tradas dos SCs que correspondem ao binário B têm agora o complemento de B. Observemos que V = 1 coloca C_i do primeiro SC em 1, o que equivale a somar 1 ao resultado final. Por exemplo:

S = A + (complemento 1 de B) + 1 = A + complemento 2 de B

Como na representação de binário com sinal o complemento 2 corresponde ao negativo de um binário positivo, concluímos que o circuito da figura 3.34 pode ser um circuito somador ou subtrator, dependendo da variável de controle V.

Capítulo 4

Circuitos sequenciais

Os circuitos lógicos combinacionais permitem funções como decodificação, soma e subtração, comparação e muitas outras. Entretanto, funções mais avançadas (que dependem do tempo, memorização de dados, sequência de operações etc.) não podem ser implementadas com o mesmo princípio. Nesse caso, devemos recorrer ao projeto de circuitos lógicos sequenciais.

Em um circuito sequencial, os valores das saídas em determinado instante dependem não só da combinação das variáveis de entrada, mas também do valor anterior, isto é, do valor que a saída tinha antes da aplicação da nova combinação de valores nas entradas. Para isso, é necessário utilizar dispositivos de memória elementares capazes de armazenar as variáveis de saída internamente a cada transição de estado (figura 4.1).

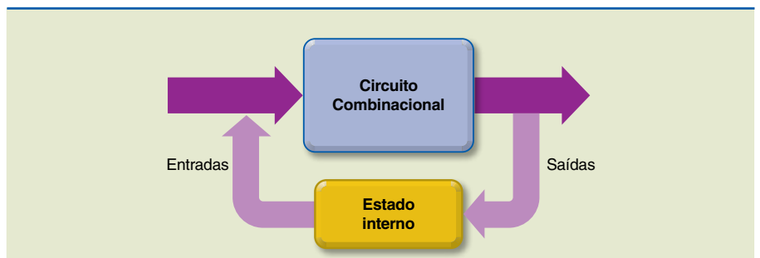


Figura 4.1
Dispositivos de memória elementar.

4.1 Elementos de memória

O *latch* RS é um elemento de memória simples com capacidade de armazenamento temporário de um bit. Esse dispositivo consiste em duas portas NOR acopladas por realimentações cruzadas (figura 4.2).

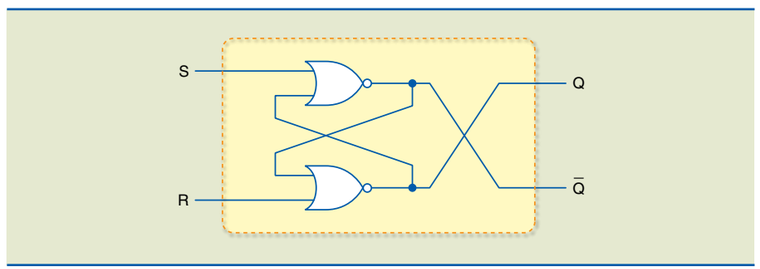


Figura 4.2
Detalhe interno do *latch* RS mostrando duas portas NOR acopladas por realimentações cruzadas.

Analisando a figura 4.2, podemos notar que as entradas S (*set*) e R (*reset*) ficam normalmente em nível "0", sendo ambas ativas em nível lógico "1". Fazendo S = 1, obtém-se Q = 1. Esse nível é mantido após a retirada do nível "1" da entrada S e permanece até que seja aplicado nível "1" na entrada R. Fazendo R = 1, obtém-se Q = 0. Esse nível é mantido após a retirada do nível "1" da entrada R e permanece até que seja aplicado nível "1" na entrada S.

A tabela verdade referente ao *latch* RS (figura 4.3) considera as entradas ativas em nível lógico alto.

Tabela verdade

Reset	Set	Q	/Q
0	0	mantém	
0	1	1	0
1	0	0	1
1	1	inválido	

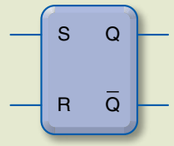


Figura 4.3
Representação do *latch* RS mostrando somente as entradas e saídas e a tabela verdade correspondente.

Analisando a tabela, podemos notar que S = 1 e R = 1 é inválido. Isso acontece porque:

- Nesse caso particular, as duas saídas Q e Q' seriam iguais a "0", o que implicaria de imediato a inconsistência com a teoria das saídas Q e Q'.
- Outro ponto crítico ocorre quando passamos desse estado para S = 0 e R = 0. Nesse caso, seguindo a tabela verdade e o comportamento do *latch*, a saída deveria permanecer inalterada, o que não acontece, gerando um estado indefinido para Qn + 1 e Q'n + 1.

Devido a essa ambiguidade, a condição S = 1 e R = 1 não é usada para *latch* RS.

O circuito do *latch* RS com portas NAND é mostrado na figura 4.4.

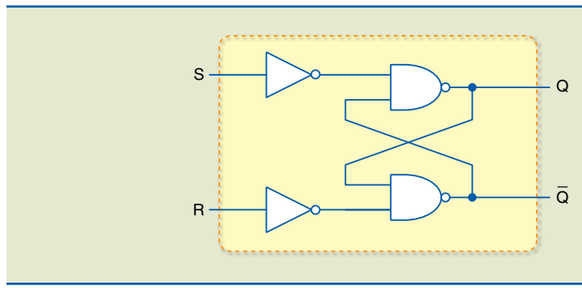
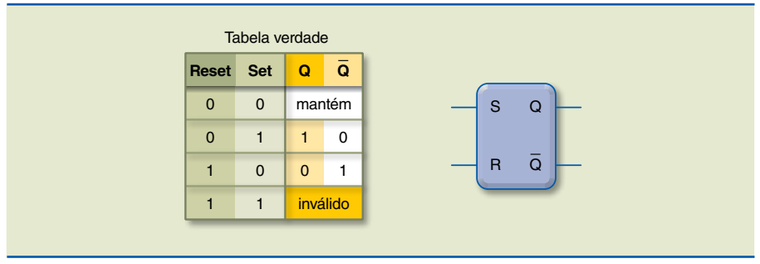


Figura 4.4
Detalhe interno do circuito do *latch* RS com portas NAND.

O circuito da figura 4.4 é equivalente ao apresentado no item anterior, portanto sua tabela verdade e símbolo lógico não se alteram (figura 4.5).

Figura 4.5
Representação do *latch* RS mostrando somente as entradas e saídas e tabela verdade correspondente.



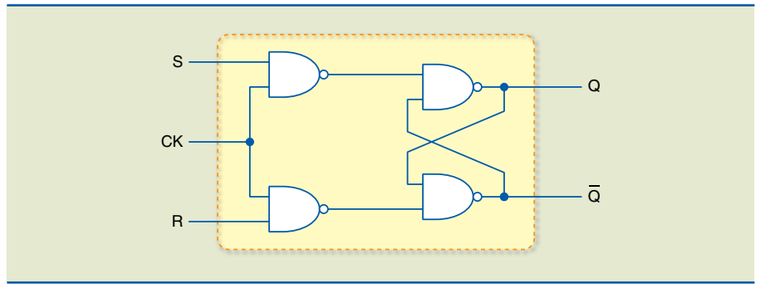
Analisando a tabela, podemos notar que S = 1 e R = 1 é inválido. Isso acontece porque:

- Nesse caso particular, as duas saídas Q e Q' seriam iguais a "1".
- Quando passamos desse estado para S = 0 e R = 0, estamos novamente gerando um estado indefinido para Qn + 1 e Q'n + 1.

Um *latch* controlado possui uma entrada *enable* que diz quando o *latch* poderá armazenar um valor. Caso *enable* = 0, o *latch* permanece em seu estado anterior, mantendo armazenado o bit. Somente quando *enable* = 1 o *latch* funcionará como antes.

A entrada *enable* também pode ser denominada *clock* (CK), ou relógio, quando ela receber um sinal de sincronismo, por isso em alguns diagramas utiliza-se a notação "CK" (figura 4.6).

Figura 4.6
Identificação das entradas S, R e ***clock*** em um *latch*.



A tabela verdade a seguir demonstra as condições das saídas, considerando as entradas R, S e *clock*.

R	S	Relógio	Q	$\bar{Q}$
X	X	0	mantém	
0	0	1	mantém	
0	1	1	1	0
1	0	1	0	1
1	1	1	erro lógico	

Observe que a condição de ambiguidade (ou erro lógico) ainda existe quando S = R = 1.

Um *latch* controlado (tipo D) é implementado colocando-se um inversor entre os terminais S e R de um *latch* RS. Nessa configuração, impede-se que as variáveis de entrada assumam valores idênticos, isto é, S = R = 0 ou S = R = 1. Assim, a entrada D passa a ser única, e os pontos correspondentes a S e R, a assumir sempre valores distintos (figura 4.7):

- Se D = 1, então S = 1 e R = 0.
- Se D = 0, então S = 0 e R = 1.

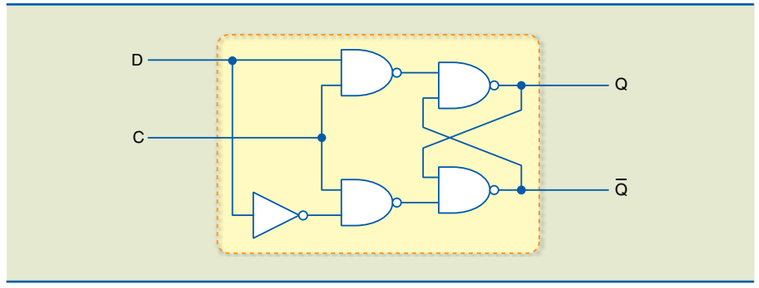


Figura 4.7
Detalhe interno de um *latch* para D = 0 e D = 1.

Observe que o problema da inconsistência foi eliminado, uma vez que é impossível aplicar sinais iguais nas entradas S e R. A figura 4.8 e sua respectiva tabela verdade possibilitam uma análise dessa configuração.

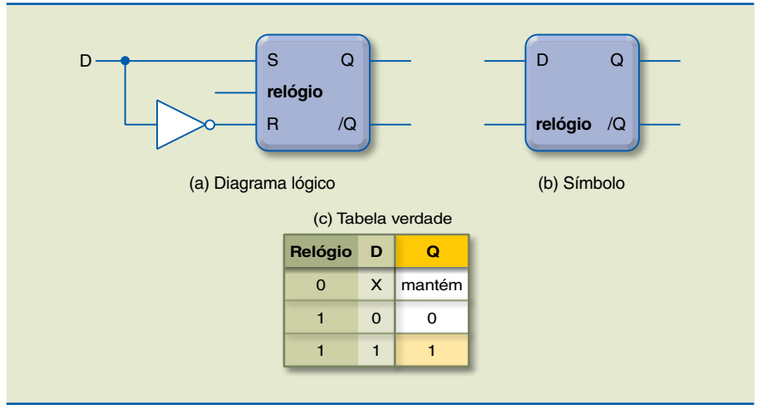


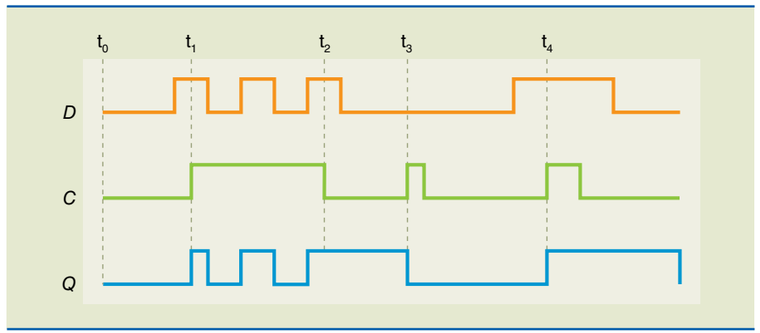
Figura 4.8
Representação do ***latch*** RS: (a) diagrama lógico, (b) símbolo e em (c) tabela verdade.

