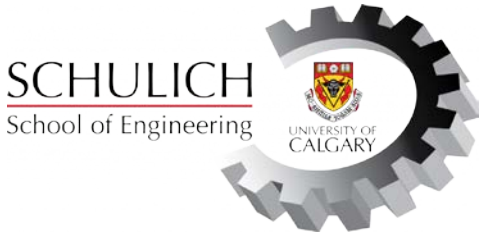


Department of Electrical and Computer Engineering

ENEL 453: Digital System Design
Fall 2021

Lab 3: FPGA Distance Measurer
Prelab and Instruction Manual

1 Overview

In this challenging design project, you will convert your FPGA-based voltmeter to a distance measurer. You will be provided a distance measurer module on a PCB. The module has a sensor, and the distance that is measured is the distance between the sensor and an object such as a paper sheet or another object placed in front of the sensor. The range of that distance is between 4 and 32 cm. This module requires a 5 VDC supply voltage and outputs an analog voltage output that is related to the distance being measured. You will interface your FPGA's ADC input to the distance measurer module and you will convert the voltage measurement to a distance measurement. You will also switch between the distance measurement and the voltage measurement.

This lab project is worth 35% of your term grade and broken up as:

- Pre-lab: 7 marks.
- In-lab demonstration in your lab period: 28 marks.
- Bonus: 10 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 8 marks.
- Note that in the demonstration, you must explain your design and be able to answer questions about it, in order to receive credit for this lab project.

The **Pre-lab assignment** (page 10) must be completed **per team** prior to the scheduled lab period and uploaded **per team** to the ENEL 453 D2L Dropbox by 1:59 pm on November 4 for B01 and November 18 for B02. The D2L timestamp will be taken as the submission time and late submissions will not be graded. 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 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 Project Requirements

Your design must feature the following:

1. Display the distance from 4.00 cm to 32.99 cm on the 7-segment displays, to two decimal places. E.g. 16.35 cm or 13.10 cm or 32.97 cm.

2. A VHDL array-based conversion of the voltage to distance. This array must use the full 12-bits of the voltage representation, and therefore, must have $2^{12} = 4096$ rows. VHDL arrays were discussed in lecture and another way to interpret the required array is as a look-up table. More detail is given on page 7 of this document.

3. The leading zero of the distance display will be blanked. E.g. 4.35 cm not 04.35 cm.

4. If the measured distance is 33.00 cm or above, then the display will show some sort of error message of your own design.

5. Be able to switch between displaying voltage (i.e. voltmeter functionality of Lab 2) or displaying the distance, controlled by a slide switch.

