

MUX Project Using VHDL

MdShahid Bin Emdad

February 13th, 2022

CSC 34200/34300

Table of Content

[Objective:](#)

[Description of Specifications, and Functionality:](#)

[Explanation:](#)

[Conclusion:](#)

Objective:

The objective of this assignment is to create a digital circuit using VHDL by following an introductory tutorial of Quartus and subsequently to create two 2:1 multiplexers (1. where each signal is one bit and 2. where each signal is 32 bits and the selector signal is one bit) by VHDL.

Description of Specifications, and Functionality:

The digital system I used in this assignment is Quartus Prime 20.1.1 and ModelSimSetup-20.1.1. There two packages needed are cyclonev and cyclonevi (both versions are 20.1.1). In the VHDL editor, I wrote my VHDL code to get the circuit output.

I. Introtutorial

First, we I find the software in my Windows OS.

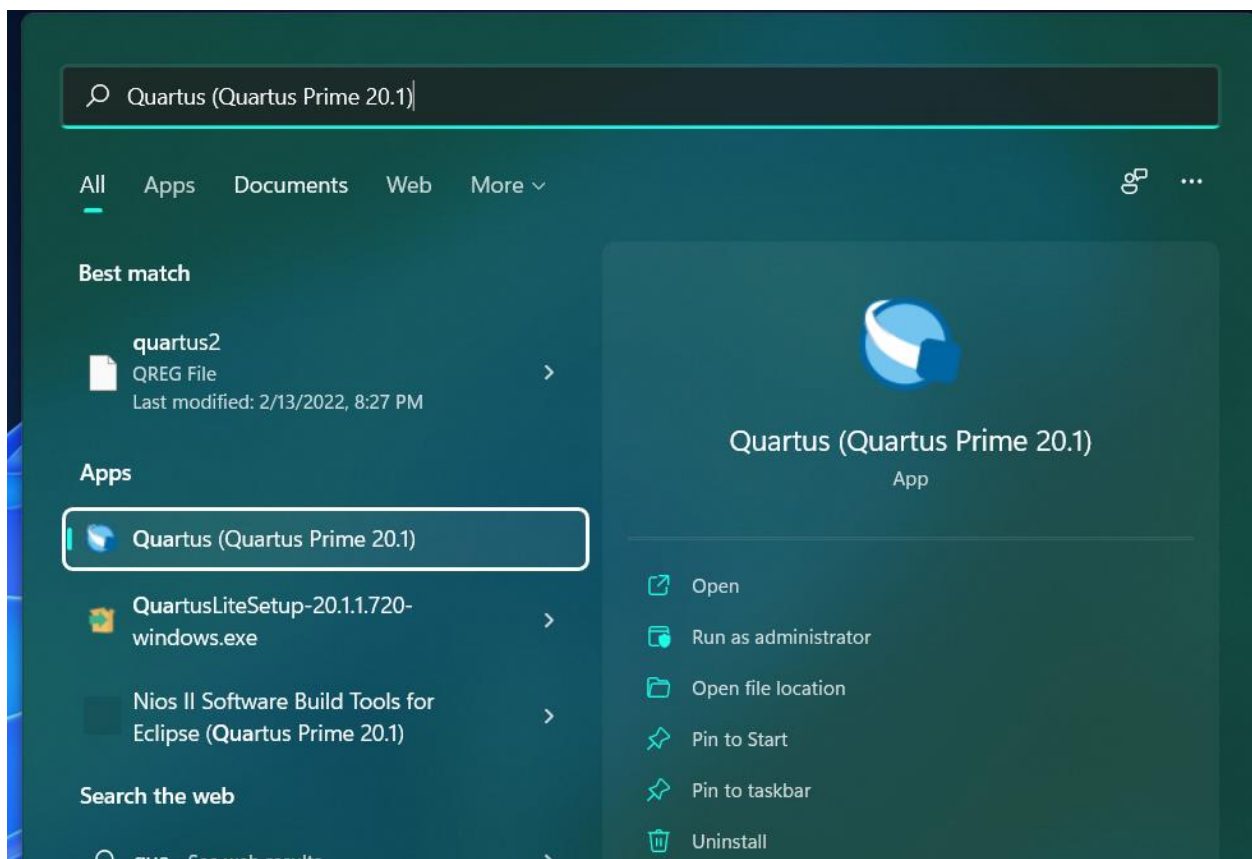


Figure 1: Quartus Software

In figure 2, the file menu is visible on the left side of the image.

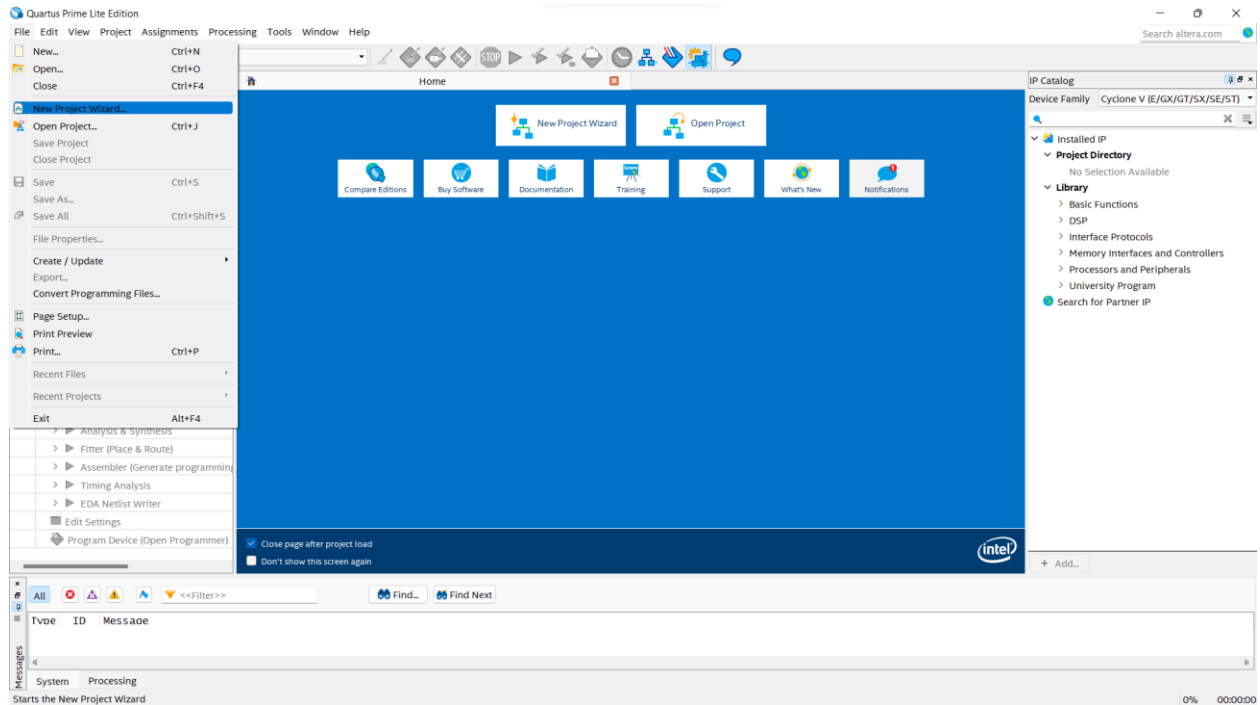


Figure 2: File Menu

In figure 3, I clicked on working directory to choose where to keep the project.

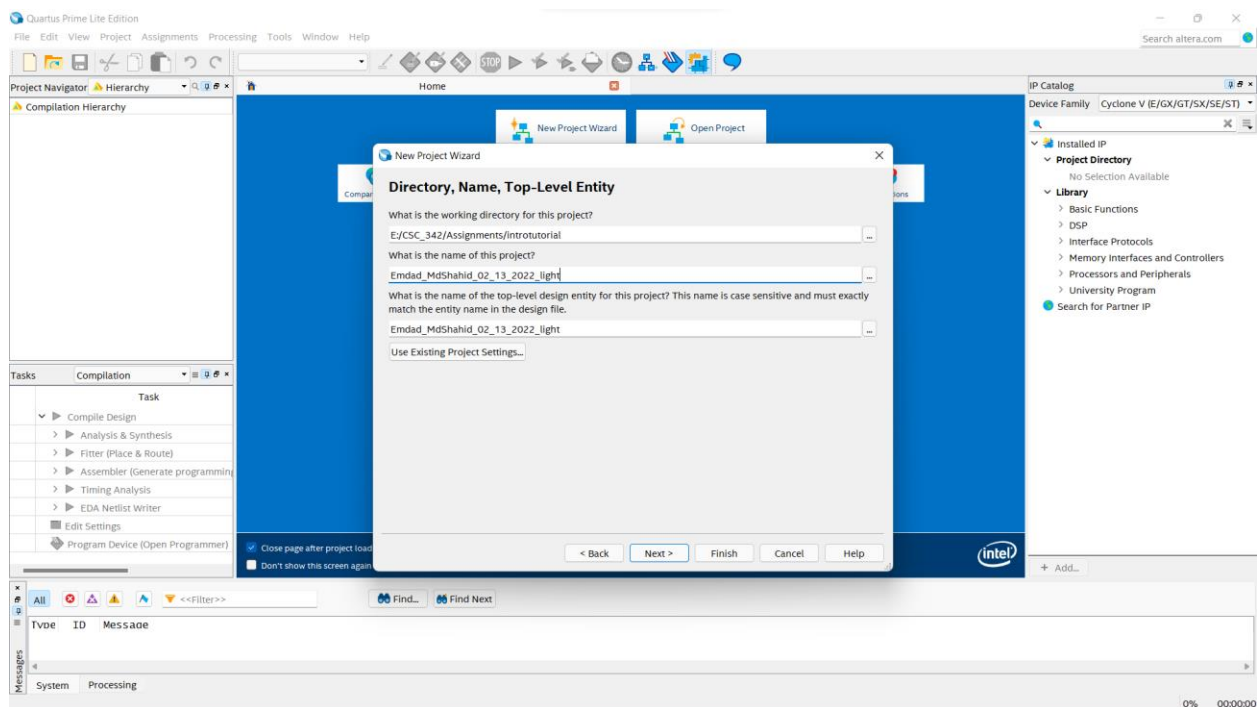


Figure 3: Project folder location

In figure 4, we make the *introtutorial* folder where all the project file will be stored.

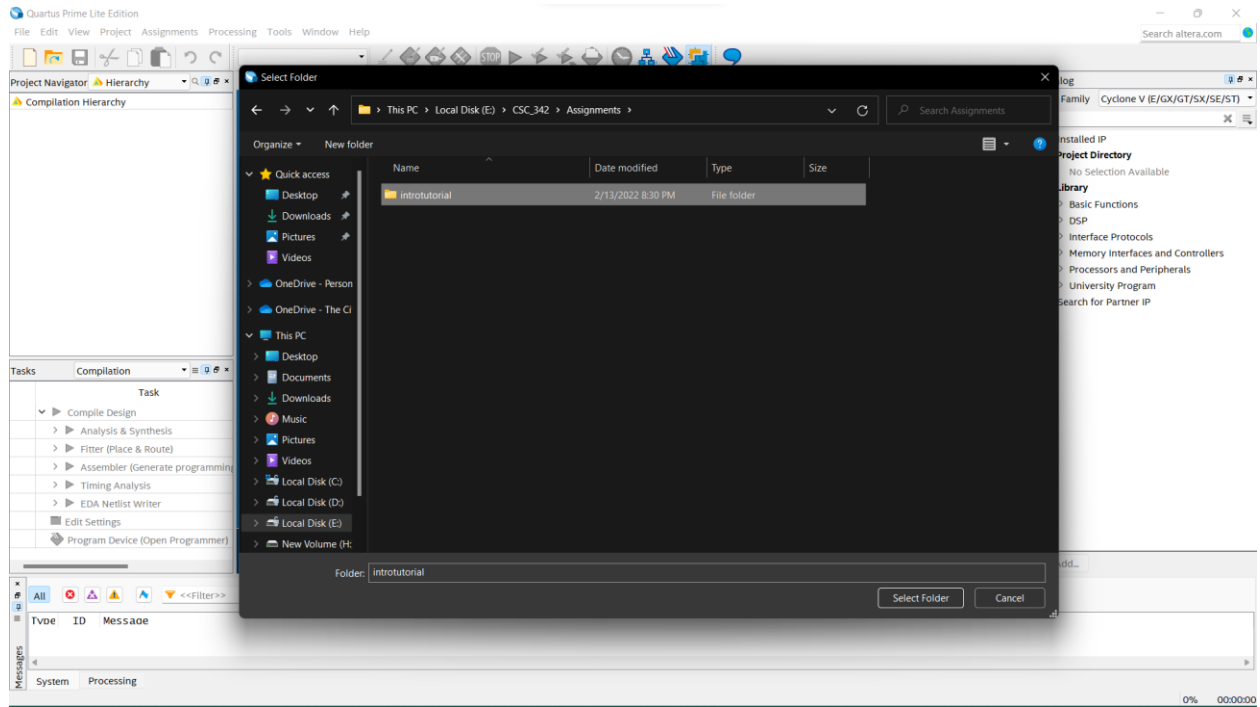


Figure 4: Project directory

In figure 5, I gave the entity name “*Emdad_MdShahid_02_13_2022_light*”.

