

# Circuitos Lógicos

## Aula 17

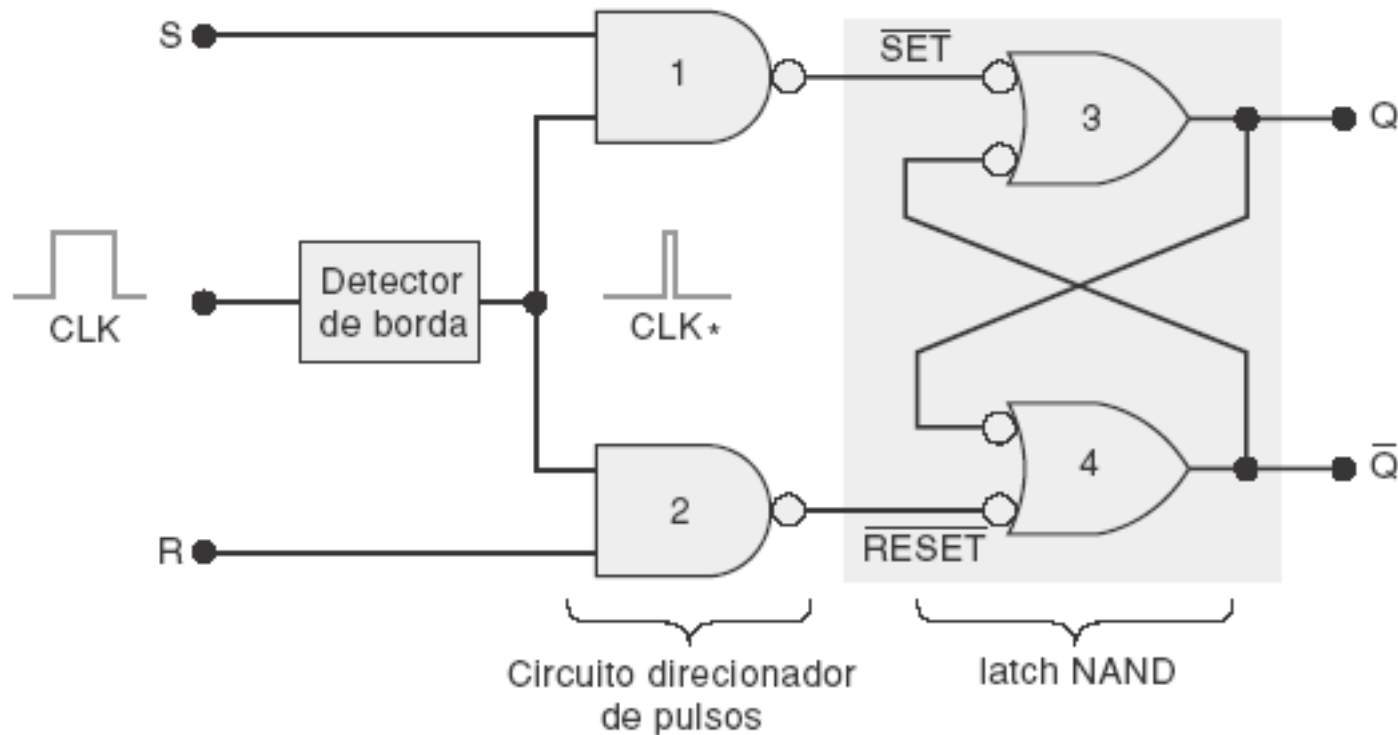
### **Aula passada**

- Circuitos síncronos
- Pulsos, impulsos e *clock*
- Detector de borda
- FF NAND com clock

### **Aula de hoje**

- Revisão FF NAND com clock (S-R)
- Flip-Flop J-K com Clock
- Flip-Flop D com Clock

# Circuito FF NAND (S-R) com *Clock*



## ■ Tabela verdade do Flip-Flop S-R?

Entradas			Saída
S	R	CLK	Q
0	0	↑	$Q_0$ (Não muda)
1	0	↑	1
0	1	↑	0
1	1	↑	Ambíguo

$Q_0$  é o nível de saída anterior a ↑ de CLK.  
↓ de CLK não produz mudança em Q.

