

---

# Circuitos Sequenciais

---

*Luciano L. Caimi*

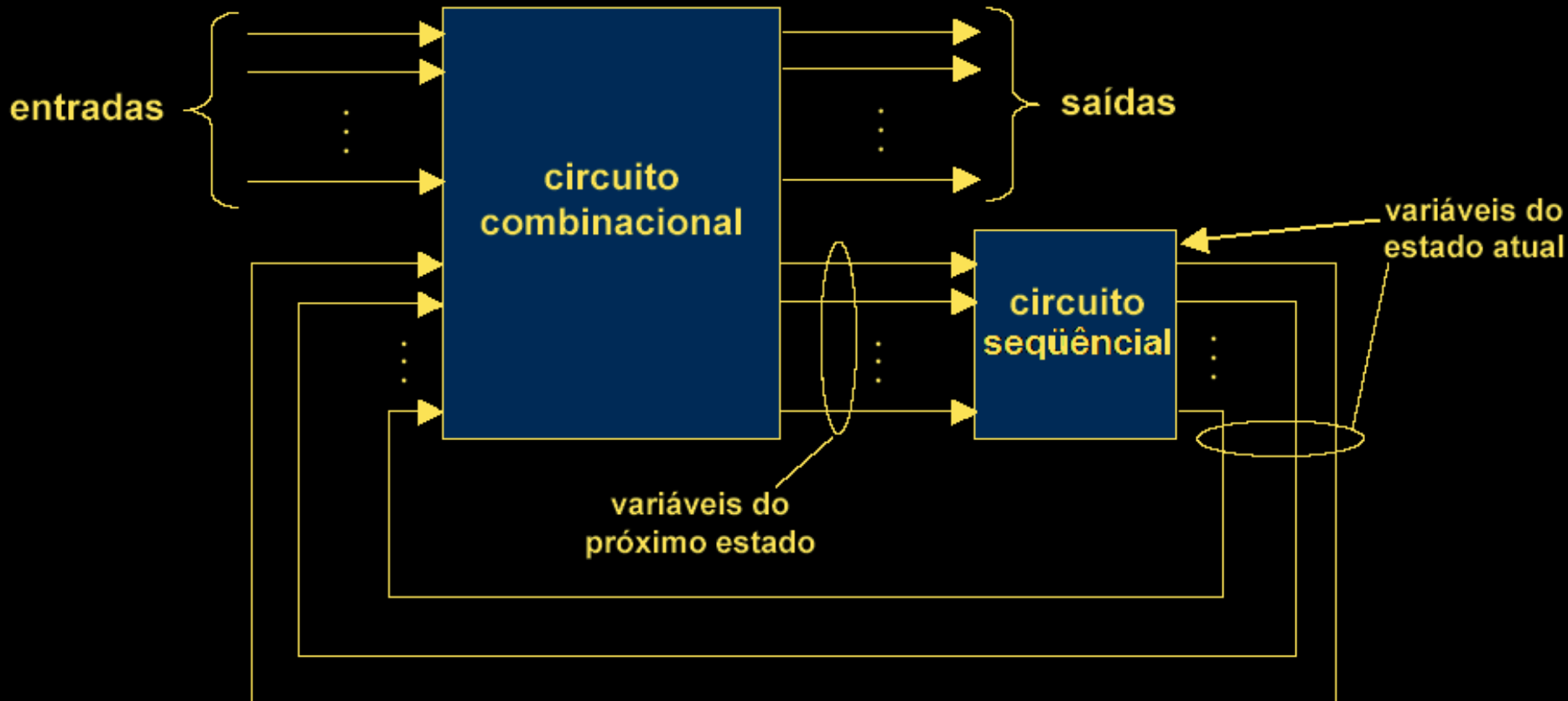
`lcaimi@uffrs.edu.br`

# Circuitos Sequenciais



- 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:**

Os circuitos sequenciais são formados a partir de duas estruturas básicas, os **Latches** e os **Flip-Flops**.

Diferentes modelos de cada um destes elementos são utilizados para construir os diferentes circuitos seqüenciais.

Os **Latches** são elementos **assíncronos**, ou seja, operam sem restrições temporais.

Os **Flip-Flops** são circuitos **síncronos**, isto é, operam com restrições de tempo.

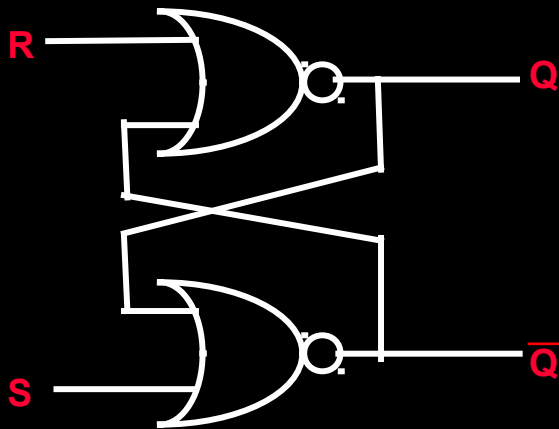
De fato, quanto ao funcionamento lógico, existem modelos de igual funcionamento tanto para os latches quanto para os flip-flops, sendo as diferenças baseadas apenas na existência ou não de restrições temporais.