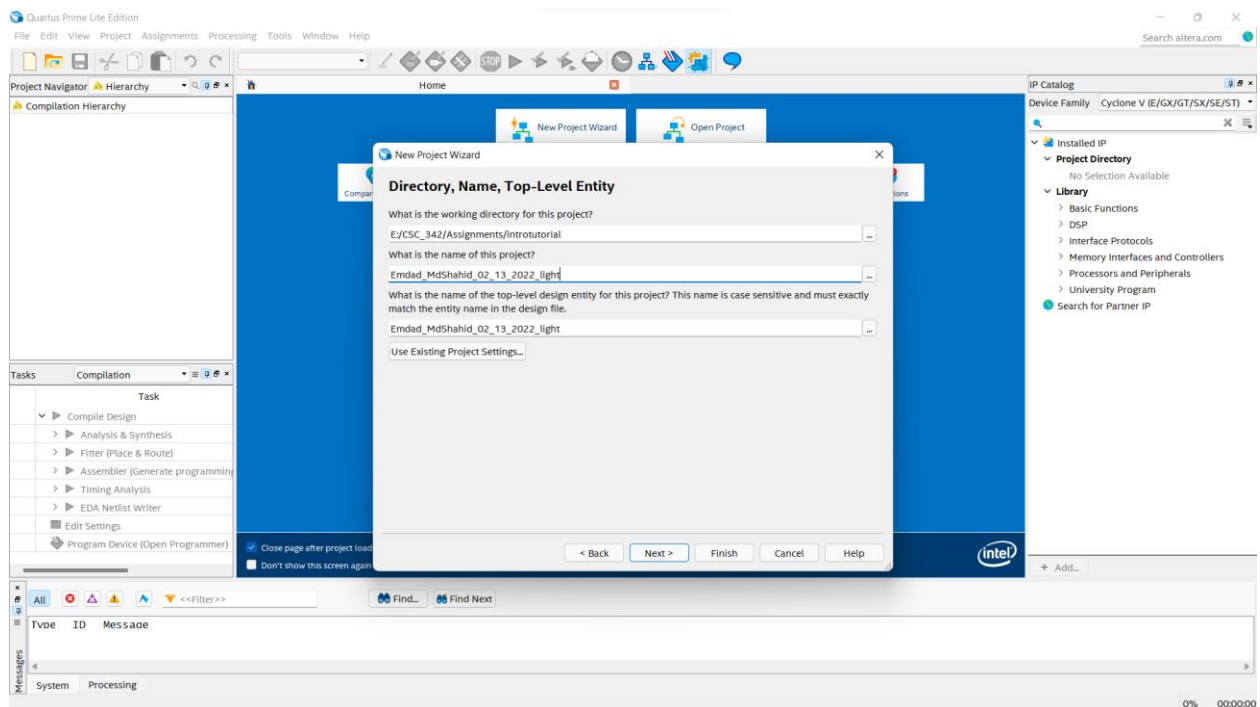


Figure 5: Entity Name

In figure 6, I was instructed to choose a project template but I did not have any project, so I selected empty project.

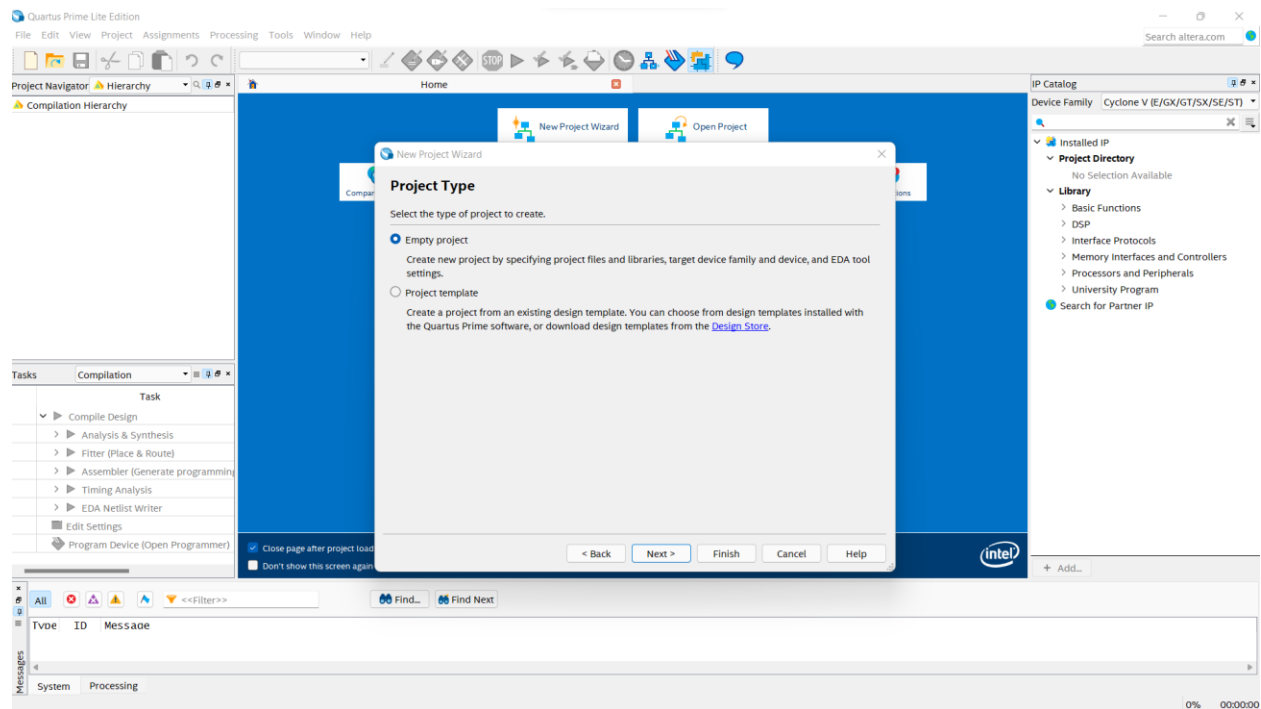


Figure 6: Project type

In figure 7, I was instructed to add any existing VHDL file. I did not have any so clicked next.

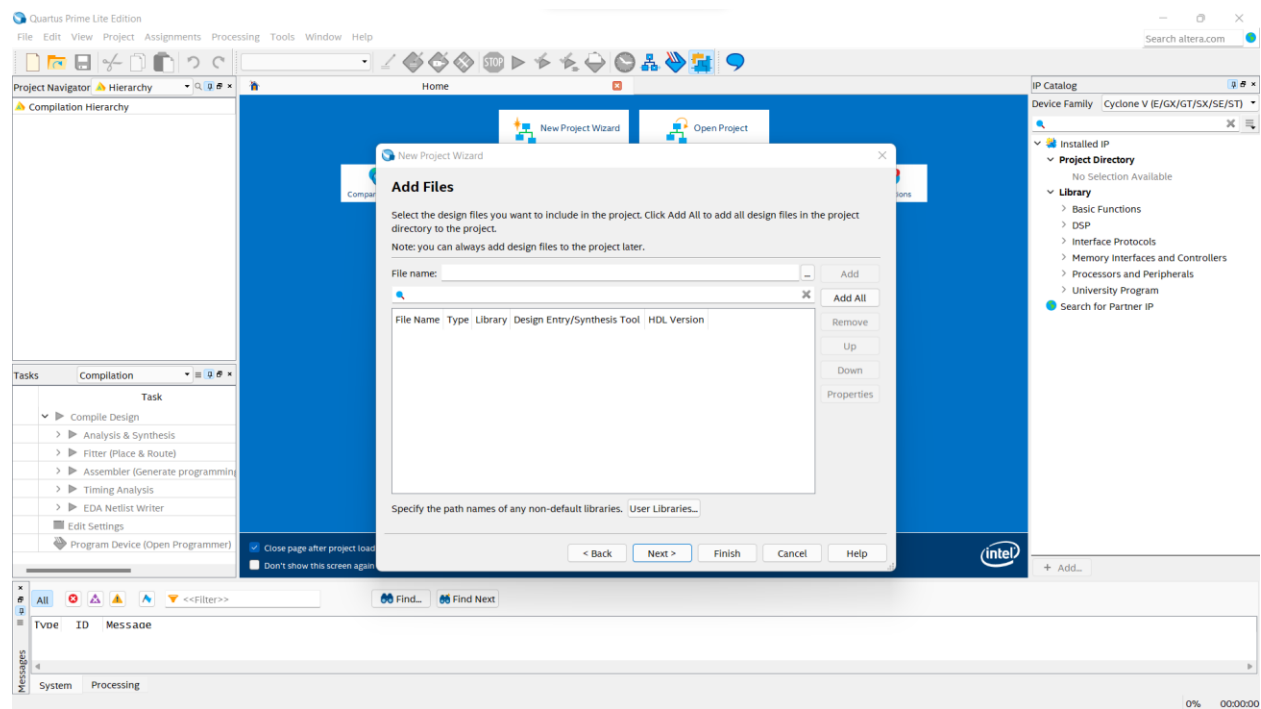


Figure 7: existing VHDL file adding

In figure 8, I was instructed to choose my device family which was already downloaded before installing.

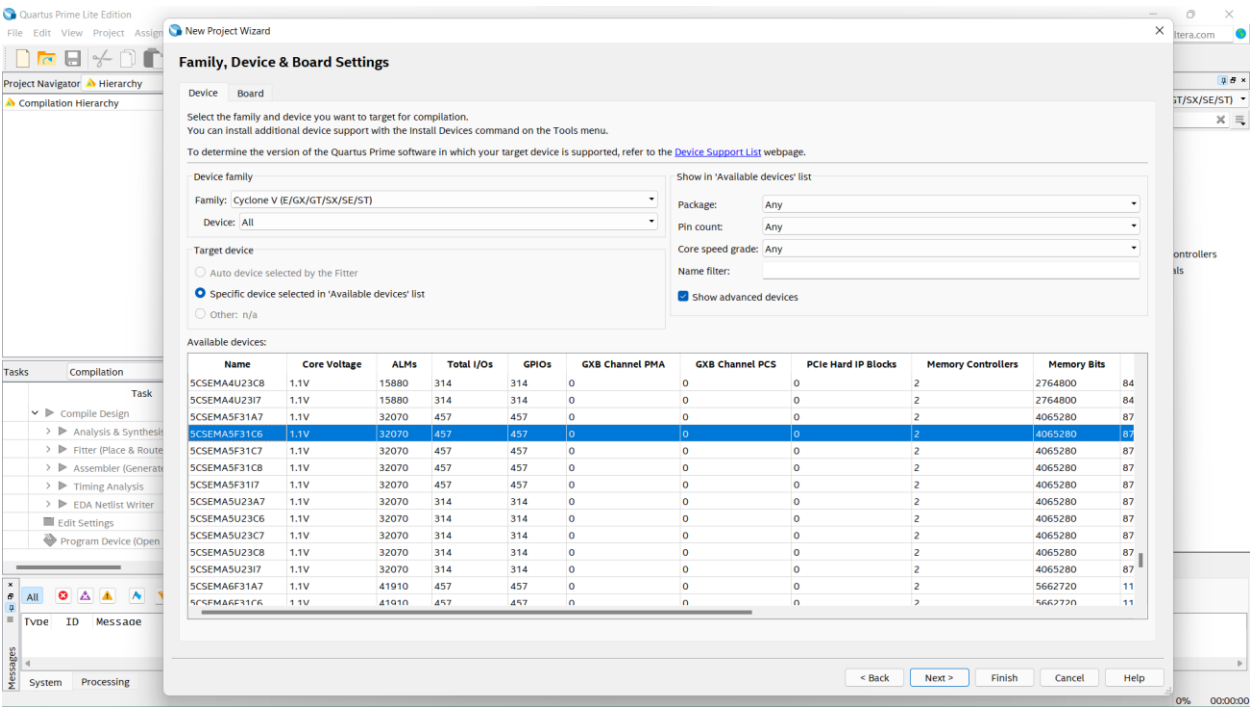


Figure 8: Device and Board Setting

In figure 9, there are EDA tool setting, I did not have any.

