Laboratory Exercise #1

Design Flow of Digital Systems

(Introduction to Intel® Quartus® Prime Lite Edition)

Name: _____ **Group:** _____

Target Course Outcomes:

CO1: Create descriptions of digital hardware components such as in combinational and sequential circuits using synthesizable Verilog HDL constructs.
CO2: Verify the functionality of HDL-based components through design verification tools.

Intended Learning Outcomes:

- Familiarize software tool (Intel® Quartus® Prime Lite Edition) and its basic features
- Create design entry of a logic circuit using Verilog HDL code
- Synthesize a circuit specified in Verilog HDL

Supplement:

To proceed with this laboratory exercise, make sure that you have already downloaded the installer for **Intel® Quartus® Prime Lite Edition version 20.1** ***(or the later version, v20.1.1).*** Please refer to **Software Tools** in Canvas *(under Unit 0: Course Orientation in Modules)* for download and other related links.

If you downloaded the Intel software using the **Combined Files (Multiple Files)** option, unzip the .tar files and run one of the following files to start the installation process: **setup.bat** or **setup.sh**.

Instructions:

After successful installation of the recommended software tool, proceed with the hands-on exercises. This laboratory exercise is intended to be done individually.

Exercise 1A: Getting Started

Each circuit designed in Quartus Prime software is placed in a ***project***. Quartus works on one project at the time and keeps all project files in a single directory or folder in the file system.

1. Begin by starting the Quartus Prime software. Figure 1 shows the main display of the software. Most of the commands are accessible by the menu buttons. To start a new project, select **File > New Project Wizard** *(or click the **New Project Wizard** in the Home Tab shown in Figure 1)*. Then, click **Next** to reach the display for setting a ***working directory***, as shown in Figure 2. You may choose where you want to place your project files *(any local directory in your computer)*. Enter the project name ***LightControl*** and click **Next**. If the directory does not exist, Quartus Prime software will ask for your permission to create the desired directory and click **Yes.**

