Impedance Measurement Sensor

Aakash Kapoor 160070060

Swadha Sanghvi 16D070037

Devesh Kumar 16D070044

Spring 2019

Contents

1 Project Description

1.1 Background and Motivation

All of us are familiar with the dry cell, the humble piece of electrical technology that enabled so much of primitive technology and pushed our thinking forward. With time, it evolved and now we have batteries and charge storage devices in a wide spectrum of shapes, sizes, capacity, core design and functionality. However, all of them have similarities in their design and operation making them easily characterizable. Studies have shown that as the useful capacity of a battery reduces, its intrinsic impedance begins to rise. Here we see the first and most basic use of impedance measurement devices - Being able to take impedance measurements at different battery capacities and, with the help of earlier models, predict how long the batteries will need to be replenished This obviously isn't the only use of impedance measurement devices.

Bioelectrical Impedance Analysis is a method used to estimate body composition, especially body fat, by measuring the impedance of the human body. As soon as this fact was discovered, this simple piece of technology suddenly got an important use within the health industry. The basis for this relies on the water content affecting the impedance of a body mass. A slight modification of this technique can be used to measure the water uptake by a body and can be extremely useful for studying the behaviour of electrochemical systems. We also have studies detailing the electrical impedance of a tissue as a function of its structure and how this can be used to distinguish between normal and cancerous tissues.

1.2 Major Objectives

- Given an input circuit or sub-circuit (DUT), we aim to plot the magnitude and phase response within a certain range of frequency.
- Given a network with its impedance plot, we plan to estimate the values of the underlying passive elements.

1.3 Final Specifications

Based on the components used within our circuit, we have arrived at the following specifications -

- Typical detection limits : Magnitude of the impedance (encompassing both resistive and reactive terms) to be at least 500Ω.
- In case of a parallel R-C circuit, our circuit is able to perform within safe limits for values of resistance typically greater than 700Ω and capacitance lesser that 30 nF.
- However, these are typical values and not necessarily indicative of the best case performance of the circuit. At the best case, we can perform upto 33 percent better than typical limits.

2 System Level Overview

Shown bwlow is the overall block diagram of our proposed implementation for this project. It involves a few major components, each of which have been discussed at length in their respective sections. For demonstration purposes, we have used an Arduino board to capture the output data instead of the FPGA because we were unable to figure out how to read 4 pins one after the other (We could read two, but we needed 4 and so it was safer to just show on the Arduino board which was working as we desired it to).

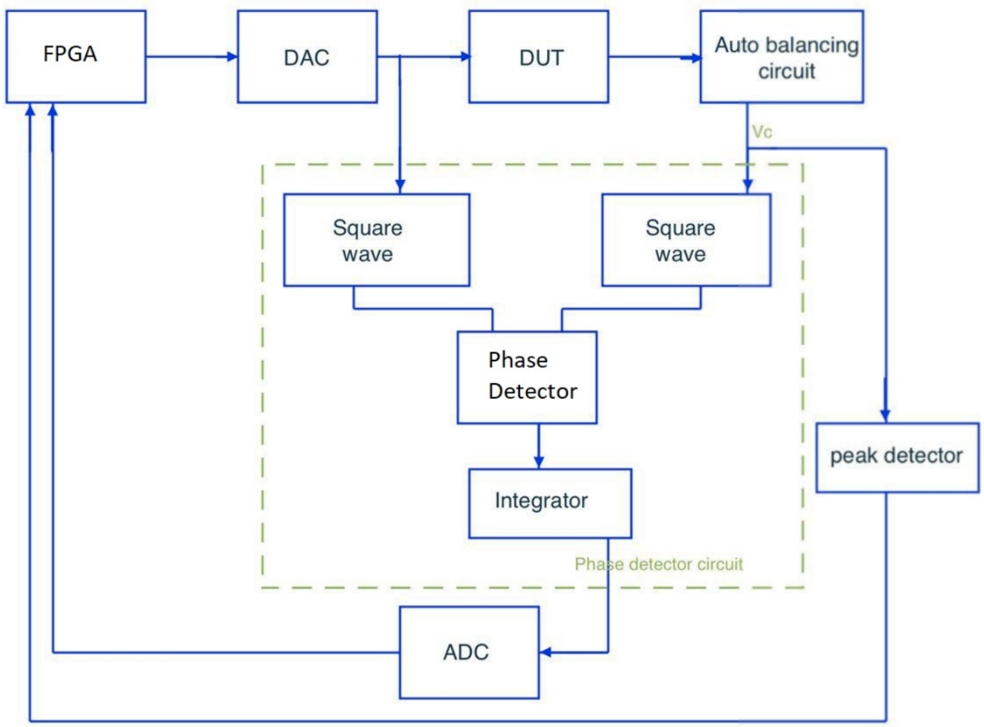

Figure 1: Block diagram of the project

2.1 Sinusoidal Signal Generation

2.1.1 De0 Nano Board

The first step in this circuit is obviously generating the frequency sweep which will power the circuit and help us perform the needed analysis. To this effect, we have used **a De0 Nano Board** to generate voltage signals between 0 and 3.3V with frequency ranging from 100 Hz to 50 kHz, each lasting for 10 seconds *(in order to allow any transients to stabilize)*.
The implementation is pretty straightforward. We have defined a look up table (LUT) which is

stored in the FPGA memory and keeps track of the values to be sent for the sinusoidal signal. After every 10 seconds, it updates an internal counter to switch to the next higher frequency signal.

