

Introdução aos Sistemas Digitais

Circuitos sequenciais



Circuitos sequenciais vs. combinatórios

Num **circuito combinatório** saídas dependem exclusivamente das entradas correntes.

Num **circuito sequencial** saídas dependem das entradas correntes e de todas as entradas que tenham surgido antes (desde o momento em que o circuito foi usado pela 1ª vez!).



Se aumentar o volume?



Circuitos sequenciais

Na prática, é impossível enumerar a sequência de todas as entradas que ocorreram no passado.

Em vez disso, as saídas num **circuito sequencial** dependem das entradas correntes e do seu **estado**.

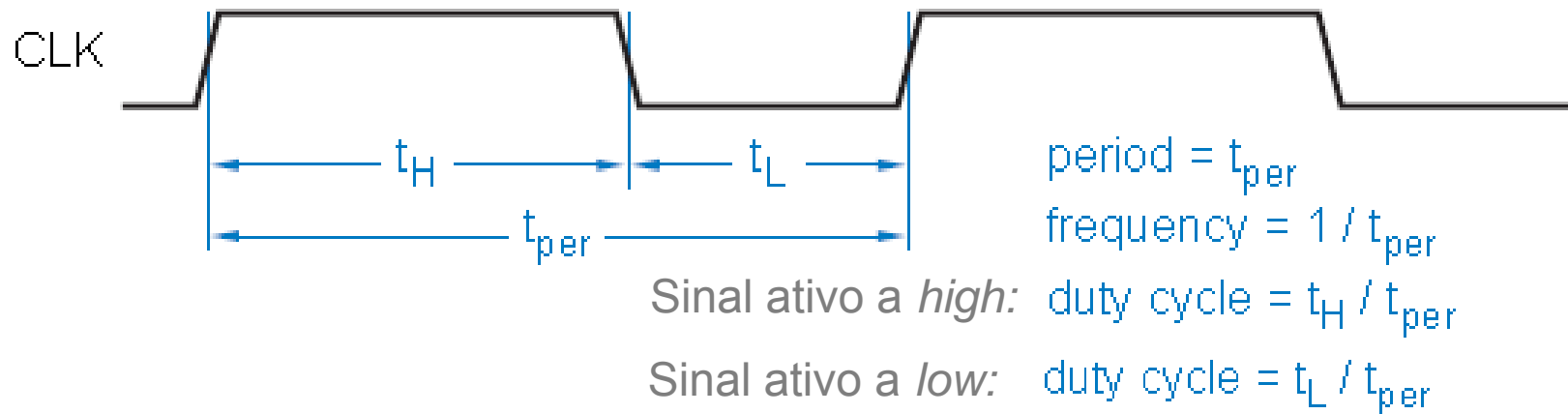
O **estado** dum circuito sequencial é uma coleção de variáveis de estado que contêm toda a informação sobre o passado, necessária para determinar o comportamento do circuito no futuro.

Um circuito com n variáveis de estado tem 2^n estados possíveis. Já que o número de estados é limitado, circuitos sequenciais são frequentemente chamados **máquinas de estados finitos**.



Sinal de relógio

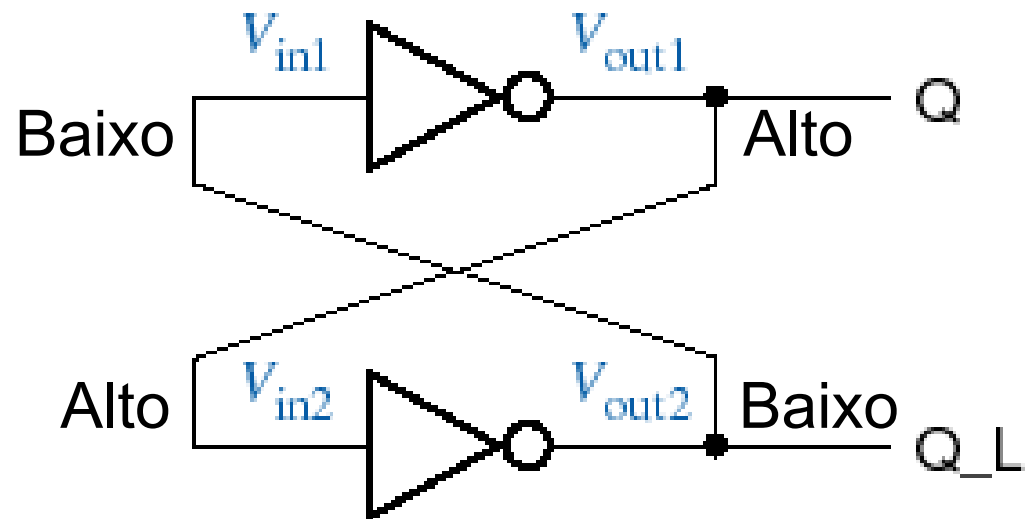
A transição de um estado para outro ocorre em momentos específicos determinados pelo sinal de **relógio** (clock).



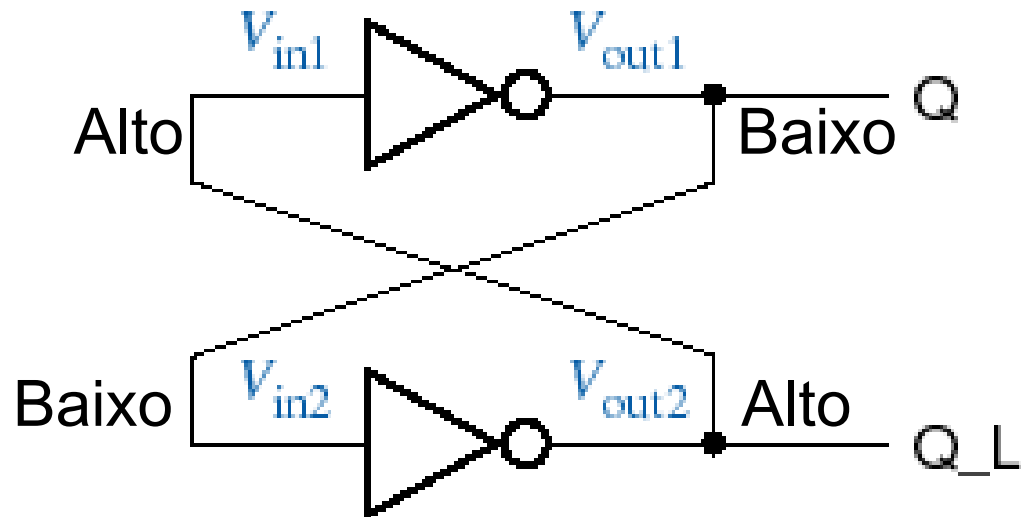
Exemplo: Qual é o período do sinal de relógio com $\text{freq} = 50 \text{ MHz}$?



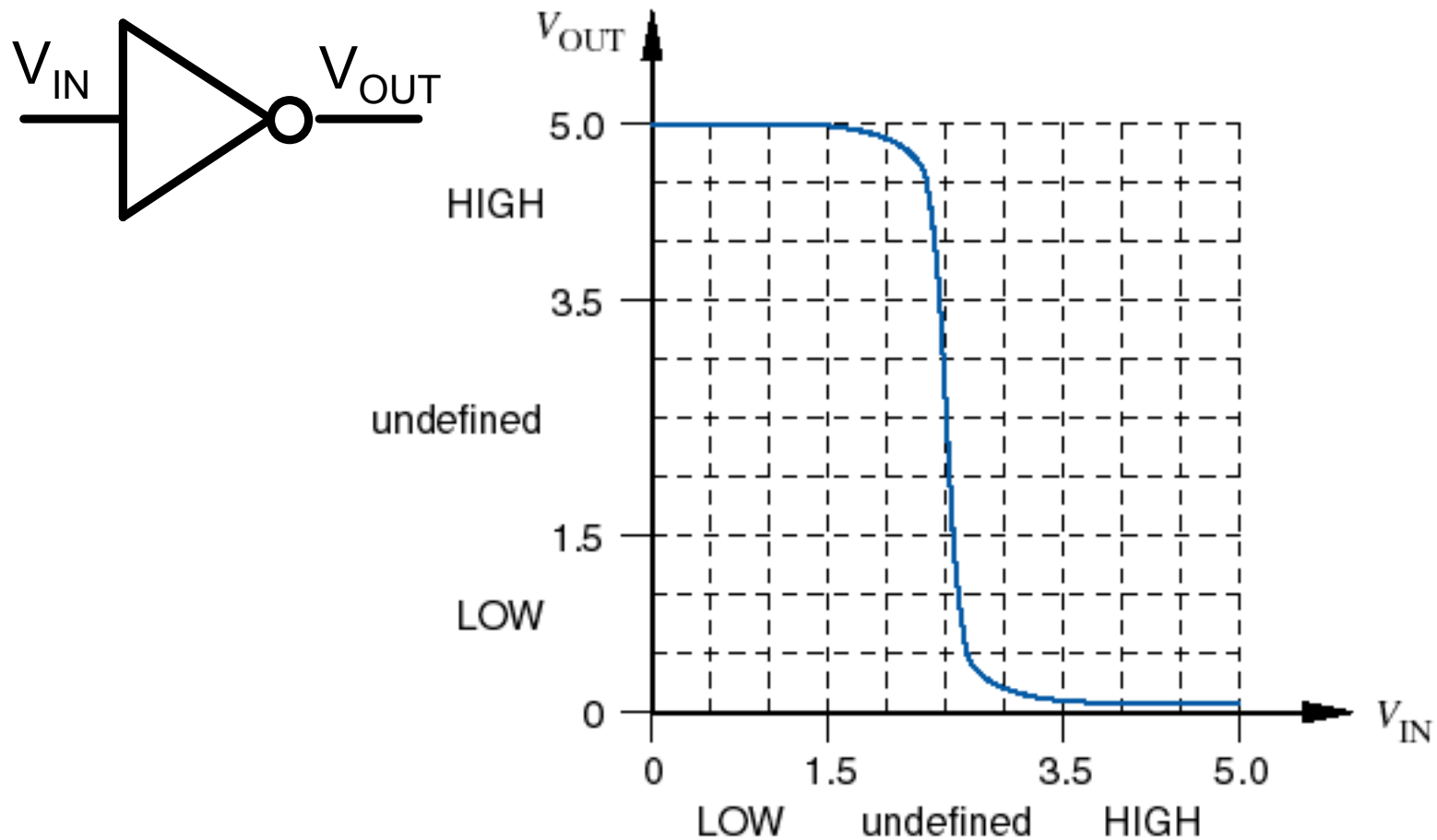
Elemento biestável



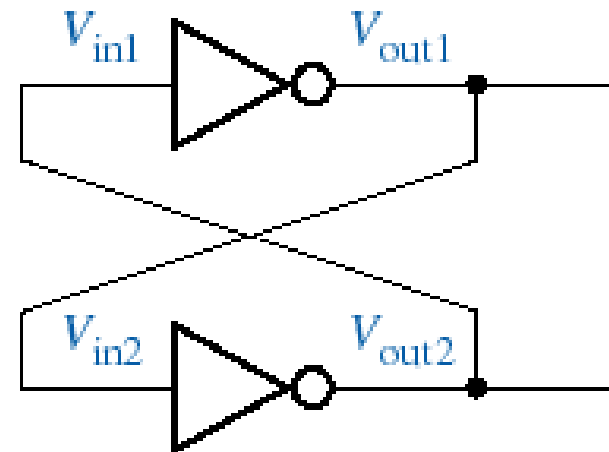
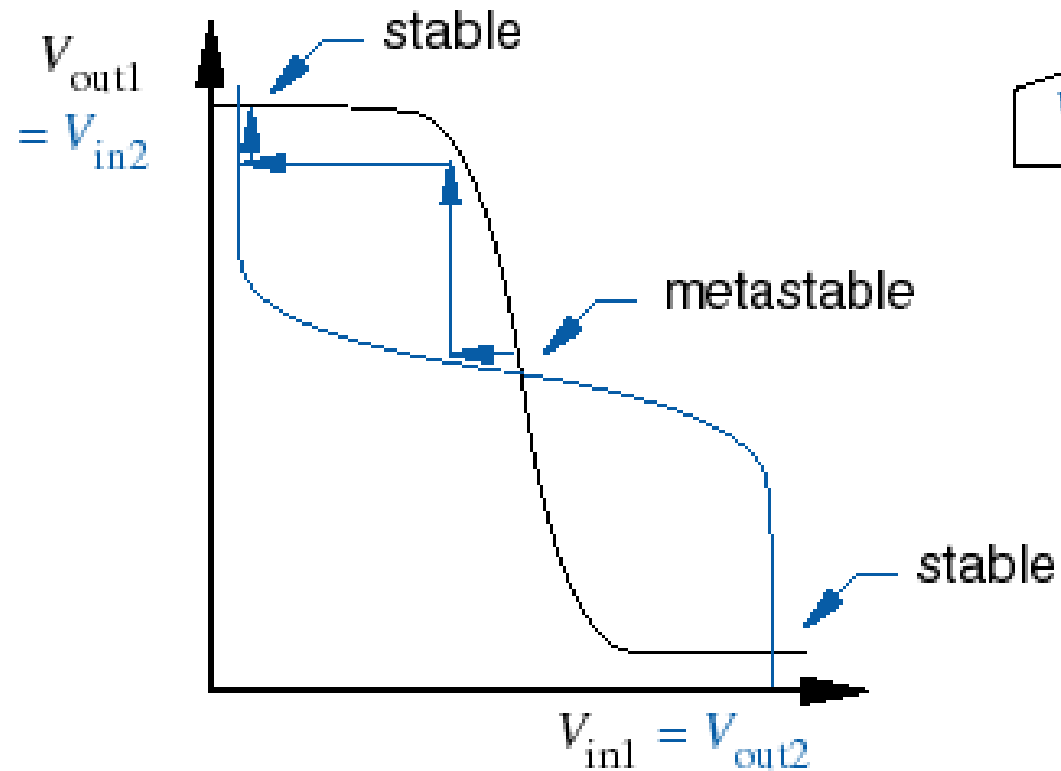
Elemento biestável (cont.)



Análise analógica



Pontos de equilíbrio

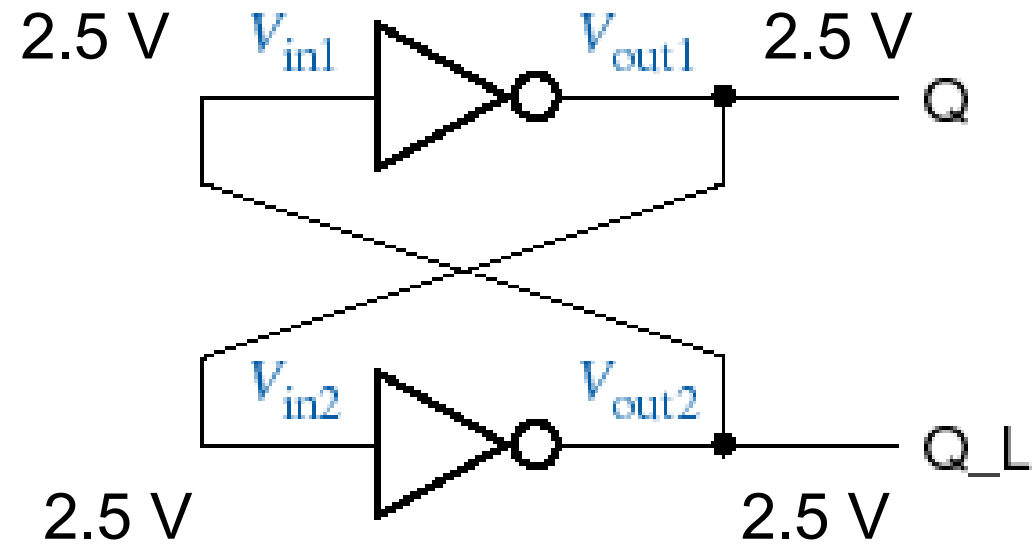


Transfer function:

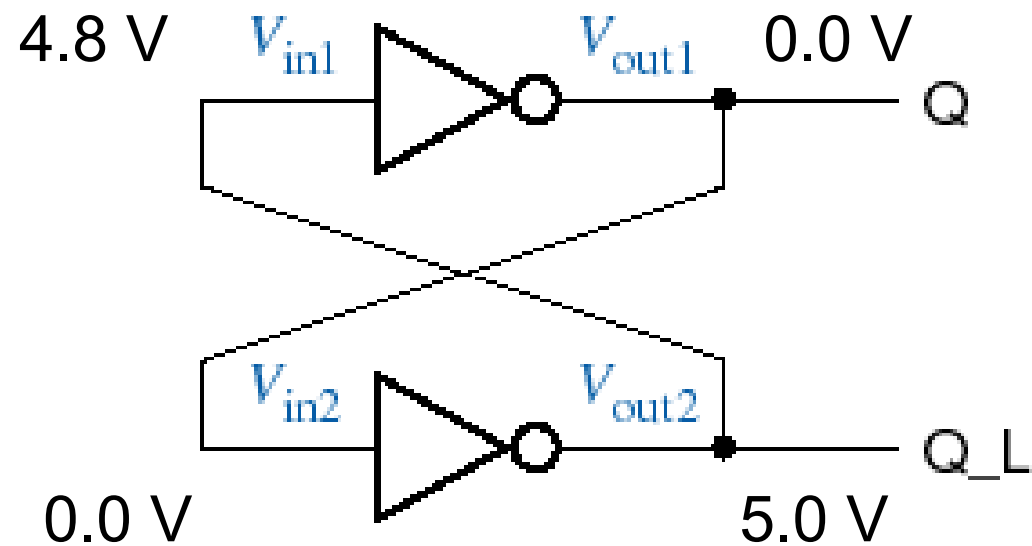
$$V_{out1} = T(V_{in1})$$

$$V_{out2} = T(V_{in2})$$

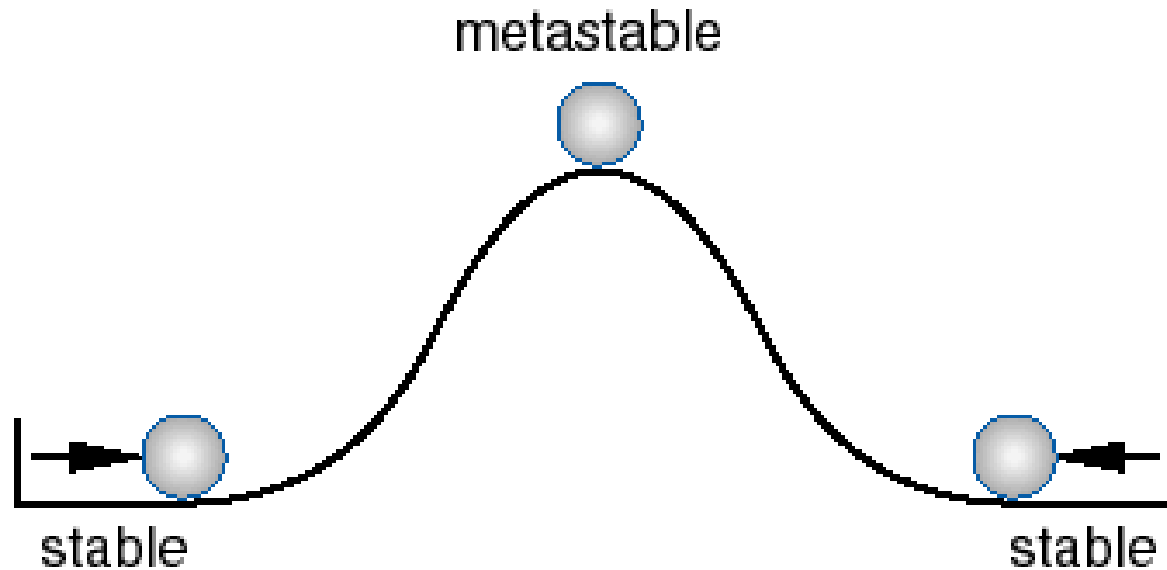
Meta-estabilidade



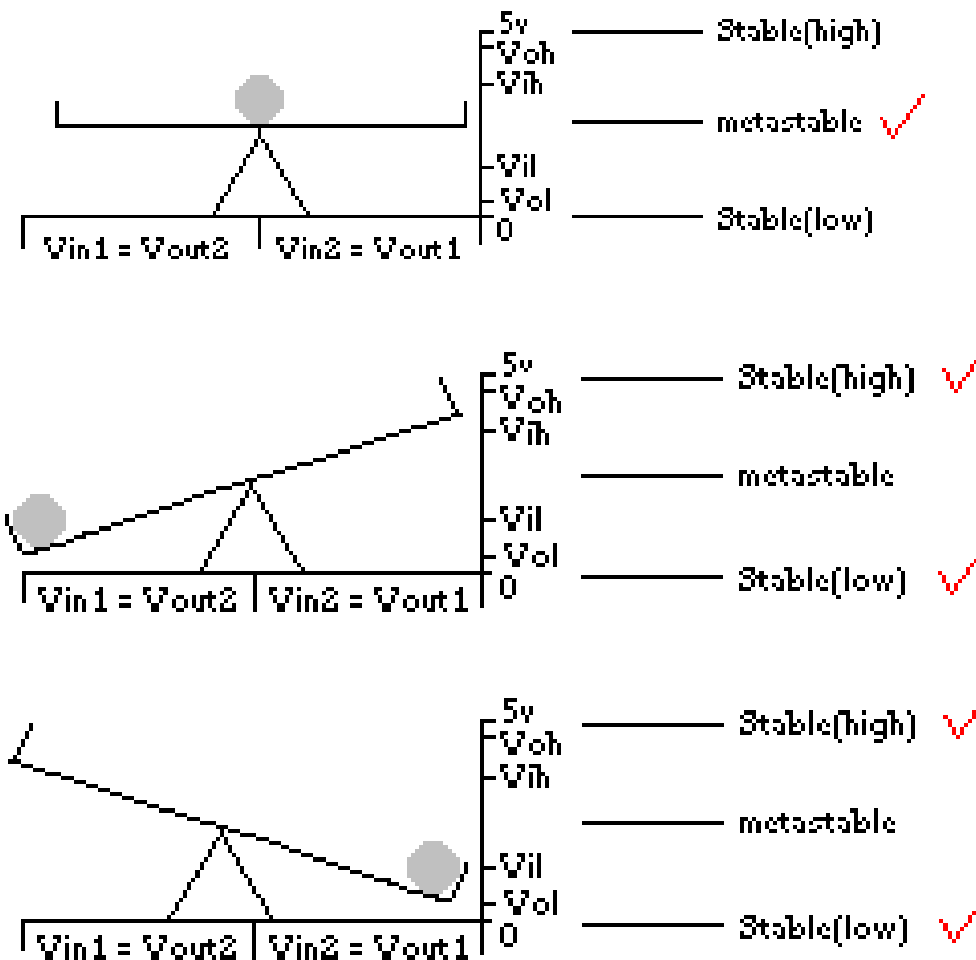
Meta-estabilidade (cont.)



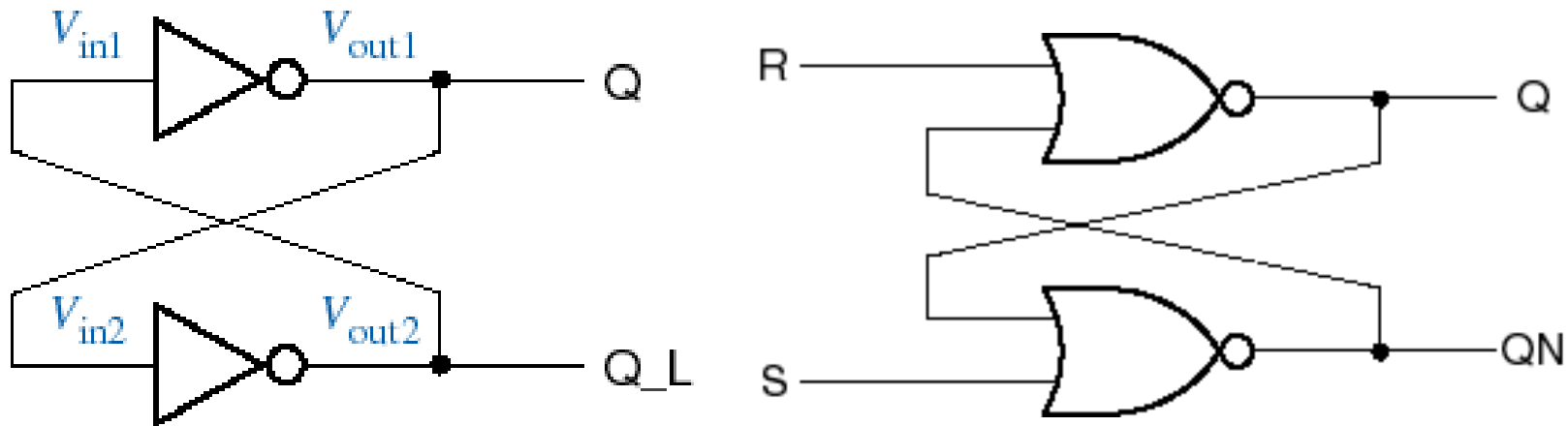
Meta-estabilidade: analogia mecânica (I)



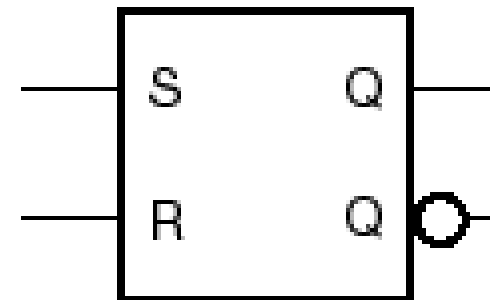
Meta-estabilidade: analogia mecânica (II)



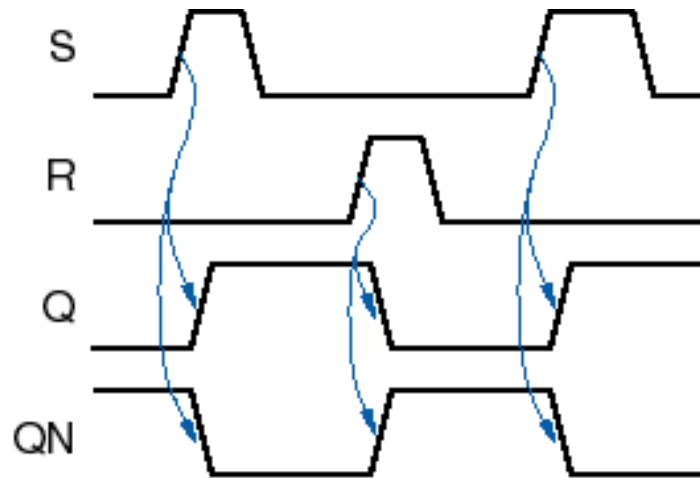
Latch S-R



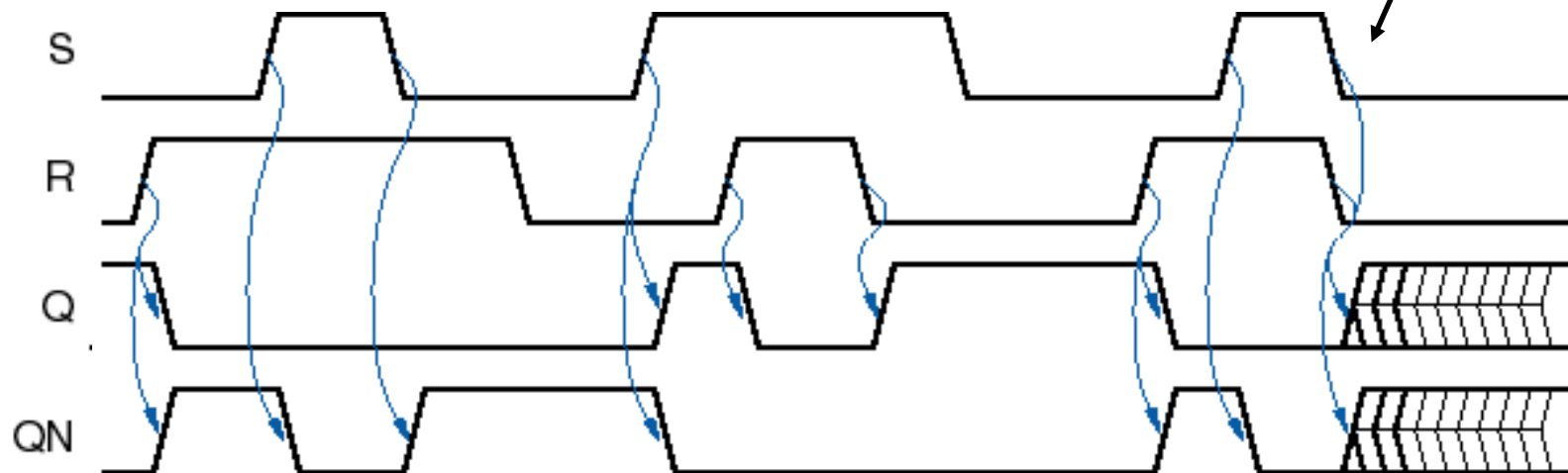
S	R	Q	Q_N
0	0	last Q	last Q_N
0	1	0	1
1	0	1	0
1	1	0	0



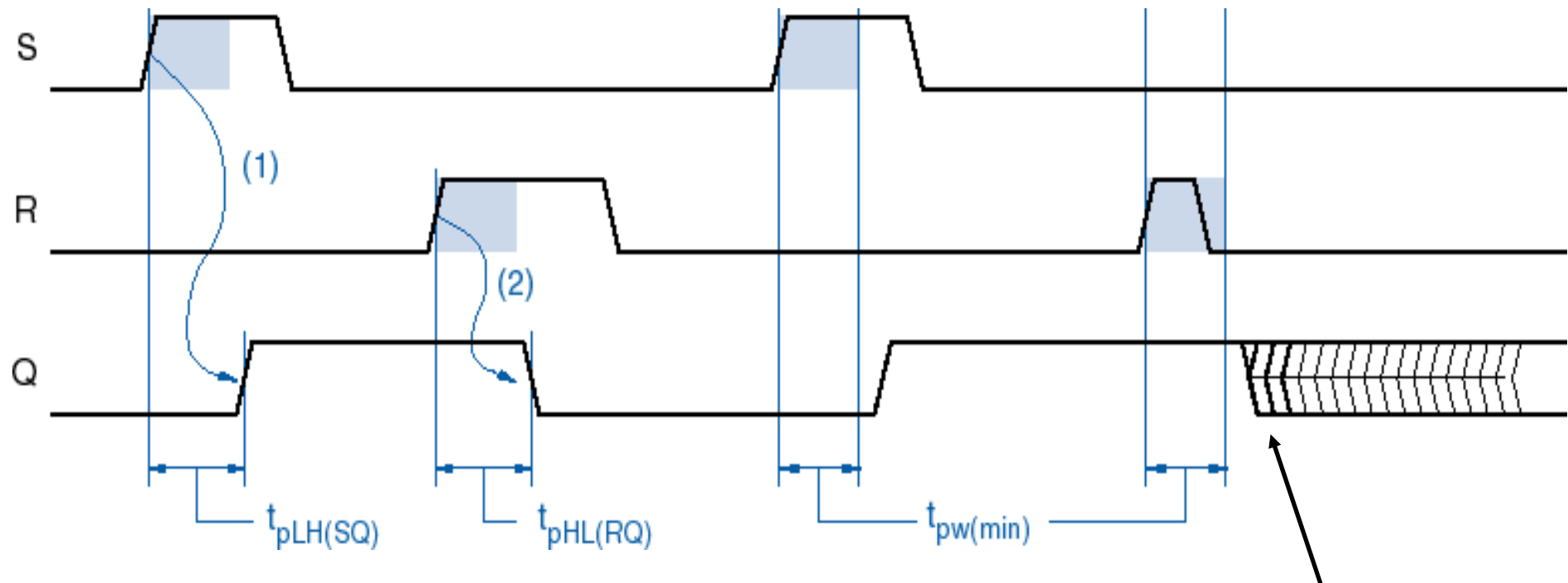
Latch S-R: diagramas temporais



Meta-estabilidade pode ocorrer se S e R forem negados simultaneamente!

