



INSTITUTO FEDERAL
Santa Catarina

Eletrônica Digital II

- Aula 3 -

Professora: Ma. Luciana Menezes Xavier de Souza
e-mail: luciana.xavier@ifsc.edu.br

Objetivos

- **Identificar e caracterizar registradores.**
- **Projetar e montar circuitos.**

Bibliografia

IDOETA, Ivan V.; CAPUANO, Francisco Gabriel. **Elementos de eletrônica digital**. Editora Erica, 2007.

MALVINO, Albert Paul. **Eletrônica digital**. Ed. 1. McGraw-Hill, 1987.

TOCCI, R. J.; WIDMER, N. S. **Sistemas Digitais: Princípios e Aplicações** . Ed. 8. LTC, 2003.

Informações Gerais

© A comunicação com a professora:

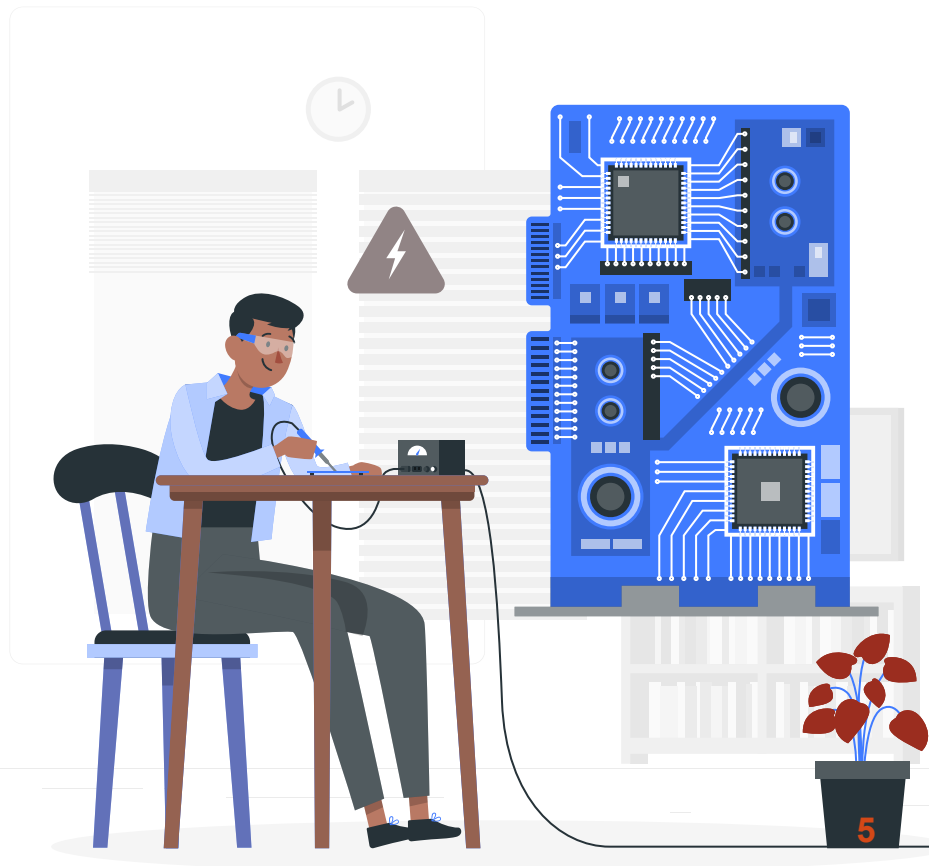
- Por e-mail:

e-mail da instituição

luciana.xavier@ifsc.edu.br

Conteúdo

- Introdução;
- Registradores;
- Exemplos.



Registradores

- O uso mais comum de flip-flops é no armazenamento de dados binários.
- Esses dados são geralmente armazenados em grupos de flip-flops denominados **registradores**.
- Basicamente, um **registrador** consiste em um **grupo de FF tipo D** que atuam no **armazenamento de dados binários**, pois um FF tem a capacidade de armazenar somente um bit, e de realizar a transferência dele.

O que são REGISTRADORES?

• O **dado binário dentro de um registrador** pode ser **movido** de um flip-flop para outro. Os registradores que permitem tal transferência são denominados de registradores de deslocamento (shift-register).

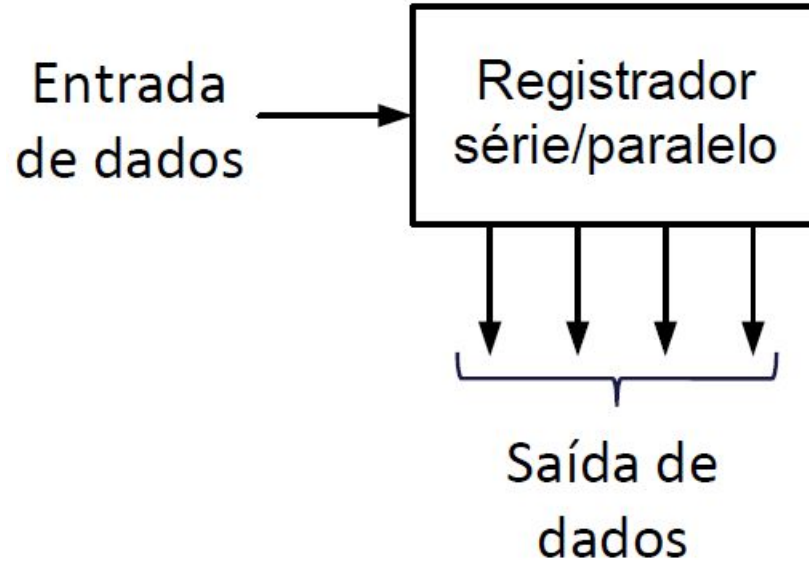
• Há 4 modos de operação básicos de um registrador:

- Registrador série/série
- Registrador série/paralelo
- Registrador paralelo/paralelo
- Registrador paralelo/série

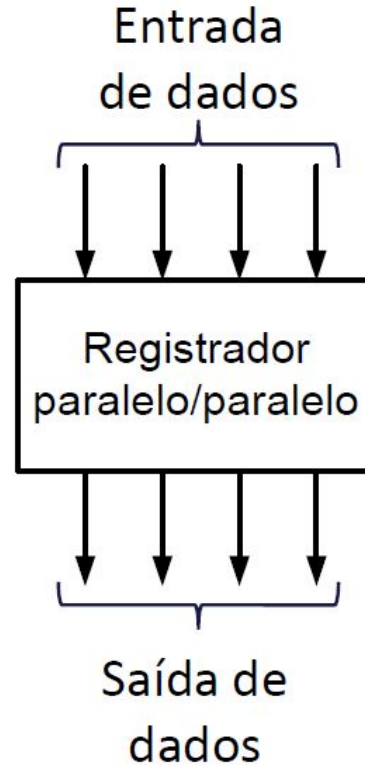
Registrador série/série



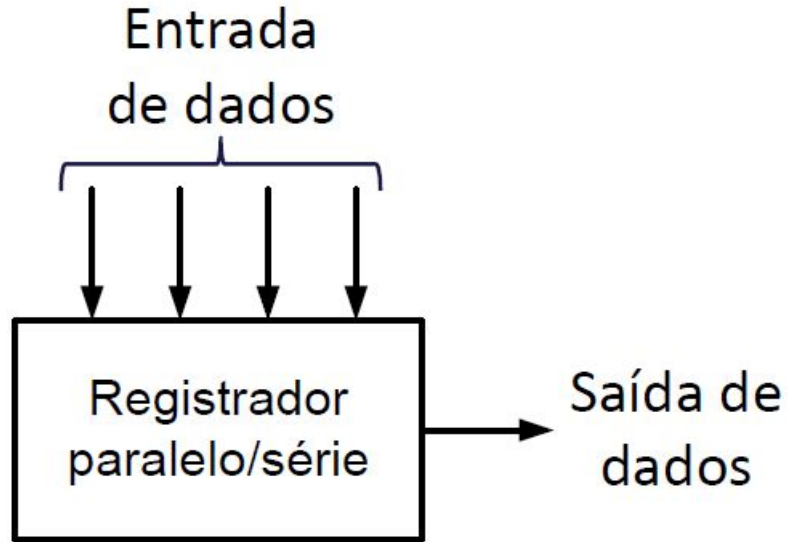
Registrador série/paralelo



Registrador paralelo/paralelo

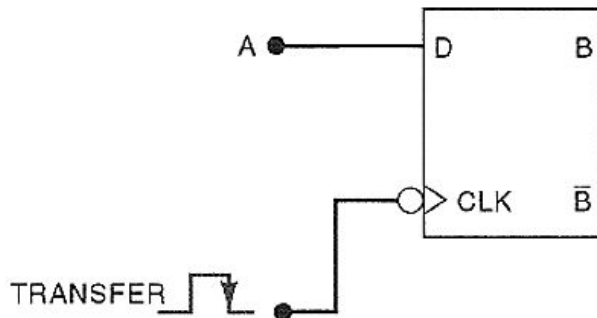


Registrador paralelo/série



Registradores

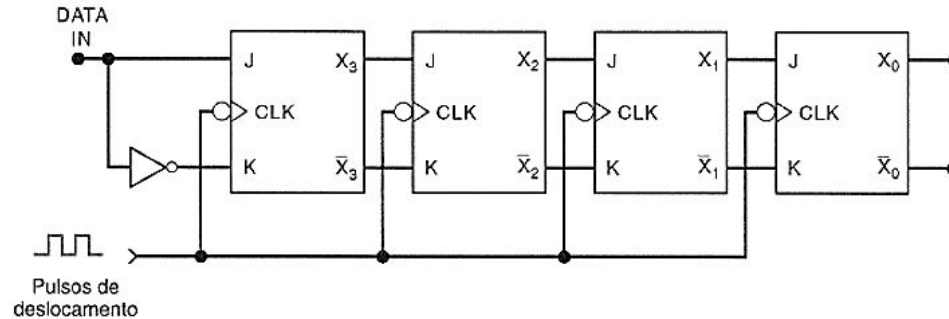
- FF tipo D no armazenamento de um dado de um bit:



- O valor lógico que está atualmente presente em A é transferido para o FF B na transição negativa (descida) do pulso transfer. Portanto, após esta transição, a **saída B terá o mesmo valor de A**.
- O grupo de FF é organizado de modo que os **números binários** a serem armazenados sejam **deslocados de um FF para o FF seguinte, a cada pulso de clock**.