2.1.2 DAC Interfacing

Now, the FPGA just generates digital signals but our circuit requires analog values. A DAC is, therefore, needed in order to provide an input to the circuit. Instead of using any DAC IC which was cumbersome and slightly expensive, we implemented a simple R-2R ladder (using resistance values as $1k\Omega$ and $2k\Omega$). And yes, we were successful in finding perfect R and 2R pairs!

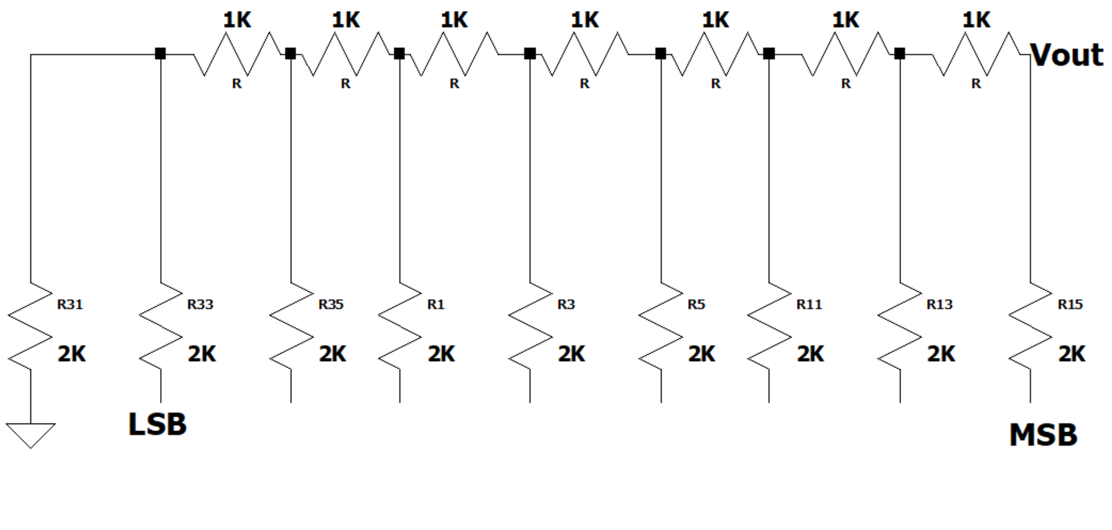

Figure 2: R-2R DAC

2.1.3 Buffer Circuit

In order to ensure effective isolation of the signal generated after the R-2R DAC, we used a standard OPAMP buffer circuit. Isolation is not the only advantage of using this circuit, though. This circuit also shifts the input signal, which was between 0 and 3.3V, to an output signal between -2.5V and 2.5V, as is needed by the subsequent sub-circuits.

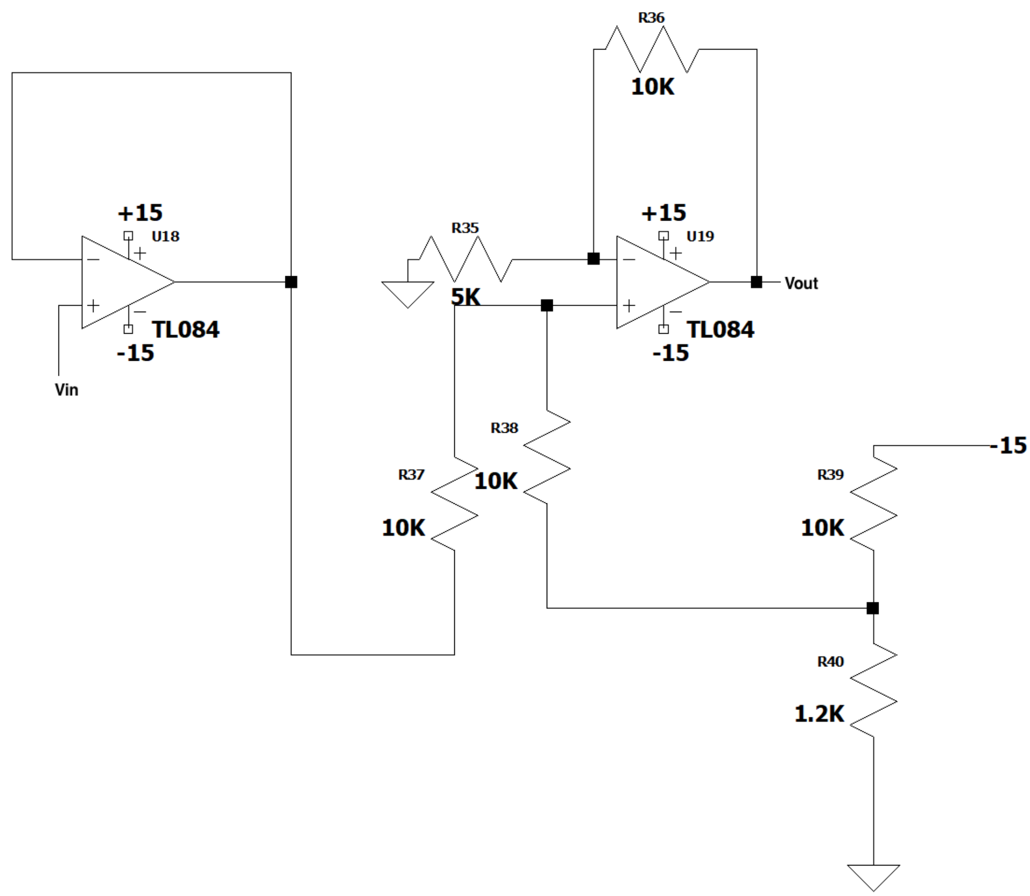

Figure 3: Buffer Circuit

2.2 Auto Balancing Bridge

This circuit takes the voltage signal generated by the overall first block and passes it through the given DUT. Now, the DUT may be resistive, inductive or capacitive. However, all we care about is that it will have a peak value varying with frequency and a phase that may be variable with frequency as well. This peak value and phase value is all that we are interested in eventually. The bridge gives us two square waves between 0 and 5V differing in phase which is equal to the phase shift introduced by the DUT. This is passed on to the phase detector and peak detector circuits which finally pass their outputs to the Arduino board.

