

LIÇÃO 6

OS ELEMENTOS BIESTÁVEIS

Na lição anterior analisamos os modos segundo os quais podemos saber o que acontece quando combinamos funções lógicas. Vimos os procedimentos utilizados para implementar um circuito a partir de uma tabela verdade ou ainda da expressão da função lógica. No entanto, as funções lógicas não consistem nos únicos blocos básicos usados nos projetos de circuitos digitais. Além dessas funções, existem outras e um grupo delas que executa funções de relevante importância nos equipamentos são as formadas pelos elementos biestáveis. Nesta lição veremos como funcionam estes elementos, os seus tipos e onde podem ser usados.

6.1 - OS FLIP-FLOPS

Os *flip-flops* são elementos de circuito que podem apresentar em seu funcionamento apenas dois estados estáveis. Não existem estados intermediários entre estes dois estados.

A aplicação de um sinal de entrada pode mudar o dispositivo de um estado para outro e como a qualquer momento podemos saber qual é o estado em que ele se encontra, é possível considerar este circuito como uma memória capaz de armazenar um bit.

O *flip-flop* é o elemento básico das chamadas memórias estáticas.

Existem diversos tipos de *flip-flops* encontrados nos circuitos digitais e que analisaremos a partir de agora.

6.2 - FLIP-FLOP R-S

O *Flip-Flop R-S* (de *Reset* e *Set*) tem sua configuração com transistores mostrada na **figura 1** e funciona da seguinte maneira:

Quando alimentamos o circuito, dada as mínimas diferenças que podem existir entre as características dos dois transistores, um deles conduzirá mais do que o outro. Supondo que este transistor seja Q_1 , há uma queda de tensão no seu coletor que reduz em consequência a corrente que polariza a base de Q_2 via R_2 .

Nestas condições, a tensão do coletor de Q_2 se mantém alta, realimentando a base de Q_1 via R_3 e a situação final do circuito é estabelecida: Q_1 satura e Q_2 fica no corte. O *flip-flop* encontra seu estado estável inicial.

O *flip-flop R-S* tem duas saídas representadas por Q e \bar{Q} , assim, na condição inicial estável, com Q_1

conduzindo, Q estará no nível baixo (0) e \bar{Q} estará no nível alto (1).

O processo que leva o *flip-flop* a este estado inicial pronto para funcionar é muito rápido, não demorando mais do que alguns microssegundos.

Quando o *flip-flop* se encontra na situação indicada, com $Q=0$ e $\bar{Q}=1$, dizemos que ele se encontra "setado" ou armado.

A mudança de estado do *flip-flop* pode ser obtida aplicando-se um sinal conveniente na entrada. Como usamos transistores NPN para comutar o *flip-flop*, temos de fazer conduzir por um instante o transistor que está cortado, ou seja, devemos aplicar um pulso positivo na entrada correspondente.

Assim, estando o *flip-flop* na condição indicada, se desejarmos mudar o estado, aplicamos o pulso na entrada SET. O transistor Q_2 conduz por um instante, realimentando via R_3 a base de Q_1 que é cortado.

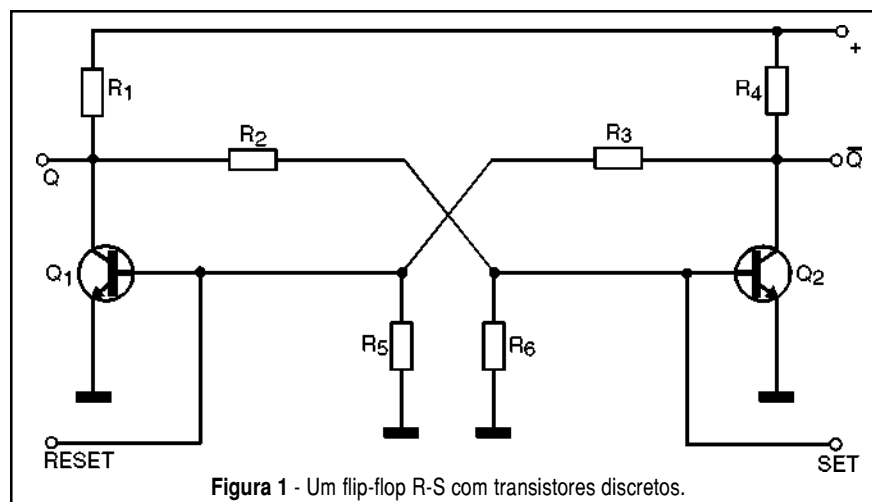
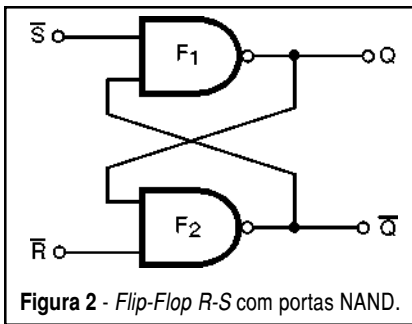


Figura 1 - Um flip-flop R-S com transistores discretos.



Com o corte, a tensão na base de Q_2 sobe via polarização de R_2 e mesmo que o pulso de disparo desapareça, o circuito se mantém no novo estado graças à realimentação.

Sua saída Q vai ao nível (1) e a saída \bar{Q} vai ao nível (0).

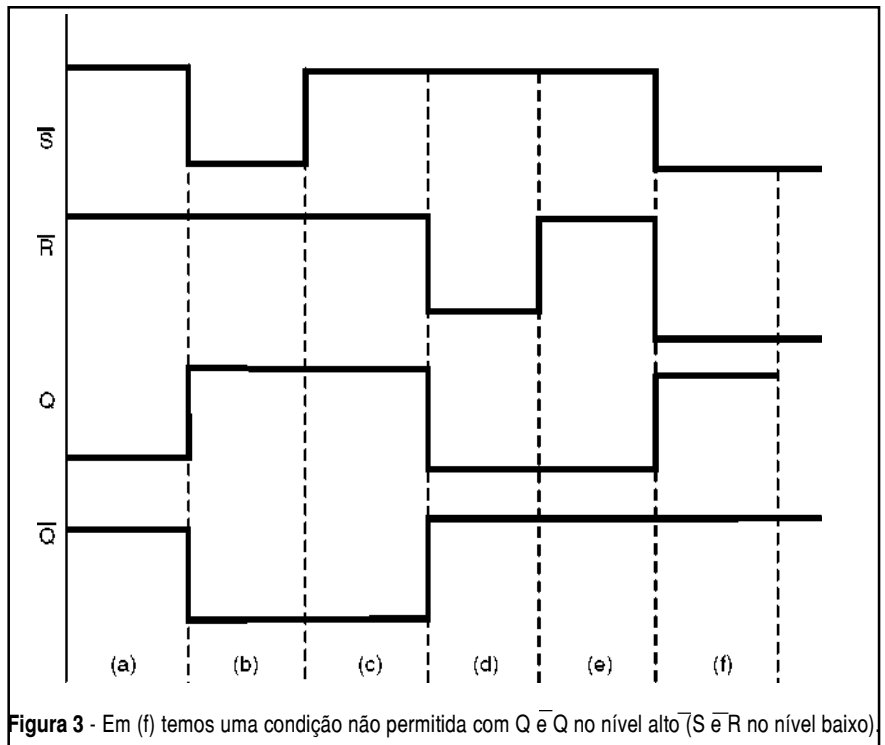
Para trocar novamente de estado o *flip-flop* R-S, aplicamos um pulso positivo na entrada RESET, levando Q_1 à saturação e Q_2 ao corte, situação que se firma mesmo depois de desaparecido o pulso graças à realimentação proporcionada pelos resistores.

