



**Instituto Politécnico Nacional
Escuela Superior de Cómputo
“ESCOM”**



Unidad de Aprendizaje:
Diseño de sistemas digitales

**Entregable No. 9
JK**

Integrantes:
Álvarez Hernández Gabriel Alexander

García Quiroz Gustavo Iván

Huesca Laureano Josué Alejandro

Muñoz Valdivia Irving Omar

Pedroza Villagómez Emir

Nombre del profesor: Flores Escobar José Antonio

Índice de contenido

Introducción	1
Marco teórico	2
Flip-flop J-K	2
Biestable JK activo por flanco	3
Biestable JK Maestro-Esclavo	3
Desarrollo	5
Implementación del código Verilog	5
Módulo FFJK	5
Módulo FFJK_tb	6
Simulación	9
Análisis del circuito RTL	9
Conclusiones	11
Referencias	12

Introducción

Los flip-flops juegan un papel fundamental como elementos de almacenamiento y sincronización. Entre estos, el Flip-Flop JK destaca por su versatilidad y amplia aplicación en circuitos secuenciales. Esta práctica se centra en el diseño, implementación y verificación de un FFJK utilizando el entorno de desarrollo Quartus II de Intel FPGA, proporcionando una experiencia práctica en el flujo de diseño digital moderno.

El objetivo principal de esta práctica es familiarizar a los estudiantes con el proceso completo de diseño digital, desde la descripción del circuito en un lenguaje de descripción de hardware (HDL) hasta la simulación y análisis del circuito resultante. Se utilizará Verilog como lenguaje de descripción, aprovechando su capacidad para modelar circuitos digitales de manera eficiente y flexible.

A lo largo de la práctica, se abordarán varios aspectos clave del diseño digital. Comenzaremos con el desarrollo del código Verilog para el FFJK, seguido por la creación de un testbench exhaustivo para verificar su funcionamiento. Luego, se procederá a la simulación del diseño en Quartus II, permitiendo observar el comportamiento del circuito bajo diferentes condiciones de entrada. Finalmente, se realizará un análisis del circuito RTL generado, proporcionando insights sobre la implementación a nivel de compuertas del diseño.

Esta práctica no solo refuerza los conceptos teóricos relacionados con los flip-flops y la lógica secuencial, sino que también proporciona experiencia práctica en el uso de herramientas de diseño FPGA modernas. Los estudiantes aprenderán a navegar por el flujo de diseño digital, desde la concepción hasta la verificación, preparándolos para abordar proyectos más complejos en el futuro.

Al completar esta práctica, los estudiantes habrán ganado una comprensión profunda del funcionamiento del FFJK, mejorado sus habilidades en programación Verilog, y adquirido experiencia valiosa en el uso de Quartus II para el diseño y simulación de circuitos digitales. Estos conocimientos y habilidades son fundamentales para cualquier ingeniero involucrado en el diseño de sistemas digitales modernos.

Marco teórico

Flip-flop J-K

Es versátil y es uno de los tipos de flip-flop más usados. Su funcionamiento es idéntico al del flip-flop S-R en las condiciones SET, RESET y de permanencia de estado. La diferencia está en que el flip-flop J-K no tiene condiciones no válidas como ocurre en el S-R.

Este dispositivo de almacenamiento es temporal que se encuentra dos estados (alto y bajo), cuyas entradas principales, J y K, a las que debe el nombre, permiten al ser activadas:

J: El grabado (set en inglés), puesta a 1 o nivel alto de la salida.

K: El borrado (reset en inglés), puesta a 0 o nivel bajo de la salida.

Si no se activa ninguna de las entradas, el biestable permanece en el estado que poseía tras la última operación de borrado o grabado. A diferencia del biestable RS, en el caso de activarse ambas entradas a la vez, la salida adquirirá el estado contrario al que tenía.

