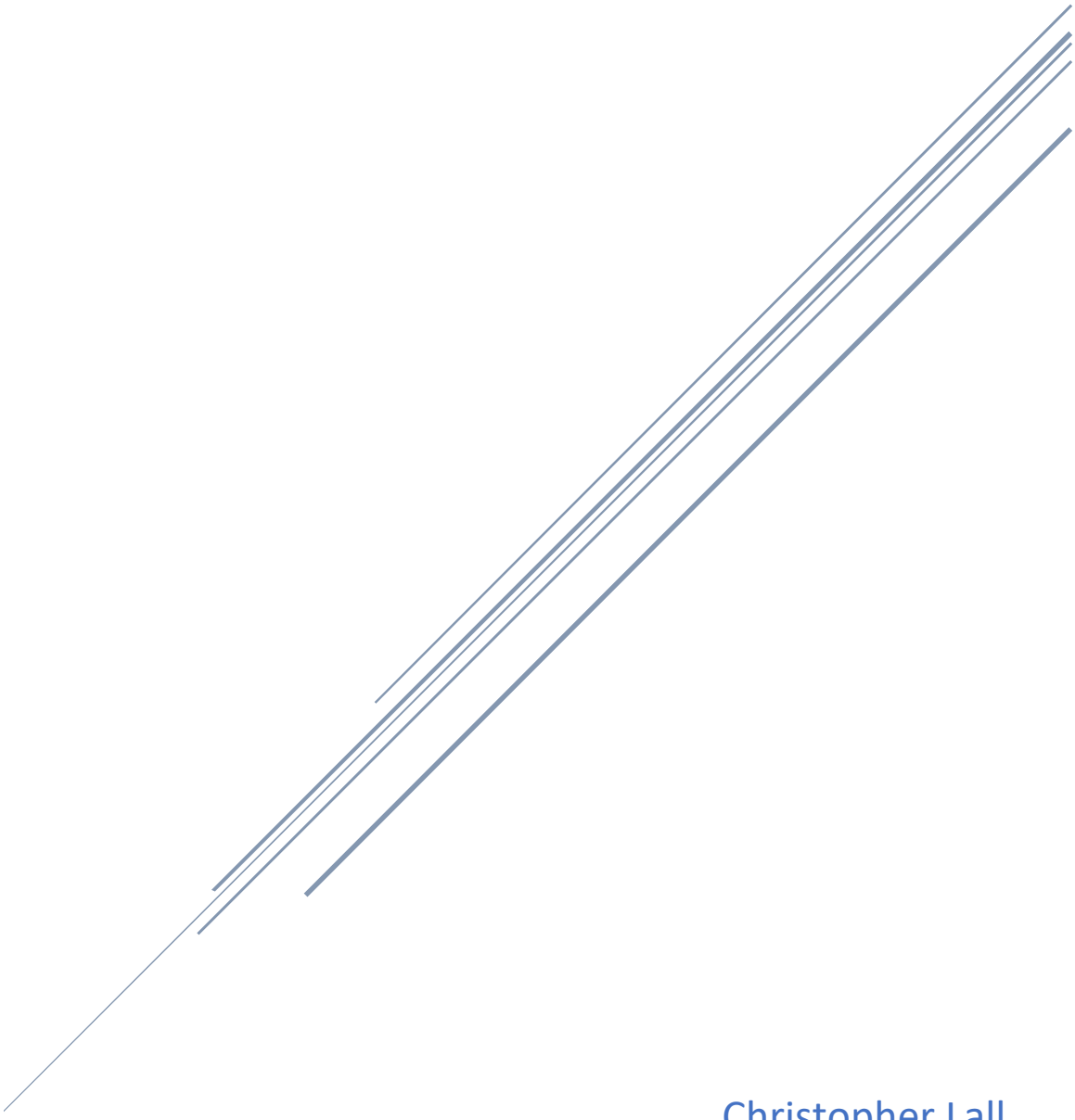
LAB ASSIGNMENT #1 - MUX

Due 2/13/2022



Christopher Lall
CSc 342/343 – Professor Gertner

Table of Contents

Section 1: Objective:

The objective of this lab is to familiarize myself with Quartus Prime Lite and create 2:1 multiplexer where each signal is one bit and 2:1 multiplexer where each signal is 32 bits and the selector signal is one bit.

Section 2: Tutorial Screenshots:

Following the tutorial pdf, we can learn how to setup and become familiar with Quartus.

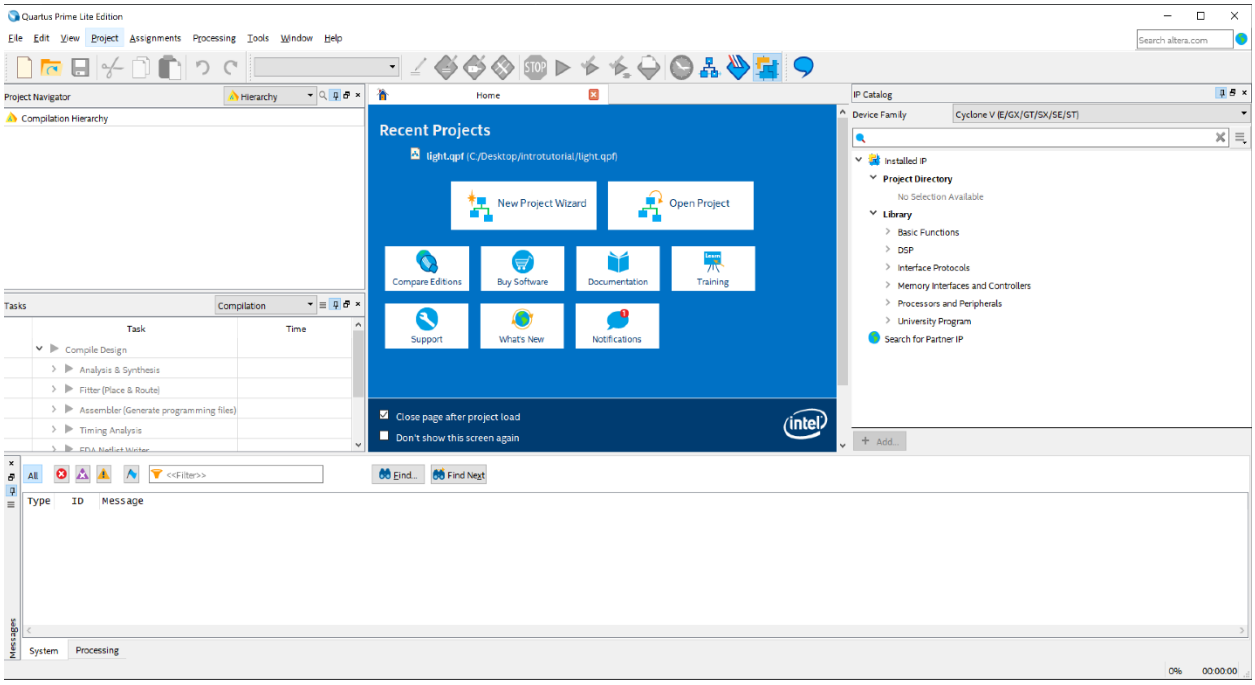


Figure 1. Opening Screen of Quartus

Now, we can get familiar with the file menu that will help us navigate:

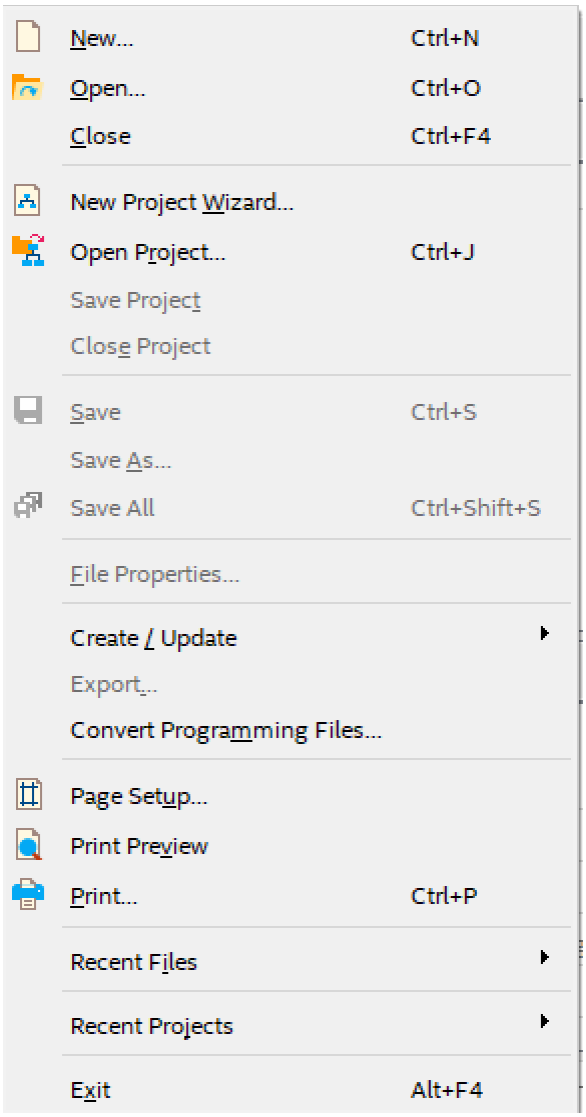


Figure 2. File Menu

From here we can create a directory.

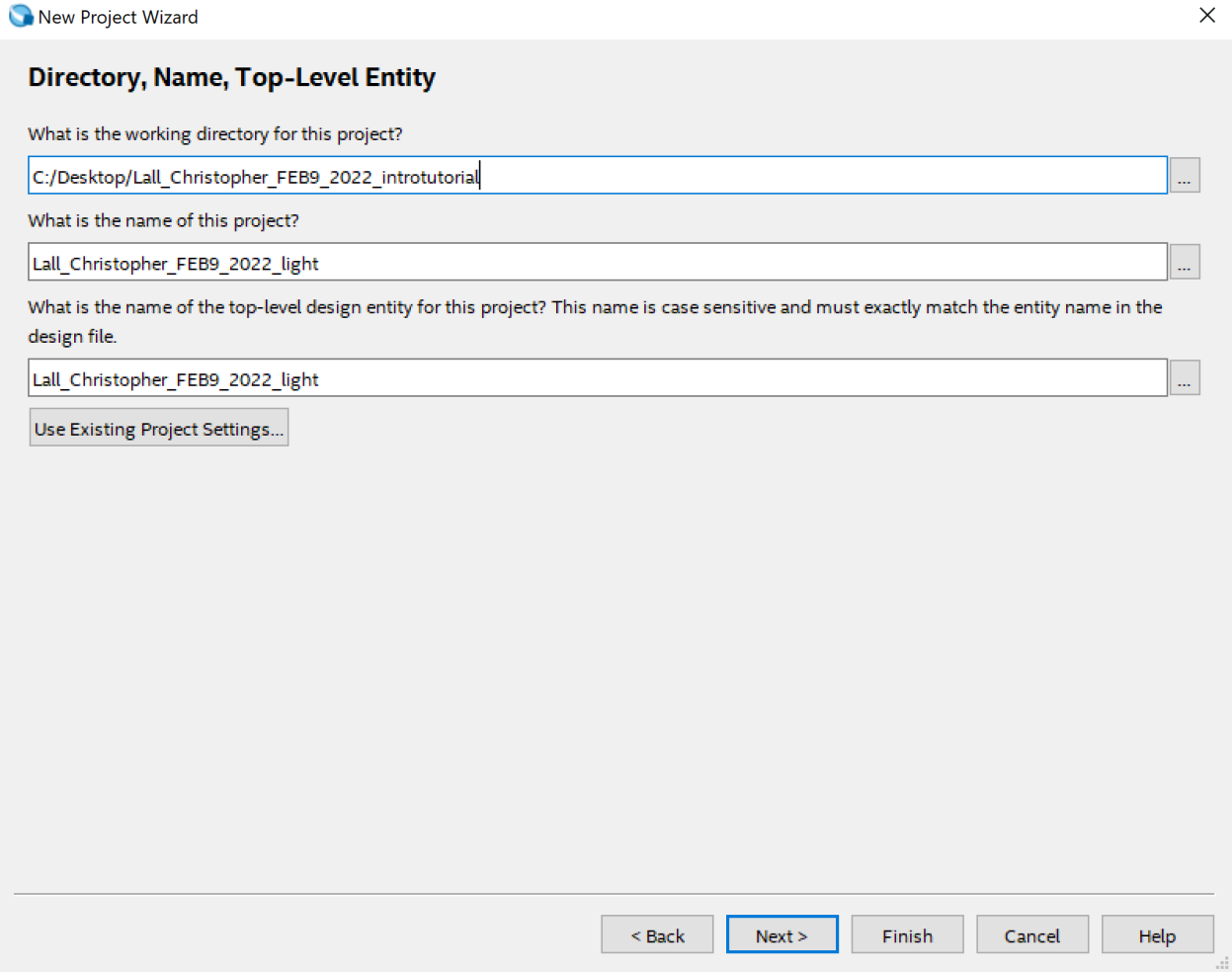


Figure 3. Creating Directory

Since we want to create a directory, we click next and in figure 4, below, we click yes to confirm.