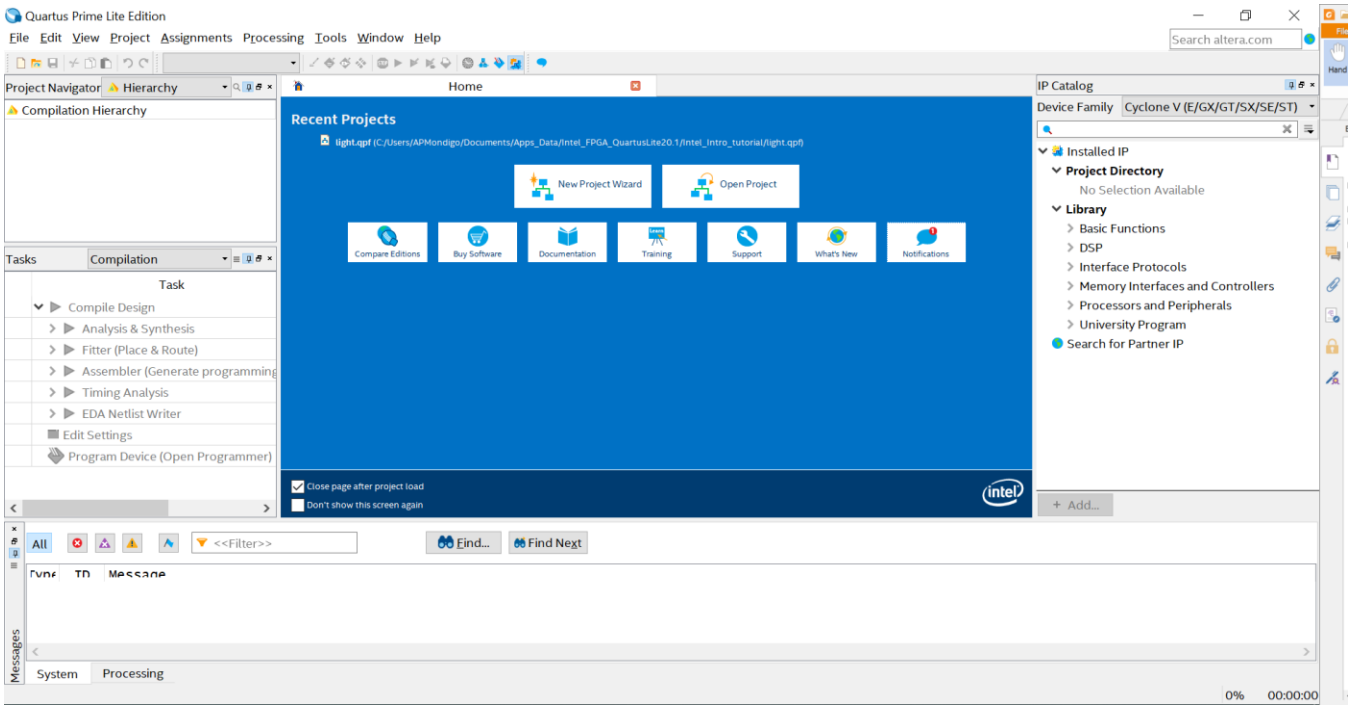


Figure 1. Main Quartus Prime Display

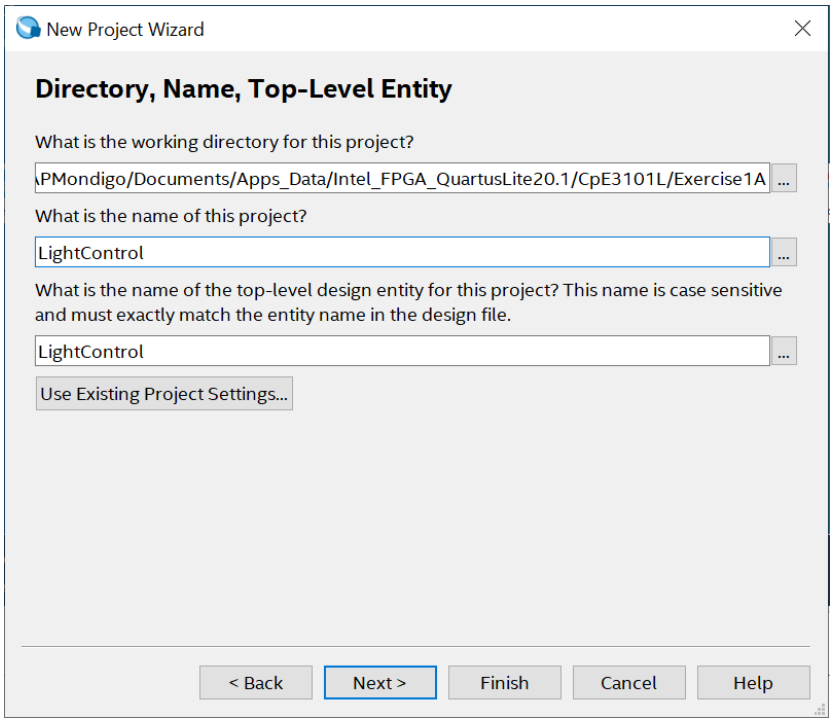


Figure 2. Create a New Project

2. Figure 3 shows the next window, where the **Project Type** is selected. Select **Empty Project** and click **Next**.

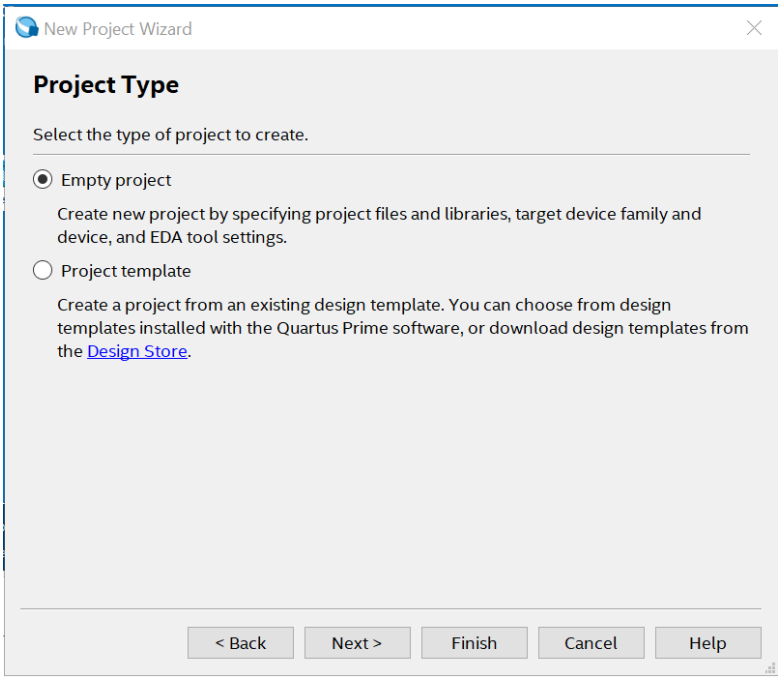


Figure 3. Choose the Project Type

3. The project wizard simplifies the management of project files especially in dealing with larger designs. If there are any existing design files to be included in the new project, the next window shown in Figure 4 allows the designer to add these files. Since this is an entirely new project, just click **Next** *(no existing files to be added).*

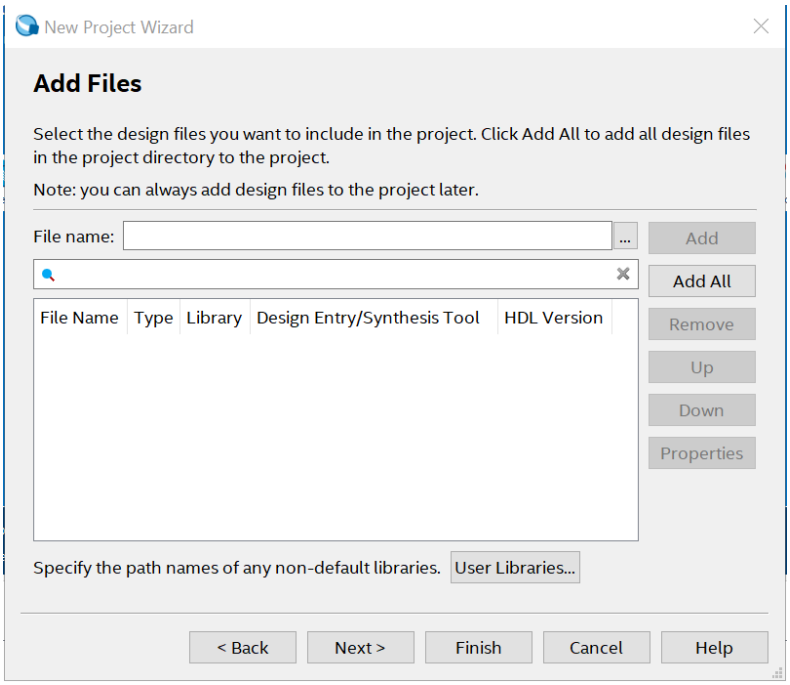


Figure 4. Include Existing Design Files

4. Then, the type of FPGA device for design circuit implementation will be selected. Figure 5 shows the next window displayed on the screen. Choose the **MAX 10** series device family for the DE-series board. For this course, target board is **DE10-Lite board**, which has an **Intel Max 10 10M50DAF484C7G FPGA**. Under the **Device** tab, explicitly search for the **specific Max 10 device (10M50DAF484C7G)** in the list of available, as shown in Figure 5.