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

# Elementos Básicos



- Latch RS



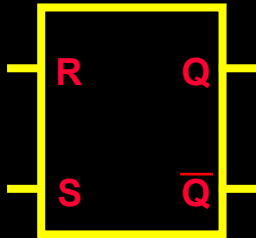
Latch RS com portas NOR

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		

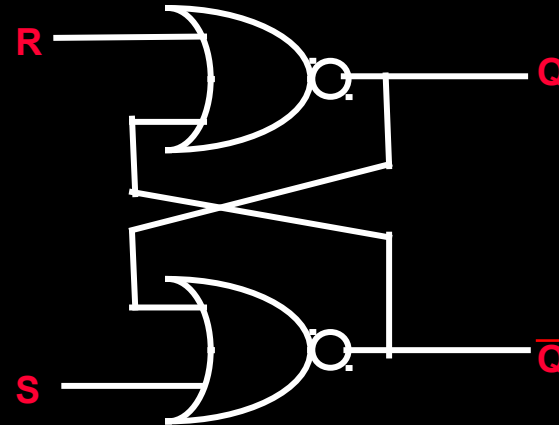
# Elementos Básicos



- Latch RS

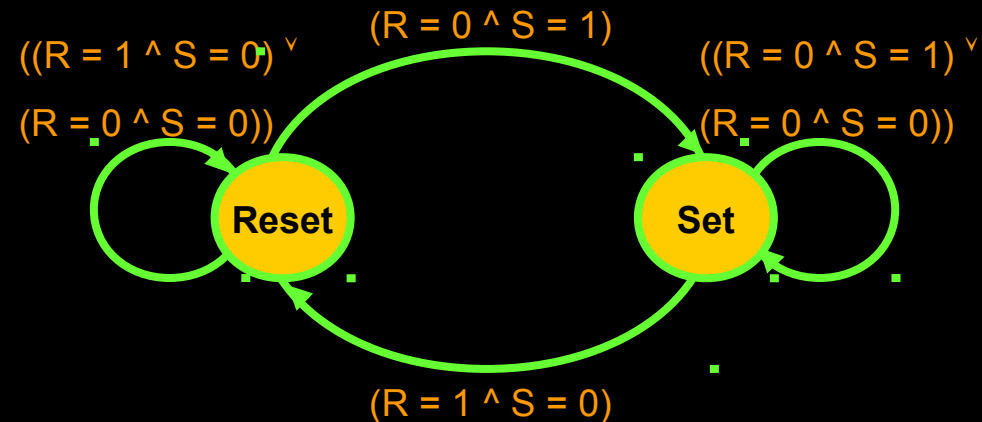


Símbolo

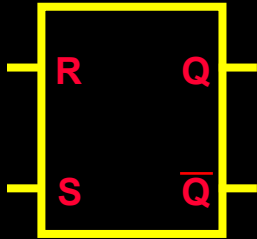


Latch RS com portas NOR

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

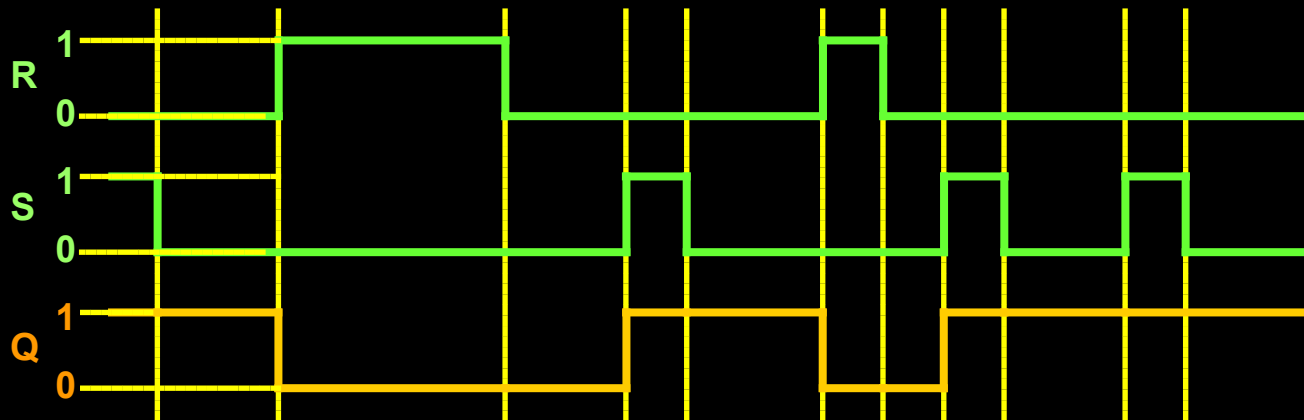


# Elementos Básicos



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

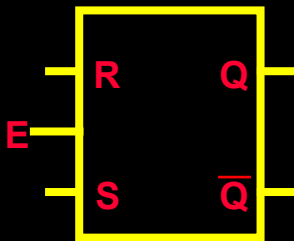
## Exemplo com Latch RS



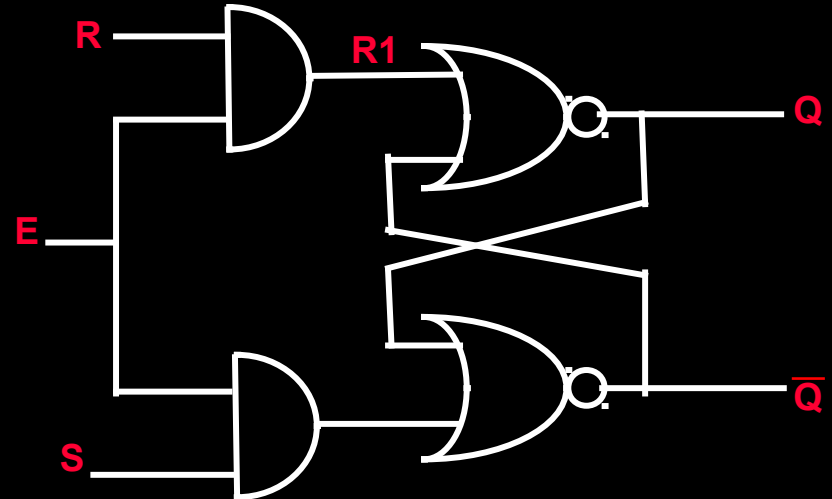
# Elementos Básicos



- Latch RS com Enable

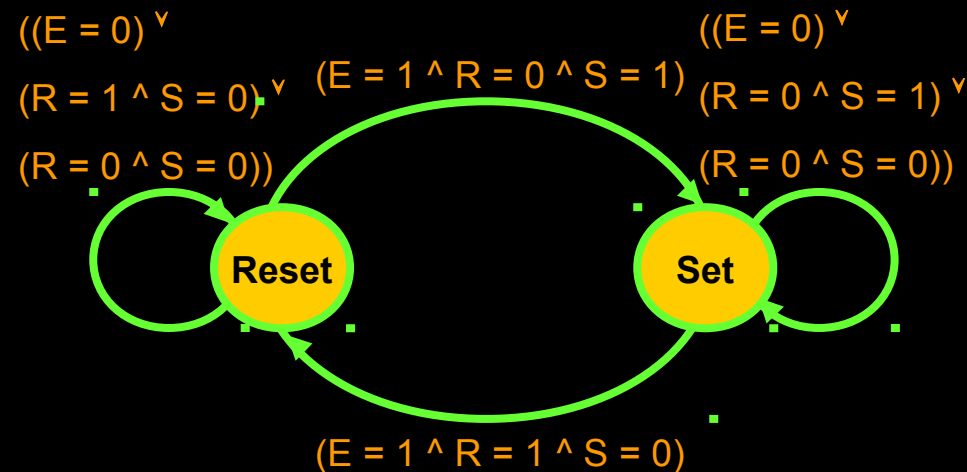


Símbolo



Latch RS com Enable

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

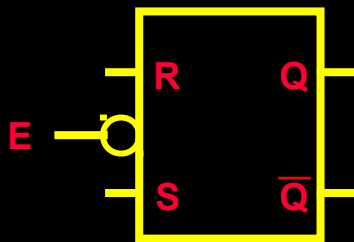




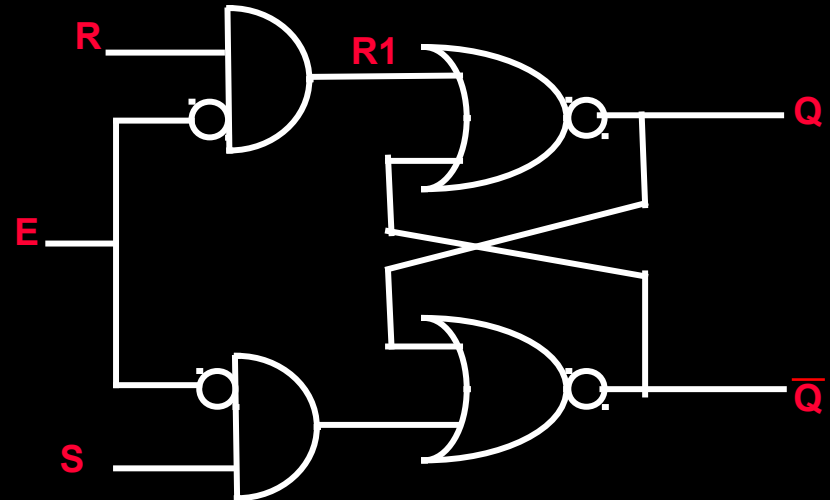
# Elementos Básicos



- Latch RS com *Enable* complementar

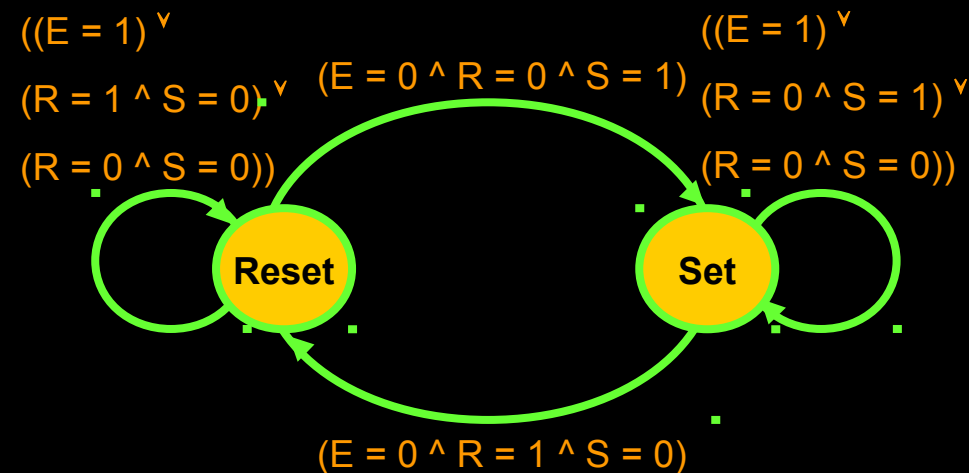


Símbolo

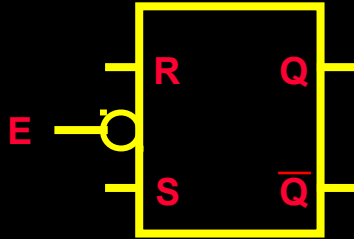


Latch RS com Enable Complementar

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

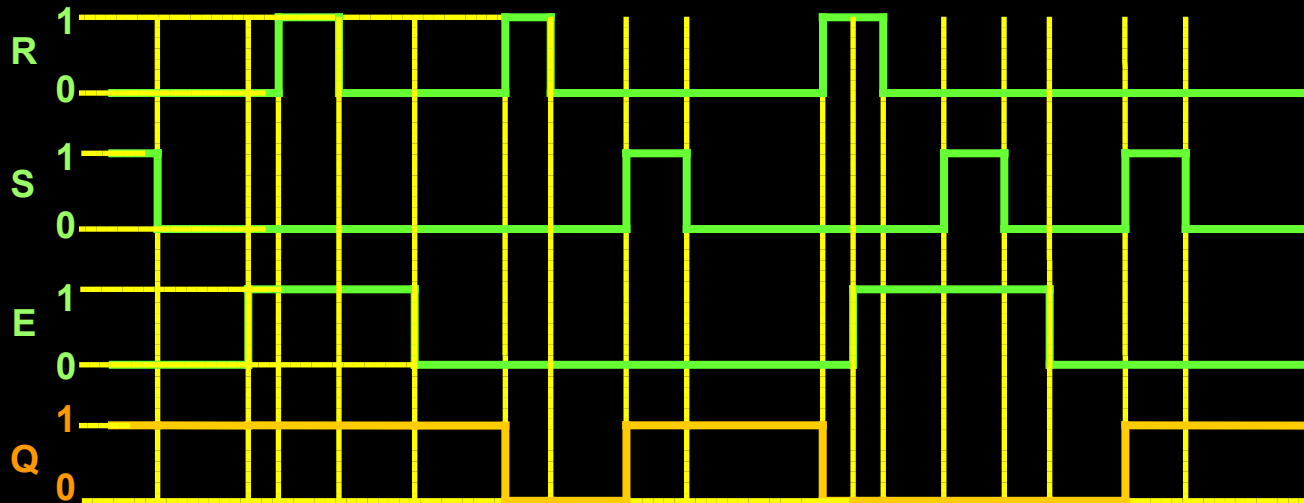


# Elementos Básicos



**Exemplo Latch RS com  
*Enable* Complementar**

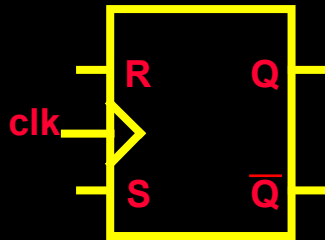
E	R	S	$Q_{t+1}$	Comentário
1	X	X	$Q_t$	mantém anterior
0	0	0	$Q_t$	mantém anterior
0	0	1	1	estado set
0	1	0	0	estado reset
0	1	1	-	estado inválido



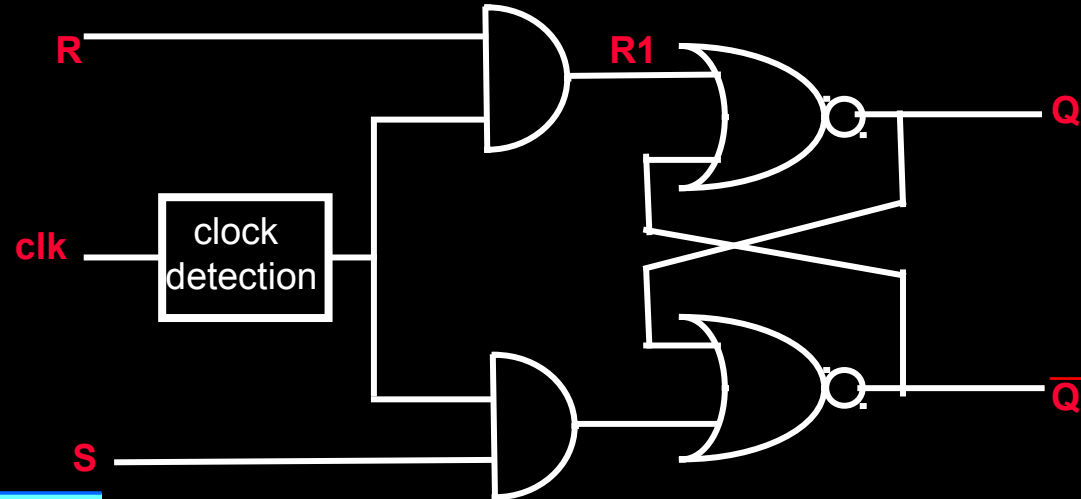
# Elementos Básicos



- Flip-Flop RS



Símbolo



Flip 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

$$((clk \neq 1) \vee$$

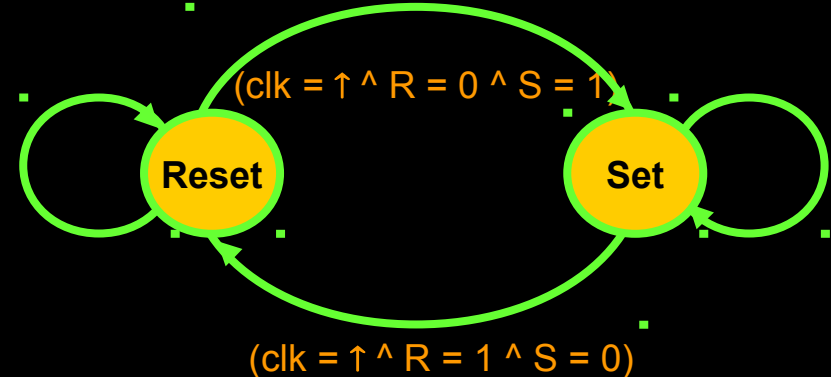
$$(clk = 1 \wedge R = 1 \wedge S = 0) \vee$$

