

Section	LA1
Bench No.	5

ECE110 Introduction to Electronics

Pre-Lab 3: Switched-Resistor Speed-Control

Name: 潘岩峰

Student ID: 323010210

Instructor: 冯哲君

Date: 10.11

Teammate/Student ID: 沈扬 323010445

This part is reserved for your instructor

Score	
Instructor Signature	
Date	

Pre-Lab 3: Switched-Resistor Speed-Control

In experiment #2, we are further familiar with multimeter and power supply, learn to use resistance network to control the speed of motor, and understand the role of considering rated power in circuit design. In this experiment, we will learn how to further fine-tune the motor speed and verify KCL and KVL. In this prelab, we will learn a new simulation software: LTSpice, using simulation software to simulate circuits and study their characteristics. At the same time, get familiar with the application of basic circuit laws, and to prepare for experiment 3.

The Interpretation of Basic Circuit Laws

In 1845, Gustav Kirchhoff, partially working from Ohm's discovery, formulated two more mathematical theories. These are known as **Kirchhoff's Laws** and they drastically aid our understanding of circuits. As we begin our own exploration of electronics, we will re-investigate these fundamental laws of circuit theory ourselves.

Ohm's law may be used to describe a very simple circuit with only a single source and a single resistor. When two resistances are connected across the same voltage drop (such as the second schematic of Figure 1), there are, in fact, *two* paths for the current to flow from the positive terminal of the source to the negative terminal. In this configuration, the two resistances are said to be connected in **parallel**. It is sometimes beneficial to think of resistors as water pipes. The narrower the pipe, the larger the resistance to current flow. Having two resistances in parallel is similar to adding an extra pipe for the water to follow compared to the single resistor configuration. In the analogy, water would flow through both pipes, but more water would flow through the wider pipe as it offers less resistance to flow.

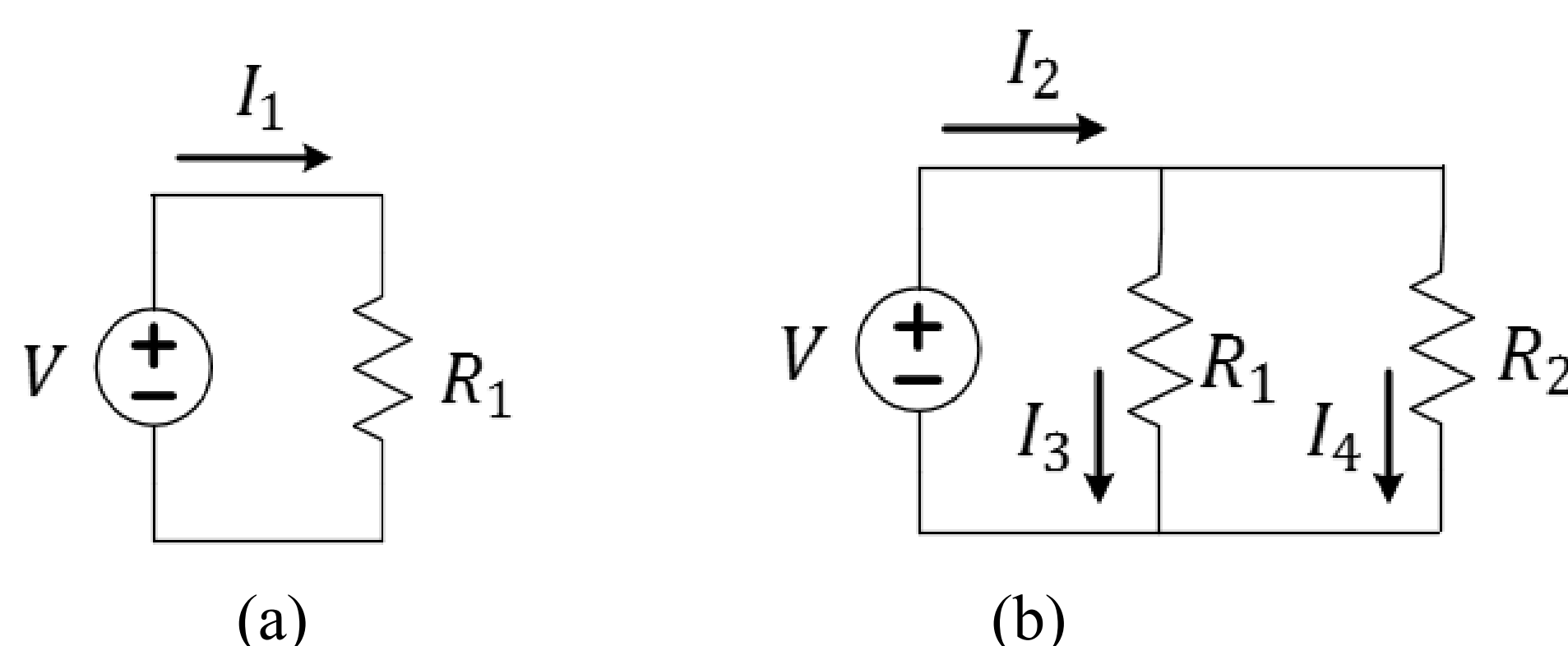


Figure 1: Two circuit schematics used to explore current in parallel resistive elements. In (a), the current has one path back to the source. In (b), the current has two parallel paths back to the source. The parallel combination will have less resistance to current flow than R_1 alone.

Question 1: Assume R_1 is the same value in both Figure 1(a) and (b). Then current I_2 will be larger than I_1 . Explain this fact using both Ohm's law and the water-pipe analogy.

Ohm's law: There are two resistances connected in parallel, thus making the whole resistance lower.

Water-pipe: There are two paths for current to go through, thus making I larger.

Question 2: If you are given that $R_1 = R_2$, how do I_3 and I_4 relate to one another? Is one greater than the other? How do they each relate to I_1 ? Explain your reasoning using Ohm's Law.

$$I_3 = I_4 = I_1, \quad I = \frac{V}{R} \quad V \text{ and } R \text{ remains the same value, so current won't change.}$$

The Role of Schematics in Electronic Circuits

There are three main ways in which you are likely to encounter a circuit design in the ECE110 lab. Since this lab is hands-on, you will build prototype circuits. A **prototype** is a preliminary version of a product that can be easily tested and modified before a final design is mass produced. In the ECE110 laboratory, we construct prototype circuits on a **breadboard** (a construction base for connecting circuit elements without the use of solder; might also be known as a protoboard, although this later term often refers to a board where a circuit can be quickly laid out and soldered). To build a prototype, you would require a written design, a *diagram*.

A **physical diagram** might be a photograph or detailed drawings depicting the physical structure of the components comprising a circuit. The physical diagram is suggestive of the physical layout of the circuit and attempts to leave little room for error for the novice experimenter. The most-concise method of providing a written guide is the circuit schematic.

A **circuit schematic** is an abstraction of a circuit that generalizes the specific components as symbols. The circuit schematic does not necessarily suggest the physical locations of the components as they may be physically arranged in the final prototype. There is, however, a one-to-one relationship between the components described in the circuit schematic, the physical diagram, and the prototype. It is important that an aspiring engineer learn to map one representation to another!

