

**Centro Federal de Educação Tecnológica de Minas Gerais**  
**ENGENHARIA DA COMPUTAÇÃO**

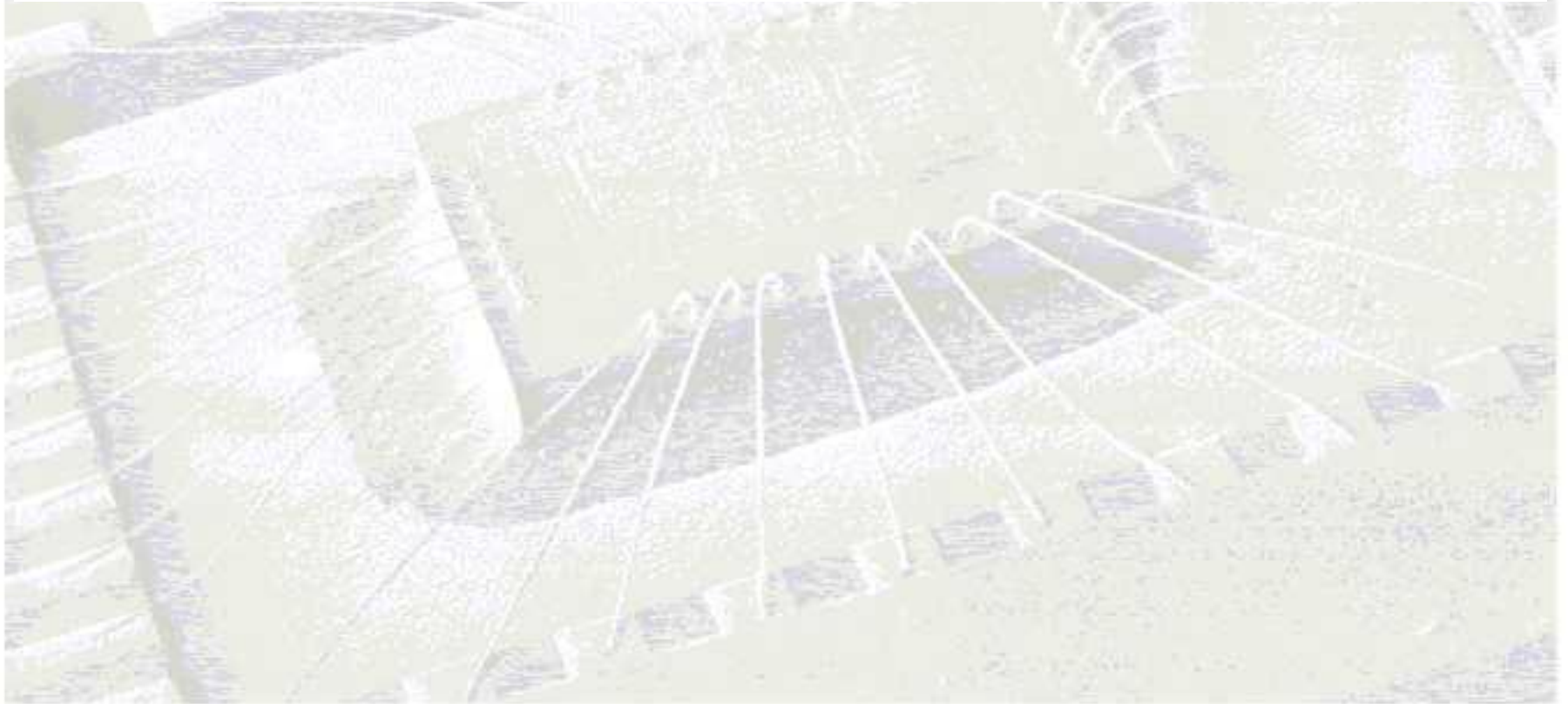
---

# **Aula 08**

## **Flip-Flop**

# Flip-Flop

Os circuitos lógicos estudados até o momento dependiam, apenas, dos níveis lógicos de entrada a cada instante de tempo, para gerar suas saídas – todos são denominados **circuitos lógicos combinacionais**.



# Flip-Flop

Os circuitos lógicos estudados até o momento dependiam, apenas, dos níveis lógicos de entrada a cada instante de tempo, para gerar suas saídas – todos são denominados **circuitos lógicos combinacionais**.

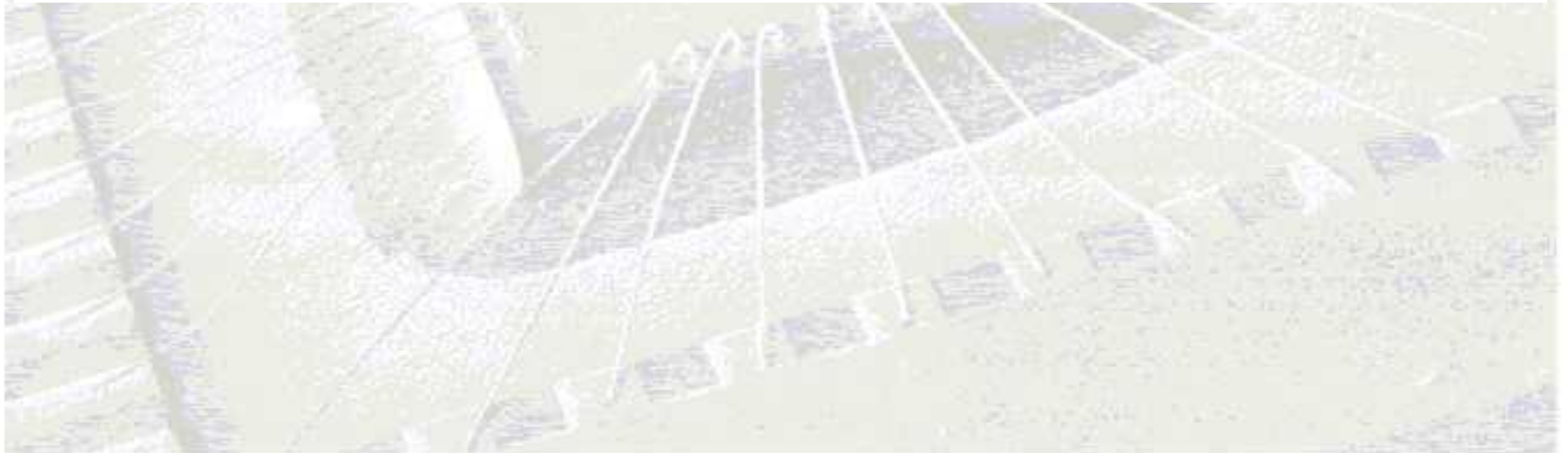
Quaisquer condições de entradas **anteriores** não têm efeito sobre as saídas **atuais**, porque um circuito lógico combinacional não possui memória.

# Flip-Flop

Os circuitos lógicos estudados até o momento dependiam, apenas, dos níveis lógicos de entrada, para a cada instante de tempo, para gerar suas saídas – todos eram **circuitos lógicos combinacionais**.

Quaisquer condições de entradas **anteriores** não têm efeito sobre as saídas **atuais**, porque um circuito lógico combinacional não possui memória.

Entretanto, a maioria dos **sistemas digitais** são constituídos de circuitos combinacionais e de elementos de memória, conforme o digrama abaixo.



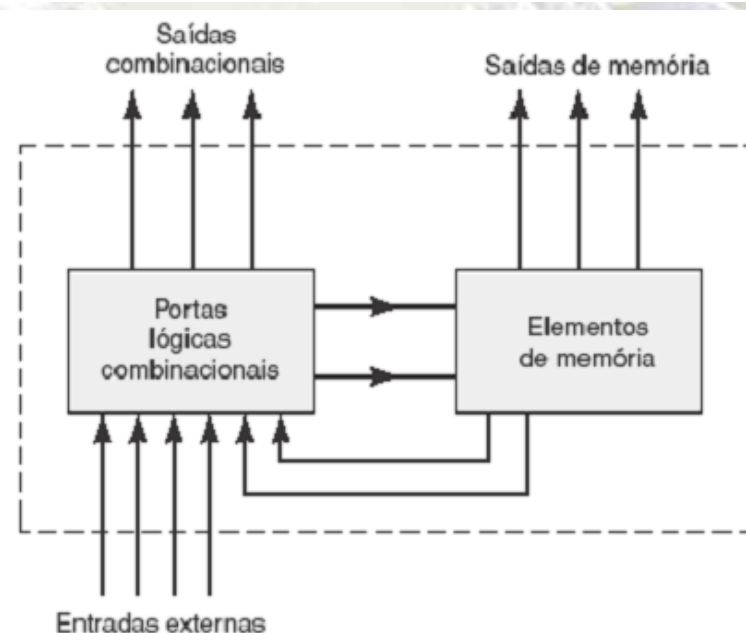


# Flip-Flop

Os circuitos lógicos estudados até o momento dependiam, apenas, dos níveis lógicos de entrada, para a cada instante de tempo, para gerar suas saídas – todos eram **circuitos lógicos combinacionais**.

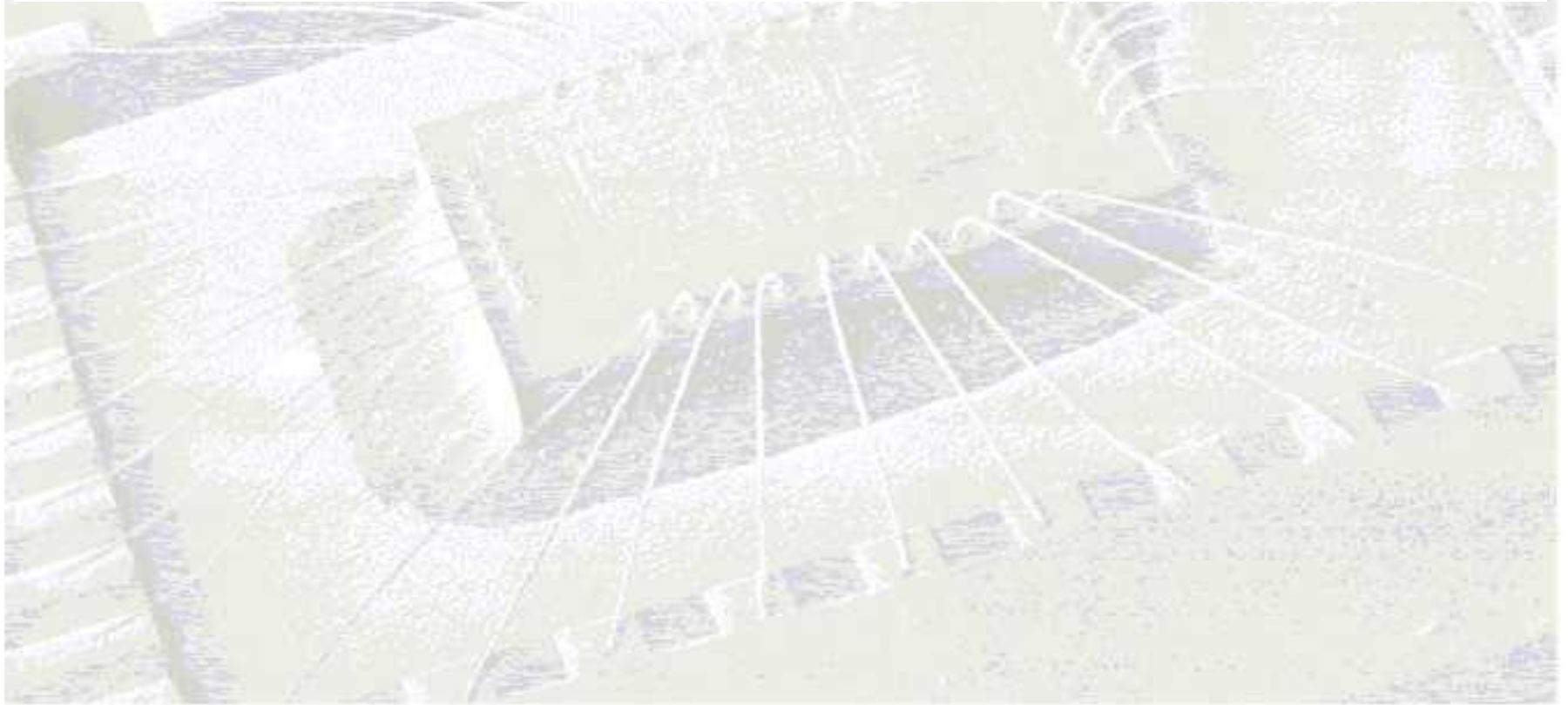
Quaisquer condições de entradas **anteriores** não têm efeito sobre as saídas **atuais**, porque um circuito lógico combinacional não possui memória.

Entretanto, a maioria dos **sistemas digitais** são constituídos de circuitos combinacionais e de elementos de memória, conforme o digrama abaixo.



# Flip-Flop

Embora uma única porta lógica, por si só, não tenha capacidade de armazenamento, algumas delas podem ser conectadas entre si, criando o circuito Flip-Flop, de tal forma que permita o armazenamento de informação.



# Flip-Flop

Embora uma única porta lógica, por si só, não tenha capacidade de armazenamento, algumas delas podem ser conectadas entre si, criando o circuito Flip-Flop, de tal forma que permita o armazenamento de informação.

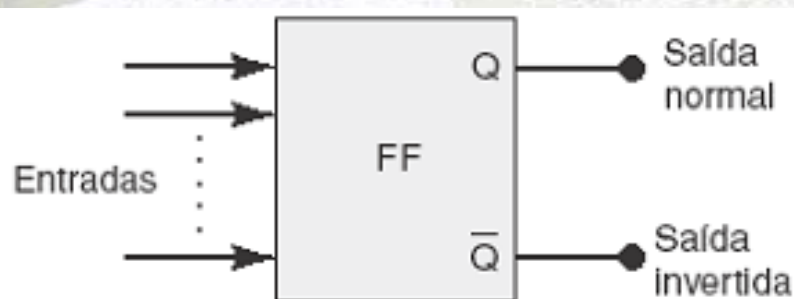
Abaixo é mostrado um tipo de símbolo genérico para representar um **Flip-Flop**. Esse símbolo apresenta as entradas e as duas saídas oposta entre si.



# Flip-Flop

Embora uma única porta lógica, por si só, não tenha capacidade de armazenamento, algumas delas podem ser conectadas entre si, criando o circuito Flip-Flop, de tal forma que permita o armazenamento de informação.

Abaixo é mostrado um tipo de símbolo genérico para representar um **Flip-Flop**. Esse símbolo apresenta as entradas e as duas saídas oposta entre si.



Importante: Q é a saída principal.

## Estados de saída

$$\underline{Q = 1, \bar{Q} = 0:}$$

chamado estado ALTO ou 1;  
também chamado estado SET

$$\underline{Q = 0, \bar{Q} = 1:}$$

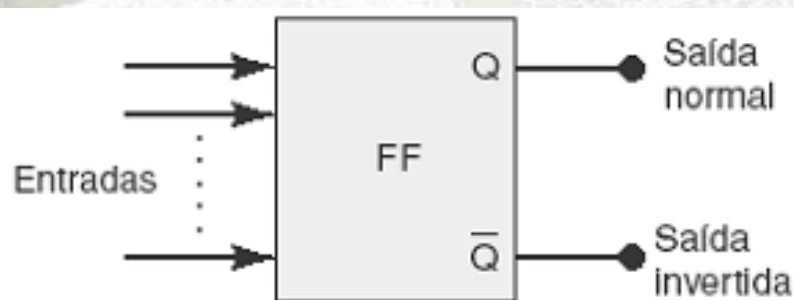
chamado estado BAIXO ou 0;  
também chamado estado  
CLEAR ou RESET



# Flip-Flop

Embora uma única porta lógica, por si só, não tenha capacidade de armazenamento, algumas delas podem ser conectadas entre si, criando o circuito Flip-Flop, de tal forma que permita o armazenamento de informação.

Abaixo é mostrado um tipo de símbolo genérico para representar um **Flip-Flop**. Esse símbolo apresenta as entradas e as duas saídas oposta entre si.



Importante: Q é a saída principal.

## Estados de saída

$$\underline{Q = 1, \bar{Q} = 0:}$$

chamado estado ALTO ou 1;  
também chamado estado SET

$$\underline{Q = 0, \bar{Q} = 1:}$$

chamado estado BAIXO ou 0;  
também chamado estado  
CLEAR ou RESET

As entradas dos Flip-Flops precisam ser apenas momentaneamente ativadas para provocar mudanças na saída, sendo que a saída permanece no novo estado mesmo após o pulso de entrada ter terminado (MEMÓRIA).