Veja que um pulso aplicado à entrada SET, o que corresponde a um bit 1, faz com que a saída Q que estava em 0 passe a 1, armazenando este bit. O *flip-flop* funciona realmente como uma memória para este bit.

Da mesma forma como utilizamos transistores bipolares NPN para obter um *flip-flop*, podemos também empregar outros tipos de componentes em configurações semelhantes. Podemos, por exemplo, elaborar *flip-flops* usando transistores PNP, caso em que a polaridade dos sinais de disparo vai ser invertida.

Da mesma forma, podemos usar transistores de efeito de campo, tanto de canal N como canal P (bipolares ou JFETs) como também transistores de efeito de campo MOS com os dois tipos de canal (N ou P). O que mudará em cada caso é o sentido de circulação das correntes e as polaridades dos sinais aplicados.

Conforme veremos na última parte desta lição, os *flip-flops* também podem ser feitos com válvulas e na realidade os primeiros que existiram eram justamente montados com estes componentes. Naquela época não existiam transistores e nem circuitos integrados.



Os *flip-flops* podem ser elaborados com portas lógicas e o R-S que estudamos pode ser facilmente obtido a partir de duas portas NAND de duas entradas, **figura 2**.

Levando em conta as tabelas verdade das portas NAND veremos que a saída da primeira porta realimenta a segunda e vice-versa, garantindo assim a manutenção dos estados obtidos quando o *flip-flop* comuta.

No entanto, a comutação deste circuito ocorre quando as entradas passam do nível alto para o baixo, ou seja, de 1 para 0. Esta condição é indicada pelos símbolos \bar{R} e \bar{S} na entradas.

O leitor pode então perceber que, quando as entradas estão ambas no nível baixo, o *flip-flop* se mantém no estado em que foi colocado por ser ligado ou por uma comutação anterior.

Por outro lado, se as entradas forem levadas simultaneamente ao nível alto, o *flip-flop* irá para um estado indeterminado que deve ser evitado. Na prática, a aplicação de níveis altos (1) nas duas entradas pode destruir o dispositivo.

O diagrama de tempos da **figura 3** mostra o que ocorre no funcionamento de um *flip-flop* por etapas que podemos analisar da seguinte forma:

- a) *Flip-flop* ressetado
- b) \bar{S} vai ao nível baixo e o *flip-flop* é setado
- c) \bar{S} vai ao nível alto e o *flip-flop* permanece setado
- d) \bar{R} vai ao nível baixo e o *flip-flop* é ressetado
- e) \bar{R} volta ao nível alto e o *flip-flop* permanece ressetado

Tudo isso pode ser representado por uma tabela verdade, da mesma forma que fazemos com as funções lógicas. Nesta tabela temos alguns novos símbolos com os quais o leitor deve começar a familiarizar-se e que são amplamente usados em Eletrônica Digital, a saber:

a) Primeira possibilidade

Q_{n-1} = representa o estado da saída Q ANTES da aplicação dos sinais.

Q_n = representa o estado da saída Q DEPOIS da aplicação dos sinais.

b) Segunda possibilidade

Q = representa o estado da saída Q ANTES da aplicação dos sinais.

Q_{n+1} = representa o estado da saída Q DEPOIS da aplicação dos sinais.

Obs: em lugar de n em alguns livros encontramos a letra t .

Os dois tipos de representação são usados.

Nas colunas e linhas em que são colocados os níveis lógicos 0 e 1, quando aparece o termo Q_n ou \bar{Q}_n significa que a saída vai para um estado indeterminado.

A tabela verdade do *flip-flop R-S* com portas NAND fica então:

R	S	Q_{n+1}	\bar{Q}_{n+1}
0	0	Q_n	\bar{Q}_n
0	1	1	0
1	0	0	1
1	1	0	0

Para obtermos um *flip-flop R-S* também podemos usar portas NOR, conforme a **figura 4**.

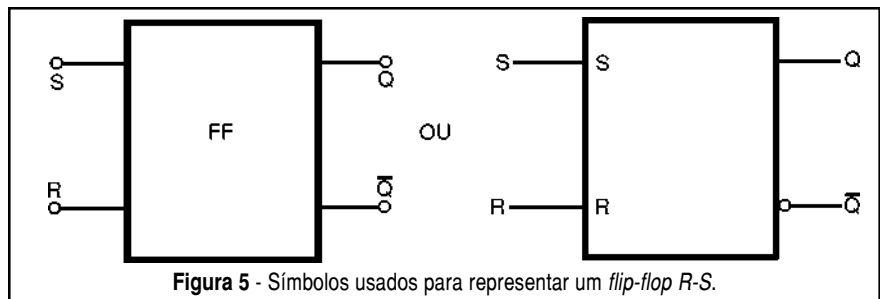
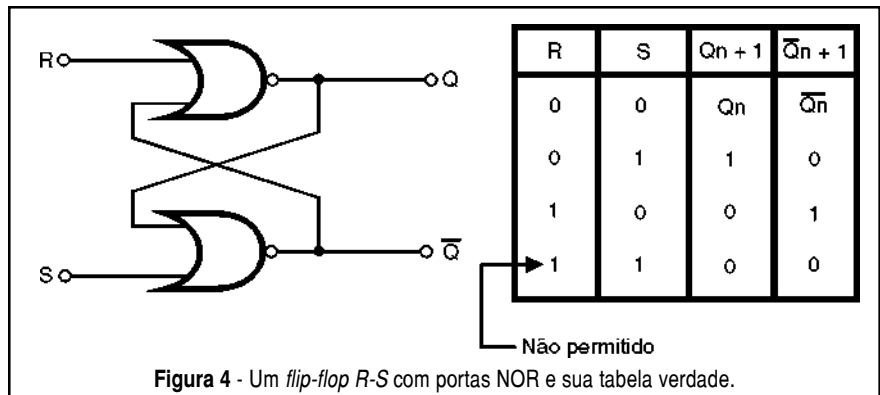
Na **figura 5** temos os símbolos adotados para representar este tipo de *flip-flop*.

Este circuito também é chamado de *R-S NOR LATCH* da mesma forma que o circuito anterior é denominado *R-S NAND LATCH*.

6.3 - FLIP-FLOP RS COM CLOCK E MESTRE-ESCRAVO

Estes circuitos chamados de *flip-flop R-S* controlados por *clock* e mestre escravo encontram uma gama de aplicações muito grande nos circuitos digitais mais complexos, já que estes são sempre comandados por um *clock*, ou seja, são circuitos lógicos sincronizados.

O uso de um circuito de controle (mestre) que determina quando o *flip-flop* (escravo) muda de estado é importante para permitir que as mudanças de estado do *flip-flop* só ocorram em determinados instantes.



Usando portas NAND podemos inicialmente implementar um *flip-flop R-S* controlado por *clock* (*Master-Slave*), conforme a figura 6.

Analisemos seu funcionamento:

Partindo da situação em que a entrada de *clock* (relógio) esteja no nível baixo, as saídas Q e \bar{Q} permanecerão no estado inicial em que se encontravam e insensíveis a qualquer variação que ocorra nas entradas S e R.

Quando a entrada de *clock* for levada ao nível 1, o circuito passa a responder aos sinais das entradas R e S.