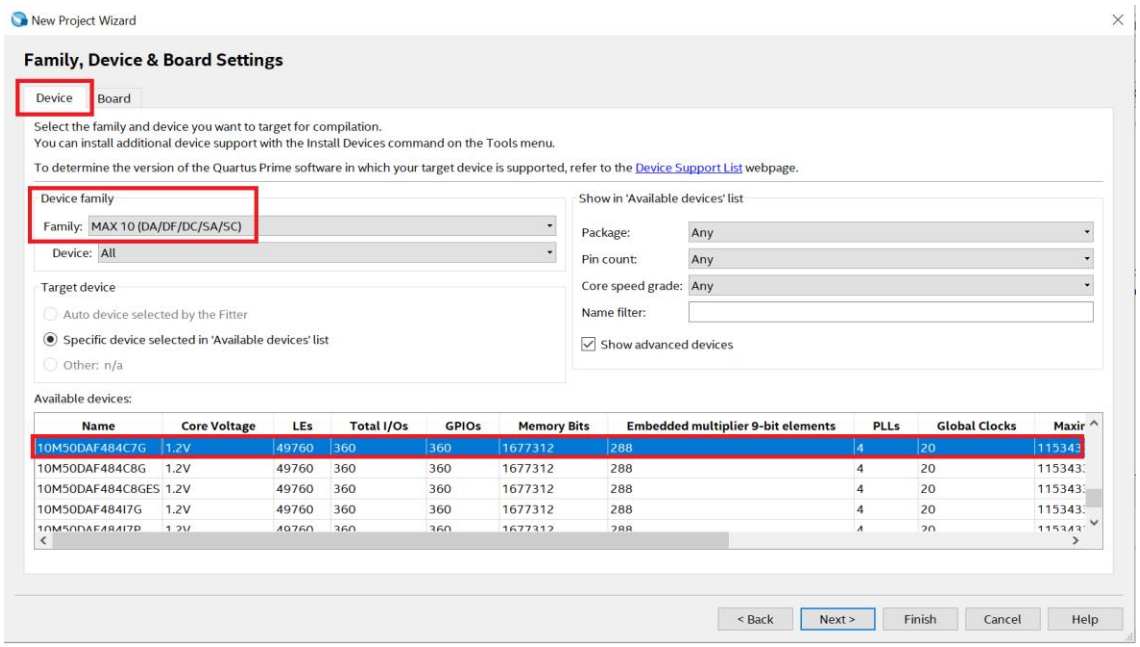


Figure 5. FPGA Family and Device Settings

5. The next window in Figure 6 is displayed to select third-party tools that will be used together with Quartus Prime tools. A commonly used term for **computer-aided design (CAD)** software for electronic circuits is **Electronic Design Automation (EDA)** tools. Mainly, we will be relying on Quartus Prime tools *(except for functional simulation)*, but for Laboratory Exercise #1, we will not choose other tools and just click **Next**.

