

# Circuitos Sequenciais

GEN 253 - Circuitos Digitais

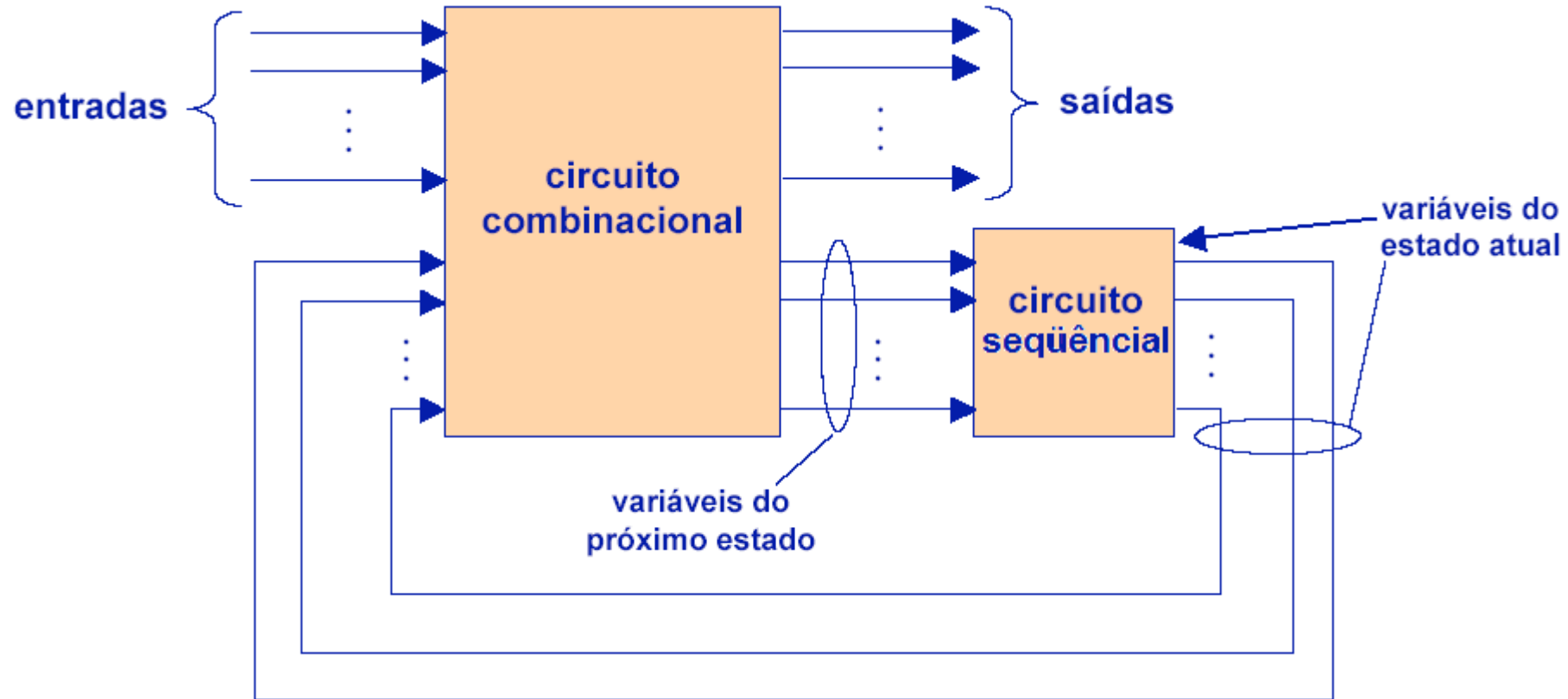
Prof. Luciano L. Caimi  
[lcaimi@uffs.edu.br](mailto:lcaimi@uffs.edu.br)

# Circuitos Digitais

Como discutido anteriormente os circuitos lógicos digitais são divididos em duas grandes áreas:

- **Circuitos Combinacionais:** em que as saídas do circuito dependem exclusivamente do valor presente na entrada. Como nos multiplexadores, somadores, codificadores, etc...
- **Circuitos Sequenciais:** onde as saídas do circuito dependem dos valores presentes nas entradas e do estado anterior em que o circuito se encontra. Como exemplo temos os registradores, os contadores, as máquinas de estado, dentre outros.

# Circuitos Sequenciais



# Circuitos Sequenciais

## Elementos básicos: Latches e Flip-Flops

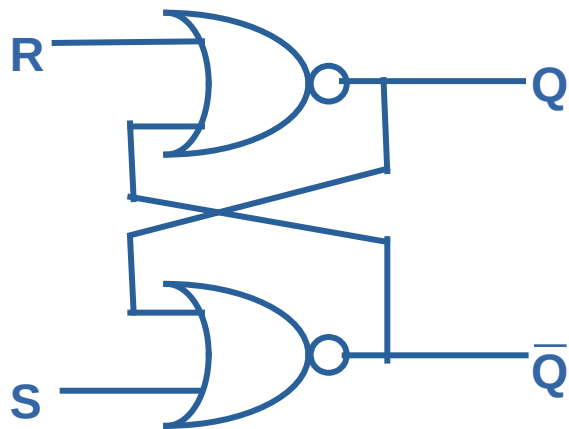
- **Latches:** são elementos **assíncronos**, ou seja, operam sem restrições temporais
- **Flip-Flops** são elementos **síncronos**, isto é, operam com restrições de tempo

O funcionamento lógico de Latches e Flip-Flops são semelhantes, sendo as diferenças baseadas na existência ou não de restrições temporais

Por definição tanto os latches como os flip-flops possuem duas saídas com valores opostos ( $Q = 1$ ,  $Q' = 0$  ou  $Q = 0$  e  $Q' = 1$ )

# Circuitos Sequenciais

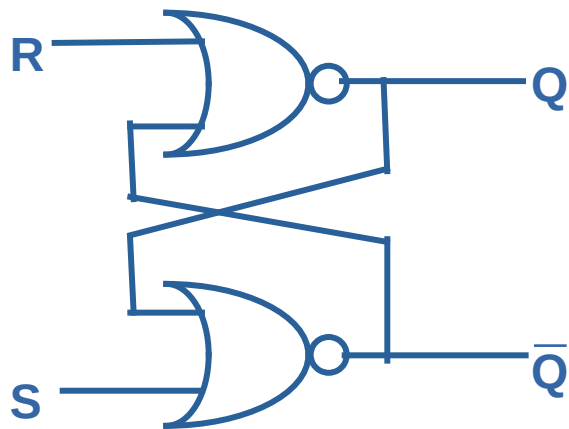
## Latch RS



R	S	$Q_t$	$Q_{t+1}$	Comentário
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

# Circuitos Sequenciais

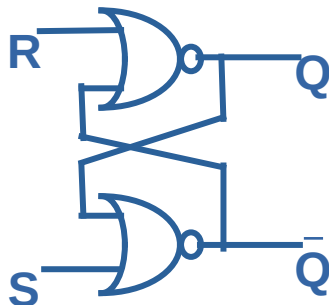
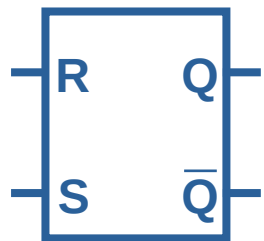
## Latch RS



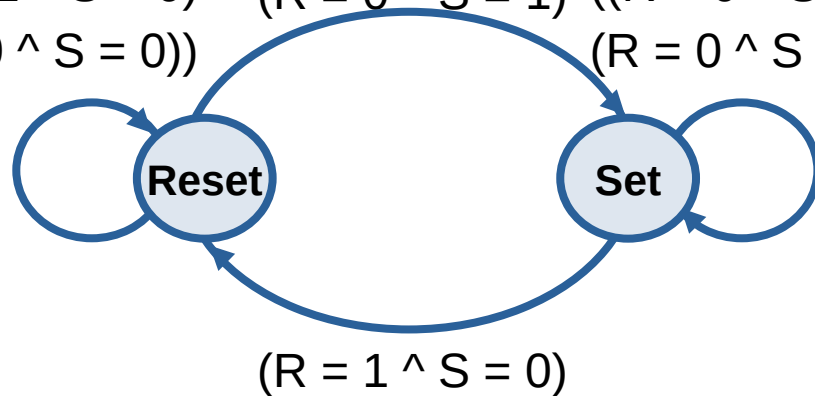
R	S	$Q_t$	$Q_{t+1}$	Comentário
0	0	0	0	Igual $Q_t$
0	0	1	1	Igual $Q_t$
0	1	0	1	Set
0	1	1	1	Set
1	0	0	0	Reset
1	0	1	0	Reset
1	1	0	-	Inválido
1	1	1	-	Inválido

# Circuitos Sequenciais

## Latch RS



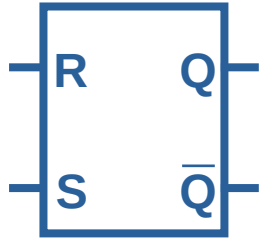
$$\begin{array}{lll} ((R = 1 \wedge S = 0) \vee (R = 0 \wedge S = 1) \vee ((R = 0 \wedge S = 1) \vee (R = 0 \wedge S = 0))) \\ ((R = 0 \wedge S = 0)) \end{array}$$



