Capítulo VII

Elementos de Memória

1 Introdução

Neste capítulo estudaremos dispositivos lógicos com dois estados estáveis, o estado **SET** e o estado **RESET**. Por isto, tais dispositivos são denominados **dispositivos biestáveis**.

Uma vez que estes dispositivos são capazes de reter indefinidamente o seu estado (SET ou RESET), eles são usados como elementos de armazenamento de informação. Informalmente, dispositivos biestáveis "memorizam" o seu estado.

Estudaremos dois tipos de dispositivos biestáveis: o ***latch*** e o ***flip-flop***. A diferença entre um *latch* e um *flip-flop* é a maneira como ocorre a troca de estado:

- Um *flip-flop* muda seu estado por ação de um pulso de disparo, denominado de ***clock***. Por este motivo, um *flip-flop* é caracterizado como um **dispositivo biestável síncrono**, porque somente muda de estado em sincronismo com a ocorrência do pulso de *clock*.

- Um *latch* , por sua vez, é caracterizado como um **dispositivo biestável assíncrono**, porque muda de estado sem necessidade de sincronismo com um trem de pulsos de controle (pulsos de *clock*).

2 O *latch* S-R

- A Figura 1 mostra o diagrama lógico de um *latch* S-R (SET-RESET) implementado com portas NAND:

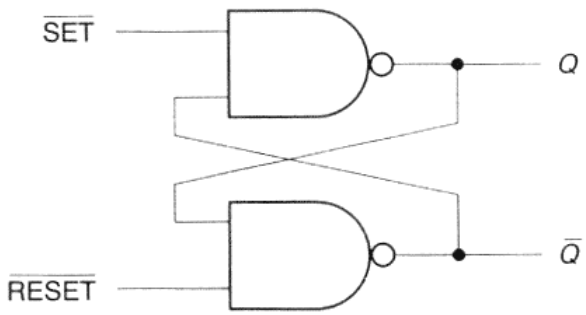


Figura 1: Diagrama lógico de um *latch* S-R implementado com portas NAND. O *latch* possui duas entradas , ($\overline{SET}$ e $\overline{RESET}$) e duas saídas (Q e $\overline{Q}$). O valor lógico das saídas Q e $\overline{Q}$ definem o estado (SET ou RESET) do *latch*.

- Ao fazermos $\overline{\text{SET}} = 0$ e $\overline{\text{RESET}} = 1$, as saídas resultam nos valores lógicos $Q = 1$ e $\overline{Q} = 0$. Nesta situação o *latch* é dito estar no estado SET, conforme mostrado na Figura 2:

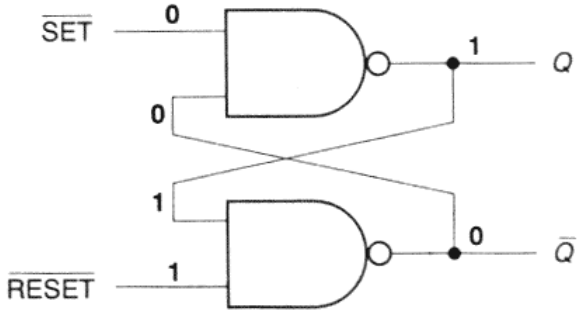


Figura 2: Colocando um *latch* S-R (NAND) no estado SET.

- Note que se $\overline{\text{SET}} = 1$ e $\overline{\text{RESET}} = 1$, o *latch* mantém o estado atual. Por exemplo, para o *latch* no estado SET mostrado na Figura 2, o estado não é alterado quando fazemos $\overline{\text{SET}} = 1$ e $\overline{\text{RESET}} = 1$:

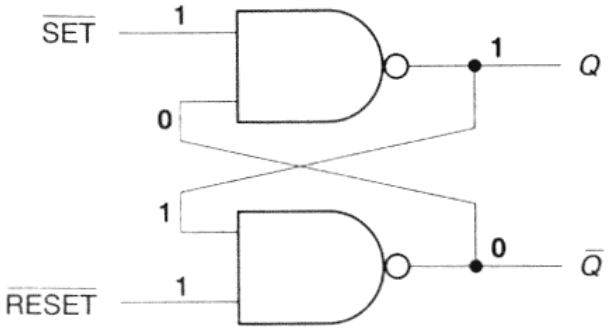


Figura 3: Mantendo o estado de um *latch* S-R (NAND).

- Quando $Q = 0$ e $\overline{Q} = 1$, o *latch* é dito estar no estado RESET. Para "resetar" o *latch* da Figura 3 fazemos $\overline{\text{SET}} = 1$ e $\overline{\text{RESET}} = 0$, conforme mostrado na Figura 4:

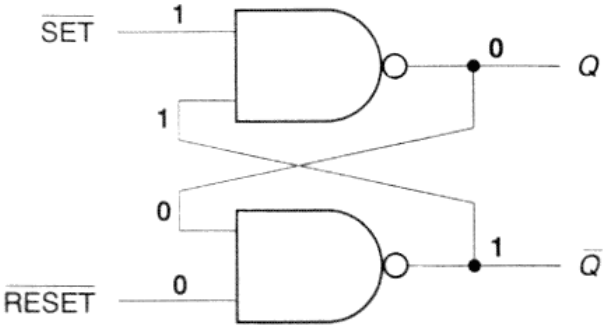


Figura 4: Colocando um *latch* S-R (NAND) no estado RESET.

- Observe que se $\overline{SET} = 0$ e $\overline{RESET} = 0$, o *latch* apresenta em suas saídas os valores lógicos $Q = 1$ e $\overline{Q} = 1$, conforme mostra a Figura 5. Esta é uma **condição inválida**, na medida em que as variáveis lógicas Q e $\overline{Q}$ possuem valores lógicos idênticos e, portanto, incoerentes com a definição destas variáveis à luz da álgebra booleana.

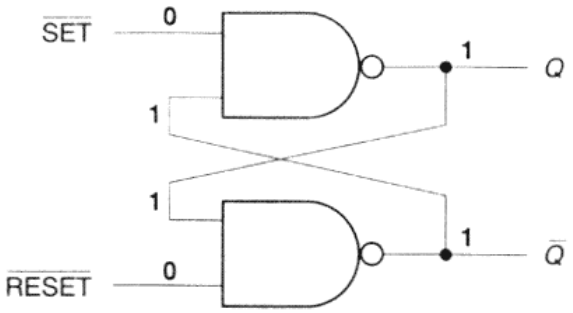


Figura 5: Condição inválida em um *latch* S-R (NAND).

- A Figura 6 mostra a Tabela Verdade de um *latch* S-R (NAND):

$\overline{SET}$	$\overline{RESET}$	Q	$\overline{Q}$	
0	0	1	1	Condição Inválida
0	1	1	0	
1	0	0	1	
1	1	Q	$\overline{Q}$	Saída Inalterada

Figura 6: Tabela Verdade de um *latch* S-R (NAND).

- As Figuras 7 a 12 mostram as características de operação um *latch* S-R implementado com portas NOR:

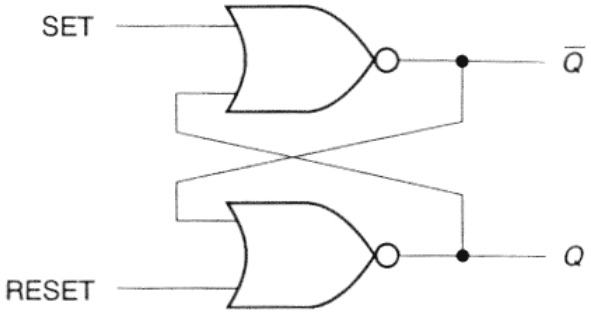


Figura 7: Diagrama lógico de um *latch* S-R implementado com portas NOR. O *latch* possui duas entradas , ($\overline{SET}$ e $\overline{RESET}$) e duas saídas (Q e $\overline{Q}$). O valor lógico das saídas Q e $\overline{Q}$ definem o estado (SET ou RESET) do *latch*.

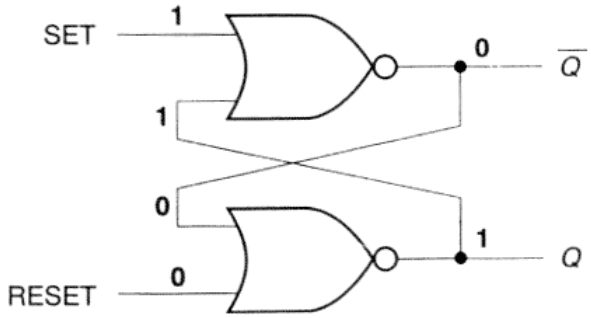


Figura 8: Colocando um *latch* S-R (NOR) no estado SET.

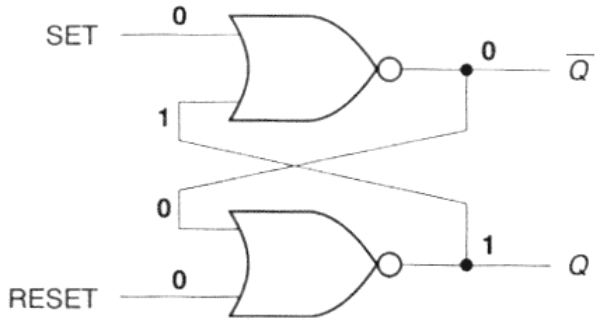


Figura 9: Mantendo o estado de um *latch* S-R (NOR).

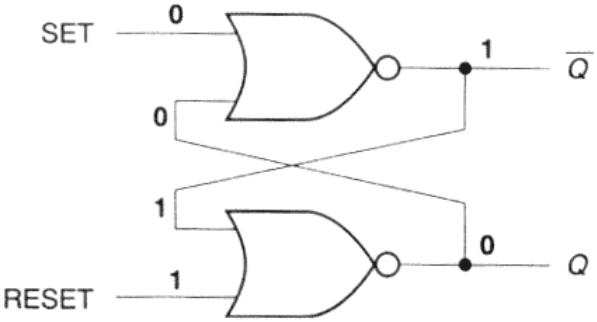


Figura 10: Colocando um *latch* S-R (NOR) no estado RESET.

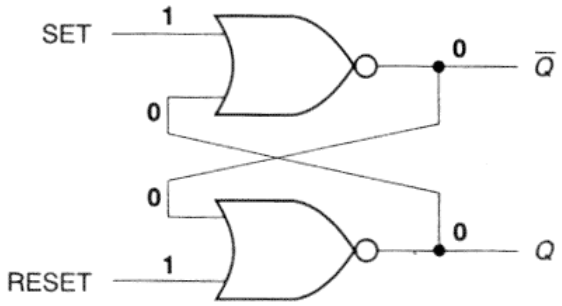


Figura 11: Condição inválida em um *latch* S-R (NOR).

$\overline{SET}$	$\overline{RESET}$	Q	$\overline{Q}$	
0	0	0	0	Condição Inválida
0	1	1	0	
1	0	0	1	
1	1	Q	$\overline{Q}$	Saída Inalterada

Figura 12: Tabela Verdade de um *latch* S-R (NOR).