# Funcionalidade de *Toggle*

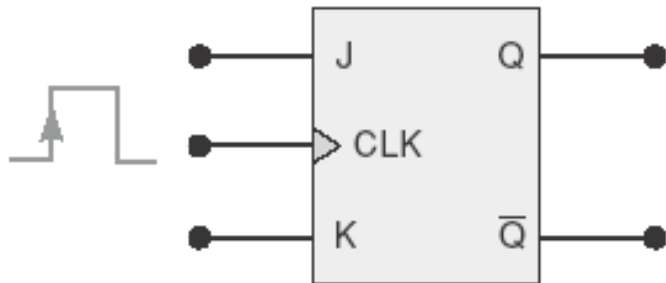
- *Toggle* (trocar): inverter o estado do Flip-Flop

- $Q_0 = 0 \rightarrow Q = 1$  ou  $Q_0 = 1 \rightarrow Q = 0$

**Usaremos uma das entradas para isto!**

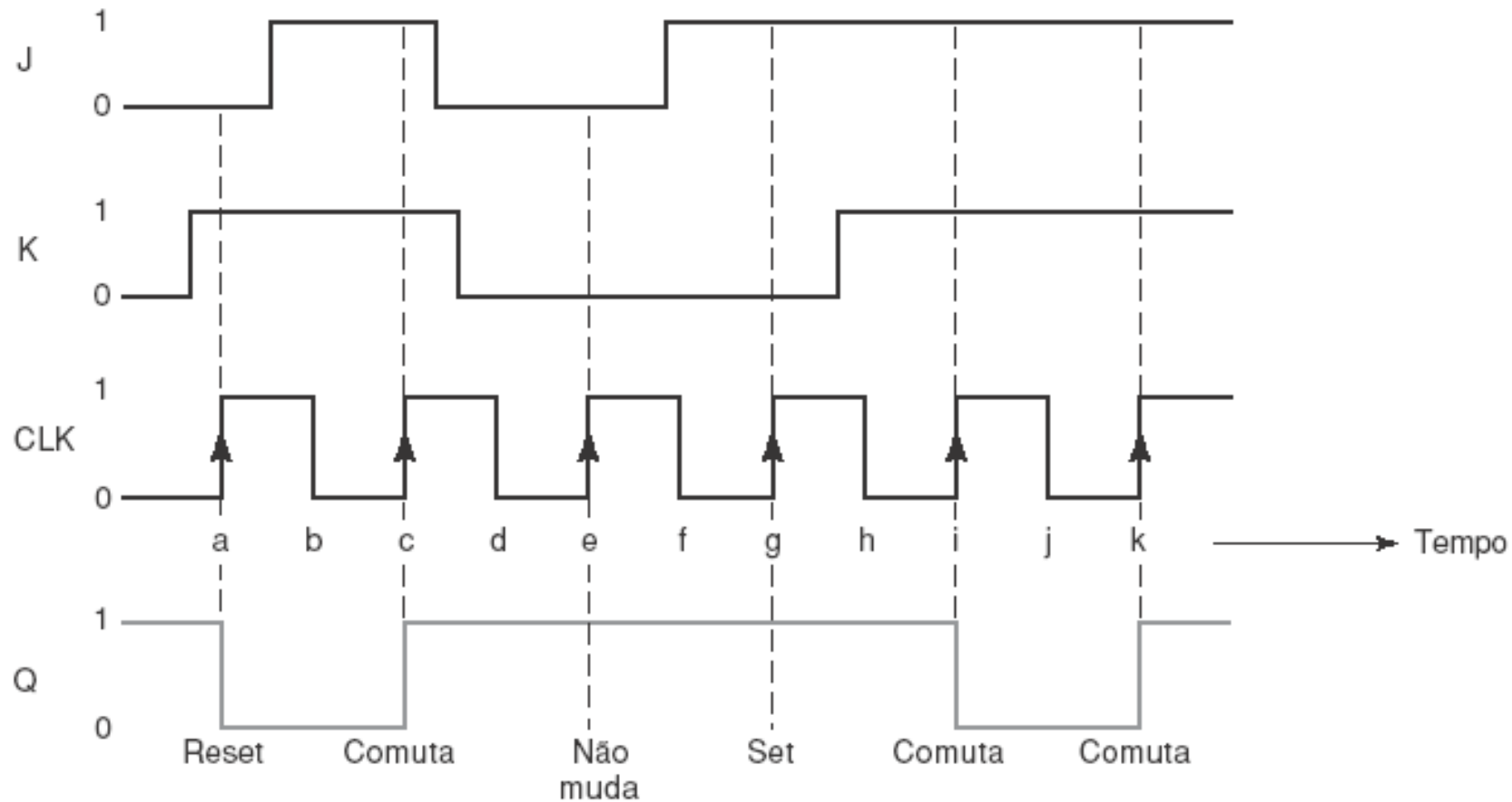
- $S=1, Q=1, \text{Clk} : \textit{Toggle}$
- Também resolve problema de ambiguidade do FF S-R
- Novo Flip-Flop: J-K