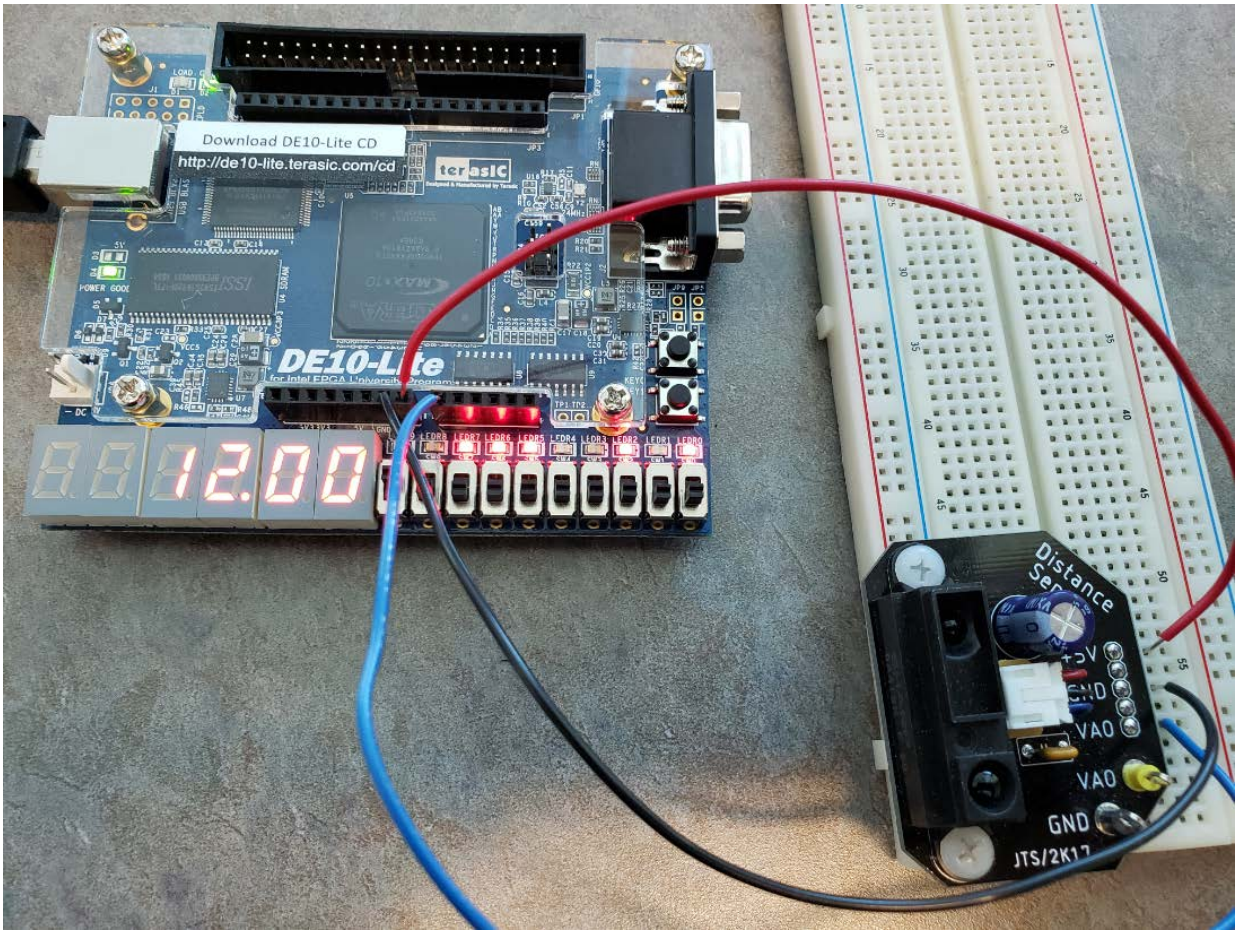
6. The design must meet a minimum timing of 50 MHz for its clock.

7. 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 nor use any other student's code. You must demonstrate clear understanding of the code you are using in order to receive credit.

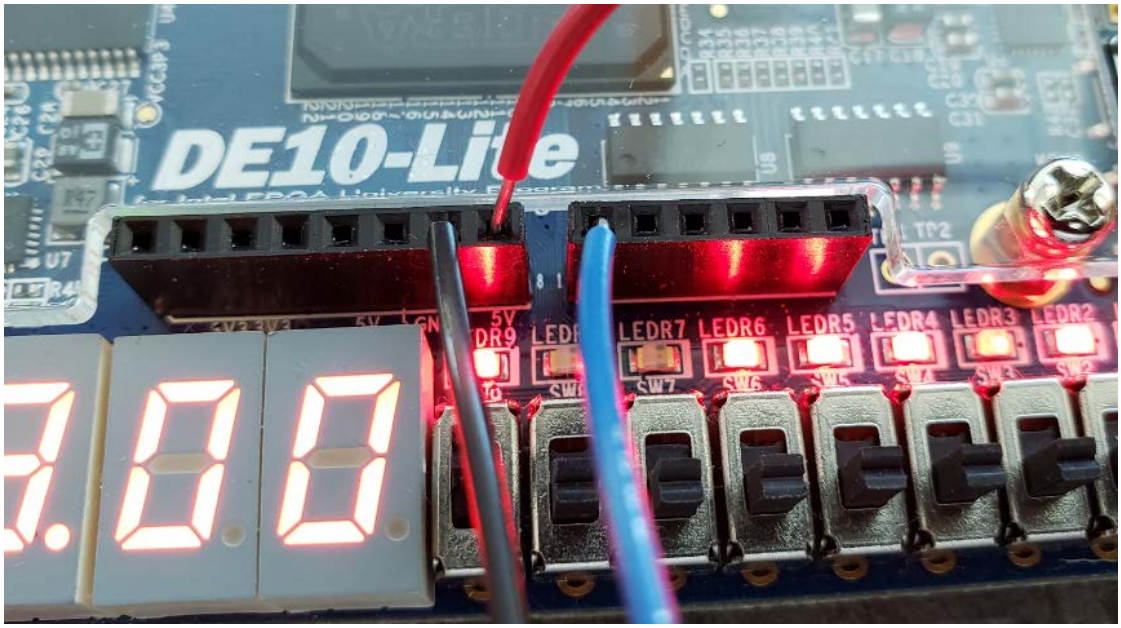
2.2 Connecting the Distance Sensor to the DE10-Lite

