

*UNIOESTE*  
*Ciência da Computação*

*Sistemas Digitais*  
*Circuitos Sequenciais*

*Prof. Jorge Habib El Khouri*  
*Prof. Antonio Marcos Hachisuca*

2020/2021

## Referências Bibliográficas

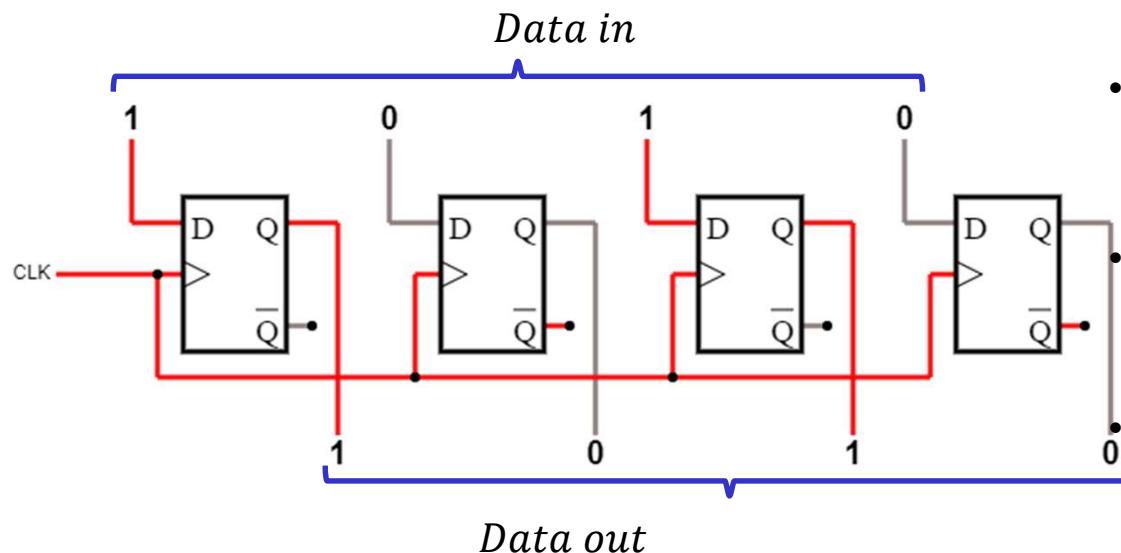
1. *Digital Fundamentals*, Thomas L. Floyd; Editora: Pearson; Edição: 11; Ano: 2015;
2. *Sistemas Digitais Princípios e Aplicações*, Ronald J. Tocci; Editora: Pearson; Edição: 11; Ano: 2011;
3. *Computer Organization and Design*, David A. Patterson; Editora: Elsevier; Edição: 1; Ano: 2017
4. *Digital Design: Principles and Practices*, John F. Wakerly; Editora: Pearson; Edição: 5; Ano: 2018;
5. *Guide to Assembly Language Programming in Linux*, Sivarama P. Dandamudi; Editora: Springer; Edição: 1; Ano: 2005.
6. *Fundamentals of Logic Design*, Roth Jr, Charles H; Kinney, Larry L; Seventh Edition. Editora: Cengage Learning, Ano: 2013.

## Sumário

- 1. Revisão – Sistemas de Numeração
- 2. Revisão – Representação de Dados
- 3. Revisão – Operações com Binários
- 4. Álgebra Booleana
- 5. Simplificação de Expressões
- 6. Mapa de Karnaugh
- 7. Elementos Lógicos Universais
- 8. Circuitos Combinacionais
- 9. Circuitos Sequenciais
  - 1. Latches
  - 2. Flip-Flop
  - 3. Registradores
  - 4. Contadores
  - 5. Máquina de Estados
  - 6. Memória RAM

## Circuitos Sequenciais Registradores

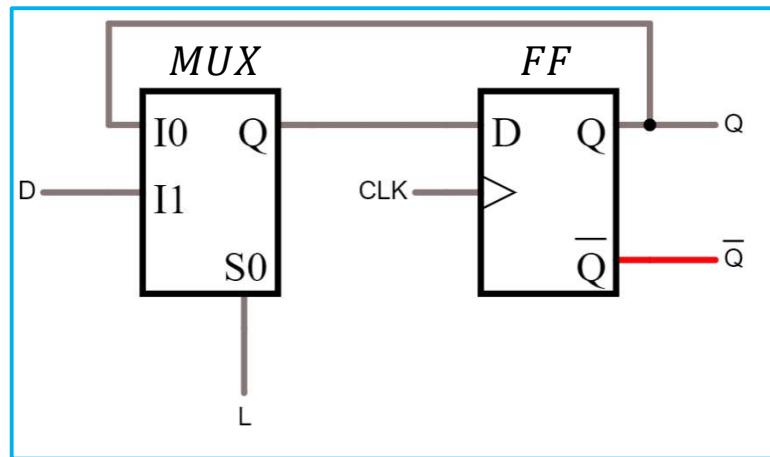
- *Flip-flops* podem ser agrupados para formar um registrador;
- O seguinte circuito exemplifica um registrador de 4 bits:



- Os dados de entrada serão transferidos para a saída na transição de borda do *clock*;  
Todos os *flip-flops* compartilham o mesmo sinal de *clock*;  
Esta configuração é denominada *parallel in – parallel out*.

## Circuitos Sequenciais Registradores

- É desejável que os dados de entrada sejam transferidos para a saída de forma controlada, além da sincronização com o *clock*;
- O seguinte circuito combina um *MUX-1* com o *flip-flop D*, agregando assim um sinal de *Load* ou *Enable* ao *flip-flop*:

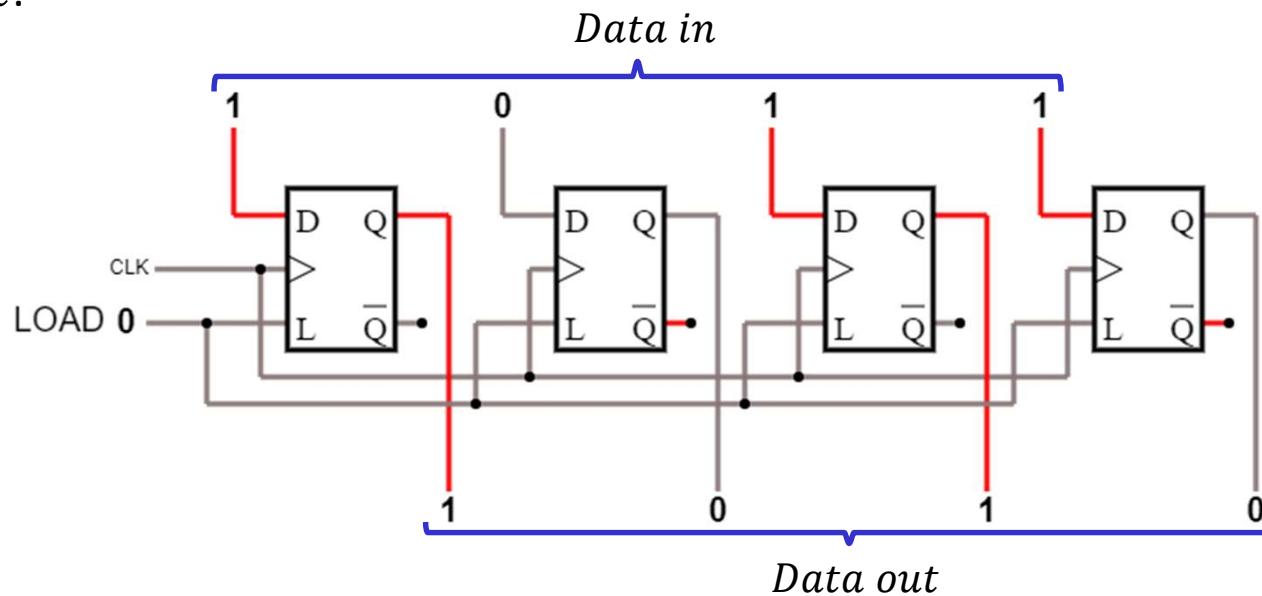


- Quando  $L == 0$ , a saída  $Q$  do *flip-flop* é mantida inalterada, em função da realimentação de  $Q$  em  $I_0$ ;
- Quando  $L == 1$ , a entrada  $D$  é transferida para a saída do *MUX*, e consequentemente para o *flip-flop*;
- Na próxima transição de borda do *clock*, o  $Q$  do *flip-flop* refletirá o valor de  $D$ ;

<https://tinyurl.com/yzykugzl>

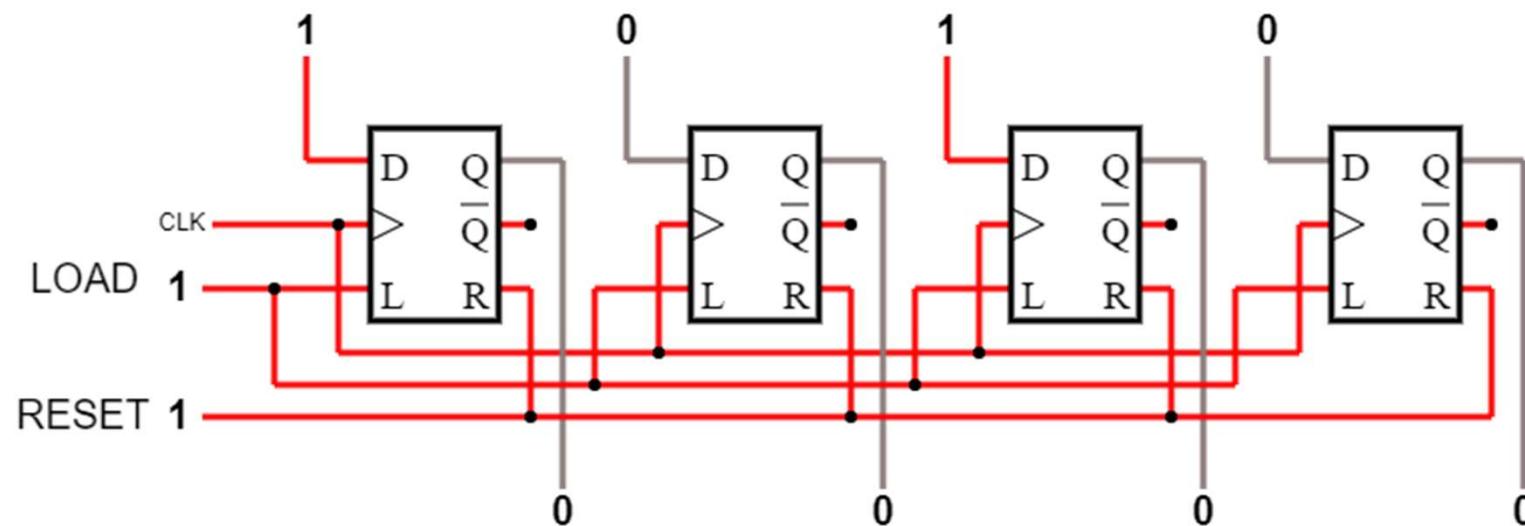
## Circuitos Sequenciais Registradores

- O seguinte circuito exemplifica um registrador de 4 bits com controle de *Load* ou *Select*:



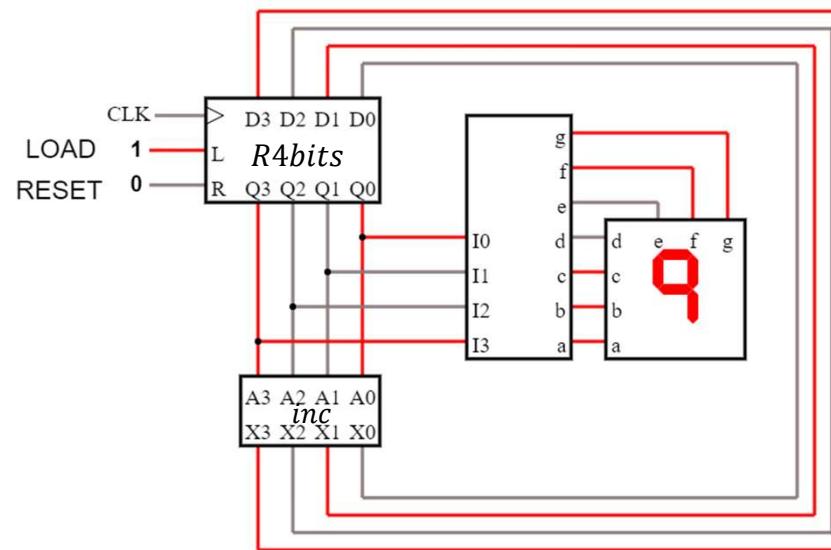
## Circuitos Sequenciais Registradores

- Uma funcionalidade adicional interessante seria zerar a saída do registrador:



## Circuitos Sequenciais Registradores

- O problema do incrementador:

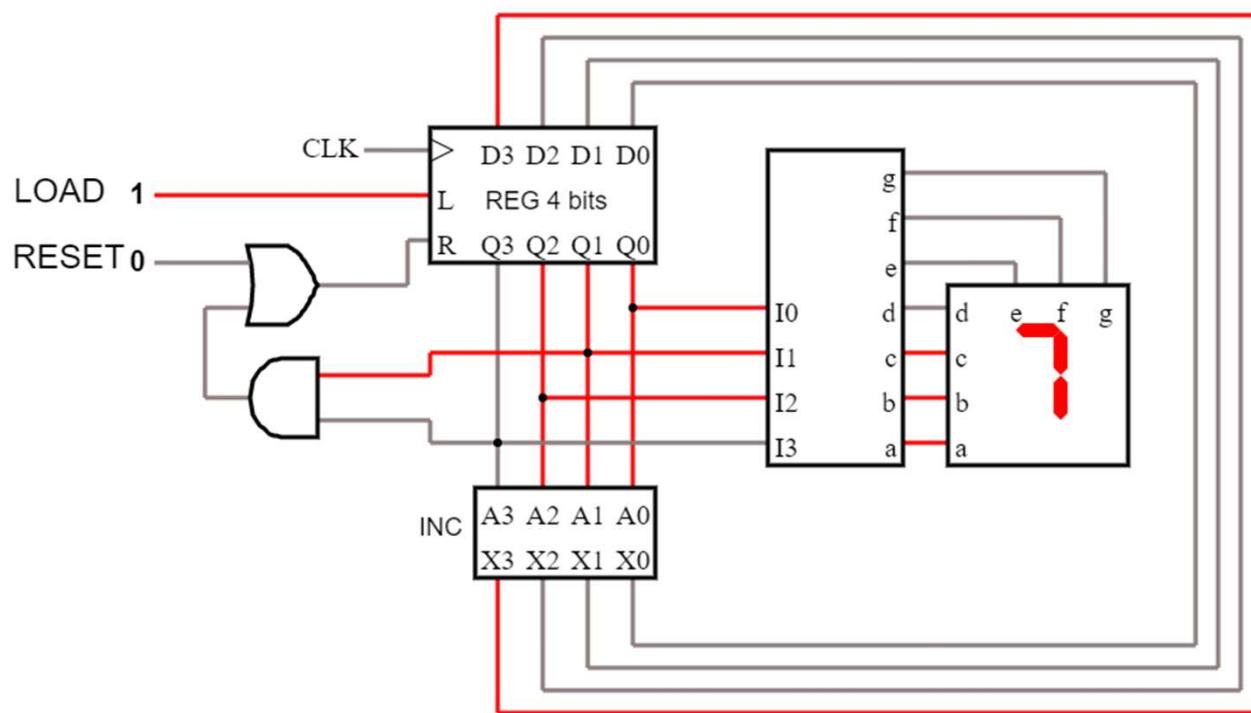


<https://tinyurl.com/yk2gzhjw>

## Circuitos Sequenciais Registradores

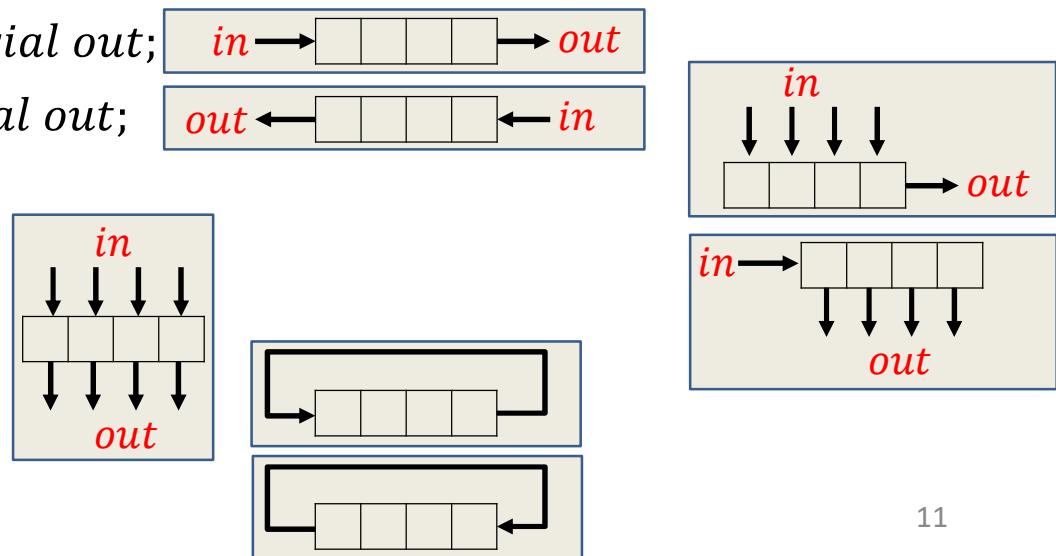
- i. Introduzir pequenas modificações no diagrama do incrementador apresentado para que a contagem seja de 0 a 9:

## Circuitos Sequenciais Registradores



## Circuitos Sequenciais Registradores

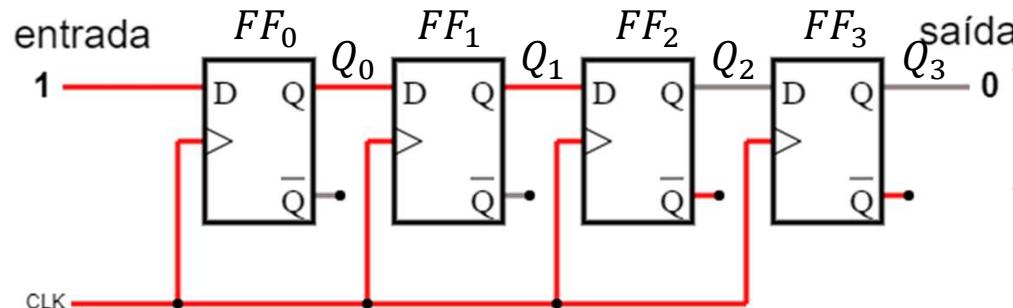
- Uma diversidade de registradores específicos são utilizados em circuitos digitais;
- Normalmente são designados como registradores de deslocamento ou *shift registers*;
- Os principais são:
  - ✓ *Serial in – shift right – serial out*; 
  - ✓ *Serial in – shift left – serial out*; 
  - ✓ *Parallel in – serial out*;
  - ✓ *Serial in – parallel out*;
  - ✓ *Parallel in – parallel out*;
  - ✓ *Rotate right*;
  - ✓ *Rotate left*;



## Circuitos Sequenciais

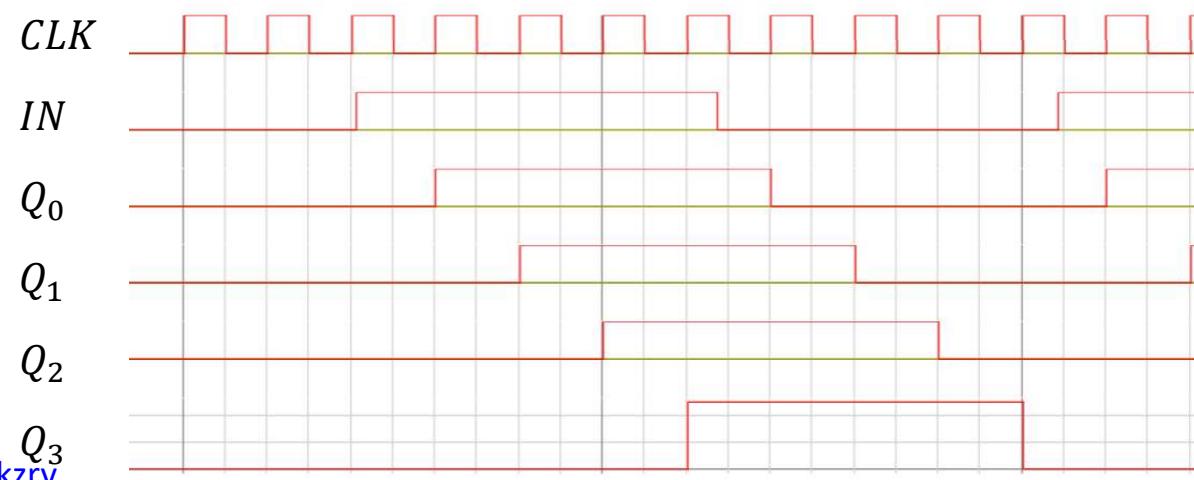
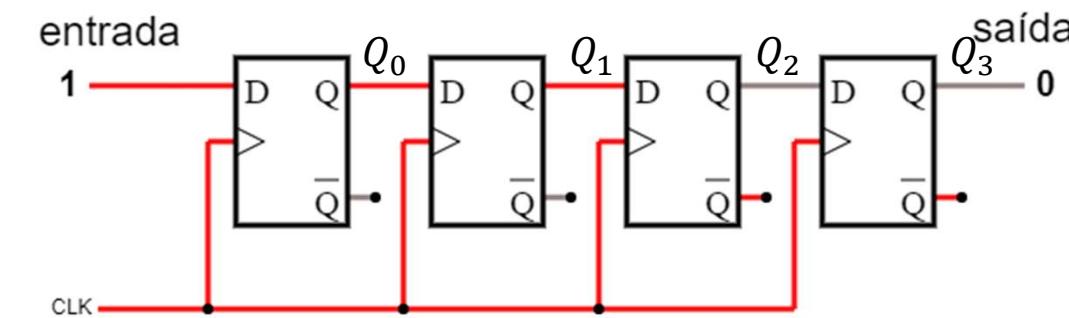
### Registrador *serial in – serial out*

- Registrador do tipo *serial in – serial out* transfere bit a bit de uma sequência de bits da entrada para a saída:
- O seguinte circuito utiliza *flip-flops D* para realizar esta lógica:



- A entrada  $D$  do  $FF_0$  será transferida para a saída  $Q_0$  na borda de subida do *clock*;
- A informação  $Q_0$  será transferida para a saída  $Q_1$  na próxima borda;
- A informação  $Q_1$  será transferida para a saída  $Q_2$  na próxima borda; e assim sucessivamente;
- O último bit transferido fica retido em  $Q_3$ ;
- Uma onda de deslocamentos ocorre a medida que um novo bit é introduzido em Q.

## Circuitos Sequenciais Registrador *serial in – serial out*

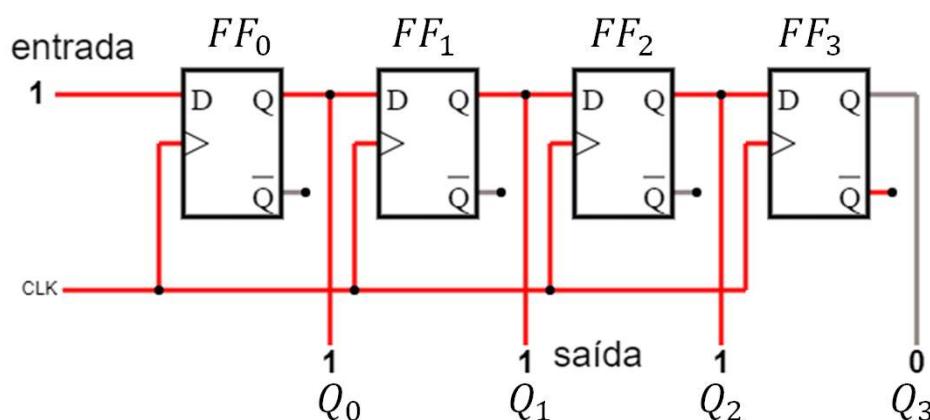


<https://tinyurl.com/yz7ykzry>

## Circuitos Sequenciais

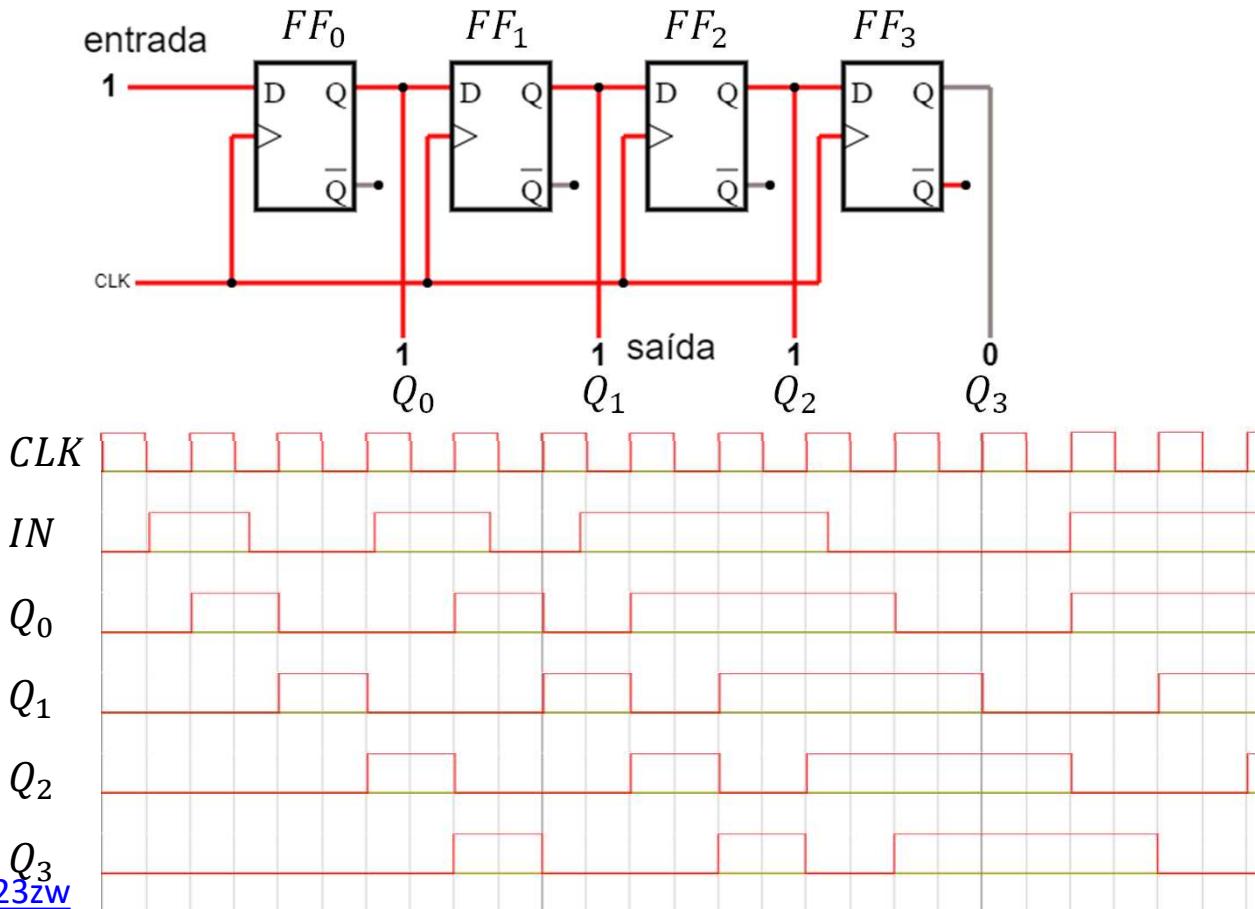
### Registrador *serial in – parallel out*

- Registrador do tipo *serial in – parallel out* transfere bit a bit de uma sequência de bits da entrada para as saídas:
- O seguinte circuito utiliza *flip-flops D* para realizar esta lógica:



- A entrada  $D$  do  $FF_0$  será transferida para a saída  $Q_0$  na borda de subida do *clock*;
- A informação  $Q_0$  será transferida para a saída  $Q_1$  na próxima borda;
- A informação  $Q_1$  será transferida para a saída  $Q_2$  na próxima borda; e
- Assim sucessivamente;
- O último bit transferido fica retido em  $Q_3$ .

