

# Aula 13

## Flip-Flop – Parte 2

**SEL 0414 - Sistemas Digitais**

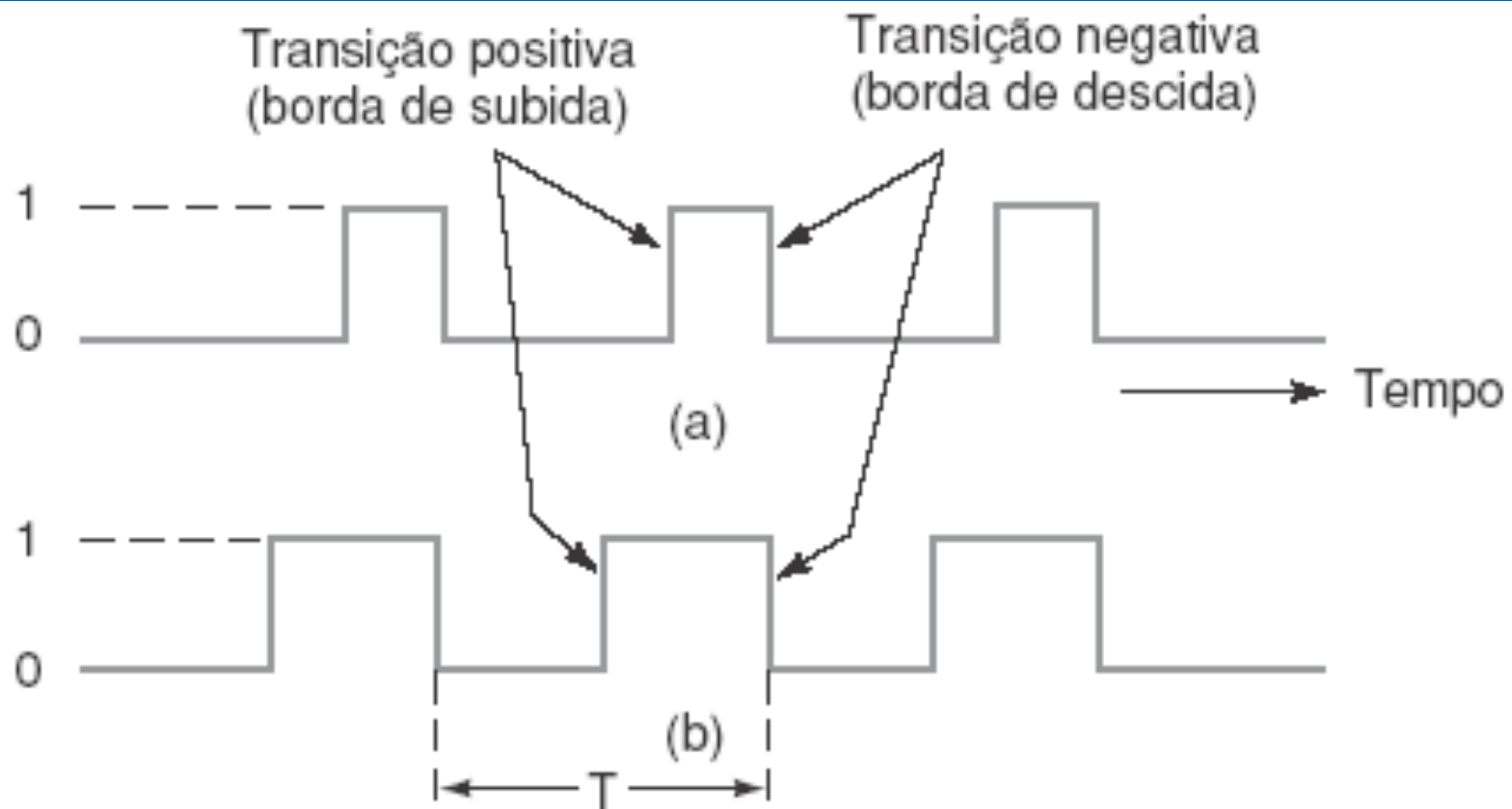
**Prof. Dr. Marcelo Andrade da Costa Vieira**

*Flip-Flops  
sensíveis à borda*

# FF Mestre-Escravo - Obsoleto

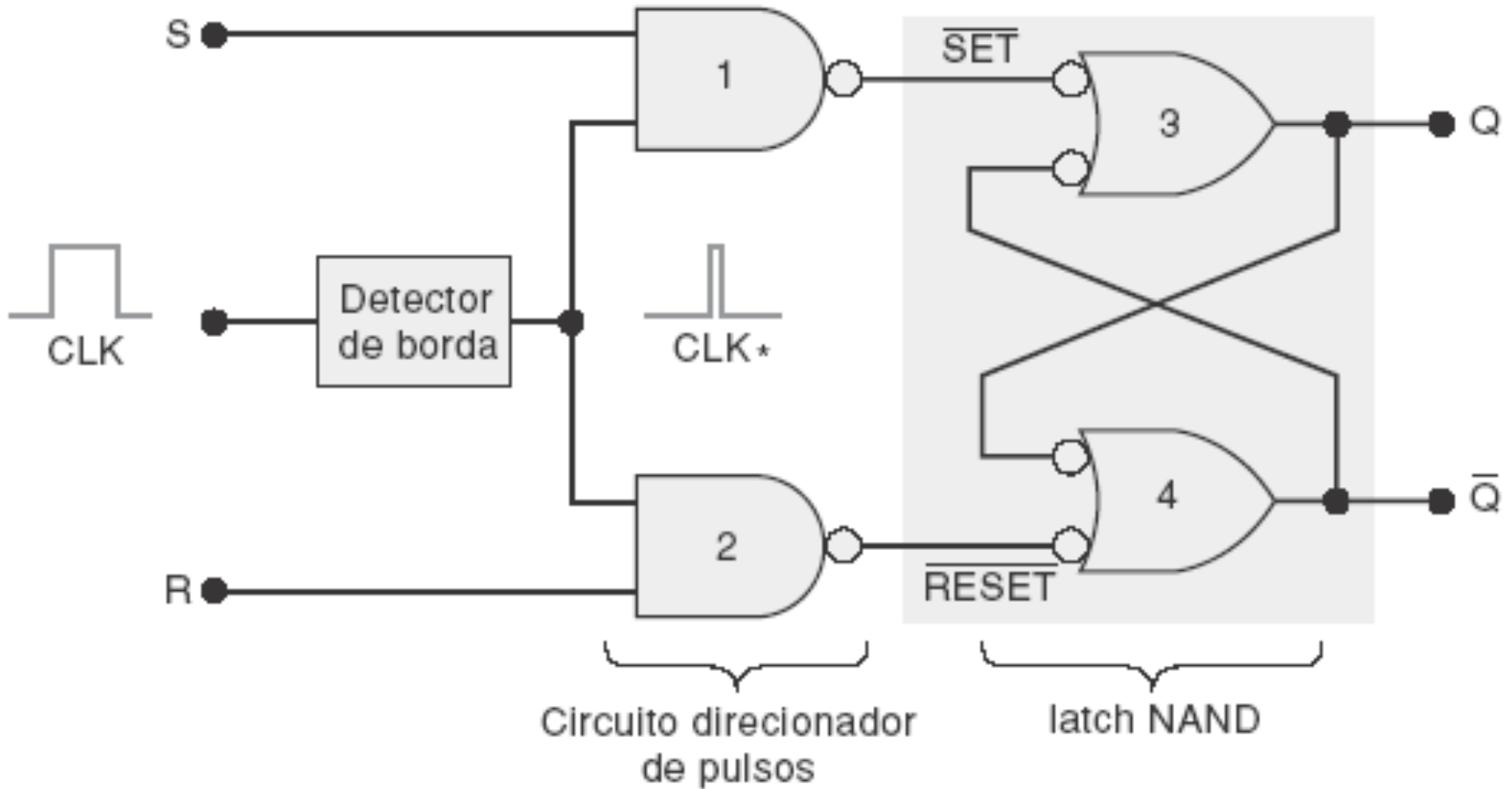
- Apesar de garantir que o estado de saída só se altere na borda de descida do pulso de *clock*, durante o  $\frac{1}{2}$  período positivo do *clock*, o estado do FF mestre pode alterar, resultando em operação imprevisível na saída;
- Assim, para um perfeito sincronismo, é necessário garantir que as estradas sejam mantidas estáveis durante todo o período do clock, alterando apenas na transição.

# Transição por borda



# FF RS Síncrono

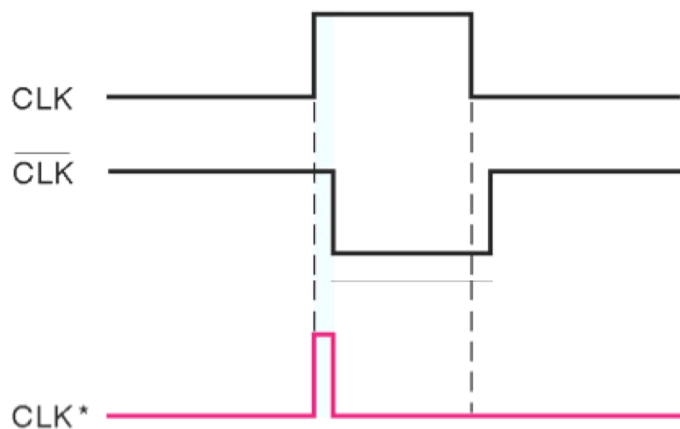
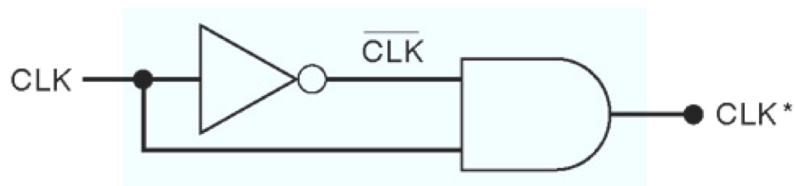
# Sensível à Borda



- Para  $Ck=0$  ou  $1 \Rightarrow Q$  e  $\bar{Q}$  não “sentirão” eventuais variações nas entradas
- Para  $Ck = \uparrow$  ou  $\downarrow \Rightarrow$  portas de entrada habilitadas por alguns nanosegundos.

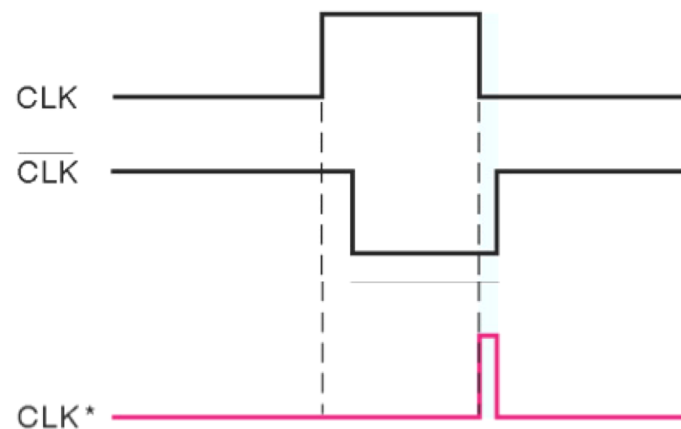
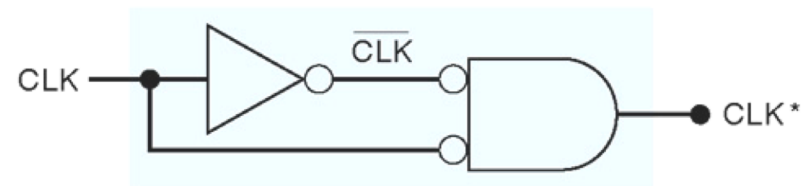
# Detector de Borda

## Borda de Subida



(a)