# Flip Flop J-K



J	K	CLK	Q
0	0	$\uparrow$	$Q_0$ (não muda)
1	0	$\uparrow$	1
0	1	$\uparrow$	0
1	1	$\uparrow$	$\bar{Q}_0$ (comuta)

(a)



(b)

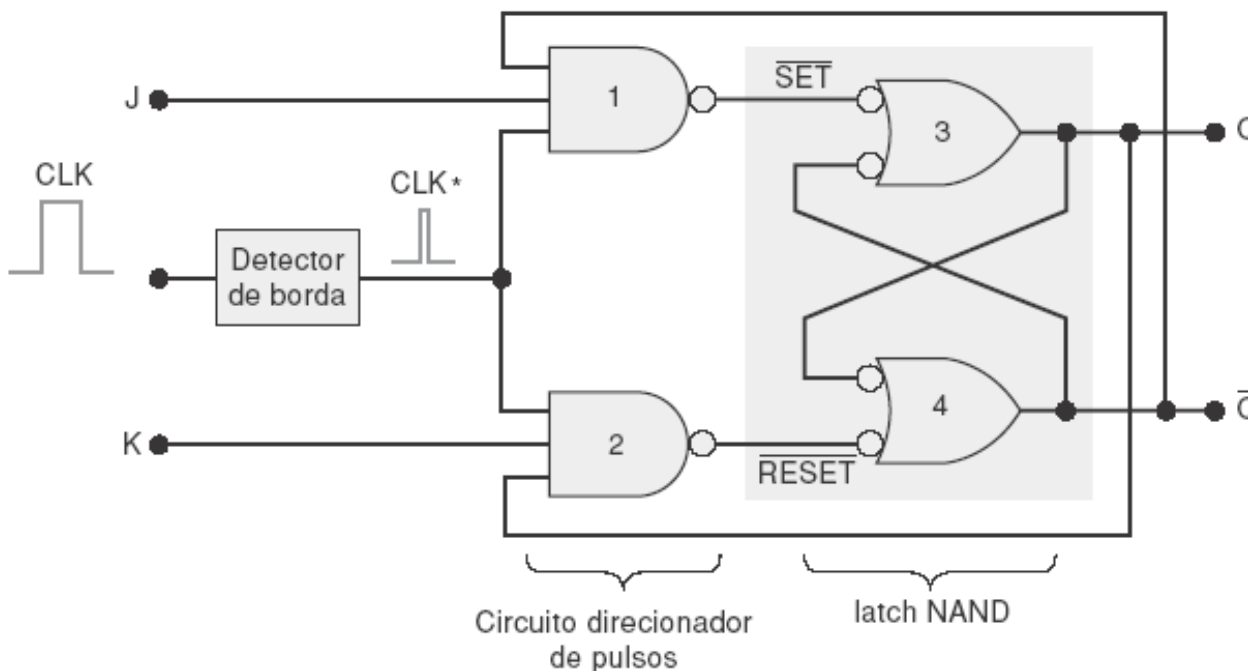
# Implementação do FF J-K



- Como implementar o *toggle*?
- dica: já fizemos um *toggle* em aula!