## Circuitos Sequenciais Registrador *serial in – parallel out*

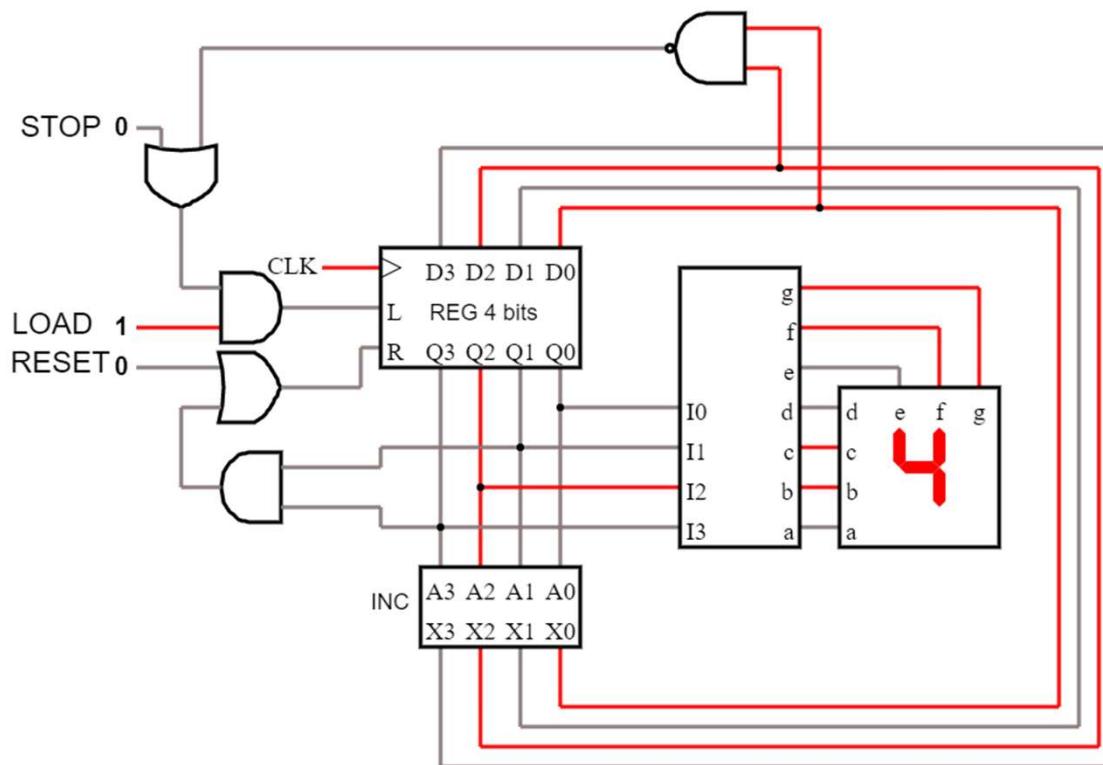


<https://tinyurl.com/yfmx23zw>

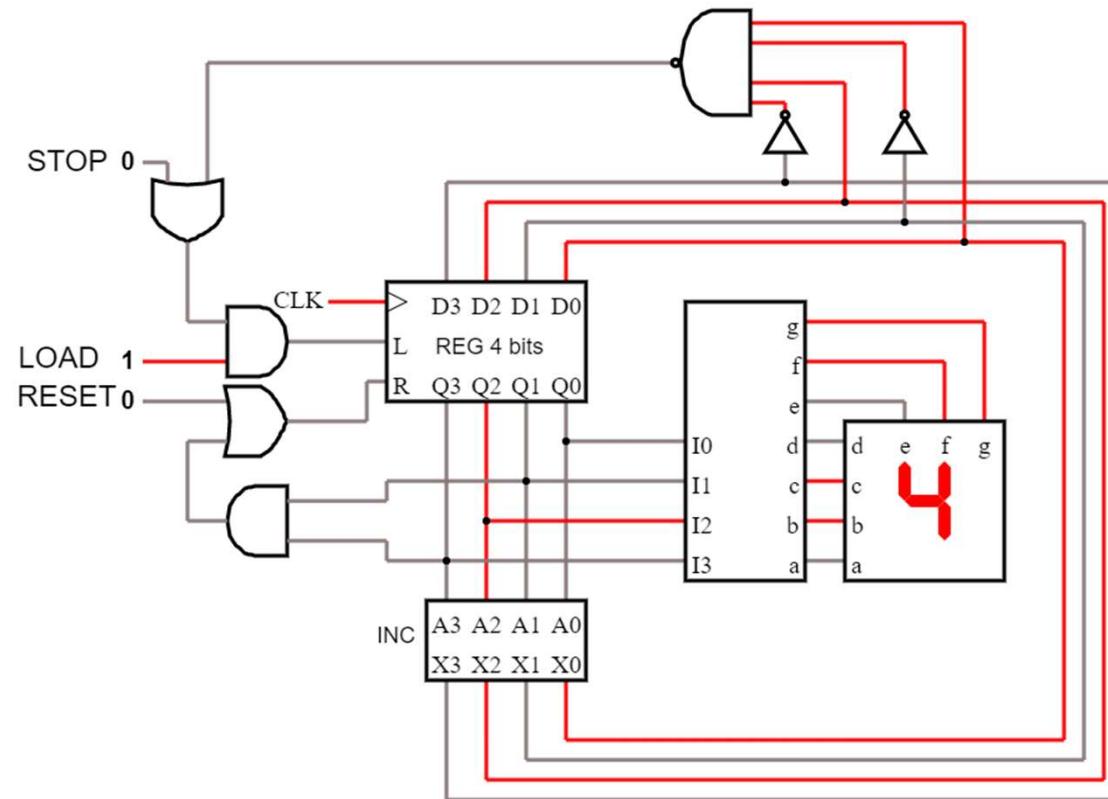
## Circuitos Sequenciais Registradores

- i. Introduzir pequenas modificações no diagrama do contador de 0 a 9, para incluir uma parada em 4:

## Circuitos Sequenciais Registradores



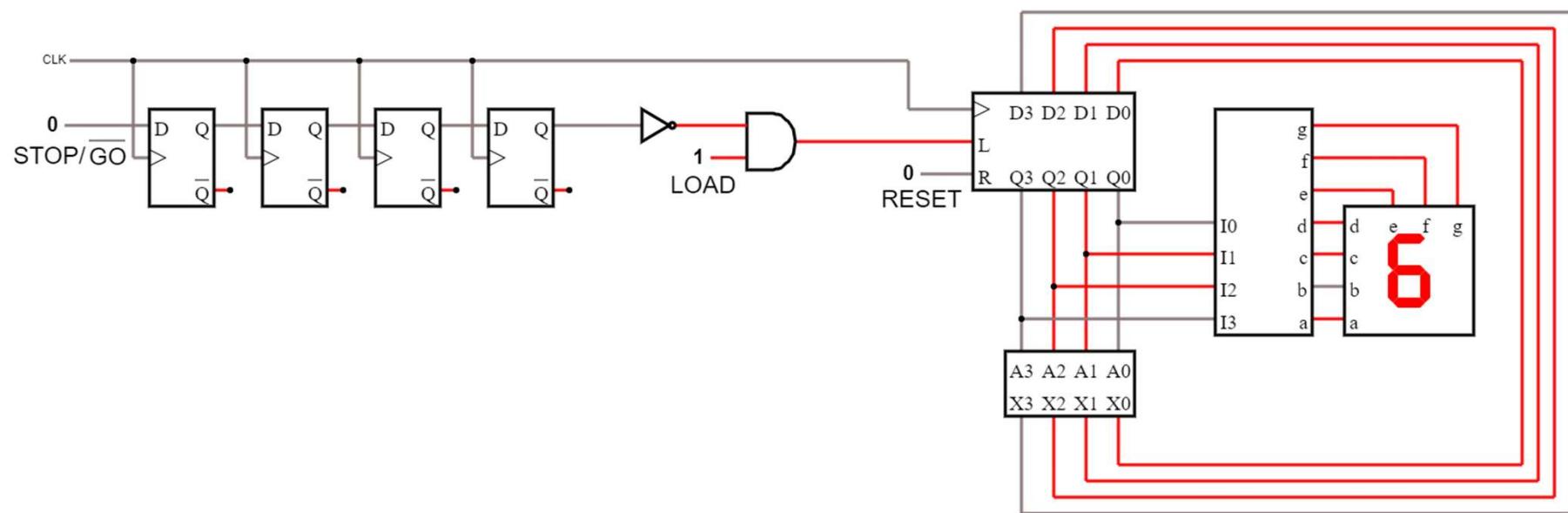
## Circuitos Sequenciais Registradores



## Circuitos Sequenciais Registradores

- ii. Adaptar o incrementador de 4 bits para incluir uma entrada que congela/retoma a contagem após 4 pulsos do *clock*:

# Circuitos Sequenciais Registradores



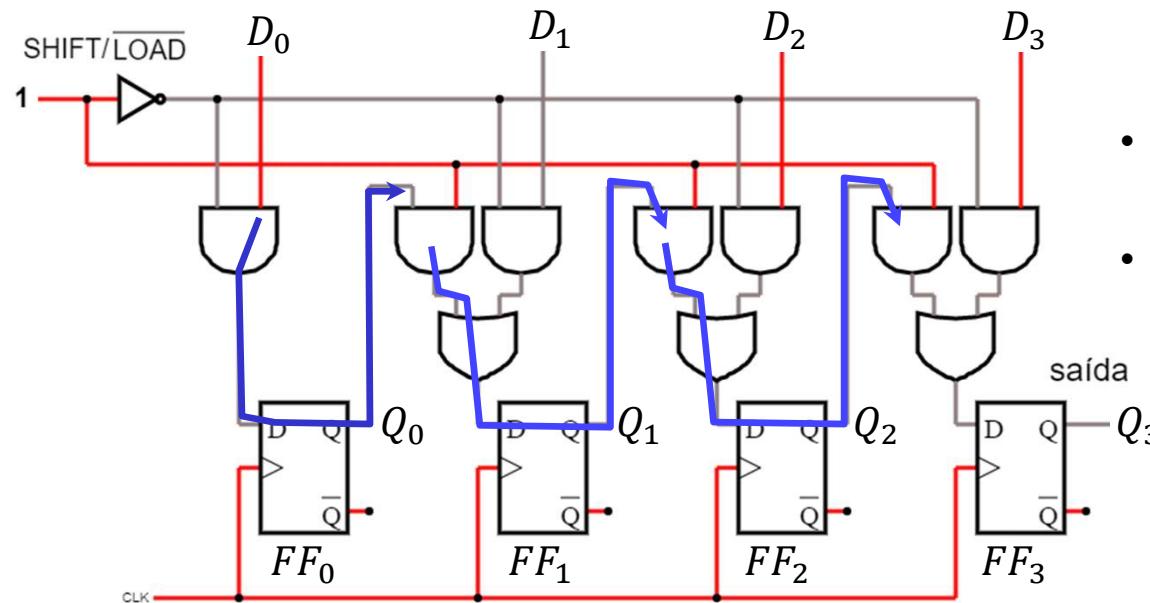
<https://tinyurl.com/ye5f6p9o>

20

## Circuitos Sequenciais

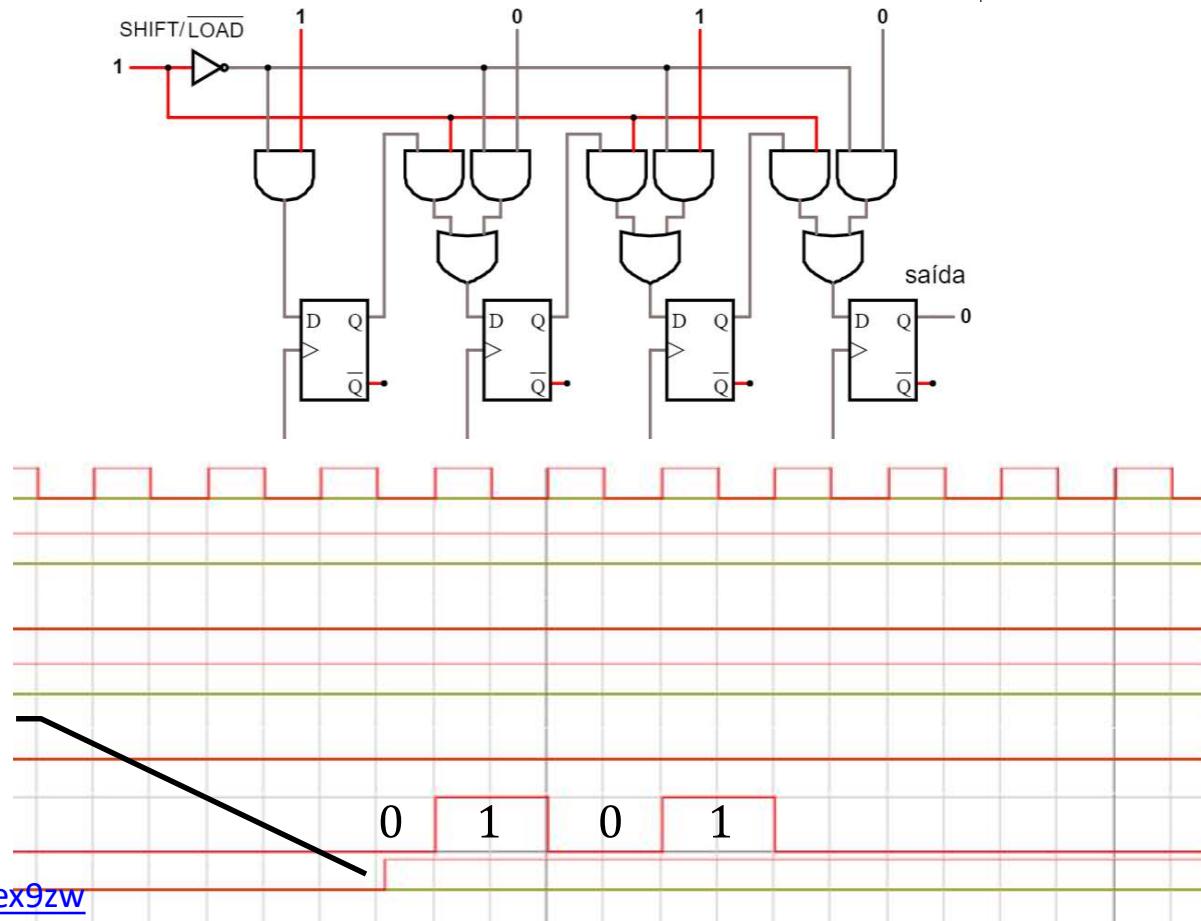
### Registrador *parallel in – serial out*

- Registrador do tipo *parallel in – serial out* transfere bit a bit a partir de uma entrada paralela de bits:
- O seguinte circuito utiliza *flip-flops D* para realizar esta lógica:



- A entrada  $SHIFT/LOAD$  em 0 habilita as entradas  $D_{0-3}$  para os *flip-flops*;
- A entrada  $SHIFT/LOAD$  em 1 desabilita as entradas  $D_{0-3}$  e habilita a conexão em cascata dos *flip-flops*;  
Dando início assim ao processo de *shift*.

## Circuitos Sequenciais Registrador *parallel in – serial out*



<https://tinyurl.com/yjxex9zw>

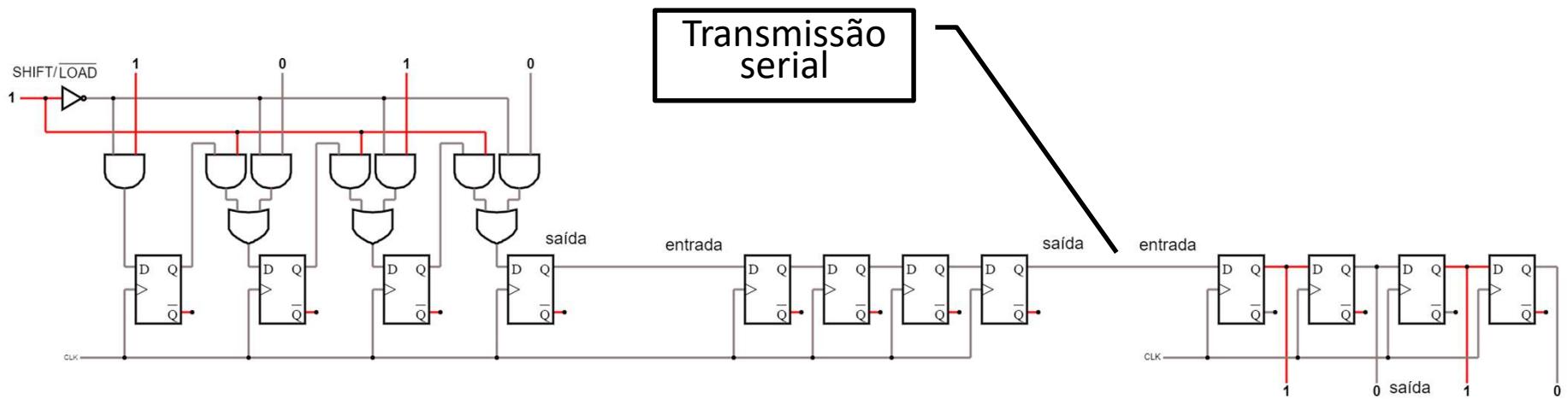
## Circuitos Sequenciais Registrador de deslocamento

iii. Implementar um circuito para transmissão serial:

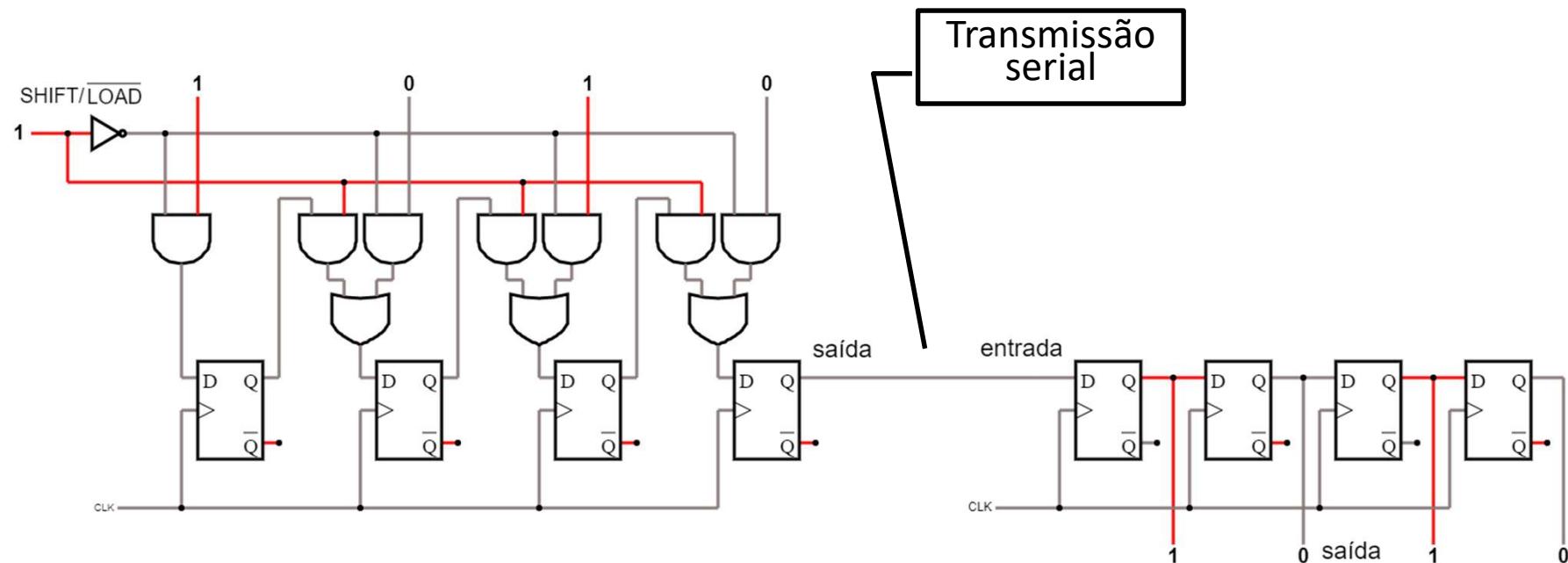
- ✓ Em uma ponta os dados são armazenados em registrador de 4 *bits*;
- ✓ Uma vez iniciada a comunicação, os bits são transferidos serialmente por uma única linha para a outra ponta;
- ✓ Na recepção os bits são novamente reagrupados;

# Circuitos Sequenciais

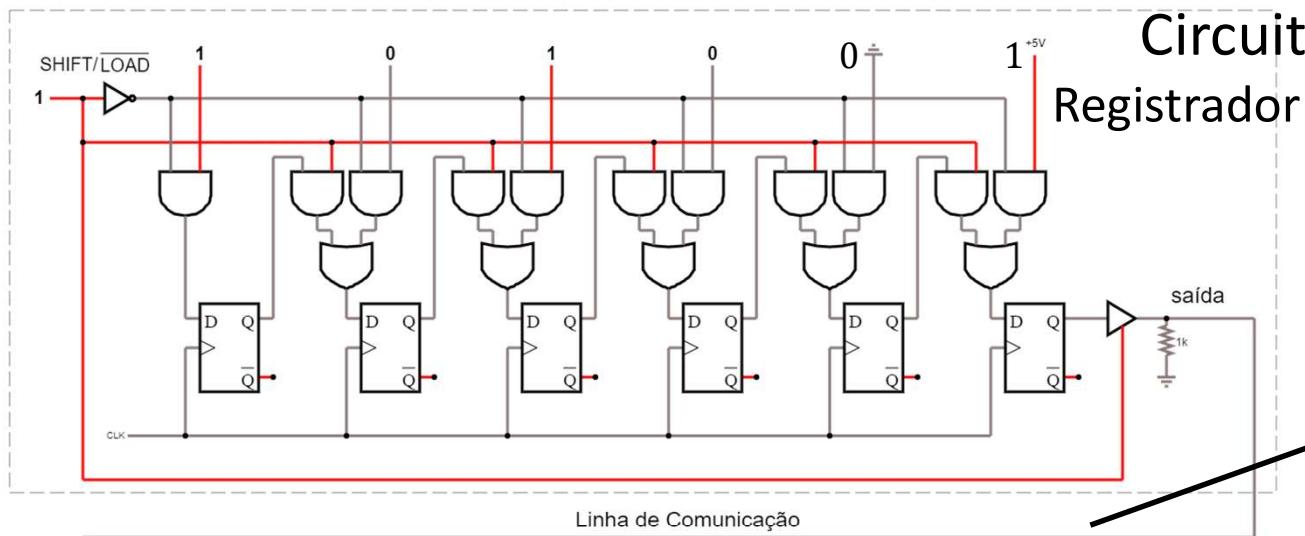
## Registrador de deslocamento



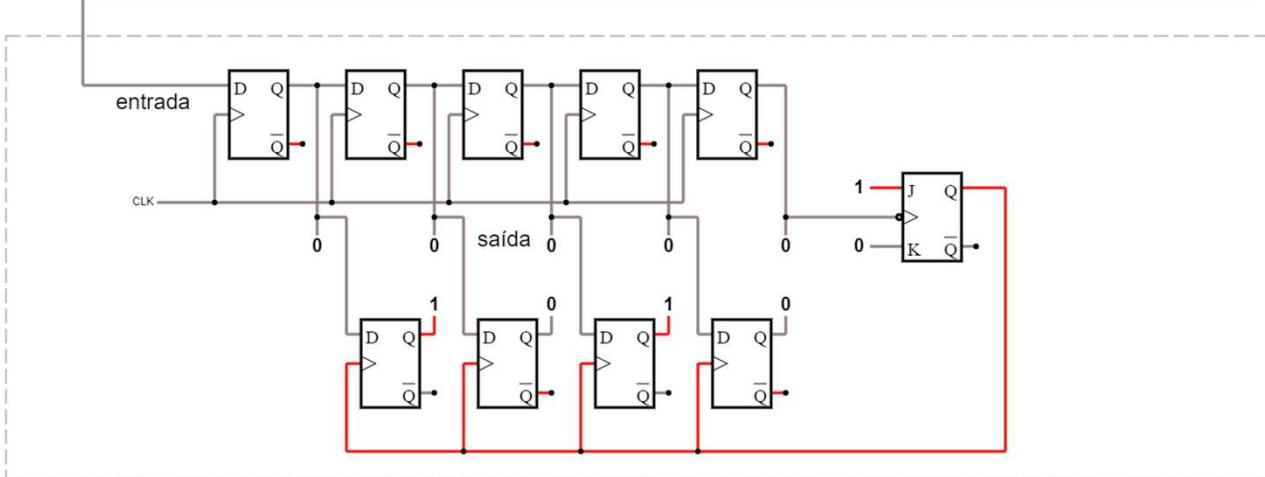
## Circuitos Sequenciais Registrador de deslocamento



## Circuitos Sequenciais Registrador de deslocamento

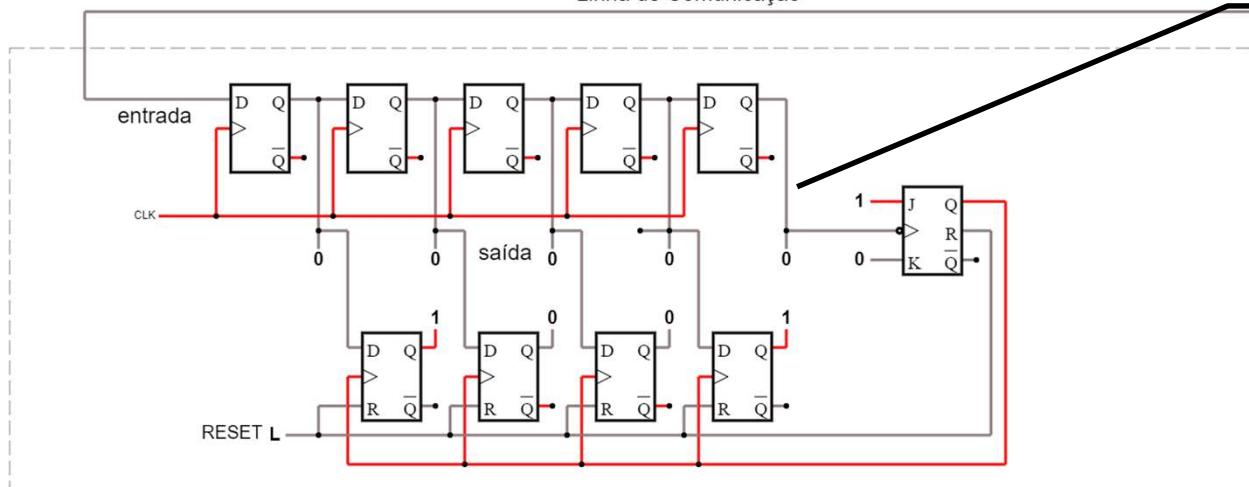
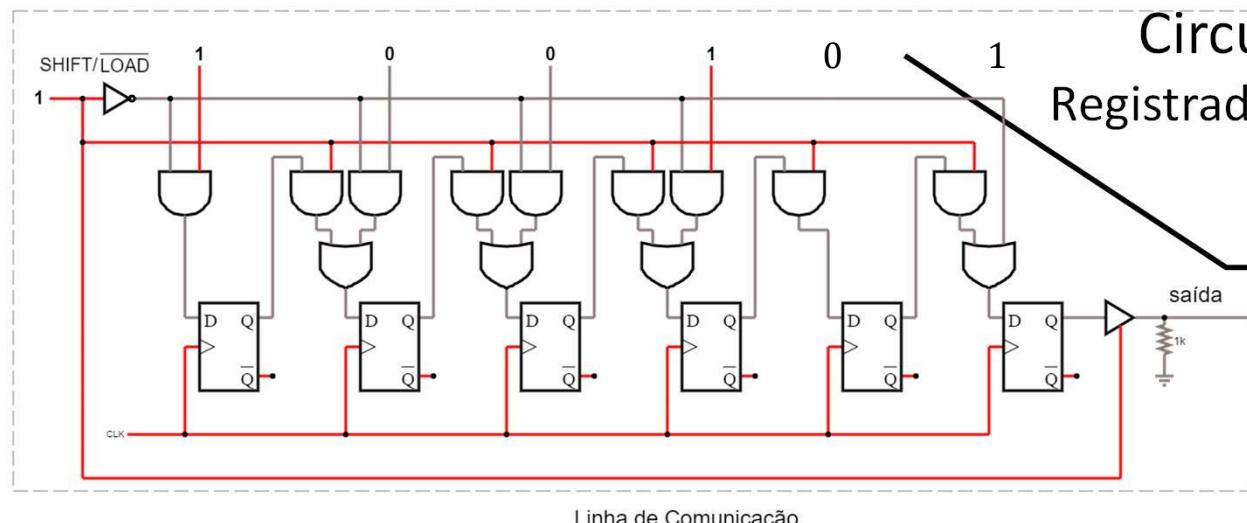


Transmissão  
serial



<https://tinyurl.com/ygfct52j>

## Circuitos Sequenciais Registrador de deslocamento



- São adicionados dois bits ao grupo;
- O 01 indica o início da transmissão;

- Quando os dois bits alcançarem este estágio da transmissão, o FF JK mudará seu estado para 1;
- É acionada a transferência dos 4 bits de dados para o registrador abaixo;
- Um botão de *RESET* permite habilitar novamente o conjunto para um novo pacote de bits.

