ENEL 453 Lab 1: FPGA Slide Switch Reader in Decimal and Hexadecimal Format

1. Revision History

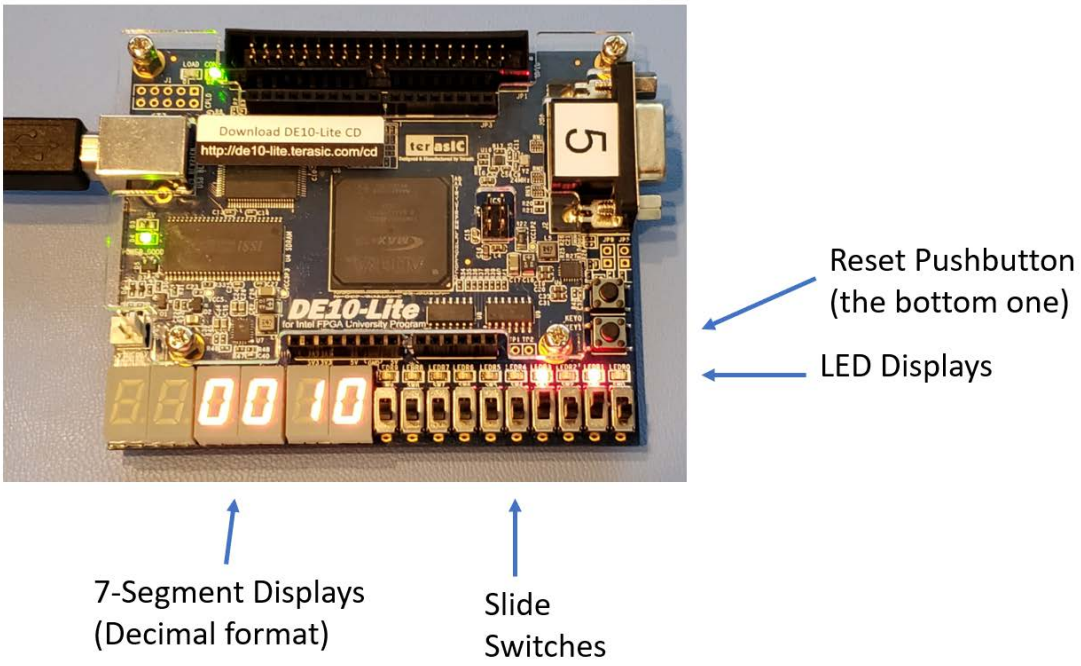
Version	Comments
v1	Initial document

Path: C:\Users\donen\Dropbox\U of C\Teaching\ENEL 453\ENEL 453 F2020\Labs\Lab 1\ENEL 453 Lab 1 v1.docx

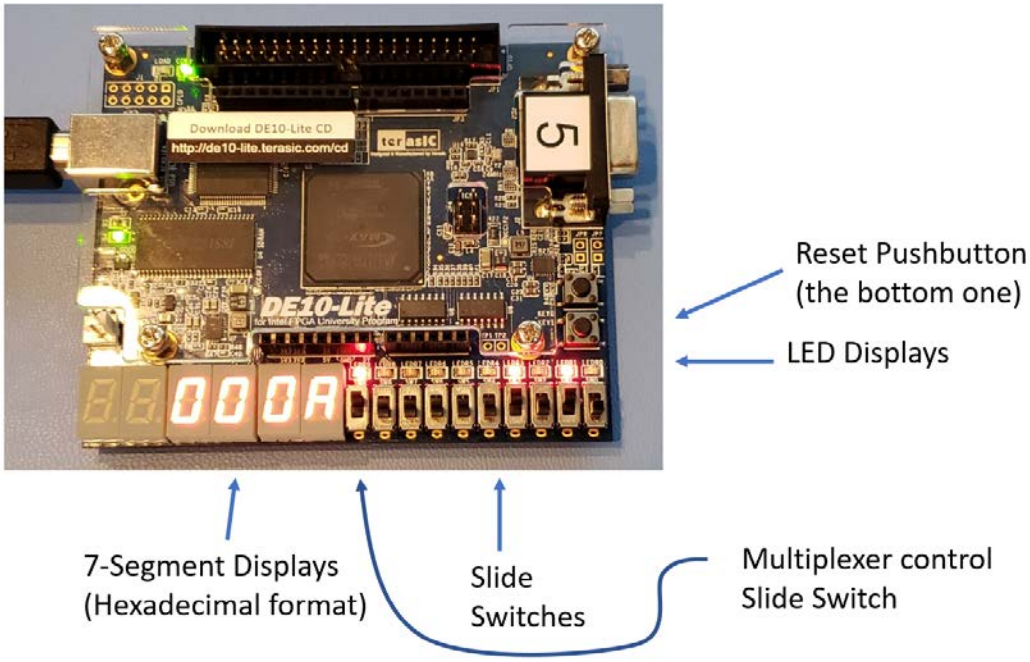
2. Introduction

This is the first lab project of the course and its focus for the team is to become familiar with Quartus, ModelSim, and the DE10-Lite FPGA board. You will be given a reference design and you will become familiar with it and extend the design to add some limited functionality.

