EE1003 Introduction to Computer I

Programming Assignment 4: MySpice

Due Date: 2022/1/5 (Wed.) 23:59:59

Background

Circuit analysis is one of the most important problems in electronic circuitry. Andy takes an electronic circuitry class and the Professor assigns a bunch of homework about circuit analysis. The homework asks students to find out the voltage of each node and current of each resistor for the given circuit. As numerous homework piled up, he decides to design a software called *circuit simulator* to help he and his "happy little partners" to finish these homework easier. After carefully exam circuits in each homework, he finds that there are certain patterns in the circuit. All the circuits are composed of a voltage source and a number of resistors. Moreover, there are *n* stages in the circuit. Each stage is composed by two types of resistors, *resistor in series* and *resistor in parallel*. The number of resistors in each type is arbitrary. An example of a circuit is shown in Figure 1. In the figure, the stage 1 consists 1resistor in series (RS1_1) and 2 resistors in parallel (RP1_1, RP1_2), while stage 2 consists 2 resistors in series (RS2_1, RS2_2) and 1 resistor in parallel (RP2_1).

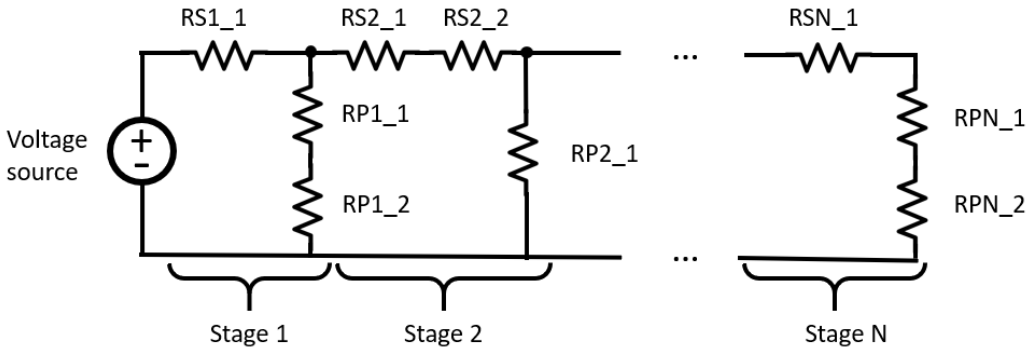

Figure 1. An example of a circuit in the homework.

Luckily, Andy is good at programming. Hence, he decides to turn the circuit into a .txt file as the input of the program. With the help of his "happy little partners", he can easily get the voltage of each node, and the current cross each resistor from the program. However, the advance problems ask students to swap and merge different stages. In order to get a high score in electronic circuitry class, Andy wants to develop a function which can swap and merge stages. In the end, Andy want to easily read the

result from a file. Therefore, he wants to output the result of the calculations and the circuit descriptions. With the help of *circuit simulator*, Andy can finally sleep well at night.

Description

Circuit Description

We use the following format to describe a circuit.

The naming rule of the resistors is **R*xy_z*** where ***x***={S, P} represents type of resistor (S for series and P for parallel), ***y*** shows the stage number, and ***z*** stands for the index of the resistors under the same type. For example, RP1_2 represents the second parallel resistor in the stage 1 as shown in the following figure. To identify the observation nodes on the circuit, we use **n*i*** to represent the connection point between series and parallel resistors where ***i*** is the stage index. For example, n1 means the node connecting series and parallel resistors in stage 1. Moreover, if more than one resistor is shown in series(parallel) path, we use **n*xy_z*** to represent the node where ***x***={S, P} represents type of resistor (S for series and P for parallel), ***y*** shows the stage number, and ***z*** stands for the index of the connection point in the series(parallel) path. For example, np1_1 shows the first connection point on the parallel path of the stage 1.

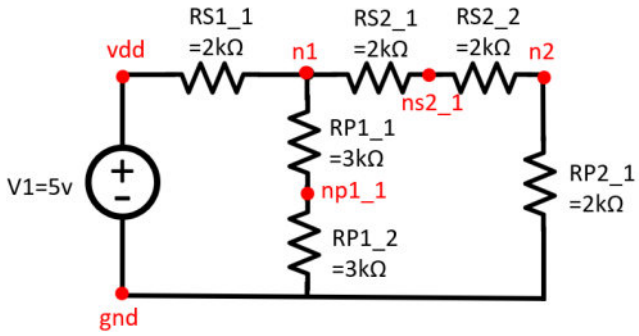

Figure 2. An example of the naming rule.

