Instituto Politécnico Nacional

Centro de Investigación en Computación

Lenguajes de descripción de hardware

Tarea 2 - Dos circuitos importantes

Profesor:

M. en C. Osvaldo Espinosa Sosa

Por:

Ing. Ricardo Aldair Tirado Torres

Ciudad de México, 18 de abril de 2024

Tabla de contenido

1. Objetivos

- Aprender acerca de la importancia de dos circuitos lógicos importantes como lo son el desplazador de barril y el flip flop tipo T.
- Comprender el funcionamiento y la implementación de un desplazador de barril y un flip flop tipo T.

2. Desplazador de barril

Actividad 1

Un desplazador de barril ("*barrel shifter*") es un circuito combinatorio que permite desplazar un dato de entrada tantas posiciones como se indique en su control. Investigar su funcionamiento y describir un desplazador de barril de 4 bits. Simular el circuito.

Un desplazador de barril (o *barrel shifter*, en ingles) es un circuito digital que desplaza (o rota) los bits de una palabra de entrada un cierto número de posiciones especificado mediante un valor binario en unas lineas de selección. El desplazamiento se puede hacer en un sentido o en ambos por medio de otra variable de control. La forma más sencilla de lograr esto es mediante el uso de una serie de multiplexores donde una salida está conectada a la entrada del siguiente multiplexor en la cadena, de una manera específica que depende de la cantidad de desplazamiento especificado. [1]

Un desplazador de barril consta de una serie de multiplexores. Cada multiplexor selecciona uno de los bits de entrada basándose en una señal de control del registro de cantidad de desplazamiento. Luego, la salida de cada multiplexor se conecta a la entrada del siguiente multiplexor de la cadena, formando un bucle. La salida del último multiplexor de la cadena es el valor desplazado.[2] Por ejemplo, un *barrel shifter* de 4 bits y con entradas A, B, C y D, puede ciclar el orden de los bits ABCD como DABC, CDAB o BCDA, en donde ningún bit se pierde. Esto muestra que se pueden cambiar todas las salidas hasta 3 posiciones a la derecha. [3]

Hay dos tipos de desplazadores de barril: el aritmético y el lógico.

- **Desplazador aritmético**: Se utiliza para desplazar números binarios con o sin signo. Está diseñado para preservar el bit de signo del número al cambiar. Si el número que se está desplazando es un número con signo, el desplazador aritmético desplazará el bit de signo junto con los demás bits.
- **Desplazador lógico**: Se usa para desplazar sólo números binarios sin signo. No conserva el bit de signo del número al realizar el cambio.

Algunas aplicaciones del desplazador de barril son:

- **Procesamiento de señales digitales**: Se utilizan para realizar operaciones rápidas de multiplicación y división. Por ejemplo, en una implementación de filtro FIR, se puede usar un desplazador de barril para cambiar los coeficientes del filtro según el orden del filtro.
- **Criptografía**: Se emplean para realizar operaciones bit a bit, como cifrado y descifrado. Por ejemplo, se puede utilizar un desplazador de barril para realizar un cambio circular en un valor binario para mejorar la seguridad del algoritmo de cifrado.
- **Arquitecturas de microprocesadores**: Se usan para cambiar el contenido de los registros, lo que permite una manipulación eficiente de los datos. Por ejemplo, en la arquitectura ARM, el desplazador de barril se utiliza para realizar operaciones de cambio y rotación en el contenido de los registros. [2]

La visualización RTL del desplazador de barril en Verilog se muestra en la Figura 1. Como se observa, la implementación se realiza utilizando solamente instancias de multiplexores, cuatro de ellos para el desplazamiento según la selección y uno extra para el sentido del desplazamiento. Las simulaciones se visualizan en la Figura 2, en donde se muestra que el dispositivo funciona de manera correcta, cargando un valor y rotándolo tanto a la izquierda como a la derecha.

En los Anexos se localiza la descripción del desplazador de barril. Al tratarse de un circuito combinatorio, la lista sensible tiene de argumento a todas las señales de entrada. Ahora bien, utilizando una estructura *case*, se evalúo la cantidad de desplazamientos que se debía hacer y con el operador *?* se seleccionó el sentido.

