ECE 2200/2210

Lab 3: Thévenin Equivalent Circuits and Digital to Analog Converters

Possible Points:100

Lab Equipment List:

- Breadboard and wires
- Resistors (560, 1k, 10k, and a few resistors of your choice)
- Triple Output DC Power Supply
- Digital Multimeter (DMM)
- Wires with connectors on one end and alligator clips on the other
- In addition, checking out a portable multimeter from the stockroom can be useful.

Partnering:

- Everyone must create their own lab report (fill out this document).
- Everyone must build their own circuits, but you may use the same measurement equipment.
- Discussions are encouraged, and you are also encouraged to answer each other's questions!
- Seek out the TA if you get stuck or need help.

Lab Procedures:

Welcome to Lab 3! This lab is designed to help you understand some of the circuit analysis techniques we've discussed in class in greater detail.

Part 1: Thévenin and Norton Equivalent Circuits (53pts)

Thévenin equivalent circuits are a very helpful tool for modeling complicated power supply systems in a more simple manner, simplifying circuit analysis, and generalizing system behavior. In this part of the lab, we will first be running some LTSpice simulations to measure the voltage and current across three different load resistors that are attached to a larger circuit. We will then use those results to determine the Thévenin and Norton equivalent circuits. Finally, we'll then build the Thévenin equivalent circuit on a breadboard and take measurements using the same three load resistance values to see how well the physical Thévenin equivalent circuit compares to the simulation of the more complicated circuit.

1. LTSpice should already be installed on the CADE lab computers. As a result, it will be easiest to complete this section by first logging into one of the CADE lab computers.

2. Download the zip file of schematics from the Canvas page for the lab and unzip it. There should be three files in the same directory when you unzip it.

3. Open the provided Lab3_Equivelent.asc file in LTSpice (this is the easiest way to open the LTSpice program, rather than trying to find the program itself on the computer). This file contains a single load resistor as well as a block that represents the circuit for which we will find the Thévenin and Norton equivalent circuits.

4. For this step, you will be running the LTSpice simulation three times (one simulation for each different load resistor). Choose three resistors that you will test as loads to the circuit you just opened in LTSpice. These resistors should have values of 100 ohm, 1 Meg, and any 3 values in between. Also, since you'll be testing these resistors later in measurements, make sure the resistors are values that exist in your lab kit.

5. After choosing your three resistor values, to prepare the simulation, begin by labeling the top node coming from the A port of X1. To do so, click on the "Label Net" tool from the toolbar (or press F4):

In the empty field, write Va. Place the label so that the box below the letters intersects with the top node coming from A and going into R_L.

6. Next, add the first resistor value. To do this, move your cursor over the R_L's schematic symbol (not its label). Your cursor should turn into a pointing finger. Right click on the symbol and enter your first value for R_L in the Resistance [Ω] field. Note that no units are required in LTSpice. However, do place the standard prefix, i.e. k for kilo- and Meg for mega-, after the number if needed. Don't worry about Tolerance or Power Rating.

7. After adding your first value for R_L, click the run button from the toolbar:

A small window will appear providing the voltage and current values in the circuit. Record this information in the "Resistor 1" column of Table 1. Here you should include units.

8. Repeat for your other two resistor values and fill in the "Resistor 2" and "Resistor 3" columns of Table 1. Make sure one of the values is Thévenin resistance (2pts/unit = 16pts)

R value	**Short Circuit Load**	**Resistor 1 = --- Ω**	**Resistor 2 = --- Ω**	**Resistor 3 = --- Ω**	**Open Circuit Load**
Voltage	0 V				
Current					0 A

Table 1

9. Find what the voltage would be when no load resistor is present (for this, you can set $R_L = \infty$ in LTSpice) For this, choose a significantly high value i.e. 5000Kohm. Record this value in Table 1 (i.e. fill in the missing value in the "Open Circuit Load" column).

10. Find what the current would be when the load is a short circuit (a wire, for this, you can set $R_L = 0$ in LTSpice). Show your work below. Record this value in Table 1 (i.e. fill in the missing value in the "Short Circuit Load" column). (Caution, finding the short circuit current without knowing what the internals of a circuit are is generally a bad idea, but thankfully, we know that nothing will physically burn up in a simulation.)