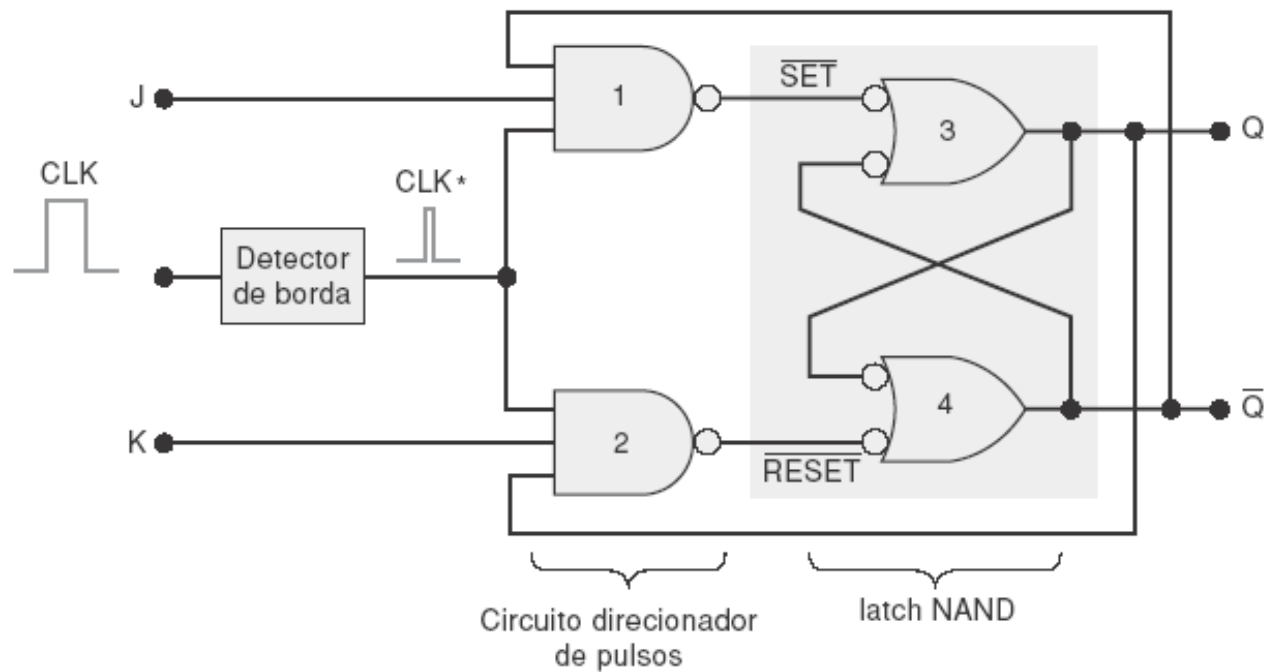
**Feedback do feedback!**

- Fundamento da computação: recursão



- Usar FF S-R *Clocked*
- Feedback do sinal Q, Q' novamente

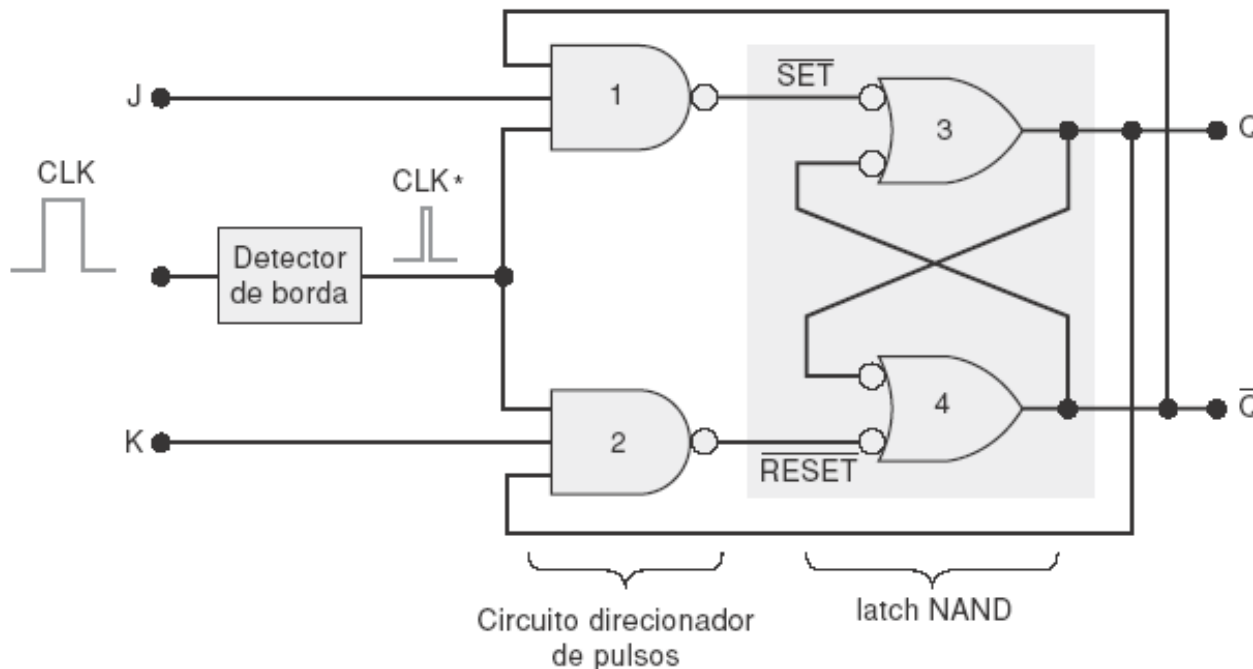
# Implementação do FF J-K



- Tabela verdade?
- Caso  $J=1, K=1$  (toggle)