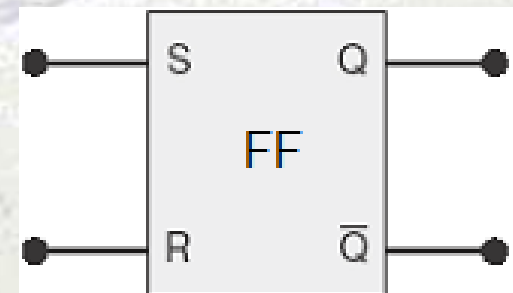
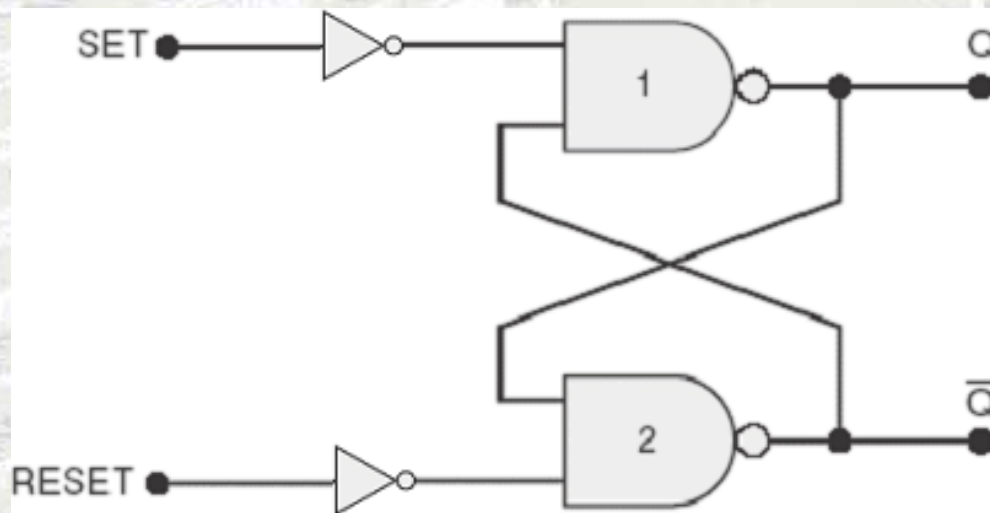
# Flip-Flop

O **Flip-Flop (FF)** é conhecido por outros nomes, incluindo ***Latch*** e ***Multivibrador Biestável***. O termo ***Latch*** é usado para certos tipos de **FFs**. O termo ***Multivibrador Biestável*** é uma denominação mais técnica para um **FF**. A principal diferença entre um FF e um Latch é a forma de disparo.

# Flip-Flop

O Flip-Flop (FF) é conhecido por outros nomes, incluindo **Latch** e **Multivibrador Biestável**. O termo **Latch** é usado para certos tipos de FFs. O termo **Multivibrador Biestável** é uma denominação mais técnica para um FF. A principal diferença entre um FF e um Latch é a forma de disparo.

Primeiramente, vamos analisar o Flip-Flop RS básico, construído a partir de portas NAND e inversores, cujo circuito é:



# Funcionamento do Flip-Flop

Para analisarmos o comportamento do FF RS básico, deve-se preencher a tabela verdade, levando em consideração as 2 variáveis de entradas (R e S) e o estado da saída anterior (Qa):

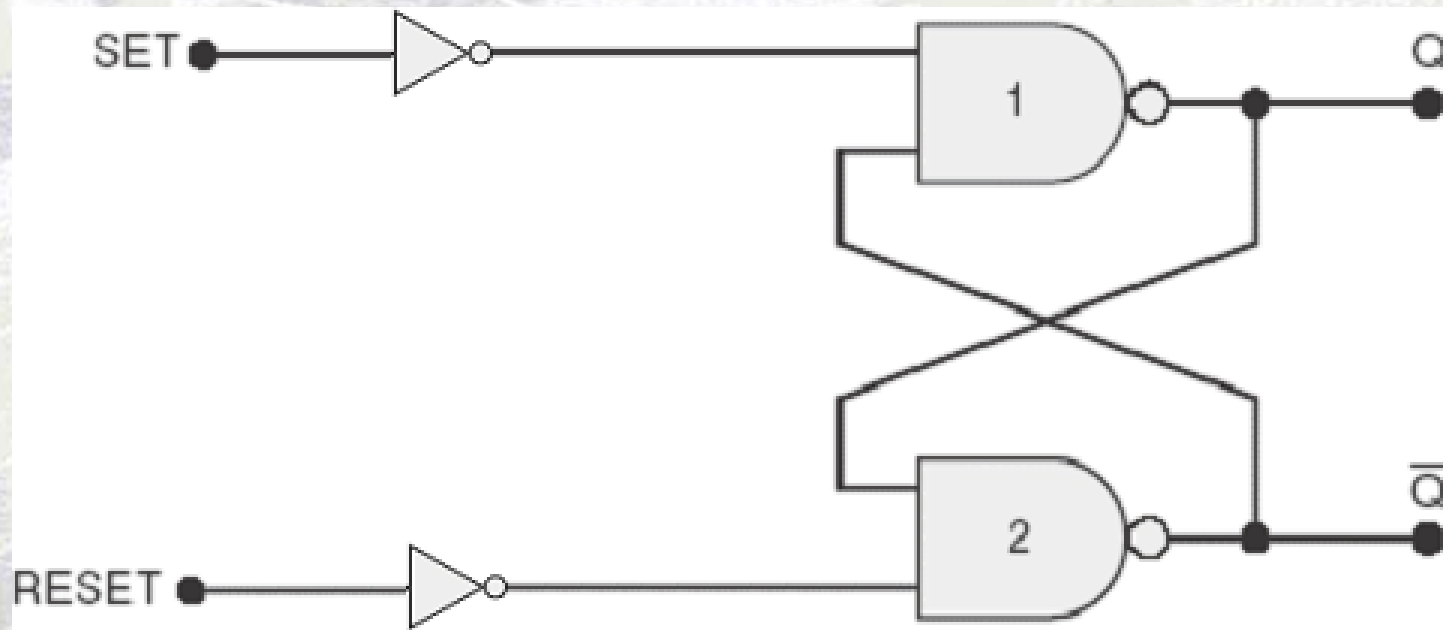
Casos	Set (S)	Reset (R)	Qa	Qf
1º	0	0	0	
2º	0	0	1	
3º	0	1	0	
4º	0	1	1	
5º	1	0	0	
6º	1	0	1	
7º	1	1	0	
8º	1	1	1	



# Funcionamento do Flip-Flop

1º Caso)

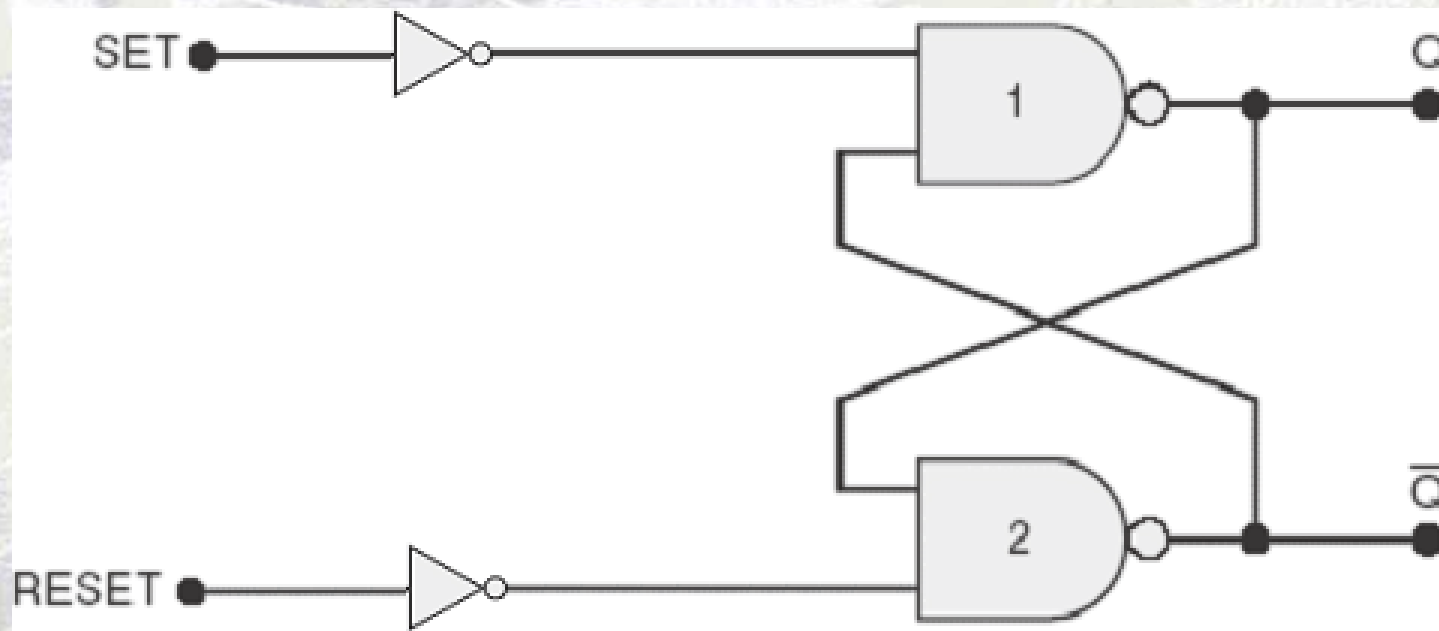
$$S = 0, \quad R = 0, \quad Q_a = 0, \quad \overline{Q}_a = 1$$



# Funcionamento do Flip-Flop

1º Caso)

$$S = 0, \quad R = 0, \quad Q_a = 0, \quad \overline{Q}_a = 1$$

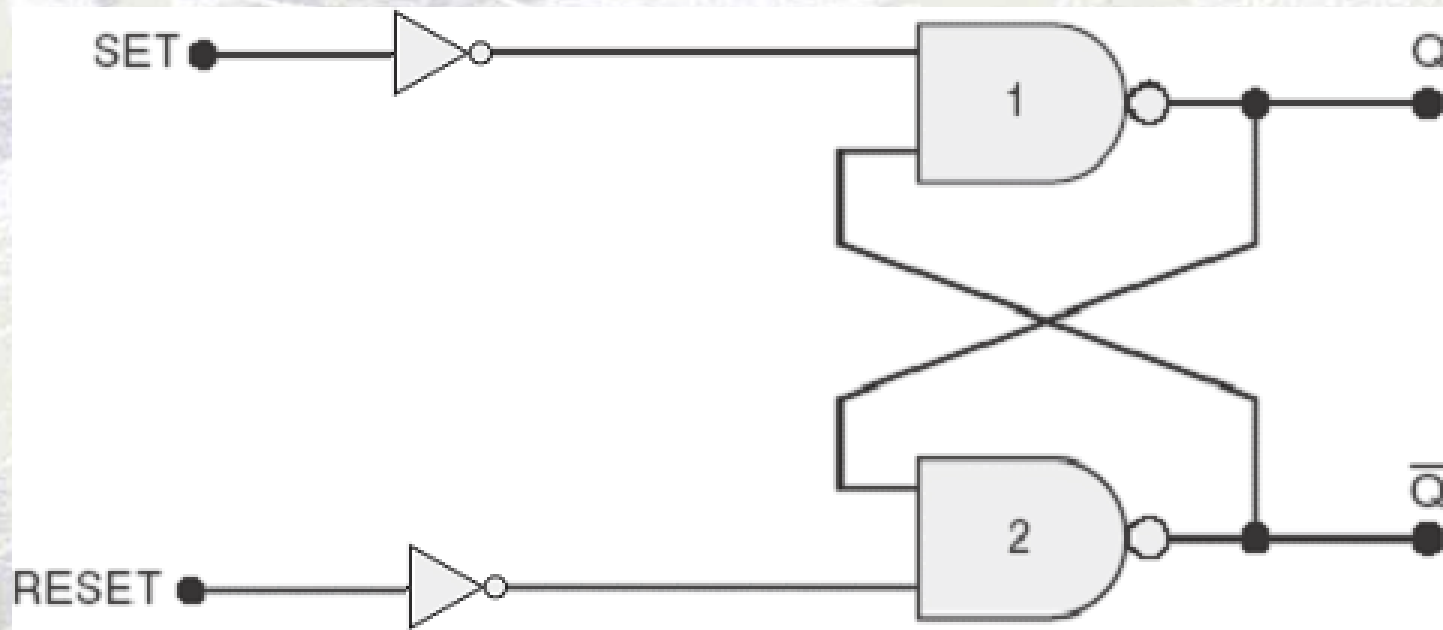


Podemos notar que este estado é estável, logo, o valor que a saída Q irá assumir será igual ao seu valor anterior à aplicação das entradas RS.

# Funcionamento do Flip-Flop

2º Caso)

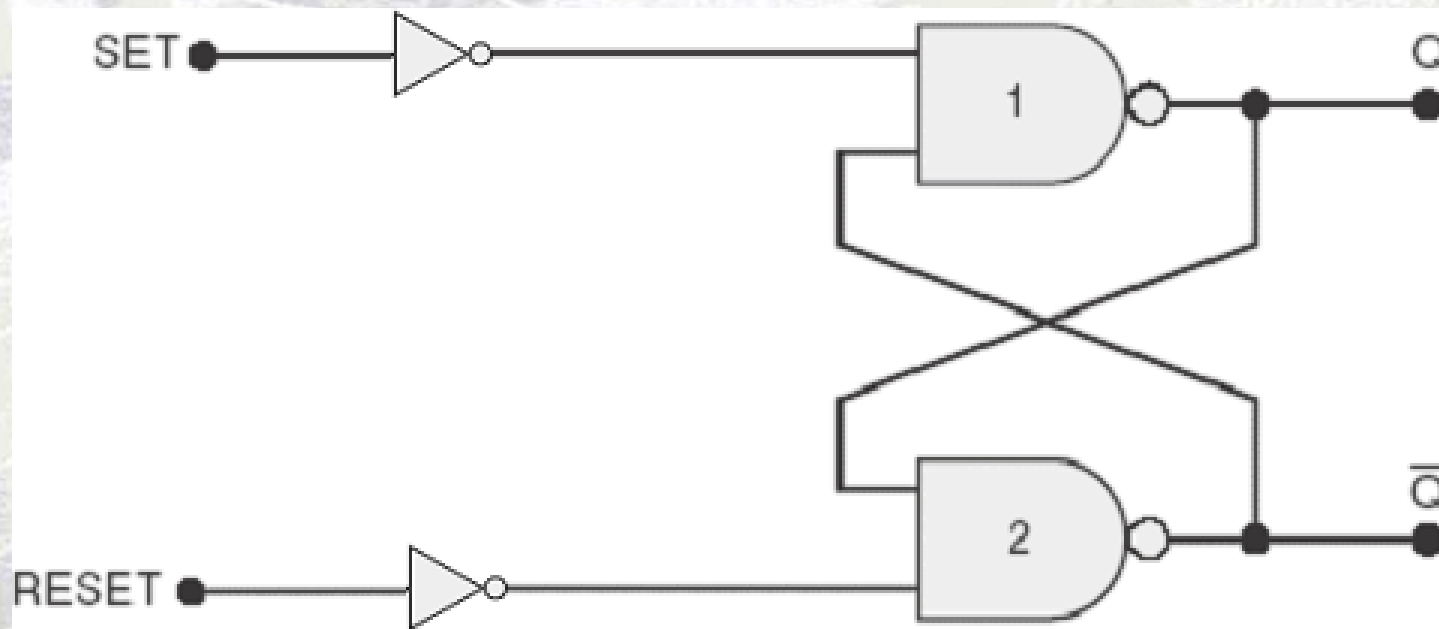
$$S = 0, \quad R = 0, \quad Q_a = 1, \quad \overline{Q}_a = 0$$



# Funcionamento do Flip-Flop

2º Caso)

$$S = 0, \quad R = 0, \quad Q_a = 1, \quad \overline{Q_a} = 0$$



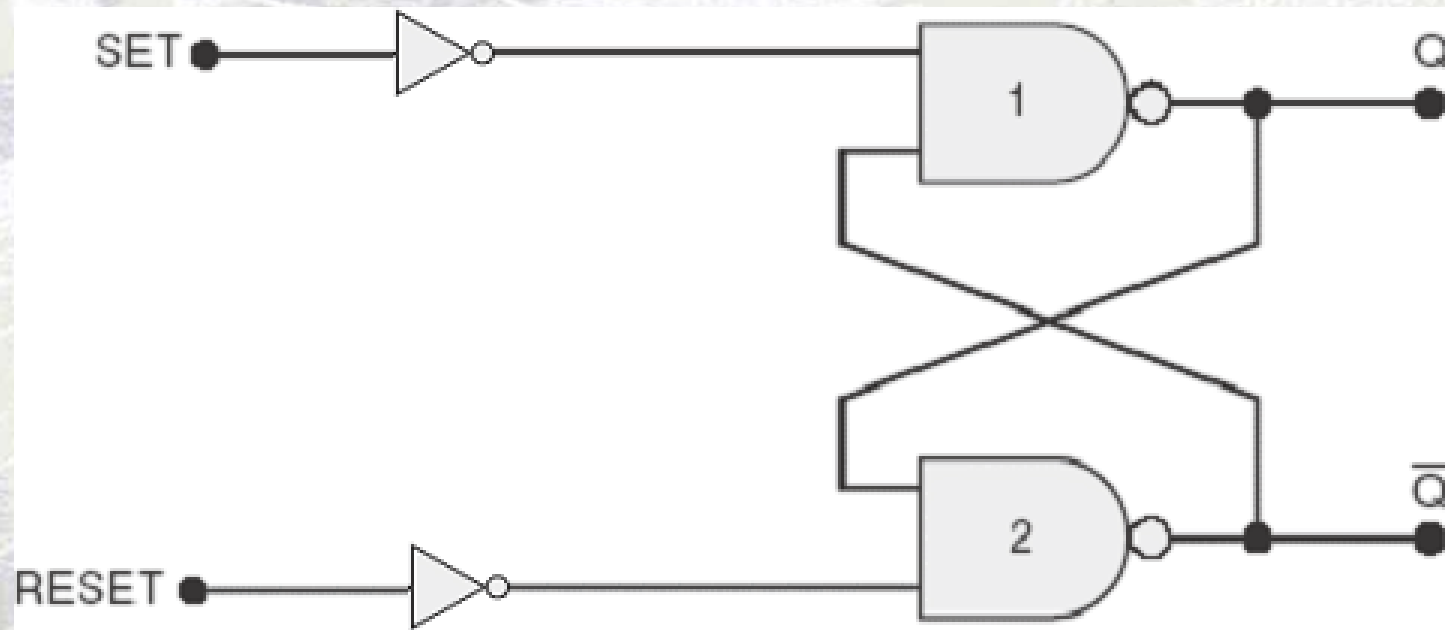
Situação idêntica ao 1º caso, onde  $Q_f = Q_a$ .