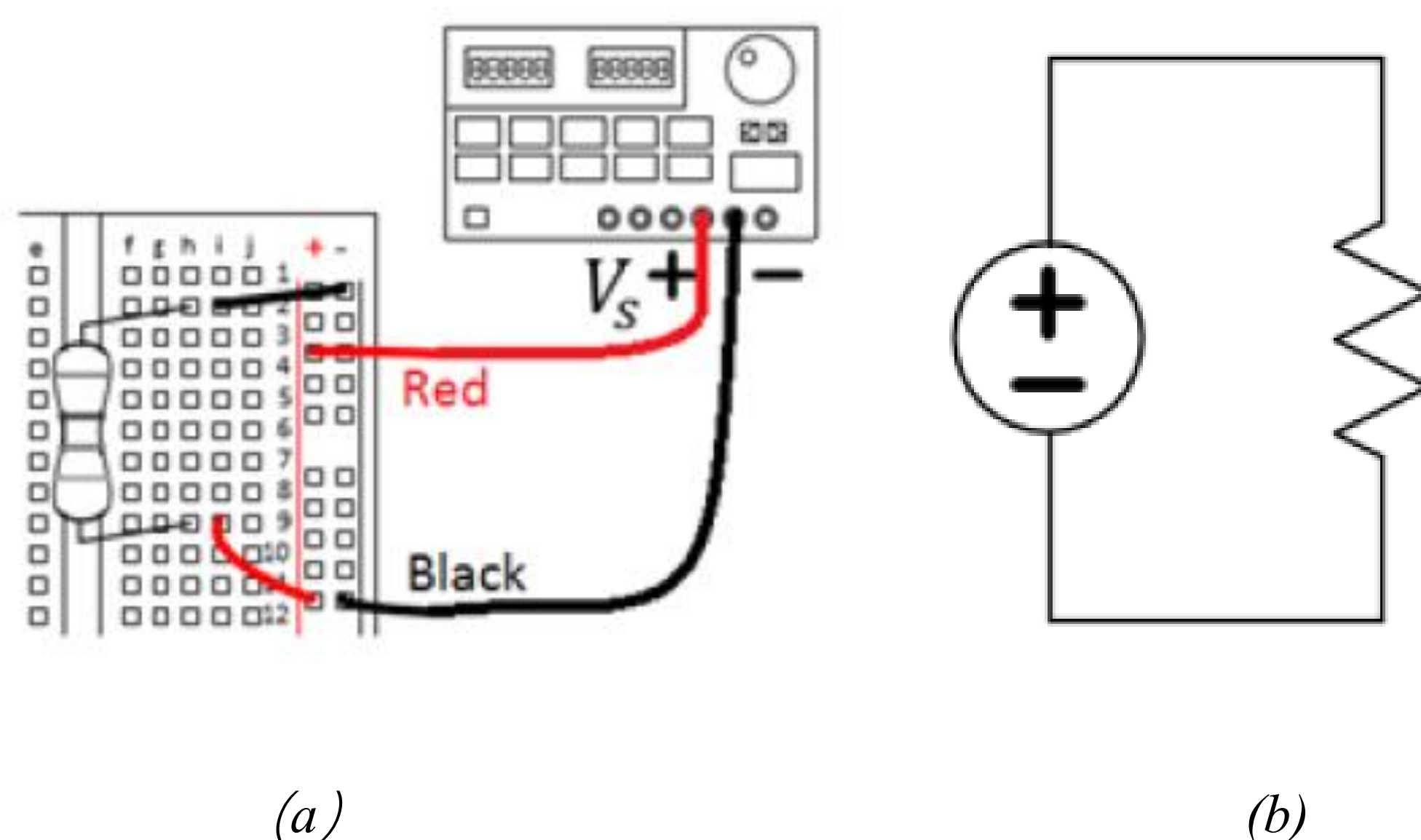


Figure 2: A physical diagram (a) and the more-abstract circuit schematic (b) for the same circuit.

In lab, you will build simple circuits using the equipment at your bench. It is important that you learn to read a circuit schematic and use it to build a physical prototype of the circuit. Sometimes, this task can be more difficult than you would think, especially when there are many components, several test points to measure, and wires going everywhere in what may appear to be a jumbled mess! If you can learn to properly interpret physical diagrams and circuit schematics, this task will become much easier for you.

Question 3: In lab 3, we will add fine-tuned speed control to the car. To do this, we will be using a switch to add two 47Ω resistors in parallel with each network of 100Ω resistors. Complete the circuit schematic for the physical diagram of Figure 3 (a). HINT: You will be adding two motors and ten resistors to complete the circuit schematic of Figure 3 (b).

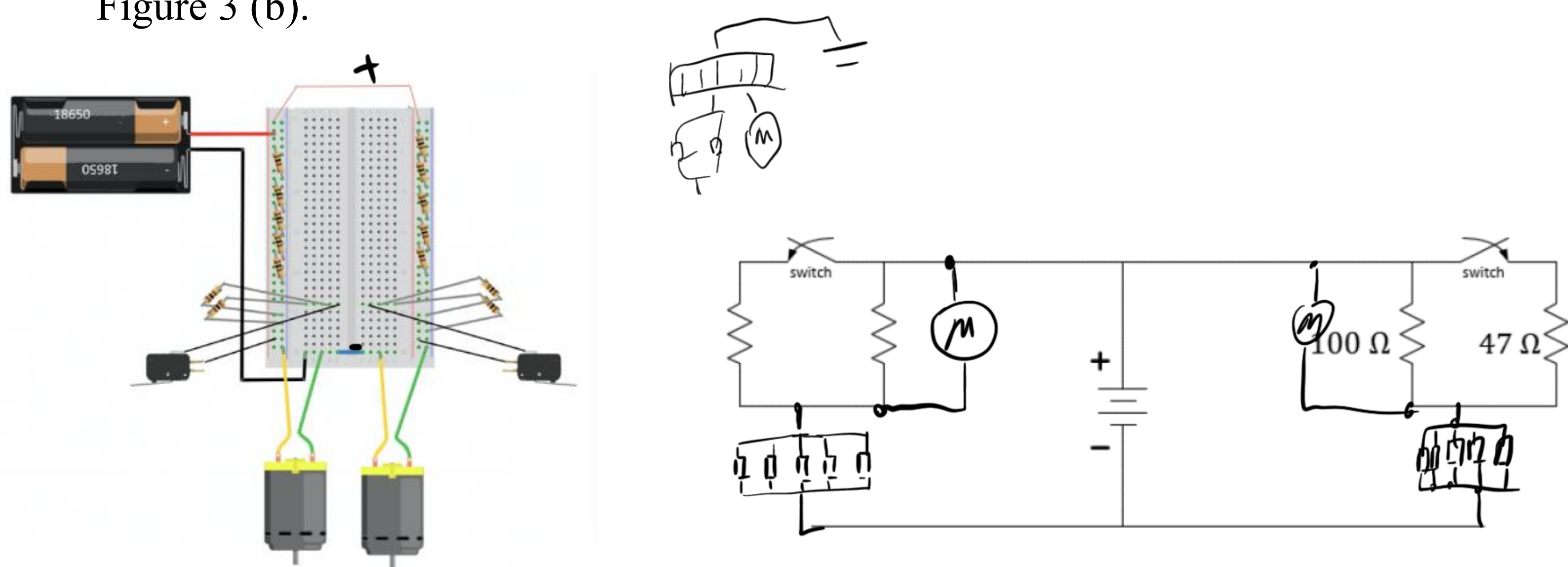


Figure 3: A physical diagram (a) and the circuit schematic *to be completed by the student* (b) for the same circuit

Analog Circuit Design and Simulation Tools

The improvement of electronic circuit design and development ability is inseparable from the development of electronic circuit simulation technology and electronic design automation technology. Circuit simulation technology is an engineering method that uses mathematical models to simulate the real behavior of electronic circuits. Electronic design automation technology is a technology formed in the process of applying computer technology to electronic design. It has been widely used in electronic circuit design and simulation, integrated circuit layout design, printed circuit board design and programmable device programming.

The design of analog circuit system requires a detailed analysis of the relationship between voltage and current of some circuits in the system. At this time, it is necessary to do transistor pole or circuit level simulation. The circuit model used in the simulation algorithm is the most basic component. Kirchhoff's current law (KCL) and Kirchhoff's voltage law (KVL) equations are established to calculate the voltage and current of each node in accordance with time.

The world's first software for analog circuit simulation, Spice (simulation program with integrated circuit emphasis), was developed in 1972 by the computer Aided Design Group at the University of California, Berkeley using FORTRAN language. In 1975 launched the official practical version, 1988 was designated as the United States national industrial standard, mainly used for IC, analog circuits, digital - analog mixed circuits, power circuits and other electronic system design and simulation. Spice emulation is a completely open policy, allowing users to modify it as they wish. Due to its practicality, SPICE emulation has been rapidly popularized and ported to multiple operating system platforms. SPICE comes in many “flavors” like HSPICE, PSpice, ngspice, and many others, all of which perform the same front-end function of circuit simulation, but with different “behind-the scenes” optimizations. LTspice is a free-to-use SPICE simulator created by Linear Technologies (now part of Analog Devices), and we’ll be using it for this class.

Where to Download LTspice

Next, we will learn how to use LTspice for circuit simulation and analysis. You can download the software by its official website: <https://www.analog.com/cn/design-center/design-tools-and-calculators/ltspice-simulator.html>. After install there will be an

icon  shown on your desktop.

Getting Start

For LTspice, while you can click through all the menus you like, sometimes knowing the keyboard shortcuts are faster. For the detail information, see appendix A.

