Fundación FULGOR

Diseño Digital Avanzado

Unidad 1 - Herramientas

Dr. Ariel L. Pola
apola@fundacionfulgor.org.ar
August 8, 2021

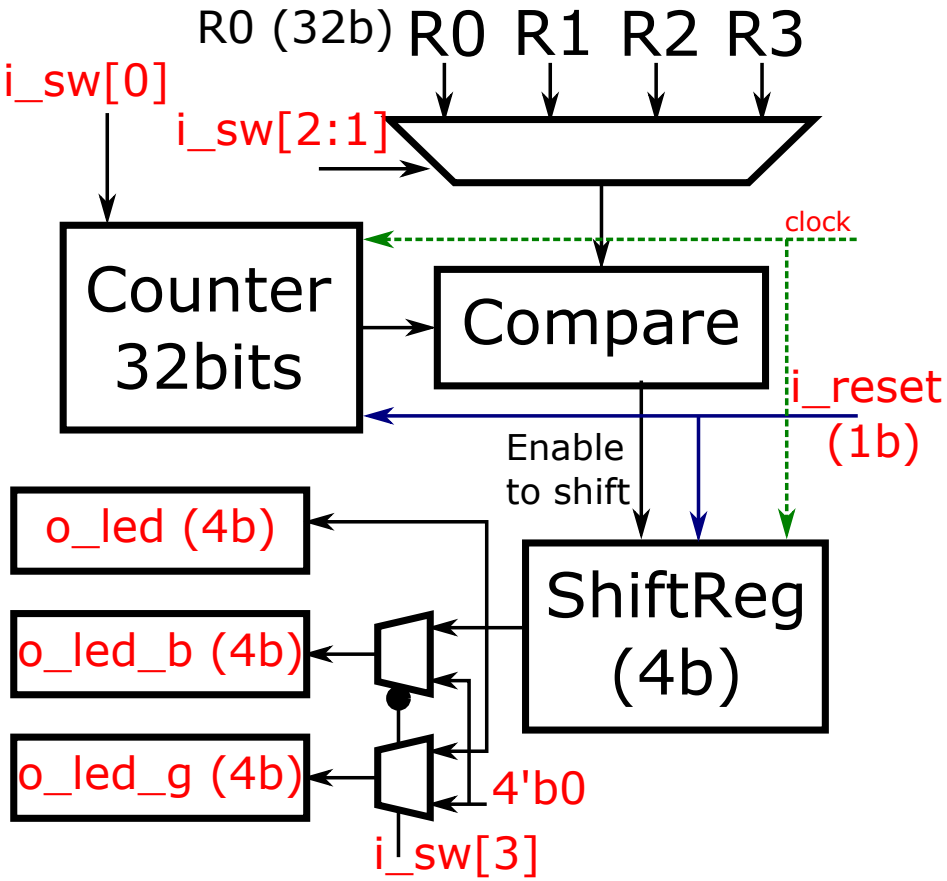
Tabla de Contenidos

Proyecto Leds

Proyecto LEDS
Implementación en FPGA - Leds

Descripción

- Los nombres en ROJO son puertos.
- *ck_rst* es el reset del sistema, el cual pone a cero el contador e inicializa el shiftregister (SR).
- *i_sw[0]* controla el enable (1) del contador. En estado (0) todo se detiene sin alterar el estado actual del contador y del SR.
- El SR se desplaza únicamente cuando el contador llegó a algún límite R0-R3.
- La elección del límite se puede realizar en cualquier momento del funcionamiento.
- *i_sw[3]* elije el color de los leds RGB.



Proyecto LEDS
Test Bench

Descripción

- El test bench (banco de pruebas) es un módulo que no tiene puertos declarados.
- Genera los estímulos de reloj y señales de control para modelar un escenario de prueba.
- Las variables utilizadas para estimular los puertos de entrada son declaradas como tipo **reg**.
- Las variables utilizadas para conectar los puertos de salida son declaradas como tipo **wire**.
- Lectura y cambio de estado de variables internas a los módulos
  - Definiendo las instancias se puede leer las variables internas
    Ejemplo, assign tb_count = tb_shiftleds.u_shiftleds.counter;
  - Utilizando **force** se cambia el valor de una variable en una instancia. Se debe definir dentro de un bloque de procesamiento **initial**.
    Ejemplo, force tb_shiftleds.u_shiftleds.o_led = 4'b0001;

Proyecto LEDS

Test Bench

Ejemplo - Generando Estímulos

```
1  `define N_LEDS 4
2  `define NB_SW 4
3
4  `timescale 1ns/100ps
5
6  module tb_shiftleds();
7
8     parameter N_LEDS   = `N_LEDS   ;
9
10    wire [N_LEDS - 1 : 0] o_led    ;
11    reg [NB_SW   - 1 : 0] i_sw     ;
12    reg                   ck_rst   ;
13    reg                   CLK100MHZ;
14
15    initial begin
16       i_sw          = 4'b0000 ;
17       CLK100MHZ     = 1'b0    ;
18       ck_rst        = 1'b0    ;
19       #100 ck_rst   = 1'b1    ;
20       #100 i_sw     = 4'b0001 ;
21       #1000000 i_sw = 4'b0011 ;
22       #1000000 i_sw = 4'b1011 ;
23       #1000000 $finish;
24    end
```

```
1
2  always #5 CLK100MHZ = ~CLK100MHZ;
3
4  shiftleds
5    #(.N_LEDS   (N_LEDS)   ,
6      .NB_SW    (NB_SW)
7     )
8  u_shiftleds
9    (.o_led     (o_led)    ,
10    .i_sw      (i_sw)     ,
11    .ck_rst    (ck_rst)   ,
12    .CLK100MHZ (CLK100MHZ)
13    );
14
15 endmodule // tb_shiftleds
```

Proyecto LEDS

Test Bench

Ejemplo - Estímulos desde Archivos

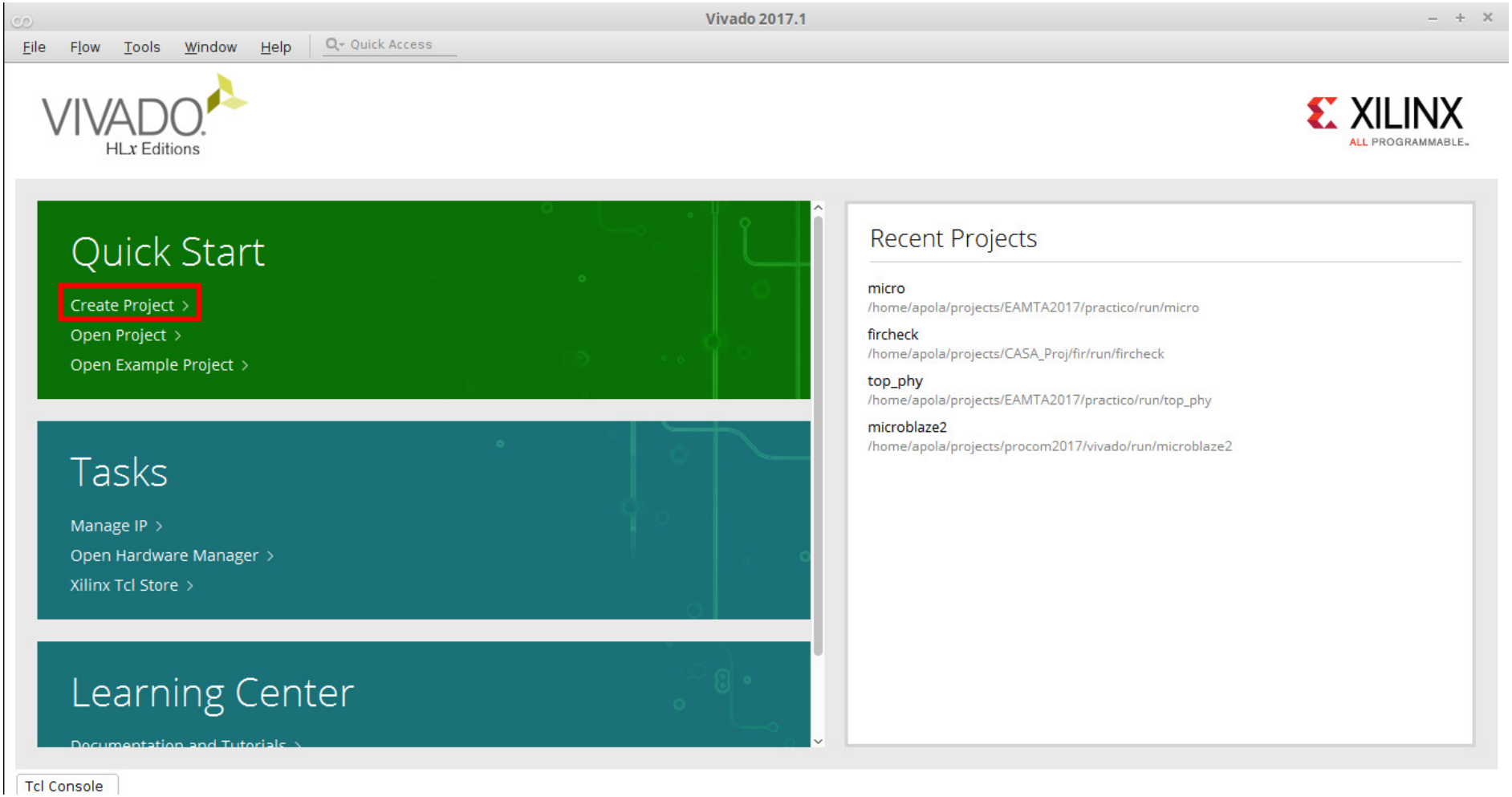
```
1  `define N_LEDS 4
2  `define NB_SW 4
3
4  `timescale 1ns/100ps
5
6  module tb_shiftleds_file();
7
8     parameter N_LEDS    = `N_LEDS    ;
9     parameter NB_SW     = `NB_SW     ;
10
11    wire [N_LEDS - 1 : 0] o_led     ;
12    reg [NB_SW   - 1 : 0] i_sw      ;
13    reg [NB_SW   - 1 : 0] sw_tmp    ;
14    reg ck_rst, CLK100MHZ, reset_tmp;
15
16    integer fid_reset,fid_sw;
17    integer code_error,code_error1;
18    integer             ptr_sw;
19
20    initial begin
21       fid_reset  = $fopen("./vectors/reset.out"
         ,"r");
22       if(fid_reset==0) $stop;
23       fid_sw = $fopen("./vectors/switch.out","r
         ");
24       if(fid_sw==0) $stop;
25       CLK100MHZ    = 1'b0  ;
26    end
```

```
1     always #5 CLK100MHZ = ~CLK100MHZ;
2
3     always@(posedge CLK100MHZ) begin
4        code_error <=
5           $fscanf(fid_reset,"%d",reset_tmp);
6        if(code_error!=1) $stop;
7
8        for(ptr_sw=0;ptr_sw<NB_SW;
9            ptr_sw = ptr_sw+1) begin
10        code_error1 <=
11           $fscanf(fid_sw,"%d",sw_tmp[(ptr_sw+1)
         -1 -: 1]);
12        if(code_error1!=1) $stop;
13    end
14
15       ck_rst <= reset_tmp;
16       i_sw   <= sw_tmp;
17       $display("%d",ck_rst);
18    end
19
20    shiftleds
21      u_shiftleds
22        (.o_led     (o_led)    ,
23         .i_sw      (i_sw)     ,
24         .ck_rst    (ck_rst)   ,
25         .CLK100MHZ (CLK100MHZ));
26
27 endmodule // tb_shiftleds
```

