**ÁREA ELECTRICIDAD Y ELECTRÓNICA**

**ELECTRÓNICA DIGITAL Y MICROCONTROLADORES**

**Profesor: Yeison Javier Montagut Ferizzola**

# TUTORIAL: CONTADOR DE CUATRO BITS EN FPGA

Este documento presenta una guía para la implementación de contadores en Basys 3.

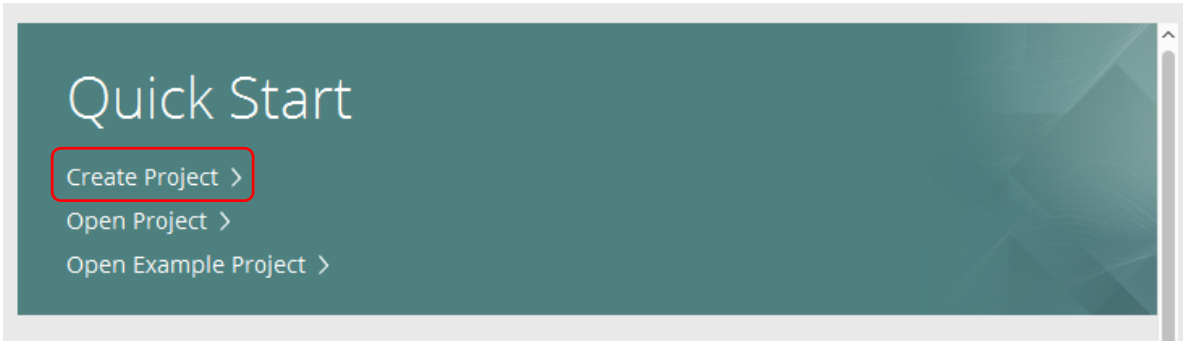
## PASOS PARA IMPLEMENTAR UN CONTADOR EN UNA FPGA

El sistema se divide en dos subsistemas: Reloj (divisor de frecuencia) y contador propiamente dicho. El sistema de desarrollo Basys 3 tiene un reloj, pero este es de una frecuencia muy alta (100MHz) y por lo tanto se hace necesario reducir esa frecuencia (por medio de un divisor de frecuencia) a una frecuencia que sea manejable, por ejemplo, en este caso 1 Hz, por lo tanto, el primer subsistema se refiere al divisor de frecuencia y el segundo subsistema es el contador propiamente dicha.

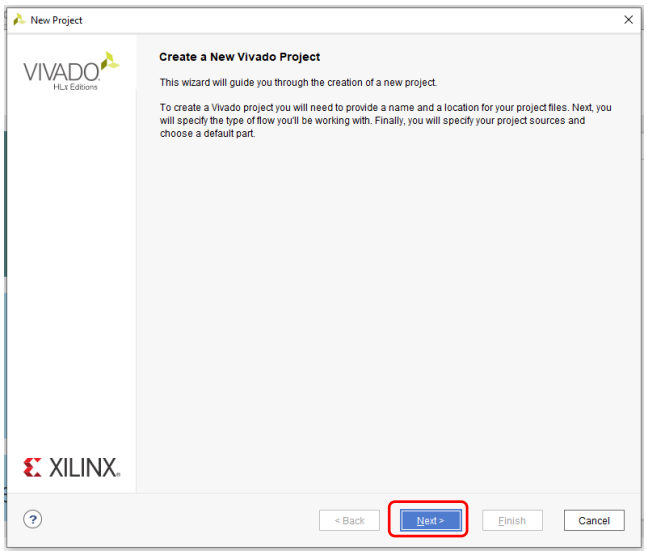
### SUBSISTEMA RELOJ

1. Iniciar el programa Vivado, dando doble clic en el icono

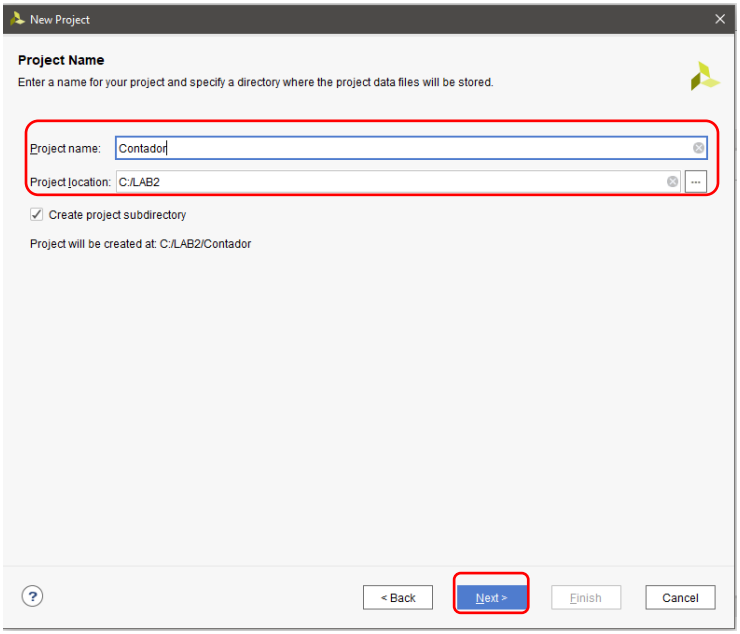

2. Clic en "*Create Project*"



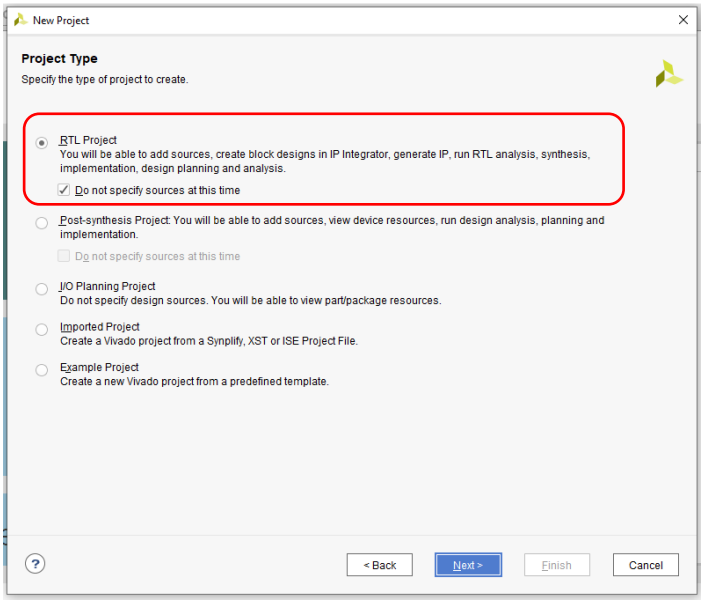

3. Se abre la siguiente ventana y clic en "*Next*"



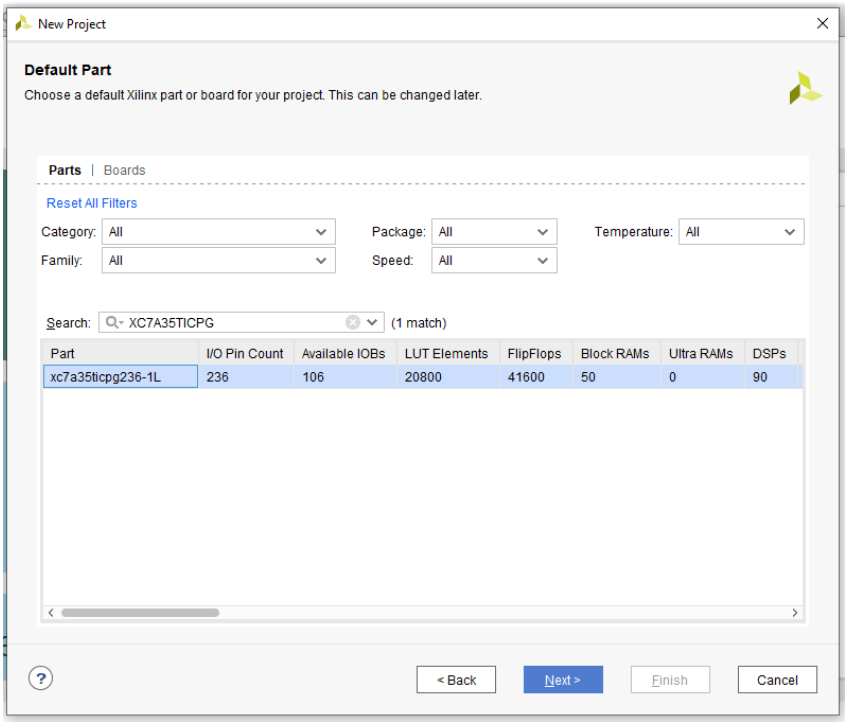

4. Dar un nombre y una ubicación al proyecto, evitar poner punto (.) en el nombre del proyecto. La ubicación debe ser fácil de encontrar preferiblemente en una carpeta del directorio raíz (C://). Clic en "*Next*"