Analisando a tabela verdade, podemos entender o funcionamento, pois:

- se *enable* = 0, o *latch* permanece no estado anterior;
- se *enable* = 1 e D = 1, temos S = 1 e R = 0; portanto, a saída Q será Q = D (Q = 1);
- se *enable* = 1 e **D** = 0, temos S = 0 e R = 1; portanto, a saída Q será Q = D (Q = 0).

Concluindo, se *enable* = 1, a saída Q acompanha a entrada **D** e, se *enable* = 0, a saída do *latch* permanece inalterada, ou seja, mantém o estado anterior.

Figura 4.9
Formas de onda dos sinais para um *latch* tipo D.



As formas de onda dos sinais para um *latch* **tipo D** são apresentadas na figura 4.9, em que:

- D é a entrada de dados;
- C, o sinal de habilitação ou *clock*;
- Q, a saída do *latch*.

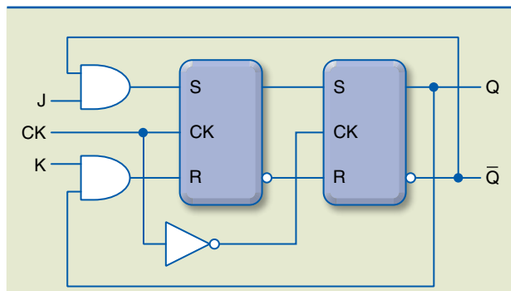
Os circuitos *latches* R, S e D apresentados anteriormente são sensíveis ao nível do sinal aplicado em sua entrada de habilitação (*enable*).

Agora, analisaremos dispositivos que dispõem de entradas de sincronismo sensíveis às **transições de nível lógico**, de "0" para "1" ou de "1" para "0". Esses dispositivos são conhecidos pela terminologia "disparados por borda" (do sinal de relógio) e podem ser de dois tipos:

- Disparados por borda de subida (transição positiva do sinal de *clock*): sensíveis às transições de nível lógico do sinal de *clock*, de "0" para "1".
- Disparados por borda de descida (transição negativa do sinal de *clock*): sensíveis às transições de nível lógico do sinal de *clock*, de "1" para "0".

Vamos iniciar analisando o ***flip-flop* J-K mestre-escravo**. Esse dispositivo possui duas entradas de dados (J e K) e tem como característica principal seus dois estágios internos, denominados mestre e escravo (figura 4.10) com a tabela verdade correspondente.

Figura 4.10
Flip-flop J-K (mestre-escravo) e tabela verdade correspondente.



J	D	Estado futuro Q futuro
0	0	Q
0	1	0
1	0	1
1	1	Q̄

A figura 4.11 mostra detalhes das ligações internas do circuito.

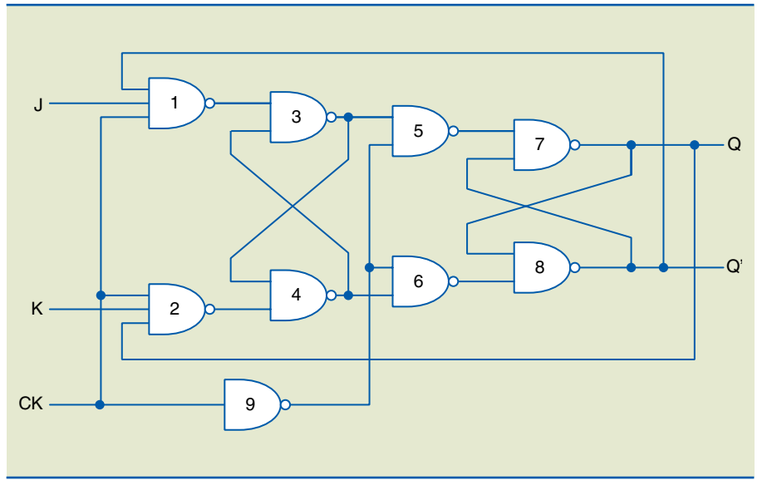


Figura 4.11
Detalhe interno de um *flip-flop J-K* (mestre-escravo).

Analisando a figura 4.11, podemos notar que:

- Se J = 0 e K = 0, as portas **1** e **2** estarão desabilitadas; portanto, após a aplicação do pulso de *clock*, o *flip-flop* não mudará de estado.

- Se J = 1 e K = 0 e Q = 0, a porta **1** habilitará (J = 1 **e** Q' = 1) e a porta **2** desabilitará (K = 0 **e** Q = 0); portanto, após a aplicação do pulso de *clock*, o estado de saída Q mudará para Q = 1.

- Se J = 1 e K = 0 e Q = 1, a porta **1** desabilitará (J = 1 **e** Q' = 0) e a porta **2** desabilitará (K = 0 **e** Q = 1); portanto, após a aplicação do pulso de *clock*, o estado de saída permanecerá inalterado (Q = 1).

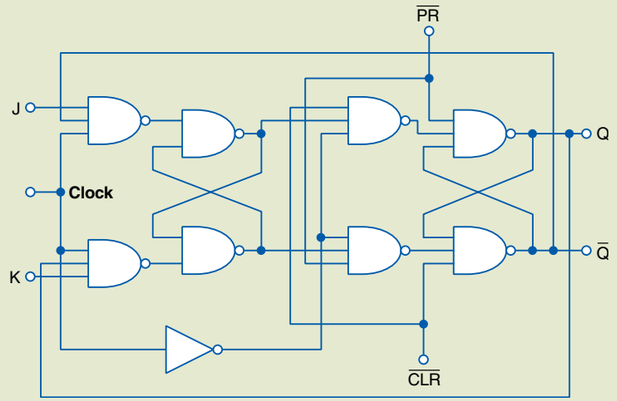
- Se J = 0 e K = 1 e Q = 0, a porta **1** desabilitará (J = 0 **e** Q' = 1) e a porta **2** desabilitará (K = 1 **e** Q = 0); portanto, após a aplicação do pulso de *clock*, o estado de saída permanecerá inalterado (Q = 0).

- Se J = 0 e K = 1 e Q = 1, a porta **1** desabilitará (J = 0 **e** Q' = 0) e a porta **2** habilitará (K = 1 **e** Q = 1); portanto, após a aplicação do pulso de *clock*, o estado de saída Q mudará para Q = 0.

- Se J = 1 e K = 1, para J = K = 1, a cada ciclo de *clock* o estado do *flip-flop* J-K se complementa; portanto, após a aplicação do sinal de *clock*, teremos: se Q = 0, a saída Q mudará para Q = 1; se Q = 1, a saída Q mudará para Q = 0.

Podemos também incluir as entradas de *preset* e *clear* nesse circuito, que passa a ter a configuração da figura 4.12. A tabela verdade inclui as entradas de *preset* (PR) e *clear* (CLR).

Figura 4.12
Detalhe interno de um *flip-flop* J-K (mestre-escravo) com as entradas *clear* e *preset* e a tabela verdade correspondente.



CLR	PR	J	K	CLK	Qn ÷ 1	Q̄n ÷ 1
		X	X	X	1	1
	1	X	X	X		1
1		X	X	X	1	
1	1			↓	Qn	Q̄n
1	1		1	↓		1
1	1	1		↓	1	
1	1	1	1	↓	Q̄n	Qn

O exemplo da figura 4.12 corresponde a um *flip-flop* J-K mestre-escravo sensível à transição negativa do sinal de relógio com entradas de *preset* e *clear* inversoras.

A figura 4.13 apresenta o circuito de *preset* e *clear* e a tabela verdade correspondente.

Figura 4.13
Configuração do *flip-flop* J-K mestre-escravo com entradas *clear* e *preset* e tabela verdade resumida.

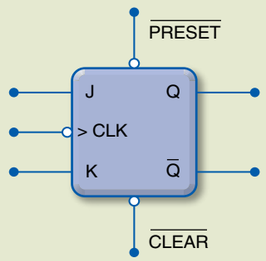


Tabela verdade

PRESET	CLEAR	Resposta do FF
1	1	Operação com clock*
0	1	Q = 1
1	0	Q = 0
0	0	Não usada

*Q irá responder a J, K e CLK

Existem outras configurações de entradas, que variam conforme o tipo de CI e o fabricante, tais como exemplificadas na figura 4.14.

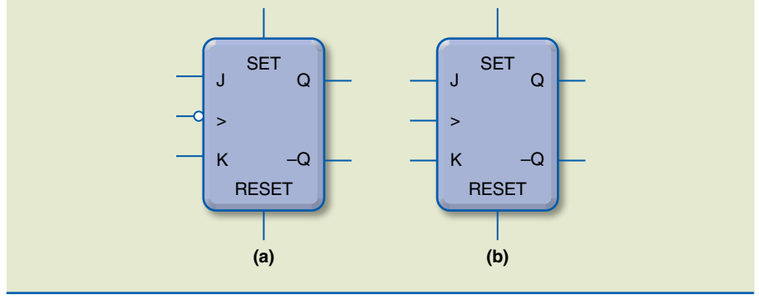


Figura 4.14
(a) *Flip-flop* J-K sensível à borda de descida e (b) *flip-flop* J-K sensível à borda de subida.

As figuras 4.15 e 4.16 apresentam dois exemplos de circuito integrado com dois *flip-flops* J-K: um da família CMOS e outro da TTL e suas tabelas verdade.

Figura 4.15
CI 4027B com dois *flip-flops* J-K sensíveis à borda de subida com entradas de *clear* e *preset* e a tabela verdade correspondente.

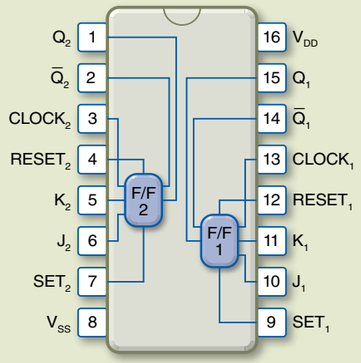


Tabela verdade

Entradas					Saídas	
CLEAR	PRESET	CLOCK	J	K	Q	Q̄
1	0	X	X	X	0	1
0	1	X	X	X	1	0
1	1	X	X	X	1	1
0	0	↑	0	0	Q_0	$\bar{Q}_0$
0	0	↑	1	0	1	0
0	0	↑	0	1	0	1
0	0	↑	1	1	TOGGLE	
0	0	0	X	X	Q_0	$\bar{Q}_0$
0	0	1	X	X	Q_0	$\bar{Q}_0$

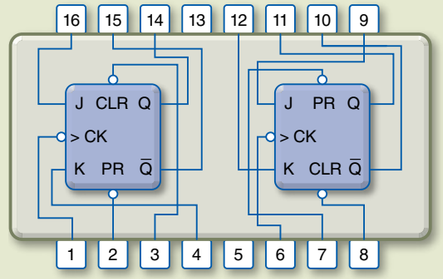


Tabela verdade

Entradas					Saídas	
CLEAR	PRESET	CLOCK	J	K	Q	Q̄
0	1	X	X	X	0	1
1	0	X	X	X	1	0
0	0	X	X	X	1	1
1	1	↓	0	0	Q_0	$\bar{Q}_0$
1	1	↓	1	0	1	0
1	1	↓	0	1	0	1
1	1	↓	1	1	TOGGLE	
1	1	1	X	X	Q_0	$\bar{Q}_0$
1	1	0	X	X	Q_0	$\bar{Q}_0$

Figura 4.16 CI 7476 com dois *flip-flops* J-K sensíveis à borda de descida e tabela verdade correspondente.

Implementação de um flip-flop D a partir do J-K

Um *flip-flop* tipo D sensível à borda pode ser obtido com um inversor entre as entradas J e K, como se pode observar na figura 4.17. Nesse tipo de *flip-flop*, a saída Q assume o nível lógico presente na entrada D toda vez que ocorre transição do sinal de *clock* (nesse exemplo, as transições de estado ocorrem no instante de subida do sinal de *clock*, conforme ilustram os gráficos).

