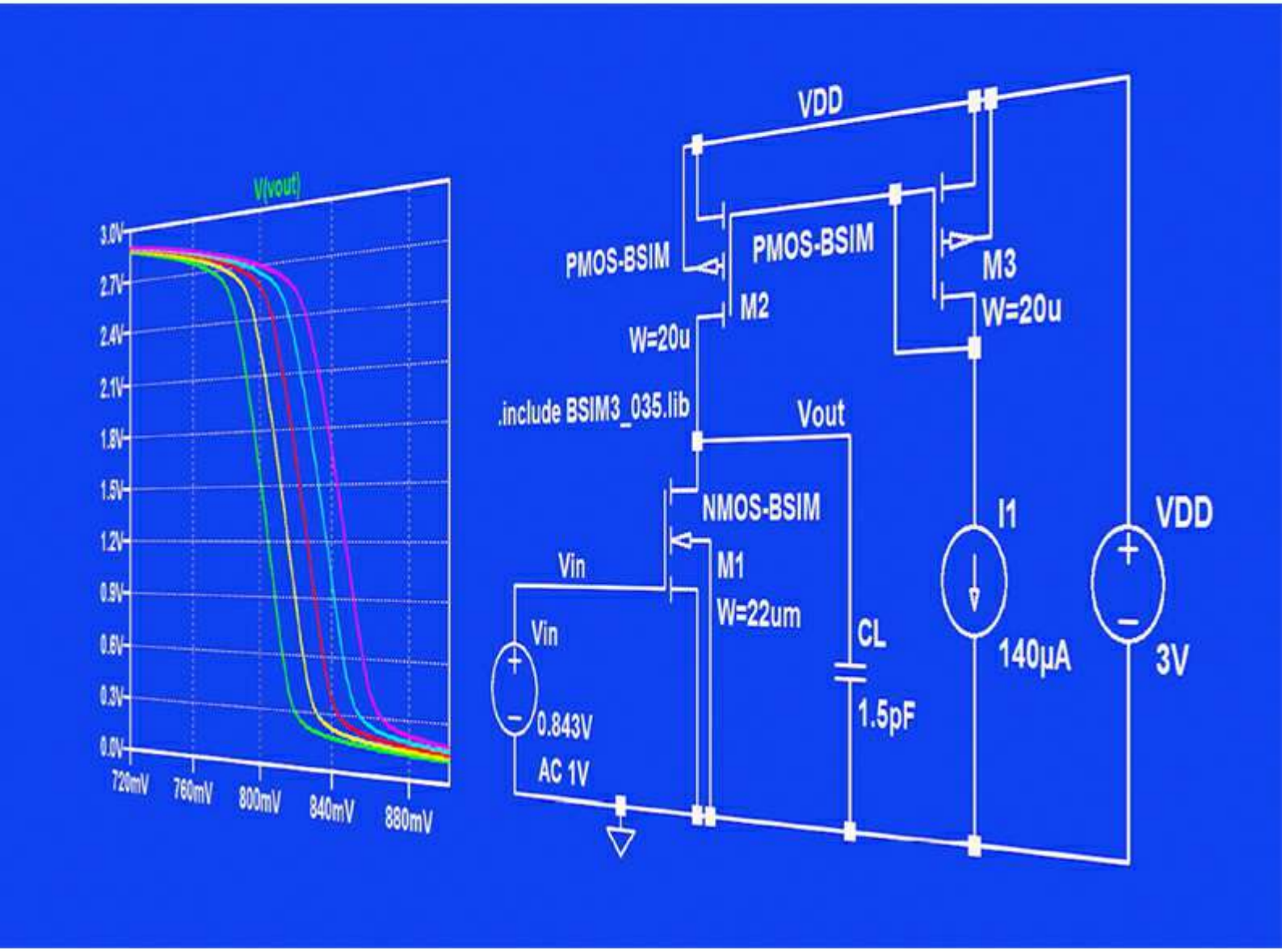
bookboon
LEARNING



CMOS Integrated Circuit Simulation with LTspice

Erik Bruun

ERIK BRUUN

CMOS INTEGRATED CIRCUIT SIMULATION WITH LTSPICE

CMOS Integrated Circuit Simulation with LTspice
3rd edition

ISBN 978-87-403-3454-8

CONTENTS

Preface

This book is about circuit simulation using the simulation program LTspice. It is intended as an introduction to LTspice and to simulation of CMOS integrated circuits with LTspice. It may serve as a supplementary textbook for an introductory course in analog integrated circuit design. The first tutorials can also be used as a general introduction to circuit simulation in an introductory course in electronic circuits. The book can be used for classroom teaching, and it can also be used for self-study. It is based on LTspice for Windows.

The book is organized as a number of tutorials, each comprising several simulation examples and a selection of end-of-chapter problems. You are highly encouraged to complete the examples yourself and to try to solve the end-of-chapter problems. 'Learning by doing' is the only efficient way of learning to use a program like LTspice.

This is the third edition of the book. In this edition, more examples and end-of-chapter problems have been added, including an example of transistor models for FinFET transistors.

Tutorials 1 and 2 introduce the fundamental concept of the circuit simulator demonstrated on circuits using passive devices (resistors, capacitors and inductors) and ideal voltage sources and current sources, both independent sources and controlled sources.

Tutorial 3 is about MOS transistor models and gives an introduction to the standard Shichman-Hodges transistor model often used for hand calculations when analyzing CMOS circuits. Also, it provides an introduction to more advanced transistor models and a comparison between the advanced transistor models and the simple Shichman-Hodges model.

Tutorial 4 gives examples of basic CMOS amplifier stages, i.e., the common-source stage, the common-drain stage, the common-gate stage and the differential pair. Both analysis and design approaches using LTspice are shown.

Tutorial 5 shows how the basic stages can be defined as subcircuits and combined into a multistage operational amplifier. Also given in this tutorial is a design example of a two-stage opamp for a feedback amplifier, generic filter blocks and a mixed analog/digital circuit. The tutorial is an introduction to hierarchical design.

Tutorial 6 is about the simulation of process and parameter variations in a circuit. In integrated circuit design, process variations pose a major challenge to the designer. Often technology files are supplied for typical process parameters and a selection of worst-case process parameters. The tutorial gives an introduction to simulation with technology files including process variations. Also supply-voltage variations and temperature variations are considered. Together, these variations are termed PVT variations.

Tutorial 7 is about import of netlist files and model files and export of output files from LTspice. The netlist files are the primary descriptive files for a circuit to be simulated by Spice. There are minor differences between netlist files originating from LTspice and other versions of Spice, but in general, it is rather straightforward to modify a netlist file to be compatible with LTspice. Several textbooks provide examples of netlist files which may be used for simulation with LTspice. A schematic is not needed. The simulation commands in LTspice can be executed directly from the netlist files.

End-of-chapter problems are provided for all tutorials to further illustrate the subject of the tutorials and a solutions manual for the problems is also available from bookboon.com:
E. Bruun 2020, *CMOS Integrated Circuit Simulation: Solutions*, Third Edition.

Finally, two appendices are included. Appendix A is a beginner's guide which may facilitate quick and easy learning of LTspice for the reader or student who is new to LTspice. Appendix B provides a number of BSIM transistor model files for use in LTspice. The transistor model files may downloaded from the webpage of this book, https://bookboon.com/en/cmos-integrated-circuit-simulation-with-ltspice-ebook.

I hope you find this book useful. If you find typos or errors, I would appreciate your feedback. Suggestions for improvement are also welcome. You may send them to me by email, eb@elektro.dtu.dk.

Acknowledgements: The author would like to acknowledge the many students who have contributed with comments and suggestion for the book. In particular, I would like to express my thanks to Mir Muntasir Hossain, Gradute Student at Ahsanullah University of Science and Technology, Dhaka, Bangladesh, for drawing my attention to the modeling of FinFET transistors and for co-authoring Example 7.3 about FinFET models.

Also, a particular acknowledgement goes to my colleague Dennis Øland Larsen who reviewed the entire manuscript for the first edition of the book and to my colleague Ivan Jørgensen who reviewed the entire manuscript for the subsequent editions of the book. They both provided many useful comments and corrections during the final phases of writing.

Erik Bruun
Department of Electrical Engineering, Technical University of Denmark

Getting started

The program LTspice is freely available from Analog Devices,
https://www.analog.com/en/design-center/design-tools-and-calculators/ltspice-simulator.html.
Just click 'Download LTspice' and follow the instructions.

This book is addressing the simulation of integrated circuits, in particular CMOS circuits, so we will not go into detail with the simulation of circuits with standard components but refer the reader to the many examples of demo circuits using standard components which are found on the LTspice website. Here you will also find a blog with several hints and video clips on how to use LTspice. However, Tutorials 1 and 2 may serve as a general introduction to LTspice. Also, a 'Getting started guide' is available from https://www.analog.com/media/en/simulation-models/spice-models/LTspiceGettingStartedGuide.pdf.

In addition, comprehensive books and guides about Spice can be found, (Tuinenga 1995) and (Vladimirescu 1994), and a manual dedicated to LTspice is also available (Brocard 2013). Also, a user group is available on the internet, https://groups.io/g/LTspice. However, the program is fairly easy and intuitive and once the installation is complete, you may go directly to the first tutorial, providing you with examples of circuits using resistors, voltage sources and current sources. A 'learning by doing' approach is perfectly feasible with LTspice.

The program also includes a 'Help' function with detailed descriptions of the commands and options in the program. The keyboard shortcut to 'Help' is 'F1' in the windows version and '⌘?' in the Mac version. If you want a paper manual for the program, you can get it using the 'Help' function: Just open 'Help' (type 'F1'), select the tab 'Contents', click on the 'Print' symbol and select 'Print the selected heading and all subtopics' in the dialogue box which opens. Your printer should be ready for printing about 110 pages.

This book is based on the Windows version of LTspice. The program is also available for Mac. There are some differences in the user interface of the two versions. This might be somewhat confusing for first-time users. As a guide to Mac users, some of the differences which may initially cause confusion are listed below.

- The toolbar shown in Fig. 1.2 is not available in the Mac version. Instead, a right-click on the drawing sheet will open a menu with several sub-menus. The 'Draft' sub-menu allows you to insert 'Components', 'Wires', 'Net Names', 'SPICE directives', etc. In particular, you should notice that the ground symbol is not available via 'Components', but it can be inserted using the keyboard shortcut (hotkey) 'G' or using 'Label Net' (hotkey F4) with the net name selected to be 'GND(global node 0)'.
- The editing commands ('Move', 'Drag', 'Duplicate', etc.) are found in the 'Edit' sub-menu. The rotate and mirror operations are available via '⌘R' and '⌘E' (or 'Ctrl-R' and 'Ctrl-E').
- The 'Simulate' command shown in Fig. 1.2 is not available in the Mac version. Instead, use 'SPICE directives' from the 'Draft' sub-menu and type in the appropriate simulation command. The help function provided by the window shown in Fig. 1.5 with different tabs for the different simulation commands can be opened by right-clicking in the 'SPICE directives' dialogue box. This opens a 'Help me Edit' option where you can select 'Analysis Cmd'. A similar help function is available for several other SPICE directives.
- The result of a 'DC operating point' simulation ('.op') is not automatically displayed in a window like shown in Fig. 1.6. Instead, a plot window opens, and you can select the currents and voltages to be displayed by pointing to relevant components and nodes in the schematic as described in Tutorial 1, Example 1.1. If you want the simulation result in a format as shown in Fig. 1.6, open the 'Spice Error Log' from the 'View' sub-menu or by '⌘L'.
- The results of a 'DC Transfer' simulation ('.tf') are not displayed in a window like shown in Fig. 1.26. Instead, a plot window opens, and using 'Add Traces' from the plot window, you can select the transfer function, the input resistance and the output resistance.
- When selecting a new 'Simulate' command, previous simulation commands are not automatically changed into comments.
- For transistors, the small-signal parameters calculated by a 'DC operating point' simulation ('.op') are listed in the 'Spice Error Log' together with the bias values of voltages and currents. Also for an 'AC Analysis', the small-signal transistor parameters for the bias point are listed in the 'Spice Error Log'.
- Not only in the schematics sheet but also in waveform plots, a right-click opens a menu with several sub-menus.
- The commands for copying schematics and waveform plots to the clipboard are found in the submenu 'View → Paste Bitmap'.

References

Brocard, G. 2013, *The LTspice IV Simulator – Manual, Methods and Applications*, First Edition, Swiridoff Verlag, Künzelsau, Germany.

Tuinenga, PW. 1995, *Spice: A Guide to Circuit Simulation and Analysis Using PSpice*, Third Edition, Prentice Hall, Upper Saddle River, USA.

Vladimirescu, A. 1994, *The SPICE book*, First Edition, John Wiley & Sons, Hoboken, USA.

Tutorial 1 – Resistive Circuits

This tutorial is an introduction to the basics of LTspice simulation of resistive circuits with voltage sources and current sources. After having completed the tutorial, you should be able to

- draw circuits using the schematic editor in LTspice.
- specify resistors, independent sources and controlled sources in LTspice.
- recognize the basic netlist structure for simple circuits in LTspice.
- run simulations of operating points, dc sweeps and small-signal transfer functions.
- run simulations with parameter sweeps.
- plot simulation results using the waveform viewer of LTspice.

Example 1.1: A resistor circuit.

The first example is a simple circuit with four resistors and a voltage source as shown in Fig. 1.1:

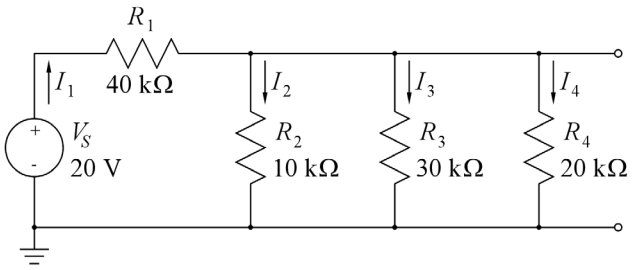


Figure 1.1: Circuit for first simulation.

Drawing the circuit: Start by opening a new file in LTspice ('File → New Schematic' or the leftmost symbol in the Editor toolbar). Next, you should draw the schematic shown in Fig. 1.1. Click (left-click) with the mouse on the resistor symbol shown in the toolbar (symbol) and place the four resistors. You may rotate a resistor by clicking on the 'rotate' symbol on the toolbar or by typing 'Ctrl-R' when placing the resistor. Right-click on the mouse (or type 'Esc') to leave the insertion command. As an alternative to picking the resistor from the toolbar, you may use the command 'Edit → Resistor', or you may simply type 'R'. The resistors may now be edited to the correct values and numbers shown in Fig. 1.1. Move the cursor to the resistor name (the reference designator, e.g., '`R1`'). If you have inserted a resistor without rotating or mirroring the symbol, the name is the upper text appearing to the right of the symbol. On the status bar at the bottom of the LTspice program window, a message will appear, telling you that with a right-click, you can edit the name of the resistor. The right-click opens

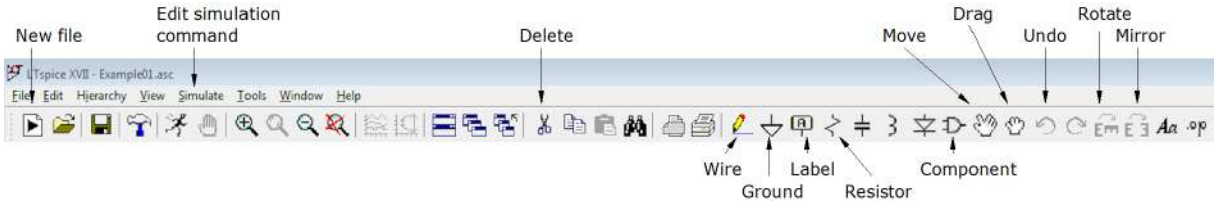


Figure 1.2: Some toolbar symbols.

a dialogue box where you can enter the new reference designator. Likewise, the value of the resistor is edited by right-clicking 'R'. This is the lower text appearing to the right of the symbol when a resistor is inserted without rotation or mirroring. If you rotate or mirror your symbol, the name (device designator) and value will move. You can always see on the status bar at the bottom of the LTspice program window if you have placed the cursor on the name of a component or on the value. Do not confuse name and value! It will lead to errors in the simulation.

A figure pointing out some of the toolbar symbols is shown in Fig. 1.2.

The voltage source V_S is inserted by selecting the 'Component' symbol on the toolbar, symbol  . Click on the symbol (left-click) and a selection box will appear with a large selection of components, see Fig. 1.3. Select 'voltage'. This results in the symbol for a voltage source. The value and the name are edited in the same way as for the resistors. Also for the voltage source, you may use the 'Edit' command instead of picking the symbol from the toolbar ('Edit → Component'), or you may simply type 'F2' which will bring you to the component selection box.

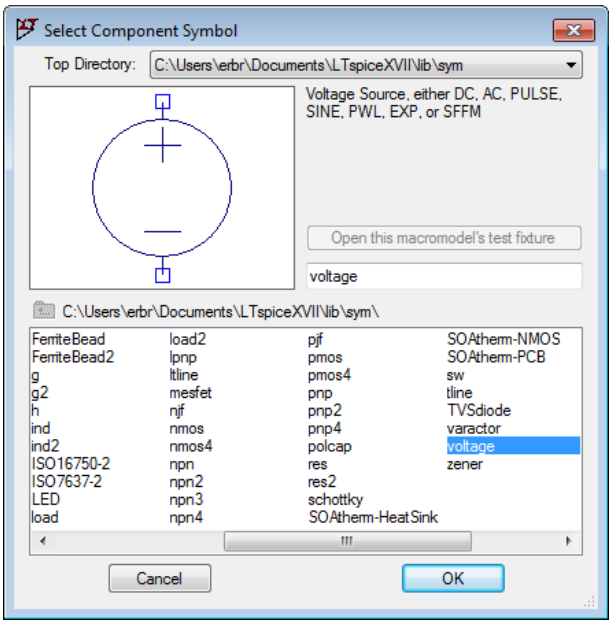


Figure 1.3: Component selection box.

The components are connected together by wires inserted with the 'wire'-symbol (symbol) or the keyboard shortcut (hotkey) 'F3'.

Also remember to insert a ground symbol (symbol or keyboard shortcut (hotkey) 'G') to indicate the reference voltage of 0 V. If the ground is missing in the schematic, LTspice will not execute a simulation.

It is a good idea to give names to important nodes in the circuit, e.g., V_1 and V_2, using the symbol 'Label Net' from the toolbar or the hotkey 'F4'. Alternatively, point to a node and right-click. This opens a dialogue box where you can select 'Label Net' and type in a name. You can also insert the ground symbol in this way by ticking 'GND(global node 0)' in the dialogue box for 'Net Name'.

If you wish to make adjustments to your schematic, you can move or drag symbols using the hotkeys 'F7' or 'F8', respectively (or symbols and on the toolbar, or the 'Edit → Move' and 'Edit → Drag' commands). Also, you can delete a symbol or a wire using 'F5', toolbar symbol or 'Edit → Delete', and you can duplicate symbols using 'F6', toolbar symbol or 'Edit → Duplicate'. These commands work not only on single symbols: When you have activated one of the commands, you can define a box by clicking and dragging using the left mouse button, and the command will work on the entire contents of the box.

The assignment of hotkeys can be seen (and edited) using the command 'Tools → Control Panel → Drafting Options → Hot Keys'.

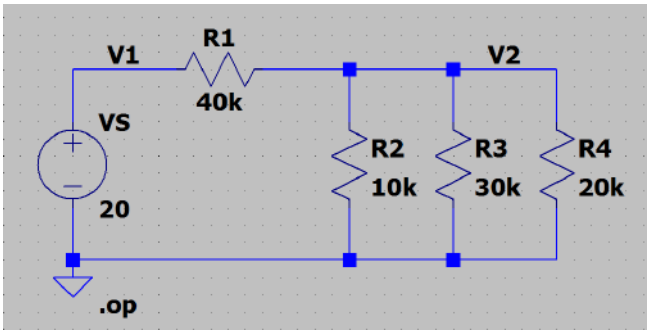


Figure 1.4: Schematic from LTspice.

The resulting schematic may look like the schematic shown in Fig. 1.4. When the schematic has been completed, you should save it (using 'File → Save as') in an appropriate folder for your circuits and using a suitable file name. You can also export the schematic to other programs. A very simple method is to use the command 'Tools → Copy bitmap to Clipboard' and then paste the schematic into another program (e.g., Microsoft Word) from the clipboard (using 'Ctrl-V').

Simulating the circuit: Now the circuit is ready to be simulated. For this, we need a simulation command. When selecting the command 'Simulate → Edit Simulation Cmd', a window opens with a number of tabs as shown in Fig. 1.5. This is a 'Help me Edit' function for the SPICE directives used to specify simulations. Each tab provides help for the basic simulation modes in LTspice. These are:

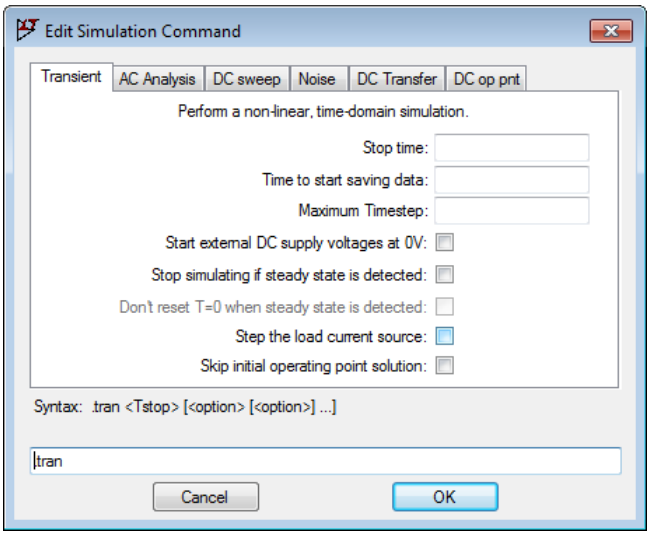


Figure 1.5: Help window for editing the simulation commands.

