

Circuitos Digitais

Prof. Esp. Pedro Luís Antonelli
Anhanguera Educacional




Plano de Ensino e Aprendizagem (PEA)

Cronograma de Aulas

Semana n°.	Tema
1	Apresentação da Disciplina e Metodologia de Trabalho. Revisão de Aritmética Binária.
2	Propriedade da Álgebra de Boole. Elementos Lógicos Básicos.
3	Minimização e Implementação de Funções.
4	Dispositivos Básicos de Circuitos Combinacionais: Decodificadores, Codificadores.
5	Dispositivos Básicos de Circuitos Combinacionais: Decodificadores, Codificadores.
6	Dispositivos Básicos de Circuitos Combinacionais: Somadores, Subtratores.
7	Dispositivos Básicos de Circuitos Combinacionais: Comparadores.
8	Dispositivos Básicos de Circuitos Combinacionais: Multiplexadores, Demultiplexadores.
9	Atividades de Avaliação.
10	Flip-Flops.
11	Flip-Flops.
12	Conceitos de Controle e Temporização.
13	Análise e Síntese de Circuitos Seqüenciais.
14	Análise e Síntese de Circuitos Seqüenciais.
15	Dispositivos Básicos de Circuitos Seqüenciais: Registradores de Deslocamento.
16	Dispositivos Básicos de Circuitos Seqüenciais: Contadores.
17	Dispositivos Básicos de Circuitos Seqüenciais: Dispositivos Lógicos Programáveis ou PLD, Memórias: ROM, PROM, EPROM, EEPROM, RAM.
18	Prova Escrita Oficial.
19	Exercícios de Revisão.
20	Prova Substitutiva.

OBJETIVOS DA AULA :

- Continuar o estudo dos Circuitos Sequenciais;
 - Circuitos Registradores de Deslocamento.
 - Circuitos Contadores Assíncronos;
- 

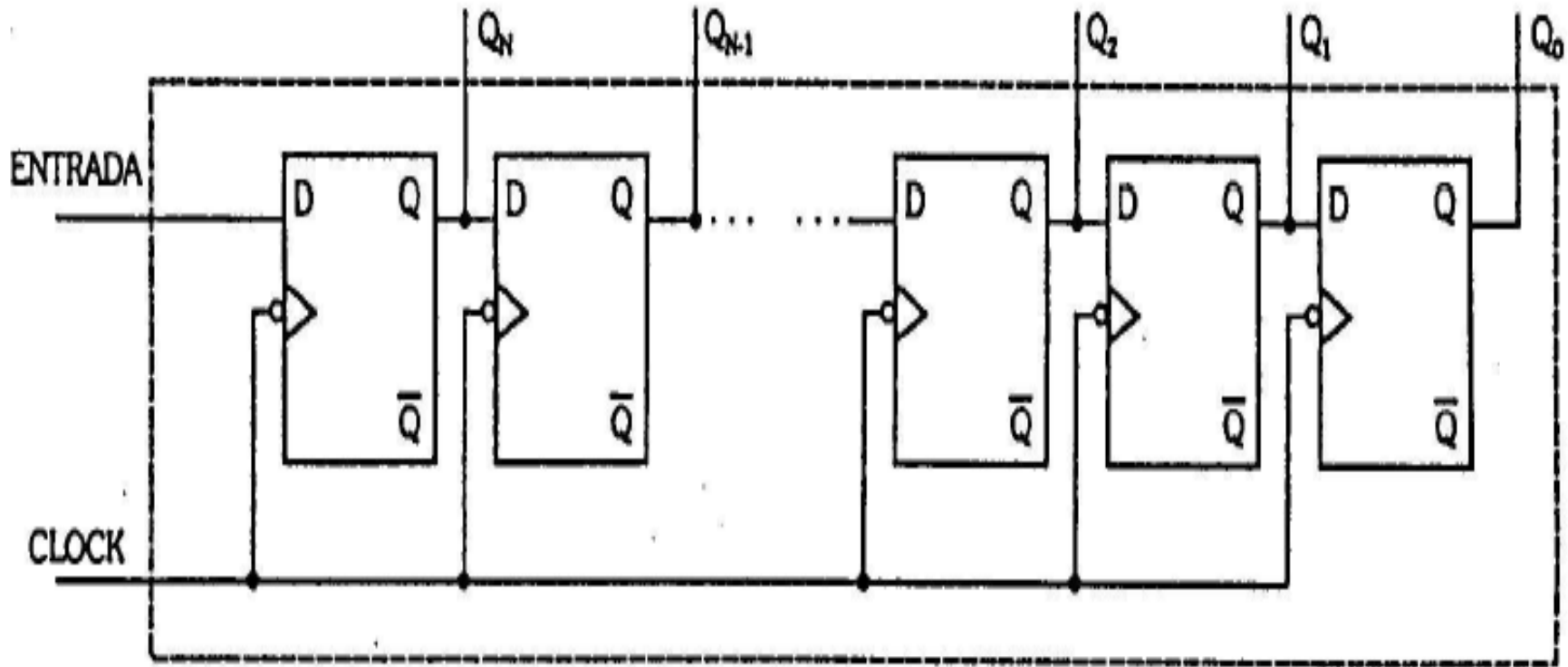
Um Flip-Flop (**FF**) pode armazenar um bit (saída **Q**) durante o período em que sua entrada “**clock**” for igual a **0**.

Para guardar uma informação de mais de um **bit**, é necessário utilizar vários **FF** conforme o tamanho do tipo do **dado** (número de bits) a ser armazenado.

Uma maneira para isso é utilizar circuitos, construídos a partir de **FF's**, denominados **Registradores de Deslocamento** (“**Shift Register**”).