No entanto, conforme o diagrama de tempos da **figura 7**, este circuito tem um inconveniente.

Como as saídas acompanham as entradas, durante o tempo em que o *clock* as habilita, estas saídas podem mudar de estado mais de uma vez,

voltando assim ao estado inicial, o que não é desejado de forma alguma.

Um modo de contornar este problema consiste na utilização de duas etapas numa configuração mais complexa, que é apresentada na **figura 8**.

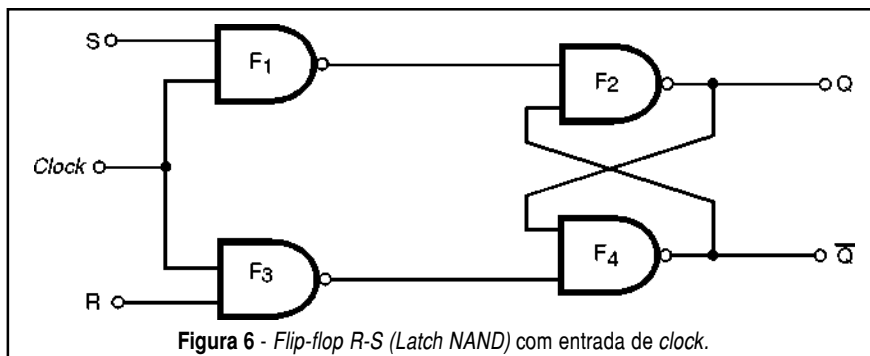
Este circuito é denominado *Flip-Flop R-S Mestre-Escravo* ou *Flip-Flop R-S Master-Slave* e faz uso de portas NAND e de um inversor, cuja finalidade é inverter o pulso de *clock*.

Neste caso, quando a entrada de *clock* for ao nível 1, o *flip-flop* mestre mudará de estado, mas o *flip-flop* escravo permanecerá insensível, mantendo seu estado.

Quando a entrada de *clock* passar para o nível lógico 0, a saída do *flip-flop* mestre será levada para o escravo.

Isso significa que o *flip-flop* em seu todo não é sensível ao nível do sinal de *clock*, ou seja, se ele é 0 ou 1, mas sim à sua transição. As saídas Q e \bar{Q} só vão mudar de estado no instante em que ocorrer a transição do sinal de *clock* do nível alto para o nível baixo. Com esta configuração é possível garantir que só vai ocorrer uma mudança de estado na presença de um pulso de *clock*.

Os *flip-flops* que funcionam desta forma são denominados "*Edge Triggered*" ou "*Disparados pela Bor-*



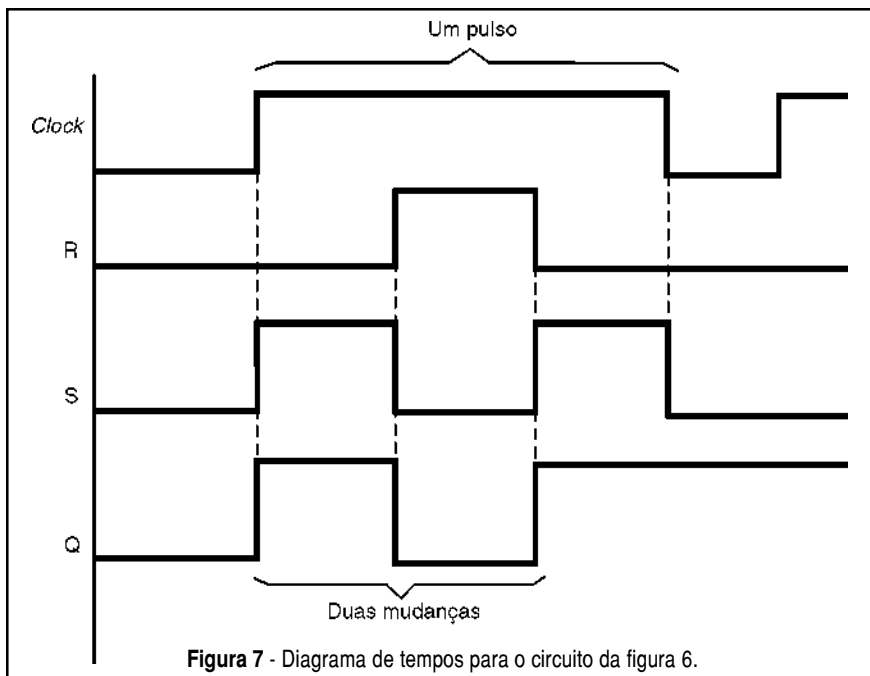


Figura 7 - Diagrama de tempos para o circuito da figura 6.

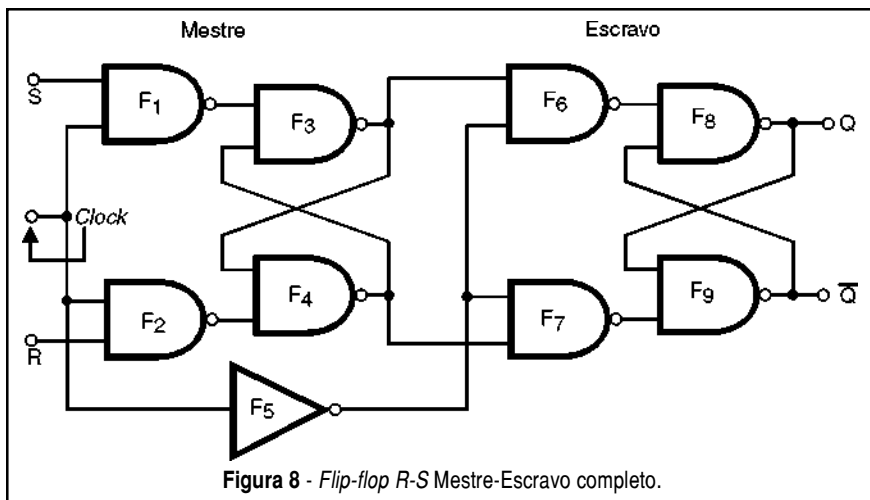


Figura 8 - Flip-flop R-S Mestre-Escravo completo.

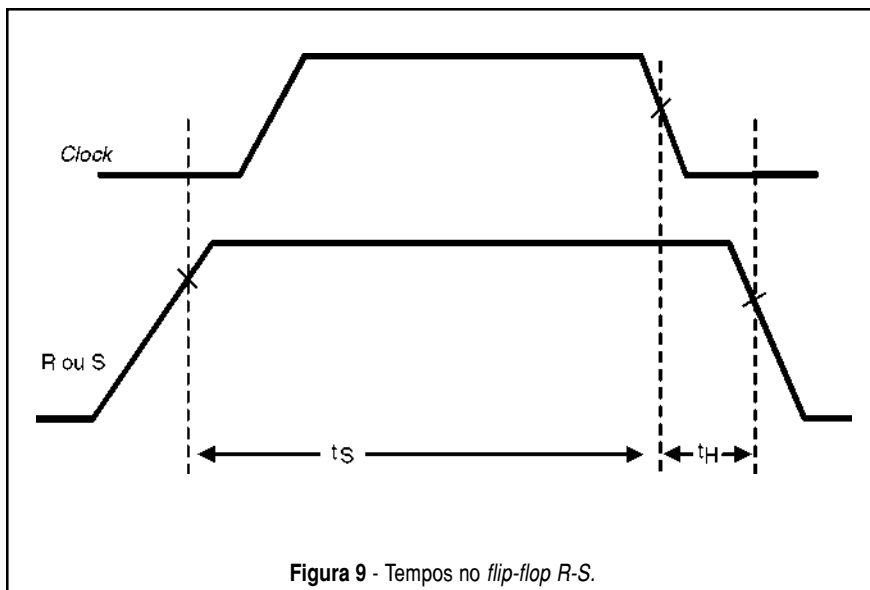


Figura 9 - Tempos no flip-flop R-S.

da”.