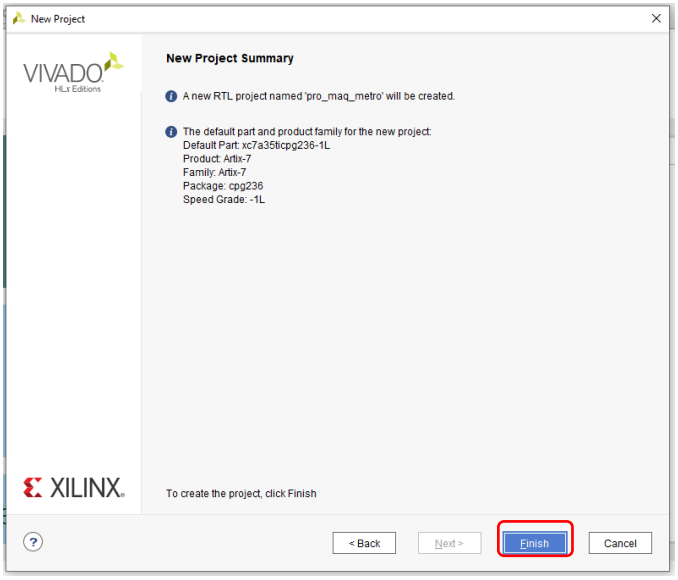

5. Seleccionar "*RTL Project*"



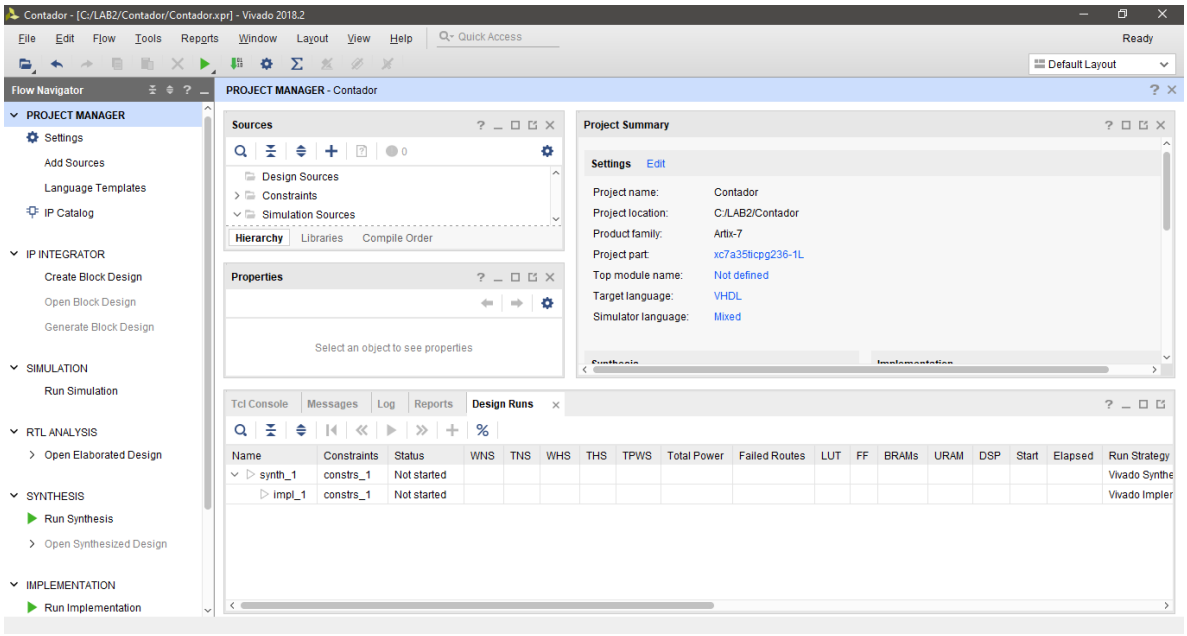

6. Seleccionar la FPGA con la que se va a trabajar, en este caso: xc7a35ticpg236-1L, clic en "*Next*"



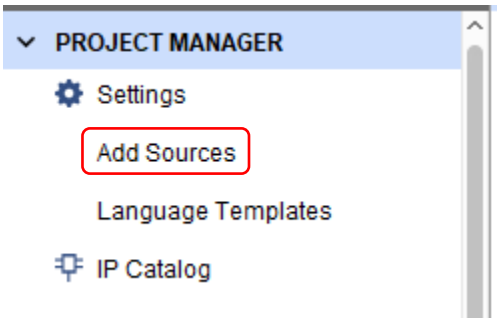

7. Aparece la siguiente ventana que indica que el proyecto se ha creado. Clic en "*Finish*"



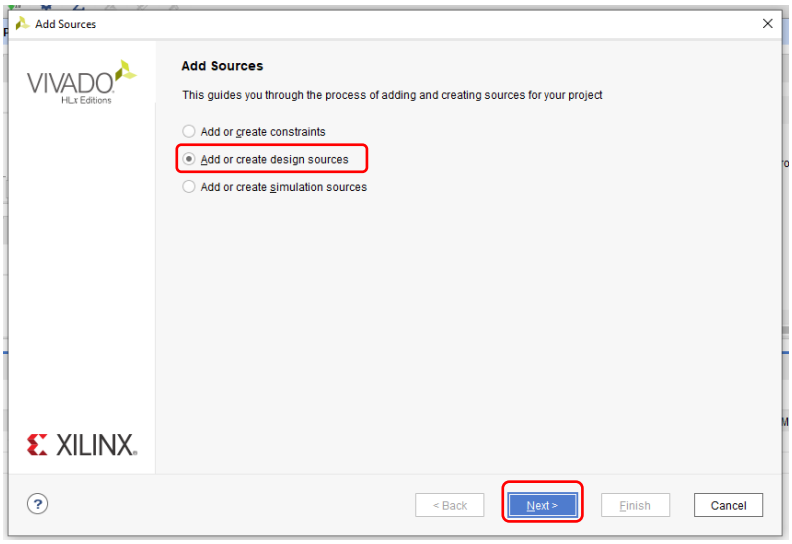

8. Se abre el espacio de trabajo de Vivado



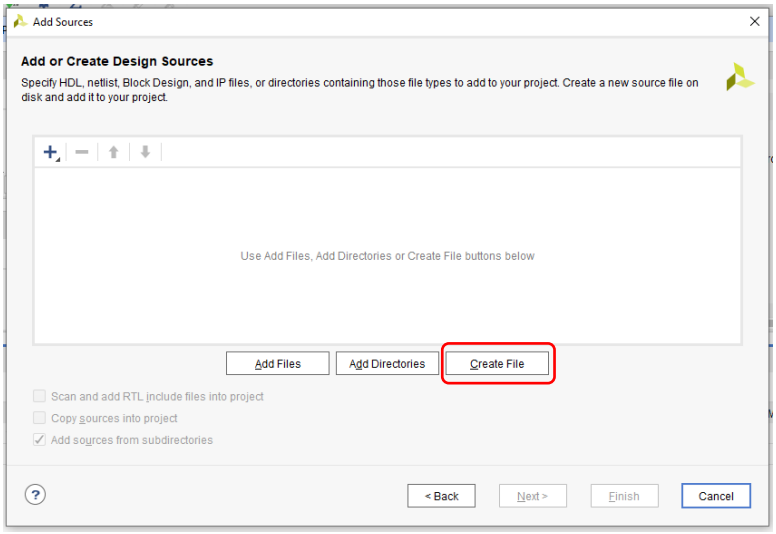

9. A continuación, se agrega el archivo vhdl que permitirá generar el divisor de frecuencia, para ello clic en "*Add Sources*"



