

# Laboratorio de Electrónica Digital II

## Práctica No. 1 - Warm Up

### Controlador de Semáforos

#### Profesores

María Katherine Plazas O. (maria.plazas@udea.edu.co)

Luis Germán García M. (german.garcia@udea.edu.co)

Febrero 27, 2024



<b>Fecha de entrega:</b>	del 5 al 7 de marzo de 2024
<b>Medio de entrega:</b>	<a href="https://virtualingenieriaudea.co/">https://virtualingenieriaudea.co/</a>
<b>Sustentación:</b>	Horario de Laboratorio
<b>Valor Práctica:</b>	5% del curso

## 1 Introducción

En esta práctica de laboratorio, el grupo de estudiantes diseñará un sistema electrónico digital basado en FPGA, para el control de los semáforos de un cruce vial. El núcleo del sistema será una máquina de estados finitos (FSM) que implementará el control de las luces de los semáforos. La descripción de la FSM junto con la lógica requerida será realizada empleando el lenguaje de descripción de hardware SystemVerilog. Se deberá emplear la herramienta de simulación ModelSIM para llevar a cabo una serie de pruebas que permitan verificar el correcto funcionamiento del

secuenciador. Finalmente, el grupo de trabajo implementará el diseño en el sistema de desarrollo DE10-Lite, donde se deberá verificar el correcto funcionamiento de todo el sistema.

## 2 Objetivo de la Práctica

Diseñar e implementar una máquina de estados finitos junto con la lógica adicional requerida, para el control de las luces de los semáforos de un cruce vial, empleando el lenguaje de descripción de hardware SystemVerilog y herramientas de desarrollo para FPGAs.

## 3 Procedimiento

Para el correcto diseño e implementación del controlador de semáforos, se sugiere realizar el siguiente procedimiento:

- a. Leer esta guía varias veces para comprender el sistema que se va a diseñar, identificando así los requerimientos que deberán cumplirse el día de la sustentación del trabajo
- b. Elaborar un diagrama de bloques de alto nivel (*diagrama de bloques principal*) para representar el problema principal junto con las entradas y las salidas del sistema.
- c. Elaborar un diagrama de estados para la FSM a implementar, teniendo en cuenta los valores de las entradas que generan las transiciones y los valores de las salidas que se establecen en cada estado.
- d. Describir el diseño de la FSM y la lógica adicional usando SystemVerilog y emplear la herramienta de simulación ModelSIM para verificar el correcto funcionamiento del mismo. Llevar a cabo las correcciones pertinentes, si fuese el caso.
- e. Implementar el diseño en la FPGA disponible en el sistema de desarrollo DE10-Lite y verificar el correcto funcionamiento del mismo. Llevar a cabo las correcciones pertinentes, si fuese el caso.
- f. Escribir un corto reporte sobre el diseño e implementación de la práctica y subirlo a la plataforma de Ingeni@ junto con los archivos principales de su diseño en Quartus Prime.
- g. Sustentar el desarrollo en el horario de laboratorio correspondiente.

## 4 Funcionamiento del Controlador de Semáforos

A continuación se indica el funcionamiento que deberá tener el controlador de semáforos, representado en la Fig. 1, el cual se implementará en un cruce vial como el que se muestra en la Fig. 2:

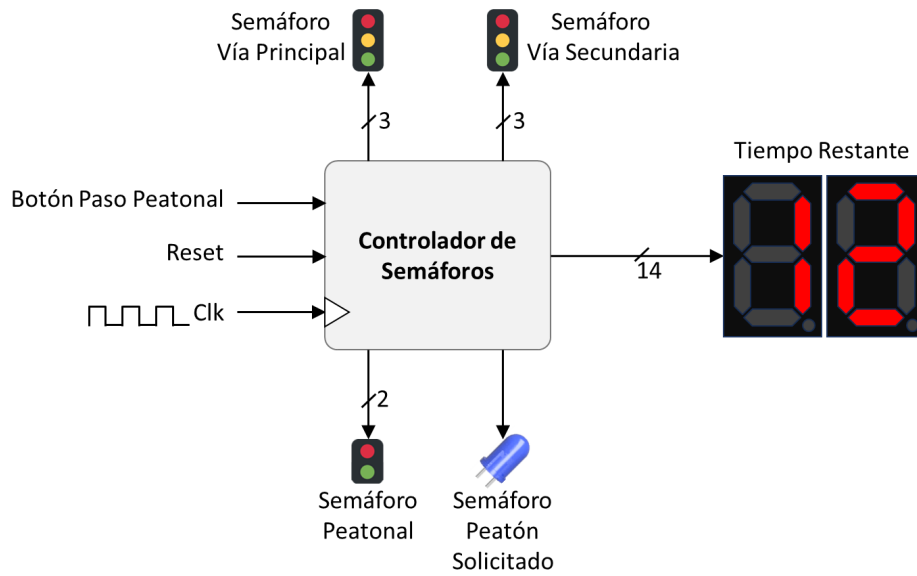


Fig. 1: Diagrama de bloques para el controlador de semáforos

El controlador de semáforos diseñado está destinado para un cruce vial que comprende una vía principal, una vía secundaria y cuatro senderos peatonales. Tanto la vía principal como la secundaria son de un solo sentido, por lo que se requerirá únicamente un semáforo para cada una. Cada semáforo para vehículos constará de tres luces: roja, amarilla y verde. En contraste, para los senderos peatonales se instalará un semáforo con dos luces: roja y verde. Los semáforos de los senderos peatonales operarán de manera idéntica en todo momento, por lo que solo será necesario controlar uno de los ocho semáforos disponibles.

Además, se integrará un display doble para visualizar en todo momento el tiempo restante para completar el estado actual de la secuencia del controlador de semáforos. Asimismo, se colocarán cuatro botones en las esquinas para que los peatones puedan solicitar que los semáforos peatonales se pongan en verde en algún momento de la secuencia. Es importante destacar que, aunque se dispongan de cuatro botones, solo se necesitará una única entrada en el controlador de semáforos, ya que los cuatro botones cumplen la misma función. Los peatones podrán verificar la activación de dicha solicitud mediante una luz externa denominada 'Semáforo Peatón Solicitado'.

La secuencia regular a ser realizada por el controlador de semáforos es la siguiente:

- Luego de un Reset, el controlador mantendrá los semáforos tanto de los vehículos como de los peatones en rojo por  $t_{RESET}$  segundos.
- Pasados los  $t_{RESET}$  segundos, el controlador pondrá el semáforo de la vía principal en verde, mientras mantiene los demás semáforos en rojo por un tiempo de  $t_{VERDE1}$  segundos.
- Pasados los  $t_{VERDE1}$  segundos, el controlador pondrá el semáforo de la vía principal en amarillo, mientras mantiene los demás semáforos en rojo por un tiempo de  $t_{AMARILLO1}$