Registrador de Deslocamento

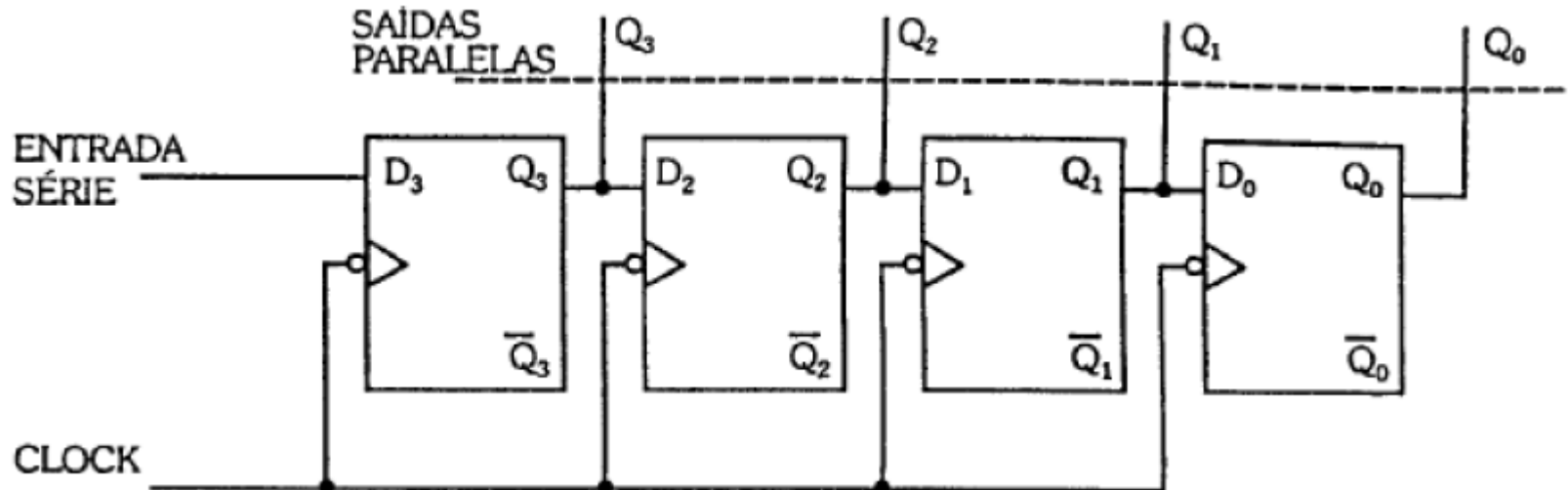
Trata-se de um, certo número de FF's tipo D ligados de tal forma que as saídas (Q) de cada bloco sejam aplicada na entradas D seguinte FF.



Conversor Série-Paralelo

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

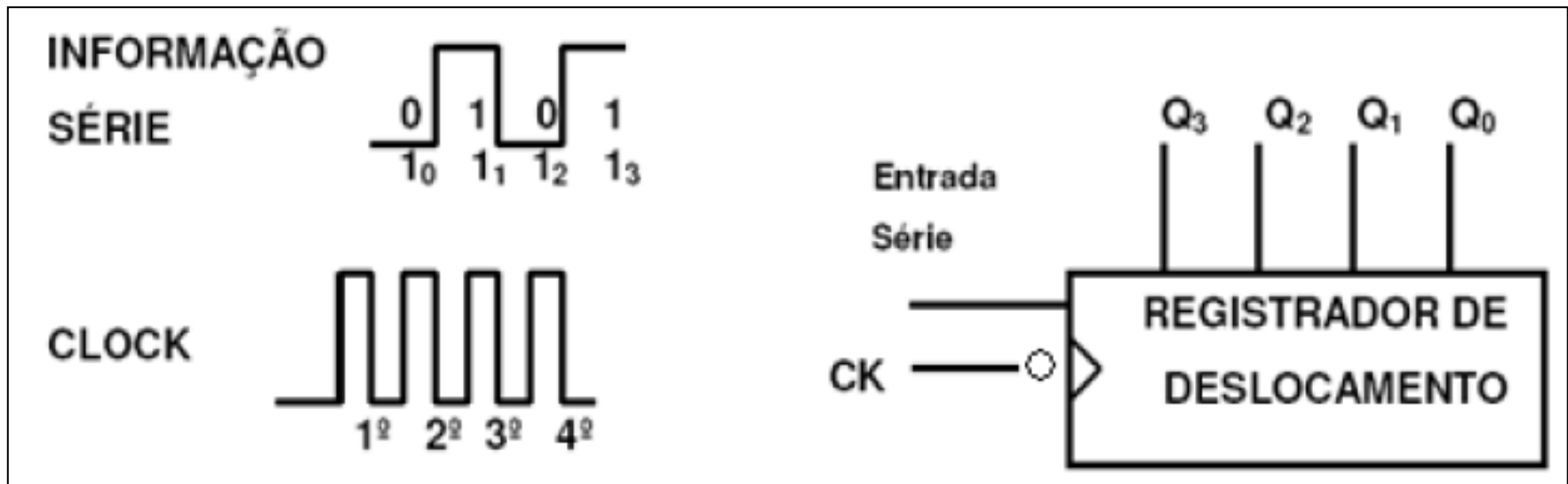
A configuração básica nessa situação, para uma informação de **4 bits**, é vista na figura:



Conversor Série-Paralelo

Como exemplo, vamos aplicar a informação série $I = 1010_2$ ($I_3 I_2 I_1 I_0$) à entrada série do registrador e analisar as saídas Q_0 , Q_1 , Q_2 e Q_3 , após os pulsos de “clock”.

Deve-se ressaltar que estes FF's atuam como **mestre-escravo** e têm sua comutação no instante da **descida** do pulso de “clock”.