Transient: Perform a nonlinear time domain simulation. This is used for finding voltages and currents as function of time, e.g., charging and discharging of a capacitor.

AC Analysis: Compute the small-signal ac behavior of the circuit linearized about its dc operating point. This is used for finding the frequency response of a circuit, e.g., the Bode plot of a gain function.

DC sweep: Compute the dc operating point of a circuit while stepping independent sources and treating capacitances as open circuits and inductances as short circuits. This is used for finding voltages and currents as functions of one (or more) signals varying in magnitude, e.g., the output voltage of an amplifier as a function of the input voltage.

Noise: Perform a stochastic noise analysis of the circuit linearized about its dc operating point. This is used for analyzing the noise performance of a circuit, e.g., finding thermal noise and flicker noise in a gain stage with MOS transistors.

DC Transfer: Find the dc small-signal transfer function. This is used for finding small-signal input resistance, output resistance and transfer function for a circuit at dc, i.e., the frequency of the input signal source is 0.

DC op pnt: Compute the dc operating point treating capacitances as open circuits and inductances as short circuits. This is used for finding dc voltages and currents in a bias point for a circuit. It is also used for finding small-signal parameters of transistors in the bias point.

For the first simulation of the circuit in Fig. 1.4, we just need to find some dc voltages and currents in some devices. This is done using the simulation command 'DC operating point' (DC op pnt). You open the tab 'DC op pnt' and select the command '.op' by clicking 'OK'. This opens a command line (SPICE directive) which can now be placed on the schematic by the cursor. Insert the command by a left-click on the mouse or by pressing 'Return'.

Output from DC op pnt simulation

```
--- Operating Point ---

V(v2):          2.4             voltage
V(v1):          20              voltage
I(R1):          -0.00044        device_current
I(R4):          0.00012         device_current
I(R3):          8e-005          device_current
I(R2):          0.00024         device_current
I(Vs):          -0.00044        device_current
```

Figure 1.6: Simulation result for circuit example from Fig. 1.4.

Next, the simulation is run by the command 'Simulate → Run' or by using the 'Run'-symbol on the toolbar. You may also just right-click anywhere in the schematic and click ' Run' in the window which opens. If there are no errors in the schematic, the simulation opens a new window with a list of all node voltages and device currents, see Fig. 1.6.

Once you have closed the window, you can re-open it by the command 'View → Visible Traces', toolbar symbol .

Notice that LTspice inherently specifies a direction of current flow for each of the components. For the voltage source '`VS`', the positive direction of current flow is into the positive terminal of the voltage source. In our case, the current is flowing out of the positive terminal of the voltage source, so in Fig. 1.6, the current '`I(Vs)`' appears with a negative value. Also the current flow in a resistor is defined with a sign. Unfortunately, you cannot from the symbol see which end of the resistor is the positive end. When you insert a resistor without rotating it or mirroring it, the positive terminal is the upper terminal, so the positive direction of current flow is downwards. If you rotate the resistor once in order to have a horizontal resistor symbol, the positive current flow is from right to left.

If your schematic contains errors, a window will open giving suggestions concerning what can be wrong. For instance, the ground symbol may be missing or a resistor value has not been specified. A slightly more tricky error has to do with the specification of component values. Be aware that a space between the value and the suffix is not allowed. If there is a space, the suffix will be ignored and the simulation will run with some unintended component values. A result window like shown in Fig. 1.6 will still be shown but when you close this, a new window with an error log will appear. Also note that the suffix for 'Mega' is Meg (or meg - LTspice is case insensitive) while the suffix for 'milli' is m (and if you insert M, LTspice will change it into m to warn you that M does not mean Mega). Another common error is a value specification using a wrong syntax. Note that a 10 kΩ resistor cannot be specified as '10∗10^3'. LTspice does not accept this notation. It has to be specified as '10k' or '10e3' (or '1e4'). If you use the wrong specification ('10∗10^3'), the '.op' simulation will still run and an output file will be shown, but when you close the output file, an error log file will automatically open, telling you that there is an error in the resistor specification (Unknown parameter "∗10^3"). Examining the output file, you will note that the '.op' simulation has been executed with a resistor value of 10 Ω.

LTspice netlist

```
* M:\LTspice\Tutorial01\fig1_04.asc
R2 V2 0 10k
R3 V2 0 30k
R4 V2 0 20k
R1 V2 V1 40k
VS V1 0 20
.op
.backanno
.end
```

Figure 1.7: LTspice netlist for circuit example from Fig. 1.4.

When you have successfully completed the '.op' simulation and closed the window with the results, you can see currents and voltages in the circuit by moving the cursor to a component or a node and reading currents and voltages on the status bar at the bottom of the LTspice program window.

It may be useful to know at least the basics about the circuit description used by LTspice. The circuit is described by a netlist, and you can see the netlist using the command 'View → SPICE Netlist'. Figure 1.7 shows the netlist corresponding to the circuit from Fig. 1.4. You would notice the syntax for a resistor, for instance R_1: '`R1 V2 V1 40k`'. Here you will recognize that the first node specified for the resistor (in this example '`V2`') is the positive terminal of the resistor. Also, you may notice that the netlist file starts with the circuit description where the lines in the netlist appear in the sequence in which you inserted the components. Following the circuit description are SPICE directives. For the netlist shown in Fig. 1.7, we just have the simulation command '.op', an autogenerated LTspice command '.backanno' (linking the schematic and the netlist) and an autogenerated '.end' directive (to mark the end of the netlist file).

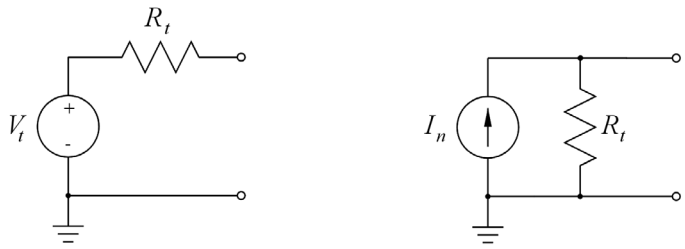


Figure 1.8: Thévenin equivalent (left) and Norton equivalent (right).

Thévenin – Norton equivalent circuits: For the circuit shown in Fig. 1.1, you may define a Thévenin equivalent and a Norton equivalent as shown in Fig. 1.8 (Hambley 2018). The Thévenin voltage V_t is the open-circuit voltage between the two rightmost terminals of the circuit in Fig. 1.1 and the Norton current I_n is the short-circuit current between the two terminals. The Thévenin resistance R_t is the ratio between the Thévenin voltage and the Norton current, i.e., $R_t = V_t/I_n$. Also, the Thévenin resistance can be found as the resistance between the circuit terminals when the independent sources in the circuit are reset, i.e., with $V_S = 0$ V. The Thévenin voltage has already been found by the simulation of the circuit in Fig. 1.4, and the result is given as the voltage '`V(v2)`' in Fig. 1.6, i.e., $V_t = 2.4$ V. The short-circuit current is found by placing a short circuit between the two rightmost terminals in the circuit. The short circuit could simply be a wire, but in this case, the current in the wire is not listed in the output file from the '.op' simulation. You may also try to insert a resistor with the value 0, but running the simulation, you will find that the output file does not show the value of the current in this resistor. You may change the resistor value to a very small value (e.g., 1e-6), and in this case, the output file will show the current in the short-circuit resistor. A better approach is to model the short circuit by a voltage source with a value of 0 V as shown in Fig. 1.9. In this case, the output file will show the current into the voltage source, and the voltage between the two terminals is 0 V, corresponding to a short circuit. When running this simulation, you will find $I_n =$`I(V1)`$= 0.5$ mA, and you can calculate R_t from $R_t = V_t/I_n = 4.8$ kΩ.

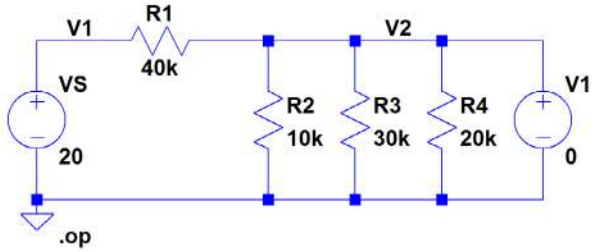


Figure 1.9: Schematic for simulating the Norton current.

Alternatively, R_t can be found by simulation: Insert a current source I_1 between the two rightmost terminals as shown in Fig. 1.10 and simulate the voltage V_2 across the current source with $V_S = 0$ V. The current source is inserted as a component where you select 'current' in the component selection window. With the current flowing into the V_2 terminal (rotate the current source symbol twice), the resistance is

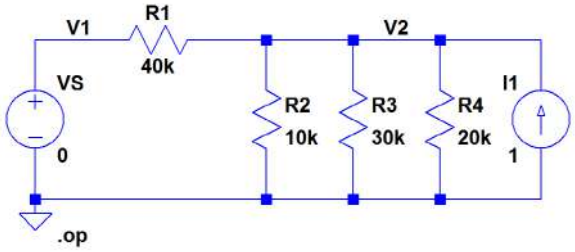


Figure 1.10: Schematic for simulating the Thévenin resistance.

found as V_2/I_1, so if the value of I_1 is selected to be 1, the value of the voltage V_2 is directly the value of the resistance between the terminals, i.e., R_t.

In Figs. 1.9 and 1.10, the background color has been changed to white using the command 'Tools → Color Preferences' which opens a 'Color Palette Editor' for specifying the colors being used for schematics, netlists and waveforms. Also, using the command 'Tools → Control Panel' and selecting the tab 'Drafting Options', the font has been changed from the default font (Tahoma) to Arial and the grid has been turned off by deselecting "Show schematic grid points'. The grid may also be turned on and off using the command 'View → Show Grid' or the keyboard shortcut 'Ctrl-G'.

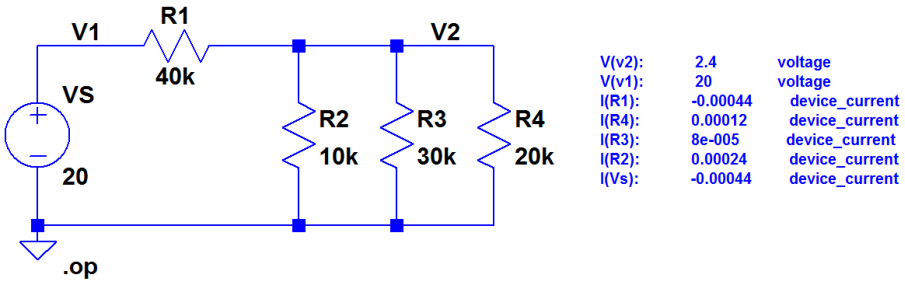
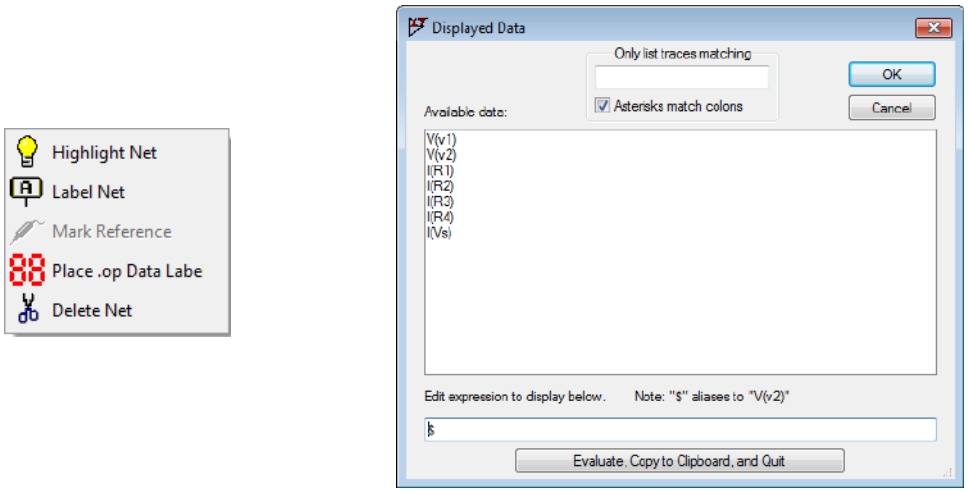


Figure 1.11: The circuit from Fig. 1.4 with the results of the '.op' simulation shown as text.

Annotating simulation results on the schematic: After having run a '.op' simulation, you may wish to display the simulation results directly on the schematic. Consider the circuit from Fig. 1.4. For this circuit, we found the results shown in Fig. 1.6. A very simple way to show these results on the schematic is to use the 'Edit → Text' command (toolbar symbol Aa, hotkey 'T') and just use normal copy and paste ('Ctrl-C', 'Ctrl-V') from the output file to the input window for the 'Edit → Text' command. The result of doing so may look like shown in Fig. 1.11. Notice that in this figure, the font size has been specified to 1.0 when inserting the text (the default is 1.5).

An alternative way to display specific simulation results is as follows: After having run the simulation, point to a net (wire) and right-click. This opens a command selection menu, see Fig. 1.12(a). Select the command 'Place .op Data Label'. This places a text box showing the voltage of the net. When you right-click on the voltage, the dialogue box shown in Fig. 1.12(b) opens, allowing you to select another quantity to display.

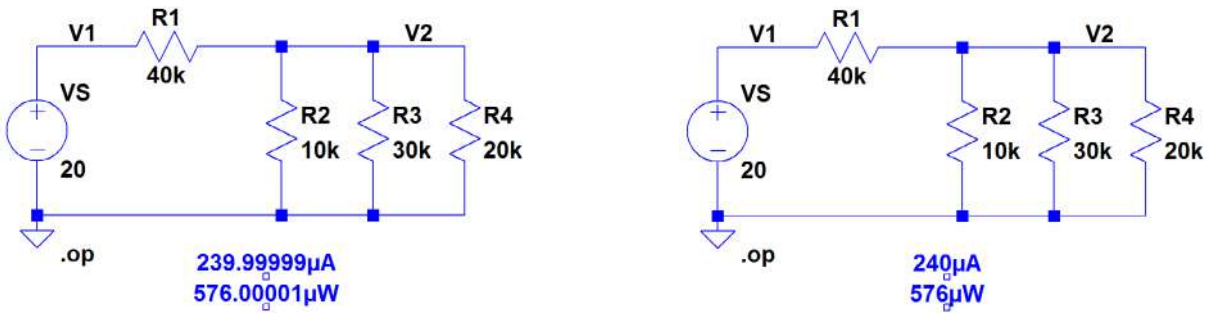


(a) (b)

Figure 1.12: Command selection menu (a) and dialogue box (b) for entering simulation results on the schematic.

Suppose that we are interested in displaying the current in R_2 and the power dissipated in R_2. The current is specified in the dialogue box, Fig. 1.12(b), by replacing the $-sign in the bottom line with '`I(R2)`'. You can also move the text box to an appropriate position by using the move command (hotkey 'F7').

Adding a new text box in the same way (i.e., by pointing to a net and right-clicking) lets you specify the expression '`V(v2)*I(R2)`' which will calculate the power in R_2. You may note that the $-sign in the dialogue box represents the voltage at the selected net, in this case '`V(v2)`', so the power may be specified as '`$*I(R2)`', but if you move the text box away from this net, the alias for the $-sign is lost and the textbox just displays '`???`'. When the text boxes have been moved, the resulting schematic may look like shown in Fig. 1.13(a). You may find that the current and power need rounding off to integer µA and µW. This can be achieved by using the function 'round(x)' in the specification window. Thus, for the current specify '`round(I(R2)*1e6)/1e6`' and for the power specify '`round(I(R2)*V(v2)*1e6)/1e6`'. Then



(a) (b)

Figure 1.13: The circuit from Fig. 1.4 with the current and the power for R_2 shown on the schematic.

the resulting schematic looks like shown in Fig. 1.13(b). You have to multiply by 10^6 (and divide the rounded result by 10^6) because the current and power is in µA and µW respectively. If you do not multiply by 10^6, the 'round(x)' function just rounds to a value of 0.

Sweeping dc voltages and currents: The simulations just shown give you values of voltages and currents in a specific operating point, i.e., for fixed values of all components and sources in the system. You can calculate the voltages and currents for other values of components and sources simply by modifying your schematic and running the '.op' simulation again. However, there is also the possibility to sweep voltage sources and current sources over a range of voltages or currents. Assume that we would like to find currents and voltages in the circuit from Fig. 1.1 for V_S varying between 10 V and 30 V. This is achieved by running a dc sweep simulation. Use the command 'Simulate → Edit Simulation Cmd' and open the tab 'DC sweep'. This opens a dialogue box where you can specify your signal source and the sweep range. Also the increment must be specified. Select for instance an increment of 1 V. When you have completed the specification for V_S, you click 'OK'. This opens a command line which can now be placed on the schematic by the cursor. Insert the command by a left-click on the mouse or by pressing 'Return'. The command is shown in the schematic as '`.dc VS 10 30 1`'. You may observe that your previous simulation command, '.op', is now modified to ';op'. This modification turns it into a comment, and only the new simulation command is executed when you run the simulation. Next, the simulation is run by the command 'Simulate → Run' or by using the 'Run'-symbol on the toolbar.

If there are no errors in the circuit and in the simulation command, a new window opens for showing plots of currents and/or voltages. The x-axis shows the voltage range specified for V_S, but initially the plot window is empty. The voltages and/or currents to be shown in the plot window can be selected in different ways: With the plot window active, you can use the command 'Plot Settings → Add trace' or the command 'Plot Settings → Visible Traces'. The command 'Visible Traces' is also available with the schematic window active ('View → Visible Traces') and on the toolbar, symbol . You may notice that the 'Add trace' command works in a different way than the 'Visible Traces' command. With the 'Add trace' command, you left-click on the traces that you want to see, and they are all listed in the window in the bottom of the dialogue box. With the 'Visible Traces' command, you select only one trace with a left-click. If you want more than one variable, use 'Ctrl-left-click' to turn on and off the traces to display. The 'Add trace' command is also available by the hotkey 'Ctrl-A'.

An alternative method for selecting traces is to point at nodes in the schematic for voltages and at components for currents. This turns the cursor into a red pointer, , an oscilloscope probe, for the voltages and a current probe for the currents, . Note that a red arrow in the current probe also shows the positive direction of current flow. Just left-click at the trace to be added and it will appear in the plot window. A double-click implies that only the selected trace is shown. Also, you may note that by pointing to a wire and pressing the 'Alt' key, you select the current in a wire. The voltage difference between two nodes can also be displayed using the voltage probe: Left-click and hold on one node and drag the mouse to

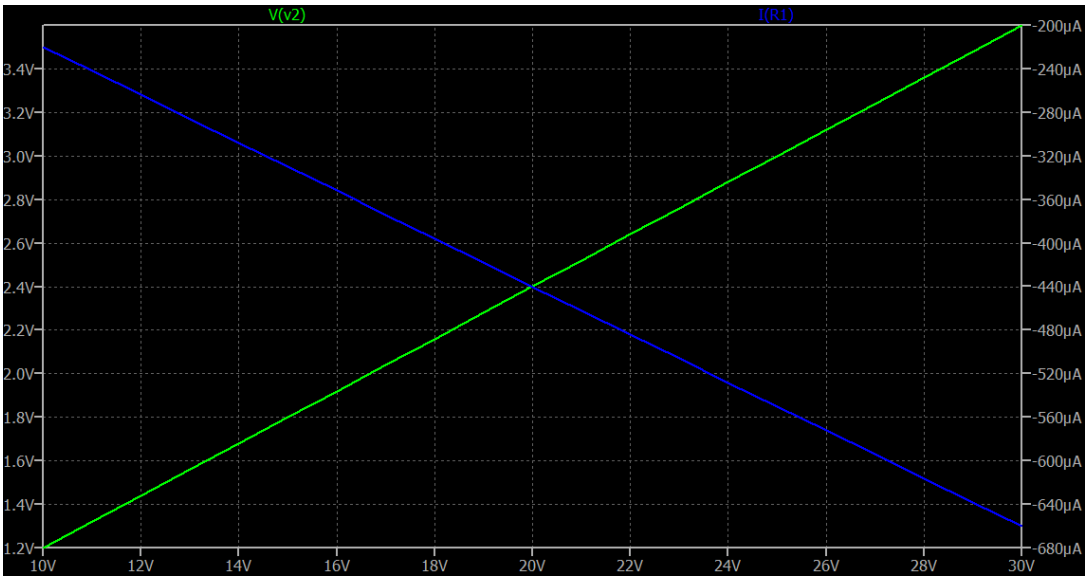


Figure 1.14: Plot of dc sweep simulation for circuit example from Fig. 1.4 using the LTspice default setup of colors.

another node. A red voltage probe will appear at the first node and a black probe at the second node. Finally, when you hold down the 'Alt' key while pointing to a device (e.g., a resistor), the cursor turns into a thermometer and the resulting plot traces the power dissipated in the device.

