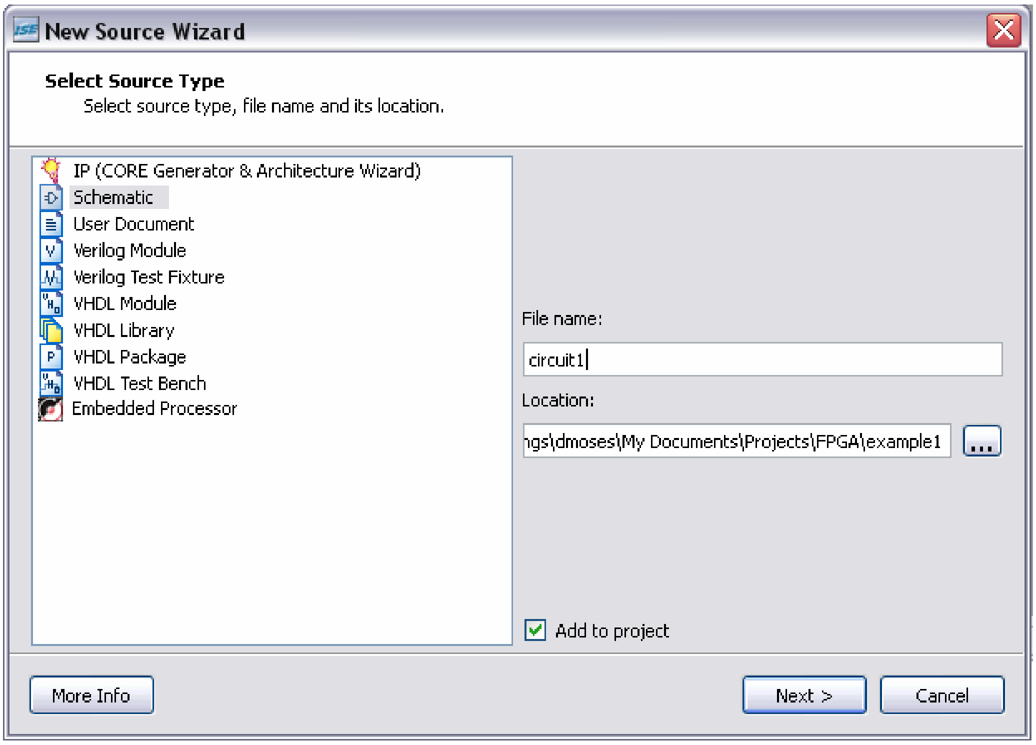
In this tutorial, we create a new source file, so select New Source from the list. This starts the New Source Wizard, which prompts you for the Source type and file name. Select Schematic and give it a meaningful name (we name it circuit1).



New Source Wizard

Select Source Type

Select source type, file name and its location.

- IP (CORE Generator & Architecture Wizard)
- Schematic
- User Document
- Verilog Module
- Verilog Test Fixture
- VHDL Module
- VHDL Library
- VHDL Package
- VHDL Test Bench
- Embedded Processor

