Department of Electrical and Computer Engineering

ENEL 453: Digital System Design
Fall 2021

Lab 2: FPGA Voltmeter
Prelab and Instruction Manual

1 Overview

In this lab project you will be given a completed and fully functioning design project that implements an FPGA-based digital voltmeter. As prerequisite knowledge, you need to have learned how to operate the basic tools from Lab 1: Quartus, ModelSim, DE10-Lite, and the basics of VHDL (from VHDL lecture as well).

You will study the provided design to help you become more familiar with VHDL. You will then complete some additional work by completing two testbenches and by adding a multiplexor to the design. For interested students, a bonus project is offered, worth 2% of your term grade.

The FPGA on the DE10-Lite board includes an Analog-to-Digital Converter (ADC), which is built-in to the FPGA. This ADC is available to the designer as an IP-block, where IP refers to Intellectual Property. This ADC IP-block has a Verilog wrapper which allows the designer to incorporate it into their design. Thus, your design in VHDL will be a mixed-language design because it also includes Verilog.

The purpose of this lab project is to develop skills in VHDL, the DE10-Lite board, and Quartus, so that you can successfully complete the design projects of Labs 3 and 4. For Lab 3, you will further develop the FPGA-based digital voltmeter so that it can measure distance with an IR sensor.

This lab project is worth 10% of your term grade and broken up as:

- Pre-lab: 2 marks.
- In-lab demonstration in your lab period: 8 marks.
- Bonus: 2 marks.
- If you are unable to complete your demonstration by the end of your assigned lab period, then you may demonstrate at the beginning of your next lab period, for a loss of 2 marks.

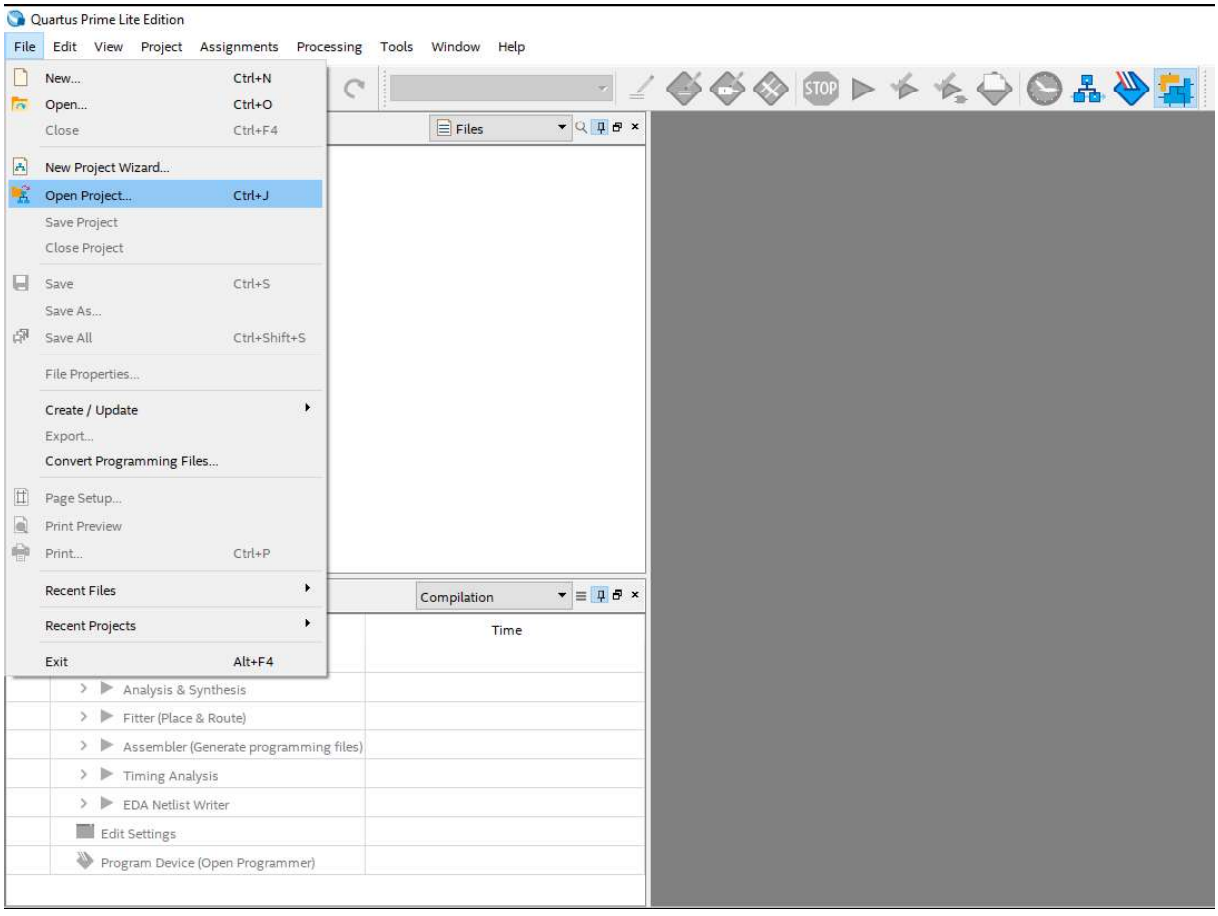
The **Pre-lab assignment** (Section 3, starting on page 8 of this document) must be completed **for each team** prior to the scheduled lab period and uploaded to the ENEL 453 D2L Dropbox by 1:59 pm on Thursday October 21 for section B02 and October 28 for Section B01. Please submit in advance of the deadline.