The waveform plot can be copied to the clipboard in the same way as the schematic: Use the command 'Tools → Copy bitmap to Clipboard' and then paste the waveform plot into another program. The resulting plot showing the voltage V_2 and the current through R_1 may look like shown in Fig. 1.14. You may find that the blue trace ('`I(R1)`') is difficult to see on the black background. You can change the color of the trace by pointing to the trace name above the plot and right-clicking. This opens a window where you can select another color. Alternatively, you may change the background color of the plot pane by the command 'Tools → Color Preferences' which opens a dialogue window where you can specify colors for waveforms, schematics and netlists. Also note that if you left-click instead of right-click on the trace name, a cursor appears which will follow the trace when you move it around by the mouse. This is useful for finding values of the trace for specific values of the voltage V_S.

Once you have closed the plot window, you can re-open it by the command 'View → Visible Traces', toolbar symbol . If you have applied the command 'Plot Settings → Save Plot Settings' before closing the plot window, it will re-open showing the selected traces, otherwise just with an empty plot window.

Another way of finding values of the currents and voltages for specific values of V_S is to use the command 'File → Export data as text' from the plot window. This opens a window for selecting waveforms to export, and when you have selected the desired waveforms and click 'OK', a '.txt' file is generated with the waveforms given in tabular form. This file can also be opened by LTspice. Use 'File → Open' (or on the toolbar) and select 'Files of type: all files'. In the file list, open the '.txt' file with the name corresponding to your circuit. The resulting table may look like shown in Fig. 1.15.

Exported file with selected traces from dc sweep simulation

vs	V(v2)	I(R1)
1.000000000000000e+001	1.200000e+000	-2.200000e-004
1.100000000000000e+001	1.320000e+000	-2.420000e-004
1.200000000000000e+001	1.440000e+000	-2.640000e-004
1.300000000000000e+001	1.560000e+000	-2.860000e-004
1.400000000000000e+001	1.680000e+000	-3.080000e-004
1.500000000000000e+001	1.800000e+000	-3.300000e-004
1.600000000000000e+001	1.920000e+000	-3.520000e-004
1.700000000000000e+001	2.040000e+000	-3.740000e-004
1.800000000000000e+001	2.160000e+000	-3.960000e-004
1.900000000000000e+001	2.280000e+000	-4.180000e-004
2.000000000000000e+001	2.400000e+000	-4.400000e-004
2.100000000000000e+001	2.520000e+000	-4.620000e-004
2.200000000000000e+001	2.640000e+000	-4.840000e-004
2.300000000000000e+001	2.760000e+000	-5.060000e-004
2.400000000000000e+001	2.880000e+000	-5.280000e-004
2.500000000000000e+001	3.000000e+000	-5.500000e-004
2.600000000000000e+001	3.120000e+000	-5.720000e-004
2.700000000000000e+001	3.240000e+000	-5.940000e-004
2.800000000000000e+001	3.360000e+000	-6.160000e-004
2.900000000000000e+001	3.480000e+000	-6.380000e-004
3.000000000000000e+001	3.600000e+000	-6.600000e-004

Figure 1.15: Table with results of dc sweep simulation for circuit example from Fig. 1.4.

Sweeping resistor values: Instead of showing variations in the circuit of Fig. 1.1 when sweeping the voltage source V_S, you might be interested in analyzing the circuit when sweeping a resistor value, e.g., the value of R_1. This can be achieved by specifying the value of R_1 as a variable parameter. To do so, the specification for R_1 should be changed on the resistor symbol. Instead of specifying the value '`40k`', the value must be specified to be '`{R1}`' (remember to include the curly brackets '`{}`'). Now you can specify a sweep range for the parameter '`R1`' by inserting a '.step' command: Click 'Edit → SPICE Directive' (or .op on the toolbar), and a dialogue window appears in which you can type a command. Alternatively, a right-click in the dialogue window opens a 'Help me Edit' option where you can select '.step Command'. If you prefer to type in the command directly in the 'Edit' dialogue window, you can close the 'Help' window by a left-click on 'Cancel' or by pressing 'Escape'. This will bring you back to the 'Edit' dialogue window. Insert the command '`.step param R1 30k 50k 2k`'. This will sweep the value of R_1 from 30 kΩ to 50 kΩ in steps of 2 kΩ. Finally, run a '.op' simulation.

If there are no errors in your circuit, the simulation will open a plot window with the resistance range of 30 kΩ to 50 kΩ as the horizontal axis. You may select voltages and currents to be displayed in the same way as for the dc sweep simulations. Figure 1.16 shows the schematic from Fig. 1.4 with the '.step' directive inserted, and it shows the resulting waveform plot of V_2. Here, the color preferences of both the waveform plot and the schematic have been modified to get a white background and black axes on the waveform plot. Also, rather than using the autorange scaling of the vertical axis, the axis has been modified to the range from 0 V to 3.5 V. This can be done by the command 'Plot Settings → Manual Limits' or by moving the mouse cursor over the axis and left-clicking. In Fig. 1.16 (and in subsequent figures showing simulation plots), the font size of the labels on the axes has been increased using the

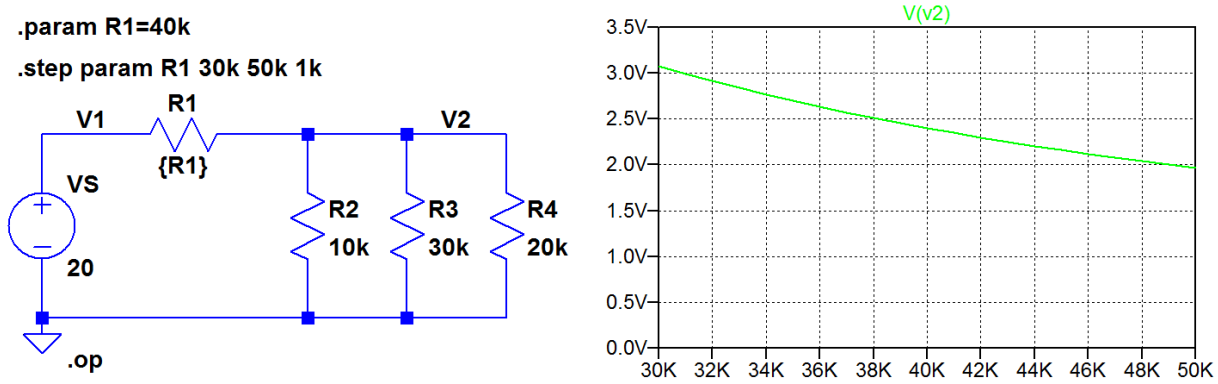


Figure 1.16: Simulation of sweep of resistor R_1 from Fig. 1.1.

command 'Tools → Control Panel' and the tab 'Waveforms' where the font has been changed to Arial and the fontsize to 18 points.

In a waveform plot, you can insert text and other annotations (e.g., cursor position) using the command 'Plot Settings → Notes & Annotations'. You may note that many of the commands described for the waveform plot are also available using a right-click in the plot. This opens a window with several commands and subwindows with even more commands.

In the plot window, you can also zoom in on details simply by clicking and dragging to define a box using the left mouse button.

For running a simulation with just one value for a variable parameter ('`R1`' in Fig. 1.16), the value of '`R1`' must be specified using a '.param' directive: Insert the SPICE directive '`.param R1=40k`' to run a simulation with $R_1 = 40$ kΩ and delete the '.step' directive or edit it into a comment by inserting an asterix (*) as the first character or by ticking 'Comment' in the editing window (after having closed the 'Help me Edit' window by a left-click on 'Cancel'). If the '.step' directive is not disabled, the simulation will run this command regardless of the '.param' specification.

The '.step' directive is a very useful command for design iterations. By defining relevant design parameters as variable parameters and stepping the values over a suitable range, you can quickly examine the influence of a parameter on the circuit characteristics. When used in combination with a '.measure' directive (see Example 2.5), accurate solutions to a design problem may easily be found. Problems 1.2, 1.5, 1.9 and 1.10 are examples of this.

Example 1.2: A transconductance amplifier.

The next example is a circuit containing a voltage-controlled current source as shown in Fig. 1.17. Essentially, this is an inverting transconductance amplifier with an input resistance R_{in}, an output resistance R_o and a transconductance g_m. In Fig. 1.17, a load resistor R_L and a signal source V_S with a source resistance R_S is connected to the amplifier.

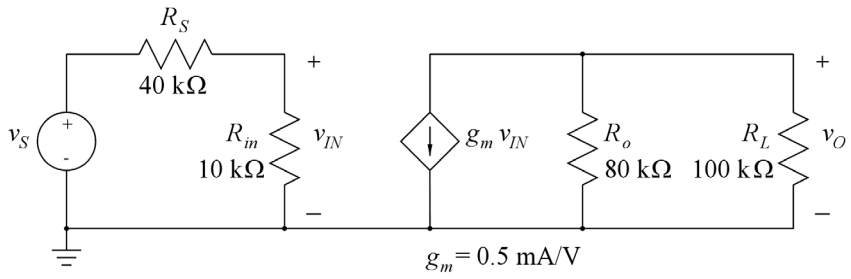


Figure 1.17: An inverting transconductance amplifier.

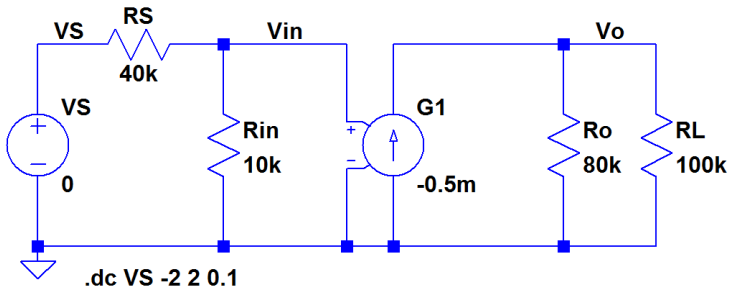


Figure 1.18: LTspice schematic for the inverting transconductance amplifier.

In this circuit, there is a new type of component, the voltage-controlled current source. LTspice has, like other Spice programs (Tuinenga 1995; Vladimirescu 1994), a voltage-controlled current source as a standard component with the circuit designator 'G'.

The LTspice schematic is shown in Fig. 1.18. The LTspice symbol for the voltage-controlled current source explicitly shows the controlling voltage as input terminals to the component symbol. In the component selection box (Fig. 1.3), you may select either 'g' or 'g2', the only difference being the polarity of the controlling voltage. The controlled current source is edited by right-clicking on the symbol. This opens a 'Component Attribute Editor' as shown in Fig. 1.19.

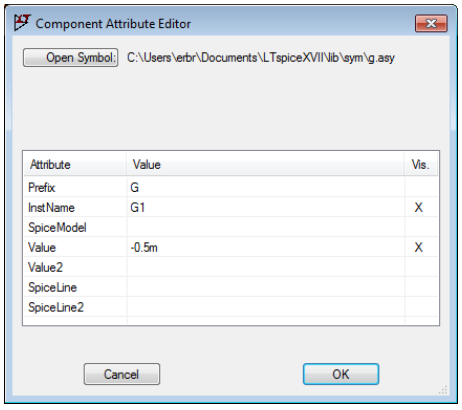


Figure 1.19: The window for editing the specifications of the voltage-controlled current source.

By double-clicking on the values for 'InstName' and 'Value', the values can be changed to the values shown in Fig. 1.18. Alternatively, just right-click on the device name (e.g., '`G1`') and the value '`G`' to edit them to the desired values in the same way as editing the value of a resistor or a dc current source.

After inserting a simulation command, you may now run a dc sweep simulation, e.g., sweeping v_S from -2 V to +2 V. The resulting plot of v_O versus v_S may look like shown in Fig. 1.20.

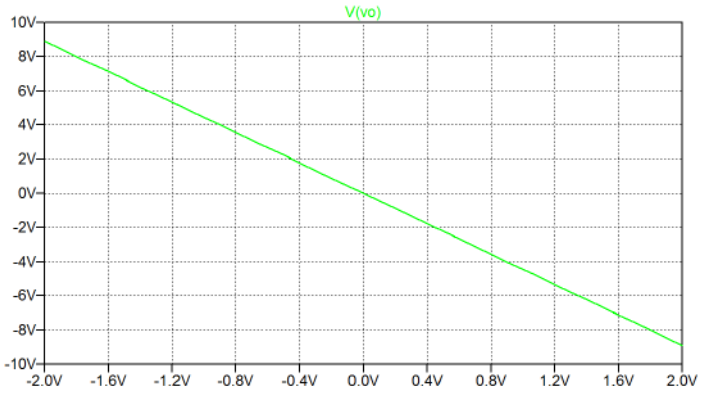
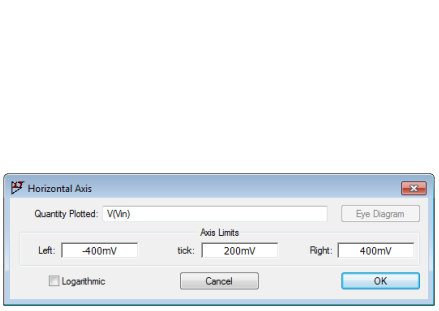


Figure 1.20: Plot of v_O versus v_S for the inverting amplifier.

For plotting v_O versus v_{IN}, you just move the cursor in the plot window to the x-axis and apply a right-click on the mouse. This opens a specification window for the x-axis as shown in Fig. 1.21 where you can change the 'Quantity Plotted' from '`Vs`' to '`V(vin)`', resulting in a plot of v_O versus v_{IN}. After having changed the 'Quantity Plotted' to '`V(vin)`' and clicked 'OK', you may re-open the specification window for the x-axis and change the axis limits to the range from -400 mV to $+400$ mV as shown in Fig. 1.21.



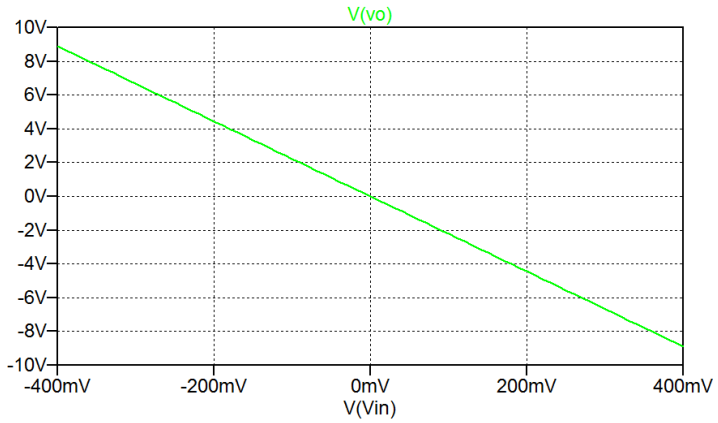


Figure 1.21: Specification window for the horizontal axis and plot of v_O versus v_{IN} for the inverting amplifier.

The arbitrary behavioral source: LTspice also provides an alternative to the voltage-controlled current source. This is an 'Arbitrary behavioral current source', device type 'bi' in the component selection. The same device can be used for both a voltage-controlled current source and a current-controlled current source. Figure 1.22 shows the circuit from Fig. 1.17 redrawn with the 'bi' symbol. Notice the definition

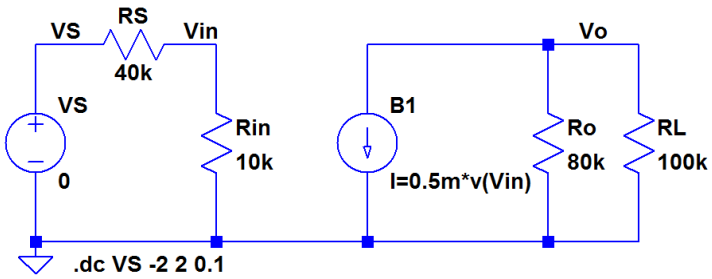


Figure 1.22: LTspice schematic for the inverting transconductance amplifier using an arbitrary behavioral current source.

line for the current source: '`I=0.5m*v(Vin)`'. You need to specify the controlling voltage as '`v(Vin)`', not just '`Vin`', otherwise you will receive an error message. Also note that the asterix (`*`) is the character indicating multiplication.

In Fig. 1.22, the symbol for the controlled current source is a circle, exactly like the symbol for an independent current source. Often in the literature, controlled sources are represented by a diamond-shaped symbol to distinguish them from the independent sources (Hambley 2018; Sedra & Smith 2016). You may actually edit the symbol for the controlled current source using the symbol editor in LTspice. When you are in the 'Component Attribute Editor' (Fig. 1.19), you click 'Open Symbol' to enter the symbol editor where you can redraw the shape of the symbol. It is a good idea to save your own symbols in a dedicated folder for this, rather than just saving the modified symbol in the default symbol library

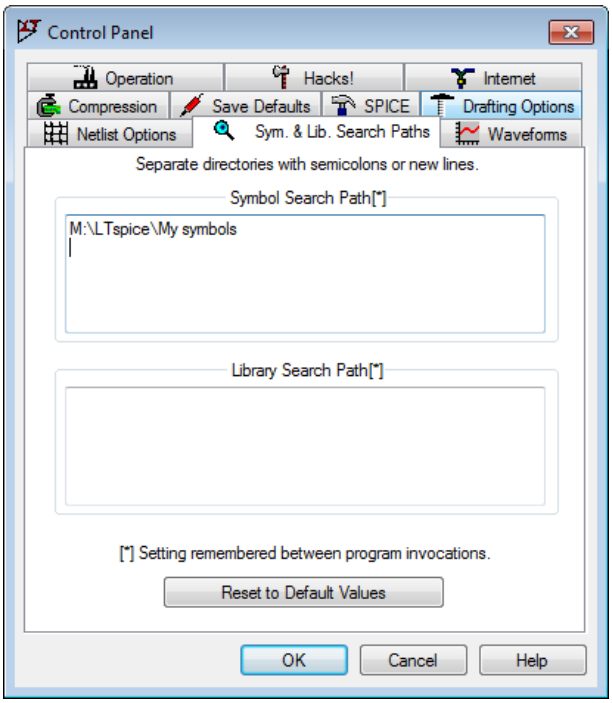


Figure 1.23: Window for defining search path to symbol folders and library folders.

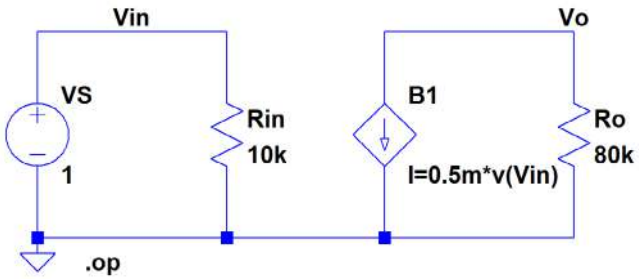


Figure 1.24: The inverting amplifier with a diamond-shaped symbol for the arbitrary behavioral current source.

which is updated every time the LTspice installation is updated. You can define a path to your own symbol folder (e.g., 'My symbols') using the command 'Tools → Control Panel → Sym. & Lib. Search Paths', see Fig. 1.23 where a path to a symbol library named 'M:\LTspice\My symbols' has been defined. When inserting a symbol from your own symbol folder, select this folder in the selection window for 'Top Directory' in the component selection box, see Fig. 1.3.

In Fig. 1.24, the transconductance amplifier is redrawn with a diamond-shaped symbol, and the load resistor R_L and source resistor R_S are omitted. The circuit shown has only linear components, and it is easy to see that the input resistance is $R_{in} = 10$ kΩ and the output resistance is $R_o = 80$ kΩ. The open-circuit voltage gain A_{voc} can be calculated from $A_{voc} = -g_m R_o = -40$ V/V, where $g_m = 0.5$ mA/V is the transconductance of the voltage-controlled current source. These values can also be found by simulation: With an input voltage of $v_S = v_{IN} = 1$ V, the output voltage is $v_O = A_{voc} \times 1$ V, so the simulated value of the output voltage directly gives the value of A_{voc}. By changing the input signal to a current source of 1 A, the value of the input voltage is $R_{in} \times 1$ A, so the simulated value of the input voltage directly gives the value of R_{in}. By resetting the input voltage ($v_S = v_{IN} = 0$) and applying a current source of 1 A to the output, the value of the output voltage is $R_o \times 1$ A, so the simulated value of the output voltage directly gives the value of R_o.

Nonlinear controlled current source: Next, we assume that the voltage-controlled current source is given by a nonlinear relation, $I = 0.5$ mA/V$^2 \times v_{IN}^2$ for $v_{IN} \geq 0$ V. The specification for '`B1`' in Fig. 1.24 must then be modified to '`I=0.5m*v(Vin)**2`'. Observe the double asterix (`**`) for raising to power of 2. With $v_{IN} = 1$ V, the '.op' simulation still results in $v_O = -40$ V, but a dc sweep of v_{IN} from 0 V to 2 V shows the nonlinear relation between v_O and v_{IN}, see the green curve in Fig. 1.25.

For this amplifier, the voltage gain is not just v_O/v_{IN}. Rather, the voltage gain is defined as the small-signal gain $A_{voc} = \partial v_O/\partial v_{IN}$ calculated in the bias point of the amplifier.

