

# MC542

## Organização de Computadores Teoria e Prática

2007

Prof. Paulo Cesar Centoducatte

[ducatte@ic.unicamp.br](mailto:ducatte@ic.unicamp.br)

[www.ic.unicamp.br/~ducatte](http://www.ic.unicamp.br/~ducatte)

# MC542

## Circuitos Lógicos

### Projeto de Circuitos Seqüenciais

“DDCA” - (Capítulo 3)

“FDL” - (Capítulo 7)

# Projeto de Circuitos Seqüenciais

## Sumário

- **Introdução**
- **Latches e Flip-Flops**
- **Projeto de Circuitos Síncronos**
- **Registradores**
  - **Uso de Registradores com Barramento**
- **Registradores de Deslocamento**
- **Contadores**
  - **Assíncronos**
  - **Síncronos**

# Introdução

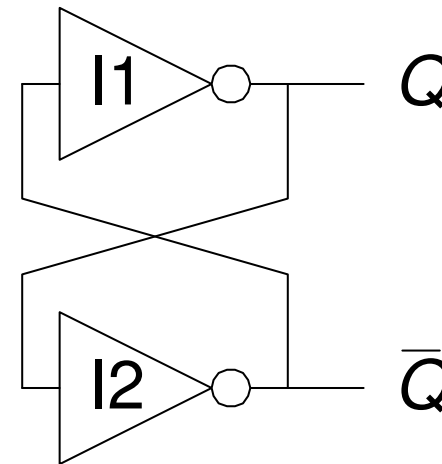
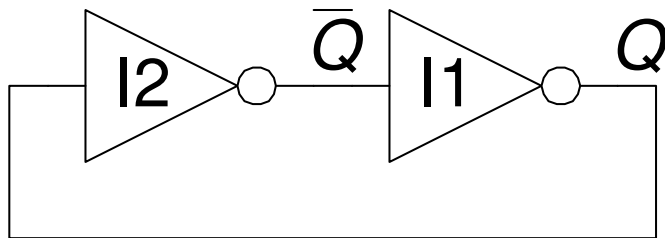
- As saídas de um circuito seqüencial dependem dos valores presente e passados de suas entradas.
- Lógica seqüencial possui memória.
- Algumas definições:
  - **Estado (State)**: conjunto de informações a cerca do circuito necessárias para se prever o seu comportamento futuro.
  - **Latches e flip-flops**: elementos de estado que armazenam um bit
  - **Circuitos seqüências Síncronos**: circuito combinacional seguido de um banco de flip-flops

# Elementos de Estados

- O estado de um circuito determina o seu comportamento futuro
- Elementos de Estado armazenam o estado
  - Circuito bi-estável
  - Latch SR
  - Latch D
  - Flip-flop D
    - » Outros tipos de flip-flops
      - JK
      - T
      - SR

# Circuito Bi-estável

- Bloco Fundamental para a construção dos outros elementos de estado
- Duas saídas:  $Q$ ,  $\bar{Q}$
- Sem entradas.

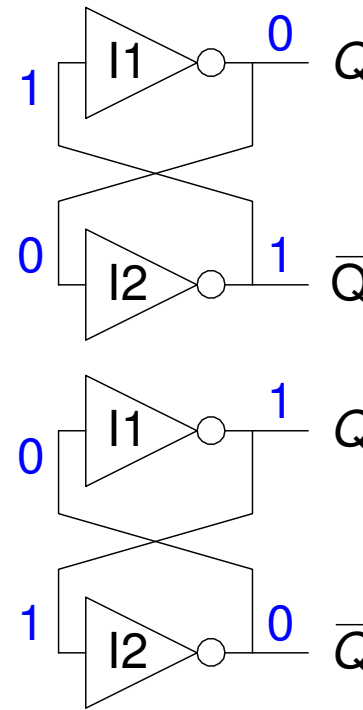


# Circuito Bi-estável: Comportamento

- Considere os dois possíveis casos:

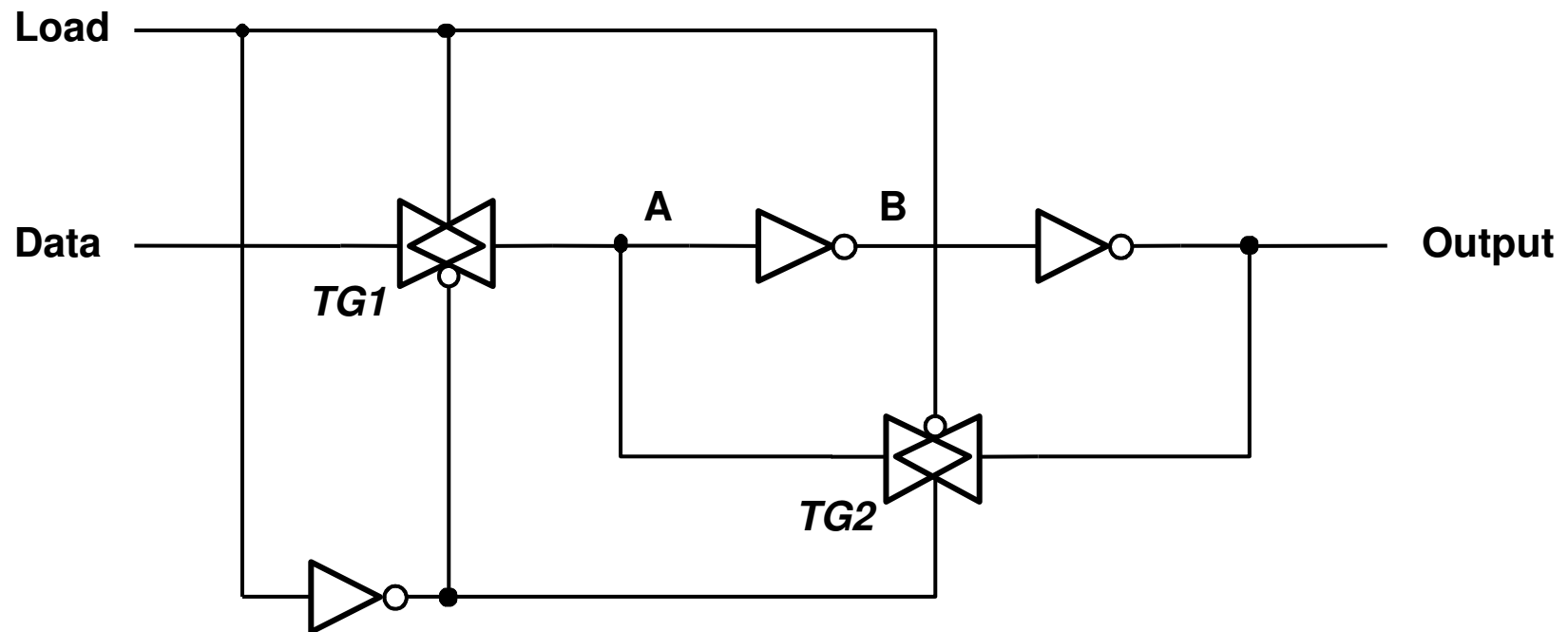
-  $Q = 0$ : então  $\bar{Q} = 1$  and  $Q = 0$

-  $Q = 1$ : então  $\bar{Q} = 0$  and  $Q = 1$



- O circuito bi-estável armazena 1 bit do estado na variável,  $Q$  (ou  $\bar{Q}$ )
- Porém não há entrada para controle do estado

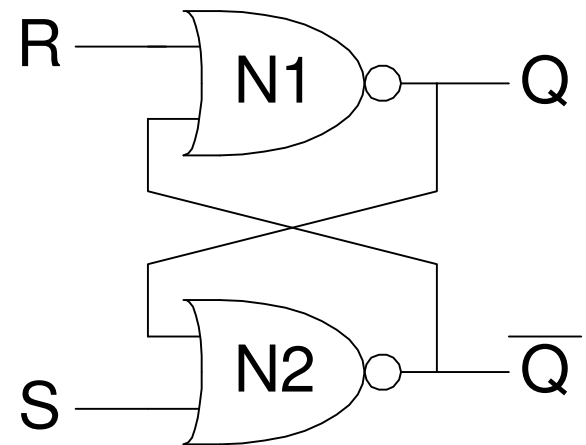
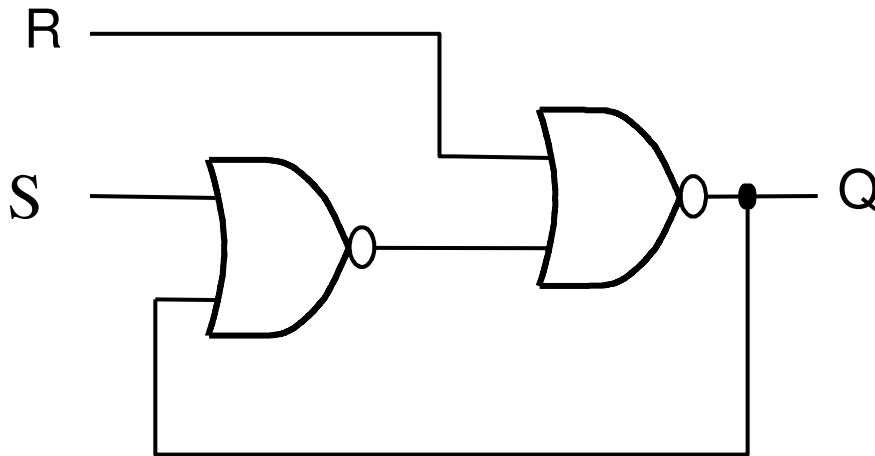
# Elemento de Memória com Controle





# SR Latch

- Latch Set/Reset (Latch SR)
- Definições
  - **Set:** Saída igual a 1
  - **Reset:** saída igual a 0
- Quando a entrada set,  $S$ , é 1 (e  $R = 0$ ),  $Q = 1$
- Quando a entrada reset,  $R$ , é 1 (e  $S = 0$ ),  $Q = 0$



- Qual a tabela verdade?

# SR Latch

- Considere os quatro casos possíveis:

$$\cdot S = 1, R = 0$$

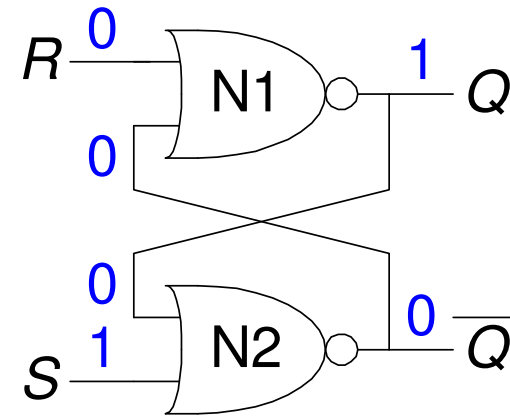
$$\cdot S = 0, R = 1$$

$$\cdot S = 0, R = 0$$

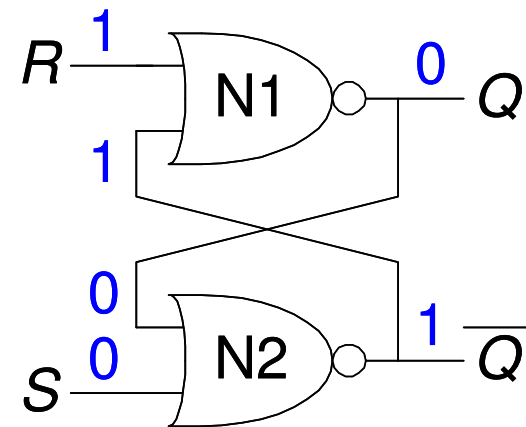
$$\cdot S = 1, R = 1$$

# SR Latch

- $S = 1, R = 0$ : then  $Q = 1$  and  $\bar{Q} = 0$

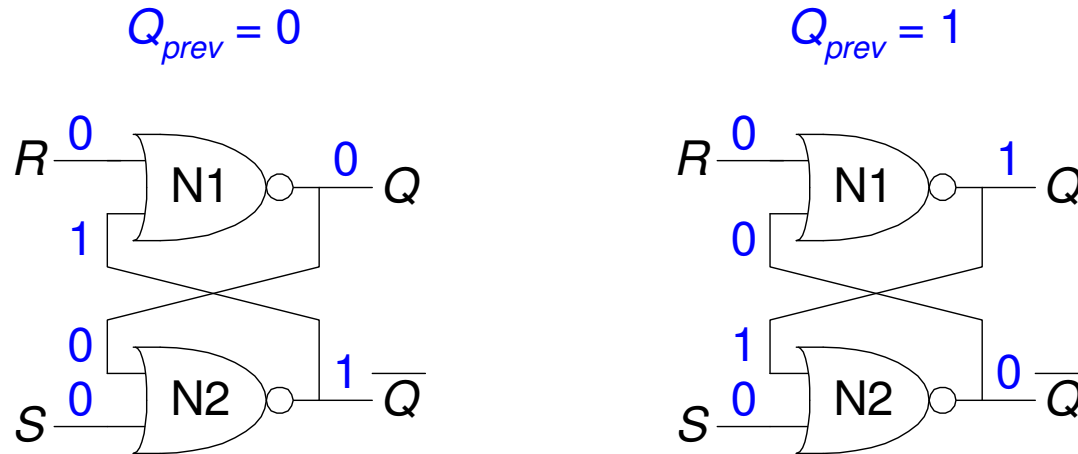


- $S = 0, R = 1$ : then  $Q = 0$  and  $\bar{Q} = 1$

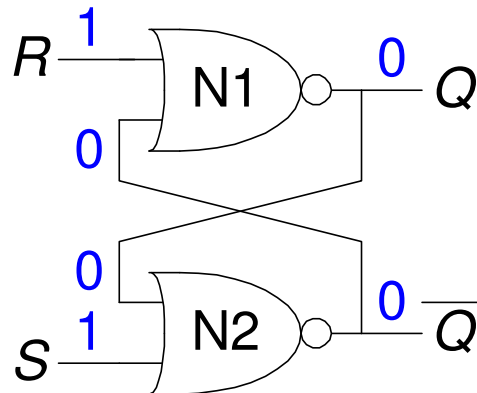


# SR Latch

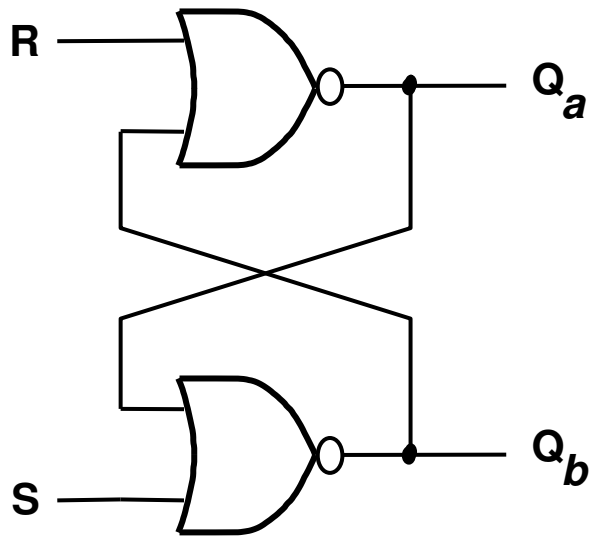
- $S = 0, R = 0$ : então  $Q = Q_{prev}$  e  $\bar{Q} = \overline{Q_{prev}}$  (memória!)



