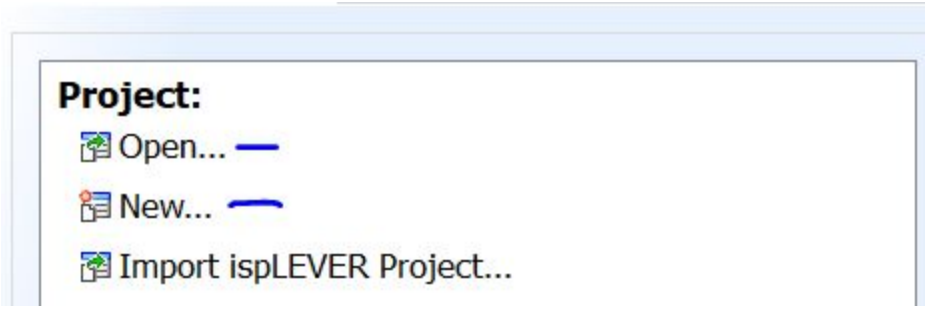
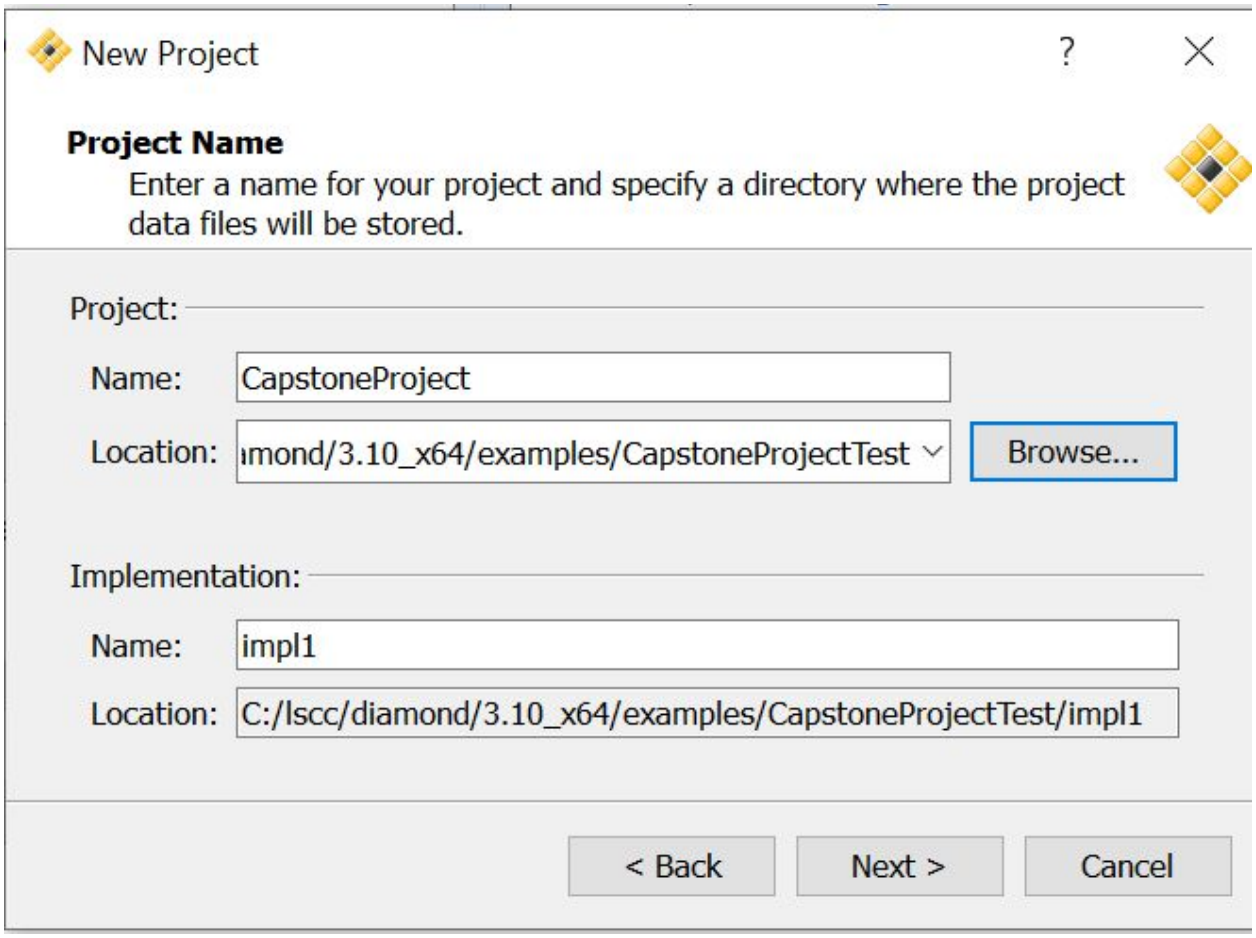
This is a guide to using the verilog code on Github with the Lattice IDE. Make sure to also have the other two .v files which contain the division module and DC PWM module.