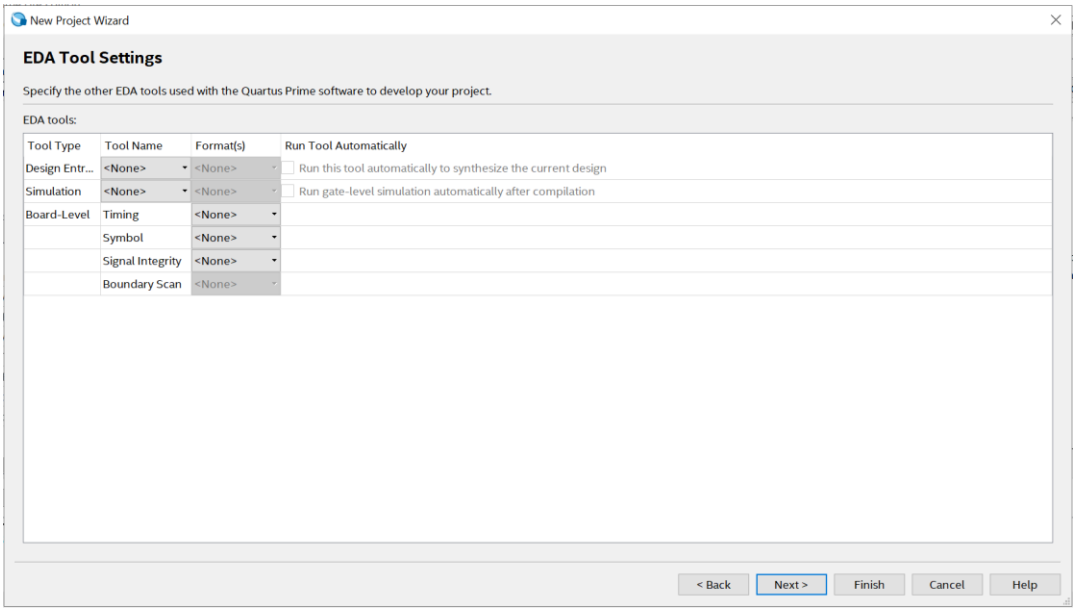


Figure 6. Specify Other EDA Tools

6. A summary of chosen settings will be shown in the next window, as displayed in Figure 7. Press **Finish.**

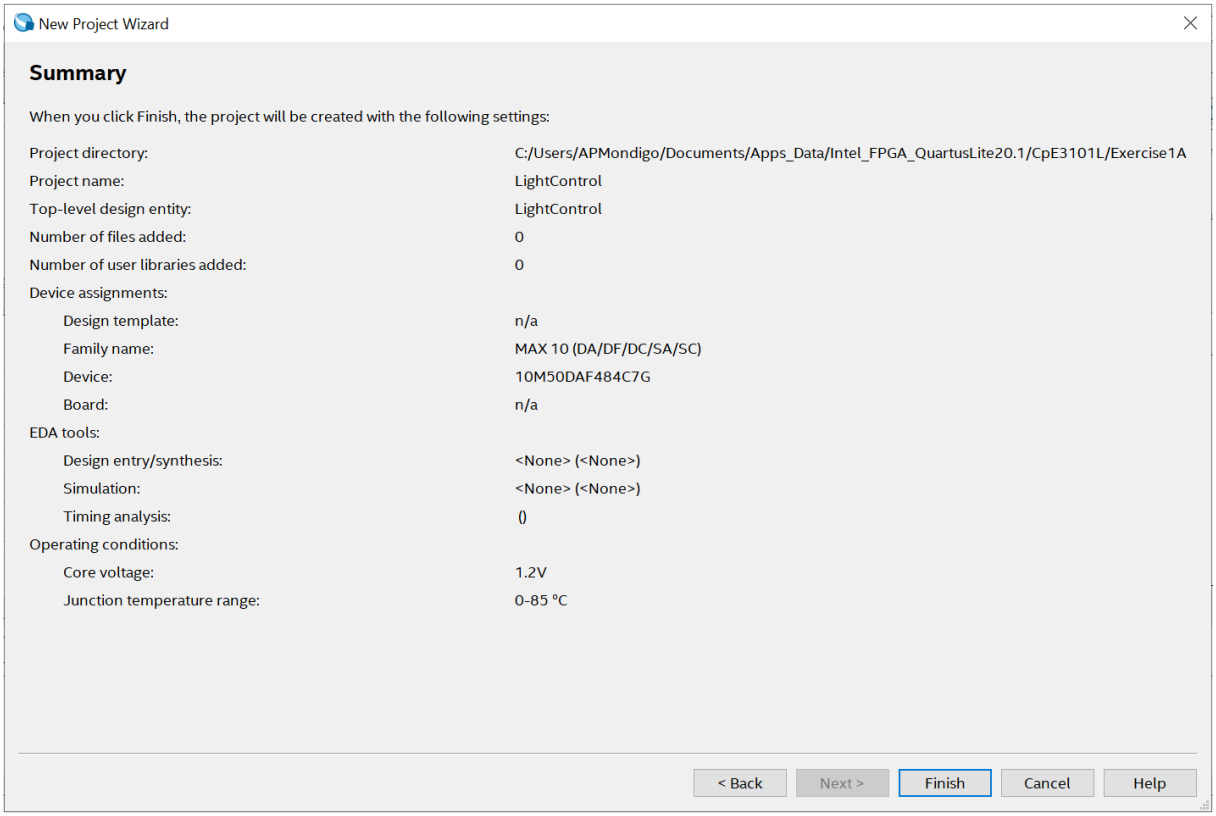


Figure 7. Summary of Project Settings

7. Now, the project wizard closes and displayed window will be the main Quartus window again, as shown in Figure 8. Take note of the title bar that shows the newly created project, ***LightControl***. The project is now successfully created.

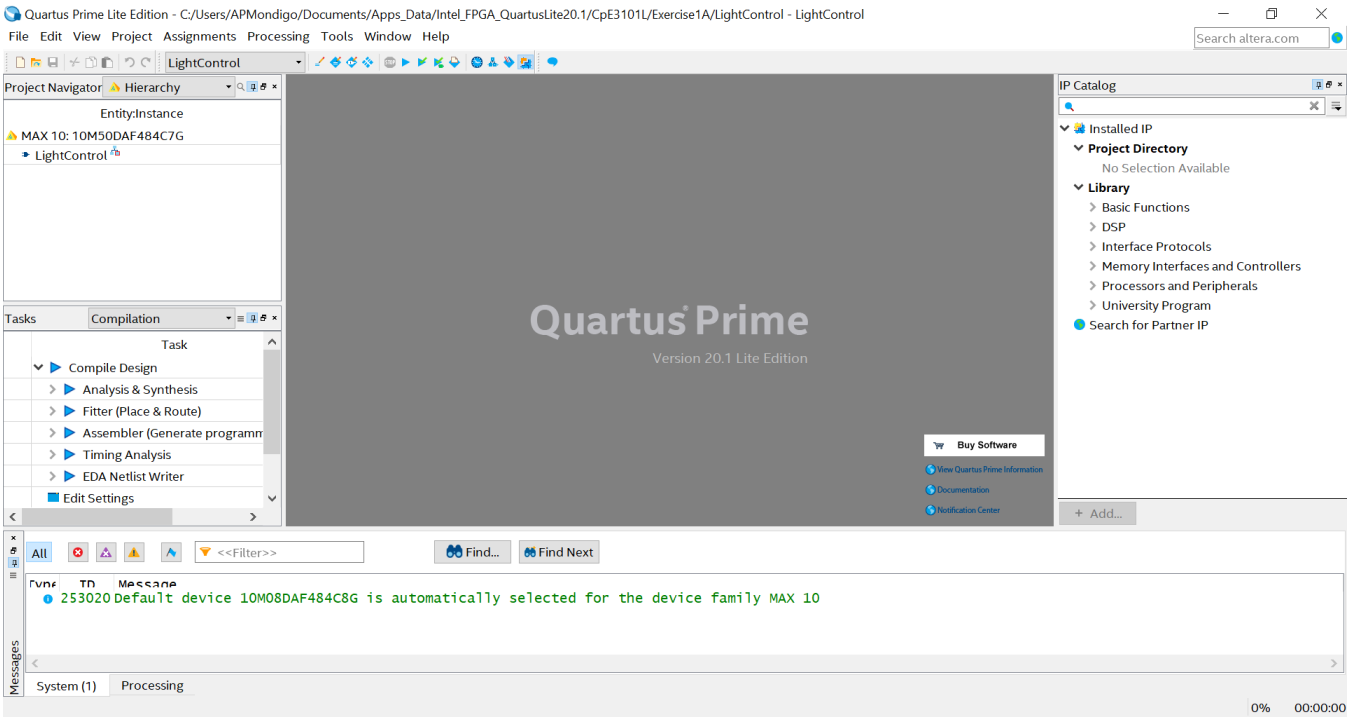
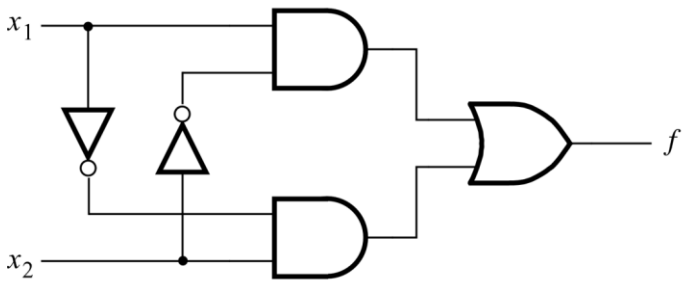