The reference design will read the slide switch inputs (SW0 to SW9) and generate a Decimal value on the 7-segment displays, based on the slide switch positions. The LEDs (LED0 to LED9) on the DE10-Lite will light up depending on the slide switch positions. In the picture below, the slide switches SW7 to SW0 represent 00001010 (Binary), which becomes converted in Decimal to 0010 (2 leading zeros).



You will extend the reference design by adding a Multiplexer to select either a Decimal value or a Hexadecimal value to enter the 7-segment displays. This Multiplexer will have a slide switch to control whether Decimal or Hexadecimal values will be presented on the 7-segment displays. In order to be able to present Hexadecimal values, you must also extend the coded values of the 7-segment decoder. In the picture below, the slide switches SW7 to SW0 represent 00001010 (Binary), which becomes converted in Hexadecimal to 000A (3 leading zeros).



The basic elements of this project are to:

- Review all lab and lecture videos to date. Especially, practice the Quartus and ModelSim projects as described in their respective introduction videos (Lecture 3), so that you can use the software tools effectively. You must get to the point where you can independently run the example project in the introduction videos, for both Quartus and ModelSim, without referring to the videos.
- Get the reference files for Lab 1 and create new Quartus project and download the FPGA configuration to the DE10-Lite. Experiment with the DE10-Lite to understand the functionality of the design in hardware.
- Modify the design in Quartus to add the required functionality. Create ModelSim testbenches as necessary to help verify the functionality of your design. Download your FPGA configuration to the DE10-Lite to verify the design in physical hardware.
- Upload the required videos and documents to the D2L Dropbox before the deadline.

3. Technical Requirements

The project technical requirements are as follows.

3.1. Insert a Multiplexer into the reference project to allow the HEX outputs to display either Hexadecimal or Decimal values. The Multiplexer must be controlled by a slide switch and you must decide which one to use. You must provide a testbench for Multiplexer.

Hints:

1. You may modify and use the MUX2TO1.vhd and tb_MUX2TO1.vhd files from the Quartus and ModelSim introduction videos to serve as the starting point for the design of your Multiplexer.
2. You will have to modify top_level.vhd to include the *component declaration* and *component instantiation* of Multiplexer. You will have to modify the signals connecting the various components in top_level, in order to make the correct connections.
3. It may be useful for you to draw out the design of top_level on a piece of paper (use the RTL Viewer to help visualize the circuit), then draw the modifications required to add and connect the Multiplexer, then code what you drew. *Drawing out circuits has been reported by many students as very helpful for designing with FPGAs.*
4. Refer to the RTL Schematic (next page) of the instructor's solution to see the logical position of the Multiplexer in the design.

Figure 1 Reference Project (given to students)

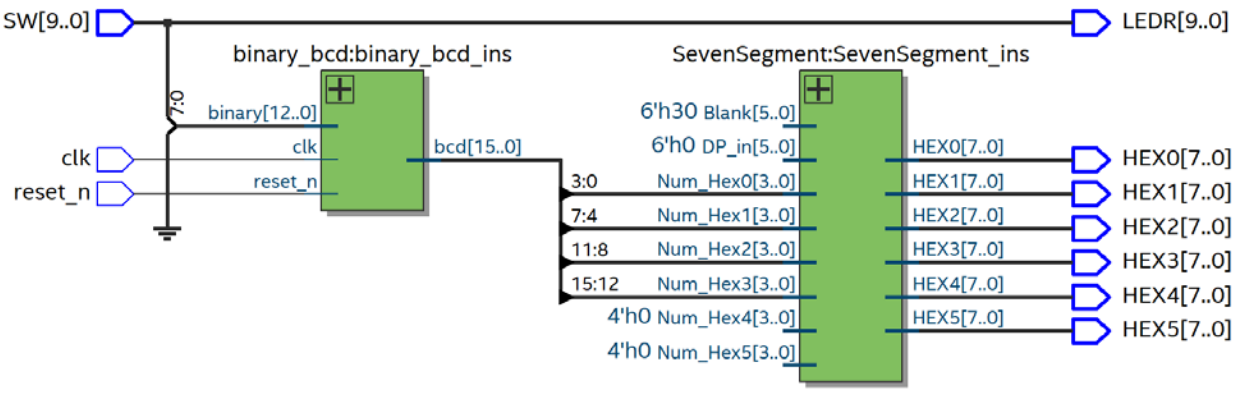
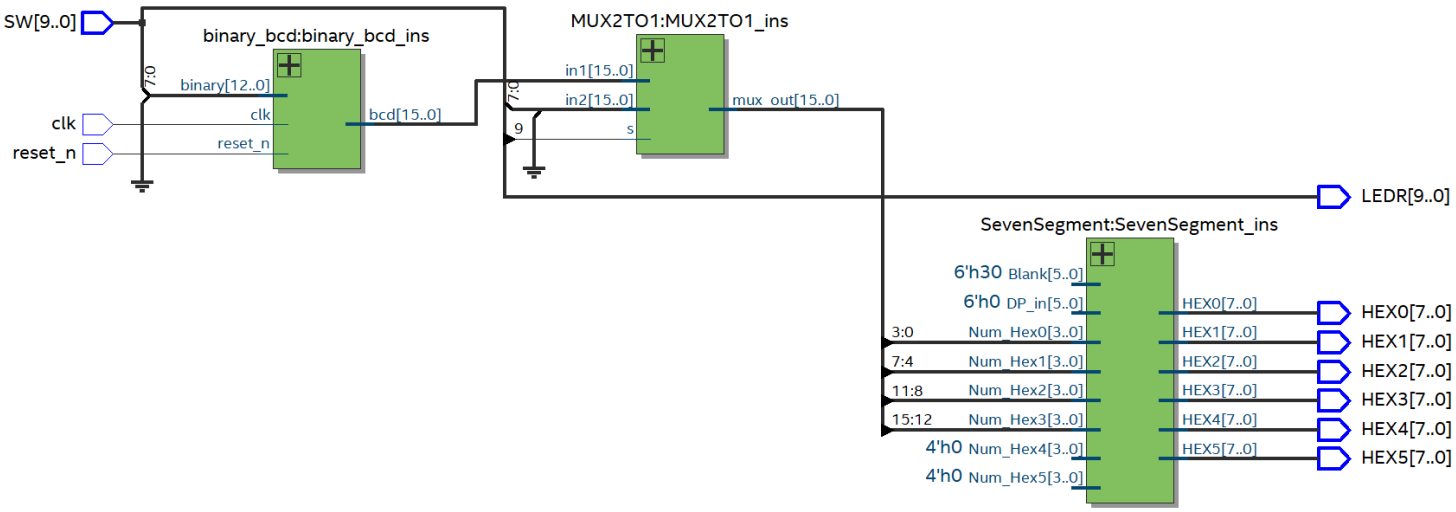