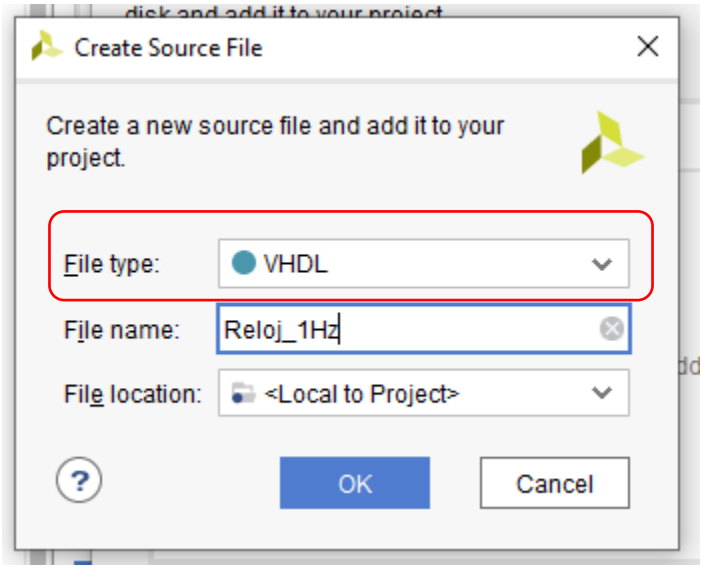

10. Aparece la siguiente ventana, en ella, seleccionar: "*Add or create design sources*" y clic en "*Next*"



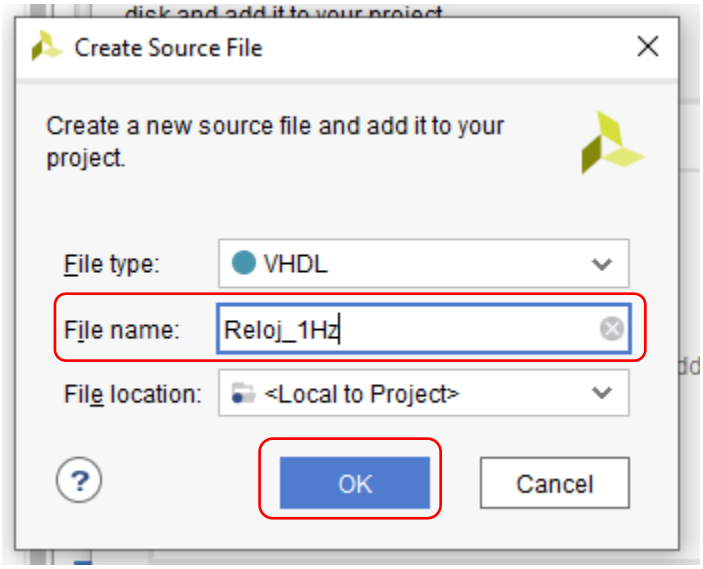

11. En la siguiente ventana clic en: "*Create Files*"



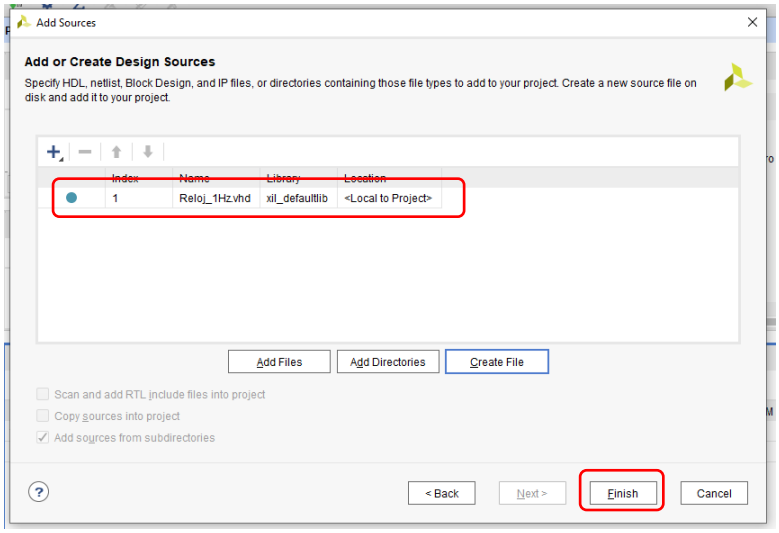

12. En la nueva ventana se selección "VHDL" en *File Type*



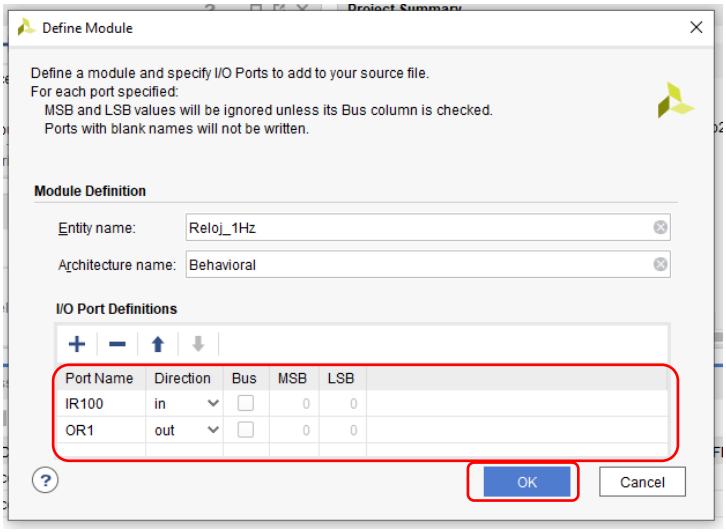
13. Se le asigna un nombre al archivo en *File name* y clic en "*Ok*"



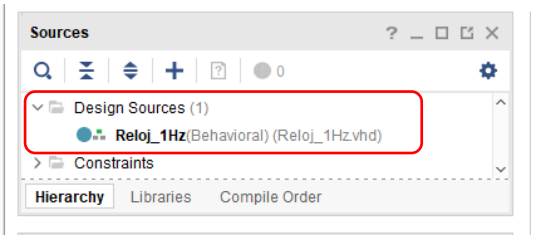

14. Revisar que aparezca el nuevo archivo creado y clic en "*Finish*"



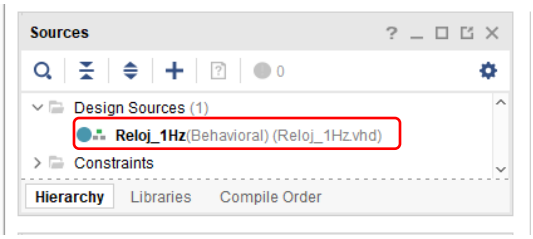

15. A continuación, crear las entradas y salidas que va a tener el divisor de frecuencia, en este caso una entrada tipo bit "IR100" y una salida tipo bit "OR1" y clic en "*Ok*"