At the end of your demonstration, your VHDL files, constraints files, and Design Record document are required to be uploaded by **each team** within 24 hours of your demonstration, in order to receive your grades and credit for this lab. Write your name and your partner's name on the first page of your document

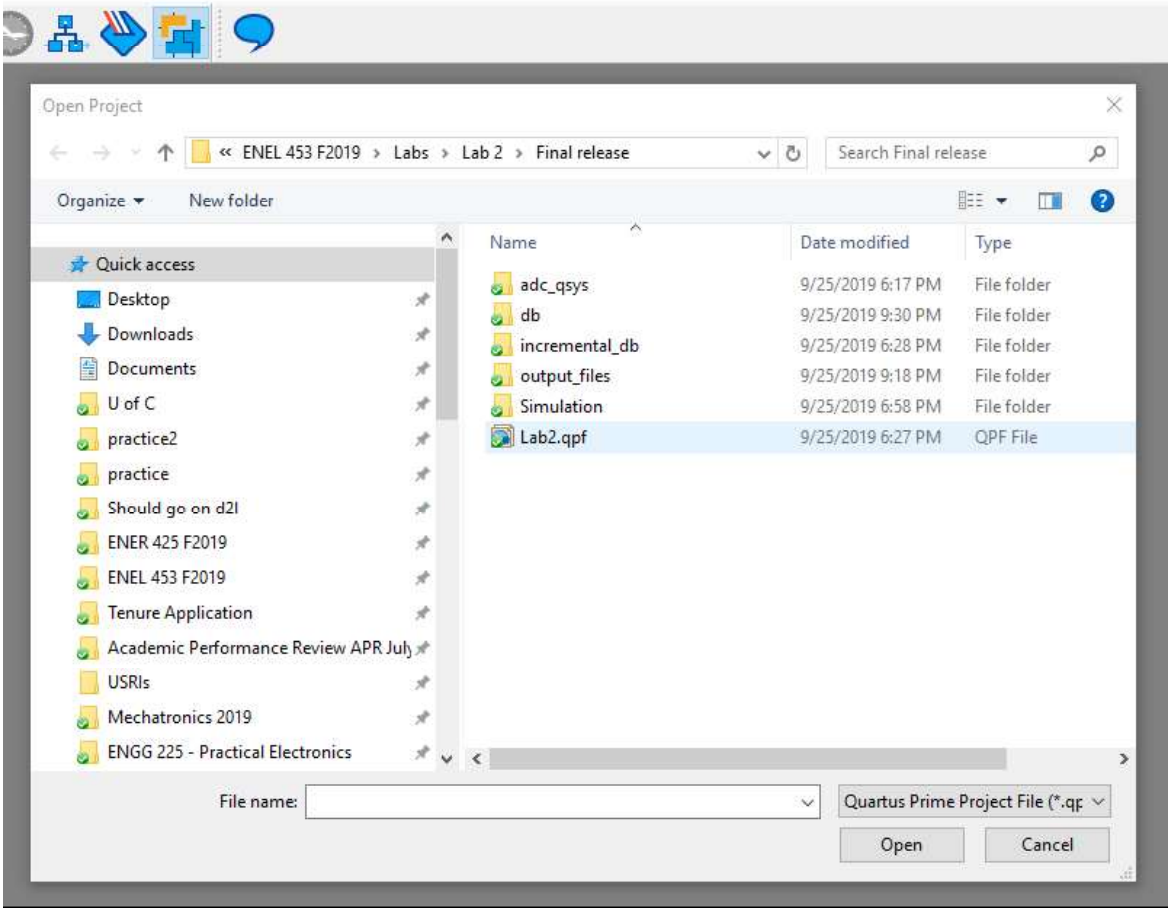
2 Project

2.1 Getting the Project

Download the project from D2L. This is a working project, so you just need to open the project file from Quartus.

