

INFORME DE LABORATORIO 1

Autores: *María Del Mar Arbeláez Sandoval, Julián
Mauricio Sánchez Ceballos*

*Laboratorio de Electrónica Digital 2
Departamento de Ingeniería Electrónica y de Telecomunicaciones
Universidad de Antioquia*

Resumen

Esta práctica contiene la implementación y simulación de un semáforo doble y de peatones, esta práctica funciona como un acercamiento a la implementación de soluciones para problemas en FPGA utilizando System Verilog como lenguaje de descripción de hardware, el modelamiento de este problema se realiza mediante un esquema de máquinas de estados, el cual mediante un contador permite transitar entre los diferentes estados que puede tener este semáforo en particular y elementos combinacionales y secuenciales para las distintas operaciones necesarias en la práctica. En la implementación de la FPGA se pudo observar como el contador del estado actual mostraba una forma práctica de observar el funcionamiento del semáforo.

Palabras clave: System Verilog, FPGA, síntesis, Máquina de estados, Flip-Flops.

Introducción

SystemVerilog, es un lenguaje de descripción de hardware y lenguaje de verificación de hardware utilizado para modelar, diseñar, simular, probar e implementar sistemas electrónicos. Como introducción a este lenguaje de descripción de hardware usado en la materia Electrónica Digital 2, se tiene la implementación de la aplicación de un semáforo que tiene que controlar el flujo de autos en dos calles, y además, debe controlar el paso de peatones. el diagrama del problema. La práctica consiste en diseñar un controlador de semáforos como se puede observar en la siguiente figura:

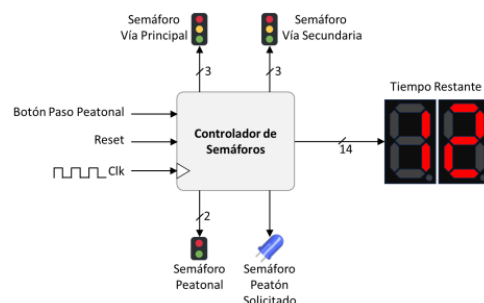


Figura 0-1: Diagrama de bloques para el controlador de semáforos.

El núcleo del planteamiento del problema es usar una máquina de estados finitos (FSM) que implementará el control de las luces del semáforo.

Se tiene estos requerimientos para el funcionamiento del cruce:

- Se empieza con un reset (Accionado con un botón) que lleva un estado de reset (Srst) por $tRESET$ segundos, dejando todos los semáforos en rojo.
- De aquí se tienen cuatro estados (Smg - Semáforo principal en verde, Smy - Semáforo principal en amarillo, Ssg - Semáforo secundario en verde, Ssy - semáforo secundario en amarillo) donde el semáforo que se esté accionando sea el único prendido y que tenga un tiempo de duración de $tVERDE_1$, $tAMAR_1$, $tVERDE_2$ y $tAMAR_2$, respectivamente.
- Si no se presiona en ningún momento el botón de peatones, se vuelve a empezar el ciclo en Smg, de lo contrario, se va, tras terminar el estado Ssy, a un tiempo $tVERDE_3$ donde se prende únicamente el semáforo de peatones (Spg). Después se pasa a un estado Sar donde están todos los semáforos en rojo por $tROJO$.

- No se recibe la señal del botón del semáforo en dos situaciones: mientras se esté pasando por el tiempo de reset ni cuando esté en verde el semáforo de peatones.

reset) en cualquier estado de la maquina de estados se presione el botón solicitando la luz verde en el semáforo peatonal. El diagrama de tiempos de ambos casos se muestran a continuación:

Diseño

Para el diseño del circuito, es importante hacer un diagrama de estados, teniendo en cuenta las variables para el cambio de estado, los estados en sí y su orden.

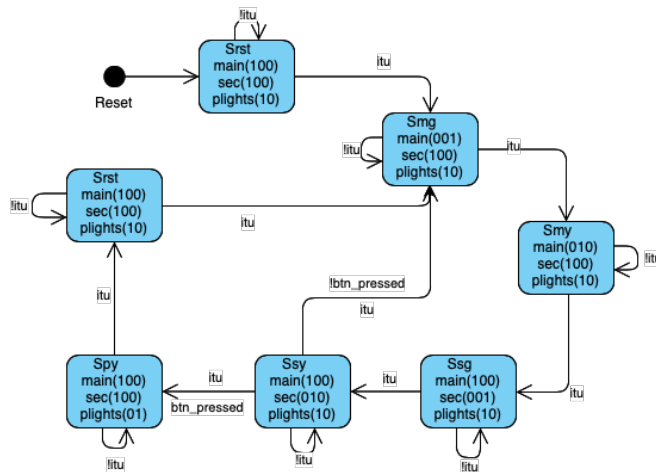


Figura 0-2: Diagrama de estados para el controlador de semáforos.

En la figura 0-2 se muestra por bloque el nombre del estado, la salida de la variable de salida del semáforo principal: main(Rojo, Amarillo, Verde), la del secundario sec(Rojo, Amarillo, Verde) y del peatonal plights(Rojo, Verde). De aquí, se saca que se tienen 7 estados, en donde solo se hace el cambio si se acaba el tiempo del estado, teniendo en cuenta que en el estado Ssy se toma en cuenta también si el botón peatonal fue presionado para pasar al estado Spy, donde se pone en verde el semáforo peatonal.

Simulación

Para la simulación se tienen dos casos, el primero es cuando durante un ciclo de la maquina de estados nadie oprime el botón solicitando la luz verde en el semáforo peatonal. El segundo caso es (excluyendo el estado de

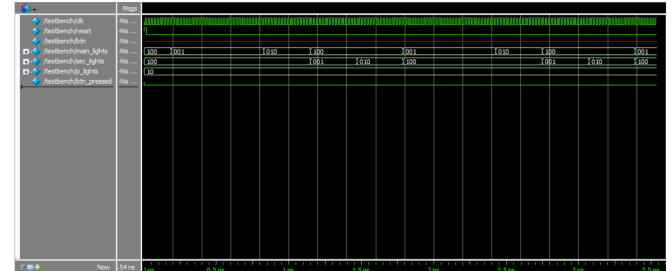


Figura 0-3: Diagrama de tiempo para el caso sin botón.

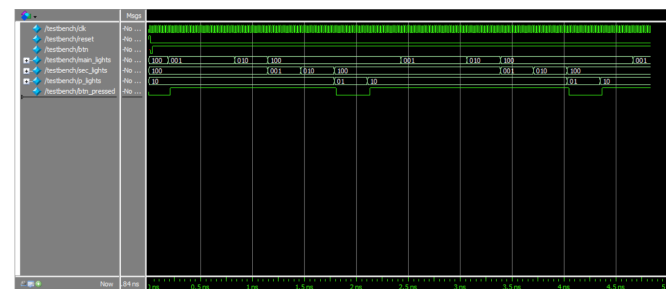


Figura 0-4: Diagrama de tiempo para el caso del botón.

Conclusiones

- Haber utilizado la implementación mediante máquinas de estados permitió el acercamiento a las singularidades y características del lenguaje System Verilog, aplicando también algunos conceptos aprendidos previamente en Electrónica Digital 1, el lenguaje System Verilog resultó, para esta práctica, una herramienta eficiente y versátil con un acercamiento más amable al estudiante por su similitud al lenguaje C, incluyendo algunos métodos y comportamiento general del lenguaje, sin dejar atrás la principal característica del lenguaje que es su capacidad de realizar descripción de hardware e implementación en paralelo.