- $S = 1, R = 1$ : então  $Q = 0$  e  $\bar{Q} = 0$  (estado invalido:  $\bar{Q} \neq \text{NOT } Q$ )



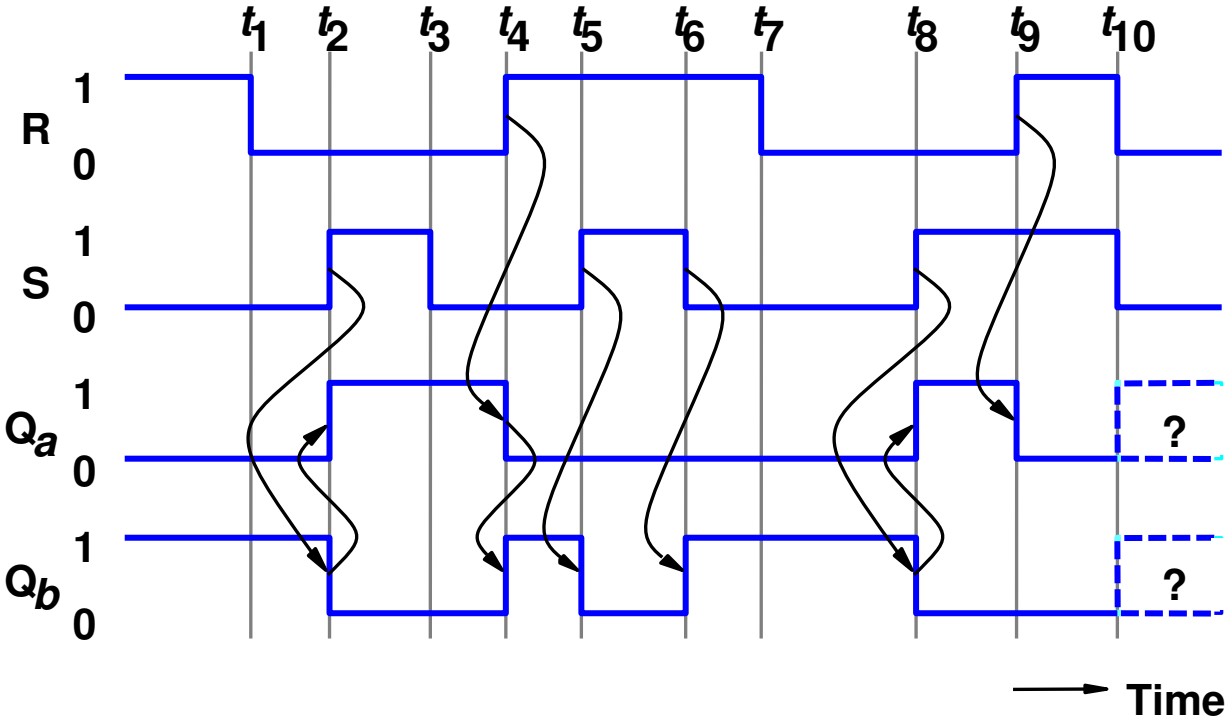
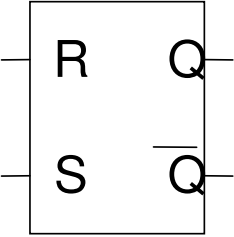
# SR Latch



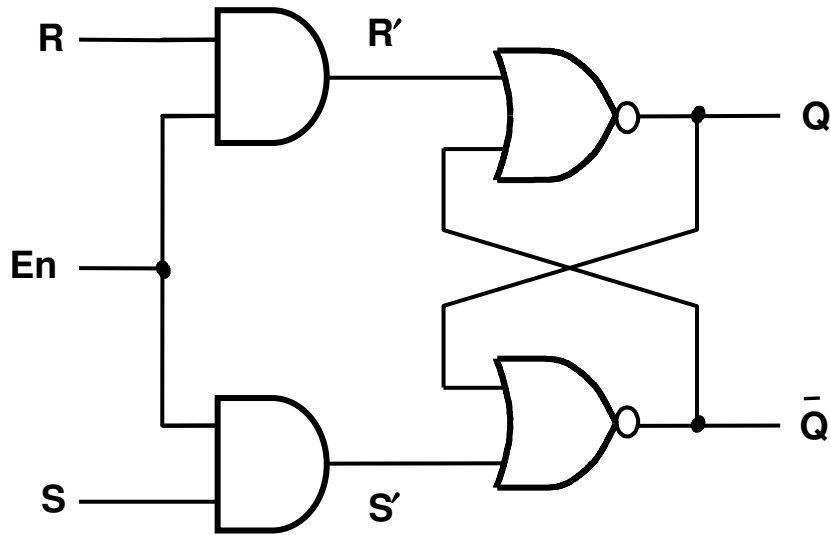
S	R	Q <sub>a</sub>	Q <sub>b</sub>	
0	0	0/1	1/0	(no change)
0	1	0	1	
1	0	1	0	
1	1	0	0	

# SR Latch

Latch SR

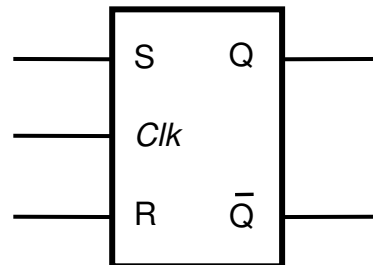
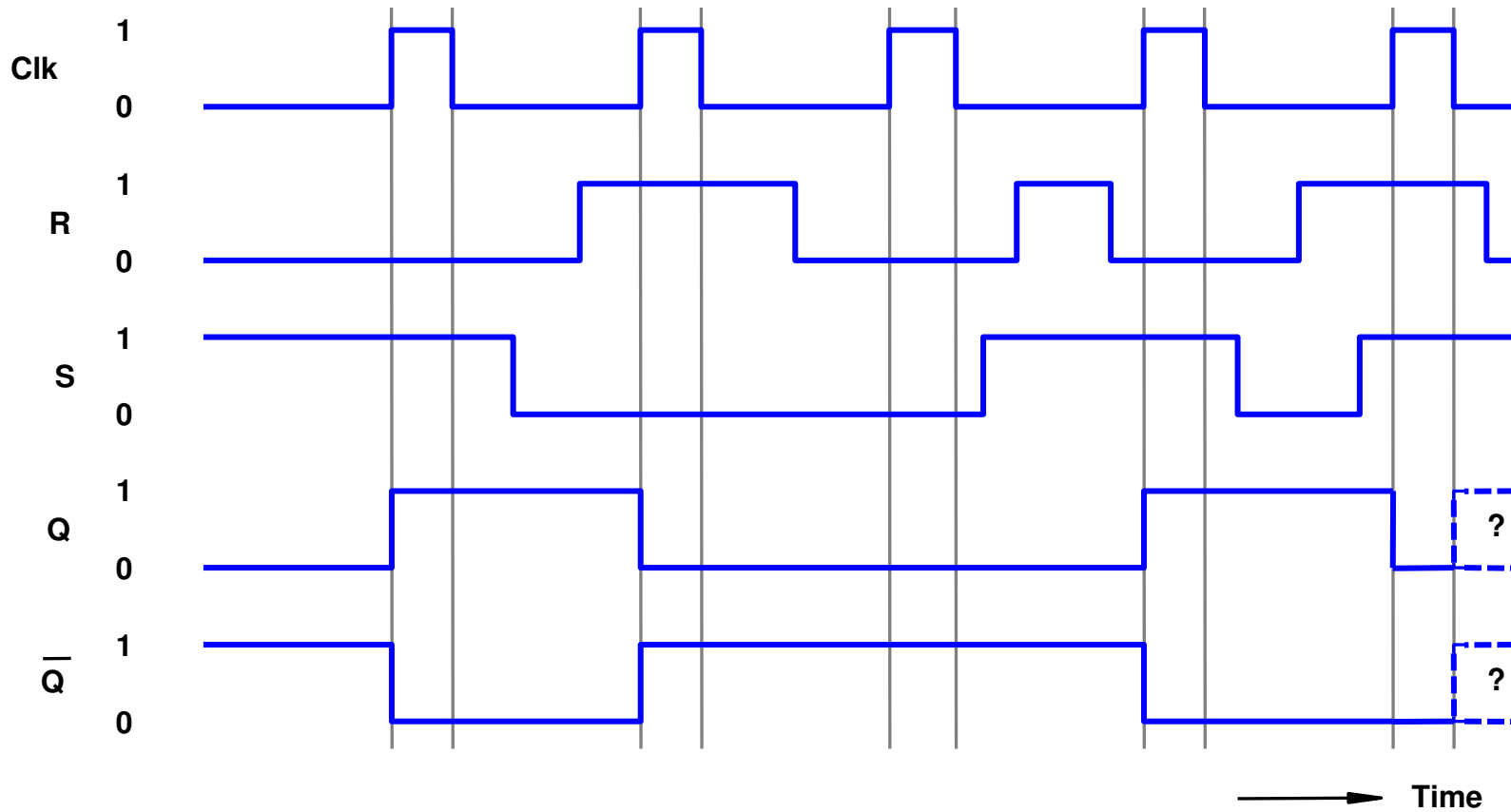


# Latch SR com Enable



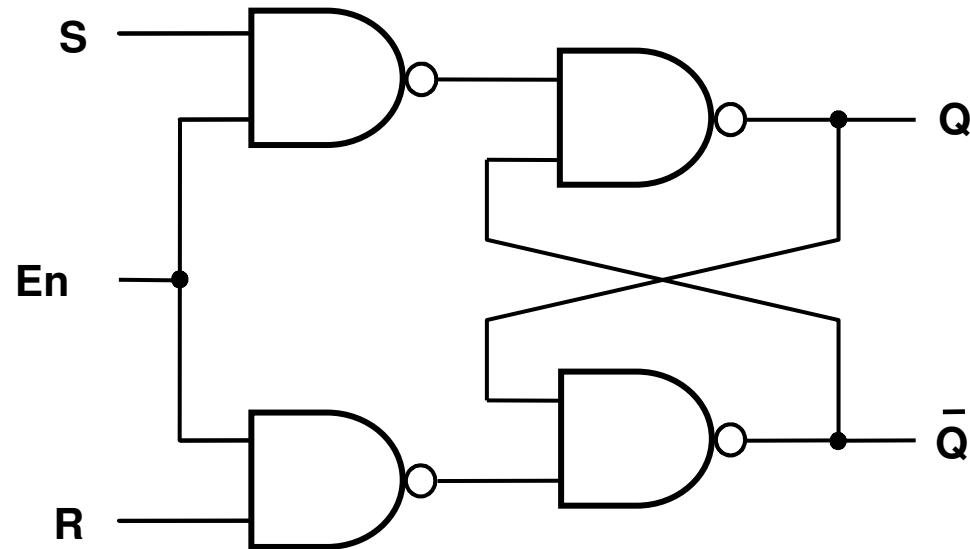
En	S	R	Q(t+1)
0	x	x	Q(t) (no change)
1	0	0	Q(t) (no change)
1	0	1	0
1	1	0	1
1	1	1	x

# Latch SR com Enable



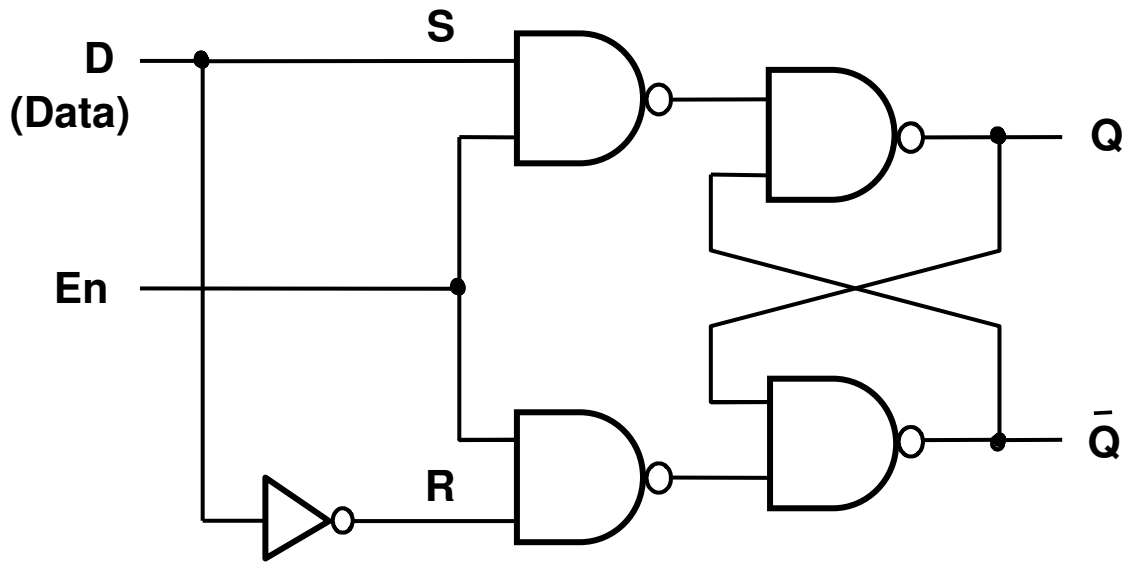


# Latch SR com Nand

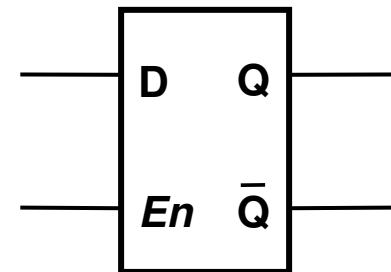
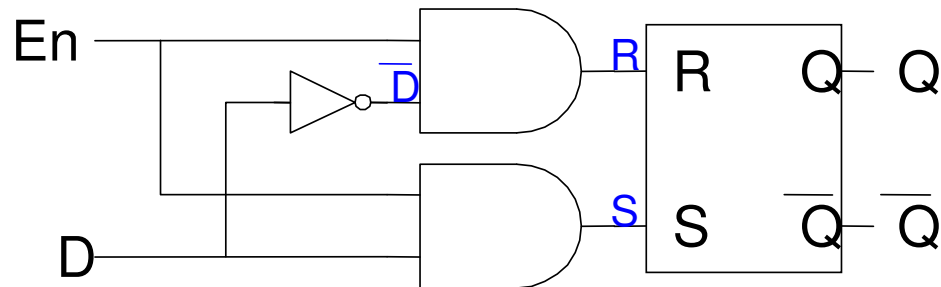


Qual a tabela verdade?

