

Definição 4.2 Um estado (totalmente) estável é uma combinação de estados internos e entradas, tal que o próximo estado interno previsível pela tabela de estados é o mesmo do estado interno corrente. Caso contrário, dizemos que o estado é instável.

Exemplo 4.1 Considere um sinal de alarme $s(\text{set})$ que vai para 1 quando houver uma situação de violação, voltando em seguida para 0. Denominando “violação” de estado S_1 e “não-violação” de estado S_0 , temos a seguinte tabela de estados

Q	s	
	0	1
S_0	S_0 (*)	S_1
S_1	S_1 (*)	S_1 (*)

Q^*

Os estados internos marcados com (*) são os estados estáveis. Veremos mais adiante que para a combinação do estado (interno) S_0 e a entrada 1 tem-se como o próximo estado interno S_1 , que através da recombinação e junto como a entrada ainda em 1, leva o circuito para o estado (estável) S_1 .

Vimos que os componentes básicos de um circuito combinacional são as portas lógicas. Os componentes básicos de um circuito sequencial, por sua vez, são os latches e flip-flops, que tem a capacidade de reter/armazenar bits (acrônimo de *binary digit*). Como estes componentes podem armazenar dois estados estáveis — 0 e 1, eles são também conhecidos como **multiplicadores bistáveis**.

Definição 4.3 Latches são dispositivos digitais capazes de armazenar/reter um valor, que varia continuamente em função das entradas em cada instante.

Definição 4.4 Flip-flops são dispositivos que mostram as suas entradas continuamente, mas as suas saídas só mudam nos instantes determinados por um sinal de relógio/gatilho (*trigger*).

Estruturamos a lógica destes component básicos com uso de **tabelas de transição**. Estas tabelas são obtidas diretamente das tabelas de estados, substituindo os estados pelas suas representações binárias.

Exemplo 4.2 No exemplo 4.1 temos dois estados S_0 e S_1 . Associando aos dois estados a representação binária 0 e 1, respectivamente, temos a seguinte tabela de transição:

Capítulo 4

Circuitos de Memória: Latches e Flip-Flops

As saídas de um circuito sequencial a cada instante dependem não só das entradas como também do histórico passado do circuito até aquele instante e. O estado corrente do circuito reflete este histórico.

Definição 4.1 Um estado (total) é uma combinação particular de estados internos (valores armazenados nos laços de recombinação) e estados de entrada (valores correntes das entradas).

Nos circuitos digitais os valores das variáveis de estado são binários (0 ou 1). Assim, um circuito com n variáveis de estado pode conter 2^n possíveis estados. Vale, entretanto, ressaltar aqui que por maior que seja o valor n , o número possível de estados que o circuito pode assumir é finito. Por isso, circuito sequenciais são também conhecidos como **máquinas de estados finitos**.

Vimos que as tabelas-verdade são suficientes para descrever a funcionalidade de um circuito combinacional. O comportamento de um circuito com memória (ou sequencial), por sua vez, pode ser representado através de uma **tabela de estados**, na qual o próximo estado (interno) e a saída aparecem como função da entrada e do estado corrente.

Entradas
Estados Correntes Próximos Estados, Saídas

Q	s
	0 1
0	0 1
1	1 1
	Q^*

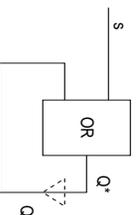
A partir da tabela de transição do dispositivo podemos derivar a sua equação característica ou equação de transição, que relaciona o próximo estado S^+ de cada componente e com o seu estado corrente S e as suas entradas I a cada instante

$$S^+ = F(S, I)$$

Exemplo 4.3 A partir da tabela de transição do exemplo 4.2 podemos, por inspeção ou com uso de mapa de Karnaugh, chegar à seguinte expressão de transição:

$$Q^* = Q + s.$$

Com esta expressão é fácil obter um diagrama lógico do circuito de interesse.



cujo comportamento lógico pode ser descrito pela seguinte tabela de funções:

s	Q^*
0	Q
1	1

Exemplo 4.4 Podemos ainda incluir uma entrada para resetar o estado do circuito no exemplo 4.1. Sendo rs as duas entradas e considerando a combinação de entradas $rs=11$ inválida, a tabela de estados do novo circuito será

Q	rs
	00 01 11 10
S_0	S_0 S_1 X S_0
S_1	S_1 X S_0 S_0
	Q^*

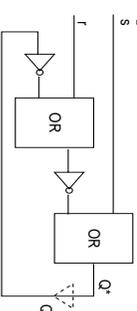
Como são dois estados, podemos usar a mesma representação binária que adotamos no exemplo 4.2 e construir a seguinte tabela de transição:

Q	rs
	00 01 11 10
0	0 1 X 0
1	1 1 X 0
	Q^*

A partir dela deriva-se a seguinte expressão algébrica:

$$Q^* = s + r'Q = s + (r'Q) = s + (r' + Q)Q \quad (4.11)$$

cujos diagrama lógico é



e cuja tabela de função é:

s	r	Q^*
0	0	Q
0	1	0
1	0	1
1	1	X

Devido ao comportamento analógico nas transições, as entradas dos circuitos digitais nunca mudam de valores **simultaneamente**, transições entre dois estados estáveis sempre ocorrem incrementalmente, podendo passar por uma sequência de estados internos instáveis na qual entre cada par de estado somente há variação de uma variável de entrada. Se o estado estável final alcançado depende da ordem da variação gradual das entradas, dizemos que há uma **corrida crítica** (critical race) (para o estado estável). A representação gráfica das transições internas é conhecida como **tabela de fluxo**.

Exemplo 4.5 Considere o circuito do exemplo 4.4, note que para o estado $Q=S_0$, uma combinação de entradas $rs=01$ leva o circuito para S_1 , e se a entrada mantiver em $rs=01$, com a redefinição o circuito se estabilizará logo em S_1 . Para o estado $Q=S_1$, a combinação de entradas $rs=10$ leva o circuito momentaneamente para S_0 , que com a redefinição chegará ao estado estável S_0 .

Q	rs			
	00	01	11	10
S0	00	S1	X	S0
S1	S1	S1	X	S0

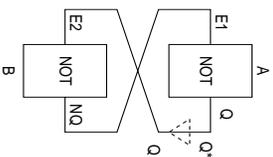
Q^*

Veremos ainda que alguns dos *latches* ou *flip-flops* podem ser construídos a partir dos outros. Nestes casos, pode-se derivar a partir da equação característica dos dispositivos-componente e da tabela de transição do circuito em projeto a **excitação** necessária. Uma forma de representação gráfica para tais excitações é a **tabela de excitação**.

4.1 Elementos Biestáveis e Portas Lógicas

Os componentes capazes de assumir dois estados distintos, 0 ou 1, podem ser construídos a partir de portas lógicas. Antes de prosseguirmos, vale a pena fazer alguns comentários sobre os estados que estes componentes podem assumir de fato.

Consideremos, por exemplo, o seguinte circuito constituído por duas portas inversoras

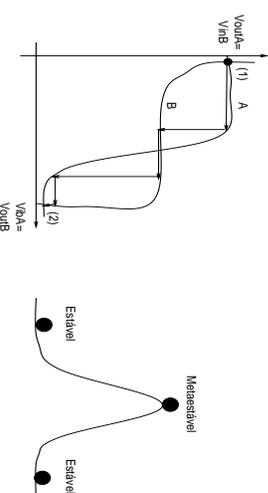


Note que a entrada da porta inversora A é **realimentada** pelo Q complementado. Como já dizemos, tais **laços de realimentação** caracterizam um circuito sequencial/com memória. No caso, o próximo estado de Q, Q^* , depende da **excitação** E1 que, por sua vez, é igual a Q. Assim, a equação de transição (de estado) é expressa por:

$$Q^* = (E1) = (Q) = Q$$

Daí se conclui que, qual for o valor inicial assumido pelo Q (0 ou 1), este valor permanecerá inalterado. Em outras palavras, o valor ficará “armazenado” no circuito.

Suponhamos que o valor armazenado seja 1 e que fosse possível chavá-lo para 0, vamos ver o **comportamento analógico** do circuito durante esta transição através de um gráfico da variação de tensão de saída (Vout) em função da tensão de entrada (Vin) em cada porta. Note no gráfico seguinte que (1) corresponde à situação inicial (VoutA = VinB = 0 e VinA = VoutB = 0). À medida que o valor de VoutA decrease, VoutB vai crescendo. Com a realimentação, VinA vai crescer, levando VoutA a 0 (2). Os pontos (1) e (2) correspondem aos dois estados digitais estáveis. Porém, há ainda um terceiro ponto de equilíbrio (3), cujo valor cai na região de indecisão de um componente digital. Chamamos este ponto de operação de **metaestável**. É metaestável, porque se não ocorrer nenhuma oscilação nas tensões, o circuito permanecerá indefinidamente neste estado.

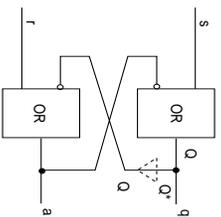


4.2 Latches

Nesta seção analisaremos o princípio de funcionamento e das principais classes de *latches*.