$$(clk = 1 \wedge R = 0 \wedge S = 0))$$

$$((clk \neq 1) \vee$$

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

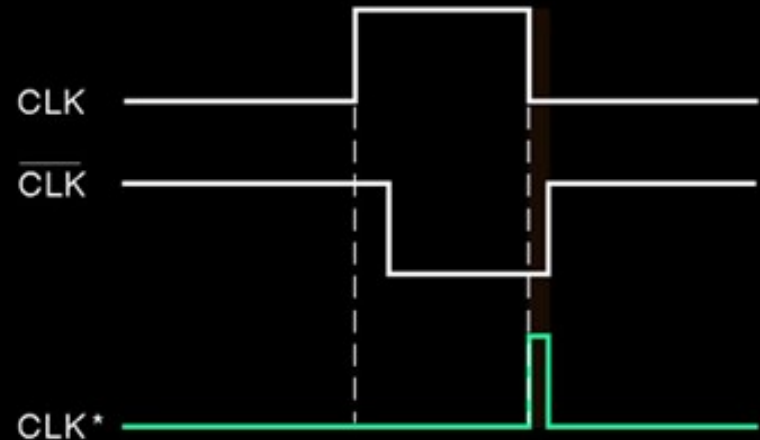
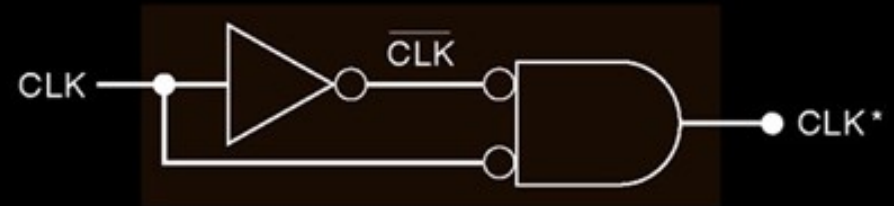
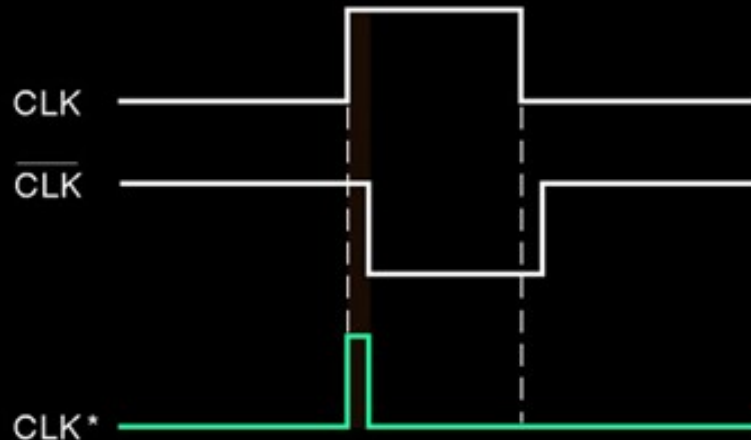
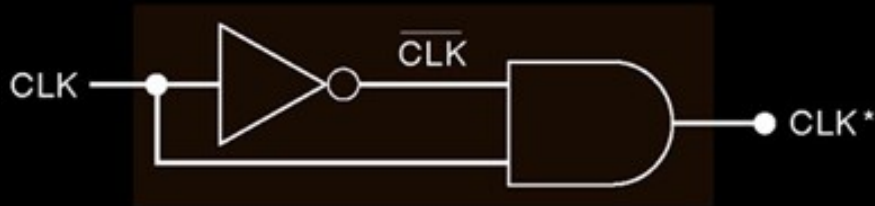
$$(clk = 1 \wedge R = 0 \wedge S = 0))$$



# Elementos Básicos



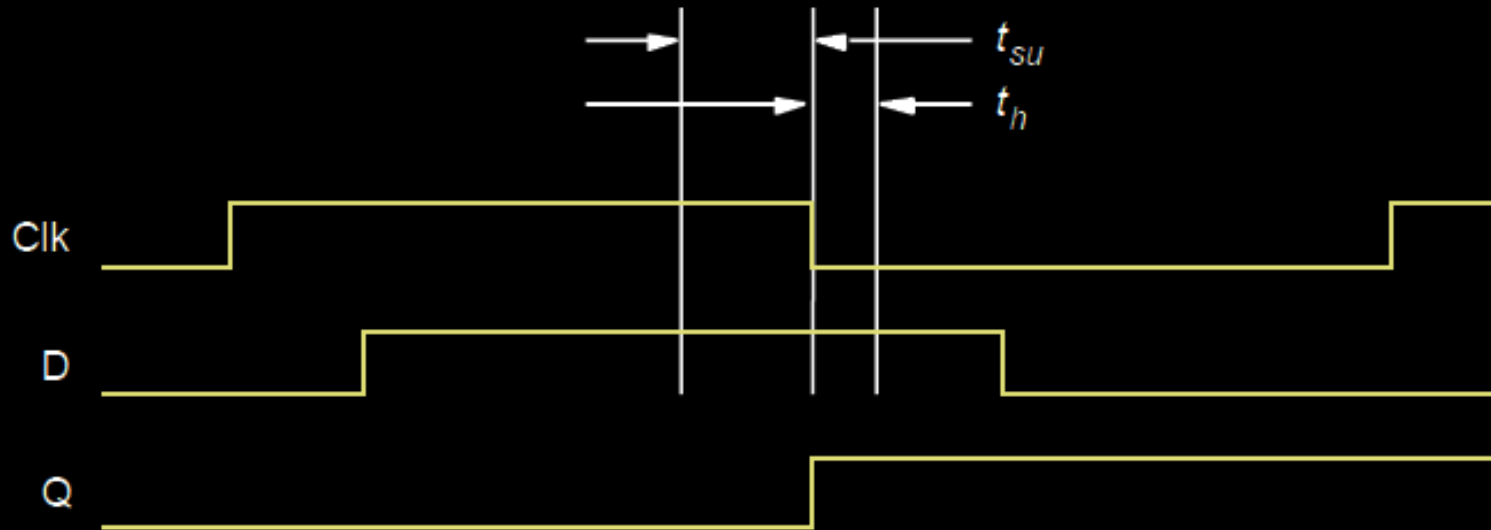
- Detecção de borda



# Elementos Básicos



## Temporização em FF

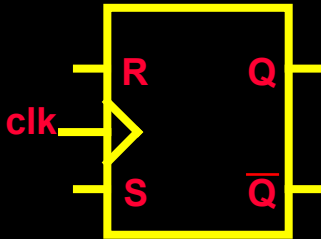


Os tempos de setup e hold são parâmetros que devem ser observados para que o flip-flop possa trabalhar de modo confiável.

O tempo de setup,  $t_{su}$ , corresponde ao intervalo no qual as entradas devem permanecer estáveis antes da transição do clock.

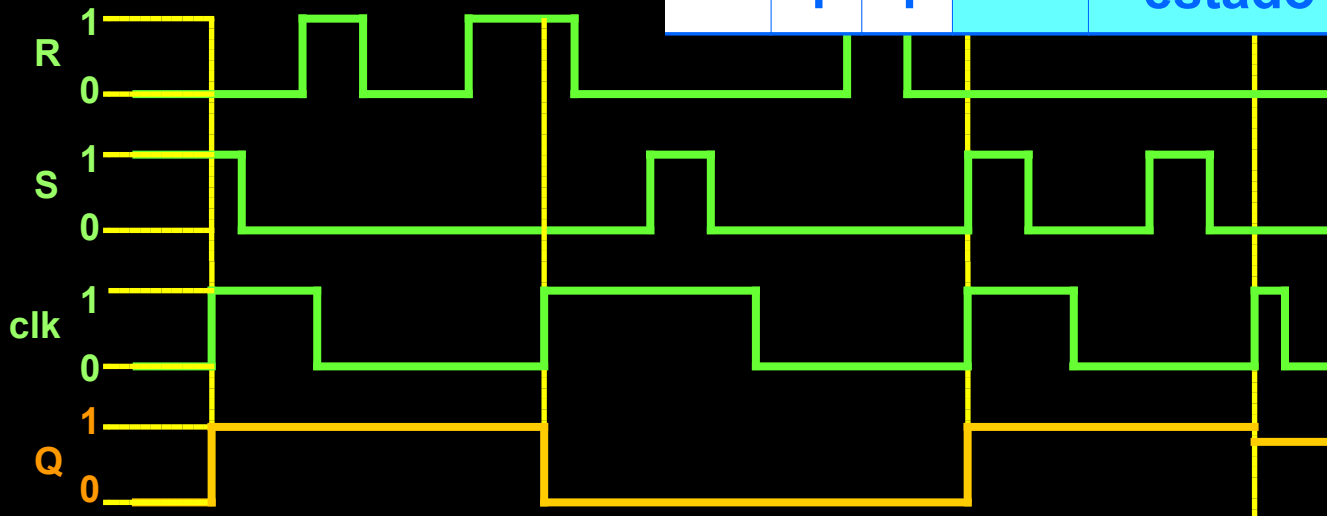
O tempo de hold,  $t_h$ , corresponde ao intervalo no qual as entradas devem permanecer estáveis depois da transição do clock.