<https://tinyurl.com/yehohd6d>

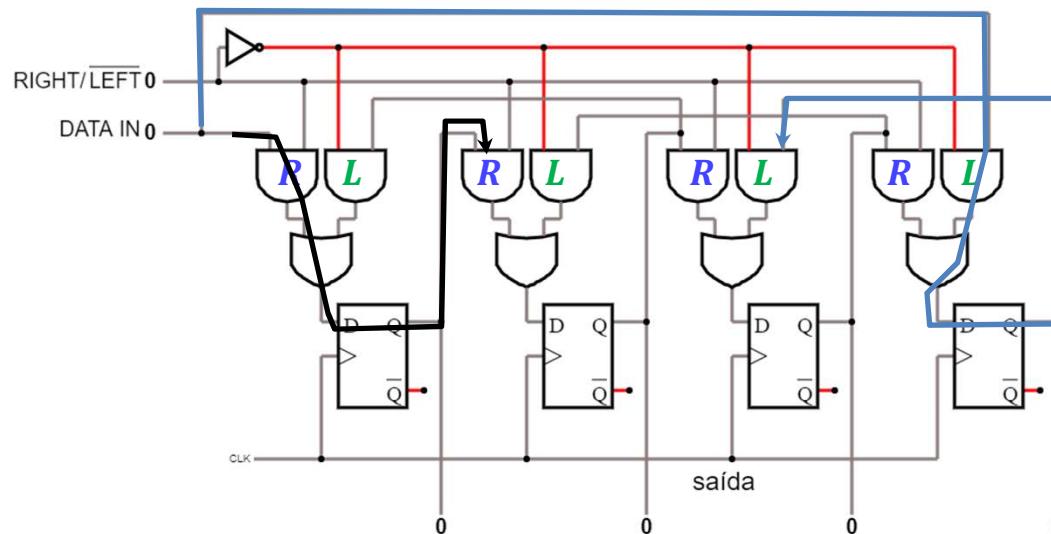
## Circuitos Sequenciais

### Registrador de deslocamento bidirecional

- Um registrador de deslocamento bidirecional permite o movimento dos bits tanto para a esquerda quanto para a direita;
- Ele pode ser implementado com uma entrada binária que habilita a transferência de um bit de dados de um estágio para outro à direita ou à esquerda.

# Circuitos Sequenciais

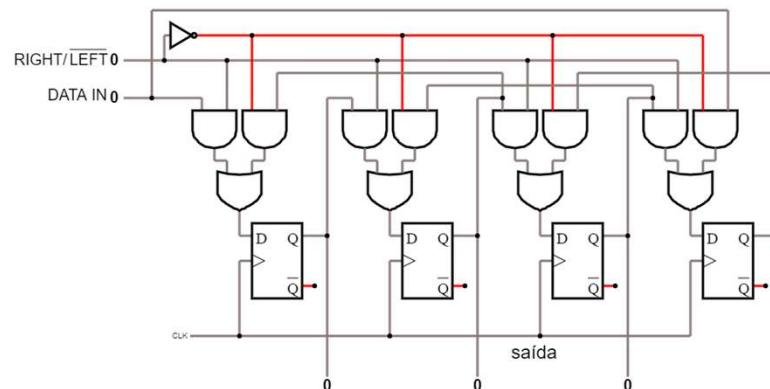
## Registrador de deslocamento bidirecional



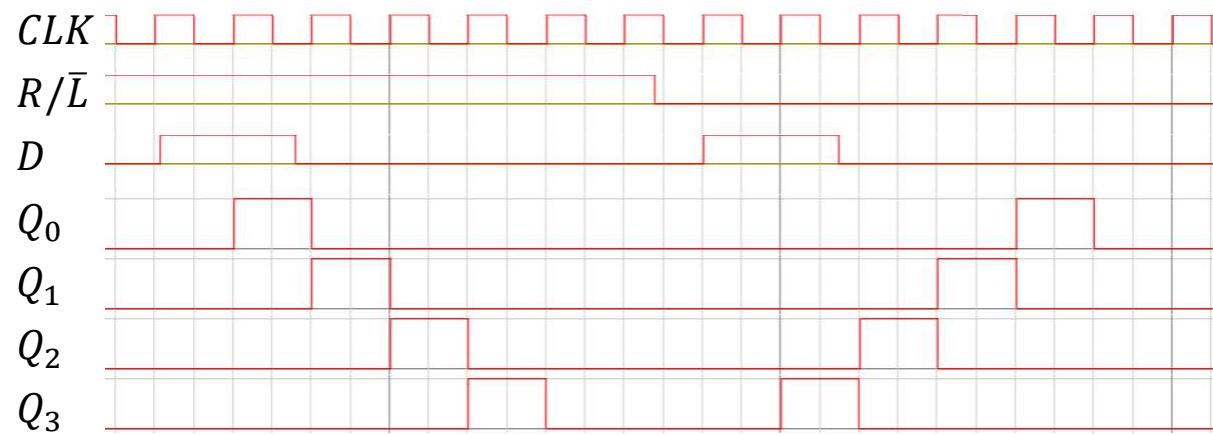
- Quando a entrada de controle  $RIGHT/\overline{LEFT} == 1$ , as portas **R** permitirão que o estado  $Q$  de cada *flip-flop* seja passado para a entrada  $D$  do próximo *flip-flop*.
- Quando ocorre um pulso de *clock*, os bits de dados serão deslocados para a direita.
- Quando a entrada de controle  $RIGHT/\overline{LEFT} == 0$  as portas **L** permitirão que a saída  $Q$  de cada *flip-flop* seja transferida para a entrada  $D$  do *flip-flop* anterior.
- Quando ocorre um pulso de *clock*, os bits de dados serão deslocados uma posição para a esquerda.

# Circuitos Sequenciais

## Registrador de deslocamento bidirecional



<https://tinyurl.com/yemjrtcy>



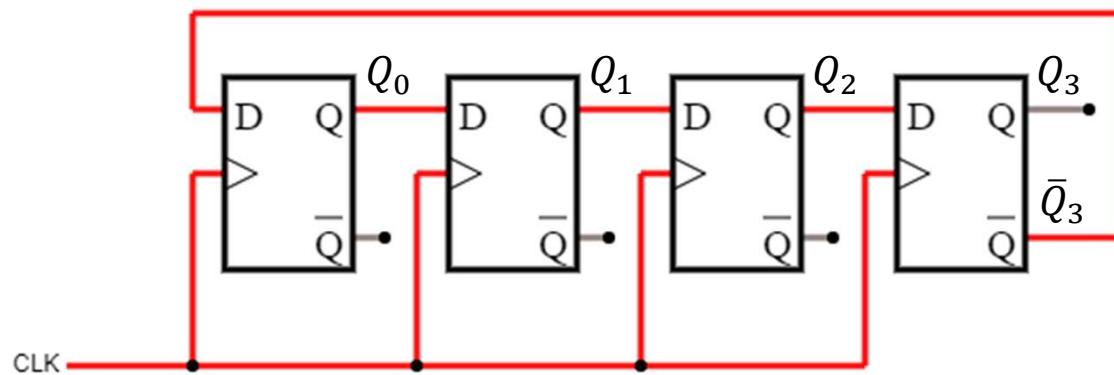
## Circuitos Sequenciais

### Registrador de deslocamento contador

- Um registrador de deslocamento contador é basicamente um registrador de deslocamento, onde a saída serial é conectada na entrada serial;
- Esta configuração produz uma sequência especial de valores;
- Os registradores contadores mais comuns são:
  - ✓ Contador *Johnson*; e
  - ✓ Contador em anel.

## Circuitos Sequenciais Contador Johnson

- Em um contador Johnson, a saída  $\bar{Q}$  do último *flip-flop* é conectada em *feedback* na entrada *D* do primeiro *flip-flop*;
- É possível implementar com outros tipos de *flip-flops*;



## Circuitos Sequenciais

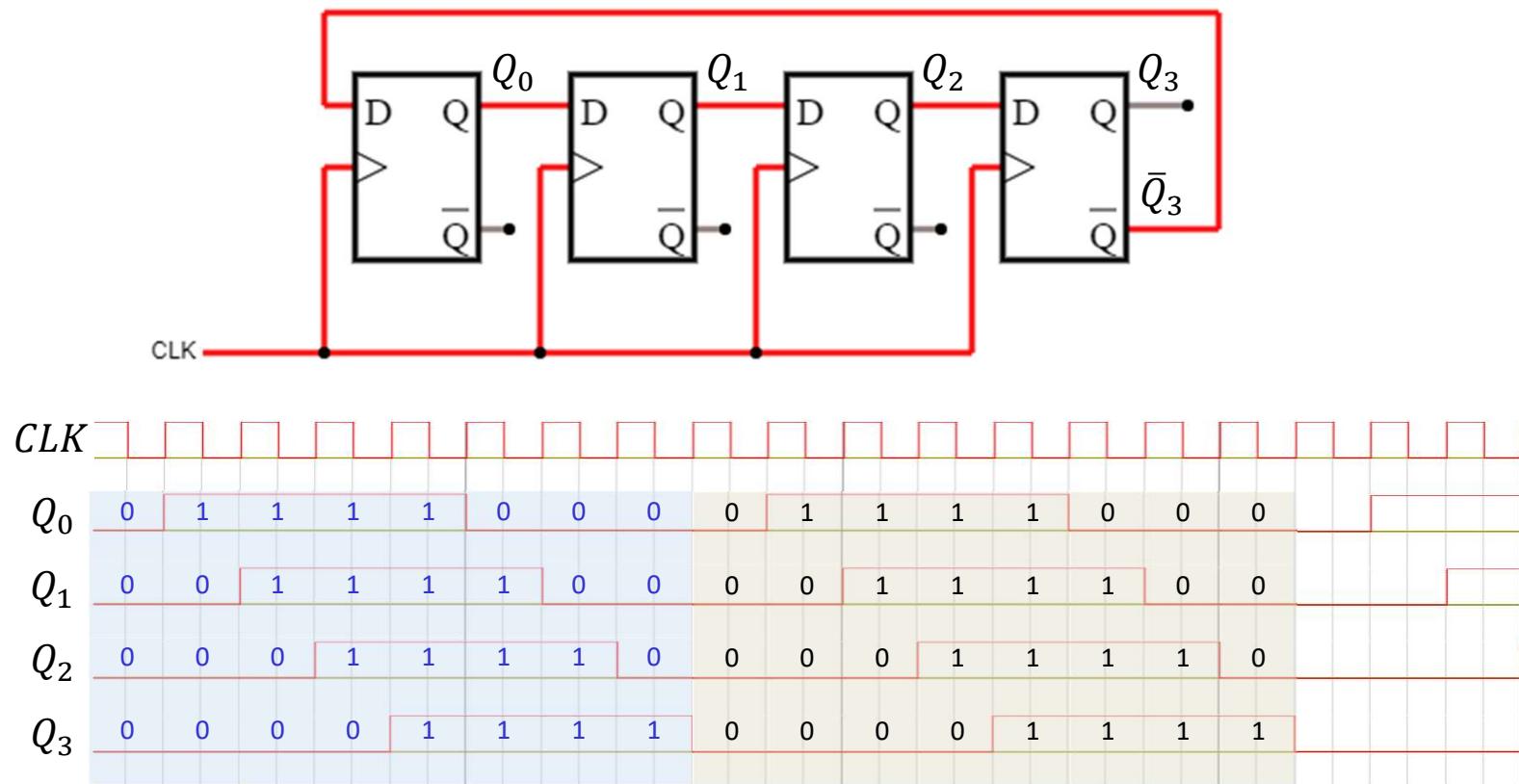
### Contador Johnson

- Se o contador iniciar em 0, este arranjo de feedback produzirá a seguinte sequência clássica de estados:

<i>Clock Pulse</i>	$Q_0$	$Q_1$	$Q_2$	$Q_3$
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	1	1	1	1
5	0	1	1	1
6	0	0	1	1
7	0	0	0	1

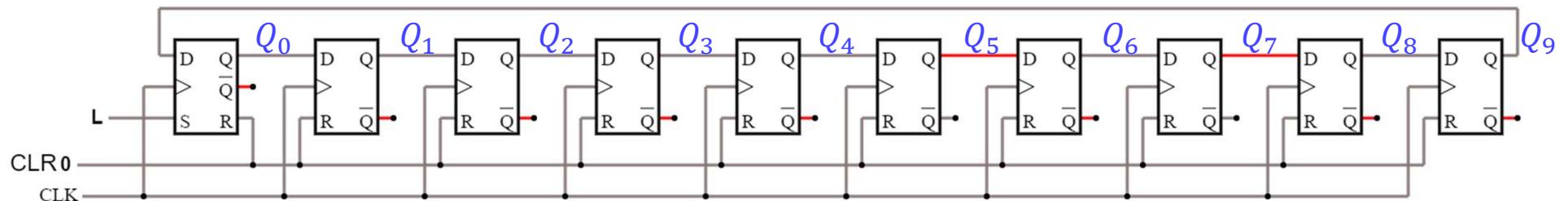
- Observe que a sequência de 4 bits tem um total de oito estados, ou padrões de bits, e a sequência de 5 bits tem um total de dez estados;
- Em geral, um contador *Johnson* produzirá um módulo de  $2 \times n$ , onde  $n$  é o número de estágios do contador.

## Circuitos Sequenciais Contador Johnson



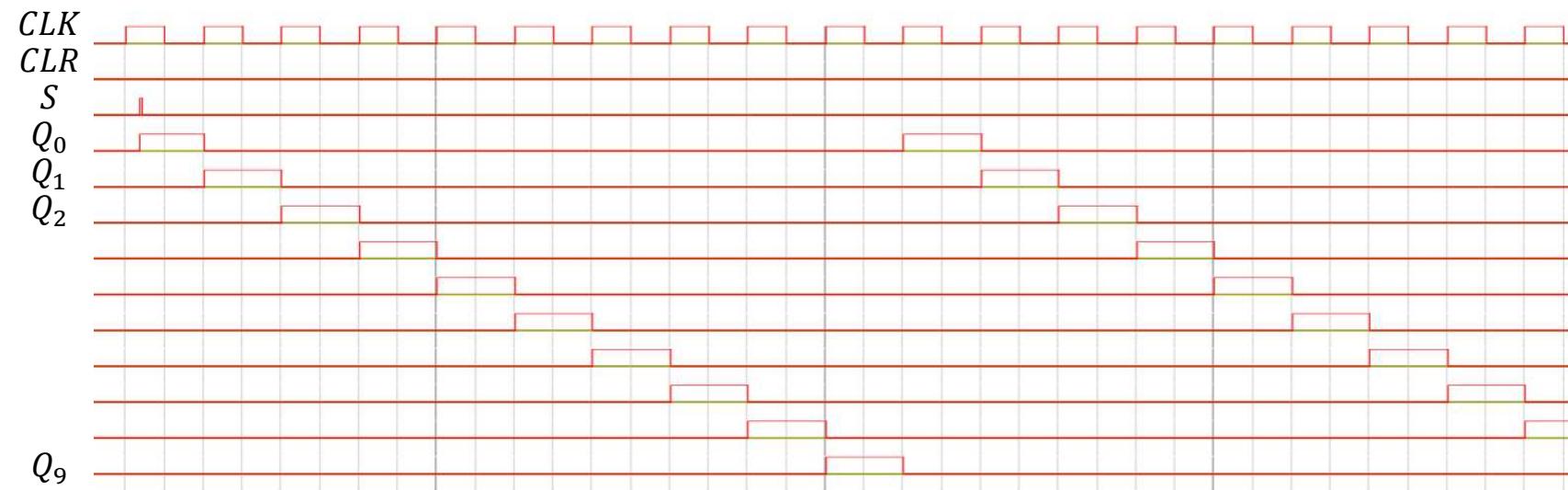
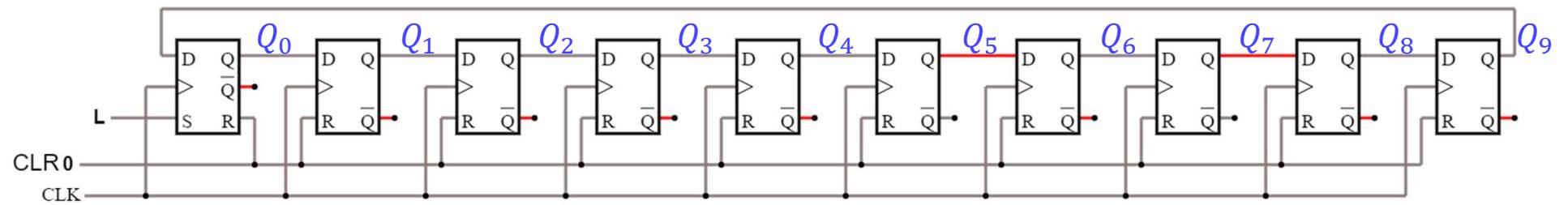
## Circuitos Sequenciais Contador em anel

- Um contador em anel é basicamente um registrador de deslocamento circular;
- A saída  $Q$  do último *flip-flop* é conectada à entrada  $D$  do primeiro;

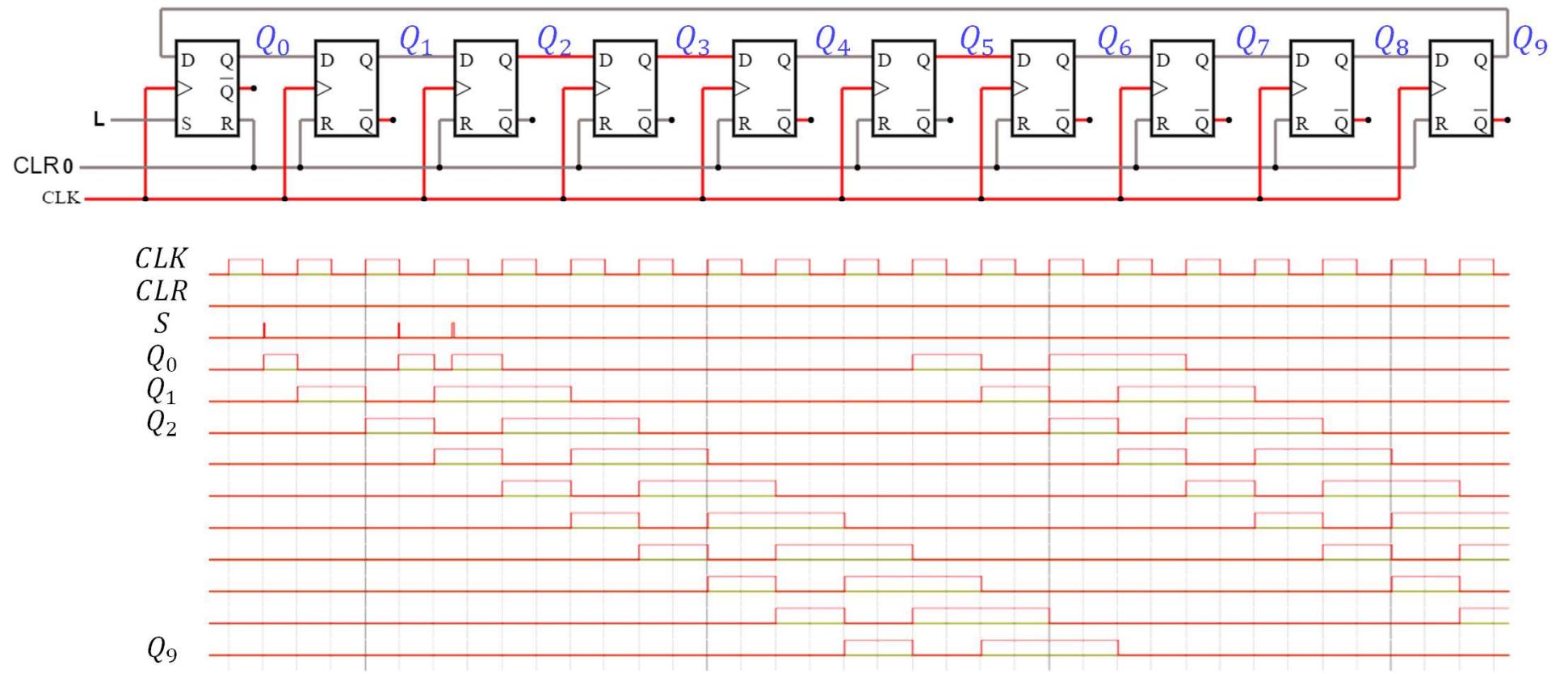


- A configuração mais comum é a de um único *bit* 1 circulando enquanto existirem pulsos do clock;
- A entrada  $S$  do primeiro *flip-flop* serve para introduzir o bit.

## Circuitos Sequenciais Contador em anel



## Circuitos Sequenciais Contador em anel



## Sumário

- 1. Revisão – Sistemas de Numeração
- 2. Revisão – Representação de Dados
- 3. Revisão – Operações com Binários
- 4. Álgebra Booleana
- 5. Simplificação de Expressões
- 6. Mapa de Karnaugh
- 7. Elementos Lógicos Universais
- 8. Circuitos Combinacionais
- 9. Circuitos Sequenciais
  - 1. Latches
  - 2. Flip-Flop
  - 3. Registradores
  - 4. Contadores
  - 5. Máquina de Estados
  - 6. Memória RAM

## Circuitos Sequenciais Contadores

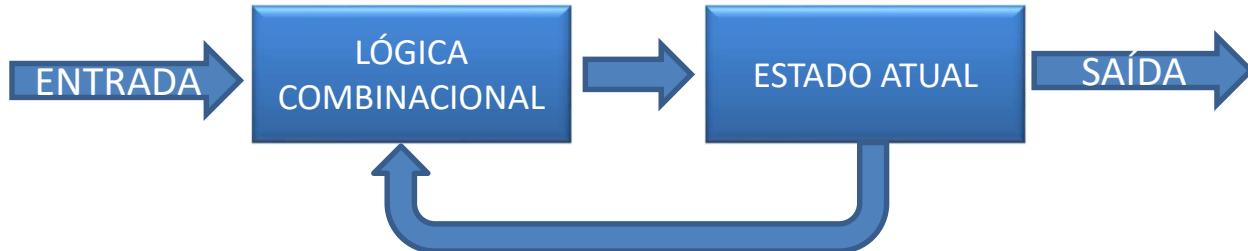
- Uma máquina de estados é um circuito sequencial com um número limitado (finito) de estados que ocorrem em uma determinada ordem;
- Um contador é um exemplo de máquina de estado, onde o número de estados é denominado módulo;
- Dois tipos básicos de máquinas de estado são: modelo de *Moore* e o modelo de *Mealy*;
- A máquina de estado de *Moore* é aquela em que as saídas dependem apenas do estado atual;
- A máquina de estado de *Mealy* é aquela em que as saídas dependem do estado atual e das entradas;
- Os dois modelos possuem uma entrada de *clock* que não é considerada uma entrada de controle.

## Circuitos Sequenciais Contadores

Máquina de  
*Moore*

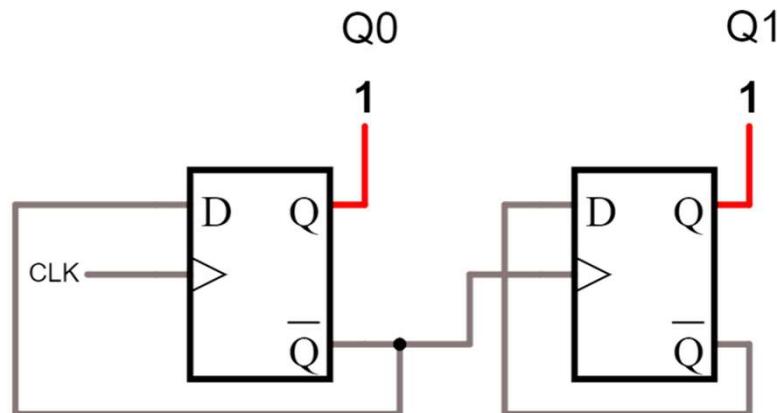


Máquina de  
*Mealy*

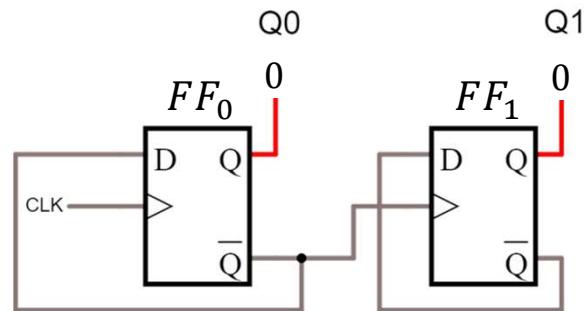


## Circuitos Sequenciais Contadores assíncronos

- O termo assíncrono refere-se a eventos que não possuem uma relação fixa de tempo entre si;
- Um contador assíncrono é aquele em que os *flip-flops* (*FF*) que integram o contador não mudam de estado exatamente ao mesmo tempo, pois não compartilham o mesmo pulso de *clock*;
- O diagrama a seguir apresenta um contador assíncrono de 2 bits:

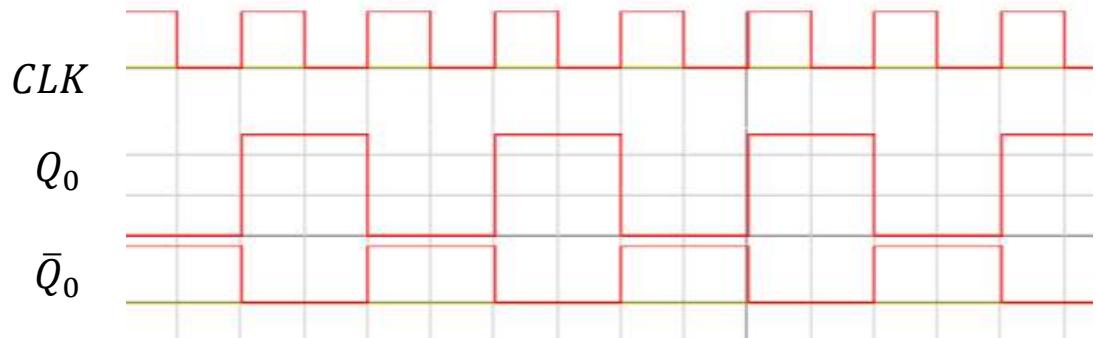
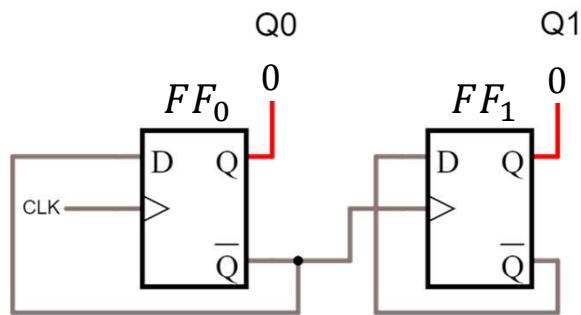


