

Informe de laboratorio N°4

Fernando Guiraud

Universidad tecnológica de Panamá
Microprocesadores
1EE141
fjguiraud@gmail.com

I. INTRODUCCIÓN

En esta práctica se va a desarrollar la FSM necesaria para operar el semáforo Peatonal y también se van a incorporar al diseño los elementos necesarios para poder descargarla sobre la tarjeta de desarrollo. Estos elementos van a ser el decodificador de 7 segmentos, un circuito anti-rebote y la adición de las restricciones necesarias (asignación de pines) para adecuar el diseño a la tarjeta.

- **Problema:**

La luz normalmente en Verde para tráfico vehicular. Cuando se presiona el BOTON cambia a luz Roja por 20 segundos. Los últimos 5 segundos se muestran en un display 7 segmentos.



II. OBJETIVO

- En este laboratorio el estudiante diseñará una máquina de estados mediante una Máquinas de Estados (FSM) para controlar un semáforo peatonal.

III. MATERIAL Y EQUIPO

- Tarjeta Elbert V2 – Spartan 3A FPGA Development Board



Figura 2: Tarjeta Spartan 3A FPGA

- Xilin ISE, 32-bit Project Navigator.

IV. DESARROLLO

El primer paso para implementar el semisumador después de configurar la tarjeta Elbert V2, consiste en definir las entradas y salidas de este proyecto:

```
entity sema is
  Port ( RST : in  STD_LOGIC;
        CLK : in  STD_LOGIC;
        CLK_out : inout STD_LOGIC;
        Dout : inout STD_LOGIC_VECTOR (4 downto 0);
        EN : inout STD_LOGIC_VECTOR (2 downto 0);
        R : out  STD_LOGIC;
        G : out  STD_LOGIC;
        L : out  STD_LOGIC_VECTOR (7 downto 0));
end sema;
```

El ISE Project Navigator inicializa las librerías básicas para comenzar a elaborar el algoritmo. Sin embargo, para implementar este algoritmo será necesario utilizar dos librerías numéricas adicionales.

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.numeric_std.all;
use IEEE.std_logic_unsigned.all;
```

Para crear este algoritmo, es necesario programar un clock que nos genere una señal que oscile en un periodo de un segundo. Para el primer paso es crear un divisor de frecuencia. Tomando en cuenta que el clock interno de la tarjeta ElbertV2 es de 12MHz, escribimos el siguiente algoritmo para generar una señal de clock_out.

```
begin
Process(RST, CLK)
    variable n: integer range 0 to 12000000:=0;
    constant p: integer :=6000000;
begin
    if(RST='0') then
        n:=0;
    elsif (CLK'event and CLK='0') then
        if n<p then
            CLK_out<='0';
            n:=n+1;
        elsif n>p then
            CLK_out<='1';
            n:=n+1;
        elsif n=2*p then
            CLK_out<='0';
            n:=0;
        else
            CLK_out<='0';
            n:=n+1;
        end if;
    end if;
end process;
```

Posteriormente creamos la maquina de estado encargada de variar entre los tres estados necesarios dentro de otro Process que utiliza la señal de RST y el CLK_out.

```
Process(RST, CLK_out)
    variable c: bit:='1';
begin
    if RST='0' then
        Dout<="10100";
        pr_state<=s0;
        G<='0';
        R<='1';
    elsif CLK_out'event and CLK_out='0' then
        case pr_state is
            when s0 =>
                G<='1';
                R<='0';
                if Dout="00101" then
                    Dout<=Dout-1;
                    pr_state<=s1;
                else
                    Dout<=Dout-1;
                    pr_state<= s0;
                end if;
        end case;
```

```
            when s1 =>
                G<='1';
                R<='0';
                if Dout="00000" then
                    Dout<="10100";
                    G<='0';
                    R<='1';
                    pr_state<=s2;
                else
                    Dout<=Dout-1;
                    pr_state<=s1;
                end if;
            when s2 =>
                if RST='1' then
                    pr_state<=s2;
                else
                    pr_state<=s0;
                end if;
        end case;
    end if;
end process;
```

El primer estado consiste en el conteo de cualquier numero mayor a 5, en este caso representado en binario (00101). En este estado el display de siete segmentos se encuentra apagado pero el contador sigue funcionando, dando una salida que nos sirve de referencia en 5 led ubicados físicamente en la tarjeta.

El segundo estado consiste en el conteo regresivo de los últimos 5 dígitos que son presentados en el display de siete segmentos.

Y por ultimo el tercer y ultimo estado, consiste en un estado de espera hasta que se oprima el botón de RST y se inicie el semáforo desde el estado uno sucesivamente.

Después de definir la maquina de estado, es necesario establecer la secuencia de encendido de cada uno de los leds del siete segmentos.

