



# CIRCUITOS LÓGICOS DIGITAIS II

## Sumário

1. Introdução .....	3
1.1 Flip-Flops .....	3
1.2.1 Flip-Flop RS Básico.....	4
1.2.2 Flip-Flop RS com Entrada Clock .....	5
1.2.3 Flip-Flop JK.....	6
1.2.4 Flip-Flop JK com Entradas Preset e Clear .....	7
1.2.5 Flip-Flop JK Mestre-Escravo .....	8
1.2.6 Flip-Flop JK Mestre-Escravo com Entrada Preset e Clear.....	10
1.2.7 Flip-Flop Tipo T .....	10
1.2.8 Flip-Flop Tipo D.....	11
1.2.9 Exercícios em Laboratório .....	12
2. Registradores de Deslocamento .....	14
2.1 Conversor Série-Paralelo.....	15
2.2 Conversor Paralelo-Série.....	16
2.2.1 Registrador de Entrada Paralela e Saída Paralela .....	17
2.2.2 Registrador de Deslocamento Utilizando como Multiplicador ou Divisor por 2.....	17
2.2.3 Exercícios.....	20
3. Contadores .....	22
3.1 Contadores Assíncronos.....	22
3.1.1 Contador de Pulsos .....	22
3.1.2 Contador de Década.....	24
3.1.3 Contador Sequência de 0 a n .....	26
3.1.4 Contadores assíncronos decrescentes .....	27
3.1.5 Contador Assíncrno Crescente/Decrescente .....	29
3.1.6 Exercícios de Contadores Assíncronos.....	30
3.2 Contadores Síncronos .....	33
3.2.1 Contadores Síncronos Gerador de Código Binário de 4 Bits.....	34
3.2.2 Contador de década .....	39
3.2.3 Gerador de uma sequência qualquer.....	41
3.2.4 Exercícios Propostos.....	47
4. Multiplexadores, Demultiplexadores e Memórias.....	48
4.1 Geração de Produtos Canônicos .....	48
4.1.1 Circuito Básico Gerador de Produtos Canônicos.....	49
4.1.2 Matriz de Simples Encadeamento.....	50
4.1.3 Matriz de Duplo Encadeamento .....	51

4.2 Multiplex .....	51
4.2.1 Projeto do Circuito de um Multiplex.....	53
4.3 Demultiplex .....	54
Referências.....	56

# 1. Introdução

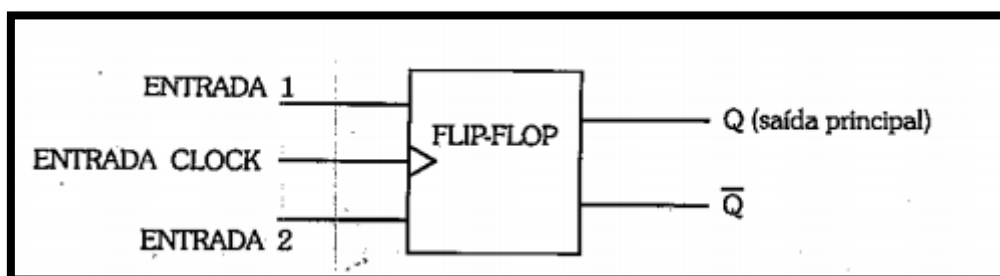
O campo da Eletrônica Digital é basicamente dividido em duas áreas: lógica combinacional e lógica sequencial.

Os circuitos combinacionais, como vimos até aqui, apresentam as saídas, única e exclusivamente, dependentes das variáveis de entrada.

Os circuitos sequências têm as saídas dependentes das variáveis de entrada e/ou de seus estados anteriores que permanecem armazenados, sendo, geralmente, sistemas pulsados, ou seja, operam sob o comando de uma sequência de pulsos denominada **clock**,

## 1.1 Flip-Flops

De forma geral, podemos representar o flip-flop como um bloco onde temos 2 saídas: Q e Q', entradas para as variáveis e uma entrada de controle (*clock*). A saída Q será a principal do bloco.



Este dispositivo possui basicamente dois estados de saída. Para o Flip-Flop assumir um destes estados é necessário que haja uma combinação das variáveis e do pulso de controle (*clock*). Após este pulso, o flip-flop permanecerá neste estado a chegada de um novo pulso de clock e, então, de acordo com as variáveis de entrada, mudará ou não de estado.

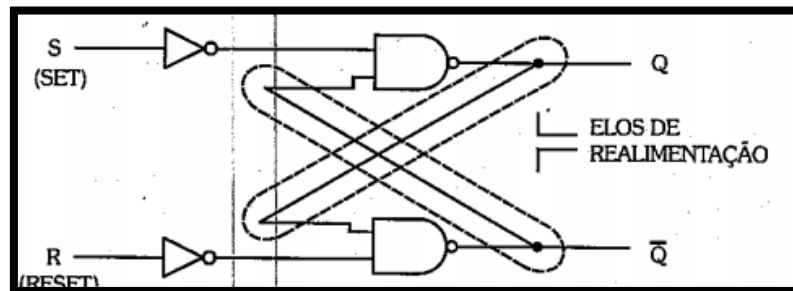
Os dois estados possíveis são:

1)  $Q = 0 \rightarrow Q' = 1$

2)  $Q = 1 \rightarrow Q' = 0$

### 1.2.1 Flip-Flop RS Básico

Primeiramente, vamos analisar o flip-flop RS básico construído a partir de portas NAND e inversoras (NOT).



Notamos que estes elos de realimentação fazem com que as saídas seja injetadas juntamente com as variáveis de entrada, ficando claro, então, que os estados que as saídas irão assumir dependerão de ambas.

Para analisarmos o comportamento do circuito, vamos construir a tabela da verdade, levando em consideração as 2 variáveis de entrada (S e R) e a saída Q anterior (Qa).

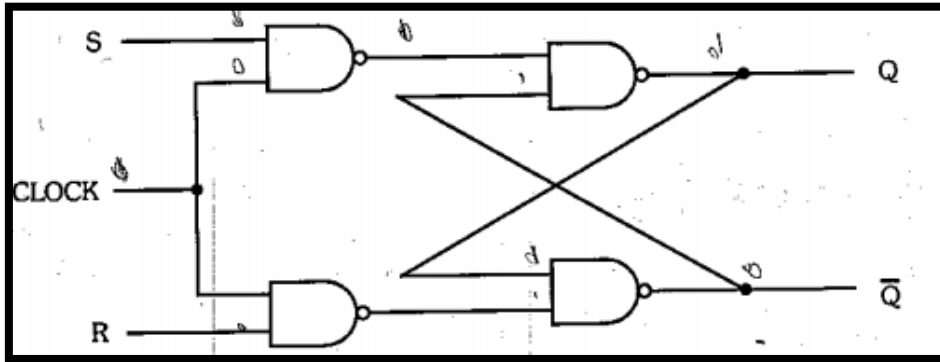
R	S	Q	$\bar{Q}$
0	0	Q <sub>a</sub> mem	Q <sub>a</sub> 'mem
0	1	1	0
1	0	0	1
1	1	ERRO	ERRO

A entrada S é denominada **SET**, pois quando acionada (nível 1), passa a saída para 1, e a entrada R é denominada **RESET**, pois quando acionada passa a saída para 0 (recompões ou zera o flip-flop). Estes termos são muito usuais na área da eletrônica digital, sendo provenientes do idioma inglês.

Este circuito irá mudar de estado apenas no instante em que mudam as variáveis de entrada. Veremos em seguida, como é o circuito de um flip-flop RS que tem sua mudança de estado controlada pela entrada do clock.

### 1.2.2 Flip-Flop RS com Entrada Clock

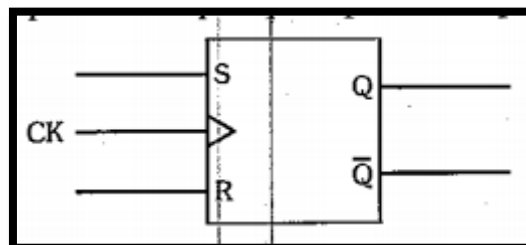
Para que um flip-flop RS básico seja controlado por uma sequência de pulsos de clock, basta trocarmos os 2 inversores por portas NAND, e às outras entradas destas portas, injetarmos o clock.



Quando a entrada do clock assumir valor 1, o circuito irá comporta-se como um flip-flop RS básico, pois as portas NAND de entrada funcionarão como os inversores do circuito anteriormente visto.

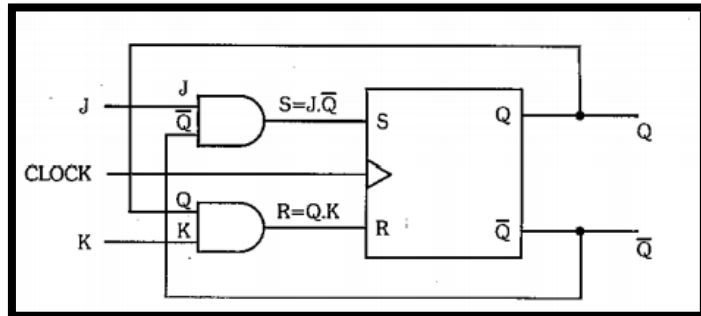
Clock	Q
0	Qa
1	RS Básico

De maneira geral, podemos concluir que o circuito irá funcionar quando a entrada, clock assumir valor 1 e manterá travada esta saída quando a entrada clock passar para 0. O flip-flop RS pode ser representado pelo bloco abaixo:



### 1.2.3 Flip-Flop JK

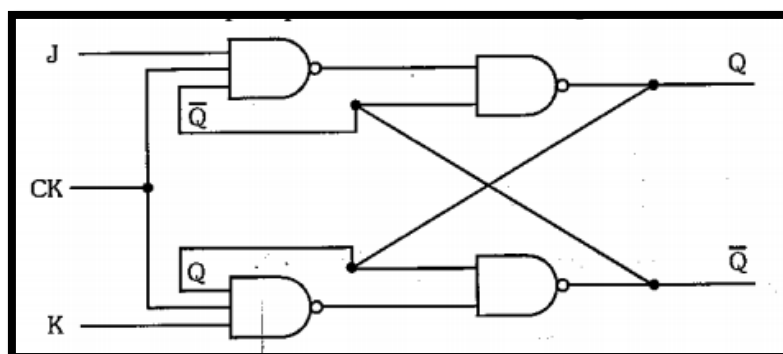
O flip-flop JK nada mais é que um flip-flop RS realimentado pelas sua saídas, conforme mostrado abaixo:



A tabela verdade resultante será:

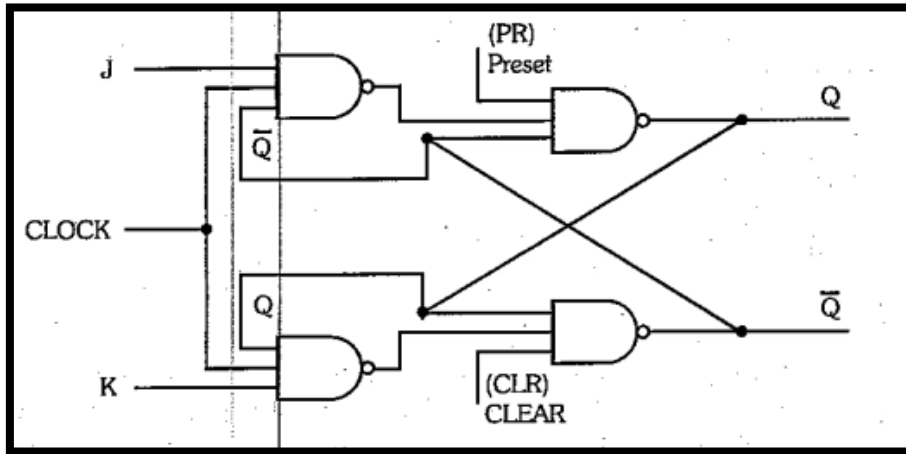
J	K	Q
0	0	Qa
0	1	0
1	0	1
1	1	Qa'

No caso  $J = 1$  e  $K = 1$ , para obter-se  $Q = Qa'$  é necessário que a entrada do clock volte a situação 0 em um tempo conveniente após a aplicação das entradas, pois, caso contrário, a saída entrará em constante mudança (oscilação), provocando novamente uma indeterminação. Este tempo deve levar em conta o tempo de atraso de propagação de cada porta lógica. Outra possibilidade para melhor desempenho, é a de inserir blocos de atraso em série com as linhas de realimentação no circuito e comutar a entrada do clock da mesma forma, ou seja, para se obter  $Q = Qa'$ .



### 1.2.4 Flip-Flop JK com Entradas Preset e Clear

O flip-flop JK poderá assumir valores  $Q = 1$  ou  $Q = 0$  mediante a utilização das entradas Preset(PR) e Clear(CLR). Estas entradas são inseridas no circuito.



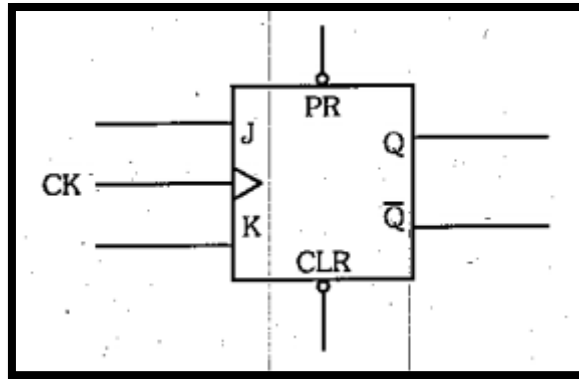
Analisando este circuito, podemos notar que com a entrada clock igual a 0 e conseqüentemente bloqueio de passagem das entradas J e K, podemos impor ao circuito saída Q igual a 1 através da aplicação à entrada Preset de nível 0. De forma análoga, podemos fazer  $Q = 0$  mediante a aplicação à entrada Clear de nível 0. Podemos notar também que com essas entradas permanecendo iguais a 1, o circuito funciona normalmente como sendo um flip-flop JK.

As entradas Preset e Clear não podem assumir valor 0, simultaneamente, pois acarretaria à saída uma situação não permitida. A entrada Clear é também denominada Reset, termo este, da mesma forma que os outros, derivado do inglês.

Tabela resumo da atuação dos Preset e Clear.