Herramienta Vivado

Herramienta Vivado

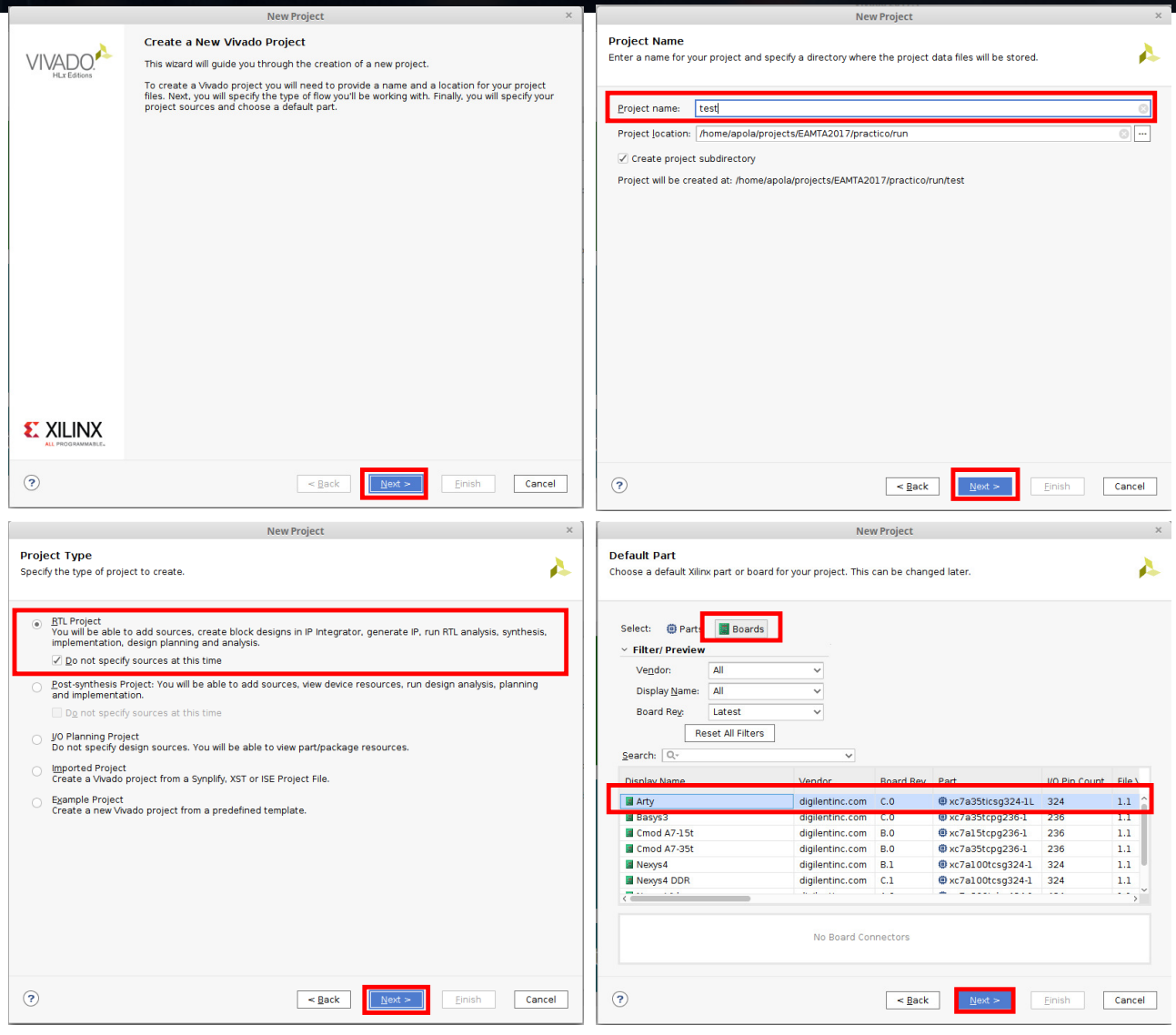
Vivado



Crea un nuevo proyecto

Herramienta Vivado

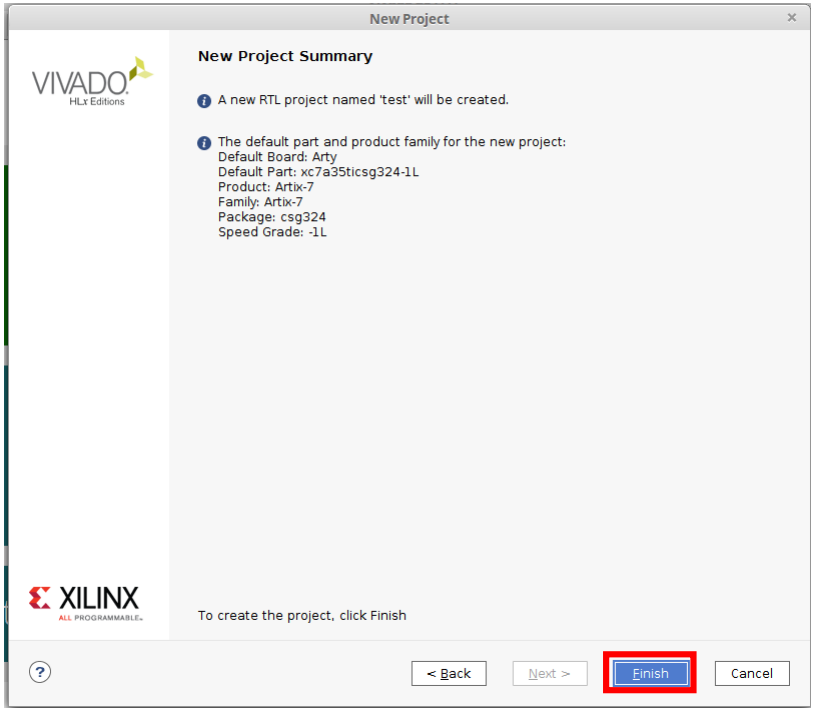
Vivado



Configura el kit de trabajo

Herramienta Vivado

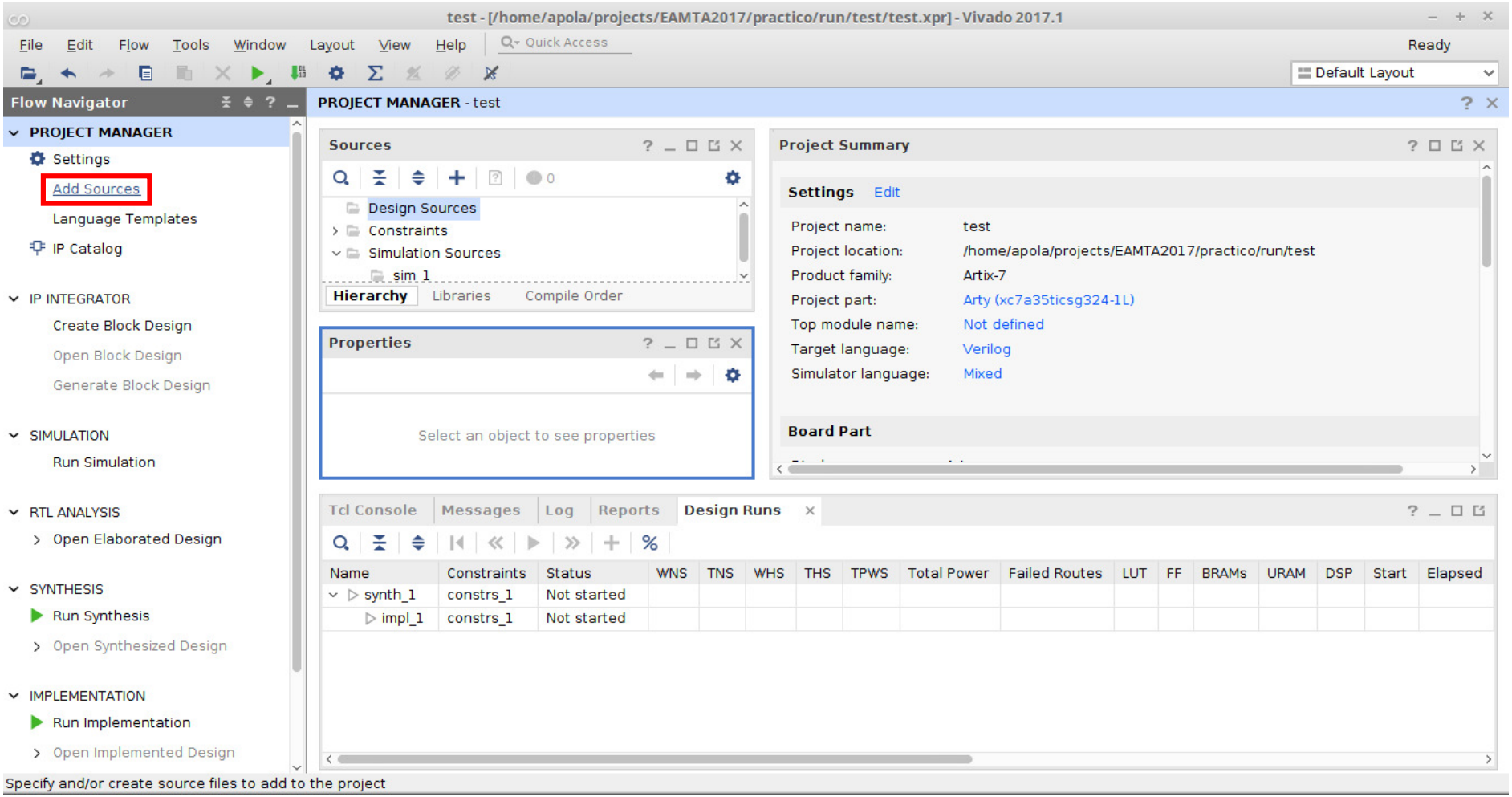
Vivado



Configura el kit de trabajo

Herramienta Vivado

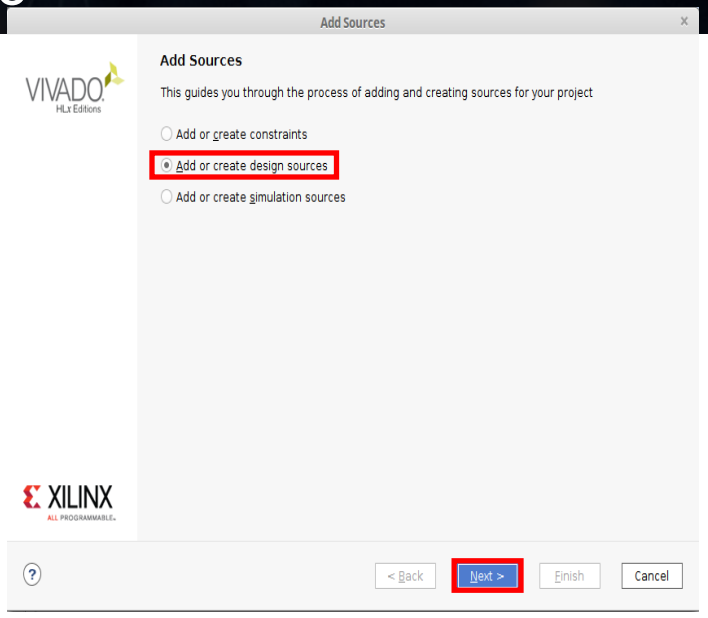
Vivado

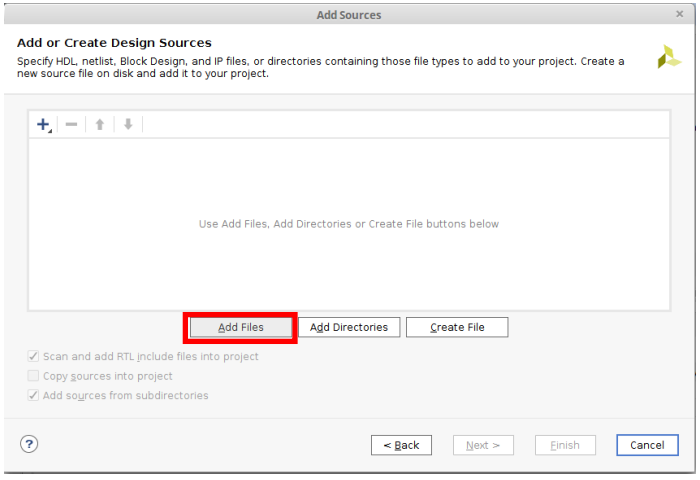


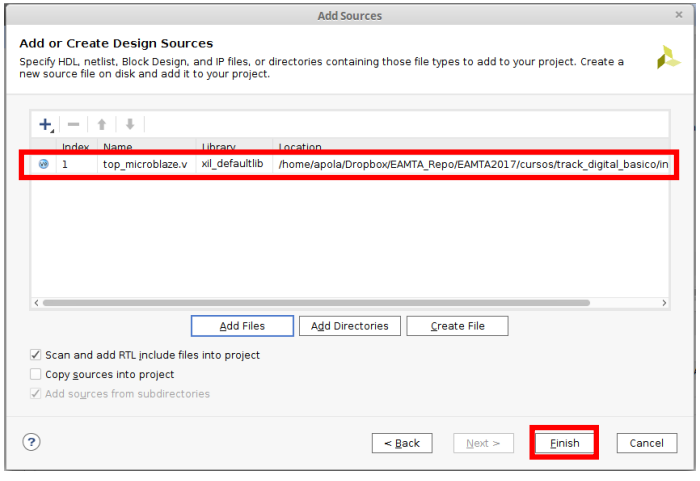
Agrega nuevas fuentes a *Design Sources*