Conversor Série-Paralelo

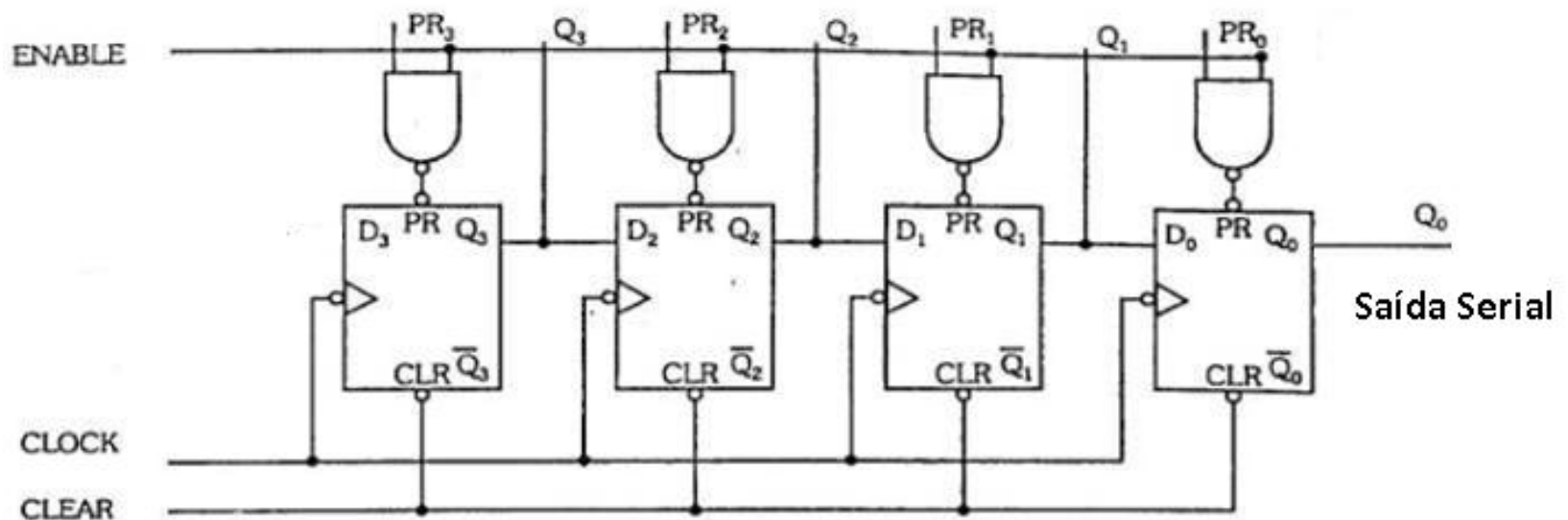
Vamos representar toda a sequencia sob a forma da tabela verdade:

Informação	Descidas de clock	Q ₃	Q ₂	Q ₁	Q ₀
I ₀ = 0	1 ^a	0	0	0	0
I ₁ = 1	2 ^a	1	0	0	0
I ₂ = 0	3 ^a	0	1	0	0
I ₃ = 1	4 ^a	1	0	1	0

Pelo motivo de deslocar a informação a cada pulso de “clock” que esse dispositivo é denominado **Registrador de Deslocamento**.

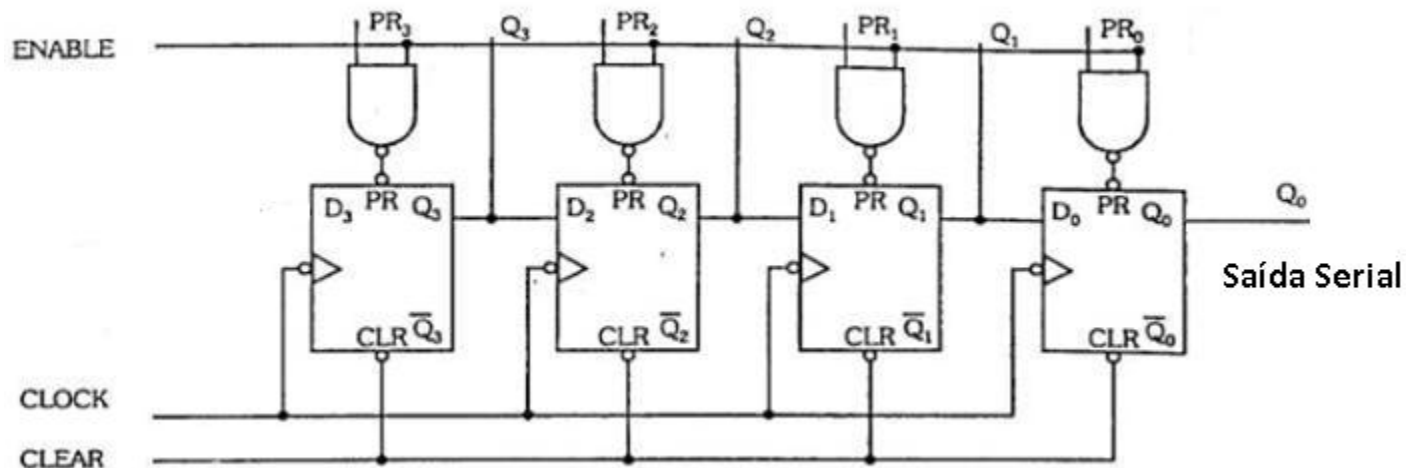
Conversor Paralelo-Série

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



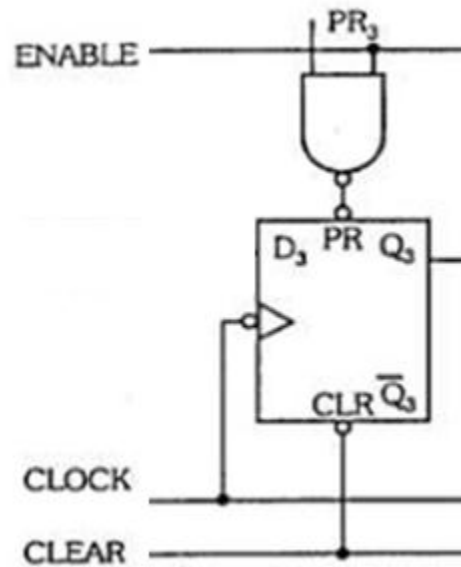
Quando a entrada “**Enable**” estiver em **0**, as entradas “**Preset**” (PR) dos **FF's** assumirão, respectivamente, níveis **1**, fazendo com que o registrador atue normalmente.

Quando a entrada “**Enable**” for igual a **1**, as entradas “**Preset**” dos **FF's** assumirão os valores complementares das entradas **PR₃**, **PR₂**, **PR₁** e **PR₀**, logo, os **FF's** irão assumir os valores que estiverem, respectivamente, em **PR₃**, **PR₂**, **PR₁** e **PR₀**.



Vamos analisar uma célula do registrador.

