

“Uma imagem vale mais do que mil palavras, mas ocupa 3 mil vezes mais memória.”

Mais Latches

Paulo Ricardo Lisboa de Almeida



Alguns trechos desses slides foram baseados nas aulas de Marco Zanata: web.inf.ufpr.br/mazalves/dis-circuitos-digitais

Relembrando

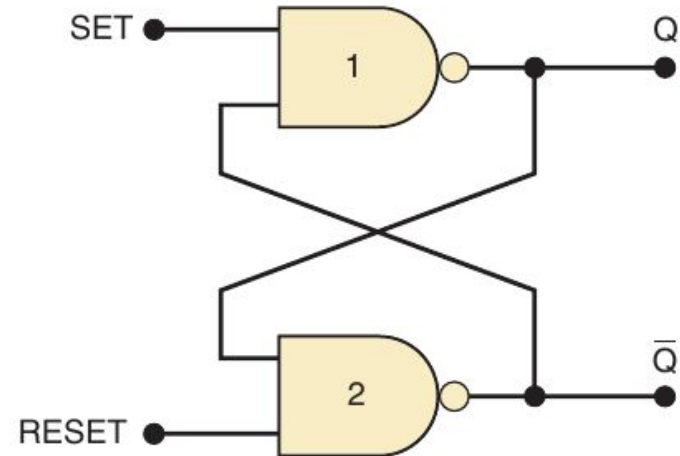
Considere o latch da aula passada.

Como é a entrada para ele se manter estável?

Como fazer um set?

Como fazer um reset?

Qual a entrada proibida?



Relembrando

Considere o latch da aula passada.

Como é a entrada para ele se manter estável?

S:1 e R:1

Como fazer um set?

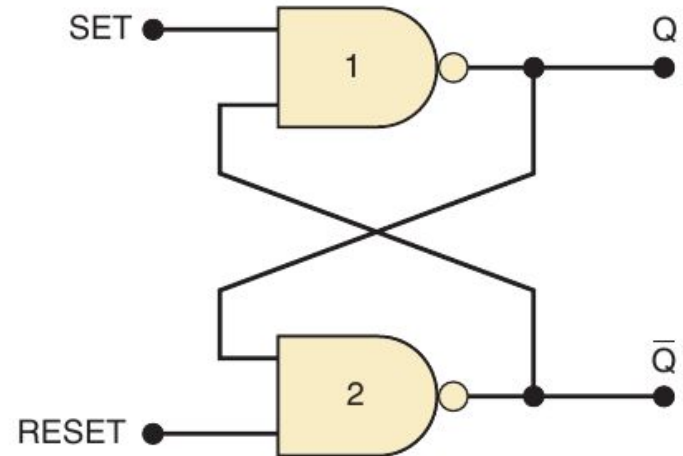
S:0 e R:1

Como fazer um reset?

S:1 e R:0

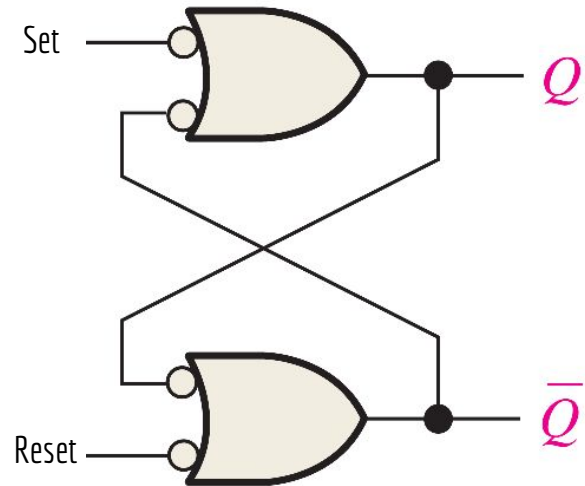
Qual a entrada proibida?

S:0 e R:0



E esse?

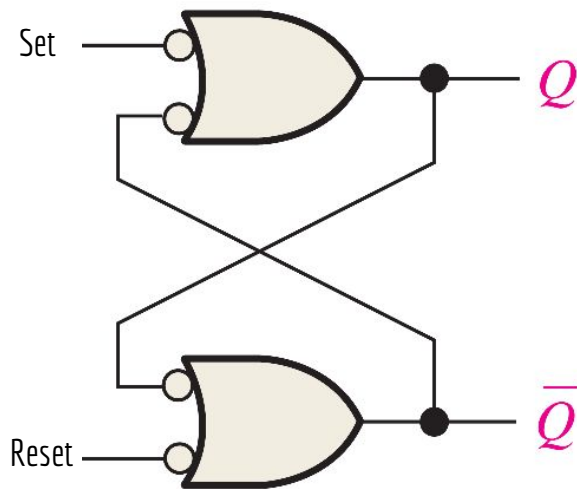
E para esse circuito?



E esse?

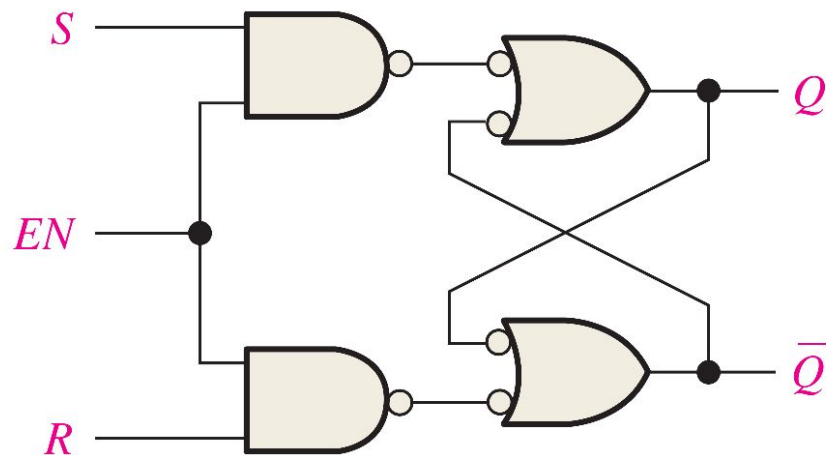
O resultado é o mesmo!

Lembre-se que por *De Morgan*, $\overline{A \cdot B} = \overline{A} + \overline{B}$



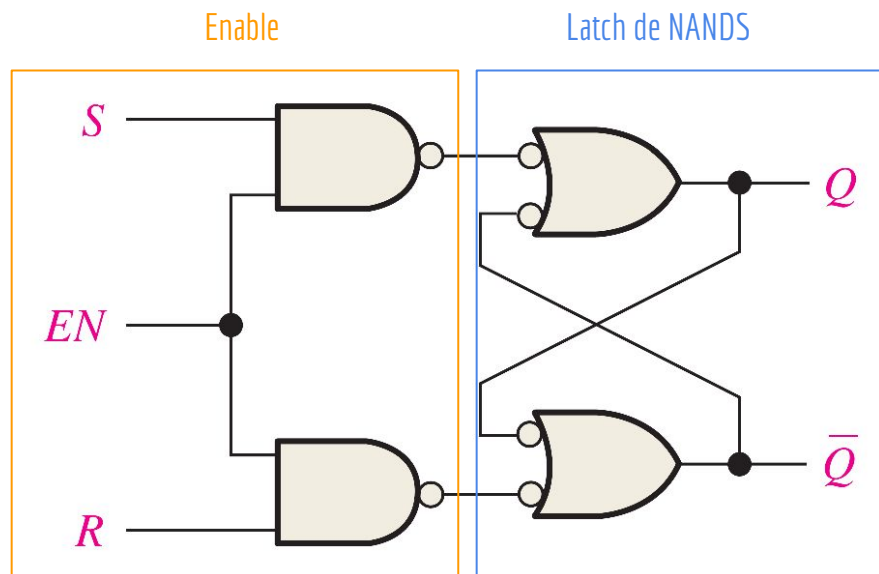
E agora?