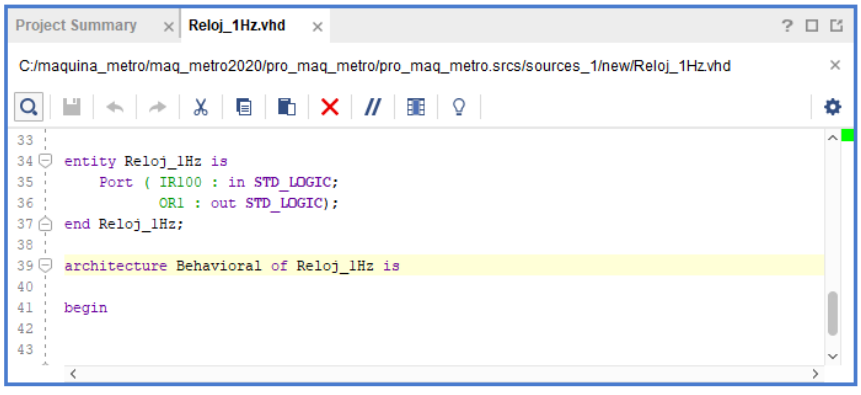

16. El nuevo archivo creado debe aparecer en:




17. Doble clic en el archivo creado




18. Se abre el archivo creado




19. En el archivo creado escribir el siguiente código:

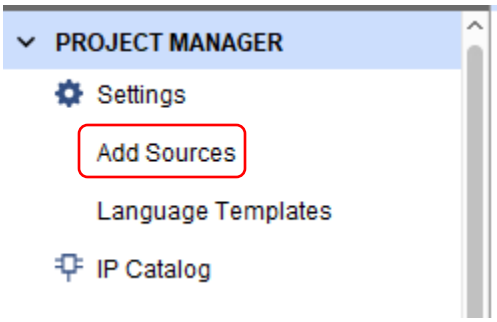
```
22  library IEEE;
23  use IEEE.STD_LOGIC_1164.ALL;
24  use IEEE.NUMERIC_STD.ALL;
25
26  -- Uncomment the following library declaration if using
27  -- arithmetic functions with Signed or Unsigned values
28  --use IEEE.NUMERIC_STD.ALL;
29
30  -- Uncomment the following library declaration if instantiating
31  -- any Xilinx leaf cells in this code.
32  --library UNISIM;
33  --use UNISIM.VComponents.all;
34
35  entity Reloj_1Hz is
36      Port ( IR100 : in STD_LOGIC;
37             OR1 : out STD_LOGIC);
38  end Reloj_1Hz;
39
40  architecture Behavioral of Reloj_1Hz is
41   signal aux : integer range 0 to 100000000 := 0;
42  signal x : STD_LOGIC;
43  begin
44  process (IR100)
45          begin
46              if rising_edge (IR100) then
47                  aux <= aux + 1;
48
49                  if (aux = 49999999) then
50                      x <= NOT x;
51                      aux <= 0;
52                 end if;
53              end if;
54     end process;
55  OR1 <= x;
56  end Behavioral;
```

20. Clic en guardar

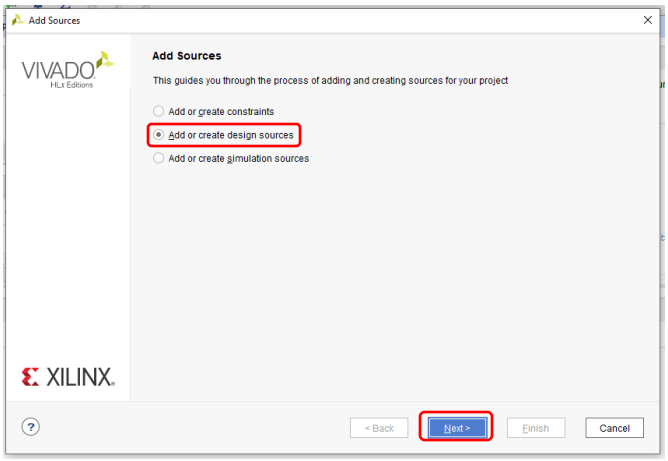
**SUBSISTEMA CONTADOR**

A continuación, se debe crear el archivo VHDL del contador

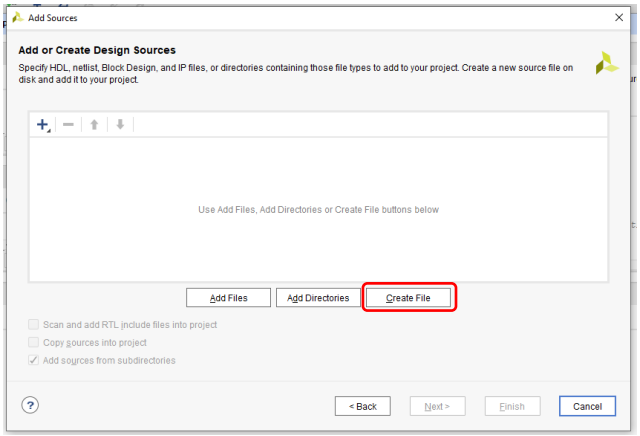
Clic en "*Add Sources*"



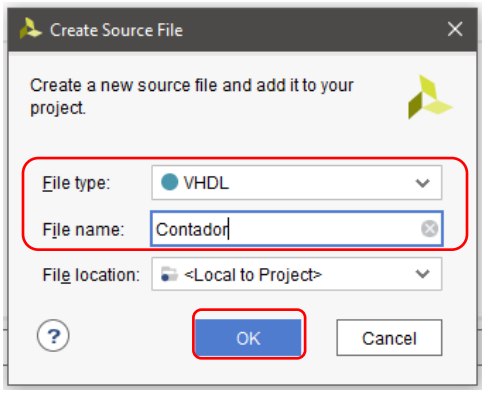

21. Seleccionar "*Add or create design sources*" y clic en "*Next*"



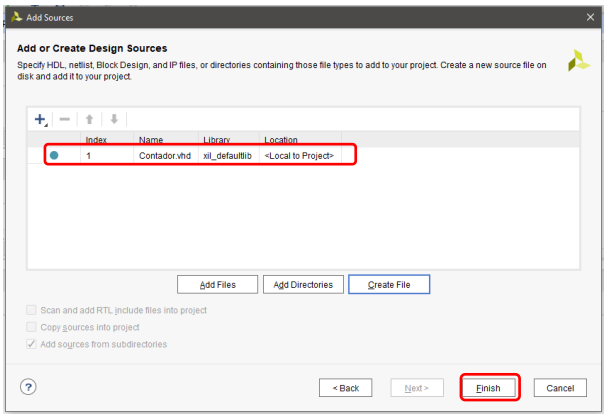

22. Clic en "*Create File*"



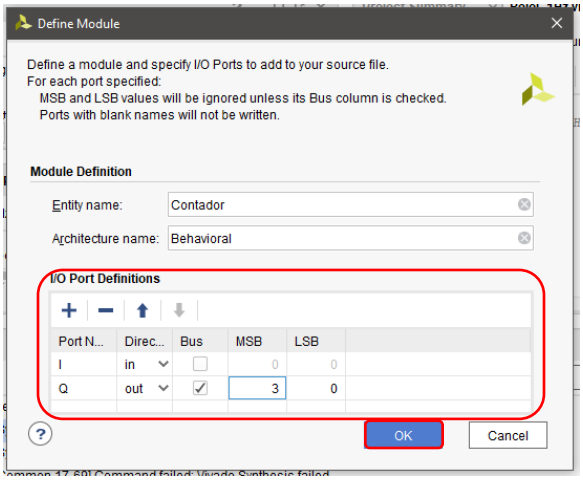

23. Seleccionar archivo tipo "VHDL", asignar un nombre al nuevo archivo y clic en "*OK*"



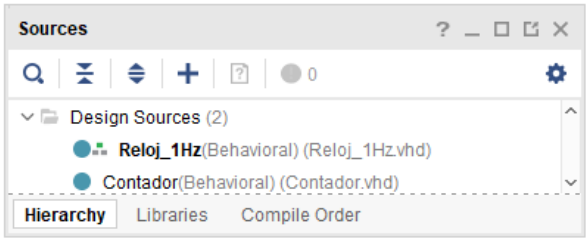
24. El archivo creado deberá aparecer en la siguiente ventana y clic en "*Finish*"




25. Agregar las entradas y salidas del contador, en este caso una entrada que es la señal de reloj y cuatro salidas. Clic en "*OK*"




26. El nuevo archivo creado aparece en




27. Doble clic en el nuevo archivo para crear el código en VHDL del contador de 4 bits

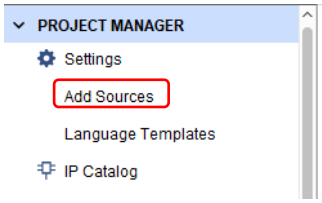
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
Use IEEE.NUMERIC_STD.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;
entity Contador is
    Port ( I : in STD_LOGIC;
           Q : out STD_LOGIC_VECTOR (3 downto 0));
end Contador;

architecture Behavioral of Contador is
signal cont_1: STD_LOGIC_VECTOR (3 downto 0) := "0000";
begin
    process (I)
        begin
            if rising_edge (I) then
                    cont_1 <= cont_1 + 1;
            end if;

        end process;