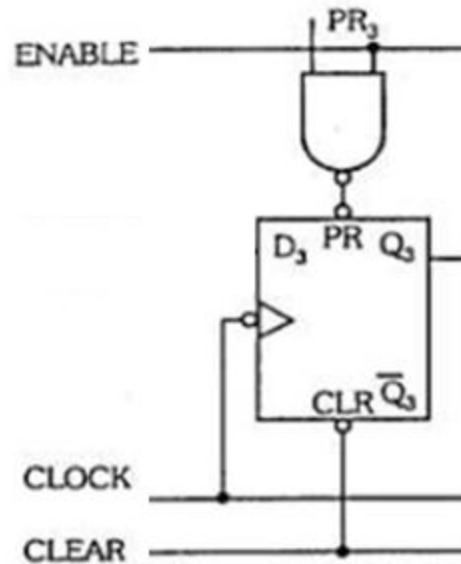
Para zerar o FF ($Q_3 = 0$), vamos inicialmente, aplicar nível 0 à entrada “Clear”.



Com “Enable” = 0, a entrada “PR” do FF irá assumir nível 1 e este irá ter um funcionamento normal como célula do registrador de deslocamento em questão, mantendo a saída no estado em que se encontra.

Conversor Paralelo-Série

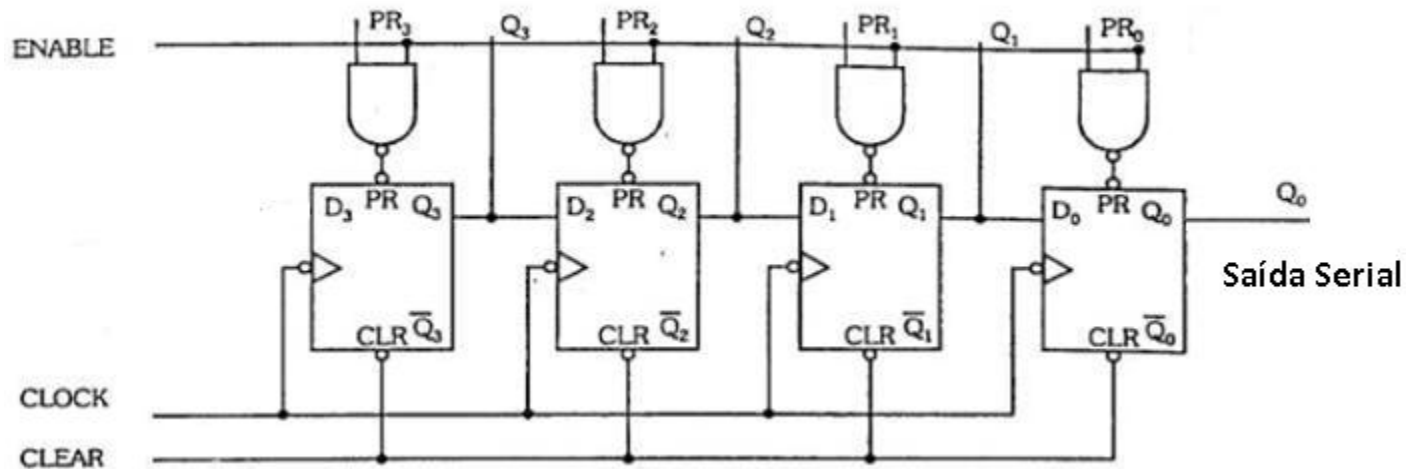
Com “Enable” = 1 e $PR_3 = 0$, a entrada PR do FF assumirá nível 1, logo, a saída Q_3 manterá o seu estado ($Q_3 = 0$).



Com “Enable” = 1 e $PR_3 = 1$, a entrada PR do FF assumirá nível 0, forçando a saída a assumir nível 1 ($Q_3 = 1$).

Conversor Paralelo-Série

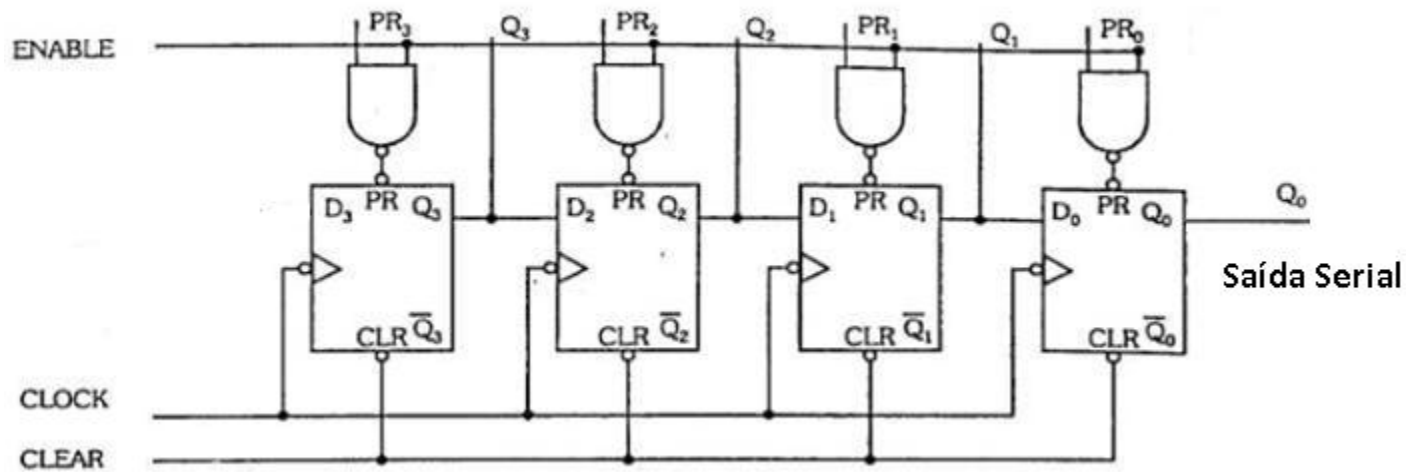
Após essa análise, concluímos que, se **zerarmos** o registrador (aplicando **0** à entrada “**Clear**”), e logo após introduzirmos a **informação paralela** (I_3 , I_2 , I_1 e I_0) pelas entradas **PR₃**, **PR₂**, **PR₁** e **PR₀**, as saídas **Q₃**, **Q₂**, **Q₁** e **Q₀** assumirão respectivamente os valores da informação.



