

TUTORIAL - VIVADO Y BASYS 3: DESARROLLO DE APLICACIONES EN FPGA

Yeison Javier Montagut Ferizzola



ÁREA ELECTRICIDA Y ELECTRÓNICA

TUTORIAL VIVADO Y BASYS 3: DESARROLLO DE APLICACIONES EN FPGA



Vivado Design Suite es la herramienta de Xilinx para el desarrollo de aplicaciones basadas en FPGA.

En este tutorial se mostrarán las herramientas básicas que el estudiante de la asignatura Electrónica Digital y Microcontroladores de la Universidad EIA requieren para implementar aplicaciones básicas basadas en FPGA usando Vivado y el sistema de desarrollo BASYS 3.

SISTEMAS DIGITALES COMBINACIONALES

Ejemplo:

Comprobar la tabla de verdad de las compuertas AND, OR, NOT, NAND, NOR, XOR XNOR.

Solución:

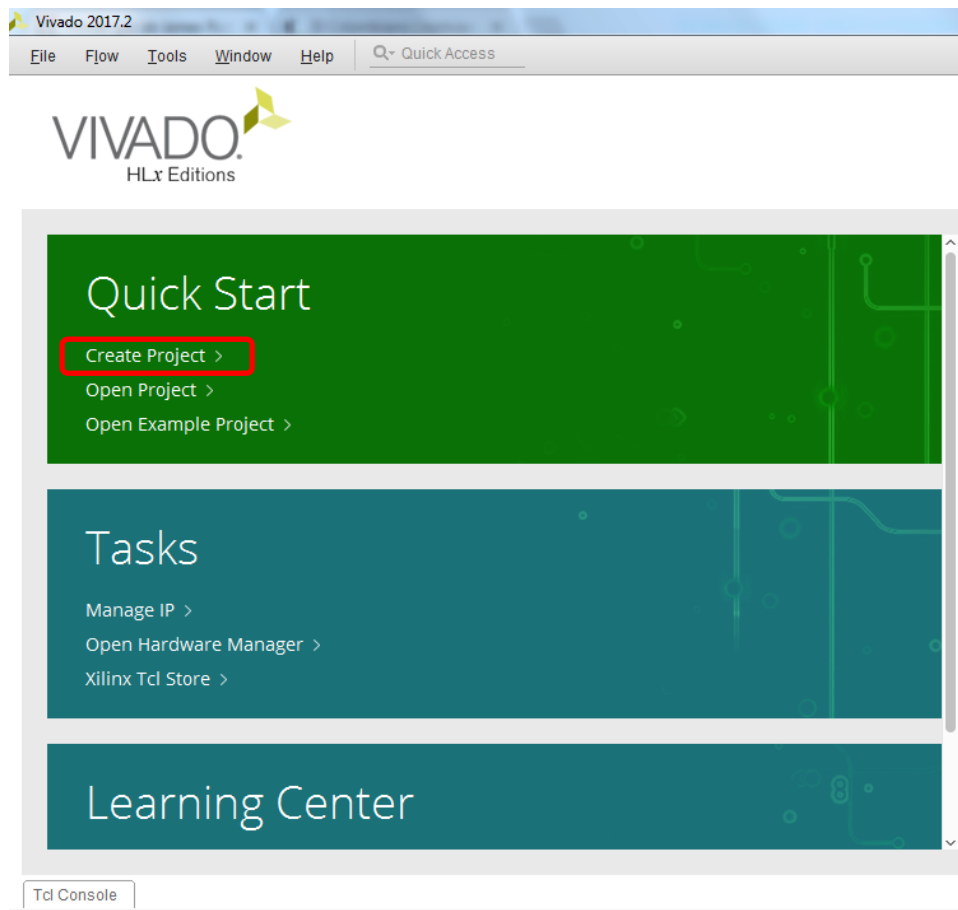
Para el desarrollo de la solución se recorren los pasos descritos a continuación:

CREACIÓN DEL PROYECTO:

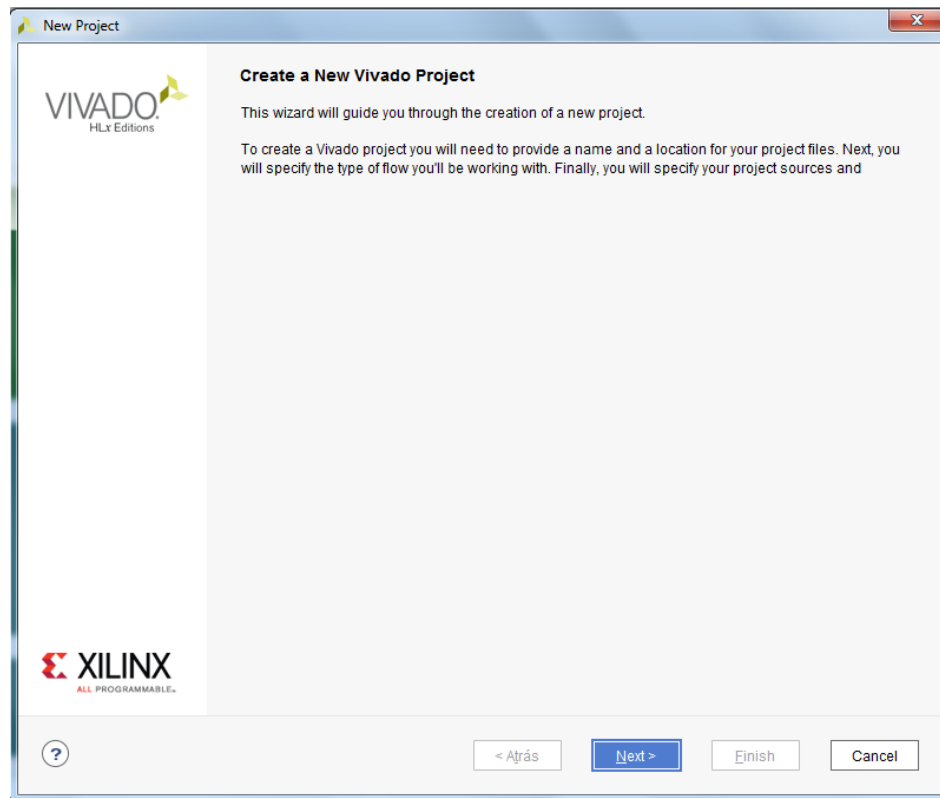
Para realizar cualquier aplicación es necesario inicialmente crear un proyecto.

1. Se inicia el programa Vivado (seleccionando cualquiera de las siguientes rutas):
 - a. Clic en el icono de vivado ubicado en el escritorio
- b. Clic en “Inicio”, clic en “Todos los Programas”, clic en “Vivado 2017.2”.
2. Una vez en el programa, se crea un proyecto:
 - a. Clic en “create project”