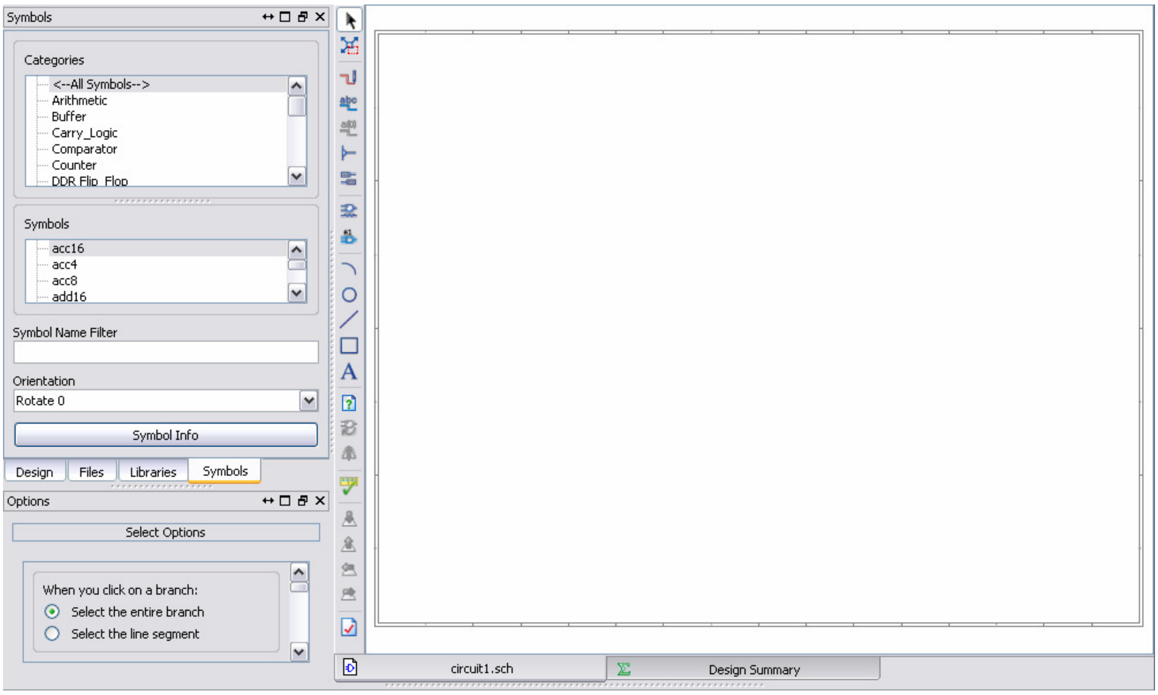
File name: circuit1

Location: ngs\dmoses\My Documents\Projects\FPGA\example1

☑ Add to project

Schematic Editor Window

Once you have created the new schematic file you can see it in the Sources panel. Double-click it to open the file in the schematic capture window as follows:



Orientation: Rotate 0

When you click on a branch:
● Select the entire branch
○ Select the line segment

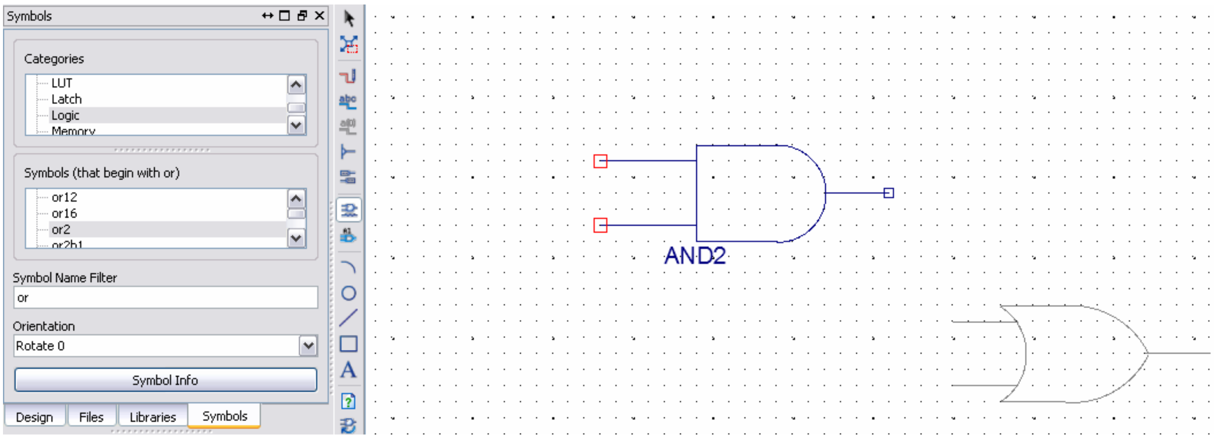
You can add symbols and shapes representing logic gates or logic circuits and then add lines representing wires to connect those shapes.

On the screen, there are two list boxes labeled Symbols and Categories.

The Symbols list shows all of the symbols in selected category in the Categories list. For example, the “<--All symbols-->” category displays all symbols in the current library in the Symbols list.

In this tutorial, we use a simple combinational logic example, and then show how it can be used as a macro symbol in another schematic module. We start with the basic logic equation: Y<= (A·B) + C.