Transferência Dados □ Registrador

Consiste em **inserir dados na entrada** do registrador, respeitando o número de bits, e efetuar o número de pulsos de clock necessários para que todo o **dado seja inserido no registrador**.

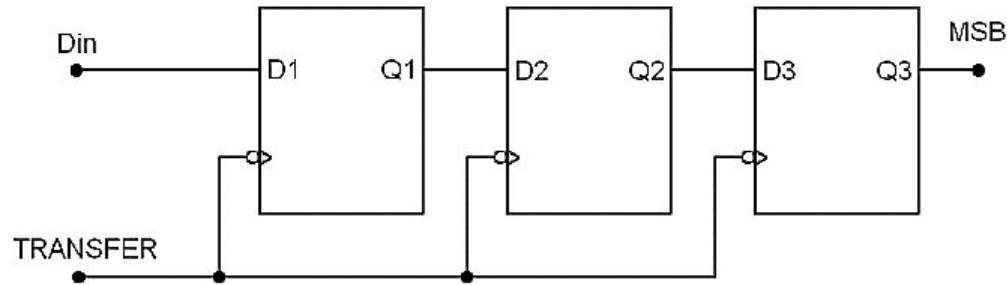


- O valor da saída X_3 é transferido para X_2 , o de X_2 para X_1 e o de X_1 para X_0 .
- Quando ocorrer uma transição (disparo na borda de descida), cada FF assumirá o valor armazenado anteriormente pelo FF que está à sua esquerda.

Transferência Dados □ Registrador

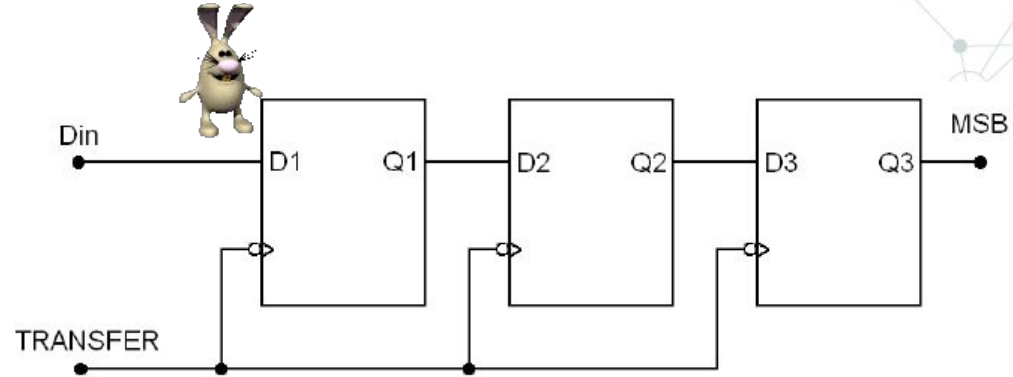
Exemplo:

Possuindo o dado 110_2 , escreva a tabela verdade da transferência de dados para o registrador da figura abaixo, considerando que inicialmente ele foi limpo.



Transferência Dados □ Registrador

1) Inicialmente:



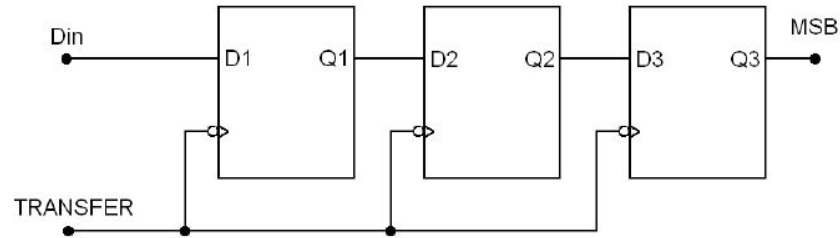
CLK	D1	D2	D3	Q1 (LSB)	Q2	Q3 (MSB)
—	1	0	0	0	0	0

2) 1º descida do Clock:

CLK	D1	D2	D3	Q1 (LSB)	Q2	Q3 (MSB)
	1	1	0	1	0	0

Transferência Dados □ Registrador

1) 2° descida do Clock:



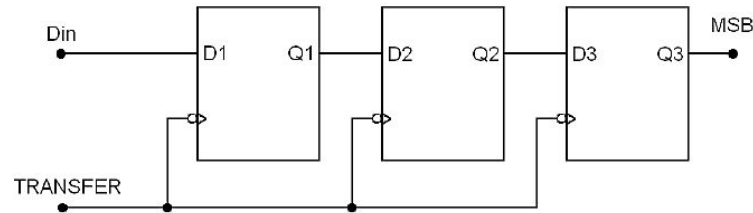
CLK	D1	D2	D3	Q1 (LSB)	Q2	Q3 (MSB)
	0	1	1	1	1	0

2) 3° descida do Clock:

CLK	D1	D2	D3	Q1 (LSB)	Q2	Q3 (MSB)
	X	0	1	0	1	1

Transferência Dados □ Registrador

Exemplo:

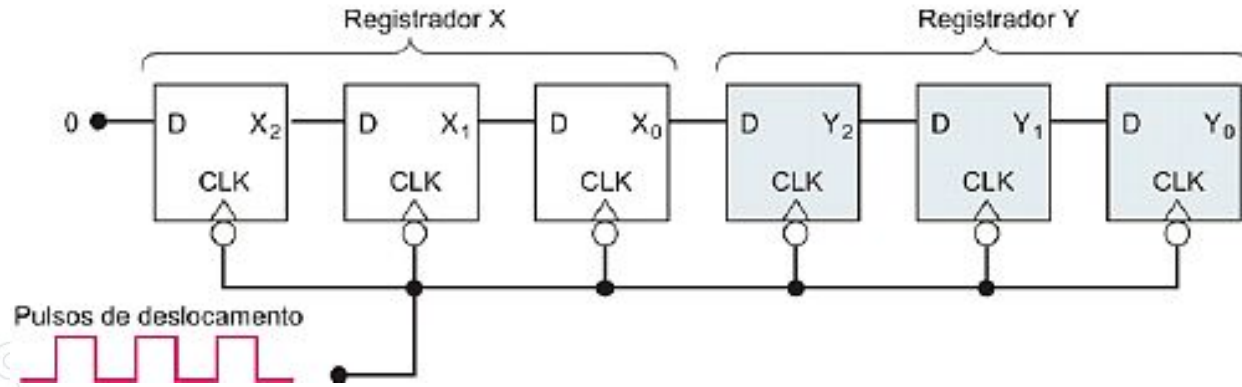


CLK	D1	D2	D3	Q1 (LSB)	Q2	Q3 (MSB)
—	1	0	0	0	0	0
	1	1	0	1	0	0
	0	1	1	1	1	0
	X	0	1	0	1	1

Transferência Registrador \square Registrador

Dois registradores de deslocamento de três bits conectados de tal modo que o conteúdo do registrador X seja transferido serialmente (deslocado) para o registrador Y.

Portanto, quando os pulsos de deslocamento são aplicados, a transferência de informação ocorre da seguinte forma: $X_2 \square X_1 \square X_0 \square Y_2 \square Y_1 \square Y_0$.



Transferência Registrador \square Registrador

Supondo que inicialmente temos o dado 101_2 armazenado no registrador X e que o registrador Y foi limpo, temos a seguinte tabela verdade abaixo:

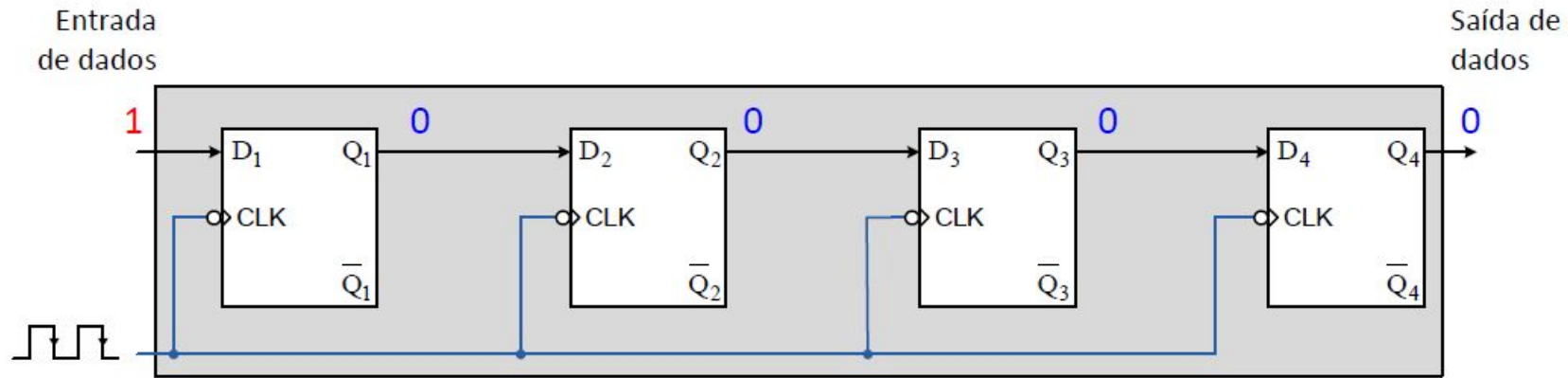
X_2	X_1	X_0	Y_2	Y_1	Y_0	
1	0	1	0	0	0	← Antes de os pulsos serem aplicados
0	1	0	1	0	0	← Depois do primeiro pulso
0	0	1	0	1	0	← Depois do segundo pulso
0	0	0	1	0	1	← Depois do terceiro pulso