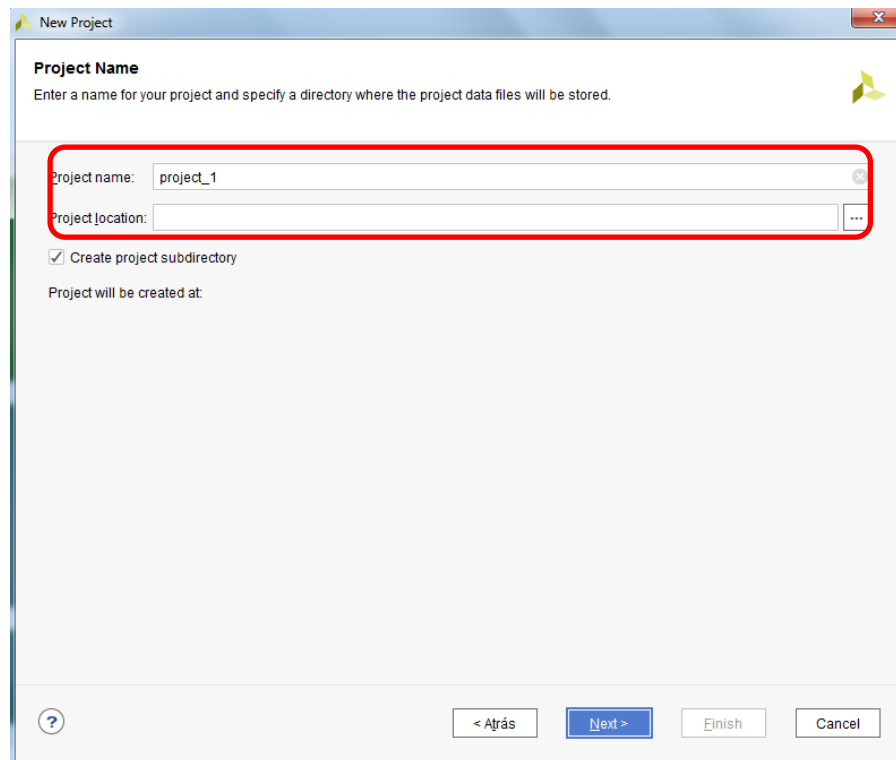




3. En la nueva ventana que se abre automáticamente se da clic en “next”:



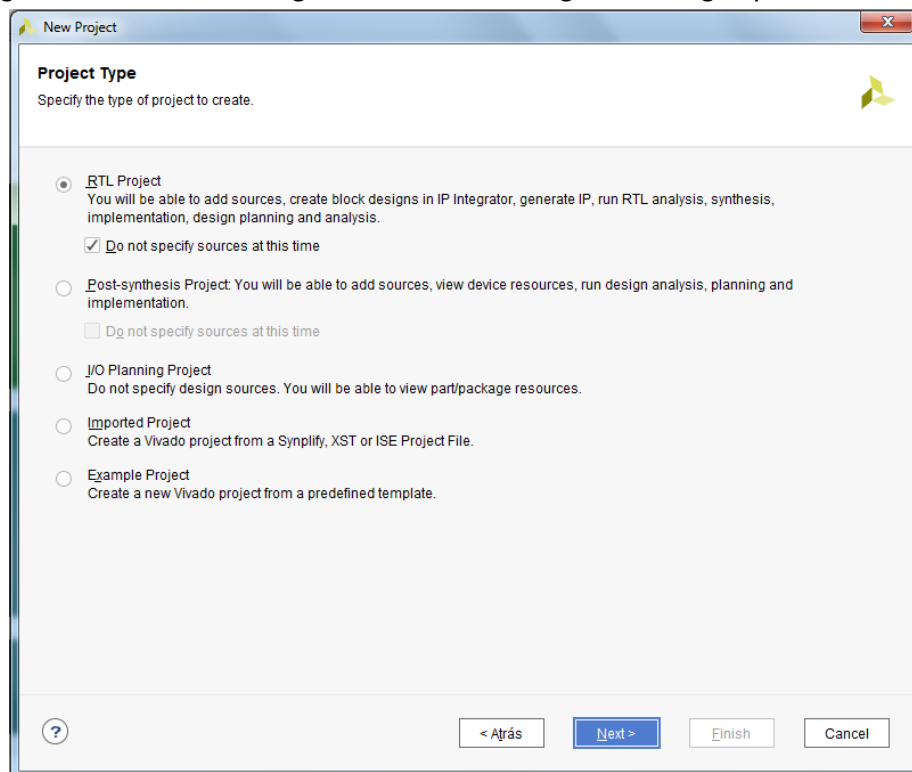
4. En la siguiente ventana se debe dar un nombre al proyecto y una ubicación y clic en “Next”



The screenshot shows the 'New Project' dialog box with the 'Project Name' tab selected. The dialog has a title bar with a yellow Vivado icon and standard window controls. The main area contains the following elements:

- Project Name**: A section header with a small Vivado icon to the right.
- Enter a name for your project and specify a directory where the project data files will be stored.**: A descriptive instruction.
- Project name:** A text input field containing 'project_1'.
- Project location:** A text input field with a browse button (three dots) to its right.
- ☒ **Create project subdirectory**: A checked checkbox.
- Project will be created at:**: A label for the next step.
- Navigation buttons**: A question mark icon, '< Atrás', 'Next >' (highlighted in blue), 'Finish', and 'Cancel'.

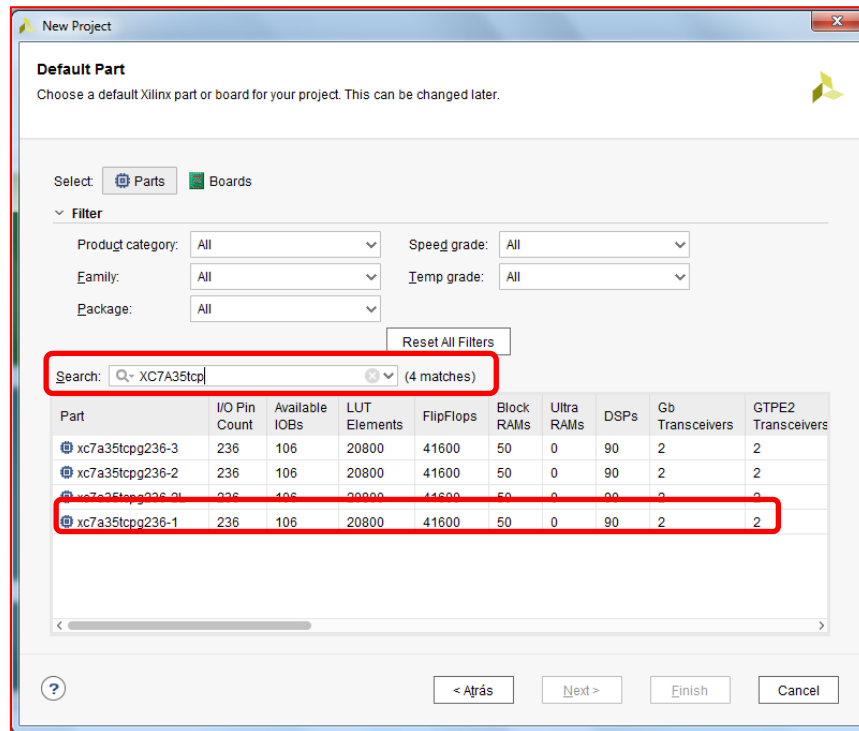
5. La siguiente ventana se configura de acuerdo a la siguiente imagen y clic en “Next”:



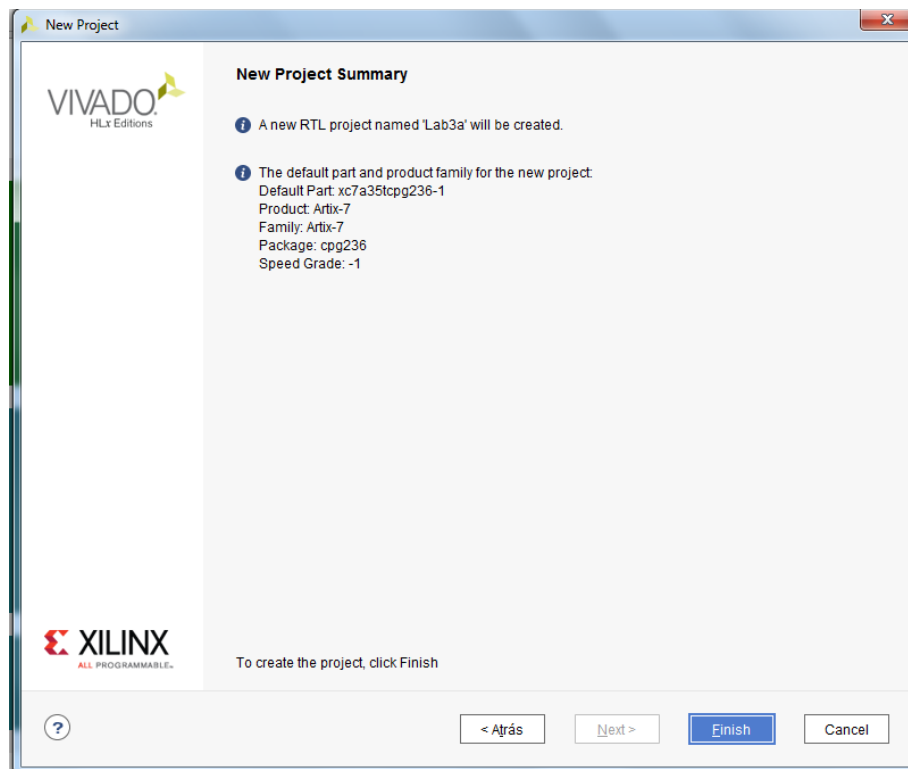
The screenshot shows the 'New Project' dialog box with the 'Project Type' tab selected. The dialog has a title bar with a yellow Vivado icon and standard window controls. The main area contains the following elements:

- Project Type**: A section header with a small Vivado icon to the right.
- Specify the type of project to create.**: A descriptive instruction.
- ☒ **RTL Project**: A selected radio button. Below it, the text reads: 'You will be able to add sources, create block designs in IP Integrator, generate IP, run RTL analysis, synthesis, implementation, design planning and analysis.' Below this is a checked checkbox labeled 'Do not specify sources at this time'.
- ☐ **Post-synthesis Project**: An unselected radio button. Below it, the text reads: 'You will be able to add sources, view device resources, run design analysis, planning and implementation.' Below this is an unchecked checkbox labeled 'Do not specify sources at this time'.
- ☐ **I/O Planning Project**: An unselected radio button. Below it, the text reads: 'Do not specify design sources. You will be able to view part/package resources.'
- ☐ **Imported Project**: An unselected radio button. Below it, the text reads: 'Create a Vivado project from a Synplify, XST or ISE Project File.'
- ☐ **Example Project**: An unselected radio button. Below it, the text reads: 'Create a new Vivado project from a predefined template.'
- Navigation buttons**: A question mark icon, '< Atrás', 'Next >' (highlighted in blue), 'Finish', and 'Cancel'.

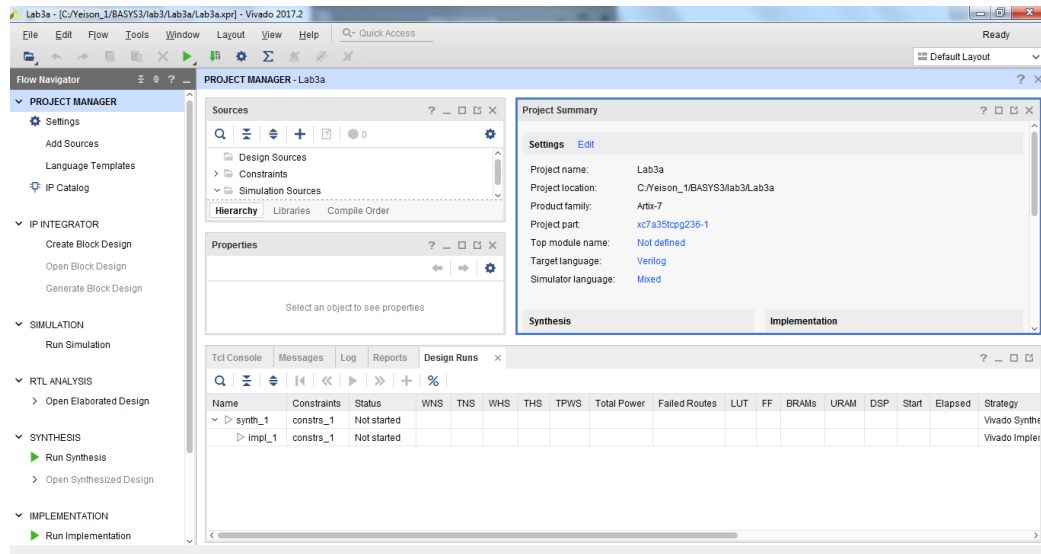
6. Seleccionar la FPGA a utilizar (xc7a35tcpg236-1) y clic en “Next”



7. Clic en “Finish”



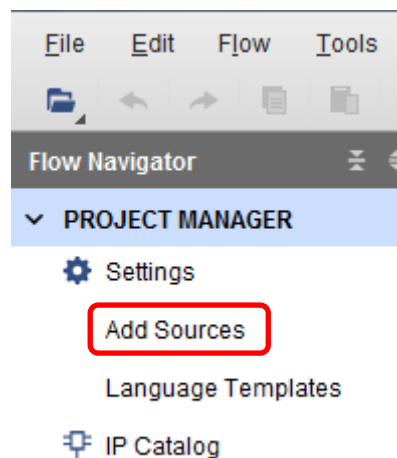
8. Se abre el entorno de desarrollo, como se muestra en la siguiente imagen.



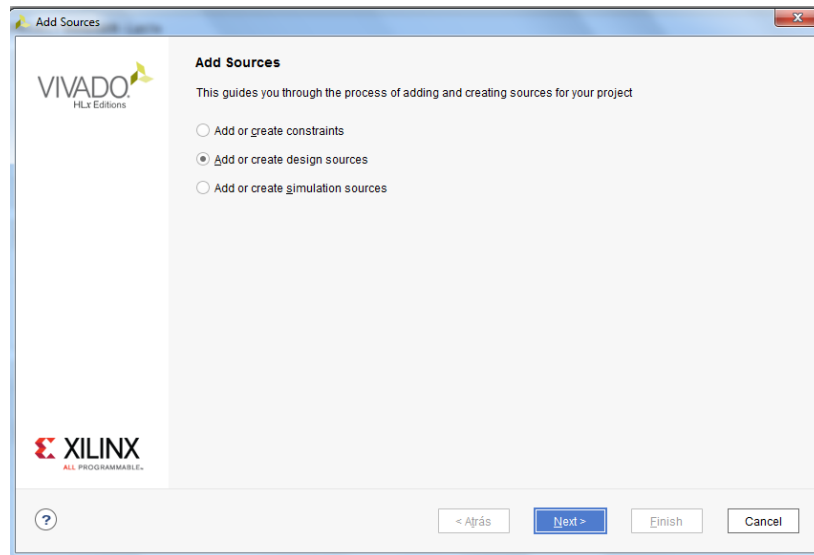
CÓDIGO EN VHDL:

Todas las aplicaciones se deben programar en VHDL, los pasos a continuación describen la creación del código en VHDL de un proyecto.

9. Clic en “Add Sources”

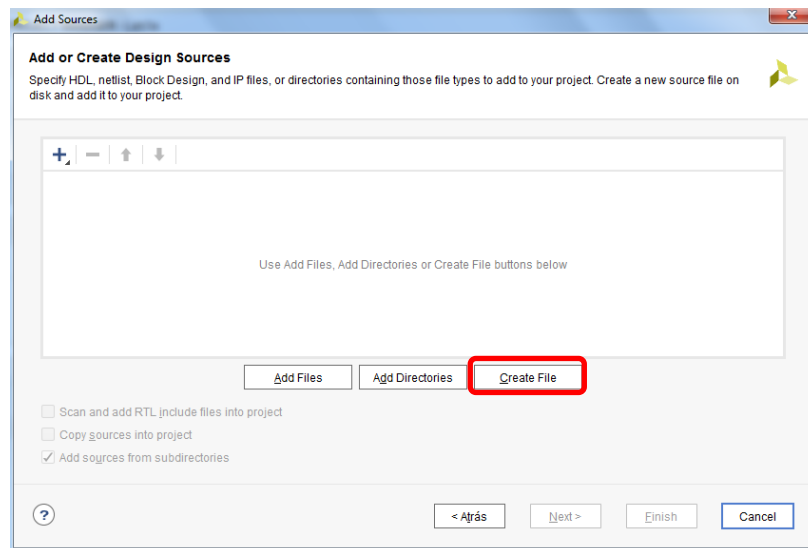


10. Se despliega una ventana que debe ser configurada de la siguiente manera:

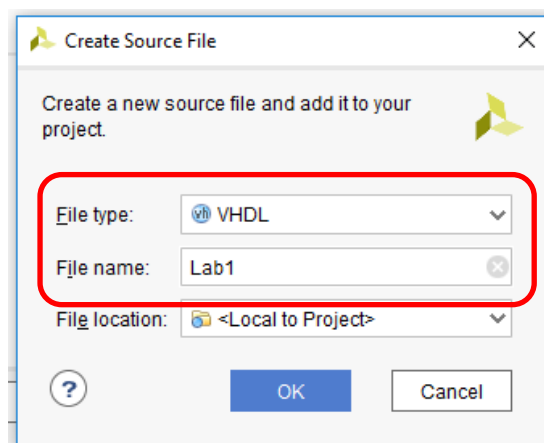


Clic en “Next”

