

Primeros Pasos Radiant

Dr. Ing. Pablo COSSUTTA

Octubre 2019

¿Qué se va a hacer?

- Instalar Radiant
 - ▶ Requiere registrarse
 - ▶ Requiere licencia
- Crear un proyecto
- Cargar los archivos en el proyecto
- Synthesis, Map, Place & Route, Bitstream
- Cargar el diseño

¿Qué se necesita?

- Archivo HDL (top.v)
- Constraints para síntesis (impl_1.ldc)
- Constraints para implementación (impl_1.pdc)

```
module top(  
    output RGB0,  
    output RGB1,  
    output RGB2,  
    input  clk  
);  
  
    reg [21:0] div = 22'd0;  
    always @(posedge clk) div <= div + 1'd1;  
  
    reg en;  
    always @(posedge clk) begin  
        if (div==22'd0) en <= 1'b1;  
        else en <= 1'b0;  
    end
```

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

```
RGB #(
    .CURRENT_MODE ("1"),
    .RGB0_CURRENT ("0b000001"),
    .RGB1_CURRENT ("0b000001"),
    .RGB2_CURRENT ("0b000001")
) RGB_INST (
    .CURREN    (1'b1),           // 1
    .RGBLEDEN  (1'b1),           // 1
    .RGB0PWM    (rgb_reg[0]),     // 1 - Blue
    .RGB1PWM    (rgb_reg[1]),     // 1 - Green
    .RGB2PWM    (rgb_reg[2]),     // 1 - Red
    .RGB0       (RGB0),           // 0
    .RGB1       (RGB1),           // 0
    .RGB2       (RGB2),           // 0
);
endmodule
```

Restricciones (Constraints)

Restricciones lógicas

- Generalmente necesarias para la síntesis

impl_1.ldc

```
create_clock -name {clk} -period 83.333  
-waveform \{0.000 41.666\} [get_ports clk]
```

Restricciones físicas

- Necesarias para la implementación

impl_1.pdc

```
ldc_set_location -site {35} [get_ports clk]
```

Trabajo en clase (Hands On)

Utilizando los conocimientos adquiridos y la documentación correspondiente a la plataforma, implementar el siguiente diseño:

```
module top(  
    output j05 ,  
    input j01 , j02 , j03 , j04 );  
  
    assign j05 = j01 & j02 & j03 & j04 ;  
endmodule
```

Cargar en la FPGA y analizar:

- Analizar RTL
- Modificar lógica y analizar recursos
- Agregar una 5^{ta} entrada adicional, repetir