

Latches e Flip-Flops

Nikolas Libert

nikolaslibert@utfpr.edu.br

Aula 09

Eletrônica Digital ET52C

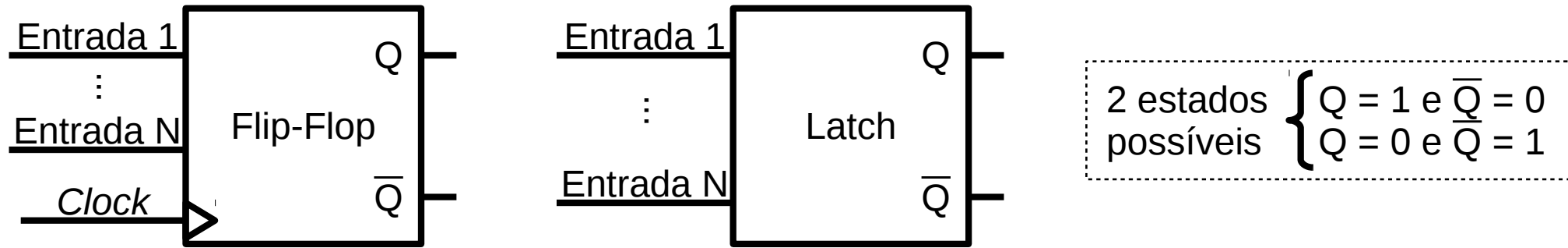
Tecnologia em Automação Industrial



Latches e Flip-Flops

- Necessários para construção de circuitos sequenciais.
- Saídas possuem dois estados possíveis.
- Elementos com memória.
- Criados pela associação de portas lógicas com realimentação:
 - Saídas conectadas à entrada.

■ Estrutura básica:



- Saídas Q e \overline{Q} indicam o estado atual do componente.
- Uma sequência correta de sinais aplicados nas entradas gera uma mudança de estado.

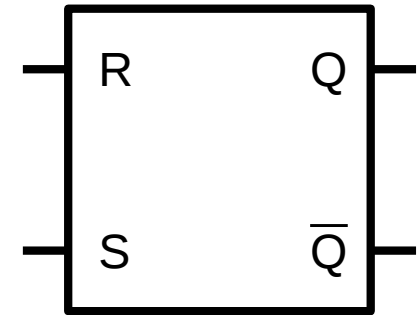
■ Flip-Flops (síncronos): mudanças de estado cadenciadas por borda de *clock*.

■ Latches (assíncronos): mudanças de estado instantâneas.

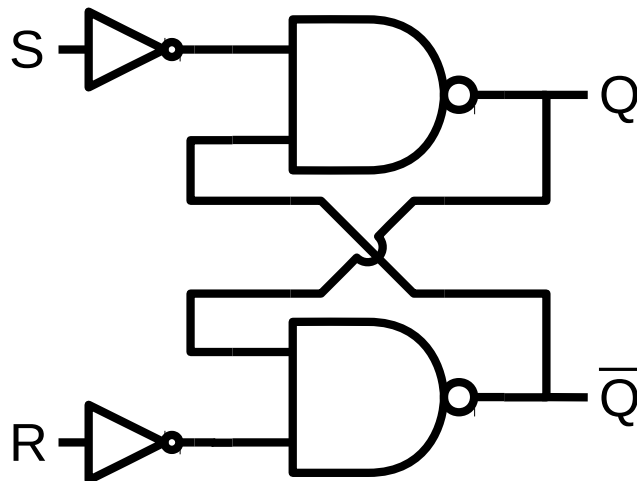
Latch RS Básico

■ Possui apenas duas entradas:

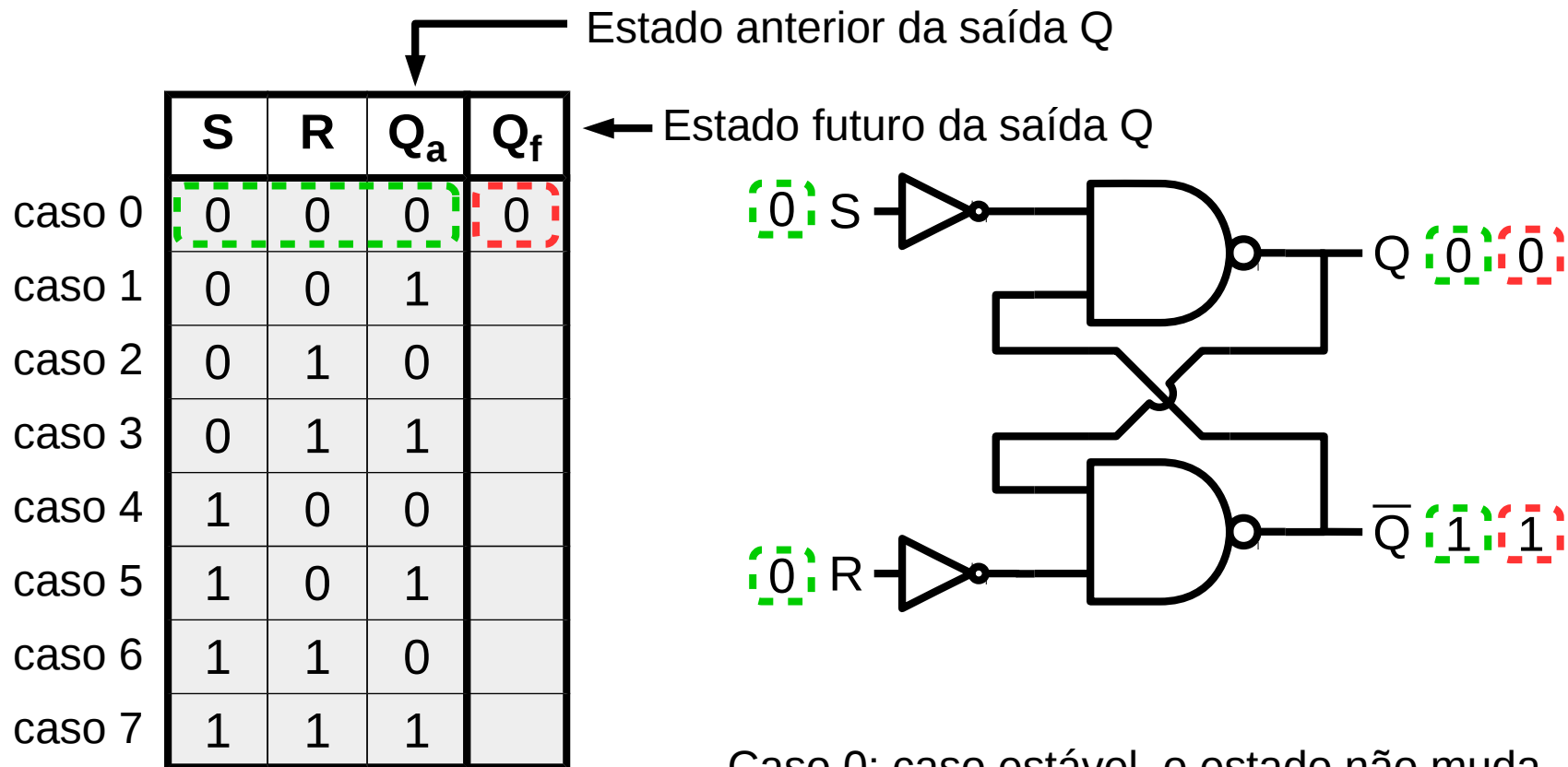
- Reset (R).
- Set (S).



■ Pode ser montado com portas Não E e inversoras.



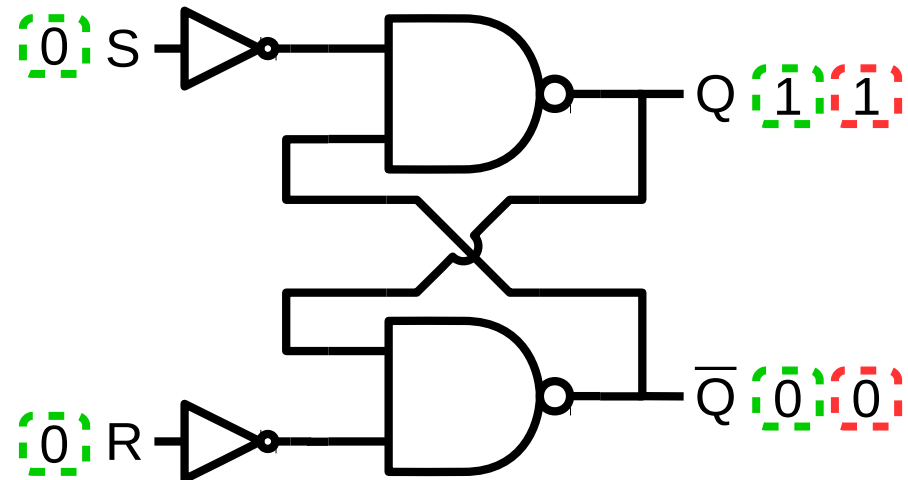
- A saída do circuito depende do valor da saída anterior.
 - Na tabela verdade, a saída anterior é colocada como variável de entrada.



Latch RS Básico

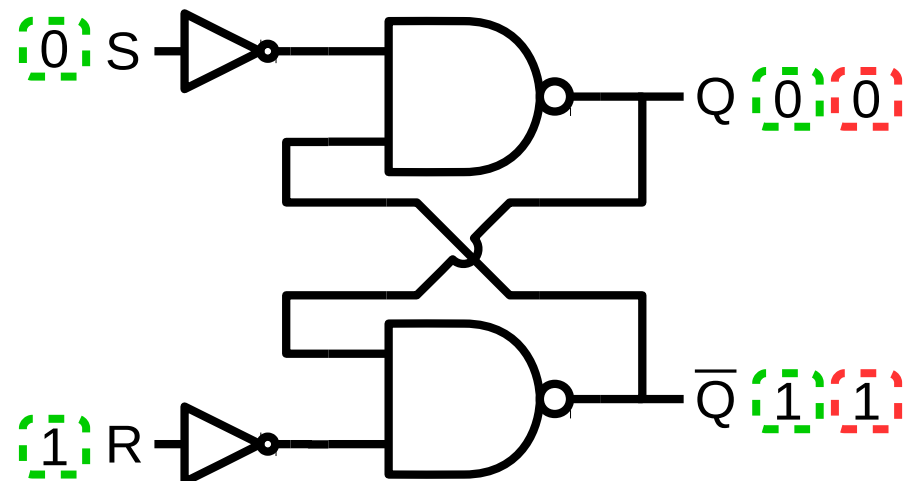
Caso 1

caso estável:
o estado de saída
não muda.



