Introduction to using VHDL Design on Quartus Prime

This tutorial will serve as an introduction to the Quartus Prime CAD system. It will provide a general overview on how CAD flow for circuit design is implemented using FPGA devices and will provide a visual representation of how the flow is implemented int the Quartus Prime software. The design process is detailed by giving step by step instructions for using Quartus prime to implement a very simple circuit on an FPGA device.

The Quartus Prime software offers full support for all popular methods of entering a description of a desired circuit into a CAD system. This tutorial will make use of the VHDL design entry method. Using VHDL the user specifies the desired circuit using the VHDL hardware description language.

CAD or Computer Aided Design software makes it simple to implement a desired logic circuit by using a programmable logic device, such as a FPGA chip. A typical FPGA CAD flow is shown in Figure 1.

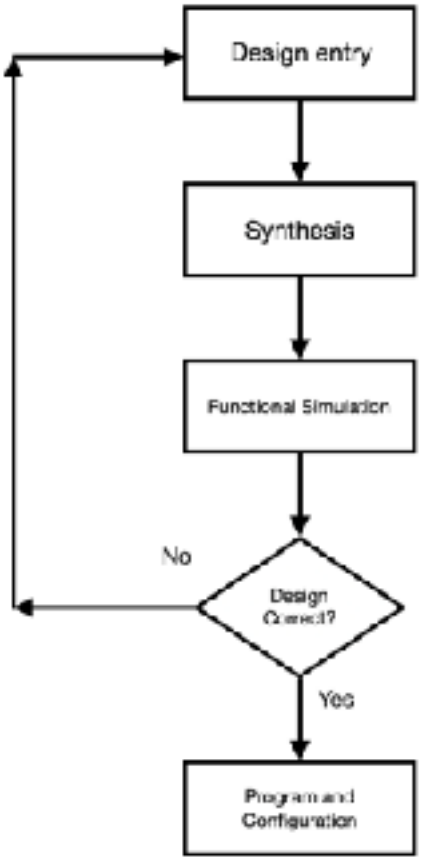


Figure 1. Typical CAD Flow

The CAD Flow involves these steps:

Design Entry-The desired circuit is specified by either using a schematic diagram, or by using a hardware description language such as VHDL or Verilog.

Synthesis - the entered design is synthesized into a circuit that is made up of logic elements (LE's) within the FPGA chip

Functional Simulation - The synthesized circuit is tested to verify that it is functional; the simulation does not account for timing issues.

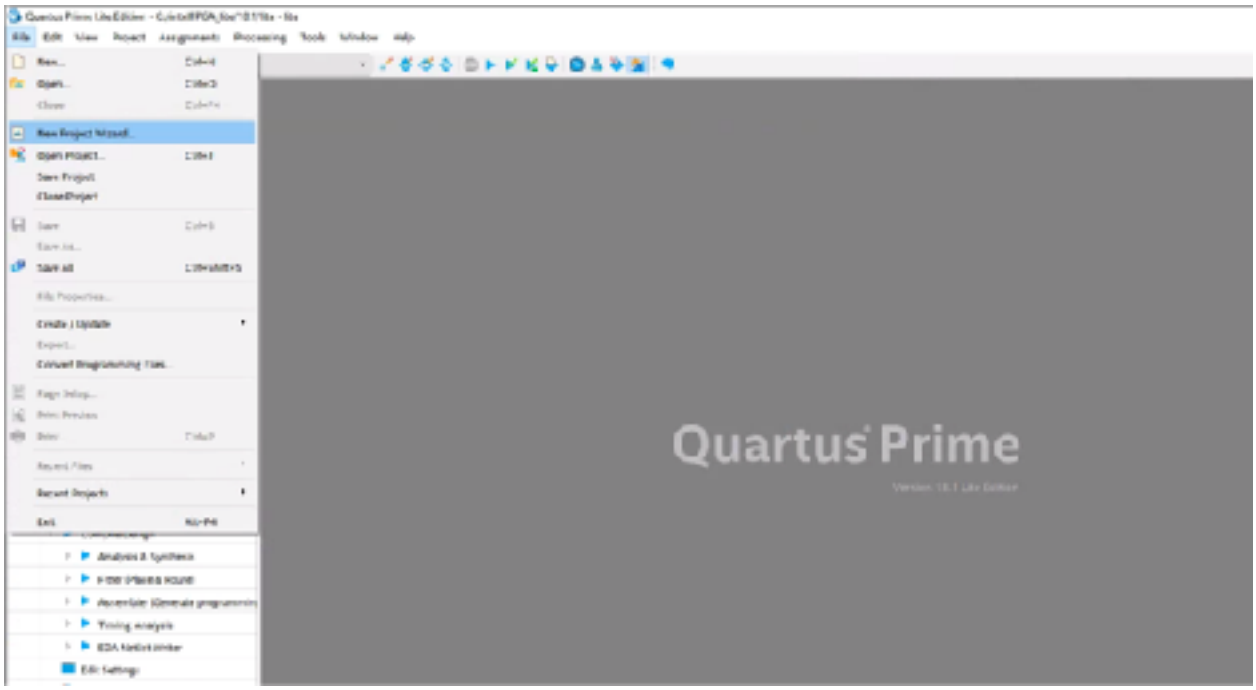
Program and configuration — the designed circuit is then implemented into the physical FPGA chip

This Lab will introduce the basic features of the Quartus prime software. It will show how the software can be used to design and implement a circuit specified by using the VHDL Hardware description language. In this lab you will learn about:

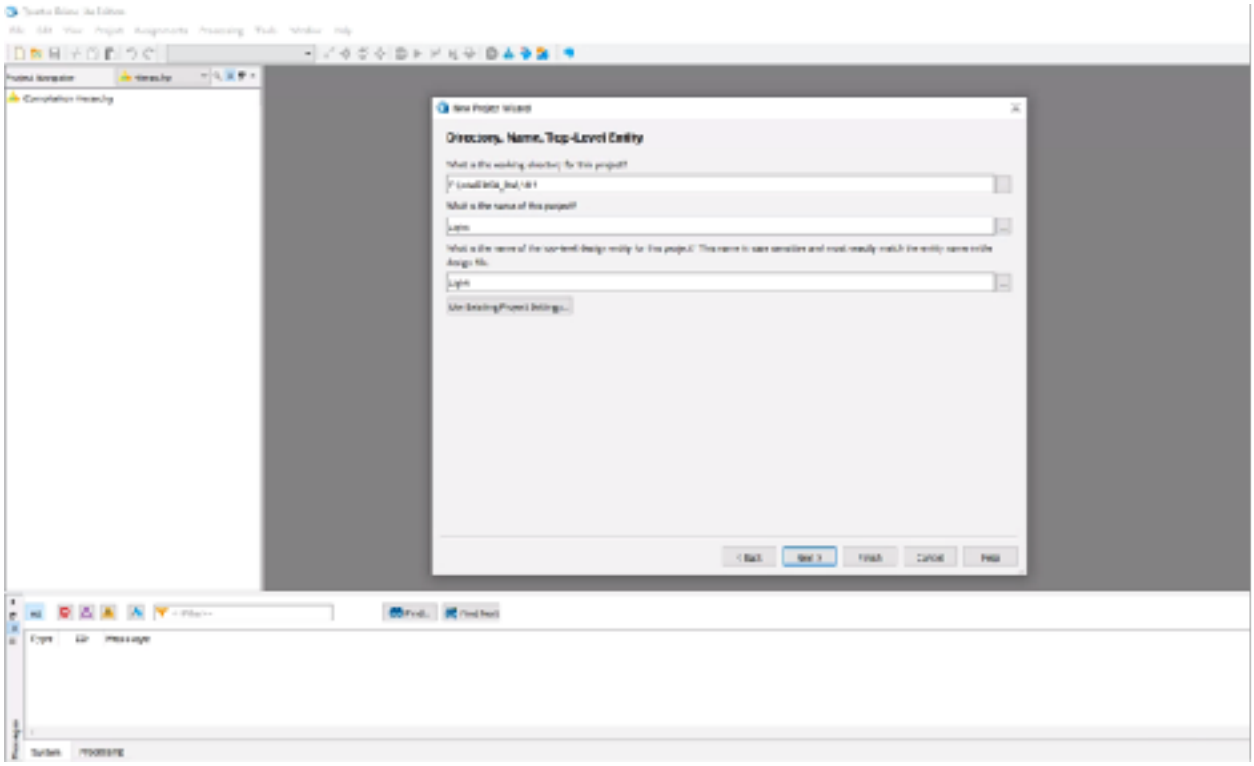
- Design entry using VHDL
- Synthesizing a circuit specified in VHDL code
- Assigning the circuit inputs and outputs to specific pins on the FPGS
- Programming and configuring the FPGA chip on the DE10 - Standard Board

Creating a new project

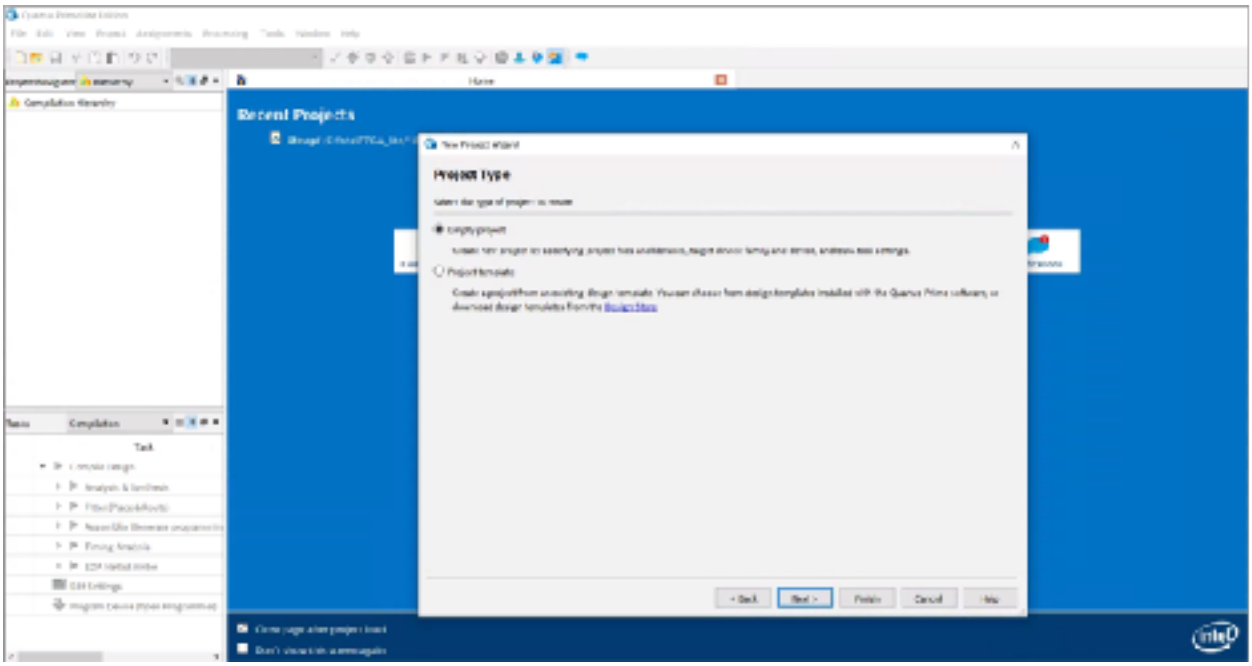
A. File > New Project wizard



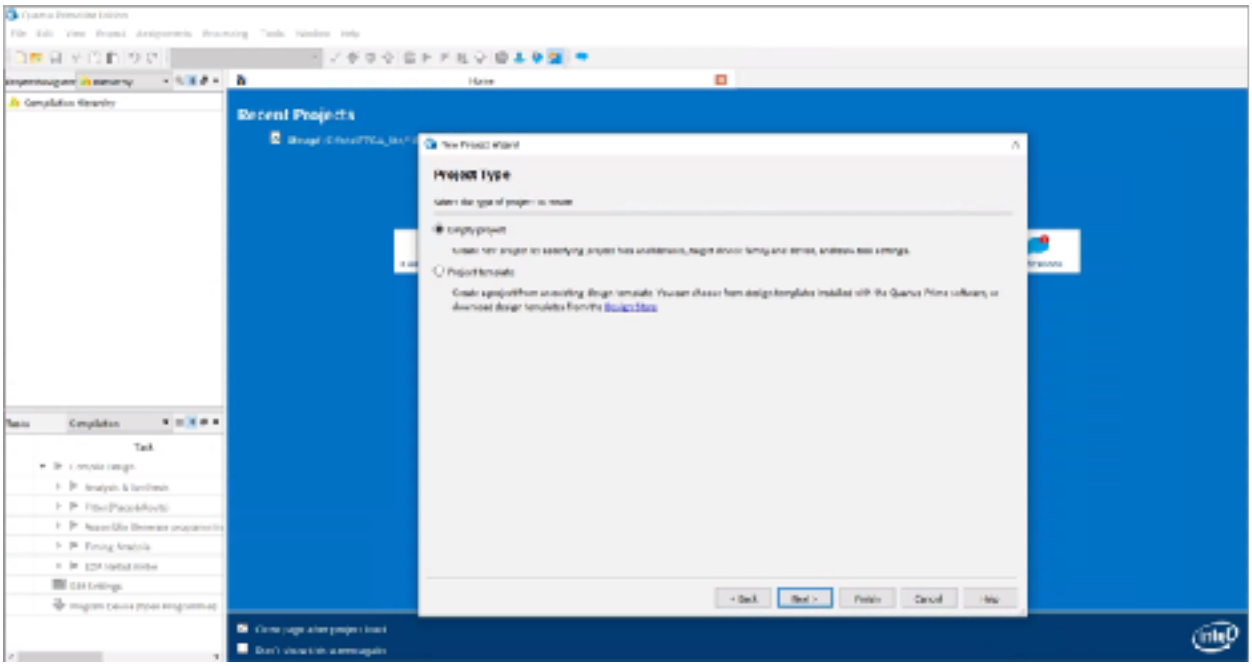
B. Name the Project and top level entity. Create a new folder and place it where you would like the project to be saved, if the project will be used again later it is recommended that you save to a flash drive.