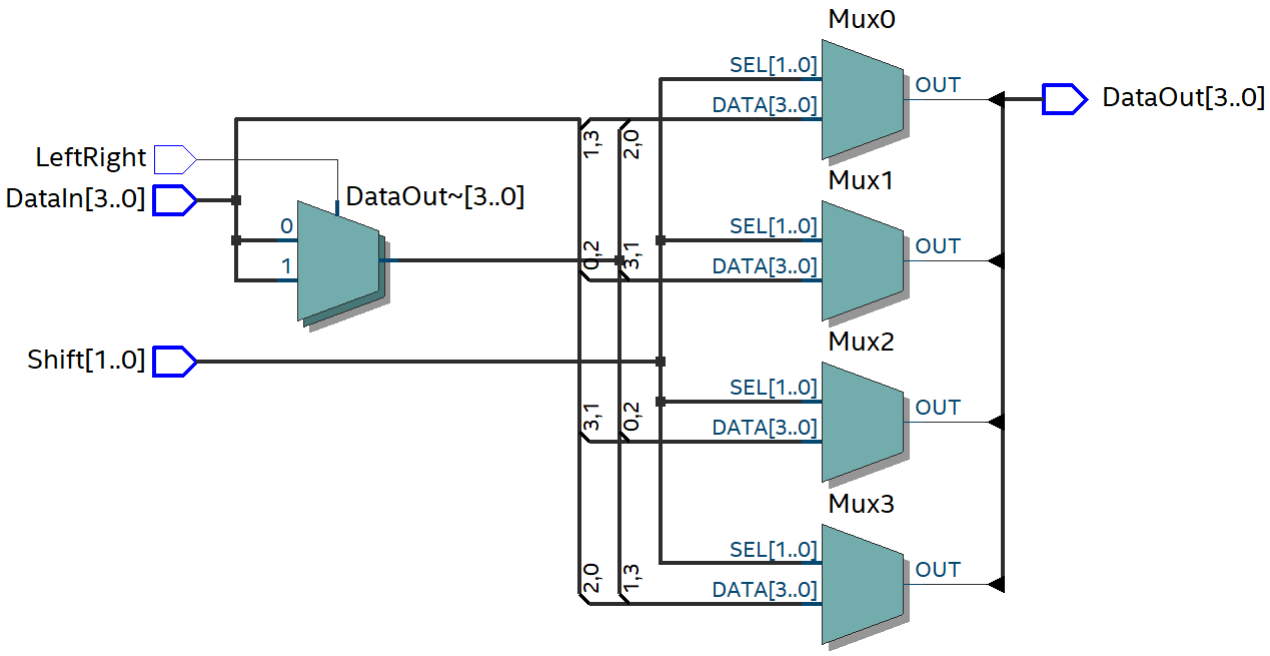

Figura 1: Diagrama RTL del desplazador de barril de 4 bits.

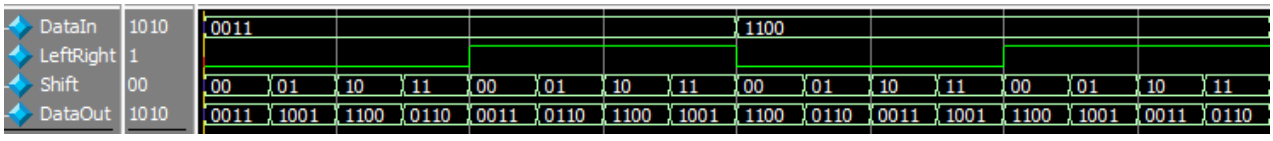

Figura 2: Simulación del desplazador de barril de 4 bits con el visor de formas de onda de ModelSim.

3. Flip Flop tipo T

Actividad 2

El flip-flop tipo T es un circuito que se usa como base en la construcción de contadores. Se le llama tipo T por "*Toggle*" es decir, cuando tiene aplicado un uno lógico a su entrada, el flip-flop invertirá el valor de su salida con el flanco de activación. Investigar su funcionamiento y describirlo. Simular el circuito.

El flip flop tipo T es un circuito lógico de entrada única que mantiene o alterna su salida según el estado de entrada. T es una abreviatura de *Toggle* o conmutar, en español. La operación que realiza este tipo de flip flop es cambiar la salida del siguiente estado por el complemento del estado actual. A diferencia del flip flip tipo SR, el tipo T evita la aparición de un estado intermedio.[4] Obviando el comportamiento del *reset* que se implemente, el funcionamiento se da de la siguiente manera: si la entrada T presenta un nivel bajo "0", el dispositivo está en su modo de memoria (no presenta cambios), y si en la entrada T se encuentra un nivel alto "1", el dispositivo cambia de estado, es decir la salida adquiere el complemento del valor actual.[5] Es de utilidad en algunas aplicaciones como:

- **División de frecuencia**: Se puede utilizar para dividir la frecuencia de una señal de reloj por dos, lo que lo hace útil en aplicaciones como relojes digitales y sintetizadores de frecuencia.
- **Multiplicación de frecuencia**: Se puede usar para multiplicar la frecuencia de una señal de reloj por dos, lo que lo hace útil en aplicaciones como sintetizadores de frecuencia y procesamiento de señales digitales.
- **Almacenamiento de datos**: Se puede emplear para almacenar un solo bit de datos, lo que lo hace útil en aplicaciones como registros de desplazamiento y dispositivos de memoria.
- **Contadores**: Se puede utilizar junto con otras puertas lógicas digitales para crear contadores binarios que pueden contar hacia arriba o hacia abajo según el diseño. Esto los hace útiles en aplicaciones en tiempo real como temporizadores y relojes. [6]