## Borda de Descida

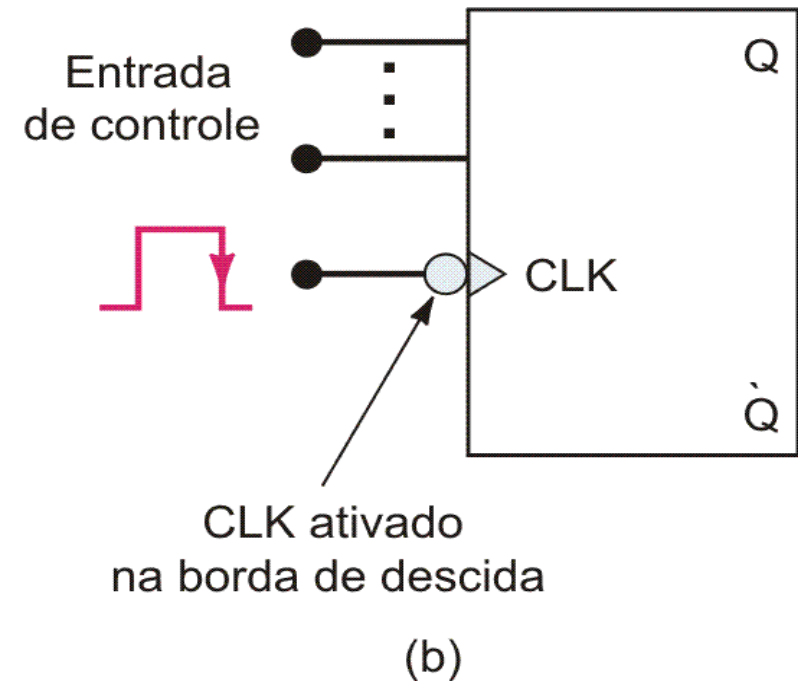
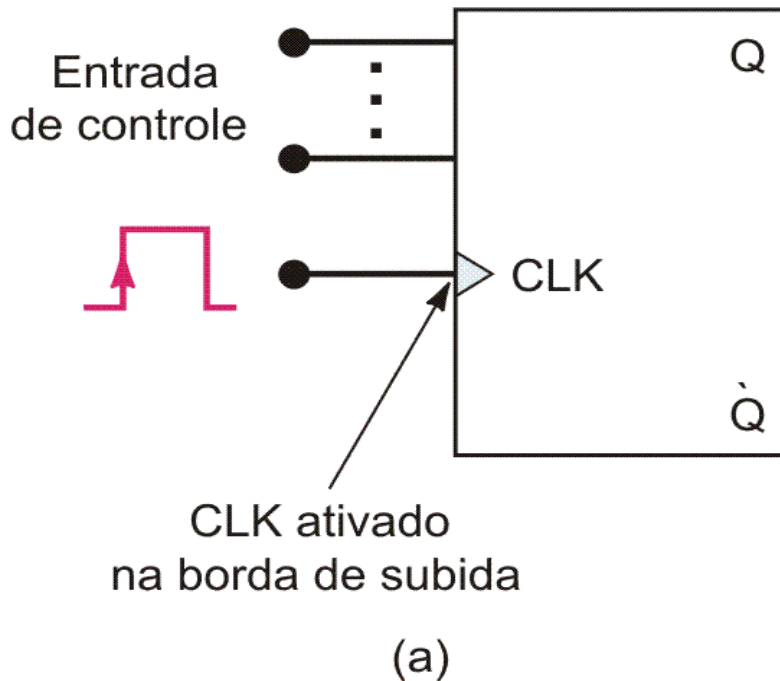


(b)

A duração dos pulsos  $\text{CLK}^*$  é normalmente de 2 a 5 ns

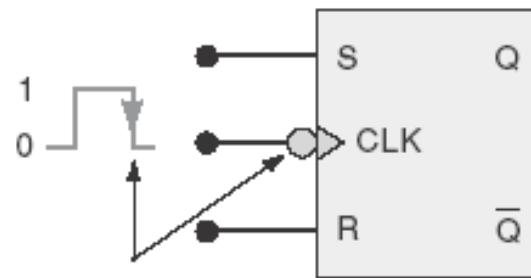
# Latch RS Síncrono

# Sensível à Borda



- Para  $Ck=0$  ou  $1 \Rightarrow Q$  e  $\overline{Q}$  não “sentirão” eventuais variações nas entradas
- Para  $Ck = \uparrow$  ou  $\downarrow \Rightarrow$  funcionamento normal (portas de entrada habilitadas)

# Latch RS Síncrono      Sensível à Borda



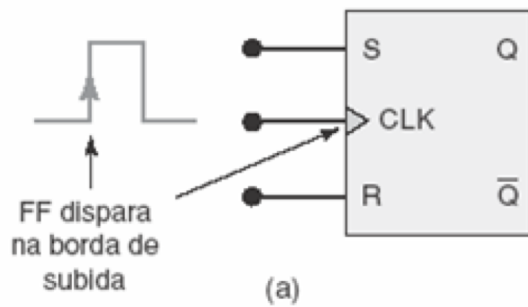
Dispara na  
borda de descida

Entradas			Saída
S	R	CLK	Q
0	0	↓	$Q_0$ (não muda)
1	0	↓	1
0	1	↓	0
1	1	↓	Ambíguo

**FIGURA 5.20**

Flip-flop *S-R* com clock disparado apenas nas bordas de descida do clock.

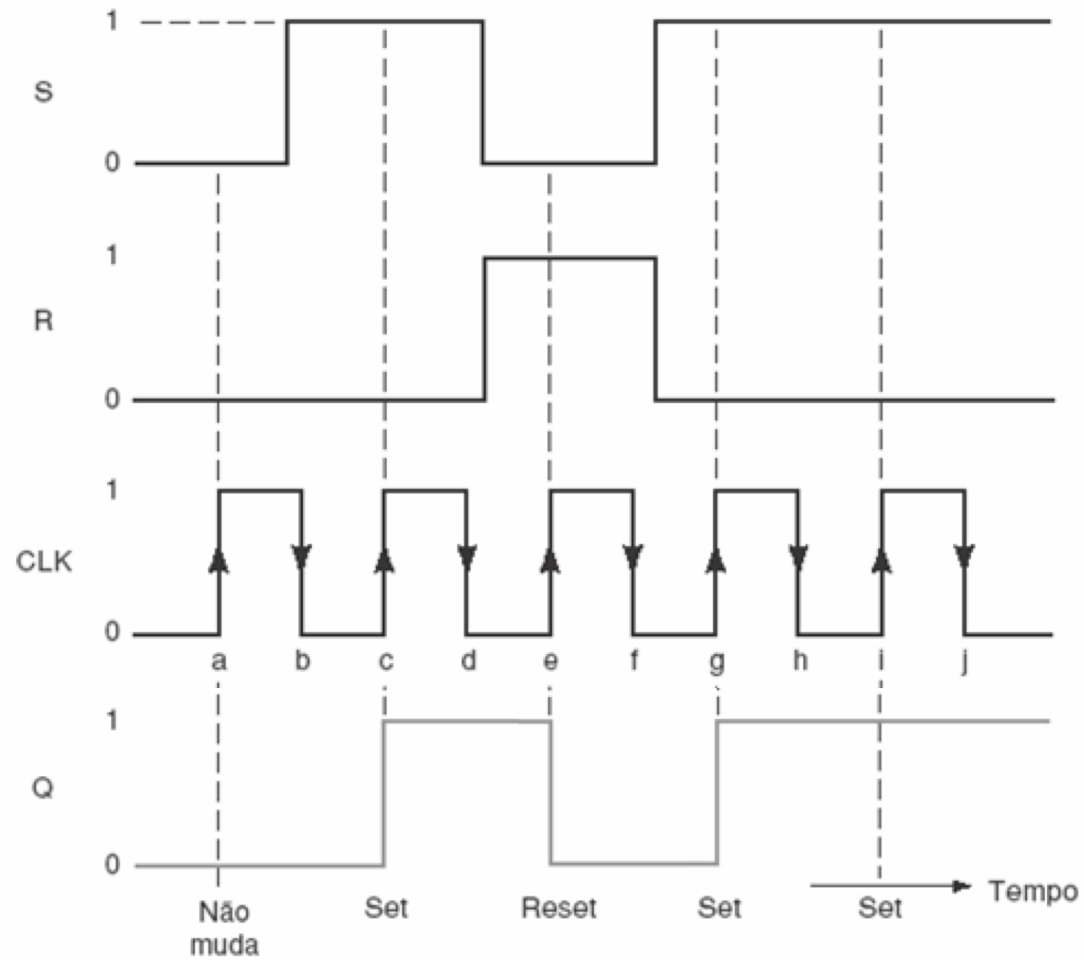




Entradas			Saída
S	R	CLK	Q
0	0	↑	$Q_0$ (Não muda)
1	0	↑	1
0	1	↑	0
1	1	↑	Ambíguo

$Q_0$  é o nível de saída anterior a  $\uparrow$  de CLK.  
 $\downarrow$  de CLK não produz mudança em Q.

(b)

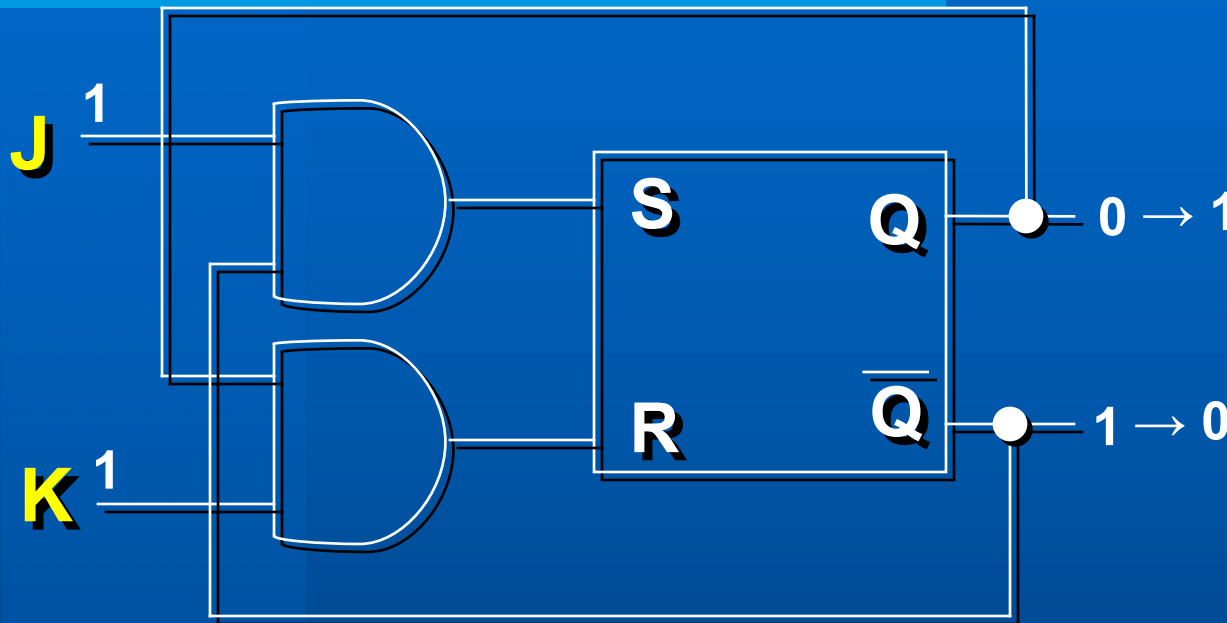


(c)

