

# Diseño o Síntesis de Circuitos Secuenciales

- 1\_ Entender bien el enunciado del problema. Si hay dudas de lo que se pide, preguntar antes de empezar a solucionarlo.
- 2\_ Decidir si es tipo Mealy o tipo Moore (Normalmente se pide tipo Moore). (El enunciado generalmente lo indica, si no se dice nada elegir siempre tipo Moore).
- 3\_ Identificar las entradas, las salidas, los estados y las variables de estados, nombrarlos y codificarlos.
- 4\_ Hacer Diagrama de Estados.
- 5\_ Elegir o Definir el tipo de Flip Flops a utilizar. Si no se dice nada elegir siempre Flip Flop tipo D.
- 6\_ Encontrar ecuaciones para el combinacional de estados y para el combinacional de salidas.
- 7\_ Seleccionar Tecnologías para implementar los combinacionales e Implementar el circuito final.

# Diseño o Síntesis de Circuitos Secuenciales

- Ejemplo 01:
  - Diseñar e implementar un circuito secuencial de Mealy, que cumpla con las especificaciones descritas más abajo. Implementar con Flip-Flops tipo D, y compuertas de cualquier tipo, a su elección. La especificaciones de transiciones de estados y de valor de salida son las siguientes:
  - Al arrancar luego de un reset tipo asíncrono, desde un estado inicial, el circuito sensa (en cada flanco ascendente de reloj) la llegada de tres o más unos consecutivos a su entrada. Dicha entrada es una señal de un bit denominada «x».
  - Mientras van entrando los tres unos consecutivos en «x», la salida «y» se mantiene en 0 y la máquina va pasando al siguiente estado, hasta que entran al menos tres unos consecutivos.

# Diseño o Síntesis de Circuitos Secuenciales

- Ejemplo 01: (Continuación)
  - Si mientras va esperando los tres unos consecutivos, la entrada «x» se pone a 0, cortando la secuencia de unos, entonces la salida «y» se pone a 1, y la máquina regresa al inicial, empezando todo de nuevo.
  - Si luego de entrar los tres unos consecutivos siguen entrando unos por la entrada «x» el comportamiento sigue igual respecto de la salida.
  - Si ya entraron al menos tres unos consecutivos, y entra por «x» un 0, la máquina de Mealy saca por su salida «y» un 1, y regresa al estado inicial empezando todo de nuevo.

# Diseño o Síntesis de Circuitos Secuenciales

- Paso 01: (Entender bien el problema)
  - Si está en su primer estado (estado inicial luego del reset) y su entrada es 0, entonces su salida es 0, y permanece en dicho primer estado.
  - Sino, si está en su primer estado y su entrada es 1, entonces su salida es 0, y pasa al segundo estado, registrando de esta manera la llegada del primer uno.
  - Sino, si está en su segundo estado y su entrada es 0, entonces su salida es 1, y regresa al primer estado porque se cortó la secuencia.
  - Sino, si está en su segundo estado y su entrada es 1, entonces su salida es 0, y avanza hacia el tercer estado, registrando de esta manera la llegada del segundo 1 consecutivo.

# Diseño o Síntesis de Circuitos Secuenciales

- Paso 01: Entender bien el problema (Continuación)
  - Sino, si está en su tercer estado y su entrada es 0, entonces su salida es 1 y regresa al primer estado, porque se cortó la secuencia.
  - Sino, si está en su tercer estado y su entrada es 1, entonces su salida es 0 y pasa a su cuarto estado, registrando de esta manera la llegada del tercer uno consecutivo.
  - Sino, si está en su cuarto estado estado y su entrada es 0, entonces su salida es 1 y regresa a su primer estado.
  - Sino, si está en su cuarto estado y su entrada es 1, entonces su salida es 0 y pasa a su cuarto estado.

# Diseño o Síntesis de Circuitos Secuenciales

- Paso 02: Definir si es Mealy o Moore
- En realidad esto se indica en el enunciado del problema como en este caso. Si no se dice nada siempre se prefiere tipo Moore.
- Recordar que en el tipo Moore las salidas, en un tiempo dado, solo dependen del estado.
- En cambio en el tipo Mealy, las salidas en un tiempo dado, dependen del estado y las entradas en ese instante.
  - Antes la duda elegir tipo Moore, pero en este caso se pidió tipo Mealy, así que recordar que este ejemplo es un secuencial tipo Mealy.

# Diseño o Síntesis de Circuitos Secuenciales

- Paso 03: Identificar las entradas, las salidas, los estados y las variables de estados, nombrarlos y codificarlos.
  - En este caso está claro que hay una entrada de un bit que es «x», una salida de un bit que es «y».
  - También está claro que hay cuatro estados, y por ende hay dos flip flops como mínimo. Cada salida Q de cada flip flop será una variable de estados.
  - Le llamaremos a la salida Q del primer flip será A, y la salida Q del segundo flip flop será llamada B.