C. Select “Empty Project” and click next

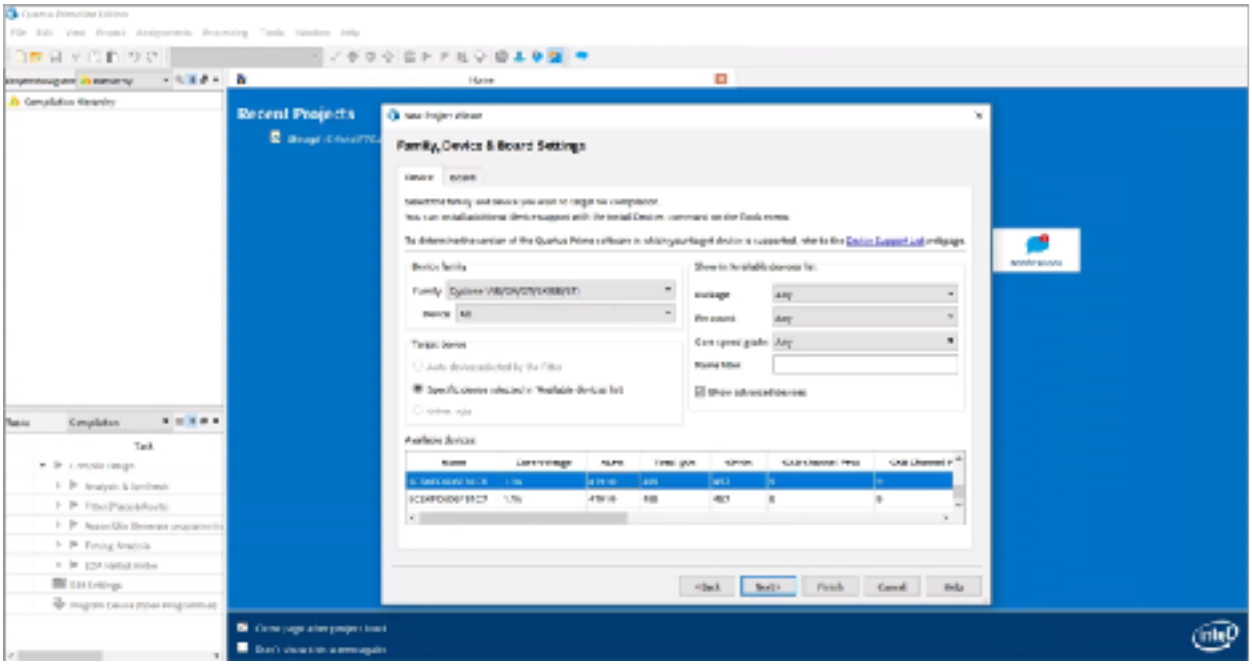


D. Click next, do not add any files to your project.

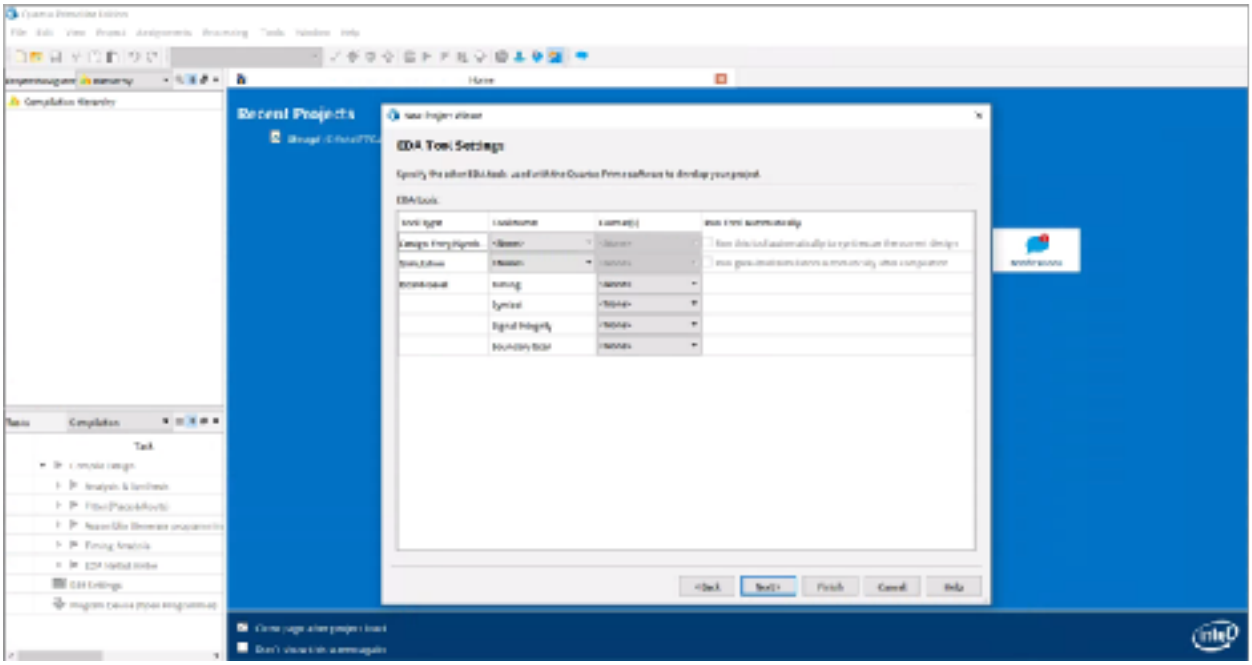


E. Select Cyclone V and choose 5CSXFC6D6F31C6, this is the device family and model number of the FPGA used in Intels DE10 standard board.

Note the model number and family name can also be viewed on the chip of your DE10

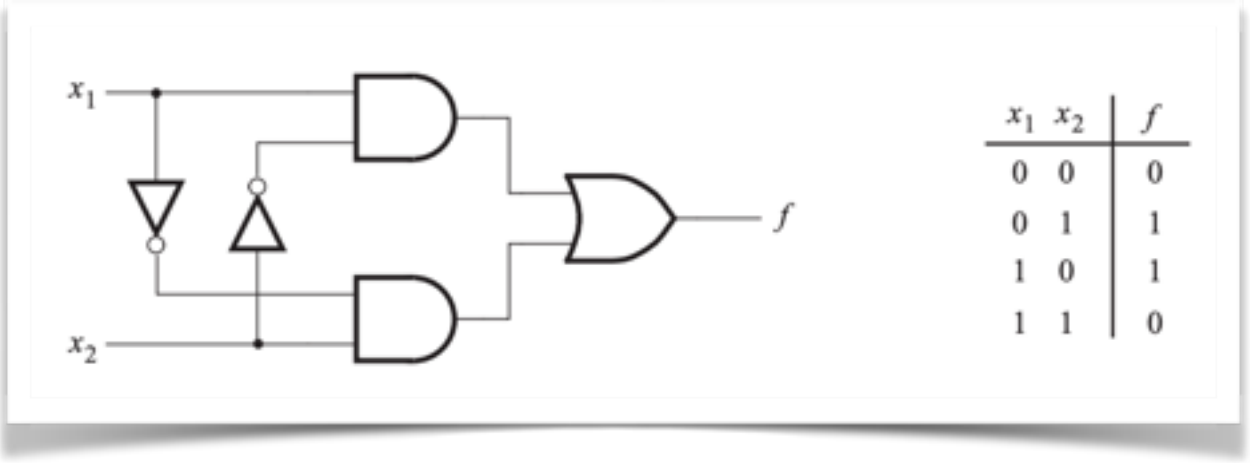


F. Click Next and Finish (do not edit any of these settings)



Design Entry Using VHDL Code

As an example, we will use a two way light controller circuit. The circuit can be used to control a single light from either of the two switches x1 or x2, where a closed switch is qual to a logic value of 1. The truth table for the circuit is also provided.



x_1	x_2	f
0	0	0
0	1	1
1	0	1
1	1	0

Light controller circuit

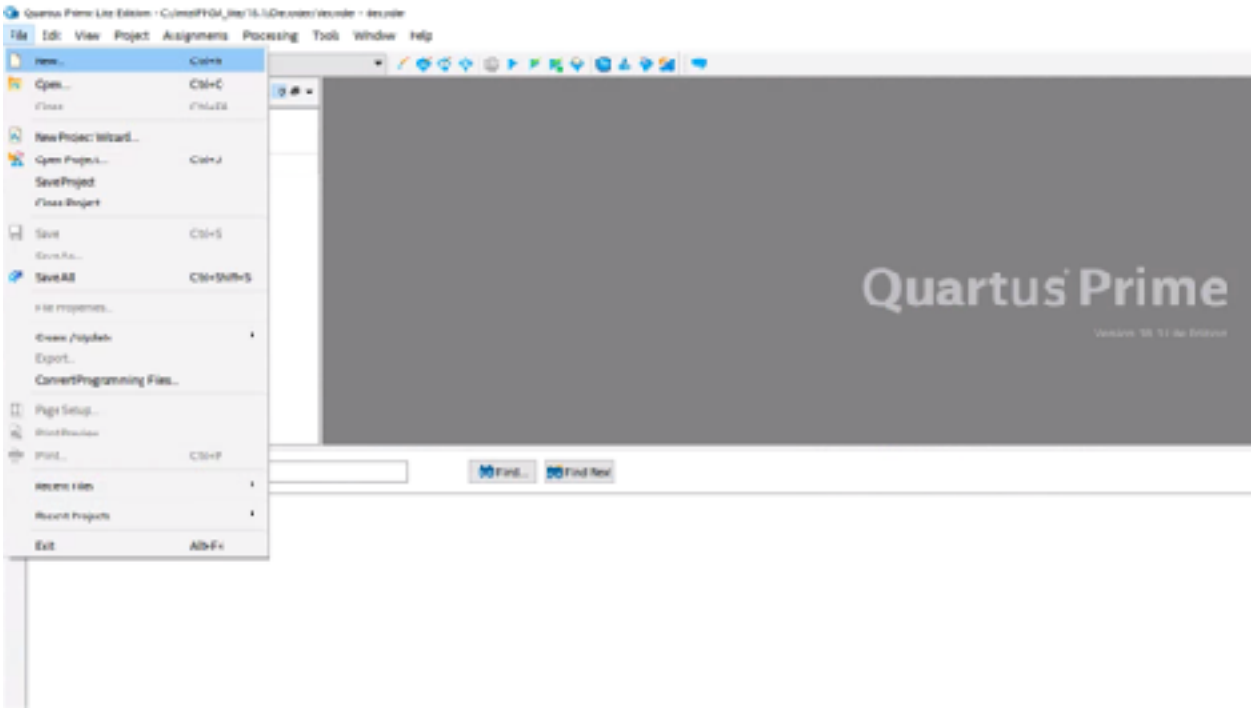
Below we can see the Light controller circuit translated into VHDL code. ***Note the VHDL entity is called light to match the name given to the project entity earlier.*** This code can be typed into a file using any text editor that can store ASCII files, or by using the Quartus prime text editors. While the file can be given any name it is common practice among designers to use the same name of the top-level VHDL entity. The file name must include the extension .vhd which indicates a VHDL side so we will use the file name " light.vhd "

```
LIBRARY ieee ;
USE ieee.std_logic_1164.all ;
ENTITY light IS
        PORT ( x1, x2 : IN STD_LOGIC ;
                f : OUT STD_LOGIC ) ;
END light ;
ARCHITECTURE LogicFunction OF light IS
BEGIN
        f <= (x1 AND NOT x2) OR (NOT x1 AND x2);
END LogicFunction ;
```

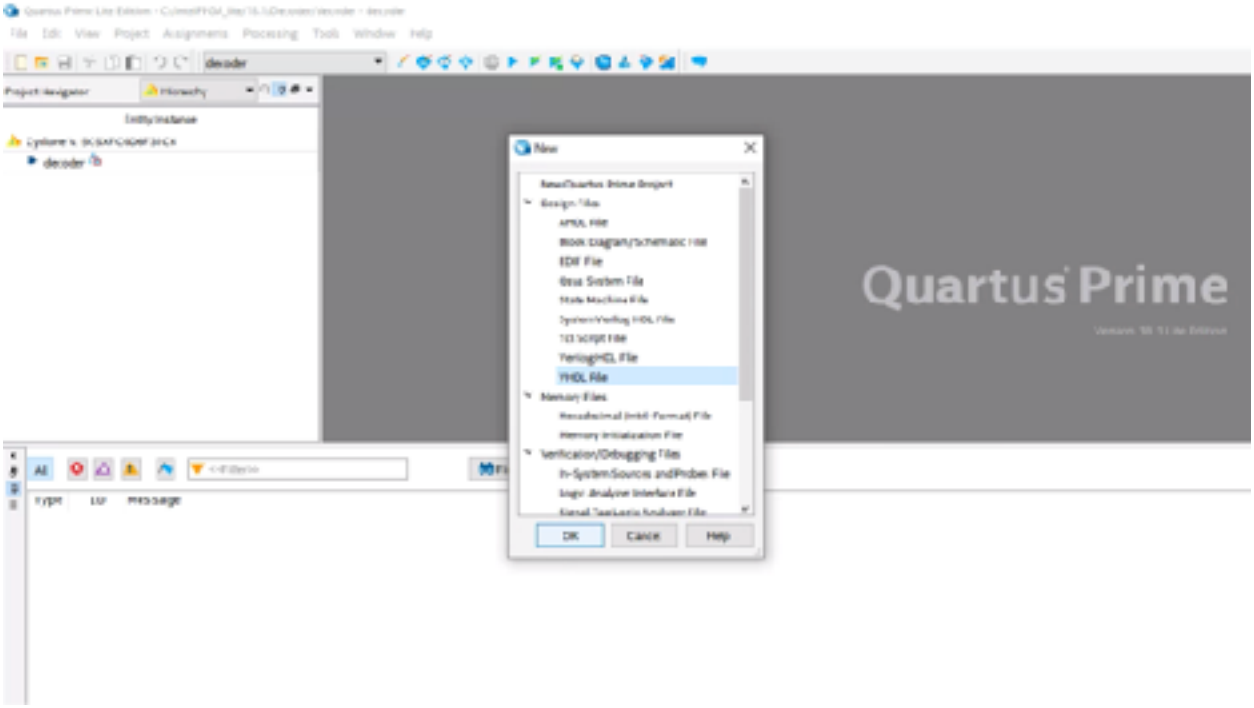
Using The Quartus Prime Text Editor

This section will show you how to use the Quartus Prime text editor to create a VHDL source code file.