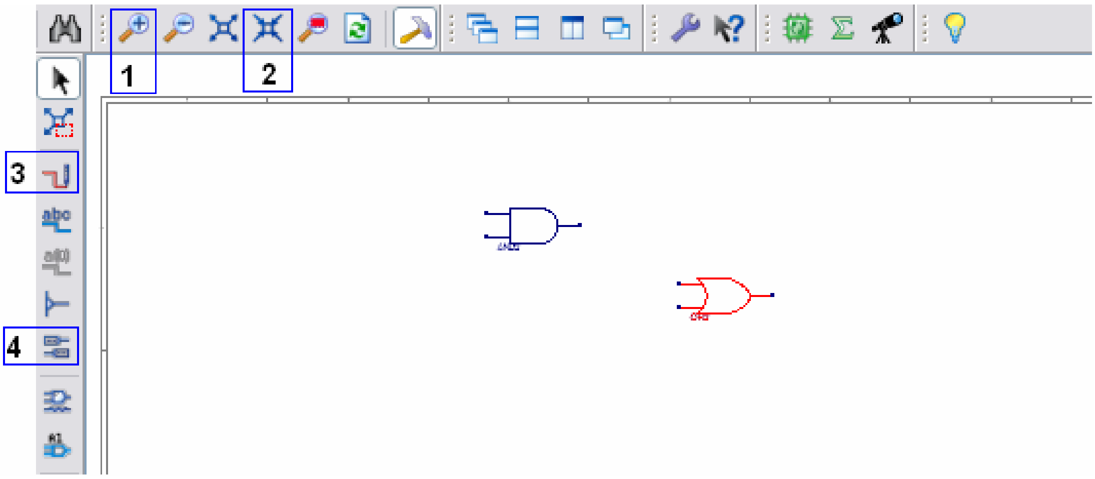
To add a simple logic gate to the circuit, left-click to select a symbol from the Symbols list, drag the cursor to the pallet where you want the symbol to be placed, and left-click again to drop the gate in the schematic. You can also find a symbol by typing a name into the symbol name filter.



Symbol Name Filter: or

Orientation: Rotate 0

The following figure contains descriptions for the tools necessary for creating a basic digital circuit using schematic capture:



1. The Magnifying Glass icon indiscriminately zooms to the center of the schematic.

2. The Zoom box icon is used to draw a box using the mouse to magnify a specific area of the schematic.

3. The Wire-Add tool icon places cursor in wire-add mode.

4. The Add I/O marker tool icon places cursor into add I/O marker mode.

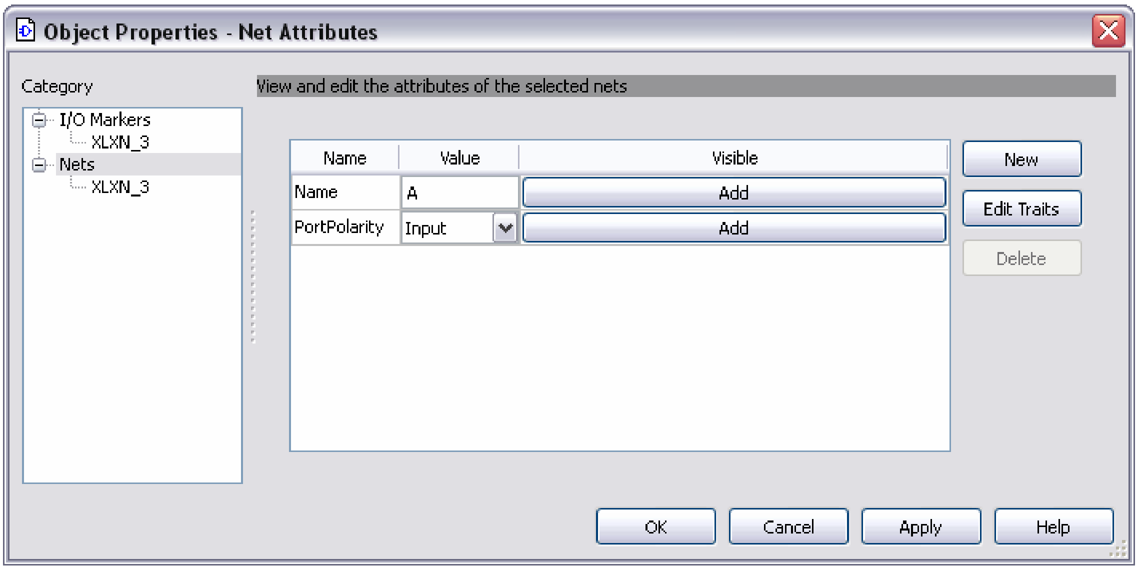
Now we connect the gates with wires. To do this, left-click on the wire-add tool button to change to wire-add mode. Drag the cursor to a component pin where four red boxes appear (they indicate that a left-click will add one end of the wire to the pin). Using this method you can add wires between pins of gates.

