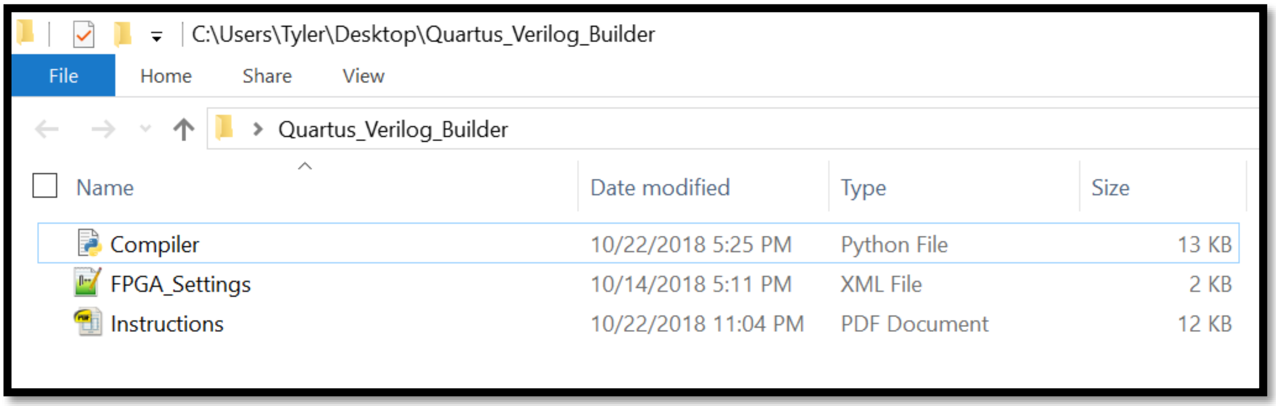
10/22/18

Quartus Verilog Builder Instructions

The Quartus Verilog builder script allows the user to choose which components on the FPGA board they would like to use and creates a Verilog file (.v) with the header already complete. The script also creates a pin assignments file in the comma-separated-values filetype (.csv) with all the necessary pin assignments. This allows the user to save time on the tedious operations of writing the file header and declaring the pin assignments manually.

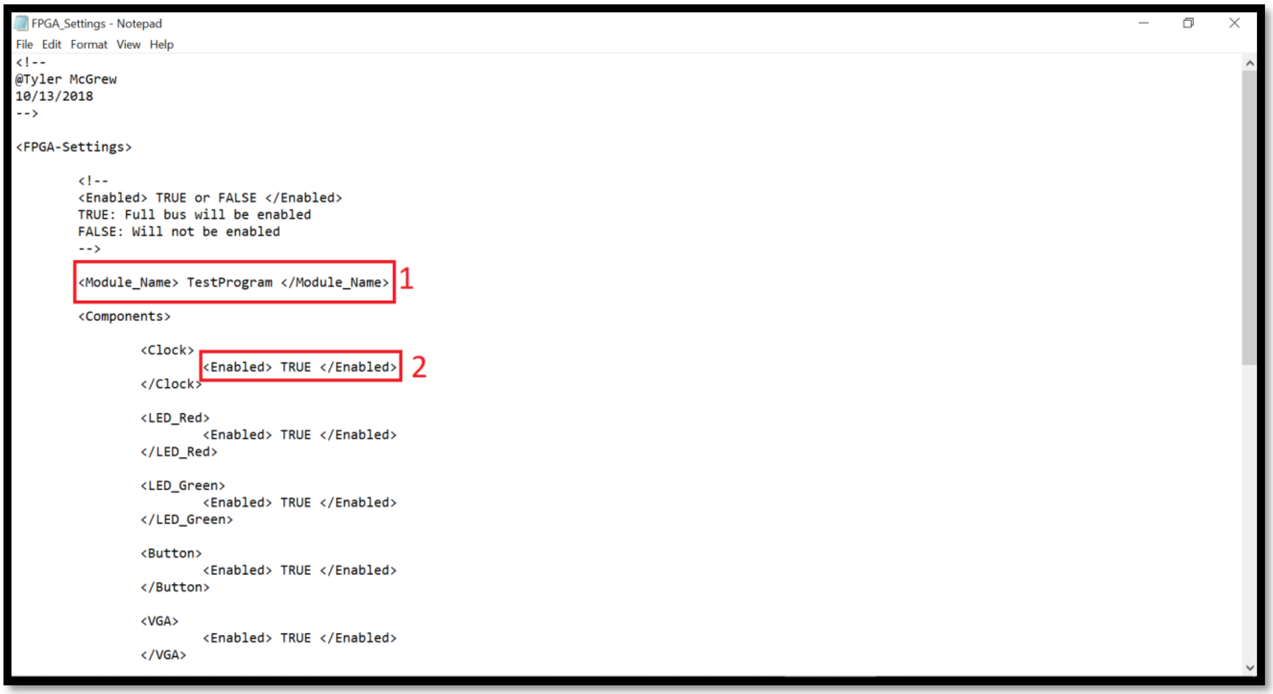
The Quartus Verilog Builder script is written in Python and works with Python 2.7 and Python 3. If you do not have Python installed on your computer, you can find the Windows link here and the Mac link here. The script has only been tested on Windows and Linux, but Python is cross-platform and should work fine on Mac as well.