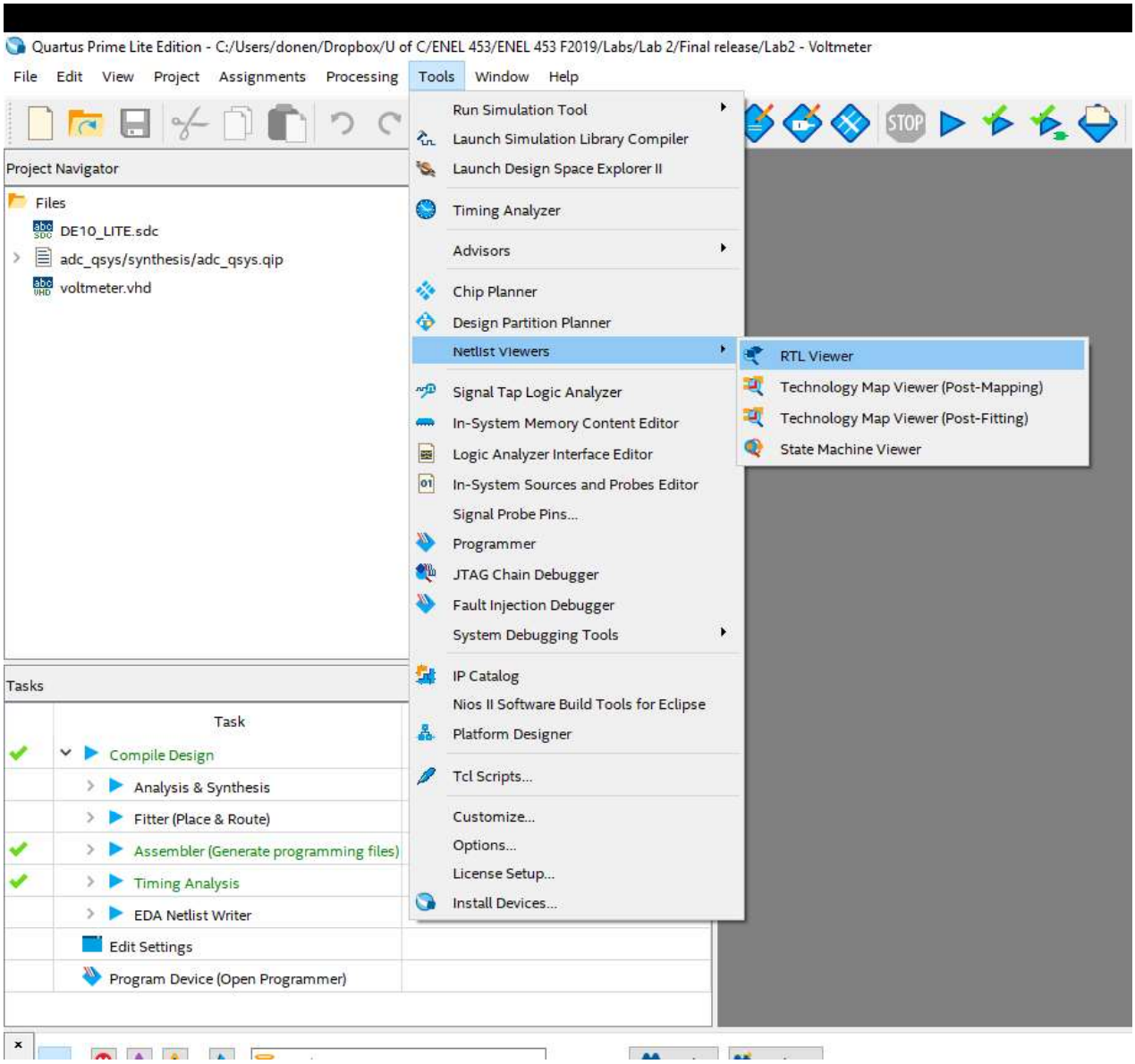


Navigate to the project file, Lab2.qpf, and select it:

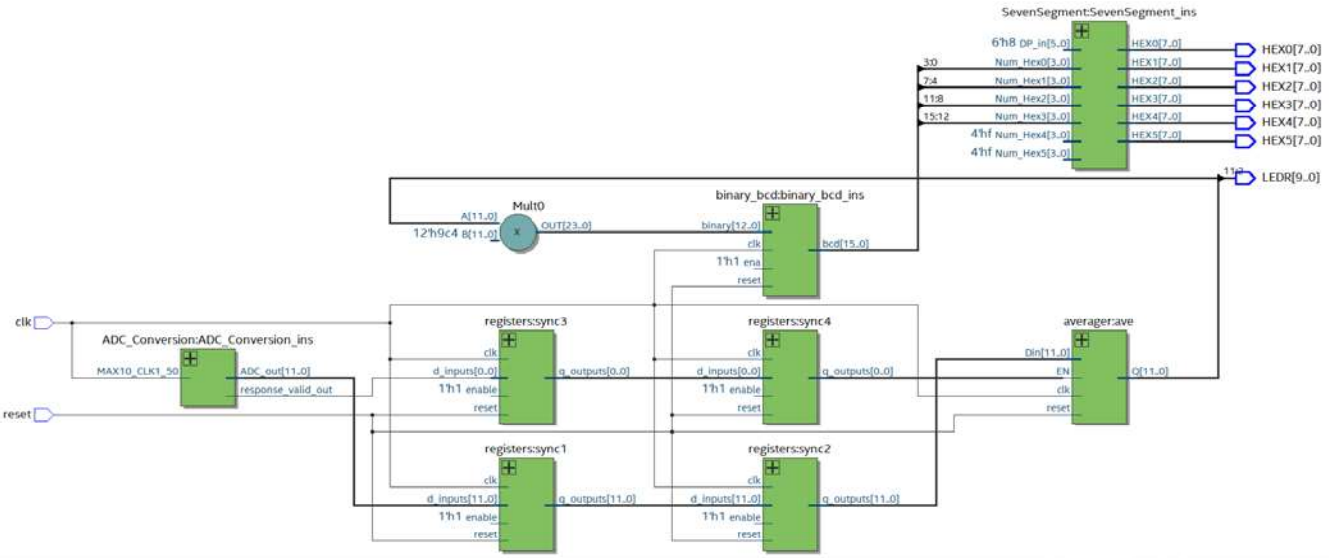


2.2 Getting an Overview of the Project

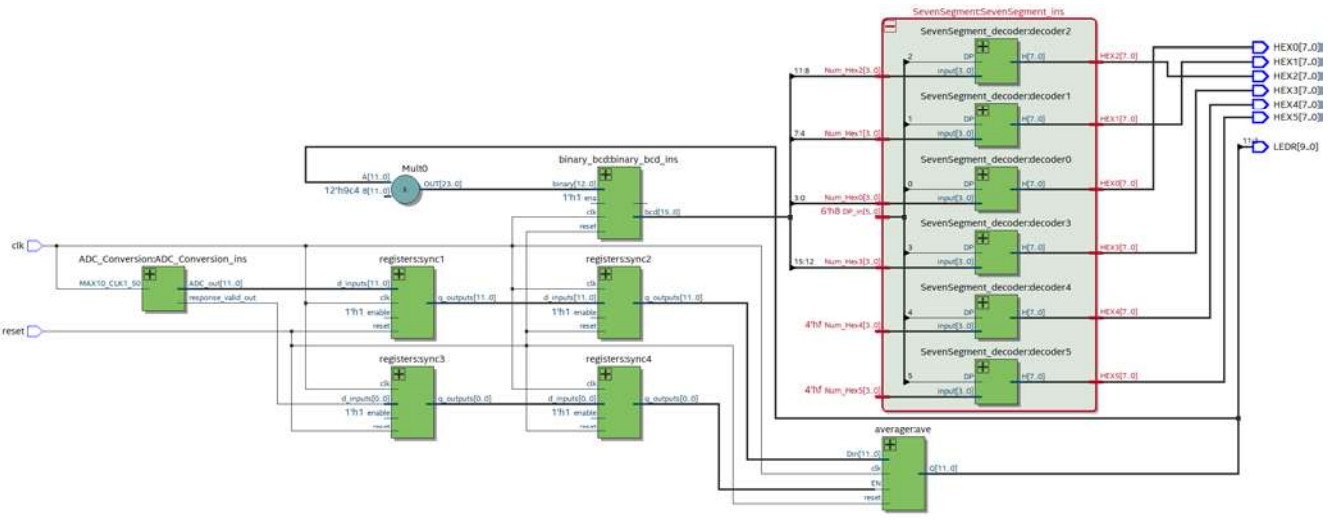
The project in D2L is already compiled, but depending on what you are doing (e.g. you've changed some files), you may need to compile it again. When the project has been compiled, open the RTL Viewer to begin to study the design.



You will see the schematic representation of the design as shown below. This is an exceptionally useful tool to understand your design and you are encouraged to check the RTL Schematic when you are designing, to get a visual understanding of your code and its interconnections.



You can drill into your blocks in the RTL Schematic, so that you can see what's inside, e.g. let's look inside the 7-Segment Decoder.