You will interface a distance measurer circuit module to your FPGA voltmeter as shown in the pictures below. The module has 3 connections to the Arduino header on the DE10-Lite board:

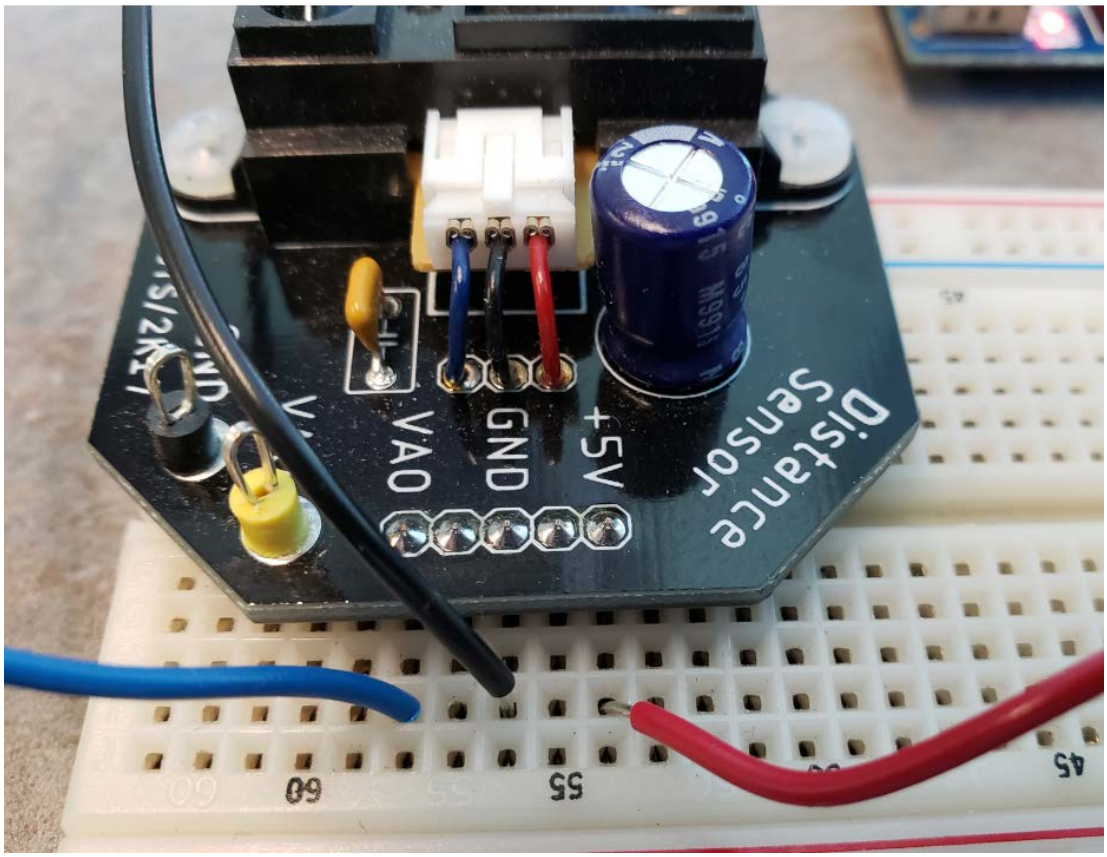
- 5 VDC supply input, labelled as **+5V** (red wire in the picture)
- Ground reference, labelled as **GND** (black wire in the picture)
- Analog voltage output, labelled as **VA0** (blue wire in the picture). This analog voltage is related to the distance measured by the sensor, as described in the next section.