# Funcionamento do Flip-Flop

3º Caso)

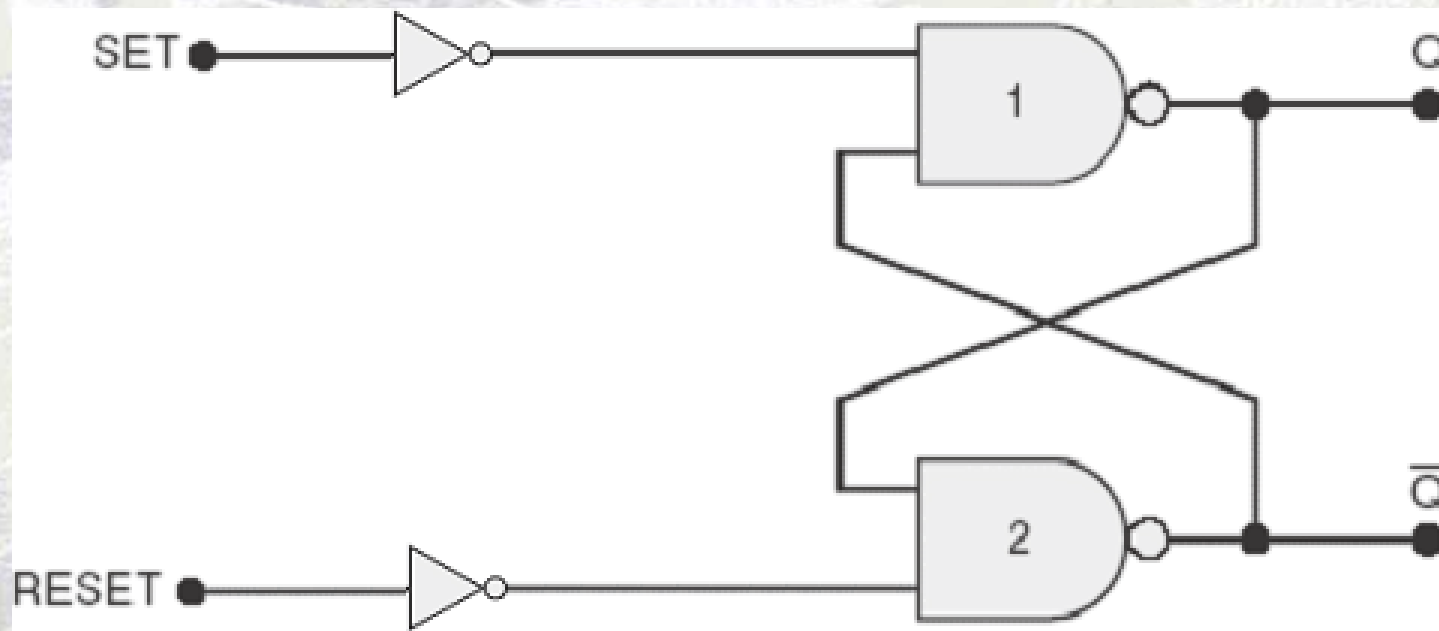
$$S = 0, \quad R = 1, \quad Q_a = 0, \quad \overline{Q}_a = 1$$



# Funcionamento do Flip-Flop

3º Caso)

$$S = 0, \quad R = 1, \quad Q_a = 0, \quad \overline{Q}_a = 1$$

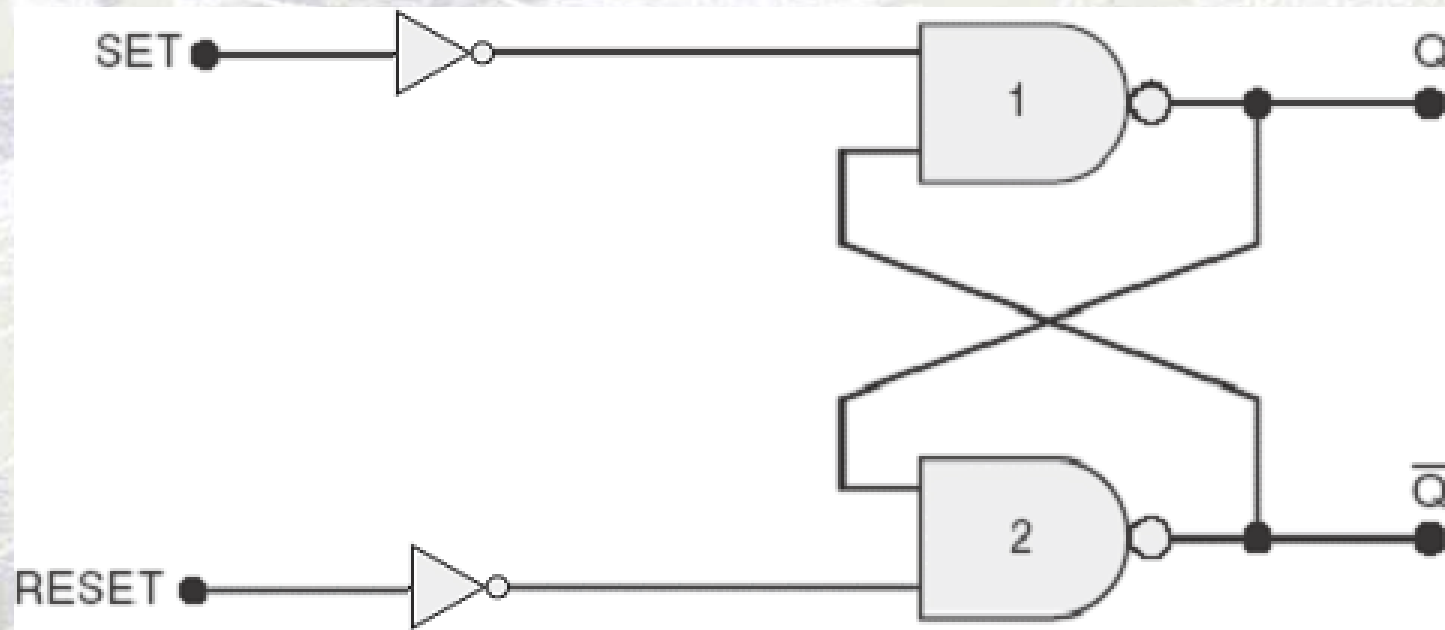


Também é um estado estável, onde a saída  $Q$  irá assumir o valor 0.

# Funcionamento do Flip-Flop

4º Caso)

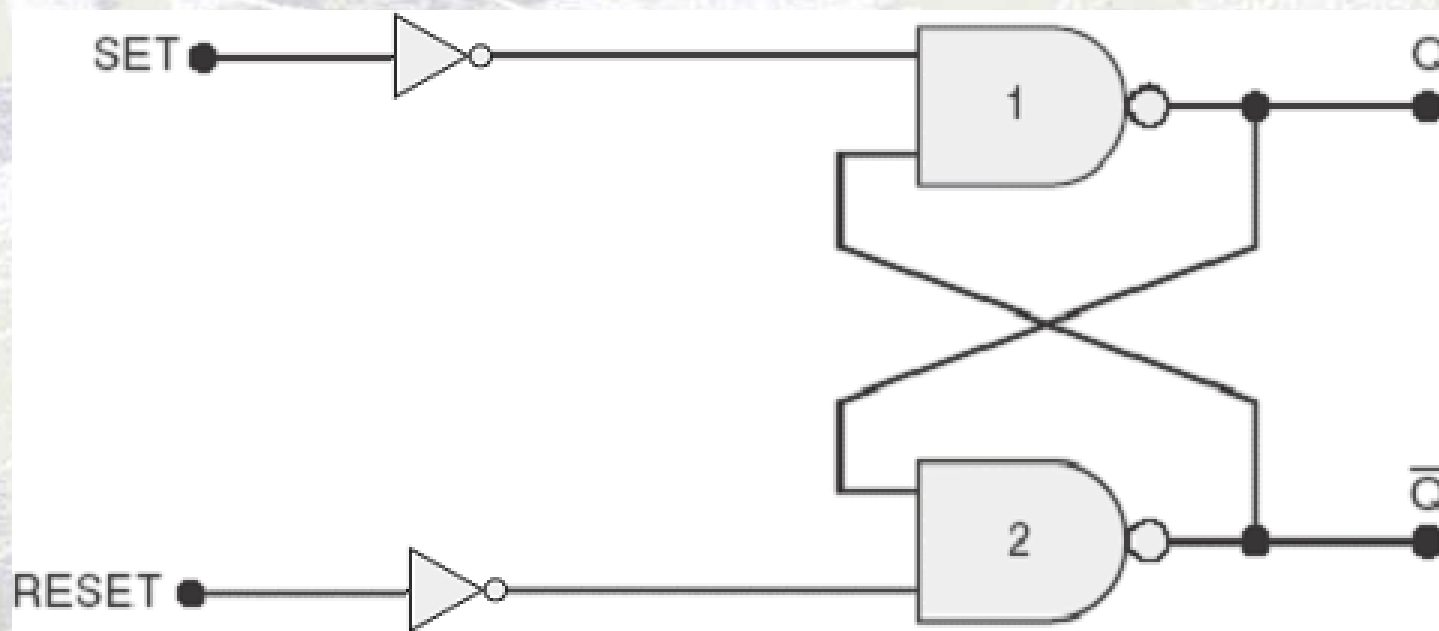
$$S = 0, \quad R = 1, \quad Q_a = 1, \quad \overline{Q}_a = 0$$



# Funcionamento do Flip-Flop

4º Caso)

$$S = 0, \quad R = 1, \quad Q_a = 1, \quad \overline{Q}_a = 0$$



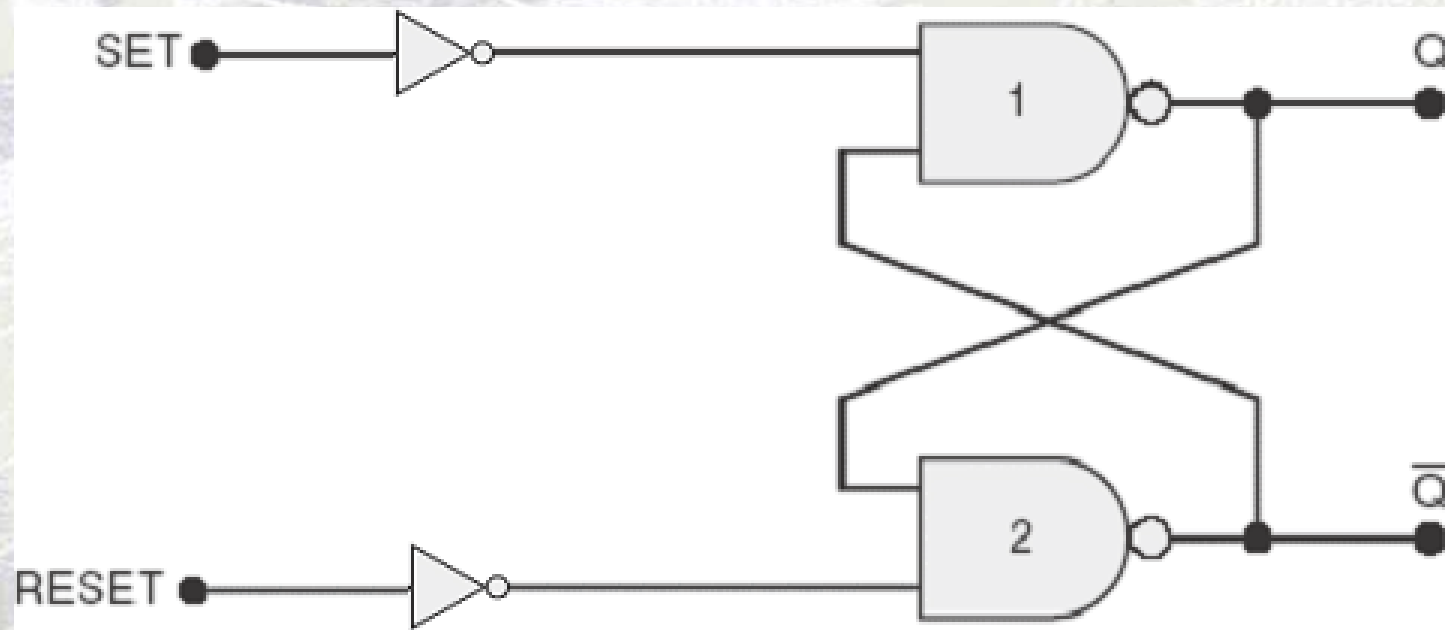
Neste caso a saída  $Q$  é forçada a assumir o valor 0. Assim,  $Q_f = 0$ !



# Funcionamento do Flip-Flop

5º Caso)

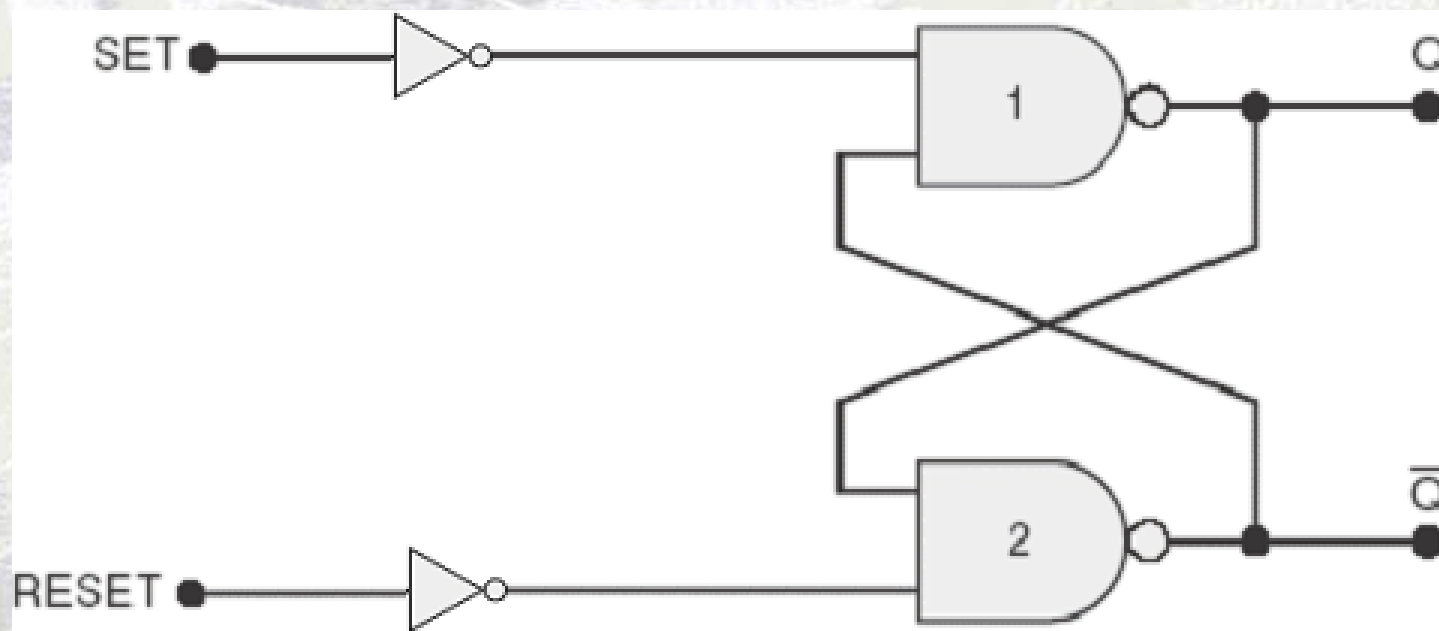
$$S = 1, \quad R = 0, \quad Q_a = 0, \quad \overline{Q}_a = 1$$



# Funcionamento do Flip-Flop

5º Caso)

$$S = 1, \quad R = 0, \quad Q_a = 0, \quad \overline{Q}_a = 1$$

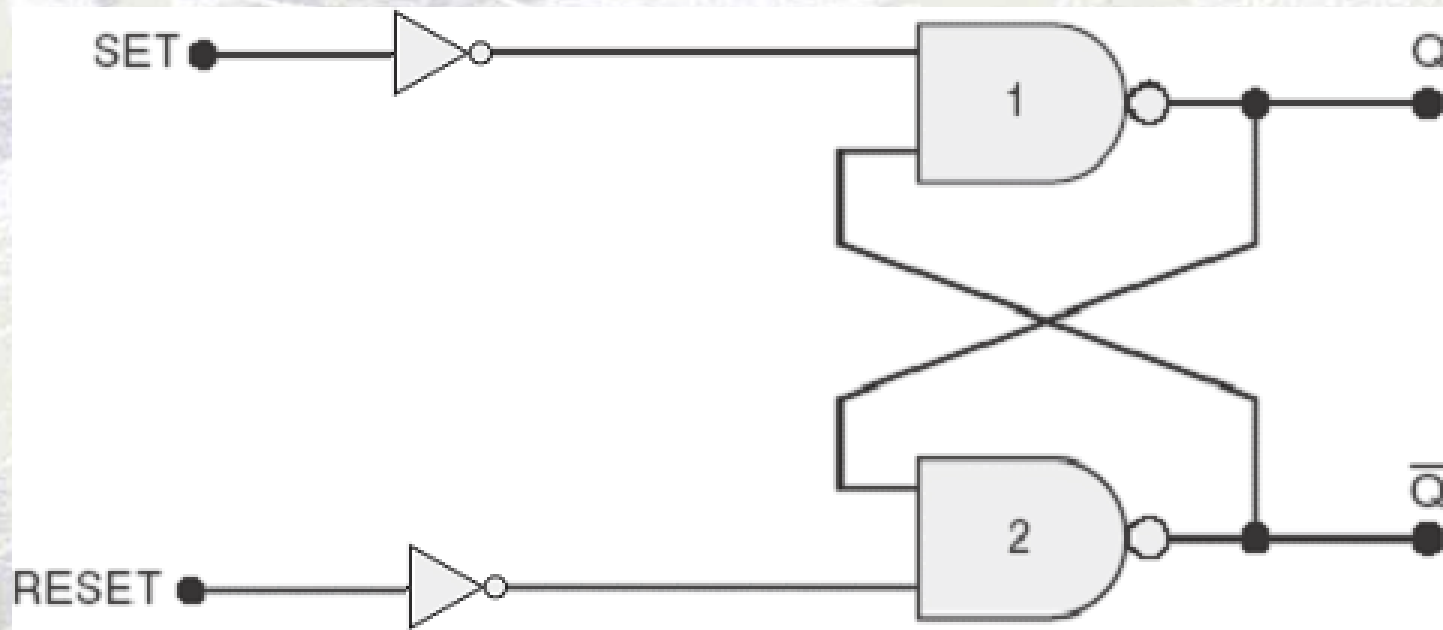


Neste caso a saída Q é forçada a assumir o valor 1. Assim,  $Q_f = 1$ !