Double click the icon to open it. You’ll see upon starting up:

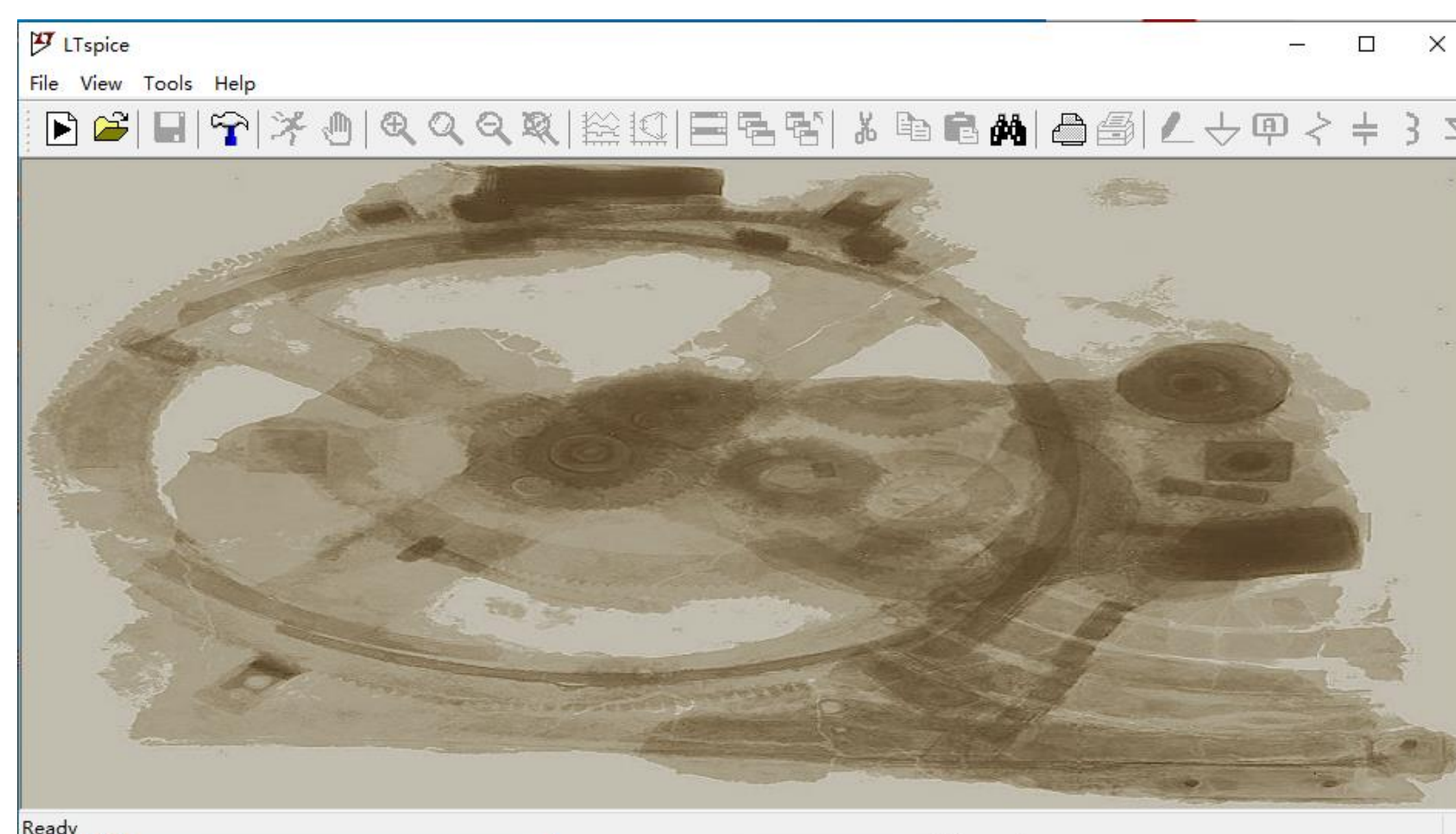


Figure 4: LTspice startup screen.

Let's get started by creating a new schematic. Go to File-> New Schematic, and the coffee stains should disappear, leaving you with a fresh page. Let's make our first circuit, show in Figure 5. Please follow the step below:

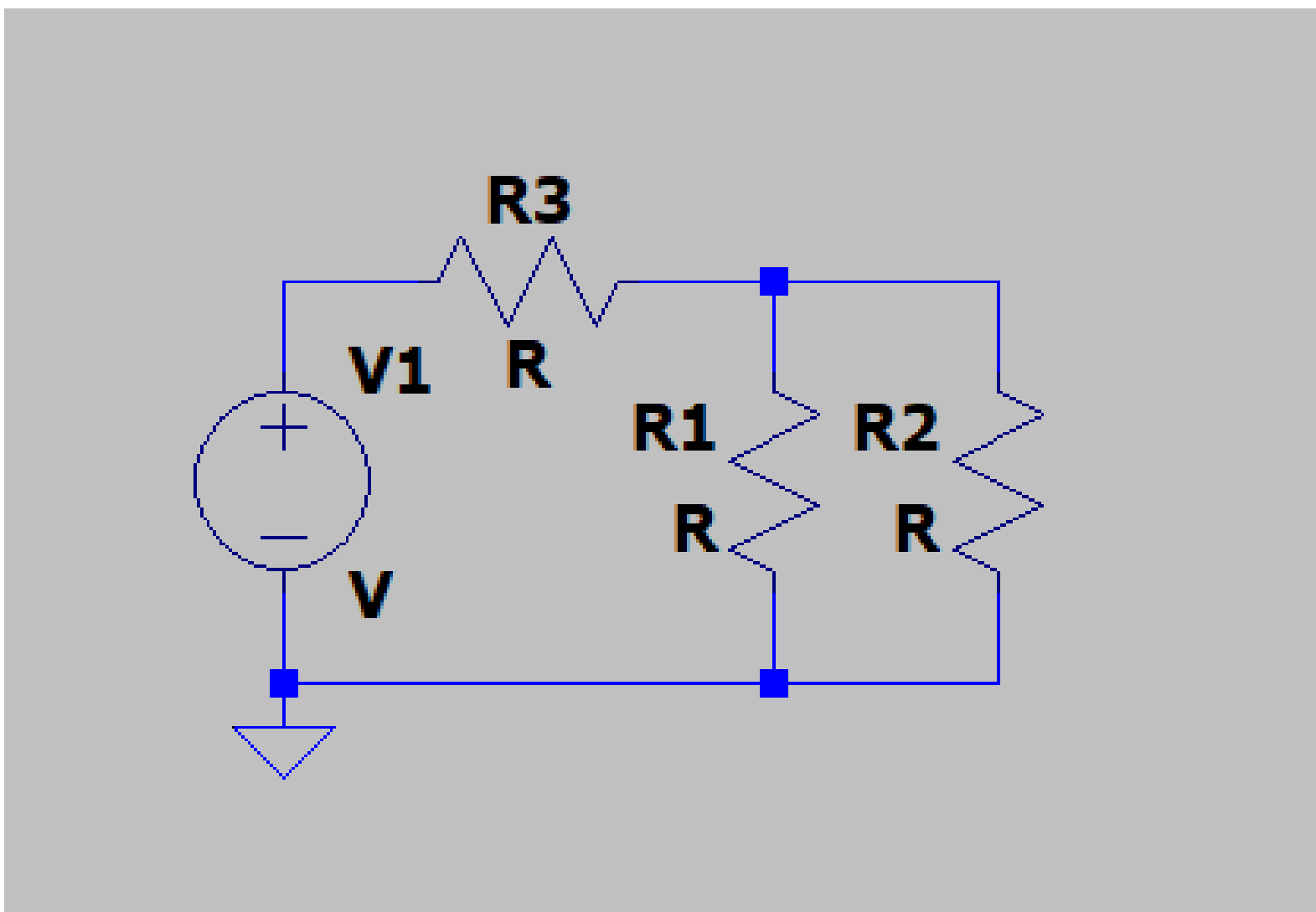
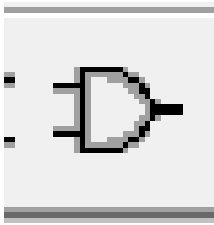


Figure 5: Basic resistor schematic

1. To add the voltage source, press the AND gate Symbol , or right click on the blank area, choose **Draft-> Component**, and search for the voltage part. You should get the following menu.