- Mostrar que outros casos funcionam
  - Set, Reset, mantém

# Detalhes do FF J-K

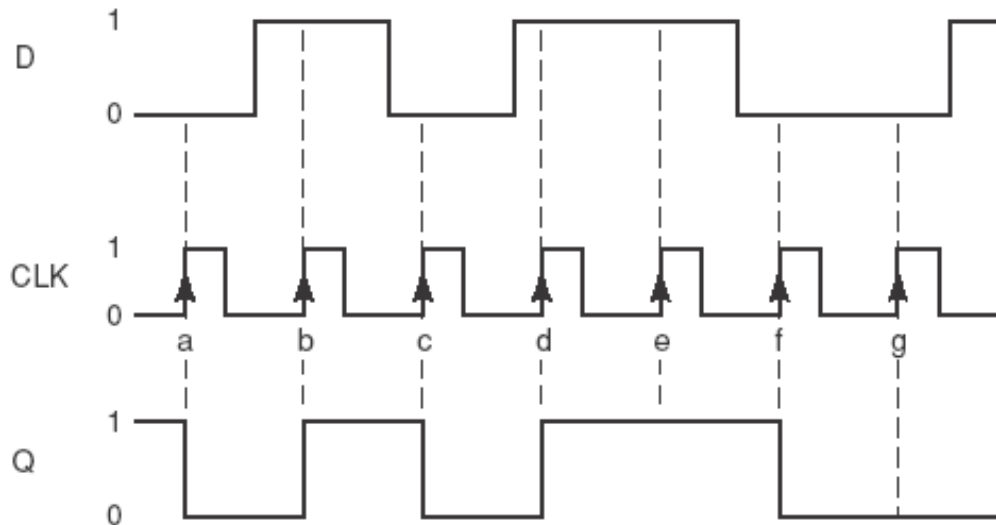
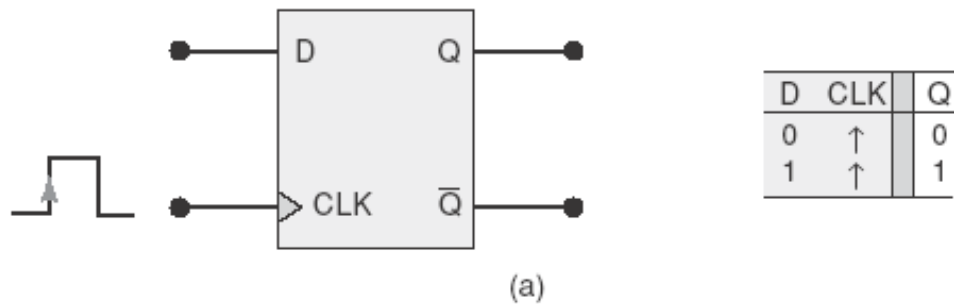
- FF J-K faz tudo o que o FF S-R faz
- Oferece funcionalidade de toggle
- Problema: Como garantir apenas um toggle por clock?