For an input bias voltage of $V_{IN} = 1$ V, we find $A_{voc} = \partial v_O/\partial v_{IN} = -R_o \times 2 \times 0.5$ mA/V$^2 \times V_{IN} = -80$ V/V. Clearly, the gain depends on the bias value of the input voltage. The voltage gain is also seen as the slope of the nonlinear relation between v_O and v_{IN}. This slope can be displayed directly in the plot window: When you click on the command 'Plot Settings → Add trace' (or hotkey 'Ctrl-A'), a window

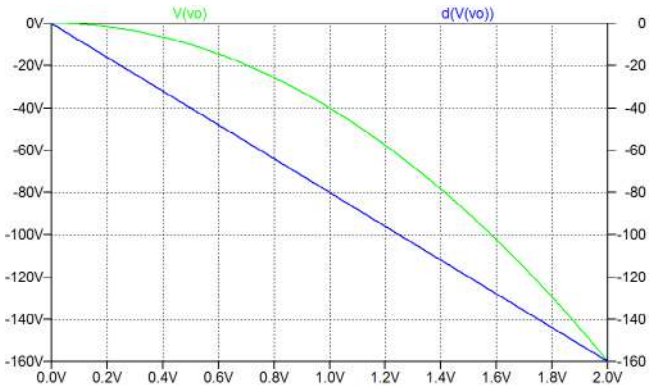


Figure 1.25: Plot of v_O versus v_S for the inverting amplifier with a nonlinear voltage-controlled current source.

opens for specifying traces to plot. The bottom line in this window lets you enter an expression to add. A large selection of mathematical operations is available (see the 'Help' menu, 'Waveform Arithmetic'), including the derivative of a variable with respect to the x-axis variable. The function '`d(V(vo))`' will give you the derivative of the output voltage with respect to the input voltage. The resulting plot is the blue line in Fig. 1.25 from which you can see that $A_{vo\mathrm{c}} = -80$ V/V as expected for $V_{IN} = 1$ V. Actually, '`d(V(vo))`' is calculated as a difference-based derivative, so in order to obtain a smooth curve as shown in Fig. 1.25, you have to use a small step size for v_S. For the plot in Fig. 1.25, a step size of 0.01 V has been applied. Using a step size of 0.1 V instead will give a staircase curve for '`d(V(vo))`'.

LTspice has another simulation command which directly gives you the small-signal transfer function at dc, the 'DC Transfer' simulation. Use the command 'Simulate → Edit Simulation Command' and choose the tab 'DC Transfer'. Here you specify the output and the source. For the circuit of Fig. 1.24, the output is '`v(Vo)`' (not just '`Vo`') and the source is '`VS`'. The resulting simulation command is '`.tf v(Vo) VS`', and after running the simulation (with '`I=0.5m*v(Vin)**2`'), a window opens with the information shown in Fig. 1.26.

Output from dc transfer simulation

```
        --- Transfer Function ---

Transfer_function:              -80             transfer
vs#Input_impedance:             10000           impedance
output_impedance_at_V(vo):      80000           impedance
```

Figure 1.26: Output from '.tf' simulation of the inverting amplifier with a nonlinear voltage-controlled current source.

Example 1.3: A current amplifier.

The next example in this tutorial is a current amplifier as shown in Fig. 1.27. The gain element in this circuit is a current-controlled current source. The current amplifier has an input resistance R_{in}, a short-circuit current gain $A_{i\mathrm{sc}}$ and an output resistance R_o.

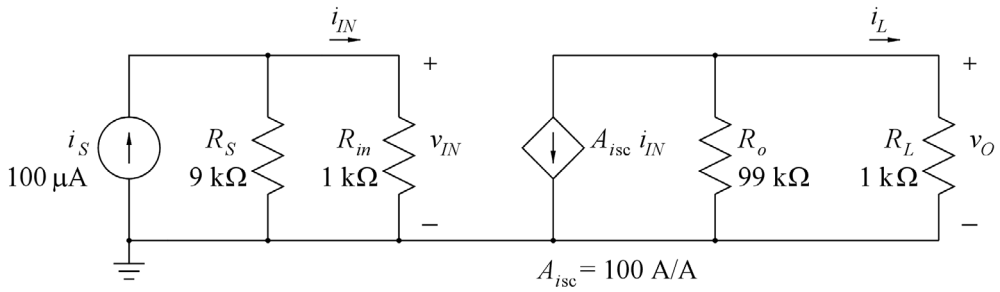


Figure 1.27: An inverting current amplifier.

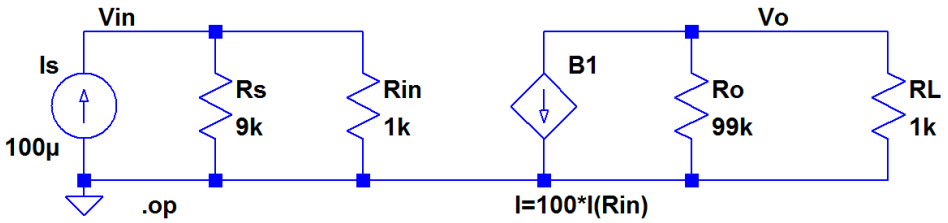


Figure 1.28: LTspice schematic for the inverting current amplifier.

A simple examination of the circuit shows an inverting current gain from the input signal i_S to the current i_L in the load resistor of A_{isc} multiplied by the current-divider ratios at the input side and the output side. With the values shown in Fig. 1.27, we find $i_L/i_S = -89.1$ A/A. For $i_S = 100$ µA, we get an output current $i_L = -8.91$ mA and an output voltage of $v_O = -8.91$ V.

In LTspice, the current-controlled current source is described either by the device type 'F' or by the 'Arbitrary behavioral current source', device type 'bi' in the component selection. Figure 1.28 shows the schematic drawn with the arbitrary behavioral current source (using a diamond-shaped symbol). Obviously, when examining Fig. 1.27, the current is controlled by the current through R_{in}, so an immediate specification for B1 would be '`I=100*I(Rin)`' as shown in Fig. 1.28. Running a '.op' simulation indeed also results in the expected values of i_L and v_O.

But running a '.tf' simulation with '`Is`' as the source and '`v(Vo)`' as the output gives a transfer function of 0 which is obviously not correct. The value to expect is $v_O/i_S = -8910$ V/A. The input resistance and the output resistance from the '.tf' simulation are shown as 900 and 990, respectively, which is as expected since the input side is a parallel connection of 1 kΩ and 9 kΩ and the output side is a parallel connection of 1 kΩ and 99 kΩ. Trying a '.tf' simulation with '`I(RL)`' as the output also results in a transfer function of 0. The reason for these errors is that some of the analyses in LTspice (e.g., '.tf' and '.ac' (see Tutorial 2)) require that a current is specified as a current through a voltage source.

Figure 1.29 shows the circuit from Fig. 1.28 redrawn with dc voltage sources of 0 V in series with R_{in} and R_L and B1 specified as '`I=100*I(V1)`'. The '.op' simulation still provides the correct result, and now also both '.tf' simulations with '`v(Vo)`' and '`i(V2)`' as the output show the expected gain. The input resistance is found from both '.tf' simulations, but the output resistance is found only from the simulation with '`v(Vo)`' as the output.

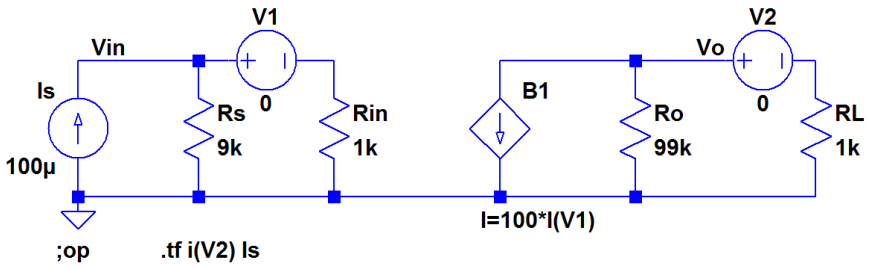


Figure 1.29: LTspice schematic for the inverting current amplifier with voltage sources in series with R_{in} and R_L and altered specification for B1.

Now, let us connect a feedback resistor R_f of 15 kΩ between output and input as shown in Fig. 1.30, shunt - shunt feedback (Sedra & Smith 2016). With this feedback resistor, the amplifier is turned into a transresistance amplifier. With a very large current gain A_{isc}, we would expect a transresistance equal to $-R_f$ and small values of input and output resistance. The '.tf' simulation with v_O as the output shows a gain (transresistance) of -12.6 kΩ, an input resistance of 136 Ω and an output resistance of 149 Ω (including R_s and R_L). Increasing A_{isc} to 1000, we find a gain very close to -15 kΩ and input and output resistances in the range of 1 to 2 Ω.

Next, let us see what happens if we change the specification of the current-controlled current source to '`I=100*I(Rin)`'. Then we find that neither the '.op' simulation, nor the '.tf' simulations will run. They both return the error message 'Analysis failed: Iteration limit reached'. This shows that LTspice is unable to find the bias point from the '.op' simulation when the current is not specified as the current through a voltage source. In other examples, the operating point may be found but with reduced precision if the current is specified as the current in a resistor. The circuit shown in Problem 1.3 is an example of such a circuit.

The lesson learned from this example is: The controlling current for a current-controlled voltage source or a current-controlled current source must be the current through an independent voltage source. Insert a dc voltage source of 0 V in series with the device carrying the controlling current and use the current in this voltage source as the controlling current.

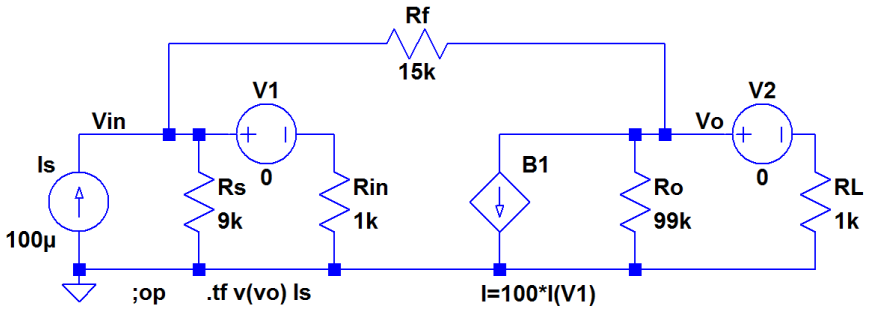


Figure 1.30: LTspice schematic for the inverting current amplifier with a feedback resistor R_f.

This is also the way to specify a controlling current when using the current-controlled current source with circuit designator 'F'. Figure 1.31 shows the circuit from Fig. 1.30 redrawn with the device 'f' instead of 'bi' and also shows the specification window for 'f'. In this window, the name of the dc voltage source for the controlling current must be specified in the line 'Value', and the current gain must be specified in the line 'Value2'. In order to make the current gain visible on the schematic, an X has been inserted in the rightmost column ('Vis.') in the specification line for 'Value2'.

Example 1.4: Debugging a schematic.

Although LTspice is easy to use and you quickly learn how to draw schematics and simulate circuits, you cannot expect to get everything correct every time you try a new circuit. Some errors in the schematic are fatal in the sense that they prevent the simulation from running. Somehow, these are not the worst errors because they are so obvious. Other errors may not prevent the simulation from running but they will lead to incorrect simulation results. Unless you notice such errors, they can be even more detrimental that the fatal errors preventing the simulation. Sometimes warnings appear in the error log file after the simulation, and it is always a good idea to examine the error log file after your simulations.

Also, it is always a good idea to consider whether your simulation results seem reasonable, or if some values are way off from what you expected. If so, it may be caused by incorrect inputs to your simulation, e.g., incorrect component values, or as we have seen in Example 1.3, it may be caused by an error such as using a current through a resistor rather than through an independent voltage source to control another voltage or current. This is an error which does not generate a warning in the error log file.

In this example, we show an LTspice schematic with a number of different kinds of errors and we show how the errors are identified and corrected by examining the error messages, the output files and error log files. Typically, you would not find all these kinds of errors in one schematic. They are shown to illustrate how they are reported in the error log file. The starting point is the inverting transconductance amplifier shown in Fig. 1.17. Figure 1.32 shows this amplifier drawn in LTspice but with a number of errors.

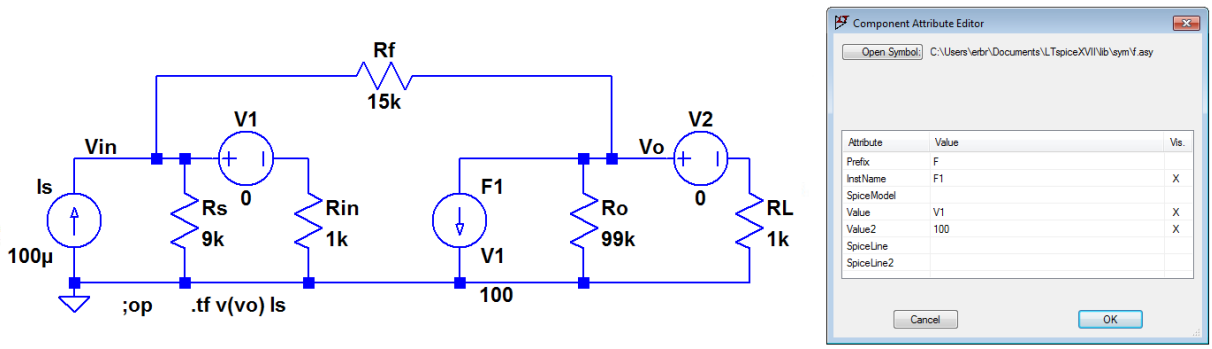


Figure 1.31: Circuit from Fig. 1.30 redrawn with a current-controlled current source instead of an arbitrary-controlled current source.

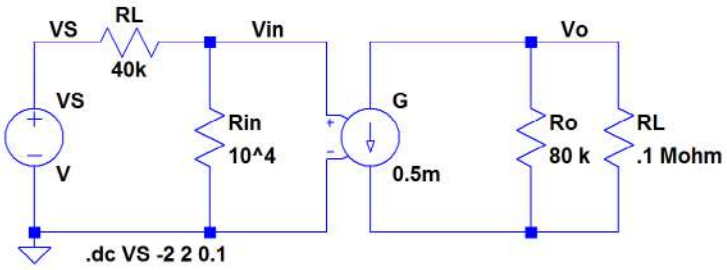
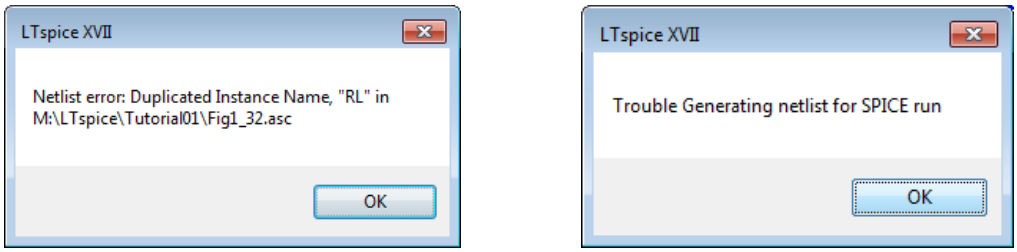


Figure 1.32: LTspice schematic **WITH ERRORS** for the inverting transconductance amplifier from Fig. 1.17.

Compared to the correct version shown in Fig. 1.18, an obvious difference is that the symbol for the controlled current source now has the arrow pointing downwards, just as in Fig. 1.17. This is achieved by selecting the component symbol 'g2' when inserting the controlled current source. It has to be rotated twice and mirrored when inserted in order to have the arrow pointing downwards and the input side to the left. The simulation specified is the same dc sweep as in Fig. 1.18.

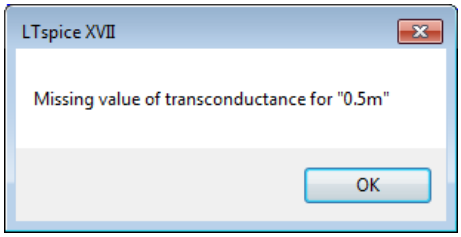


(a) (b)

Figure 1.33: Error messages from the simulation of the circuit in Fig. 1.32.

Now, let us try to run the simulation. This just opens the window shown in Fig. 1.33(a), and the simulation does not run. When closing the window, a new window opens, see Fig. 1.33(b). Obviously, there is an error in the labeling of the resistors. When examining the schematic in Fig. 1.32, we notice that both the load resistor and the signal source resistor are labeled 'RL'. This is creates an ambiguity preventing LTspice from generating a netlist, so we must change the label for the signal source resistor to 'RS'.

After having done so, let us try to run the simulation again. This just opens the window shown in Fig. 1.34(a), and the simulation does not run. When closing the window, a new window opens with the error log file, see Fig. 1.34(b). From this, we see that a fatal error is 'Missing value of transconductance for "0.5m"'. Apparently, LTspice does not accept the way, the controlled current source has been specified. What went wrong is that the rotations and the mirroring of the symbol 'g2' have swapped the position of the name and the value of the device. In Fig. 1.18, the name appears above the value, but here, it is the opposite. When you have inserted the component, you can actually see which is the name and which is the value: The name by default always has a number, the value does not. Also, you can see on the status bar at the bottom of the LTspice schematic window if you have placed the cursor on the name of a component or on the value.



(a)

LTspice Error Log

```
Error on line 2 : rin vin 0 10^4
                  Unknown parameter "^4"
Error on line 4 : vs vs 0 v
                  Unknown parameter "v"
Error on line 5 : rl vo n001 .1 mohm
                  Unknown parameter "mohm"
Error on line 6 : ro vo n001 80 k
                  Unknown parameter "k"
Error on line 7 : g$0.5m vo n001 vin 0 g
                  Unknown parameter "g"
Fatal Error: Missing value of transconductance for "0.5m"
```

(b)

Figure 1.34: More error messages from the simulation of the circuit in Fig. 1.32.

As you can see from Fig. 1.34(b), there are some other errors as well, but before considering these, let us correct the fatal error. We simply swap the name and the value of the controlled current source and run the simulation again. This time, the simulation runs and opens a window for plotting the simulation results. But also the error log file opens. However, since the simulation actually did run, we may use the plot window for plotting v_O versus v_S in the same way as in Fig. 1.20. The resulting plot is shown in Fig. 1.35.

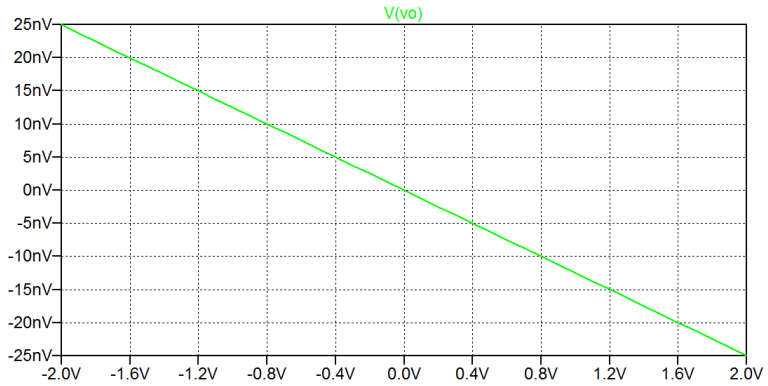


Figure 1.35: Plot of v_O versus v_S for the inverting amplifier with errors in the schematic.

When comparing this plot to Fig. 1.20, it is quite evident that something is still very wrong. The y-axis scaling is very different, so it is a good idea to re-open the error log file. This is done by the command 'View → SPICE Error Log' or by typing 'Ctrl-L', and the error log file shown in Fig. 1.36 appears.

All the errors listed in the error log file shown in Fig. 1.34(b) are still there. Obviously, they have to do with the specification of the voltage source v_S and the resistors R_{in}, R_o and R_L. For v_S, the specification of the voltage has been forgotten. It is still just the 'V' which appeared when the voltage source symbol was inserted in the schematic. LTspice runs the simulation anyway, assuming a specification of 0 V, and the value does not really matter for the simulation specified because v_S is swept from -2 V to 2 V. However, if you forget the specification of a resistor value, it is considered a fatal error and the simulation does not run.

LTspice Error Log

```
Error on line 2 : rin vin 0 10^4
                  Unknown parameter "^4"
Error on line 4 : vs vs 0 v
                  Unknown parameter "v"
Error on line 5 : rl vo n001 .1 mohm
                  Unknown parameter "mohm"
Error on line 6 : ro vo n001 80 k
                  Unknown parameter "k"
ERROR: Node N001 is floating and connected to current source G

Vs: Missing value, assumed 0V @ DC
```

Figure 1.36: Error messages from the simulation of the circuit in Fig. 1.32 after having corrected the specification of the controlled current source.

For the resistors, the error log file reports unknown parameters. Looking in detail at each resistor, we see that for R_{in}, the problem is the notation '10^4'. This is treated by LTspice as a value of 10 Ω and the following '^4' is considered an unknown parameter. The correct way of specifying a value of '10^4' is '1e4'. For the resistor R_o, the error in the specification is the space between '80' and 'k'. This causes LTspice to assume a resistor value of 80 Ω and the suffix 'k' is just neglected as an unknown parameter. For R_L, there is also a space between the value and the suffix, causing LTspice to assume a value of 0.1 Ω. Notice that it is acceptable to omit the '0' before the decimal point. If you try to correct the value of R_L just by deleting the space before the suffix, you will observe that LTspice changes 'Mohm' into 'mohm', implying that instead of specifying '0.1 Megaohm', you have specified '0.1 milliohm'. The suffix must be changed to 'Meg' in order to obtain the desired value.

With all the resistors corrected, you may now run a simulation and it will open the plot window where you can specify a plot of v_o versus v_S and it looks exactly like the plot in Fig. 1.20. However, the error log file is also automatically opened and it provides an error message that node N001 is floating. It does not prevent LTspice from running the simulation, and since the output voltage appears in the plot window to be the same as in Fig. 1.20, LTspice must have assumed a reasonable value for the floating node. Looking in detail at the schematic in Fig. 1.32, you may identify the problem with the floating node: There is no connection between ground on the input side of the amplifier and ground on the output side of the amplifier.