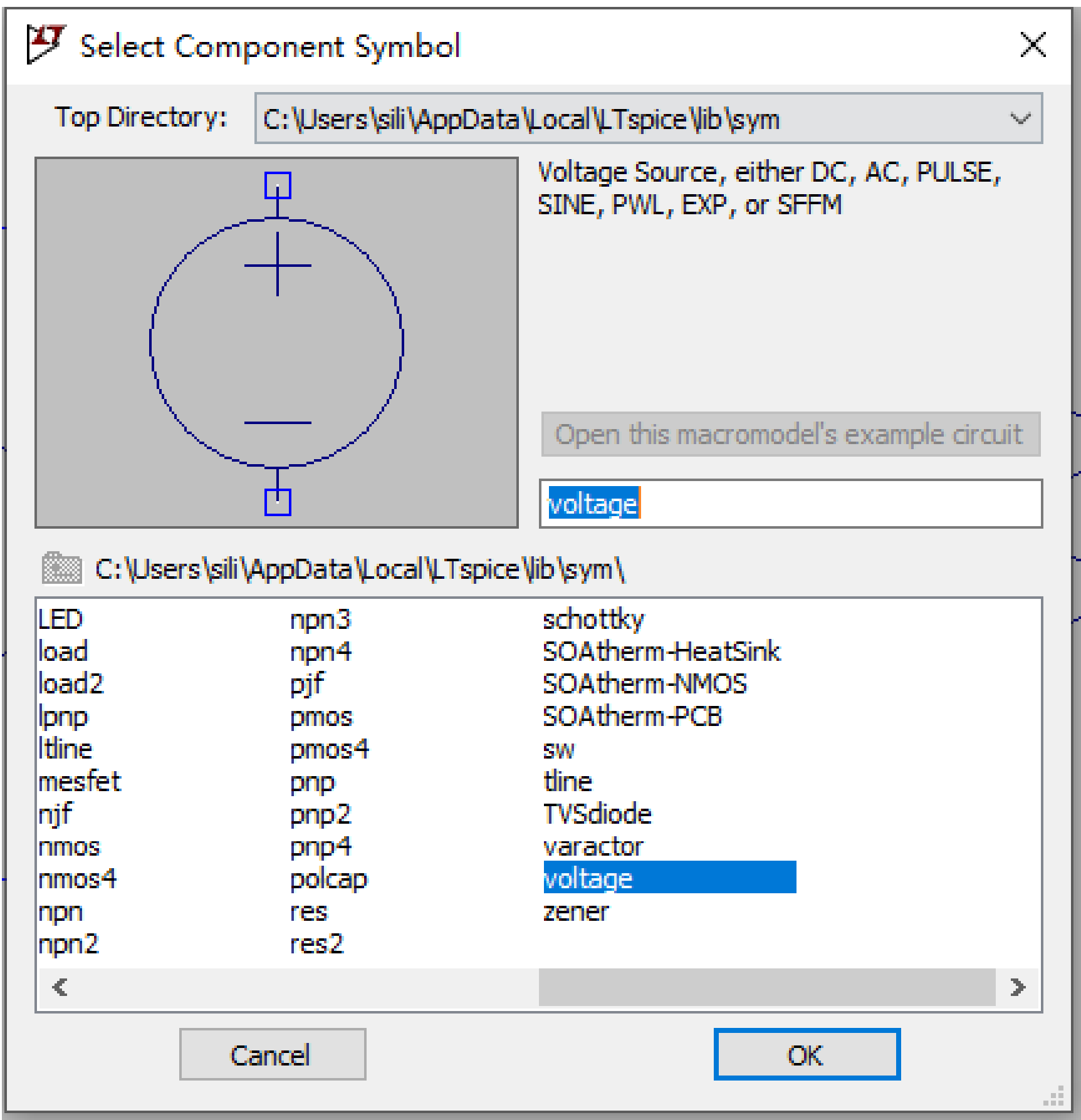
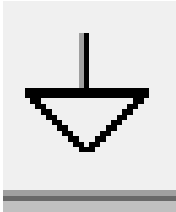
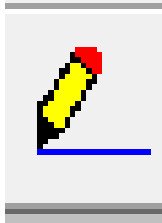


Figure 6: Adding component window.

2. To add resistor, also click , search for res, and draw on the schematic, or

directly click  to choose resistor.

3. To add GND, click  on the menu, and connect all these component by

wires(click  to add wires).

4. After draw the schematic, we can assign values to all the parts by right-clicking on them. Note LTSpice has two useful features: first, it auto assigns the units for us, so we can say the resistance is 50 or 50Ω, it doesn't care. Second, it implements the SI prefixes, so we can say 4.7n instead of 0.0000000047. However, it is case-insensitive. 10mΩ is 10 milli-Ω, and 10MegΩ is 10 Mega-Ω, Just putting 10M would be interpreted as milli- prefix, so be careful. For the unit, you can refer to Table 1. AS an example, we chose R1= 2kΩ, R2= 2kΩ, and R3= 1kΩ. Note we can also choose a specific “real-life” model, which will come in handy when using active devices.

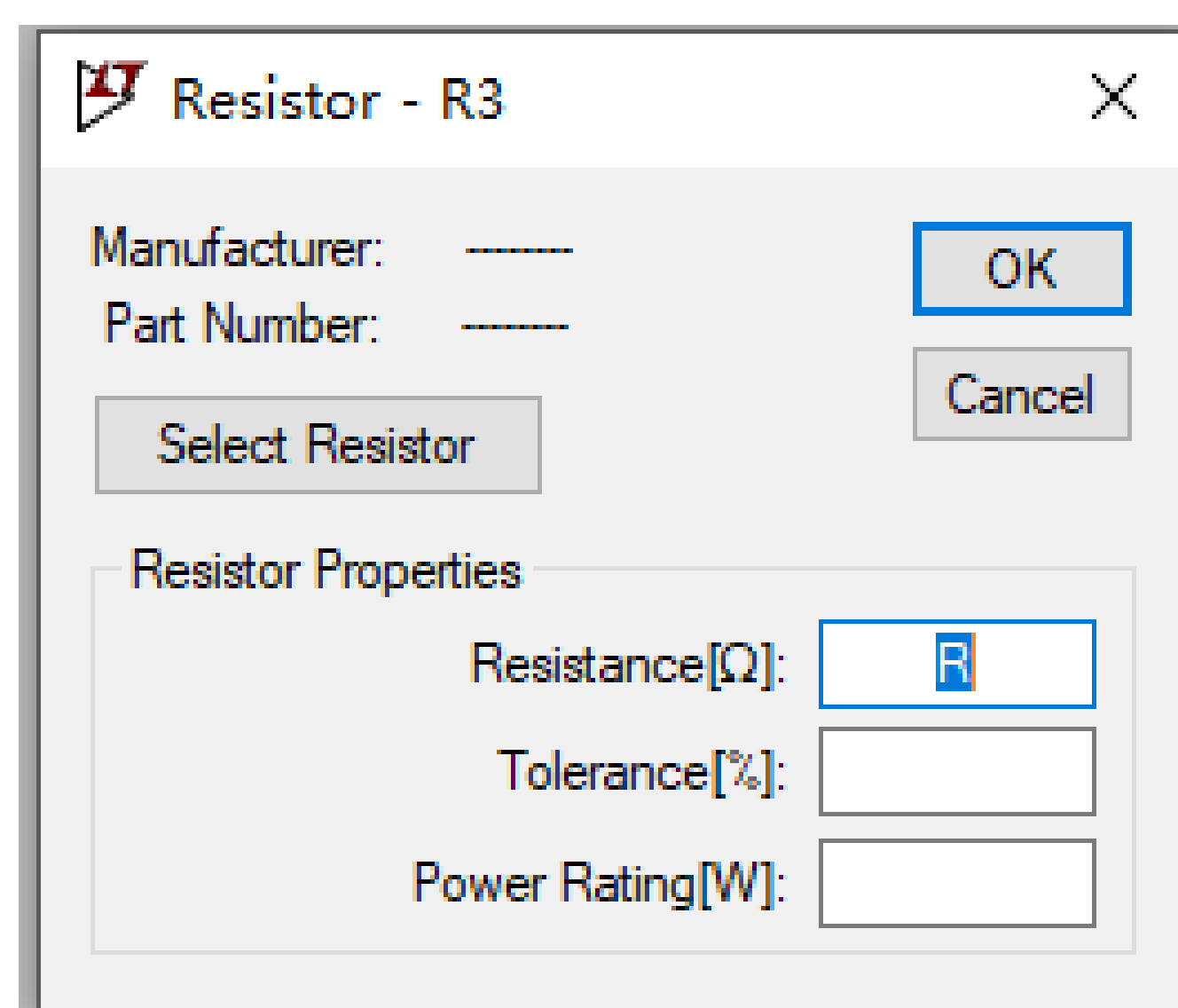


Figure 7: Editing components window

Unit symbol	Meaning
T or t	Tera= 10^{12}
G or g	Giga= 10^9
MEG or meg	Meg= 10^6
K or k	Kilo= 10^3
M or m	Milli= 10^{-3}
U or u (LTspice will use μ to replace it)	Micro= 10^{-6}
N or n	Nano= 10^{-9}
P or p	Pico= 10^{-12}
F or f	Femto= 10^{-15}

Table 1: Unit symbol in LTspice

5. Next, we will set V1. Right-click on it and choose *advance*. Copy the value show in Figure 8. The different function choices are:

(none) – Just a constant DC output

PULSE – Square wave output (series of pulses)

