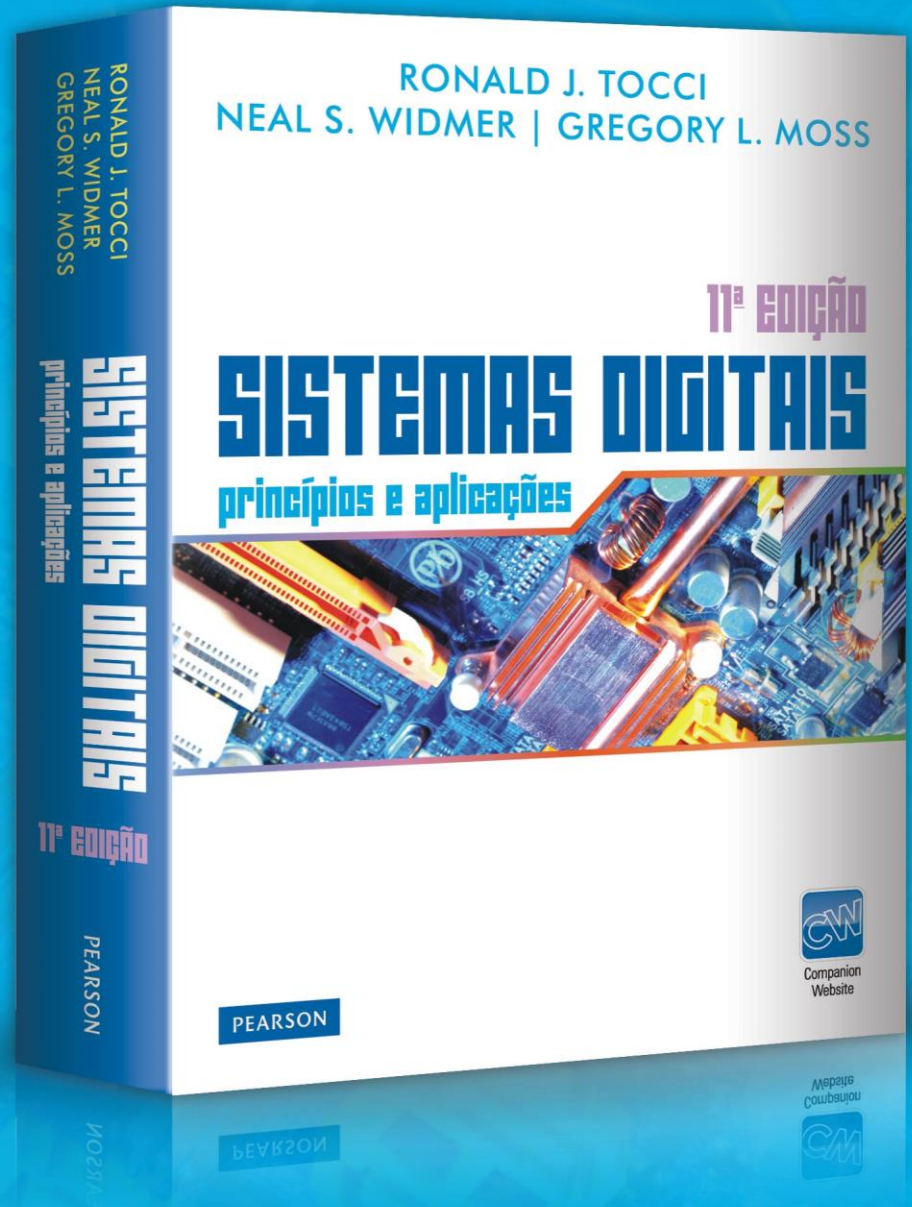


Capítulo 7

Contadores e Registradores

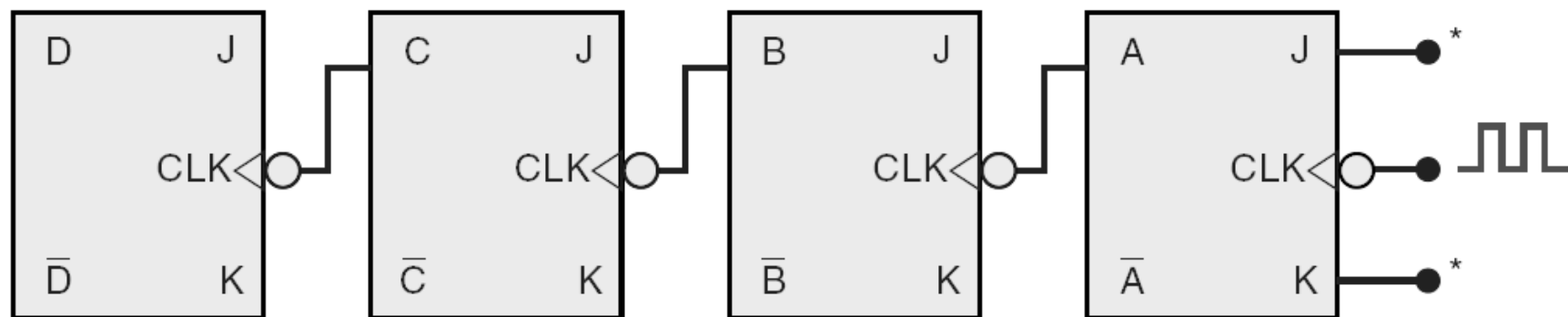


Os temas abordados nesse capítulo são:

- Operação e características dos contadores síncronos e assíncronos.
- Análise e avaliação dos vários tipos de contadores.
- Esquemas usados para decodificar diferentes tipos de contadores.
- Circuitos de contadores usando diferentes níveis de abstração no HDL.
- Operação de vários tipos de registros IC.
- Mudança de registradores de deslocamento e contadores com registradores de deslocamento usando HDL.
- Técnicas de análise de defeitos usadas em sistemas de lógica combinacional na análise de sistemas sequenciais.

7.1 Contadores assíncronos

- Revisão de operações de quatro bits.
- Clock aplicado apenas na entrada CLK para flip-flop A .
- J e K são altos em todos os flip-flops.
- A saída de flip-flop A é clock de entrada para o flip-flop B .
- As saídas dos flip-flops D , C , B e A são um número binário de 4 bits, sendo o D o MSB.

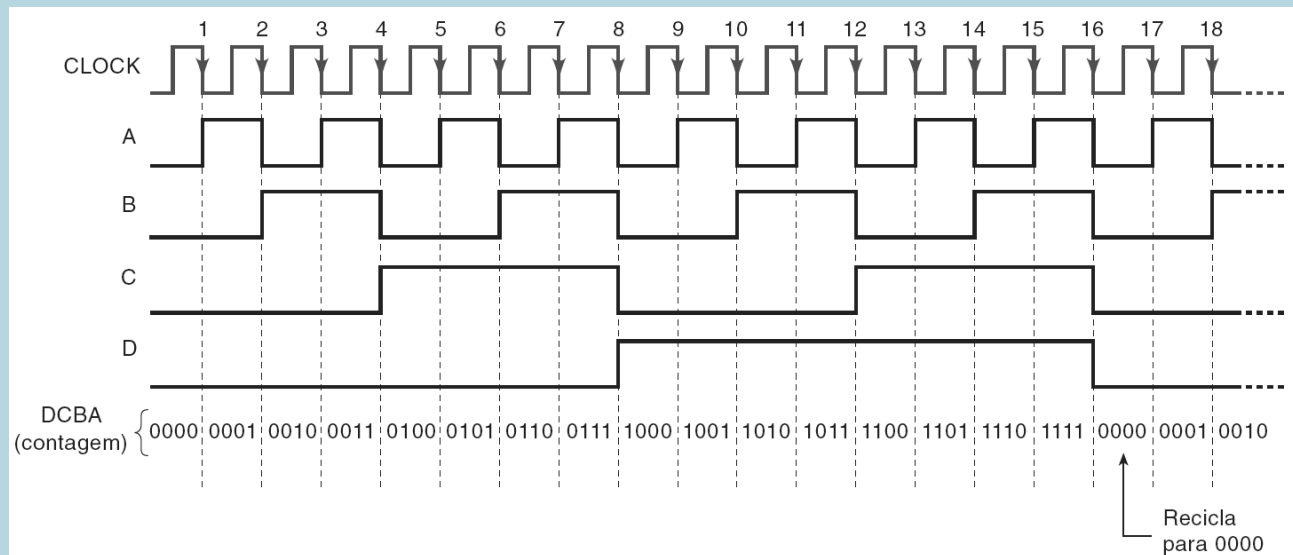


*Supõe-se que todas as entradas J e K sejam 1.

7.1 Contadores assíncronos

Revisão de operações de quatro bits

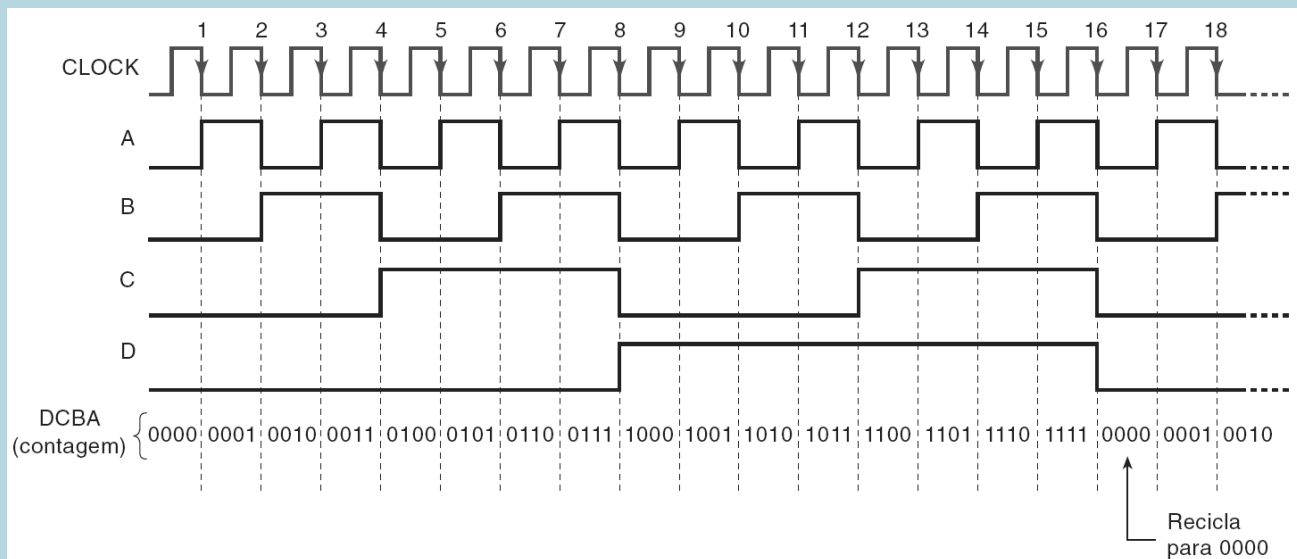
- Após a borda de descida do 15º pulso do clock, os flip-flops do contador estão na condição 1111.



Um contador assíncrono não muda em sincronismo exato com o relógio.

7.1 Contadores assíncronos

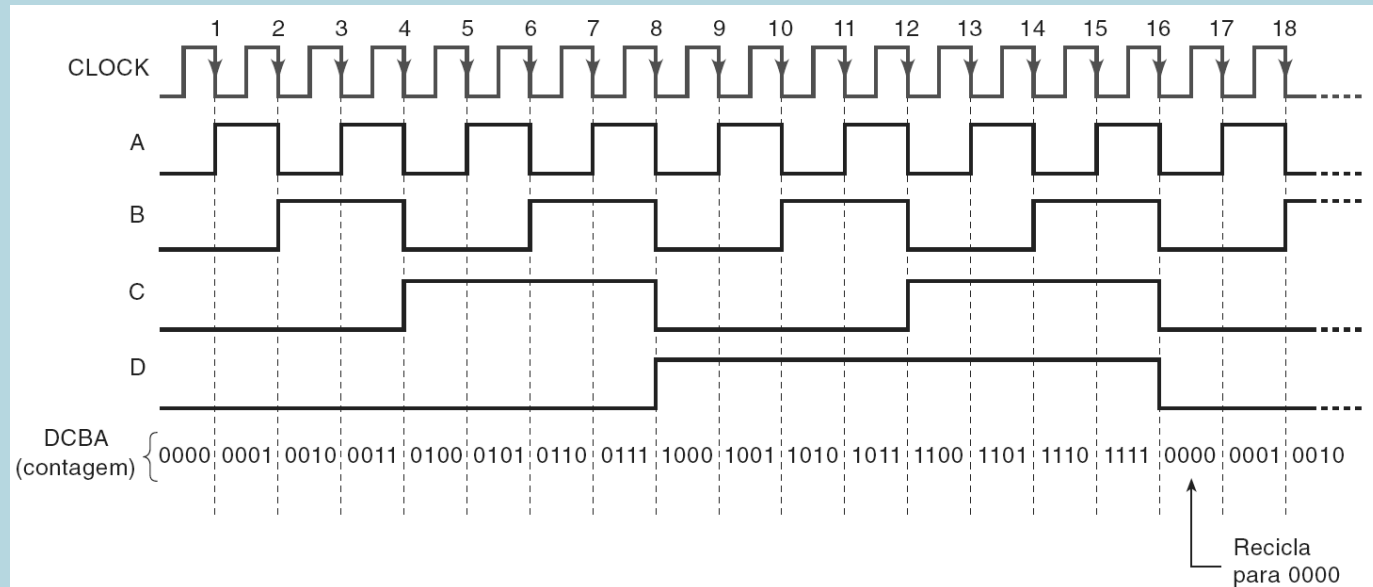
- O número MOD é igual ao número de estados pelos quais o contador passa antes da reciclagem.



Adicionando FFs o número MOD irá aumentar.

7.1 Contadores assíncronos

- A divisão de frequência - cada FF terá uma frequência de saída de metade da entrada.



A frequência de saída do último contador FF será a frequência do clock dividida pelo MOD do contador.

7.1 Contadores assíncronos

- Os esquemas normalmente são desenhados da esquerda para a direita.

Os contadores são desenhados da direita para a esquerda. Com isso, os MSBs e LSBs aparecem nas posições apropriadas.

7.2 Atraso de propagação em contadores assíncronos

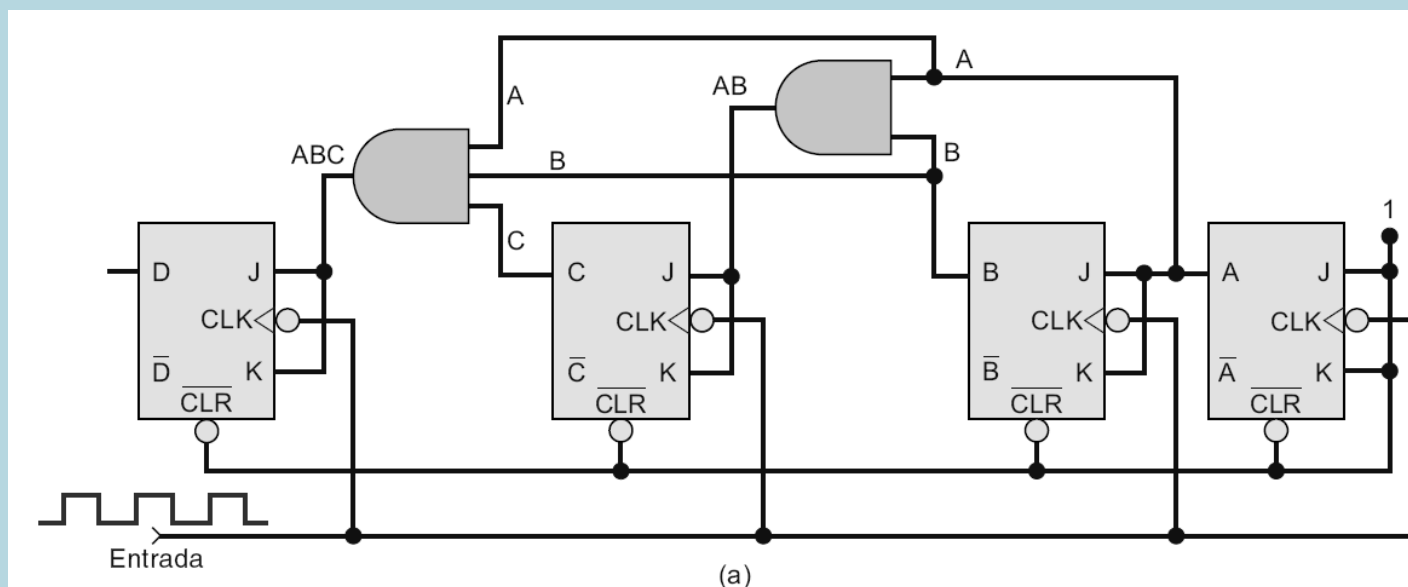
- Contadores ondulantes são o tipo mais simples de contadores binários, visto que requerem poucos componentes para que uma determinada operação seja produzida.
- O atraso de propagação cumulativo pode causar problemas em altas frequências.
- Se o período entre os pulsos de entrada é maior do que o atraso de propagação total do contador, os problemas podem ser evitados.
- Para o funcionamento adequado: $T_{\text{clock}} \geq N \times t_{\text{pd}}$
Frequência máxima: $F_{\text{max}} = 1/N \times t_{\text{pd}}$

7.2 Atraso de propagação em contadores assíncronos

- Os contadores assíncronos não são muito úteis em altas frequências, especialmente para os contadores com grande número de bits.
- Padrões de contagem errados podem gerar **falhas**.
- Sinais produzidos por sistemas usando contadores assíncronos.

7.3 Contadores síncronos (Paralelos)

- Cada *FF* tem entradas J e K que são ALTAS apenas quando as saídas de todos os *FFs* de ordem inferior forem ALTAS.



O atraso de propagação total será o mesmo para qualquer número de *FFs*.