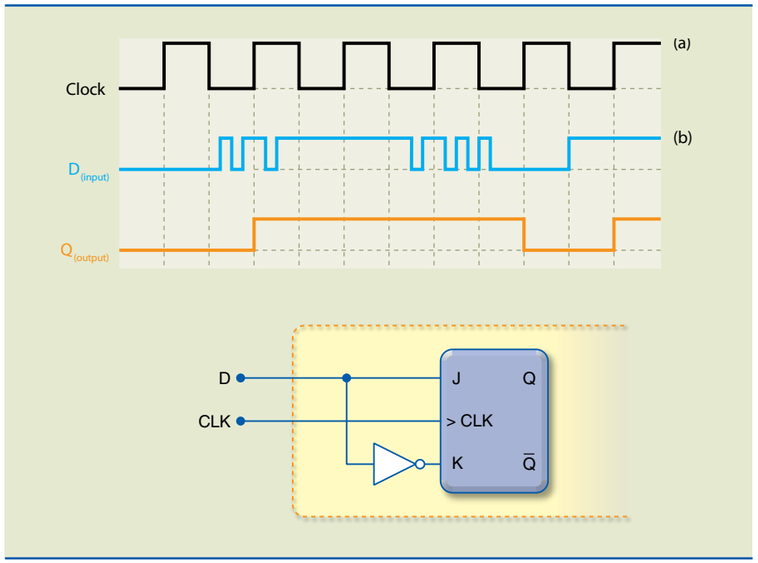


Figura 4.17
(a) *Flip-flop* tipo D a partir do J-K e (b) as formas de onda da entrada e da saída em função do *clock*.

Exemplos de circuitos integrados de *flip-flops* tipo D CMOS e TTL

4013 – Dois flip-flops tipo D sensíveis à borda de subida com entradas de preset e clear

A figura 4.18 apresenta esse dispositivo, e a tabela verdade correspondente.

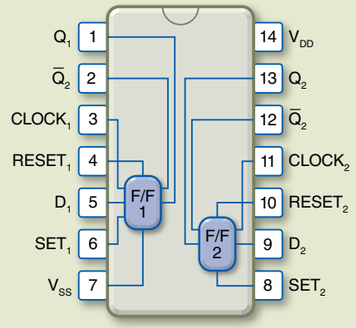


Tabela verdade

Entradas				Saídas	
CLEAR	PRESET	CLOCK	D	Q	Q̄
0	1	X	X	1	0
1	0	X	X	0	1
1	1	X	X	1	1
0	0	↑	0	0	1
0	0	↑	1	1	1
0	0	0	X	Q_0	$\bar{Q}_0$
0	0	1	X	Q_0	$\bar{Q}_0$

Figura 4.18
Dois *flip-flops* tipo D sensíveis à borda de subida com entradas de *preset* e *clear* e a tabela verdade correspondente.

7474 – Dois flip-flops tipo D sensíveis à borda de subida com entradas de preset e clear inversoras

A figura 4.19 apresenta esse dispositivo, e tabela verdade correspondente.

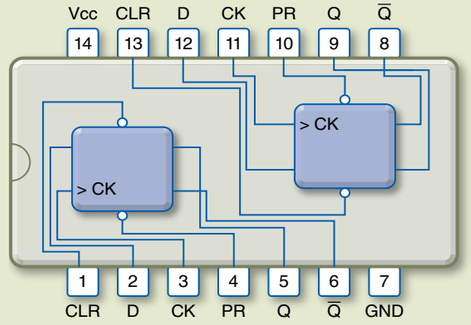


Tabela verdade

Entradas				Saídas	
PRESET	CLEAR	CLOCK	D	Q	Q̄
0	1	X	X	1	0
1	0	X	X	0	1
0	0	X	X	1	1
1	1	↑	1	1	0
1	1	↑	0	0	1
1	1	0	X	Q_0	$\bar{Q}_0$
1	1	1	X	Q_0	$\bar{Q}_0$

Figura 4.19
CI 7474 com dois *flip-flops* tipo D sensíveis à borda de subida com entradas de *preset* e *clear* inversoras.

Implementação de um *flip-flop* T a partir do J-K

O *flip-flop* T ou *toggle* muda sua saída a cada transição do sinal de *clock* (ver exemplo na figura 4.20, na transição positiva). Consequentemente, a frequência do sinal de saída é metade da frequência do sinal de entrada aplicado na entrada T.

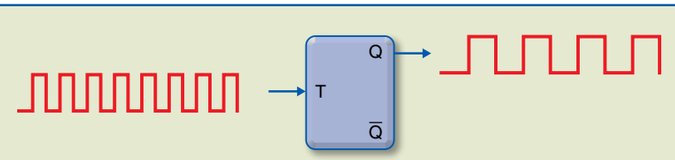


Figura 4.20
Representação da mudança da saída para um *flip-flop* T.

O *flip-flop* T é obtido a partir do *flip-flop* J-K aplicando nível lógico alto tanto na entrada J como na K (figura 4.21).

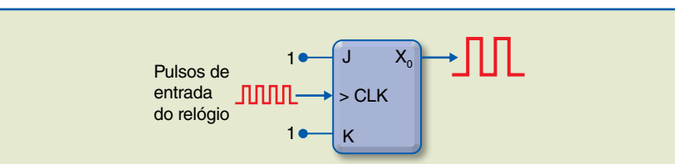


Figura 4.21
Representação mostrando como um *flip-flop* T é obtido a partir do *flip-flop* J-K.

A figura 4.22 mostra as formas de onda correspondentes nas saídas Q e Q', a partir do *clock*.

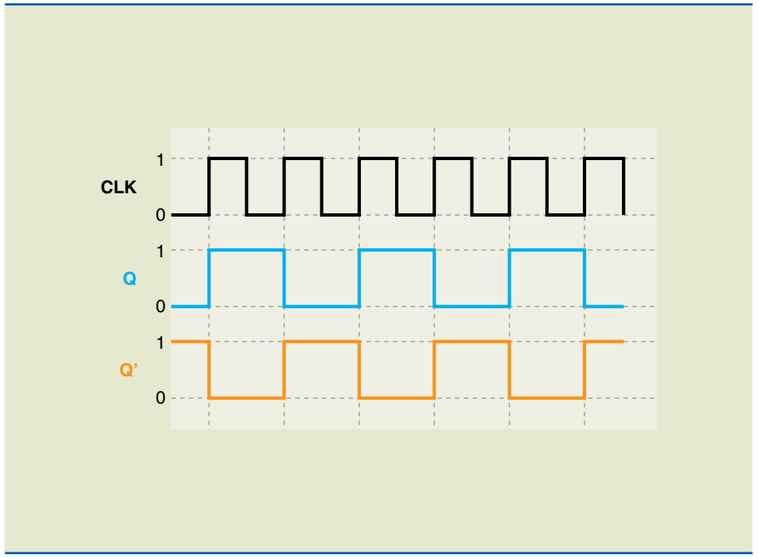


Figura 4.22
Formas de onda correspondentes nas saídas Q e Q', a partir do *clock*.

4.2 Contadores

Contadores são circuitos digitais que geram determinada sequência de estados, sob o comando de um sinal de *clock*. São utilizados na contagem de pulsos provenientes de chaves e de sensores, na construção de temporizadores e relógios digitais, para gerar sequências de pulsos e medir frequência, e também fazem parte de circuitos eletrônicos como conversores analógico-digital e digital-analógico, geradores de endereços de matrizes de memória etc.

Os contadores são basicamente divididos em duas categorias – **assíncronos** e **síncronos** – e podem ser classificados de acordo com a sequência (crescente ou decrescente) e com o módulo (binário, decimal, módulo n).

4.2.1 Contadores assíncronos

Os contadores assíncronos não possuem entradas comuns de sinal de *clock*. O sinal inicial é aplicado no primeiro estágio; os demais recebem o sinal do estágio anterior.

Contador binário

Um contador binário pode ser construído a partir de *flip-flops* J-K conectando a saída de uma célula à entrada de *clock* da célula seguinte. As entradas J e K de todos os *flip-flops* são mantidas em nível lógico "1" para produzir o efeito *toggle* a cada pulso de *clock*. Para cada dois pulsos de *clock* na entrada de determinada célula é produzido um pulso na respectiva saída. Isso resulta uma sequência binária quando o número de *flip-flops* é igual a quatro. Esse dispositivo geralmente é chamado de contador de pulsos (*ripple counter*).

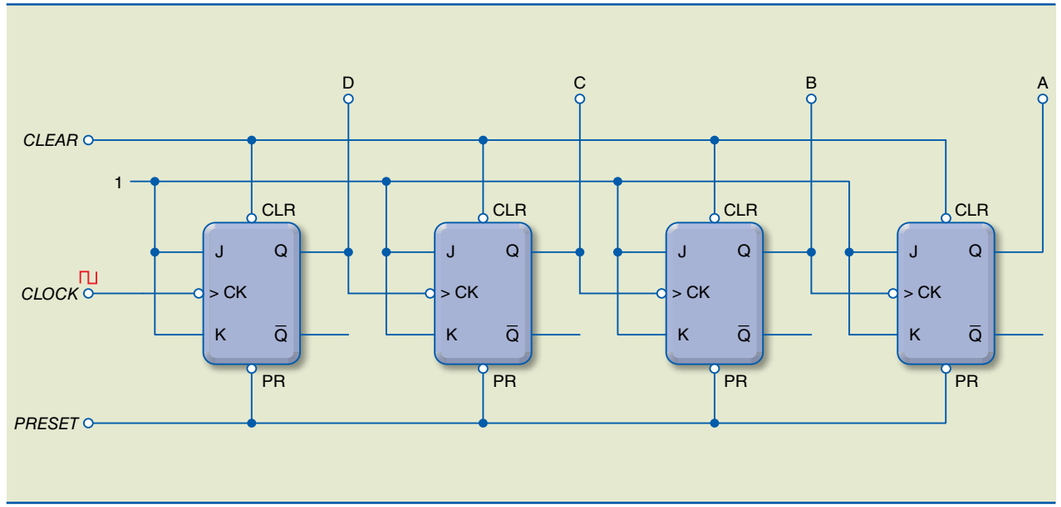


Figura 4.23
Representação de um Contador de pulsos.

As formas de onda nas saídas em função do sinal de *clock* são apresentadas na figura 4.24.

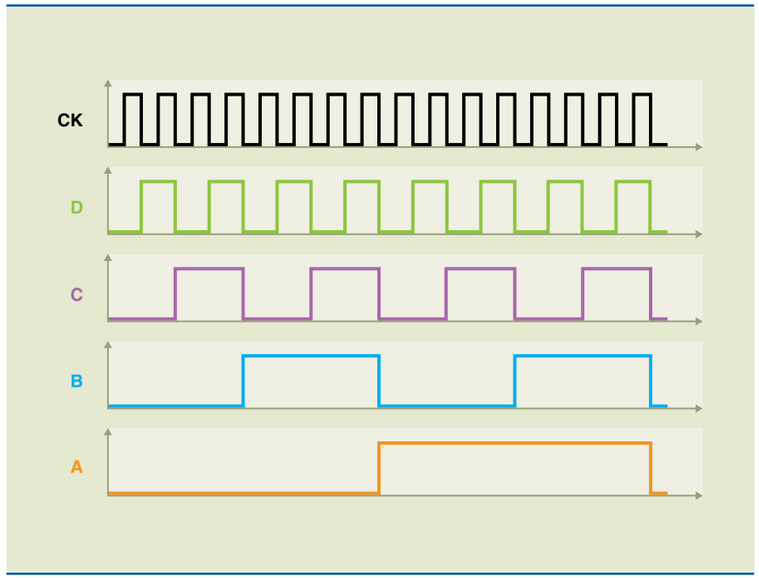


Figura 4.24
Formas de onda nas saídas em função do sinal de *clock*.

Contador de década (BCD *counter*)

Uma das representações de dados numéricas mais utilizadas é o decimal codificado em binário (BCD – *binay coded decimal*). Nessa codificação, cada número decimal inteiro é representado por um código binário de quatro dígitos, conforme a tabela 4.1.

Tabela 4.1

Decimal	BCD 8421
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Observe, no exemplo da figura 4.25, a equivalência entre um número decimal e sua representação em BCD.