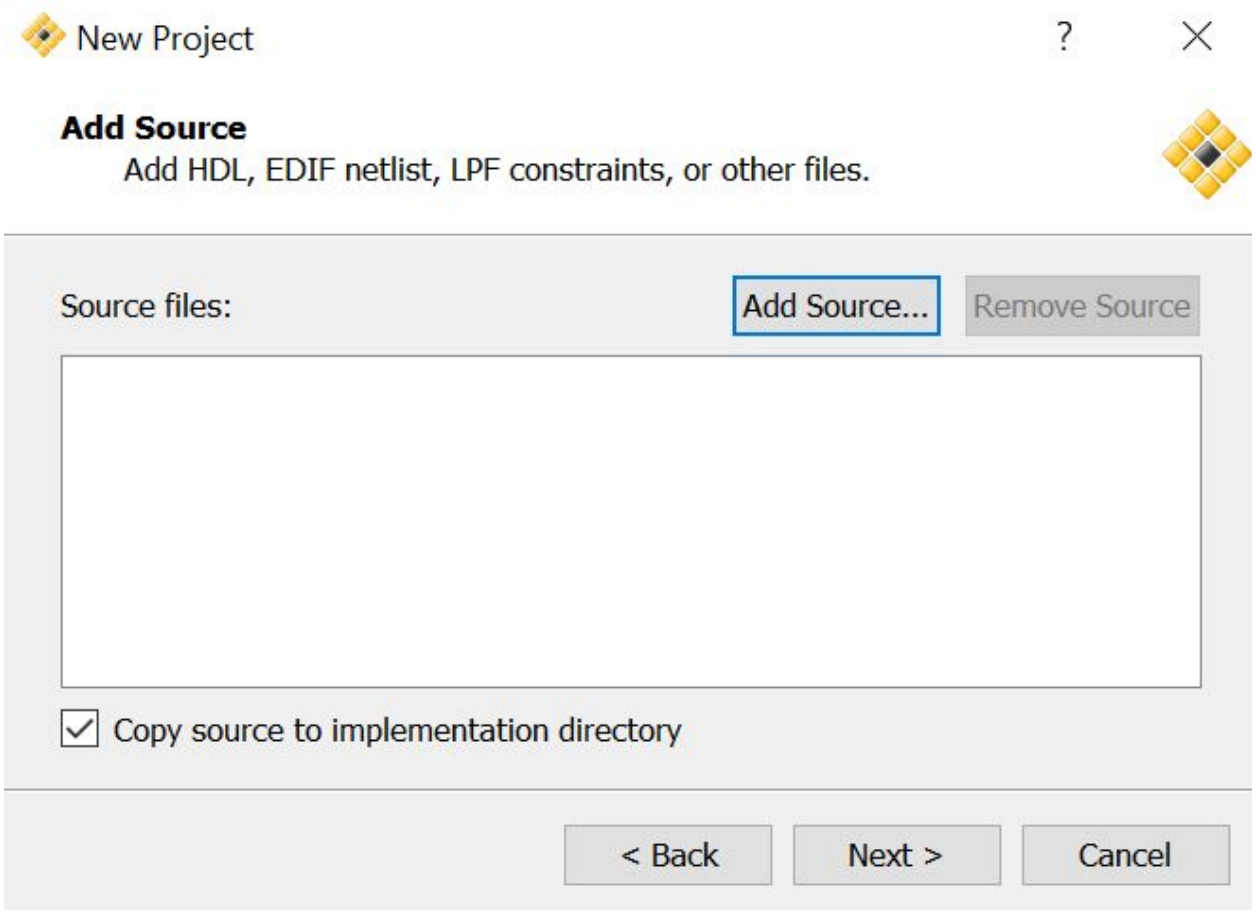
Open or create a new project



If you open new, choose a new location for your project. Make sure it's different from previous implementations.