SINE – Sinusoidal wave

EXP – Exponentially decaying/rising wave

SFFM – Single frequency FM wave

PWL – Piece-wise linear wave described by a set of (x, y) point

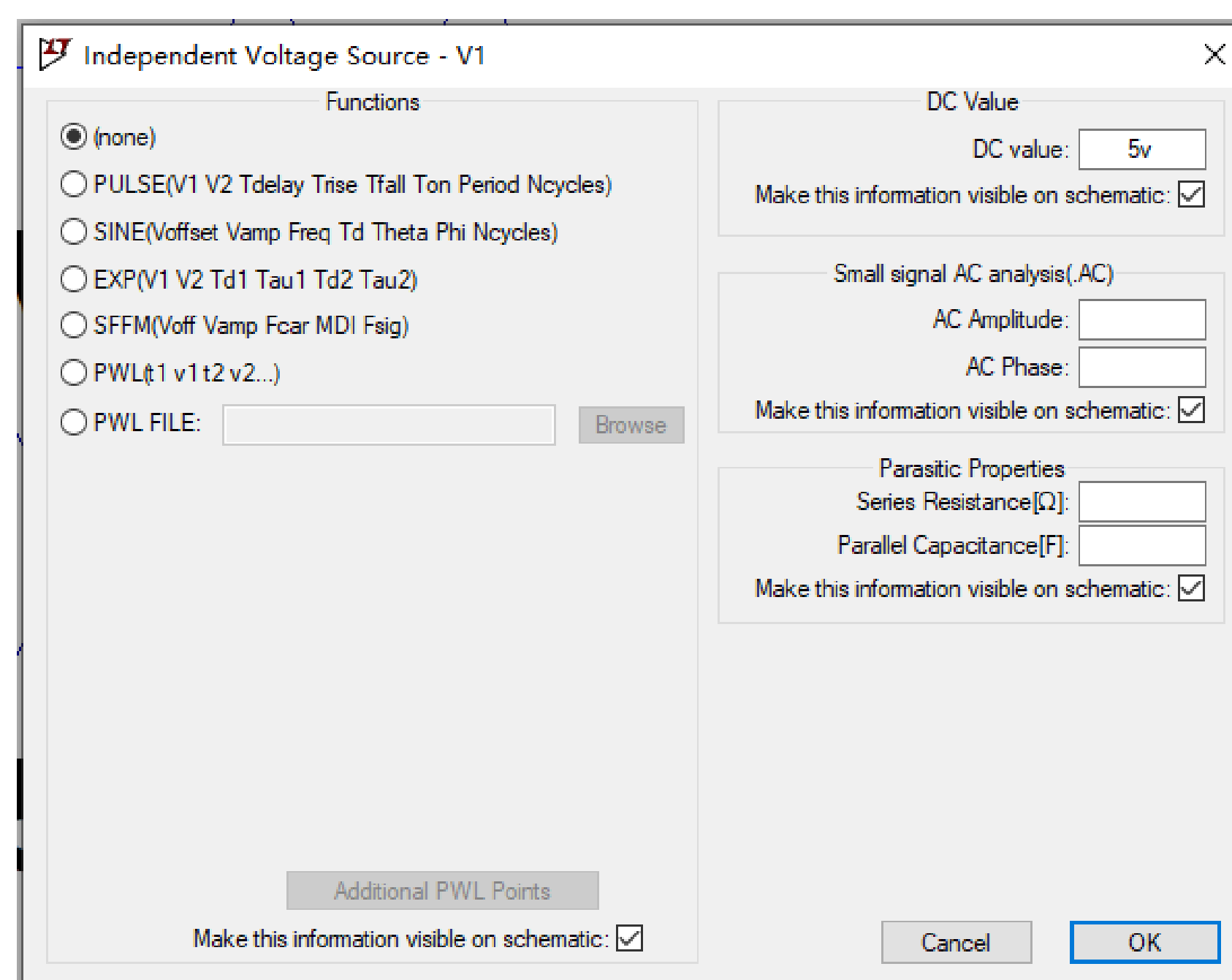


Figure 8: V1 settings

6. LTspice doesn't work with our schematic directly, but instead compiles it down to a netlist. You can view the netlist by going to View->SPICE Netlist. Our example netlist is:

```
V1 N001 0 5v
R1 0 N002 2k
R2 0 N002 2k
R3 N002 N001 1k
```

Each line can be read as: [Device Type] [Device Name] [Connection nodes] [Parameters]. The device type is selected based on its first letter, so all resistors must be called R<name>, similarly all capacitors are named C<name>. Based on the device type, SPICE will then read the next N “words” as connections. The remaining “words” are treated as parameters.

7. Simulation: LTspice has five total simulation types: transient, AC, DC, noise, TF, and OP. We'll only elaborate on the common ones here, more information about the other two can be found online. To add a simulation type, go to Simulate->Edit Simulation Command. Once you have created one, you can right-click on the command to bring the menu back up. The simulation commands appear as text on the schematic, in the exact same way it would in a netlist. You can run a simulation by pressing the “running man” icon, or pressing Ctrl+B. Note only one simulation can be run at a time, but you can have several simulation commands at once. LTspice will simply ask you when simulation you want to run. While a source can have multiple simulation properties (e.g., a voltage source can output a PULSE in transient and a sine wave in AC), only the parameters relevant to the simulation will be used.

- a) In this session, we use DC simulation to see the V-I curve characteristic, click *Simulation->Edit Simulation Cmd*, then choose *DC sweep*.

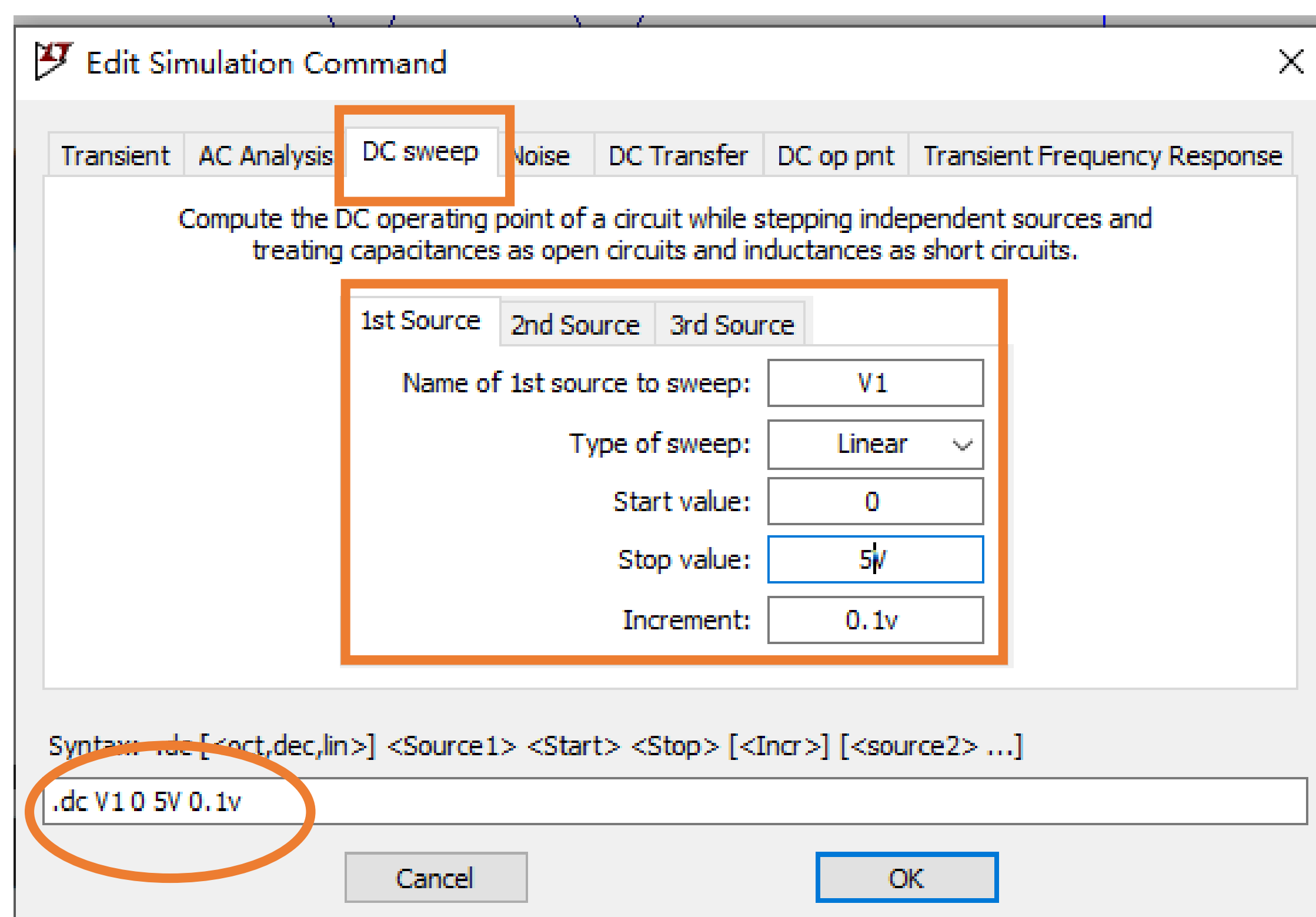


Figure 9: DC simulation settings.