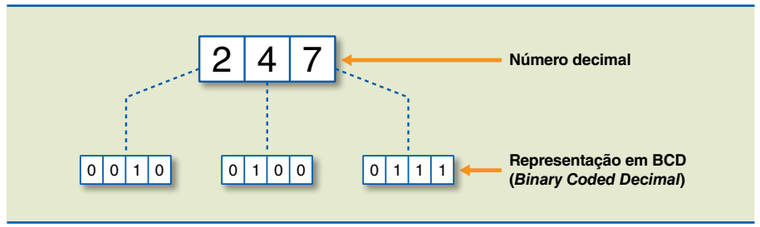


Figura 4.25
Equivalência entre o número 247 e sua representação em BCD.

Um contador BCD ou contador de décadas (figura 4.26) pode ser construído a partir de um contador binário capaz de encerrar a transmissão de pulsos quando a contagem atinge o estado correspondente ao número decimal 9 (1001 em binário).

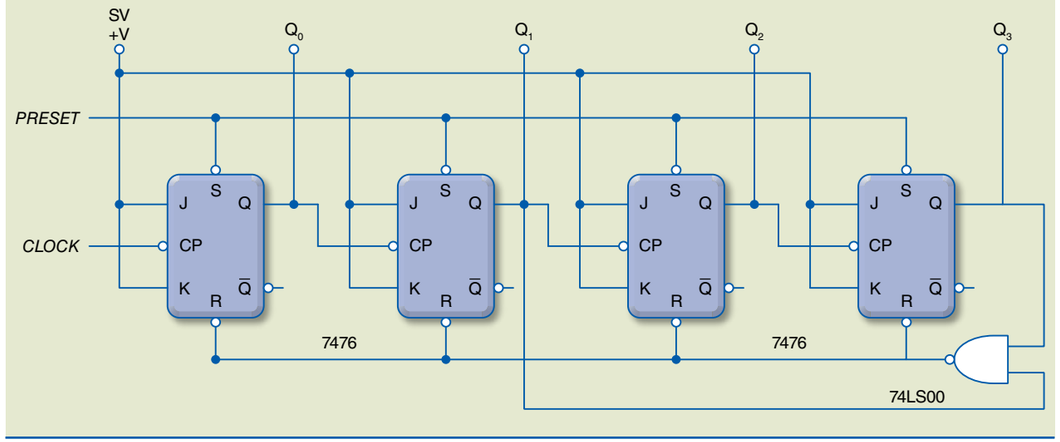


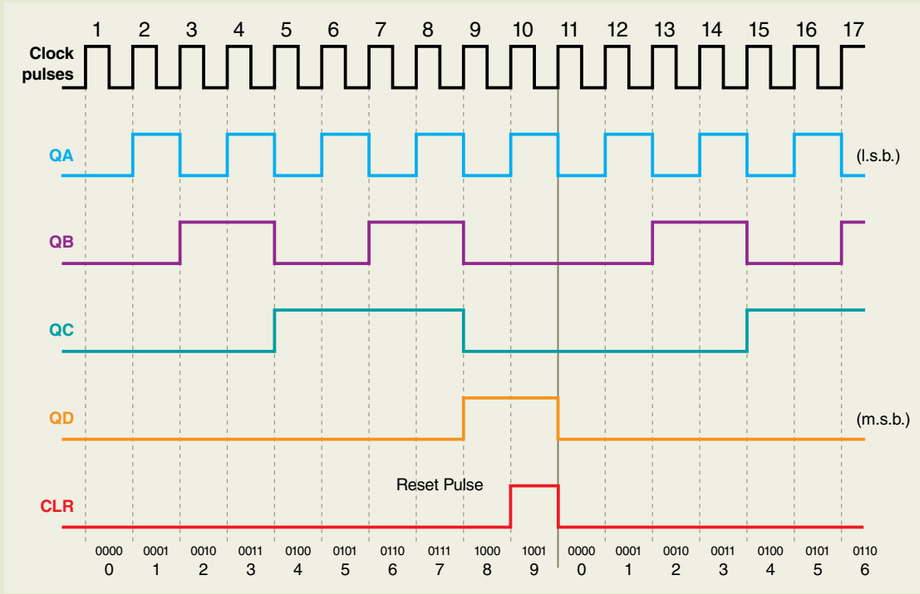
Figura 4.26
Circuito lógico do contador BCD.

Análise do funcionamento

Observando a figura 4.26, podemos analisar o funcionamento do contador BCD. Vamos considerar a situação: uma vez que o pulso seguinte levaria ao estado correspondente ao binário 1010, bastaria conectarmos os dois bits altos (Q_3 e Q_1) às entradas de uma porta NAND cuja saída é ligada à entrada assíncrona de *clear* dos *flip-flops*. Isso provocaria um *reset* automático nos *flip-flops* após o número 9, reiniciando, assim, a contagem.

Observe na figura 4.27 as formas de onda para o *flip-flop* da figura 4.26.

Figura 4.27
Formas de ondas na saída em função do ***clock*** inicial.



4.2.2 Contadores síncronos

Nesse tipo de contador, o sinal de *clock* é comum a todos os *flip-flops* que o compõem, ou seja, todos os estágios são sincronizados simultaneamente.

É possível projetar um **contador** síncrono utilizando *flip-flops* tipo D. Para isso, devemos seguir as etapas:

1) Especificar a sequência do contador.

Por exemplo, a sequência é:

5, 7, 3, 2, 6 → repetidamente, ou seja, em binário: 101, 111, 011, 010, 110.

2) Gerar a tabela de estados (tabela 4.2)

Tabela 4.2

Estado Atual				Estado Futuro			
	A	B	C		A	B	C
0	0	0	0	X	X	X	X
1	0	0	1	X	X	X	X
2	0	1	0	6	1	1	0
3	0	1	1	2	0	1	0
4	1	0	0	X	X	X	X
5	1	0	1	7	1	1	1
6	1	1	0	5	1	0	1
7	1	1	1	3	0	1	1

3) Determinar quais os sinais de entrada necessários para forçar os *flip-flops* a assumir os valores desejados na sequência (tabela 4.3).

Tabela 4.3

Estado Atual				Estado Futuro				Entradas dos *Flip-Flops*		
	A	B	C		A	B	C	D_A	D_B	D_C
0	0	0	0	X	X	X	X	X	X	X
1	0	0	1	X	X	X	X	X	X	X
2	0	1	0	6	1	1	0	1	1	0
3	0	1	1	2	0	1	0	0	1	0
4	1	0	0	X	X	X	X	X	X	X
5	1	0	1	7	1	1	1	1	1	1
6	1	1	0	5	1	0	1	1	0	1
7	1	1	1	3	0	1	1	0	1	1

De acordo com a tabela 4.3, há três funções a serem implementadas: D_A, D_B e D_C, que podem ser apresentadas conforme a figura 4.28.

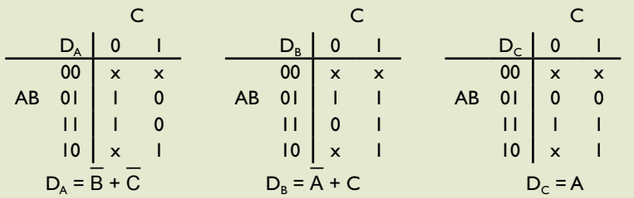


Figura 4.28
Diagramas de Karnaugh para as funções D_A, D_B e D_C.

Dessa maneira, podemos montar um circuito lógico que atenda a essas funções (figura 4.29).

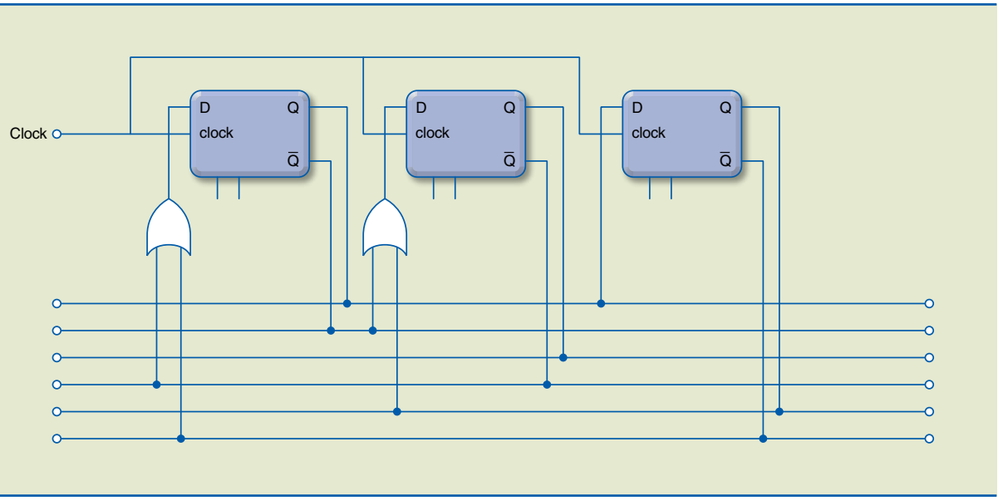


Figura 4.29
Circuito lógico para as funções D_A, D_B e D_C.

Implementação do circuito utilizando flip-flops tipo J-K

A figura 4.30 apresenta um *flip-flop* J-K e a tabela verdade correspondente. Observe que Qa é o valor anterior da saída Q antes da aplicação dos valores das entradas J e K. As mudanças somente ocorrem na variação (descida) de "1" para "0" dos pulsos aplicados na entrada de *clock*.

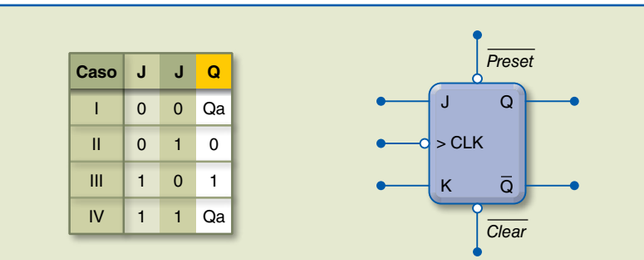


Figura 4.30
Flip-flop J-K.

Com base nas informações da tabela verdade, podemos elaborar uma **tabela de transição de estados do J-K** (tabela 4.4).

Tabela 4.4

Casos	Qa	Q	J	K
I e II	0	0	0	X
III e IV	0	I	I	X
II e IV	I	0	X	I
I e III	I	I	X	0

A tabela de transições (tabela 4.5) apresenta as entradas necessárias para forçar os valores nas saídas dos *flip-flops* a ir para a sequência desejada.

Tabela 4.5

Estado Atual				Estado Futuro				Entradas dos Flip-Flops					
	A	B	C		A	B	C	J_A	K_A	J_B	K_B	J_C	K_c
0	0	0	0	X	X	X	X	X	X	X	X	X	X
I	0	0	I	X	X	X	X	X	X	X	X	X	X
2	0	I	0	6	I	I	0	I	X	X	0	0	X
3	0	I	I	2	0	I	0	0	X	X	0	X	I
4	I	0	0	X	X	X	X	X	X	X	X	X	X
5	I	0	I	7	I	I	I	X	0	I	X	X	0
6	I	I	0	5	I	0	I	X	0	X	I	I	X
7	I	I	I	3	0	I	I	X	I	X	0	X	0

De acordo com a tabela 4.5, podemos elaborar o mapa de Karnaugh identificando as funções a serem implementadas, conforme mostra a figura 4.31.

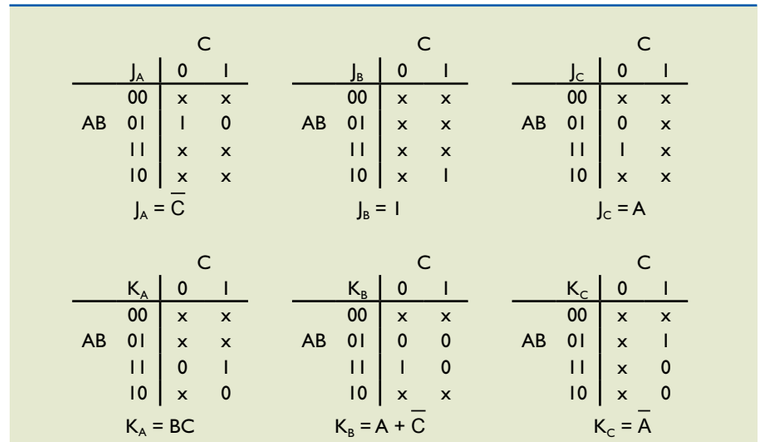


Figura 4.31
Diagramas de Karnaugh para as funções J_A, J_B, J_C, K_A, K_B e K_C.

