

Latches e Flip-flops (Parte 1) – ELD22102

Engenharia Eletrônica

Prof. Reginaldo Steinbach

reginaldo.steinbach@ifsc.edu.br

- Sequencial x combinacional
- Circuitos sequenciais assíncronos e síncronos
- Latches e flip-flops
- Latch SR
- Flip-flop SR
- Flip-flop JK
- Flip-flop D
- Entradas assíncronas

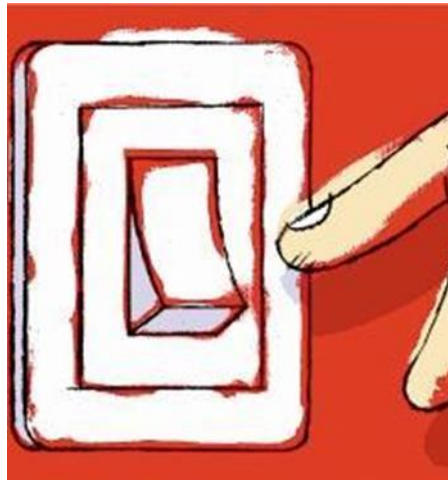
Referência:

TOCCI, *Sistemas digitais: princípios e aplicações (Capítulo 5)*.
11ª. edição. Pearson Prentice Hall, 2011

- **Sequencial x combinacional**
- Circuitos sequenciais assíncronos e síncronos
- Latches e flip-flops
- Latch SR
- Flip-flop SR
- Flip-flop JK
- Flip-flop D
- Flip-flop T
- Entradas assíncronas

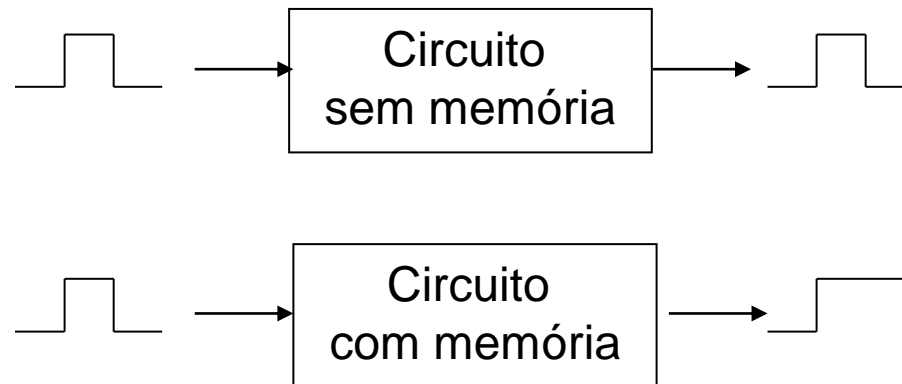
Sequencial x Combinacional

- Qualquer dispositivo ou circuito que tem dois estados estáveis é dito **biestável** como, por exemplo, uma chave de conexão. Ela pode estar aberta ou fechada, dependendo da posição da tecla. Esta chave possui uma memória, visto que ela permanecerá em um estado definido até que alguém ou algo mude a posição da tecla.



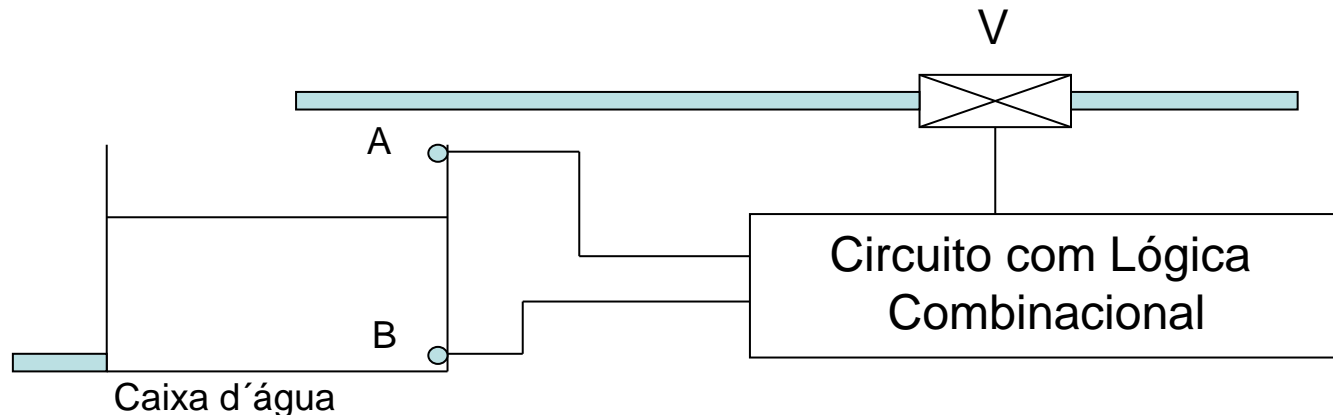
Sequencial x Combinacional

- Quando um sinal de entrada é aplicado num dispositivo, a saída muda em resposta à entrada. Quando o sinal de entrada é removido, a saída retorna ao seu estado original. Este dispositivo não exibe a propriedade de memória, já que sua saída volta ao estado anterior.
- Existem dispositivos e circuitos digitais que possuem memória, onde quando um sinal de entrada é aplicado, a saída poderá mudar seu estado, mas permanecerá neste estado mesmo após a entrada ter sido removida. Esta propriedade de reter sua resposta a uma entrada momentânea é chamada **memória**.



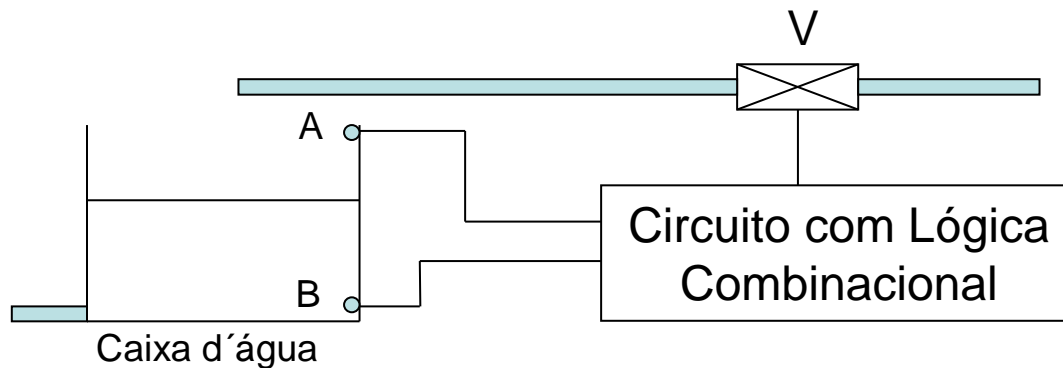
Sequencial x Combinacional

- Nos sistemas digitais baseados em lógica combinacional. Os estados de suas saídas são dependentes apenas dos estados presentes (instantâneos) das entradas, logo, estes sistemas não conseguem lidar com a variável tempo e perceber sequências de eventos, portanto, são incapazes de resolver qualquer problema que envolva a noção de tempo.
- Um exemplo: Pretende-se que a válvula V seja aberta quando o sensor B estiver descoberto e só volte a ser fechada quando o sensor A estiver coberto.



Sequencial x Combinacional

- Se tentarmos resolver o problema usando lógica combinacional, poderemos sugerir a seguinte tabela da verdade:



Usando a convenção:

- Sensores A e B: 0 – descoberto
1 – coberto
- Válvula V : 0 – fechada
1 – aberta

A B	V
0 0	1
0 1	0
1 0	X (impossível)
1 1	0

Observa-se que a água não atingirá o sensor A, pois quando a água encobrir o sensor B, a válvula já se fechará. Portanto não funciona como era desejado. Por outro lado, de V for aberto toda vez que o sensor A estiver descoberto, haverá um abre/fecha excessivo da válvula, podendo danificar a mesma.

Sequencial x Combinacional

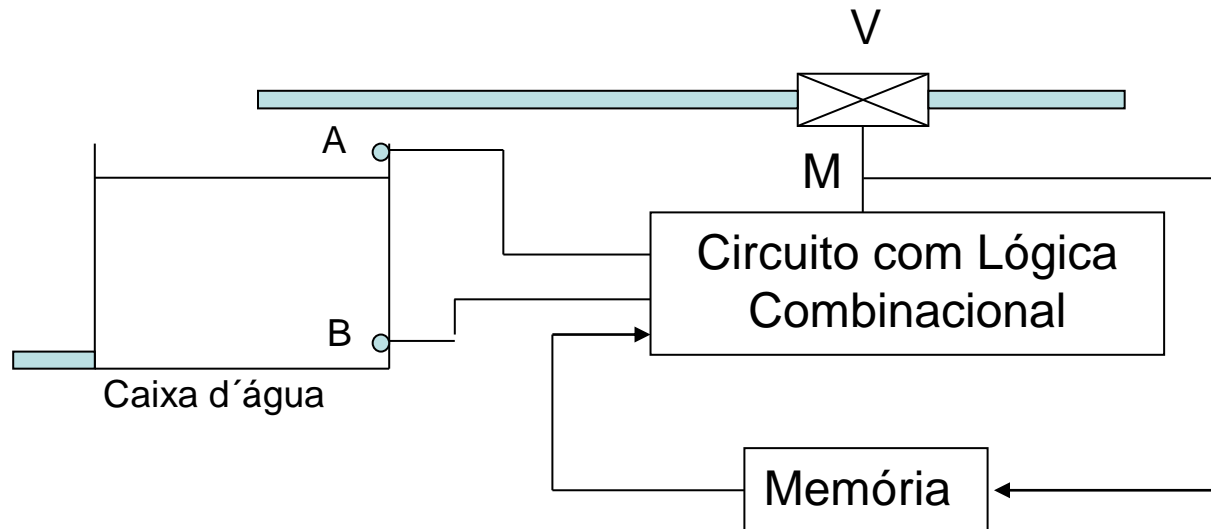
- Na verdade precisamos da seguinte tabela:

A B	V
0 0	1
0 1	? (0, quando a água está descendo e 1 quando estiver subindo)
1 0	X (impossível)
1 1	0

Portanto, alguma variável precisa informar à lógica se no momento a caixa está em processo de enchimento ou esvaziamento, “lembrando” qual foi o último estado alcançado, isto é, cheio ($V=0$) ou vazio ($V=1$).

O novo diagrama é mostrado a seguir.

Sequencial x Combinacional



Sequencial x Combinacional

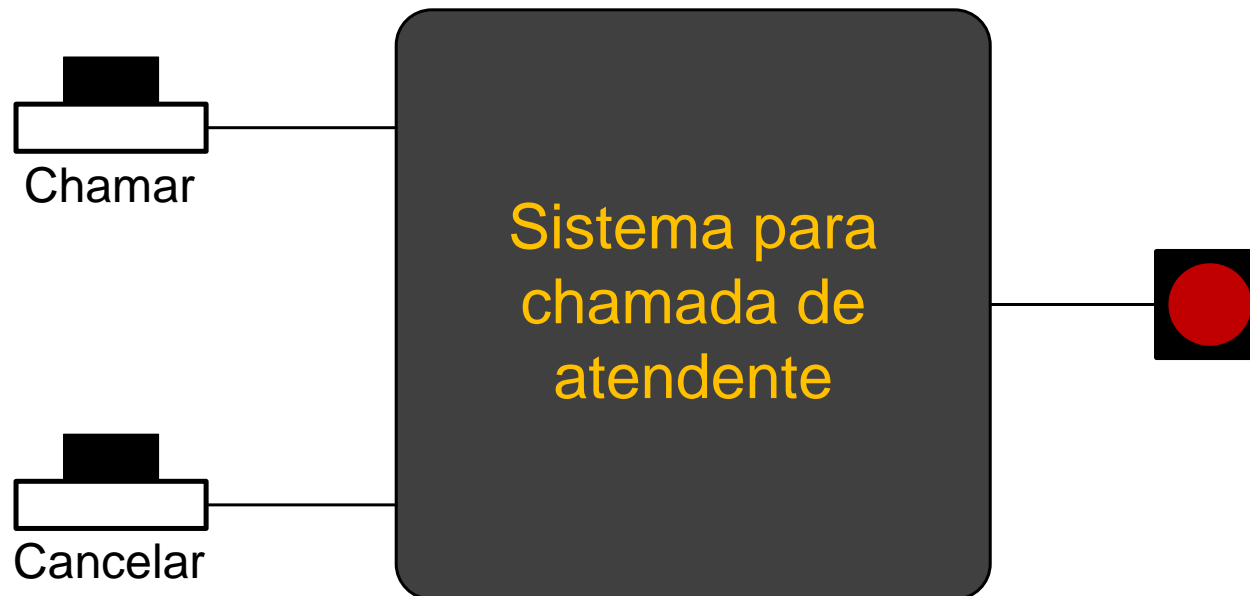
- Uma nova tabela da verdade, incorporando a variável de memorização é mostrada abaixo.

M A B	V
0 0 0	1 (liga válvula, caixa acabou de esvaziar)
0 0 1	0 (caixa esvaziando)
0 1 0	X (impossível)
0 1 1	0 (válvula foi recentemente desligada)
1 0 0	1 (válvula foi recentemente ligada)
1 0 1	1 (caixa enchendo)
1 1 0	X (impossível)
1 1 1	0 (desliga válvula, caixa acabou de encher)

- Ao conjunto do bloco combinacional mais o dispositivo de memória chamamos **lógica sequencial**. Nesta lógica, os estados presentes das saídas não dependem apenas dos estados das entradas, mas também dos estados anteriores do próprio sistema.
- Para a solução do problema da caixa d'água usamos um dispositivo de memória capaz de armazenar um bit, que é a definição funcional de **flip-flop**.

Sequencial x Combinacional

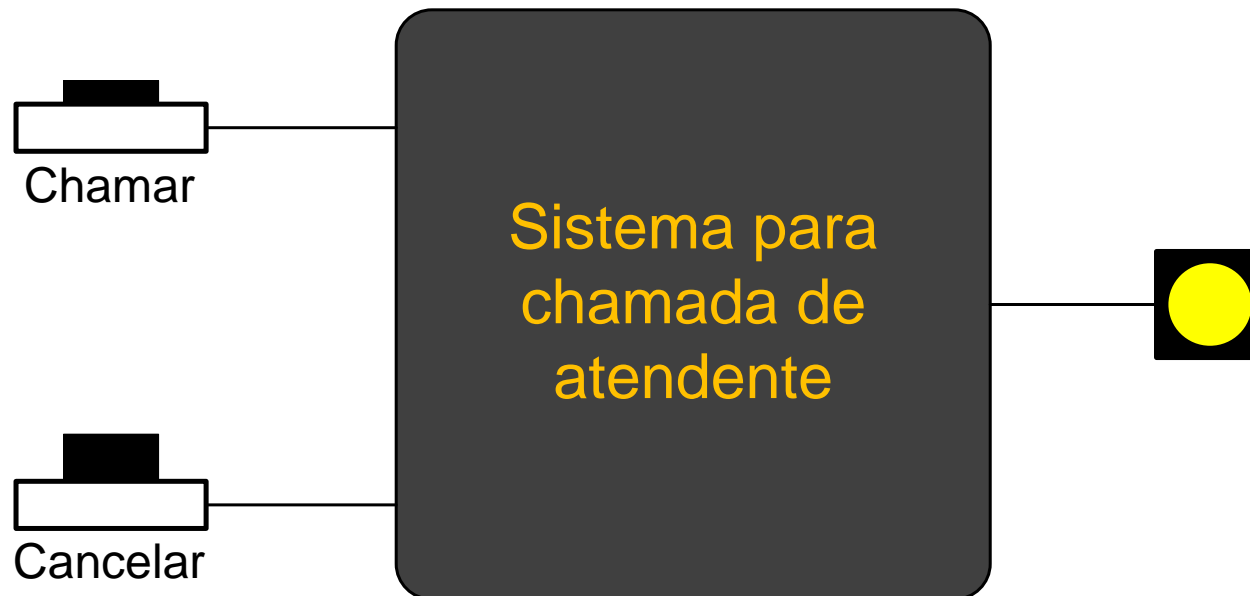
- Considere que deseja-se projetar o seguinte sistema:



- Um toque no botão “chamar” **acende** a lâmpada e um toque no botão “cancelar” **apaga** a lâmpada.

Sequencial x Combinacional

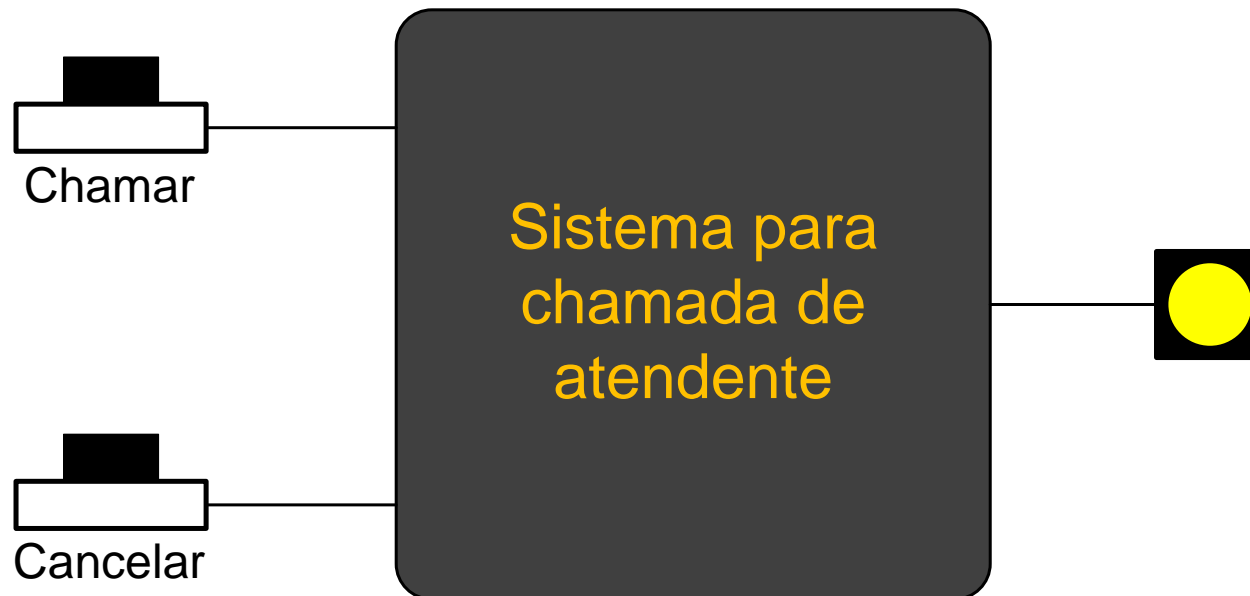
- Considere que deseja-se projetar o seguinte sistema:



- Um toque no botão “chamar” **acende** a lâmpada e um toque no botão “cancelar” **apaga** a lâmpada.

Sequencial x Combinacional

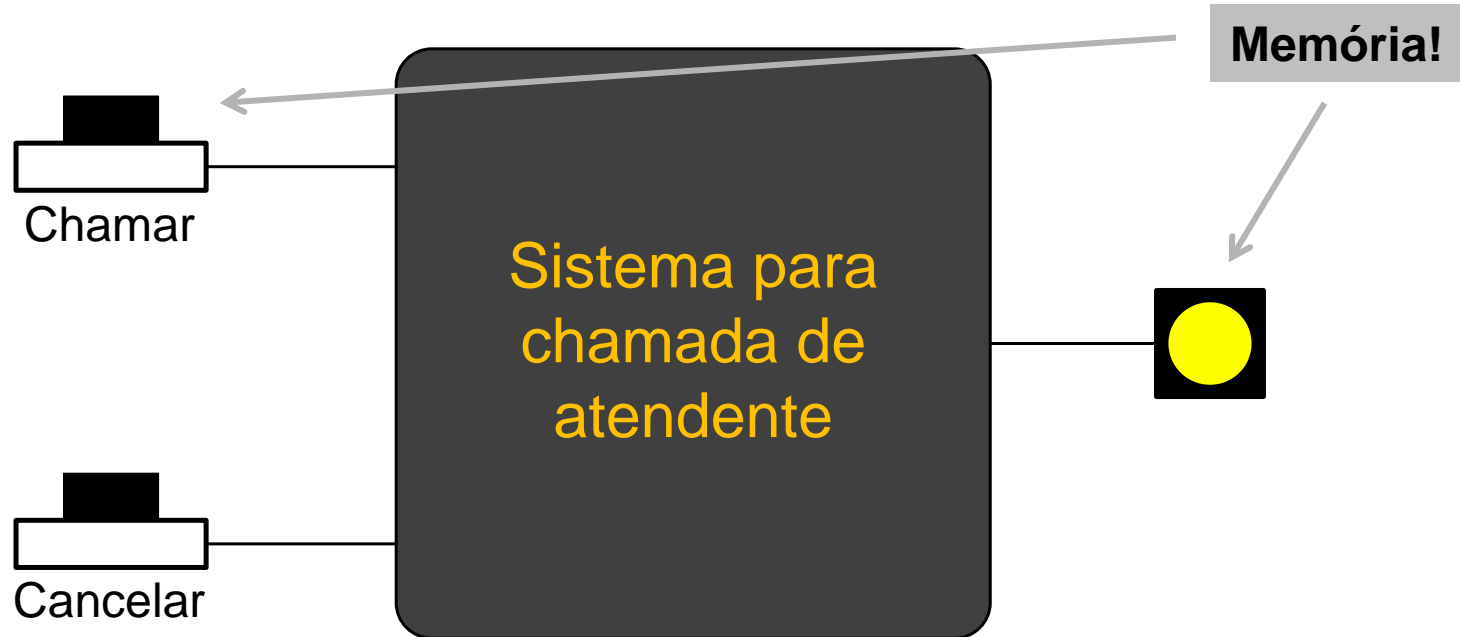
- Considere que deseja-se projetar o seguinte sistema:



- Um toque no botão “chamar” **acende** a lâmpada e um toque no botão “cancelar” **apaga** a lâmpada.

Sequencial x Combinacional

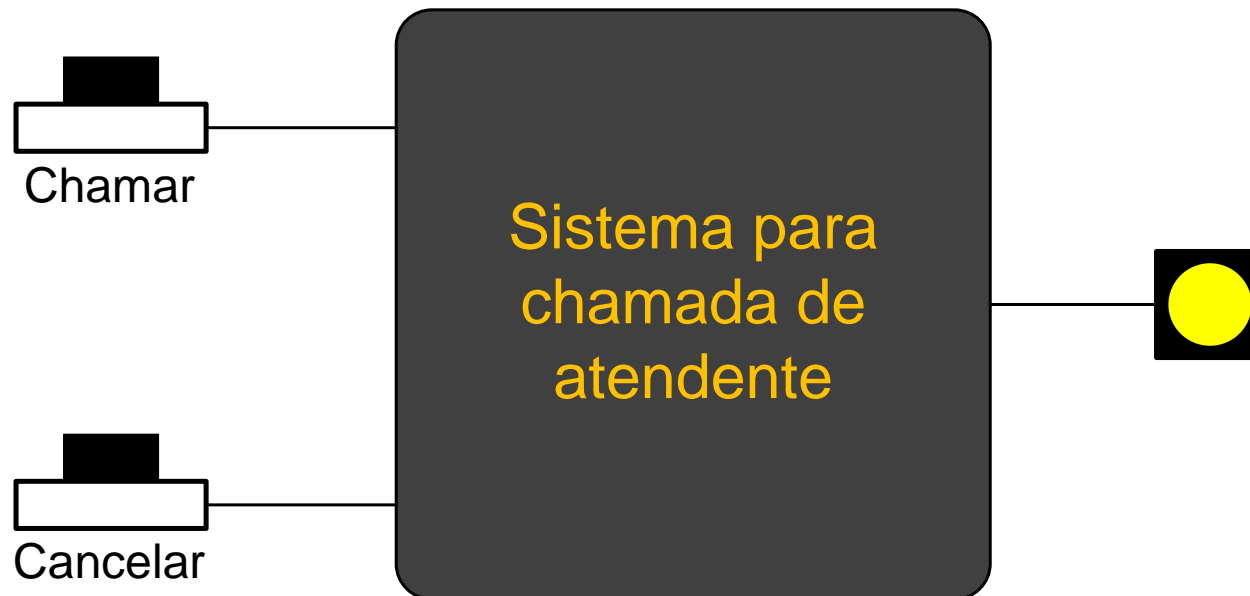
- Considere que deseja-se projetar o seguinte sistema:



- Um toque no botão “chamar” **acende** a lâmpada e um toque no botão “cancelar” **apaga** a lâmpada.

Sequencial x Combinacional

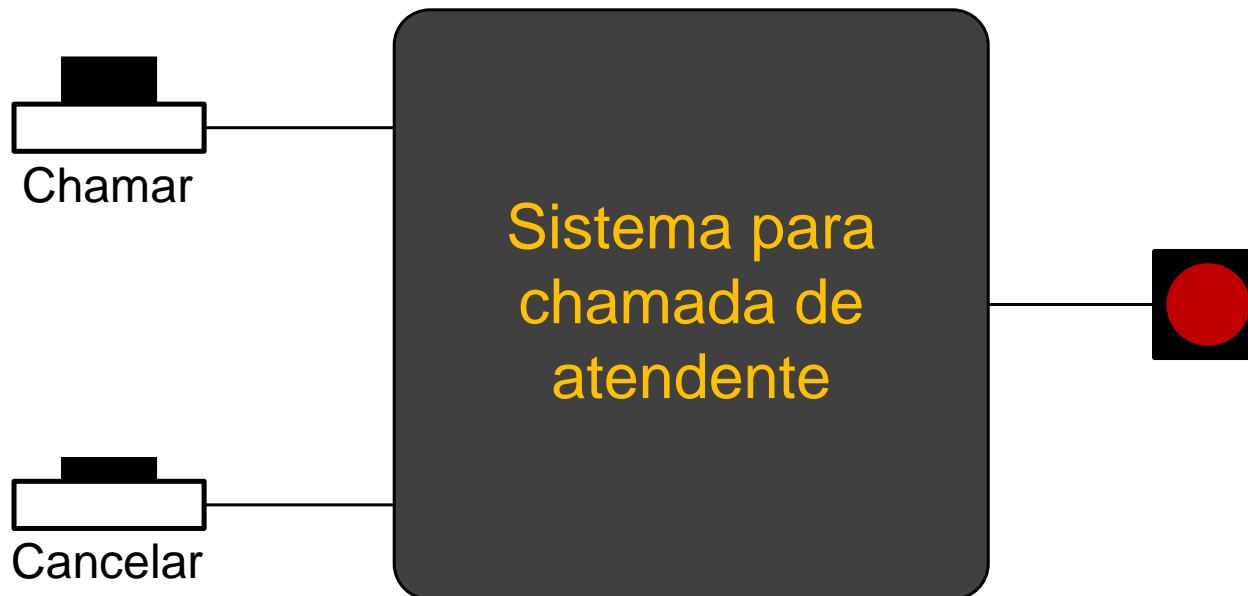
- Considere que deseja-se projetar o seguinte sistema:



- Um toque no botão “chamar” **acende** a lâmpada e um toque no botão “cancelar” **apaga** a lâmpada.