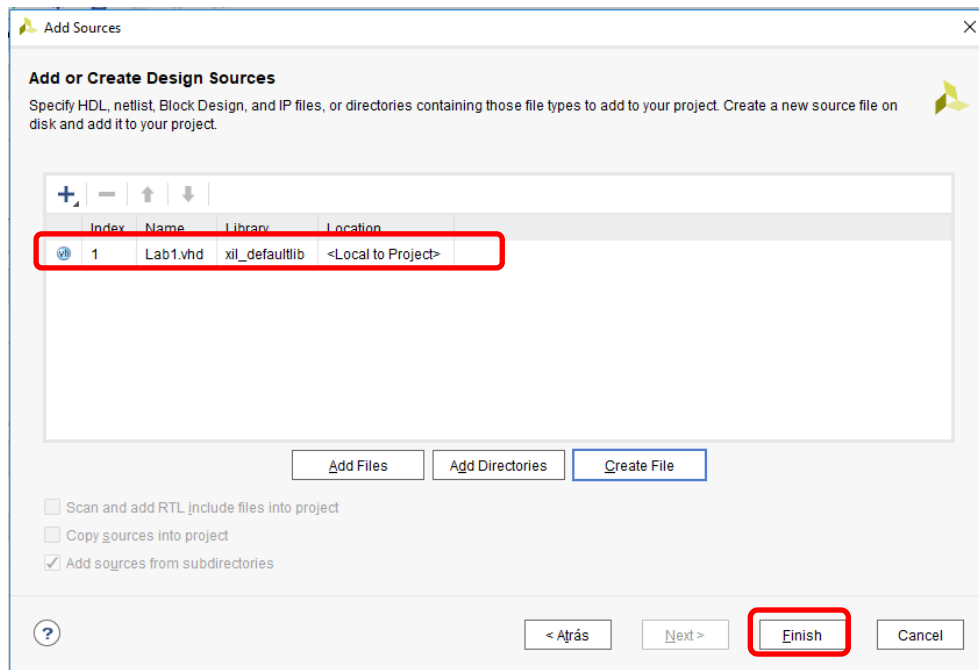
11. Clic en “Create File”



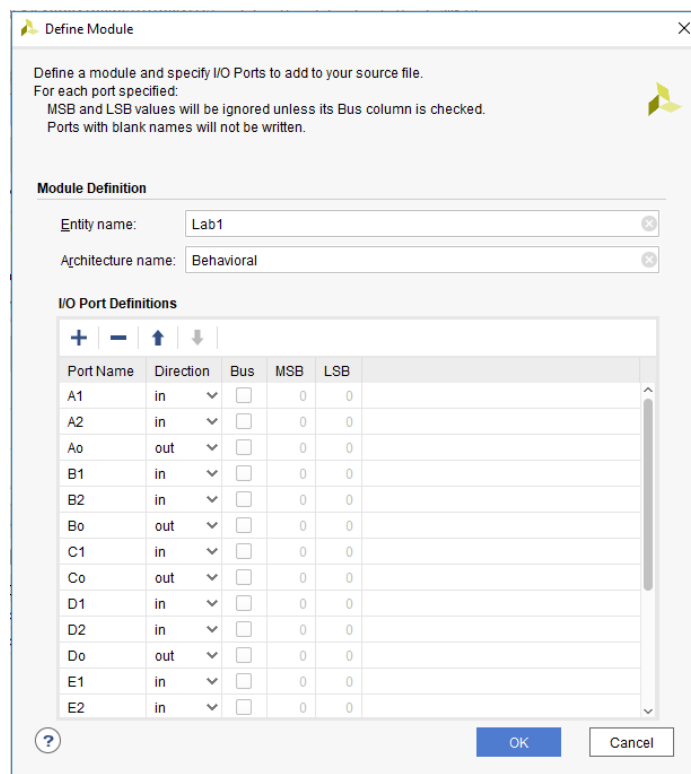
12. Se despliega una nueva ventana, en dicha ventana seleccionar el tipo de archivo como “VHDL” y se le asigna un nombre. Click en “OK”



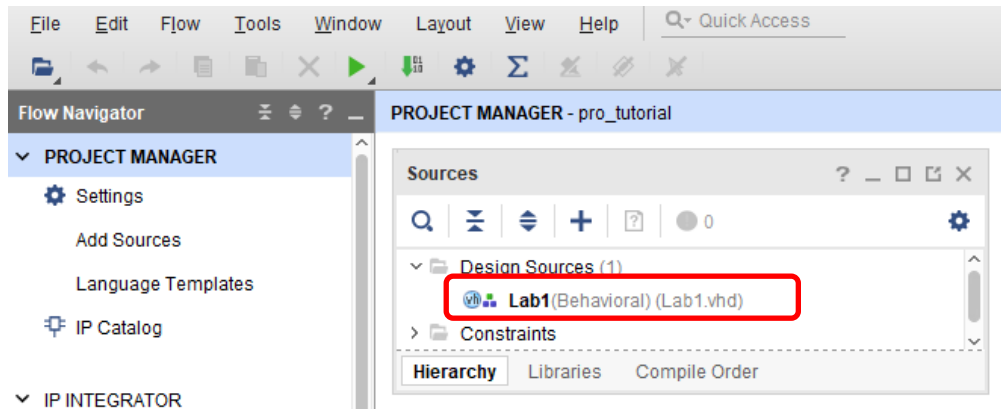
13. Verificar que el archivo esté en la lista y clic en “Finish”



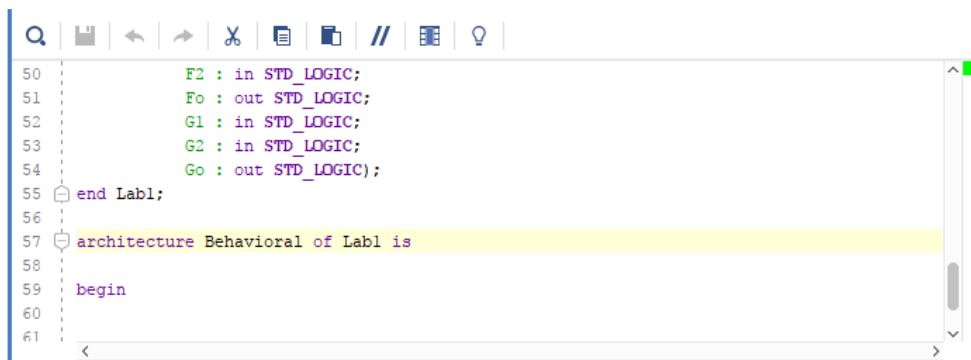
14. Se abre una nueva ventana, en dicha ventana indicar los pines de entrada y salida del circuito, para este caso las entradas serán A1, A2 (compuerta AND), B1, B2 (compuerta OR), C1 (compuerta NOT), D1, D2 (compuerta NAND), E1, E2 (compuerta NOR), F1, F2 (compuerta XOR), G1 y G2 (compuerta XNOR); por su parte las salidas serán: Ao (compuerta AND), Bo (compuerta OR), Co (compuerta NOT), Do (compuerta NAND), Eo (compuerta NOR), Fo (compuerta XOR) y Go (compuerta XNOR). Finalmente, clic en “OK”