    Q <= cont_1;

end Behavioral;
```

28. Clic en guardar

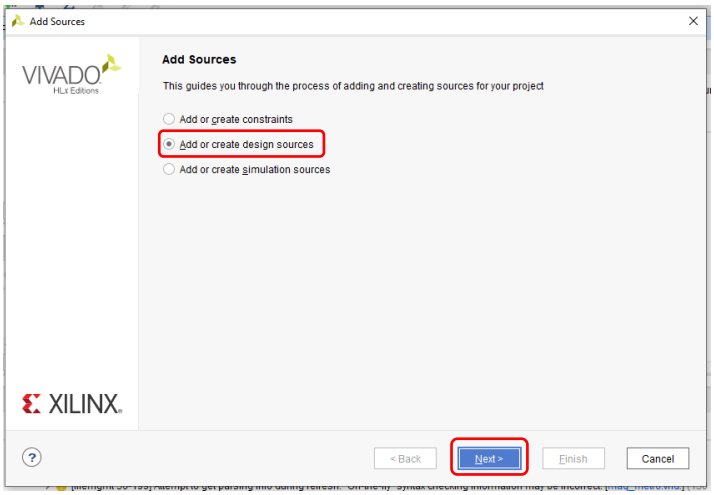
**SISTEMA GENERAL**

A continuación, se debe crear un archivo que contenga a todos los dos subsistemas creados, a ese archivo será nuestro archivo TOP.

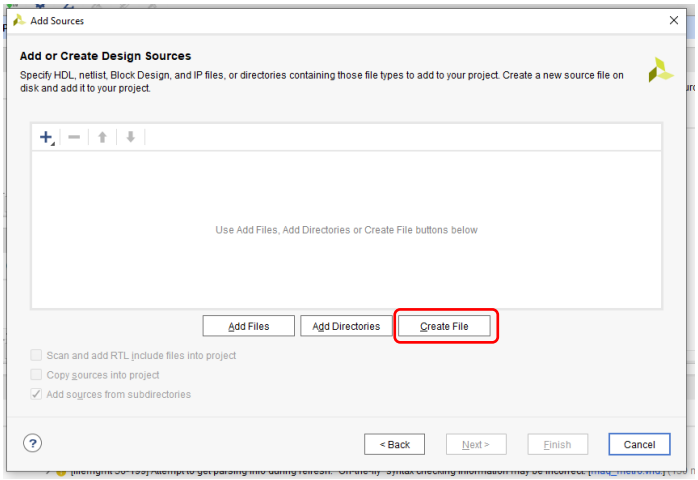
29. Clic en "*Add Sources*"



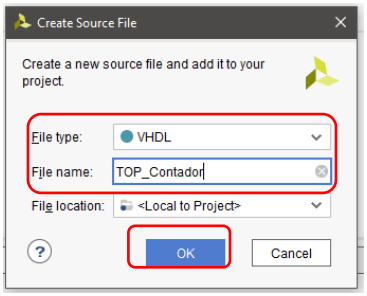

30. Seleccionar y clic en "*Next*"



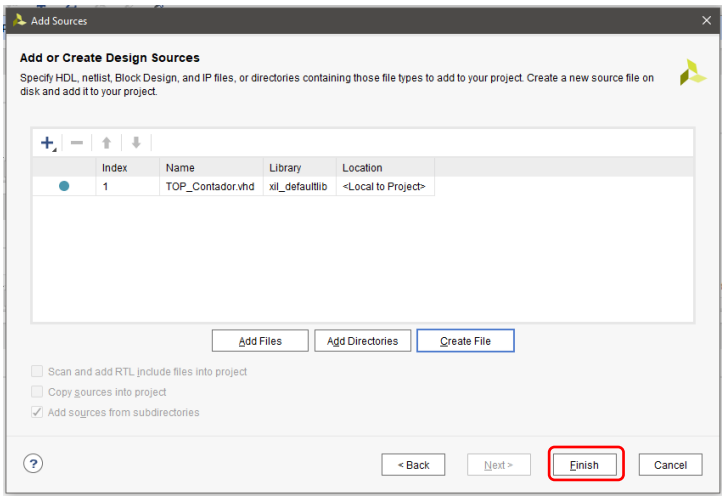

31. Clic en



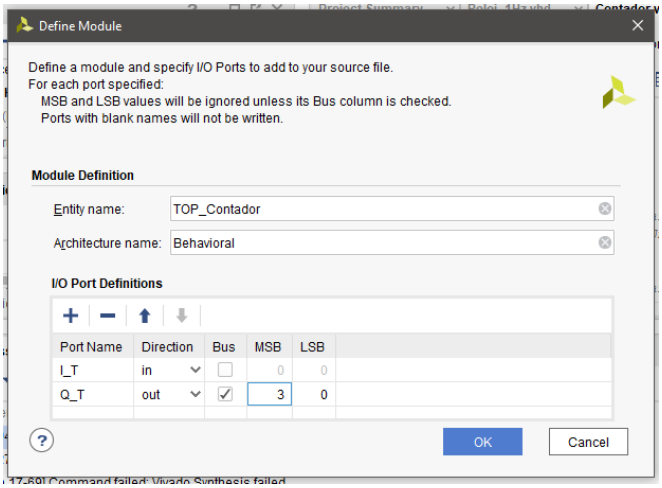

32. Nombre y tipo de archivo



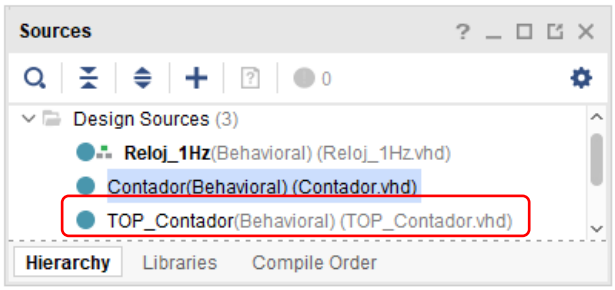

33. Clic en




34. Se crean las salidas y entradas del sistema que va a contener los otros dos subsistemas




35. Aparece un nuevo archivo en

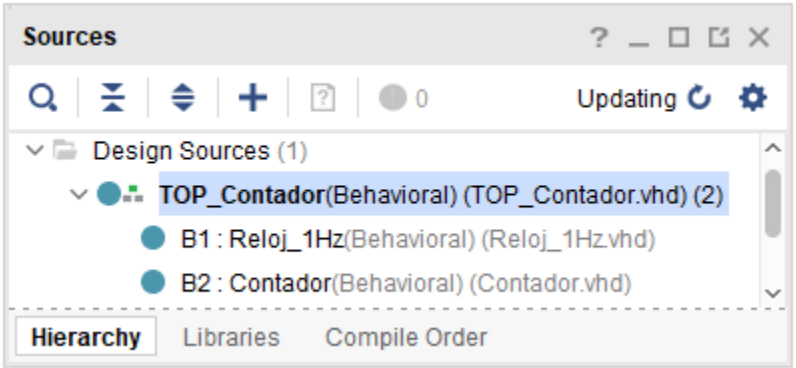



36. Doble clic en ese archivo
37. Se comienzan a agregar la entidad de los otros subsistemas como si fueran componentes del nuevo sistema

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
Use IEEE.NUMERIC_STD.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;
-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;
entity TOP_Contador is
    Port ( I_T : in STD_LOGIC;
           Q_T : out STD_LOGIC_VECTOR (3 downto 0));
end TOP_Contador;

architecture Behavioral of TOP_Contador is
component Reloj_1Hz is
    Port ( IR100 : in STD_LOGIC;
           OR1 : out STD_LOGIC);
end component;

component Contador is
    Port ( I : in STD_LOGIC;
           Q : out STD_LOGIC_VECTOR (3 downto 0));
end component;
signal aux1 : std_logic;
begin
B1: Reloj_1Hz PORT MAP(I_T,aux1);
B2: Contador PORT MAP(aux1,Q_T);

end Behavioral;
```