Essa maneira de entrarmos com a informação no registrador é chamada **entrada paralela de informação**, sendo a entrada responsável pela habilitação da mesma.

Conversor Paralelo-Série

Para que o registrador de deslocamento funcione como **Conversor Paralelo-Série**, necessitamos **zerá-lo** e em seguida, introduzir a informação como já descrito, recolhendo na saída Q_0 a mesma informação de modo série.



É fácil de notar que a saída Q_0 assume primeiramente o valor I_0 e a cada **descida** do pulso de "clock", irá assumir sequencialmente os valores I_1 , I_2 e I_3 .

Os Contadores são circuitos digitais que variam os seus estados, sob o comando de um “clock”, numa sequencia predeterminada.

São utilizados principalmente para **contagens diversas, divisões de frequência, medição de frequência e tempo, geração de formas de onda e conversão de analógico para digital.**

Basicamente, estes sistemas, são divididos em duas categorias:

- Contadores Assíncronos *;
- Contadores Síncronos.

São caracterizados por seus **FF's** funcionarem de maneira **assíncrona** (sem sincronismo), não tendo entradas de “**clock**” em comum.

Neste tipo de circuito, a entrada “**clock**” se faz apenas no primeiro **FF**, sendo as outras derivadas das saídas dos blocos anteriores.

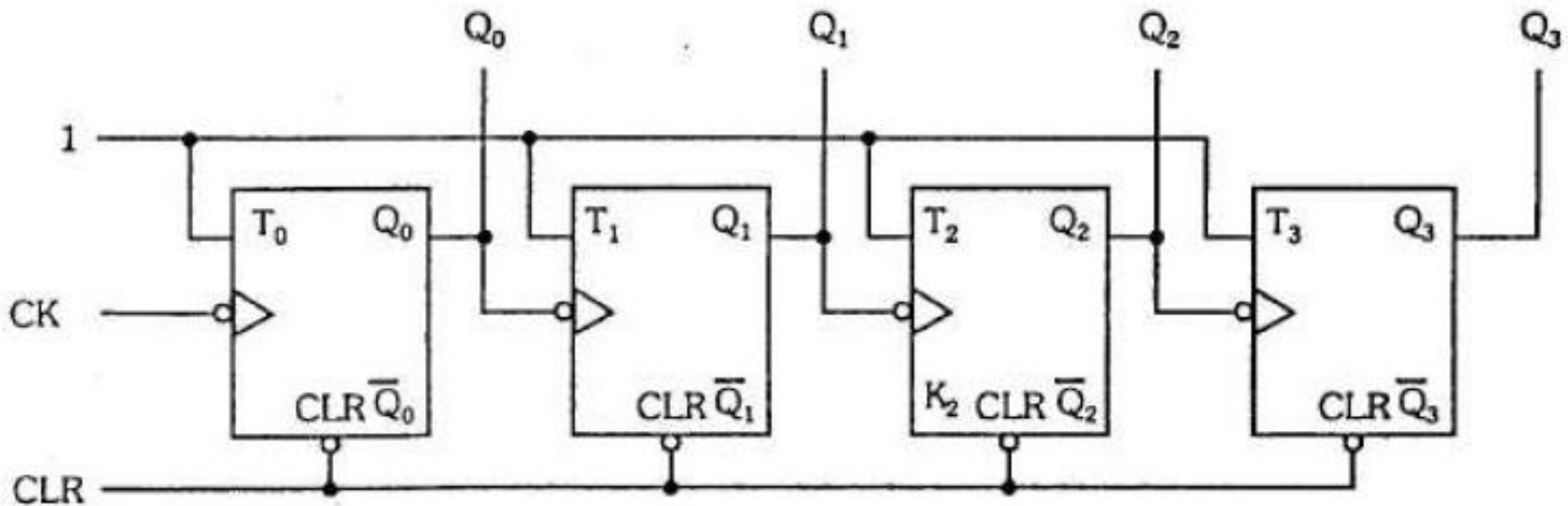
Dentre os mais comuns estão:

- **Contador de Pulsos;**
- **Contador de Década;**
- **Contador Assíncrono Crescente/Decrescente.**

Contadores de Pulsos (Assíncronos)

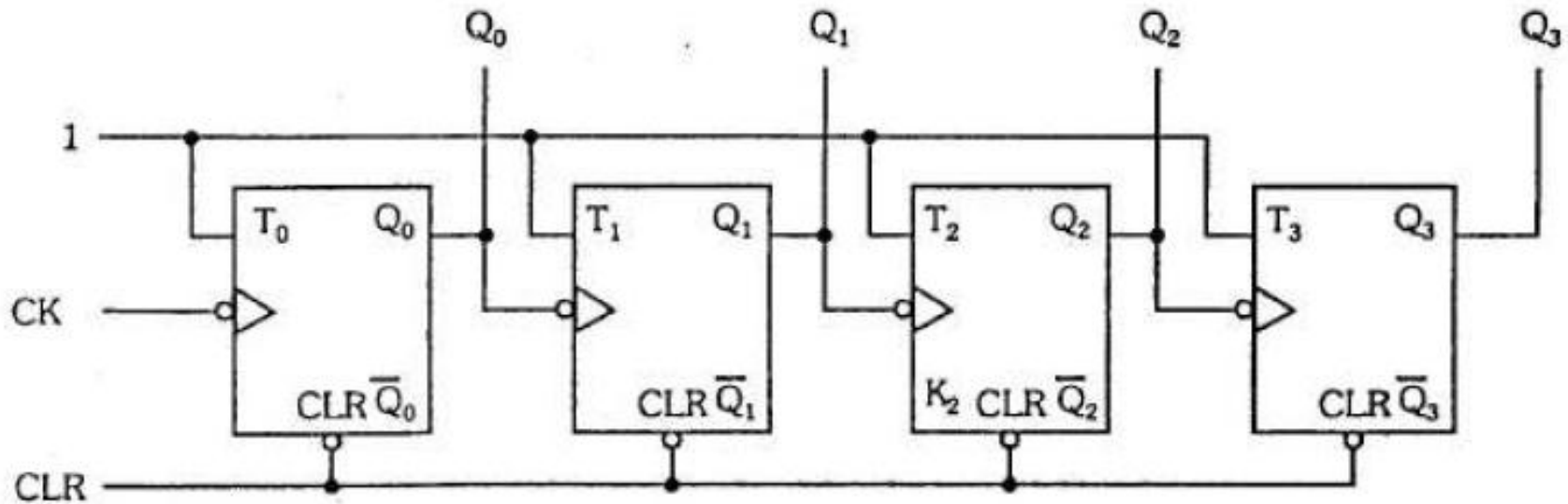
A principal característica de um contador de pulsos é apresentar nas saídas, o **sistema binário** em sequência.