4.2.1 Latch RS

O circuito que satisfaz a equação (4.1) é conhecida como **latch RS** – um dispositivo que é sensível a duas entradas R e S. Uma implementação desta expressão pode ser com as portas OR



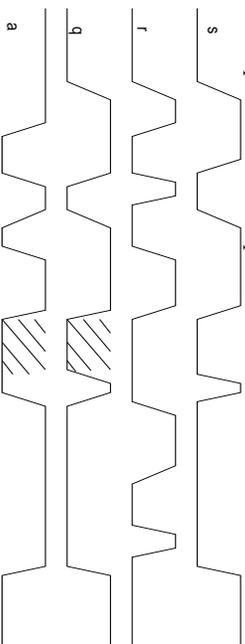
Dois símbolos lógicos comumente utilizados para representá-lo são



Uma análise cuidadosa como a do exemplo 4.5 nos leva à seguinte tabela de função para este *latch*:

r	s	q*	a*
0	0	q	a
0	1	0	1
1	0	1	0
1	1	1	1

O seguinte diagrama de tempo ilustra o comportamento deste latch para uma sequência de entradas típica

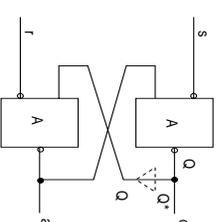


Note que para a combinação de entradas $rs=11$, $q^*=a^*=1$. E se for seguida pela combinação $rs=00$, nada se pode dizer sobre o próximo estado. Portanto, deve-se evitar esta combinação de entradas neste tipo de *latch*.

Devido a facilidades de implementação tecnológica, variantes deste circuito são encontrados

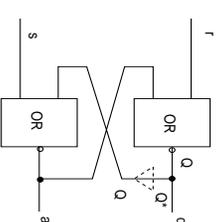
- com uso de portas NAND

$$Q^* = (s + r'q)Y = (s' \cdot (r'q)Y)$$



- com uso de portas NOR

$$Q^* = ((s + q) \cdot r')Y = ((s + q)Y + r')Y = ((s + q)Y + r)Y$$



Note que todas as equações de transição derivadas para os diferentes variantes são equivalentes à expressão:

$$Q^* = s + r'q, \tag{4.2}$$

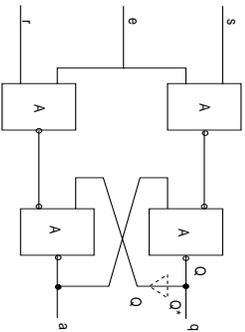
que é conhecida como **equação característica** dos *latches* RS.

Podemos ainda incluir facilmente uma entrada capaz de **desabilitar** a resposta do *latch* em relação às entradas, protegendo o seu estado corrente contra ruídos nas entradas, como mostra a seguinte tabela de transição

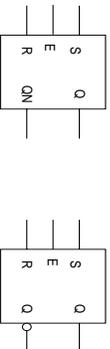
eQ		rS			
00	01	11	10		
0	0	0	0	0	0
0	1	1	1	1	1
1	1	1	1	1	0
1	0	1	1	1	0

Neste caso, a equação de transição passa a ser

$$Q^* = ((se) + (r'q) + (e'q))r = ((se)r + (r'e)q)r = ((s \cdot e) \cdot ((r \cdot e)q))r =$$



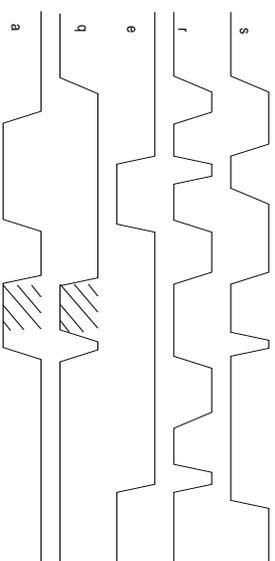
Os seguintes símbolos são usuais para representar os *latches* com habilitação.



A seguinte tabela de função resume o comportamento deste dispositivo para diferentes combinações de entradas:

r	s	e	q*	q*	a*
0	0	1	q	q	a
0	1	1	0	1	1
1	0	1	1	1	0
1	1	1	1	1	1
X	X	0	q	q	a

O comportamento do circuito no tempo, para diferentes padrões de entrada, é ilustrado no seguinte diagrama de tempo



4.2.2 Latch D

Latches RS são úteis nas situações em que precisamos setar um valor em resposta a uma condição e resetá-lo quando tal condição não seja mais satisfaita. Quando a aplicação se limita a simples armazenamento, o *latch D* é o mais recomendado, cujo comportamento pode ser descrito com a seguinte tabela de estados

q	eD			
	00	01	11	10
S ₀	S ₀	S ₀	S ₁	S ₀
S ₁	S ₁	S ₁	S ₁	S ₀

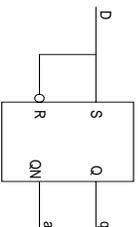
que em representações binárias seria

q	eD			
	00	01	11	10
0	0	0	1	0
1	1	1	1	0

O *latch D* pode ser realizado com uso do *latch* RS. A partir da equação (4.2) podemos ver que a excitação necessária no *latch* RS habilitado ($e=1$) é

q	D	
	1	0
0	10	01
1	10	01

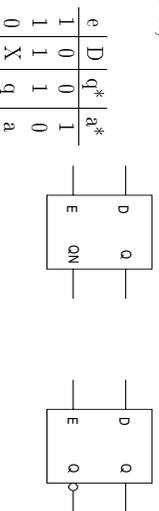
Ou seja, S=D e R=D'. O diagrama lógico deste *latch* é então