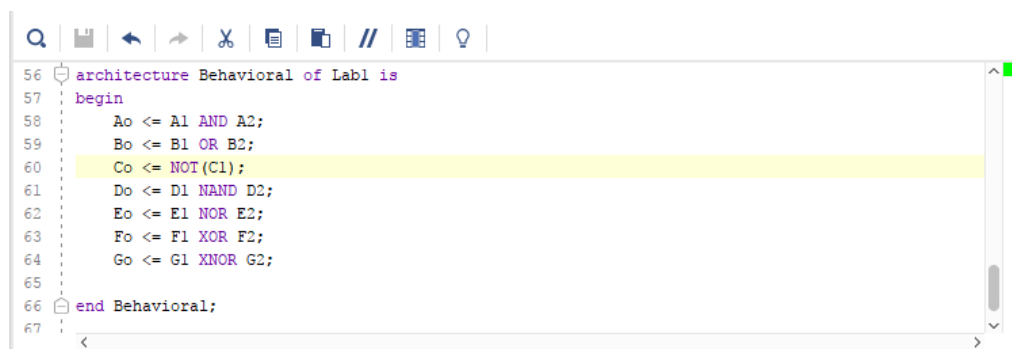
15. Doble clic en el archivo generado



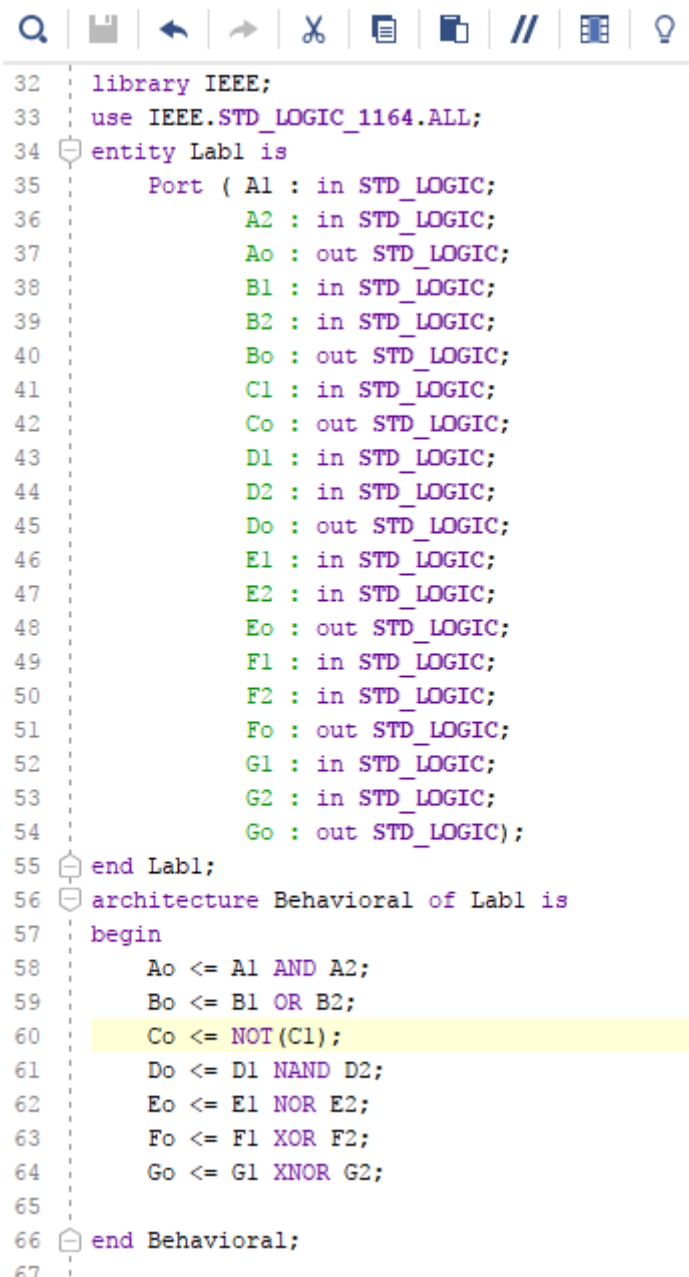
16. Se abre el archivo donde se escribe el código del proyecto.



17. Se escribe el código del proyecto en VHDL, para este caso, el código en VHDL para cada una de las compuertas



18. El programa completo quedaría de la siguiente manera:



```

32  library IEEE;
33  use IEEE.STD_LOGIC_1164.ALL;
34  entity Lab1 is
35      Port ( A1 : in STD_LOGIC;
36            A2 : in STD_LOGIC;
37            Ao : out STD_LOGIC;
38            B1 : in STD_LOGIC;
39            B2 : in STD_LOGIC;
40            Bo : out STD_LOGIC;
41            C1 : in STD_LOGIC;
42            Co : out STD_LOGIC;
43            D1 : in STD_LOGIC;
44            D2 : in STD_LOGIC;
45            Do : out STD_LOGIC;
46            E1 : in STD_LOGIC;
47            E2 : in STD_LOGIC;
48            Eo : out STD_LOGIC;
49            F1 : in STD_LOGIC;
50            F2 : in STD_LOGIC;
51            Fo : out STD_LOGIC;
52            G1 : in STD_LOGIC;
53            G2 : in STD_LOGIC;
54            Go : out STD_LOGIC);
55  end Lab1;
56  architecture Behavioral of Lab1 is
57  begin
58      Ao <= A1 AND A2;
59      Bo <= B1 OR B2;
60      Co <= NOT (C1);
61      Do <= D1 NAND D2;
62      Eo <= E1 NOR E2;
63      Fo <= F1 XOR F2;
64      Go <= G1 XNOR G2;
65
66  end Behavioral;
67

```

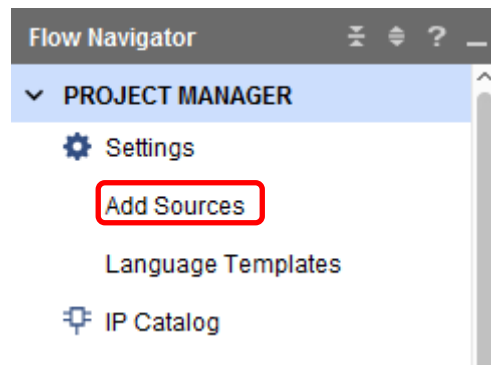
19. Clic en el icono de guardar



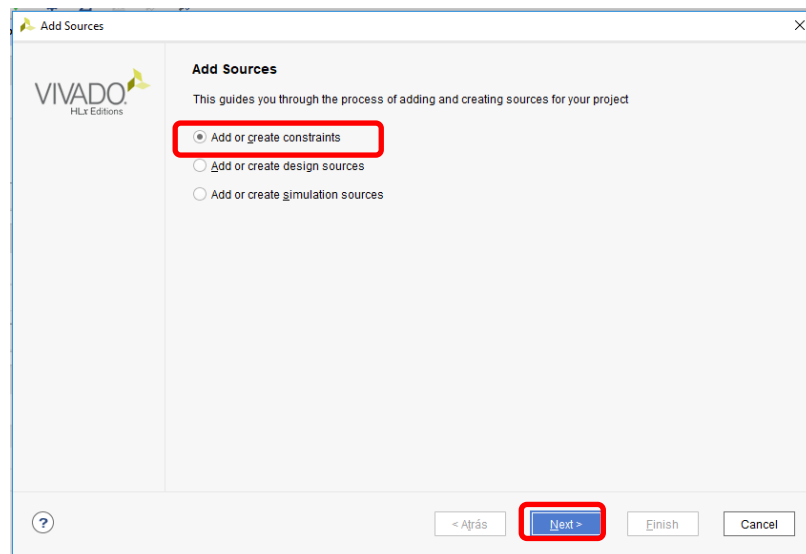
ASIGNACIÓN DE PINES:

Es necesario indicar cuales pines de la tarjeta BASYS 3 van a ser usados como entradas y salidas de la aplicación. Los pasos a continuación describen la forma de asignar las entradas y salidas.

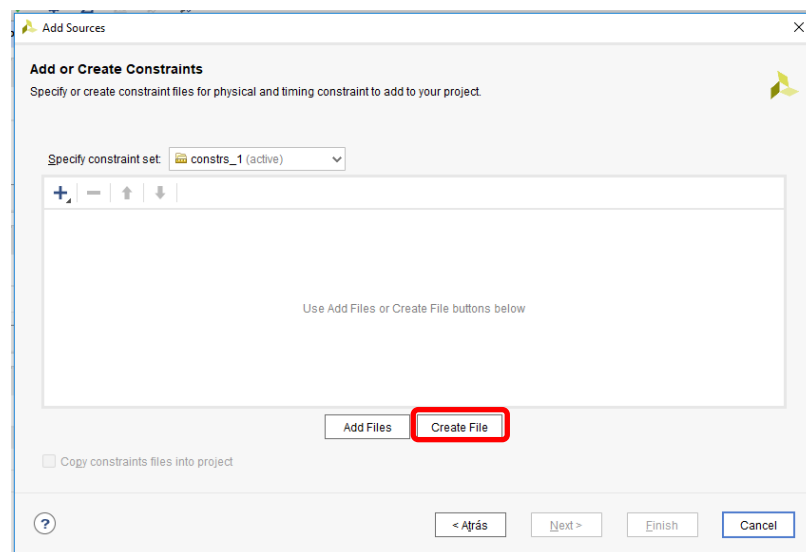
20. Se deben asignar los pines de entrada y salida del sistema de desarrollo BASYS 3 al código desarrollado, para ello:
 - a. Clic en "Add Sources"