# Elementos Básicos



## Exemplo FlipFlop RS

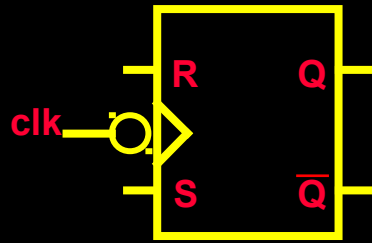
clk	R	S	$Q_{t+1}$	Comentário
1	X	X	$Q_t$	mantém anterior
0	X	X	$Q_t$	mantém anterior
↓	X	X	$Q_t$	mantém anterior
↑	0	0	$Q_t$	mantém anterior
↑	0	1	1	estado set
↑	1	0	0	estado reset
↑	1	1	-	estado inválido



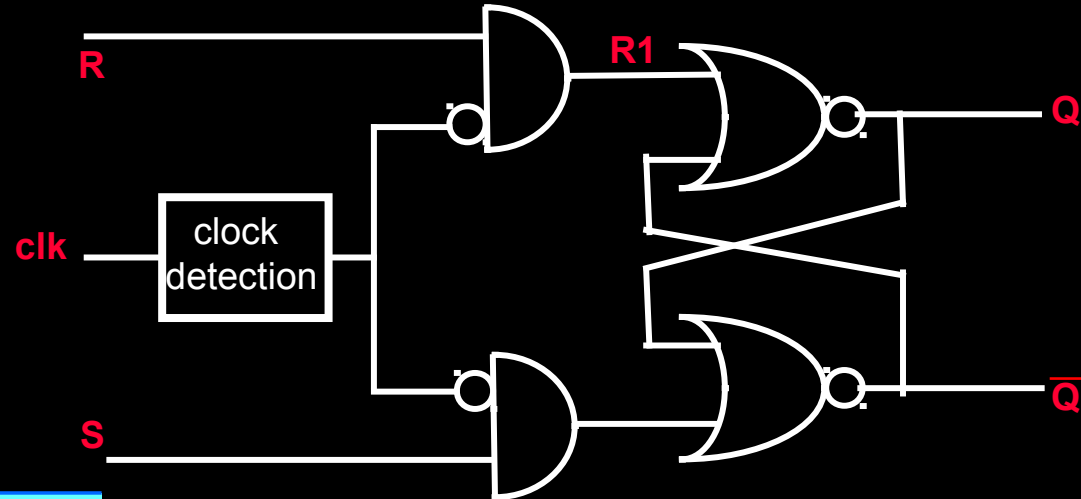
# Elementos Básicos



## • Flip-Flop RS



Símbolo



Flip 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

$$((clk \neq \downarrow) \vee$$

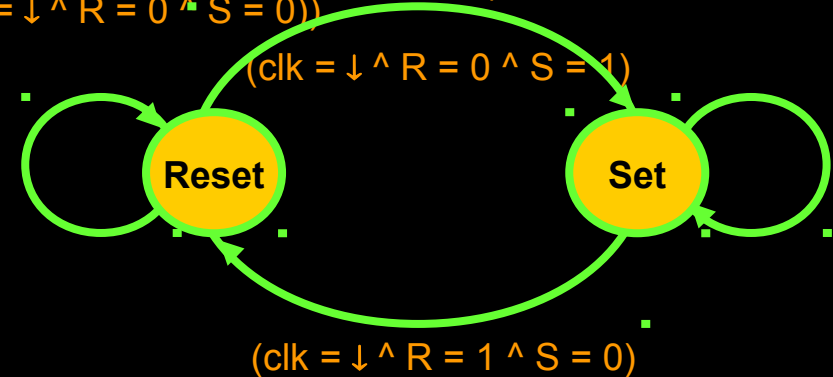
$$(clk = \downarrow \wedge R = 1 \wedge S = 0) \vee$$

$$(clk = \downarrow \wedge R = 0 \wedge S = 0))$$

$$((clk \neq \downarrow) \vee$$

$$(clk = \downarrow \wedge R = 0 \wedge S = 1) \vee$$

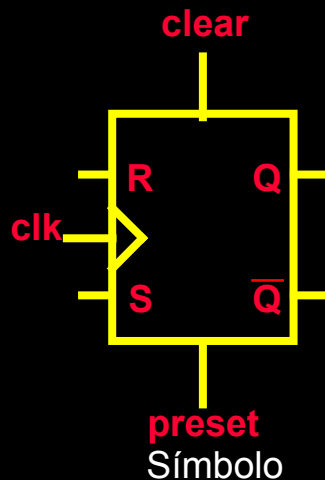
$$(clk = \downarrow \wedge R = 0 \wedge S = 0))$$



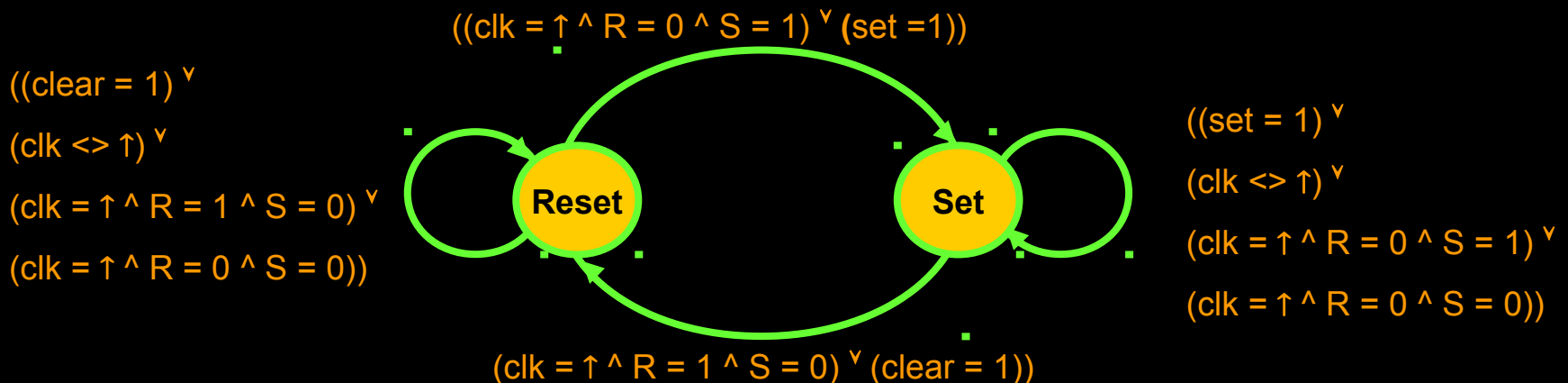
# Elementos Básicos



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



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

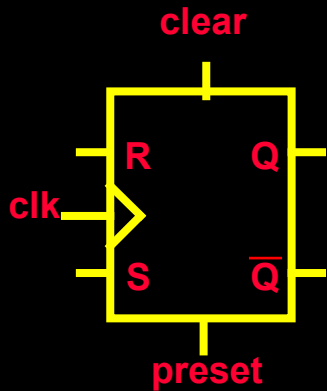




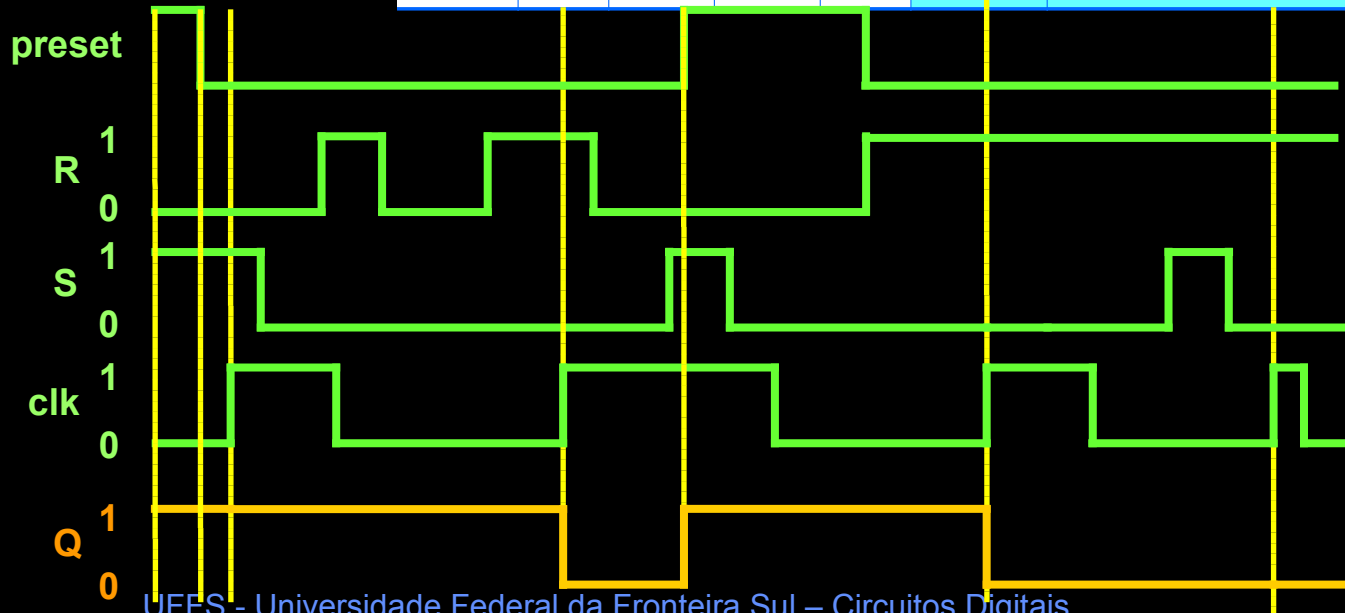
# Elementos Básicos



Flip-Flop RS com  
entradas assíncronas  
de set e clear



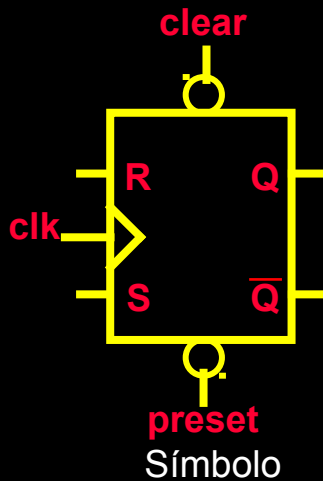
clear	pre set	clk	R	S	$Q_{t+1}$	Comentário
0	1	X	X	X	1	estado set
1	0	X	X	X	0	estado reset
0	0	$\uparrow$	0	0	$Q_t$	mantém anterior
0	0	$\uparrow$	0	1	1	estado set
0	0	$\uparrow$	1	0	0	estado reset
0	0	$\uparrow$	1	1	-	estado inválido



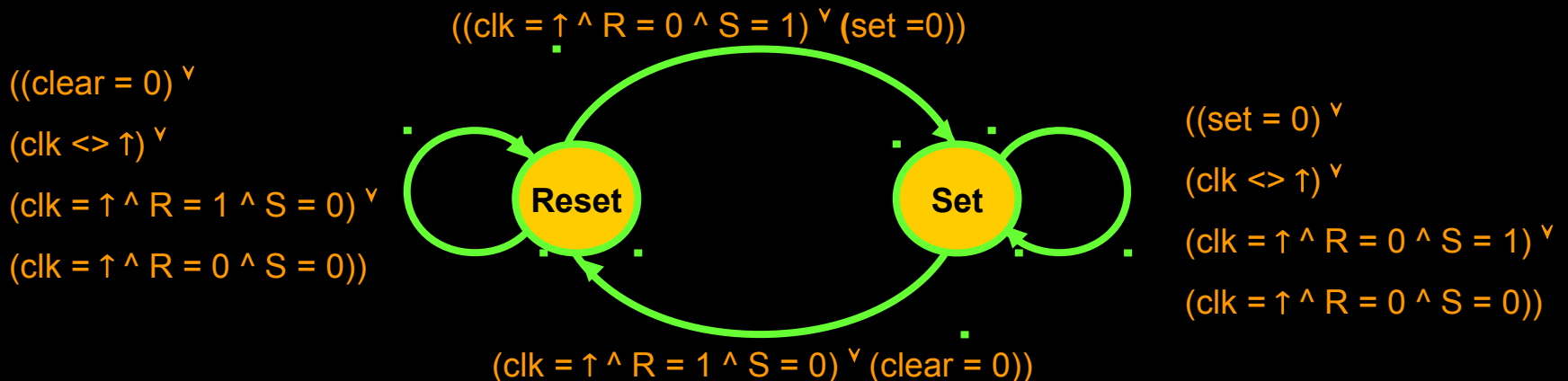
# Elementos Básicos



Flip-Flop RS com  
entradas assíncronas  
de preset e clear  
complementares



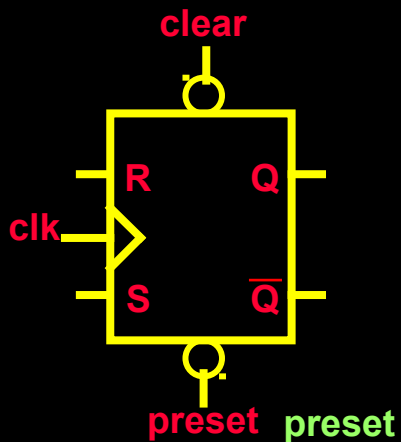
Comportamento idêntico ao circuito  
anterior usando, desta vez, lógica  
complementar nas entradas  
preset e clear.