5 minutos para você tentar identificar a função dessas portas extras no circuito.



E agora?

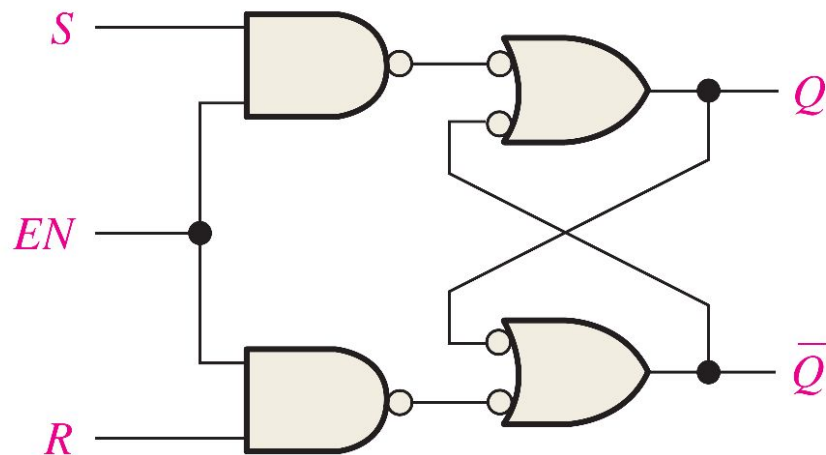
Foi adicionado um circuito de **habilitação** (*enable*).



E agora?

Foi adicionado um circuito de **habilitação** (*enable*).

Se o enable for **0**, não importa S ou R, o Latch vai receber **1 e 1**.



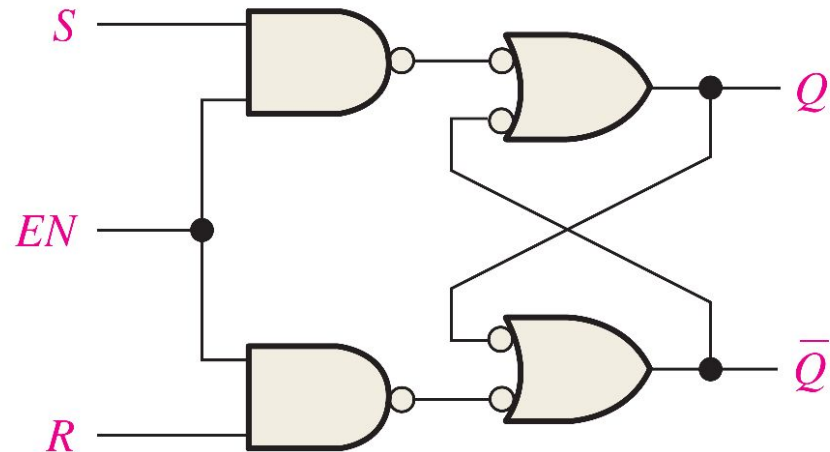
E agora?

Foi adicionado um circuito de **habilitação** (*enable*).

Se o enable for **1**, os pulsos em S e R são aceitos.

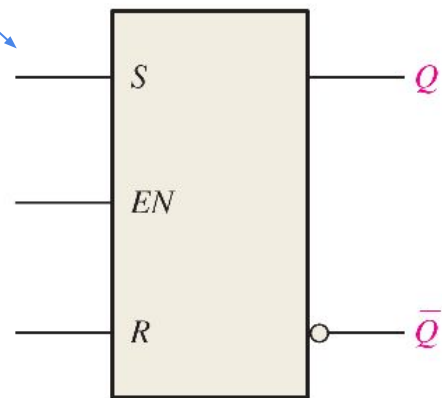
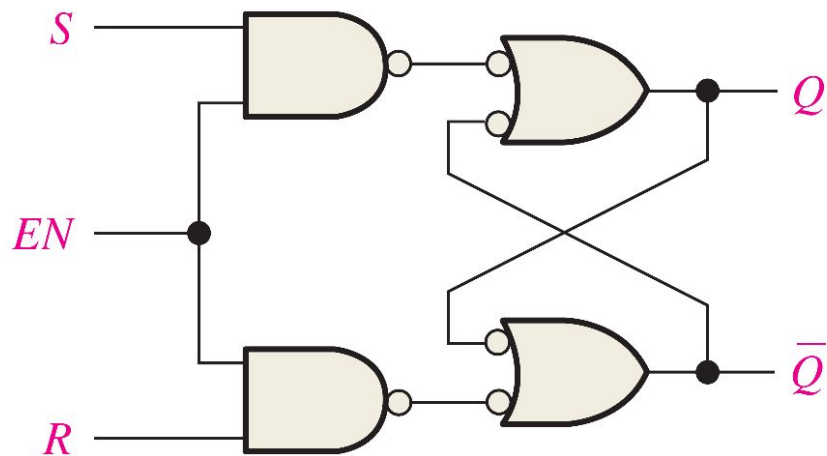
Se S=1, o primeiro NAND vai enviar um pulso de 0 na entrada Set do Latch.

Uma análise similar é feita quando é enviado R=1.



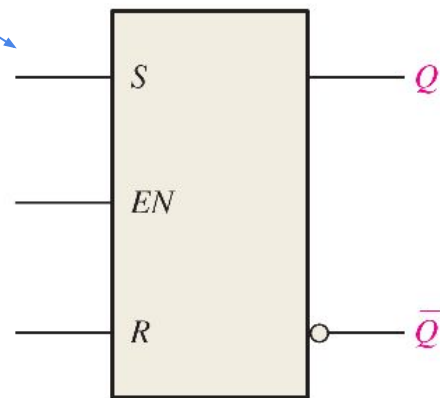
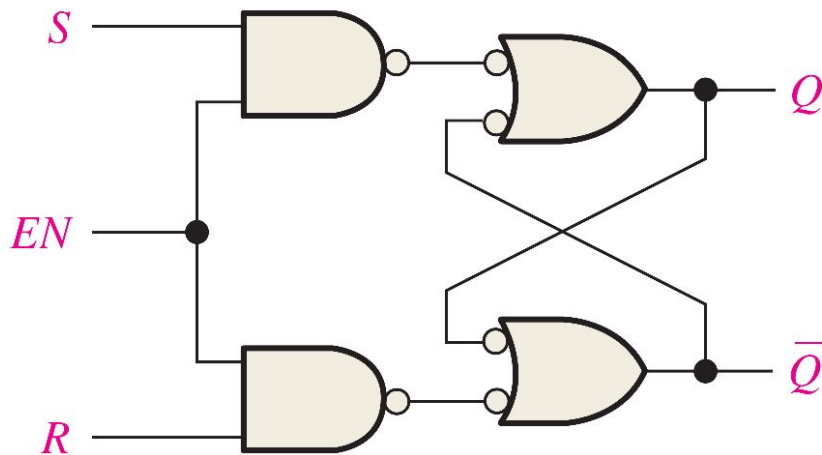
Símbolo Lógico

Note que as entradas não são negadas.
Para fazer um set ou reset, precisamos ...?

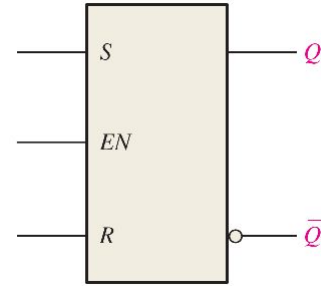
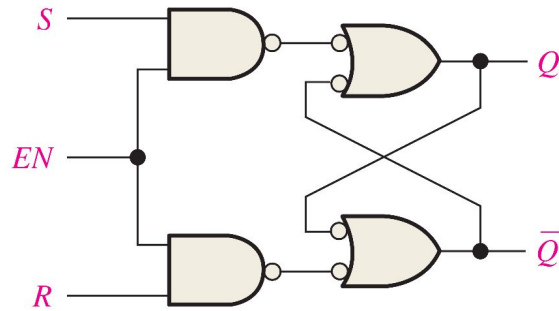


Símbolo Lógico

Note que as entradas não são negadas.
Para fazer um set ou reset, precisamos dar um pulso de 1 em S ou R.