Sequencial x Combinacional

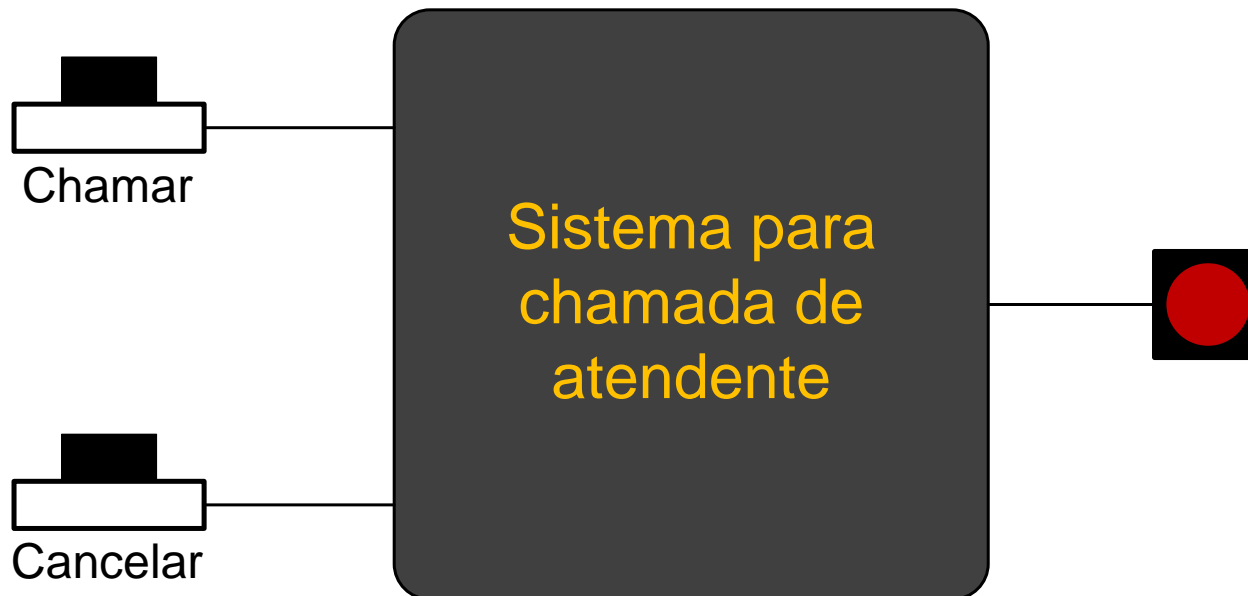
- Considere que deseja-se projetar o seguinte sistema:



- Um toque no botão “chamar” **acende** a lâmpada e um toque no botão “cancelar” **apaga** a lâmpada.

Sequencial x Combinacional

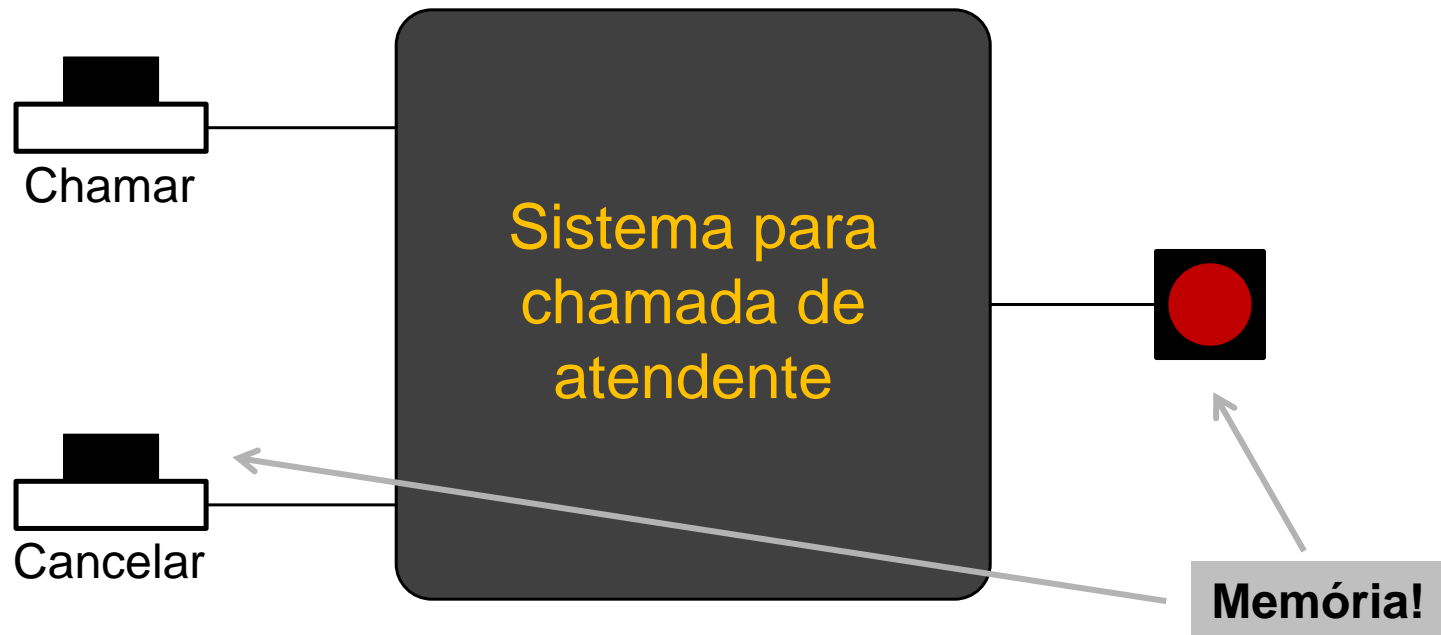
- Considere que deseja-se projetar o seguinte sistema:



- Um toque no botão “chamar” **acende** a lâmpada e um toque no botão “cancelar” **apaga** a lâmpada.

Sequencial x Combinacional

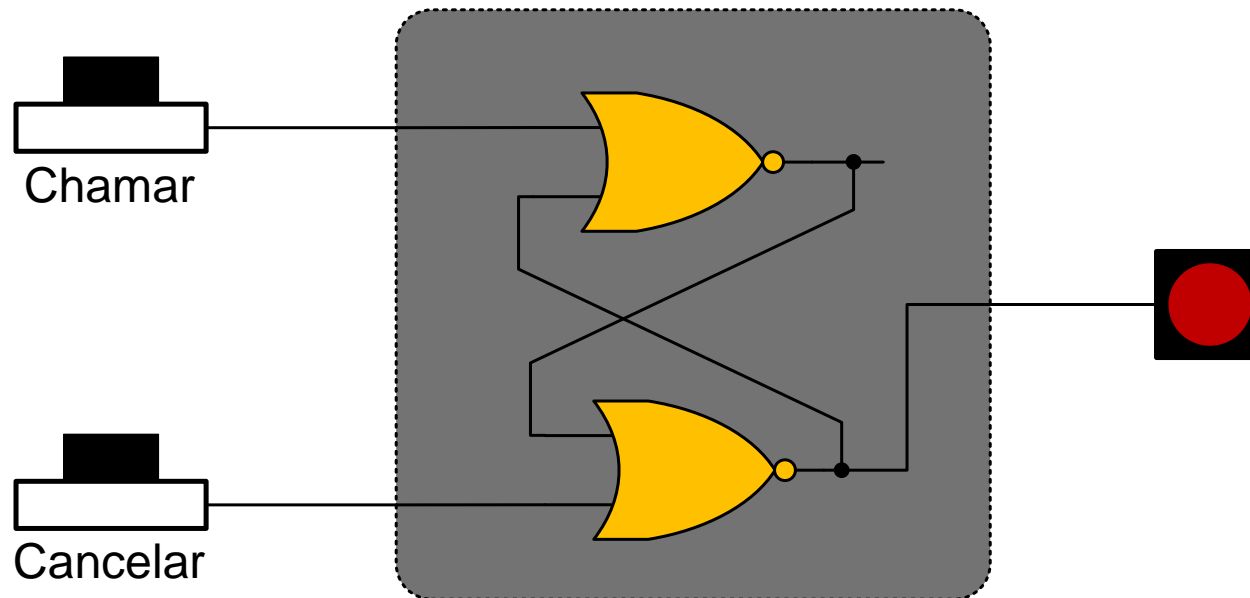
- Considere que deseja-se projetar o seguinte sistema:



- Um toque no botão “chamar” **acende** a lâmpada e um toque no botão “cancelar” **apaga** a lâmpada.

Sequencial x Combinacional

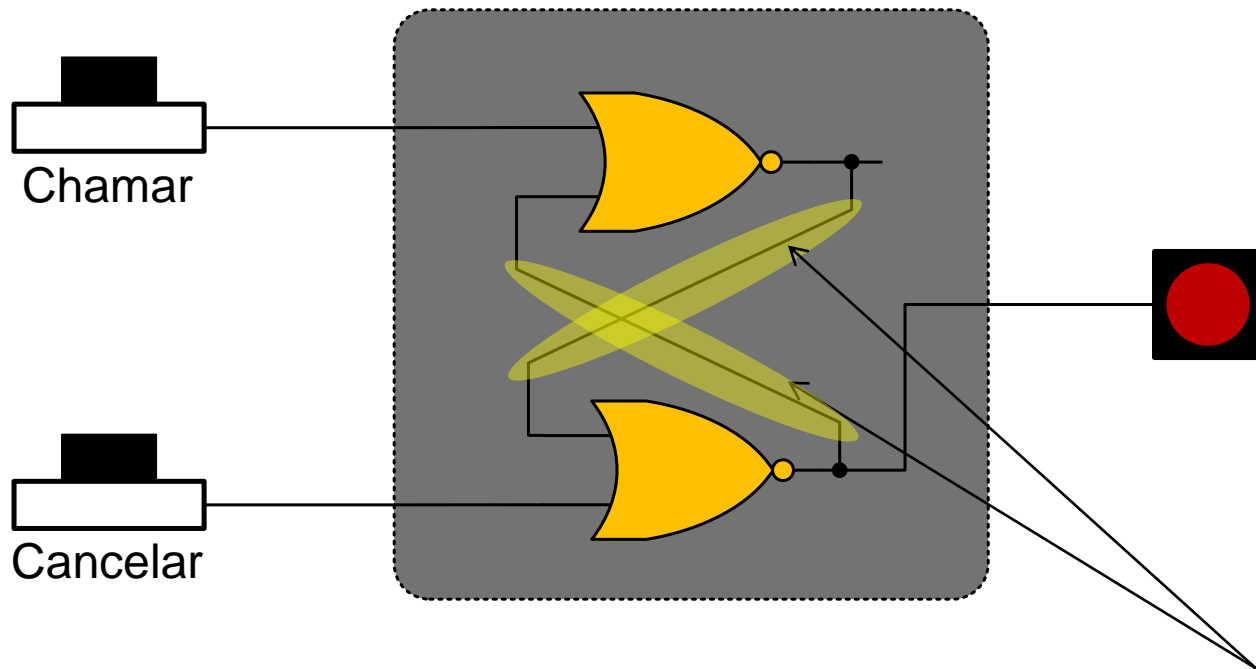
- Uma possível solução:



- Esse é um circuito que tem **memória**!
 - A saída **não depende somente das entradas atuais**, mas também do que aconteceu no **passado**.

Sequencial x Combinacional

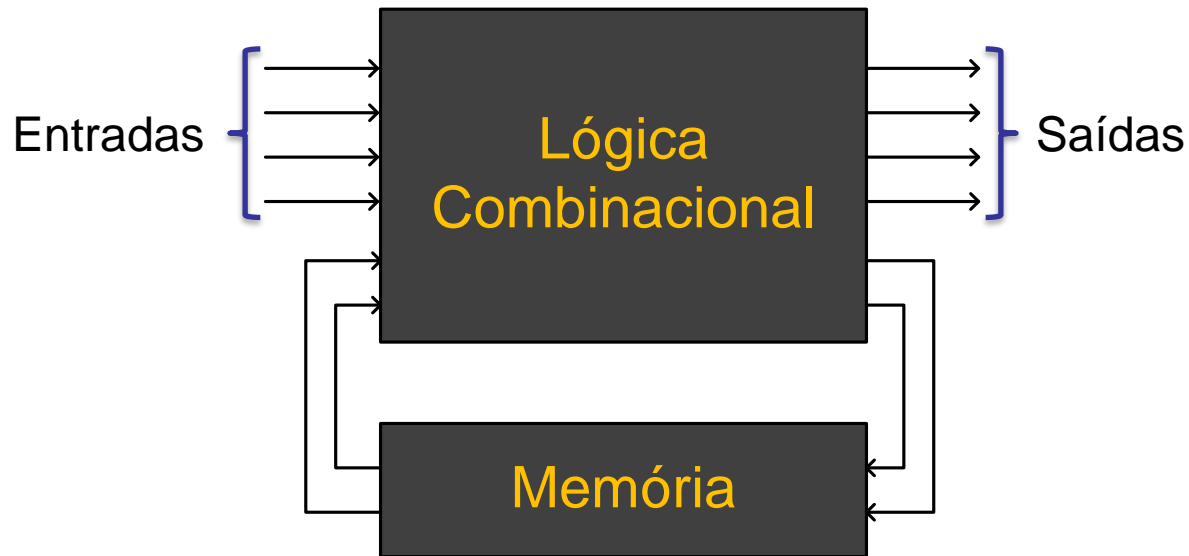
- Uma possível solução:



- A chave para a **memória** em circuitos lógicos é a **realimentação**.

Sequencial x Combinacional

Circuito Sequencial



- Sequencial x combinacional
- Circuitos sequenciais assíncronos e síncronos
- Latches e flip-flops
- Latch SR
- Flip-flop SR
- Flip-flop JK
- Flip-flop D
- Flip-flop T
- Entradas assíncronas

Sistemas assíncronos x síncronos

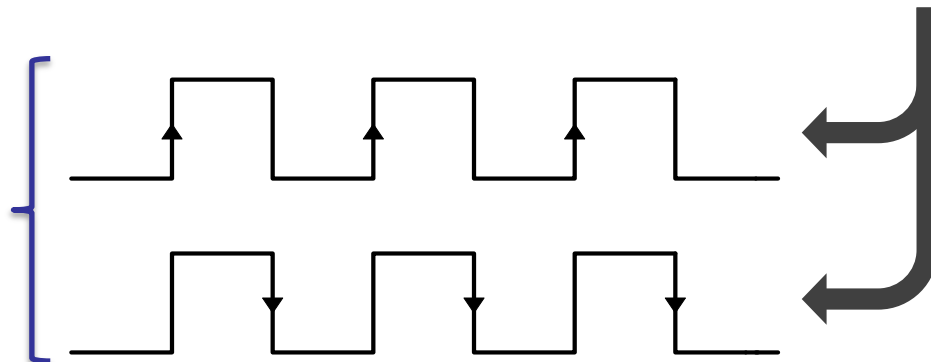
- Sistema assíncrono

- As saídas dos circuitos lógicos podem mudar de **estado** (nível lógico) assim que as entradas tiverem seus níveis lógicos alterados.

- Sistema síncrono

- Os instantes nos quais as saídas dos circuitos lógicos podem mudar de estado são determinadas por um sinal de **clock**.

Usualmente, sistemas síncronos mudam de **estado** na borda de **subida** ou **descida** do sinal do **clock**

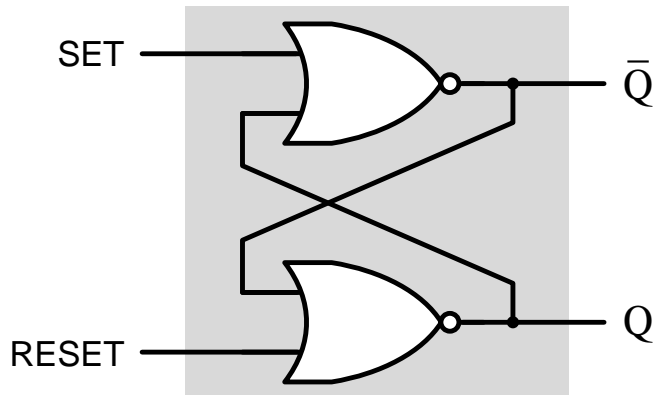


- Sequencial x combinacional
- Circuitos sequenciais assíncronos e síncronos
- **Latches e flip-flops**
- Latch SR
- Flip-flop SR
- Flip-flop JK
- Flip-flop D
- Flip-flop T
- Entradas assíncronas

Latches e Flip-flops

- Circuitos sequenciais elementares
- Têm capacidade de armazenar informação
- Unidade elementar de memória => **1 bit**
- **Latch** – assíncrono
- **Flip-flop** – versão síncrona do **latch**

Latch SR (com portas NOR)



- Circuito básico: **Latch SR** (com portas NOR)
- A saída de cada porta NOR é conectada na entrada da outra porta (**realimentação**).
- As saídas das portas NOR (denominadas Q e \bar{Q}) são as **saídas** do Latch.

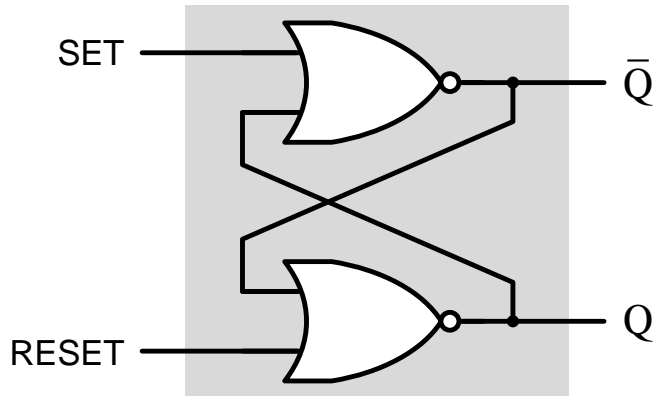
- Sob condições normais de operação as saídas Q e \bar{Q} são inversas uma da outra.
- Sempre que nos referirmos ao **estado** do Latch, estaremos nos referindo ao estado da **saída Q**.

Estados do Latch SR

estado **alto**
 $Q = 1$ e $\bar{Q} = 0$

estado **baixo**
 $Q = 0$ e $\bar{Q} = 1$

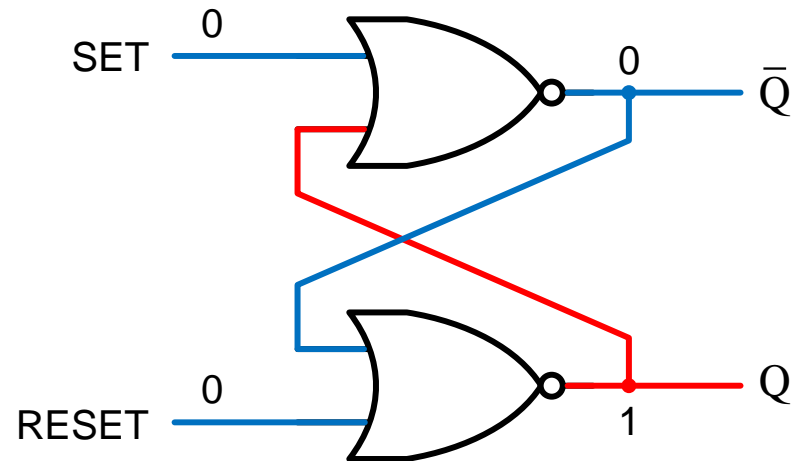
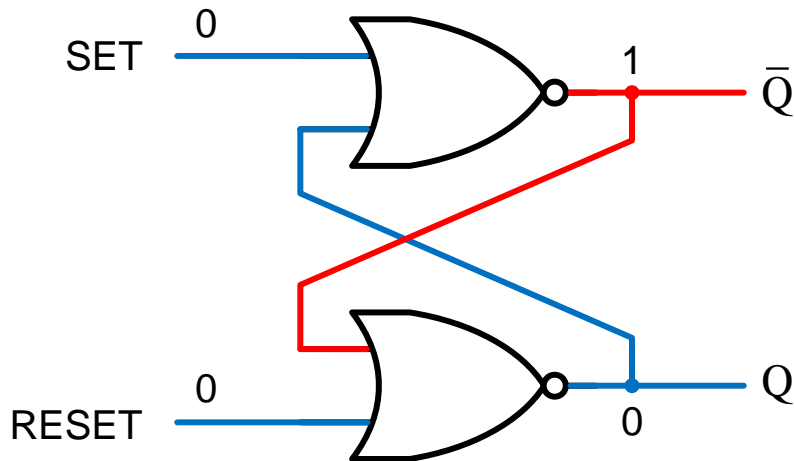
Latch SR (com portas NOR)

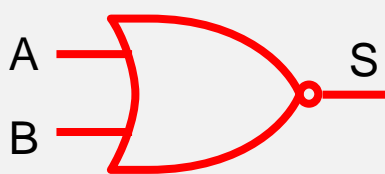


- Duas entradas, denominadas **SET** e **RESET**.
- As entradas SET e RESET ficam, usualmente, em nível lógico **baixo**.
- Para alterar o **estado** do Latch (ou seja, sua saída Q), uma das entradas (SET ou RESET) deve ser levada **momentaneamente** para nível lógico **alto**.
- Os dois **estados** são possíveis quando ambas as entradas **SET** e **RESET** estão em nível lógico **baixo**.

Latch SR (com portas NOR)

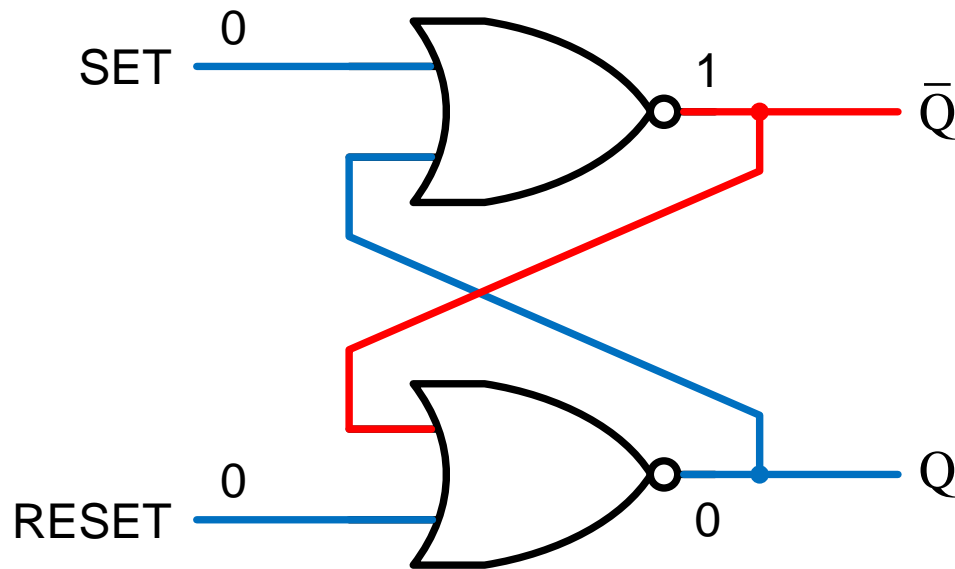
- Os dois **estados** são possíveis quando ambas as entradas **SET** e **RESET** estão em nível lógico **baixo**.

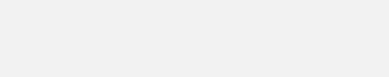


		
A	B	S
0	0	1
0	1	0
1	0	0
1	1	0

Latch SR (com portas NOR)

Análise do cenário 1

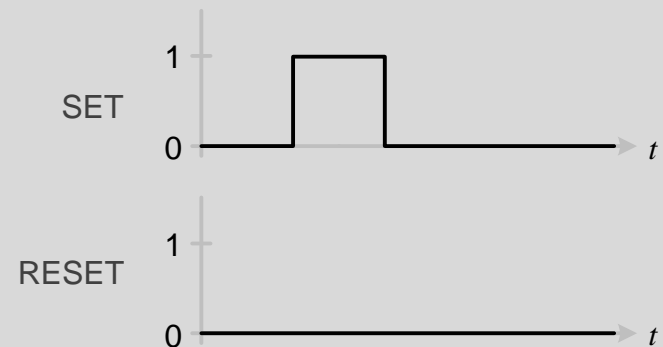




A	B	S
0	0	1
0	1	0
1	0	0
1	1	0

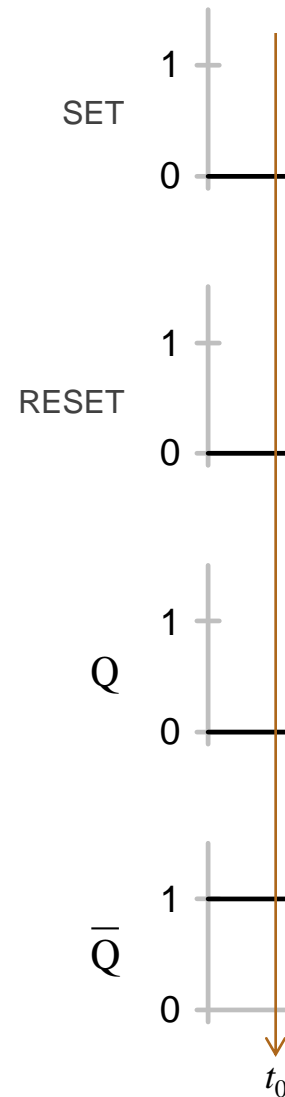
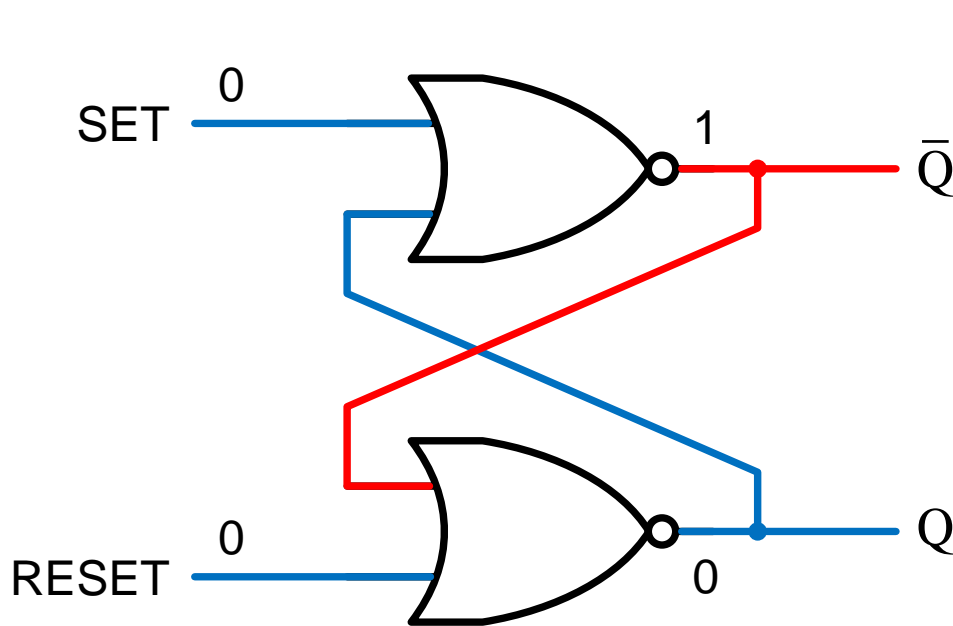
Cenário 1:

- Inicialmente, as entradas **SET** e **RESET** estão em nível lógico **baixo** e o **estado** do Latch (isto é, a saída Q), em nível lógico **baixo**.
- Na entrada **SET** é aplicado um pulso de nível lógico **alto**, mantendo a entrada **RESET** no nível lógico **baixo**.