(This circuit will be the input circuit of the following examples)

Function Description

The main menu of the simulator owns the following options:

Please type 1 to input file

Please type 2 to calculate voltage and current

Please type 3 to swap stage

Please type 4 to merge stage

Please type 5 to output file

Please type 6 to exit

Please round the calculate results to **three decimal places** and keep the magnitude of the result **greater than 1** and **less than 999.999**. For instance, 0.500mA and 50000.000Ohm are not allowed. Please change them to 500.000uA and 50.000kOhm. The rage of the stage is [0-15]. The rage of resistors in series or in parallel of each stage is [1-15].

1. Input file

   When user types 1 in the main menu, the input data should load into the program. If the file loaded successfully, program should print success message on the scream. The formats of the input file are shown below:

   Voltage source:

   V1 vdd gnd 5v

   Resistance in series at first stage:

   RS1_1 vdd n1 2K

   Resistance in parallel at second stage:

   RP2_1 n2 gnd 2K

   The prefix of the program includes: **u, m, K, Meg, G**.

```
*** Welcome to MySpice ***
==========================================
Please type 1 to input file
Please type 2 to calculate voltage and current
Please type 3 to swap stage
Please type 4 to merge stage
Please type 5 to output file
Please type 6 to exit
Enter your selection Here:1
==========================================
Please enter the name of the input file: myspice.txt
The file is not found.
Please enter the name of the input file again: input.txt
Loading the file...
The input file successfully loaded!
```

2. Calculate voltage and current

First, with Kirchhoff's Current Law, we can easily calculate current cross each resistor. The directions of the current are shown in Figure 3.

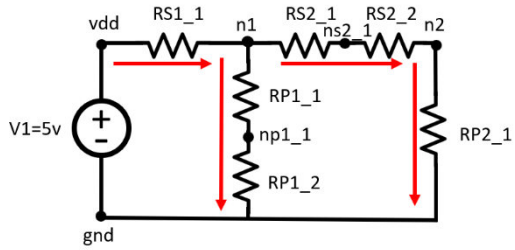

Figure 3. An example of the directions of the current.

Second, with Kirchhoff's Voltage Laws, we can easily calculate voltage of each node. The polarity of the resistor is shown in Figure 4.

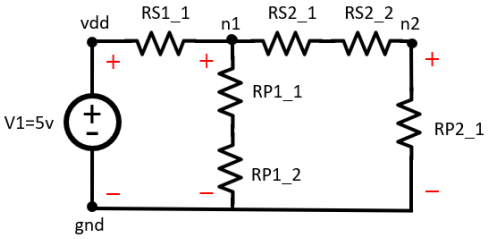

Figure 4. An example of the parity of the resistor.

Please print the current cross each resistor and voltage of each node on the scream. Your program needs to provide a fool-proof mechanism to prevent that there is no circuit to calculate. Please print "There is no circuit".

In this work, you must use **void calculation (…)** as the function prototype.

Note that the vision of the length of star (*) may change with different fonts. Hence, please make sure the **stars (*) are aligned in CodeBlocks**. The specific format of the calculation result is shown in the Figure 5.

Note that the file is opened with notepad++.

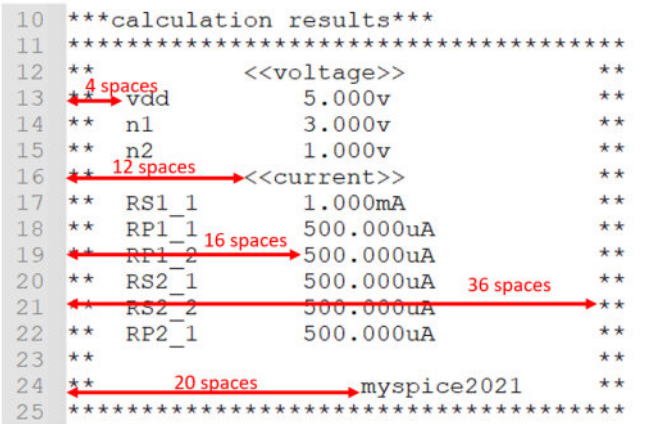

Figure. 5 The format of the calculation result.

```
========================================
Please type 1 to input file
Please type 2 to calculate voltage and current
Please type 3 to swap stage
Please type 4 to merge stage
Please type 5 to output file
Please type 6 to exit
Enter your selection Here: 2
========================================
Calculating...
The result is successfully calculated!
**********************************
**          <<voltage>>          **
**   vdd        5.000v           **
**   n1         3.000v           **
**   n2         1.000v           **
**          <<current>>          **
**   RS1_1      1.000mA          **
**   RP1_1      500.000uA        **
**   RP1_2      500.000uA        **
**   RS2_1      500.000uA        **
**   RS2_2      500.000uA        **
**   RP2_1      500.000uA        **
**                               **
**                  myspice2021  **
**********************************
```

3. swap stage

   When user types 3 in the main menu, the program will swap two stages.