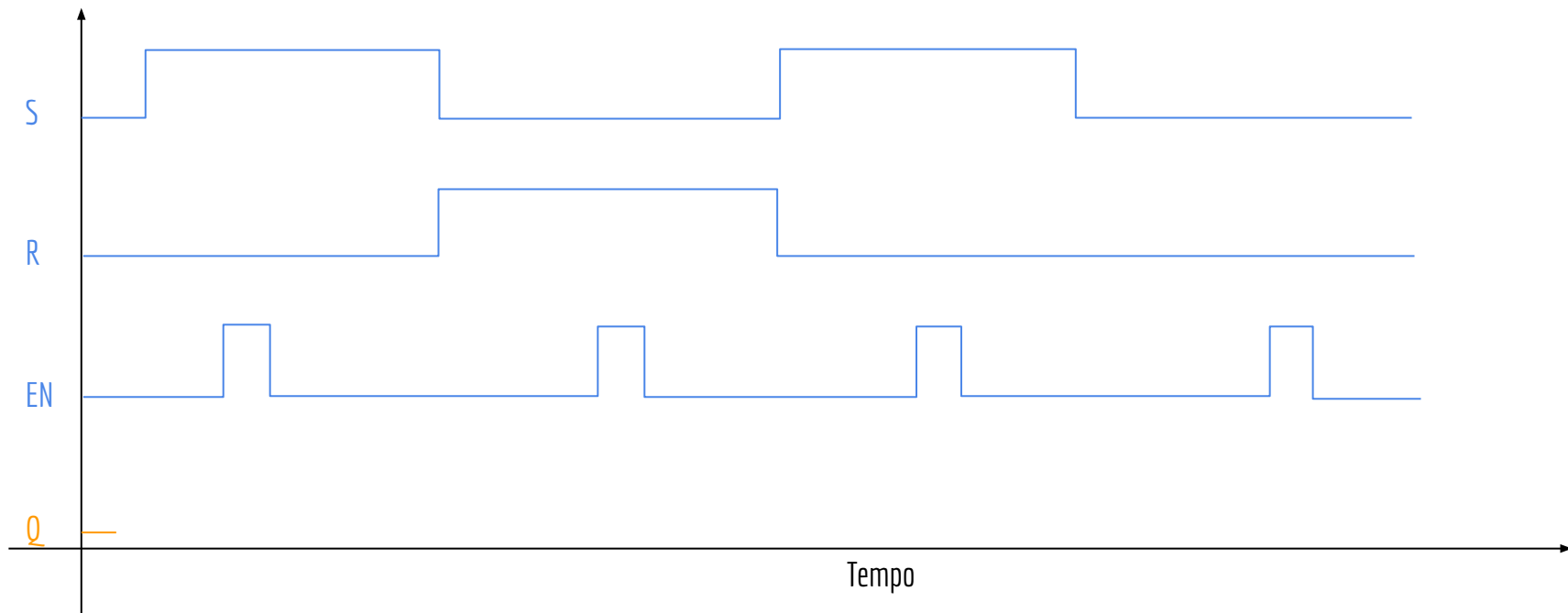
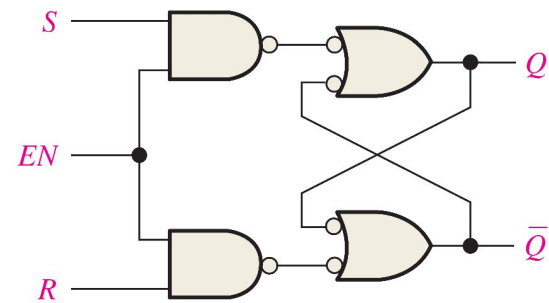
Então...



EN	S	R	Q
1	0	1	0 (reset)
1	1	0	1 (set)
1	0	0	Mantém Q
0	X	X	Mantém Q

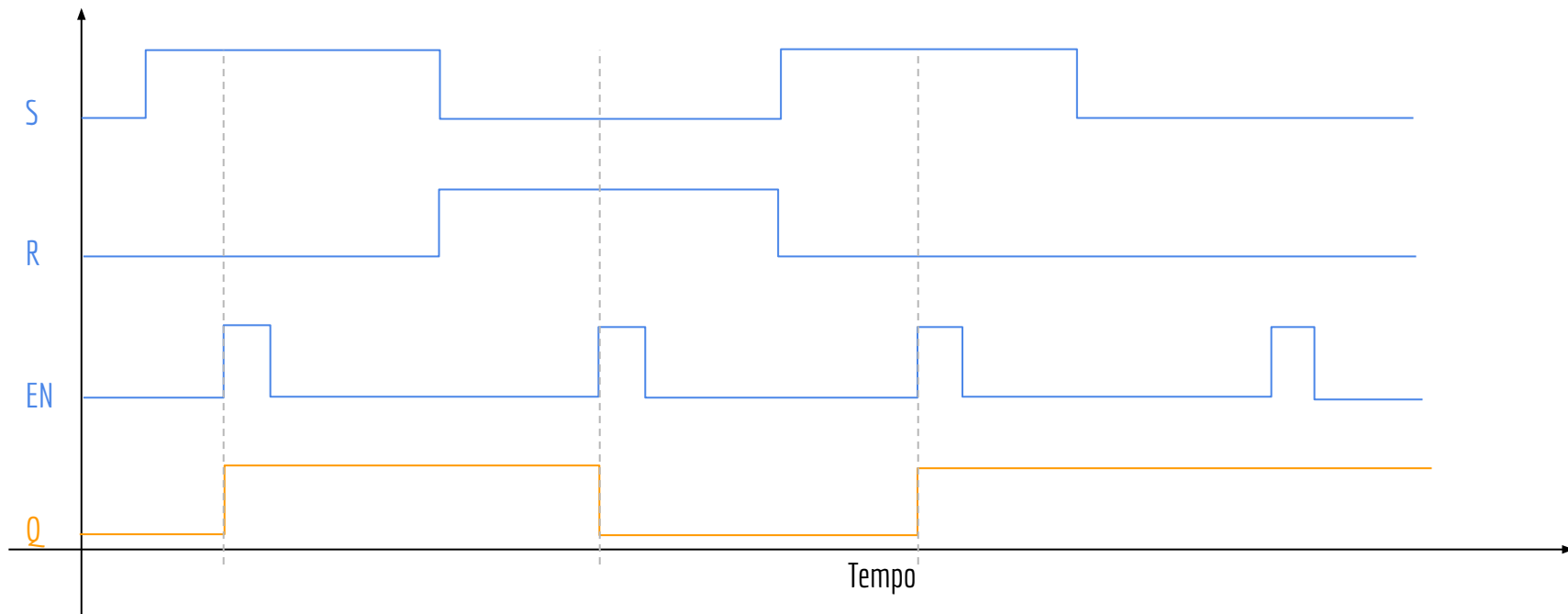
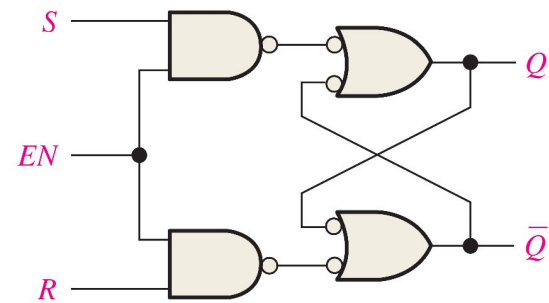
Faça você mesmo

Faça o diagrama de temporização.



Faça você mesmo

Faça o diagrama de temporização.