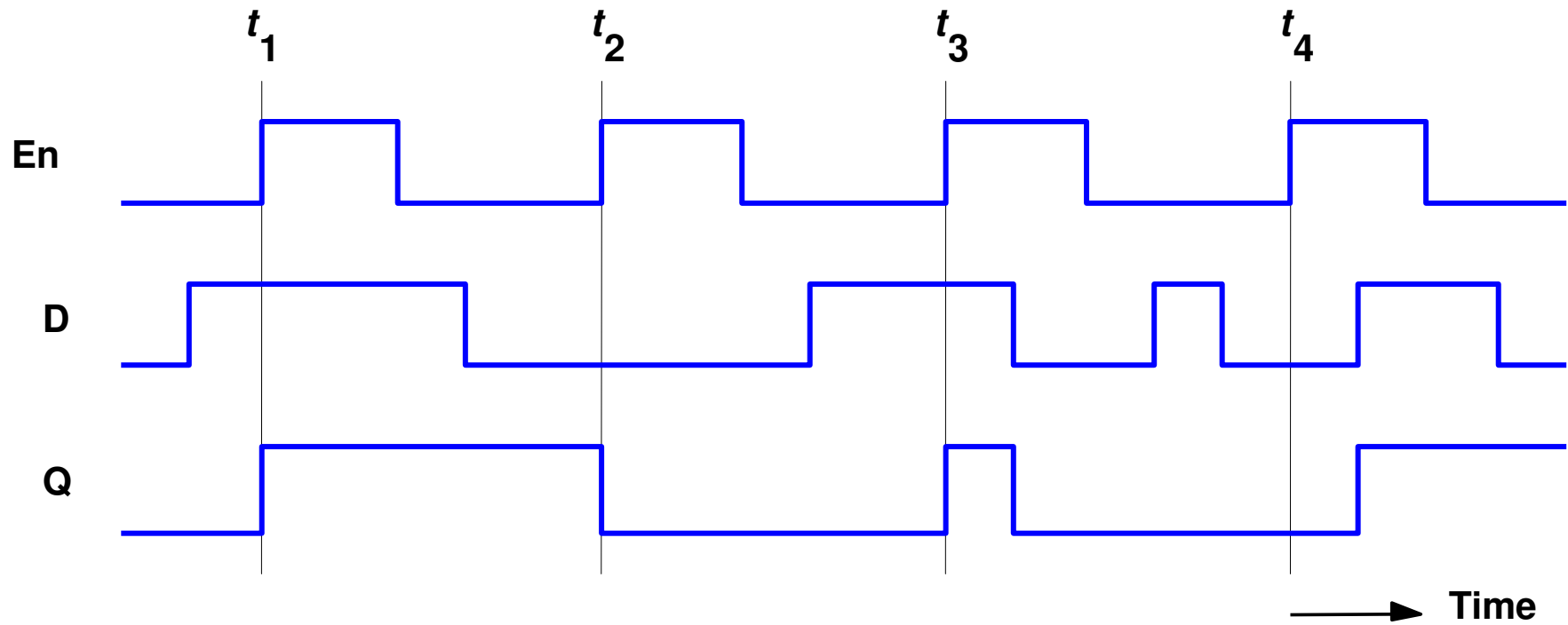
# Latch D



En	D	$Q(t+1)$
0	x	$Q(t)$
1	0	0
1	1	1



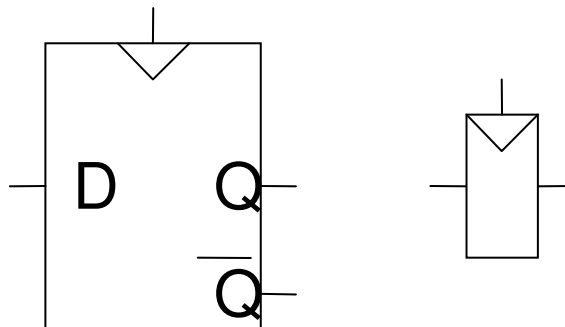
# Latch D



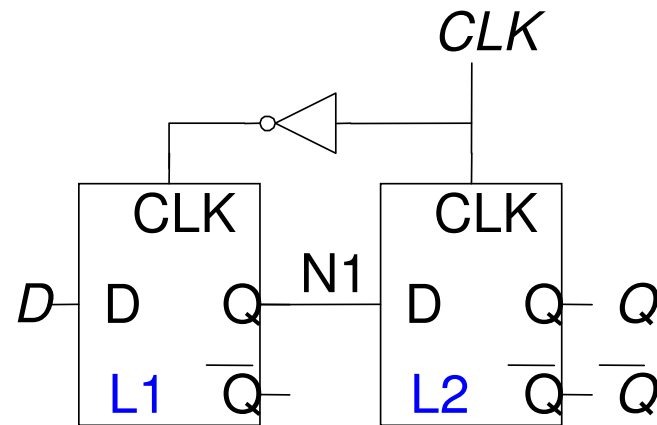
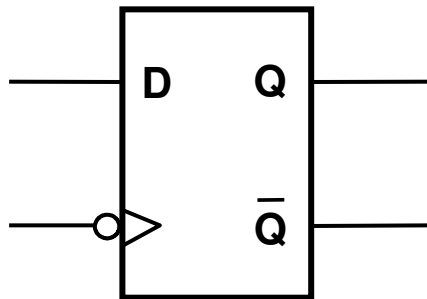
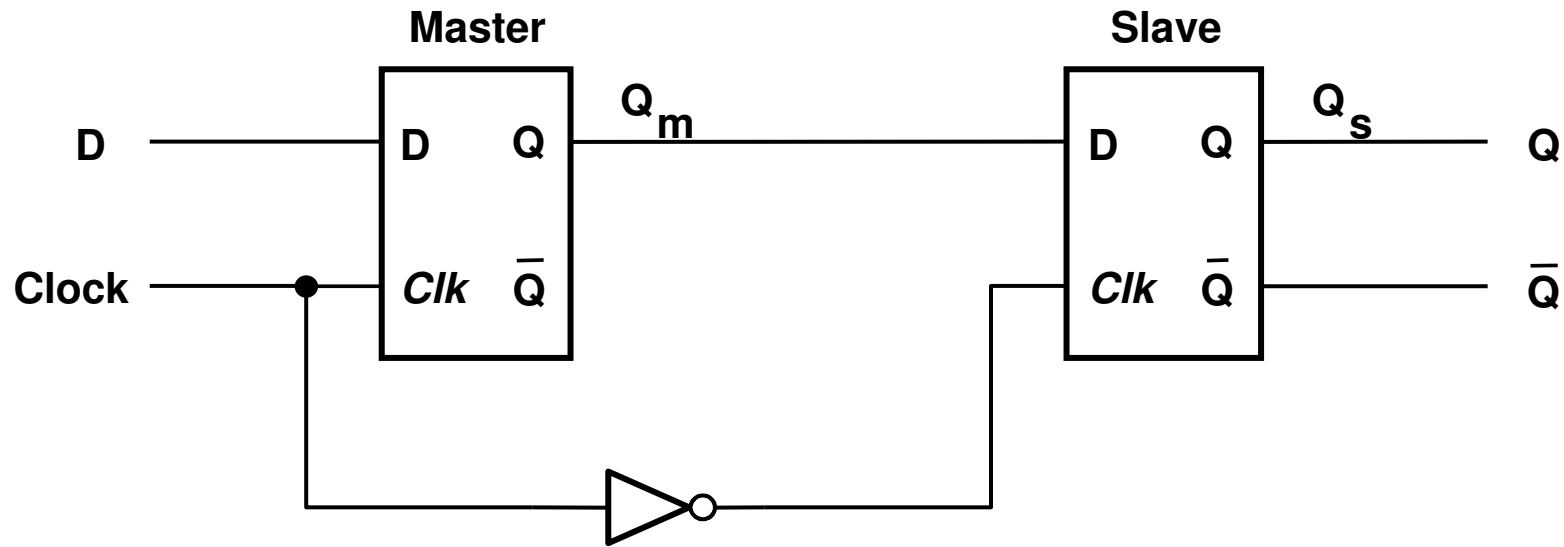
# Flip-Flop D

- Duas entradas: *CLK*, *D*
- *Q* só muda na borda (subida ou decida) do *CLK*
- O flip-flop "samples" *D* na borda do *CLK*
- O flip-flop é chamado de dispositivo *edge-triggered* devido a ser ativo na borda do clock

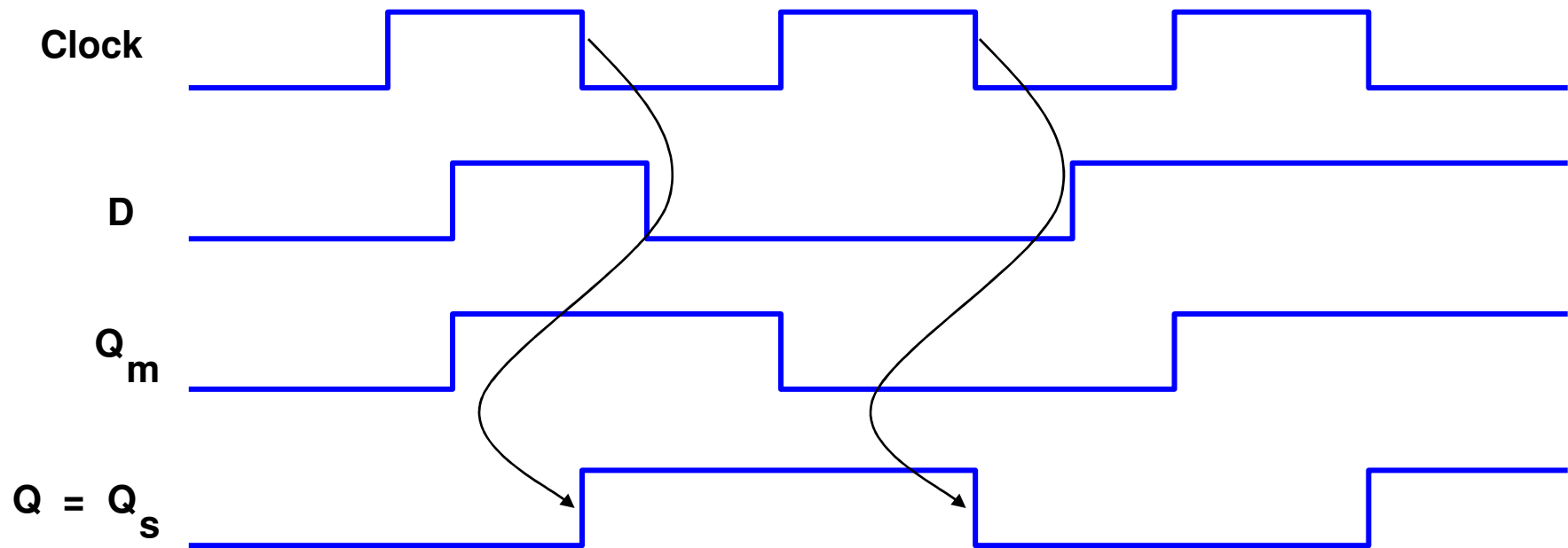
## Flip-Flop D



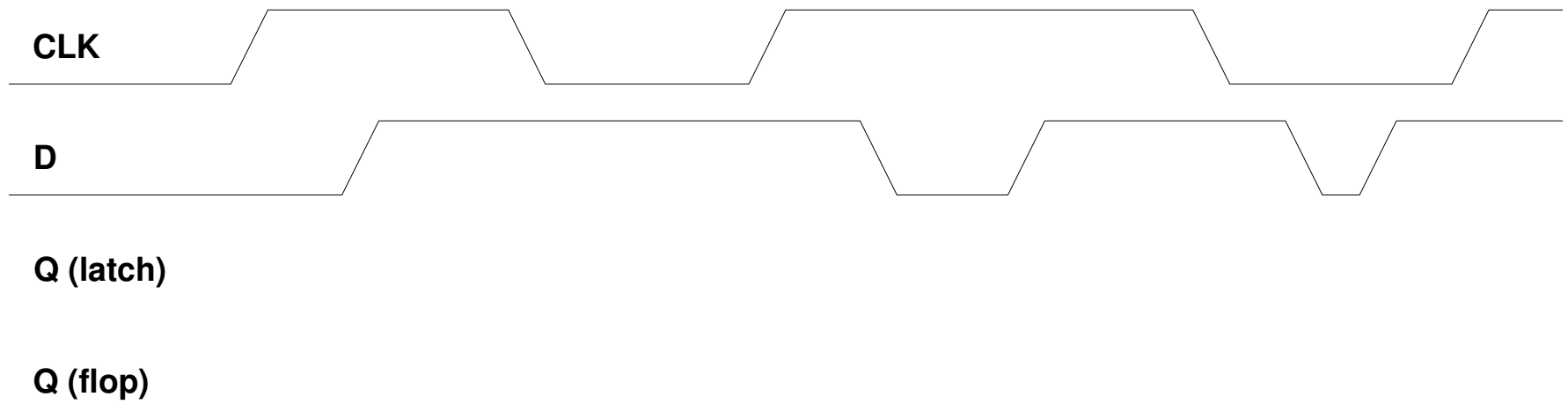
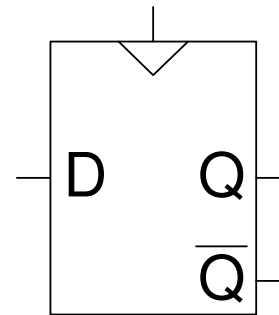
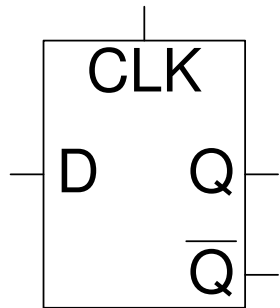
# Flip-Flop D Mestre-Escravo



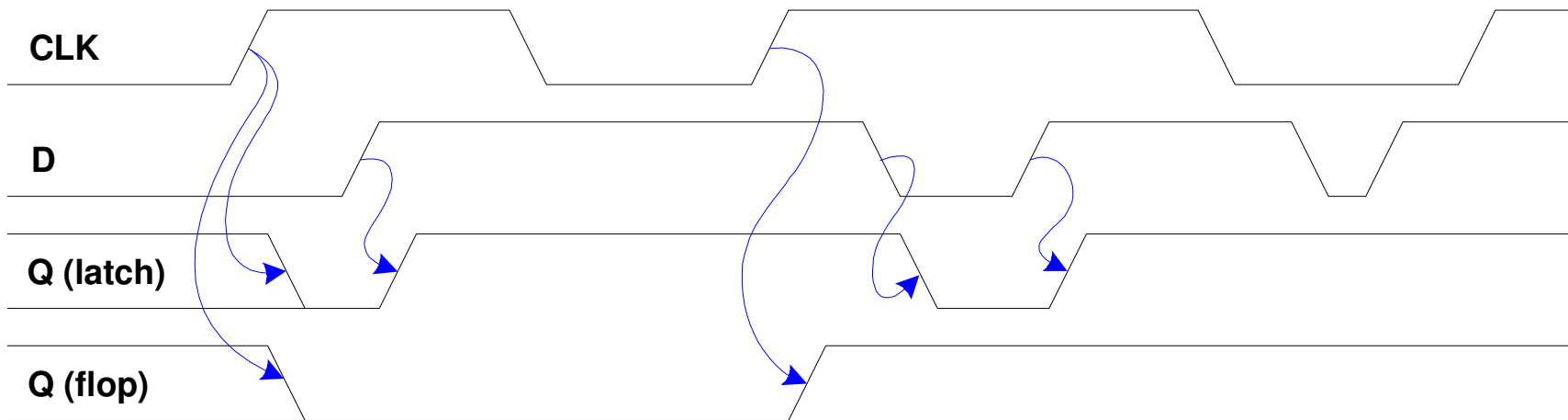
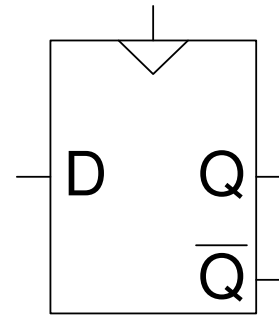
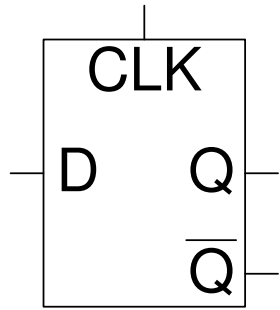
# Flip-Flop D Mestre-Escravo