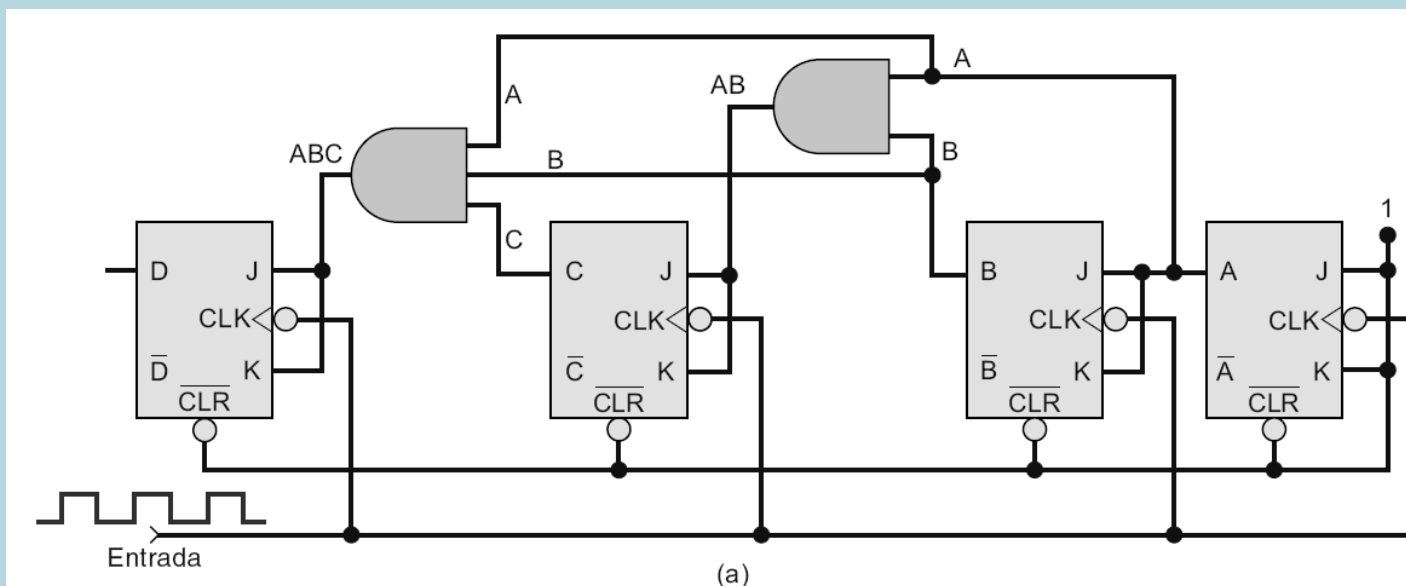
7.3 Contadores síncronos (Paralelos)

- Para este circuito contar corretamente em determinada borda de descida do clock, apenas aqueles *FFs* que supostamente devem comutar nessa borda de descida do clock devem ter $J = K = 1$, quando ocorrer essa transição.

Contagem	D	C	B	A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1
0	0	0	0	0
.
.
.	.	etc.	.	.

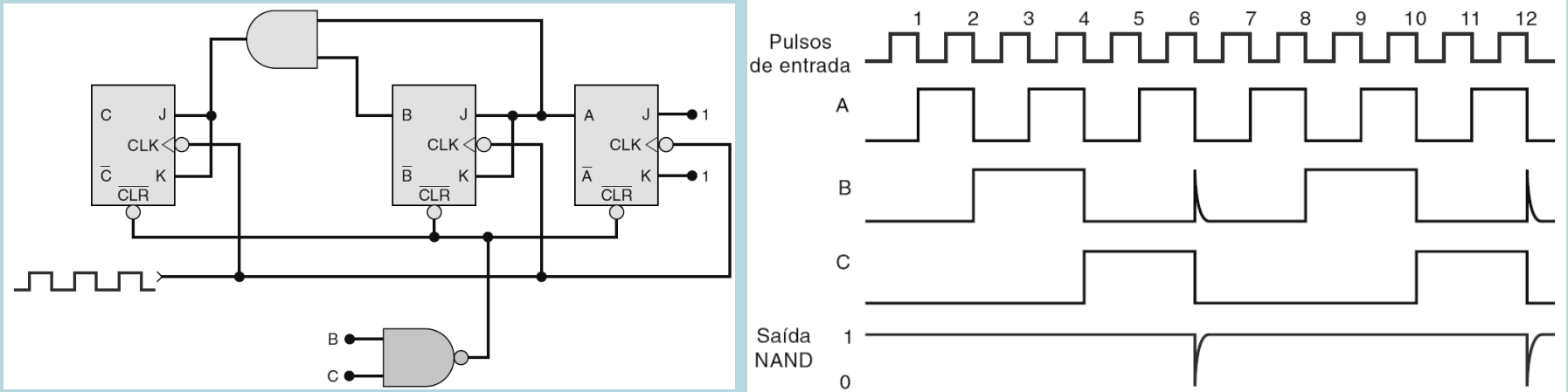
7.4 Contadores de Módulo $< 2^N$

- O contador síncrono básico mostrado é limitado aos números MOD iguais a potências de 2.



7.4 Contadores de Módulo $< 2^N$

- O contador básico pode ser modificado para produzir números MOD menor que 2^N . Faz com que o contador *salte estados* que normalmente são parte da sequência de contagem.



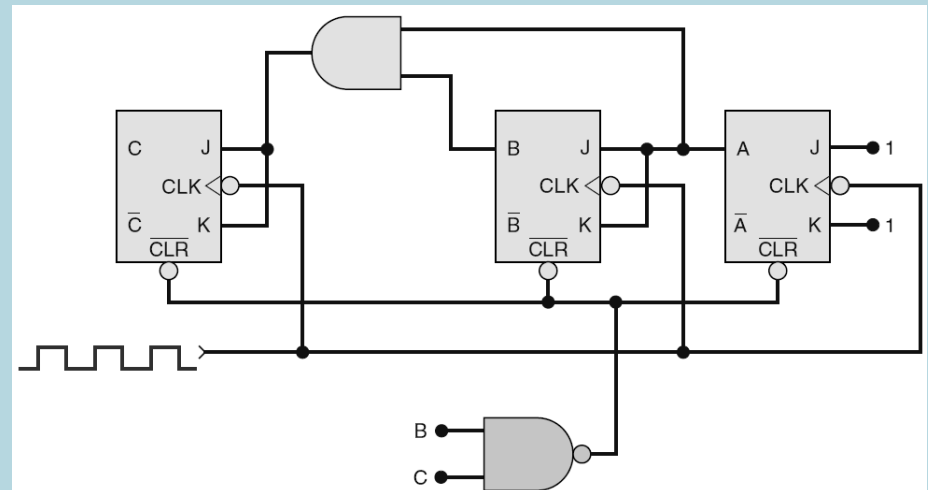
Contadores MOD-6 produzidos a partir de um contador MOD-8 quando ocorre uma contagem de seis (110).

7.4 Contadores de Módulo $< 2^N$

- Alterar o número MOD.

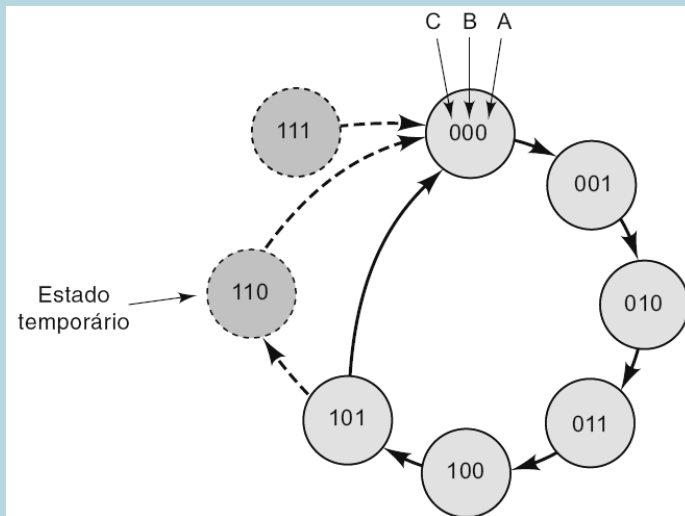
Para tal, encontre o menor MOD necessário para que 2^N seja menor ou igual a exigência e conecte uma porta NAND para as entradas assíncronas de todos os *FFs*.

Determinar quais são *FFs* ALTO na contagem desejada e conectar as saídas desses *FFs* às entradas NAND.



7.4 Contadores de Módulo $< 2^N$

Diagrama de transição de estados para o contador MOD-6

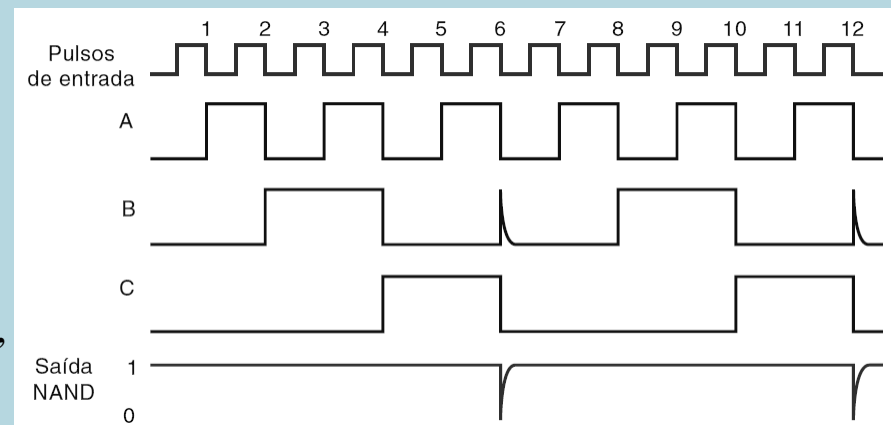


Cada círculo representa um dos possíveis estados do contador .

As setas indicam uma mudança de estado para outro, em resposta a um pulso de clock de entrada.

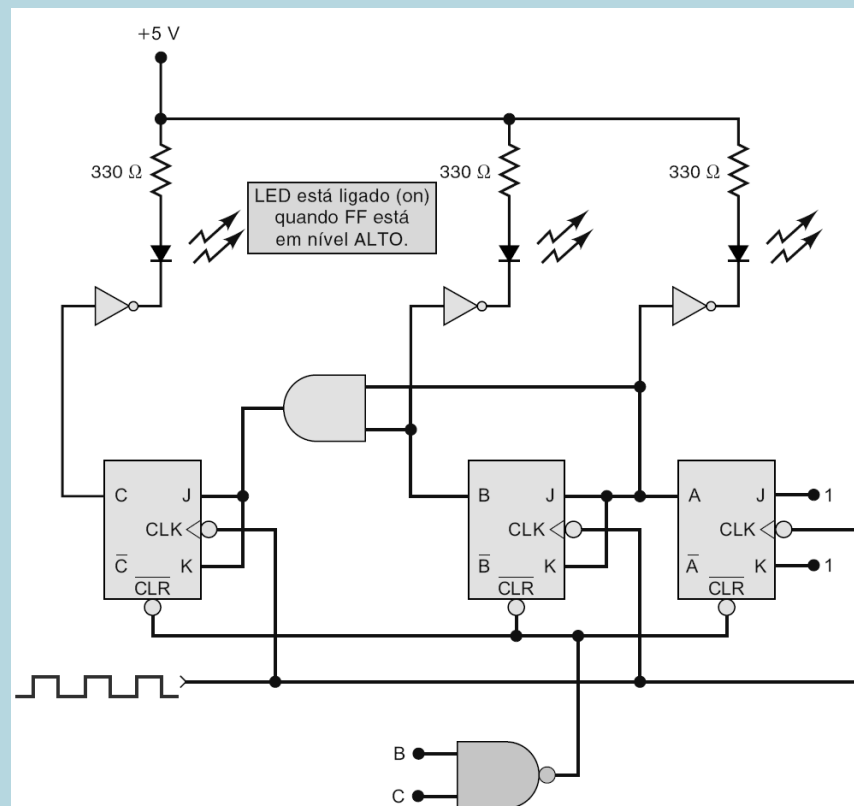
Não há seta para o estado 111 pois o contador não pode avançar esse estado.

O estado 111 pode ocorrer em power-up quando o FFs pode cair em qualquer estado.



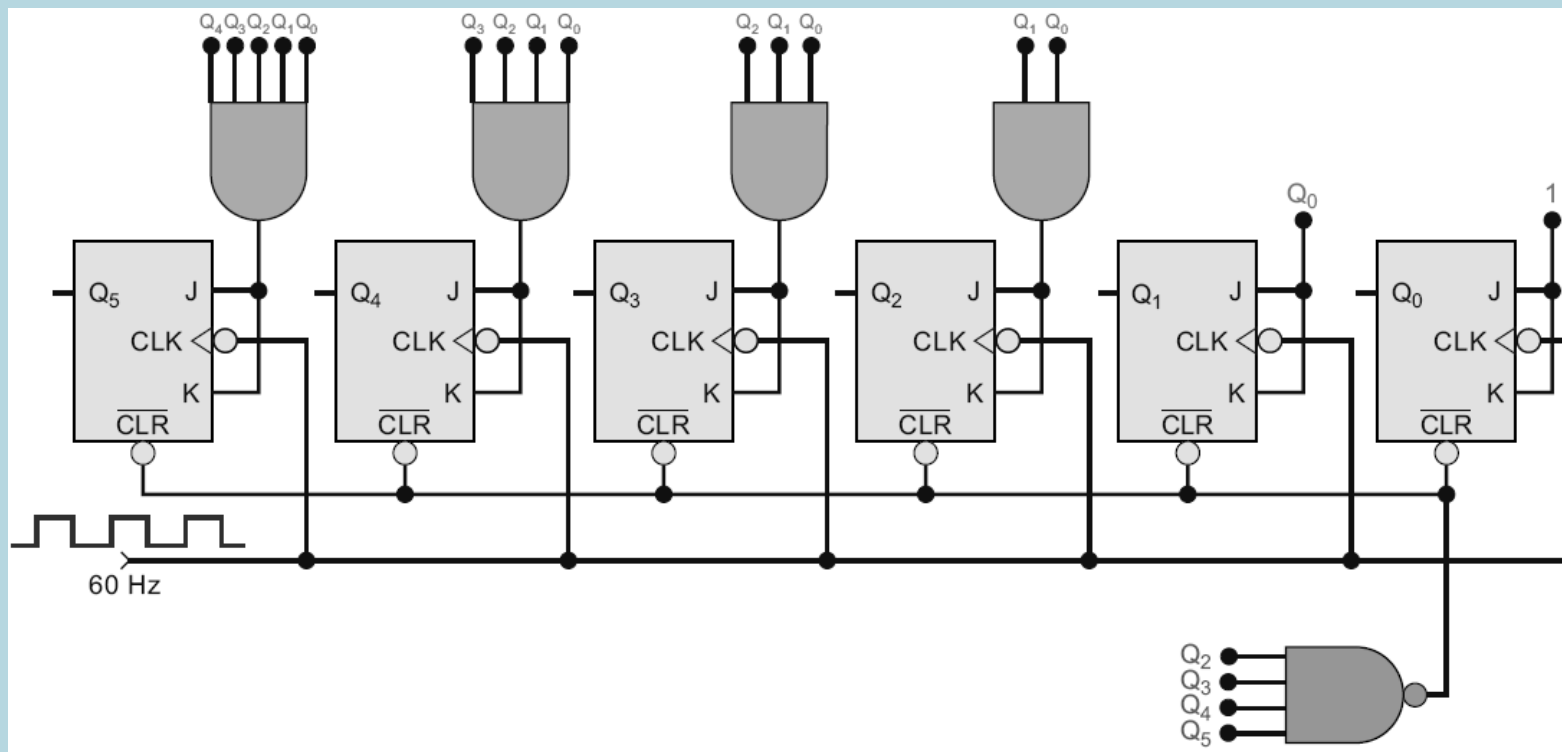
7.4 Contadores de Módulo $< 2^N$

- A exibição dos estados do contador são muitas vezes realizados com um LED.



7.4 Contadores com o número de MOD $< 2^N$

Contador de módulo 60.

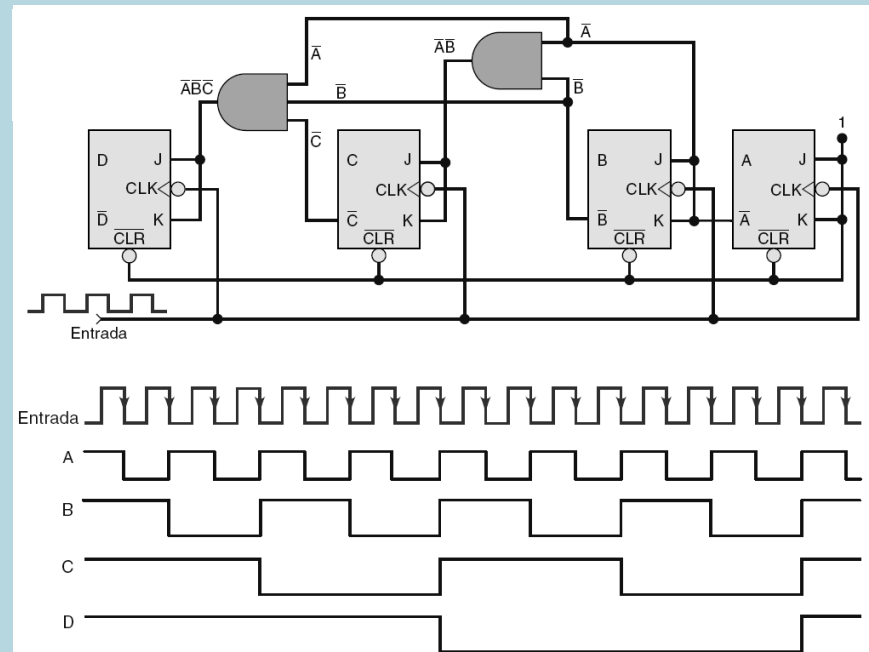


7.5 Contadores síncronos decrescentes e crescentes/ decrescentes

- Um contador síncrono decrescente é construído de forma semelhante a um contador crescente.

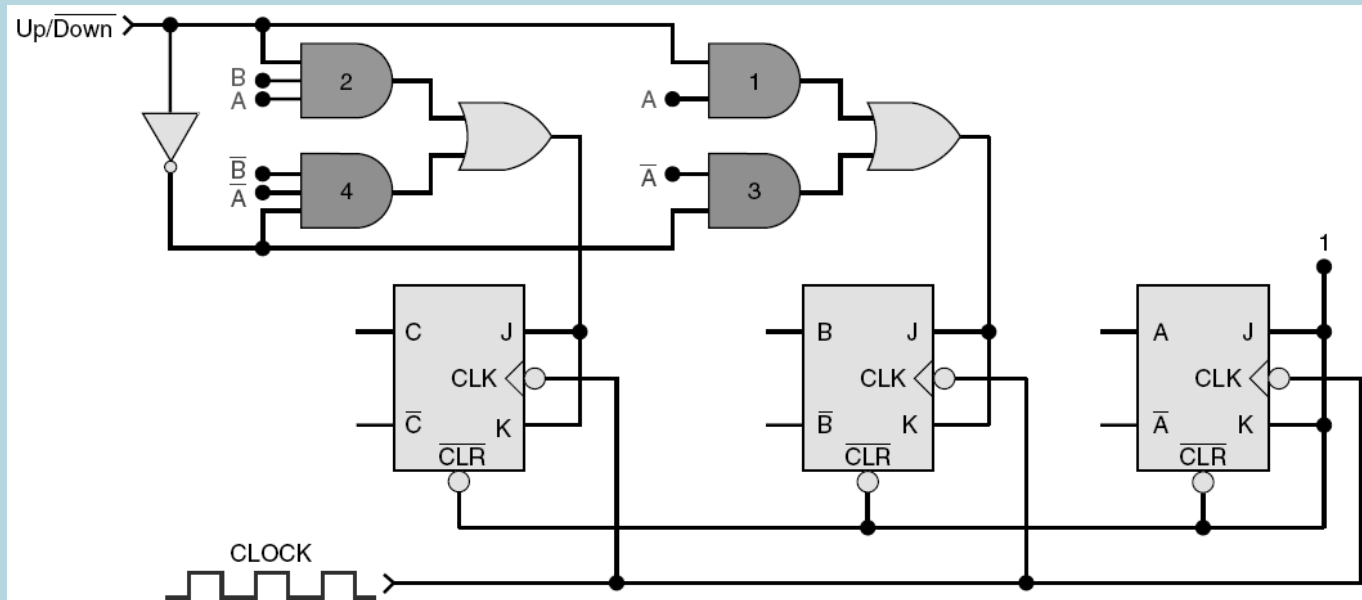
Ele usa as saídas *FFs* invertidas para controlar as entradas *J* e *K* de ordem mais alta.