Below is a close-up picture of the connections to the Arduino header on the DE10-Lite board. These are the same connections as Lab 2, except with the addition of the red 5 VDC wire.

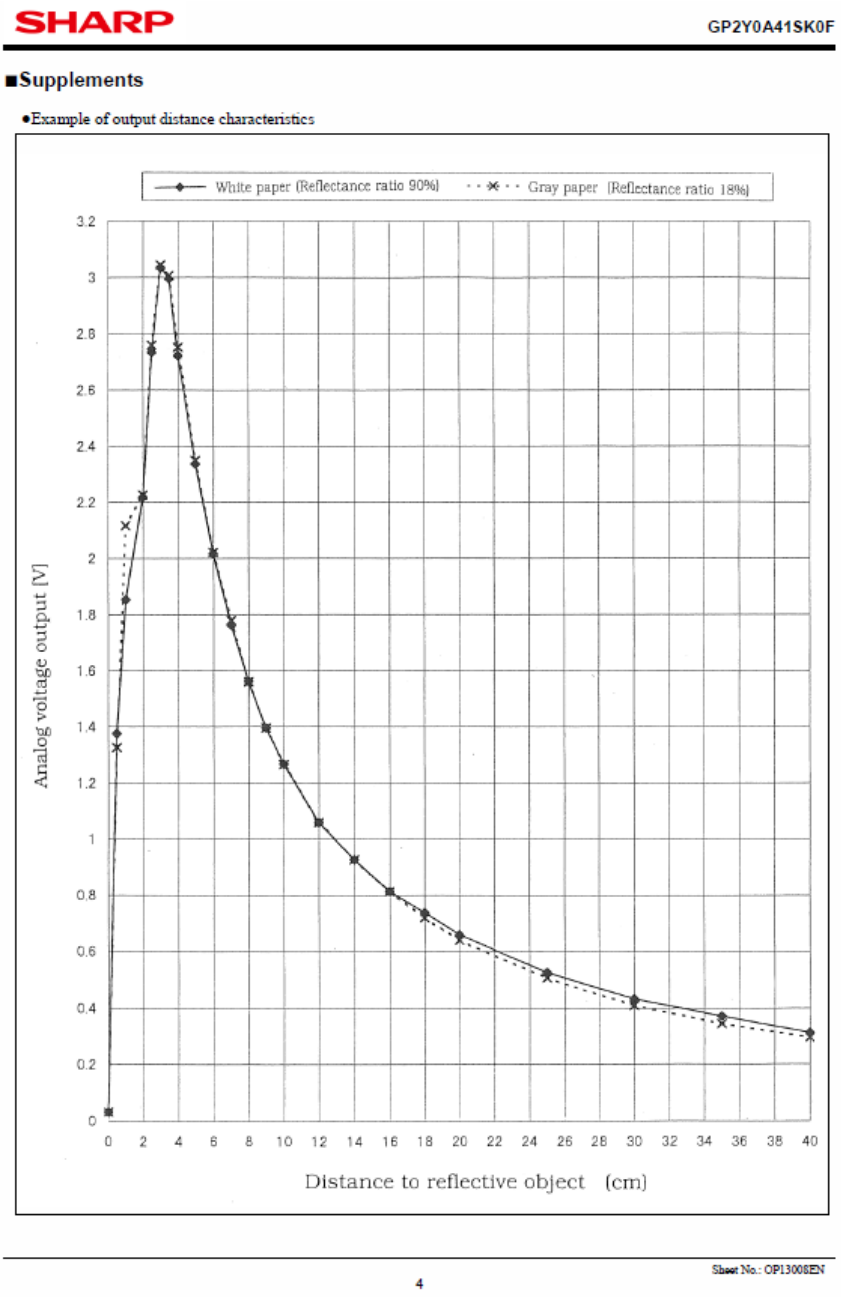


Below is a close-up picture of the connections to the distance measurer circuit module.



2.3 Sharp GP2Y0A41SK0F Distance Sensor Data Sheet

The distance measurer circuit module schematic diagram and PCB layout are provided in D2L. The module uses the Sharp GP2Y0A41SK0F sensor and its data sheet is also provided in D2L. You should review the data sheet and page 4 is the most relevant, as it shows the relationship of the analog voltage output of the sensor, to the measured distance, as shown below.



