Tutorial

Quartus Install and Verify Setup

EECS 3216, Digital Systems Engineering
Modelling, Implementation and Validation

Winter 2021

By Jaspal Singh, Engineering Technologist,
Lassonde School of Engineering. EECS
York University, Toronto.

1 QUARTUS SW INSTALL AND TESTING INSTRUCTIONS

1.1 FOR THE WINDOWS SYSTEMS

1. Link for Download (You will have to register, before downloading)

   https://fpgasoftware.intel.com/

   Currently go for version **20.1 (Lite)** version which is free. Download file size is around 2GB, while the total install size (Quartus, ModelSim, Device files) is around 14 GBytes and it takes about an hour, for the installation!

   You may have to disable your virus protection, while downloading and installing. Due to download sizes, it might be better to download **Individual Files**, instead of the combined files.

   In that case, download the following files individually and install them *sequentially*.

   - Quartus Prime Lite Edition (including the NIOS EDS)

     For using the DE10-Lite board, you will also need to install the..
   - Max 10 FPGA device support file. (if you download this file, at the time of Quartus install, this driver would get installed automatically)
     Note, if you have **already** downloaded the device files, *Quartus at the time of it's Install* will give you the option to install the downloaded Device files also, the same time.

   - ModelSim Intel FPGA Edition
     Note, when installing ModelSim, point it to the *same directory*, where you installed Quartus, e.g. C:\IntelFpga_lite\20.1 or as per your own case.

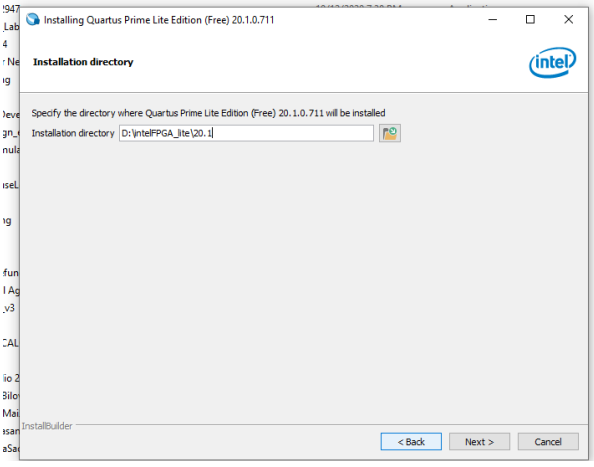
   For your reference, also download the '*Intel Quartus Prime software, User Guide*' This document will help you get acquainted with the software's IDE.

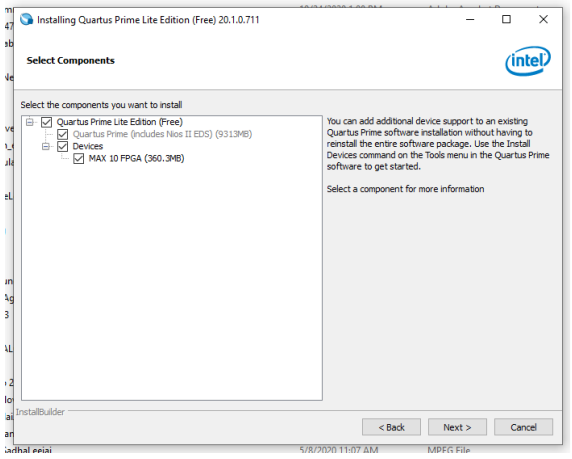
   Use one of the following links:

   https://www.intel.com/content/www/us/en/programmable/products/design-software/fpga-design/quartus-prime/user-guides.html

