

Lógica Secuencial Sincrónica

- Circuitos Secuenciales

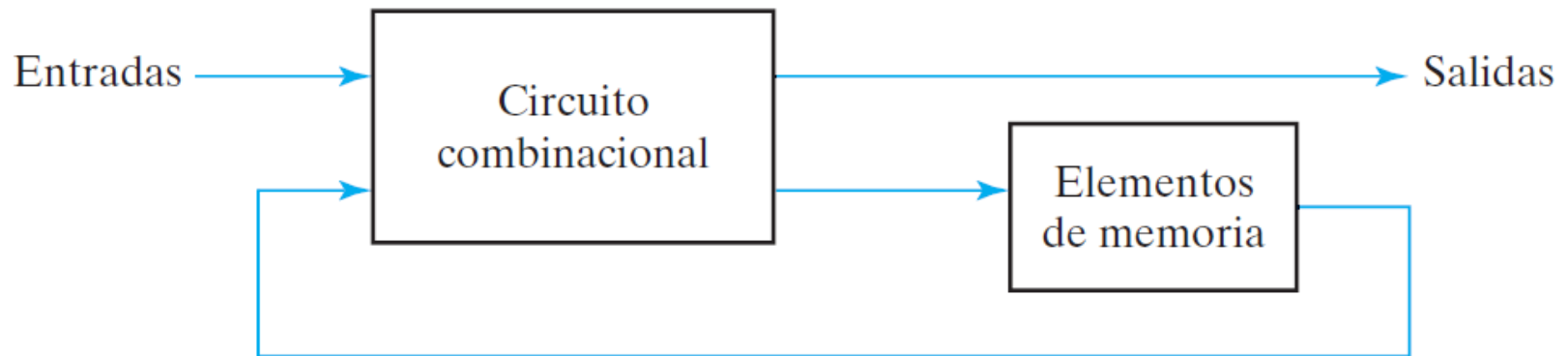
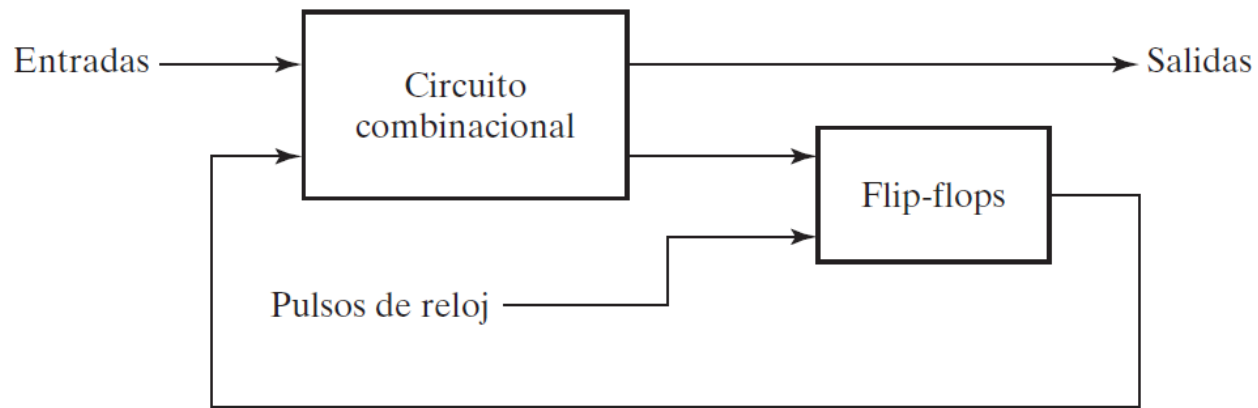


FIGURA 5-1

Diagrama de bloques de un circuito secuencial

Lógica Secuencial Sincrónica

- Circuitos Secuenciales Sincrónicos



a) Diagrama de bloques



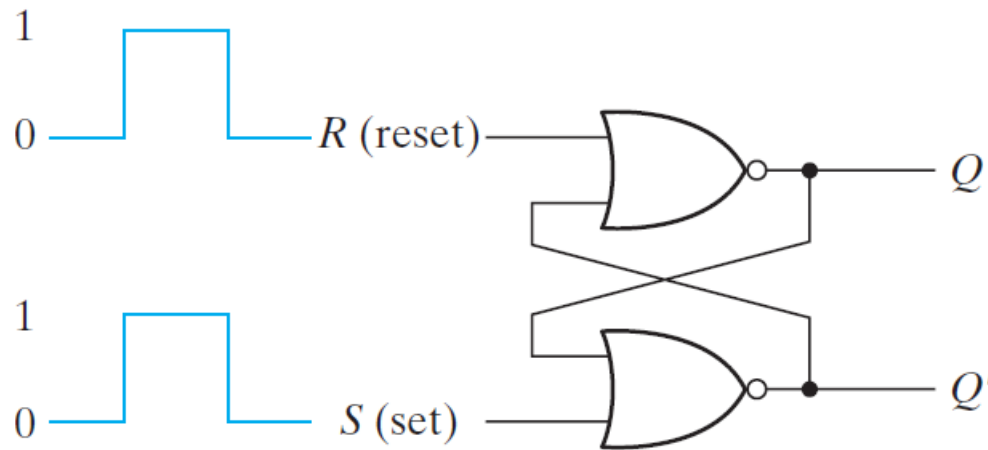
b) Diagrama de temporización de los pulsos de reloj

FIGURA 5-2

Circuito secuencial síncrono con reloj

Latches

- Latch S-R



a) Diagrama lógico

S	R	Q	Q'	
1	0	1	0	(Después de $S = 1, R = 0$)
0	0	1	0	
0	1	0	1	(Después de $S = 0, R = 1$)
0	0	0	1	
1	1	0	0	

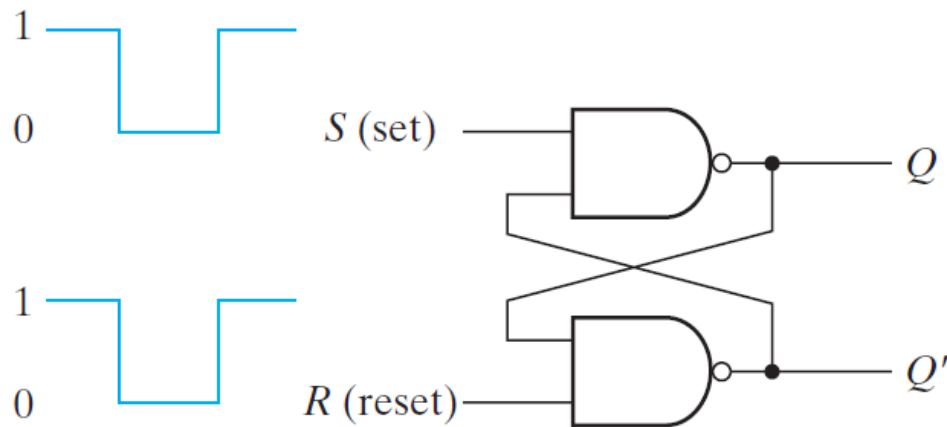
b) Tabla de función

FIGURA 5-3

Latch SR con compuertas NOR

Latches

- Latch S' - R'



S	R	Q	Q'
1	0	0	1
1	1	0	1
0	1	1	0
1	1	1	0
0	0	1	1

(Después de $S = 1, R = 0$)

(Después de $S = 0, R = 1$)

a) Diagrama lógico

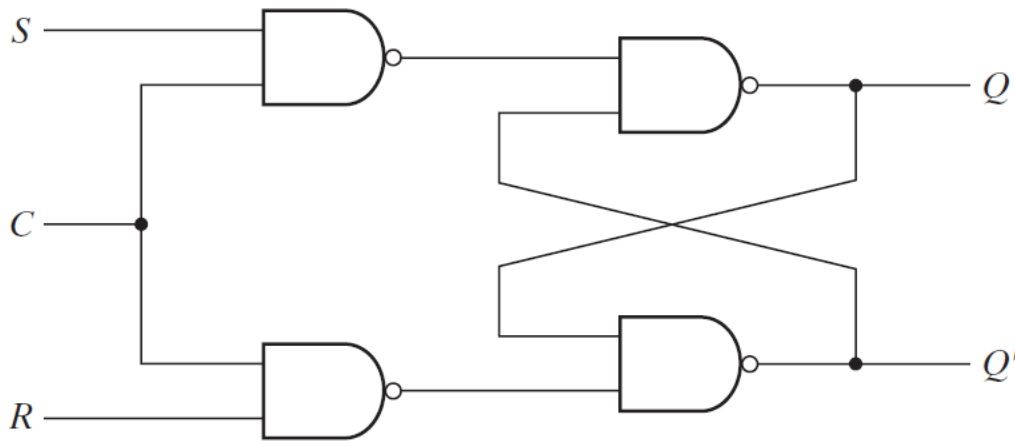
b) Tabla de función

FIGURA 5-4

Latch SR con compuertas NAND

Latches

- Latch S' - R' con Control



a) Diagrama lógico

C	S	R	Siguiente estado de Q
0	X	X	Sin cambio
1	0	0	Sin cambio
1	0	1	$Q = 0$; estado restablecido
1	1	0	$Q = 1$; estado establecido
1	1	1	Indeterminado