# Diseño o Síntesis de Circuitos Secuenciales

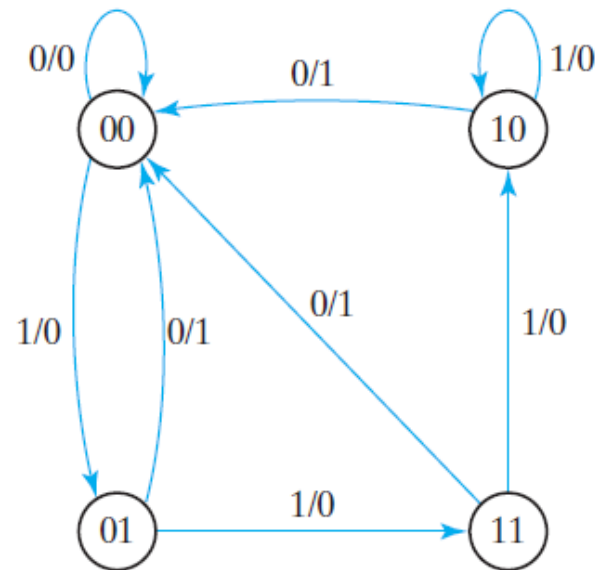
- Paso 03: Identificar las entradas, las salidas, los estados y las variables de estados, nombrarlos y codificarlos. (Continuación)
- Los estados se codificarán según la siguiente tabla:
- Notar que la codificación de estados es totalmente arbitraria y aquí por ejemplo, al tercer estado se codifico con (11) y al cuarto con (10).

A	B	Estado	Código de Estado (AB)
0	0	Primero	00
0	1	Segundo	01
1	1	Tercero	11
1	0	Cuarto	10



# Diseño o Síntesis de Circuitos Secuenciales

- Paso 04: Hacer Diagrama de estados



**FIGURA 5-16**

Diagrama de estados del circuito de la figura 5-15

# Diseño o Síntesis de Circuitos Secuenciales

- Paso 05: Elegir el tipo de flip flop a utilizar.
- Siempre por defecto elegimos primero al flip flop tipo D.
- De todas maneras aunque nos pidieran otro tipo de flip flop siempre partimos de un tipo D, resolvemos para tipo D, y luego modificamos las ecuaciones como se indicará después.
- En resumen, elegir siempre primero tipo D, y resolver primero para tipo D. Si se piden otro tipo de flip flop, se pueden transformar después las ecuaciones.

# Diseño o Síntesis de Circuitos Secuenciales

- Paso 06: Encontrar las ecuaciones del combinacional de estados y del combinacional de salida.

**Tabla 5-3**

*Segunda forma de la tabla de estados*

Estado actual	Siguiente 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

# Diseño o Síntesis de Circuitos Secuenciales

- Paso 06: Encontrar las ecuaciones del combinacional de estados y combinacioal de salida.

**Tabla 5-2**

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

<b>Estado actual</b>		<b>Entrada</b>	<b>Siguiente estado</b>		<b>Salida</b>
<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

# Diseño o Síntesis de Circuitos Secuenciales

- Paso 06: Encontrar las ecuaciones del combinacional de estados y combinacional de salida.