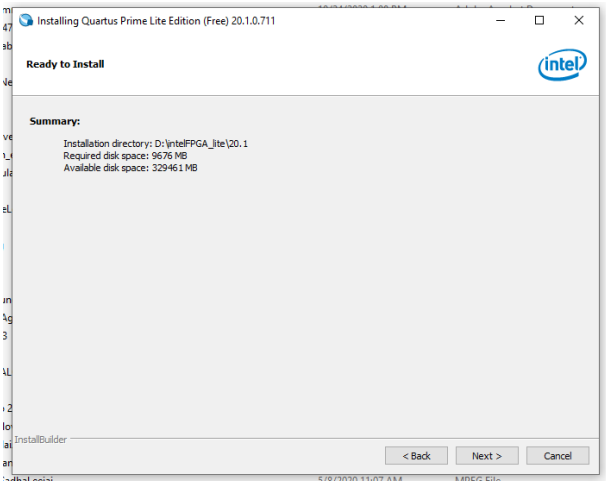
   https://tinyurl.com/y7xaw5u4

1.1.1 INSTALLING QUARTUS AND MAX 10 DEVICE FILES

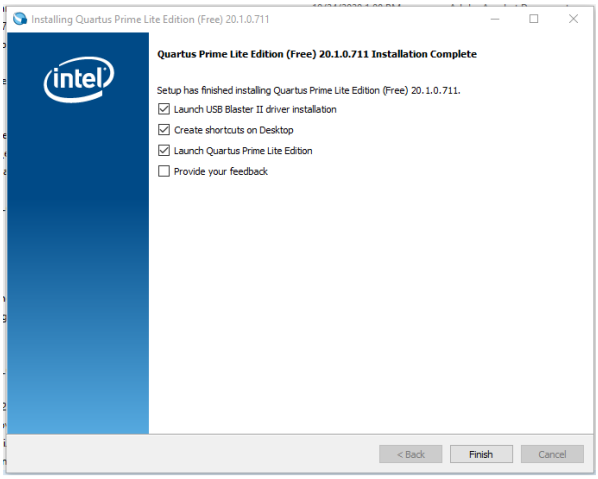


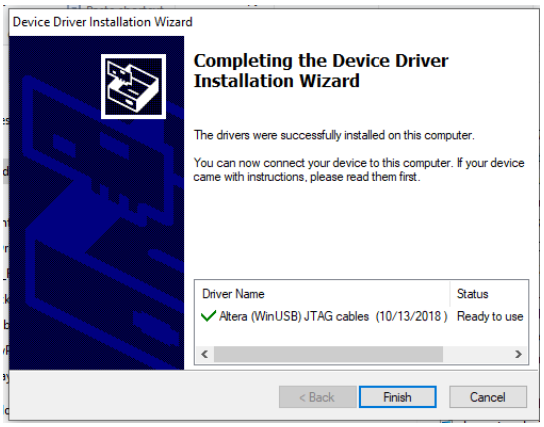


Specify the Install Directory

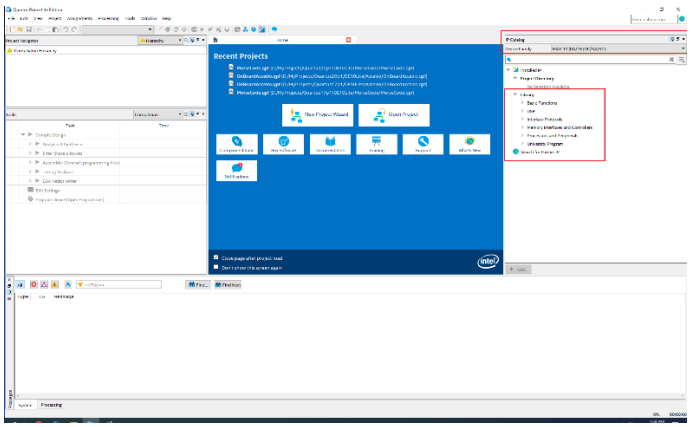


If already downloaded Max10 Device file, check it off



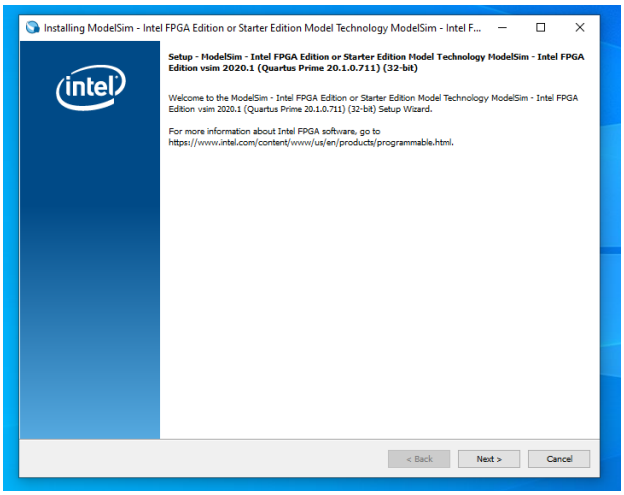


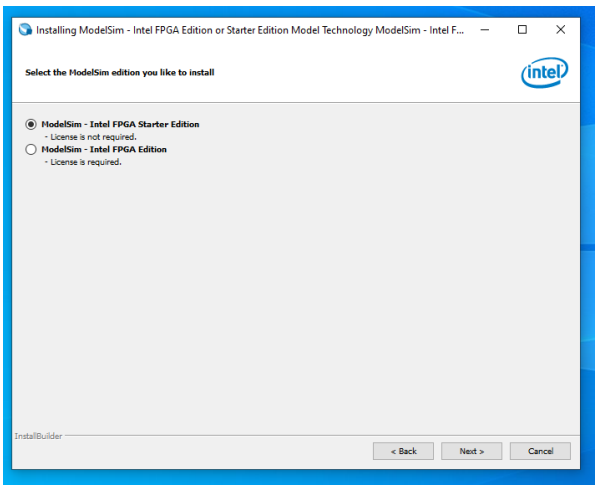
USB Blaster driver is installed correctly.



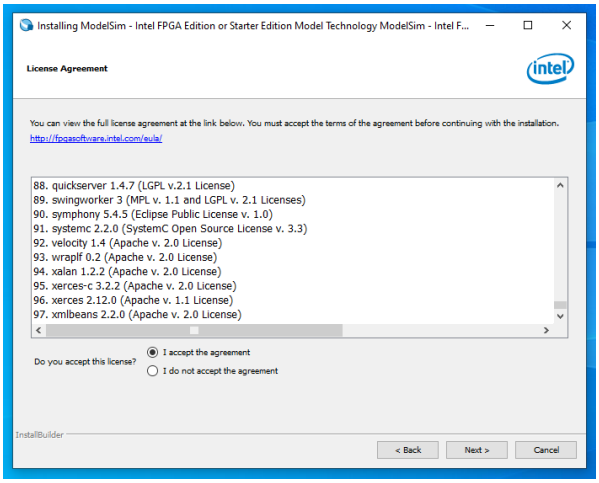
Check that you see Max 10 Family and Installed Libraries

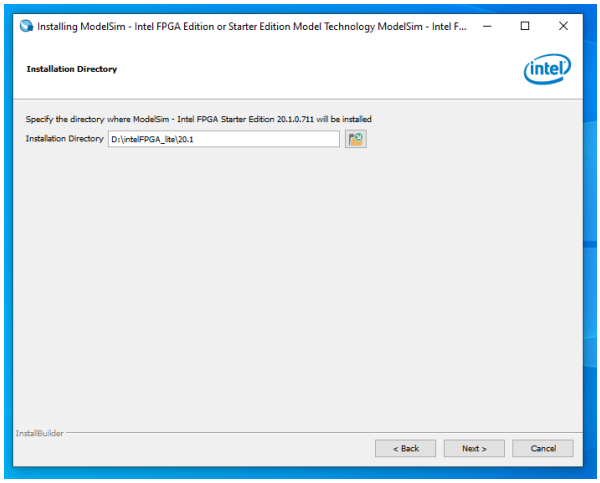
1.1.2 INSTALLING MODELSIM



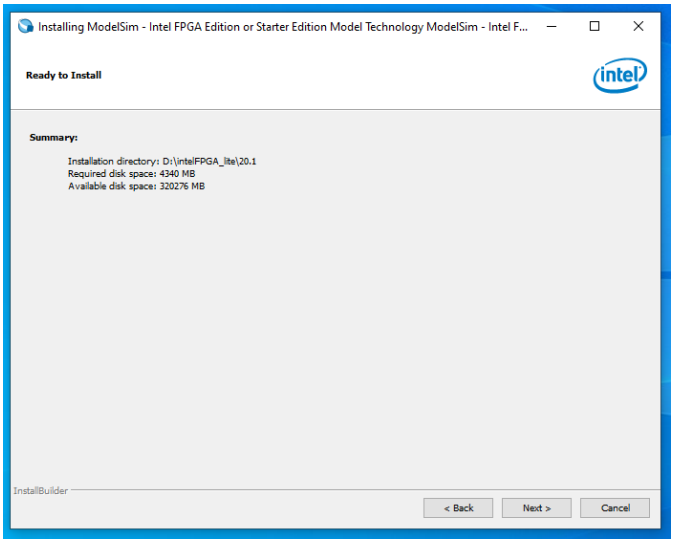


Select the free ‘Starter’ Edition





Use same Quartus 20.1 install path!



Last step, before the install begins.

1.2 USB BLASTER DRIVER, INSTALL AND CHECK

****Note**,** if the USB Blaster is not installed properly, you won't be able to Program your DE10-Lite board!

The USB Blaster driver **is included** in the download for the Quratus Prime itself. Verify that you do have a folder by the name **usb-blaster-ii** under this path, where text in blue is, *as per your own unique case*.

your_drive:\intelFPGA_Lite\20.1\quartus\drivers

If run into trouble with your USB driver and are having trouble communicating with the Board, check these instructions:

https://www.terasic.com.tw/wiki/Altera_USB_Blaster_Driver_Installation_Instructions

https://mil.ufl.edu/3701/docs/quartus/byteblaster/usb-blaster_driver_install.pdf

1.3 TEST QUARTUS INSTALL AND USE THE IDE

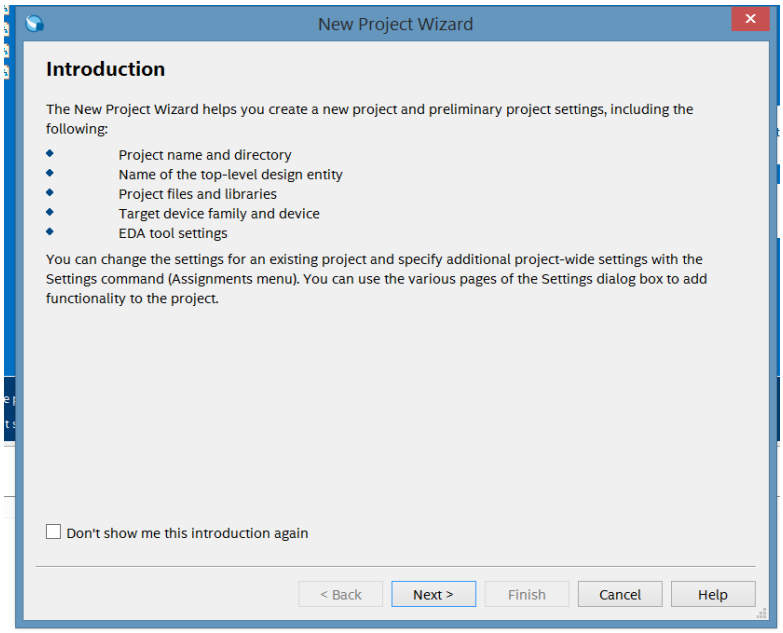
Perform the steps under this section…
1) To ensure that your SW installation went fine and
2) Get familiar in using Quartus IDE, for doing your labs. Having the Quartus Prime SW, user guide handy, would be useful.