Se a mudança de estado ou disparo (gatilhamento) ocorre quando o sinal de *clock* passa de 0 para 1, os *flip-flops* são denominados “*positive edge-triggered*”, enquanto que, se o disparo ocorre quando o *clock* vai do nível 1 para 0, ou seja, na queda do nível lógico, os *flip-flops* são chamados de “*negative edge-triggered*”.

Neste tipo de circuito é muito importante levar em conta, num projeto de maior velocidade, os tempos em que todo o processo ocorre.

Assim, partindo do diagrama de tempos da figura 9, vemos que a saída do *flip-flop* só completa sua mudança de estado depois de um certo tempo, do pulso de *clock* ter sido aplicado.

Dois tempos são importantes neste tipo de circuito.

a) **tH: Hold Time ou Tempo de Manutenção** é o tempo em que a entrada deve permanecer ainda no circuito para que seu nível lógico seja reconhecido pelo *flip-flop*.

b) **tS: Setup Time** ou tempo em que a entrada do *flip-flop* deve permanecer no estado desejado antes da transição do *clock* que vai provocar a mudança de estado do circuito.

Dois entradas podem ser acrescentadas neste circuito, verifique a figura 10, dotando-o de recursos importantes para aplicações práticas.

Uma das entradas é denominada *PRESET* (/PR) ou pré-ajuste e tem por função levar imediatamente as saídas do circuito a um estado determinado ($Q=1$ e $/Q=0$), independentemente do que esteja acontecendo nas demais entradas.

Sua ativação ocorre quando /PR estiver em 0 e /CLR em 1, no caso apresentado, pois a / sobre a identificação indica que ela está ativa no nível baixo.

A outra entrada denominada *CLEAR* ou apagamento tem por função levar as saídas aos estados $Q=0$ e $/Q=1$, independentemente do que estiver ocorrendo nas demais entradas.

É importante observar que estas duas entradas não podem ser ativadas ao mesmo tempo, pois isso levaria o circuito a um estado indeterminado que inclusive poderia

causar problemas aos seus componentes.

A tabela verdade para este circuito nos mostra três novos símbolos que normalmente são usados em Eletrônica Digital.

X representa uma condição irrelevante qualquer que ela seja, não haverá influência no que ocorre na saída.

A seta para cima indica a transição do nível baixo para o nível do sinal na entrada ou saída representadas, enquanto que a seta apontando para baixo indica uma transição do nível baixo para o nível alto do sinal correspondente.

6.4 - O FLIP-FLOP J-K MESTRE-ESCRAVO

O *flip-flop J-K* mestre-escravo ou “*master-slave*” pode ser implementado por funções lógicas comuns, adquirindo a configuração básica mostrada na **figura 11**.

Um problema que observamos nos *flip-flops R-S* é que temos uma situação “proibida” que ocorre quando as entradas R e S vão ao nível alto ao mesmo tempo e que pode levar o circuito a um estado indeterminado. Esta situação acontece principalmente nas aplicações em computação, quando uma parte do sinal de saída é usada para realimentar a entrada. Nestas condições podem ocorrer as situações de conflito com a produção de oscilações indesejadas.

Esta situação pode ser contornada com a utilização de uma nova configuração, que é justamente a do *flip-flop J-K* utilizada nas aplicações práticas e que analisaremos a seguir.

Podemos ter quatro combinações possíveis para os sinais aplicados nas entradas J e K, conforme observamos na tabela abaixo.

J	K
0	0
1	0
0	1
1	1

Analisemos cada uma das combinações:

a) J=0 e K=0

Quando a entrada de *clock (CLK)*

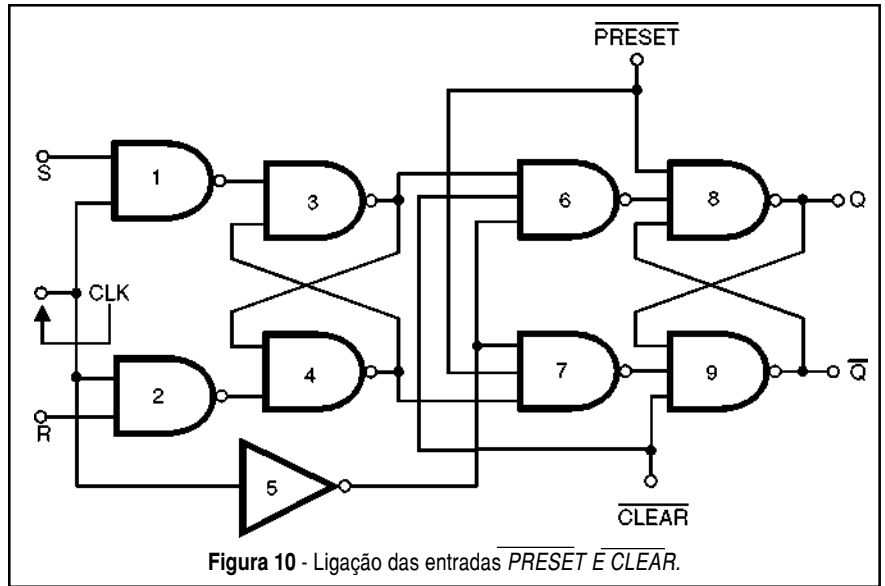


Figura 10 - Ligação das entradas \overline{PRESET} e \overline{CLEAR} .

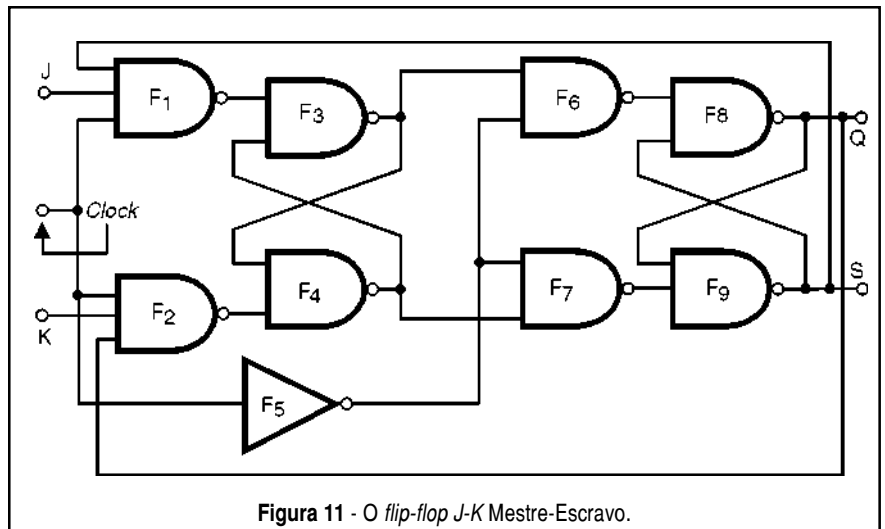


Figura 11 - O *flip-flop J-K* Mestre-Escravo.

passa por uma transição negativa do sinal, o *flip-flop* mantém sua condição original, ou seja, não muda de estado.

b) J=1 e K=0

Quando a entrada de *clock (CLK)* passa por uma transição negativa, o *flip-flop* é “setado”. Se já estiver setado, ele permanece nesta condição.

c) J=0 e K=1

Quando a entrada de *clock (CLK)* passa por uma transição negativa, o *flip-flop* é “ressetado”. Se já estiver nesta condição, ele permanece.

d) J=1 e K=1

Nesta condição, ao receber uma transição negativa na entrada de *clock*

(*CLK*), o *flip-flop* muda de estado (*TOGGLE*). Se estiver setado, ele resseta e se estiver ressetado, ele é setado.