CLR	PR	Q
0	0	Não permitido
0	1	0
1	0	1
1	1	Funcionamento Normal

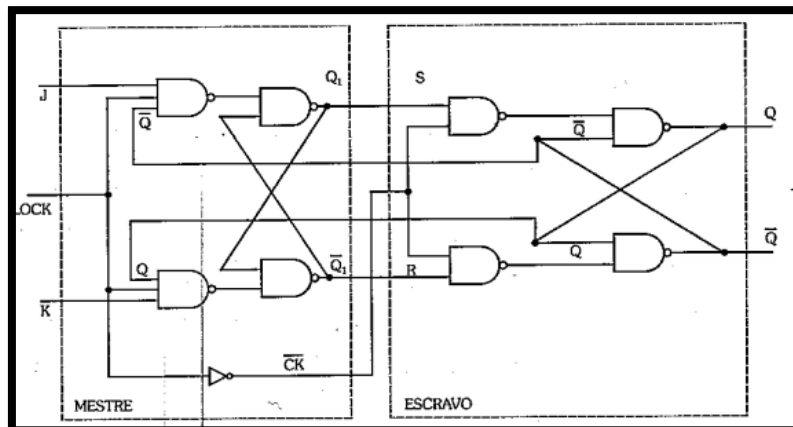




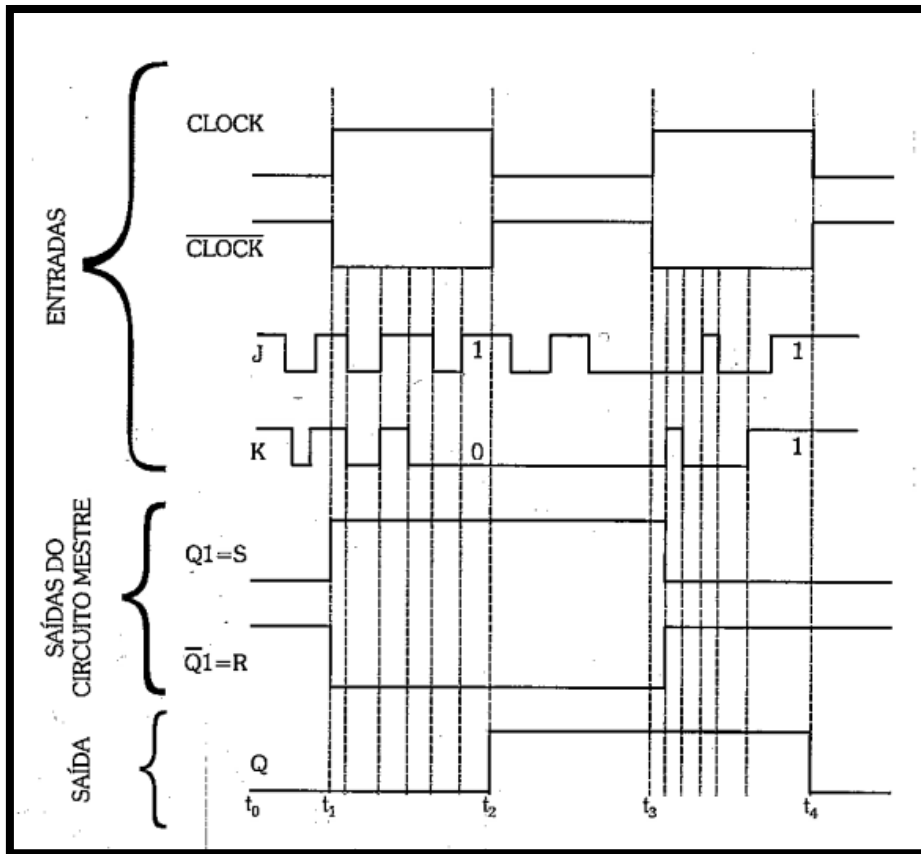
### 1.2.5 Flip-Flop JK Mestre-Escravo

O flip-flop JK apresenta uma característica indesejável. Quando o clock for igual a 1, teremos o circuito funcionando como sendo um circuito combinacional, pois haverá passagem das entradas J, K e também da realimentação. Nessa situação, se houver uma mudança nas entradas J e K, o circuito apresentará uma nova saída, podendo alterar seu estado tantas vezes quantas alterarem os estados das entradas J e K.

Para resolver esse problema, foi criado o flip-flop JK Mestre-Escravo.



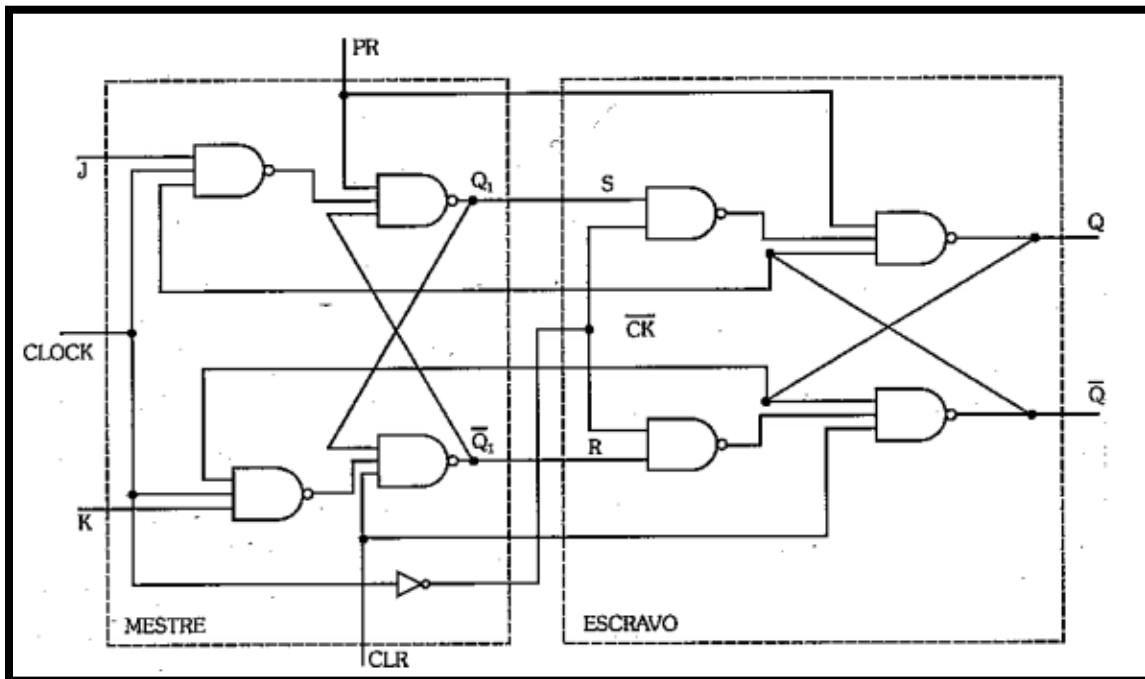
Primeiramente, devemos notar que quando o clock for igual a 1, haverá a passagem das entradas J e K (circuito mestre), porém não haverá passagem das saídas Q1 e Q1' (entradas S e R do circuito escravo), pois enquanto o clock do circuito mestre for igual a 1, no circuito escravo será 0, as saídas Q1 e Q1' ficarão bloqueadas no ultimo estado assumido e entrarão em R e S desbloqueadas, mudando o estado do circuito escravo e conseqüentemente das saídas Q e Q'. O problema da variação das entradas J e K foi resolvido, pois o circuito só reconhecerá as entradas J e K no instante da passagem do clock para 0.



A tabela verdade resultante será:

J	K	Q
0	0	Qa
0	1	0
1	0	1
1	1	Qa'

### 1.2.6 Flip-Flop JK Mestre-Escravo com Entrada Preset e Clear



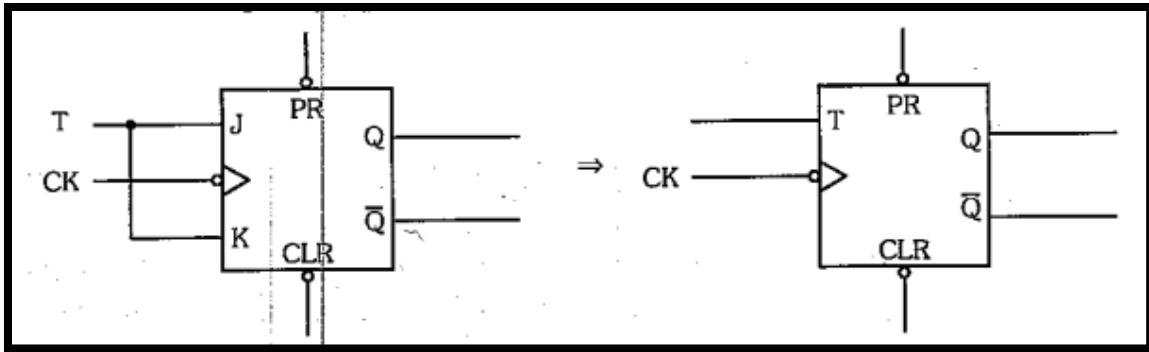
O controle de Preset, quando assumir valor 0, fará com que a saída do circuito Q assumira o valor 1. O mesmo ocorre com o controle de Clear, fazendo com que a saída assumira valor 0. Notamos que ambos, por estarem ligados simultaneamente aos circuitos Mestre-Escravo, atuam independentemente da entrada clock.

Tabela resumo da atuação dos Preset e Clear.

CLR	PR	Q
0	0	Não permitido
0	1	0
1	0	1
1	1	Funcionamento Normal

### 1.2.7 Flip-Flop Tipo T

Este FF é obtido a partir de um JK Mestre-Escravo com as entradas J e K curto-circuitadas (uma ligada a outra), logo quando J assumir valor 1, K também assumirá valor 1, e quando J assumir o valor 0, K também assumirá o valor 0. Obviamente, no caso desta ligação não irão ocorrer nunca entradas como:  $J = 0$  e  $K = 1$ ;  $J = 1$  e  $K = 0$ .



A tabela verdade completa será então:

J	K	T	Q
0	0	0	Qa
0	1	Não existe	/
1	0	Não existe	/
1	1	1	Qa'

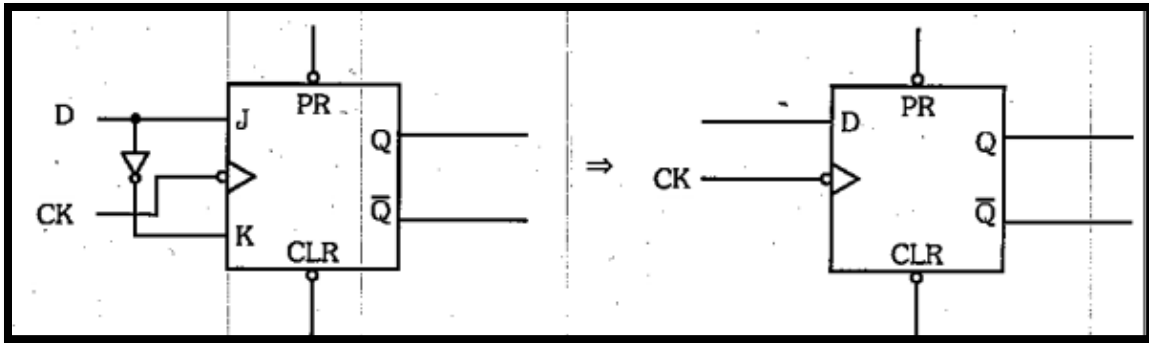
Tabela verdade resumida:

T	Qf
0	Qa
1	Qa'

Devido ao fato de o flip-flop tipo T, com a entrada T igual a 1, complementar a saída Qa' a cada descida de clock, este será utilizado como célula principal dos **contadores assíncronos**. O FFT não é encontrado na série de circuitos integrados comerciais, sendo na prática montando a partir de um JK Mestre-Escravo.

### 1.2.8 Flip-Flop Tipo D

Este FF é obtido a partir de um JK Mestre-Escravo com a entrada K invertida (por inverter) em relação a J. Logo, neste flip-flop, teremos as entradas da seguinte forma: J = 0 e K = 1; J = 1 e K = 0 apenas.



A tabela verdade completa será então:

J	K	D	Q
0	0	Não existe	/
0	1	0	0
1	0	1	1
1	1	Não existe	/

Tabela verdade resumida:

D	Qf
0	0
1	1

### 1.2.9 Exercícios em Laboratório

- Acesse o site <https://simulador.io>
- Create new Circuit

Exercício 1 - Simule os Flip Flops RS, JK e D utilizando Leds. Para cada posição que o

Led Acender tire um print.

Para realizar o exercício você irá precisar dos seguintes componentes.

- Led

□ FFRS, FFJK, FFD e CLK

□ Dois Switchs.

Exercício 2 – Adicione um registrador de ondas (Shifting Register) no sistema e defina o clock para apenas 10 pulsos.

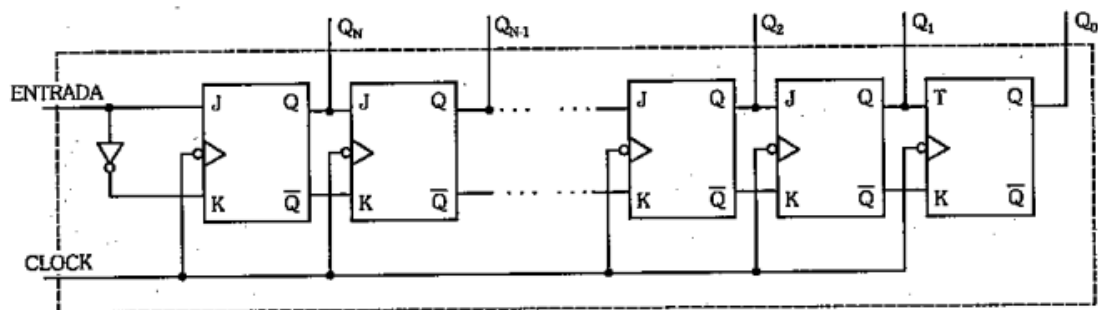
Exercício 3 – Simule o funcionamento de um FF Tipo T.

Exercício 4 – Utilizando as portas lógicas faça o funcionamento de um FFJK e adicione um registrador nas saídas após o led.

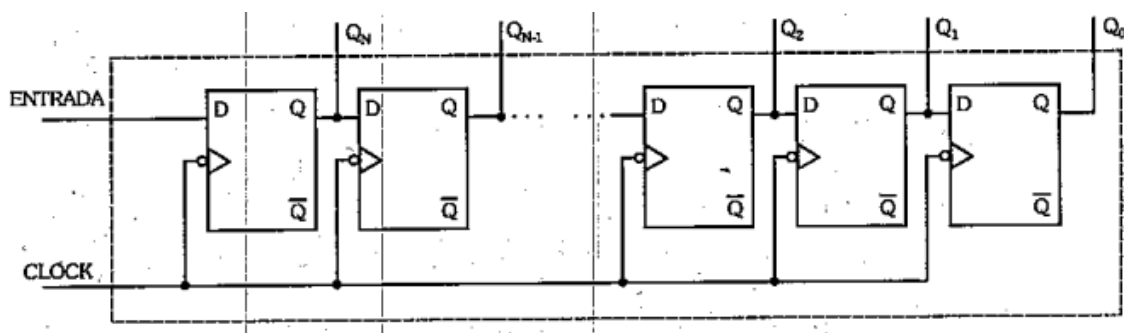
Exercício 5 – Utilizando as portas lógicas faça o funcionamento de um FFRS e adicione um registrador nas saídas após o led.

## 2. Registradores de Deslocamento

Como vimos, o flip-flop pode armazenar durante o período em que sua entrada clock for igual a 0, um bit apenas (saída Q). Porém, se necessitarmos guardar uma informação de mais de um bit, o flip-flop irá tornar-se insuficiente. Para isso utilizamos de um sistema denominando **Registrador de Deslocamento (Shift Register)**. Trate-se de um certo número de flip-flops tipo JK mestre-escravo ligado de tal forma que as saídas de cada block seja, aplicadas nas entradas ligadas a um flip-flop tipo D. A figura abaixo representa um Registrador de Deslocamento generalizado para N+1 bits.



Pelo fato de os flip-flops envolvidos atuarem como os do tipo D, este circuito, para facilitar, pode ser construído apenas com flip-flops do tipo D. A figura abaixo mostra a mesma estrutura geral, porém, composta apenas com flip-flops D.



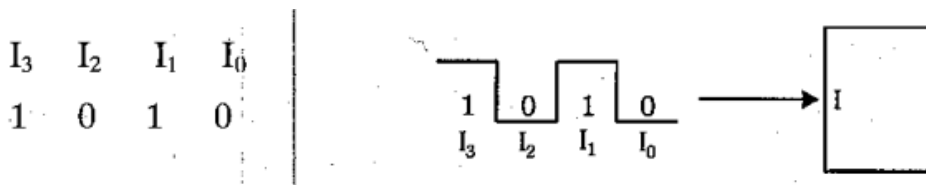
## 2.1 Conversor Série-Paralelo

Chamamos de informação paralela a uma informação na qual todos os bits se apresentam simultaneamente. Uma informação paralela necessita tantos fios quantos forem os bits contidos nela, além, logicamente, do fio referencia do sistema (terra). Conforme ilustrado na figura abaixo:



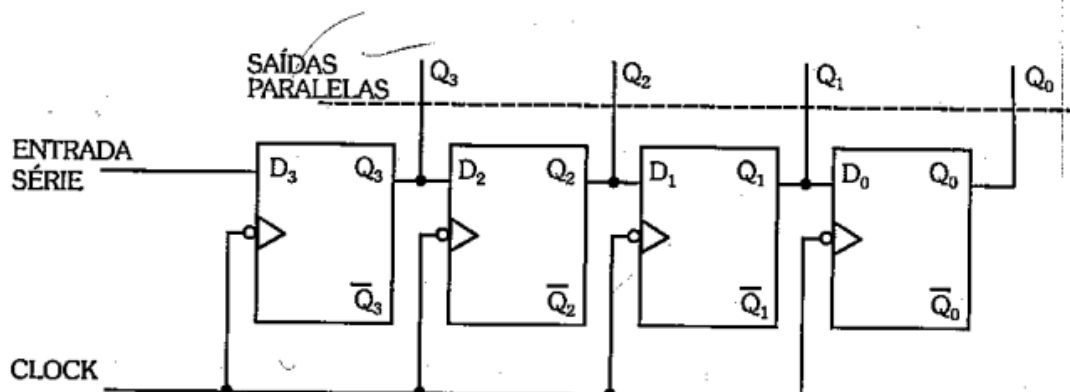
Notamos que esta informação necessita de 4 fios para ser transmitida ou inserida no bloco.