Name	Date modified	Type	Size
Compiler	10/22/2018 5:25 PM	Python File	13 KB
FPGA_Settings	10/14/2018 5:11 PM	XML File	2 KB
Instructions	10/22/2018 11:04 PM	PDF Document	12 KB

Within the folder containing this instruction file, you will also find two other files. The first - FPGA_Settings - is an XML file with a basic structure allowing the user to choose the name of the Verilog file and the components, both input and output, that they would like to use in their project. The second - Compiler – is the Python script that generates the Verilog file and the pin assignments file.

To begin, open the FPGA_Settings.xml file using a standard text editor, such as Notepad, and you will see something like this:

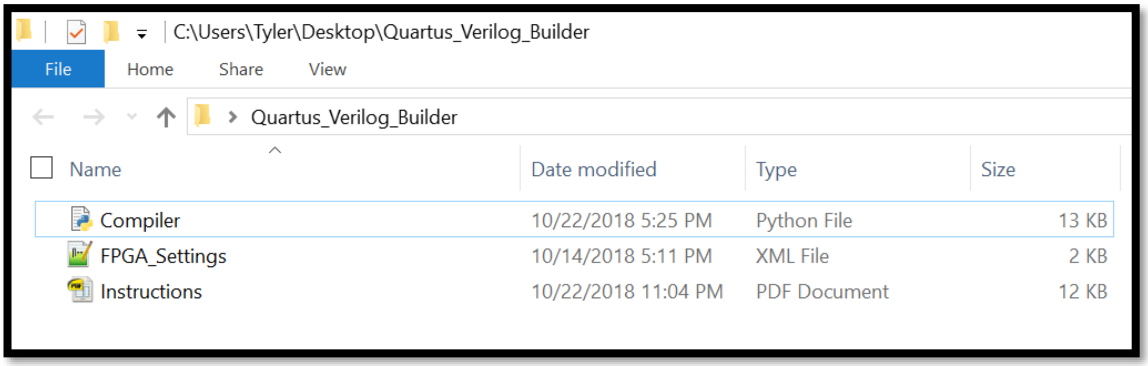


```
<!--
@Tyler McGrew
10/13/2018
-->

<FPGA-Settings>

        <!--
        <Enabled> TRUE or FALSE </Enabled>
        TRUE: Full bus will be enabled
        FALSE: Will not be enabled
        -->

        <Module_Name> TestProgram </Module_Name>

        <Components>

                <Clock>
                        <Enabled> TRUE </Enabled>
                </Clock>

                <LED_Red>
                        <Enabled> TRUE </Enabled>
                </LED_Red>

                <LED_Green>
                        <Enabled> TRUE </Enabled>
                </LED_Green>

                <Button>
                        <Enabled> TRUE </Enabled>
                </Button>

                <VGA>
                        <Enabled> TRUE </Enabled>
                </VGA>
```

1 First, change the Module_Name to what you would like your Verilog file to be named. Currently, the file will be named TestProgram.v if the script is run. Change the text that currently says “TestProgram” to whatever you want the module and file name to be.