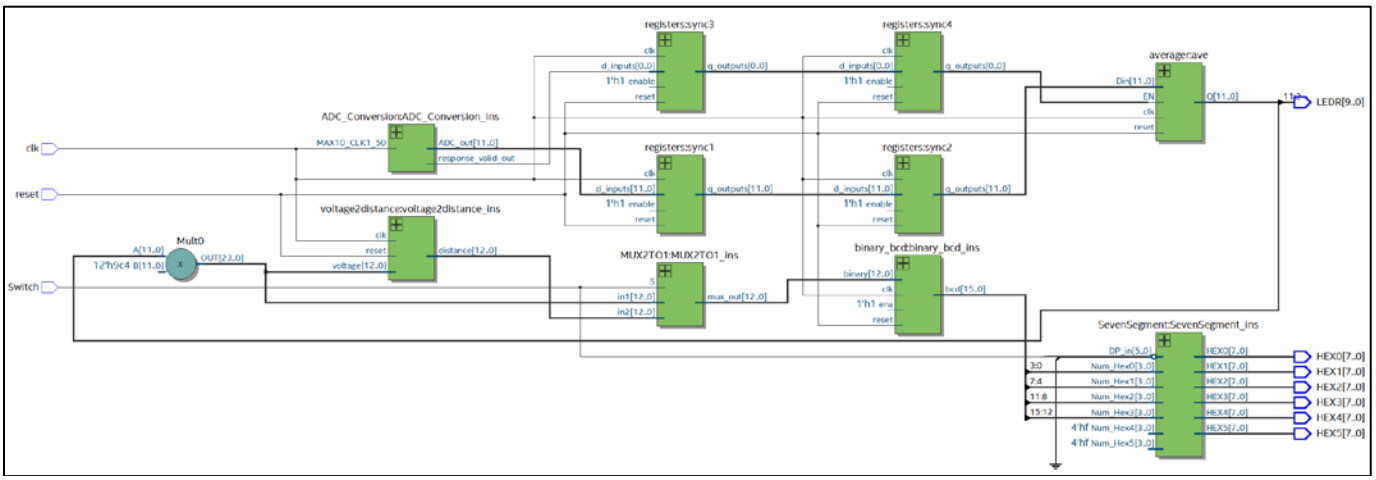
You will note that the voltage-distance relationship is non-linear, so conversion is not straightforward. You will devise a way to take the 12-bit voltage reading from your ADC (i.e. Lab 2 voltmeter project) and convert it to a distance and display this distance on the 7-segment displays. When performing your mathematics, use the graph on page 4 of the data sheet as it will be more clear than the above picture.

2.4 Converting Voltage to Distance

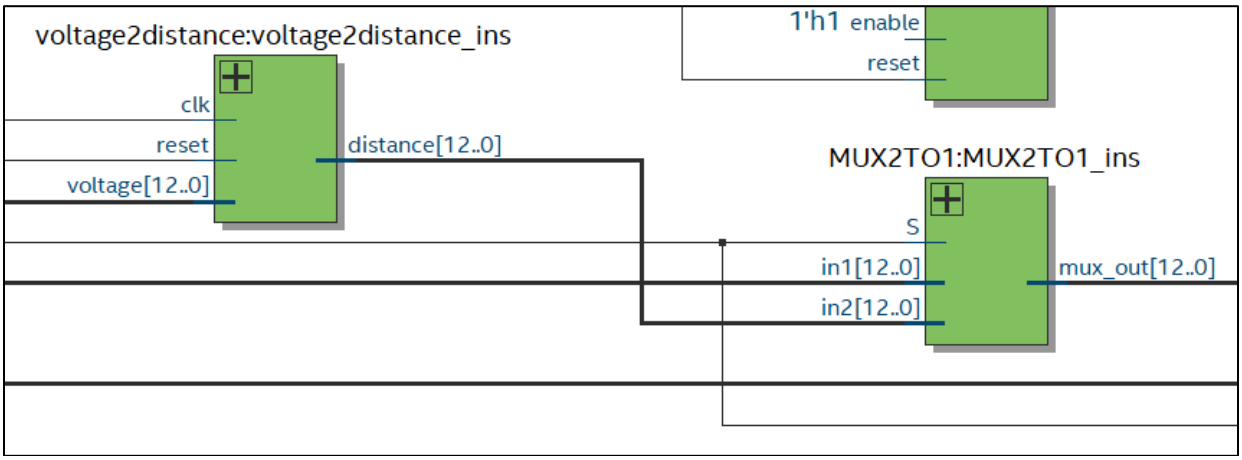
You are required to use arrays to implement the module that converts the voltage reading to distance. To help you get started, this section describes a reference design that partially meets the requirement.

