Colin May

Passive Circuit Analysis with LTspice®

An Interactive Approach

MOREMEDIA

Springer

Passive Circuit Analysis with LTspice®

Colin May

Passive Circuit Analysis with LTspice®

An Interactive Approach

Springer

Colin May (retired)
University of Westminster
London, UK

Simulation files can be found at: https://www.springer.com/us/book/9783030383039. In order to use the simulation files with the extension '.asc' referenced in the text it is necessary to download the free programme LTspice. The simulation files are grouped in folders for each chapter. These can be downloaded into an appropriate enclosing folder. The essential files in each are the schematic '.ASC' files which can be run by LTspice. Some schematics rely on symbols and sub-circuits created in the text. Most (if not all) can be found in the folders 'Mysym' and 'Mysubs'. These can be downloaded and included in the LTspice search paths. Additionally, some schematics, especially chapter 2, use commercial sub-circuits and symbols provided by device manufacturers. These too can be downloaded and must be included in the search paths. Some additional files are also included, but not for every schematic. The '.NET' files are netlist files that can be run by LTspice instead of the '.ASC' file but these do not show a circuit schematic. The '.PLT' files are pre-determined settings for how the traces should be drawn and also annotations to the trace. They are not executable files.

ISBN 978-3-030-38303-9 ISBN 978-3-030-38304-6 (eBook)
https://doi.org/10.1007/978-3-030-38304-6

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

In Memoriam
Ελὲνη Αθανασοπούλου
(1979–2019)
‘Σε ἐναν ‘Αγγελο...’

Eleni Athanasopoulou
She was taken too early

Foreword

This book is idiosyncratic in that it includes oddities not included in the general run of things, such as the Hamon voltage divider, the Murray loop test, the Elmore delay and the Thiele-Small loudspeaker model. These are applications of passive circuits which were encountered during countless trawls through the Internet and seemed both ingenious and interesting. But it is hoped that the text also encompasses the usual range of circuits.

SPICE is an analytical tool, just as manual analysis is, a very powerful tool that can return far more accurate results than are generally possible with pencil-and-paper analysis because it can include nonlinearities and temperature effects, but still it is just an analytical tool. At present, it is not possible to insert the required system performance and for SPICE to create an appropriate circuit: this, perhaps, is a project for Mike Engelhardt. So LTspice does not absolve the circuit designer from the onerous task of designing the circuit – it will only report how the circuit performs.

We must also answer the question "Why bother using SPICE?" To this are our several answers. Granted that it takes a little time to learn the basics – and a bit more to become proficient – it will be found afterwards that circuits can be simulated and tested faster than is possible with manual calculations. But does this mean that prototyping on the workbench is redundant? Far from it. LTspice will only analyse the data it is given: so stray capacitance and inductance that bedevil practical circuits will only come to light on the real physical circuit. On the other hand, the exigences of health and safety considerations, as well as the time needed to construct prototypes, make LTspice an attractive adjunct where component values can be changed and tested in seconds. And it is also invaluable in one other important area which is often overlooked – production yield. Components are not exact but have tolerances. It is clearly unreasonable to build a few thousand actual circuits to see which are out of specifications, but no problem with LTspice – even if it means leaving it to run all night.

In the early days, that is, pre-millenium, some versions of SPICE were prone to unfortunate errors; and although it is not something germane in this text, in one version at least, it was possible to mistakenly connect an opamp intended as a unity gain buffer, with the feedback taken to the positive input instead of the negative, and to find that the circuit still worked as a buffer. Nowadays, such errors have been fixed, but still the user must beware of false results.

These still occur and originate from two sources; one is the default settings of LTspice. These are accessed through **Tools- > Control Panel**, and there are three of importance: the first is that by default the results are compressed; in most cases, this is very useful since it greatly reduces the size of the simulation file – the *raw* file – by a factor of at least ten. However, it does mean that there is imprecision in measuring sharp resonant peaks or transitions, and then it must be turned off. The second is that by default, inductors have a parallel damping resistor and a small series resistance: it is this latter that can lead to false reductions in the output of a tuned circuit. The third area of concern is the rise and fall times of pulse input waveforms where the default of zero has significant slopes; setting each to something like *1e-15* results in vertical edges, but it is always worth checking. Otherwise, the default settings have been used throughout this book, only needing the Gear method to resolve a rather awkward example of the definition of a capacitor by its charge in Chap. 7, nor generally adjusting the settings in **Tools** → **Control Panel** → **SPICE**, although tolerance changes were tried for one or two simulations that ran slowly or stopped, but there did not seem to be any improvement. This leads to a second consideration, and that although it is very unlikely that LTspice miscalculates the performance of a circuit, it is always a good idea to start with a simple circuit, without the complications of nonlinearities or temperature effects, and check that the results agree with calculation.