Herramienta Vivado

Vivado



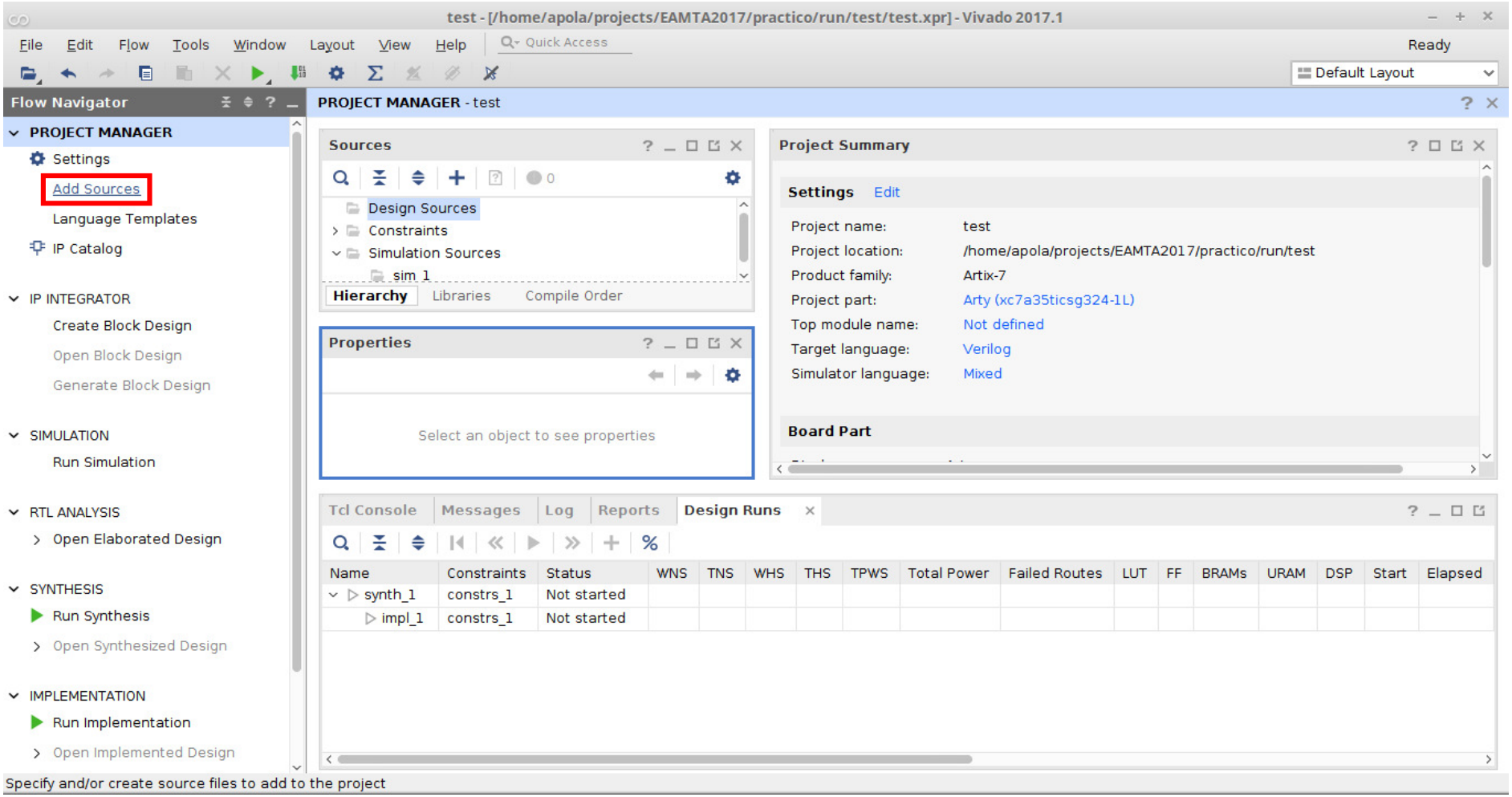




Seleccionar todos los archivos verilog relacionados al diseño

Herramienta Vivado

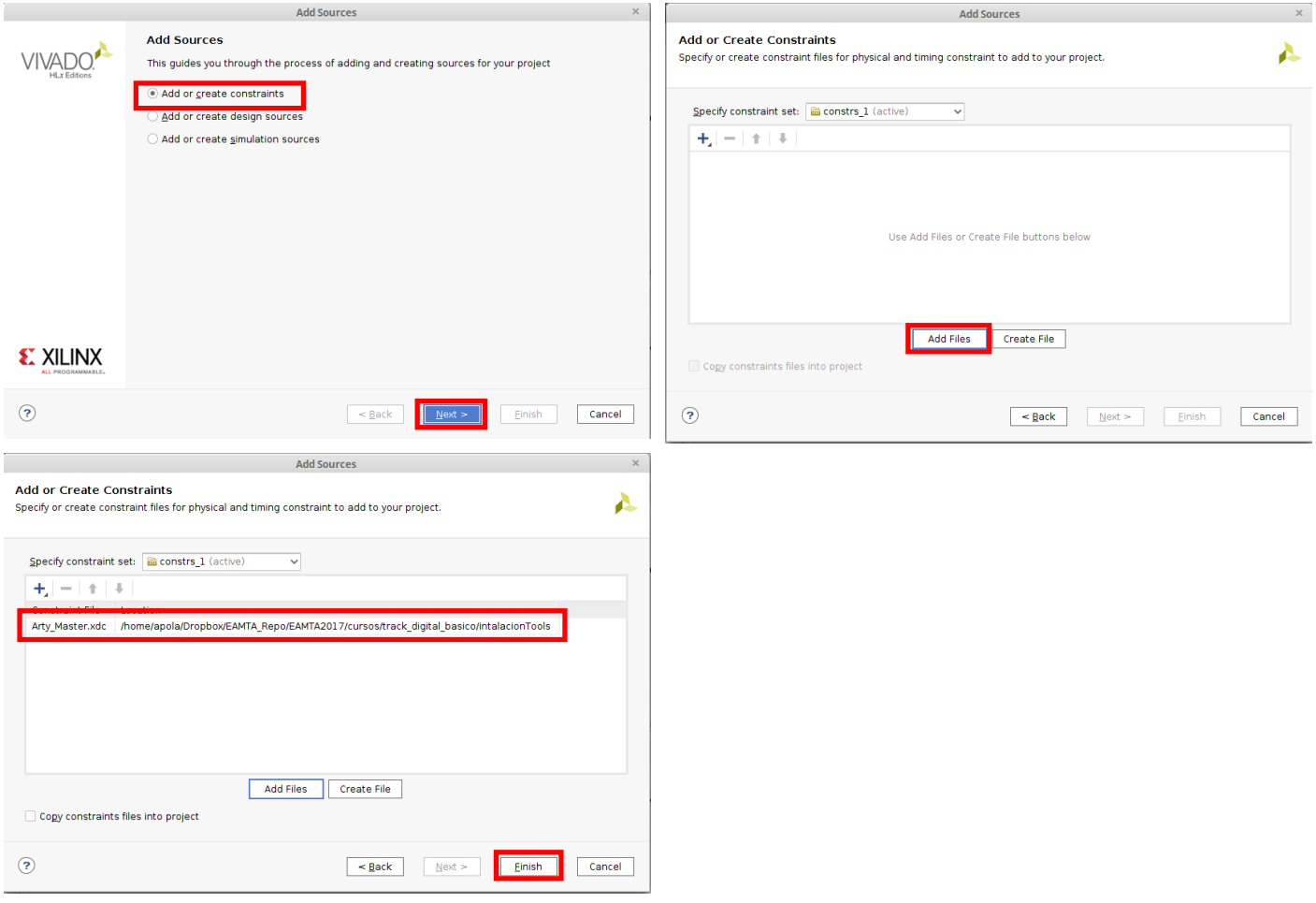
Vivado



Agrega nuevas fuentes a *Constraints*