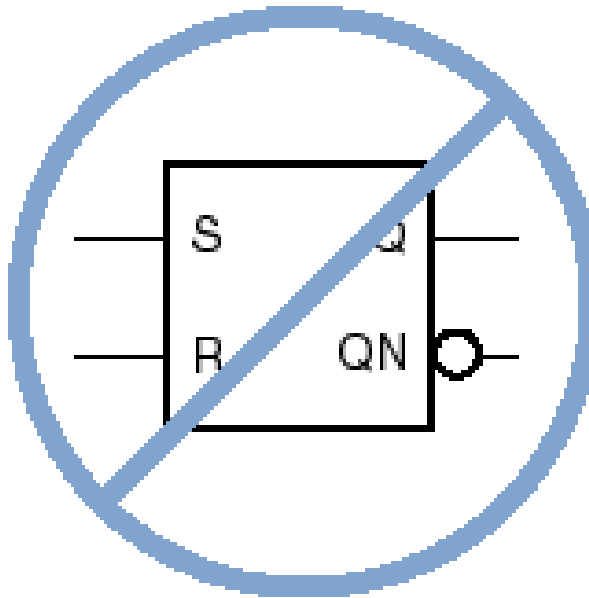
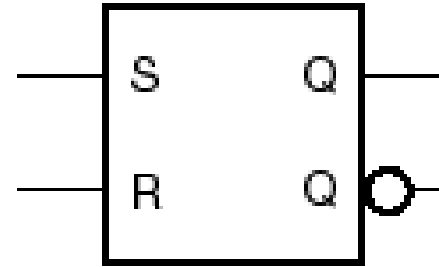
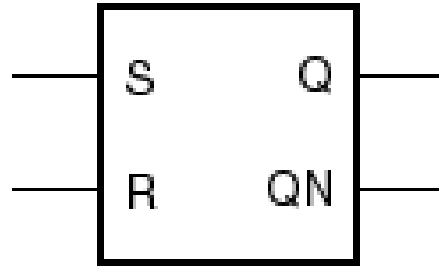


Latch S-R: parâmetros temporais

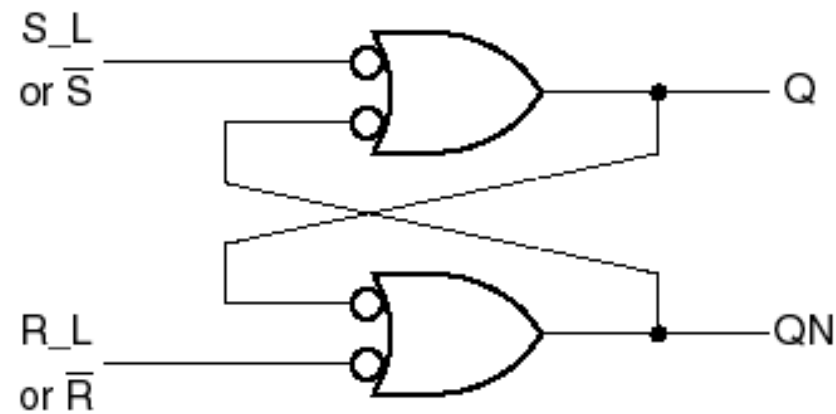


Meta-estabilidade pode ocorrer se o impulso (aqui, R) não atingir a largura mínima!

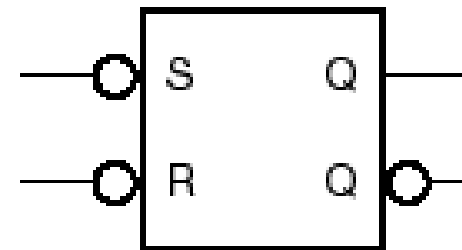
Latch S-R: símbolo



Latch S-R com portas NAND

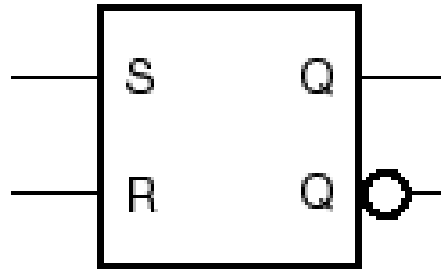


S_L	R_L	Q	QN
0	0	1	1
0	1	1	0
1	0	0	1
1	1	last Q	last QN



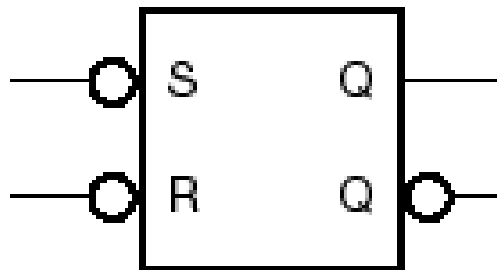
Comparação de Latches S-R

com portas NOR



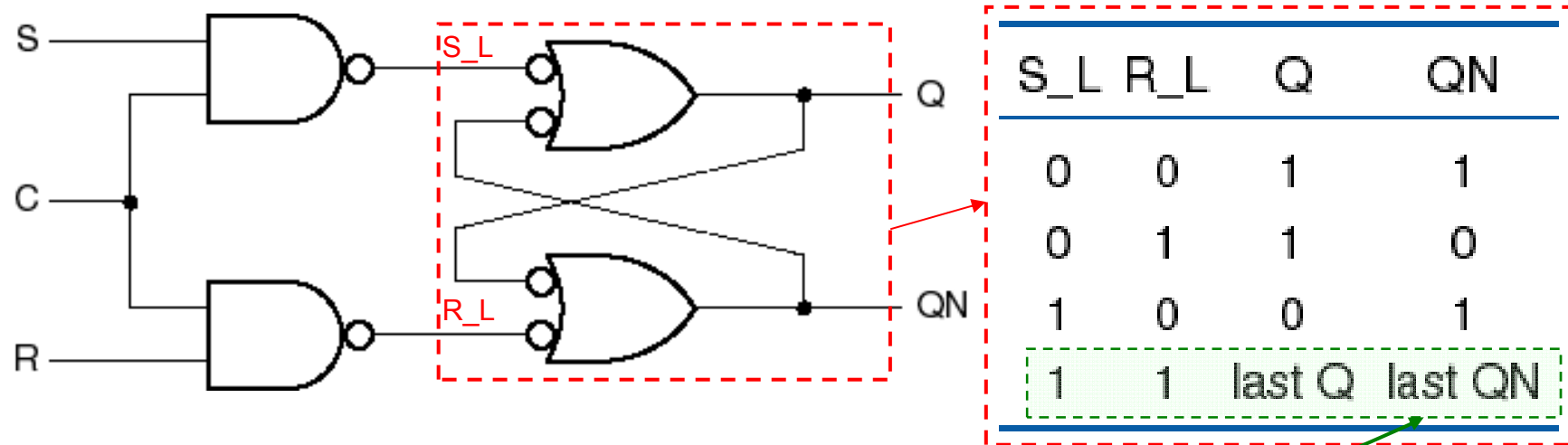
S	R	Q	QN
0	0	last Q	last QN
0	1	0	1
1	0	1	0
1	1	0	0

com portas NAND

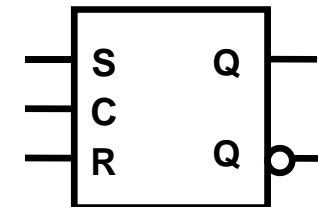


S_L	R_L	Q	QN
0	0	1	1
0	1	1	0
1	0	0	1
1	1	last Q	last QN

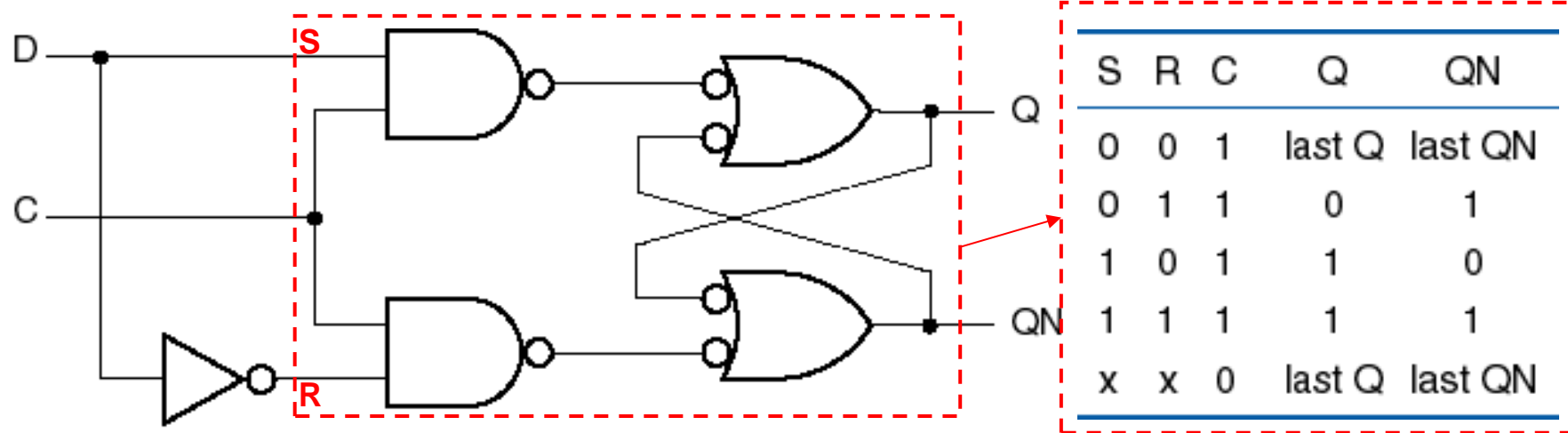
Latch S-R com *enable* (C)



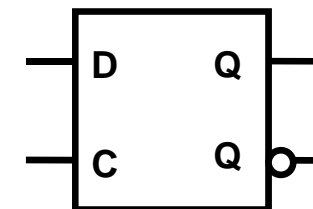
S	R	C	Q	QN
0	0	1	last Q	last QN
0	1	1	0	1
1	0	1	1	0
1	1	1	1	1
x	x	0	last Q	last QN



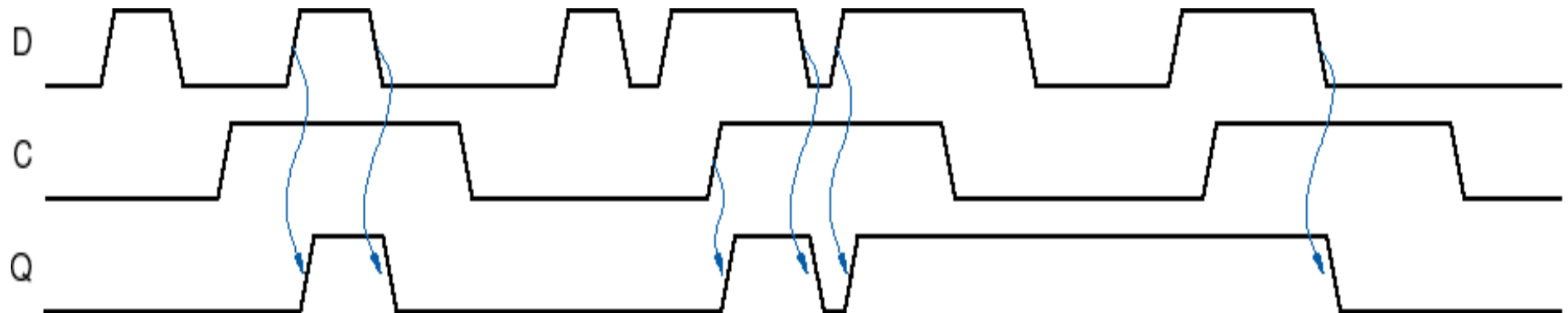
Latch D



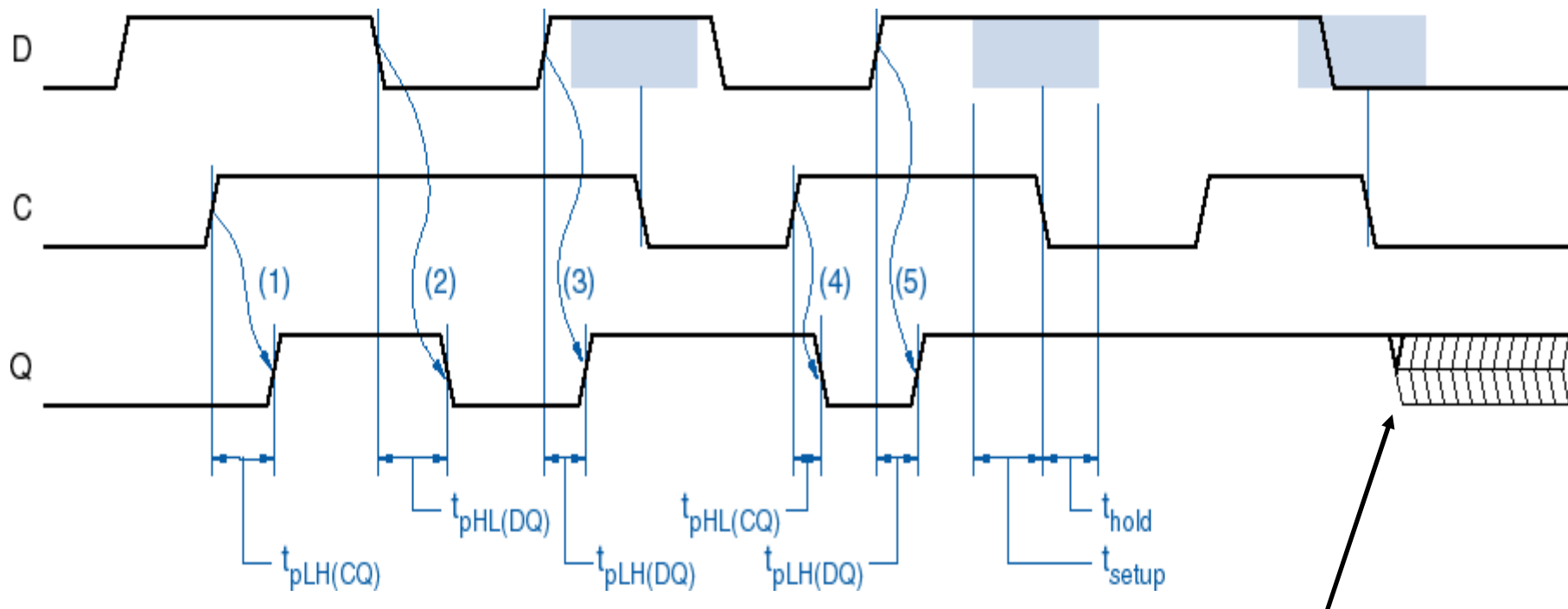
C	D	Q	QN
1	0	0	1
1	1	1	0
0	x	last Q	last QN



Latch D: diagrama temporal



Latch D: parâmetros temporais

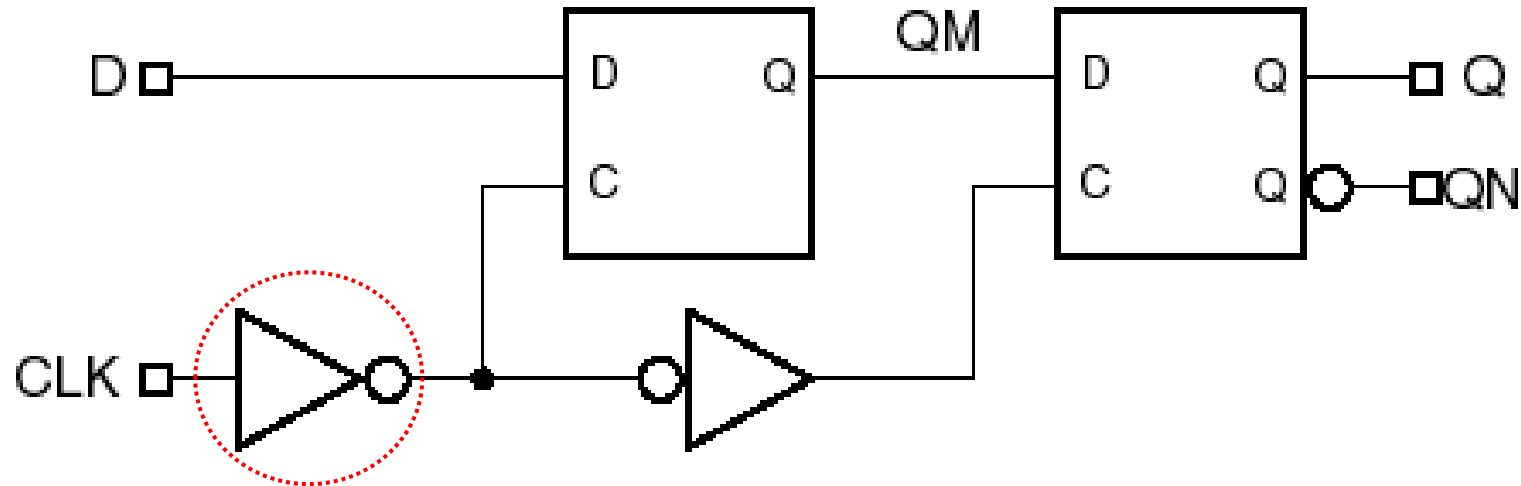


Meta-estabilidade pode ocorrer se D variar muito perto do fecho da latch!

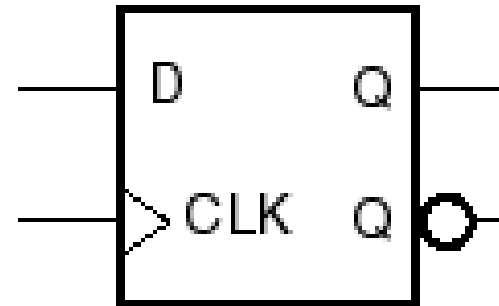
Latches vs. Flip-flops

- **Latch:** analisa as suas entradas **continuamente** e muda as suas saídas **em qualquer instante**.
- **Flip-flop:** normalmente analisa as suas entradas e muda as suas saídas **apenas em instantes determinados** por um sinal de relógio.

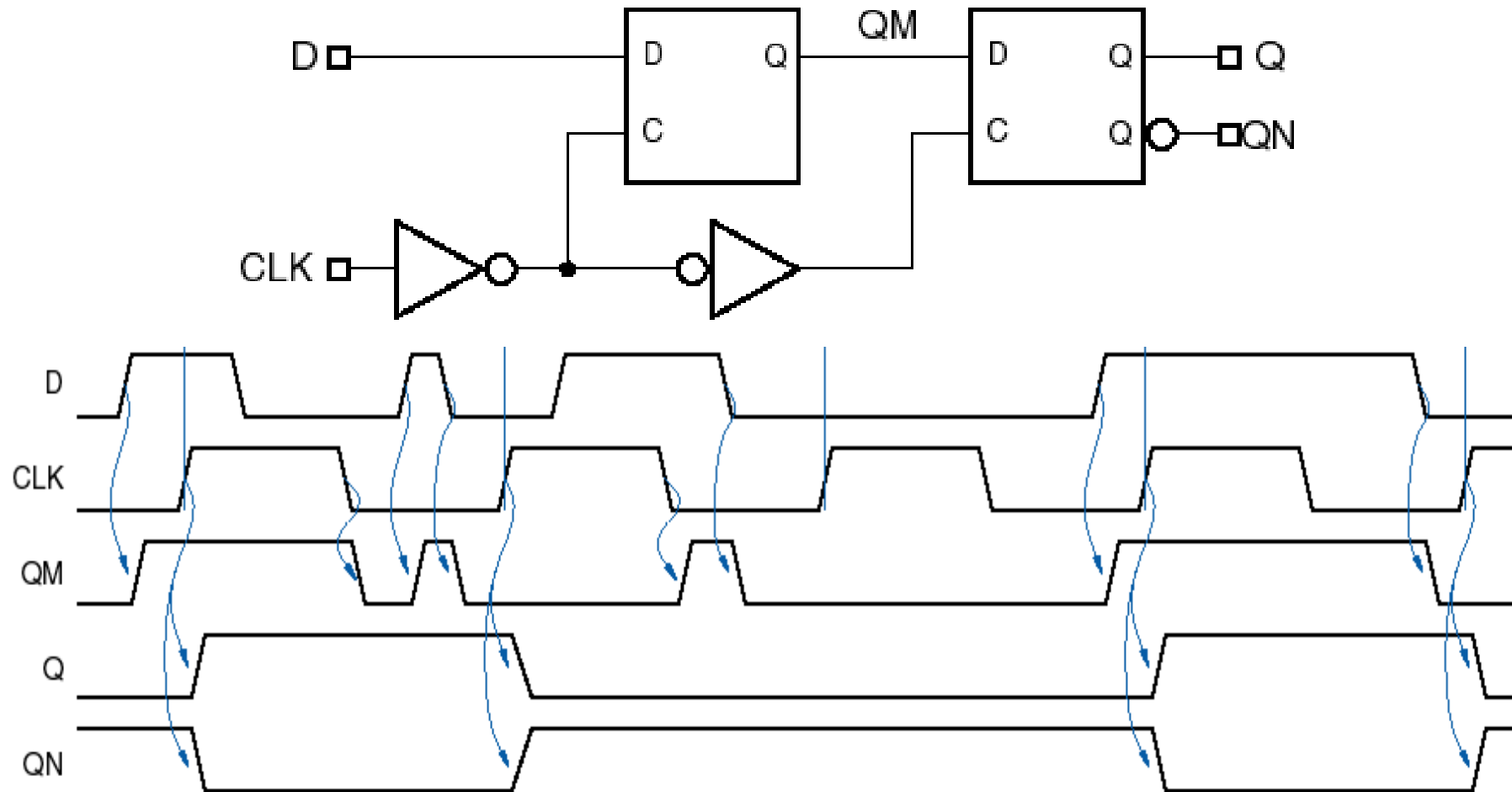
Flip-flop D (*edge-triggered*)



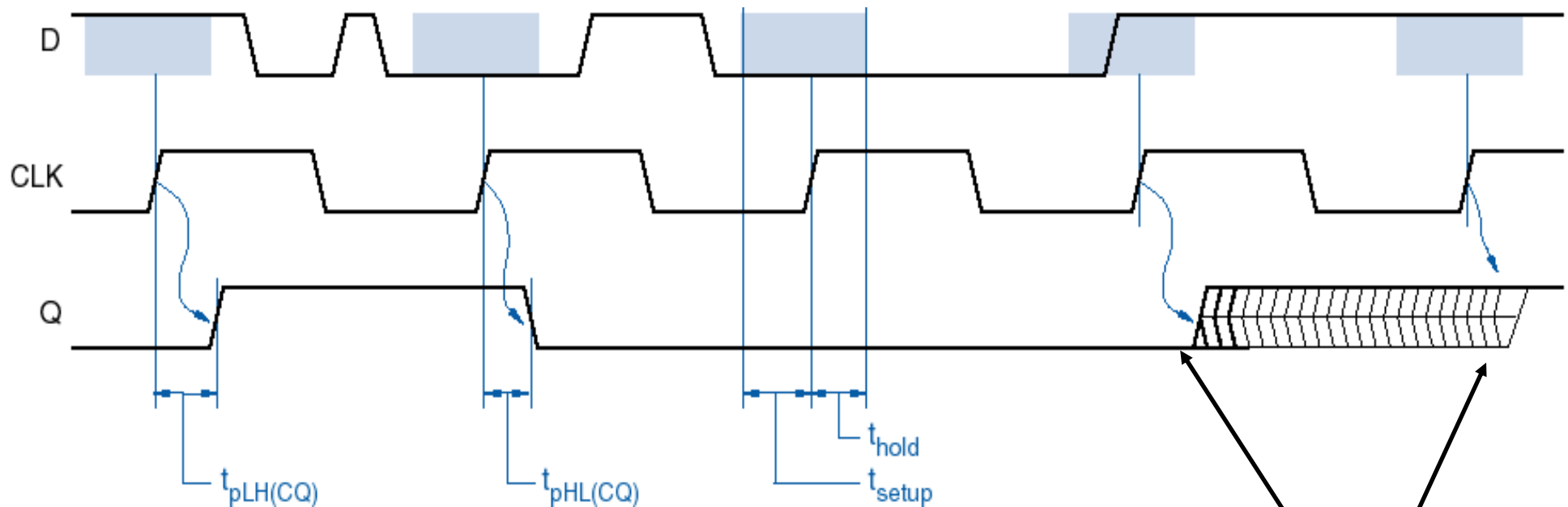
D	CLK	Q	QN
0		0	1
1		1	0
x	0	last Q	last QN
x	1	last Q	last QN



Flip-flop D: Diagramas temporais



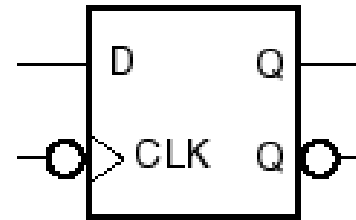
Flip-flop D: Parâmetros temporais



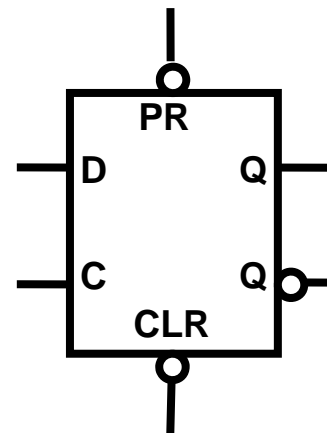
**Meta-estabilidade pode ocorrer
se D variar muito perto do fecho da latch master...
...Mas a situação pode ser corrigida**

Variantes do Flip-flop D

- *Negative-edge-triggered*

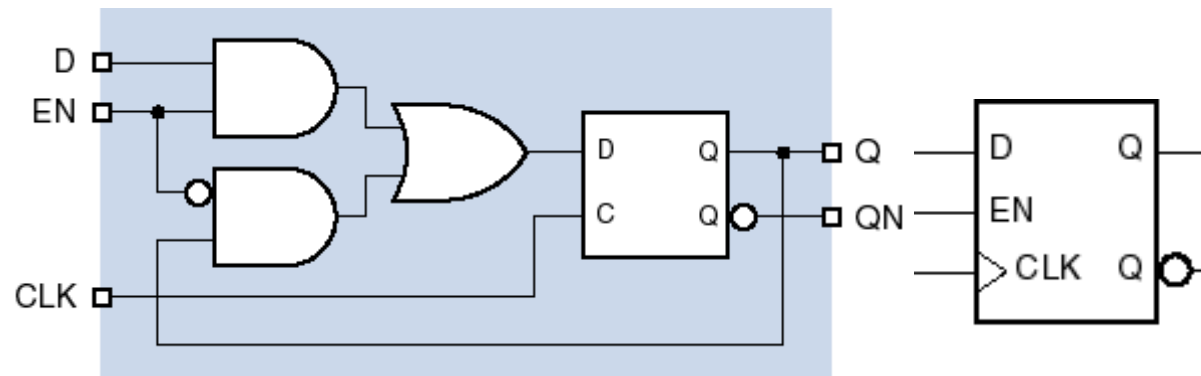


- Com entradas assíncronas

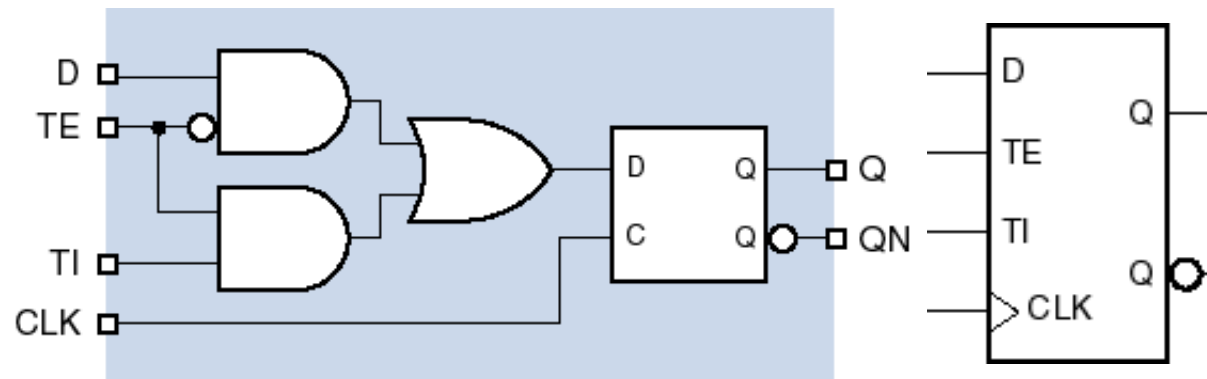


Variantes do Flip-flop D

- Com *clock enable* (EN)