The second source of errors is in models and subcircuits. Models are predefined in SPICE and it is not within our power to add extra parameters. Therefore any shortcomings, such as the failure of bipolar transistor models to include the base-emitter breakdown, can only be dealt with by creating a subcircuit. These are another matter and are essential where the performance of a device such as a thermistor cannot adequately be replicated by an existing model, and it is therefore necessary to construct a circuit to create the correct characteristics. In some cases, there are agreed subcircuit structures, but not always, and the performance of the subcircuit may fall short of the actual device which we shall explore, especially in Chap. 3.

The structure of this book, as far as possible, is to start with simple circuits and basic analytical tools and then progress to more complex circuits and advanced analytical methods on the basis that it is more productive to move from the known to the unknown rather than the other way round. Also, an exhaustive listing of, say, the *.meas* statement is apt to be rather indigestible, and so its features are introduced as needed, scattered over several chapters. On the other hand, there seemed no alternative but to list, with illustrations, the enormous capabilities of the voltage and current sources as Chap. 5. Additionally, some topics such as RCL circuits are covered twice, but in different fashions.

Perforce, the text is essentially mathematical because LTspice makes a no-compromise analysis of the circuit, returning results correct to six or seven figures, including aspects of the circuit's performance that are not always covered in text books. This sometimes results in puzzling results prompting a “What the (insert expletive) is happening here?" These can often be the most fruitful simulations bringing to light side effects that were overlooked in a simple manual analysis (or were omitted due to an inadequate grasp of the theory).

Worked examples with accompanying simulations pervade the book, but there are no end-of-chapter lists of questions, just a few suggested explorations at the end each section. The hope is that the theory and examples will stimulate more investigations, which should be seen as a challenge, not an onerous burden. Indeed, if the thought of having to simulate a circuit brings a deep sigh and a heavy heart, it is respectfully asked if electronics is the right career – life is too short to spend one's working days doing something that does not bring deep satisfaction.

LTspice is under continuous development, so some simulations that caused problems now run smoothly, and for others the comments may not now be relevant. Also, a few circuit symbols have changed. And sources now require five rather than four parameters.

Inevitably, there are mistakes, for which I crave the readers' indulgence and a gentle smile rather than *schadenfreude*. To err is human, but for a really good foul up you need a computer. I hope I have avoided the opacity variously attributed to Robert Browning at https://www.google.com/search?client=firefox-b-d&q=only+god+and+browning.

Patra, Greece Colin May

Contents

Chapter 1
LTspice Essentials

1.1 Introduction

The aim of this book is to provide what, hopefully, is a useful and enjoyable introduction to LTspice mainly because the text is structured so that many simulations that go hand-in-hand with the analysis to reinforce and extend the topic are available from the website. For this reason, there are illustrative worked examples and suggested explorations, but no end-of-chapter questions since it is entirely possible to create a circuit, make the analysis and check the result by simulation. And this is open-ended leaving room for flights of fancy in circuit design. And because these are only simulations, a few hundred kV or a million amps or two will not bring down the wrath of the Health and Safety executive nor excite the fears of the laboratory staff. And under this later head, it must be emphasized that simulation is not a replacement for building and testing the actual circuit. Certainly there are cases where this is just not possible; the design of integrated circuits is the prime example. Otherwise, after the LTspice analysis, we need to build the circuit to find out if there were things we had forgotten in the simulation: stray capacitances, the internal resistance of a source, things like that.

And it must always be remembered that SPICE in whatever form is an analytical tool, a very powerful one, to be sure, but still only a means of analysis, and it does not absolve the circuit designer of the hard work of designing the circuit in the first place, although judicious use of 'what if?' simulations can often give helpful clues about the circuit's performance. In addition, by using SPICE, we can quickly find out more about a circuit's behaviour than would be economically possible to calculate. A very simple example is the optimum load resistor for maximum power transfer from a DC source. The calculation is not difficult, but it takes time to repeat the sums to find the seriousness of a mismatch. We have the answer in seconds with SPICE. And many circuits have start-up transients that are difficult to calculate and which are easily seen using SPICE. Many years ago, in the pre-SPICE

C. May, *Passive Circuit Analysis with LTspice®*,
https://doi.org/10.1007/978-3-030-38304-6_1

days, the start-up current surge of a main power supply could only be estimated using graphs. This was a serious matter because it could easily cause device failure if it was not limited.