Síncrono, MOD-16, contadores decrescentes e ondas de saída.



7.5 Contadores síncronos decrescentes e crescentes/ decrescentes

- Em um contador paralelo crescente/ decrescente, a entrada de controle controla se as entradas J e K dos FFs seguintes serão acionadas pelas saídas normais ou pelas saídas invertidas dos FFs .



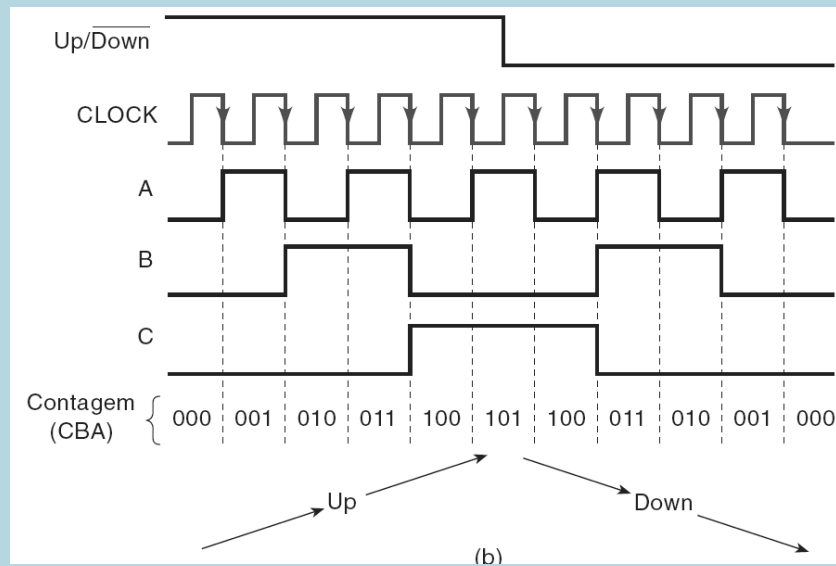
7.5 Contadores síncronos decrescentes e crescentes/ decrescentes

Nos primeiros cinco pulsos de clock

- O contador conta **Up**. $\overline{\text{Up/Down}} = 1$.

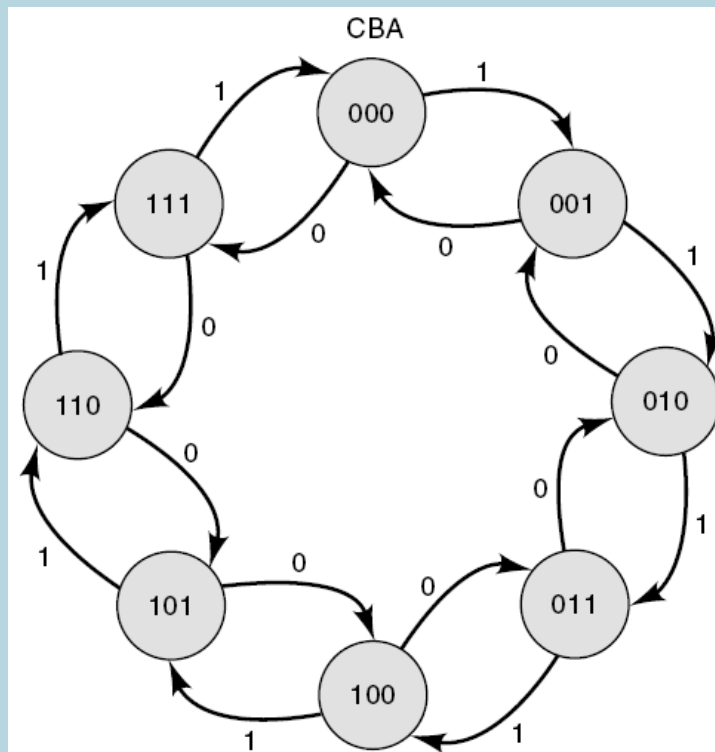
Nos últimos cinco pulsos

- O contador conta **Down**. $\overline{\text{Up/Down}} = 0$.



7.5 Contadores síncronos decrescentes e crescentes/ decrescentes

- As setas representam transições de estado que ocorrem na borda de descida do sinal de clock. Há duas setas partindo de cada círculo de estado.

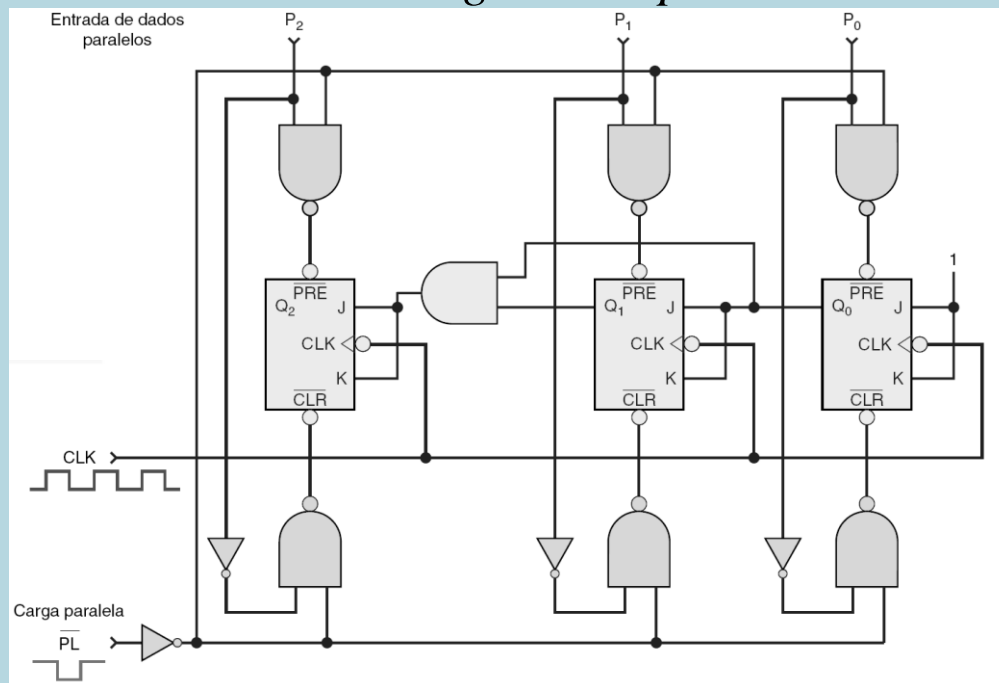


O próximo estado para esse contador depende do nível lógico aplicado na entrada de controle.

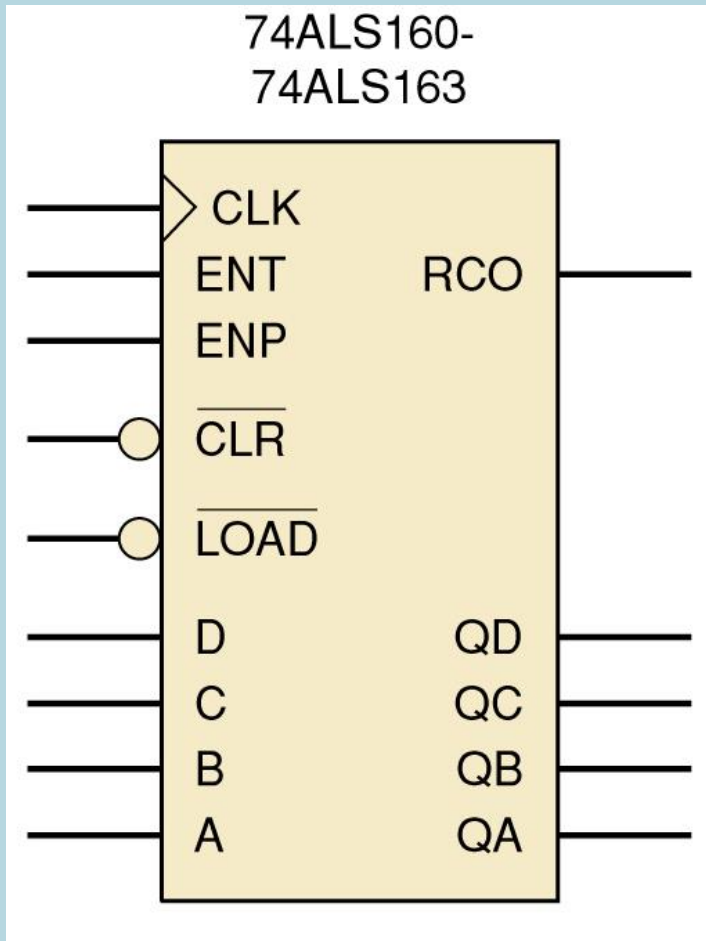
7.6 Contadores com Carga Paralela

- Um contador pré-configurável pode ser definido para qualquer ponto inicial desejado - síncrono ou assíncrono.
- Pré-configurar também é chamado de *carregamento paralelo* do contador.

Contador síncrono
com carga paralela
assíncrona.



7.7 Circuitos integrados de contadores síncronos



O contador contém quatro *FFs*.

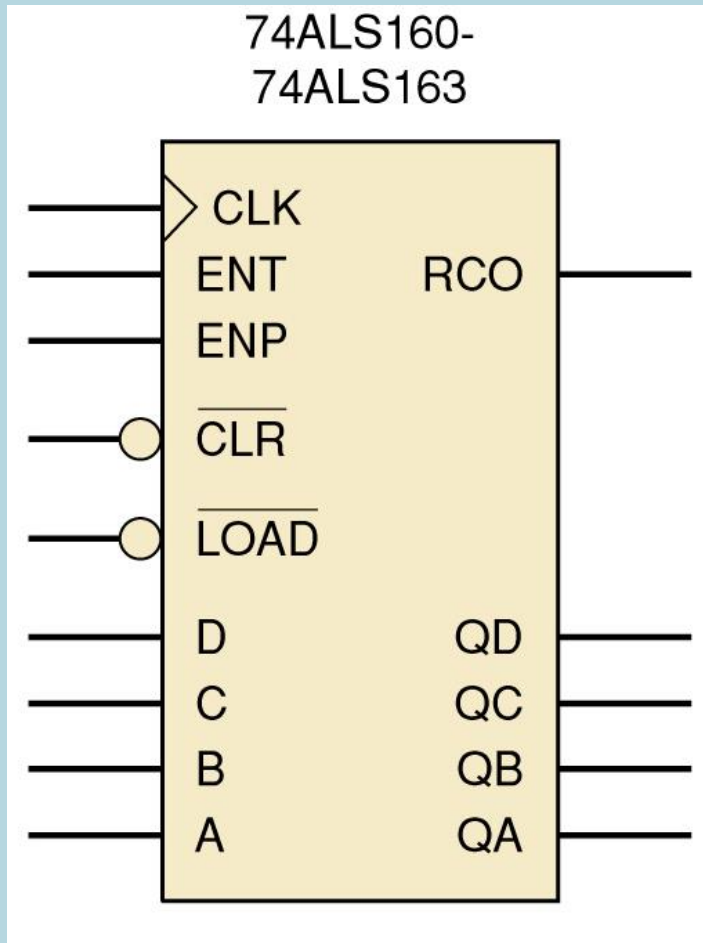
Os *FFs* são disparados por uma borda de subida na entrada CLK.

Função CLEAR - entrada ativa em nível baixo assíncrona.

Pode ser pré-configurado para qualquer valor de aplicação de uma entrada LOAD ativa em baixo para *A, B, C, D* e entradas.

Número do componente	Módulos
74ALS160	10
74ALS161	16
74ALS162	10
74ALS163	16

7.7 Circuitos integrados de contadores síncronos



Esta série de chips de contadores CI tem mais um pino de saída, o RCO, uma saída para detectar (decodificar) o último ou o estado terminal do contador.

É muito útil na ligação entre dois ou mais chips contadores em um arranjo de multiplataformas para criação de contadores maiores.

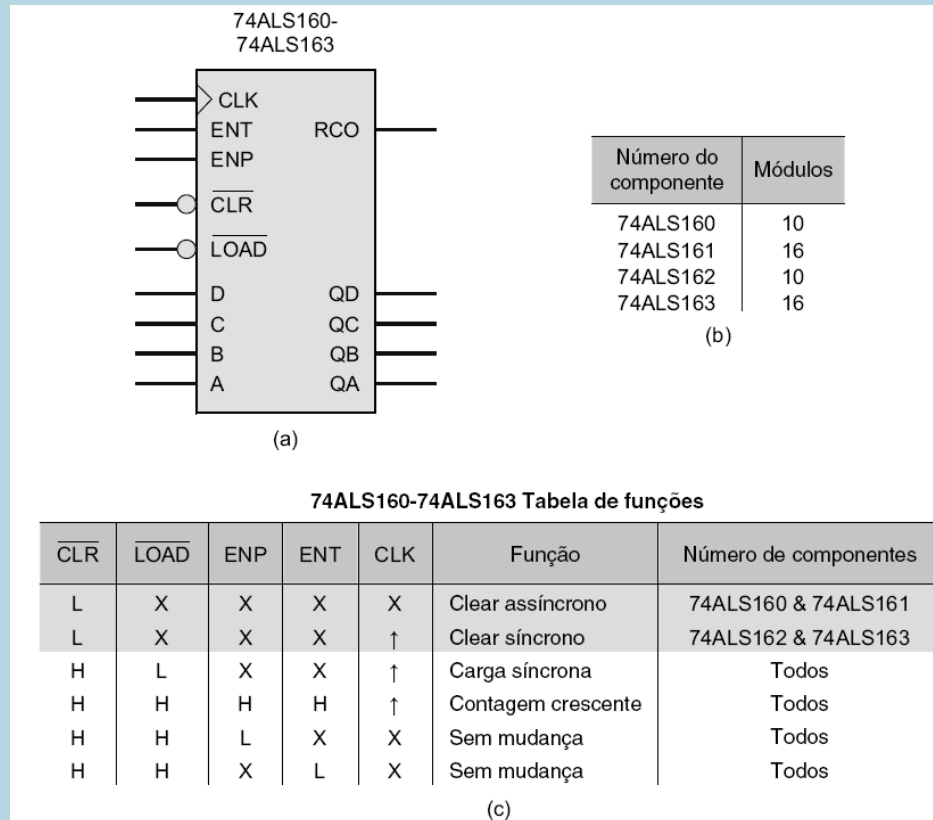
7.7 Circuitos integrados de contadores síncronos

- TTL 74ALS160 - 74ALS163 Tabela de funções

$\overline{\text{CLR}}$	$\overline{\text{LOAD}}$	ENP	ENT	CLK	Função	Número de componentes
L	X	X	X	X	Clear assíncrono	74ALS160 & 74ALS161
L	X	X	X	↑	Clear síncrono	74ALS162 & 74ALS163
H	L	X	X	↑	Carga síncrona	Todos
H	H	H	H	↑	Contagem crescente	Todos
H	H	L	X	X	Sem mudança	Todos
H	H	X	L	X	Sem mudança	Todos

7.7 Circuitos integrados de contadores síncronos

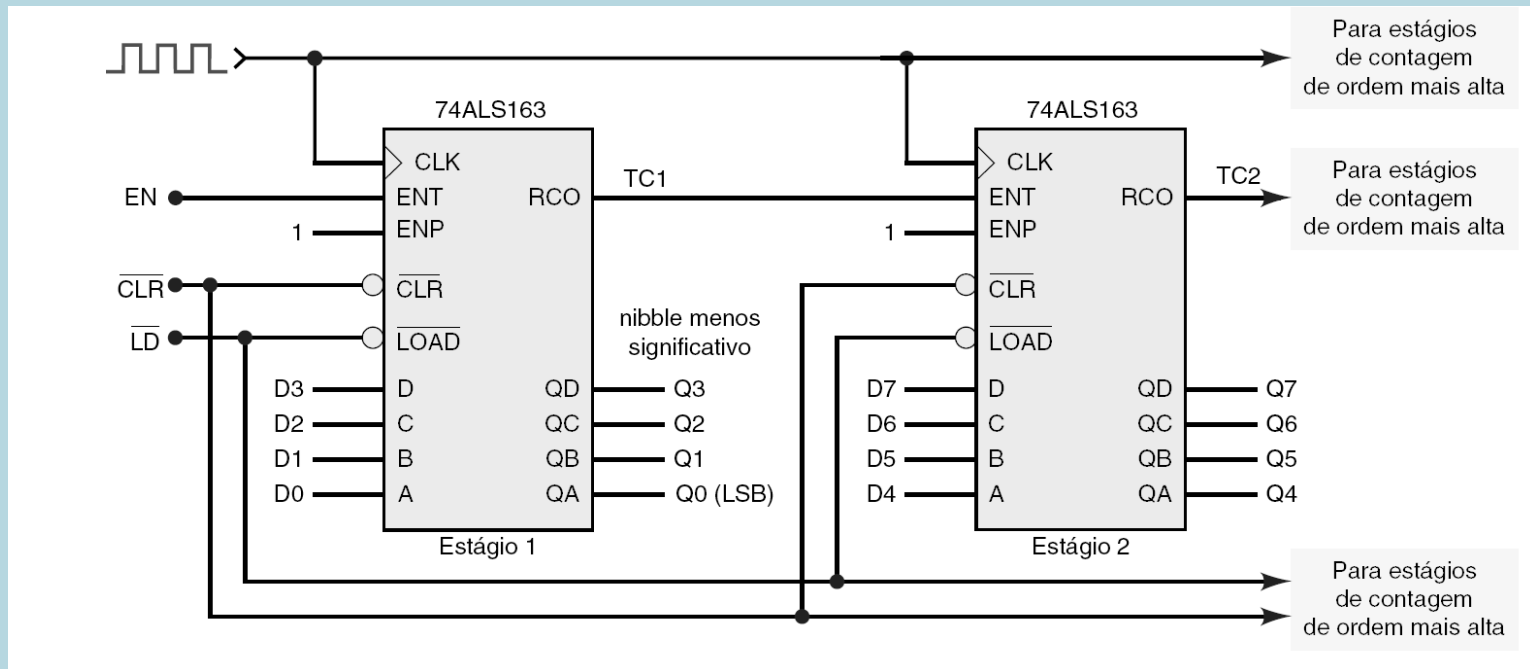
- 74ALS190 e 74ALS191 CIs, contadores de quatro bits.



7.7 Circuitos integrados de contadores síncronos

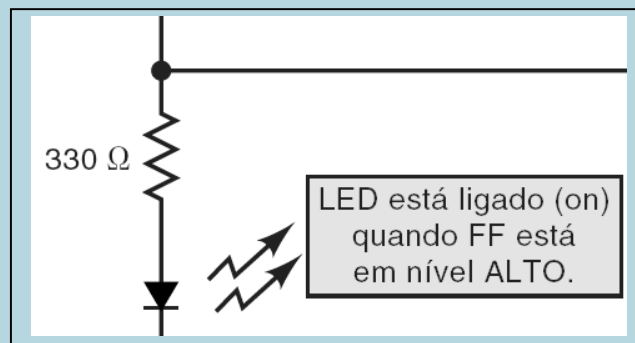
- Muitos contadores padrão CI foram concebidos para tornar mais fácil conectar múltiplos chips e criar circuitos com uma gama maior de contagem.