38. Clic en guardar

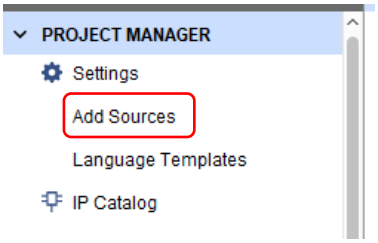
39. Observar el nuevo aspecto de



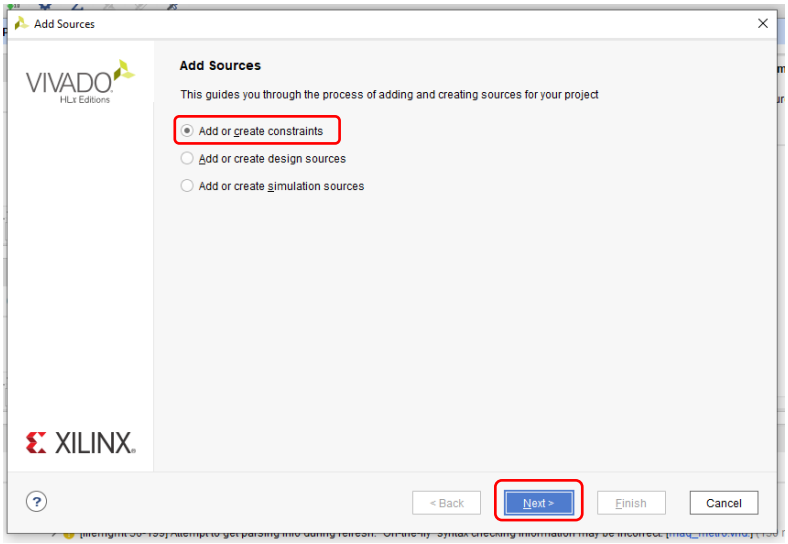
## CONEXIÓN DE LAS ENTRADAS Y SALIDAS DEL SISTEMA DESARROLLADO CON LAS ENTRADAS Y SALIDAD DE LA BASYS 3

Una vez creado el archivo de programación lo que viene a continuación es la conexión de las entradas y salida del sistema desarrollado con la señal de reloj y leds de la placa de desarrollo Basys 3