Informações sérei é aquela que utiliza apenas 1 fio, sendo que os bits de informação vê sequencialmente, um após o outro. Como exemplo, vamos utilizar a mesma informação, porém em série:



Notamos que esta informação necessita de 1 fio para ser transmitida ou inserida no bloco.

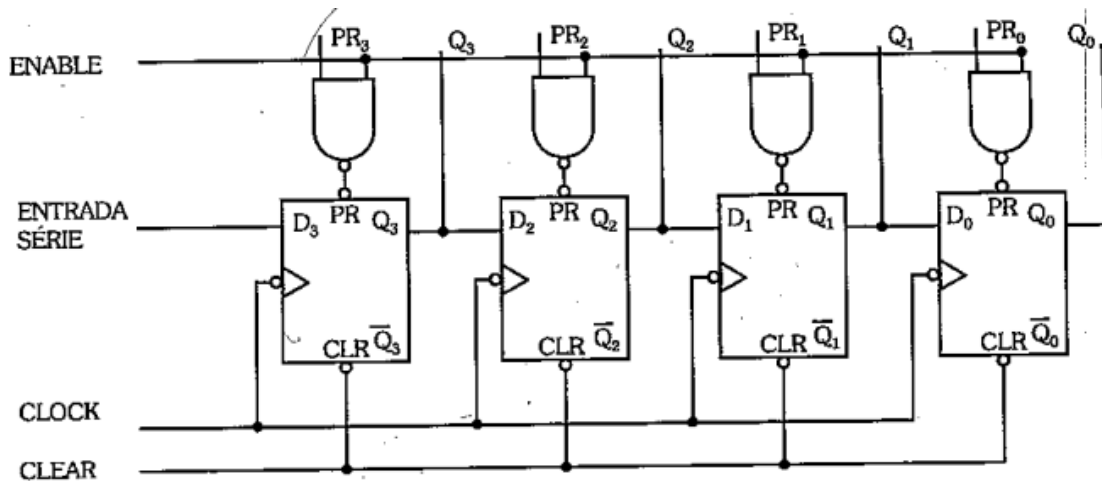
O Registrador de Deslocamento pode ser usado para converter uma informação série em paralela, ou seja, funcionar como um converser série paralelo.



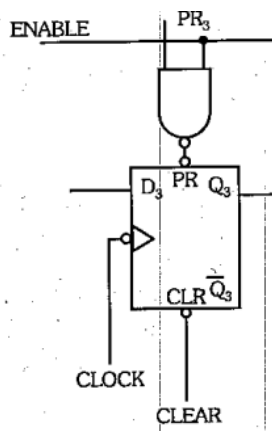


## 2.2 Conversor Paralelo-Série

Para entrarmos com uma informação paralela, necessitamos de um registrador que apresente as entradas *Preset* e *Clear*, pois é através destas que fazemos com que o Registrador armazene a informação paralela. O registrador com estas entradas é visto na figura abaixo:



Primeiramente, vamos estudar o funcionamento da entrada *ENABLE*. Quando a entrada *enable* estiver em 0, as entradas *preset*(PR) dos flip-flops assumirão, respectivamente, níveis 1, fazendo com que o registrador atue normalmente. Quando a entrada *enable* for igual a 1, as entradas *preset* dos flip-flops assumirão os valores complementares das entradas PR3, PR2, PR1 e PR0, logo, os flip-flops irão assumir os valores que estiverem, respectivamente, em PR3, PR2, PR1 e PR0. Para entendermos melhor, vamos analisar uma célula do registrador.



Para zerar (*clear*) o flip-flop ( $Q_3 = 0$ ), vamos inicialmente, aplicar nível 0 à entrada *clear*.

Com  $enable = 0$ , a entrada PR do flip-flop irá assumir nível 1 e este irá ter um funcionamento normal como célula do registrador de deslocamento, mantendo a saída no estado em que se encontra.

Com  $enable = 1$  e  $PR_3 = 0$ , a entrada PR do flip-flop assumirá nível 1, logo a saída  $Q_3$  manterá o seu estado ( $Q_3 = 0$ ). Com  $enable = 1$  e  $PR_3 = 1$ , a entrada do PR do flip-flop assumirá nível 0, forçando a saída assumir nível 1 ( $Q_3 = 1$ ).

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  $PR_3, PR_2, PR_1$  e  $PR_0$ , as saídas  $Q_3, Q_2$ , e  $Q_0$  assumirão respectivamente os valores da informação. Essa maneira de entrarmos com a informação no registrador é chamada entrada paralela de informação, sendo a entrada *enable* responsável pela habilitação da mesma.

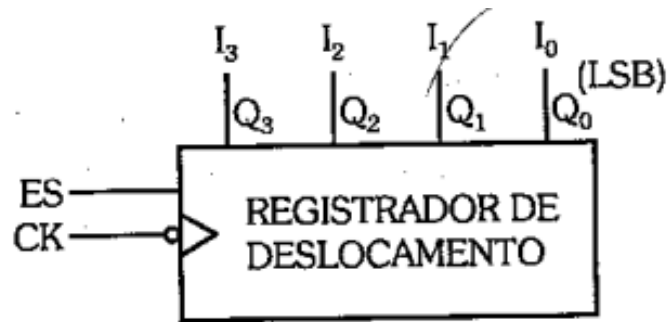
Para que o registrador de deslocamento funcione como **Conversor Paralelo-Série**, se necessitarmos zerá-lo e em seguida, introduzir a informação como já descrito, recolhendo na saída  $Q_0$  a mesma informação de modo série. É fácil de notar que a saída  $Q_0$  assume primeiramente o valor  $I_0$  e a cada descida do pulso do *clock*, irá assumir sequencialmente os valores  $I_1, I_2$  e  $I_3$ .

### 2.2.1 Registrador de Entrada Paralela e Saída Paralela

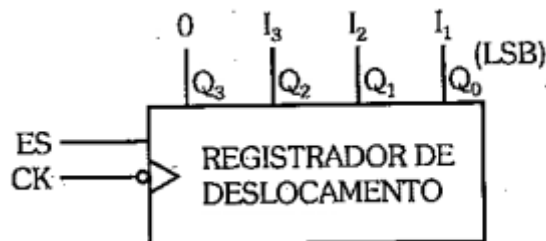
A entrada paralela, como já visto, se faz através dos terminais *preset* e *clear*. Se inibirmos a entrada de *clock*, a informação contida no registrador pode ser acessada pelos terminais de saída  $Q_3, Q_2, Q_1$  e  $Q_0$ .

### 2.2.2 Registrador de Deslocamento Utilizando como Multiplicador ou Divisor por 2

Como vimos, se entrarmos com uma informação de deslocamento, teremos as seguintes situações de saídas:



Se essa informação for considerada um número binário e deslocarmos registrador uma casa à direita, entrando com 0 na entrada série, teremos a seguinte situação:



Podemos notar que essa operação, em binário, significa dividirmos um número por 2. Para exemplificar, vamos analisar a informação:

$$I = 1010 \text{ (10)}$$

$$\text{Registrador} \rightarrow Q_3 = 1, Q_2 = 0, Q_1 = 1 \text{ e } Q_0 = 0$$

Se fizermos um deslocamento para a direita, teremos na saída a seguinte situação:

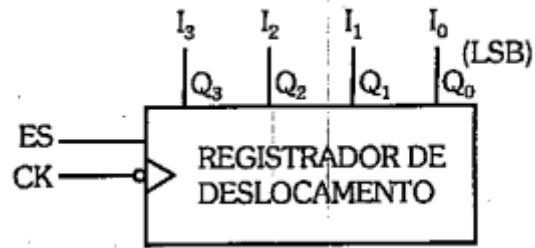
$$Q_3 = 0, Q_2 = 0, Q_1 = 1 \text{ e } Q_0 = 0$$

Notamos que a informação recolhida na saída será:

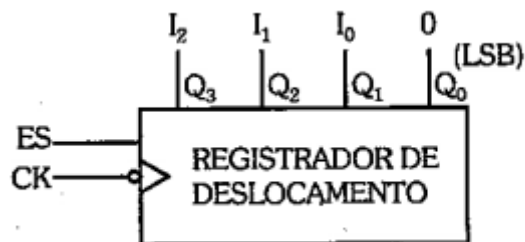
$$I = 0101 \text{ (5)}$$

Podemos verificar que o número foi dividido por 2. Esta operação de deslocarmos a informação para a direita é também conhecida por *Shift Right*, termo designativo em inglês.

Podemos estruturar um registrador que permita o deslocamento para a esquerda. Se entrarmos com uma informação no registrador, teremos:



Se aplicarmos um deslocamento à esquerda, levando a saída Q0 para 0, teremos a seguinte situação:



Podemos notar que essa operação significa multiplicar um número binário por 2. Para exemplificar, utilizaremos a informação:

$I = 0001$

Registrador  $\rightarrow Q_3 = 0, Q_2 = 0, Q_1 = 0$  e  $Q_0 = 1$ .

Se fizermos um deslocamento para a esquerda, teremos na saída, a seguinte situação:

$Q_3 = 0, Q_2 = 0, Q_1 = 1$  e  $Q_0 = 0$

Notamos que a informação recolhida na saída será:

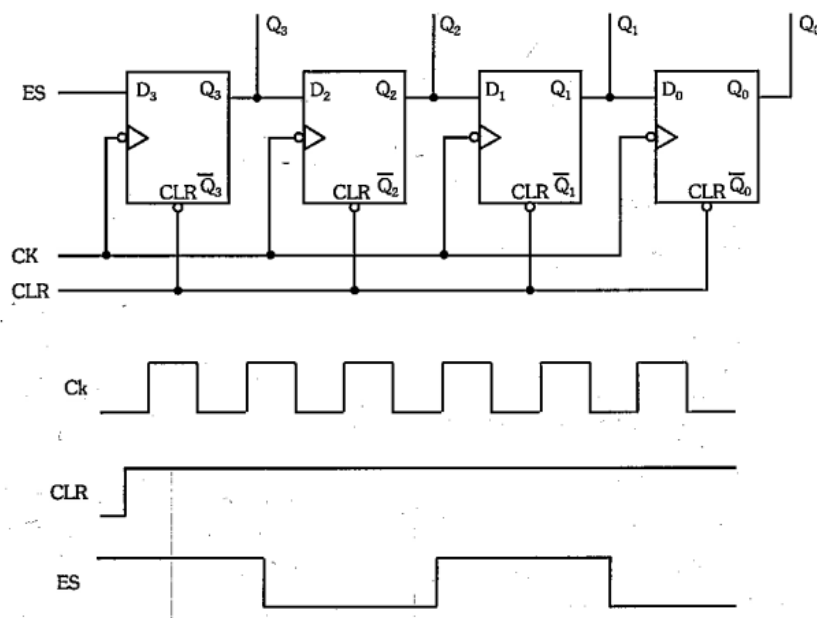
$I = 0010$  (2)

Podemos facilmente verificar que o número foi multiplicado por 2.

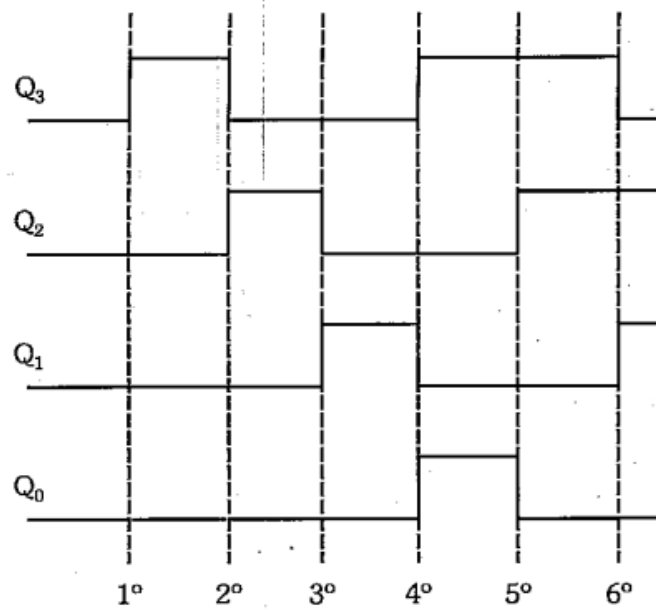
O deslocamento à esquerda é também conhecido como *Shift-Left*, termo designativo em inglês.

### 2.2.3 Exercícios

1- A partir dos sinais aplicados às entradas, esboce as formas de onda das saídas para o Registrador de Deslocamento de 4 bits, visto na figura abaixo:

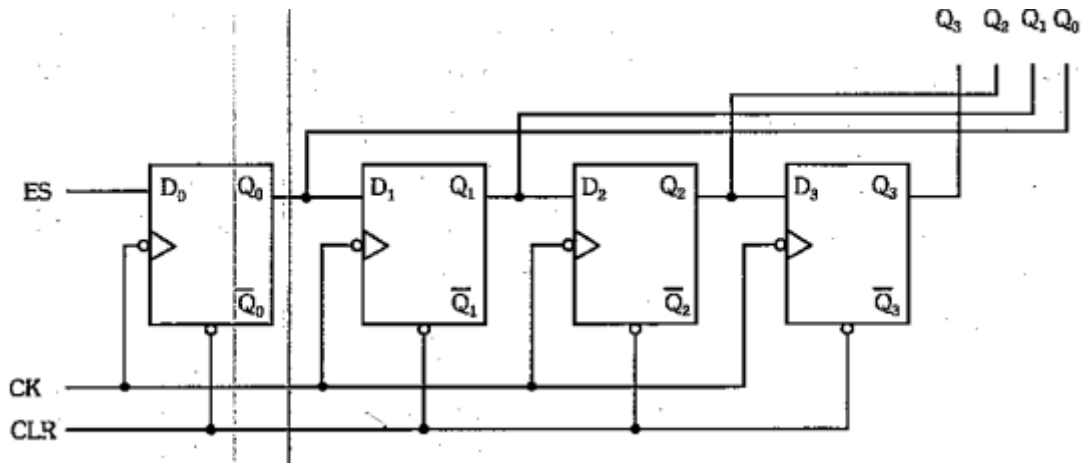


Resposta:



2- Esquematize o circuito de um Registrador de Deslocamento de 4 bits, para efetuar a operação de deslocamento à esquerda.

Resposta:



### 3. Contadores

Contadores são circuitos digitais que varia os seus estados, sob o comando de um **clock**, de acordo com uma sequência pré determinada. São utilizados principalmente para contagens diversas, divisão 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** e **Contadores Síncronos**.