Latches do Tipo D

Ainda não evitamos o problema do $S=R=1$ quando o Enable está ativo.

Latches do Tipo D

Ainda não evitamos o problema do $S=R=1$ quando o Enable está ativo.

Vamos manter uma entrada de dados (**data**) D única.

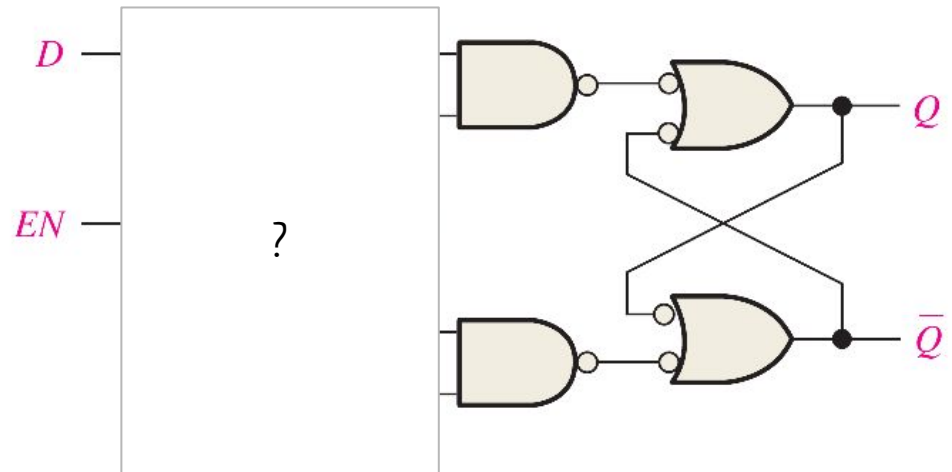
Criamos um latch do **Tipo D** (Tipo Data).

Faça você mesmo

O Latch do tipo D possui uma entrada D, e o Enable.

Quando Enable = 1, o valor de D é armazenado no Latch.

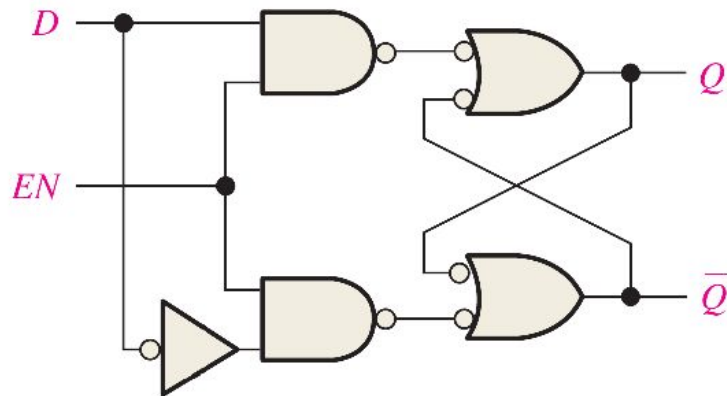
Tente modificar o circuito você mesmo para que isso ocorra.



Latches do Tipo D

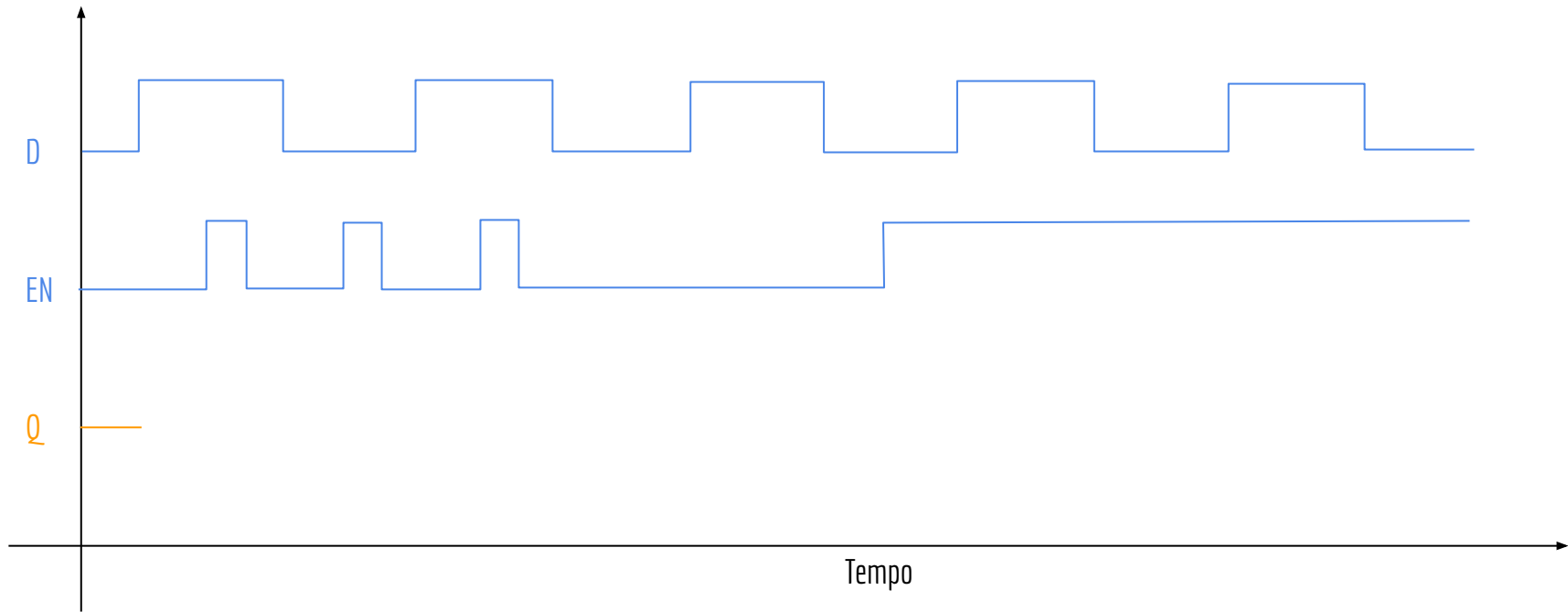
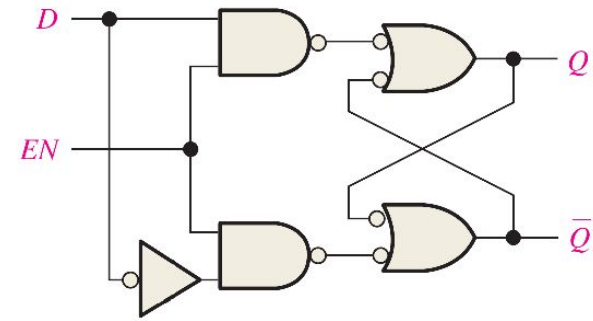
O Latch do tipo D possui uma entrada D, e o Enable.

Quando Enable = 1, o valor de D é armazenado no Latch.