Caso 2

caso estável:
o estado de saída
não muda.

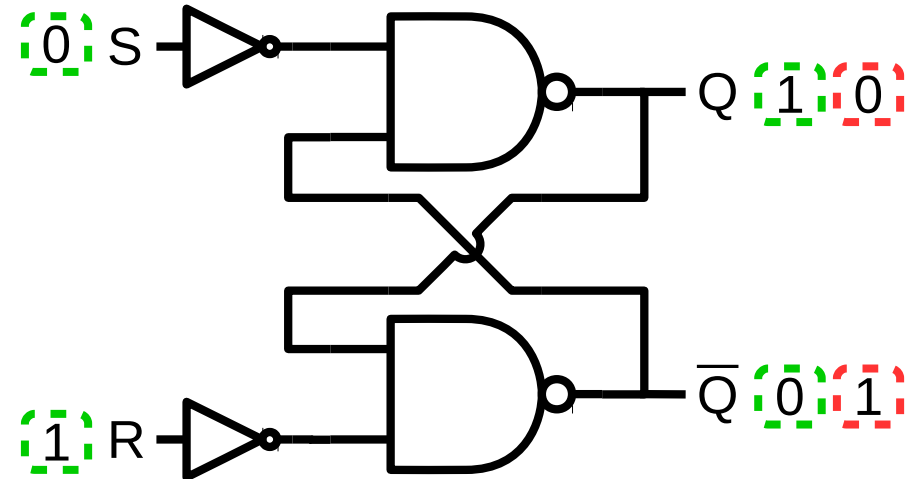


	S	R	Q_a	Q_f
caso 0	0	0	0	0
caso 1	0	0	1	1
caso 2	0	1	0	0
caso 3	0	1	1	
caso 4	1	0	0	
caso 5	1	0	1	
caso 6	1	1	0	
caso 7	1	1	1	

Latch RS Básico

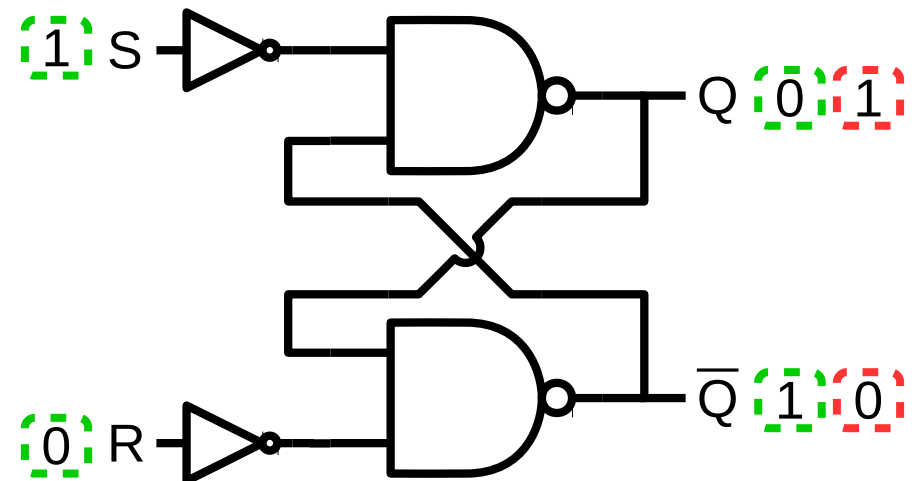
Caso 3

comutação no estado de saída.



Caso 4

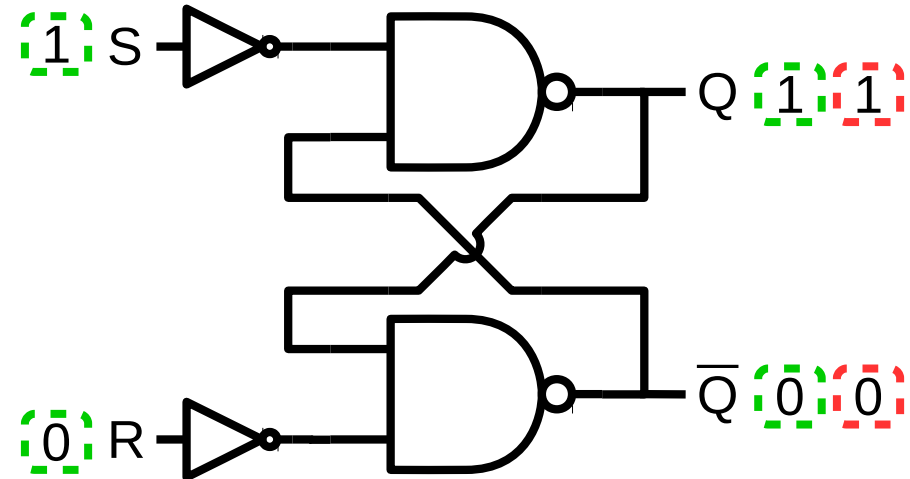
comutação no estado de saída.



	S	R	Q_a	Q_f
caso 0	0	0	0	0
caso 1	0	0	1	1
caso 2	0	1	0	0
caso 3	0	1	1	0
caso 4	1	0	0	1
caso 5	1	0	1	
caso 6	1	1	0	
caso 7	1	1	1	

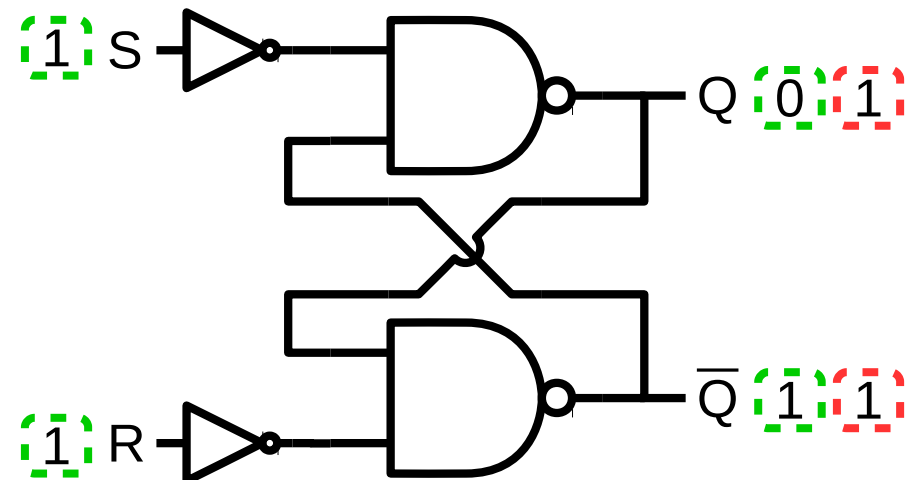
Caso 5

caso estável:
o estado de saída
não muda.



Caso 6

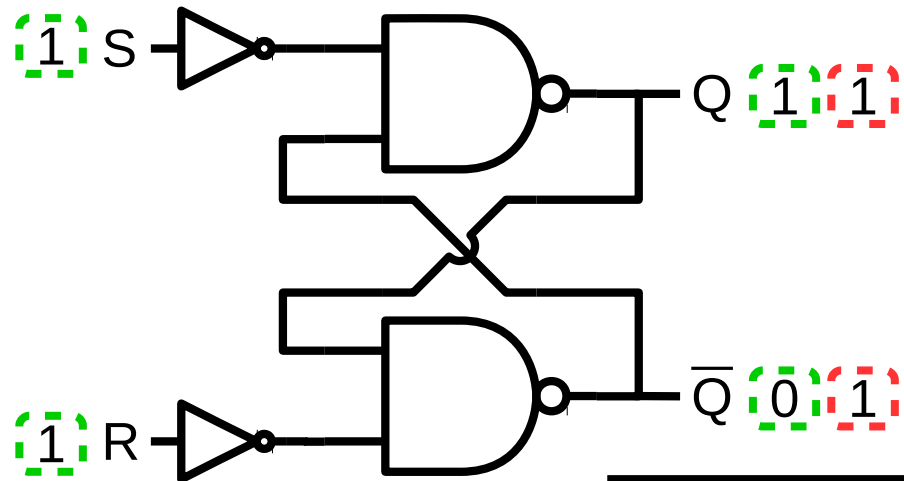
caso indesejável:
as duas saídas
ficam com mesmo
valor.



deveria sempre
haver uma saída Q
e \bar{Q} .

	S	R	Q_a	Q_f
caso 0	0	0	0	0
caso 1	0	0	1	1
caso 2	0	1	0	0
caso 3	0	1	1	0
caso 4	1	0	0	1
caso 5	1	0	1	1
caso 6	1	1	0	1
caso 7	1	1	1	

Latch RS Básico



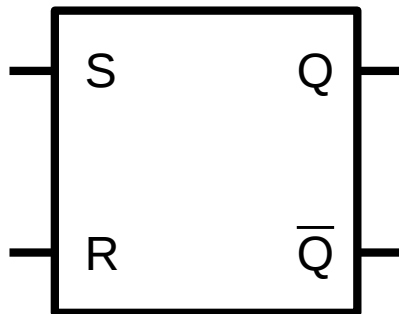
Caso 7

Indesejável: as duas saídas ficam com mesmo valor.
deveria sempre haver uma saída Q e \bar{Q} .