But perhaps the greatest advantage of simulations is that of including the imperfections of practical components: inductors have resistance, capacitors also, and everything can change with temperature, not just linearly but with a quadratic or higher polynomial relationship, or even exponentially. These can give the LTspice user scope for ingenuity in finding ways to model these circuits. Many years ago, the late Bob Pease of Analog Devices created a SPICE shoot-out of a few circuits of no great complexity but extremely difficult to simulate to the required accuracy. CPUs have moved on since then, so perhaps now it is only time for a cup of coffee, but then it took an all-night run to get an answer. And this is another useful feature of SPICE; doing, for example, a statistical analysis to estimate the production yield means that we can get on with something useful and leave SPICE to crunch the numbers.

Those familiar with LTspice can safely skip this chapter, or perhaps skim the last section. Otherwise a good starting point to trace the interesting history and development of SPICE is: https://docs.easyeda.com/en/Simulation/Chapter14-Device-models/index.html.

The book, perforce, is mathematical. This is for two reasons: the first is that it is never safe blindly to accept the results of a simulation – there should at least be an approximate analysis that agrees in the main with the simulation. This is not to say that the simulation is faulty – although that is possible if the analysis is poorly set up – but rather as a check that the underlying concept is sound. The second reason is that LTspice will show second-order effects that are often glossed over: in particular, the resistance of the inductance in a tuned circuit and the effects of pulse excitation of a tuned circuit rather than a sinusoid. These will be exhaustively dealt with in an appropriate chapter.

There were two conflicting requirements in the layout of the book: one was to cover the gamut of LTspice's capabilities and the other to explore the range of passive circuits. Trying to cram, for example, all the 'Simulate' options into a chapter seemed rather indigestible and disruptive of an orderly flow of topics since some analyses are appropriate to DC and others not. Therefore, apart from the chapter on voltage and current sources, the approach has been to start with simple concepts and circuits – such as Ohm's Law in this chapter – and build to more complex circuits and analytical tools for DC, then to move to AC and capacitors and inductors and higher things. The result is that the LTspice commands, directives, call them what you will, are introduced as they are needed, and often only in part so that, for example, the *.meas* directive is scattered through four chapters.

The contents of this book are idiosyncratic. Certainly it is to be hoped that the analysis and applications of the most popular passive circuits have been covered somewhere, but there are also such diverse items as spark transmitters, loudspeakers and thermal modelling. In short, anything that looked interesting or unusual. Under this head comes the Hamon Potential Divider that creates a highly accurate division by 10 using not-so-close tolerance resistors, the Murray and Varley loop tests to find faults in cables and the Tapped Capacitor Impedance Divider to efficiently match a source to a load.

It should be mentioned that filters in one form or another appear in more than one chapter. It can be argued that simple RC circuits are filters of sorts. Therefore having discussed those, it is no great stretch to cascaded RC filters, although the ‘T’ and Bridged ‘T’ are somewhat more tricky. Likewise, having explored inductors, it is appropriate to deal with notch filters with inductors replacing capacitors. But it still seemed appropriate to create a special chapter on filters which deals with the higher-order RCL filters and more esoteric methods.

So now to the contents of this chapter. First we shall find out how to build a simple circuit using Ohm’s Law as a vehicle for exploring some of the most important properties of the programme including (of course) how to select and place components, how to assign values, how to edit them by rotating and flipping and even removing them. But first it is worth saying a little about drafting conventions.

Then having built the circuit, we can establish the DC conditions and view the results, using the flexible cosmetic abilities of LTspice to change colours, line thicknesses, and so on. Finally we can save the results in various formats.

But above all, simulation should be fun: it should excite curiosity. If you do not enjoy the intellectual challenge, perhaps you are in the wrong business.

1.2 Representing the Circuit

Information about a circuit is communicated through a circuit diagram using symbols to represent the physical components of the circuit. There are, however, two distinct applications of a circuit diagram

The first is to represent an actual physical circuit; typically it has a reference letter and number for each component as well as it its value, for example, *Rs 56k 5%*. The symbols therefore represent the physical devices which may not be ideal: for example, a resistor may also have some inductance. These diagrams often contain test point waveforms or voltage or current measurements and are used by service and repair staff. In passing, it should be noted that the size of the symbols is of no significance – a 100 ohm resistor is not drawn larger than a 10 ohm one – nor is the relative positioning of the symbols on the sheet of paper any guide to the position of the actual component on a printed circuit board.