$$D_A = Ax + Bx$$

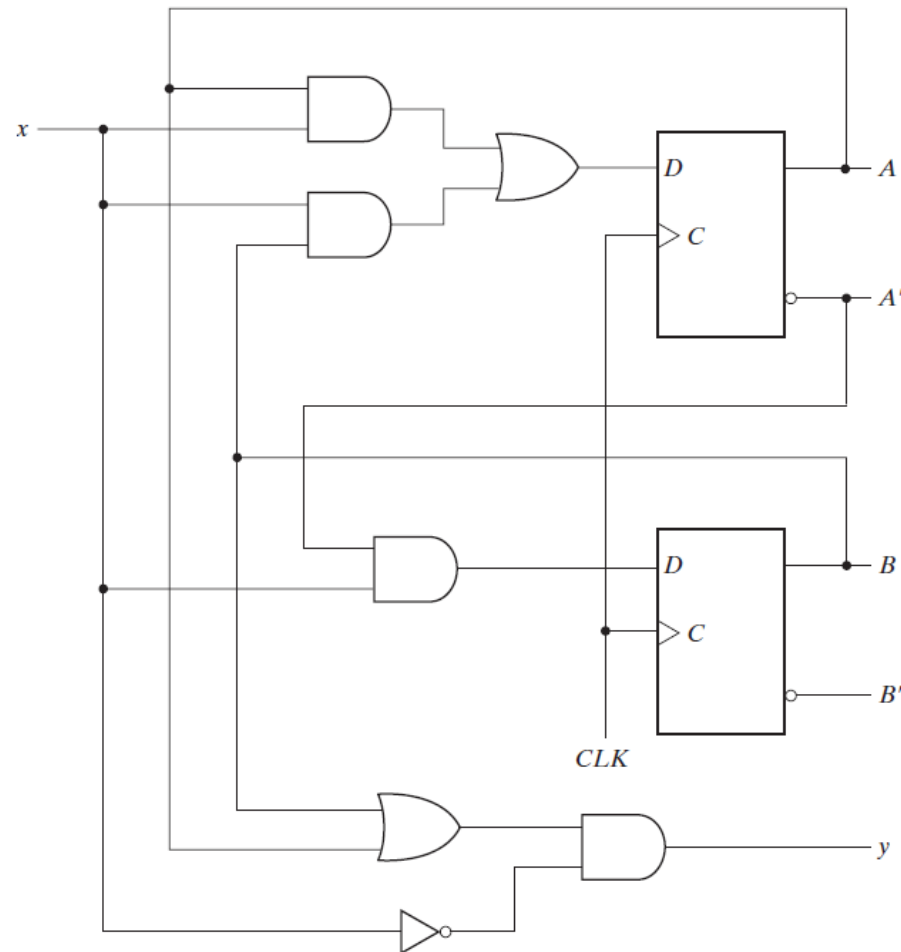
$$D_B = A'x$$

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

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

# Diseño o Síntesis de Circuitos Secuenciales

- Paso 7:



**FIGURA 5-15**  
Ejemplo de circuito secuencial

# Diseño o Síntesis de Circuitos Secuenciales

- Obtención Directa de ecuaciones de Estados y salidas:

**Tabla 5-2**

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

<b>Estado actual</b>		<b>Entrada</b>	<b>Siguiente estado</b>		<b>Salida</b>
<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

# Diseño o Síntesis de Circuitos Secuenciales

- Obtención Directa de ecuaciones de Estados y salidas:
- Simplemente sumamos los minitérminos
- $DA = A'BX + AB'X + ABX$
- $DB = A'BX + A'B'X$
- $Y = A'BX' + AB'X' + ABX'$



# Diseño o Síntesis de Circuitos Secuenciales

- Obtención Directa de ecuaciones de Estados y salidas:
- Si COMPARAMOS SE PUEDEN SACAR DIRECTAMENTE DEL DIAGRAMA DE ESTADOS.
- TENIENDO EN CUENTA LA CODIFICACION DE ESTADOS:

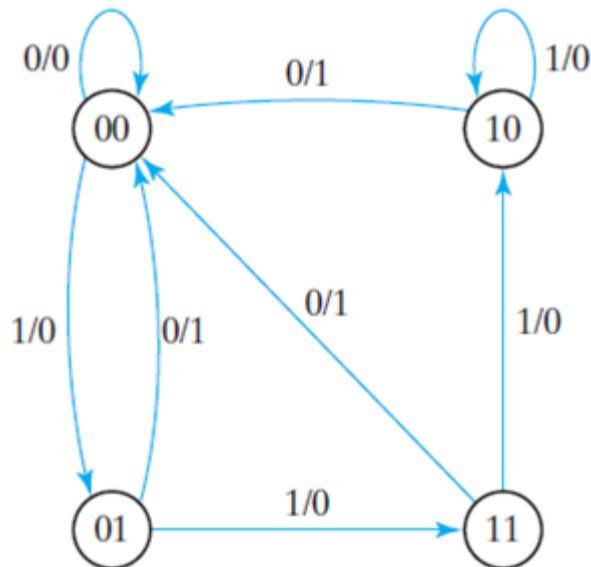
A	B	Estado	Código de Estado (AB)
0	0	Primero	00
0	1	Segundo	01
1	1	Tercero	11
1	0	Cuarto	10