	S	R	Q_a	Q_f	\bar{Q}_f	
caso 0	0	0	0	0	1	} Q_f continua igual a Q_a
caso 1	0	0	1	1	0	
caso 2	0	1	0	0	1	} Q_f é fixado em 0
caso 3	0	1	1	0	1	
caso 4	1	0	0	1	0	} Q_f é fixado em 1
caso 5	1	0	1	1	0	
caso 6	1	1	0	1	1	} Não permitido
caso 7	1	1	1	1	1	

■ Tabela verdade resumida:

- Nível alto no pino S (*Set*): Saída muda para nível alto.
- Nível alto no pino R (*Reset*): Saída muda para nível baixo.
- Nenhuma entrada ativada: Saída permanece como estava.
- As duas entradas ativadas: comportamento indesejado ($Q=\overline{Q}=1$).

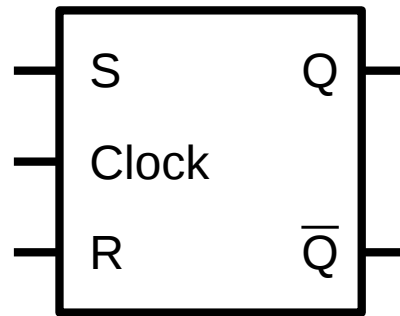


S	R	Q_f
0	0	Q_a
0	1	0
1	0	1
1	1	X

Circuito funciona como
memória de 1 bit

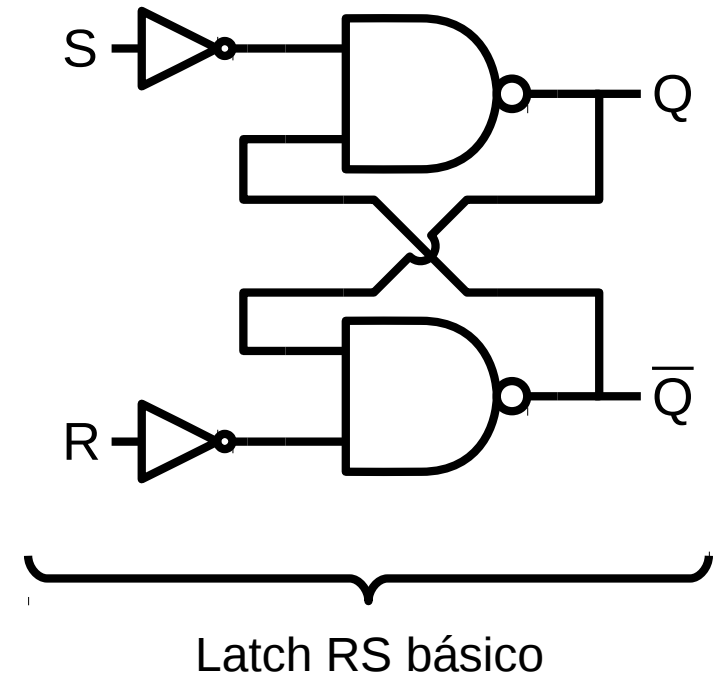
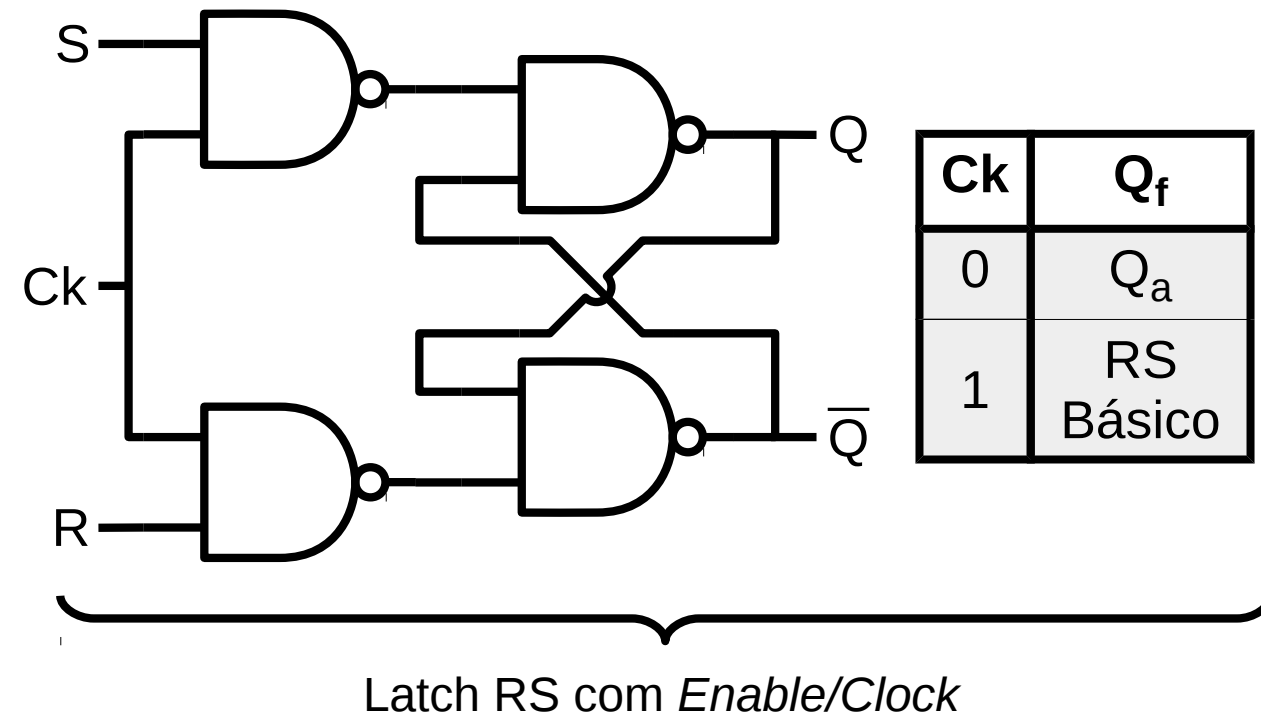
Latch RS com *Enable/Clock*

- Possui uma entrada de *enable/clock* adicional:
 - Alterações nos pinos R e S só refletem na saída se simultaneamente a entrada de *Clock* estiver em nível alto.



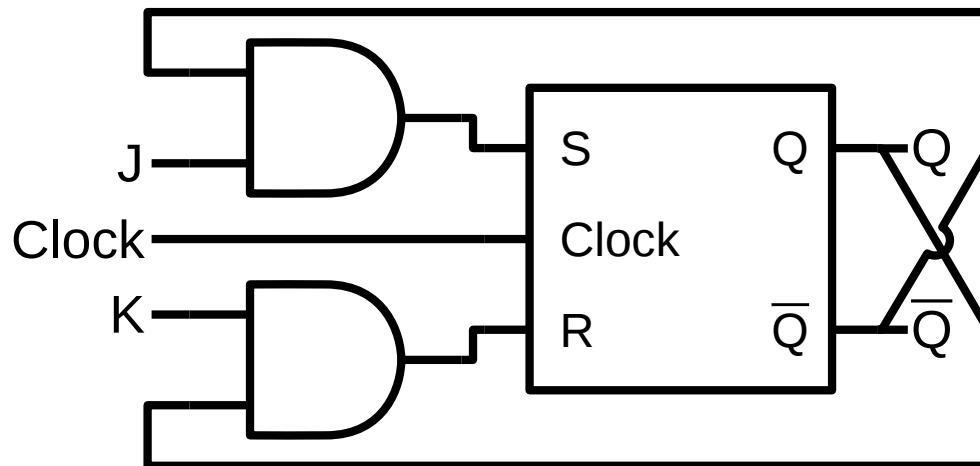
Latch RS com *Enable/Clock*

- Se a entrada de *Clock* estiver em nível baixo, S e R não podem interferir nas saídas.
- Se a entrada de *Clock* estiver em nível alto, o circuito funciona como um Latch RS normal.



Latch JK

- Pode ser obtido por meio de um latch RS com realimentação adicional.

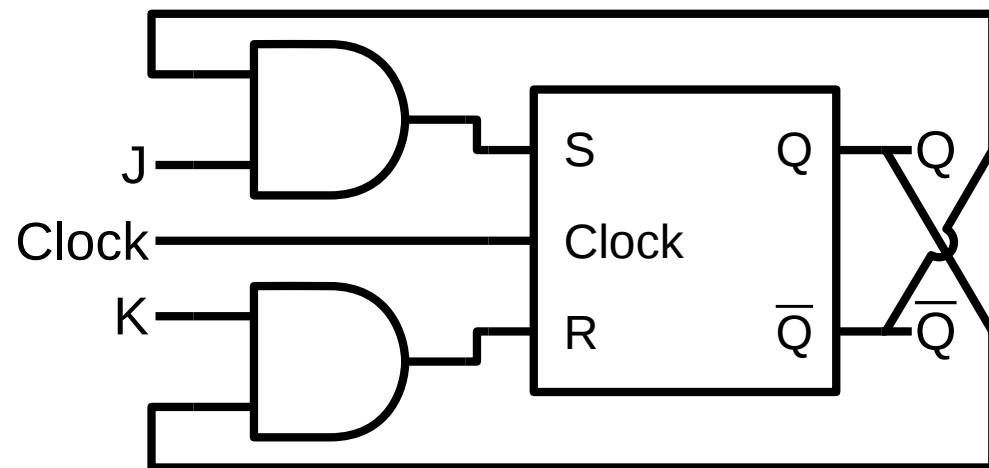
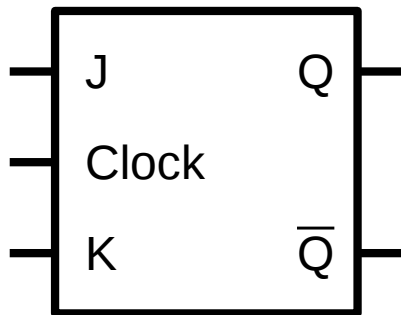


- A entrada S só será acionada se J estiver em nível alto e Q em nível baixo.

- A entrada R só será acionada se K estiver em nível alto e Q estiver em nível alto.

- O caso não permitido ($R=S=Q=\bar{Q}=1$) não ocorrerá mais. Será impossível ativar S e R simultaneamente.