- b) Fill in the following DC-Sweep-entries for the voltage source V1:

Start value = 0 Volts

Stop value = 5 Volts

Increment = 1 Volt

Type of sweep: Linear

- c) Check the so generated simulation command: “.dc V1 0 5V 0.1V” in the lower part of window as show in Figure 9

- d) Close with OK. Add two output labels. Click Edit->Label Net, Fill in the setting like show in Figure 10. Connect V1 Output with the positive end of Voltage source. Repeat the step to add V2 Output and connect with the node between R_1 and R_3 . After add the node the schematic should looks like Figure 11.

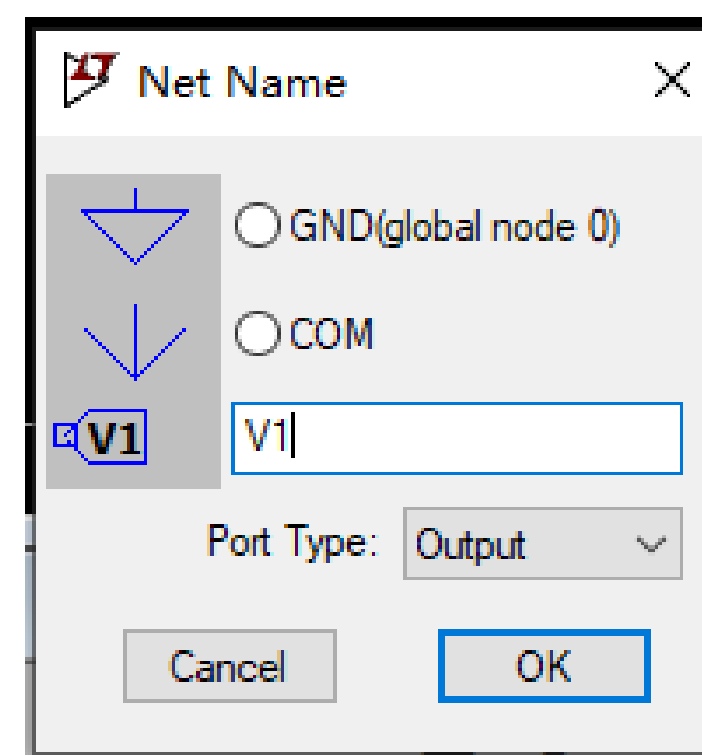


Figure 10: Label Net setting (Output)

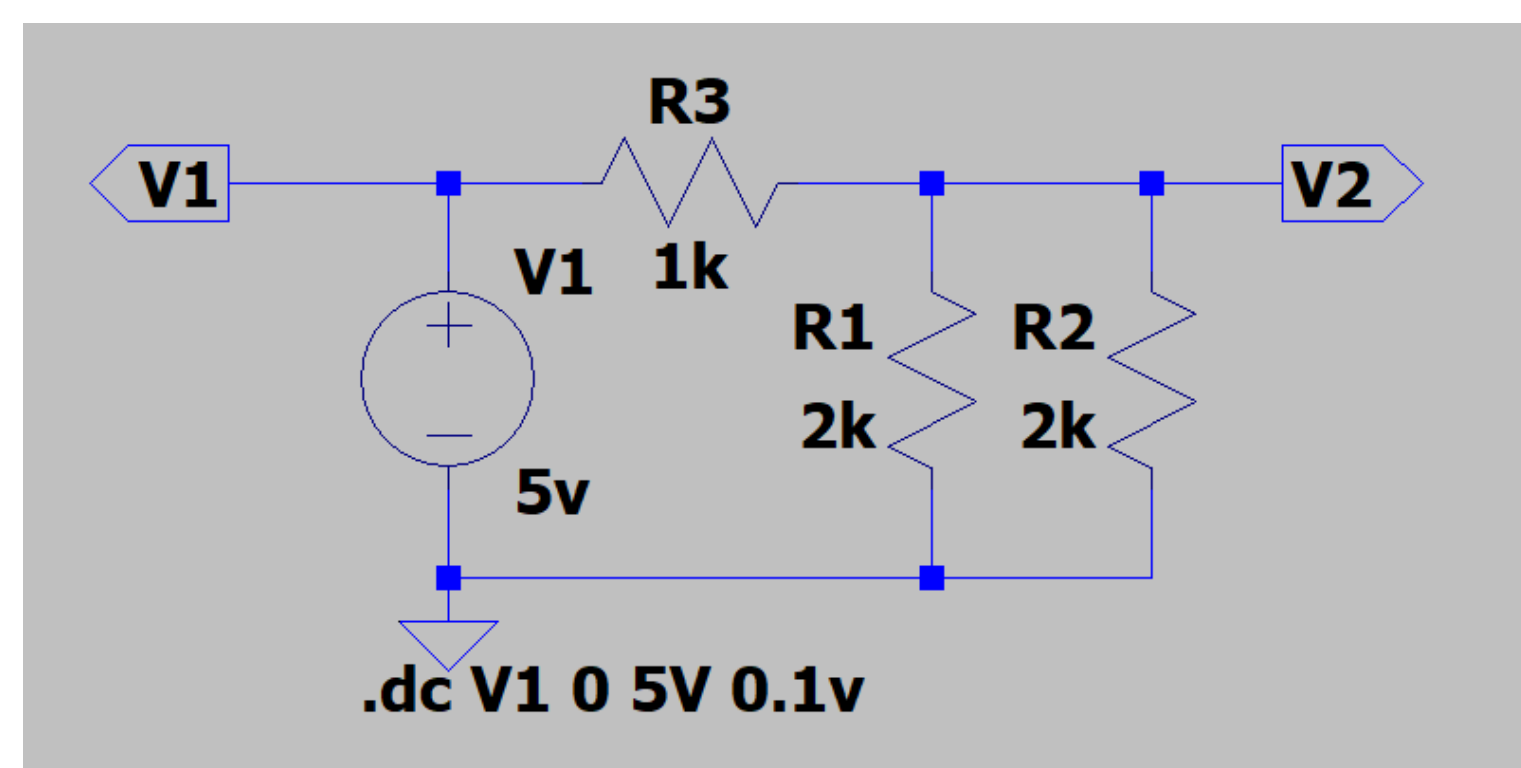
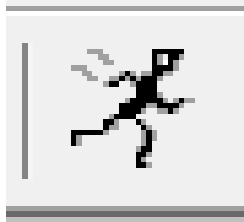
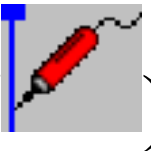


Figure 11: Schematic after add output label

- e) Then start simulation. Click  to start the simulation. Move the cursor to the label V1. When the curse suddenly looks like a “voltage sensor()”, left click and you get screen like Figure 12.