This section **assumes** that you have successfully installed Quartus SW, Max 10 device support files and the USB Blaster driver.

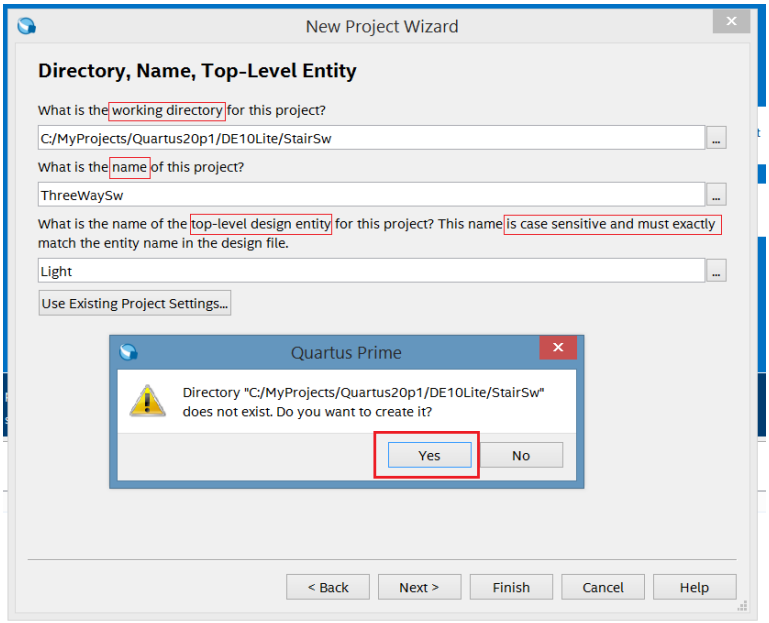
1.3.1 NEW PROJECT WIZARD

Start Quartus SW, from the desktop shortcut or from Windows Start menu.

1. Open a New Project Wizard, by selecting
   File -> New Project Wizard You would see the following screen..



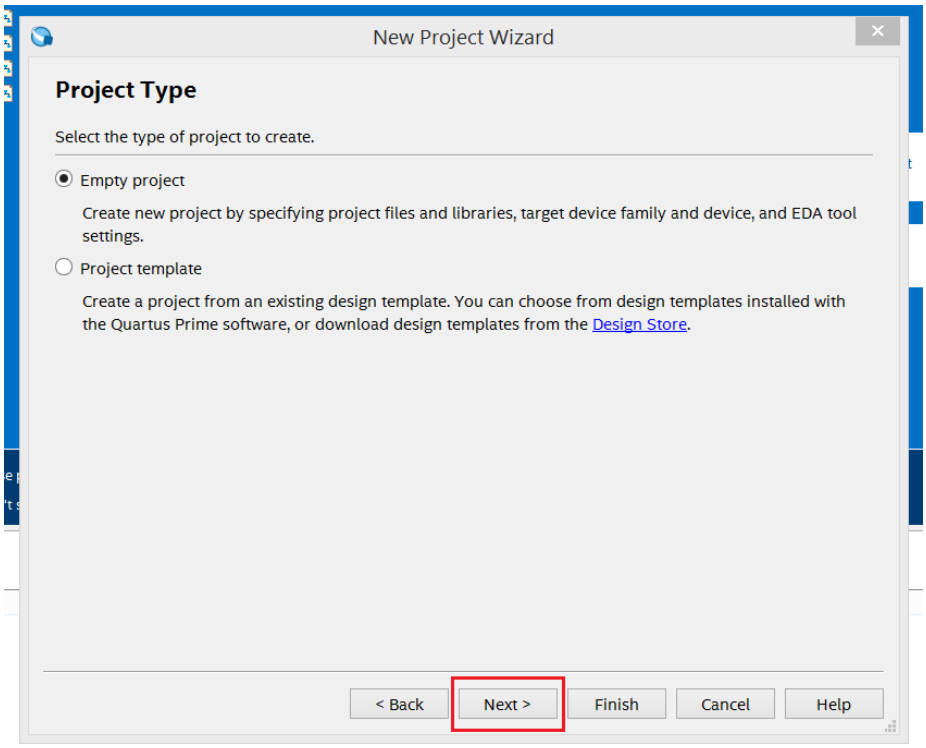
Click on 'Next' and you will see the screen shown below….



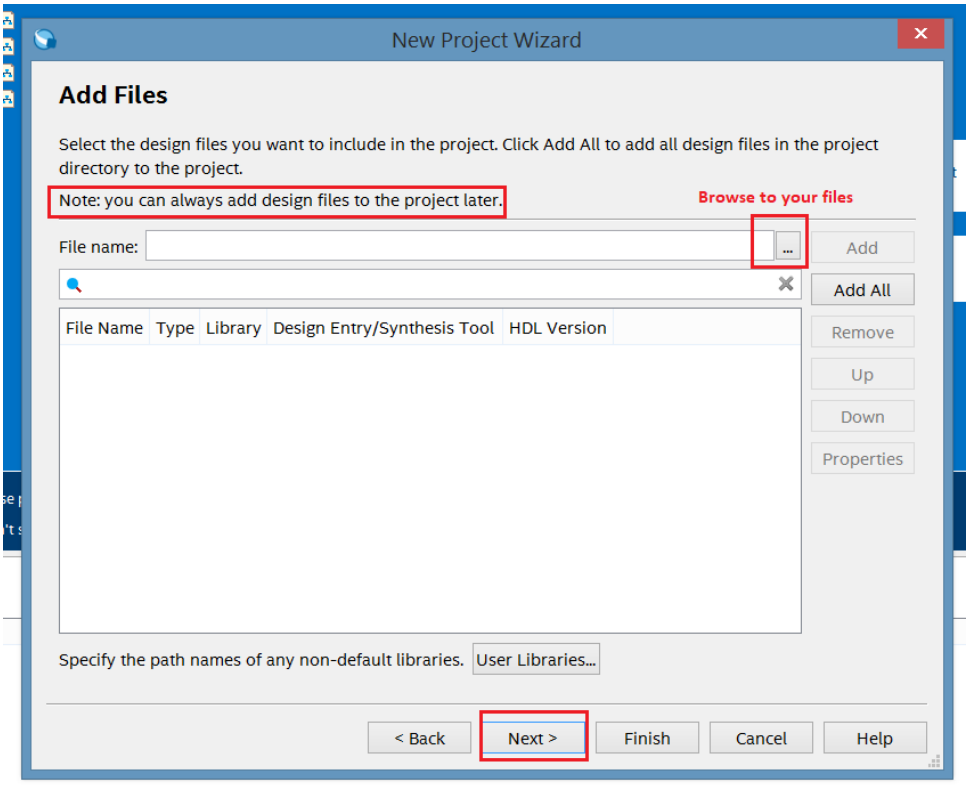
On this window, fill out…

a) The path and the name of the directory, where you want the project to be created and stored.
b) The name of your project
c) The name of the Top Level module, in your Project. It is **important** that the name is entered **exactly,** including the case, as it in your code, so that it **matches.**