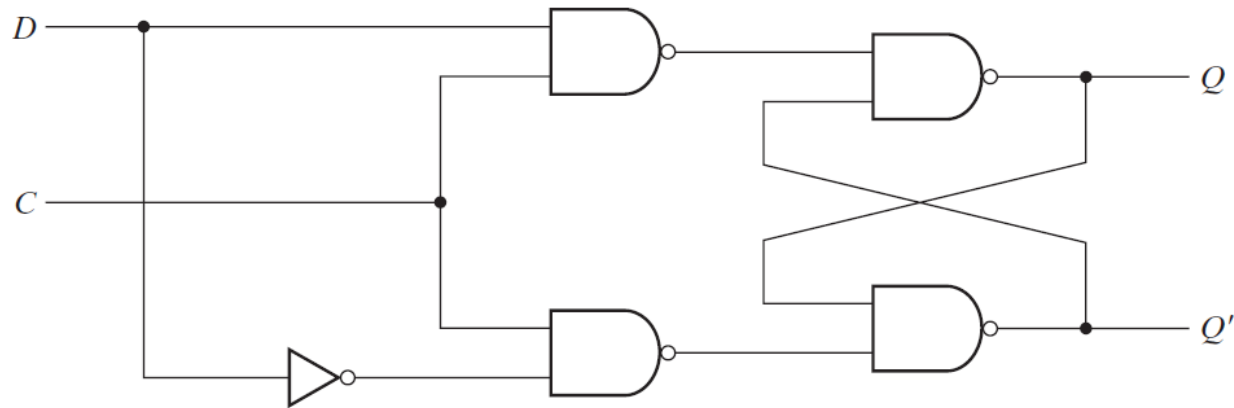
b) Tabla de función

FIGURA 5-5

Latch SR con entrada de control

Latches

- Latch D



a) Diagrama lógico

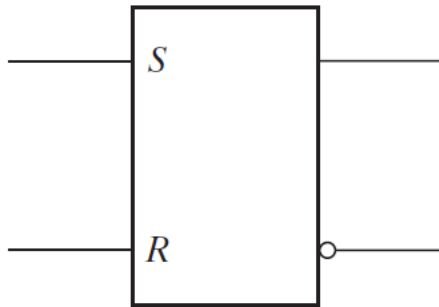
C	D	Siguiente estado de Q
0	X	Sin cambio
1	0	$Q = 0$; estado restablecido
1	1	$Q = 1$; estado establecido

b) Tabla de función

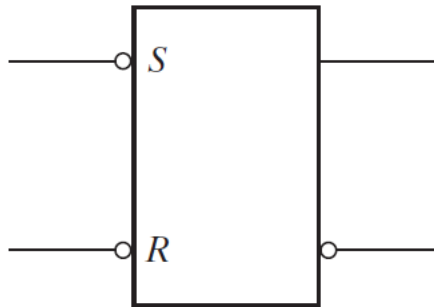
FIGURA 5-6
Latch D

Latches

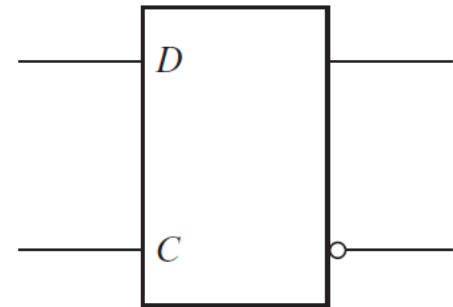
- Símbolos Gráficos



SR



\overline{SR}



D

FIGURA 5-7

Símbolos gráficos de latches

Flip-Flops

- Sensibilidad a niveles y a flancos



a) Respuesta al nivel positivo



b) Respuesta al borde positivo



c) Respuesta al borde negativo

FIGURA 5-8

Respuesta al reloj en un latch y un flip-flop

Flip-Flops

- Flip Flop D Maestro Esclavo

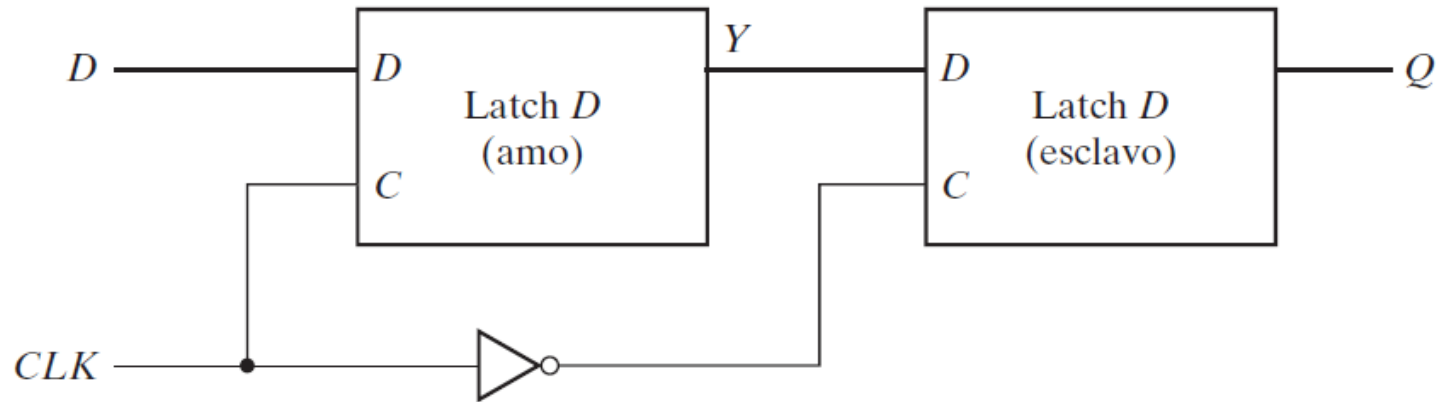


FIGURA 5-9

Flip-flop D amo-esclavo

Flip-Flops

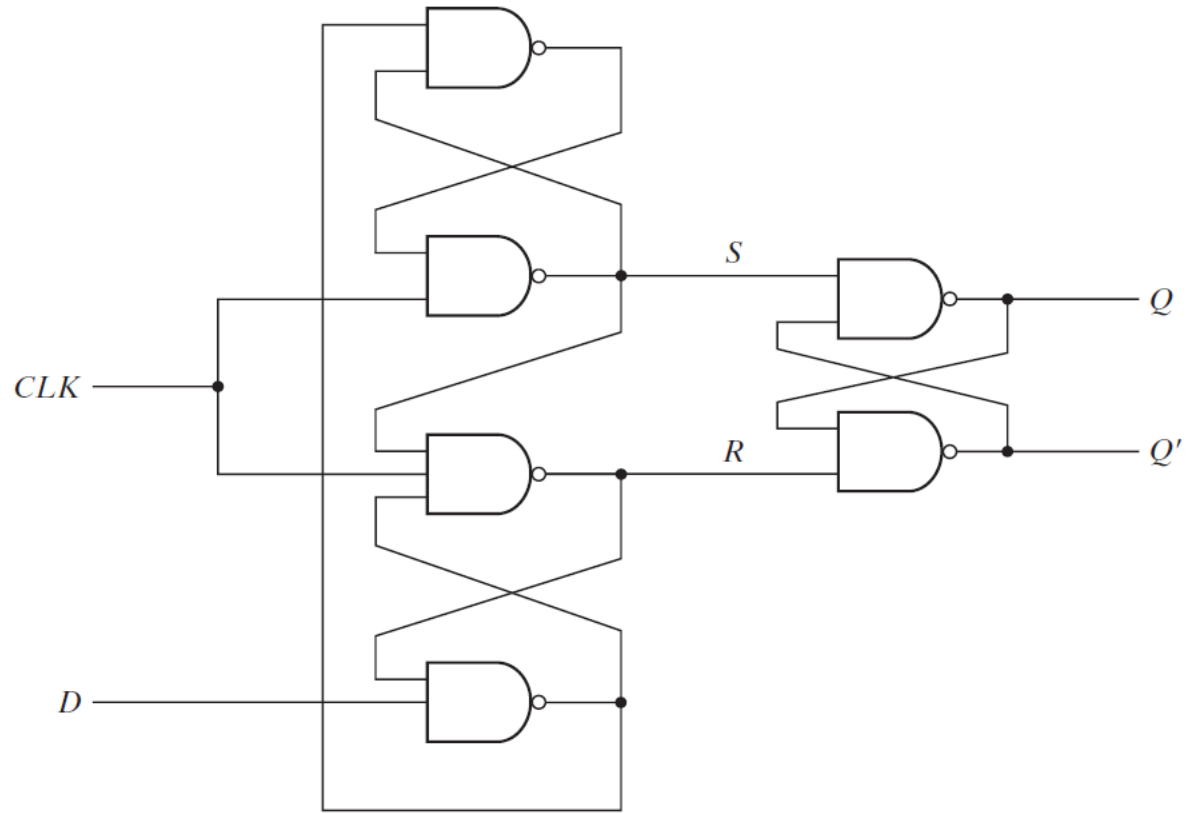
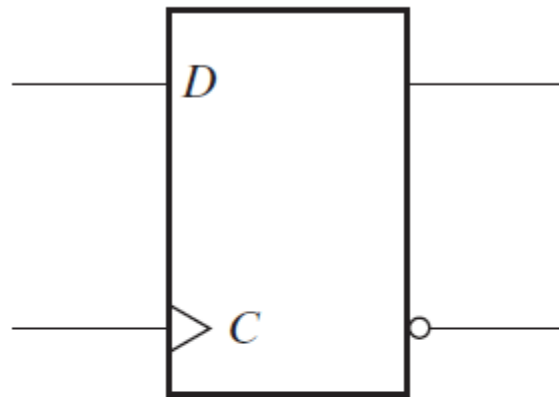


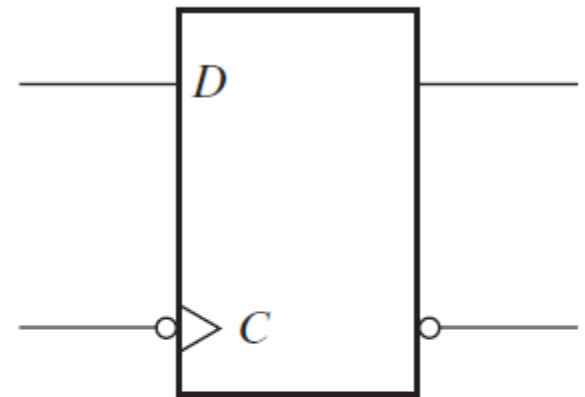
FIGURA 5-10

Flip-flop tipo D disparado por borde positivo

Flip-Flops



a) Borde positivo



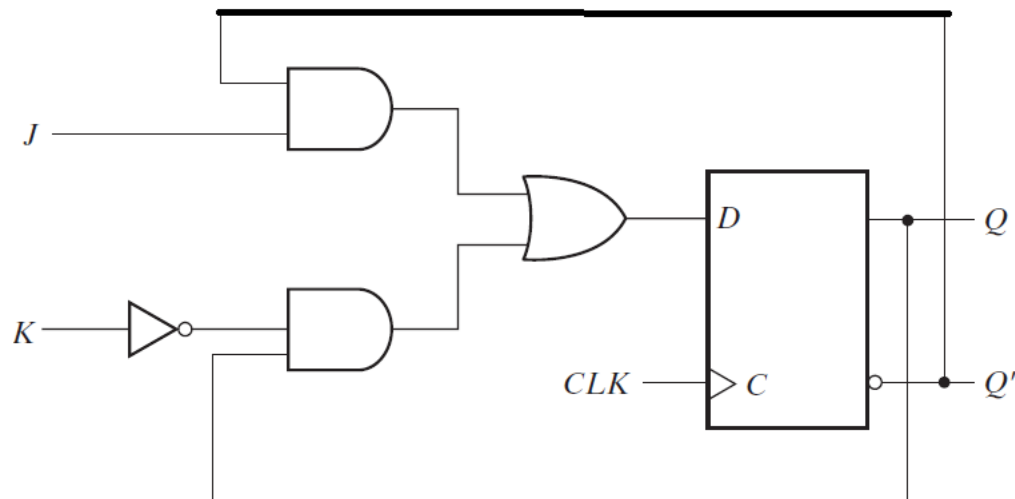
b) Borde negativo

FIGURA 5-11

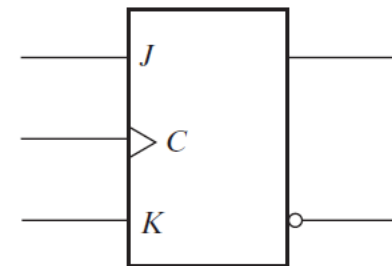
Símbolo gráfico para el flip-flop *D* disparado por borde

