

Primeros Pasos Apio

Dr. Ing. Pablo COSSUTTA

Octubre 2019

¿Qué se va a hacer?

- Instalar Python
 - ▶ Seleccionar pip
- Instalar Apio
- Crear un directorio de trabajo
- Cargar los archivos
- Synthesis, Map, Place & Route, Bitstream
- Cargar el diseño
- Simular

¿Qué se necesita?

- Archivo HDL (`gs_apio.v`)
- Constraints para implementación (`gs_apio.pcf`)
- Testbench (`gs_apio_tb.pcf`)

gs_apio.v - Definition and clock divider

- Similar al anterior solo modificar el nombre rgb_reg

```
module top(  
    output RGB0,  
    output RGB1,  
    output RGB2,  
    input  clk  
);  
  
localparam DIV_N = 22;  
  
reg [DIV_N-1:0] div = 0;  
always @(posedge clk) begin  
    div <= div + 1'd1;  
end  
  
reg en;  
always @(posedge clk) begin  
    if (div==0) en <= 1'b1;  
    else en <= 1'b0;  
end
```

```
reg [2:0] rgb = 3'd0;  
always @(posedge clk) begin  
    if (en) begin  
        case (rgb)  
            3'b000: rgb <= 3'b001;  
            3'b001: rgb <= 3'b010;  
            3'b010: rgb <= 3'b100;  
            3'b100: rgb <= 3'b000;  
            default: rgb <= 3'b000;  
        endcase  
    end  
end
```

- Cambia el nombre de la primitiva

```
SB_RGBA_DRV_RGBA_DRIVER (  
    .CURREN(1'b1),  
    .RGBLEDEN(1'b1),  
    .RGB0PWM(rgb[0]),  
    .RGB1PWM(rgb[1]),  
    .RGB2PWM(rgb[2]),  
    .RGB0(RGB0),  
    .RGB1(RGB1),  
    .RGB2(RGB2)  
);  
  
// Set parameters of RGBA_DRIVER  
defparam RGBA_DRIVER.CURRENT_MODE = "0b0";  
defparam RGBA_DRIVER.RGB0_CURRENT = "0b000001"  
defparam RGBA_DRIVER.RGB1_CURRENT = "0b000001"  
defparam RGBA_DRIVER.RGB2_CURRENT = "0b000001"  
  
endmodule
```

Restricciones (Constraints)

Restricciones lógicas

- No disponibles en Yosys

Restricciones físicas

- Necesarias para la implementación
- Hay que definir todo lo que sea E/S

```
gs_apio_1.pcf
```

```
set_io clk 35  
set_io RGB0 39  
set_io RGB1 40  
set_io RGB2 41
```

Testbench (analizar) gs_apio_tb.v

```
'default_nettype none
'timescale 1 ns / 100 ps

module led_cnt_tb();
parameter DURATION = 200;

reg clk;
always #1 clk = ~clk;

wire RGB0, RGB1, RGB2;
top top_inst (.RGB0(RGB0), .RGB1(RGB1), .RGB2(RGB2), .clk(clk));

initial begin
    $dumpfile("gs_apio_tb.vcd");
    $dumpvars(0, led_cnt_tb);
    #(DURATION) $display("End of simulation");
    $finish;
end

endmodule
```

Primitiva faltante

- Necesitamos la primitiva SB_RGBA_DRV para poder simular

```
// Define primitive for simulation
module SB_RGBA_DRV #(
    parameter CURRENT_MODE=0,
    parameter RGB0_CURRENT=0,
    parameter RGB1_CURRENT=0,
    parameter RGB2_CURRENT=0
)()
    input  CURREN,
    input  RGBLEDEN,
    input  RGB0PWM,
    input  RGB1PWM,
    input  RGB2PWM,
    output RGB0,
    output RGB1,
    output RGB2
);
assign RGB0 = (CURREN & RGBLEDEN) ? RGB0PWM : 1'b0;
assign RGB1 = (CURREN & RGBLEDEN) ? RGB1PWM : 1'b0;
assign RGB2 = (CURREN & RGBLEDEN) ? RGB2PWM : 1'b0;
endmodule
```

Testbench (corregido) gs_apio_tb.v

```
'default_nettype none
'timescale 1 ns / 100 ps

module top_tb();
parameter DURATION = 200;

reg clk = 0;
always #1 clk = ~clk;

wire RGB0, RGB1, RGB2;
top top_inst (.RGB0(RGB0), .RGB1(RGB1), .RGB2(RGB2), .clk(clk));

initial begin
    $dumpfile("gs_apio_tb.vcd");
    $dumpvars(0, top_tb);
    #(DURATION) $display("End of simulation");
    $finish;
end

endmodule
```


Trabajo en clase (Hands On)

- Registrar 2 veces la entrada
 - ▶ Prevenir metaestabilidad
- Realizar el debouncing de una entrada correspondiente a un pulsador
 - ▶ Considerando un debounce de N ciclos (ambas direcciones)
 - ▶ Al detectar un pulso se enciende el led Rojo durante 500ms
- Realizar un testbench completo
- Implementar y verificar experimentalmente
- Modificar para que no vuelva a encenderse si se mantiene el pulsador apretado