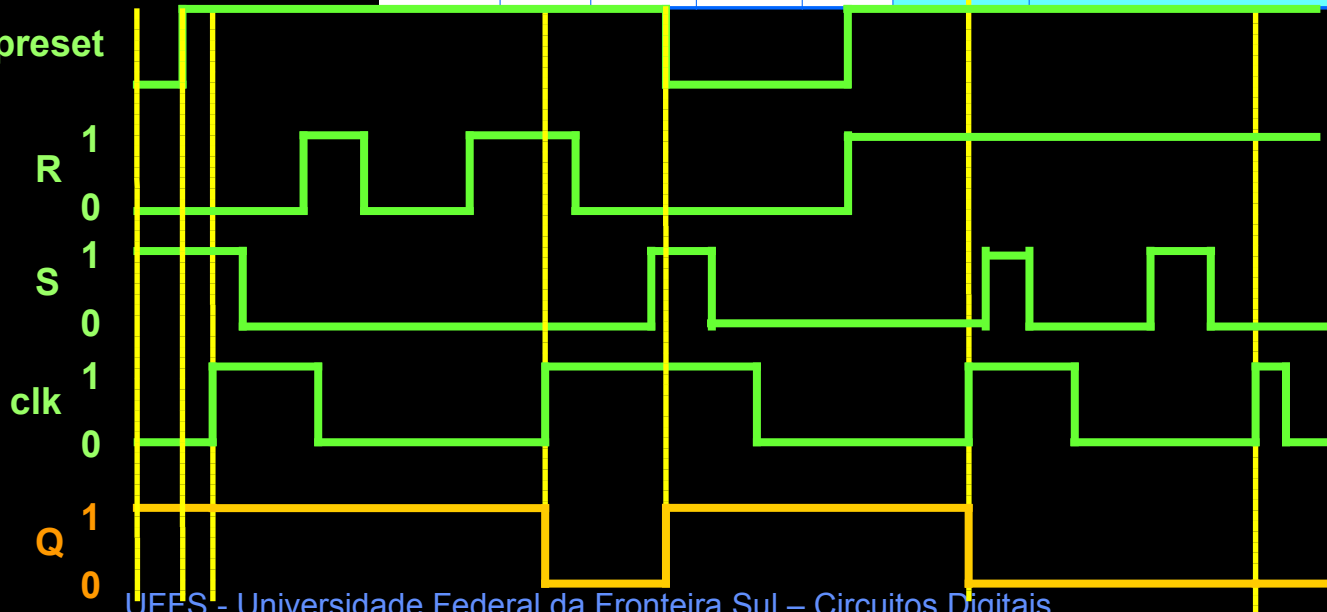
# Elementos Básicos



Flip-Flop RS com  
entradas assíncronas  
de set e clear  
complementares



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



# Elementos Básicos



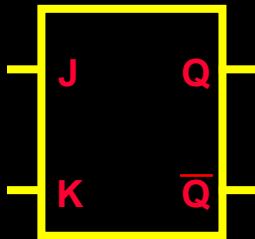
Existem vários outros latches e Flip-flop, são eles: **JK**, **D** e **T**

A seguir veremos cada um deles. É importante ter presente que a mudança acontece basicamente na tabela verdade, mas todas as possíveis implementações vistas anteriormente estão presentes, isto é, com enable, com enable complementar, gatilhado na transição de subida, na de descida, com set e clear, com set e clear complementar, ...

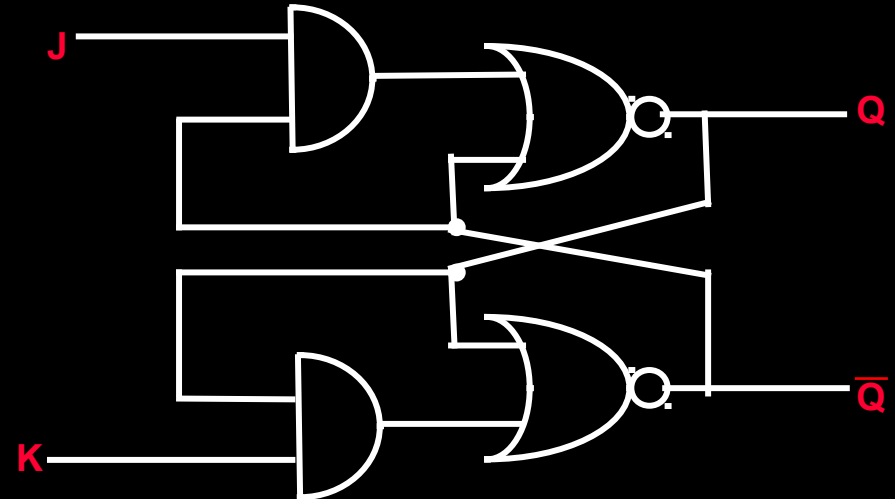
# Elementos Básicos



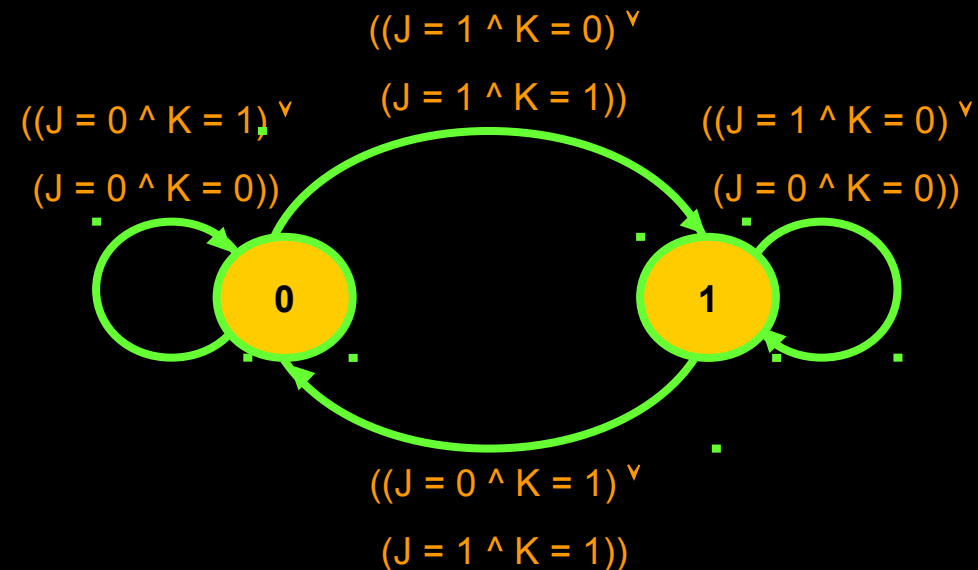
- Latch JK



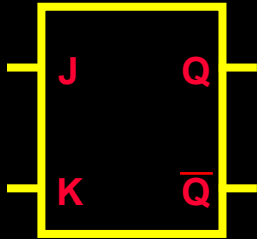
Símbolo



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 estado

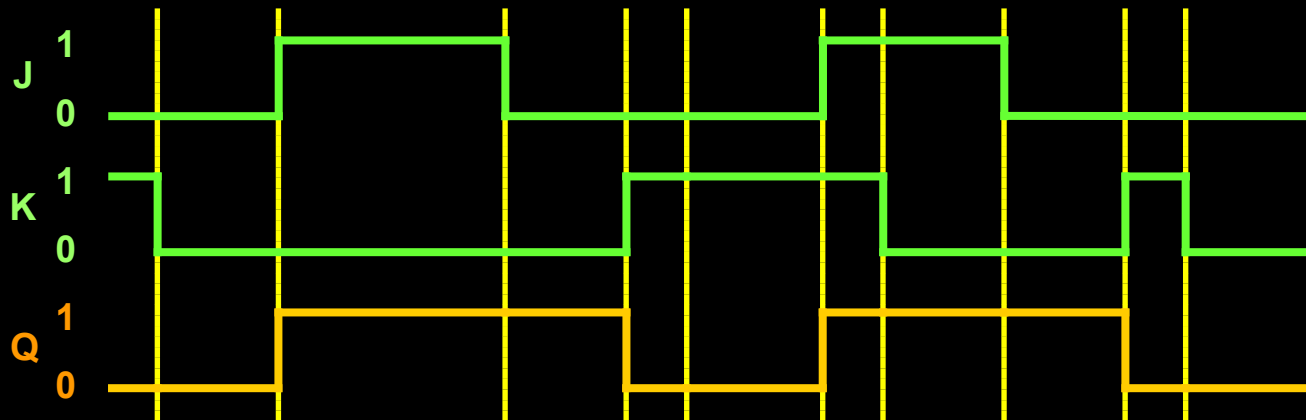


# Elementos Básicos

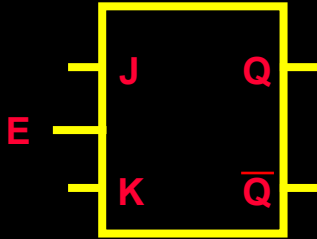


J	K	$Q_{t+1}$	Comentário
0	0	$Q_t$	mantém anterior
0	1	0	estado reset
1	0	1	estado set
1	1	$\overline{Q_t}$	inverte estado