The second, and perhaps the most common use, is in circuit design and analysis. Here, the symbols are understood to represent ideal components: resistors only have resistance; likewise wires are perfect conductors and possess neither resistance nor inductance. If it is necessary to include, say, the inductance of a resistor, then it can be shown as a separate ideal inductor in series with the resistor. LTspice, however, allows these secondary effects to be incorporated into the symbol, as we shall see in a later chapter.

1.3 Drawing Conventions

Obviously, there must be an accepted convention for representing electronic components and in drawing circuits. In Europe, the International Electrotechnical Commission (IEC) symbols predominate, whereas in the USA the IEEE usage is preferred. The divergences are few and should not give rise to confusion: in particular, the IEEE symbol for a resistor is a zigzag line. This can occasion heated debate about whether the line should first go to the left or to the right (with the attendant political overtones), how many zigzags should there be and what angle between them, whereas, apart from the ratio of width to length, not much more can be said about the IEC symbol of a simple rectangle. Other differences will be discussed as we encounter them. The LTspice toolbar and the directory immediately opened when picking a component use the IEEE symbols although there already are a few IEC symbols in a separate folder and it is not difficult to make more.

The drawing convention is that as far as possible, a circuit is read from left to right – that is, with inputs on the left and outputs on the right – and from top to bottom. Lines representing conductors are drawn horizontally or vertically, or at 45° in special circumstances. Connections are drawn as filled circles, and only three wires should join at a point, not four. This is to avoid the 'cake crumb' effect where lines crossing shown by '+' could be turned into four lines joining by an extraneous speck of dust in the photocopier. LTspice, however, does not enforce this. If it is important to emphasize the point that a component is connected directly to another, for example, that resistor *Rs* is connected to point *A*, then an 'offset junction' can be used with the lines at 45° as shown in the rather fanciful circuit of Fig. 1.1 which illustrates some of the conventions. It also uses the IEC symbols rather than the ones used by LTspice.

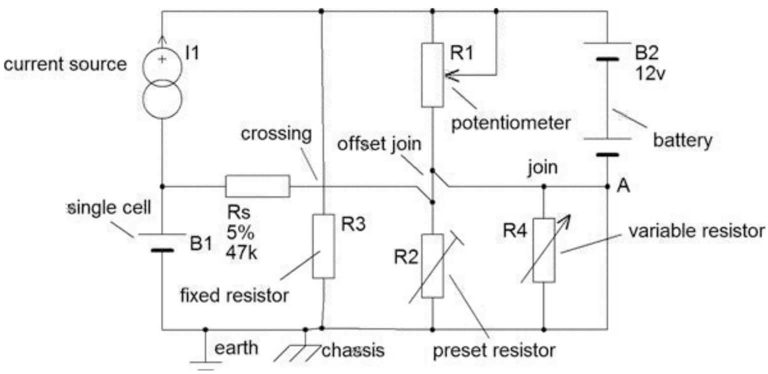

Fig. 1.1 Circuit Diagram

1.3.1 Component Symbols

The ones appropriate to this and the following chapter are as follows.

1.3.1.1 Voltage Sources

Voltage sources are either DC or AC. For our purposes, DC is a steady, unchanging voltage, either positive or negative. AC sources will be dealt with in later chapters.

So, turning to DC sources, there is a technical difference between a single cell and a battery: a single cell may either be single use (the traditional 'dry cell') or, more commonly now, rechargeable. They rely on the electrochemical potential difference between two metals. Typically this is around 1.2–1.6 V. A commercial battery consists of a number of identical cells in series and are often shown just by the single cell symbol with the voltage written beside it as *B2 12V*. Otherwise we may draw duplicated single cells, or we may draw two cells spaced apart and joined by a dashed or dotted line to indicate that there are others in between.

LTspice has symbols for both a cell and a battery in the ***misc*** folder, but – being lazy – it's easiest to use the general symbol for a voltage source consisting of a circle with '+' '−' drawn on it, and this also serves for AC sources. We should note that the cell and battery symbols are simply alternative representations of the voltage source circle and have the same attributes. So if we really want to create confusion, we can use the battery symbol and assign it an AC value, or have a cell with a pulse output.