```
L<="00000011" when Dout = "00000" else
    "10011111" when Dout = "00001" else
    "00100101" when Dout = "00010" else
    "00001101" when Dout = "00011" else
    "10011001" when Dout = "00100" else
    "01001001" when Dout = "00101" else
    "11111111";

EN<="011" when Dout = "00000" else
    "011" when Dout = "00001" else
    "011" when Dout = "00010" else
    "011" when Dout = "00011" else
    "011" when Dout = "00100" else
    "011" when Dout = "00101" else
    "111";
```

El bus de datos L, almacena la combinación correspondiente de cada dígito del 0 al 5, necesarios

para el conteo regresivo del segundo estado de la máquina de estado.

Este proceso ocurre de manera simultanea a la maquina de estado y no corresponde a ningún process.

V. RESULTADOS

Para comprobar la funcionalidad de este código, procedemos a transferir el programa a la tarjeta, para esto tenemos que crear un archivo ucf que contenga la ruta de las entradas y salidas de la tarjeta que correspondan físicamente a el hardware deseado.

En este caso, representaremos el clk con el reloj interno de la tarjeta, RST con un botón, la variable Dout que almacena la cuenta en binario en segundo plano se representa en 5 leds de la tarjeta y el bus de datos L de 8 bits, se asigna al display de siete segmentos. El bus EN, corresponde a los pines de habilitación de los dígitos del display de siete segmentos, al ser 3 displays, se tienen 3 bits de habilitación. Por último, las señales G y R, corresponden a dos leds de la tarjeta, cada uno representando el estado de verde o rojo del semáforo.

```

NET "Dout[4]"          LOC = P46  |
NET "Dout[3]"          LOC = P47  |
NET "Dout[2]"          LOC = P48  |
NET "Dout[1]"          LOC = P49  |
NET "Dout[0]"          LOC = P50  |

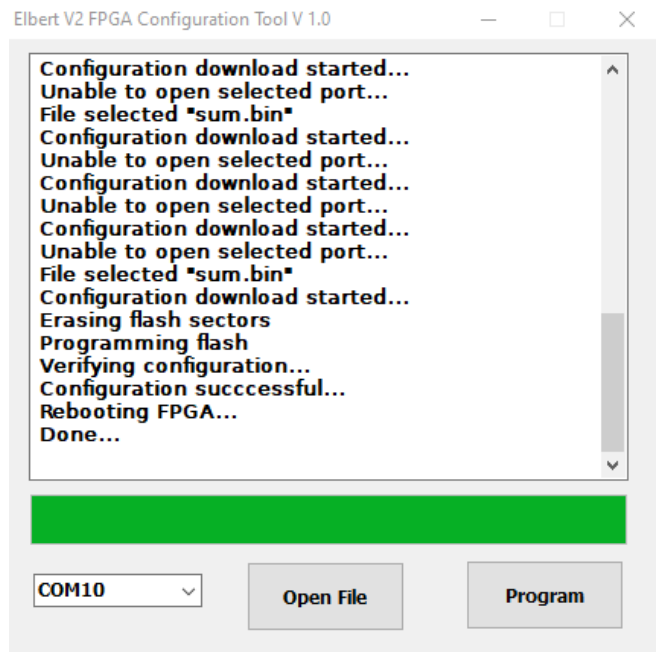
NET "CLK"              LOC = P129 |
NET "RST"              LOC = P80  |

NET "L[7]"             LOC = P117 | IOSTANDARD
NET "L[6]"             LOC = P116 | IOSTANDARD
NET "L[5]"             LOC = P115 | IOSTANDARD
NET "L[4]"             LOC = P113 | IOSTANDARD
NET "L[3]"             LOC = P112 | IOSTANDARD
NET "L[2]"             LOC = P111 | IOSTANDARD
NET "L[1]"             LOC = P110 | IOSTANDARD
NET "L[0]"             LOC = P114 | IOSTANDARD

NET "EN[0]"            LOC = P124 | IOSI
NET "EN[1]"            LOC = P121 | IOSI
NET "EN[2]"            LOC = P120 | IOSI