Um arranjo de **múltiplos estágios** ou em **cascata**.



7.8 Decodificando um contador

- Contadores digitais são frequentemente utilizados em aplicações nas quais a contagem representada pelo estado dos *FFs*, de alguma forma, deve ser determinada ou exibida.
- Um dos meios mais simples para exibir o conteúdo de um contador é conectar a saída *FF* a um LED.

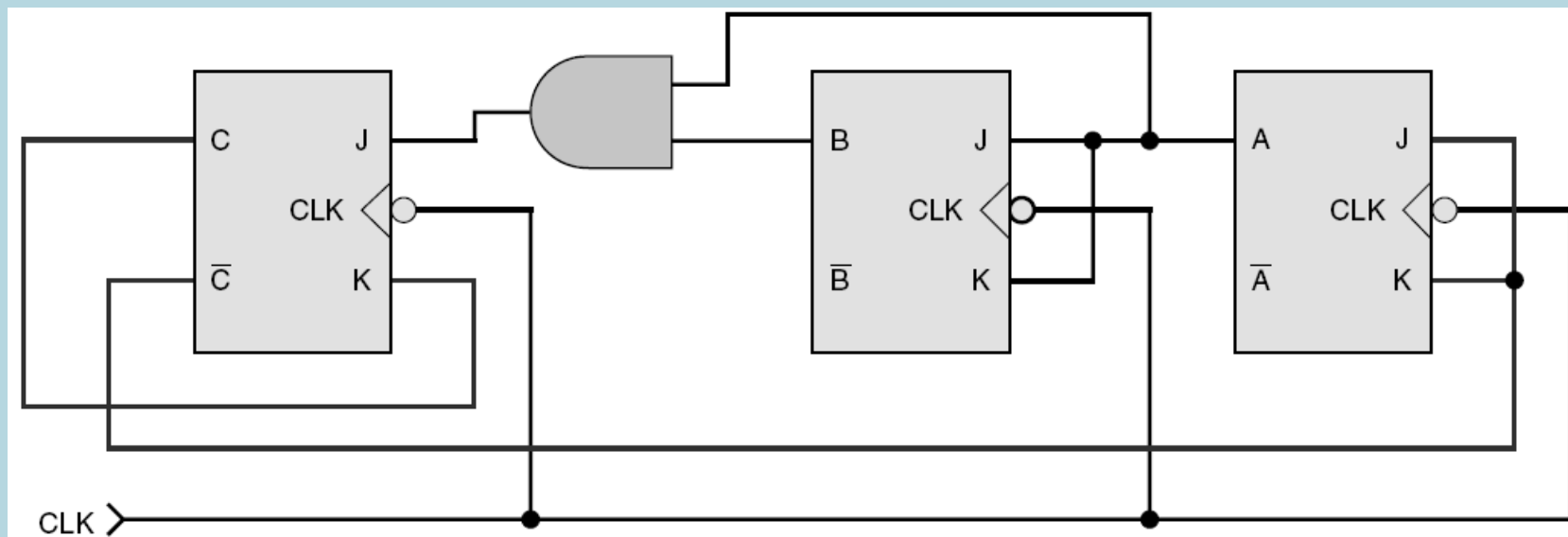


O método que utiliza LEDs para visualização se torna inconveniente quando o tamanho (número de bits) do contador aumenta. É muito mais difícil decodificar a exibição mentalmente.

7.8 Decodificando um contador

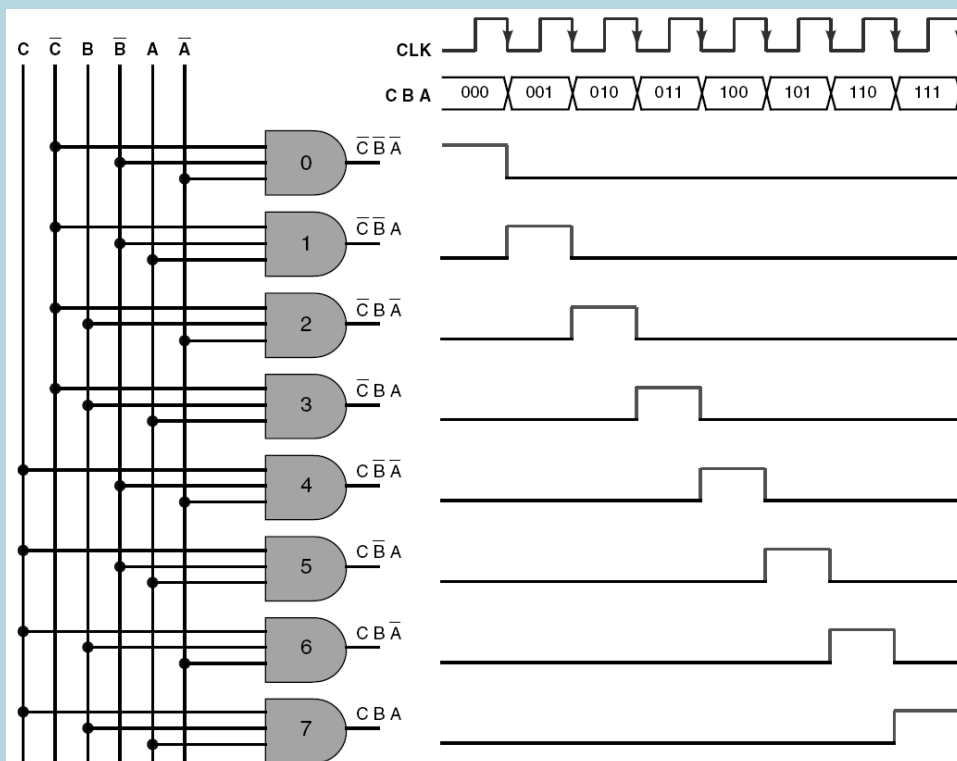
- A decodificação é a conversão de uma saída binária para um valor decimal, uma forma imediatamente reconhecida.

O decodificador ativo em nível ALTO mostrado pode ser usado para acender um LED, representando cada número decimal.



7.8 Decodificando um contador

- **Decodificação** é a conversão de uma saída binária para um valor decimal, uma forma reconhecida imediatamente.



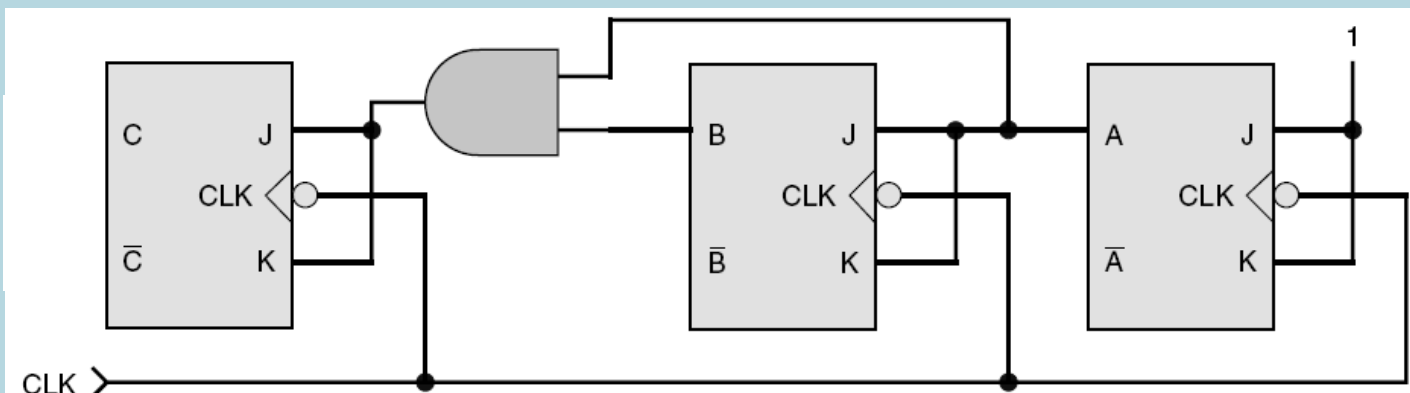
MOD-8

Saída do decodificador ativo em nível ALTO.

7.8 Decodificando um contador

O decodificador ativo em nível ALTO mostrado pode ser mudado para um tipo de ativo em nível BAIXO.

A decodificação ativo em nível BAIXO é obtida substituindo as portas AND por portas NAND.



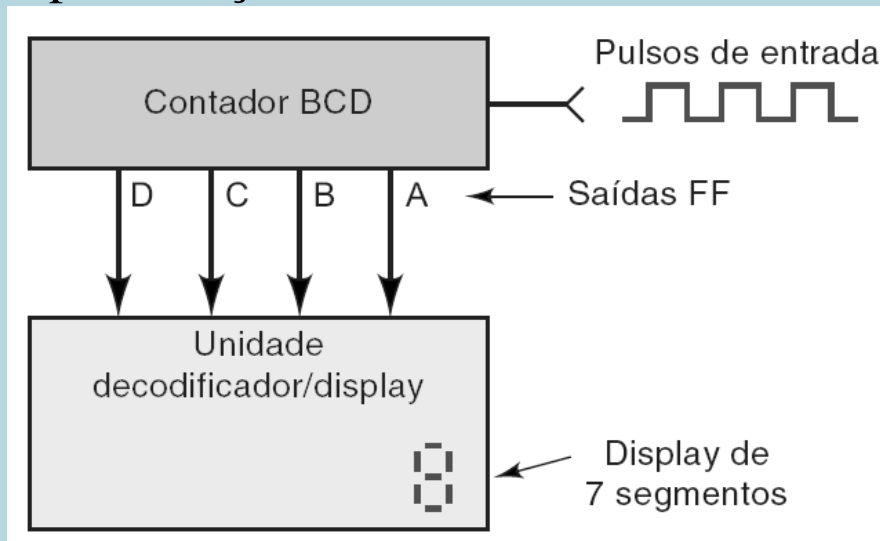
As saídas do decodificador agora produzem um sinal normalmente alto, que se torna baixo apenas quando ocorre o número que está sendo decodificado.

7.8 Decodificando um contador

- O contador BCD tem 10 estados, decodificados para fornecer 10 saídas correspondentes aos dígitos decimais de 0 – 9.

Representado pelos estados dos *FFs* do contador.

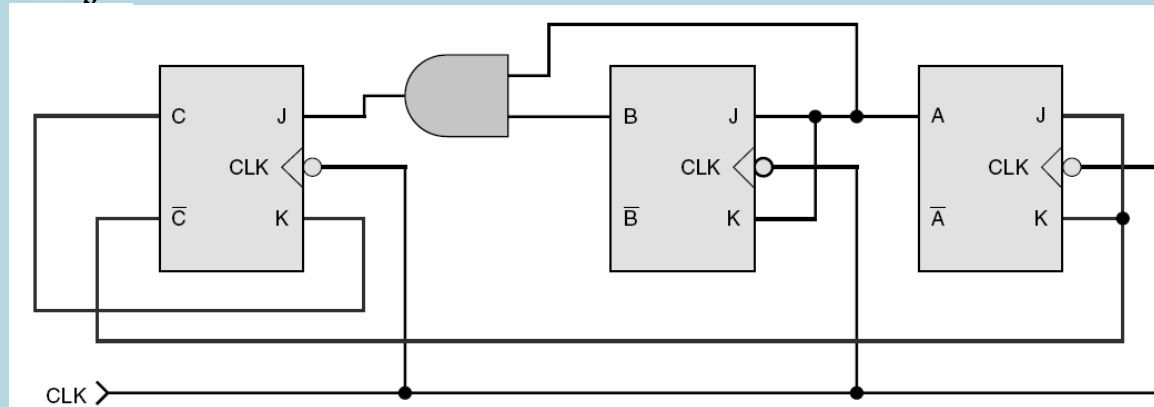
As 10 saídas podem controlar 10 LEDs, indicadores individuais para uma apresentação visual.



Em vez de 10 LEDs separados, os contadores BCD geralmente têm sua contagem exibida em um único dispositivo de exposição.

7.9 Analisando contadores síncronos

Contadores síncronos podem ser customizados para gerar qualquer sequência de contagem desejada.



Contador síncrono
up.

$$\begin{aligned} J_C &= A \cdot B \\ K_C &= C \\ J_B &= K_B = A \\ J_A &= K_A = \bar{C} \end{aligned}$$

Controle de
expressões de entrada

7.9 Analisando contadores síncronos

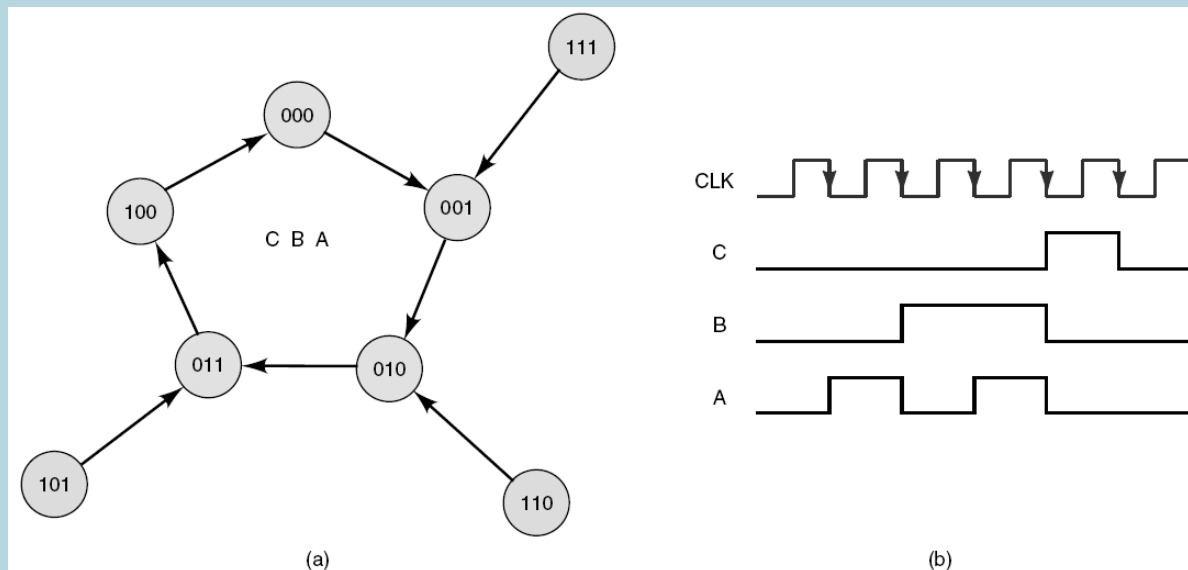
- Analisar projetos de contadores desse tipo prevendo entradas de controle FF para cada estado do contador.

A tabela de **estado presente/ estado futuro** é muito útil para esse fim.

Estado ATUAL			Entradas de controle						PRÓXIMO estado		
C	B	A	J_C	K_C	J_B	K_B	J_A	K_A	C	B	A
0	0	0	0	0	0	0	1	1	0	0	1
0	0	1	0	0	1	1	1	1	0	1	0
0	1	0	0	0	0	0	1	1	0	1	1
0	1	1	1	0	1	1	1	1	1	0	0
1	0	0	0	1	0	0	0	0	0	0	0
1	0	1	0	1	1	1	0	0	0	1	1
1	1	0	0	1	0	0	0	0	0	1	0
1	1	1	1	1	1	1	0	0	0	0	1

7.9 Analisando contadores síncronos

Onda e diagrama de transição de estados de contador síncrono.

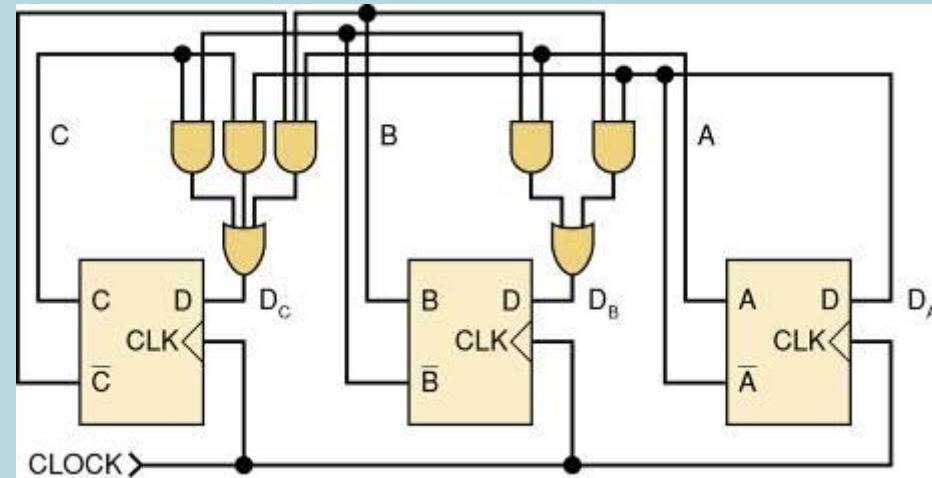


Informação destacada
indica que esse projeto de contador é de
autocorreção.

7-9 Analisando contadores síncronos

Contador síncrono construído com *FFs* do tipo D.

O controle de circuito será tipicamente mais complexo do que um contador *J-K* equivalente.



Estado ATUAL			Entradas de controle			PRÓXIMO estado		
C	B	A	D_C	D_B	D_A	C	B	A
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	0
0	1	0	0	1	1	0	1	1
0	1	1	1	0	0	1	0	0
1	0	0	1	0	1	1	0	1
1	0	1	1	1	0	1	1	0
1	1	0	1	1	1	1	1	1
1	1	1	0	0	0	0	0	0

O número de entradas síncronas para controlar é reduzido pela metade.

7-10 Projeto de Contadores Síncronos

- Métodos comuns de projeto usando flip-flops J - K em uma configuração de contadores síncronos.

Determinar o número desejado de bits (FFs) e a sequência de contagem desejada.

Desenhar o diagrama de transição de estado mostrando *todos* os estados possíveis, incluindo aqueles que não fazem parte da sequência de contagem desejada.

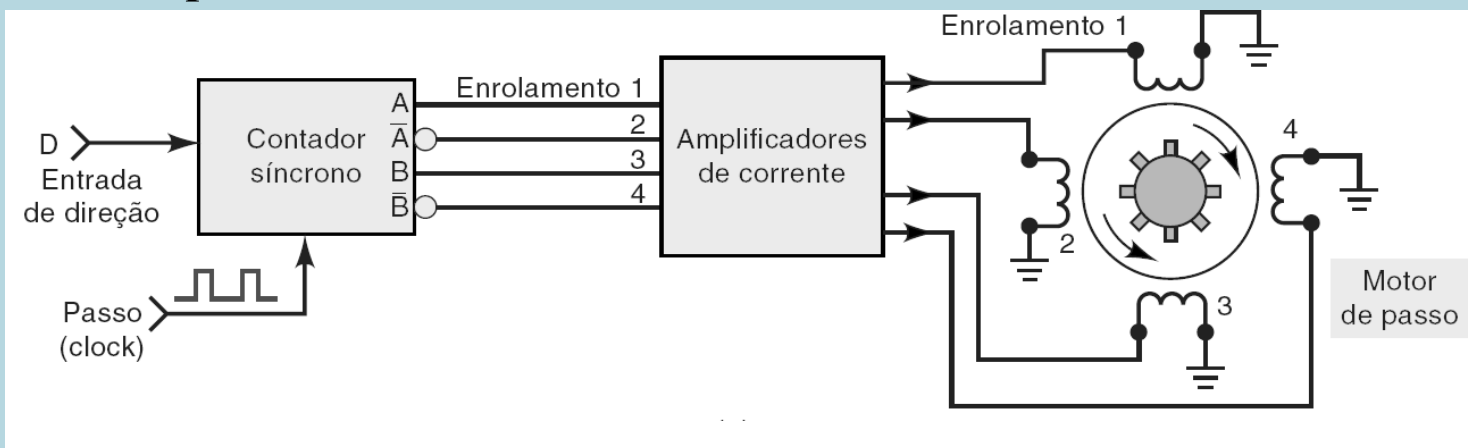
Usar o diagrama de transição de estados para criar uma tabela que liste *todos* os estados ATUAIS e os PRÓXIMOS.