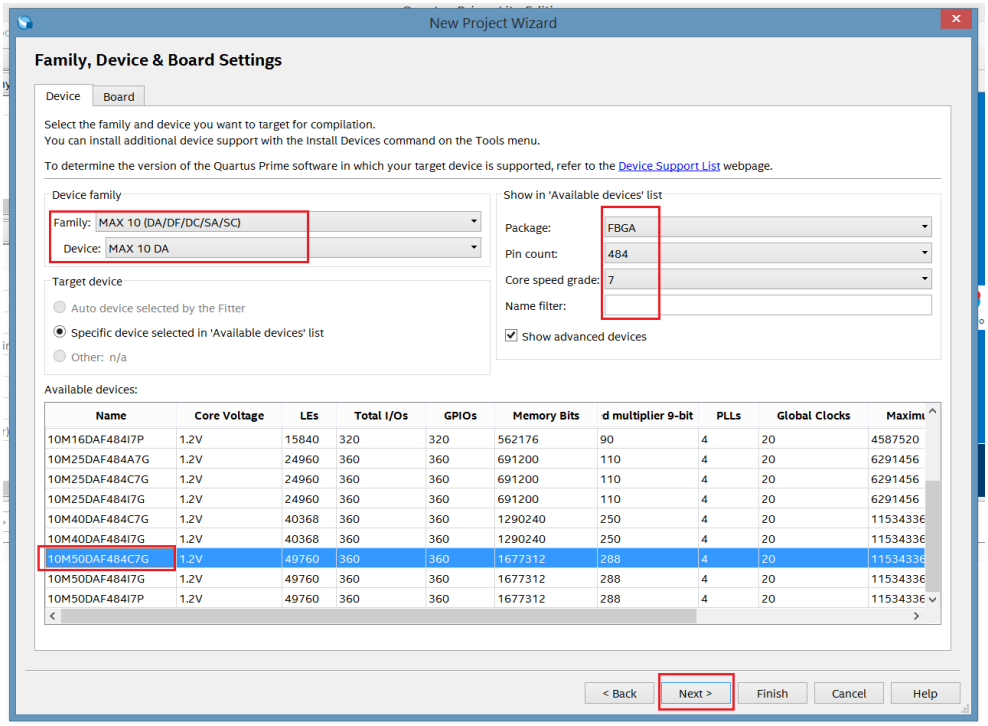
If the working directory is not existing, you will get a message, asking for your permission to create the directory. Click Yes and then go to the next screen, by clicking Next.



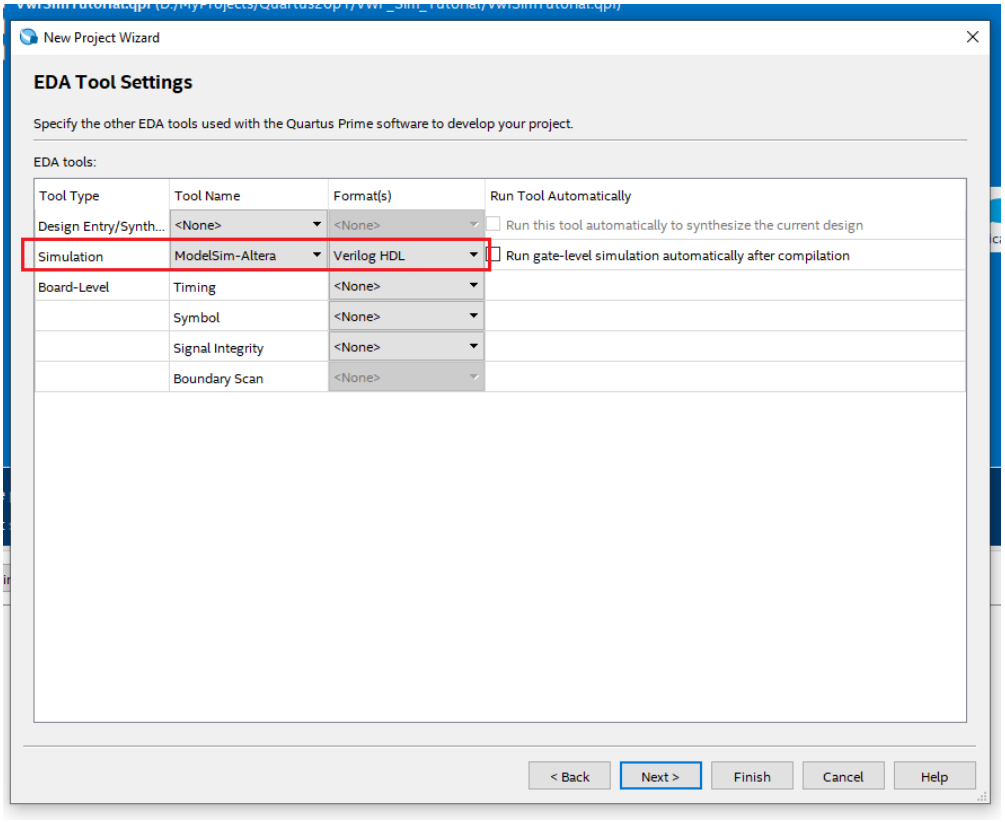
Use the default *Empty Project* setting and click Next. This takes you to the tab, for adding files, to your project.



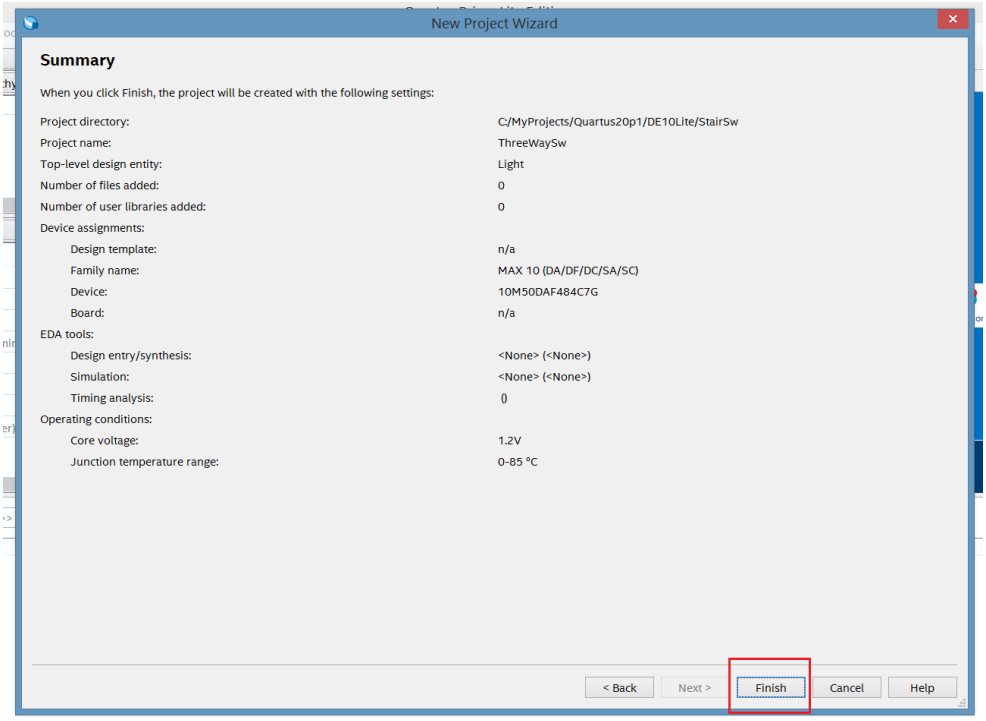
For the time being, we won't add files here, in this step. Note, you can always add files to the project, later also. Click Next, to proceed to wizards' next step.