The impact of the missing ground connection may be analyzed by plotting the voltage of node N001, ground on the output side. This plot is shown in Fig. 1.37, and you can see that the node is not really at ground but shows some fluctuations, so the final step in the debugging of the circuit is to insert the missing connection to ground on the output side, and the resulting schematic is shown in Fig. 1.38.

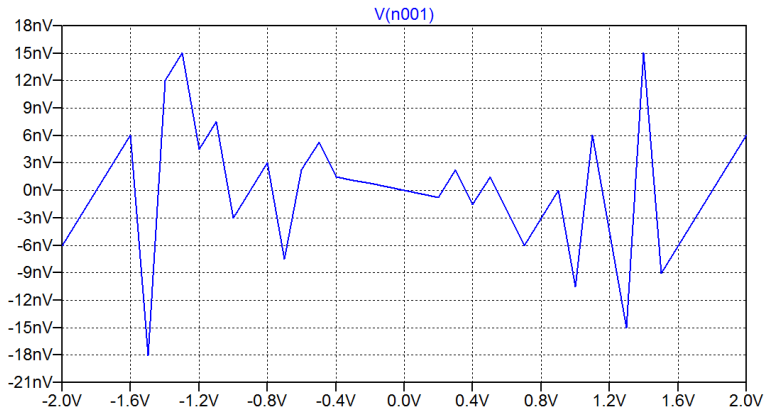


Figure 1.37: Plot of the floating output ground voltage versus v_S for the inverting amplifier with errors in the schematic.

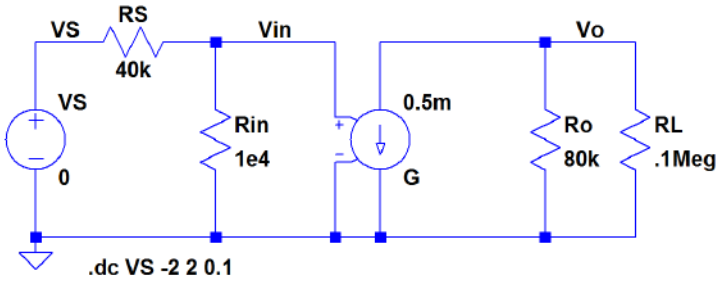


Figure 1.38: LTspice schematic with all errors from Fig. 1.32 corrected.

Finally, we show here the netlists both for the circuit from Fig. 1.32 (with the labeling of the signal source resistor corrected to 'RS') and the circuit from Fig. 1.38. You will observe that the differences in the two netlists can all be related to the error messages shown in Figs. 1.34 and 1.36.

LTspice netlist

```
* M:\LTspice\Tutorial01\Fig1_31.asc
Rin Vin 0 10^4
RS Vin VS 40k
VS VS 0 V
RL Vo N001 .1 Mohm
Ro Vo N001 80 k
G$0.5m Vo N001 Vin 0 G
.dc VS -2 2 0.1
.backanno
.end
```

LTspice netlist

```
* M:\LTspice\Tutorial01\Fig1_36.asc
Rin Vin 0 1e4
RS Vin VS 40k
VS VS 0 0
RL Vo 0 .1Meg
Ro Vo 0 80k
G Vo 0 Vin 0 0.5m
.dc VS -2 2 0.1
.backanno
.end
```

Figure 1.39: LTspice netlists for circuits from Fig. 1.32 (left) and Fig. 1.38 (right).

Hints and pitfalls

- The suffix for 'milli' is 'm'. The suffix for 'Mega' is 'meg' (or 'Meg'). After the suffix, you may insert the unit (e.g., A for ampere). An alternative suffix is 'e' followed by the power of 10, e.g., 'e-3' for 'milli'. You cannot use '∗10^-3' as a suffix for 'milli'.
- Do NOT insert a space between a component value and the suffix or unit.
- Always define a ground node in your circuit.
- Many commands can be selected either via a command and subcommand (e.g. 'Edit → Resistor'), a toolbar symbol (e.g., ), or a hotkey (e.g., R). The assignment of hotkeys can be seen using the command 'Tools → Control Panel → Drafting Options → Hotkeys'.
- A right-click on the schematic drawing opens a menu with several sub-menus. The 'Draft' sub-menu allows you to insert 'Components', 'Wires', 'Net Names', 'SPICE directives', etc.
- A right-click in the window for entering a SPICE directive opens a 'Help me Edit' option.
- The commands 'Drag', 'Move', 'Duplicate' and 'Delete' work not only on single symbols. When you have activated one of the commands, you can define a box by clicking and dragging using the left mouse button. The command works on the entire contents of the box.
- When you have several identical components in your circuit, it is convenient to edit just one instance of the component to the correct value and then use the 'Duplicate' command (F6), rather than inserting and editing each component individually.
- See Problem 1.8 for more hints on drawing schematics.
- When you move the mouse cursor to a component symbol or text, the status bar at the bottom of the LTspice program window gives information about editing options.
- Color preferences can be edited for both schematics and waveforms using the command 'Tools → Color Preferences'.
- Font sizes on schematics and waveform plots can be modified using the command 'Tools → Control Panel' and the appropriate tab (e.g., 'Drafting Options' or 'Waveforms').
- If you have closed a window with results (e.g., from a '.op' simulation or a waveform plot), you can re-open it by the command 'View → Visible Traces', toolbar symbol  .
- A right-click on a waveform plot opens a menu with several sub-menus.
- In a waveform plot, you can zoom in on details by clicking and dragging to define a box using the left mouse button.
- Schematics and waveform plots can be copied to the clipboard with the command 'Tools → Copy bitmap to Clipboard' and then pasted into another program (e.g., Microsoft Word).
- The controlling current for a current-controlled voltage source or a current-controlled current source must be the current through an independent voltage source. Insert a dc voltage source of 0 V in series with the device carrying the controlling current and use the current through this voltage source as the controlling current.

References

Hambley, AR. 2018, *Electrical Engineering, Principles and Applications*, Seventh Edition, Pearson Education Ltd., Harlow, UK.

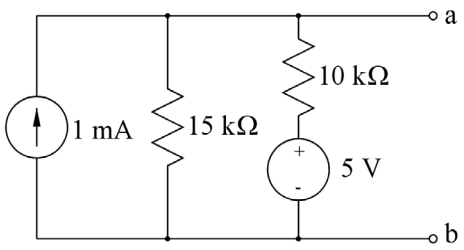
Sedra, AS. & Smith, KC. 2016, *Microelectronic Circuits*, International Seventh Edition, Oxford University Press, New York, USA.

Tuinenga, PW. 1995, *Spice: A Guide to Circuit Simulation and Analysis Using PSpice*, Third Edition, Prentice Hall, Upper Saddle River, USA.

Vladimirescu, A. 1994, *The SPICE book*, First Edition, John Wiley & Sons, Hoboken, USA.

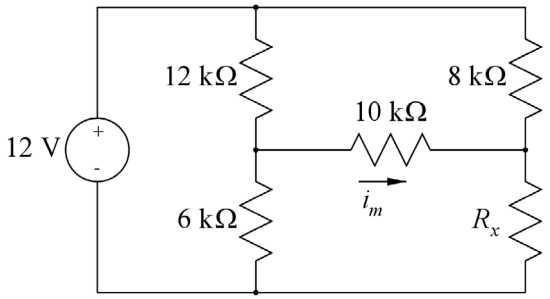
Problems

Problem 1.1



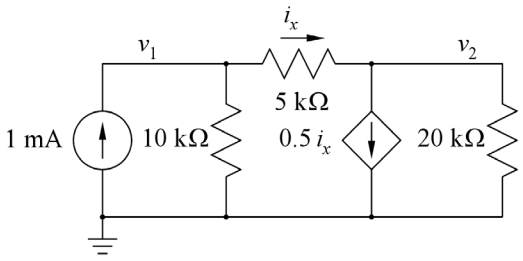
For the circuit shown above, find the Thévenin voltage V_t and the Thévenin resistance R_t. A load resistor of $R_L = 3$ kΩ is now connected between the terminals a and b. Find the power dissipated in R_L.

Problem 1.2



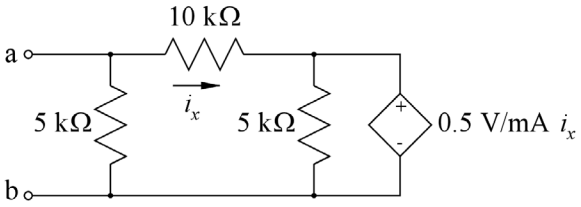
For the circuit shown above, determine the value of resistor R_x so that the current i_m in the 10 kΩ resistor is 30 μA.

Problem 1.3



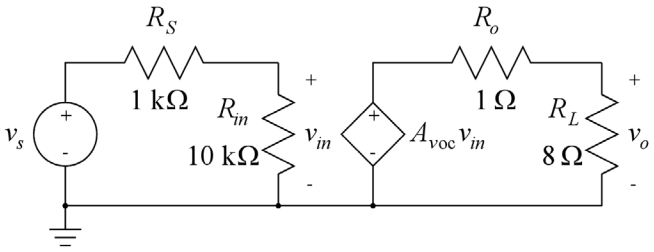
For the circuit shown above, determine the value of the voltages v_1 and v_2 and the current i_x.

Problem 1.4



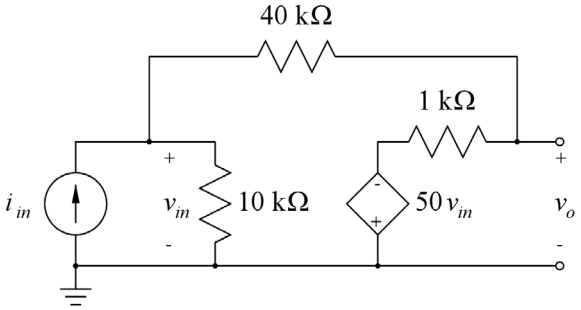
For the circuit shown above, find the equivalent resistance looking into terminals a and b.

Problem 1.5



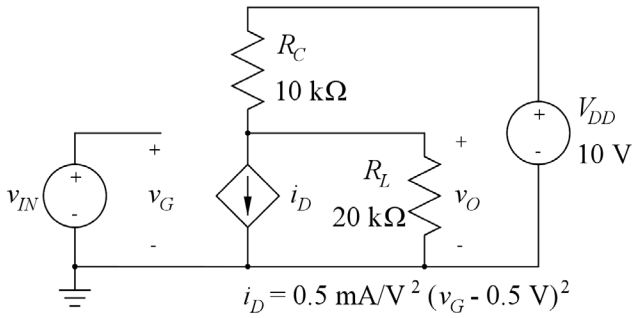
For the circuit shown above, find the value of the gain A_{voc} which gives an output power in R_L of 1 W when the signal voltage v_s is 50 mV. With this value of A_{voc}, plot the output power versus the signal voltage v_s for v_s in the range from 0 mV to 100 mV.
Also plot the output power versus the input voltage v_{in} for v_s in the range from 0 mV to 100 mV.

Problem 1.6



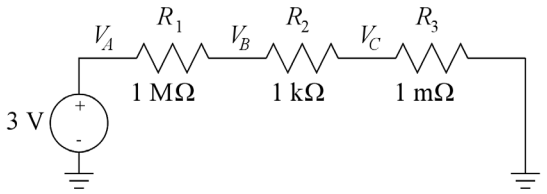
The circuit shown above is a transresistance amplifier built from an inverting voltage amplifier with an input resistance of 10 kΩ, an output resistance of 1 kΩ and an open-circuit voltage gain of −50 V/V and a feedback resistor with a value of 40 kΩ. Find the open-circuit transresistance R_{moc}, the input resistance R_{in} and the output resistance R_o of the resulting transresistance amplifier.

Problem 1.7



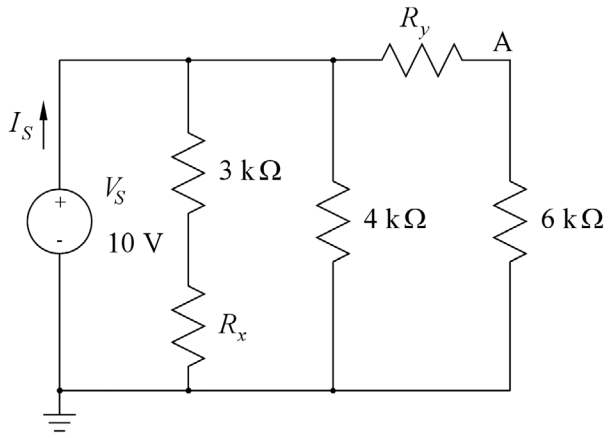
The figure above shows a nonlinear transconductance amplifier. Find the values of bias voltages and currents for an input bias voltage (quiescent voltage) of $V_{IN} = 1.0$ V. Plot the output voltage v_O for the input voltage in the range from 0.5 V to 1.8 V. Find the small-signal voltage gain v_o/v_{in} for an input bias voltage of $V_{IN} = 1.0$ V and plot the small-signal voltage gain as a function of the input bias voltage for the input bias voltage in the range from 0.5 V to 1.8 V.

Problem 1.8



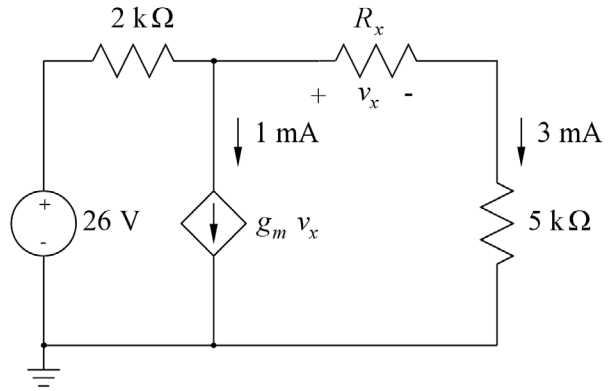
The figure above shows a series connection of three resistors and a voltage source.
Try three different ways of drawing the schematic:
(1): Insert the components and draw the connections between them.
(2): Insert the components (including the ground symbols) and draw an unbroken wire (hotkey 'F3') from the leftmost ground symbol across the components to the rightmost ground symbol.
(3): Insert the ground symbols, draw an unbroken wire between them, and then insert the components directly on top of the wire.
Observe how LTspice 'cleans up' the wiring.
Find the voltages V_A, V_B and V_C.

Problem 1.9



The figure above shows a simple resistive circuit with five resistors where only three of the resistor values are known. However, the voltage of node A is $V_A = 3$ V, and the current I_S supplied by the dc voltage source V_S is $I_S = 4.25$ mA.
Use LTspice to find the values of the two resistors R_x and R_y.

Problem 1.10



The figure above shows a simple circuit with three resistors, an independent voltage source and a voltage-controlled current source. The value of one of the resistors and the value of the transconductance of the voltage-controlled current source are unknown. The other devices have the values shown in the figure together with the values of two of the currents in the circuit.
Use LTspice to find the values of the resistor R_x and the transconductance g_m.

Answers

1.1: $V_t = 9$ V; $R_t = 6$ kΩ; $P_{R_L} = 3$ mW.

1.2: $R_x = 3.31$ kΩ.

1.3: $v_1 = 6$ V; $v_2 = 4$ V; $i_x = 0.4$ mA.

1.4: $R_{\text{ab}} = 3.387$ kΩ.

1.5: $A_{v\text{oc}} = 70$ V/V.

1.6: $R_{m\text{oc}} = -36.28$ kΩ; $R_{in} = 744$ Ω; $R_o = 90.7$ Ω.

1.7: Bias point: $I_D = 0.125$ mA; $I_C = 0.417$ mA; $I_L = 0.292$ mA; $V_O = 5.83$ V.
Small-signal voltage gain with $V_{IN} = 1.0$ V: $v_o/v_{in} = -3.33$ V/V.

1.8: $V_A = 3.0000$ V; $V_B = 2.997$ mV; $V_C = 2.997$ nV.

1.9: $R_x = 5$ kΩ; $R_y = 14$ kΩ.

1.10: $R_x = 1$ kΩ; $g_m = 0.333$ mA/V.

Tutorial 2 – Circuits with Capacitors and Inductors

This tutorial introduces the fundamentals of transient simulations and ac simulations. After having completed the tutorial, you should be able to

- specify a transient simulation in LTspice.
- specify an ac simulation in LTspice.
- use the simulation plots for finding circuit properties such as time constants and -3 dB frequencies.
- use simple components specified by a '.model' directive.
- specify initial conditions for capacitors and inductors.
- use the '.measure' directive for finding circuit properties such as time constants.
- simulate complex impedances.

Example 2.1: An *RC* network.

The first example is a simple *RC* network with two resistors and a capacitor as shown in Fig. 2.1:

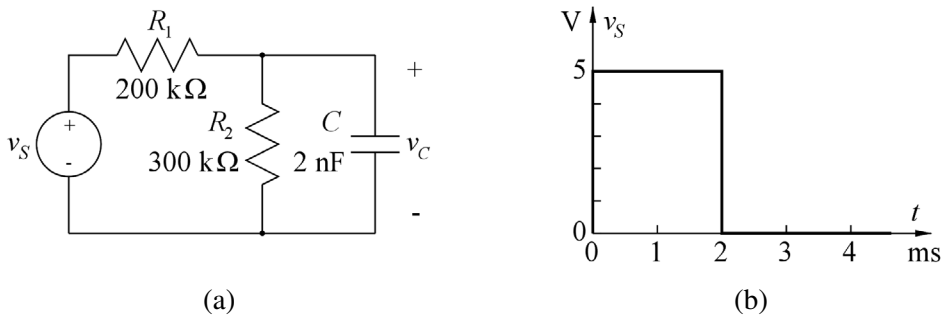


Figure 2.1: *RC* network (a) and input voltage v_S to the network (b).

Transient response: Let us assume that the voltage source v_S is a time-varying voltage as shown in Fig. 2.1(b). The voltage jumps from 0 V to a value of 5 V at the time $t = 0$ s and returns to 0 V at the time $t = 2$ ms. This will cause the capacitor to charge and discharge. For a simulation of the charging and discharging, we will run a transient simulation and specify the voltage source v_S as a time-varying voltage. The circuit is drawn in LTspice using the selection of editing commands as in Tutorial 1. The capacitor is available both as a command, 'Edit → Capacitor', as a toolbar symbol ⫩, and as a hotkey 'C'.

When specifying the voltage source v_S, you point to the centre of the symbol. This turns the cursor into a hand ☜. A right-click opens a window for specifying the voltage source. In this window, left-click on 'Advanced'. This opens a dialogue box as shown in Fig. 2.2 where you may select time-varying functions, e.g., PWL, piecewise linear. A series of boxes for entering times and values will appear. Notice that the voltage cannot be changed abruptly, so the vertical edges shown in Fig. 2.1(b) must have a certain slope. Corresponding to the timing in Fig. 2.1(b), you may enter time 1 = 0s, value 1 = 0V, time 2 = 0.1us, value 2 = 5V, time 3 = 2ms, value 3 = 5V, time 4 = 2.0001ms, value 4 = 0V. In this way, the voltage changes between 0 V and 5 V in 0.1 µs. An alternative to specifying time 4 as 2.0001ms is to specify time 4 = {2ms+0.1us}. Remember to include the curly brackets '{}', otherwise you will receive an error message and the simulation will not run. Also note that you may include the units ('s' for seconds and 'V' for volts). This makes it easier to read the specification shown on the schematic. The specification is shown unless you untick the box for 'Make this information visible on schematic'. This is not recommended. The specification does take up some space on the schematic, but you may move this information to a convenient place in the schematic using the 'Move' or 'Drag' command.

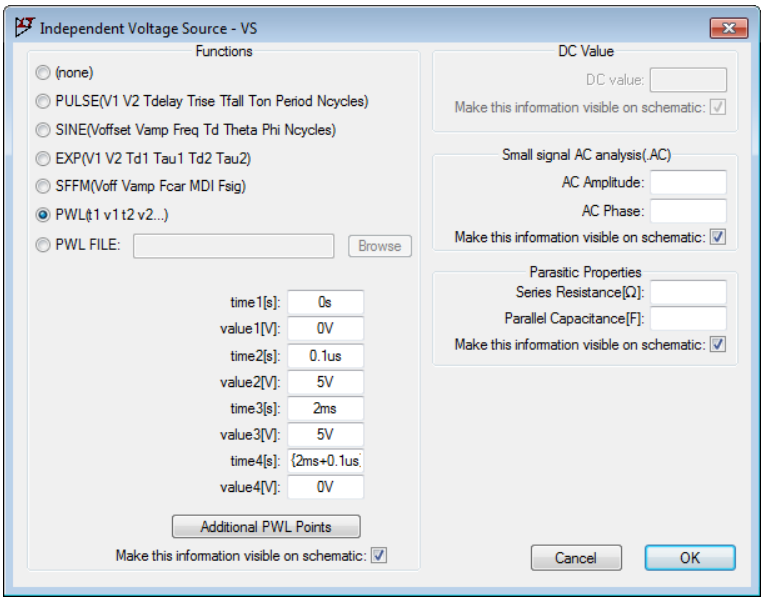


Figure 2.2: Specification window for the voltage source v_S.

Next, you should specify the simulation. Use the command 'Simulate → Edit Simulation Cmd' and open the tab 'Transient', see Fig. 1.5. For a simulation of the charging and discharging of the capacitor, you can run the simulation starting at time $t = 0$ and stopping at the time specified in the box for 'Stop Time'. You can just leave the rest of the specification boxes empty for a simple simulation of the charging and discharging. When inserting a stop time of 4 ms, the transient simulation will show both the charging and the discharging. The circuit is now ready for simulation. If there are no errors, the simulation opens a plot window with a time axis, and by pointing to 'VC' on the schematic (the red pointer,), the capacitor voltage v_C is shown in the plot window. Figure 2.3 shows both the LTspice schematic and the resulting plot window.