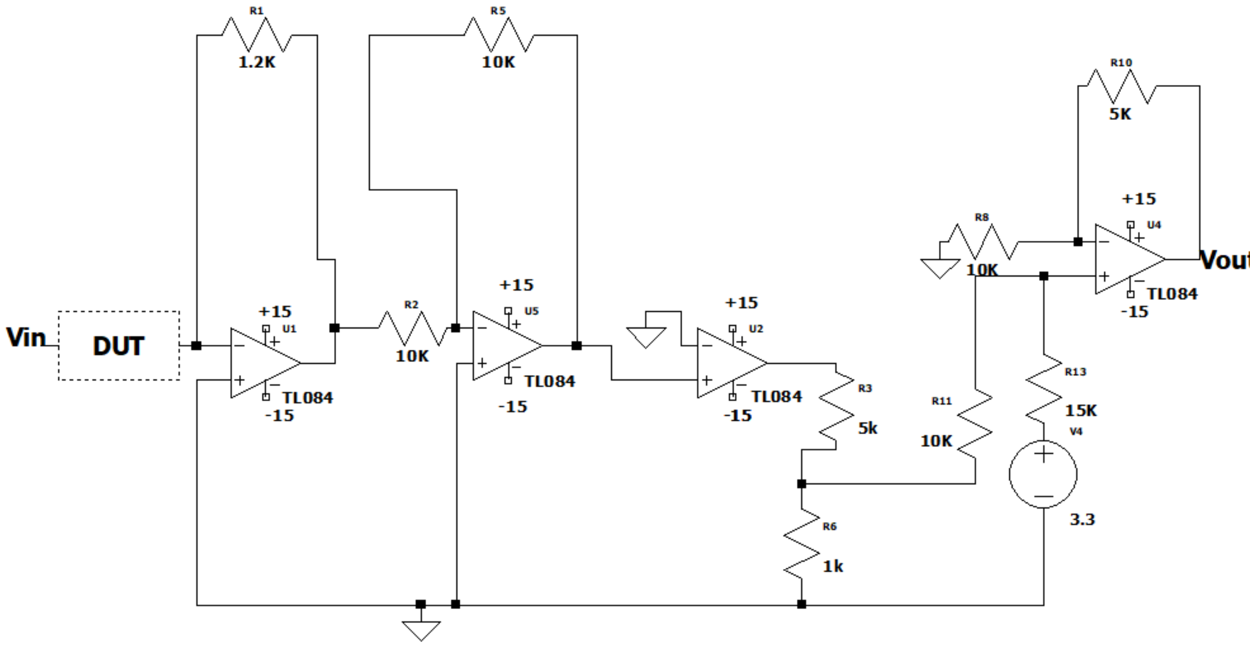

Figure 4: Auto - Balancing bridge

2.3 Phase Detection

Given two phase shifted waves, this circuit returns a DC voltage which varies linearly with the phase (as expected). The agreement of the predicted value with the actual phase difference is excellent and is fairly accurate.
We connect the previously generated phase shifted square wave to the clock of the D flip flop and connect D to +5 V. Whenever there is an edge in first wave it will trigger the Q1 to be high. Q1 will be high till the second square wave becomes high. As soon as second square wave becomes high it will make Q2 high which will further make clear active. The output is then passed through a low pass filter to give a final DC output which is then read by the Arduino Board.

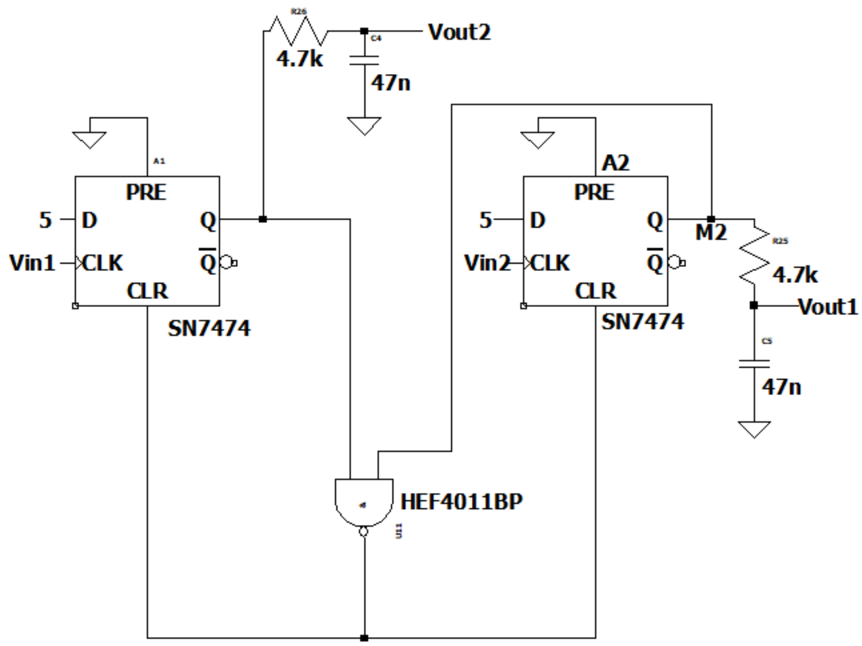

Figure 5: Phase Detector

2.4 Peak Detection

Given a square wave, this circuit returns the peak DC voltage of the input signal. The output is accurate for a lower frequency range than 50 kHz. However, this is easily rectified by using proper RC values such that the attenuation does not occur till 50 kHz. Additionally, we have fabricated two of these circuits on the PCB and their values shall be divided in the receiver code. Hence, even if there is attenuation, that shall be accounted for because of voltage division. This behavior is acceptable because our output depends only on the amplitude of the input signal and not on its frequency. The internal components of the circuit have been adjusted in accordance with our operating limits.