Podemos elaborar a tabela verdade da **figura 12** para indicar o que ocorre com este *flip-flop*.

Observe o uso das setas para indicar as transições de sinal na entrada de *clock* que comandam o funcionamento deste tipo de circuito.

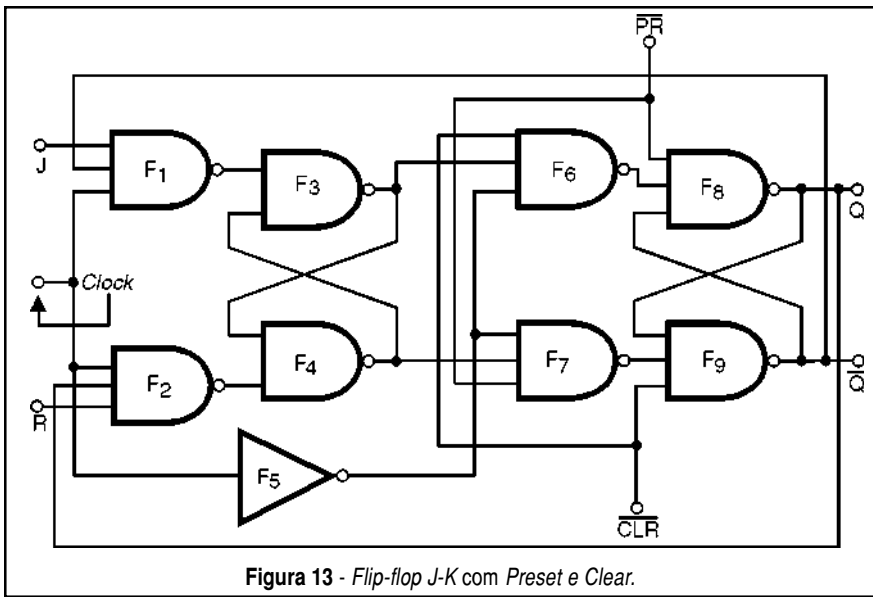
Da mesma forma que nas outras configurações estudadas, podemos também incluir as entradas de *PRESET* e *CLEAR* neste circuito que ficará da maneira apresentada na **figura 13**.

Uma tabela verdade incluindo as entradas de *PRESET (PR)* e *CLEAR (CLR)* é mostrada na **figura 14**.

Uma maneira melhor de analisar-

CLK	J	K	Q_{n+1}	\bar{Q}_{n+1}	
↓	∅	∅	Q_n	\bar{Q}_n	Não muda
↓	∅	1	∅	1	Reseta
↓	1	∅	1	∅	Seta
↓	1	1	\bar{Q}_n	Q_n	Complementa (Toggle)

Figura 12 - Tabela verdade para o flip-flop J-K Mestre-Escravo.



\bar{CLR}	\bar{PR}	J	K	CLK	Q_{n+1}	\bar{Q}_{n+1}
∅	∅	X	X	X	1	1
∅	1	X	X	X	∅	1
1	∅	X	X	X	1	∅
1	1	∅	∅	↓	Q_n	\bar{Q}_n
1	1	∅	1	↓	∅	1
1	1	1	∅	↓	1	∅
1	1	1	1	↓	\bar{Q}_n	Q_n

Figura 14 - Tabela verdade para o flip-flop J-K com Preset e Clear.

mos o funcionamento deste circuito é através de um diagrama de tempos, em que observamos as formas de onda nos diversos pontos de entrada e saída. Este diagrama de tempos para o flip-flop J-K é mostrado na figura 15.

Analisemos alguns trechos importantes deste diagrama mostrando o que acontece:

a) Neste instante CLR e PR estão no nível baixo, Q e /Q estão no nível alto, que é uma condição não permitida.

b) Aplica-se então o sinal PR, que indo ao nível alto, faz com que o flip-flop seja ressetado.

c) A aplicação de um pulso na entrada CLR que vai ao nível alto, e a ida de PR ao nível baixo fazem agora com que o flip-flop seja setado.

d) CLR e PR são mantidos no nível alto a partir deste instante. Com J=0 neste trecho e K indo ao nível alto, o flip-flop será ressetado na próxima transição negativa do sinal de clock.

e) Ainda com CLR e PR no nível alto (esta condição se manterá daqui por diante) e a saída J=0 e k=1, o flip-flop permanecerá ressetado.

f) Com J=1 e K=0, o flip-flop é setado na transição seguinte do pulso de clock.

g) Com J=1 e K=0, não ocorrem mudanças de estado.

h) Com J=1 e K=1 na transição seguinte do pulso de clock, o flip-flop muda de estado (complementa ou "toggle"). Se estiver ressetado, como neste caso, ele é setado.

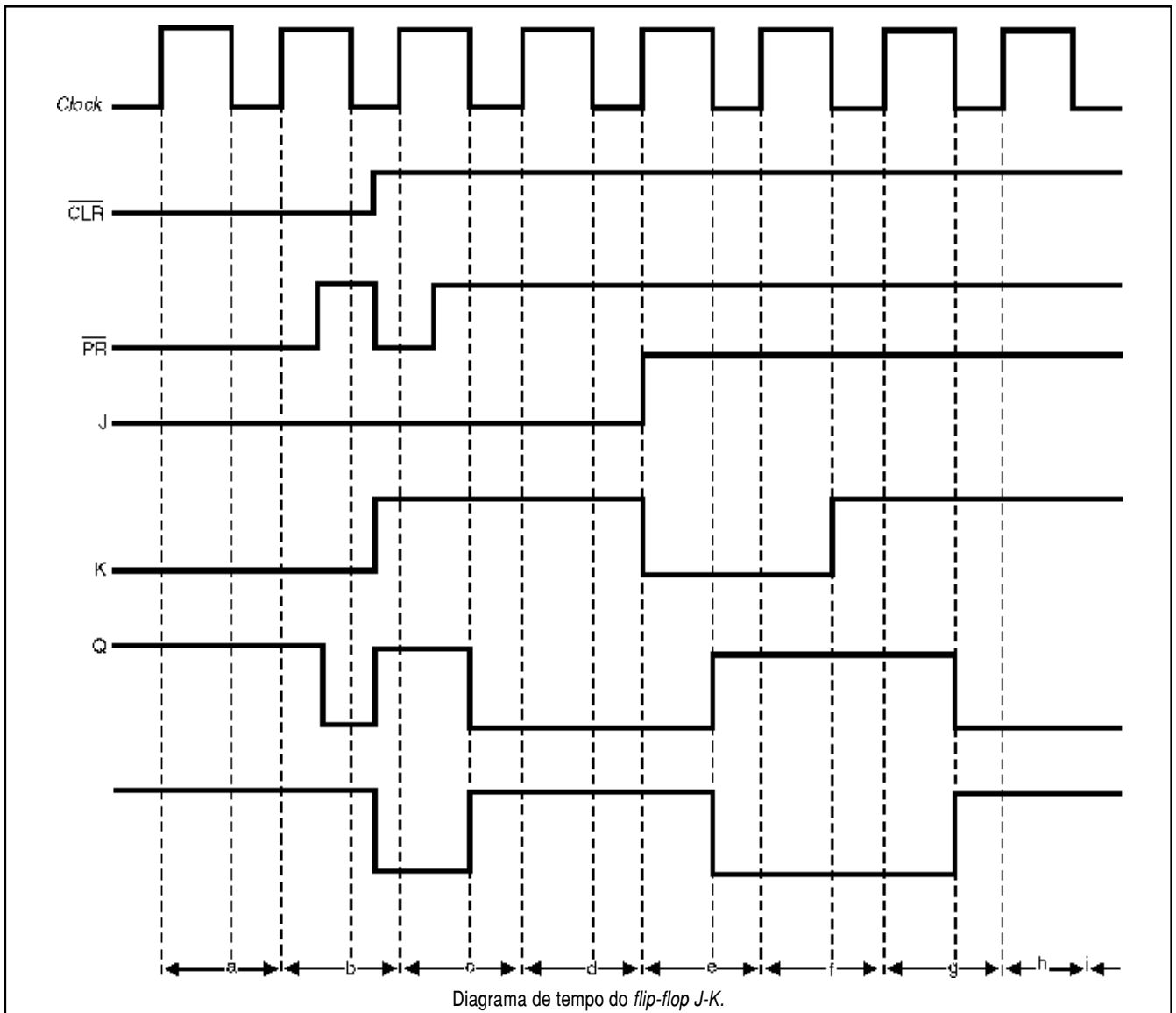
i) Mantendo J=1 e K=1 com nova transição do pulso de clock, o flip-flop muda de estado outra vez, ou seja, complementa.