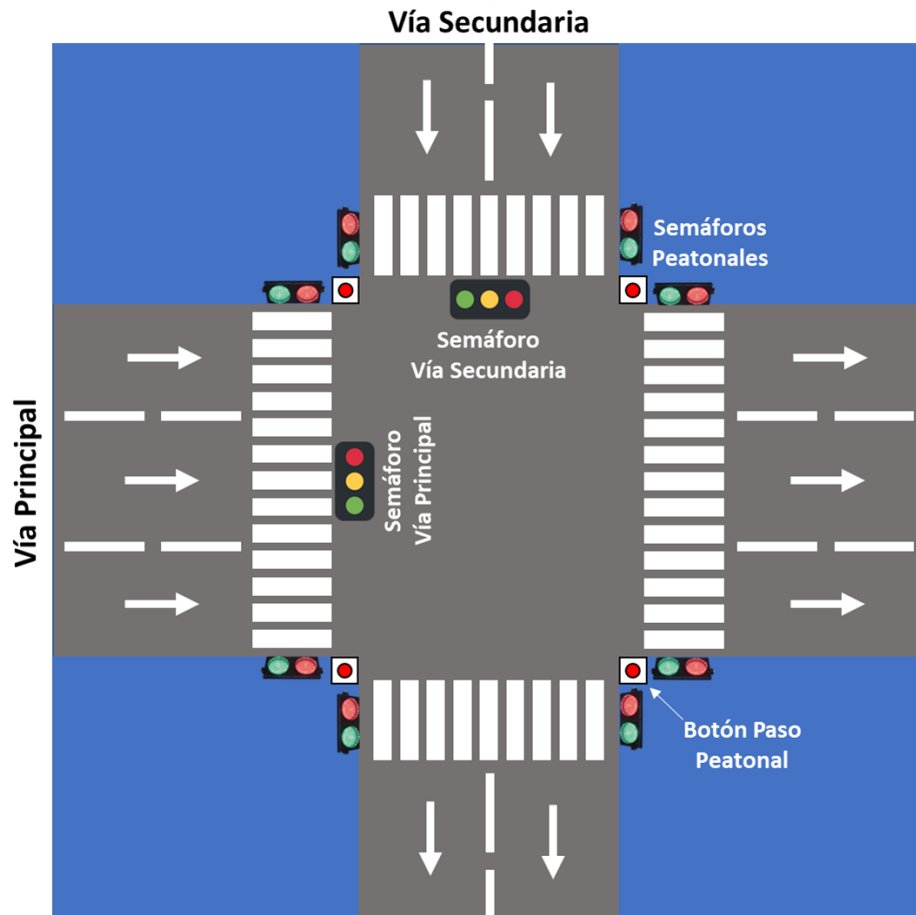


Fig. 2: Diagrama del cruce vial

segundos.

- d. Pasados los  $t_{AMARILLO1}$  segundos, el controlador pondrá el semáforo de la vía secundaria en verde, mientras mantiene los demás semáforos en rojo por un tiempo de  $t_{VERDE2}$  segundos.
- e. Pasados los  $t_{VERDE2}$  segundos, el controlador pondrá el semáforo de la vía secundaria en amarillo, mientras mantiene los demás semáforos en rojo por un tiempo de  $t_{AMARILLO2}$  segundos.
- f. Pasados los  $t_{AMARILLO2}$  segundos, el controlador repetirá la secuencia, poniendo nuevamente el semáforo de la vía principal en verde.

La anterior secuencia podrá ser extendida si, en cualquier momento, un peatón acciona el "Botón Paso Peatonal", lo cual activará la señal luminosa "Semáforo Peatón Solicitado". Si esta señal ("Semáforo Peatón Solicitado") está activa en el momento en que el tiempo  $t_{AMARILLO2}$  es completado, el controlador pondrá el semáforo peatonal en verde en lugar del semáforo de la vía

principal, además de poner los demás semáforos en rojo, por un tiempo de  $t_{VERDE3}$  segundos. Transcurridos los  $t_{VERDE3}$  segundos, el controlador pondrá todos los semáforos en rojo por  $t_{ROJO}$  segundos. Concluido este tiempo ( $t_{ROJO}$  segundos), el controlador volverá a repetir la anterior secuencia, poniendo el semáforo de la vía principal en verde. Durante el tiempo en que el semáforo peatonal se encuentre en verde o el controlador esté en estado de Reset, no se permitirá la activación de la señal "Semáforo Peatón Solicitado". Finalmente, tenga en cuenta que, cuando el controlador ponga en verde el semáforo peatonal, la señal "Semáforo Peatón Solicitado" deberá ser desactivada automáticamente.

Los diferentes periodos de tiempo descritos en la secuencia se indican en la Tab. 1. Recuerde que, en cada etapa de la secuencia, se deberá visualizar el tiempo restante en los displays de 7 segmentos. La visualización debe darse en formato decimal. El rango de valores a visualizar debe estar entre 1 y 99 segundos.