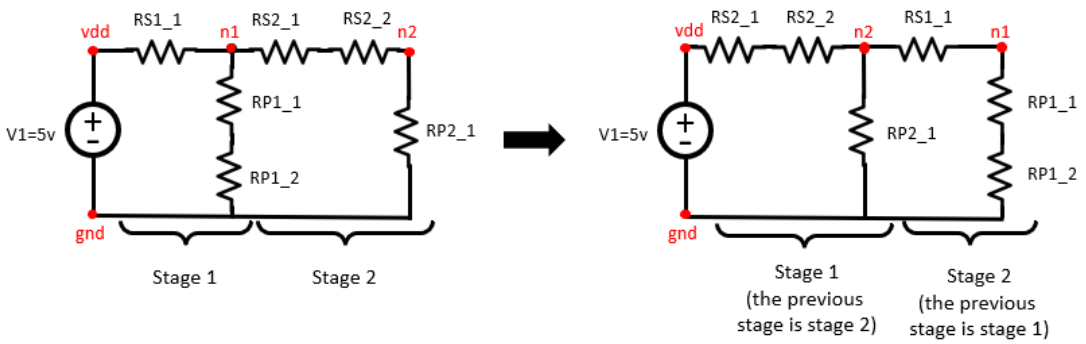

Figure 6. An example of swapping stage.

Your program needs to provide a fool-proof mechanism to prevent that there is no circuit to calculate. Please print "There is no circuit".

In this work, you must use **void swap_stage (…)** as the function prototype.

```
==========================================
Please type 1 to input file
Please type 2 to calculate voltage and current
Please type 3 to swap stage
Please type 4 to merge stage
Please type 5 to output file
Please type 6 to exit
Enter your selection Here: 3
==========================================
Please enter the first stage you want to swap: 4
The node doesn't exist. Please enter again.
Please enter the first stage you want to swap: 1
Please enter the second stage you want to swap: 2
RS1_1 is changing node from vdd to n2 …
RS2_1 is changing node from n1 to vdd …
The stage is successfully swapped!
```

4. merge stage

The program will merge two stages into one stage and replace the first stage.

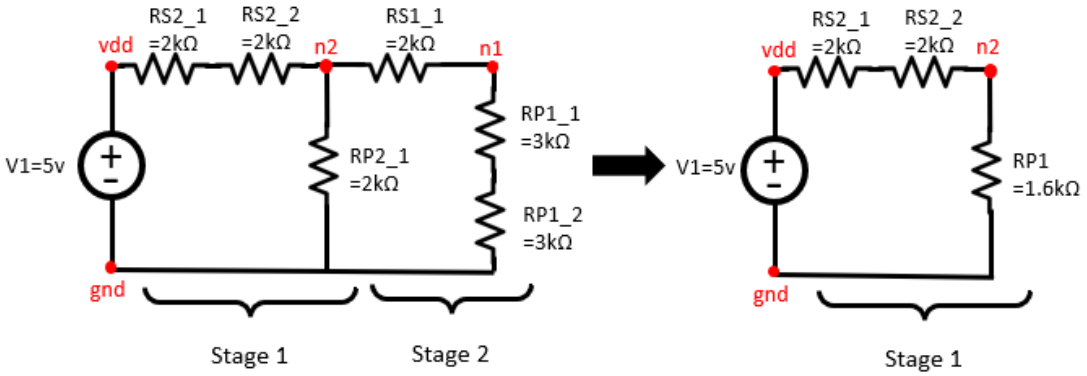

Figure 7. An example of merging stage.

Your program needs to provide a fool-proof mechanism to prevent that there is no circuit to calculate. Please print "There is no circuit".

In this work, you must use **void merge_stage (…)** as the function prototype.

```
==========================================
Please type 1 to input file
Please type 2 to calculate voltage and current
Please type 3 to swap stage
Please type 4 to merge stage
Please type 5 to output file
Please type 6 to exit
Enter your selection Here: 4
==========================================
Please enter the first stage you want to merge: 1
Please enter the second stage you want to merge: 2
Resistors are merged into RP1.
=> RP1 n2 gnd 1.600 k
The stage is Successfully merged!
```

5. output file

   If user want to extract the calculated result, the program will output the **result.txt** file. The output file should contain the result of the calculations and the circuit descriptions. And, the program should show success message on the scream.