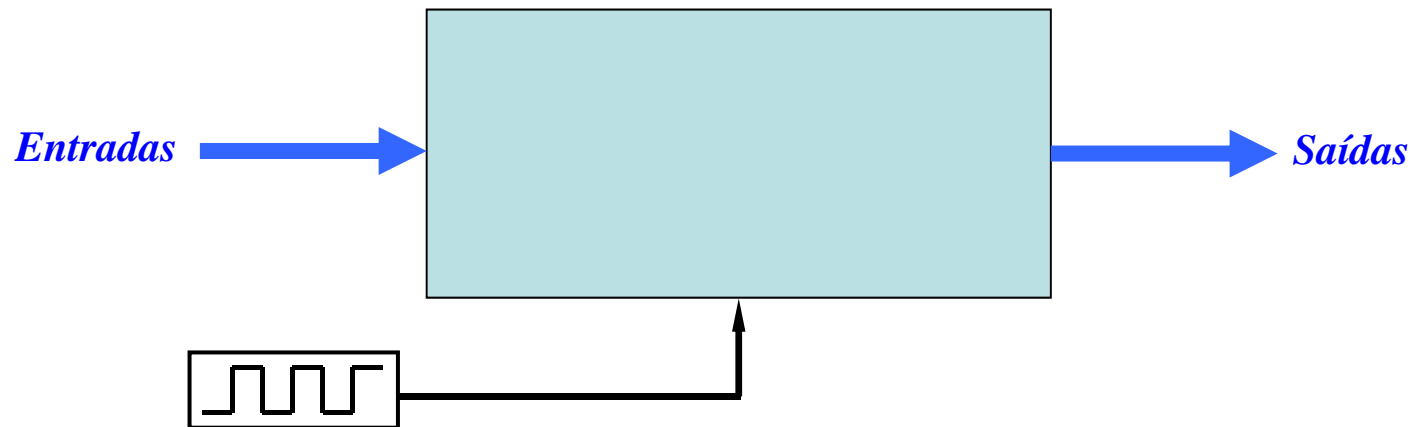
- De teste (*Scan*)



Caraterização de flip-flops

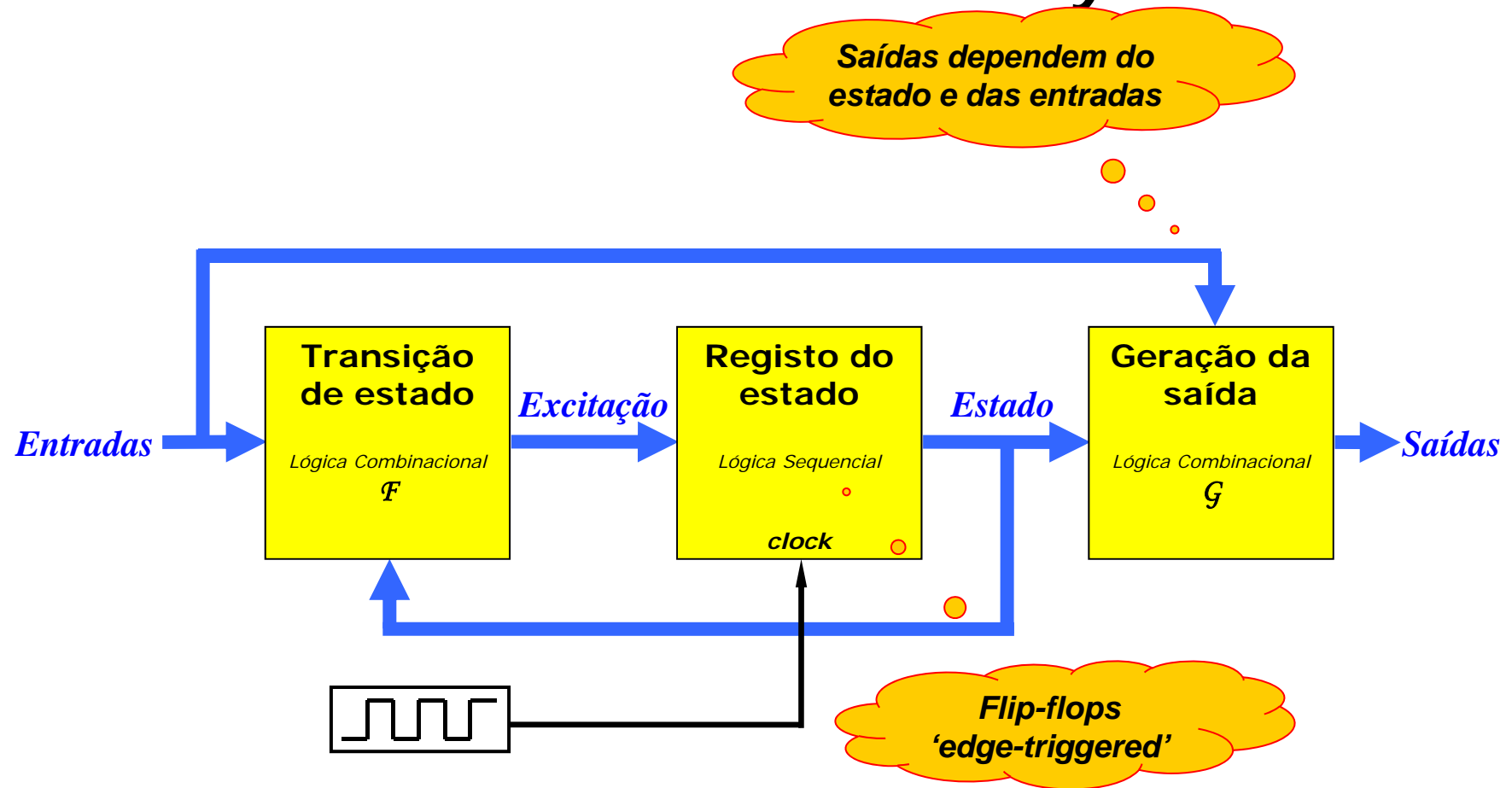
- Tabelas de funcionamento
- Tabelas de transições
 - Equações caraterísticas
- Tabelas de excitação
- Diagramas de estado

Circuitos sequenciais síncronos (ou *máquinas de estados finitos*)



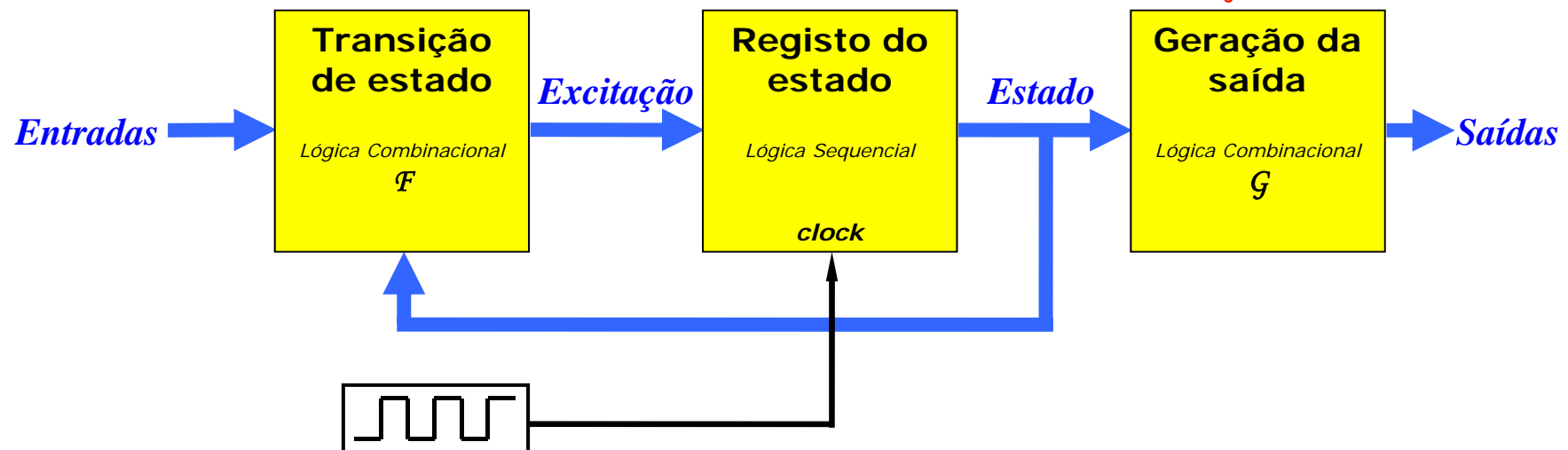
- Modelos estruturais: *Mealy*, *Moore*...
- Metodologia de análise
- Metodologia de síntese: diagramas de estados...

Máquina de estados finitos: estrutura de *Mealy*

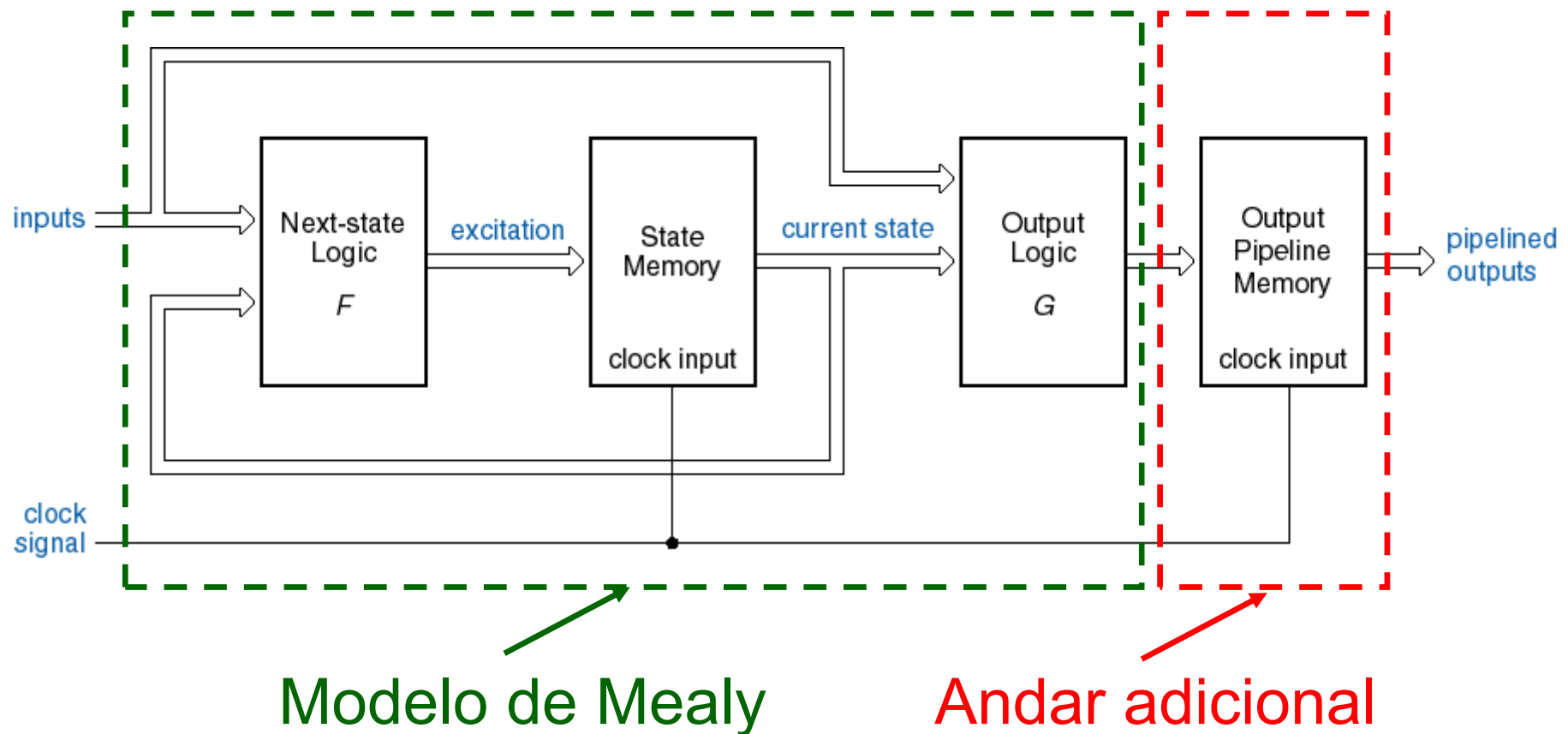


Máquina de estados finitos: estrutura de *Moore*

Saídas dependem
apenas do estado



Variante: saídas em *pipeline*



Máquinas de estados: metodologia de análise

Etapa 1.

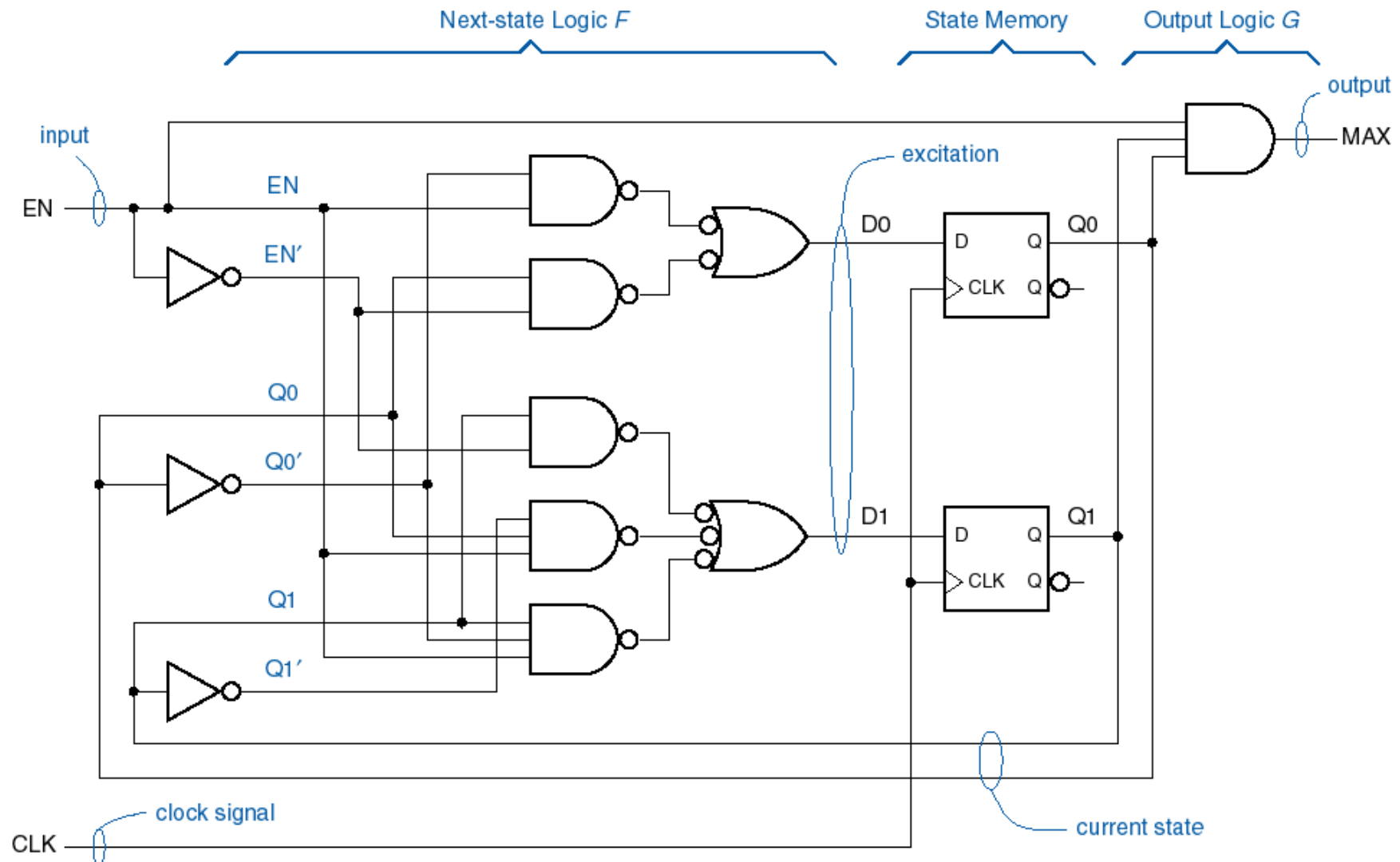
- a) Determinar a função F (*lógica de transição*).
- b) Usando a equação característica dos FF, deduzir as **equações de transição** de estados (trivial com FF do tipo D).
- c) Determinar a função G (*lógica de saída*).

Etapa 2.

- a) Construir **tabela de transição** de estados
 - Para cada combinação estado/entrada, indicar o estado seguinte.
- b) Construir **tabela de saídas**
 - Para cada combinação estado/entrada, indicar os valores de saída (pode ser combinada com a tabela de transição de estados)

Etapa 3. Desenhar **diagrama de estados**

Análise de máquinas de estados: exemplo



Etapa 1

- Equações de excitação (bloco combinatório F)

$$D0 = Q0 \cdot EN' + Q0' \cdot EN$$

$$D1 = Q1 \cdot EN' + Q1' \cdot Q0 \cdot EN + Q1 \cdot Q0' \cdot EN$$

- Equações características (dos flip-flops usados)

$$Q0^* = D0$$

$$Q1^* = D1$$

- Equações de transição

$$Q0^* = Q0 \cdot EN' + Q0' \cdot EN$$

$$Q1^* = Q1 \cdot EN' + Q1' \cdot Q0 \cdot EN + Q1 \cdot Q0' \cdot EN$$

- Equações de saída

$$MAX = Q1 \cdot Q0 \cdot EN$$

Etapa 2: Tabelas

Equações de transição:
$$\begin{cases} Q0^* = Q0 \cdot EN' + Q0' \cdot EN \\ Q1^* = Q1 \cdot EN' + Q1' \cdot Q0 \cdot EN + Q1 \cdot Q0' \cdot EN \end{cases}$$