2 Next, enable or disable each of the components. Each component contains a line in the form “<Enabled> TRUE </Enabled>” where TRUE will enable that component or component bus, and FALSE will not enable that component. It is a good idea to only enable the components you will need so the Verilog header file is not excessively crowded. The pin assignments will be created for every bus regardless, so you can add other components to the Verilog file manually if needed.

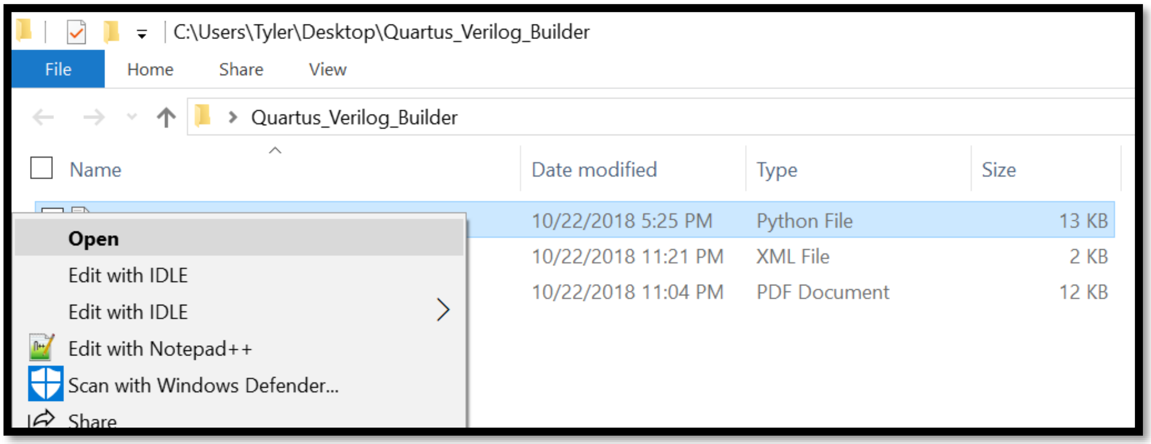
3 Save the file, not changing the file name or location, and exit back to the folder.



C:\Users\Tyler\Desktop\Quartus_Verilog_Builder

Quartus_Verilog_Builder

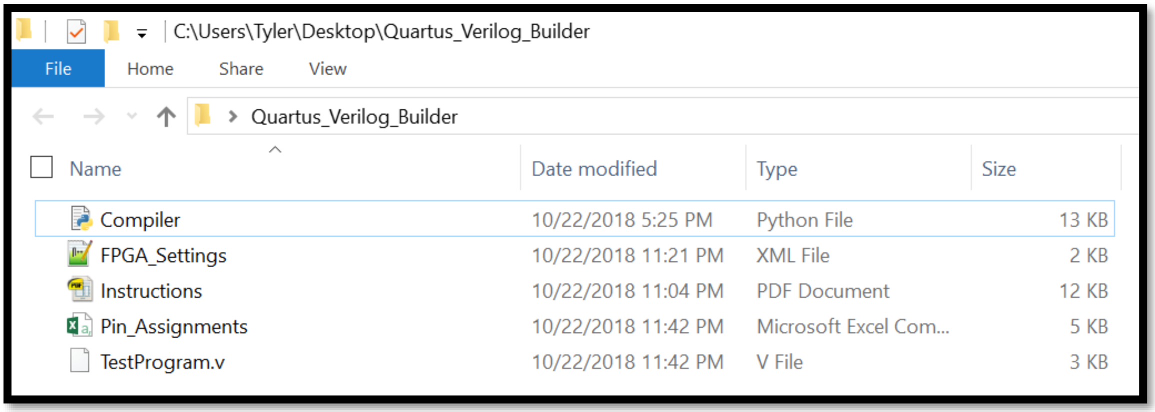
Name	Date modified	Type	Size
Compiler	10/22/2018 5:25 PM	Python File	13 KB
FPGA_Settings	10/14/2018 5:11 PM	XML File	2 KB
Instructions	10/22/2018 11:04 PM	PDF Document	12 KB



C:\Users\Tyler\Desktop\Quartus_Verilog_Builder

Quartus_Verilog_Builder

Name	Date modified	Type	Size
	10/22/2018 5:25 PM	Python File	13 KB
	10/22/2018 11:21 PM	XML File	2 KB
	10/22/2018 11:04 PM	PDF Document	12 KB