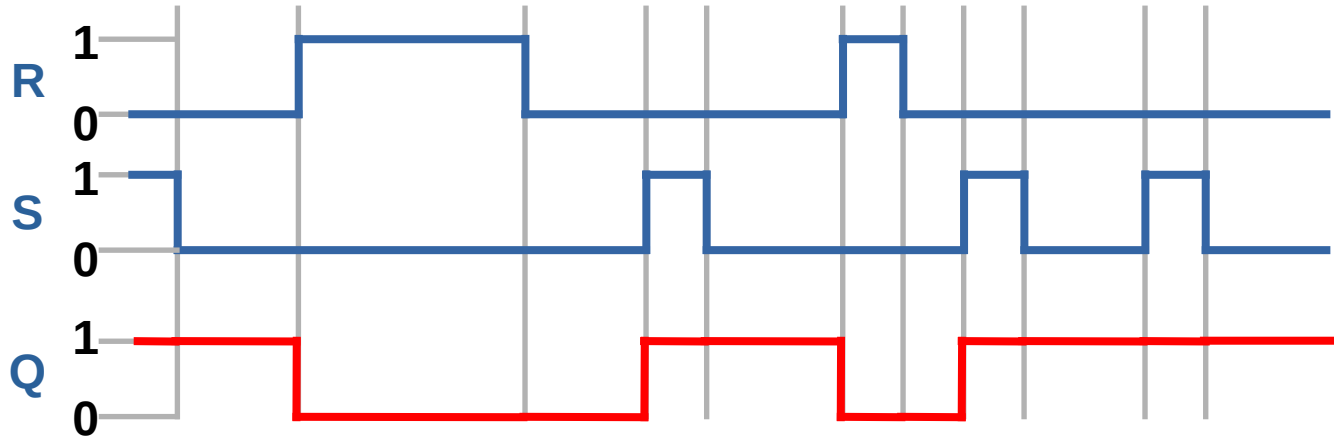
R	S	$Q_{t+1}$	Comentário
0	0	$Q_t$	mantém anterior
0	1	1	estado set
1	0	0	estado reset
1	1	-	estado inválido

# Circuitos Sequenciais

## Latch RS



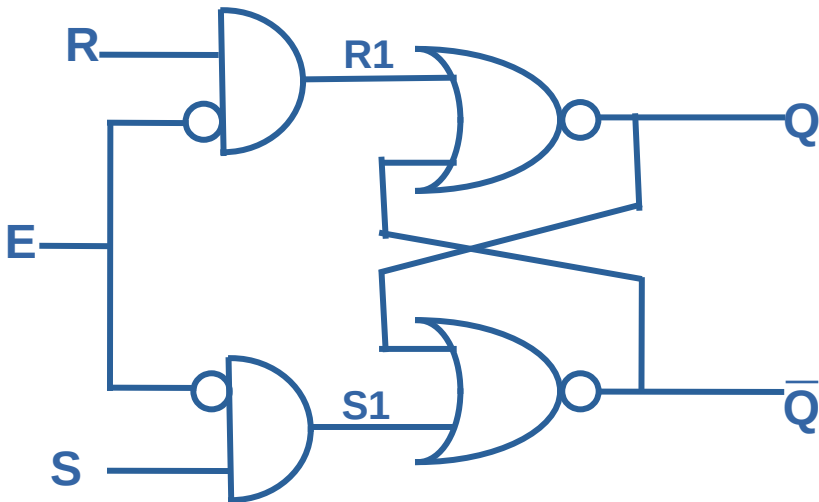
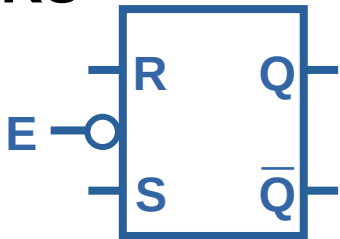
R	S	$Q_{t+1}$	Comentário
0	0	$Q_t$	mantém anterior
0	1	1	estado set
1	0	0	estado reset
1	1	-	estado inválido





# Circuitos Sequenciais

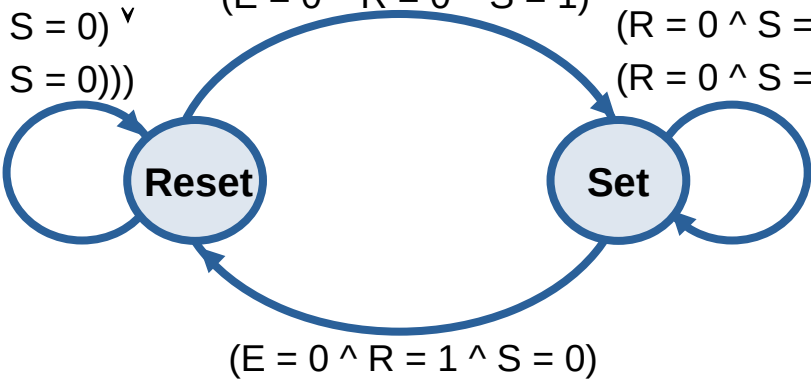
## Latch RS



E	R	S	$Q_{t+1}$	Comentário
1	X	X	$Q_t$	mantém anterior
0	0	0	$Q_t$	mantém anterior
0	0	1	1	estado set
0	1	0	0	estado reset
0	1	1	-	estado inválido

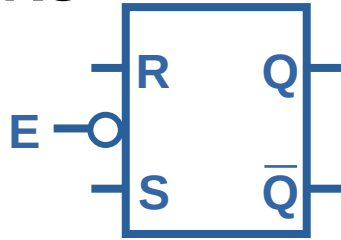
$((E = 1) \vee ((E = 0) \wedge (R = 1 \wedge S = 0) \vee (R = 0 \wedge S = 0)))$

$((E = 1) \vee ((E = 0) \wedge (E = 0 \wedge R = 0 \wedge S = 1) \vee (R = 0 \wedge S = 1) \vee (R = 0 \wedge S = 0)))$

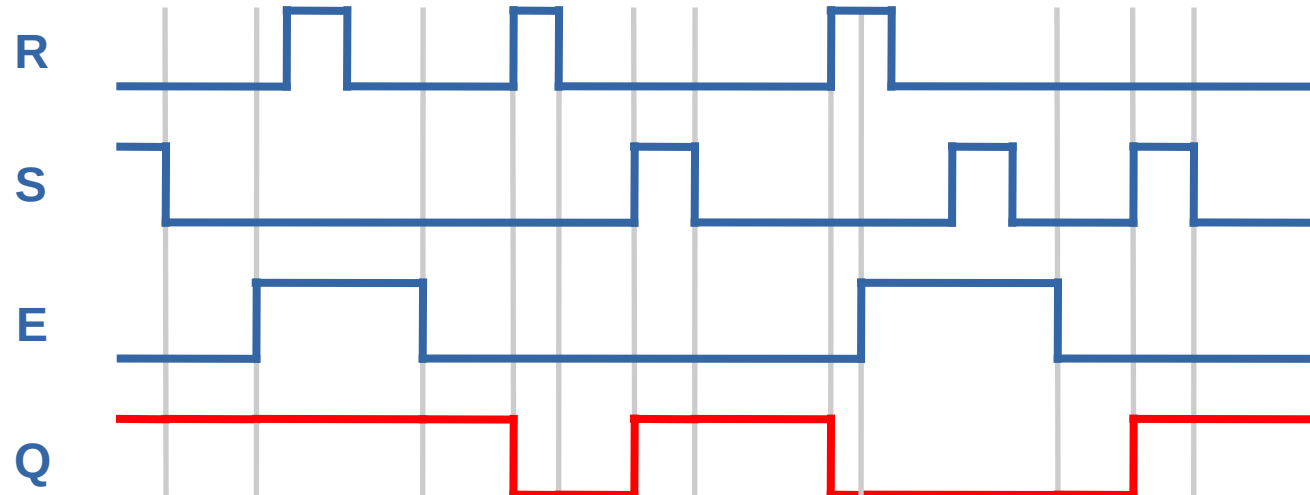


# Circuitos Sequenciais

## Latch RS

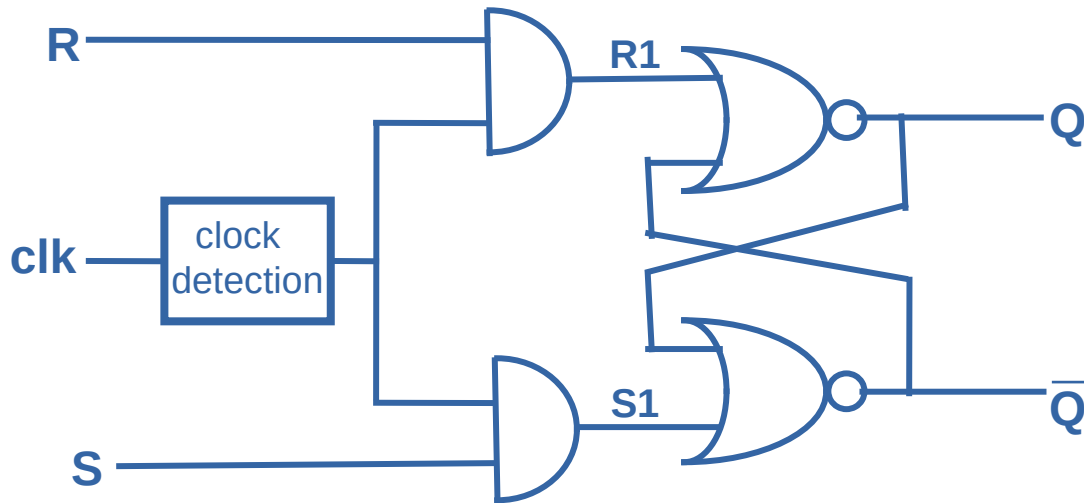
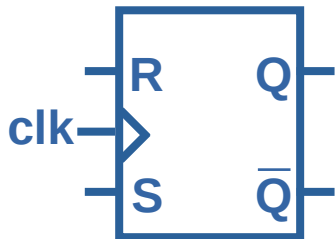


E	R	S	$Q_{t+1}$	Comentário
1	X	X	$Q_t$	mantém anterior
0	0	0	$Q_t$	mantém anterior
0	0	1	1	estado set
0	1	0	0	estado reset
0	1	1	-	estado inválido



# Circuitos Sequenciais

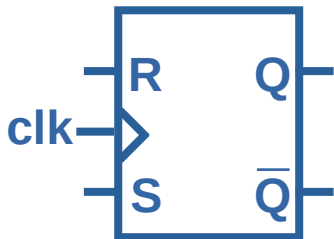
## Flip-Flop RS



Como fica a tabela-verdade?

