SISTEMAS ELÉCTRÓNICOS DIGITALES

GUIÓN DE PRÁCTICAS DE LABORATORIO

Versión 2021

PRÁCTICA 3

MÁQUINAS DE ESTADO

Escuela Técnica Superior de Ingeniería y diseño Industrial
Departamento de Electricidad, Electrónica, Automática y Física Aplicada.

1 OBJETIVOS DE LA PRÁCTICA

Con la realización de esta práctica se persiguen los siguientes objetivos:

- Familiarizarse con el desarrollo de máquinas de estado
- Familiarizarse con la creación de programas de pruebas o *testbenches*.
- Familiarizase con el uso de restricciones, en particular con las de asignación de patillas.

2 INTRODUCCIÓN

Las **máquinas de estado finitas** (**FSM** por sus siglas en inglés "Finite State Machines") son circuitos secuenciales utilizados en muchos sistemas digitales para controlar el comportamiento de sistemas y las rutas de flujo de datos.

Las FSM ya se han visto en asignaturas como "Automática", por lo que deberían ser bien conocidas por el alumno. En esta práctica se introduce el concepto de FSM síncrona de Moore aplicado a VHDL.

Como se ha dicho, las FSM se utilizan para diseñar programas de ordenador y circuitos lógicos secuenciales. Se conciben como una máquina abstracta que puede estar en uno de un número finito de estados definidos por el usuario. La máquina puede estar en un solo estado a la vez. El estado en que se encuentra en un momento dado se llama el estado actual. Se cambia de un estado a otro cuando se produce un evento o condición desencadenante. Esto se llama una transición. Una FSM particular se define por una lista de sus estados y la condición de activación para cada transición.

El comportamiento de las máquinas de estado se puede observar en muchos dispositivos actuales que realizan una secuencia de acciones predeterminada en respuesta a una secuencia de eventos. Algunos ejemplos simples son las máquinas expendedoras que dispensan productos cuando se deposita la combinación adecuada de monedas, los ascensores que llevan a los pasajeros a los pisos superiores antes de bajar, los semáforos que cambian la secuencia cuando los coches están esperando, y las cerraduras que requieren la entrada de una combinación de números en el orden correcto.

Las máquinas de estado se modelan utilizando dos tipos básicos de redes secuenciales: Mealy y Moore. En un máquina de Mealy la salida depende tanto del estado actual como de las entradas actuales. En la máquina de Moore, la salida depende solo del estado actual. **Por simplicidad, se recomienda el uso de máquinas de Moore**. También se recomienda que sean **síncronas**, es decir, que los cambios de estado se produzca de acuerdo a una señal sincronizada (reloj).

2.1 Máquina de Moore

A continuación se muestra un modelo general de una máquina secuencial de Moore. Su salida se genera desde el registro de estado actual. El siguiente estado se determina utilizando la entrada actual y el estado actual. Normalmente las máquinas de Moore se describen usando tres bloques, uno de los cuales debe ser secuencial (PROCESS) y los

otros dos pueden ser PROCESS o una combinación de PROCESS y DATAFLOW. Los bloques son: un bloque para calcular el estado siguiente, un bloque para actualizar el estado actual y otro para generar la salida.

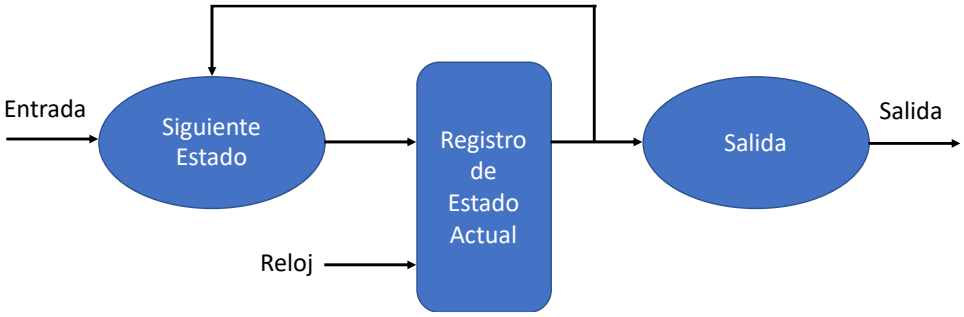

Figura 1. Modelo de bloques de una máquina de Moore

3 DESARROLLO DE LA PRÁCTICA

En esta práctica se va a realizar un FSM sencilla, con el objetivo de adaptarla fácilmente a las necesidades de los trabajos del curso.

Se va a implementar un sistema compuesto por un botón y cuatro LEDs. Existirán cuatro estados (S0, S1, S2 y S3). En cada estado se encenderá un LED diferente. Cada vez que se pulse el botón se cambiará de estado de forma secuencial.

El diagrama de la máquina de estados se puede observar en la Figura 2.

