

CPE 1140

# Circuits / DC Circuit Fundamentals Lab

Fall 2021

Laboratory Report

Lab# 4

# Lab Series and Parallel DC Circuit Analysis

Submitted by: Bruce Liu

Laboratory Date: 9/30/2021

Date of Submission: 10/3/2021

Part A:

|  |  |  |
| --- | --- | --- |
| Predicted values | Measured values | Tolerance |
| Rp = 1.421 kΩ | Rp = 1.411 kΩ | Rp = 1349.950 – 1492.050 Ω |
| Rt = 3.021 kΩ | Rt = 3.002 kΩ | Rt = 2869.950 – 3172.050 Ω |
| VS = 15.000 V Note: set to 15V | VS = 15.001 V | Null |

R4 + R5 = 3.0kΩ

Rp = 1/(1/+2.7kΩ+1/3.0kΩ) = 1.421kΩ

Rt = 1421+470+510+620 =3.021kΩ

Values are within tolerance ranges. Note: power supply was reporting 14.9 when the multimeter measured 15.001 V I took the value of the more accurate tool. Methods of calculating resistance parallel and series of resistance are accurate.

Part B:

|  |  |  |
| --- | --- | --- |
| Voltage drops Simulated (V) | Voltage drops Predicted (V) | Voltage drops measured (V) |
| V1 = 2.334 | V1 = 2.333 | V1 = 2.333 |
| V2 = 3.079 | V2 = 3.078 | V2 = 3.095 |
| V3 = 7.054 | V3 = 7.055 | V3 = 7.055 |
| V4 = 2.352 | V4 = 2.351 | V4 = 2.339 |
| V5 = 4.702 | V5 = 4.702 | V5 = 4.710 |
| V6 = 2.532 | V6 = 2.532 | V6 = 2.515 |
| V input = 14.996 | V Sum loop = 14.996 | V Sum loop = 14.992 |

V sum loop = V1 + V2 + V4 + V5 +V6

Measure and predicted values are nearly identical. I don’t have extraneous values in my measurements. I did have an extraneous value during the lab when the resistors 4 and 5 were not connected to each other. Which resulted in resistor 6 reading at 3.5 ish.

The voltages chosen for the loop was for V sum loop = V1 + V2 + V4 + V5 +V6 for no reason in the prelab and later recalculated here as well. But in the future, I would use the loop with the least elements. Since it would be a better time save. I should have used V3 rather than V4 + V5 .

Part C:

|  |  |  |
| --- | --- | --- |
| Current simulated (mA) | Current predicted (mA) | Current measured (mA) |
| Ia = 4.967 | Ia = 4.965 | Ia = 4.999 |
| Ic  = 2.613 | Ic  = 2.613 | Ic = 2.630 |
| Id  = 2.351 | Id  = 2.351 | Id = 2.368 |

Current divider, simulation, and the lab measurements agree. No extraneous differences.