Seu circuito básico apresenta um grupo de **4 FF's** do tipo **T**, os quais possuem a entrada $T = 1$, originando na saída $Q_f = \overline{Q_a}$, a cada descida de "clock".



Contadores de Pulsos (Assíncronos)

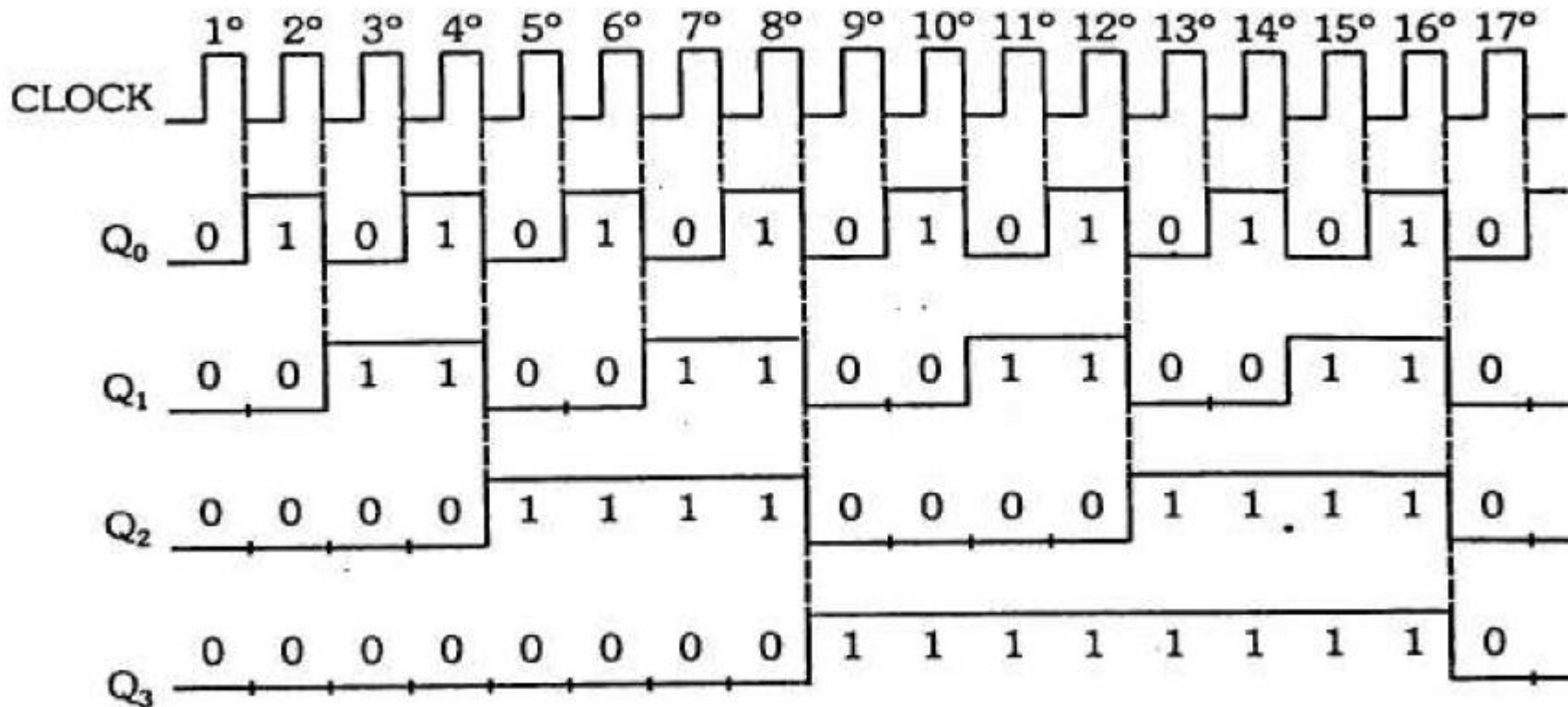
A entrada dos pulsos se faz através da entrada “clock” do 1º FF, sendo as entradas “clock” dos FF’s seguintes, conectadas às saídas Q dos respectivos antecessores conforme circuito visto na figura.



Considerando **Q₀** como **bit** menos significativo (**LSB**) e **Q₃** como mais significativo (**MSB**), temos nas saídas o sistema binário em sequência (**0000** a **1111**).

Contadores de Pulsos (Assíncronos)

A figura apresenta toda a sequência obtida graficamente, a partir da variação aplicada à entrada “clock” do sistema. Nota-se, que após a 16ª descida de “clock”, o contador irá **reiniciar a contagem**.