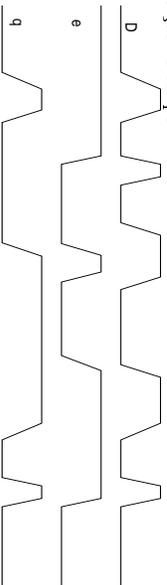
É fácil, também, verificar que a equação característica deste *latch* é

$$Q^* = D. \quad (4.3)$$

A sua tabela de função e os símbolos usuais para designá-lo são, respetivamente,



O seguinte diagrama de tempo mostra o comportamento deste *latch* em função do tempo



4.2.3 Latch JK

Um *latch* JK tem um comportamento igual ao dos *latches* RS, com exceção na combinação de entradas $kj=11$ para a qual o próximo estado é sempre o complementar do estado corrente

Q		kj			
		00	01	11	10
S_0	S_1	S_0	S_1	S_0'	S_0
S_1	Q^*	S_1	S_1'	S_1'	S_0

Em termos de valores binários,

Q		kj			
		00	01	11	10
0	0	0	1	1	0
1	1	1	0	0	1

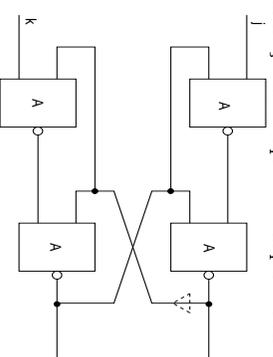
cujas equação de transição envolve as seguintes variáveis:

$$Q^* = jq' + kq \quad (4.4)$$

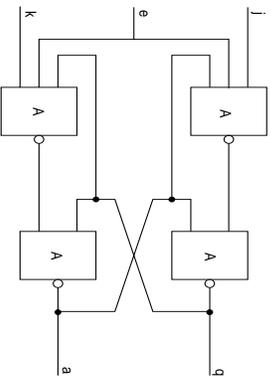
Desenvolvendo a expressão, podemos chegar a

$$\begin{aligned} Q^* &= jq' + kq = (jq' + kq)j = \\ &= ((jq') \cdot (kq))j = ((jq') \cdot (kq))j = \\ &= ((jq') \cdot (kq + qj))j = ((jq') \cdot ((k + q)q))j = \\ &= ((jq') \cdot ((kq)jq))j \end{aligned}$$

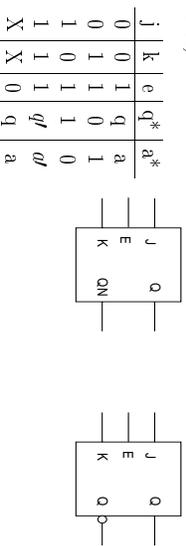
Uma implementação desta expressão com portas NAND seria



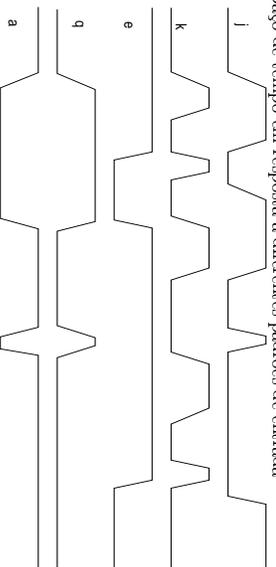
Se incluímos o sinal de controle de habilitação, e , para proteger o estado corrente e do dispositivo contra ruídos nas entradas, teremos o seguinte diagrama lógico:



A sua tabela de função e os símbolos usuais para designá-lo são, respetivamente,



O seguinte diagrama de tempo mostra o comportamento deste *latch* no espaço de tempo em resposta a diferentes padrões de entrada



Um *latch* JK pode também ser construído a partir de um *latch* RS adicionando duas portas AND, se observarmos que a equação característica do *latch* RS é

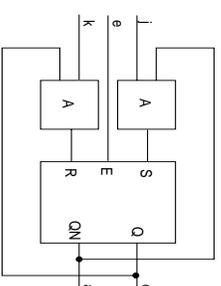
$$Q^* = s + r'q.$$

Ou seja, a tabela de excitação do *latch* RS deve ser a seguinte para termos as transições expressas pela equação (4.4):

Q		k			
		00	01	11	10
j	0	00	01	01	00
	1	00	00	10	10
	RS				

A partir desta tabela de excitação é fácil ver que

$$s = jq' \quad e \quad r = kq$$



4.3 Flip-Flops

Diferentemente dos *latches*, que enquanto habilitados tem as suas saídas refletindo constantemente as entradas, os *flip-flops* são os dispositivos que só mudam as suas saídas nos tempos determinados pelas transições de um sinal de controle que denominaremos de **sinal de relógio** (*clock*) ou **sinal de gatilho** (*trigger*).