## Circuitos Sequenciais Contadores assíncronos



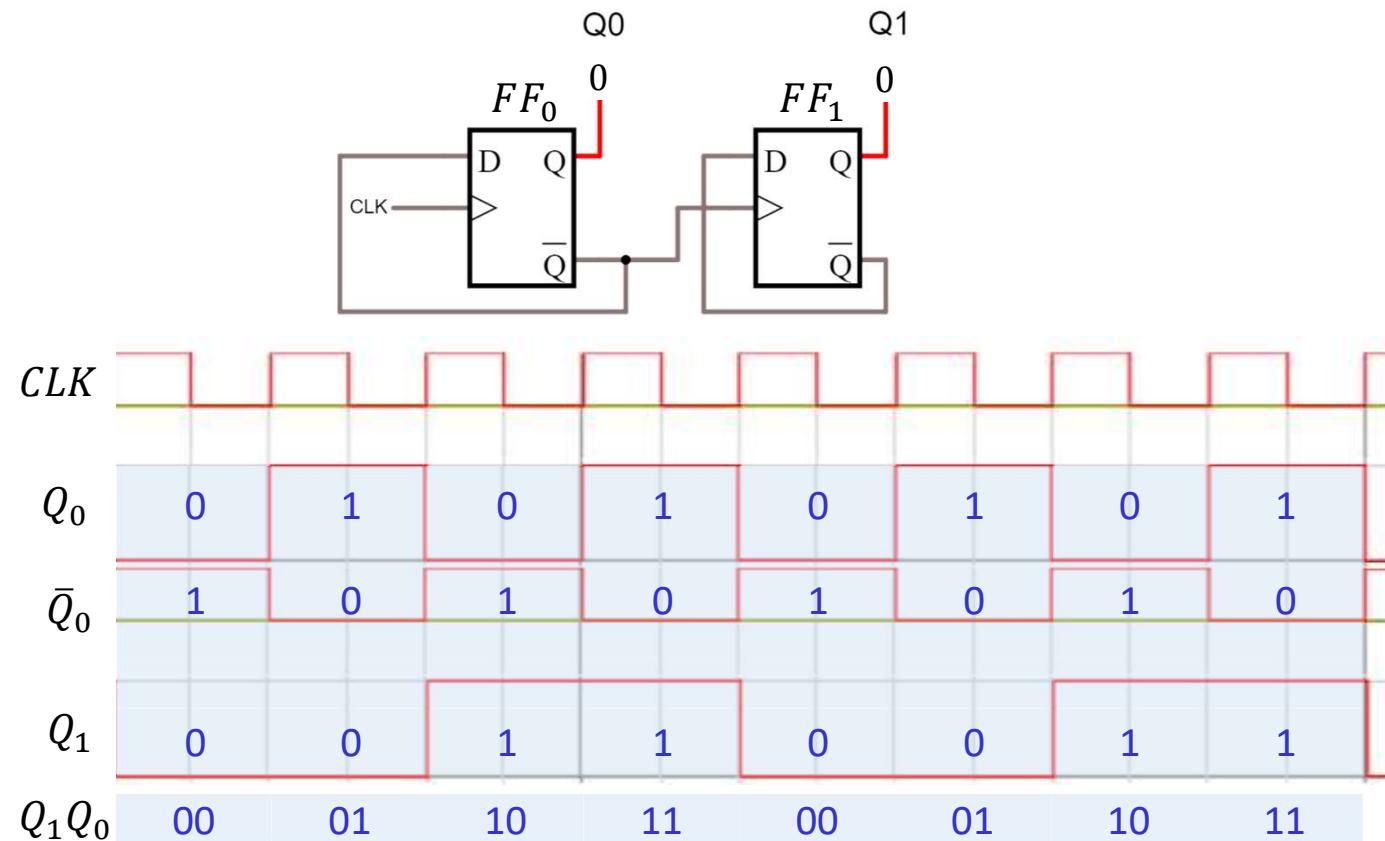
- Para compreender o funcionamento deste circuito, iremos analisar o comportamento do  $FF_0$  para alguns pulsos do clock;
- Inicialmente  $Q_0 = 0$  e  $\bar{Q}_0 = 1$ , realimentando  $D_0 = \bar{Q}_0 = 1$ .
- Na próxima borda do *clock* teremos  $Q_0 = 1$  e  $\bar{Q}_0 = 0$ , realimentando  $D_0 = \bar{Q}_0 = 0$ ;
- Na próxima borda do *clock* teremos  $Q_0 = 0$  e  $\bar{Q}_0 = 1$ , realimentando  $D_0 = \bar{Q}_0 = 1$ ;
- A cada borda de subida do *clock* a saída  $Q_0$  vai alternando de valor;
- O seguinte gráfico ilustra o comportamento:

## Circuitos Sequenciais Contadores assíncronos

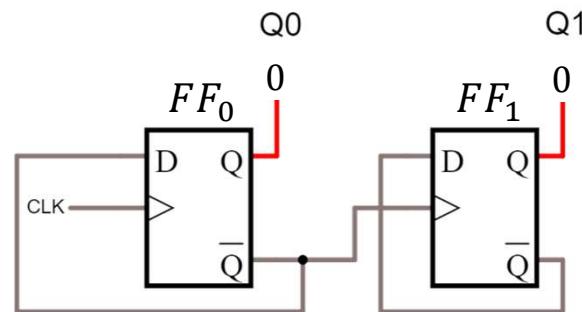


- Iremos analisar o comportamento do  $FF_1$  em função dos pulsos recebidos a partir de  $\bar{Q}_0$ ;
- Sempre que  $\bar{Q}_0$  comutar de 0 para 1, o  $FF_1$  irá transferir a entrada  $D_1$  para  $Q_1$ ;
- Inicialmente  $Q_1 = 0$  e  $\bar{Q}_1 = 1$ , realimentando  $D_1 = \bar{Q}_1 = 1$ .
- Na próxima borda de  $\bar{Q}_0$  teremos  $Q_1 = 1$  e  $\bar{Q}_1 = 0$ , realimentando  $D_1 = \bar{Q}_1 = 0$ ;
- Na próxima borda do  $\bar{Q}_0$  teremos  $Q_1 = 0$  e  $\bar{Q}_1 = 1$ , realimentando  $D_1 = \bar{Q}_1 = 1$ ;
- A cada borda de subida do  $\bar{Q}_0$  a saída  $Q_1$  vai alternando de valor;

## Circuitos Sequenciais Contadores assíncronos



## Circuitos Sequenciais Contadores assíncronos



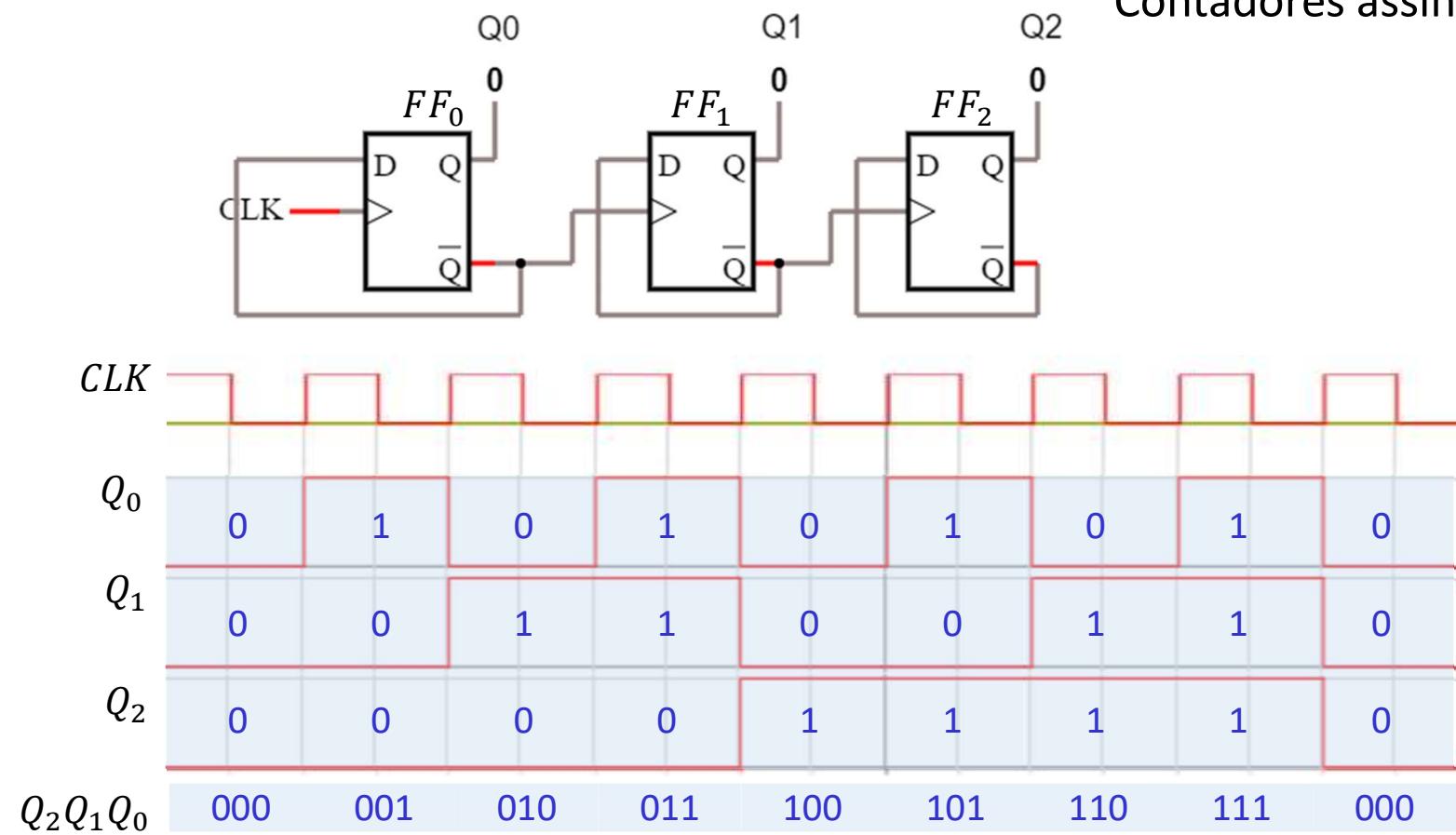
- A sequência de estados pode ser visualizada na seguinte tabela:

<i>Clock Pulse</i>	$Q_1$	$Q_0$
Inicial	0	0
1	0	1
2	1	0
3	1	1
4	0	0

## Circuitos Sequenciais Contadores assíncronos

- Construir um contador assíncrono de 3 *bits*.

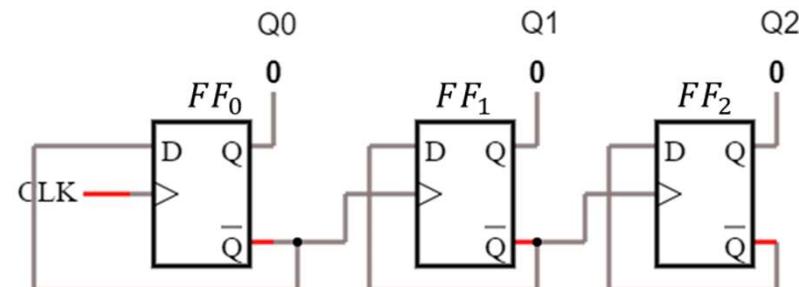
## Circuitos Sequenciais Contadores assíncronos



47

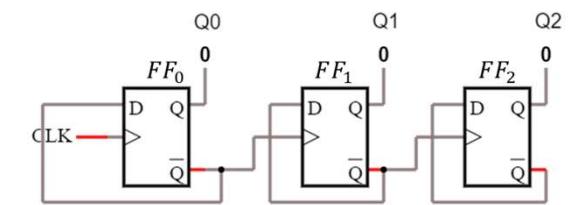
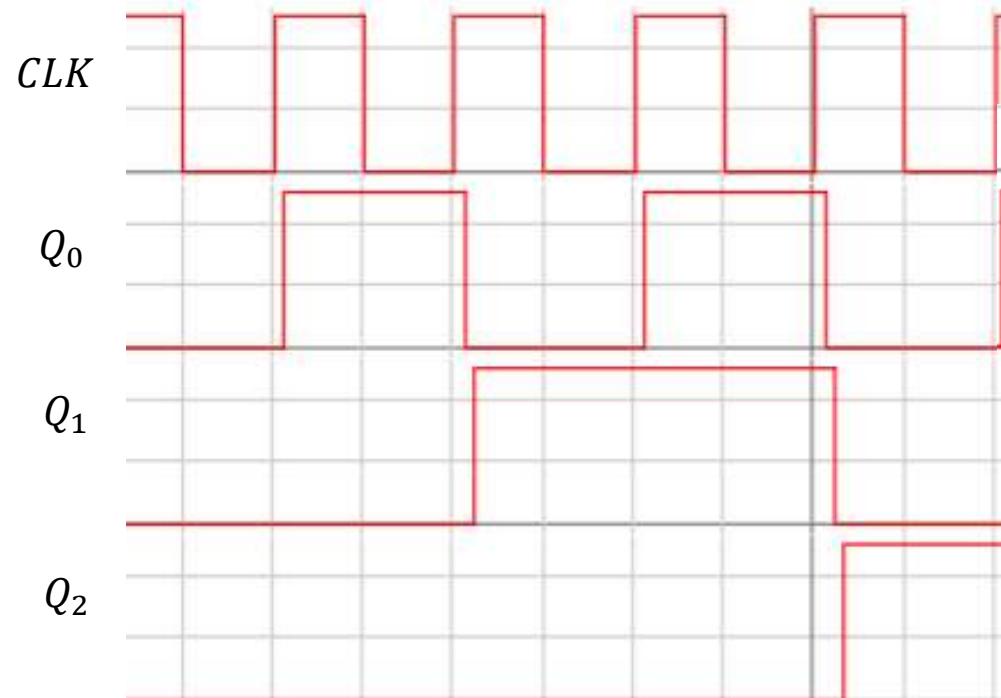
## Circuitos Sequenciais Contadores assíncronos

- Os contadores assíncronos também são chamados de contadores de *ripple*, uma vez que:
  - ✓ O efeito da borda de subida do *clock* afeta apenas o  $FF_0$ ;
  - ✓ O  $FF_1$  não opera perfeitamente sincronizado com o  $FF_0$ ;
  - ✓ O  $FF_1$  irá atuar quando o sinal  $\bar{Q}_0$  pulsar na sua entrada de *clock*;
  - ✓ Isto ocorrerá após o sinal transitar pelo  $FF_0$ ;
  - ✓ Assim, o sinal chegará com um atraso devido ao tempo de propagação ( $t_p$ );



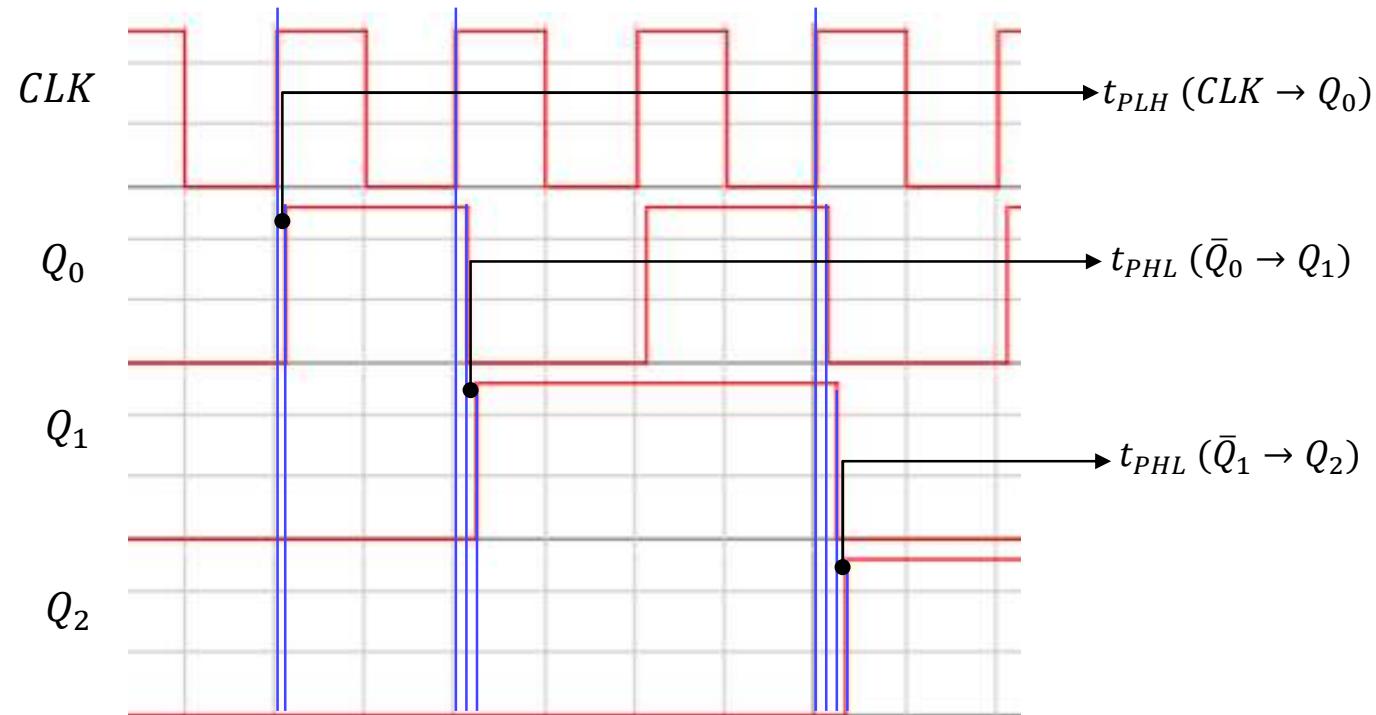
## Circuitos Sequenciais Contadores assíncronos

- Na prática, o gráfico de tempo terá a seguinte forma:



## Circuitos Sequenciais Contadores assíncronos

- Na prática, o gráfico de tempo terá a seguinte forma:



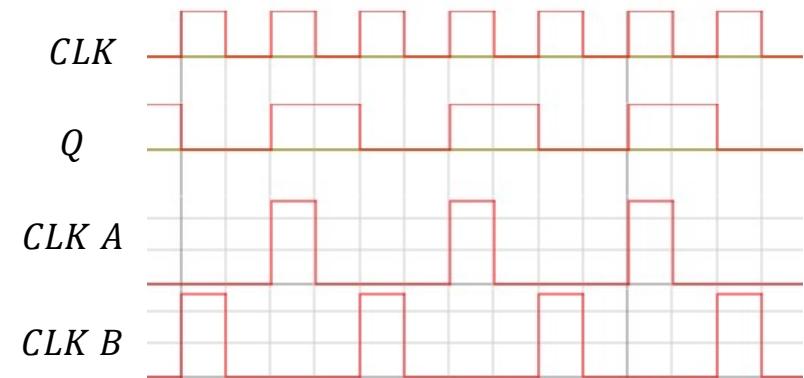
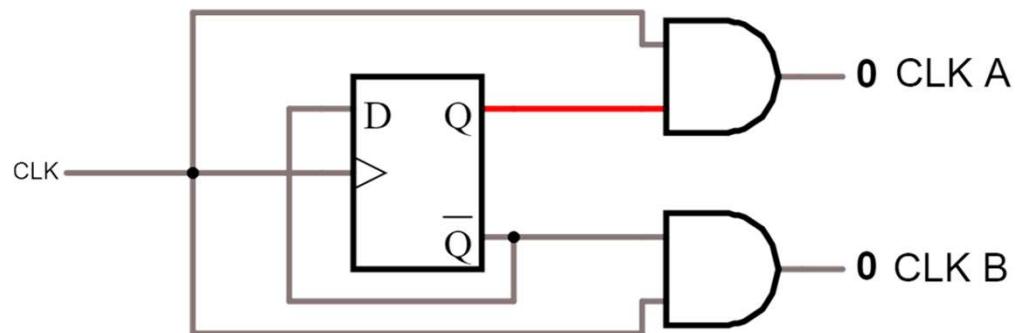
## Circuitos Sequenciais

### Contadores assíncronos

- O atraso cumulativo do contador assíncrono se torna uma grande desvantagem no desenvolvimento de muitos aplicativos, pois limita a taxa na qual o contador pode ser incrementado e consequentemente gera problemas de decodificação;
- Estes problemas de decodificação são denominados *glitch* ou *spike* que são estados fora da sequência predeterminada para o contador;
- O atraso cumulativo máximo em um contador deve ser menor que o período do *clock*;

## Circuitos Sequenciais Contadores assíncronos

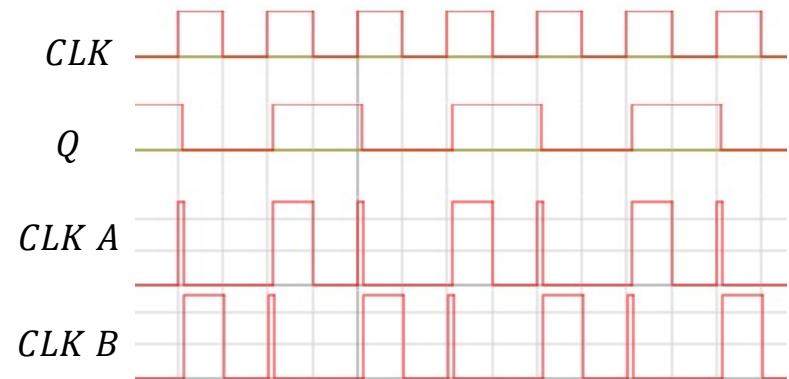
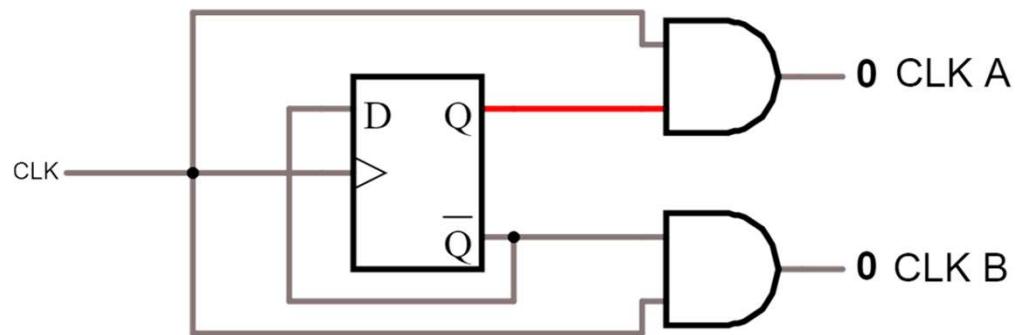
- Considere o seguinte circuito:



- Idealmente,  $CLK A$  será *HIGH* quando  $CLK$  e  $Q$  forem *HIGH*; e
- $CLK B$  será *HIGH* quando  $CLK$  e  $\bar{Q}$  forem *HIGH*;

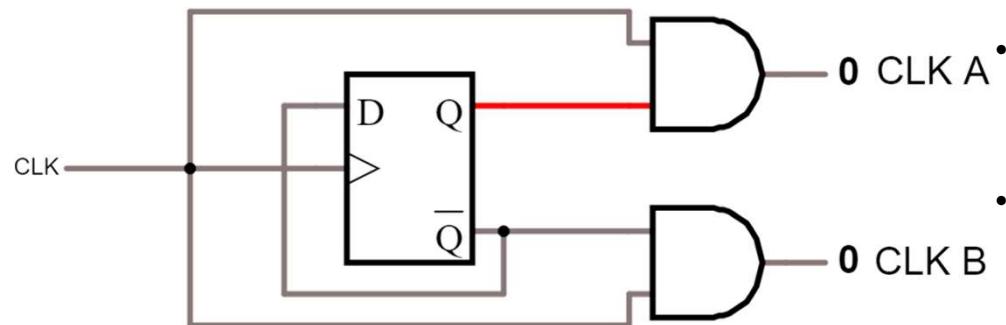
## Circuitos Sequenciais Contadores assíncronos

- Considere o seguinte circuito:

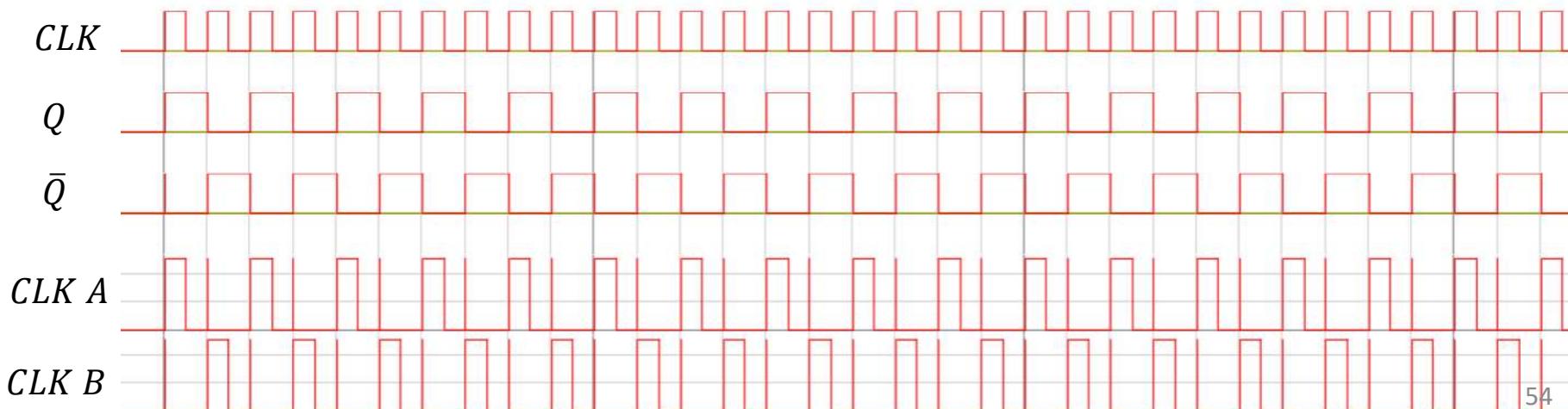


- Porém, na realidade há um atraso na transferência do sinal através do *FF*;
- Gerando os *glitches*, conforme ilustrado no diagrama de tempo.

## Circuitos Sequenciais Contadores assíncronos

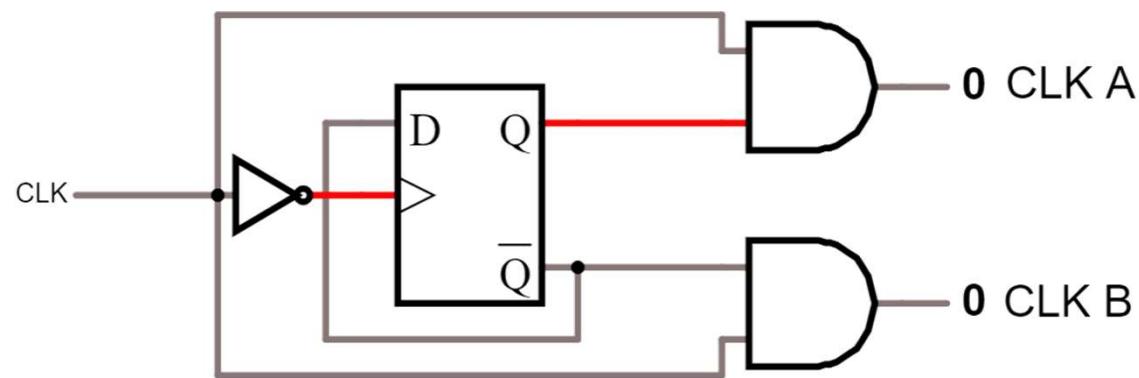


- Analisando o diagrama de tempo gerado para  $CLK A$  e  $CLK B$ , observamos alguns ruídos (*glitches*);
- Enquanto o  $CLK$  muda de *LOW* para *HIGH*, a saída  $Q$  levará um  $t_p$  para mudar de *HIGH* para *LOW*;
- Durante este  $t_p$  ambos serão *HIGH*.