Below is top-level RTL schematic of a complete system for Lab 3, that meets only the simple requirements of converting voltage to distance. Two modules have been added, **voltage2distance** and **MUX2TO1**. An external connection to a slide switch has been added (**Switch** input), to be able to switch between distance and voltage display on the 7-segment displays.

You do not necessarily need to follow this architecture in your design project and you will most likely need to modify the architecture anyways, to fulfill the complete design requirements.



Below is a close-up view of the two additional modules.



Below is the complete VHDL code for the **voltage2distance** module, which is provided to you as a starting point to help you understand the conversion process. This is a very crude design and DOES NOT feature the array-based approach required for Lab 3.

```
1   LIBRARY ieee;
2   USE ieee.std_logic_1164.all;
3   USE ieee.numeric_std.all;
4
5   ENTITY voltage2distance IS
6      PORT(
7         clk            :  IN    STD_LOGIC;
8         reset          :  IN    STD_LOGIC;
9         voltage        :  IN    STD_LOGIC_VECTOR(12 DOWNTO 0);
10        distance       :  OUT   STD_LOGIC_VECTOR(12 DOWNTO 0));
11  END voltage2distance;
12
13  ARCHITECTURE behavior OF voltage2distance IS
14
15  begin
16
17  bcd_process: process (reset, clk)
18  begin
19        if (reset='1') then
20           distance<=(others=>'0');
21        elsif (rising_edge(clk)) then
22           if    (unsigned(voltage)>2900) then distance<=STD_LOGIC_VECTOR(to_unsigned(300,distance'length));
23           elsif (unsigned(voltage)>2750) then distance<=STD_LOGIC_VECTOR(to_unsigned(400,distance'length));
24           elsif (unsigned(voltage)>2350) then distance<=STD_LOGIC_VECTOR(to_unsigned(500,distance'length));
25           elsif (unsigned(voltage)>2100) then distance<=STD_LOGIC_VECTOR(to_unsigned(600,distance'length));
26           elsif (unsigned(voltage)>1780) then distance<=STD_LOGIC_VECTOR(to_unsigned(700,distance'length));
27           elsif (unsigned(voltage)>1580) then distance<=STD_LOGIC_VECTOR(to_unsigned(800,distance'length));
28           elsif (unsigned(voltage)>1400) then distance<=STD_LOGIC_VECTOR(to_unsigned(900,distance'length));
29           elsif (unsigned(voltage)>1290) then distance<=STD_LOGIC_VECTOR(to_unsigned(1000,distance'length));
30           elsif (unsigned(voltage)>1080) then distance<=STD_LOGIC_VECTOR(to_unsigned(1200,distance'length));
31           elsif (unsigned(voltage)> 920) then distance<=STD_LOGIC_VECTOR(to_unsigned(1400,distance'length));
32           elsif (unsigned(voltage)> 810) then distance<=STD_LOGIC_VECTOR(to_unsigned(1600,distance'length));
33           elsif (unsigned(voltage)> 720) then distance<=STD_LOGIC_VECTOR(to_unsigned(1800,distance'length));
34           elsif (unsigned(voltage)> 660) then distance<=STD_LOGIC_VECTOR(to_unsigned(2000,distance'length));
35           elsif (unsigned(voltage)> 550) then distance<=STD_LOGIC_VECTOR(to_unsigned(2500,distance'length));
36           elsif (unsigned(voltage)> 410) then distance<=STD_LOGIC_VECTOR(to_unsigned(3000,distance'length));
37           elsif (unsigned(voltage)> 380) then distance<=STD_LOGIC_VECTOR(to_unsigned(3500,distance'length));
38           elsif (unsigned(voltage)> 300) then distance<=STD_LOGIC_VECTOR(to_unsigned(4000,distance'length));
39           else distance<=STD_LOGIC_VECTOR(to_unsigned(5000,distance'length));
40           end if;
41        end if;
42  end process;
43
44
45  end behavior;
```

Let us examine line 22:

```
22          if    (unsigned(voltage)>2900) then distance<=STD_LOGIC_VECTOR(to_unsigned(300,distance'length));
```

The “if” statement checks whether the value of the voltage exceeds 2.9V (or more explicitly, 2900 mV). To be able to check against the decimal value of 2900, the type conversion “unsigned” is used to convert **voltage** from the type *std_logic_vector* to the type *unsigned*.