1.3.1.2 Resistors

Simple resistors are shown as a rectangle, but notice that *Rs* also has its value and tolerance, and power rating could be added, but often, to avoid cluttering the diagram, this information is tabulated elsewhere rather than included in the diagram.

Variable resistors as *front panel controls* (i.e. those freely adjustable by the user, such as the volume control of a radio) are drawn with a diagonal arrow (R4), whereas *preset resistors,* whose value is set during calibration and not easily accessible, are shown with a 'hammer' (R2) rather than an arrow. These are two-terminal resistors, but very often, we use a *potentiometer* which is a three-terminal resistor with a sliding contact that divides the total resistance (R1). The difference is that we have a constant resistance between the ends and if we apply a signal between them, we can tap off a portion of the signal using the slider. Manufacturers generally only make potentiometers, and we make a variable resistor by leaving one end unconnected, or connect it to the slider which is how *R1* is drawn.

1.3.1.3 Current Sources

Current sources are difficult to find in practice and are often shown as two linked circles (I1), but diamonds with arrows are often used. Generally there is no confusion, and the context explains what is meant. LTspice uses a circle with an arrow pointing in the direction of current flow and is used for DC and AC sources.

1.3.1.4 Ground Connection

There is a subtle but important difference between the symbol for an earth connection which implies a low resistance conduction path to the earth itself (ideally zero ohms) and a chassis connection where the connection is made to a substantial metallic structure – traditionally the ‘chassis’ or framework on which the circuit was built – but which may not actually be connected to earth. LTspice uses a triangle, point down.

1.3.1.5 Connections

As was stated before, these need not be physical wires, but printed-circuit tracks or anything else offering a path with low resistance. It is implicit on the circuit diagram that these have zero resistance. LTspice uses a straight line for the wire and a small square for a join.

1.4 Drawing the Circuit and Ohm’s Law

Now before we say that this is blindingly obvious, and why did it take so long to arrive at this law, it is worth remembering the state of electrical science at that point. Stable, reproducible voltage sources were not easy to come by, and the galvanometer had only recently been invented. Wikipedia has some excellent articles on this. So, let us state the law, then see how we can observe it (not prove it) using LTspice.

$$V = IR$$

1.4.1 Drawing the Circuit

The workspace is where the circuit schematic will be drawn. In addition, text comments can be added, and the schematic given a name, if so desired. It will also be found that, by default, the SPICE commands are also shown on the workspace. This can be inhibited, but in general, it is very useful to see what simulation has been

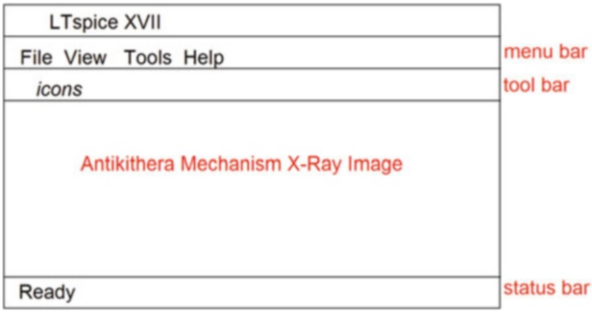

Fig. 1.2 LTspice screen

performed. In the following sections, stylized menus and edit boxes are shown with comments enclosed in brackets and some – but not all – of the captions, menu items and so on.

1.4.1.1 The Opening Screen

Download and open LTspice. A simplified opening screen is shown in Fig. 1.2. The very top row has the LTspice icon in the left corner followed by the version name, currently **LTspice XV11.** Below that is the ‘menu bar’ and underneath the icons of the ‘tool bar’ and then the workspace. The current default background is an X-ray of the Antikythera Mechanism – a truly remarkable machine showing that the ancient Greeks were not only philosophers, mathematicians and architects of the highest order but also instrument makers whose skill would not be equalled, let alone surpassed, for at least another thousand years. In this and later dialogues and windows the text presented by LTspice is shown in regular font, comments and explanations are shown in red, and typical user input is in bold. Menu selections are shown by an adjacent ‘<’.

Then, at the bottom of the window, the ‘status bar’ carries the single word – ‘Ready’. Users of Ubuntu or other versions of Linux may see something different.