Figure 8. Quartus Prime Window for the Newly Created Project

Exercise 1B: Design Entry and Synthesis with Verilog HDL

Using the newly created project ***LightControl*** in Exercise 1A, we will create our first design entry using Verilog HDL. As a design example, a **two-way light controller circuit** shown in Figure 9 will be used. This circuit simply controls a single light (output f) from either two switches (inputs x_1 and x_2). Its truth table is also shown. You may notice that this is the Exclusive-OR function of the two inputs, but for this exercise, the specific gates in Figure 9 will be used.



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

Figure 9. Two-Way Light Controller

For the design entry with Verilog HDL, any text editor may be used. In this exercise, however, Quartus Prime text editor will be used to demonstrate the tool. ***The filename can be given any name, but it is a common designer's practice to use the same filename as the name of the top-level Verilog module***. In this case, it is ***LightControl*** and its extension must be .***v***, to indicate a Verilog file (i.e., **LightControl.v**).

1. Select **File > New** and choose **Verilog HDL File** under **Design Files.** This will open the text editor window.

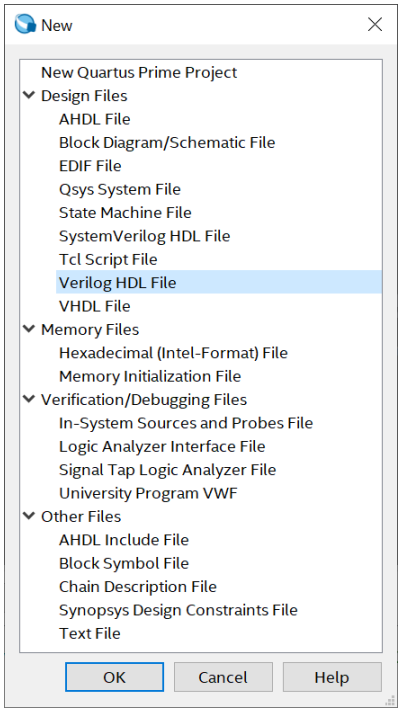


Figure 10. Create a New Verilog File

2. Using the text editor, specify the file name (**LightControl.v**) by selecting **File > Save As** to open the window shown in Figure 11. Make sure that the **Save as type** is **Verilog HDL Files**. In addition, select or put a checkmark on the option **Add file to current project** and click **Save.**

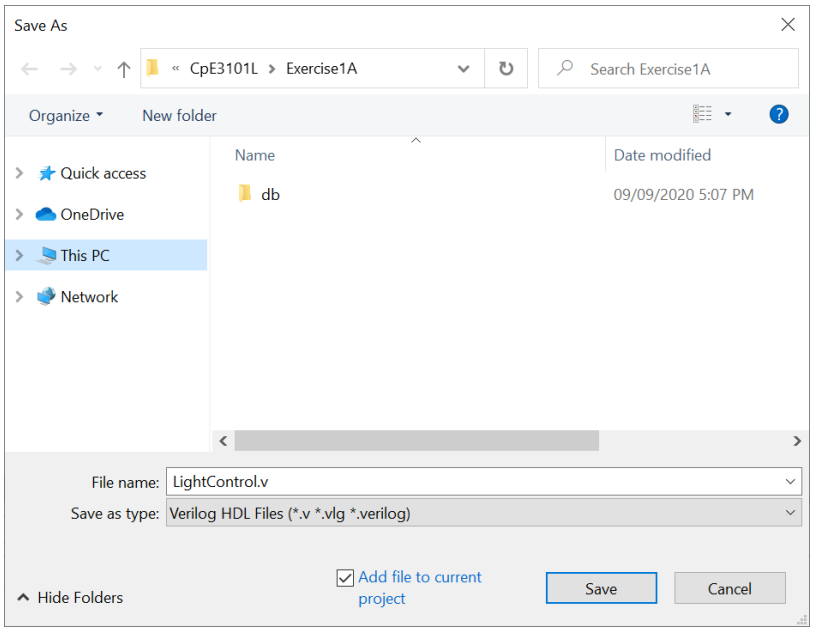


Figure 11. Save and Add Verilog File to the Current Project

3. Once the file is named appropriately, ***enter the Verilog HDL code of the two-way light controller***, as shown in Figure 12. At this point, don't worry about the Verilog HDL code details yet. However, please take extra caution in the design entry since **Verilog is a case sensitive language. Save** the file.

LightControl.v

```
1  //
2  // Verilog HDL code for a two-way light controller
3  //
4  module LightControl (x1, x2, f);
5
6     input    x1, x2;
7     output   f;
8
9     wire     wN1_out, wN2_out, wA1_out, wA2_out;
10
11    not      N1 (wN1_out, x1);
12    not      N2 (wN2_out, x2);
13    and      A1 (wA1_out, x1, wN2_out);
14    and      A2 (wA2_out, wN1_out, x2);
15    or       O1 (f, wA1_out, wA2_out);
16
17 endmodule
18
```

Figure 12. Verilog HDL Code (LightControl.v)

4. In case the Verilog file was not successfully added to the current project, you may also click on **Project > Add Current File to Project** in the menu bar**.** To see the list of files already included in ***LightControl***, select **Assignments > Settings.** A window, as shown in Figure 13, will be displayed. Under **Category**, select **Files.** Make sure that **LightControl.v** is included in the list. Similarly, adding more files to a current project may also be done here.