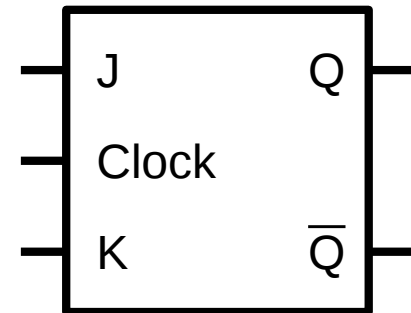
- Considerando entrada de *clock* sempre igual a 1, ache a tabela verdade do latch JK.



	J	K	Q_a	\bar{Q}_a	S	R	Q_f	\bar{Q}_f
caso 0	0	0	0					
caso 1	0	0	1					
caso 2	0	1	0					
caso 3	0	1	1					
caso 4	1	0	0					
caso 5	1	0	1					
caso 6	1	1	0					
caso 7	1	1	1					

- Considerando entrada de *clock* sempre igual a 1, ache a tabela verdade do latch JK.

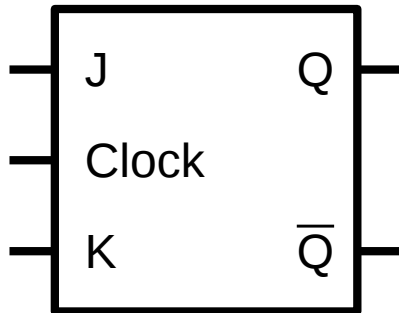
	J	K	Q_a	\overline{Q}_a	S	R	Q_f	\overline{Q}_f	
caso 0	0	0	0	1	0	0	0	1	} Q_f continua igual a Q_a
caso 1	0	0	1	0	0	0	1	0	
caso 2	0	1	0	1	0	0	0	1	} Q_f é fixado em 0
caso 3	0	1	1	0	0	1	0	1	
caso 4	1	0	0	1	1	0	1	0	} Q_f é fixado em 1
caso 5	1	0	1	0	0	0	1	0	
caso 6	1	1	0	1	1	0	1	0	} Q_f é o oposto de seu estado anterior
caso 7	1	1	1	0	0	1	0	1	



No caso em que $J=K=1$, as entradas devem ser ativadas por pouco tempo, caso contrário a saída oscilará entre 0 e 1.

■ Tabela verdade resumida:

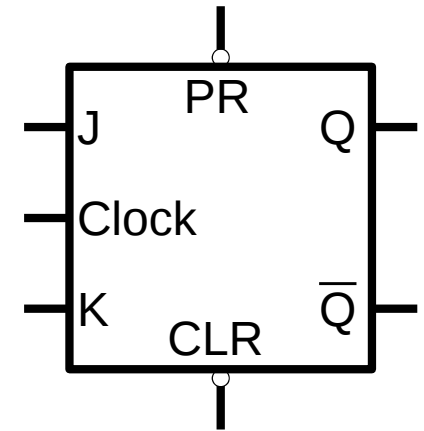
- Nível alto no pino J: Saída muda para nível alto.
- Nível alto no pino K: Saída muda para nível baixo.
- Nenhuma entrada ativada: Saída permanece como estava.
- Nível alto nos pinos J e K: Nível lógico da saída é invertido.



J	K	Q_f
0	0	Q_a
0	1	0
1	0	1
1	1	$\overline{Q_a}$

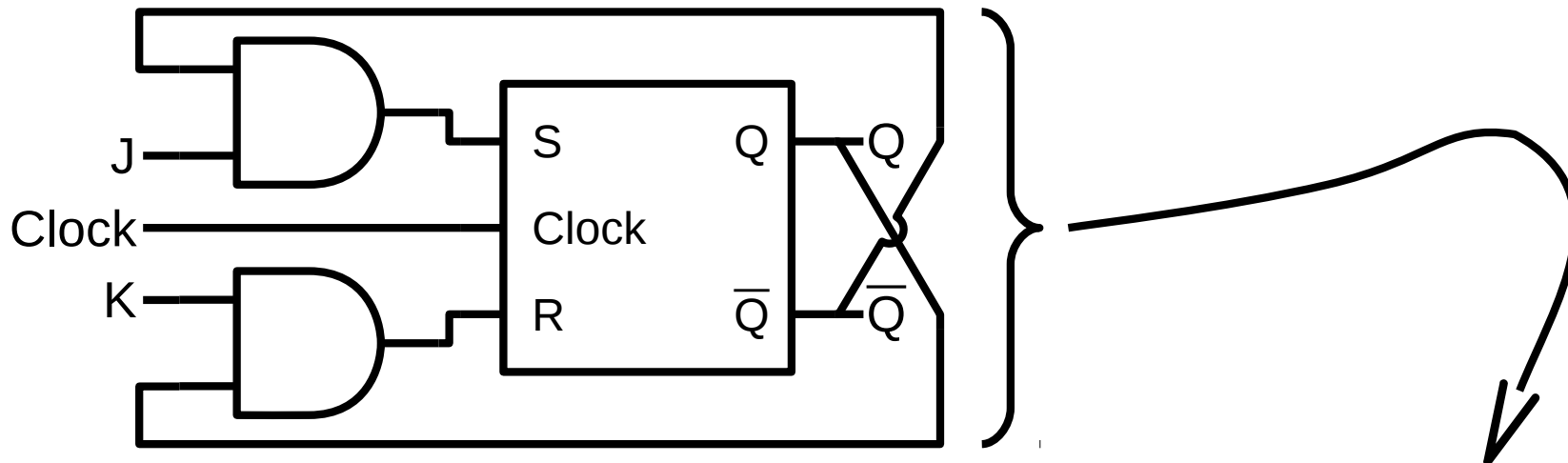
Situação indesejada do Latch RS ($R=S=1$) foi eliminada.

Latch JK com *Preset* e *Clear*

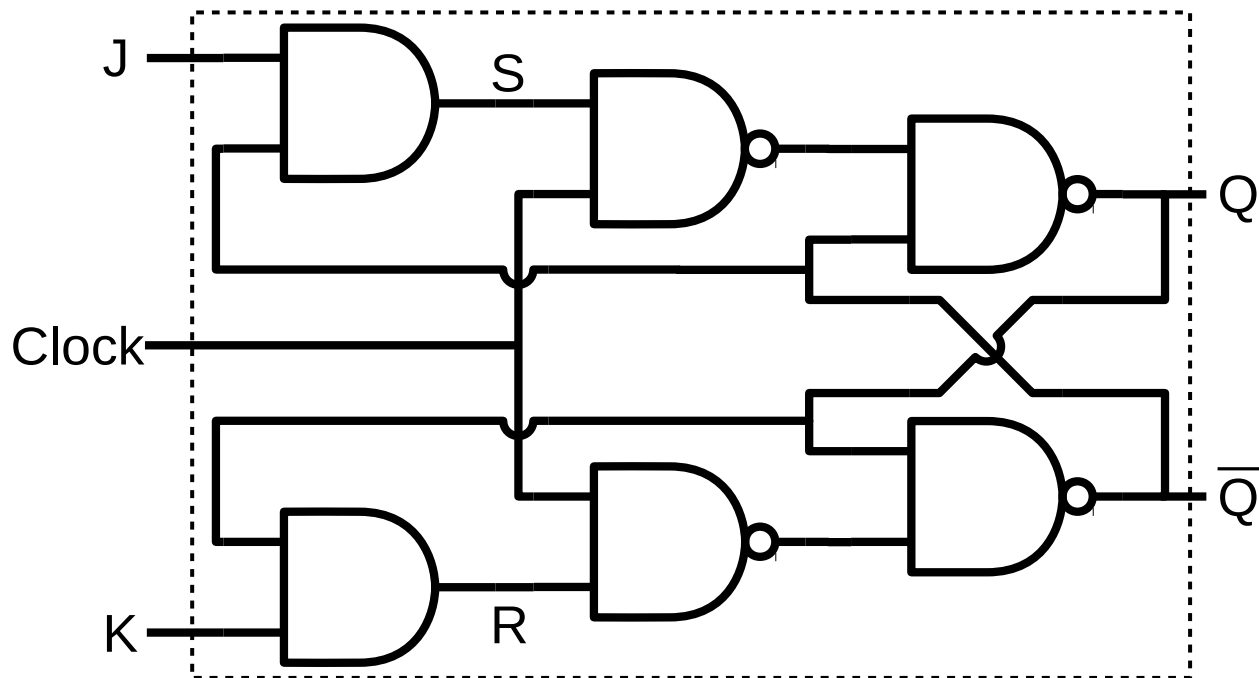


- Possui duas entradas adicionais.
 - Se a entrada *Preset* for acionada, Q inicia em nível alto.
 - Se a entrada *Clear* for acionada, Q inicia em nível baixo.
- *Preset* e *Clear* serão sempre entradas assíncronas.
 - Alteram a saída instantaneamente, independente do sinal de *clock*.

