<u>1</u>	X
111	0 <u>0</u> <u>1</u>	X	1	X	1	X	0	X	0

Tabela 11: Tabela verdade final do exercício.

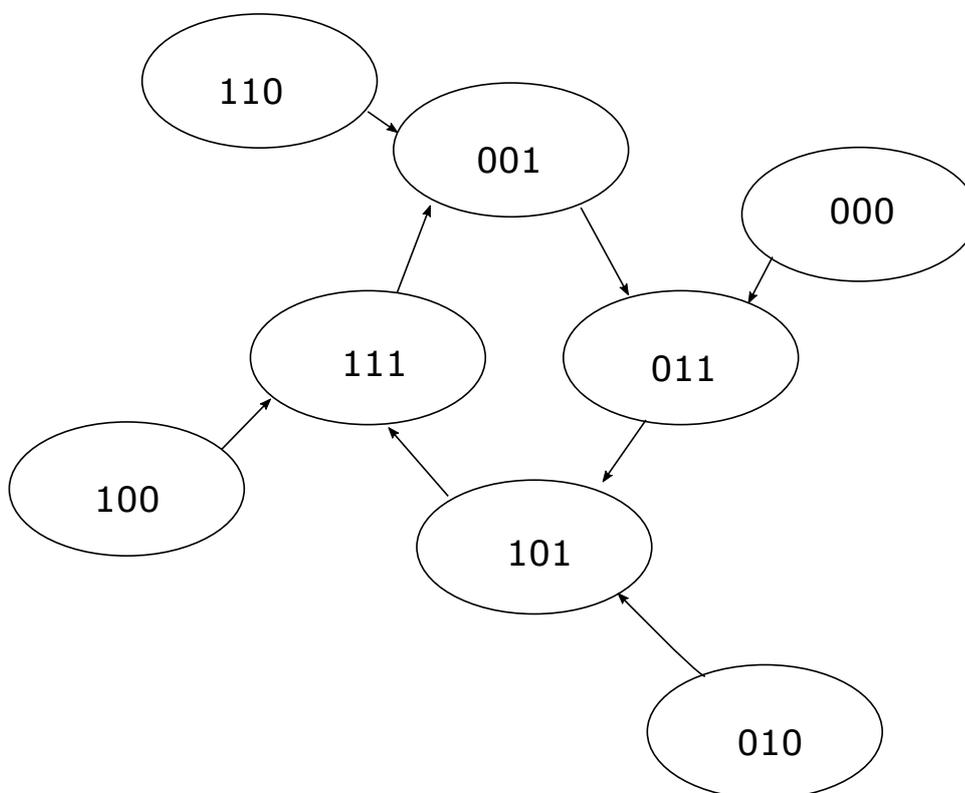


Figura 5: Diagrama final do exercício.

### 3.4 Revisão

Nesse capítulo, você aprendeu o que deve ser feito para um contador síncrono que tem estágios fora da contagem principal.

1. Começamos novamente escrevendo a tabela verdade. Note que já é possível colocar os valores *don't care* nos seus lugares até em estados fora da contagem.
2. Deixamos os outros sublinhados não preenchidos e levamos para os mapas de Karnaugh.
3. Apenas no mapa que podemos preencher os sublinhados, aí, voltamos para a tabela para preencher o próximo estado.
4. Desenhamos o diagrama de estado. Se verificarmos loop fora da contagem principal, alguns termos sublinhados serão trocados para forçar próximos estados diferentes.