11. Now that you have filled in Table 1, we will create a plot of the Table 1 voltages (y-axis) vs. the Table 1 resistances (x-axis) using MATLAB. Open MATLAB and create a new .m file. (The CADE lab computers have MATLAB installed.)

12. At the beginning of your .m file, paste and fill in the following (anything that comes after a % is a comment and is not read by MATLAB when running the program):

```
% Fill in your name here
% Date
% Short description of what this code does

clear;  % good to clear all variables before starting
close all; % helpful to close old figures so we know
           % which plot is our new one
```

13. At the end of the MATLAB file you just created, add the following. Also fill in the resistances, voltages, and currents from Table 1:

```
% Create an array that will hold your resistor values
% separate the resistor values with a comma (e.g. [100, 200, 300])
resistance = [ ]; % ohms

% Create an array that will hold your load voltage values
voltage = [ ];          % Volts

% Create an array that will hold your load current values
current = [ ];      % mA
```

14. At the end of the same MATLAB file, let's create the voltage plot. Paste in the following:

```
% Plot the voltages vs. resistances
yyaxis left;
plot(resistance, voltage,'LineWidth',2);
xlabel('Resistance (ohms)', 'FontSize',14); % label the x-axis
ylabel('Voltage (V)', 'FontSize', 14); % label the left-side y-axis
```

15. Try running your program. What do you see? If you've chosen load resistor values ranging from ~100 and ~1 Meg as requested, the results for the large resistor values probably dominate your plot (more than the results for the smaller resistor values). So, try instead to plot the (x-axis) resistor values on a log scale so you can see more of what is going on across all resistor values. Replace:

```
plot(resistance, voltage,'LineWidth',2);
```

with:

```
semilogx(resistance, voltage,'LineWidth',2);
```

If you ever want to know what a MATLAB command is doing, you can type "help semilogx" (for example) at the command line.

16. Rerun your program and see how the plot changed. Hopefully you can see more of what is going on.

17. Now let's add the currents to the plot. Paste in the following at the end of your .m file:

```
% add this so the next plot will show up in the same figure
hold on

% Plot the currents vs. resistances
yyaxis right;
semilogx(resistance, current,'LineWidth',2);

% label the right-side y-axis
ylabel('Current (mA)', 'FontSize', 14);
```

18. And, we'll add a title and a legend to the plot. Paste in the following at the end of your code:

```
% add a title for your plot
title('Load Voltage / Current vs Load Resistance', 'FontSize',14);
% add a legend
legend('Voltage (V)', 'Current (mA)', 'FontSize', 14);
hold off
```

19. Run your program and see how the results look. Is this what you expect?

20. Lastly, let's create a separate plot of the power dissipated by the load (as Fig. 2). Paste the following at the end of your .m file and run the program one last time.

```
% Let's create a separate plot of the power dissipated by the load
figure (2);
% multiply the voltage * current to find the power
% dissipated by the load
% don't forget to convert the current to mA
% first before multiplying!
power = current.*0.001.*voltage;
% plot the results
semilogx(resistance,power, 'LineWidth', 2)
% add a title for your plot
title('Power Dissipated by the Load', 'FontSize',14);
xlabel('Resistance (ohms)', 'FontSize',14);
ylabel('Power (W)', 'FontSize', 14); % label the right-side y-axis
```

21. Paste Fig. 1 below. (3pts)

<put Fig. 1 from your MATLAB code here>

22. Comment on the shape of Fig. 1 (is it linear, not linear, and is this what you would expect?). If you want to, you can obtain more voltage and current results for other resistor values and add them to your plots. Hint: It's probably not linear since we used a log scale for the x-axis. (3pts)

    <fill in >

23. Paste Fig. 2 below. (3pts)

    <put Fig. 1 from your MATLAB code here>

24. Comment on the shape of Fig. 2. (Is this what you would expect? What does the peak correspond to? Note the peak might not be exactly where you expect it to be, since we only tested certain resistor values.) (3pts)

    <fill in >

25. Based on the data in Table 1, find the Thévenin/Norton Resistance and draw both types of equivalent circuits below. (6pts)

    <put both diagrams here>

26. You should find that the Thévenin Resistance you found from the simulation results is a resistor that is in your kit. Build the Thévenin equivalent circuit on your breadboard.