Dessa maneira, podemos montar um circuito lógico que atenda a essas funções (figura 4.32).

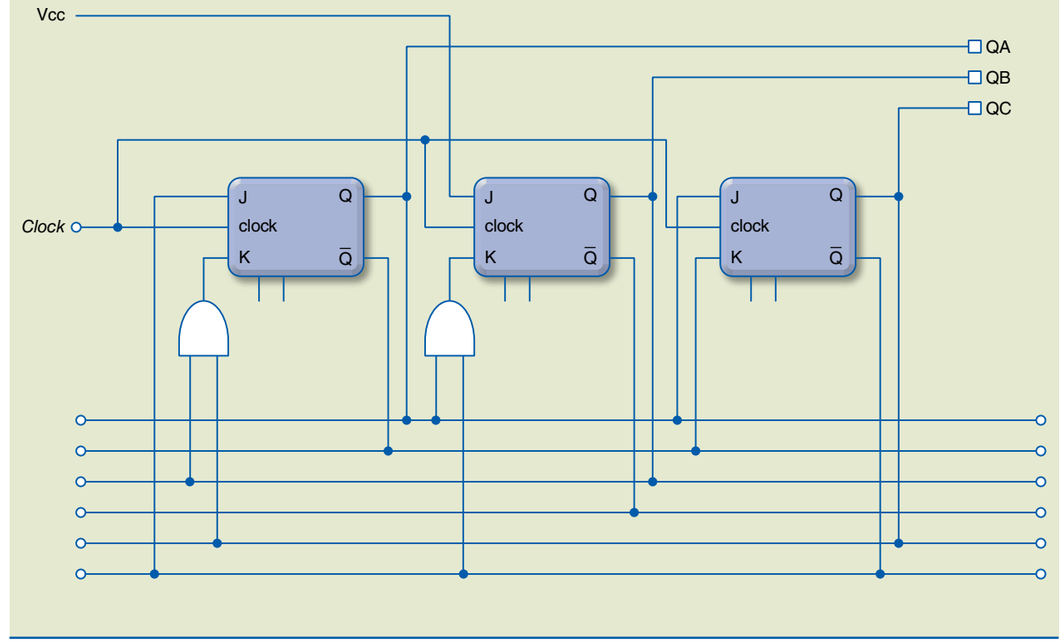


Figura 4.32
Circuito lógico para as funções J_A, J_B, J_C, K_A, K_B e K_C.

Projeto: contador decimal (BCD) síncrono

As informações necessárias para montar um contador decimal (BCD) síncrono são as seguintes:

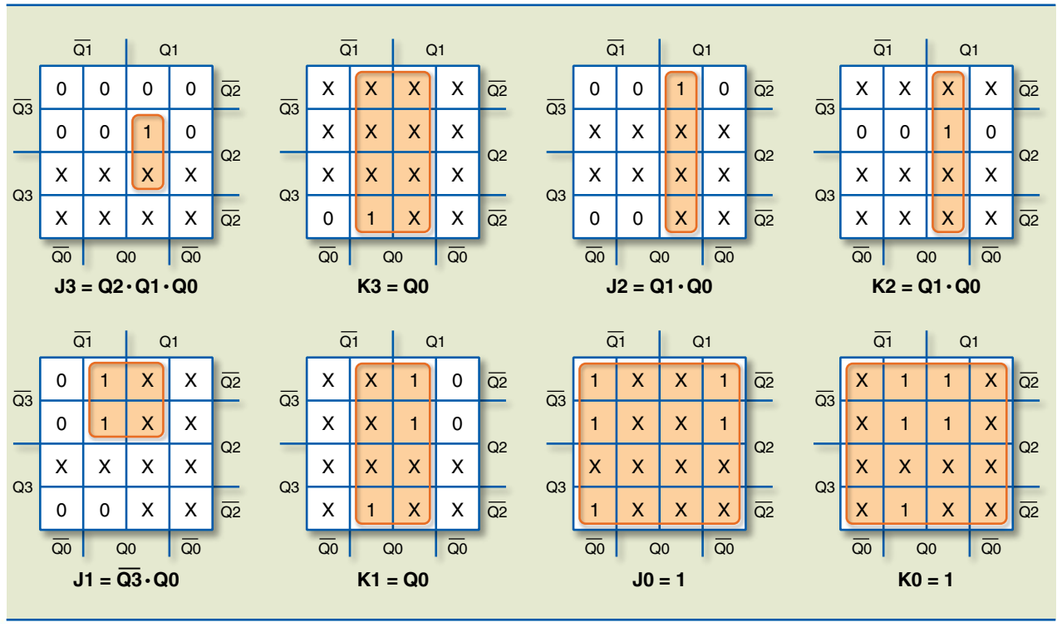
Tabela 4.6

1) Tabela de estados.

	Saídas Atuais				Saídas Futuras				Entradas dos *flip-flops*							
Estados	Q3	Q2	Q1	Q0	Q3	Q2	Q1	Q0	J3	K3	J2	K2	J1	K1	J0	K0
1	0	0	0	0	0	0	0	1	0	x	0	x	0	x	1	x
2	0	0	0	1	0	0	1	0	0	x	0	x	1	x	x	1
3	0	0	1	0	0	0	1	1	0	x	0	x	x	0	1	x
4	0	0	1	1	0	1	0	0	0	x	1	x	x	1	x	1
5	0	1	0	0	0	1	0	1	0	x	x	0	0	x	1	x
6	0	1	0	1	0	1	1	0	0	x	x	0	1	x	x	1
7	0	1	1	0	0	1	1	1	0	x	x	0	x	0	1	x
8	0	1	1	1	1	0	0	0	1	x	x	1	x	1	x	1
9	1	0	0	0	1	0	0	1	x	0	0	x	0	x	1	x
10	1	0	0	1	0	0	0	0	x	1	0	x	0	x	x	1

Figura 4.33
Diagramas de Karnaugh e expressões lógicas.

2) Diagramas de Karnaugh e expressões lógicas (figura 4.33).



3) Circuito lógico (figura 4.34).

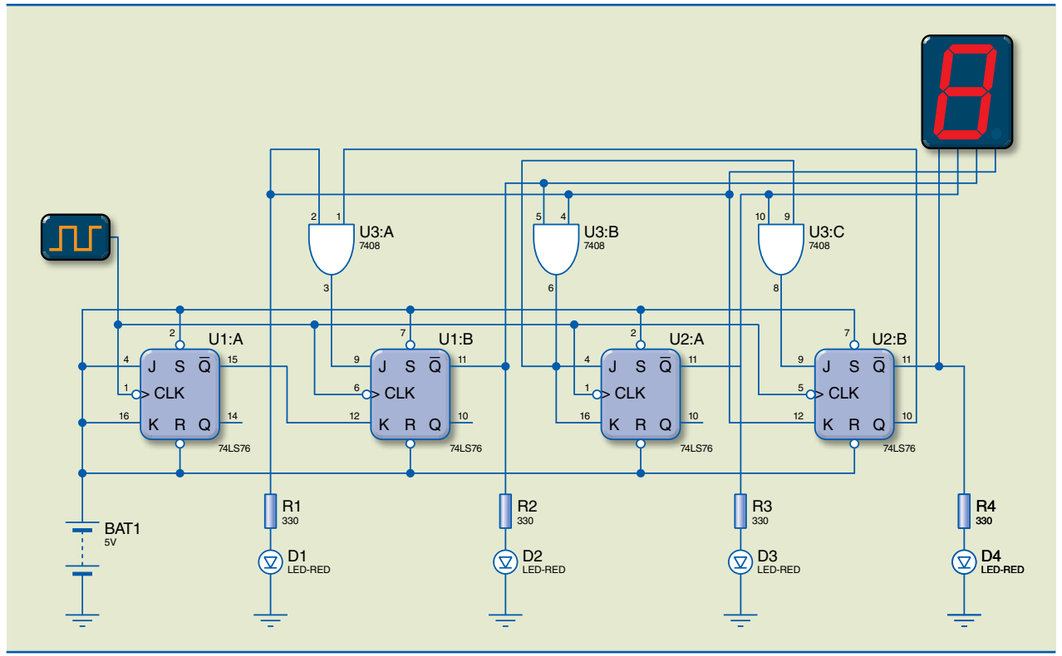


Figura 4.34
Circuito lógico contador decimal (BCD) síncrono.

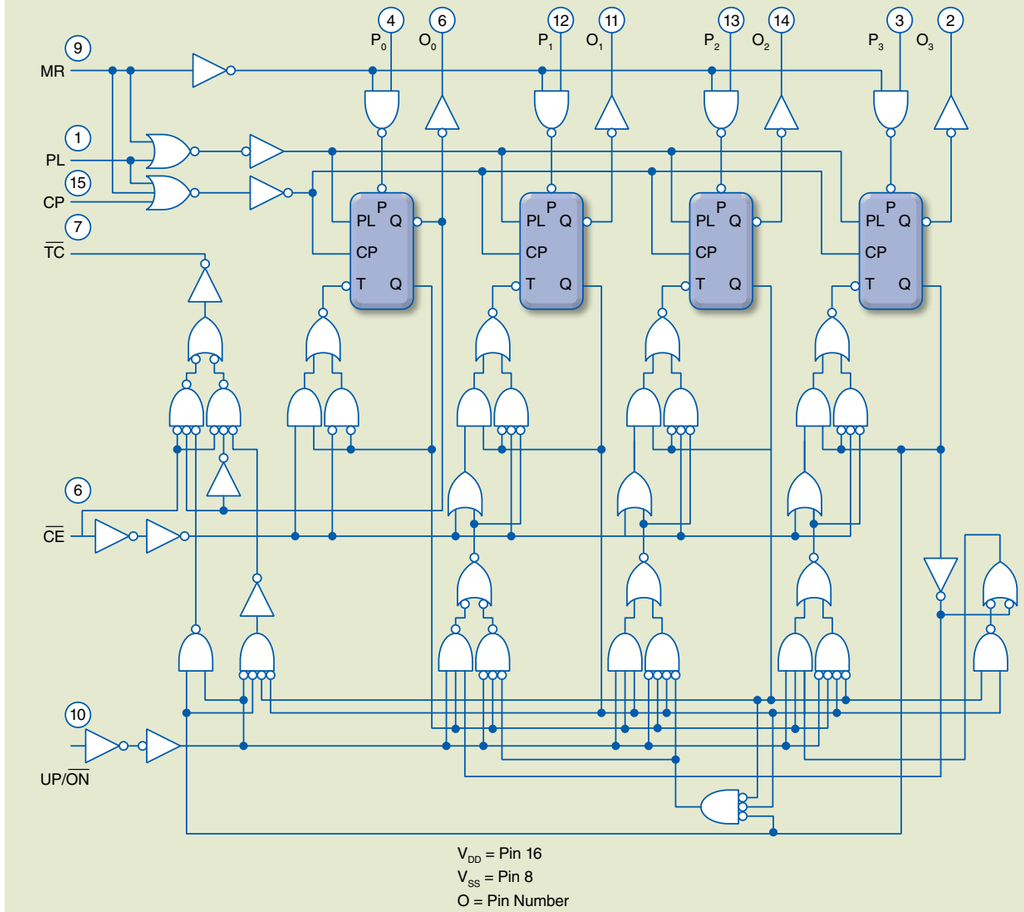
Exemplo de circuito integrado CMOS 4510 BCD counter

Pin Description		
Pin nº	Symbol	Name and Function
1	PL	parallel load input (active HIGH)
4, 12, 13, 3	D0 to D3	parallel inputs
5	CE	count enable input (active LOW)
6, 11, 14, 2	Q0 to Q3	parallel outputs
7	TC	terminal count output (active LOW)
8	GND	ground (0V)
9	MR	asynchronous master reset input (active HIGH)
10	UP/DN	up/down control input
15	CP	clock input (LOW-to-HIGH, edge-triggered)
16	VCC	positive supply voltage

Tabela 4.7
Descrição dos pinos do circuito integrado CMOS 4510 BCD *counter.*

Figura 4.35
Identificação dos pinos do circuito integrado CMOS 4510 BCD *counter.*

Figura 4.36
Detalhe das ligações internas do circuito integrado CMOS 4510 BCD *counter.*



Exemplo de circuito de teste para o contador de décadas 4510.