Exemplo 1: Determine as formas de onda nas saídas Q e $\overline{Q}$ de um *latch* S-R (NAND) e nas saídas Q e $\overline{Q}$ de um *latch* S-R (NOR) quando as formas de onda de entrada são conforme a figura a seguir.

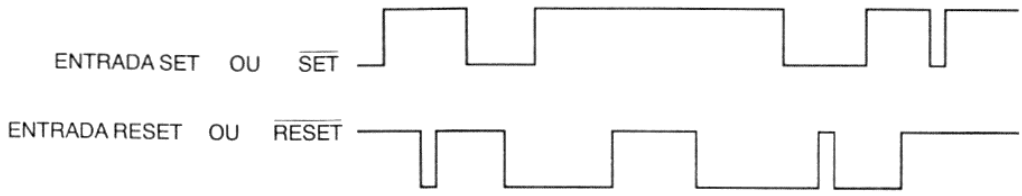


Figura 13

Solução:

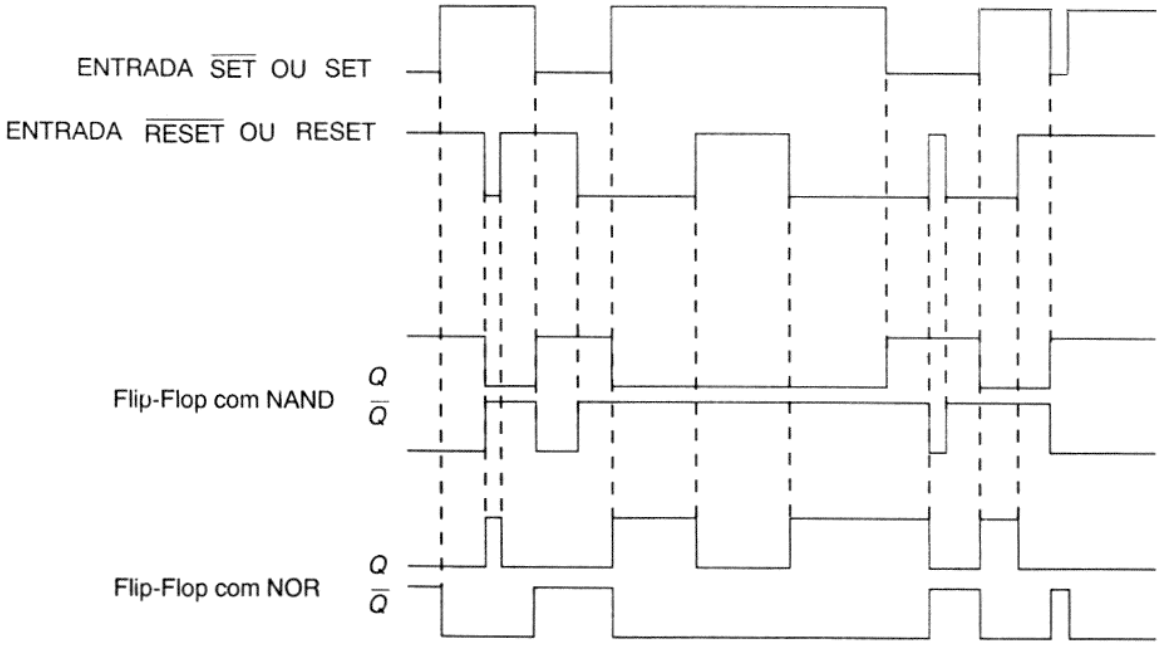


Figura 14

- Uma aplicação simples de um *latch* é o circuito de *debouncing* (*bouncing* – repique) em chaves comutadoras:

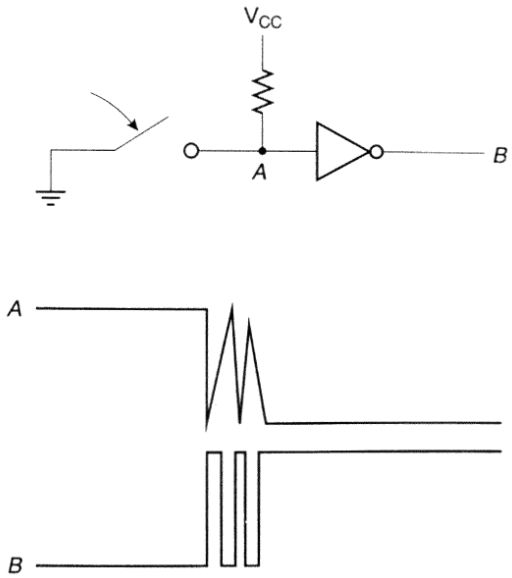


Figura 15: Efeito indesejável de *bouncing* em uma chave comutadora.

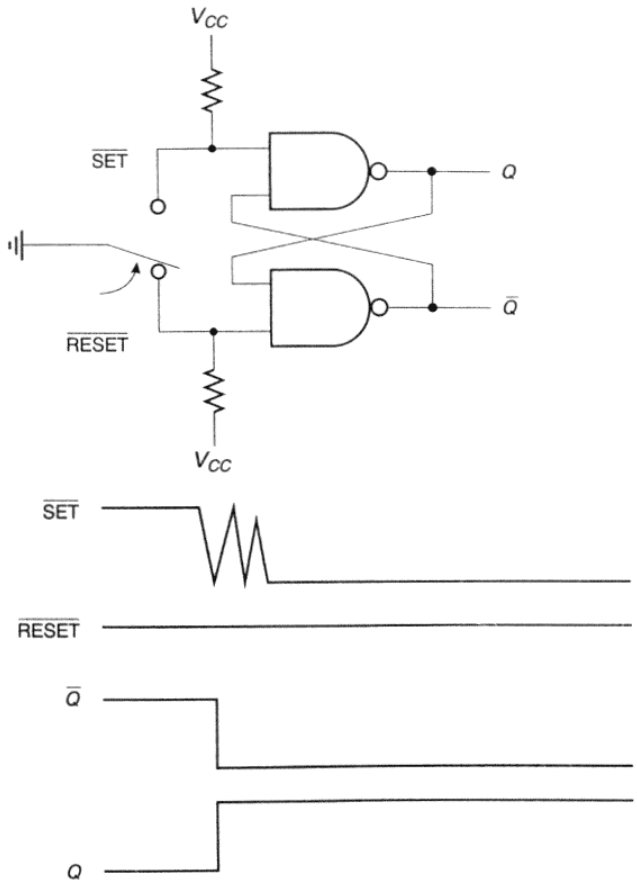


Figura 16: Solução do problema de *bouncing* em uma chave comutadora utilizando um *latch* S-R (NAND).

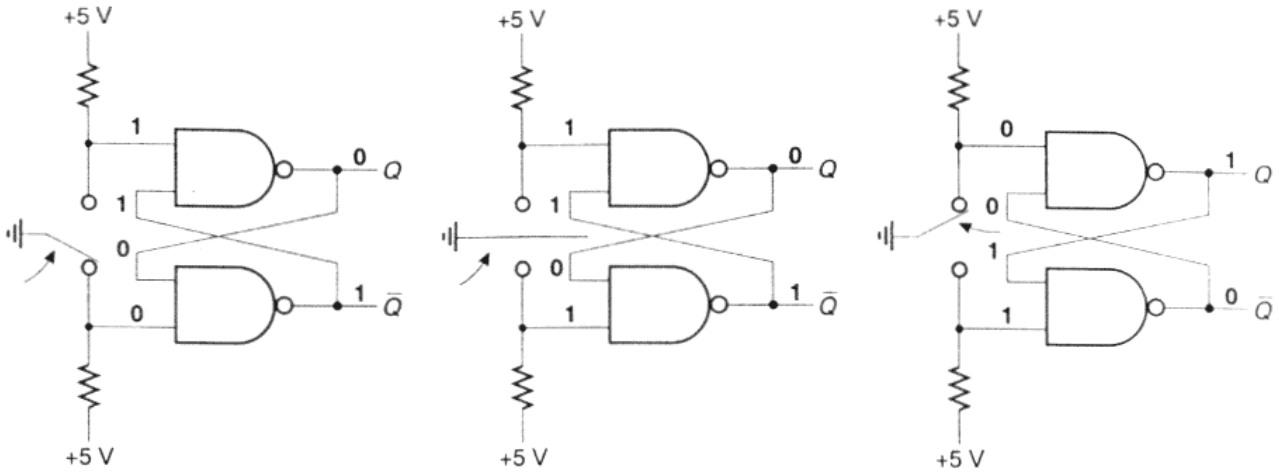


Figura 17: Note que enquato a chave do circuito de *debouncing* da Figura 16 encontra-se movimentando-se a meio caminho entre seus dois contatos, o *latch* mantém o estado.

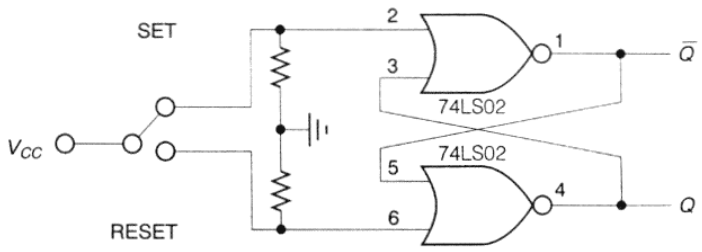


Figura 18: Circuito de *debouncing* utilizando um *latch* S-R (NOR).

3 O *flip-flop* S-R

● A Figura 19 mostra o diagrama lógico de um *flip-flop* S-R (SET-RESET) implementado com portas NAND:

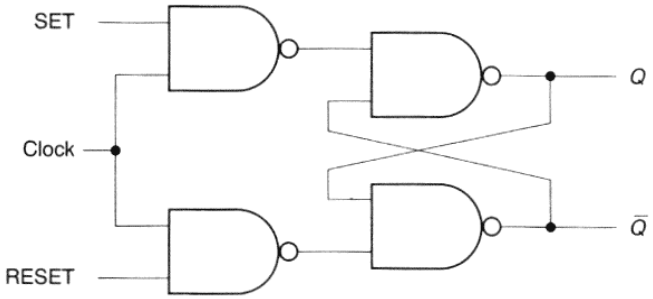


Figura 19: Diagrama lógico de um *flip-flop* S-R implementado com portas NAND. O *flip-flop* possui três entradas (SET, RESET e CLOCK) e duas saídas (Q e $\overline{\mathrm{Q}}$). O valor lógico das saídas Q e $\overline{\mathrm{Q}}$ definem o estado (SET ou RESET) do *flip-flop*. Note que ação da entrada CLOCK é habilitar ($\mathrm{CLOCK} = 1$) ou desabilitar ($\mathrm{CLOCK} = 0$) a mudança de estado do *flip-flop*, mudança que está ao encargo das entradas SET e RESET.