## Exemplo com Latch JK

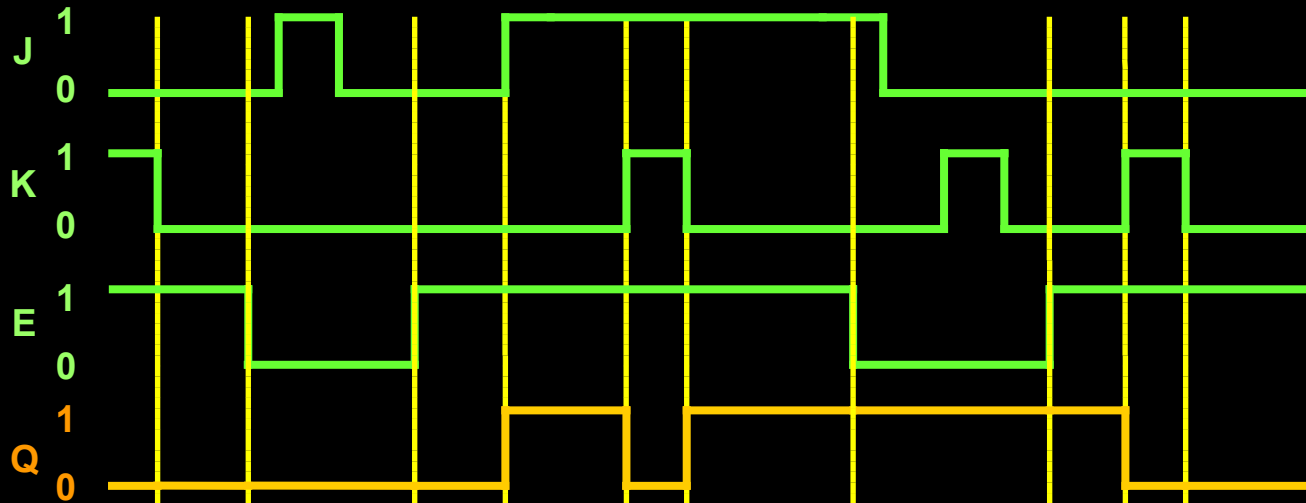


# Elementos Básicos



**Exemplo Latch JK com Enable**

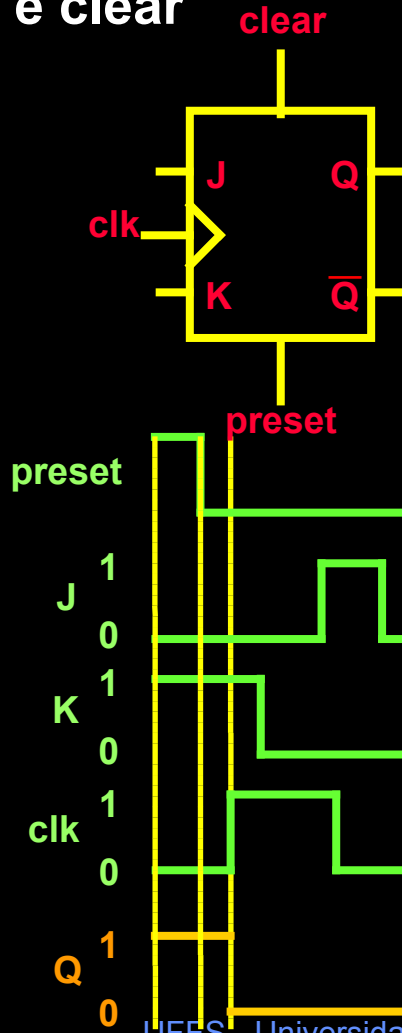
E	J	K	$Q_{t+1}$	Comentário
0	X	X	$Q_t$	mantém anterior
1	0	0	$Q_t$	mantém anterior
1	0	1	0	estado reset
1	1	0	1	estado set
1	1	1	$\overline{Q_t}$	inverte estado



# Elementos Básicos



Flip-Flop JK com entradas assíncronas de preset e clear



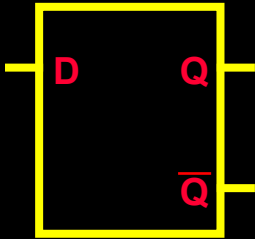
clear	pre set	clk	J	K	$Q_{t+1}$	Comentário
0	1	X	X	X	1	estado set
1	0	X	X	X	0	estado reset
0	0	↑	0	0	$Q_t$	mantém anterior
0	0	↑	0	1	0	estado reset
0	0	↑	1	0	1	estado set
0	0	↑	1	1	$Q_t$	inverte estado



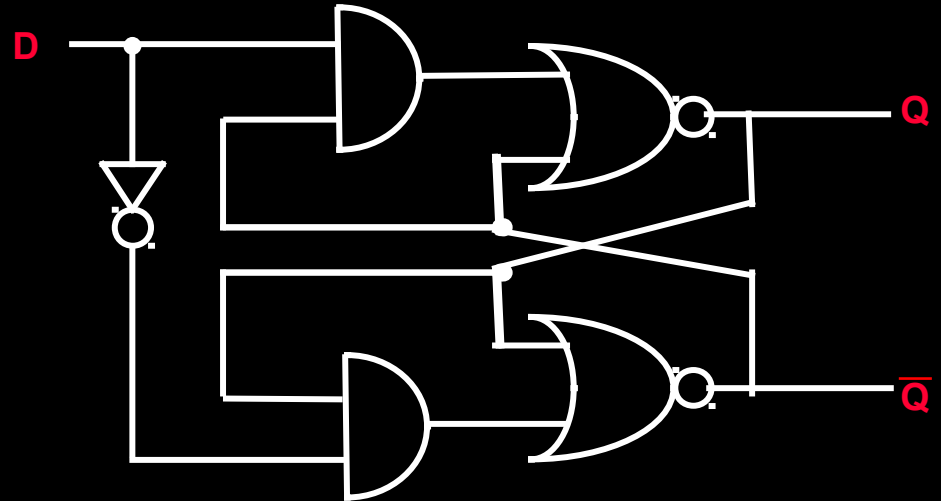
# Elementos Básicos



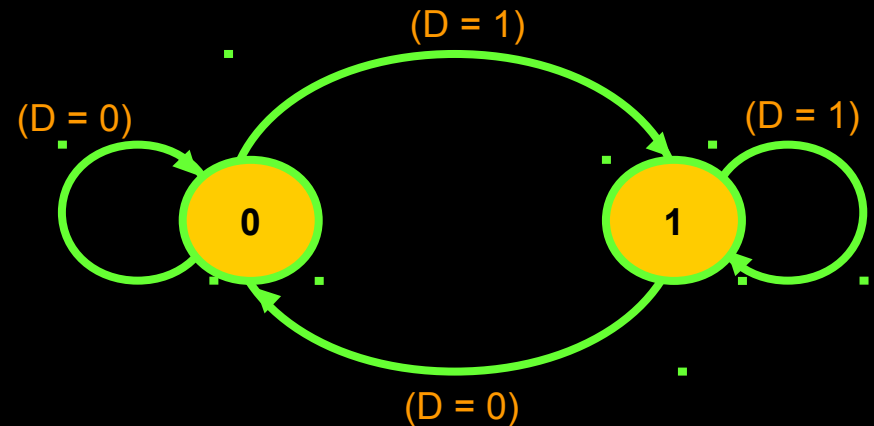
- Latch D



Símbolo



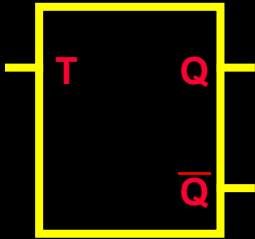
	0	1
0	estado reset	
1	estado set	



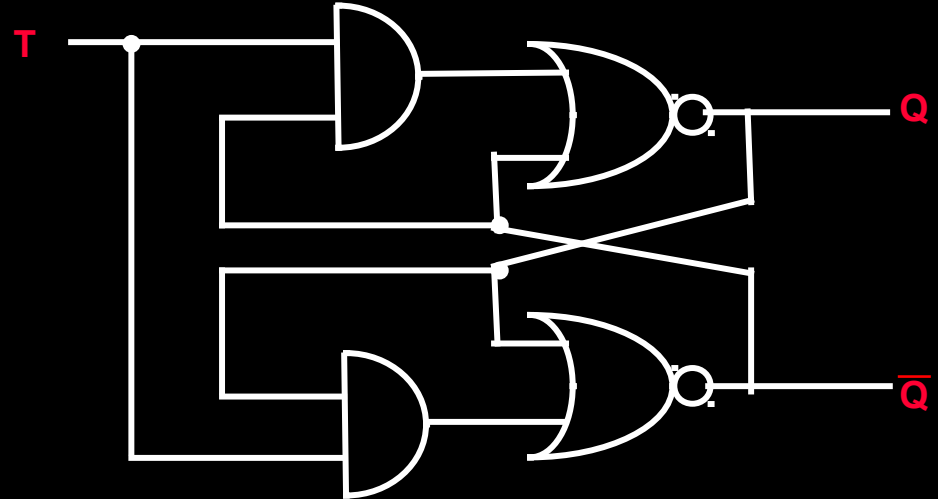
# Elementos Básicos



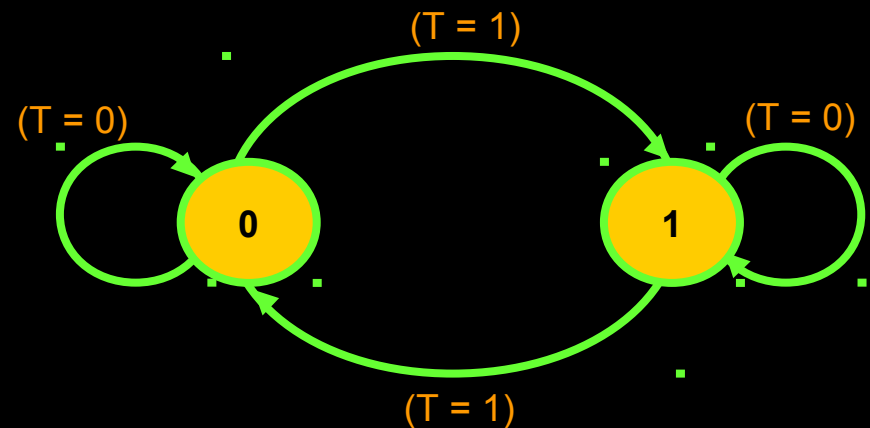
- Latch T



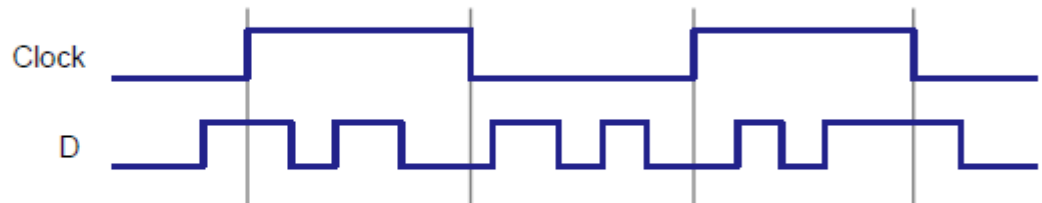
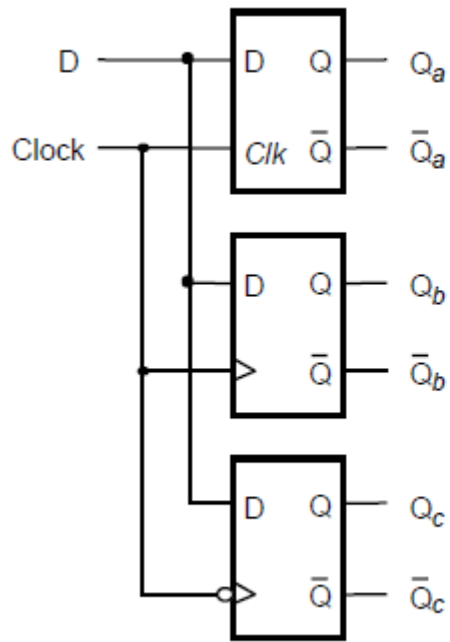
Símbolo