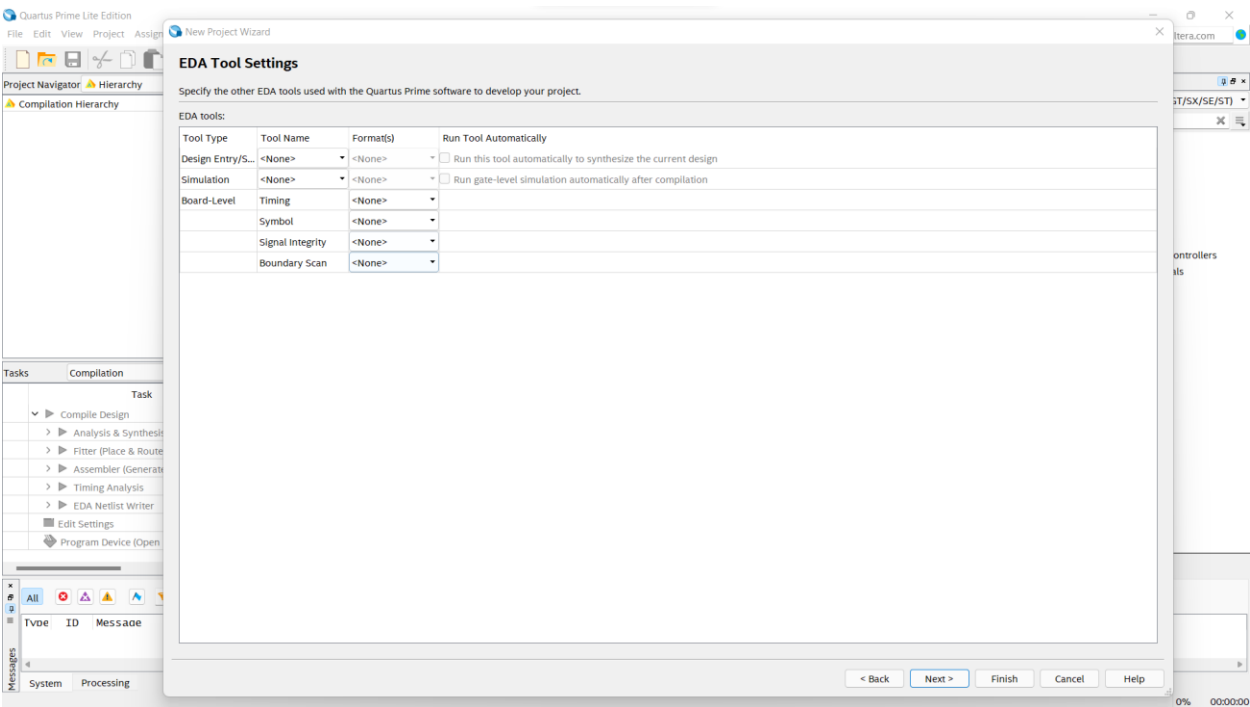


Figure 9: EDA Tool setting

In figure 10, here is my summary specs for my project with all the details.

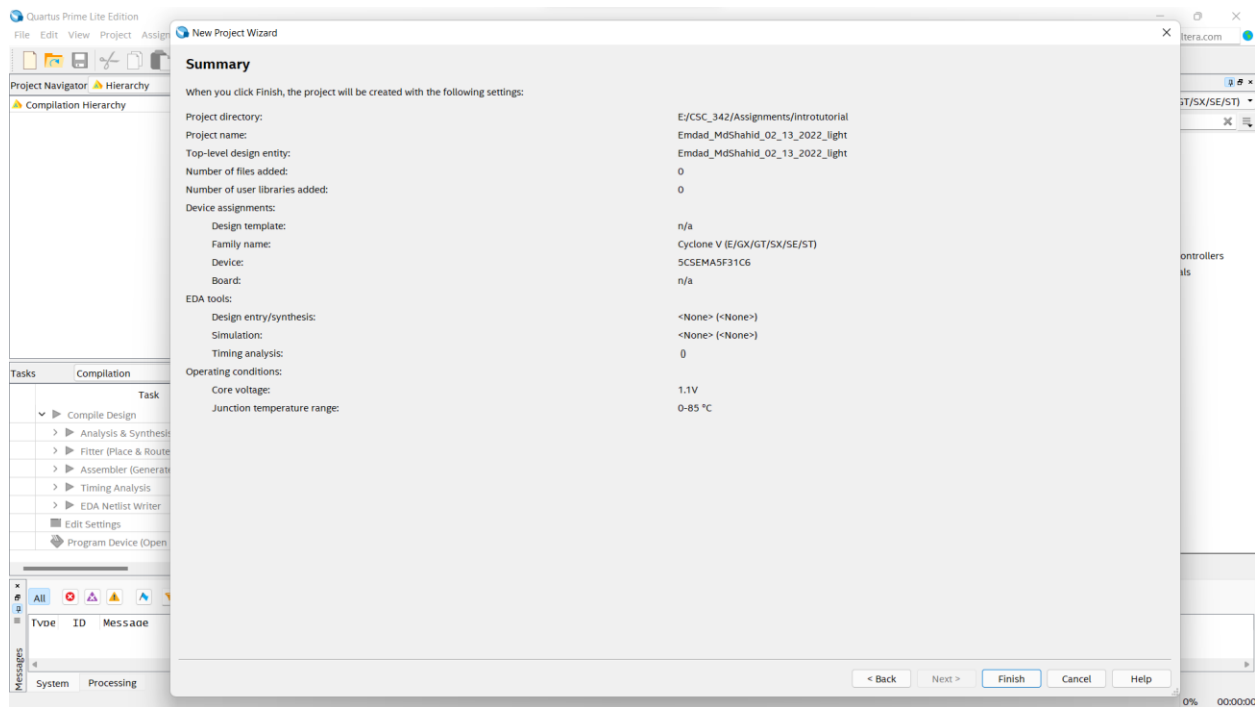


Figure 10: Summary Page

In figure 11, we can see the Quartus home page.

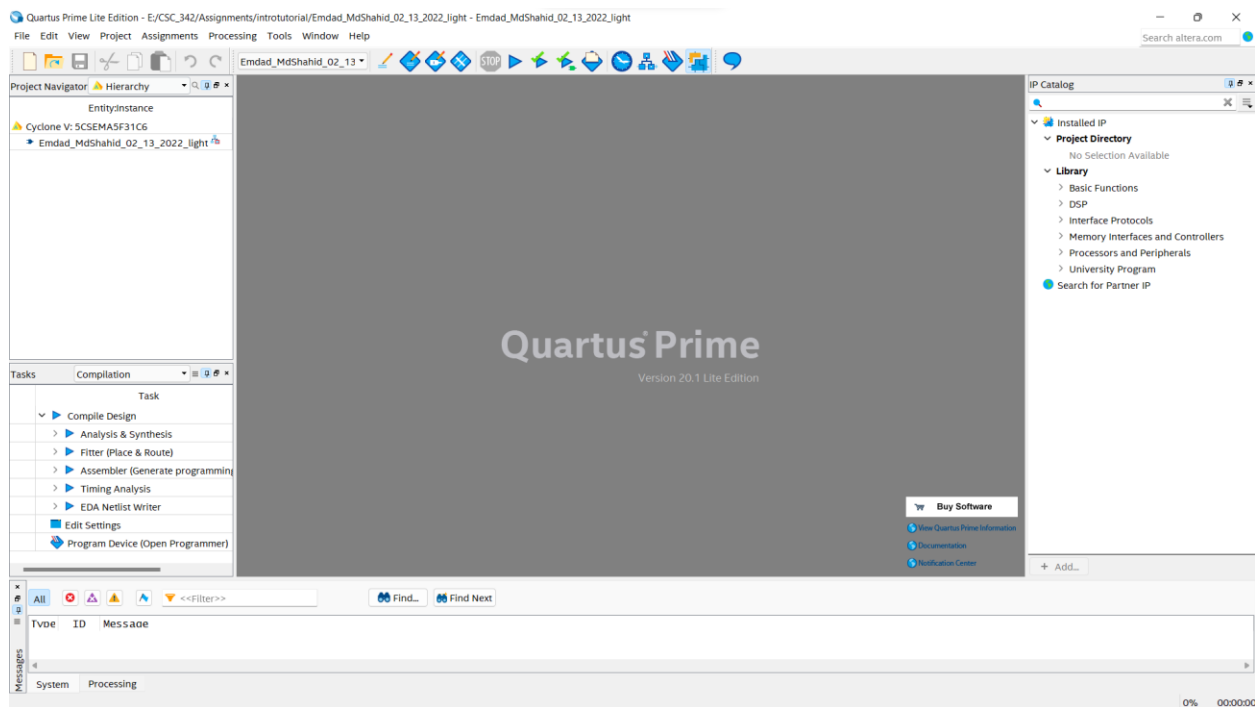


Figure 11: Quartus Home Page

In figure 12, I clicked on **File** again and then new to get the VHDL editor.

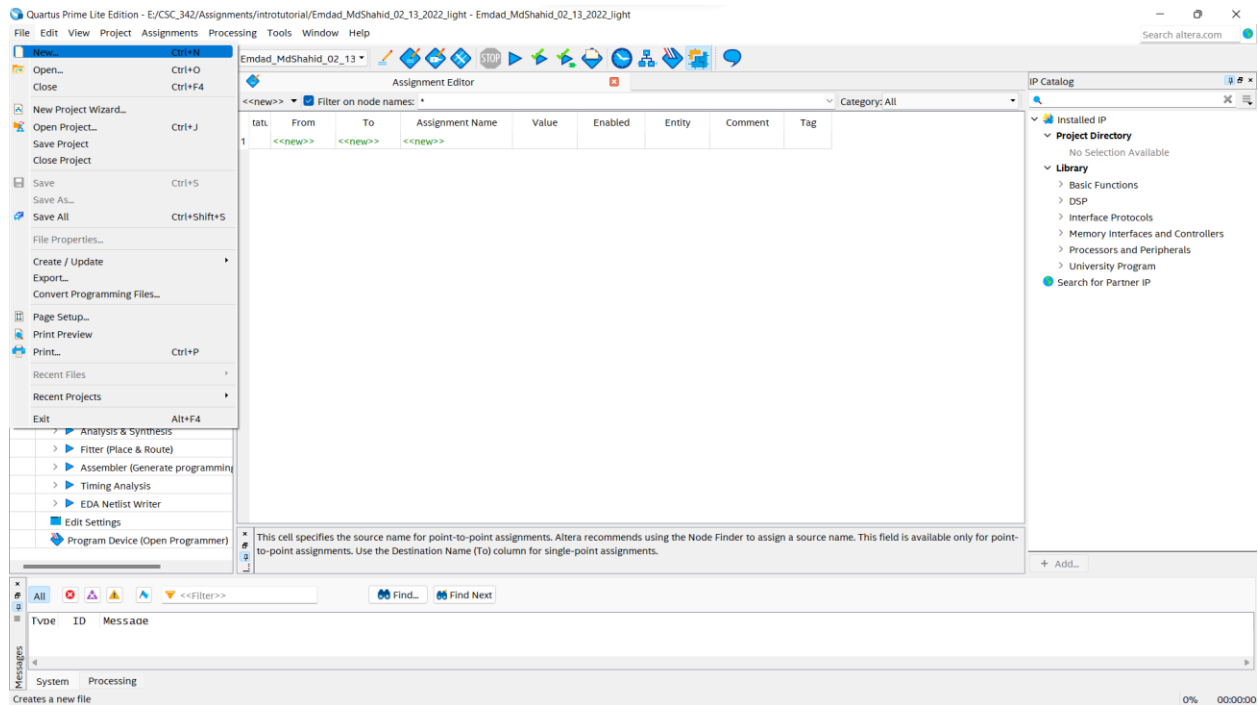


Figure 12: Creating VHDL file

In figure 13, I selected **VHDL file** to create the vhd file.