Herramienta Vivado

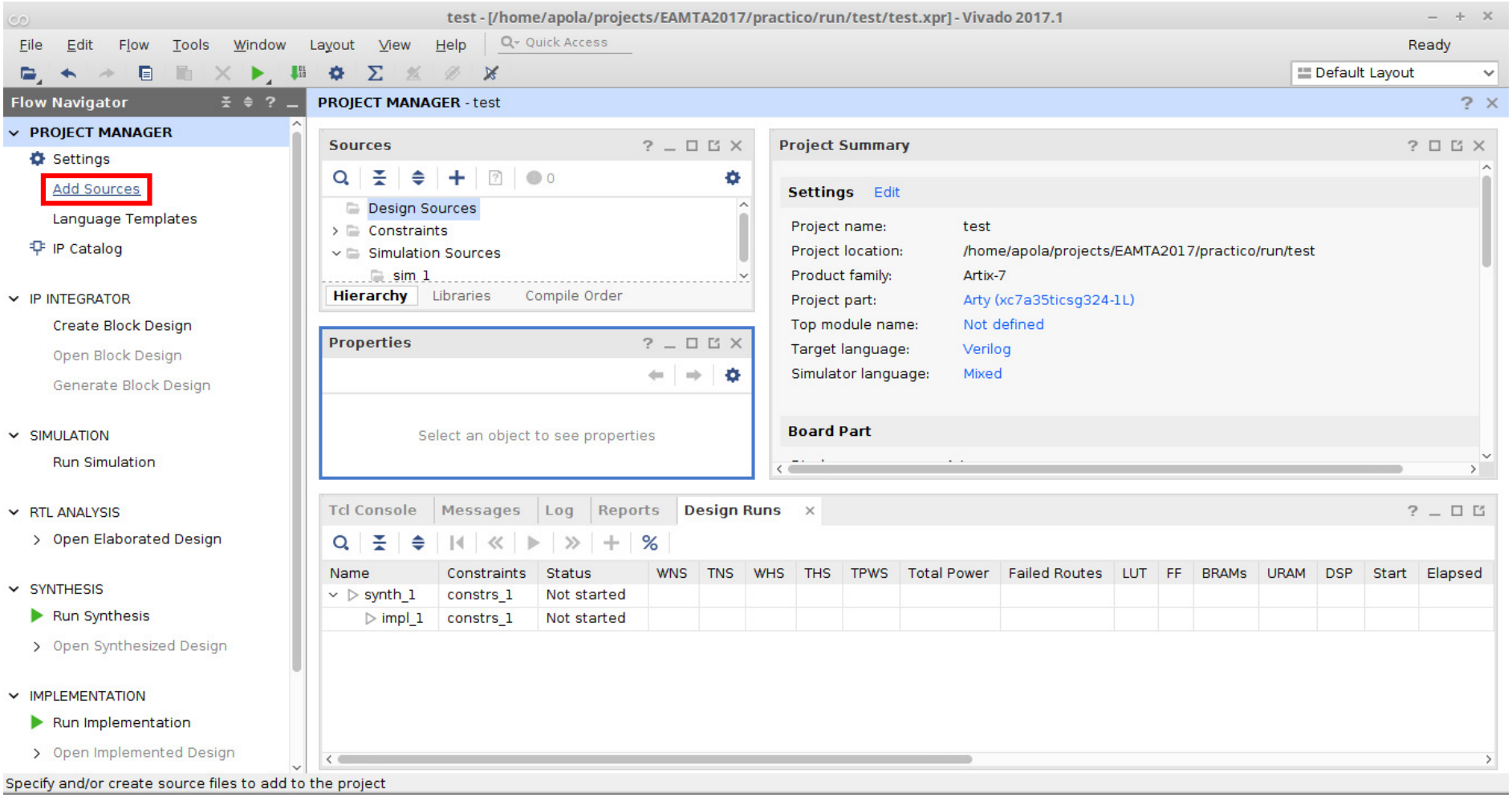
Vivado



Selecciona el archivo xdc

Herramienta Vivado

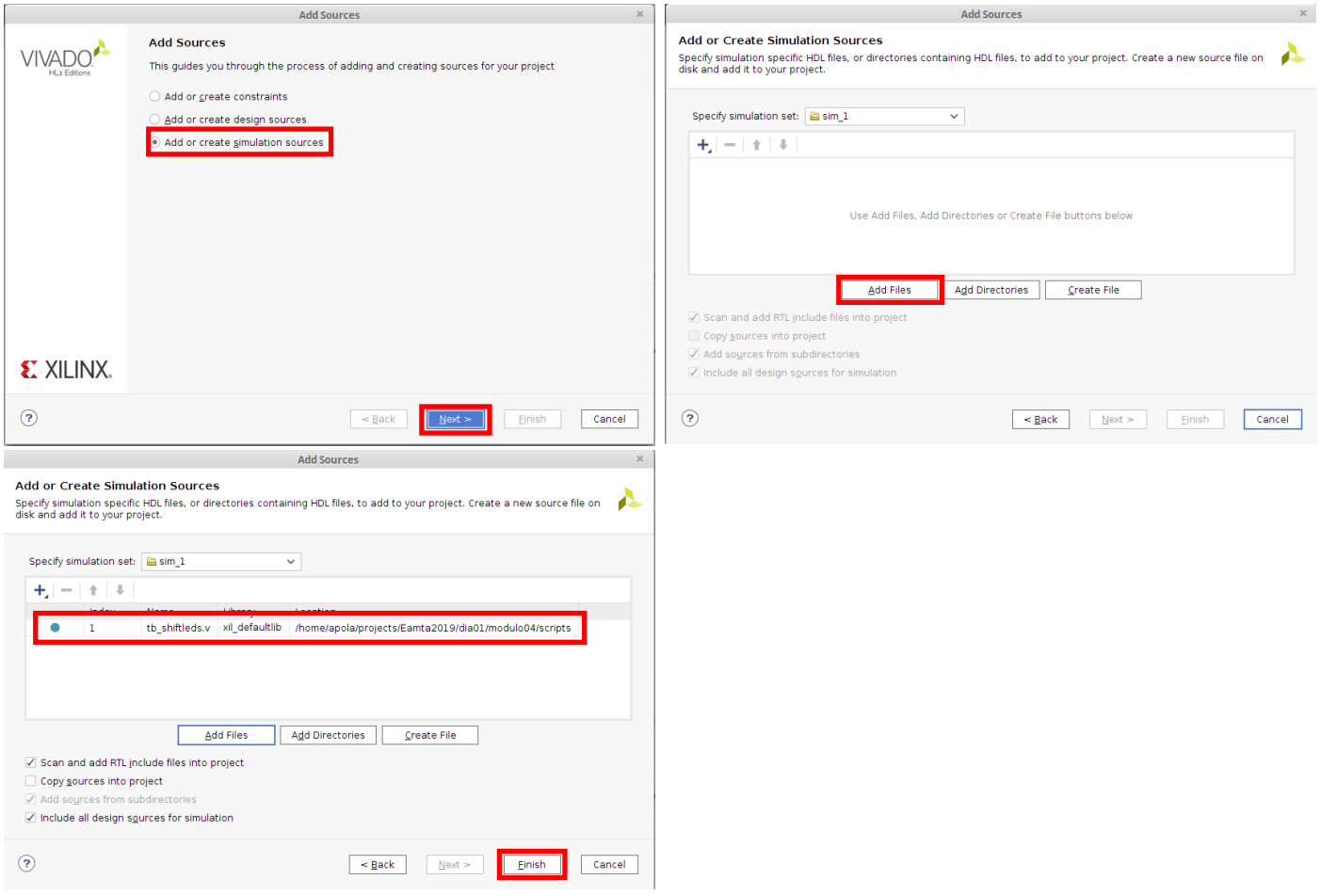
Vivado



Agrega nuevas fuentes a *Simulation Sources*