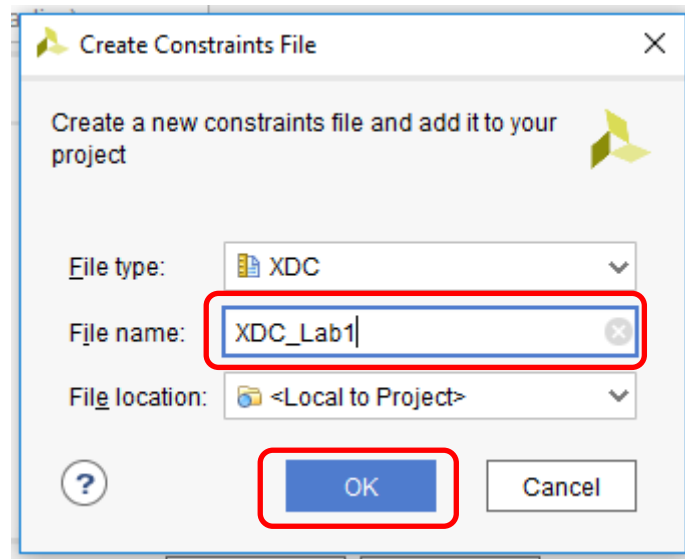
- b. En la nueva ventana que se despliega, se selecciona la opción “Add or create constraints” y clic en “Next”:



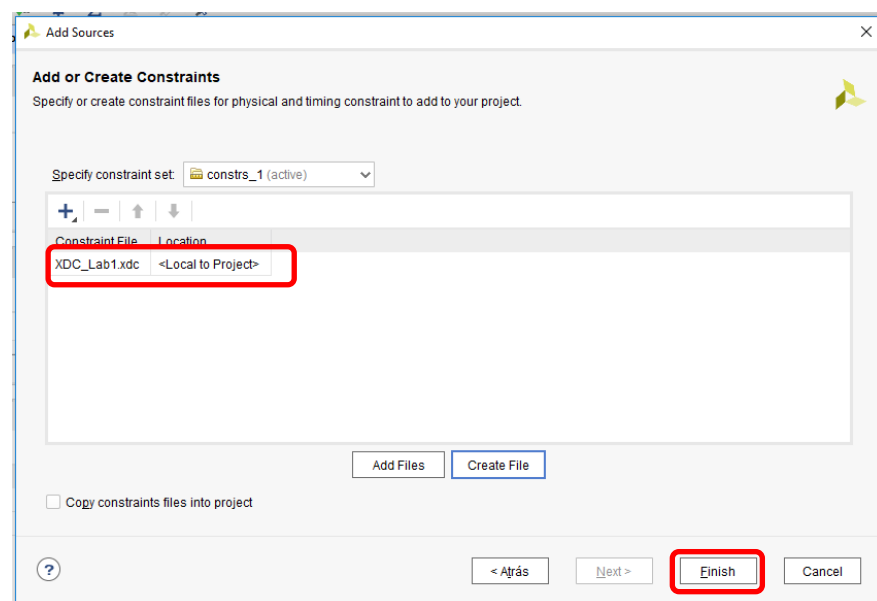
- c. En la siguiente ventana se debe dar clic en “Create File”



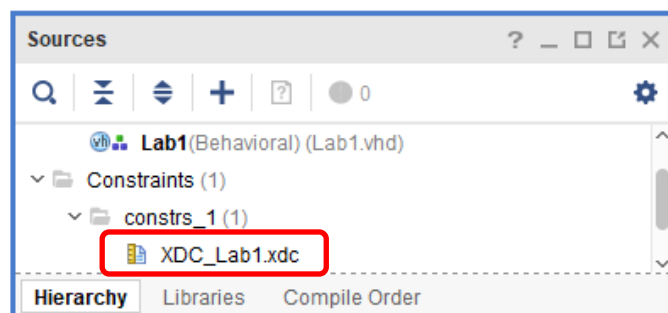
- d. En la ventana que se despliega, se le asigna un nombre al nuevo archivo y clic en "OK"



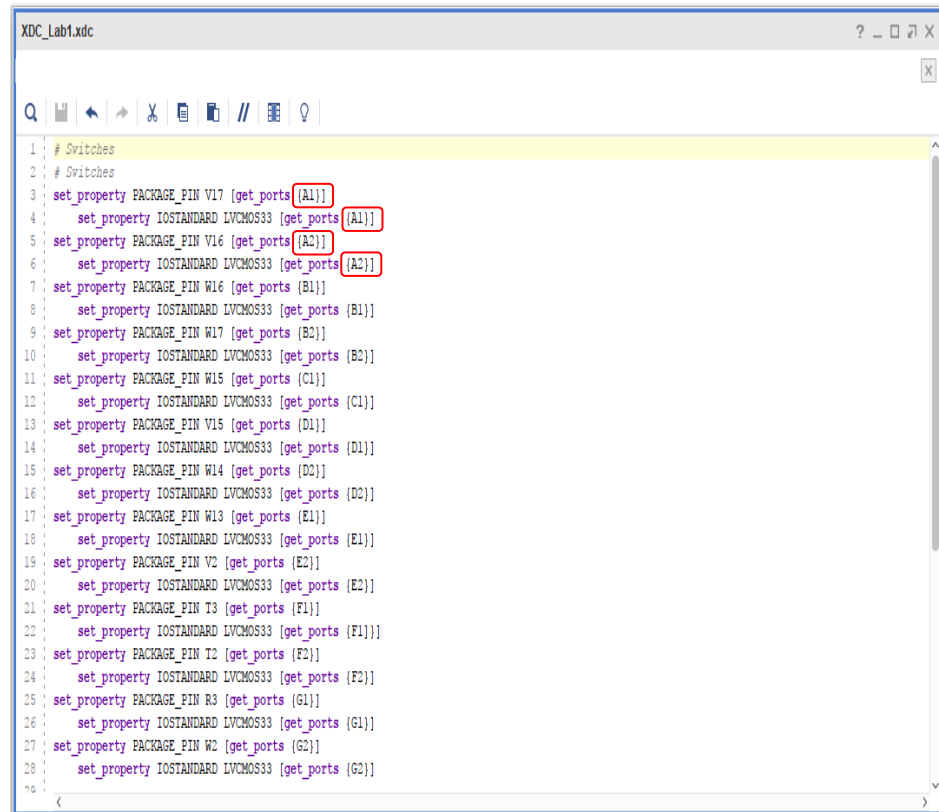
- e. Confirmar que el nuevo archivo creado, aparece en la lista de archivos y clic en "Finish"



- f. A continuación, se debe abrir el nuevo archivo creado, el cual se encuentra en la pestaña "Sources" en el panel principal.