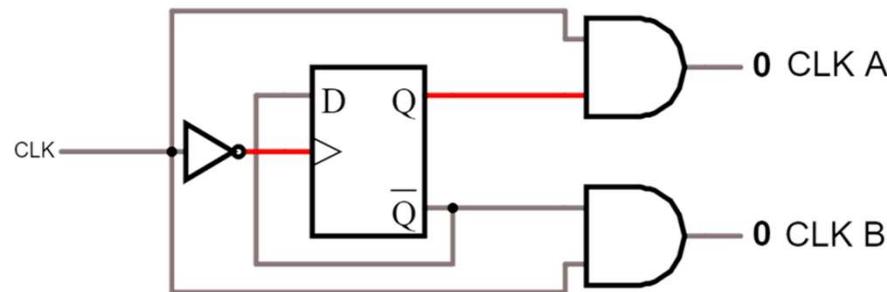


## Circuitos Sequenciais Contadores assíncronos

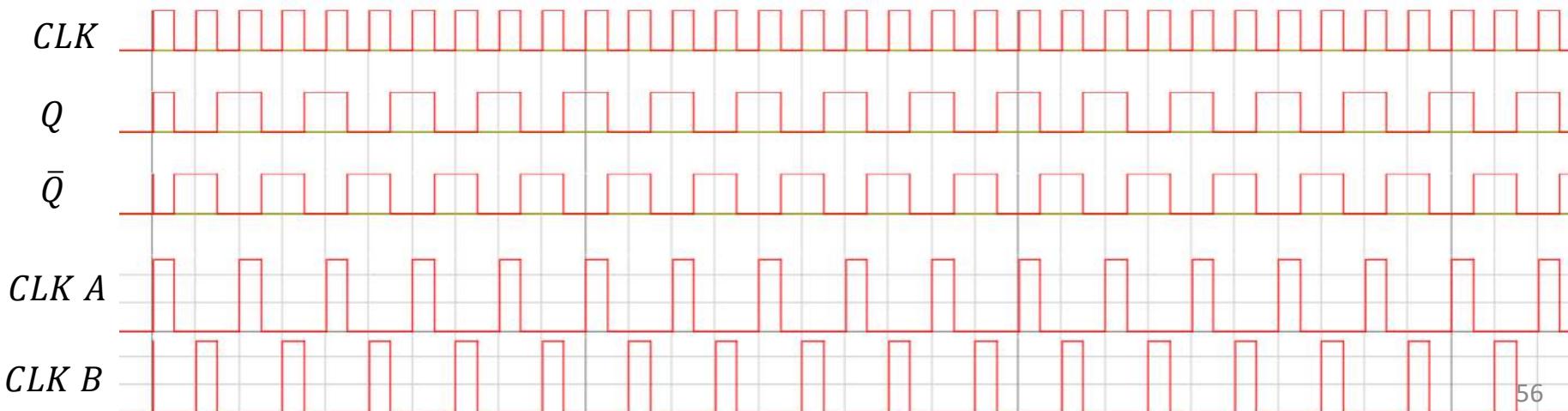
- Nesta configuração, é possível contornar o problema, invertendo a borda de atuação:



## Circuitos Sequenciais Contadores assíncronos



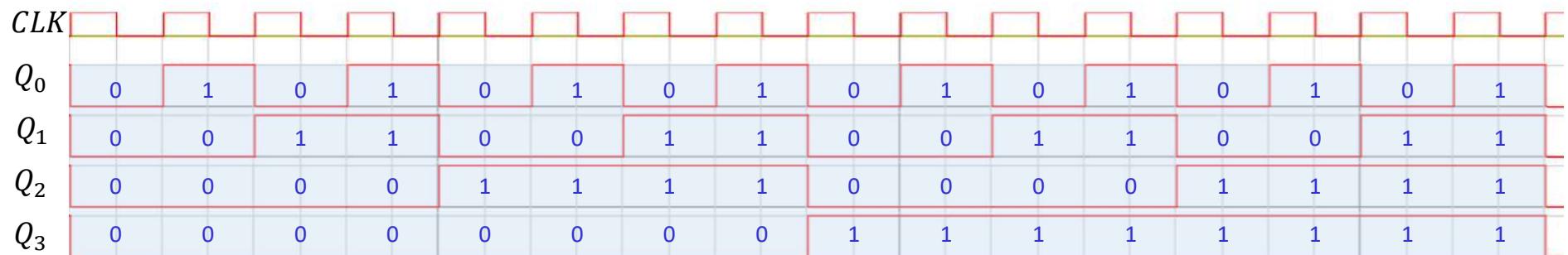
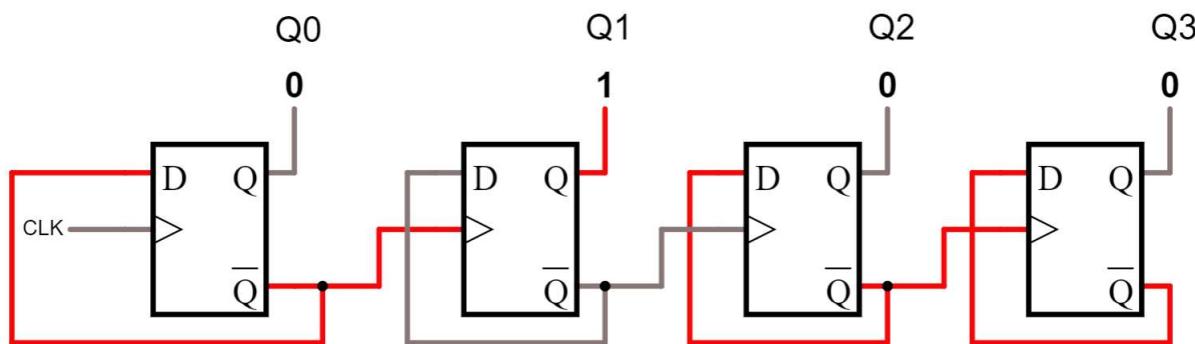
- O *glitch* anterior era devido as duas entradas do *AND* serem *HIGH* durante o  $t_p$ ;
- Agora, enquanto o *CLK* muda de *HIGH* para *LOW*, a saída  $Q$  levará um  $t_p$  para mudar de *LOW* para *HIGH*;
- Durante este  $t_p$  ambos serão *LOW*, não gerando o *glitch*.



## Circuitos Sequenciais Contadores assíncronos

- Construir um contador assíncrono de 4 *bits* utilizando  $FF\ D$ .

## Circuitos Sequenciais Contadores assíncronos

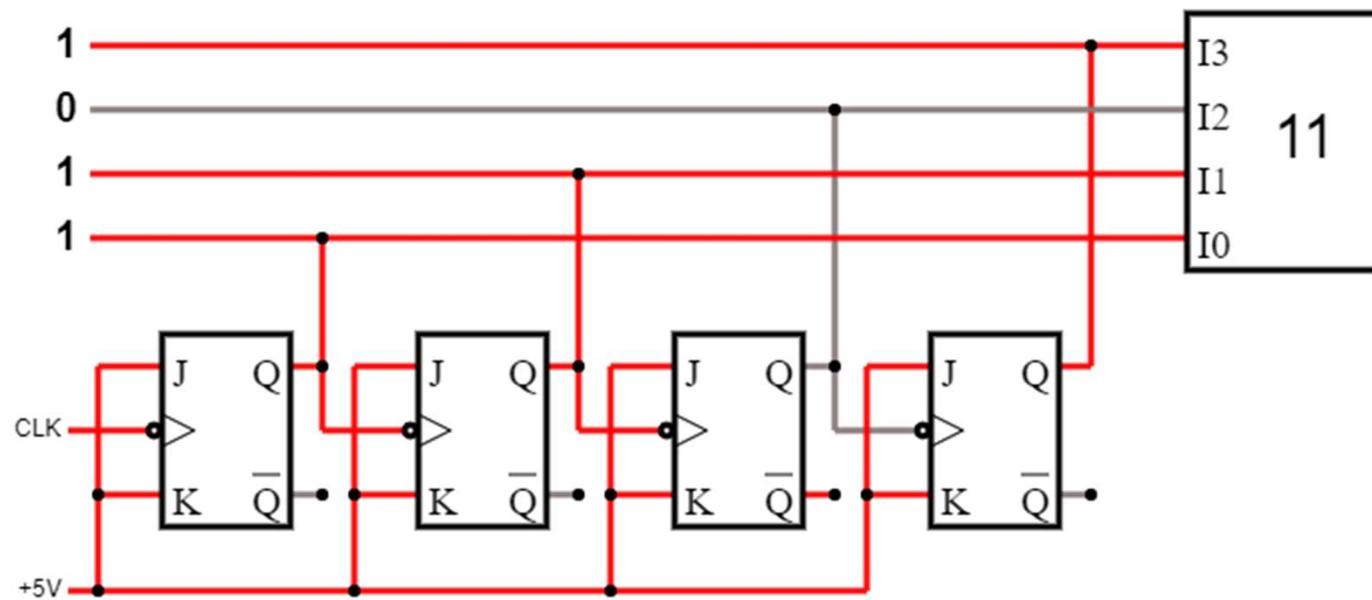


<https://tinyurl.com/yhhnyd2j>

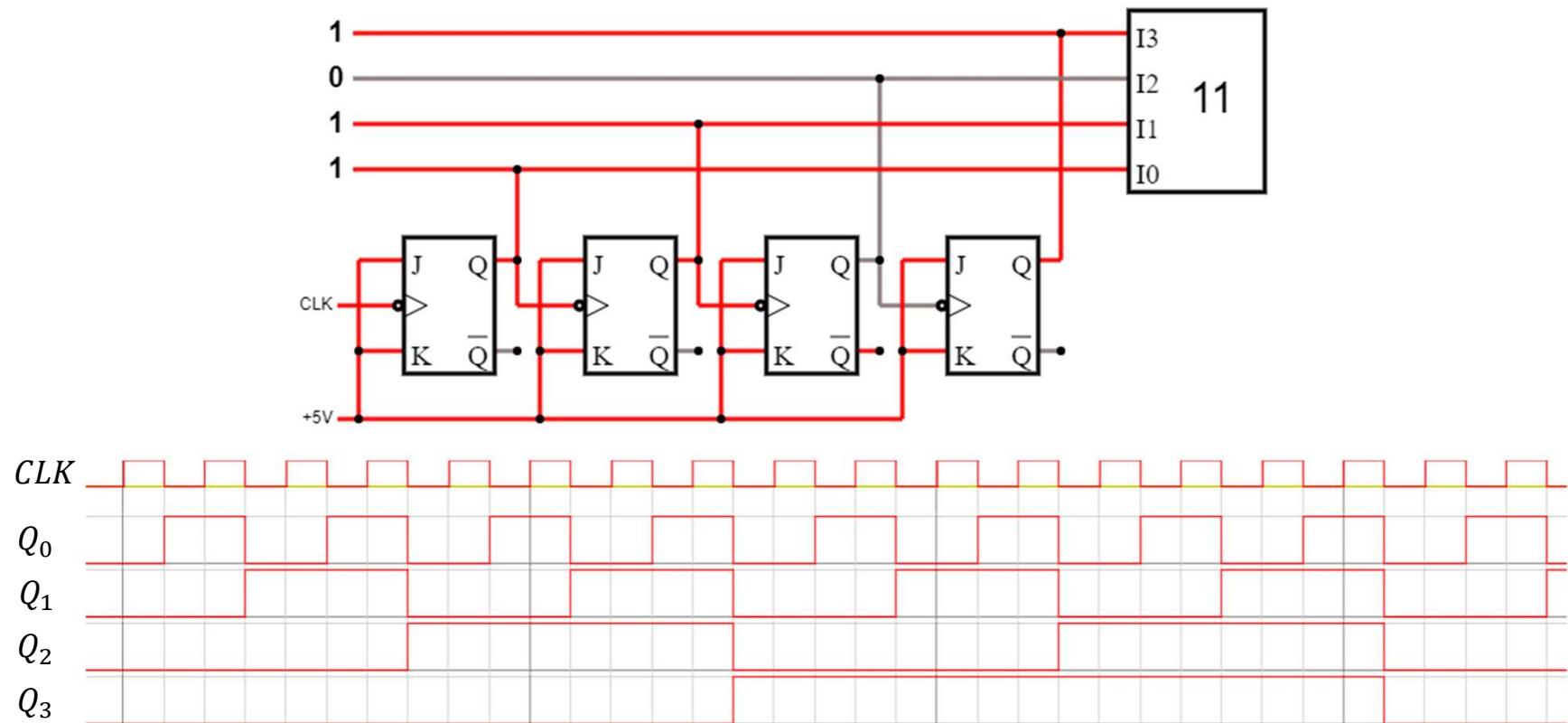
## Circuitos Sequenciais Contadores assíncronos

- Construir um contador assíncrono de 4 *bits* utilizando  $FF\ J-K$ .

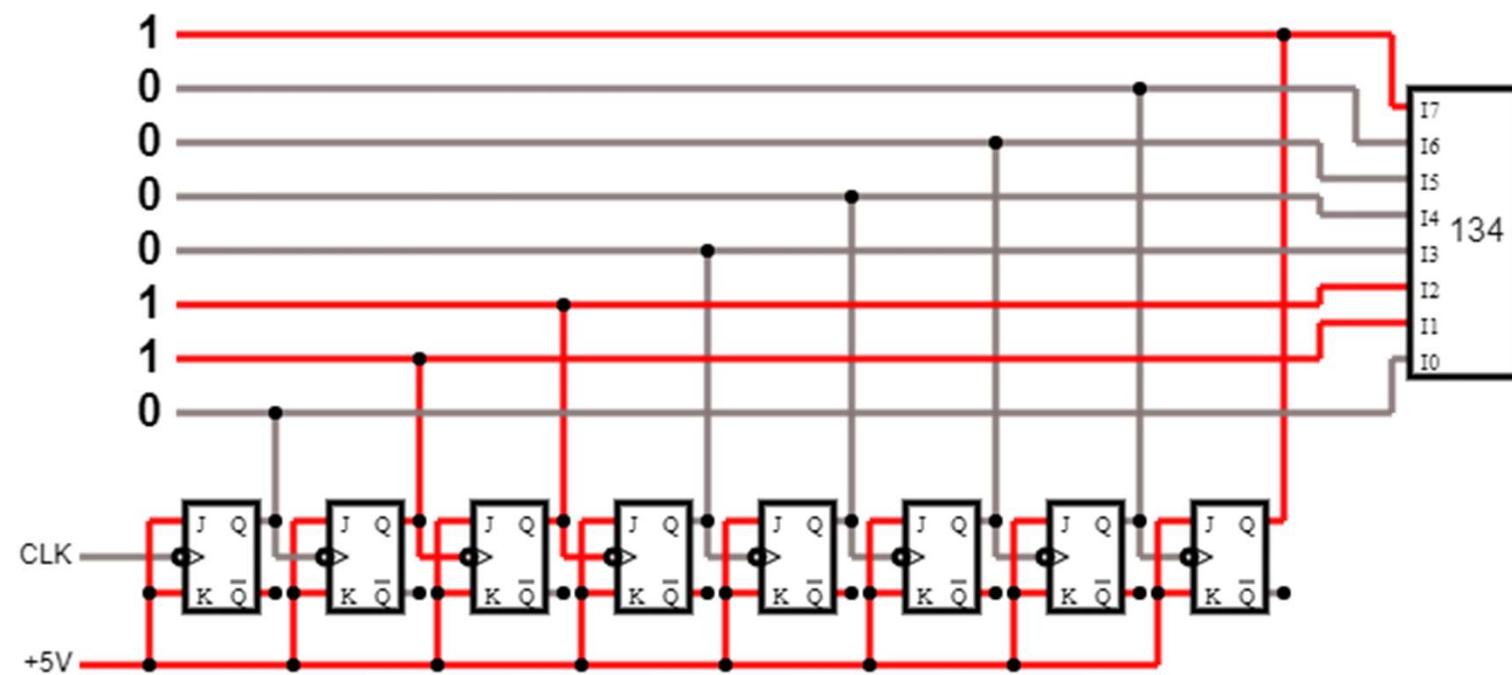
## Circuitos Sequenciais Contadores assíncronos



## Circuitos Sequenciais Contadores assíncronos



## Circuitos Sequenciais Contadores assíncronos



## Circuitos Sequenciais

### Contadores assíncronos

- O módulo de um contador é o número de estados únicos gerados;
- O número máximo possível de estados (módulo máximo) de um contador é  $2^n$ , onde  $n$  é o número de *flip-flops* no contador;
- Os contadores podem ser projetados para ter um número de estados menor que o máximo de  $2^n$ ;
- Este tipo de sequência é denominada de sequência truncada;
- É muito comum precisar de contador com sequência truncada de módulo dez (denominado *MOD10*);
- Os contadores com dez estados são chamados de contadores de década;

## Circuitos Sequenciais

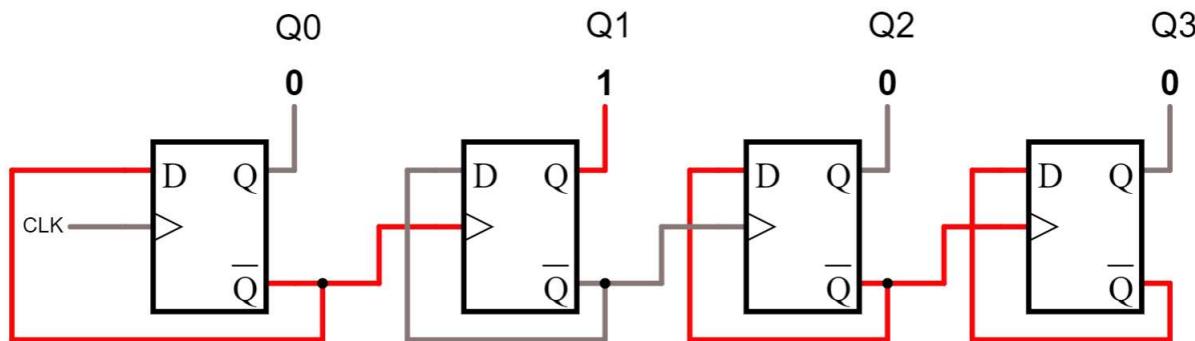
### Contadores assíncronos

- Um contador de décadas com uma sequência de contagem de zero (0000) a nove (1001) é um contador de décadas *BCD* porque sua sequência de dez estados produz o código *BCD*;
- Para obter uma sequência truncada, é necessário forçar o *reset* do contador antes de circular por todos os seus estados possíveis;
- Por exemplo, o contador de décadas *BCD* deve reinicializar o estado para 0000 após o estado 1001;
- Um contador de década requer quatro *flip-flops*;

## Circuitos Sequenciais Contadores assíncronos

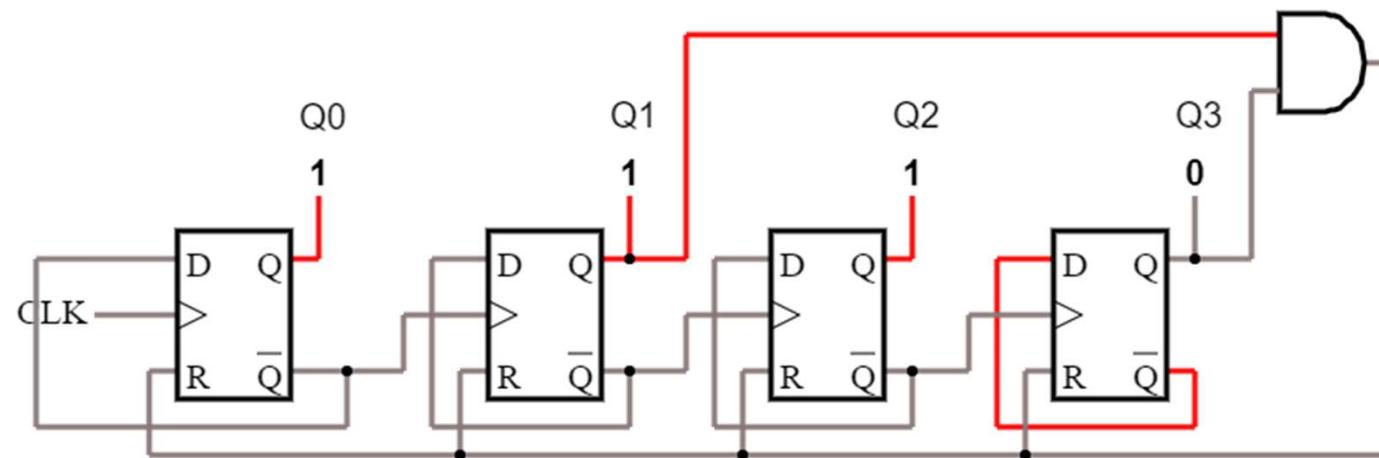
- Construir um contador *BCD (MOD10)* assíncrono de 4 bits.

## Circuitos Sequenciais Contadores assíncronos



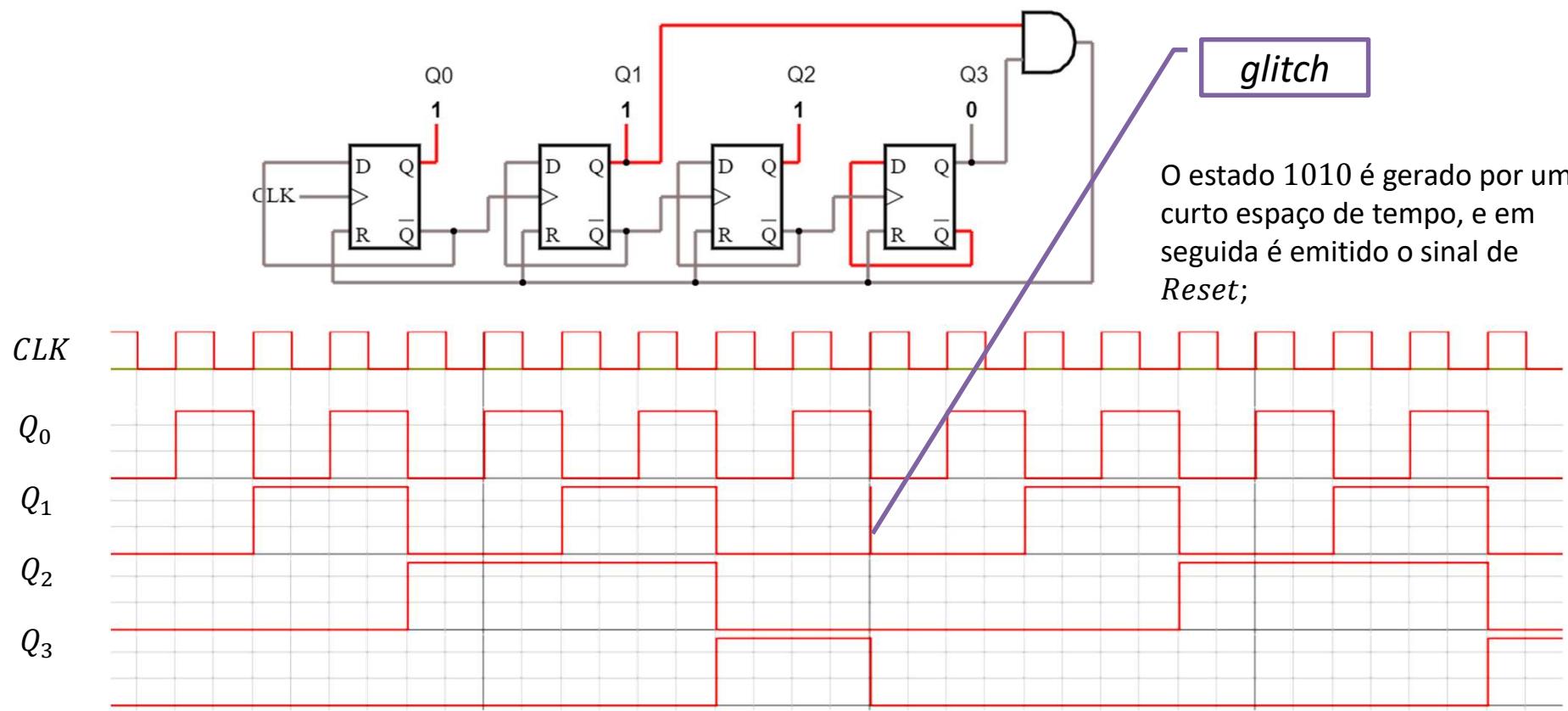
Contador assíncrono de 4 bits.

## Circuitos Sequenciais Contadores assíncronos

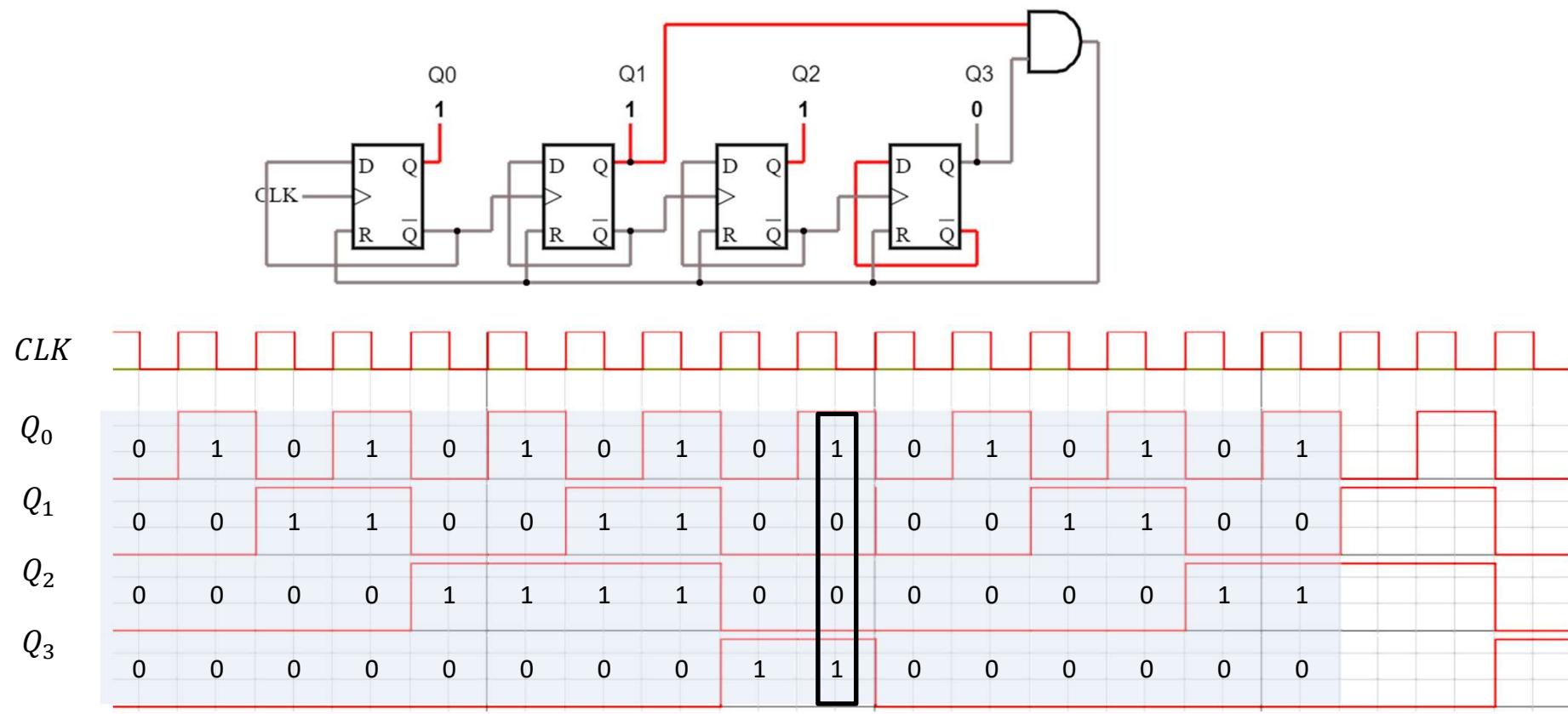


Contador *BCD (MOD10)* assíncrono de 4 bits.