Veja que quando as entradas J e K estão no nível alto, o circuito se comporta como um disparador, mudando de estado a cada transição negativa do pulso de clock.

6.5 - O FLIP-FLOP TIPO D

Este é também um circuito de flip-flop muito usado, cujo símbolo é mostrado na figura 16.

Este flip-flop possui uma única entrada que comanda todo o circuito. Esta entrada é que lhe dá nome. De-



nominada "Data" (dados), é abreviada por D, daí o nome do dispositivo.

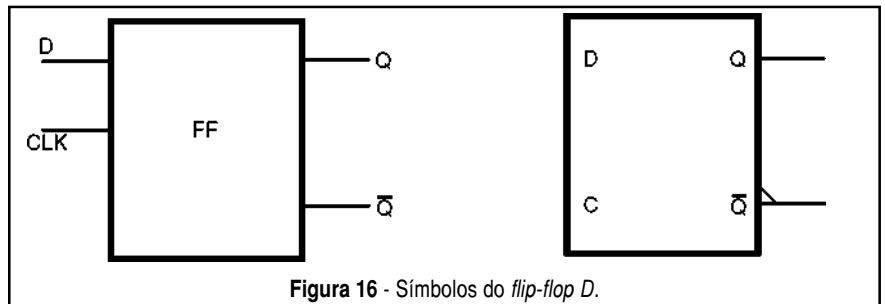
Este *flip-flop* opera de uma maneira muito simples: no pulso de *clock*, ele assume o estado da entrada, conforme podemos ver pela sua tabela verdade:

D	Q _{n+1}
0	0
1	1

6.6 - FLIP-FLOP TIPO T

O nome vem de "Toggle" ou complementação, seu símbolo é mostrado na **figura 17**.

O que este circuito faz pode ser entendido facilmente pelo diagrama



de tempos mostrado na **figura 18**. Quando a entrada T deste circuito está no nível baixo, o *flip-flop* se mantém em seu estado anterior, mesmo com a aplicação do pulso de *clock*. No entanto, quando a entrada T está no nível alto, o *flip-flop* muda de estado. Se estava setado, ele resseta e se estava ressetado, ele seta.

Este comportamento significa na realidade a divisão da frequência de *clock* por dois. Em outras palavras, este circuito se comporta como um divisor de frequência, encontrando aplicações práticas bastante importantes em Eletrônica Digital.

Um exemplo de aplicação é dado na **figura 19** em que associamos di-

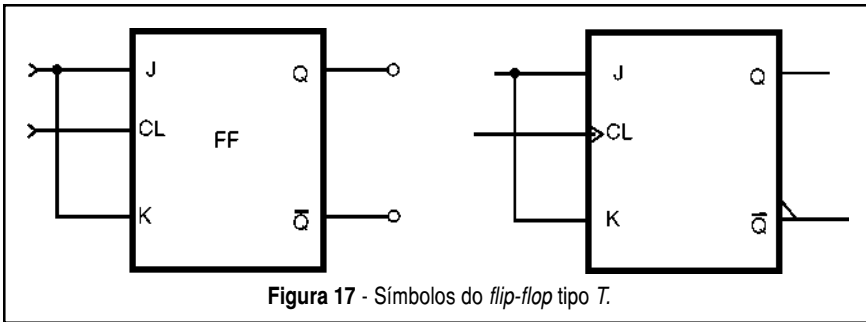


Figura 17 - Símbolos do *flip-flop* tipo T.

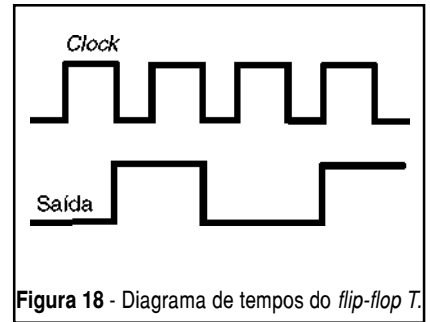


Figura 18 - Diagrama de tempos do *flip-flop* T.

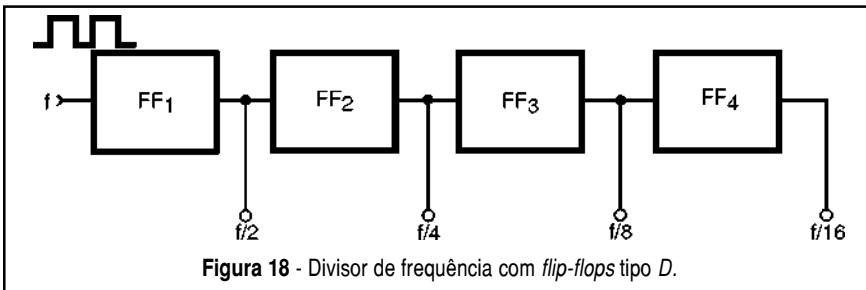


Figura 18 - Divisor de frequência com *flip-flops* tipo D.

6.7 - TRANSFORMANDO FLIP-FLOPS

Da mesma maneira como podemos obter qualquer função lógica complexa a partir de funções simples, o que foi visto em lições anteriores, também podemos “brincar” com os *flip-flops*, obtendo outros tipos a partir de um tipo básico.

Assim, usando um *flip-flops* R-S ou J-K que são comuns e algumas portas lógicas, podemos obter *flip-flops* de outros tipos.

Na **figura 20** temos algumas con-

versos *flip-flops* do tipo T em série, de modo que passando através de cada um, a frequência do sinal de entrada é dividida por 2.