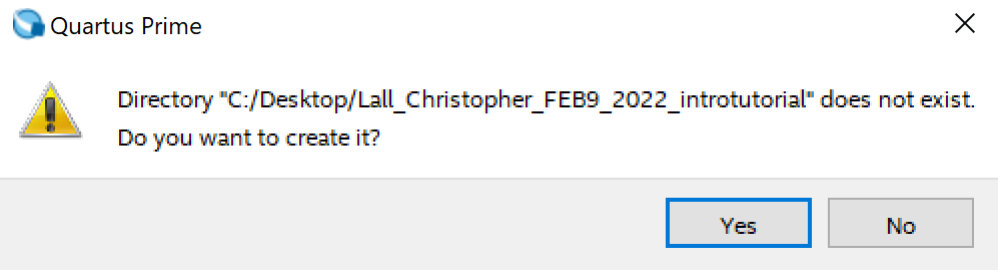


Figure 4. Confirmation of creating directory

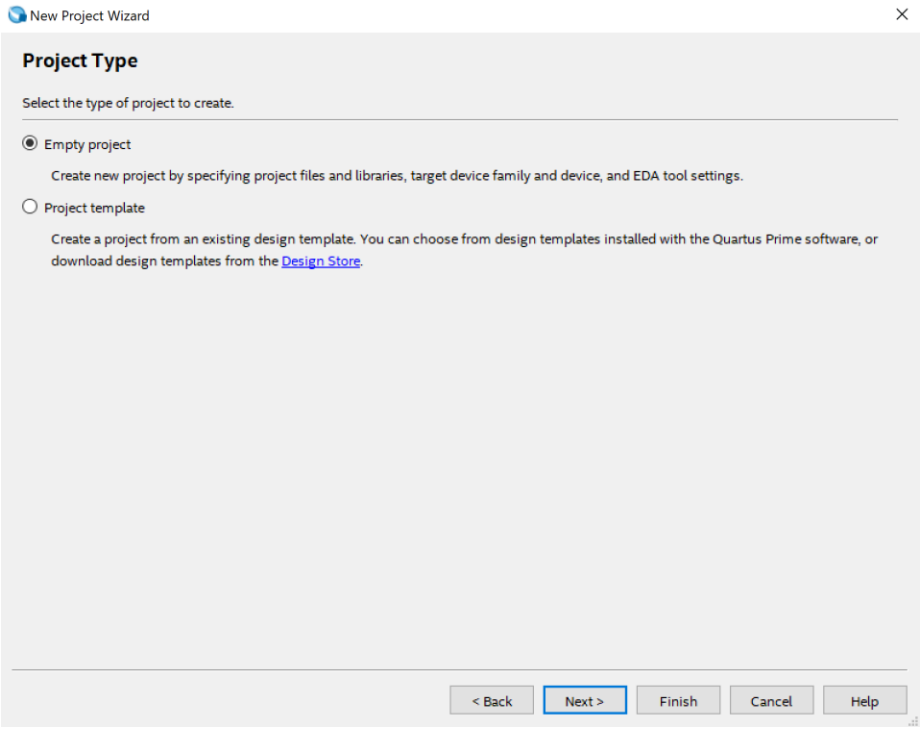


Figure 5. Selecting Project Type

Once we click next we arrive at figure 6.