Flip-Flops

- Flip Flop J-k



a) Diagrama de circuito



b) Símbolo gráfico

FIGURA 5-12
Flip-Flop JK

$$D = JQ' + K'Q$$

Flip-Flops

- Flip Flop T

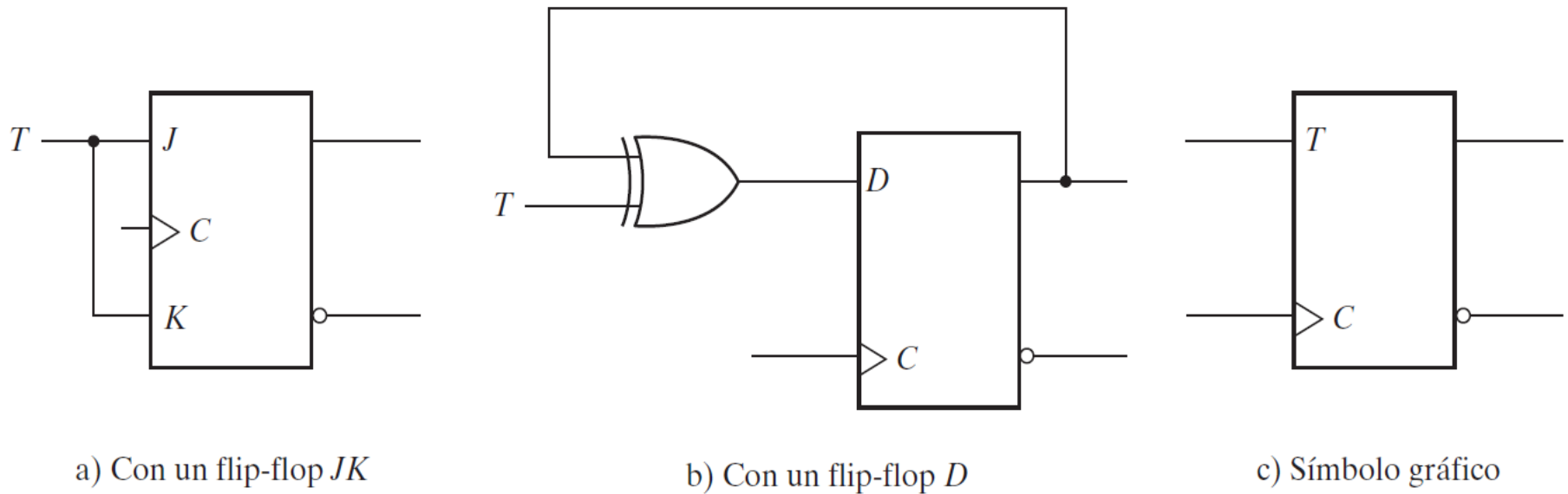


FIGURA 5-13
Flip-Flop T

$$D = T \oplus Q = TQ' + T'Q$$

Flip-Flops

- Tablas de Transición

Tabla 5-1

Tablas características de flip-flops

Flip-Flop JK			
J	K	$Q(t + 1)$	
0	0	$Q(t)$	Sin cambio
0	1	0	Restablecer
1	0	1	Establecer
1	1	$Q'(t)$	Complementar