# Funcionamento do Flip-Flop

6º Caso)

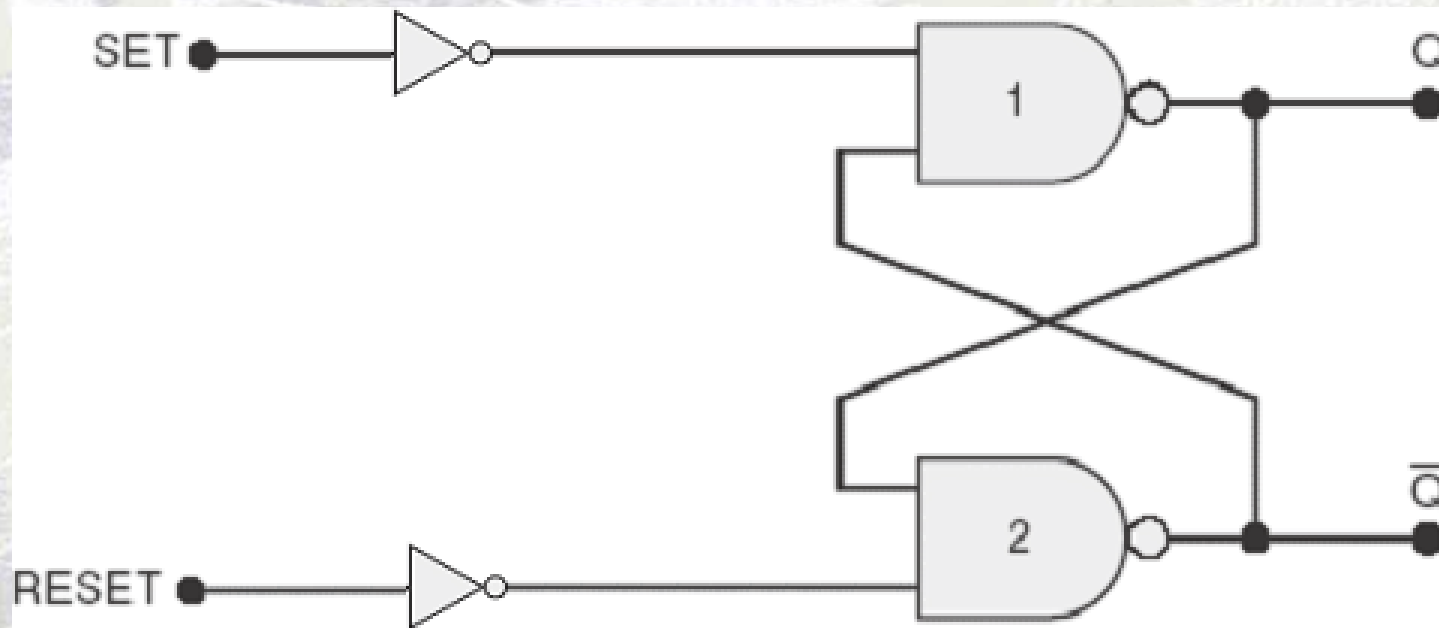
$$S = 1, \quad R = 0, \quad Q_a = 1, \quad \overline{Q}_a = 0$$



# Funcionamento do Flip-Flop

6º Caso)

$$S = 1, \quad R = 0, \quad Q_a = 1, \quad \overline{Q}_a = 0$$



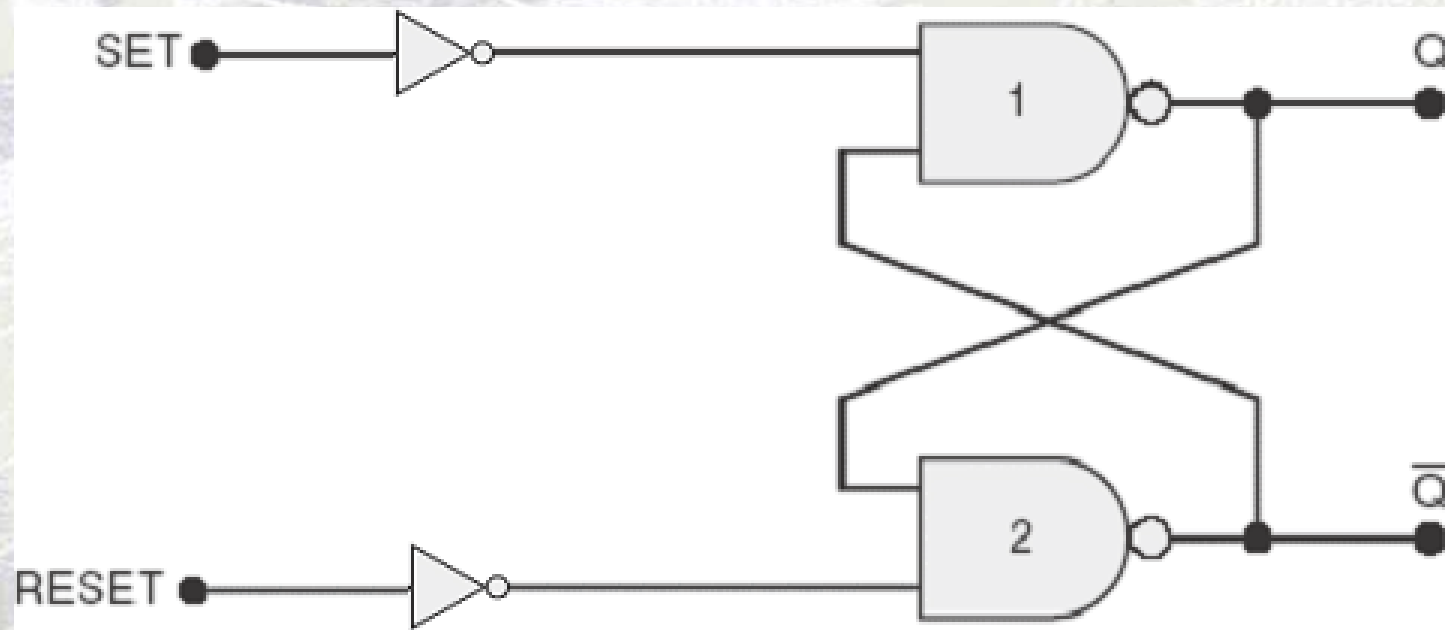
Situação idêntica ao 5º caso, onde  $Q_f = 1$ .



# Funcionamento do Flip-Flop

7º Caso)

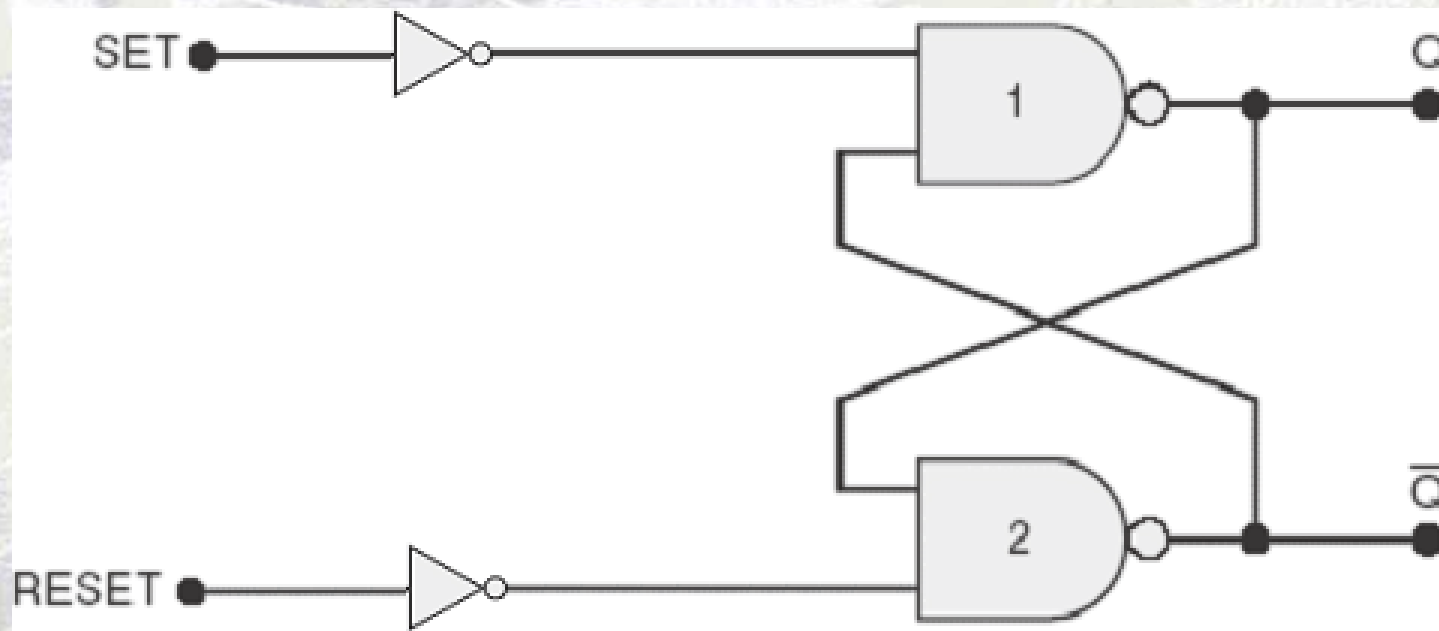
$$S = 1, \quad R = 1, \quad Q_a = 0, \quad \overline{Q}_a = 1$$



# Funcionamento do Flip-Flop

7º Caso)

$$S = 1, \quad R = 1, \quad Q_a = 0, \quad \overline{Q}_a = 1$$

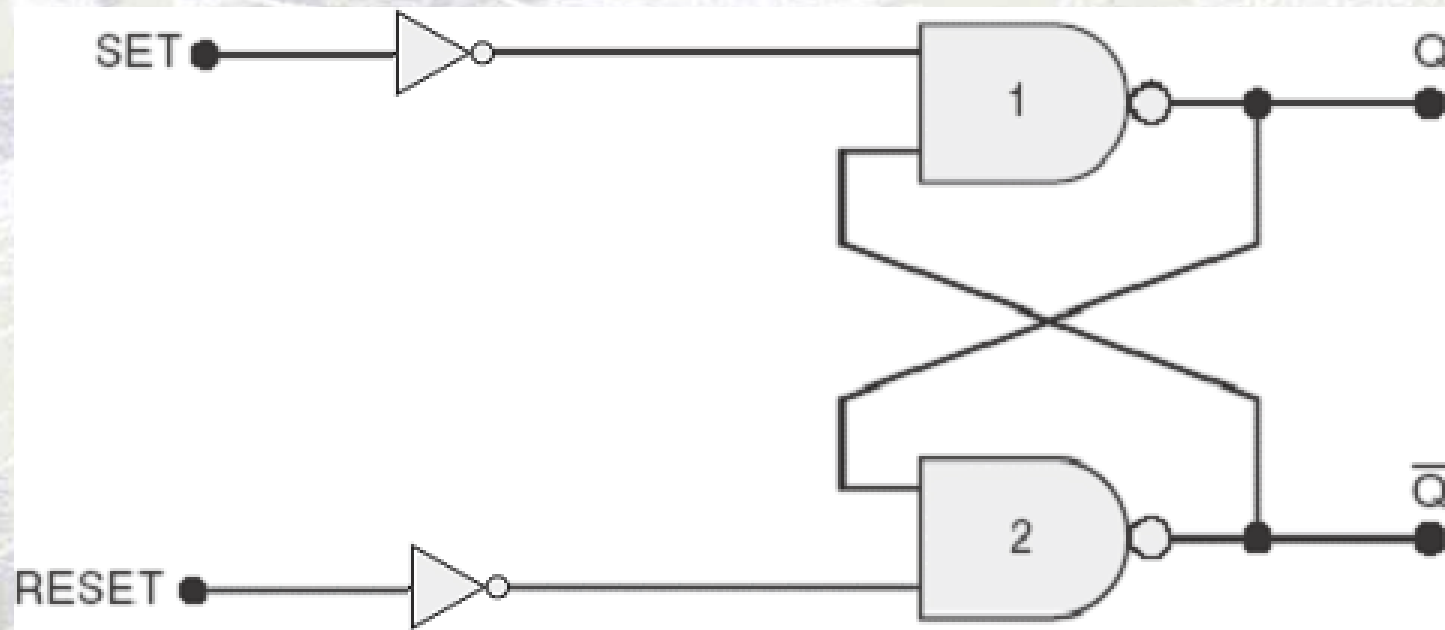


Neste caso, as duas saídas do FF são forçadas a assumir o valor 1. Este é um estado instável do Flip-Flop RS!

# Funcionamento do Flip-Flop

8º Caso)

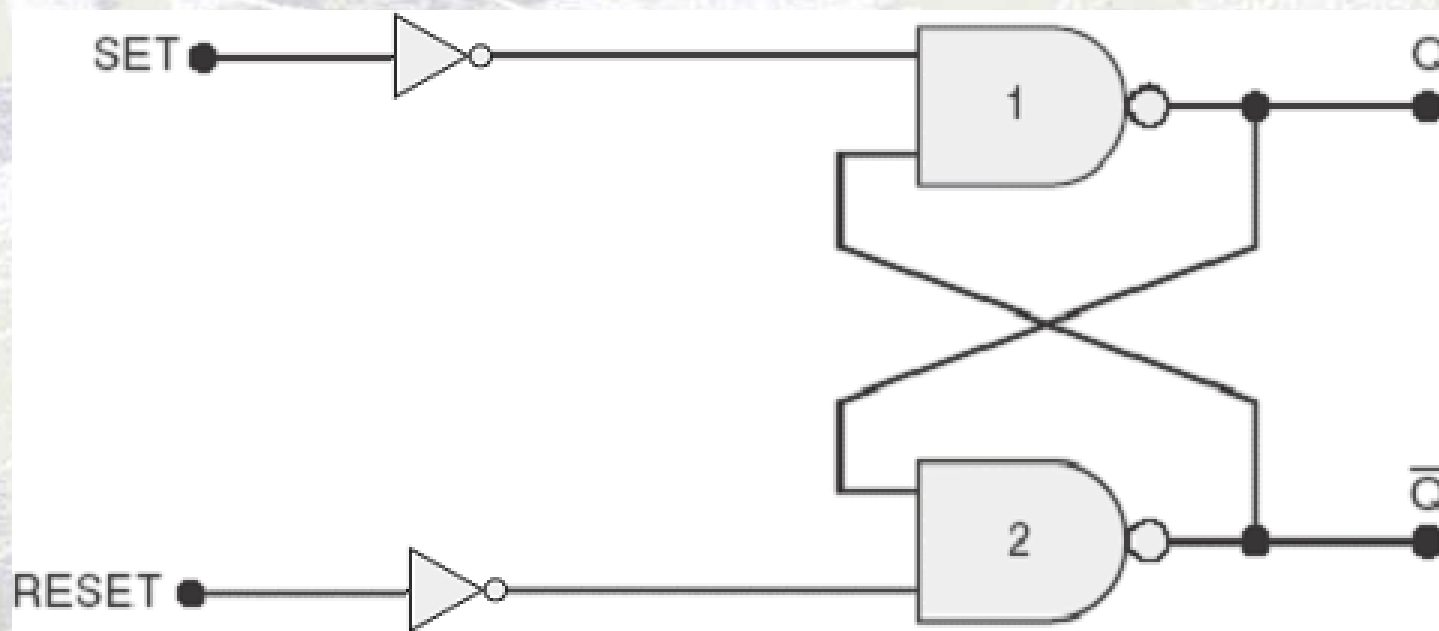
$$S = 1, \quad R = 1, \quad Q_a = 1, \quad \overline{Q}_a = 0$$



# Funcionamento do Flip-Flop

8º Caso)

$$S = 1, \quad R = 1, \quad Q_a = 1, \quad \overline{Q}_a = 0$$



Este é um estado instável do Flip-Flop RS, idêntico ao 7º caso.