# Flip-Flop D vs. Latch D

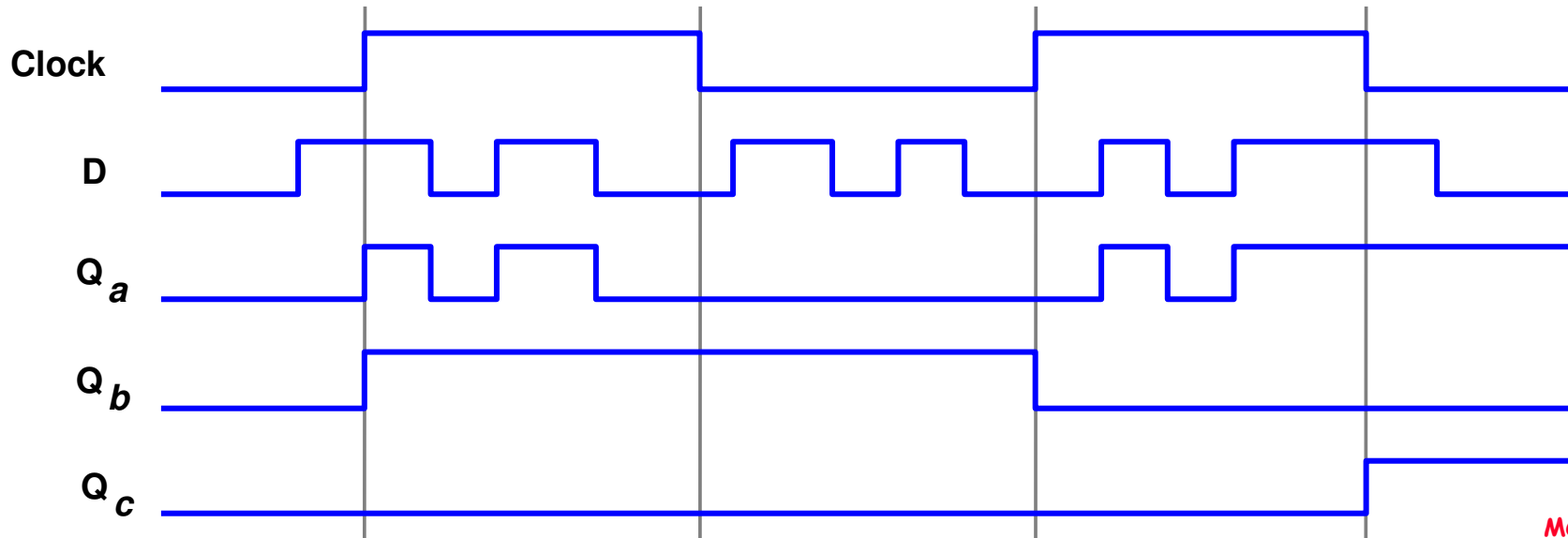
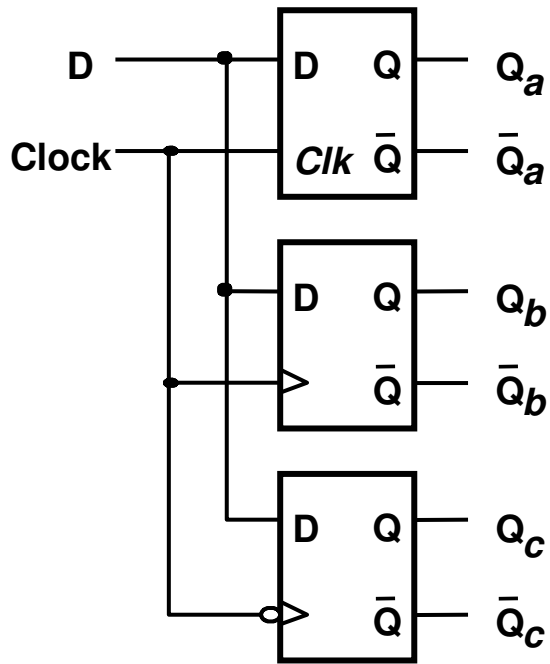


# Flip-Flop D vs. Latch D

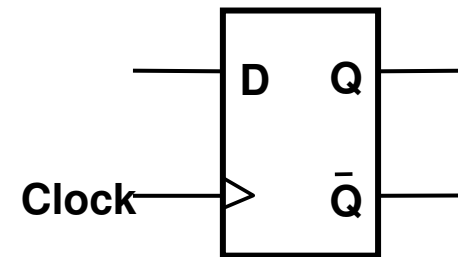
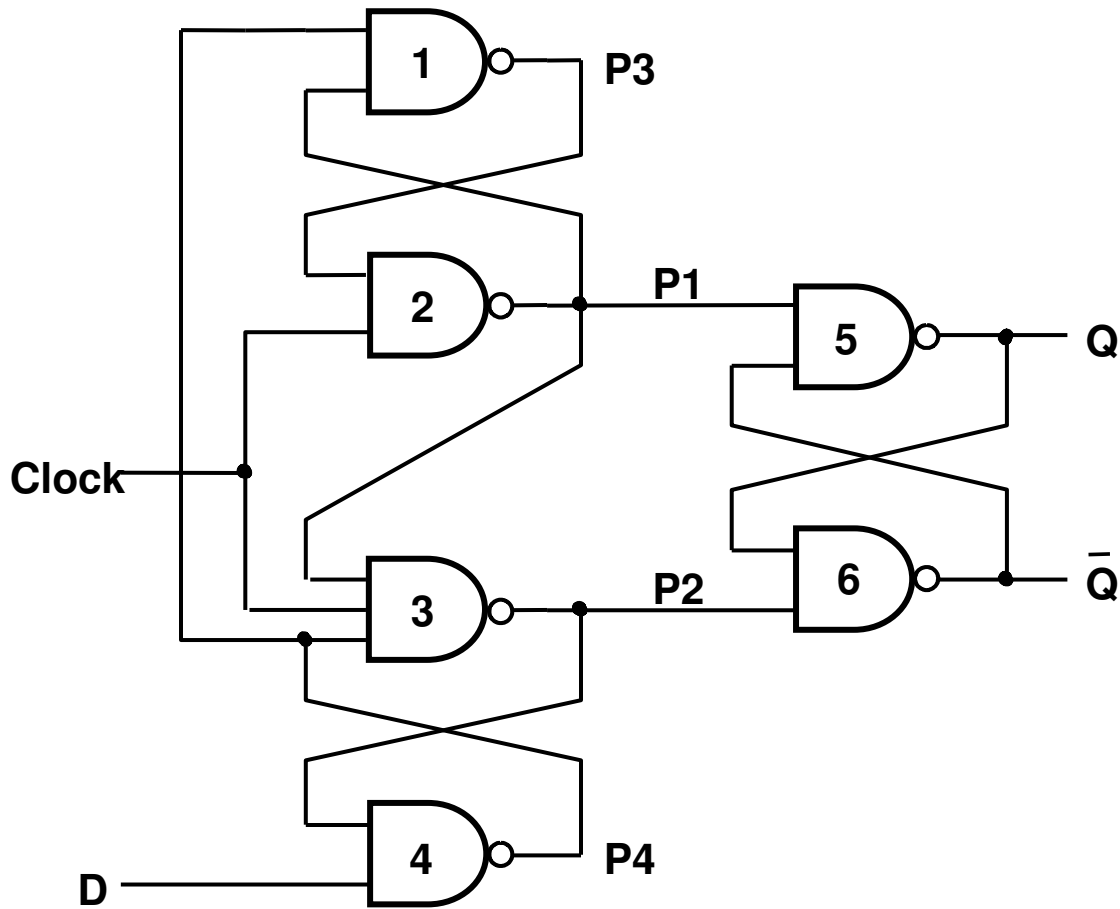




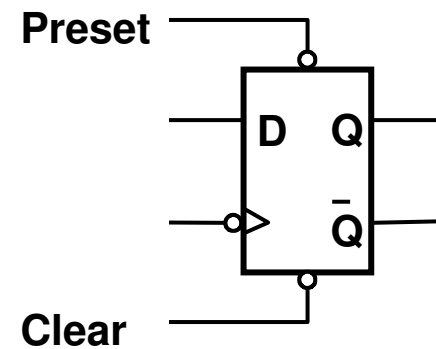
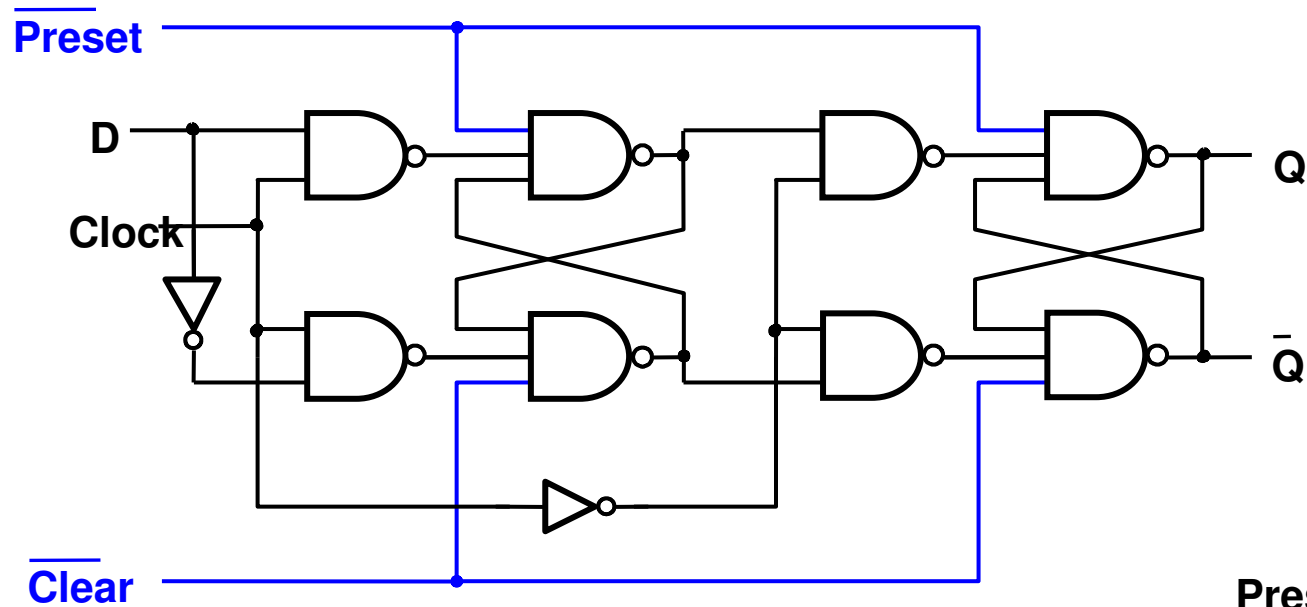
# Flip-Flop



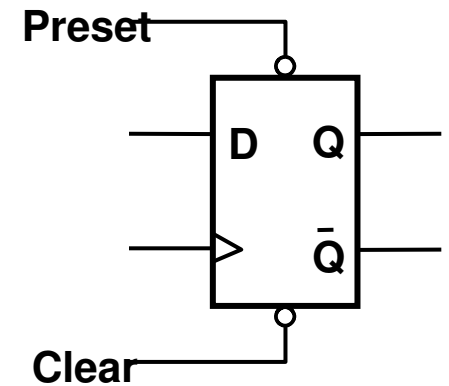
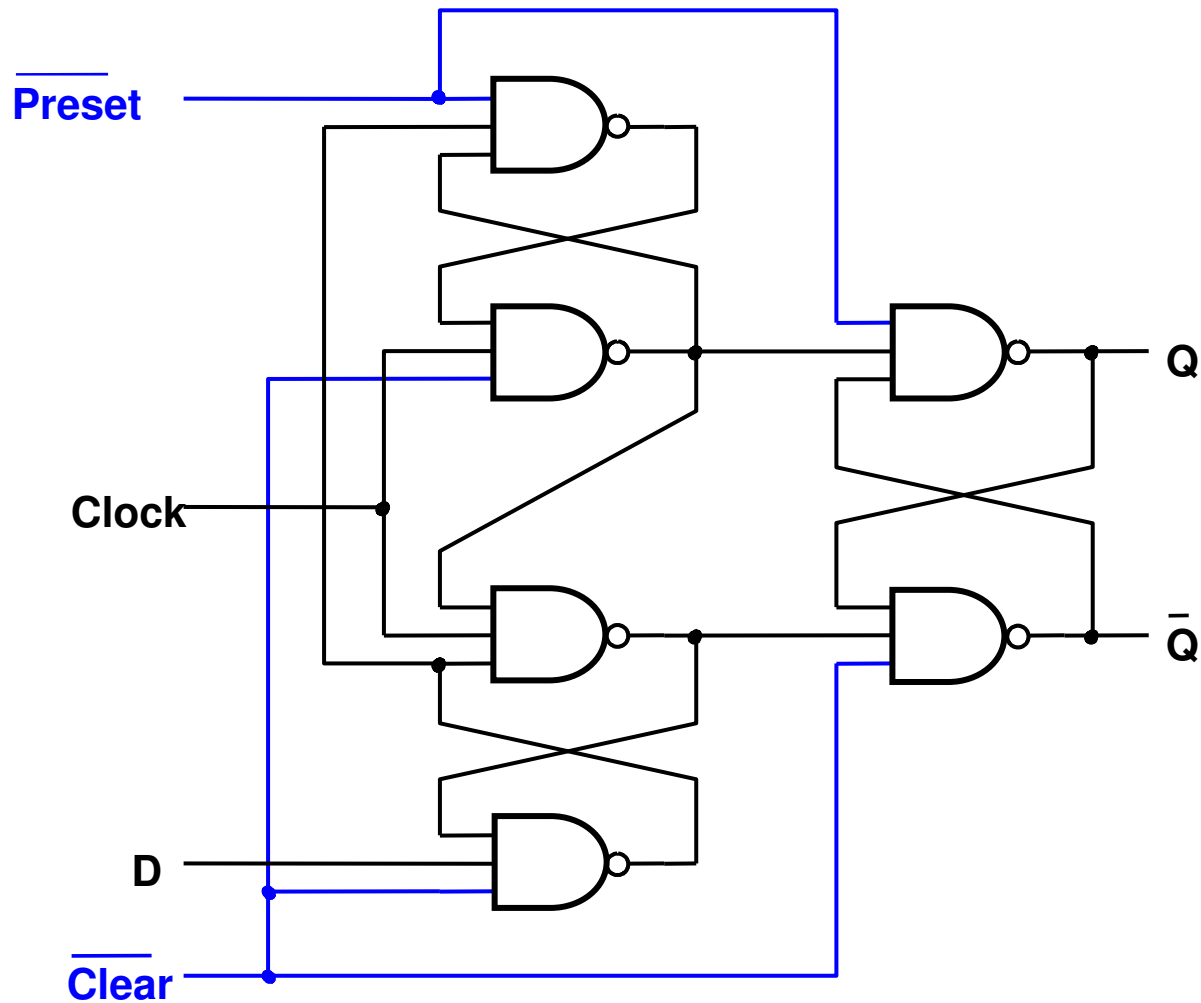
# Flip-Flop D Sensível à Borda de Subida



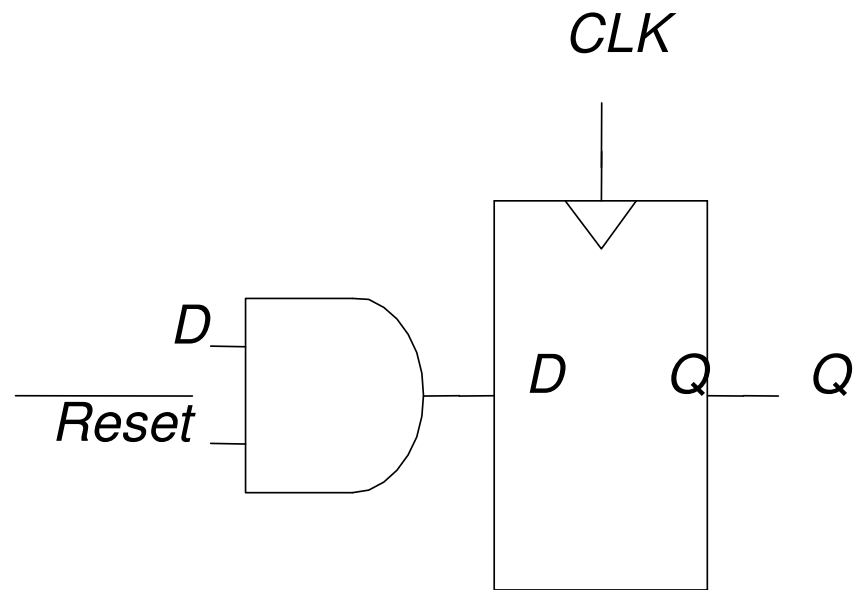
# FF D Mestre-Escravo com Preste e Clear Assíncronos