Flip-Flop D		
D	$Q(t + 1)$	
0	0	Restablecer
1	1	Establecer

Flip-Flop T		
T	$Q(t + 1)$	
0	$Q(t)$	Sin cambio
1	$Q'(t)$	Complementar

Flip-Flops

- Ecuación Característica Flip Flop D

$$Q(t + 1) = D$$

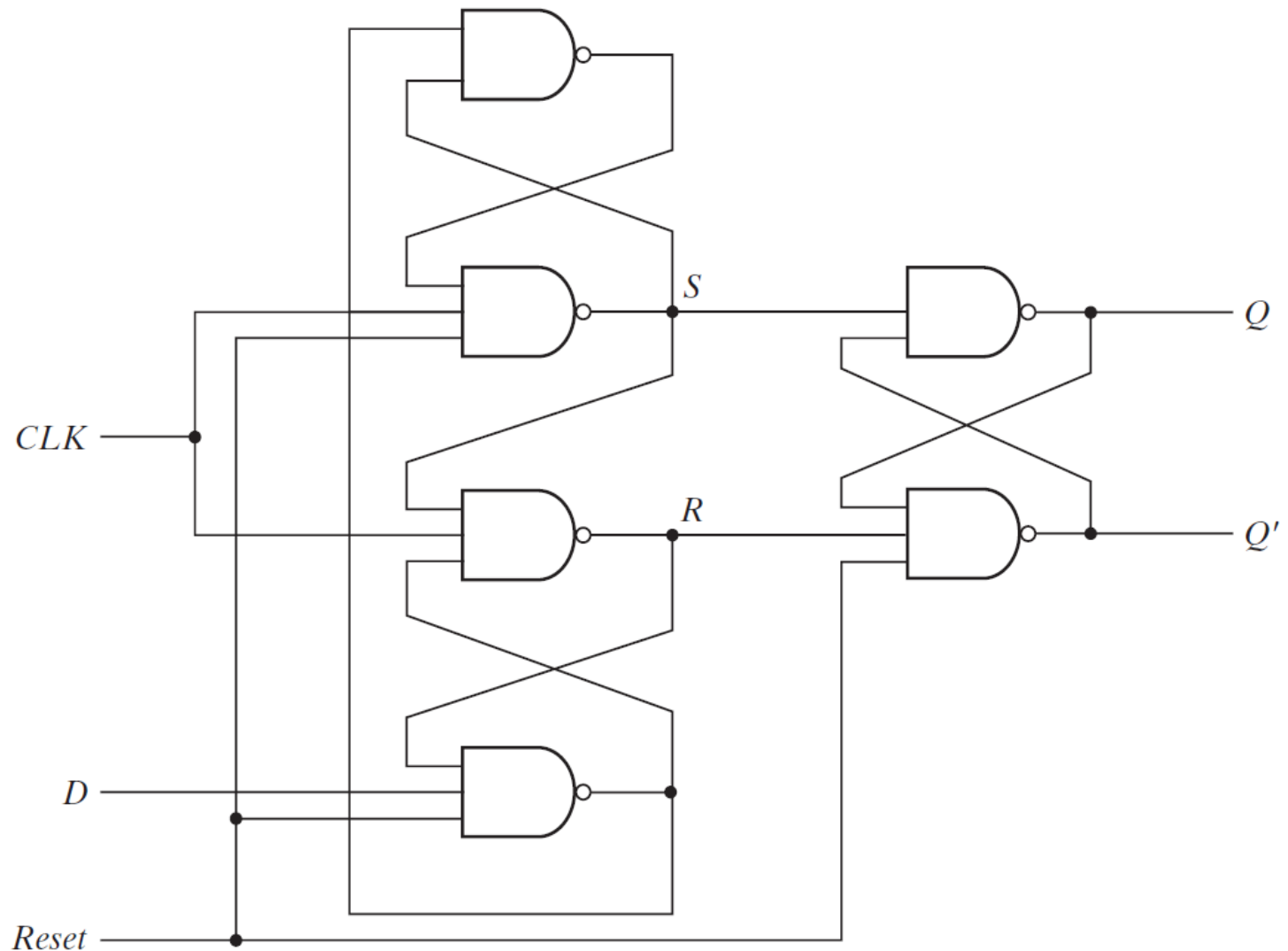
- Ecuación Característica Flip Flop J-k

$$Q(t + 1) = JQ' + K'Q$$

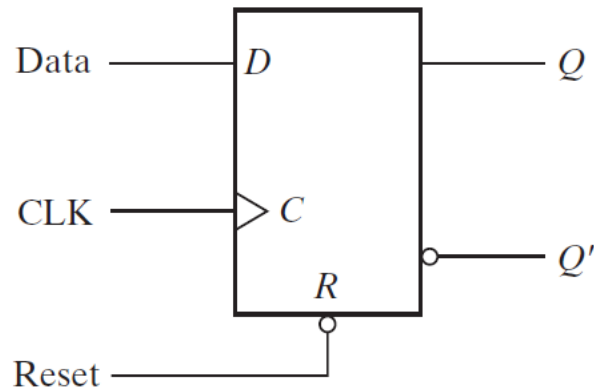
- Ecuación Característica Flip Flop T

$$Q(t + 1) = T \oplus Q = TQ' + T'Q$$

Flip-Flops con Reset asíncrono



Flip-Flops con Reset asíncrono



b) Símbolo gráfico

R	C	D	Q	Q'
0	X	X	0	1
1	\uparrow	0	0	1
1	\uparrow	1	1	0

c) Tabla de función

FIGURA 5-14

Flip-flop D con restablecimiento asincrónico

Análisis con Flip Flop D

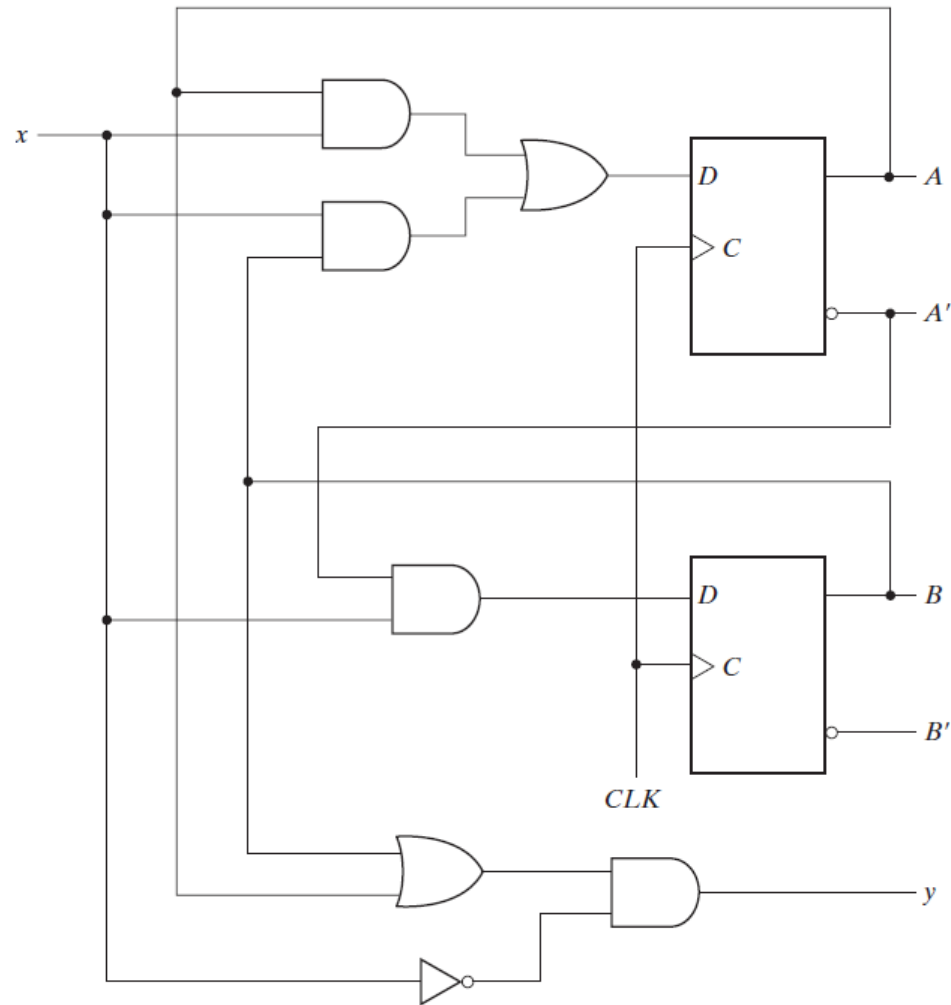


FIGURA 5-15
Ejemplo de circuito secuencial

Análisis con Flip Flop D

$$A(t + 1) = A(t)x(t) + B(t)x(t)$$

$$B(t + 1) = A'(t)x(t)$$

$$A(t + 1) = Ax + Bx$$

$$B(t + 1) = A'x$$

$$y(t) = [A(t) + B(t)]x'(t)$$

$$y = (A + B)x'$$

Análisis con Flip Flop D

- Tabla de Estados

Tabla 5-2

Tabla de estados para el circuito de la figura 5-15

Estado actual		Entrada	Siguiente estado		Salida
<i>A</i>	<i>B</i>		<i>A</i>	<i>B</i>	
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	1	0	0
1	1	0	0	0	1
1	1	1	1	0	0

Análisis con Flip Flop D

- Tabla de Estados

Tabla 5-3

Segunda forma de la tabla de estados

Estado actual	Siguiete estado		Salida	
	$x = 0$	$x = 1$	$x = 0$	$x = 1$
AB	AB	AB	y	y
00	00	01	0	0
01	00	11	1	0
10	00	10	1	0
11	00	10	1	0

Análisis de Circuitos Secuenciales

- Diagrama de Estados

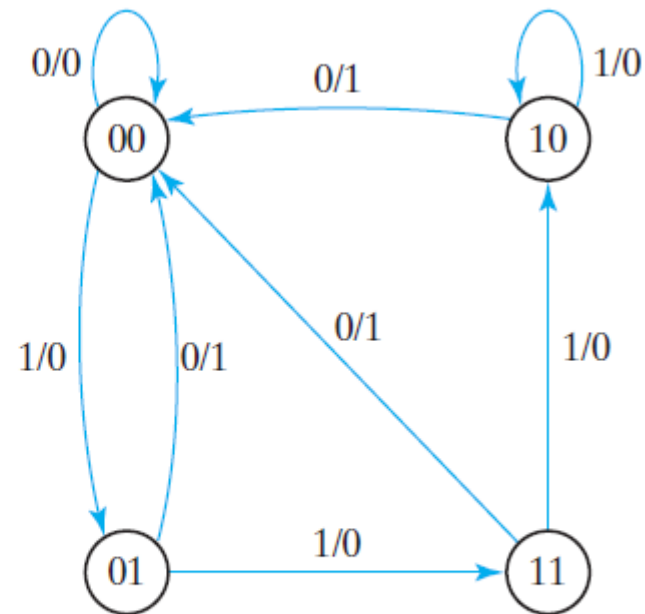


FIGURA 5-16

Diagrama de estados del circuito de la figura 5-15

Análisis de Circuitos Secuenciales

- Ecuaciones de Estado y Ecuación de Salida

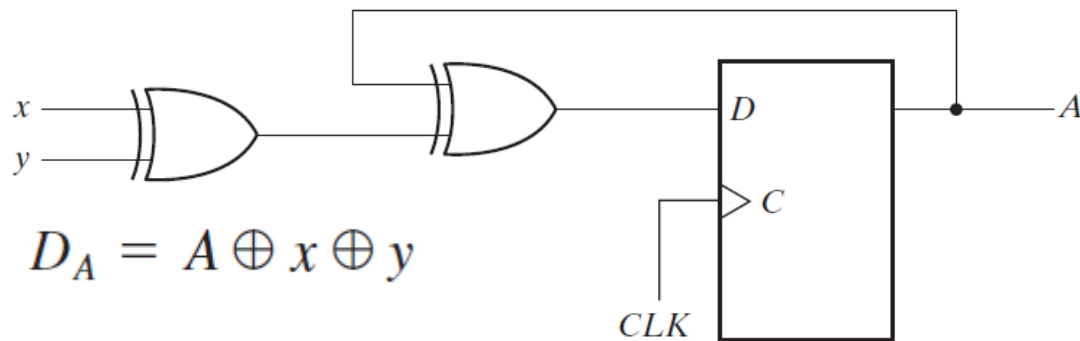
$$D_A = Ax + Bx$$

$$D_B = A'x$$

$$y = (A + B)x'$$

$$D: Q(t + 1) = D_Q.$$

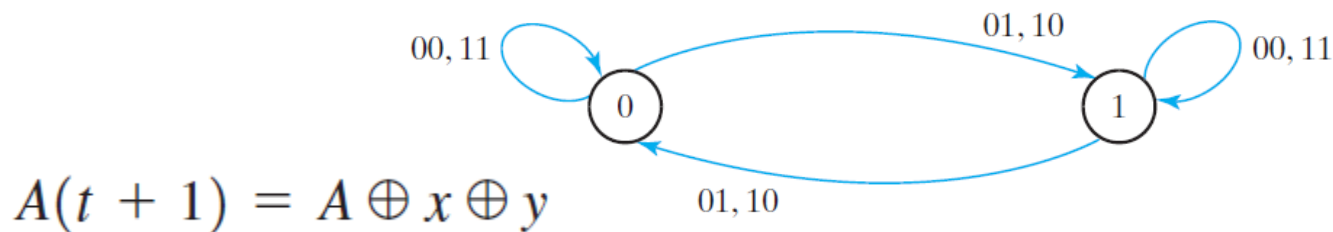
Análisis con Flip Flop D



a) Diagrama de circuito

Estado actual	Salidas		Siguiente estado
A	x	y	A
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

b) Tabla de estados



c) Diagrama de estados

FIGURA 5-17

Circuito secuencial con flip-flop D

Análisis con Flip Flop J-K

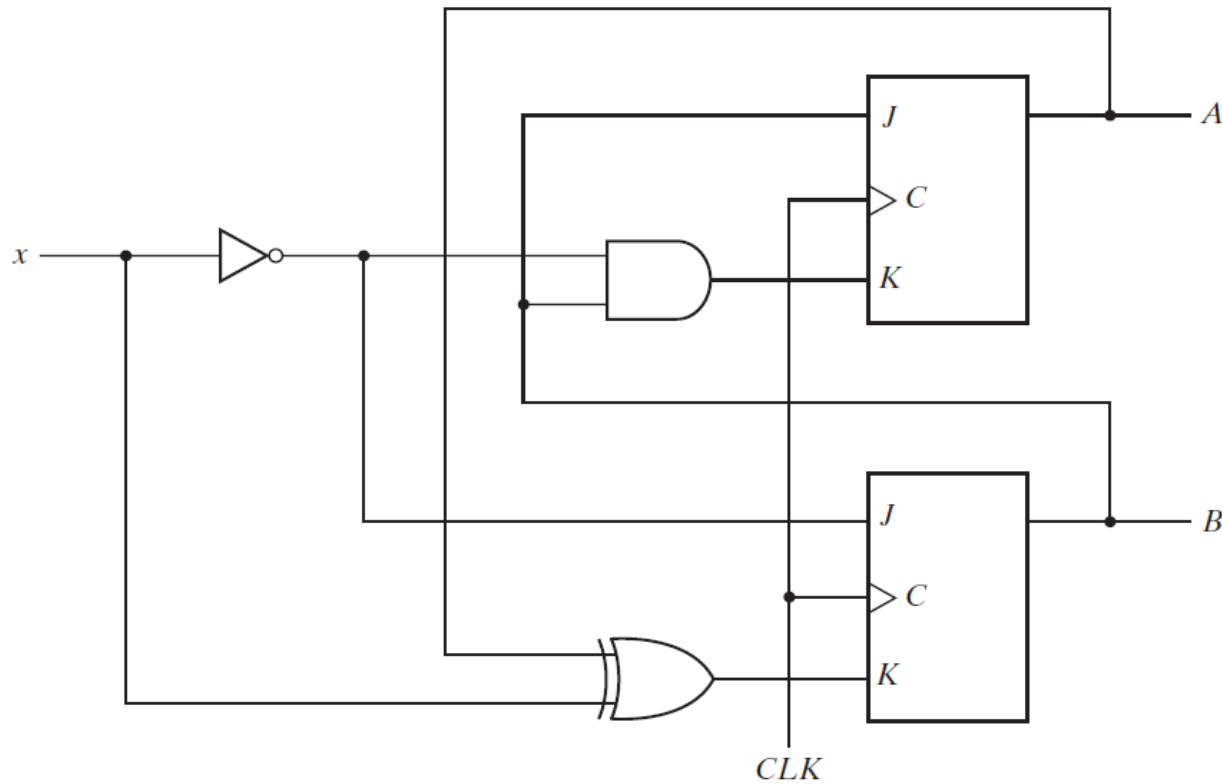


FIGURA 5-18

Circuito secuencial con flip-flop JK

$$\begin{aligned} J_A &= B & K_A &= Bx' \\ J_B &= x' & K_B &= A'x + Ax' = A \oplus x \end{aligned}$$