Equação de saída: $MAX = Q1 \cdot Q0 \cdot EN$

<i>EN</i>		
<i>Q1 Q0</i>	<i>0</i>	<i>1</i>

<i>EN</i>		
<i>S</i>	<i>0</i>	<i>1</i>

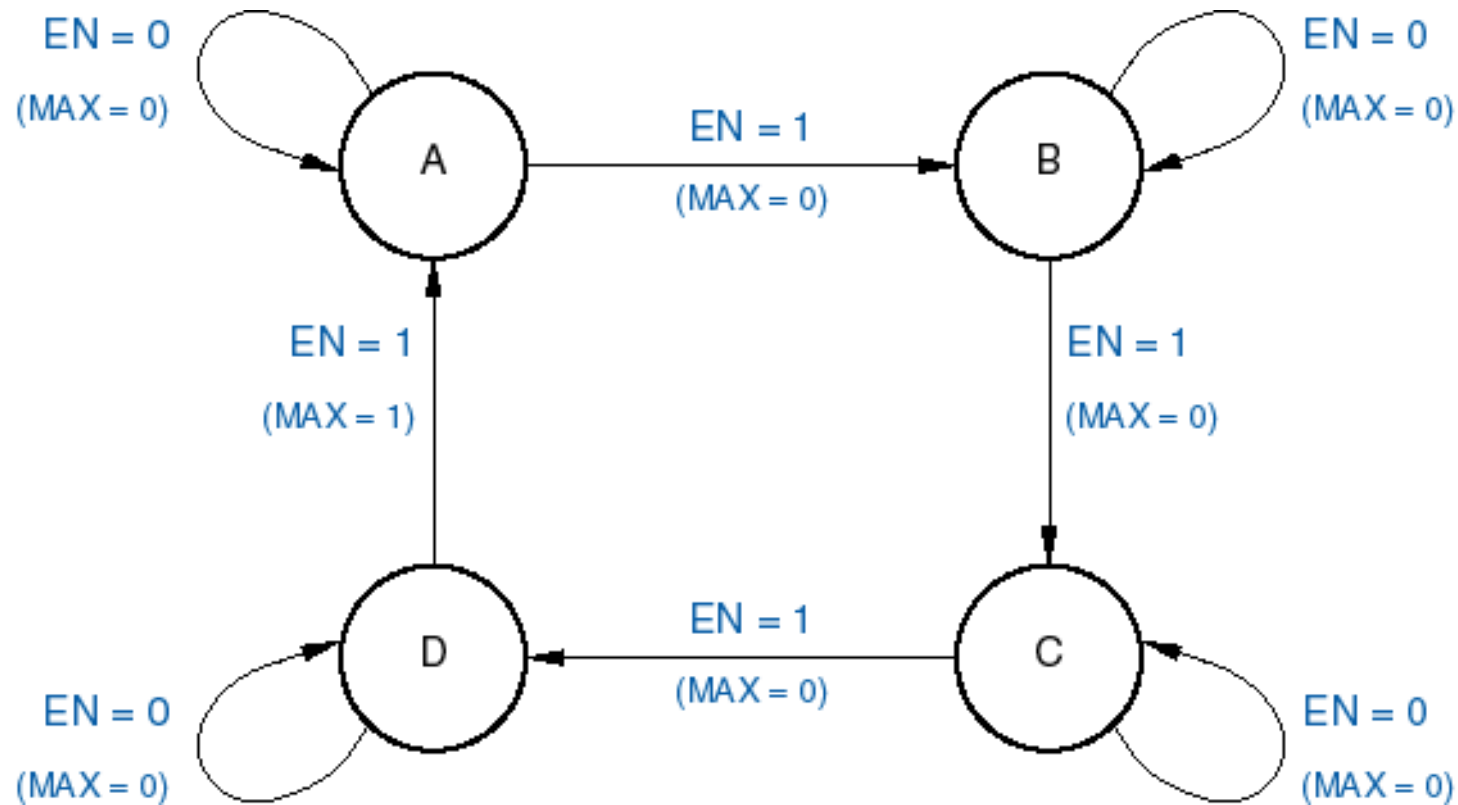
<i>EN</i>		
<i>S</i>	<i>0</i>	<i>1</i>

Tabela de transições

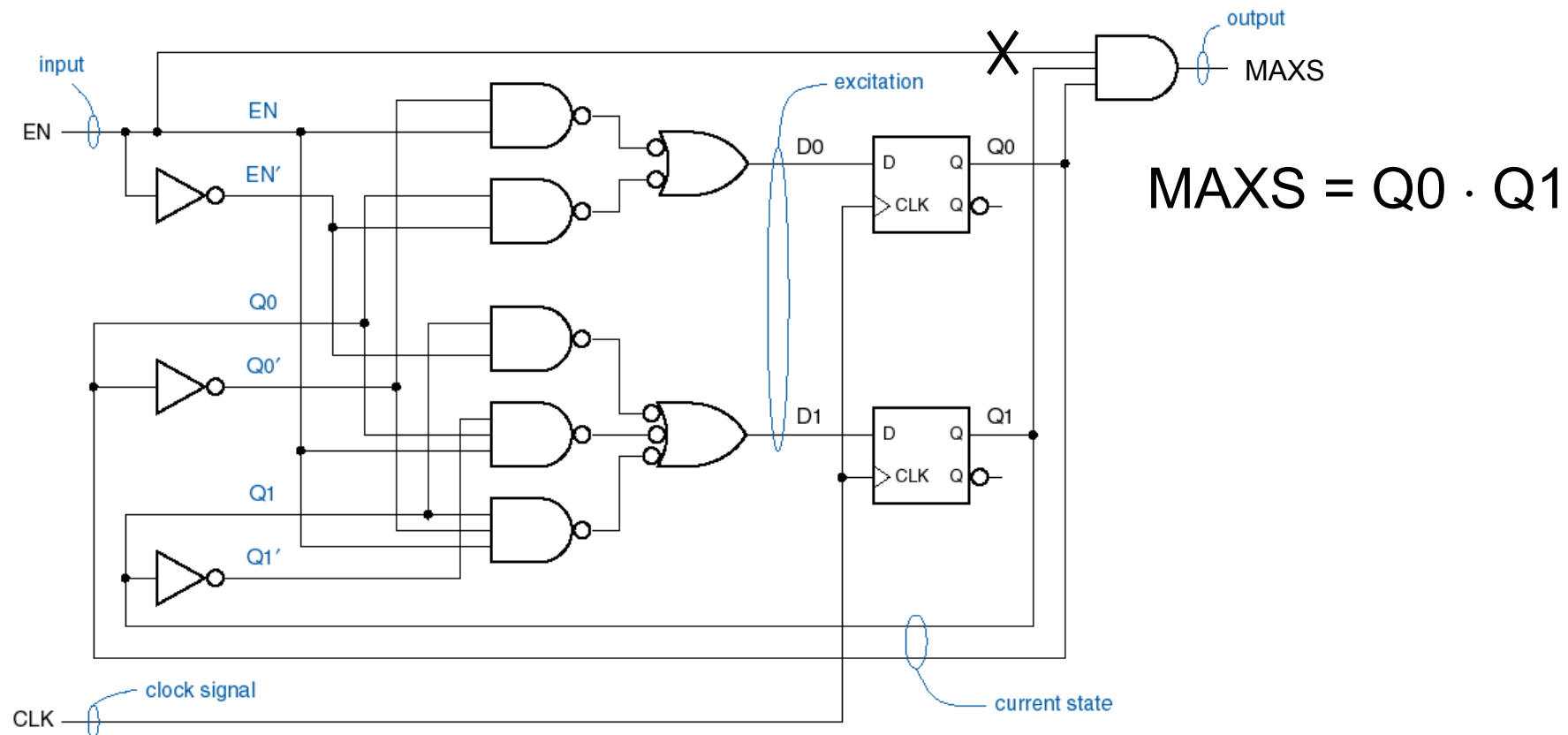
Tabela de estados

Tabela de estados/saídas

Etapa 3: Diagrama de estados

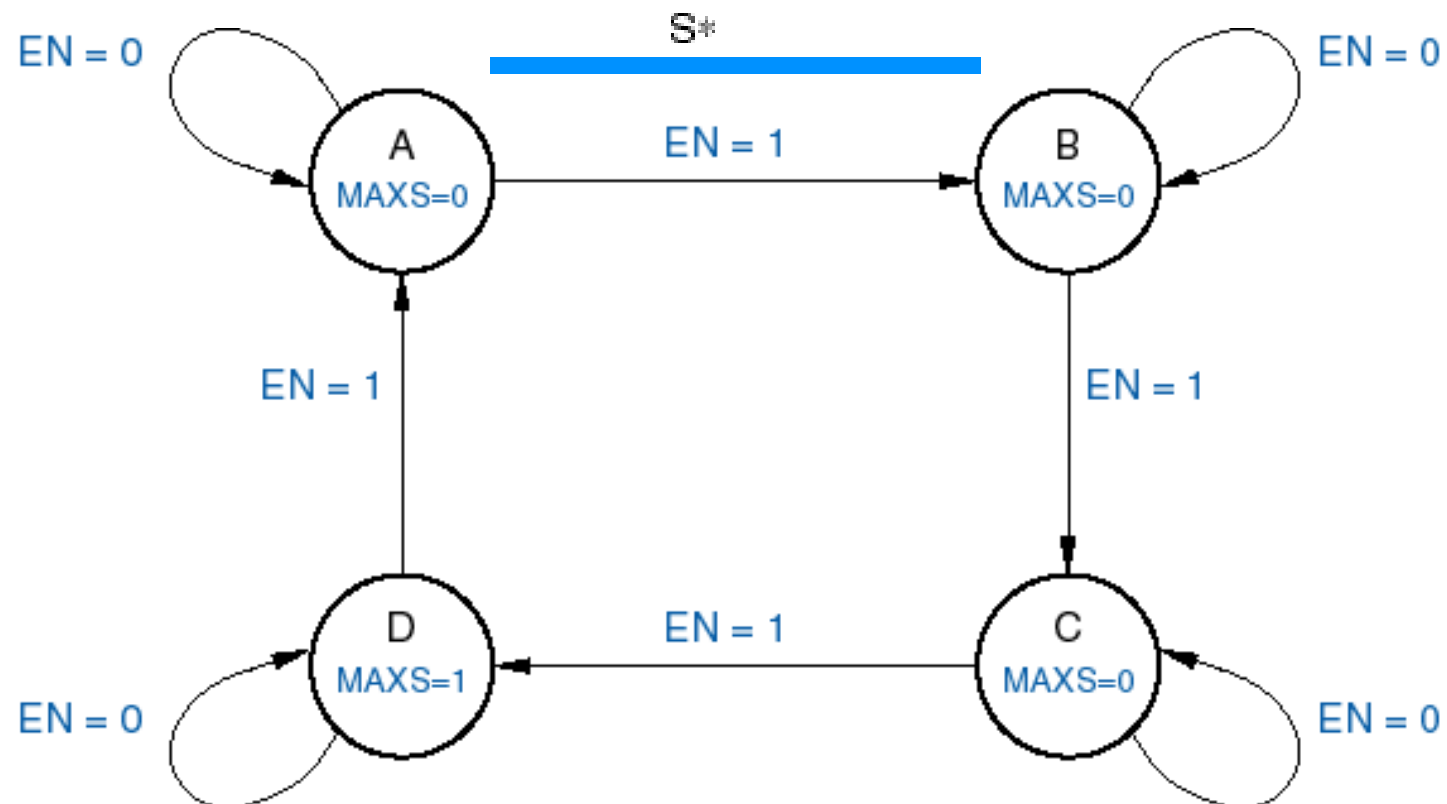


Variação ao exemplo anterior

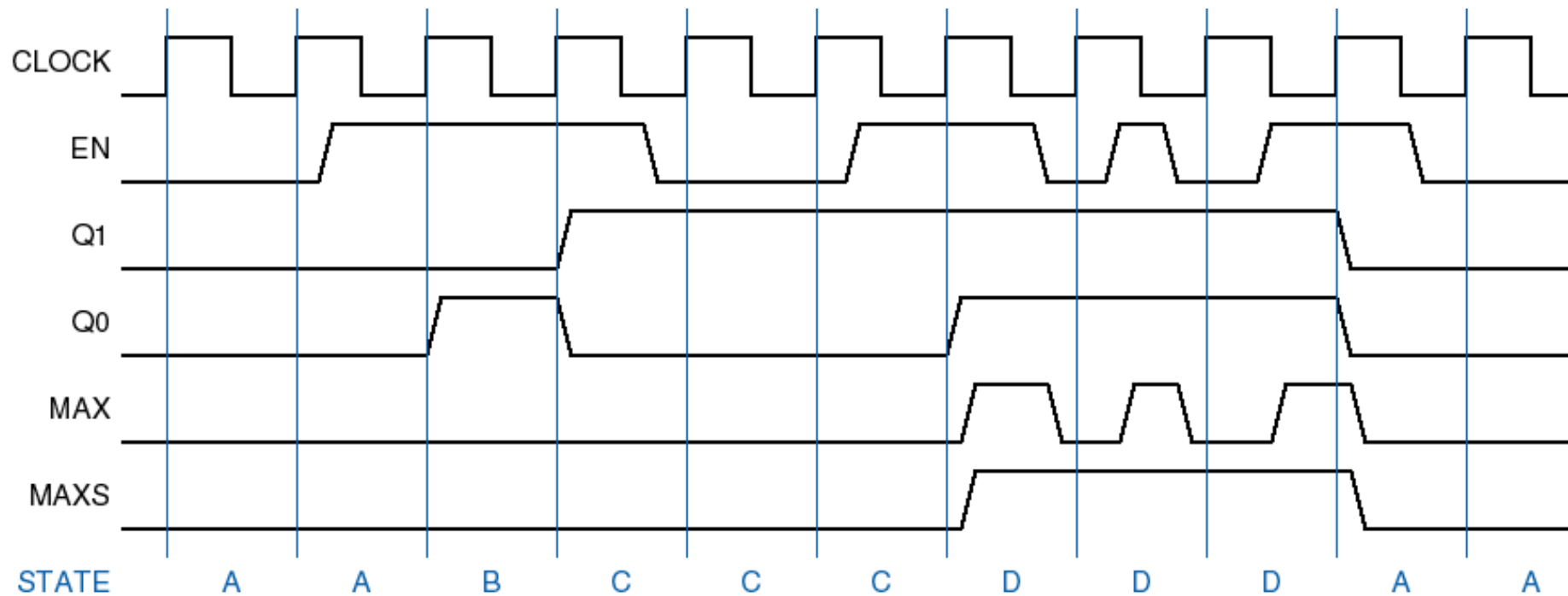


- Transformou-se numa máquina de *Moore*

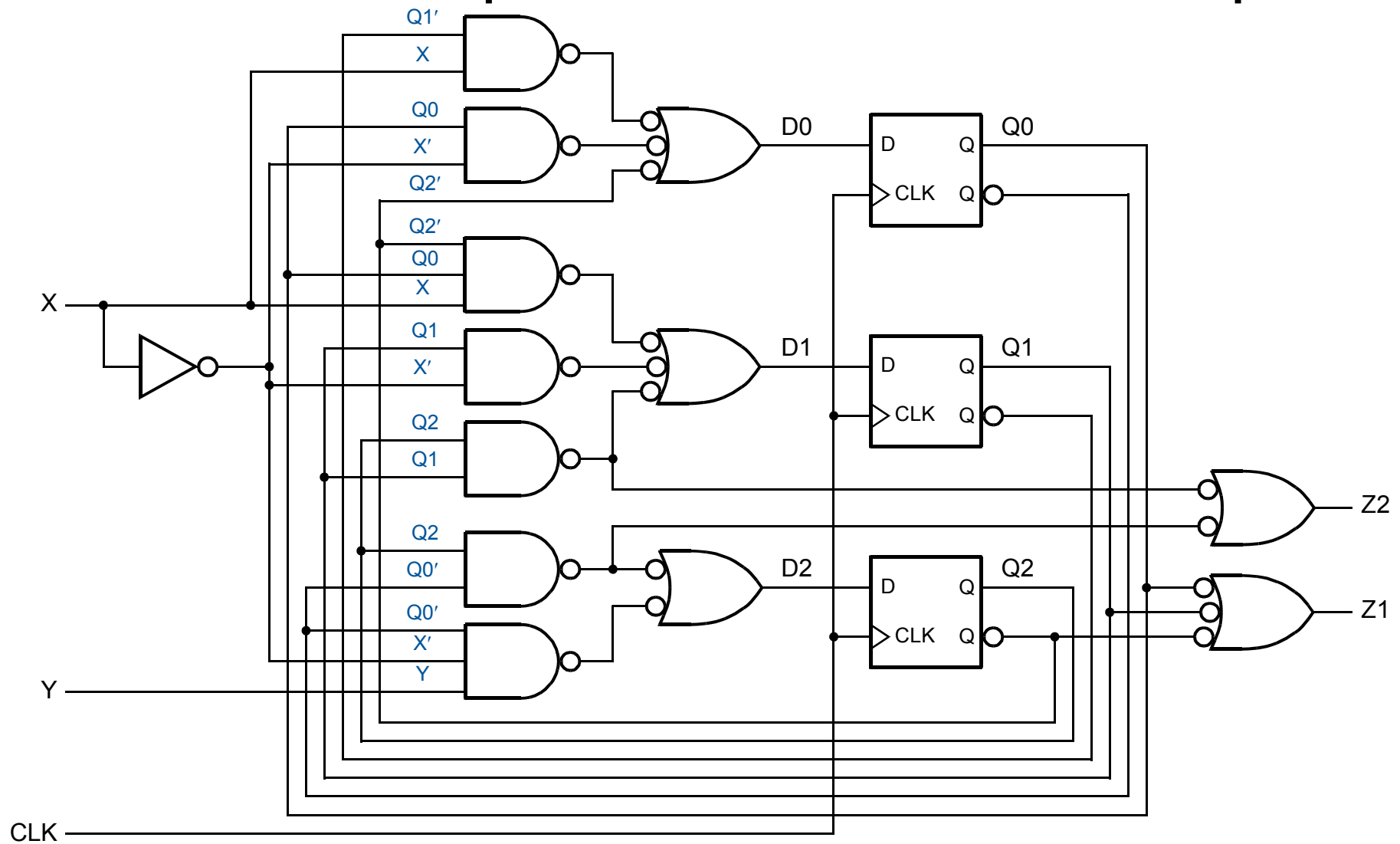
<i>S</i>	<i>EN</i>		<i>MAXS</i>
	<i>0</i>	<i>1</i>	
A	A	B	0
B	B	C	0
C	C	D	0
D	D	A	1



Diagramas temporais



Análise de máquinas de estados: exemplo II



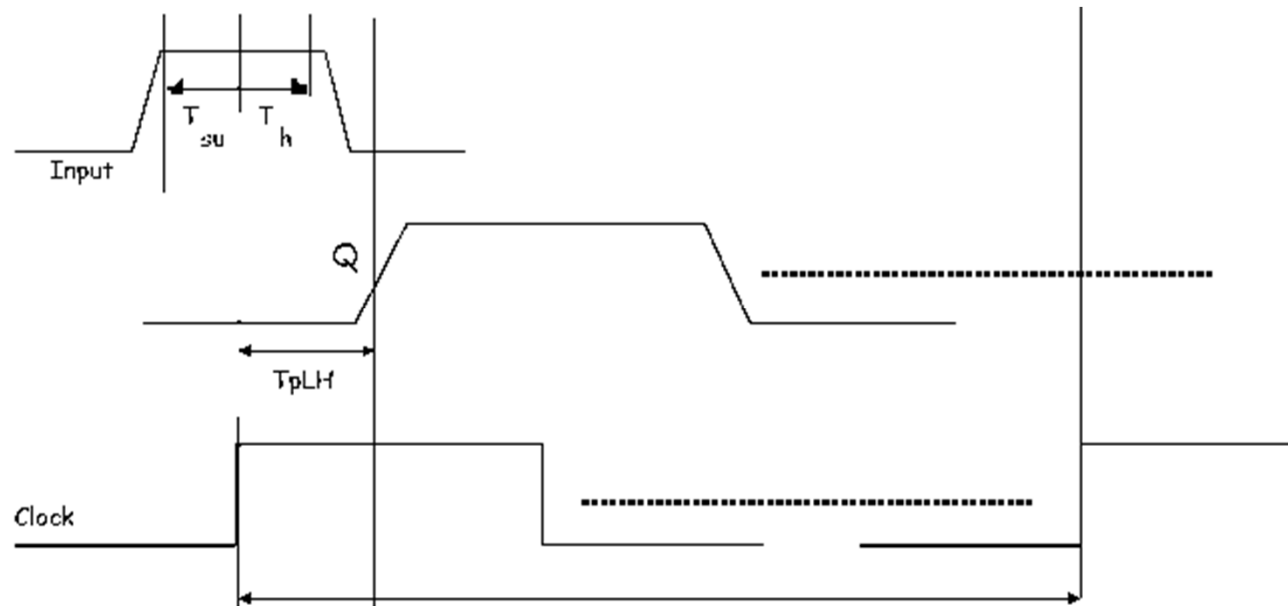
Limitações temporais

Dadas especificações temporais dos elementos de memória e dos tempos de propagação dos elementos combinatórios coloca-se o problema de determinar qual a frequência máxima de funcionamento dum sistema sequencial síncrono.

Parâmetros temporais:

- flip-flops: T_{setup} , T_{hold} , $\max(T_{\text{pHL}}, T_{\text{pLH}})$
- lógica para cálculo do estado seguinte: T_p

Limitações temporais



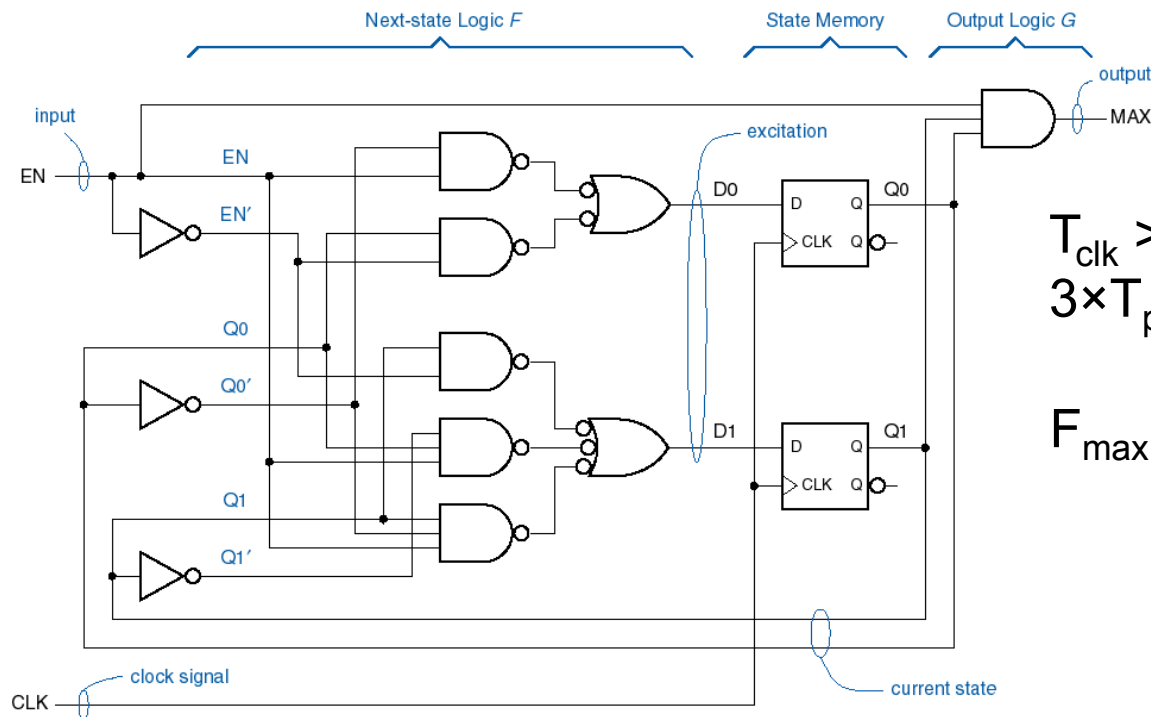
$$T_{clk} > T_{setup} + \max(T_{pHL}, T_{pLH}) + T_p$$

$$T_{hold} < \min(T_{pHL}, T_{pLH}) + T_p$$

Limitações temporais

Exemplo I:

- $T_{\text{setup}} = 10\text{ns}$, $T_{\text{hold}} = 5\text{ns}$, $T_{\text{pHL}} = T_{\text{pLH}} = 20\text{ ns}$
- T_p portas elementares = 10 ns

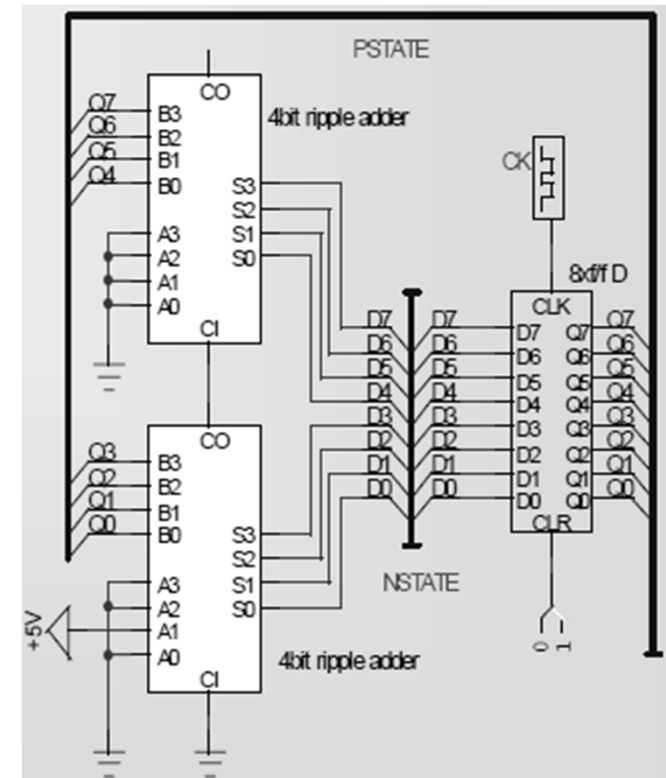


$$T_{\text{clk}} > T_{\text{setup}} + \max(T_{\text{pHL}}, T_{\text{pLH}}) + 3 \times T_p = 10 + 20 + 30 = 60\text{ ns} = T_{\text{min}}$$

$$F_{\text{max}} = 1/T_{\text{min}} = 16.7\text{ MHz}$$

Exercício

- Mostre que o diagrama de estados do sistema da figura corresponde a um funcionamento periódico com 128 estados.
- Se o tempo de atraso de cada etapa elementar de soma for de 20 ns calcule a frequência máxima de funcionamento do sistema. Admita $T_{\text{setup}} = T_{\text{hold}} = 5 \text{ ns}$ e $T_{\text{pHL}} = T_{\text{pLH}} = 15 \text{ ns}$.
- Repita o cálculo admitindo que a estrutura de soma é do tipo “*carry lookahead*”



Máquinas de estados finitos: metodologia de projeto/síntese

1. Construir tabela de estados/saídas e/ou diagrama de estados
Mnemónicas para designar os estados
 2. Minimizar número de estados
 3. Escolher variáveis de estado e decidir codificação dos estados
-
4. Construir tabela de transições/saídas
 5. Escolher tipo de flip-flop para o registo de estado
 6. Construir tabela de excitação
 7. Deduzir funções de excitação
Na forma mínima: mapas de Karnaugh...
 8. Deduzir funções de saída
Na forma mínima: mapas de Karnaugh...
 9. Desenhar diagrama lógico

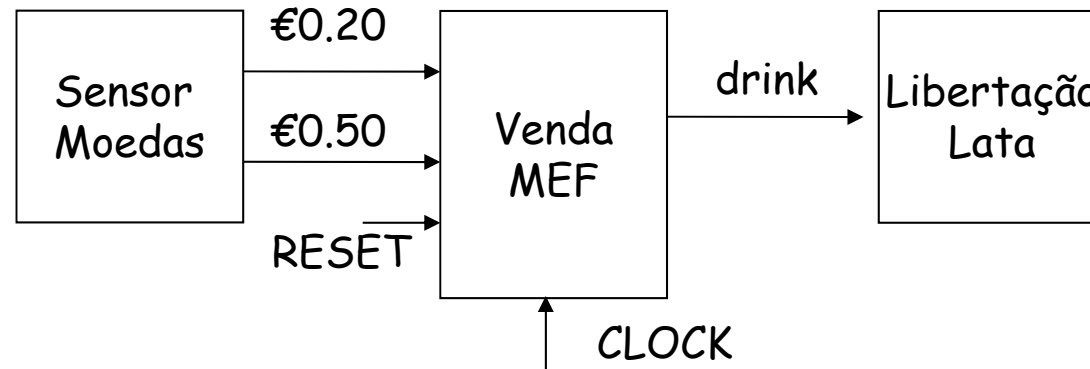
Exemplo 1

- Máquina de venda de bebidas

- Requisitos gerais:

- entrega lata de cerveja (sem álcool) após depósito de € 0.60
 - uma única entrada para moedas de € 0.20 e € 0.50
 - a máquina não dá troco

- Passo 1: perceber o problema (fazer um desenho!...)

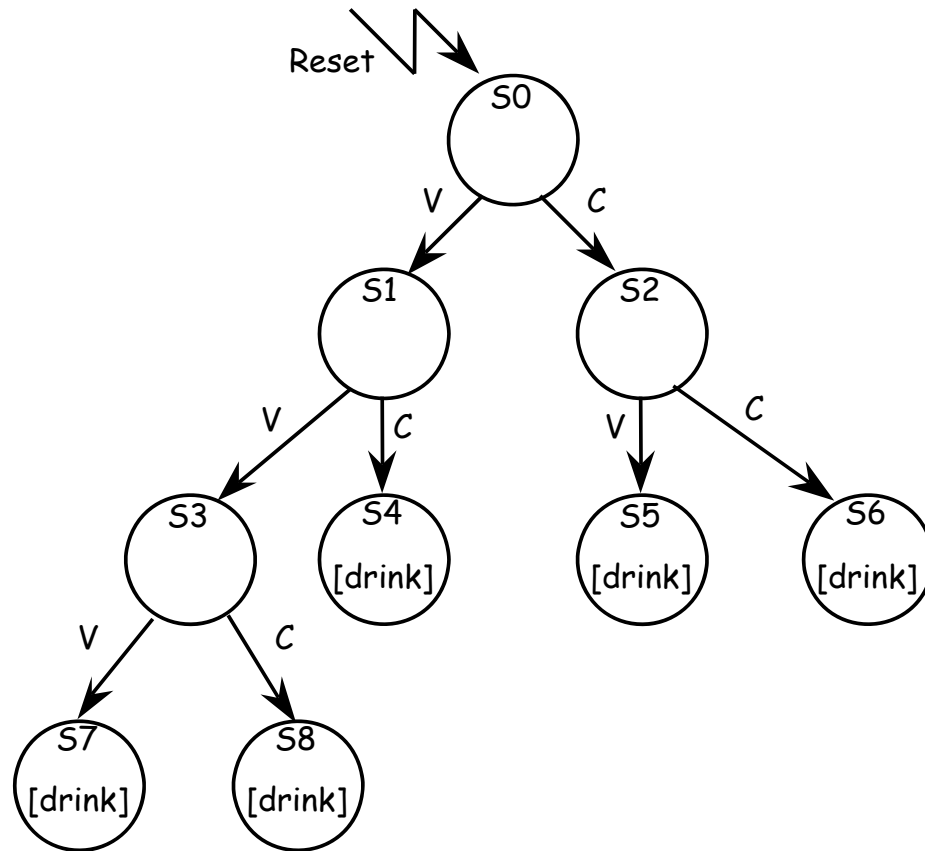


Exemplo 1 – Análise de requisitos

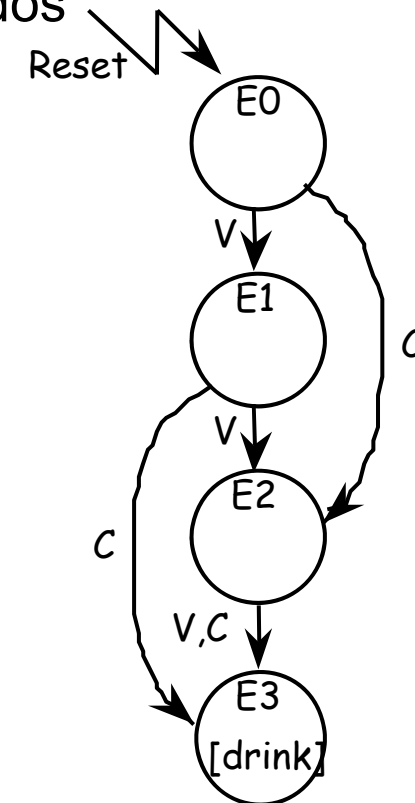
- Começar por identificar as sequências de entradas que levam diretamente à abertura
 - 3 moedas de €0.20
 - 1 moeda de €0.20 + 1 moeda de €0.50
 - 1 moeda de €0.50 + 1 moeda de €0.20
 - 2 moedas de €0.50
 - 2 moedas de €0.20 + 1 moeda de €0.50
- Identificar entradas e saídas:
 - Entradas:
 - V (sensor ativo para €0.20)
 - C (sensor ativo para €0.50)
 - Saída
 - drink

Exemplo 1 – Diagrama de Estados

- Diagrama de estados primário

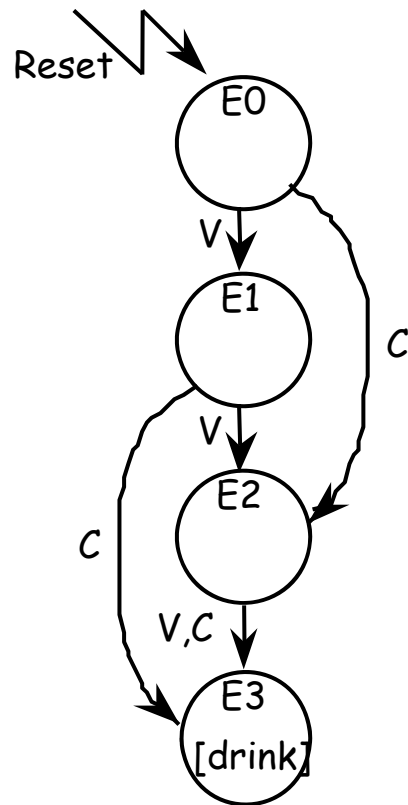


- Diagrama de estados correto com reutilização de estados



Exemplo 1 – Tabela de Estados

- Tabela de Estados/Saídas decorre diretamente do Diagrama de Estados



Cstate	Inputs		Nstate	Output
	V	C		
E0	0	0	E0	0
	0	1	E2	0
	1	0	E1	0
	1	1	X	X
E1	0	0	E1	0
	0	1	E3	0
	1	0	E2	0
	1	1	X	X
E2	0	0	E2	0
	0	1	E3	0
	1	0	E3	0
	1	1	X	X
E3	X	X	E3	1

Exemplo 1 – Tabela de Transições

- Codificação dos estados
 - State Assignment
- Processo arbitrário *a priori*
- Eg:
 - E0 = 00
 - E1 = 01
 - E2 = 10
 - E3 = 11
- Substituindo na Tabela de Estados os símbolos de cada estado pela respetiva codificação obtemos a Tabela de Transições/Saídas

Cstate Q1Q0	Inputs		Nstate (Q1Q0)*	OPEN
	V	C		
00	0	0	00	0
	0	1	10	0
	1	0	01	0
	1	1	X	X
01	0	0	01	0
	0	1	11	0
	1	0	10	0
	1	1	X	X
10	0	0	10	0
	0	1	11	0
	1	0	11	0
	1	1	X	X
11	X	X	11	1

Exemplo 1 – Equações

- Exercício: deduza a partir dos mapas de Karnaugh que

$$D_0 = V \oplus Q_0 + Q_1 \cdot (C + V)$$

$$D_1 = Q_1 + C + V \cdot Q_0$$

$$drink = Q_1 \cdot Q_0$$

Codificação de estados

- N° de estados: q
- N° mínimo de variáveis: $m \in \mathbb{N} : \log_2 q \leq m < \log_2 q + 1$
- Um número superior a m pode ser vantajoso...
- Neste exemplo: 4 estados, 2 variáveis de estado (Q1,Q0):

E0 ==> 00

E1 ==> 01

E2 ==> 10

E3 ==> 11

Codificação: diretrizes gerais

- Escolher um código para o estado inicial que facilite a operação de RESET (normalmente 00..00 ou 11..11)
- Minimizar o número de variáveis que variam em cada transição de estado
- Em grupos de estados relacionados, manter o máximo possível de variáveis constantes.
- Atribuir códigos próximos a estados funcionalmente 'parecidos'.
- Aproveitar a liberdade de escolha quando há estados não usados.
- Decompor as variáveis de estado em grupos com significado funcional bem definido relativamente a entradas e/ou saídas.
- Avaliar eventuais vantagens de usar um número de variáveis superior ao mínimo.

Estados não usados

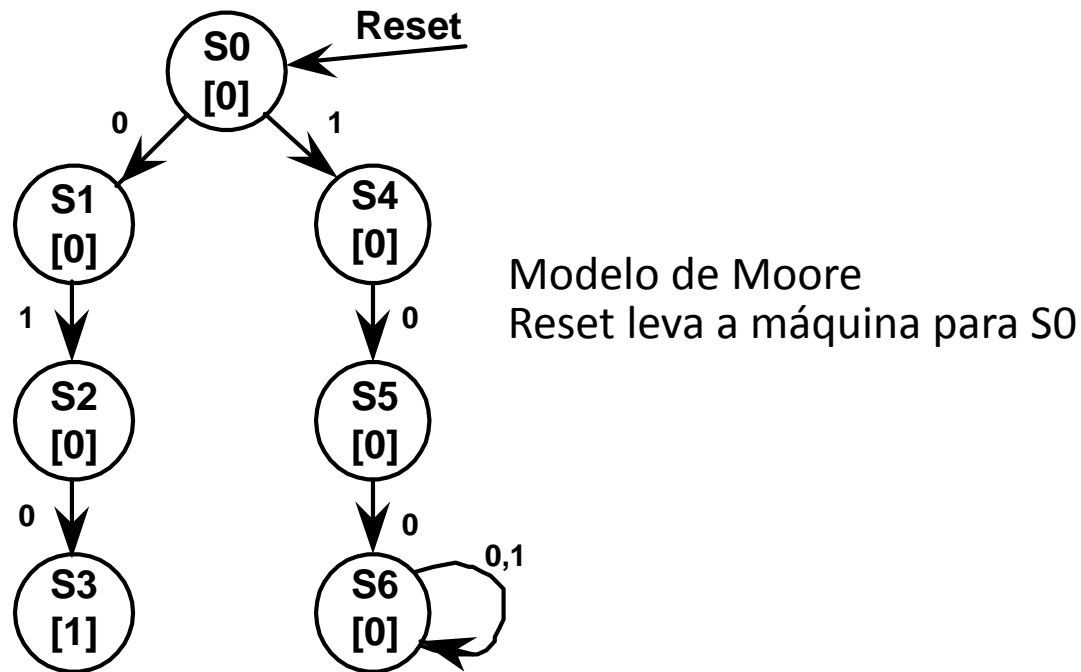
- Critério de **risco mínimo**
 - especificar estados seguintes (possivelmente estado inicial) para precaver situações anómalas => circuitos de excitação mais condicionados e por isso mais caros.
- Critério de **custo mínimo**
 - não especificar estado seguinte; tirar partido das irrelevâncias para minimizar lógica de excitação => diminuição de custo.

Exemplo 2: Detetor de Sequências

- Reconhecimento de padrões em frases de comprimento finito:
 - Um reconhecedor de frases finitas tem uma entrada (X) e uma saída (Z). A saída é ativa sempre que a sequência de entrada ...010... é observada, desde que a sequência 100 nunca tenha surgido.
 - Exemplo do comportamento entrada/saída:
 - X: 00101010010...
 - Z: 00010101000...

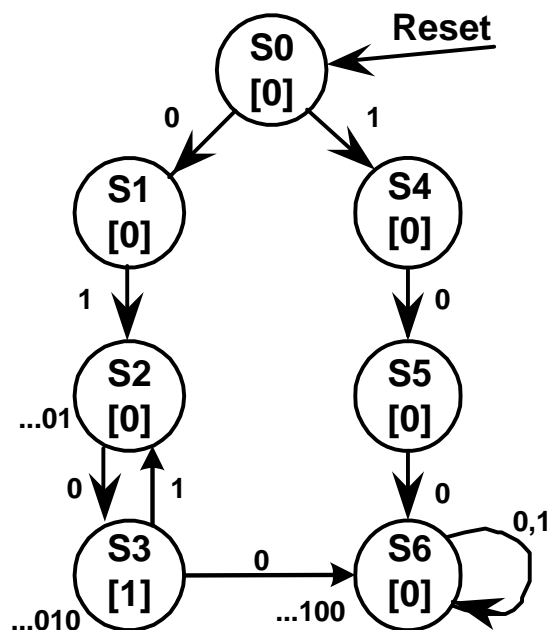
Exemplo 2: Diagrama de Estados

- Desenhar o diagrama de estados para os padrões que devem ser reconhecidos i.e., 010 e 100.