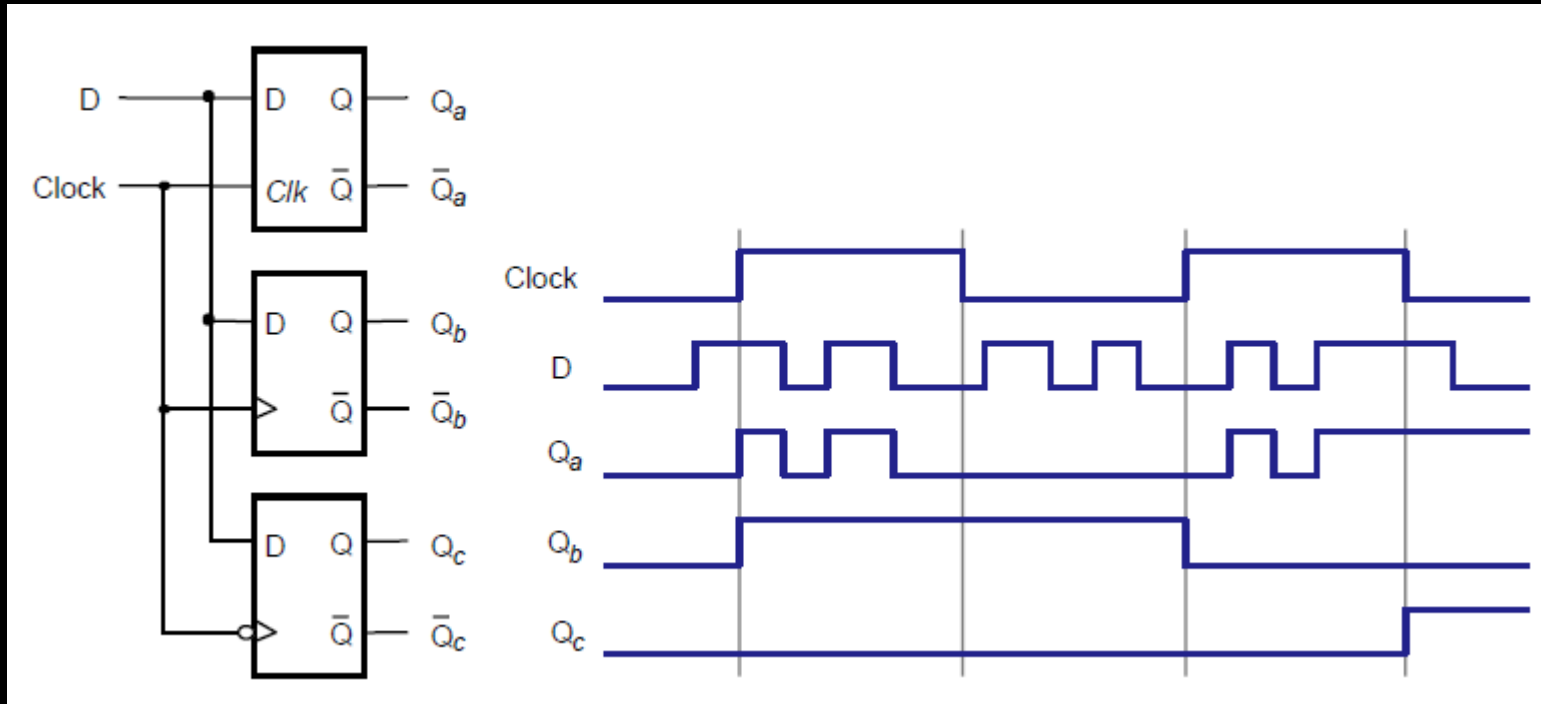
0	$Q_t$	mantém anterior
1	$\overline{Q_t}$	inverte estado



# Elementos Básicos



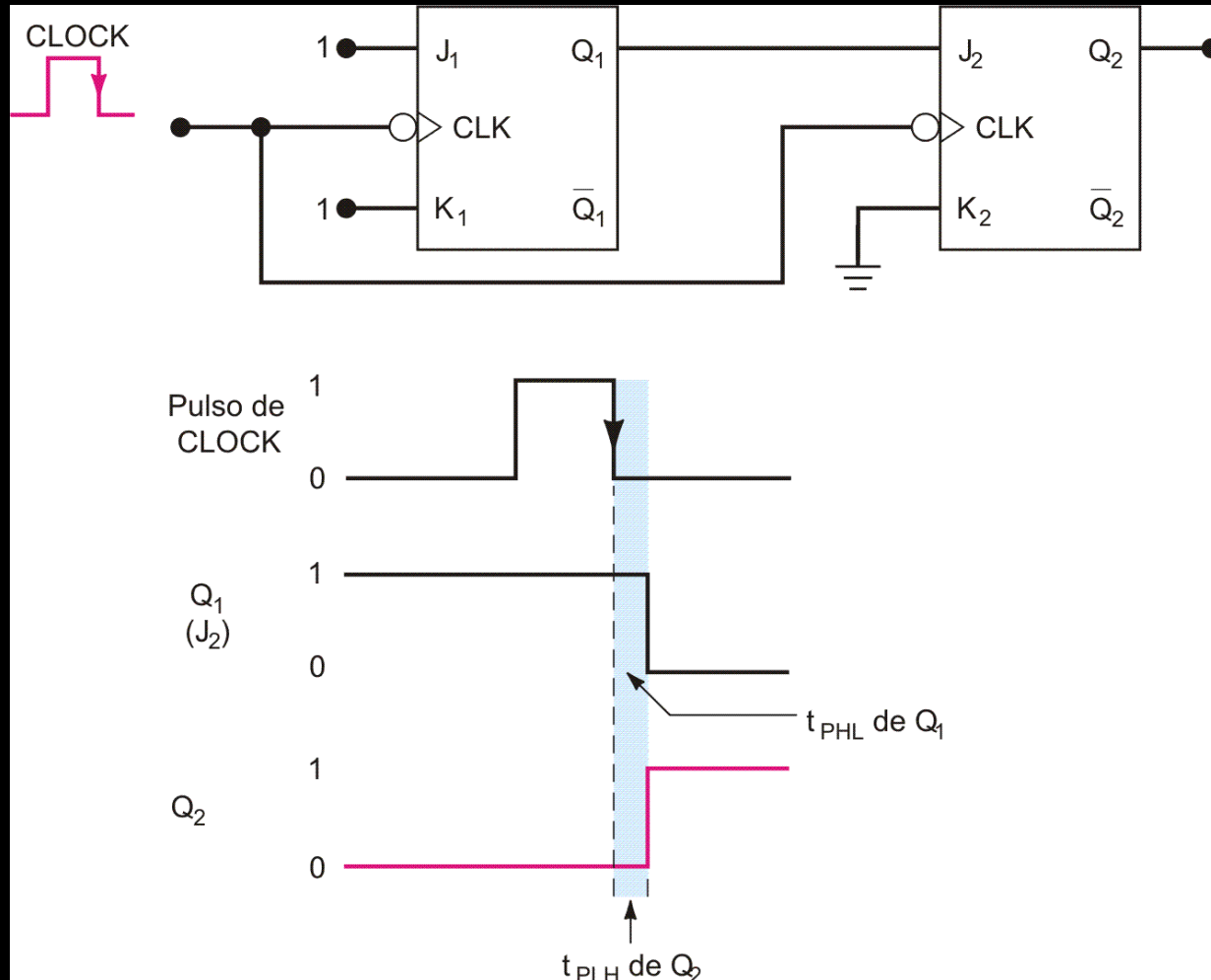
# Elementos Básicos



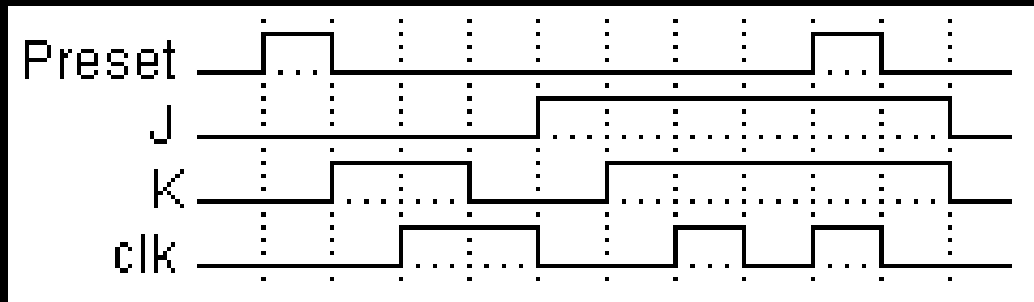
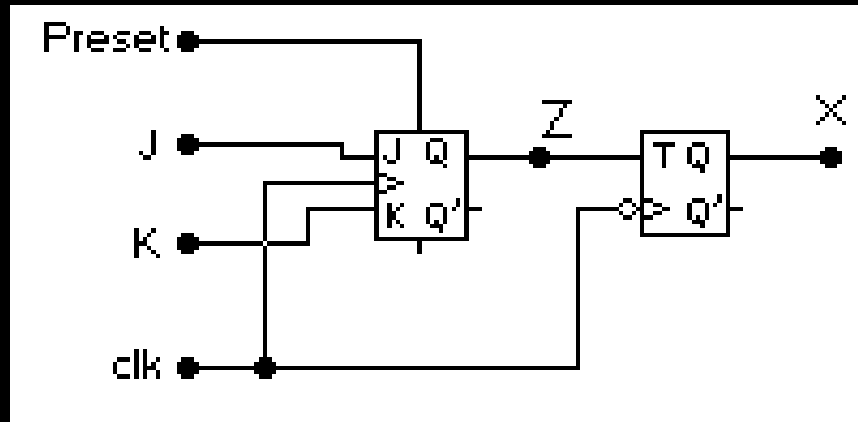
# Elementos Básicos