Figure 2 Instructor's Solution



3.2. Modify the SevenSegment_decoder.vhd file to extend the 7-segment digit decoding to cover Hexadecimal numbers (A, B, C, D, E, and F). The current file only covers Decimal numbers (0 to 9).

Hint: The SevenSegment_decoder.vhd file has a comment on *line 35* to indicate where to make the edits (e.g. where to continue the decoding pattern from the previous lines). Code snippet below.

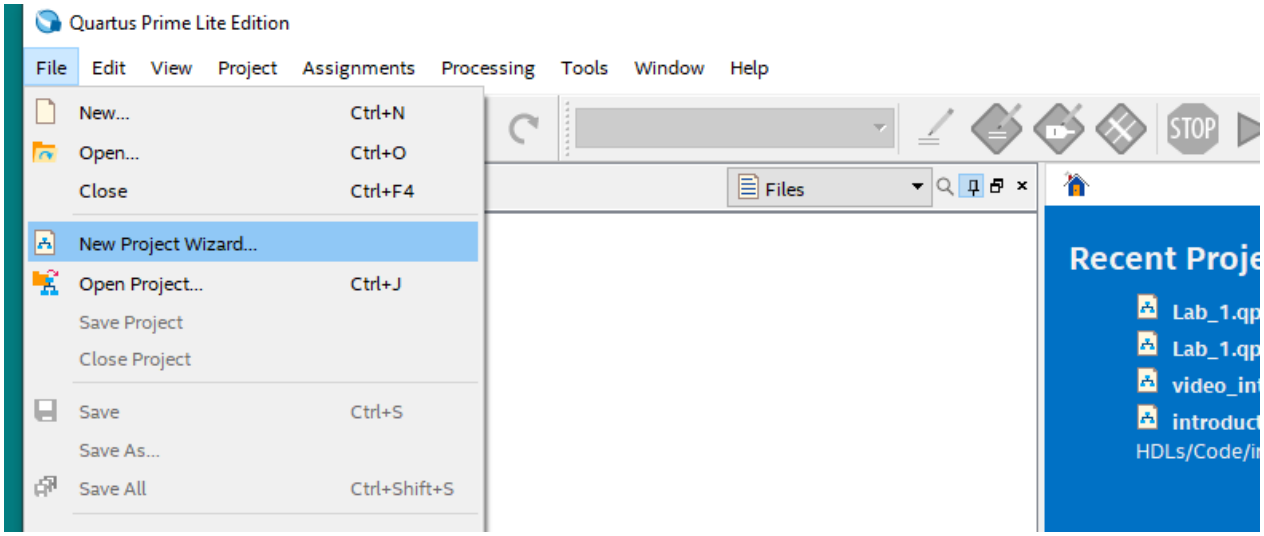
```
1  -- --- Seven segment component
2  library IEEE;
3  use IEEE.STD_LOGIC_1164.ALL;
4
5  entity SevenSegment_decoder is
6      Port (
7             H        : out STD_LOGIC_VECTOR (7 downto 0);
8             input    : in  STD_LOGIC_VECTOR (3 downto 0);
9             DP,Blank : in  STD_LOGIC
10           );
11 end SevenSegment_decoder;
12
13 architecture Behavioral of SevenSegment_decoder is
14
15 signal seven_seg : STD_LOGIC_VECTOR (6 downto 0);
16
17 begin
18    Process (input,Blank)
19    begin
20       if(Blank = '1') then
21          seven_seg(6 downto 0)<="0000000"; -- this blanks the display
22       else -- don't blank and display the required digit
23                                                 --  Segments, 0=LED off
24          Case input is                          --  6543210
25             when "0000" => seven_seg(6 downto 0)<="0111111"; -- 0
26             when "0001" => seven_seg(6 downto 0)<="0000110"; -- 1
27             when "0010" => seven_seg(6 downto 0)<="1011011"; -- 2
28             when "0011" => seven_seg(6 downto 0)<="1001111"; -- 3
29             when "0100" => seven_seg(6 downto 0)<="1100110"; -- 4
30             when "0101" => seven_seg(6 downto 0)<="1101101"; -- 5
31             when "0110" => seven_seg(6 downto 0)<="1111101"; -- 6
32             when "0111" => seven_seg(6 downto 0)<="0000111"; -- 7
33             when "1000" => seven_seg(6 downto 0)<="1111111"; -- 8
34             when "1001" => seven_seg(6 downto 0)<="1100111"; -- 9
35             -- students add this block ------
36
37             ---------------------------------
38             when others => seven_seg(6 downto 0)<="0000000"; -- blank
39          End Case;
40       end if;
41    End Process;
```