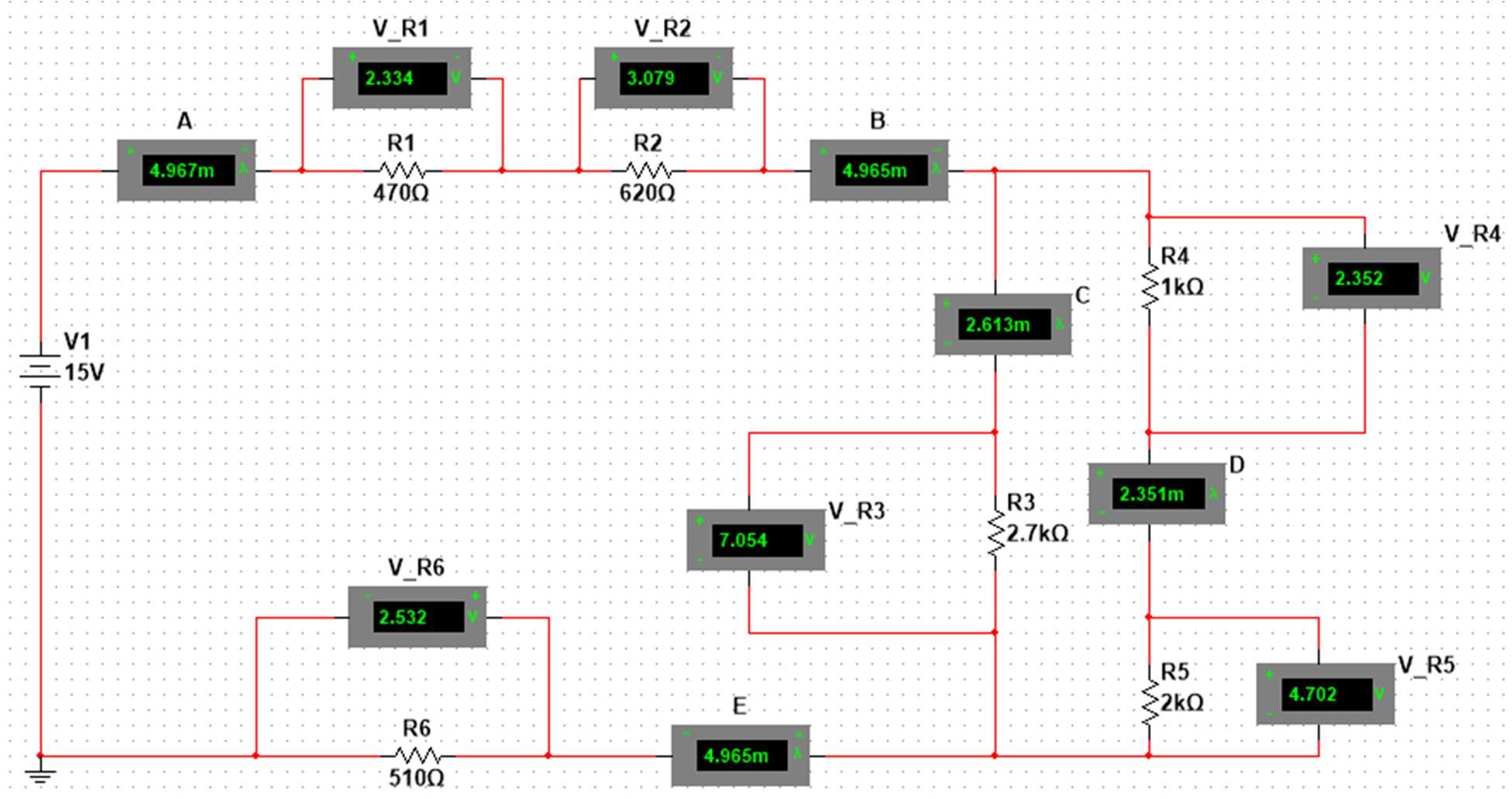
Part D:

I found that computing resistors from the farthest from the source seems to be easiest. I started to do the calculations mostly by hand for this lab. Since pushing an intermediate calculation to an array was going to be more work that it was worth.

The resistance of the series, parallel, and total values are accurate. The voltage drops are also accurate in between the simulated, calculated, and lab measured.

Appendix:

Multisim Simulation:



Prelab calculation code:

#include <stdio.h>

void resistance\_tolernaces()

{

double set\_of\_resistors[] = {470.0,

620.0,

2700.0,

1000.0,

2000.0,

510.0,

1421.0, 3021.0};

double resistor\_low = 0.0, resistor\_high = 0.0, toleranace\_value = 0.0; printf("\n\n");

for(int count = 0; count < 8; count++)

{

toleranace\_value = set\_of\_resistors[count] \*5/100; resistor\_low = set\_of\_resistors[count] - toleranace\_value; resistor\_high = set\_of\_resistors[count] + toleranace\_value;

printf("The tolerance range of %.4lf resistor is %.4lf - %.4lf \n\n", set\_of\_resistors[count], resistor\_low, resistor\_high);

}

/\*note you need to make a library file .h and a main function to use this.

in anaconda powershell I used gcc -O3 main.c functions.h functions.c -o lab\_4

you also need to install gcc to use this command run lab\_4.exe

\*/