Transferência Registrador \square Registrador

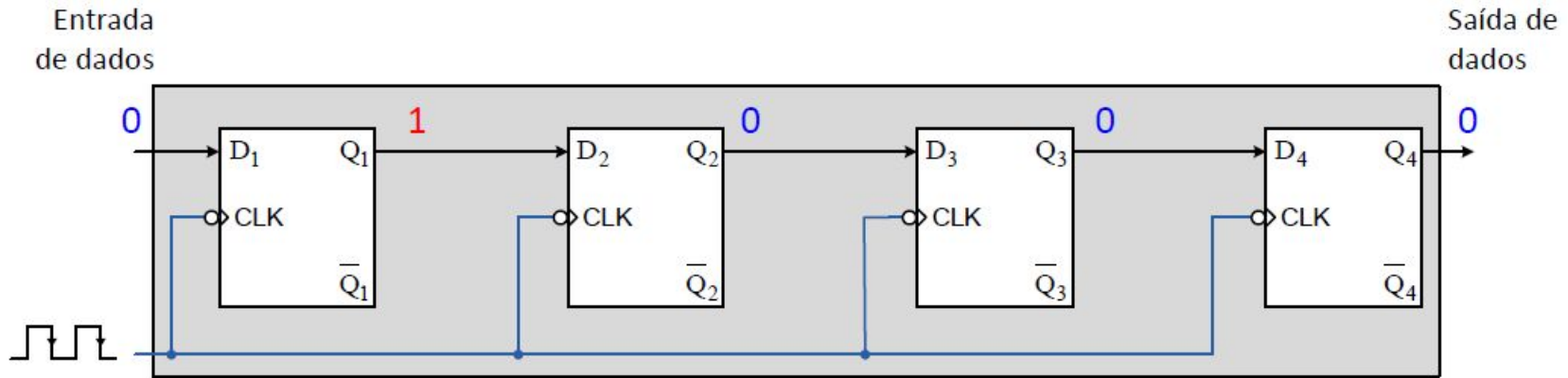
A partir dessa tabela podemos concluir que:

- \square A **cada descida** do pulso de clock, **cada FF assume o valor** que foi armazenado no FF à sua **esquerda**, antes da ocorrência do pulso.
- \square **Após 3 pulsos, todo** o conteúdo presente no registrador **X está** presente no registrador **Y**.
- \square Portanto, a **transferência completa de 3 bits** necessita de **3 pulsos** de deslocamento.

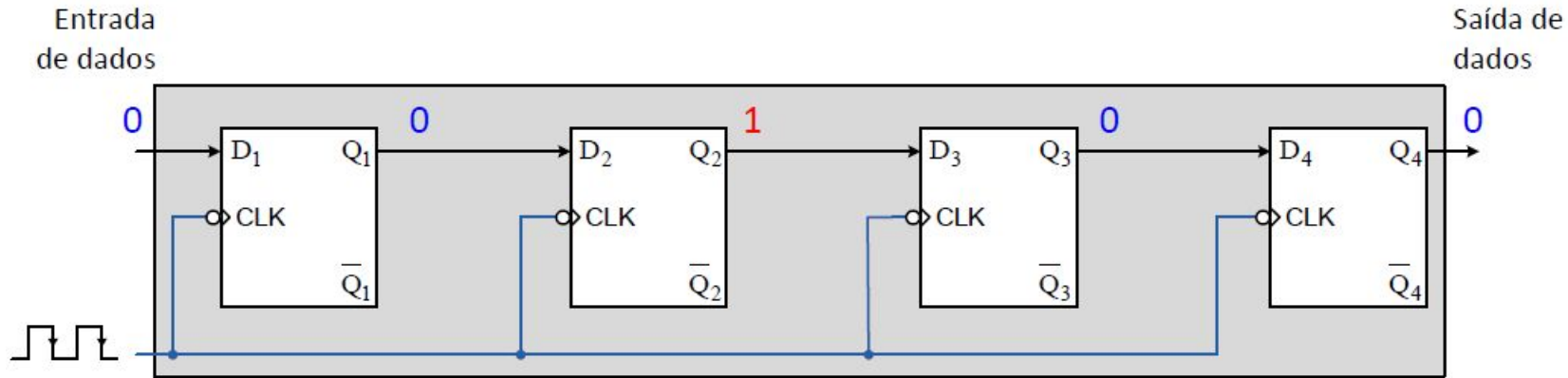
Exemplo: Com o registrador série/série leve o bit 1 até a saída de dados



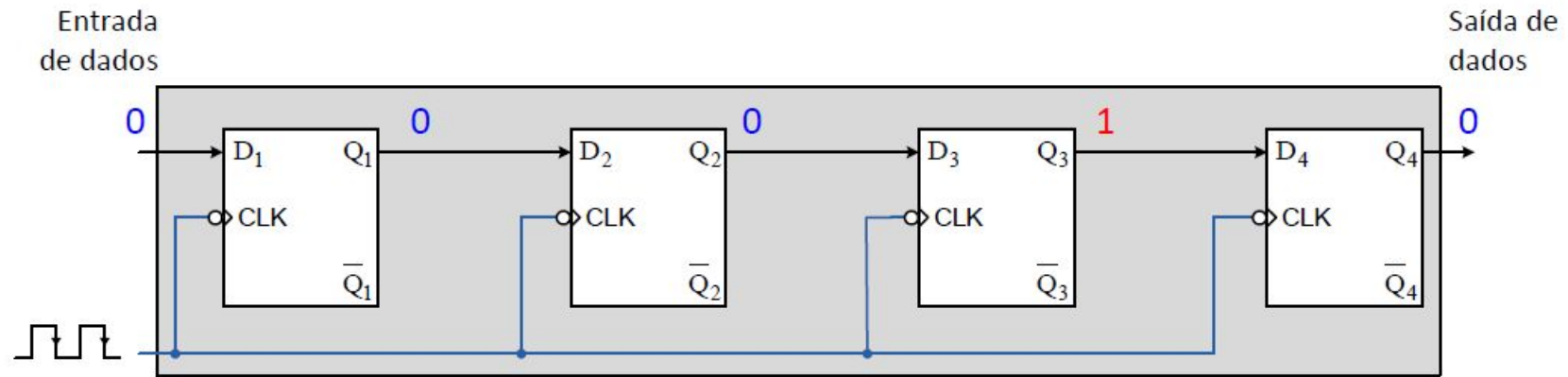
Exemplo: Leve o bit 1 até a saída de dados



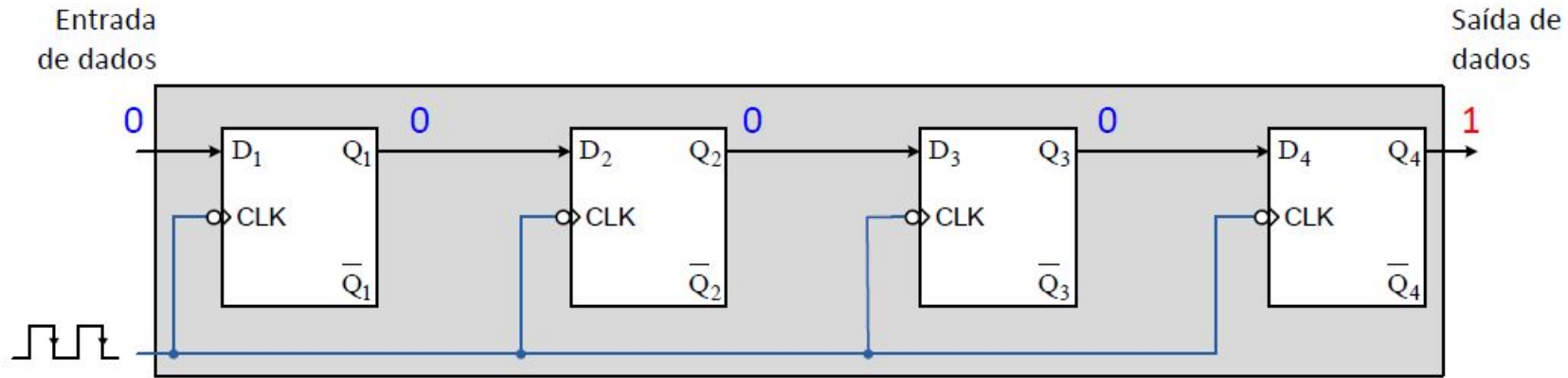
Exemplo: Leve o bit 1 até a saída de dados