- Garantir impulso bem pequeno!
- Valores de Q e  $Q'$  *depois* do toggle veem  $\text{CLK}^* = 0$

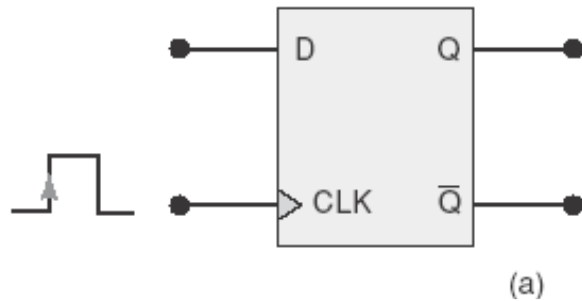
# Flip-Flop D Clocked

- Elemento de memória simples
- Armazena valor do bit de entrada ao sinal clock
  - $D = 0 \rightarrow \text{SET}$ ;  $D = 1 \rightarrow \text{Reset}$





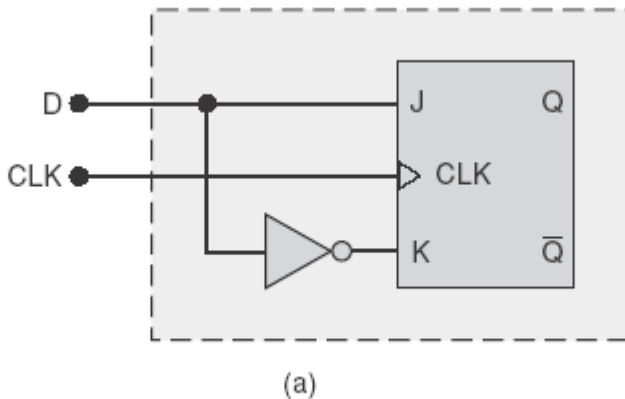
# Implementação do FF D



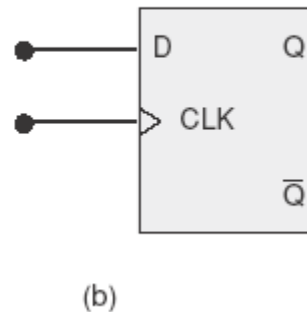
D	CLK	Q
0	↑	0
1	↑	1

- Idéias?
- dica: usar o que já conhecemos

- Usar FF J-K
- Usar D para gerar entrada J