# ***FLIP-FLOP JK***

# Flip-Flop JK

Condição Inicial →  $Q = 0$

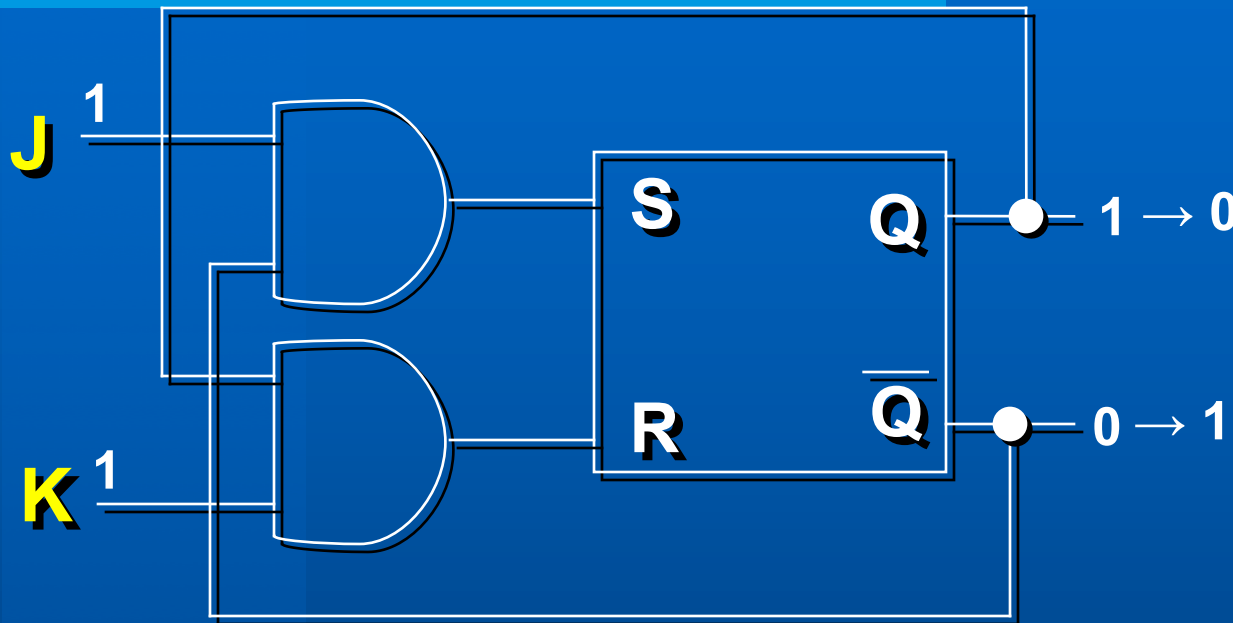


J	K	Q
0	0	$Q_0$
0	1	0
1	0	1
1	1	1

Resolver o problema do “estado ambíguo” quando as duas entradas são iguais a 1

# Flip-Flop JK

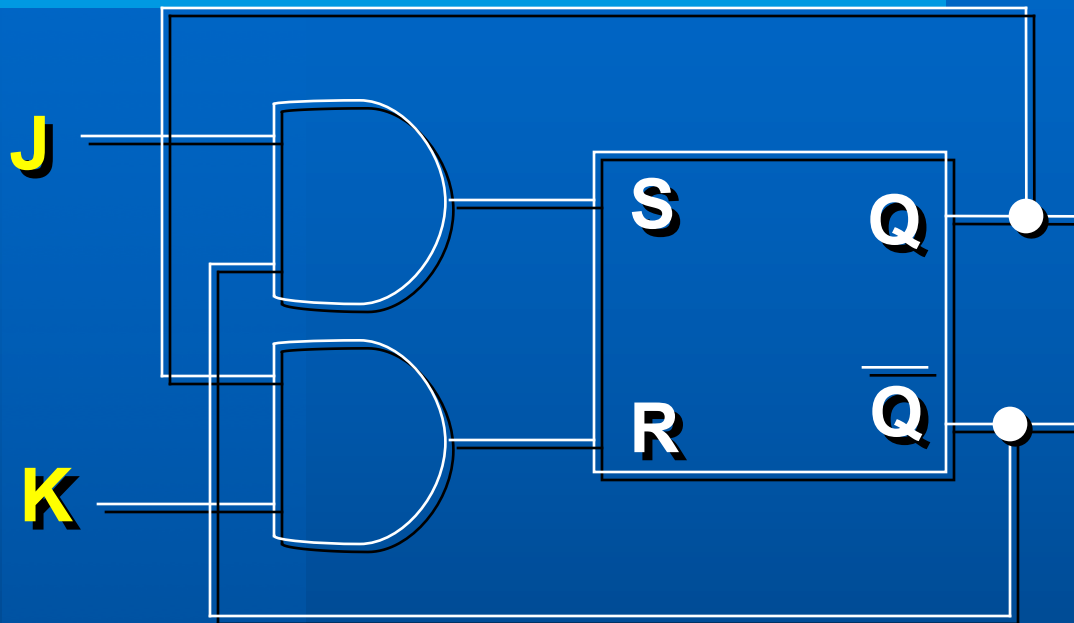
Condição Inicial → Q = 1



J	K	Q
0	0	$Q_0$
0	1	0
1	0	1
1	1	0

Resolver o problema do “estado ambíguo” quando as duas entradas são iguais a 1

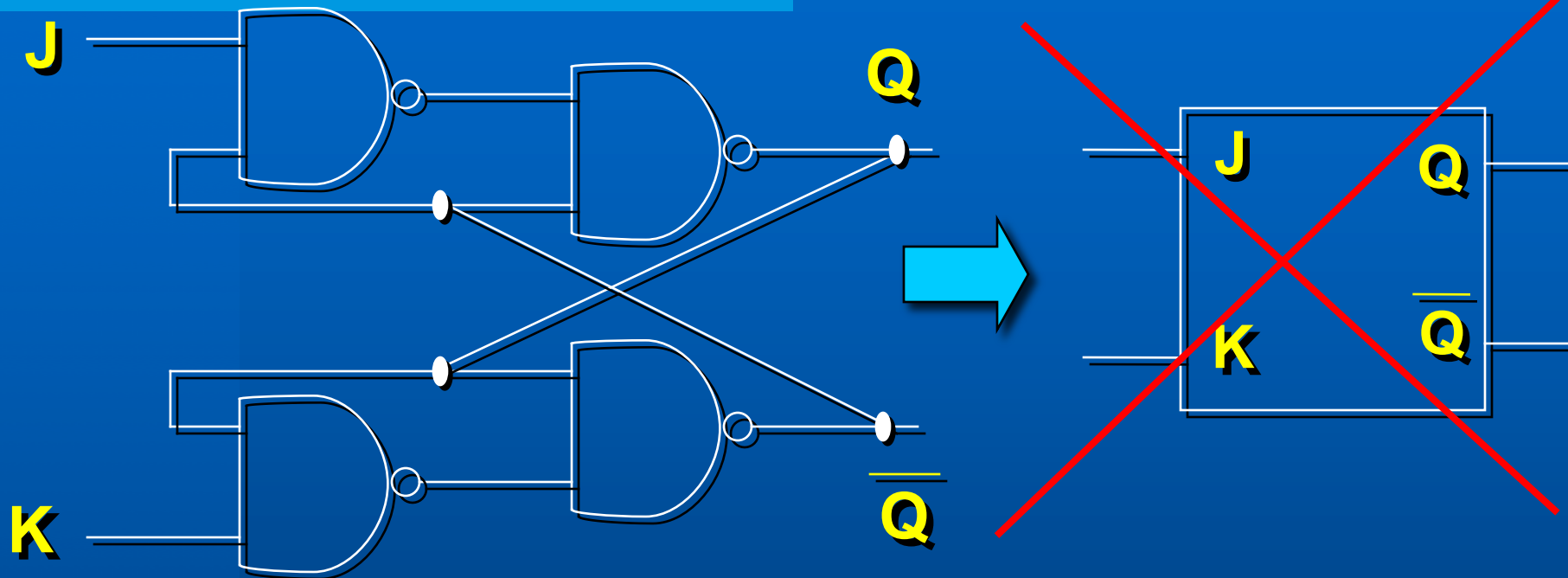
# Flip-Flop JK



J	K	Q
0	0	$Q_0$
0	1	0
1	0	1
1	1	$\overline{Q_0}$

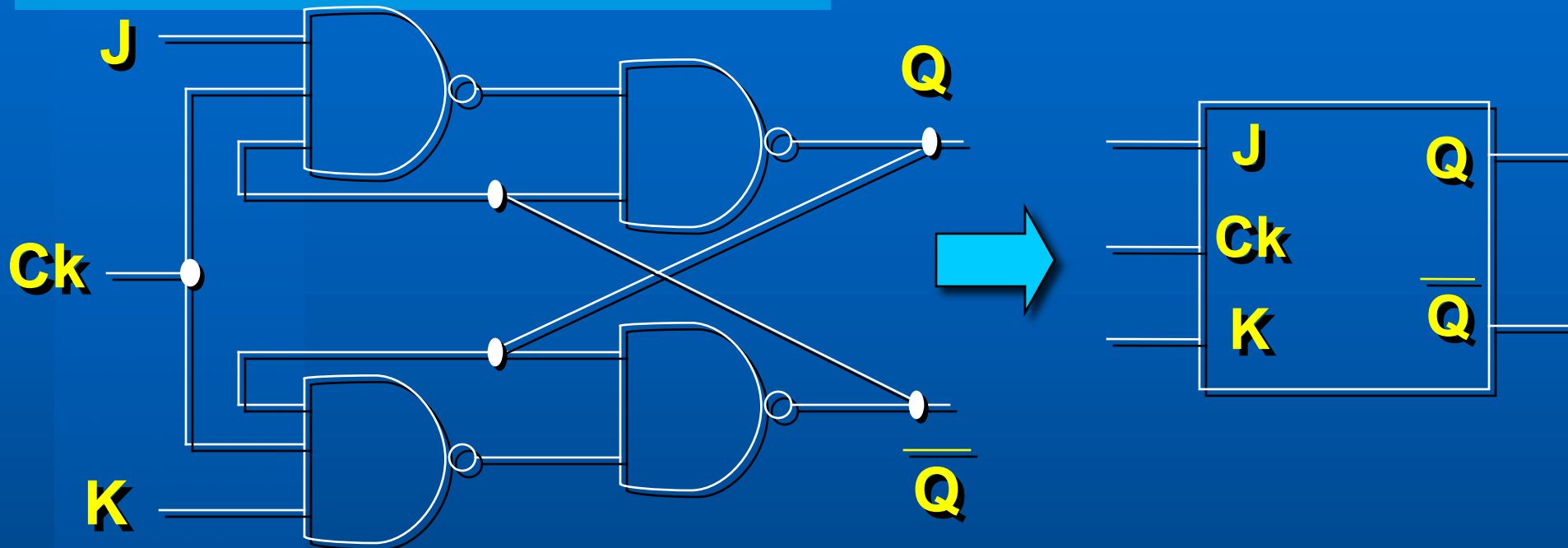
Comutação – “Toggle”  
Inverte o estado anterior

## Circuito básico - Assíncrono



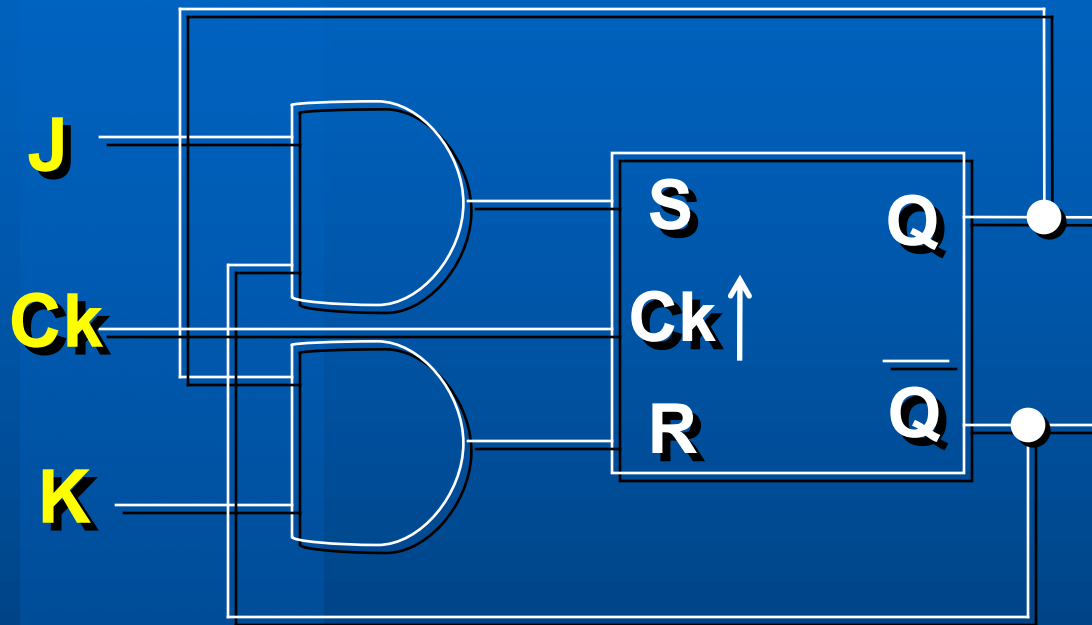
**Problema do JK comum:** Se não houver um sistema de sincronismo, no modo “toggle” ( $J = K = 1$ ) os estados ficam invertendo a cada instante e o circuito fica instável.