This is an important tab. For DE10-Lite board, fill out as shown above and click Next.



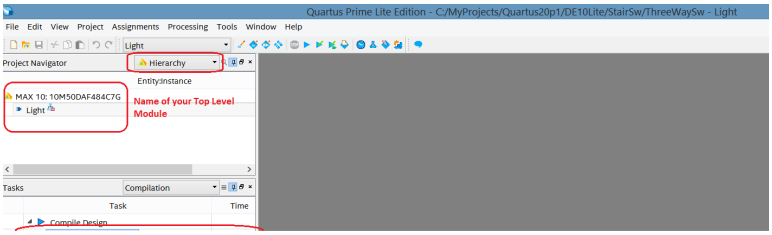
In the EDA Tool Settings window, select ‘ModelSim-Altera’ as the Tool and Verilog-HDL as the Format.



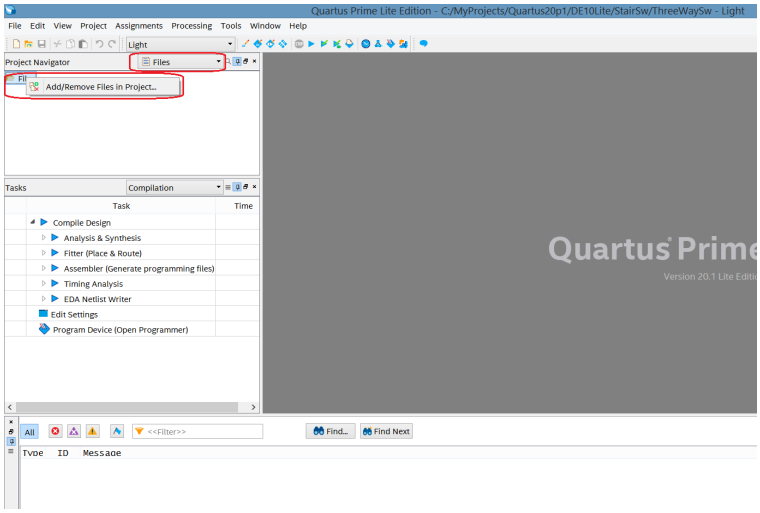
Do the same for this last slide and click Finish.

1.3.2 ADDING FILES

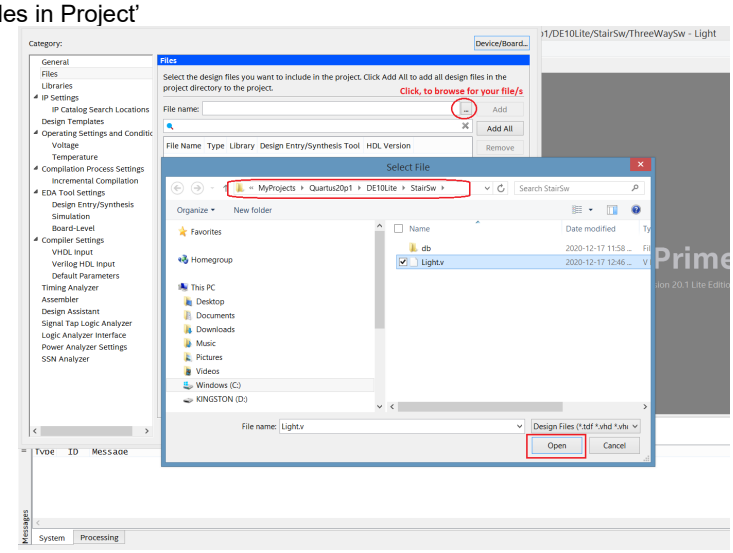
After you have done the above steps, Quartus would show you a view like below, showing only your Top level Module, as you declared it in the previous steps, since no files and lower hierarchy modules are currently present.



To add the files, you would do the following steps…



Select Files from the drop down menu. *Right click* on the 'Files' under the Project Navigator and select 'Add/Remove Files in Project'



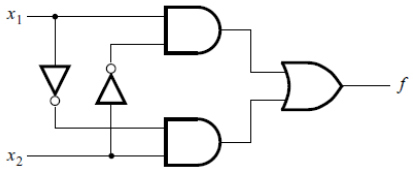
Browse to the directory which contains your HDL (Verilog, VHDL, SystemVerilog) file. In the case shown a Verilog file called Light.v is visible. This file was previously created and saved under the project directory.

Here is the sample code in this file (Credit: Altera document ‘Quartus II Introduction Using Verilog Design’). Using a texteditor of your choice, you can copy and save this file, under the directory, from where you want to import it, preferably your project directory, into your project. (Pay attention to the name of the module!)

```
module Light (x1, x2, f);
    input   x1, x2;
    output  f;

    assign f = (x1 & ~x2) | (~x1 & x2);
endmodule
```

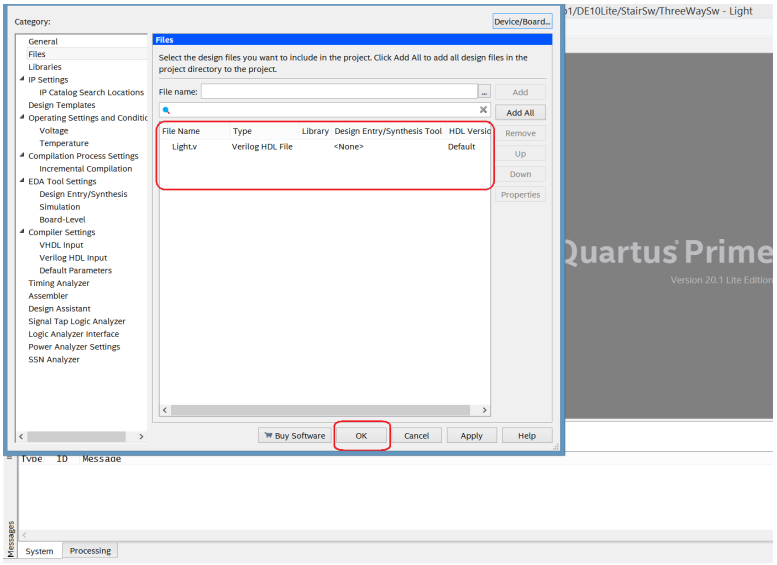
This is design of a simple two-way light controller, which we typically find in our homes, on stairs. Below is the diagram and Truth table of this circuit.