4. Resources Provided

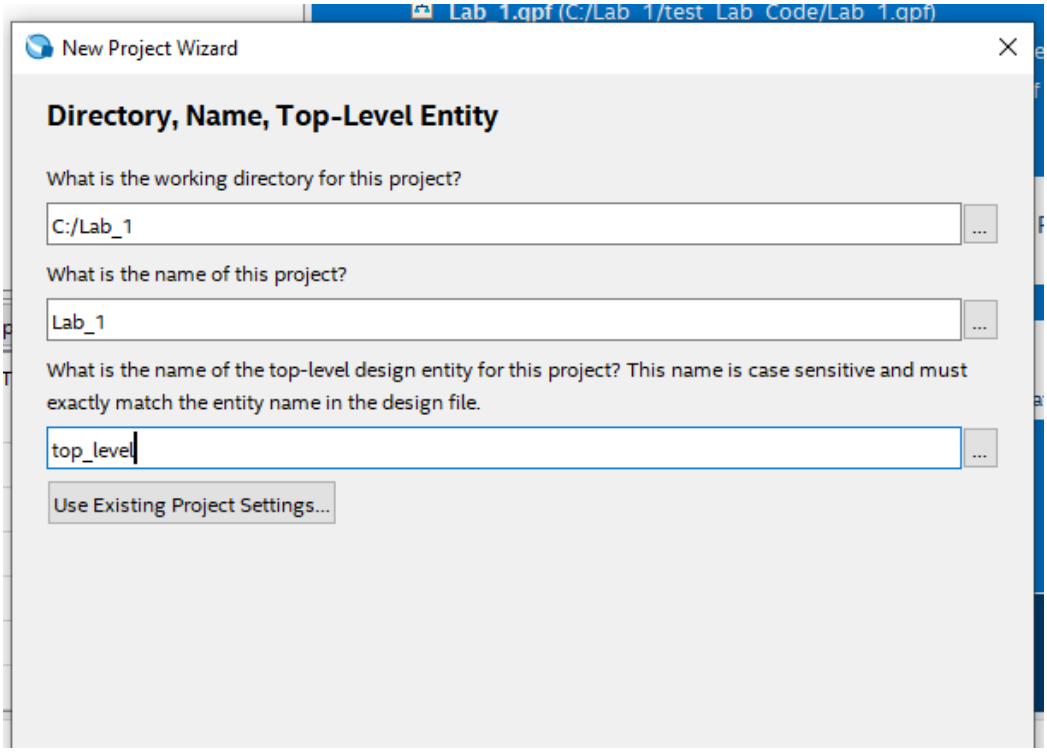
- DE10-Lite kit.
- Reference code that demonstrates:
  - Reading the slide switches;
  - Lighting LEDs based on slide switch positions;
  - Converting the slide switch positions to Decimal representation; and
  - Displaying Decimal digits on the 7-segment displays.
- Lab and lecture videos.

5. Creating the Project for Reference Design

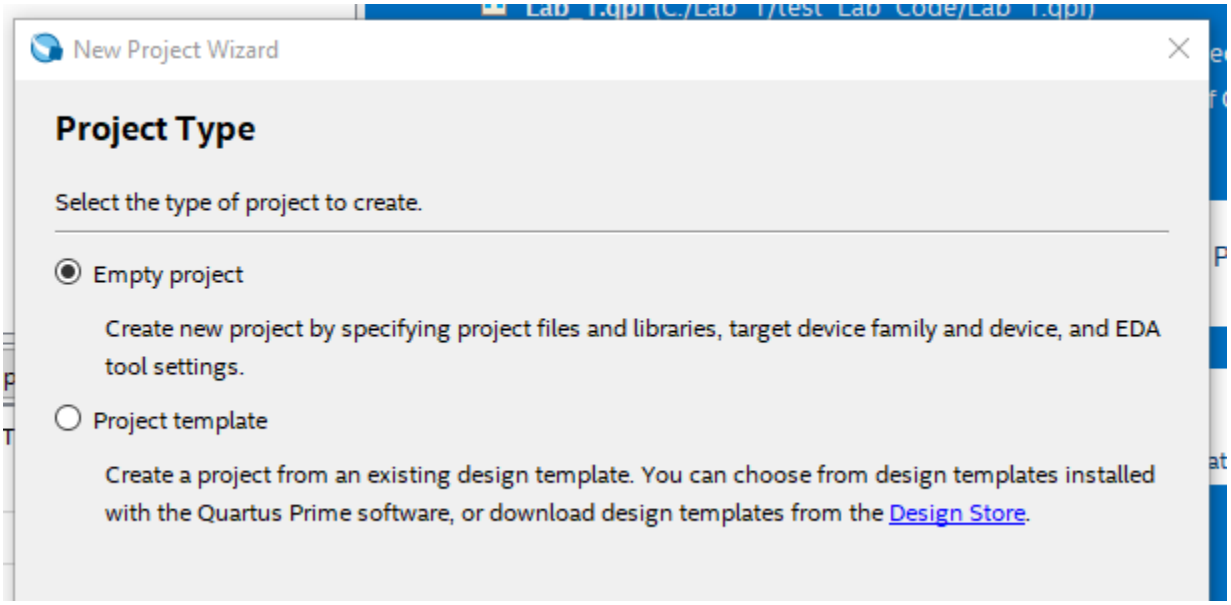
1. Download the .vhd, .sdc, and .qsf files from D2L and place them in a new directory, and you can call the directory Lab_1.
2. Launch Quartus and click File > New Project Wizard.