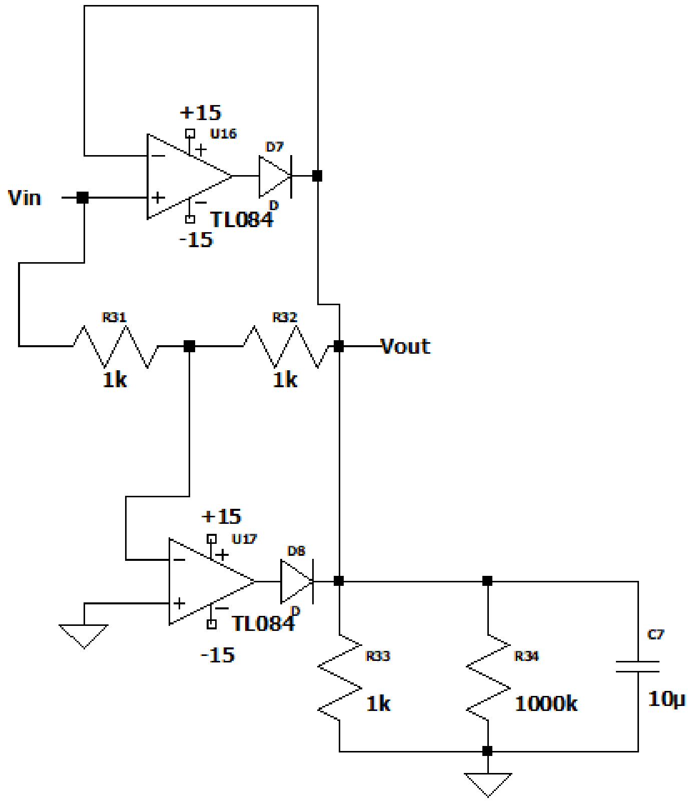

Figure 6: Peak Detector

3 PCB Design

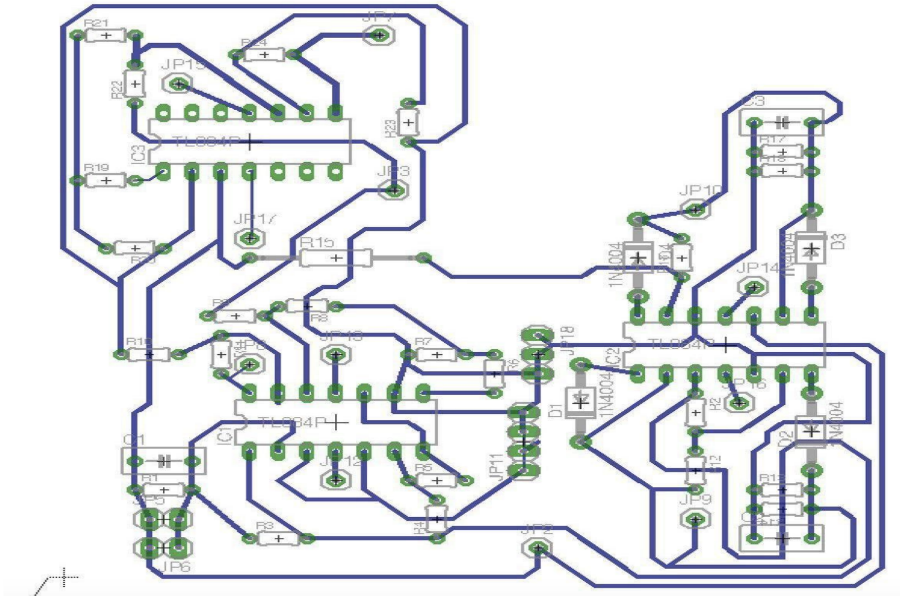

Figure 7: Schematic : Auto Balancing Bridge and Peak Detector

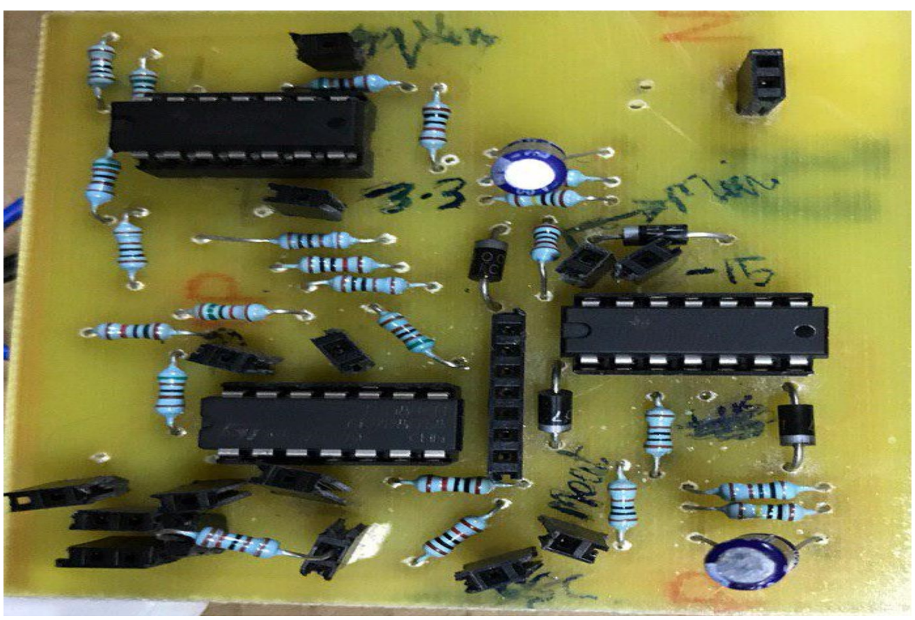

Figure 8: Fabricated : Auto Balancing Bridge and Peak Detector

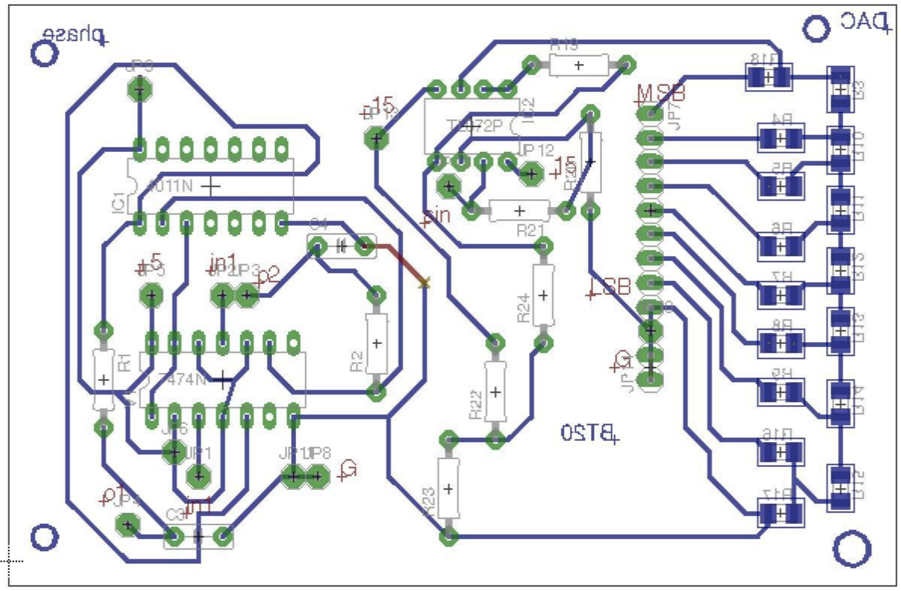

Figure 9: Schematic ; Phase Detector and R-2R DAC

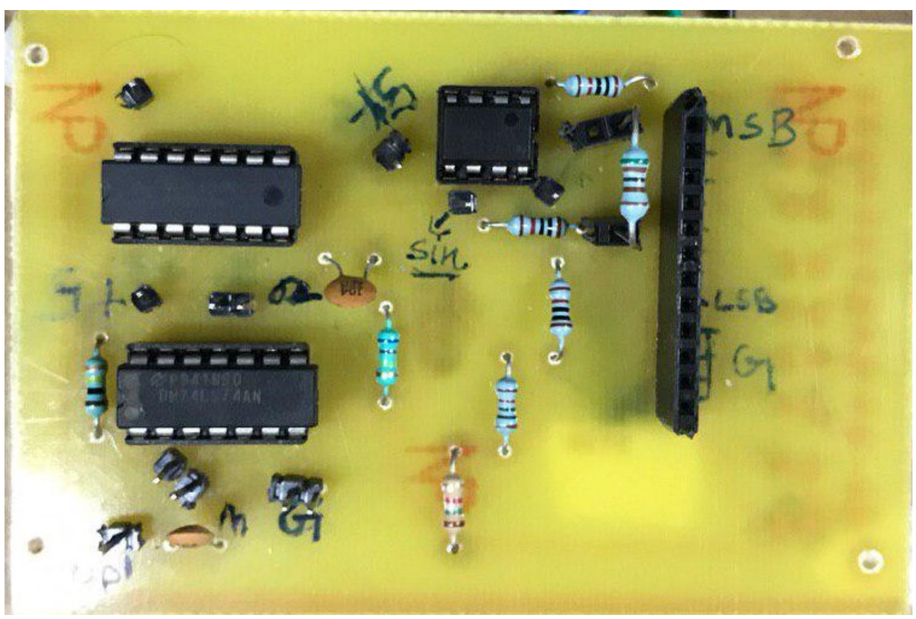

Figure 10: Fabricated : Auto Balancing Bridge and Peak Detector

4 Circuit Characterization

4.1 Peak Detector

4.1.1 Results

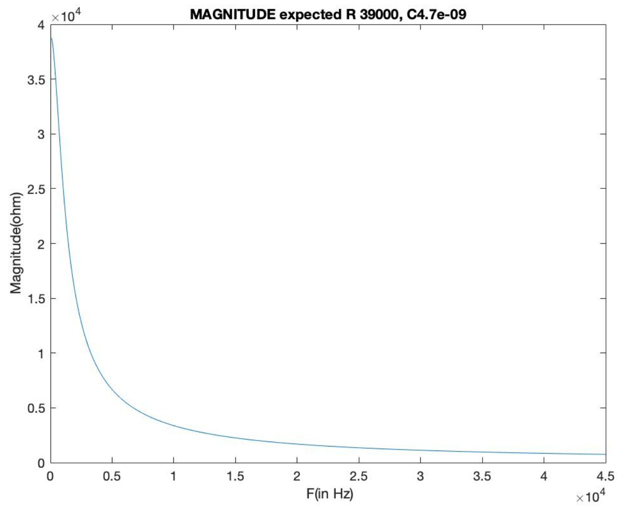

Figure 11: Magnitude Plot (Simulation): DUT 1