To move a gate, you must go back to select mode by either selecting the cursor in the tool bar or pressing the Esc button on the keyboard.

To terminate one end of a wire without connecting it to a gate, double-click where you want the wire to end in space. This is used for general circuit inputs and outputs.

Adding top-level I/O markers to your circuit tells the synthesizer and simulator tools which ports to regard as overall inputs and outputs. To add these to your circuit, left-click on the Add I/O marker tool icon to place the cursor into add I/O marker mode. Left-click on the end of a wire to add an I/O marker and then repeat until a marker is placed on all inputs and outputs.

Go back to regular cursor mode and double left-click on an I/O marker. When the I/O marker's object properties dialog box appears, select the Nets category and enter a meaningful value for the Name field of the I/O marker. The finished dialog box should look similar to the following:

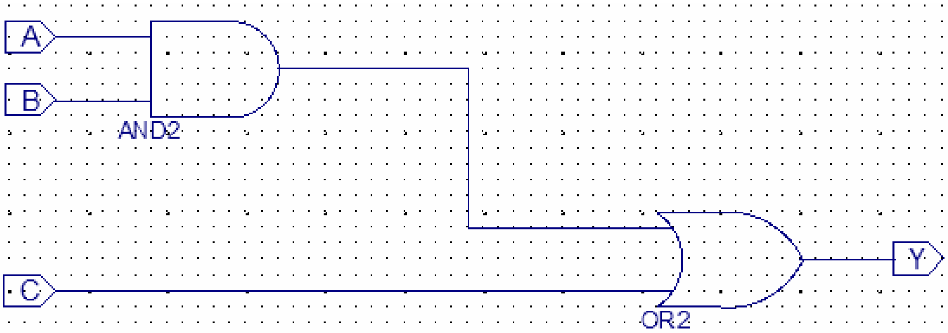

Object Properties - Net Attributes

Category
- I/O Markers
  - XLXN_3
- Nets
  - XLXN_3

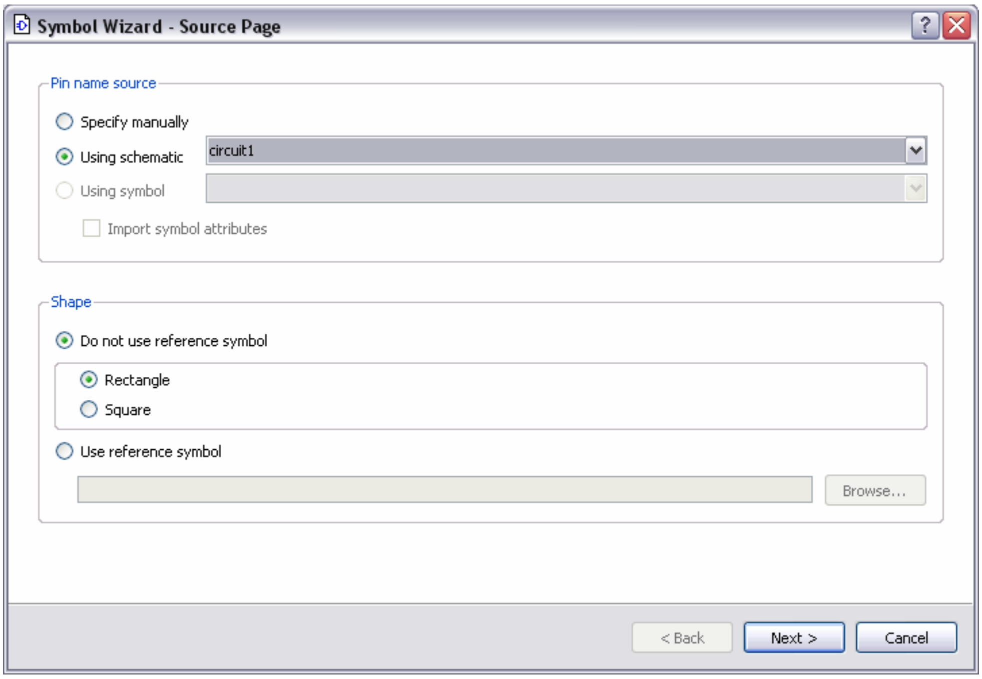
View and edit the attributes of the selected nets

Name	Value	Visible
Name	A	Add
PortPolarity	Input	Add

Repeat this process for all I/O. The finished circuit should look similar to the following:



Now we create a macro symbol for future use of this completed circuit. To create a macro circuit, select Tools > Symbol Wizard from the main menu toolbar. The Symbol Wizard dialog box appears. Under "Pin name source", select "Using schematic" which defaults to the circuit you just created, circuit1.