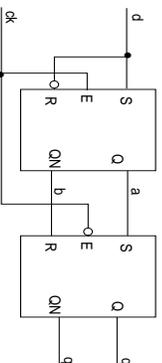
Quanto à forma de “engatilhar” um *flip-flop*, existem essencialmente duas classes de *flip-flops*:

- sensível à borda (de subida ou de entrada do sinal de gatilho) e
- sensível ao pulso.

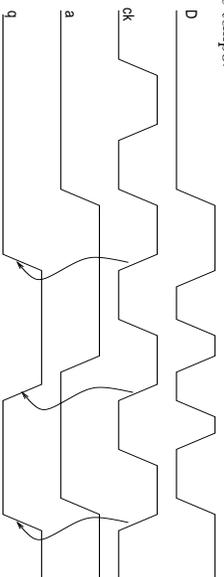
Nesta seção analisaremos o funcionamento dos diferentes variantes dos *flip-flops*.

4.3.1 Flip-Flops D

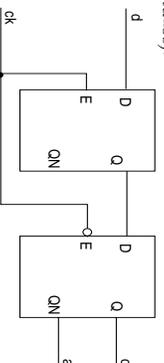
Uma forma mais simples de construir um *flip-flop* D é colocar em configuração **mestre-escravo** dois *latches* RS com habilitação (*flip-flop* D de 4 estados)



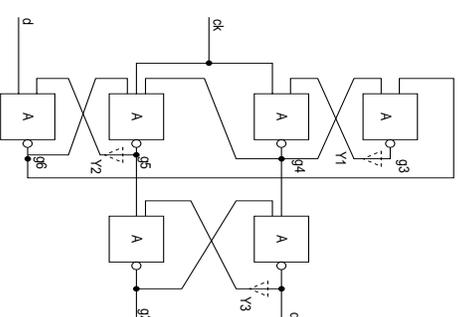
A configuração **maestro-escravo** se caracteriza por ter duas instâncias para mudar o estado do dispositivo em relação ao sinal de controle *ck*: a primeira ($k=1$) para captar as entradas (excitação) e a segunda ($k=0$) para atualizar as saídas (novo estado). A transição para o próximo estado ocorre exatamente na transição de 0 para 1 do sinal *ck* e o valor da saída depende exclusivamente das entradas naquele instante. Portanto, dizemos que é um **flip-flop sensível à borda (de descida)**, como ilustra o seguinte diagrama de tempo.



Podemos também construir um *flip-flop* D com dois *latches* D com habilitação (de 4 estados).



Uma outra forma alternativa para construir um *flip-flop* D é com uso de portas NAND (de 8 estados) conforme mostra o seguinte diagrama lógico.



Neste circuito temos três variáveis de estado internos: Y_1 , Y_2 e Y_3 , capazes de representar 8 (2^3) estados distintos. Por inspeção, podemos derivar a partir do diagrama as seguintes equações de transição:

$$\begin{aligned} Y_1^* &= ((Y_1 Ck) \cdot (Y_2 d) \gamma) + (Y_1 Ck) \mu + (Y_2 d) \mu = Y_1 Ck + Y_2 d \\ Y_2^* &= ((Y_2 d) \gamma \cdot Ck \cdot (Y_1 Ck) \mu) + ((Y_2 d) Ck) \mu + (Y_1 Ck) \mu = Y_2 d + Ck \mu + Y_1 \\ Y_3^* &= ((Y_3 \cdot (Y_2 d + Ck \mu + Y_1) \gamma) \cdot (Y_1 Ck) \mu) = Y_2 Y_3 d + Ck Y_3 \mu + Y_1 Y_3 + Y_1 Ck \end{aligned} \quad (4.5)$$

e as seguintes funções de saída:

$$\begin{aligned} q &= Y_3^* = Y_2 Y_3 d + Ck Y_3 \mu + Y_1 Y_3 + Y_1 Ck \\ g_2 &= Y_3 \mu + (Y_2 d + Ck \mu + Y_1) \gamma = Y_3 \mu + Y_1 Ck \mu (Y_2 d) \mu \\ &= Y_3 \mu + Y_1 Ck (\mu Y_2 + d) = Y_3 \mu + Y_1 Y_2 Ck + Y_1 Ck d \end{aligned}$$

Com estas equações podemos montar a tabela de transições do circuito e identificar os estados (totalmente) estáveis nas transições. Para facilitar a análise, incluímos ainda na tabela as saídas (g_2) para cada estado corrente.