Usando 4 *flip-flops*, podemos dividir a frequência por 2, 4, 8 e 16.

Este tipo de divisor de frequência

é muito usado, existindo até circuitos integrados que possuem sequências de mais de 10 *flip-flops* ligados desta forma.

Na prática não temos os *flip-flops* tipo D como componentes prontos para uso. Estes *flip-flops* podem ser obtidos a partir de outros e isso será visto no item seguinte.

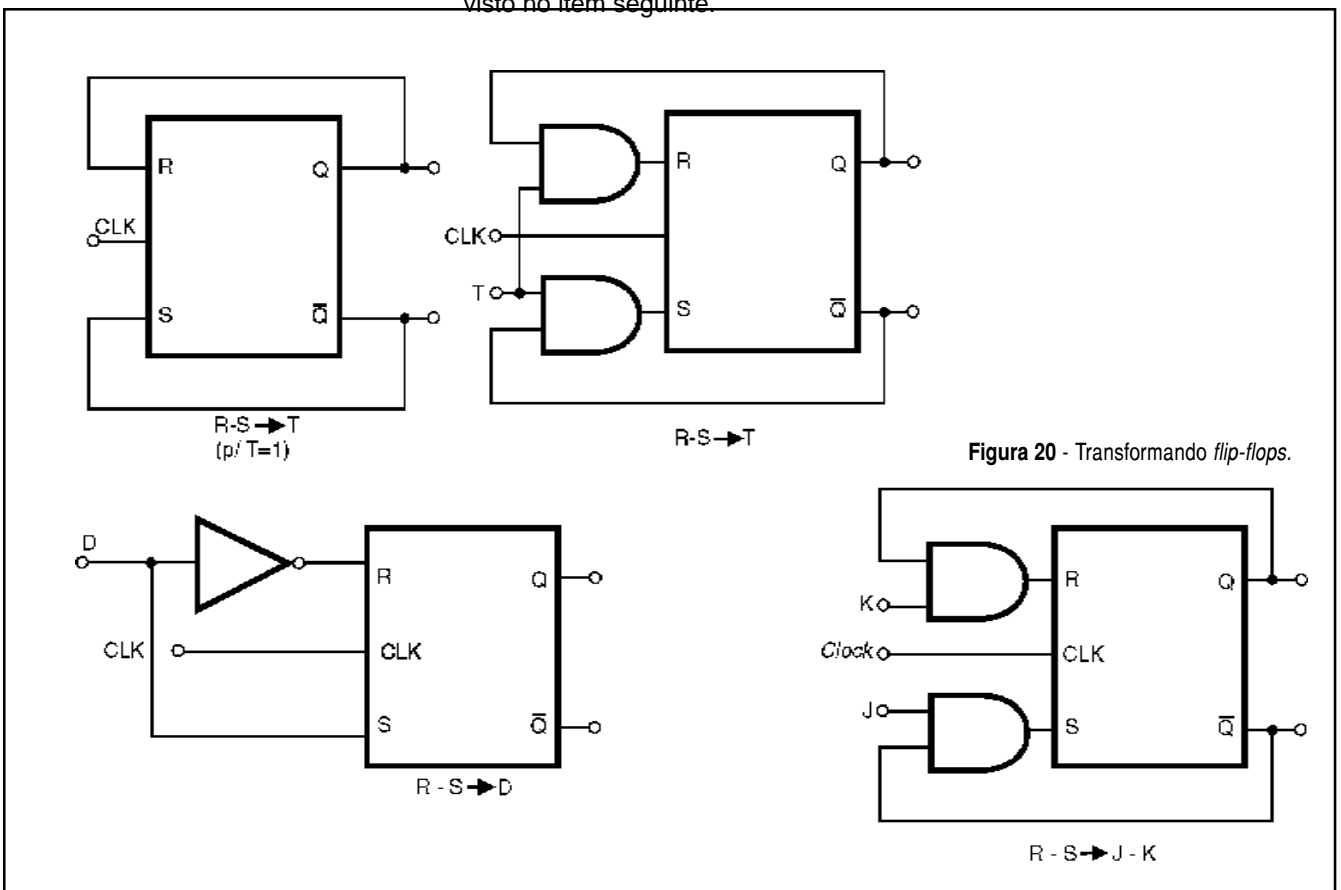
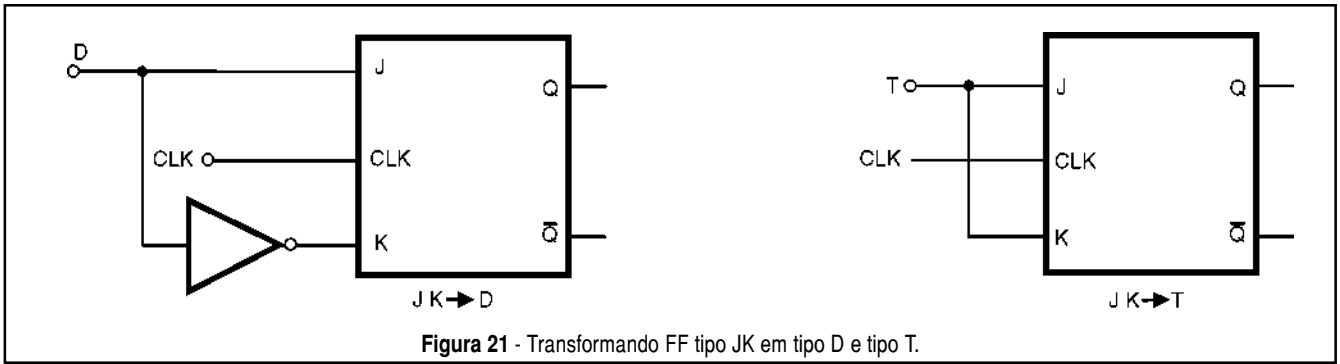


Figura 20 - Transformando *flip-flops*.



versões que podem ser feitas utilizando-se *flip-flops* do tipo R-S.

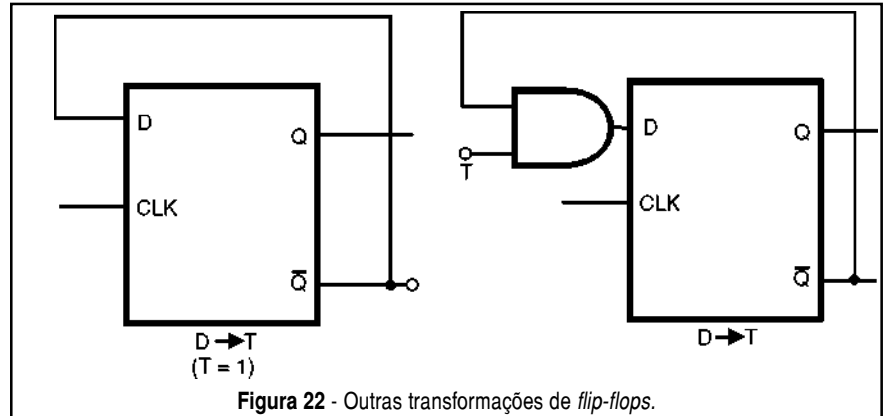
O modo de funcionamento de cada um pode ser facilmente entendido se o leitor tentar associar as tabelas verdade dos *flip-flops* que foram estudados nesta lição às tabelas verdade das portas agregadas, considerando os sinais de realimentação.

Na **figura 21** temos o modo de obter *flip-flops* tipo D e T a partir de *flip-flops* do tipo J-K.