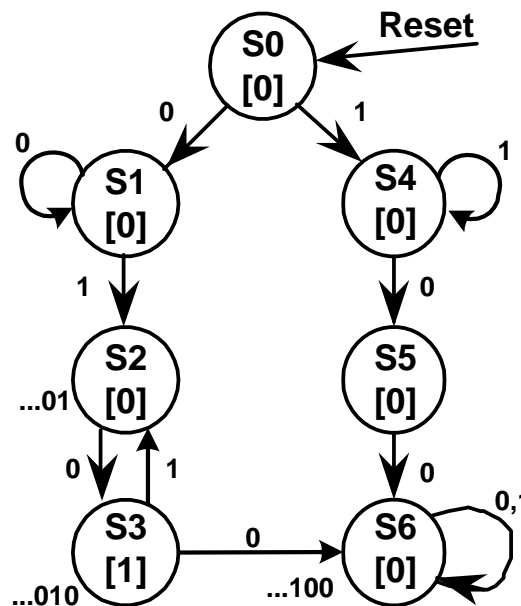


Exemplo 2: Diagrama de Estados

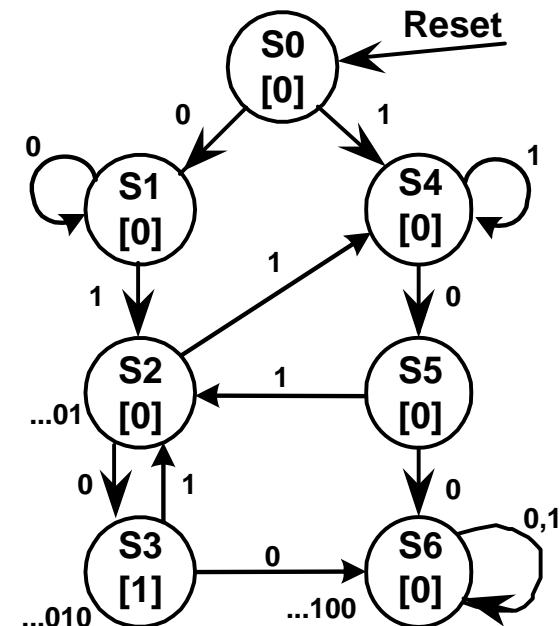
- Completar o diagrama analisando as condições de transição de cada estado



Transições em S3



Transições em S1 e S4



Transições em S2 e S5

Exemplo 2: Revisão

- Revisão do procedimento:
 - Escrever sequências de teste com as entradas/saídas para perceber a especificação
 - Criar uma sequência de estados e transições para as sequências que se pretende ver reconhecidas
 - Acrescentar transições em falta; reusar estados existentes o mais possível
 - Verificar o comportamento entrada/saída do diagrama de estados para assegurar que funciona como pretendido

Exemplo 2: Implementação

- Exercício:
 - Proponha o diagrama de estados seguindo o modelo de Mealy
 - Sugira uma codificação adequada de estados e deduza a tabela de transições/saída
 - Obtenha equações para funções de excitação/saída

Revisão da metodologia de síntese

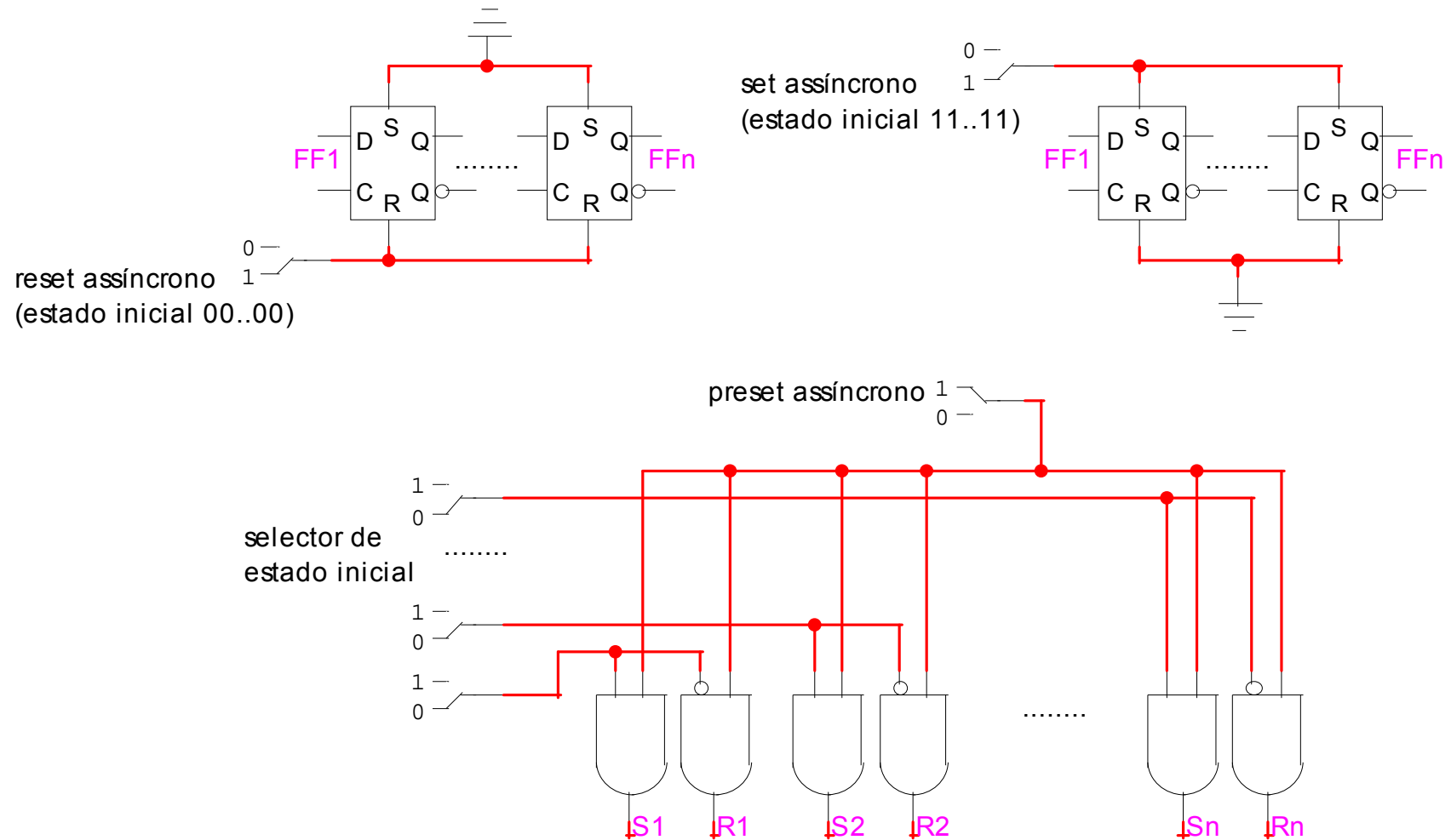
Projetar uma MEF que detete a ocorrência de 3 “1” numa sequência de comprimento 5. As sequências a detetar começam forçosamente por “11”. A MEF deverá ser tal que depois de detetados os “11” iniciais a próxima sequência só pode começar ao fim de mais de 3 entradas quer a resposta final seja “1” ou “0”

Exemplo:

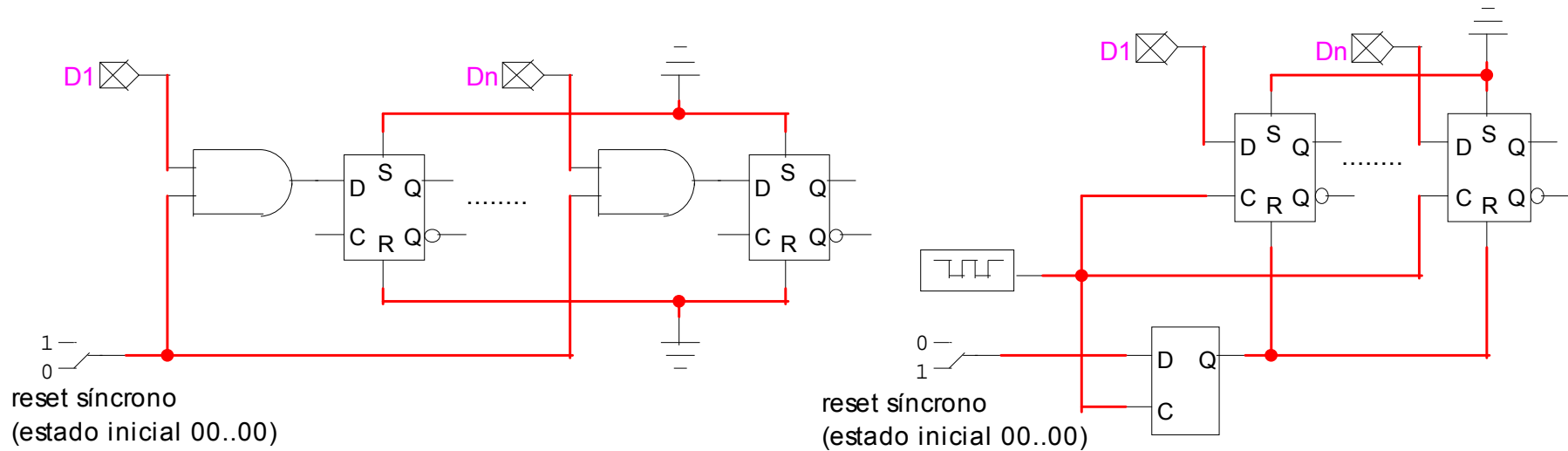
X: 100110110011010...
Y: 0000000000000001...



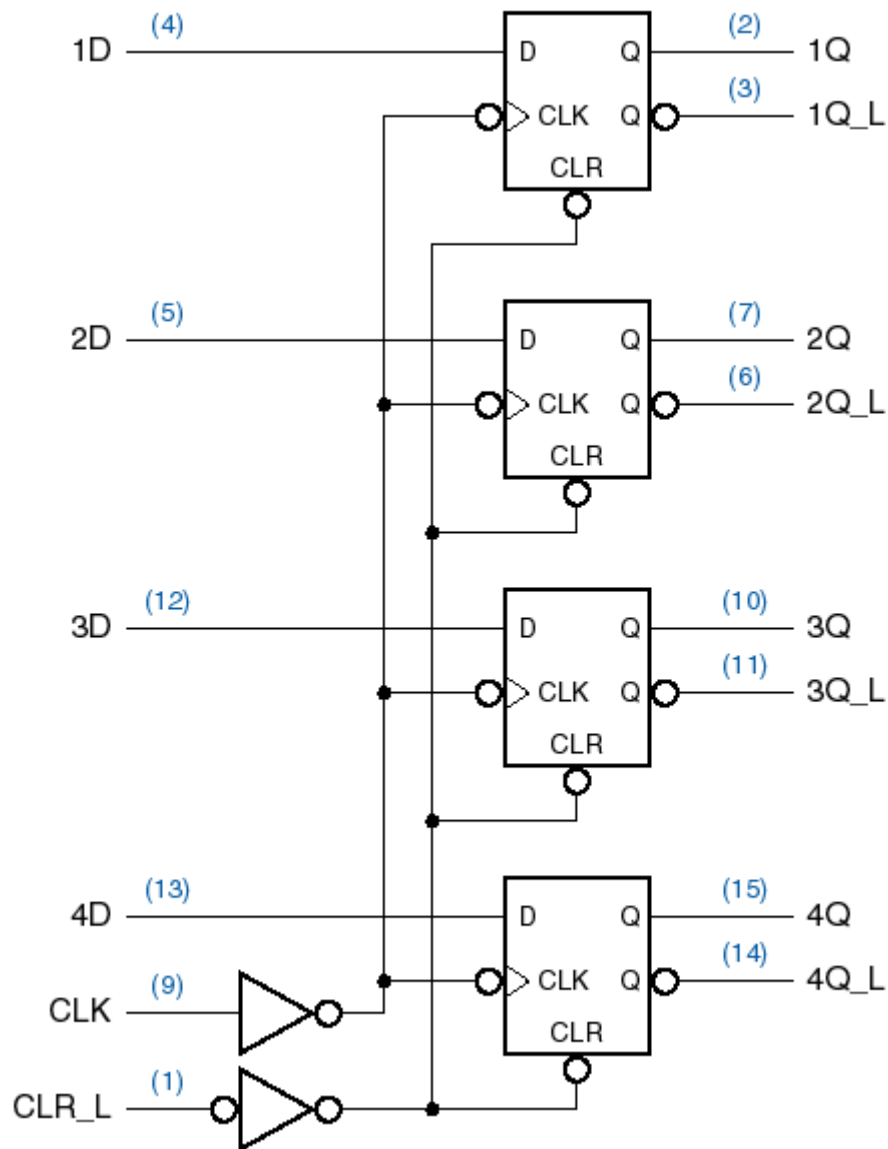
(Re)inicialização assíncrona



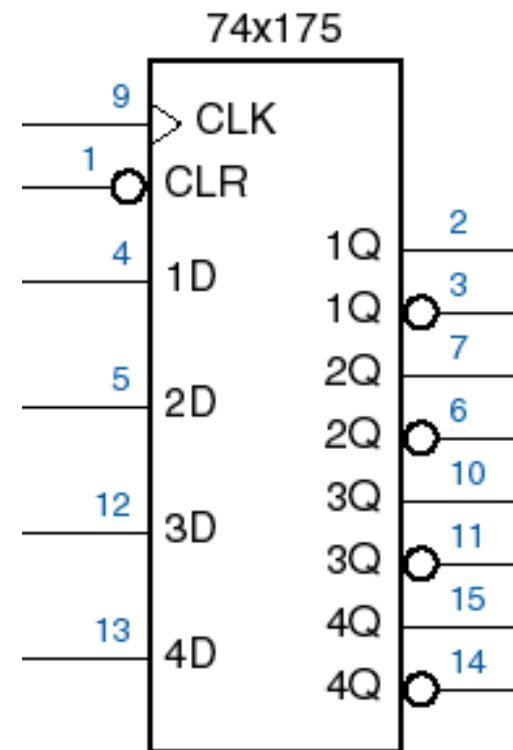
(Re)inicialização síncrona

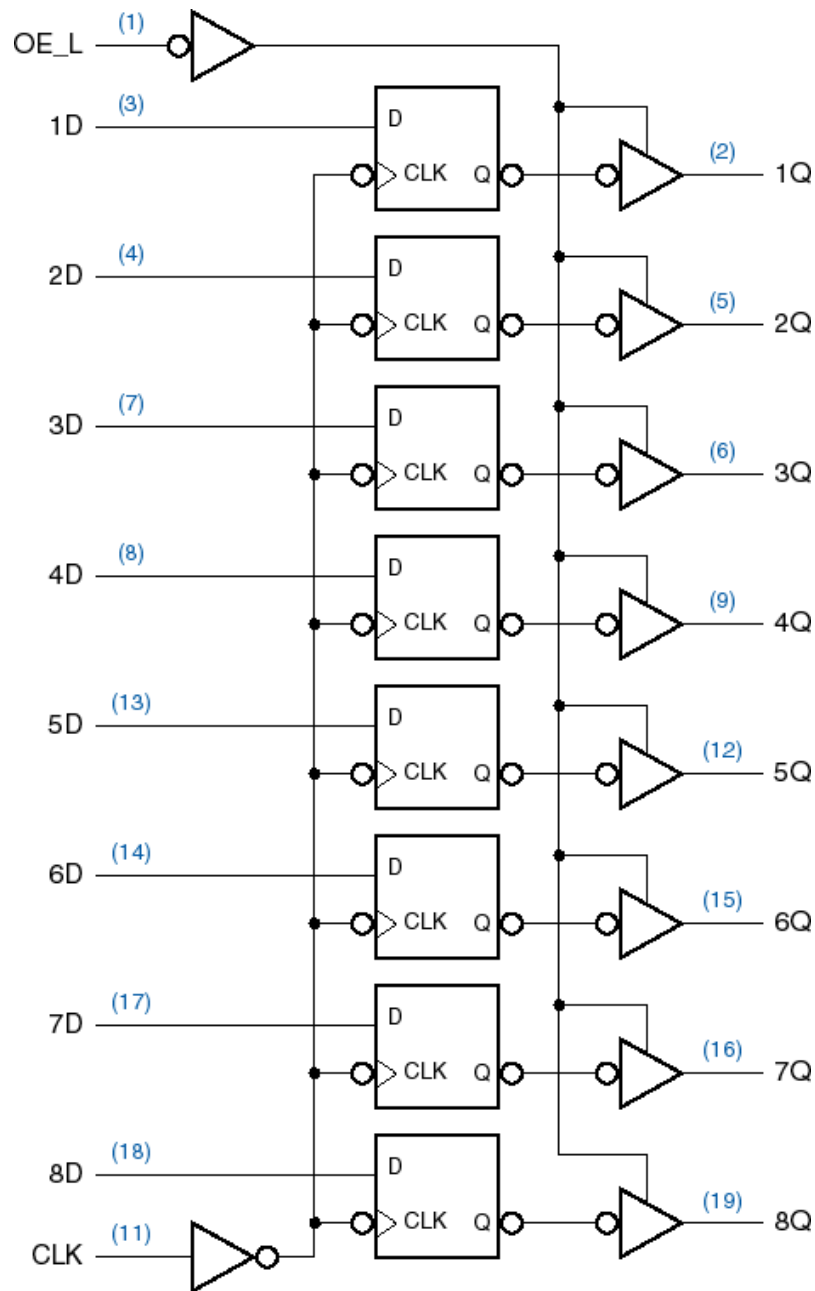


Latches e registros *multibit*



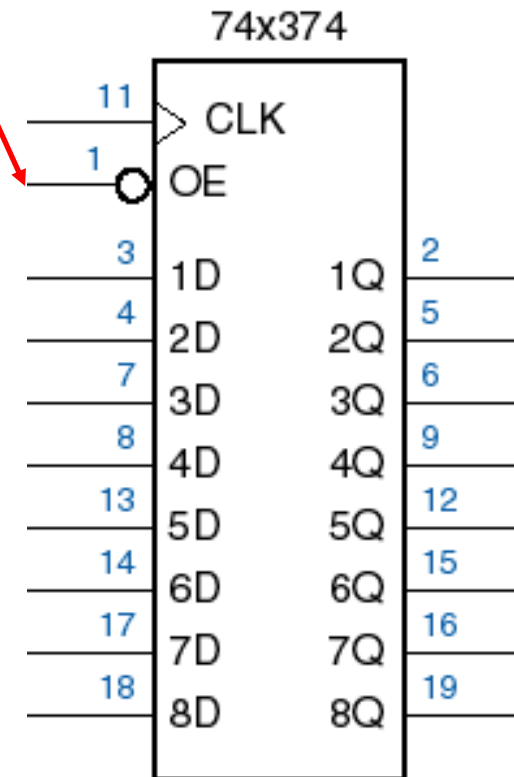
Registro de 4 bits



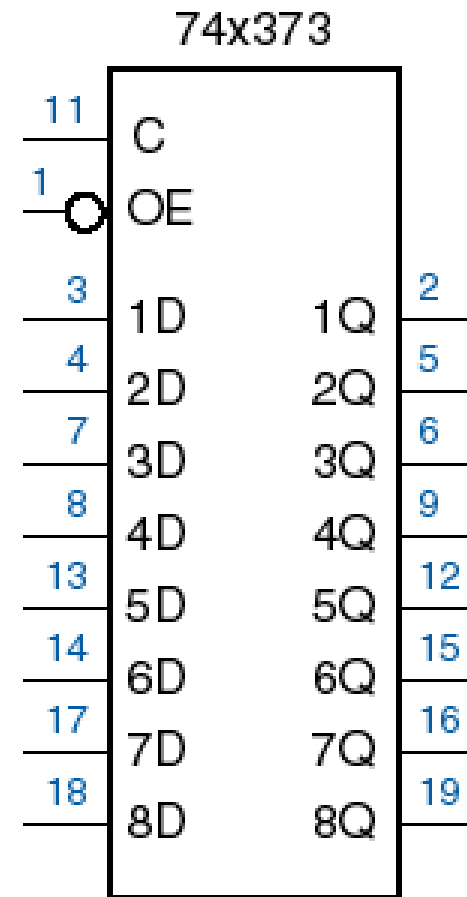
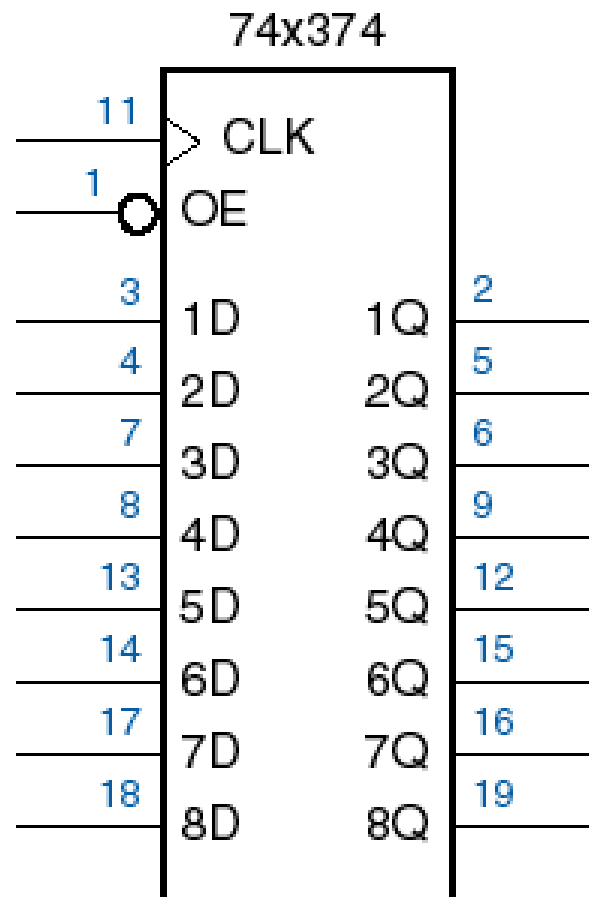


Registro de 8 bits (octal)

Saídas de 3 estados!