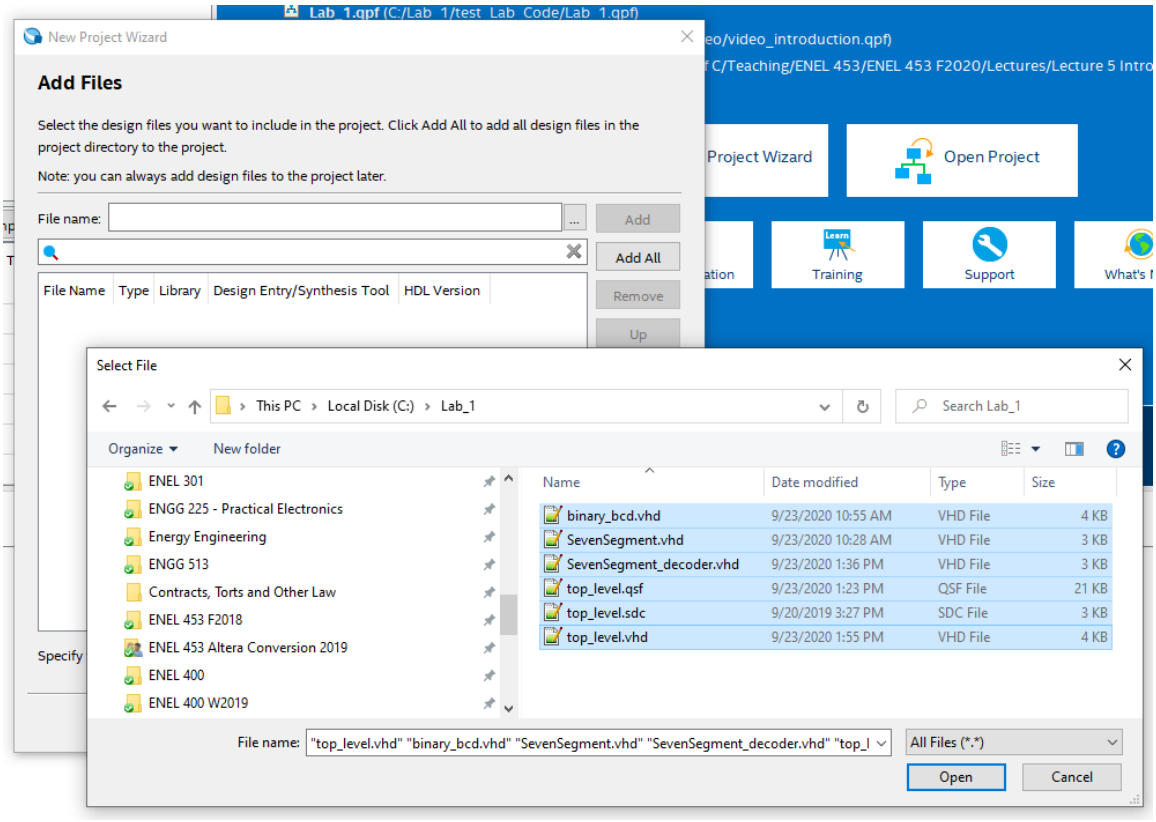
3. Click Next and fill in the New Project Wizard as follows (assuming your project directory is C:/Lab_1).