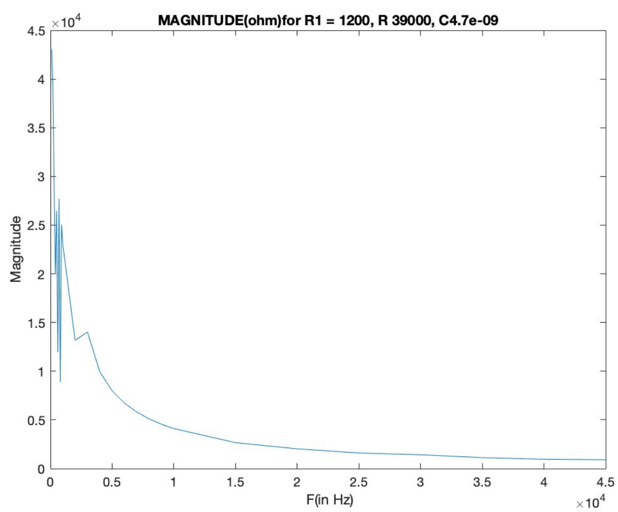

Figure 12: Magnitude Plot (Measured) : DUT 1

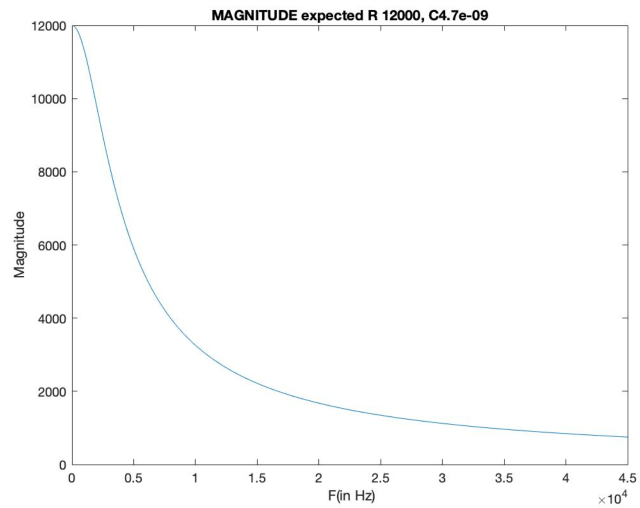

Figure 13: Magnitude Plot (Simulation): DUT 2

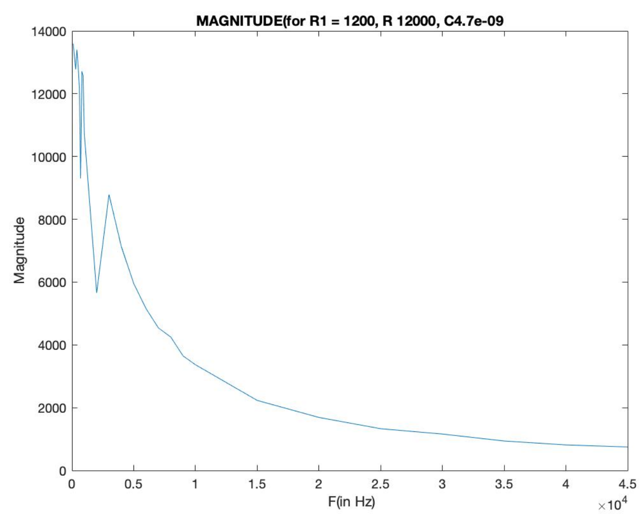

Figure 14: Magnitude Plot (Measured) : DUT 2

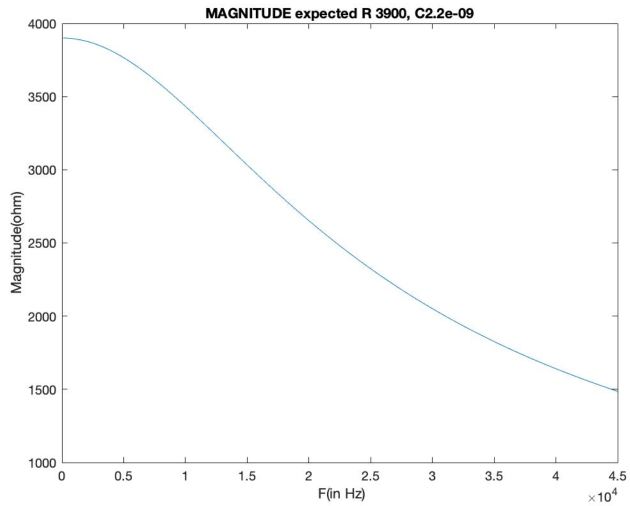

Figure 15: Magnitude Plot (Simulation): DUT 3

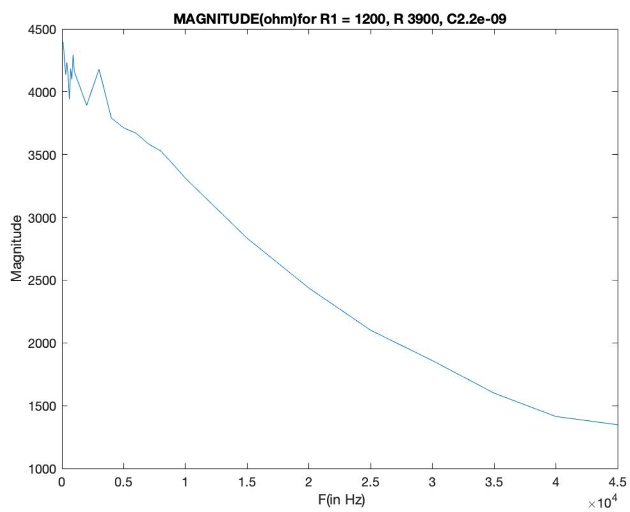

Figure 16: Magnitude Plot (Measured) : DUT 3

4.2 Phase Detector

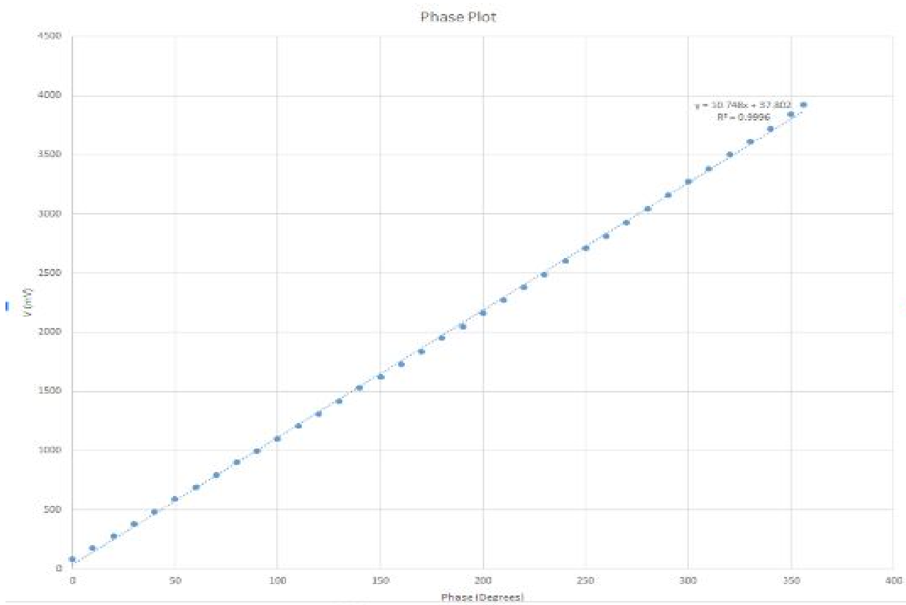

Figure 17: Phase Characterization

4.2.1 Results

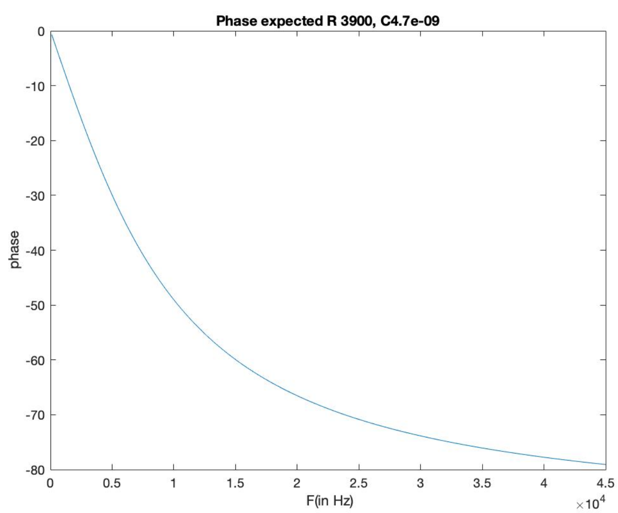

Figure 18: Phase Plot (Simulation)

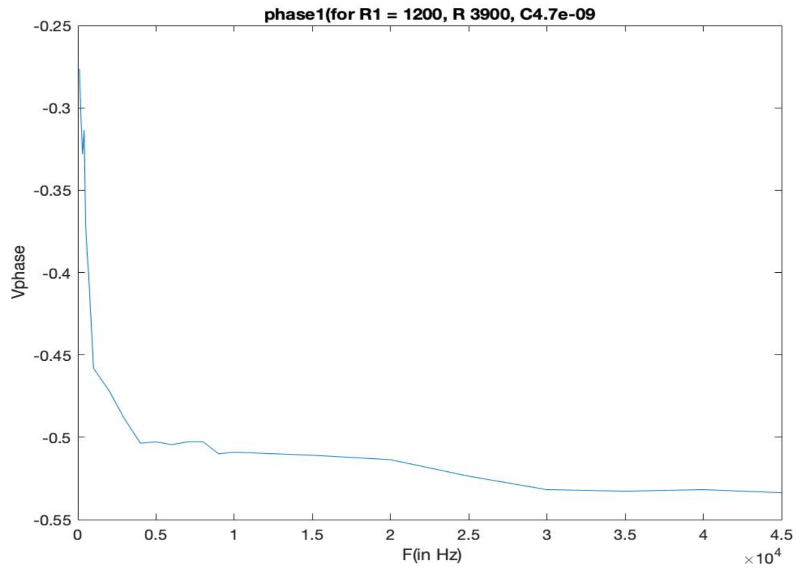

Figure 19: Phase Plot (Measurement)

5 Test Results

5.1 Derivation

Given a parallel R-C Circuit as our unknown circuit, we know that for the DUT ($\omega = 2\pi f$)

$$Z = \frac{R}{\sqrt{1+\omega^2 C^2 R^2}}$$

$$\theta = tan^{-1}(-\omega RC)$$

Using these, we obtain

$$R = Zsec(\theta)$$

$$C = -\frac{sin\theta}{Z\omega}$$

Again, the value of θ can be calculated from our earlier phase detector circuit characterization by the equation as follows -

$$V_{measured}(mV) = 10.32\theta + 72.34$$

5.2 Problems Faced

While we are able to do a frequency sweep analysis (which was the biggest objective we had in mind), we faced an unexpected error in the values from the phase detection circuit when we were giving input signals from the De0 Nano board. In order to try and remove the errors, we performed the following steps -

- Increased the time per signal from 1 second to 10 second just in case the D-Flip Flop didn't have such a fast response.
- Checked if there was a need for decoupling capacitance. This step did improve the readings but it wasn't as good as we would like to have
- Performed multiple characterizations to observe if there was some non linear relation between the expected and achieved values. However, we did not find any (neither did we expect to because the circuit worked well under isolation).