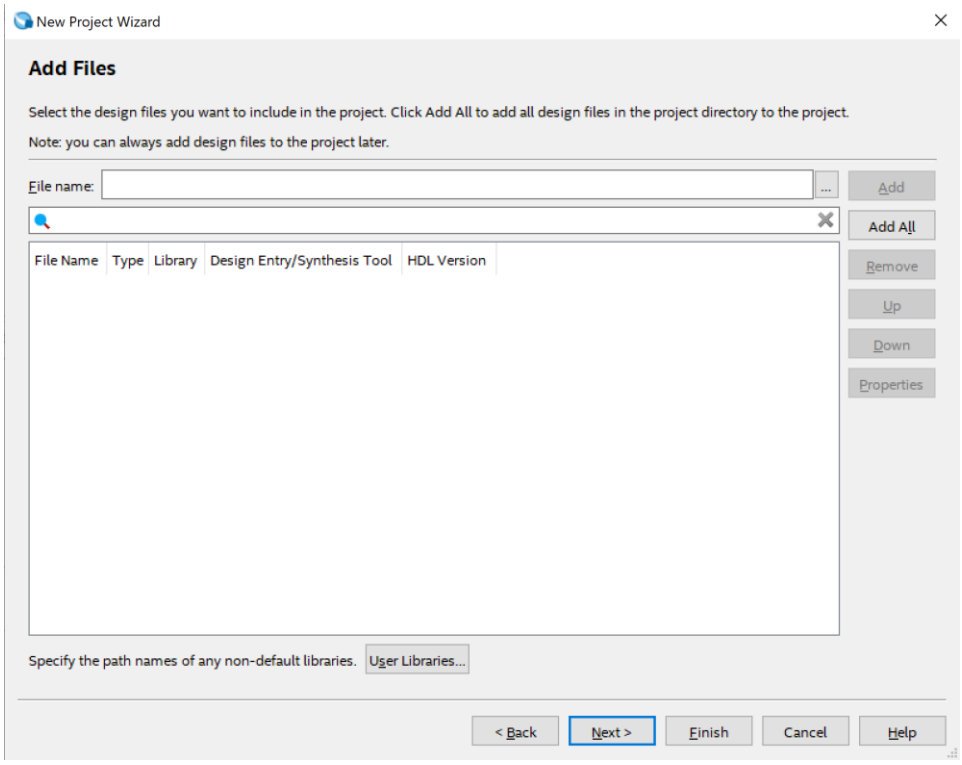


Figure 6. Adding Files

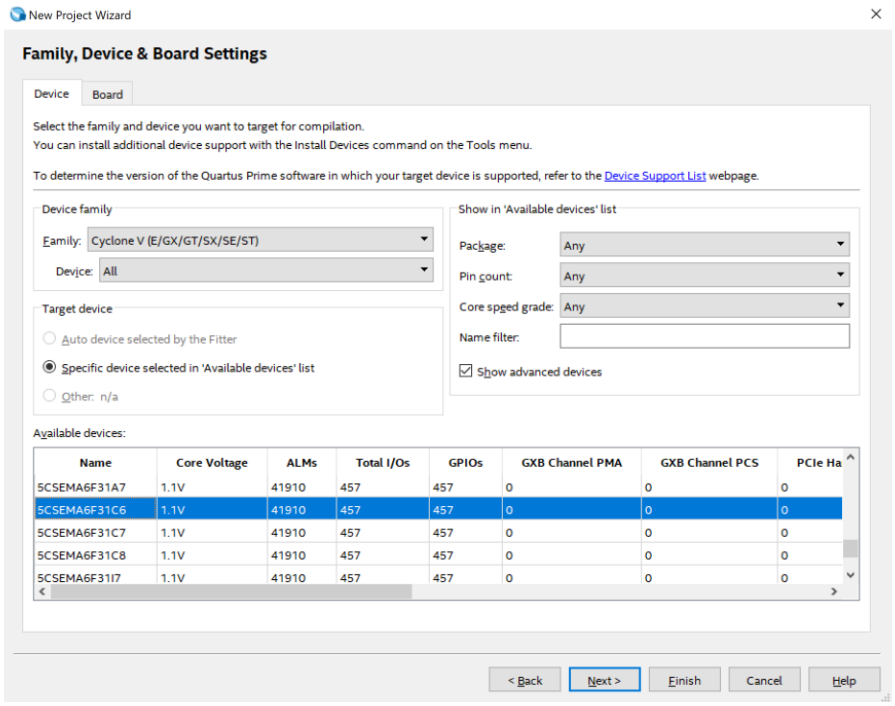


Figure 7. Choosing Family, Device, and Board Settings

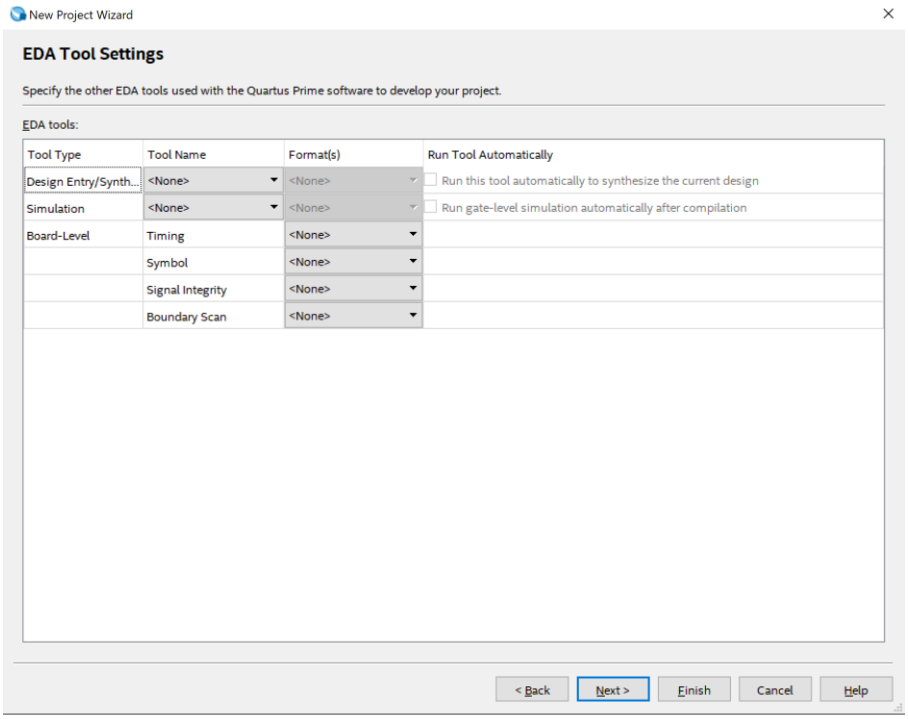


Figure 8. EDA Tool Settings