## Circuitos Sequenciais Contadores assíncronos



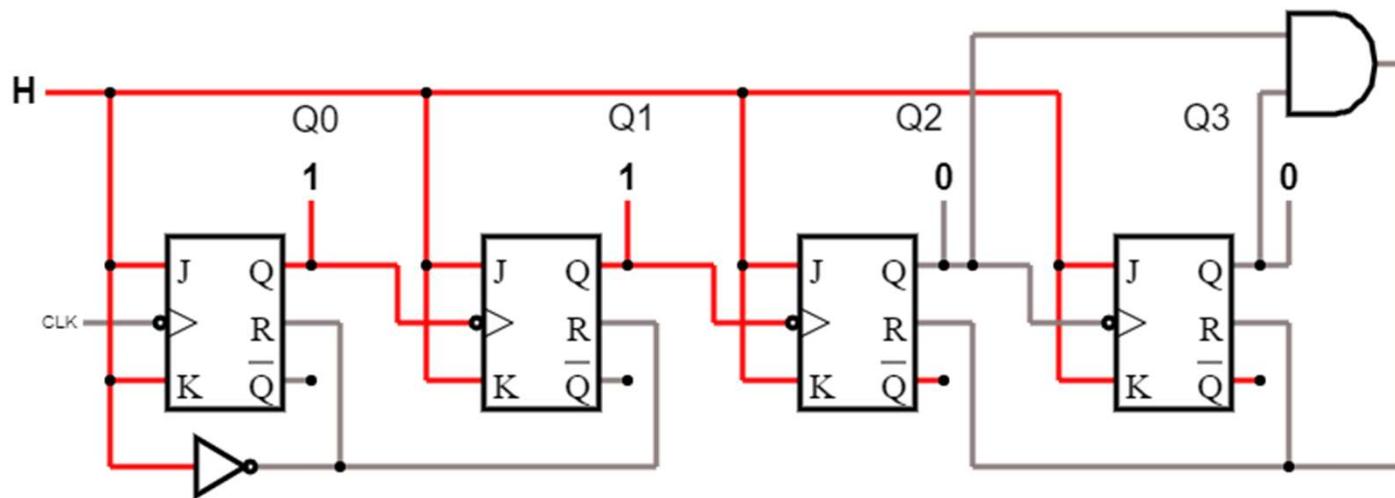
## Circuitos Sequenciais Contadores assíncronos



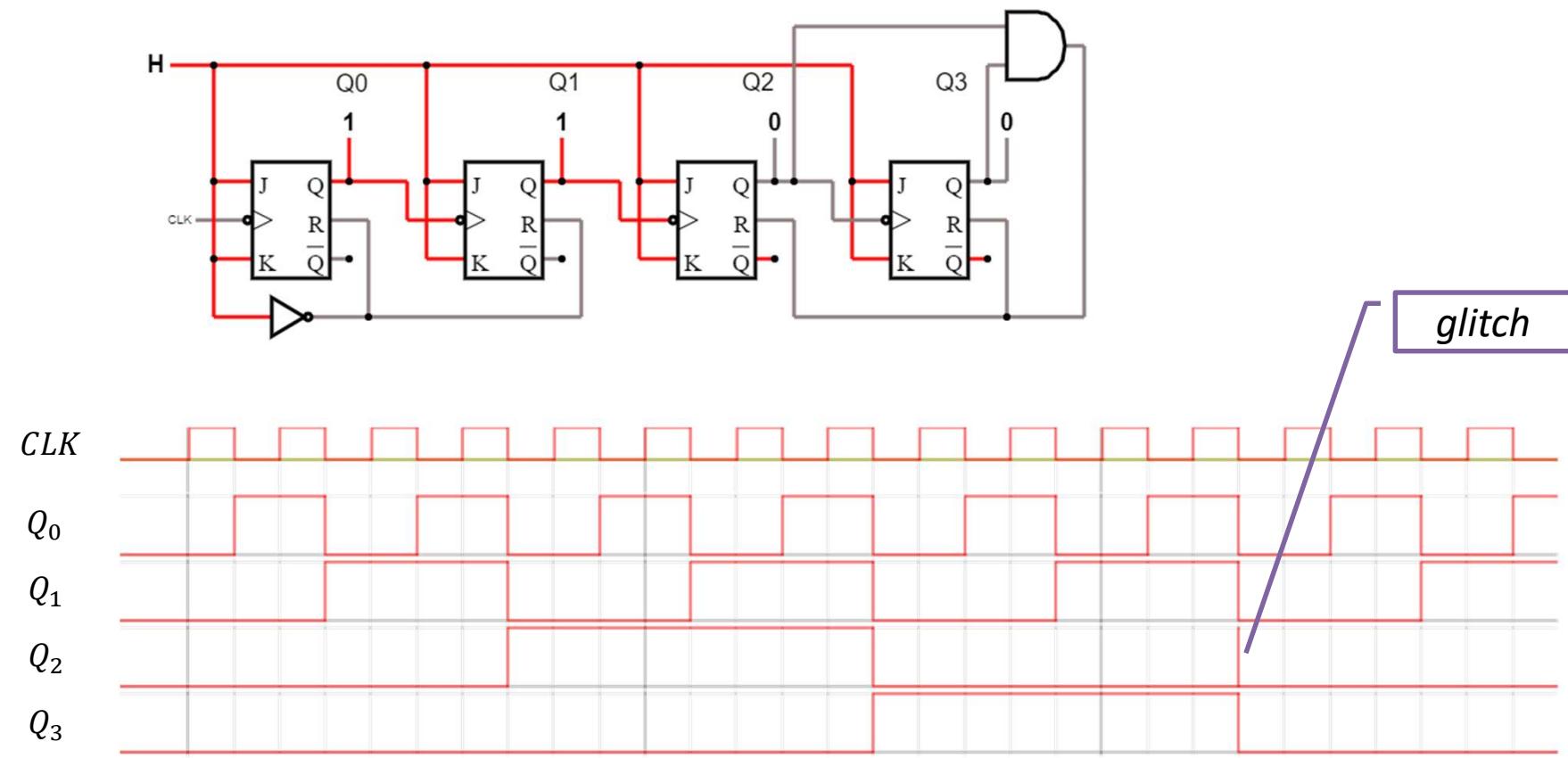
## Circuitos Sequenciais Contadores assíncronos

- Construir um contador módulo 12 assíncrono de 4 *bits* utilizando *flip-flops J-K*.

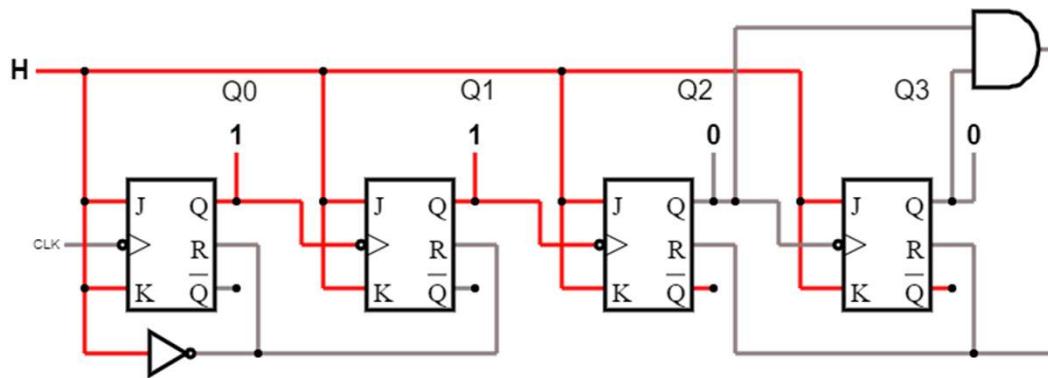
## Circuitos Sequenciais Contadores assíncronos



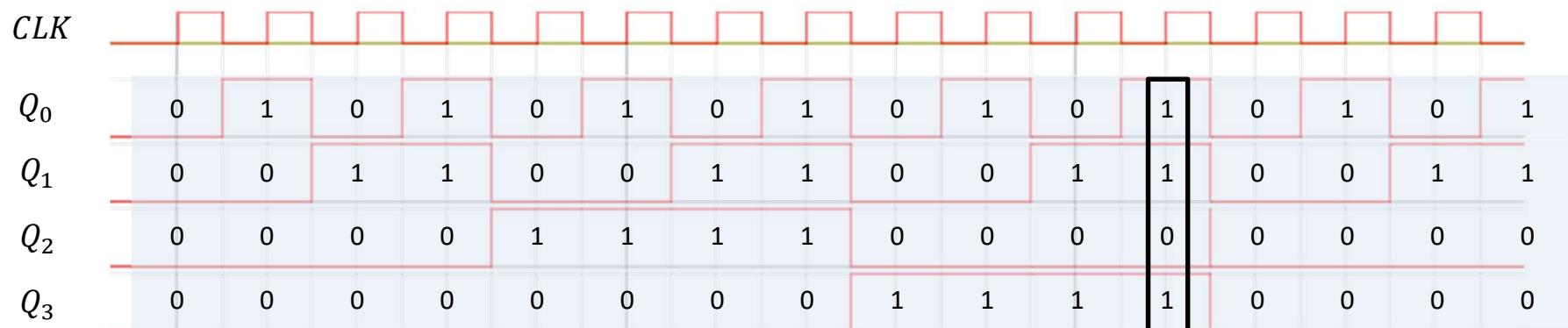
## Circuitos Sequenciais Contadores assíncronos



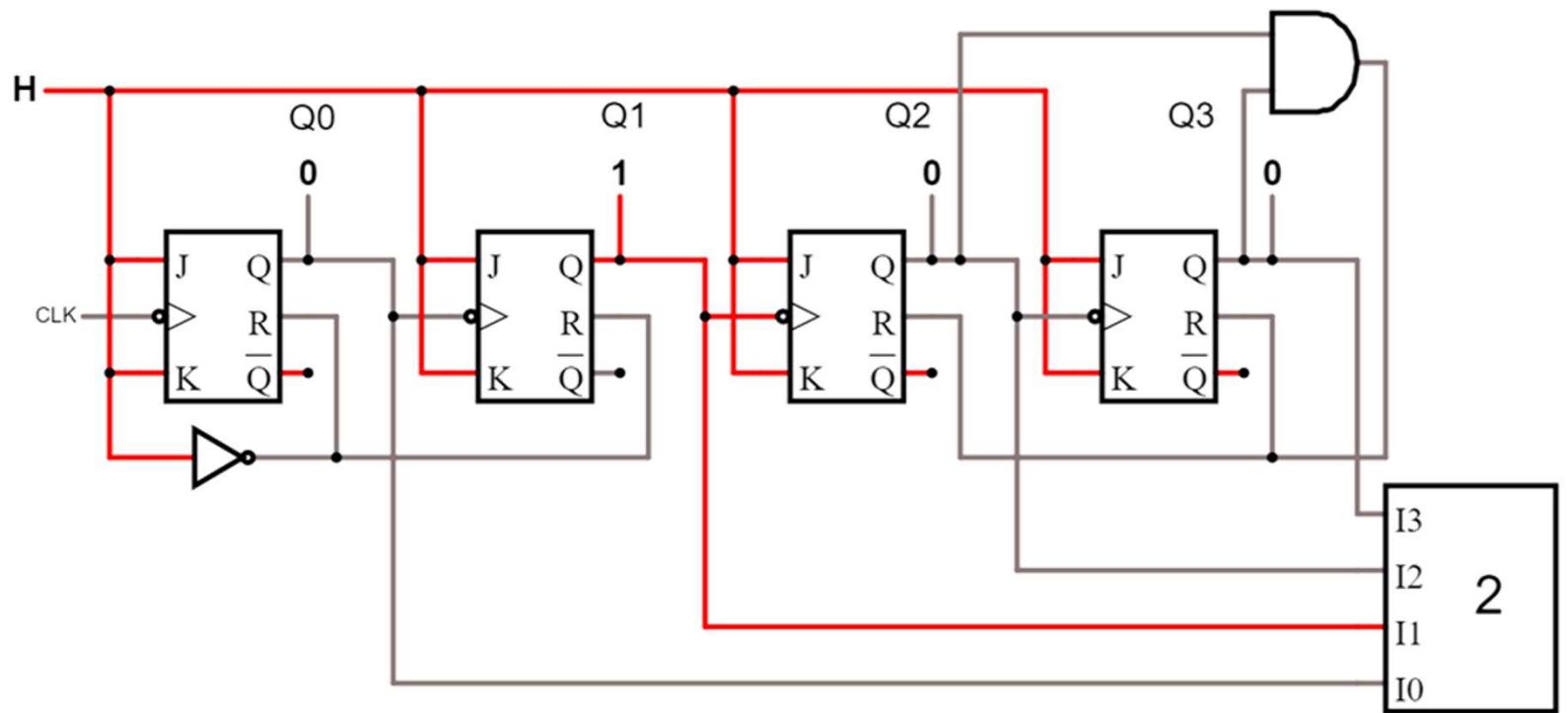
## Circuitos Sequenciais Contadores assíncronos



- Todos os  $FF$  estão em modo *toggle* ( $J = 1$  e  $K = 1$ );
- O  $FF_0$  alterna a cada pulso do *clock*;
- O  $FF_1$  alterna sempre que  $Q_0$  comutar de 1 para 0;
- Comportamento semelhante terão  $FF_2$  e  $FF_3$ ;
- O conjunto sofre um Reset sempre que alcançar o estado 1100.

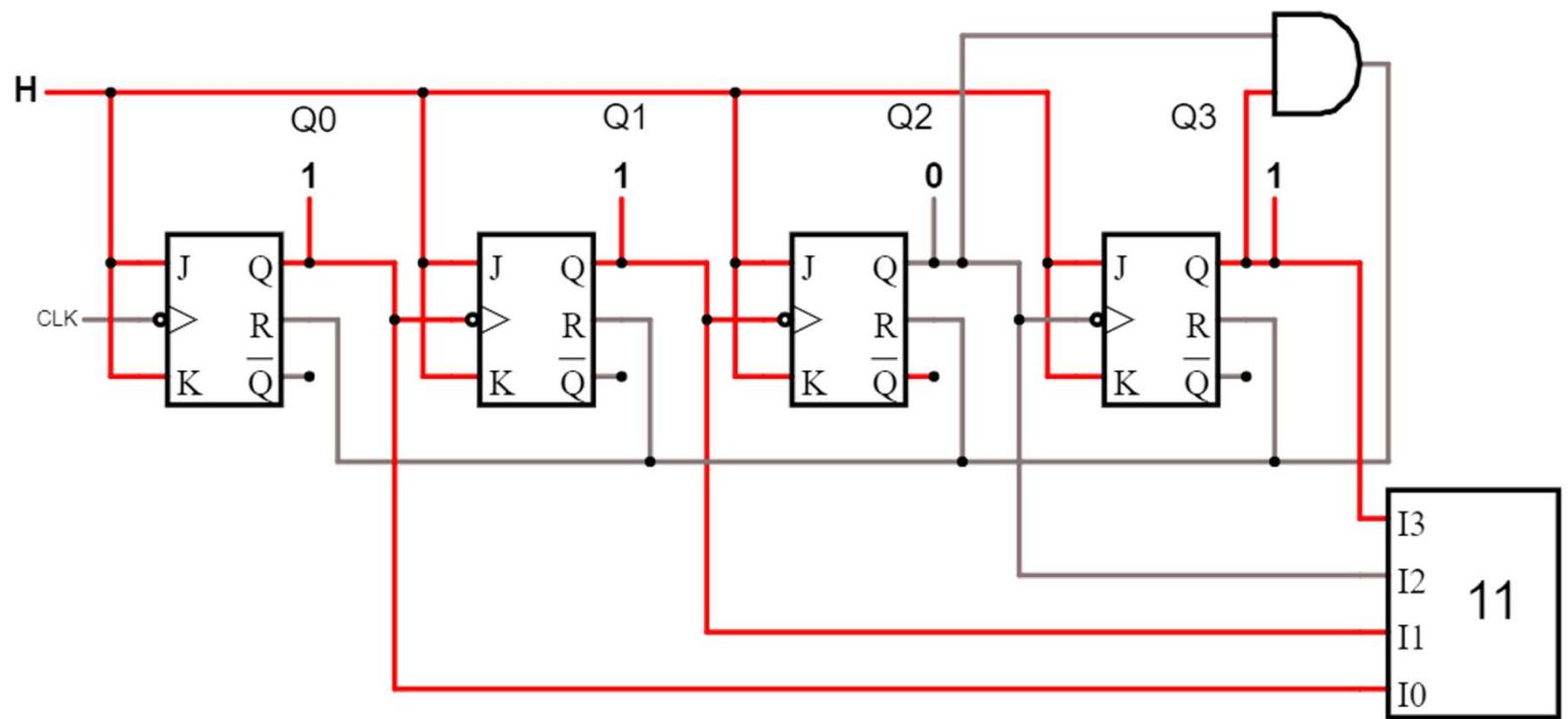


## Circuitos Sequenciais Contadores assíncronos



<https://tinyurl.com/yfbmhwyw>

## Circuitos Sequenciais Contadores assíncronos



<https://tinyurl.com/ye93u95s>

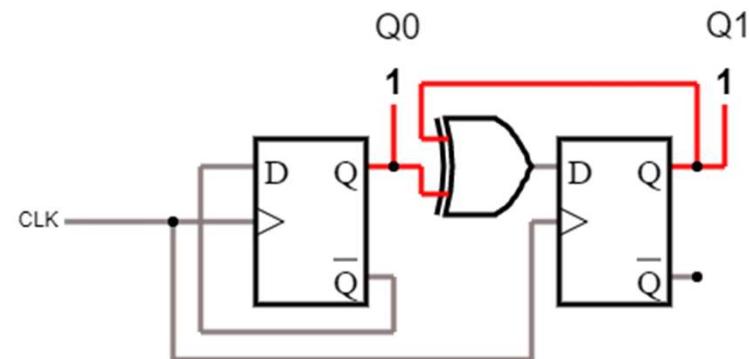
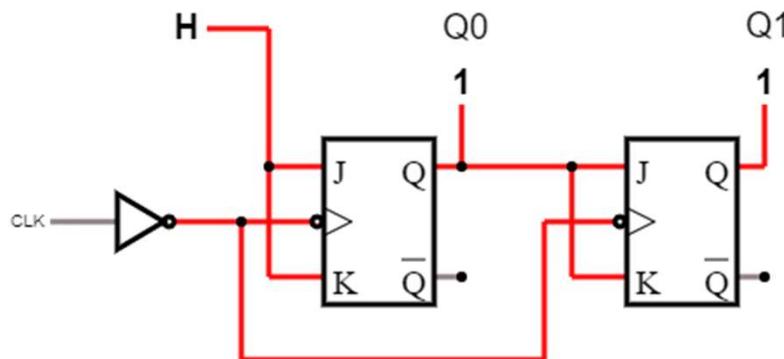
## Circuitos Sequenciais

### Contadores síncronos

- O termo síncrono refere-se a eventos que ocorrem cadenciados com uma determinada frequência;
- Em um contador síncrono, todos os *flip-flops* são acionados ao mesmo tempo por um pulso de *clock* compartilhado;
- Os *flip-flops J–K* são usados para ilustrar a maioria dos contadores síncronos;
- Os *flip-flops D* também podem ser usados, mas geralmente requerem lógicas adicionais, por não terem as funcionalidades de alternância ou manutenção de estado.

## Circuitos Sequenciais Contadores síncronos

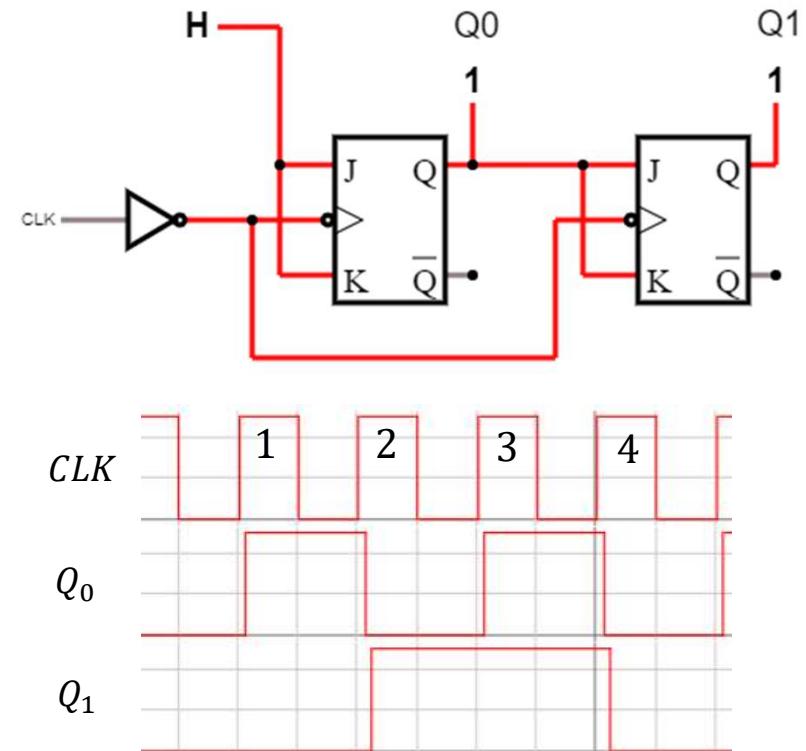
- O diagrama a seguir ilustra um contador binário síncrono de 2 bits;
- Observe a diferença em relação ao contador assíncrono (compartilhamento do *clock*);
- No contador síncrono as entradas  $J_1$  e  $K_1$  do  $FF_1$  são utilizadas a fim de obter uma sequência binária;
- Uma implementação de *flip-flop D* é mostrada ao lado.



<https://tinyurl.com/yg8lc2zf>

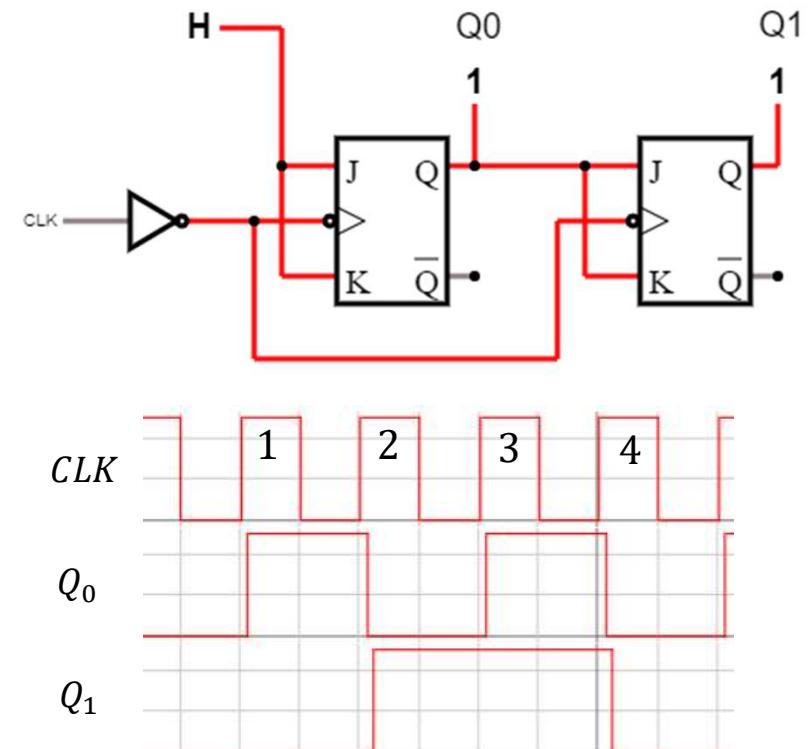
## Circuitos Sequenciais Contadores síncronos

- Partiremos do estado inicial 00 para os dois *flip-flops* ( $FF_0$  e  $FF_1$ );
- Na primeira borda positiva do *clock*, o  $FF_0$  irá alternar e  $Q_0$  irá para *HIGH*;
- As entradas  $J_1$  e  $K_1$  estão em *LOW* uma vez que  $Q_0$  ainda não comutou para *HIGH*.
- Lembre-se de que há um atraso de propagação entre a aplicação da borda de disparo do *clock* até que a saída  $Q$  realmente faça uma transição;
- Portanto,  $J_1 = 0$  e  $K_1 = 0$  quando a borda de subida do primeiro pulso de *clock* é aplicada;
- Esta é uma condição de manutenção, portanto,  $FF_1$  não muda de estado;



## Circuitos Sequenciais Contadores síncronos

- Após  $CLK_1$ ,  $Q_0 = 1$  e  $Q_1 = 0$  (que é o estado binário  $1 = 01$ );
- Em  $CLK_2$ ,  $FF_0$  alternará e  $Q_0$  irá para *LOW*;
- Neste momento o  $FF_1$  estará com um *HIGH* ( $Q_0 = 1$ ) em  $J_1$  e  $K_1$ , fazendo com que o *flip-flop* alterne  $Q_1$  para *HIGH*;
- Assim, após  $CLK_2$ ,  $Q_0 = 0$  e  $Q_1 = 1$  (que é o estado binário  $2 = 10$ );
- Em  $CLK_3$ ,  $FF_0$  alterna novamente para o estado *HIGH* ( $Q_0 = 1$ ) e  $FF_1$  permanece *HIGH* ( $Q_1 = 1$ ), uma vez que suas entradas  $J_1$  e  $K_1$  estão em *LOW* ( $Q_0 = 0$ );
- Neste pulso,  $Q_0 = 1$  e  $Q_1 = 1$  (que é um estado binário  $3 = 11$ ).

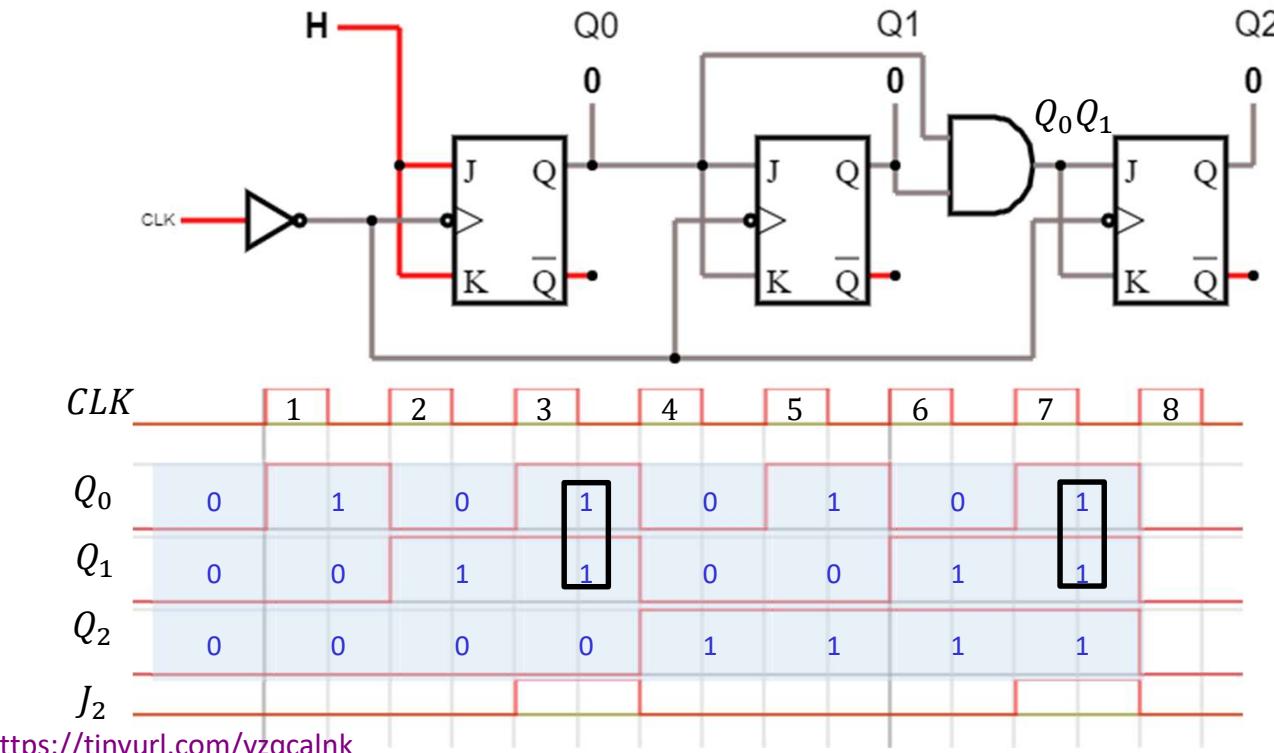


## Circuitos Sequenciais Contadores síncronos

- Construir um contador síncrono de 3 *bits* utilizando *flip-flops J–K*.

## Circuitos Sequenciais Contadores síncronos

- Construir um contador síncrono de 3 bits utilizando *flip-flops J-K*.



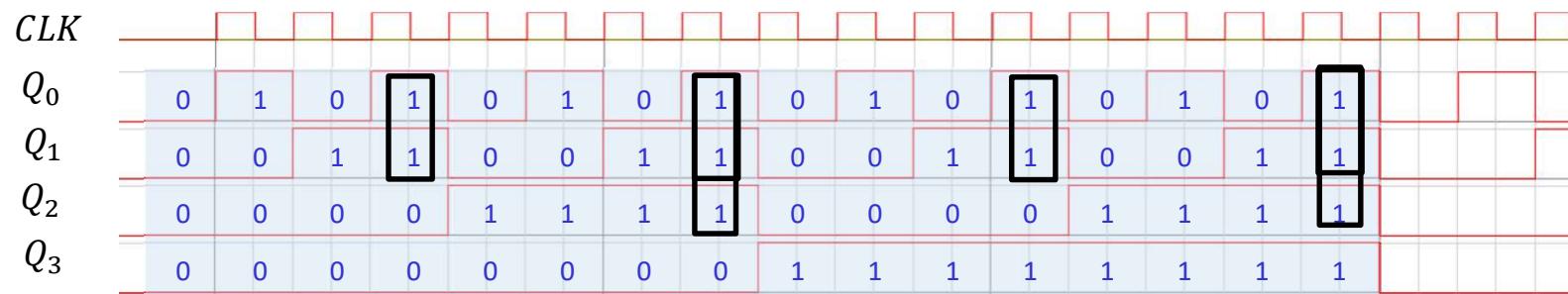
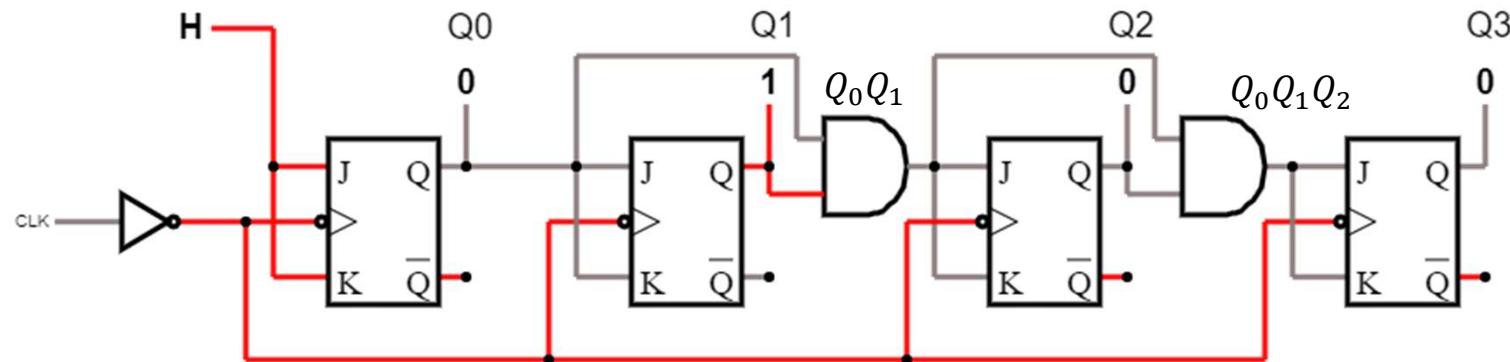
O  $FF_2$  é liberado para modo *toggle*, no próximo pulso do *clock*, quando  $Q_0Q_1 = 1$ .