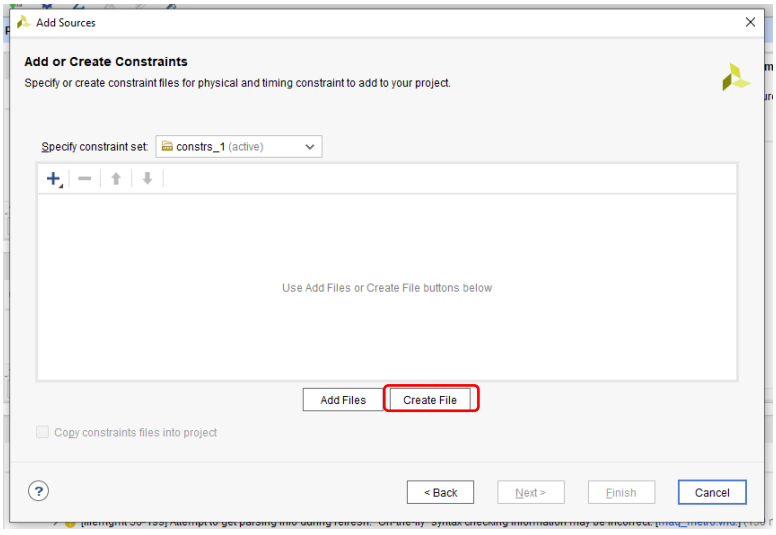
40. Clic en



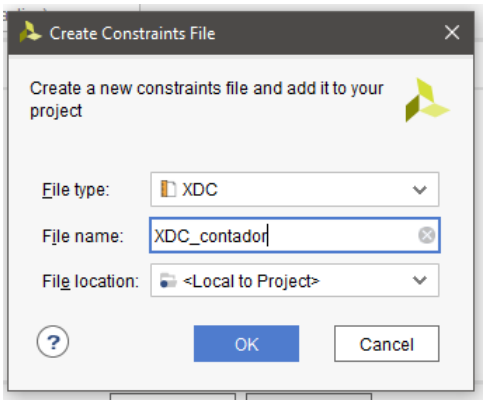

41. Seleccionar "*Add or create constraints*" y clic en "*Next*"



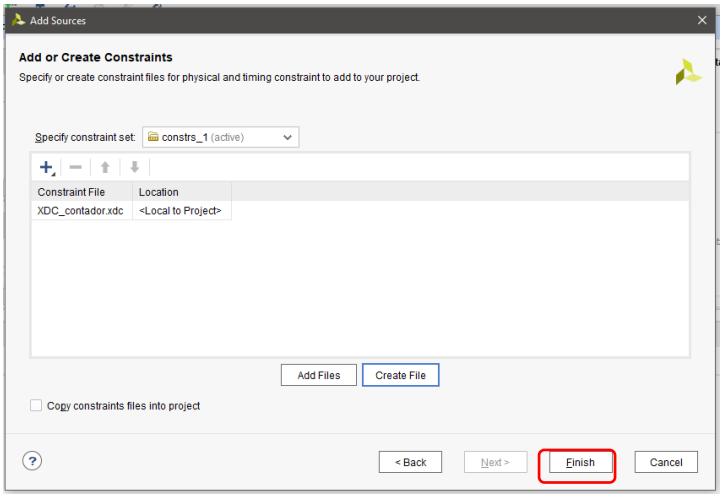

42. Clic en "*Create File*"



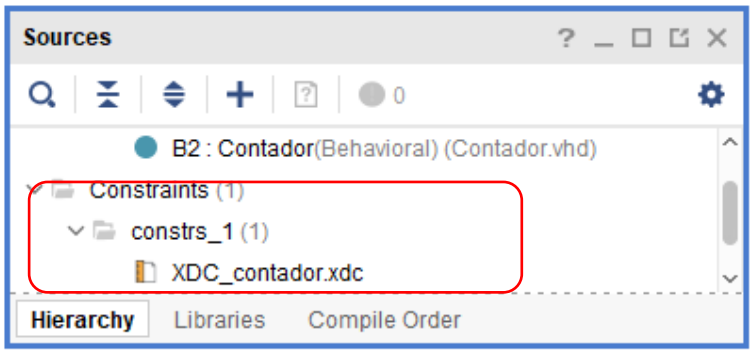

43. Se le asigna un nombre y clic en "*OK*"



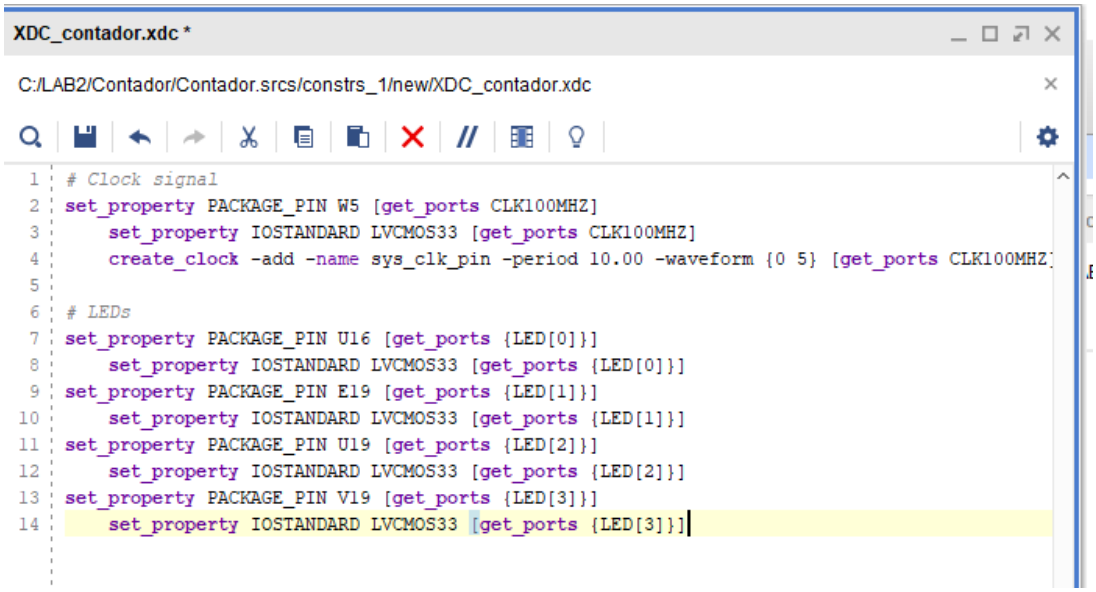

44. Clic en "*Finish*"



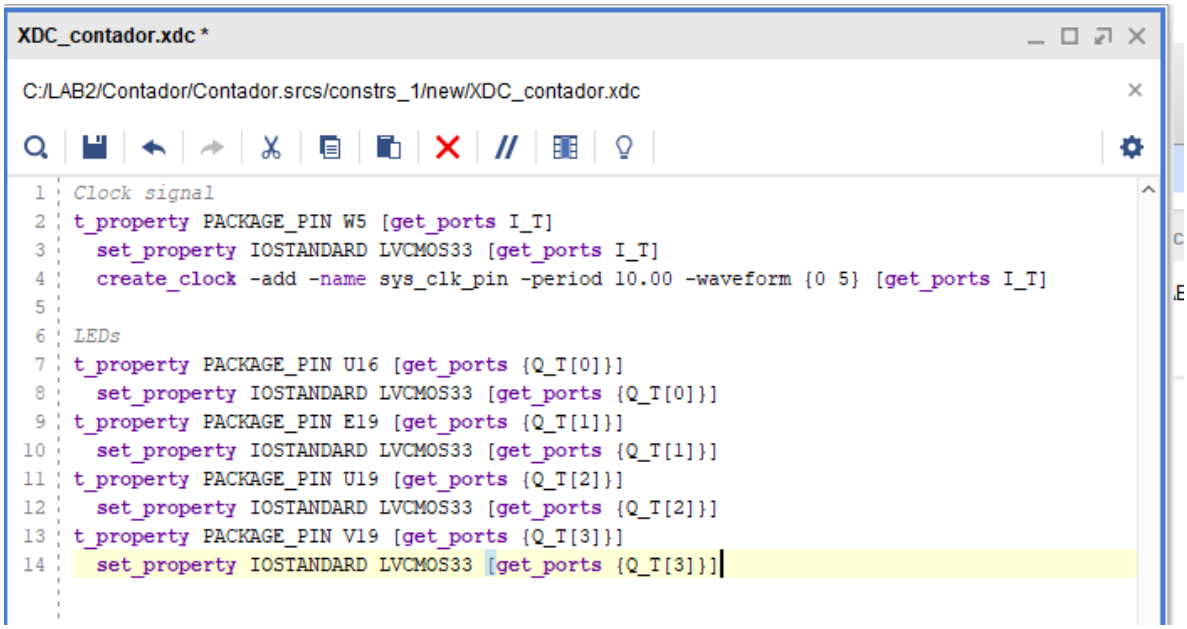

45. El nuevo archivo creado aparece en




46. Doble clic sobre el nuevo archivo creado y es posible editarlo
47. Para editar el archivo creado se debe ingresar a la siguiente dirección: https://github.com/Digilent/Basys3/blob/master/Projects/XADC_Demo/src/constraints/Basys3_Master.xdc
    Copiar las entradas, salidas y señal de reloj que se necesitan y pegarlos en el nuevo archivo creado.
48. El archivo creado deberá tener el siguiente aspecto:

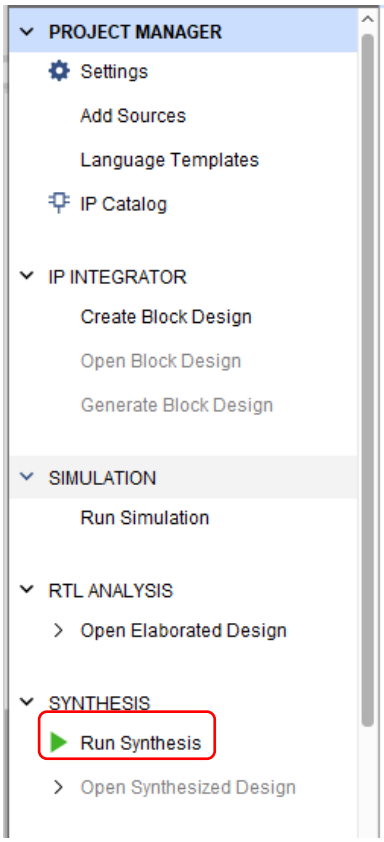



49. Reemplazar en el nuevo archivo creado el nombre de las entradas y salidas por los nombres usados en el proyecto

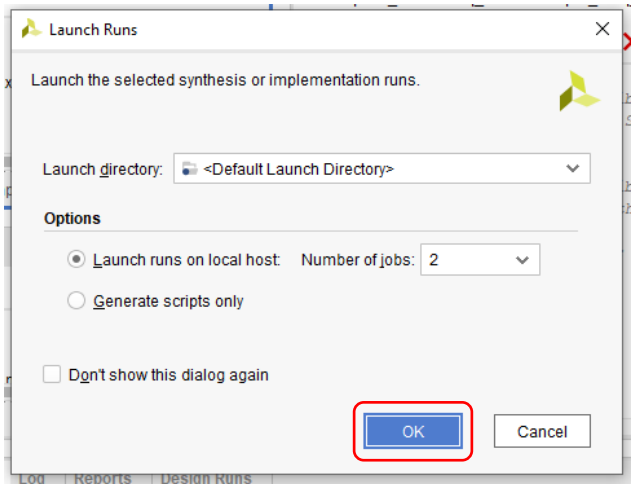



50. A continuación, clic en guardar

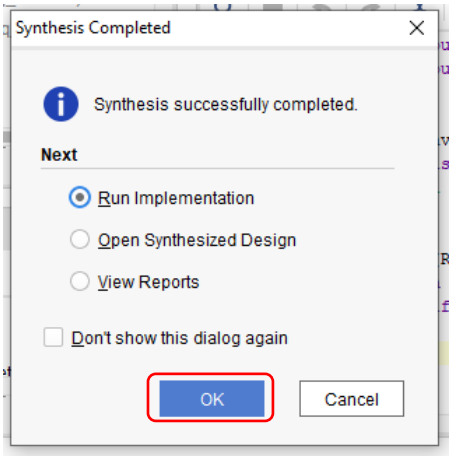
51. Y luego clic en "*Run Synthesis*"