27. Using the circuit you just built, test the three load resistor values from your kit that you previously tested in the simulation and record the voltage and current in Table 2 below. (2pts/unit=12pts)

R_L value	Resistor 1 = __ Ω	Resistor 2 = __ Ω	Resistor 3 = __ Ω
Voltage			
Current			

Table 2

28. Comment on how well the equivalent circuit you built (Table 2 results) produces the output of the simulated circuit (Table 1 results). (2pts)

    <fill in >

29. Now that the equivalent circuits have been found, open up the Lab3_Internals.asc file which contains the internal circuit of the block in the simulation.

30. Analytically calculate the Thévenin or Norton equivalent circuit using the internal circuit diagram. (3pts)

    <put your diagrams here>

31. What load resistor value for this circuit would dissipate the most power? How do you know? (2pts)

    <fill in >

Part 2: Digital to Analog Converter via Superposition (22pts)

In many electrical systems, it is very desirable to use signals which are “On” or “Off” rather than a continuous voltage. One reason is that such systems can operate with a very low power level. Also, they are very resistant to noise. And, they can even be used to convey information about a continuous (or “analog”) signal.

The process of converting these digital signals to analog ones is completed by a Digital to Analog Converter (or DAC for short). Some of the simplest forms of DAC’s use superposition to accomplish this goal. For this lab we will build a 2-bit DAC to demonstrate this, although real-world DACs are often 8, 10, or 12 bits.

1. To start, build the circuit shown in Fig. 1 using your breadboard. Because two separate voltage sources are required, use the +25V supply as one power supply, and 6V supply for the other. To use both, turn on one, then use the appropriate button to switch to the other to turn it on. If you have connected everything properly, you should get Vout to be about 4.8 V when both sources are on.

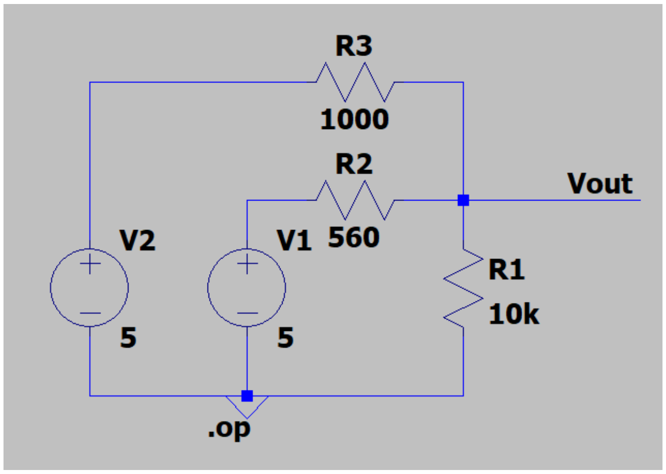

Fig. 1

2. In this circuit, V1 and V2 represent the two-bit digital signal coming into the DAC. “On” is when a the voltage source is set to 5 V, and “Off” is when a voltage source is set to 0 V or replaced with a wire. Record all four possible combinations (Off Off, Off On, On Off, and On On) in Table 3. (2pts/unit=8pts)

	ON	**OFF**
ON		
OFF		

Table 3

3. Show that this circuit is behaving in superposition (Hint: See if the sum satisfies). (2pts)

<fill in >

4. Comment on why circuits behave this way (Hint: you might consider mentioning that resistors are linear). (2pts)

<fill in >

5. Comment on why this sort of circuit may be useful in industry or for other applications. (2pts)

<fill in >

6. To get a little more experience with simulation, build the DAC circuit in LTSpice. Your TA will be able to help you if you run into problems.