Análisis con Flip Flop J-K

$$J_A = B \quad K_A = Bx'$$

$$J_B = x' \quad K_B = A'x + Ax' = A \oplus x$$

$$A(t + 1) = JA' + K'A$$

$$B(t + 1) = JB' + K'B$$

$$A(t + 1) = BA' + (Bx')'A = A'B + AB' + Ax$$

$$B(t + 1) = x'B' + (A \oplus x)'B = B'x' + ABx + A'Bx'$$

Análisis con Flip Flop J-K

$$J_A = B \quad K_A = Bx'$$

$$J_B = x' \quad K_B = A'x + Ax' = A \oplus x$$

Tabla 5-4

Tabla de estados de un circuito secuencial con flip-flops JK

Estado Actual		Entrada	Siguiendo estado		Entradas de flip-flop			
A	B		A	B	J_A	K_A	J_B	K_B
0	0	0	0	1	0	0	1	0
0	0	1	0	0	0	0	0	1
0	1	0	1	1	1	1	1	0
0	1	1	1	0	1	0	0	1
1	0	0	1	1	0	0	1	1
1	0	1	1	0	0	0	0	0
1	1	0	0	0	1	1	1	1
1	1	1	1	1	1	0	0	0

Análisis con Flip Flop J-K

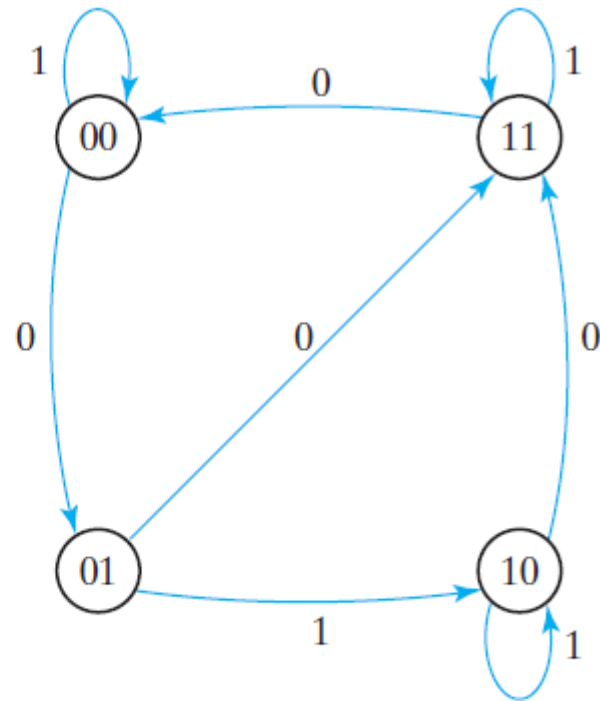
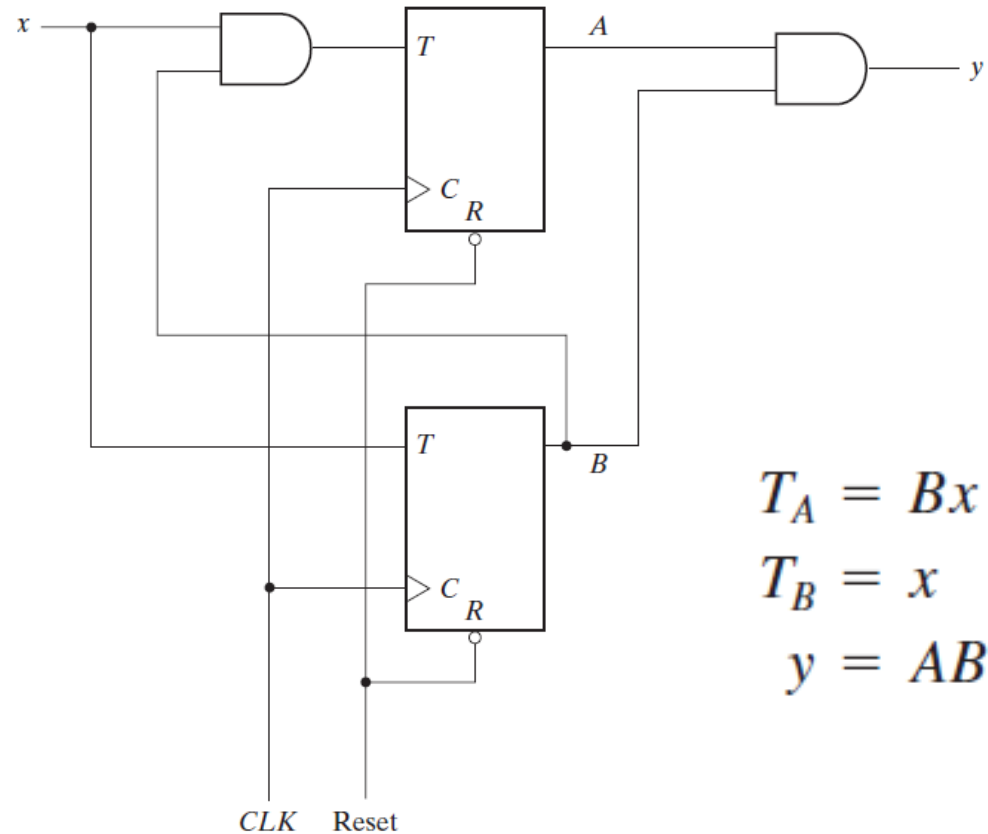


FIGURA 5-19

Diagrama de estados del circuito de la figura 5-18

Análisis con Flip Flop T



a) Diagrama de circuito

FIGURA 5-20

Circuito secuencial con dos flip-flops T

Análisis con Flip Flop T

$$Q(t + 1) = T \oplus Q = T'Q + TQ'$$

$$T_A = Bx$$

$$T_B = x$$

$$y = AB$$

$$A(t + 1) = (Bx)'A + (Bx)A' = AB' + Ax' + A'Bx$$

$$B(t + 1) = x \oplus B$$

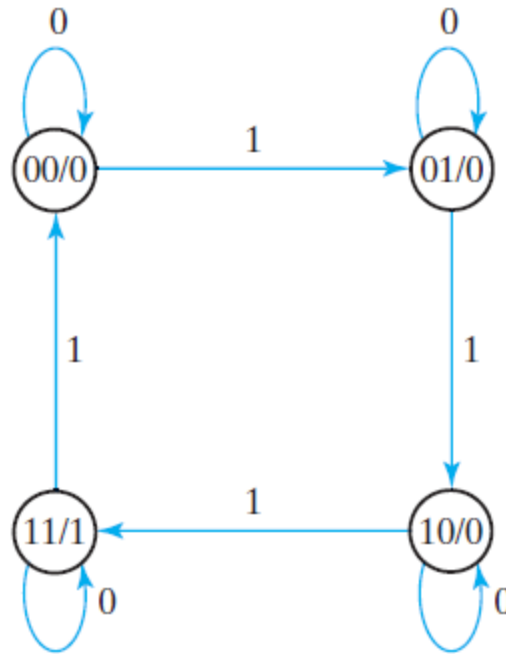
Análisis con Flip Flop T

Tabla 5-5

Tabla de estados para un circuito secuencial con flip-flops T

Estado actual		Entrada	Siguiente estado		Salida
<i>A</i>	<i>B</i>		<i>A</i>	<i>B</i>	
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	1	0
0	1	1	1	0	0
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	1

Análisis con Flip Flop T



b) Diagrama de estados

FIGURA 5-20

Circuito secuencial con dos flip-flops *T*

HDL para Secuenciales

```
initial
begin
    clock = 1'b0 ;
    repeat (30)
        #10 clock = ~ clock;
end
```

```
initial
begin
    clock = 1'b0;
    #300 $finish;
end
always
    #10 clock = ~clock;
```

always @ (expresión de control de sucesos)

Enunciados procedimentales de asignación.

HDL para Secuenciales

Ejemplo HDL 5-1

```
//Descripción de un latch D (Véase la figura 5-6)
module D_latch (Q,D,control);
    output Q;
    input D,control;
    reg Q;
    always @ (control or D)
        if (control) Q = D;      //Igual que: if (control == 1)
endmodule
```

Ejemplo HDL 5-2

```
//Flip-flop D
module D_FF (Q,D,CLK);
    output Q;
    input D,CLK;
    reg Q;
    always @ (posedge CLK)
        Q = D;
endmodule
```

HDL para Secuenciales

Ejemplo HDL 5-2

```
//Flip-flop D
module D_FF (Q,D,CLK);
    output Q;
    input D,CLK;
    reg Q;
    always @ (posedge CLK)
        Q = D;
endmodule
```

```
//Flip-flop D con restablecimiento asincrónico.
module DFF (Q,D,CLK,RST);
    output Q;
    input D,CLK,RST;
    reg Q;
    always @(posedge CLK or negedge RST)
        if (~RST) Q = 1'b0;    // Igual a: if (RST == 0)
        else Q = D;
endmodule
```

HDL para Secuenciales

Ejemplo HDL 5-3

```
//Flip-flop T hecho con flip-flop D y compuertas
module TFF (Q,T,CLK,RST);
    output Q;
    input T,CLK,RST;
    wire DT;
    assign DT = Q ^ T ;
    //Crear ejemplar del flip-flop D
    DFF TF1 (Q,DT,CLK,RST);
endmodule
```

```
//Flip-flop JK hecho con flip-flop D y compuertas
module JKFF (Q,J,K,CLK,RST);
    output Q;
    input J,K,CLK,RST;
    wire JK;
    assign JK = (J & ~Q) | (~K & Q);
    //Crear ejemplar de flip-flop D
    DFF JK1 (Q,JK,CLK,RST);
endmodule
```