Y1Y2Y3	ck d			
	00	01	11	10
000	010,01	010,01	(000,01)	(000,01)
001	011,10	011,10	000,01	000,01
010	(010,01)	110,01	110,01	000,01
011	(011,10)	111,10	111,10	000,01
100	010,01	010,01	111,11	111,11
101	011,10	011,10	111,10	111,10
110	010,01	(110,01)	111,11	111,11
111	011,10	(111,10)	(111,10)	(111,10)

Y1Y2Y3.ag2

A partir desta tabela de transição, podemos verificar que o comportamento lógico deste circuito é equivalente ao dos dois circuitos anteriores. Analisando, caso a caso, para diferentes estados correntes:

Y1Y2Y3	ck d			Y1Y2Y3	ck d			
	00	01	11		10	00	01	11
000	010,01	010,01	(000,01)	(000,01)	010,01	010,01	(000,01)	(000,01)
001	011,10	011,10	000,01	000,01	001	011,10	000,01	000,01
010	(010,01)	110,01	110,01	000,01	010	(010,01)	110,01	110,01
011	(011,10)	111,10	111,10	000,01	011	(011,10)	111,10	111,10
100	010,01	010,01	111,11	111,11	100	010,01	010,01	111,11
101	011,10	011,10	111,10	111,10	101	011,10	011,10	111,10
110	010,01	(110,01)	111,11	111,11	110	010,01	(110,01)	111,11
111	011,10	(111,10)	(111,10)	(111,10)	111	011,10	(111,10)	(111,10)

Y1Y2Y3.ag2

Estado=000; ck d=11

Estado=000; ck d=10

Estado=010; ck d=00

Estado=011; ck d=00

Y1Y2Y3	ck d			Y1Y2Y3	ck d				
	00	01	11		10	00	01	11	10
000	010,01	010,01	(000,01)	(000,01)	000	010,01	010,01	(000,01)	(000,01)
001	011,10	011,10	000,01	000,01	001	011,10	011,10	000,01	000,01
010	(010,01)	110,01	110,01	000,01	010	(010,01)	110,01	110,01	000,01
011	(011,10)	111,10	111,10	000,01	011	(011,10)	111,10	111,10	000,01
100	010,01	010,01	111,11	111,11	100	010,01	010,01	111,11	111,11
101	011,10	011,10	111,10	111,10	101	011,10	011,10	111,10	111,10
110	010,01	(110,01)	111,11	111,11	110	010,01	(110,01)	111,11	111,11
111	011,10	(111,10)	(111,10)	(111,10)	111	011,10	(111,10)	(111,10)	(111,10)

Y1Y2Y3.ag2

Estado=110; ck d=01

Estado=111; ck d=01

Y1Y2Y3	ck d			Y1Y2Y3	ck d				
	00	01	11		10	00	01	11	10
000	010,01	010,01	(000,01)	(000,01)	000	010,01	010,01	(000,01)	(000,01)
001	011,10	011,10	000,01	000,01	001	011,10	011,10	000,01	000,01
010	(010,01)	110,01	110,01	000,01	010	(010,01)	110,01	110,01	000,01
011	(011,10)	111,10	111,10	000,01	011	(011,10)	111,10	111,10	000,01
100	010,01	010,01	111,11	111,11	100	010,01	010,01	111,11	111,11
101	011,10	011,10	111,10	111,10	101	011,10	011,10	111,10	111,10
110	010,01	(110,01)	111,11	111,11	110	010,01	(110,01)	111,11	111,11
111	011,10	(111,10)	(111,10)	(111,10)	111	011,10	(111,10)	(111,10)	(111,10)

Y1Y2Y3.ag2

Estado=111; ck d= 11

Y1Y2Y3	ck d			Y1Y2Y3	ck d		
	00	01	11		00	01	11
000	010,01	010,01	(000)01	010,01	010,01	(000)01	(000)01
001	011,10	011,10	000,01	011,10	011,10	000,01	000,01
010	(010)01	110,01	110,01	(010)01	110,01	110,01	000,01
011	(011)0	111,10	000,01	(011)0	111,10	111,10	000,01
100	010,01	010,01	111,11	010,01	010,01	111,11	111,11
101	011,10	011,10	111,10	011,10	011,10	111,10	111,10
110	010,01	(110)01	111,11	010,01	(110)01	111,11	111,11
111	011,10	(111)0	(111)0	011,10	(111)0	(111)0	(111)0

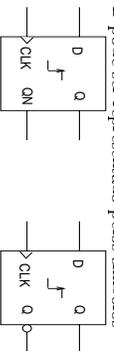
A partir das transições apresentadas nas tabelas, concluímos que

1. não há mudança no valor da saída q quando o sinal ck varia de 1 (estados 000 e 111) a 0 (estados 010, 110).
2. não há mudança no valor da saída q quando o sinal ck permanece constante, como na mudança dos estados 110 e 111 para 010 e 011, respectivamente.
3. há mudança no valor de saída quando o sinal ck varia de 0 para 1. Com exceção do estado 111, a saída segue o valor da entrada d.