# Diseño o Síntesis de Circuitos Secuenciales OBTENCIÓN DIRECTA DE ECUACIONES

$$DA = A'BX + AB'X + ABX$$

$$DB = A'BX + A'B'X$$

$$Y = A'BX' + AB'X' + ABX'$$



A	B	Estado	Código
0	0	Primero	00
0	1	Segundo	01
1	1	Tercero	11
1	0	Cuarto	10

# 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

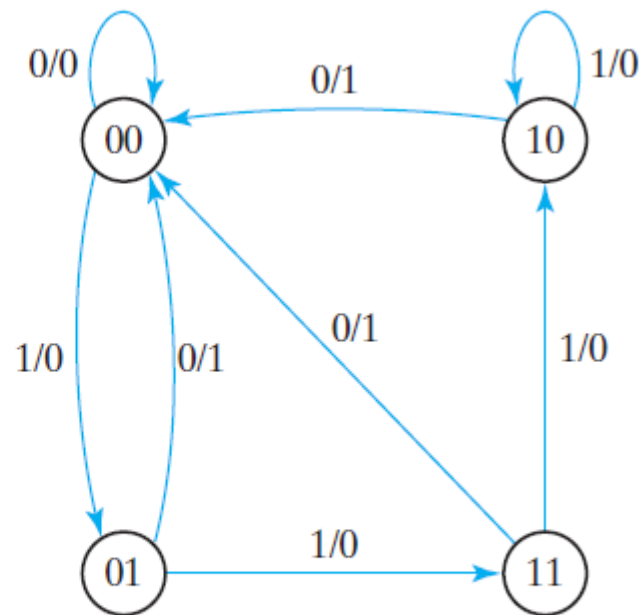
---

```
//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

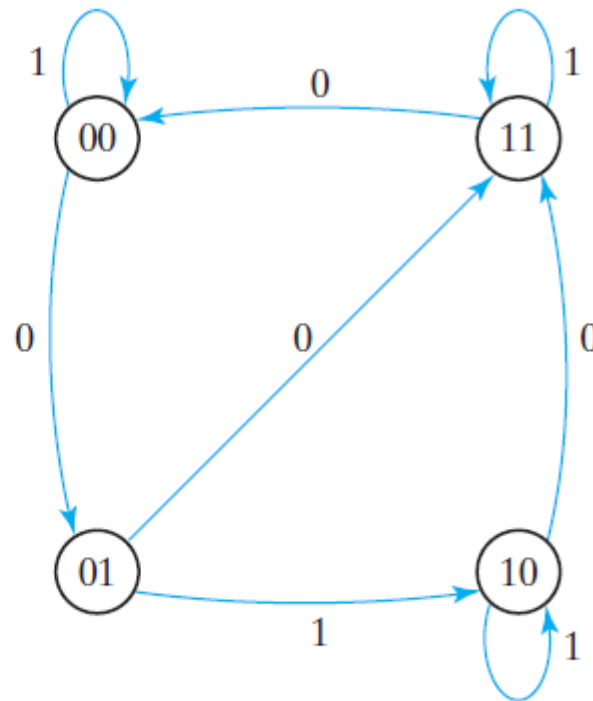
# HDL para Secuenciales

## 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

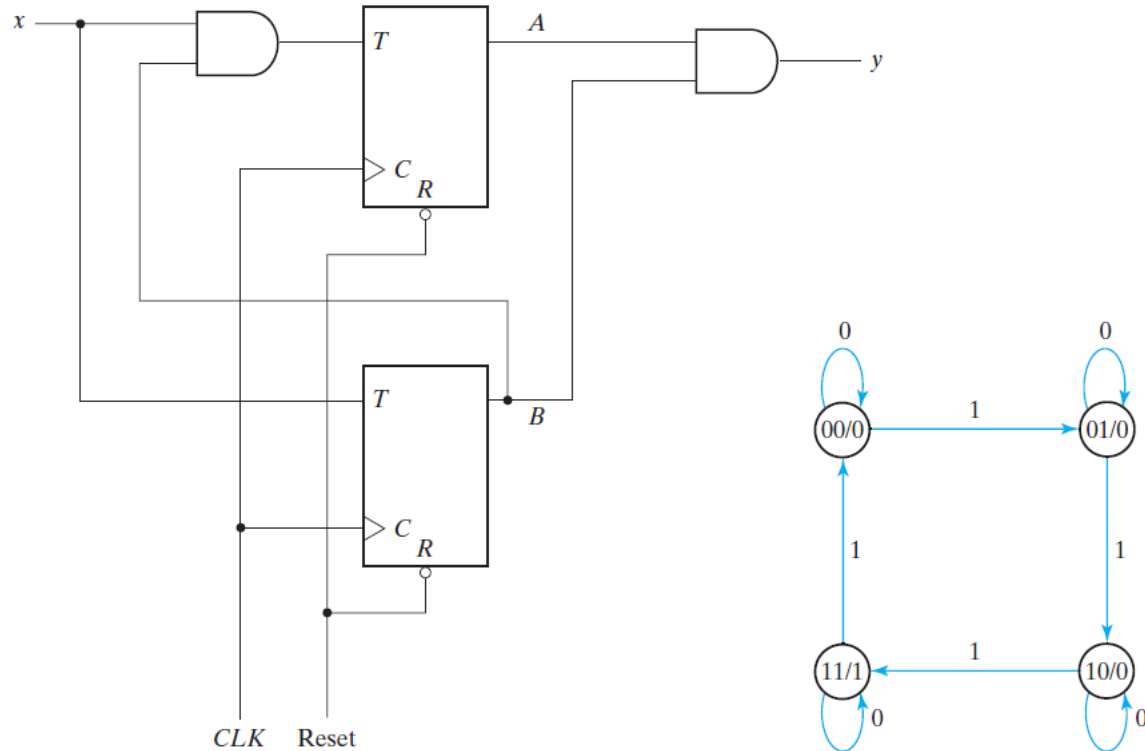
## Ejemplo HDL 5-6

---