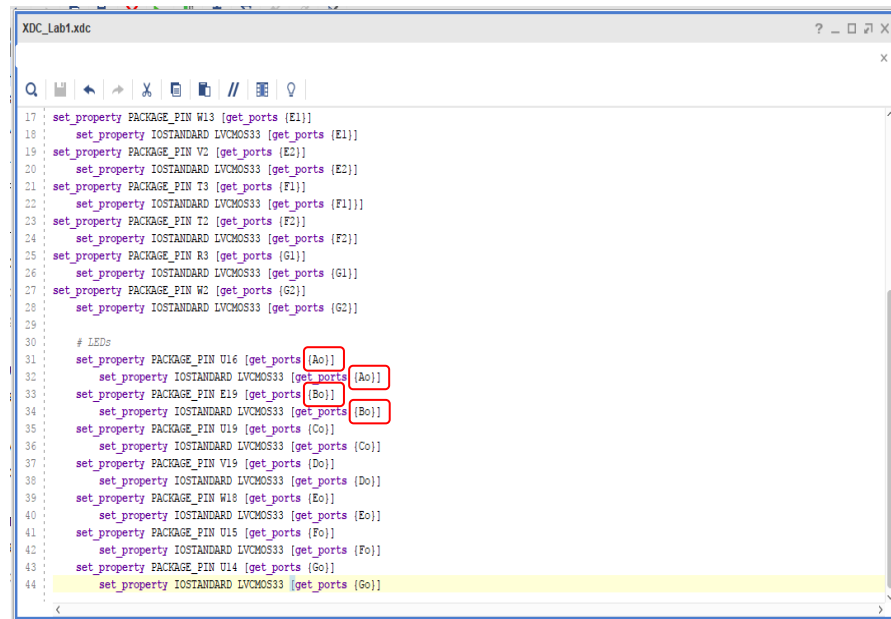


- g. Una vez se abra el archivo .xdc, se debe ingresar al siguiente link:
https://github.com/Digilent/Basys3/blob/master/Projects/XADC_Demo/src/constraints/Basys3_Master.xdc
Copiar los pines que se necesitan para el proyecto y pegarlo en el nuevo archivo creado (para este caso Entradas – *Switches* – y Salidas – *LEDs* -).
- h. Reemplazar el nombre de los pines que aparecen en la página por el nombre de los pines que se asignaron en el código a las entradas.



```
1 # Switches
2 # Switches
3 set_property PACKAGE_PIN V17 [get_ports {A1}]
4 set_property IOSTANDARD LVCMOS33 [get_ports {A1}]
5 set_property PACKAGE_PIN V16 [get_ports {A2}]
6 set_property IOSTANDARD LVCMOS33 [get_ports {A2}]
7 set_property PACKAGE_PIN W16 [get_ports {B1}]
8 set_property IOSTANDARD LVCMOS33 [get_ports {B1}]
9 set_property PACKAGE_PIN W17 [get_ports {B2}]
10 set_property IOSTANDARD LVCMOS33 [get_ports {B2}]
11 set_property PACKAGE_PIN W15 [get_ports {C1}]
12 set_property IOSTANDARD LVCMOS33 [get_ports {C1}]
13 set_property PACKAGE_PIN V15 [get_ports {D1}]
14 set_property IOSTANDARD LVCMOS33 [get_ports {D1}]
15 set_property PACKAGE_PIN W14 [get_ports {D2}]
16 set_property IOSTANDARD LVCMOS33 [get_ports {D2}]
17 set_property PACKAGE_PIN W13 [get_ports {E1}]
18 set_property IOSTANDARD LVCMOS33 [get_ports {E1}]
19 set_property PACKAGE_PIN V2 [get_ports {E2}]
20 set_property IOSTANDARD LVCMOS33 [get_ports {E2}]
21 set_property PACKAGE_PIN T3 [get_ports {F1}]
22 set_property IOSTANDARD LVCMOS33 [get_ports {F1}]
23 set_property PACKAGE_PIN T2 [get_ports {F2}]
24 set_property IOSTANDARD LVCMOS33 [get_ports {F2}]
25 set_property PACKAGE_PIN R3 [get_ports {G1}]
26 set_property IOSTANDARD LVCMOS33 [get_ports {G1}]
27 set_property PACKAGE_PIN W2 [get_ports {G2}]
28 set_property IOSTANDARD LVCMOS33 [get_ports {G2}]
29
```

- i. Reemplazar el nombre de los pines que aparecen en la página por el nombre de los pines que se asignaron en el código a las entradas.



```

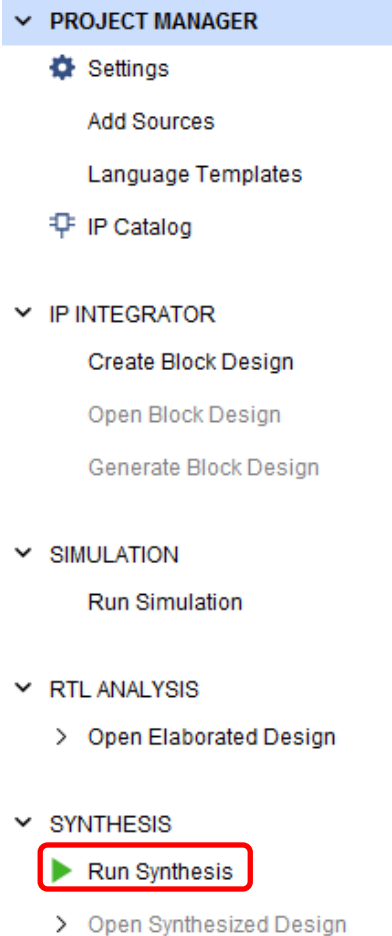
17 set_property PACKAGE_PIN W13 [get_ports {E1}]
18 set_property IOSTANDARD LVCMOS33 [get_ports {E1}]
19 set_property PACKAGE_PIN V2 [get_ports {E2}]
20 set_property IOSTANDARD LVCMOS33 [get_ports {E2}]
21 set_property PACKAGE_PIN T3 [get_ports {F1}]
22 set_property IOSTANDARD LVCMOS33 [get_ports {F1}]
23 set_property PACKAGE_PIN T2 [get_ports {F2}]
24 set_property IOSTANDARD LVCMOS33 [get_ports {F2}]
25 set_property PACKAGE_PIN R3 [get_ports {G1}]
26 set_property IOSTANDARD LVCMOS33 [get_ports {G1}]
27 set_property PACKAGE_PIN W2 [get_ports {G2}]
28 set_property IOSTANDARD LVCMOS33 [get_ports {G2}]
29
30 # LEDs
31 set_property PACKAGE_PIN U16 [get_ports {Ao}]
32 set_property IOSTANDARD LVCMOS33 [get_ports {Ao}]
33 set_property PACKAGE_PIN E19 [get_ports {Bo}]
34 set_property IOSTANDARD LVCMOS33 [get_ports {Bo}]
35 set_property PACKAGE_PIN U19 [get_ports {Co}]
36 set_property IOSTANDARD LVCMOS33 [get_ports {Co}]
37 set_property PACKAGE_PIN V19 [get_ports {Do}]
38 set_property IOSTANDARD LVCMOS33 [get_ports {Do}]
39 set_property PACKAGE_PIN W18 [get_ports {Eo}]
40 set_property IOSTANDARD LVCMOS33 [get_ports {Eo}]
41 set_property PACKAGE_PIN U15 [get_ports {Fo}]
42 set_property IOSTANDARD LVCMOS33 [get_ports {Fo}]
43 set_property PACKAGE_PIN U14 [get_ports {Go}]
44 set_property IOSTANDARD LVCMOS33 [get_ports {Go}]

```

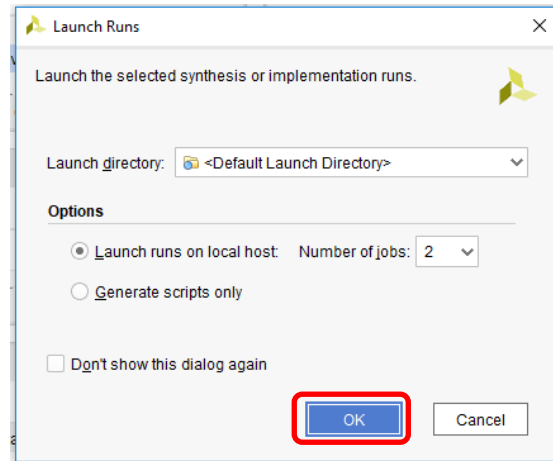
j. Clic en guardar

SINTESIS:

21. A continuación, se sintetiza el proyecto. Clic en “Run Synthesis”



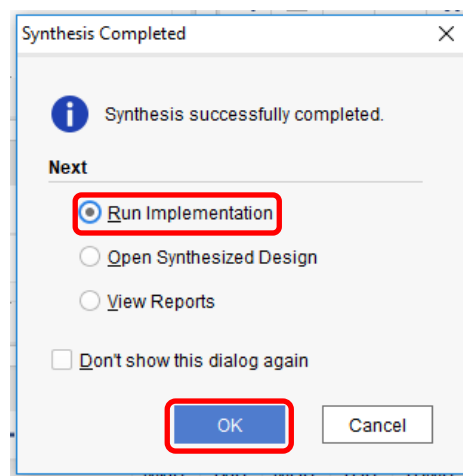
- a. En la ventana que se abre clic en “OK”



- b. Esperar que termine el proceso

IMPLEMENTACIÓN:

22. Una vez terminado la síntesis del proyecto se continua en la implementación del proyecto. Es posible hacerlo mediante dos rutas:
- a. Seleccionando “Run Implementation” y dando clic en “OK” en la ventana que se abre al final el proceso de síntesis.