La visualización RTL del flip flop tipo T en Verilog se muestra en la Figura 3. Se observa que la implementación se hace instanciando un flip flop tipo D, pero colocando la señal T en la terminal ENA y usando un lazo de retroalimentación negativa entre la salida Q y la entrada D. Las simulaciones para el código en Verilog se visualizan en la Figura 4, en donde se muestra que el dispositivo funciona de manera correcta, ya que en un inicio no se tiene un valor en la salida, hasta que se activa el *reset*. Con el valor de la salida inicializado, se observa que al tener un valor alto en T, al siguiente flanco de subida del reloj se cambia el valor de la salida por su complemento.

En los Anexos se localiza la descripción del flip flop tipo T con *reset* asíncrono. Se utilizó una lista sensible para el flanco de subida de la señal de reloj y el *reset*. Dentro de esta estructura se empleó la sentencia *if* para comparar el valor de RST:

- Si es 1, se restablece el valor de la salida a 0.
- Si es 0, continúa leyendo la sintaxis de código.

Y una sentencia *else if* para comparar el valor de T:

- Si es 1, se cambia el valor de la salida Q por el complemento del valor actual.
- Si es 0, la salida no cambia y se mantiene su último valor.

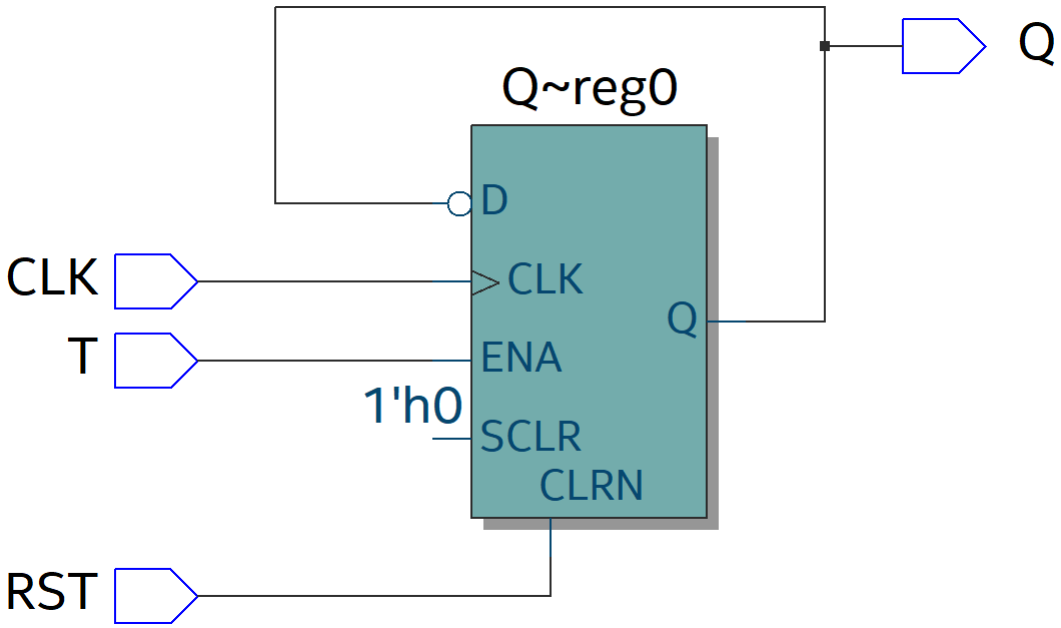

Figura 3: Diagrama RTL del flip flop tipo T con *reset* asíncrono.

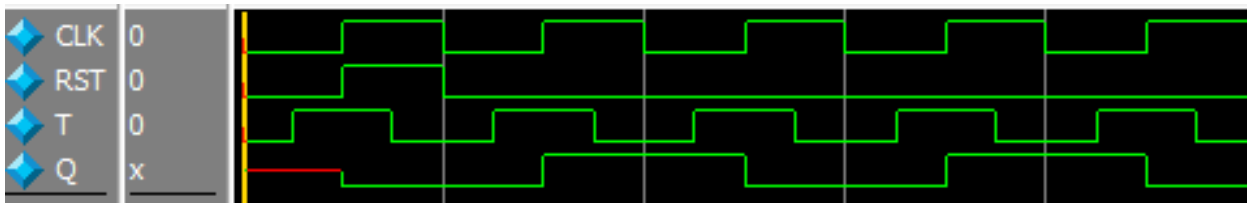

Figura 4: Simulación del flip flop tipo T con *reset* asíncrono con el visor de formas de onda de ModelSim.