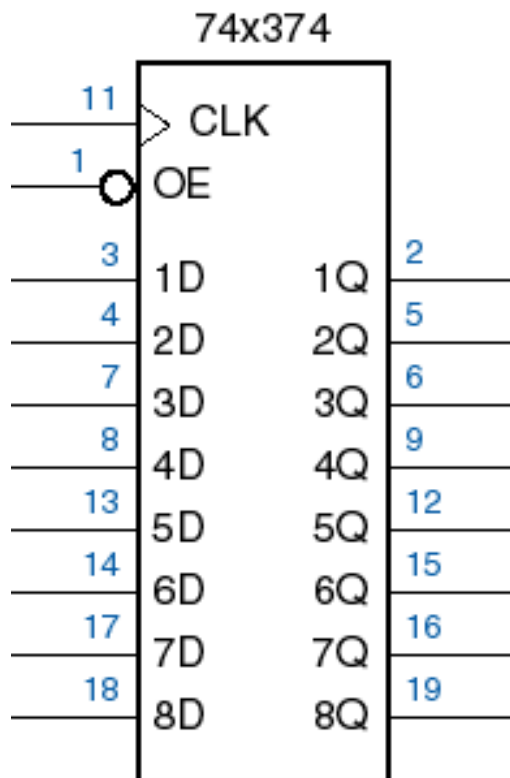


Registo vs. Latch

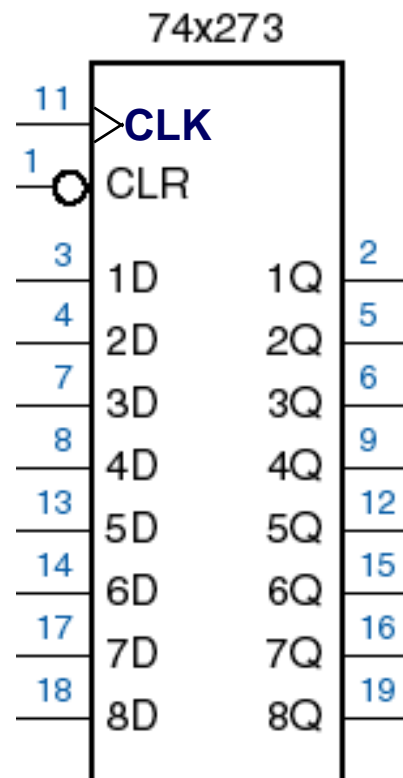


Registo de 8 bits: variantes

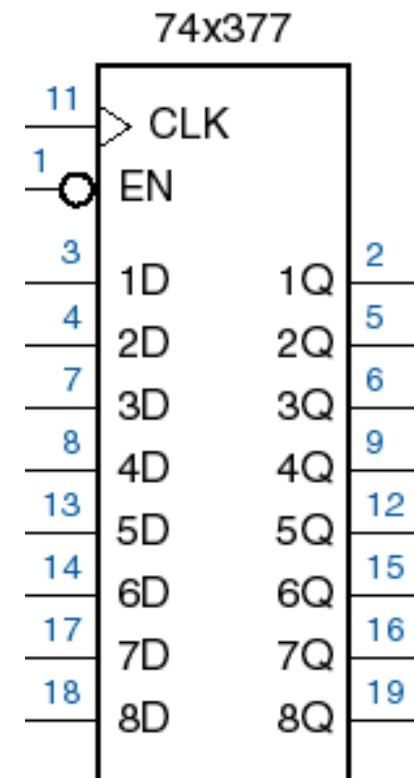
Saídas de
3 estados

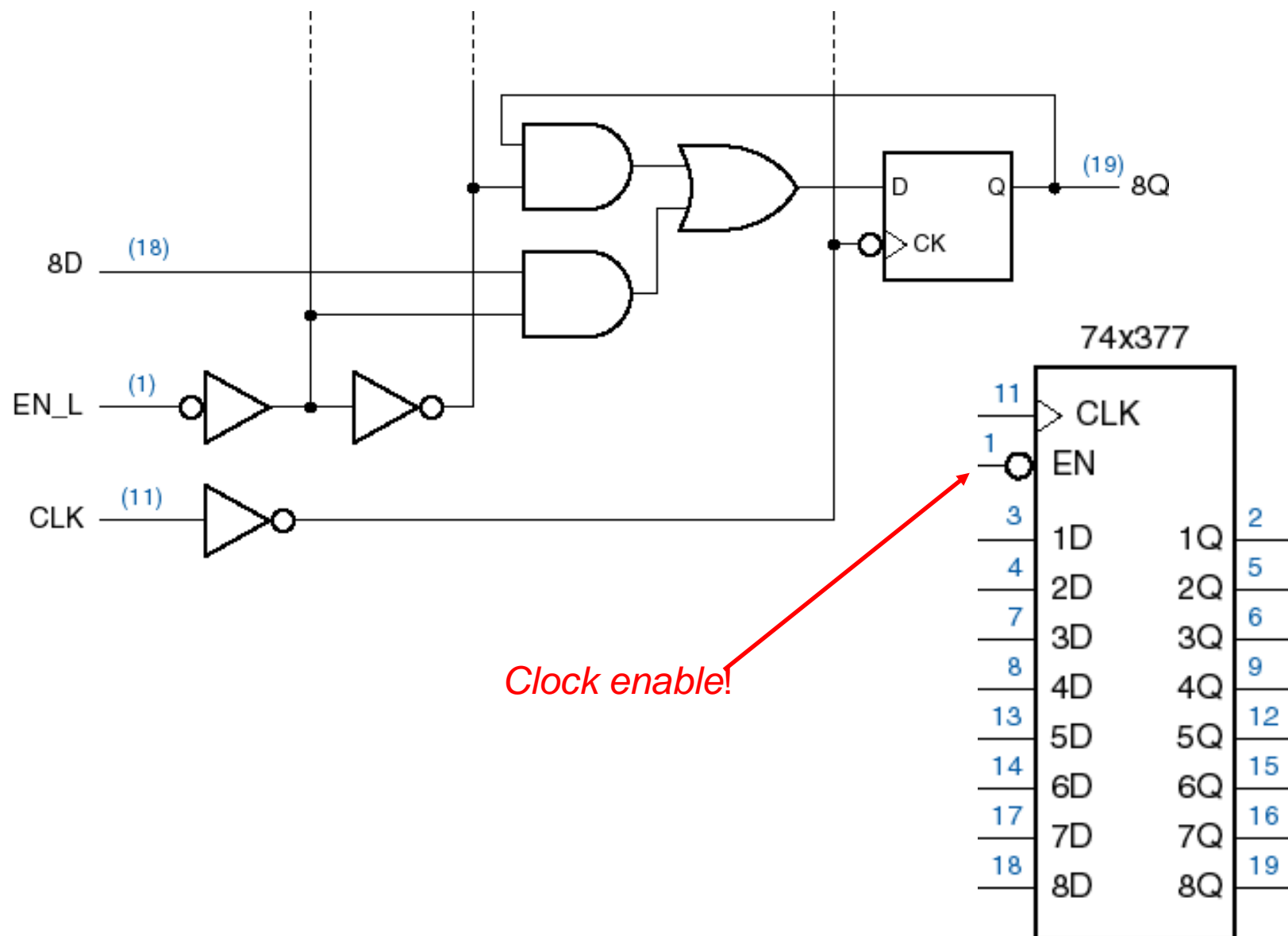


CLR
assíncrono



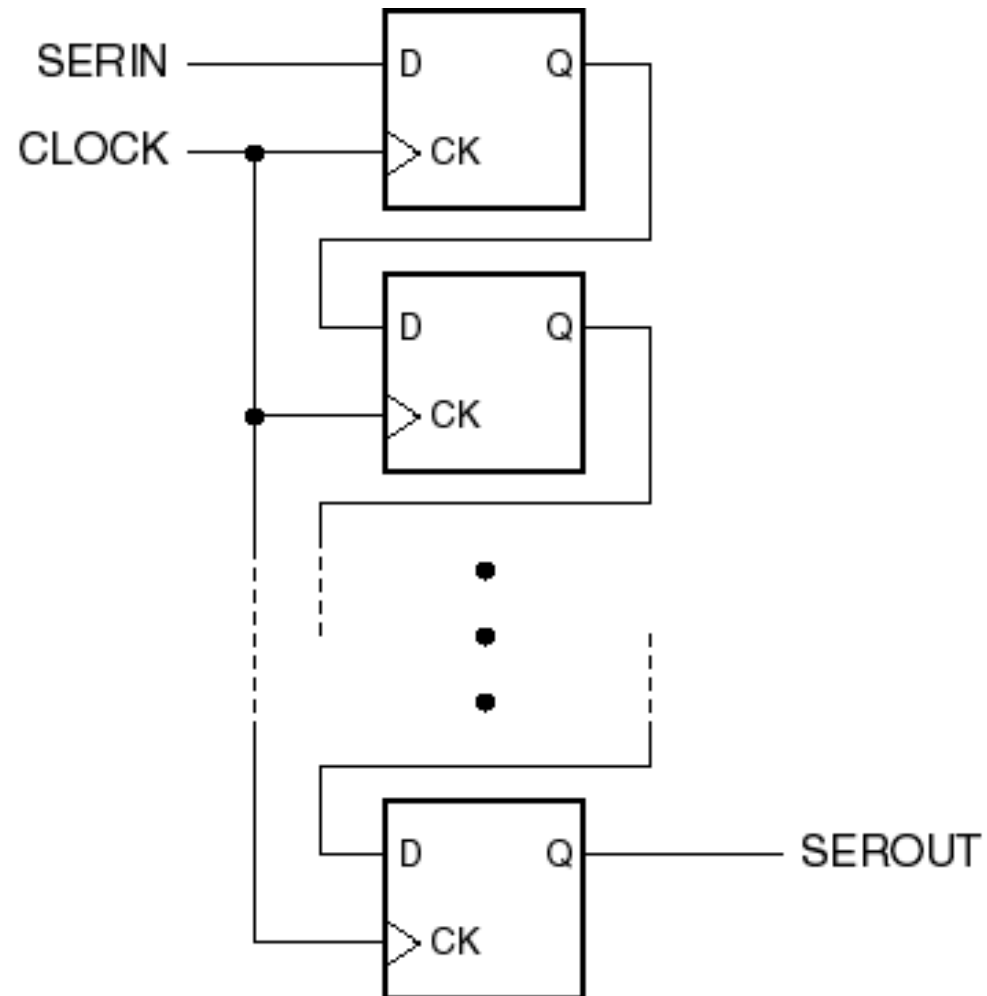
*Clock
enable*





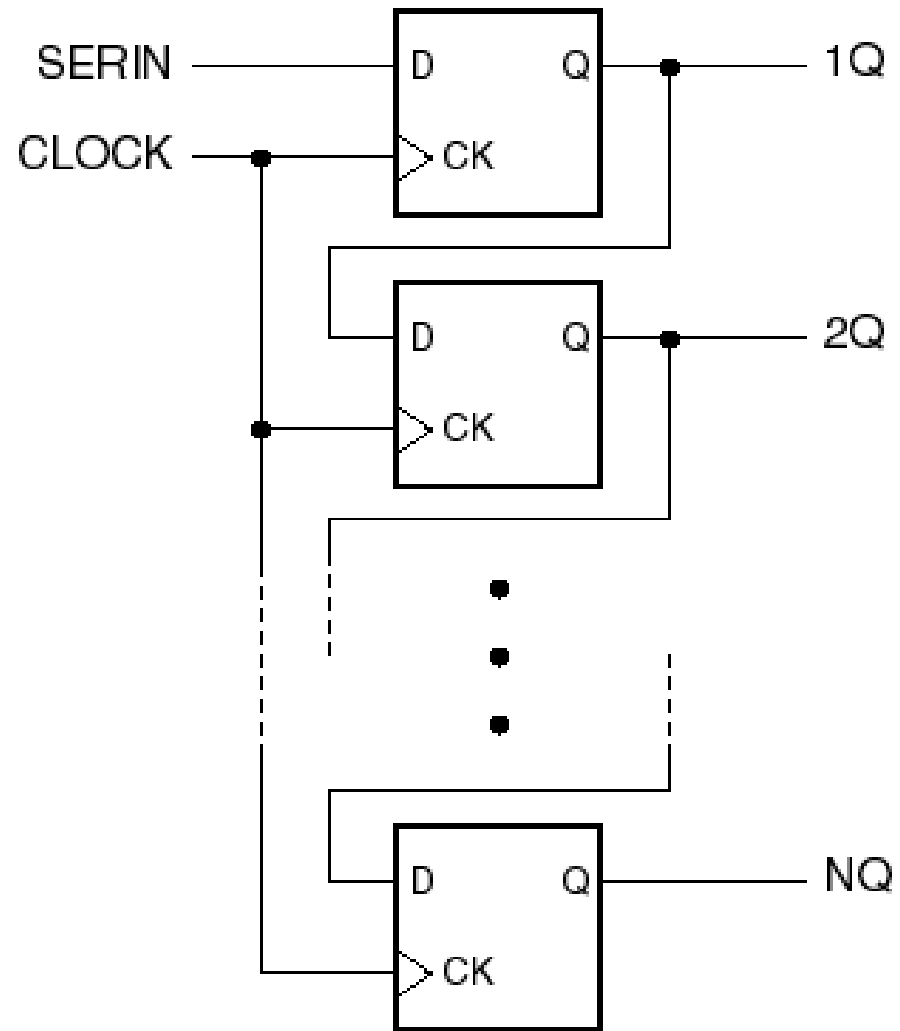
Registros de deslocamento

1) *Serial-in, serial-out*



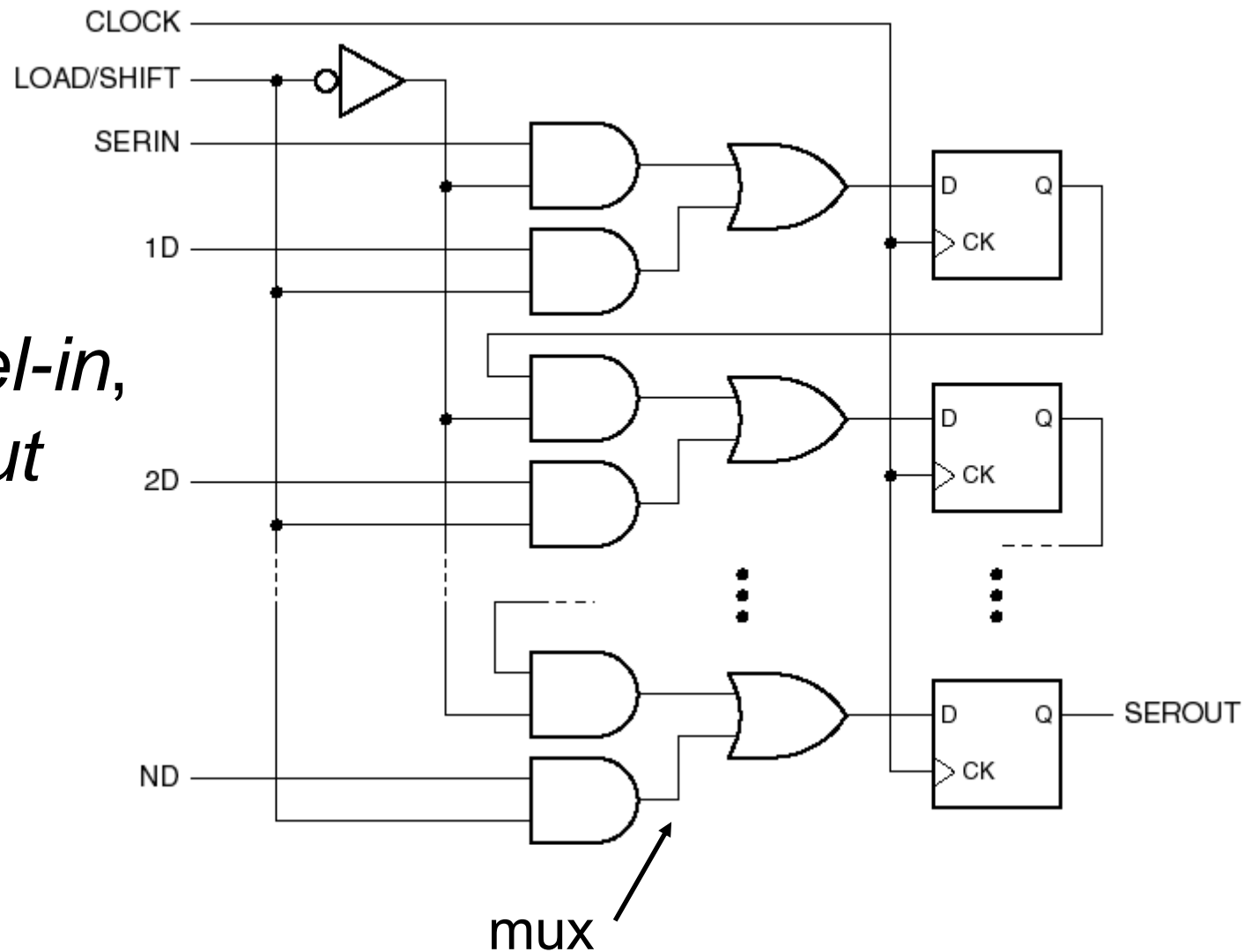
Conversão série/paralelo

2) *Serial-in, parallel-out*

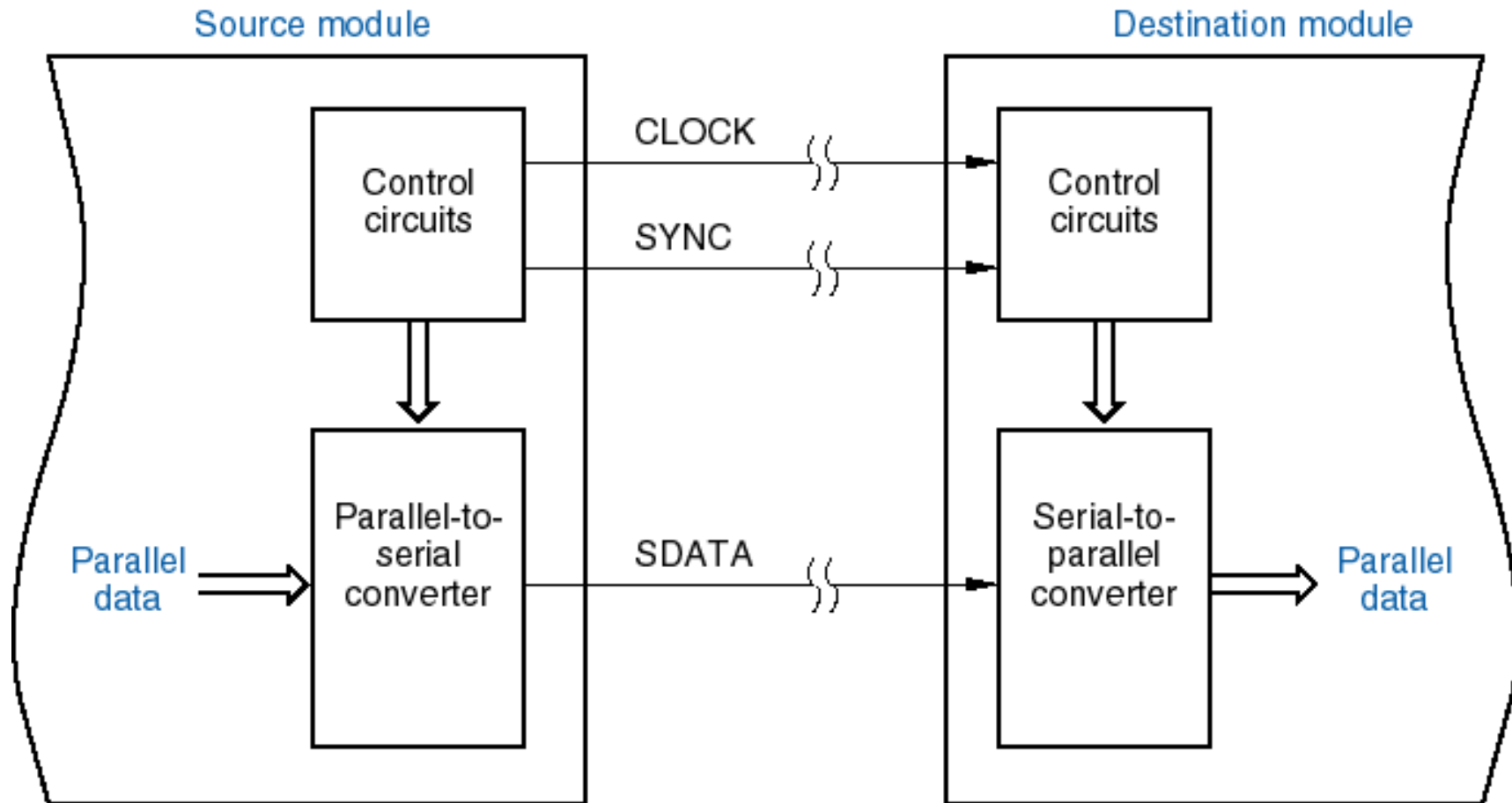


Conversão paralelo/série

3) *Parallel-in, serial-out*

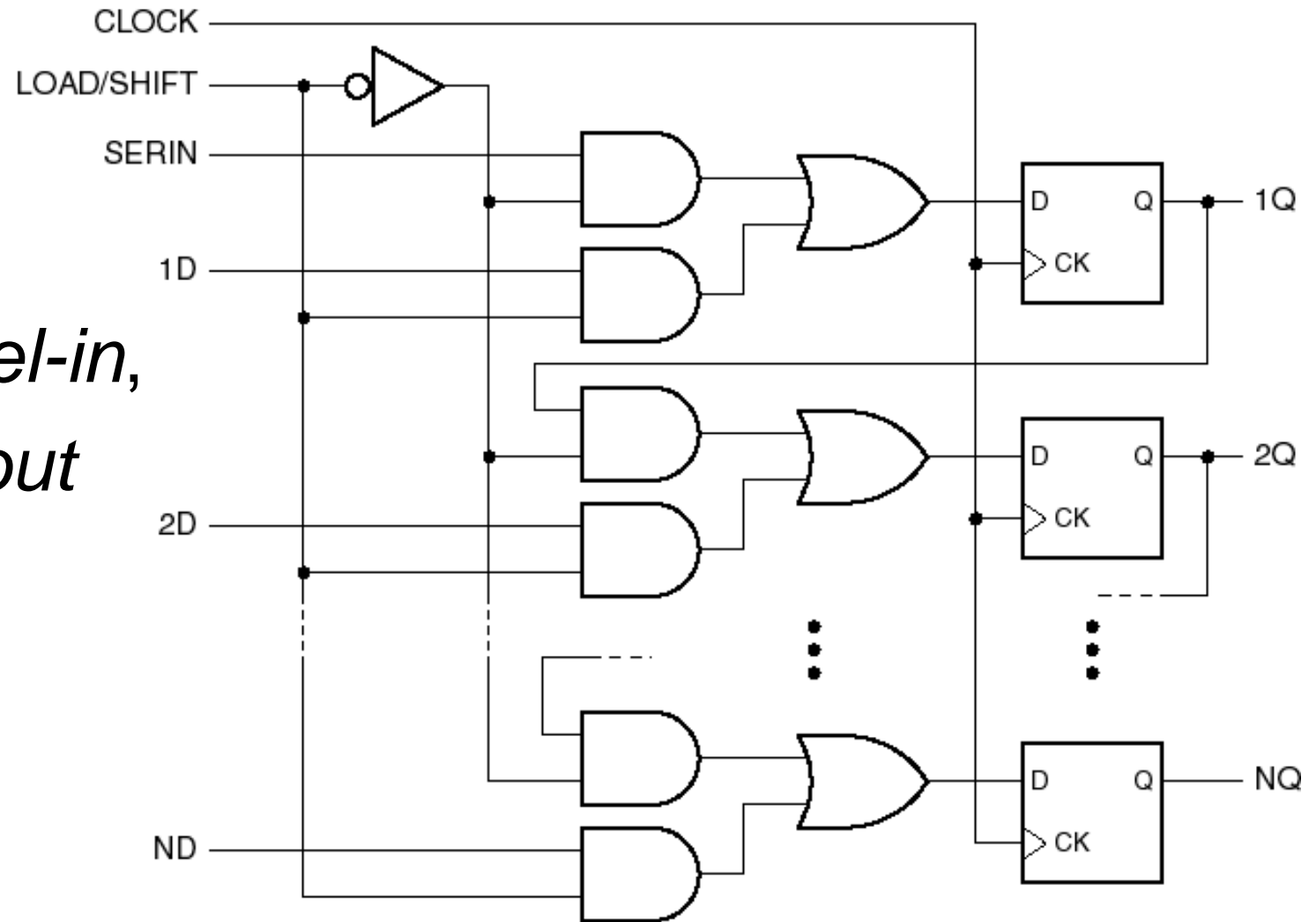


Sistemas de transmissão série



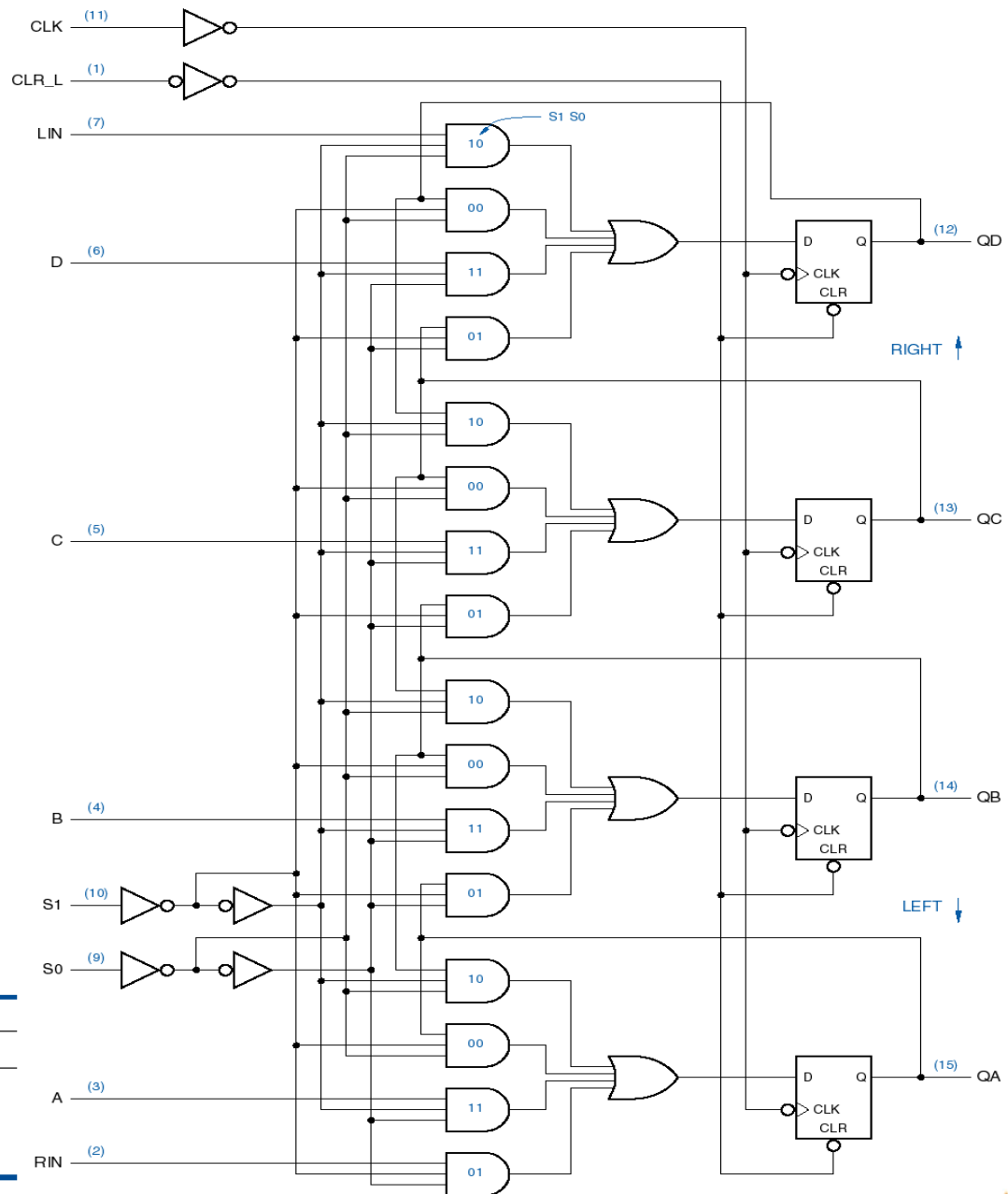
Qualquer conversão

4) *Parallel-in, parallel-out*

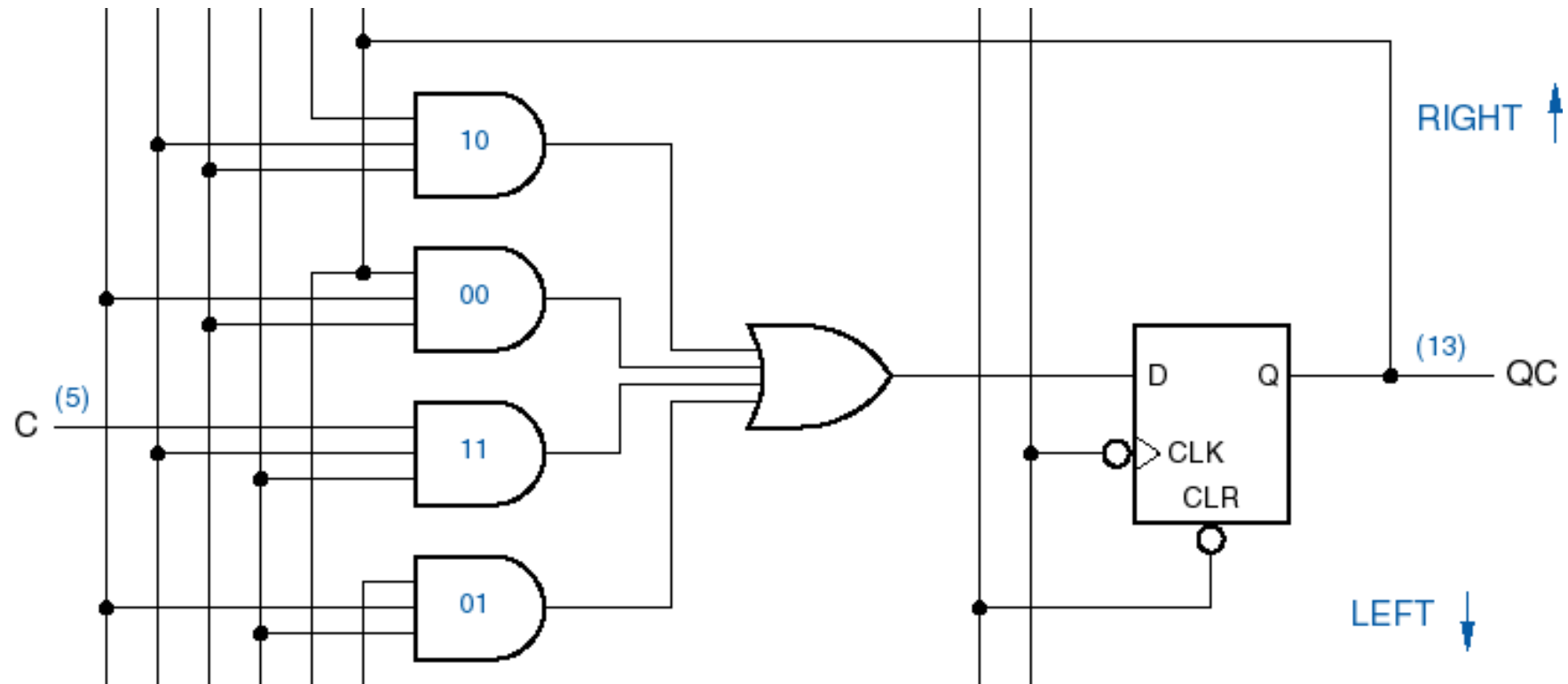


Registro de deslocamento universal 74x194

Function	Inputs		Next state			
	S1	S0	QA*	QB*	QC*	QD*
Hold	0	0	QA	QB	QC	QD
Shift right	0	1	RIN	QA	QB	QC
Shift left	1	0	QB	QC	QD	LIN
Load	1	1	A	B	C	D



Detalhe de um andar do 74x194



Contador em anel (*ring counter*)

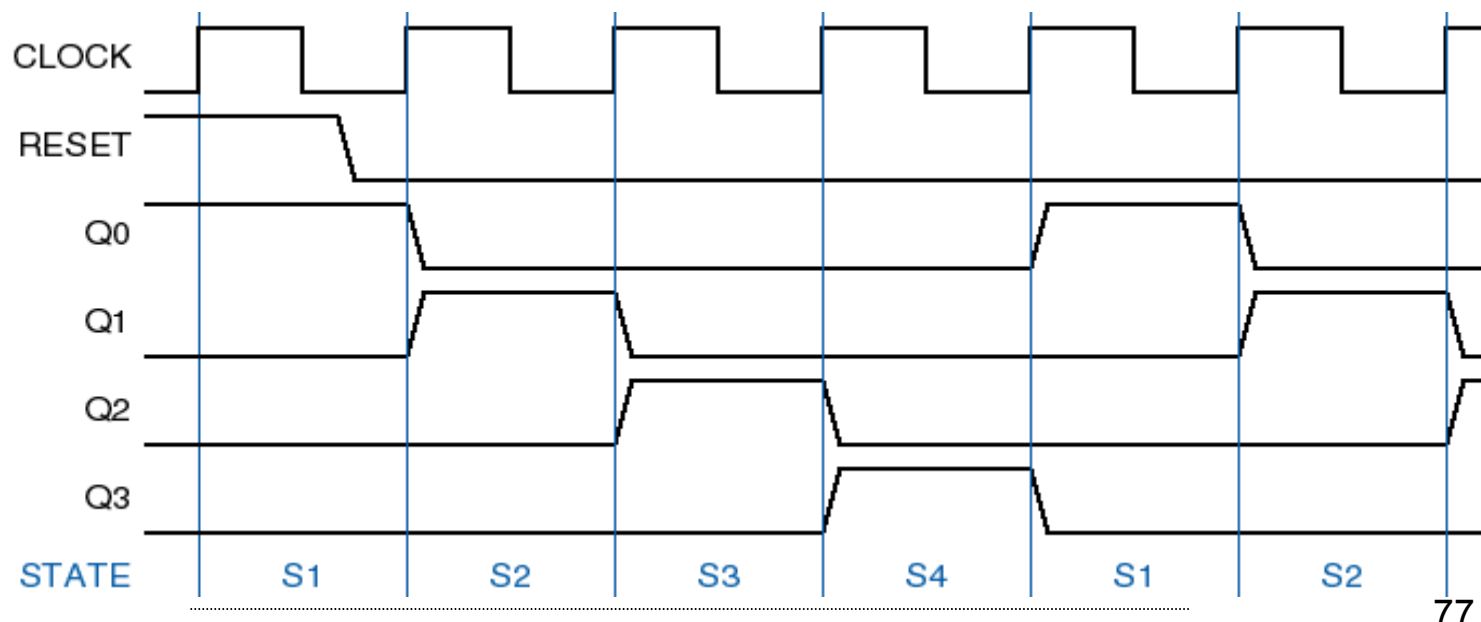
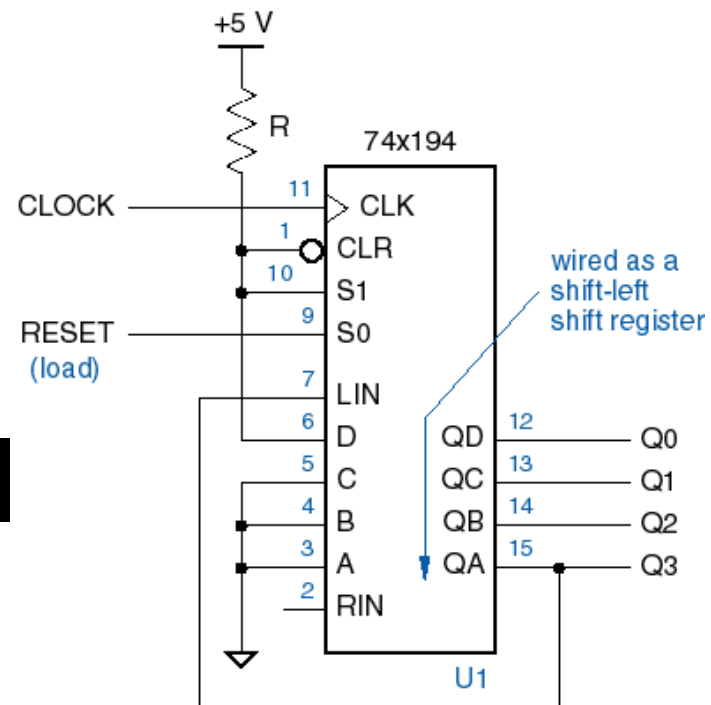
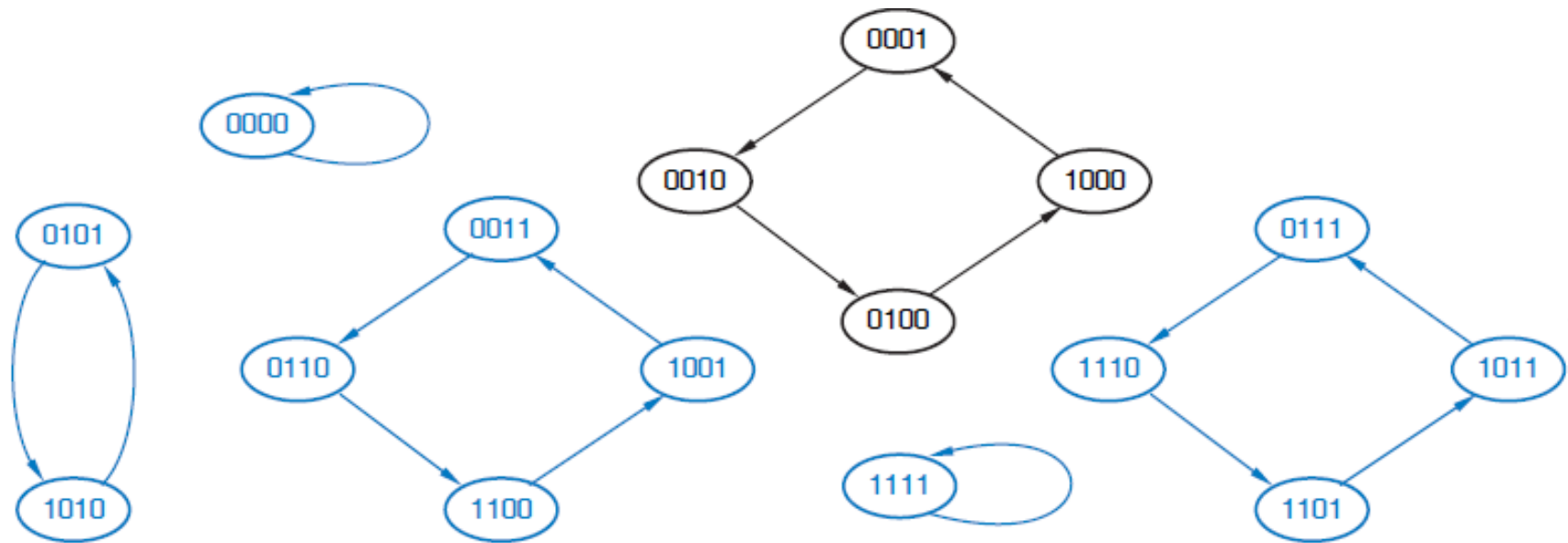
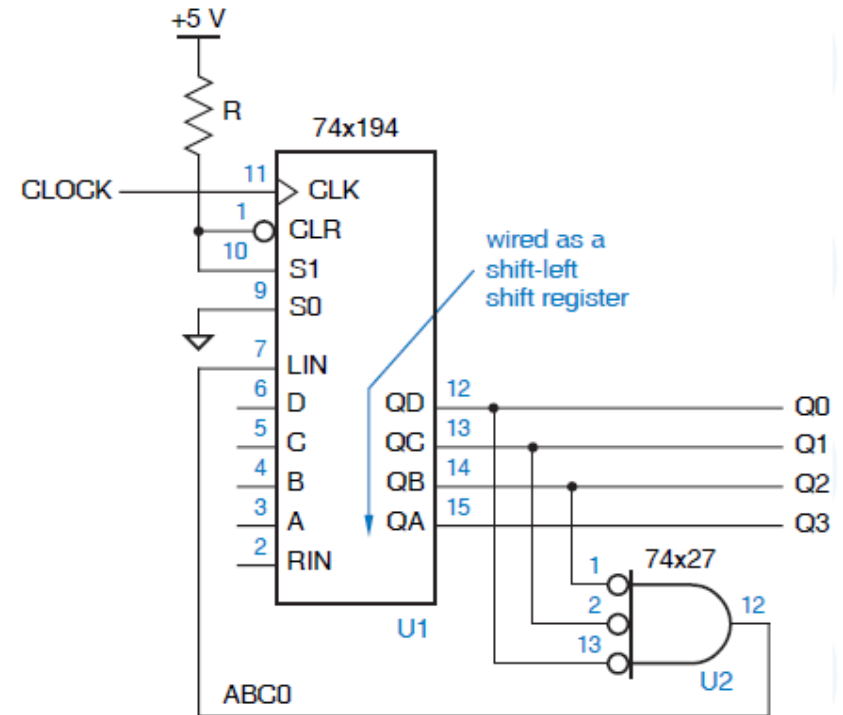
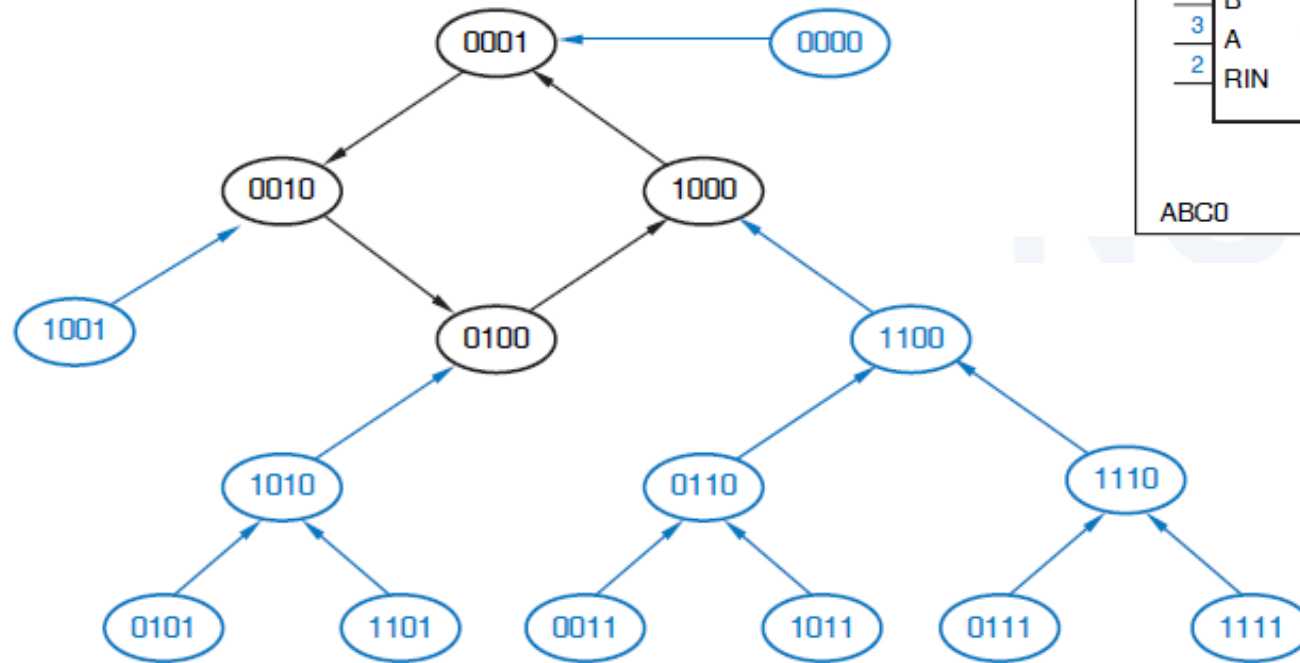


