

Tecnología de Computadores

Práctica 6



**Universidad
Rey Juan Carlos**

1. Objetivo

En esta sexta sesión de prácticas se pretende que el alumno se familiarice con los circuitos secuenciales. Para ello se analizará el funcionamiento de la señal de reloj en una FPGA y se trabajará implementando circuitos secuenciales sencillos basados en biestables.

Una vez terminada la práctica el alumno será capaz de:

- Entender el concepto de señal de reloj en los circuitos secuenciales
- Diferenciar el comportamiento de un latch frente al de un flip-flop
- Diseñar circuitos secuenciales sencillos en VHDL
- Implementar diseños secuenciales sencillos en una FPGA

2. Desarrollo de la práctica

La señal de reloj.

El reloj es el elemento fundamental en los circuitos secuenciales que permite la sincronización de señales y módulos. La placa utilizada en las prácticas tiene una señal de reloj que se genera en un pin concreto a una frecuencia concreta por lo que todos los circuitos secuenciales síncronos pueden conectarse a dicha señal para funcionar a la misma frecuencia. Para interiorizar el concepto de “señal de reloj” se propone el siguiente ejercicio de creación de un LED intermitente.

Construiremos un circuito con una señal de entrada de un bit y una salida también de un bit. La salida de un bit estará conectada directamente a la entrada. Lo que vamos a hacer a continuación es conectar la entrada del circuito al pin de la placa que genera la señal de reloj y la salida del circuito a un LED de la misma. De esta forma, como la señal de reloj es una señal cuadrada periódica, haremos que el LED parpadee de forma intermitente.

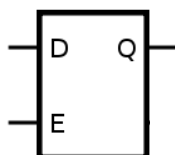
Escriba el código VHDL de una entidad que conecta una entrada denominada CLK con una salida denominada LED. Llame “*reloj*” a dicha entidad. Añada el fichero de constraints “*reloj_constraints.xdc*” proporcionado por el profesor. Añada también el testbench “*tb_reloj.vhd*” y realice la simulación del código sencillo. Observe que el comportamiento de la salida LED es el esperado y determine la frecuencia de la señal de reloj.

Genere ahora el bitstream del diseño y cárguelo en la FPGA. Observe el funcionamiento del LED. ¿Es normal este comportamiento? ¿A qué cree que es debido?

Cree ahora un nuevo proyecto en Vivado, añada el código VHDL “*reloj_profe.vhd*” proporcionado por el profesor junto con el fichero “*reloj.xdc*”. Genere el bitstream del nuevo diseño y cárguelo en la FPGA. Observe el funcionamiento del LED y analice el código VHDL que se le ha proporcionado. ¿Es normal este nuevo comportamiento? ¿A qué cree que es debido?

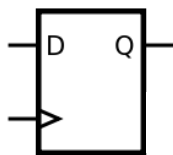
Biestables. Latches vs. Flip-flops.

En esta segunda parte de la práctica se trabajará con biestables. Como se ha visto en clase de teoría, los biestables se dividen en dos categorías: latches y flip-flops. Los latches son elementos de memoria asíncronos (no tienen señal de reloj), mientras que los flip-flops son elementos síncronos (sí tienen señal de reloj). Escriba el código VHDL de un latch D con enable como el mostrado en la siguiente imagen. Llame a la entidad “*latchD*”.



Simule el latch utilizando el testbench “*tb_latchD.vhd*” proporcionado y entienda su funcionamiento. Añada el fichero de constraints “*latchD_constraints.xdc*” e implemente el circuito en la FPGA. Analice su funcionamiento manipulando las entradas del circuito.

Cree ahora un nuevo proyecto en Vivado y escriba el código VHDL de un flip-flop D como el mostrado en la siguiente imagen. Llame CLK a la entrada de la señal de reloj.



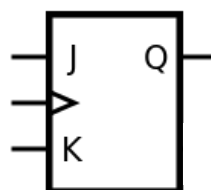
Simule el flip-flop utilizando el testbench “*tb_flipflopD.vhd*” proporcionado y entienda su funcionamiento. Añada el fichero de constraints “*flipflopD_constraints.xdc*” e implemente el circuito en la FPGA. Analice su funcionamiento manipulando la entrada del circuito. ¿Existe alguna diferencia de funcionamiento entre el latch y el flip-flop? ¿A qué cree que es debido?

Cree un tercer proyecto en Vivado, añada el código VHDL “*flipflop_profe.vhd*” proporcionado por el profesor y el fichero de constraints “*flipflopD_constraints.xdc*”. Implemente este diseño en la FPGA. Analice el código VHDL proporcionado y su funcionamiento en la FPGA. ¿Existe alguna diferencia entre este flip-flop, el anterior y el latch? ¿A qué es debido?

3. Ejercicio propuesto

Finalmente, a modo de ejercicio para profundizar en la escritura de código VHDL secuencial, se propone implementar en la FPGA un flip-flop J-K como el de la siguiente imagen.

J	K	Q
0	0	Q_{prev}
0	1	0
1	0	1
1	1	\bar{Q}_{prev}



Edite el fichero VHDL “*flipflopJK.vhd*” para añadir el comportamiento del flip-flop J-K resumido en la tabla de verdad anterior. Añada al proyecto el fichero de constraints “*flipflopJK_constraints.xdc*” e implemente el circuito en la FPGA. ¿Qué ocurre al mantener los interruptores asociados a J y K en la posición cero? ¿Qué ocurre cuando ambos interruptores están a uno?