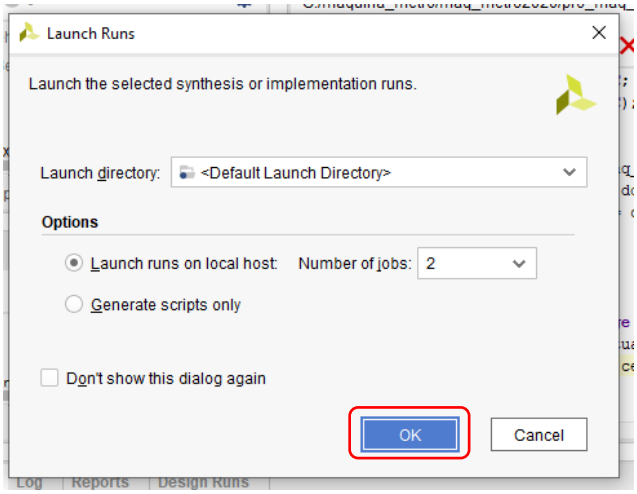

52. Clic en “OK”



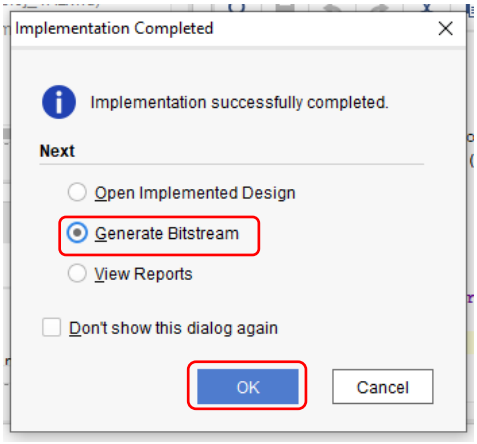

53. Esperar, una vez terminado clic en “OK”



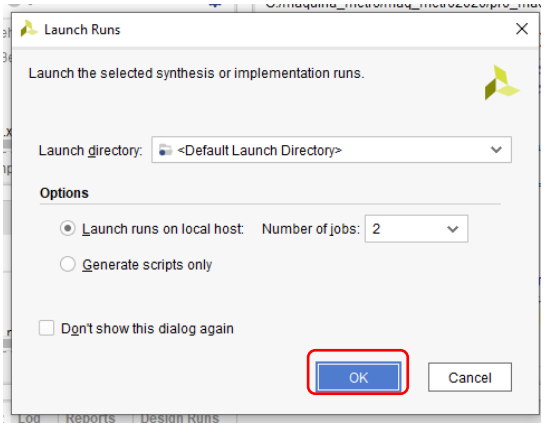

54. Clic en “OK”



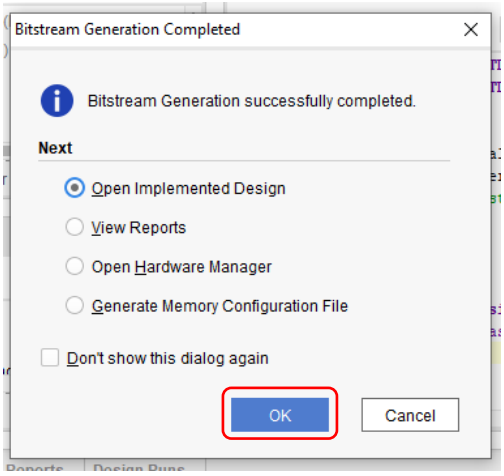

55. Seleccionar “Generate Bitstream” y clic en OK




56. Clic en “OK”



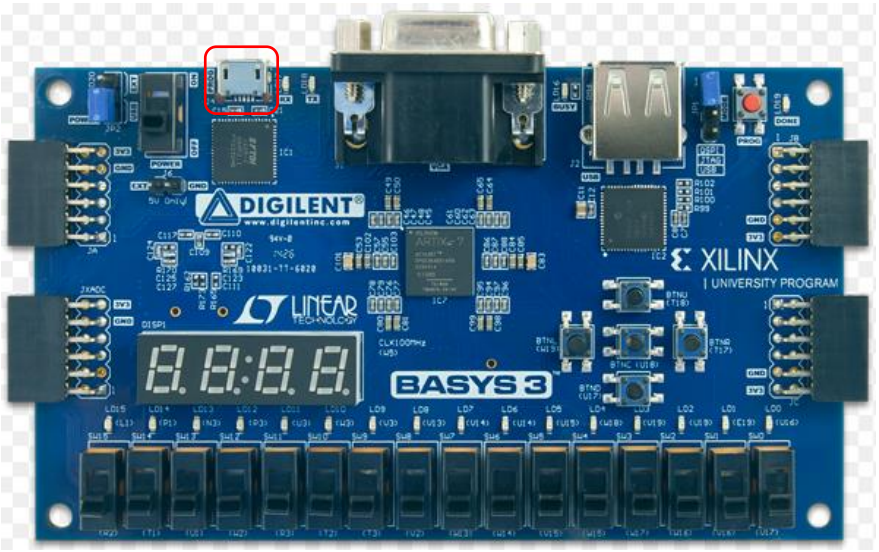

57. Clic en “OK”



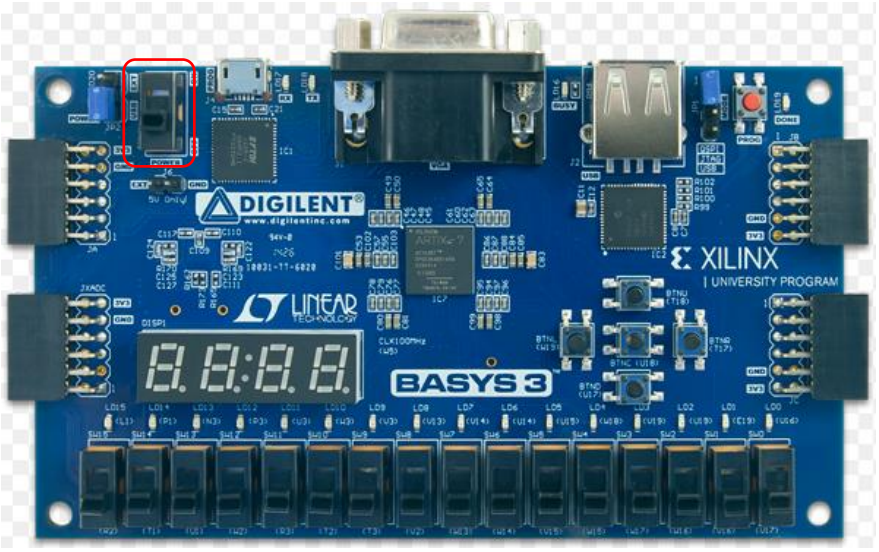

**PROGRAMACIÓN**

A continuación, se continua con la programación de la FPGA y comprobación de funcionamiento

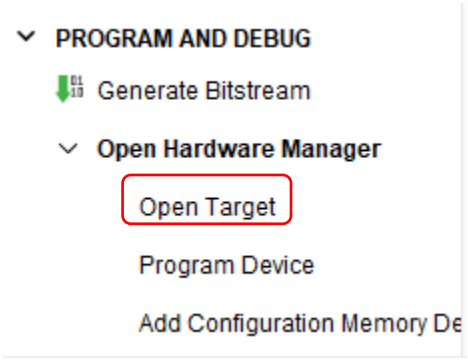
58. Conectar el sistema de desarrollo Basys 3 con el PC por medio de un cable USB. El cable USB se debe conectar a la Basys 3 por medio del conector señalado



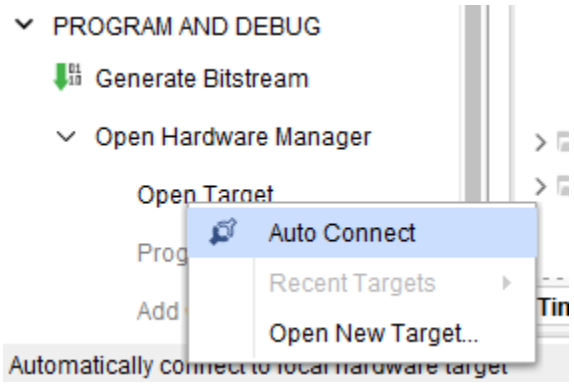

59. Encender el sistema de desarrollo Basys 3



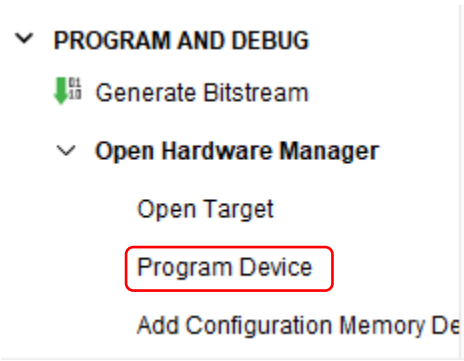

60. En “Open Hardware Manager” clic en “Open Target”



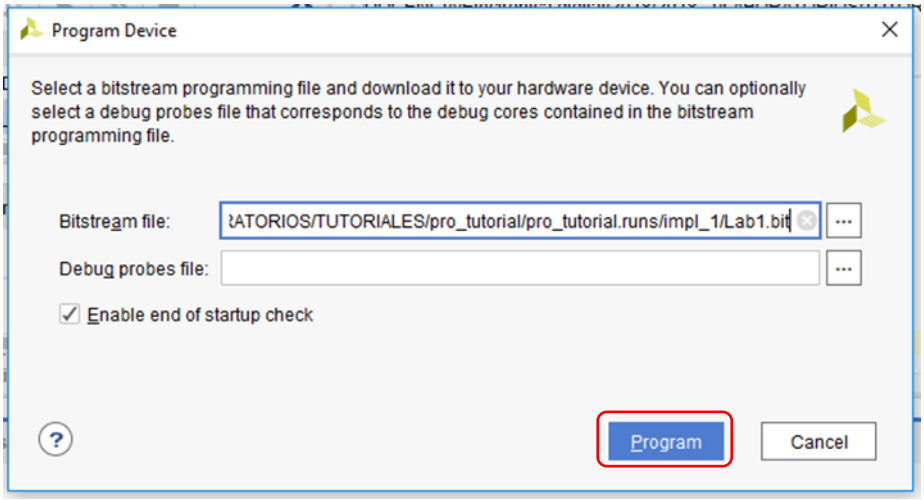

61. Clic en “*Auto Connect*”




62. Clic en “Program Device” y luego clic en la pestaña que se activa “xc7a35t_0”




63. En la ventana que se abre clic en “Program”




**COMPROBAR FUNCIONAMIENTO**