Latch SR (com portas NOR)

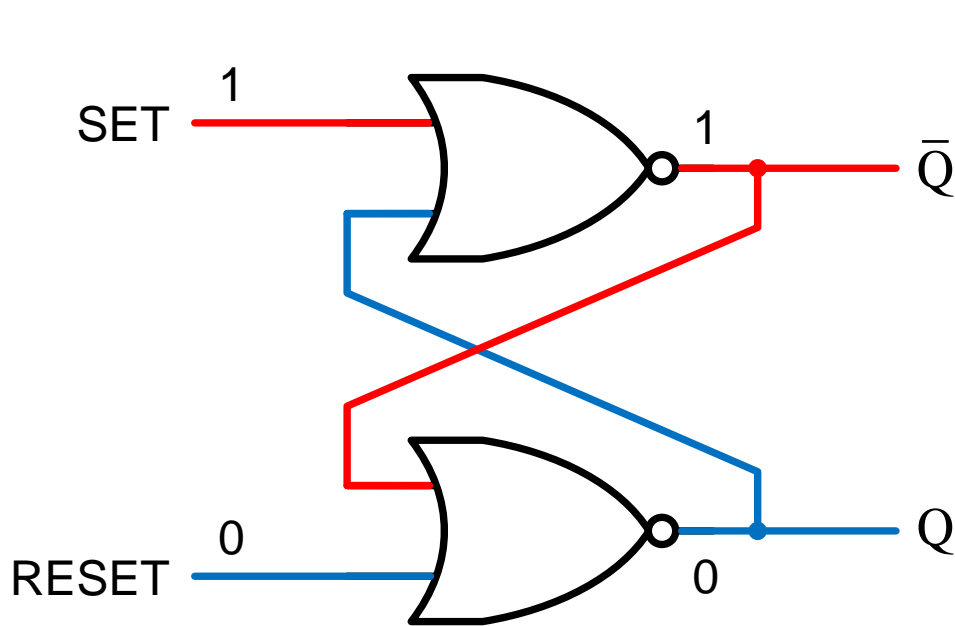
Análise do cenário 1



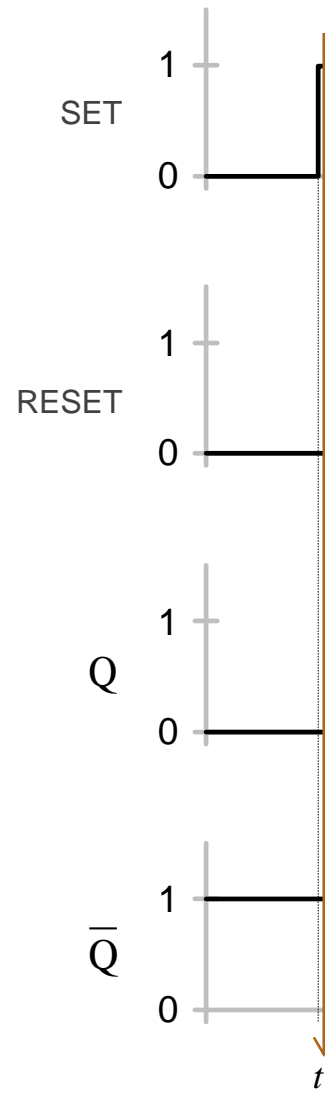
A	B	S
0	0	1
0	1	0
1	0	0
1	1	0

Latch SR (com portas NOR)

Análise do cenário 1

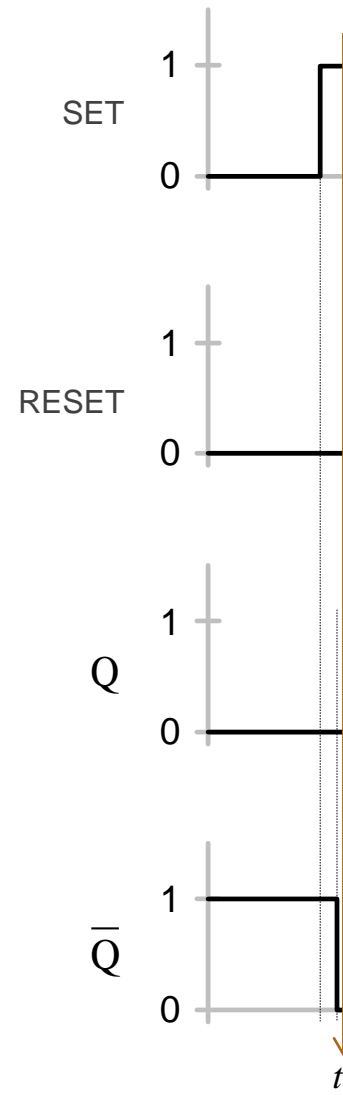
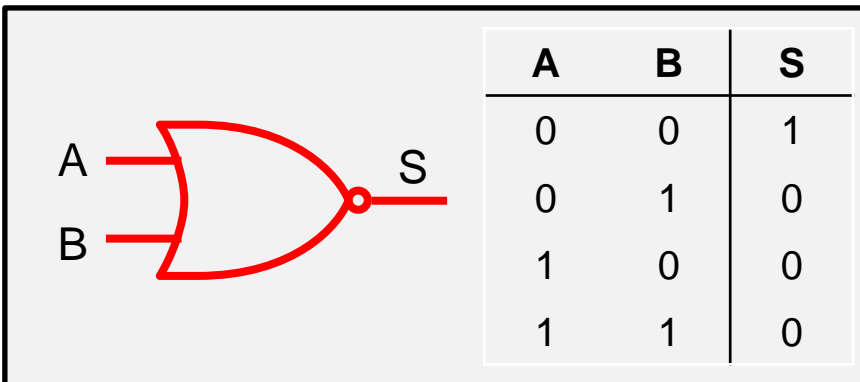
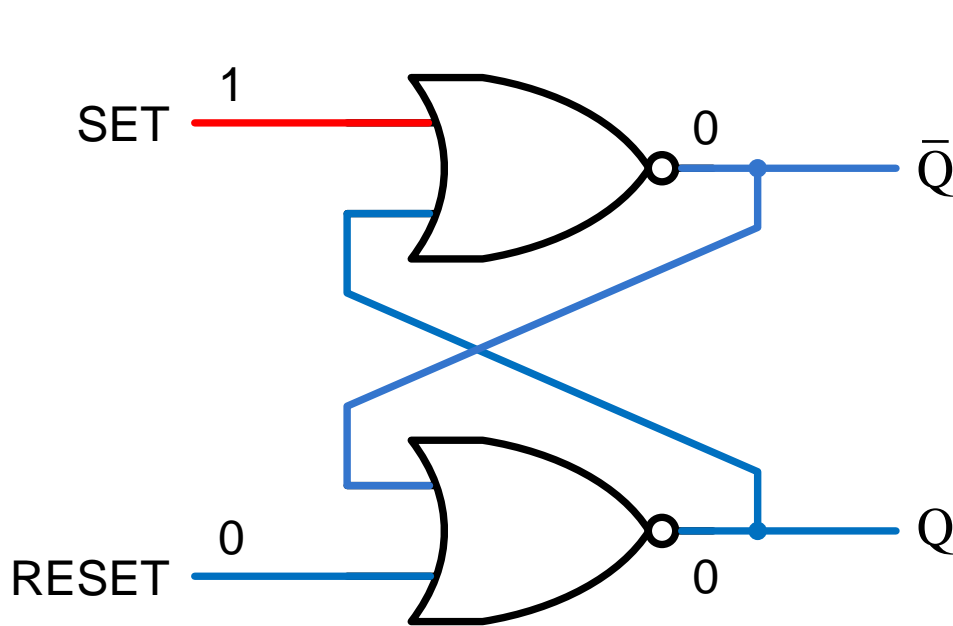


A	B	S
0	0	1
0	1	0
1	0	0
1	1	0



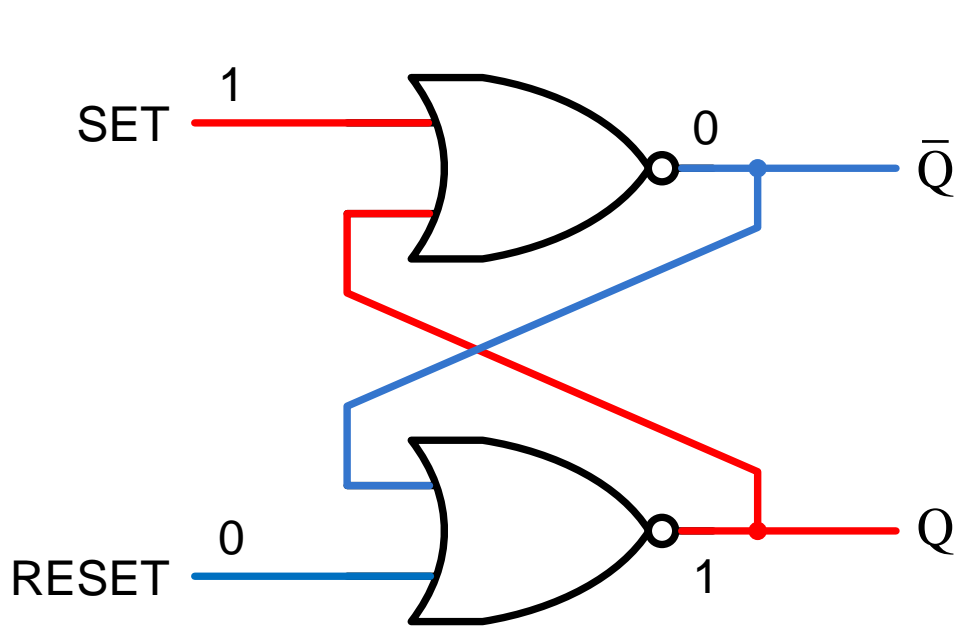
Latch SR (com portas NOR)

Análise do cenário 1

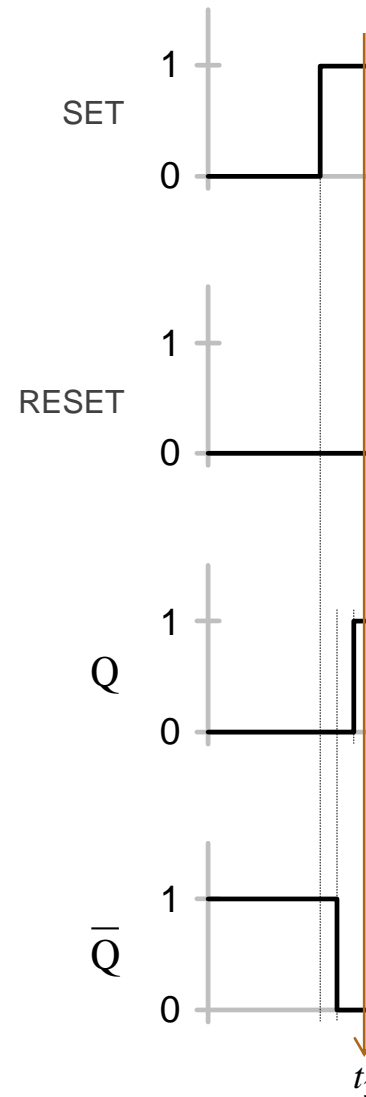


Latch SR (com portas NOR)

Análise do cenário 1

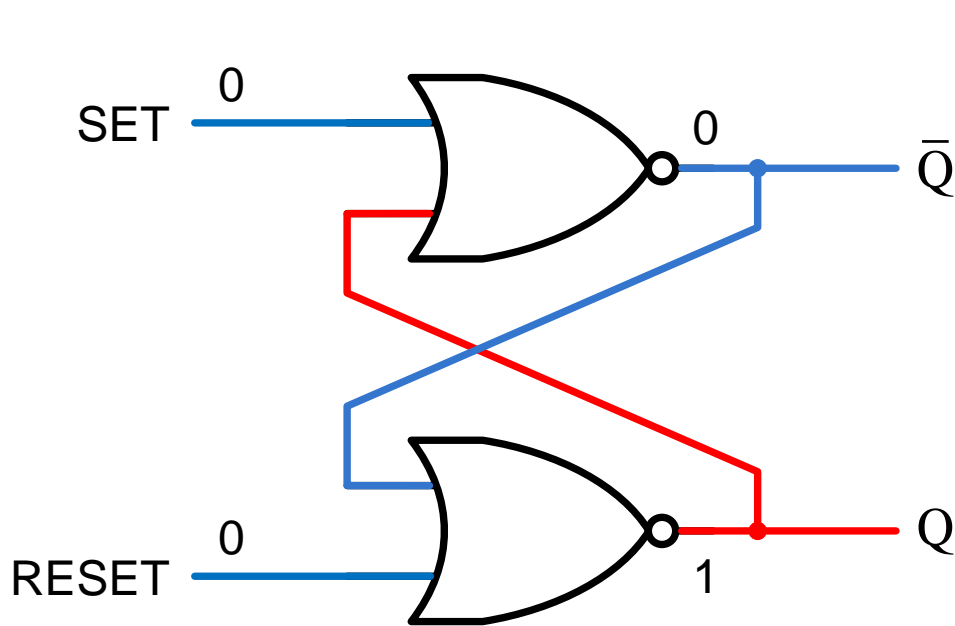


A	B	S
0	0	1
0	1	0
1	0	0
1	1	0

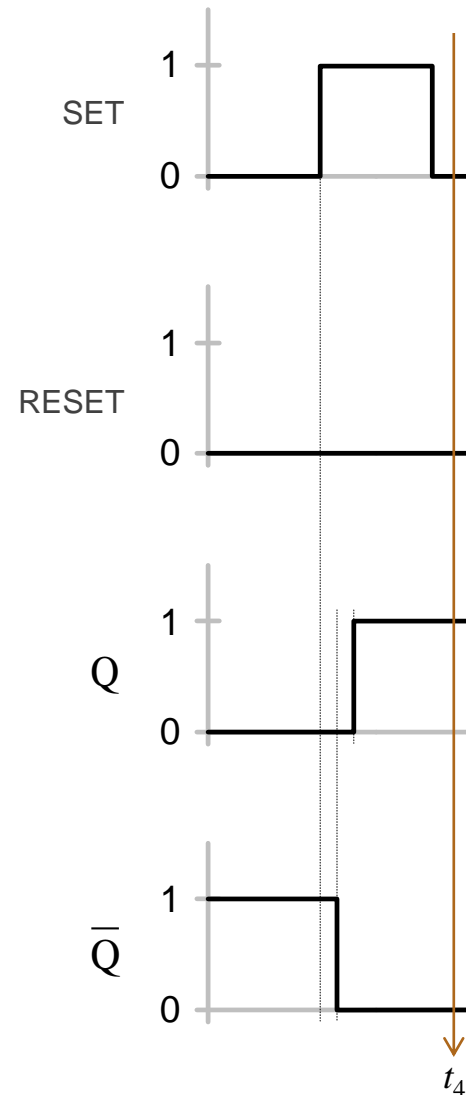


Latch SR (com portas NOR)

Análise do cenário 1

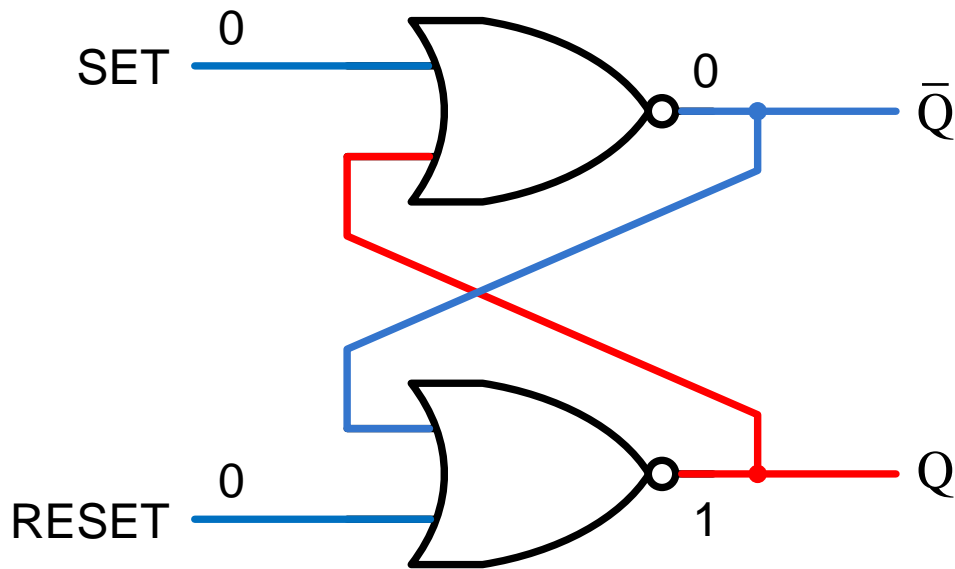


A	B	S
0	0	1
0	1	0
1	0	0
1	1	0

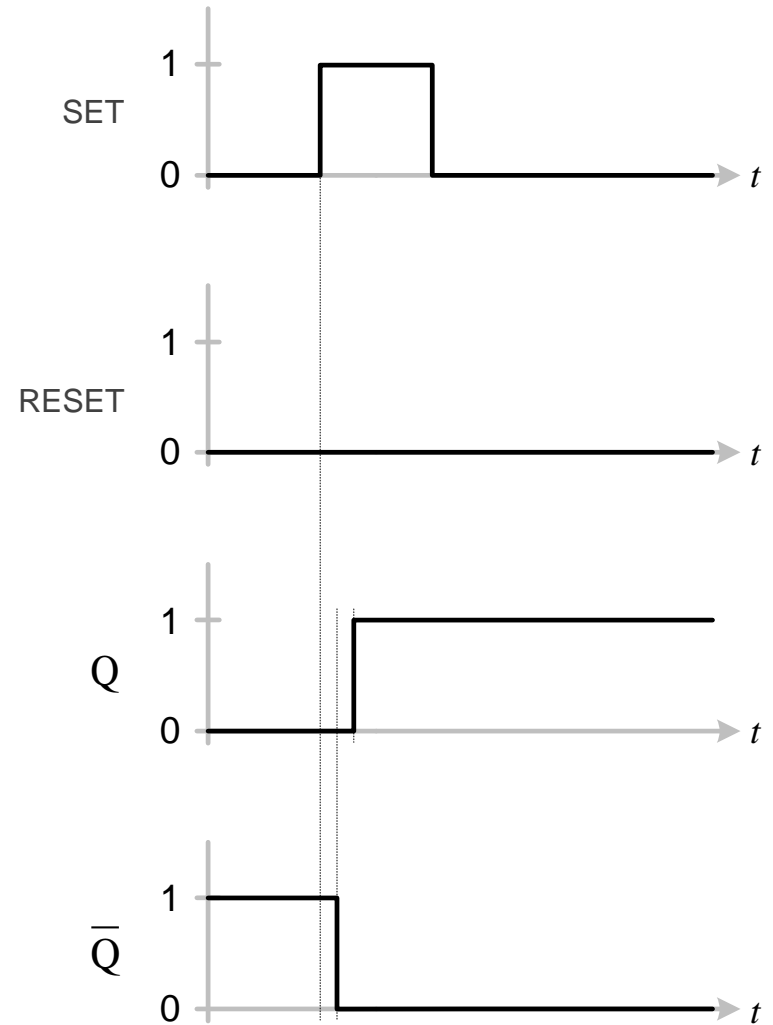


Latch SR (com portas NOR)

Análise do cenário 1

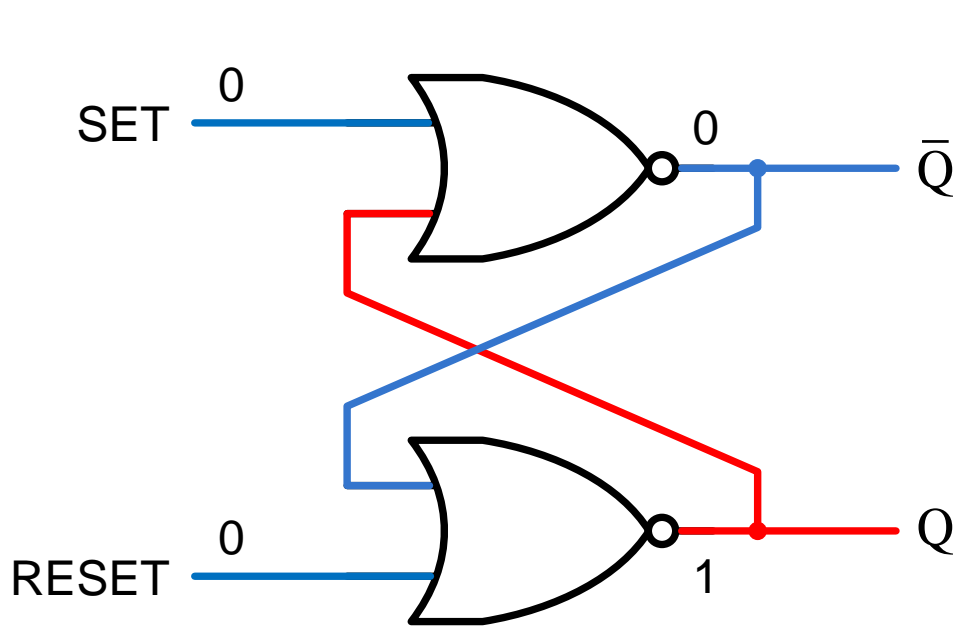


A	B	S
0	0	1
0	1	0
1	0	0
1	1	0

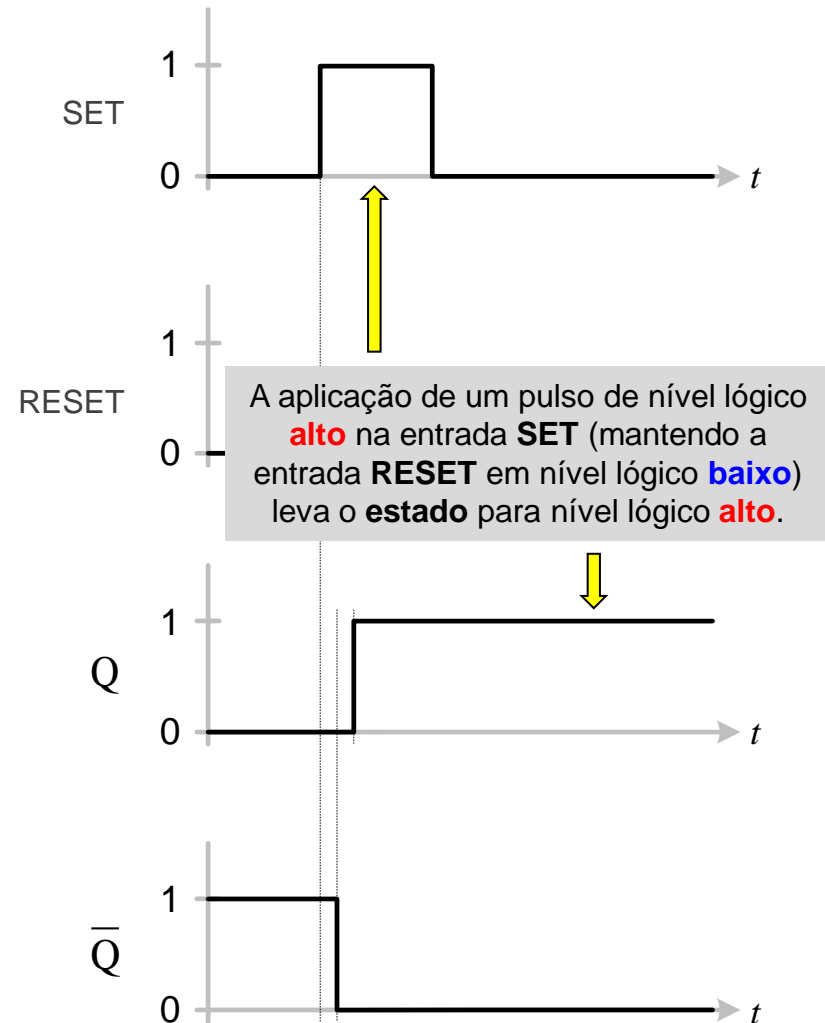


Latch SR (com portas NOR)

Análise do cenário 1

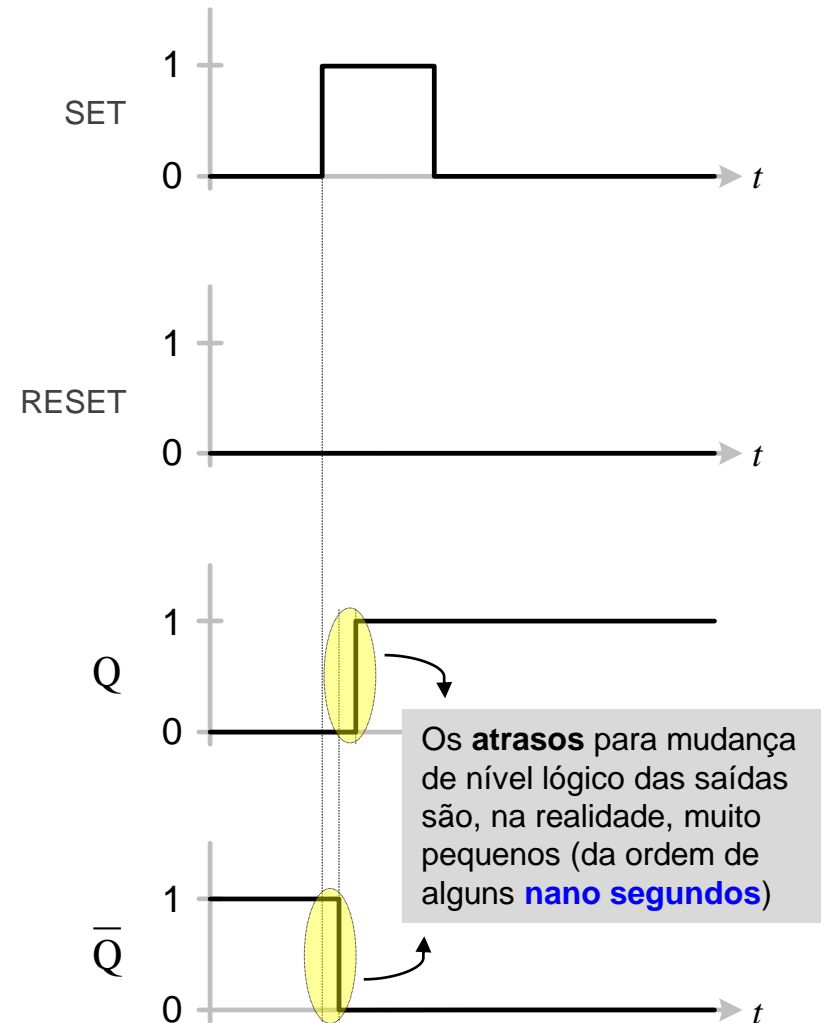
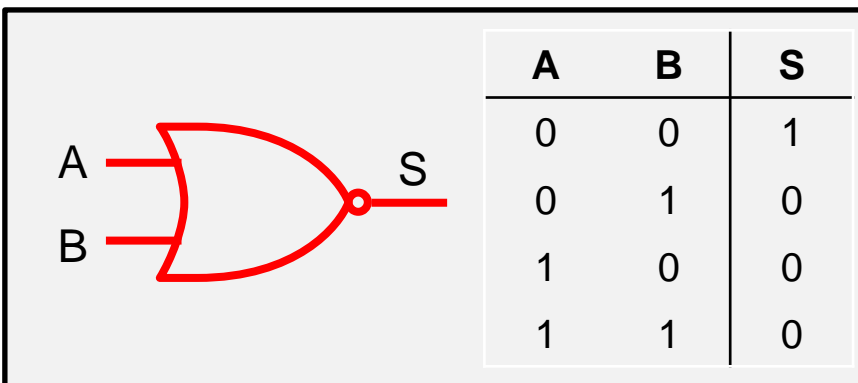
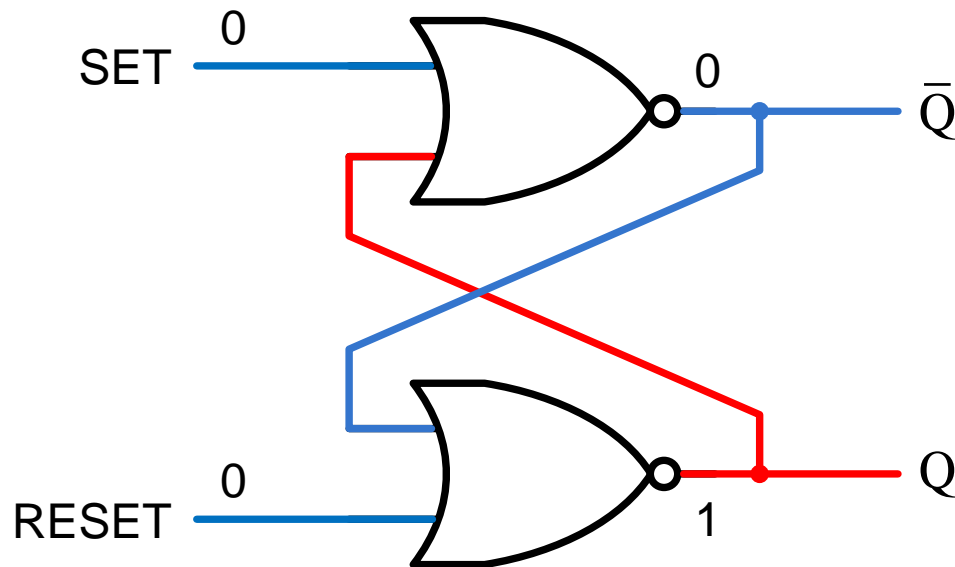