   Your program needs to provide a fool-proof mechanism to prevent that there is no circuit to calculate. Please print "There is no circuit".

```
==========================================
Please type 1 to input file
Please type 2 to calculate voltage and current
Please type 3 to swap stage
Please type 4 to merge stage
Please type 5 to output file
Please type 6 to exit
Enter your selection Here: 5
==========================================
Please enter the name of the output file: result.txt
Exporting the file...
The output file successfully exported!
```

6. Exit

   Enter 6 to exit.

```
========================================
Please type 1 to input file
Please type 2 to calculate voltage and current
Please type 3 to swap stage
Please type 4 to merge stage
Please type 5 to output file
Please type 6 to exit
Enter your selection Here: 6
========================================
Bye~
```

Bonus

Thevenin's theorem

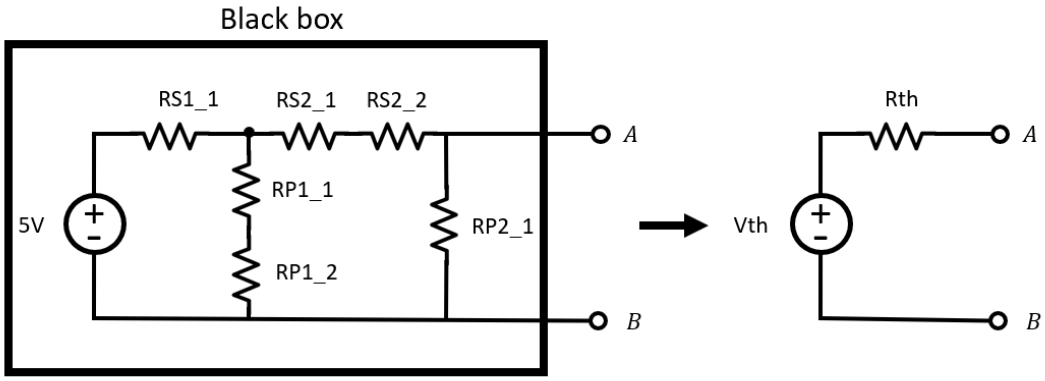

Figure 8. An example of Thevenin equivalence circuit.

Please derive the Thevenin equivalence circuit and export into an output file, which contain the result of the calculations.

Submission Requirement

You have to submit a source code file (not the entire project) named as StudID_PA4.cpp (ex: 986253465_PA4.cpp) and a report named StudID_Name_PA4_report.pdf (ex: 986253465_陳聿廣_PA4_report.pdf). If you implement a bonus version, please separate it from the original version. Name your source code file of the bonus version as StudID_PA4_bonus.cpp (ex: 986253465_PA4_bonus.cpp) and upload to ee-class with the original version. Note that

the only acceptable report file format is .pdf, no .doc/.docx or other files are acceptable. **BE SURE to follow the naming rule mentioned above. Otherwise, your program will be not graded.**

We don't restrict the report format and length. In your report, you have to at least describe:

1. How to compile and execute your program; (You can use screenshot to explain)
2. The completion of the assignment; (If you complete all requirements, just specify all)
3. The hardness of this assignment and how you overcome it;
4. Bonus function(s) you implement if any
5. Any suggestions about this programming assignment?

You can also put anything related to the PA in your report, such as pseudocode, control flow diagram, programming developing thought, etc. Your program will be judged with Code::Blocks 20.03 and GNU GCC Compiler (MinGW-W64 project version 8.1.0, 32/64 bit, SHE). We will use C++ 11 standard.

Grading

The grading is as follows:

(1) Correctness of your code: 50%
(2) Readability of your code: 10%
(3) The report: 10%
(4) Demo session: 30%
(5) Bonus (at most): 10%

Please submit your assignment on time. Otherwise, the penalty rule will apply:

- Within 72hrs delay: 20% off
- More than 3 days: 0 point

Be sure to attend a demo session (the time will be announced later). If you have questions, please E-mail to both me (andyygchen@ee.ncu.edu.tw) and TA 何宜真 (gumi627@gmail.com)

Reference

[1] Chapter 2 ~ Chapter 5, Paul Deitel and Harvey Deitel, "C++ how to program late objects version," 7th edition, Person 2011.

[2] How to calculate circuit in parallel: https://youtu.be/z0D1mtf0dLo

[3] How to calculate circuit in series:
https://youtu.be/lyxljU4RD-o
[4] How to calculate Thevenin's Theorem:
https://youtu.be/zTDgziJC-q8