A este comportamento, dizemos que o *flip-flop* é **sensível a borda de subida**. Sintetizando, temos a seguinte tabela de função:

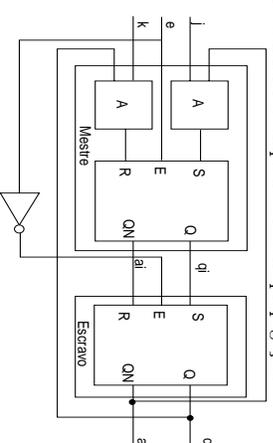
d	ck	q*	q1*
0	┐	0	1
1	┐	1	0
X	0	q	q1
X	1	q	q1

Um *flip-flop* D pode ser representado pelos símbolos



4.3.2 Flip-Flops JK

Uma análise cuidadosa dos diagramas lógicos apresentados na seção 4.2.3 nos leva a concluir que para a combinação de entradas $jk=11$, se os pulsos $j=1$ e $k=1$ forem muito largos, o estado do *latch* ficará variando entre 0 e 1. Lembrando dos atrasos de propagação, podemos, porém, afirmar que se os atrasos de propagação forem maiores do que a duração dos pulsos nas entradas, tal oscilação não ocorrerá. O uso da configuração **mestre-esravo** nos permite controlar o tempo de atraso na propagação.

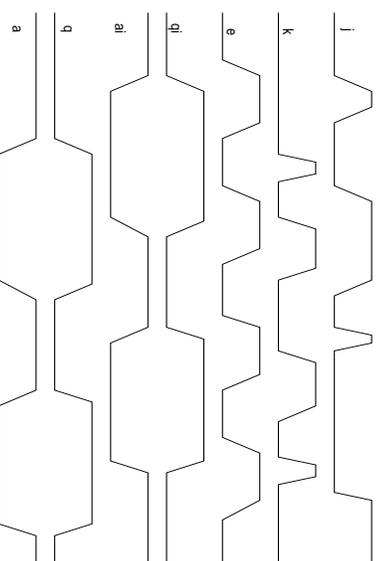


Observe que os sinais de realimentação (do escravo), q e a , são mantidos invariantes na primeira instância e nesta instância as entradas só afetam as variáveis qi e ai

$$qi = j\bar{q} = j\bar{a} \quad e \quad ai = kq$$

Numa segunda instância, os valores de qi e ai são mantidos invariantes e “transferidos” para as saídas. Assim, pode-se “segurar” o estado do dispositivo mesmo com as entradas mantidas em 11.

Vale ressaltar ainda que os valores qi e ai que são transferidos para o estágio de saída (escravo) durante a segunda instância dependem do “histórico” das entradas jk durante o pulso $e=1$. Note no seguinte diagrama de tempo que na terceira borda de descida do sinal e as entradas do mestre são $rs = 00$ indicando que as saídas qi e ai que serão transferidos para o escravo dependem do instante anterior. Portanto, diferentemente dos dispositivos sensíveis a borda que vimos até agora, estes *flip-flops* são conhecidos como **sensíveis a pulsos**.



Uma forma para representá-lo



As funções deste *flip-flop* são sintetizadas na seguinte tabela (de função)

j	k	clk	q*	q'
X	X	0	q	q'
0	0	1	q	q'
0	1	1	0	1
1	0	1	1	0
1	1	1	q'	q

Podem-se construir *flip-flops JK sensíveis a borda* com uso de *flip-flops*

D. Partindo da tabela de transição dos *flip-flops JK*

Q		kj		Q*	
0	0	0	1	1	1
0	1	1	1	0	0
1	0	1	0	1	1
1	1	0	0	0	0

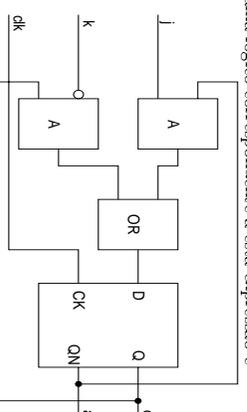
e sabendo que a equação característica de um *flip-flop D* é $Q^* = D$, então a tabela de excitação é