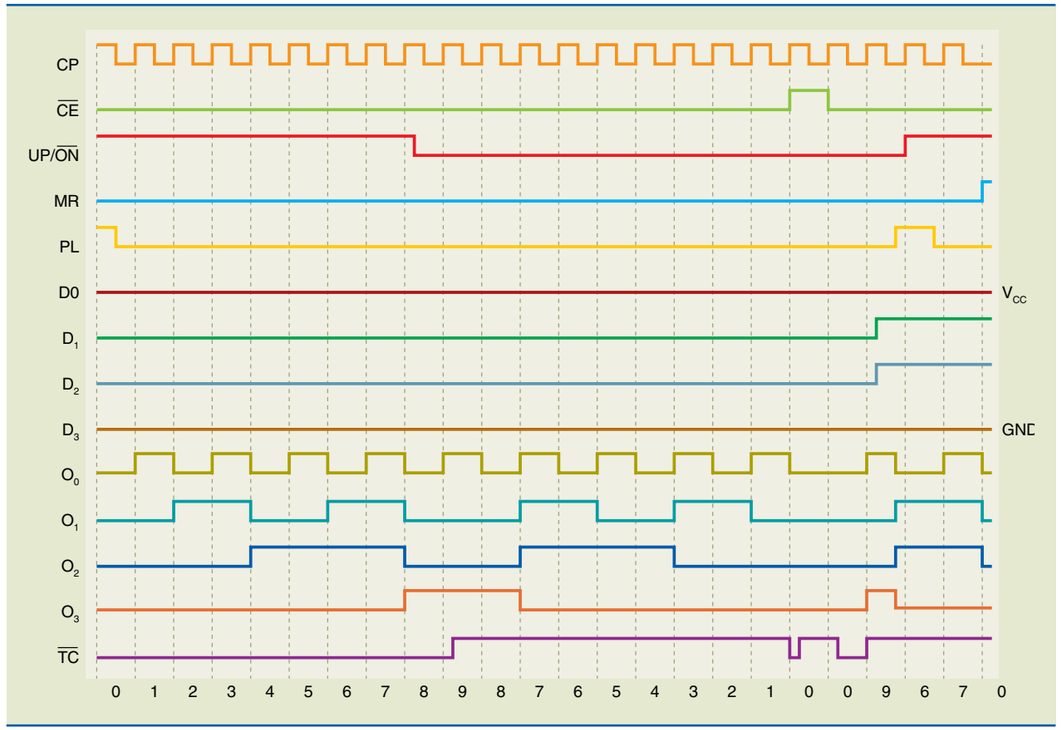


Figura 4.37
Diagrama de tempos do circuito integrado CMOS 4510 BCD *counter.*

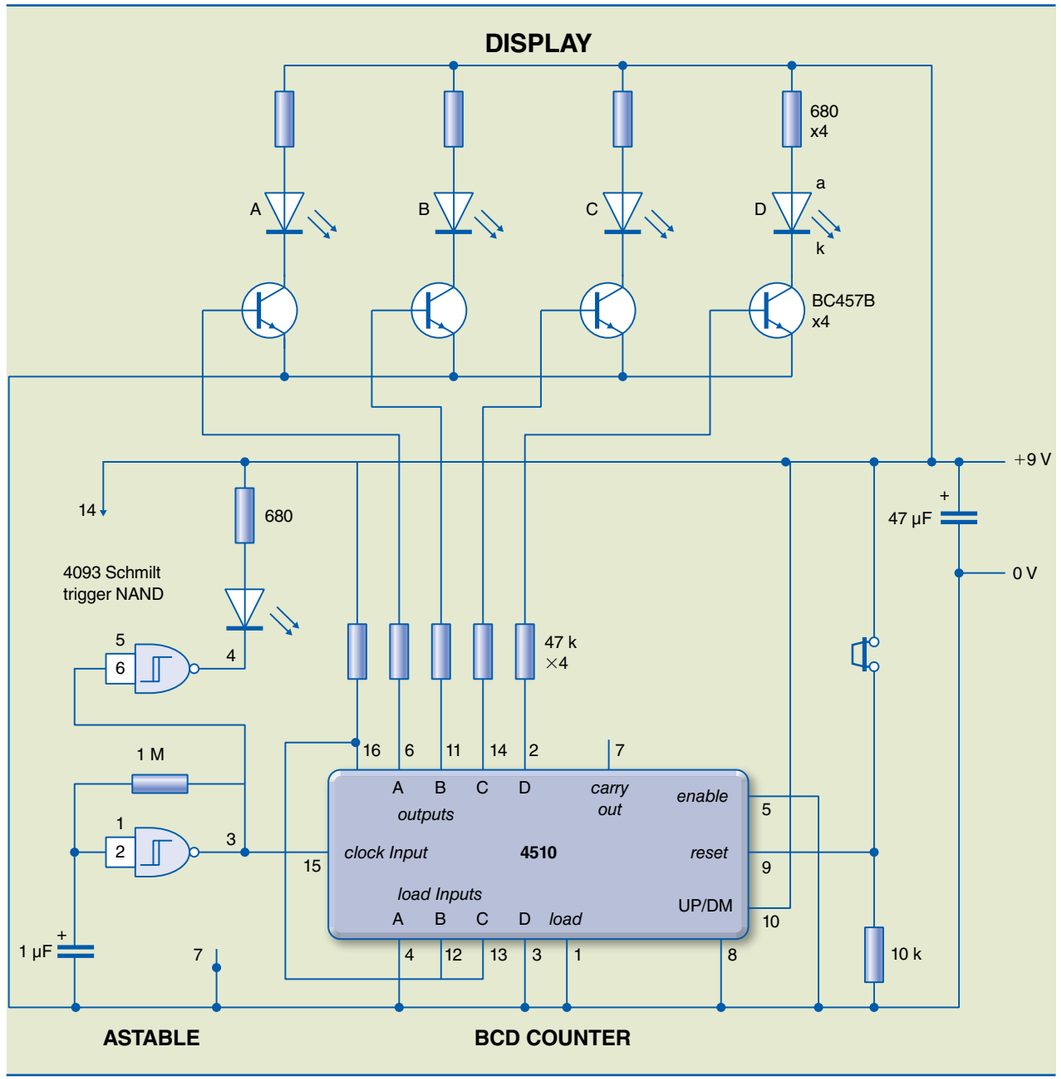


Figura 4.38
Detalhes internos de circuito de teste para o contador de décadas 4510.

Um contador crescente/decrescente tem a lógica interna apresentada na figura 4.39.

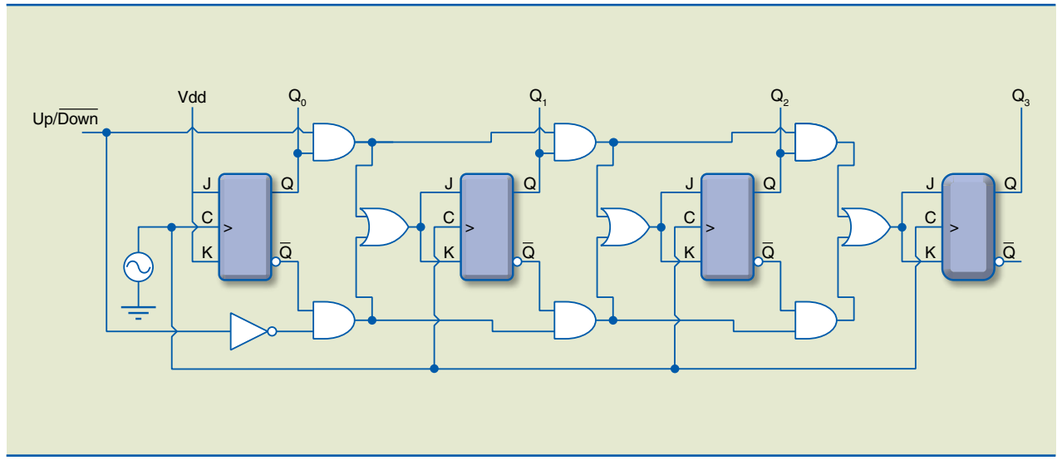


Figura 4.39
Detalhes internos de um contador crescente/decrescente.

Contador em anel

Contador em anel é um conjunto de *flip-flops* conectados em cascata à saída do último estágio conectado à entrada do primeiro, fechando um anel. Um uso comum desse circuito consiste em um único bit = 1, que circula através das saídas. Por exemplo, se forem utilizados quatro *flip-flops*, haverá quatro estados de saída (0001 / 0010 / 0100 / 1000), e cada um deles se repetirá a cada quatro ciclos de *clock*. Nesse caso, ele pode ser usado como um contador cíclico de n estados. O circuito da figura 4.40 mostra um contador em anel módulo 4.

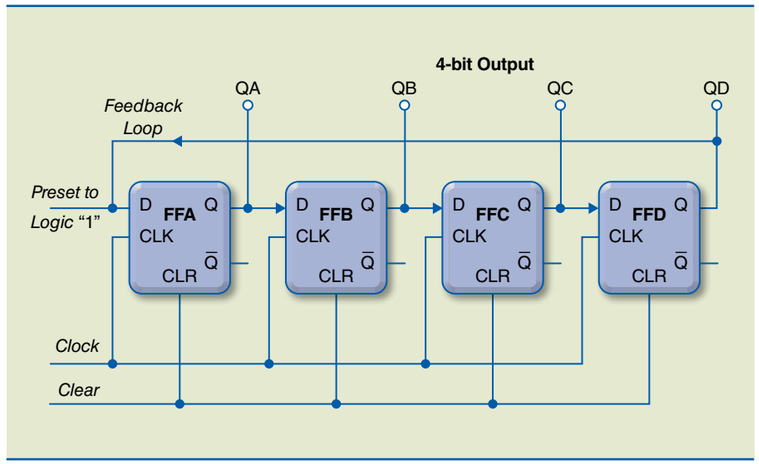


Figura 4.40
Contador em anel módulo 4.

A tabela 4.8 apresenta a sequência de estados do circuito de contador em anel módulo 4.

Tabela 4.8

Pulso de Clock	Q3	Q2	Q1	Q0	
0	0	0	0	1	↑
1	0	0	1	0	
2	0	1	0	0	
3	1	0	0	0	

Contador Johnson

Contador Johnson (ou contador de anel torcido) é um contador em anel modificado, no qual a saída do último estágio invertida é realimentada para a entrada do primeiro estágio. O circuito da figura 4.41 mostra um contador Johnson módulo 4.

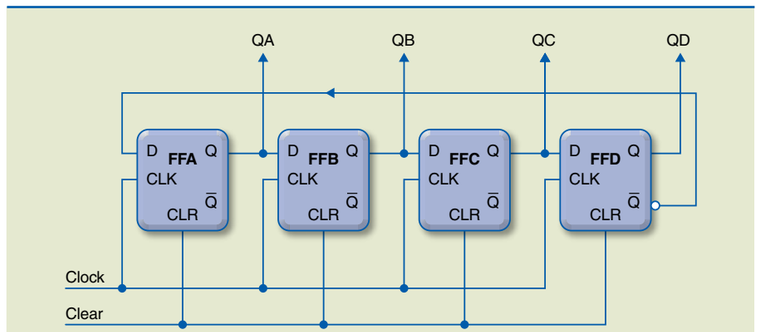


Figura 4.41
Contador Johnson módulo 4.