Herramienta Vivado

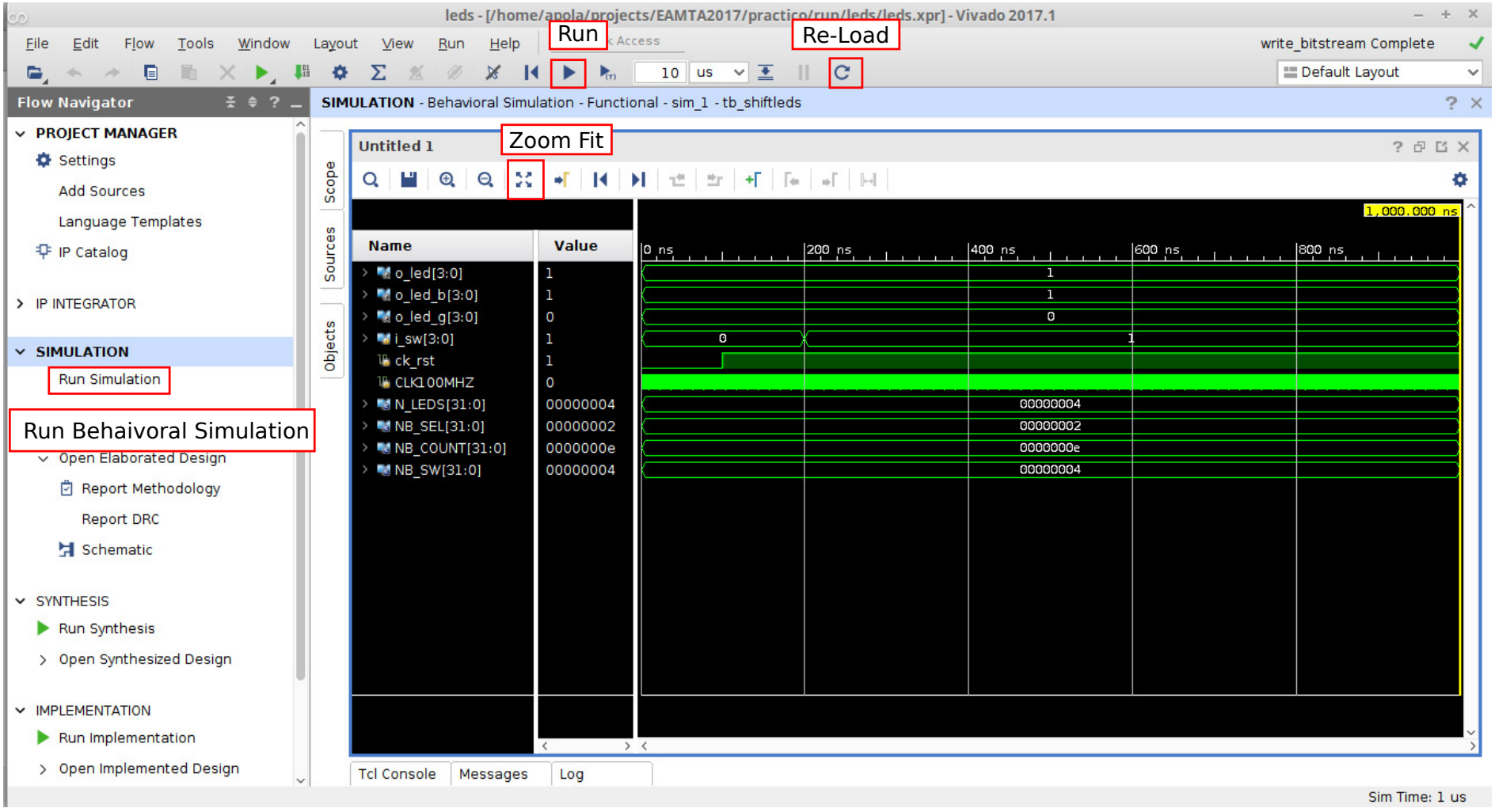
Vivado



Selecciona el archivo verilog para simulación

Herramienta Vivado

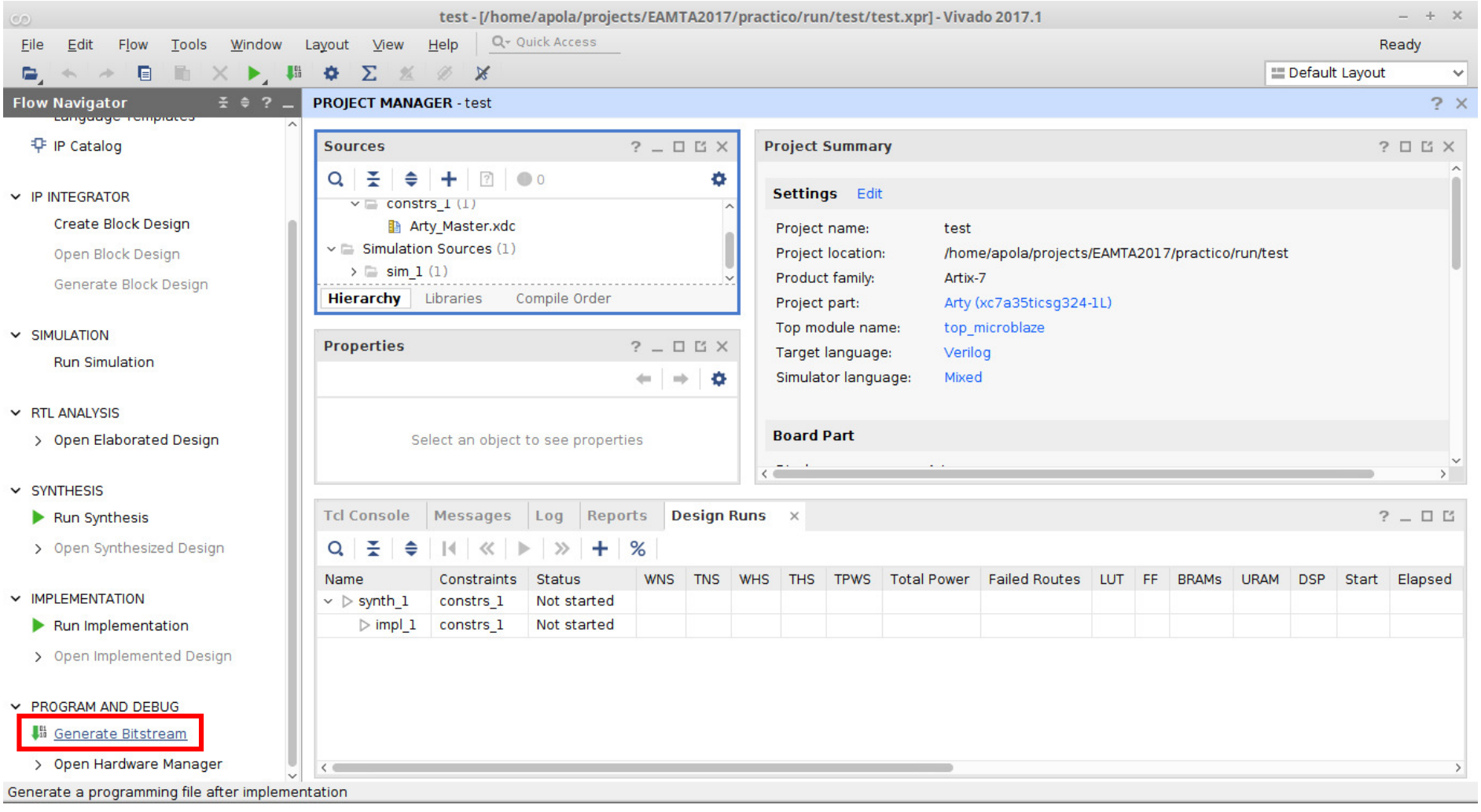
Run Simulation - Run Behavioral Simulation