## FF JK síncrono – Sensível à Nível



**Problema do JK sensível à nível:** quando o  $Ck=1$ , há passagem das entradas e realimentações; se, nesse instante, houver mudança de J e/ou K, haverá nova saída ➔ comutação para outro estado mais de uma vez durante o mesmo pulso de Ck

# Flip-Flop JK sensível à borda de subida

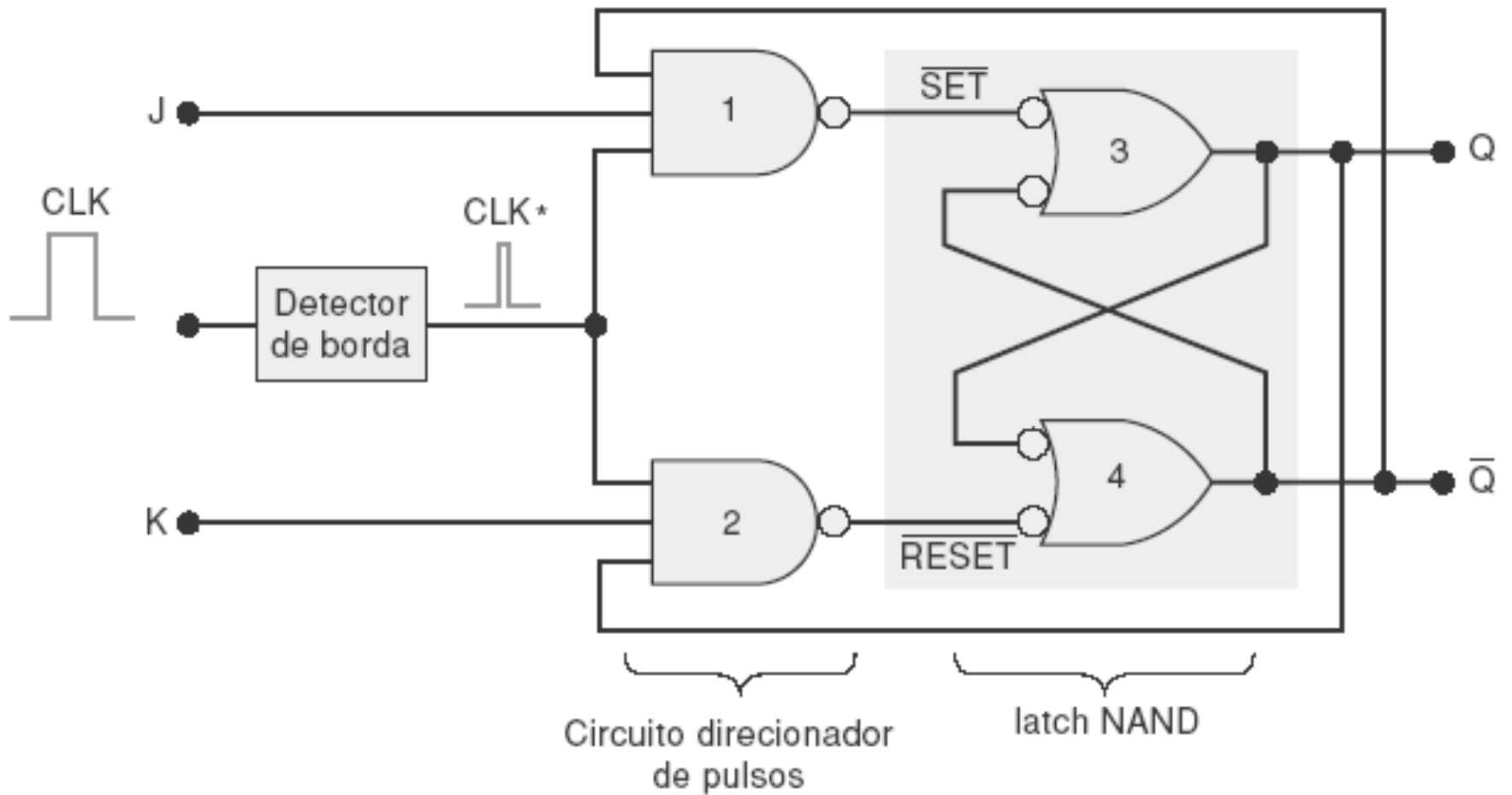


J	K	$Q^*$
0	0	$Q_0$
0	1	0
1	0	1
1	1	$\overline{Q_0}$

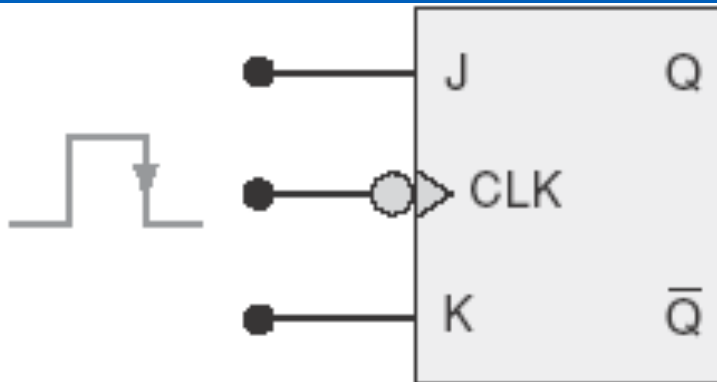
\* Na transição da borda do clock



# Circuito interno de um FF JK sensível à borda

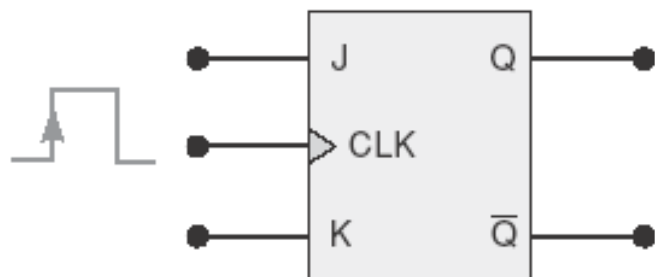


# Esquema Geral de um FF JK disparado por borda de descida:



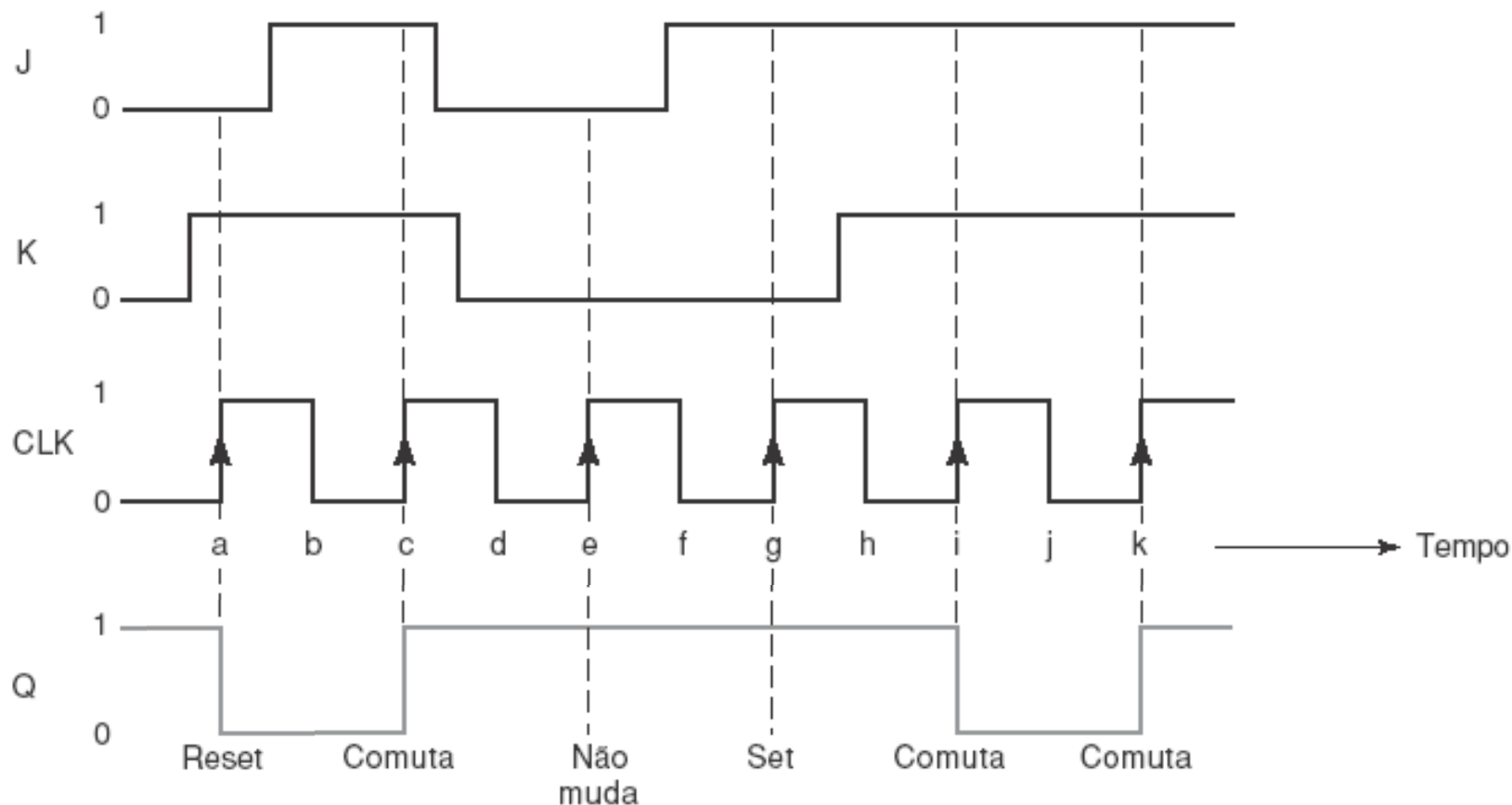
J	K	CLK	Q
0	0	↓	$Q_0$ (não muda)
1	0	↓	1
0	1	↓	0
1	1	↓	$\bar{Q}_0$ (comuta)

# Flip-Flop JK sensível à borda

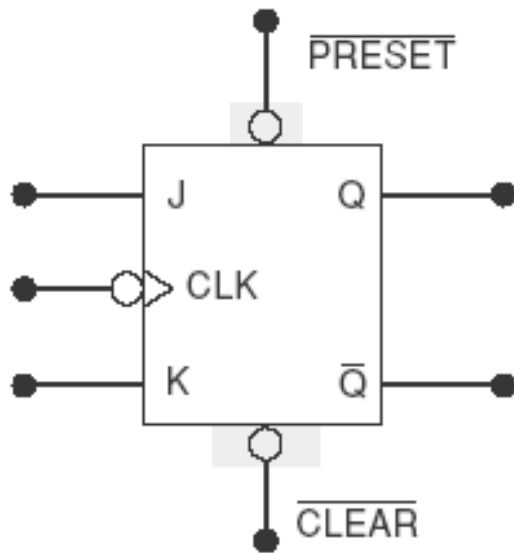


J	K	CLK	Q
0	0	↑	$Q_0$ (não muda)
1	0	↑	1
0	1	↑	0
1	1	↑	$\overline{Q_0}$ (comuta)

(a)