```
//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



**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

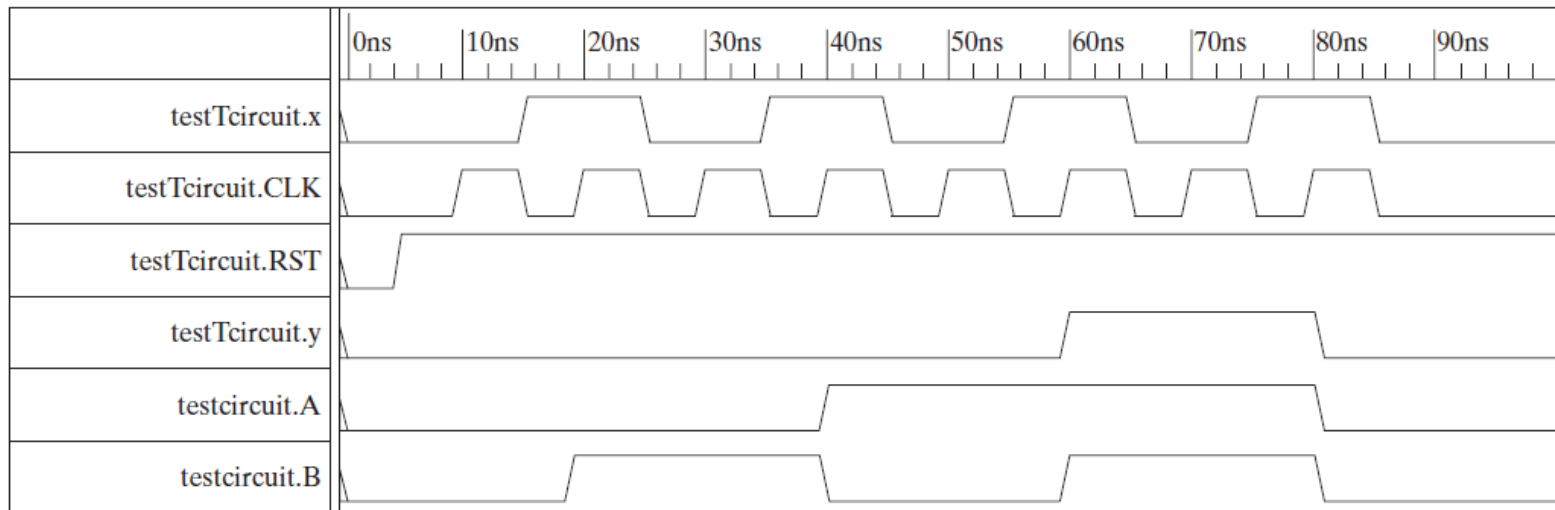
```
//Estímulo para probar el circuito secuencial
module testTcircuit;
    reg x,CLK,RST;    //entradas del circuito
    wire y,A,B;       //salida del circuito
    Tcircuit TC (x,y,A,B,CLK,RST);    // se crea un ejemplar
                                        // del circuito