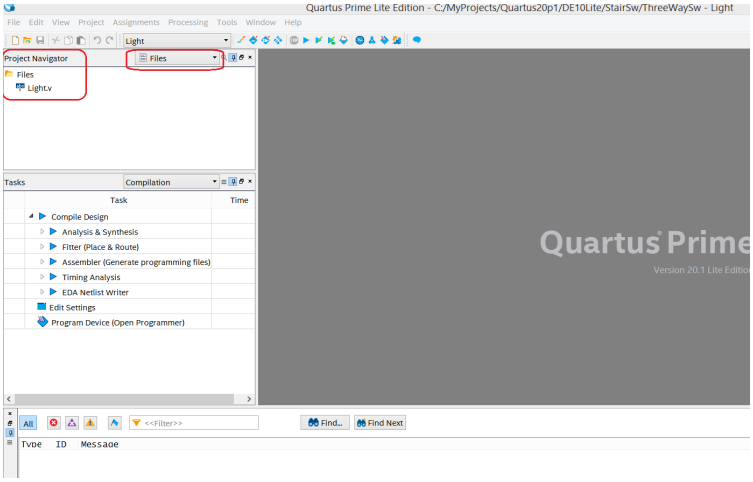
x_1	x_2	f
0	0	0
0	1	1
1	0	1
1	1	0

Figure 12. The light controller circuit.

When you click ‘open’ on the last picture shown above, and then click ‘Add’, you see the following window… **Note**, you can open multiple files, in one go. In that case, click on ‘Add All’, so that they get added to the project.



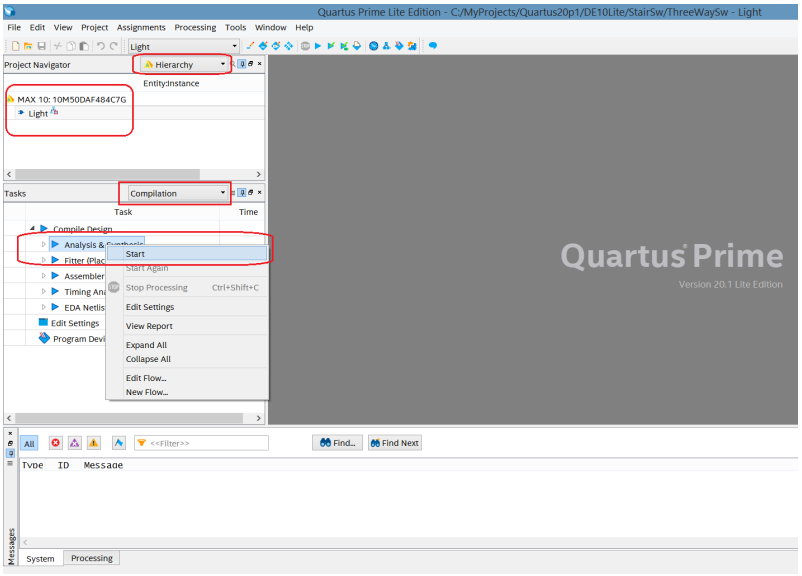
Click ok and the file/s are added to the project. When we do it, the following screen is seen and you see that our file Lights.v has been added to the project.



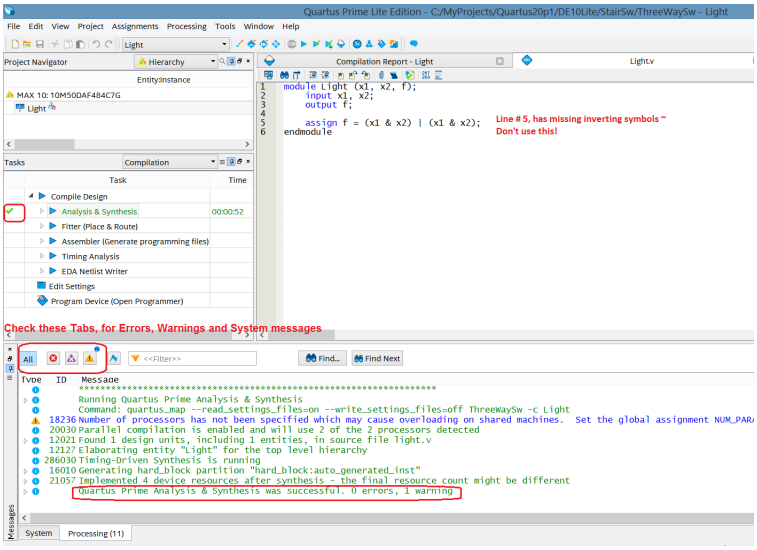
1.3.3 COMPILING THE DESIGN

After we have added the file/s the next step is to compile the design, fix any errors, look at the warnings and preparing the design, for checking / testing on the board.

Right click on the 'Analysis & Synthesis' step, under the Compile Design options, and select Start.



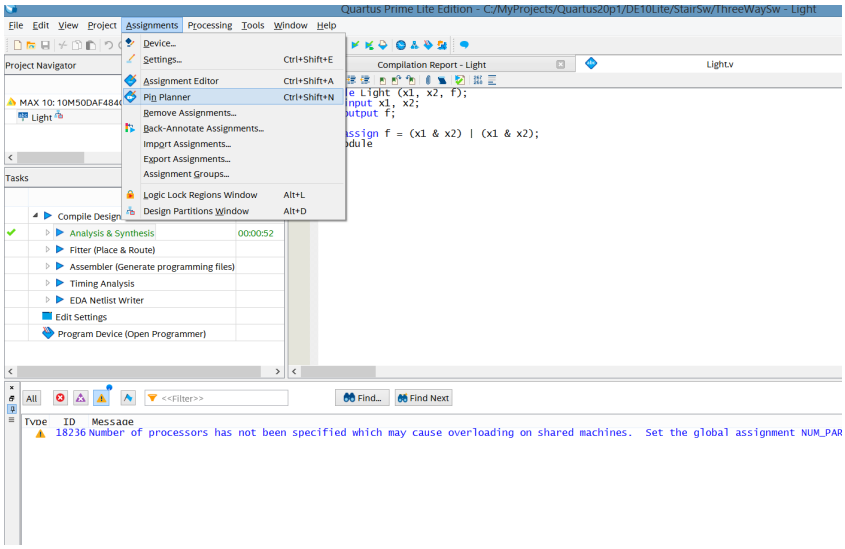
After the compilation step is complete, if there are any errors, fix them. Also closely look at the error and warning messages, as they often provide good clues to where the problem / potential problems lie in your code.



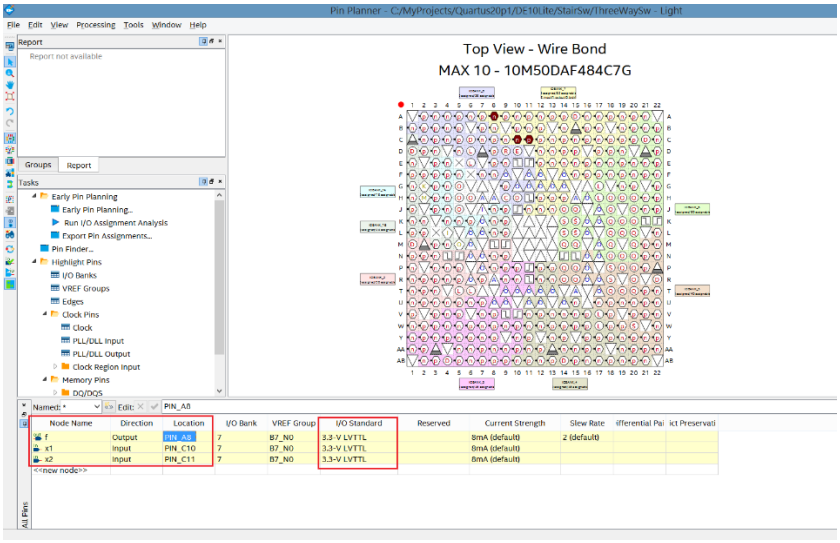
After the compile result is satisfactory, the next step ususally is to *lock the Pins* (aka assigining the input, output ports of the design, to specific pins of the FPGA chip), so that the design, when ready can be tested on the board. This simple design only has three ports, so please assign manually, as shown in the picture below.