Contador de Década (Assíncronos)

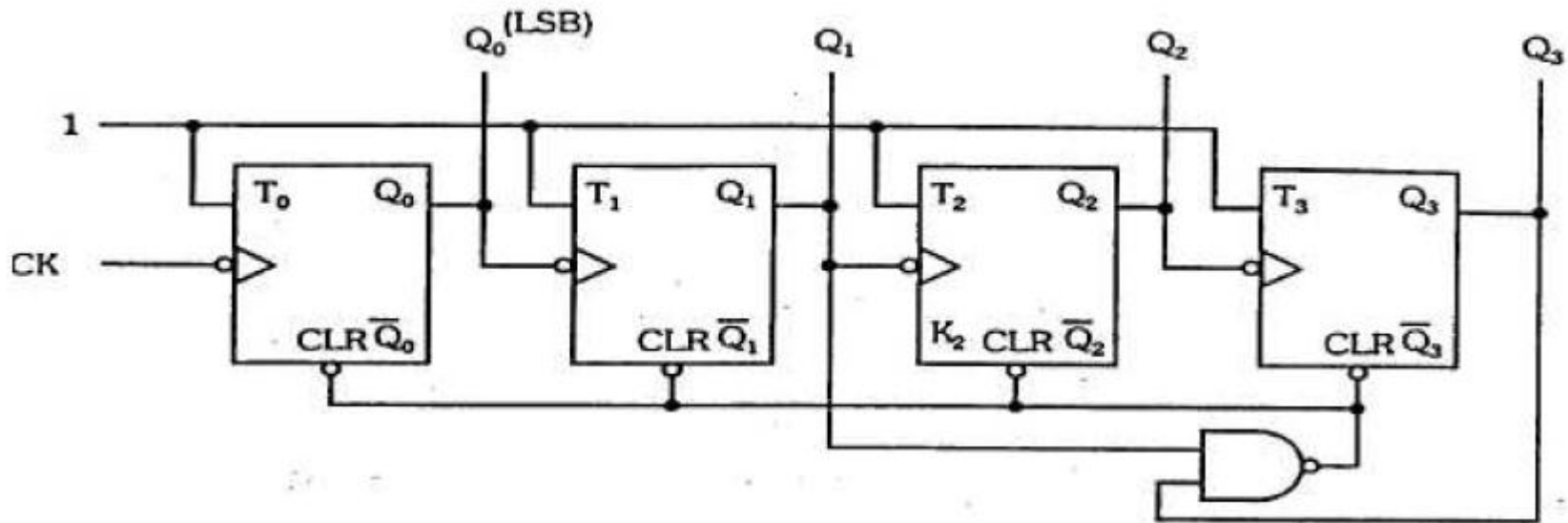
O **contador de década** é o circuito que efetua a contagem em números binários de 0000_2 a 1001_2 (**0** a **9** no sistema decimal).

Dizemos que esse contador trabalha com os números no padrão **BCD** (**Binary Coded Decimal**), onde trabalhamos com os algarismos decimais (**0** a **9**) codificados em binário.

Descidas de clock	Q ₃	Q ₂	Q ₁	Q ₀	CLR
1ª	0	0	0	0	1
2ª	0	0	0	1	1
3ª	0	0	1	0	1
4ª	0	0	1	1	1
5ª	0	1	0	0	1
6ª	0	1	0	1	1
7ª	0	1	1	0	1
8ª	0	1	1	1	1
9ª	1	0	0	0	1
10ª	1	0	0	1	1
	1	0	1	0	0

Contador de Década (Assíncronos)

Para que o contador conte somente de 0 a 9, deve-se jogar um nível 0 na entrada “Clear” assim que surgir o caso 10 (1010_2), ou seja, no 10^{o} pulso. Seu circuito pode ser visto na figura.

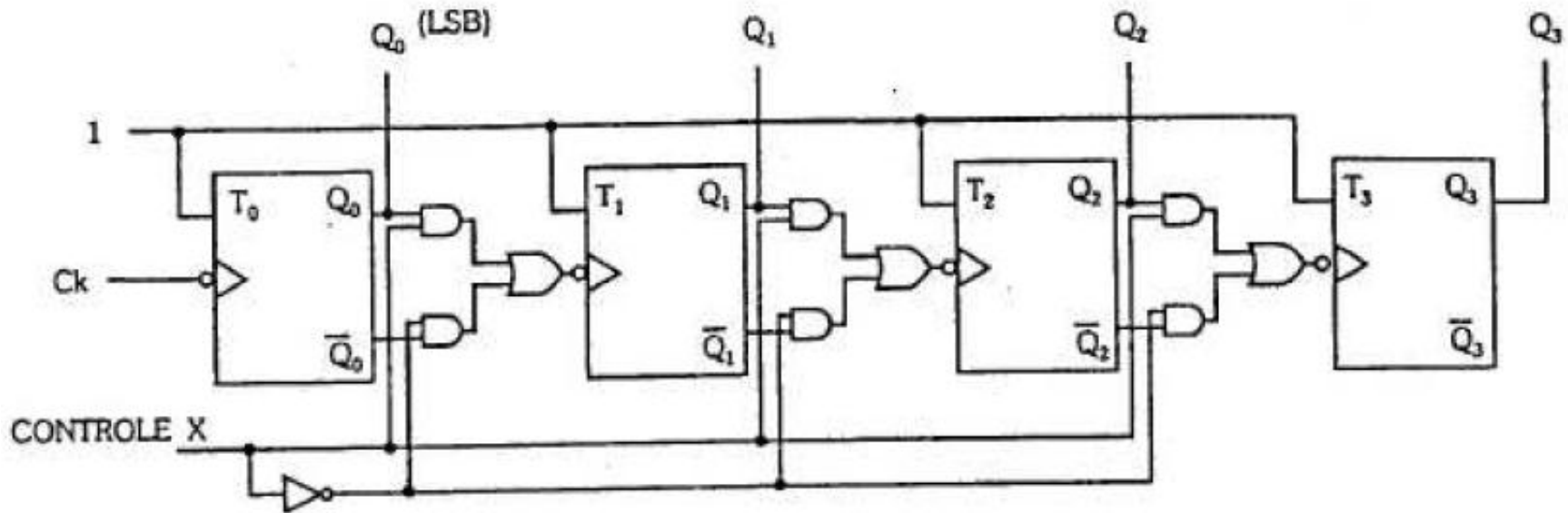


Este contador poderá ser utilizado como **divisor de frequência** por 10 para uma **onda quadrada** aplicada à entrada “clock”, pois possui 10 estados de saída.

Contador Assíncrono Crescente/Decrescente)

Podemos construir um contador que execute a contagem **crescente** ou **decrescente**.

Para isso, utilizamos uma **variável de controle** que quando assume **1**, faz o circuito executar **contagem crescente** e quando assume **0**, faz a **contagem decrescente**.



Contador Assíncrono Crescente/Decrescente)

Notamos que, no circuito, quando o controle X estiver em 1 , às saídas Q_0 , Q_1 e Q_2 estarão bloqueadas, fazendo com que entrem as saídas Q_0 , Q_1 e Q_2 nas entradas “clock” dos FF’s respectivamente. Isto fará com que o contador conte **crescentemente**.