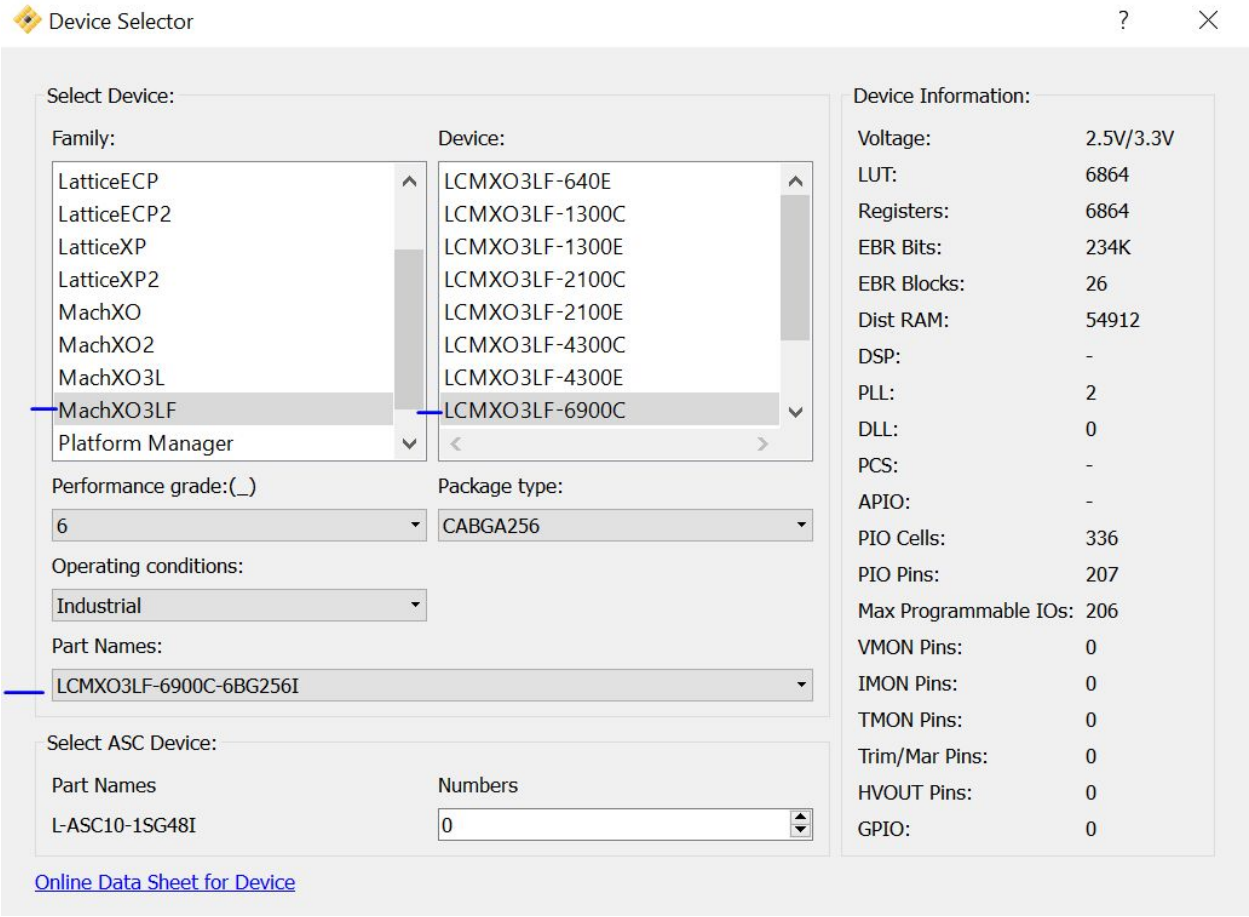


Then add your source files, these are the verilog files that you can add on if you downloaded them from GitHub, or from other example projects like the starter blink project.



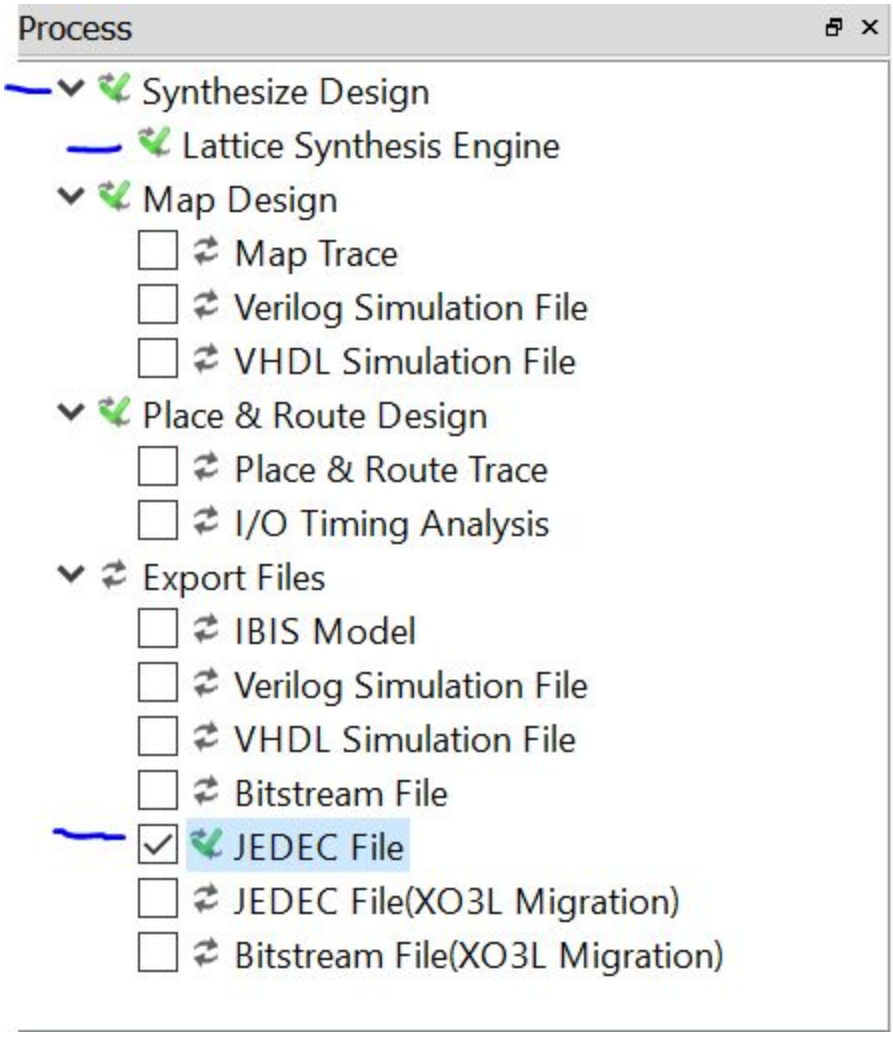
The next page will ask you to input device information, the picture below shows the settings we needed for our project. The blue lines point to where I needed to change from original settings,

so make sure those are right at least.