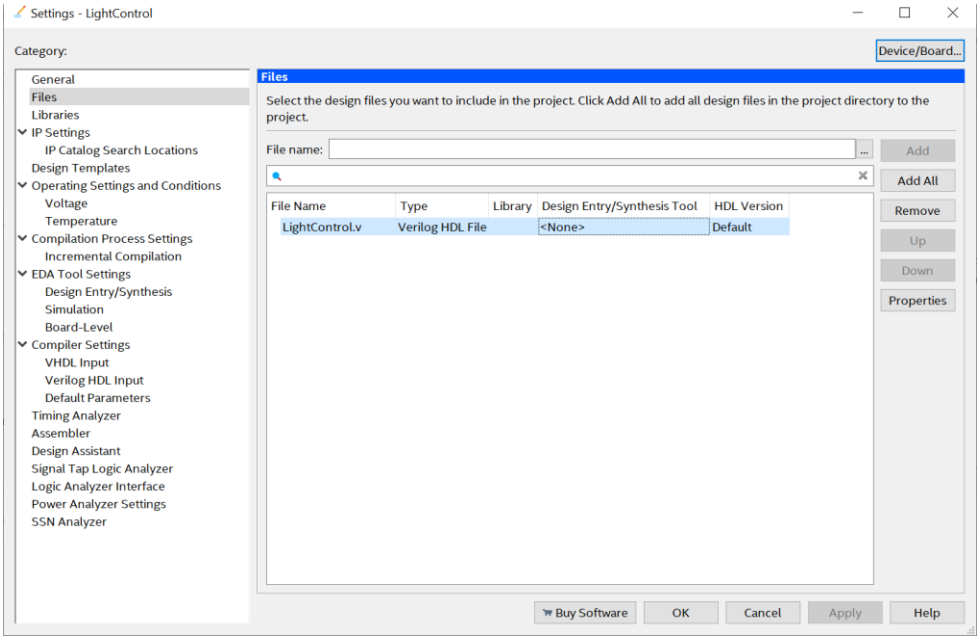


Figure 13. Settings Window

5. The design entry or Verilog HDL code in this project **(LightControl.v)** is processed by several Quartus Prime tools, such as ***analyze*** the code, ***synthesize*** the circuit, and ***generate*** a hardware implementation on the target FPGA. The tools are controlled by an application program in Quartus called ***Compiler.***

   In this exercise however, there's no need to fully compile the design yet. For a faster design check, even without functional or logic simulation, a design entry can be analyzed and synthesized. Under **Tasks** window pane, expand **Compile Design** and click on **Analysis & Synthesis** button. Refer to Figure 14.

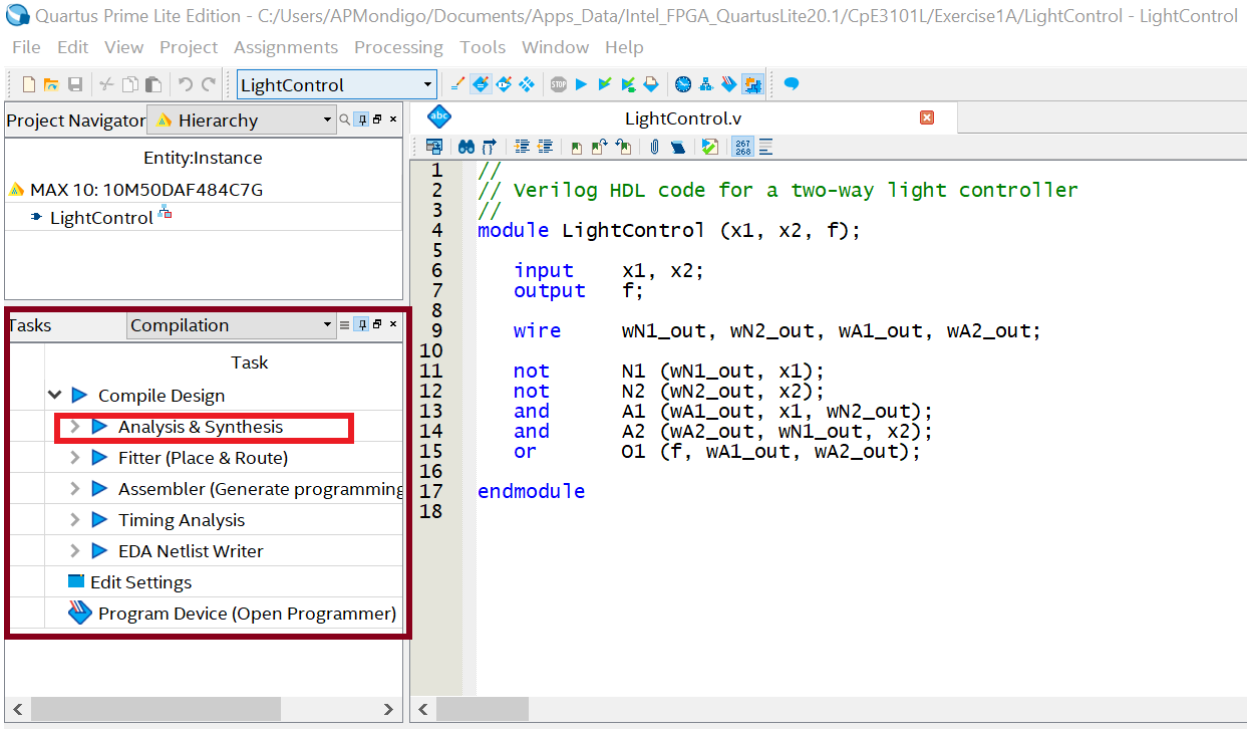


Figure 14. Analysis and Synthesis of Design Entry

6. As synthesis is being done, its progress can be shown in the **Tasks** window at the left side. At the **Message** window at the bottom, various messages are displayed. In case of errors, appropriate messages will be shown here. Once analysis and synthesis are done *(indicated by a green check mark)*, a **Compilation Report** is displayed. All these are shown in Figure 15.