# Circuitos Sequenciais

## Flip-Flop RS

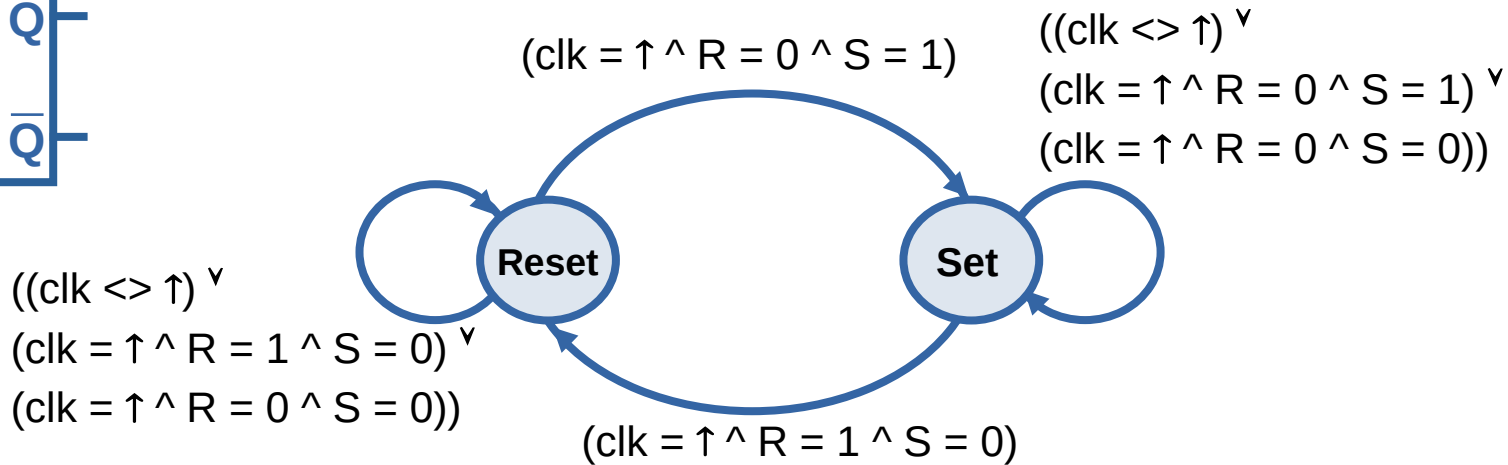
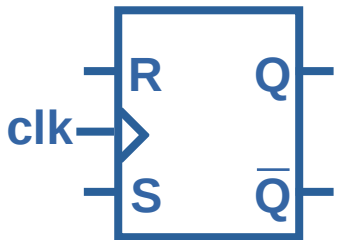


clk	R	S	$Q_{t+1}$	Comentário
1	X	X	$Q_t$	mantém anterior
0	X	X	$Q_t$	mantém anterior
↓	X	X	$Q_t$	mantém anterior
↑	0	0	$Q_t$	mantém anterior
↑	0	1	1	estado set
↑	1	0	0	estado reset
↑	1	1	-	estado inválido

Como fica o diagrama de estados?

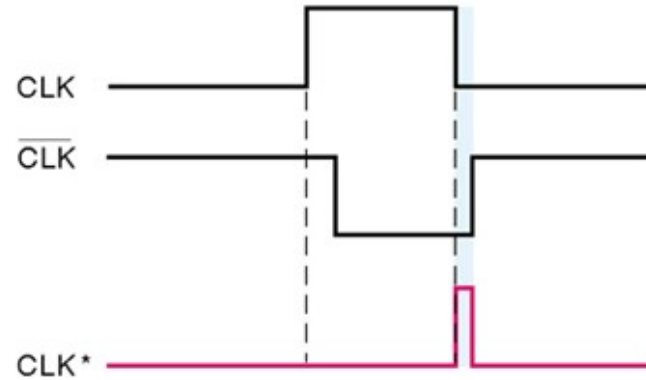
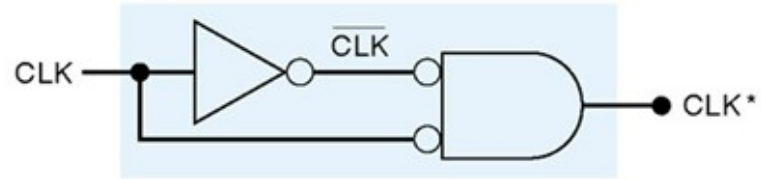
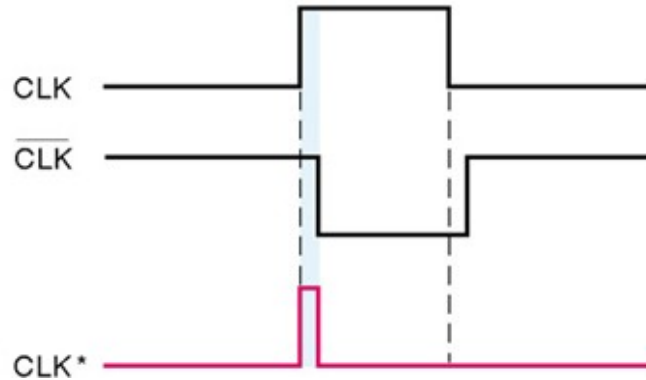
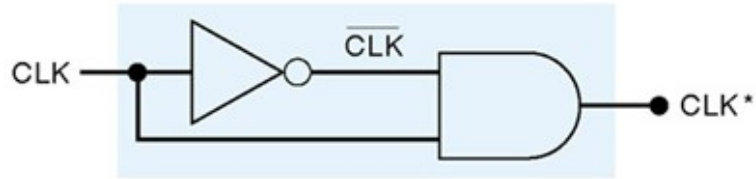
# Circuitos Sequenciais

## Flip-Flop RS



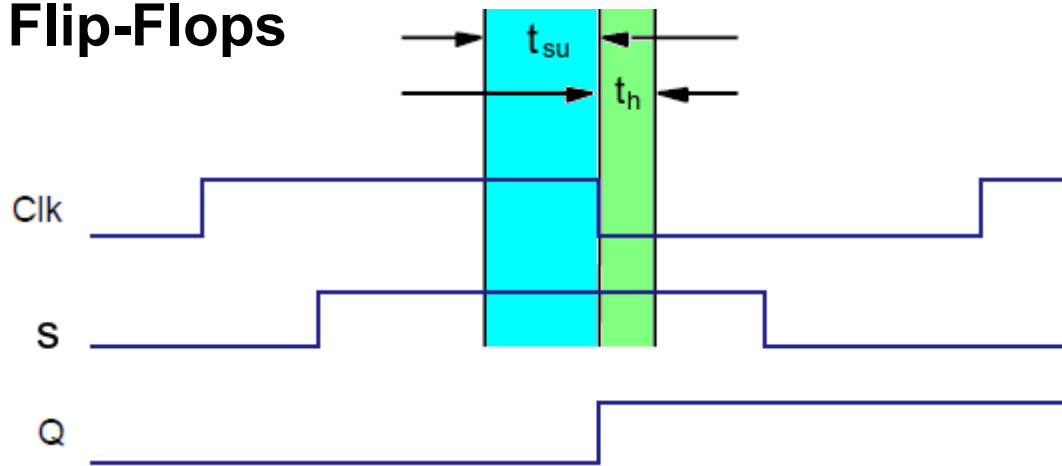
# Circuitos Sequenciais

## Detecção de borda



# Circuitos Sequenciais

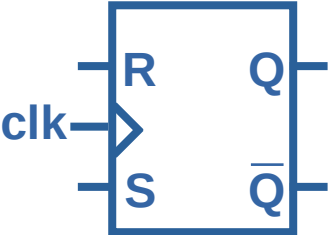
## Temporização de Flip-Flops



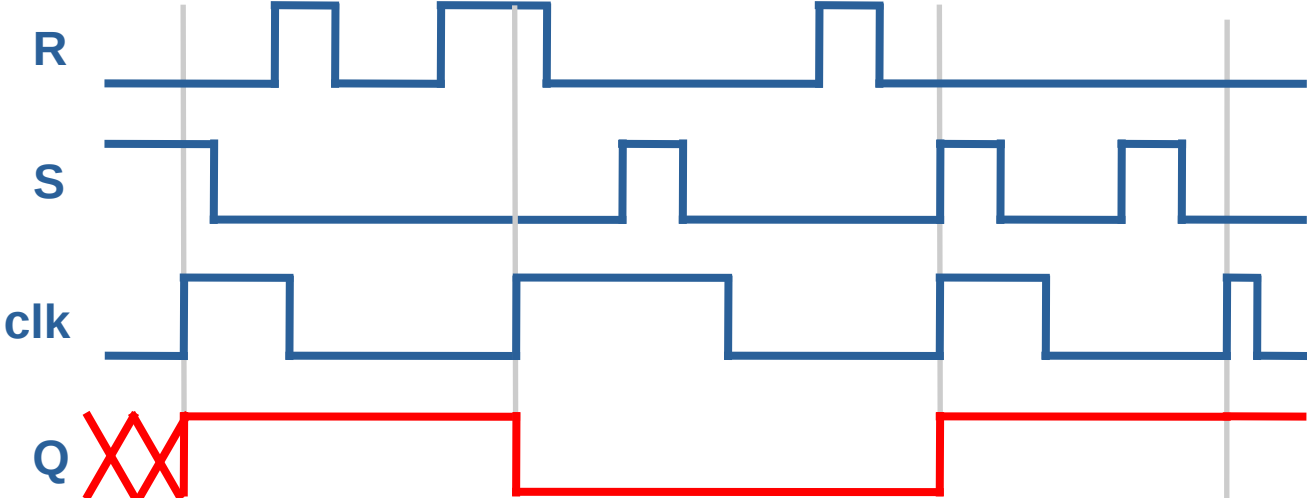
- Os tempos de **setup** e **hold** são parâmetros que devem ser observados para que o flip-flop possa trabalhar de modo confiável
- O **tempo de setup ( $t_{su}$ )**, corresponde ao intervalo no qual as entradas devem permanecer estáveis antes da transição do clock
- O **tempo de hold ( $t_h$ )**, corresponde ao intervalo no qual as entradas devem permanecer estáveis depois da transição do clock