    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

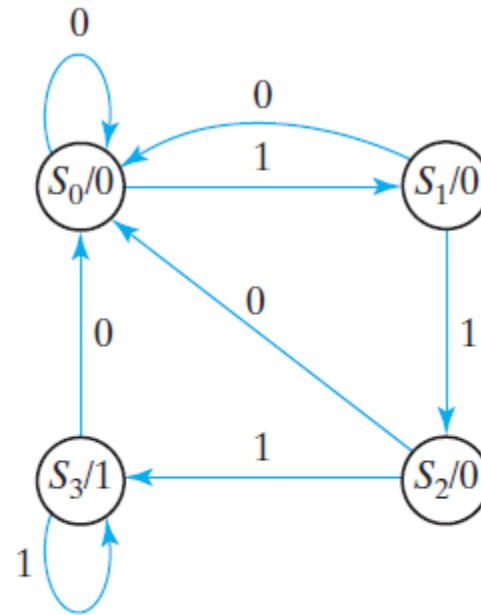


**FIGURA 5-21**

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

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

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

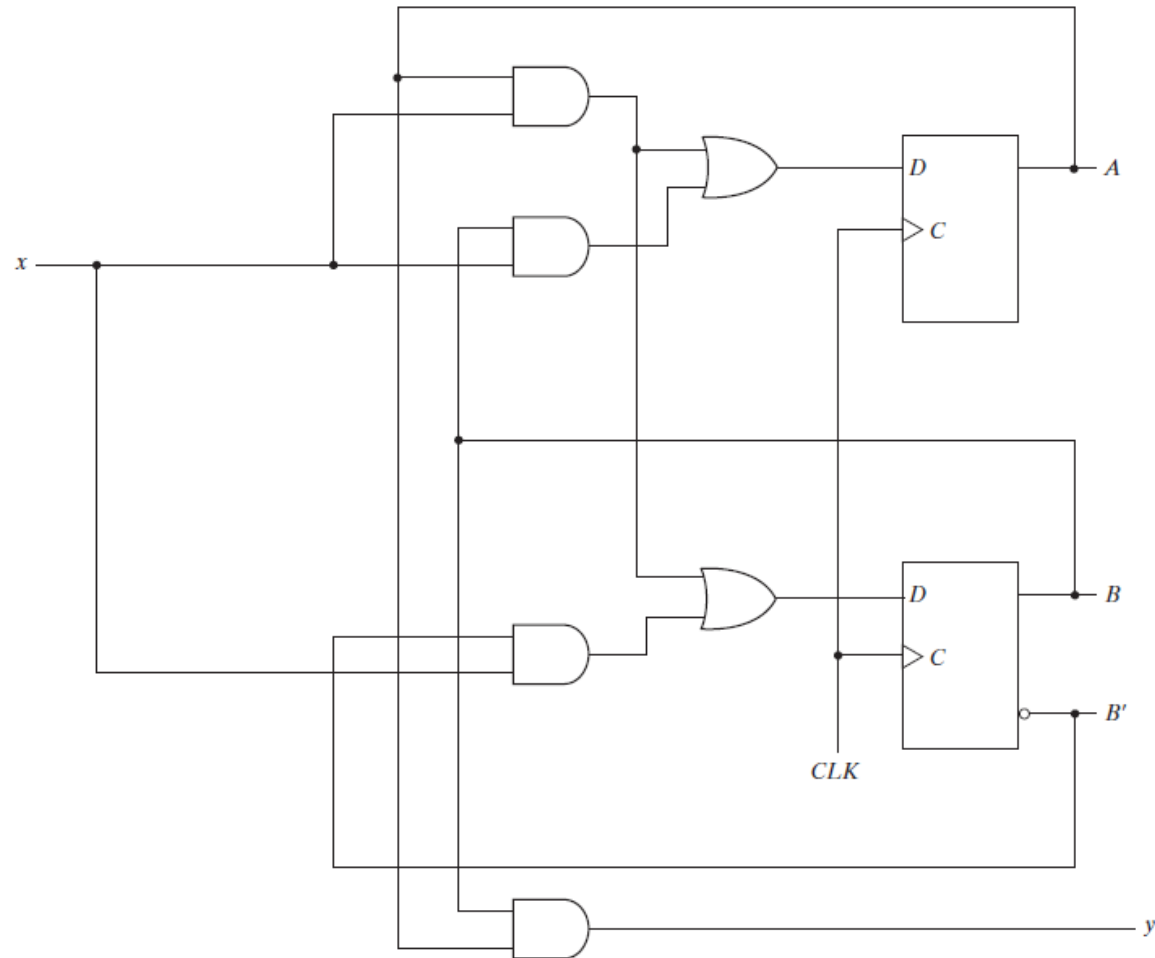
$$y(A, B, x) = \Sigma(6, 7)$$

$$D_A = Ax + Bx$$

$$D_B = Ax + B'x$$

$$y = AB$$