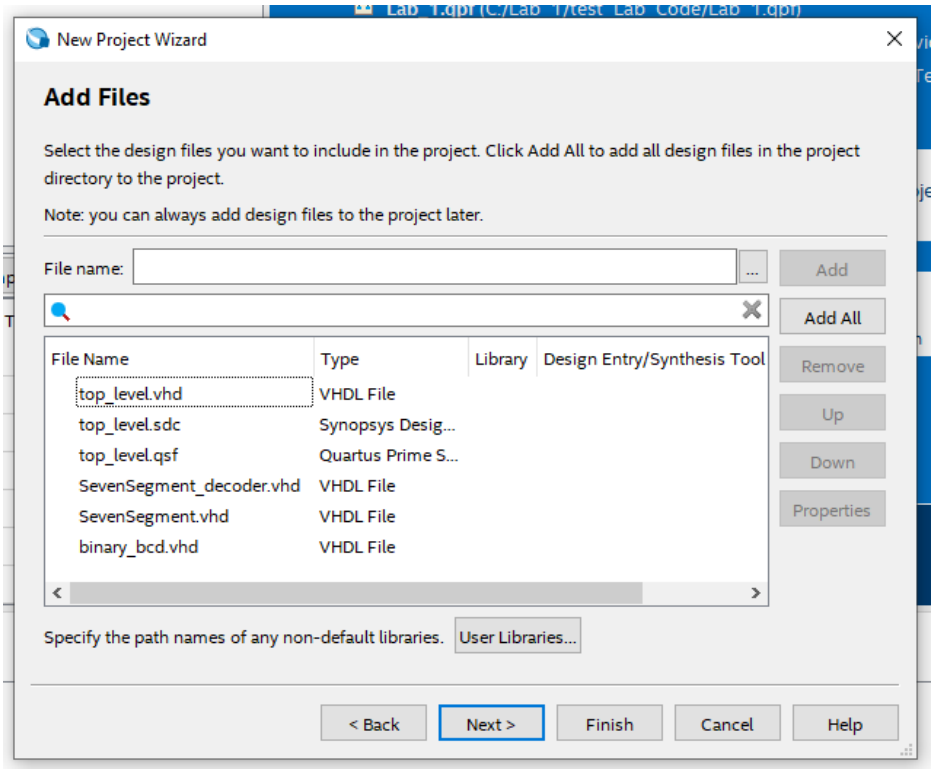


4. Click Next and select Empty project.

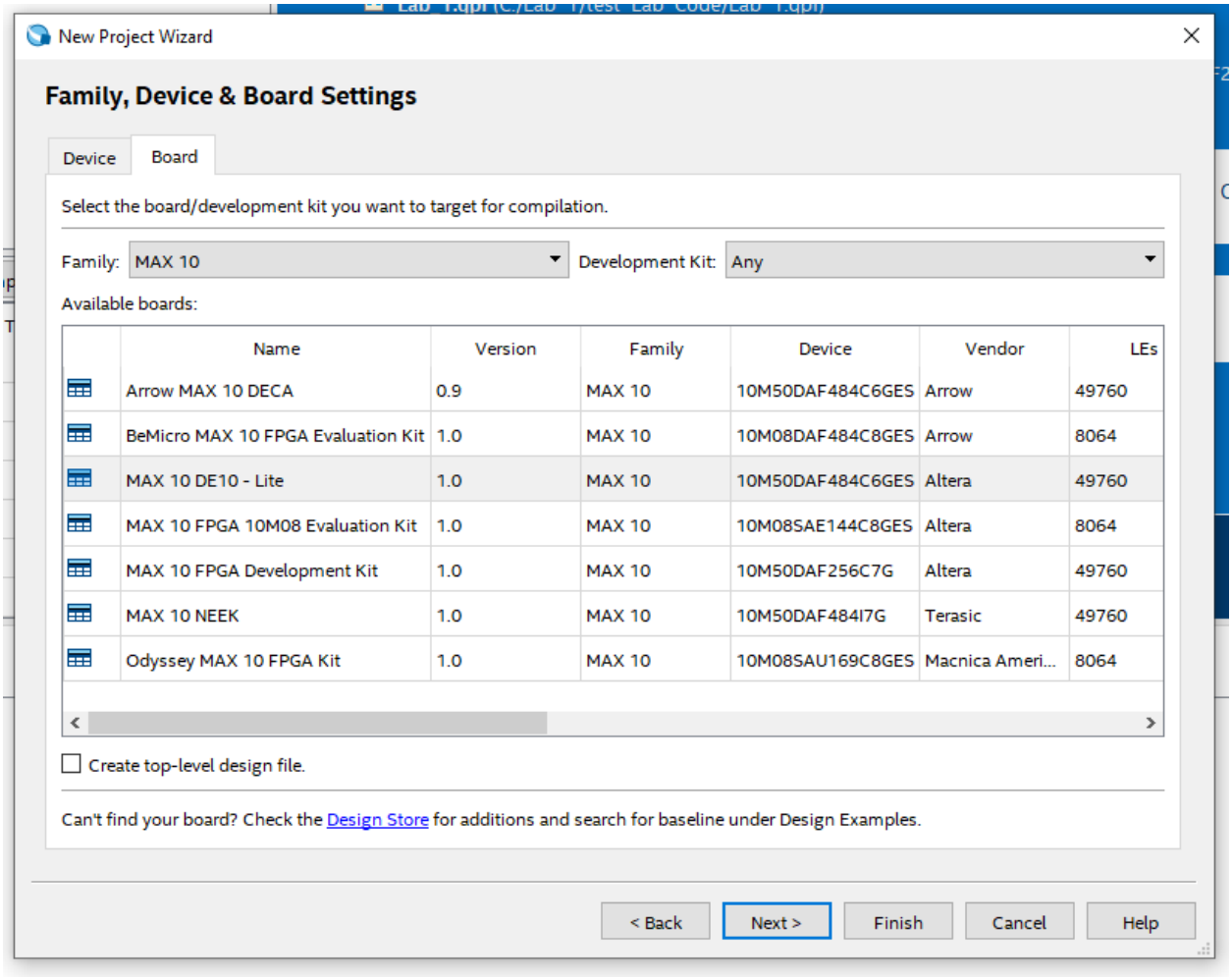


5. Click Next and browse (using the … button) to your Lab_1 directory where the files are, and add them.

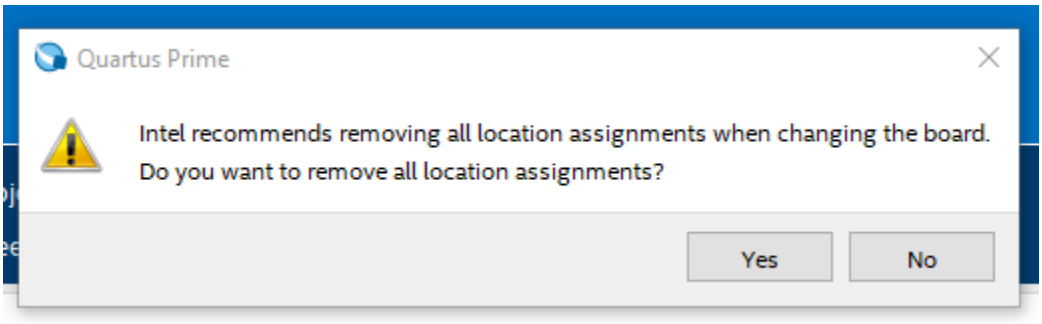




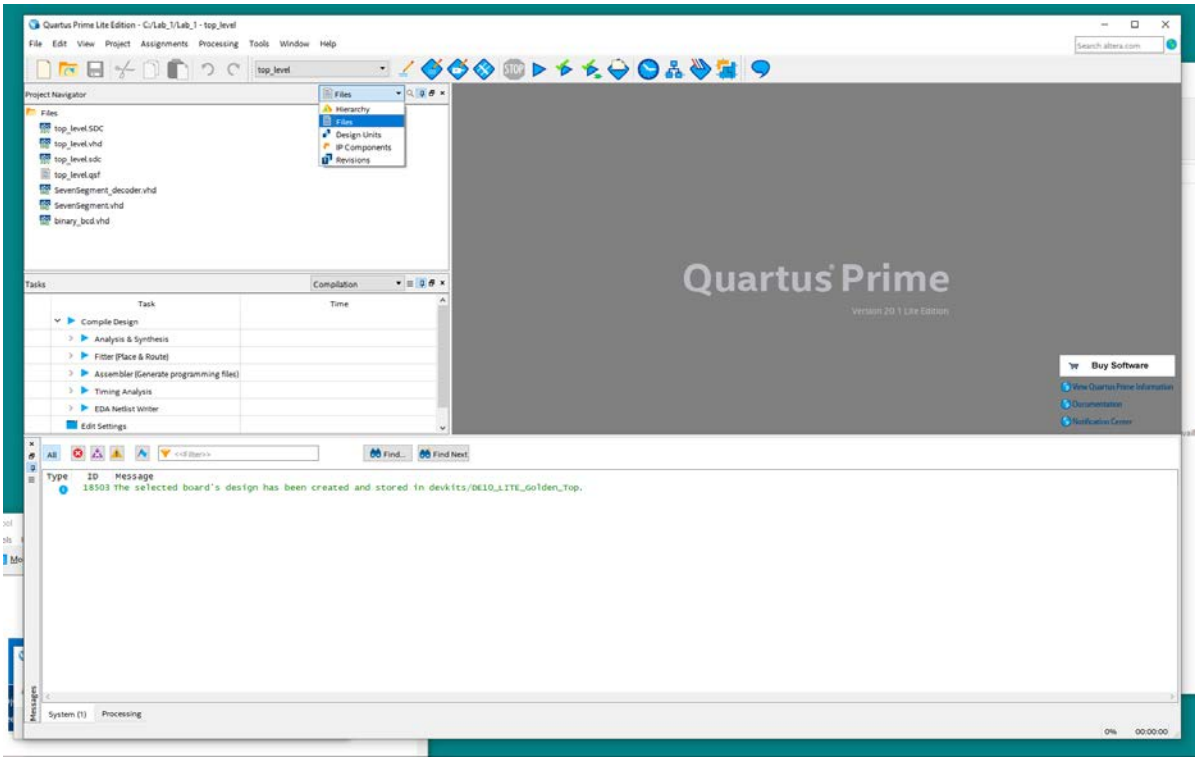
6. Click Next and select the DE10-Lite board (third from the top) from the Board tab. Unclick the "Create top-level design file" box



7. Click Next and Next and Finish. Then click "**No**" in the pop up window that says "Intel recommends removing all location assignments when changing the board. Do you want to remove all location assignments?" ***This is an important step so that you project doesn't get messed up and become difficult to debug. Otherwise, you will have to keep importing assignments every time you want to compile.***



8. Click Hierarchy and then click Files, to see the project files.



9. The previous steps should be an effective reminder on how to create a project in Quartus. Based on the Quartus introduction video, from now on, you should be able to recall to:
    a. View and edit the project files in the Quartus editor;
    b. Compile the files;
    c. Program the DE10-Lite FPGA board;
    d. Launch the RTL Viewer to see the schematic representation of the design; and
    e. Launch the Pin Planner to verify the actual locations of the assigned input and output pins from your top_level, matched to the .qsf file.

10. Based on the ModelSim introduction video, you should be able to create a project and simulate the design and view the results in the ModelSim waveform viewer.

6. Deliverables

The team will upload the following to the D2L Dropbox by **11:59 pm Friday October 8**.