La ecuación característica del biestable JK que describe su comportamiento es:

$$Q_{\text{siguiente}} = J\bar{Q} + \bar{K}Q$$

Y su tabla de verdad es:

J	K	Q	Q _{siguiente}
0	0	0	0
0	0	1	1
0	1	X	0
1	0	X	1
1	1	0	1
1	1	1	0

Tabla 1 Flip-Flop JK

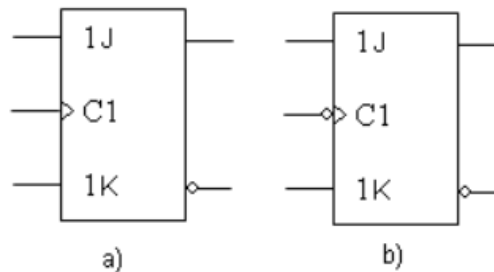
Una forma más compacta de la tabla de verdad es (Q representa el estado siguiente de la salida en el próximo flanco de reloj y q el estado actual):

J	K	Q
0	0	q
0	1	0
1	0	1
1	1	\bar{q}

Tabla 2 Módulo FFJK

Flip-Flop JK activo por flanco

Junto con las entradas J y K existe una entrada C de sincronismo o de reloj cuya misión es la de permitir el cambio de estado del biestable cuando se produce un flanco de subida o de bajada, según sea su diseño. Su denominación en inglés es J-K Flip-Flop Edge-Triggered. De acuerdo con la tabla de verdad, cuando las entradas J y K están a nivel lógico 1, a cada flanco activo en la entrada de reloj, la salida del biestable cambia de estado. A este modo de funcionamiento se le denomina modo de basculación (toggle en inglés).



Flip-Flop JK Maestro-Escavo

Aunque aún puede encontrarse en algunos equipos, este tipo de biestable, denominado en inglés J-K Flip-Flop Master-Slave, ha quedado obsoleto, ya que ha sido reemplazado por el tipo anterior.

Su funcionamiento es similar al JK activo por flanco: en el nivel alto (o bajo) se toman los valores de las entradas J y K y en el flanco de bajada (o de subida) se refleja en la salida.

Otra forma de expresar la tabla de verdad del biestable JK es mediante la denominada tabla de excitación:

J	K	Q	Q _{siguiente}
0	X	0	0
1	X	0	1
X	1	1	0
X	0	1	1
X=no importa			

Tabla 3 Flip-Flop JK maestro-esclavo.

Desarrollo

El código Verilog describe un módulo de Flip-Flop JK (FFJK) con entradas para reloj (clk_i), reinicio (rst_i), J (j_i) y K (k_i), y salidas Q (q_o) y Q negada (qn_o). El módulo utiliza una estructura always para detectar los flancos de subida del reloj y del reinicio. Dentro de esta estructura, se implementa la lógica del FFJK mediante una declaración case que evalúa las entradas J y K combinadas, determinando el próximo estado de la salida Q según las reglas de funcionamiento de un FFJK.

Implementación del código Verilog

Módulo FFJK

```
/*
Proyecto: FFJK
Archivo: FFJK.v
Descripcion: Descripcion de un FFJK
Asignatura: DSD
Prof: Flores Escobar Jose Antonio
Equipo: Coloque a los integrantes...
        Álvarez Hernández Gabriel Alexander
        Bueno Aguilar Alexis Haziell
        Garcia Quiroz Gustavo Ivan
        Huesca Laureano Josue Alejandro
        Muñoz Valdivia Irving Omar
*/
module FFJK(
    input      clk_i,
    input      rst_i,
    input      j_i,
    input      k_i,
    output reg  q_o,
    output      qn_o
);
    //wire [1:0] jk_w;
    //assign jk_w = {j,k};
    //case ()
    // J\K
    assign qn_o = ~q_o;
    always @(posedge clk_i, posedge rst_i)
    begin
        if(rst_i)
            q_o <= 1'b0;
        else
            case ({j_i,k_i})
                2'b00: q_o <= q_o;
                2'b10: q_o <= 1'b1;
                2'b01: q_o <= 1'b0;
                2'b11: q_o <= ~q_o;
            endcase
        end
    end
endmodule
```