Symbol Wizard - Source Page

Pin name source

- ○ Specify manually
- ● Using schematic: circuit1
- ○ Using symbol
- □ Import symbol attributes

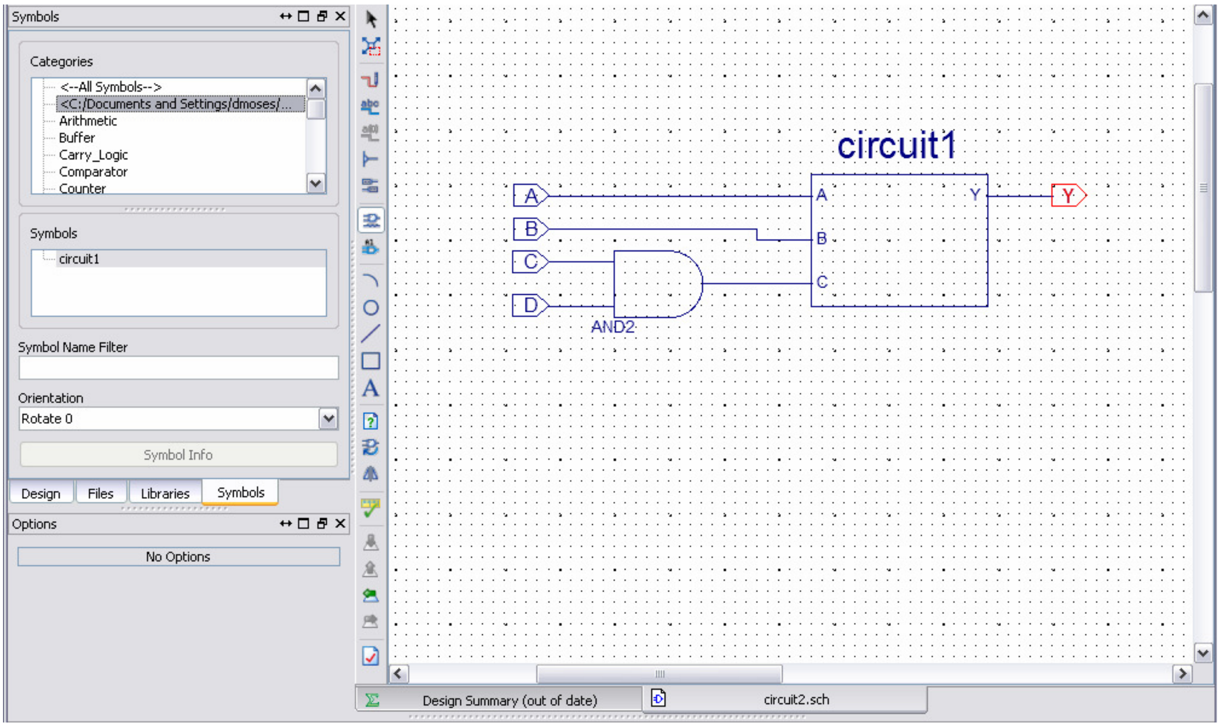
Shape

- ● Do not use reference symbol
  - ● Rectangle
  - ○ Square
- ○ Use reference symbol

The next two dialog boxes present general pin placement and symbol size options that you can modify depending on how you want the schematic symbol to appear. The final dialog box gives you a preview of the symbol, given the settings you've selected. Click Finish to finalize the symbol or Back to modify the symbol settings.

Save changes to the current schematic editor session before closing the schematic file and return to the Project Navigator main window. Right-click on the device to add a new schematic source file; we call ours circuit2. In the new schematic editor session you can now find the newly created symbol for circuit1 in the symbols window and add it to your overall design.