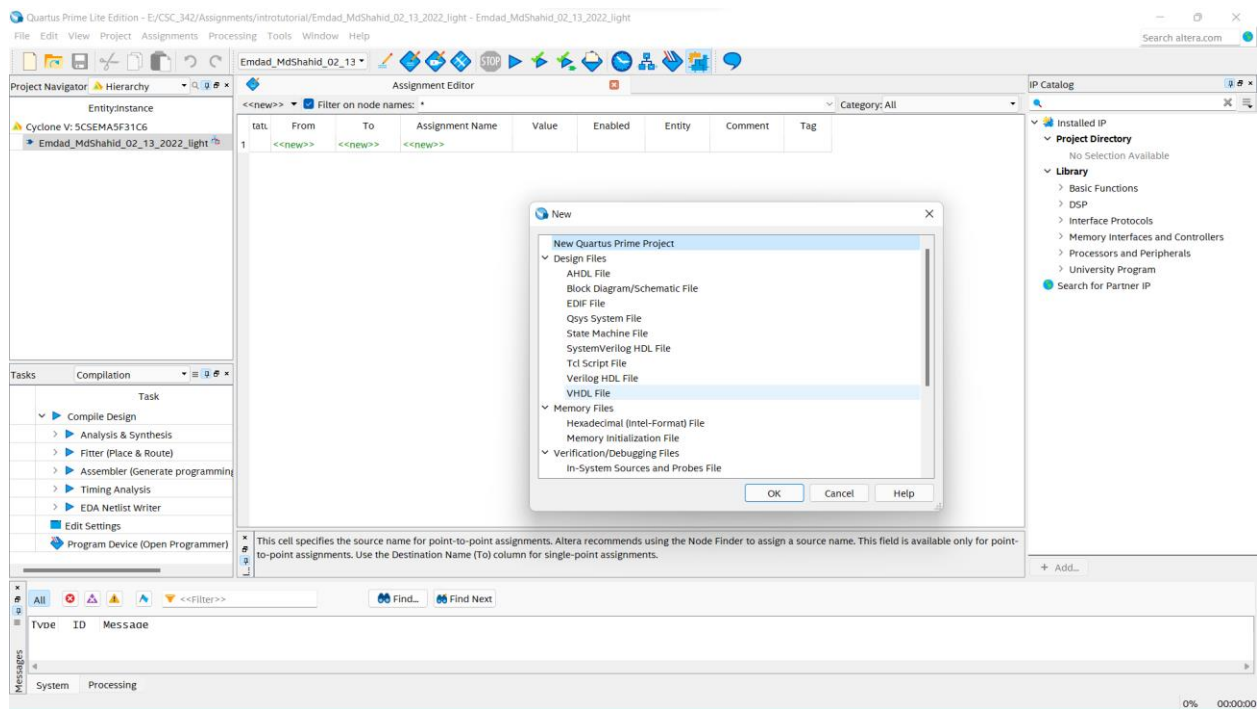


Figure 13: VHDL file selection

In figure 14, I then clicked on *file* again to save the VHDL file by renaming with entity.

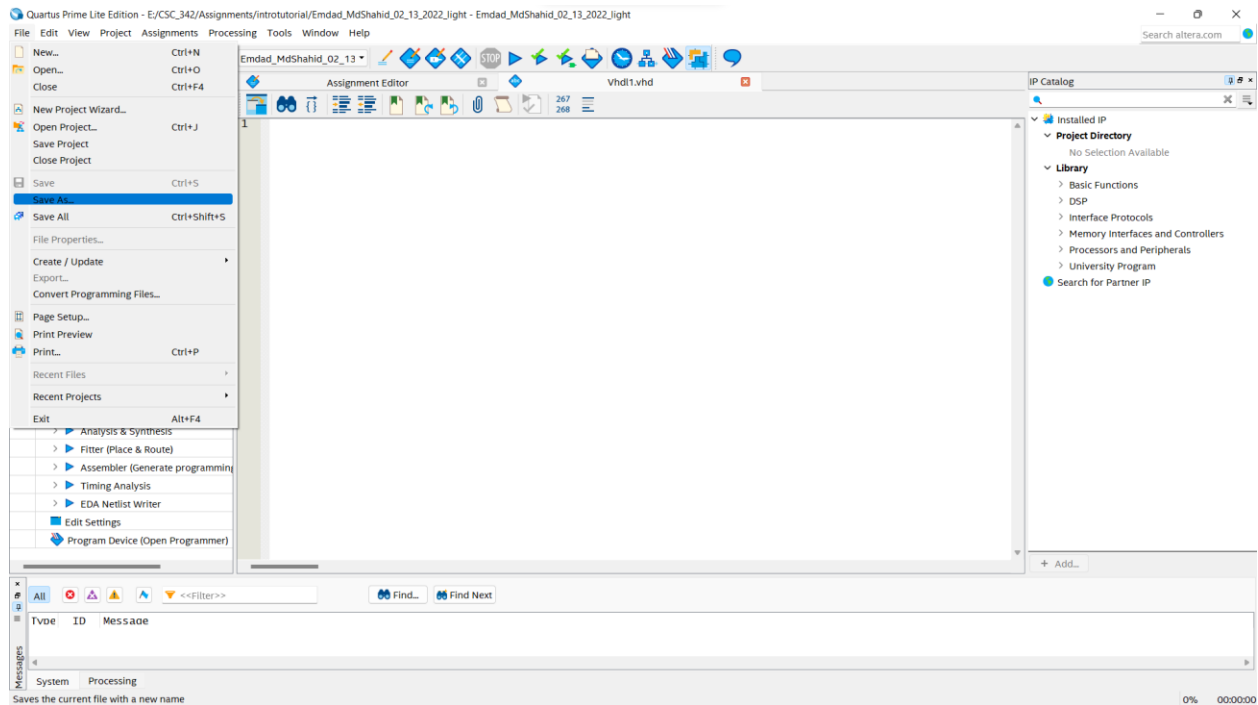


Figure 14: Renaming the VHDL file

In figure 15, I then saved the VHDL file as same as the entity name. I checked on the *add to the project folder*.

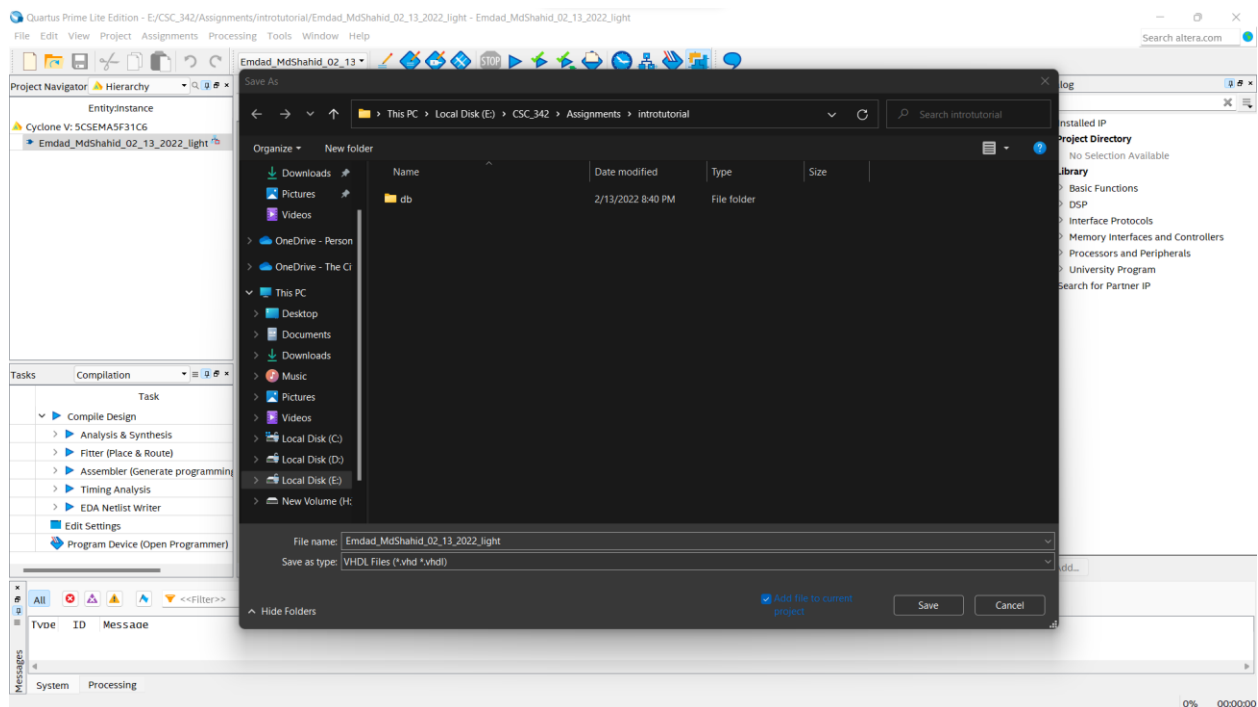


Figure 15: saving the VHDL file

In figure 16, we can see the confirmation message at the bottom about saving the file.

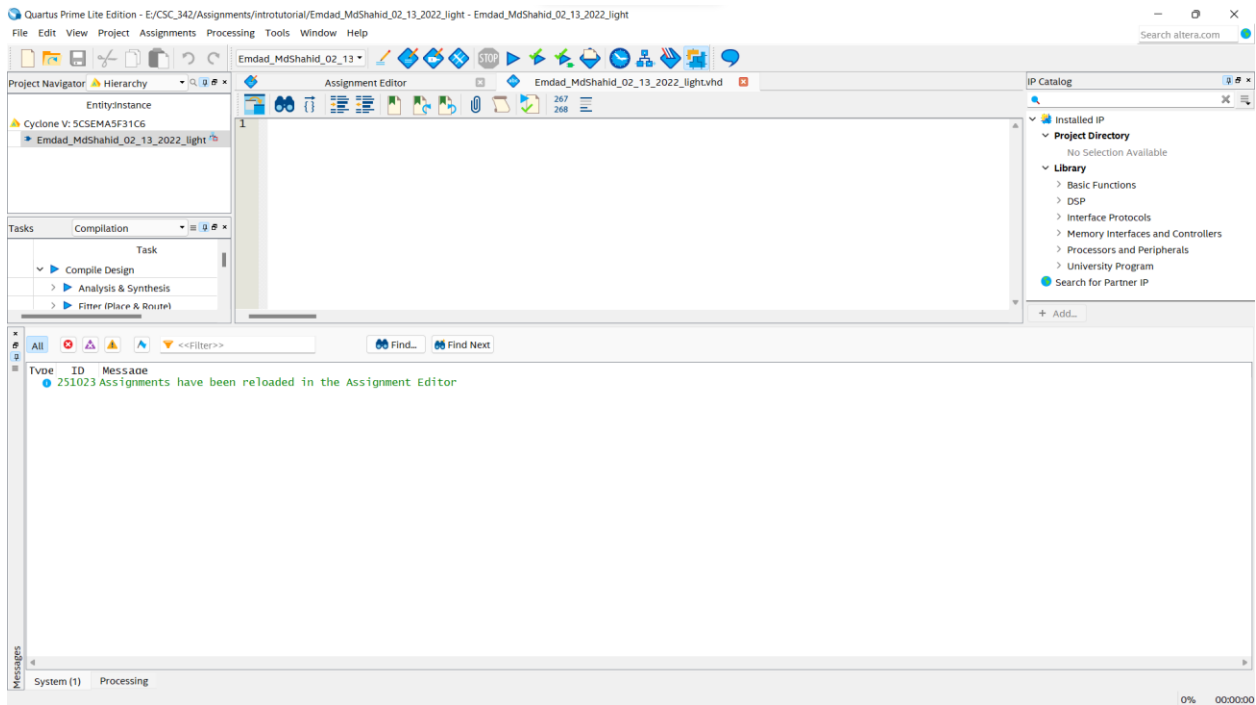


Figure 16: Confirmation about saving the VHDL file

In figure 17, I put my code in the VHDL editor to compile.