Figura 2 Módulo FFJK

El módulo FFJK implementa un Flip-Flop JK, que es un elemento fundamental en el diseño de circuitos secuenciales digitales. Este tipo de flip-flop tiene dos entradas de control, J y K, que determinan su comportamiento. El FFJK puede mantener su estado actual, establecerse a 1, reiniciarse a 0, o invertir su estado, dependiendo de las combinaciones de estas entradas. Además, cuenta con una entrada de reloj para sincronización y una entrada de reinicio asíncrono. El módulo proporciona dos salidas: Q

y su complemento Q negada. Esta versatilidad hace que el FFJK sea útil en una amplia gama de aplicaciones, como contadores, registros de desplazamiento y máquinas de estados, permitiendo el almacenamiento y manipulación de datos en sistemas digitales.

Módulo FFJK_tb

El módulo FFJK_tb es un testbench diseñado para verificar el funcionamiento correcto del Flip-Flop JK (FFJK). Este testbench instancia el módulo FFJK y genera estímulos para probar todas sus funcionalidades. Utiliza un reloj con un período de 100 ps y aplica diferentes combinaciones de las entradas J y K para verificar los estados de no cambio, set, reset y toggle. También prueba el reset asíncrono. El testbench emplea un bloque always para generar el reloj y un bloque initial para aplicar los estímulos de prueba. Además, utiliza la directiva \$monitor para registrar y mostrar los cambios en las señales durante la simulación. Este enfoque permite una verificación completa del comportamiento del FFJK bajo diversas condiciones, asegurando que funcione correctamente antes de su implementación en hardware.


```

module FFJK_tb();

// Señales para conectar con el módulo FFJK
reg clk_i;
reg rst_i;
reg j_i;
reg k_i;
wire q_o;
wire qn_o;

// Instancia del módulo FFJK
FFJK uut (
    .clk_i(clk_i),
    .rst_i(rst_i),
    .j_i(j_i),
    .k_i(k_i),
    .q_o(q_o),
    .qn_o(qn_o)
);

// Generación del reloj (período de 100 ps)
always begin
    #50 clk_i = ~clk_i; // Medio período es 50 ps
end

// Proceso de estímulos
initial begin
    // Inicialización de señales
    clk_i = 0;
    rst_i = 1;
    j_i = 0;
    k_i = 0;

    // Liberación del reset
    #200 rst_i = 0;

    // Prueba de diferentes combinaciones
    #200 j_i = 0; k_i = 0; // No cambio
    #200 j_i = 1; k_i = 0; // Set
    #200 j_i = 0; k_i = 1; // Reset
    #200 j_i = 1; k_i = 1; // Toggle
    #200 j_i = 0; k_i = 0; // No cambio

    // Prueba de reset asíncrono
    #200 rst_i = 1;
    #100 rst_i = 0;

    // Más pruebas
    #200 j_i = 1; k_i = 0;
    #200 j_i = 1; k_i = 1;

```

```
// Más pruebas
#200 j_i = 1; k_i = 0;
#200 j_i = 1; k_i = 1;

// Fin de la simulación
#200 $finish;
end

endmodule
```

Figura 3 Módulo FFJK_tb

Simulación

Después, se inicia la herramienta de simulación RTL (generalmente ModelSim-Altera) desde Quartus II. En el simulador, se carga el testbench y se ejecuta la simulación. Se observan las formas de onda generadas, que muestran el comportamiento de las señales del FFJK a lo largo del tiempo. Estas ondas permiten verificar si el FFJK responde correctamente a los estímulos aplicados, como los cambios en las entradas J y K, los flancos de reloj y el reset asíncrono. Analizando estas formas de onda, se puede confirmar si el FFJK funciona según lo esperado en todas las condiciones de prueba especificadas en el testbench.