# 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*

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

a)  $JK$

b)  $T$

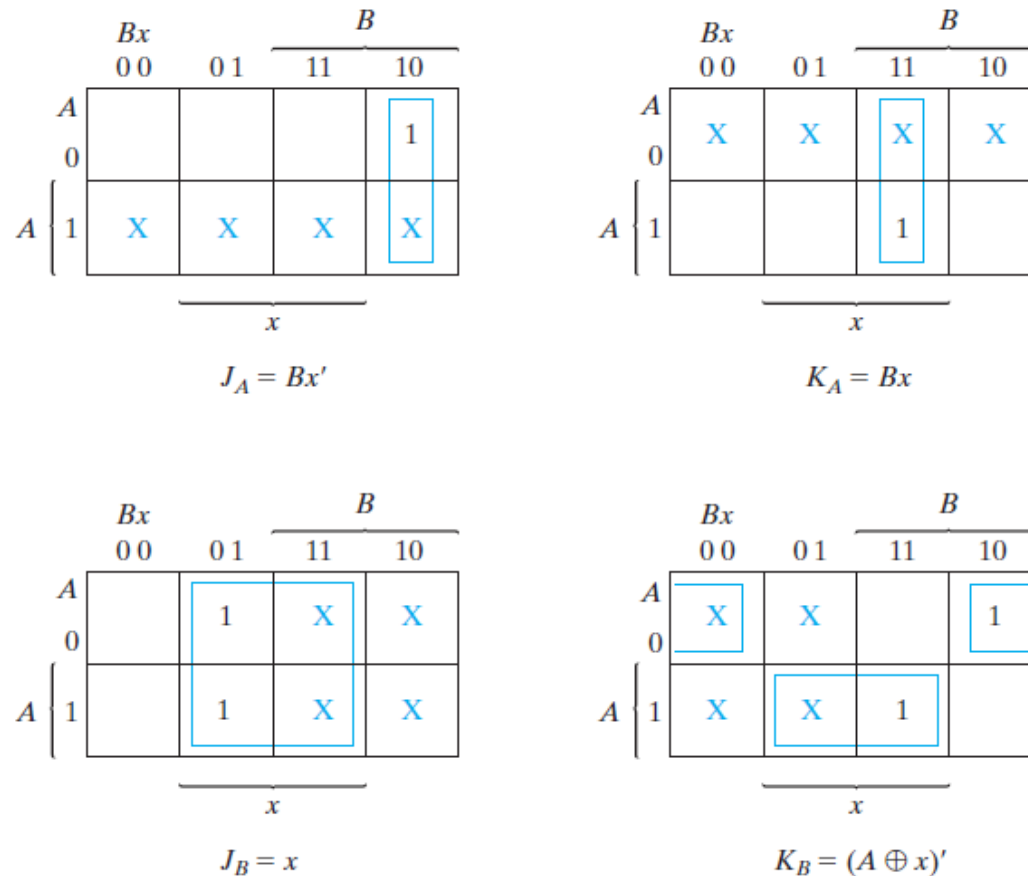
# Proceso de Diseño con FF JK

**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<sub>A</sub></i>	<i>K<sub>A</sub></i>	<i>J<sub>B</sub></i>	<i>K<sub>B</sub></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

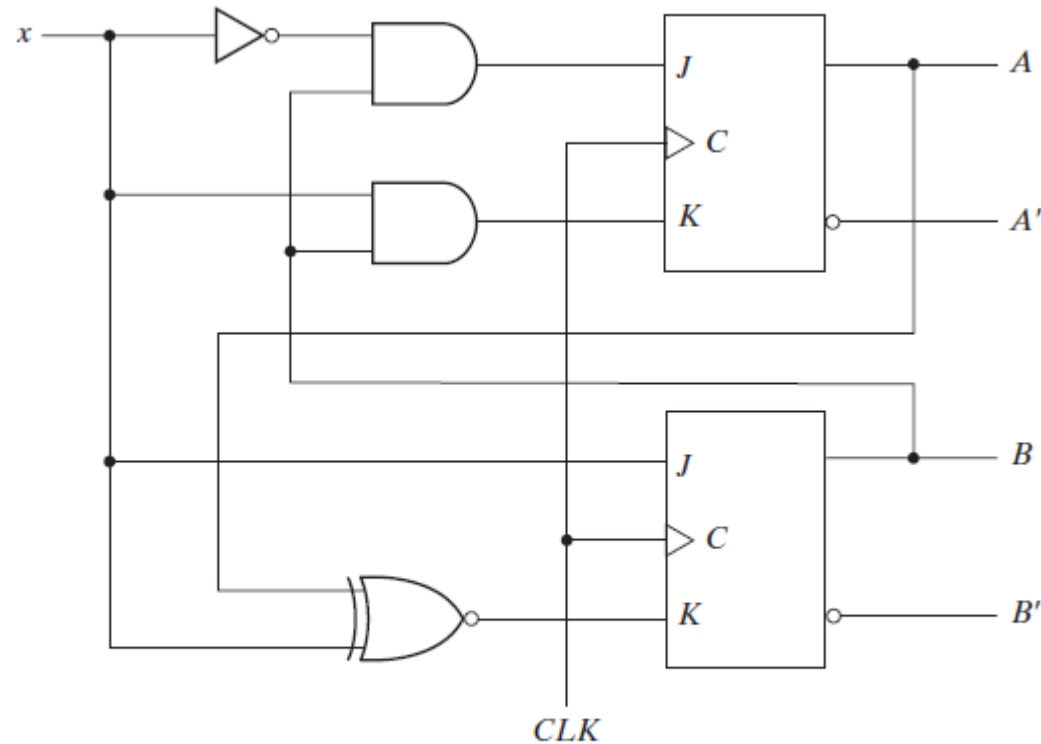
# Proceso de Diseño con FF JK



**FIGURA 5-27**

Mapas para las ecuaciones de entrada  $J$  y  $K$