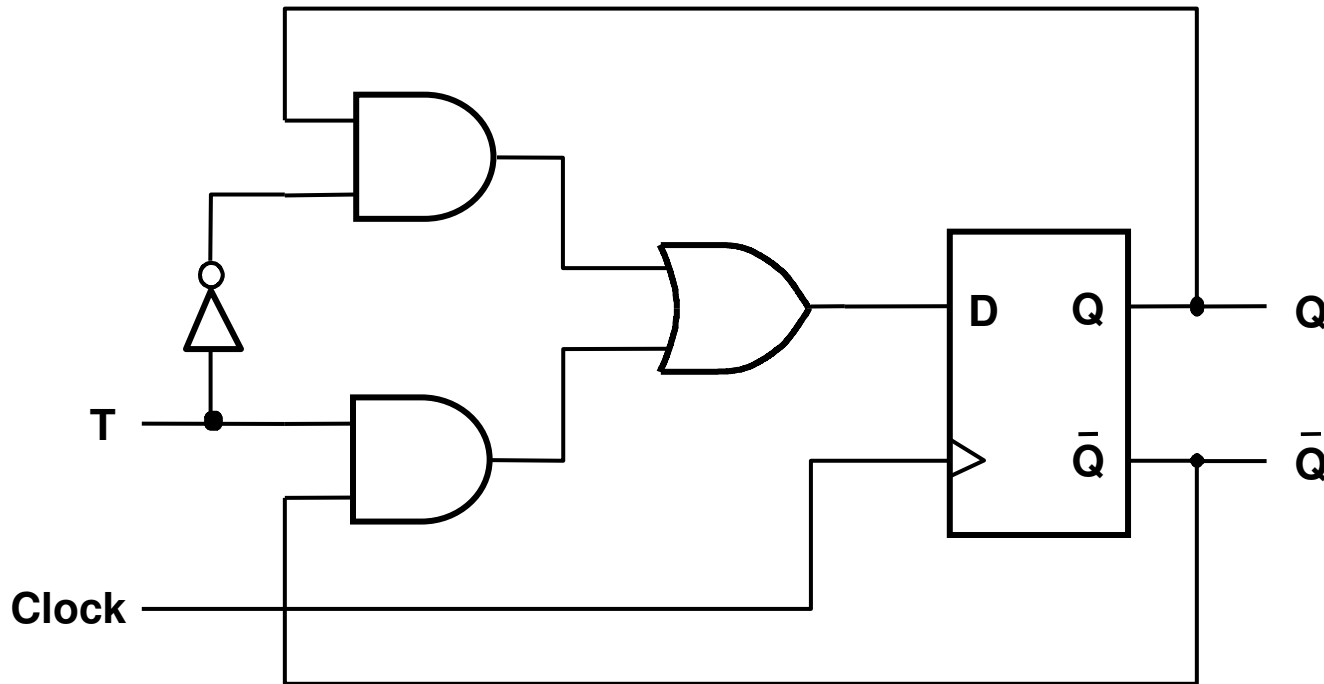
# FF D Sensível à Borda com Preste e Clear Assíncrono



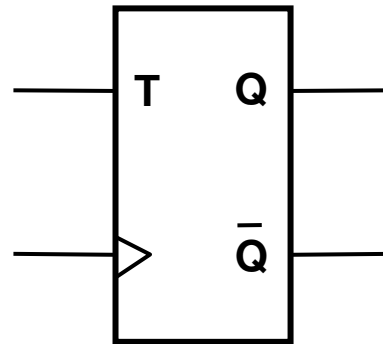
# FF D Com Reset Síncrono



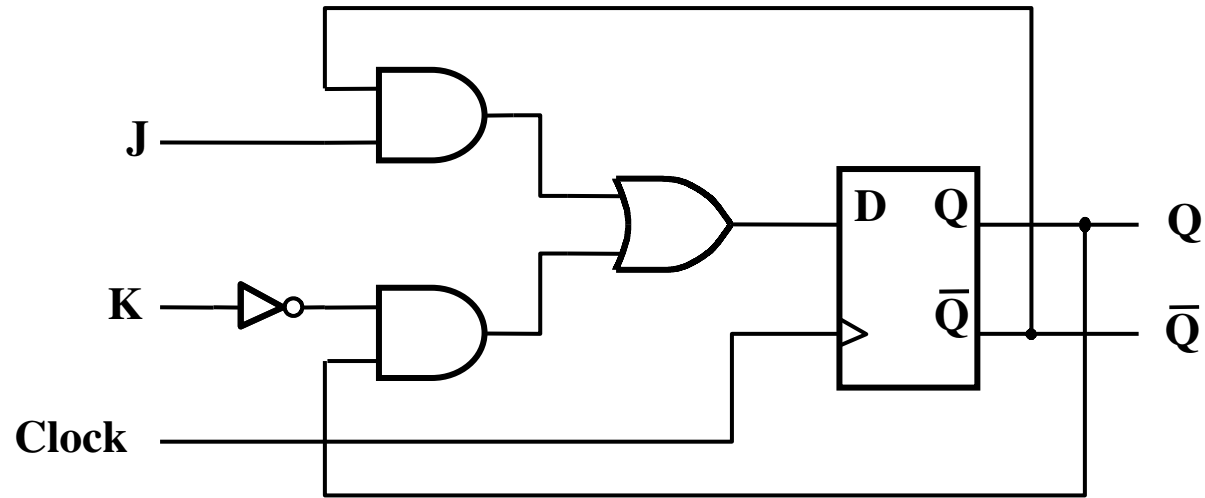
# Flip-Flop Tipo T



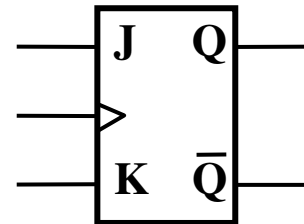
T	$Q(t+1)$
0	$Q(t)$
1	$\bar{Q}(t)$



# Flip-Flop Tipo JK

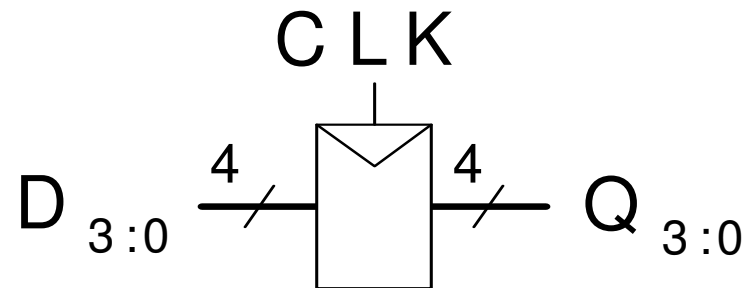
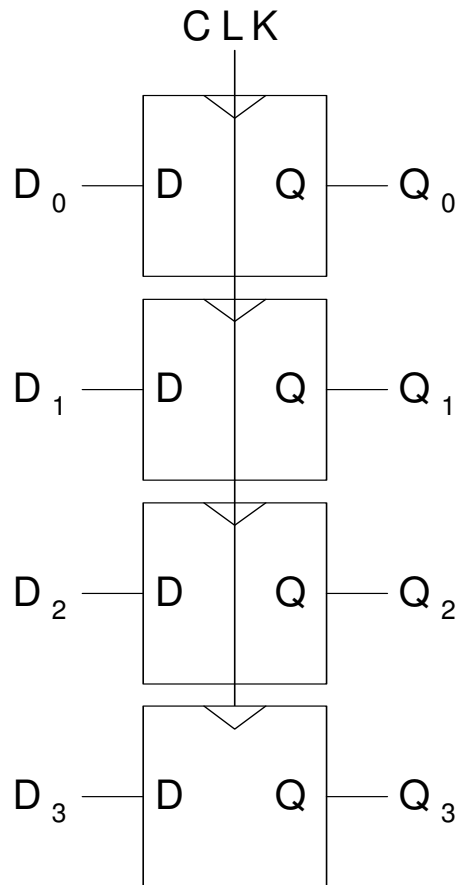


J	K	$Q(t+1)$
0	0	$Q(t)$
0	1	0
1	0	1
1	1	$\bar{Q}(t)$



# Registradores

- Conjunto de elementos de memória (flip-flops) utilizados para armazenar  $n$  bits.
- Utilizam em comum os sinais de clock e controle



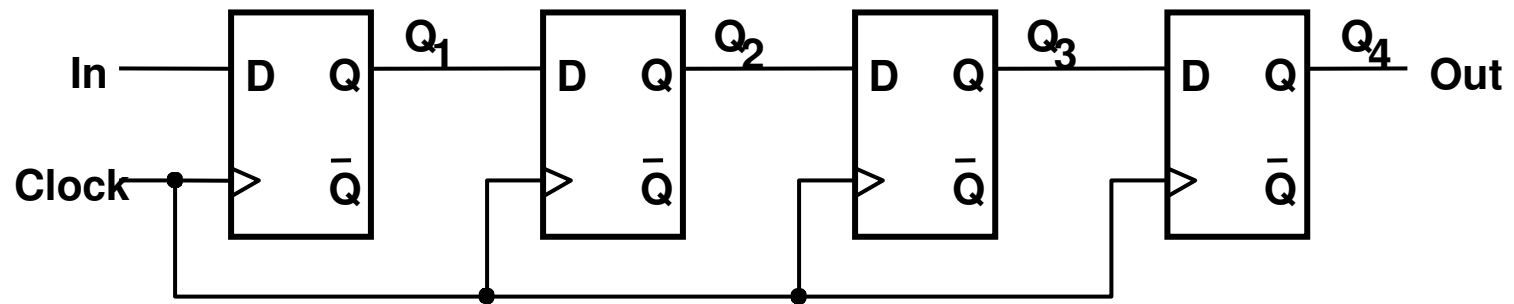


# Shift Register

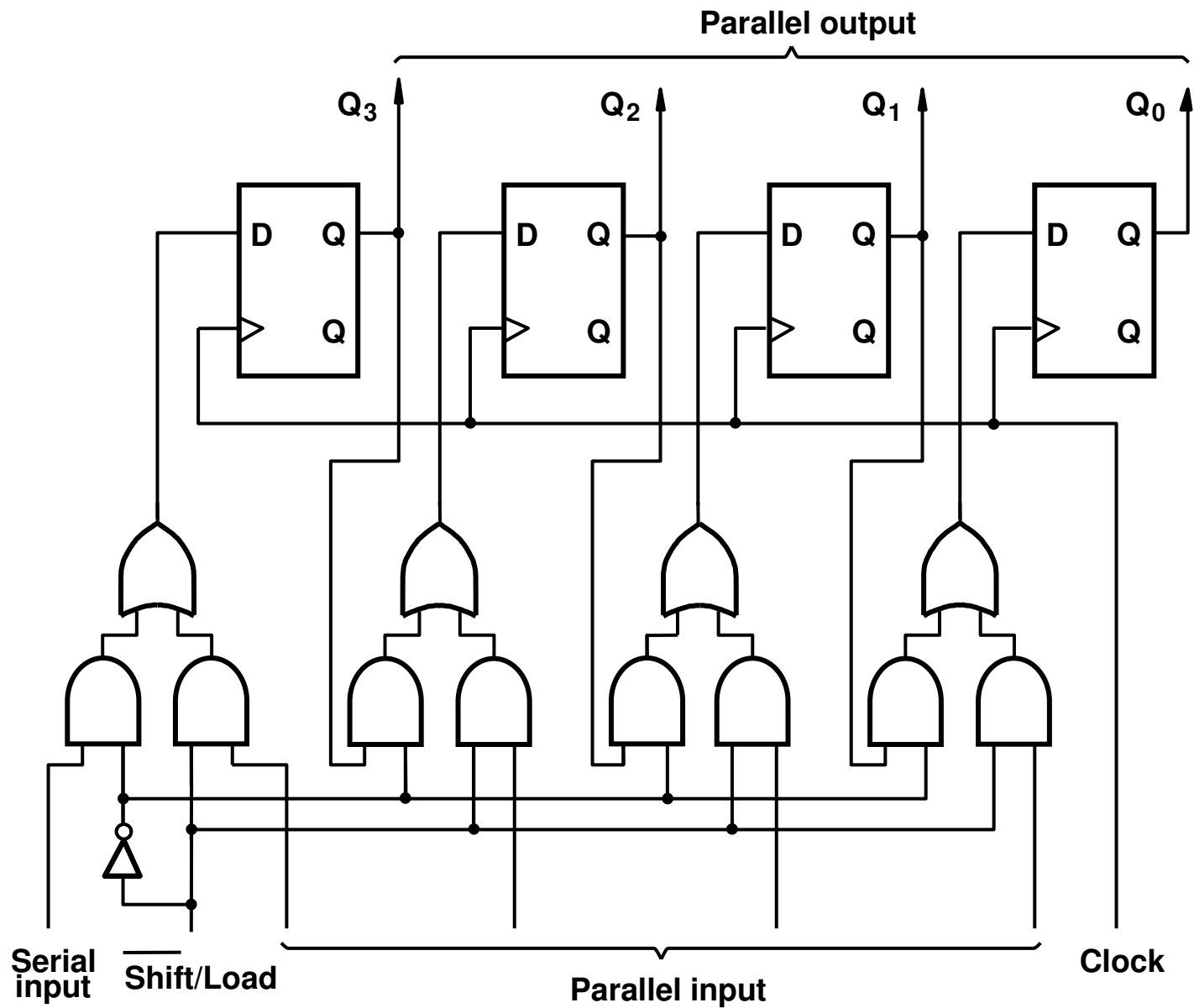
Apresenta o seguinte comportamento:

	In	Q <sub>1</sub>	Q <sub>2</sub>	Q <sub>3</sub>	Q <sub>4</sub> = Out
$t_0$	1	0	0	0	0
$t_1$	0	1	0	0	0
$t_2$	1	0	1	0	0
$t_3$	1	1	0	1	0
$t_4$	1	1	1	0	1
$t_5$	0	1	1	1	0
$t_6$	0	0	1	1	1
$t_7$	0	0	0	1	1