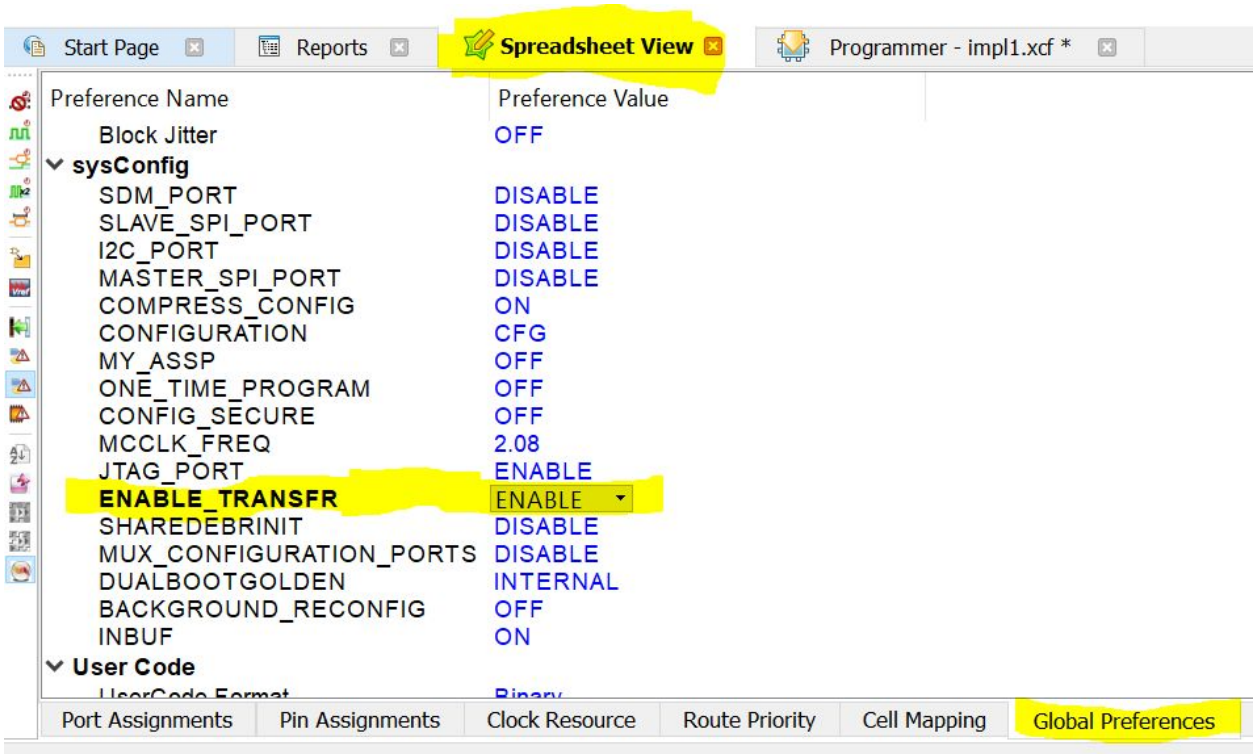
That should be the last window. Next the files to use should be opened. So far I use the ProjectCodeVX.v , PWM_Signal_DC.v , ClockDivisionforPWM.v

After making sure you have these under your files. Click process.