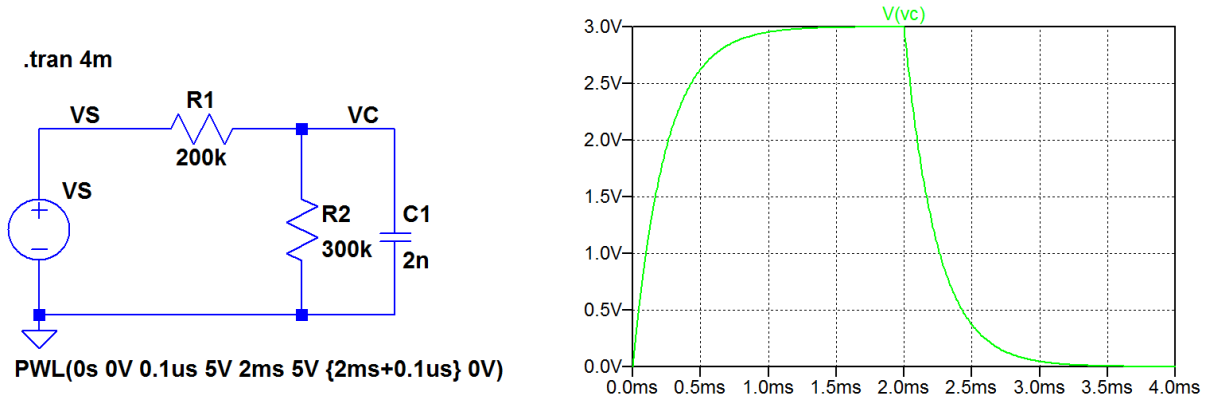


Figure 2.3: LTspice schematic and simulation results for the circuit in Fig. 2.1.

It is evident that the charging and discharging takes place with the same time constant. For the charging, a simple analysis of the circuit gives

$$v_C = V_0\,(1-\exp(-t/\tau)) \tag{2.1}$$

$$V_0 = V_s\,\frac{R_2}{R_1+R_2} = 3\text{ V} \tag{2.2}$$

$$\tau = \frac{1}{(R_1 \parallel R_2)\,C} = 240\ \mu\text{s} \tag{2.3}$$

The time constant can also be found from the simulation of the charging. From $v_C = V_0\,(1-\exp(-t/\tau))$, we find that at $t=\tau$, the voltage is $V_0\,(1-1/e)$. Therefore, if we scale the output by a factor of $[V_0\,(1-1/e)]^{-1}$, the scaled voltage is 1 V when $t=\tau$. By using the command 'Plot Settings → Add trace' (or the hotkey 'Ctrl-A'), you can open a window for selecting traces. Alternatively, if you have already plotted '`V(vc)`', open the waveform editor simply by a right-click on the trace name '`V(vc)`' above the plot. In the waveform editor, you can enter '`V(vc)/3/(1-1/e)`' and click 'OK'. Notice that the waveform editor in LTspice recognizes '`e`' (or '`E`') as the base for the natural logarithm. This generates a new trace in the plot, scaled so that the voltage is 1 V for $t=\tau$. You can find this time by left-clicking on the trace name above the plot to activate a cursor which follows the trace when you move it around by the mouse. Also, a window opens showing the position of the cursor, so you just move the cursor until the vertical position is 1 V and read the horizontal position of the cursor as the value of τ.

Figure 2.4 shows the plot window with the scaled capacitor voltage. In this figure, the x-axis has been scaled to show only the interval from 0 to 2 ms. The figure also shows the window with the coordinates for the position of the cursor. If this window is moved to be inside the simulation plot window, it is copied together with the plot window using 'Tools → Copy bitmap to Clipboard'. Alternatively, it can be copied separately to the clipboard using the standard 'Print screen' function ('Alt-PrtScn'). Using the command 'Plot Settings → Notes & Annotations → Label Curs. Pos.', you can also directly insert the cursor position in the plot as shown in Fig. 2.4.

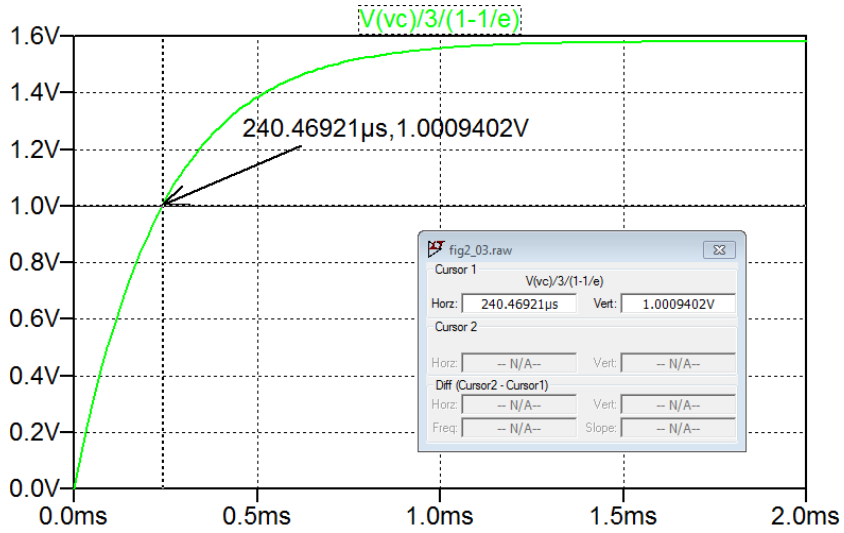


Figure 2.4: Plot window with scaled output for finding the time constant.

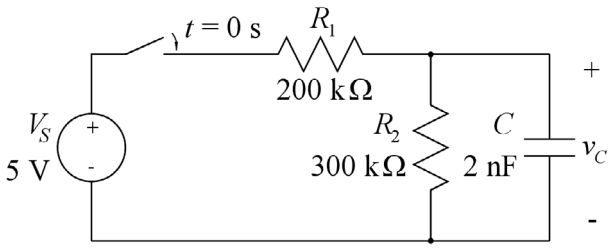


Figure 2.5: The *RC* network from Fig. 2.1 with a switch between the voltage source and the resistor R_1.

Next, we introduce a small change to the circuit: Instead of switching the voltage source v_S between 0 V and 5 V, we insert a switch as shown in Fig. 2.5. The switch is open for $t < 0$, closes when $t = 0$ s, and re-opens at $t = 2$ ms. This will cause the capacitor to charge as before, but the discharge will be only through R_2. In LTspice, the switch can be modeled by the component 'sw' from the component selection  . This is a voltage-controlled switch, so it requires a control voltage to specify the state of the switch.

Unlike a resistor or a capacitor, the switch cannot be specified simply by a value. The properties of the switch are given in a '.model' specification in LTspice. You can find the detailed syntax for the required '.model' specification using the 'Help' function in LTspice.

When you insert the switch in the schematic, it appears with a reference to a default switch model 'SW'. If you wish to change some parameters of the switch, you must include a '.model' specification with the new parameters and it is a good idea to use another name for the model, rather than the default 'SW'. Even if you do not change the parameters from their default values, it is highly recommended to include the '.model' directive. If the '.model' directive is missing, LTspice will prompt a warning that the model cannot be found, and you have the option of continuing with the default model or cancel the simulation

so that you can insert a '.model' directive. If you continue the simulation with the default model, you will receive an error message when the simulation has been completed that the model for the switch could not be found.

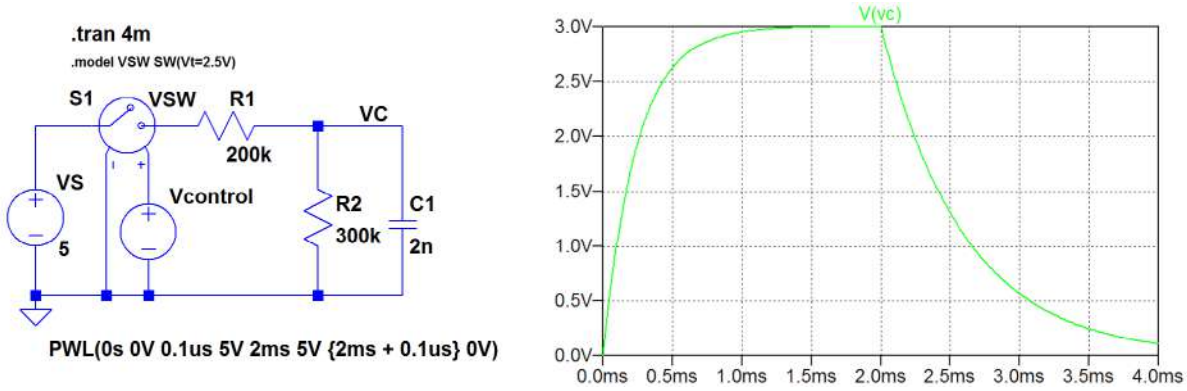


Figure 2.6: LTspice schematic and simulation results for the circuit in Fig. 2.5.

Figure 2.6 shows the LTspice schematic including a '.model' specification which is inserted using the command 'Edit → SPICE Directive' (or the toolbar symbol .op). The '.model' specification first specifies that the device to be modeled is the device using the model name 'VSW' in the schematic. Next, the model used is the standard LTspice model 'SW' which is specified by some parameters given in the brackets. In this case, only the threshold of the switch is changed from the default value of 0 V, so that we can use the signal specification from Fig. 2.3 for the control signal to the switch. Also note that the model is named 'VSW' to distinguish it from the default name.

The '.model' specification is shown on the schematic with a smaller font size than otherwise used in the schematic. The font size is selected when inserting the specification (or when editing the specification). You can also make a 'global' change of the font size on schematics using the command 'Tools → Control Panel' and the tab 'Drafting Options'.

In this schematic, the voltage V_S is specified as a dc voltage, and the time-varying signal to control the switch is the voltage source 'Vcontrol' which is specified as a piecewise-linear voltage source with the same specification as the input voltage for the circuit in Fig. 2.3. Also shown in Fig. 2.6 is the simulation result for v_C, compare Fig. 2.3.

It is clear from the simulation that the time constant for discharging the capacitor is now larger than the time constant for charging. The time constant for charging is the same as before, i.e., 240 µs. The time constant for discharging is now $\tau = R_2 C = 600$ µs. By scaling the voltage v_C in the same way as in Fig. 2.4, you can use the cursors in the plot window to find the time constants. Note that two cursors are available, so by placing one cursor at the start time for the discharge (2 ms) and the other so that the vertical difference between the two cursors is 1 V, you can estimate the time constant to be the horizontal difference between the two cursors. In order to make it easier to position one cursor on 2 ms, you may change the x-axis limits to show only the discharge, see Fig. 2.7.

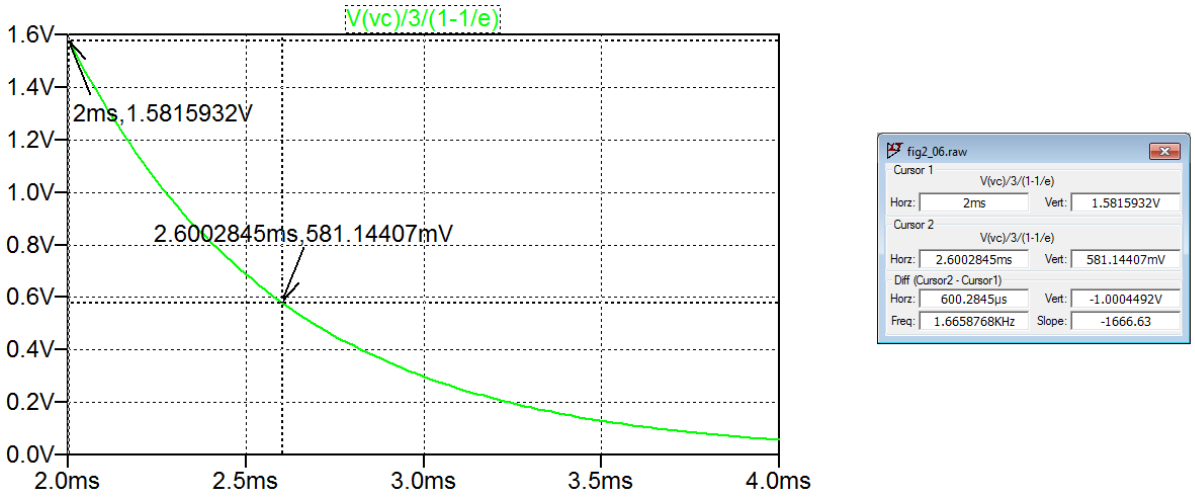


Figure 2.7: Plot window with scaled output for finding the time constant.

Ac response: The circuit shown in Fig. 2.1 is a first-order lowpass filter with the transfer function $V_c(j\omega)/V_s(j\omega) = G_0/(1+j(\omega/\omega_0))$ where $G_0 = R_2/(R_1+R_2) = 0.6$ (or -4.44 dB) is the low frequency gain and $\omega_0 = 1/\tau = 1/((R_1 \parallel R_2)C) = 4.17\times10^3$ s^{-1} (or 663 Hz) is the -3 dB frequency. This frequency response is normally shown in a Bode plot. In LTspice, the transfer function is simulated using the '.ac' simulation command. Use 'Simulate → Edit Simulation Cmd' and select the tab 'AC Analysis'. Here you can specify the type of sweep, start frequency, stop frequency and number of points. For a Bode plot of the frequency response, it would be reasonable to select the type of sweep to be 'Octave' or 'Decade' starting at 10 Hz and ending at 100 kHz. The number of points per octave or decade may be selected to 10. When you click 'OK', the simulation command can be placed on the schematic. If you still have the transient simulation command in your schematic, it is changed into a comment.

Also the voltage source V_s must be specified. Right-click on the symbol, and in the window with 'Advanced' settings, set the ac amplitude for ac small-signal analysis to 1. In this way, when plotting the output voltage V_c, the plot will directly show the transfer function. The dc bias point used for the '.ac' simulation is the bias point calculated with the value of V_S set to the initial value of the transient specification, in this example 0 V. If you need to specify a different dc bias value, you may set the time-varying function to '(none)' in order to open the specification box for a dc value. Sometimes it can be advantageous to split the voltage source v_S into two separate, series connected voltage sources so that you do not have to change or remove the time-varying signal specification but can set the desired bias value for the '.ac' simulation as the sum of the initial transient value and a series connected dc value.

Running the simulation opens a plot window with a horizontal frequency axis. When selecting '`V(vc)`' as the trace to show, both an amplitude plot and a phase plot appear as shown in Fig. 2.8. In this plot, the color of the curves has been changed from green to red to make the curves more visible and the vertical scales have been modified to the range 0 dB to -50 dB and 0° to -100°.

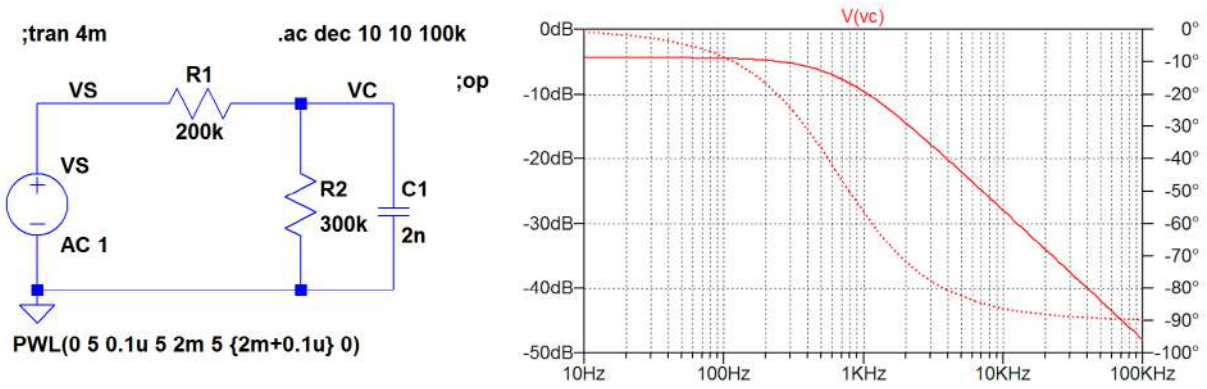


Figure 2.8: LTspice schematic and ac simulation results for the lowpass filter from Fig. 2.1.

The -3 dB frequency is found using the cursors in a way similar to what was done for finding time constants: You may place one cursor at a very low frequency and move the second cursor until it is 3 dB below the first cursor and then read the position of the second cursor. Alternatively, just move a cursor to the frequency where the phase is $-45°$. For a first order lowpass filter, this corresponds to the -3 dB frequency.

Be aware that the ac analysis is a small-signal analysis calculated from the bias point of the circuit. For the circuit shown here with only linear components, the bias point is not important, but for circuits with nonlinear components (e.g., MOS transistors), it is important to run the ac analysis from the correct bias point.

Example 2.2: A half-wave rectifier with a smoothing filter.

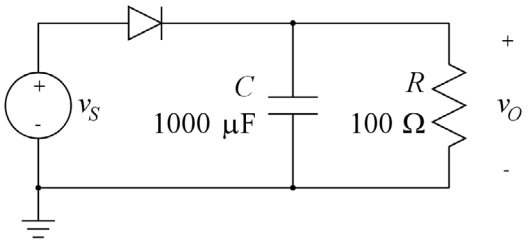


Figure 2.9: A half-wave rectifier with a smoothing filter.

The next example is a half-wave rectifier as shown in Fig. 2.9. When drawing the schematic, the diode symbol is selected from the component selection box ( ). Like the switch in Fig. 2.6, the diode is modeled by a '.model' specification. If you omit the '.model' specification, the diode defaults to the standard Shockley model $i_D = I_S\left[\exp\left(v_D/(nV_T)\right) - 1\right]$ (Hambley 2018) where the default value of the saturation current is $I_S = 10^{-14}$ A and the default value of the emission coefficient is $n = 1$. V_T is the thermal voltage (26 mV at room temperature).

Several models for different commercially available discrete type diodes are included with LTspice and are contained in a library file. When selecting the diode, you point to the centre of the diode symbol. This

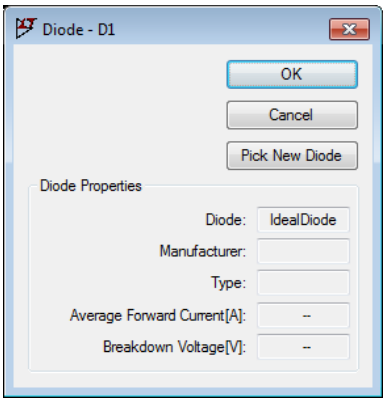


Figure 2.10: Window for selecting diode model.

turns the cursor into a hand ☜. A right-click on the mouse opens a window as shown in Fig. 2.10. By clicking 'Pick New Diode', you open a window with a selection of standard component diodes. Selecting a diode and clicking 'OK' will insert the diode name on the schematic and insert a link to the appropriate '.model' statement in the LTspice netlist file. If the selected diode is a Zener diode (or another type of diode), the symbol is also changed into the appropriate diode symbol.

Alternatively, you can specify your own diode model. You can find a description of the parameters for the diode model using LTspice 'Help'. In addition to the Shockley diode model, LTspice provides the option of using a piecewise-linear diode model in which you can specify the resistance in forward direction and in reverse direction and a forward threshold voltage to enter conduction. You may also specify the reverse breakdown region (particularly useful for a Zener-diode), see the 'Help' function in LTspice where you can also find the default values of the parameters. If you specify just one of the parameters for the piecewise-linear model, this model will be used rather than the Shockley model.

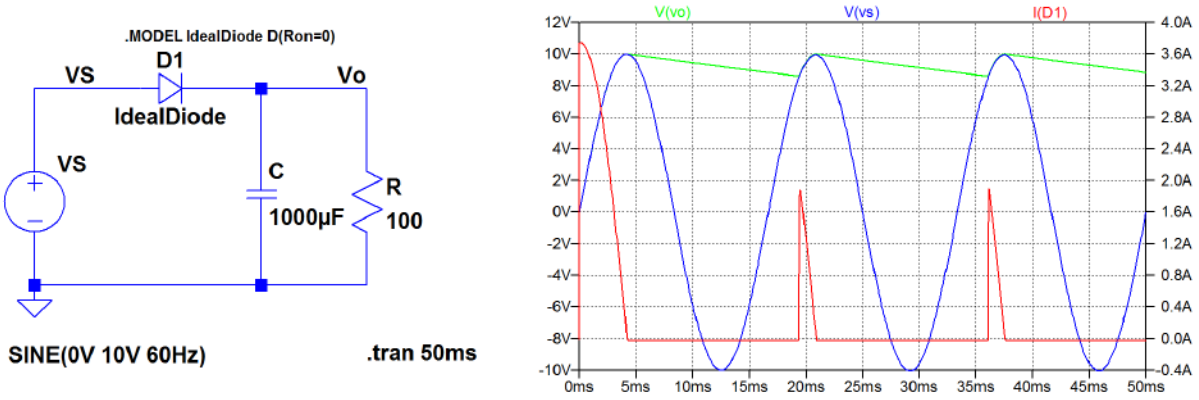


Figure 2.11: LTspice schematic and transient simulation results for the half-wave rectifier from Fig. 2.9.