Note, when the design has many ports, it is often fasters and less error-prone, to '*import*' the pin assignments. How to do this shall be explained later, in more detail. For now, manually assign the ports x, y and f, to the pins, as shown below.

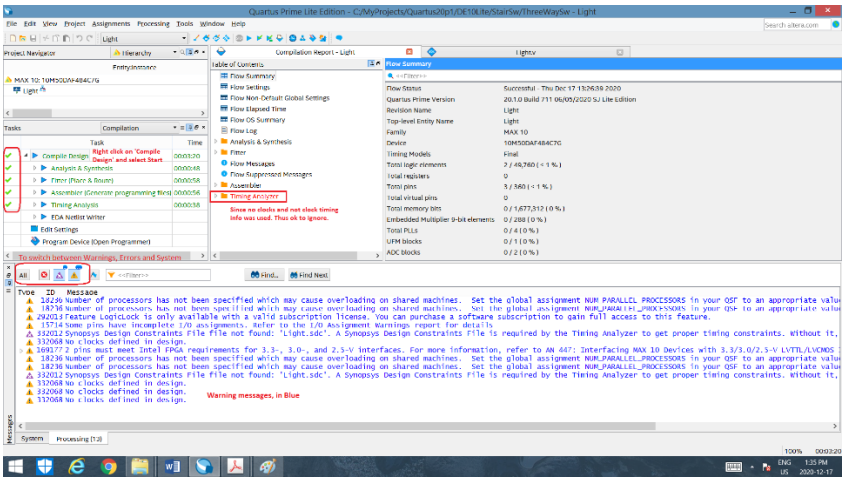
First, open the Pin Planner tool, as shown.



Then in the Pin Planner tool, manually assign the pin locations, to the three ports.



After this step, close the Pin Planner tool and then we will do a complete Compile Step on our design. Once the compile step is successfully done, you would see green check marks, next to the four main steps of the design compile process, as shown below.

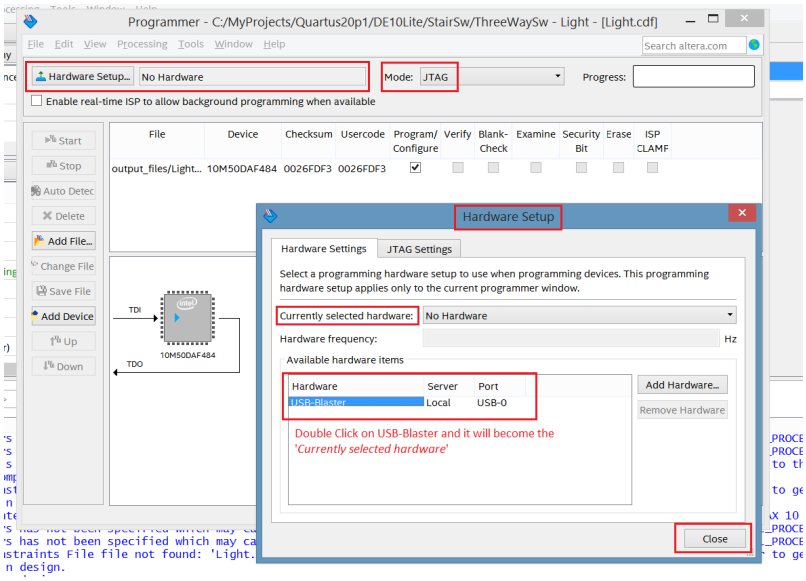


You may notice, that the Timing Analyzer is in red. This is because we do not have any clocks in our design and as such no clock timing information was used. This can be safely ignored.

1.3.4 PROGRAMMING THE BOARD

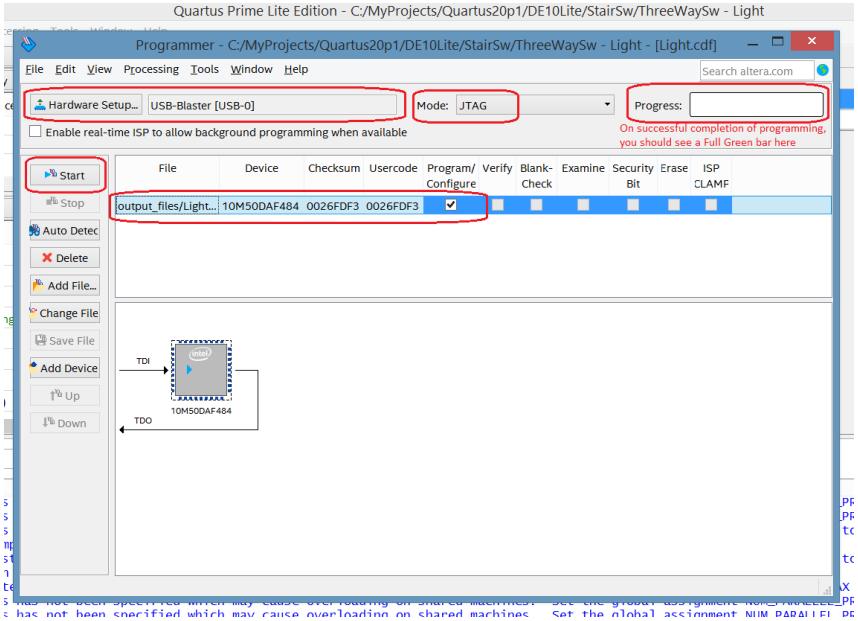
After the compile step is over, it is time to program the board and test the design on the board. For this we need to use the Programmer tool.
Note, Before attempting this step, the DE10-Lite board should already be connected to your laptop, via the USB cable. When the board is powered up, the Hex displays shows numbers between 0 and F, while the LEDS blink on and off, in a rotoary form.

Do menu Tools -> Programmer (or click on the Progammer icon) and you would see the following window.



Make sure, that Mode is JTAG!
Initially under Hardware Setup, you may see ‘No Hardware’. If this is the case, double click on Hardware Setup, to open it’s window. If your USB Blaster driver was installed properly, you will see USB-Blaster available. Double click on it, so it becomes the ‘currently selected hardware’ and then Close the window.

You should now be back into the Programmer Tool, with USB-Blaster now visible, as shown below.

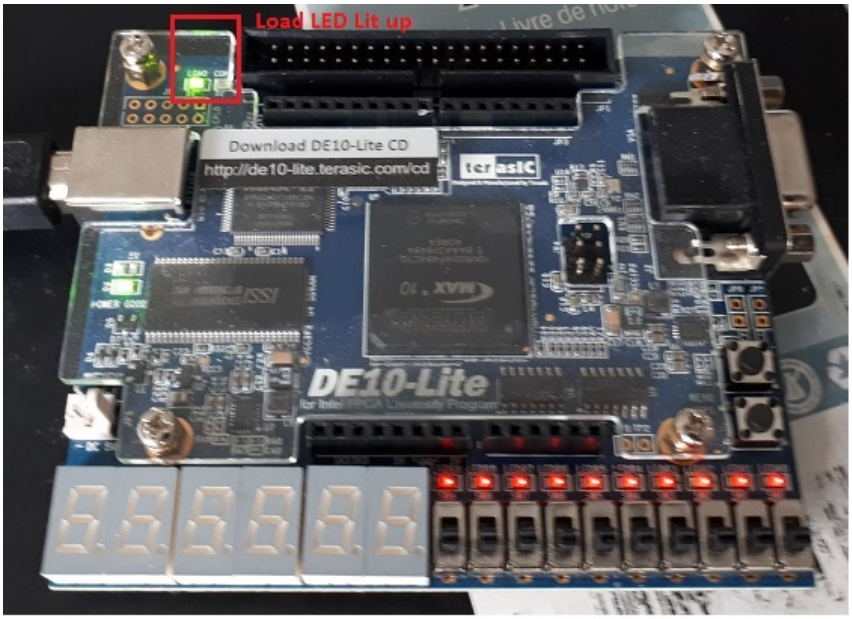


Make sure, Program / Configure is checked off, next to your SOF file.