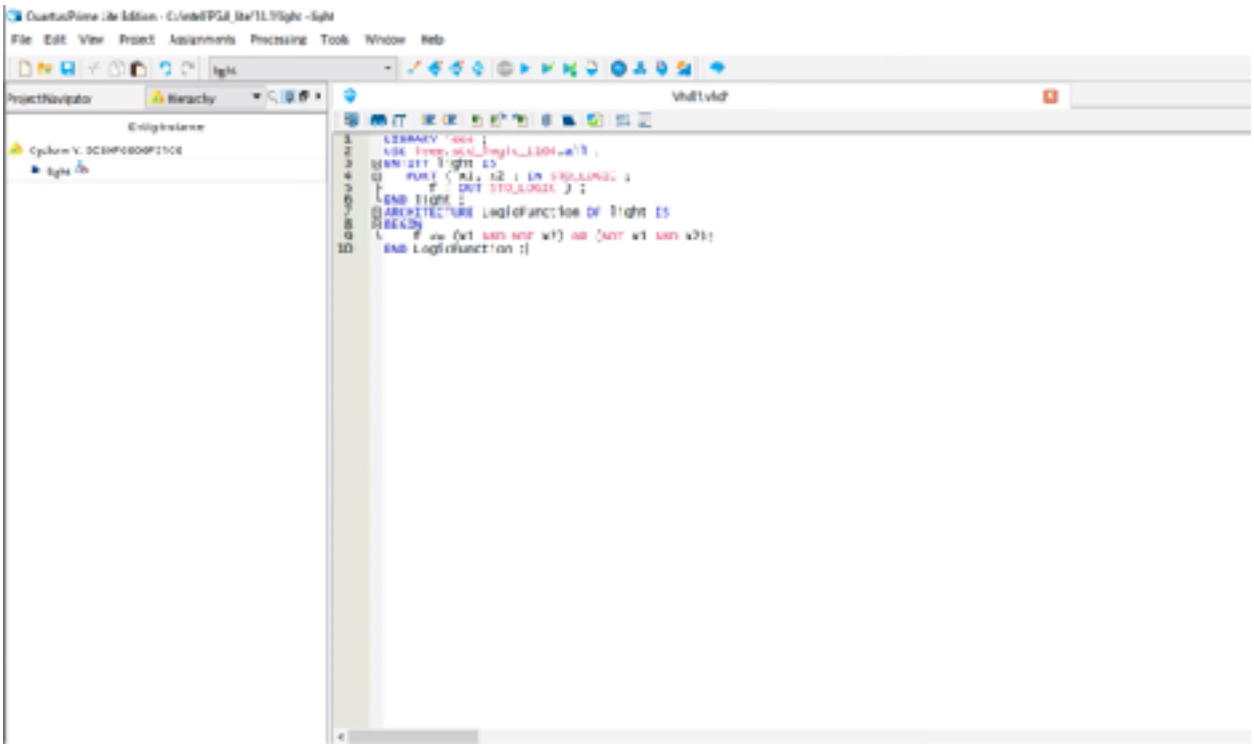
A. Go to File > New



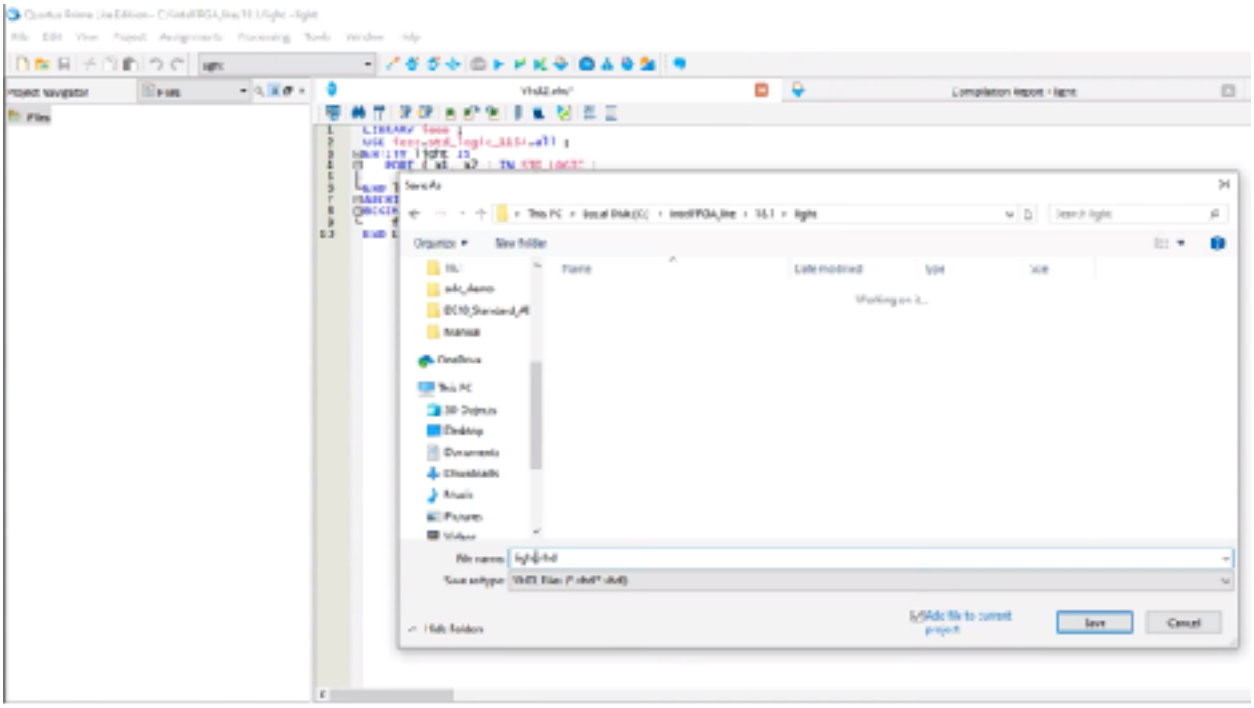
B. Highlight VHDL File Under Design files and Click OK



C. You are then brought to the quartus prime text editor. This is where you will write your VHDL code, in this project you will be using the code provided earlier with the light controller circuit.



D. Go to File > Save As and save the file as light.vhd in your project folder. Be sure to always name your file the same as the entity name or your program may not compile.

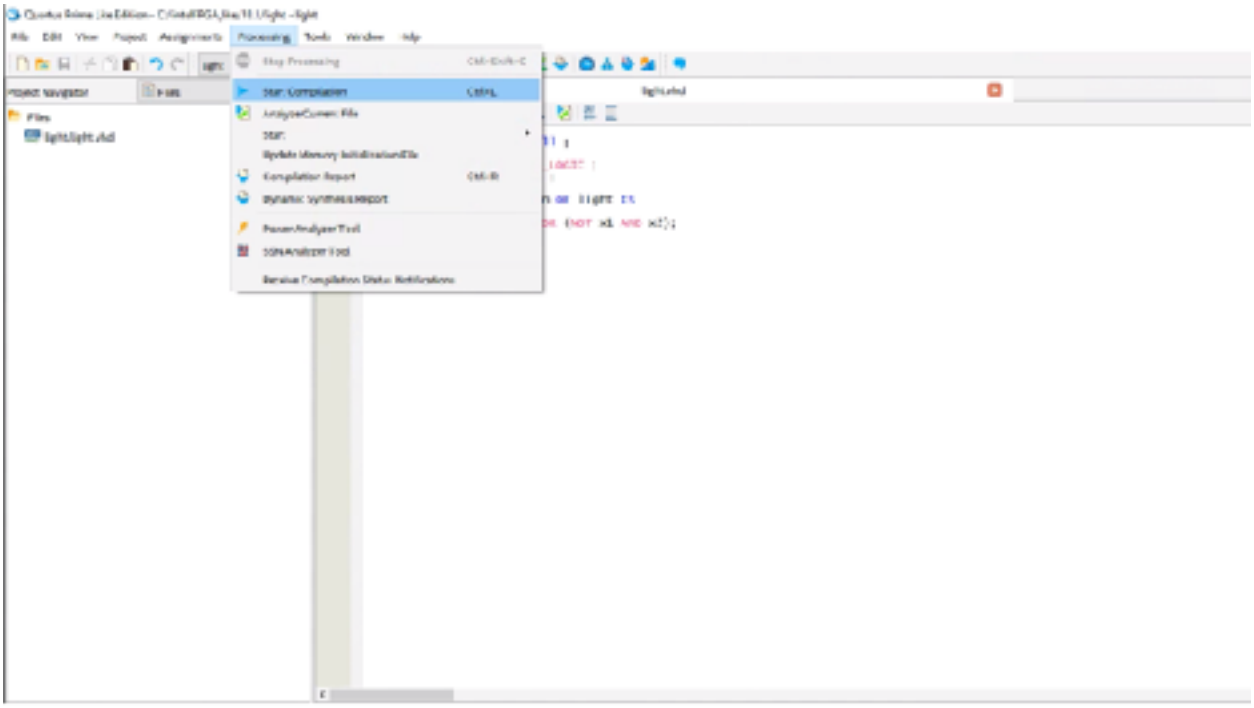


Compiling The Designed Circuit

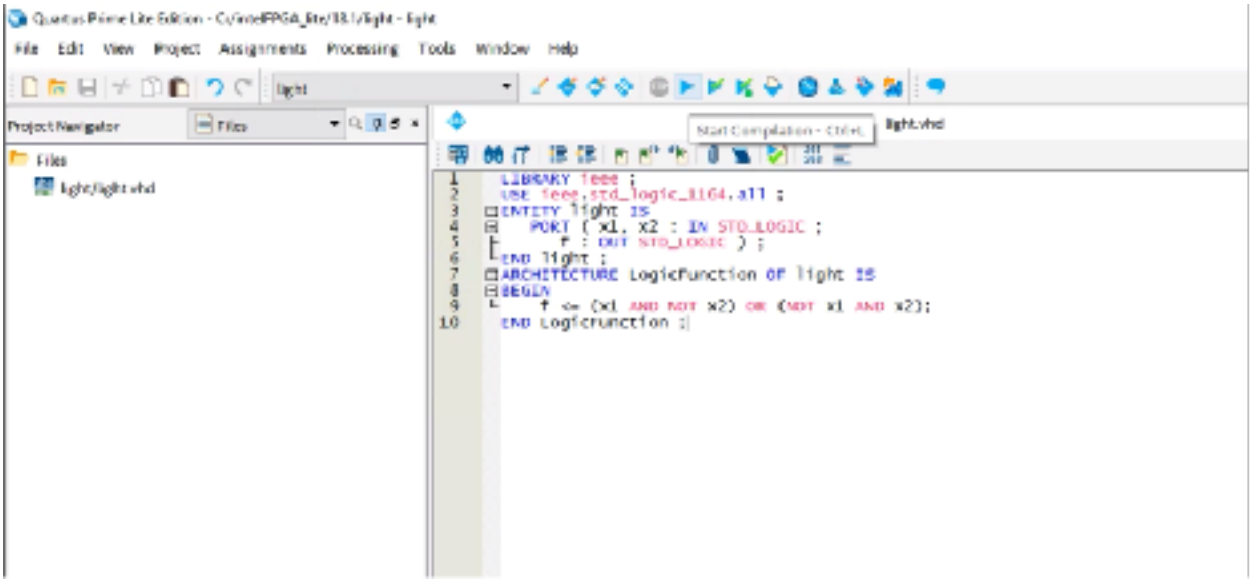
The created VHDL code in the light.vhd will need to be processed by several tools within the Quartus prime program that analyze the code, synthesize the circuit and generate a implementation for the specific chip that was chosen, in our case the Cyclone V/ 5CSXFC6D6F31C6. These tools are controlled using an application program called the compiler.

To run the compiler you can

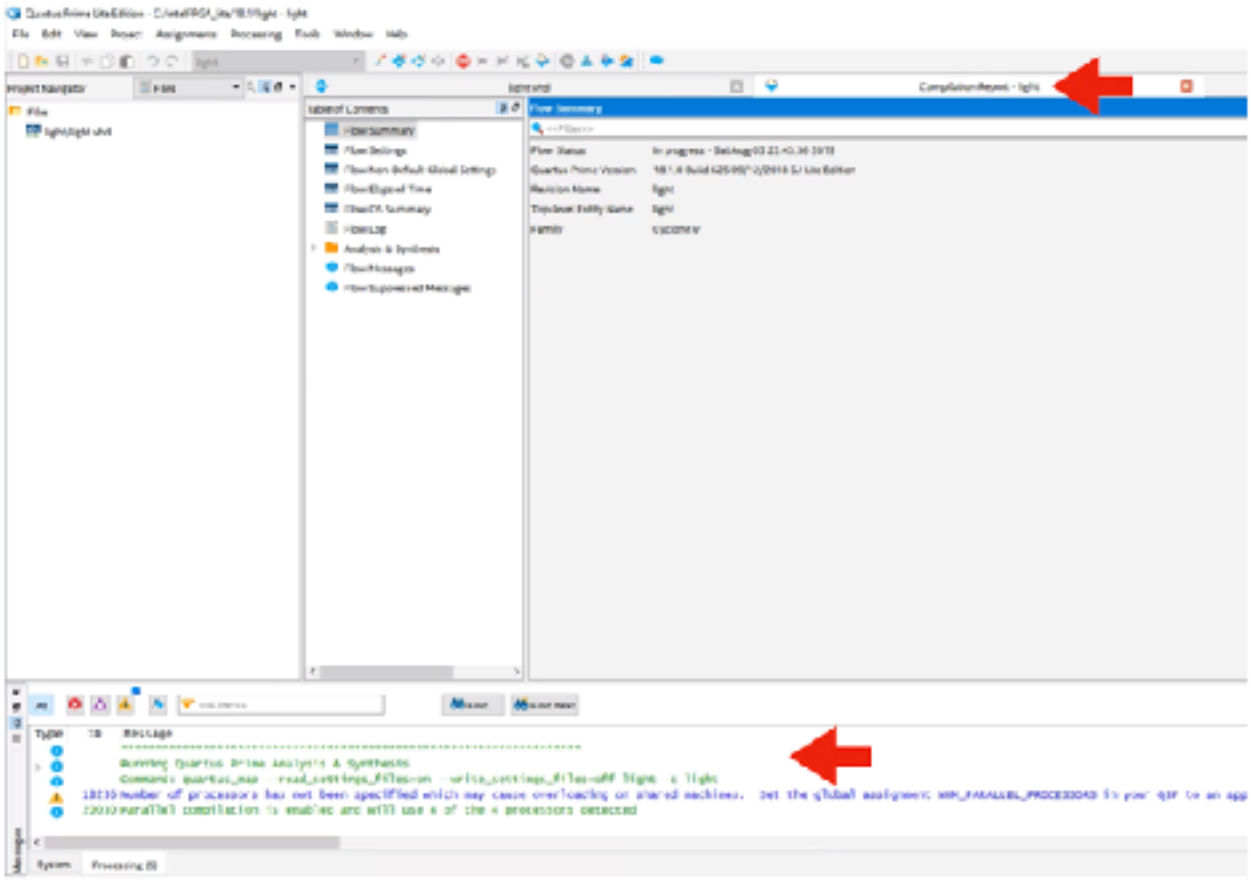
Go to Processing > Start compilation (Ctrl + L)



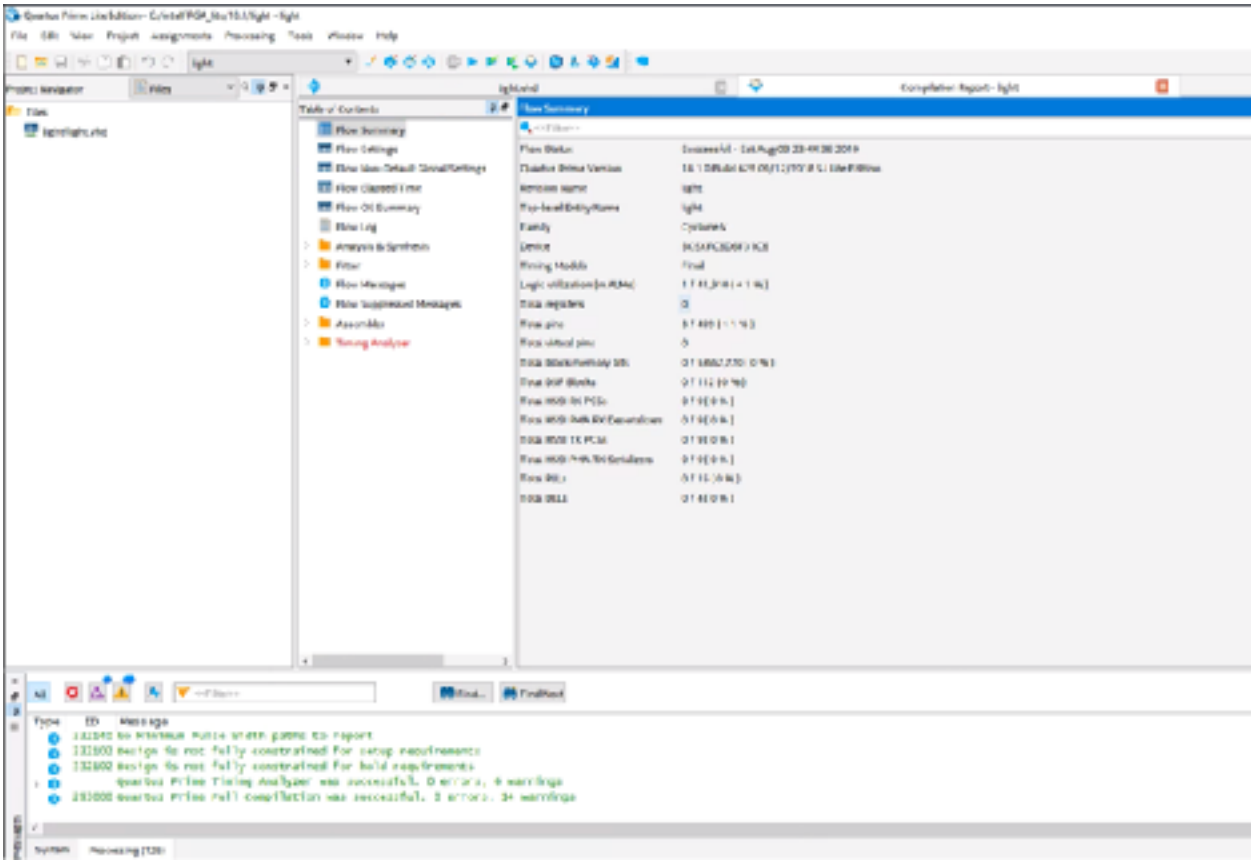
Or Click on the Start compilation button in the Quartus toolbar



Run the compiler for your light.vhd file. As the program is being compiled, it will provide you with a compilation report in a new window and report its progress in the bottom window of the program in the processing tab.