Clock	SET	RESET	Q	$\overline{Q}$	
0	0	0	Q	$\overline{Q}$	Saída Inalterada
0	0	1	Q	$\overline{Q}$	
0	1	0	Q	$\overline{Q}$	
0	1	1	Q	$\overline{Q}$	
1	0	0	Q	$\overline{Q}$	
1	0	1	0	1	
1	1	0	1	0	
1	1	1	1	1	Condição Inválida

Figura 20: Tabela Verdade de um *flip-flop* S-R (NOR).

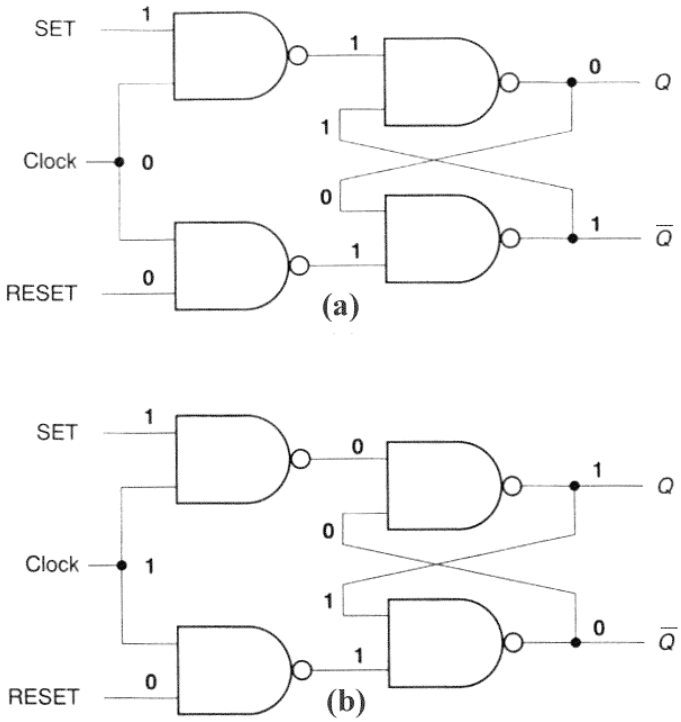


Figura 21: Habilitando a mudança de estado de um *flip-flop* S-R (NOR). Em (a) o *flip-flop* encontra-se no estado RESET apesar de $\mathrm{SET} = 1$ porque $\mathrm{CLOCK} = 0$, desabilitando a mudança de estado. Em (b) $\mathrm{CLOCK} = 1$, habilitando a troca de estado do *flip-flop* para o estado SET.

Exemplo 2: Determine as formas de onda nas saídas Q e $\overline{\mathrm{Q}}$ do *flip-flop* S-R (NAND) mostrado na Figura 19 quando as formas de onda de entrada são conforme a figura a seguir.

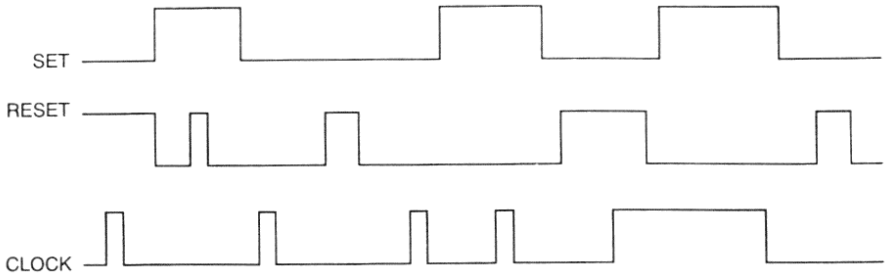


Figura 22

Solução:

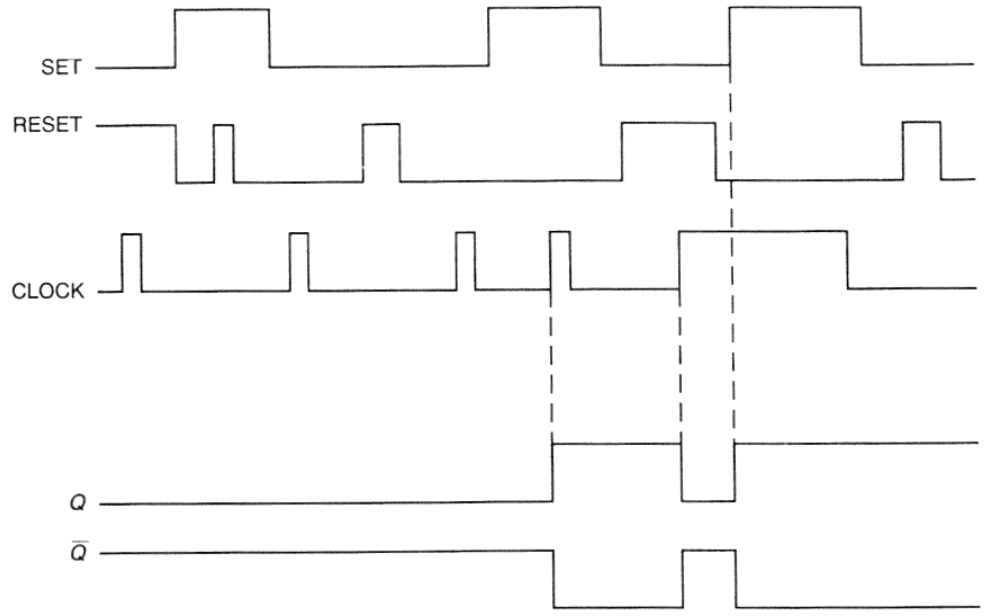


Figura 23

4 O *flip-flop* D

● A Figura 24 mostra o diagrama lógico de um *flip-flop* D (*Data* – dados) implementado com portas NAND:

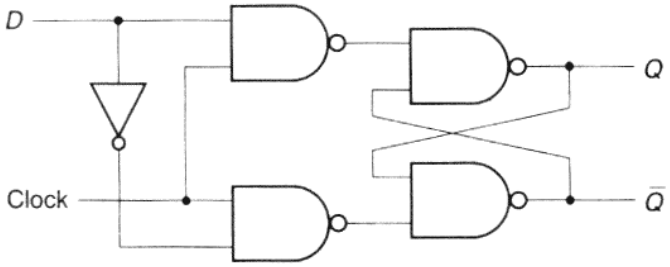


Figura 24: Diagrama lógico de um *flip-flop* D implementado com portas NAND.

O *flip-flop* possui duas entradas (D e $CLOCK$) e duas saídas (Q e $\overline{Q}$). A principal característica funcional de um *flip-flop* D é que o valor lógico da entrada de dados D é transferido para a saída Q toda vez que $CLOCK = 1$.

D	Clock	Q	$\overline{Q}$	
0	0	Q	$\overline{Q}$	Saída Inalterada
1	0	Q	$\overline{Q}$	
0	1	0	1	
1	1	1	0	

Figura 25: Tabela Verdade de um *flip-flop* D (NAND).

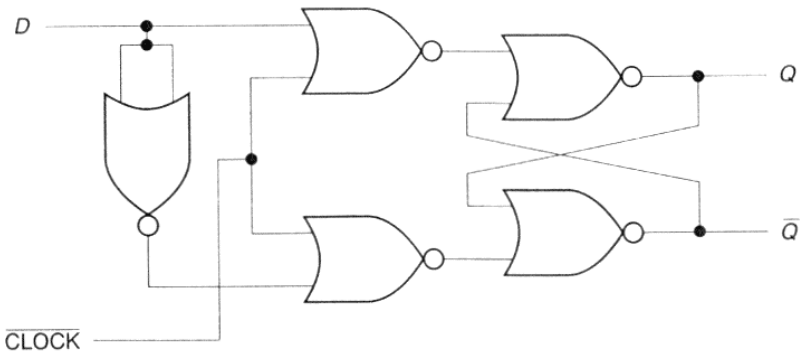


Figura 26: Diagrama lógico de um *flip-flop* D implementado com portas NOR.

● Uma das aplicações de um *flip-flop* D é o armazenamento de palavras binárias. Por exemplo, a Figura 27 mostra o diagrama de Interligação de uma porta de saída com o barramento de dados de um sistema microprocessado.

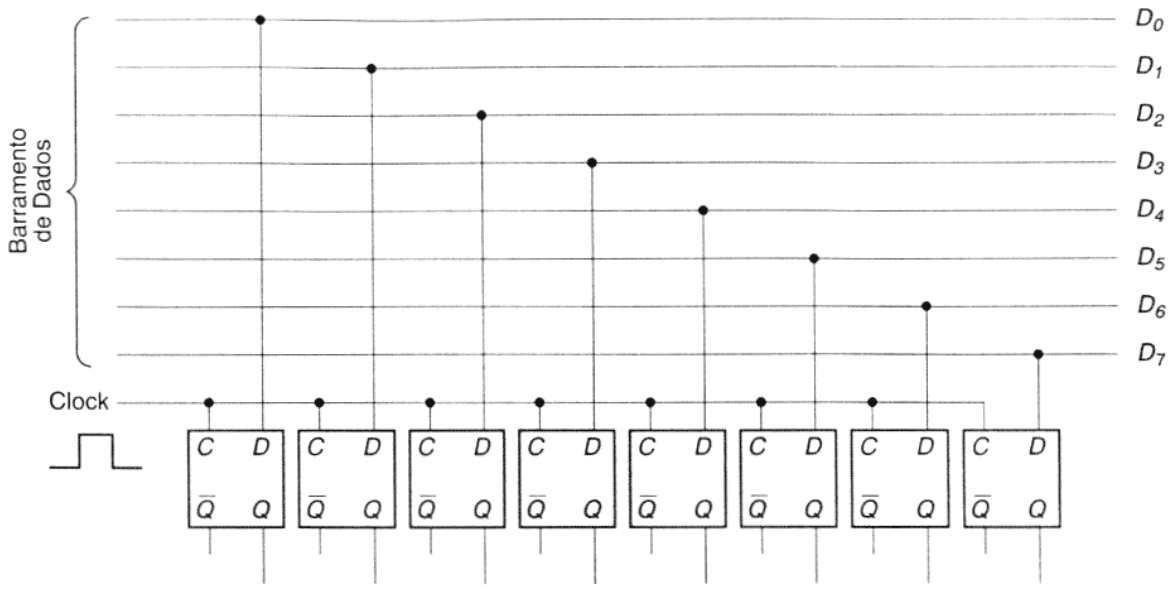


Figura 27: Interligação de uma porta de saída com o barramento de dados de 8 bits de um sistema microprocessado.

● Quando o microprocessador recebe uma instrução para enviar uma palavra binária de 8 bits para a porta em questão, a palavra é colocada no barramento por um breve instante de tempo até a ocorrência do pulso de *clock*.

● Uma vez ocorrido o *clock*, a palavra de 8 bits é transferida para as saídas Q dos 8 *flip-flops* D, ficando ali armazenada até que o periférico conectado à porta a utilize.

- Note que com este esquema de armazenamento, o barramento não fica "preso" ao periférico conectado à porta, não havendo necessidade de parar o processamento até que o periférico utilize a palavra de 8 bits a ele destinada.