#### 3.1 Contadores Assíncronos

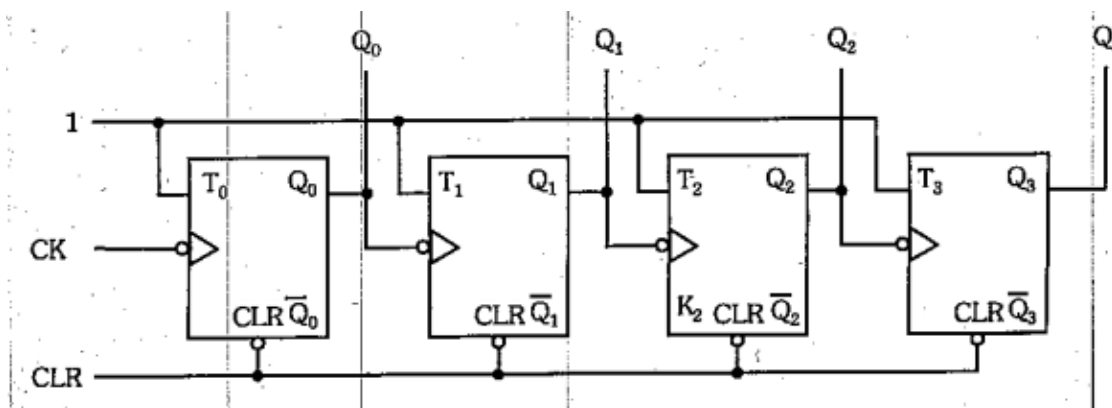
São caracterizados por seus flip-flops funcionarem de maneira assíncrono (sem sincronismo), não tendo entradas clock em comum. Neste tipo de circuito, a entrada do clock se faz apenas no primeiro flip-flop, sendo as outras derivadas das saídas dos blocos anteriores.

##### 3.1.1 Contador de Pulsos

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

Seu circuito básico apresenta um grupo de 4 Flip-Flops do tipo T ou JK Mestre-Escrava, os quais possuem entrada T ou, no caso, J e K iguais a 1, originando na saída  $Q_f = Q_a'$ , a cada descida de clock,

A entrada dos pulsos se faz através da entrada clock do 1º flip-flop, sendo as entradas clock dos flip-flops seguintes, conectadas as saídas Q dos respectivos antecessores conforme circuito visto abaixo:



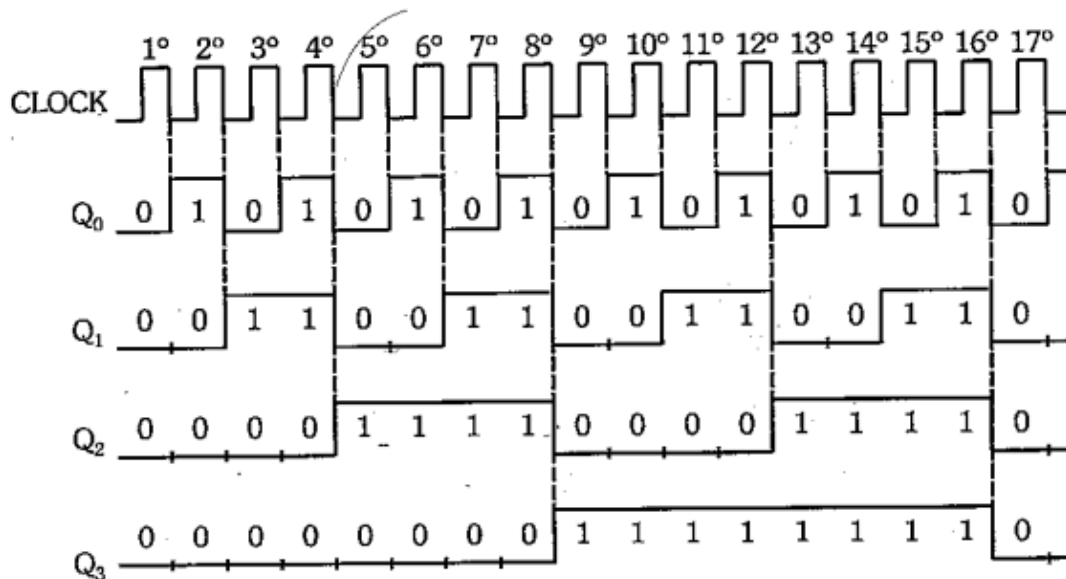
Vamos fazer, inicialmente, com que todos os flip-flops assumam saídas iguais a 0, através da aplicação de um nível 0 à entrada *clear*. A cada descida do pulso de *clock*, o 1º flip-flop irá mudar de estado, sendo esta troca aplicada à entrada do 2º flip-flop, fazendo com que este troque de estado a cada descida da saída Q0, assim sucessivamente.

Vamos analisar este comportamento através da tabela:

Descidas de clock	Saídas				
	Q <sub>0</sub>	Q <sub>1</sub>	Q <sub>2</sub>	Q <sub>3</sub>	
1ª	0	0	0	0	(Estado inicial, imposto por CLR = 0)
2ª	1	0	0	0	(Após a 1ª descida de clock: Q <sub>0</sub> =1)
3ª	0	1	0	0	(Após a 2ª descida: Q <sub>0</sub> =0 e Q <sub>1</sub> =1, obtido pela descida de Q <sub>0</sub> )
4ª	1	1	0	0	(Q <sub>0</sub> =1 e Q <sub>1</sub> permanece igual a 1)
5ª	0	0	1	0	(Q <sub>0</sub> =0 ⇒ Q <sub>1</sub> =0 ⇒ Q <sub>2</sub> =1)
6ª	1	0	1	0	(Q <sub>0</sub> =1, Q <sub>1</sub> e Q <sub>2</sub> permanecem)
7ª	0	1	1	0	(Q <sub>0</sub> =0 ⇒ Q <sub>1</sub> =1)
8ª	1	1	1	0	(Q <sub>0</sub> =1)
9ª	0	0	0	1	(Q <sub>0</sub> =0 ⇒ Q <sub>1</sub> =0 ⇒ Q <sub>2</sub> =0 ⇒ Q <sub>3</sub> =1)
10ª	1	0	0	1	(Q <sub>0</sub> =1)
11ª	0	1	0	1	(Q <sub>0</sub> =0 ⇒ Q <sub>1</sub> =1)
12ª	1	1	0	1	(Q <sub>0</sub> =1)
13ª	0	0	1	1	(Q <sub>0</sub> =0 ⇒ Q <sub>1</sub> =0 ⇒ Q <sub>2</sub> =1)
14ª	1	0	1	1	(Q <sub>0</sub> =1)
15ª	0	1	1	1	(Q <sub>0</sub> =0 ⇒ Q <sub>1</sub> =1)
16ª	1	1	1	1	(Q <sub>0</sub> =1)
17ª	0	0	0	0	(Q <sub>0</sub> =0 ⇒ Q <sub>1</sub> =0 ⇒ Q <sub>2</sub> =0 ⇒ Q <sub>3</sub> =0)

Considerando Q0 como bit menos significativo e Q3 como mais significativo, temos nas saídas o sistema binário em sequência (0000 a 1111). Notamos ainda, que após a 16ª descida do clock, o contador irá reiniciar a contagem:





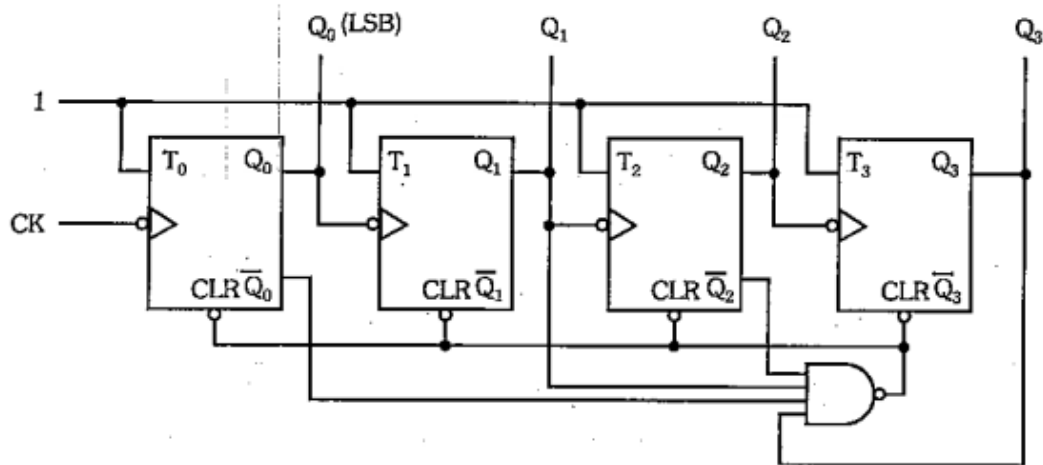
Analisando os gráficos, notamos que o período de Q<sub>0</sub> é o dobro do período do clock, logo, a frequência de Q<sub>0</sub> será a metade da frequência do clock, pois  $f = 1/T$ . Analisando a saída Q<sub>1</sub>, veremos que seu período é o dobro de Q<sub>0</sub> e o quádruplo do clock, logo, na sua frequência será a metade de Q<sub>0</sub> e um quarto da frequência do pulso de clock. Isto se estenderá sucessivamente aos demais flip-flops. Assim sendo, podemos notar que uma das aplicações dos contadores será a de dividir a frequência de sinais (onda quadrada) aplicados a entrada clock. No caso deste contador, a divisão será por um número múltiplo de  $2^N$ , onde N é o número de flip-flops utilizados.

### 3.1.2 Contador de Década

O contador de década é o circuito que efetua a contagem em números binários de 0 a 9<sub>10</sub> (10 algarismos). Isso significa acompanhar a sequência do código BCD 8421 de 0000 até 1001.

Para construir este circuito, utilizamos o contador de pulsos, interligando as entradas clear dos flip-flops.

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), ou seja, no 10<sup>o</sup> pulso.

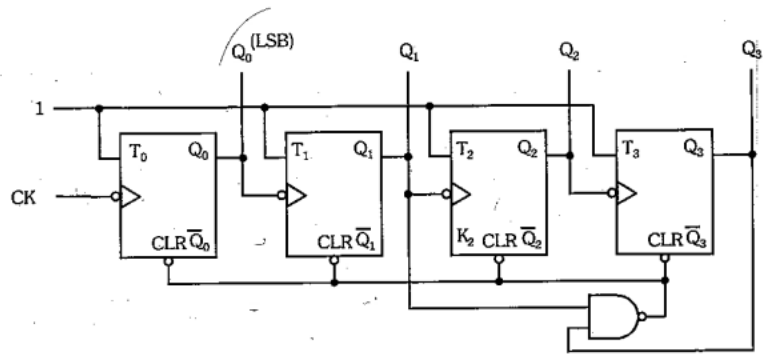


Temos, neste caso, a seguinte tabela da verdade:

Descidas de clock	$Q_3$	$Q_2$	$Q_1$	$Q_0$	CLR
1 <sup>a</sup>	0	0	0	0	1
2 <sup>a</sup>	0	0	0	1	1
3 <sup>a</sup>	0	0	1	0	1
4 <sup>a</sup>	0	0	1	1	1
5 <sup>a</sup>	0	1	0	0	1
6 <sup>a</sup>	0	1	0	1	1
7 <sup>a</sup>	0	1	1	0	1
8 <sup>a</sup>	0	1	1	1	1
9 <sup>a</sup>	1	0	0	0	1
10 <sup>a</sup>	1	0	0	1	1
	1	0	1	0	0

Após a 10<sup>a</sup> descida do clock, o contador tende a assumir o estado 10 = 0,  $Q_1 = 1$ ,  $Q_2 = 0$ ,  $Q_3 = 1$ , porém, neste instante, a entrada clear vai para 0, zerando o contador, ou seja, fazendo com que assumo o estado 0 reiniciando a contagem.

Uma outra forma de obter o mesmo clear ou reset no caso 1010, utilizando uma porta NE com menos entradas, consiste em ligar apenas  $Q_3$  e  $Q_1$  nesta, pois só serão iguais a 1 simultaneamente neste caso, zerando as saídas do mesmo jeito.



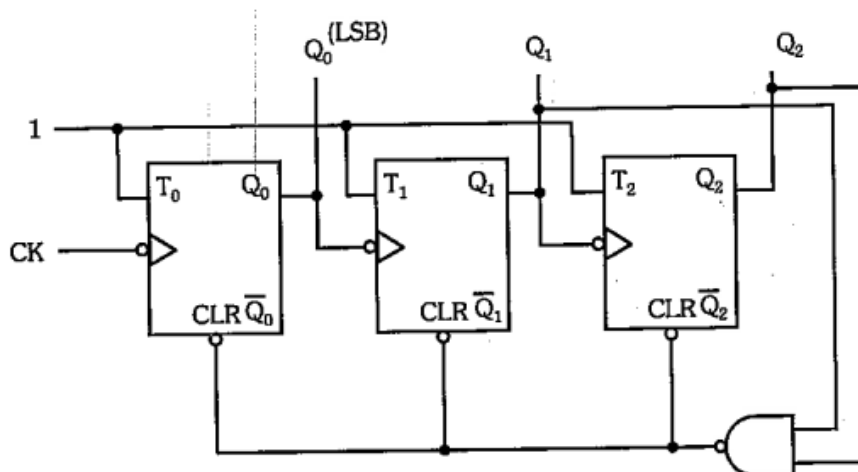
Este contador poderá ser utilizado como divisor de frequência por 10 para uma onda quadrada aplicada à entrada clock, pois possui 10 estados de saída.

### 3.1.3 Contador Sequência de 0 a n

Utilizando o mesmo processo, podemos fazer um contador de 0 até um número n qualquer. Para isso, basta apenas verificarmos quais as saídas do contador para o caso seguinte a n, colocarmos estas saídas numa porta NE e à saída desta ligarmos as entradas clear dos flip-flops.

Para exemplificar, vamos elaborar o circuito de um contador de 0 a 5. Nesse caso, desejamos que o contador recomece a contagem após o estado 5, ou seja, passe para 0 todos os flip-flops.

Neste caso, o estado seguinte a n será o 6, ocasionando nas saídas  $q_2=1$ ,  $Q_1=1$  e  $Q_0=1$  - 110. Quando ocorrer, então deverá haver um 0 nas entradas clear interligadas, levando o contador a 0. Devemos, para tanto, ter na entrada da porta NE, a ligação de  $Q_2$  e  $Q_1$ , pois na sequência da contagem, estas irão assumir níveis 1 simultaneamente apenas no caso 110.



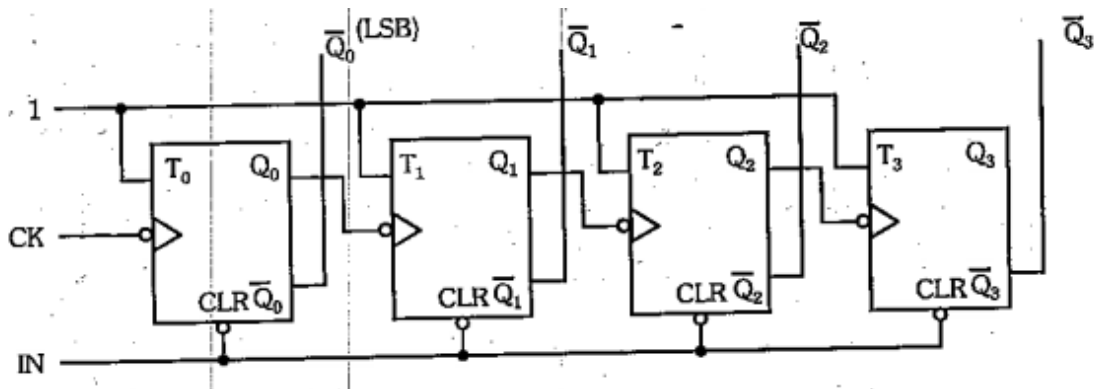
### 3.1.4 Contadores assíncronos decrescentes

Os contadores podem também ser classificados pelo tipo de contagem que executam, ou seja, se executam contagem crescente ou decrescente. A estes contadores damos os nomes de **contadores crescentes** e **contadores decrescentes** respectivamente.

Observe agora, os contadores que efetuam a contagem decrescente.