1.4.2 Placing Components

Click **File→New schematic**, Fig. 1.3 and the background image (which defaults to the Antikythera Mechanism) will be replaced by a grey field with a matrix of small dots, and the cursor will change to a large, thin ‘+’. We are now going to build a very simple circuit to illustrate Ohm’s Law.

There can be up to four ways of placing a component. We shall explore these with a resistor.

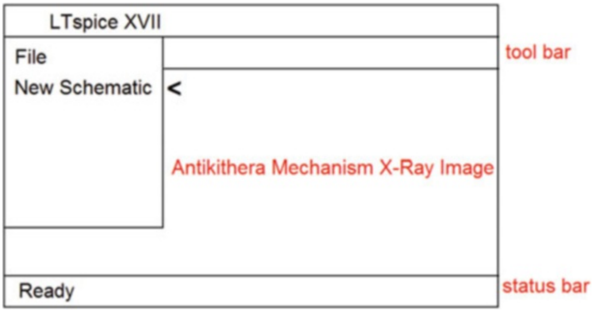

Fig. 1.3 New schematic

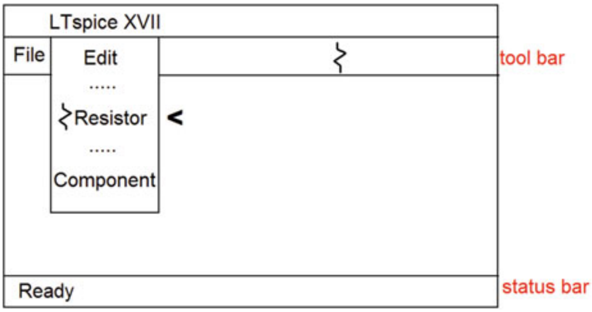

Fig. 1.4 Select resistor

1.4.2.1 Resistors

Left click **Edit** and select **Resistor**, Fig. 1.4. A large zigzag resistor will now replace the cursor. Move it to some convenient spot and left click to place it on the workspace.

A new *R2* will now replace *R1* and move with the mouse. Click '**Esc**' to dismiss it.

Example – Placing a Component

To test all the following methods, we need to remove unwanted resistors. We do that by pressing **F5**, and the cursor turns into a pair of open scissors. Move over the unwanted component and left click. The component will vanish. Right click to dismiss the scissors.

The second method is to click on the resistor symbol in the tool bar. The third is to right click on the schematic, and from the pop-up menu, select **Draft→Component** to open the **Select Component Symbol** dialogue, Fig. 1.5. The fourth, available only to some components, is to press the initial letter of their name whilst in the schematic. Thus *R* will create a resistor. We can test all these leaving just one resistor on the schematic.

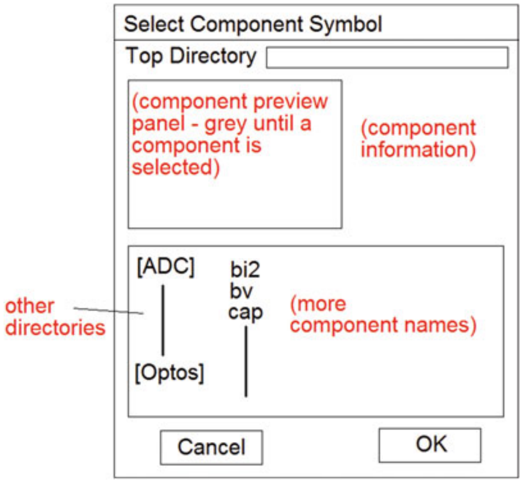

Fig. 1.5 Select comp

The 'Select Component Symbol' Dialogue
This is what we shall mostly use to select components. At the top is the full path to the component. Below, to the left, is the symbol itself and in the space to the right perhaps some words of explanation. Immediately below is a list box that can be scrolled horizontally with all the accessible components and, at the start at the left, directories for more components. We then have the choice to **Cancel** or accept **OK**.

1.4.2.2 Voltage Source

Return to **Edit,** but this time a voltage source is not available from the drop-down list so we must select **Component** (and note the shortcut of **F2**). This also will open the **Select Component Symbol** dialogue.

Scroll right, select **voltage,** and a circle with '+' and '−' signs will be seen in the 'component preview panel' in Fig. 1.5. Click **OK** and place to the left of the resistor and level with it. Alternatively, click **[Misc]** and select **cell** or **battery**. These only change the symbol, not the underlying functionality, so, rather bizarrely, as we noted above, we can have an AC battery. For convenience (and simplicity), nearly all the circuits use the **voltage** symbol, but please feel free to change.