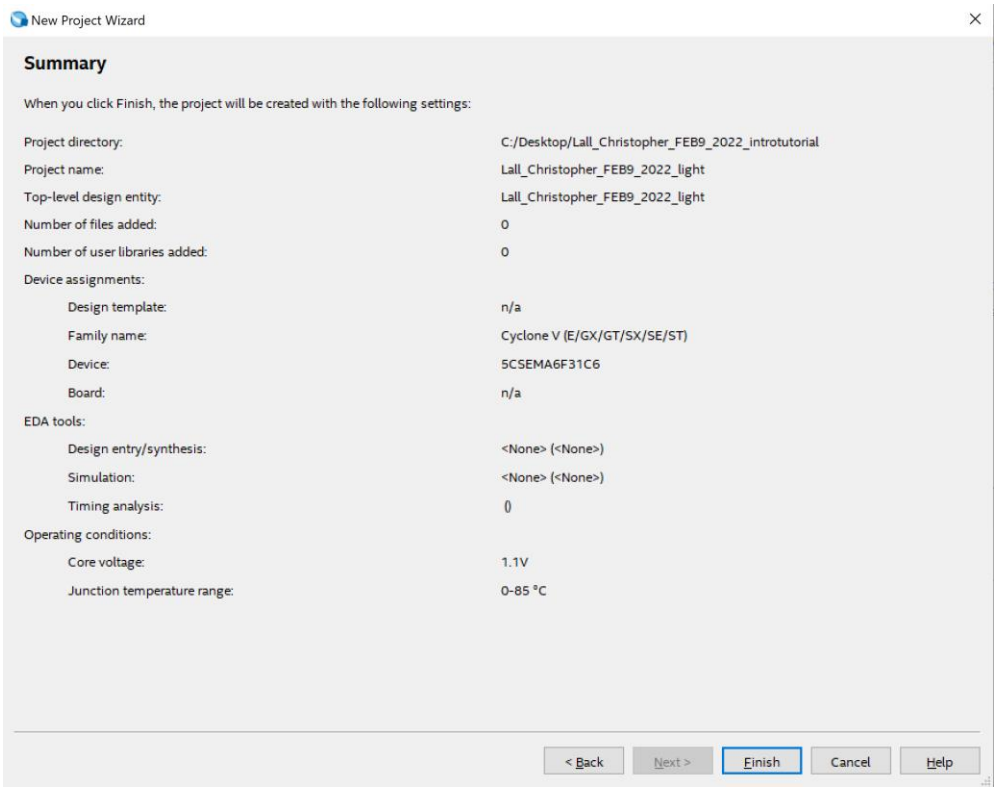


Figure 9. Summary

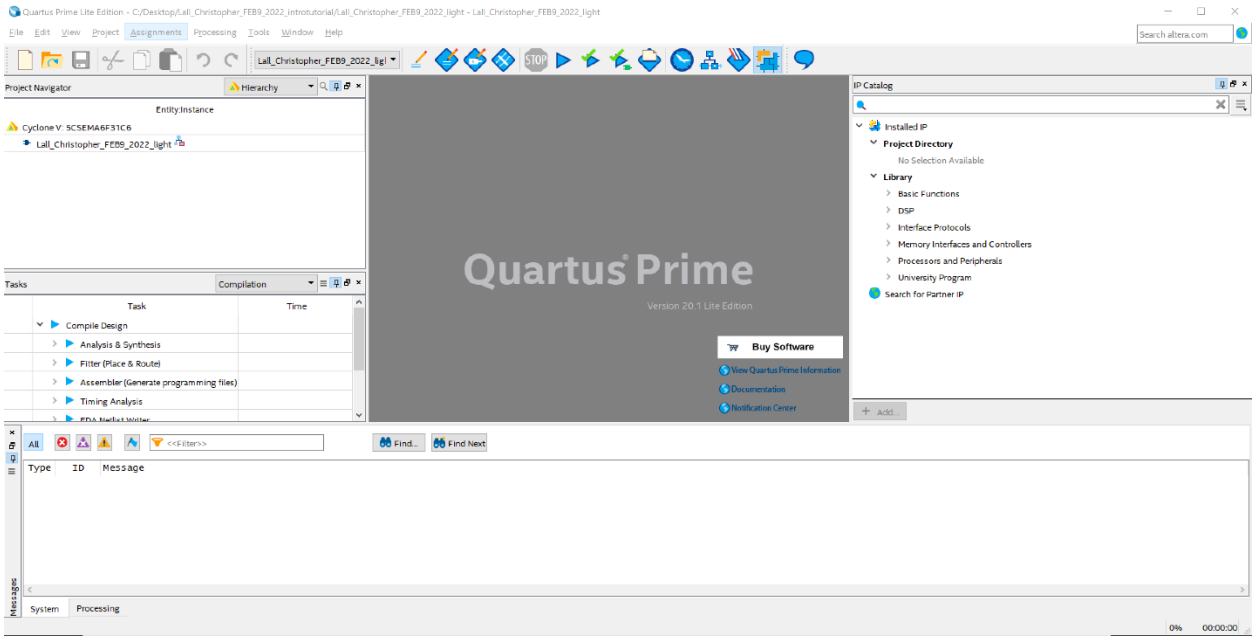


Figure 10. Screen after Finishing

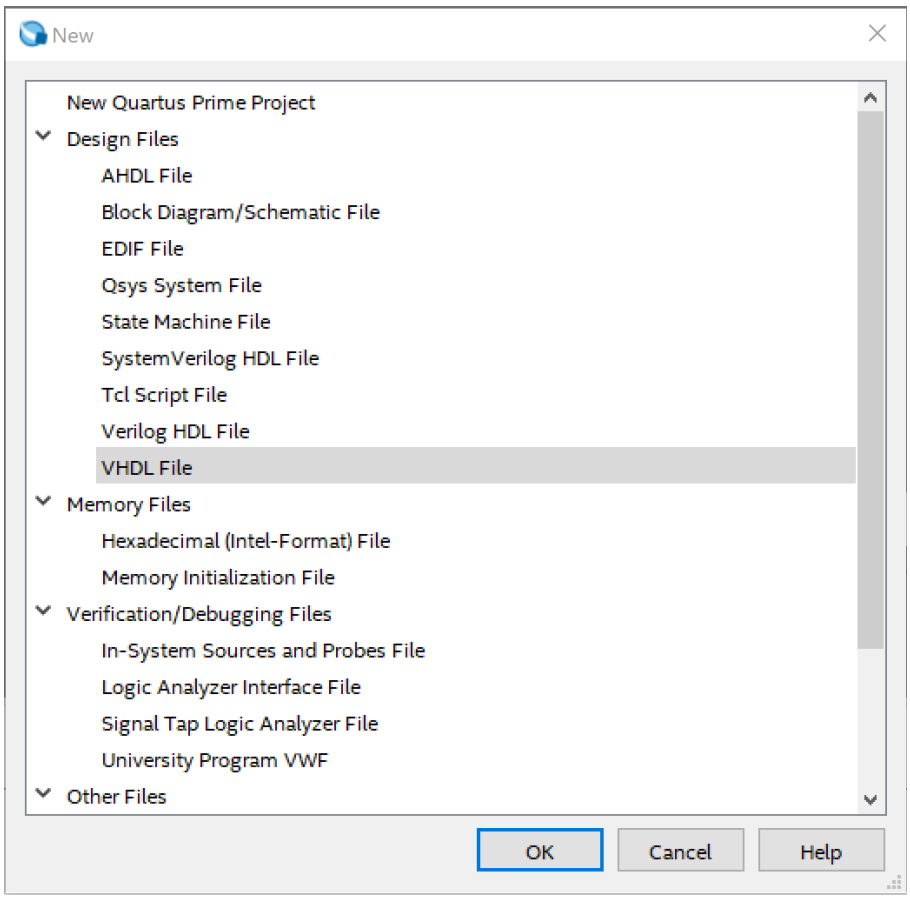


Figure 11. VHDL Selection

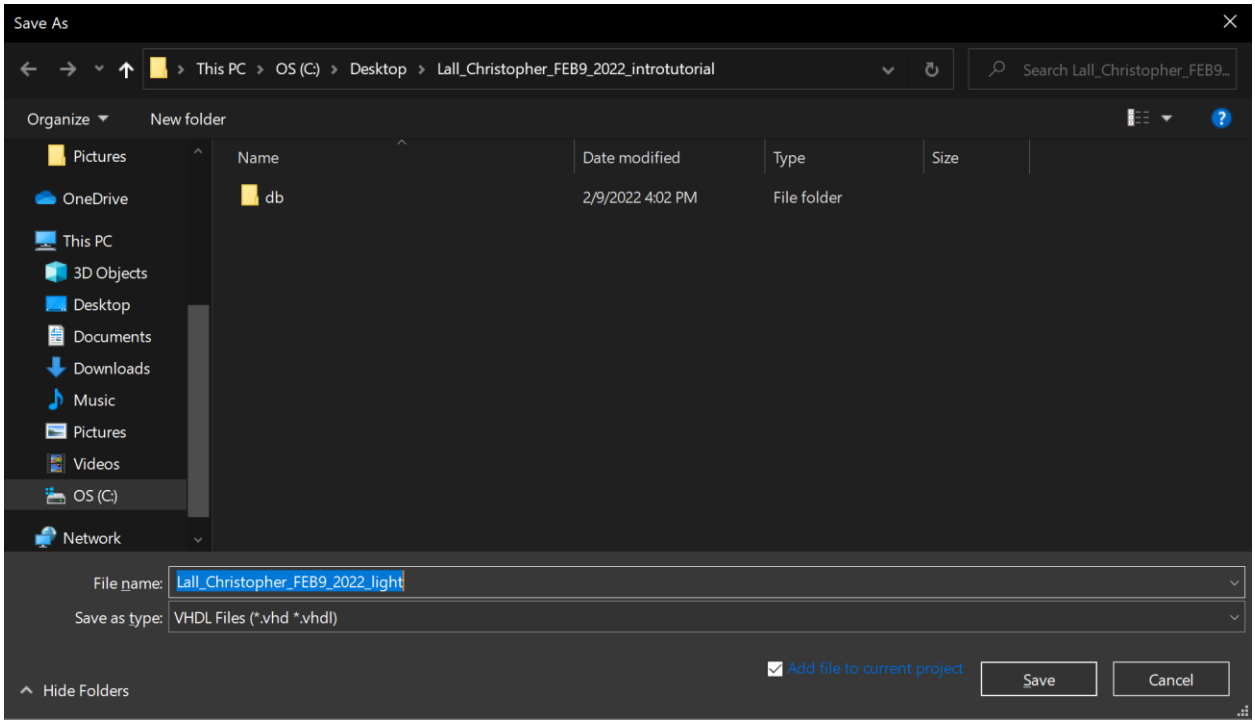


Figure 12. Select File