- Usar  $D'$  para gerar entrada K



≡



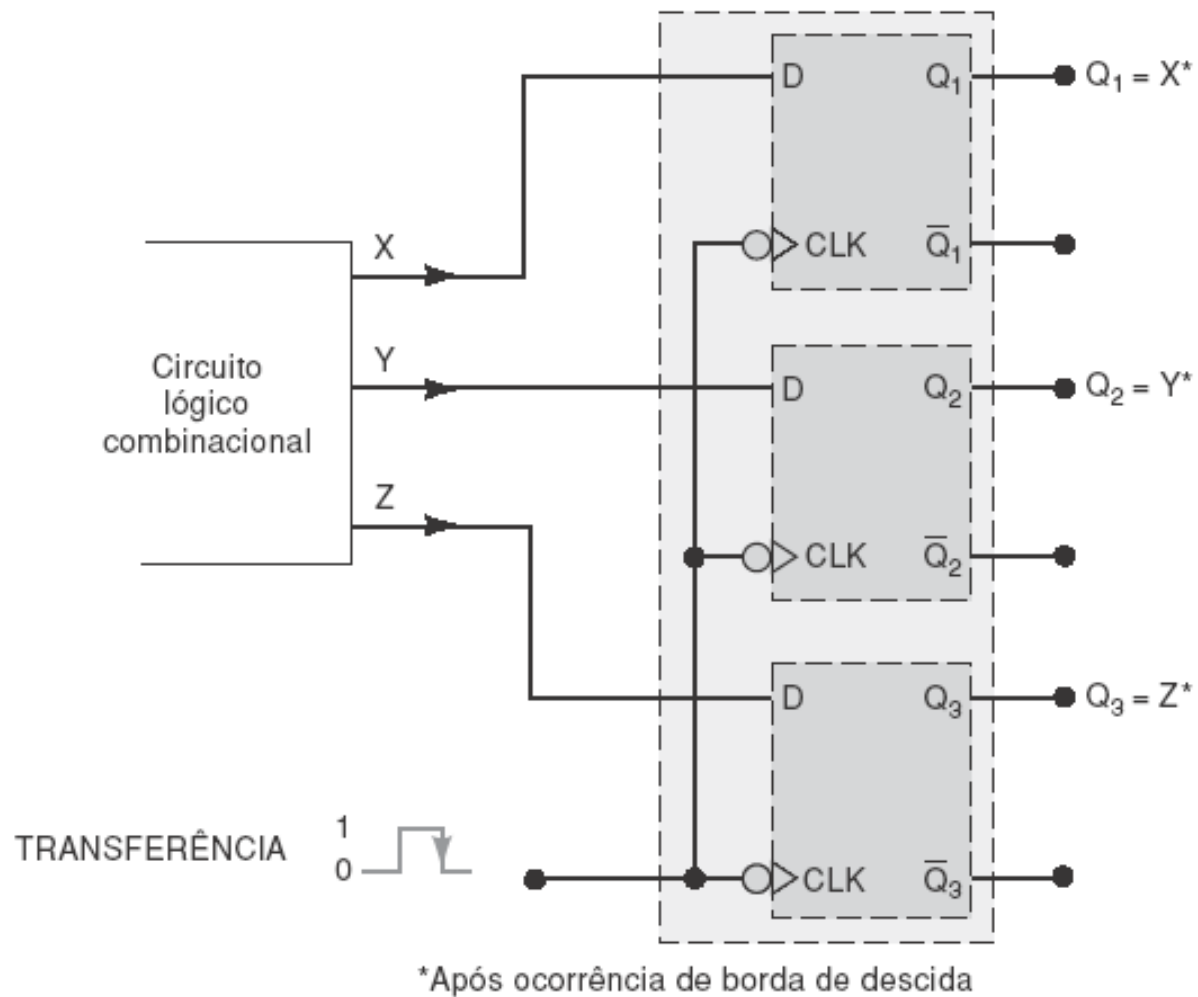
- Tabela verdade?
- $D=0$ ,  $D=1$

# Vantagem e Uso do FF D



- FF D gera na saída seu valor de entrada
- **Inútil?**
- Saída é igual a entrada no instante de clock e permanece o mesmo entre clocks (independente da entrada)
- FF sustenta saída!
- Fornece entrada para próxima fase (módulo) do circuito

# Exemplo FF D



- Transferência paralela de dados
- Circuito combinacional pode trabalhar tranquilo...
- Saída é gravada e transferida para próxima parte