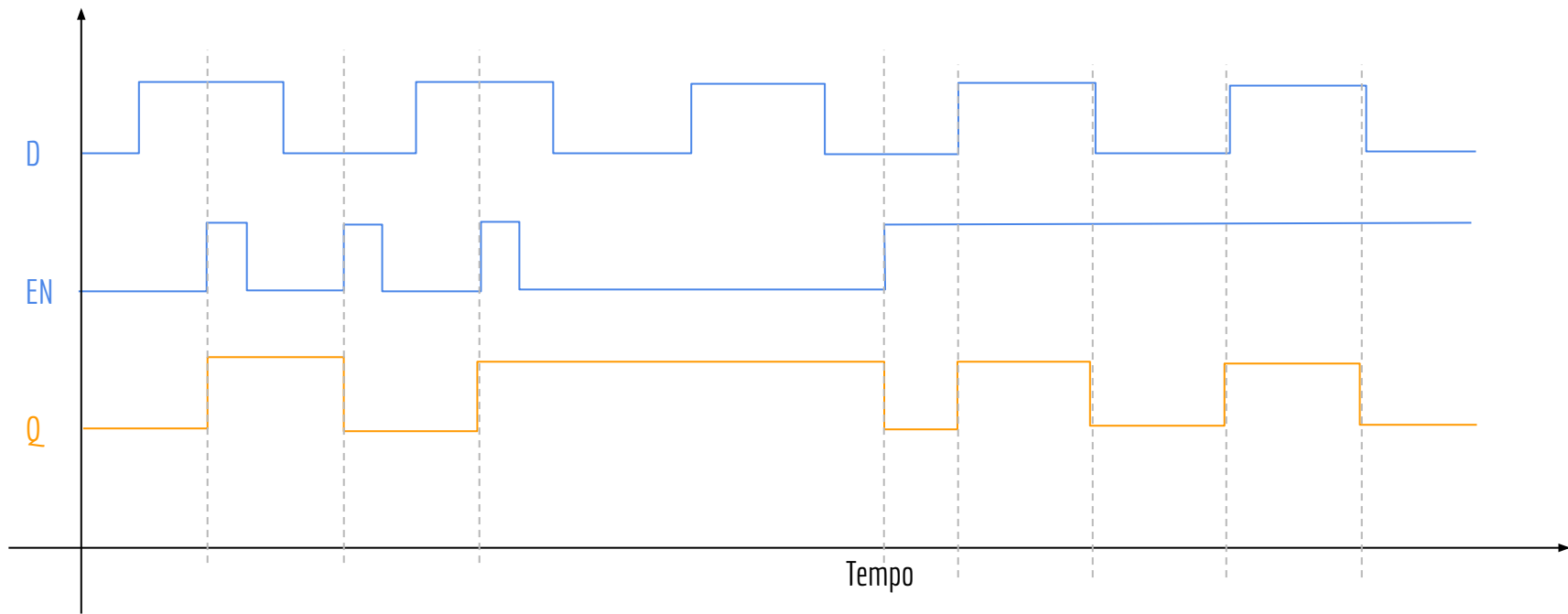
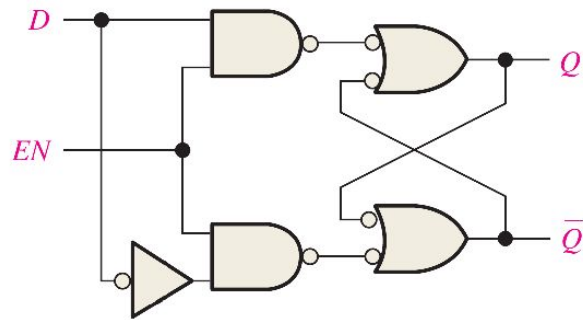
Faça você mesmo

Faça o diagrama de temporização.



Faça você mesmo

Faça o diagrama de temporização



Exemplo de Uso

Registrador de armazenamento: armazena uma palavra de dado.

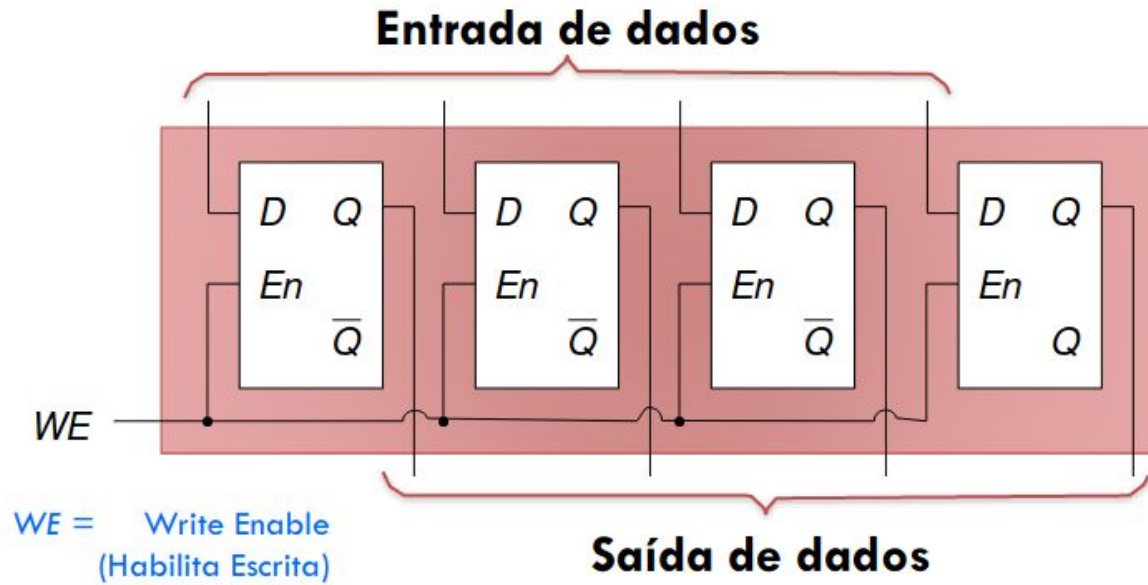
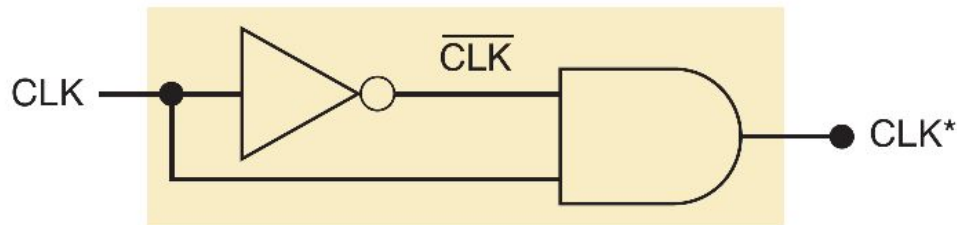


Figura de Marco Zanata.

Sincronizando

O que o circuito a seguir faz?

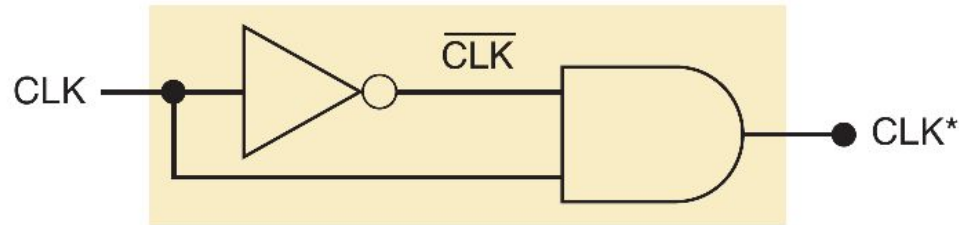


Sincronizando

O que o circuito a seguir faz?

Parece que a saída sempre é zero.

Mas no mundo real, **temos atrasos**.

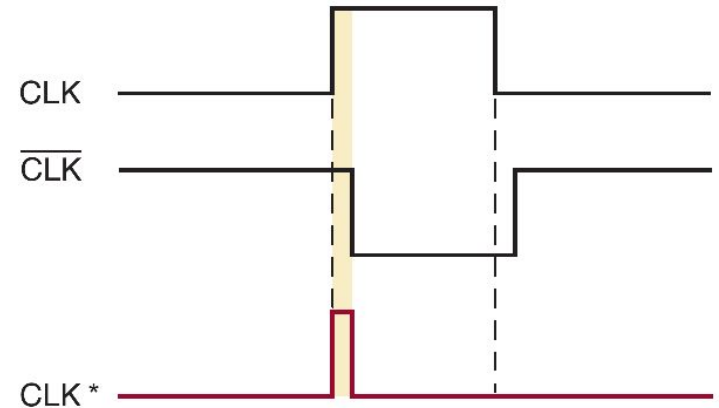
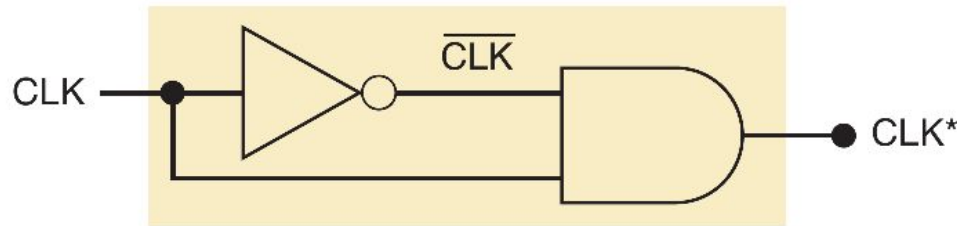


Sincronizando

O que o circuito a seguir faz?