1.4.2.3 Ground

This is accessible by all the methods for a resistor or press *G* in the workspace. This is important: there must be a ground point somewhere in every circuit. Place it between the voltage source and the resistor and a little below them.

1.4.2.4 Alternative Symbols

The IEC resistor, capacitor and a few others as well as cells and batteries can be accessed through **Edit→Component;** then in the **Select Component Symbol** dialogue, select **[Misc]** . They behave exactly as the default items. This is ('Ohm's Law 5.asc'). Later we can build a library of IEC symbols.

1.4.2.5 Component Names

LTspice assigns these in sequence such as *R1, R2, R3*. . .. To give a more evocative name, move the mouse over the name, and the cursor will change to an 'I-bar', and the status line will read 'Right click to edit the Name of..' which will open the **Enter new reference designator for**. . .. Dialogue. The names must be unique.

Component Reference Designator
This small dialogue opens when we are over the name of a component. The **Justification** options have little effect except that if we choose (not visible) we cannot recover it. The font size option applies to that instance only, as does making the text vertical.

If we just want to reposition the name or the value, press **F8** and the cursor will turn to a closed hand, left click on the element, move it, and right click to finish.

1.4.3 Connecting the Circuit

We are now ready to connect the circuit. Click on the pencil, and the cursor will turn to two dotted lines at right angles spanning the workspace. Move the intersection of the lines to one of the open square terminals of a component and left click. This will start the wire. Now move the cursor to another terminal, and note that the wire can only be drawn on a rectangular grid. Left click to finish the wire.

Draw the remaining wires and dismiss the wiring tool. You should now have the schematic ('Ohm's Law 1.asc') . Make sure the earth symbol is connected.

There is a trick to make life easier, and that is to draw straight through the components rather than connecting one end at a time. Create the schematic Fig. 1.6 of just the five resistors. (Press **Ctrl+R** to rotate the resistor before it is placed) Then select the wire tool and left click on point A. Now move straight to point *B* and left

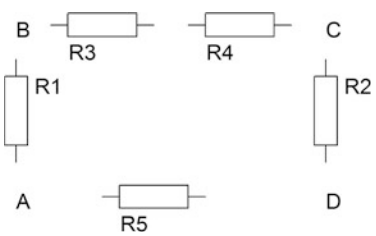

Fig 1.6 Easy wiring

click again: the wire will be trimmed to the ends of the resistor. Continue the wire to point *C* and left click again, and the wire will join up resistors *R3* and *R4*. And continuing in like fashion to point *D* then back to *A* will join up the circuit.

You can zoom in and out using the mouse wheel, and hold down the left mouse button and drag to move the circuit.

1.4.4 Adding Values

LTspice insists that every component has a value. At present, they only have labels.

1.4.4.1 Resistors

Move the mouse over the resistor and note that the status bar now has 'Right click to edit . . . .' with the name of the component under the mouse and the cursor changes to a pointing hand: now right click.

The **Resistor – R1** dialogue will appear, Fig. 1.7 with the **Resistance** edit box highlighted. Insert some number such as *100* and click **OK**. LTspice accepts all the multipliers including 'u' for 'µ', but note that 'm' or 'M' is always 'milli'. Either use scientific notation such as 1e6 for a megohm resistor, else '1Meg' is accepted.

We should also note that we can enter the tolerance and power rating of the resistor. LTspice accepts them but does not use them. We can click on the **Select Resistor** button to open the catalogue and select a resistor from there. As it happens, the **Mfg.** and **Part No.** fields are empty, but had they been filled the information would have been copied to the dialogue.

1.4.4.2 Voltage Sources

Move to the voltage source, right click again, and the dialogue **Voltage Source – V1** will appear, Fig. 1.8. Insert some value such as ***10*** in **DC value[V]** and click **OK**. Ignore the **Advanced** button – that is for later chapters.