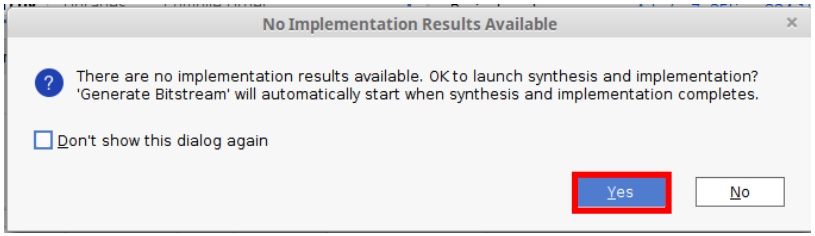
Simulando el comportamiento del diseño

Herramienta Vivado

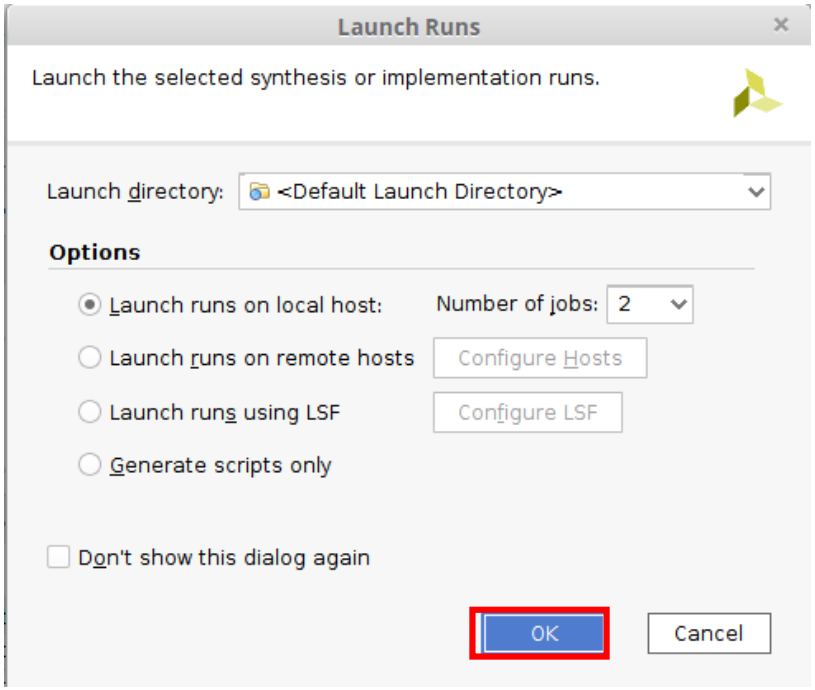
Vivado



Generar el *bitstream*



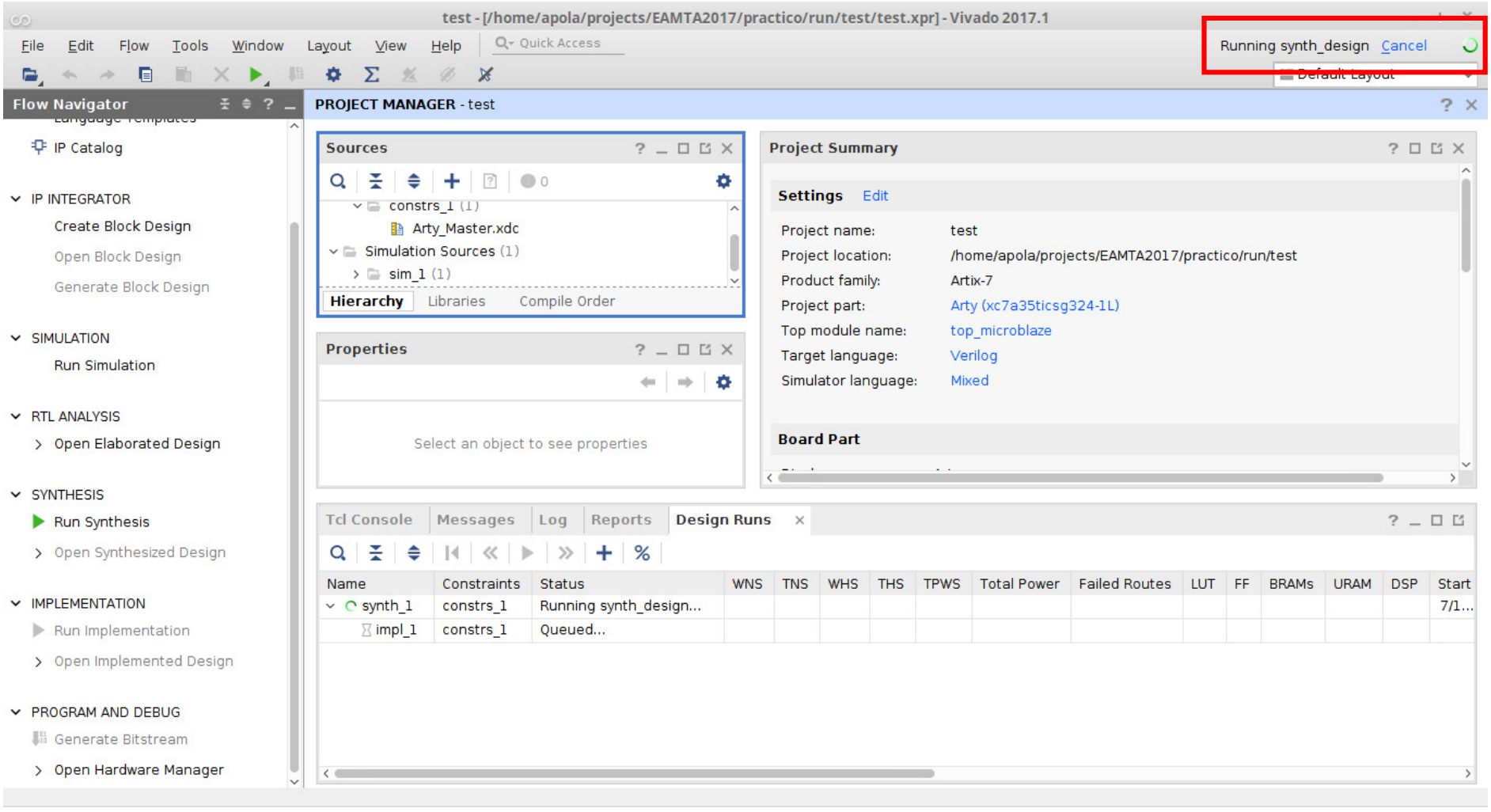
Generar el *bitstream*



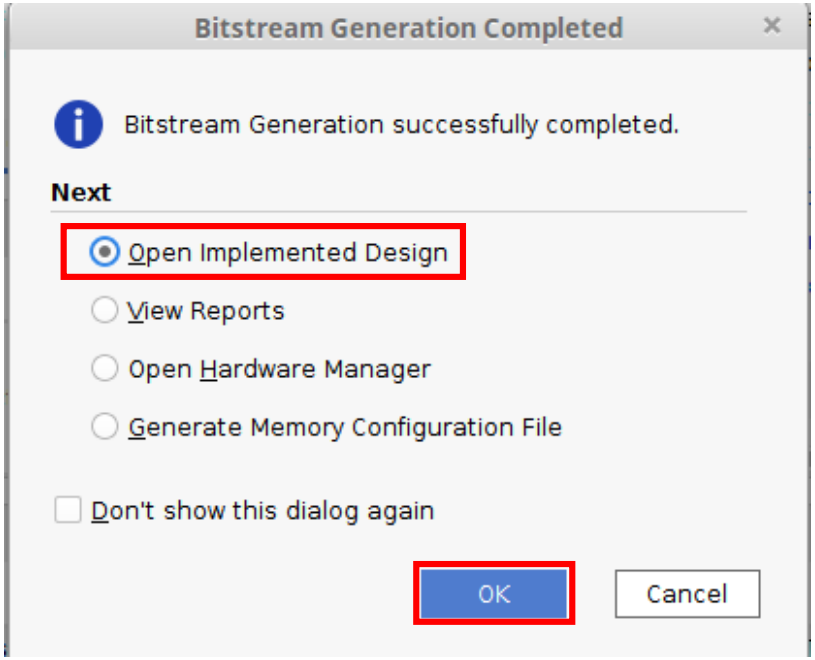
