



# **INSTITUTO POLITÉCNICO NACIONAL ESCUELA SUPERIOR DE CÓMPUTO**



## **Carrera:**

Ingeniería en sistemas computacionales

## **Unidad de Aprendizaje:**

Sistemas Embebidos

## **Practica 2:**

Práctica de Marquesina - Verilog

## **Integrantes:**

Cazares Cruz Jeremy Sajid

Bazo Machuca Ilse Paola

Acosta Cortes Gerardo

Guzmán Ballesteros Armando

## **Fecha de entrega:**

07 de noviembre de 2023

## Descripción General

En este proyecto vamos a crear una marquesina usando un display de 7 segmentos en una FPGA de la familia Cyclone IV E (modelo EP4CE115F29C7). Donde se mostrará la palabra "FPGA" en el display de 7 segmentos y luego se irá recorriendo.

## Descripción de la práctica:

Se definen los puertos de entrada y salida:

- **input clk:** El reloj del sistema.
- **input [1:0] switch:** Nuestros switches de entrada de 2 bits  
**input [1:0] switch1:** Nuestros switches de entrada de 2 bits.
- **output [6:0] display0, display1, display2, display3, display4, display5, display6, display7:** Estos son puertos de salida de 7 bits cada uno que controlan los displays de siete segmentos.

```
module fpga
(
    input clk,
    input [1:0] switch,
    input [1:0] switch1,
    output [6:0] display0,
    output [6:0] display1,
    output [6:0] display2,
    output [6:0] display3,
    output [6:0] display4,
    output [6:0] display5,
    output [6:0] display6,
    output [6:0] display7
);
```

A continuación, se declaran varios registros y variables que se utilizarán en el módulo:

- **reg fre:** Un registro que almacena la frecuencia de actualización actual.
- **reg cont:** Un registro para almacenar el valor máximo de conteo antes de actualizar los displays.
- **reg[22:0] count:** Un registro de 23 bits llamado "count" utilizado para contar el número de ciclos de reloj.
- **reg [2:0] disp0, disp1, disp2, disp3, disp4, disp5, disp6, disp7:** Estos registros almacenan los valores que se mostrarán en los displays de siete segmentos.
- **reg clk2:** Un registro que toma el estado del reloj.
- **reg [2:0] temp:** Un registro temporal utilizado para realizar intercambios de valores.

- **parameter frecuencia=50000000:** Se define un parámetro con un valor predeterminado de 50,000,000 (50 MHz).

```
parameter frecuencia=50000000;
reg fre;
reg cont;
reg[22:0] count;
reg [2:0] disp0, disp1, disp2, disp3, disp4, disp5, disp6, disp7;
reg clk2;
reg [2:0] temp;
```

Dentro del módulo initial begin, se establecen valores iniciales para varios registros y variables:

```
initial begin
    count=0;
    fre=10;
    clk2=0;
    cont=frecuencia/(2*fre);
    disp0 = 011;
    disp1 = 010;
    disp2 = 001;
    disp3 = 000;
    disp4 = 111;
    disp5 = 111;
    disp6 = 111;
    disp7 = 111;
end
```

Luego en nuestro módulo Always se ejecuta en cada flanco de subida del reloj actualizando los valores de los displays y otros registros en función de las entradas y del conteo.

- Se selecciona una frecuencia de actualización en función del valor de **switch1**.
- El contador **count** se utiliza para contar los flancos de subida del reloj.
- Los valores de los **displays** se desplazan en función de los valores de **switch**.

```

always @(posedge clk)
begin
    case (switch1)
        2'b00:
            fre = 2;
        2'b01:
            fre = 4;
        2'b10:
            fre = 6;
        2'b11:
            fre = 8;
    endcase

```

```

        if(count==cont)begin
            //clk2 = -clk2;
            count=0;
            if(switch[0]==0 && switch[1]==1)begin
                temp = disp7;
                disp7 = disp6;
                disp6 = disp5;
                disp5 = disp4;
                disp4 = disp3;
                disp3 = disp2;
                disp2 = disp1;
                disp1 = disp0;
                disp0 = temp;
            end
            else if(switch[0]==1 && switch[1]==0)begin
                temp = disp0;
                disp0 = disp1;
                disp1 = disp2;
                disp2 = disp3;
                disp3 = disp4;
                disp4 = disp5;
                disp5 = disp6;
                disp6 = disp7;
                disp7 = temp;
            end
        end
    else begin
        count=count+1;
    end
end

```

Se colocan varios módulos del tipo "disp7" para mostrar los valores que va a mostrar el display de siete segmentos. Cada instancia toma un valor disp[ ] como

entrada y muestra el resultado en uno de nuestros valores display [ ] pasando cada uno de nuestros valores de disp [ ] ocurriendo un total de 8 veces.

```
disp7 disp0(  
    .input_number(disp0),  
    .display_output(display0)  
);  
  
disp7 disp1(  
    .input_number(disp1),  
    .display_output(display1)  
);
```

Ahora vamos a definir nuestro modulo “disp7” donde primero declaramos nuestros puertos

- **input [2:0] num:** Un puerto de entrada de 3 bits.
- **output reg [6:0] display:** Un puerto de salida de 7 bits el cual es la salida de nuestro display de 7 segmentos.