# Circuitos Sequenciais

## Flip-Flop RS



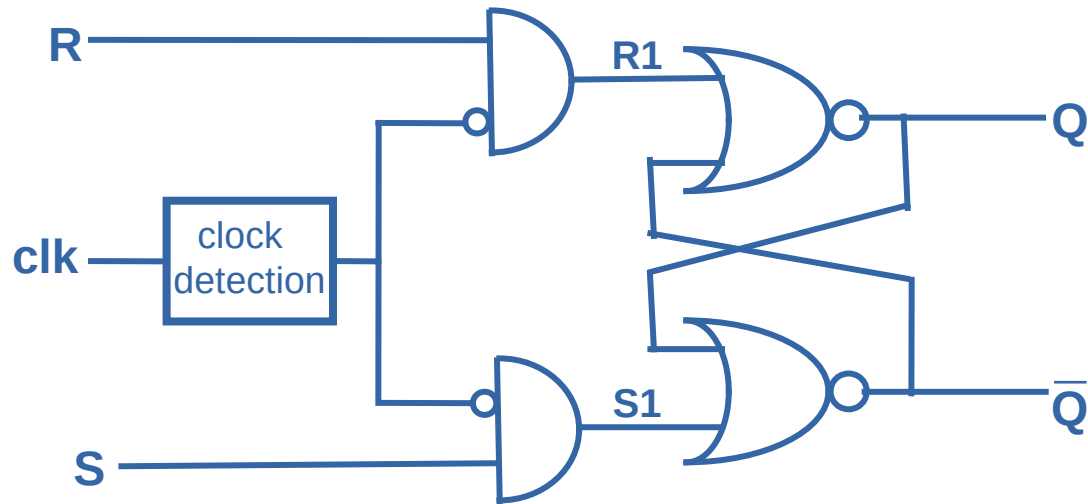
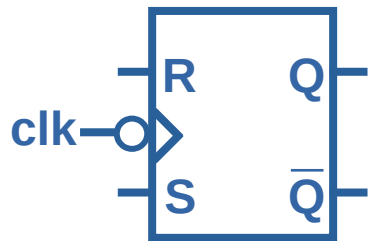
clk	R	S	$Q_{t+1}$	Comentário
1	X	X	$Q_t$	mantém anterior
0	X	X	$Q_t$	mantém anterior
↓	X	X	$Q_t$	mantém anterior
↑	0	0	$Q_t$	mantém anterior
↑	0	1	1	estado set
↑	1	0	0	estado reset
↑	1	1	-	estado inválido





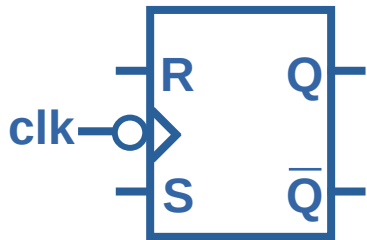
# Circuitos Sequenciais

## Flip-Flop RS gatilhado na descida do clock



# Circuitos Sequenciais

## Flip-Flop RS gatilhado na descida do clock

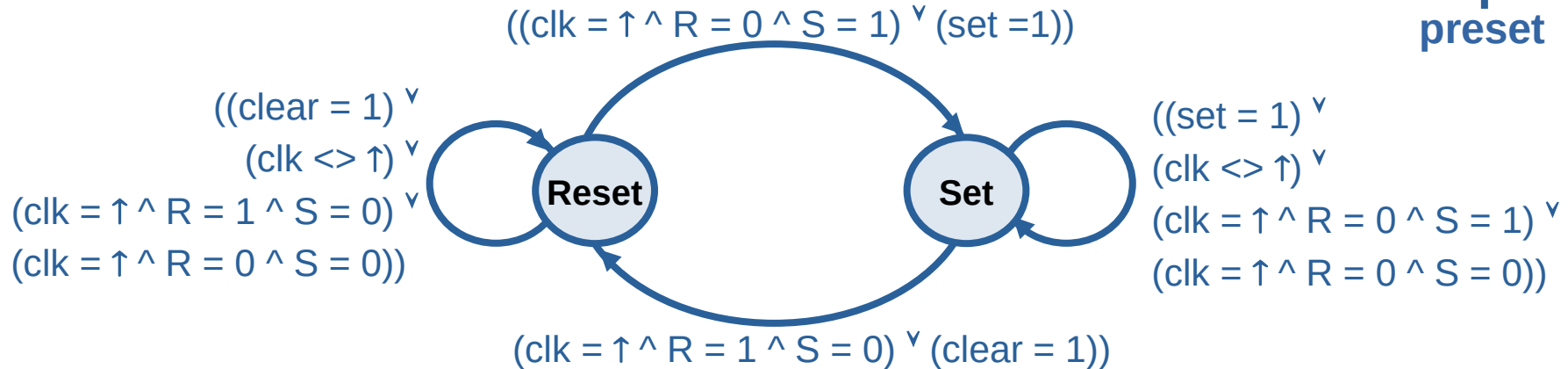
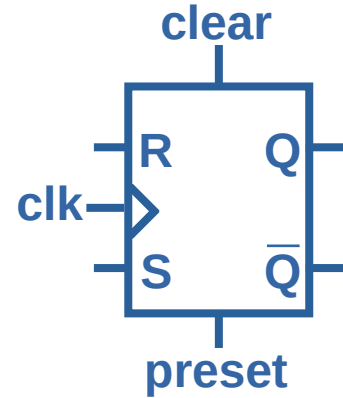


clk	R	S	$Q_{t+1}$	Comentário
1	X	X	$Q_t$	mantém anterior
0	X	X	$Q_t$	mantém anterior
↑	X	X	$Q_t$	mantém anterior
↓	0	0	$Q_t$	mantém anterior
↓	0	1	1	estado set
↓	1	0	0	estado reset
↓	1	1	-	estado inválido

# Circuitos Sequenciais

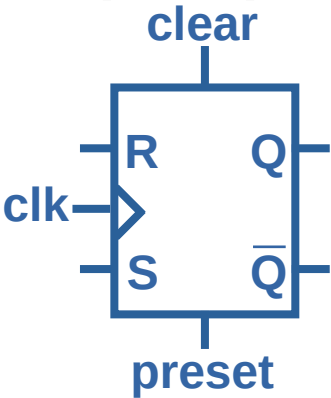
## Flip-Flop RS com entradas assíncronas de *preset* e *clear*

- Nesta configuração duas entradas assíncronas permitem colocar o flip-flop com saída  $Q = 0$  (clear) ou  $Q = 1$  (preset) independente da transição de clock, permitindo, por exemplo a inicialização do FF