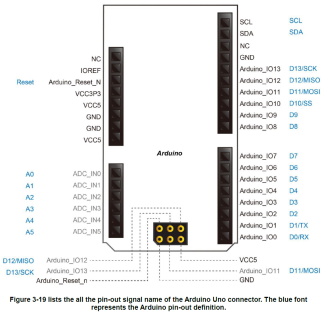
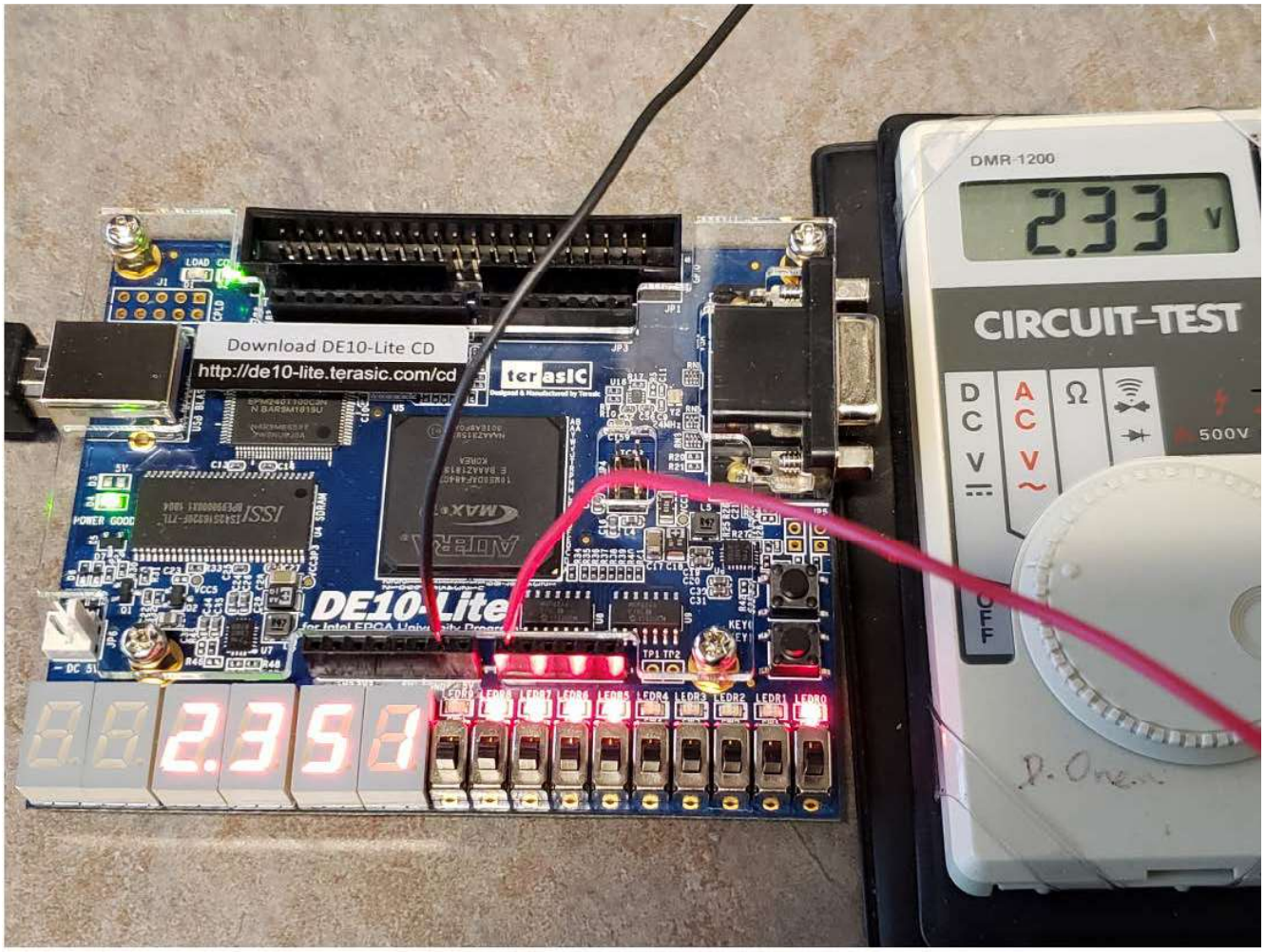