# Funcionamento do Flip-Flop

Assim, temos:

Casos	Set (S)	Reset (R)	Qa	Qf	$\overline{Qf}$
1º	0	0	0	0	1
2º	0	0	1	1	0
3º	0	1	0	0	1
4º	0	1	1	0	1
5º	1	0	0	1	0
6º	1	0	1	1	0
7º	1	1	0	1	1
8º	1	1	1	1	1

Resumindo, temos:

Set (S)	Reset (R)	Qf
0	0	Qa
0	1	0
1	0	1
1	1	

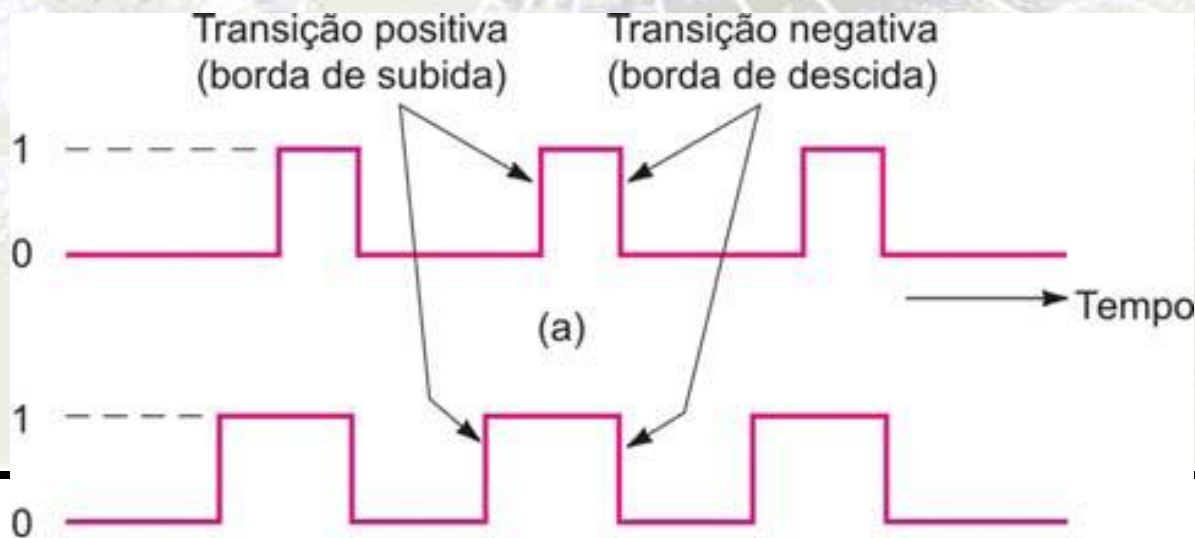
← Não Permitido!



# Sinais de Clock

Os circuitos digitais (ou sistemas digitais) podem funcionar tanto no modo **assíncrono** quanto no **síncrono**. A diferença entre eles é:

- nos **sistemas assíncronos**, as saídas de circuitos lógicos podem mudar de estado a qualquer momento em que uma ou mais entradas mudem de estado.
- nos **sistemas síncronos**, os momentos exatos em que uma saída qualquer pode mudar de estado são determinados por um sinal normalmente denominado **CLOCK**.



# Sinais de Clock

## Sistemas Síncronos

1. Os sistemas digitais, em sua maioria, são síncronos (embora tenham algumas partes assíncronas).
2. A análise de defeitos desses sistemas é mais fácil pois as saídas desses circuitos só podem mudar de estado em instantes específicos.
3. A sincronização dos eventos com o sinal de clock é obtida com o uso de **flip-flops com clock** que são projetados para mudarem de estado em uma das transições do sinal de clock.

# Sinais de Clock

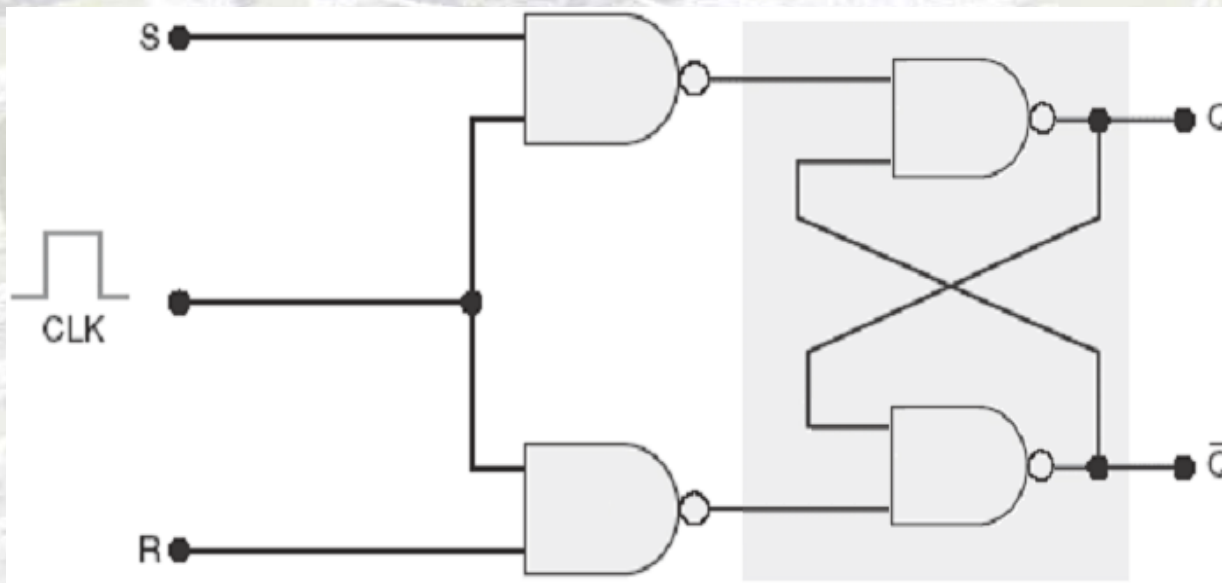
## Sistemas Síncronos

1. Os sistemas digitais, em sua maioria, são síncronos (embora tenham algumas partes assíncronas).
2. A análise de defeitos desses sistemas é mais fácil pois as saídas desses circuitos só podem mudar de estado em instantes específicos.
3. A sincronização dos eventos com o sinal de clock é obtida com o uso de **flip-flops com clock** que são projetados para mudarem de estado em uma das transições do sinal de clock.

Vários tipos de FFs com clock são usados em um grande número de aplicações, como por exemplo, contadores, registradores, divisores de frequência, etc.

# Flip-Flop com Entrada de Clock

Para que o Flip-Flop RS seja controlado por uma sequência de pulsos, basta substituir os **INVERSORES** por portas **NAND**. O circuito será:

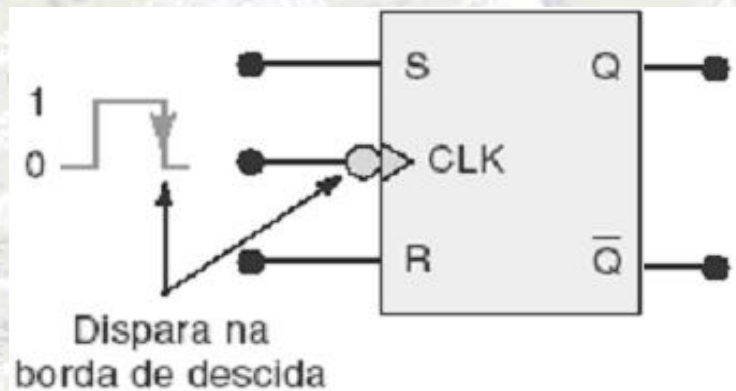
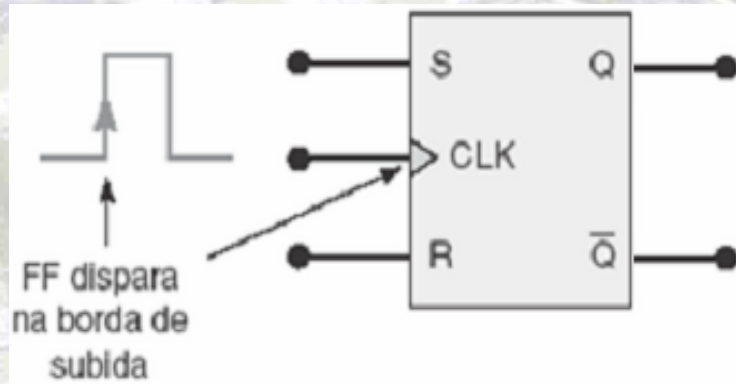


Clock	Qf
0	Qa
1	FF-RS

Quando a entrada de **CLOCK** for igual a zero, o Flip-Flop permanece no seu estado anterior, mesmo que as entradas RS variem. Quando o CLOCK assumir o valor 1, o circuito se comporta como um FF-RS básico.

# Flip-Flop com Entrada de Clock

Bloco lógico do Flip-Flop com entrada de CLOCK:

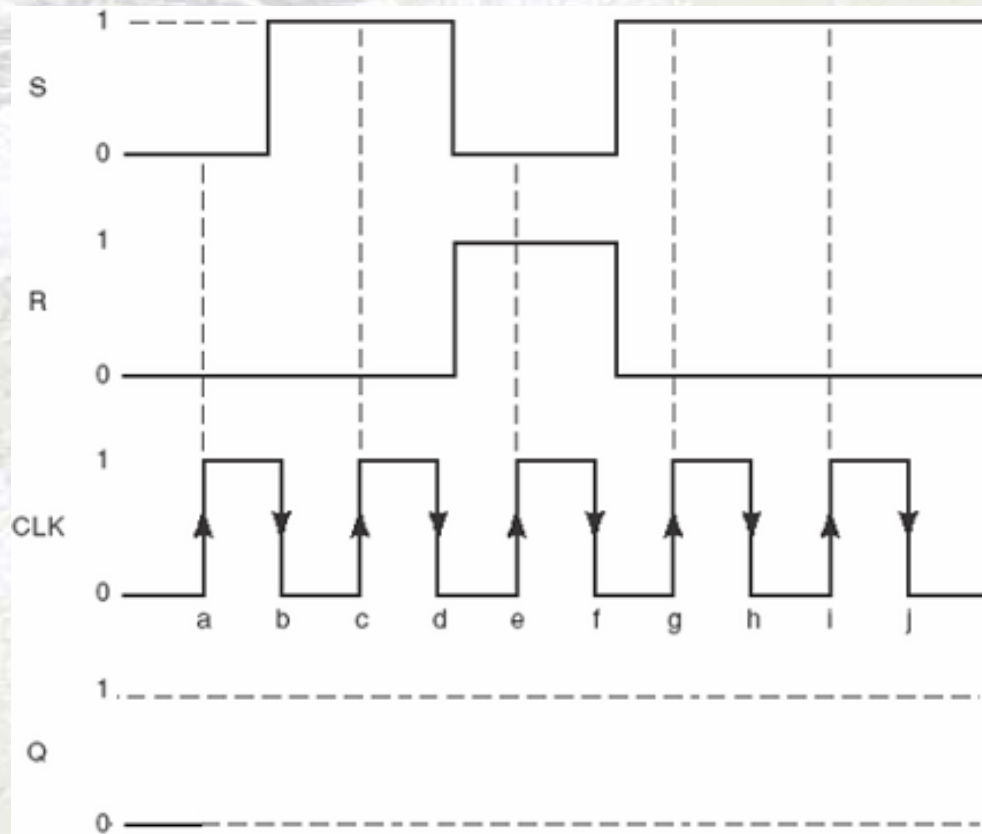


Os Flip-Flops são, normalmente, disparados por bordas (de subida ou descida). Já os Latches possuem uma entrada de habilitação (ENABLE), sendo os mesmos disparados por nível (Alto ou Baixo).