   In the **Flow Summary** under **Compilation Report**, it can be seen that ***only 1 logic element and 3 pins are needed*** to implement the two-way light controller circuit on the target FPGA chip.

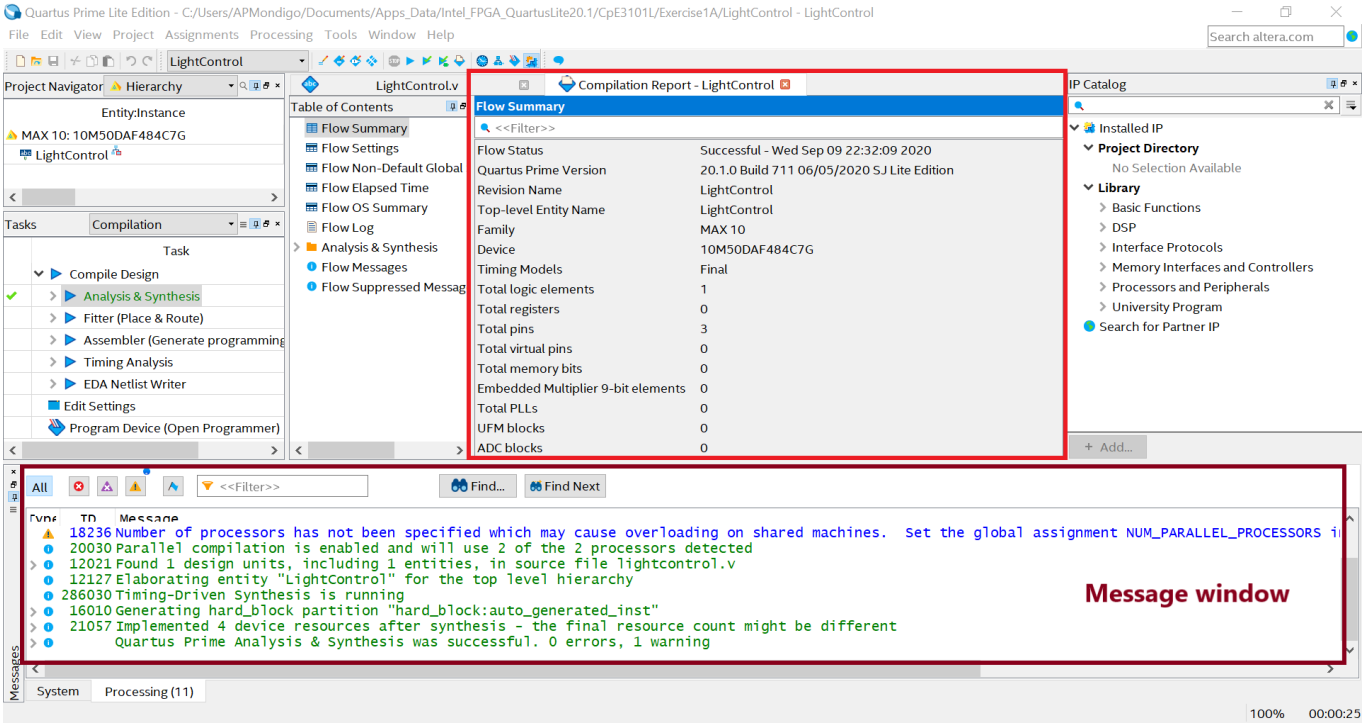


Figure 15. Display after Successful Analysis and Synthesis

7. To briefly see the synthesized circuit, you may do so by expanding the **Analysis & Synthesis** button. Under **Netlist Viewers**, click **RTL Viewer.** This is shown in Figure 16. ***RTL means register transfer level*** and the ***RTL Viewer*** allows a preview of the schematic design netlist *(i.e., interconnections of logic elements and other components)* after synthesis. However, this view is not the final structure of the design, since not all optimizations are included in synthesis. Instead, this is the closest possible view to the original design. See Figure 17 and compare it with Figure 9.