Parece que a saída sempre é zero.

Mas no mundo real, **temos atrasos**.

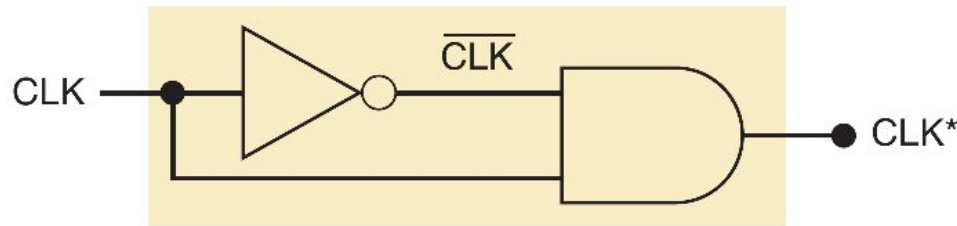


Sincronizando

O que o circuito a seguir faz?

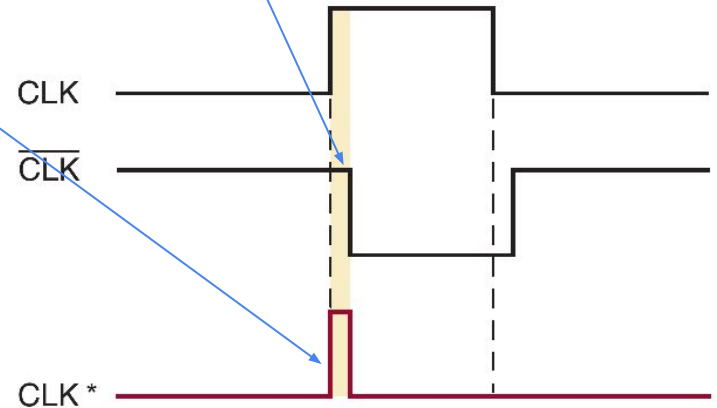
Parece que a saída sempre é zero.

Mas no mundo real, **temos atrasos**.



A saída é um pico de curta duração na borda de subida.

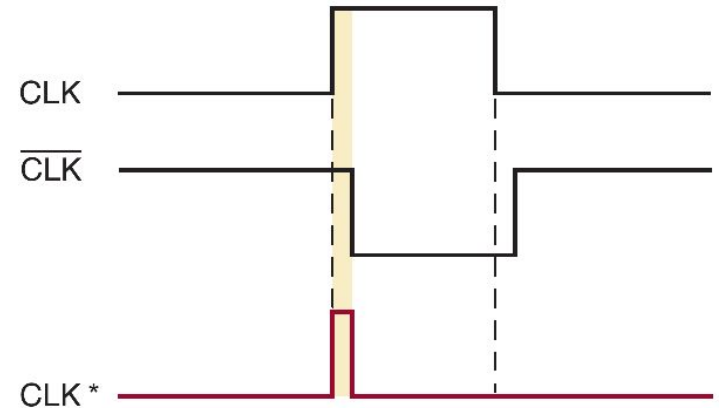
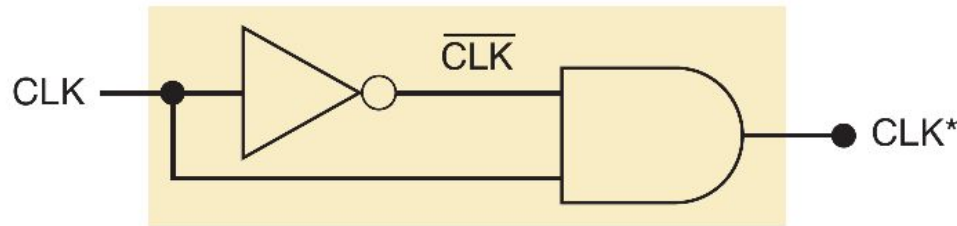
A porta NOT demora um tempo para inverter o sinal de entrada



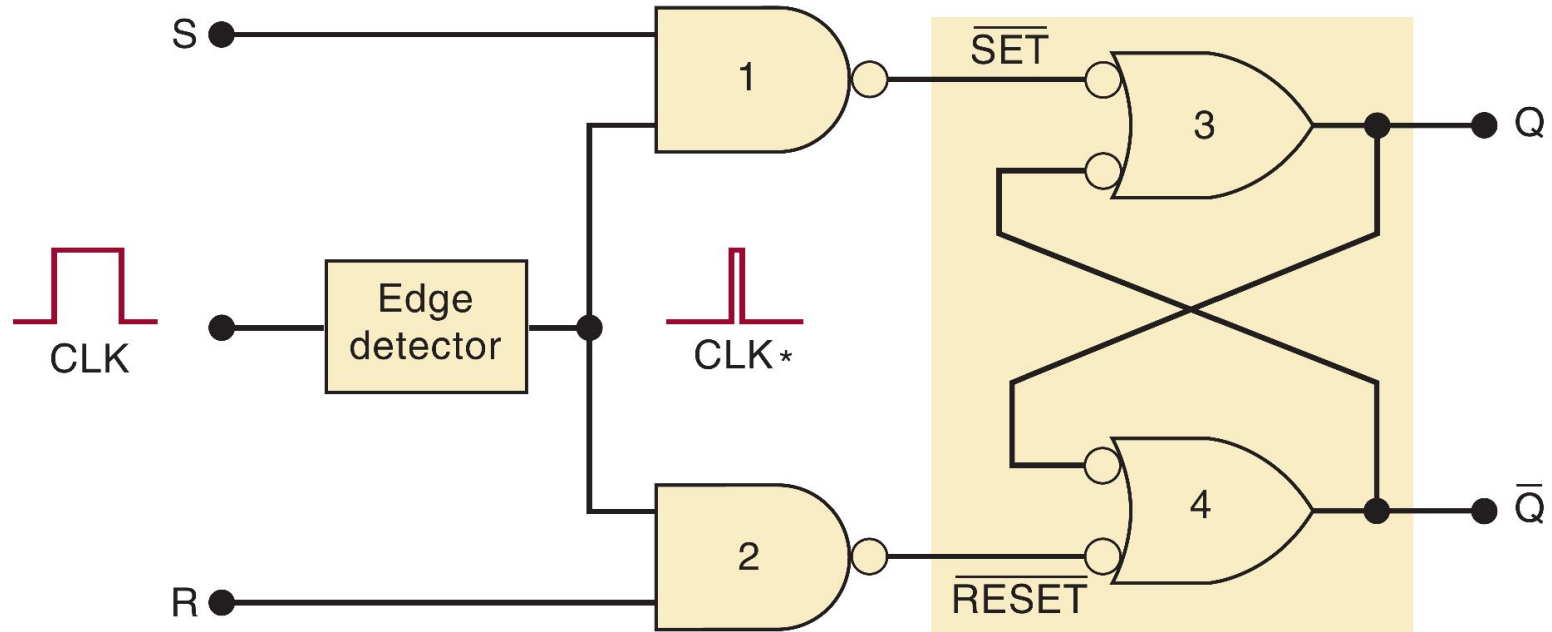
Sincronizando

Devido aos atrasos, o circuito gera um pico de curta duração na borda de subida.