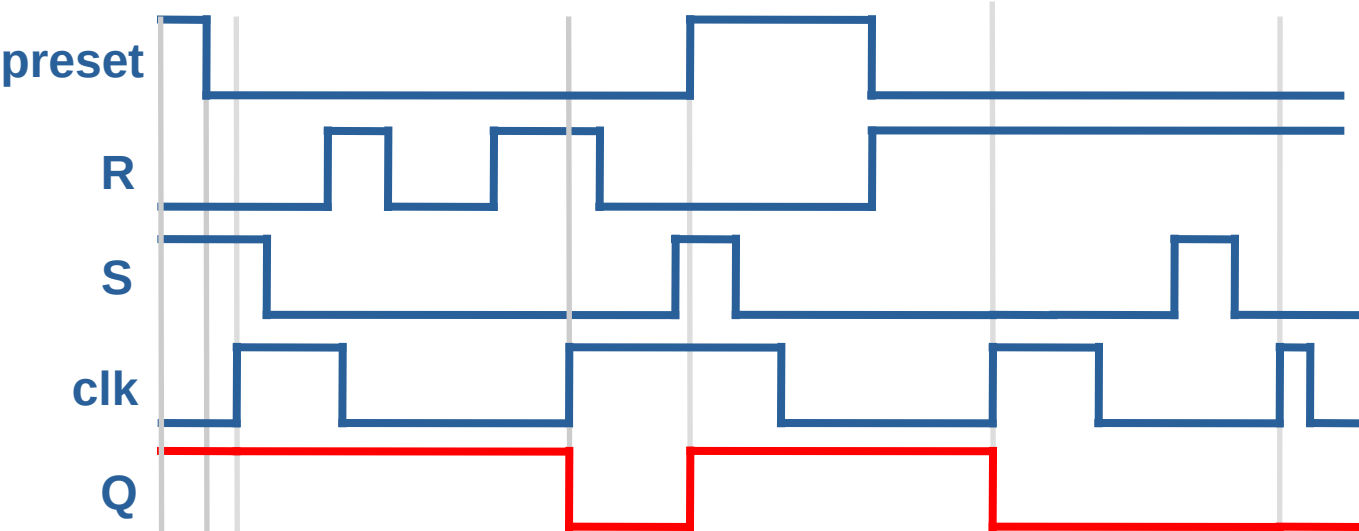


# Circuitos Sequenciais

## Flip-Flop RS com *preset* e *clear*



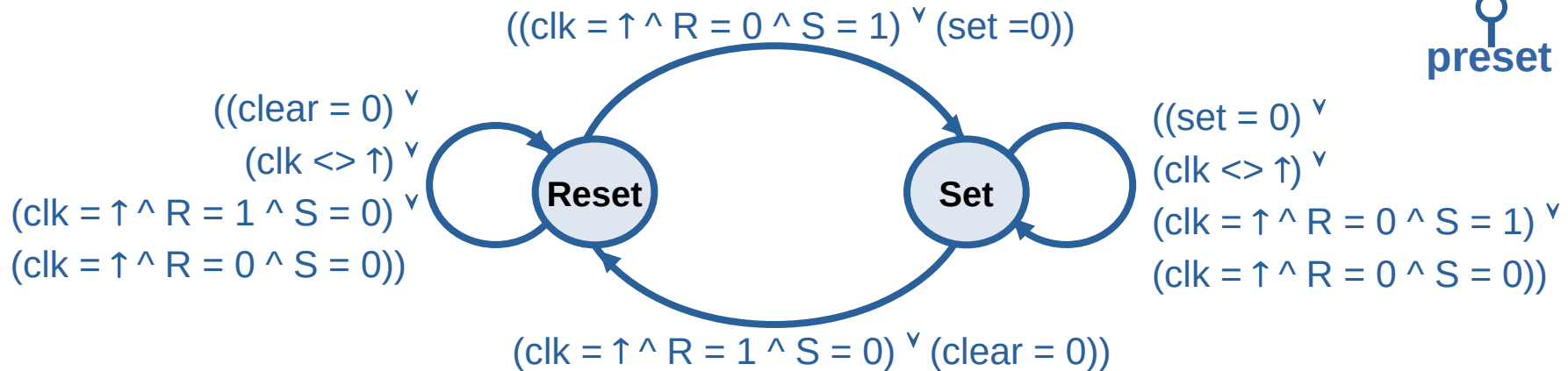
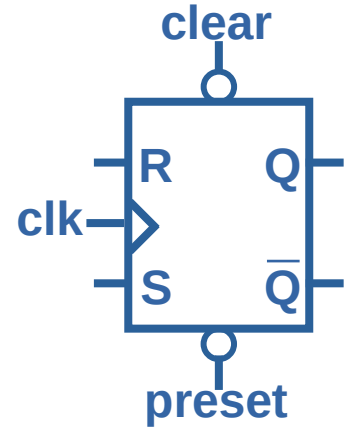
clear	preset	clk	R	S	$Q_{t+1}$	Comentário
0	1	X	X	X	1	estado set
1	0	X	X	X	0	estado reset
0	0	↑	0	0	$Q_t$	mantém anterior
0	0	↑	0	1	1	estado set
0	0	↑	1	0	0	estado reset
0	0	↑	1	1	-	estado inválido



# Circuitos Sequenciais

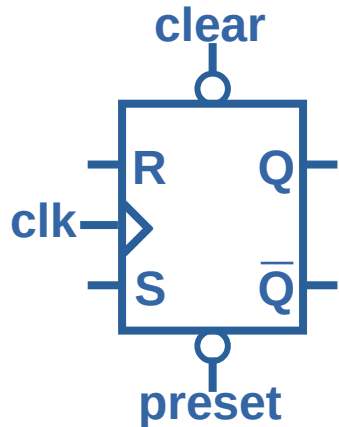
## Flip-Flop RS com entradas assíncronas de *preset* e *clear* complementar

- Comportamento igual ao circuito anterior mas utilizando a lógica complementar nas entradas *preset* e *clear*, ou seja, ativa em nível lógico zero

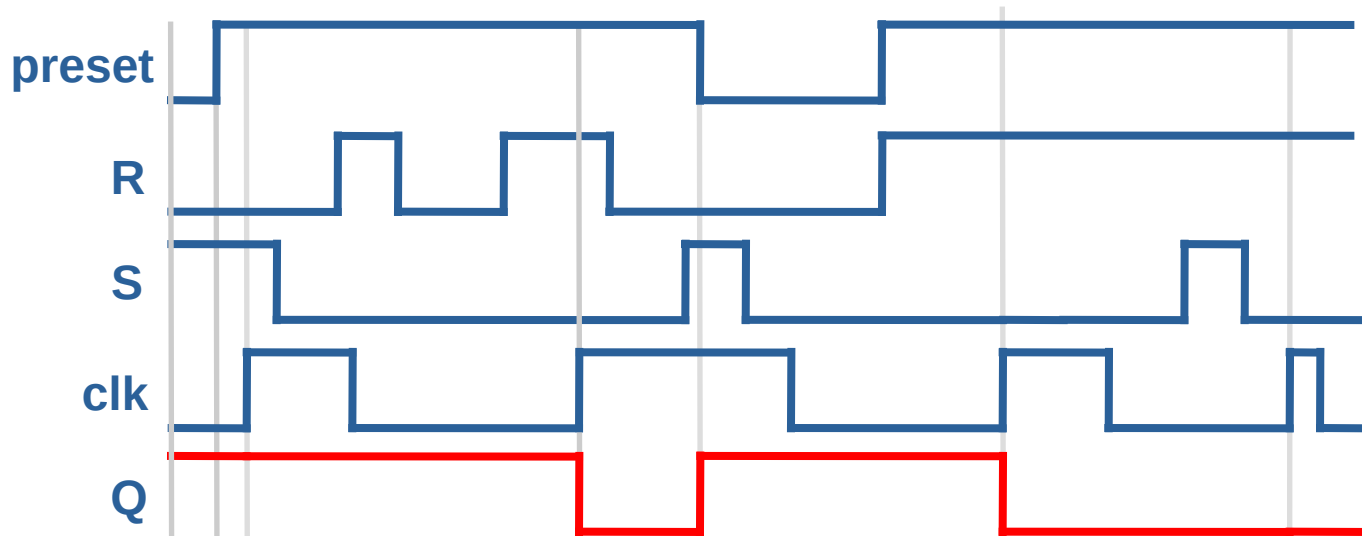


# Circuitos Sequenciais

## Flip-Flop RS com *preset* e *clear* complementar



clear	preset	clk	R	S	$Q_{t+1}$	Comentário
1	0	X	X	X	1	estado set
0	1	X	X	X	0	estado reset
1	1	↑	0	0	$Q_t$	mantém anterior
1	1	↑	0	1	1	estado set
1	1	↑	1	0	0	estado reset
1	1	↑	1	1	-	estado inválido

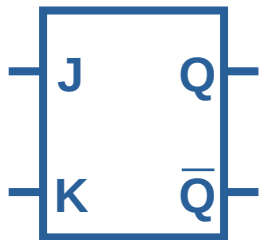


# Circuitos Sequenciais

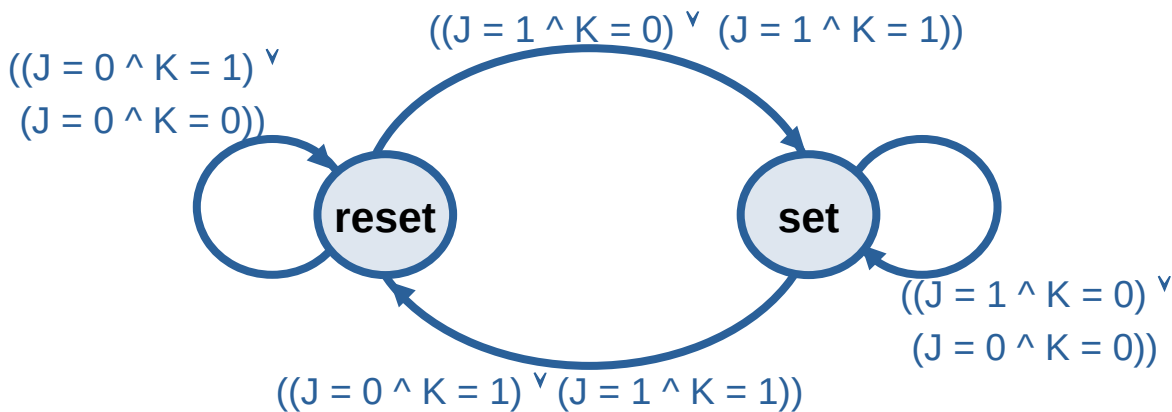
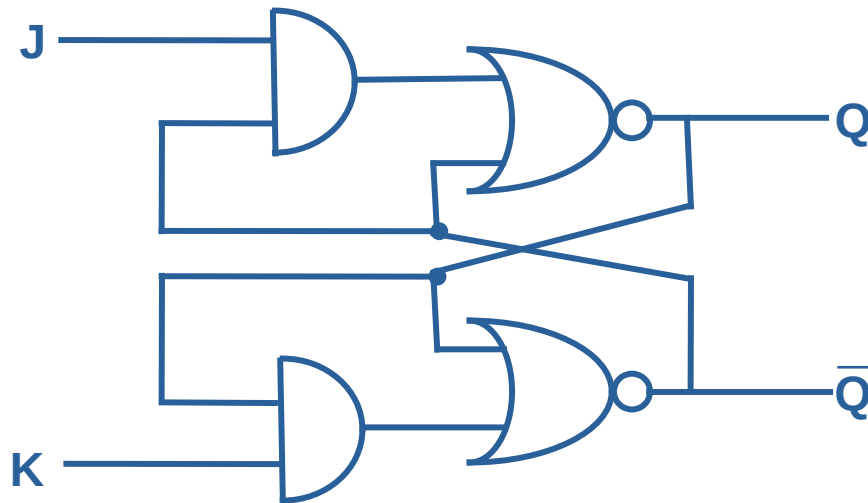
- Além das Latches e Flip-Flops tipo RS existem outros tipos elementares:
  - JK
  - D
  - T
- Embora ocorra variações na funcionalidade de cada tipo, as variações quanto ao enable normal e complementar, transição de gatilho do clock, preset e clear normal e complementar, estão presentes em todos os tipos