Exemplo: Leve o bit 1 até a saída de dados



Exemplo: Leve o bit 1 até a saída de dados

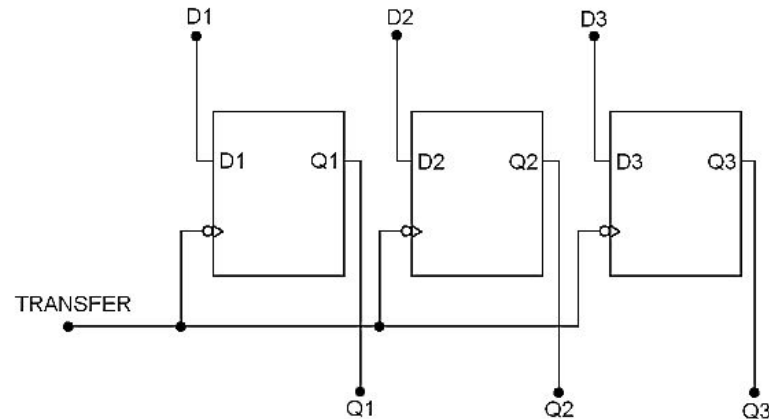


Transferência Paralela de Dados

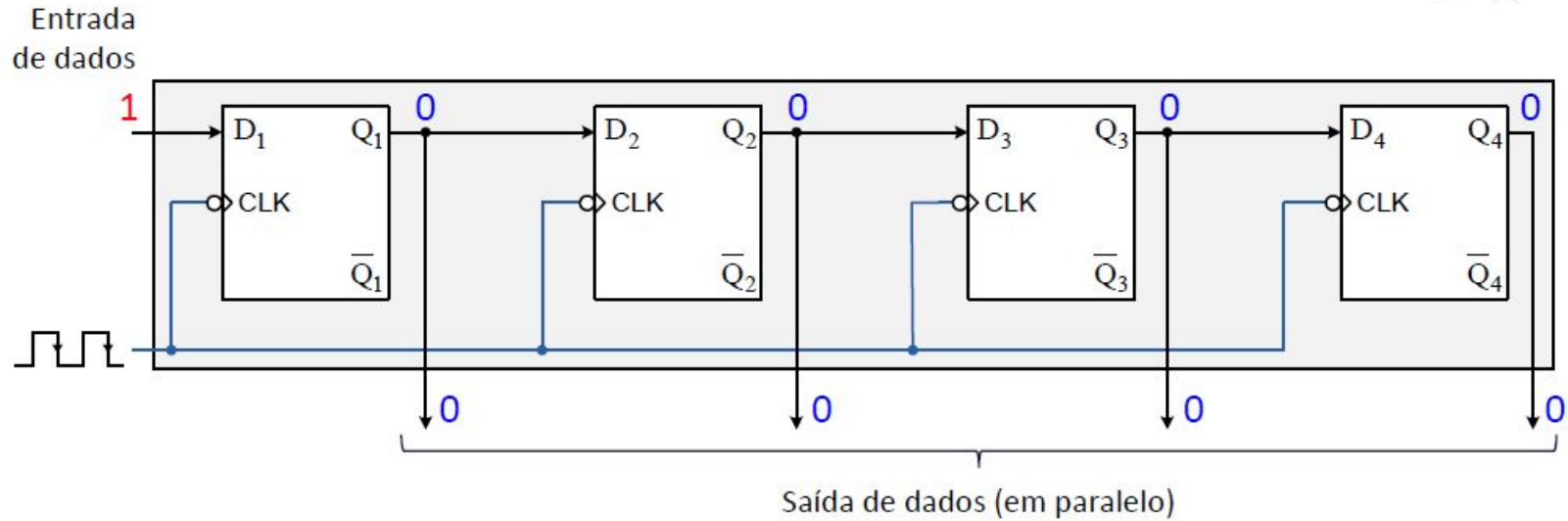
O grupo de FF é organizado de maneira que o dado binário a ser armazenado seja transferido simultaneamente para todos os FF, com a aplicação de apenas 1 pulso de transferência ou clock.

Transferência Dados □ Registrador

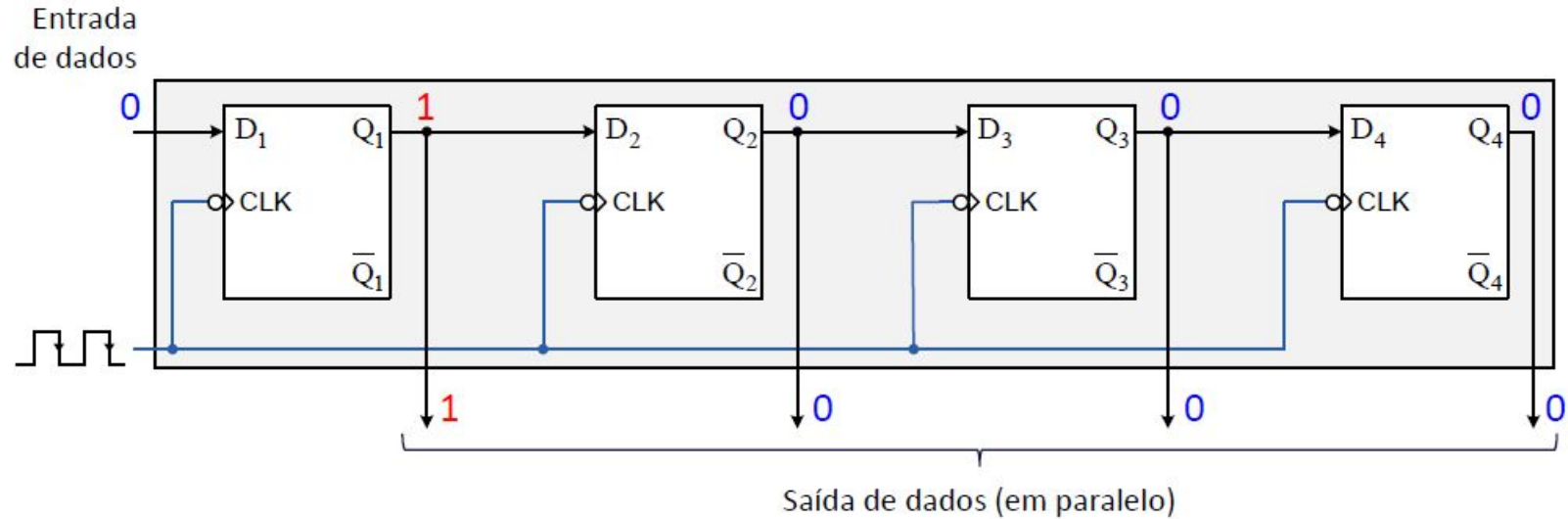
Consiste em inserir o dado a ser armazenado diretamente na entrada do registrador, efetuando-se 1 pulso de transferência.



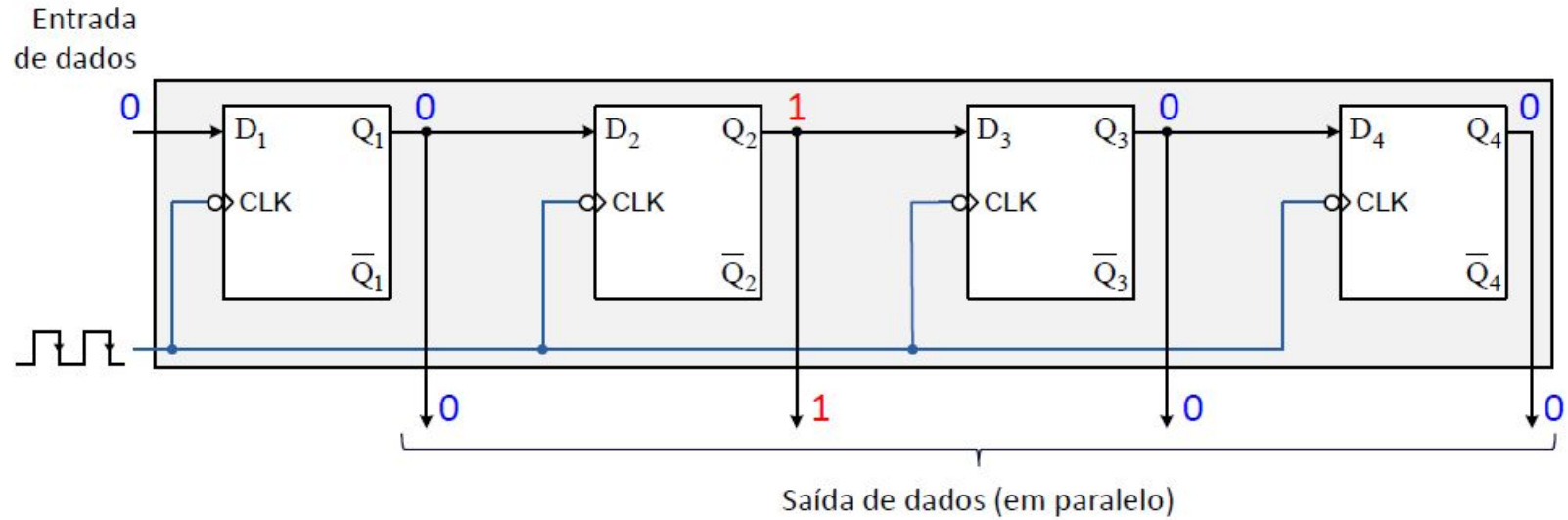
Exemplo: Com o registrador **série/paralelo** leve o bit 1 até a saída de dados



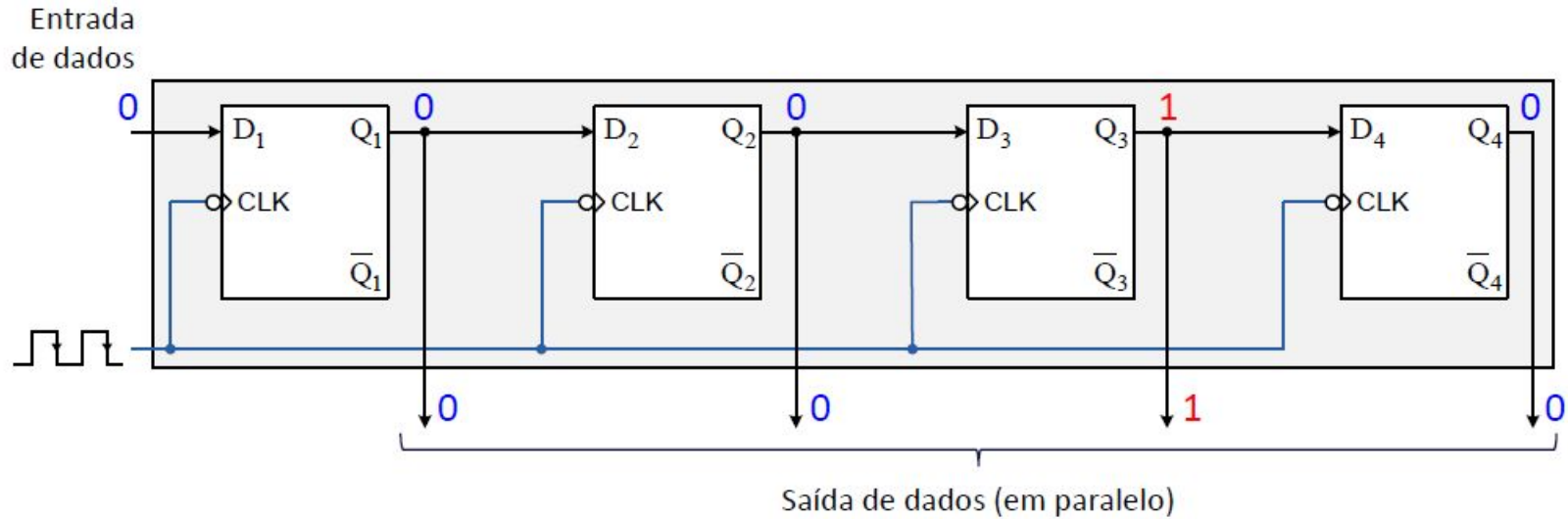
Exemplo: Com o registrador série/paralelo leve o bit 1 até a saída de dados