A	B	S
0	0	1
0	1	0
1	0	0
1	1	0



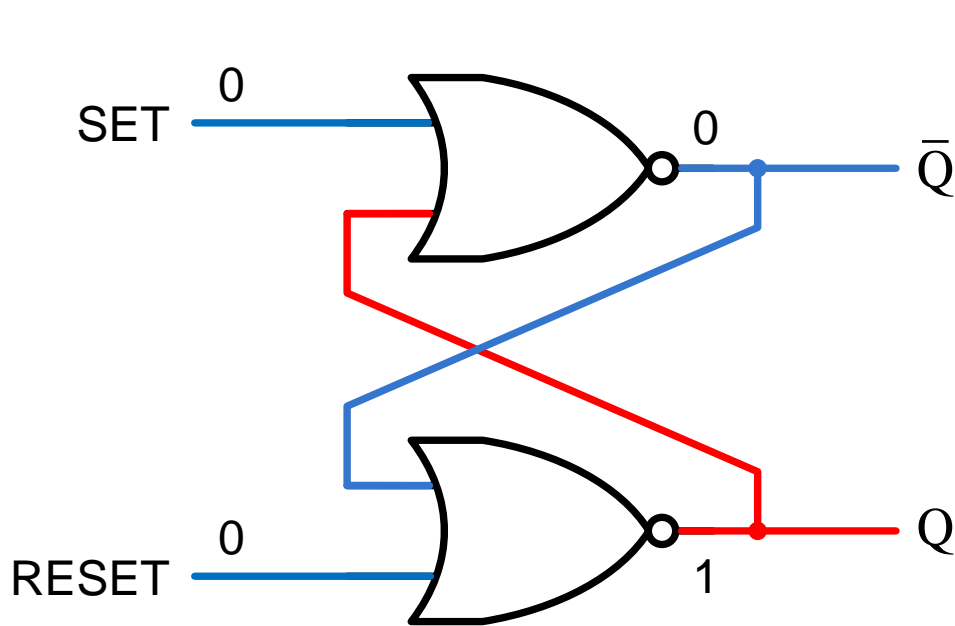
Latch SR (com portas NOR)

Análise do cenário 1

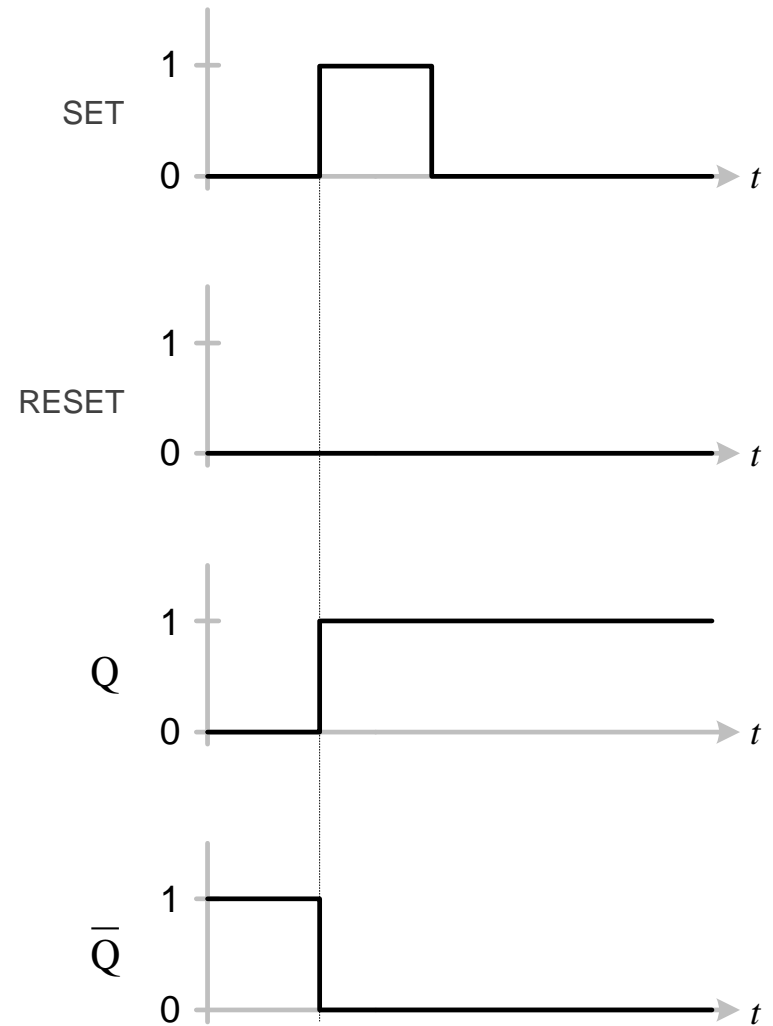


Latch SR (com portas NOR)

Análise do cenário 1

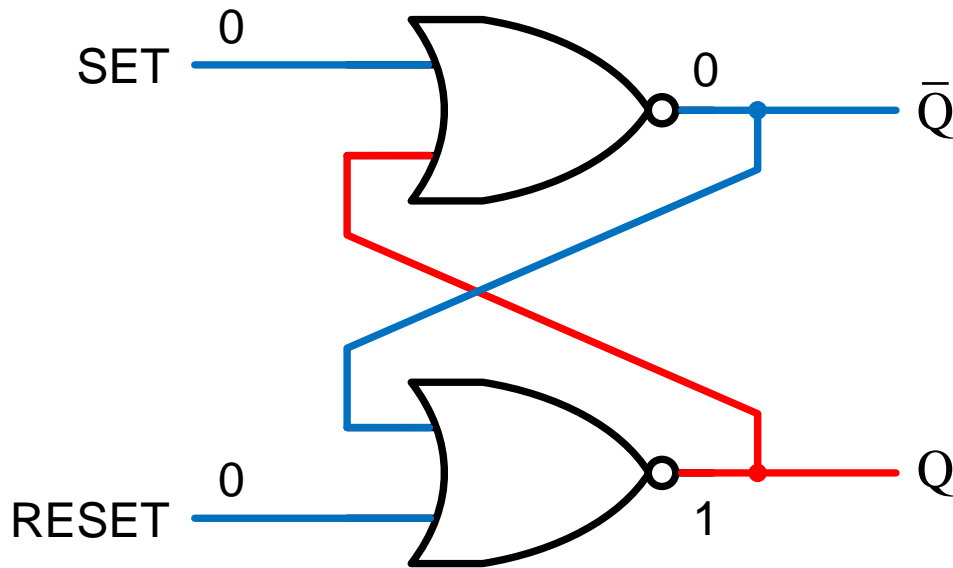


A	B	S
0	0	1
0	1	0
1	0	0
1	1	0



Latch SR (com portas NOR)

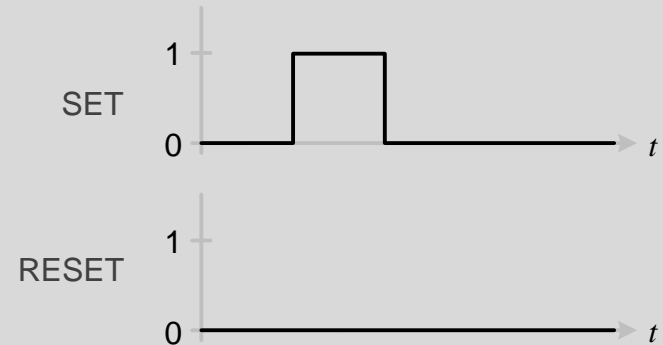
Análise do cenário 2



A	B	S
0	0	1
0	1	0
1	0	0
1	1	0

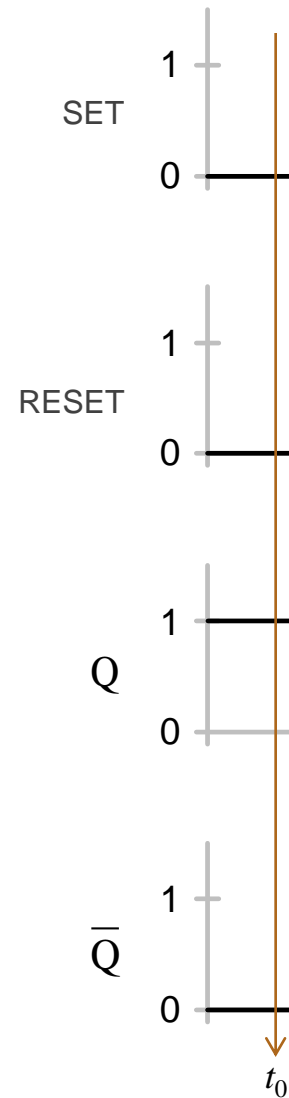
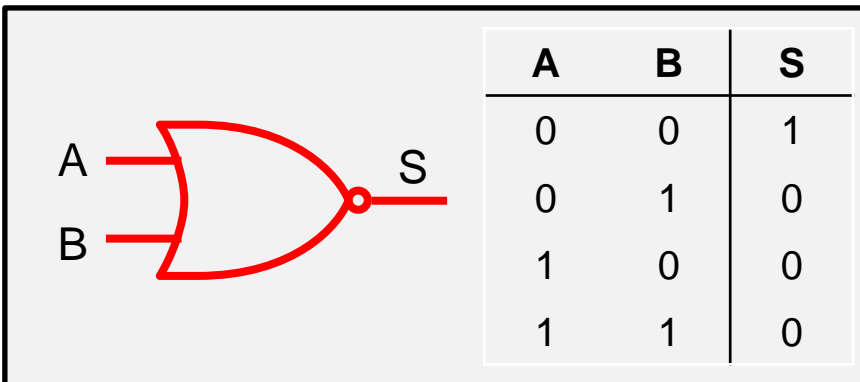
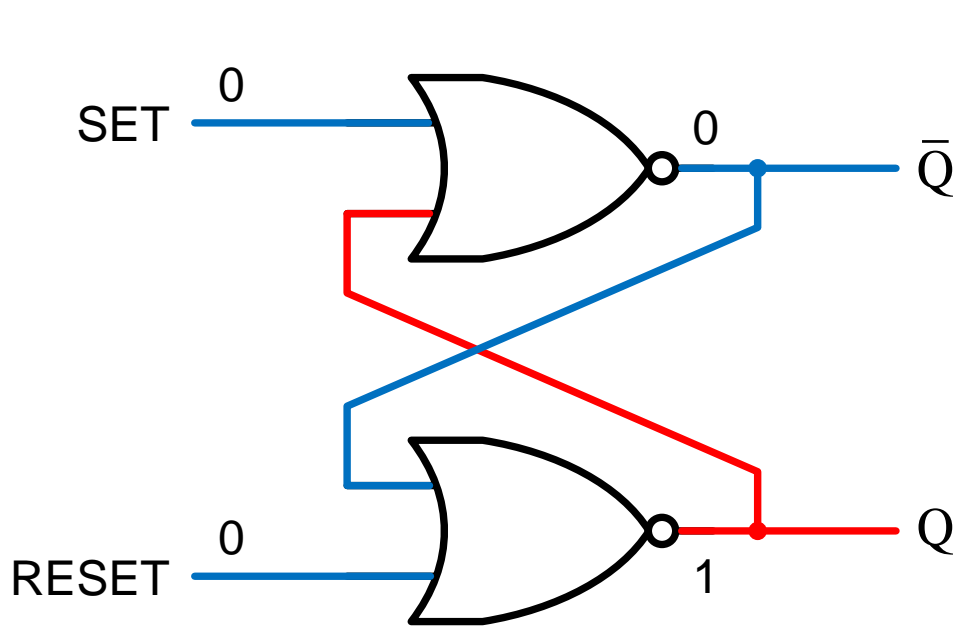
Cenário 2:

- Inicialmente, as entradas **SET** e **RESET** estão em nível lógico **baixo** e o **estado** do Latch (isto é, a saída Q), em nível lógico **alto**.
- Na entrada **SET** é aplicado um pulso de nível lógico **alto**, mantendo a entrada **RESET** no nível lógico **baixo**.



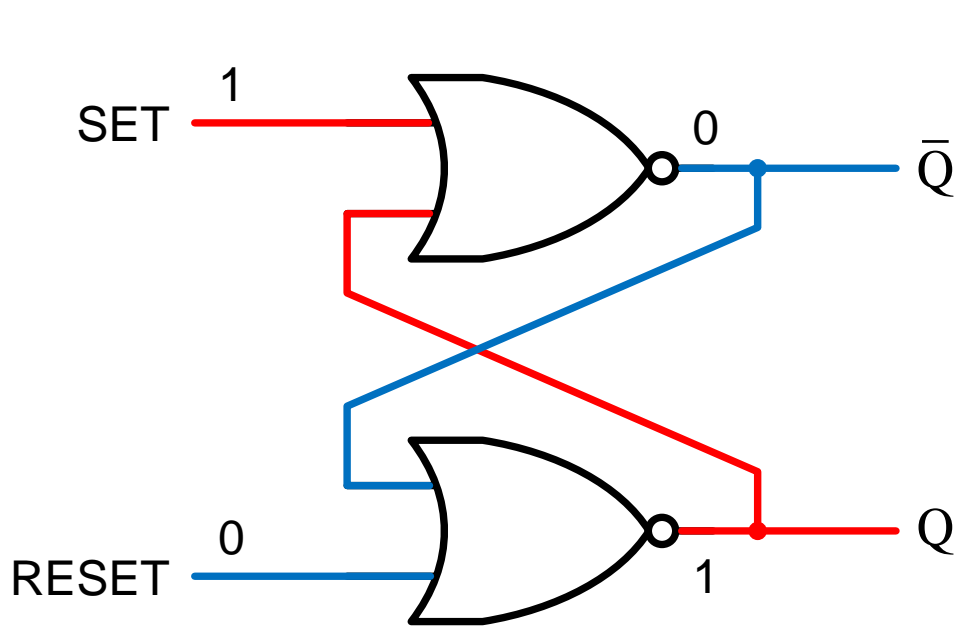
Latch SR (com portas NOR)

Análise do cenário 2

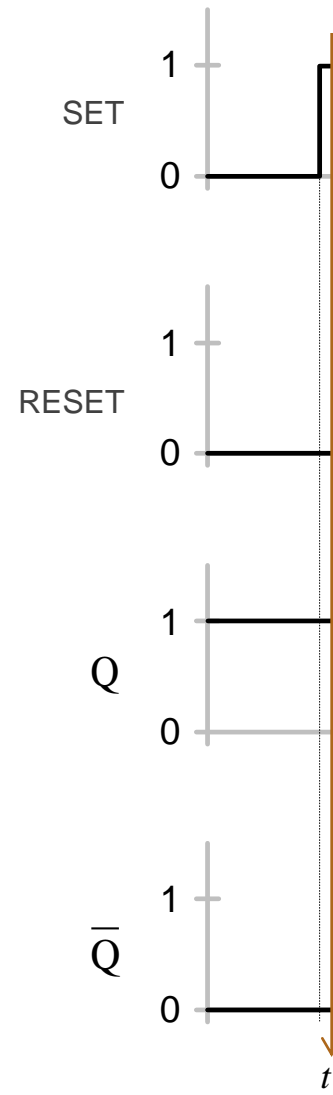


Latch SR (com portas NOR)

Análise do cenário 2

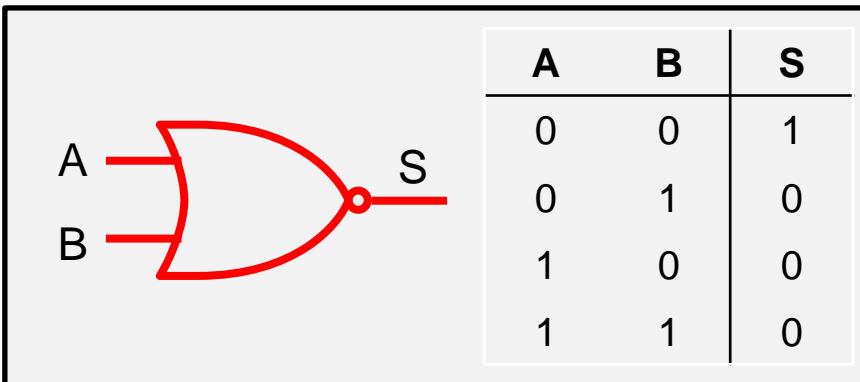
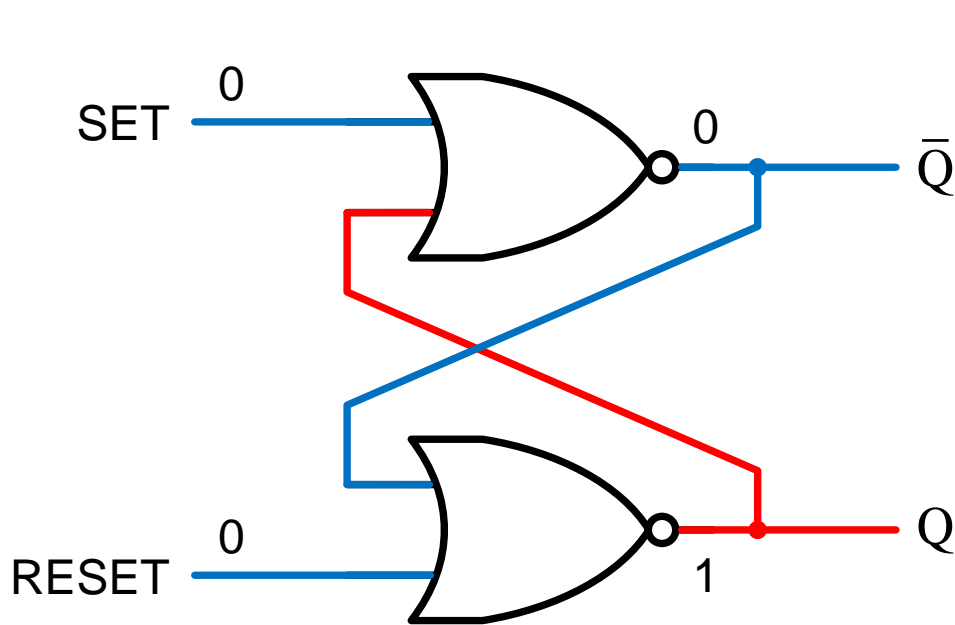


A	B	S
0	0	1
0	1	0
1	0	0
1	1	0



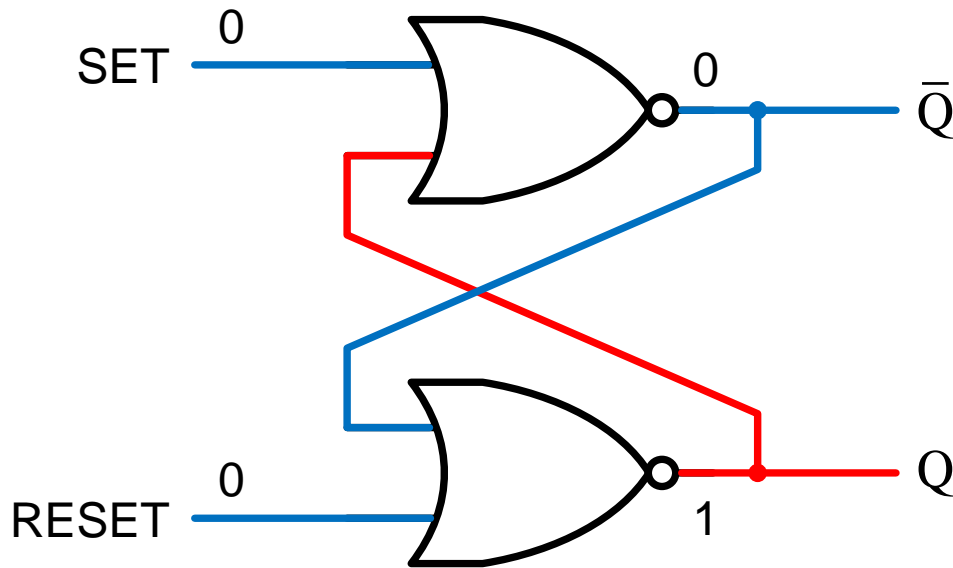
Latch SR (com portas NOR)

Análise do cenário 2

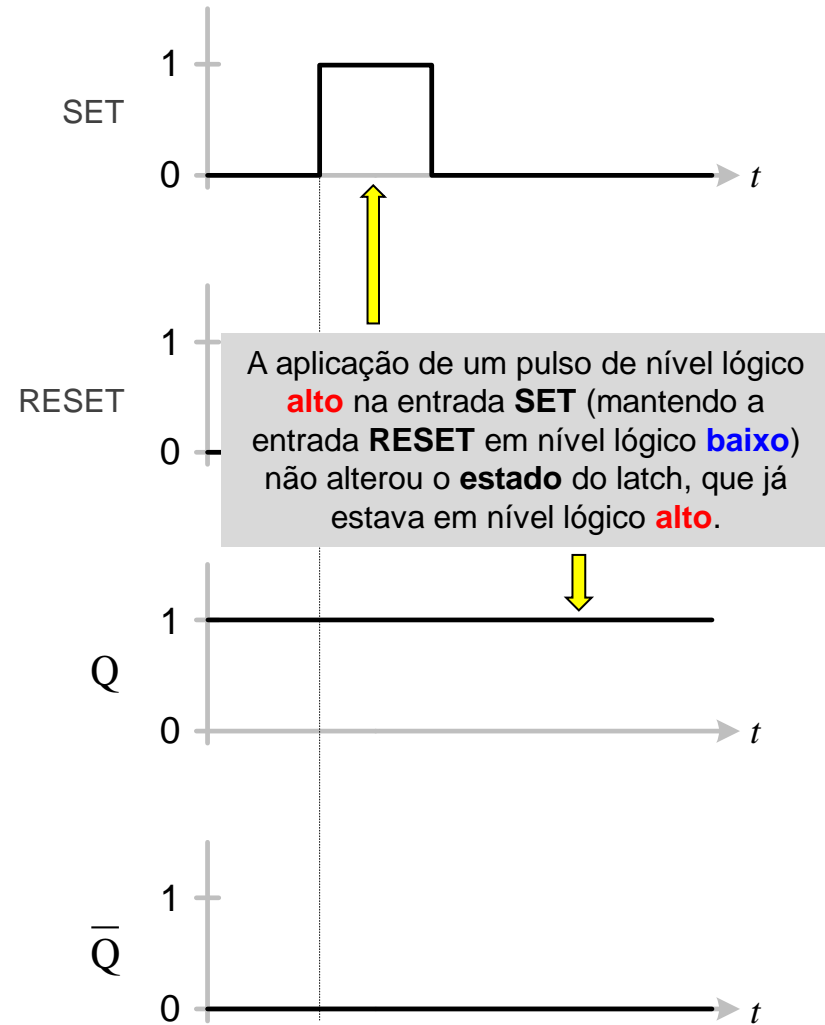


Latch SR (com portas NOR)

Análise do cenário 2

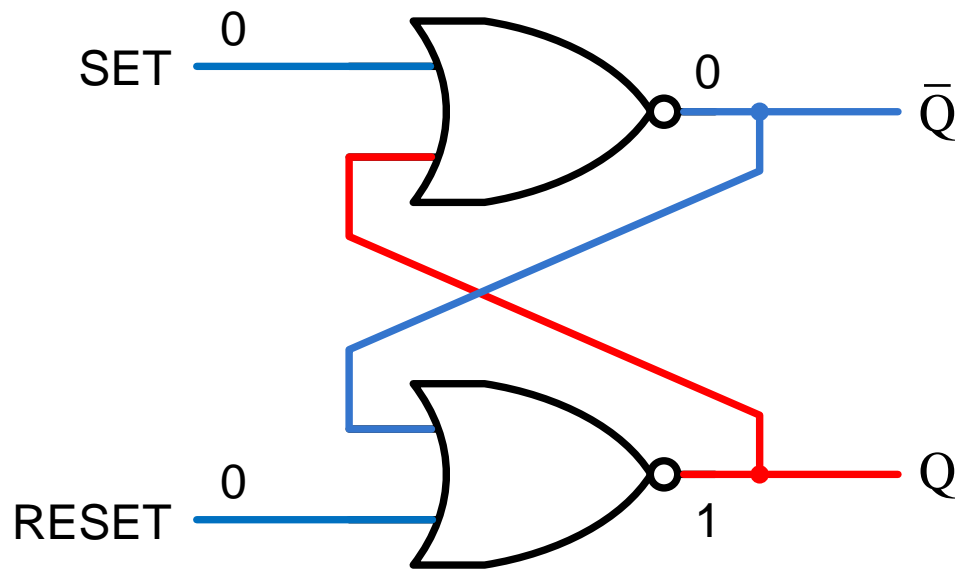


A	B	S
0	0	1
0	1	0
1	0	0
1	1	0



Latch SR (com portas NOR)

Conclusões relativas aos cenários 1 e 2

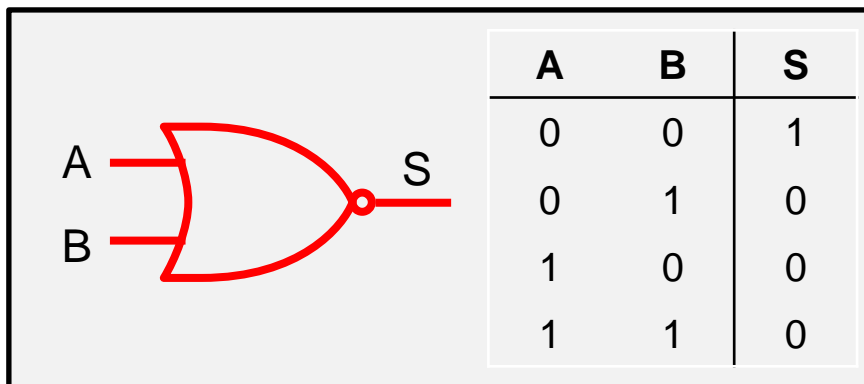
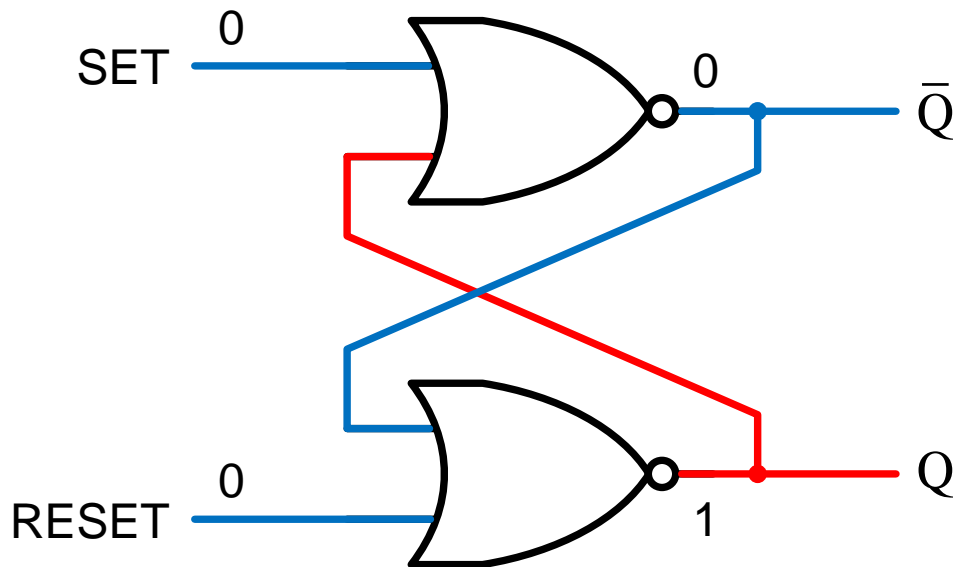


Conclusões:

- 1) A aplicação de um pulso de nível lógico **alto** na entrada **SET** (mantendo a entrada **RESET** em nível lógico **baixo**) leva o **estado** do latch para nível lógico **alto**.
- 2) Caso o **estado** do latch já esteja em nível lógico **alto**, **nada muda**.