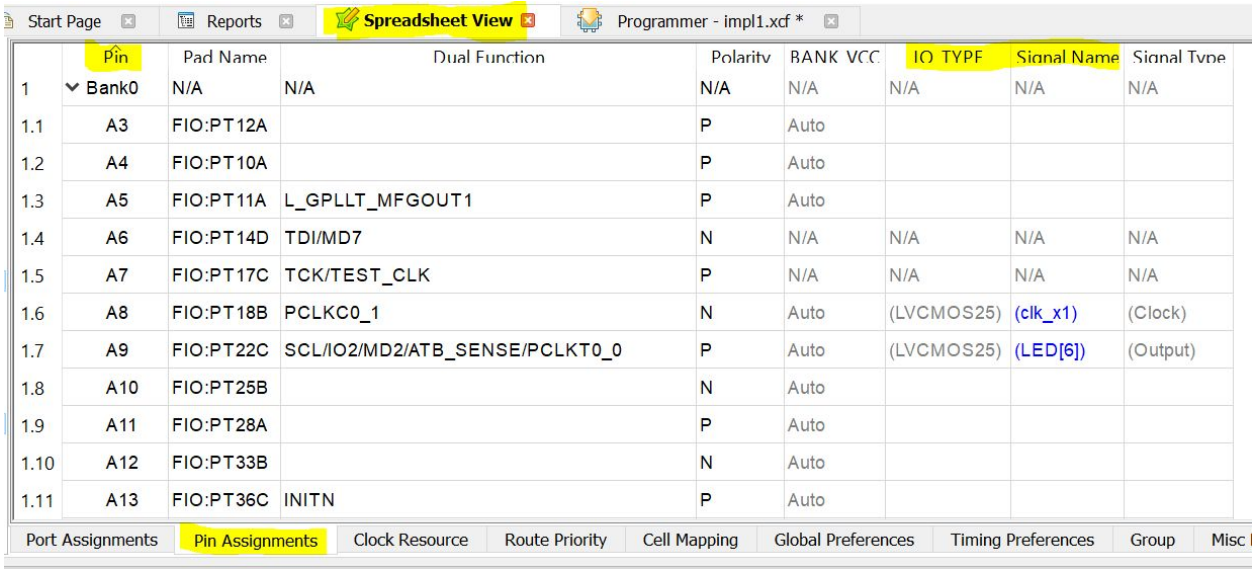


Now you can assign things in the Spreadsheet tab. (Spreadsheet is icon circled below)

After this, it should open a **Spreadsheet View** tab.The highlighted text below shows that you need to change enable transfer to TRUE.

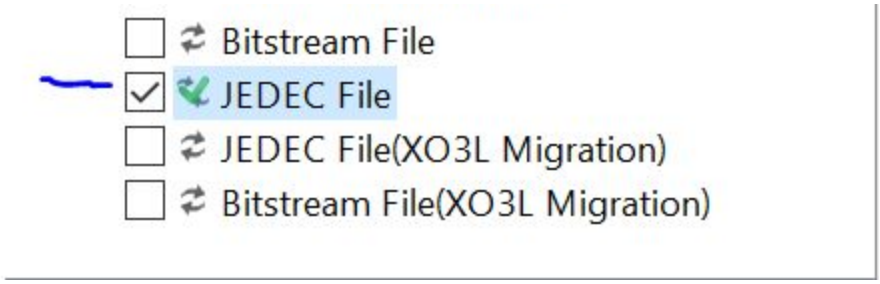


Now you can change the pin assignments, its one of tabs along the bottom. Shown below.