Tab. 1: Tiempos para el controlador de semáforos

Parámetro	Símbolo	Segundos
Tiempo de espera en verde vía principal	$t_{VERDE1}$	18
Tiempo de espera en amarillo vía principal	$t_{AMAR1}$	4
Tiempo de espera en verde vía secundaria	$t_{VERDE2}$	10
Tiempo de espera en amarillo vía secundaria	$t_{AMAR2}$	3
Tiempo de espera en verde - peatones	$t_{VERDE3}$	5
Tiempo de espera en rojo - peatones	$t_{ROJO}$	2
Tiempo después de RESET	$t_{RESET}$	3

## 5 Entrega

El grupo de trabajo deberá escribir un breve reporte en formato IEEE, de máximo 2 páginas, que contenga los siguientes elementos (es posible emplear la guía para reportes en la página del curso):

- Abstract:** resumen del diseño e implementación de la práctica
- Diseño y Simulación:** diagrama de estados de la FSM y una o varias imágenes que muestren el resultado de la simulación realizada. Tenga en cuenta que la simulación debe mostrar toda la secuencia, incluido la puesta en verde del semáforo peatonal.
- Conclusiones:** dos o tres conclusiones sobre el trabajo realizado por el grupo de trabajo. Indicar el tiempo que les tomó realizar la práctica en las conclusiones.

Crear un archivo comprimido que incluya el reporte y los archivos importantes de su proyecto en Quartus Prime como se describe a continuación:

- Reporte:** archivo con extensión .pdf

- b. **Archivos proyecto Quartus:** carpeta del proyecto en Quartus Prime Lite Edition después de realizar un *Clean Project*.

El nombre del archivo comprimido deberá tener el siguiente formato:

*p1\_primerapellidointegrante1\_primerapellidointegrante2\_horariolaboratorio.zip*.

Ejemplo: si el primer apellido de ambos integrantes es **Plazas** y **Garcia**, respectivamente, y el laboratorio es el Martes 9-12, entonces el archivo debe ser nombrado: *p1\_plazas\_garcia\_m9-12.zip*.

## 6 Evaluación

La evaluación de la práctica se divide en tres partes: funcionamiento (40%), sustentación (40%) y reporte (20%). La nota del funcionamiento se asigna por igual a todos los integrantes del grupo de trabajo (máximo dos personas por equipo), mientras que la nota de sustentación es individual. La sustentación podrá realizarse de tres maneras posibles:

- a. Solicitud de cambios al código por parte del profesor de laboratorio.
- b. Un par de preguntas orales que pongan a prueba los conocimientos del estudiante sobre el desarrollo de la práctica.
- c. Una presentación corta a todo el grupo de laboratorio.

En caso un estudiante obtenga una nota inferior a 3.0 en la sustentación, la nota final de la práctica para el estudiante en mención será la que obtuvo en la sustentación, es decir, no se tendrá en cuenta el funcionamiento en el cálculo. Finalmente, en caso no se realice entrega del reporte, la nota asignada a la práctica de laboratorio será de 0.0 así se haya sustentado.

Cada grupo de trabajo deberá sustentar la práctica en un tiempo de 15 minutos, 8 minutos para revisar la implementación y 7 minutos para preguntas. Es importante tener abierto el proyecto y programada la FPGA para cuando el profesor llegue a su puesto de trabajo. No habrá tiempo para hacer correcciones de último momento.

## 7 Referencias

- a. Quartus Prime Lite Edition  
<https://www.intel.com/content/www/us/en/products/details/fpga/development-tools/quartus-prime.html>
- b. Sistema de desarrollo DE10-Lite  
<http://de10-lite.terasic.com/>
- c. Tutorial de SystemVerilog  
<https://verilogguide.readthedocs.io/en/latest/>