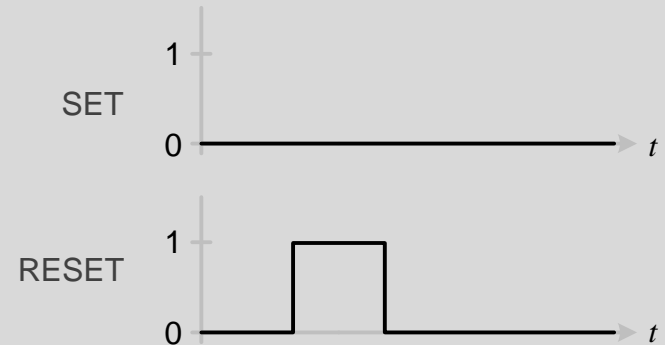
Latch SR (com portas NOR)

Análise do cenário 3



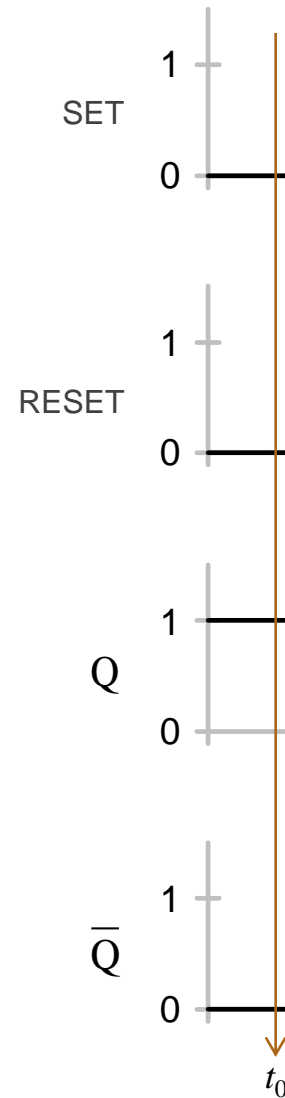
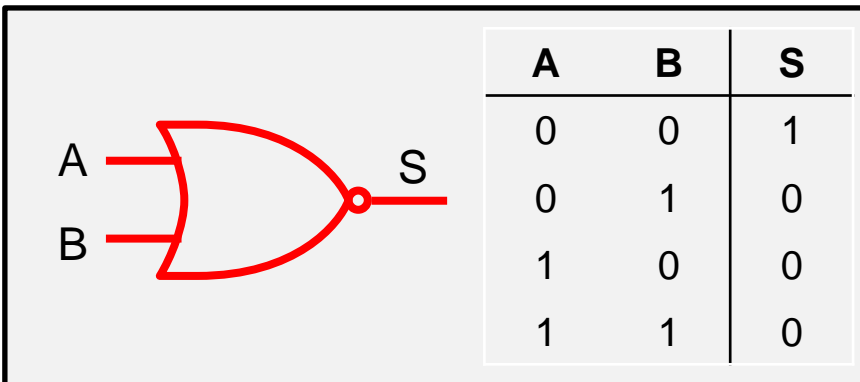
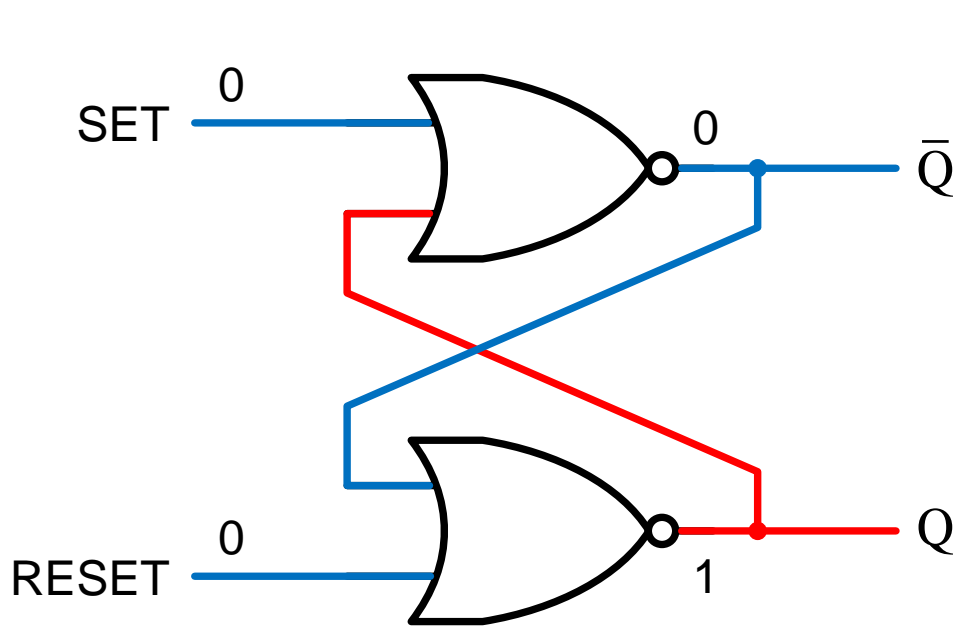
Cenário 3:

- Inicialmente, as entradas **SET** e **RESET** estão em nível lógico **baixo** e o **estado** do Latch (isto é, a saída Q), em nível lógico **alto**.
- Na entrada **RESET** é aplicado um pulso de nível lógico **alto**, mantendo a entrada **SET** no nível lógico **baixo**.



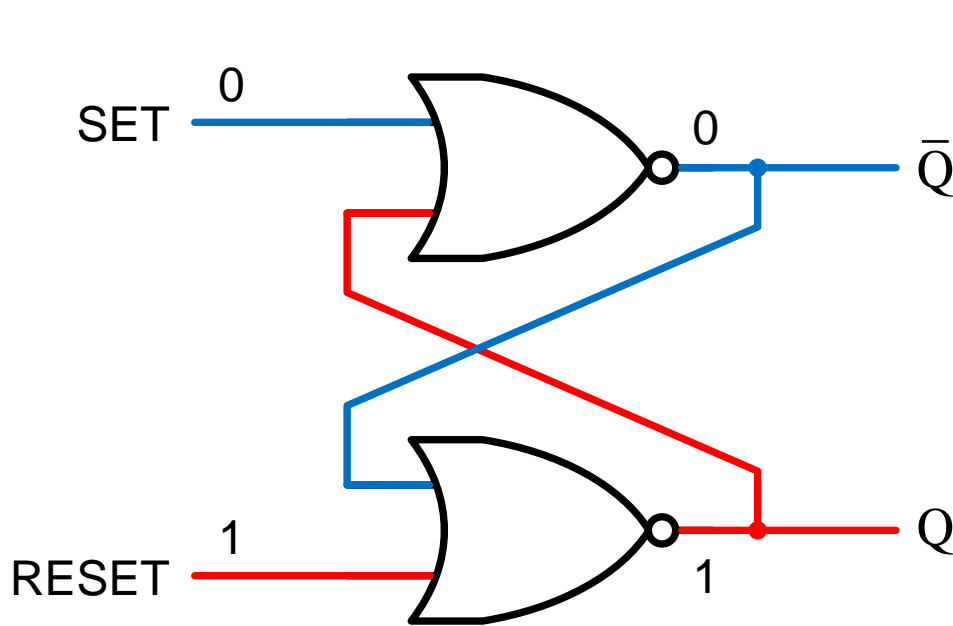
Latch SR (com portas NOR)

Análise do cenário 3

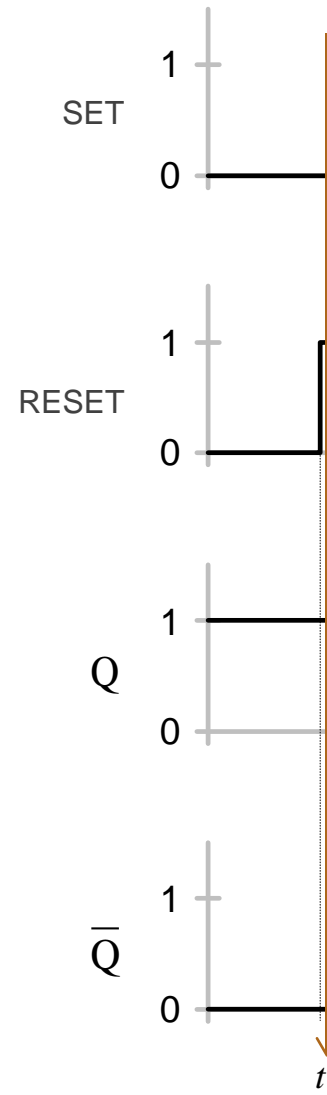


Latch SR (com portas NOR)

Análise do cenário 3

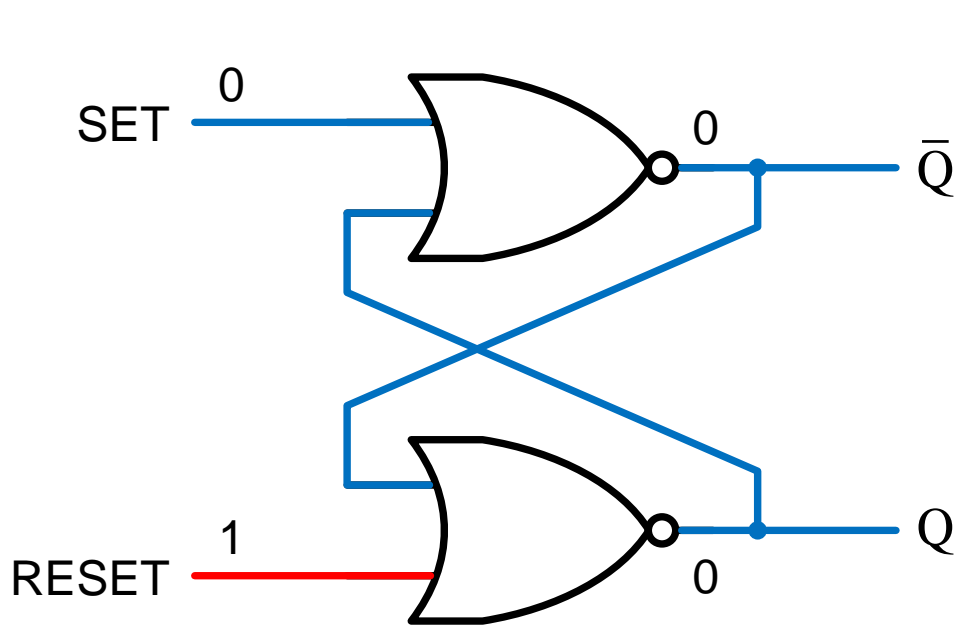


A	B	S
0	0	1
0	1	0
1	0	0
1	1	0

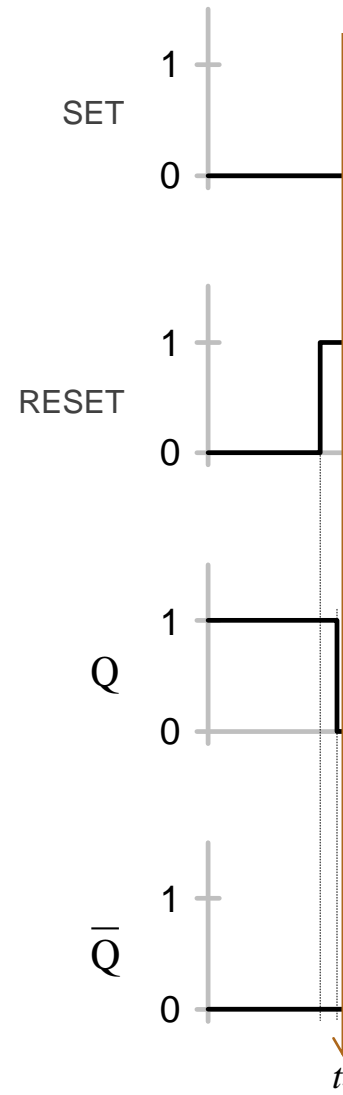


Latch SR (com portas NOR)

Análise do cenário 3

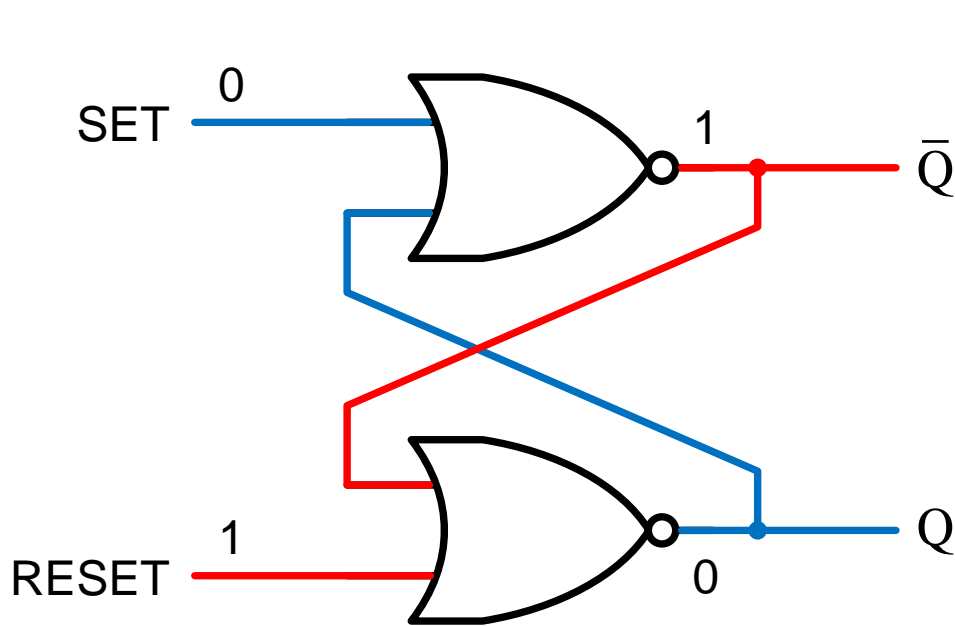


A	B	S
0	0	1
0	1	0
1	0	0
1	1	0

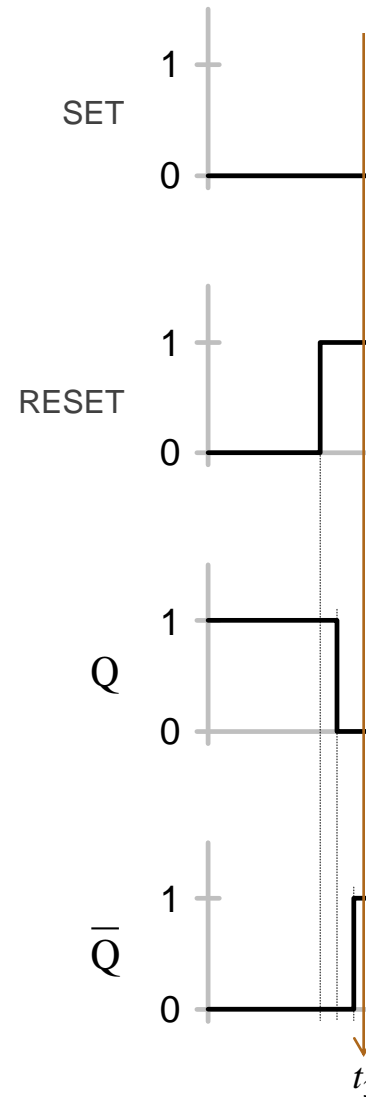


Latch SR (com portas NOR)

Análise do cenário 3

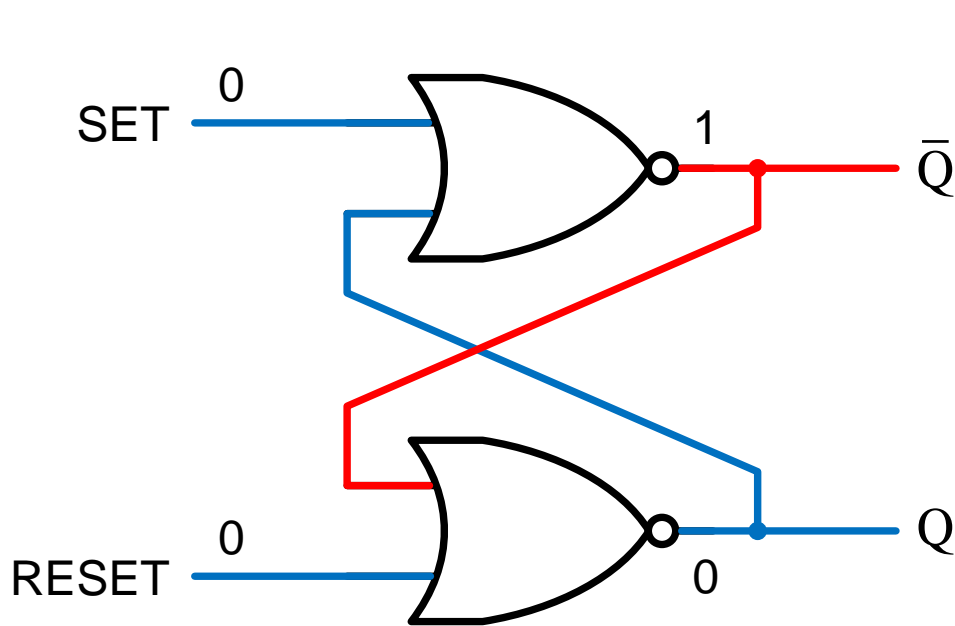


A	B	S
0	0	1
0	1	0
1	0	0
1	1	0

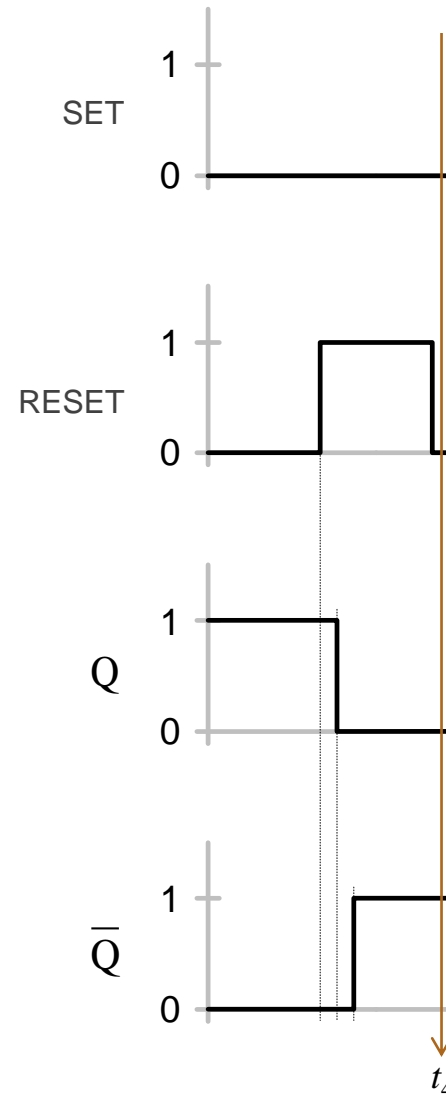


Latch SR (com portas NOR)

Análise do cenário 3

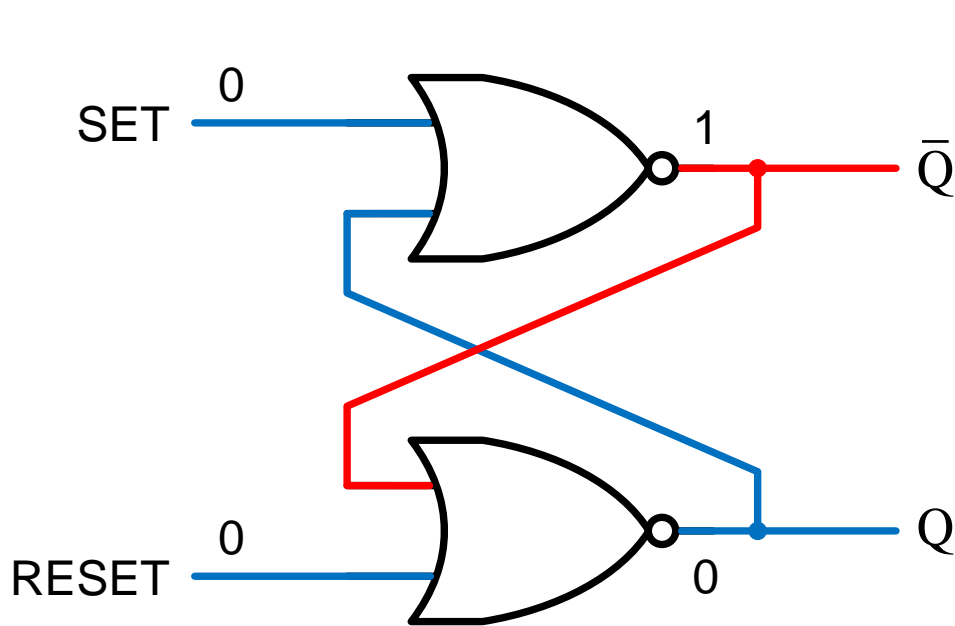


A	B	S
0	0	1
0	1	0
1	0	0
1	1	0

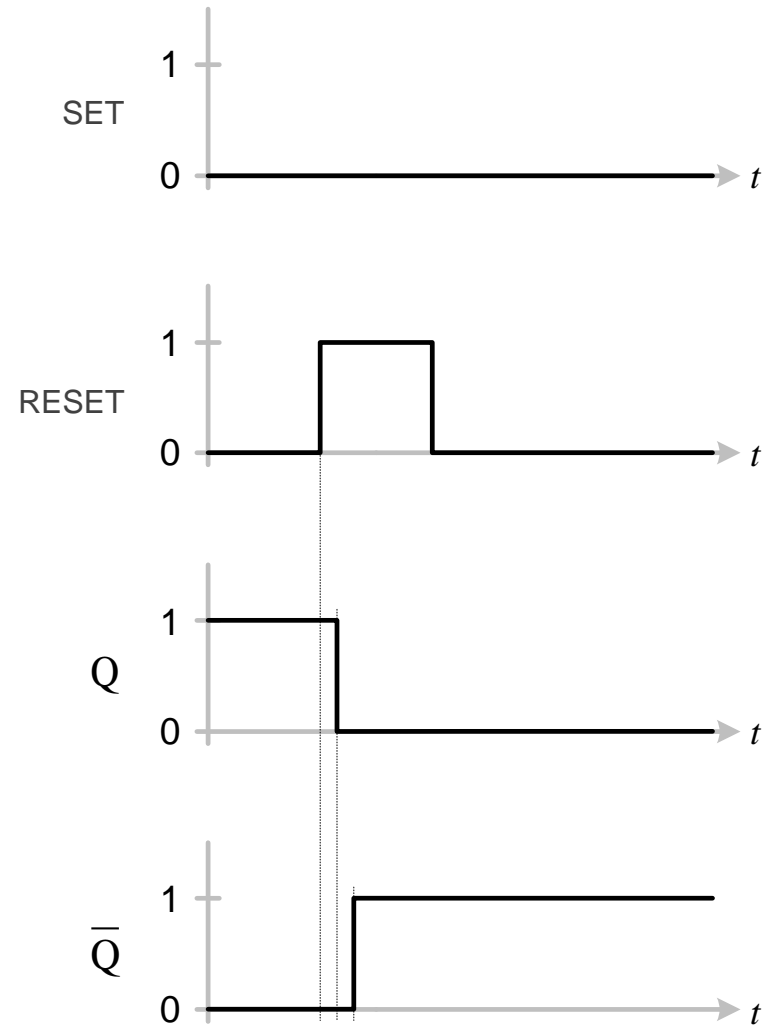


Latch SR (com portas NOR)