Adicionar uma coluna para cada entrada J e K e indicar os níveis necessários para produzir a transição para o PRÓXIMO estado.

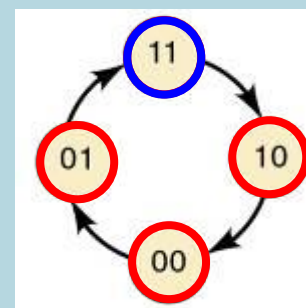
Projetar os circuitos lógicos para gerar os níveis exigidos em cada entrada.

Implementar as expressões finais.

- O controle de um *motor de passo* gira em passos discretos, tipicamente 15 graus por passo, em vez de girar em movimento contínuo.
Uma aplicação prática de projeto de contadores síncronos.
- Reversível, dependendo do nível de saída.

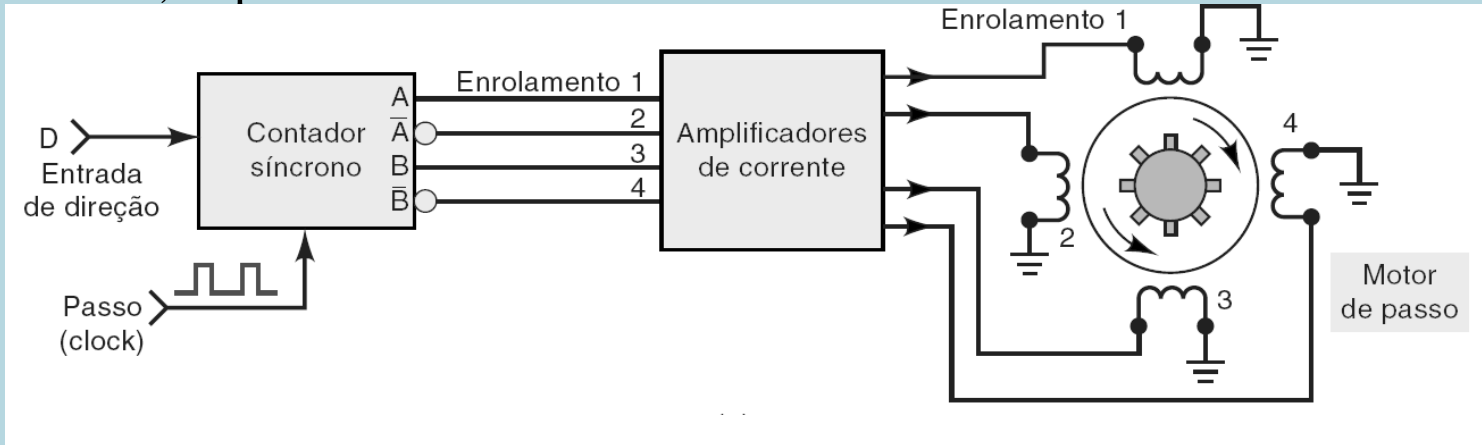


Estado ATUAL			PRÓXIMO estado		Entradas de controle			
D	B	A	B	A	J _B	K _B	J _A	K _A
0	0	0	0	1	0	x	1	x
0	0	1	1	1	1	x	x	0
0	1	0	0	0	x	1	0	x
0	1	1	1	0	x	0	x	1
1	0	0	1	0	1	x	0	x
1	0	1	0	0	0	x	x	1
1	1	0	1	1	x	0	1	x
1	1	1	0	1	x	1	x	0

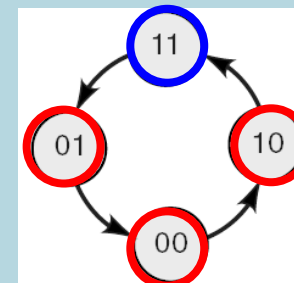


Rotação no sentido horário
D = 0

- O controle de um *motor de passo* gira em passos discretos, tipicamente 15 graus por passo, em vez de girar em movimento contínuo.
Uma aplicação prática de projeto de contadores síncronos.
- Reversível, dependendo do nível de saída.



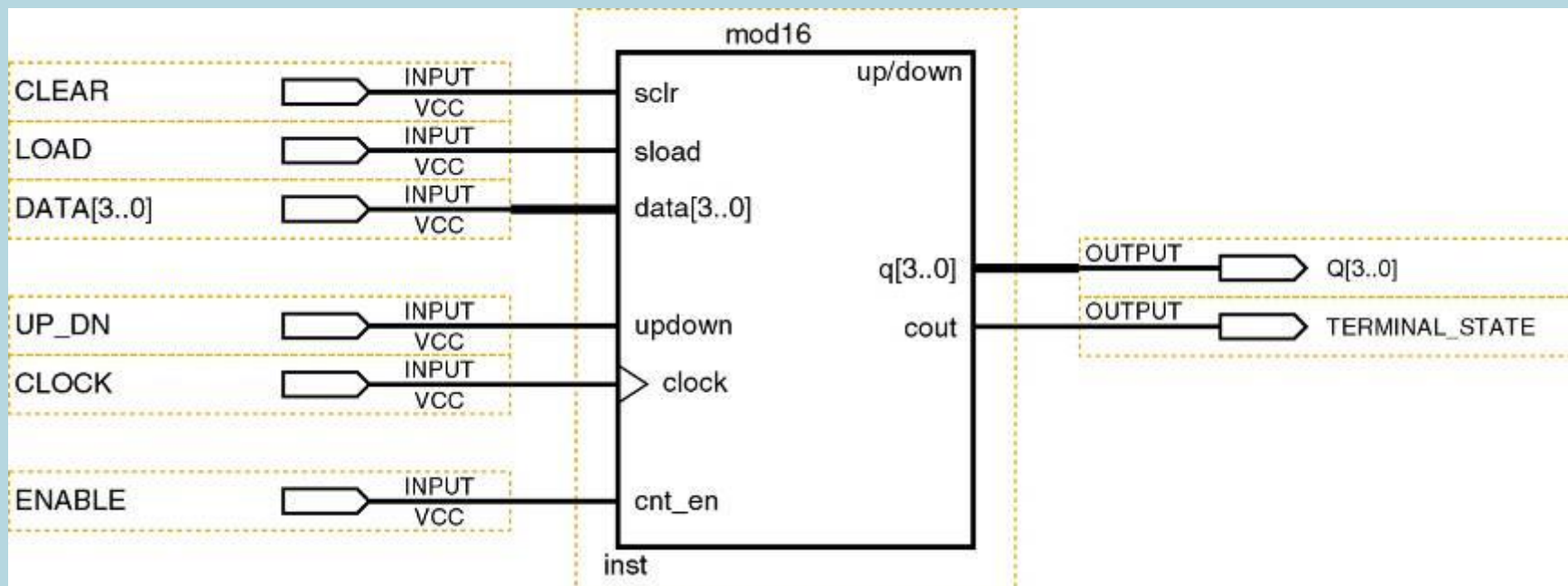
Estado ATUAL			PRÓXIMO estado		Entradas de controle			
D	B	A	B	A	J _B	K _B	J _A	K _A
0	0	0	0	1	0	x	1	x
0	0	1	1	1	1	x	x	0
0	1	0	0	0	x	1	0	x
0	1	1	1	0	x	0	x	1
1	0	0	1	0	1	x	0	x
1	0	1	0	0	0	x	x	1
1	1	0	1	1	x	0	1	x
1	1	1	0	1	x	1	x	0



Rotação no sentido anti-horário
D = 1

7-11 Funções de biblioteca Altera para contadores

- O Editor de Bloco Quartus pode programar um PLD com qualquer contador usando flip-flops e portas .
- Essas macrofunções estão na biblioteca maxplus2.

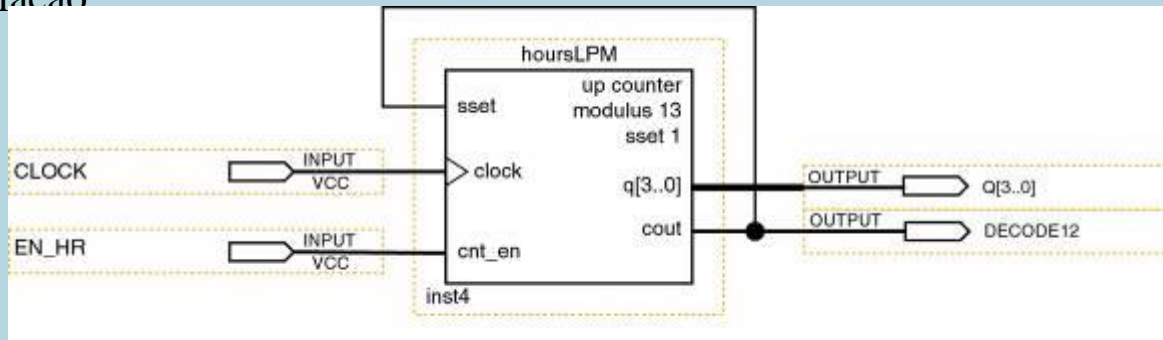


Contador de módulo 16 completo.

7-11 Funções de biblioteca Altera para contadores

Contador de horas de um relógio digital . Diagrama de blocos. Configurações MegaWizard .

Resultados da simulação



How wide should the 'q' output bus be? bits

What should the counter direction be?

Up only

Down only

Create an 'updown' input port to allow me to do both
[1 counts up; 0 counts down]

Which type of counter do you want?

Plain binary

Modulus, with a count modulus of

Do you want any optional additional ports?

Clock Enable Carry-in

Count Enable Carry-out

Do you want any optional inputs?

Synchronous inputs

Clear

Load

Set

Set to all 1's

Set to

Asynchronous inputs

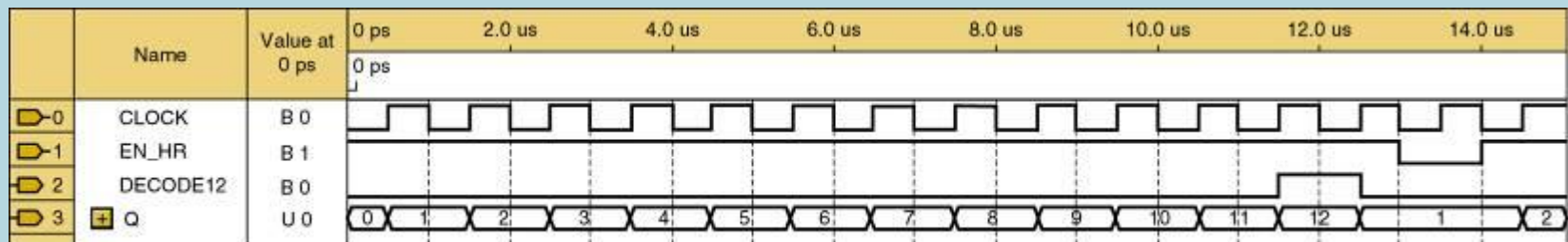
Clear

Load

Set

Set to all 1's

Set to 0



7-12 Contadores básicos usando HDL

- Métodos para descrever circuitos de contador usando HDL utilizam principalmente técnicas síncronas.

Todos os flip-flops são atualizados simultaneamente em resposta ao evento do clock.

Todos os bits em uma sequência de contagem vão do seu estado ATUAL para o estado prescrito como PRÓXIMO, simultaneamente.

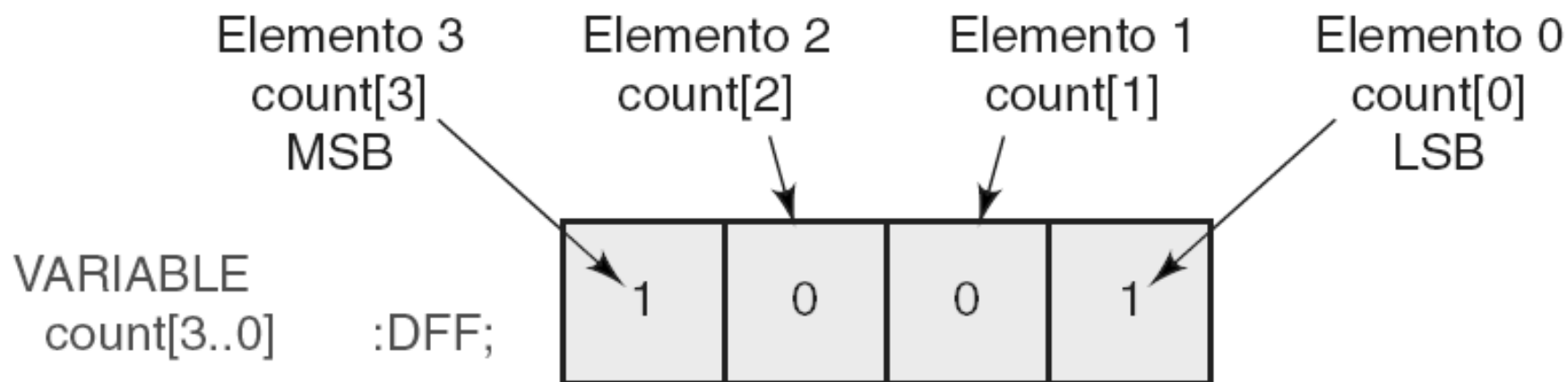
- Métodos de descrição de transição de estado

A tabela de estado ATUAL/ PRÓXIMO é equivalente a tabela-verdade.

7-12 Contadores básicos usando HDL

- Descrições de estado em AHDL

O primeiro passo importante é declarar os pinos de saída do contador devidamente, como um vetor de bits.



- Descrições de estado em AHDL

```
1  SUBDESIGN fig7_43
2  (
3      clock      :INPUT;
4      q[2..0]    :OUTPUT;    -- declara matriz de 3 bits de saída
5  )
6  VARIABLE
7      count[2..0] :DFF;      -- declara um registrador de flip flops D.
8
9  BEGIN
10     count[].clk = clock; -- conecta todos os clocks a uma fonte síncrona
11     IF count[].q < 4 THEN -- observe que count[] é o mesmo que count[].q
12         count[].d = count[].q + 1; -- incrementa o valor atual em 1
13     ELSE count[].d = 0; --recicla para 0; força os estados não usados a 0
14     END IF;
15     q[] = count[].q; -- transfere conteúdo do registrador para as saídas
16 END;
```

O estado atual do contador é avaliado (*count []. q*) na linha 11, e se ele é menor do que o valor de contagem mais alto, ele usa a descrição *count[].d count.q+ 1* (linha 12).

- Descrições de Estado em AHDL

```
1  SUBDESIGN fig7_43
2  (
3      clock      :INPUT;
4      q[2..0]    :OUTPUT;    -- declara matriz de 3 bits de saída
5  )
6  VARIABLE
7      count[2..0] :DFF;      -- declara um registrador de flip flops D.
8
9  BEGIN
10     count[].clk = clock; -- conecta todos os clocks a uma fonte síncrona
11     IF count[].q < 4 THEN -- observe que count[] é o mesmo que count[].q
12         count[].d = count[].q + 1; -- incrementa o valor atual em 1
13     ELSE count[].d = 0; --recicla para 0; força os estados não usados a 0
14     END IF;
15     q[] = count[].q; -- transfere conteúdo do registrador para as saídas
16 END;
```

Quando o contador atinge o estado desejado mais alto (Ou superior), a declaração do teste *IF* será falsa, resultando em um valor de entrada de **PRÓXIMO** estado igual a zero, que recicla o contador.

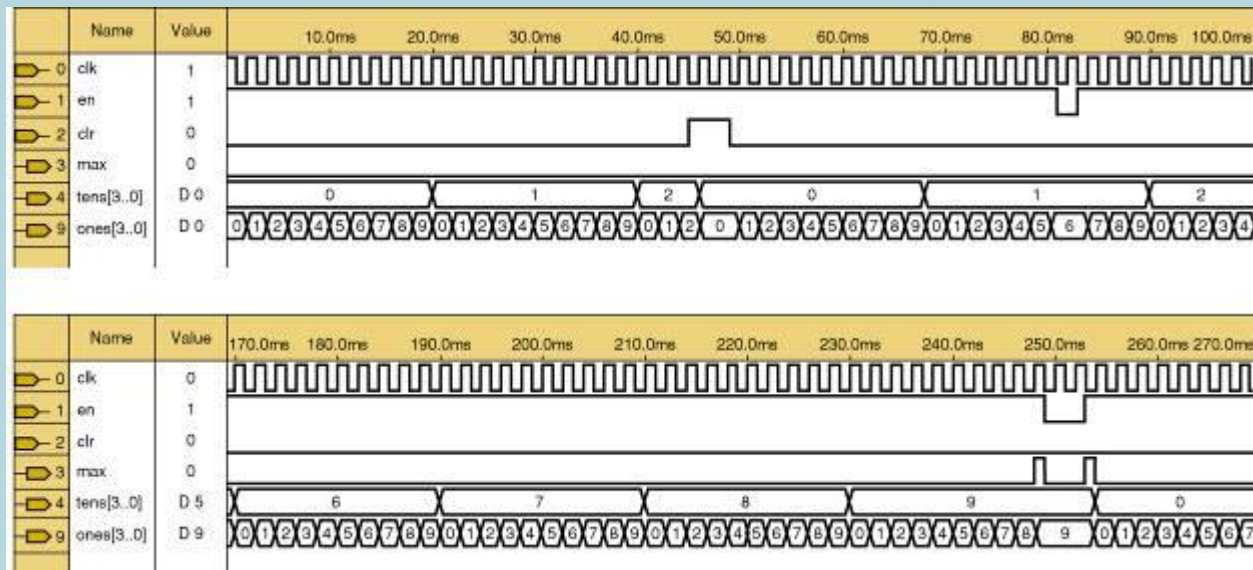
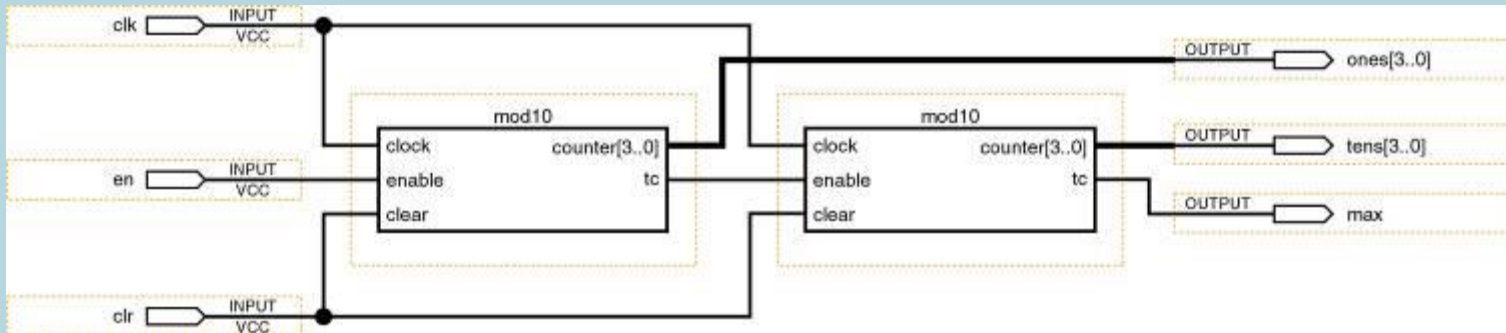
7-13 Conectando módulos em HDL

- Projetar grandes sistemas digitais é muito mais fácil se o sistema é subdividido em pequenos módulos gerenciáveis interligados.