## 4 Exercício resolvido

**Monte um contador síncrono que passa pela sequência 13 0 15 5 11 10 6 1 12 2 4, utilizando flip-flops tipo JK e o projeto mais barato. Apresente, também, o diagrama de estados *completo* do contador.**

Estado atual	Próximo estado	Entrada necessária											
		$Q_D$	$Q_C$	$Q_B$	$Q_A$	$J_D$	$K_D$	$J_C$	$K_C$	$J_B$	$K_B$	$J_A$	$K_A$
0000													
0001													
0010													
0011													
0100													
0101													
0110													
0111													
1000													
1001													
1010													
1011													
1100													
1101													
1110													
1111													

Estado atual	Próximo estado	Entrada necessária			
$Q_D Q_C Q_B Q_A$	$Q_D Q_C Q_B Q_A$	$J_D$ $K_D$	$J_C$ $K_C$	$J_B$ $K_B$	$J_A$ $K_A$
0000	1111				
0001	1100				
0010	0100				
0011	_____				
0100	1101				
0101	1011				
0110	0001				
0111	_____				
1000					
1001					
1010					
1011					
1100					
1101					
1110					
1111					

Estado atual	Próximo estado	Entrada necessária			
$Q_D Q_C Q_B Q_A$	$Q_D Q_C Q_B Q_A$	$J_D$ $K_D$	$J_C$ $K_C$	$J_B$ $K_B$	$J_A$ $K_A$
0000	1111				
0001	1100				
0010	0100				
0011	_____				
0100	1101				
0101	1011				
0110	0001				
0111	_____				
1000	_____				
1001	_____				
1010	0110				
1011	1010				
1100	0010				
1101	0000				
1110	_____				
1111	0101				

Estado atual	Próximo estado	Entrada necessária									
		$Q_D Q_C Q_B Q_A$	$Q_D Q_C Q_B Q_A$	$J_D$	$K_D$	$J_C$	$K_C$	$J_B$	$K_B$	$J_A$	$K_A$
0000	1111			1	X						
0001	1100			1	X						
0010	0100			0	X						
0011	_____			—	X						
0100	1101			1	X						
0101	1011			1	X						
0110	0001			1	X						
0111	_____			—	X						
1000	_____			X	—						
1001	_____			X	—						
1010	0110			X	1						
1011	1010			X	0						
1100	0010			X	1						
1101	0000			X	1						
1110	_____			X	—						
1111	0101			X	1						

Estado atual	Próximo estado	Entrada necessária									
		$Q_D Q_C Q_B Q_A$	$Q_D Q_C Q_B Q_A$	$J_D$	$K_D$	$J_C$	$K_C$	$J_B$	$K_B$	$J_A$	$K_A$
0000	1111			1	X	1	X				
0001	1100			1	X	1	X				
0010	0100			0	X	1	X				
0011	_____			—	X	—	X				
0100	1101			1	X	X	0				
0101	1011			1	X	X	1				
0110	0001			1	X	X	1				
0111	_____			—	X	X	—				
1000	_____			X	—	—	X				
1001	_____			X	—	—	X				
1010	0110			X	1	1	X				
1011	1010			X	0	0	X				
1100	0010			X	1	X	1				
1101	0000			X	1	X	1				
1110	_____			X	—	X	—				
1111	0101			X	1	X	0				

Estado atual	Próximo estado	Entrada necessária							
$Q_D Q_C Q_B Q_A$	$Q_D Q_C Q_B Q_A$	$J_D$	$K_D$	$J_C$	$K_C$	$J_B$	$K_B$	$J_A$	$K_A$
0000	1111	1	X	1	X	1	X		
0001	1100	1	X	1	X	0	X		
0010	0100	0	X	1	X	X	1		
0011	_____	—	X	—	X	X	—		
0100	1101	1	X	X	0	0	X		
0101	1011	1	X	X	1	1	X		
0110	0001	1	X	X	1	X	1		
0111	_____	—	X	X	—	X	—		
1000	_____	X	—	—	X	—	X		
1001	_____	X	—	—	X	—	X		
1010	0110	X	1	1	X	X	0		
1011	1010	X	0	0	X	X	0		
1100	0010	X	1	X	1	1	X		
1101	0000	X	1	X	1	0	X		
1110	_____	X	—	X	—	X	—		
1111	0101	X	1	X	0	X	0		