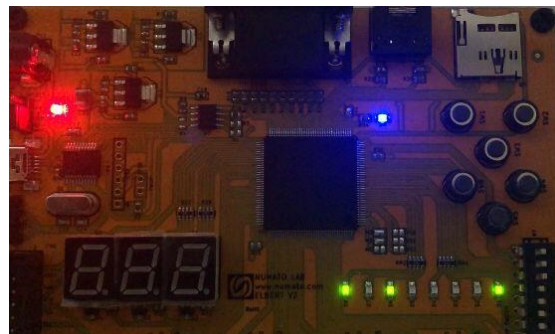
NET "G"                LOC = P54  | IOSTA
NET "R"                LOC = P55  | IOSTA
  
```

Ahora procedemos a transferir el código a la tarjeta con el ejecutable proporcionado por el fabricante, en este caso Numato.

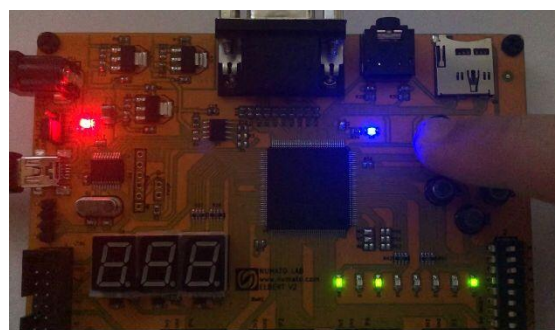


Escogemos el puerto correspondiente y enviamos el programa.

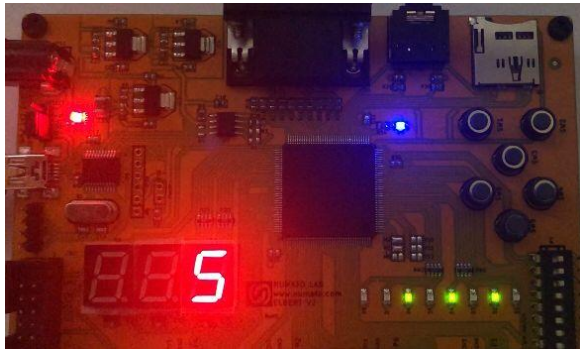
En las siguientes imágenes podemos ver el funcionamiento de la máquina de estado.



Esta primera imagen corresponde al estado de espera, el contador digital representado por los primeros 5 leds de la tarjeta se mantiene en 20 y el ultimo led representa el led rojo del semáforo.

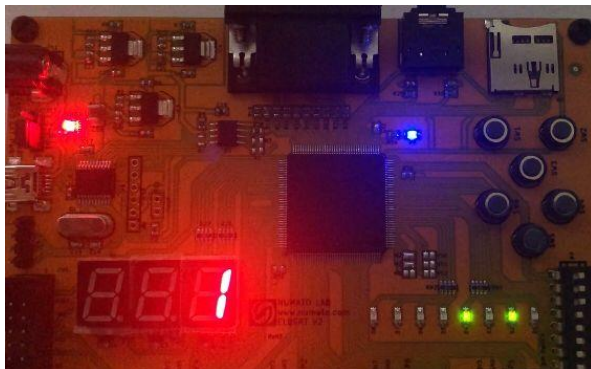


Al presionar el botón de RST que corresponde al primer switch, comienza la cuenta en segundo plano sin ser presentada en el display de siete segmentos hasta que llega al dígito 5.



Después se mantiene el conteo hasta que llega a cero y vuelve a estar en estado de espera a que se presione el botón de RST.

En la entrega del informe se adjunta un video del funcionamiento en tiempo real de la tarjeta.



VI. CONCLUSIÓN

Las máquinas de estado nos sirven para definir distintos procesos de manera estructurada y de manera secuencial, siendo una herramienta muy útil al momento de implementar un sistema de control.

Las máquinas de estado son secuencias lógicas que, utilizando distintas comprobaciones de estado, se mueven de una función a otra de cualquier orden deseado.

Un display de siete segmentos consta de ocho leds correspondientes a las siete partes de un dígito más un punto decimal y la correcta escritura de los dígitos depende de la secuencia en binario o hexadecimal que indique el orden de encendido de los leds.

VII. REFERENCIAS

D. L. Perry, VHDL. New York: McGraw-Hill, 1991