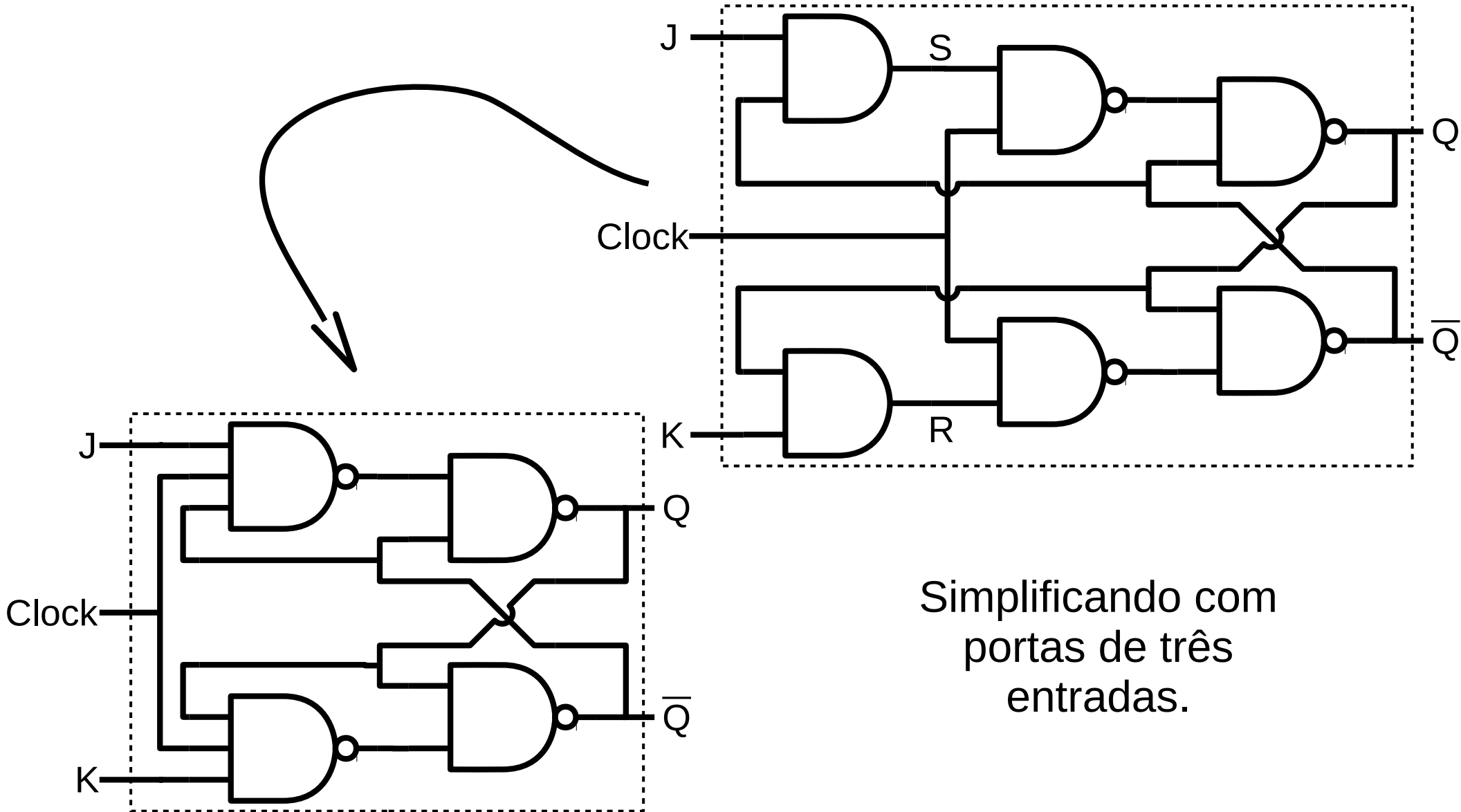
■ Representação do latch JK com portas lógicas:



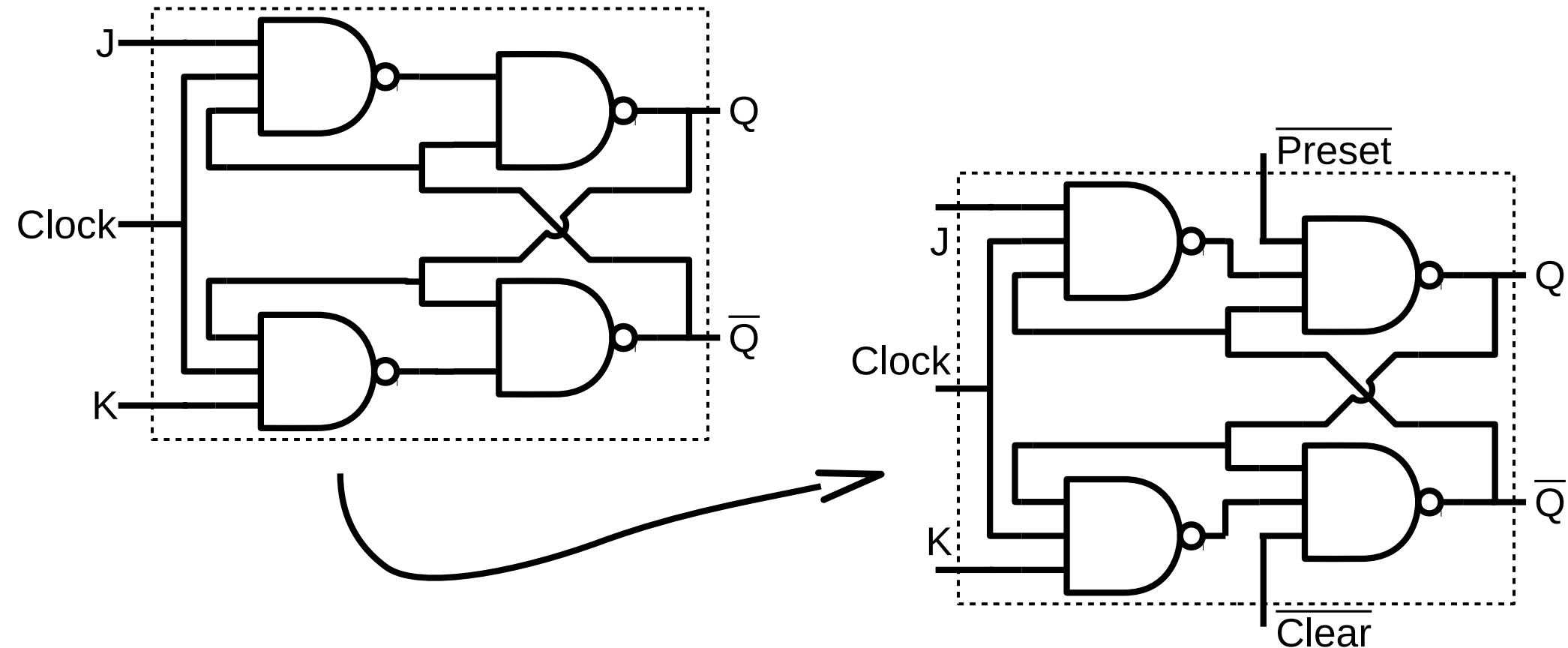
Substituindo o bloco do latch RS por seu circuito lógico.



■ Representação do latch JK com portas lógicas:



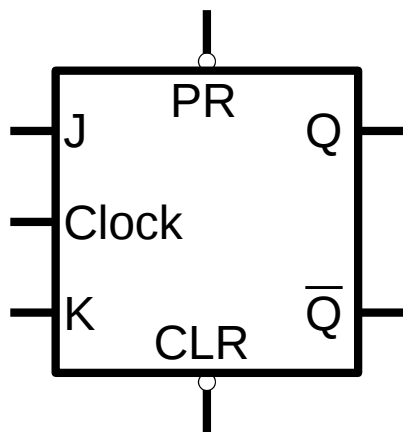
■ Adição dos pinos de *Preset* e *Clear*:



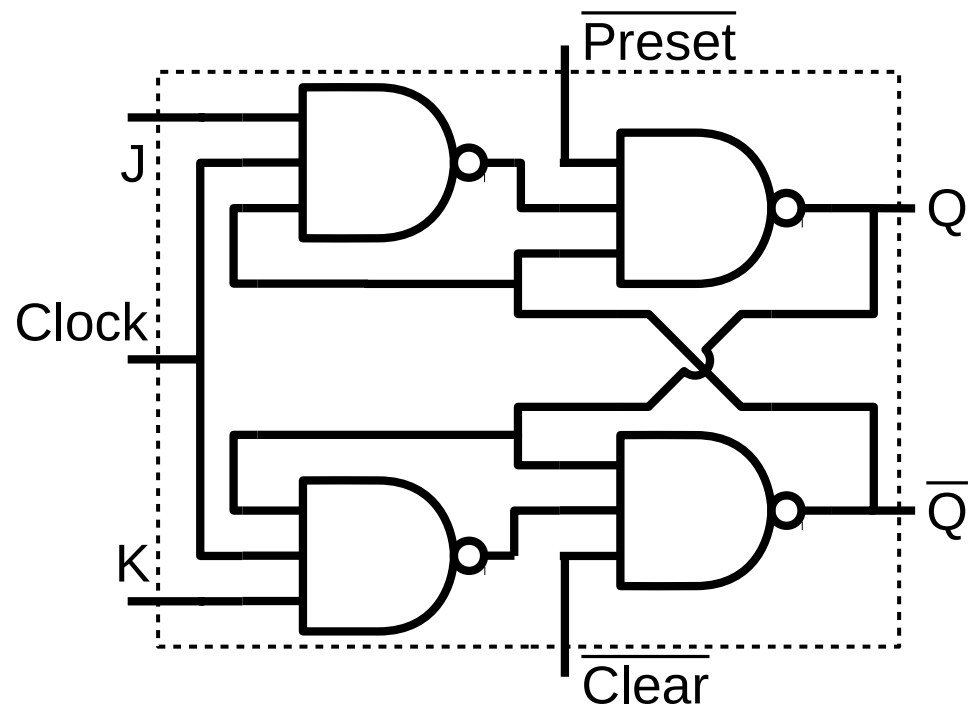
- Para $\overline{\text{Preset}}$ ou $\overline{\text{Clear}}$ em nível alto, o circuito funcionará como antes.
- Para $\overline{\text{Preset}}$ em nível baixo, a saída Q ficará em nível alto.
- Para $\overline{\text{Clear}}$ em nível baixo, a saída \bar{Q} ficará em nível alto.