A essência do conceito de **projeto hierárquico**.

- Considerar uma reciclagem, contador BCD MOD-100 com um síncrono evidente.
- Criar um módulo contador MOD-10 BCD, cascadeando de forma síncrona dois desses em um arquivo de projeto de alto nível.

Projeto de diagrama de blocos e simulação dos resultados para o projeto do contador BCD MOD-100.



7-13 Conectando módulos em HDL

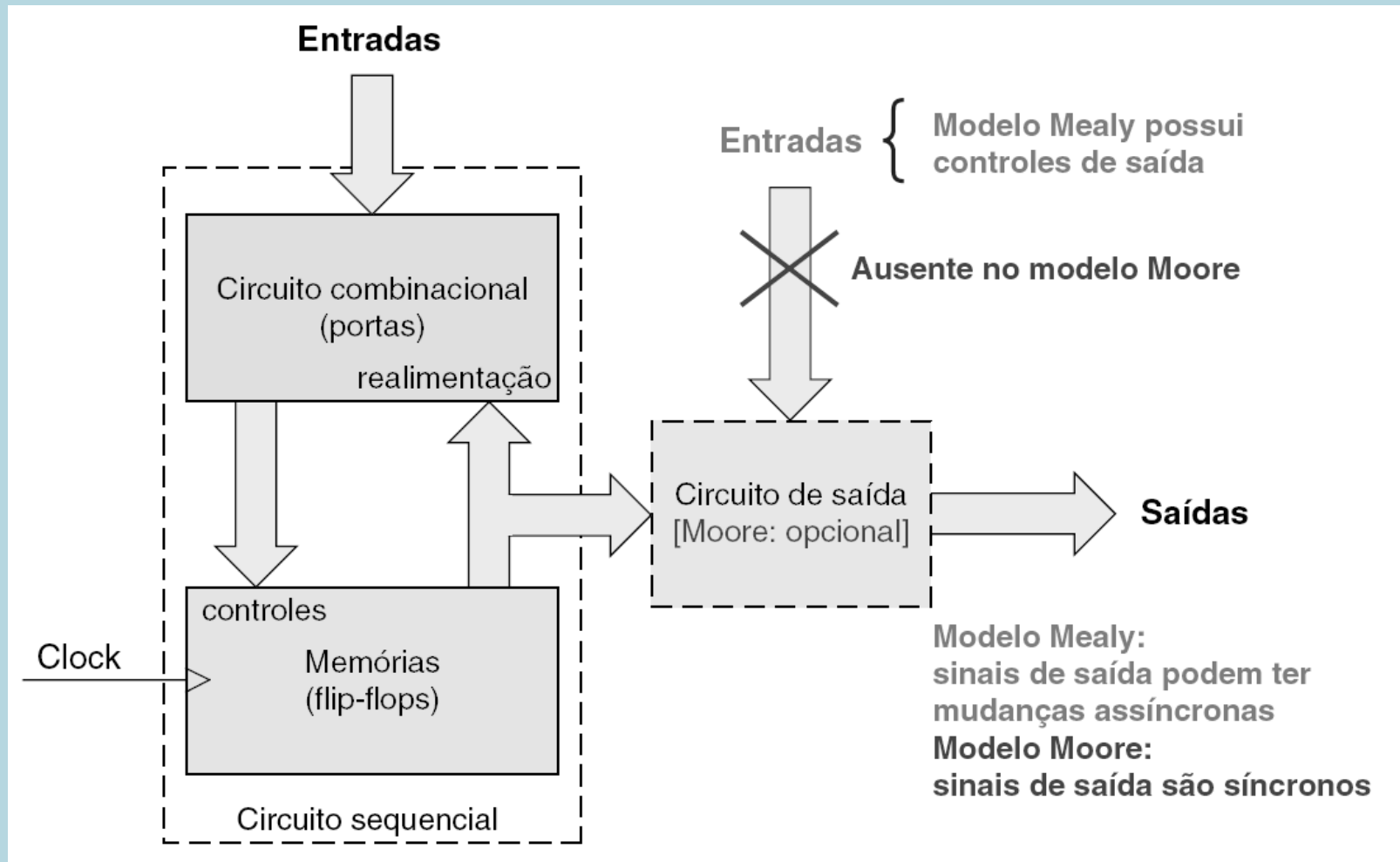
- Decodificando o contador AHDL MOD-5

```
1  SUBDESIGN fig7_39
2  (
3      clock      :INPUT;
4      q[2..0]    :OUTPUT;
5  )
6  VARIABLE
7      count[2..0] :DFF;      --cria um registrador de 3 bits
8  BEGIN
9      count[].clk = clock;  --conecta todos os clocks em paralelo
10
11      CASE count[] IS
12          --          Present          Próximo
13          -----
14              WHEN 0      => count[].d = 1;
15              WHEN 1      => count[].d = 2;
16              WHEN 2      => count[].d = 3;
17              WHEN 3      => count[].d = 4;
18              WHEN 4      => count[].d = 0;
19              WHEN OTHERS => count[].d = 0;
20          END CASE;
21      q[] = count[].q;      --conecta o registrador a pinos de saída
22  END;
```

7-14 Máquinas de estado

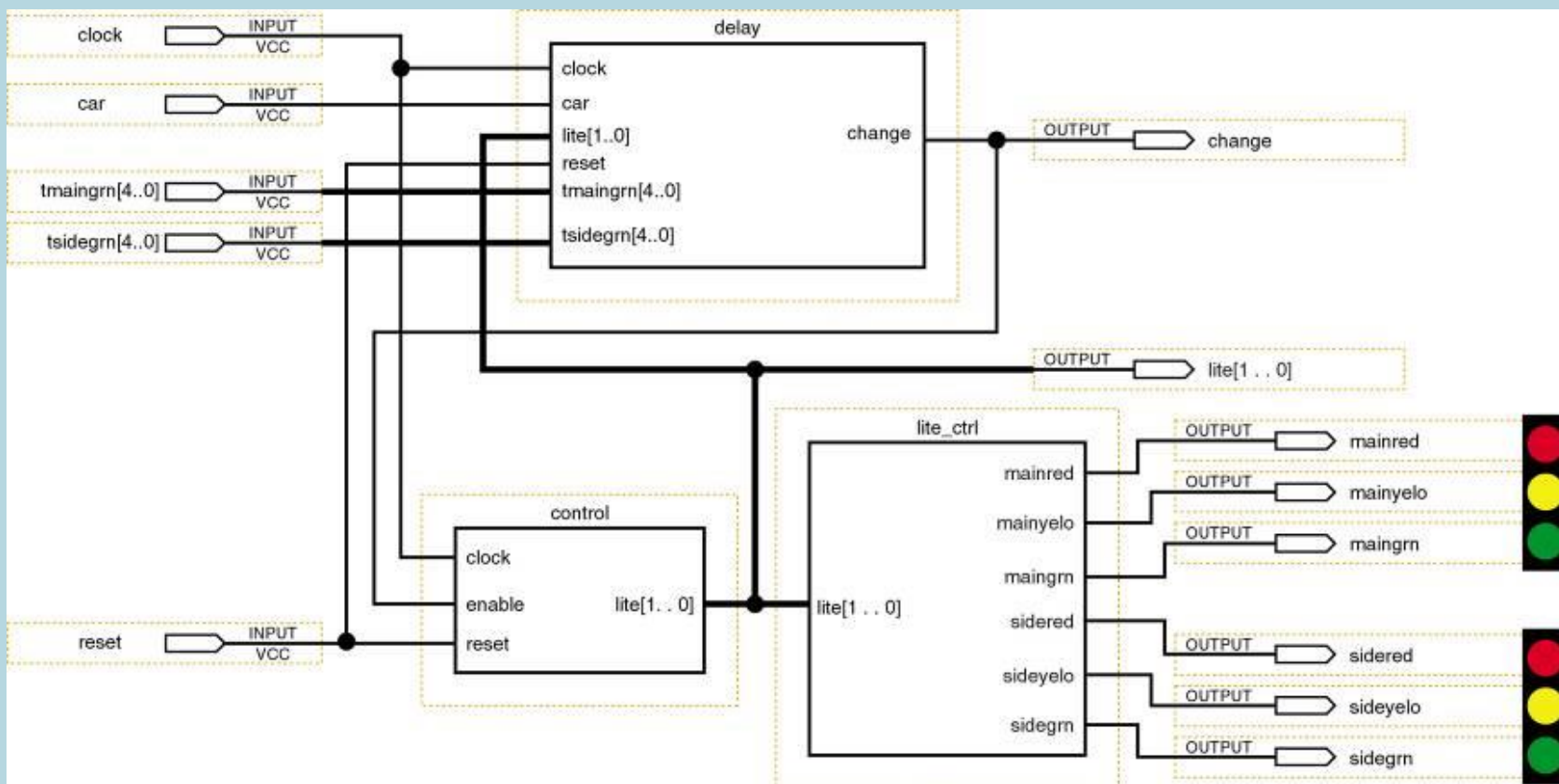
- O termo **máquina de estado** se refere a um circuito que tem uma sequência através de um conjunto de estados predeterminados controlados por um clock e outros sinais de entrada.
- O termo contador é usado para circuitos sequenciais que têm uma sequência de contagens numéricas regulares.
- Na verdade, as coisas contadas são pulsos de clock. Mas os pulsos podem representar muitos tipos de eventos.
 - A distinção geral entre os dois termos:
Um *contador* é comumente usado para *contar* eventos.
Uma *máquina de estado* é comumente usada para *controlar* os acontecimentos.

Diagrama em blocos de contadores e máquinas de estado.



7-14 Máquinas de estado

Máquinas de estado. Controlador de farol de trânsito.



7-15 Circuitos integrados de registradores

- Os vários tipos de registradores podem ser classificados de acordo com a maneira pela qual os dados podem ser apresentados no registro para armazenamento e pelo modo como saem do registrador.

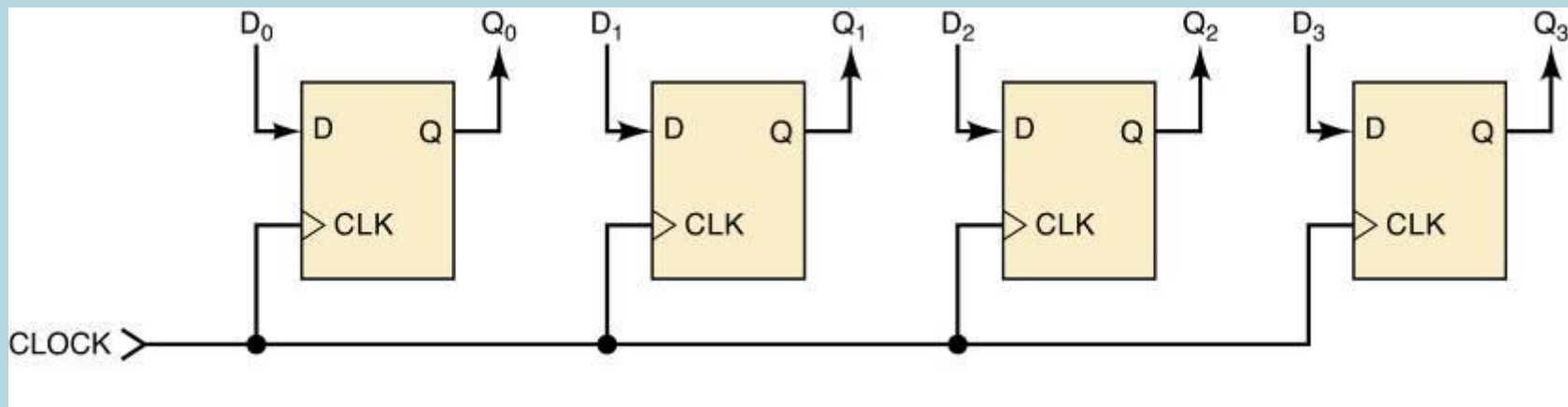
O fluxo de dados seriais através de um registro geralmente é chamado de deslocamento (*shifting*), tanto para a esquerda quanto para a direita.

- Se o dado de saída serial é realimentado para a entrada serial do mesmo registrador, a operação é chamada de rotação de dados.
- A entrada paralela de dados é usualmente descrita como uma carga de registrador.

7-15/16 Circuitos integrados de registradores/ CIs de registradores

74ALS174/74HC174

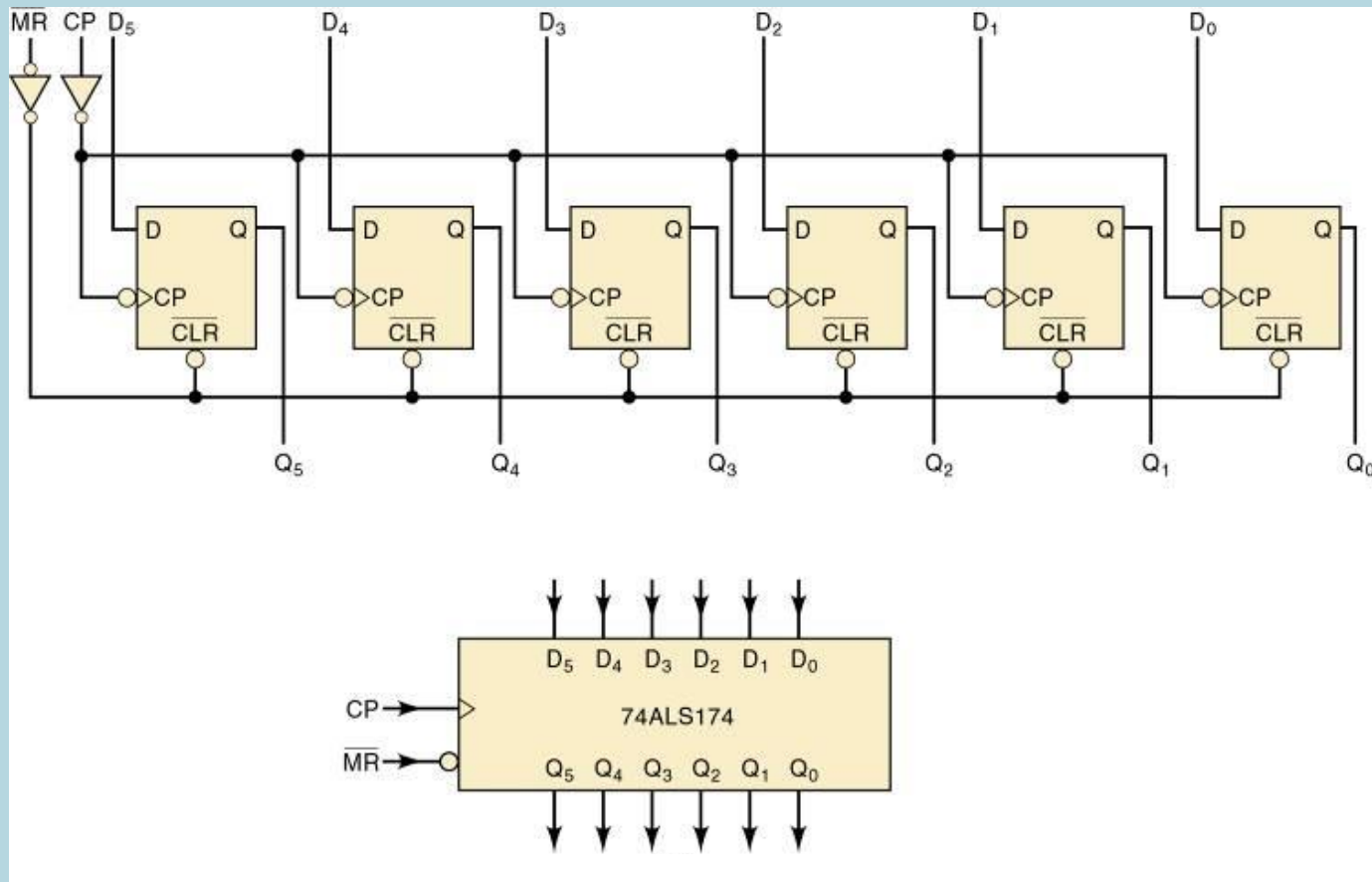
Entrada paralela/ Saída paralela (PIPO)



Um grupo de flip-flops que podem armazenar vários bits simultaneamente e no qual todos os bits de valor binário armazenados estão diretamente disponíveis.

7-15/16 Circuitos integrados de registradores/ CIs de registradores

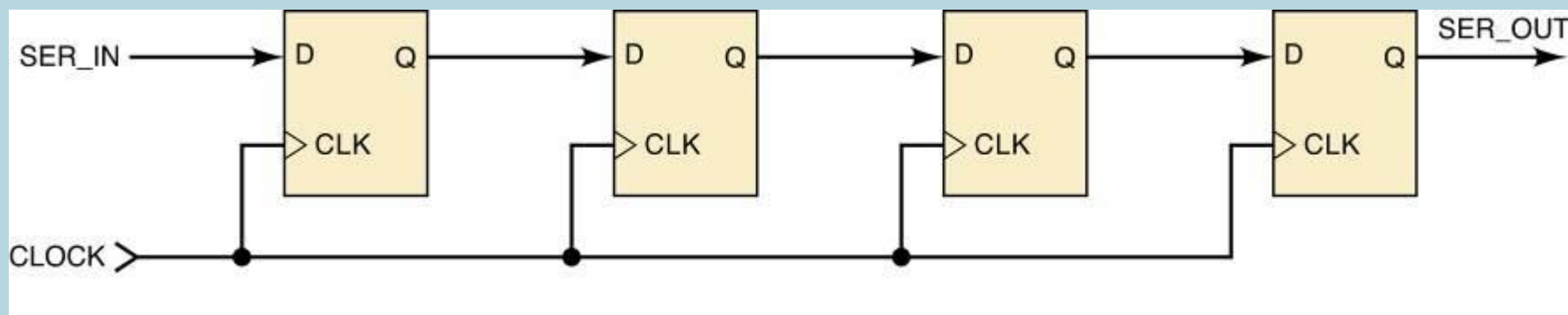
74ALS174/74HC174



7-15/16 Circuitos integrados de registradores/ CIs de registradores

74ALS166/74HC166

Entrada serial/ Saída serial (SISO)



Dados carregados um de cada vez se movem um de cada vez, com cada pulso do relógio através dos flip-flops para a outra extremidade do registro, e saem um de cada vez na mesma ordem, como originalmente carregados.