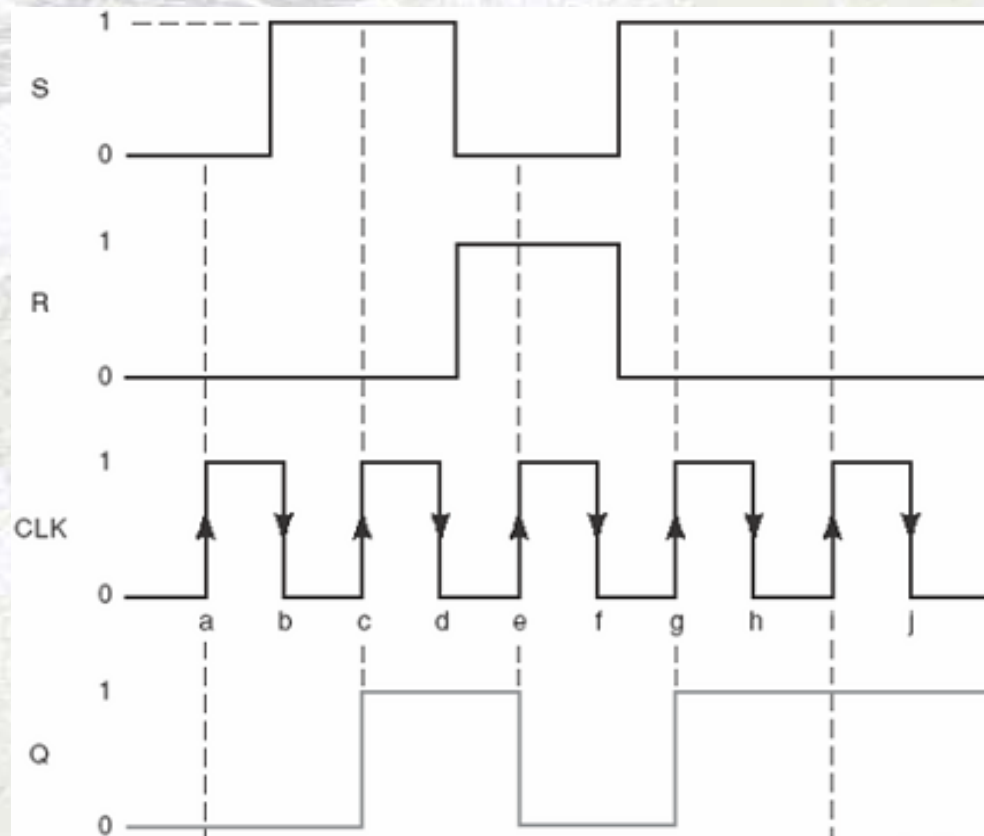
# Flip-Flop com Entrada de Clock

**Exercício - 1:** Qual será a forma de onda na saída para as entradas RS, sendo o FF sensível a borda de subida:



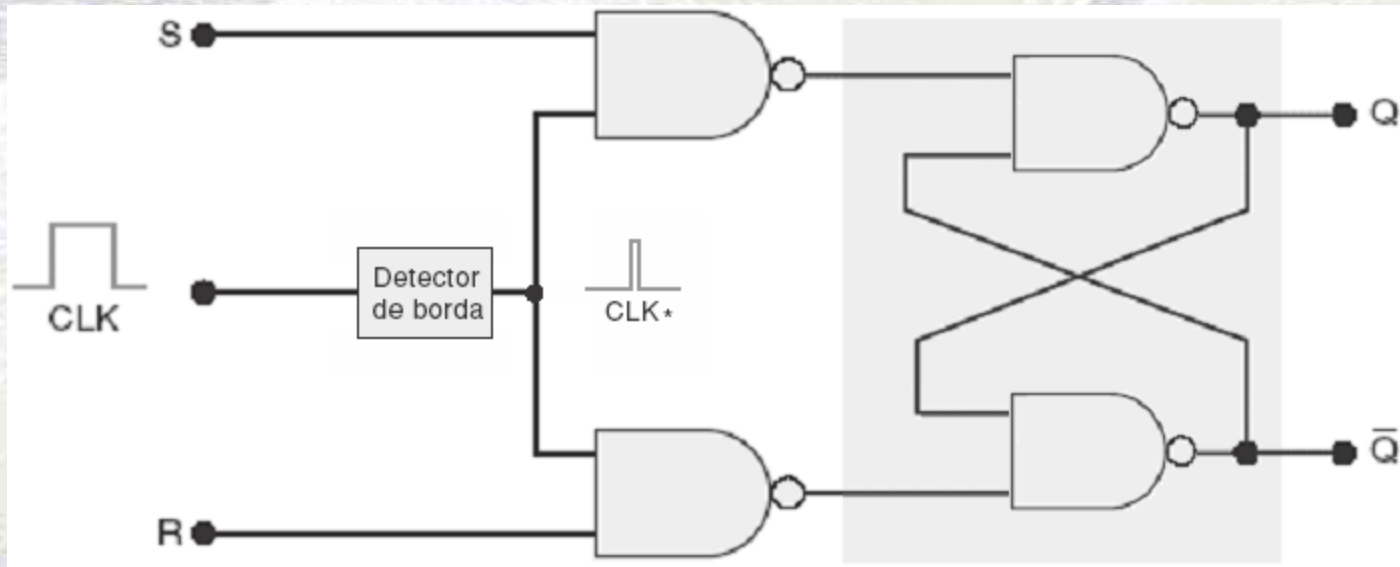
# Flip-Flop com Entrada de Clock

**Exercício - 1:** Qual será a forma de onda na saída para as entradas RS, sendo o FF sensível a borda de subida:



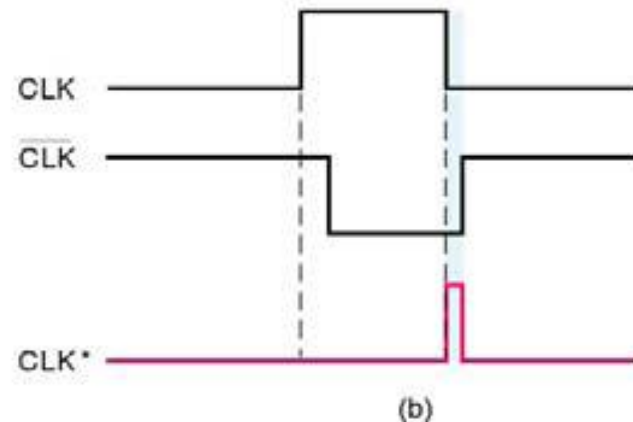
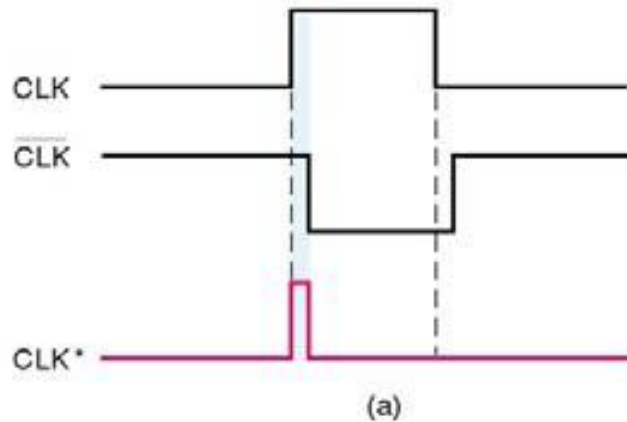
# Circuito Detector de Borda

Bloco lógico do Flip-Flop RS básico, com entrada de CLOCK, disparado por borda é:



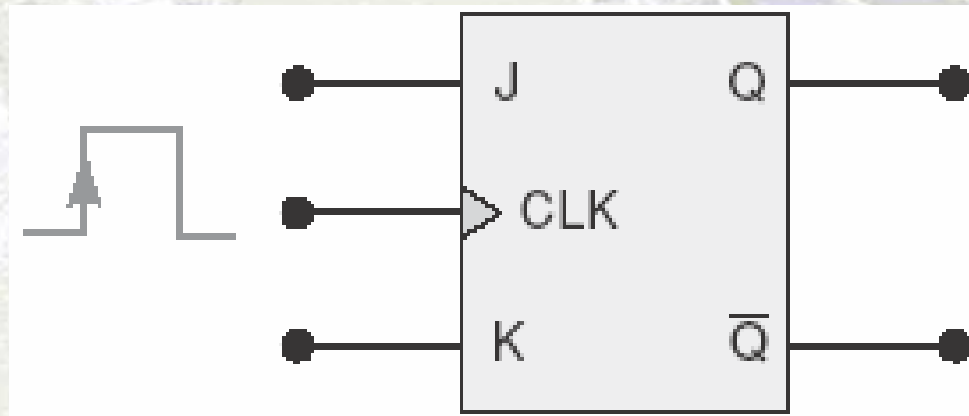
# Circuito Detector de Borda

O circuito detector de borda pode ser implementado a partir dos circuitos abaixo, sendo: (a) borda positiva e (b) borda negativa.



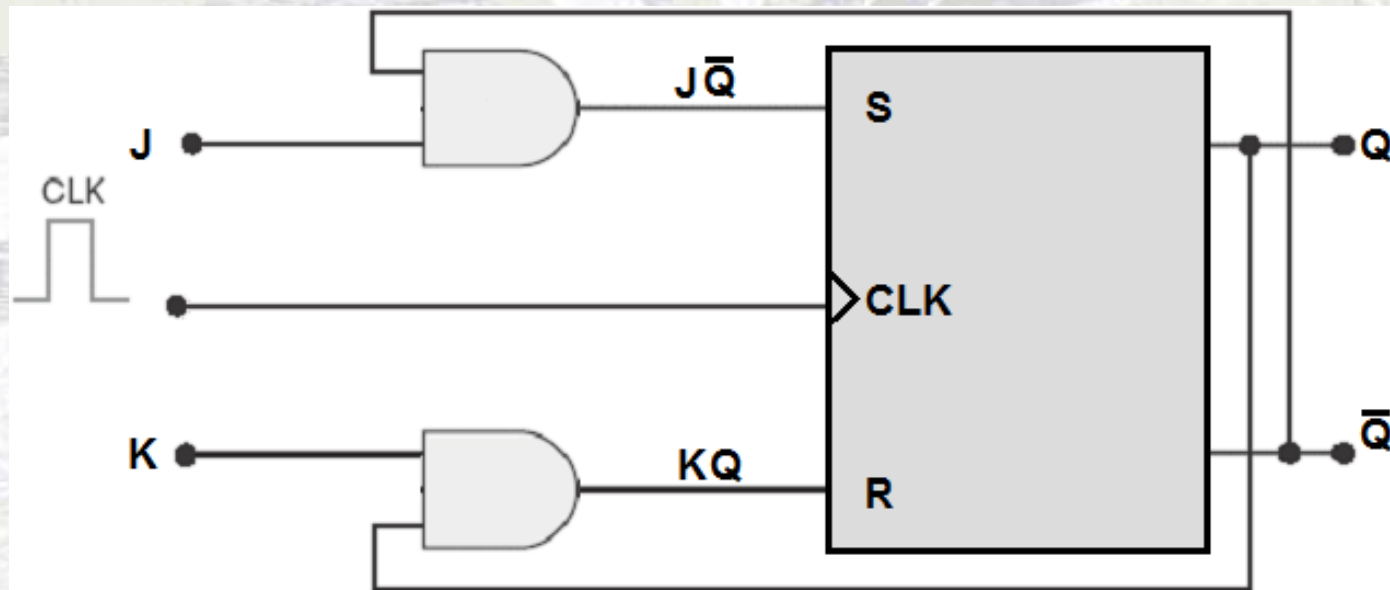
# Flip-Flop JK

A figura abaixo apresenta o símbolo lógico para um **FF JK com CLK** disparado na borda positiva. As entradas J e K controlam o estado do FF da mesma forma que fazem as entradas S e R para um FF RS básico com CLK, **exceto por uma importante diferença.**





# Flip-Flop JK



Casos	J	K	Qa	$\overline{Qa}$	S	R	Qf
1º	0	0	0	1	0	0	0
2º	0	0	1	0	0	0	1
3º	0	1	0	1	0	0	0
4º	0	1	1	0	0	1	0
5º	1	0	0	1	1	0	1
6º	1	0	1	0	0	0	1
7º	1	1	0	1	1	0	1
8º	1	1	1	0	0	1	0

# Flip-Flop JK

Resumindo a Tabela Verdade do FF JK é:

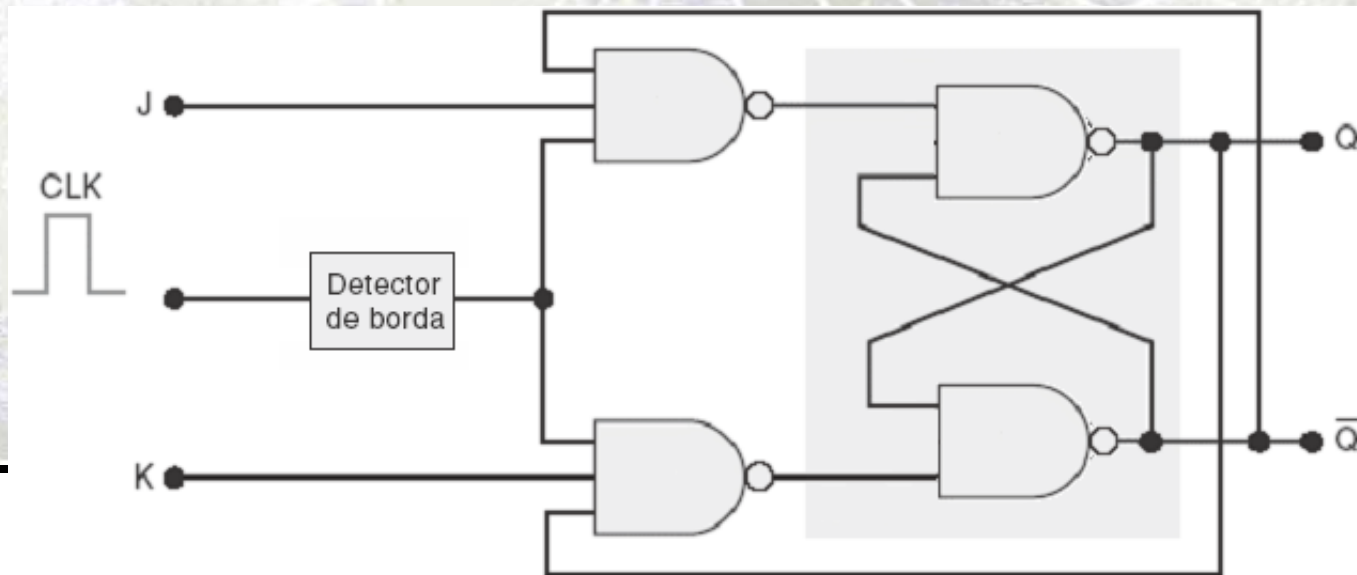
J	K	Qf	
0	0	Qa	não muda
0	1	0	reset
1	0	1	set
1	1	$\overline{Qa}$	toggle

# Flip-Flop JK

Resumindo a Tabela Verdade do FF JK é:

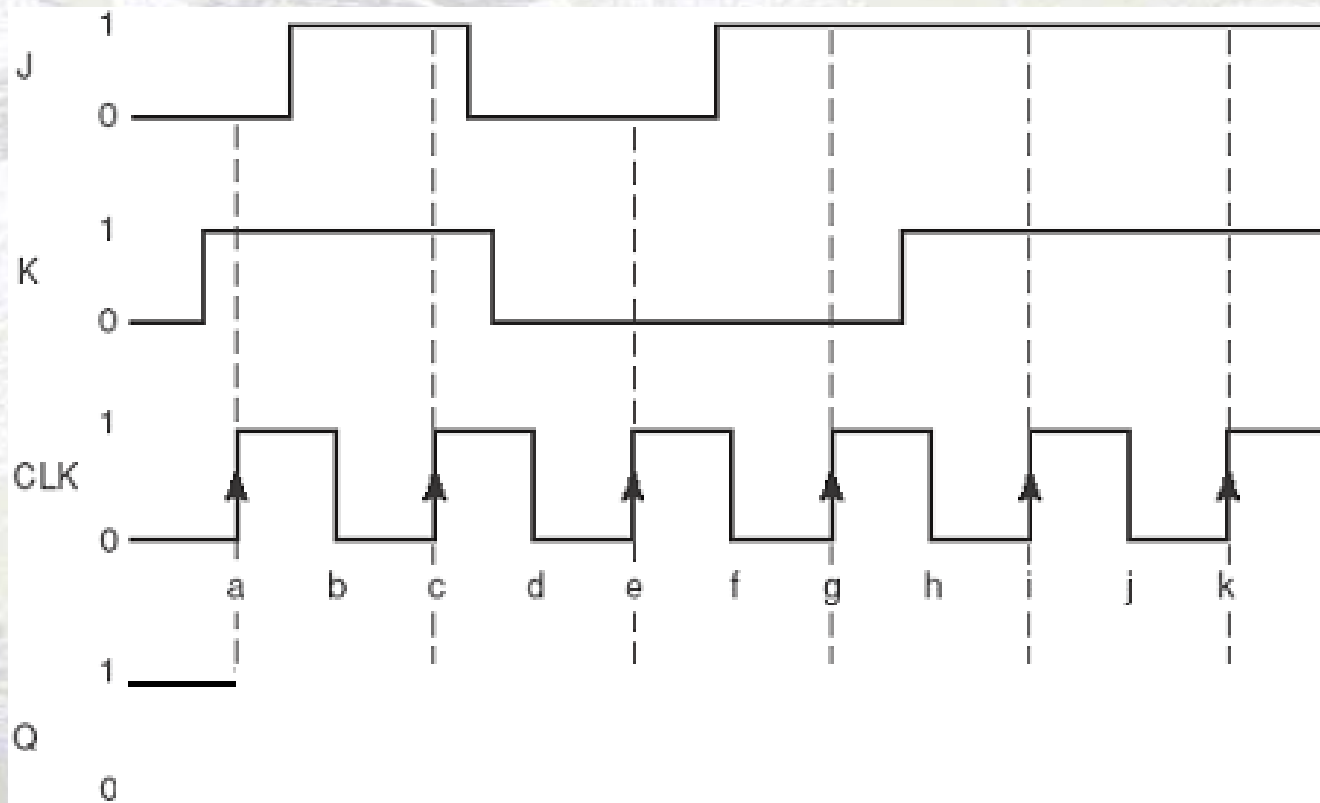
J	K	Qf	
0	0	Qa	não muda
0	1	0	reset
1	0	1	set
1	1	$\overline{Qa}$	toggle

Outra possibilidade de construção do FF JK



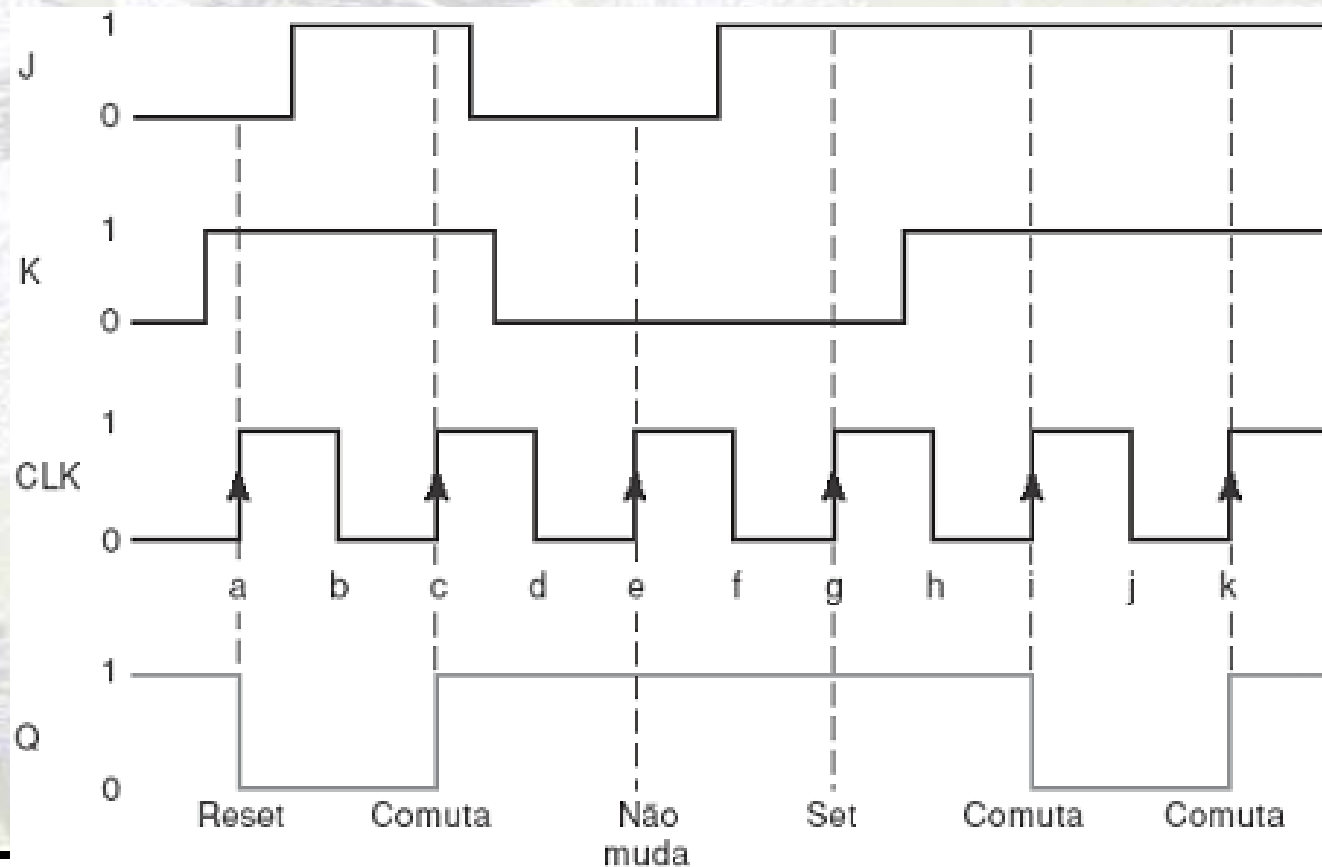
# Flip-Flop JK

**Exercício 2** – Dado um Flip-Flop JK sensível a borda de subida, esboce a forma de onda na saída Q:



# Flip-Flop JK

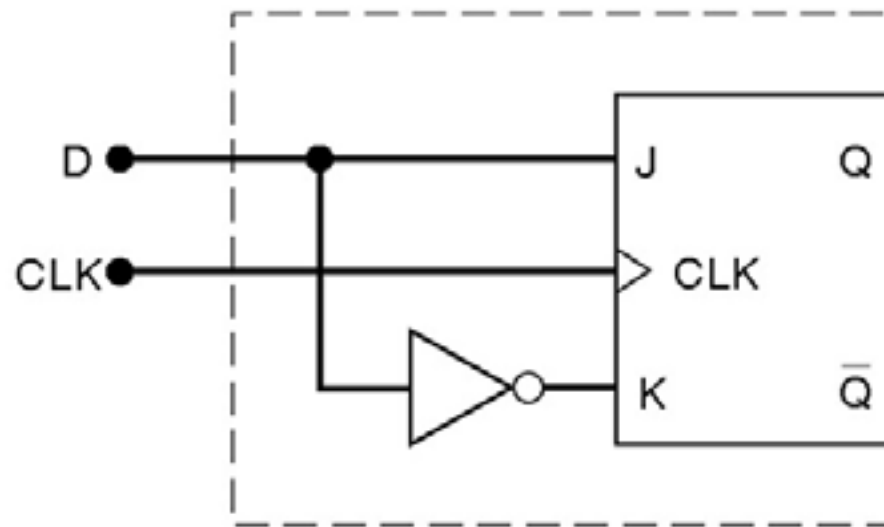
**Exercício 2** – Dado um Flip-Flop JK sensível a borda de subida, esboce a forma de onda na saída Q:



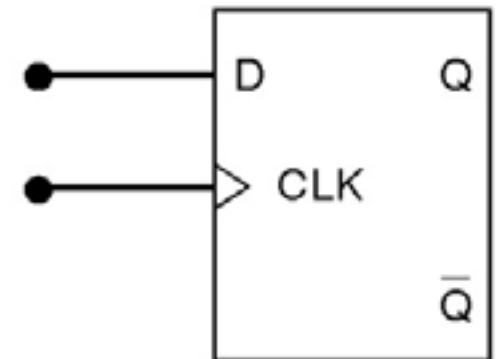


# Flip-Flop D

Um **Flip-Flop Tipo D**, disparado por borda, é facilmente implementado acrescentando um único INVERSOR a um FF JK disparado por borda, como ilustrado na figura abaixo.

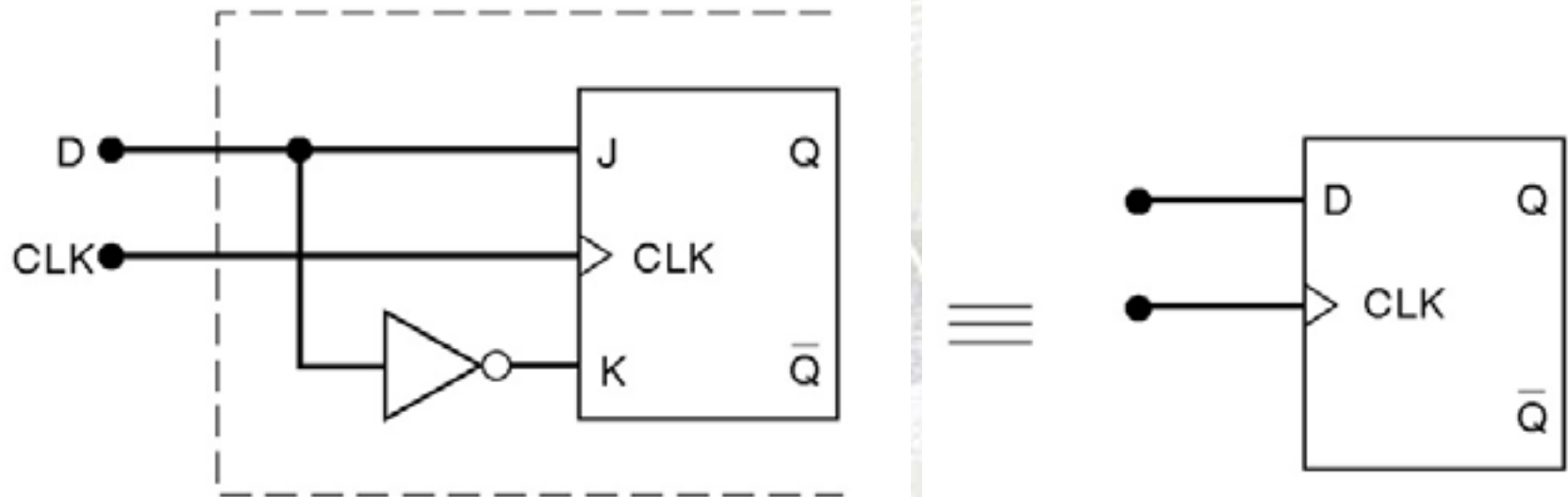


≡



# Flip-Flop D

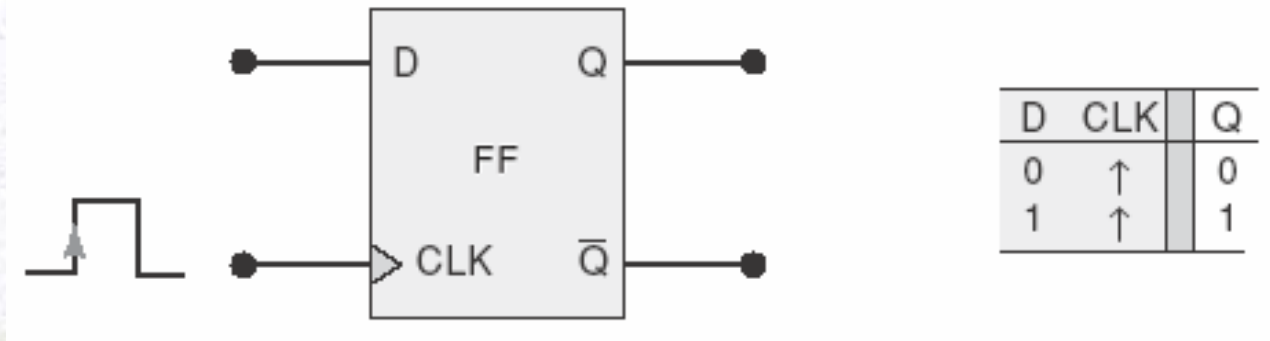
Um **Flip-Flop Tipo D**, disparado por borda, é facilmente implementado acrescentando um único INVERSOR a um FF JK disparado por borda, como ilustrado na figura abaixo.



**Importante:** Nesta configuração (FF Tipo D) as entradas JK podem assumir os valores 0 e 1 para  $D=0$ , ou 1 e 0 quando  $D=1$ .

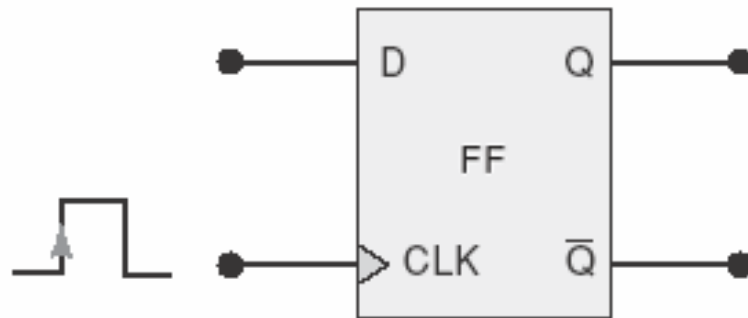
# Flip-Flop D

A figura abaixo mostra o símbolo lógico e a tabela-verdade para um **Flip-Flop D com CLK**. Ao contrário dos FF RS e JK, o FF D tem apenas uma entrada de controle síncrona, **entrada D**, que representa a palavra **data (dado)**.



# Flip-Flop D

A figura abaixo mostra o símbolo lógico e a tabela-verdade para um **Flip-Flop D com CLK**. Ao contrário dos FF RS e JK, o FF D tem apenas uma entrada de controle síncrona, **entrada D**, que representa a palavra **data (dado)**.

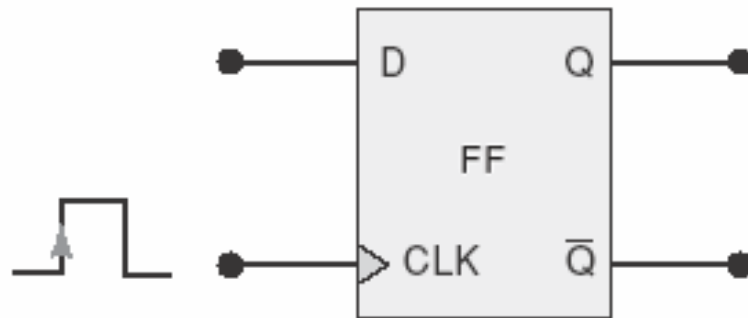


D	CLK	Q
0	↑	0
1	↑	1

A operação do **flip-flop D** é bastante simples: a saída **Q** irá para o mesmo estado lógico presente na **entrada D** quando ocorrer uma transição positiva em **CLK**.

# Flip-Flop D

A figura abaixo mostra o símbolo lógico e a tabela-verdade para um **Flip-Flop D com CLK**. Ao contrário dos FF RS e JK, o FF D tem apenas uma entrada de controle síncrona, **entrada D**, que representa a palavra **data (dado)**.



D	CLK	Q
0	↑	0
1	↑	1

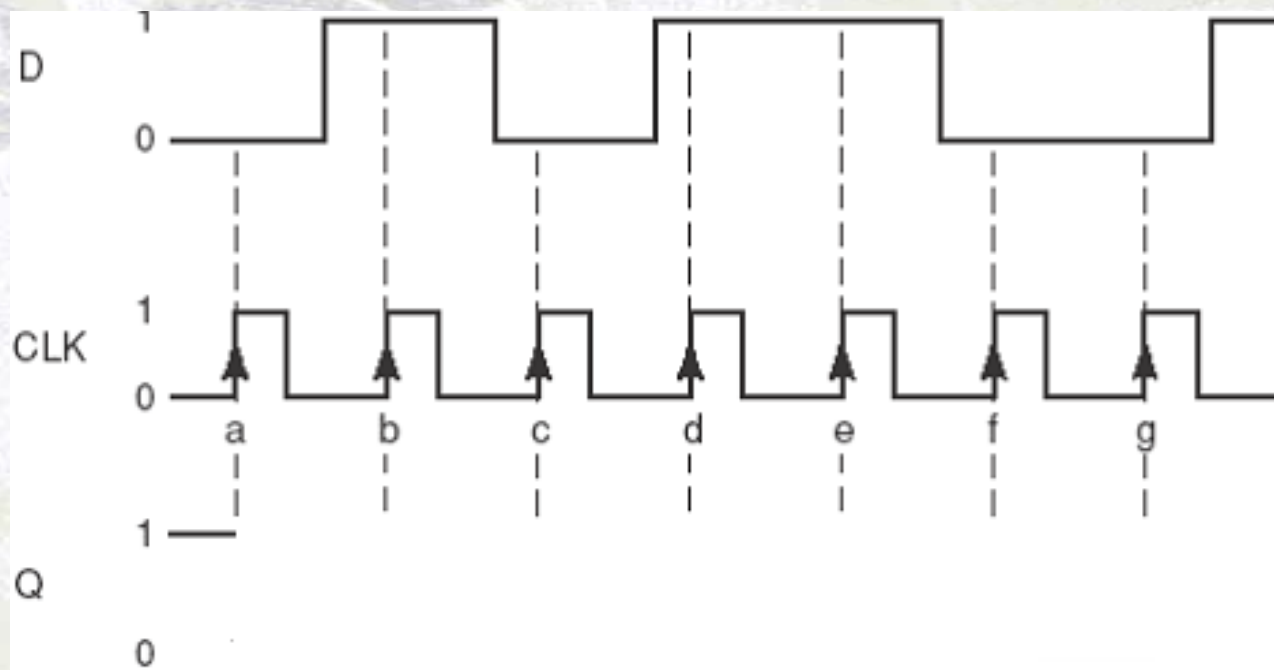
A operação do **flip-flop D** é bastante simples: a saída **Q** irá para o mesmo estado lógico presente na **entrada D** quando ocorrer uma transição positiva em **CLK**.

**Resumindo:** o nível lógico presente na entrada D será *armazenado* no Flip-Flop no instante em que ocorrer a transição do CLK.