1. VHDL code for their project:
    a. RTL code, i.e. the design code, for the complete project.
    b. The testbench for the Multiplexer you designed, which exercises the complete functionality of the Multiplexer (use the tb_MUX2TO1 as a reference).
    c. A testbench for the top_level, demonstrating the system switching from Decimal to Hexadecimal, and showing Reset behavior.
2. The .sdc and .qsf files for their project.
3. A Design Record which is a PDF document that contains the following screenshots. Please refer to the Lab 1 Design Record Example document to see the required format and instructions on how to obtain the screenshots.
    a. Your version of the RTL schematic as shown in *Figure 2* (after the required changes have been completed, e.g. adding the Multiplexer).
    b. Your Slow 1200mV 85C Model Fmax Summary – we want to see the maximum frequency of your design.
    c. Your Messages Window.
    d. ModelSim simulation waveforms for MUX2TO1 and for top_level.
    e. Note, you do not have to show the useful instructions on how to obtain the screenshots, as shown in the example document. You just have to show the screenshots, the document title, and the student names.
4. Three videos of no more than 2 minutes each, delivered by the respective leads. All students in a team must present at least one video. You must feature your face in the video, as you are speaking. The videos must be recorded in "one-take." No video editing, splicing, cutting, speeding up or slowing down, etc. The videos must be uploaded to the D2L Dropbox in MP4 format.
    a. <u>Design Lead Video:</u>
        i. Use Quartus and your face must be present throughout the presentation. You can start a Zoom session and share the screen with yourself and this should make your face visible in a small picture.

   ii. Introduce yourself and your role.
   iii. Using Quartus: Explain the design and how it works, starting with an overview of the RTL schematic and dive into the design modules as needed, especially how the Multiplexer functions within the design. Then explain the RTL code, starting with top_level and explain the lower level modules as needed.

b. Simulation Lead Video:
   i. Use ModelSim and your face must be present throughout the presentation. You can start a Zoom session and share the screen with yourself and this should make your face visible in a small picture.
   ii. Introduce yourself and your role.
   iii. Using ModelSim: Explain the testbench code, starting with tb_top_level and show and explain its simulation waveforms in ModelSim. Then repeat this explanation for tb_MUX2TO1. Be sure to explain how the testbench exercises the design and how the simulation displays the correct functionality of the design (this is the purpose of the testbench).

c. Implementation Lead Video:
   i. Introduce yourself and your role, while looking into the camera, so that your face is visible.
   ii. Focus the camera on your DE10-Lite board and demonstrate the required functionality of the system, while explaining what you are doing and the results you are seeing and how it confirms the required behavior.
      1. Demonstrate Reset behavior.
      2. Demonstrate switching the slide switches and obtaining the correct Decimal values on the 7-segment displays. Also demonstrate the slide switch LEDs turning on and off to match the slide switch positions.
      3. Demonstrate switching to Hexadecimal mode and repeating the previous demonstration, but for Hexadecimal.

7. Grading Rubric

Lab 1 will be weighted at 3% of the final course grade. The grades of the 3 presentations will be averaged with equal weighting for all the students in the team. Example:

Design Lead Video:	A (4.0)
Simulation Lead Video:	B+ (3.3)
Implementation Lead Video:	B+ (3.3)
Grade assigned to all students of the team:	3.53 (10.6/3 = 3.5333… rounded to two decimal places)

The Design Record will be used primarily as a reference but a poor-quality Design Record will detract from the presentation grades.

7.1. Presentation Rubric

The video presentations will be assessed on a grade-point scale for a single grade out of 4.00. The grade will interpreted from the description below.

Grade	Description
4.00 (A)	Excellent: superior performance, showing comprehensive understanding of subject matter. *To earn an "A" the submitted work must fully complete and fully meets the project requirements. The work is of such high quality that it could serve as an exemplar for the project from both the technical and communication perspectives.*
3.70 (A-)	Excellent: superior performance, showing comprehensive understanding of subject matter. *The submitted work is fully complete and meets the project requirements. The work is very high quality from both the technical and communication perspectives.*
2.70 to 3.30 (B- to B+)	Good: clearly above average performance with knowledge of subject matter generally complete. *The submitted work is fully complete and meets the project requirements. The work is very high quality but has minor weakness or weaknesses either technically and/or in communications.*
1.70 to 2.30 (C- to C+)	Satisfactory: basic understanding of the subject matter. *Submitted work is complete but has significant weakness or weaknesses either technically and/or in communications. Generally indicates insufficient effort or accomplishment is marginal.*
1.00 to 1.30 (D to D+)	Minimal pass: marginal performance; generally insufficient preparation for subsequent courses in the same subject. *Weak effort demonstrated by missing elements or superficially completed elements either technically and/or in communications.*
0.00 (F)	Fail: unsatisfactory performance or failure to meet course requirements. *Clear lack of effort or ability to accomplish the project requirements.*

Notes: Grade can be reduced down to 0.0 for not complying with requirements or for unprofessional behavior. Excessive video lengths will be penalized one letter grade (e.g. B+ to B) for every 15 seconds overlength.

Late penalty: one letter grade per day late (e.g. A- to B+, not A- to B-), according to the D2L Dropbox timestamp. Students are responsible for retaining the D2L Dropbox submission confirmation email. Submissions must be fully complete to be eligible for grading of the lab project. This includes code and design files, videos, and Design Record, otherwise the late penalty will apply to the entire lab project.