7-15/16 Circuitos integrados de registradores/ CIs de registradores

74ALS166/74HC166

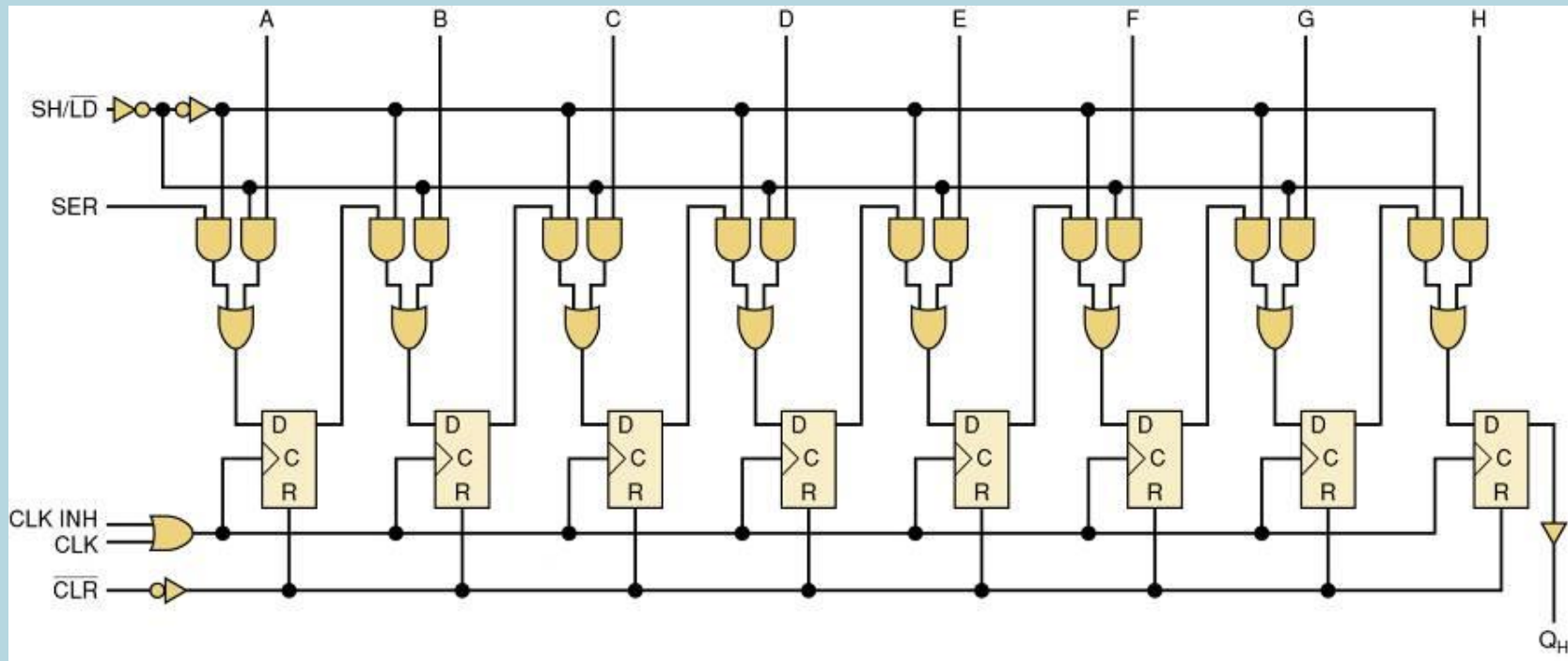
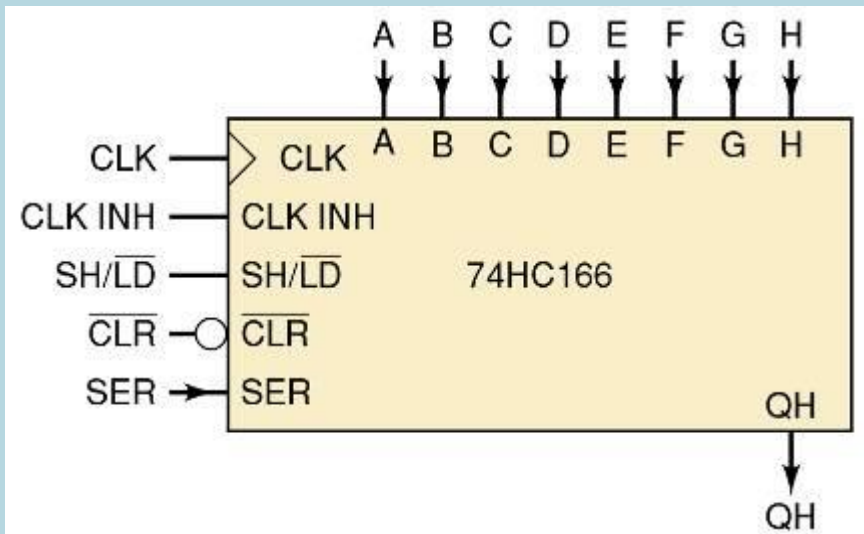


Diagrama de Circuito

7-15/16 Circuitos integrados de registradores/ CIs de registradores

74ALS166/74HC166



Símbolo lógico.

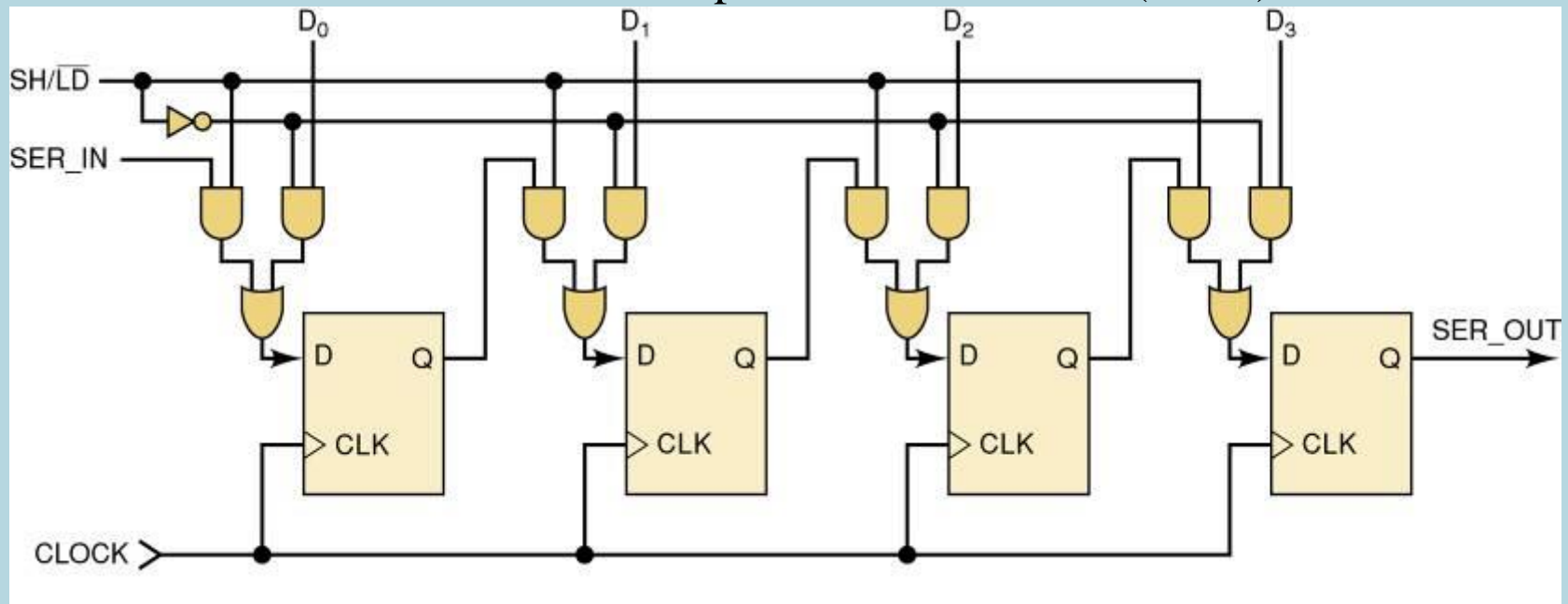
Tabela de funções.

ENTRADAS						SAÍDAS		
						INTERNAS		QH
CLR	SH/LD	CLK INH	CLK	SER	PARALELO A...H	QA	QB	
L	X	X	X	X	X	L	L	L
H	X	L	L	X	X	QA0	QB0	QH0
H	L	L	↑	X	a...h	a	b	h
H	H	L	↑	H	X	H	QAn	QGn
H	H	L	↑	L	X	L	QAn	QGn
H	X	H	↑	X	X	QA0	QB0	QH0

7-15/16 Circuitos integrados de registradores/ CIs de registradores

74ALS165/74HC165

8 bits em entrada paralela/ saída serial (PISO)



Mudança de série é sempre síncrona, assim como o relógio é necessário para garantir que os dados dos movimentos de entrada só se movam um bit de cada vez com cada borda de clock apropriada.

7-15/16 Circuitos integrados de registradores/ CIs de registradores

74ALS165/74HC165

8-bit entrada paralela/ saída serial (PISO)

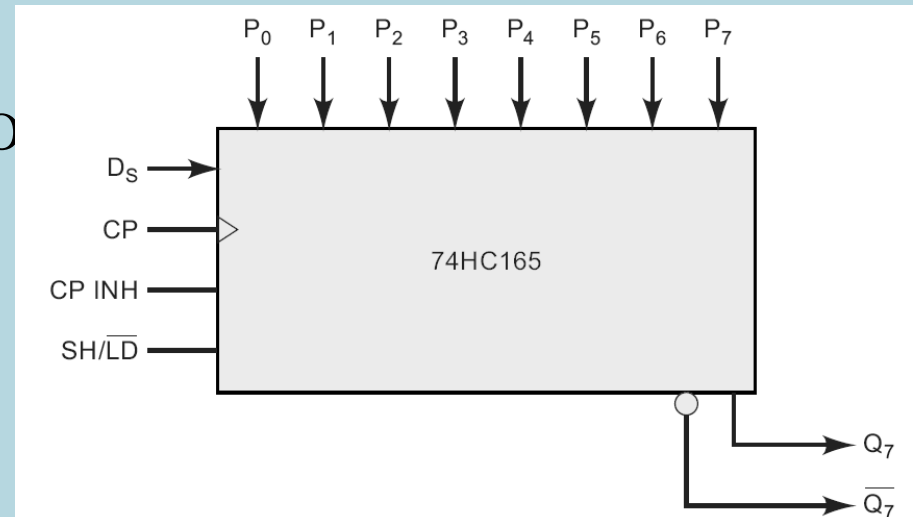


Tabela de funções

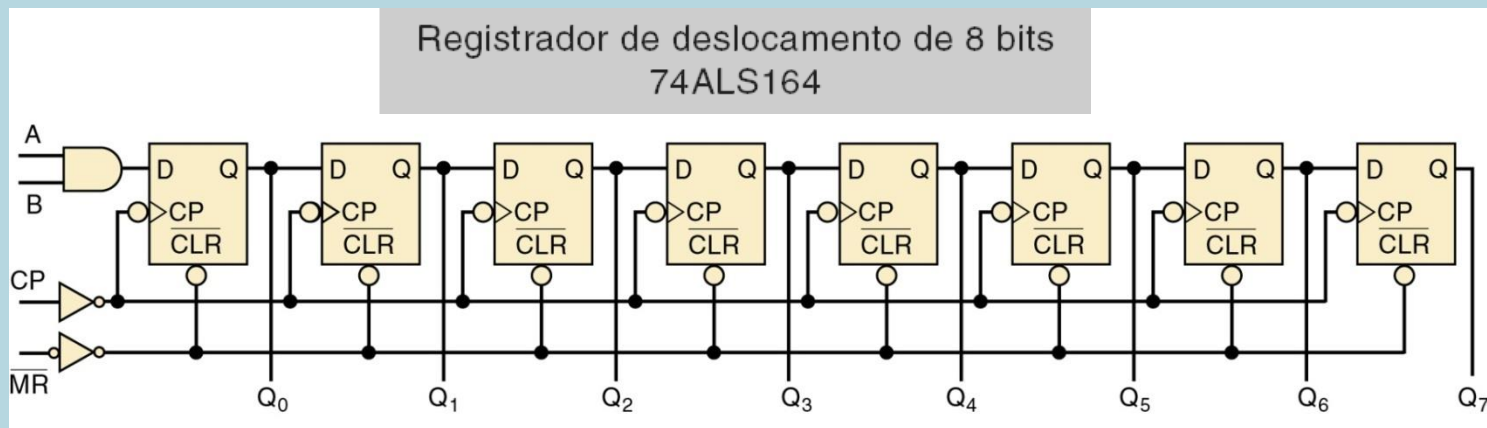
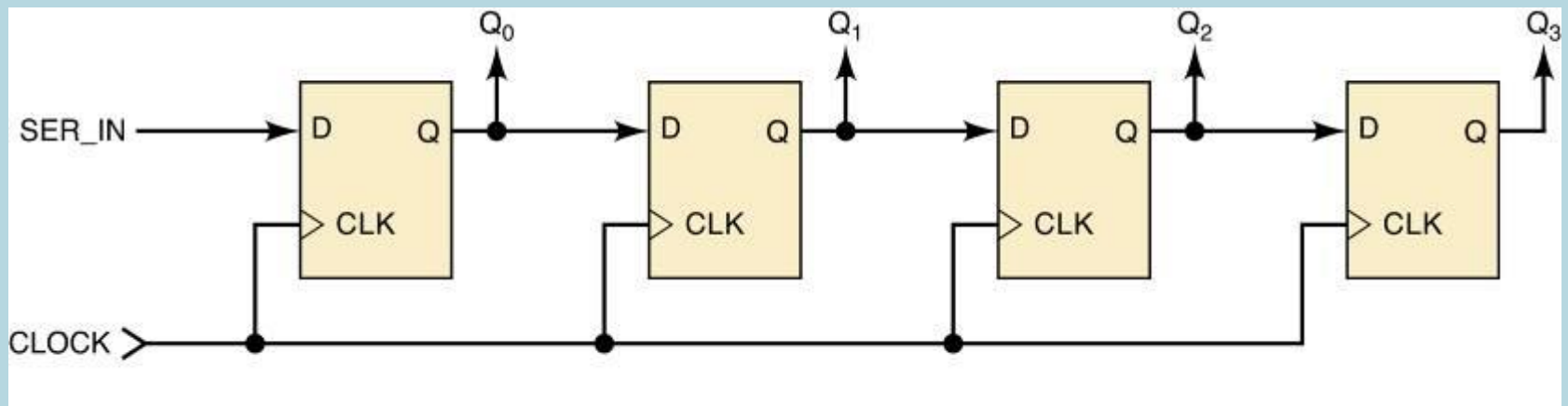
Entradas			Operação
SH/LD	CP	CP INH	
L	X	X	Carga paralela
H	H	X	Sem mudança
H	X	H	Sem mudança
H	\downarrow	L	Deslocamento
H	L	\downarrow	Deslocamento

H = nível alto
 L = nível baixo
 X = irrelevante
 \downarrow = borda de subida

7-15/16 Circuitos integrados de registradores/ CIs de registradores

74ALS164/74HC164

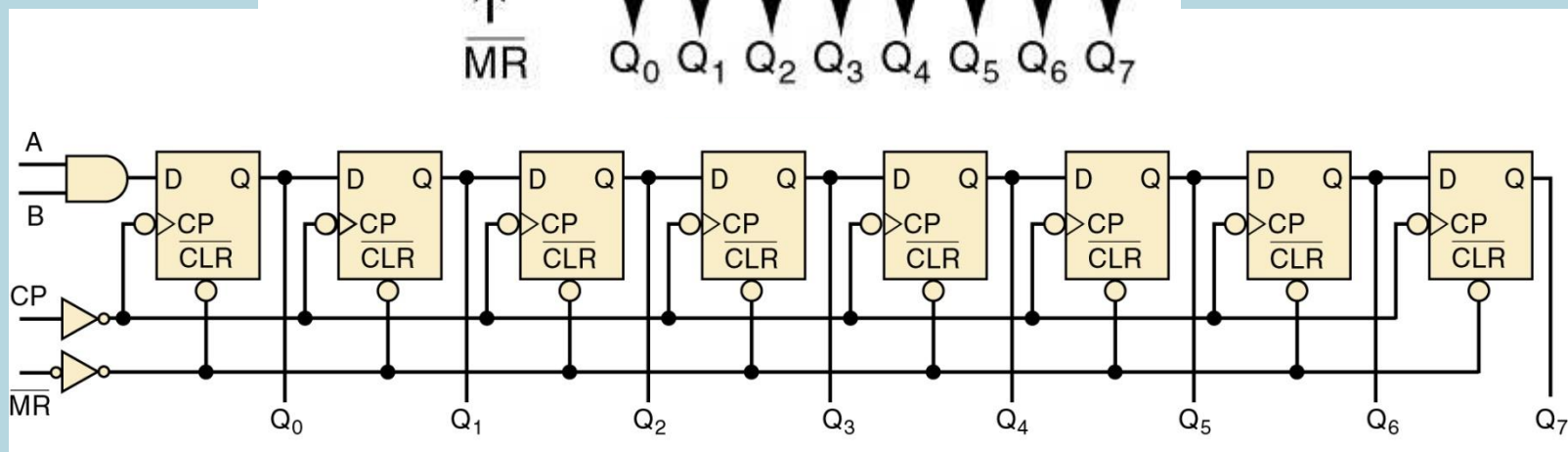
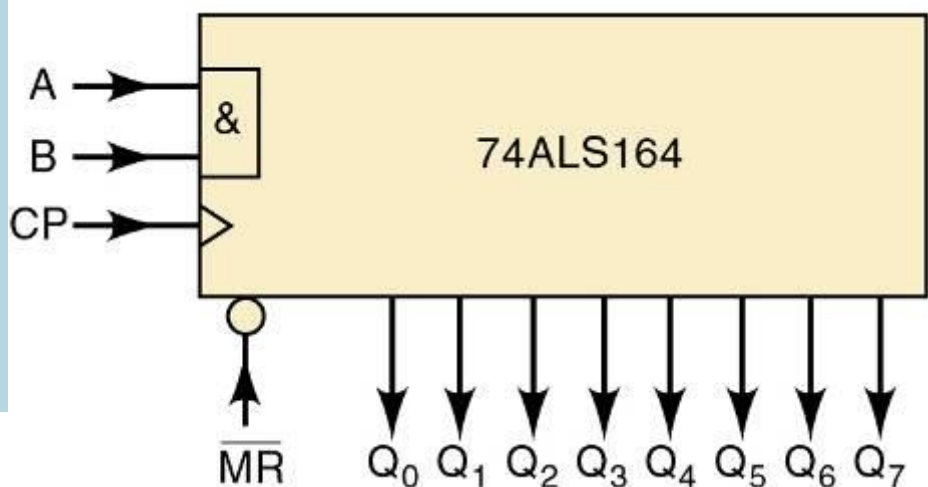
8-bit entrada serial/ saída paralela (SIPO)



7-15/16 Circuitos integrados de registradores/ CIs de registradores

74ALS164/74HC164

8-bit entrada serial/ saída paralela (SIPO)