5 O *flip-flop* D MS (*Master-Slave*)

- A Figura 28 mostra o diagrama lógico de um *flip-flop* D MS (*master-slave* = mestre-escravo):

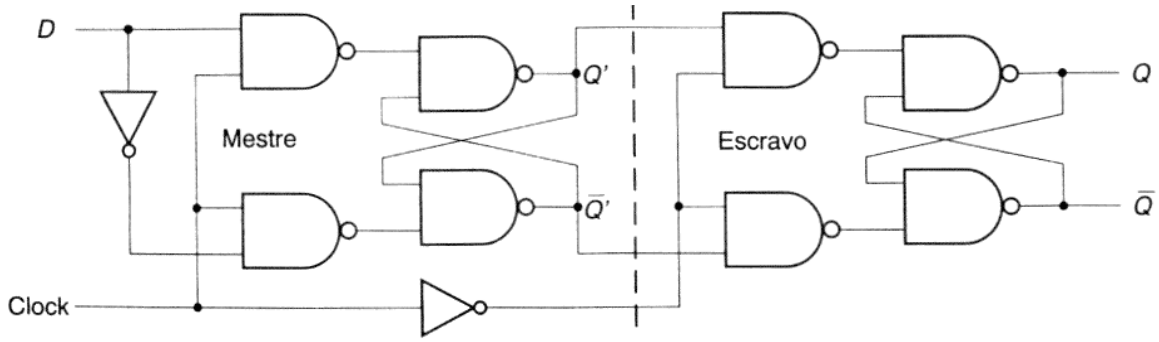


Figura 28: Diagrama lógico de um *flip-flop* D MS implementado com portas NAND. O *flip-flop* subdivide-se em uma seção "Mestre" e em uma seção "Escravo". A seção "Mestre" é um *flip-flop* D e é habilitada pelo sinal CLOCK. A seção "Escravo" é um *flip-flop* S-R e é habilitada pelo sinal $\overline{\mathrm{CLOCK}}$.

- A principal característica funcional deste *flip-flop* D MS é que o valor lógico da entrada de dados D é transferido para a saída Q **apenas na borda de descida** do pulso de CLOCK, conforme mostra a Figura 29:

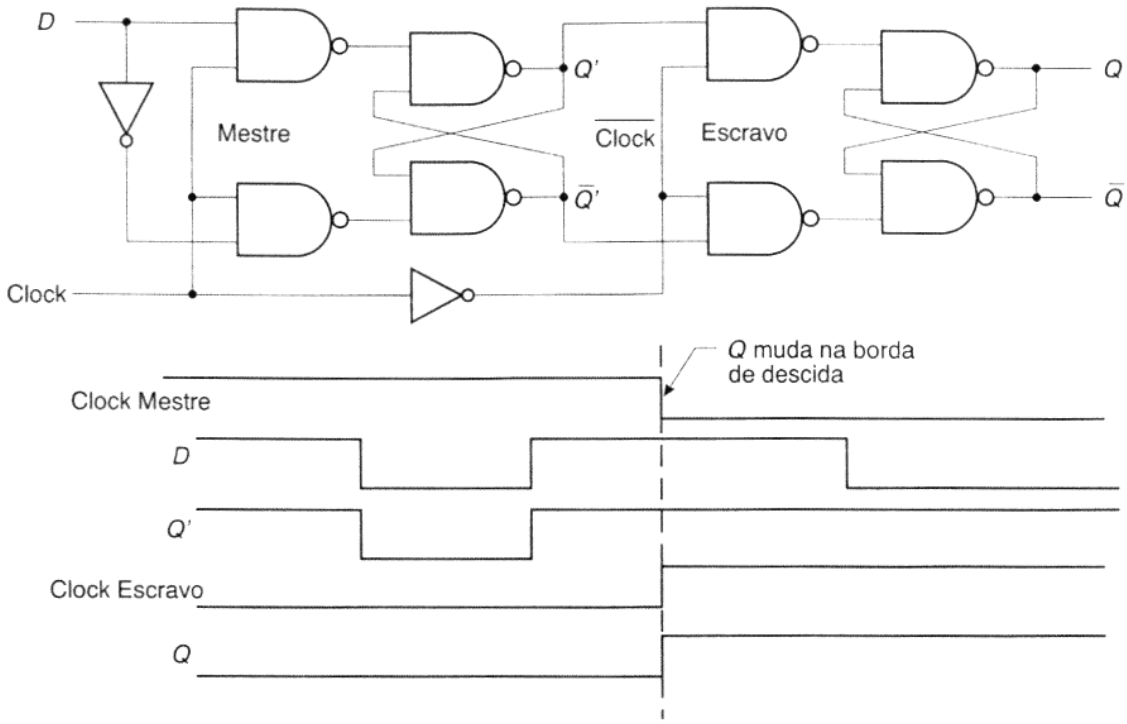


Figura 29: Processo de mudança de estado no *flip-flop* D MS da Figura 28.

- Na Figura 29, quando $\mathrm{CLOCK}=1$, a seção "Mestre" é habilitada. Nesta situação o valor lógico da entrada de dados D é transferido para a saída Q'. A seção "Escravo" permanece desabilitada porque $\overline{\mathrm{CLOCK}}=0$, o que mantém inalterado o valor lógico na saída Q.

- Quando ocorre a transição $\mathrm{CLOCK}=1\rightarrow\mathrm{CLOCK}=0$, a seção "Mestre" é desabilitada e as saídas Q' e $\overline{\mathrm{Q}'}$ mantêm inalterado seus valores lógicos. Nesta situação, a seção "Escravo" transfere os valores lógicos de Q' e $\overline{\mathrm{Q}'}$ respectivamente para Q e $\overline{\mathrm{Q}}$ porque $\overline{\mathrm{CLOCK}}=1$.

- Note que havendo uma mudança do valor lógico da entrada de dados D enquanto $\mathrm{CLOCK}=0$ nenhuma alteração ocorre porque a seção "Mestre" está desabilitada para esta situação.

$\Rightarrow$ Portanto, a saída Q (e $\overline{\mathrm{Q}}$) pode mudar seu valor lógico apenas no instante de tempo em que ocorre a borda de descida do *clock*, quando assume o valor lógico que está aplicado à entrada D neste instante. O valor lógico da entrada D permanece "memorizado" na saída Q até o instante em que ocorre a próxima borda de descida do *clock*.

- Se o inversor for trocado de posição conforme mostra a Figura 30, o *flip-flop* mudará de estado **na borda de subida** do *clock*:

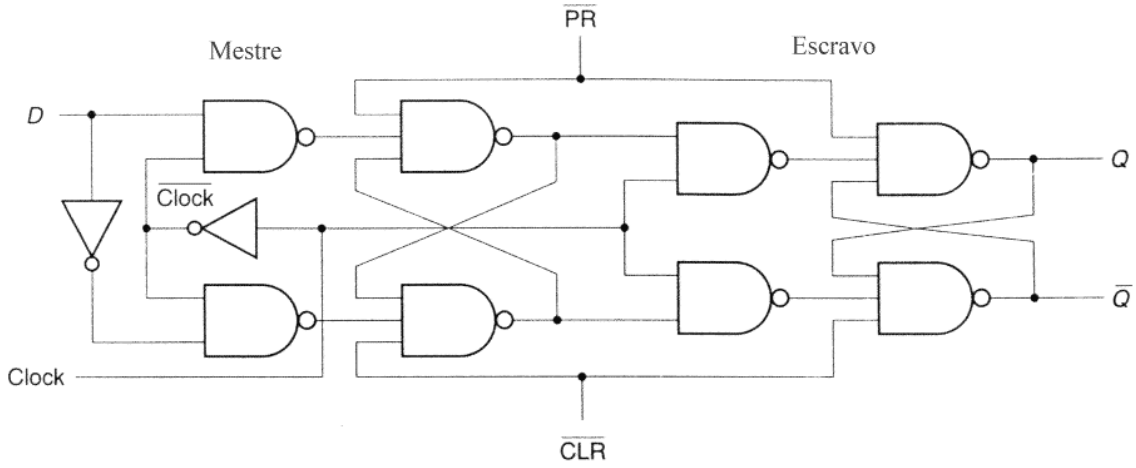


Figura 30: Diagrama lógico de um *flip-flop* D MS com mudança de estado na borda de subida do *clock*. As entradas $\overline{CLR}$ (*clear*–limpar) e $\overline{PR}$ (*preset*–"presetar") são entradas assíncronas que alteram o estado do *flip-flop*, independentemente do *clock*. Quando $\overline{PR} = 0$ o *flip-flop* é incondicionalmente colocado no estado SET ($Q = 1$ e $\overline{Q} = 0$) e quando $\overline{CLR} = 0$ o *flip-flop* é incondicionalmente colocado no estado RESET ($Q = 0$ e $\overline{Q} = 1$).

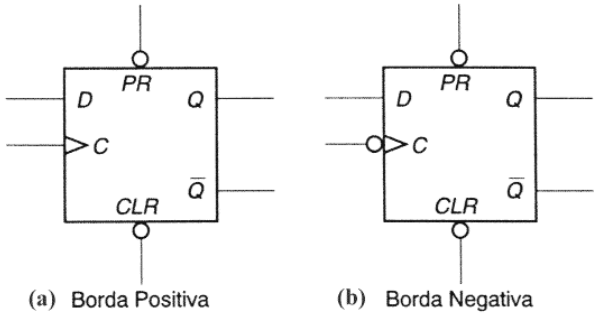


Figura 31: Símbolo lógico de um *flip-flop* D MS (a) com mudança de estado na borda de subida do *clock* e (b) com mudança de estado na borda de descida do *clock*.

$\overline{CLR}$	$\overline{PR}$	Clock	*D*	*Q*	$\overline{Q}$	
0	1	*X*	*X*	0	1	
1	0	*X*	*X*	1	0	
0	0	*X*	*X*	1	1	Condição Inválida
1	1	⎍	1	1	0	
1	1	⎍	0	0	1	

Figura 32: Tabela Verdade de um *flip-flop* D MS com mudança de estado na borda de subida do *clock*.