Latch JK com *Preset* e *Clear*

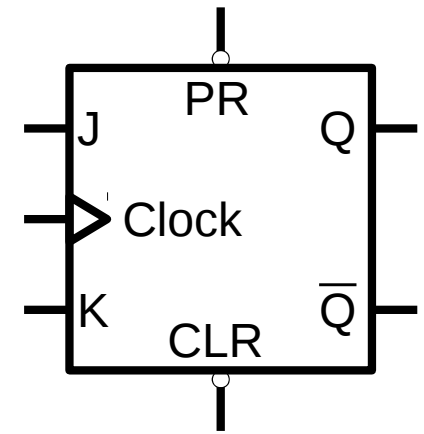
■ Latch JK com *Preset* e *Clear*



$\overline{\text{Clr}}$	$\overline{\text{Pr}}$	Q_f
0	0	não permitido
0	1	0
1	0	1
1	1	funcionamento normal



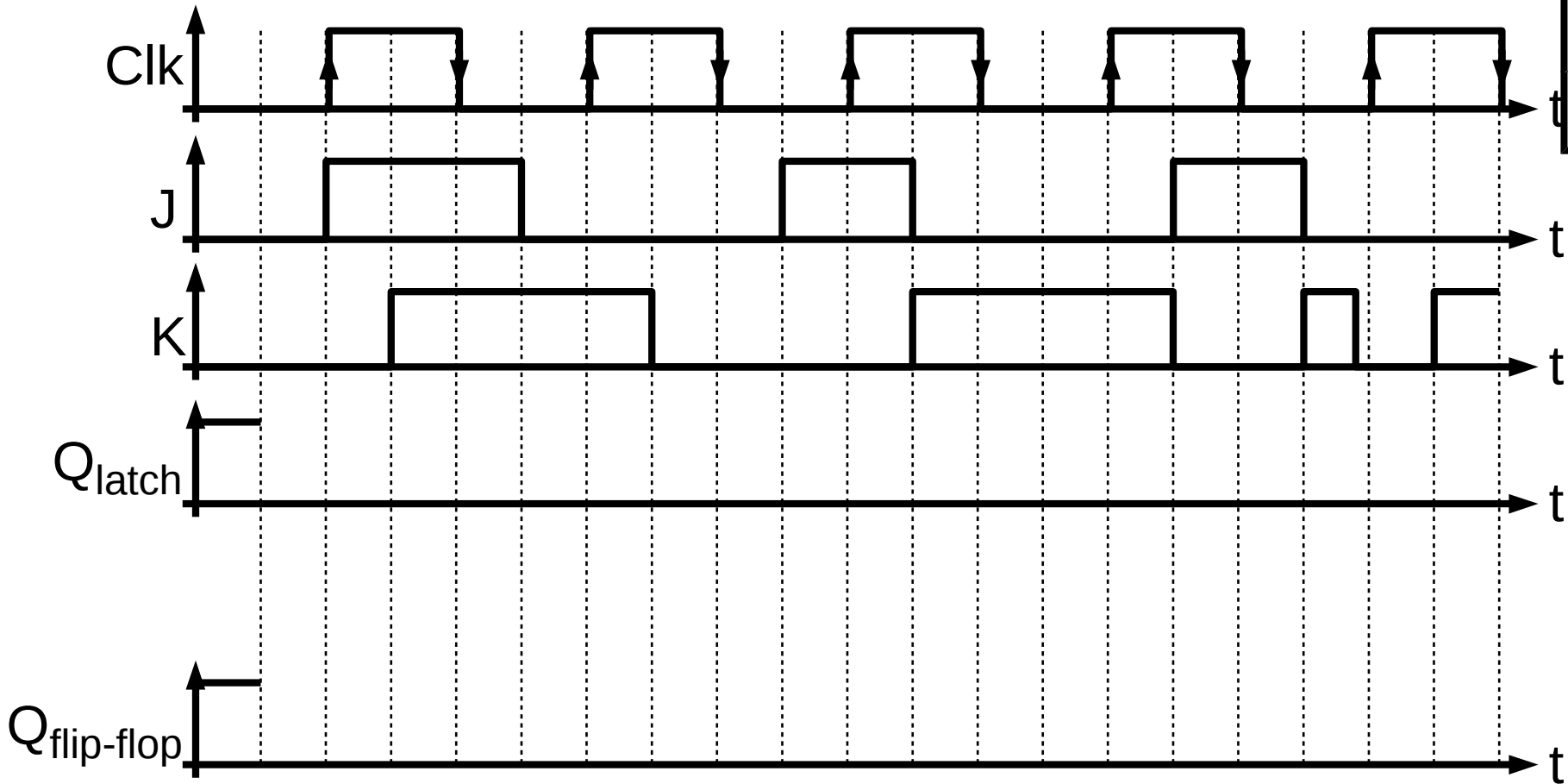
Flip-Flop JK mestre escravo



- Quando o sinal de *clock* do latch JK está em nível alto, a saída reflete alterações nas entradas instantaneamente.
- No flip-flop JK, as saídas podem mudar apenas nas bordas de *clock*.
- São elementos síncronos.

■ Exemplo: Latch JK x Flip-Flop JK

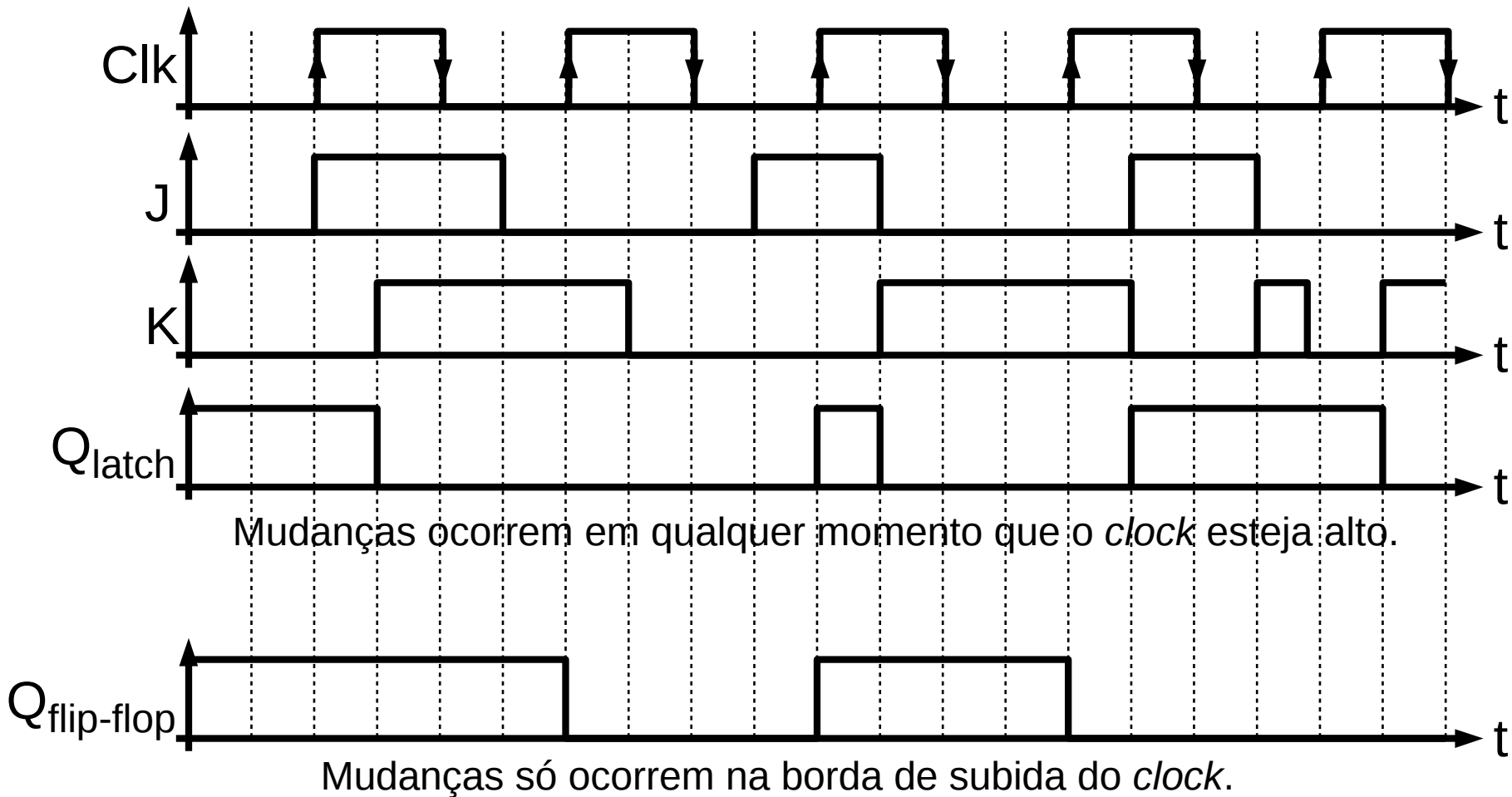
– Ache a forma de onda das saídas.