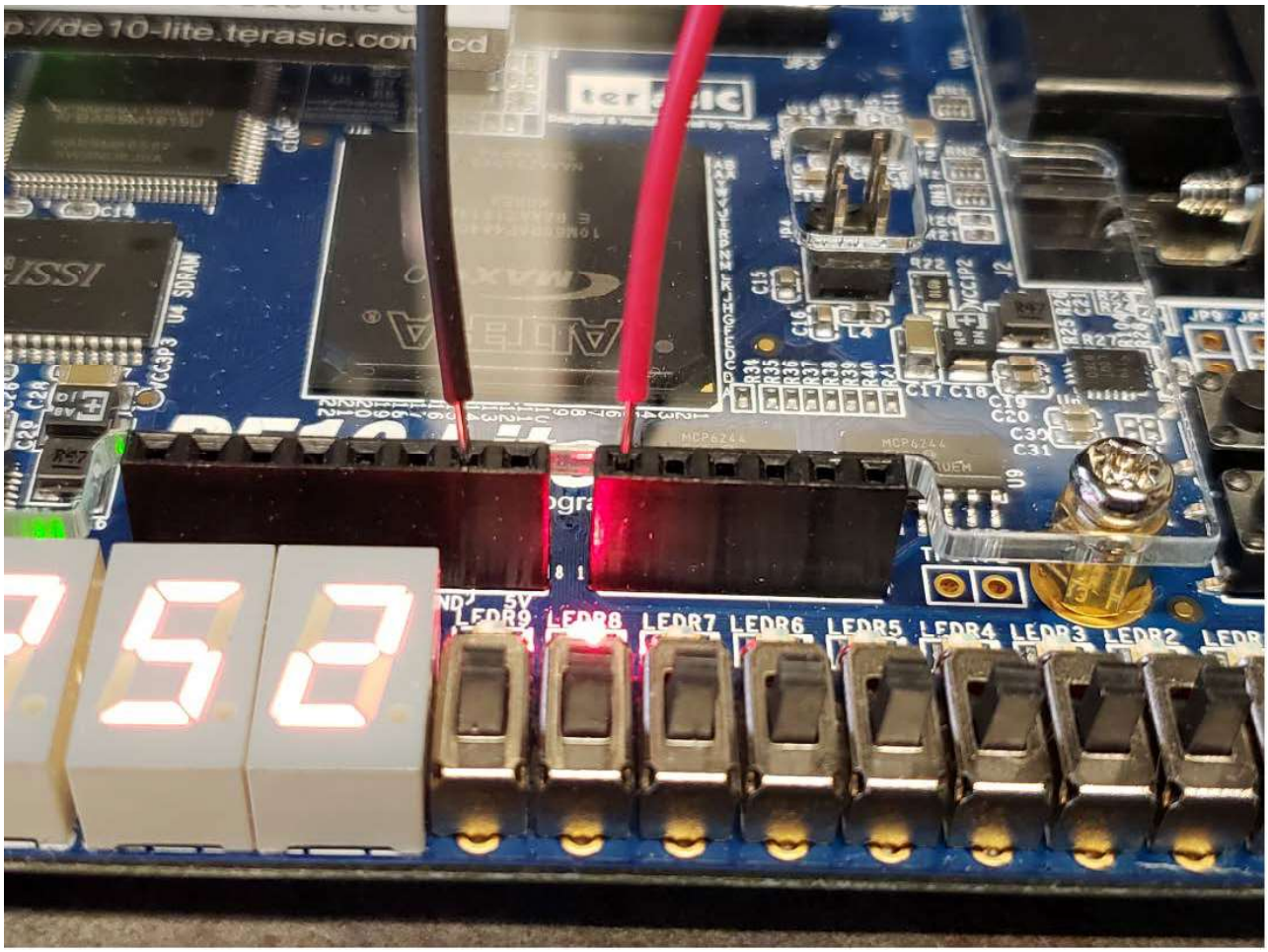
Figure 3-19 lists the all the pin-out signal name of the Arduino Uno connector. The blue font represents the Arduino pin-out definition.

2.3 Testing the Project

Download the design to your DE10-Lite board. Determine how you will setup up an experiment to physically test the FPGA voltmeter. The FPGA voltmeter accepts a voltage of **0V DC to 5 V DC,** do not exceed this range or you may damage the DE10-Lite board. You will need to:

1. Provide a voltage at 0 V DC, 3.3 V DC, and 5 V DC. You can access these voltages on the Arduino header and explained on the DE10-lite manual.
2. Connect the ground of the voltage output to pin 7 of JP7 of the Arduino header on the DE10-Lite board; and connect the positive of the voltage source to pin 1 of JP8 of the Arduino header, as shown below and in more detail on the next page (black wire is ground, red is positive).
3. Connect some sort of voltmeter to confirm what is the actual voltage going to the FPGA voltmeter input.





2.4 Studying the Project

Experiment with the board and begin to study the VHDL code, using the RTL schematic as a guide. From studying the VHDL code, learn how the design is partitioned and connected in hierarchy. Learn to be able to read the code and interpret what kind of hardware it describes.

2.5 Make a Change to the Project – Add a Multiplexor

Now that you have some familiarity with the design, you will notice that there is an **averager** module in the design. This module averages the values over 16 samples. What you need to do is test whether the averager makes a difference to the performance of the FPGA voltmeter.

1. **You will design a multiplexer to include in the design, to switch either the raw input of the averager or the output of the averager, to be displayed on the 7-segment displays. The multiplexer will be controlled by a slide switch on the board, i.e one of the ones above in the picture.**

2. **You must also create a VHDL testbench for your multiplexer.**

3. **You will modify your pre-lab's top-level testbench to also demonstrate the multiplexor functionality.**

2.6 Bonus Make Another Change to the Project – Enhance the Averager

For a bonus of 2 marks, design an alternative to the averager. The requirements are:

1. The new averager should essentially be a "drop-in" replacement to the original. The design should not be altered in any other significant way.

2. The averager must be parametrizable, meaning that **generic statements** will be used to create an arbitrary number of sample for averaging, e.g. 4, 16, 64, 128. Note the original averager takes the average of 16 inputs.