<i>Clock Pulse</i>	$Q_2$	$Q_1$	$Q_0$
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1
8	0	0	0

## Circuitos Sequenciais Contadores síncronos

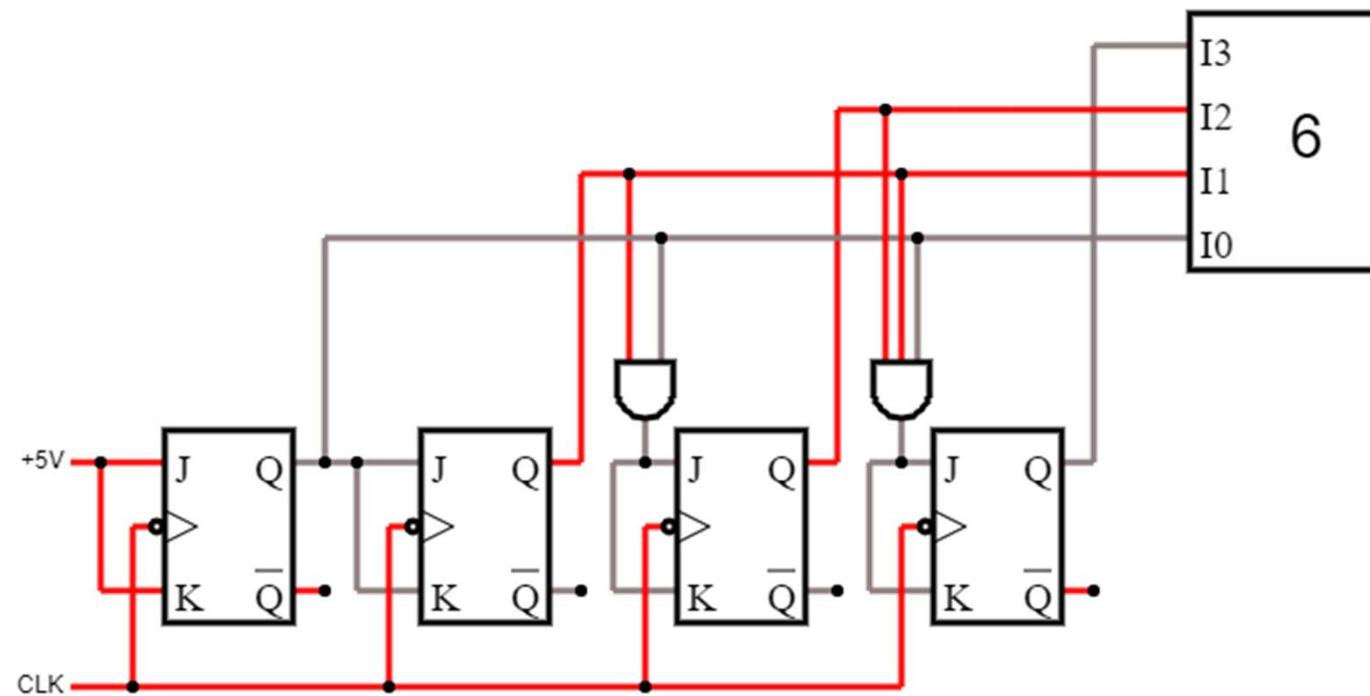
- Construir um contador síncrono de 4 *bits* utilizando *flip-flops J–K*.

## Circuitos Sequenciais Contadores síncronos



<https://tinyurl.com/ygkpwqod>

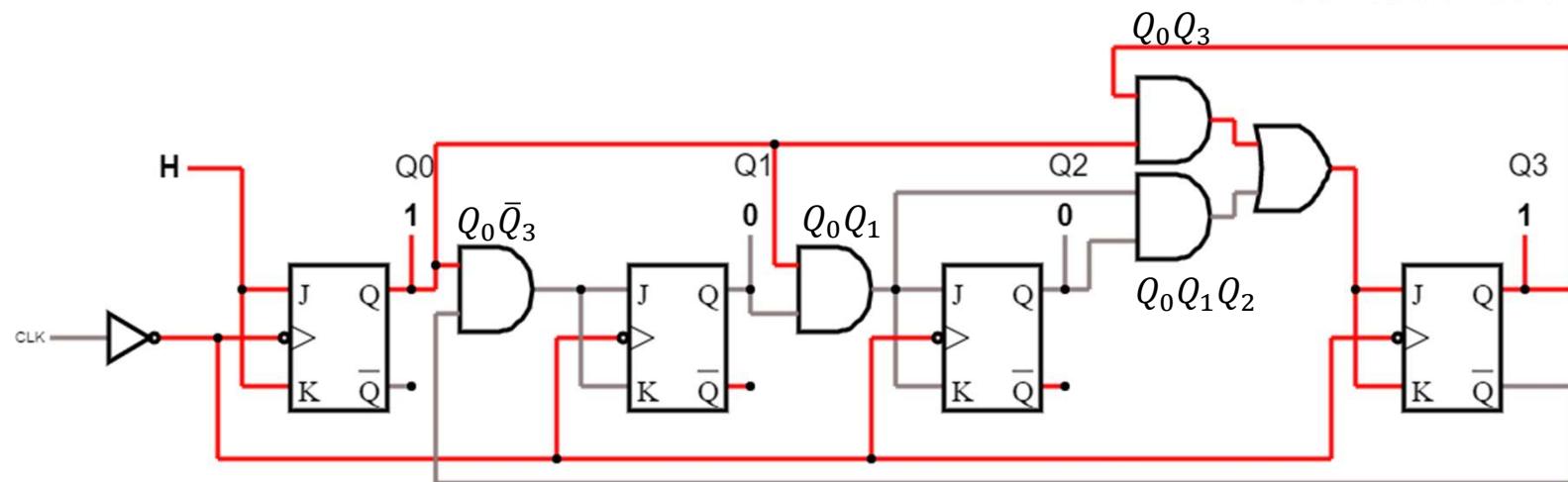
## Circuitos Sequenciais Contadores síncronos



## Circuitos Sequenciais Contadores síncronos

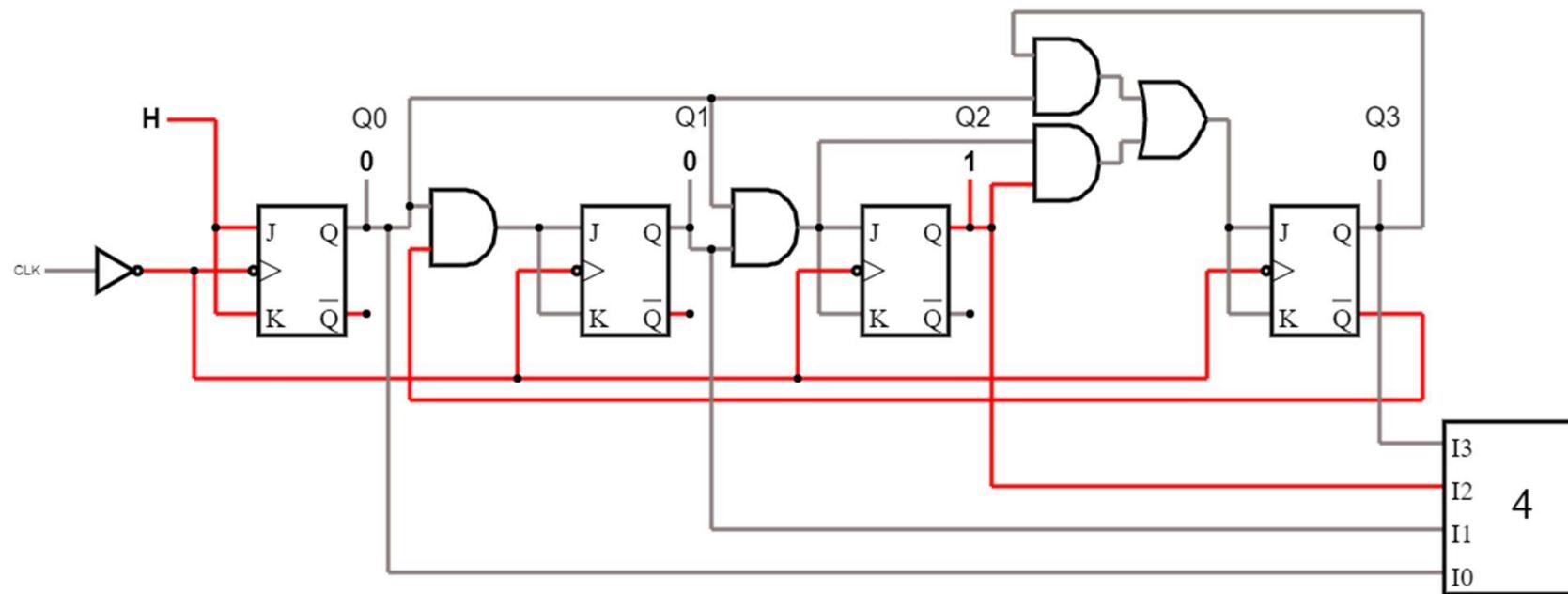
- Construir um contador síncrono módulo 10 (*BCD*) utilizando *flip-flops J-K*.

## Circuitos Sequenciais Contadores síncronos

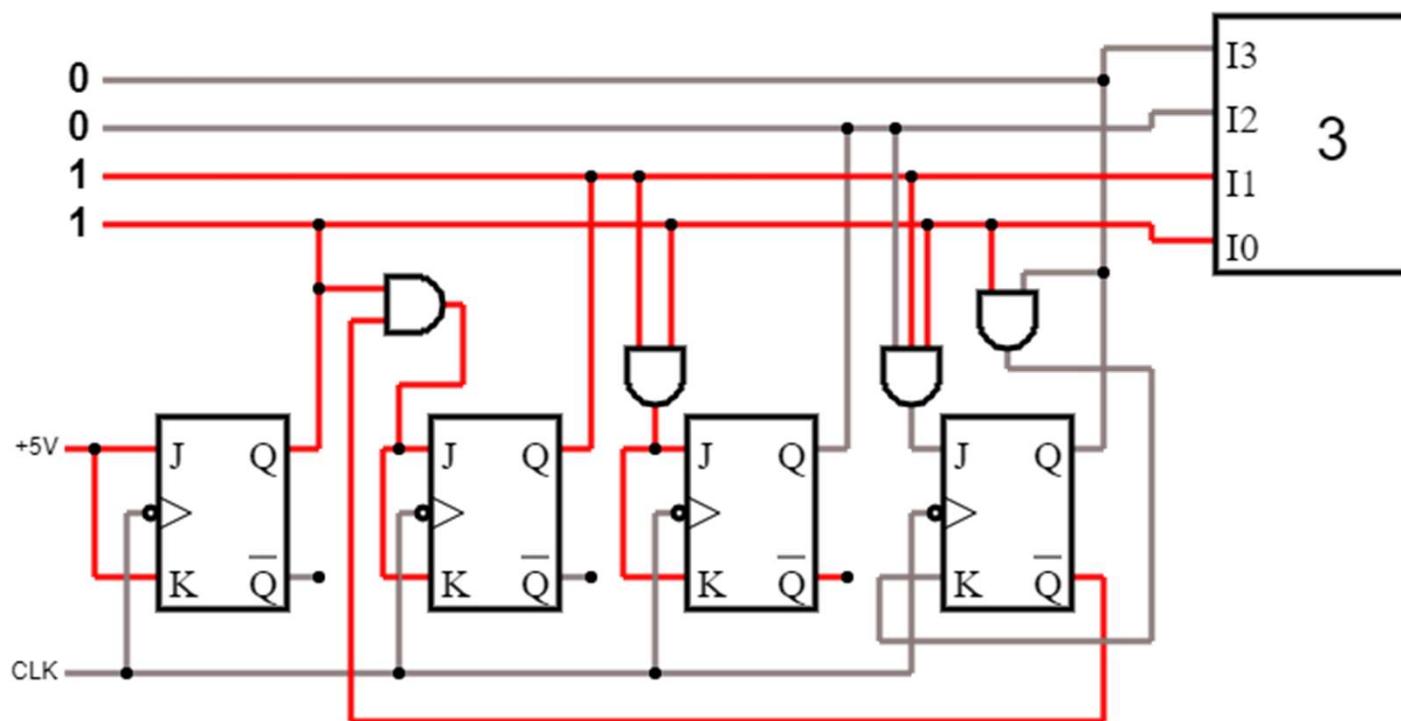


$CLK$	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
$Q_0$	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
$Q_1$	0	0	1	1	0	0	1	1	0	0	0	0	1	1	0	1	0
$Q_2$	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0
$Q_3$	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1

## Circuitos Sequenciais Contadores síncronos



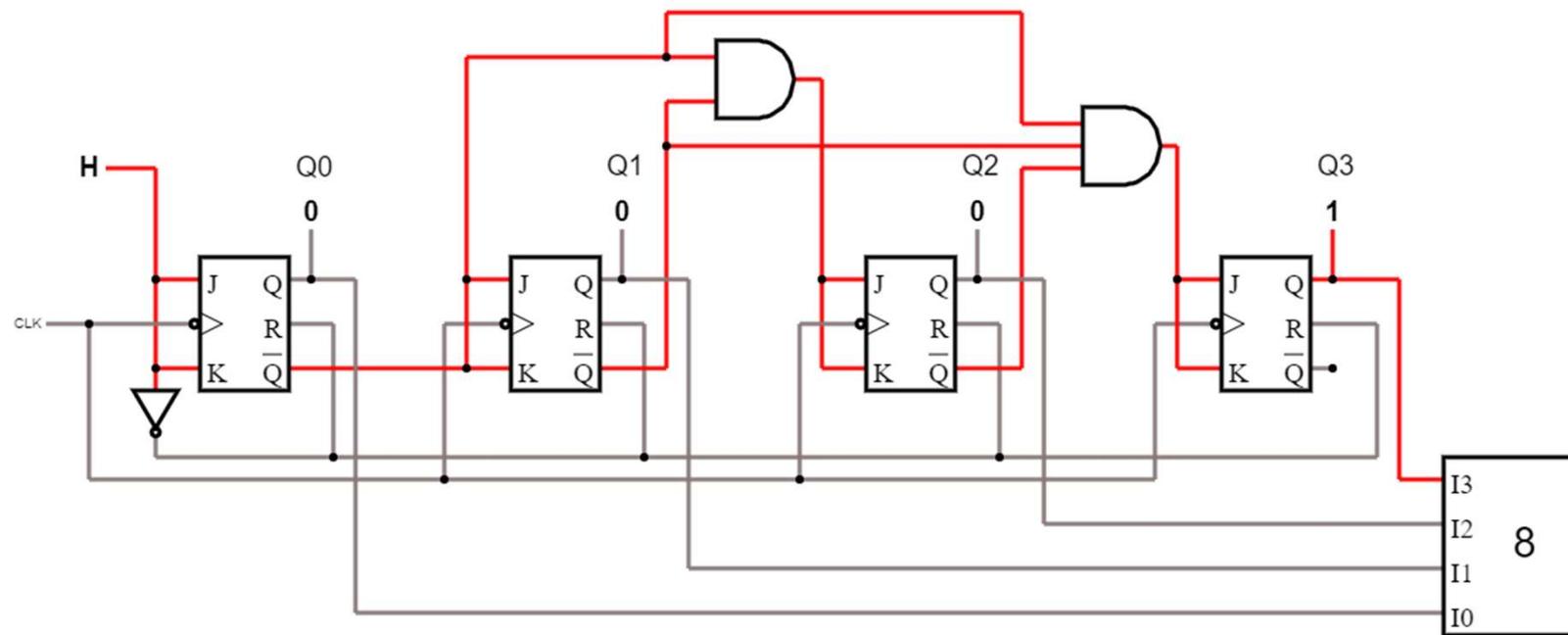
## Circuitos Sequenciais Contadores síncronos



## Circuitos Sequenciais Contadores síncronos

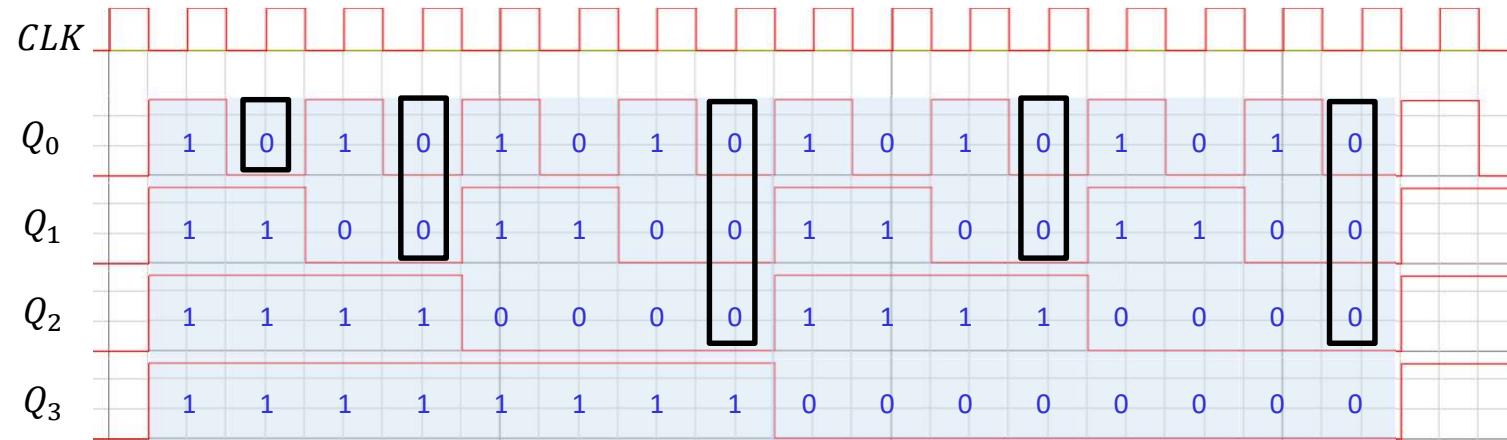
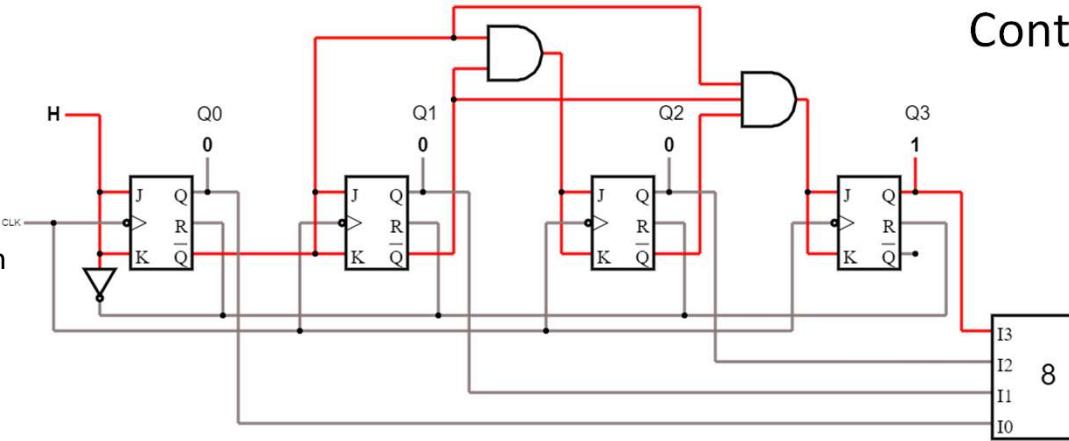
- Construir um contador síncrono 4 *bits* decrescente.

## Circuitos Sequenciais Contadores síncronos



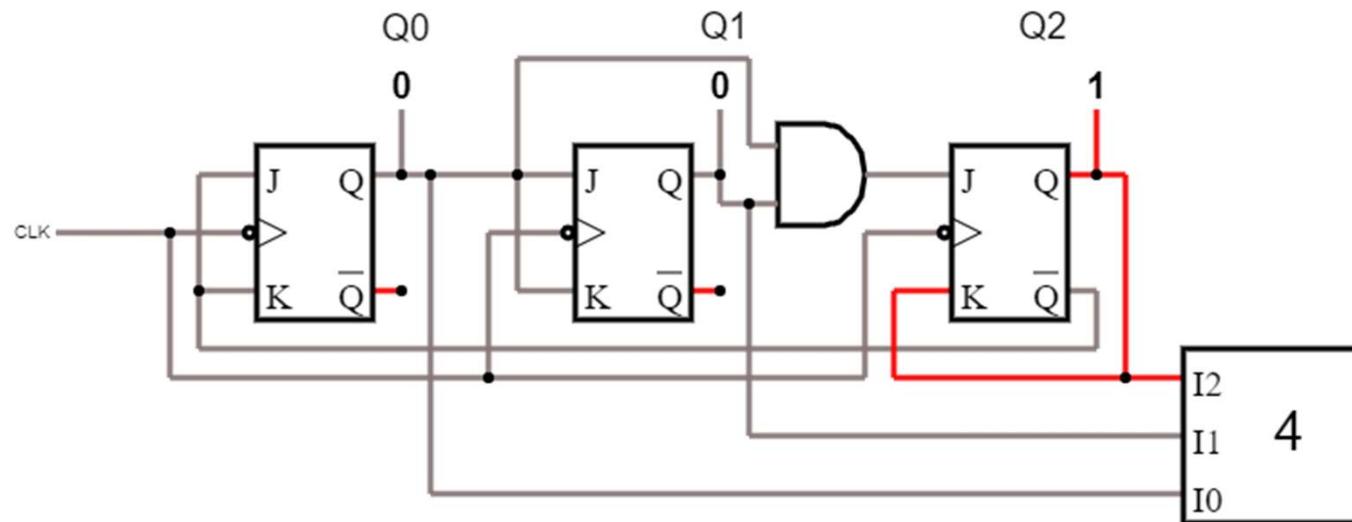
## Circuitos Sequenciais Contadores síncronos

O *reset* inicial coloca  $\bar{Q}$  em 1 fazendo com que todos os *FF* entre em modo *toggle* e comutem para 1 no primeiro pulso do clock.



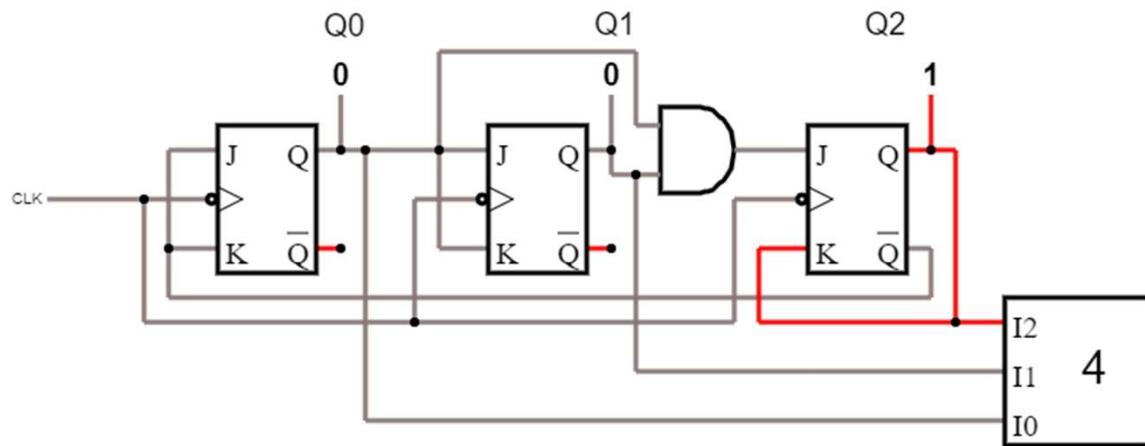
## Circuitos Sequenciais Contadores síncronos

- Analisar o contador síncrono a seguir. Determinar sua função. Desenhe o seu diagrama de tempo. Implementar em um simulador.



## Circuitos Sequenciais Contadores síncronos

a. Expressões das Entradas de Controle:



$$J_0 = K_0 = \bar{Q}_2$$

$$J_1 = K_1 = Q_0$$

$$J_2 = Q_0 Q_1$$

$$K_2 = Q_2$$

## Circuitos Sequenciais Contadores síncronos

### b. Determinação dos Estados:

Estado inicial =  $Q_2Q_1Q_0 = 000$  e aplicando as expressões:  $J_0 = K_0 = \bar{Q}_2$

$$J_0 = K_0 = \bar{Q}_2 = 1 \Rightarrow Q_0 = 1 \quad J_1 = K_1 = Q_0$$

$$J_1 = K_1 = Q_0 = 0 \Rightarrow Q_1 = 0 \quad J_2 = Q_0Q_1$$

$$J_2 = Q_0Q_1 = 0 \quad K_2 = Q_2$$

$$K_2 = Q_2 = 0 \quad \Rightarrow Q_2 = 0$$

Estado final =  $Q_2Q_1Q_0 = 001$ .