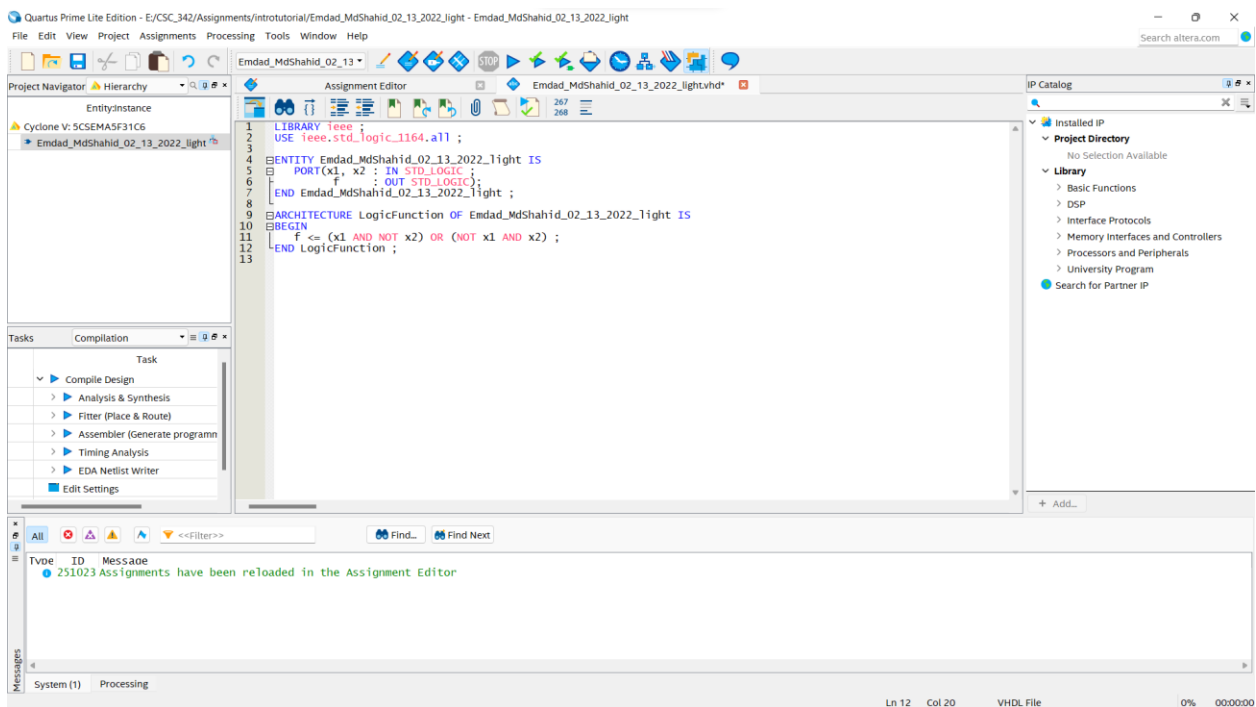


Figure 17: Inserted Code in the editor

In figure 18, I had to save the file to compile.

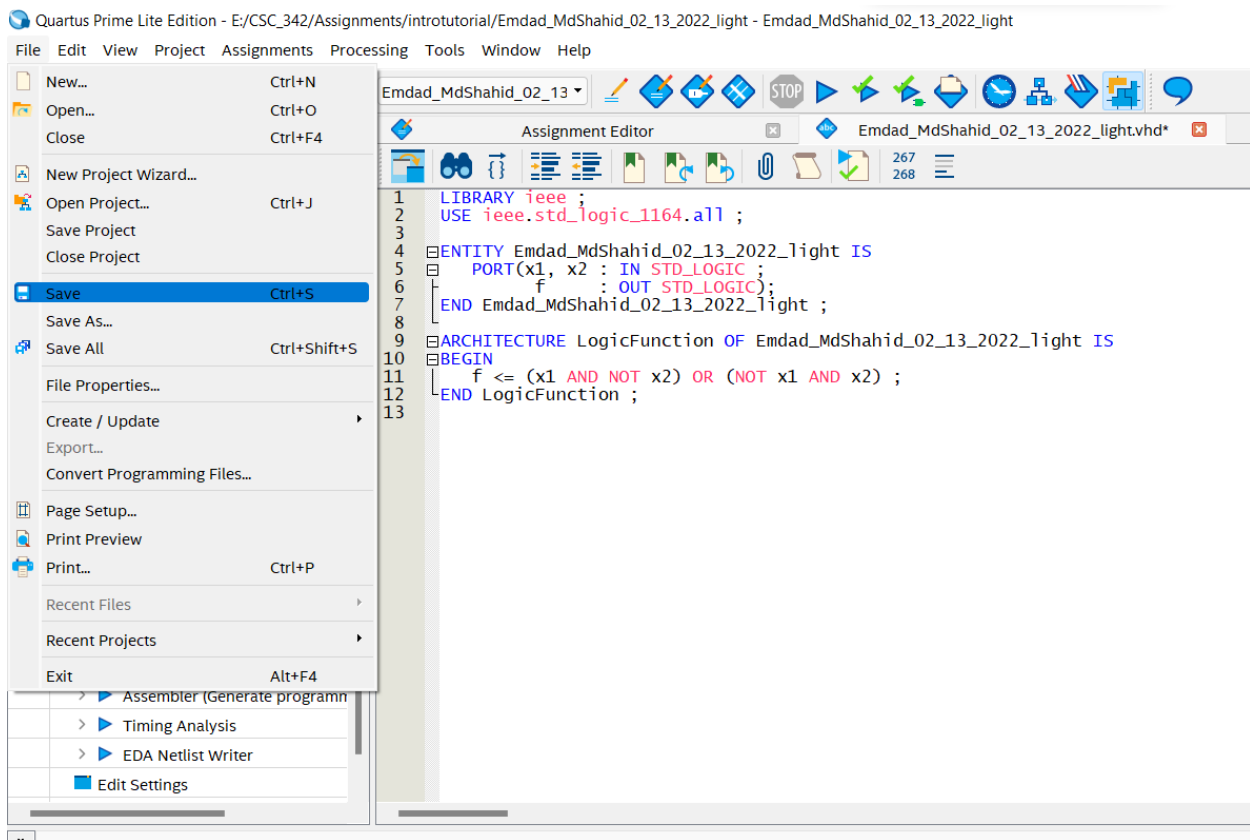


Figure 18: Saving the modified file

In figure 19, I clicked on **Assignments** to click on **Setting**.

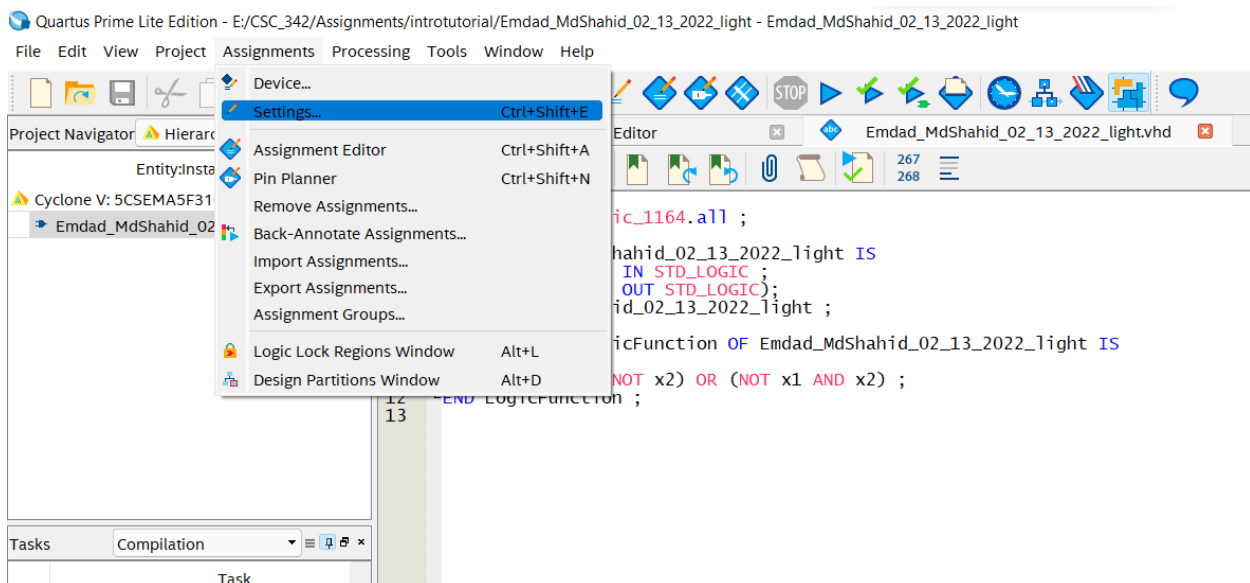


Figure 19: Assignment Setting

In figure 20, I was checking if my VHDL was there so I can compile.

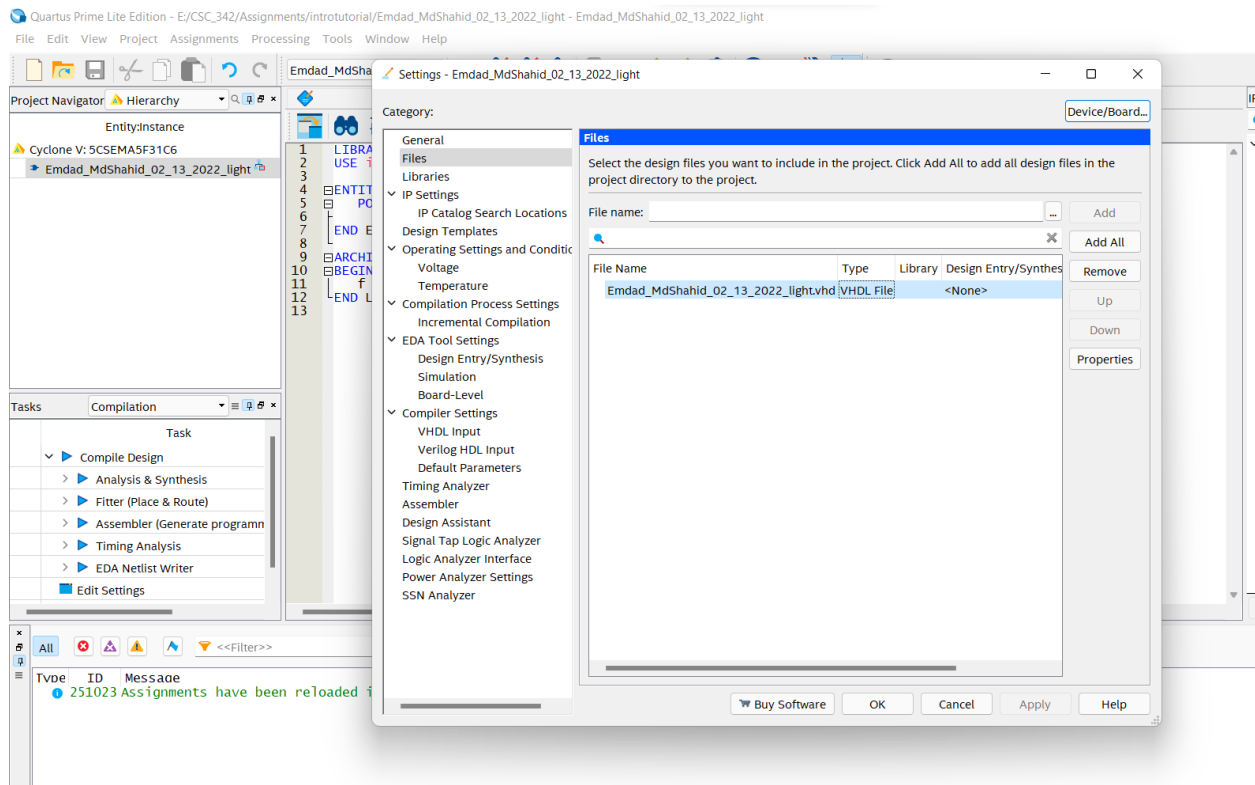


Figure 20: Checking the VHDL file location

In figure 21, I clicked **Processing** and then **Start Compilation** to compile the program.