HDL para Secuenciales

```
//Flip-flop JK hecho con flip-flop D y compuertas
module JKFF (Q,J,K,CLK,RST);
    output Q;
    input J,K,CLK,RST;
    wire JK;
    assign JK = (J & ~Q) | (~K & Q);
    //Crear ejemplar de flip-flop D
    DFF JK1 (Q,JK,CLK,RST);
endmodule
```

```
//Flip-flop D
module DFF (Q,D,CLK,RST);
    output Q;
    input D,CLK,RST;
    reg Q;
    always @ (posedge CLK or negedge RST)
        if (~RST) Q = 1'b0;
        else Q = D;
endmodule
```

HDL para Secuenciales

Ejemplo HDL 5-4

```
//Descripción funcional de flip-flop JK
module JK_FF (J,K,CLK,Q,Qnot);
    output Q,Qnot;
    input J,K,CLK;
    reg Q;
    assign Qnot = ~ Q ;
    always @ (posedge CLK)
        case ({J,K})
            2'b00: Q = Q;
            2'b01: Q = 1'b0;
            2'b10: Q = 1'b1;
            2'b11: Q = ~ Q;
        endcase
endmodule
```

HDL para Secuenciales

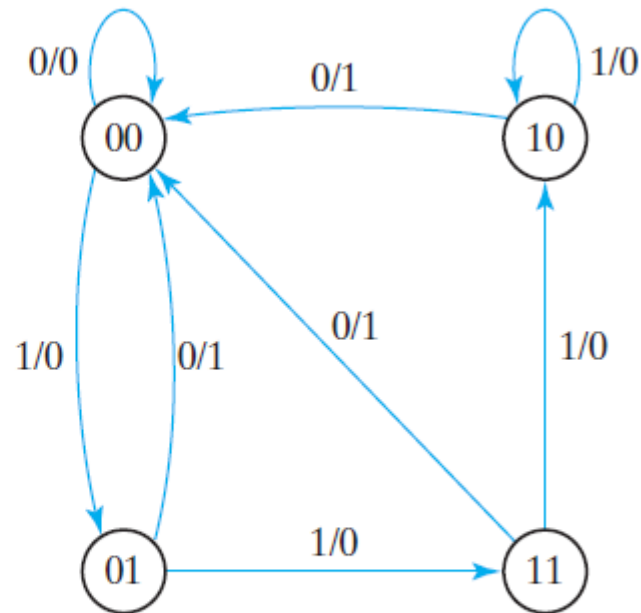


FIGURA 5-16

Diagrama de estados del circuito de la figura 5-15

HDL para Secuenciales

Ejemplo HDL 5-5

```
//Diagrama de estados Mealy (figura 5-16)
module Mealy_mdl (x,y,CLK,RST);
    input x,CLK,RST;
    output y;
    reg y;
    reg [1:0] Prstate, Nxtstate;
    parameter S0 = 2'b00, S1 = 2'b01, S2 = 2'b10, S3 = 2'b11;
    always @ (posedge CLK or negedge RST)
        if (~RST) Prstate = S0; //Iniciar en estado S0
        else Prstate = Nxtstate; //Operaciones de reloj
    always @ (Prstate or x) //Determinar siguiente estado
        case (Prstate)
            S0: if (x) Nxtstate = S1;
                else Nxtstate = S0;
            S1: if (x) Nxtstate = S3;
                else Nxtstate = S0;
            S2: if (~x)Nxtstate = S0;
                else Nxtstate = S2;
            S3: if (x) Nxtstate = S2;
                else Nxtstate = S0;
        endcase
    always @ (Prstate or x) //Evaluar salida
        case (Prstate)
            S0: y = 0;
            S1: if (x) y = 1'b0; else y = 1'b1;
            S2: if (x) y = 1'b0; else y = 1'b1;
            S3: if (x) y = 1'b0; else y = 1'b1;
        endcase
endmodule
```


HDL para Secuenciales

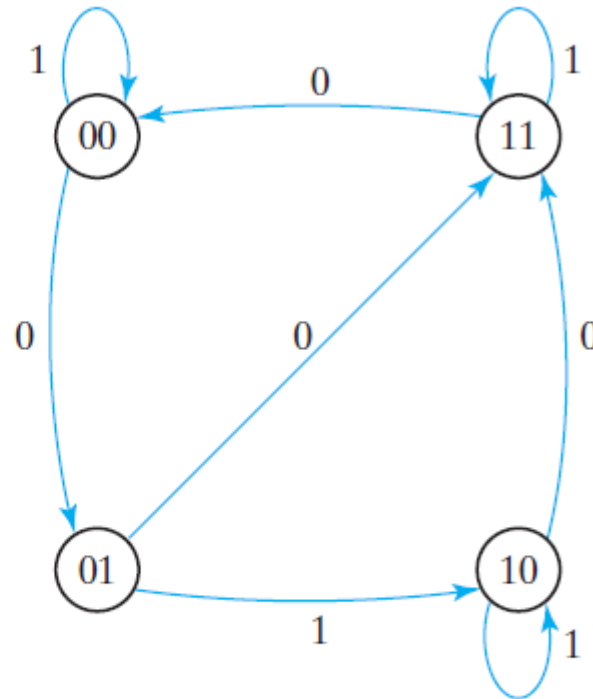


FIGURA 5-19

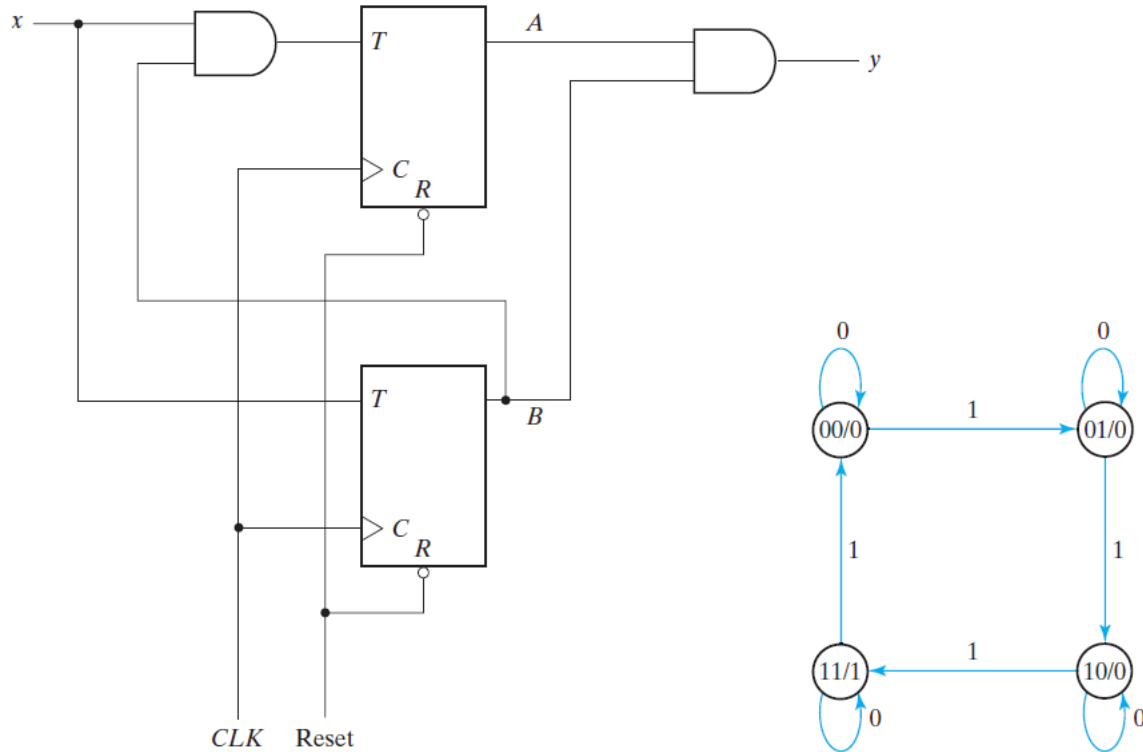
Diagrama de estados del circuito de la figura 5-18

HDL para Secuenciales

Ejemplo HDL 5-6

```
//Diagrama de estados de Moore (figura 5-19)
module Moore_md1 (x,AB,CLK,RST);
    input x,CLK,RST;
    output [1:0]AB;
    reg [1:0] state;
    parameter S0 = 2'b00, S1 = 2'b01, S2 = 2'b10, S3 = 2'b11;
    always @ (posedge CLK or negedge RST)
        if (~RST) state = S0; //Iniciar en estado S0
        else
            case (state)
                S0: if (~x) state = S1; else state = S0;
                S1: if (x) state = S2; else state = S3;
                S2: if (~x) state = S3; else state = S2;
                S3: if (~x) state = S0; else state = S3;
            endcase
        assign AB = state; //Salida de flip-flops
endmodule
```

HDL para Secuenciales



a) Diagrama de circuito

b) Diagrama de estados

FIGURA 5-20

Circuito secuencial con dos flip-flops T

HDL para Secuenciales

Ejemplo HDL 5-7

```
//Descripción estructural de circuito secuencial
//Véase la figura 5-20a)
module Tcircuit (x,y,A,B,CLK,RST);
    input x,CLK,RST;
    output y,A,B;
    wire TA,TB;
//Ecuaciones de entrada de flip-flop
    assign TB = x,
           TA = x & B;
//Ecuación de salida
    assign y = A & B;
//Se crean ejemplares de flip-flops T
    T_FF BF (B,TB,CLK,RST);
    T_FF AF (A,TA,CLK,RST);
endmodule

//Flip-flop T
module T_FF (Q,T,CLK,RST);
    output Q;
    input T,CLK,RST;
    reg Q;
    always @ (posedge CLK or negedge RST)
        if (~RST) Q = 1'b0;
        else Q = Q ^ T;
endmodule
```

HDL para Secuenciales

```
//Estímulo para probar el circuito secuencial
module testTcircuit;
    reg x,CLK,RST;    //entradas del circuito
    wire y,A,B;       //salida del circuito
    Tcircuit TC (x,y,A,B,CLK,RST);    // se crea un ejemplar
                                        // del circuito

    initial
        begin
            RST = 0;
            CLK = 0;
            #5 RST = 1;
            repeat (16)
                #5 CLK = ~CLK;
        end
    initial
        begin
            x = 0;
            #15 x = 1;
            repeat (8)
                #10 x = ~ x;
        end
endmodule
```