If your program successfully compiled the program will show successful in the flow status of the compilation report and in the processing message box. If your compilation was unsuccessful, Check the error report for any syntax errors such as a misspelled word or missing semicolon. Double clicking on the error will usually take you to the line of where the error occurred.

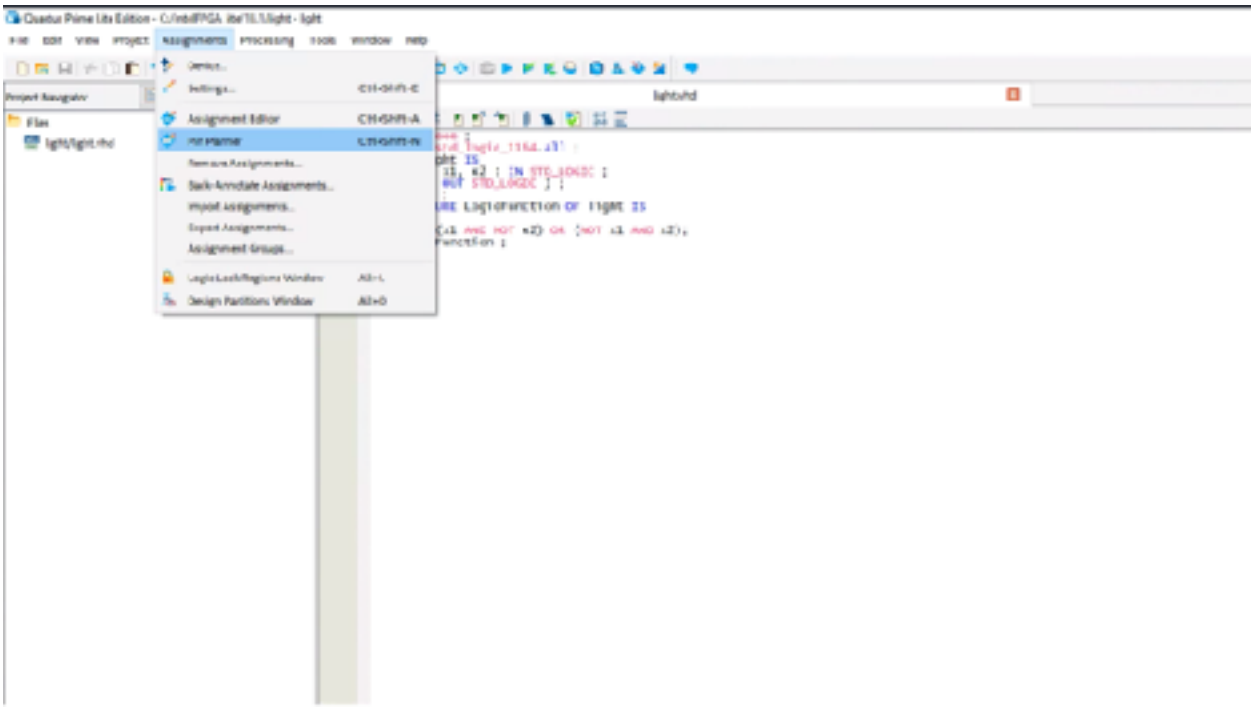


Pin Assignment

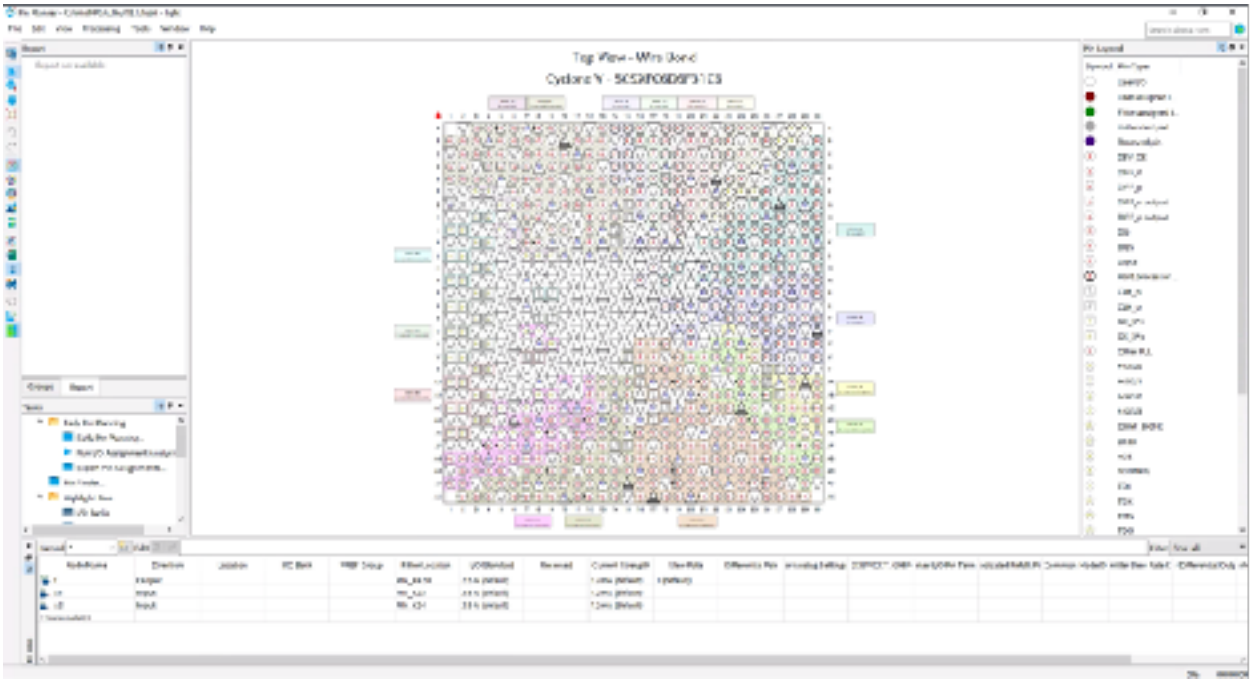
During the compilation, the Quartus Prime compiler choses any pins within the FPGA to emulate the inputs and outputs. However the DE10 board has hardwired connections between the FPGA pins and other components as the board such as the switches and GPIO pins. For this lab we will use two toggle switches labeled SW0 and SW1 to provide external inputs x1 and x2 used in our code. These switches are connected to FPGA pins PIN_AB30 and PIN_Y27, respectively. We fill connect output f to an onboard LED labeled LEDR0 on our board. The LED is hardwired to FPGA pin PIN_AA24.

Pin assignments are made by using the pin planner.

Go to Assignments > Pin Planner (Ctrl + Shift + A)

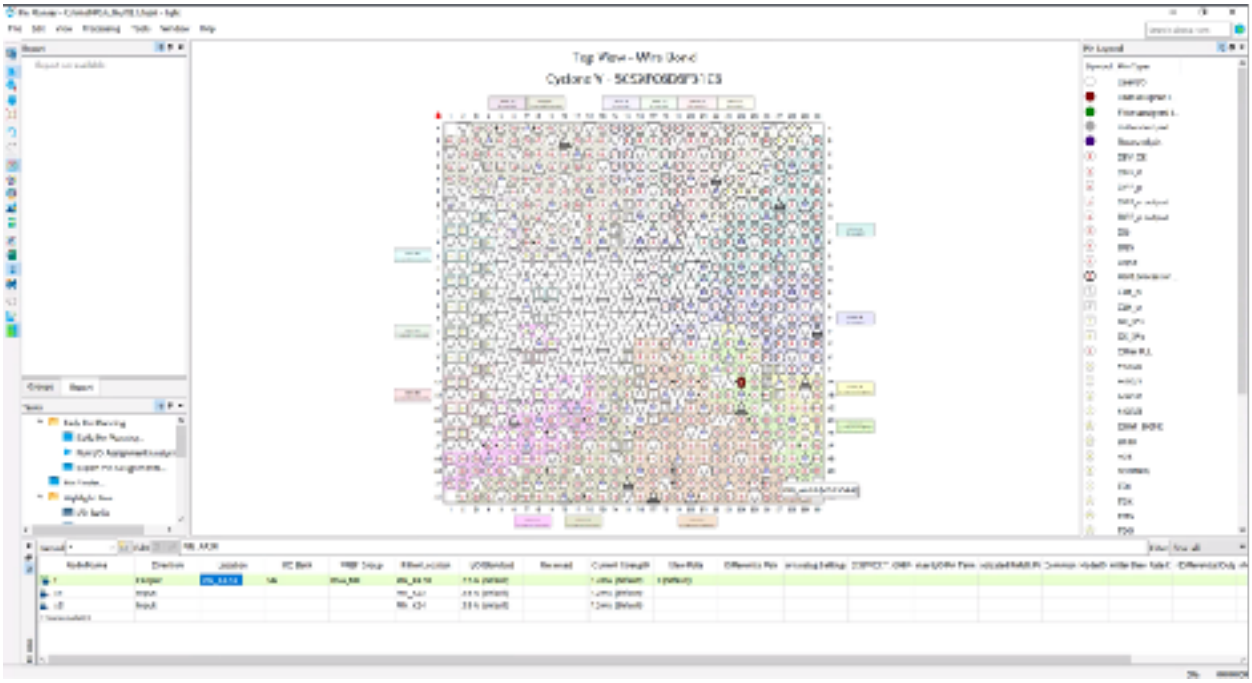


After clicking on pin planner it will open a new window. In the pin planner window, all the hardwired connections the FPGA has within the board are displayed, as well as the nodes that were created during the compilation of the VHDL program. Nodes are the inputs and outputs that are declared in your program. In this case the inputs are x1, x2 and the output is f.



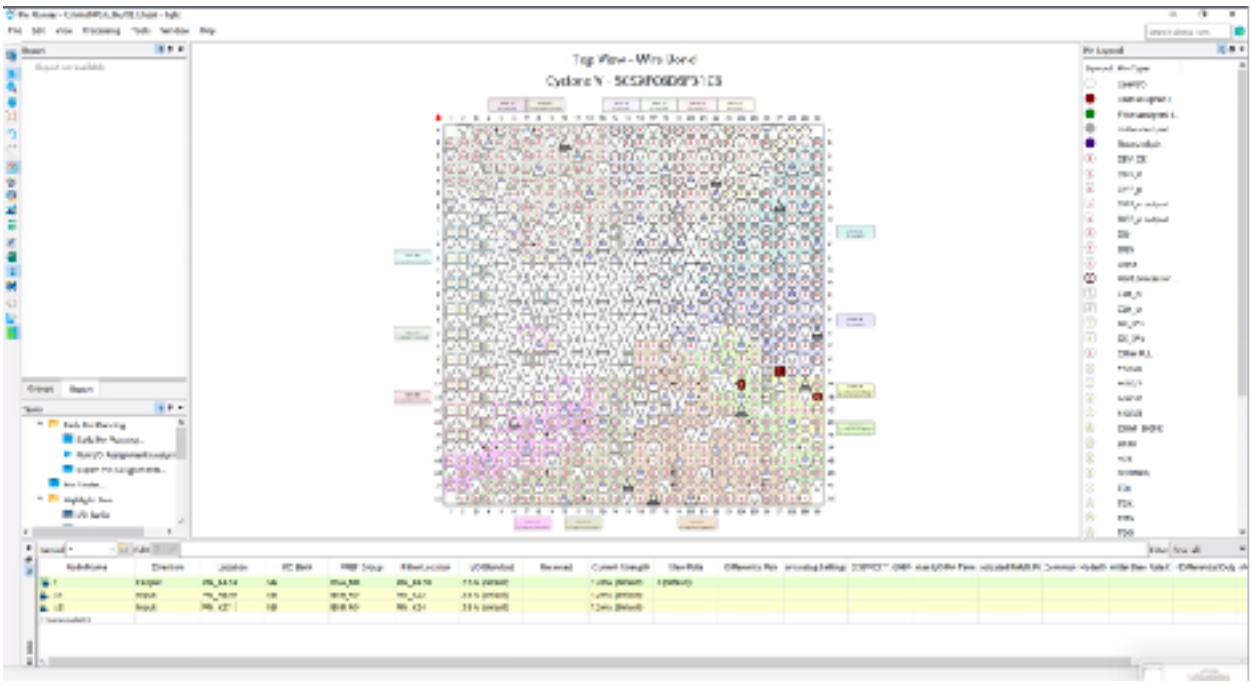
To assign the nodes to pins we will need to change the location in the bottom portion of the window.

Click on the Location of node f, here we can assign the output to LED0 by typing in PIN_AA24.



Do not edit any of the other settings as they are automatically generated by the pin planner.