C:\Users\Tyler\Desktop\Quartus_Verilog_Builder

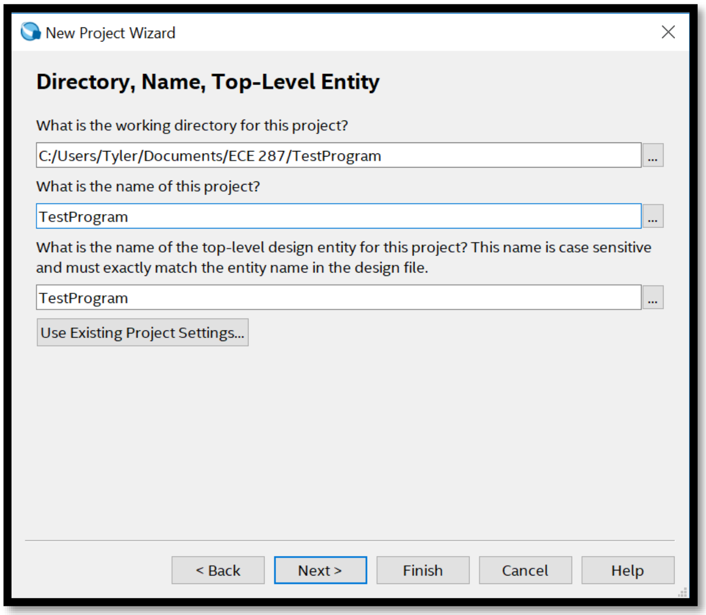
Quartus_Verilog_Builder

Name	Date modified	Type	Size
Compiler	10/22/2018 5:25 PM	Python File	13 KB
FPGA_Settings	10/22/2018 11:21 PM	XML File	2 KB
Instructions	10/22/2018 11:04 PM	PDF Document	12 KB
Pin_Assignments	10/22/2018 11:42 PM	Microsoft Excel Com...	5 KB
TestProgram.v	10/22/2018 11:42 PM	V File	3 KB

Run the Compiler Python script by double clicking it or right clicking and opening/running it. This will create the Pin_Assignments (.csv) file and the Verilog (.v) file with a pre-created header.

Create your Quartus project like normal, with three notes –

1. Name the project the same name as what you named the Verilog file in the FPGA_Settings. In this example, that would be "TestProgram".



New Project Wizard

Directory, Name, Top-Level Entity

What is the working directory for this project?

C:/Users/Tyler/Documents/ECE 287/TestProgram

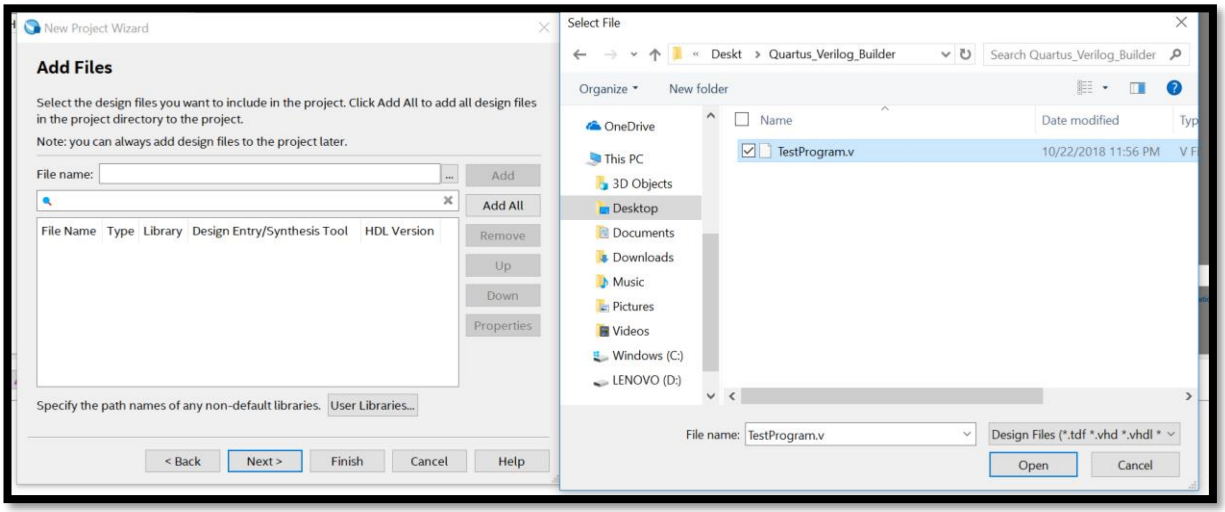
What is the name of this project?