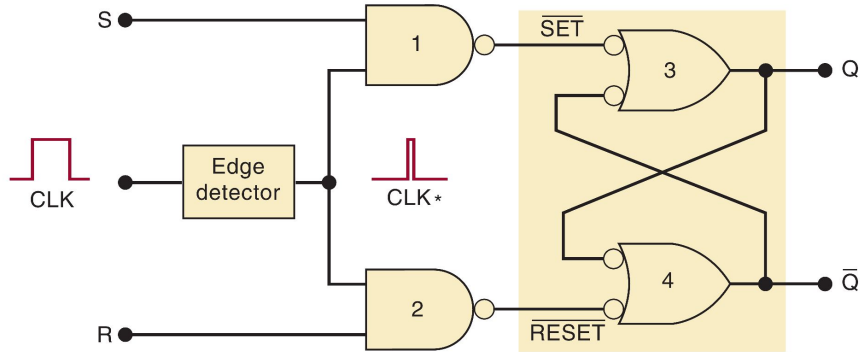
Esse é um **detector de borda de subida**, ou detector de borda positiva.



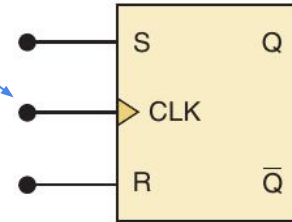
Sincronizando



Sincronizando

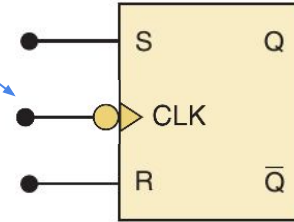


Indica que é sincronizado pela
borda de subida do clock.



Sincronizando - Borda de descida

Uma negação indica que é sincronizado pela borda de descida.



Flip-Flop

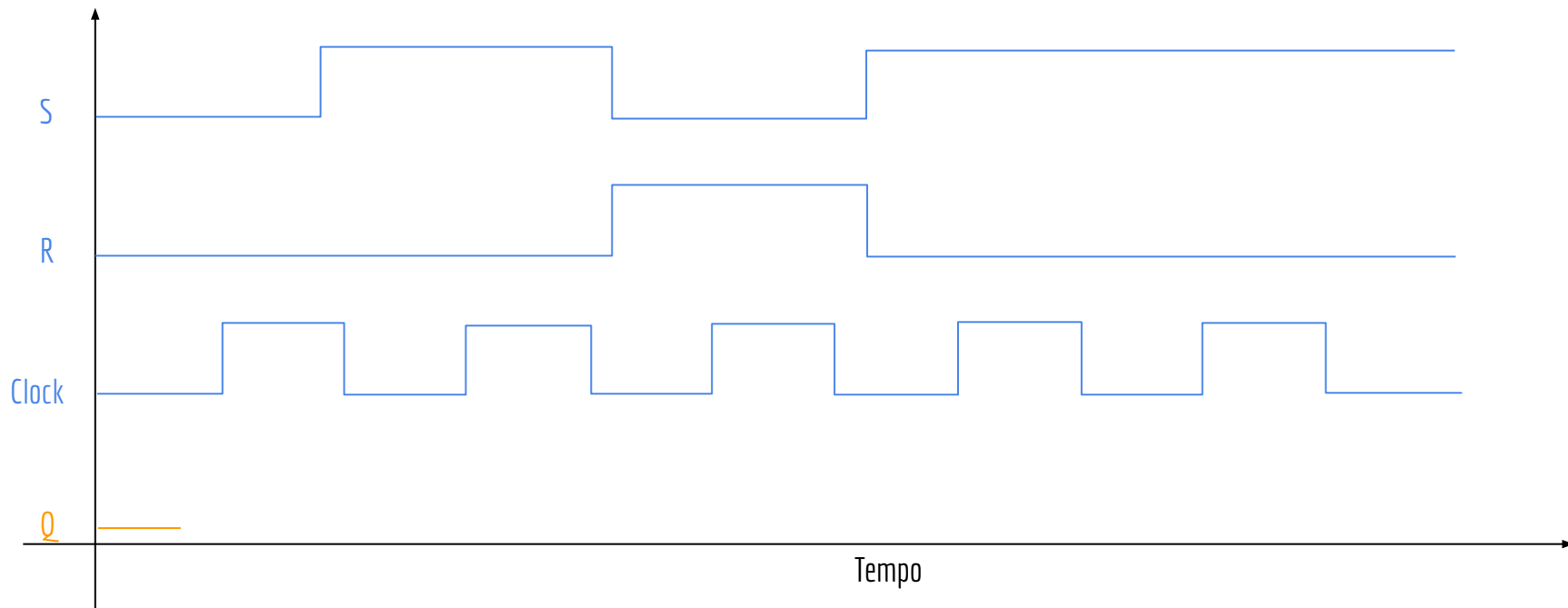
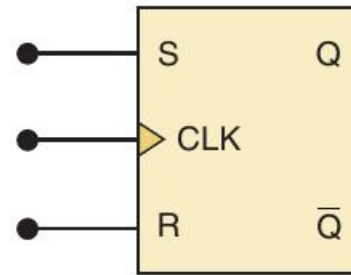
Ao plugar um detector de bordas na porta Enable, podemos sincronizar a escrita dos Latches via um sinal de clock.

Temos um **Flip-Flop**.

Latch Sincronizado.

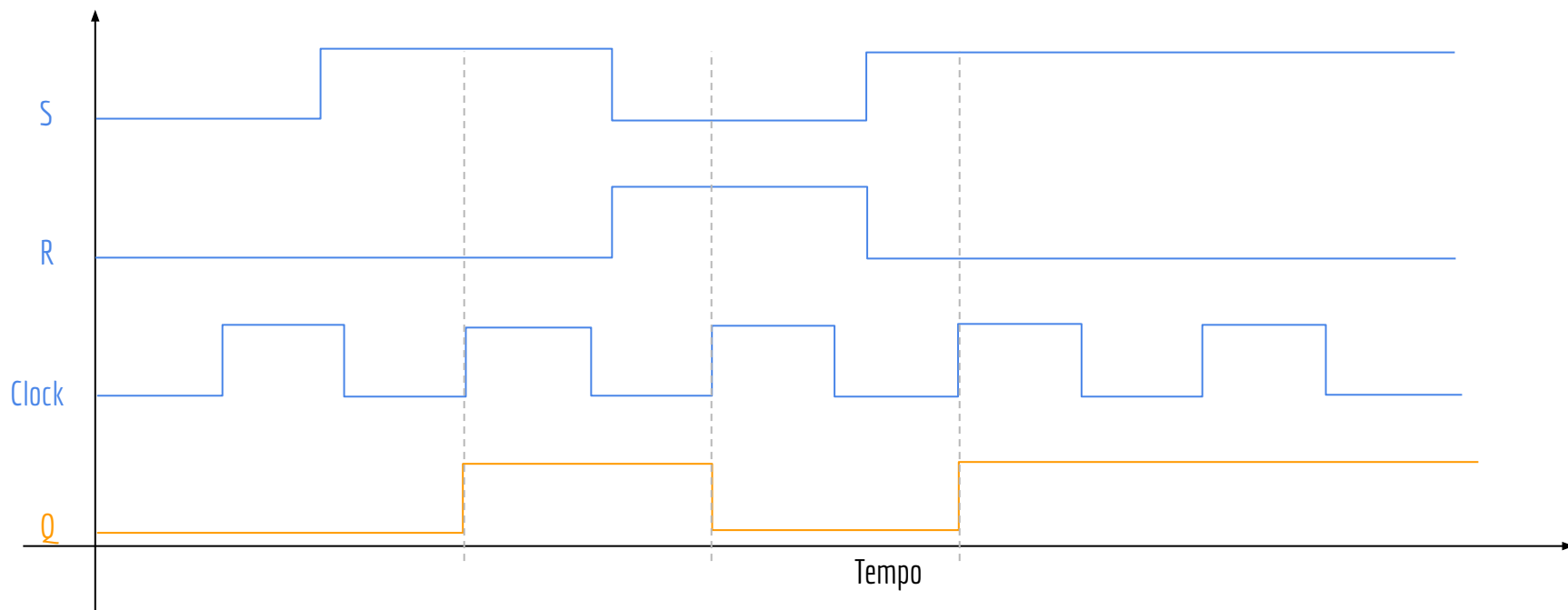
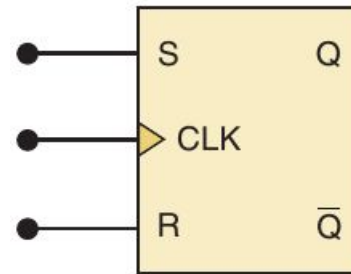
Faça você mesmo