Generar el *bitstream*

Herramienta Vivado

Vivado



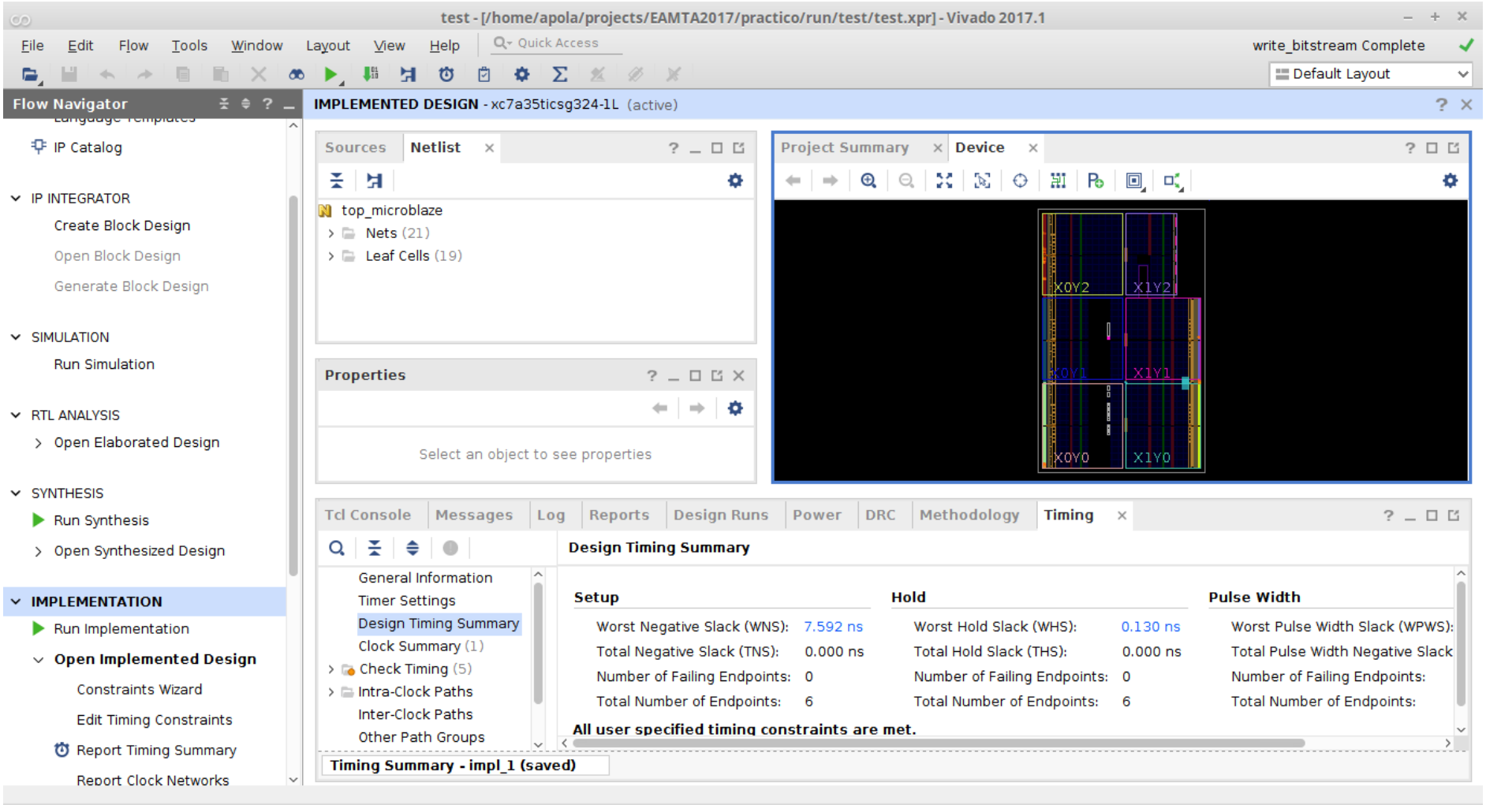
Ejecutando tareas de implementación



Abriendo el modelo implementado

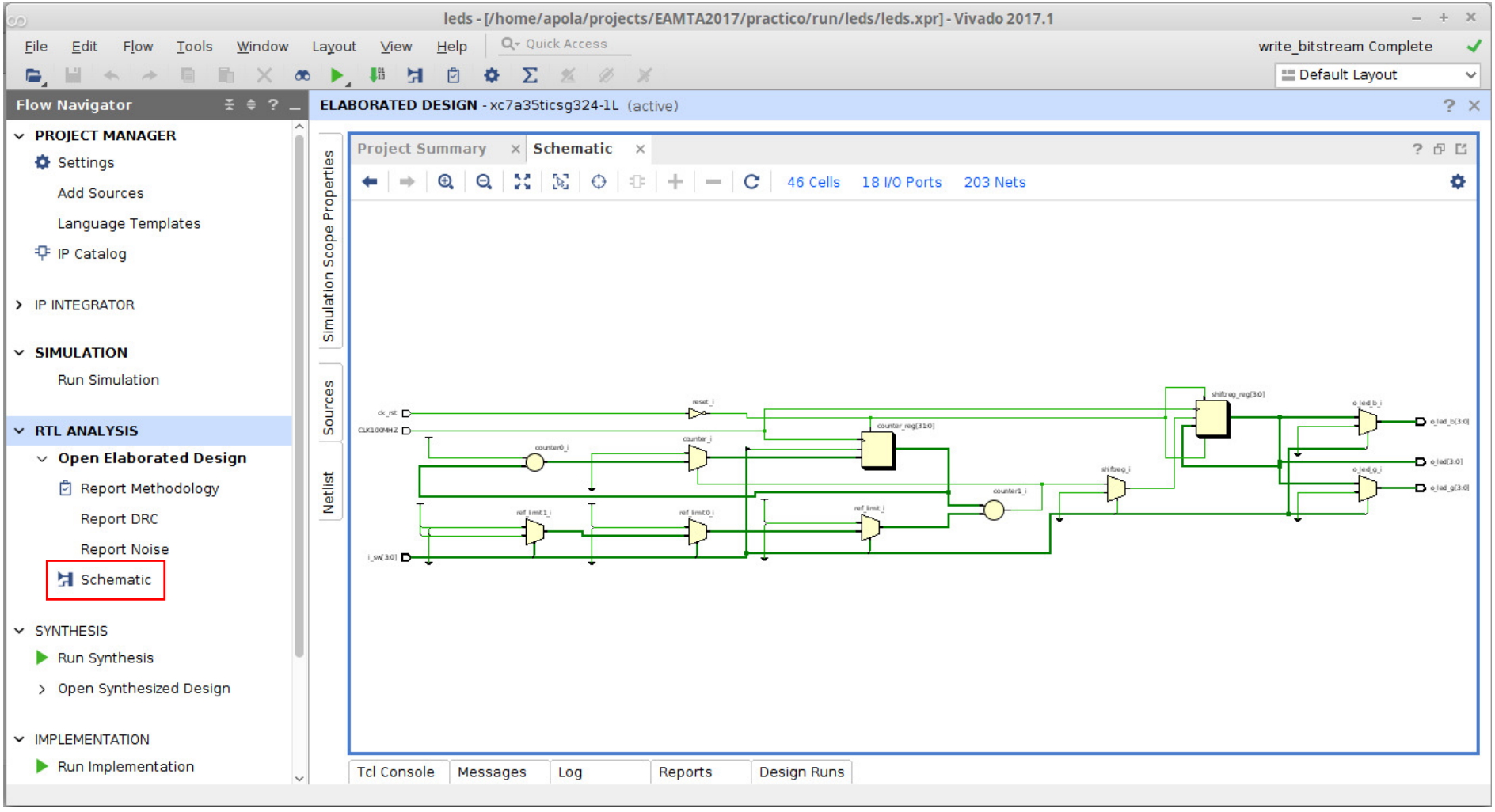
Herramienta Vivado

Vivado