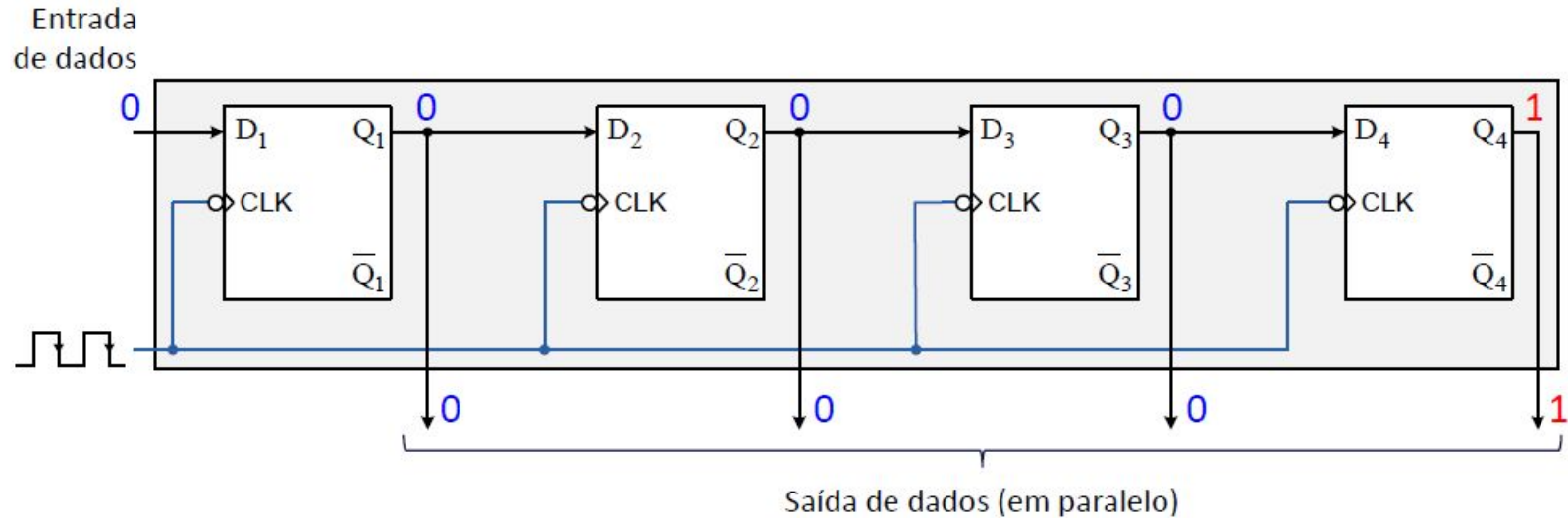
Exemplo: Com o registrador série/paralelo leve o bit 1 até a saída de dados



Exemplo: Com o registrador série/paralelo leve o bit 1 até a saída de dados



Exemplo: Com o registrador série/paralelo leve o bit 1 até a saída de dados

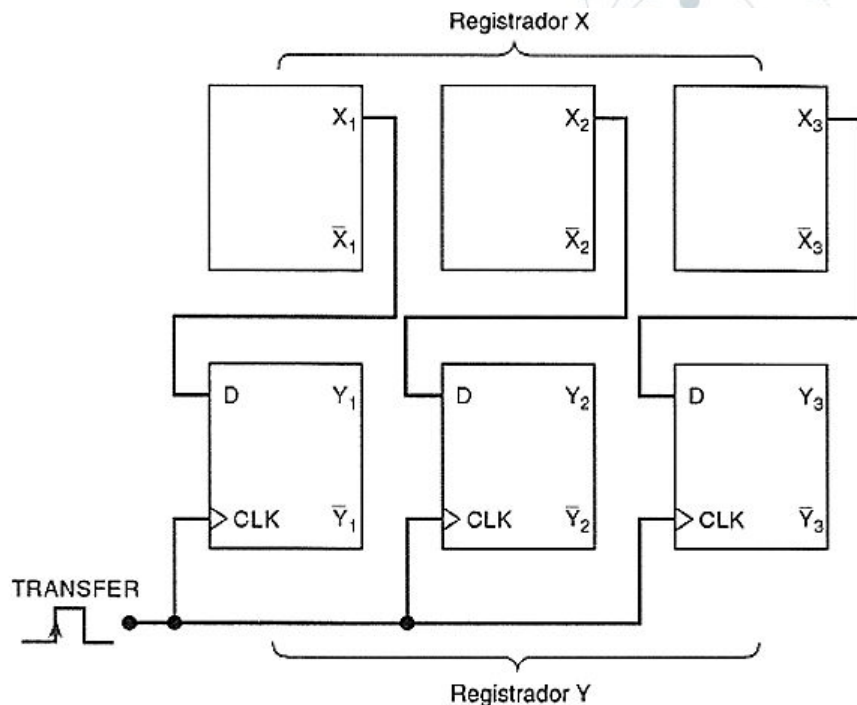


Transferência Paralela de Dados

Transferência Registrador \square Registrador

Dois registradores, X e Y, interligados para executar uma transferência paralela de dados, ou seja, após a aplicação de 1 pulso de transferência, temos todo o conteúdo de X armazenado também em Y.

A **transferência paralela** de dados entre registradores **não altera o conteúdo da fonte**, enquanto na **transferência serial altera o gradativamente o valor do registrador** que atua como fonte de dados.



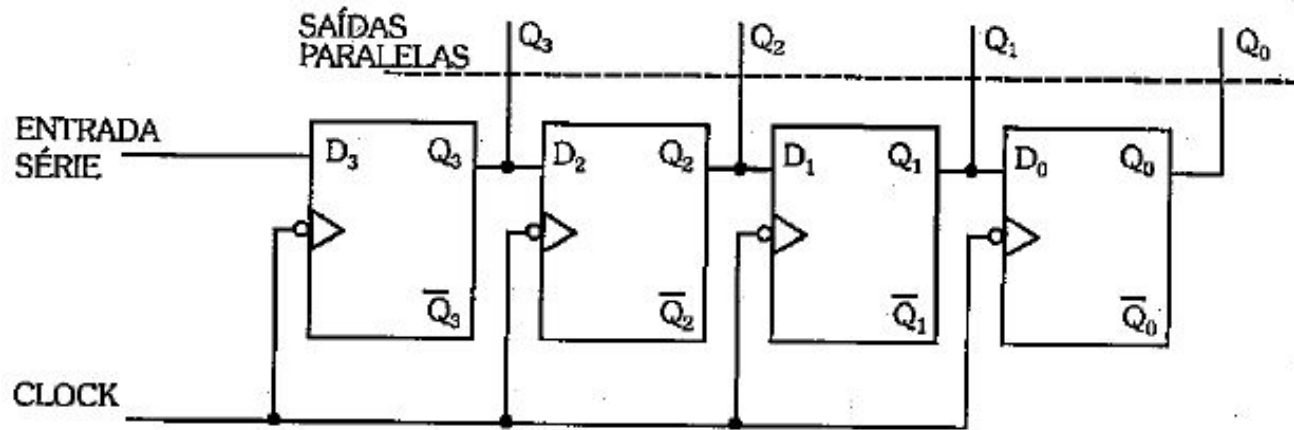
Transferência Serial x Paralela de Dados

A escolha de um tipo particular de transferência, serial ou paralela, depende da aplicação e das especificações fornecidas.

PARALELA	SERIAL
1) Todas as informações são transferidas simultaneamente na ocorrência de um único pulso de transferência, não importando o número de bits que estejam sendo transferidos → MAIOR VELOCIDADE.	1) A transferência completa de N bits, necessita de N pulsos de clock → MENOR VELOCIDADE.
2) Requer um maior número de conexões entre TX e RX → MAIOR CUSTO.	2) Necessita de apenas uma conexão entre TX e RX → MENOR CUSTO.

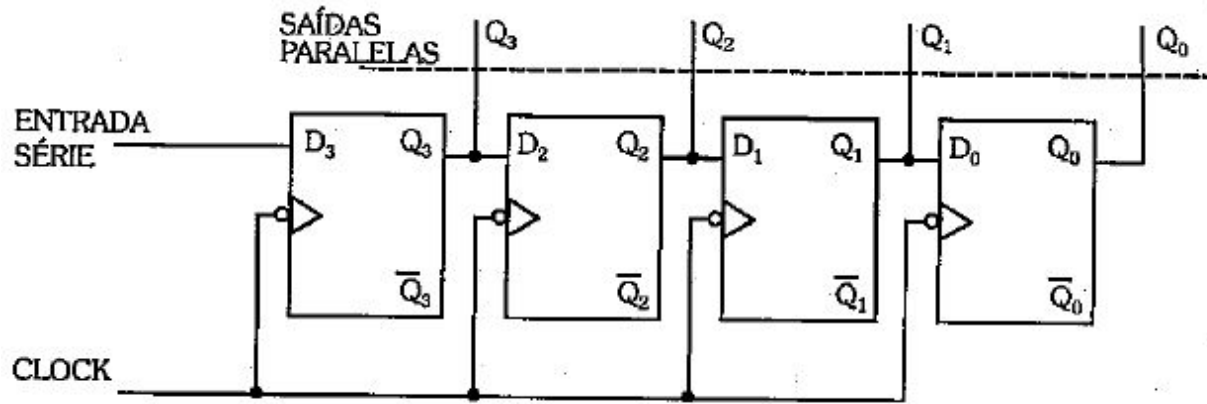
Registrador de Deslocamento □ Conversor

O Registrador de Deslocamento pode ser utilizado para **converter uma informação série em paralela**, ou seja, funciona como Conversor Série Paralelo.