Análise do cenário 3

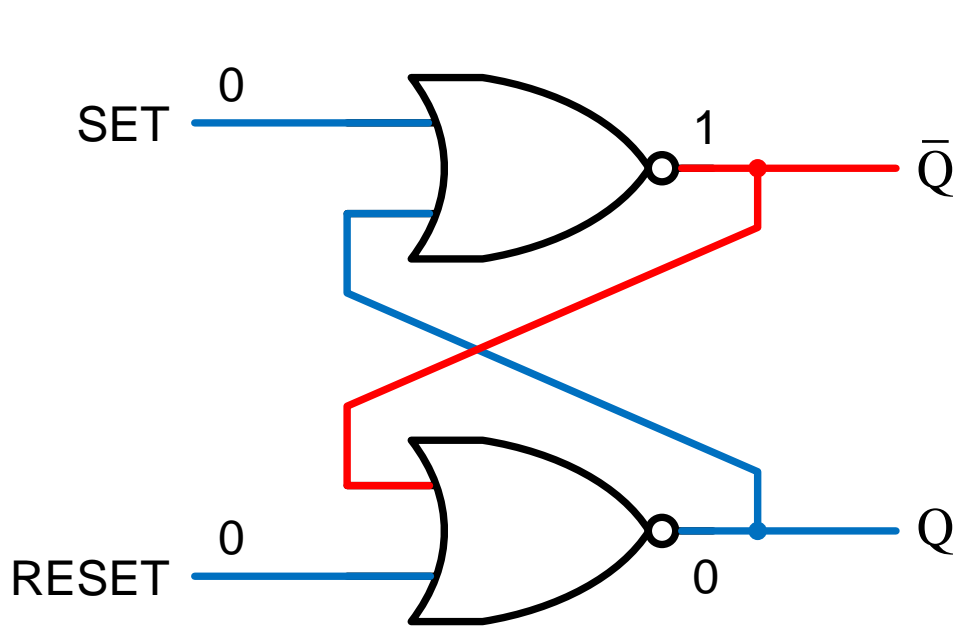


A	B	S
0	0	1
0	1	0
1	0	0
1	1	0

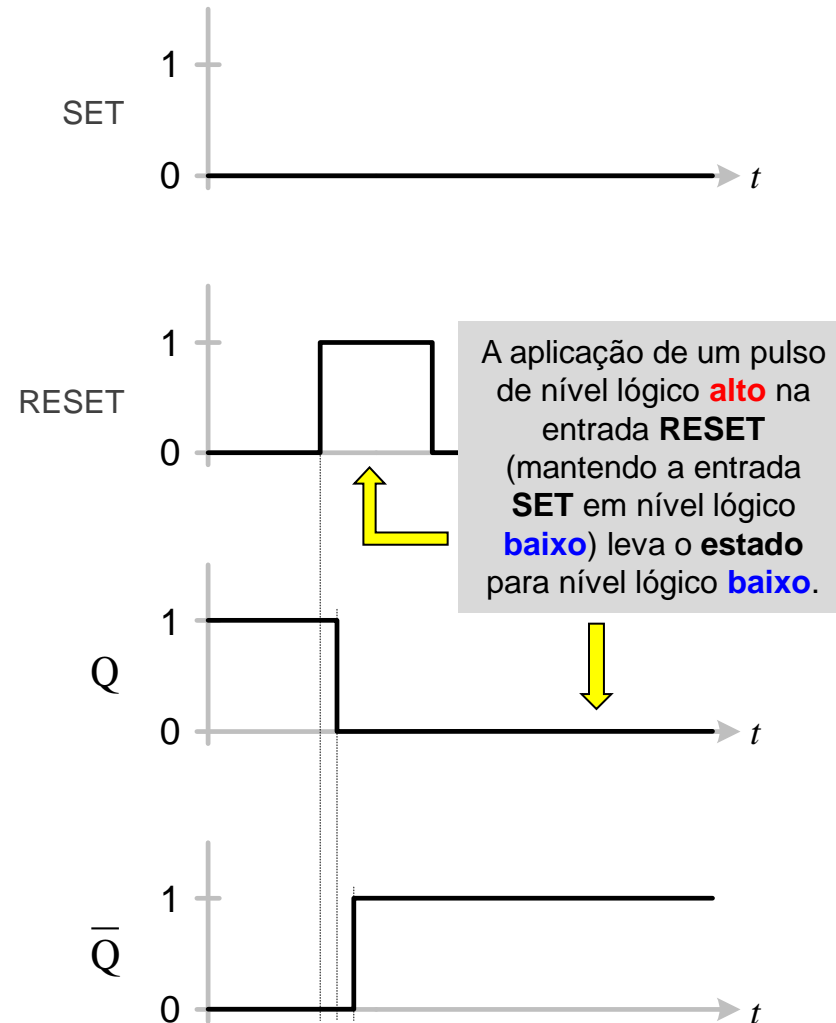


Latch SR (com portas NOR)

Análise do cenário 3

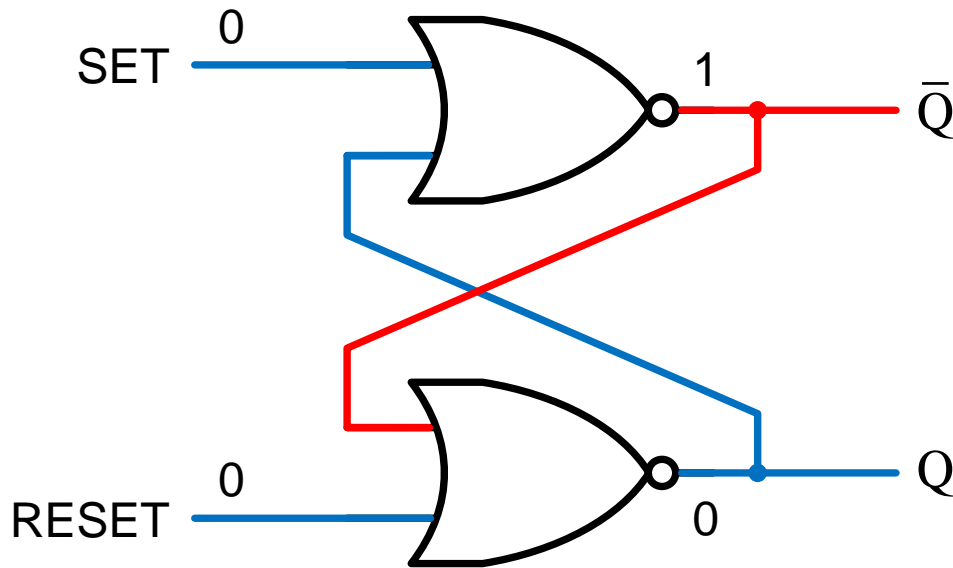



A	B	S
0	0	1
0	1	0
1	0	0
1	1	0



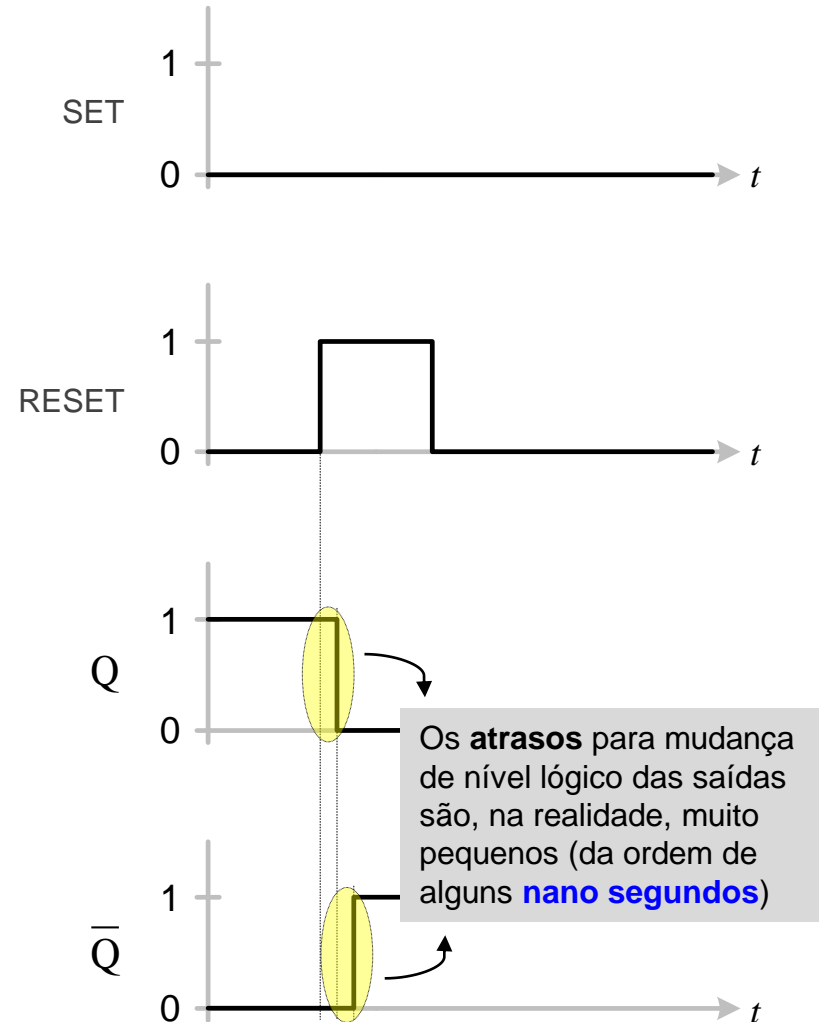
Latch SR (com portas NOR)

Análise do cenário 3



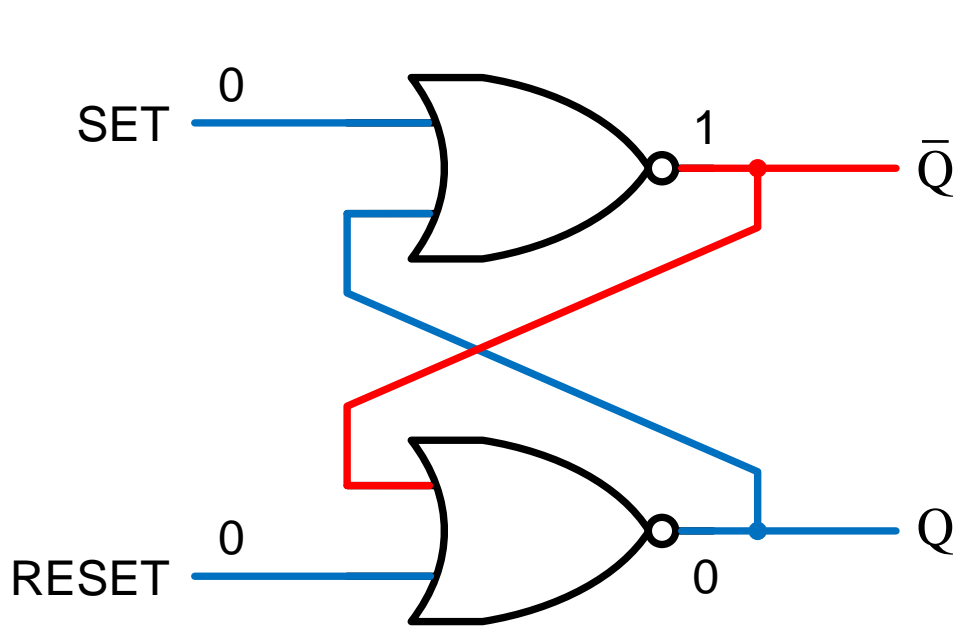


A	B	S
0	0	1
0	1	0
1	0	0
1	1	0

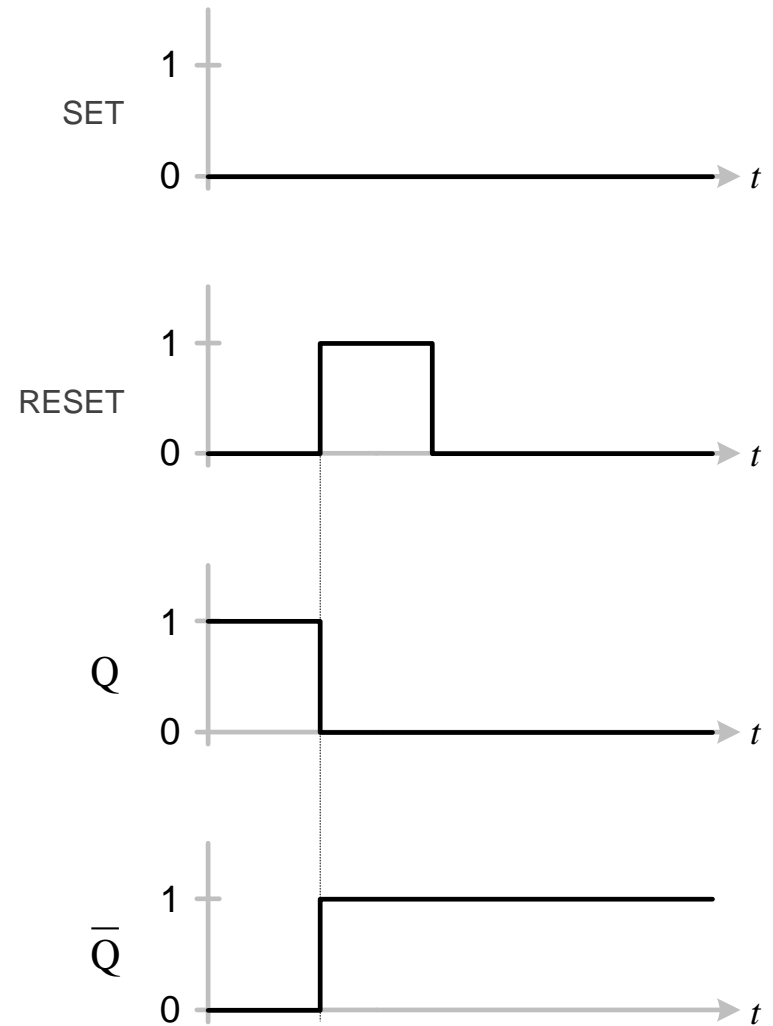


Latch SR (com portas NOR)

Análise do cenário 3

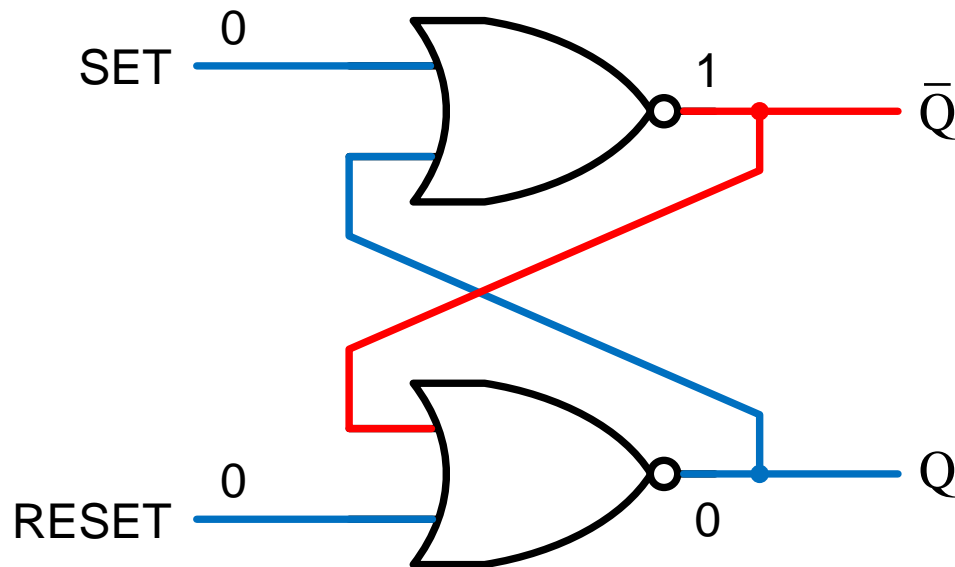



A	B	S
0	0	1
0	1	0
1	0	0
1	1	0



Latch SR (com portas NOR)

Análise do cenário 4

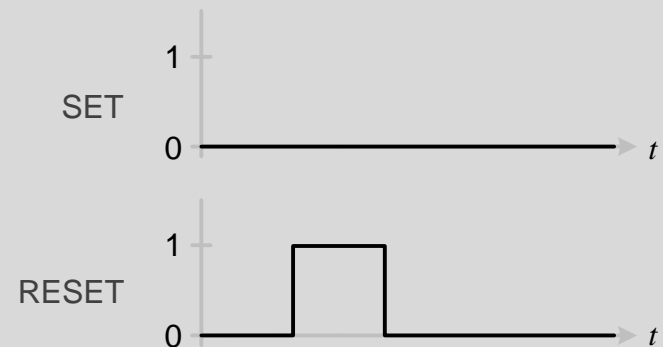




A	B	S
0	0	1
0	1	0
1	0	0
1	1	0

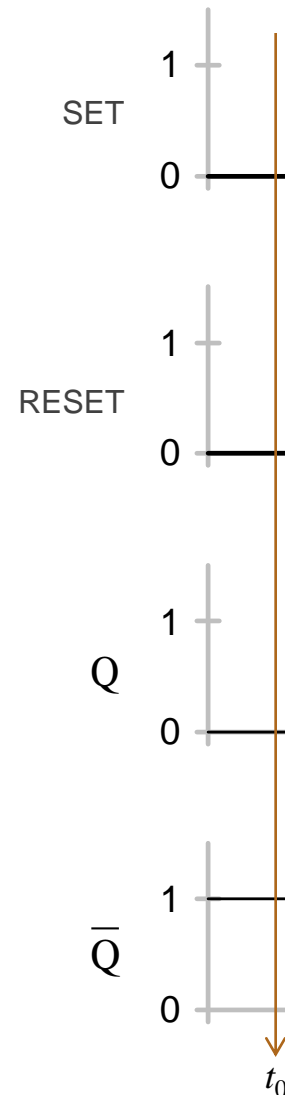
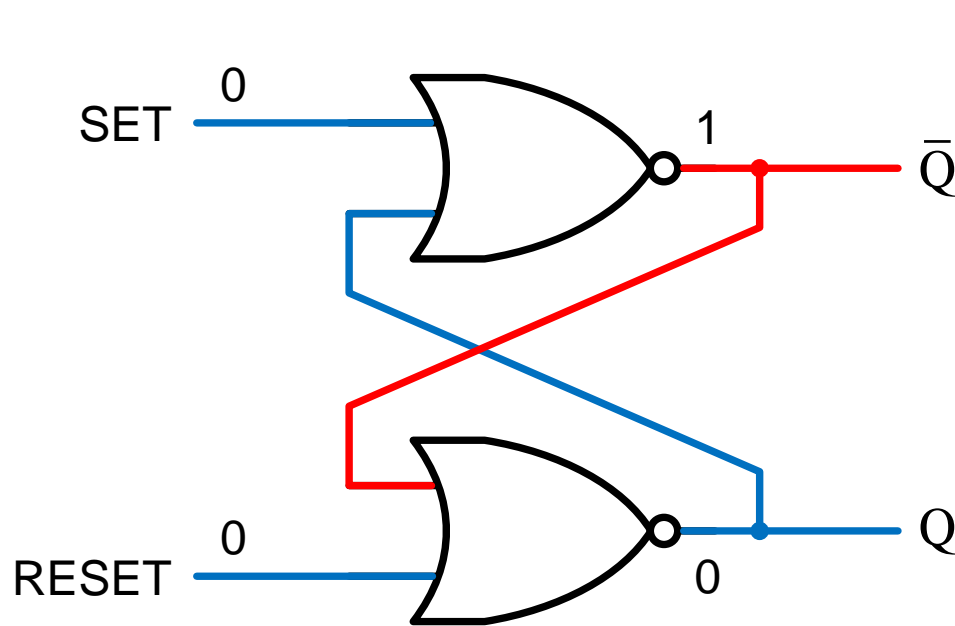
Cenário 4:

- Inicialmente, as entradas **SET** e **RESET** estão em nível lógico **baixo** e o **estado** do Latch (isto é, a saída Q), em nível lógico **baixo**.
- Na entrada **RESET** é aplicado um pulso de nível lógico **alto**, mantendo a entrada **SET** no nível lógico **baixo**.



Latch SR (com portas NOR)

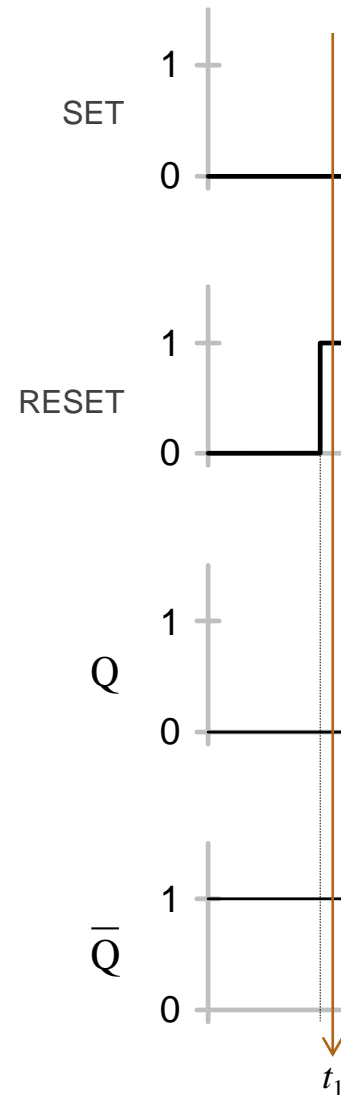
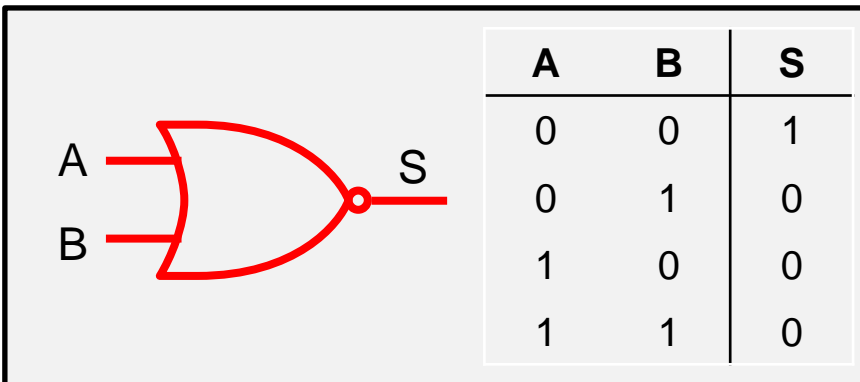
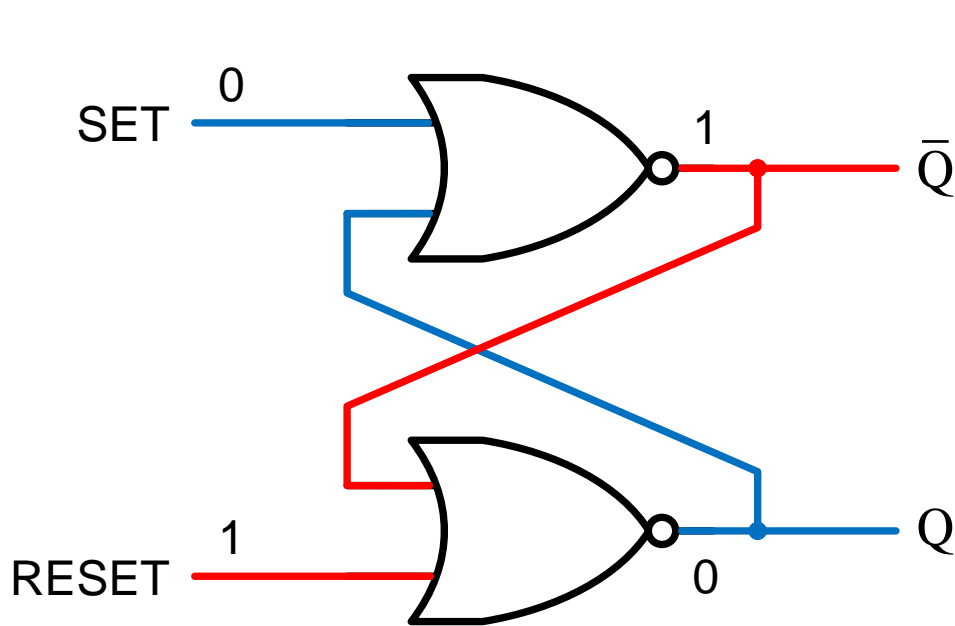
Análise do cenário 4



A	B	S
0	0	1
0	1	0
1	0	0
1	1	0

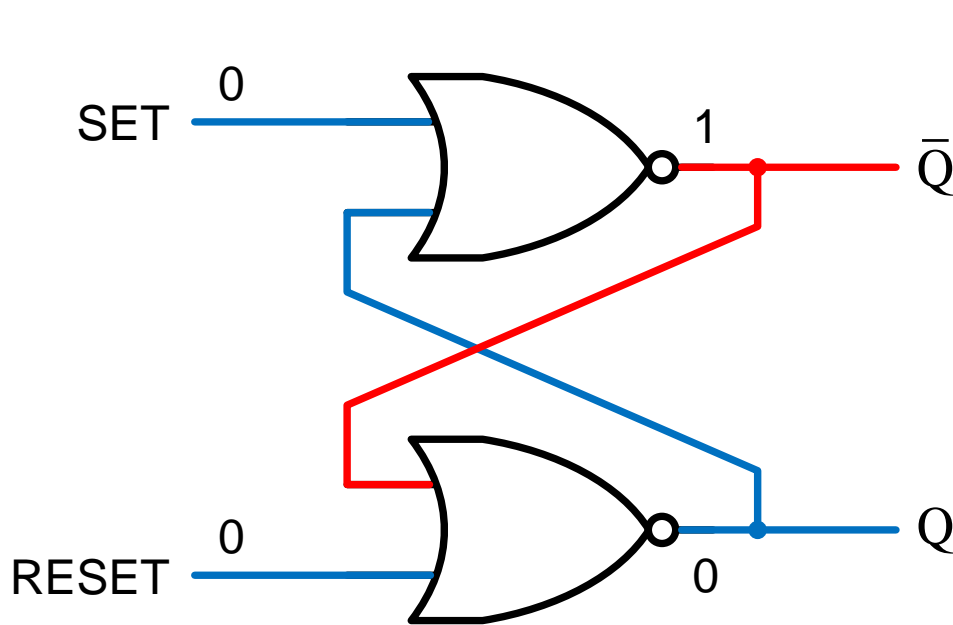
Latch SR (com portas NOR)

Análise do cenário 4

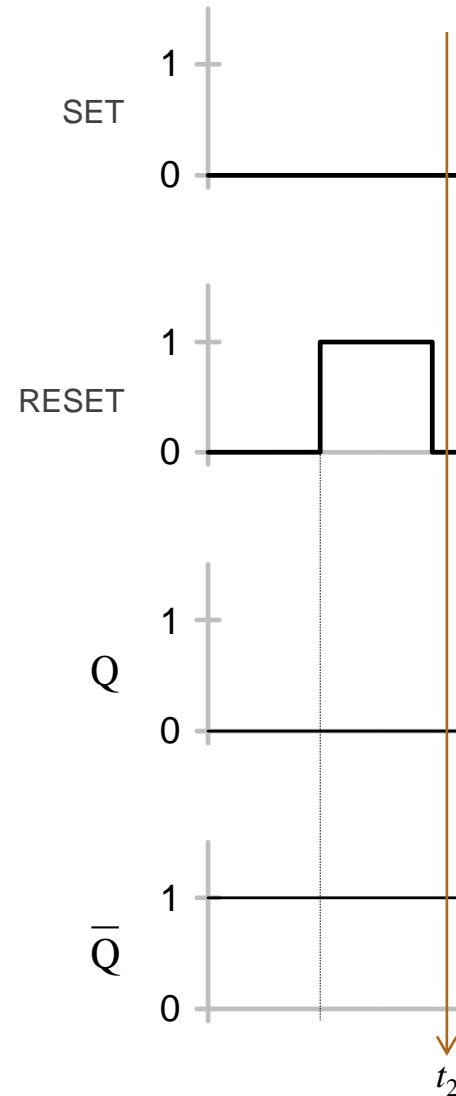


Latch SR (com portas NOR)