A tabela 4.9 apresenta a sequência de estados gerada pelo circuito do contador Johnson módulo 4.

Tabela 4.9

Pulso de Clock	Q3	Q2	Q1	Q0	
0	0	0	0	0	↑
1	0	0	0	1	
2	0	0	1	1	
3	1	1	1	1	
4	1	1	1	1	
5	1	1	1	0	
6	1	1	0	0	
7	1	0	0	0	

4.3 Registradores de deslocamento

Registrador é um circuito formado por interligações de *flip-flops* com a finalidade de armazenar informação binária (número binário) pelo tempo que for necessário.

Os registradores são utilizados em operações aritméticas de complementação, multiplicação e divisão, em conversão de uma informação série em paralela e também em vários outros tipos de circuitos digitais.

4.3.1 Informação série e informação paralela

Uma informação é chamada **informação série** quando os bits são apresentados sequencialmente, um após o outro, necessitando somente de uma via para o transporte dos bits. Esse modo de transferir informação é conhecido como **deslocamento em série**.

Uma informação é chamada **informação paralela** quando os bits são apresentados simultaneamente; assim, a transferência da informação acontece em um único instante. É necessária uma quantidade de vias para transmissão igual ao número de bits da informação. Esse modo é conhecido como **deslocamento paralelo**.

A entrada e a saída de um registrador podem ser configuradas nesses dois modos, resultando em quatro possibilidades: entrada série – saída série, entrada série – saída paralela, entrada paralela – saída série, entrada paralela – saída paralela (figura 4.42).

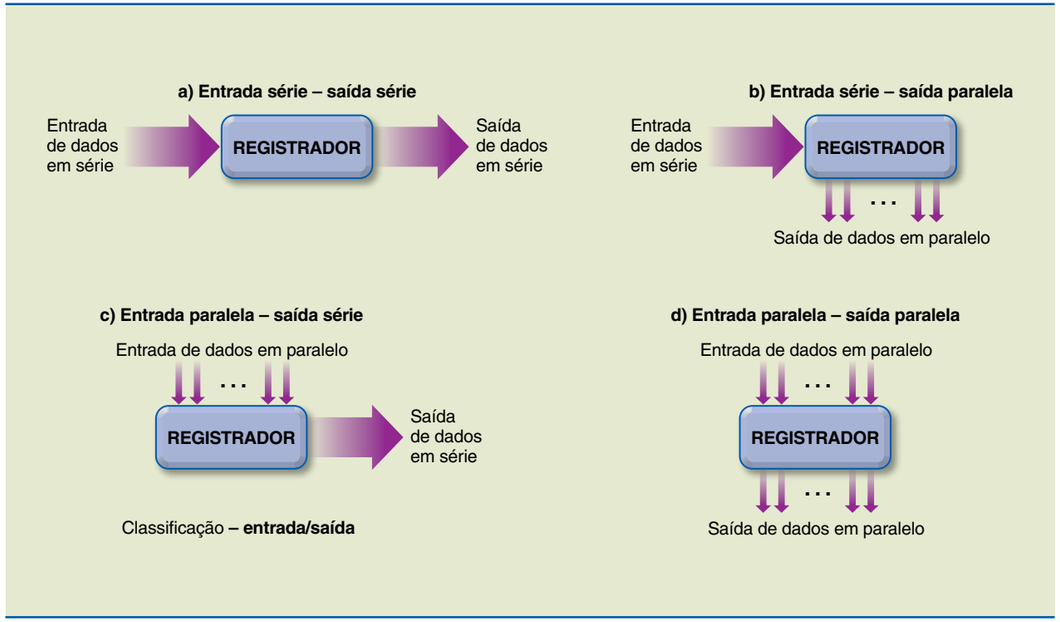
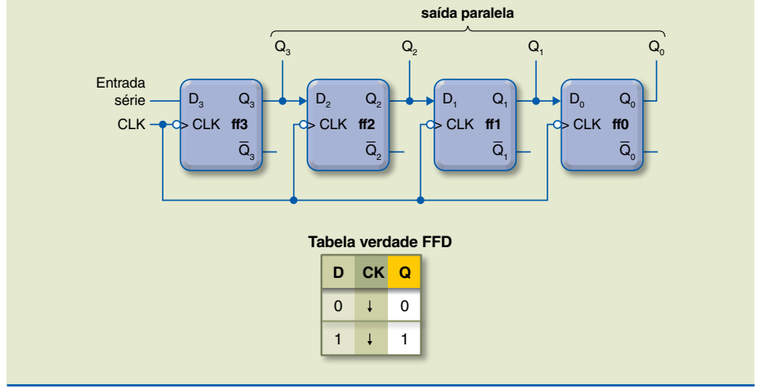


Figura 4.42
Configurações da entrada e da saída de um registrador: (a) entrada série – saída série, (b) entrada série – saída paralela, (c) entrada paralela – saída série e (d) entrada paralela – saída paralela.

4.3.2 Registrador de deslocamento para a direita

O circuito da figura 4.43 mostra como o registrador de deslocamento pode ser montado usando *flip-flops* tipo D.

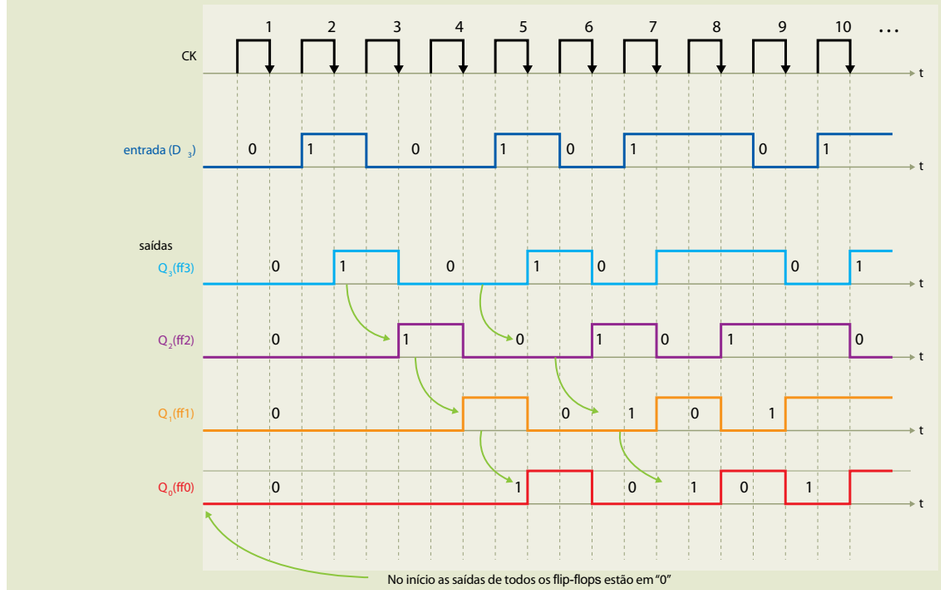
Figura 4.43
Registrador de deslocamento para a direita usando *flip-flop* tipo D e tabela verdade.



No símbolo dos *flip-flops (ffs)* da figura 4.43, a "bolinha" na entrada do *clock* indica sensibilidade à borda negativa. Os *flip-flops* desse circuito são do tipo D, sensível à borda negativa, como podemos observar pelos símbolos dos *ffs*. Na tabela verdade, a seta apontando para baixo indica sensibilidade à borda negativa.

Figura 4.44
Formas de onda do registrador de quatro bits.

As formas de onda de entrada e saída do registrador de quatro bits são apresentadas na figura 4.44.



Em um *flip-flop* mestre-escravo, a atualização da saída devido à transição do *clock* só ocorre imediatamente após o fim da transição do *clock*.

No circuito figura 4.43, observamos que o *clock* ocorre simultaneamente em todos os *flip-flops*. No momento da transição negativa do *clock*, D_2 tem como entrada o valor de Q_3 anterior à transição do *clock*, pois Q_3 somente terá seu valor atualizado após o fim da transição.

Vamos verificar como progride o primeiro bit "1" de entrada nos *flip-flops* do registrador em análise. Acompanhe pelas formas de onda na figura 4.44.

- Após o segundo pulso, o bit "1" é colocado na saída Q_3 do ff3.
- Após o terceiro pulso, o bit "1" é colocado na saída Q_2 do ff2.
- Após o quarto pulso, o bit "1" é colocado na saída Q_1 do ff1.
- Após o quinto pulso, o bit "1" é colocado na saída Q_0 do ff0.
- Após o sexto pulso, o bit "1" é perdido, ou seja, não está na saída de nenhum *flip-flop* do circuito.

Como podemos observar, o primeiro bit "1" deslocou-se para a direita a cada pulso de *clock*. O deslocamento que ocorreu com o bit "1" ocorre com os demais bits. Esse deslocamento que os bits de entrada apresentam a cada pulso de *clock* deu origem ao nome registrador de deslocamento.

4.4 Registrador de deslocamento para a esquerda

Para obter o registrador de deslocamento para a esquerda, basta mudar a ordem dos *flip-flops* e a entrada do registrador passará a ser no primeiro *flip-flop* da direita (figura 4.45).

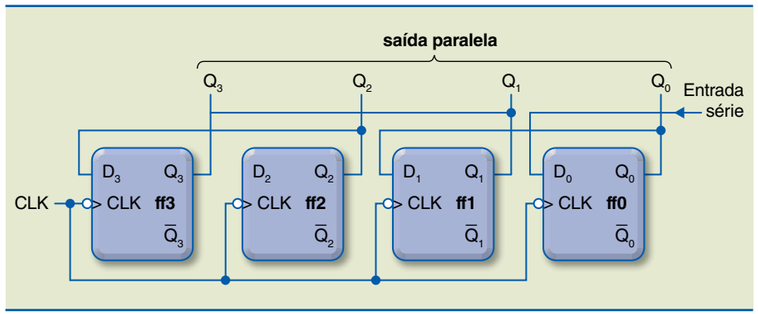


Figura 4.45
Registrador de deslocamento para a esquerda.

CI 7491 – Registrador de deslocamento de oito bits – entrada série e saída série (figura 4.46)

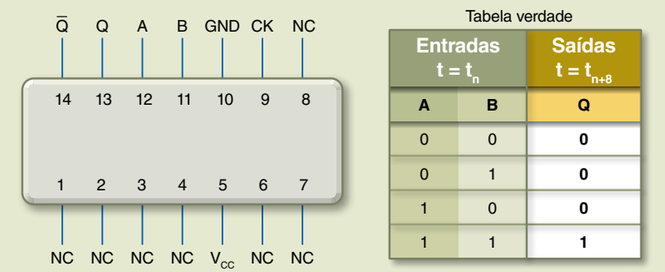


Figura 4.46
Registrador de deslocamento de oito bits: (a) identificação dos pinos do CI 7491, (b) tabela verdade e (c) detalhe do circuito interno do CI.

Se uma das entradas for "0", será transferido "0" para a saída do registrador após oito pulsos de *clock*, independentemente da outra entrada. Se uma das entradas for "1", o valor da outra entrada ("0" ou "1") será transferido para a saída do registrador após oito pulsos de *clock*. Podemos usar uma entrada como controle e a outra como entrada de dados.