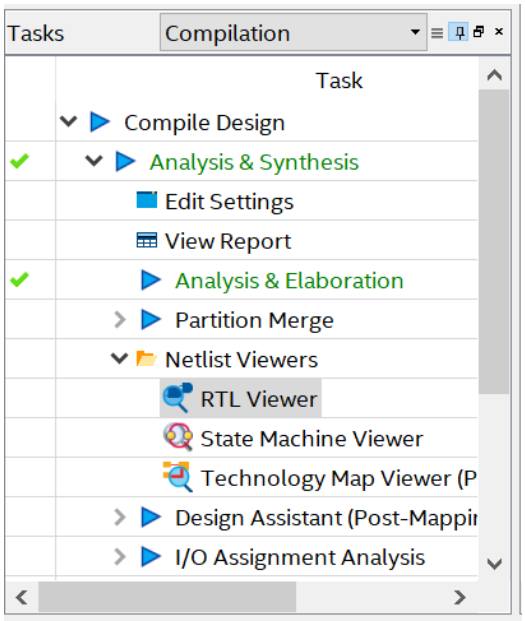


Figure 16. RTL Viewer

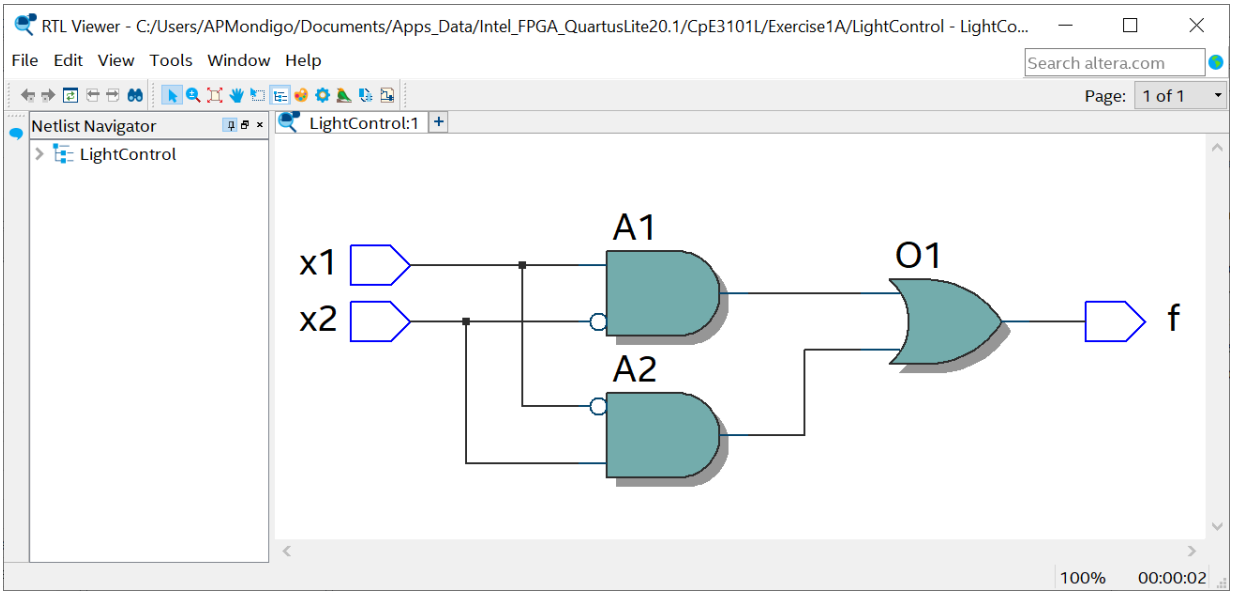


Figure 17. Synthesized Schematic Diagram for LightControl.v (as seen in RTL Viewer)

Note: **Functional simulation procedures** will be performed in Laboratory Exercise #2. **FPGA pin assignment** and **other settings for FPGA programming and configuration** will be skipped in this current version (Laboratory Exercise #1 v3.0) since FPGA implementation is not currently possible.

Exercise 1C: Half Adder Circuit

Create a new project following the detailed procedures in Exercise 1B and add this Verilog description of a ***half adder circuit*** to the project. After the design entry, analyze and synthesize the design. After successful synthesis, use the RTL Viewer to check the schematic diagram of the synthesized design.

Project Name: **HalfAdder**	***Verilog HDL Filename:*** **HalfAdder.v**
Inputs: x, y	***Outputs:*** C, S

An **entity diagram** shows a top-level block diagram of the intended circuit design. Figure 18 shows the entity diagram of the half adder circuit, while Figure 19 shows its truth table and logic diagram. For the design entry, use the Verilog HDL code shown in Figure 20. Again, remember that **Verilog is case sensitive.**

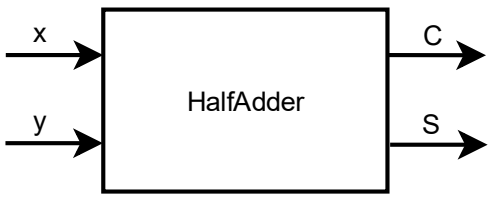
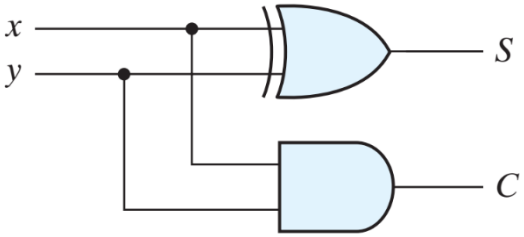


Figure 18. Entity Diagram of Half Adder Circuit



x	y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Figure 19. Half Adder Circuit and Truth Table

HalfAdder.v

```
1  //
2  // Verilog HDL code for a half adder circuit
3  //
4  module HalfAdder (x, y, C, S);
5
6     input    x, y;
7     output   C, S;
8
9     xor      X1 (S, x, y);
10    and      A1 (C, x, y);
11
12 endmodule
13
```

Figure 20. Design Entry for Half Adder in Verilog HDL

For the Laboratory Report, prepare and take screen shots of the following items for Exercise 1C:

- **Project Settings Summary**, shown after creating new project *(such as in Figure 7)*
- **Verilog HDL Design Entry**, showing your ***name*** and ***course group with schedule*** in the comments *(such as in Figure 20)*
- **Compilation Report for the Flow Summary**, showing the ***entire Quartus Prime window*** *(such as in Figure 15)*
    - Indicate how many logic elements and pins were used.
- **Schematic Diagram of Synthesized Circuit** using RTL Viewer *(such as in Figure 17)*

Other instructions for **Laboratory Report #1** will be given in Canvas.

Evaluation:

Criteria \ Level	1.0 Outstanding	2.0 Competent	3.0 Marginal	5.0 Not Acceptable	Rating
CO1: Verilog Design Entry	Verilog HDL description is correct and follows specified instructions.	--	Verilog HDL description is correct but DID NOT follow the specified instructions.	NO Verilog HDL description is presented or the design entry is INCORRECT.	

CO1: Design Synthesis	Verilog HDL description is synthesized and compiled with NO issues (with 0-1 warning). There is clear evidence in the lab report that supports this result.	Verilog HDL description is synthesized and compiled with MINOR issues on constructs (with 2-3 warnings). There is clear evidence in the lab report that supports this result.	Verilog HDL description is synthesized and compiled with MAJOR issues on constructs (with 4 or more warnings). There is evidence in the lab report that supports this result.	Simulation has FAILED or there is NO evidence showing a successful simulation.	