Análise do cenário 4

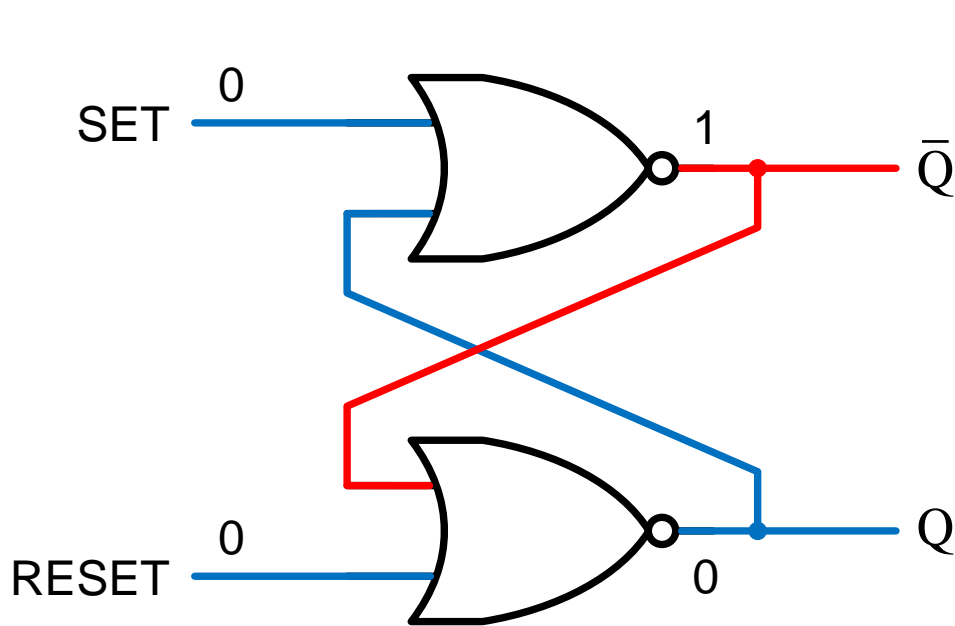


A	B	S
0	0	1
0	1	0
1	0	0
1	1	0

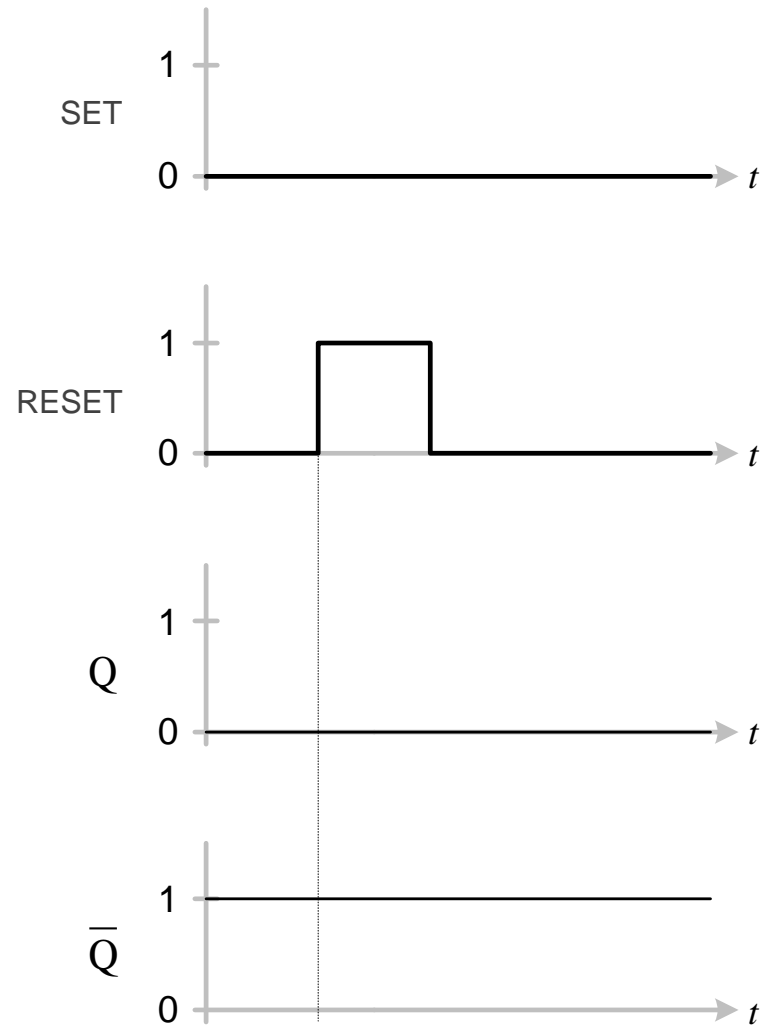


Latch SR (com portas NOR)

Análise do cenário 4

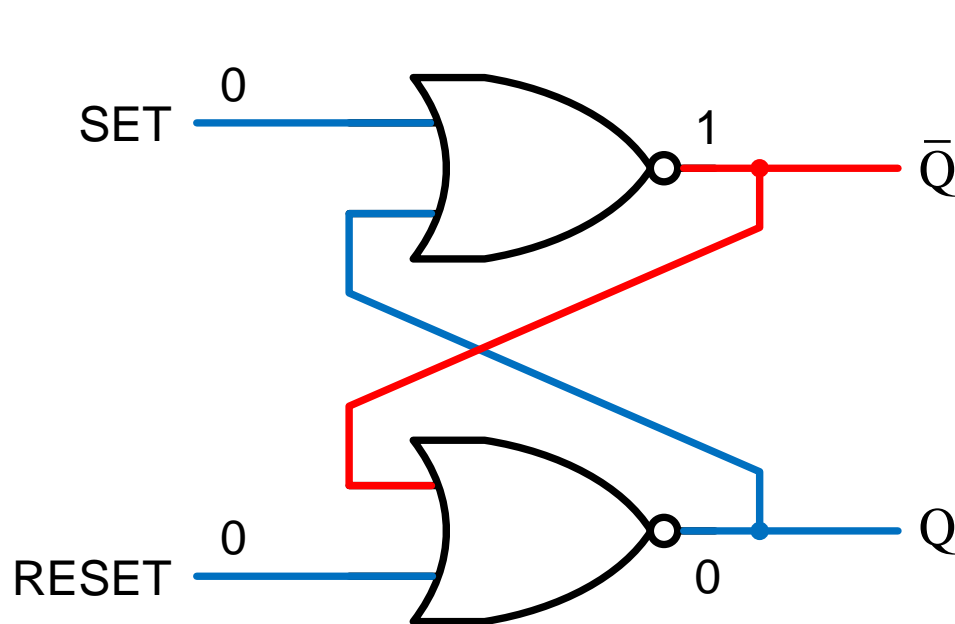


A	B	S
0	0	1
0	1	0
1	0	0
1	1	0

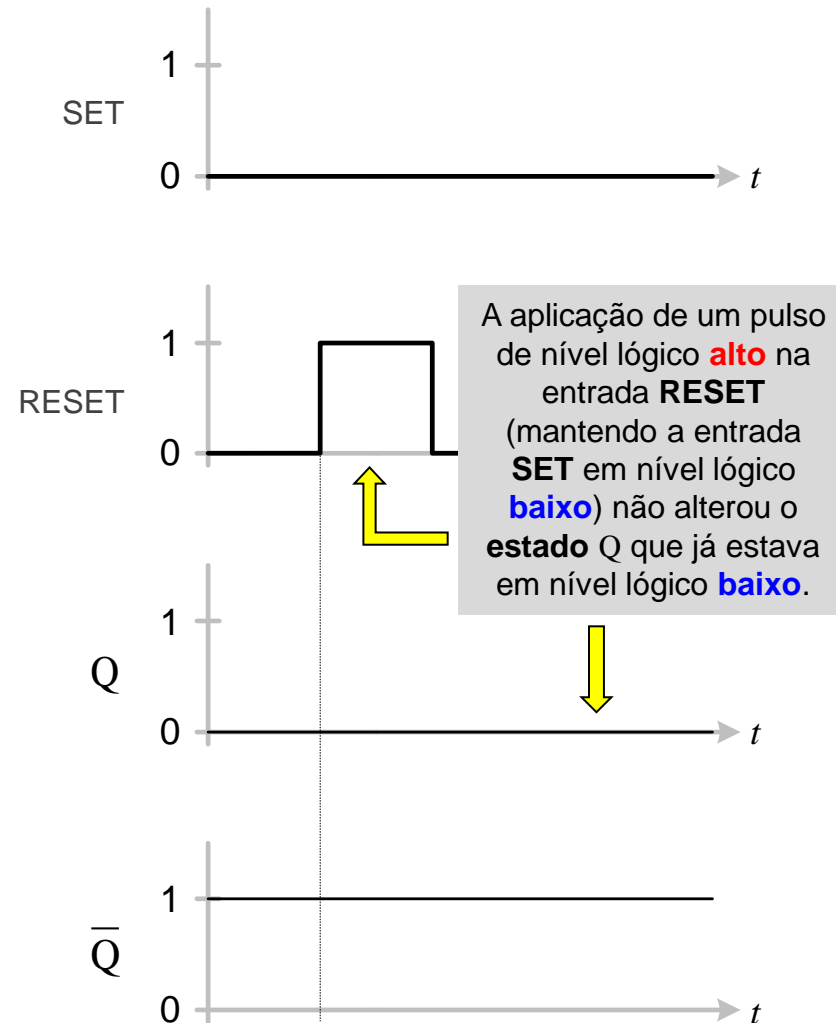


Latch SR (com portas NOR)

Análise do cenário 4

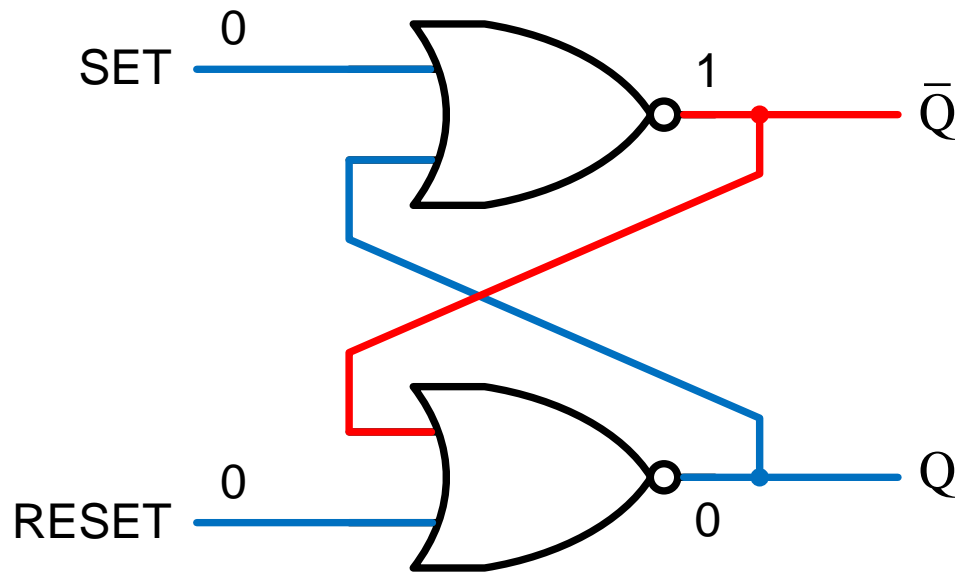


A	B	S
0	0	1
0	1	0
1	0	0
1	1	0



Latch SR (com portas NOR)

Conclusões relativas aos cenários 3 e 4

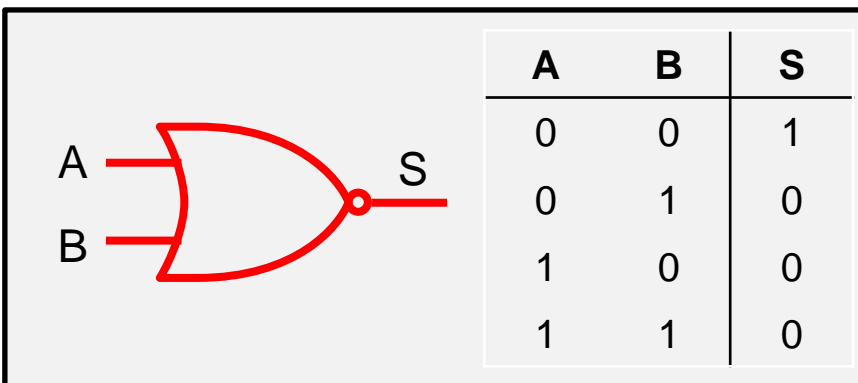
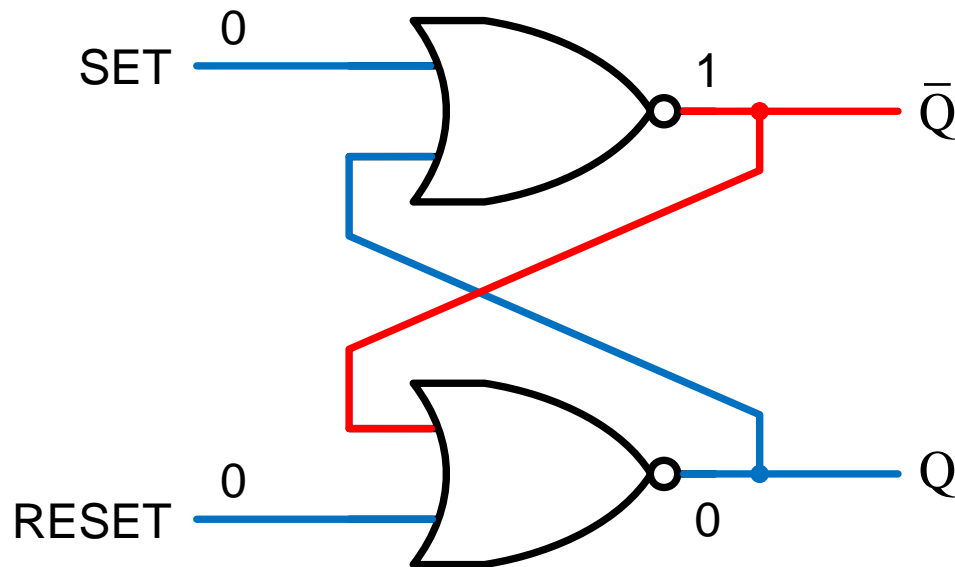


Conclusões:

- 1) A aplicação de um pulso de nível lógico **alto** na entrada **RESET** (mantendo a entrada **SET** em nível lógico **baixo**) leva o **estado** do latch para nível lógico **baixo**.
- 2) Caso o **estado** do latch já esteja em nível lógico **baixo**, **nada muda**.

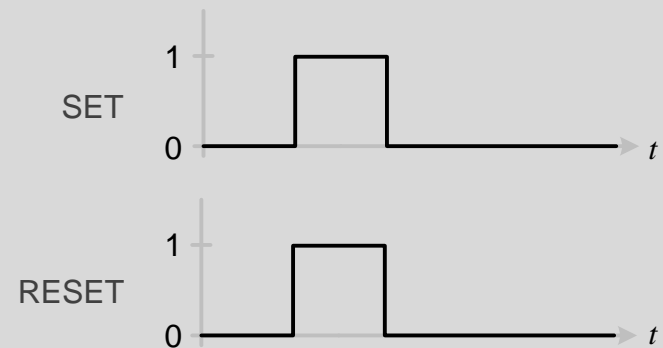
Latch SR (com portas NOR)

Análise do cenário 5



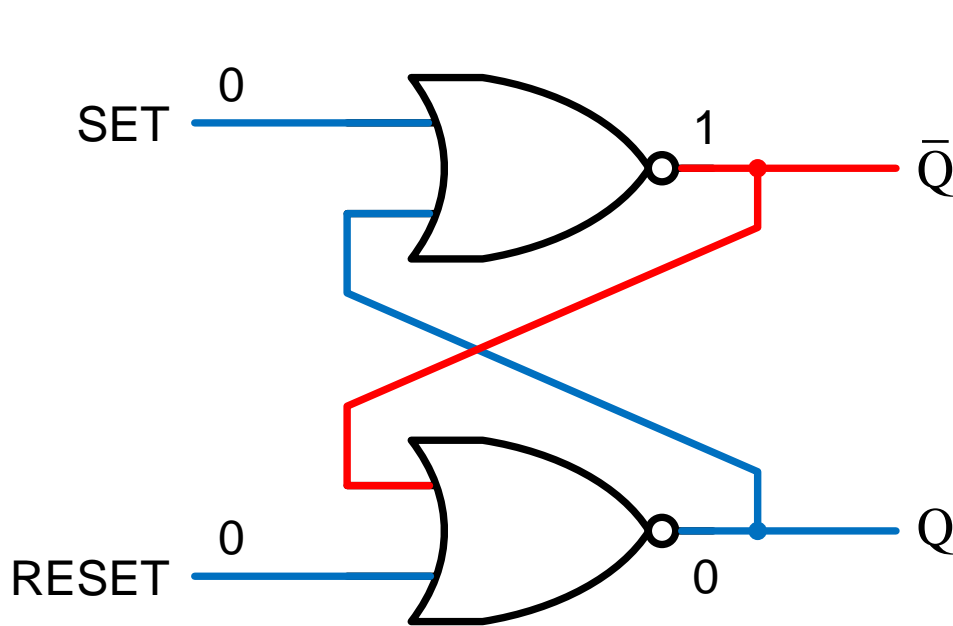
Cenário 5:

- a) Inicialmente, as entradas **SET** e **RESET** estão em nível lógico **baixo** e o **estado** do Latch (isto é, a saída Q), em nível lógico **baixo**.
- b) Na entrada **SET** e **RESET** é aplicado um pulso de nível lógico **alto**.

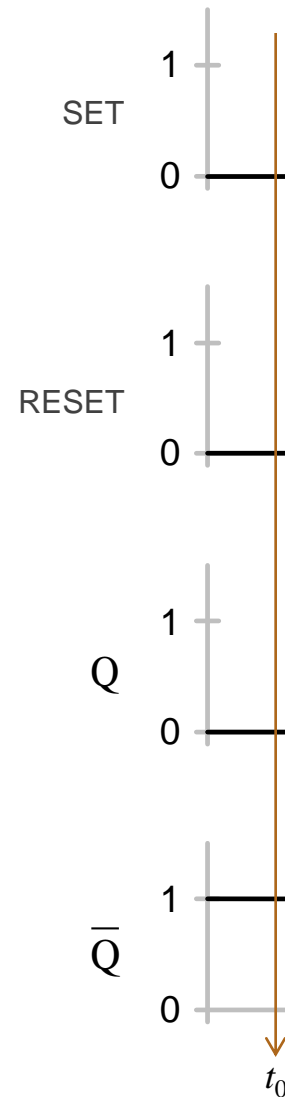


Latch SR (com portas NOR)

Análise do cenário 5

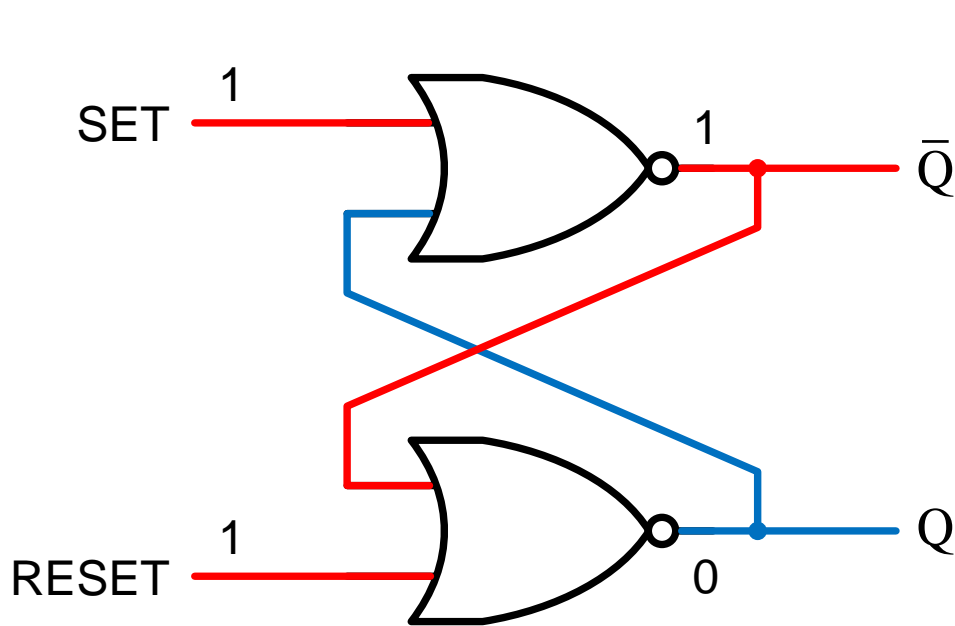


A	B	S
0	0	1
0	1	0
1	0	0
1	1	0

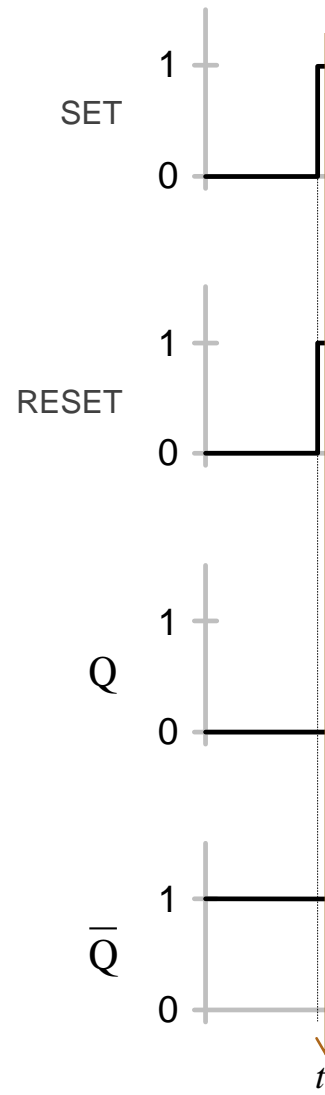


Latch SR (com portas NOR)

Análise do cenário 5

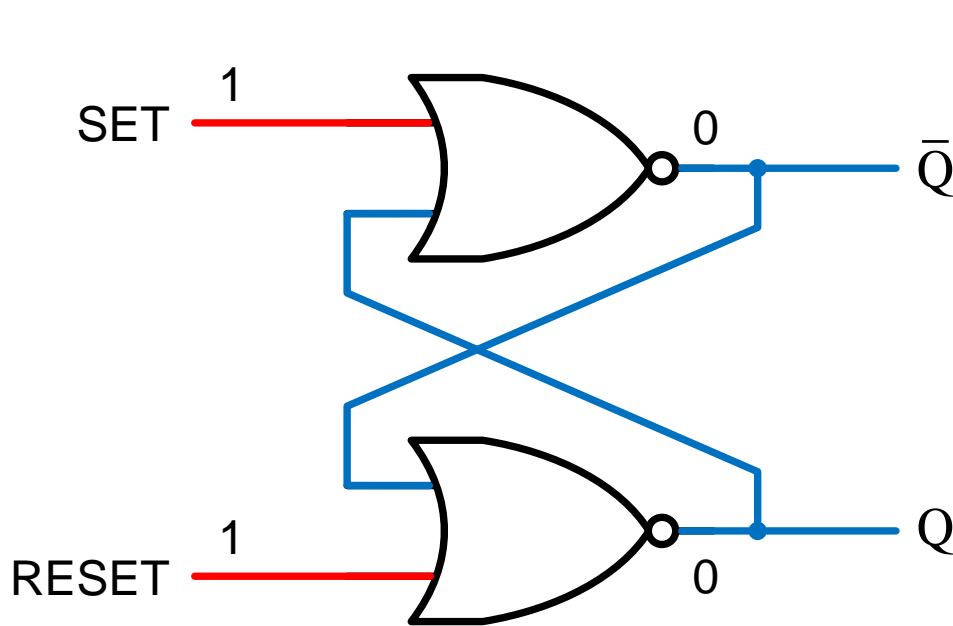


A	B	S
0	0	1
0	1	0
1	0	0
1	1	0

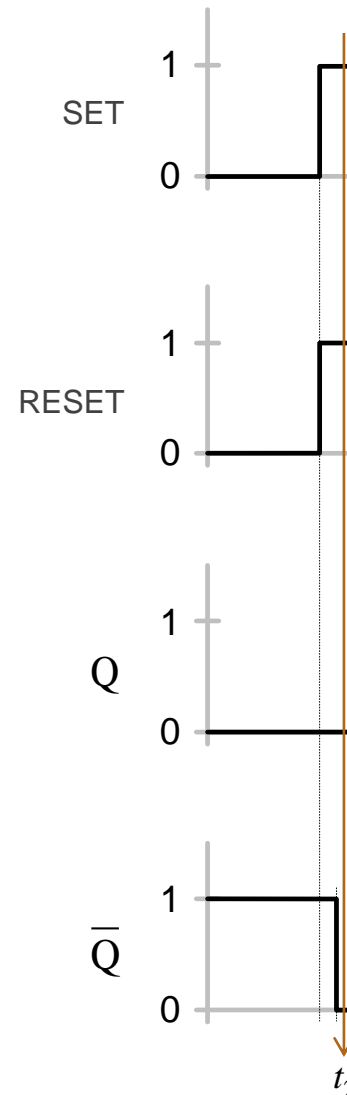


Latch SR (com portas NOR)