# Shift Register



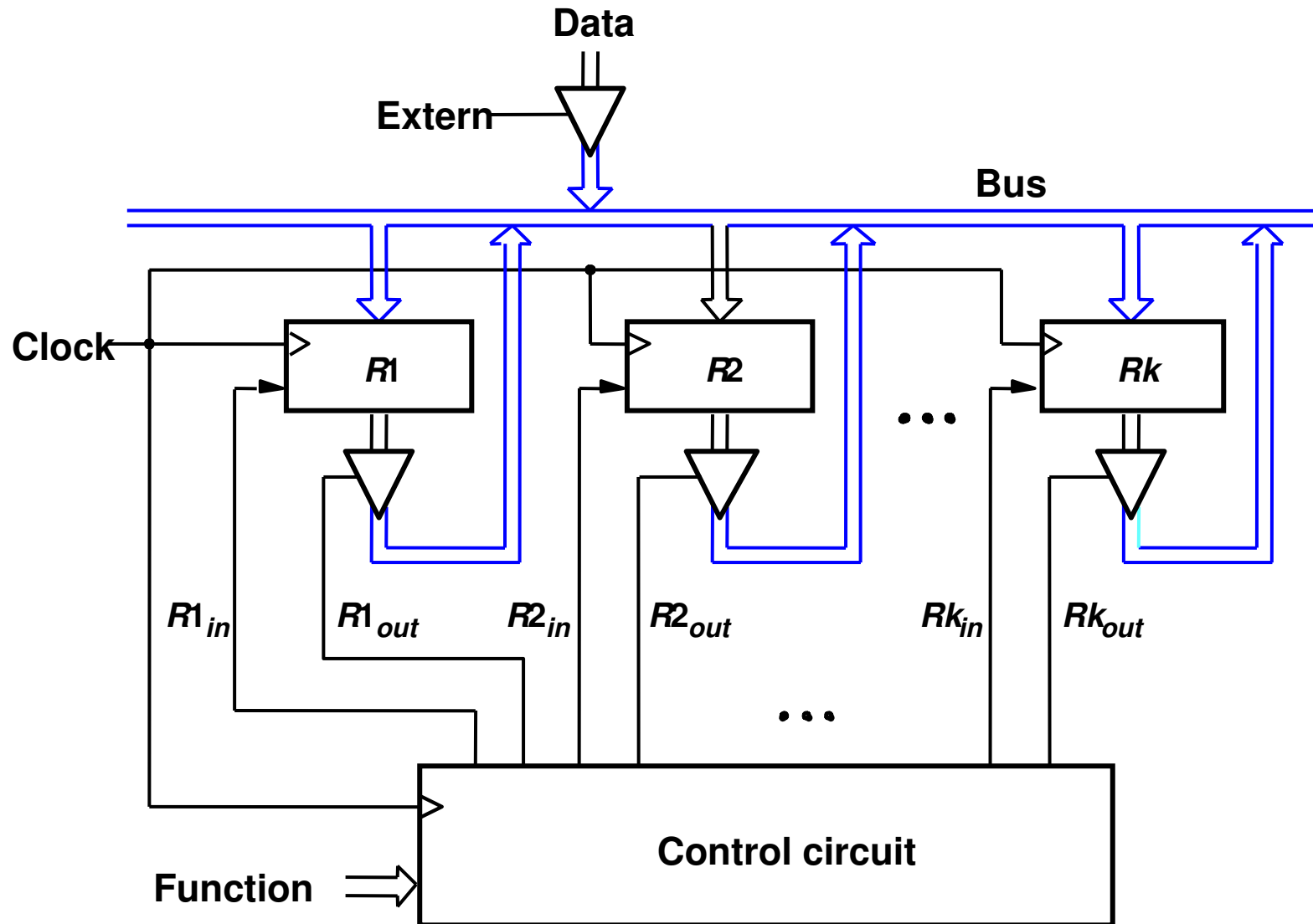
# Shift Register com Carga Paralela



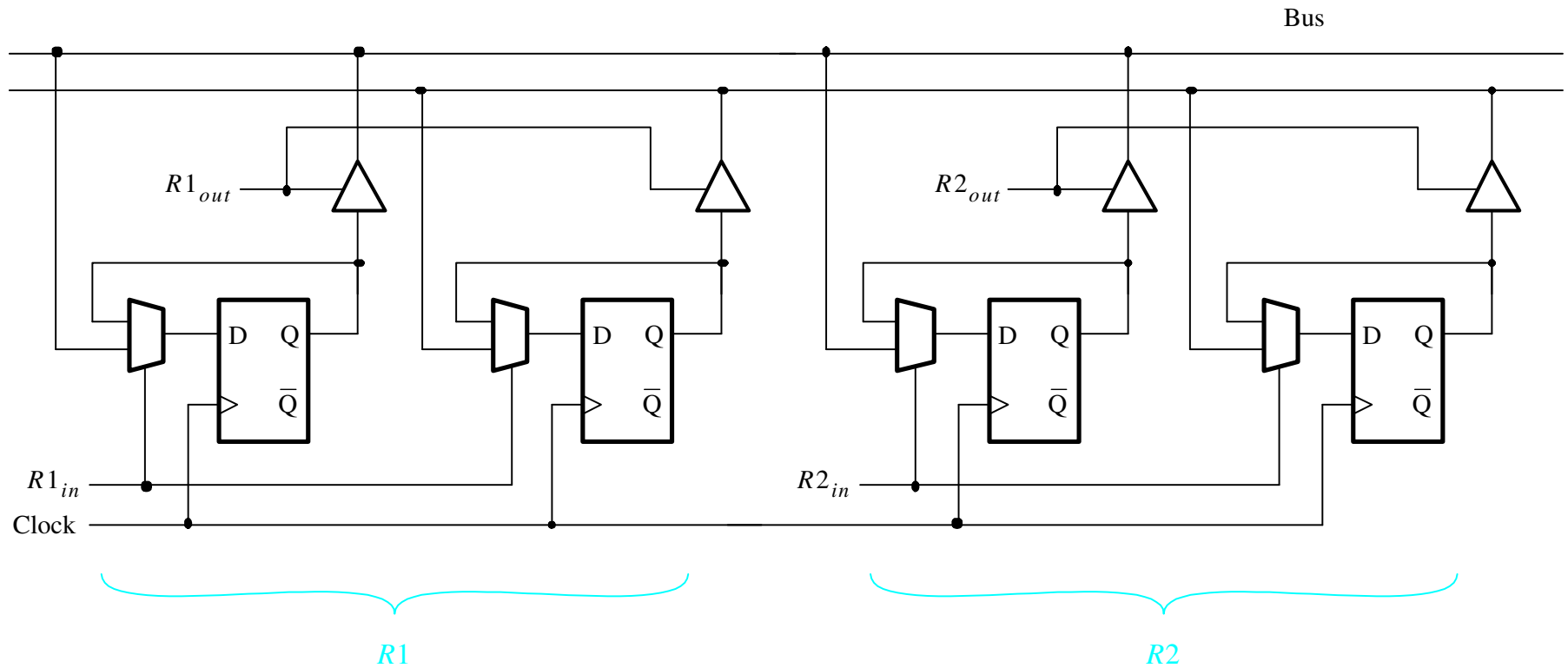
# Shift Register Universal

- Entrada Serial
    - Deslocamento a Esquerda
    - Deslocamento a Direita
  - Carga Paralela
  - Saída Paralela
- 
- Exercício: Desenhe o Diagrama do Shift Register Universal de 4 bits.

# Registadores em um Barramento



# Registadores em um Barramento



# Contadores

- **Assíncronos**
- **Síncronos**

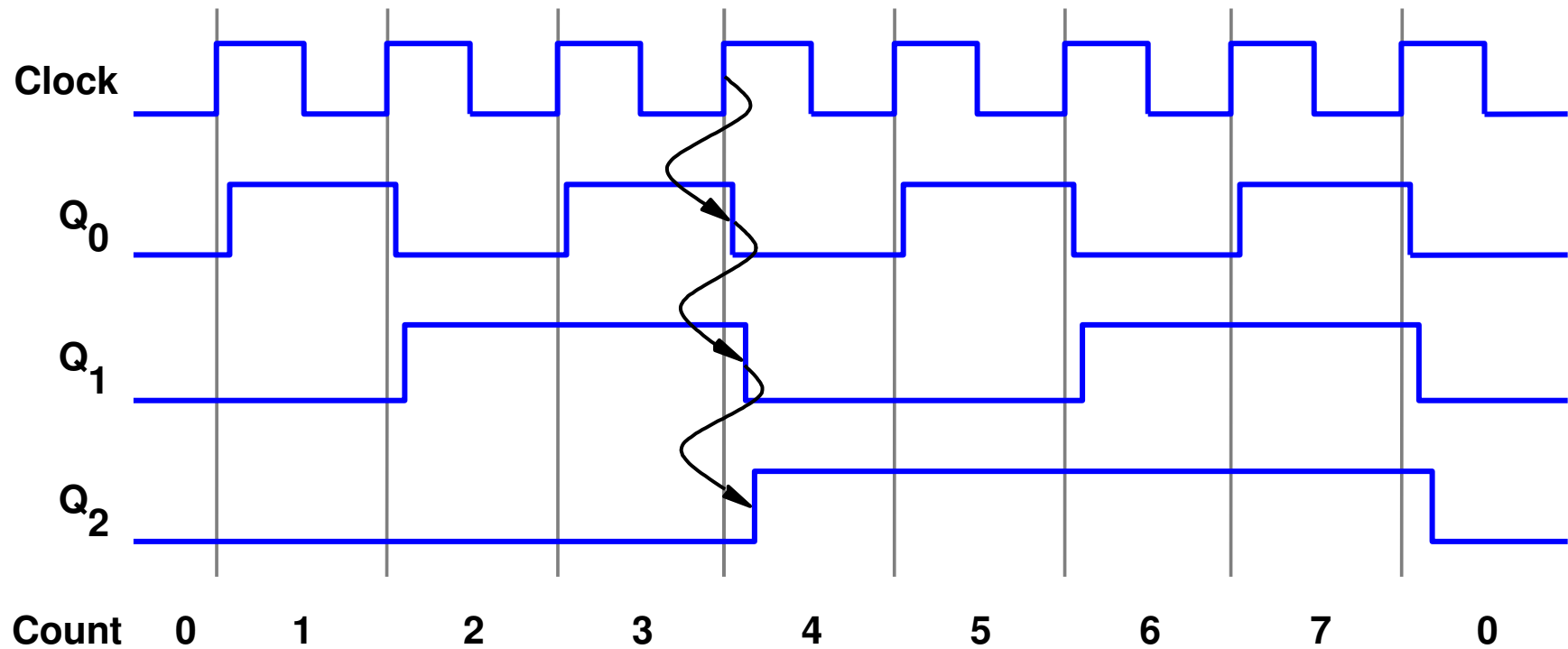
# Contadores

## Contador Binário

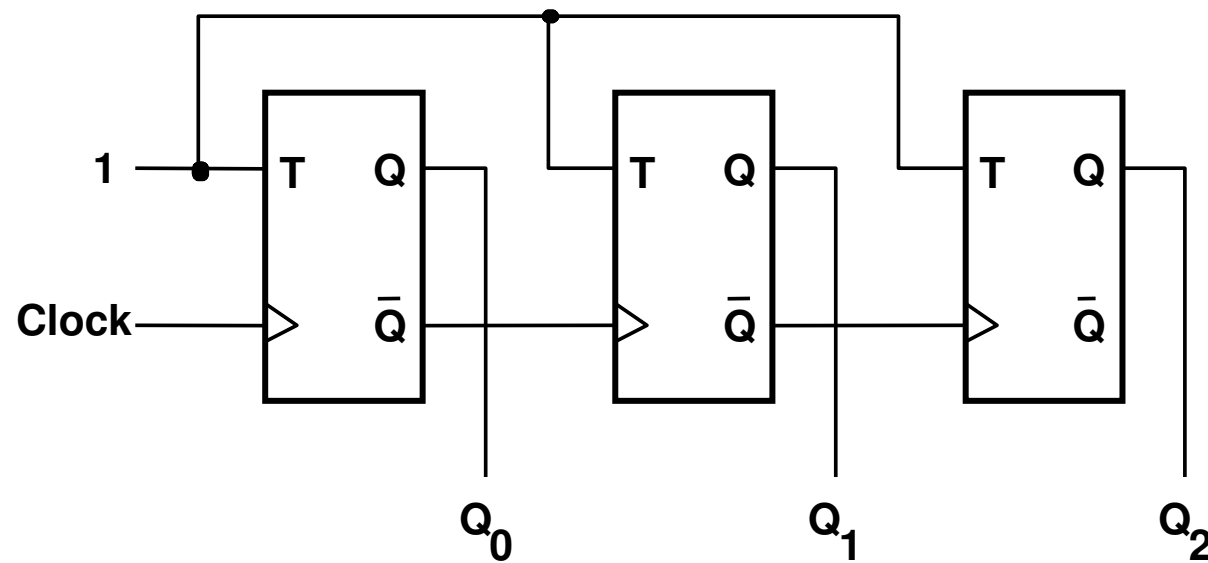
	clk	Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>
<i>t</i> <sub>0</sub>	↑	0	0	0
<i>t</i> <sub>1</sub>	↑	0	0	1
<i>t</i> <sub>2</sub>	↑	0	1	0
<i>t</i> <sub>3</sub>	↑	0	1	1
<i>t</i> <sub>4</sub>	↑	1	0	0
<i>t</i> <sub>5</sub>	↑	1	0	1
<i>t</i> <sub>6</sub>	↑	1	1	0
<i>t</i> <sub>7</sub>	↑	1	1	1