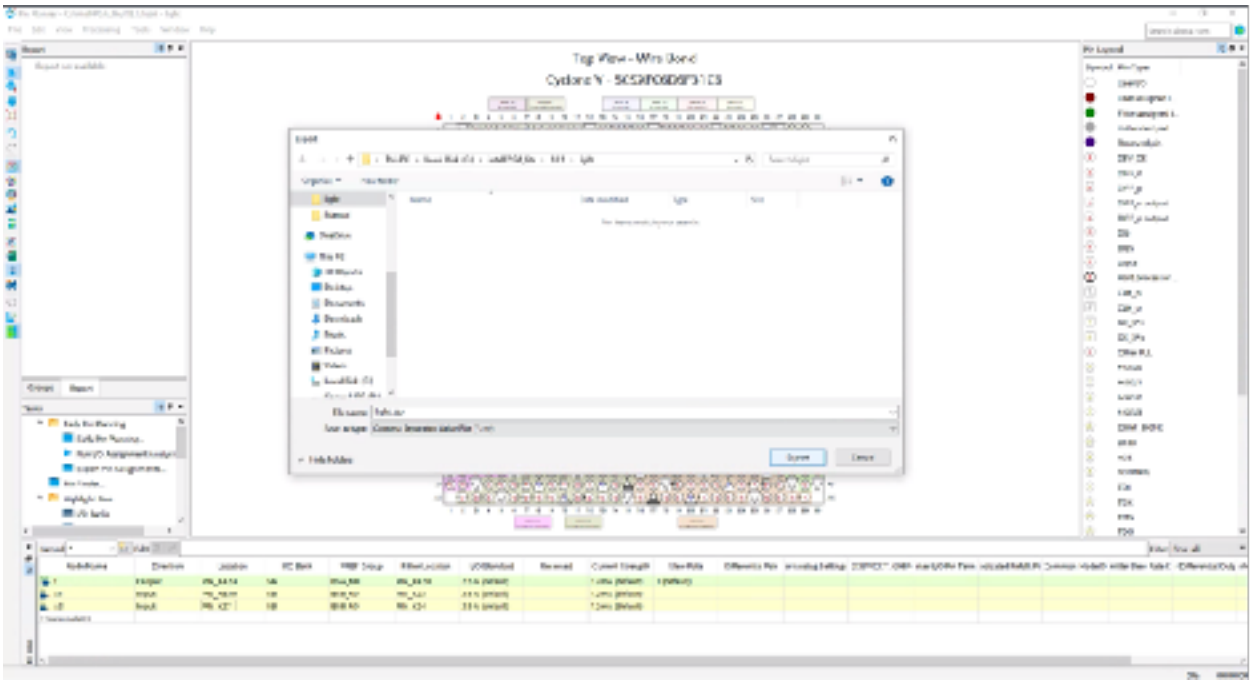
Repeat this step for assigning PIN_AB30 and PIN_Y27 to inputs x1 and x2 respectively.



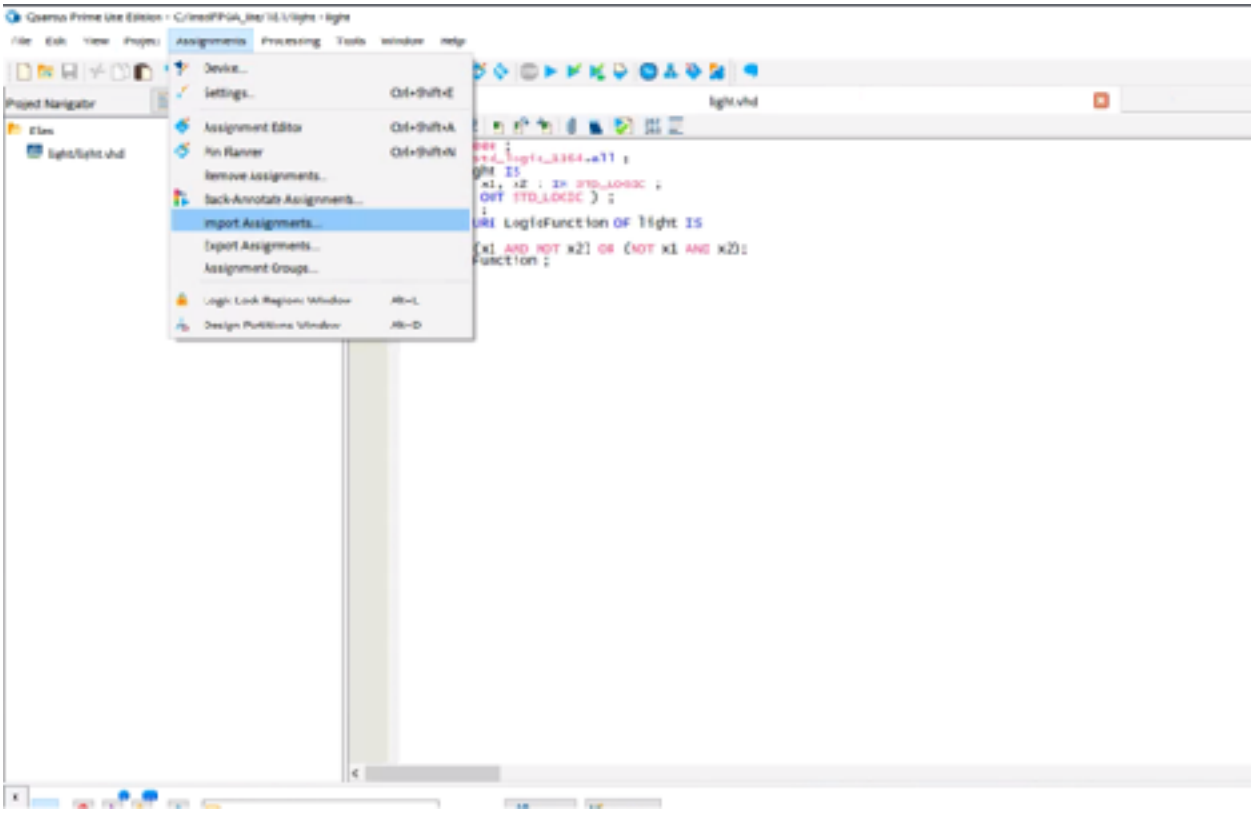
After creating the pin assignments you will need to export them into the main Quartus Prime program. The quartus prime program has a useful feature that allows the user to easily export and import pin assignments using the CSV(comma separated value) format, which are common text file formats that contain comma-eliminated values. This file format is often used alongside spreadsheet programs but can also be created using ASCII text editors. The format for our simple project is

To, Location
x1, PIN_AB30
x2, PIN_Y27
f, PIN_AA24

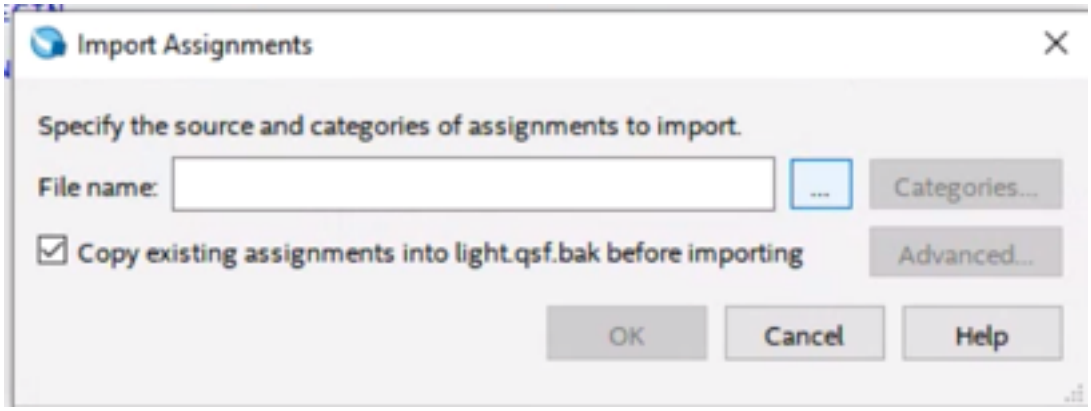
To export your pin assignments file go to File > Export, choose the folder where your project is stored and click on Export. The Files will be saved as light.csv.



You can import them into the main Quartus program by going to Assignments > Import



In the pop up window click on … , and import the light.csv file that was just created and click OK.



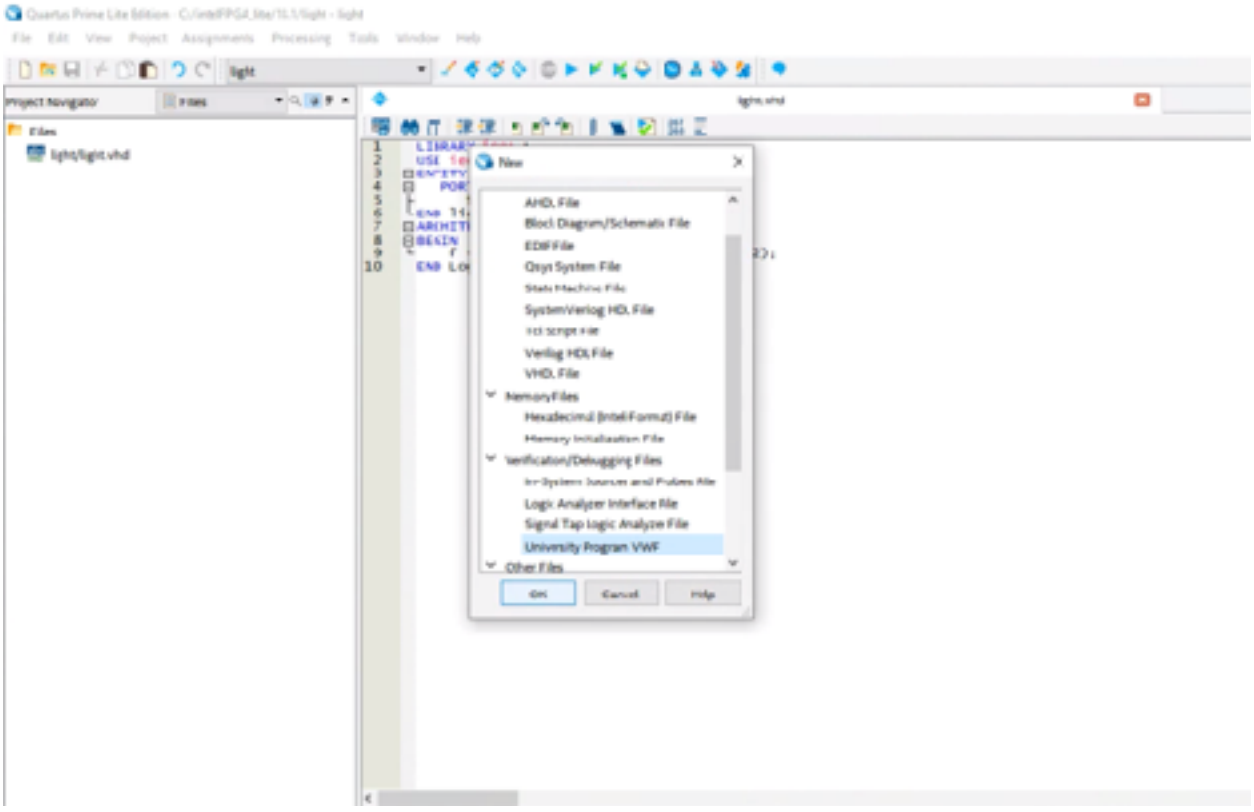
For convenience when creating large designs all relevant pin assignments for the DE10 standard board can be found within the DE10 standard cd which can be downloaded from the terasic website. The file uses the names found in the DE10 standard user manual. If we were to create pin assignments for our circuit by importing this file we would have to use the same names in out VHDL design file, SW(0), SW(1), and LEDR(0) for x1, x2, and f respectively.

Simulating the designed circuit

Before implementing the designed circuit in the FPGA chip on the DE10 standard board, it is important to simulate the circuit to assure it is correct. The Quartus prime software includes a simulation tool that can be used to simulate the behavior of the designed circuit.

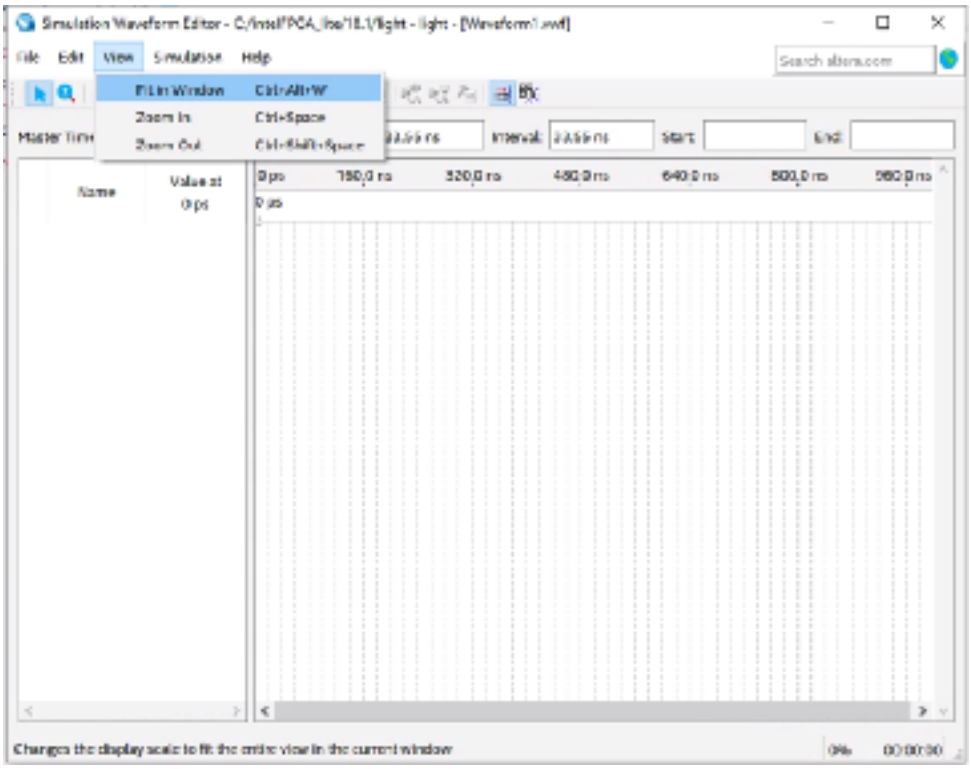
To simulate a circuits waveform

A. Go to File > New > Verification/Debugging Files > University Program VMF and Click OK

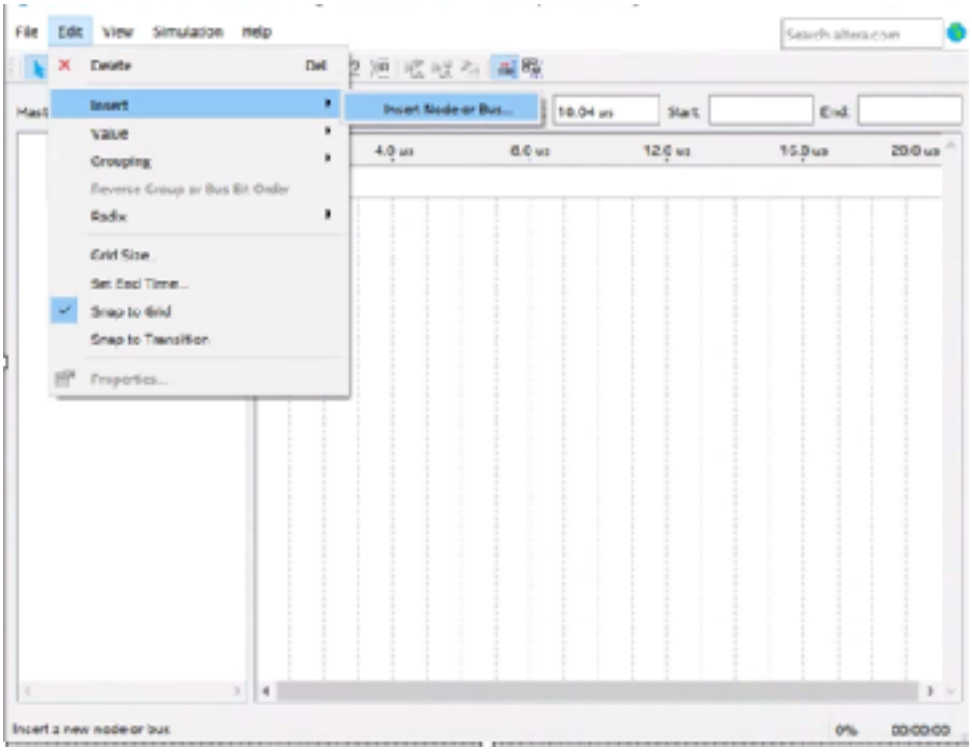


B. Save the WaveForm file as light. It is recommenced that you save this file to the same location as your project solution.