J	K	Q _f
0	0	Q _a
0	1	0
1	0	1
1	1	$\overline{Q_a}$

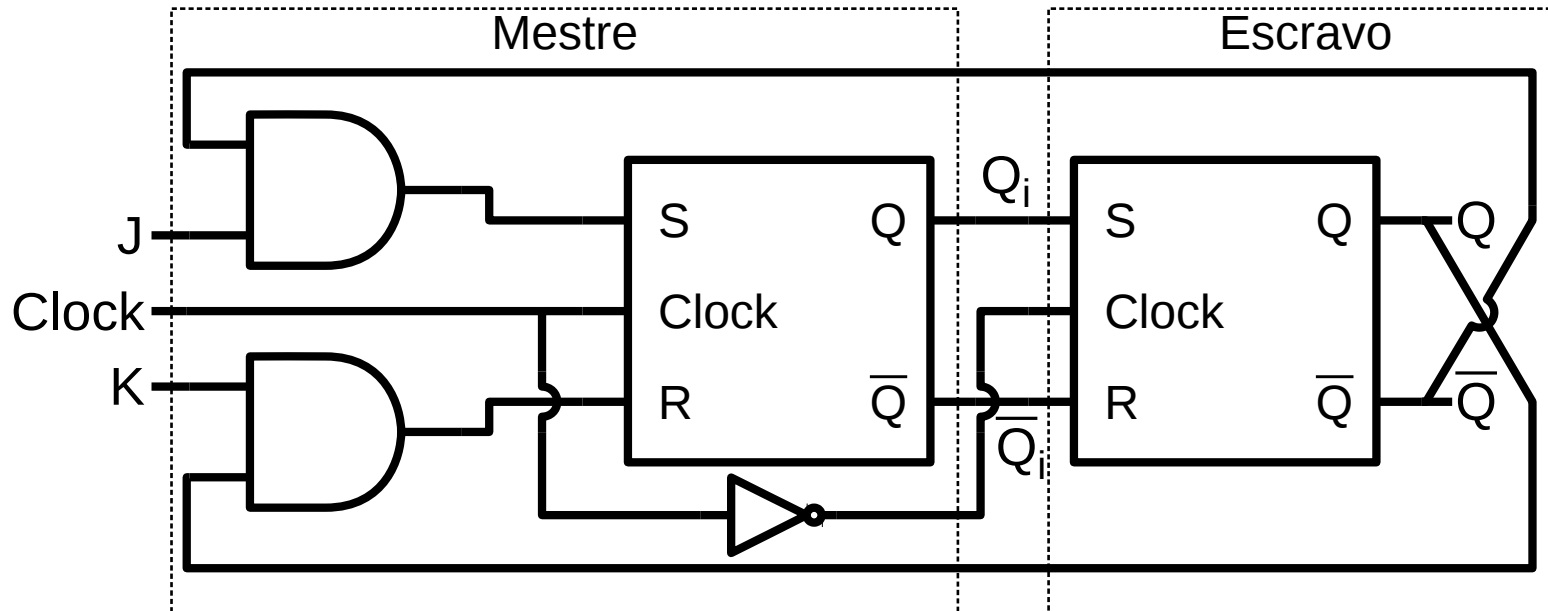
■ Exemplo: Latch JK x Flip-Flop JK

- Ache a forma de onda das saídas.



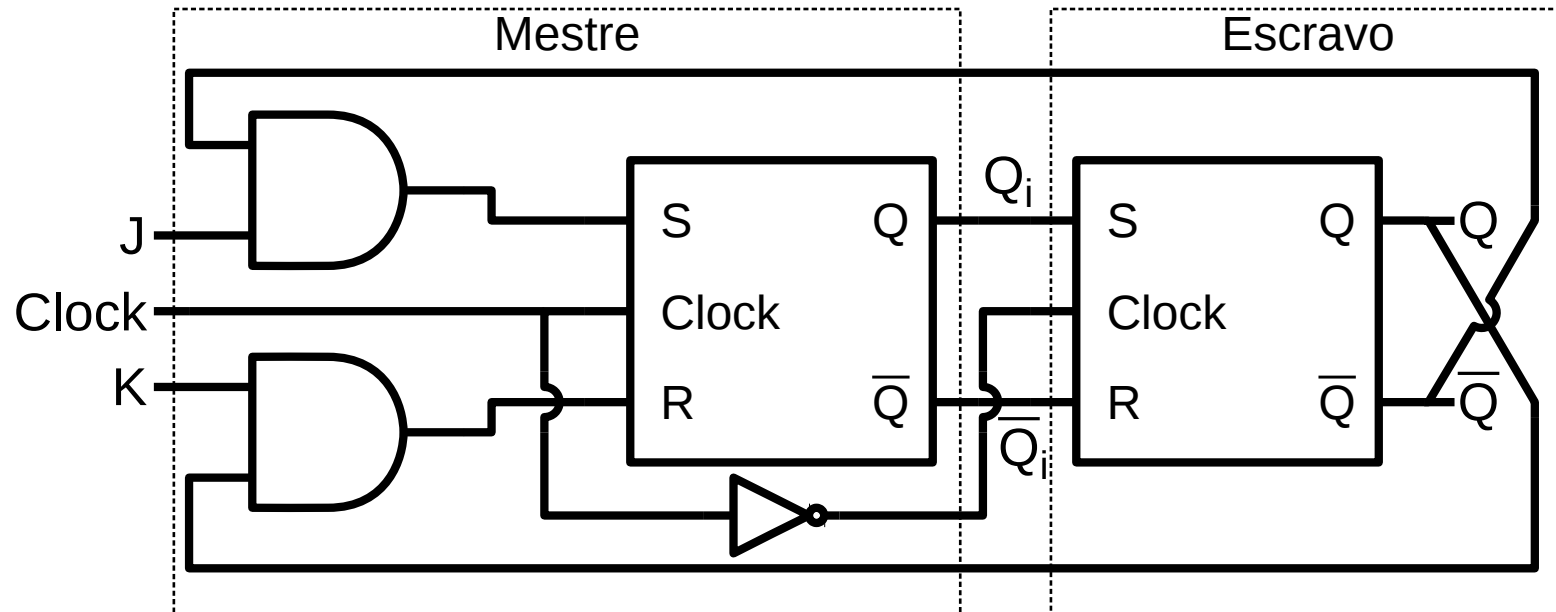
■ Flip-Flop JK mestre escravo

- Possui saída intermediária Q_i .
- Saída intermediária é alterada pelo estágio Mestre no nível alto de *clock*.
- Saída intermediária é copiada para o Escravo no nível baixo de *clock*.

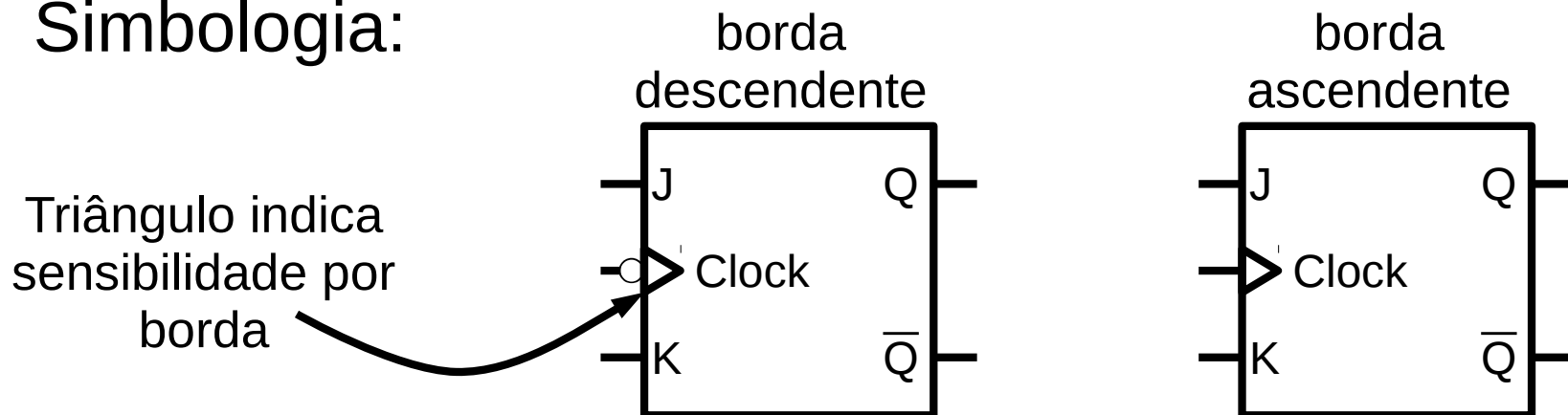


Flip-Flop JK mestre escravo

- Mudanças na saída principal serão observadas na borda descendente de *clock*.

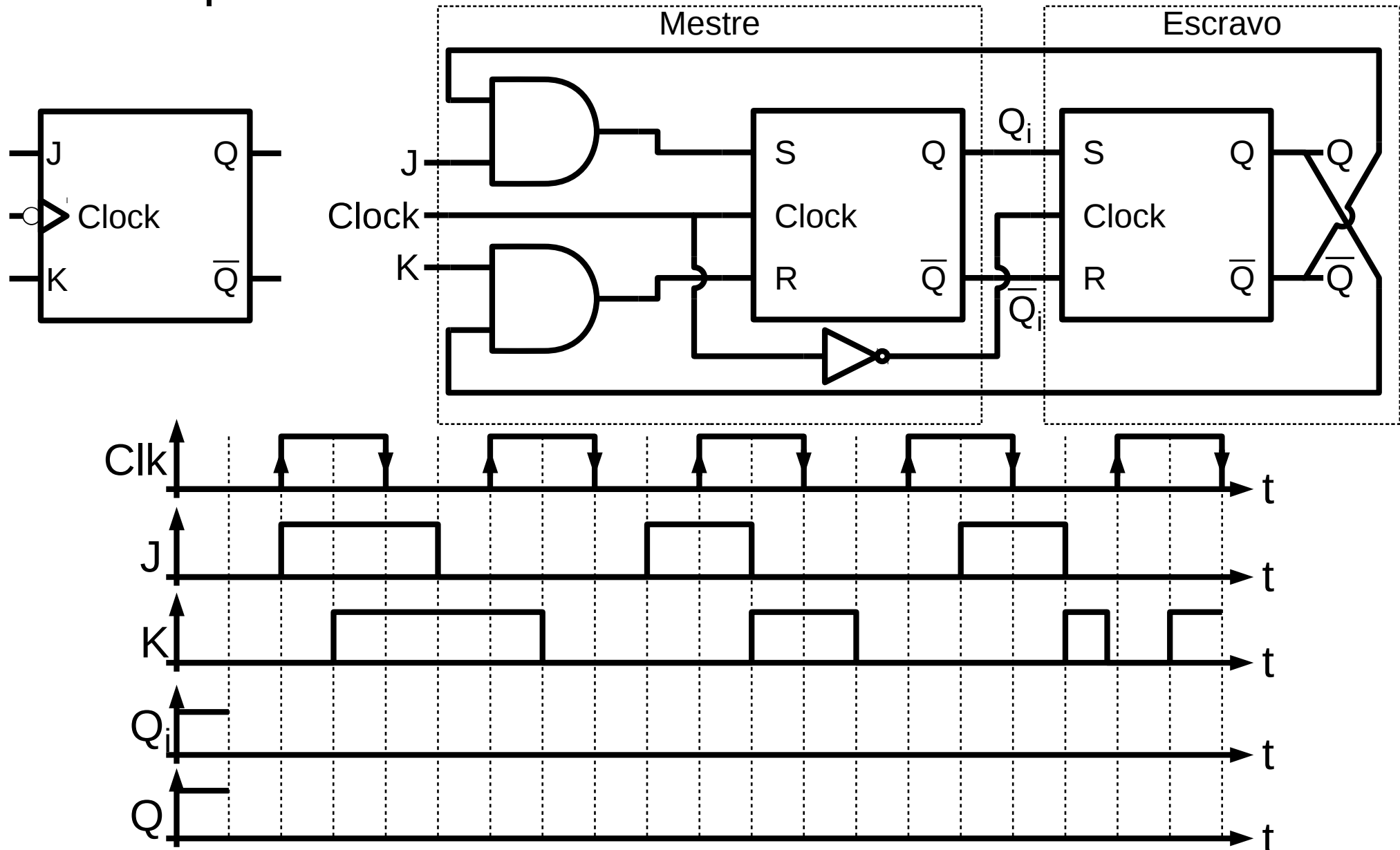


- Simbologia:



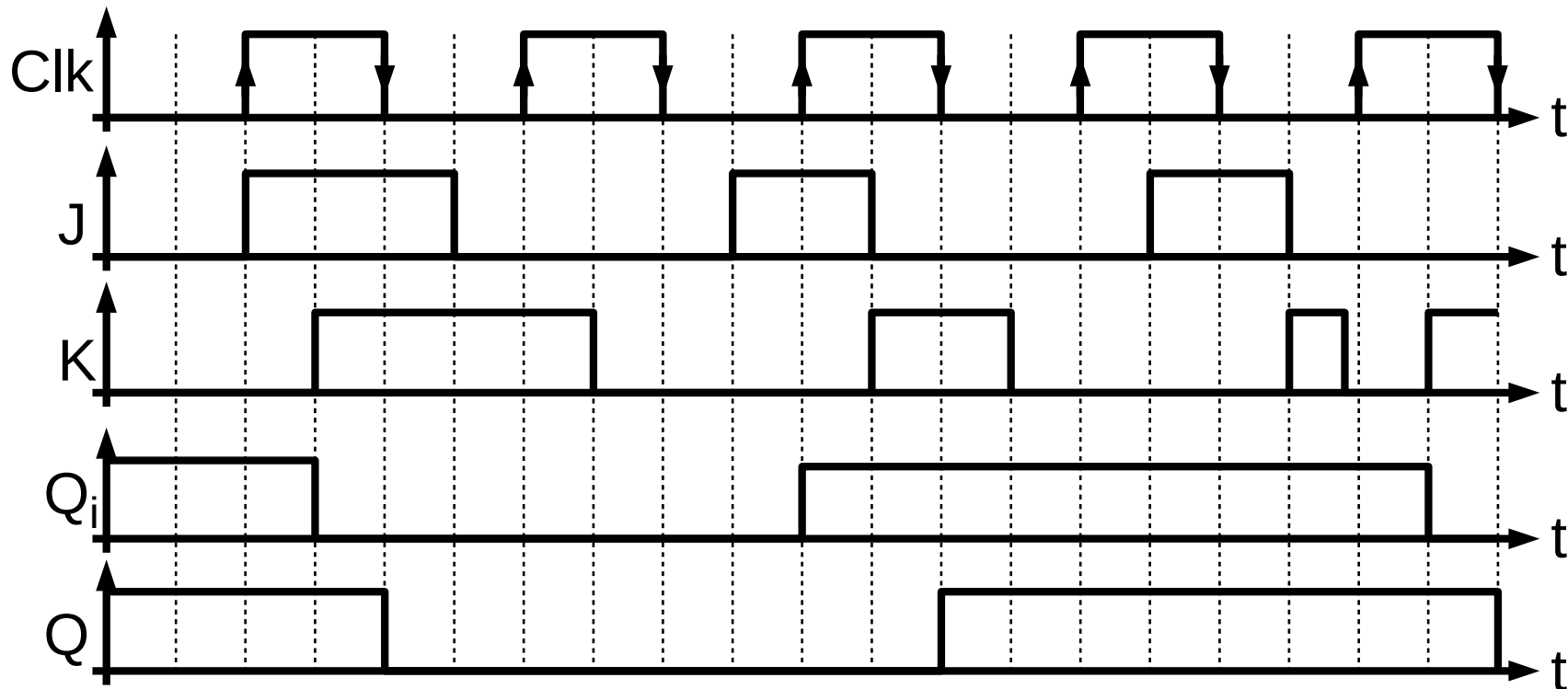
Flip-Flop JK mestre escravo

■ Exemplo: Ache a forma de onda na saída.



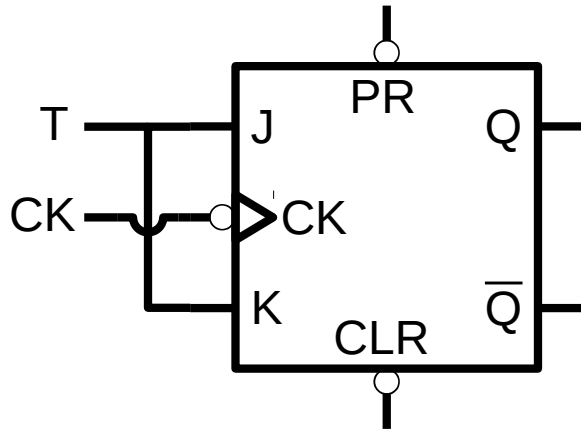
Flip-Flop JK mestre escravo

- Exemplo: Ache a forma de onda na saída.



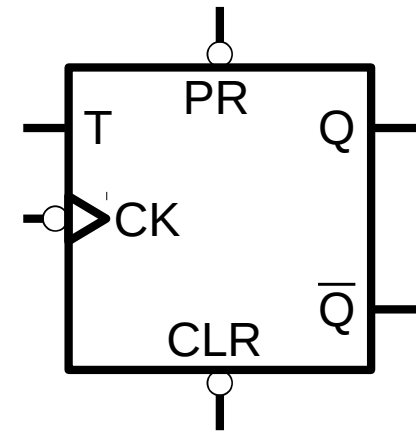
Flip-Flop T (*Toggle*)

- Flip-Flop JK com entradas J e K interligadas.



J	K	Q_f
0	0	Q_a
θ	\pm	θ
\pm	θ	\pm
1	1	$\overline{Q_a}$

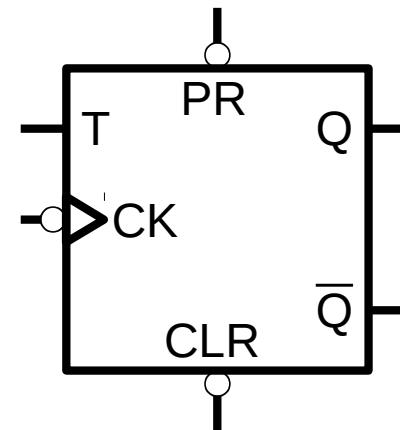
} Estas situações deixam de existir



■ Flip-Flop T:

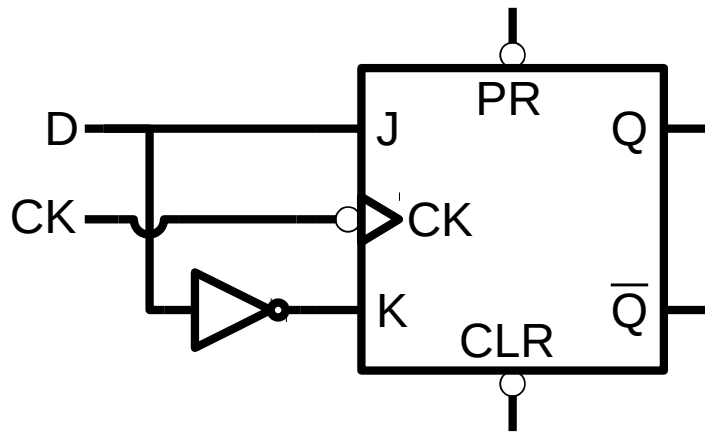
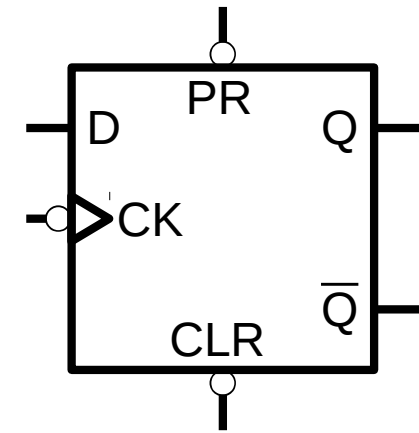
- Não é encontrado na série de CIs convencionais. Deve ser montado à partir de um JK.
- Utilizados na construção de contadores assíncronos.
- Se T estiver ativo, a saída é complementada a cada descida de *clock*.

T	Q_f
0	Q_a
1	$\overline{Q_a}$



Flip-Flop D (*Data*)

- Flip-Flop JK com entradas J e K interligadas por inversor.



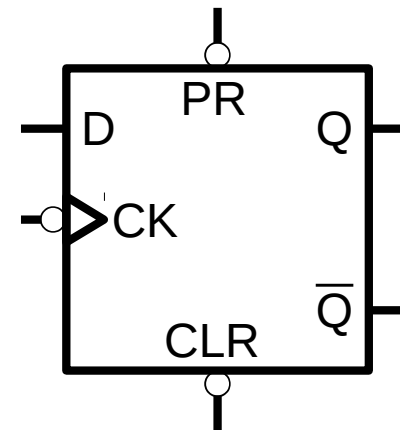
J	K	Q_f
θ	θ	Θ_a
0	1	0
1	0	1
\pm	\pm	$\bar{\Theta}_a$

Estas situações deixam de existir

■ Flip-Flop D:

- Utilizados na construção de registradores de deslocamento.
- Armazena o dado inserido na entrada D a cada descida de *clock*.

D	Q_f
0	0
1	1



- IDOETA, I. V., CAPUANO, F. G. *Elementos de Eletrônica Digital*, 41ª Edição, Érica, São Paulo, 2013.
- PEDRONI, Volnei. *Eletrônica Digital Moderna e VHDL*, Campus, Rio de Janeiro, 2010.