CI 74164 – Registrador de deslocamento de oito bits com entrada de *reset* (figura 4.47)

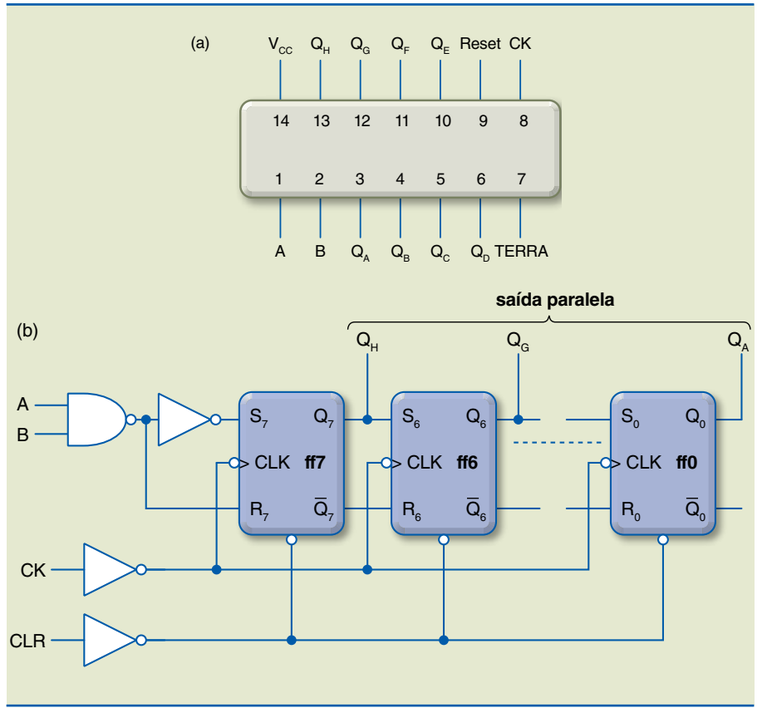


Figura 4.47
Registrador de deslocamento de oito bits: (a) identificação dos pinos do CI 74164 e (b) detalhe do circuito interno do CI 74164.

O CI 74164 é um registrador de deslocamento só para a direita, podendo ser utilizado como entrada série e saída série ou paralela. É sensível à transição negativa do *clock* e normalmente é usado com uma das entradas séries (A ou B) alta e os dados são enviados para outra entrada.

4.4.1 Circuito registrador de deslocamento – entrada série ou paralela

Os dados em paralelo transferidos para o registrador não devem ser colocados diretamente nas saídas dos *flip-flops*, pois com a ação do *clock* eles se deslocam, ocorrendo conflito. Assim, os dados em paralelo podem ser carregados no registrador pelo terminal *preset*.

Para obtermos um registrador com entrada paralela, necessitamos de *flip-flops* com *clear* e *preset*. Como sabemos, o terminal *clear* serve para colocar todas as saídas dos *flip-flops* internos em "0", ou seja, zerar ("setar") as saídas, e o *preset*, para colocar todas as saídas em "1" , ou seja, "setar" todas as saídas. O *clear* e o *preset* não podem estar ativos ao mesmo tempo, pois haveria conflito nas saídas.

Vamos avaliar como atuam o *clear* e o *preset* para que possamos obter um registrador com entrada paralela. Para isso, admitamos que o *clear* e o *preset* sejam ativos em "0".

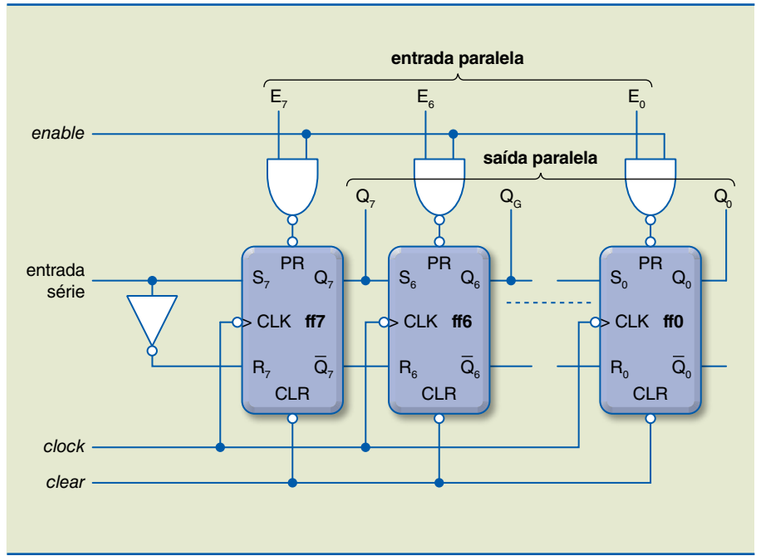
Primeiro, ativamos o *clear* (CLR = 0) zerando as saídas. Uma vez zeradas as saídas, desativamos essa função, dando condição de funcionamento normal ao registrador.

Colocamos bit a bit nos *presets* dos *flip-flops* a informação que corresponde à entrada paralela, ou seja, os terminais de *preset* estão sendo usados como entradas paralelas. Nos *presets* em que o valor colocado é "1", o *flip-flop* correspondente mudará a saída de "0" para "1" e será possível, portanto, transferir para o registrador os dados da entrada através dos *presets*. Isso feito, desativamos os *presets*, dando condição de funcionamento normal ao registrador (ver figura 4.48 – terminal *enable*).

Assim preparado, o registrador deslocará normalmente, com a ação do *clock*, os dados nele inseridos.

Vamos considerar o registrador de deslocamento para a direita da figura 4.43 e nele acrescentar *preset* e *clear* ativados em "0", com acesso possível ao *preset* de cada *flip-flop* interno. Os *clears* são interligados zerando simultaneamente, quando ativados, todos os *flip-flops* (figura 4.48).

Figura 4.48
Registrador de deslocamento – entrada série ou paralela; saída série ou paralela.



O terminal *enable* controla a função do PR, selecionando-o para ser entrada ou funcionamento normal de PR.

O terminal Q_0 ou outro da saída paralela pode ser considerado saída série, dependendo do atraso desejado na transferência do sinal ou outra condição específica do caso em questão. O referido atraso é aquele ocasionado na passagem do sinal da entrada do *flip-flop* interno para sua saída que leva um período de relógio.

Assim, um trem de pulsos na entrada se reproduzirá em Q_0 com atraso igual a (n 1)T, em que T é o período de relógio (*clock*).

4.4.2 Associação de registradores – registrador de maior capacidade

A figura 4.49 apresenta dois registradores entrada série, A_1 A_2 - - - - - D_3 D_4 com quatro bits de saída colocados em cascata para a obtenção de um registrador saída com oito bits.

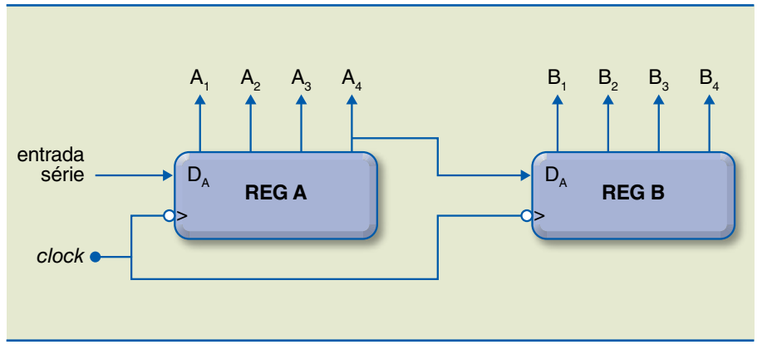


Figura 4.49
Dois registradores entrada série.

A saída série de REG A (A_4) é direcionada para a entrada série de REG B (D_4). O arranjo equivale a um registrador entrada série com saída paralela de oito bits. Podemos usar qualquer A_i ou B_i como saída série; a escolha dependerá do atraso desejado.

4.4.3 Registrador como multiplicador ou divisor por 2

Consideremos um número natural binário de oito bits, por exemplo: 1 1 0 0 0 1 1 0.

Vamos supor que esse número esteja carregado em um registrador de deslocamento e sofra deslocamento para a direita. Perde-se o bit "0" e fica indefinido o bit 7, que consideramos "0" (figura 4.50).

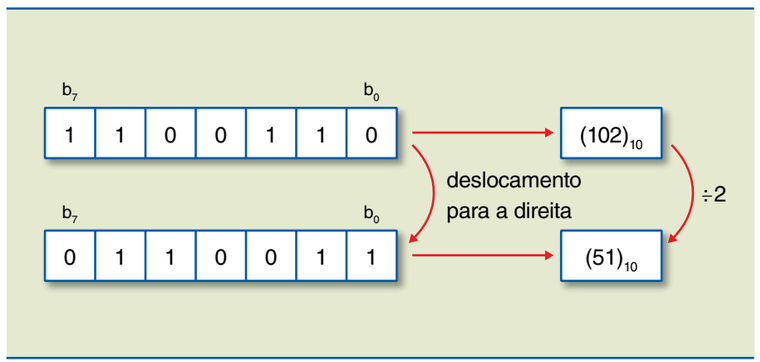


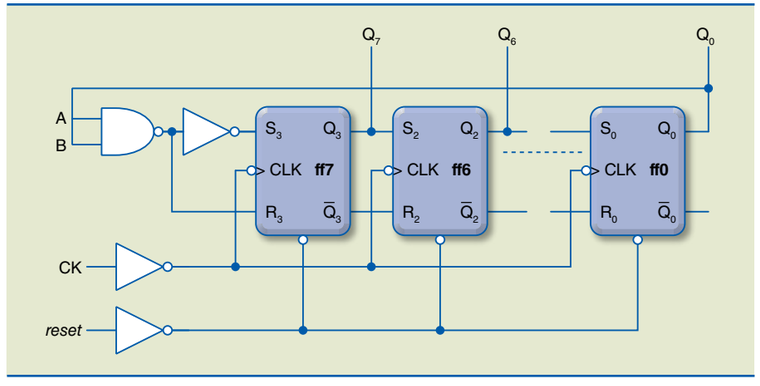
Figura 4.50
Registrador como multiplicador ou divisor por 2.

A operação de deslocamento para a direita, como vimos, pode ser associada à divisão por 2. Inversamente, um deslocamento para a esquerda pode ser associado à multiplicação por 2.

4.4.4 Registrador de deslocamento em anel

No circuito da figura 4.51, vamos conectar Q_0 à entrada série (A e B interligadas).

Figura 4.51
Registrador de deslocamento em anel de quatro bits.

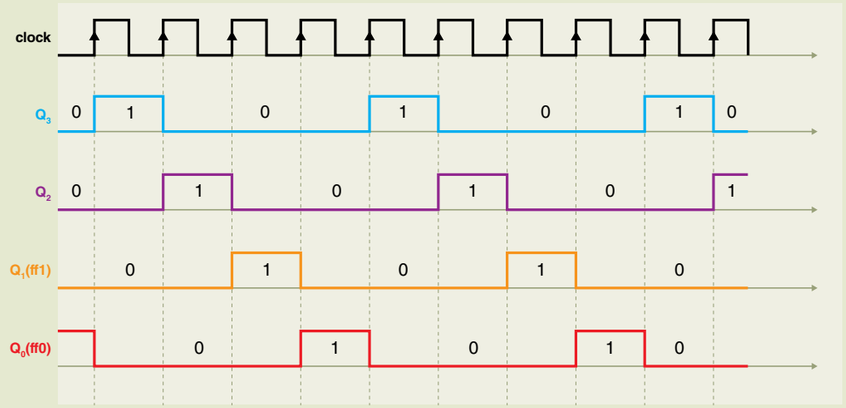


Vamos avaliar o circuito da figura 4.51, admitindo a condição inicial $Q_3 = Q_2 = Q_1 = 0$ e $Q_0 = 1$.

Na primeira transição positiva do *clock* (CK), Q_3 vai para "1", e Q_0, para "0"; as demais saídas permanecem em "0". Na segunda transição positiva, Q_2 vai para "1", e Q_3, para "0"; portanto, $Q_0 = Q_1 = 0$. Assim, "1" vai deslocando-se para a direita até $Q_0 = 1$ e as demais saídas = 0. Eventos distintos podem ser comandados (controlados) pelas saídas, obedecendo a uma sequência bem definida e em intervalos de tempo determinados pelo *clock*. Cada evento será comandado pela saída que estiver com valor "1".

Figura 4.52
Formas de onda de dois ciclos completos.

Na figura 4.52 estão registradas as formas de onda de dois ciclos completos.



Para concluir, observamos que, embora os *flip-flops* internos sejam sensíveis à transição negativa, devido ao inversor, é na transição positiva do *clock* que as mudanças ocorrem, lembrando também que, a partir do segundo *flip-flop*, o valor efetivo da entrada é o anterior à transição do *clock*, pois os *flip-flops* têm como base o *flip-flop* J-K mestre-escravo.