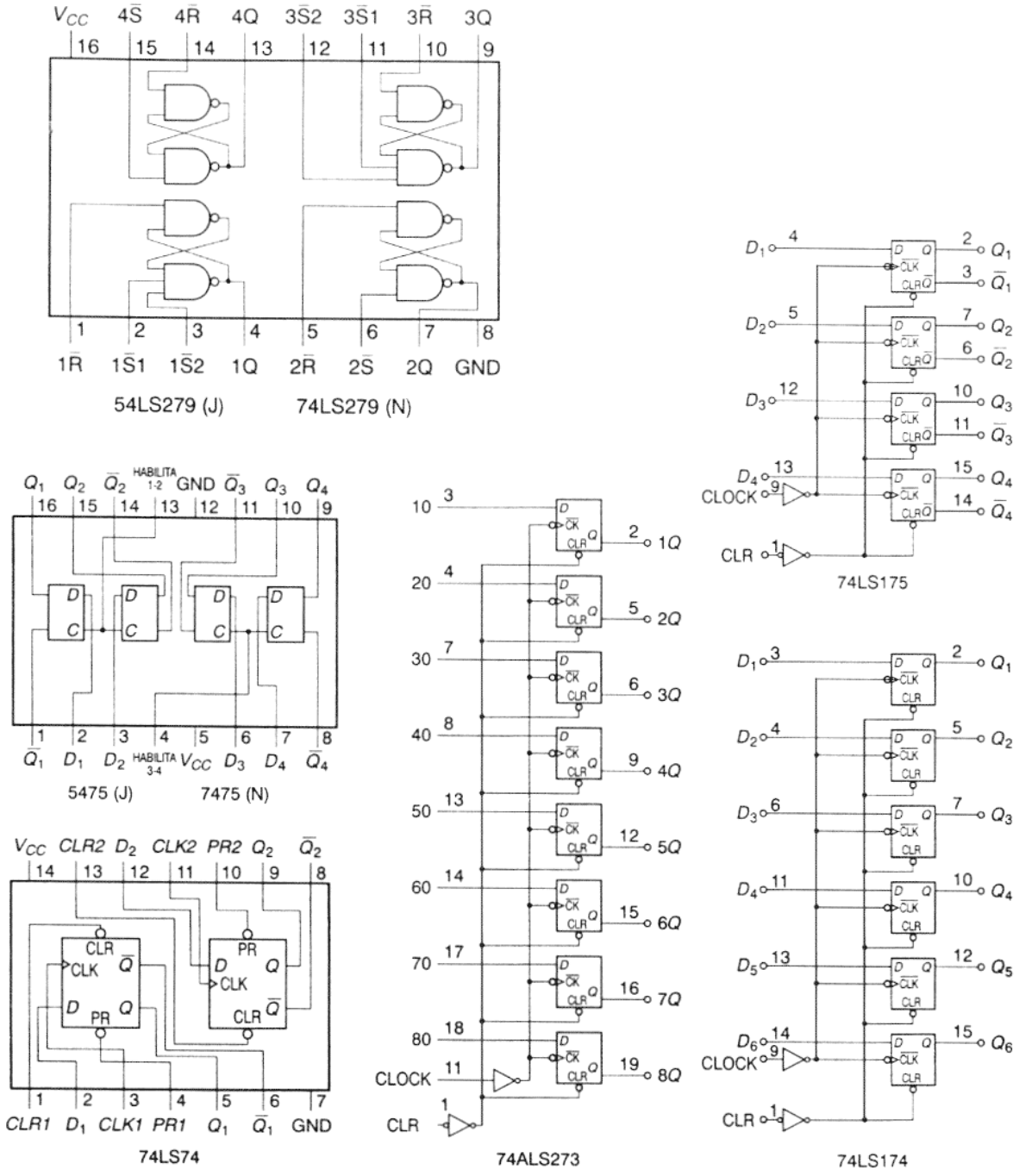


Figura 33: *Flip-flops* D comuns implementados em CIs TTL.

Exemplo 3: Determine as formas de onda nas saídas Q e $\overline{Q}$ do *flip-flop* D MS mostrado na Figura 34 a seguir.

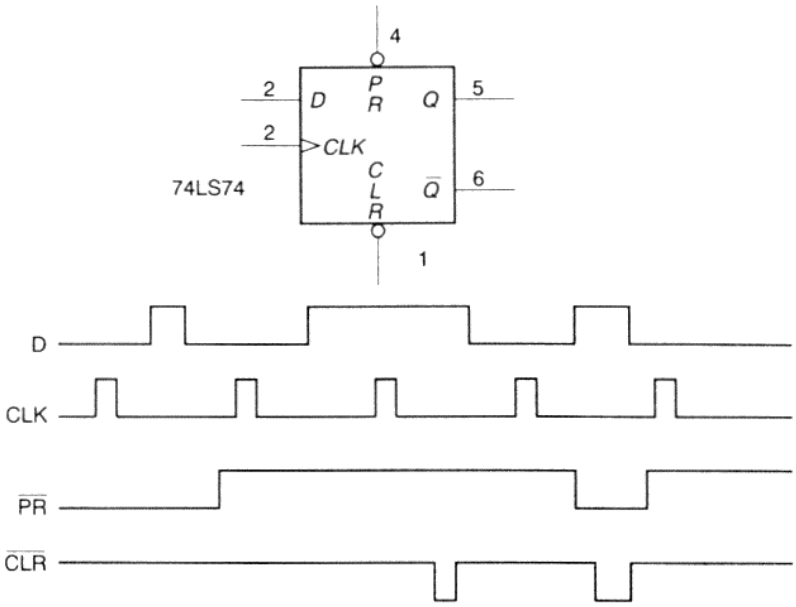


Figura 34

Solução:

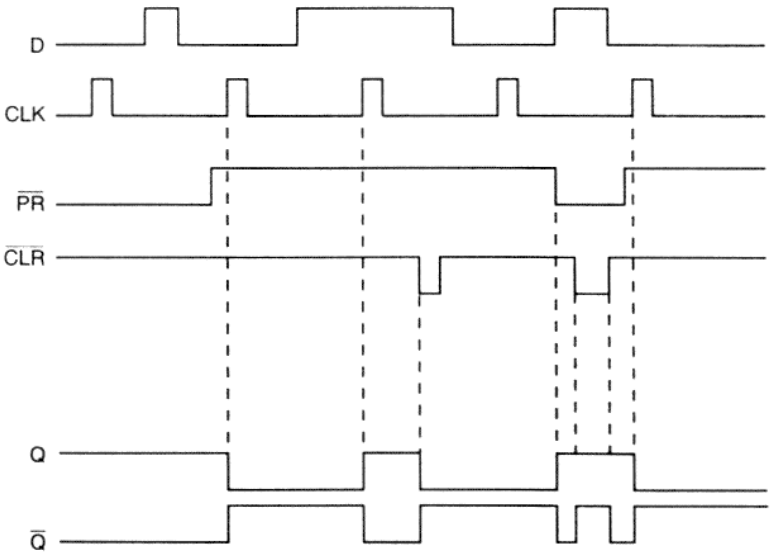


Figura 35

6 O *flip-flop* D Disparado pela Borda (*edge-triggered*)

● A Figura 36 mostra o diagrama lógico de um *flip-flop* D *edge-triggered* (*edge-triggered* = disparado pela borda do pulso de *clock*):

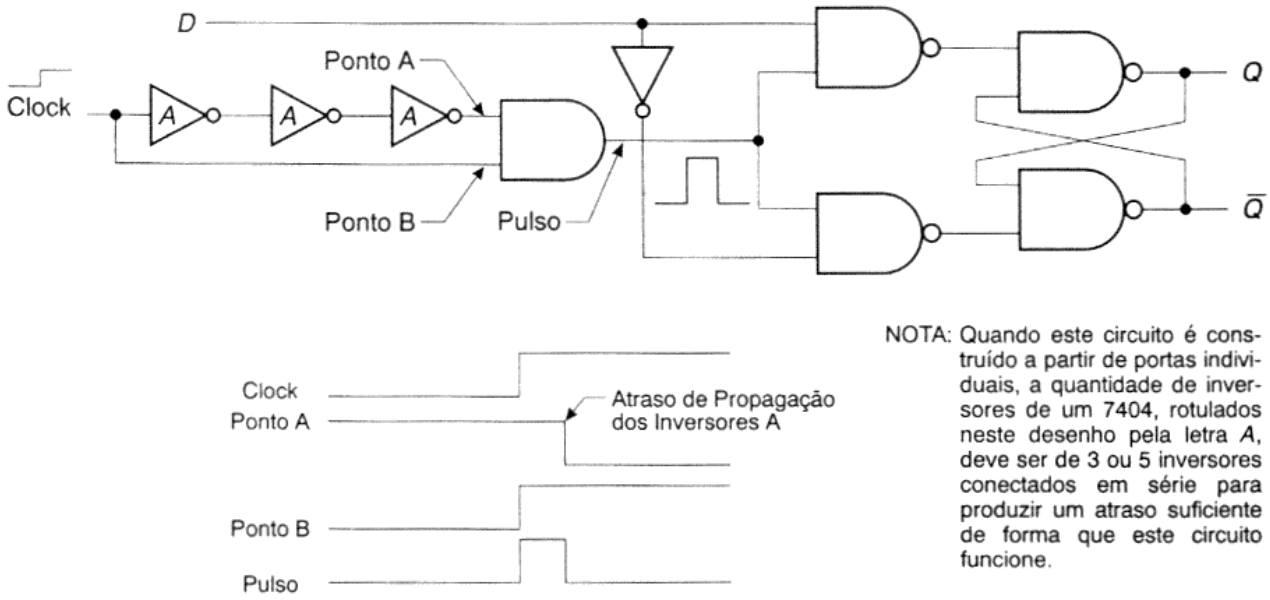


Figura 36: Diagrama lógico de um *flip-flop* D *edge-triggered* com mudança de estado na borda de subida do *clock*. Para obter a mudança de estado na borda de descida do *clock* basta acrescentar um inversor na entrada de *clock*.
Nota: Nos últimos anos os *flip-flops edge-triggered* vem gradativamente substituindo os *flip-flops master-slave*.

7 O *flip-flop* T MS (T – *toggle*)

● Quando ligamos a saída $\overline{Q}$ de um *flip-flop* D à entrada de dados D obtemos um *flip-flop* T (*toggle* – chavear seqüencialmente de modo alternado entre dois estados) , conforme mostrado na Figura 37:

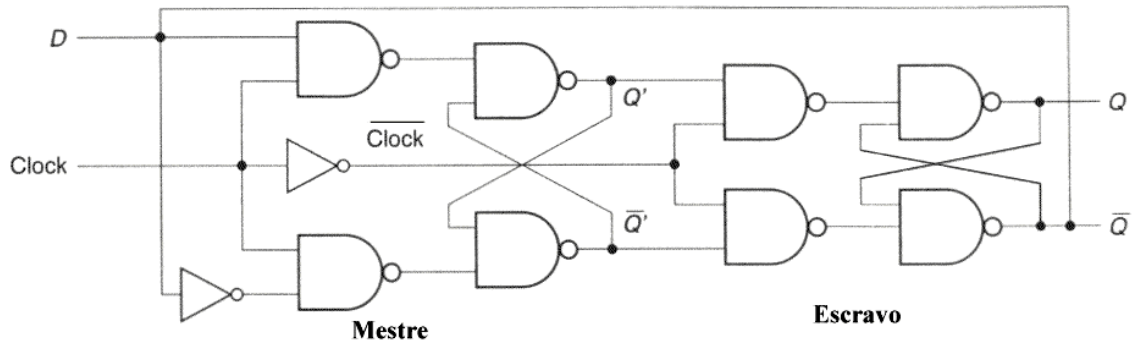


Figura 37: Diagrama lógico de um *flip-flop* T. Uma vez que $\overline{Q}$ recebe o inverso do valor lógico da entrada de dados D a cada instante em que ocorre a descida do *clock*, então o *flip-flop* alterna de estado exatamente nestes instantes conforme mostra a Figura 38.