<i>CLK</i>	<i>J</i>	<i>K</i>	<i>Q</i>	$\bar{Q}$	<i>Função</i>
?	0	0	$Q_{i-1}$	$\bar{Q}_{i-1}$	Mantém <i>Q</i>
$\uparrow$	0	1	0	1	Reset <i>Q</i>
$\uparrow$	1	0	1	0	Set <i>Q</i>
$\uparrow$	1	1	$\bar{Q}_{i-1}$	$Q_{i-1}$	Inverte

## Circuitos Sequenciais Contadores síncronos

### b. Determinação dos Estados:

Estado =  $Q_2Q_1Q_0 = 001$  e aplicando as expressões:

$$J_0 = K_0 = \bar{Q}_2 = 1 \Rightarrow Q_0 = 0$$

$$J_1 = K_1 = Q_0 = 1 \Rightarrow Q_1 = 1$$

$$J_2 = Q_0Q_1 = 0$$

$$K_2 = Q_2 = 0 \quad \Rightarrow Q_2 = 0$$

Estado final =  $Q_2Q_1Q_0 = 010$ .

$$J_0 = K_0 = \bar{Q}_2$$

$$J_1 = K_1 = Q_0$$

$$J_2 = Q_0Q_1$$

$$K_2 = Q_2$$

<i>CLK</i>	<i>J</i>	<i>K</i>	<i>Q</i>	$\bar{Q}$	<i>Função</i>
?	0	0	$Q_{i-1}$	$\bar{Q}_{i-1}$	Mantém <i>Q</i>
↑	0	1	0	1	Reset <i>Q</i>
↑	1	0	1	0	Set <i>Q</i>
↑	1	1	$\bar{Q}_{i-1}$	$Q_{i-1}$	Inverte

## Circuitos Sequenciais Contadores síncronos

### b. Determinação dos Estados:

Estado =  $Q_2Q_1Q_0 = 010$  e aplicando as expressões:

$$J_0 = K_0 = \bar{Q}_2 = 1 \Rightarrow Q_0 = 1$$

$$J_1 = K_1 = Q_0 = 0 \Rightarrow Q_1 = 1$$

$$J_2 = Q_0Q_1 = 0$$

$$K_2 = Q_2 = 0 \Rightarrow Q_2 = 0$$

Estado final =  $Q_2Q_1Q_0 = 011$ .

$$J_0 = K_0 = \bar{Q}_2$$

$$J_1 = K_1 = Q_0$$

$$J_2 = Q_0Q_1$$

$$K_2 = Q_2$$

<i>CLK</i>	<i>J</i>	<i>K</i>	<i>Q</i>	$\bar{Q}$	<i>Função</i>
?	0	0	$Q_{i-1}$	$\bar{Q}_{i-1}$	Mantém <i>Q</i>
↑	0	1	0	1	Reset <i>Q</i>
↑	1	0	1	0	Set <i>Q</i>
↑	1	1	$\bar{Q}_{i-1}$	$Q_{i-1}$	Inverte

## Circuitos Sequenciais Contadores síncronos

b. Determinação dos Estados:

$$\text{Estado} = Q_2Q_1Q_0 = 011:$$

$$J_0 = K_0 = \bar{Q}_2 = 1 \Rightarrow Q_0 = 0$$

$$J_1 = K_1 = Q_0 = 1 \Rightarrow Q_1 = 0$$

$$J_2 = Q_0Q_1 = 1$$

$$K_2 = Q_2 = 0 \quad \Rightarrow Q_2 = 1$$

$$\text{Estado final} = Q_2Q_1Q_0 = 100.$$

$$\text{Estado} = Q_2Q_1Q_0 = 100:$$

$$J_0 = K_0 = \bar{Q}_2 = 0 \Rightarrow Q_0 = 0$$

$$J_1 = K_1 = Q_0 = 0 \Rightarrow Q_1 = 0$$

$$J_2 = Q_0Q_1 = 0$$

$$K_2 = Q_2 = 1 \quad \Rightarrow Q_2 = 0$$

$$\text{Estado final} = Q_2Q_1Q_0 = 000.$$

## Circuitos Sequenciais Contadores síncronos

c. Conclusão:

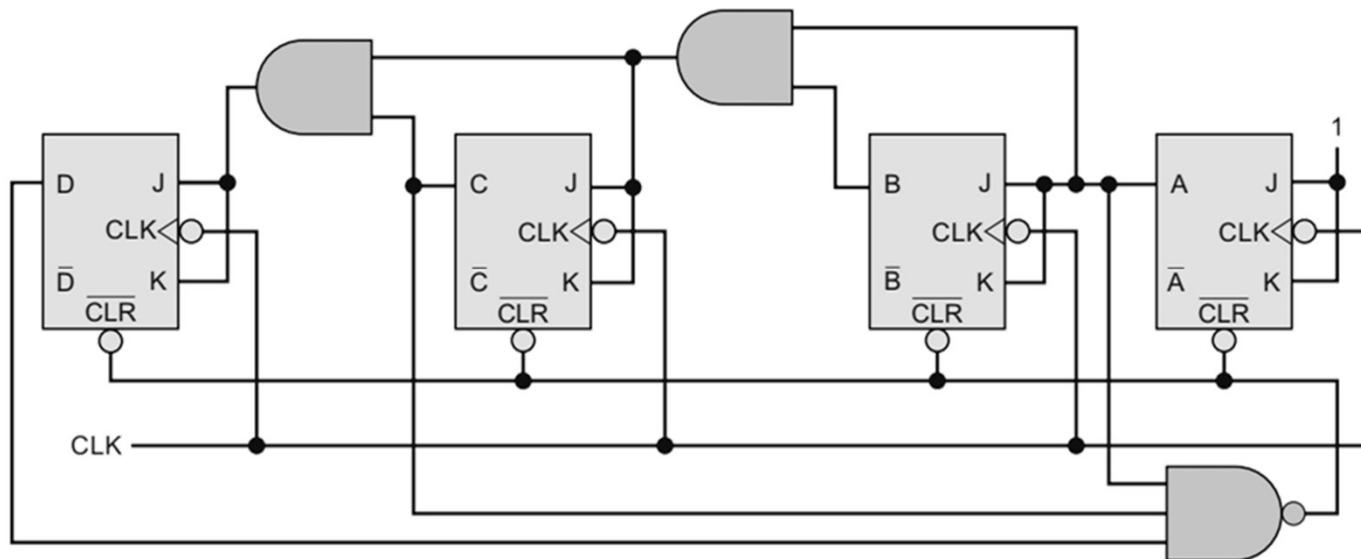
Estado Atual			Entradas de Controle					Próximo Estado			
$Q_2$	$Q_1$	$Q_0$	$J_2$	$K_2$	$J_1$	$K_1$	$J_0$	$K_0$	$Q_2$	$Q_1$	$Q_0$
0	0	0	0	0	0	0	1	1	0	0	1
0	0	1	0	0	1	1	1	1	0	1	0
0	1	0	0	0	0	0	1	1	0	1	1
0	1	1	1	0	1	1	1	1	1	0	0
1	0	0	0	1	0	0	0	0	0	0	0
1	0	1	0	1	1	1	0	0	0	1	1
1	1	0	0	1	0	0	0	0	0	1	0
1	1	1	1	1	1	1	0	0	0	0	1

Os Estados gerados para  $Q_2Q_1Q_0 = \{000, 001, 010, 011, 100\}$ , o que define um contador síncrono de 3 bits módulo 5.

AUTOCORRETOR: Próximo estado retoma o conjunto válido.

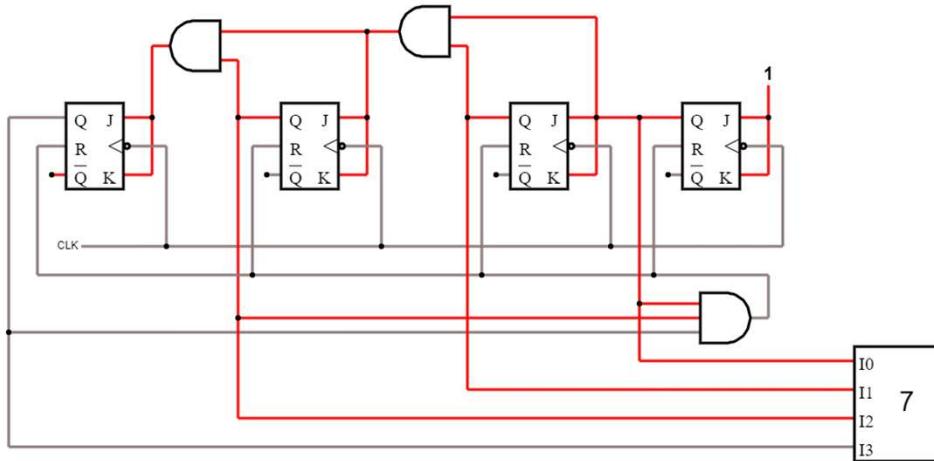
## Circuitos Sequenciais Contadores síncronos

- i. Analisar o contador síncrono abaixo. Desenhe o seu diagrama de tempo e determine o seu módulo. [TOCCI 7.37]. Implementar em um simulador.



## Circuitos Sequenciais Contadores síncronos

a. Expressões das Entradas de Controle:



$$J_0 = K_0 = 1$$

$$J_1 = K_1 = Q_0 \quad R_{0,1,2,3} = Q_3 Q_2 Q_0$$

$$J_2 = K_2 = Q_0 Q_1$$

$$J_3 = K_3 = Q_2 Q_0 Q_1$$

## Circuitos Sequenciais Contadores síncronos

### b. Determinação dos Estados:

Estado inicial =  $Q_3Q_2Q_1Q_0 = 0000$  e aplicando as expressões:  $J_0 = K_0 = 1$

$$J_0 = K_0 = 1 \Rightarrow Q_0 = 1$$

$$J_1 = K_1 = Q_0 = 0 \Rightarrow Q_1 = 0$$

$$J_2 = K_2 = Q_0Q_1 = 0 \Rightarrow Q_2 = 0$$

$$J_3 = K_3 = Q_0Q_1Q_2 = 0 \Rightarrow Q_3 = 0$$

$$J_1 = K_1 = Q_0$$

$$J_2 = K_2 = Q_0Q_1$$

$$J_3 = K_3 = Q_0Q_1Q_2$$

Estado final =  $Q_3Q_2Q_1Q_0 = 0001$ .

<i>CLK</i>	<i>J</i>	<i>K</i>	<i>Q</i>	$\bar{Q}$	<i>Função</i>
?	0	0	$Q_{i-1}$	$\bar{Q}_{i-1}$	Mantém <i>Q</i>
$\uparrow$	0	1	0	1	Reset <i>Q</i>
$\uparrow$	1	0	1	0	Set <i>Q</i>
$\uparrow$	1	1	$\bar{Q}_{i-1}$	$Q_{i-1}$	Inverte

## Circuitos Sequenciais Contadores síncronos

b. Determinação dos Estados:

Estado =  $Q_3Q_2Q_1Q_0 = 0001$  e aplicando as expressões:

$$J_0 = K_0 = 1 \Rightarrow Q_0 = 0$$

$$J_1 = K_1 = Q_0 = 1 \Rightarrow Q_1 = 1$$

$$J_2 = K_2 = Q_0Q_1 = 0 \Rightarrow Q_2 = 0$$

$$J_3 = K_3 = Q_0Q_1Q_2 = 0 \Rightarrow Q_3 = 0$$

$$J_0 = K_0 = 1$$

$$J_1 = K_1 = Q_0$$

$$J_2 = K_2 = Q_0Q_1$$

$$J_3 = K_3 = Q_0Q_1Q_2$$

Estado final =  $Q_3Q_2Q_1Q_0 = 0010$ .

<i>CLK</i>	<i>J</i>	<i>K</i>	<i>Q</i>	$\bar{Q}$	<i>Função</i>
?	0	0	$Q_{i-1}$	$\bar{Q}_{i-1}$	Mantém <i>Q</i>
$\uparrow$	0	1	0	1	Reset <i>Q</i>
$\uparrow$	1	0	1	0	Set <i>Q</i>
$\uparrow$	1	1	$\bar{Q}_{i-1}$	$Q_{i-1}$	Inverte

## Circuitos Sequenciais Contadores síncronos

b. Determinação dos Estados:

Estado =  $Q_3Q_2Q_1Q_0 = 0010$  e aplicando as expressões:

$$J_0 = K_0 = 1 \Rightarrow Q_0 = 1$$

$$J_1 = K_1 = Q_0 = 0 \Rightarrow Q_1 = 1$$

$$J_2 = K_2 = Q_0Q_1 = 0 \Rightarrow Q_2 = 0$$

$$J_3 = K_3 = Q_0Q_1Q_2 = 0 \Rightarrow Q_3 = 0$$

$$J_0 = K_0 = 1$$

$$J_1 = K_1 = Q_0$$

$$J_2 = K_2 = Q_0Q_1$$

$$J_3 = K_3 = Q_0Q_1Q_2$$

Estado final =  $Q_3Q_2Q_1Q_0 = 0011$ .

<i>CLK</i>	<i>J</i>	<i>K</i>	<i>Q</i>	$\bar{Q}$	<i>Função</i>
?	0	0	$Q_{i-1}$	$\bar{Q}_{i-1}$	Mantém <i>Q</i>
$\uparrow$	0	1	0	1	Reset <i>Q</i>
$\uparrow$	1	0	1	0	Set <i>Q</i>
$\uparrow$	1	1	$\bar{Q}_{i-1}$	$Q_{i-1}$	Inverte

## Circuitos Sequenciais Contadores síncronos

b. Determinação dos Estados:

$$\text{Estado} = Q_3Q_2Q_1Q_0 = 0011:$$

$$J_0 = K_0 = 1 \Rightarrow Q_0 = 0$$

$$J_1 = K_1 = Q_0 = 1 \Rightarrow Q_1 = 0$$

$$J_2 = K_2 = Q_0Q_1 = 1 \Rightarrow Q_2 = 1$$

$$J_3 = K_3 = Q_0Q_1Q_2 = 0 \Rightarrow Q_3 = 0$$

$$\text{Estado} = Q_3Q_2Q_1Q_0 = 0100:$$

$$J_0 = K_0 = 1 \Rightarrow Q_0 = 1$$

$$J_1 = K_1 = Q_0 = 0 \Rightarrow Q_1 = 0$$

$$J_2 = K_2 = Q_0Q_1 = 0 \Rightarrow Q_2 = 1$$

$$J_3 = K_3 = Q_0Q_1Q_2 = 0 \Rightarrow Q_3 = 0$$

$$\text{Estado final} = Q_3Q_2Q_1Q_0 = 0100.$$

$$\text{Estado final} = Q_3Q_2Q_1Q_0 = 0101.$$

## Circuitos Sequenciais Contadores síncronos

b. Determinação dos Estados:

$$\text{Estado} = Q_3Q_2Q_1Q_0 = 0101:$$

$$J_0 = K_0 = 1 \Rightarrow Q_0 = 0$$

$$J_1 = K_1 = Q_0 = 1 \Rightarrow Q_1 = 1$$

$$J_2 = K_2 = Q_0Q_1 = 0 \Rightarrow Q_2 = 1$$

$$J_3 = K_3 = Q_0Q_1Q_2 = 0 \Rightarrow Q_3 = 0$$

$$\text{Estado} = Q_3Q_2Q_1Q_0 = 0110:$$

$$J_0 = K_0 = 1 \Rightarrow Q_0 = 1$$

$$J_1 = K_1 = Q_0 = 0 \Rightarrow Q_1 = 1$$

$$J_2 = K_2 = Q_0Q_1 = 0 \Rightarrow Q_2 = 1$$

$$J_3 = K_3 = Q_0Q_1Q_2 = 0 \Rightarrow Q_3 = 0$$

$$\text{Estado final} = Q_3Q_2Q_1Q_0 = 0110.$$

$$\text{Estado final} = Q_3Q_2Q_1Q_0 = 0111.$$

## Circuitos Sequenciais Contadores síncronos

b. Determinação dos Estados:

$$\text{Estado} = Q_3Q_2Q_1Q_0 = 0111:$$

$$J_0 = K_0 = 1 \Rightarrow Q_0 = 0$$

$$J_1 = K_1 = Q_0 = 1 \Rightarrow Q_1 = 0$$

$$J_2 = K_2 = Q_0Q_1 = 1 \Rightarrow Q_2 = 0$$

$$J_3 = K_3 = Q_0Q_1Q_2 = 1 \Rightarrow Q_3 = 1$$

$$\text{Estado} = Q_3Q_2Q_1Q_0 = 1000:$$

$$J_0 = K_0 = 1 \Rightarrow Q_0 = 1$$

$$J_1 = K_1 = Q_0 = 0 \Rightarrow Q_1 = 0$$

$$J_2 = K_2 = Q_0Q_1 = 0 \Rightarrow Q_2 = 0$$

$$J_3 = K_3 = Q_0Q_1Q_2 = 0 \Rightarrow Q_3 = 1$$

$$\text{Estado final} = Q_3Q_2Q_1Q_0 = 1000.$$

$$\text{Estado final} = Q_3Q_2Q_1Q_0 = 1001.$$

## Circuitos Sequenciais Contadores síncronos

b. Determinação dos Estados:

$$\text{Estado} = Q_3Q_2Q_1Q_0 = 1001:$$

$$J_0 = K_0 = 1 \Rightarrow Q_0 = 0$$

$$J_1 = K_1 = Q_0 = 1 \Rightarrow Q_1 = 1$$

$$J_2 = K_2 = Q_0Q_1 = 0 \Rightarrow Q_2 = 0$$

$$J_3 = K_3 = Q_0Q_1Q_2 = 0 \Rightarrow Q_3 = 1$$

$$\text{Estado} = Q_3Q_2Q_1Q_0 = 1010:$$

$$J_0 = K_0 = 1 \Rightarrow Q_0 = 1$$

$$J_1 = K_1 = Q_0 = 0 \Rightarrow Q_1 = 1$$

$$J_2 = K_2 = Q_0Q_1 = 0 \Rightarrow Q_2 = 0$$

$$J_3 = K_3 = Q_0Q_1Q_2 = 0 \Rightarrow Q_3 = 1$$

$$\text{Estado final} = Q_3Q_2Q_1Q_0 = 1010.$$

$$\text{Estado final} = Q_3Q_2Q_1Q_0 = 1011.$$

## Circuitos Sequenciais Contadores síncronos

b. Determinação dos Estados:

$$\text{Estado} = Q_3Q_2Q_1Q_0 = 1011:$$

$$J_0 = K_0 = 1 \Rightarrow Q_0 = 0$$

$$J_1 = K_1 = Q_0 = 1 \Rightarrow Q_1 = 0$$

$$J_2 = K_2 = Q_0Q_1 = 1 \Rightarrow Q_2 = 1$$

$$J_3 = K_3 = Q_0Q_1Q_2 = 0 \Rightarrow Q_3 = 1$$

$$\text{Estado} = Q_3Q_2Q_1Q_0 = 1100:$$

$$J_0 = K_0 = 1 \Rightarrow Q_0 = 1$$

$$J_1 = K_1 = Q_0 = 0 \Rightarrow Q_1 = 0$$

$$J_2 = K_2 = Q_0Q_1 = 0 \Rightarrow Q_2 = 1$$

$$J_3 = K_3 = Q_0Q_1Q_2 = 0 \Rightarrow Q_3 = 1$$

$$\text{Estado final} = Q_3Q_2Q_1Q_0 = 1100.$$

$$\text{Estado final} = Q_3Q_2Q_1Q_0 = 1101.$$

## Circuitos Sequenciais Contadores síncronos

b. Determinação dos Estados:

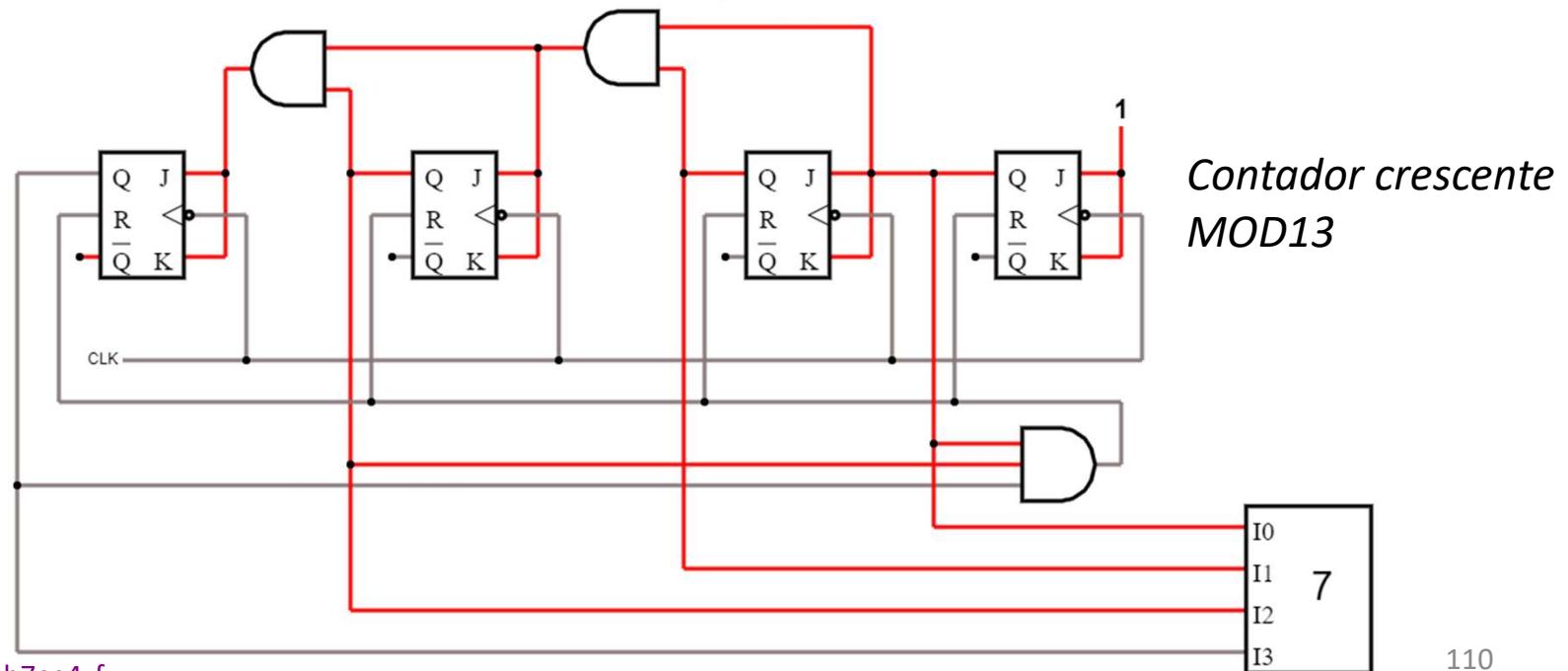
Estado =  $Q_3Q_2Q_1Q_0 = \textcolor{red}{1101}$ :

$$R_{0,1,2,3} = Q_3Q_2Q_0 = 1$$

Estado final =  $Q_3Q_2Q_1Q_0 = 0000$ .

## Circuitos Sequenciais Contadores síncronos

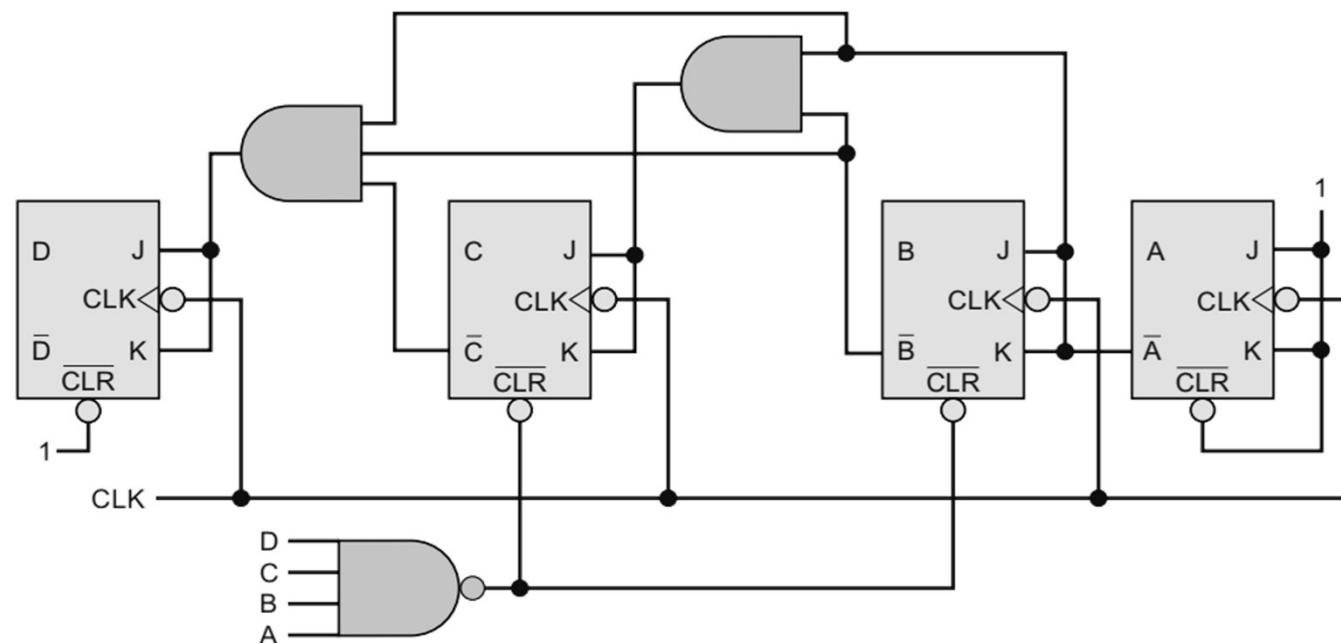
- i. Analisar o contador síncrono abaixo. Desenhe o seu diagrama de tempo e determine o seu módulo. [TOCCI 7.37]. Implementar em um simulador.



<https://tinyurl.com/yh7cp4zf>

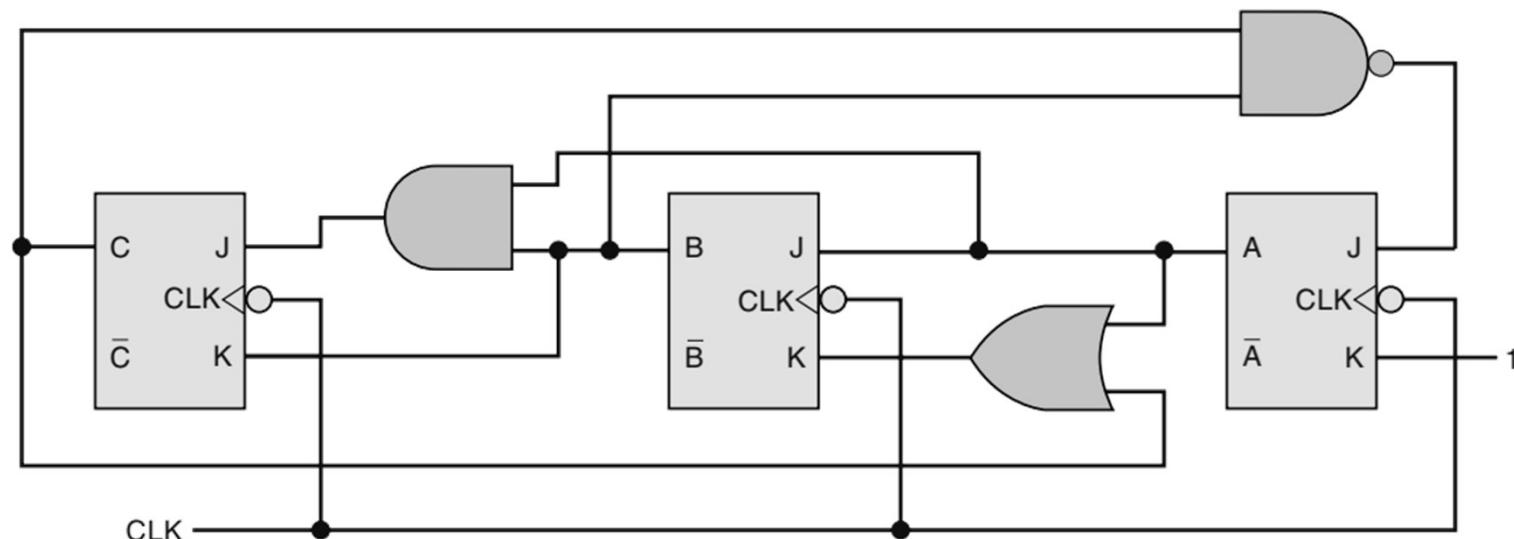
## Circuitos Sequenciais Contadores síncronos

- ii. Analisar o contador síncrono abaixo. Desenhe o seu diagrama de tempo e determine o seu módulo. [TOCCI 7.38]. Implementar em um simulador.



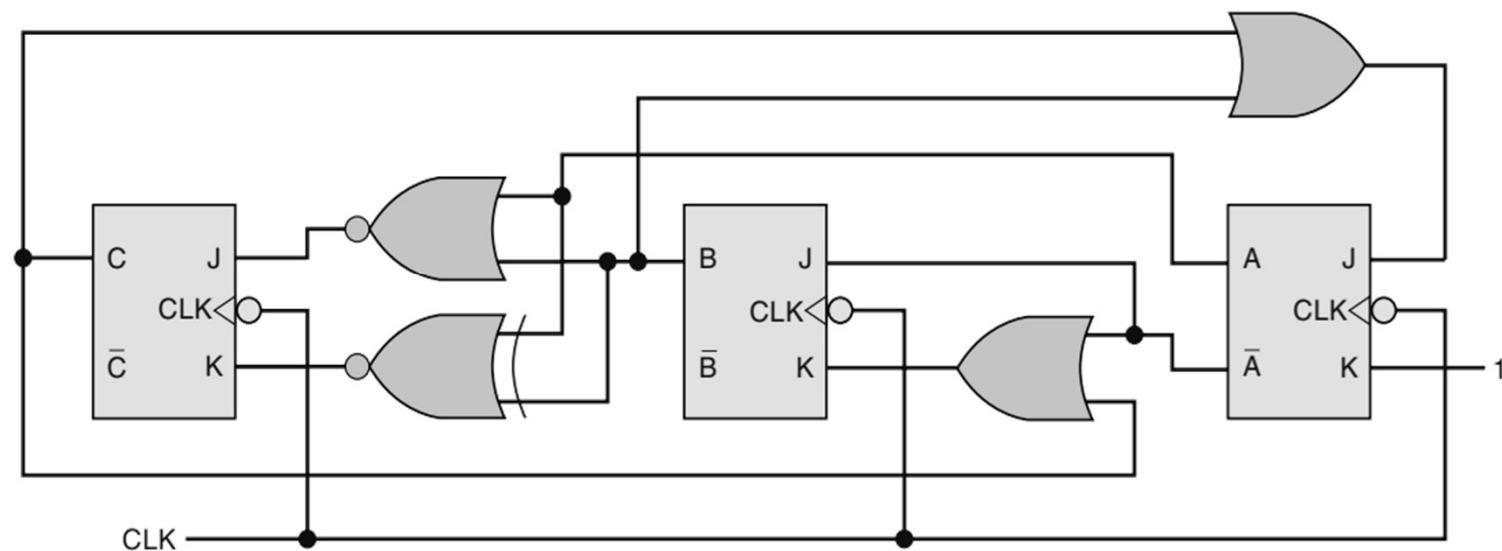
## Circuitos Sequenciais Contadores síncronos

- iii. Analisar o contador síncrono abaixo. Desenhe o seu diagrama de tempo e determine o seu módulo. [TOCCI 7.39]. Implementar em um simulador.



## Circuitos Sequenciais Contadores síncronos

- iv. Analisar o contador síncrono abaixo. Desenhe o seu diagrama de tempo e determine o seu módulo. [TOCCI 7.40]. Implementar em um simulador.



## Circuitos Sequenciais Contadores síncronos

- v. Analisar o contador síncrono 4 bits crescente e decrescente. Desenhe o seu diagrama de tempo. Implementar em um simulador.