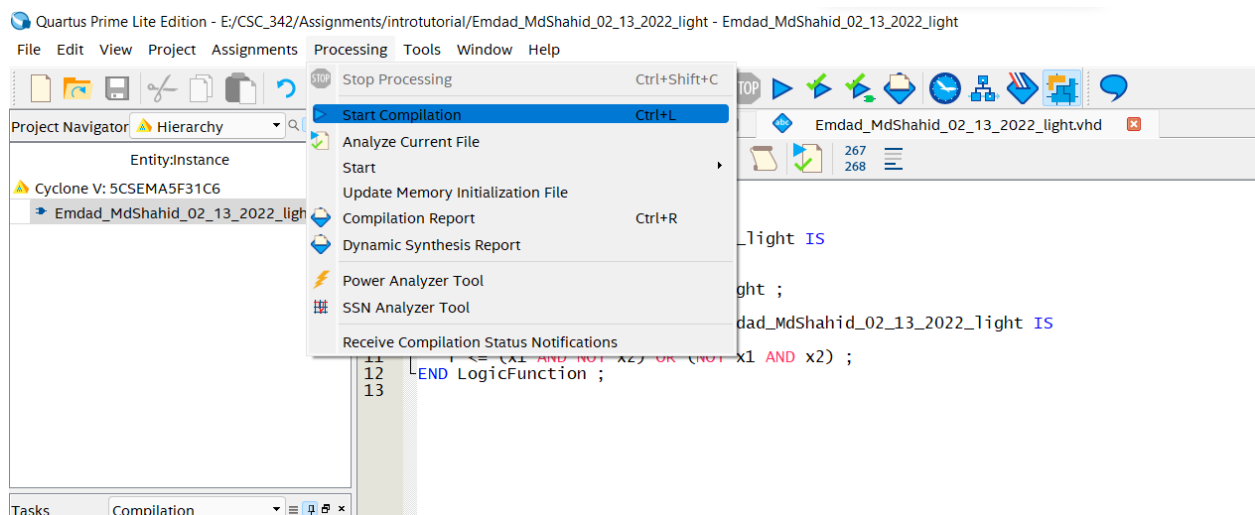


Figure 21: Starting to compile

In figure 22, I can see the compilation report page. How long did it take and all test pass.

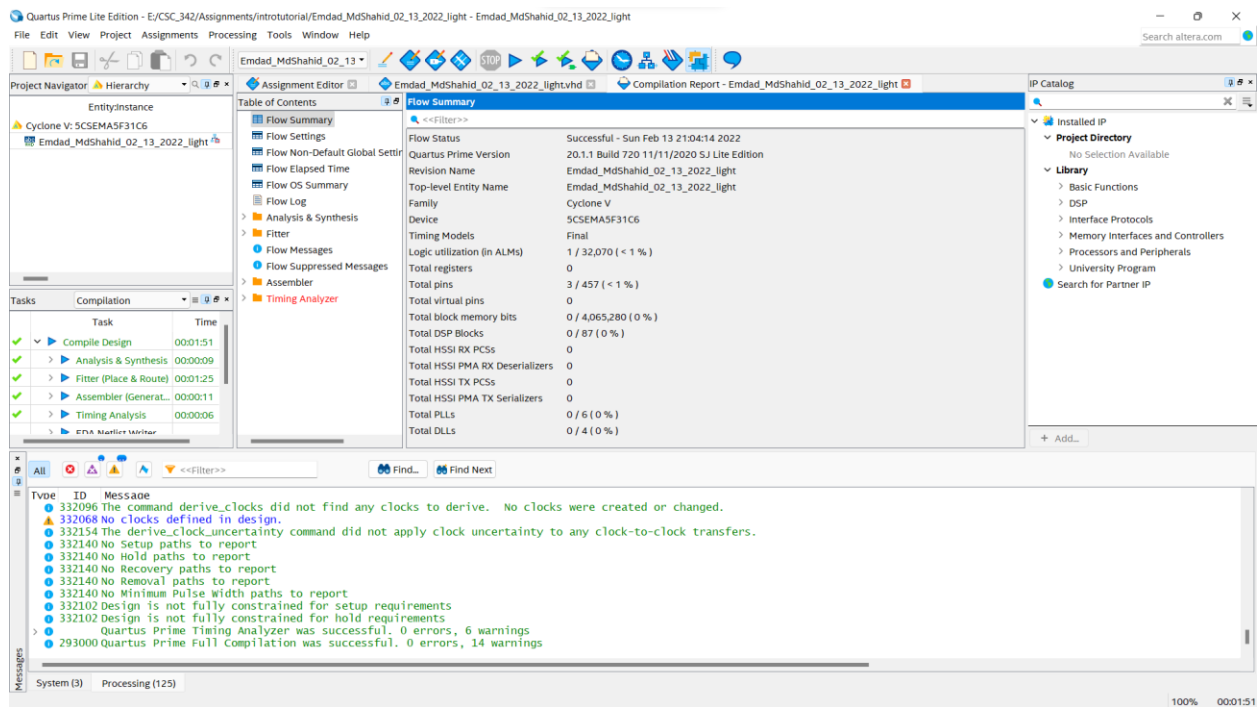


Figure 22: Compilation report

II. 2:1 mux 1-bit

To do the 2nd part of the project, I clicked on **file** menu again to create **New Project Wizard**.

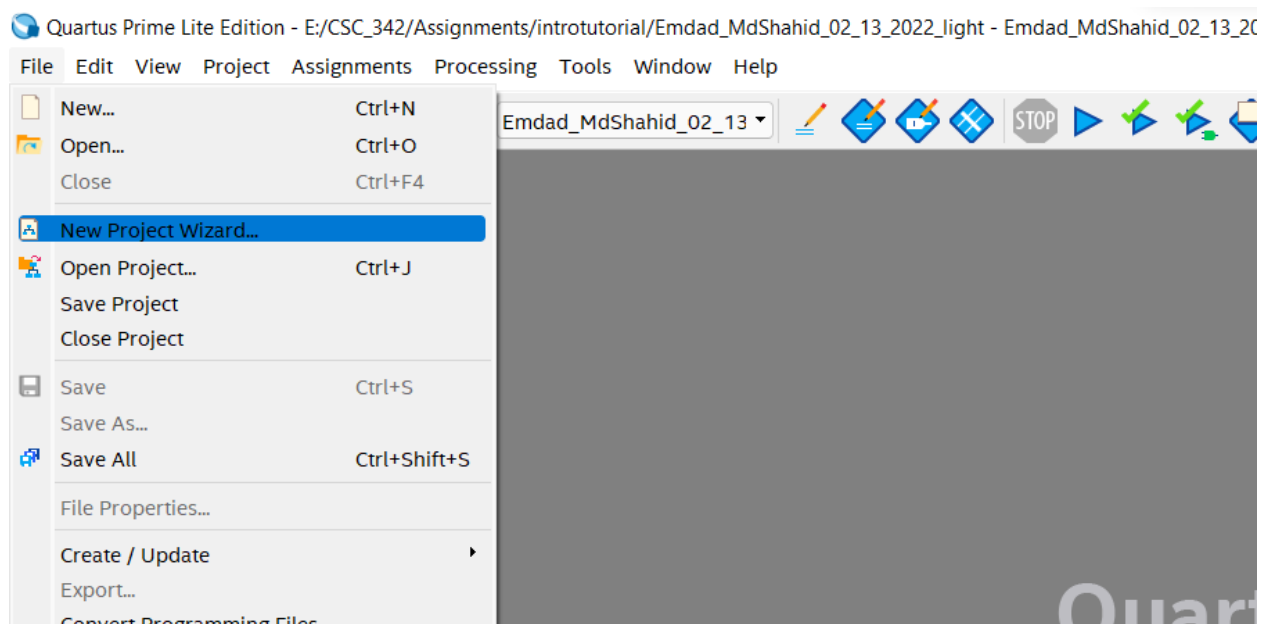


Figure 23: New Project Creation

In figure 24, I then created a separate folder for the project and select the directory.

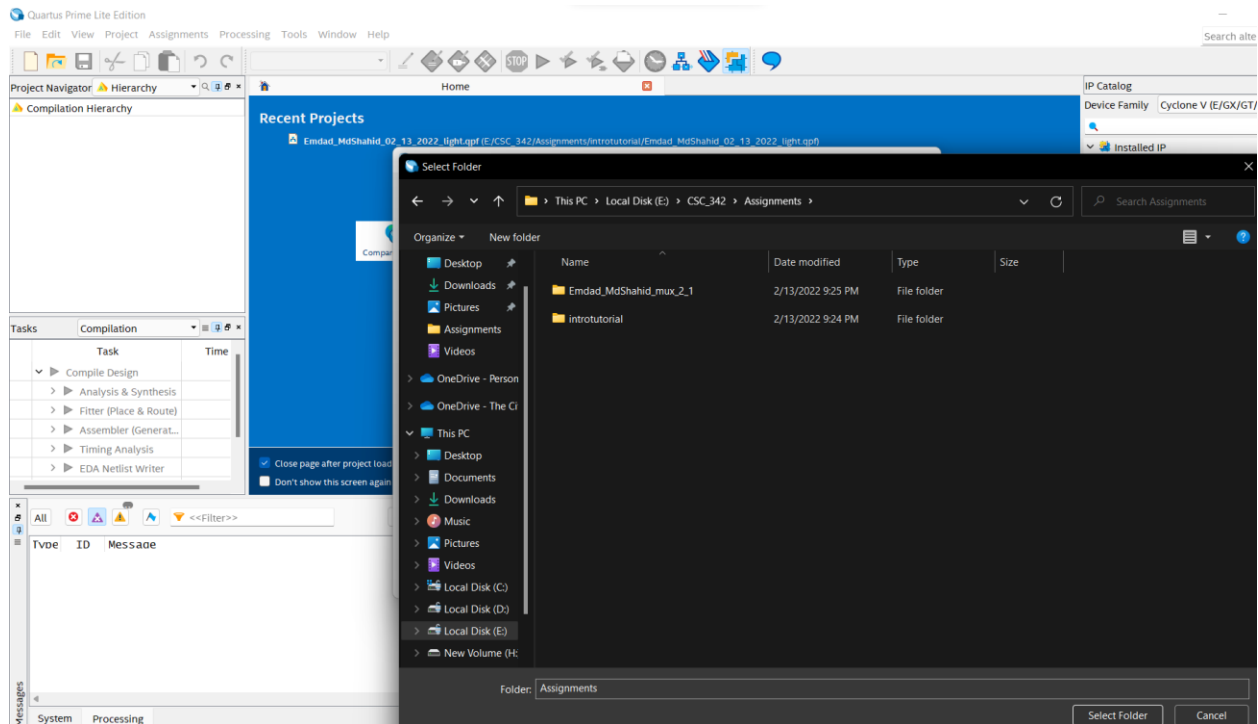


Figure 24: Directory

In figure 25, I chose the directory and gave the entity name *Emdad_MdShahid_mux_2_1_32bit*.

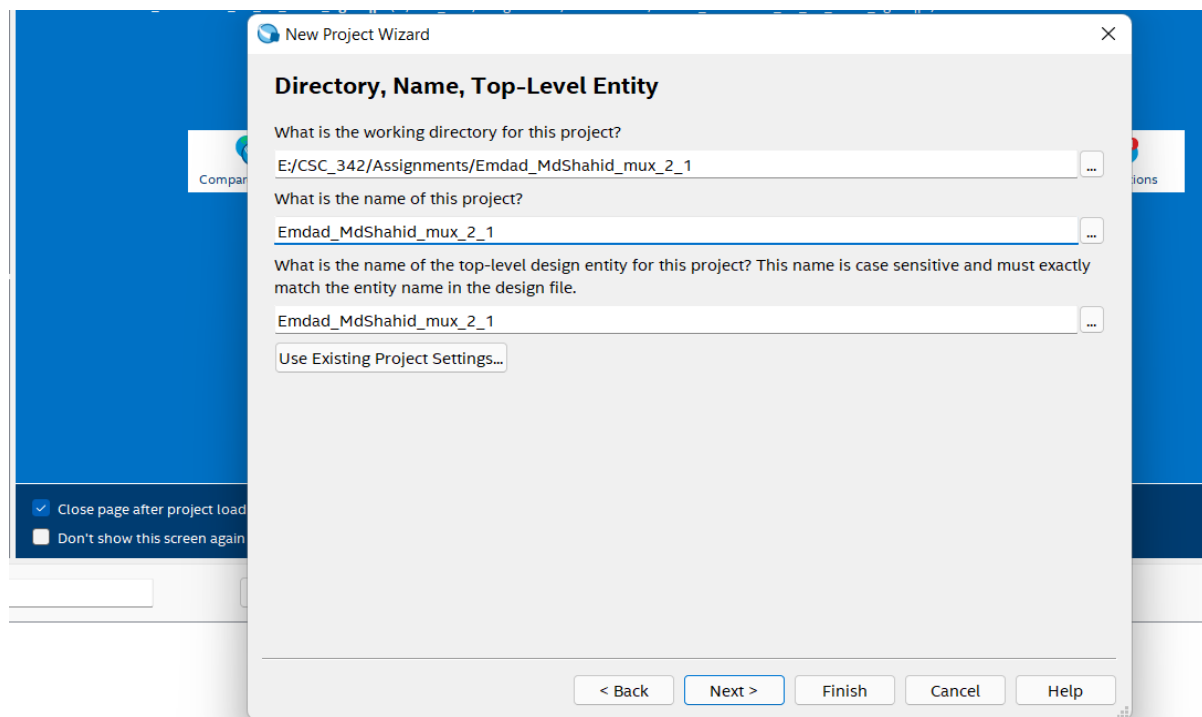


Figure 25: Directory and Entity Name

In figure 26, I can show the summary of project.