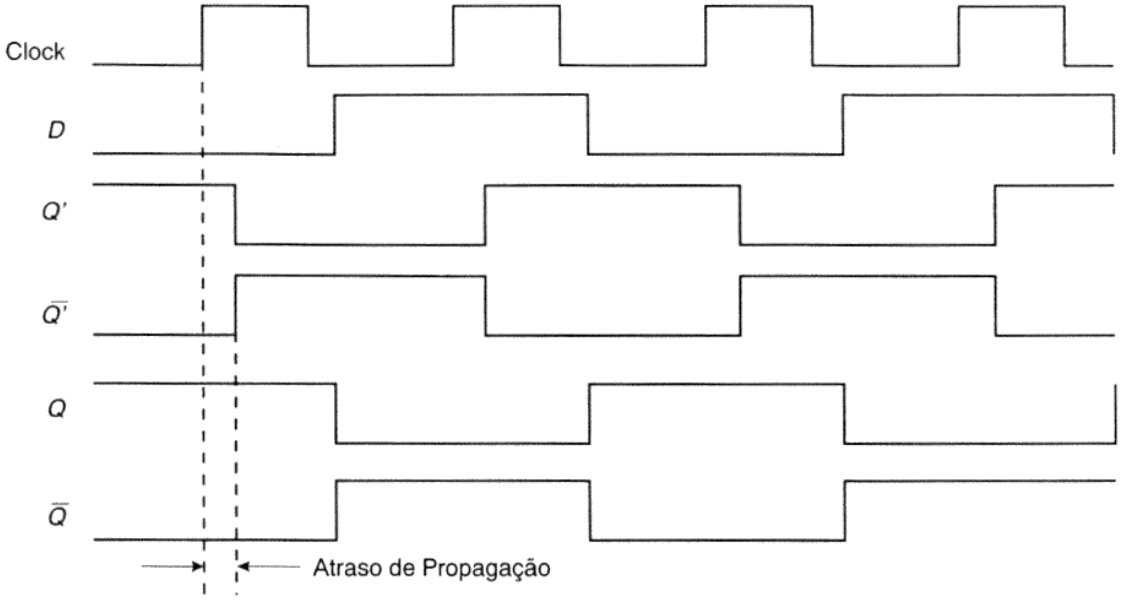


Figura 38: Formas de onda do *flip-flop* T mostrado na Figura 37. O atraso de propagação mostrado é originado nas portas NAND. Quando ocorre a descida do *clock* de 1→0, o "Mestre" é desabilitado primeiro, evitando que Q' e $\overline{Q'}$ mudem seu valor lógico. Alguns nanossegundos após, as saídas Q e $\overline{Q}$ mudam de valor lógico igualando-se respectivamente à Q' e $\overline{Q'}$. Com isto D iguala-se ao novo valor de $\overline{Q}$, mas o "Mestre" não muda de estado porque está desabilitado por $CLOCK = 0$. A mudança de estado só ocorrerá na próxima borda de descida do *clock*.

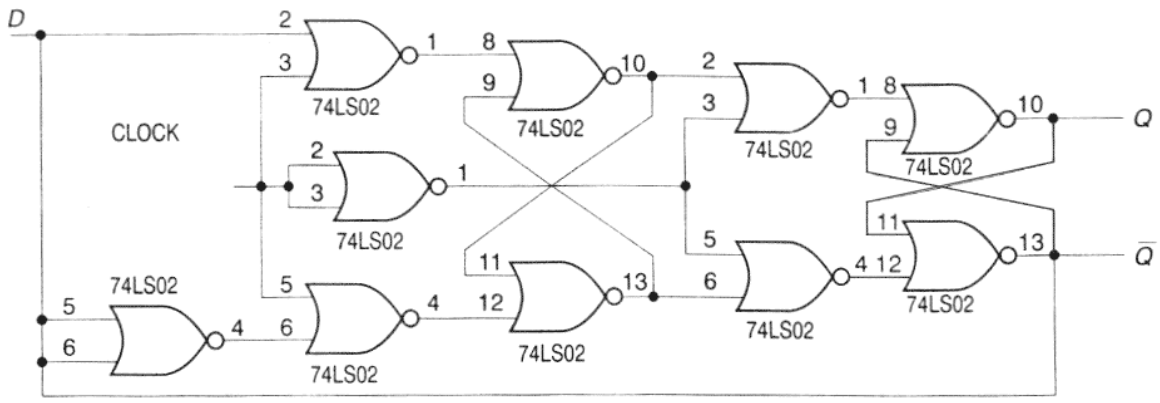
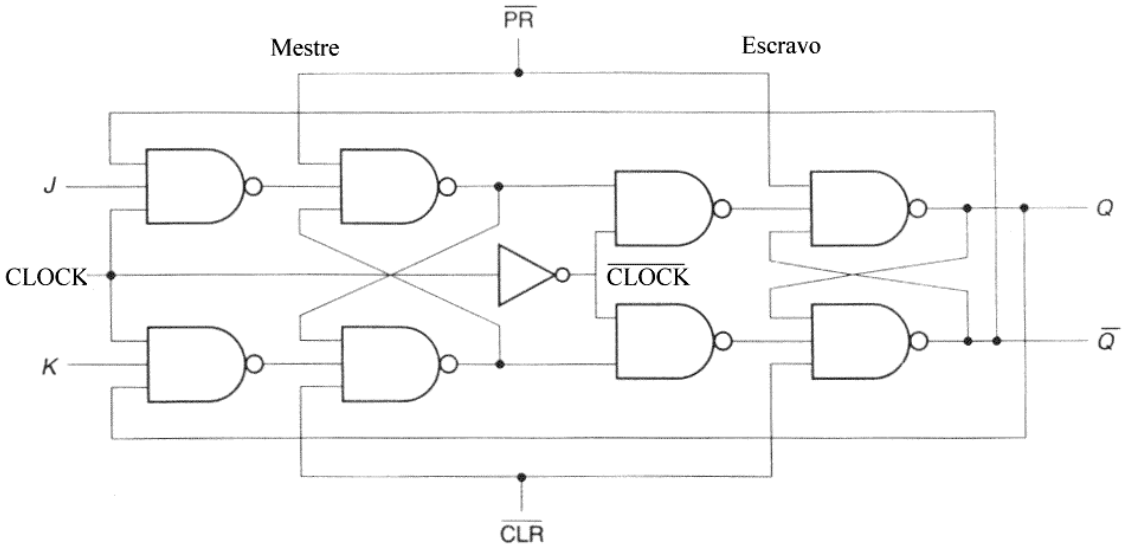


Figura 39: Diagrama lógico de um *flip-flop* T implementado com portas NOR.

8 O *flip-flop* JK

● A Figura 40 mostra o diagrama lógico de um *flip-flop* JK *master-slave* (existe também o JK *edge-triggered*):



$\overline{PR}$	$\overline{CLR}$	J	K	CLOCK	Q	$\overline{Q}$	
0	1	X	X	X	1	0	
1	0	X	X	X	0	1	
0	0	X	X	X	1	1	Condição Inválida
1	1	0	1	↓	0	1	
1	1	1	0	↓	1	0	
1	1	0	0	X	Q	$\overline{Q}$	Saída Inalterada
1	1	1	1	↓	Toggle		

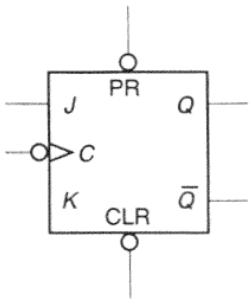


Figura 40: Diagrama lógico, Tabela Verdade e símbolo lógico de um *flip-flop* JK MS implementado com portas NAND. Note que a mudança de estado deste *flip-flop* JK ocorre na borda de descida do *clock*.

- Note da Figura 40 que se $J = K = 0$ então o "Mestre" é desabilitado, e, portanto, o "Escravo" não muda o valor lógico das saídas Q e $\overline{Q}$. Portanto o estado do *flip-flop* JK fica "memorizado" para esta situação.

- Se $J = \overline{K}$ então as saídas Q e $\overline{Q}$ recebem respectivamente os valores lógicos das entradas J e K no instante em que ocorre a borda de descida do *clock* (comporta-se semelhantemente a um *flip-flop* D).

- Se $J = K = 1$ então a saída Q controla a habilitação da porta NAND que recebe a entrada K e a saída $\overline{Q}$ controla a habilitação da porta NAND que recebe a entrada J. Isto faz com que a cada instante em que ocorre a descida do *clock* o *flip-flop* JK alterne de estado exatamente nestes instantes (comporta-se como um *flip-flop* T).

- As entradas assíncronas $\overline{CLR}$ e $\overline{PR}$ têm prioridade sobre todas as demais entradas.

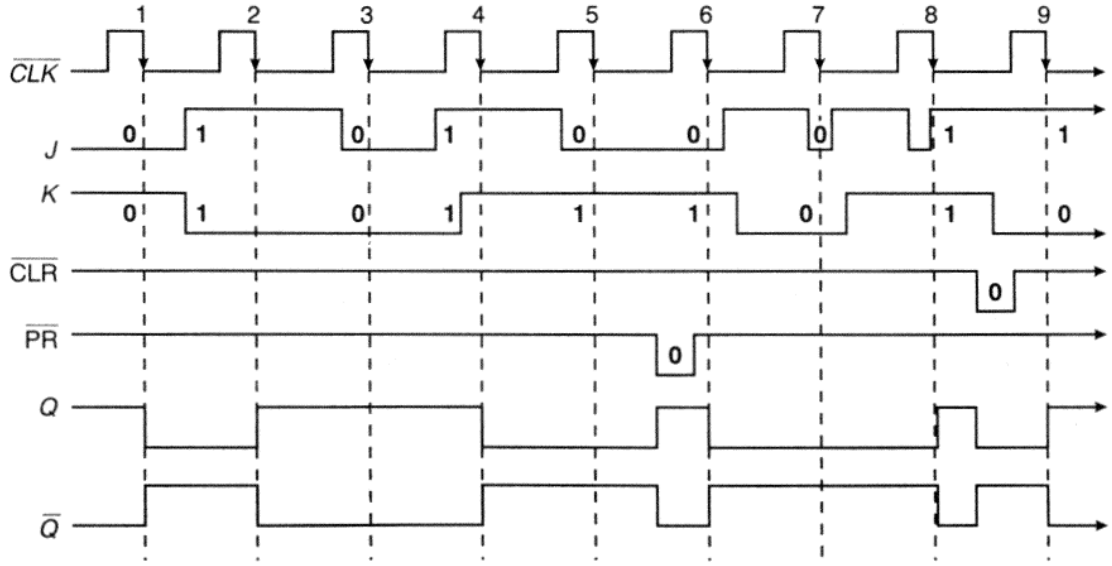


Figura 41: Exemplo de formas de onda para Q e $\overline{Q}$ obtidas na operação de um *flip-flop* JK MS, de acordo com a Tabela Verdade da Figura 40. Note a prioridade de ação das entradas assíncronas $\overline{CLR}$ e $\overline{PR}$. Note também que os valores lógicos de Q e $\overline{Q}$ são determinados pelos valores das entradas J e K antes da borda de descida do *clock* (exceto quando os valores lógicos de Q e $\overline{Q}$ são determinados por ação de $\overline{CLR}$ ou $\overline{PR}$).