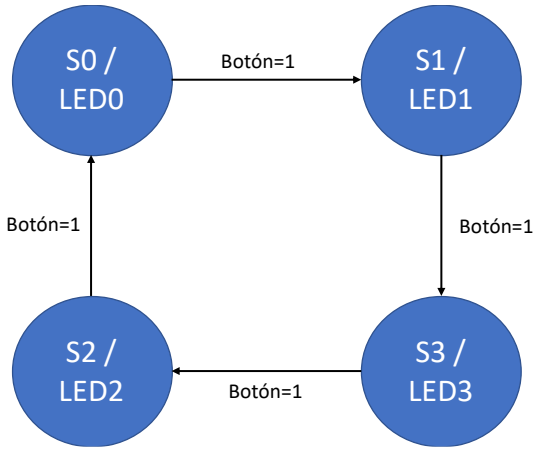

Figura 2. Diagrama de la máquina de estados (Moore)

Para su implementación se va a realizar un proyecto con la siguiente estructura de componentes:

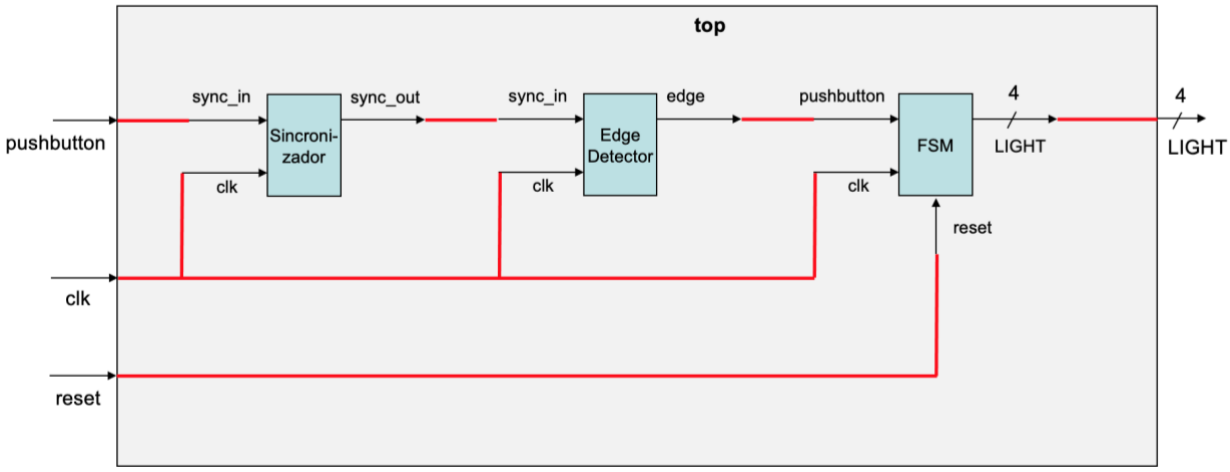

Figura 3. Diagrama de componentes

Los componentes "Sincronizador" y "Edge_Detector" son los mismo utilizados en la práctica 2. Utilice la señal de reloj proporcionada por la placa. Al activar RESET (mediante el pulsador correspondiente de la placa), la máquina de estados vuelve a S0.

Se propone el siguiente código para la FSM:

```
entity fsm is
  port (
    RESET      : in  std_logic;
    CLK        : in  std_logic;
    PUSHBUTTON : in  std_logic;
    LIGHT      : out std_logic_vector(0 TO 3)
  );
end fsm;

architecture behavioral of fsm is
  type STATES is (S0, S1, S2, S3);
  signal current_state: STATES := S0;
  signal next_state: STATES;
begin
  state_register: process (RESET, CLK)
  begin

            COMPLETAR

  end process;
```

```
nextstate_decod: process (PUSHBUTTON, current_state)
begin
  next_state <= current_state;
  case current_state is
    when S0 =>
      if PUSHBUTTON = '1' then
        next_state <= S1;
      end if;
    when S1 =>
      if PUSHBUTTON = '1' then
        next_state <= S2;
      end if;
    when S2 =>
      if PUSHBUTTON = '1' then
        next_state <= S3;
      end if;
    when S3 =>
      if PUSHBUTTON = '1' then
        next_state <= S0;
      end if;
    when others =>
      next_state <= S0;
  end case;
end process;

output_decod: process (current_state)
begin
  LIGHT <= (OTHERS => '0');
  case current_state is
    when S0 =>
      LIGHT(0) <= '1';
    when S1 =>
      LIGHT(1) <= '1';
    when S2 =>
      LIGHT(2) <= '1';
    when S3 =>
      LIGHT(3) <= '1';
    when others =>
      LIGHT <= (OTHERS => '0');
```

```
    end case;
  end process;
end behavioral;
```

Tarea 1

Analice el código de la FSM y asegúrese de que comprende su funcionamiento.

Complete el PROCESS “state_register” para que actualice el estado en cada flanco de reloj y lo resetee cuando la señal de RESET valga ‘0’.

Tarea 2 (Tarea para casa)

Cree un banco de pruebas para la FSM y verifique su funcionamiento a través de una simulación de comportamiento.

Tarea 3

Implemente el circuito en la placa Nexys4DDR. Use uno de los botones como PUSHBUTTON, el botón de RESET como RESET, y los LEDs LED0-LED3 como la salida “LIGHT”. Realice el flujo de diseño, genere el bitstream y descárguelo en la placa Nexys4 DDR. Verifique la funcionalidad.

Tarea 4

Utilice el tiempo restante para esbozar un primer diseño de la máquina de estados de su trabajo. Coméntelo con su profesor.

--- Fin de la práctica ---