If the above condition is satisfied, then **distance** takes the value of *300*, with the appropriate type conversions. In this case, 300 represents 3.00 cm, meaning that the decimal point must be placed in the correct location in the 7-segment displays.

By reading and understanding the remainder of the above VHDL code, you will notice that the conversion is very coarse, with only 1 cm resolution in the distance range from 3 cm to 10 cm (lines 22 to 29), and becoming even more coarse with 5 cm resolution in the distance range 20 cm to 40 cm (lines 34 to 38). Note that the above design violates the Lab 3 design requirements in many ways (e.g. no error message after 33.00 cm, no 12-bit resolution).

Your implementation of this module must use the 12-bits of the ADC output and must be array-based. This means your array should have 2^{12} = 4096 rows. Do not have a 4096 chain of if/elsif’s, you must use an array!

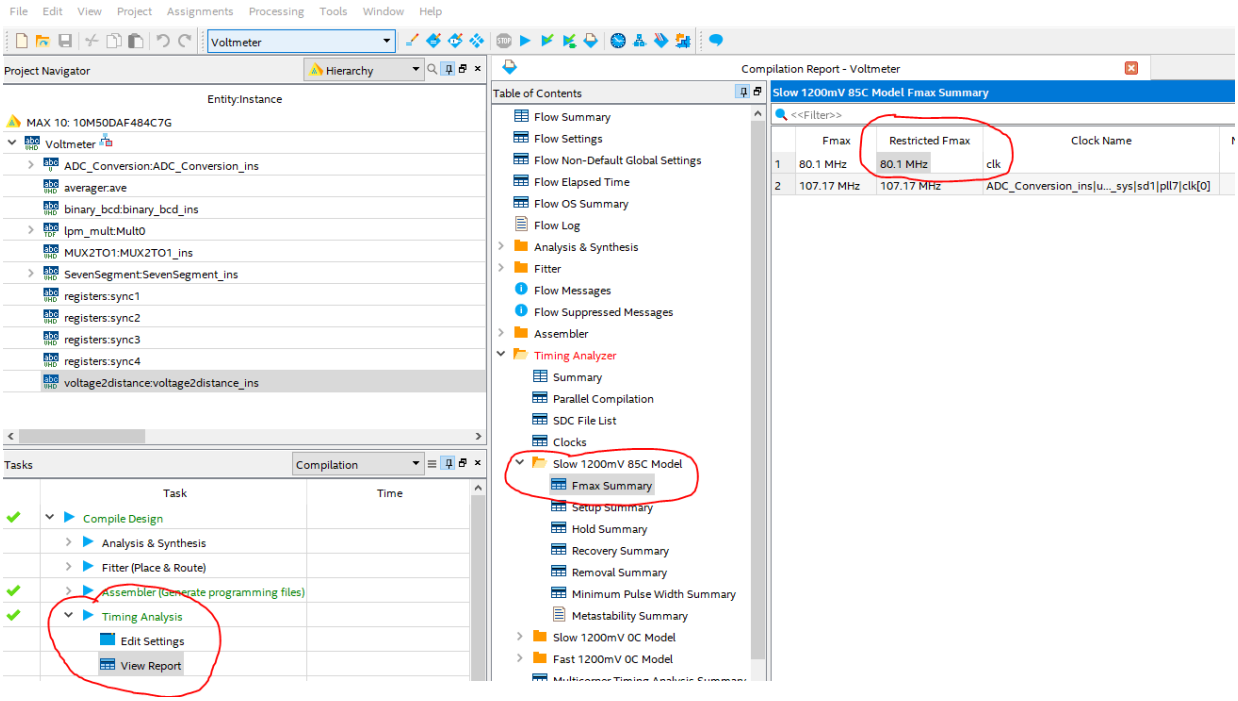
2.5 Determing the Maximum Clock Frequency of Your Design