HDL para Secuenciales

198 Capítulo 5 Lógica secuencial síncrona

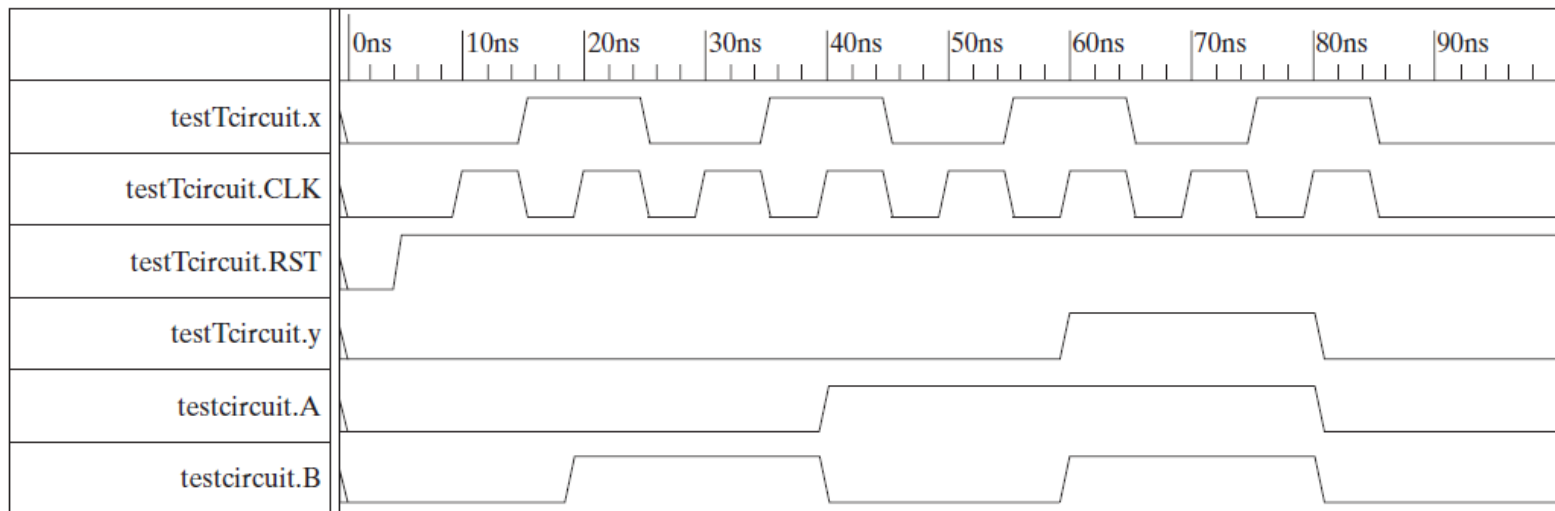


FIGURA 5-21

Salida de la simulación del ejemplo HDL 5-7

Proceso de Diseño con FF D

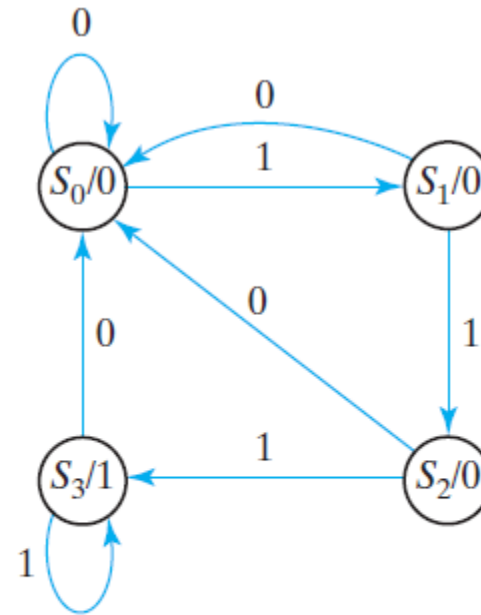


FIGURA 5-24

Diagrama de estados para el detector de sucesiones

Proceso de Diseño con FF D

Tabla 5-11

Tabla de estados para el detector de sucesiones

Estado actual		Entrada	Siguiete Estado		Salida
A	B		A	B	
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	1	1	0
1	1	0	0	0	1
1	1	1	1	1	1

Proceso de Diseño con FF D

$$A(t + 1) = D_A(A, B, x) = \Sigma(3, 5, 7)$$

$$B(t + 1) = D_B(A, B, x) = \Sigma(1, 5, 7)$$

$$y(A, B, x) = \Sigma(6, 7)$$

$$D_A = Ax + Bx$$

$$D_B = Ax + B'x$$

$$y = AB$$

Proceso de Diseño con FF D

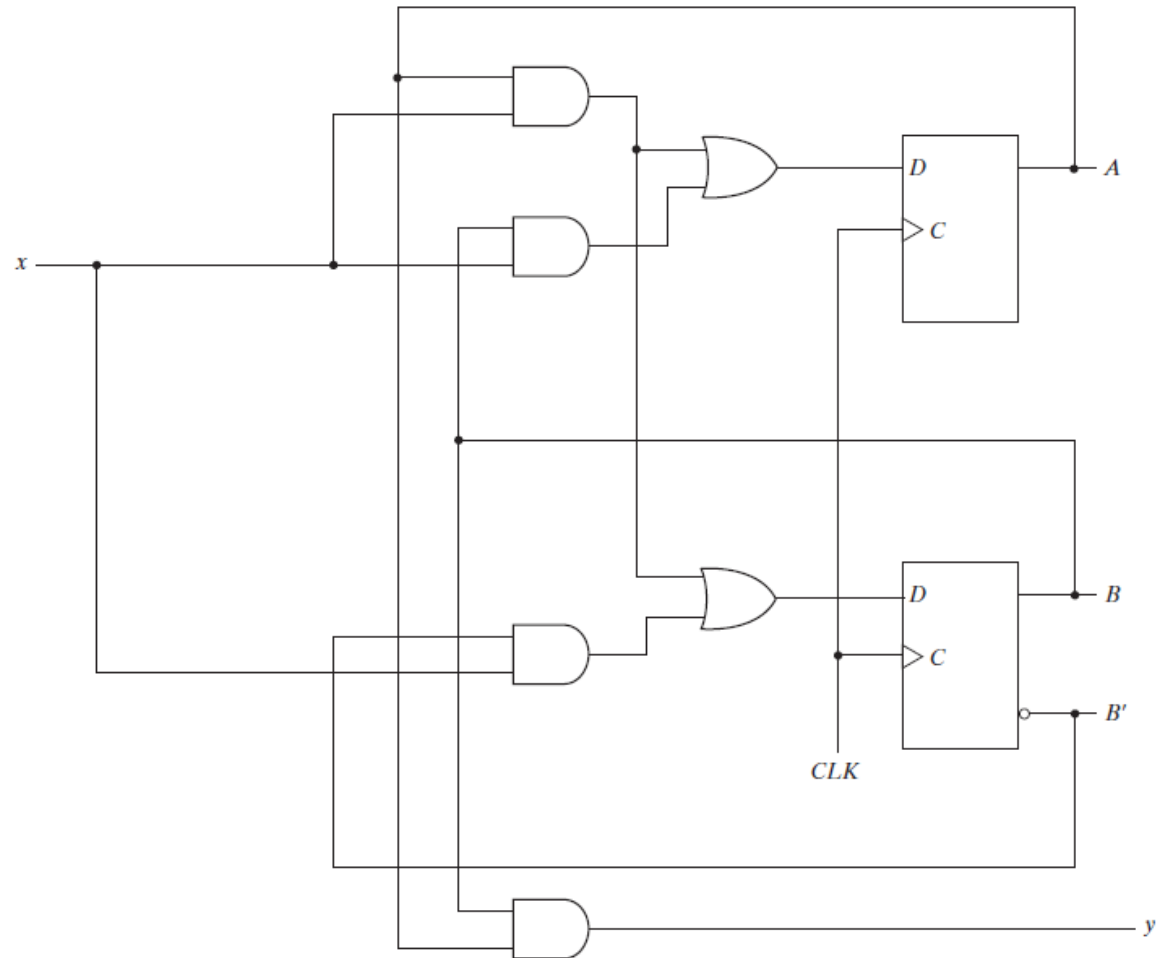


FIGURA 5-26
Diagrama lógico del detector de sucesiones

Proceso de Diseño con FF JK

Tabla 5-12

Tablas de excitación de flip-flops

$Q(t)$	$Q(t + 1)$	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

a) JK

$Q(t)$	$Q(t + 1)$	T
0	0	0
0	1	1
1	0	1
1	1	0

b) T

Proceso de Diseño con FF JK

Tabla 5-13

Tabla de estados y entradas de flip-flops JK

Estado Actual		Entrada	Siguiete Estado		Entradas del flip-flop			
<i>A</i>	<i>B</i>	<i>x</i>	<i>A</i>	<i>B</i>	<i>J_A</i>	<i>K_A</i>	<i>J_B</i>	<i>K_B</i>
0	0	0	0	0	0	X	0	X
0	0	1	0	1	0	X	1	X
0	1	0	1	0	1	X	X	1
0	1	1	0	1	0	X	X	0
1	0	0	1	0	X	0	0	X
1	0	1	1	1	X	0	1	X
1	1	0	1	1	X	0	X	0
1	1	1	0	0	X	1	X	1