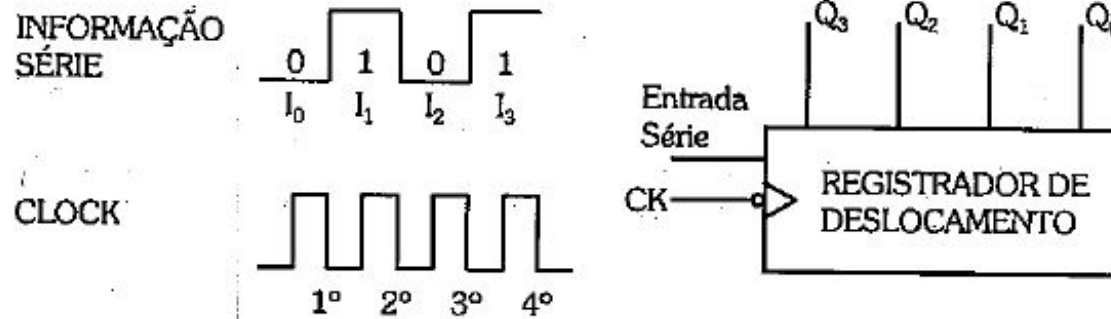
Conversor Série - Paralelo

Exemplo: Vamos aplicar a informação série $I = 1010$ ($I_3 I_2 I_1 I_0$) à entrada série do registrador e analisar as saídas Q_0 , Q_1 , Q_2 e Q_3 , após os pulsos de clock.



Conversor Série - Paralelo

Entraremos com a informação (1010) como mostrado na figura abaixo.

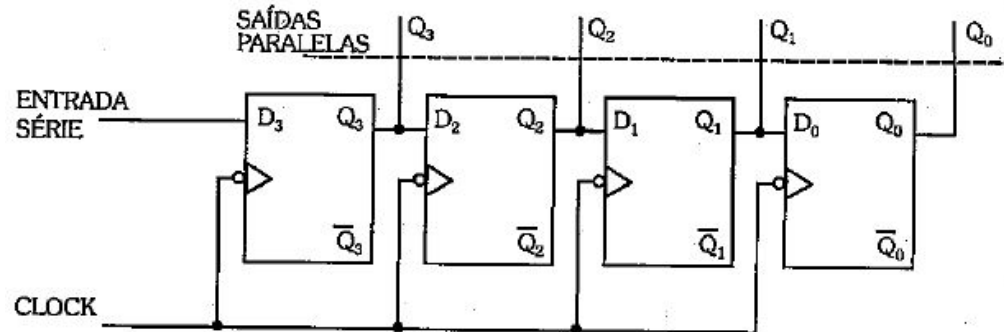


Supondo que inicialmente as saídas Q_3 , Q_2 , Q_1 e Q_0 do registrador estejam em nível 0.

Conversor Série - Paralelo

Ao injetarmos na entrada o 1º bit de informação ($I_0=0$) e houver a **descida do pulso de clock**, o flip-flop 3 irá apresentar a saída 0 ($D_3 = 0 \Rightarrow Q_3 = 0$).

Após este pulso de clock, aparecerá na entrada, o bit seguinte de informação ($I_1=1$) e na descida do **2º pulso de clock**, teremos a passagem de I_0 para o flip-flop 2 ($D_2 = 0 \Rightarrow Q_2 = 0$) e Q_3 assumirá o valor do bit de informação I_1 (entrada série = $D_3 = 1 \Rightarrow Q_3 = 1$).



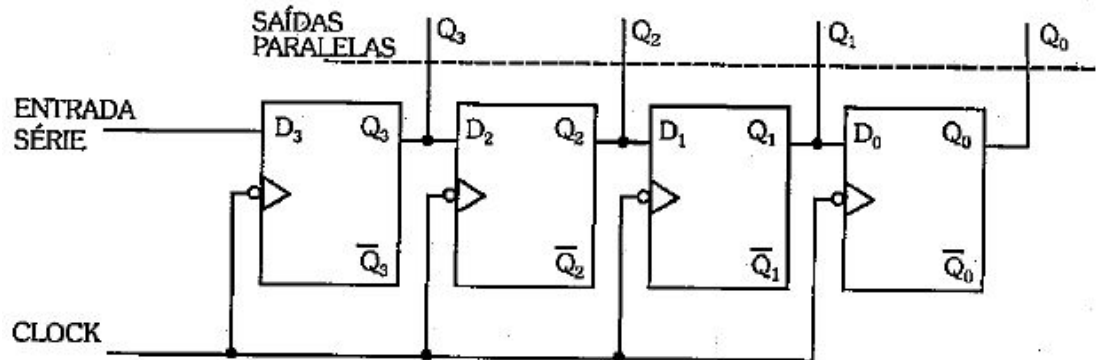
Conversor Série - Paralelo

Após a descida do 3º pulso de clock, ficaremos com a seguinte situação:

- $Q_1 = 0$ ($D_1 = Q_2 = 0$ □ $Q_1 = 0$),
- $Q_2 = 1$ ($D_2 = Q_3 = 1$ □ $Q_2 = 1$) e
- $Q_3 = 0$ ($D_3 = I_2 = 0$ □ $Q_3 = 0$),

Após a descida do 4º pulso de clock, teremos a seguinte situação:

- $Q_0 = 0$ ($D_0 = Q_1 = 0$ □ $Q_0 = 0$),
- $Q_1 = 1$ ($D_1 = Q_2 = 1$ □ $Q_1 = 1$),
- $Q_2 = 0$ ($D_2 = Q_3 = 0$ □ $Q_2 = 0$) e
- $Q_3 = 1$ ($D_3 = I_3 = 1$ □ $Q_3 = 1$),



Conversor Série - Paralelo

Após o 4º pulso de clock, a informação I estará armazenada no registrador de deslocamento e aparecerá nas saídas Q_3 , Q_2 , Q_1 e Q_0 como sendo uma informação paralela.

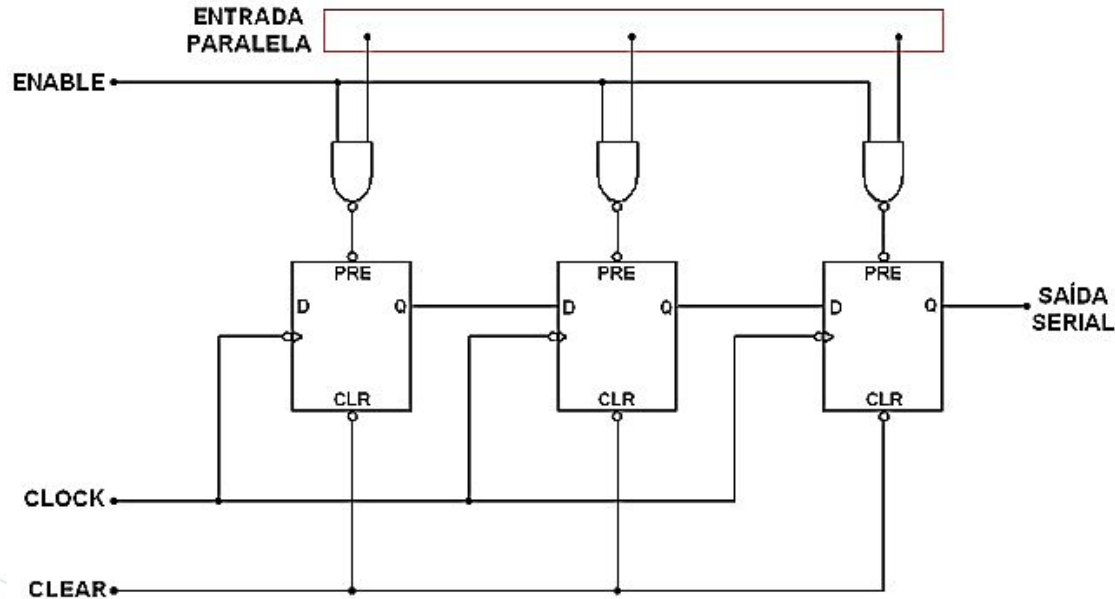
Sequência sob a forma da tabela da verdade:

Informação	Descida de clock	Q_3	Q_2	Q_1	Q_0
$I_0 = 0$	1ª	0	0	0	0
$I_1 = 1$	2ª	1	0	0	0
$I_2 = 0$	3ª	0	1	0	0
$I_3 = 1$	4ª	1	0	1	0

Por deslocar a informação a cada pulso de clock □ Registrador de Deslocamento.

Conversor Paralelo - Série

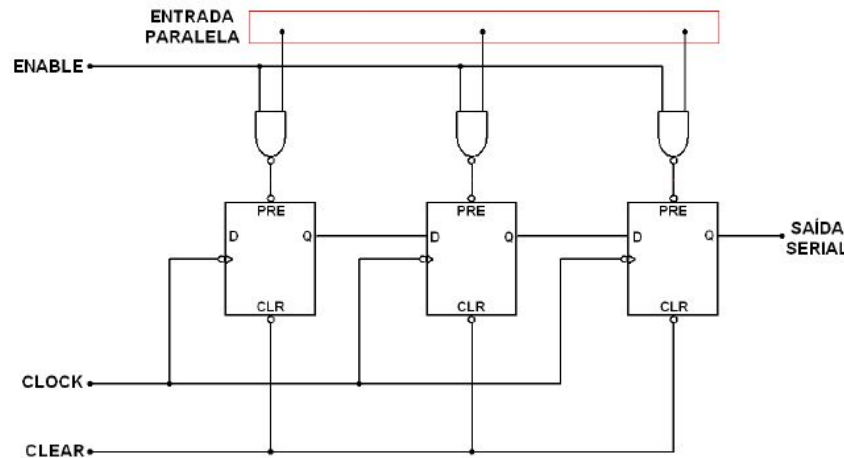
Para **entrarmos** com uma informação **paralela**, precisamos de um registrador que apresente entradas *Preset* e *Clear*, pois é através destas que fazemos com que o **registrador armazene a informação paralela**.