Note, if you don’t see the SOF file automatically selected, click on ‘Add File’ and browse your project’s Output directory and select your SOF file, so that it gets added to the Programmer.

Click Start and programming of your board shall starts. If everything went well, the Progress bar should get completely full with a green bar.

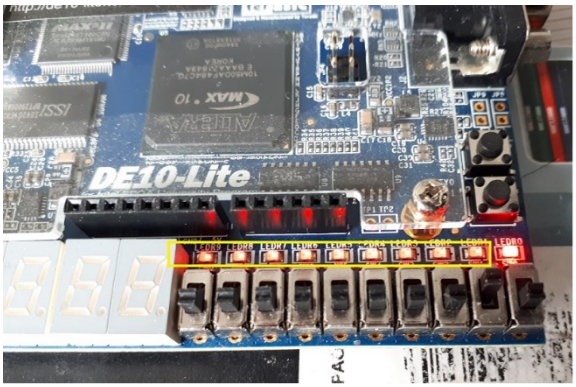
During programming, you will see that the ‘Load’ LED on the board lights up, as shown below.



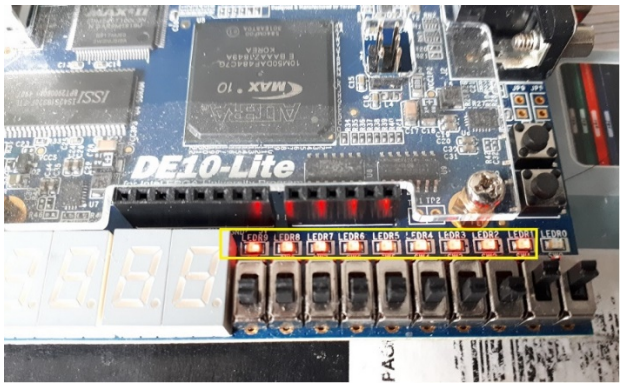
Board being Programmed.

1.3.5 TESTING DESIGN ON BOARD

This design is a simple two-switch controller, commonly seen around stairways. The X and Y inputs are the switches SW1 and SW0, while the F output is going LEDR0.
When the design is working properly, you will notice results, as shown below.



X is on, Y is off. The Light is on.



Both X and Y are on, the Light is off.

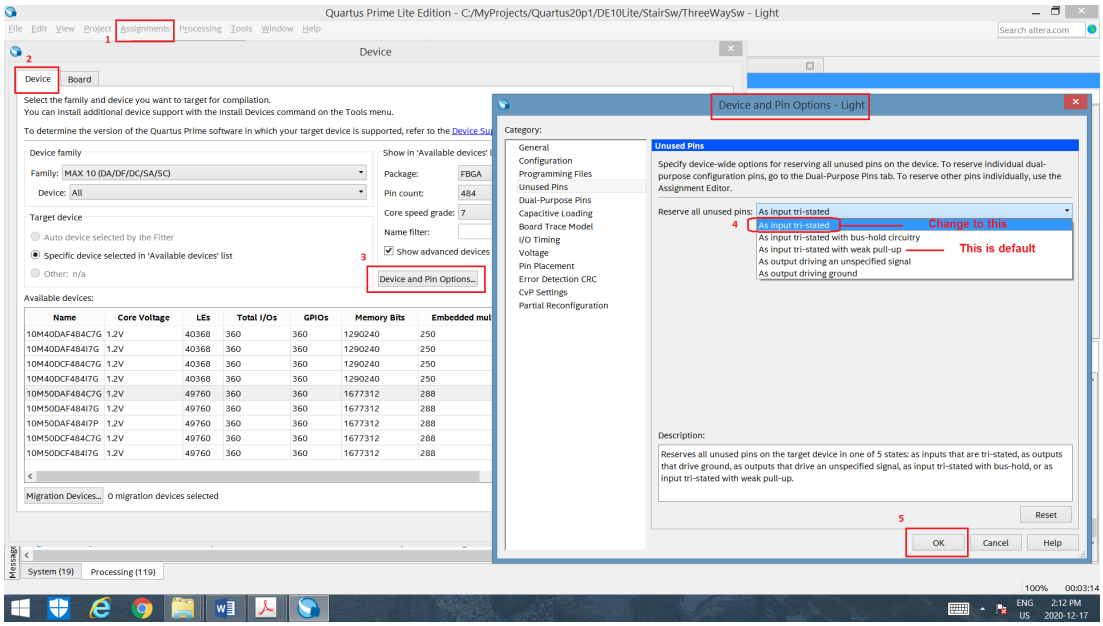
By completing these steps, you have verified your setup and are now ready for your Labs!

1.3.6 A SMALL TWEAK

You would have noticed that in the pictures above, the 9 LEDs to the left of LEDR0 are also dimly lit up.

If want to correct this, go to menu Assignments -> Device

Then follow the steps, as outlined in the picture below. Apply these changes. Re-compile and re-program your design and now those un-used LEDs won't light up.



1.4 OTHER USEFUL LINKS

This is Intel's **User forum**, which includes useful info on the FPGA related matters too. You can post your issues and search for previously posted / resolved similar issues.

1. https://community.intel.com/ You will be **required to Register**, to use if fully.

2 INSTRUCTIONS FOR THE MACOS SYSTEMS

There are few options for running Quartus on a Mac. The steps in the link below have been **tested and tried** during a previous school term and are recommended, with few extra points, as mentioned below.

Prior to following the steps mentioned in the link below, you would require Ubantu Desktop running in Virtualbox.

1. Install VirtualBox from this link
   https://www.virtualbox.org/

   After you complete the install of the Virtual Box, you should **also install** the "VirtualBox Extension pack".

   For that, follow instructions on this link:
   https://www.nakivo.com/blog/how-to-install-virtualbox-extension-pack/

2. Then install Ubantu from this link
   https://ubuntu.com/download/desktop

3. Thereafter follow the steps mentioned under this link, however make sure to keep extra 'virtual disk space' as mentioned below.

   https://siytek.com/quartus-mac-virtualbox-ubuntu/

However make following changes in these sections

- Sec 3.1 Recommended virtual disk space'

  Set the **virtual box disk space to 50GB** to allow enough space for Quartus, *instead of 30GB* in the online link. Additionally, create this space on your MAC itself and **not** on an external hard drive.

- Sec 4. Install Quartus

  Download version 20.1 Lite, for Linux (Registration is required)

  Make sure you also download and install ModelSim Altera and Max10 device files.

- Thereafter follow sections 5 and 6, as per the link.

After these steps, to verify your setup, follow the steps in Section 1.3 of **this** tutorial, "*Test Quartus Install and Use the IDE"*

If you run into any issues with install or in verifying your setup, report it to this email:

tech@eecs.yorku.ca