Your design must not violate the 50 MHz clock frequency timing of the DE10-Lite board. Below is a screenshot of where to find the Quartus reported maximum clock frequency of your design. The red circles show the key information, and moving from the leftmost red circle to the rightmost:

1. Access the timing report from `Timing Analysis > View Report`

2. Select `Slow 1200mV 85C Model` > `Fmax Summary` (this is the worst-case clock)

3. Observe that the maximum report clock frequency for the signal **`clk`** is `80.1 MHz`, which is greater than the actual 50 MHz clock on our DE10-Lite board. This means that our design will not violate the timing requirements, and if designed with the correct functionality, will work reliably. However, if the reported `Fmax` was below 50 MHz, then our design may not work reliably, even though our functionality was correct.



You will include a screenshot similar to the above and paste it into your Design Record, to verify that your design does not violate timing.

2.6 Bonus: Make Another Change to the Project – Enhance the Measurer

A bonus of up to 10 marks is available only after the original project requirements (Section 2.1) have been met. The bonus mark requirements are:

1. (1.5 marks) Modify the design of the synchronizer to use the **generate** statement and the **generic** statement to instantiate a series of synchronization flip flops, so that you can specify a variable number of synchronization flip flops, ranging from 1 to 3. This means replacing the **registers.vhd** in the original design.

2. (3 mark) Increase the length of the averager so that it provides a very stable distance display (i.e. non-fluctuating) and yet can react in real-time to changes in distance (i.e. smooth, monotonic transitions in the distance display).

3. (5.5 marks) Incorporate another distance measurement mode to measure the distance between 0 cm and 4 cm. Use a slide switch to control switching to this short range mode. Both the original distance measurement mode and the voltmeter mode must also be available. You must be able to explain why you need to manually switch between the short range and long range modes.

4. The design must meet a minimum timing of 50 MHz for its clock in order to be eligible for bonus marks.

5. 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 nor use any other student's code. You must demonstrate clear understanding of the code you are using in order to receive credit.

3. Pre-lab Assignment

1. Convert sevensegment.vhd to use the **generate** statement to instantiate the 6 seven-segment decoders.

2. Rewrite the 1-process state machine in binary_bcd.vhd to be a 2-process state machine, with the state register being in the synchronous process and the next-state and output logic being in the combinational process.

3. There is a problem with the **reset** in registers.vhd. Identify the problem, fix it, and make a comment at the top of the file explain the change.

4. ~~There is a problem with the process **process_DP** in the top-level, voltmeter.vhd. Identify the problem, fix it, and make a comment just before the process to explain the change.~~

5. Test the changes with your top-level testbench from Lab 2, to verify that the operation is correct.

6. Upload your VHDL files for the above and a screenshot of your top-level testbench waveforms showing the correct operation of your circuit.

4. Lab Demo and Design Record Document

Perform a demonstration of the working FPGA distance measurer 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 design.
2. Your constraints files (.sdc and .qsf).
3. Your top-level VHDL testbench, and your 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 distance measurer.
   4.4 Screenshot of your Timing Report, showing the maximum clock frequency of your design.
   4.5 Note that no write-up is required.

5. Grading

Your Lab 3 project demo will be assessed in the last 2 hours 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	7
In-lab demonstration: • Accurate reporting of the voltage readings on the 7-segment displays, at various input voltages between 0 and 5 V DC. • Accurate reporting of the distance readings on the 7-segment displays, at various distances between 4.00 cm and 32.99 cm. • Switch between voltage and distance measurements with a slide switch. • Array-based voltage-to-distance conversion. • Blanking of the leading zeros in the distance measurement. • Error message at distances 33.00 cm or above. • Reset behavior. • Your design meets a minimum of 50 MHz timing (show the `Timing Report`). • Explain your design and answer questions about it. 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 8 marks. B01 demonstration: Nov. 4 B02 demonstration: Nov. 18	28
To receive the above marks and credit for the lab, **all the files specified in Section 4 must be uploaded to the Dropbox in D2L 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.	