Estado atual	Próximo estado	Entrada necessária							
$Q_D Q_C Q_B Q_A$	$Q_D Q_C Q_B Q_A$	$J_D$	$K_D$	$J_C$	$K_C$	$J_B$	$K_B$	$J_A$	$K_A$
0000	1111	1	X	1	X	1	X	1	X
0001	1100	1	X	1	X	0	X	X	1
0010	0100	0	X	1	X	X	1	0	X
0011	_____	—	X	—	X	X	—	X	—
0100	1101	1	X	X	0	0	X	1	X
0101	1011	1	X	X	1	1	X	X	0
0110	0001	1	X	X	1	X	1	1	X
0111	_____	—	X	X	—	X	—	X	—
1000	_____	X	—	—	X	—	X	—	X
1001	_____	X	—	—	X	—	X	X	—
1010	0110	X	1	1	X	X	0	0	X
1011	1010	X	0	0	X	X	0	X	1
1100	0010	X	1	X	1	1	X	0	X
1101	0000	X	1	X	1	0	X	X	1
1110	_____	X	—	X	—	X	—	—	X
1111	0101	X	1	X	0	X	0	X	0

$J_D:$ 

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

 $K_D:$ 

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

 $J_C:$ 

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

 $K_C:$ 

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

$J_B:$ 

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

 $K_B:$ 

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

 $J_A:$ 

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

 $K_A:$ 

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

$J_D$ :

		$Q_B Q_A$			
		00	01	11	10
$Q_D Q_C$	00	1	1	<u>0</u>	0
	01	1	1	<u>1</u>	1
	11	X	X	X	X
	10	X	X	X	X

$$J_D = Q_C + \overline{Q_B}$$

$K_D$ :

		$Q_B Q_A$			
		00	01	11	10
$Q_D Q_C$	00	X	X	X	X
	01	X	X	X	X
	11	1	1	1	<u>1</u>
	10	<u>1</u>	<u>0</u>	0	1

$$K_D = Q_C + \overline{Q_A}$$

$J_C$ :

		$Q_B Q_A$			
		00	01	11	10
$Q_D Q_C$	00	1	1	<u>0</u>	1
	01	X	X	X	X
	11	X	X	X	X
	10	<u>1</u>	<u>1</u>	0	1

$$J_C = \overline{Q_B} + \overline{Q_A}$$

$K_C$ :

		$Q_B Q_A$			
		00	01	11	10
$Q_D Q_C$	00	X	X	X	X
	01	0	1	<u>0</u>	1
	11	1	1	0	<u>1</u>
	10	X	X	X	X

$$K_C = \overline{Q_B} \cdot Q_A + Q_B \cdot \overline{Q_A} + Q_D \cdot \overline{Q_B}$$

$J_B$ :

		$Q_B Q_A$			
		00	01	11	10
$Q_D Q_C$	00	1	0	X	X
	01	0	1	X	X
	11	1	0	X	X
	10	<u>0</u>	<u>1</u>	X	X

$J_B = \overline{Q_D \oplus Q_C \oplus Q_B \oplus Q_A}$ , utilizando a regra do mapa de Karnaugh xadrez.

$K_B$ :

		$Q_B Q_A$			
		00	01	11	10
$Q_D Q_C$	00	X	X	<u>1</u>	<u>1</u>
	01	X	X	<u>1</u>	<u>1</u>
	11	X	X	0	<u>0</u>
	10	X	X	0	0

$K_B = \overline{Q_D}$

$J_A$ :

		$Q_B Q_A$			
		00	01	11	10
$Q_D Q_C$	00	1	X	X	0
	01	1	X	X	1
	11	0	X	X	<u>0</u>
	10	<u>0</u>	X	X	0

$J_A = \overline{Q_D} \cdot Q_C + \overline{Q_D} \cdot \overline{Q_A} = \overline{Q_D} \cdot (Q_C + \overline{Q_A})$

$K_A$ :

		$Q_B Q_A$			
		00	01	11	10
$Q_D Q_C$	00	X	1	<u>0</u>	X
	01	X	0	<u>1</u>	X
	11	X	1	0	X
	10	X	<u>0</u>	1	X

$K_A = Q_D \oplus Q_C \oplus Q_B \oplus Q_A$ , utilizando a regra do mapa de Karnaugh xadrez.