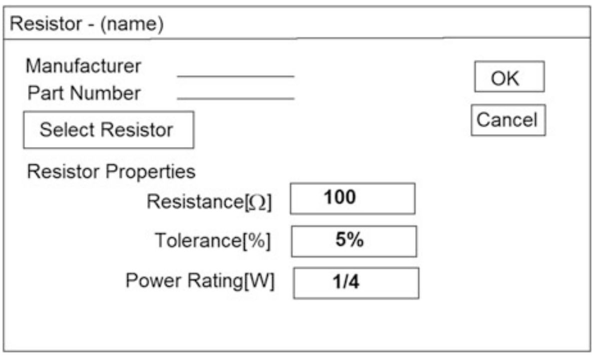

Fig. 1.7 Resistor dialogue

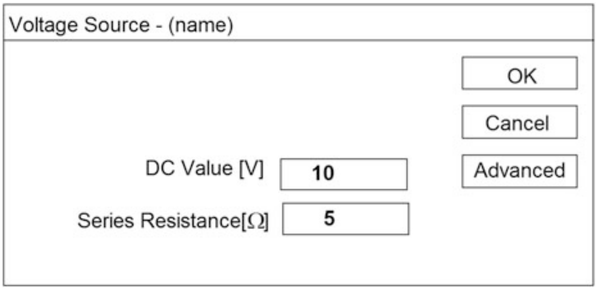

Fig. 1.8 V Source dialogue

We can also insert a series resistor to represent the internal resistance of the source. This is computationally more efficient than an external resistor but leave it empty now. The circuit is complete.

1.4.5 Editing the Circuit

With such a simple circuit, it is unlikely that much will need to be done. However, for practice, try the following, and note that, in practically every case, a left click will finish the action and **Esc** or a right click will cancel it.

To Cancel a Component
While it is still being dragged on the schematic and before it has been placed, right click or press *Esc* to cancel it.

To Remove Unwanted or Wrong Components
Press **F5** or select the scissors on the toolbar. Move the scissors over the component and left click; right click afterwards to dismiss the scissors.

To Move or Rotate a Component Without Its Connecting Wires
This can be useful for swapping components. It is also useful to be able to rotate a component through 180° since the direction of the current measured by LTspice depends on this.

Before it has been placed, press **Ctrl+R**. Once it has been placed, press **F7** or select the open hand from the toolbar, move the hand over the component, and left click. The hand will disappear and now **Ctrl+R** will rotate the component. At this time you can also drag the component without its wires to some other position. Left click to finish the move, then **Esc** to restore the cursor. If **Esc** is pressed or a right click is made, before the left click, the move is cancelled.

To Flip a Component Left-Right
While it is still being placed, press **Ctrl + E** then left click to restore the hand and **Esc** to finish. It can be dragged to the 'E' and reversed 'E' button on the toolbar, but this

can be confusing. Once it has been placed, select **F7** or **F8** then press **Ctrl + E. Esc** will undo it.

To Move a Component with Its Connecting Wires
Press **F8** or select the closed hand from the toolbar. Trying to rotate or flip a component with its wires usually creates a tangle, and it is better to select **F7** to handle the component alone. However, it can be useful for sliding components along a wire to make a more aesthetic layout.

To Move Several Components and Their Connecting Wires
Click **F8** then hold down the left mouse button to draw an enclosing rectangle. Release the key and drag. Left click to finish or right click to cancel. And **Edit→Undo** will also restore the previous setting.

To Move Just the Name of a Component, or Its Value
Click **F7** then click on the item and drag it.

To Change the Value or Name of a Component
Once the component has been given a value, move the cursor over it and it will change to an 'I-bar' and a right click will again bring up the editor. Similarly we can edit the name.

To Copy Components
For a single component, press **F6**, move over the component and left click. Now a copy can be dragged without any connecting wires. As usual, left click to place it, then right click to restore the cursor, or right click or **Esc** to cancel before it is placed.

To copy more than one component, press **F6** then hold down the left mouse button and drag an enclosing rectangle. Release the button. If the rectangle encloses wires with both ends connected, the wire will be copied as well as the enclosed components.

To Undo or Redo
Click the left or right arc to the right of the closed hand or **Edit→Undo/Redo** or right click in the schematic window then **Edit→Undo/Redo** from the pop-up menu. This works for a depth that is sufficient for any reasonable schematic. It can be used for 'what if?' explorations by changing a the value of a component (or, indeed, changing the component itself) and then restoring the original value.

To Give a Node a Label
The nodes are labelled from 001 onwards by default. It is often more convenient to give them a meaningful label. Move the mouse over a wire: right click and select **Label net**. The label with a small square box underneath will then replace the cursor. Move the box over a wire and left click to place it or right click to dismiss it.

To Remove Several Components
Select the scissors tool, move to a point outside the components to delete then drag to define a rectangle enclosing the components. Release the mouse button and they will be deleted, but **Redo** will bring them back again.