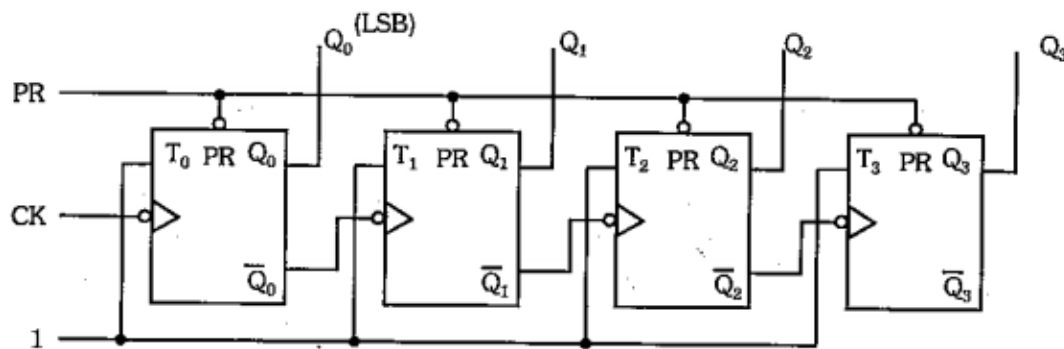
Decimal	Binário
15	1 1 1 1
14	1 1 1 0
13	1 1 0 1
12	1 1 0 0
11	1 0 1 1
10	1 0 1 0
9	1 0 0 1
8	1 0 0 0
7	0 1 1 1
6	0 1 1 0
5	0 1 0 1
4	0 1 0 0
3	0 0 1 1
2	0 0 1 0
1	0 0 0 1
0	0 0 0 0

O circuito que efetua a contagem decrescente é o mesmo circuito que efetua a contagem crescente, com a única diferença de extrairmos as saídas dos terminais Q'0, Q'1, Q'2 e Q'3, sendo que o terminal Q'0, o bit menos significativo. Podemos notar pela tabela verdade, que a contagem decrescente nada mais é que o complemento da contagem crescente.

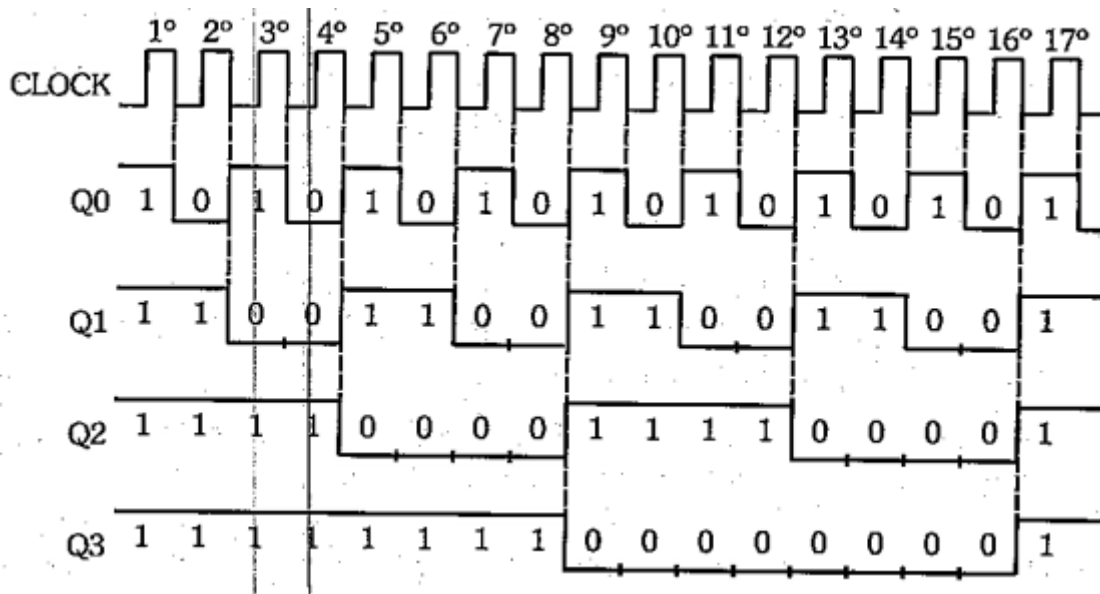


O estado inicial (1111) pode ser obtido pela aplicação de nível 0 na entrada IN, que irá zerar todos os flip-flops nas saídas Q, porém irá impor níveis 1 nas saídas Q'.

Um outro modo de montador decrescente é injetando nas entradas clock dos flip-flops, as saídas complementares como é mostrado na figura abaixo:

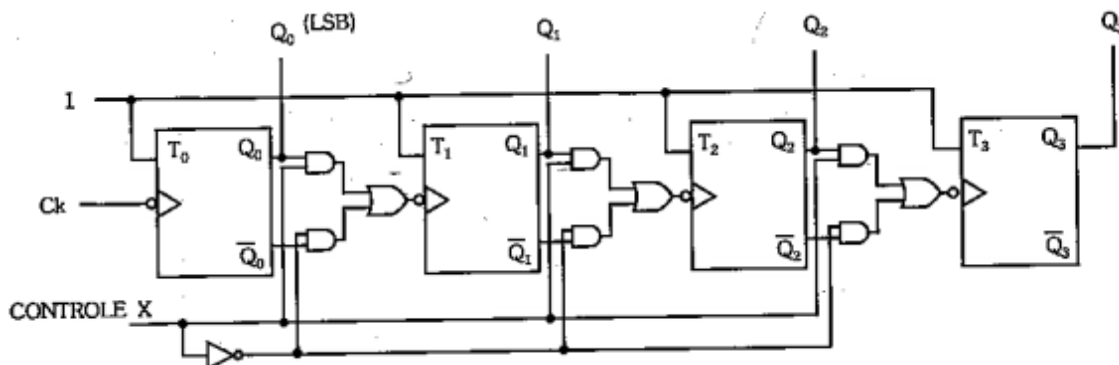


Neste circuito, os clocks dos flip-flops são respectivamente,  $Q_0$ ,  $Q_1$  e  $Q_2$ , logo  $Q_1$ ,  $Q_2$  e  $Q_3$  irão trocar de estado nas subidas de  $Q_0$ ,  $Q_1$  e  $Q_2$ , respectivamente, originando a contagem decrescente. O estado inicial pode ser obtido pela passagem da entrada PR para 0, estabelecendo nível 1 à saída de todos os flip-flops. A figura abaixo mostra todas as formas de onda do sistema, desde a aplicação de uma onda quadrada à entrada clock.



### 3.1.5 Contador Assíncrno Crescente/Decrescente

Podemos contruir 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.



Notamos que no circuito, quando o controle X estiver em 1, as saídas Q'0, Q'1 e Q'2 estarão bloqueadas, fazendo cm que entre as saídas Q0,Q1 e Q2 nas entradas clock dos flip-flops respectivamente. Isso fará com que o contador conte crescentemente.

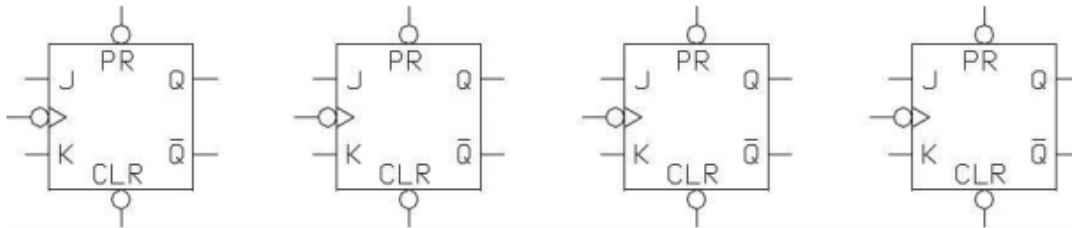
Quando o controle X estiver em 0, a situação inverter-se-á e, por conseguinte, o contador contará decrescentemente.

Notamos, ainda que Q0 será a saída do bit menos significativo.

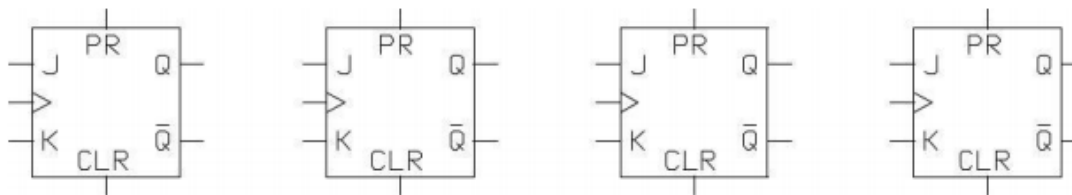
Contador crescente/decrecente é também denominado Up/Down counter.

### 3.1.6 Exercícios de Contadores Assíncronos.

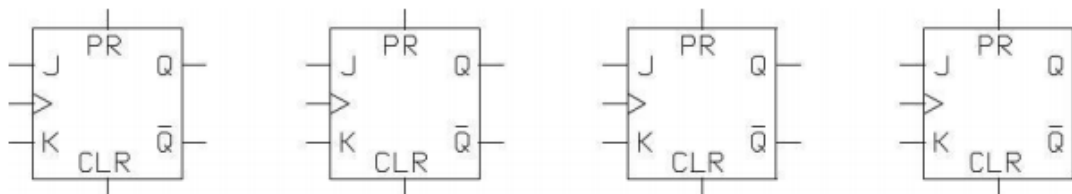
1- Interligue os FF abaixo de modo a formar um contador de 0 a 13 com terminal de RESET, indicando as saídas a entrada de clock. Acrescente as portas lógicas necessárias.



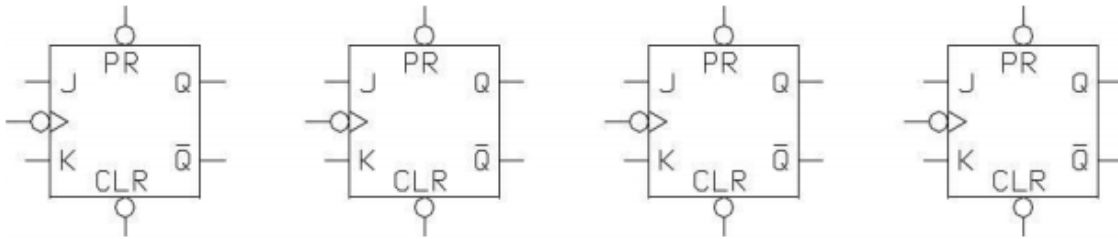
2- Interligue os FF abaixo de modo a formar um contador de 0 a 14 com terminal de RESET, indicando as saídas a entrada de clock. Acrescente as portas lógicas necessárias.



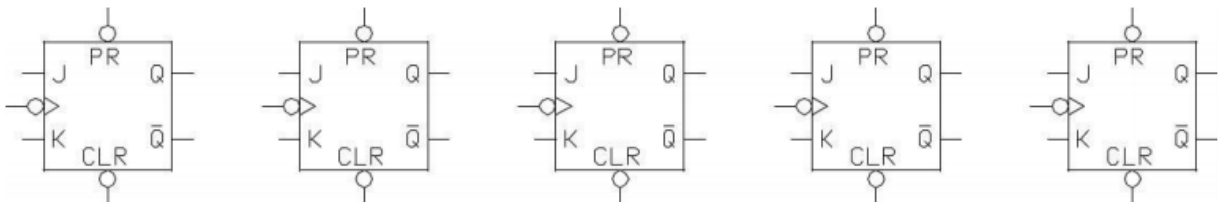
3- Interligue os FF abaixo de modo a formar um contador de 15 a 6 com terminal de RESET, indicando as saídas a entrada de clock. Acrescente as portas lógicas necessárias.



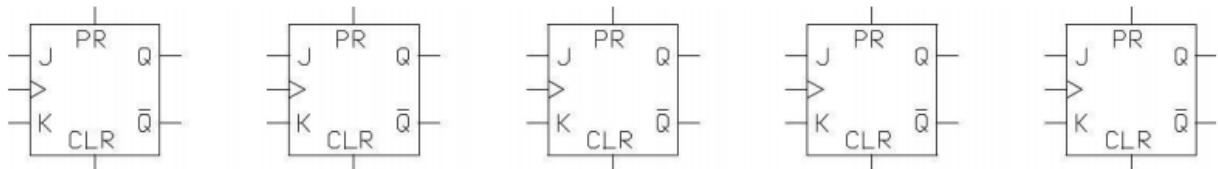
4- Interligue os FF abaixo de modo a formar um contador de 9 a 0 com terminal de RESET, indicando as saídas a entrada de clock. Acrescente as portas lógicas necessárias.



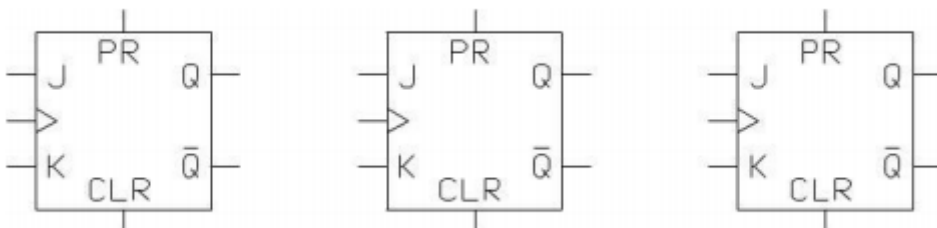
5- Interligue os FF abaixo de modo a formar um contador de 0 a 25 com terminal de RESET, indicando as saídas a entrada de clock. Acrescente as portas lógicas necessárias.



6- Interligue os FF abaixo de modo a formar um contador de 5 a 19 com terminal de RESET, indicando as saídas a entrada de clock. Acrescente as portas lógicas necessárias.

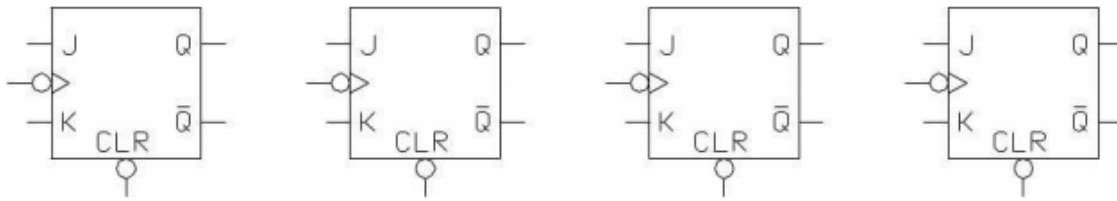


7- Interligue os FF abaixo de modo a formar um contador de 3 a 7 com terminal de RESET, indicando as saídas a entrada de clock. Acrescente as portas lógicas necessárias.

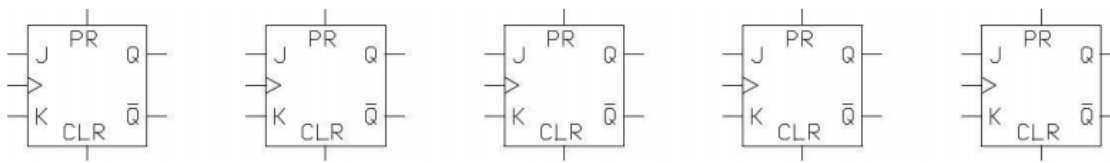




8- Interligue os FF abaixo de modo a formar um contador de 6 a 11 com terminal de RESET, indicando as saídas a entrada de clock. Acrescente as portas lógicas necessárias.



9- Interligue os FF abaixo de modo a formar um divisor de frequência por 20. Indique a entrada e a saída do divisor.



### 3.2 Contadores Síncronos

Estes contadores possuem entradas clock curto-circuitadas, ou seja, o clock entra em todos os flip-flops simultaneamente, fazendo todos atuarem de forma sincronizada.

Para que haja mudanças de estado, devemos então estudar o comportamento das entradas J e K dos vários flip-flops, para que tenhamos nas saídas, as sequências desejadas.