TestProgram

What is the name of the top-level design entity for this project? This name is case sensitive and must exactly match the entity name in the design file.

TestProgram

2. During the "Add Files" step, include the Verilog file you just created by running the script.



New Project Wizard

Add Files

Select the design files you want to include in the project. Click Add All to add all design files in the project directory to the project.

Note: you can always add design files to the project later.

File name:

File Name	Type	Library	Design Entry/Synthesis Tool	HDL Version

Specify the path names of any non-default libraries.

Select File

Deskt > Quartus_Verilog_Builder

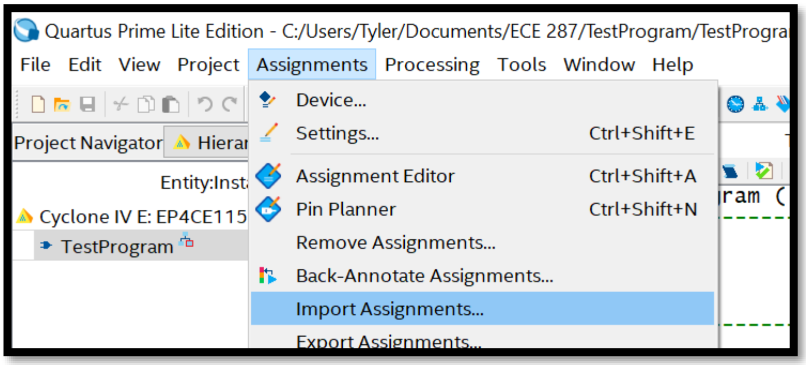
Search Quartus_Verilog_Builder

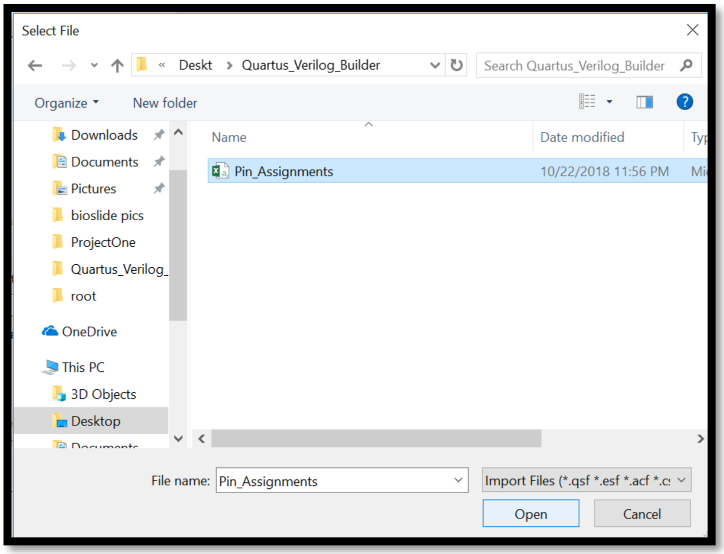
Name	Date modified	Typ
☑ TestProgram.v	10/22/2018 11:56 PM	V F

File name: TestProgram.v

Design Files (*.tdf *.vhd *.vhdl *

3. Import the (.csv) pin assignments file by clicking Assignments > Import Assignments and select the Pin_Assignments file that was created before.


Quartus Prime Lite Edition - C:/Users/Tyler/Documents/ECE 287/TestProgram/TestProgra



Pin_Assignments 10/22/2018 11:56 PM Mi

File name: Pin_Assignments

Import Files (*.qsf *.esf *.acf *.c:

The project should now include a Verilog file with a header, as well as defined pin assignments for each of the components- inputs and outputs that you need.

If you have any questions, you can email me at mcgrewtj@miamioh.edu