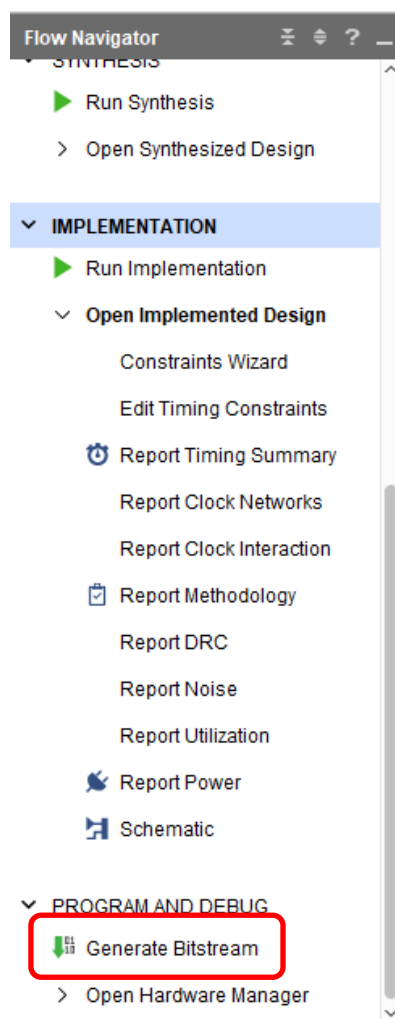


- b. Dando clic en “Run Implementation”

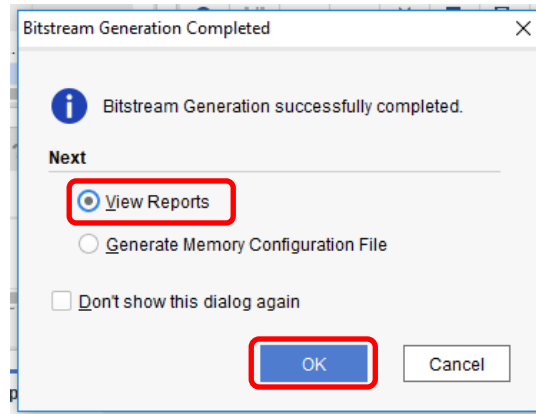
- ▼ IP INTEGRATOR
 - Create Block Design
 - Open Block Design
 - Generate Block Design
- ▼ SIMULATION
 - Run Simulation
- ▼ RTL ANALYSIS
 - > Open Elaborated Design
- ▼ SYNTHESIS
 - ▶ Run Synthesis
 - > Open Synthesized Design
- ▼ IMPLEMENTATION
 - ▶ Run Implementation
 - > Open Implemented Design

BITSTREAM:

23. Generación de *Bitstream*. Para ello clic en “*Generate Bitstream*”

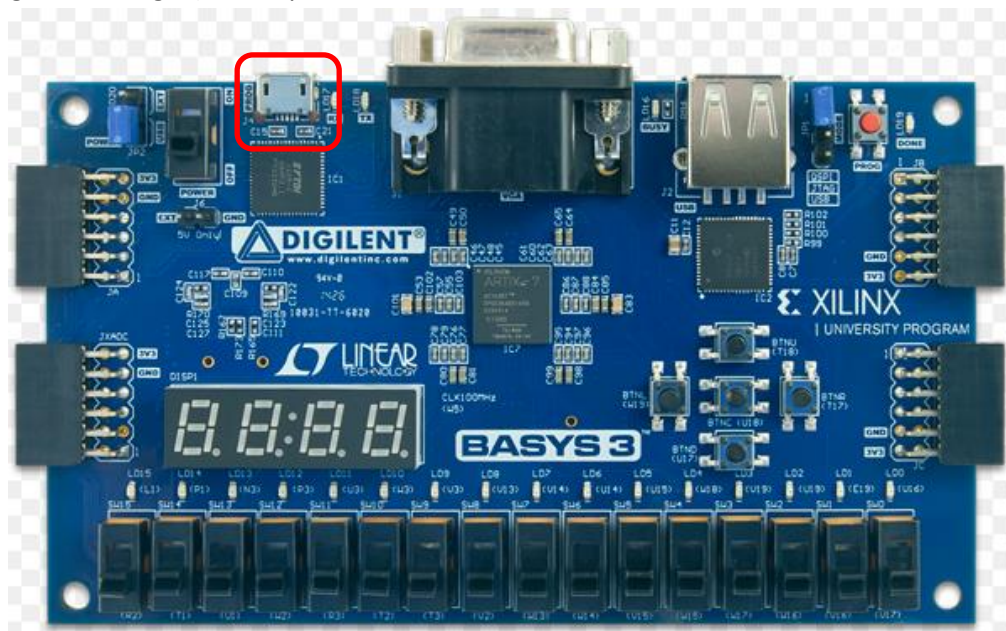


24. Seleccionar “View Reports” y clic en “OK”



PROGRAMACIÓN:

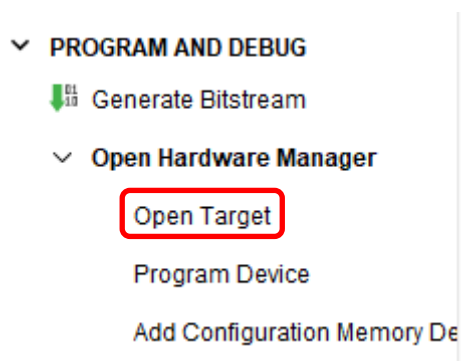
25. A continuación, el sistema de desarrollo BASYS 3 se conecta al PC por medio del cable USB que une el conector de programación del sistema de desarrollo (indicado en la siguiente imagen) con el puerto del PC.



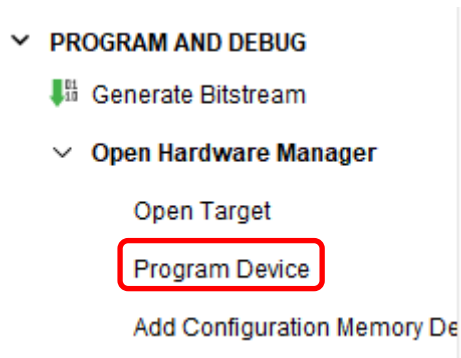
26. Encender el sistema de desarrollo BASYS 3



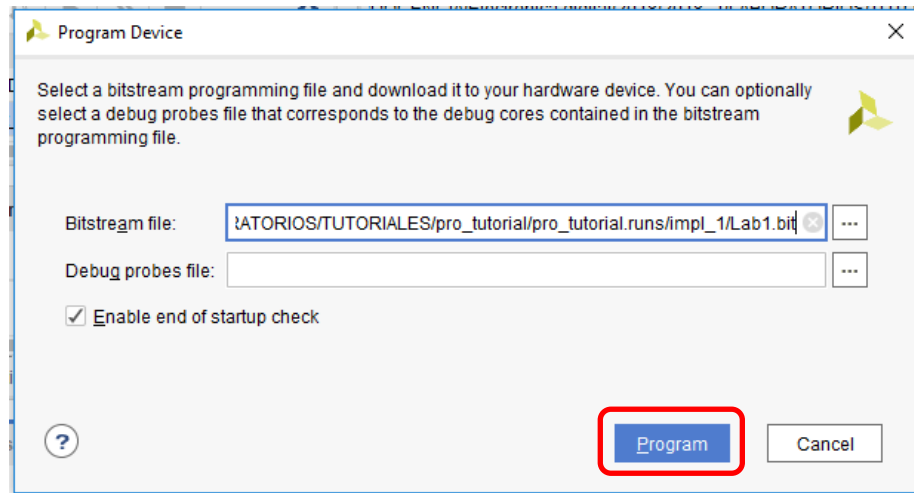
27. En “Open Hardware Manager” clic en “Open Target”



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



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



COMPROBACIÓN DE FUNCIONAMIENTO:

30. Una vez programada la tarjeta se procede a comprobar el funcionamiento del circuito implementado, para este caso, se hace uso de los interruptores y los LED's.
31. Compruebe la tabla de verdad de cada una de las compuertas implementadas.