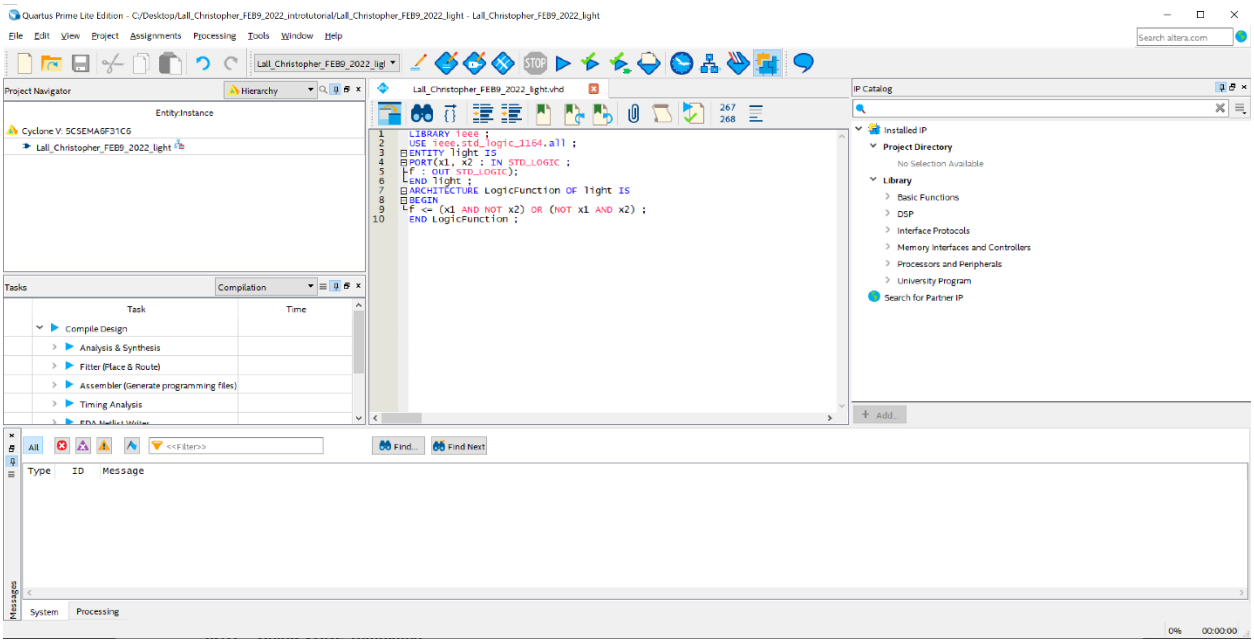


Figure 13. VHDL Edit

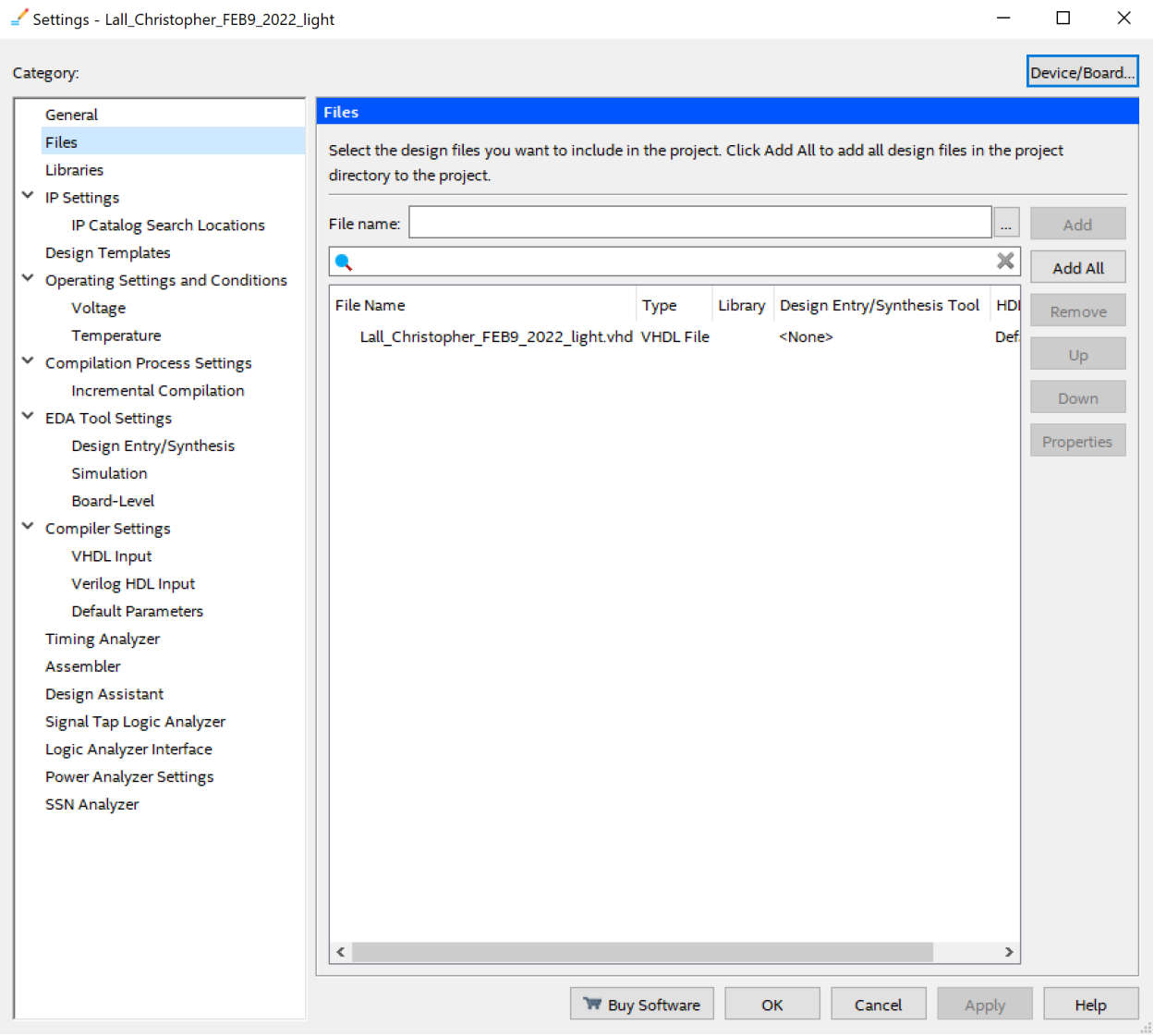


Figure 14. FILE APPLY

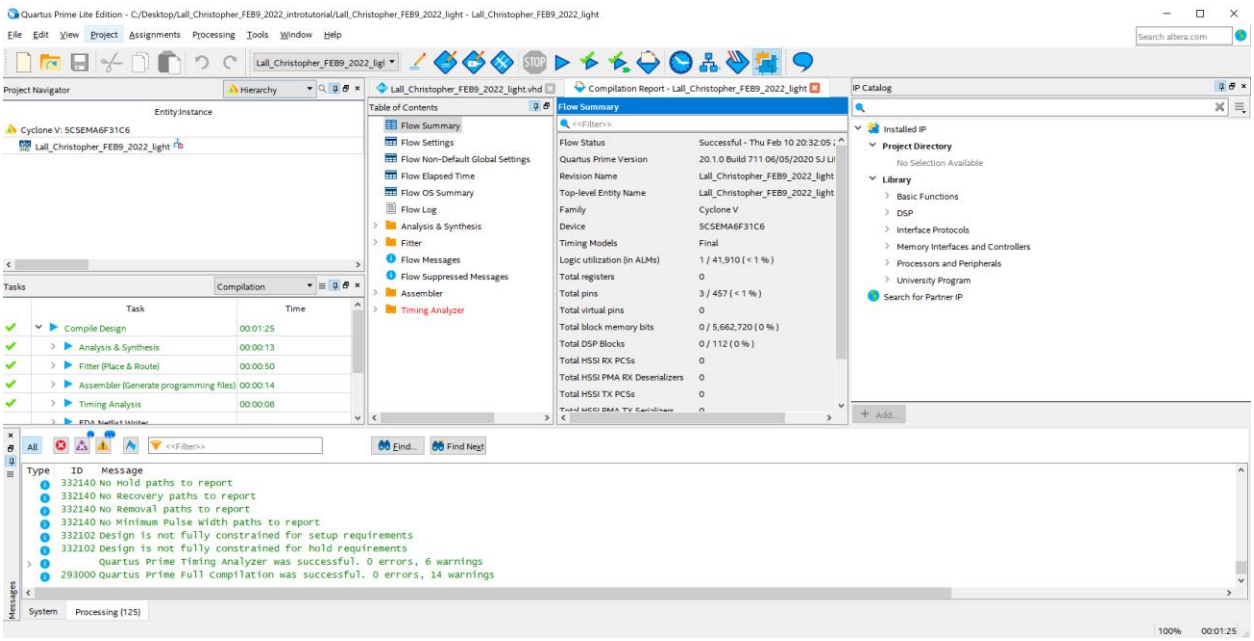


Figure 15. Compilation

After compilation, I am going to do pin assignments:

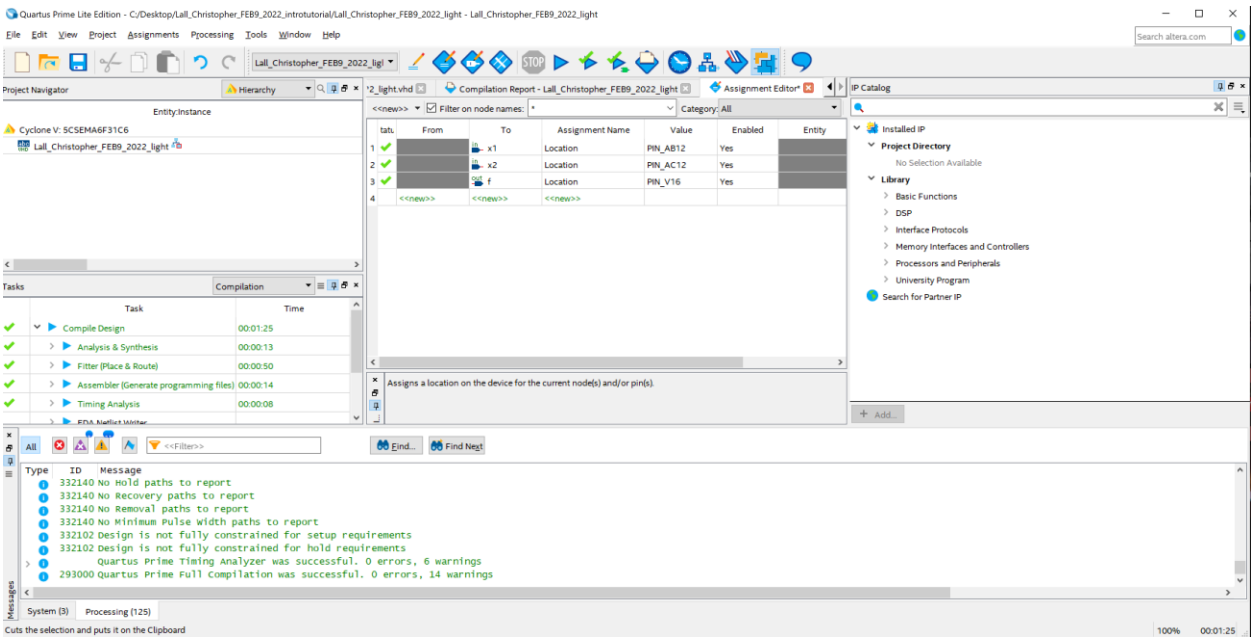


Figure 16. Pin Assignment

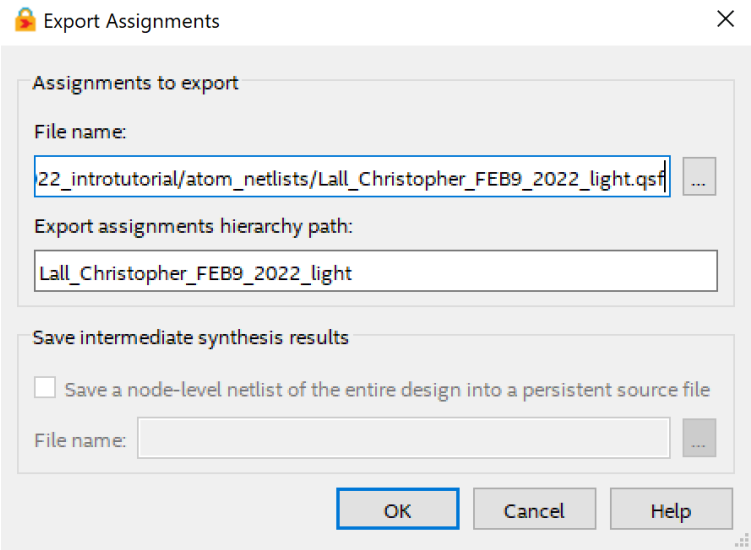


Figure 17. Export Assignment

Section 3: 2:1 Mux Screenshots

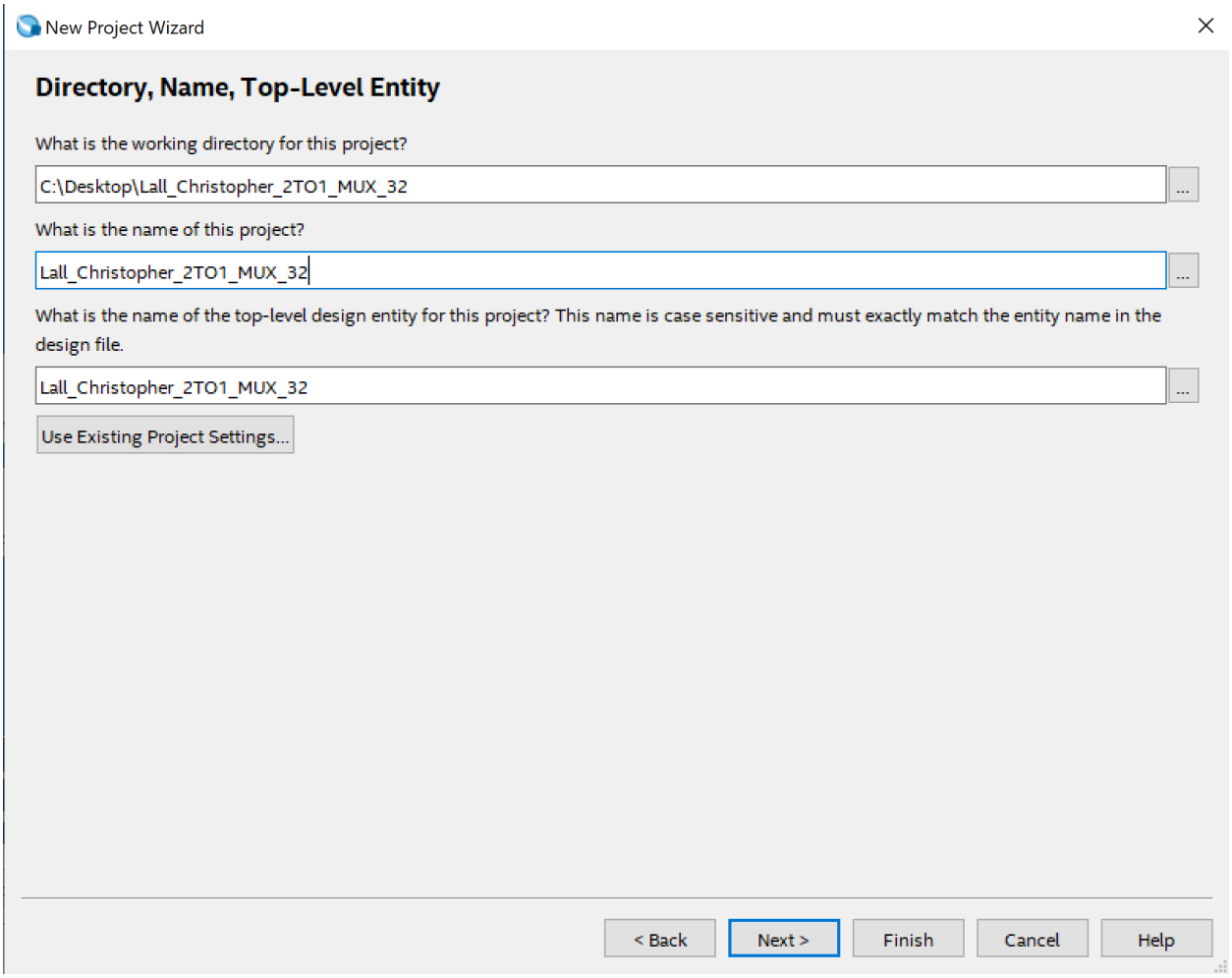


Figure 18. Create directory for MUX project

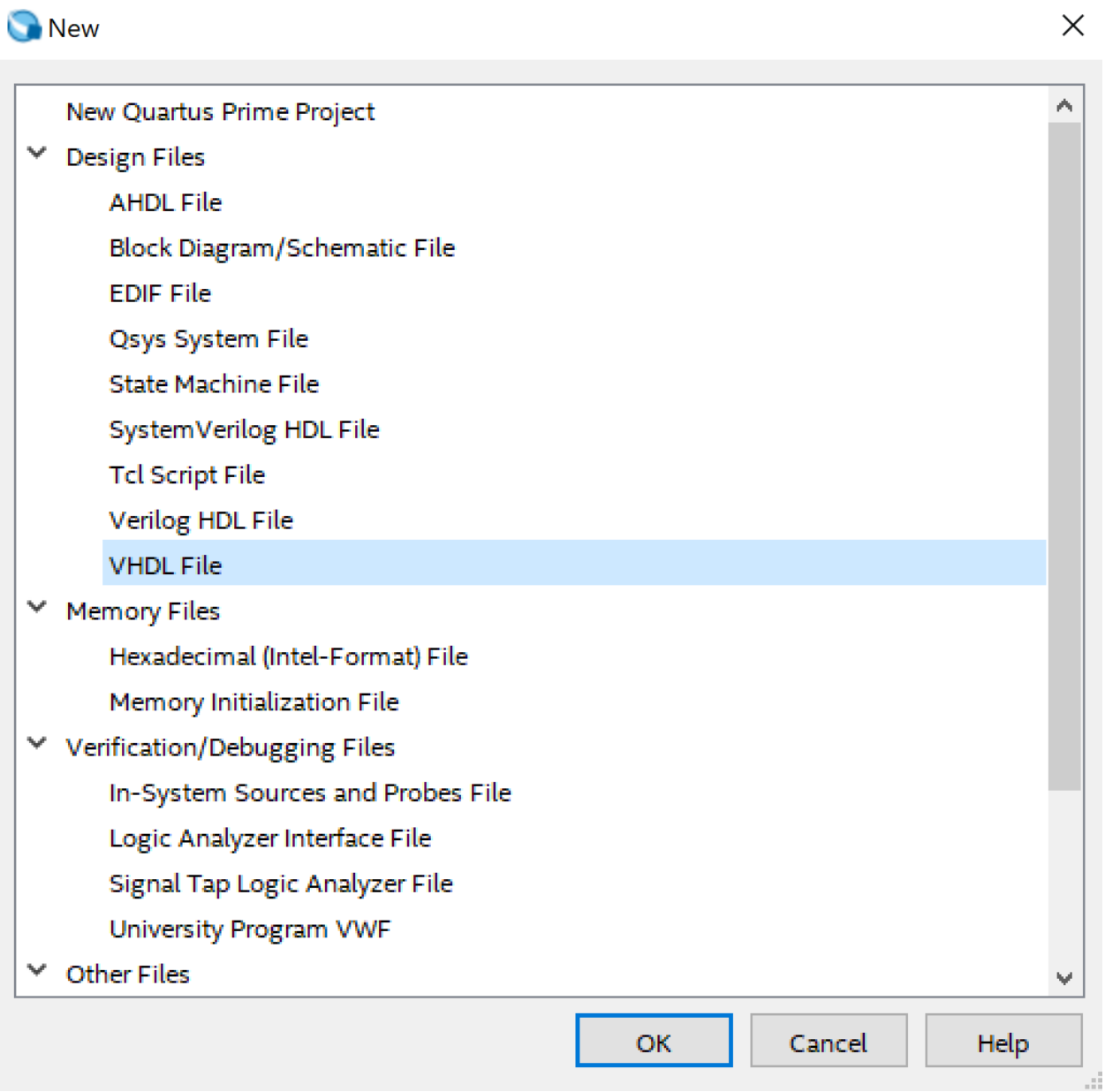


Figure 19. Creating VHDL File for MUX

Once we create this file, we can being coding our design to compile.

Section 3.1: 2:1 Mux VDHL Code

Our VHDL file for the 2:1 Mux was created by:

```
-- VERSION      "Version 20.1.0 Build 711 06/05/2020 SJ Lite Edition"
-- CREATED      "Thu Feb 10 22:14:26 2022"

LIBRARY ieee;
USE ieee.std_logic_1164.all;

LIBRARY work;

ENTITY Lall_Christopher_FEB9_2022_2TO1_MUX IS
   PORT
   (
      Lall_Christopher_A :  IN  STD_LOGIC;
      Lall_Christopher_B :  IN  STD_LOGIC;
      Lall_Christopher_S :  IN  STD_LOGIC;
      Lall_Christopher_Out :  OUT  STD_LOGIC
   );
END Lall_Christopher_FEB9_2022_2TO1_MUX;

ARCHITECTURE bdf_type OF Lall_Christopher_FEB9_2022_2TO1_MUX IS

SIGNAL   SYNTHESIZED_WIRE_0 :  STD_LOGIC;
SIGNAL   SYNTHESIZED_WIRE_1 :  STD_LOGIC;
SIGNAL   SYNTHESIZED_WIRE_2 :  STD_LOGIC;


BEGIN



SYNTHESIZED_WIRE_0 <= NOT(Lall_Christopher_S);



SYNTHESIZED_WIRE_1 <= Lall_Christopher_S AND Lall_Christopher_B;


SYNTHESIZED_WIRE_2 <= Lall_Christopher_A AND SYNTHESIZED_WIRE_0;


Lall_Christopher_Out <= SYNTHESIZED_WIRE_1 OR SYNTHESIZED_WIRE_2;


END bdf_type;
```

Figure 20. 2:1 MUX Vhdl

Section 3.2: 2:1 Mux VDHL Code Compilation

Once we compile this code, we get a compilation report. See figure, 19.

Flow Summary

<<Filter>>

Flow Status	Successful - Thu Feb 10 21:49:35 2022
Quartus Prime Version	20.1.0 Build 711 06/05/2020 SJ Lite Edition
Revision Name	Lall_Christopher_FEB9_2022_2TO1_MUX
Top-level Entity Name	Lall_Christopher_FEB9_2022_2TO1_MUX
Family	Cyclone V
Device	5CSEMA6F31C6
Timing Models	Final
Logic utilization (in ALMs)	1 / 41,910 (< 1 %)
Total registers	0
Total pins	4 / 457 (< 1 %)
Total virtual pins	0
Total block memory bits	0 / 5,662,720 (0 %)
Total DSP Blocks	0 / 112 (0 %)
Total HSSI RX PCSs	0
Total HSSI PMA RX Deserializers	0
Total HSSI TX PCSs	0
Total HSSI PMA TX Serializers	0
Total PLLs	0 / 6 (0 %)
Total DLLs	0 / 4 (0 %)

Figure 21. Compilation report for 2:1 MUX

Here, we can see that the compilation report was successful as it did not throw any errors.

Section 4: 2:1 MUX 32bit VHDL code

```
1   LIBRARY ieee;
2   USE ieee.std_logic_1164.all;
3
4   LIBRARY work;
5
6
7
8   entity Lall_Christopher_2TO1_MUX_32 is
9           Port(CLK : in STD_LOGIC;
10          A : in STD_LOGIC_VECTOR (31 downto 0);
11          B : in STD_LOGIC_VECTOR (31 downto 0);
12          X : out STD_LOGIC_VECTOR (31 downto 0));