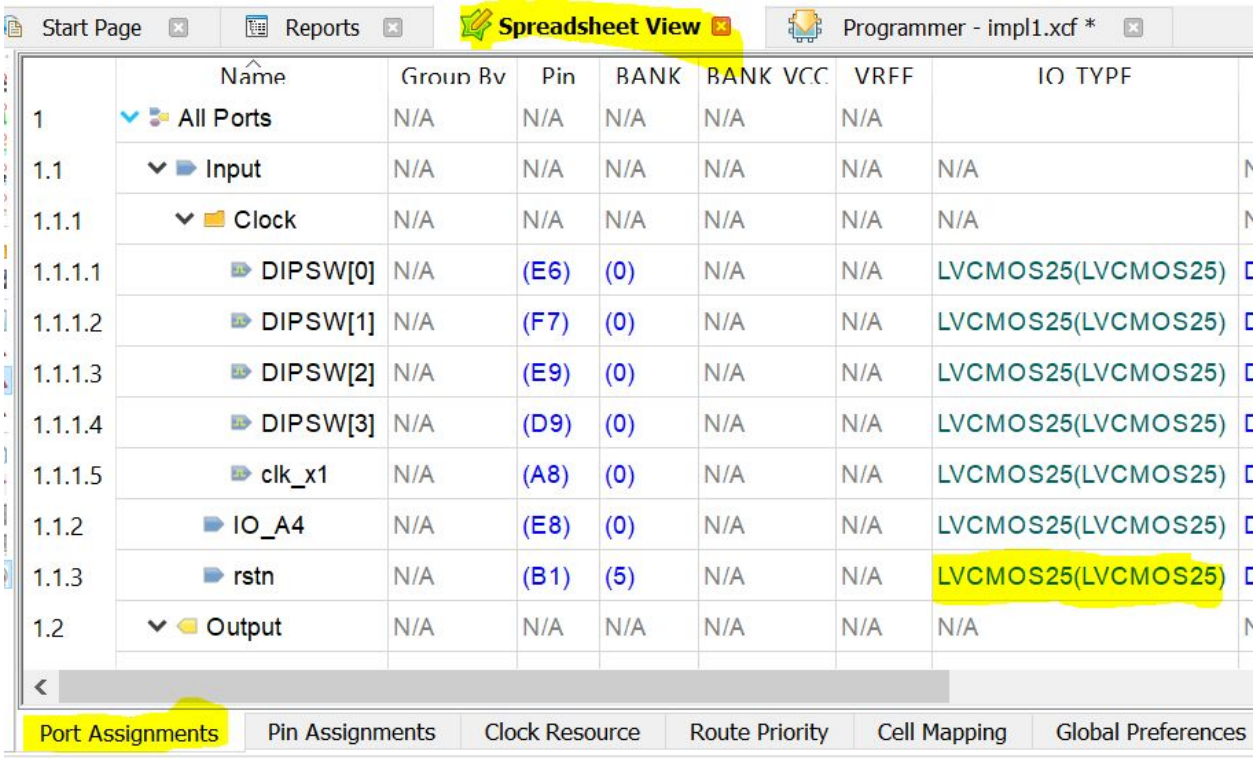


Click on the signal name to correct pad name.
Now we need to click process again to make sure the spreadsheet view changes are added to the jedec file.

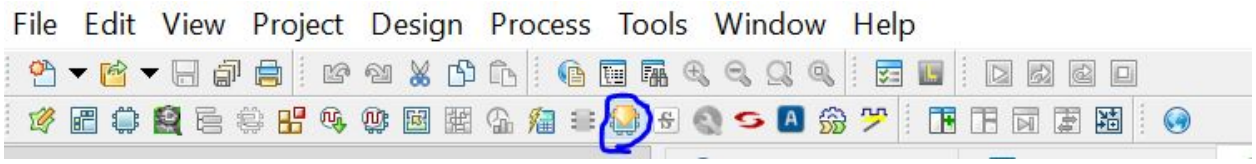
Click on the process icon again. Before you do, you'll notice the check mark isn't there anymore. This indicates changes has been made.



Lastly, make sure to check that the voltages are set to default in Port Assignments. I've never really had issues with it, but our Lattice Sponsor warned me about making sure the voltages are set to the default. Image below shows the highlighted area.