Q		kj		D	
0	0	0	1	1	0
0	1	1	1	0	0
1	0	1	0	1	1
1	1	0	0	0	0

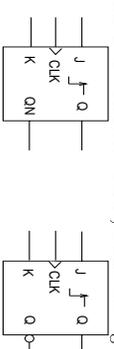
e a equação de excitação do *flip-flop D* é

$$D = jq' + kq$$

O diagrama lógico correspondente e a esta expressão é



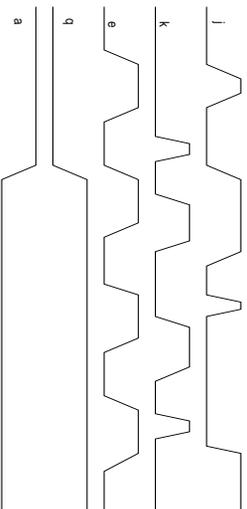
Para diferenciá-lo dos sensíveis à borda, utiliza-se a seguinte representação



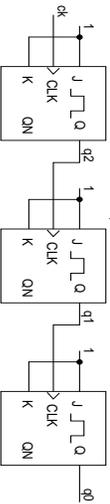
e a sua tabela de funções contém as seguintes possíveis combinações

j	k	clk	q*	q'
X	X	0	q	q'
X	X	1	q	q'
0	0	1	q	q'
0	1	1	0	1
1	0	1	1	0
1	1	1	q'	q

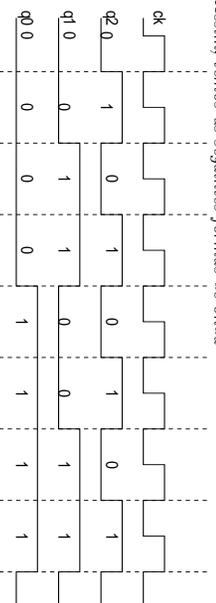
O seu comportamento no tempo é ilustrado no seguinte diagrama de tempo



Exemplo 4.6 Descreva as formas de onda q_2 , q_1 e q_0 para o circuito abaixo (condição inicial $q_2 = q_1 = q_0 = 0$)

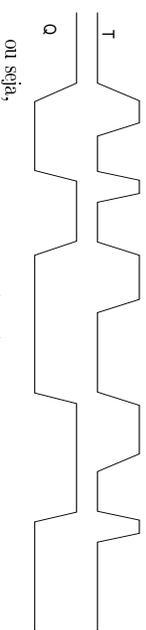


Note que é um *flip-flop sensível a pulso*. Portanto, as mudanças de estado só ocorrem nas transições de borda de descida do sinal de relógio CLK de cada *flip-flop*. Ainda mais, as combinações de entradas são $j_k=11$, implicando que as mudanças são sempre as de “complementar” o sinal anterior. Assim, temos as seguintes formas de onda

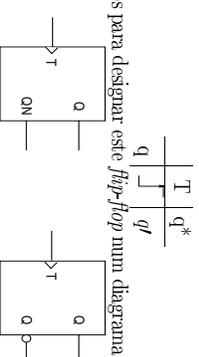


4.3.3 Flip-Flops T

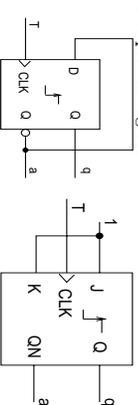
Os *flip-flops T (toggle)* mudam de estado a cada transição do relógio. Mais precisamente, o próximo estado é sempre o complemento do estado corrente, como ilustra o seguinte diagrama de tempo



Os símbolos para designar este *flip-flop* num diagrama lógico é



Estes *flip-flops* não são disponíveis comercialmente, pois eles são fáceis de serem implementados com uso dos *flip-flops D* e *JK*, como mostram os seguintes esquemas lógicos.



4.4 Resumo

Sintetizando, foram apresentados os *flip-flops SR*, *D*, *JK* e *T*, cujas saídas podem depender da combinação das entradas e do estado corrente conforme mostra a seguinte tabela:

Tipos	Equações Características
<i>Latch SR</i>	$q^* = s + r' \cdot q$
<i>Latch D</i>	$q^* = d$
<i>Flip-flop D</i>	$q^* = d$
<i>Flip-flop D</i> com <i>enable e</i>	$q^* = e \cdot d + e'q$
<i>Flip-flop SR</i> sensível a pulso	$q^* = s + r'q$
<i>Flip-flop JK</i> sensível a pulso	$q^* = j \cdot q' + k' \cdot q$
<i>Flip-flop JK</i> sensível a borda	$q^* = j \cdot q' + k' \cdot q$
<i>Flip-flop T</i>	$q^* = q'$
<i>Flip-flop T</i> com <i>enable e</i>	$q^* = e \cdot q' + e'q$

É, sob o ponto de vista das possíveis transições que podem ocorrer na saída q , as seguintes excitações devem ser aplicadas nestes *flip-flops*:

q^*	q	D	S	R	J	K	T
0	0	0	0	X	0	X	0
0	1	1	1	0	1	X	1
1	0	0	0	1	X	1	1
1	1	1	X	0	X	0	0