# Contador Binário - FF Tipo T

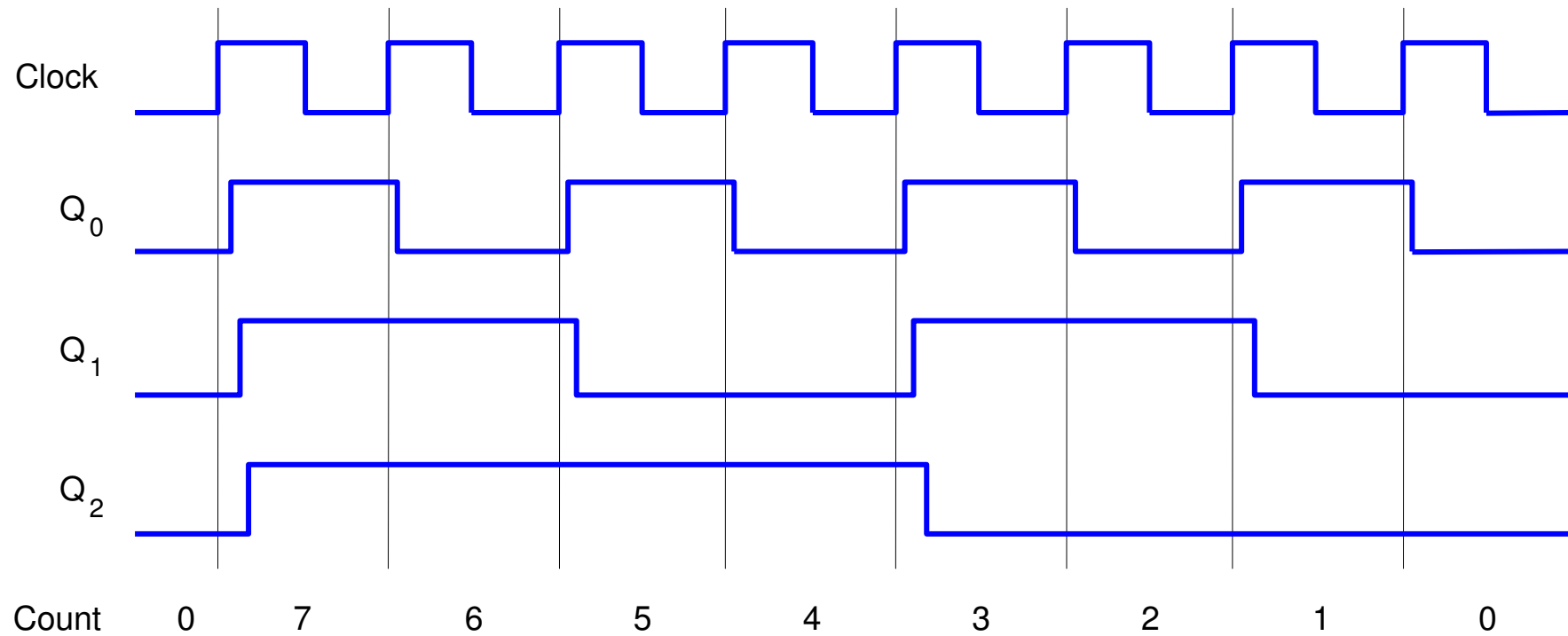


# Contador Binário - FF Tipo T

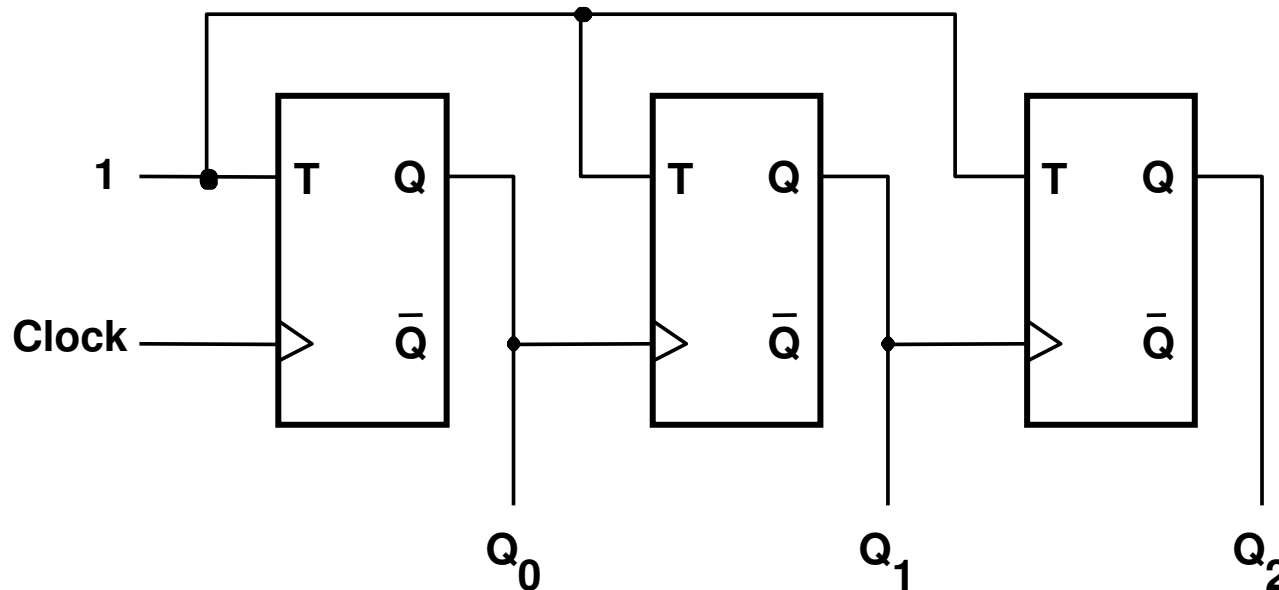


## Contador Binário Assíncrono (up-counter)

# Contador Binário - FF Tipo T (Down-Counter)

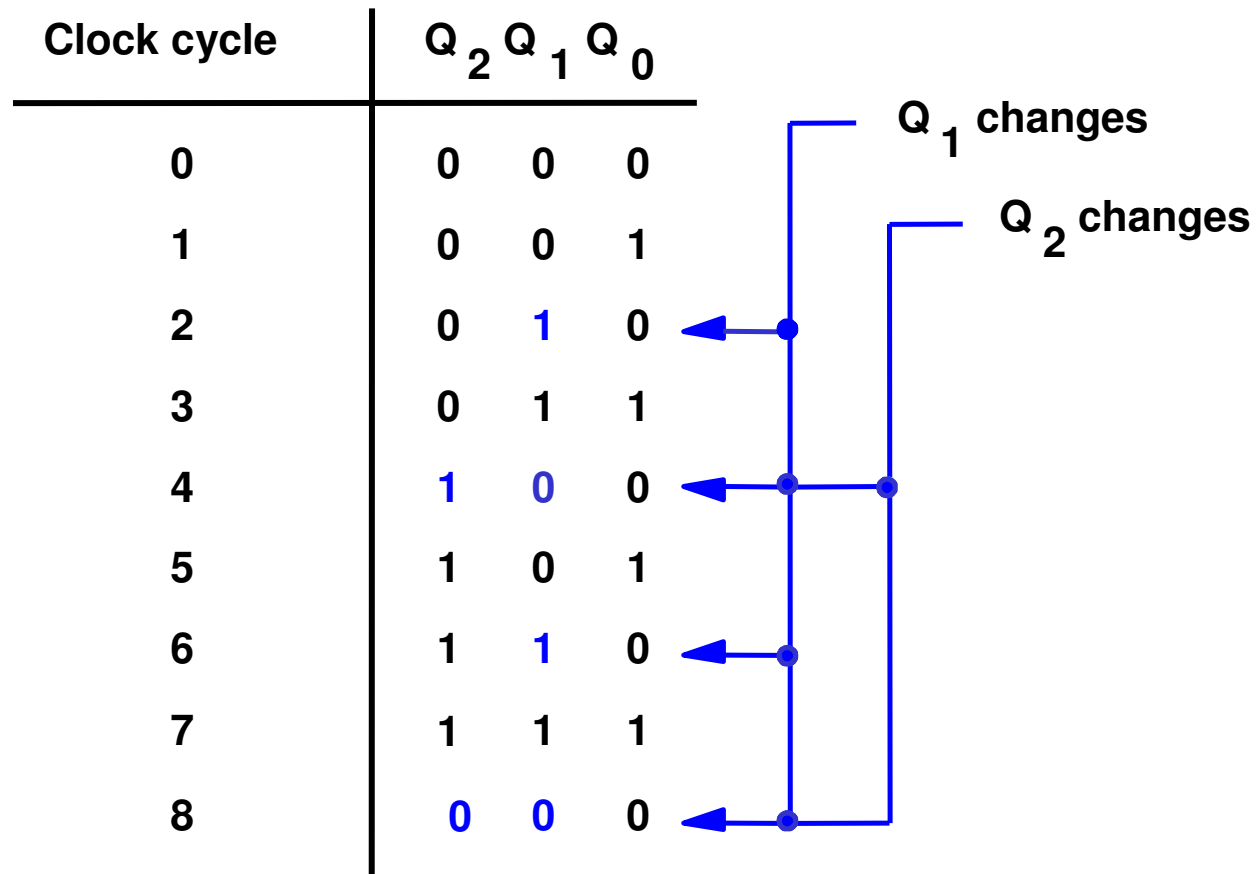


## Contador Binário - FF Tipo T (Down-Counter)

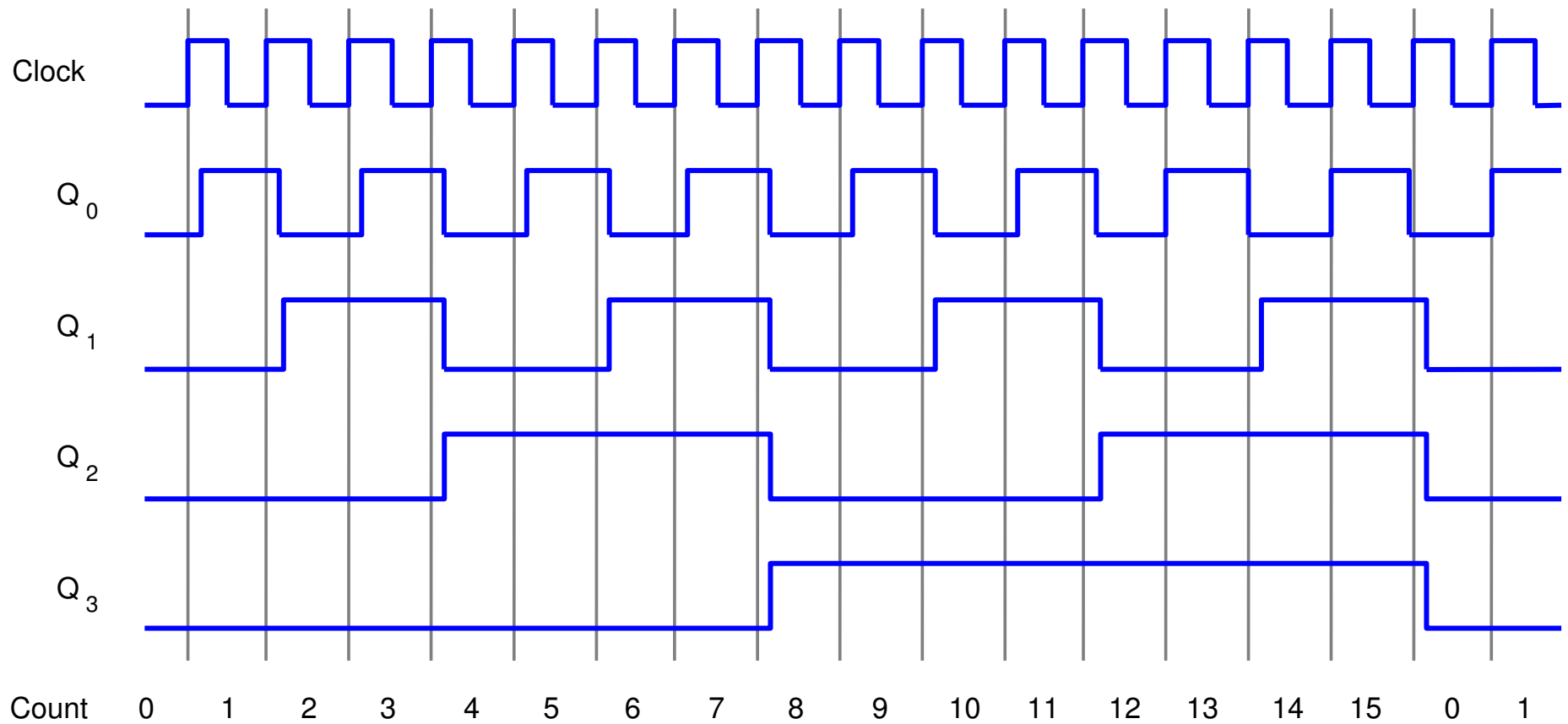


## Contador Binário Assíncrono (down-counter)

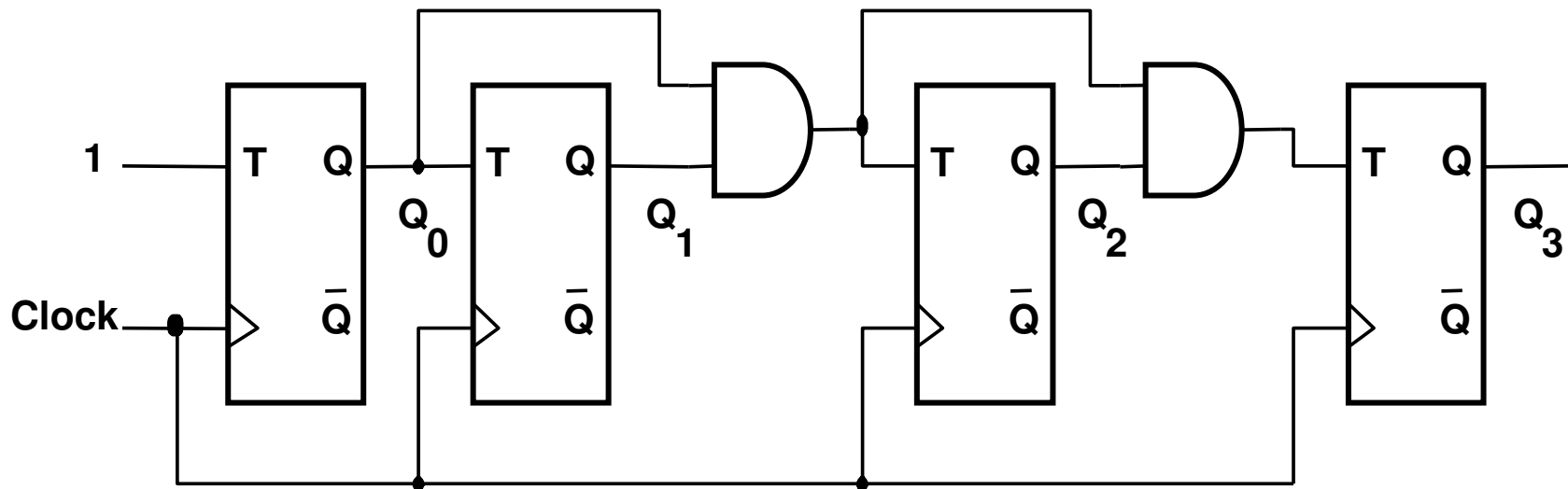
# Contadores Síncronos



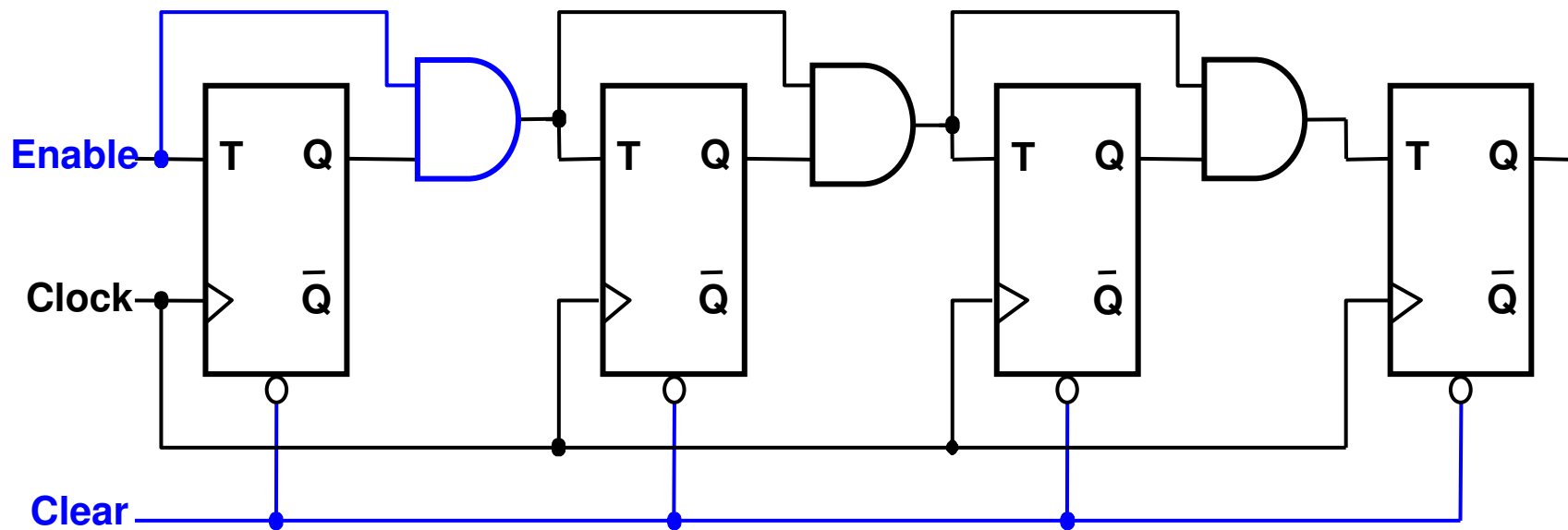
# Contador Binário Síncrono



# Contador Binário Síncrono



# Contador Binário Síncrono com Enable e Clear

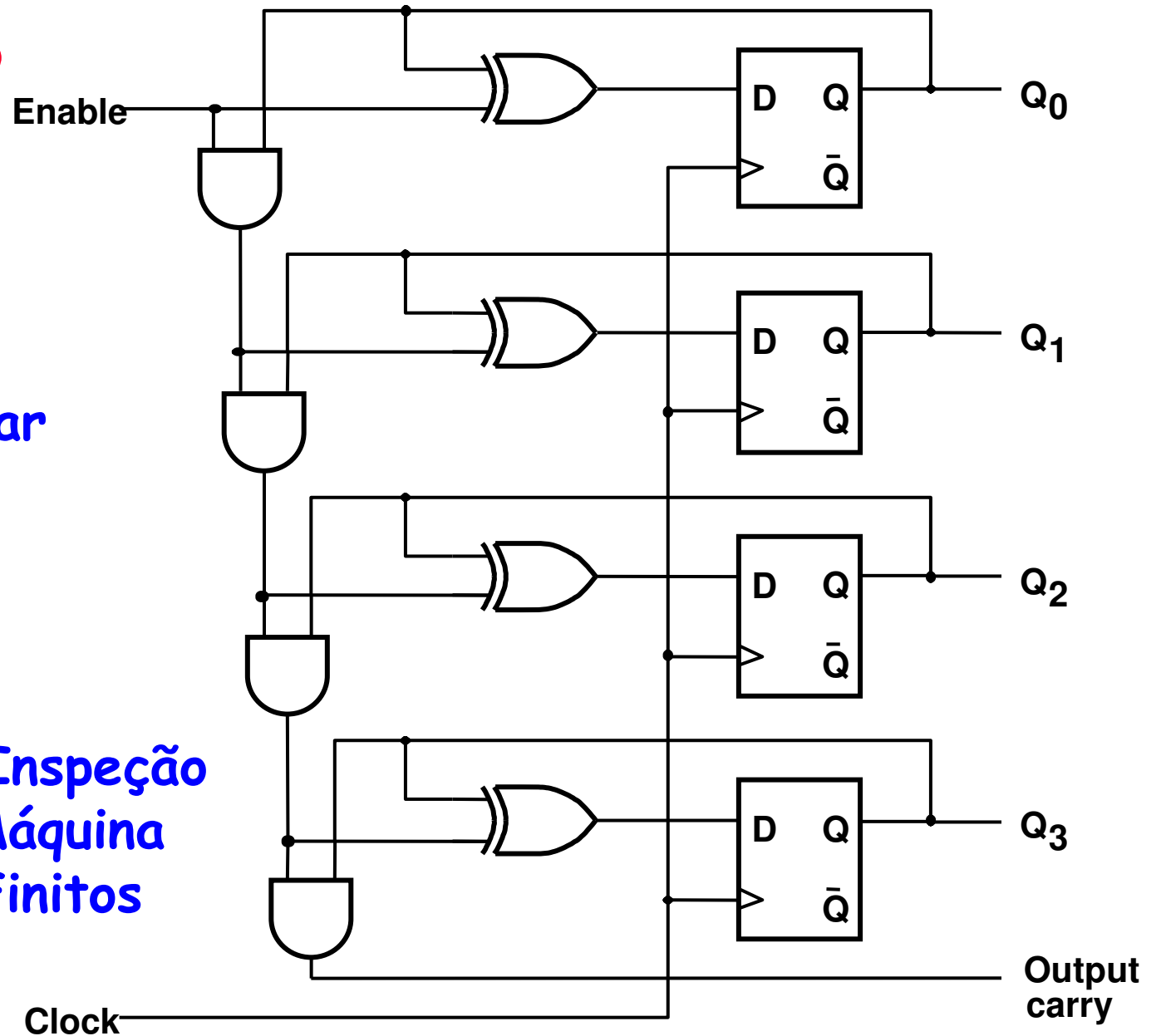