# Circuitos Sequenciais

## Latch JK



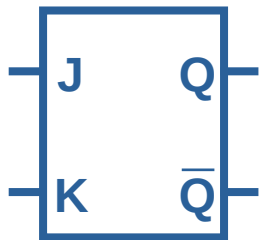
J	K	$Q_{t+1}$	Comentário
0	0	$Q_t$	mantém anterior
0	1	0	estado reset
1	0	1	estado set
1	1	$\overline{Q_t}$	inverte anterior



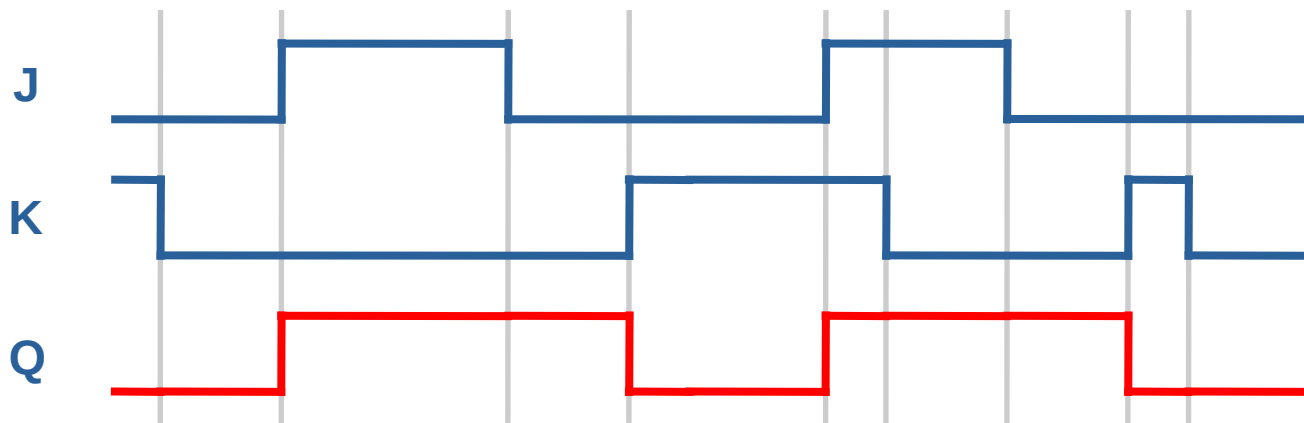


# Circuitos Sequenciais

## Latch JK

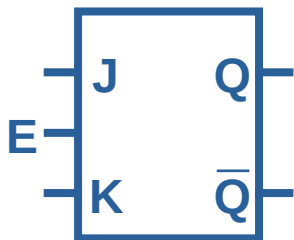


J	K	$Q_{t+1}$	Comentário
0	0	$Q_t$	mantém anterior
0	1	0	estado reset
1	0	1	estado set
1	1	$\overline{Q_t}$	inverte anterior

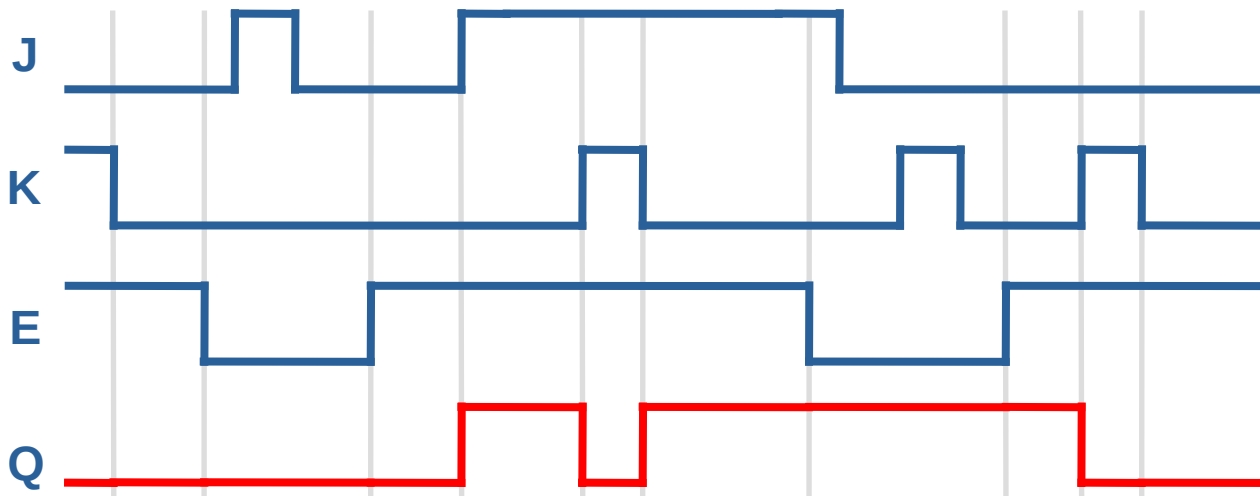


# Circuitos Sequenciais

## Latch JK com Enable

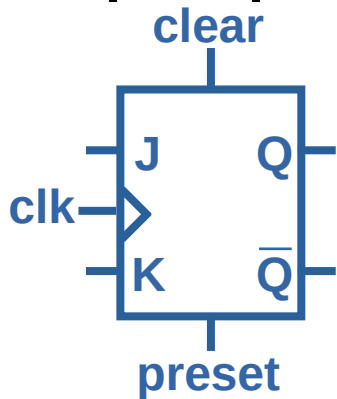


E	J	K	$Q_{t+1}$	Comentário
0	X	X	$Q_t$	mantém anterior
1	0	0	$Q_t$	mantém anterior
1	0	1	0	estado reset
1	1	0	1	estado set
1	1	1	$\bar{Q}_t$	inverte anterior

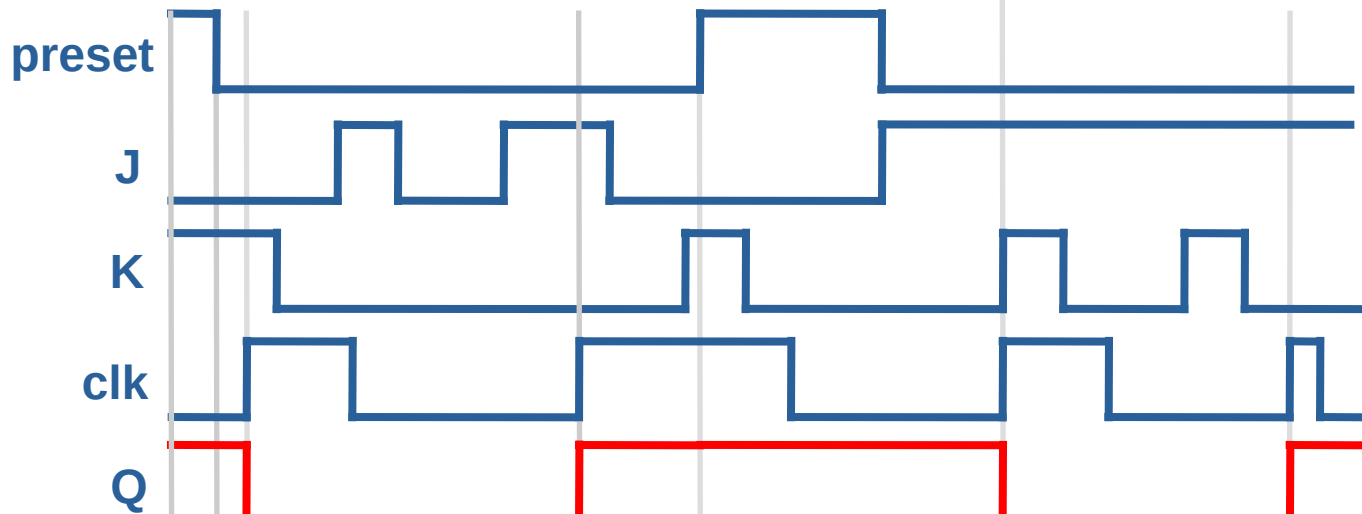


# Circuitos Sequenciais

## Flip-Flop JK com *preset* e *clear*

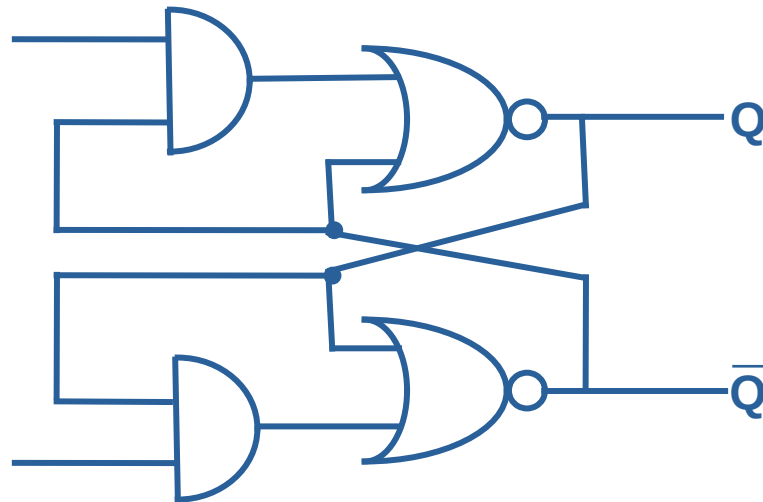
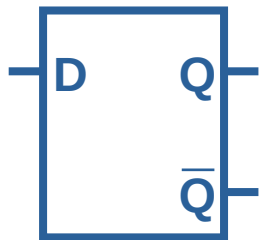