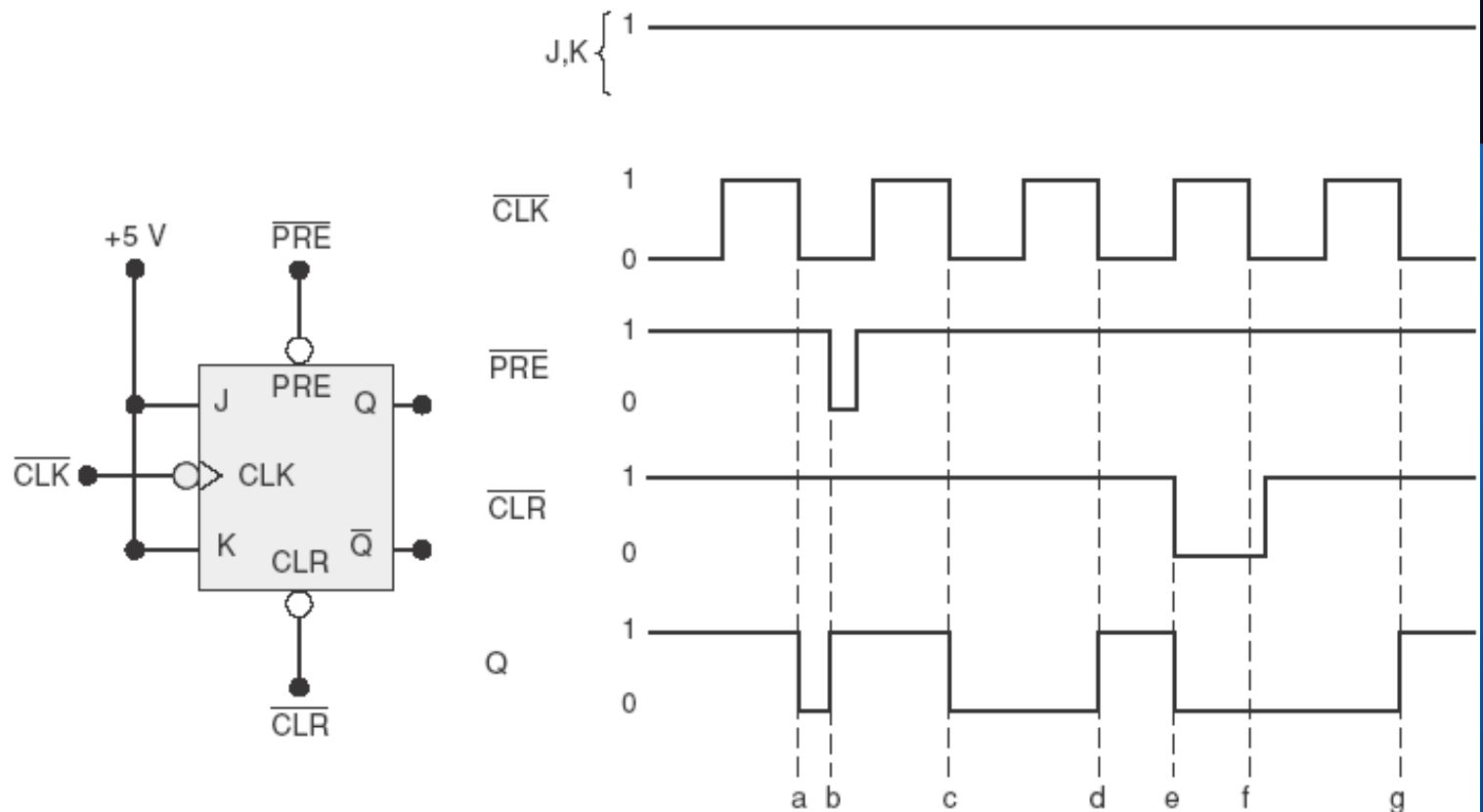


# FF JK síncrono com entradas Assíncronas *Preset e Clear*



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)

# FF Tipo T com entradas Assíncronas

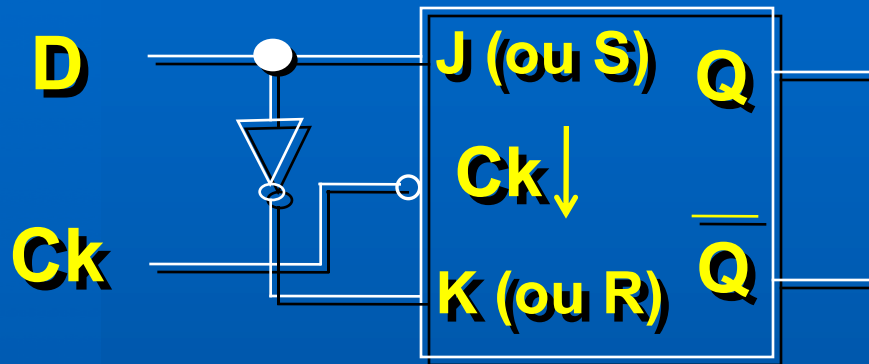


Ponto	Operação
a	Comutação síncrona na borda de descida em $\overline{\text{CLK}}$
b	Set assíncrono em $\overline{\text{PRE}} = 0$
c	Comutação síncrona
d	Comutação síncrona
e	Clear assíncrono em $\overline{\text{CLR}} = 0$
f	$\overline{\text{CLR}}$ se sobrepõe à borda de descida de $\overline{\text{CLK}}$
g	Comutação síncrona

# ***OUTROS TIPOS DE FLIP-FLOPS***

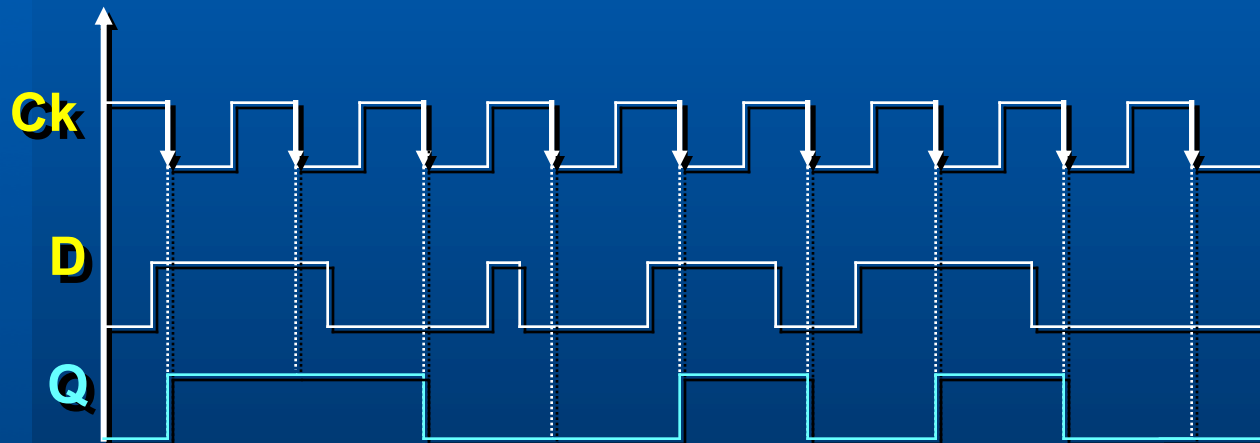
**FF tipo D**

# FF Tipo D ("Data")



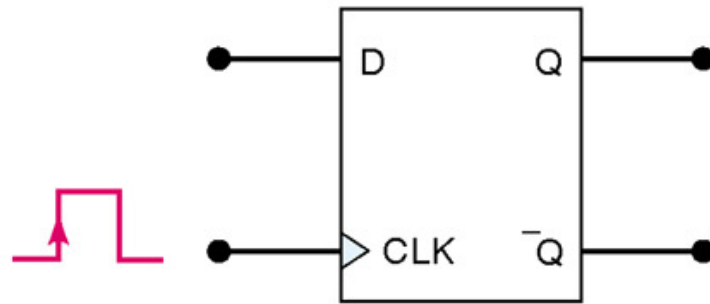
D	Q
0	0
1	1

Ck =  $\uparrow$  ou  $\downarrow$



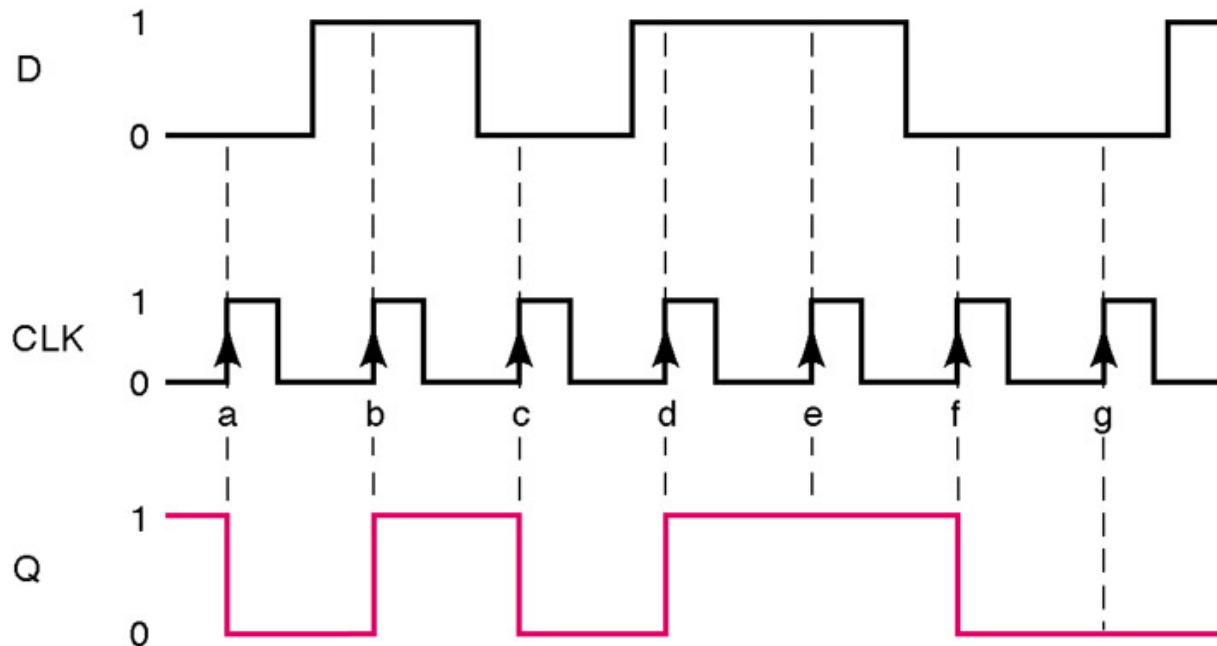


# FF Tipo D sensível à borda de subida



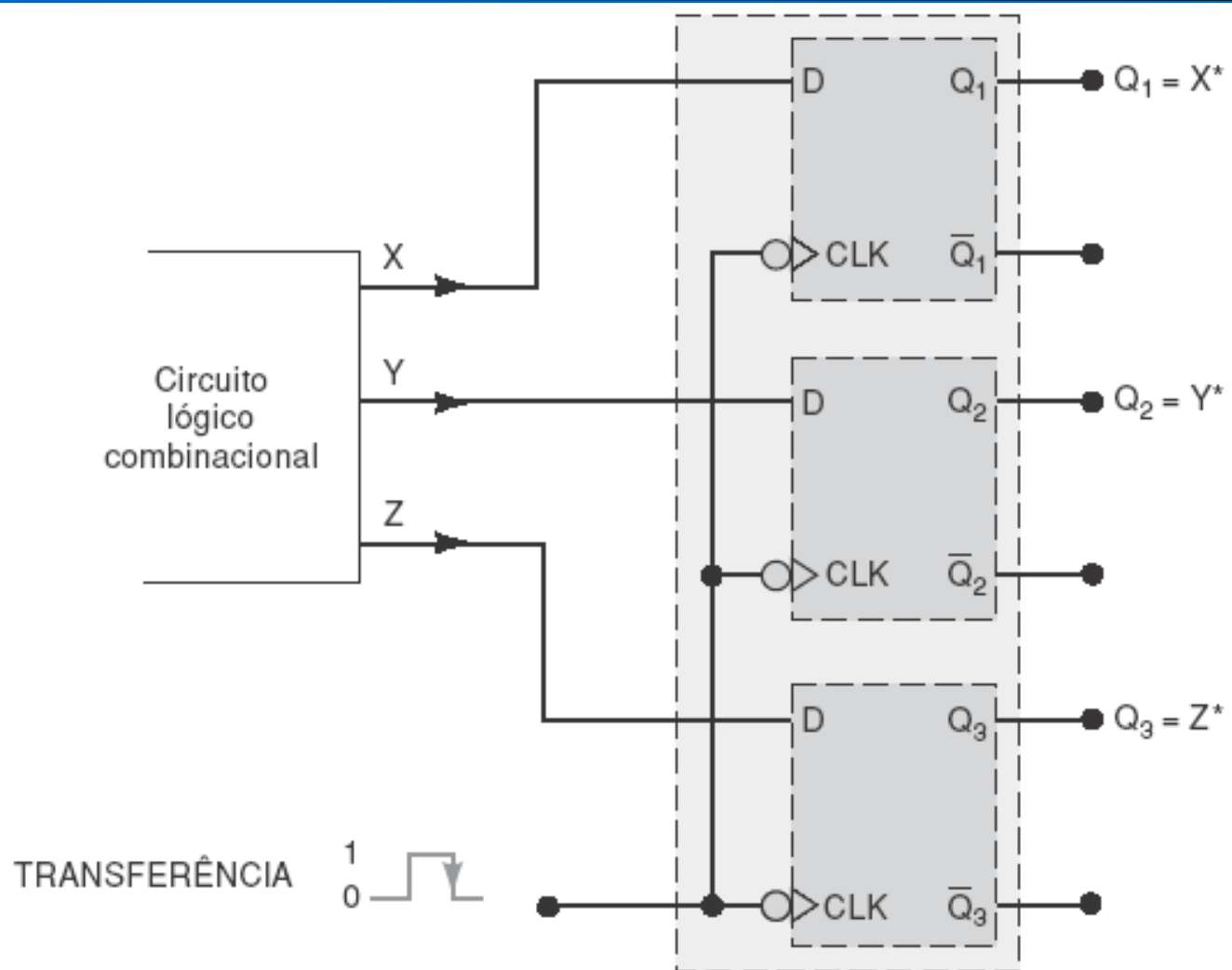
D	CLK	Q
0	↑	0
1	↑	1

(a)



# Aplicações do FF Tipo D:

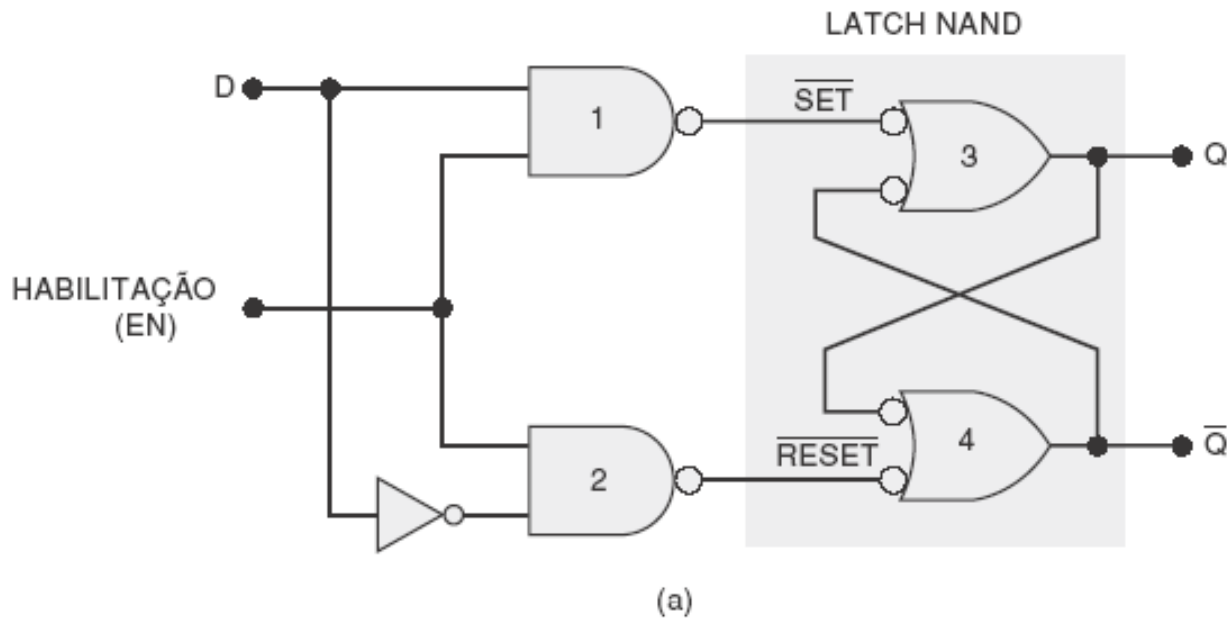
- Transferência de dados em paralelo



\*Após ocorrência de borda de descida

# Latch D “Transparente”

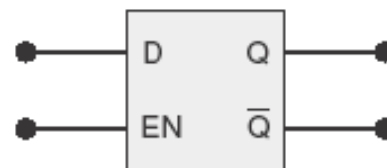
Sensível à Nível



Entradas		Saída
EN	D	Q
0	X	$Q_0$ (não muda)
1	0	0
1	1	1

'X' indica irrelevante.  
 $Q_0$  é o estado imediatamente anterior a EN para o nível BAIXO.

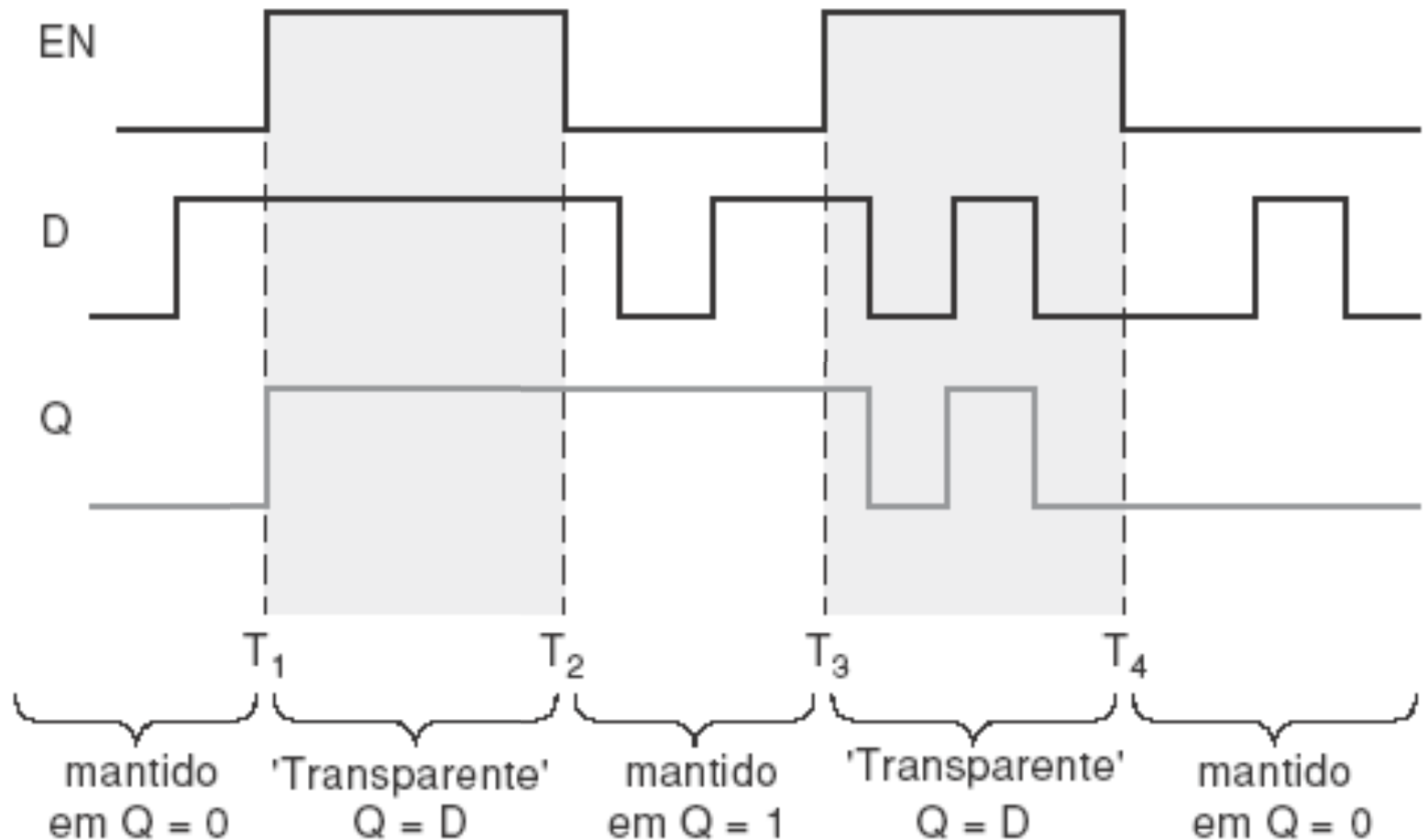
(b)



(c)

# Funcionamento do Latch D “Transparente”

Sensível à nível - mantém o estado anterior até o próximo nível “transparente” - memória



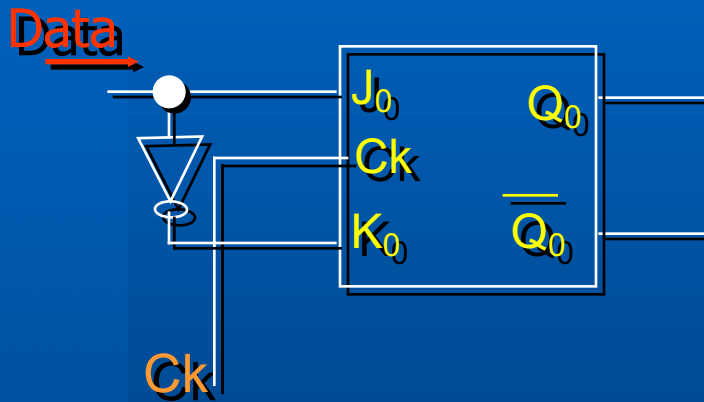
# REGISTRADORES

# REGISTRADORES

## Utilização:

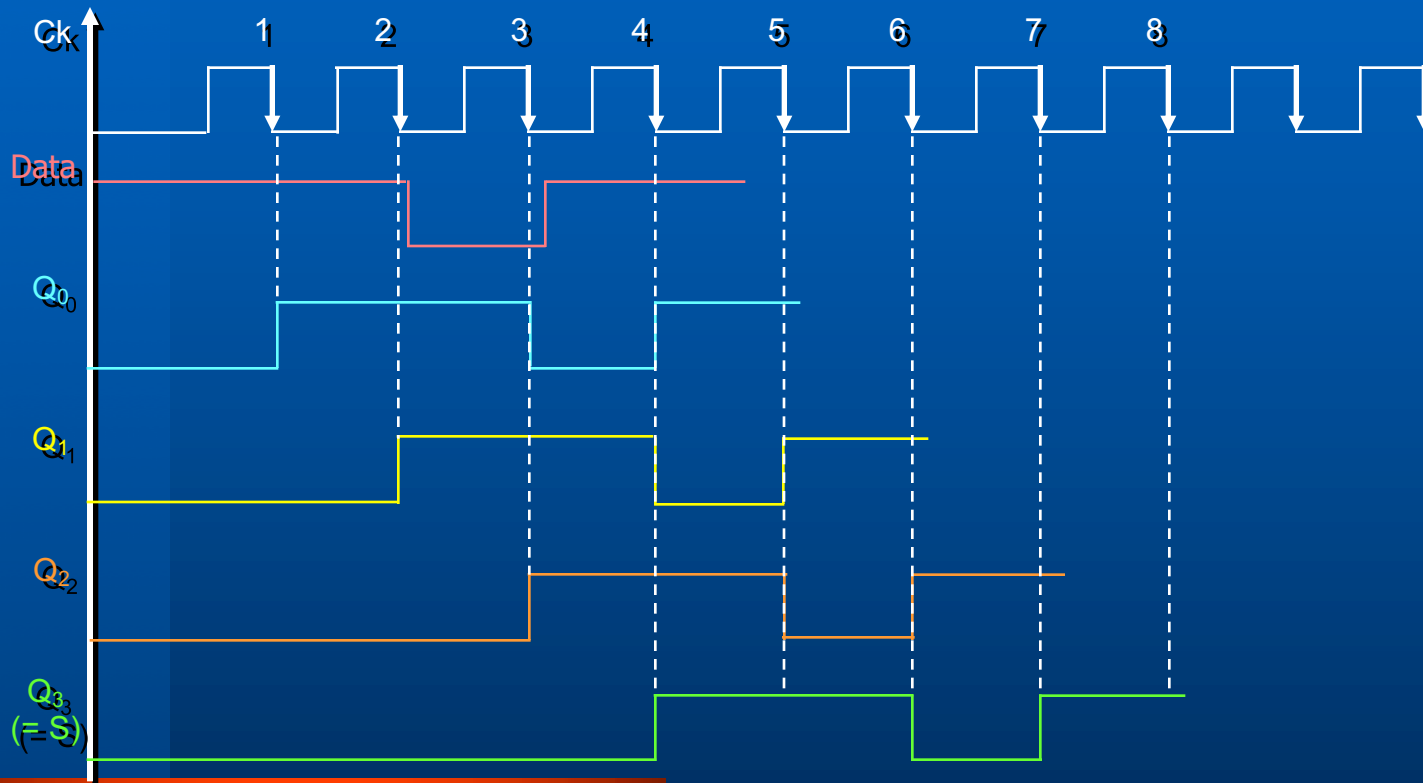
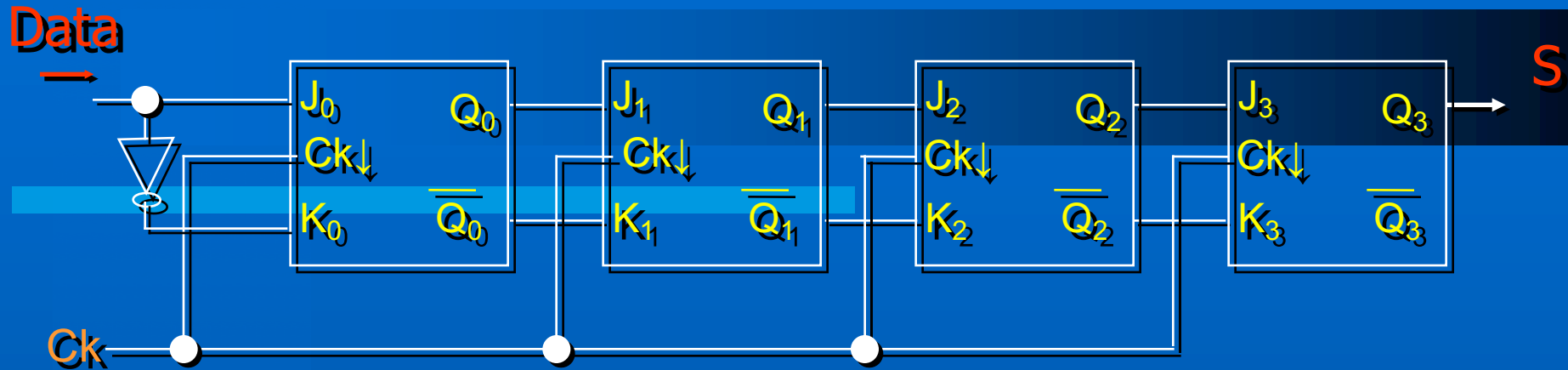
1. Armazenamento de informações com mais de 1 bit (tipo mais simples de MEMÓRIA)
2. Aplicação em:
  - ✓ Conversores (série/paralelo, paralelo/série...)
  - ✓ Contadores, multiplicadores binários
  - ✓ Memórias, computadores, microprocessadores, microcontroladores.

# 1. Transferência de dados



<b>J</b>	<b>K</b>	<b>Q</b>
0	0	$Q_0$
0	1	0
1	0	1
1	1	$\overline{Q_0}$

# Registrador de Deslocamento – *Shift Register*



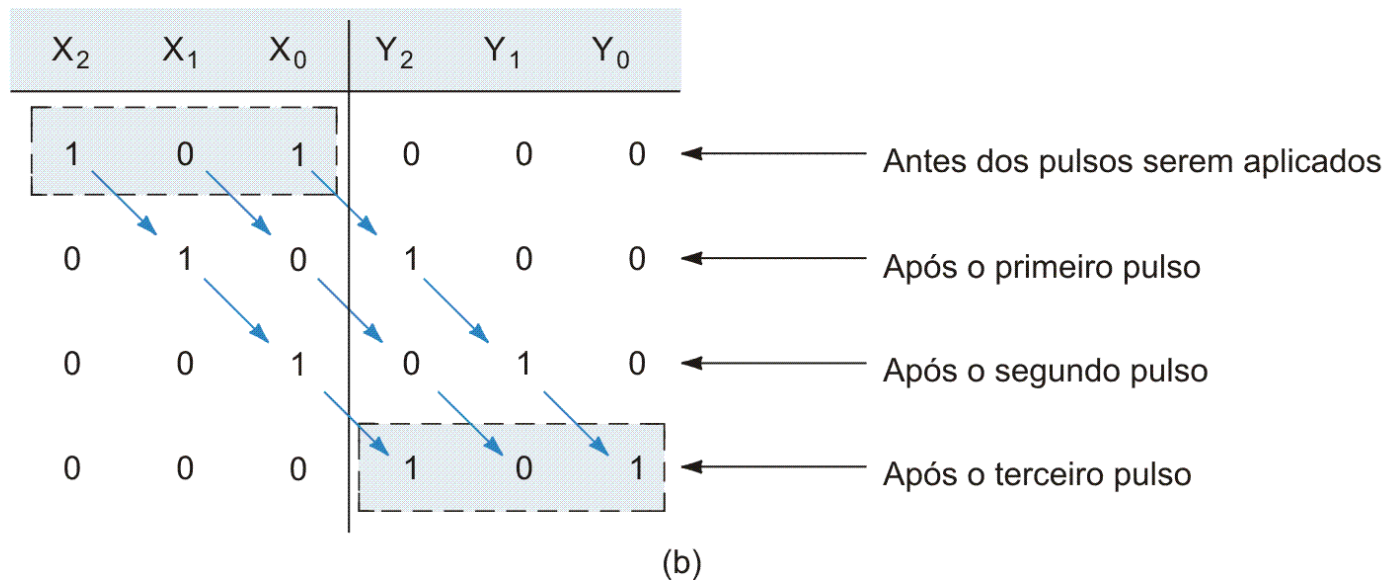
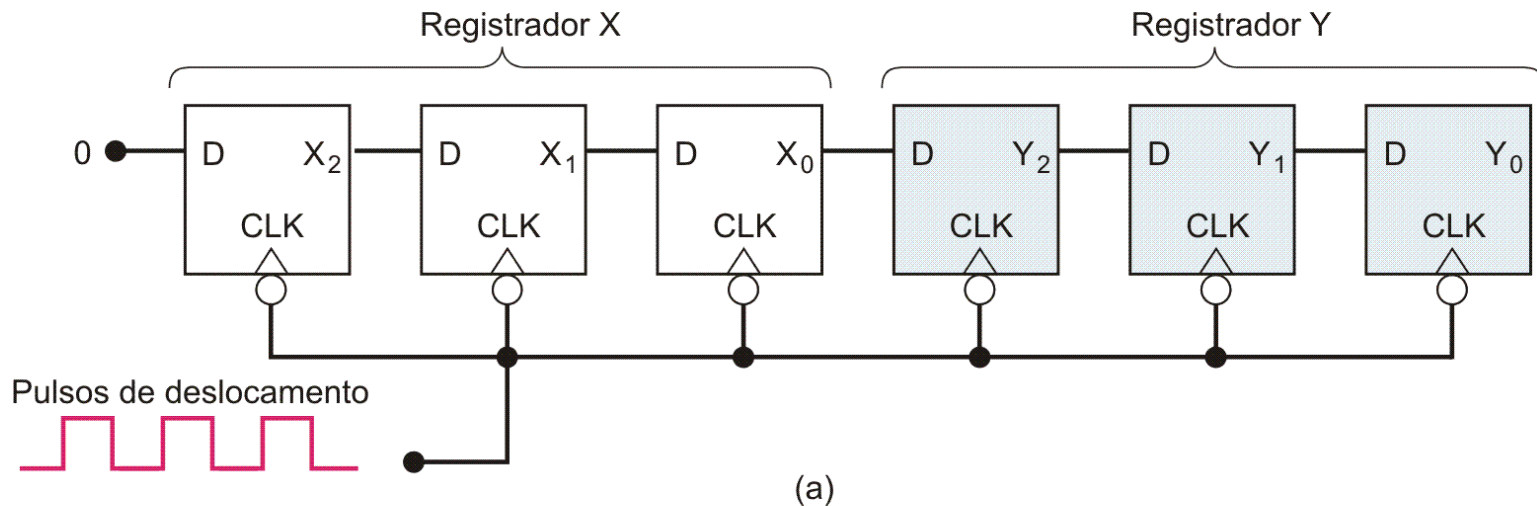


# Registrador de Deslocamento – *Shift Register*

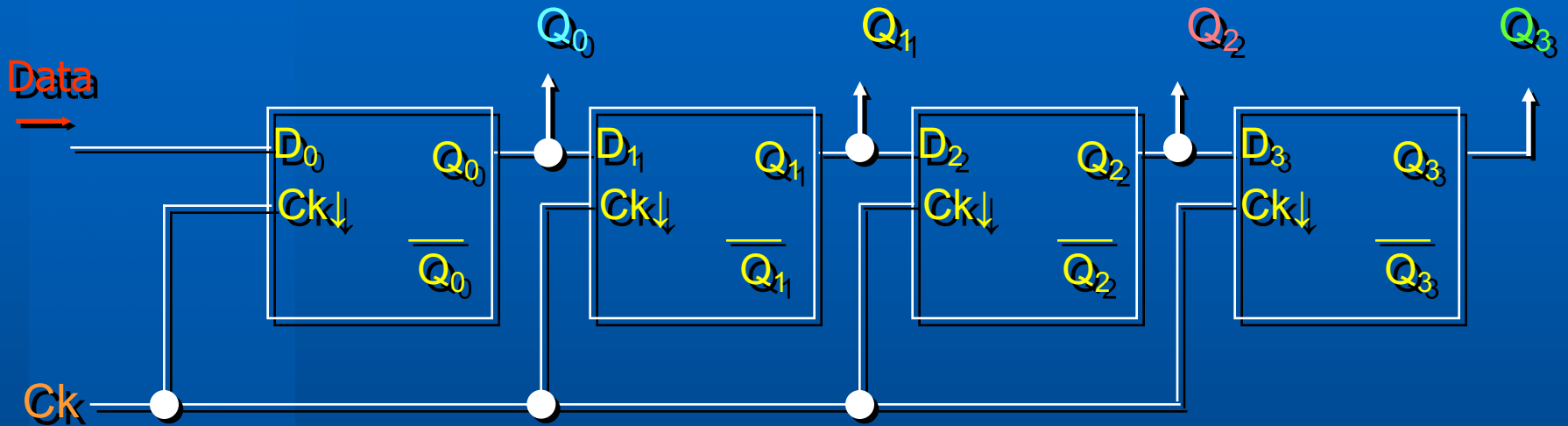
Sequência de entrada	$Q_0$	$Q_1$	$Q_2$	$Q_3$	Bordas do CK
1 1 0 1 →	0	0	0	0	
	1	0	0	0	1
	1	1	0	0	2
	0	1	1	0	3
	1	0	1	1	4
		1	0	1	5
			1	0	6
				1	7

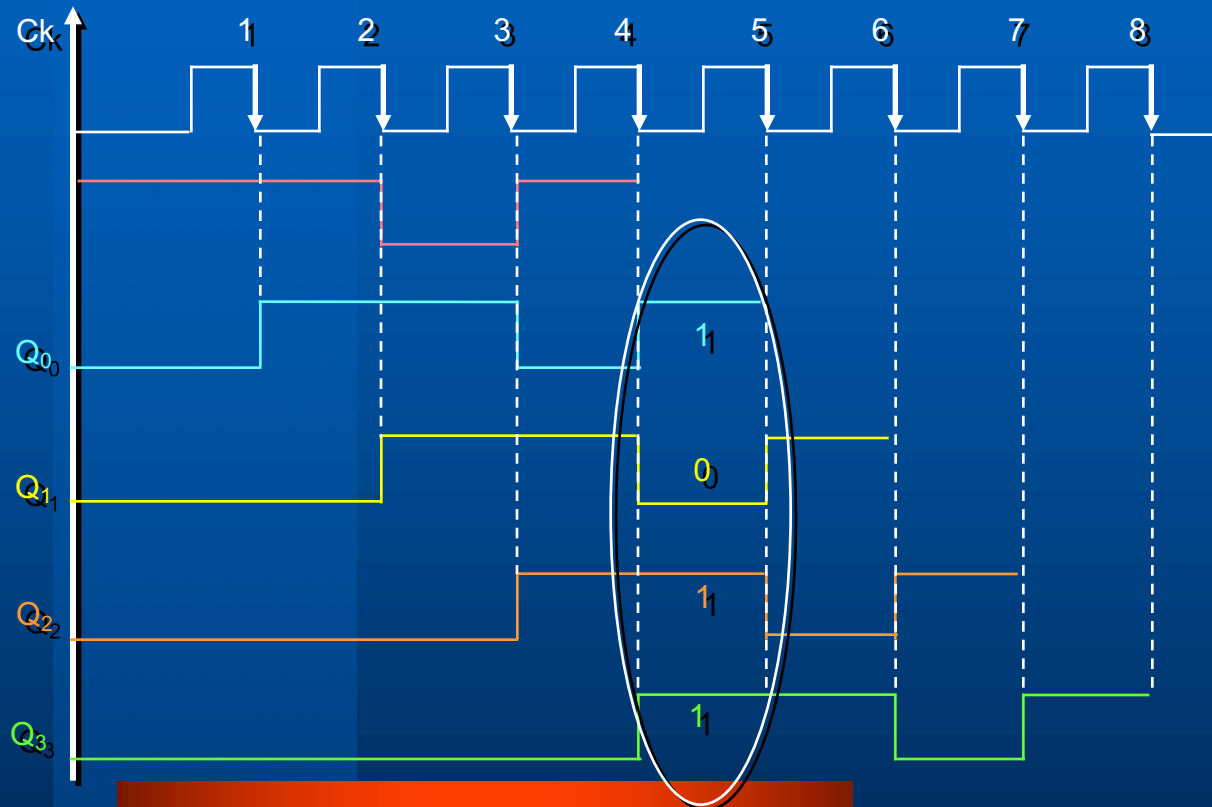
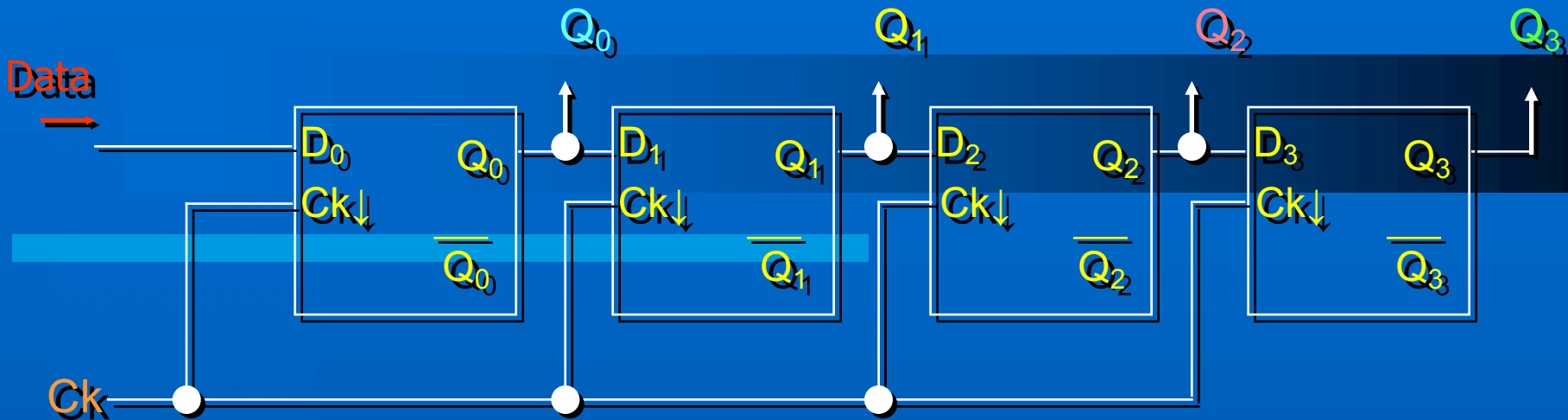
Saída serial

## 2. Transferência serial de dados de um registrador X para outro registrador Y



# 3. Conversor Serial/Paralelo

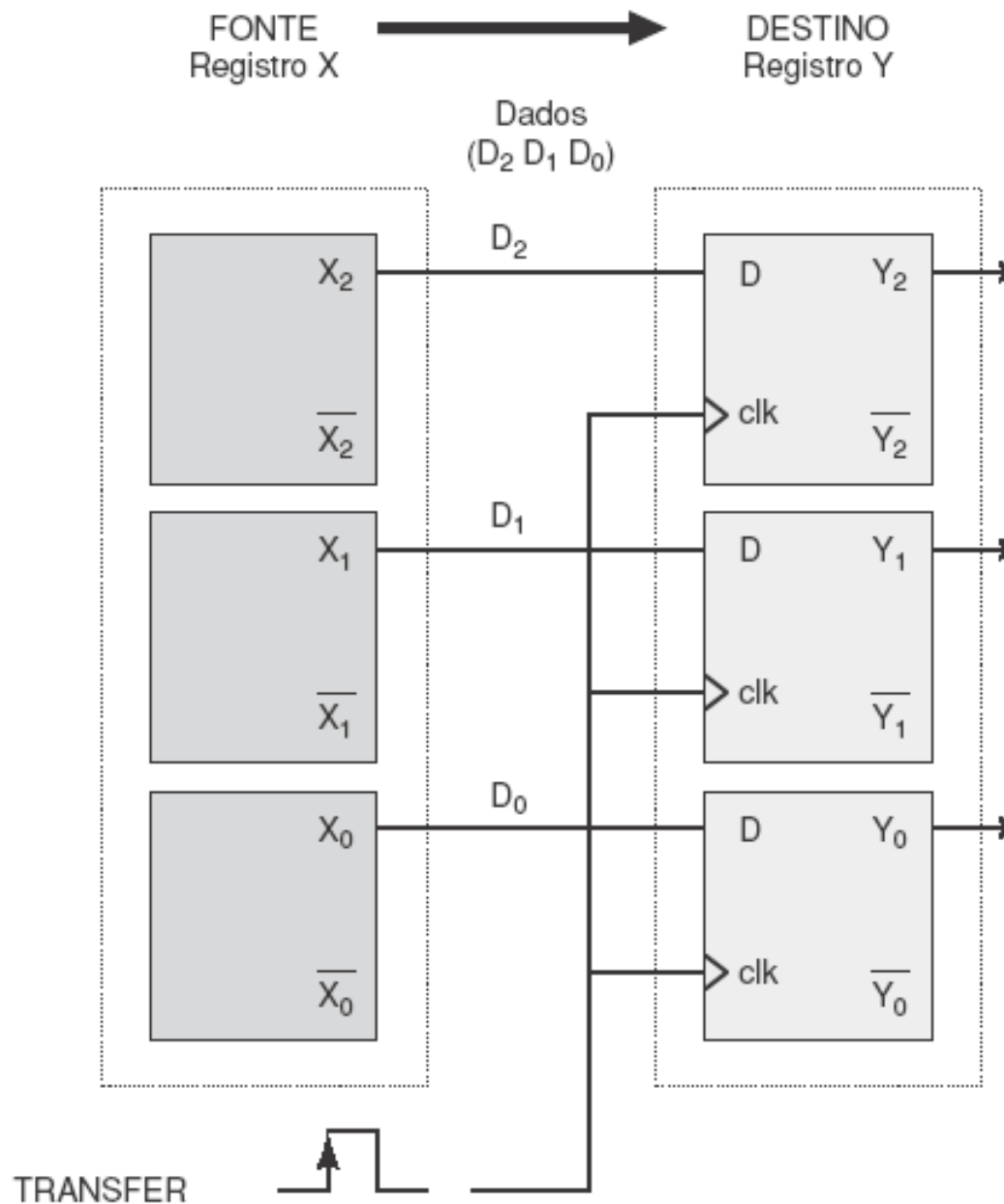




D	Q <sub>0</sub>	Q <sub>1</sub>	Q <sub>2</sub>	Q <sub>3</sub>	Ck
1	0	0	0	0	1
1	1	0	0	0	2
0	1	1	0	0	3
1	0	1	1	0	4

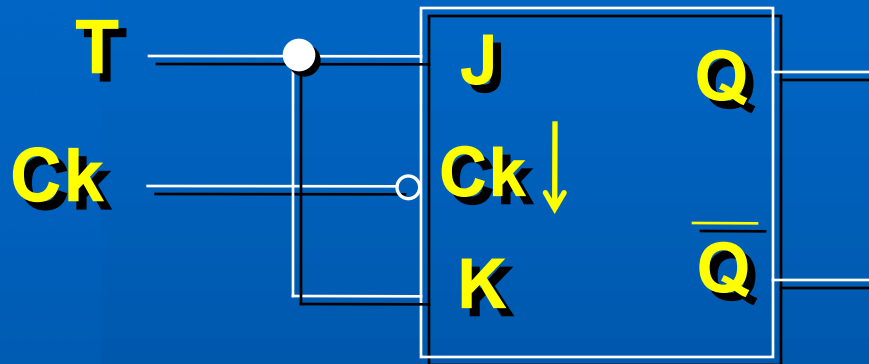
1 1 0 1  
 MSB = Q<sub>3</sub>                      LSB = Q<sub>0</sub>

# Transferência Paralela de Dados

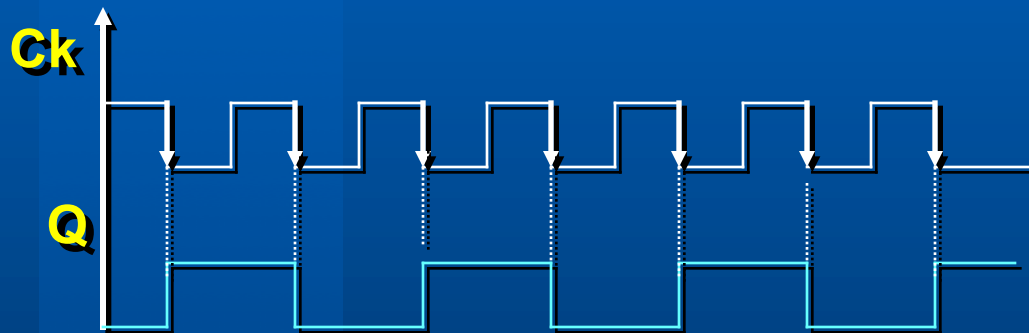


**FF tipo T**

# FF Tipo T (“Toggle”)



T	Q
0	$Q_0$
1	$\overline{Q_0}$



$T = 1$

$f_Q = f_{Ck} / 2$

Divisor por 2

FIM