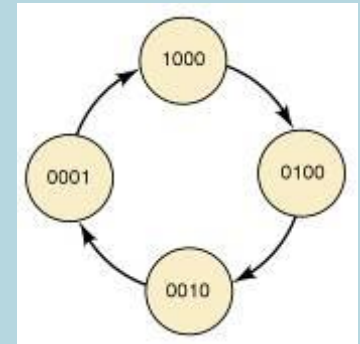
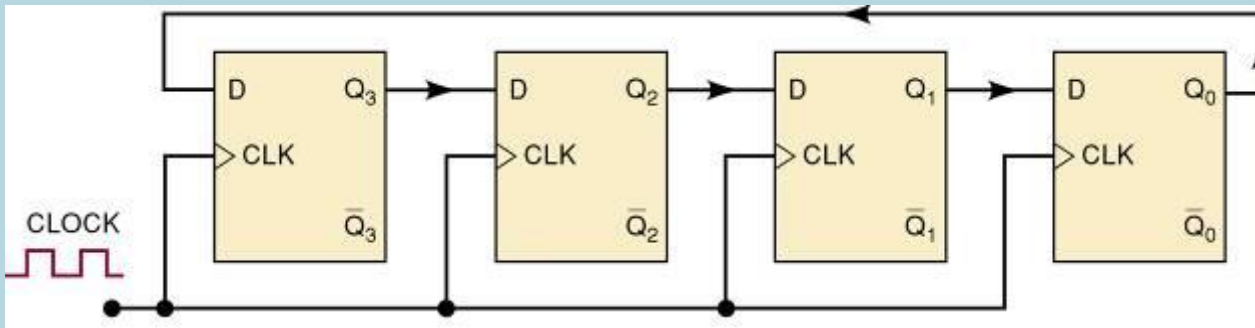


7-17 Contadores com registradores de deslocamento

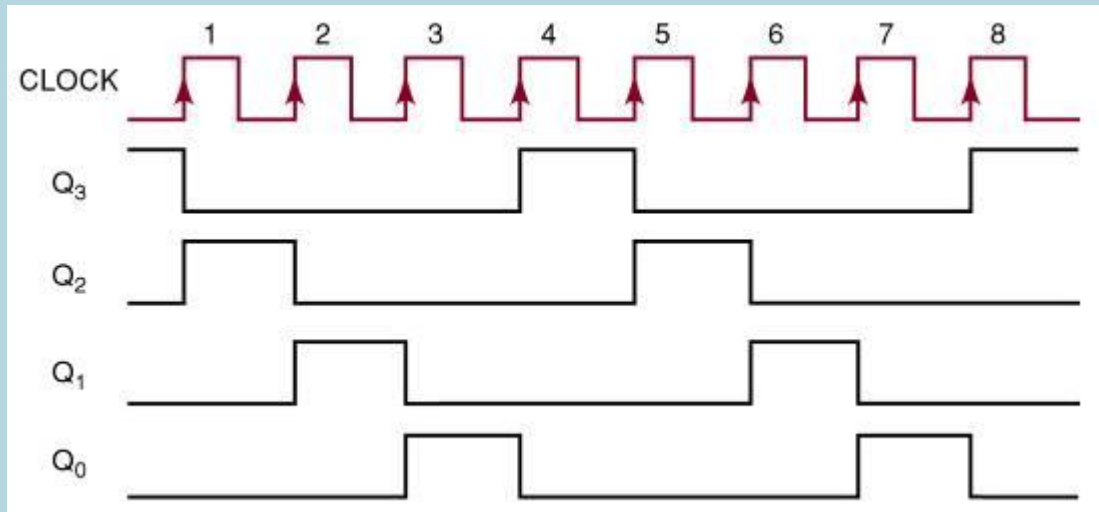
- Contadores com registradores de deslocamento usam a *realimentação*, o que significa que a saída do último *FF* do registrador é conectada de volta no primeiro flip-flop.

7-17 Contadores com registradores de deslocamento

Um **contador em anel** é um **registrador circular** conectado de maneira que o último *FF* desloque seu valor para o primeiro.

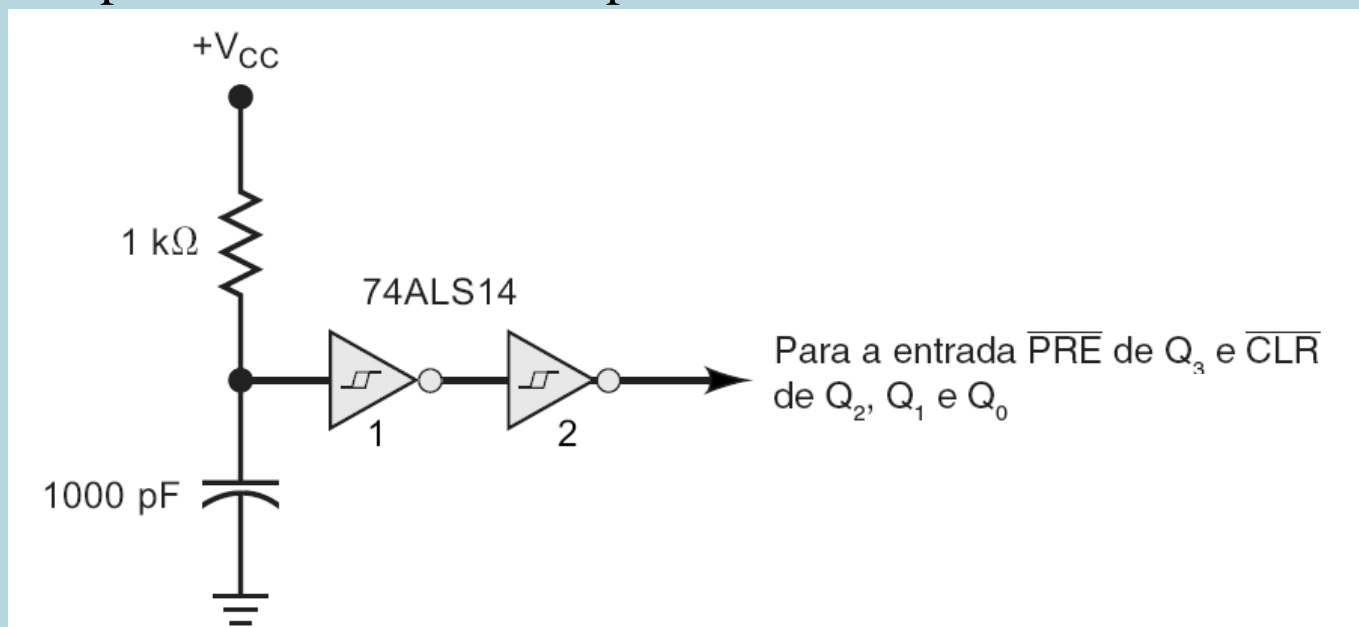


Q ₃	Q ₂	Q ₁	Q ₀	Pulsos de CLOCK
1	0	0	0	0
0	1	0	0	1
0	0	1	0	2
0	0	0	1	3
1	0	0	0	4
0	1	0	0	5
0	0	1	0	6
0	0	0	1	7
·	·	·	·	·
·	·	·	·	·



7-17 Contadores com registradores de deslocamento

- Para funcionar adequadamente, um contador em anel deve começar com apenas um FF no estado 1 e todos os outros no estado 0.
Como o estado de partida será imprevisível, o contador deve ser predefinido antes dos pulsos de clock serem aplicados.

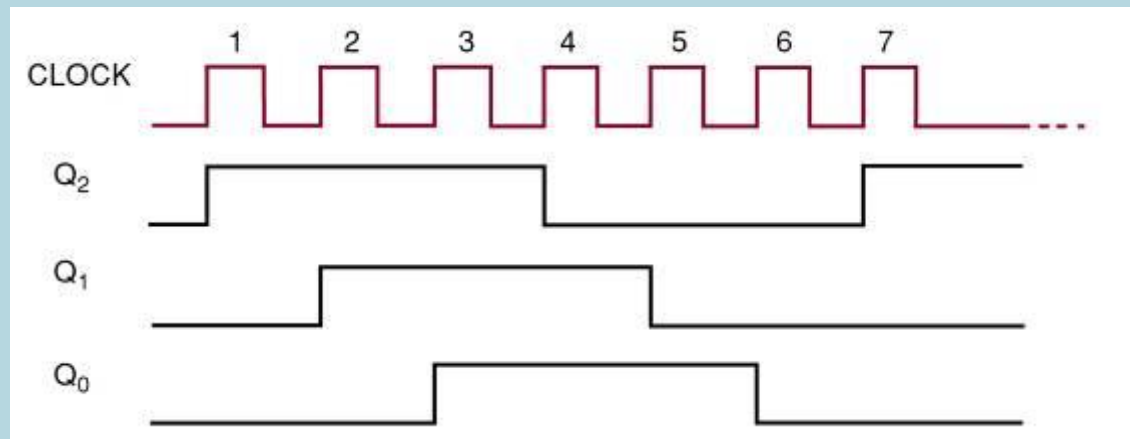
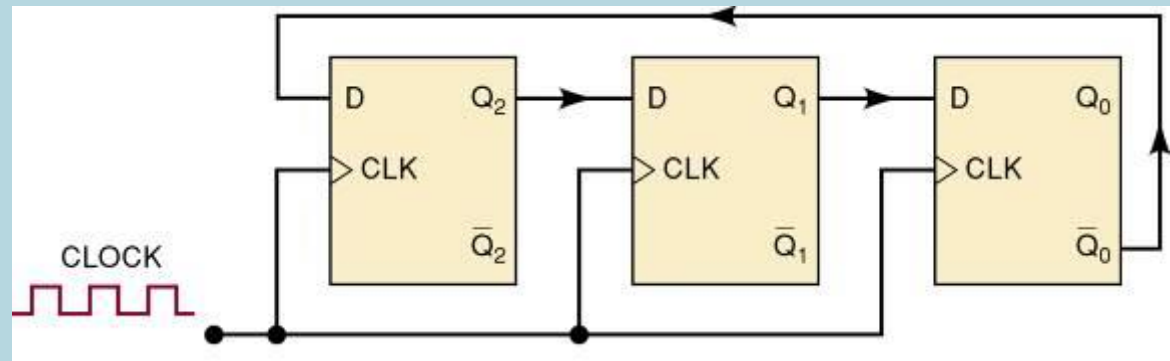


Circuito para assegurar que o contador em anel inicie o estado de ligação em 1000.

7-17 Contadores com registradores de deslocamento

- No contador **Johnson** ou **em anel torcido**, a saída *invertida* do último *FF* é conectada à entrada do primeiro *FF*.

Q_2	Q_1	Q_0	Pulsos de CLOCK
0	0	0	0
1	0	0	1
1	1	0	2
1	1	1	3
0	1	1	4
0	0	1	5
0	0	0	6
1	0	0	7
1	1	0	8
.	.	.	.
.	.	.	.
.	.	.	.



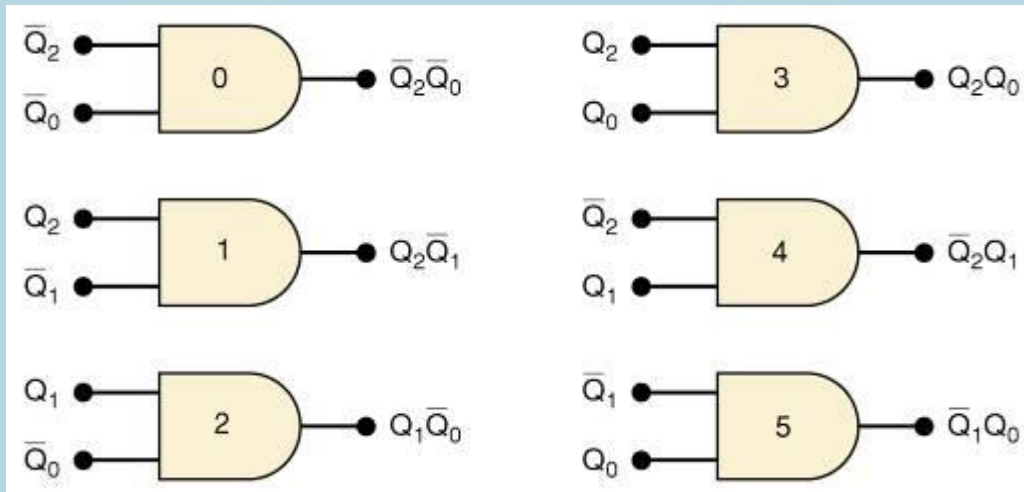
7-17 Contadores com registradores de deslocamento

- Para um número MOD dado, um contador Johnson requer apenas metade dos *FFs* que um contador em anel requer.

Exige portas de decodificação, enquanto um contador em anel não exige.

- Um contador Johnson usa uma porta lógica para decodificar cada conta.

Cada porta exige apenas duas entradas, independentemente do número de *FFs* no contador.



Q_2	Q_1	Q_0	Porta ativa
0	0	0	0
1	0	0	1
1	1	0	2
1	1	1	3
0	1	1	4
0	0	1	5

7-17 Contadores com registradores de deslocamento

- Poucos contadores em anel ou contadores Johnson estão disponíveis como CIs.
- A mudança de um fio CI para um contador em anel ou um contador Johnson é relativamente simples.
- Alguns dos CIs contadores Johnson CMOS (74HC4017, 74HC4022) incluem os circuitos de decodificação completos no mesmo chip do contador.

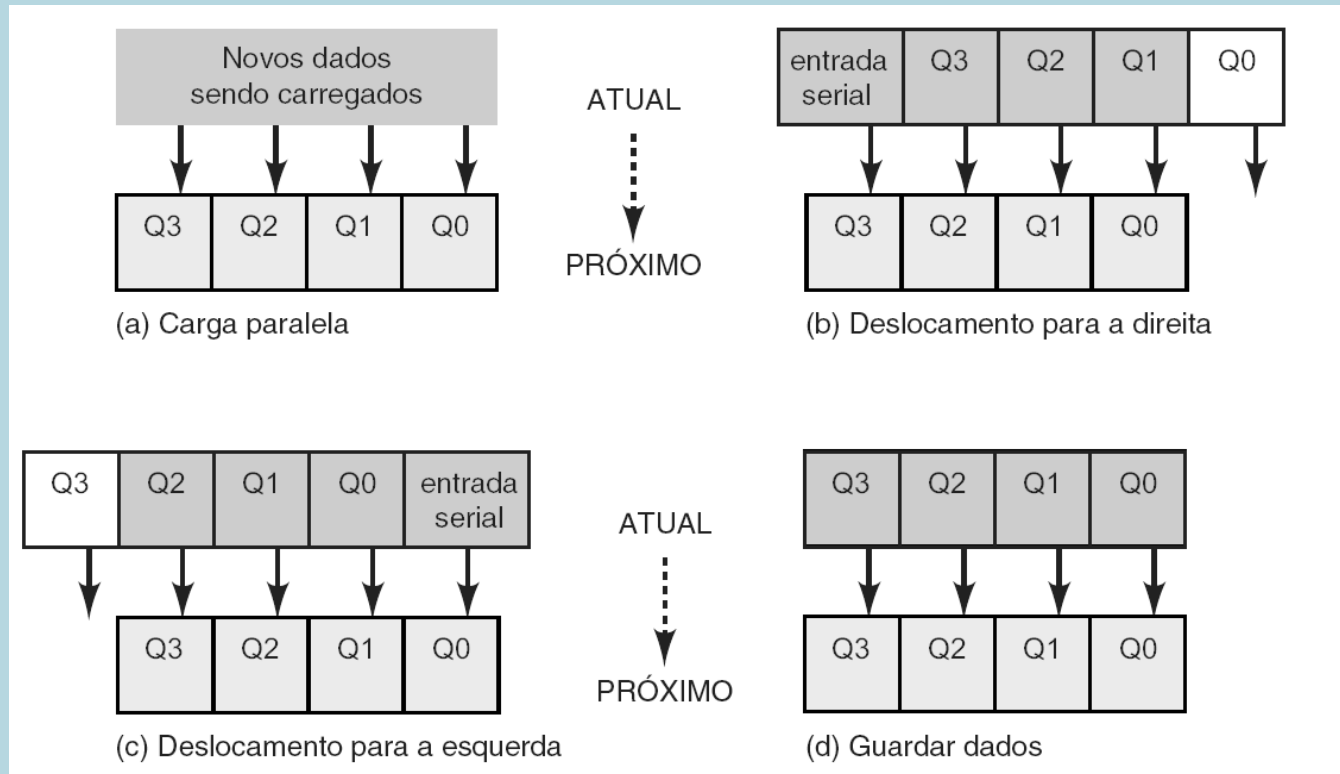
7-18 Análise de defeitos

- Flip-flops, contadores e registradores são os principais componentes em **sistemas de lógica sequencial**.
Os sistemas sequenciais sofrem os mesmos tipos de falhas que os sistemas combinacionais: circuitos abertos, curtos, falhas internas em CIs, etc.
- Resolução básica de problemas:
 - Observar a operação do sistema.
 - Usar o raciocínio analítico para determinar as possíveis causas.
 - Utilizar equipamentos de teste para isolar a falha exata.

7-20 Registradores em HDL

- As técnicas de HDL usam vetores de bits para descrever os dados de um registrador e transferi-los em um formato paralelo ou serial.
- Concatenação (ação de agrupar em uma sequência específica) de um conjunto desejado de bits de dados pode ser usada para descrever o movimento de dados para mudança do serial.

Quatro flip-flops realizando operações de transferência de carga paralela, deslocamento para a direita, desvio à esquerda e armazenamento de dados.



Bits são transferidos de forma síncrona, o que significa que todos eles se movem simultaneamente em uma única borda de clock.

7-21 Contadores em anel em HDL

- Um contador em anel é um registrador de deslocamento que circula um único nível lógico ativo por todos os seus *FFs*.
- Módulo é igual ao número de *FFs* no registrador e, assim, sempre existem muitos estados não utilizados e inválidos.
- Com planejamento, podemos garantir que contadores cheguem a sequência desejada, não importando o estado inicial.

Independentemente do estado em que o contador é inicializado, eventualmente ele se enche de zeros.

Em um momento, a lógica muda em um nível ALTO para iniciar a sequência em anel.

7-21 Contadores em anel em AHDL

- No código AHDL mostrado, as linhas 11 e 12 controlam a entrada serial usando a estratégia que acabamos de descrever. O operador (`==`) avalia se as expressões de ambos os lados são iguais.

```
SUBDESIGN fig7_92
(
  clk      :INPUT;
  q[3..0]  :OUTPUT;
)
VARIABLE
  ff[3..0] :DFF;
  ser_in   :NODE;
BEGIN
10   ff[].clk = clk;
11   IF ff[3..1].q == B"000" THEN ser_in = VCC; -- autoinício
12   ELSE ser_in = GND;
13   END IF;
14   ff[3..0].d = (ser_in, ff[3..1].q);    -- deslocamento para a direita
15   q[] = ff[].q;
16   END;
```

7-22 Monoestáveis em HDL

- O conceito de um contador pode ser aplicado para implementar um **monoestável** digital usando HDL.
- Um monoestável *não disparável* ignora a entrada do disparo enquanto a saída do pulso ainda estiver ativa.
- Um monoestável *disparável* inicia um pulso em resposta a um disparo. Cada vez que uma borda do disparo subsequente ocorre *antes* do pulso estar completo, o temporizador interno do pulso se reinicia.