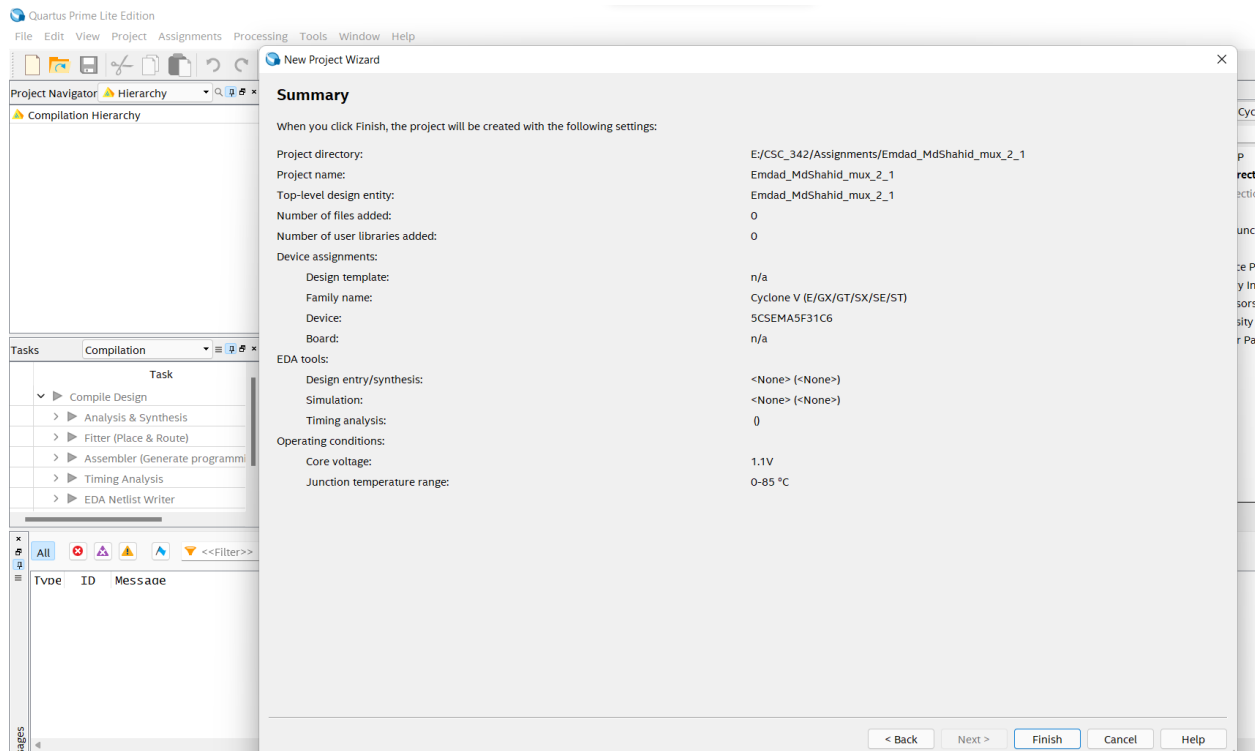


Figure 26: Summary Page

In figure 27, I clicked on the **file** menu to create the VHDL file.

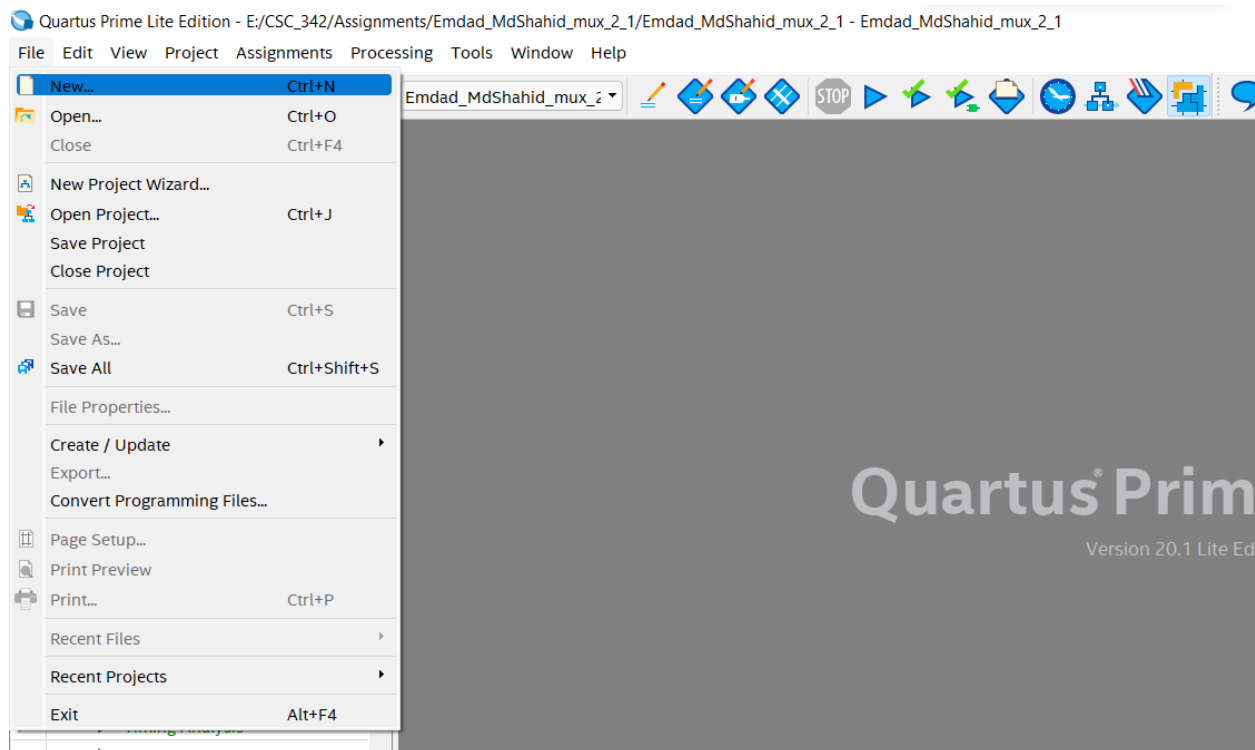


Figure 27: new file

In figure 28, I saved the file in the same directory.

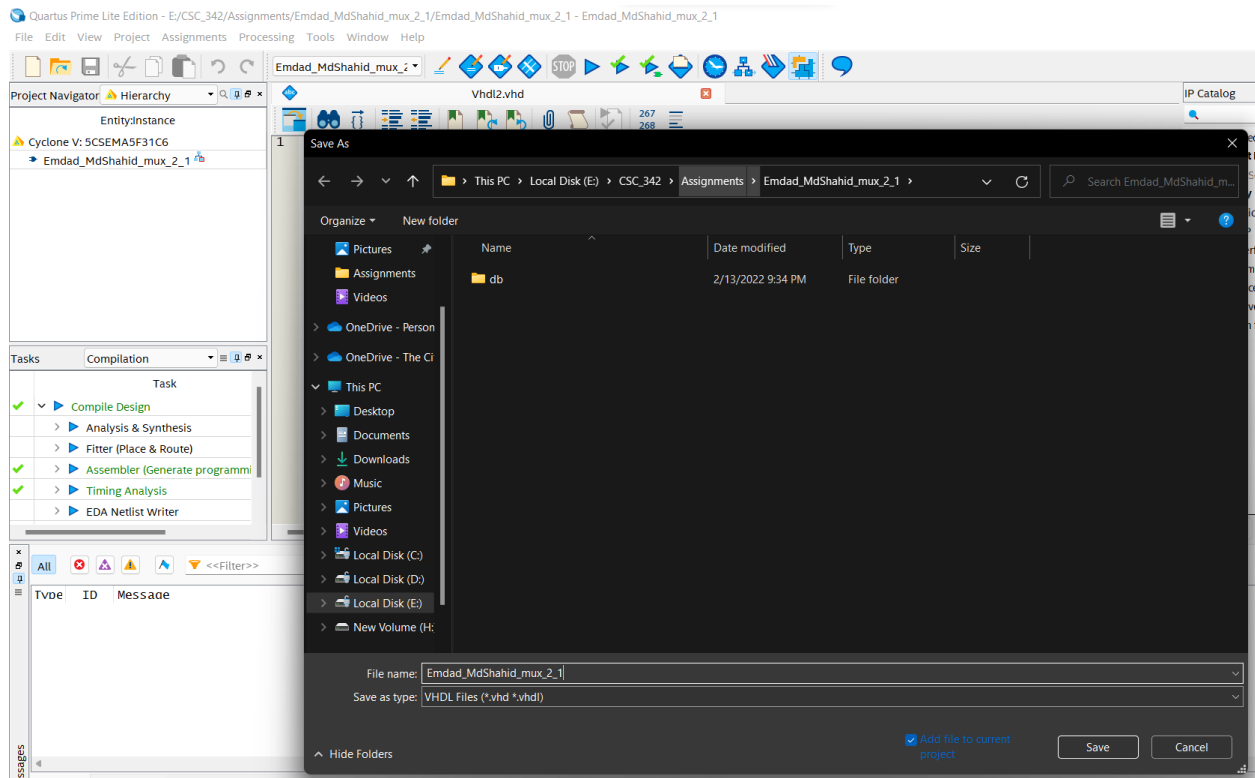


Figure 28: saving file

In figure 29, I wrote my code for the Part B.

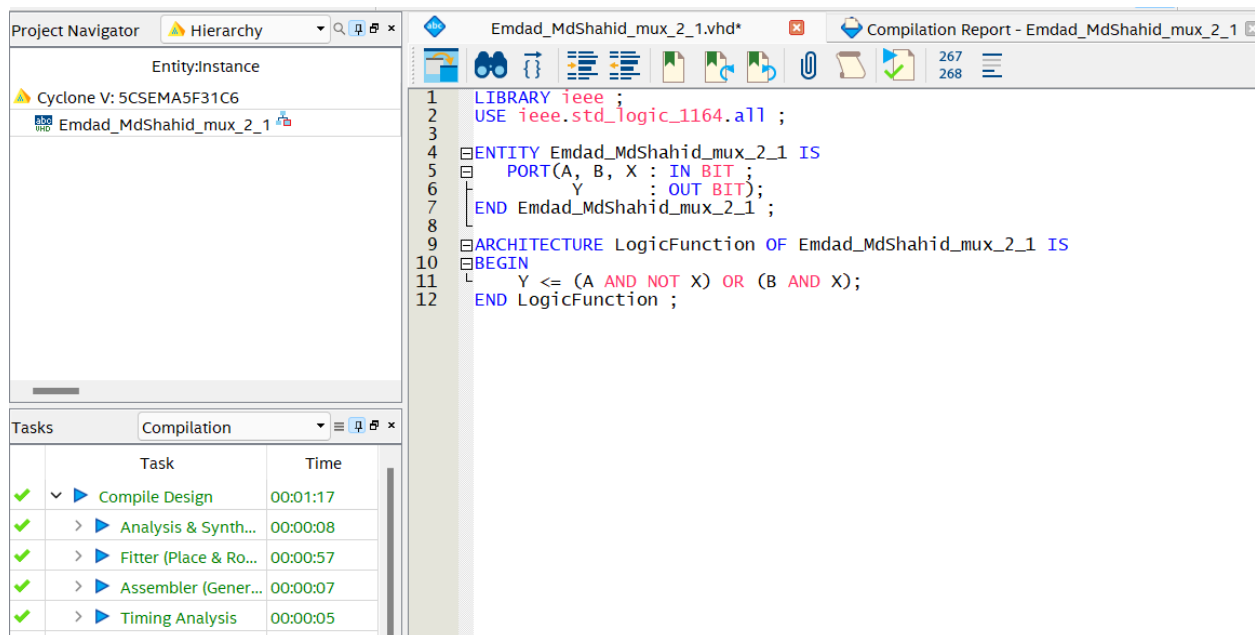


Figure 29: Part B code

In figure 30, we can see the Compilation Report of the Part B project.