Para estudarmos os contadores síncronos devemos sempre escrever a tabela da verdade, estudando quais devem ser as entradas J e K dos vários flip-flops, para que estes assumam o estado seguinte. Para isso, vamos utilizar a tabela da verdade do flip-flop JK:

J	K	Qf	
0	0	Qa	(mantém o estado)
0	1	0	(fixa 0)
1	0	1	(fixa 1)
1	1	$\bar{Q}a$	(inverte o estado)

A partir desta tabela, construímos outras, relacionando os estados de saída e as entradas J e K.

Qa(atual)	Qf(futuro)	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

**1ª Linha:** A saída atual Qa vale 0 (Qa = 0) e eu quero que a próxima saída Qf valha também 0 (Qf = 0) então eu vou precisar que a entrada J tenha o valor de 0 (J = 0) agora se olharmos a Tabela da Verdade do FF JK vemos que se o K valer 1 ou 0 a saída ainda será 0 (Qf = 0) por tanto atribuímos X ao K que significa tanto faz.

**2ª Linha:** A saída atual  $Q_a$  vale 0 ( $Q_a = 0$ ) e eu quero que a próxima saída  $Q_f$  valha também 1 ( $Q_f = 1$ ) então eu vou precisar que a entrada  $J$  tenha o valor de 1 ( $J = 1$ ) agora se olharmos a Tabela da Verdade do FF JK vemos que se o  $K$  valer 1 ou 0 a saída ainda será 1 ( $Q_f = 1$ ) por tanto atribuímos X ao  $K$  que significa tanto faz.

**3ª Linha:** A saída atual  $Q_a$  vale 1 ( $Q_a = 1$ ) e eu quero que a próxima saída  $Q_f$  valha 0 ( $Q_f = 0$ ) então ao olharmos a tabela da verdade a entrada  $J$  pode valer 0 ou 1 portanto  $J = X$  e a entrada  $K$  valha 1 ( $K = 1$ ).

**4ª Linha:** A saída atual  $Q_a$  vale 1 ( $Q_a = 1$ ) e eu quero que a próxima saída  $Q_f$  valha 1 ( $Q_f = 1$ ) então ao olharmos a tabela da verdade a entrada  $J$  pode valer 0 ou 1 portanto  $J = X$  e a entrada  $K$  valha 0 ( $K = 0$ ).

### 3.2.1 Contadores Síncronos Gerador de Código Binário de 4 Bits

Cada Flip-Flop **equivale a um bit** para isso vamos utilizar 4 flip-flops JK Mestre-Escravo, a tabela abaixo apresenta a sequência máxima alcançada.

ck	$Q_3$	$Q_2$	$Q_1$	$Q_0$
1ª	0	0	0	0
2ª	0	0	0	1
3ª	0	0	1	0
4ª	0	0	1	1
5ª	0	1	0	0
6ª	0	1	0	1
7ª	0	1	1	0
8ª	0	1	1	1
9ª	1	0	0	0
10ª	1	0	0	1
11ª	1	0	1	0
12ª	1	0	1	1
13ª	1	1	0	0
14ª	1	1	0	1
15ª	1	1	1	0
16ª	1	1	1	1

Essa tabela apresenta a sequência que as saídas dos flip-flops devem assumir em função da presença de pulsos de clock. Para o projeto, devemos estudar, para cada caso, o comportamento das entradas J e K dos flip-flops e levantar o circuito necessário para gerar a sequência.

1º Passo: Como sabemos que são 4 flip-flops sabemos também que será necessário 4 saídas Q – vamos nomeá-las com – Q3, Q2,Q1,Q0 e escrever os equivalente binários de 0 a 15 conforme ilustrado na tabela abaixo.

<b>Decimal</b>	<b>Q3</b>	<b>Q2</b>	<b>Q1</b>	<b>Q0</b>
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

2º Passo: Para da saída Q tenha em mente que haverá uma entrada J e uma K. Logo para Q3 então haverá J3 e K3, Q2 haverá J2 e K2 e por ai vai. Então vamos sempre analisar o estado futuro.

Decimal	Q3	Q2	Q1	Q0	J3	K3	J2	K2	J1	K1	J0	K0
0	0	0	0	0	0	X	0	X	0	X	1	X
1	0	0	0	1	0	X	0	X	1	X	X	1
2	0	0	1	0	0	X	0	X	X	0	1	X
3	0	0	1	1	0	X	1	X	X	1	X	1
4	0	1	0	0	0	X	X	0	0	X	1	X
5	0	1	0	1	0	X	X	0	X	X	X	1
6	0	1	1	0	0	X	X	0	X	0	1	X
7	0	1	1	1	1	X	X	1	X	1	X	1
8	1	0	0	0	X	0	0	X	0	X	1	X
9	1	0	0	1	X	0	0	X	1	X	X	1
10	1	0	1	0	X	0	0	X	0	0	1	X
11	1	0	1	1	X	0	1	X	X	1	X	1
12	1	1	0	0	X	0	X	0	0	X	1	X
13	1	1	0	1	X	0	X	0	1	X	X	1
14	1	1	1	0	X	0	X	0	X	0	1	X
15	1	1	1	1	X	1	X	1	X	1	X	1

3º Passo: Notamos que, no projeto, o estado 0 foi considerado após o estado 15, pois ao final, o contador deve reiniciar a contagem.

Para obter as epressões de J3, K3, J2, K2, J1, K1, J0 e K0, simplificado vamos utilizar o mapa de Karnaugh.

**J<sub>3</sub>:**

	$\bar{Q}_1$	$Q_1$	
$\bar{Q}_3$	0	0	0
	0	1	0
$Q_3$	X	X	X
	X	X	X
	$\bar{Q}_0$	$Q_0$	$\bar{Q}_0$

(a)

$$J_3 = Q_2 Q_1 Q_0$$

$$\therefore J_3 = K_3 = Q_2 Q_1 Q_0$$

**K<sub>3</sub>:**

	$\bar{Q}_1$	$Q_1$	
$\bar{Q}_3$	X	X	X
	X	X	X
$Q_3$	0	0	0
	0	0	0
	$\bar{Q}_0$	$Q_0$	$\bar{Q}_0$

(b)

$$K_3 = Q_2 Q_1 Q_0$$

**J<sub>2</sub>:**

	$\bar{Q}_1$	$Q_1$	
$\bar{Q}_3$	0	0	0
	X	X	X
$Q_3$	X	X	X
	0	0	0
	$\bar{Q}_0$	$Q_0$	$\bar{Q}_0$

(c)

$$J_2 = Q_1 Q_0$$

$$\therefore J_2 = K_2 = Q_1 Q_0$$

**K<sub>2</sub>:**

	$\bar{Q}_1$	$Q_1$	
$\bar{Q}_3$	X	X	X
	0	0	0
$Q_3$	0	0	0
	X	X	X
	$\bar{Q}_0$	$Q_0$	$\bar{Q}_0$

(d)

$$K_2 = Q_1 Q_0$$

**J<sub>1</sub>:**

	$\bar{Q}_1$	$Q_1$	
$\bar{Q}_3$	0	1	X
$Q_3$	0	1	X
$\bar{Q}_2$	0	1	X
$Q_2$	0	1	X
$\bar{Q}_0$	0	1	X
$Q_0$	0	1	X

(e)

$$J_1 = Q_0$$

$$\therefore J_1 = K_1 = Q_0$$

**K<sub>1</sub>:**

	$\bar{Q}_1$	$Q_1$	
$\bar{Q}_3$	X	X	1
$Q_3$	X	X	1
$\bar{Q}_2$	X	X	1
$Q_2$	X	X	1
$\bar{Q}_0$	X	X	1
$Q_0$	X	X	1

(f)

$$K_1 = Q_0$$

**J<sub>0</sub>:**

	$\bar{Q}_1$	$Q_1$	
$\bar{Q}_3$	1	X	X
$Q_3$	1	X	X
$\bar{Q}_2$	1	X	X
$Q_2$	1	X	X
$\bar{Q}_0$	1	X	X
$Q_0$	1	X	X

(g)

$$J_0 = 1$$

$$\therefore J_0 = K_0 = 1$$

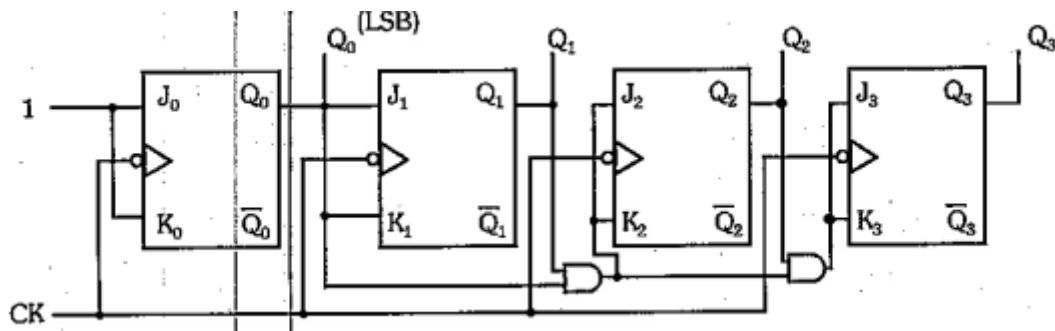
**K<sub>0</sub>:**

	$\bar{Q}_1$	$Q_1$	
$\bar{Q}_3$	X	1	1
$Q_3$	X	1	1
$\bar{Q}_2$	X	1	1
$Q_2$	X	1	1
$\bar{Q}_0$	X	1	1
$Q_0$	X	1	1

(h)

$$K_0 = 1$$

4º Passo: Montando o circuito de acordo com as equações.



### 3.2.2 Contador de década

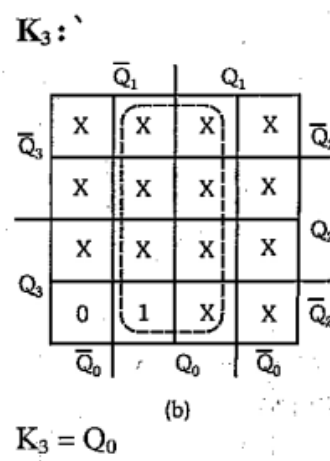
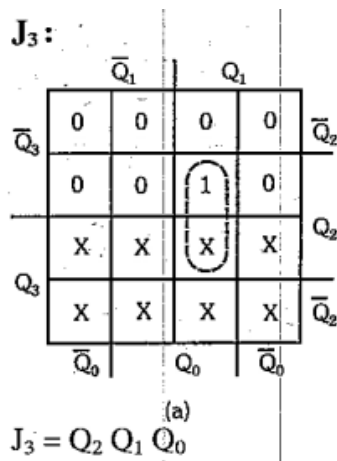
Vamos construir um contador de década síncrono. Para isso, utilizaremos o mesmo processo já vista.

Primeiramente, vamos verificar o comportamento das entradas J e K.

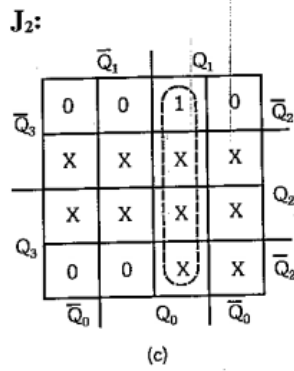
$Q_3$	$Q_2$	$Q_1$	$Q_0$	$J_3$	$K_3$	$J_2$	$K_2$	$J_1$	$K_1$	$J_0$	$K_0$
0	0	0	0	0	X	0	X	0	X	1	X
0	0	0	1	0	X	0	X	1	X	X	1
0	0	1	0	0	X	0	X	X	0	1	X
0	0	1	1	0	X	1	X	X	1	X	1
0	1	0	0	0	X	X	0	0	X	1	X
0	1	0	1	0	X	X	0	1	X	X	1
0	1	1	0	0	X	X	0	X	0	1	X
0	1	1	1	1	X	X	1	X	1	X	1
1	0	0	0	X	0	0	X	0	X	1	X
1	0	0	1	X	1	0	X	0	X	X	1

Supondo conseguir o estado inicial através das entradas clear, vamos considerar os estados não pertencentes à sequência como irrelevantes.

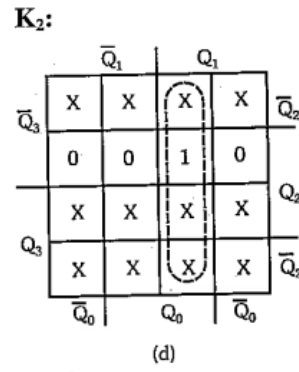
Vamos transpor para o mapa de Karnaugh e simplificar.



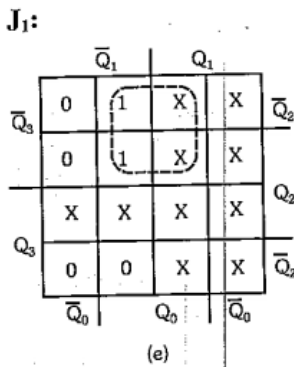




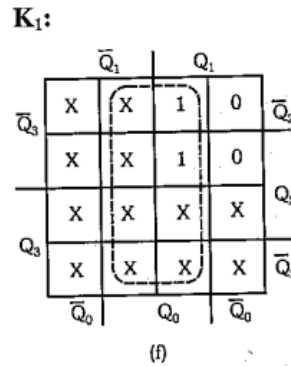
$$J_2 = Q_1 Q_0$$



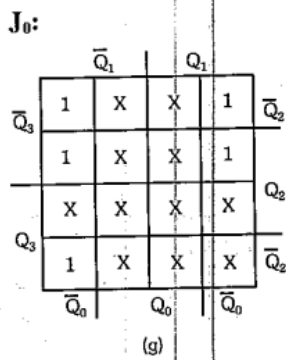
$$K_2 = Q_1 Q_0$$



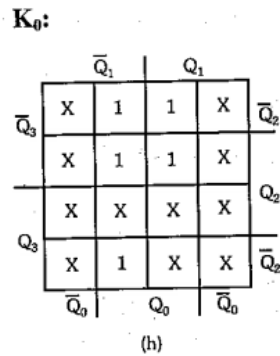
$$J_1 = Q_0 \bar{Q}_3$$



$$K_1 = Q_0$$

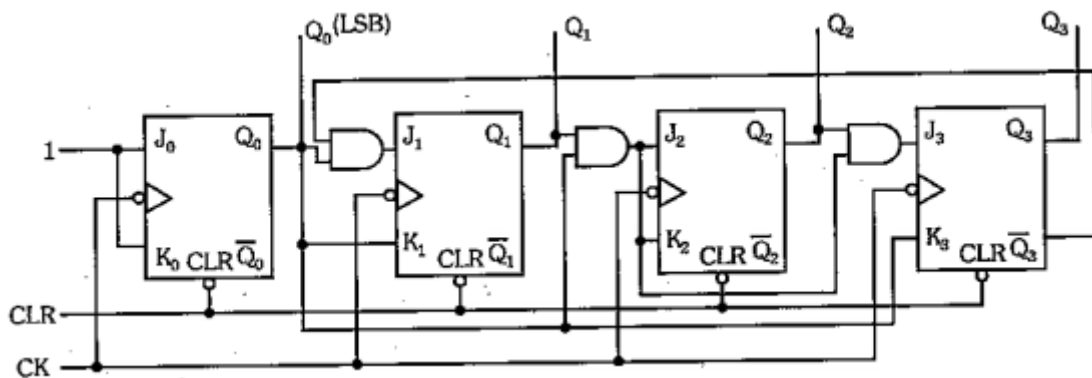


$$J_0 = 1$$



$$K_0 = 1$$

Vamos, mediante as expressões obtidas, esquematizar o circuito do contador de década.