clear	preset	clk	J	K	$Q_{t+1}$	Comentário
0	1	X	X	X	1	estado set
1	0	X	X	X	0	estado reset
0	0	↑	0	0	$Q_t$	mantém anterior
0	0	↑	0	1	0	estado reset
0	0	↑	1	0	1	estado set
0	0	↑	1	1	$\overline{Q_t}$	inverte estado

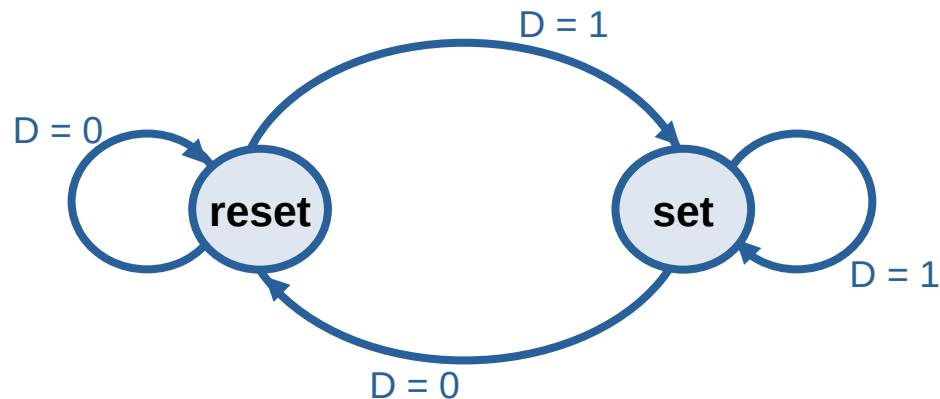


# Circuitos Sequenciais

## Latch D

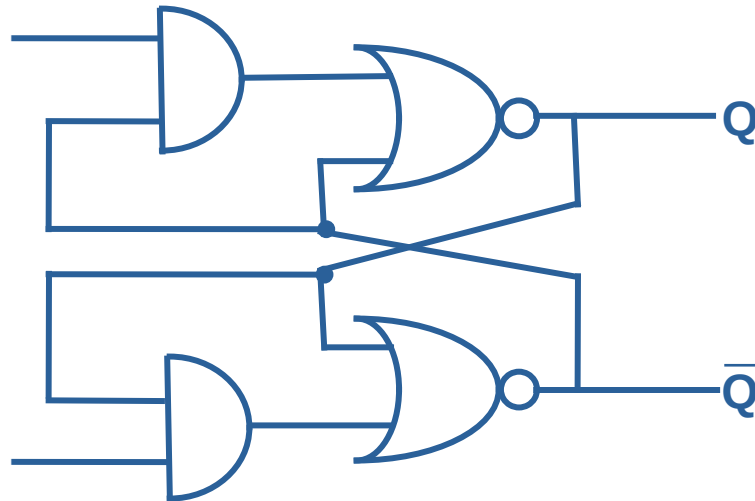
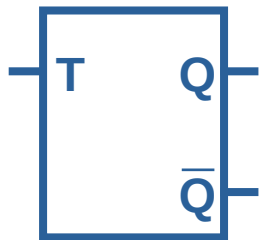


D	K	$Q_{t+1}$	Comentário
0	0	$Q_t$	mantém anterior
0	1	0	estado reset
1	0	1	estado set
1	1	$\overline{Q_t}$	inverte anterior

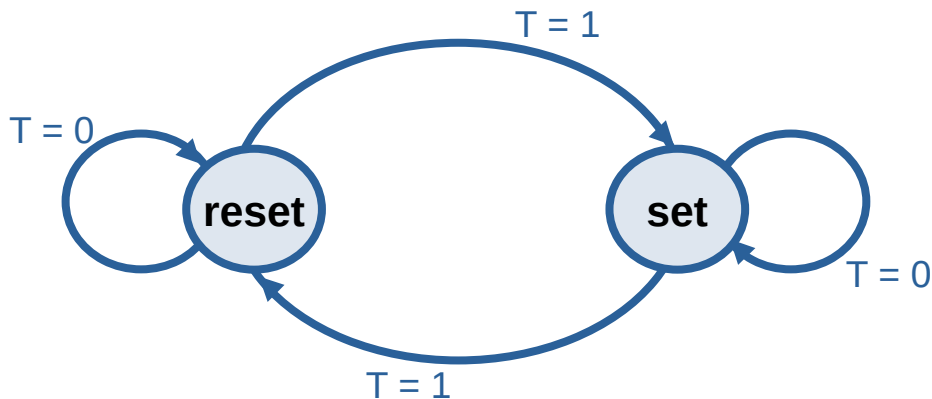


# Circuitos Sequenciais

## Latch T

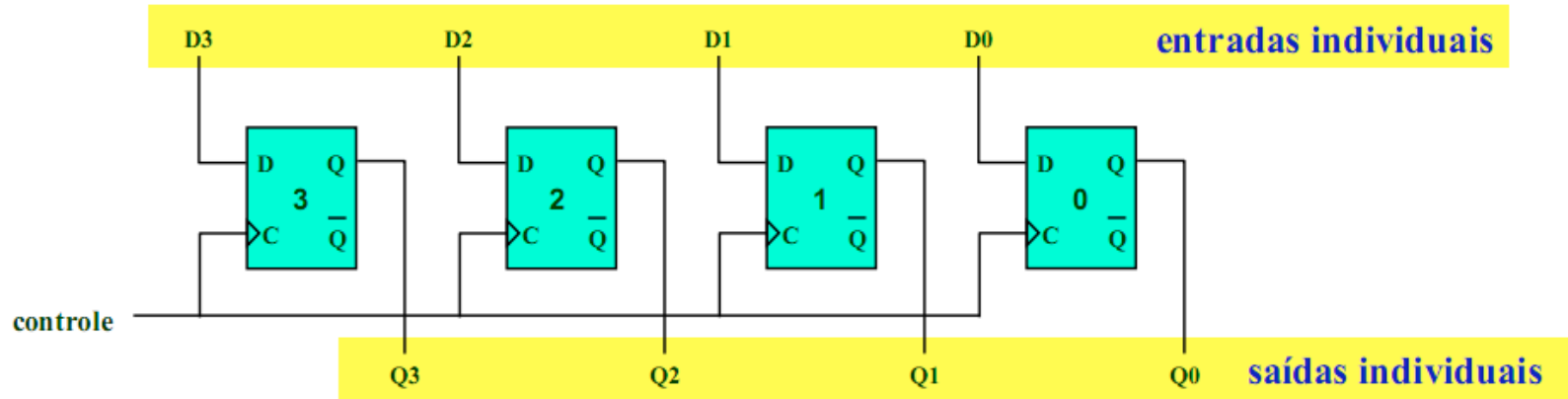


T	K	$Q_{t+1}$	Comentário
0	0	$Q_t$	mantém anterior
0	1	0	estado reset
1	0	1	estado set
1	1	$\overline{Q_t}$	inverte anterior