3. The new averager must have a testbench.

4. If you adapt code from other sources (e.g. internet), you must cite it in your VHDL file and disclose it during your demonstration. You are not permitted to share code with other students in the class. You must demonstrate clear understanding of the code you are using in order to receive credit.

3. Pre-lab Assignment

The Pre-lab assignment shall include:

1. (1 mark) A VHDL testbench for the top-level of the design. FYI the top-level entity in the design is called **Voltmeter**. The testbench should have appropriate stimulus for the **clk** and **reset** inputs (these are the only inputs). Note, this testbench is for the initial version of the Voltmeter design, as provided to you, not the enhanced version with the multiplexor.
2. (1 mark) A PDF of the simulation waveform from the above testbench. This waveform should show the design being successfully exercised, and you may wish to consider showing several screenshots of the waveform, at different time scales. You do not have to write anything on the PDF, the waveforms should be self-explanatory.

Hints on how you will complete the pre-lab assignment:

1. You will use ModelSim to create the simulation. Please refer to Lab 1 to recall how to use it.
2. You will use the VHDL design files provided to you. However, for the ADC's Verilog file, we have created a VHDL simulation file called "test_ADC.vhd" and its entity is "test_DE10_Lite". We created the new simulation file because the ADC's Verilog file will not simulate. Therefore, you will replace the Verilog file with its VHDL simulation file. The basic procedure is as follows:
    a. Create a ModelSim project and add the Voltmeter VHDL files to it.
    b. Be sure to also add "test_ADC.vhd".

c. Edit the top-level file (voltmeter.vhd) so that the "test_DE10_Lite" component is called instead of "ADC_Conversion". You'll have to change this in two places: the *component declaration*, and the *instantiation*. Note, you should only have to change the name of the entity, not the signals, because they are identical in both the VHDL and Verilog versions. You may also alternatively edit the voltmeter.vhd file separately in another editor and bring it into ModelSim.
d. Compile the files in ModelSim.
e. Start the simulation and add the appropriate signals to the waveform.
f. Run the simulation and capture snapshots of the waveform.
g. When you go back to Quartus to synthesize and download your design, be sure to restore the ADC *component declaration* and *instantiation* back to the original, within voltmeter.vhd (i.e. reverse what you did in step **c**.).

4. Lab Demo and Design Record Document

Perform a demonstration of the working FPGA voltmeter during your lab period, and explain your design.
Within 24 hours of completing the demonstration, upload to D2L the following:

1. Your VHDL design files for the complete voltmeter design with the multiplexor.
2. Your constraints files (.sdc and .qsf).
3. Your top-level VHDL testbench, and your multiplexor VHDL testbench.
4. Your Design Record document with the following:
   4.1 Your name and your partner's name on the first page of your document.
   4.2 A screenshot of your top-level RTL schematic.
   4.3 Screenshots of the simulation waveforms showing the correct operation of the top-level design of the voltmeter, with the multiplexor incorporated. This means you should show the behavior of the voltmeter design with the switch control for the multiplexor switching from raw to averaged readings.
   4.4 Note that no write-up is required.

5. Grading

Your Lab 2 project demo will be assessed in the last hour of the lab period, your TA will inform you of your timeslot.

You must show the following to the TA to obtain your credit for the lab, and the TA may request further demonstration than listed below.

All team members must be able to answer questions about any part of the lab project in order to receive credit for the project.

Description	Marks
Pre-lab assignment	2
In-lab demonstration: • Accurate reporting of the voltage readings on the 7-segment displays. • Reset behavior. • Multiplexor behavior. The above functionality must be demonstrated to receive credit for completing the lab. Students must be able to explain their project to receive credit for completing the lab. Late demonstrations can be held at the beginning of your next lab period, for a reduction of 2 marks. B02 demonstration: Thursday Oct. 21 B01 demonstration: Thursday Oct. 28	8
To receive the above marks and credit for the lab, **all the files specified in Section 4 must be uploaded within 24 hours after you demonstrate**. Do not worry about special formatting or documenting the screenshots, just paste them into a Word document, PDF it, and upload the PDF.	