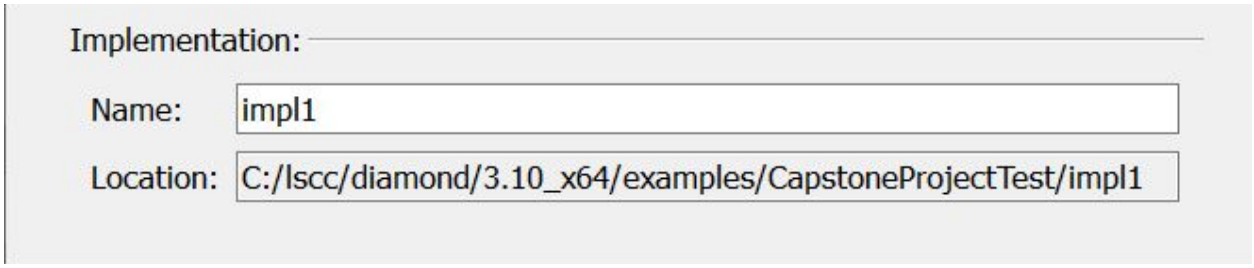


Now we're ready to program. Click on the programmer. Icon is circled below.



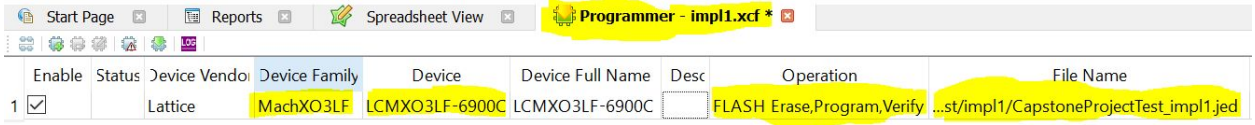
This opens up the programmer. You might get a window asking you to create an implementation. It should have already been created when creating a new project. If it still comes up, just create it then and make sure it's under the same folder that was created for the

project.



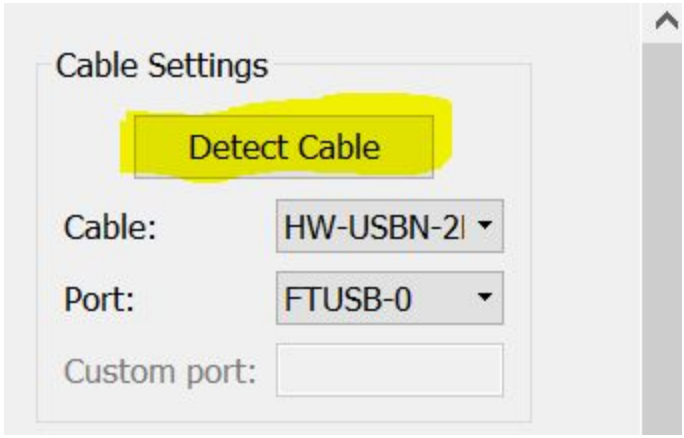
Above shows an example of the file location made.

Now we need to make sure the programmer will program the right board. The image below highlights

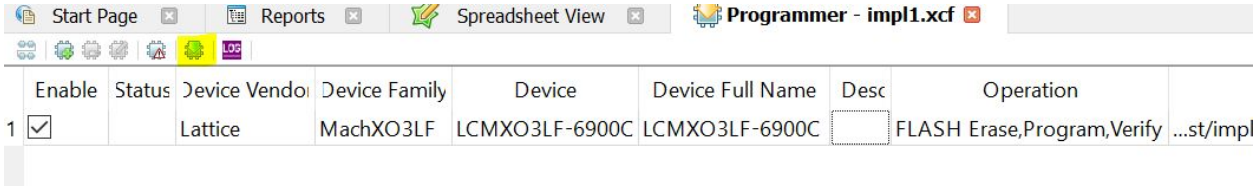


The highlighted sections are essential to keep. You can technically change Operation to do different things to the FPGA. And changing File Name allows you to choose another implementation for example.

After those are set up. Make sure you can detect the cable.



Should detect USB-0.



Highlighted button is the one to press for programming the board.
If the programming was successful, the output should be like below.