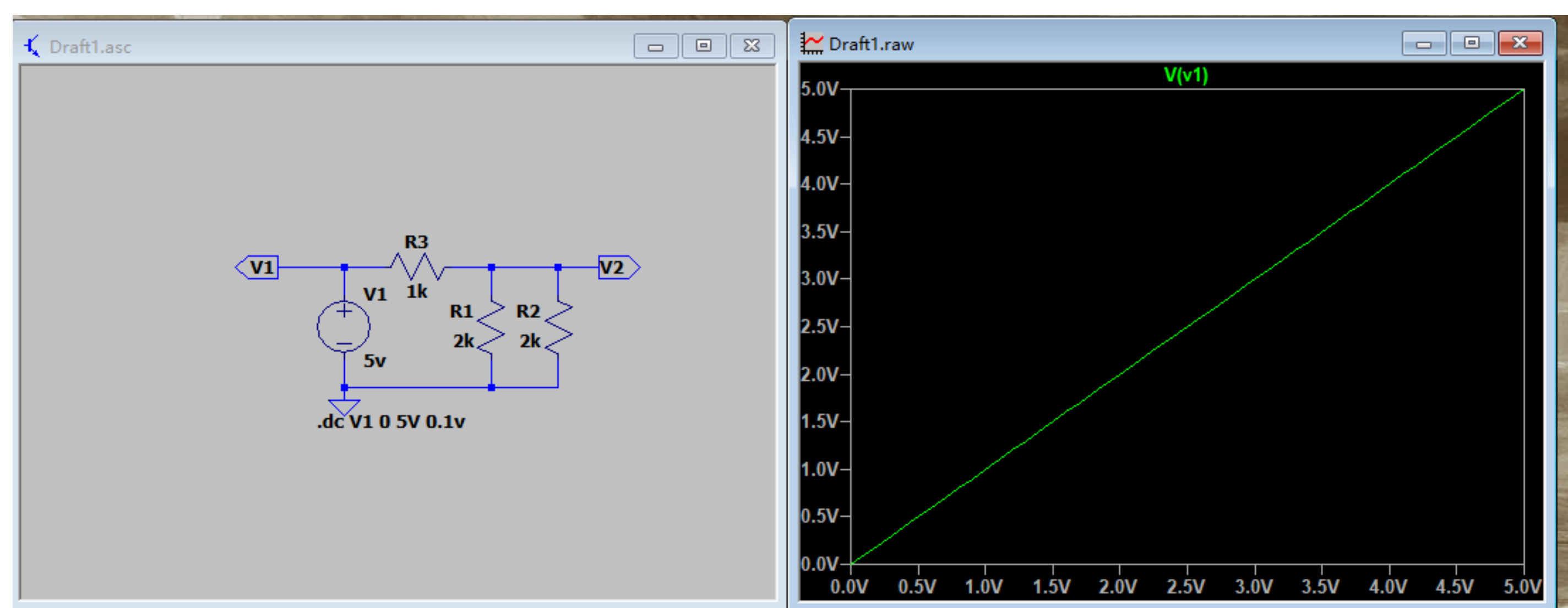


Figure 12: DC simulation example

- f) Use Cursor to read the specific value in the simulation figure. Add a cursor by left clicking on the trace name, above the plot. You can add two cursor measures (x_1, y_1) , (x_2, y_2) , and $(\Delta x, \Delta y)$. To move cursors, use left and right arrows, or click and drag. As show in Figure 13.

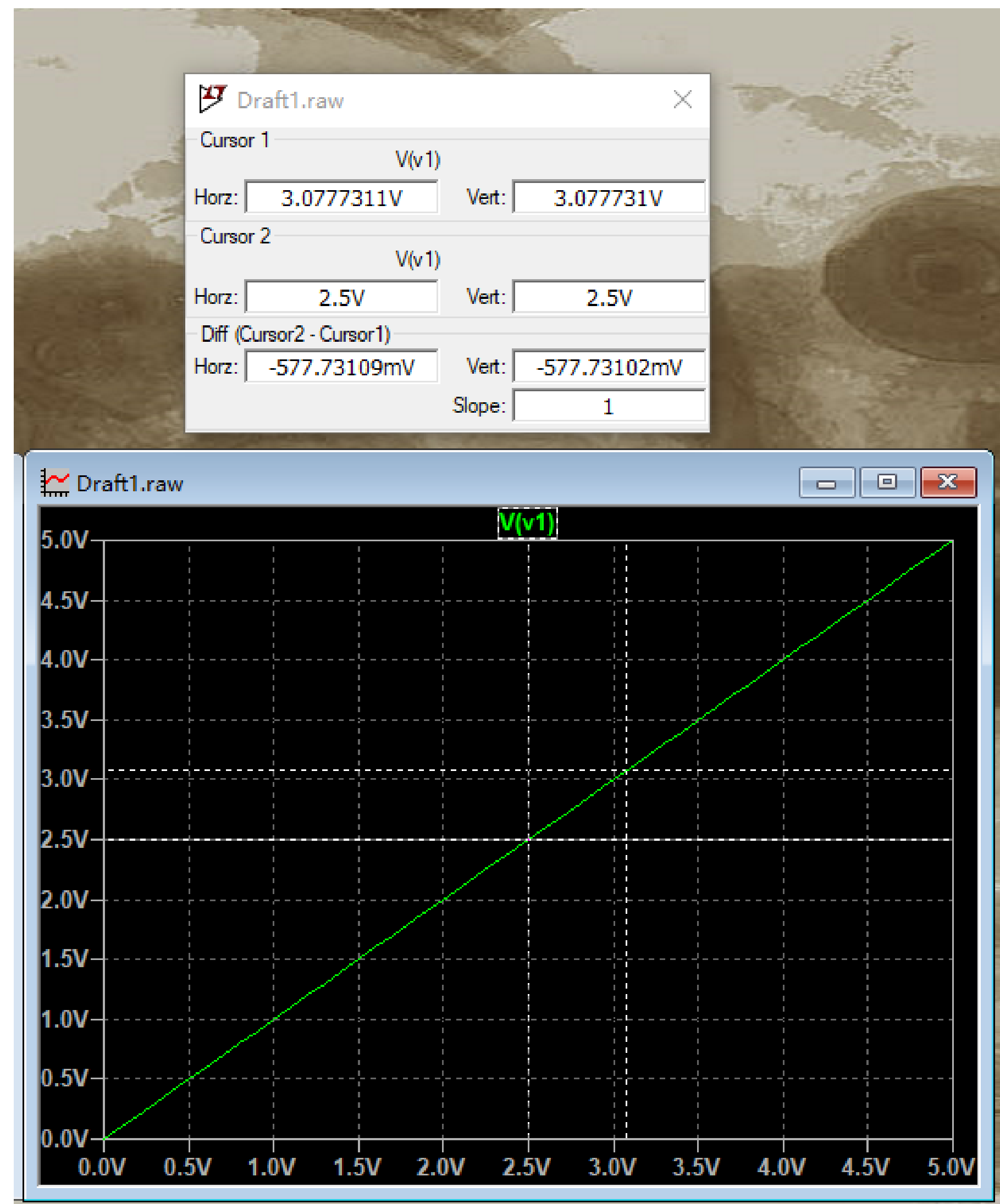


Figure 13: Curse example in a DC simulation.

Question 4: Add voltage trace for V2 and use curse to read the value of V1, and V2, Fill in Table 2. Also attach the figure you simulate in the end of this prelab.

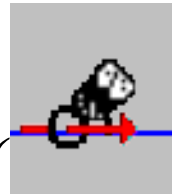
Voltage (V, x-axis)	V1(volts)	V2(volts)	Comment
1V	1 V	0.5 V	
1.5V	1.5 V	0.75 V	
2V	2 V	1 V	
3V	3 V	1.5 V	
4V	4 V	2 V	
5V	5 V	2.5 V	

Table 2: Simulation Result-voltage

Question 5: Can you find the relationship between V1 and V2, is the relationship what you expected? Why?

$$V_1 = 2V_2 \quad R_3 = R_1 + R_2$$

g) Now let's do some simulation for the current, use the same setting as above, run the simulation, and Move the cursor to the wire connect with R3, **Press**

“Alt”, when the curse suddenly looks like a “Current sensor ()” left click and you will get the simulation figure, repeat this for R1.