Veja que a simples conexão da entrada K ao J no *flip-flop* do tipo J-K o transforma em um *flip-flop* tipo T. Esta possibilidade é muito interessante, já que *flip-flops* J-K são disponíveis em tecnologia TTL e CMOS e podem ser usados em circuitos divisores de frequência. Na verdade, já utilizamos esta configuração em diversos projetos práticos que publicamos. Finalmente, temos outras duas transformações importantes de *flip-flops* mostradas na **figura 22**.

No primeiro caso temos uma transformação de um *flip-flop* tipo D em *flip-flop* tipo T, bastando para isso que a saída complementar /Q seja ligada à entrada D, realimentando o circuito.

A segunda transformação, que leva um *flip-flop* tipo D a funcionar como tipo T, exige o emprego de uma porta AND adicional na realimentação



do sinal que é retirado da saída complementar /Q.

6.8 - NOS COMPUTADORES

Encontramos os *flip-flops* nos computadores como elementos fundamentais de muitos circuitos.

Uma aplicação é na própria divisão de frequência dos *clocks*. Conforme o leitor sabe, existem setores de um PC que devem operar com velocidades menores que a fornecida pelo *clock* principal. É o caso dos barramentos onde são ligadas as placas de expansão, os modems e as saídas de dados paralela e serial.

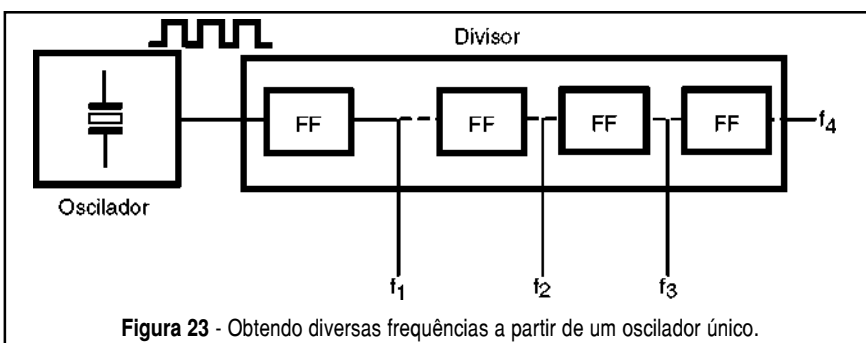
Assim, em lugar de usar um *clock* para cada frequência desejada, o que se faz é empregar um *clock* único e dividir sua frequência conforme as

exigências de frequências mais baixas, observe a **figura 23**.

No caso dos computadores, tanto o próprio *clock* como a sequência de *flip-flops* divisores podem ser obtidos num único circuito integrado.

Um ponto importante que deve ser levado em conta e que estudaremos nas lições futuras é a possibilidade de ligar os *flip-flops* em conjunto com outras funções, de modo que a frequência possa ser dividida por qualquer número e não somente por potências de (2,4,8,16,32,64, etc).

Outra aplicação importante é como célula de memória. Oito *flip-flops* ligados lado a lado podem armazenar um *byte* inteiro. Cada *flip-flop* armazena um bit. Existem diversas memórias internas de um PC que nada mais são do que *flip-flops* que podem ser habilitados tanto para a leitura de dados como para introdução (gravação de dados). Existem ainda outras funções importantes implementadas a partir de *flip-flops* e que serão estudadas futuramente.



6.9. OS FLIP-FLOPS ANTIGOS

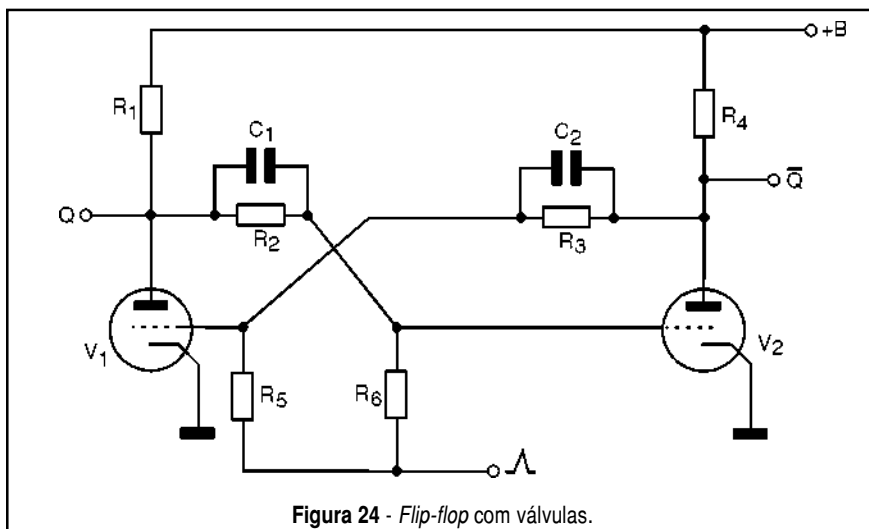


Figura 24 - Flip-flop com válvulas.

A configuração do *flip-flop* não é nova. Na verdade, foi em 1919 que dois pesquisadores americanos chamados **Eccles** e **Jordan** apresentaram o primeiro circuito de *flip-flop* usando válvulas, confira na figura 24.

Por este motivo, muitos ainda chamam os *flip-flops* de “Básculas ou Circuitos Eccles-Jordan”.

Em 1930, os físicos já usavam estes circuitos ligados em série para dividir a contagem dos pulsos de contadores Geiger de radiação e obterem valores menores mais facilmente totalizados nas pesquisas.

Um contador binário usando uma lâmpada neon ligada às válvulas foi desenvolvido usando estes *flip-flops* em 1940, mas foi somente depois disso que os primeiros computadores digitais passaram a usar estes circuitos de uma forma mais intensa, até o advento do transistor e depois dos circuitos integrados.

QUESTIONÁRIO

1. Os elementos biestáveis de um circuito:

- Possuem apenas um estado estável
- Possuem dois estados instáveis
- Possuem dois estados estáveis
- Possuem um número indeterminado de estado estáveis

2. Usado como elemento de memória, um *flip-flop* pode armazenar:

- 1 bit
- 1 byte
- meio byte
- 2 bits

3. Um *flip-flop R-S* “setado” apresenta que níveis lógicos em suas saídas:

- $Q=0$ e $\bar{Q}=0$
- $Q=0$ e $\bar{Q}=1$
- $Q=1$ e $\bar{Q}=0$
- $Q=1$ e $\bar{Q}=1$

4. Os *flip-flops* “negative edge-triggered” mudam de estado quando:

- O pulso *clock* vai do nível baixo para o nível alto
- O pulso de *clock* vai do nível alto para o nível baixo
- O pulso de *clock* estabiliza no nível baixo
- O pulso de *clock* estabiliza no nível alto

5. Para que um *flip-flop J-K* Mestre escravo tenha a condição “toggle”, quais são os níveis lógicos que devem ser colocados na entrada J e K?

- $J=0$ e $K=0$
- $J=0$ e $K=1$
- $J=1$ e $K=0$
- $J=1$ e $K=1$

6. Quatro *flip-flops* do tipo T ligados um após o outro (em série) recebem uma frequência de 1 600 Hz na sua entrada. Qual é a frequência obtida na saída do último *flip-flop*? (o sinal deve ser retangular).

- 800 Hz
- 400 Hz
- 200 Hz
- 100 Hz

Respostas:

1-C, 2-A, 3-C, 4-B, 5-D, 6-D