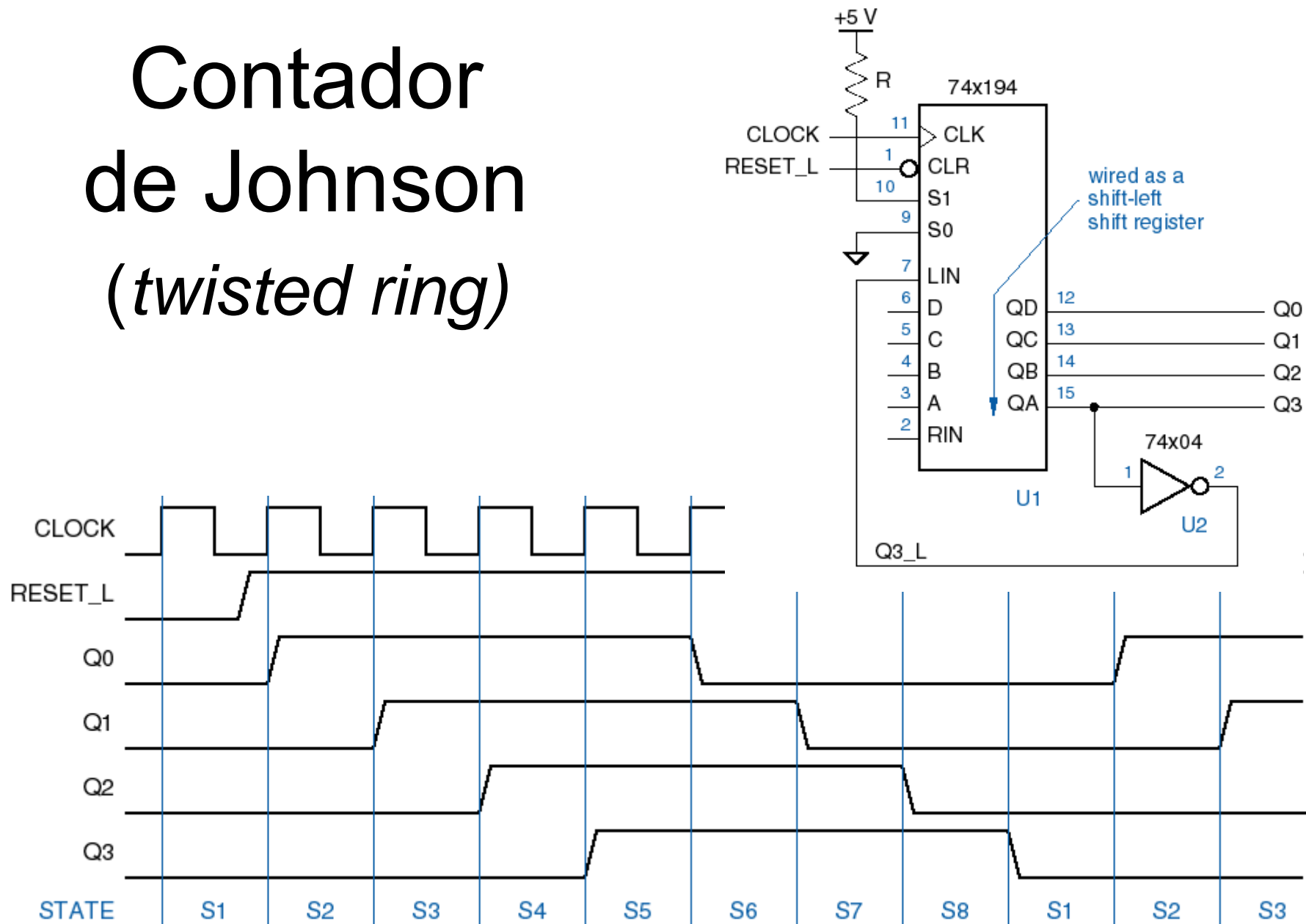
Diagrama de estados



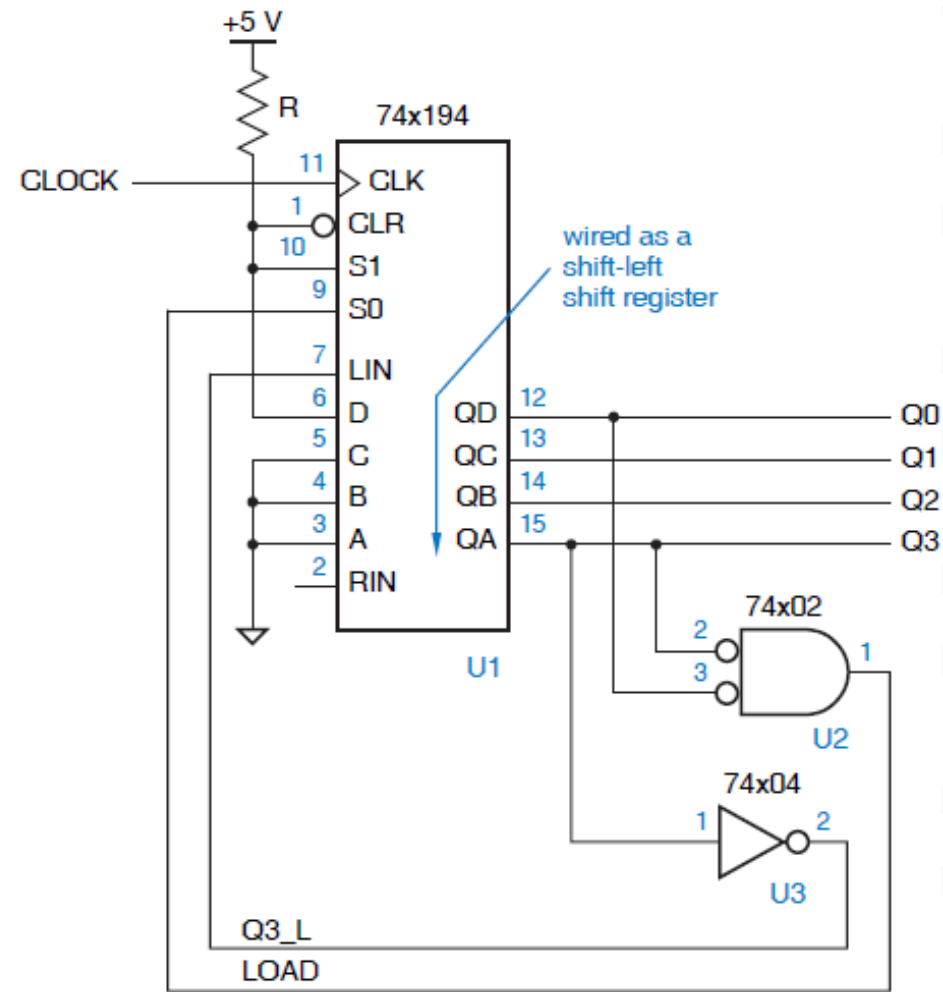
Self-correcting ring counter



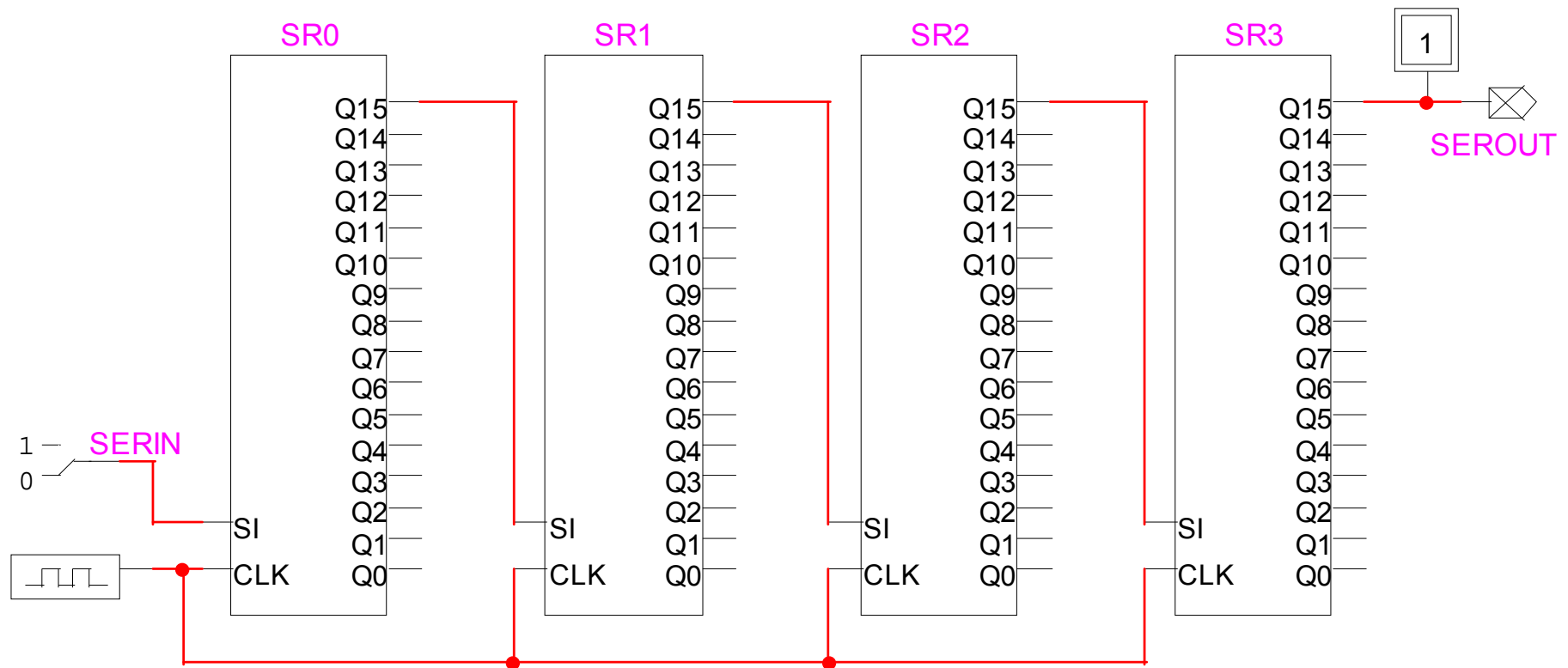
Contador de Johnson (*twisted ring*)



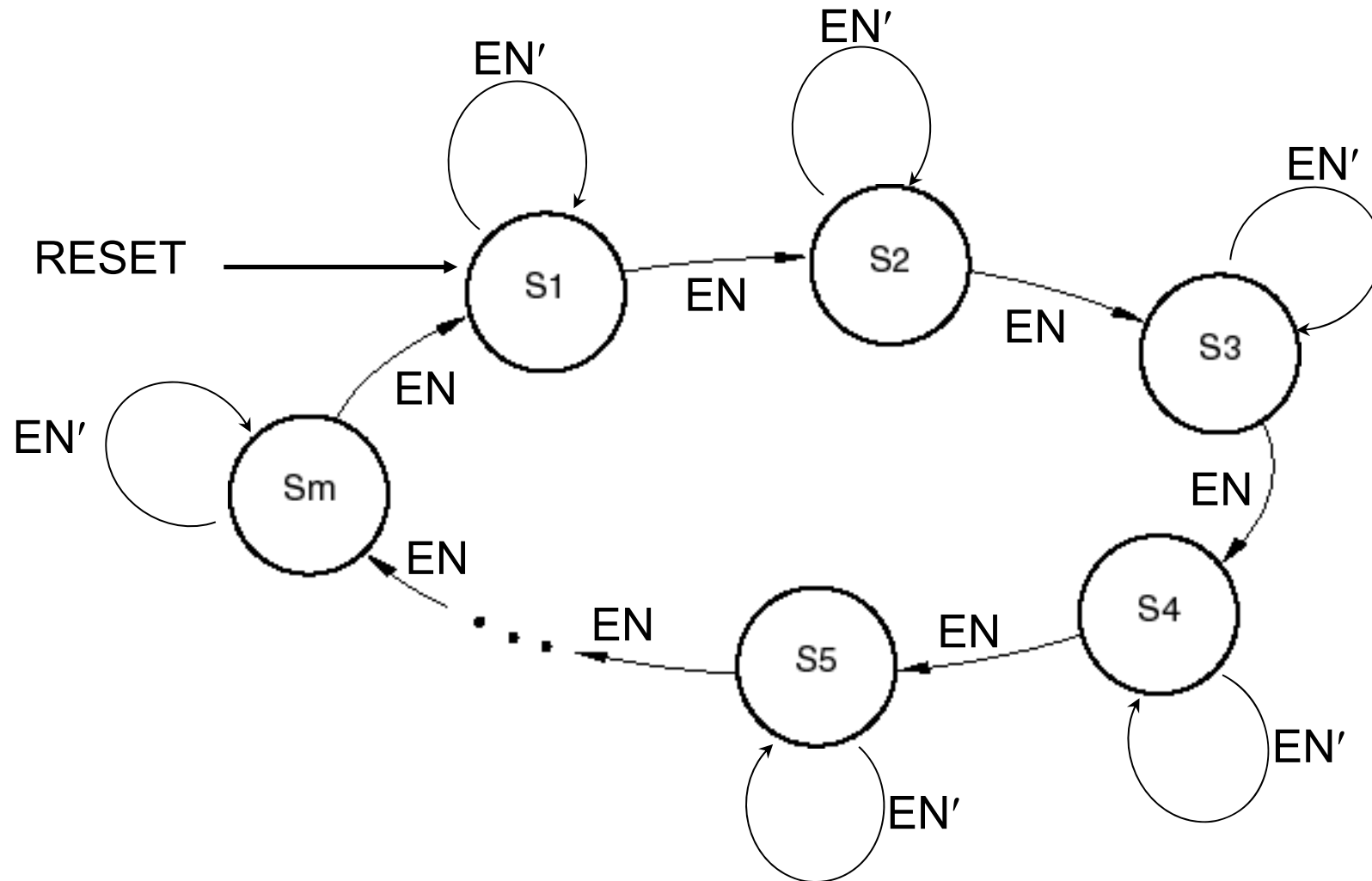
Self-correcting Johnson counter



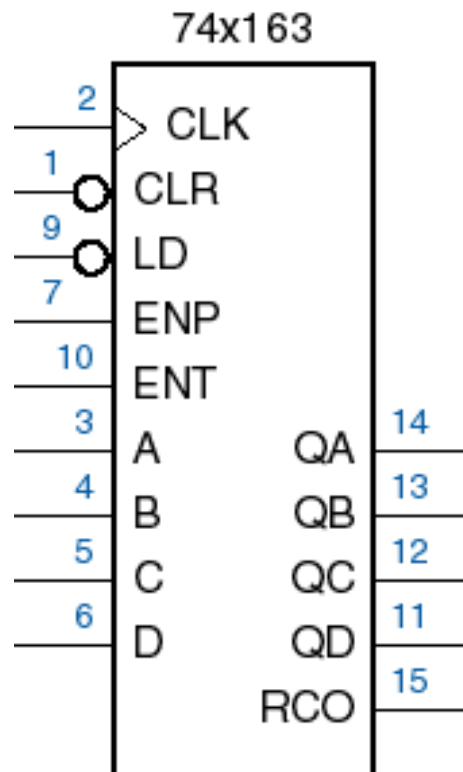
Shift-registers em cascata



Contadores

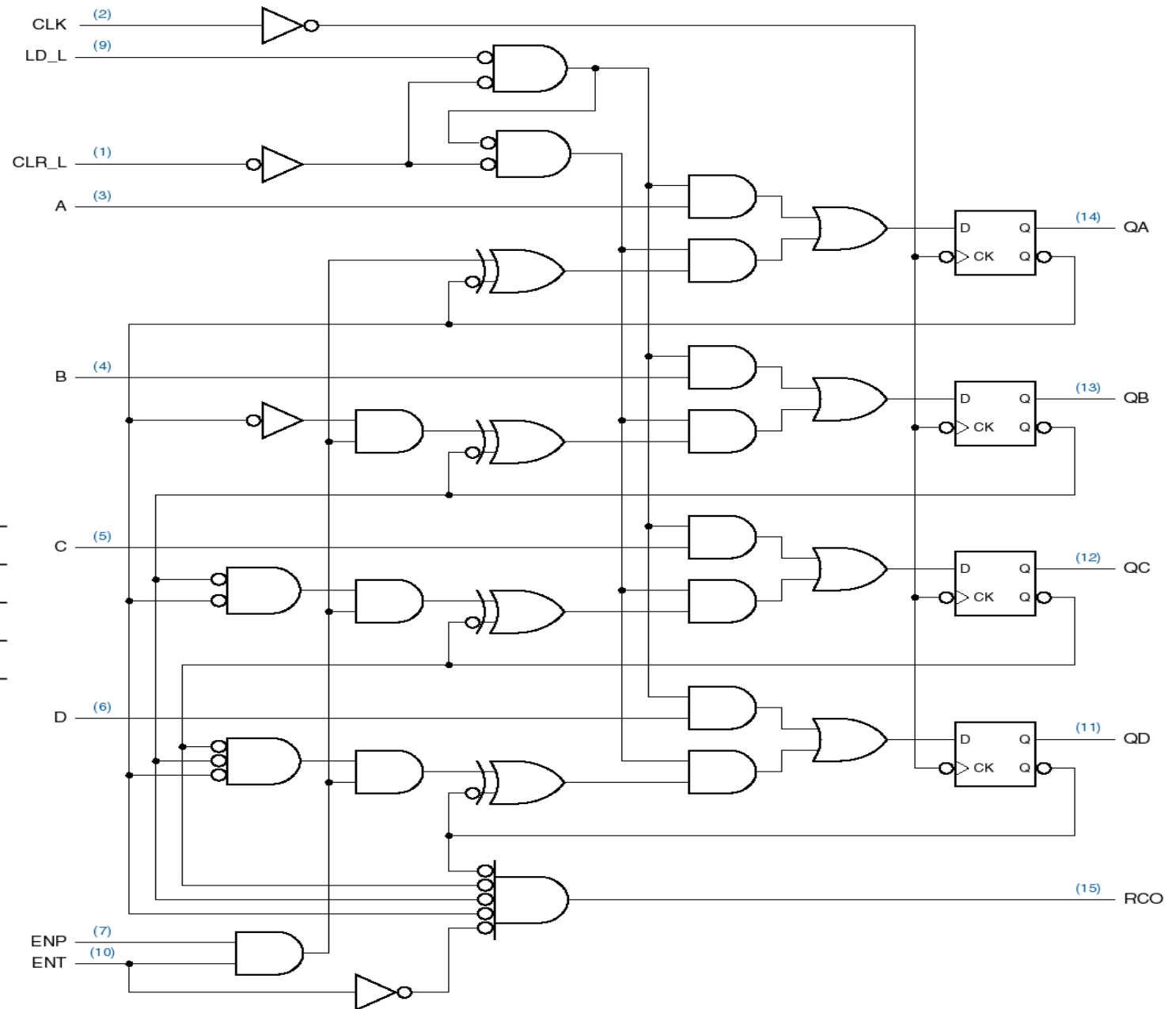
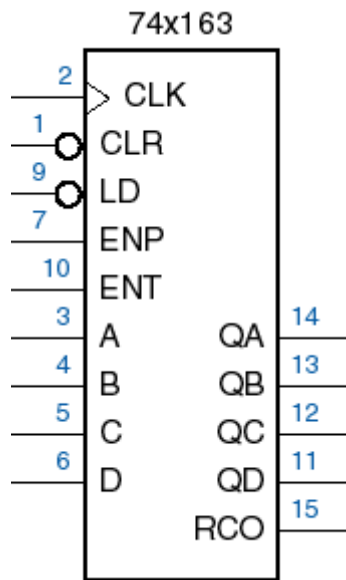


Contadores MSI

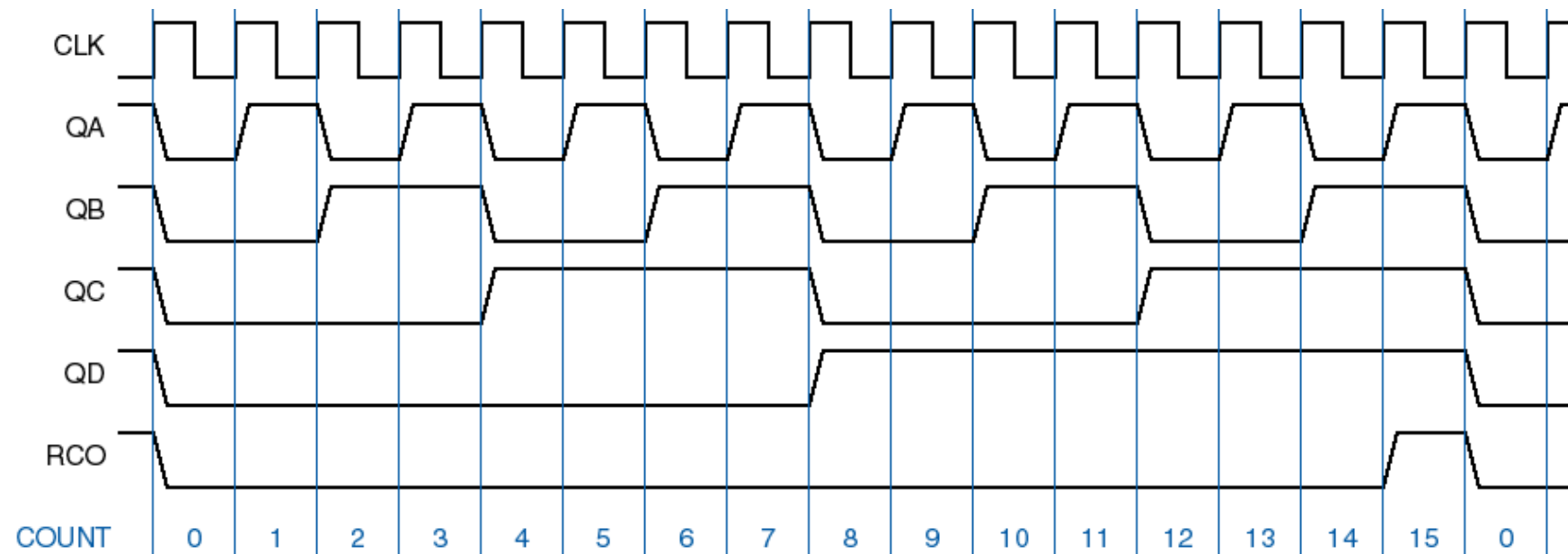
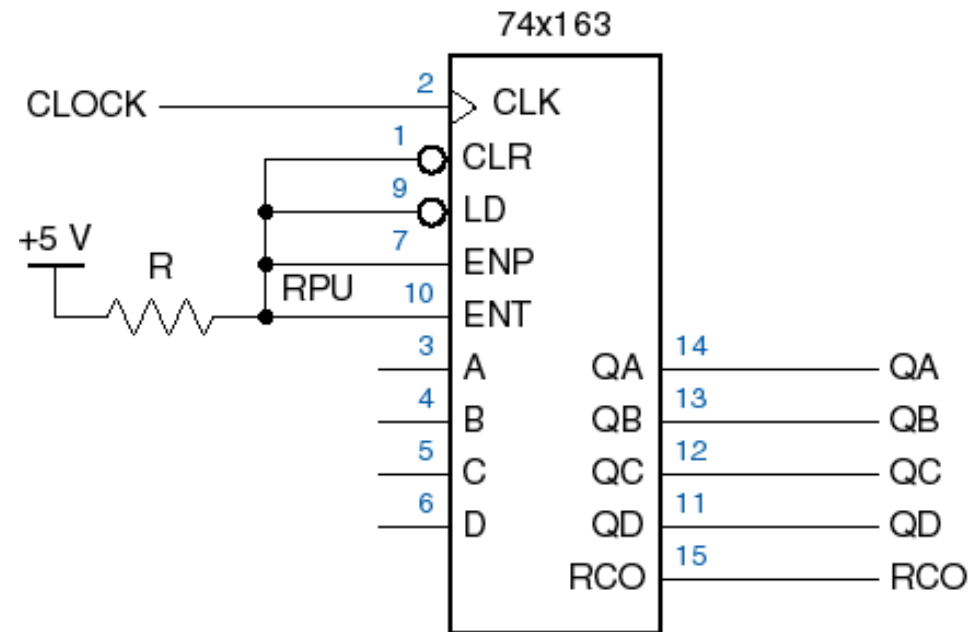


Inputs				Current State				Next State			
CLR_L	LD_L	ENT	ENP	QD	QC	QB	QA	QD*	QC*	QB*	QA*
0	x	x	x	x	x	x	x	0	0	0	0
1	0	x	x	x	x	x	x	D	C	B	A
1	1	0	x	x	x	x	x	QD	QC	QB	QA
1	1	x	0	x	x	x	x	QD	QC	QB	QA
1	1	1	1	0	0	0	0	0	0	0	1
1	1	1	1	0	0	0	1	0	0	1	0
1	1	1	1	0	0	1	0	0	0	1	1
1	1	1	1	0	0	1	1	0	1	0	0
1	1	1	1	0	1	0	0	0	1	0	1
1	1	1	1	0	1	0	1	0	1	1	0
1	1	1	1	0	1	1	0	0	1	1	1
1	1	1	1	0	1	1	1	1	0	0	0
1	1	1	1	1	0	0	0	1	0	0	1
1	1	1	1	1	0	0	1	1	0	1	0
1	1	1	1	1	0	1	0	1	0	1	1
1	1	1	1	1	0	1	1	1	1	0	0
1	1	1	1	1	1	0	0	1	1	0	1
1	1	1	1	1	1	1	0	1	1	1	0
1	1	1	1	1	1	1	1	0	1	1	1
1	1	1	1	1	1	1	1	0	0	0	0