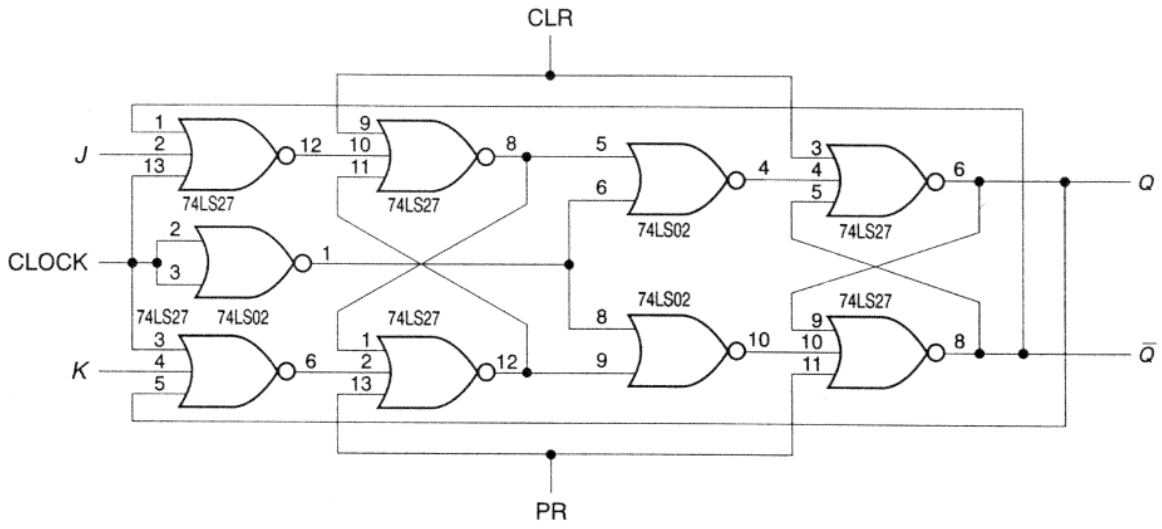


Figura 42: Diagrama lógico de um *flip-flop* JK MS implementado com portas NOR.

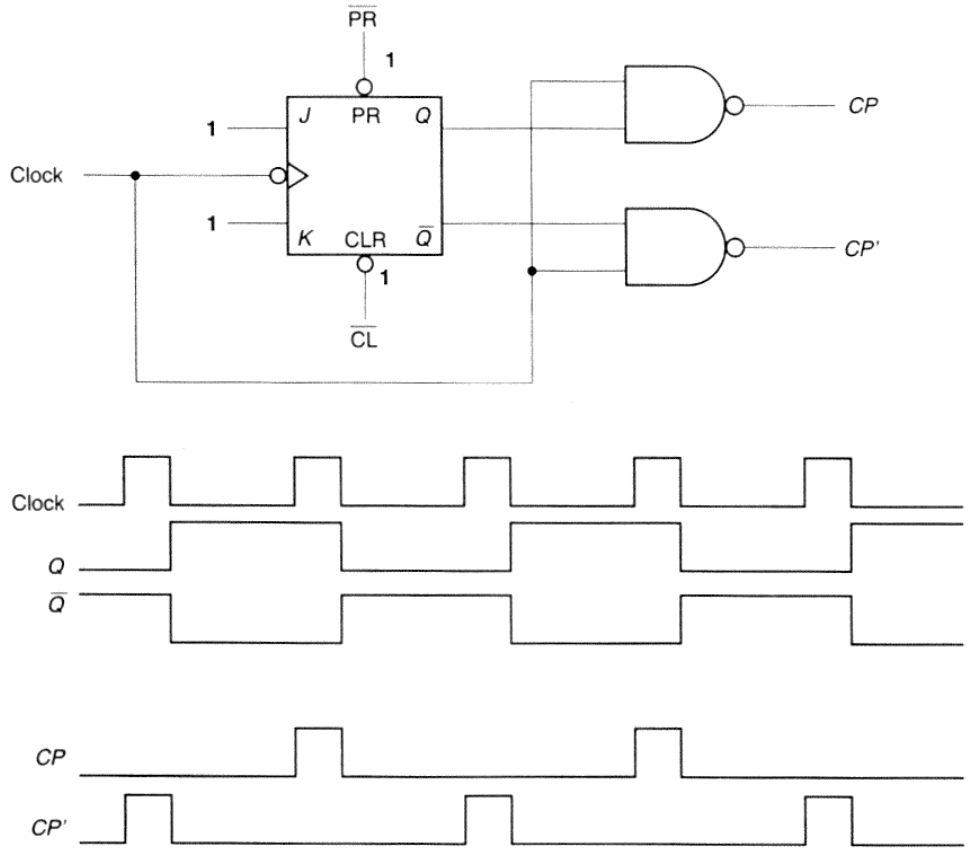


Figura 43: Exemplo de aplicação – divisor de freqüência ($\div 2$) usando um *flip-flop* JK. Note que o sinal CP tem a metade da freqüência do *clock*. Note também que o sinal CP' tem a mesma freqüência de CP mas é defasado de 180° do mesmo. Se não for necessário dois sinais defasados 180°, elimina-se uma das portas NAND.

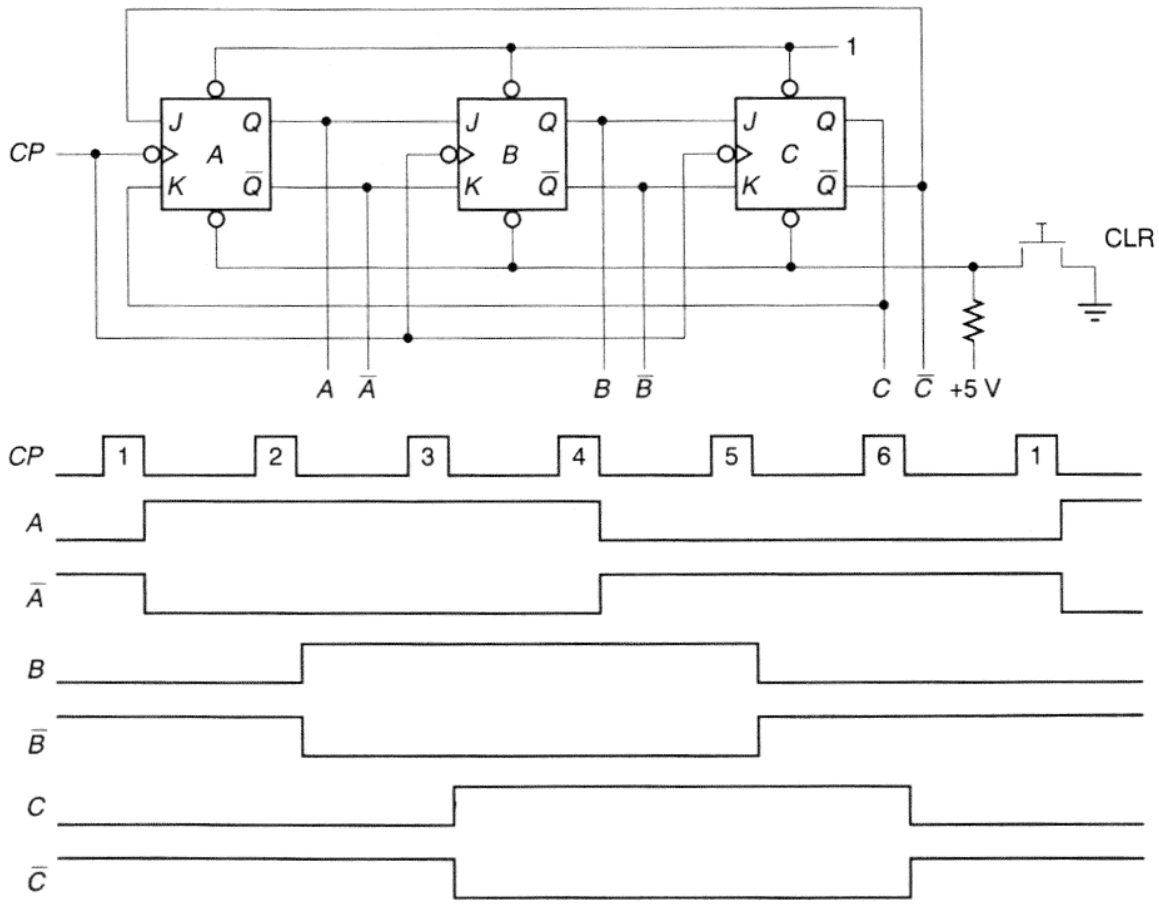


Figura 44: Exemplo de aplicação – divisor de freqüência ($\div 3$) usando 3 *flip-flops* JK. Note que as saídas Q e $\overline{Q}$ de cada *flip-flop* são interligadas com as entradas J e K do *flip-flop* seguinte, formando um anel fechado. Por causa disto, a cada borda de descida do *clock* as saídas Q e $\overline{Q}$ de cada *flip-flop* são transferidas para a saídas do *flip-flop* seguinte. Note também as saídas dos *flip-flops* estão defasadas de 120°.