Faça o diagrama de temporização



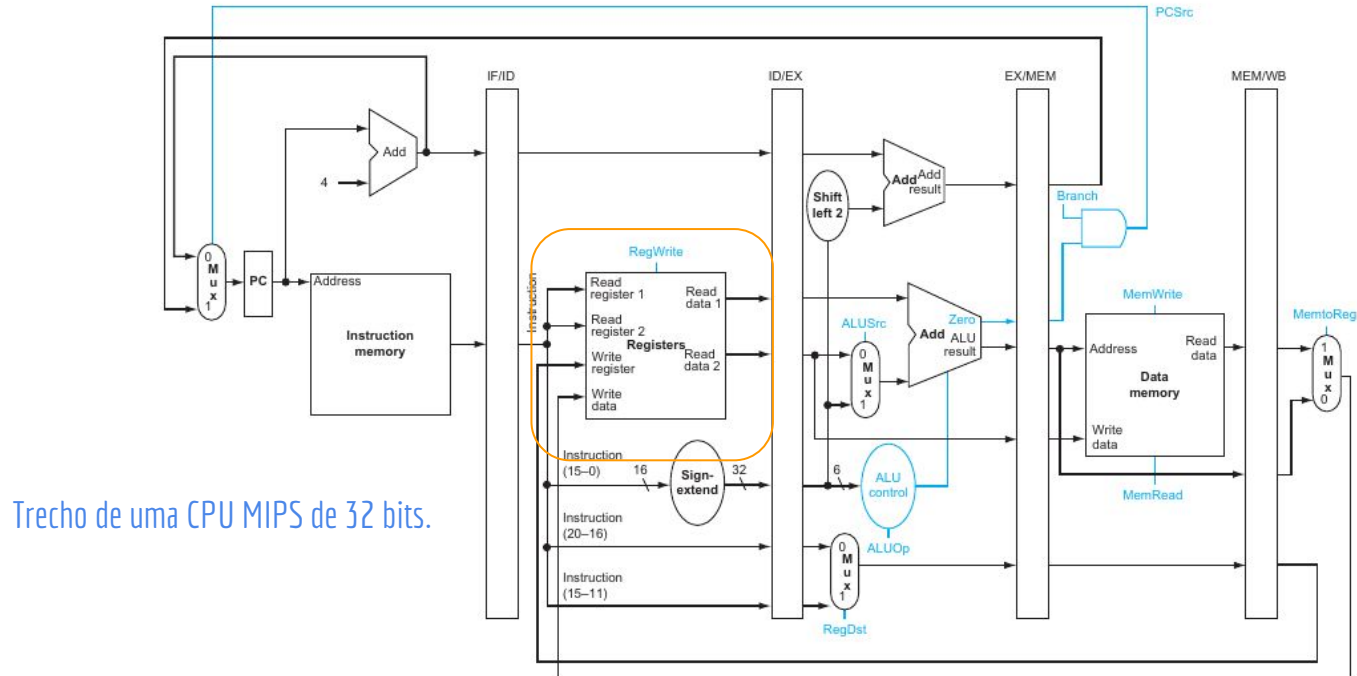
Faça você mesmo

Faça o diagrama de temporização



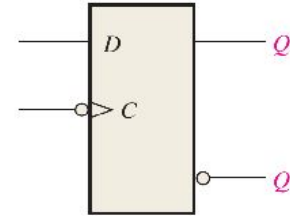
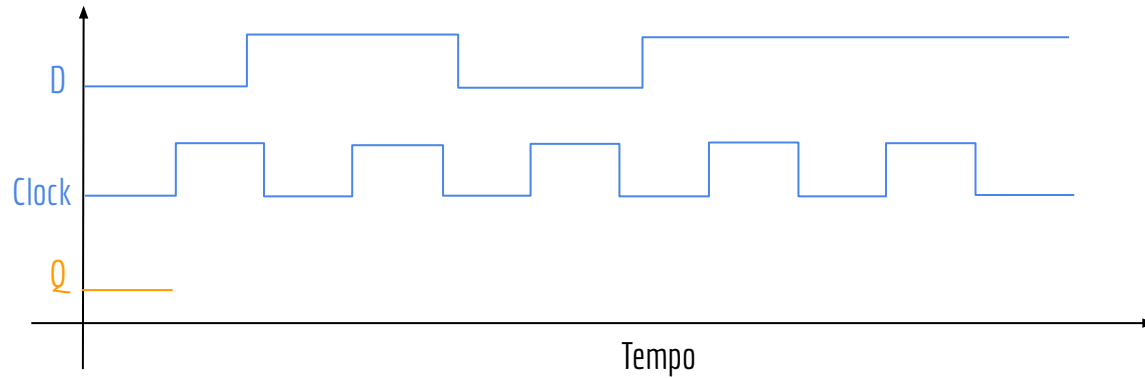
Exemplo de uso de Flip-Flops

Bancos de registradores (memórias “rápidas”) da CPU podem ser construídas com flip-flops.



Exercícios

1. Considere sincronização por borda de subida, e faça o diagrama de temporização para o Flip-Flop Tipo D a seguir:



2. Faça o mesmo que no exercício 1, mas agora considere sincronização por borda de descida.
3. Crie um circuito de detecção de borda de descida.

Exercícios

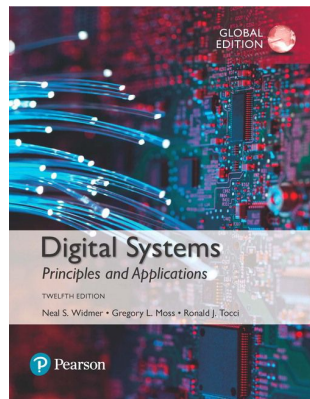
4. Adicione um circuito de Enable em um Latch de NORs.

- a. Como é o comportamento deste circuito?
- b. Faça os diagramas de temporização feitos em aula para esse novo circuito. Como é o comportamento nos diagramas?

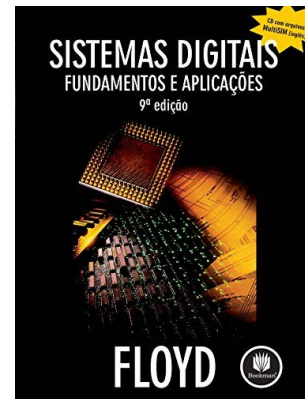
5. Veja novamente este vídeo com um teste empírico com atrasos em circuitos para entender melhor o detector de bordas: <https://youtu.be/ndz5BYBnt6U>

Referências

Ronald J. Tocci, Gregory L. Moss, Neal S. Widmer. Sistemas digitais. 10a ed. 2017.



Thomas Floyd. Sistemas Digitais: Fundamentos e Aplicações. 2009.



Licença

Esta obra está licenciada com uma Licença [Creative Commons Atribuição 4.0 Internacional](https://creativecommons.org/licenses/by/4.0/).