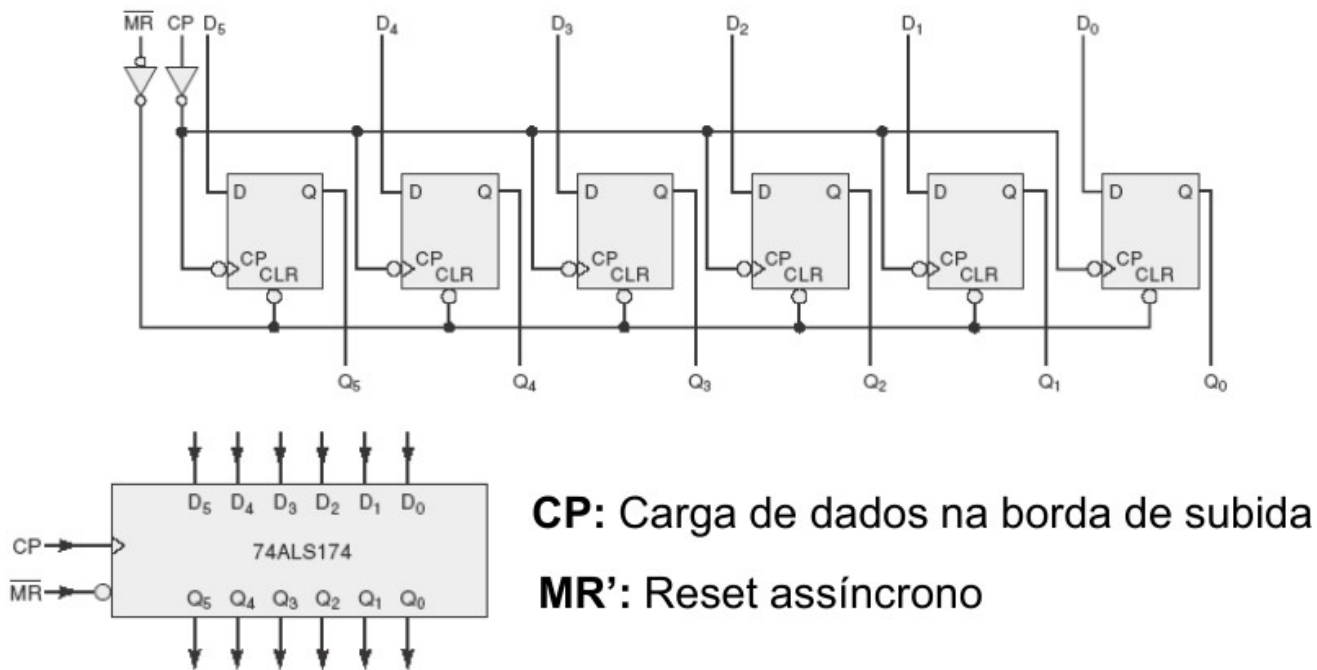
# Circuitos Sequenciais

## Registrador de carga paralela



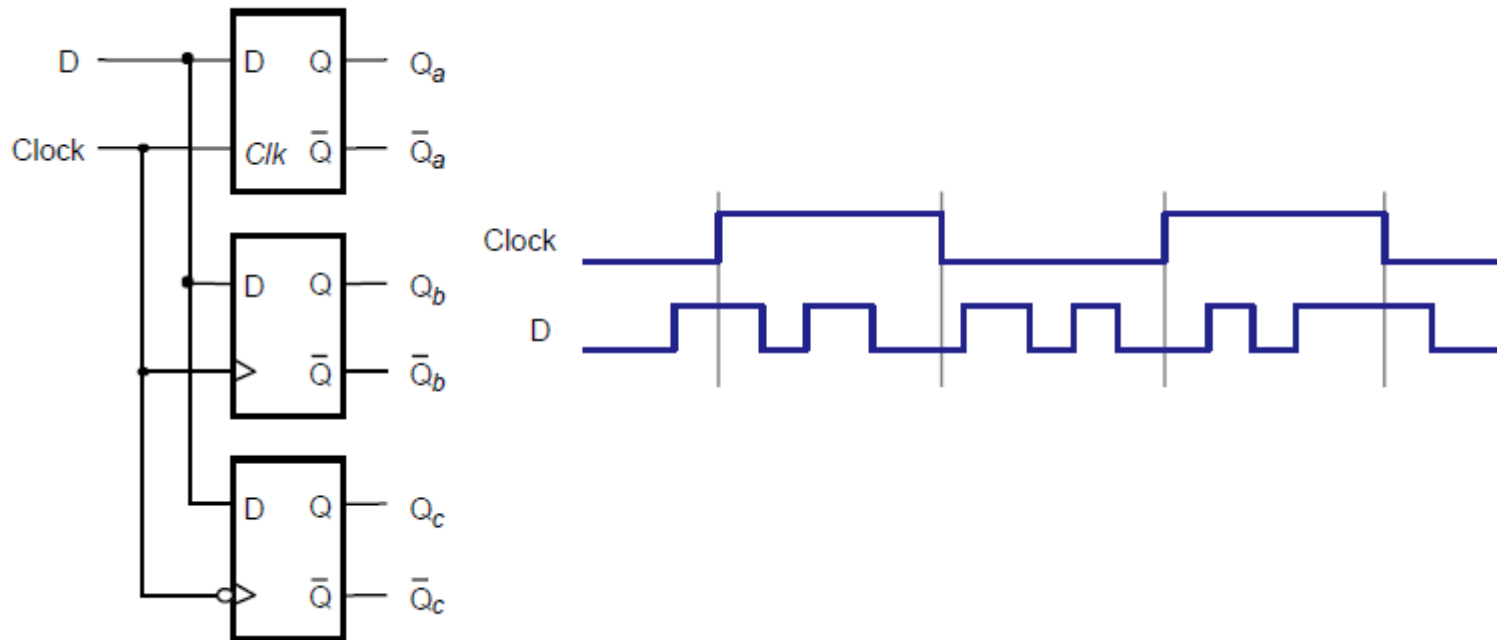
# Circuitos Sequenciais

## Registrador de carga paralela



# Circuitos Sequenciais

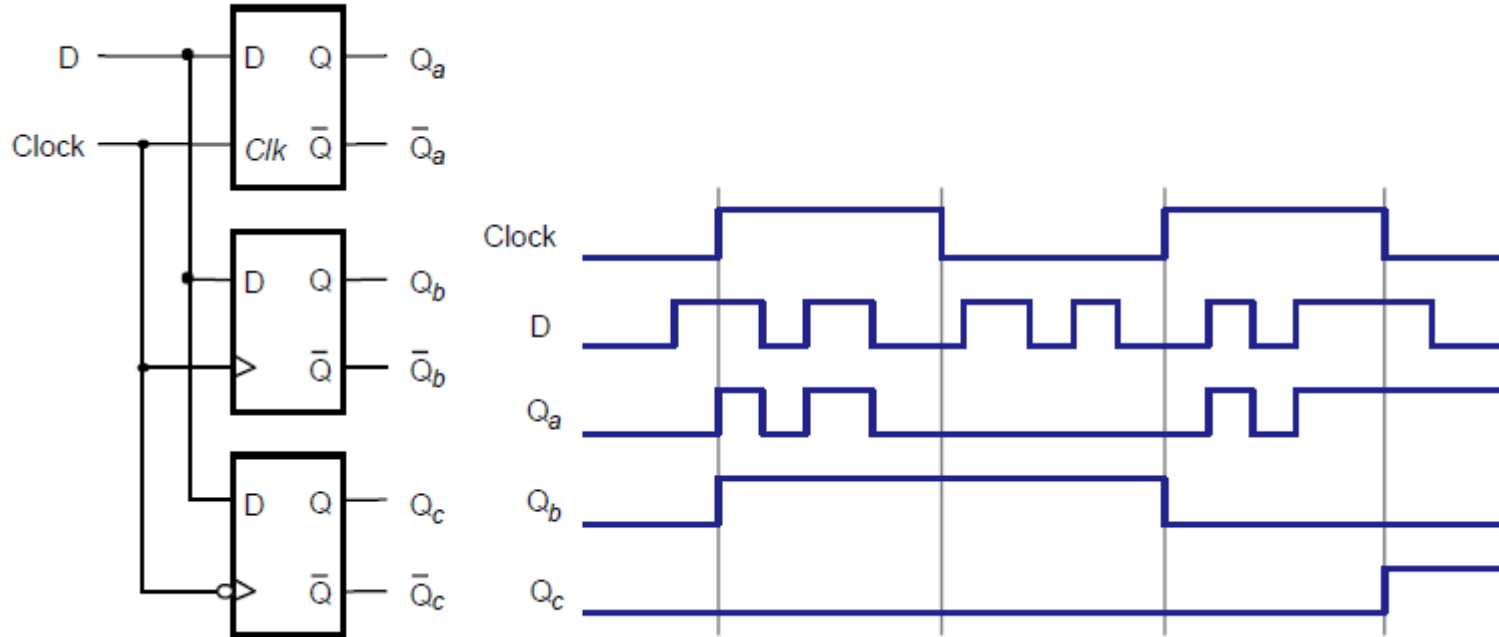
**Exercício: apresente as saídas Qa, Qb e Qc**





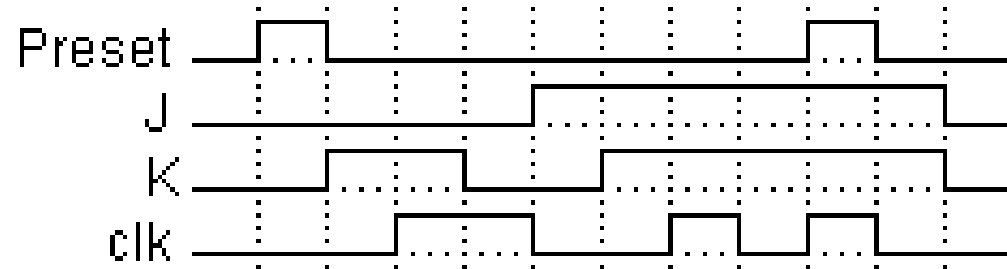
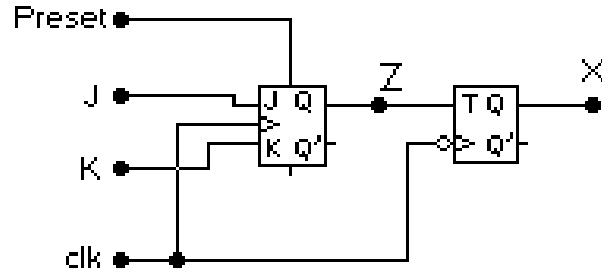
# Circuitos Sequenciais

**Exercício: apresente as saídas  $Q_a$ ,  $Q_b$  e  $Q_c$**



# Circuitos Sequenciais

**Exercício: apresente os valores de Z e X**



# Circuitos Sequenciais

## Banco de Registradores

- Neste exemplo:
  - entrada e armazenamento de dados de 4 bits
  - 8 registradores
  - 2x3 bits de seleção de leitura
  - 3 bits de seleção de escrita
  - 1 bit para escrita (clk)