13  end Lall_Christopher_2TO1_MUX_32;
14  architecture Behavioral of Lall_Christopher_2TO1_MUX_32 is
15  begin
16      X <= A when (CLK = '1') else B;
17  end Behavioral;
```

Figure 22. VHDL Code for MUX 32Bit

Section 4.1 Mux 32bit VDHL Code Compilation

Flow Status	Successful - Fri Feb 11 18:16:55 2022
Quartus Prime Version	20.1.0 Build 711 06/05/2020 SJ Lite Edition
Revision Name	Lall_Christopher_2TO1_MUX_32
Top-level Entity Name	Lall_Christopher_2TO1_MUX_32
Family	Cyclone V
Device	5CSEMA6F31C6
Timing Models	Final
Logic utilization (in ALMs)	17 / 41,910 (< 1 %)
Total registers	0
Total pins	97 / 457 (21 %)
Total virtual pins	0
Total block memory bits	0 / 5,662,720 (0 %)
Total DSP Blocks	0 / 112 (0 %)
Total HSSI RX PCSs	0
Total HSSI PMA RX Deserializers	0
Total HSSI TX PCSs	0
Total HSSI PMA TX Serializers	0
Total PLLs	0 / 6 (0 %)
Total DLLs	0 / 4 (0 %)

Figure 21. Compilation of MUX 32 VHDL Code 32bit

Section 5: Explanation

The tutorial screenshots was a demonstration to familiarize myself with Quartus Prime Lite. By following the tutorial, I also got screenshots very similar, if not the same, as the given pdf. In the screenshots, you can see how I started a project, named my project, started a file,

named my file, create specific directories, open menus, select important options, open files, and run code.

Moving on the 2:1 Mux where each signal is one bit, I use what I learned in the tutorial to help me create and run my code. I start by creating a project. The screenshots in section 3 show how I was able to create a directory to put my project in. Within that project I was able to create a VHDL file where I will code the MUX where each signal is one bit. Once that is done, I simply compile the program, and since there were no errors, I move on.

Very similarly to the 2:1 Mux where each signal is one bit, section 4 focuses on another Mux where each signal is 32 bits, and the selector signal is one bit. Using the exact same steps for section 3 (which I did not include in section 4 because it is exactly the same apart from the naming convention), I was able to open a new directory, and create a VHDL file where I coded this new Mux. Once that was done, I then compiled the code and got no error which means it was a success for me.

Section 6: Conclusion

The introduction to Quartus Prime Lite via the tutorial helped me to familiarize myself with all options, menus, and ways to do important things. Using the tutorial, I created 2 mux's. One Mux was a Mux where each signal is one bit, and the second Mux was a Mux where each signal is 32bits, and the selector signal is one bit. This means that the project was doable and I was able to create code that compiled, as seen in the screenshots.