Figure 2.11 shows the LTspice schematic and the simulation of the rectified voltage and the current through the diode, using an ideal diode model. Notice that the name of the diode model has been changed from the default 'D' to 'IdealDiode' by right-clicking on 'D' in the diode symbol and modifying it to

the new name of the diode model. Also note that in the specification of the sinusoidal input 'VS' only the dc offset, the amplitude and the frequency need to be specified. In the simulation plot window, using the cursors, you may find the ripple voltage V_r (in this case 1.38 V) and the peak current $i_{D\max}$ after the transient start-up phase (in this case 1.9 A). Notice that the start-up phase leads to a substantially higher value of i_D. As an exercise, you may compare these values to the approximated analytical expressions given by Sedra and Smith (2016, pp. 212-213): $V_r = V_p/(fCR)$ and $i_{D\max} = (V_p/R)(1+2\pi\sqrt{2fCR})$ where V_p and f are the amplitude and the frequency of the sinusoidal input voltage, respectively.

Also see what happens if the diode model is replaced by the Shockley default model.

Example 2.3: An amplifier with a capacitive feedback network.

The third example is a non-inverting amplifier with capacitive feedback, see Fig. 2.12. The amplifier is assumed to be an ideal voltage-controlled voltage source with infinite input resistance, zero output resistance and a gain of $A = 100$ V/V. A simple ac analysis results in

$$V_o(j\omega)/V_s(j\omega) = \frac{1+C_1/C_2}{1+(1+C_1/C_2)/A} \quad \text{for } \omega > 0. \tag{2.4}$$

Be aware that ω has to be positive. For $\omega = 0$, the impedance of the capacitors is infinite and the gain is not defined. Figure 2.13 shows the circuit drawn with LTspice and with a simulation command for an ac analysis from 10 kHz to 10 MHz. This simulation results in a simulated gain of 9.286 dB or 2.913 V/V as expected from Eq. (2.4). For the ac simulation, you cannot specify the start frequency to be 0. It has to be positive. Analytically, at dc (i.e., $\omega = 0$), the gain cannot be found and the input node to the inverting input of the amplifier is floating. Since at dc, no current can flow from this node, the capacitors C_1 and C_2 may be charged to some arbitrary dc voltages. For the ac simulation, LTspice must calculate a bias point, and for the circuit shown in Fig. 2.13, LTspice assumes a bias value of 0 V for v_F. After running the simulation, you may view the error log by using the command 'View → SPICE Error Log' (or 'Ctrl-L') and you will see that you get a warning: 'Node VF is floating'. In Fig. 2.13, the dc value of v_S has been specified to 0 V, and running an operating point simulation ('.op') on the circuit results in all voltages in the circuit being 0.

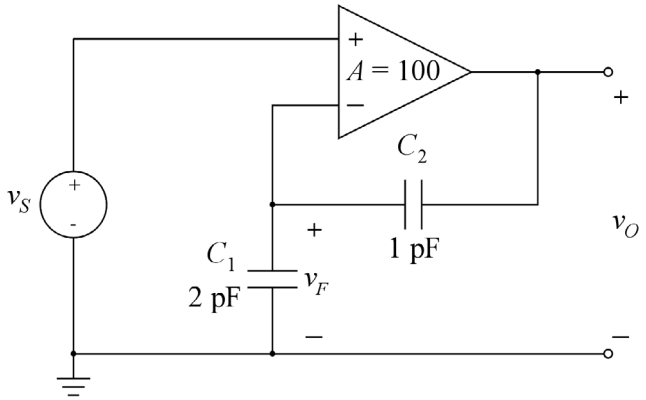


Figure 2.12: An amplifier with capacitive feedback.

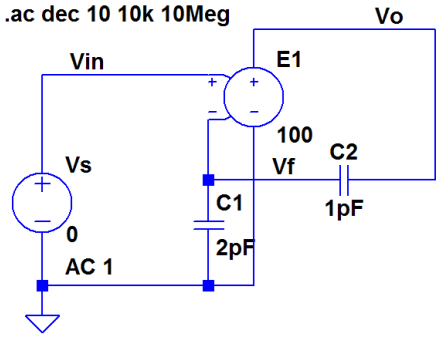
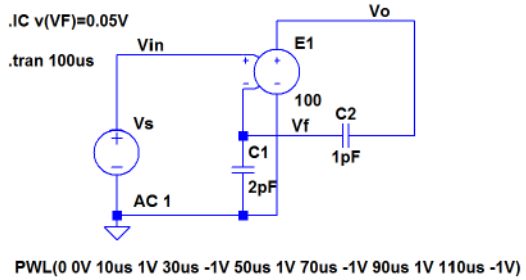


Figure 2.13: LTspice schematic for the circuit shown in Fig. 2.12.



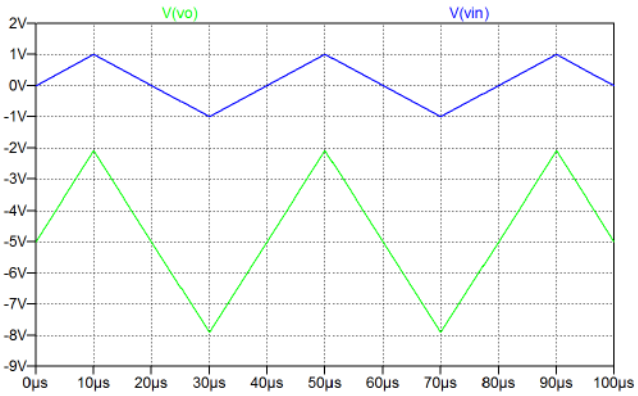


Figure 2.14: LTspice schematic for the circuit shown in Fig. 2.12 with specifications for an initial value of v_F and the simulation result for a transient simulation.

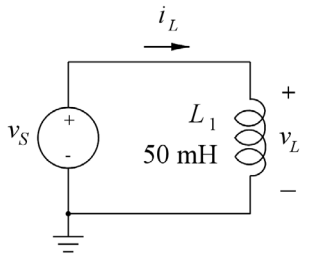
If we change the dc value of v_S to 1 V and run an operating point analysis, we find that the simulated value of the output voltage is now 100 V, i.e., the amplifier just provides the open loop gain A. This may not be a realistic situation for a practical circuit, and again an examination of the error log gives a warning: 'Node VF is floating'. One possible solution to the problem of a floating node is to establish a dc path to the node. For the circuit in Fig. 2.13, a very large resistor (many gigaohms) may be connected in parallel with one of the capacitors. If it is connected in parallel with C_1, the dc gain is the open loop gain of the amplifier, i.e., $A = 100$ V/V. If it is connected in parallel with C_2, the dc gain of the amplifier is $A/(1+A) = 0.9901$ V/V. Actually, you can specify a resistor in parallel with a capacitor by a right-click on the capacitor and using the window which opens for specifying an 'Equiv. Parallel Resistance[Ω]'.

There is an alternative way of handling a floating node: The initial value of the node voltage may be specified using a '.ic' SPICE directive. Assume for instance that the circuit of Fig. 2.12 has an initial value of the feedback voltage v_F of 0.05 V. With an input voltage of $v_S = 0$ V, this gives an initial value of the output voltage of $v_O = -A\,v_F = -5$ V. Assume also that the input voltage v_S is a periodic triangular voltage with a period of 40 µs, an amplitude of 1 V, a mean value of 0 V, and a start value of 0 V at time $t = 0$. Figure 2.14 shows the schematic with this specification of v_S, and the initial value of v_F is set by the SPICE directive (command 'Edit → SPICE Directive') '`.ic v(VF)=0.05V`'. Also shown

in Fig. 2.14 is the result of the transient simulation, showing the input voltage and the output voltage. It is evident that the gain of the circuit is about 3 V/V as expected and that the output voltage is offset by -5 V.

Example 2.4: An ideal inductor.

Consider an inductor L_1 connected directly to a voltage source v_S as shown in Fig. 2.15. The relation between current $i_L(t)$ and voltage $v_L(t)$ for the inductor is given by Eq. (2.5) shown in Fig. 2.15 where $i_L(t_0)$ is the current in the inductor at time $t = t_0$. For an ideal inductor with $v_S = 0$ (i.e., a short-circuited inductor), a constant current may flow in the inductor. When v_S changes value from an initial value of 0 V, the current in the inductor changes. As an example, assume that v_S is a square-wave signal with an amplitude of 1 V, a duty cycle of 50% and a period of 2 ms (corresponding to a frequency of 500 Hz). Also assume that the rise time and fall time of the square wave is 100 µs and that the mean value of $v_S(t)$ is 0. If the mean value of $v_S(t)$ is different from 0, the integral of $v_S(t)$ will be infinite for $t \to \infty$ which is clearly not acceptable.



$$i_L(t) = \frac{1}{L}\int_{t_0}^{t} v_L(t)dt + i_L(t_0) \tag{2.5}$$

Figure 2.15: An ideal inductor connected to a voltage source.

Even though the circuit is very simple, it does present some challenges to the simulation. First, you may notice that a dc solution only makes sense for $v_S = 0$ V. If v_S is a dc voltage with a value different from 0 V, the current in the inductor is infinite. For $v_S = 0$ V, the dc value of the current in L_1 is $i_L(t_0)$. Running a simple '.op' simulation, you will find that LTspice calculates a value of 0 for v_L and i_L when v_S is specified as a dc voltage with a value of 0 V. But if you change the dc value of v_S to some other value (e.g., 4 V), the '.op' simulation will still run but it will not give a meaningful result for i_L, and the error log does not give any warnings.

You may specify the value of $i_L(t_0)$ using the '.ic' SPICE directive. For the '.op' simulation, this turns the inductor into a dc current source with the specified value of $i_L(t_0)$, and the '.op' simulation gives the correct result for $i_L(t_0)$.

Next, we will consider a transient simulation with the square-wave voltage signal defined above. We assume that the square wave is defined for all values of t (starts at $-\infty$ and continues to $+\infty$). For the transient simulation, we may use the 'Pulse' function to specify the square wave. Figure 2.16 shows the LTspice schematic with a 'Pulse' specification for v_S. The syntax for the 'Pulse' specification is given in the dialogue box for specifying v_S. The number of cycles ('Ncycles') need not be specified. The pulses

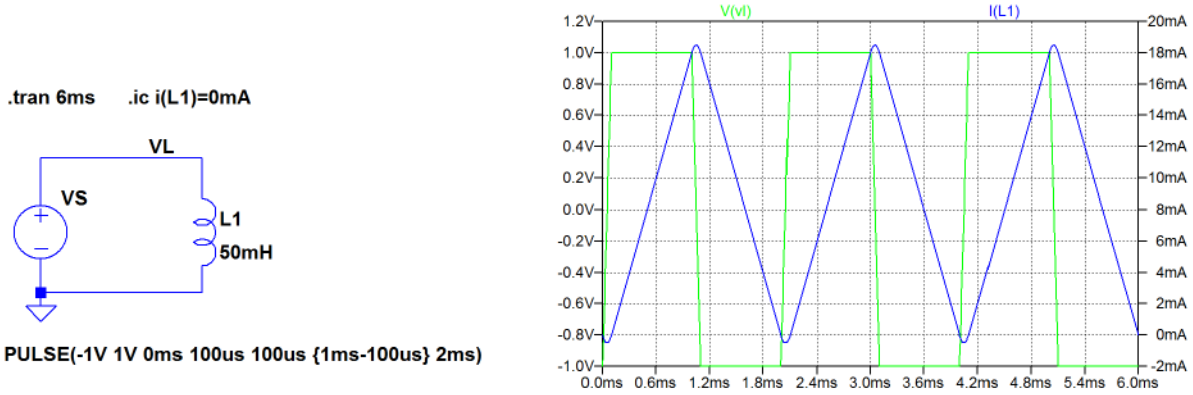


Figure 2.16: LTspice schematic for the circuit shown in Fig. 2.15 with specifications for an initial value of i_{L_1} and the simulation result for a transient simulation with a pulse input.

will continue for the total simulation time. For the transient simulation, LTspice starts by calculating the dc point for $t = 0$. With the 'Pulse' specification shown, the value of v_S for $t = 0$ is -1 V, so LTspice computes a wrong dc value for i_L. Hence, a '.ic' directive is necessary to specify the initial value of i_L, also when this initial value is 0. Also note in the 'Pulse' specification that the 'Ton' time is specified as '`{1ms-100us}`' in order to ensure that the average value of v_S is 0. Also shown in Fig. 2.16 is a plot of the simulation result. Both the voltage v_L and the current i_L is shown.

As an exercise you may run the same simulation with a different initial value of i_L. Also, see what happens if you forget the '.ic' directive.

Example 2.5: Revisiting the capacitor charging and discharging.

Example 2.1 showed how the charging and discharging of a capacitor via an *RC* network could be analyzed using voltage sources defined as time-varying voltages or using voltage-controlled switches. However, as we have learned in Example 2.3, an initial voltage can be defined for a capacitor for a transient analysis. This makes it possible to analyze charging and discharging without introducing the time-varying voltage sources or controlled switches. Thus, the charging of the capacitor in the circuit from Fig. 2.3 can be simulated with a dc value of 5 V for v_S and an initial value of 0 V for the capacitor voltage v_C as shown in Fig. 2.17.

Likewise, the discharge can be simulated with a dc value of 0 V for v_S and an initial value of 3 V for the capacitor voltage v_C, see Fig. 2.18.

Also the circuit from Fig. 2.6 can be simulated without the switch. The simulation of the charging of C is the same as shown in Fig. 2.17. For the simulation of the discharging, simply remove the voltage source v_S and specify an initial value of 3 V for the capacitor voltage v_C, see Fig. 2.19.

Of course, a similar approach can be used for analyzing the magnetization and demagnetization of an inductor using a specification of the initial value of the current in the inductor.

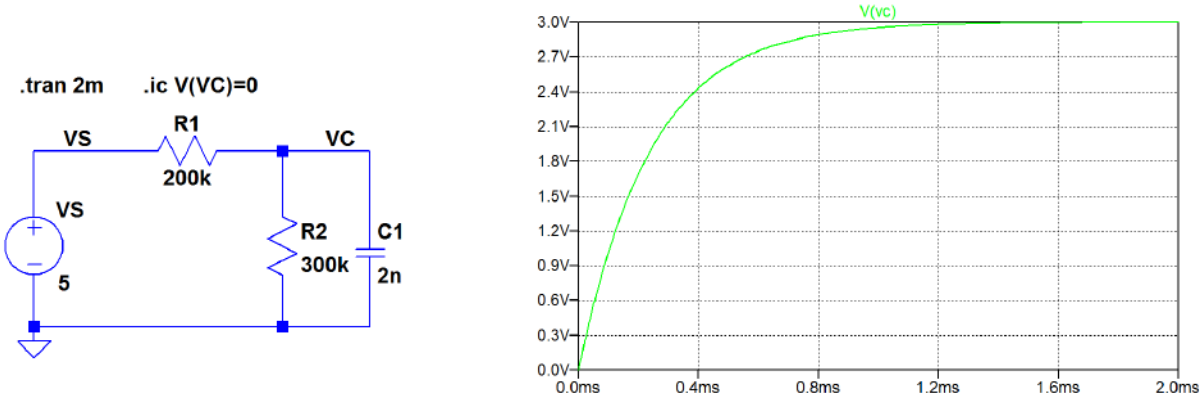


Figure 2.17: Simulation of capacitor charging using a dc voltage source and a specification of the initial capacitor voltage.

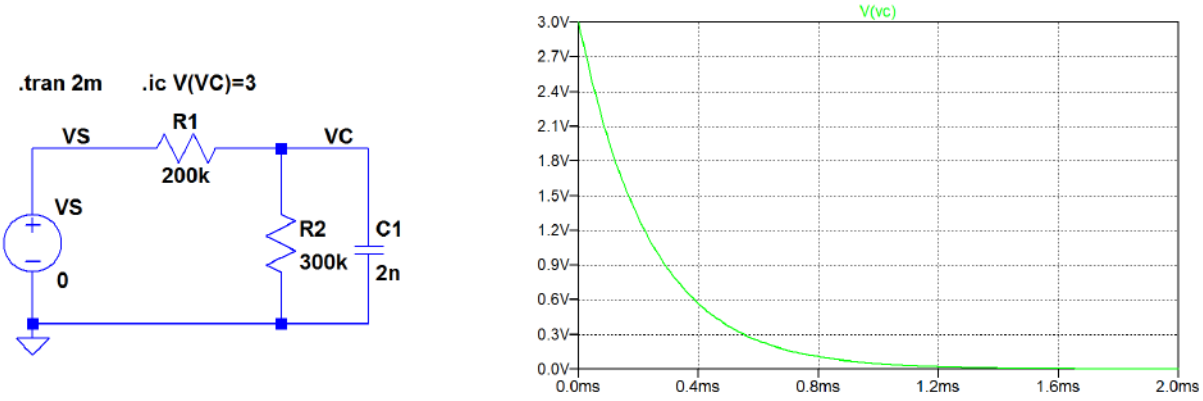


Figure 2.18: Simulation of capacitor discharging using a dc voltage source of 0 V and a specification of the initial capacitor voltage.

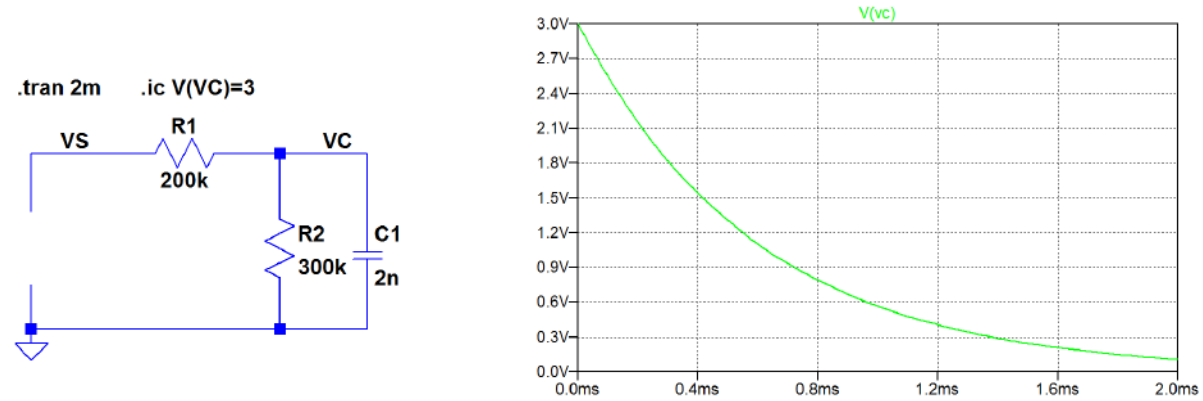


Figure 2.19: Simulation of capacitor discharging in the circuit from Fig. 2.6.

As an alternative to the simple inspection of the charge and discharge using the waveform plot and the cursor as shown in Fig. 2.4, you can use the SPICE directive '.meas' or '.measure'. With this directive, you can find the time for which the value of a variable reaches a specified value. For measuring the time constant for charging, you need the time for which v_C is equal to the final value multiplied by $(1 - 1/e)$.

For the circuit in Fig. 2.17, this is achieved by the SPICE directive:

'`.meas tau when v(VC)=3*(1-1/e)`'.

Here, '`tau`' is the time constant to be evaluated. By default, the time measurement starts at $t = 0$ and the time for stopping the time measurement is specified by the condition '`when v(VC)=3*(1-1/e)`'. After having run the simulation, the result of the '.meas' directive is found in the error log file ('Ctrl-L').

For the circuit in Fig. 2.17, the the error log file reports '`tau: v(vc)=3*(1-1/e) AT 0.000240194`'.

Likewise, for the discharge time constant in Figs. 2.18 and 2.19 you can use the directive:

'`.meas tau when v(vC)=3/e`'.

For Fig. 2.18, the error log file reports '`tau: v(vc)=3/e AT 0.000239805`' and for Fig. 2.19, the error log file reports '`tau: v(vc)=3/e AT 0.000599547`'.

The SPICE directive '.meas' provides a very versatile possibility for post-processing of the simulation results. For details concerning the syntax of '.meas', use the 'Help' function in LTspice or see Brocard (2013). Also, the 'Help me Edit' function available by a right-click in the dialogue box for entering a SPICE directive provides a guide to the syntax of the '.meas' directive.

The command is particularly useful when several identical post-processing operations are to be performed. Examples of this are demonstrated in Tutorials 3 and 6.

Example 2.6: Determining capacitances and resistances in *RC* networks.

In Tutorial 1, we saw how a simple '.op' simulation can be used to determine the resistance of a network with two terminals, for instance a Thévenin resistance, see Fig. 1.10. For a circuit with both capacitors and resistors, a '.op' simulation cannot be used to find the impedance. However, often the basic structure of an impedance is known. This may be the case for the input impedance or the output impedance of an amplifier, and using an 'a priori' knowledge about the topology of the impedance, a '.ac' simulation can be used to find the values of equivalent capacitances and resistances. In this example, we show how the values of resistors and capacitors can be simulated in a few simple topologies: a series connection of a resistor and a capacitor, a parallel connection of a resistor and a capacitor, and an *RC* network like shown in Example 2.1, and we also apply the simulation methods to an amplifier configuration with capacitive feedback.

***RC* series connection:** For the *RC* series connection shown in Fig. 2.20, the relation between the voltage $V(j\omega)$ and the current $I(j\omega)$ applied to the series connection is given by

$$V = I\left(R + \frac{1}{j\omega C}\right) \tag{2.6}$$

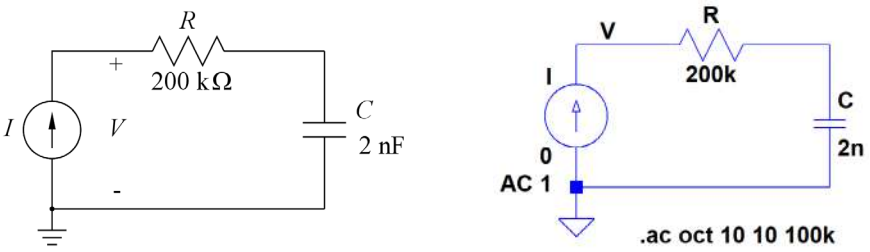


Figure 2.20: *RC* series network connected to a current source (a) and the corresponding LTspice schematic with a '.ac' simulation command (b).