Estado atual	Próximo estado	Entrada necessária											
		$Q_D$	$Q_C$	$Q_B$	$Q_A$	$J_D$	$K_D$	$J_C$	$K_C$	$J_B$	$K_B$	$J_A$	$K_A$
0000	1111	1	X	1	X	1	X	1	X	1	X	1	X
0001	1100	1	X	1	X	0	X	X	1	X	X	1	X
0010	0100	0	X	1	X	X	1	0	X	1	X	0	X
0011	_____	<u>0</u>	X	<u>0</u>	X	X	<u>1</u>	X	<u>1</u>	X	<u>1</u>	X	<u>0</u>
0100	1101	1	X	X	0	0	X	1	X	1	X	1	X
0101	1011	1	X	X	1	1	X	X	1	X	X	0	X
0110	0001	1	X	X	1	X	1	X	1	X	1	1	X
0111	_____	<u>1</u>	X	X	<u>0</u>	X	<u>1</u>	X	<u>1</u>	X	<u>1</u>	X	<u>1</u>
1000	_____	X	<u>1</u>	<u>1</u>	X	<u>0</u>	X	<u>0</u>	X	<u>0</u>	X	<u>0</u>	X
1001	_____	X	<u>0</u>	<u>1</u>	X	<u>1</u>	X	X	<u>0</u>	X	X	<u>0</u>	X
1010	0110	X	1	1	X	X	0	0	X	0	0	X	X
1011	1010	X	0	0	X	X	0	X	0	X	X	1	X
1100	0010	X	1	X	1	1	X	0	X	0	X	0	X
1101	0000	X	1	X	1	0	X	X	1	X	X	1	X
1110	_____	X	<u>1</u>	X	<u>1</u>	X	<u>0</u>	X	<u>0</u>	X	<u>0</u>	X	X
1111	0101	X	1	X	0	X	0	X	0	X	X	0	X

Estado atual	Próximo estado	Entrada necessária											
		$Q_D$	$Q_C$	$Q_B$	$Q_A$	$J_D$	$K_D$	$J_C$	$K_C$	$J_B$	$K_B$	$J_A$	$K_A$
0000	1111	1	X	1	X	1	X	1	X	1	X	1	X
0001	1100	1	X	1	X	0	X	X	1	X	X	1	X
0010	0100	0	X	1	X	X	1	0	X	1	X	0	X
0011	<u>0001</u>	<u>0</u>	X	<u>0</u>	X	X	<u>1</u>	X	<u>1</u>	X	<u>1</u>	X	<u>0</u>
0100	1101	1	X	X	0	0	X	1	X	1	X	1	X
0101	1011	1	X	X	1	1	X	X	1	X	X	0	X
0110	0001	1	X	X	1	X	1	X	1	X	1	1	X
0111	<u>1100</u>	<u>1</u>	X	X	<u>0</u>	X	<u>1</u>	X	<u>1</u>	X	<u>1</u>	X	<u>1</u>
1000	<u>0100</u>	X	<u>1</u>	<u>1</u>	X	<u>0</u>	X	<u>0</u>	X	<u>0</u>	X	<u>0</u>	X
1001	<u>1111</u>	X	<u>0</u>	<u>1</u>	X	<u>1</u>	X	X	<u>0</u>	X	X	<u>0</u>	X
1010	0110	X	1	1	X	X	0	0	X	0	0	X	X
1011	1010	X	0	0	X	X	0	X	0	X	X	1	X
1100	0010	X	1	X	1	1	X	0	X	0	X	0	X
1101	0000	X	1	X	1	0	X	X	1	X	X	1	X
1110	<u>0010</u>	X	<u>1</u>	X	<u>1</u>	X	<u>0</u>	X	<u>0</u>	X	<u>0</u>	X	X
1111	0101	X	1	X	0	X	0	X	0	X	X	0	X