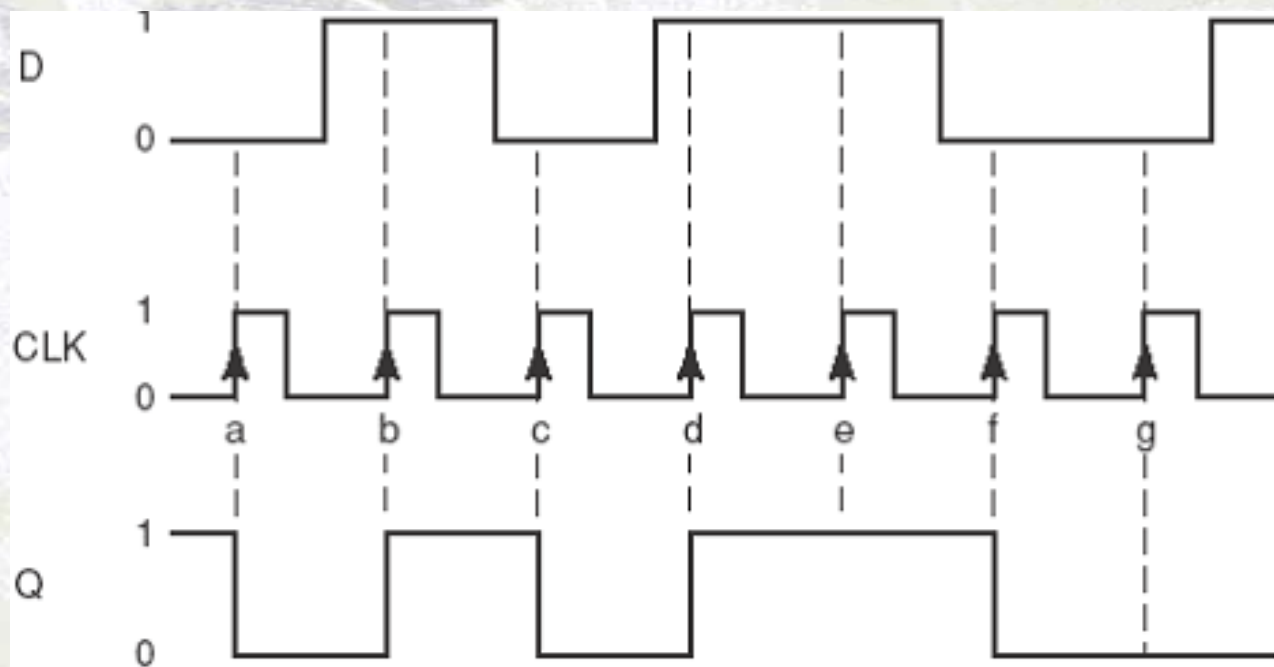
# Flip-Flop D

**Exercício 3** – Esboçar a forma de onda na saída Q, para o FF D sensível a borda de subida:



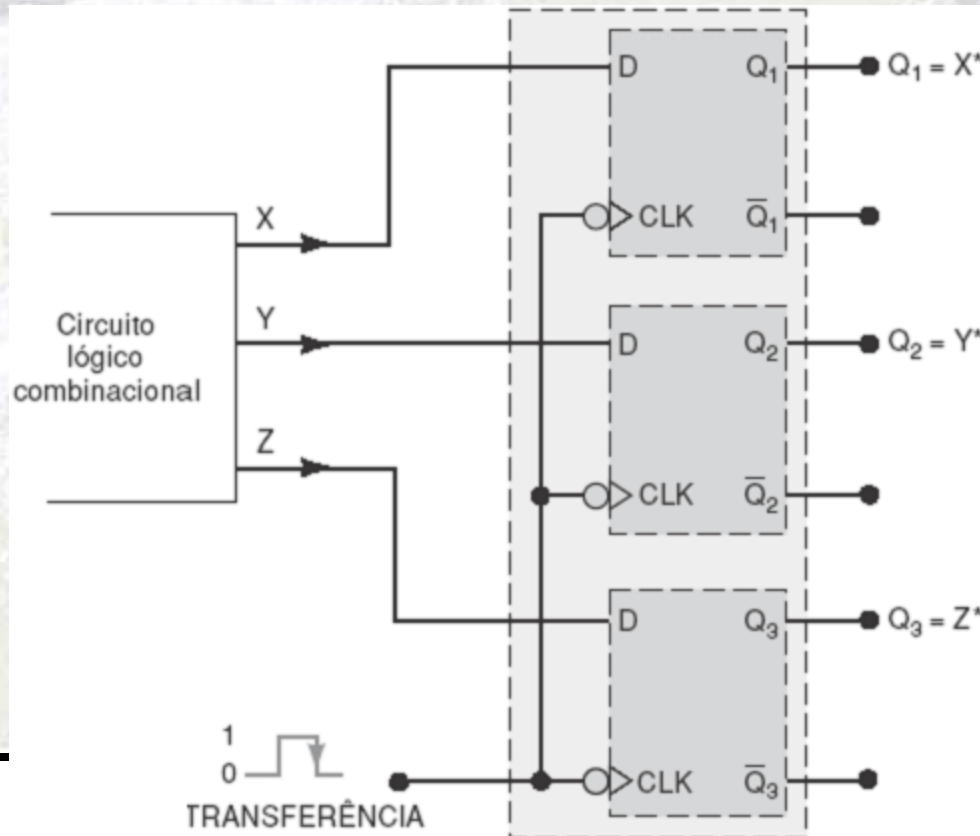
# Flip-Flop D

**Exercício 3** – Esboçar a forma de onda na saída Q, para o FF D sensível a borda de subida:



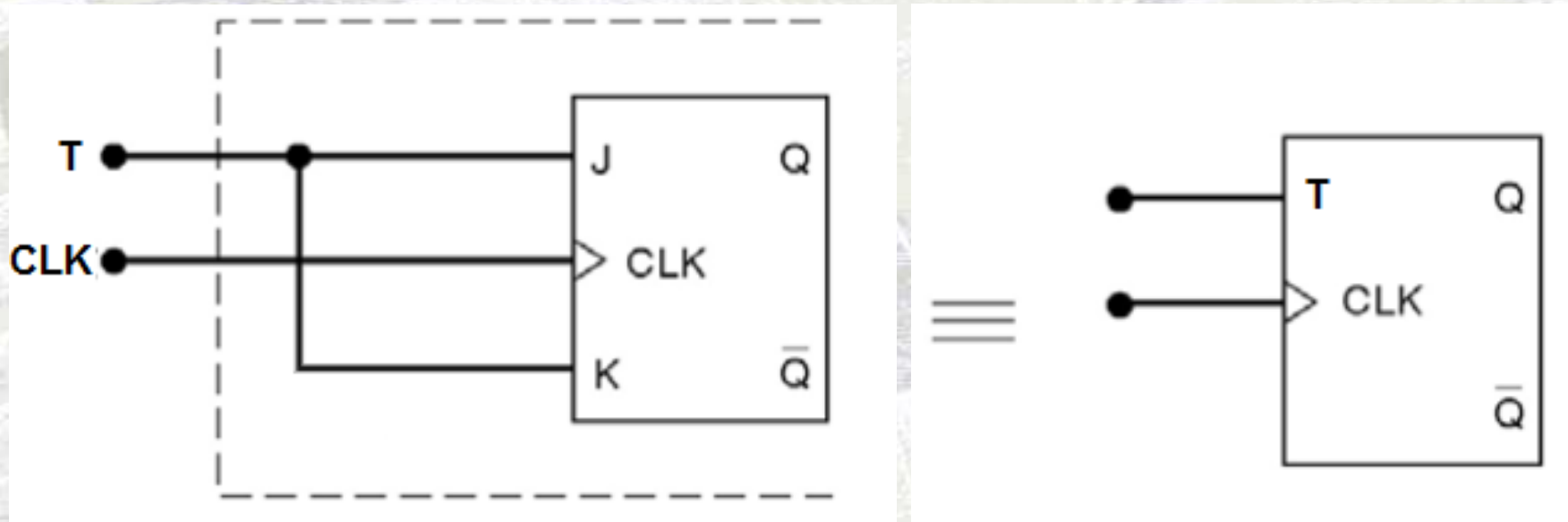
# Aplicação do Flip-Flop D

**Transferência de dados paralela sincronizada** - Como Q assume o mesmo valor de D em instantes determinados, as saídas Q1, Q2 e Q3 irão receber os dados fornecidos pelo circuito lógico combinacional, simultaneamente, somente quando acontecer um pulso de descida.



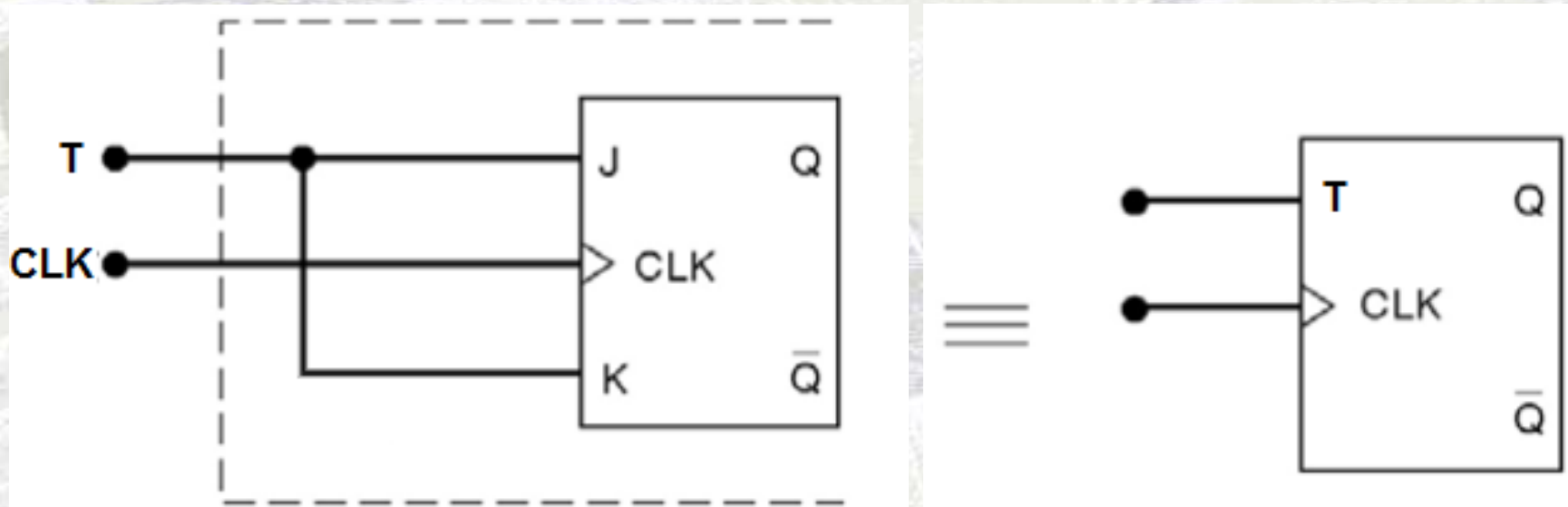
# Flip-Flop T

Um **Flip-Flop Tipo T**, disparado por borda, é facilmente implementado aplicando o mesmo sinal de entrada nas duas entradas JK, simultaneamente.



# Flip-Flop T

Um **Flip-Flop Tipo T**, disparado por borda, é facilmente implementado aplicando o mesmo sinal de entrada nas duas entradas JK, simultaneamente.



**Importante:** Nesta configuração (FF Tipo T) as entradas JK podem assumir o valor 0, para  $T=0$ , ou 1 para  $T=1$ .



# Flip-Flop JK com Preset / Clear

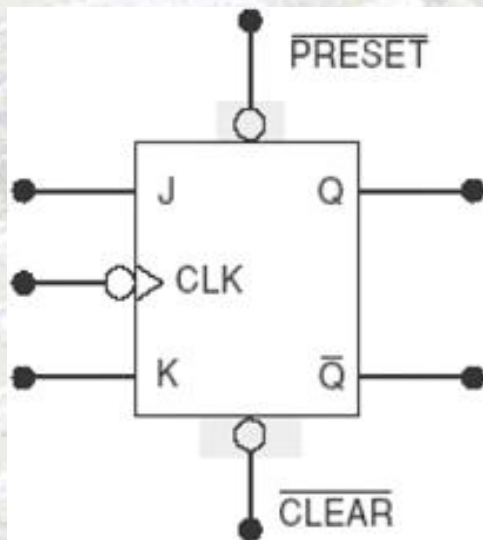
Para os FFs com **CLK** estudados até o momento, as entradas S, R, J, K, T e D têm sido denominadas **entradas de controle**. Essas entradas também podem ser ditas **entradas síncronas**, pois seus efeitos na saída do FF são sincronizados com a entrada **CLK**.



# Flip-Flop JK com Preset / Clear

Para os FFs com **CLK** estudados até o momento, as entradas S, R, J, K, T e D têm sido denominadas **entradas de controle**. Essas entradas também podem ser ditas **entradas síncronas**, pois seus efeitos na saída do FF são sincronizados com a entrada **CLK**.

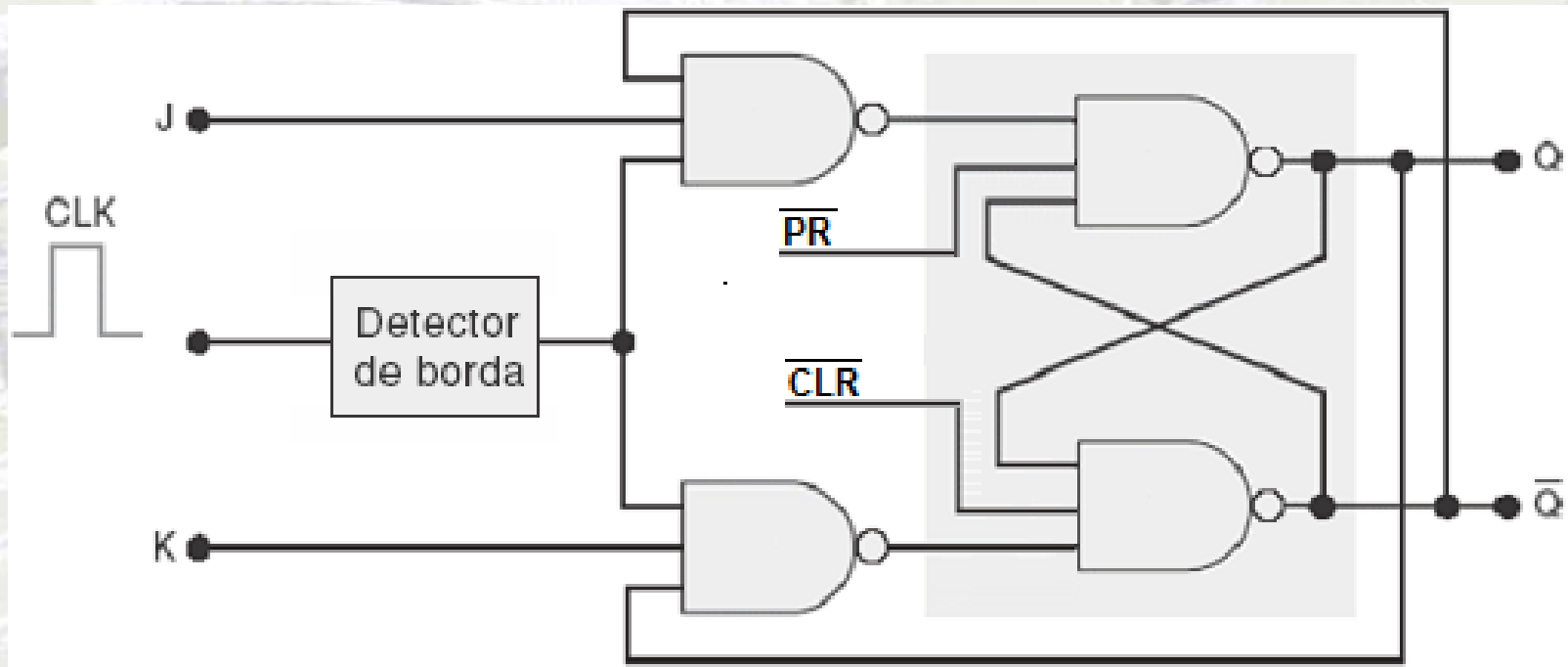
As entradas **assíncronas** *PRESET* e *CLEAR* permitem, mesmo que o CLK seja igual a zero, impor a saída Q igual a 1 ou 0, respectivamente.



J	K	CLK	$\overline{\text{PRE}}$	$\overline{\text{CLR}}$	Q
0	0	↓	1	1	Q (não muda)
0	1	↓	1	1	0 (reset síncrono)
1	0	↓	1	1	1 (set síncrono)
1	1	↓	1	1	$\overline{Q}$ (toggle síncrono ou comutação síncrona)
x	x	x	1	1	Q (não muda)
x	x	x	1	0	0 (clear assíncrono)
x	x	x	0	1	1 (preset assíncrono)
x	x	x	0	0	(Inválido)

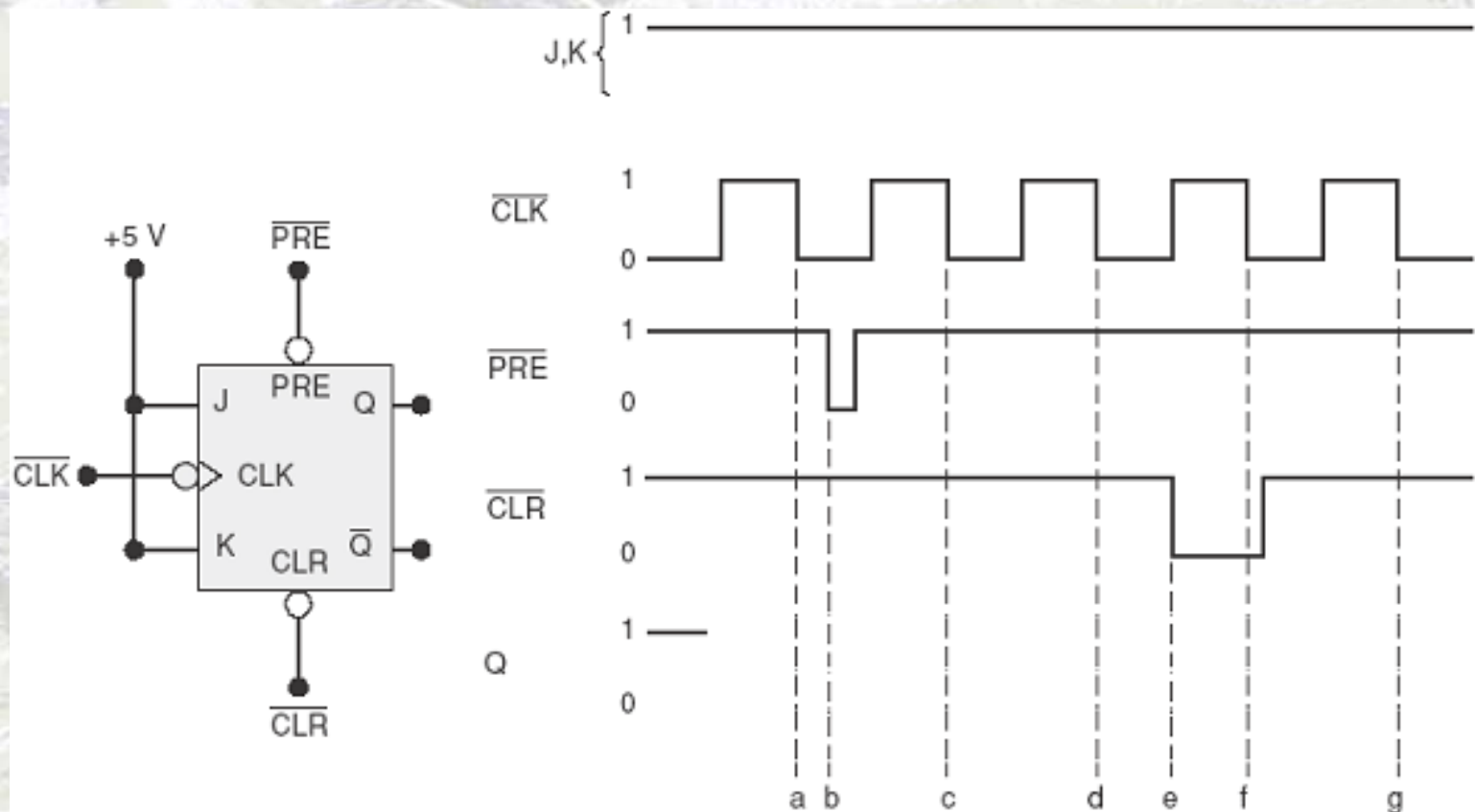
# Flip-Flop JK com Preset / Clear

O circuito lógico do FF JK com entradas assíncronas (*Preset* e *Clear*) é dado por:



# Flip-Flop JK com Preset / Clear

**Exercício 4** – Esboçar a forma de onda na saída Q, para o FF T, com entradas assíncronas, sensível a borda de descida:



# Flip-Flop JK com Preset / Clear

**Exercício 4** – Esboçar a forma de onda na saída Q, para o FF T, com entradas assíncronas, sensível a borda de descida:

