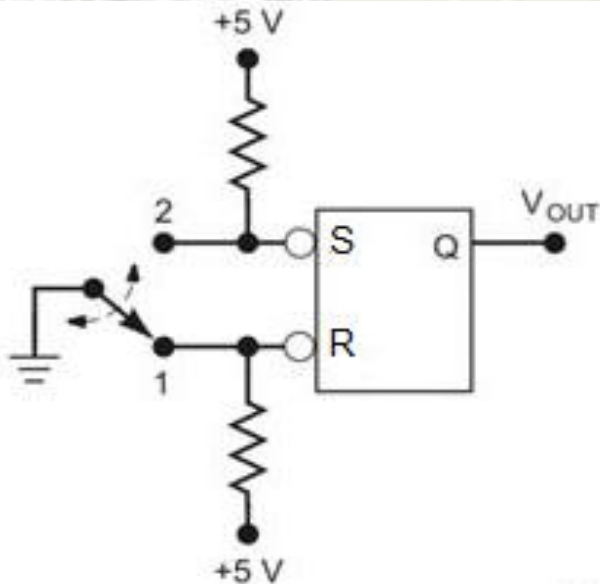
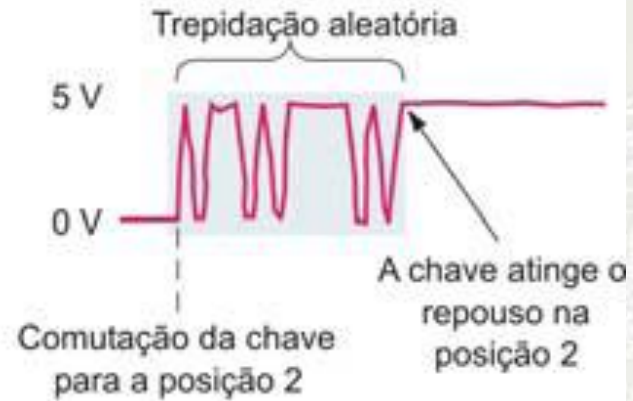
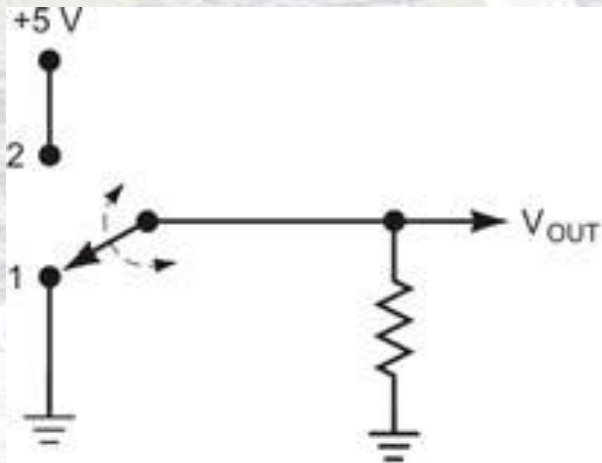


Centro Federal de Educação Tecnológica de Minas Gerais
ENGENHARIA DA COMPUTAÇÃO

Aula 09

Registradores de Deslocamentos e Contadores

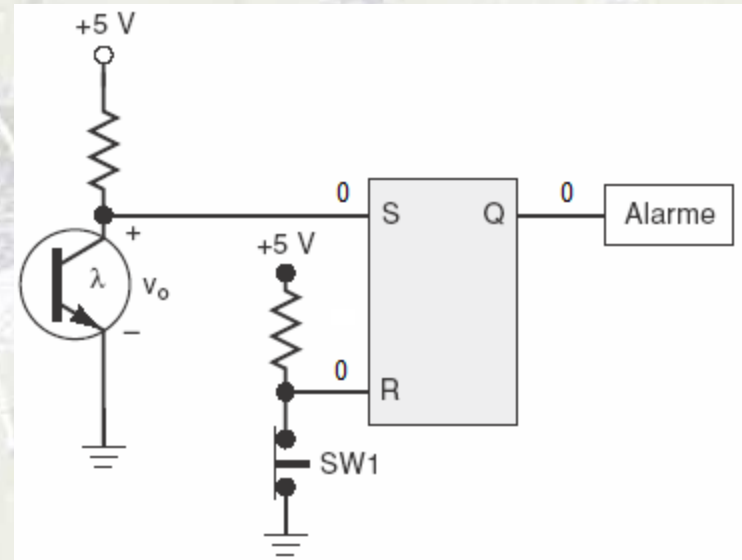
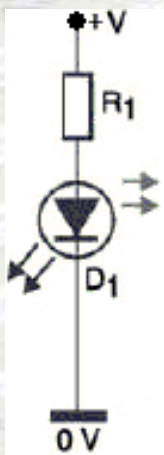
Aplicação Flip-Flop



Aplicação Flip-Flop

ALARME

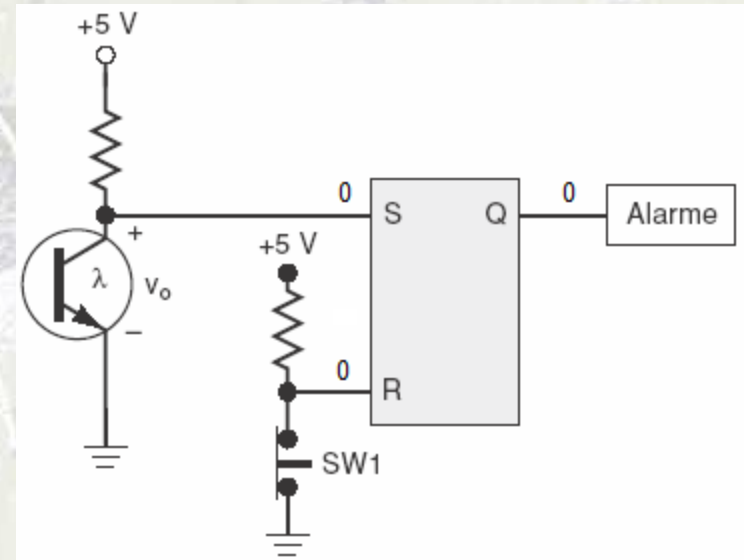
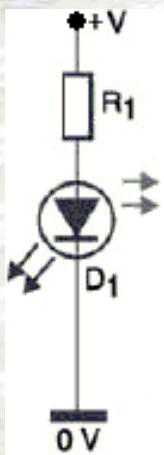
Considere que o fototransistor abaixo esteja inicialmente recebendo um feixe do diodo emissor de luz (LED) **D1**, e que o Latch RS tenha sido previamente levado para **Q = 0** (**resetado**) ao abrir a chave **SW1** momentaneamente.



Aplicação Flip-Flop

ALARME

Considere que o fototransistor abaixo esteja inicialmente recebendo um feixe do diodo emissor de luz (LED) **D1**, e que o Latch RS tenha sido previamente levado para **Q = 0** (**resetado**) ao abrir a chave **SW1** momentaneamente.

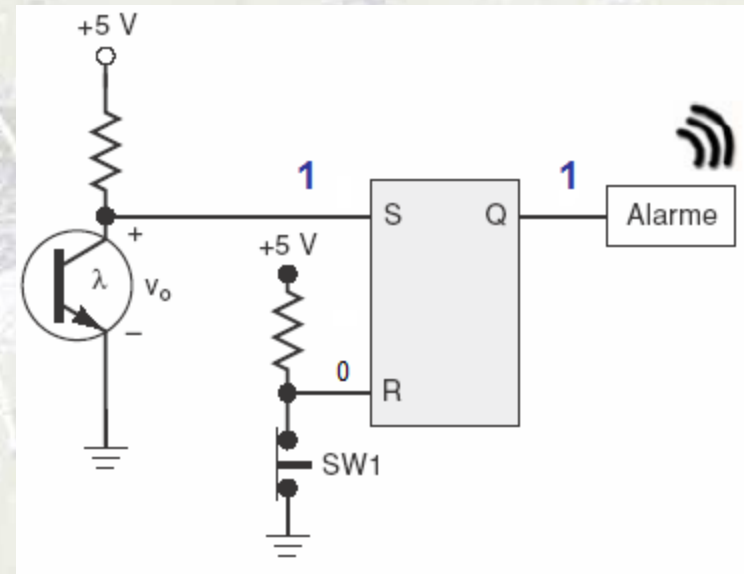
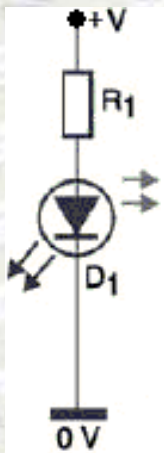


O que acontece se o feixe for momentaneamente interrompido ???

Aplicação Flip-Flop

ALARME

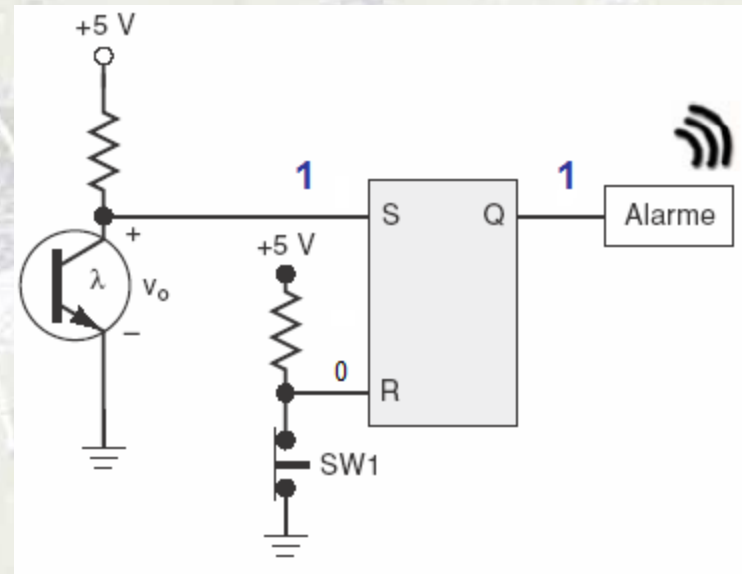
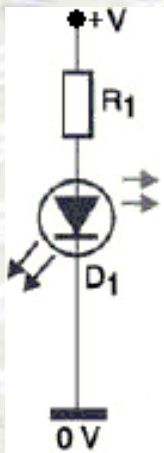
Considere que o fototransistor abaixo esteja inicialmente recebendo um feixe do diodo emissor de luz (LED) **D1**, e que o Latch RS tenha sido previamente levado para **Q = 0** (**resetado**) ao abrir a chave **SW1** momentaneamente.



Aplicação Flip-Flop

ALARME

Considere que o fototransistor abaixo esteja inicialmente recebendo um feixe do diodo emissor de luz (LED) **D1**, e que o Latch RS tenha sido previamente levado para **Q = 0** (resetado) ao abrir a chave **SW1** momentaneamente.

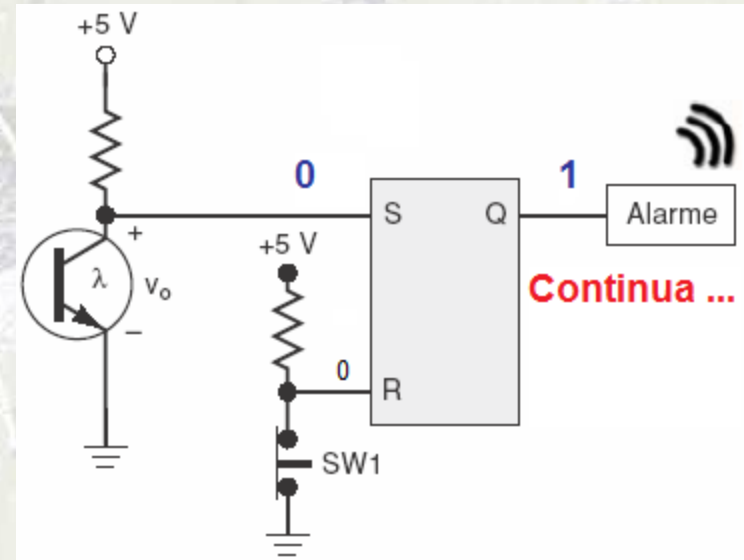
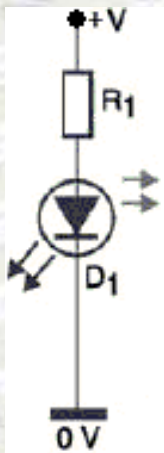


... e se o feixe for restabelecido ???

Aplicação Flip-Flop

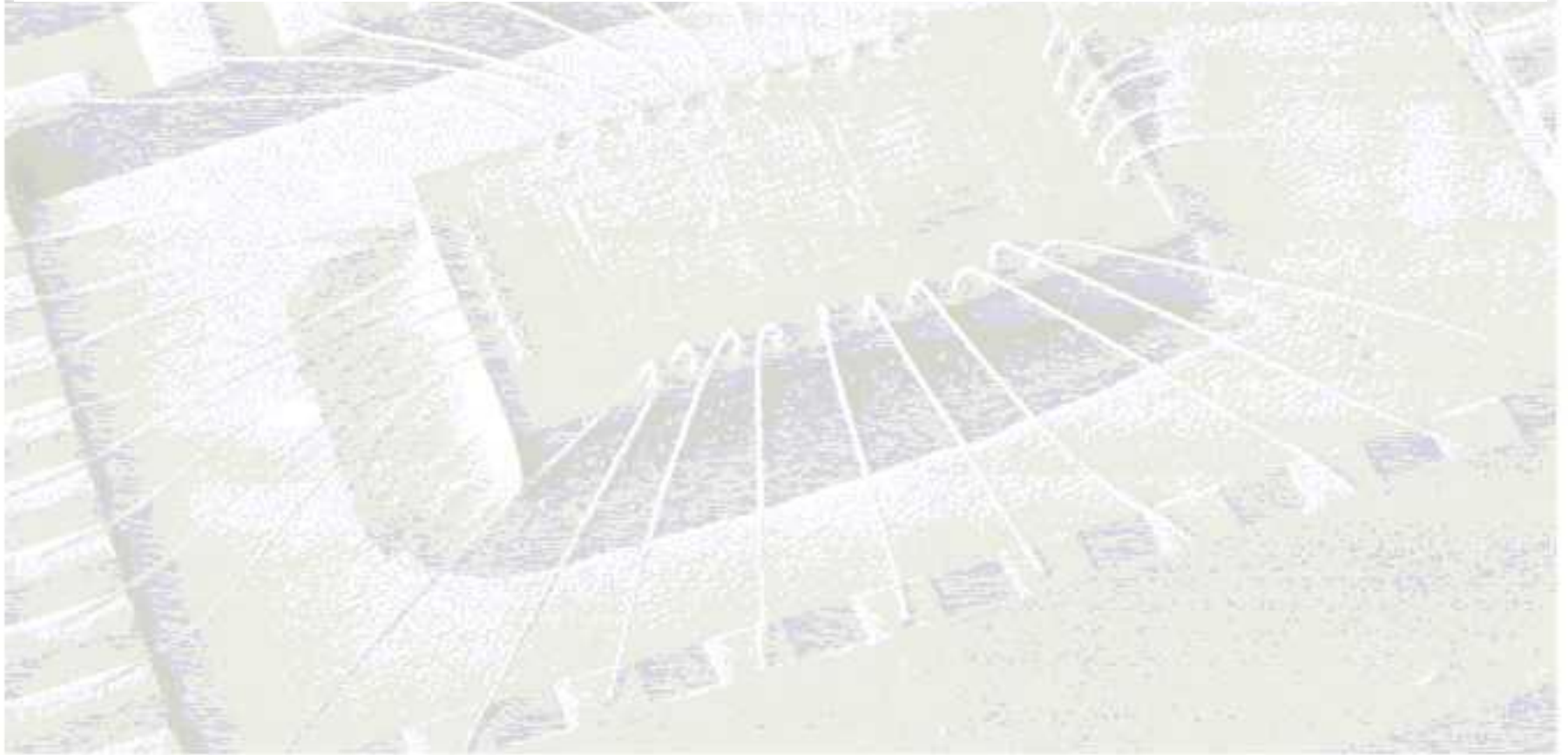
ALARME

Considere que o fototransistor abaixo esteja inicialmente recebendo um feixe do diodo emissor de luz (LED) **D1**, e que o Latch RS tenha sido previamente levado para **Q = 0** (**resetado**) ao abrir a chave **SW1** momentaneamente.



Sincronização de Flip-Flops

A maioria dos **sistemas digitais** opera de forma, essencialmente, **síncrona** e a maioria dos sinais muda de estado em sincronismo com as transições do **CLK**.



Sincronização de Flip-Flops

A maioria dos **sistemas digitais** opera de forma, essencialmente, **síncrona** e a maioria dos sinais muda de estado em sincronismo com as transições do **CLK**.

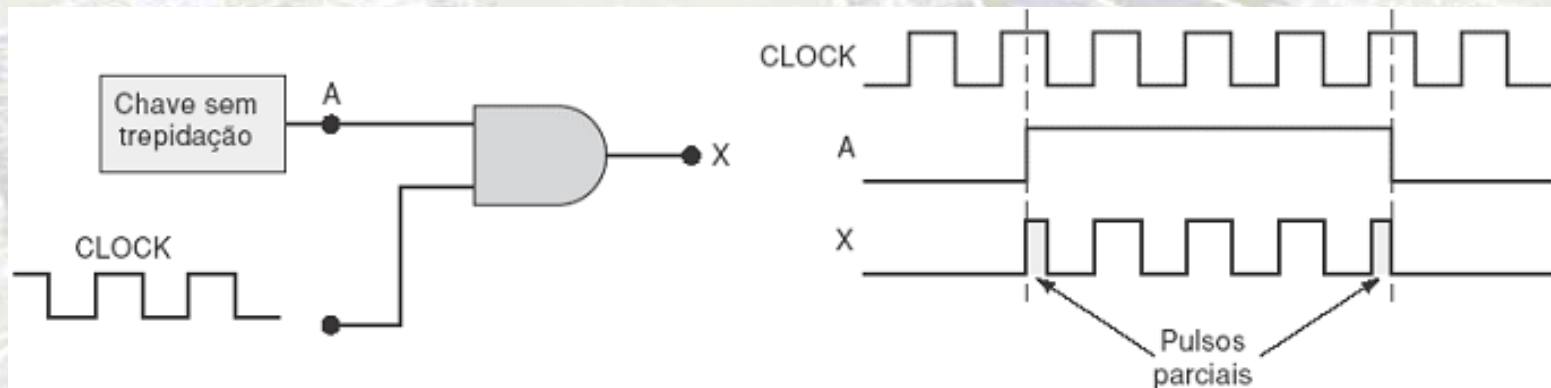
Em muitos casos, entretanto, haverá um sinal externo que não estará sincronizado com o **CLK**, em outras palavras, ele será um **sinal assíncrono**. Esse sinal aleatório poderá produzir resultados imprevisíveis e indesejados.

Sincronização de Flip-Flops

A maioria dos **sistemas digitais** opera de forma, essencialmente, **síncrona** e a maioria dos sinais muda de estado em sincronismo com as transições do **CLK**.

Em muitos casos, entretanto, haverá um sinal externo que não estará sincronizado com o **CLK**, em outras palavras, ele será um **sinal assíncrono**. Esse sinal aleatório poderá produzir resultados imprevisíveis e indesejados.

A figura abaixo mostra uma situação em que o sinal de entrada *A* é gerado a partir de uma chave, sem o efeito de trepidação, acionada por um operador.

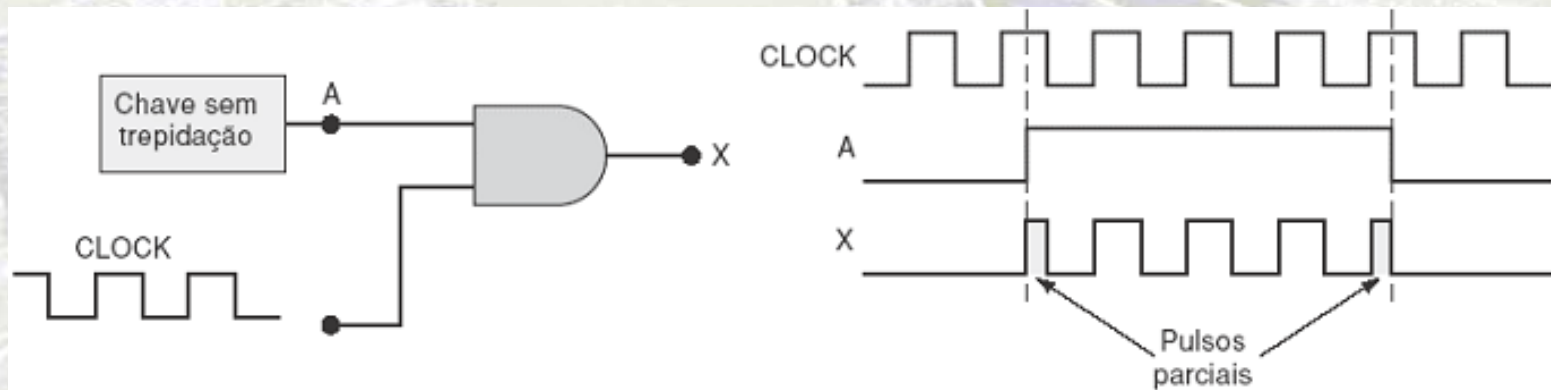


Sincronização de Flip-Flops

A maioria dos **sistemas digitais** opera de forma, essencialmente, **síncrona** e a maioria dos sinais muda de estado em sincronismo com as transições do **CLK**.

Em muitos casos, entretanto, haverá um sinal externo que não estará sincronizado com o **CLK**, em outras palavras, ele será um **sinal assíncrono**. Esse sinal aleatório poderá produzir resultados imprevisíveis e indesejados.

A figura abaixo mostra uma situação em que o sinal de entrada *A* é gerado a partir de uma chave, sem o efeito de trepidação, acionada por um operador.

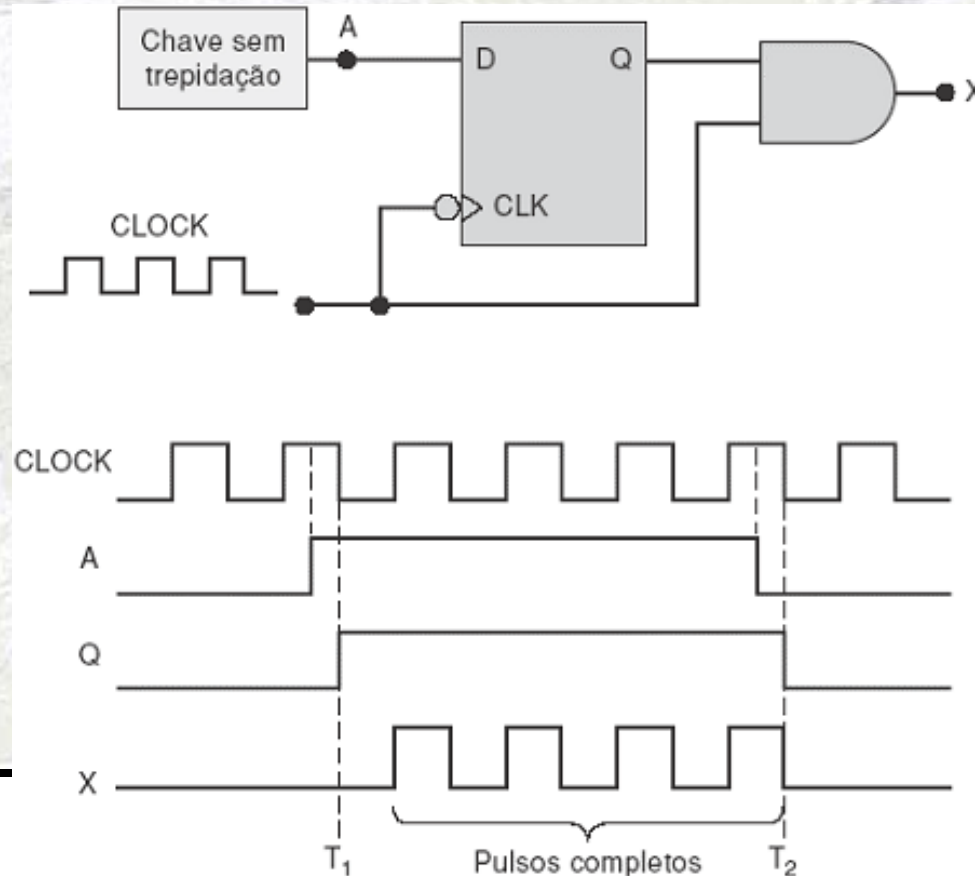


O problema é que por ser assíncrona a entrada *A*, ela pode produzir pulsos parciais de CLK na saída *X*. **Como evitar a ocorrência desses pulsos parciais???**

Sincronização de Flip-Flops

Solução:

Ao se adicionar um flip-flop *D* ao circuito, podemos solucionar o problema de pulsos parciais na saída.



Detectando uma Sequência

Em muitas situações, uma saída é ativada apenas quando as entradas são ativadas em uma determinada sequência. Isso não pode ser realizado usando apenas lógica combinacional, sendo necessário o uso da característica de armazenamento dos FFs.



Detectando uma Sequência

Em muitas situações, uma saída é ativada apenas quando as entradas são ativadas em uma determinada sequência. Isso não pode ser realizado usando apenas lógica combinacional, sendo necessário o uso da característica de armazenamento dos FFs.

Por exemplo, uma porta AND pode ser usada para determinar quando duas entradas *A* e *B* estão ambas em nível ALTO, mas a sua saída responderá da mesma forma independente de qual entrada foi primeiro para o nível ALTO.

Porém, suponha que desejamos gerar uma saída em nível ALTO ***apenas se a entrada A for para nível ALTO e, alguns instantes depois, a entrada B for para nível ALTO.***

Detectando uma Sequência

Em muitas situações, uma saída é ativada apenas quando as entradas são ativadas em uma determinada sequência. Isso não pode ser realizado usando apenas lógica combinacional, sendo necessário o uso da característica de armazenamento dos FFs.

Por exemplo, uma porta AND pode ser usada para determinar quando duas entradas *A* e *B* estão ambas em nível ALTO, mas a sua saída responderá da mesma forma independente de qual entrada foi primeiro para o nível ALTO.

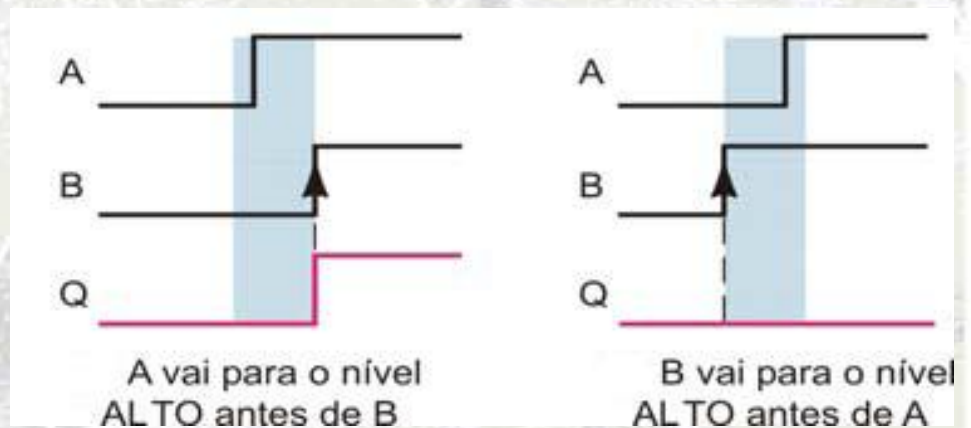
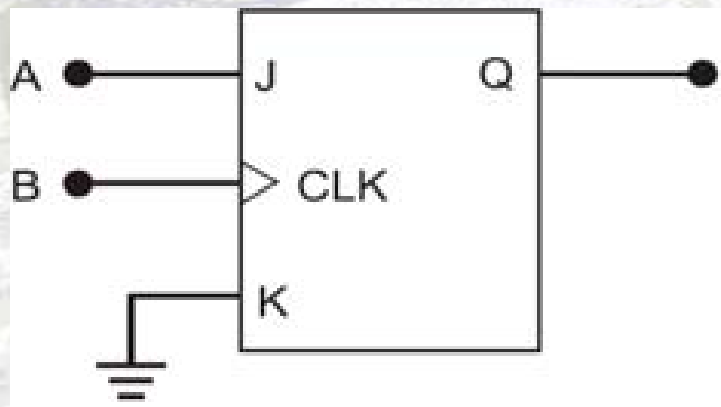
Porém, suponha que desejamos gerar uma saída em nível ALTO ***apenas se a entrada A for para nível ALTO e, alguns instantes depois, a entrada B for para nível ALTO.***

Como implementar esse sistema ????????????

Detectando uma Sequência

Solução:

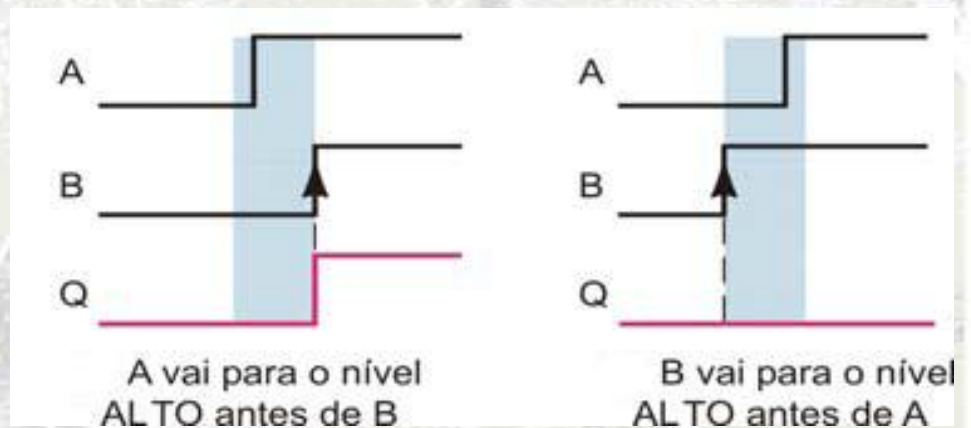
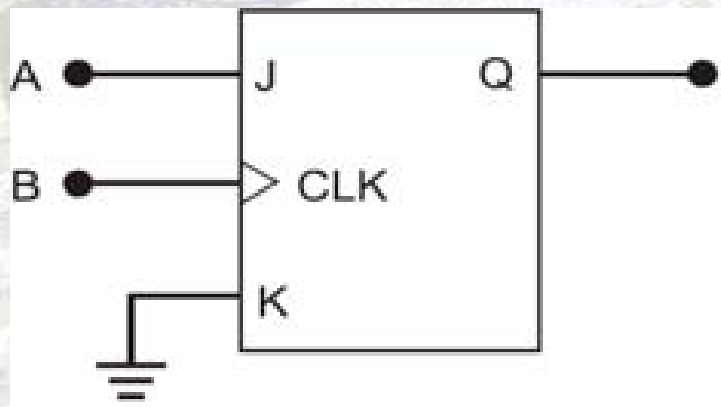
As formas de onda abaixo mostram como o sistema baseado no flip-flop *J-K* obedece ao comportamento desejado.



Detectando uma Sequência

Solução:

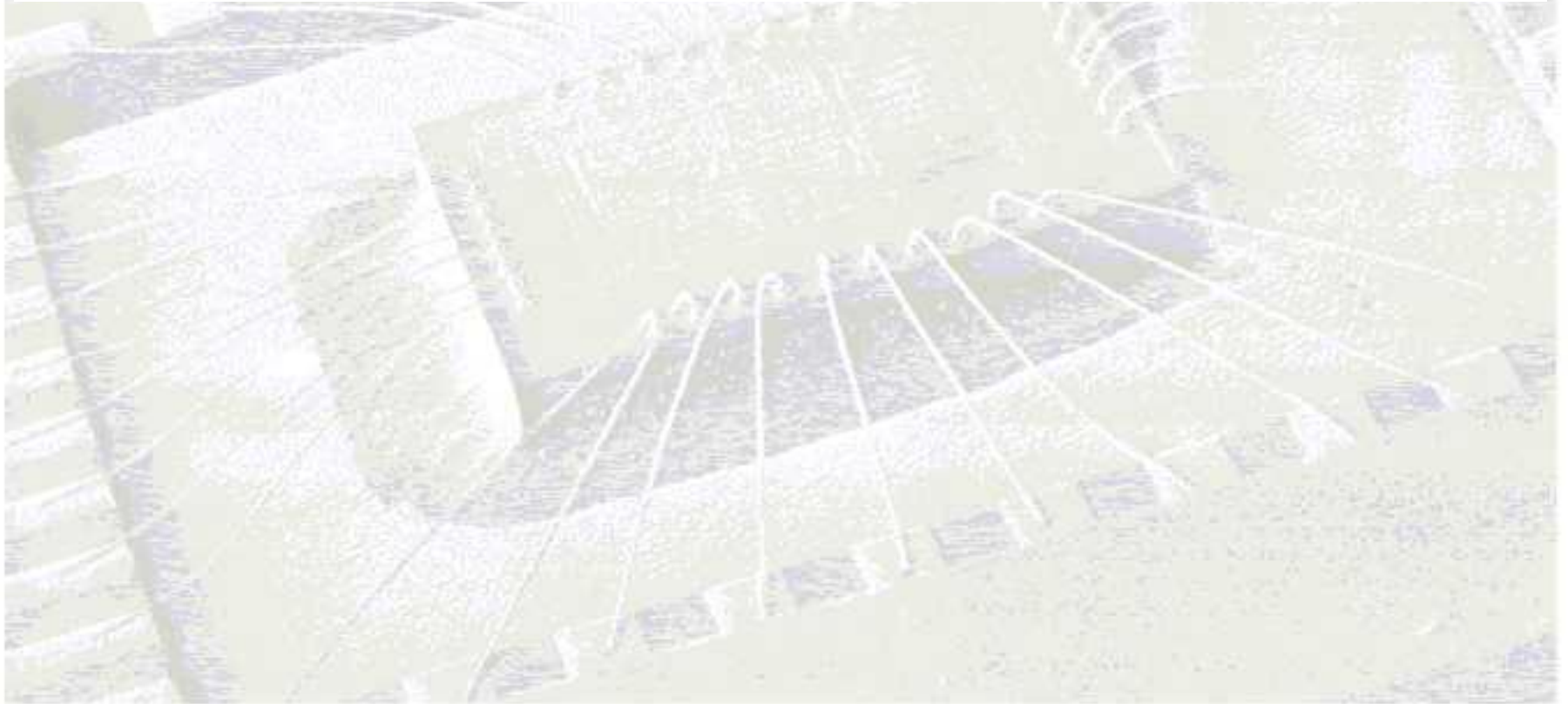
As formas de onda abaixo mostram como o sistema baseado no flip-flop *J-K* obedece ao comportamento desejado.



Resumindo: Para que esse circuito funcione, adequadamente, a entrada *A* tem de estar em nível ALTO antes da entrada *B*.

Armazenamento Dados

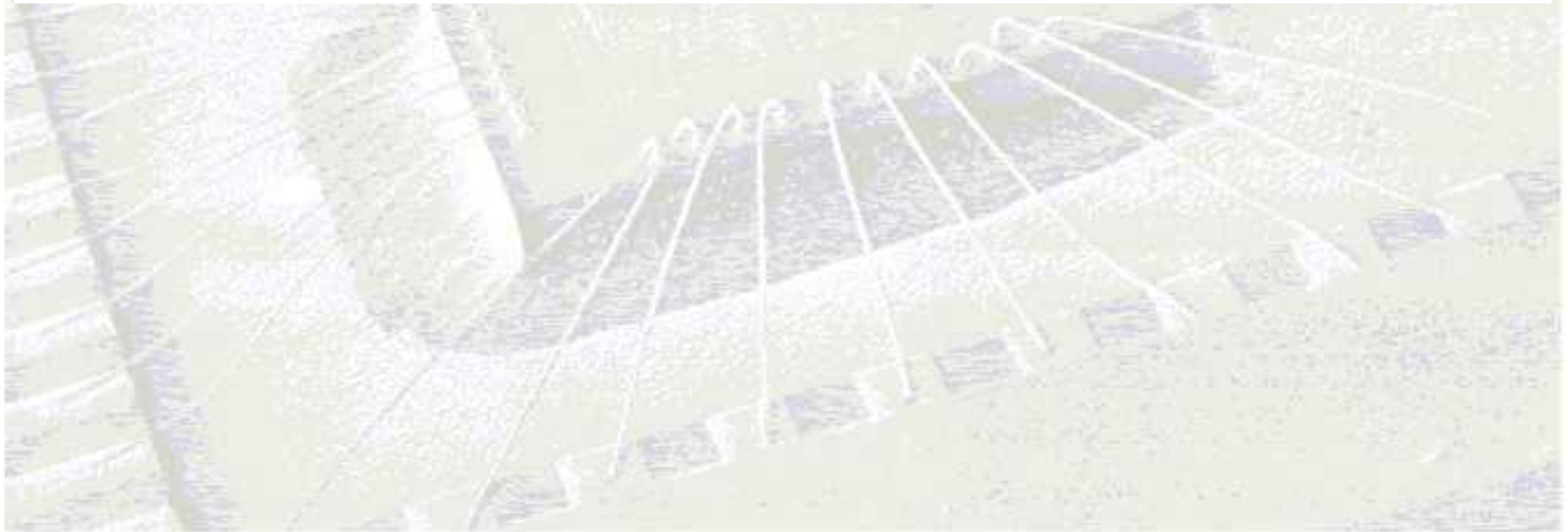
O uso mais comum de flip-flops é no armazenamento de dados ou informações. Esses dados são geralmente armazenados em **grupos de FFs** denominados **registradores**.



Armazenamento Dados

O uso mais comum de flip-flops é no armazenamento de dados ou informações. Esses dados são geralmente armazenados em **grupos de FFs** denominados **registradores**.

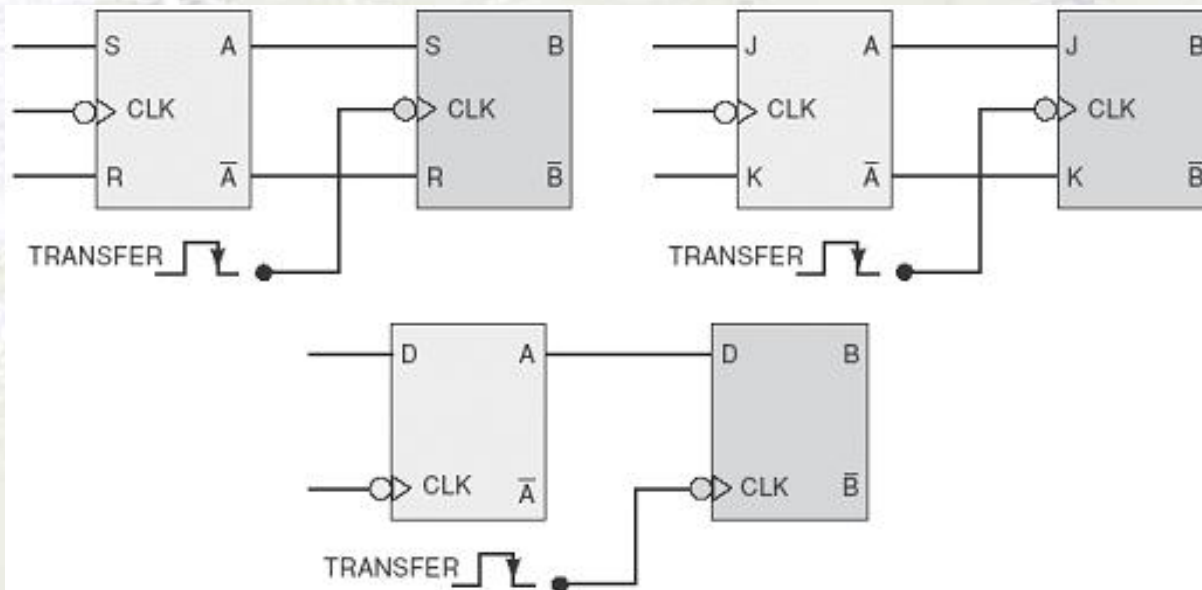
A operação mais comum realizada sobre os dados armazenados em FFs (ou registradores) é a operação de **transferência de dados**. Essa operação é ilustrada abaixo para os flip-flops *RS*, *JK* e *D*.



Armazenamento Dados

O uso mais comum de flip-flops é no armazenamento de dados ou informações. Esses dados são geralmente armazenados em **grupos de FFs** denominados **registradores**.

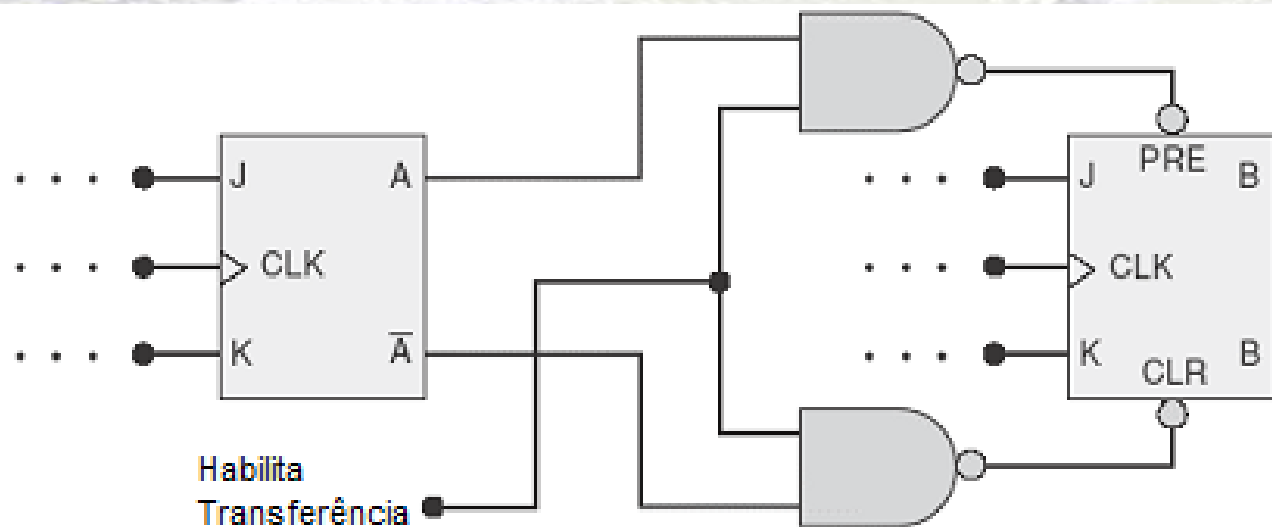
A operação mais comum realizada sobre os dados armazenados em FFs (ou registradores) é a operação de **transferência de dados**. Essa operação é ilustrada abaixo para os flip-flops *RS*, *JK* e *D*.



Essas operações de transferência são síncronas pois utilizam o mesmo CLK.

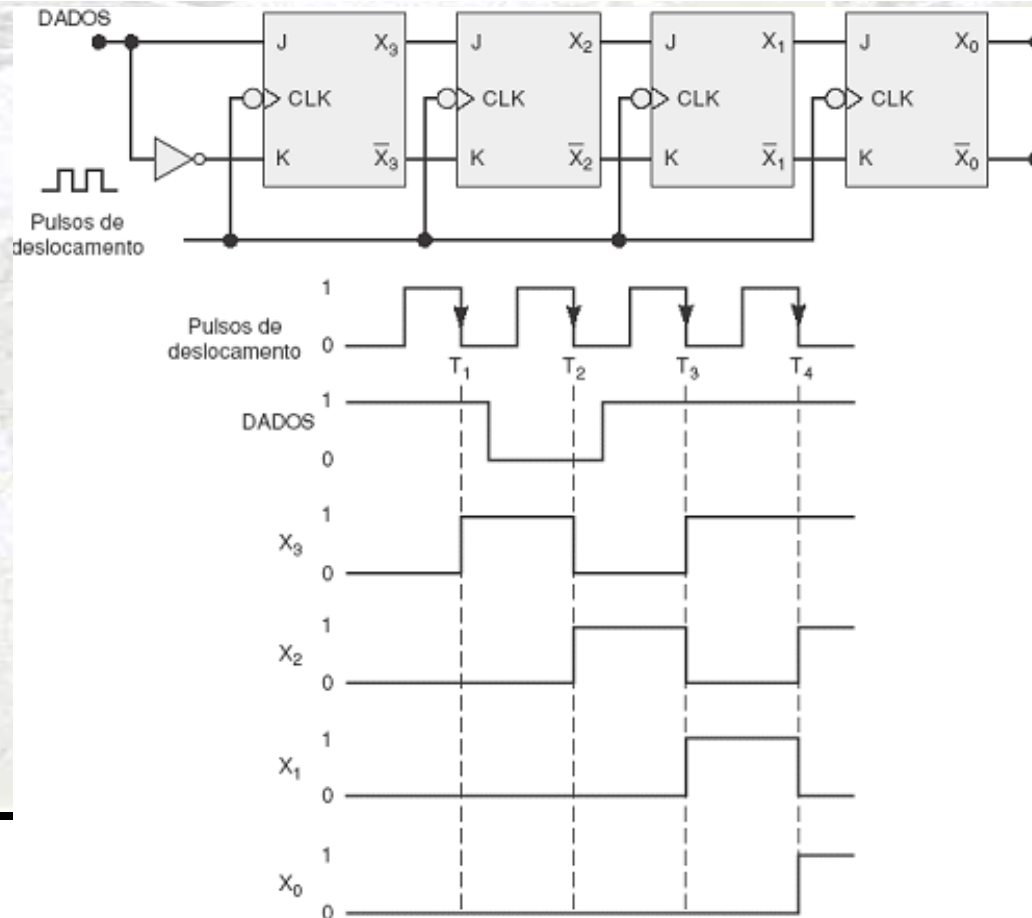
Transferência de Dados

A operação de transferência também pode ser obtida usando entradas assíncronas de um FF (**transferência assíncrona**), conforme o circuito da figura abaixo.



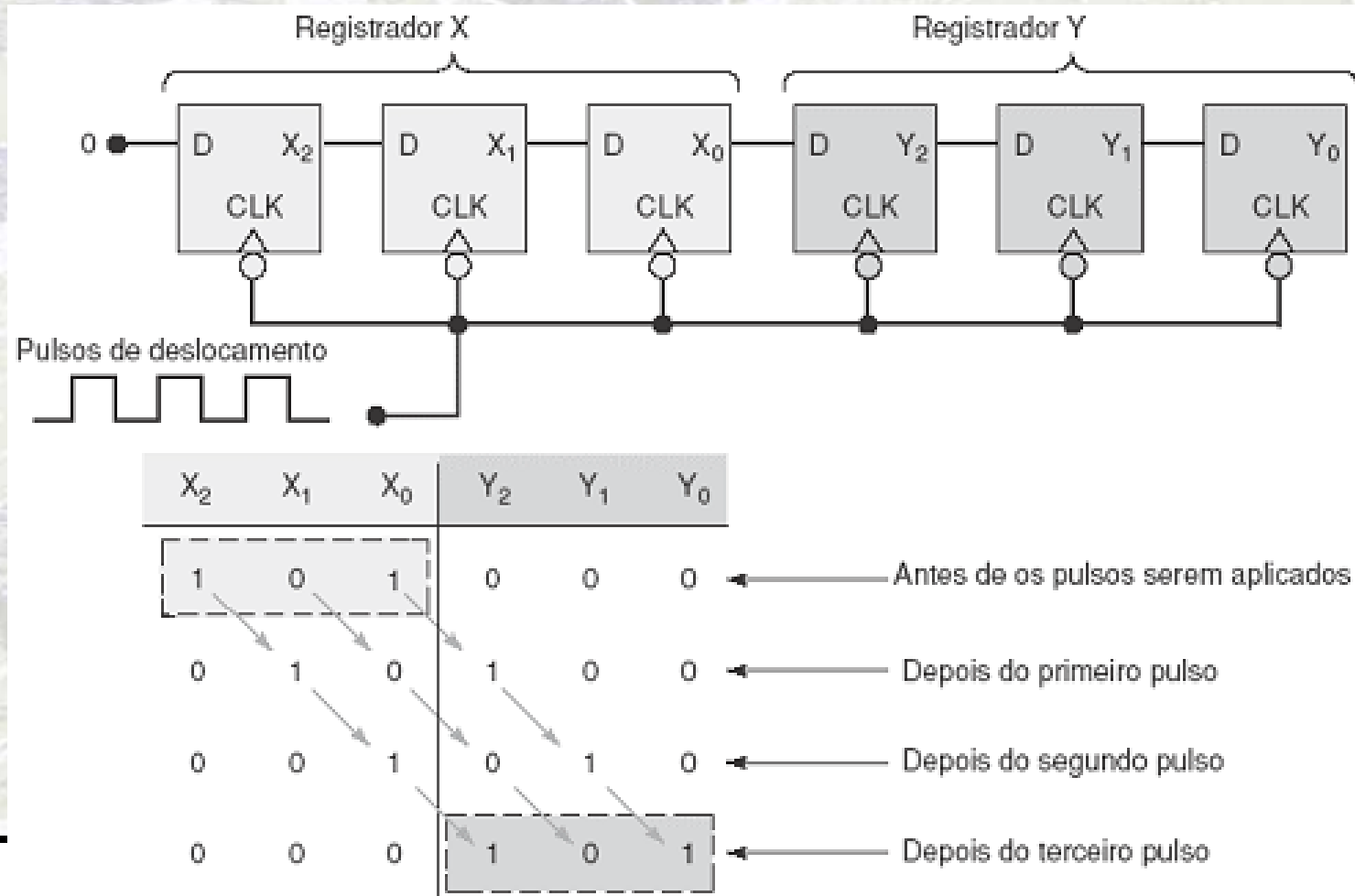
Transferência Serial de Dados

Um **registrador de deslocamento** é um grupo de FFs organizados de modo que os números binários armazenados nos FFs sejam deslocados de um FF para o seguinte a cada pulso de CLK.



Transferência Serial de Dados

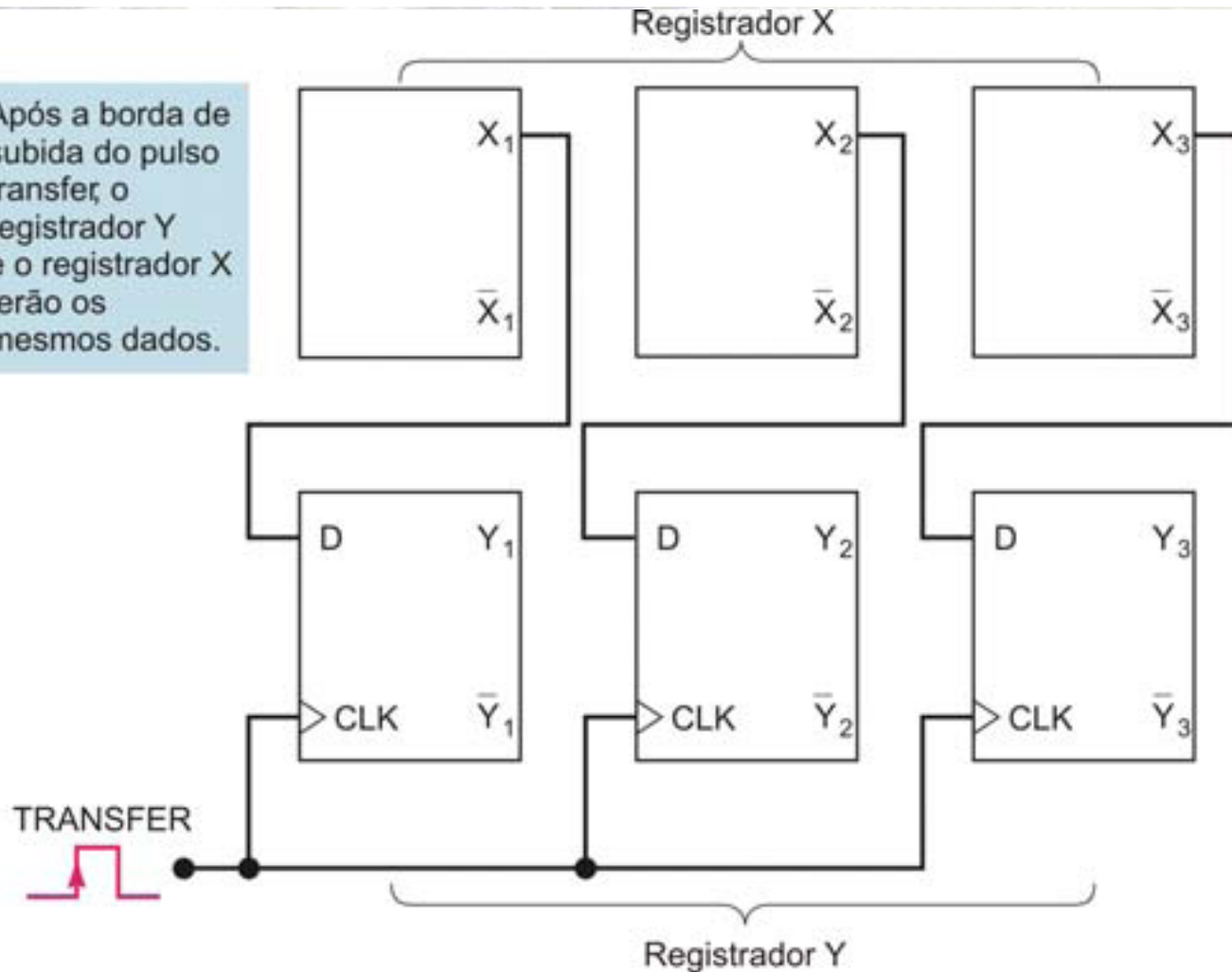
Transferência serial entre registradores



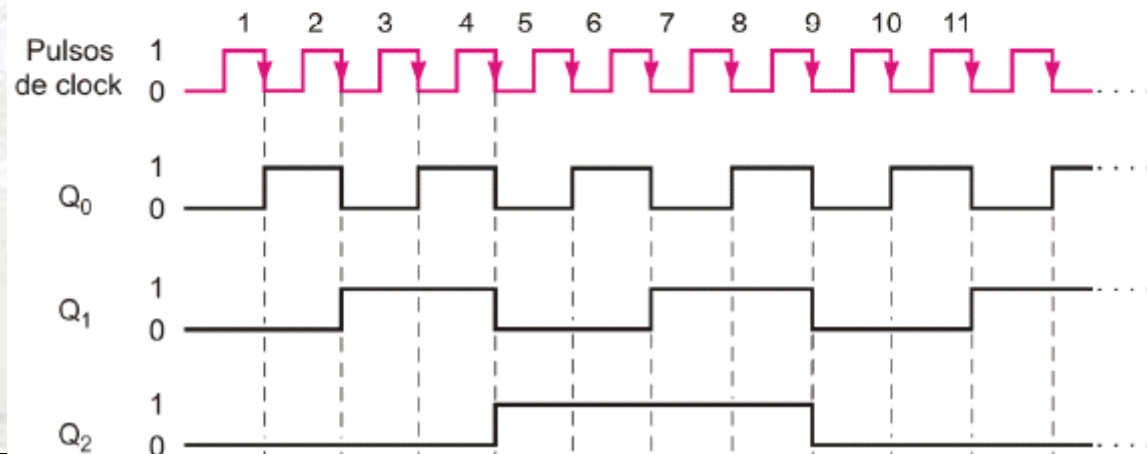
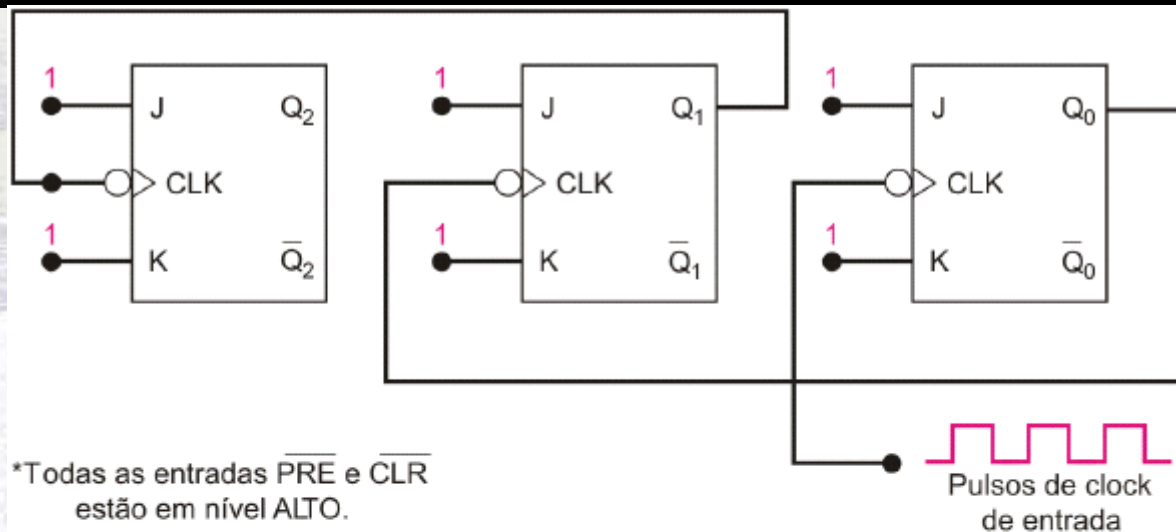
Transferência Paralela de Dados

Transferência paralela entre registradores

Após a borda de subida do pulso transfer, o registrador Y e o registrador X terão os mesmos dados.



Divisão de Frequência

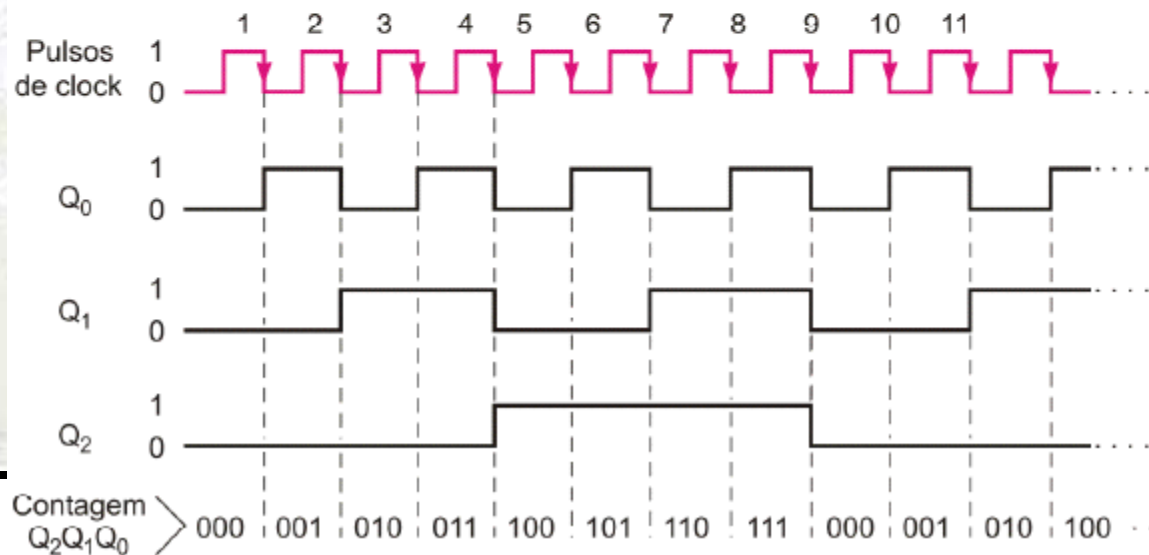
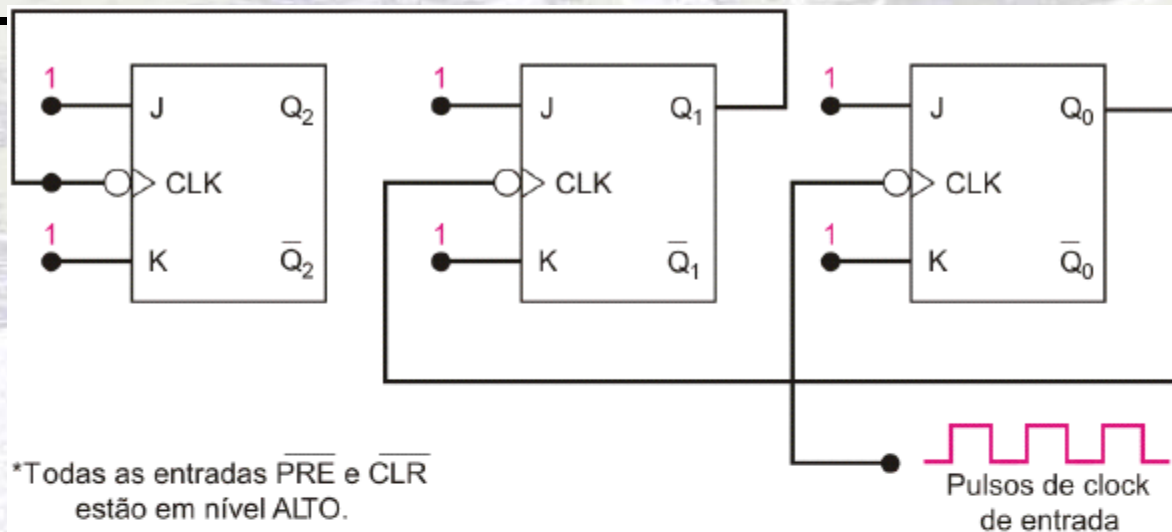


Operação de Contagem

Além de funcionar como um divisor de frequência, o circuito apresentado também funciona como um **contador binário**. Isso pode ser demonstrado analisando-se a seqüência de estados dos FFs após a ocorrência de cada pulso de CLK. A **tabela de estados** do circuito mostra esse comportamento.

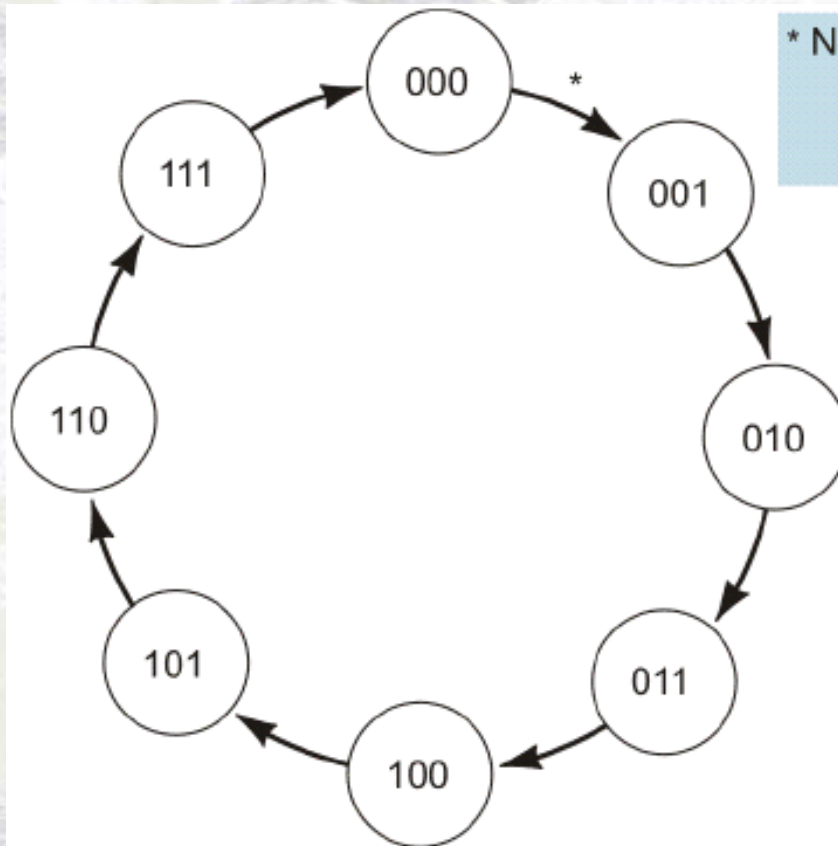
2^2 Q_2	2^1 Q_1	2^0 Q_0	
0	0	0	Antes de aplicar os pulsos de clock
0	0	1	Após o pulso #1
0	1	0	Após o pulso #2
0	1	1	Após o pulso #3
1	0	0	Após o pulso #4
1	0	1	Após o pulso #5
1	1	0	Após o pulso #6
1	1	1	Após o pulso #7
0	0	0	Após o pulso #8 retorna para 000
0	0	1	Após o pulso #9
0	1	0	Após o pulso #10
0	1	1	Após o pulso #11
.	.	.	.
.	.	.	.

Contagem Binária



Contagem Binária

Outra forma de mostrar como os estados dos FFs mudam a cada pulso de CLK aplicado é através do uso de um **diagrama de transição de estados**.



* Nota: cada seta representa a ocorrência de um pulso de clock