## Mais sobre Temporização



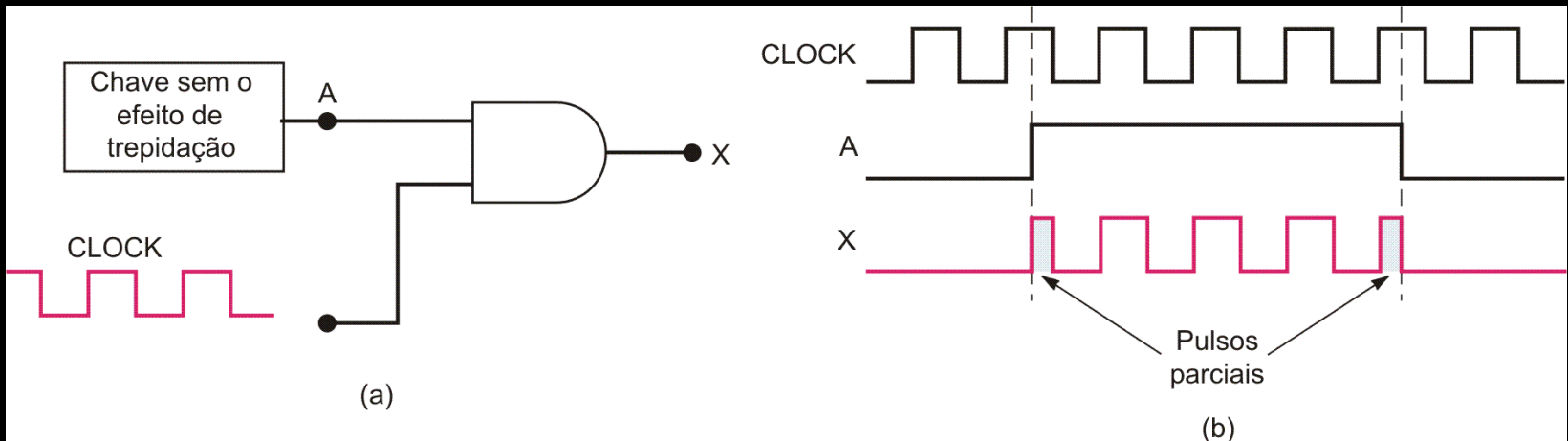
# Exercício



# Exemplos



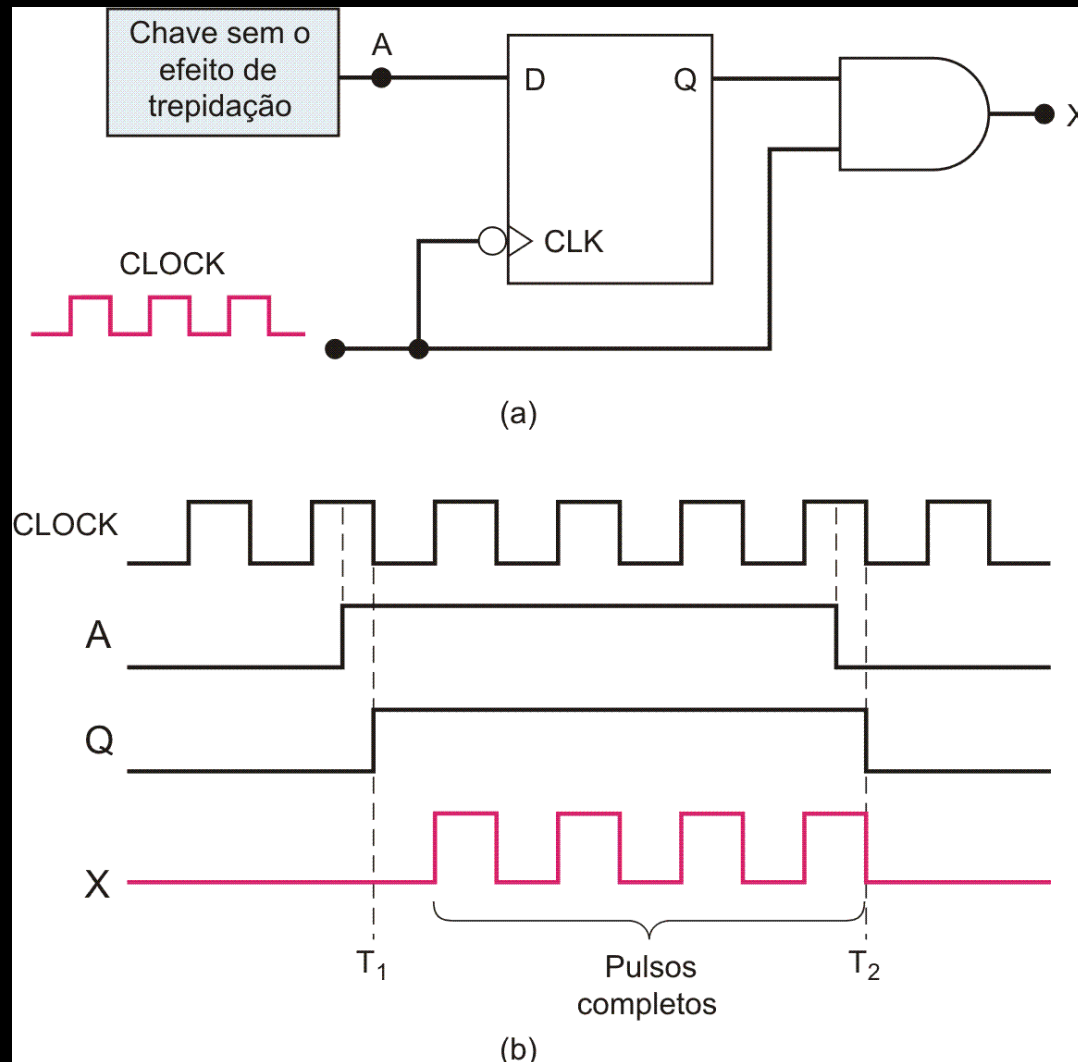
## Habilitando o clock (o problema)



# Exemplos



## Habilitando o clock (a solução)

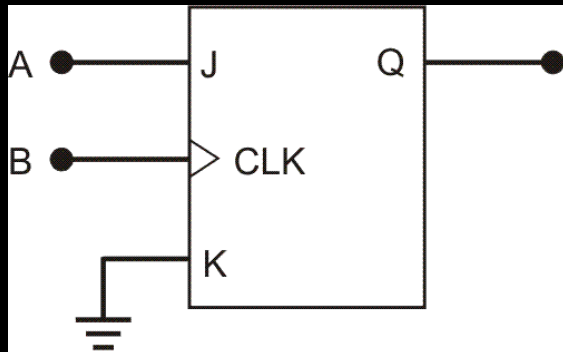




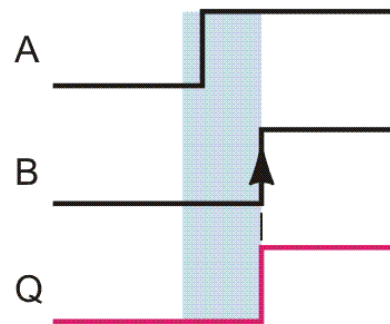
# Exemplos



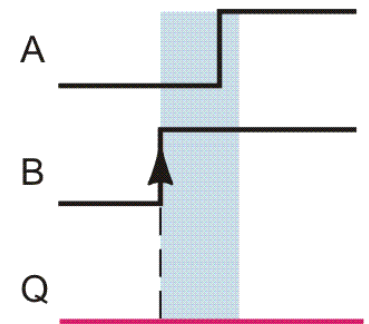
## Detecção de seqüência de entrada



(a)



(b) A vai para o nível ALTO antes de B

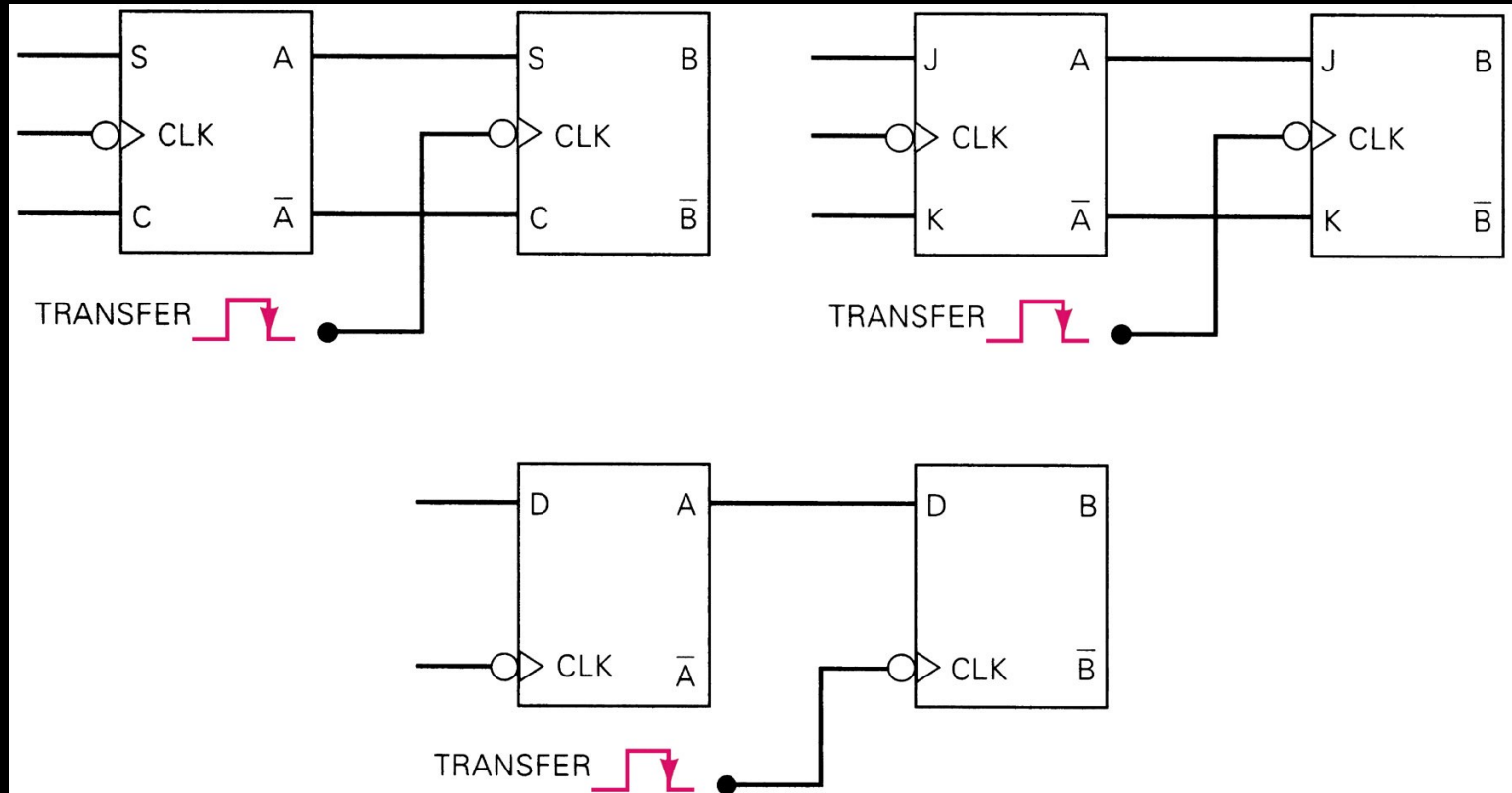


(c) B vai para o nível ALTO antes de A

# Exemplos



## Transferência entre FF



# Circuitos Sequencias



## ► Registradores

### Registrador de carga paralela