Conversor Paralelo - Série

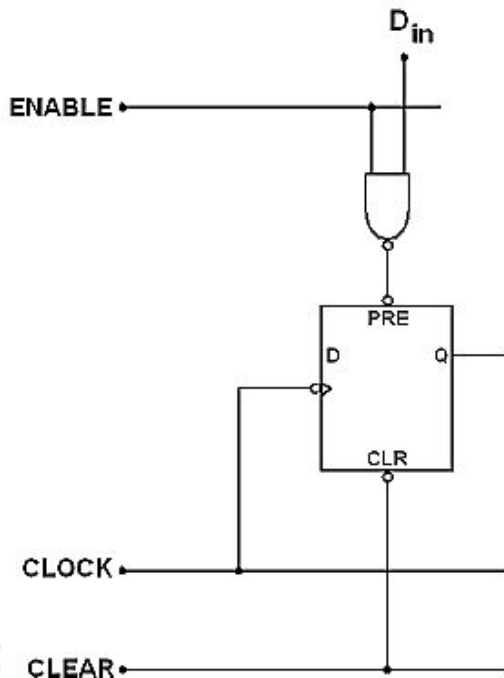
Quando a entrada **Enable** estiver em **0**, as entradas Preset (**PRE**) dos flip-flops assumirão, respectivamente, **níveis 1**, fazendo com que o **registrador atue normalmente**.

Quando a entrada **Enable** estiver em **1**, as entradas Preset (**PRE**) dos flip-flops assumirão os **valores complementares** das entradas PRE, logo os flip-flops irão assumir os valores que estiverem, respectivamente, em cada entrada PRE.



Conversor Paralelo - Série

Para um melhor entendimento vamos analisar uma célula do registrador:



- Para que esse conversor funcione, é necessário inicialmente um pulso de nível 0 na entrada **CLEAR** dos **FF**.
- Com **ENABLE = 0**, a entrada **PRE** assumirá nível 1 e o **FF** irá funcionar como um **registrador de deslocamento comum**.
- Com **ENABLE = 1** e **Din = 0**, **PRE** assumirá nível 1, logo, a saída **Q** manterá seu estado anterior (que era 0).
- Com **ENABLE = 1** e **Din = 1**, **PRE** assumirá nível 0, logo a saída **Q** assumirá valor 1.

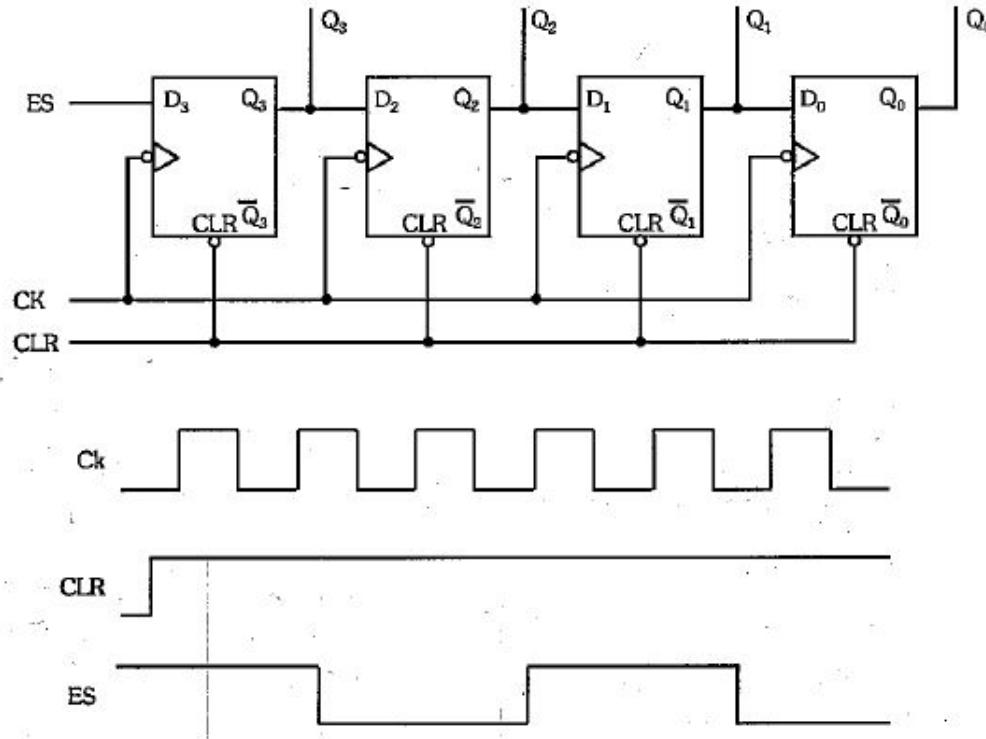
Conversor Paralelo - Série

Após essa análise, concluímos que:

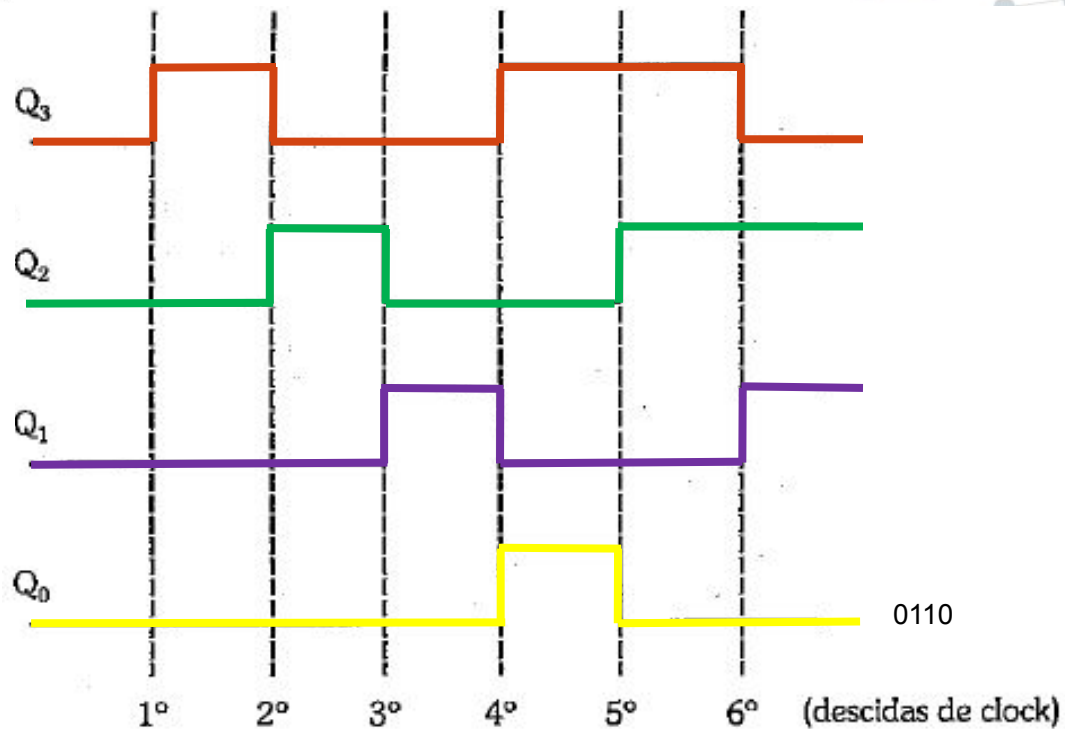
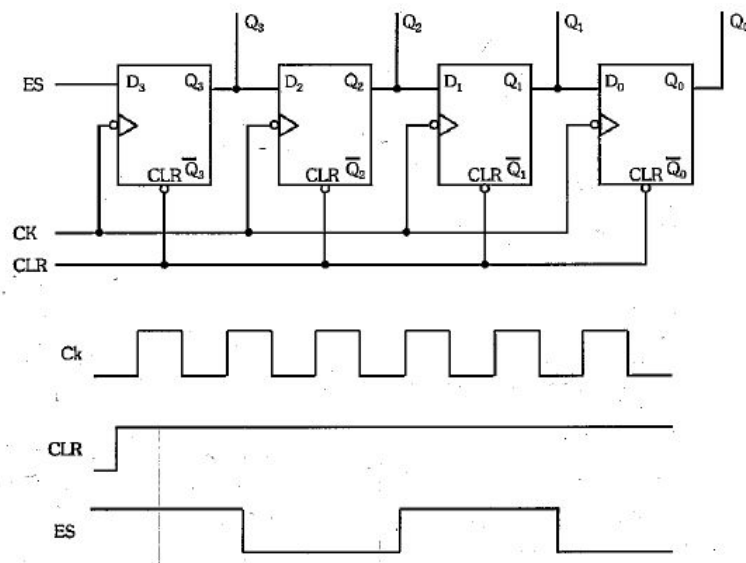
- Se **zerarmos** o registrador e logo após introduzirmos a informação paralela, as **saídas Q** dos FF **assumirão** os valores inseridos respectivamente nessas **entradas**.
- Depois de inserida paralelamente essa informação, basta colocarmos **ENABLE = 0** novamente e prosseguir como um registrador de deslocamento comum.
- Com esse conjunto obtivemos um registrador com Entrada Paralela e Saída Serial (EPSS).

Exercício

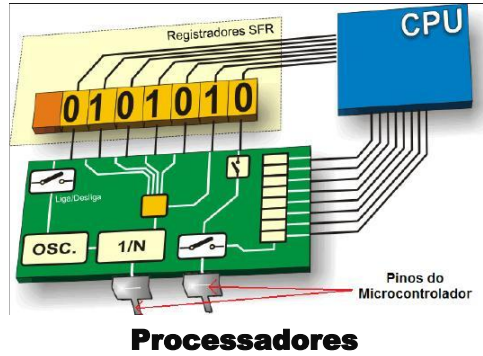
A partir dos sinais aplicados às entradas, esboce as formas de onda das saídas para o Registrador de Deslocamento de bits, visto abaixo.