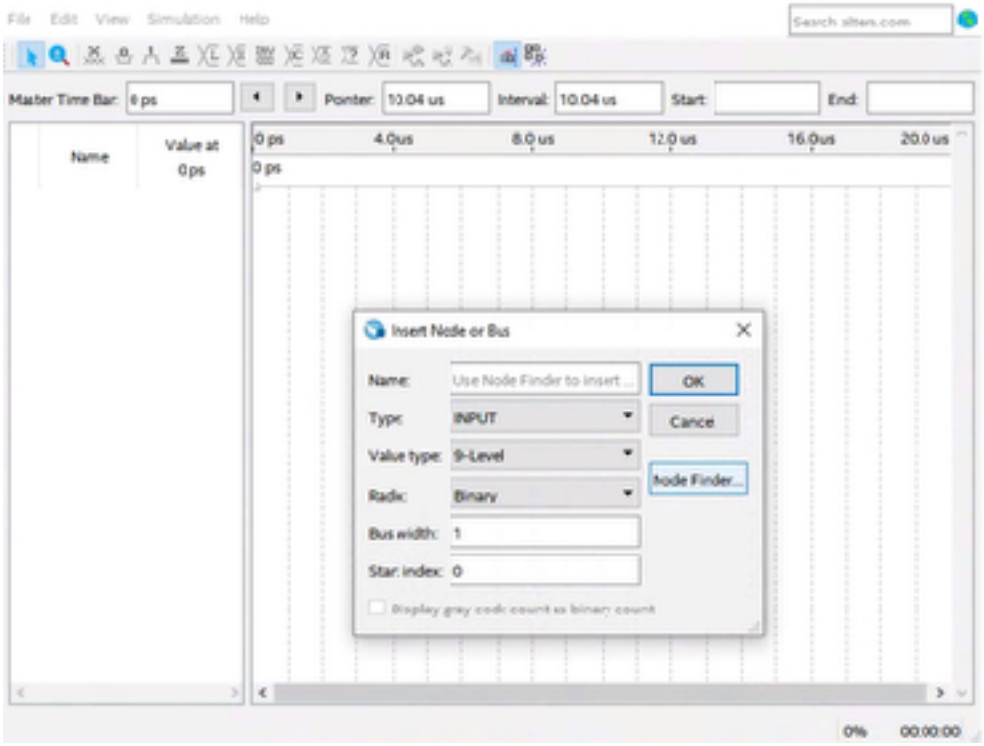
C. In the simulation wave form editor go to View > Fit To window and set the End Time to 20us by going to Edit > Set End Time



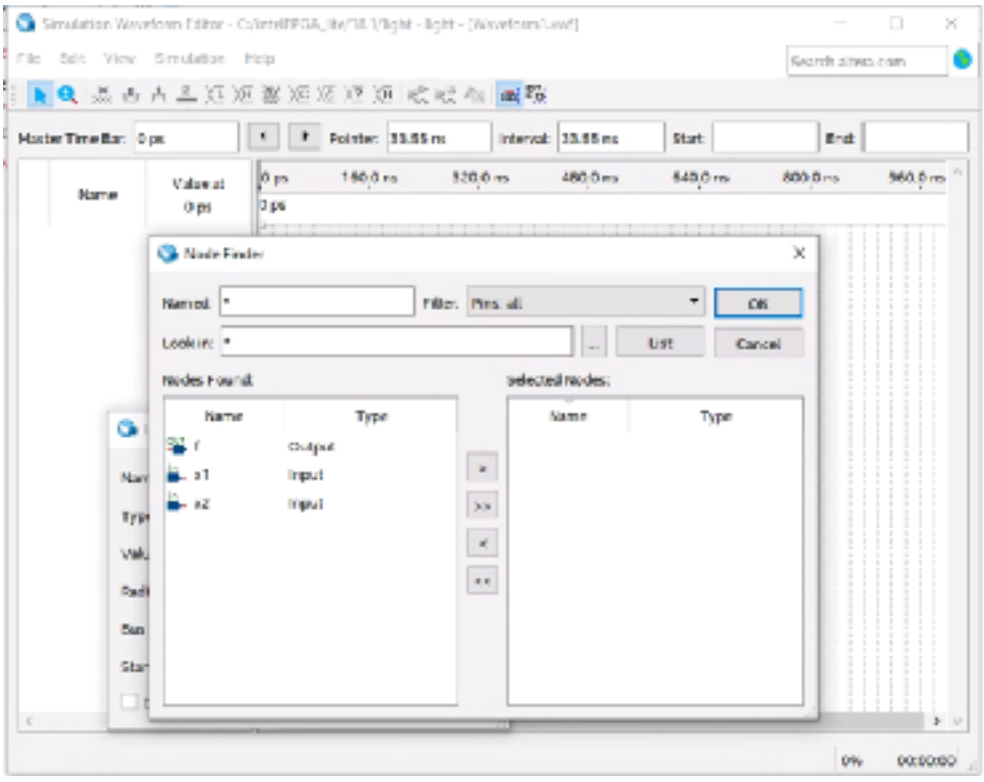
D. Go to Insert > insert Node or Bus



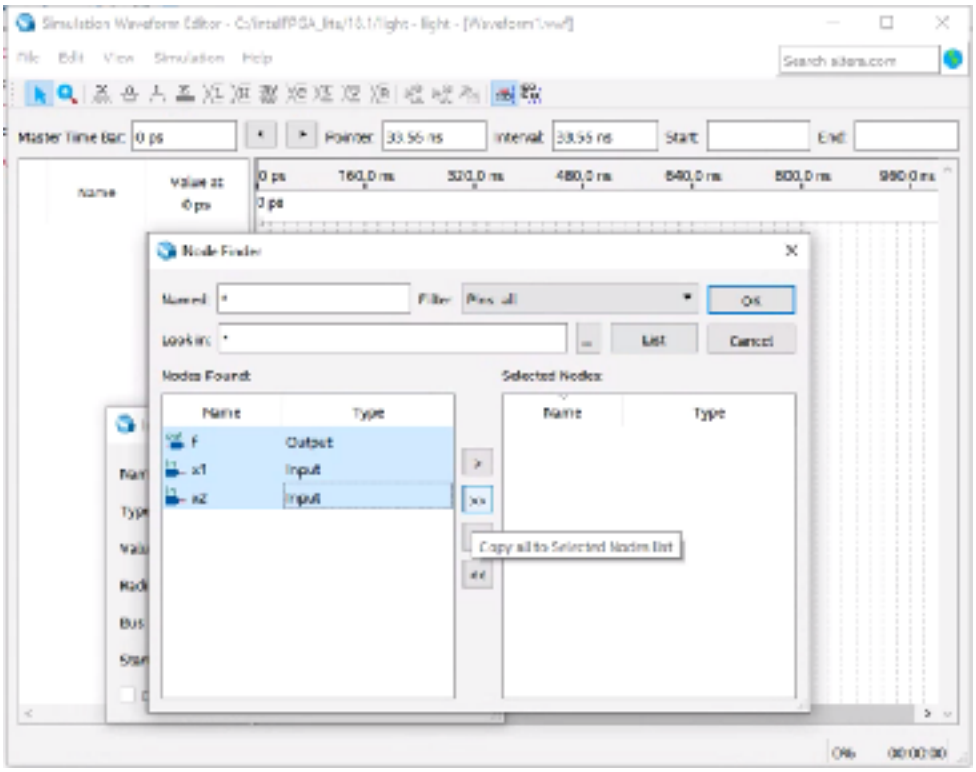
E. Click On Node Finder



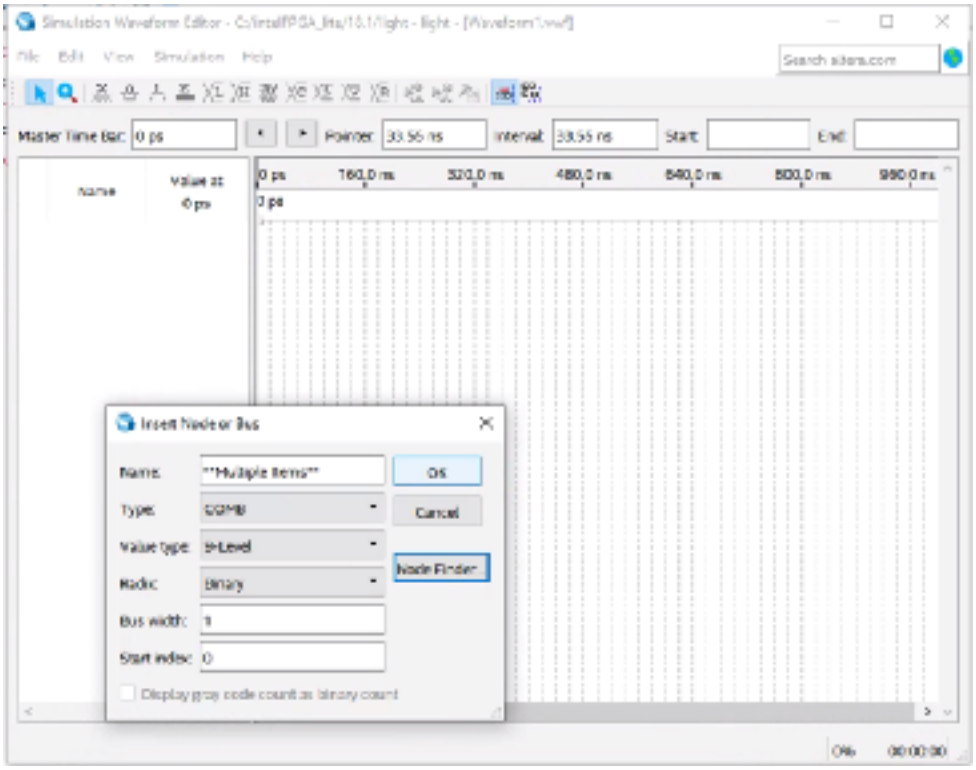
F. Select Filter : Pins : All and click On List



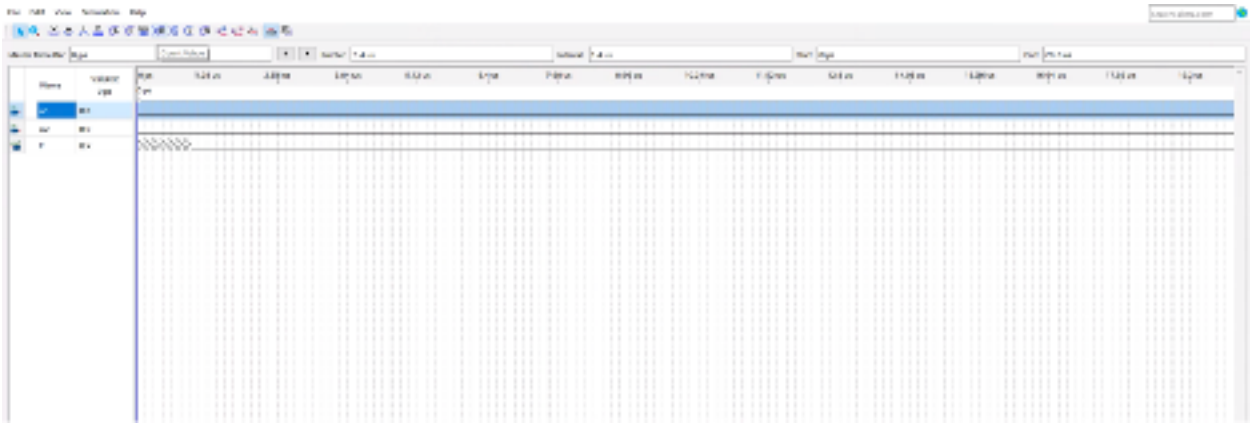
G. Select all the nodes by **Clicking on >>** and **click OK**



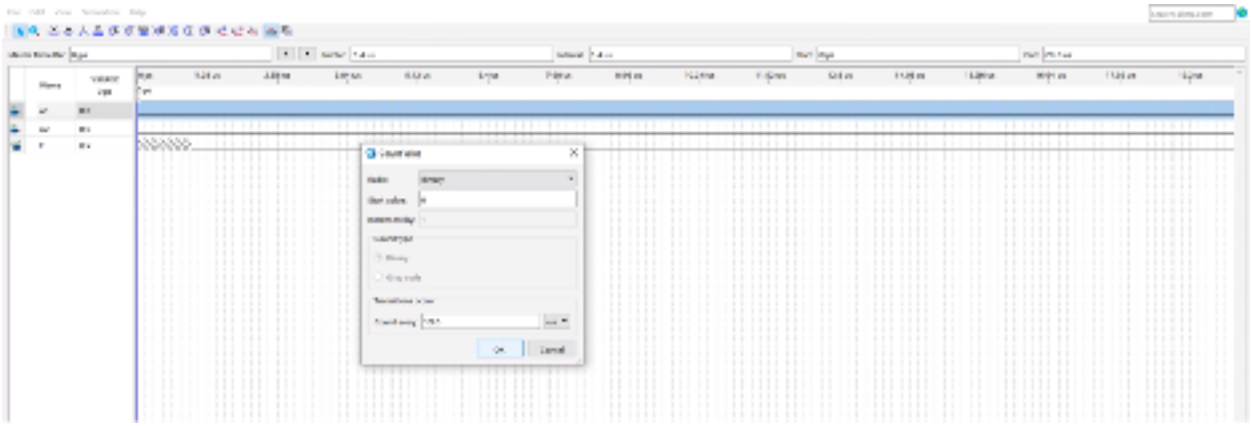
H. Click on OK and OK again, you should see **Multiple Items under name and type these are the inputs and outputs from your VHDL code



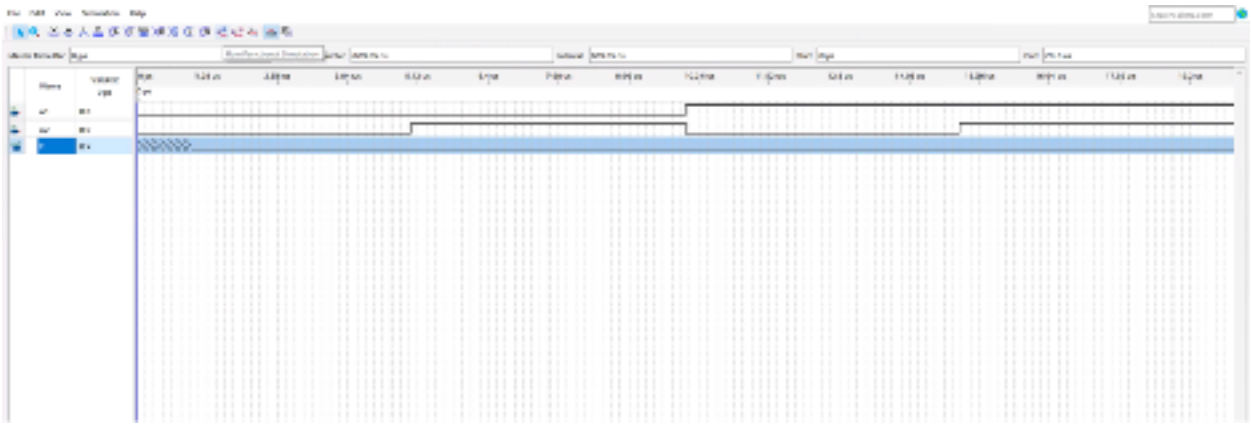
I. Highlight x1 and Go to count value in the simulation waveform editor tool bar.