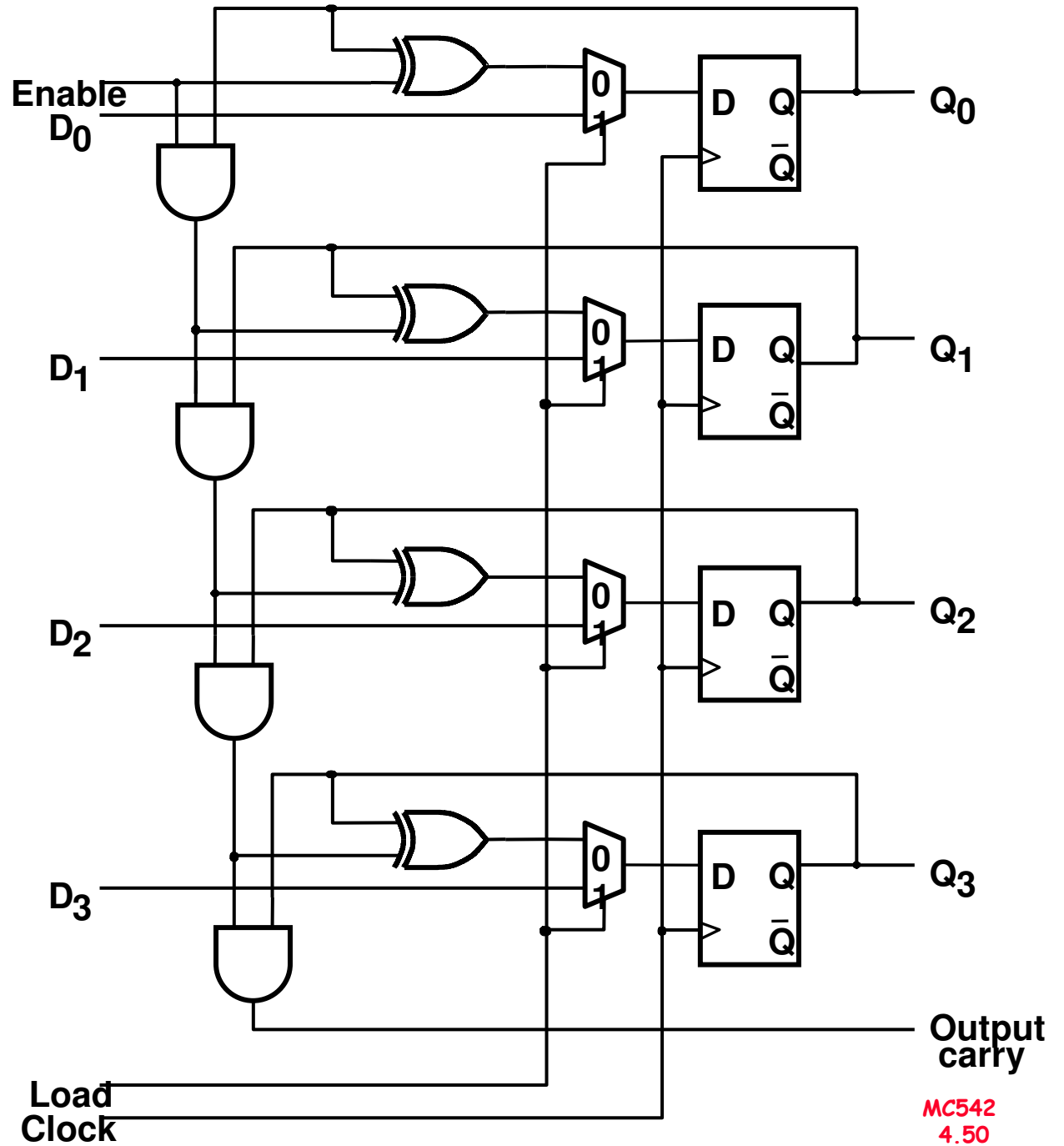
# Contador de 4 bits com FF D

Como determinar as funções de excitação de cada FF?

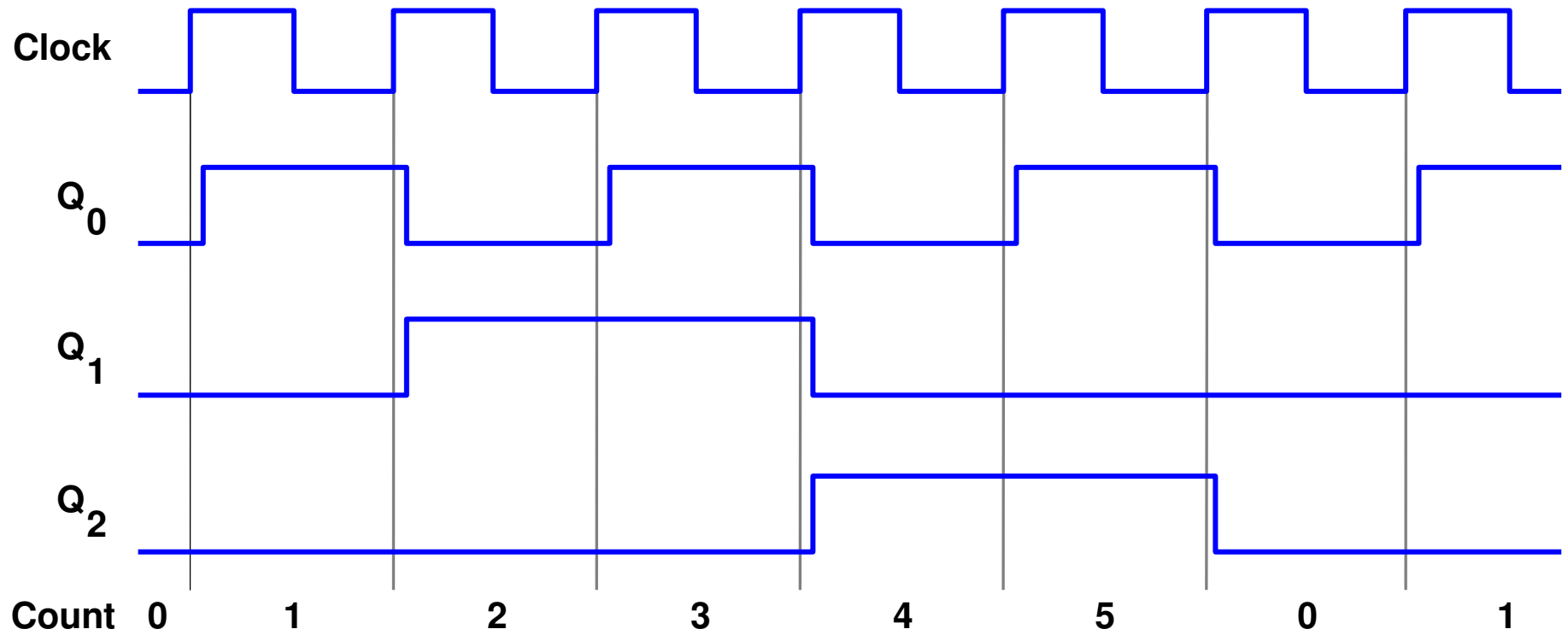
1. Projeto por Inspeção
2. Projeto de Máquina de Estados Finitos (FSM)



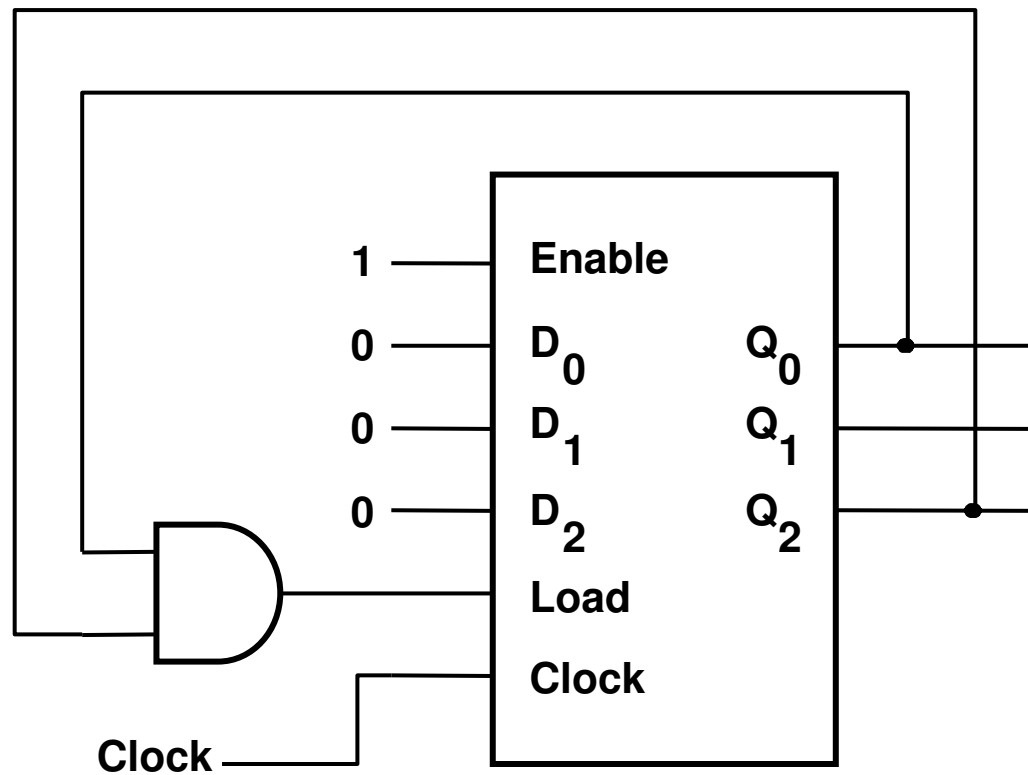
# Contador de 4 bits com FF D com Carga Paralela



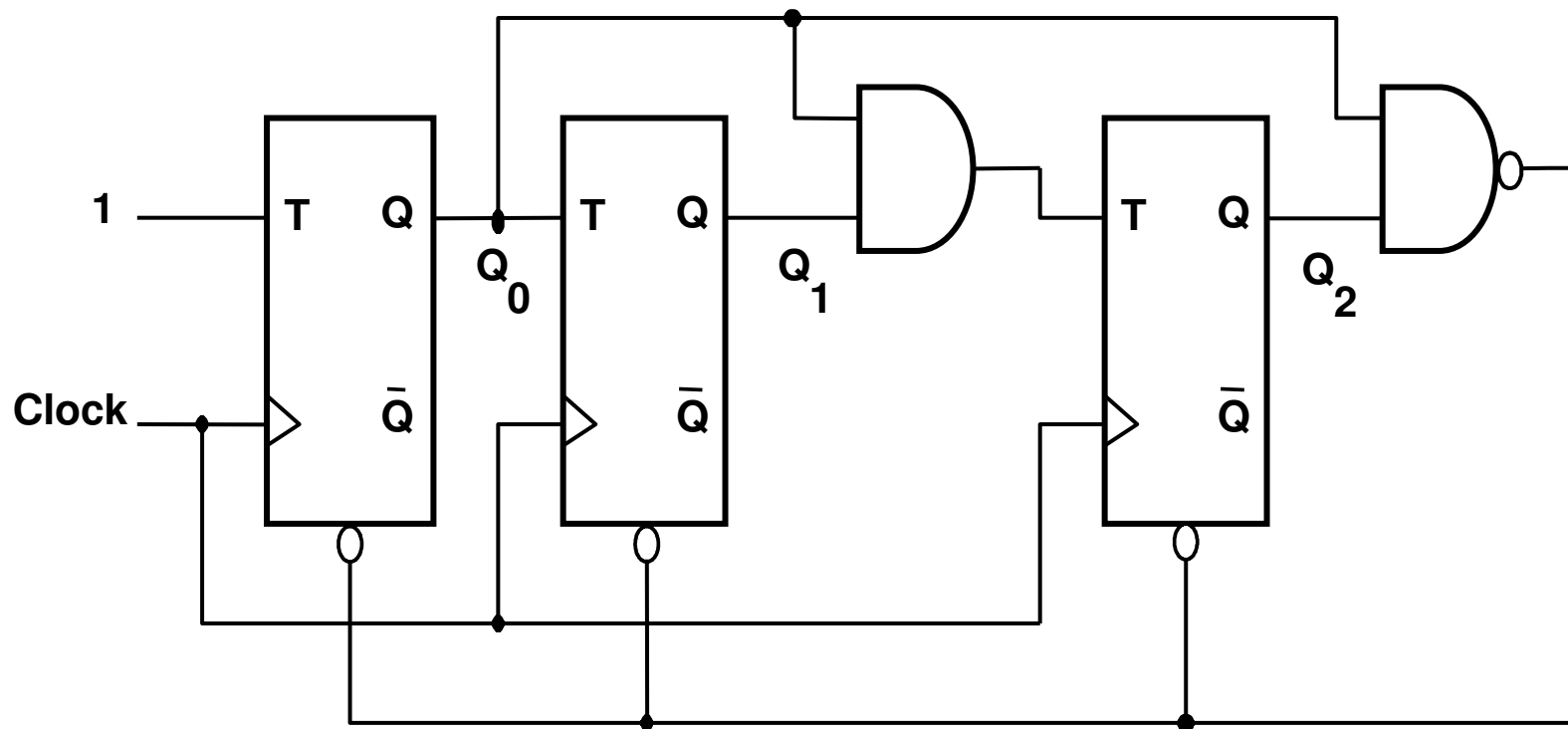
# Contador Módulo (ejemplo: Módulo 6)



# Contador Módulo (exemplo: Módulo 6)



# Contador Módulo 6 com Reset Assíncrono



# Contador Módulo 6 com Reset Assíncrono