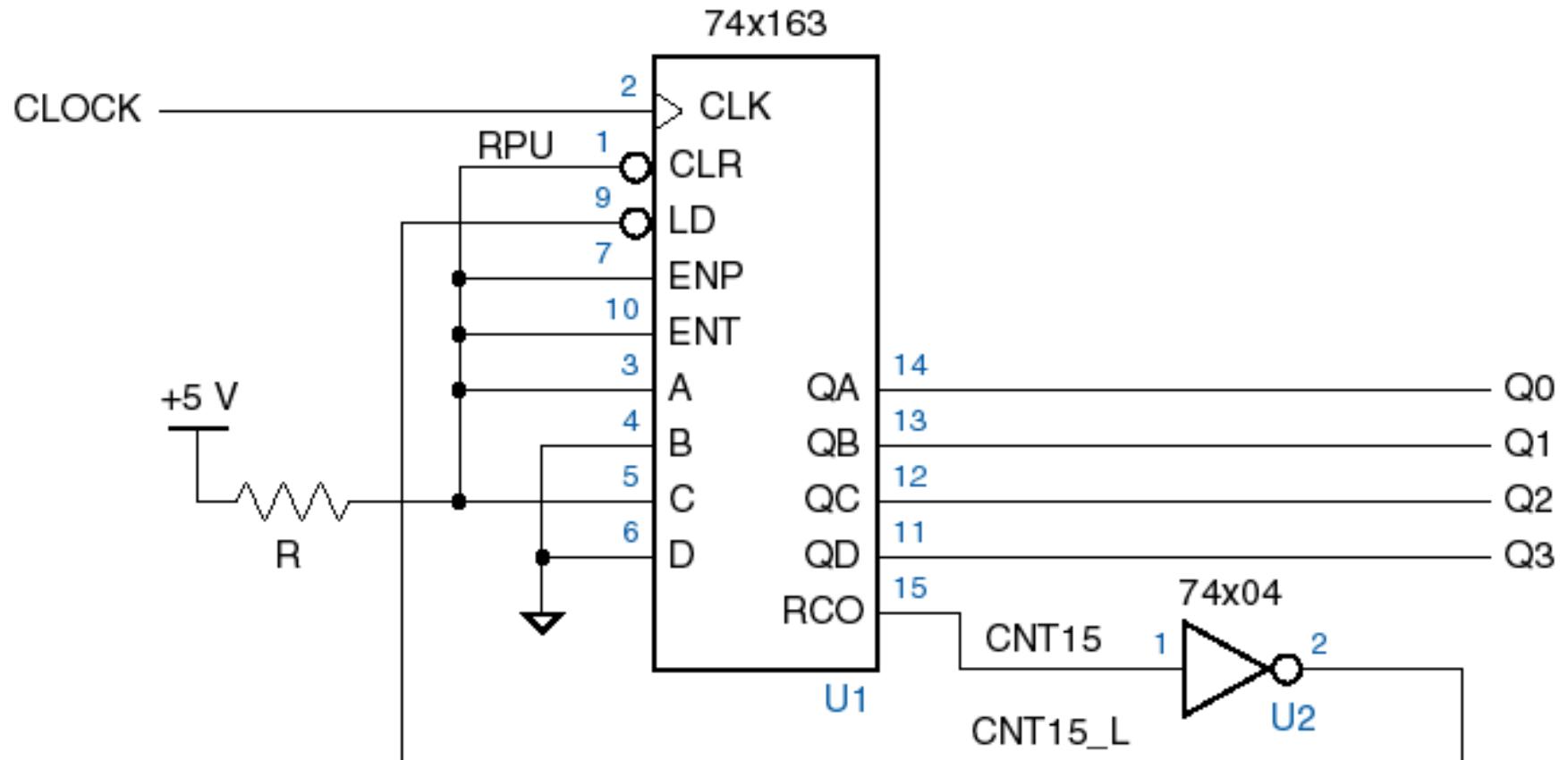
Diagrama interno



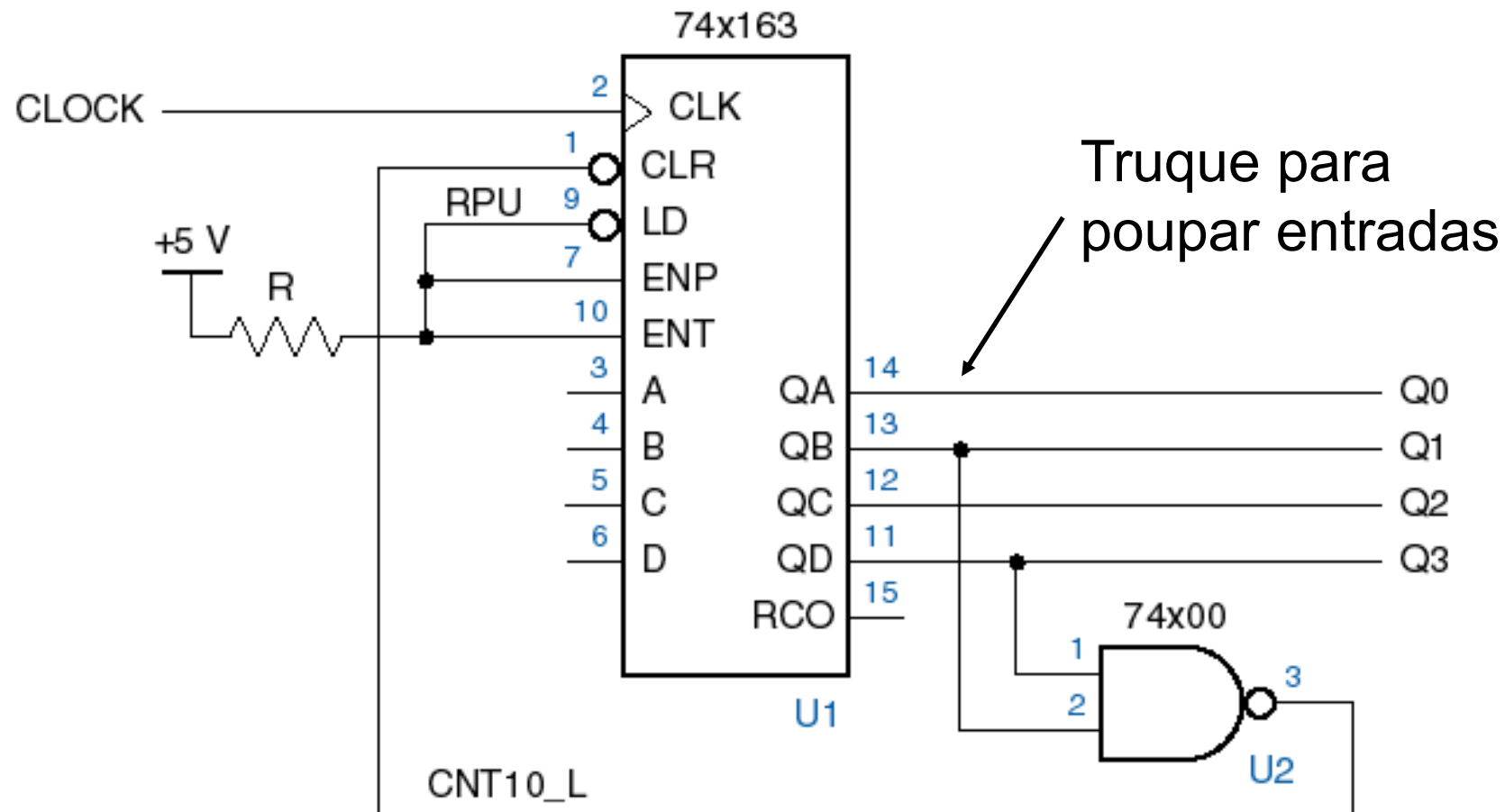
Funcionamento em modo livre (*free-running*)



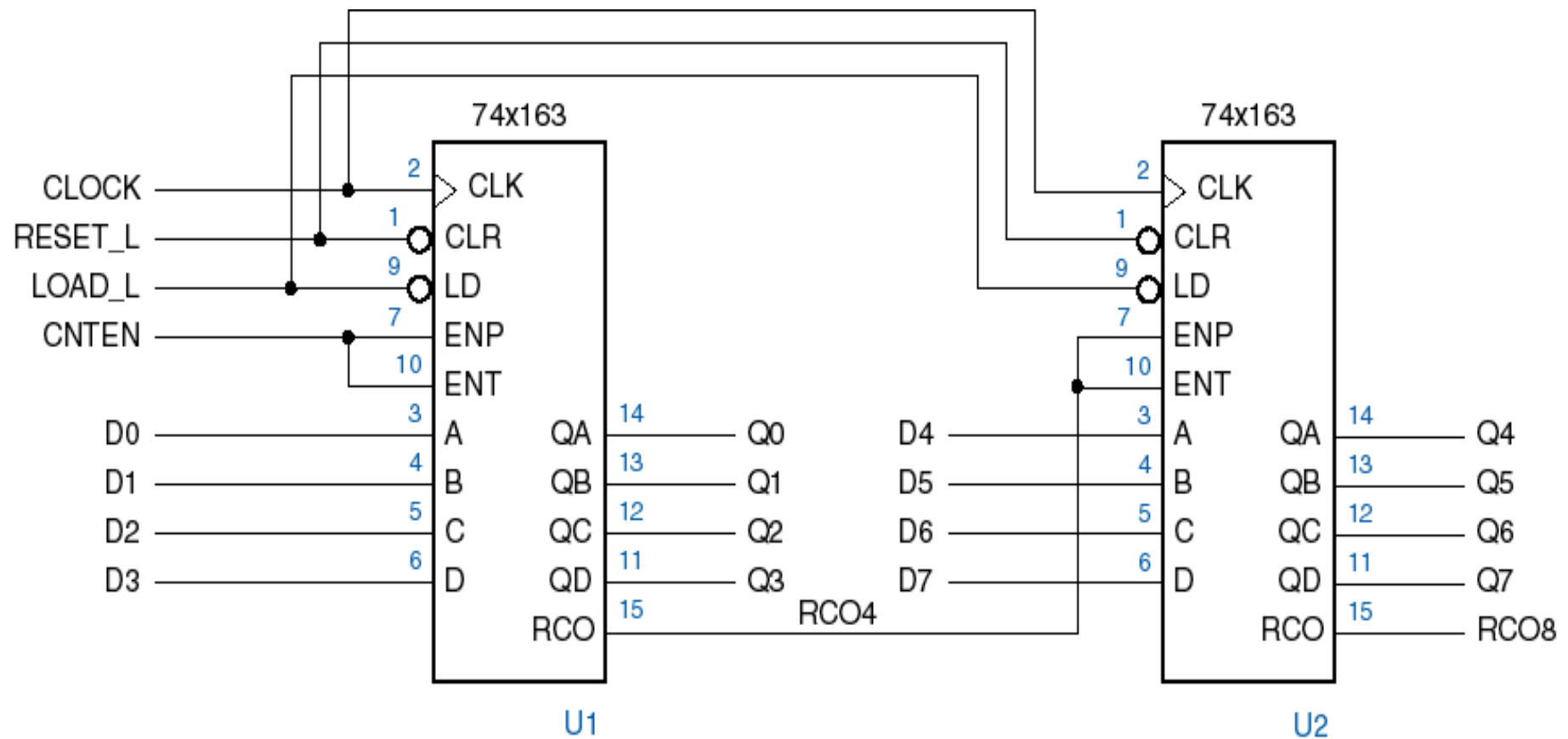
Controlo da sequência de contagem: exemplo



Controlo da sequência de contagem: outro exemplo

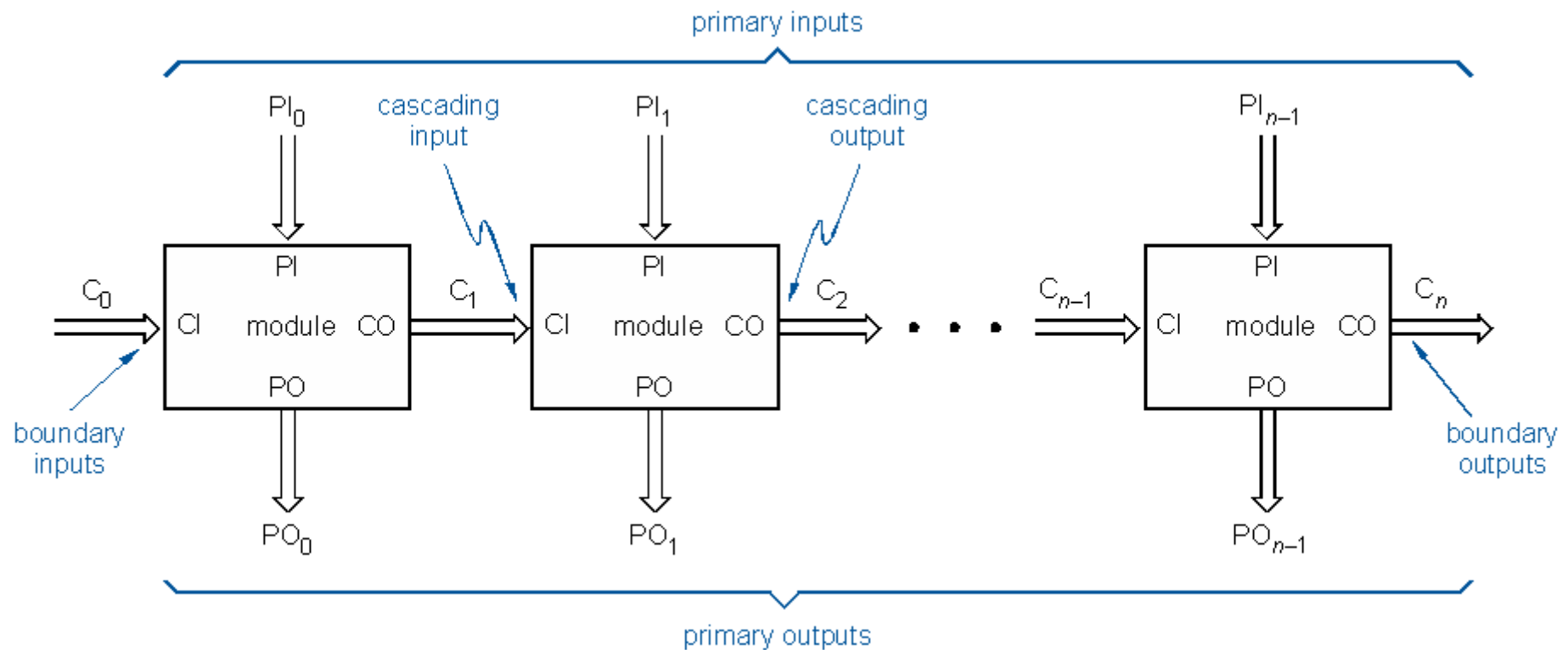


Contadores em cascata



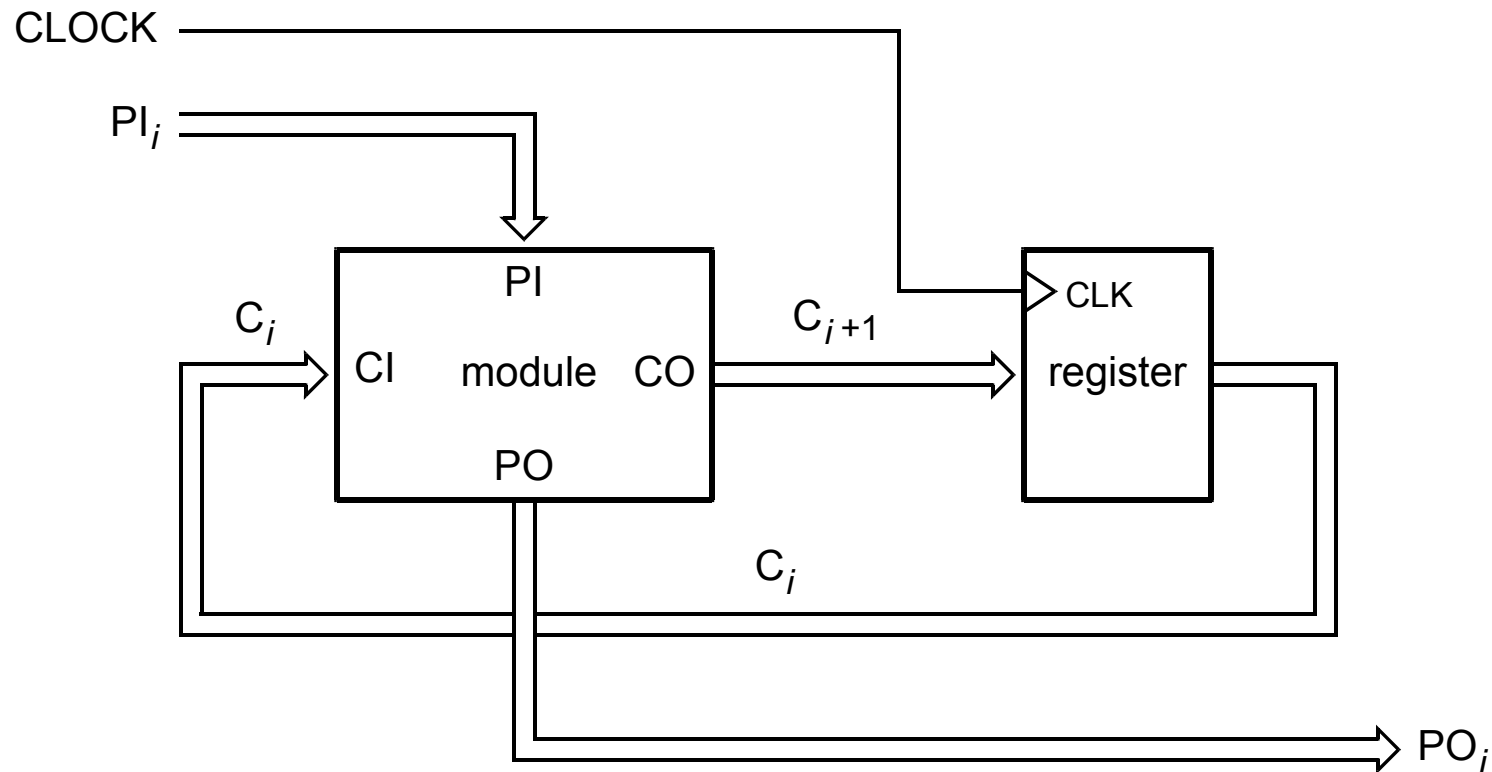
Circuitos sequenciais vs. iterativos

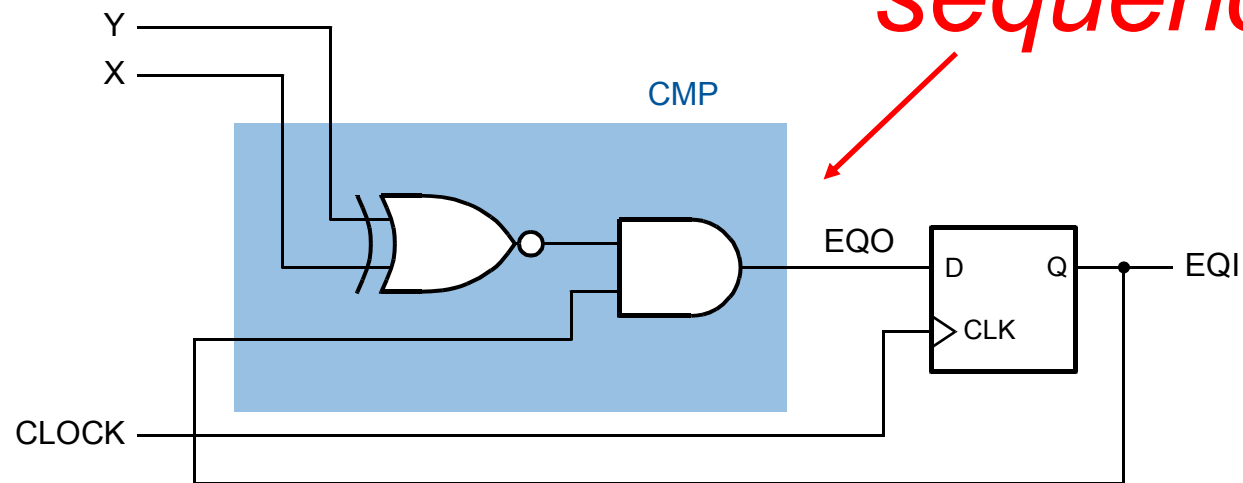
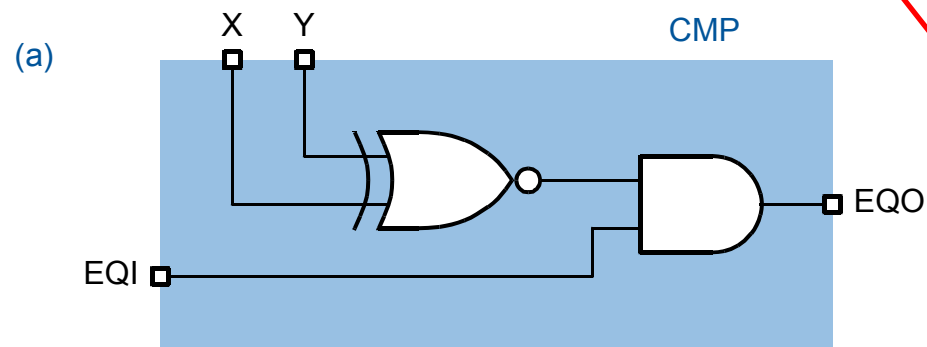
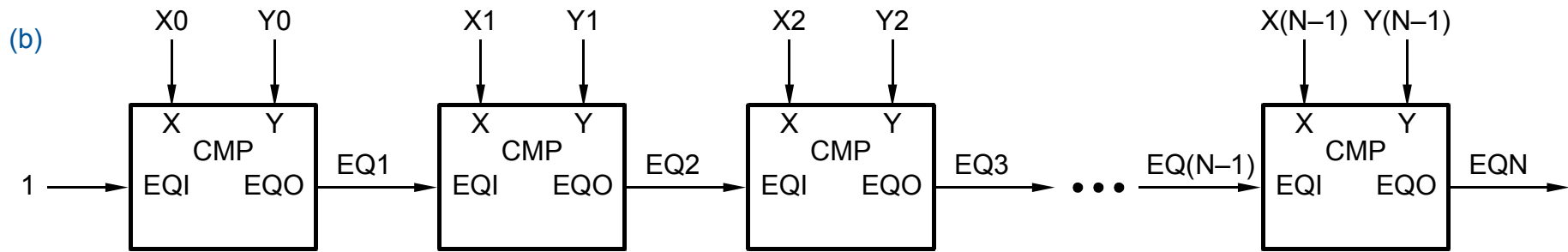
Circuito Iterativo:



Circuitos sequenciais vs. iterativos

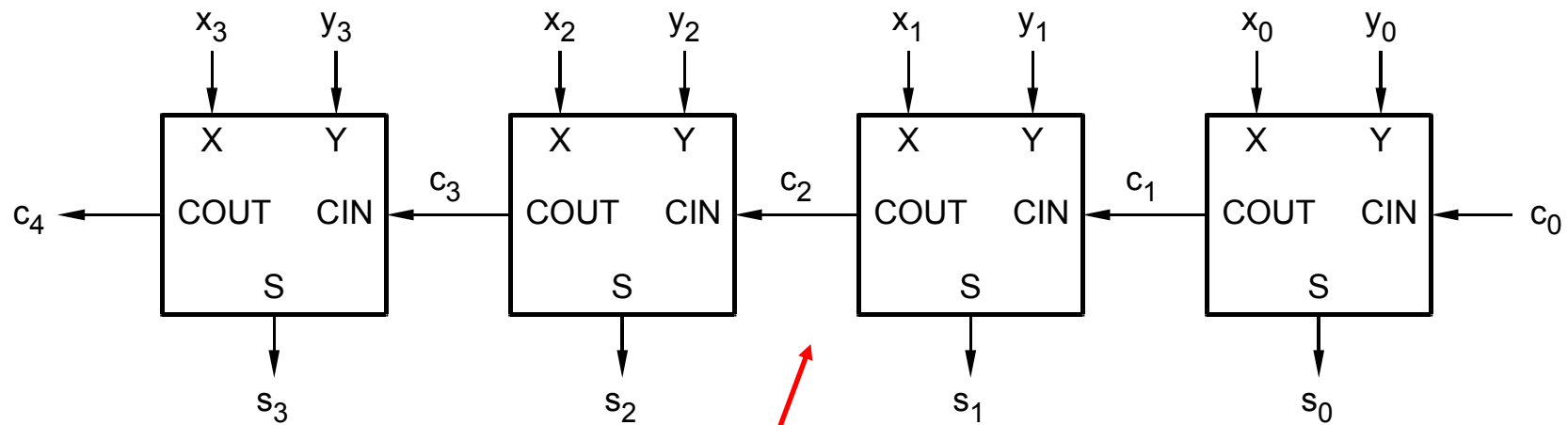
Circuito Sequencial:





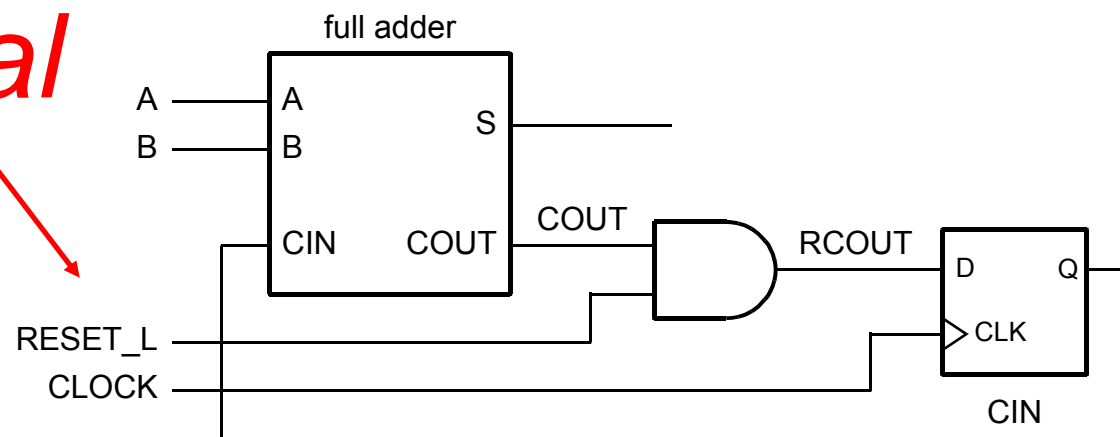
Comparador:
iterativo

sequencial



Somador: *iterativo*

sequencial



Exercícios

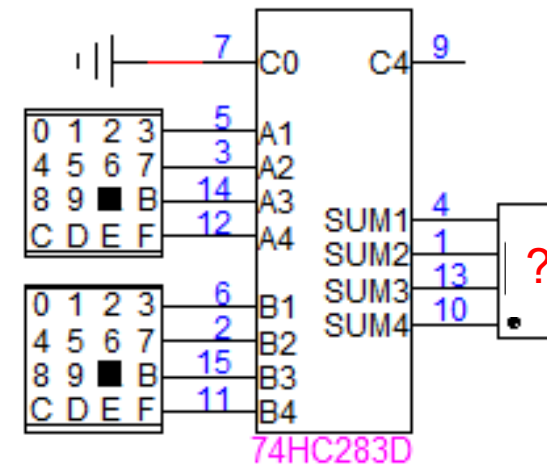
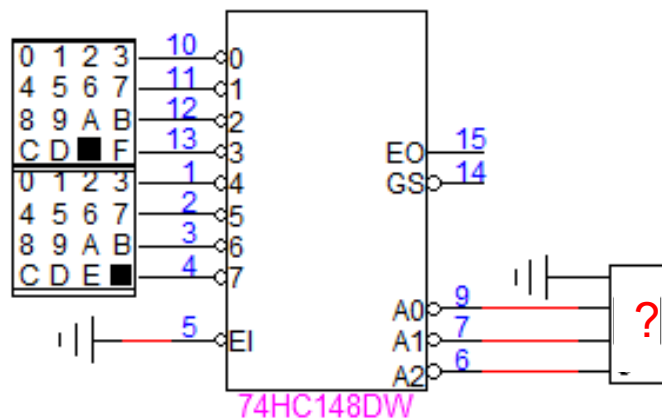
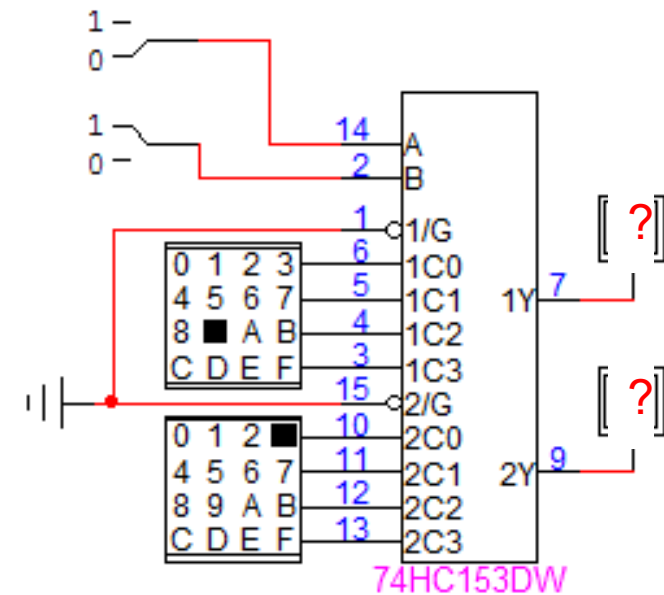
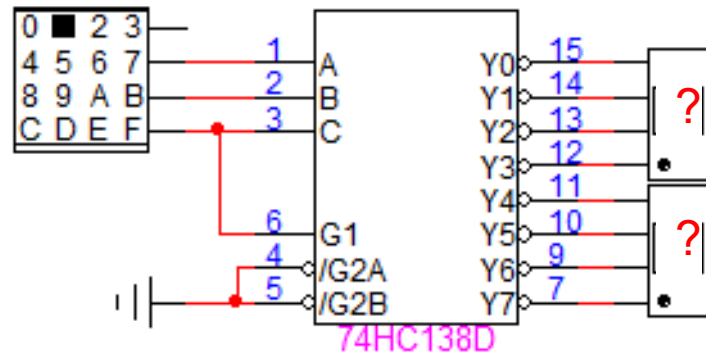
Utilizando o contador binário módulo 16, componente 74x163, construa um circuito que implemente sequência de contagem seguinte: (4,5,6,7,8,9,A,B,C), (4,5,6,7,8,9,A,B,C), ...

Projete uma linha de atraso de comprimento N, variável entre 1 e 16.
O comprimento desejado é especificado por 4 entradas A3 A2 A1 A0 contendo a representação binária de N-1.
Utilize apenas *shift registers* de 4 bits e multiplexers 4:1.
Minimize o número de componentes usados.



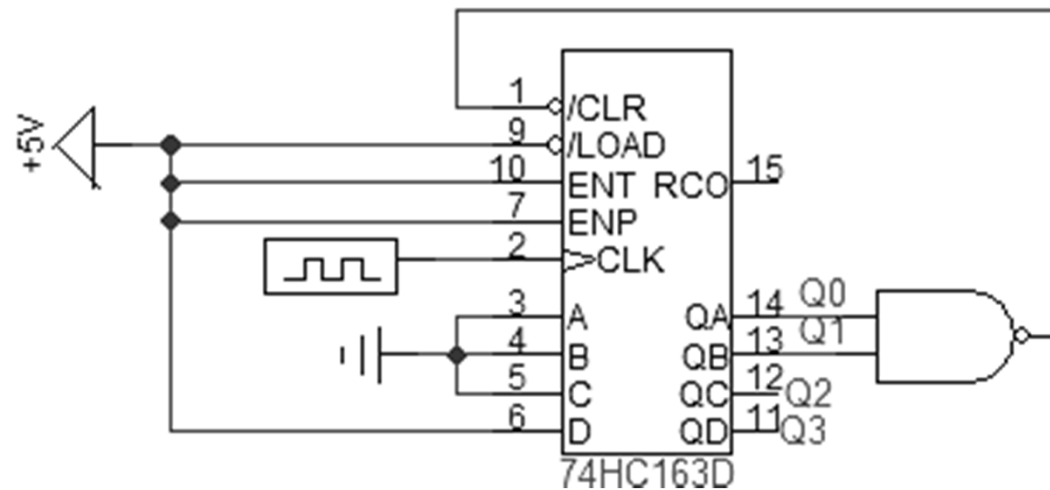
Exercícios (cont.)

Determine os valores mostrados nos *displays*:



Exercícios (cont.)

O circuito da figura ao lado inclui um contador binário módulo 16 com entradas de load e clear síncronas. Determine a sequência de contagem seguida pelo circuito.



Exercícios (cont.)

Analise o circuito da figura ao lado que inclui um registo de deslocamento de 4 bits, que faz deslocamento no sentido $Q0 \rightarrow Q3$. Assumindo que o estado atual é $Q3Q2Q1Q0 = 0011$, qual será o estado seguinte?