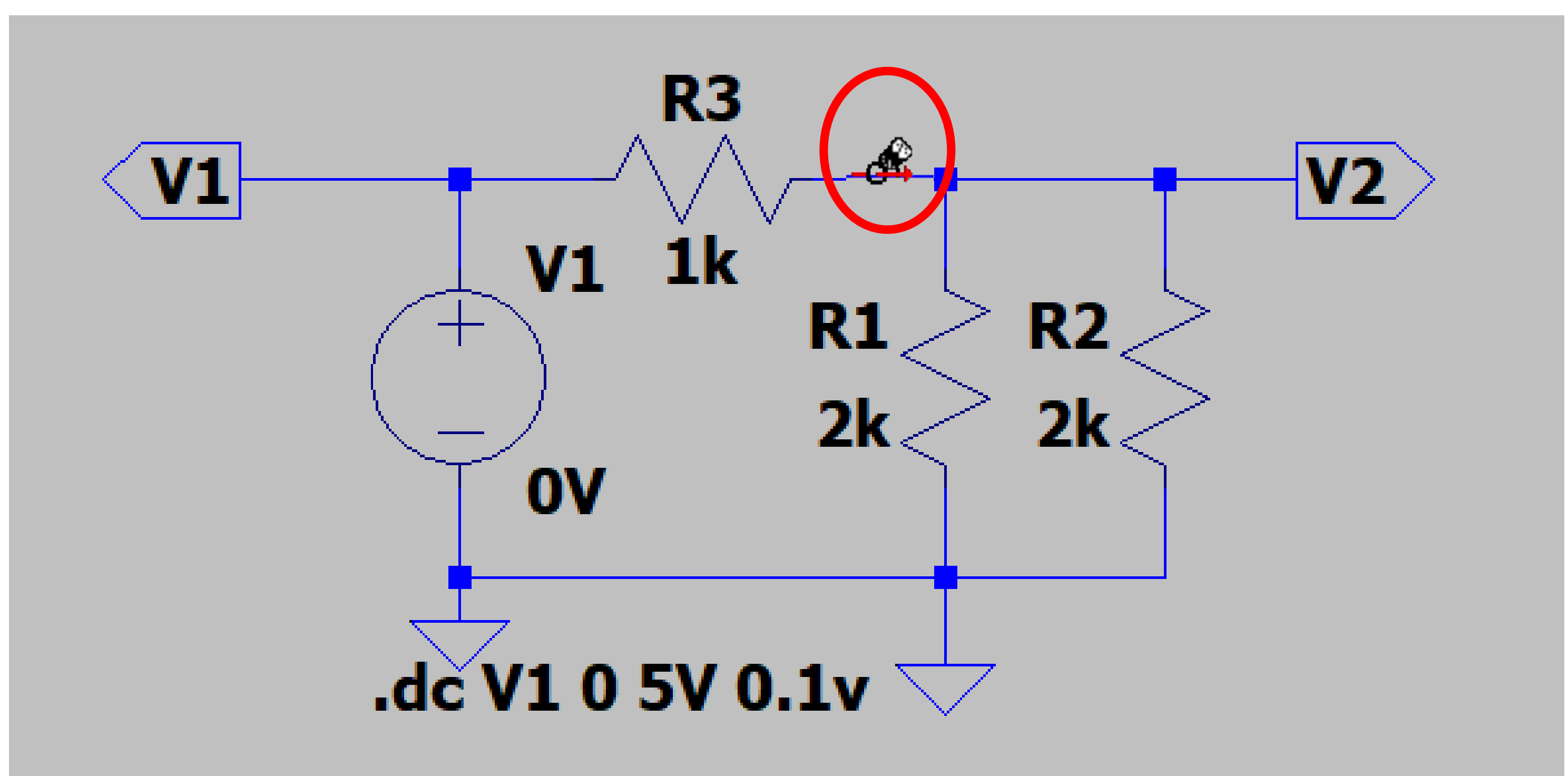


Figure 14: How to get current result in the simulation

Question 6: Use curse to read the value, fill in the Table 3.

Voltage (V, x-axis)	Current through R3(mA)	Current through R1(mA)	Comment
1V	0.5	0.25	
1.5V	0.75	0.375	
2V	1	0.5	
3V	1.5	0.75	
4V	2	1	
5V	2.5	1.25	

Table 3: Simulation Result-current

Question7: Can you find the relationship between the current through R1 and R3, is the relationship what you expected? Why?

$$I_{R_3} = 2I_{R_1}$$

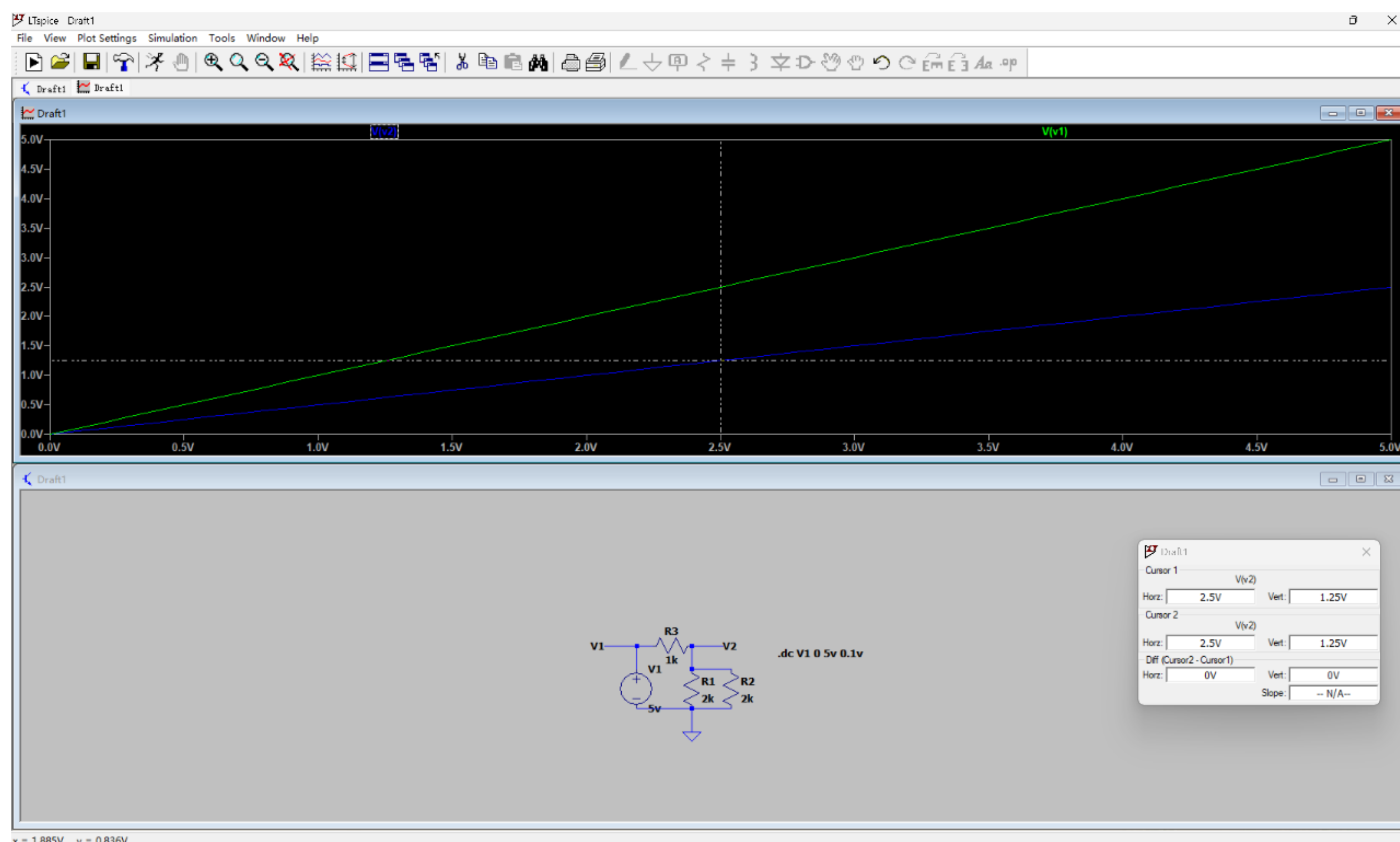
$$\therefore V_{R_3} = V_{R_2}$$

$$R_3 = \frac{1}{2}R_2$$

































































$$I = \frac{V}{R}$$

$$\therefore I_{R_3} = 2I_{R_2}$$

(Optional)Question 8: You will find that you may get a negative value by directly click on the resistors(e.g. R3), can you find the reason why it's a negative value?



Appendix A:

COMMANDS		SHORTCUTS			
SPICE Analysis		Schematic and Symbol Editing Modes		Place Component Modes*	
.OP	find the DC operating point	Choose Mode then select component Exit mode: Press [Esc] or right-click		Press [Esc] or right-click to exit place component mode	
.TRAN	perform nonlinear transient analysis				
.AC	perform small signal AC analysis	[F5] or [Delete] or [Ctrl][X]		[R]	
.DC	perform DC source sweep analysis	[F6] or [Ctrl][C]		[C]	
.TF	find the DC small-signal transfer function	[F7]		[L]	
.NOISE	perform noise analysis	[F8]		[D]	
SPICE Directives		[Esc]		[G]	
.BACKANNO	annotate subcircuit pin names on port currents	Zoom and Grid		[V]	
.END	end of netlist			[S]	
.ENDS	end of subcircuit definition	[Ctrl][Z]		[T]	
.FOUR	compute fourier component	[Ctrl][B]		[F2]	
.FUNC	user defined functions	[Space]		[F3]	
.FERRET	download a file from URL	[Ctrl][E]		[F4]	
.GLOBAL	declare global nodes	[Ctrl][G]			
.JC	set initial conditions	TRICKS		*Rotate and Mirror	
.INCLUDE	include file	Waveforms			
.LIB	include library			[Ctrl][R]	
.LOADBIAS	load a previously solved DC solution	[Alt] click		[Ctrl][E]	
.MACHINE	arbitrary state machine	[Ctrl] click		Undo/Redo	
.MEASURE	evaluate user-defined electrical quantities	Schematics			
.MODEL	define a SPICE model			[F9]	
.NET	compute network parameters in .AC analysis	hold [Ctrl]			
.NODESET	supply hints for initial DC solution	[Ctrl][Alt][Shift][H]		[F9] or [Ctrl][Z]	
.OPTIONS	set simulator options	NUMBERS			
.PARAM	user-defined parameters	Prefixes (Case Insensitive)		[F9] or [Ctrl][Z]	
.SAVE	limit the quantity of saved data			Constants	
.SAVEBIAS	save operating point to disk	[Alt] click		[Ctrl][R]	
.STEP	parameter sweeps	[Alt] click		[Ctrl][E]	
.SUBCKT	define a subcircuit	[Ctrl] click		### Levels of Undo	
.TEMP	temperature sweeps			[F9]	
.TEXT	user-defined string	hold [Ctrl]			
.WAVE	write selected nodes to a .WAV file	[Ctrl][Alt][Shift][H]		[F9] or [Ctrl][Z]	
any text preceded by an underscore, e.g. "_FAULT" is displayed with an overbar, active low, signal		Tera		[F9] or [Ctrl][Z]	