Análise do cenário 5

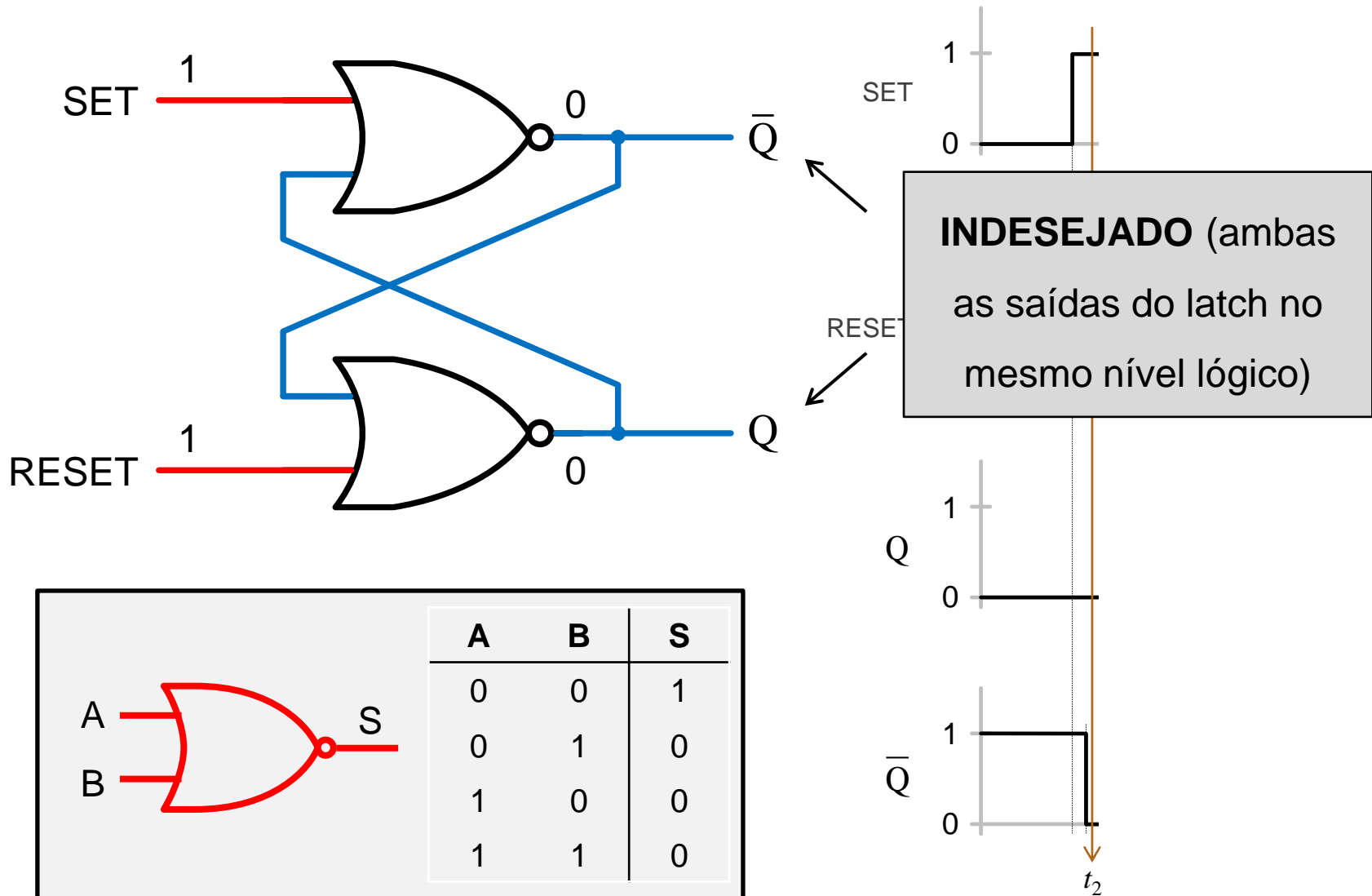


A	B	S
0	0	1
0	1	0
1	0	0
1	1	0



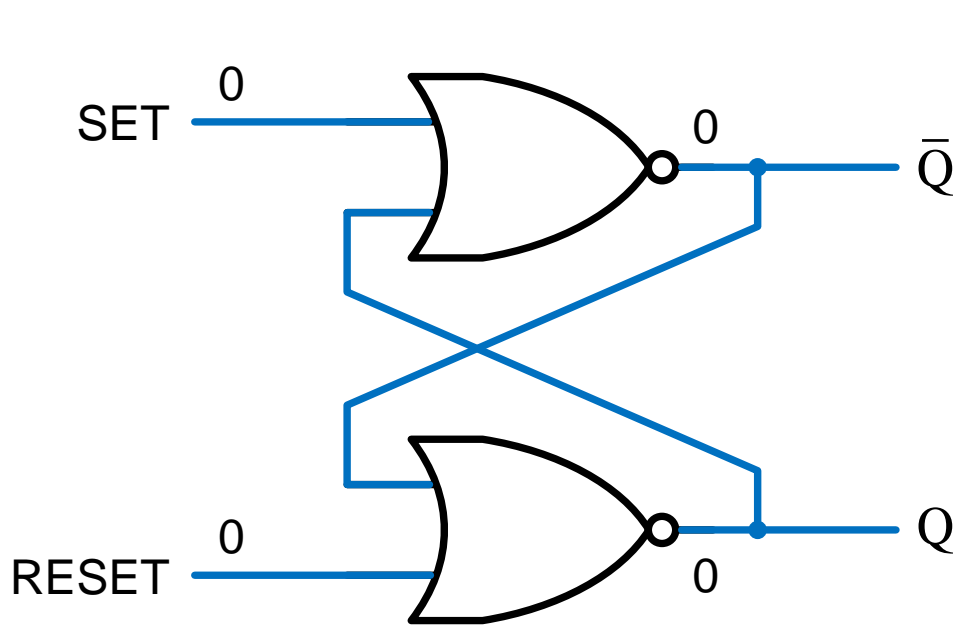
Latch SR (com portas NOR)

Análise do cenário 5

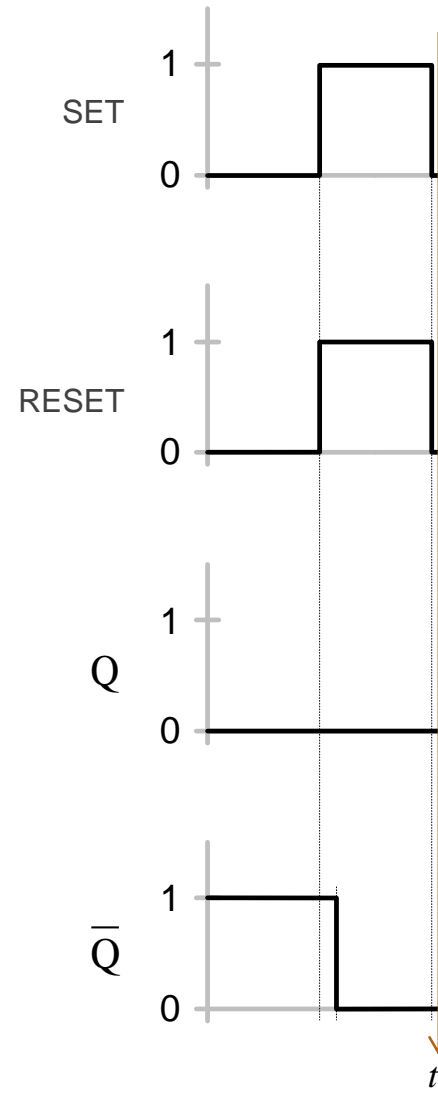


Latch SR (com portas NOR)

Análise do cenário 5

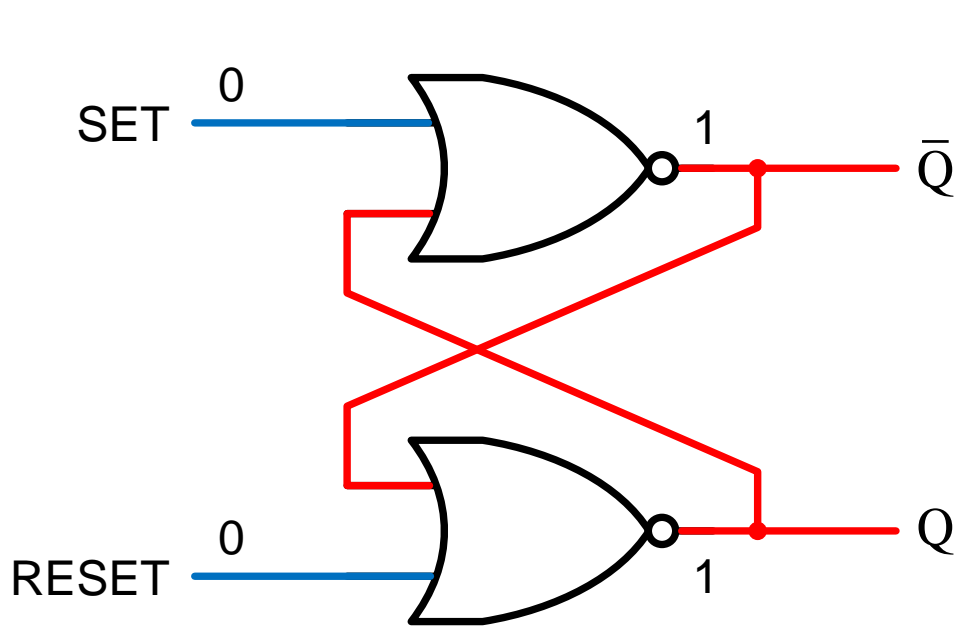


A	B	S
0	0	1
0	1	0
1	0	0
1	1	0

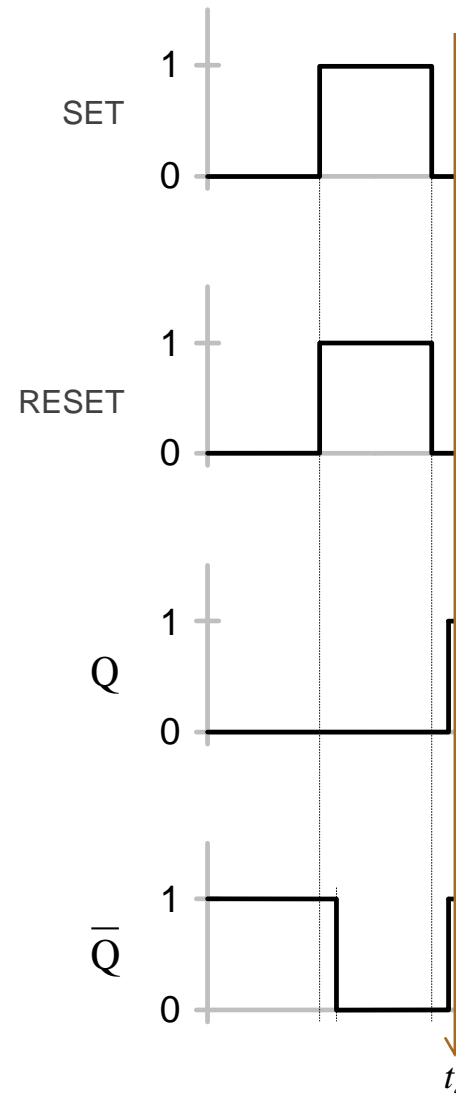


Latch SR (com portas NOR)

Análise do cenário 5

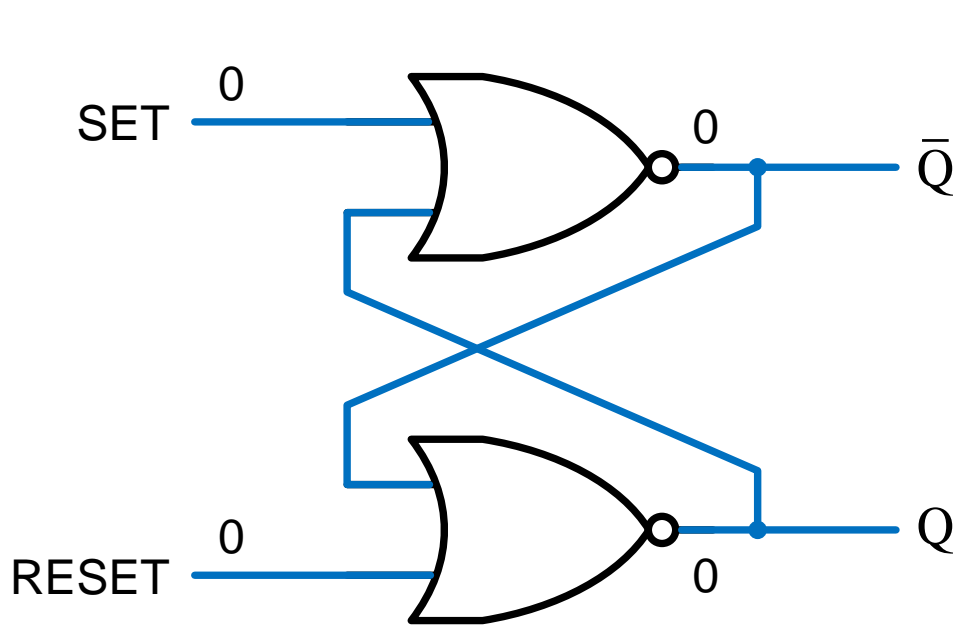


A	B	S
0	0	1
0	1	0
1	0	0
1	1	0

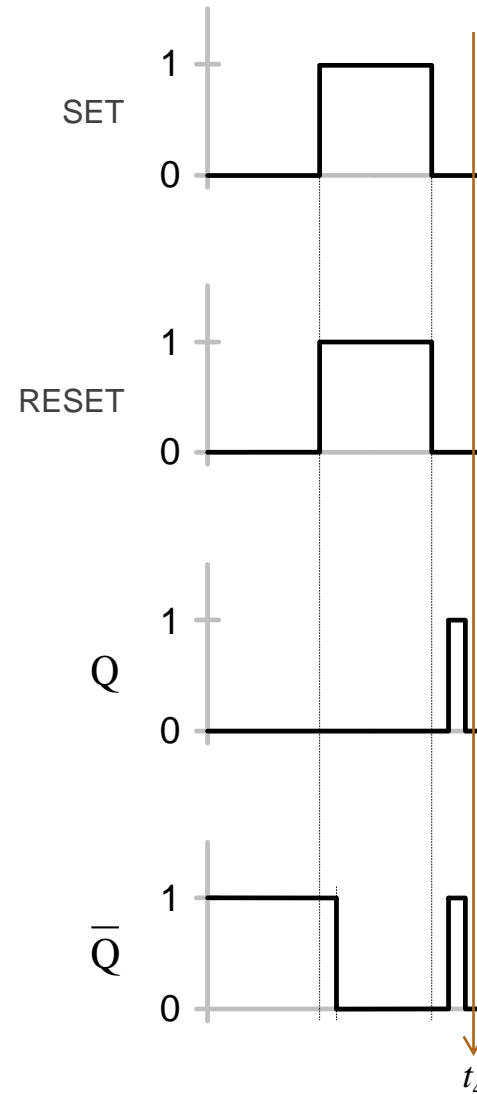


Latch SR (com portas NOR)

Análise do cenário 5

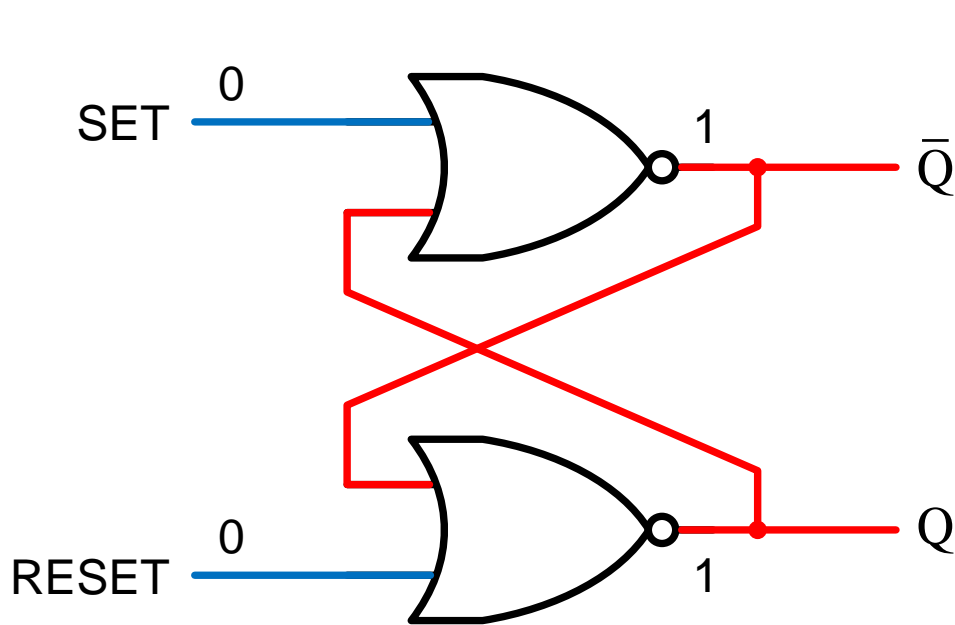


A	B	S
0	0	1
0	1	0
1	0	0
1	1	0

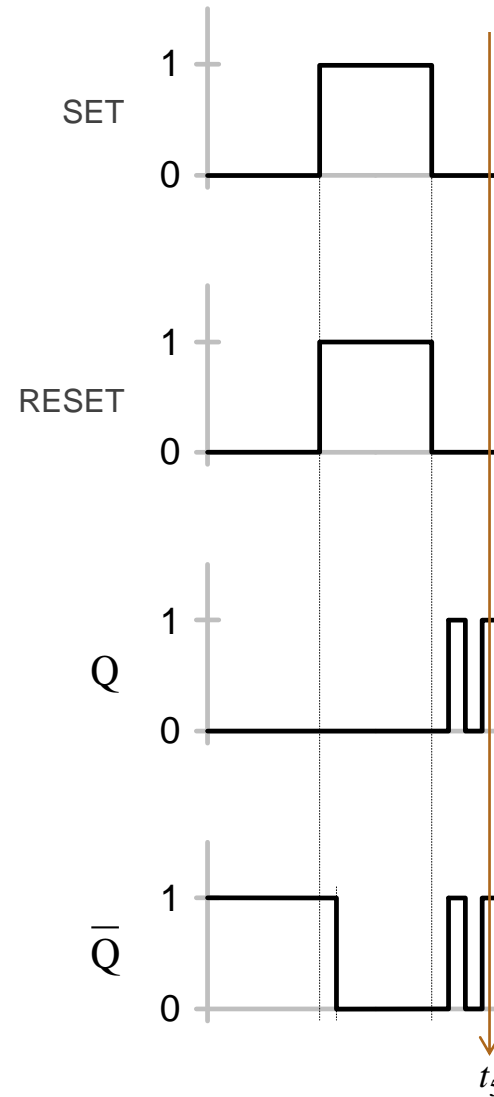


Latch SR (com portas NOR)

Análise do cenário 5

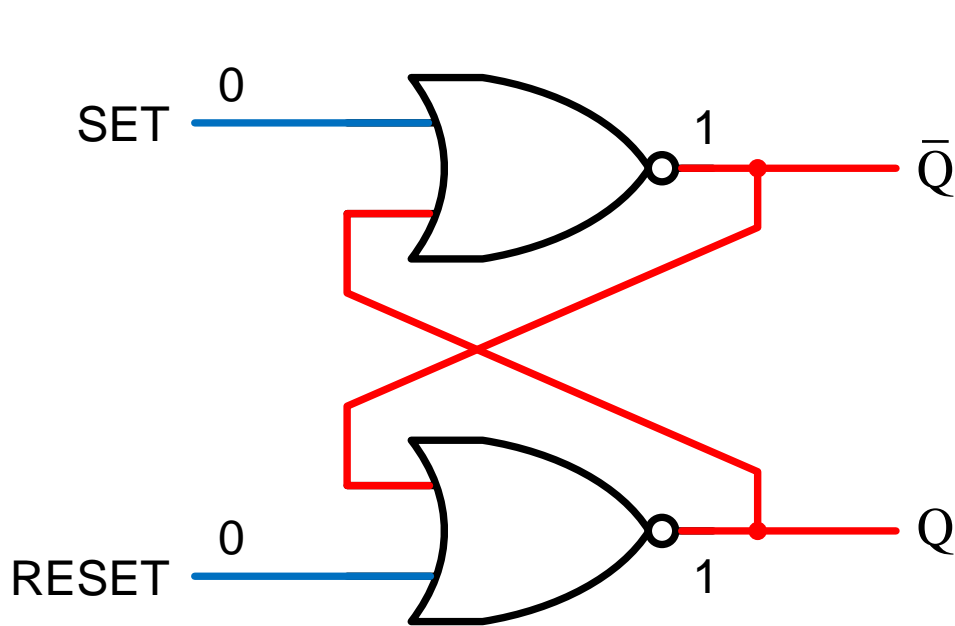


A	B	S
0	0	1
0	1	0
1	0	0
1	1	0

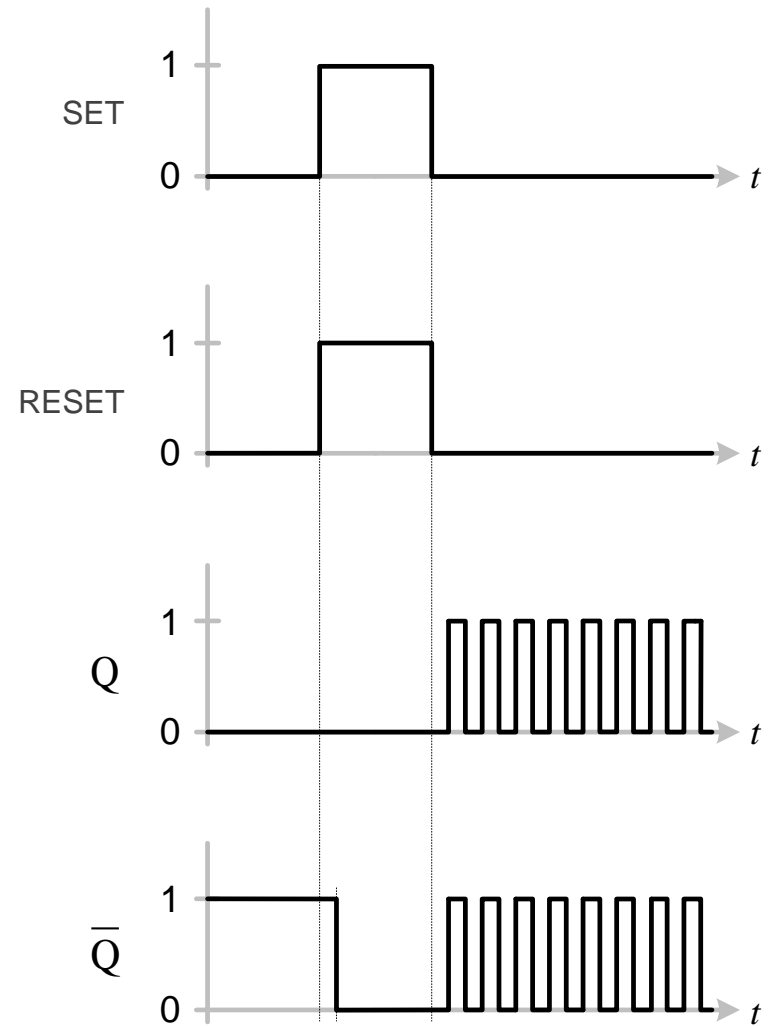


Latch SR (com portas NOR)

Análise do cenário 5

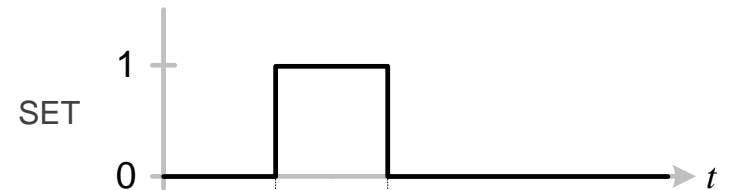
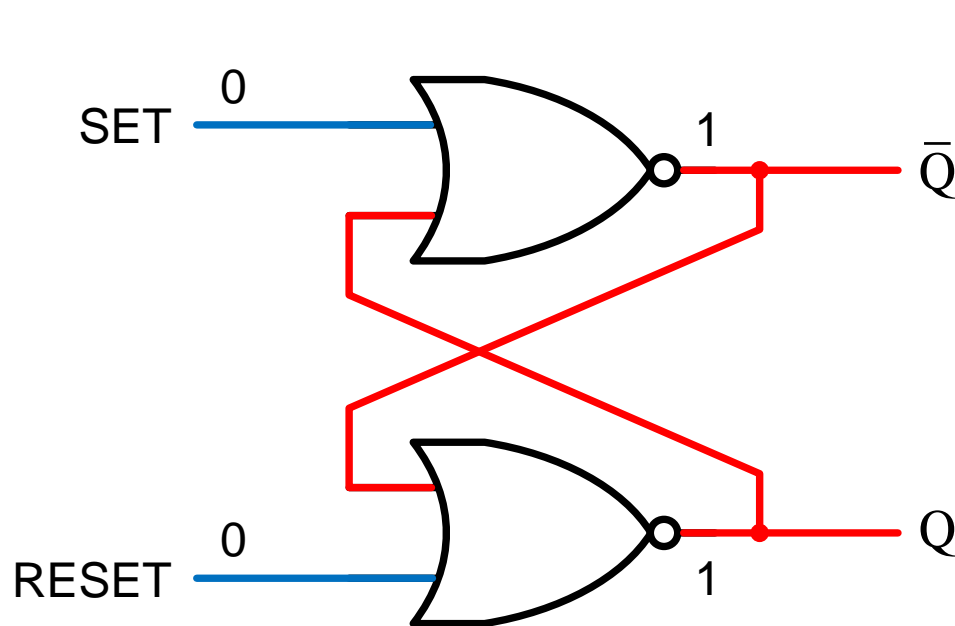


A	B	S
0	0	1
0	1	0
1	0	0
1	1	0

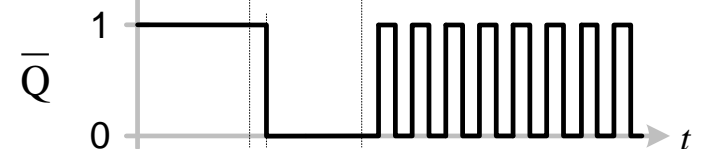
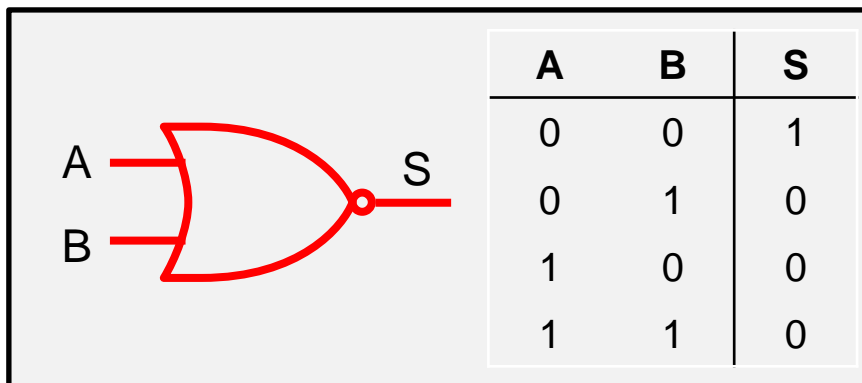


Latch SR (com portas NOR)

Análise do cenário 5



Em circuitos **reais**, o estado final do latch irá se estabilizar em 0 ou 1, pois os tempos de resposta de cada porta não são exatamente iguais. Além disso, dificilmente as entradas **SET** and **RESET** retornarão para nível lógico **alto** exatamente no mesmo instante.



Latch SR (com portas NOR)

Tabela verdade (tabela de transição)

- A **tabela verdade** descreve o comportamento do Latch SR

SET	RESET	Saída Q
0	0	Q_0
1	0	$Q = 1$
0	1	$Q = 0$
1	1	Proibido

← Estado anterior

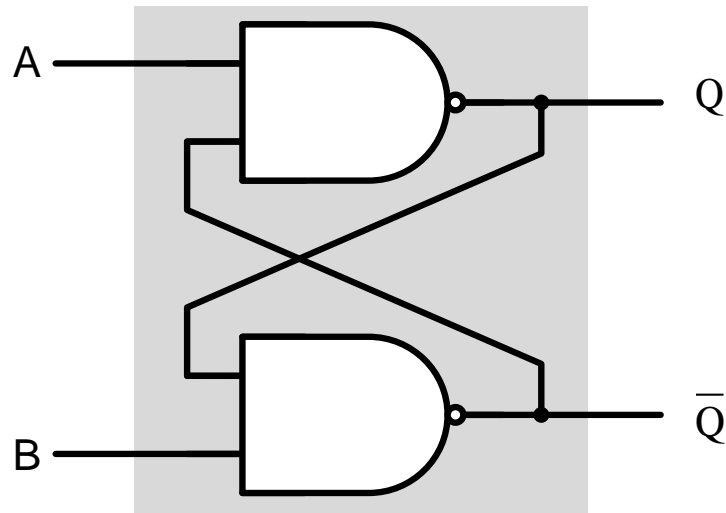
Procedimento para análise de latches

- Determinar o nível lógico de “descanso” das entradas. Ou seja, determinar o nível lógico (0 ou 1) nas entradas que permite que ambos os **estados** ($Q = 1$ ou $Q = 0$) sejam possíveis.
- Avaliar o comportamento do latch nos cenários*:
 - 1ª entrada sai momentaneamente do nível lógico de “descanso”, permanecendo a 2ª entrada inalterada.
 - 2ª entrada sai momentaneamente do nível lógico de “descanso”, permanecendo a 1ª entrada inalterada.
 - As duas entradas saem momentaneamente do nível lógico “descanso”.

*cada análise deve ser realizada para as duas **condições iniciais** possíveis, isto é, **estado inicial** do latch em nível lógico **alto** ($Q = 1$) e **estado inicial** em nível lógico **baixo** ($Q = 0$).

Exercício 1

- Determinar a tabela verdade do latch mostrado abaixo



A	B	S
0	0	1
0	1	1
1	0	1
1	1	0