Proceso de Diseño con FF JK

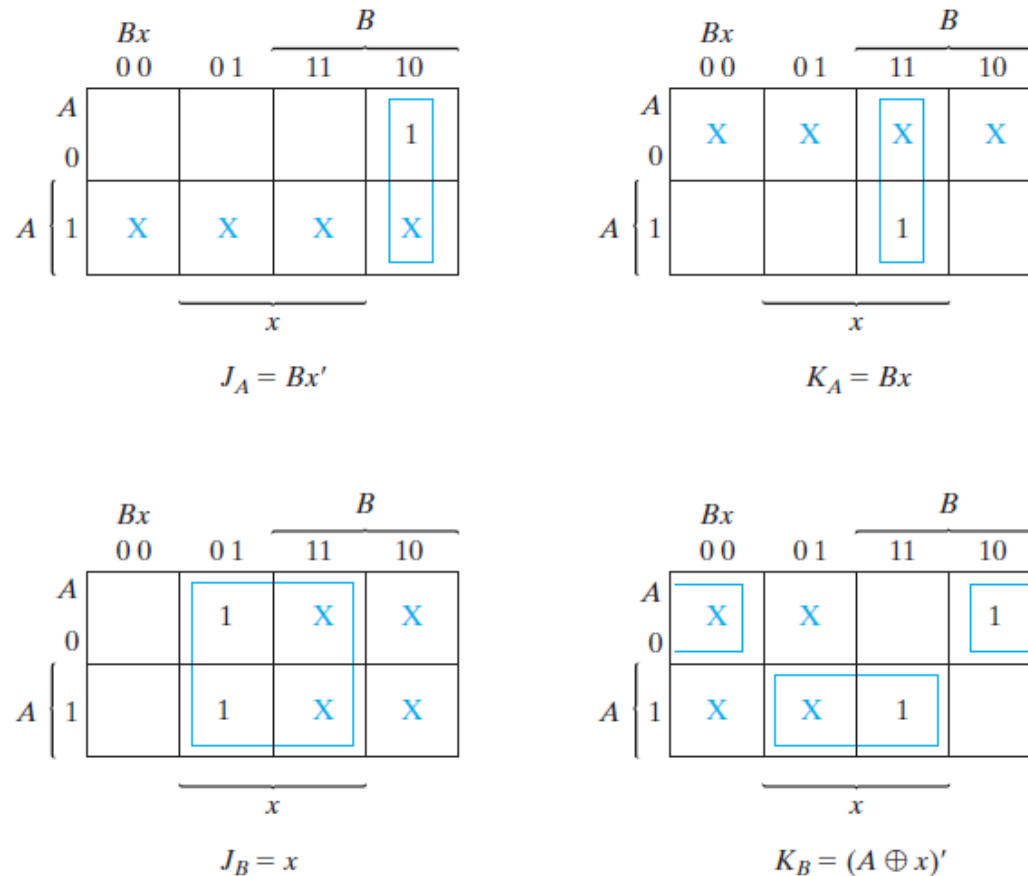


FIGURA 5-27

Mapas para las ecuaciones de entrada J y K

Proceso de Diseño con FF JK

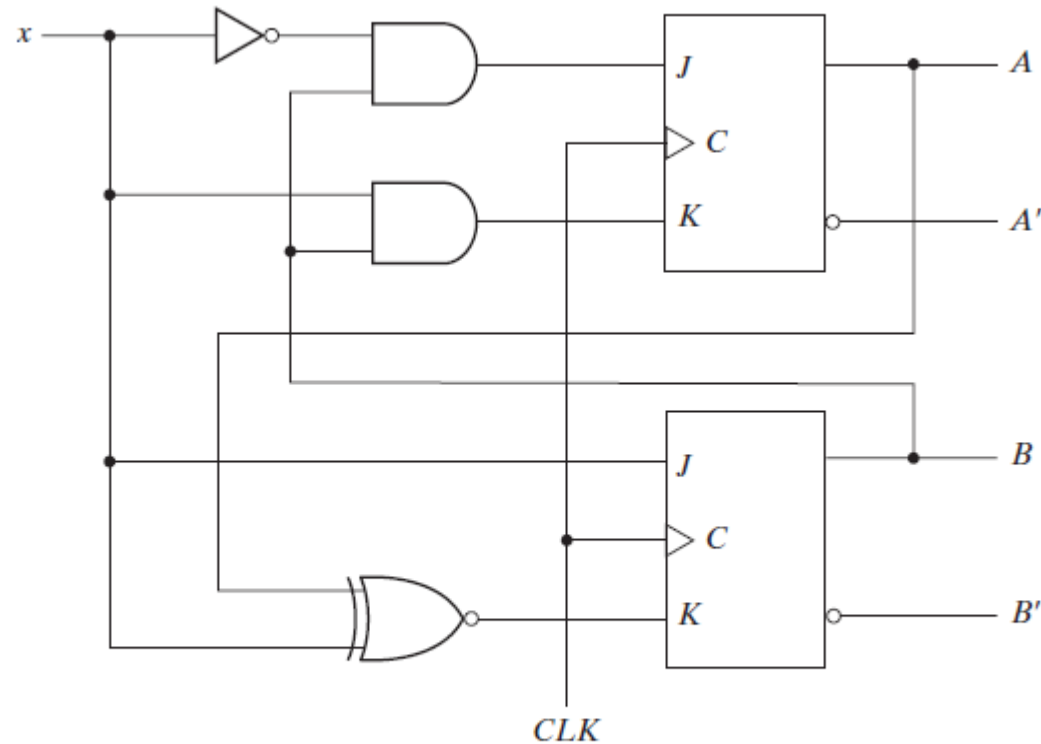


FIGURA 5-28

Diagrama lógico para el circuito secuencial con flip-flops JK

Proceso de Diseño con FF T

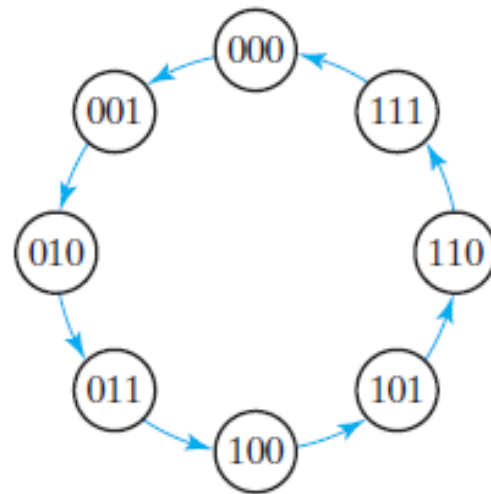


FIGURA 5-29

Diagrama de estados de un contador binario de tres bits

Proceso de Diseño con FF T

Tabla 5-14

Tabla de estados para el contador de tres bits

Estado actual			Siguiente estado			Entradas de los flip-flops		
A_2	A_1	A_0	A_2	A_1	A_0	T_{A2}	T_{A1}	T_{A0}
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	1	1	1
1	0	0	1	0	1	0	0	1
1	0	1	1	1	0	0	1	1
1	1	0	1	1	1	0	1	1
1	1	1	0	0	0	1	1	1

Proceso de Diseño con FF T

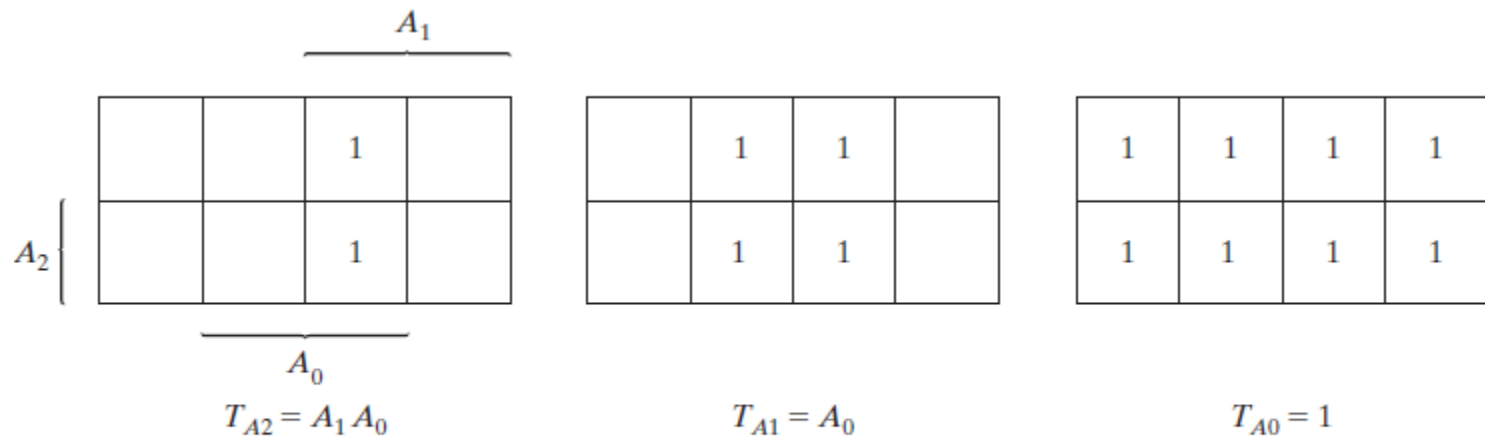


FIGURA 5-30

Mapas para el contador binario de tres bits

Proceso de Diseño con FF T

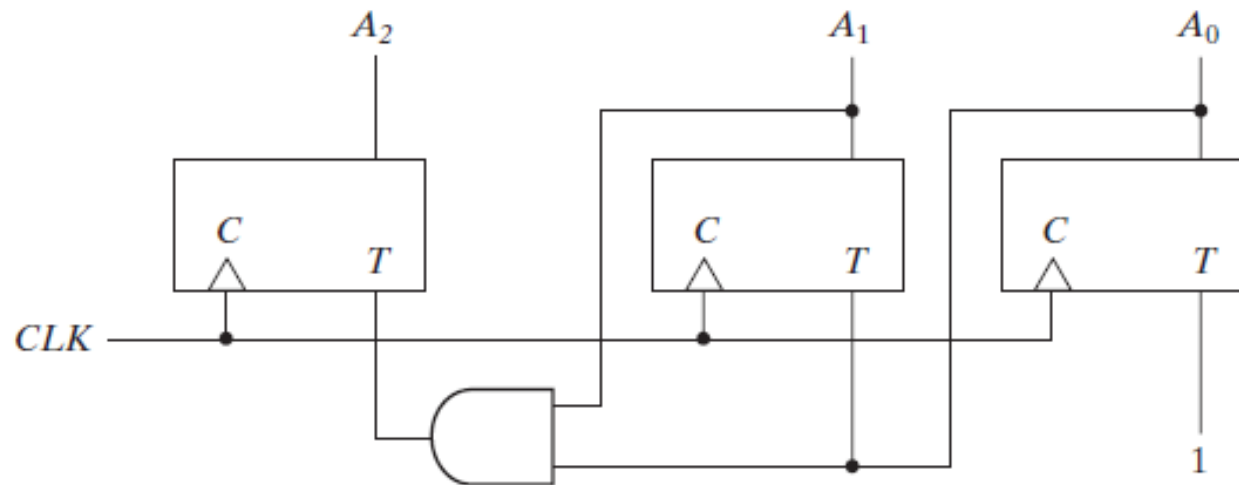


FIGURA 5-31

Diagrama lógico del contador de tres bits