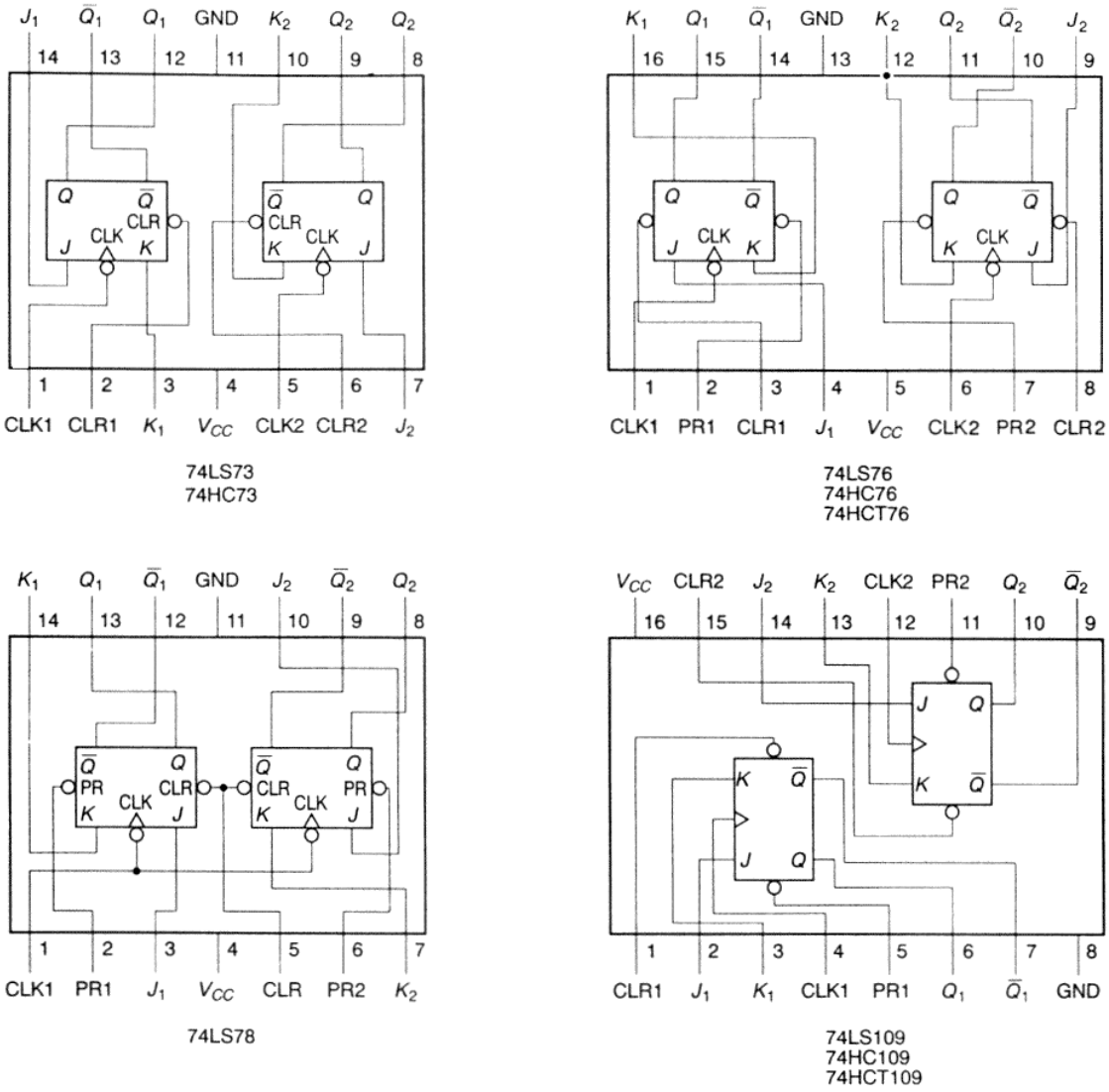


Figura 45: *Flip-flops* JK comuns implementados em CIs TTL.

9 Tempos de Comutação

9.1 Atraso de Propagação (*Propagation Delay Time*)

- É o intervalo de tempo requerido após a aplicação de um sinal de entrada para que ocorra a mudança resultante na saída. Os seguintes atrasos de propagação são relevantes na operação de um *flip-flop*:

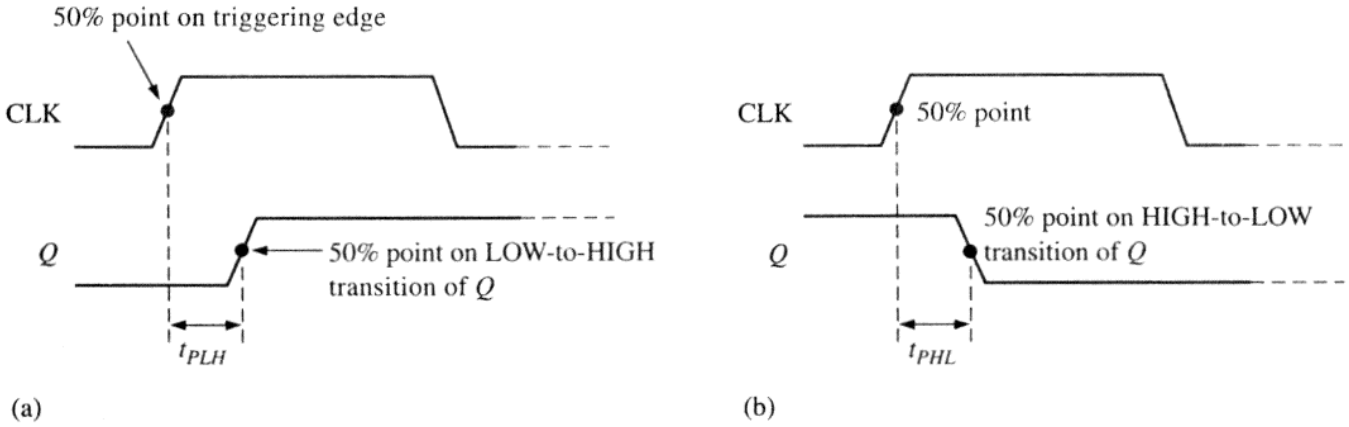


Figura 46: (a) Atraso de propagação t_{PLH} medido a partir da borda de disparo (*triggering edge*) do pulso de *clock* até a transição LOW→HIGH na saída Q. (b) Atraso de propagação t_{PHL} medido a partir da borda de disparo do pulso de *clock* até a transição HIGH → LOW na saída Q.

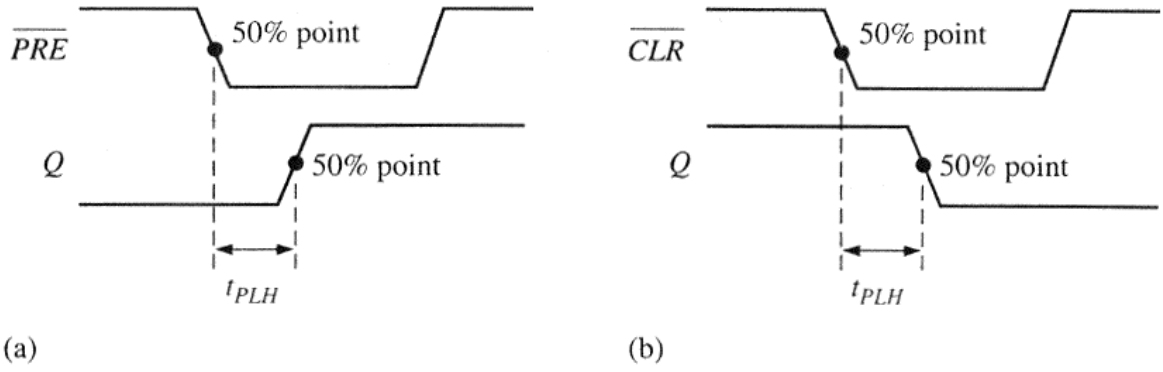


Figura 47: (a) Atraso de propagação t_{PLH} medido a partir do sinal $\overline{\mathrm{PR}}$ até a transição LOW→HIGH na saída Q. (b) Atraso de propagação t_{PHL} medido a partir do sinal $\overline{\mathrm{CLR}}$ até a transição HIGH → LOW na saída Q.

9.2 *Setup Time*

- É o mínimo intervalo de tempo em que os sinais aplicados nas entradas (D, J, K, SET ou RESET) devem ser mantidos constantes **antes** da transição de estado imposta pela borda de disparo (*triggering edge*) do pulso de *clock* para que a transição ocorra de maneira confiável:

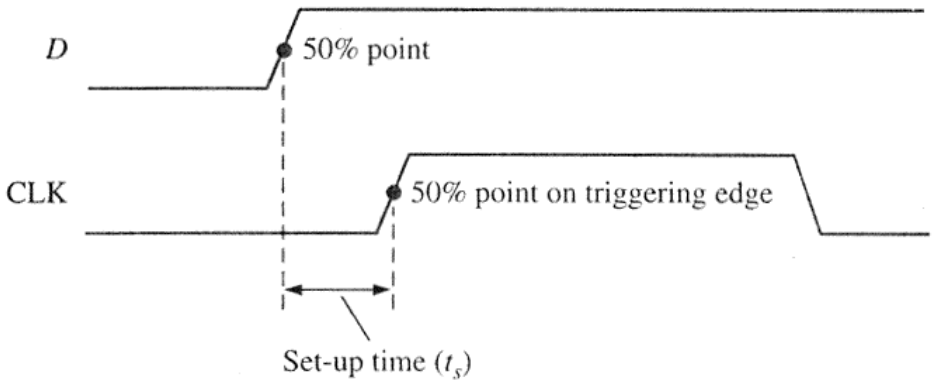


Figura 48: *Setup Time* (t_s) para um *flip-flop* D.

9.3 *Hold Time*

- É o mínimo intervalo de tempo em que os sinais aplicados nas entradas (D, J, K, SET ou RESET) devem ser mantidos constantes **depois** da transição de estado imposta pela borda de disparo (*triggering edge*) do pulso de *clock* para que a transição possa ser completada de maneira confiável:

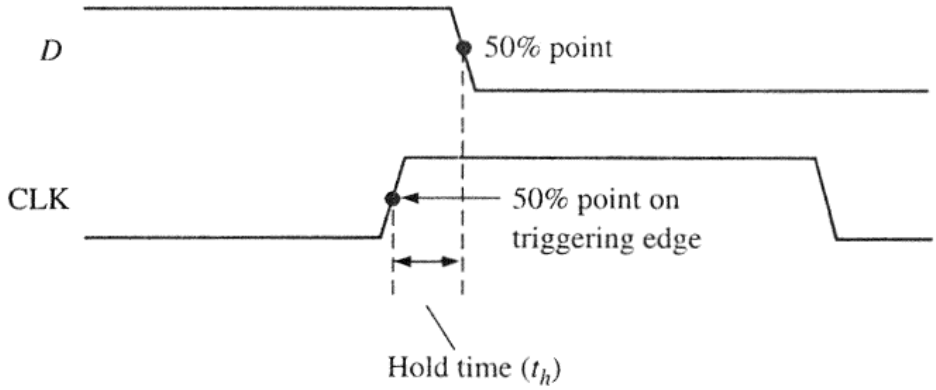


Figura 49: *Hold Time* (t_h) para um *flip-flop* D.

Parameters	CMOS		TTL	
(Times in ns)	**74HC74A**	**74AHC74**	**74LS74A**	**74F74**
t_{PHL} (CLK to Q)	17	4.6	40	6.8
t_{PLH} (CLK to Q)	17	4.6	25	8.0
t_{PHL} ($\overline{CLR}$ to Q)	18	4.8	40	9.0
t_{PLH} ($\overline{PRE}$ to Q)	18	4.8	25	6.1
t_s (set-up time)	14	5.0	20	2.0
t_h (hold time)	3.0	0.5	5	1.0
t_W (CLK HIGH)	10	5.0	25	4.0
t_W (CLK LOW)	10	5.0	25	5.0
t_W ($\overline{CLR}/\overline{PRE}$)	10	5.0	25	4.0
f_{max} (MHz)	35	170	25	100
Power (mW), quiescent	0.012	1.1		
Power (mW), 50% duty cycle			44	88

Figura 50: Tempos de Comutação típicos de um *flip-flop* D (CI 7474 CMOS e/ou TTL).