Quartus Prime Lite Edition - E:\CSC_342\Assignments\Emdad_MdShahid_mux_2_1\Emdad_MdShahid_mux_2_1 - Emdad_MdShahid_mux_2_1

File Edit View Project Assignments Processing Tools Window Help

Emdad_MdShahid_mux_2_1.vhd

Compilation Report - Emdad_MdShahid_mux_2_1

Project Navigator

Entity:Instance

Cyclone V: 5CSEMA5F31C6

Emdad_MdShahid_mux_2_1

Table of Contents

- Flow Summary
- Flow Settings
- Flow Non-Default Global Settings
- Flow Elapsed Time
- Flow OS Summary
- Flow Log
- Analysis & Synthesis
- Fitter
- Flow Messages
- Flow Suppressed Messages
- Assembler
- Timing Analyzer

Flow Summary

Flow Status: Successful - Sun Feb 13 21:42:41 2022

Quartus Prime Version: 20.1.1 Build 720 11/11/2020 SJ Lite Edition

Revision Name: Emdad_MdShahid_mux_2_1

Top-level Entity Name: Emdad_MdShahid_mux_2_1

Family: Cyclone V

Device: 5CSEMA5F31C6

Timing Models: Final

Logic utilization (in ALMs): 1 / 32,070 (< 1 %)

Total registers: 0

Total pins: 4 / 457 (< 1 %)

Total virtual pins: 0

Total block memory bits: 0 / 4,065,280 (0 %)

Total DSP Blocks: 0 / 87 (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 %)

Tasks

Compilation

Task	Time
Compile Design	00:01:17
Analysis & Synth...	00:00:08
Fitter (Place & Ro...	00:00:57
Assembler (Gener...	00:00:07
Timing Analysis	00:00:05
EDA Netlist Writer	
Edit Settings	

Find... Find Next

Messages

Type	ID	Message
Warning	332154	The derive_clock_uncertainty command did not apply clock uncertainty to any clock-to-clock transfers.
Warning	332140	No Setup paths to report
Warning	332140	No Hold paths to report
Warning	332140	No Recovery paths to report
Warning	332140	No Removal paths to report
Warning	332140	No Minimum Pulse Width paths to report
Warning	332102	Design is not fully constrained for setup requirements
Warning	332102	Design is not fully constrained for hold requirements
Information		Quartus Prime Timing Analyzer was successful. 0 errors, 6 warnings
Information	293000	Quartus Prime Full Compilation was successful. 0 errors, 14 warnings

Figure 30: Part B code Compilation Report

II. 2:1 mux 32-bit

To do the 3rd part of the project, I clicked on *file* menu again to create *New Project Wizard*. In figure 31, I gave the directory and Entity Name.

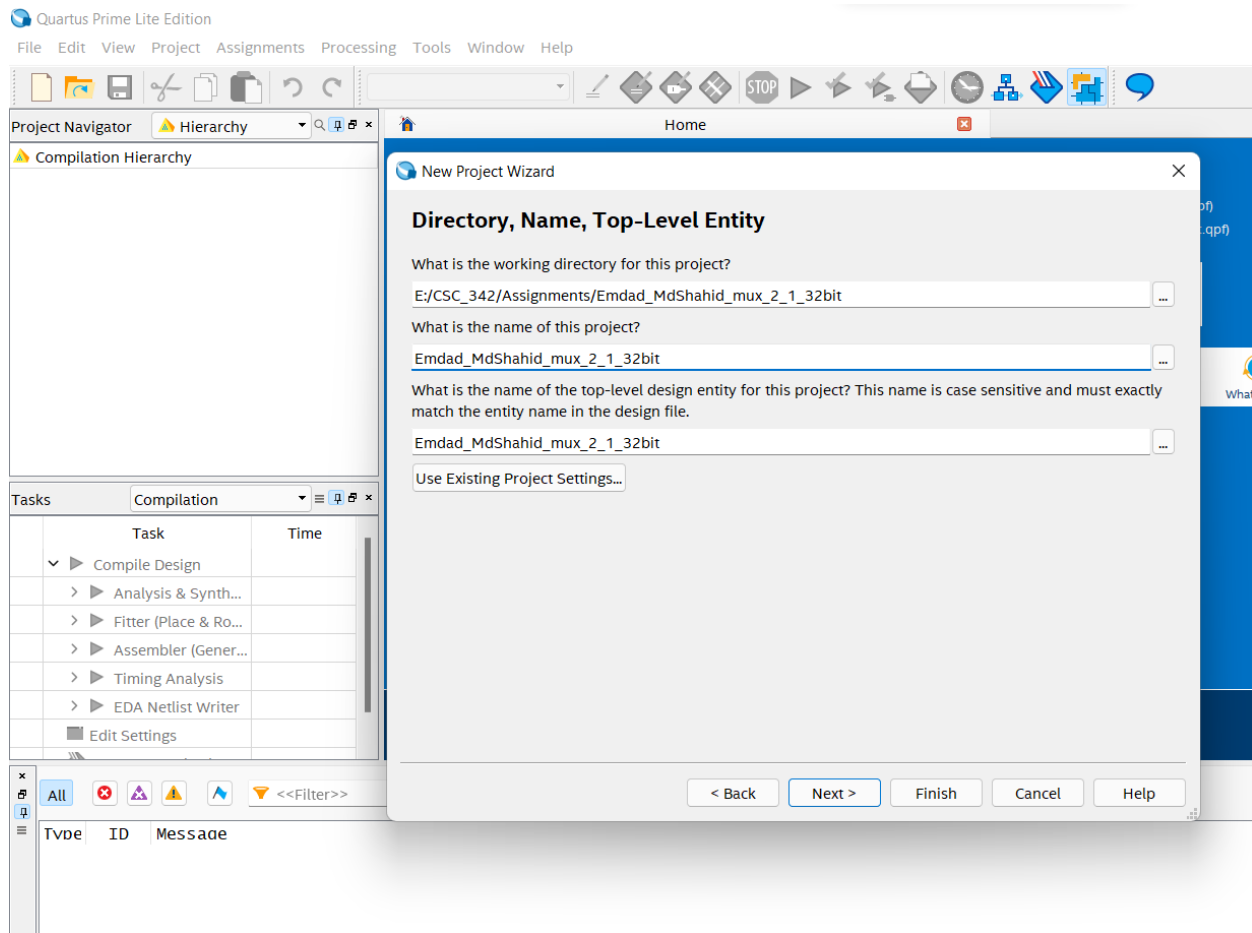


Figure 31: Directory and Entity Name

In figure 32, I wrote my code to compile for Part C.

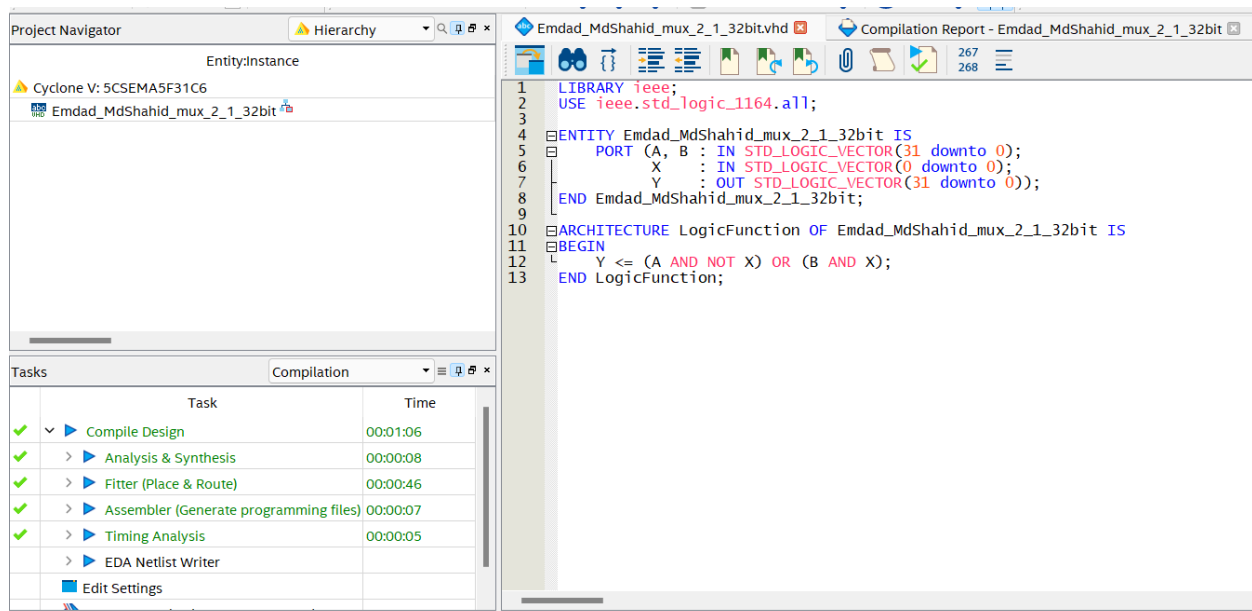


Figure 32: Part C code

In figure 33, We can see the compilation report for the Part C code.

The screenshot displays the Quartus Prime Lite Edition interface. The top menu bar includes File, Edit, View, Project, Assignments, Processing, Tools, Window, and Help. The Project Navigator on the left shows the hierarchy for 'Cyclone V: 5CSEMA5F31C6' and the selected file 'Emdad_MdShahid_mux_2_1_32bit'. The main window is divided into several panes:

- Table of Contents:** Lists various reports including Flow Summary, Flow Settings, Flow Non-Default Global Settings, Flow Elapsed Time, Flow OS Summary, Flow Log, Analysis & Synthesis, Fitter, Flow Messages, Flow Suppressed Messages, Assembler, and Timing Analyzer.
- Flow Summary:** Provides a detailed overview of the compilation process.

Flow Status	Successful - Sun Feb 13 22:38:04 2022
Quartus Prime Version	20.1.1 Build 720 11/11/2020 SJ Lite Edition
Revision Name	Emdad_MdShahid_mux_2_1_32bit
Top-level Entity Name	Emdad_MdShahid_mux_2_1_32bit
Family	Cyclone V
Device	5CSEMA5F31C6
Timing Models	Final
Logic utilization (in ALMs)	1 / 32,070 (< 1 %)
Total registers	0
Total pins	97 / 457 (21 %)
Total virtual pins	0
Total block memory bits	0 / 4,065,280 (0 %)
Total DSP Blocks	0 / 87 (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 %)
- Tasks:** A table showing the progress of various tasks.

Task	Time
Compile Design	00:01:06
Analysis & Synthesis	00:00:08
Fitter (Place & Route)	00:00:46
Assembler (Generate programming files)	00:00:07
Timing Analysis	00:00:05
- Messages:** A list of messages generated during compilation.
 - 332096 The command derive_clocks did not find any clocks to derive. No clocks were created or changed.
 - 332068 No clocks defined in design.
 - 332154 The derive_clock_uncertainty command did not apply clock uncertainty to any clock-to-clock transfers.
 - 332140 No Setup paths to report
 - 332140 No Hold paths to report
 - 332140 No Recovery paths to report
 - 332140 No Removal paths to report
 - 332140 No Minimum Pulse Width paths to report
 - 332102 Design is not fully constrained for setup requirements
 - 332102 Design is not fully constrained for hold requirements
 - Quartus Prime Timing Analyzer was successful. 0 errors, 6 warnings
 - 293000 Quartus Prime Full Compilation was successful. 0 errors, 109 warnings

Figure 33: Part C code Compilation Report

Explanation:

For part A, I totally followed the instructor tutorial. For Part B, I did use the tutorial code and modified the port In and Out as the requirement was to have 2:1 multiplexer but 1 bit. With some internet research, it was easy to and the equation did the job. For Part C, I also did use the tutorial code and modified the port In and Out as the requirement was to have 2:1 multiplexer where each signal is 32 bits and the selector signal is one bit. It was not easy to do as the Quartus software giving error not taking BIT with STD_LOGIC_VECTOR. I used AND and OR gate to compile the program.

Conclusion:

This assignment was helpful to learn VHDL syntax and code and compile in Quartus software. It was a thorough and brief tutorial to learn the software. The operations and condition logics were helpful to learn easily. I relearned and reviewed the concept again and which will help me in the course further.