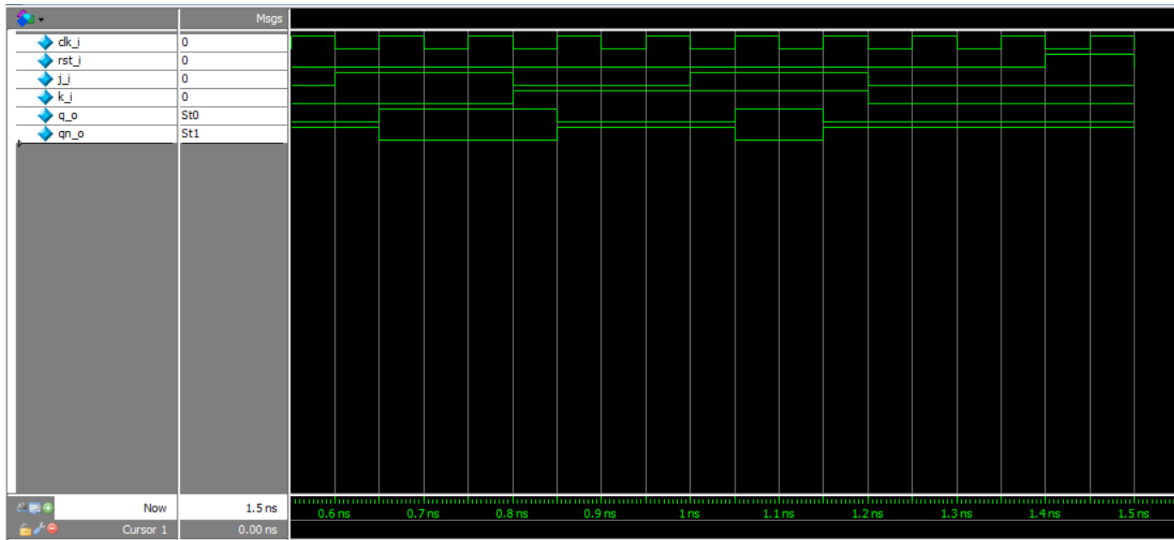


Figura 4 Simulación

Análisis del circuito RTL

El Análisis del circuito RTL (Register Transfer Level) para el módulo FFJK implica examinar la representación a nivel de compuertas y registros que Quartus II genera a partir del código Verilog. Este análisis se realiza mediante el visor RTL de Quartus II, que muestra una representación gráfica del circuito sintetizado. Para el FFJK, se observarían los elementos clave como el flip-flop principal, la lógica combinacional que implementa las funciones J-K, y las conexiones de las señales de entrada y salida. El diagrama RTL mostraría cómo las entradas J y K se combinan para controlar el estado del flip-flop, cómo se implementa la funcionalidad de toggle, y cómo se conecta el reset asíncrono. Este análisis permite verificar que la síntesis del diseño corresponde correctamente a la descripción Verilog y ayuda a identificar posibles optimizaciones o problemas en la implementación del circuito antes de proceder a etapas posteriores del diseño, como el place and route.



Conclusiones

La conclusión de esta práctica de diseño e implementación del Flip-Flop JK (FFJK) en Quartus II demuestra la eficacia del flujo de diseño digital moderno. A través del desarrollo del código Verilog, la creación de un testbench exhaustivo, la simulación en Quartus II y el análisis del circuito RTL, se logró verificar y validar el funcionamiento correcto del FFJK. Este proceso permitió comprobar todas las funcionalidades del flip-flop, incluyendo sus estados de set, reset, toggle y no cambio, así como su respuesta al reset asíncrono. La práctica resalta la importancia de la simulación y el análisis RTL en el diseño de circuitos digitales, ya que estas etapas permiten detectar y corregir problemas antes de la implementación física. Además, proporciona una valiosa experiencia en el uso de herramientas de diseño FPGA y en la aplicación de conceptos fundamentales de diseño digital, preparando así a los estudiantes para enfrentar proyectos más complejos en el futuro.

Referencias

[1] “Diseño de circuitos digitales y tecnología de computadores/Memorias”, Wikibooks.org. [En línea]. Disponible en: https://es.wikibooks.org/wiki/Dise%C3%B1o_de_circuitos_digitales_y_tecnolog%C3%ADa_de_computadores/Memorias. [Consultado: 27-jun-2024].

[2] Wikipedia contributors, “JKFF”, Wikipedia, The Free Encyclopedia, 10-jun-2008. [En línea]. Disponible en: <https://en.wikipedia.org/w/index.php?title=JKFF&oldid=218455752>.