# Proceso de Diseño con FF JK

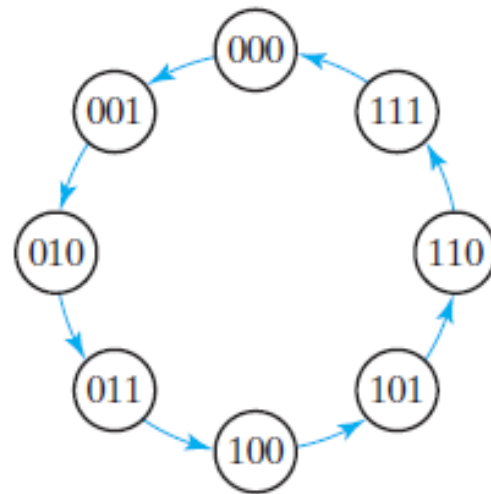


**FIGURA 5-28**

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

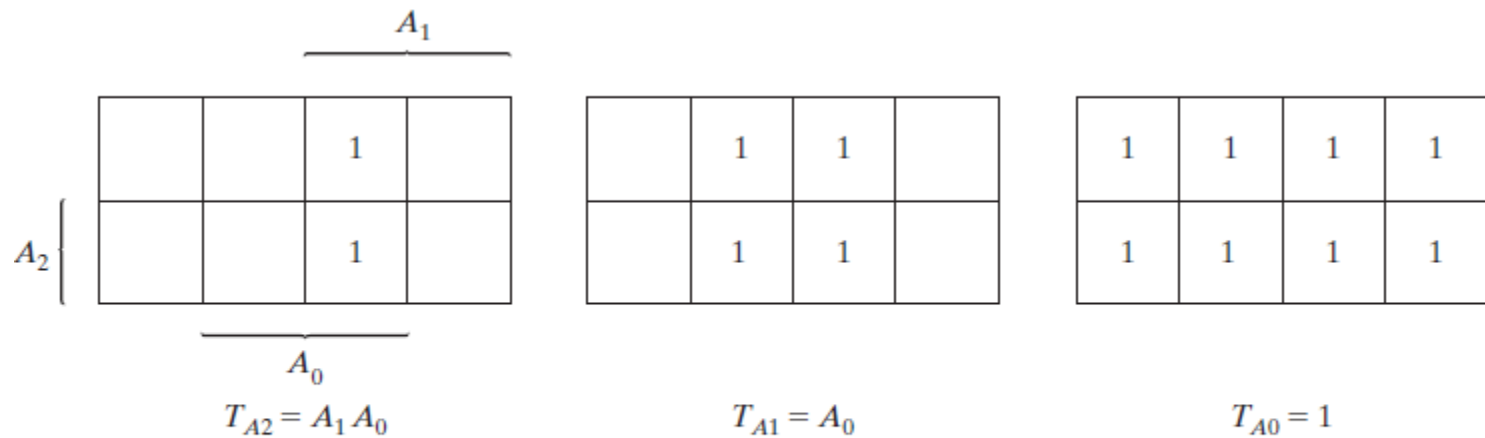
# Proceso de Diseño con FF T

**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

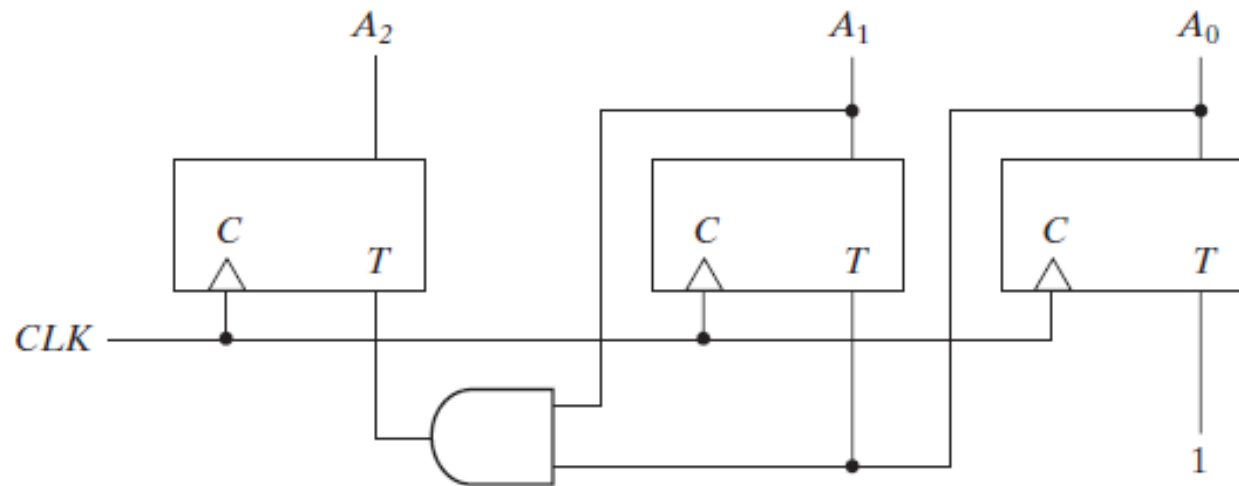
# Proceso de Diseño con FF T



**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