The following figure shows our completed example of using a macro symbol to build the overall circuit Y<= (A · B) + (C · D):



Orientation: Rotate 0

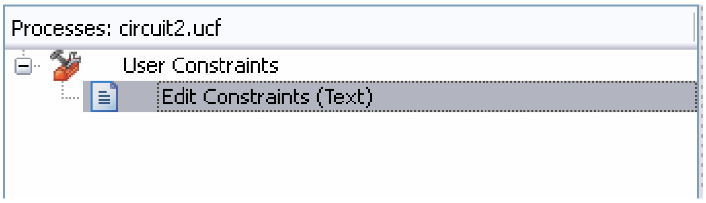
Using this method, you can create complex circuits by adding any components from the Xilinx library or by building your own components and adding them to an even larger design.

UCF File Creation

The Xilinx tools use a User Constraints File (.ucf file) to define user constraints like physical pin to circuit net mappings. This is sometimes referred to as an Implementation Constraints File. The .ucf file can be modified inside ISE using a text editor.

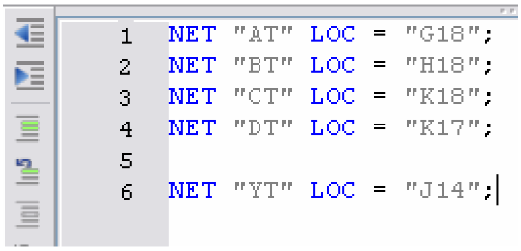
To add a .ucf file to your design, go to the Sources window and right-click the source file that requires user constraints. Select the Add New Source option in the drop-down menu. The New Source Wizard prompts you for the Source type and file name. Select Implementation Constraints File and give it a meaningful name (we name it circuit2).

To edit the .ucf file, select it in the Sources window, expand the User Constraints option in the Processes window below, and double-click the Edit Constraints (Text) option. A blank text editor appears.



To associate a physical pin with a given net name, type: NET “netname” LOC =”XXX”; on a line in the .ucf file. In the statement, “netname” (quotes included) is the name of the net to attach to pin number XXX (quotes included).

For our example project, the four inputs are assigned to switches 0 through 3 and the output is assigned to LED0 on the Nexys2 board. The finished .ucf file is as follows:



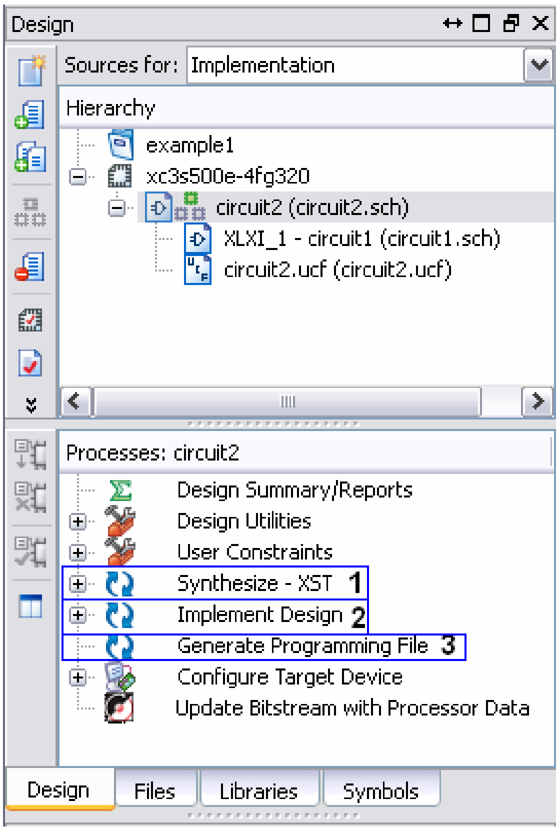
```
NET "AT" LOC = "G18";
NET "BT" LOC = "H18";
NET "CT" LOC = "K18";
NET "DT" LOC = "K17";

NET "YT" LOC = "J14";
```

Programming File Generation

Now we are ready to create a programming file (.bit) for the Nexys2 FPGA.

Go to the Sources window and select the top-level module (indicated by the three blocks shown with the source name.)



Sources for: Implementation

Now go to the Processes window where there are three particular processes in a row:

1. Synthesize – XST
2. Implement Design
3. Generate Programming file

Run the synthesis process by either double-clicking on Synthesize or left-clicking and selecting the run option. This process analyzes the circuit you have created, checking for valid connections, syntax, and structure, to verify that the circuit is valid and synthesizable.

If the Synthesize process does not return any errors, you can move on and run the Implement Design process. This process uses various algorithms to map out the digital circuit and then creates place and route information so that it can be placed on the physical FPGA.