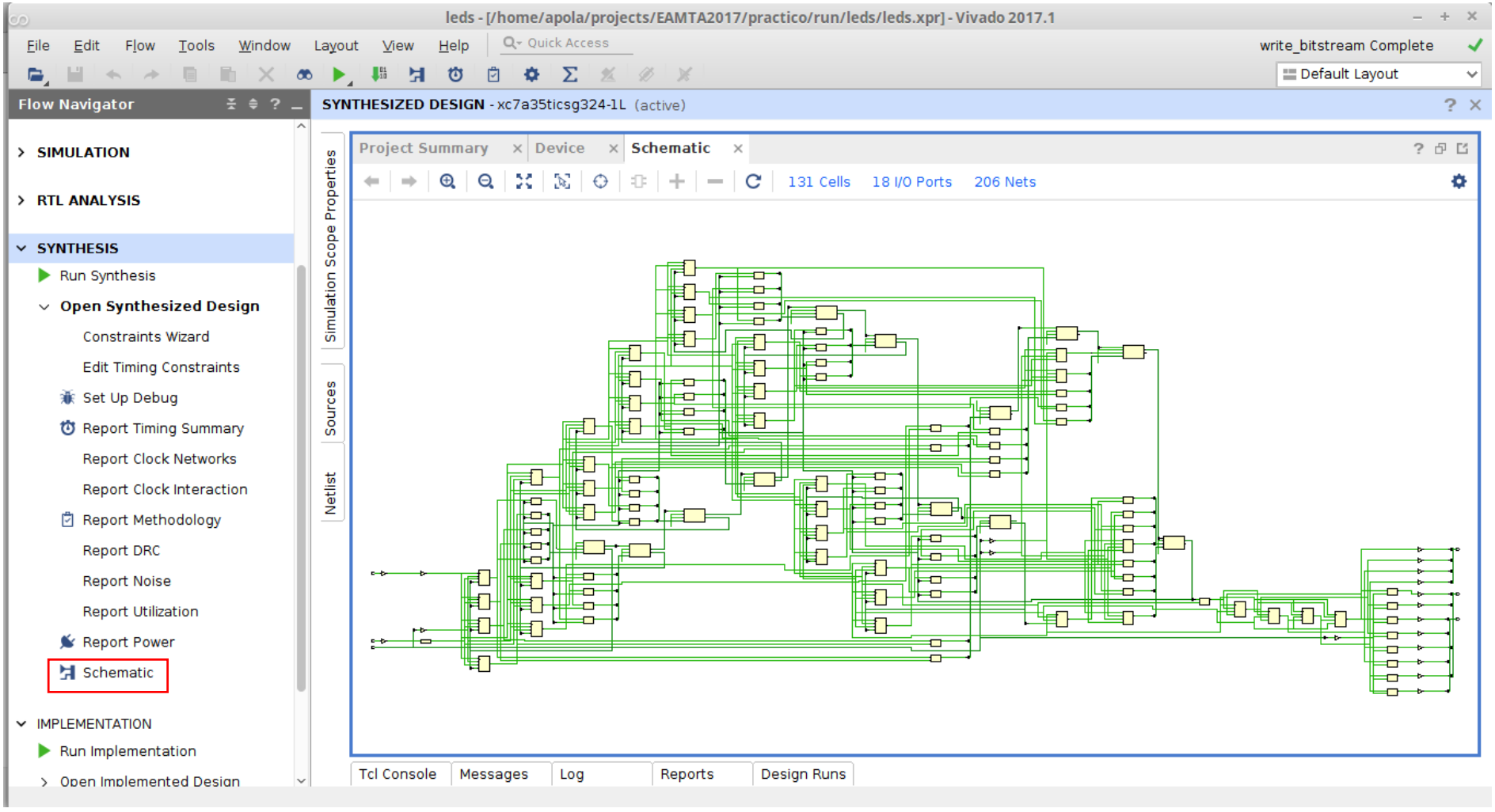
Diseño implementado

Herramienta Vivado
RTL Schematic



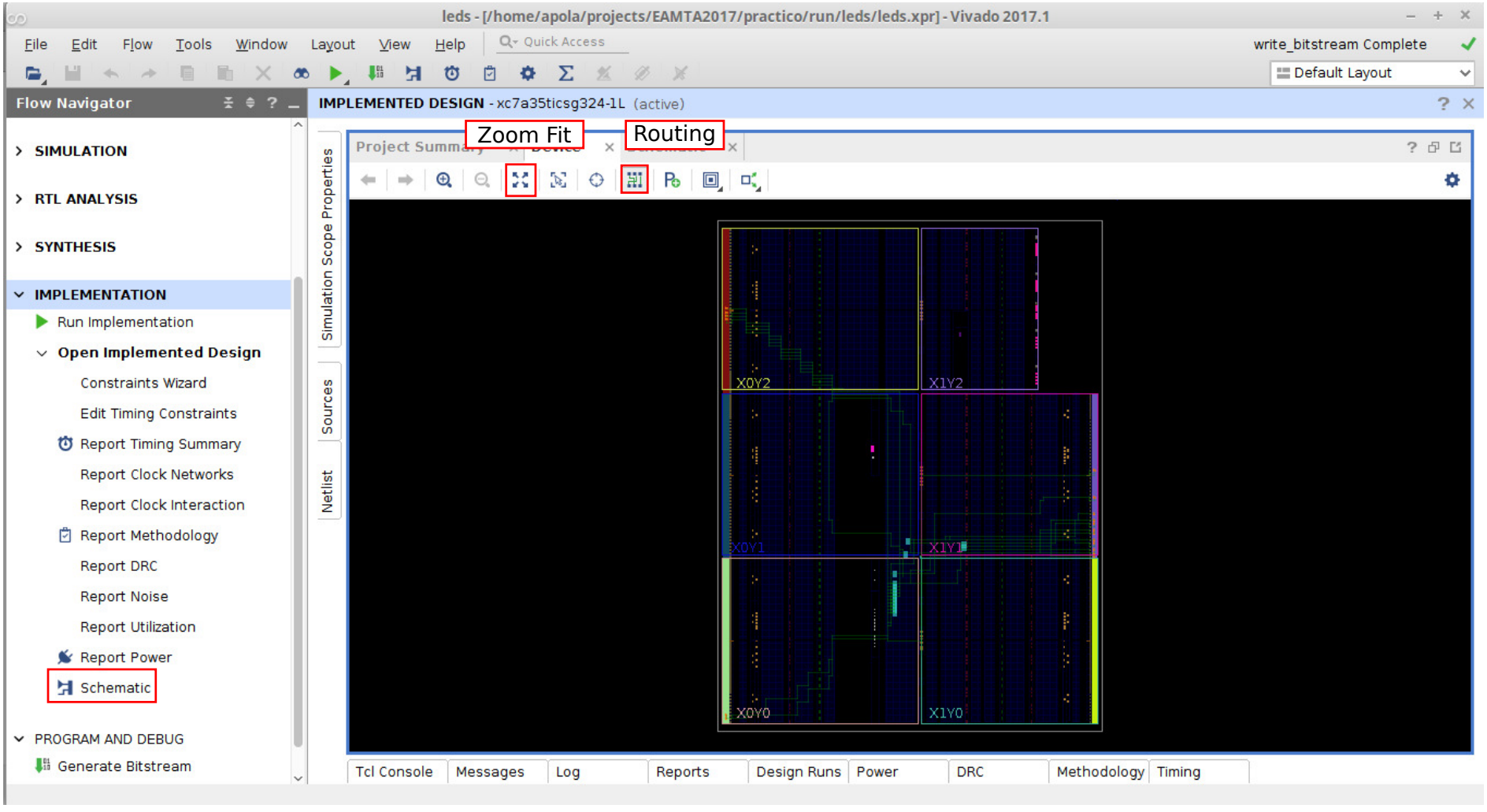
Esquemático RTL

Herramienta Vivado
Synthesis - Schematic



Esquemático Implementación

Herramienta Vivado
Implementation - Schematic



Implementation