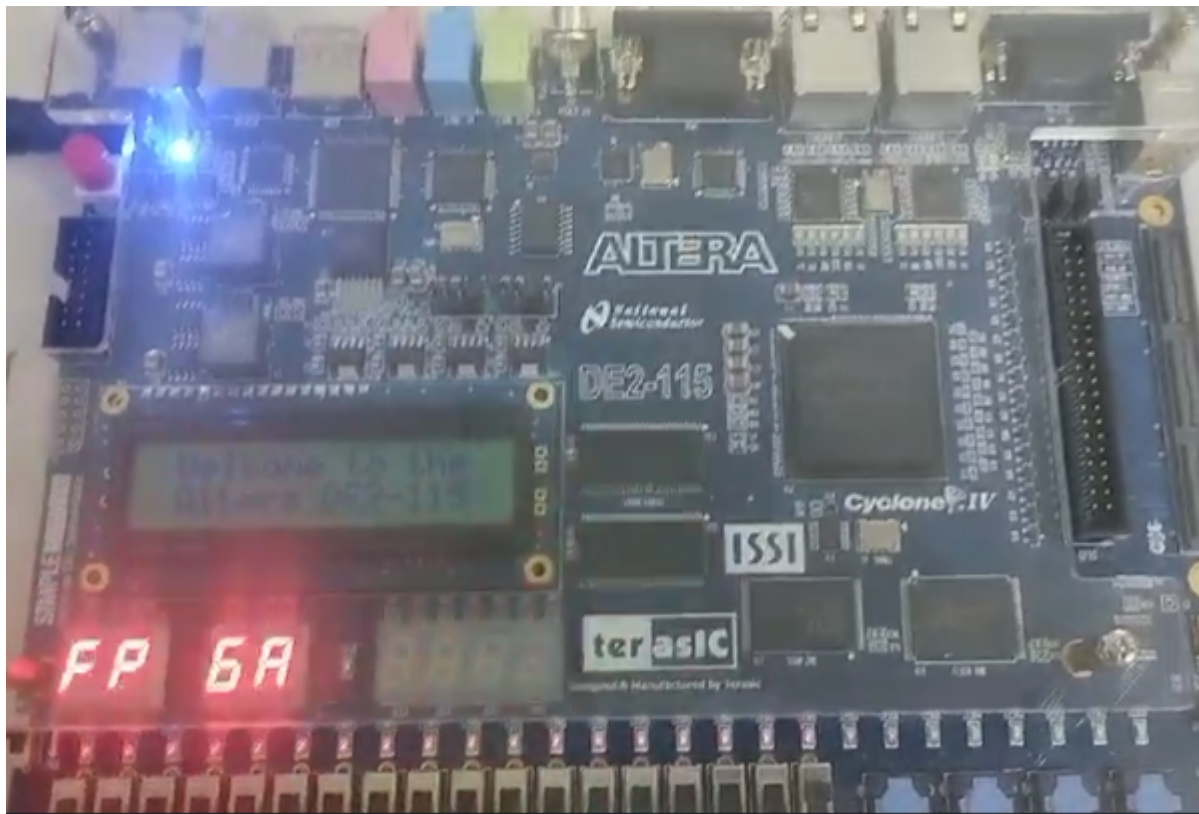
```
module disp7(  
    input [2:0] num,  
    output reg [6:0] display  
);
```

Luego tenemos nuestro módulo always el cual va a ir actuando dependiendo del valor de nuestra entrada “número” entrando de esta forma a un case el cual funciona como decodificador indicando que se debe mostrar en el display de salida.

Los primeros 4 casos representan las combinaciones del display de 7 segmentos para que muestre las letras f,p,g,a para formar la palabra “fpga”.

```
always @(num)  
begin  
    case(num)  
        3'b000: // f  
            display = 7'b0001110;  
        3'b001: // p  
            display = 7'b0001100;  
        3'b010: // g  
            display = 7'b0000010;  
        3'b011: // a  
            display = 7'b0001000;  
        3'b111: // vacio  
            display = 7'b1111111;  
    endcase  
end
```

## Aplicación en la FPGA Cyclone IV E



## Conclusión

Este proyecto es una marquesina en la cual tuvimos que usar herramientas como el uso de **always** para detectar cambios en los valores de entrada y así poder modificar los valores del display con nuestro modulo disp7 el cual consiste de un **case**. Este proyecto es capaz de ejecutarse una FPGA Cyclone IV E. permitiéndonos modificar los valores de entrada de los switches para así modificar la salida de nuestros display de 7 segmentos.