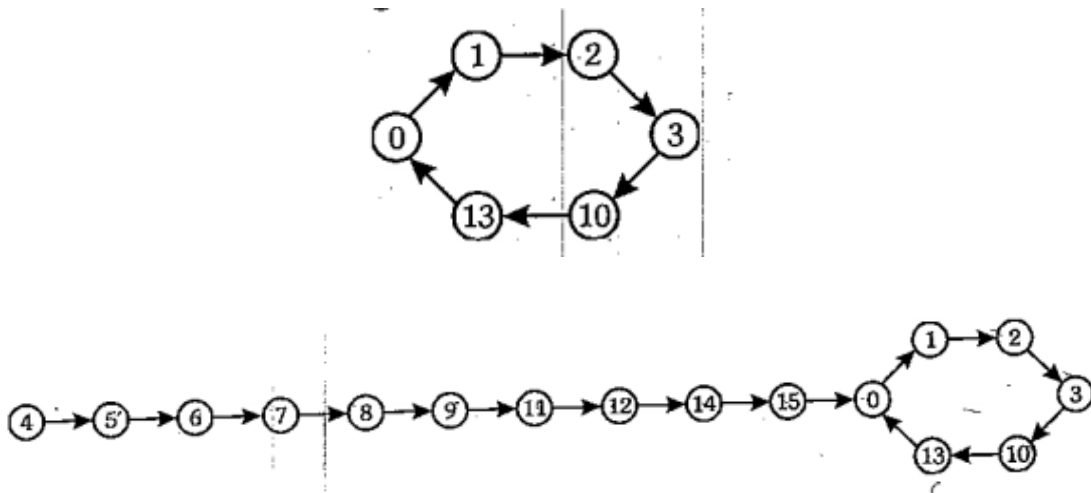


### 3.2.3 Gerador de uma sequência qualquer

Podemos construir um contador que gere uma sequência qualquer. Para isso, basta estabelecermos a sequência e seguirmos o método já conhecido, ou seja, o da determinação das entradas J e K. Os estados que não fizerem parte da sequência deverão ser considerados como **condições irrelevantes**.

Para exemplificarmos, vamos construir um contador que gere a sequência: 0,1,2,3,10,13,0

O loop que o contador deve efetuar para acompanhar a sequência é visto no diagrama de estados abaixo:



1º Passo: Como iremos contar até 13 obrigatoriamente vamos precisar de 4 bits, isso significa, 4 flip-flops JK mestre escravo.

Decimal	Atual			
	Q3	Q2	Q1	Q0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

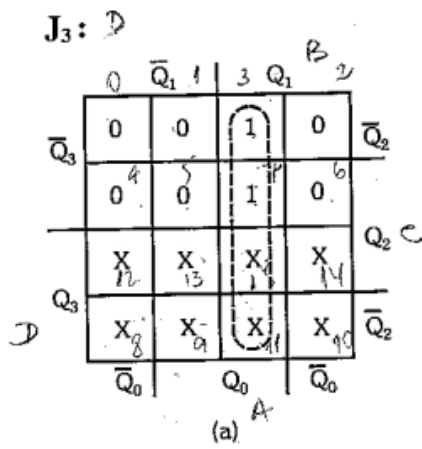
2º Passo: Vamos adicionar agora os próximos estados futuros da tabela.

Decimal	Atual				Futuro			
	Q3	Q2	Q1	Q0	Q3	Q2	Q1	Q0
0	0	0	0	0	0	0	0	1
1	0	0	0	1	0	0	1	0
2	0	0	1	0	0	0	1	1
3	0	0	1	1	1	0	1	0
4	0	1	0	0	X	X	X	X
5	0	1	0	1	X	X	X	X
6	0	1	1	0	X	X	X	X
7	0	1	1	1	X	X	X	X
8	1	0	0	0	X	X	X	X
9	1	0	0	1	X	X	X	X
10	1	0	1	0	1	1	0	1
11	1	0	1	1	X	X	X	X
12	1	1	0	0	X	X	X	X
13	1	1	0	1	0	0	0	0
14	1	1	1	0	X	X	X	X
15	1	1	1	1	X	X	X	X

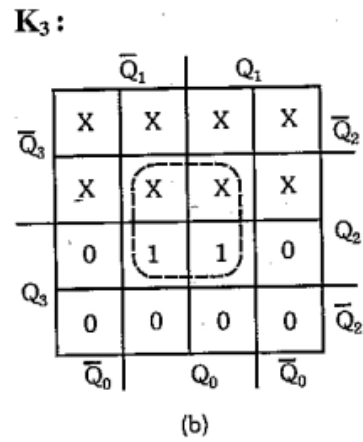
3º Passo: Agora iremos combinar as hipóteses.

Decimal	Atual				Futuro				J3	K3	J2	K2	J1	K1	J0	K0
	Q3	Q2	Q1	Q0	Q3	Q2	Q1	Q0								
0	0	0	0	0	0	0	0	1	0	X	0	X	0	X	1	X
1	0	0	0	1	0	0	1	0	0	X	0	X	1	X	X	1
2	0	0	1	0	0	0	1	1	0	X	0	X	X	0	1	X
3	0	0	1	1	1	0	1	0	1	X	0	X	X	0	X	1
4	0	1	0	0	X	X	X	X	X	X	X	X	X	X	X	X
5	0	1	0	1	X	X	X	X	X	X	X	X	X	X	X	X
6	0	1	1	0	X	X	X	X	X	X	X	X	X	X	X	X
7	0	1	1	1	X	X	X	X	X	X	X	X	X	X	X	X
8	1	0	0	0	X	X	X	X	X	X	X	X	X	X	X	X
9	1	0	0	1	X	X	X	X	X	X	X	X	X	X	X	X
10	1	0	1	0	1	1	0	1	X	0	1	X	X	1	1	X
11	1	0	1	1	X	X	X	X	X	X	X	X	X	X	X	X
12	1	1	0	0	X	X	X	X	X	X	X	X	X	X	X	X
13	1	1	0	1	0	0	0	0	X	1	X	1	0	X	X	1
14	1	1	1	0	X	X	X	X	X	X	X	X	X	X	X	X
15	1	1	1	1	X	X	X	X	X	X	X	X	X	X	X	X

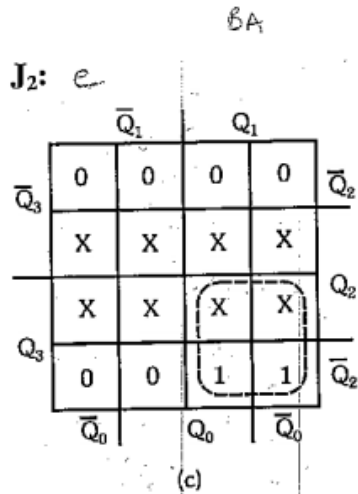
4º Passo: Vamos, agora, mediante a utilização dos diagramas, obter as expressões simplificadas das entradas J e K.



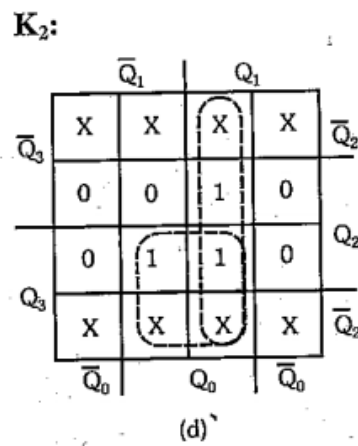
$$J_3 = Q_1 Q_0$$



$$K_3 = Q_2 Q_0$$



$$J_2 = Q_3 Q_1$$



$$K_2 = Q_3 Q_0 + Q_1 Q_0$$

$J_1: \text{ } \ell$

	$\bar{Q}_1$	$Q_1$	
$\bar{Q}_3$	0	1 X	X
	0	1 X	X
$Q_3$	1	0 X	X
	0	1 X	X
	$\bar{Q}_0$	$Q_0$	$\bar{Q}_0$

(e)

$$J_1 = Q_0 \bar{Q}_3 + Q_0 \bar{Q}_2 + Q_3 Q_2 \bar{Q}_0$$

$$J_1 = Q_0 (\bar{Q}_3 + \bar{Q}_2) + \bar{Q}_0 (Q_3 Q_2)$$

$$J_1 = Q_0 (\overline{Q_3 Q_2}) + \bar{Q}_0 (Q_3 Q_2)$$

$$J_1 = Q_0 \oplus (Q_3 Q_2)$$

$K_1:$

	$\bar{Q}_1$	$Q_1$	
$\bar{Q}_3$	X	X	0 0
	X	X	1 0
$Q_3$	X	X	1 0
	X	X	1 1
	$\bar{Q}_0$	$Q_0$	$\bar{Q}_0$

(f)

$$K_1 = Q_2 Q_0 + Q_3 \bar{Q}_2$$

$J_0: \text{ } \lambda$

	$\bar{Q}_1$	$Q_1$	
$\bar{Q}_3$	1 X	X 1	
	1 X	X 1	
$Q_3$	0 X	X 1	
	1 X	X 1	
	$\bar{Q}_0$	$Q_0$	$\bar{Q}_0$

(g)

$$J_0 = \bar{Q}_3 + \bar{Q}_2 + Q_1$$

$K_0:$

	$\bar{Q}_1$	$Q_1$	
$\bar{Q}_3$	X	1	1 X
	X	1	1 X
$Q_3$	X	1	1 X
	X	0	1 X
	$\bar{Q}_0$	$Q_0$	$\bar{Q}_0$

(h)

$$K_0 = \bar{Q}_3 + Q_2 + Q_1$$

5º Passo: O circuito obtido de acordo com a tabela.





## 4. Multiplexadores, Demultiplexadores e Memórias

Neste capítulo, vamos falar de assuntos de grande importância. Trata-se do Multiplex, do Demultiplex e das Memórias, utilizáveis em circuitos com Microprocessadores.

Os circuitos multiplex são utilizados nos casos em que necessitamos enviar em certo número de informações, contidas em vários canais, a um só canal.

Os circuitos demultiplex efetuam a função inversa à dos multiplex, ou seja, enviam as informações, vindas de um único canal, a vários canais.

Ambos os circuitos são largamente empregados dentro de sistemas digitais, bem como na área de transmissão de dados.

As memórias são blocos que armazenam informações codificadas digitalmente. Dividem-se basicamente em dois grupos: as memórias de escrita e leitura e de apenas leitura. Têm sua grande aplicação em sistemas digitais, utilizando principalmente na área da informática.

### 4.1 Geração de Produtos Canônicos

Como foi visto, com  $n$  variáveis booleanas podemos fazer  $2^n$  combinações. Por exemplo, com 2 variáveis podemos formar  $2^2 = 4$  possibilidades, sendo estas:

$$\begin{array}{ll} 0) \bar{A} \cdot \bar{B} \rightarrow A = 0 & \text{e} \quad B = 0 \\ 1) \bar{A} \cdot B \rightarrow A = 0 & \text{e} \quad B = 1 \\ 2) A \cdot \bar{B} \rightarrow A = 1 & \text{e} \quad B = 0 \\ 3) A \cdot B \rightarrow A = 1 & \text{e} \quad B = 1 \end{array}$$

Vamos considerar a expressão referente ao caso 0:  $P_0 = A' \cdot B'$ . Este produto será igual a 1 somente quando  $A = B = 0$ .

No caso 1, temos:  $P_1 = A' \cdot B$ , que será igual a 1 somente quando  $A = 0$  e  $B = 1$ .

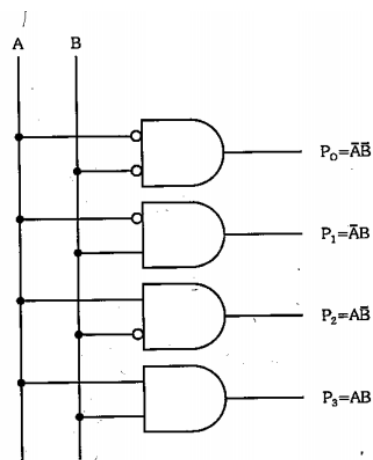
No caso 2, temos:  $P_2 = A \cdot B'$ , que será igual a 1 somente quando  $A = 1$  e  $B = 0$ .

No caso 3, temos:  $P_3 = A \cdot B$ , que será igual a 1 somente quando  $A = 1$  e  $B = 1$

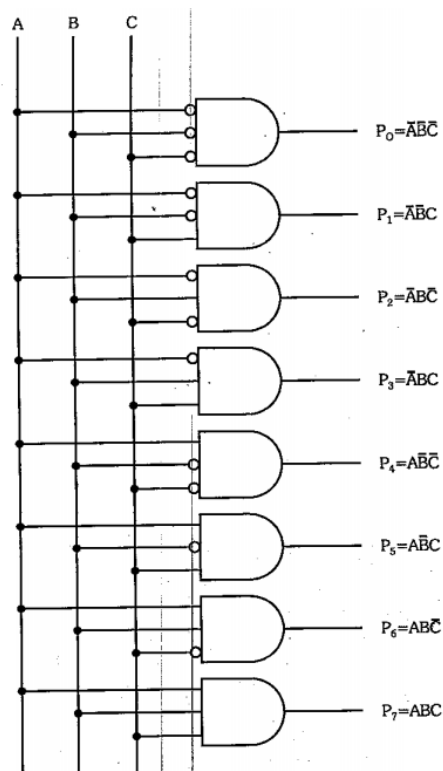
Estes quatro produtos possíveis com 2 variáveis são denominados **produtos canônicos**. Então, com  $n$  variáveis, temos  $2^n$  produtos canônicos;

#### 4.1.1 Circuito Básico Gerador de Produtos Canônicos

Podemos esquematizar circuitos para gerar produtos canônicos. Um primeiro e mais simples de ser entendido é o constituído por portas AND e inversores. A figura abaixo mostra um exemplo para 2 variáveis de entrada.

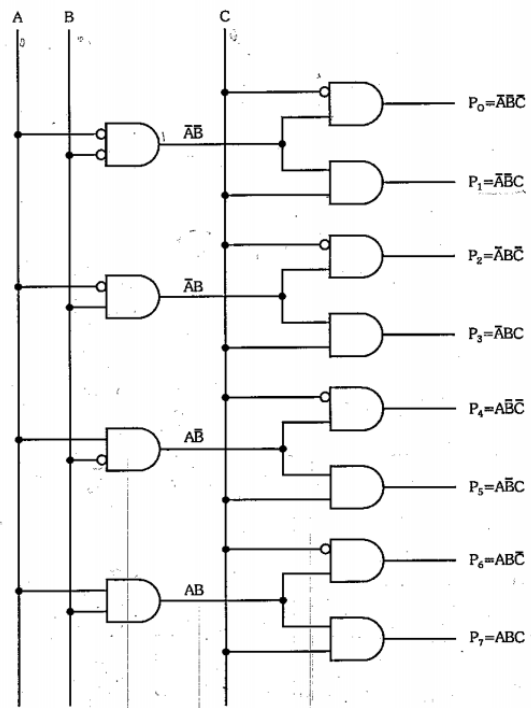


Seguindo o mesmo esquema básico, para 3 variáveis, temos o circuito abaixo:



### 4.1.2 Matriz de Simples Encadeamento

Um segundo processo de geração de produtos canônicos é o conhecido **Matriz de Simples Encadeamento**, que utiliza somente portas AND de 2 entradas. O circuito no caso de 2 variáveis, é idêntico ao já visto, utilizando 4 portas AND de 2 entradas. Para 3 variáveis, temos o circuito abaixo:



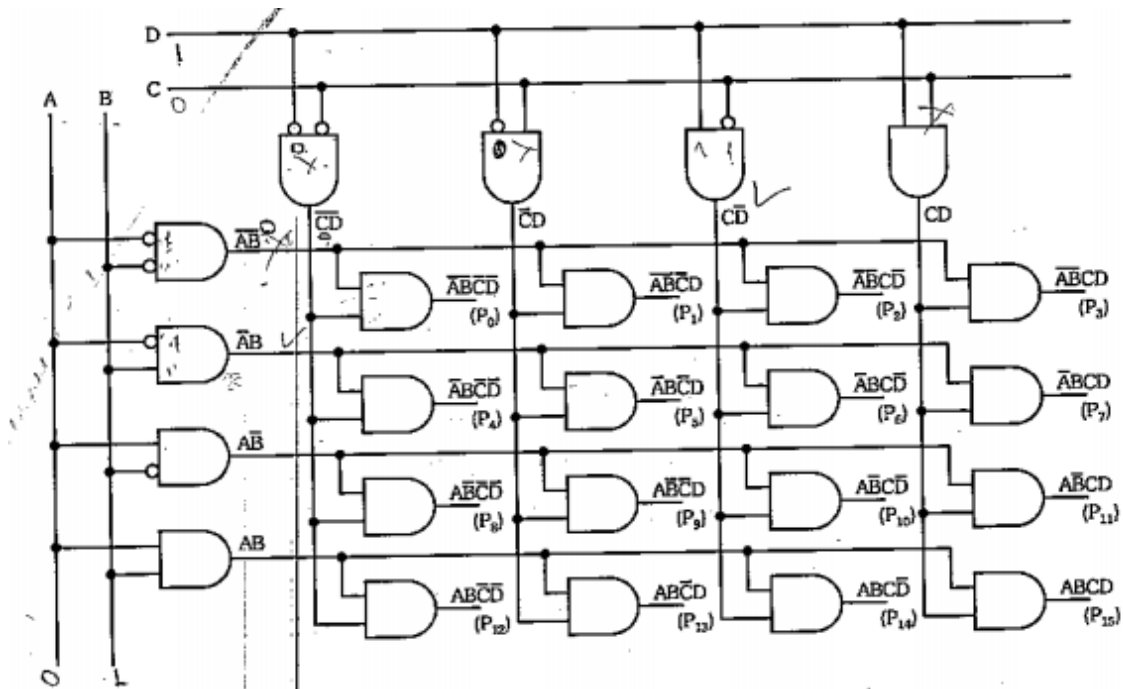
Notamos que este circuito foi desenvolvido a partir de 2 variáveis, visto anteriormente. Se quisermos montar um gerador de produtos canônicos de 4 variáveis, basta colocar 2 portas AND com entradas D' e D, respectivamente, em cada saída do circuito de 3 variáveis e assim sucessivamente para maior número de variáveis.