We believe that we could have done the following things better -

- Fabricate only one PCB instead of three to reduce the loose connections.
- Ensured proper isolation between each sub circuit so that no issues because of that could arrive. The possibility of such a situation led us to wasting a lot of time for the debugging.
- We should have used only the De0 Nano board for taking the readings one after the other. However, we were unable to do this because of a lack of supporting documentation or coding skills in VHDL.
- We could have used better op-amps with a faster slew to increase our operating frequency.
- We could analysed data better using machine learning models to find a phase and voltage relationship.

5.3 Further Work

We are currently unable to get a phase detector output which is reproducible and matches the simulation results as exactly as the peak detector output (which is very close to the expected results). Now, once that step is done, we will be able to achieve the estimation of R, C and L values for a given circuit model.

5.4 Final Results

After replacing an IC (the D-Flip Flops that we used), we discovered a better matching with the expected results. These have been shown below. The figures are in the order are -
The figures on the top are the actual measurements and the ones on the bottom are the simulated results. Also, the ones on the left are the magnitude plots and the ones on the

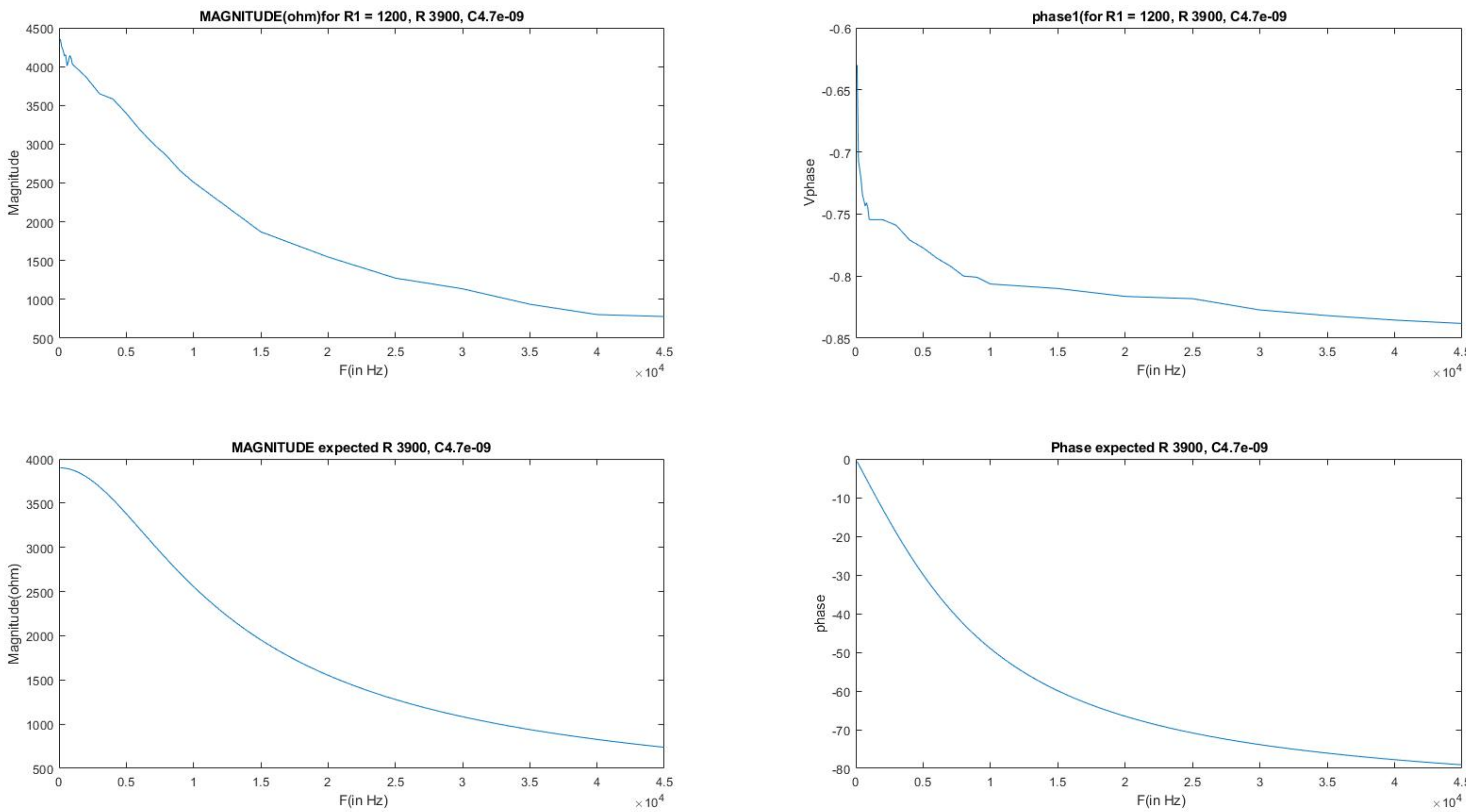

Figure 20: Final Measurements