4. Conclusiones

En conclusión, se implementaron los 2 circuitos en lenguaje Verilog de manera exitosa.

Para el desplazador de barril de 4 bits, se comprendió como es que se implementa este tipo de circuito combinatorio y su importancia en varias aplicaciones como lo son el procesamiento de señales, la criptografía y sobre todo en la arquitectura de microprocesadores.

Para el flip flop tipo T, se implementó de manera correcta y se diferencio su funcionamiento contra un tipo de flip flop implementado anteriormente (flip flop tipo D).

Se comprobó el funcionamiento de los circuitos utilizando las simulaciones de forma de onda en ModelSim y con el visor RTL se analizó como es que la herramienta de Quartus implementa los dispositivos.

En los Anexos se pueden encontrar los códigos implementados junto con sus respectivos bancos de pruebas.

Referencias

[1] M. Rouse, "Barrel shifters," https://www.techopedia.com/definition/17863/barrel-shifter, 2017.

[2] A. Mathur, "Barrel shifters," https://www.linkedin.com/pulse/barrel-shifters-aditya-mathur/, 2023.

[3] A. Limones, "Barrel shifter," https://www.scribd.com/document/329471576/Barrel-Shifter, 2015.

[4] oemsecrets, "Understanding the t flip-flop," https://www.oemsecrets.com/articles/understanding-the-t-flip-flop, 2021.

[5] M. Olmo, "El flip flop t," http://hyperphysics.phy-astr.gsu.edu/hbasees/Electronic/Tflipflop.html.

[6] Anónimo, "Introduction to t flip-flops and its working," https://www.electronicsforu.com/technology-trends/learn-electronics/t-flip-flop-circuit-truth-table-limitations-applications, 2023.

5. Anexos

5.1. Descripciones del hardware

```
module Barrel_Shifter(
 input     [3:0]  DataIn,
 input     [1:0]  Shift,
 input            LeftRight,
 output reg [3:0]  DataOut
);

always @(*) begin
 case(Shift)
  2'b00: DataOut = DataIn;
  2'b01: DataOut = LeftRight ? {DataIn[2:0], DataIn[3]} : {DataIn[0],
   DataIn[3:1]};
  2'b10: DataOut = LeftRight ? {DataIn[1:0], DataIn[3:2]} : {DataIn[1:0],
   DataIn[3:2]};
  2'b11: DataOut = LeftRight ? {DataIn[0], DataIn[3:1]} : {DataIn[2:0],
   DataIn[3]};
  default: DataOut = 4'b0000;
 endcase
end

endmodule
```

Programa 1: Descripción en Verilog del desplazador de barril de 4 bits.

```
module FF_T (
 input     CLK,
 input     RST,
 input     T,
 output reg Q
);

always @(posedge CLK or posedge RST)
begin
 if (RST)
  Q <= 1'b0;
 else if (T)
  Q <= ~Q;
end

```

```
endmodule
```

Programa 2: Descripción en Verilog del flip flop tipo T con *reset* asíncrono.

5.2. Bancos de pruebas (*Test Benches*)

```
‘timescale 1 ns/ 1 ps
module Barrel_Shifter_vlg_tst();
 reg   [3:0]   DataIn;
 reg   [1:0]   Shift;
 reg      LeftRight;
 wire   [3:0]   DataOut;

 Barrel_Shifter i1 (
  .DataIn(DataIn),
  .LeftRight(LeftRight),
  .Shift(Shift),
  .DataOut(DataOut)
 );

 initial
 begin
  DataIn = 4’b0011; LeftRight = 1’b0;
  #40; LeftRight = 1’b1;
  #40; DataIn = 4’b1100; LeftRight = 1’b0;
  #40; LeftRight = 1’b1;
  #40; DataIn = 4’b0101; LeftRight = 1’b0;
  #40; LeftRight = 1’b1;
  #40; DataIn = 4’b1010; LeftRight = 1’b0;
  #40; LeftRight = 1’b1;
  $display("Running testbench at CIC");
 end

 always
 begin
  Shift = 2’b00;
  #10; Shift = 2’b01;
  #10; Shift = 2’b10;
  #10; Shift = 2’b11; #10;
 end

endmodule
```

Programa 3: Banco de prueba para el Programa 1.

```
`timescale 1 ns/ 1 ps
module FF_T_vlg_tst();
 reg CLK;
 reg RST;
 reg T;
 wire Q;

 FF_T i1 (
  .CLK(CLK),
  .Q(Q),
  .RST(RST),
  .T(T)
 );

 initial
 begin
  CLK = 0; RST = 0; T = 0;
  #50; RST = 1; #50; RST = 0;
  $display("Running testbench at CIC");
 end

 always
 begin
  #25; T = ~T;
  #25; CLK = ~CLK;
 end

endmodule
```

Programa 4: Banco de prueba para el Programa 2.