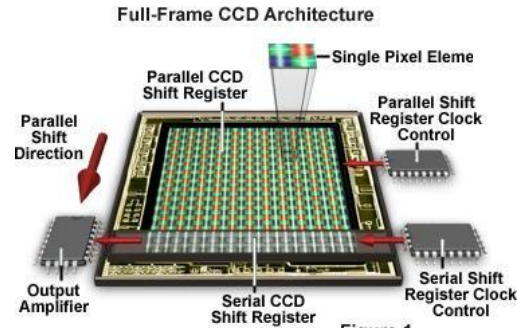
Resposta



Registrador de Deslocamento



Aplicações



Captura de Imagens

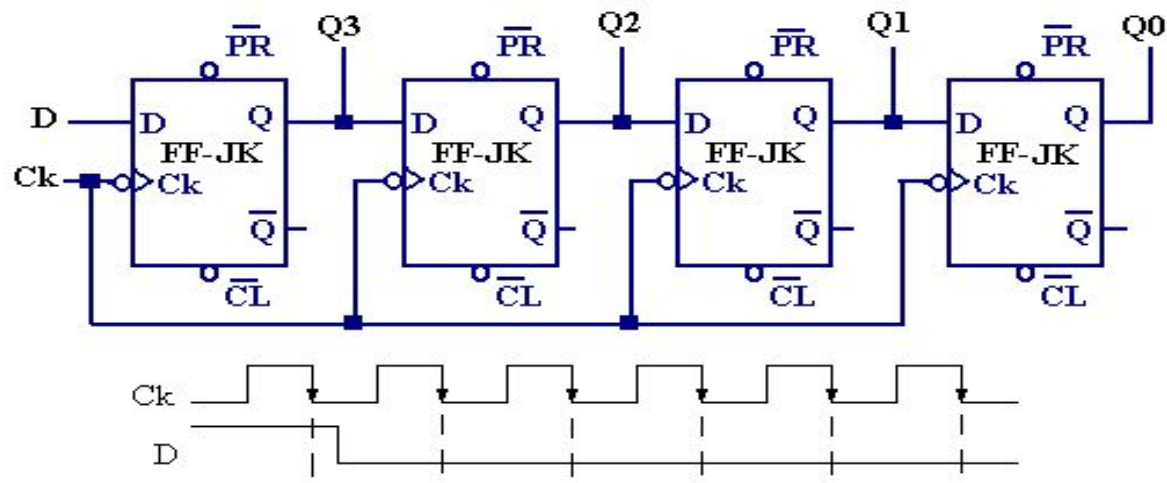


Comunicação de Periféricos

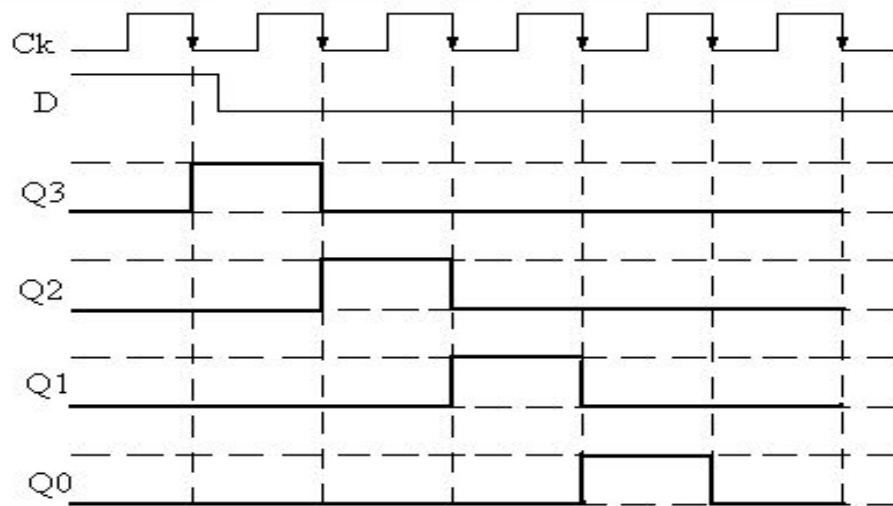
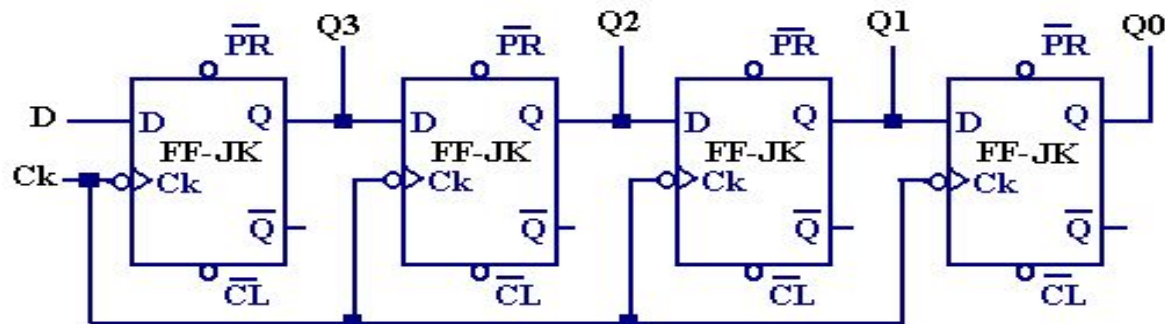


Exercícios

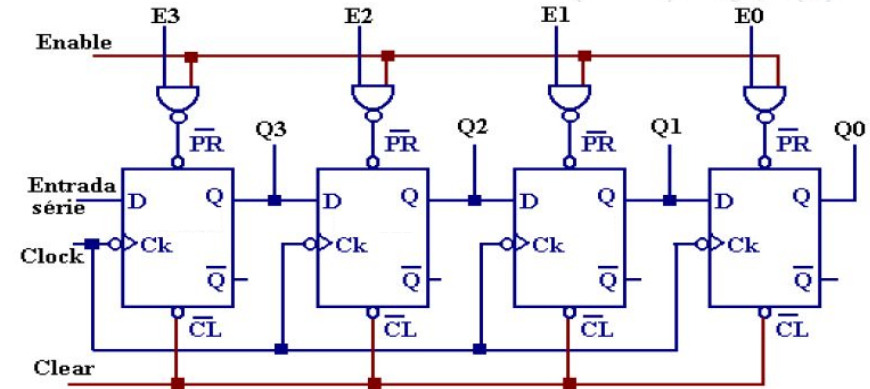
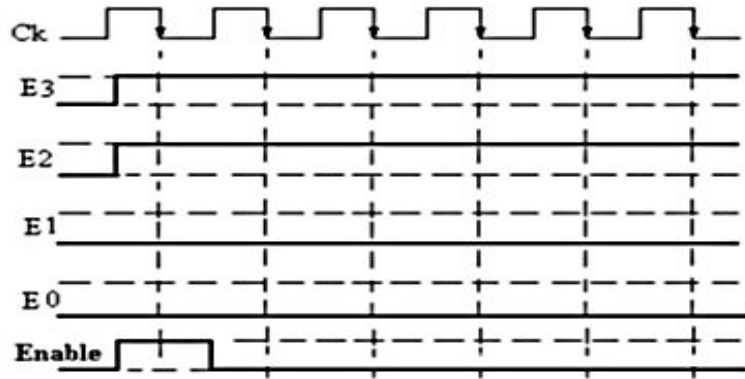
Registrador Série-Paralelo: Encontre as formas de onda de Q3, Q2, Q1 e Q0



Registrador Série-Paralelo: Encontre as formas de onda de Q3, Q2, Q1 e Q0



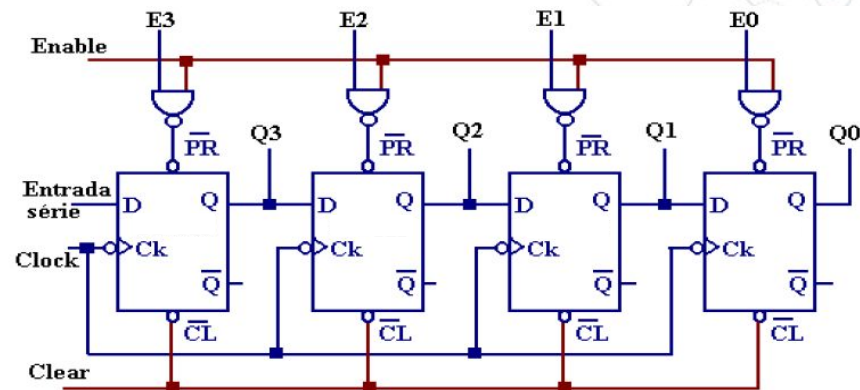
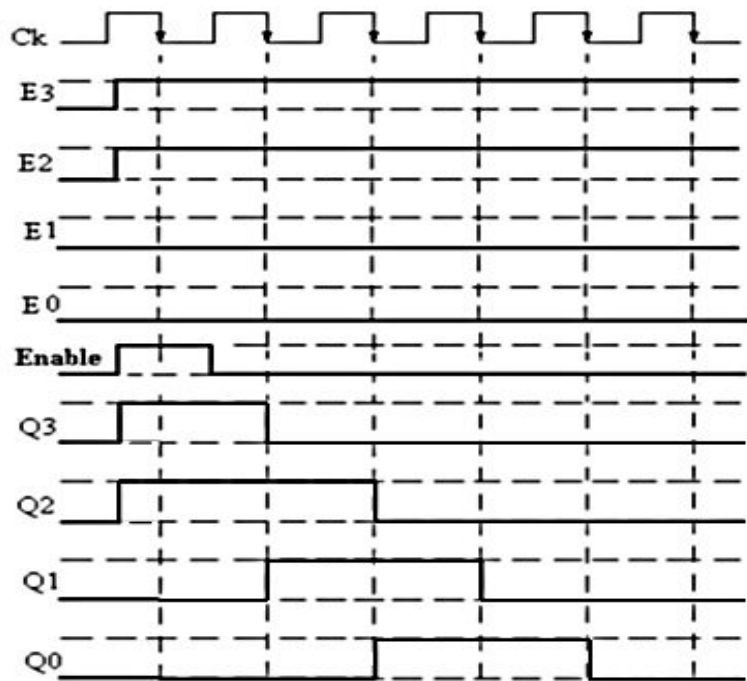
Registrador Paralelo-Paralelo: Encontre as formas de onda de Q3, Q2, Q1 e Q0



Enable = 0 – **Habilita entrada Série**

Enable = 1 – **Habilita as entradas Paralelas**

Registrador Paralelo-Paralelo: Encontre as formas de onda de Q3, Q2, Q1 e Q0



Enable = 0 – Habilita entrada Série

Enable = 1 – Habilita as entradas Paralelas