

# **Tecnología de Computadores**

## **Práctica 8**



**Universidad  
Rey Juan Carlos**

## 1. Objetivo

En esta octava sesión de prácticas se pretende que el alumno se familiarice con la creación de máquinas de estado en VHDL y su aplicación práctica.

Una vez terminada la práctica el alumno será capaz de:

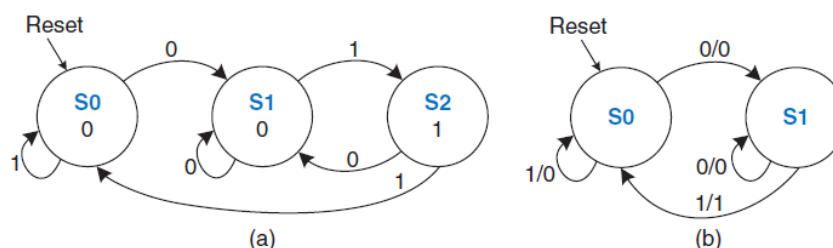
- Entender el funcionamiento de una máquina de estados en VHDL
- Crear una máquina de estados en VHDL a partir de su diagrama de estados
- Crear una máquina de estados en VHDL a partir de sus ecuaciones lógicas
- Diseñar un sistema realista de control temporizado mediante una máquina de estados

## 2. Desarrollo de la práctica

### Máquinas de estado. Moore vs. Mealy

Las máquinas de estado finito (FSM) son circuitos secuenciales utilizados en muchos sistemas de control. En esta primera parte de la práctica se pretende que el alumno se familiarice con la representación de las máquinas de estado de Moore y de Mealy en VHDL.

Para empezar, observe los dos diagramas de estado presentados a continuación:



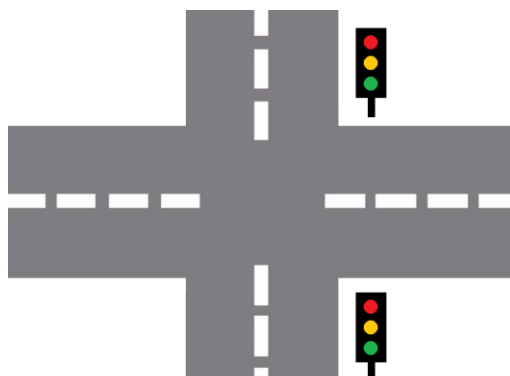
A la izquierda (a) se muestra el diagrama de una máquina de estados de Moore (la salida únicamente depende del estado actual), mientras que a la derecha (b) se muestra el diagrama de una máquina de estados de Mealy (la salida depende del estado actual y del valor de la entrada). Observe los códigos VHDL “*patternMoore.vhd*” y “*patternMealy.vhd*” proporcionados y responda a las siguientes preguntas:

- Ambos códigos describen el mismo funcionamiento de un sistema, ¿cuál es este funcionamiento? Si no lo reconoce a simple vista puede realizar un testbench y simular una de las máquinas
- ¿Cuáles son las principales diferencias entre ambos códigos?
- ¿Cuál de los dos códigos produce un circuito con menor número de recursos? ¿Y cuál genera el circuito más rápido?

Como se ha visto en teoría, una máquina de estados se puede representar en VHDL a partir de su diagrama de estados (como los códigos anteriores) o a partir de sus ecuaciones de excitación y de salida. Esta segunda forma de representación se denomina **forma canónica**. Partiendo del diagrama de estados de la máquina de Mealy anterior, obtenga las ecuaciones de excitación y de salida. Escriba el código VHDL de la forma canónica de la máquina de Mealy anterior. Cree un testbench para simular ambas implementaciones de la máquina (canónica y “*patternMealy.vhd*”). ¿Observa alguna diferencia? ¿A qué cree que es debido?

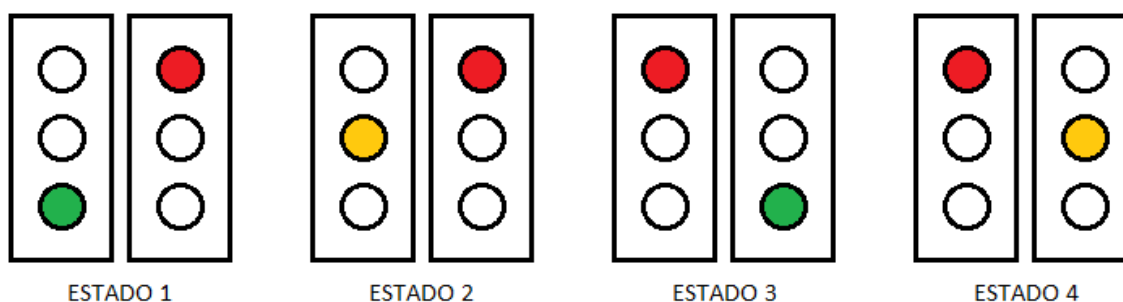
### 3. Ejercicio propuesto

Para terminar la práctica, se le propone al alumno que realice un sistema de control de un cruce con semáforos como el de la siguiente imagen:



El sistema de semáforos controla dos calles, una calle vertical con más tráfico que la calle horizontal que la cruza. Debido a esto, el semáforo de abajo permanece más tiempo en verde que el otro. La descripción del sistema es la siguiente:

- Se comienza en el estado 1, transcurridos 10 segundos se pasa al estado 2.
- Estando en el estado 2, se pasa al estado 3 transcurridos 2 segundos.
- Estando en el estado 3, se pasa al estado 4 transcurridos 5 segundos.
- Estando en el estado 4, se pasa al estado 1 transcurridos 2 segundos.



Dibuje el diagrama de estados y, ayudándose de los códigos de Moore/Mealy presentados antes, escriba el código VHDL que controla este sistema de semáforos. Utilice el fichero de constraints “*cruce\_constraints.xdc*” proporcionado para mapear las 6 luces de los dos semáforos a los LEDs RGB de la placa e implemente el diseño en la FPGA para validar su funcionamiento.

**Ayuda 1:** si observa el fichero de constraints, únicamente aparecen dos salidas para cada semáforo (R y G), el amarillo se consigue encendiendo el color rojo y el verde a la vez.

**Ayuda 2:** ¿no sabe cómo contar segundos? quizás en alguna práctica anterior encuentre ideas...