Quando o controle X estiver em 0 , a situação inverter-se-á e, por conseguinte, o contador contará **decrescentemente**.

Notamos, ainda, que Q_0 será a saída do bit menos significativo (**LSB**).

O contador crescente/decrescente é também denominado **Up/Down Counter**, que é o termo designativo em inglês..

BLOCOS LÓGICOS BÁSICOS








PORTA	Simbologia	Tabela da Verdade	Função Lógica	Expressão															
E AND		<table border="1" style="margin: auto;"> <thead> <tr><th>A</th><th>B</th><th>S</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	A	B	S	0	0	0	0	1	0	1	0	0	1	1	1	<p>Função E: Assume 1 quando todas as variáveis forem 1 e 0 nos outros casos.</p>	$S=A \cdot B$
A	B	S																	
0	0	0																	
0	1	0																	
1	0	0																	
1	1	1																	
OU OR		<table border="1" style="margin: auto;"> <thead> <tr><th>A</th><th>B</th><th>S</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	A	B	S	0	0	0	0	1	1	1	0	1	1	1	1	<p>Função OU: Assume 0 quando todas as variáveis forem 0 e 1 nos outros casos.</p>	$S=A+B$
A	B	S																	
0	0	0																	
0	1	1																	
1	0	1																	
1	1	1																	
NÃO NOT		<table border="1" style="margin: auto;"> <thead> <tr><th>A</th><th>S</th></tr> </thead> <tbody> <tr><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td></tr> </tbody> </table>	A	S	0	1	1	0	<p>Função NÃO: Inverte a variável aplicada à sua entrada.</p>	$S=\overline{A}$									
A	S																		
0	1																		
1	0																		
NE NAND		<table border="1" style="margin: auto;"> <thead> <tr><th>A</th><th>B</th><th>S</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	A	B	S	0	0	1	0	1	1	1	0	1	1	1	0	<p>Função NE: Inverso da função E.</p>	$S=\overline{(A \cdot B)}$
A	B	S																	
0	0	1																	
0	1	1																	
1	0	1																	
1	1	0																	
NOU NOR		<table border="1" style="margin: auto;"> <thead> <tr><th>A</th><th>B</th><th>S</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	A	B	S	0	0	1	0	1	0	1	0	0	1	1	0	<p>Função NOU: Inverso da função OU.</p>	$S=\overline{(A+B)}$
A	B	S																	
0	0	1																	
0	1	0																	
1	0	0																	
1	1	0																	
OU EXCLUSIVO		<table border="1" style="margin: auto;"> <thead> <tr><th>A</th><th>B</th><th>S</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	A	B	S	0	0	0	0	1	1	1	0	1	1	1	0	<p>Função OU Exclusivo: Assume 1 quando as variáveis assumirem valores diferentes entre si.</p>	$S=A \oplus B$ $S=\overline{A} \cdot B + A \cdot \overline{B}$
A	B	S																	
0	0	0																	
0	1	1																	
1	0	1																	
1	1	0																	
COINCIDÊNCIA		<table border="1" style="margin: auto;"> <thead> <tr><th>A</th><th>B</th><th>S</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	A	B	S	0	0	1	0	1	0	1	0	0	1	1	1	<p>Função Coincidência: Assume 1 quando houver coincidência entre os valores das variáveis.</p>	$S=A \odot B$ $S=\overline{A} \cdot \overline{B} + A \cdot B$
A	B	S																	
0	0	1																	
0	1	0																	
1	0	0																	
1	1	1																	

Tabela comparativa das bases

Decimal	Binário	Octal	Hexadecimal
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Bibliografia da apresentação

- 1) <http://leandrodriguesilva.wordpress.com/temas-sugeridos/tecnologias-de-acesso/05/08/2013>
- 2) IDOETA, Ivan; CAPUANO, Francisco. Elementos de Eletrônica Digital. 37ª ed. São Paulo: Erica, 2006
- 3) Ronald J. Tocci e Neal S. Widmer; Sistemas Digitais: Princípios e Aplicações, Pearson Prentice Hall, 2003.
- 4) <http://www.poli.br/~marcilio/Sistemas%20Digitais/1o%20Exercicio/Apostila%20completa%20de%20eletronica%20digital..pdf> acesso em 05/08/2013
- 5) <http://www.aprender eletronica.com.br/img/portaslogicas.jpg> acesso em 18/08/2013
- 6) http://www.profesormolina.com./a///r/electronica/componentes/int/sist_comb/image032.jpg acesso em 18/06/2013

Bibliografia Básica Padrão

1) IDOETA, Ivan; CAPUANO, Francisco. **Elementos de Eletrônica Digital**. 37ª ed. São Paulo: Erica, 2006.

Bibliografia Básica Unidade: Faculdade Anhanguera de Rio Claro (FRC)

1) WAGNER, Flávio R.; REIS, André I.; RIBAS, Renato P.. **Fundamentos de Circuitos Digitais**. 1ª ed. Porto Alegre: Sagra Luzzatto, 2006.

2) LOURENÇO, Antônio C. de et al. **Circuitos Digitais**. 9ª ed. São Paulo: Erica, 2010.

Bibliografia Complementar: Faculdade Anhanguera de Rio Claro (FRC)

- 1) SANTOS, José Carlos Barbosa. **Introdução aos Sistemas Digitais**. 1ª ed. Porto Alegre: Bookman, 2000.
- 2) D'AMORE, Roberto. **VHDL: Descrição e Síntese de Circuitos Digitais**. 1ª ed. Rio de Janeiro: LTC, 2005.
- 3) UYEMURA, John P.. **Sistemas Digitais : Uma Abordagem Integrada**. 1ª ed. São Paulo: Thomson Learning, 2002.
- 4) TAUB, Herbert. **Circuitos Digitais e Microprocessadores**. 1ª ed. São Paulo: Macgraw-Hill, 1984.
- 5) CRUZ, Eduardo C. A. **Circuitos digitais**. 1ª ed. São Paulo: Erica, 2007, v.1.