Para n variáveis, temos N portas de 2 entradas onde  $N = 2^{n+1} - 4$ . Este tipo de matriz é também conhecido como piramidal.

### 4.1.3 Matriz de Duplo Encadeamento

Este tipo de matriz é muito importante pelo fato de ser utilizado em circuitos multiplex e na estrutura de algumas memórias.

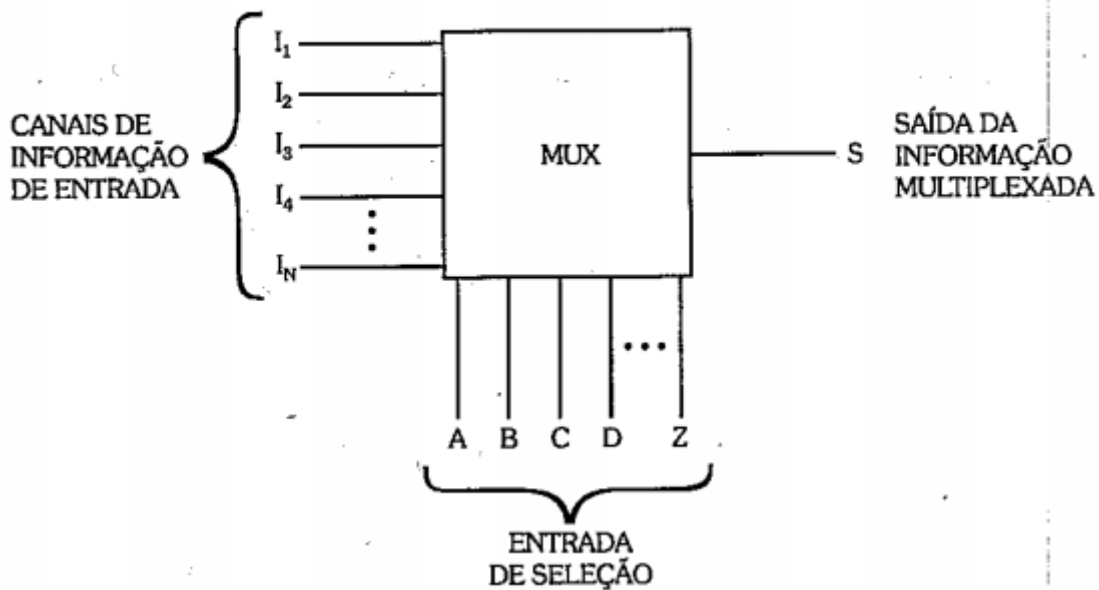
Vamos construir uma matriz de duplo encadeamento para a geração de produtos canônicos de 4 variáveis.



Para entendermos o funcionamento desta matriz, vamos utilizar, por exemplo, a entrada 5 (0101). Neste caso,  $P_5$  ( $\overline{A}B\overline{C}D$ ) estará em nível 1 e todas as outras saídas estão em nível 0. Analisando os demais casos, veremos que cada um apresentará saída 1 para uma entrada específica.

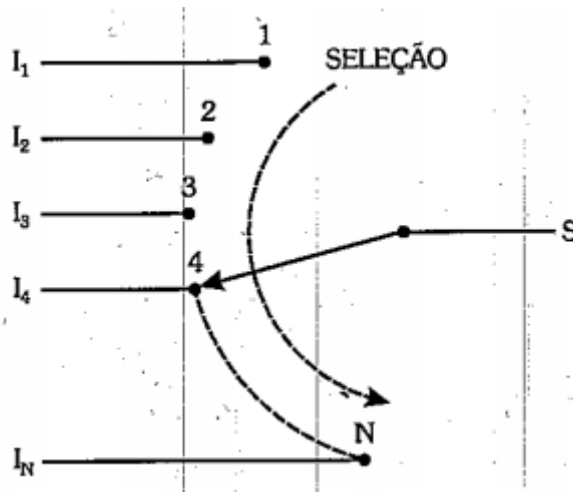
### 4.2 Multiplex

Como dissemos no início deste capítulo, o circuito multiplex é utilizado para enviarmos as informações contidas em vários canais (fios), a um só canal (fio). Esquematizando o bloco multiplex, temos:



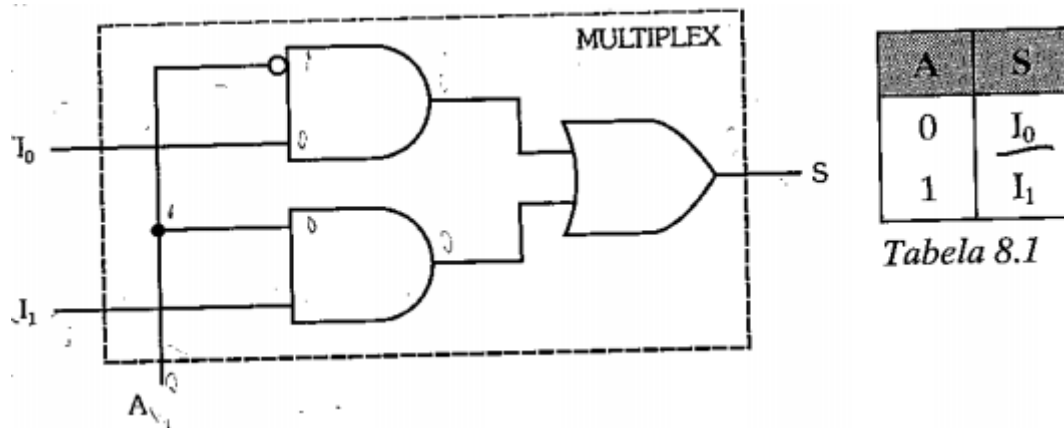
A entrada de seleção tem como finalidade escolher qual das informações de entrada, ou qual dos canais de informação deve ser ligado a saída.

Um circuito elementar que efetua uma multiplexão é uma chave seletora de 1 pólo de n posições.



Se quisermos ligar, por exemplo, a informa  $I_1$  na saída, basta selecionarmos a posição 1 da chave seletora. Se quisermos conectar á saída a informação  $I_2$ , selecionamos a posição 2 e assim, sucessivamente.

Este é o processo básico do funcionamento de um multiplex, sendo que as entradas de seleção irão indicar qual a informação a ser conectada à saída, ou seja, no exemplo, as variáveis de seleção irão comutar a posição da chave seletora.



#### 4.2.1 Projeto do Circuito de um Multiplex

Para projetarmos um multiplex, devemos relacionar, principalmente, a possibilidade de que as entradas de seleção irão assumir com a informação de entrada que deve ser conectada à saída. Para isso, montamos uma tabela da verdade onde serão colocadas todas as possibilidades de seleção e as respectivas informações que devem aparecer na saída.

Para mostrarmos passo a passo a elaboração de multiplex, vamos iniciar, efetuando o projeto de um multiplex de 4 canais ou entradas de informações.

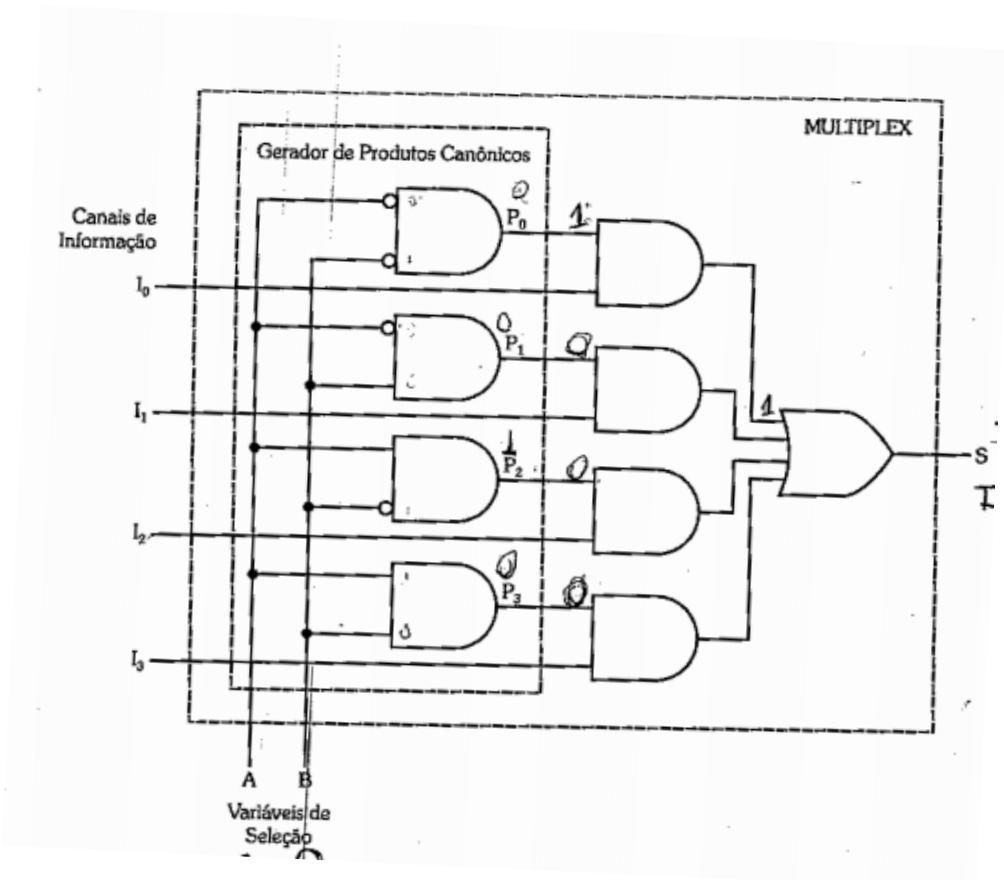
Para que possamos conectar aleatoriamente 4 entradas à saída, necessitamos de 2 variáveis de seleção. Com isso, podemos montar a tabela da verdade:

Variáveis de Seleção		Saída
A	B	S
0	0	$I_0$
0	1	$I_1$
1	0	$I_2$
1	1	$I_3$

Montando a tabela, relacionamos os valores assumidos pela saída para cada possibilidade das variáveis de seleção, obtendo, a partir disso, o respectivo produto canônico.

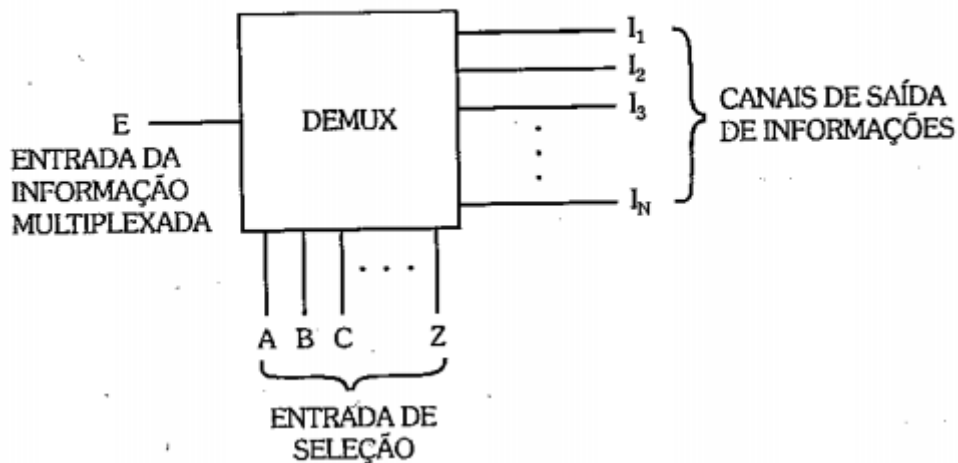
Variáveis de Seleção:	Situação na saída:
Caso 0 0 ( $P_0 = \bar{A} \cdot \bar{B}$ )	$S = I_0$
Caso 0 1 ( $P_1 = \bar{A} \cdot B$ )	$S = I_1$
Caso 1 0 ( $P_2 = A \cdot \bar{B}$ )	$S = I_2$
Caso 1 1 ( $P_3 = A \cdot B$ )	$S = I_3$

Em função destas expressões, esquematizados o circuito. A figura abaixo mostra o circuito obtido do multiplex de 4 canais propostos.



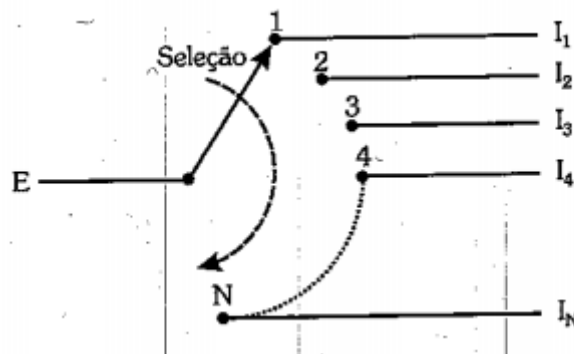
### 4.3 Demultiplex

Entende-se por demultiplex como sendo o bloco que efetua a função inversa ao multiplex, ou seja, a de enviar informações contidas em um canal a vários canais de saída. A figura abaixo mostra um bloco demultiplex genérico.



As entradas de seleção têm como finalidade escolher qual o canal de informação de saída que deve ser conectado a entrada, ou seja, deverá endereçar o canal de saída, ao qual a informação deve se dirigir.

Um circuito elementar que efetuar uma demultiplexação é visto na figura abaixo.



Neste circuito, se quisermos ligar a informação de entrada ao canal de saída I1, basta selecionarmos a posição 1 da chave seletora, surgindo informação somente na saída I1. Se quisermos que a informação de entrada seja conectada ao canal de saída I2, basta selecionarmos a posição 2 e assim sucessivamente.



## Referências

CAPUANO, IDOETA. Elementos Eletrônica Digital. 6 d.

ALEXANDER, Charles, SADIKU, Matthew. Fundamentos de Circuitos elétricos. 5ª Ed. Bookman

SHAMIEH, Cathleen. Eletrônica para Leigos. 3ªEd. Alta Books