7. Once you have put in all of the components, select run. The simulation option you want is DC Operating point (.op). Try turning on and off the sources (setting them to 0 V) and see if they match your actual values. Record your results in Table 4. (2pts/unit=8pts)

	ON	**OFF**
ON		
OFF		

Table 4

Supplementary Note: This design can actually be extended for more bits by using larger and larger resistors. For instance, the next bit would use a 2k resistor, the next a 4k resistor, and so on. However, due to tolerancing, this approach is not used all that much, and R-2R ladders, which are similar to this design, are much more common.

Part 3: Exposing a Problem using Equivalent Circuits (25pts)

A variant of the DAC we considered in Part 2 is called an R-2R ladder. R-2R ladders are commonly used in low cost, low resolution parts because of its relative simplicity over other DAC designs. However, although these types of DACs are common, they can introduce some problems:

- First, R-2R ladders require very accurate resistor values when creating many bit converters. Otherwise, the resulting voltages will be very inaccurate.
- Second, the output voltages are connected in such a way that they can be inaccurate under load (when another electrical component is connected to the output of the R-2R ladder). To see this type of problem in action, consider the equivalent circuit of a DAC R-2R ladder as shown in Fig. 2, which includes a load (resistor) connected to the output of the DAC (on the right side of the diagram).

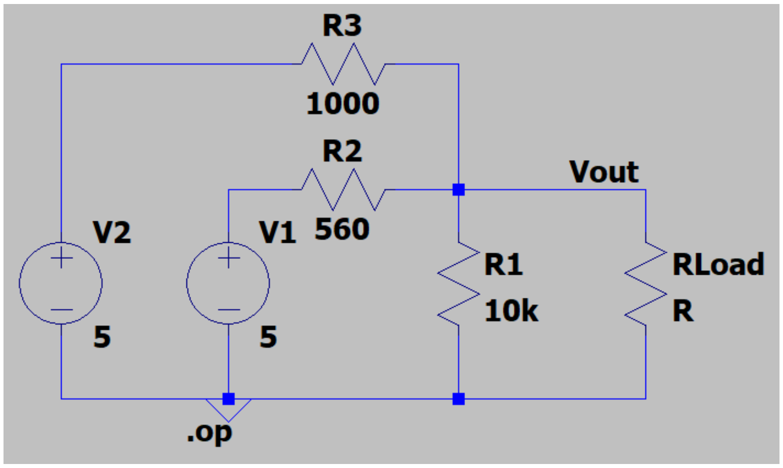

Fig. 2 2-bit R-2R ladder DAC.

1. Draw a Thévenin equivalent circuit for the circuit shown in Fig. 2. Make sure to note the Thèvenin voltage and the resistance in your diagram. (6pts)

<fill in >

2. Does the Thévenin resistance in this circuit change if certain voltage sources are turned off? Why or Why not? Would this change if the sources were current sources? (3pts)

   <fill in >

3. Once you have an equivalent circuit, think through how attaching different values of load resistors would change the circuit. Specifically, record in Table 4 what the load voltage would be with no load, a very large resistor, a resistor that causes the maximum power to be dissipated, and a very small resistor. You might consider connecting the resistors to the circuit to test the equivalent circuit as well. (2pts/unit=8pts)

R_L	No Load	Very Large Resistor	Resistor that Dissipates Max Power	Very Small resistor
V_L				

Table 4

4. What was the effect of the varying resistance? How does that lead to a problem for this DAC? (4pts)

   <fill in >

5. For what type of load resistance would this DAC be most accurate to the original voltages? How does this relate to the amount of current that can be supplied by the DAC? (4pts)

   <fill in >

Supplementary Note: The voltages from these DAC's can be made much more accurate by using a voltage follower (comprised of an op amp), which you either just learned about in class or will shortly. It overcomes the issue by recreating the input voltage using a different power supply, thus meaning that the output can draw as much current as the new power supply can handle without changing the input voltage.

6. Before you tear down your last circuit and leave the lab today, make sure to have your Lab TA check off your lab by showing what you accomplished in your lab report.

7. Turn in this completed report to Canvas.

8. Submit your .asc and .m files to Canvas along with this report.

9. Put all of the lab supplies away, including your own lab kit and anything you may have checked out!