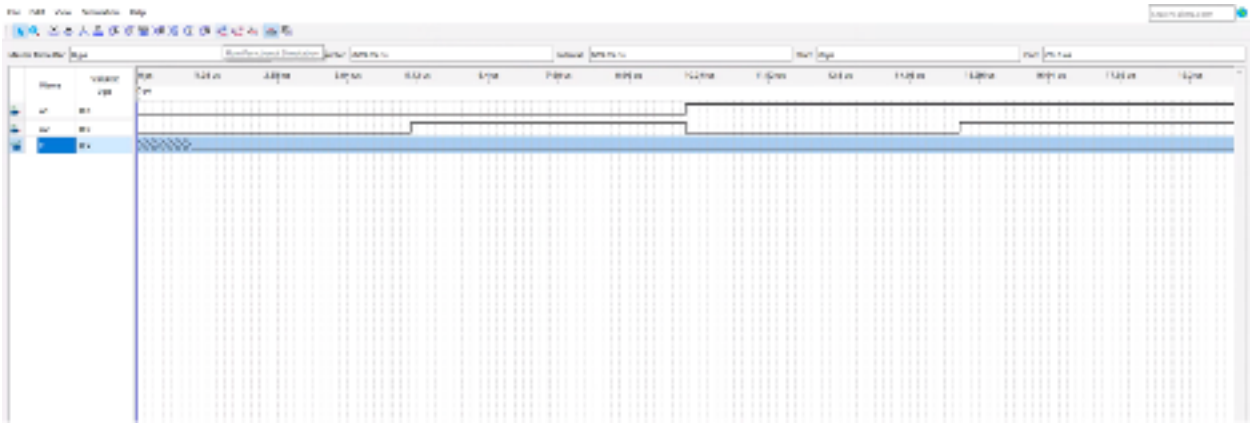
Set the Radix to Binary, start value 0, and count every 10us



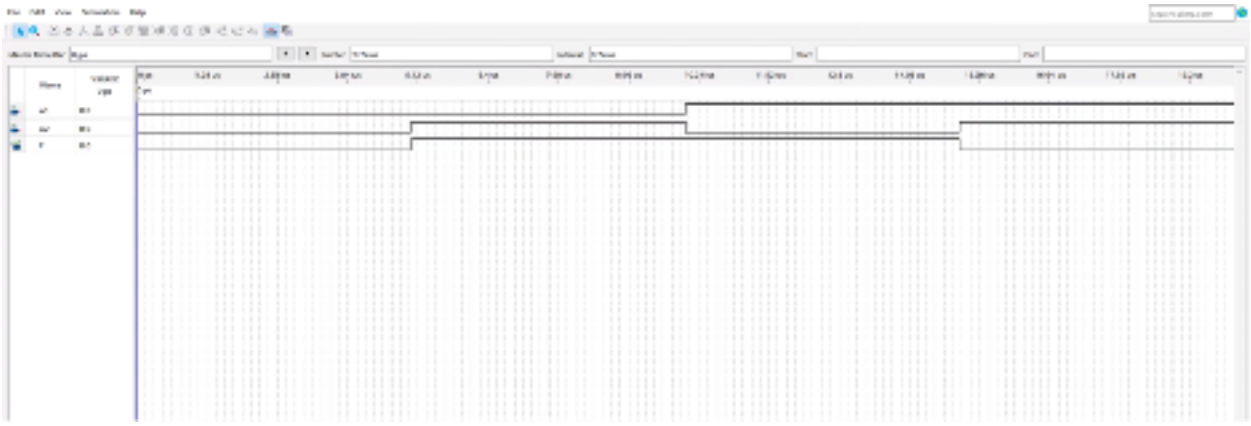
Repeat the step for x2 and set it to count every 5us



Now run the simulation using the run functional simulation button in the simulation waveform editor toolbar.



Your final result should resemble the wave form below

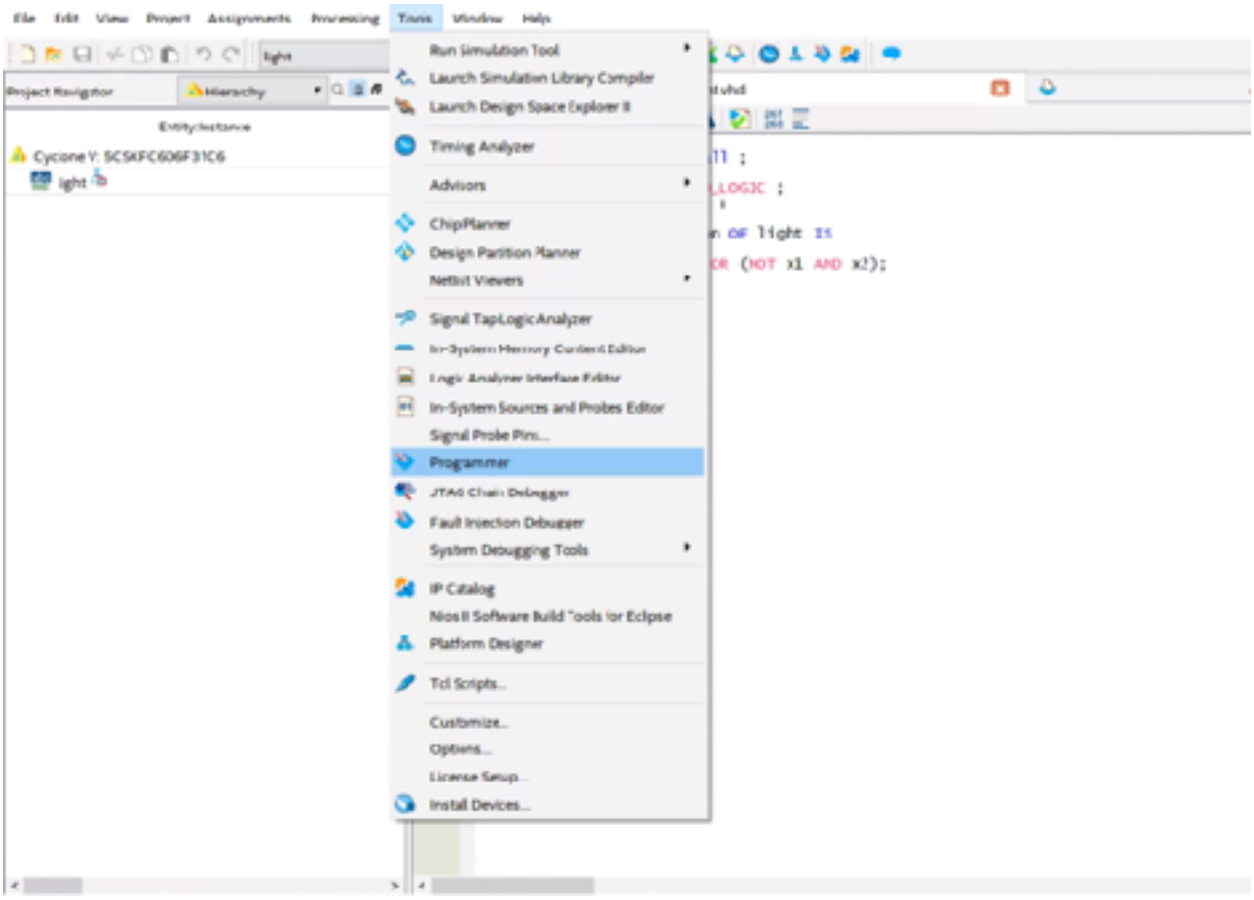


Programming the FPGA

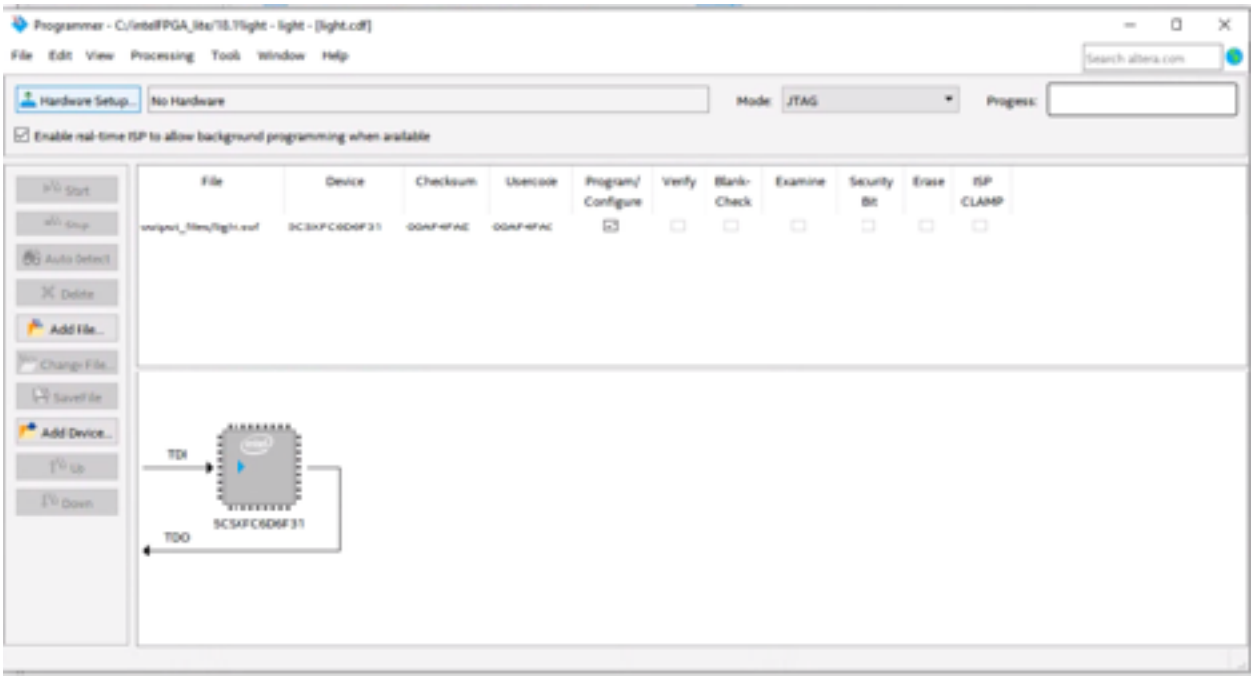
The FPGA device must be programmed and configured in order to run the designed circuit. The required configuration file is generated by the Quartus Prime compiler. To configure our board we will using a USB-Blaster connected to the board to transfer the data and configuring it using the JTAG programming.

JTAG Programming

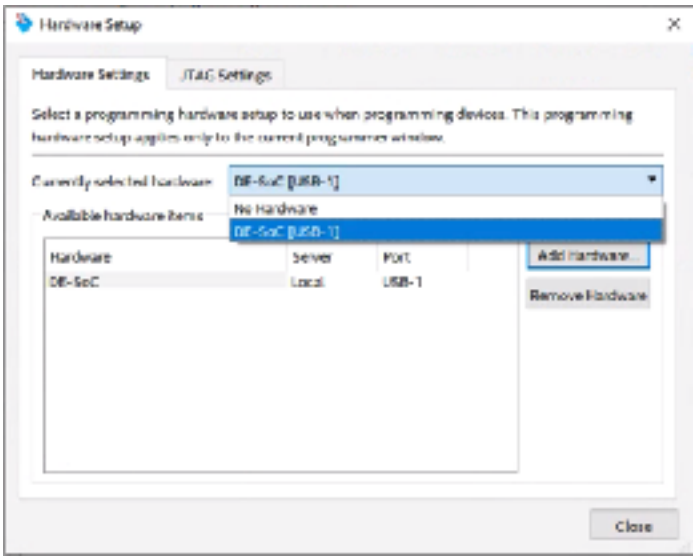
To program the board we will need to use the Quartus Programmer. The programmer can be found under tools in the top menu bar. Go to Tools > Programmer.



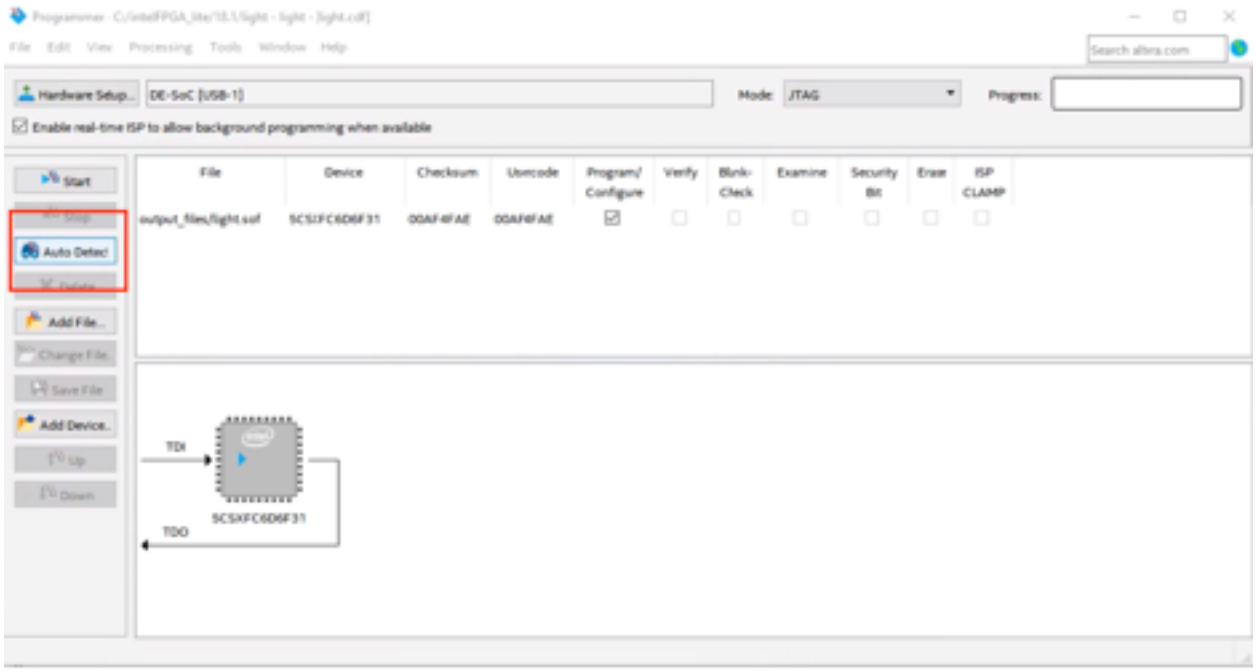
Click on the Programmer and it will open a new window.
Once in the programmer you will need to change the hardware setup to read your DE10 Standard board. Click on hardware setup

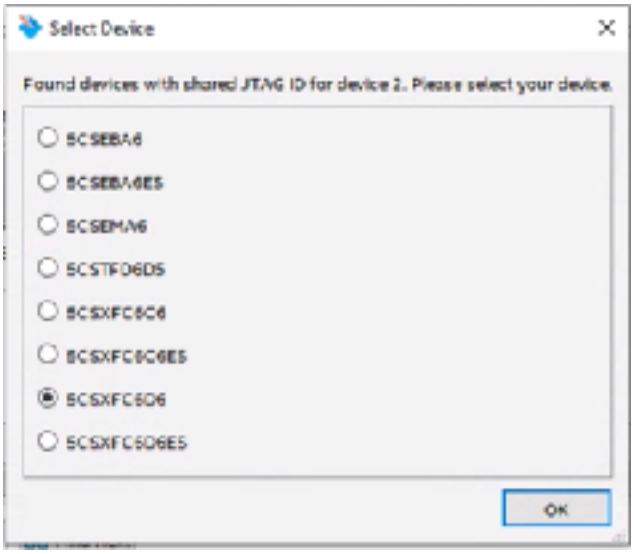


Inside the hardware setup window choose **currently selected hardware** and select DE-SoC [USB-1] from the menu. And close the menu.

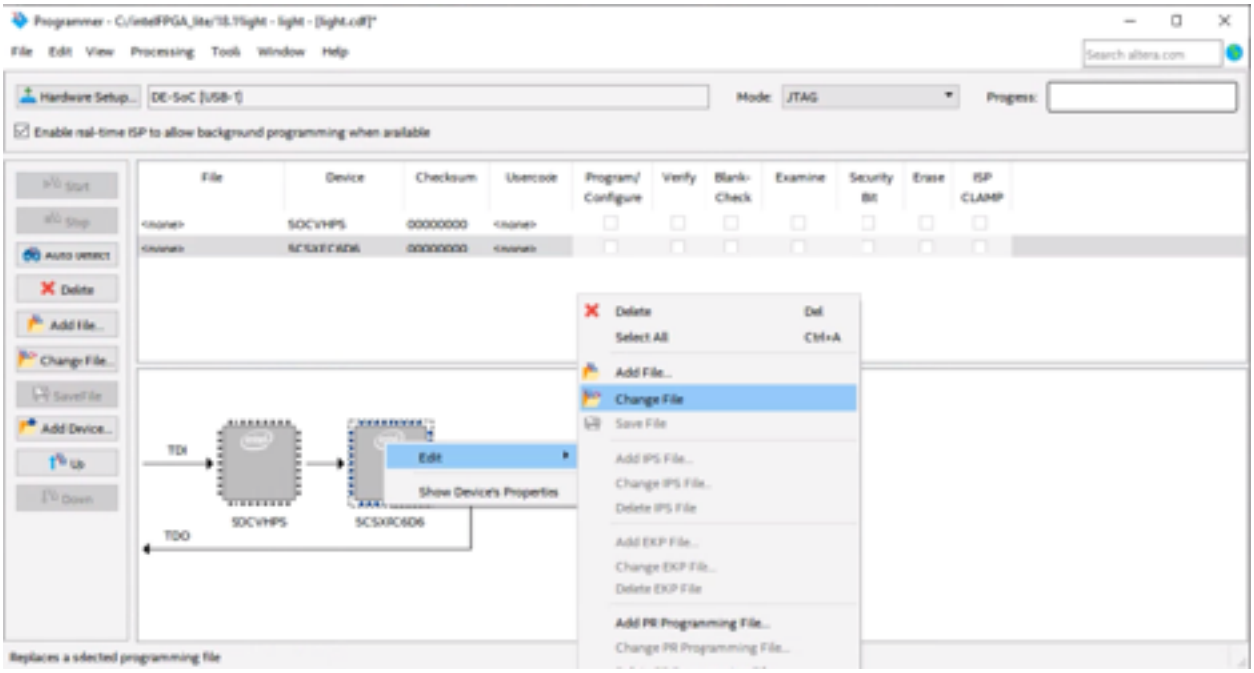


This will bring you back to the main programmer menu. Here you will now see an option to auto detect in the side menu. Click on it and it will give you a popup asking which chip you have. For the DE10 Standard we will need to choose **5CSXFC6D6.** If the programmer asks to update the device list, click yes otherwise move onto the next step.

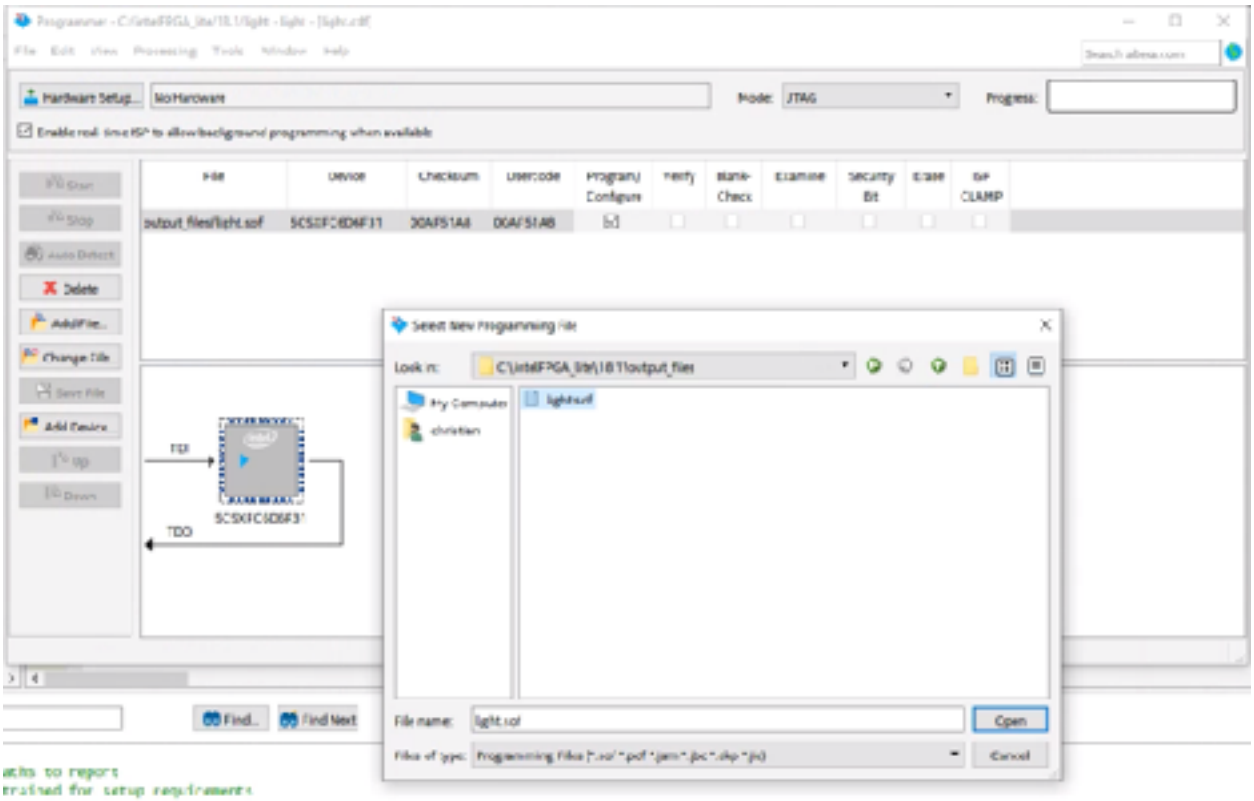




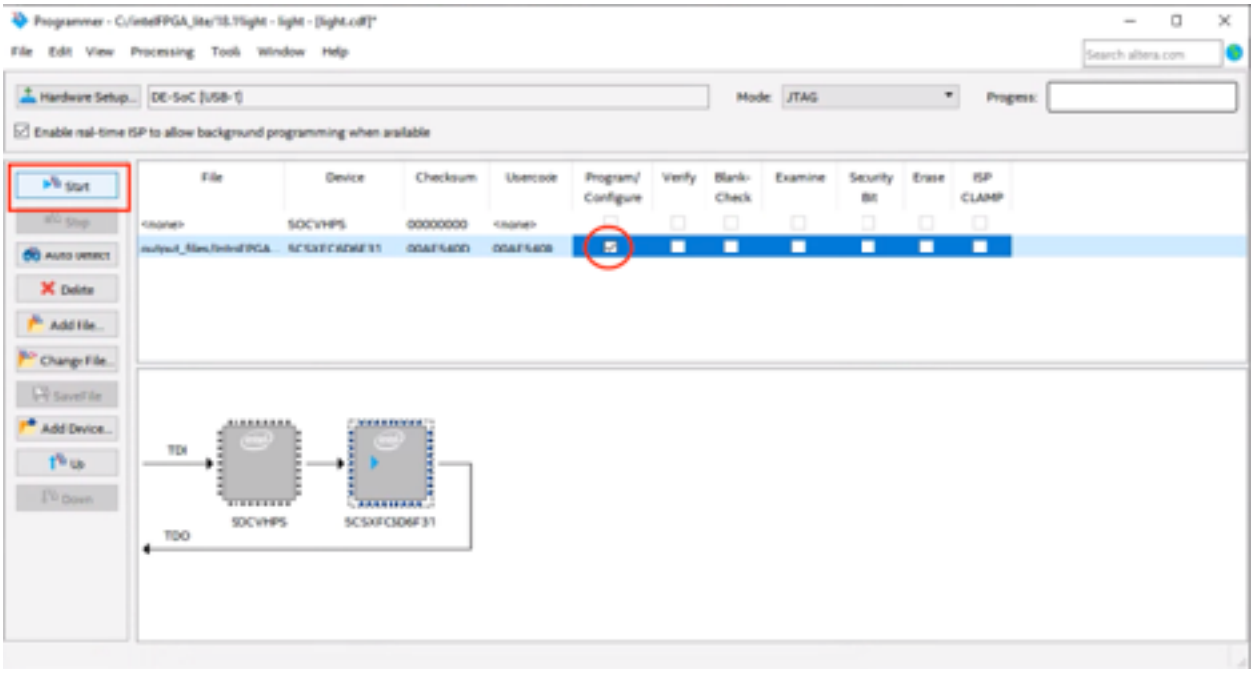
The programmer will now display 2 chips on screen the SOCVHPS is the DE10-stardard hard processor system and the 5CSXFC6D6 is our FPGA. Since we are working with the FPGA we will need to program over the FPGA's default program. In order to program the FPGA you will need to right click on the chip and select change file.



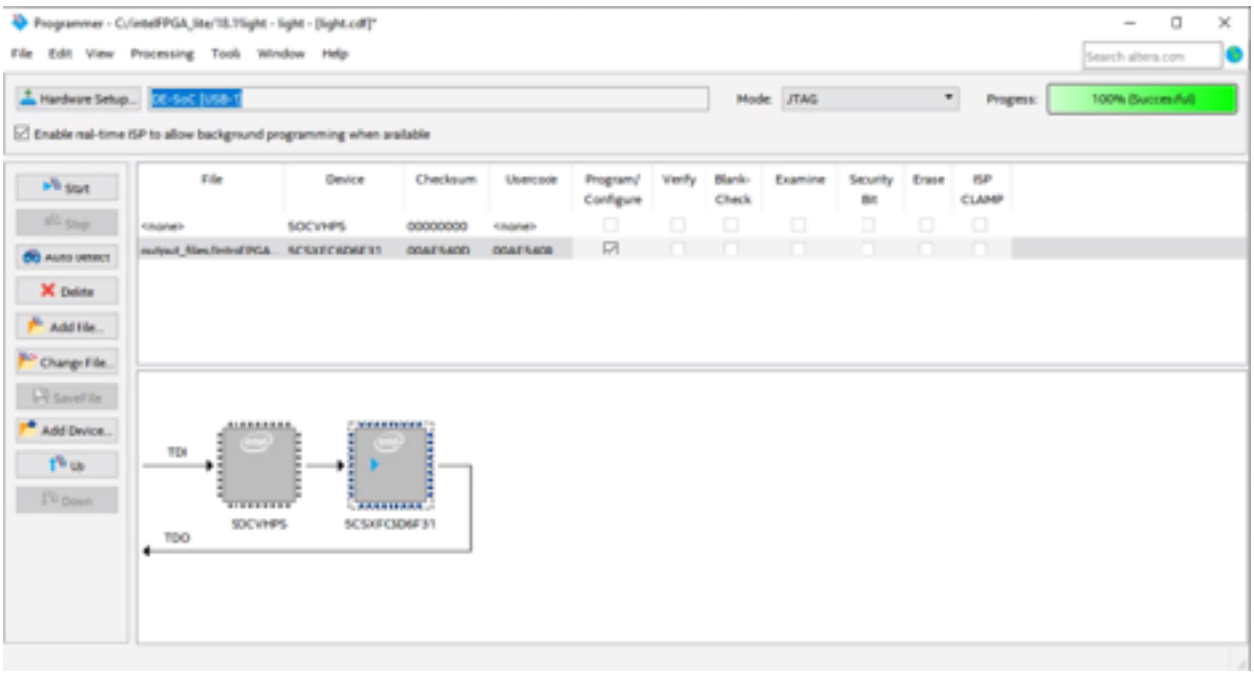
You should have a light.sof file in your project folder that was automatically generated after compiling and assigning the pins. This will be the file that the FPGA will run to emulate the created circuit. Once you are done click open.



After the file is open you will need to check the Program/Configure box and click Start.



If you programmed it correctly the progress bar will read 100% successful.



Test the Designed Circuit

Once you have successfully configured the board with your sof file you can now test the implemented circuit. Flip the switches and test all four combinations (00 to 11). If the lights match the output of the truth table shown earlier you have programmed your circuit correctly.