Assuming that the current I is real, i.e., $\mathrm{Re}\,(I) = I$ and $\mathrm{Im}\,(I) = 0$, we find

$$\mathrm{Re}\,(V) + j\,\mathrm{Im}\,(V) \quad = \quad I\,(R + \frac{1}{j\omega C}) = I\,(R - \frac{j}{\omega C}) \tag{2.7}$$

$$\Rightarrow R = \frac{\mathrm{Re}(V)}{I} \quad \wedge \quad C = \frac{-I}{\omega\,\mathrm{Im}(V)} = \frac{-I}{2\pi f\,\mathrm{Im}(V)} \tag{2.8}$$

In LTspice, we can just define an ac value of 1 for the current source and plot '`Re(V(V))`' in order to find R and '`-1/(2*pi*frequency*Im(V(V)))`' in order to find C after having run a '.ac' simulation over a suitable frequency range. Notice that the value of π is simply defined as '`pi`' in the waveform arithmetic and the frequency f is defined as '`frequency`'. In Fig. 2.20, the component values are taken from Fig. 2.1, so a reasonable frequency range would be from 10 Hz to 100 kHz as for the '.ac' analysis shown in Fig. 2.8. By default, LTspice opens the plot windows with a logarithmic y-axis. This can be changed to a linear axis by moving the cursor over the y-axis and right-clicking. Figure 2.21 shows the plots corresponding to R and C, and using the cursors, you verify that the simulated values correspond to the actual values for the components. In Fig. 2.21, the plots of R and C have been shown in two different plot panes. This is achieved by using the command 'Plot Settings → Add Plot Pane'.

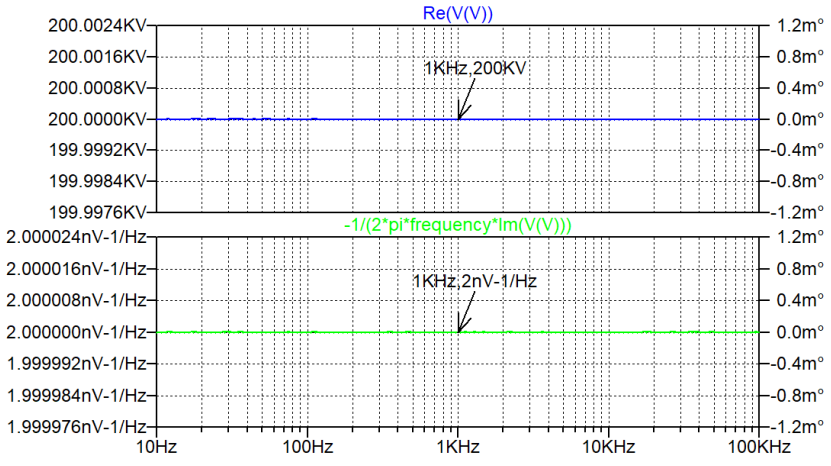


Figure 2.21: Plot of R (top) and C (bottom) as found from the simulation specified in Fig. 2.20.

Alternatively, we may apply a voltage source $V(j\omega)$ and find R and C from a simulation of the current in the RC series connection connection. Using Eq. (2.6) with $\mathrm{Re}(V) = V$ and $\mathrm{Im}(V) = 0$, we find

$$V = (\mathrm{Re}(I) + j\,\mathrm{Im}(I))(R + \frac{1}{j\omega C}) = R\,\mathrm{Re}(I) + \frac{\mathrm{Im}(I)}{\omega C} + j(R\,\mathrm{Im}(I) - \frac{\mathrm{Re}(I)}{\omega C}) \tag{2.9}$$

Considering the real part and the imaginary part separately, we obtain

$$\frac{V}{R} = \mathrm{Re}(I) + \frac{\mathrm{Im}(I)}{\omega CR} \tag{2.10}$$

$$\frac{1}{\omega CR} = \frac{\mathrm{Im}(I)}{\mathrm{Re}(I)} \tag{2.11}$$

Inserting Eq. (2.11) in Eq. (2.10), we find

$$\frac{V}{R} = \mathrm{Re}(I) + \frac{(\mathrm{Im}(I))^2}{\mathrm{Re}(I)} = \frac{(\mathrm{Re}(I))^2}{\mathrm{Re}(I)} + \frac{(\mathrm{Im}(I))^2}{\mathrm{Re}(I)} = \frac{|I|^2}{\mathrm{Re}(I)} \tag{2.12}$$

$$\Rightarrow R = \frac{V\,\mathrm{Re}(I)}{|I|^2} \tag{2.13}$$

Inserting Eq. (2.13) in Eq. (2.11), we find

$$C = \frac{|I|^2}{\omega V\,\mathrm{Im}(I)} \tag{2.14}$$

In LTspice, we define an ac value of 1 for the voltage and plot '`-Re(I(V))/Abs(I(V))**2`' in order to find R and '`Abs(I(V))**2/(2*pi*frequency*(-Im(I(V))))`' in order to find C. The minus signs are because '`I(V)`' is defined positive into the voltage source.

Figure 2.22 shows the plots corresponding to R and C (with linear y-axes) for the circuit from Fig. 2.20 with the current source replaced by a voltage source. Evidently, this simulation returns the same values of R and C as the simulation with a current source as the input.

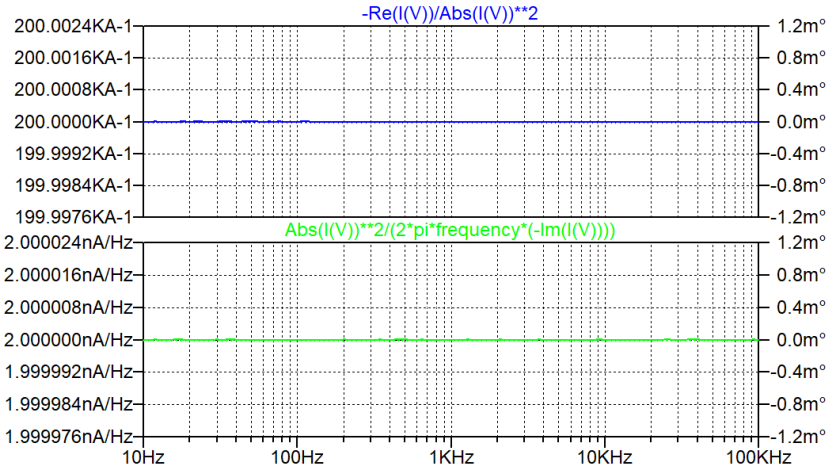


Figure 2.22: Plot of R (top) and C (bottom) for the circuit in Fig. 2.20 with the current source replaced by a voltage source.

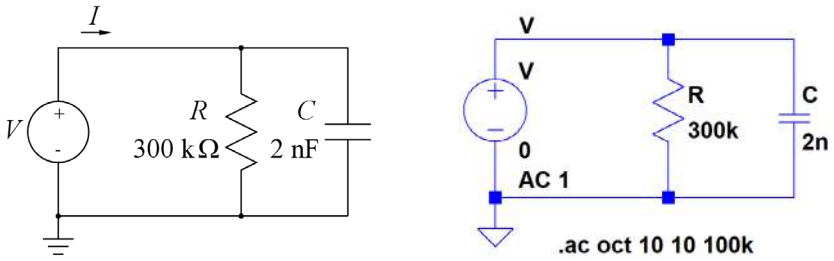
***RC* parallel connection:** For an *RC* parallel connection, we may find the current $I(j\omega)$ as a function of an applied voltage $V(j\omega)$ as shown in Fig. 2.23. Here, $I(j\omega)$ is given by

$$I = V\left(\frac{1}{R} + j\omega C\right) \tag{2.15}$$

Assuming that the voltage V is real, i.e., $\mathrm{Re}\,(V) = V$ and $\mathrm{Im}\,(V) = 0$, we find

$$\mathrm{Re}\,(I) + j\,\mathrm{Im}\,(I) = V\left(\frac{1}{R} + j\omega C\right) \tag{2.16}$$

$$\Rightarrow R = \frac{V}{\mathrm{Re}(I)} \quad \wedge \quad C = \frac{\mathrm{Im}(I)}{\omega V} = \frac{\mathrm{Im}(I)}{2\pi f V} \tag{2.17}$$



(a) (b)

Figure 2.23: *RC* parallel network connected to a voltage source (a) and the corresponding LTspice schematic with a '.ac' simulation command (b).

In LTspice, we can just define an ac value of 1 for the voltage and plot '`-1/Re(I(V))`' in order to find R and '`-Im(I(V))/(2*pi*frequency)`' in order to find C after having run a '.ac' simulation over a suitable frequency range. The minus signs are because '`I(V)`' is defined positive into the voltage source, and this is the opposite direction compared to Fig. 2.23(a). In Fig. 2.23, the component values are taken from Fig. 2.1, so a reasonable frequency range would be from 10 Hz to 100 kHz as for the '.ac' analysis shown in Fig. 2.8.

Figure 2.24 shows the plots corresponding to R and C (with linear y-axes), and using the cursors, you verify that the simulated values correspond to the actual values for the components. In Fig. 2.24, the plots of R and C have been shown in two different plot panes in the same way as for Fig. 2.21.

Alternatively, we may apply a current source $I(j\omega)$ and find R and C from a simulation of the voltage across the *RC* parallel connection. Using Eq. (2.15) with $\mathrm{Re}\,(I) = I$ and $\mathrm{Im}\,(I) = 0$, we find

$$I = (\mathrm{Re}\,(V) + j\,\mathrm{Im}\,(V))\left(\frac{1}{R} + j\omega C\right) = \frac{\mathrm{Re}\,(V)}{R} - \omega C\,\mathrm{Im}\,(V) + j\left(\frac{\mathrm{Im}\,(V)}{R} + \omega C\,\mathrm{Re}\,(V)\right) \tag{2.18}$$

Considering the real part and the imaginary part separately, we obtain

$$RI = \mathrm{Re}\,(V) - \omega CR\,\mathrm{Im}\,(V) \tag{2.19}$$

$$\omega CR = -\frac{\mathrm{Im}\,(V)}{\mathrm{Re}\,(V)} \tag{2.20}$$

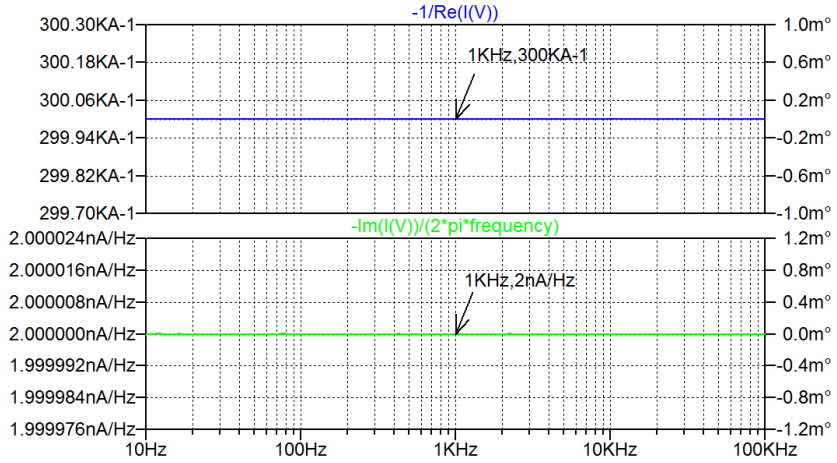


Figure 2.24: Plot of R (top) and C (bottom) as found from the simulation specified in Fig. 2.23.

Inserting Eq. (2.20) in Eq. (2.19), we find

$$RI = \mathrm{Re}\,(V) + \frac{(\mathrm{Im}\,(V))^2}{\mathrm{Re}\,(V)} = \frac{(\mathrm{Re}\,(V))^2}{\mathrm{Re}\,(V)} + \frac{(\mathrm{Im}\,(V))^2}{\mathrm{Re}\,(V)} = \frac{|V|^2}{\mathrm{Re}\,(V)} \tag{2.21}$$

$$\Rightarrow R = \frac{|V|^2}{I\,\mathrm{Re}\,(V)} \tag{2.22}$$

Inserting Eq. (2.22) in Eq. (2.20), we find

$$C = -\frac{I\,\mathrm{Im}\,(V)}{\omega\,|V|^2} \tag{2.23}$$

In LTspice, we define an ac value of 1 for the current and plot '`Abs(V(V))**2/Re(V(V))`' in order to find R and '`-Im(V(V))/(2*pi*frequency*Abs(V(V))**2)`' in order to find C.

Figure 2.25 shows the plots corresponding to R and C (with linear y-axes) for the circuit from Fig. 2.23 with the voltage source replaced by a current source. Evidently, this simulation returns the same values of R and C as the simulation with a voltage source as the input.

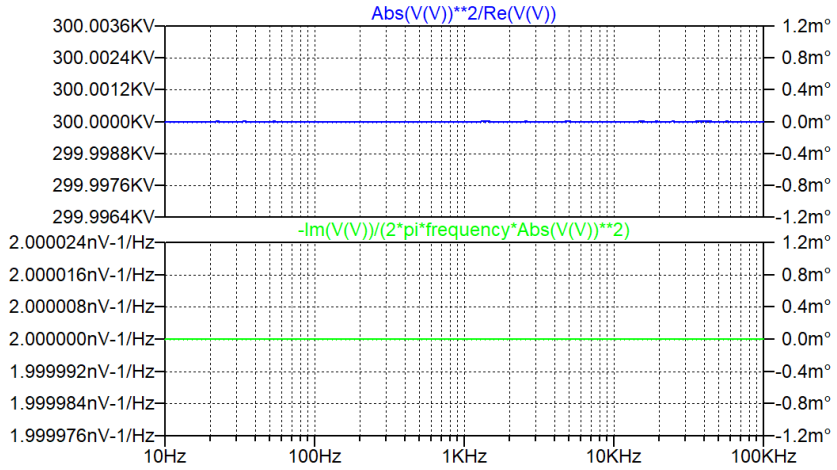
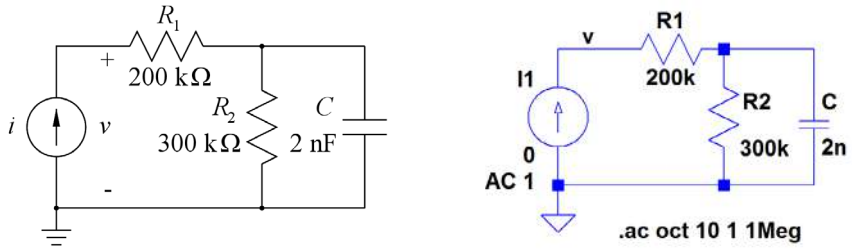


Figure 2.25: Plot of R (top) and C (bottom) for the circuit in Fig. 2.23 with the voltage source replaced by a current source.

The *RC* network from Figure 2.1: Also for an *RC* network as shown in Fig. 2.1 with two resistors and a capacitor, the values of the capacitor and the resistors can be found from a simulation of the relation between current and voltage at the input of the network. Figure 2.26 shows the circuit redrawn with a current source $I(j\omega)$ connected to the network.



(a) (b)

Figure 2.26: *RC* network from Fig. 2.1 connected to a current source (a) and the corresponding LTspice schematic with a '.ac' simulation command (b).

Again we assume that $I(j\omega)$ is real. For this circuit, we find

$$V(j\omega) = I\left(R_1 + \frac{R_2}{1+j\omega R_2 C}\right) = I\left(R_1 + \frac{R_2(1-j\omega R_2 C)}{1+(\omega R_2 C)^2}\right) \tag{2.24}$$

Equating the real parts and the imaginary parts, we find for $I = 1$

$$\mathrm{Re}\,(V) = R_1 + \frac{R_2}{1+(\omega R_2 C)^2} \tag{2.25}$$

$$\mathrm{Im}\,(V) = \frac{-\omega (R_2)^2 C}{1+(\omega R_2 C)^2} \tag{2.26}$$

From Eq. (2.25), we find that for $\omega \to \infty$, $\mathrm{Re}\,(V) \to R_1$ and for $\omega \to 0$, $\mathrm{Re}\,(V) \to R_1 + R_2$. This means that R_1 and $R_1 + R_2$ can be found from a plot of '`Re(V(V))`', provided the '.ac' simulation sweeps over a frequency range showing both very high frequencies and very low frequencies. This is not surprising: By inspecting the circuit, you see that at high frequencies where the capacitor approaches a short circuit, the impedance is R_1, and at low frequencies where the capacitor approaches an open circuit, the impedance is $R_1 + R_2$.

From Eq. (2.26), we find that for $\omega \to \infty$, $\mathrm{Im}\,(V) \to -1/(\omega C)$, and this implies that C can be found as $-1/(\mathrm{Im}\,(V)\omega)$. So from a plot of '`-1/(Im(V(V))*2*pi*frequency)`', C is found at a frequency high enough that the curve saturates.

Figure 2.27 shows these plots from which we verify that the simulated values correspond to the actual values for the components. Notice that the frequency sweep has been extended to start at 1 Hz in order to show the low frequency saturation of $\mathrm{Re}\,(V)$, corresponding to $R_1 + R_2$. In Fig. 2.27, the phase plots are not shown. They are disabled by moving the cursor to the right-hand y-axis, right-clicking and selecting 'Don't plot phase'.

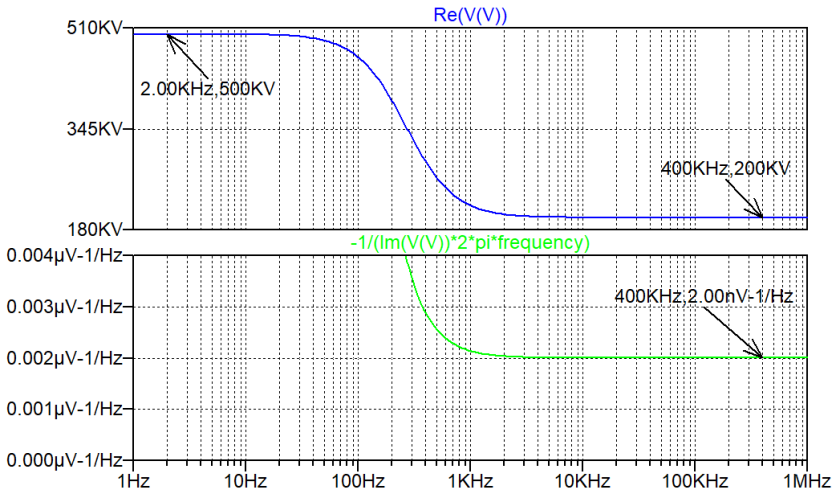


Figure 2.27: Plot of R_1 and $R_1 + R_2$ (top) and C (bottom) as found from the simulation specified in Fig. 2.26.

Input impedance of an inverting amplifier: For the previous circuits shown in this example, it was easy to recognize the values of resistors and capacitors because they were inserted directly in the LTspice schematics. This is not always the case. Consider the inverting amplifier shown in Fig. 2.28.

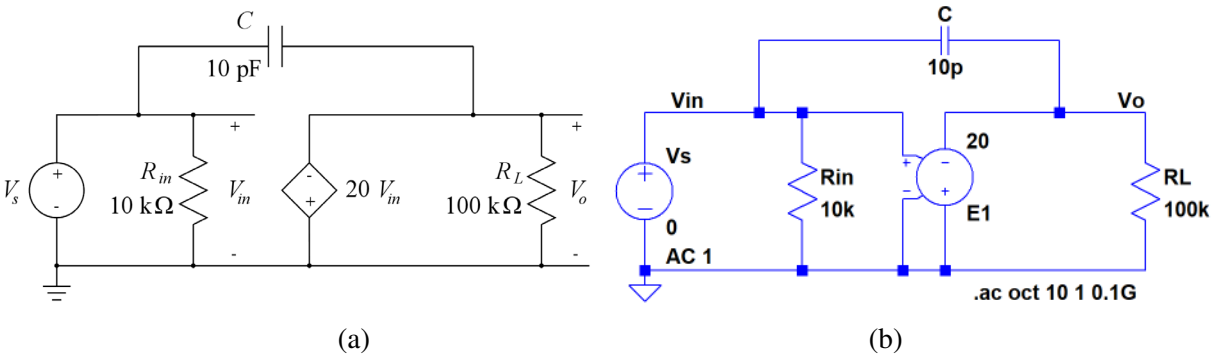


Figure 2.28: An inverting amplifier with capacitive feedback (a) and the corresponding LTspice schematic with a '.ac' simulation command (b).

This amplifier has a feedback path with a capacitor C so we must expect the input impedance to have both a capacitive element from the feedback capacitor C and a resistive element from the input resistor R_{in}. Let us assume that the capacitive element appears as an input capacitance in parallel with R_{in}. In this case, the input impedance can be found using the approach from Figs. 2.23 and 2.24. So after running the '.ac' simulation specified in the LTspice schematic in Fig. 2.28, we should plot '`-1/Re(I(Vs))`' in order to find the input resistance and '`-Im(I(Vs))/(2*pi*frequency)`' in order to find the input capacitance.

Figure 2.29 shows these plots. From the plots, we find an input resistance of 10 kΩ as expected, and we see that the input capacitance is 210 pF. You may recognize this value of the input capacitance as

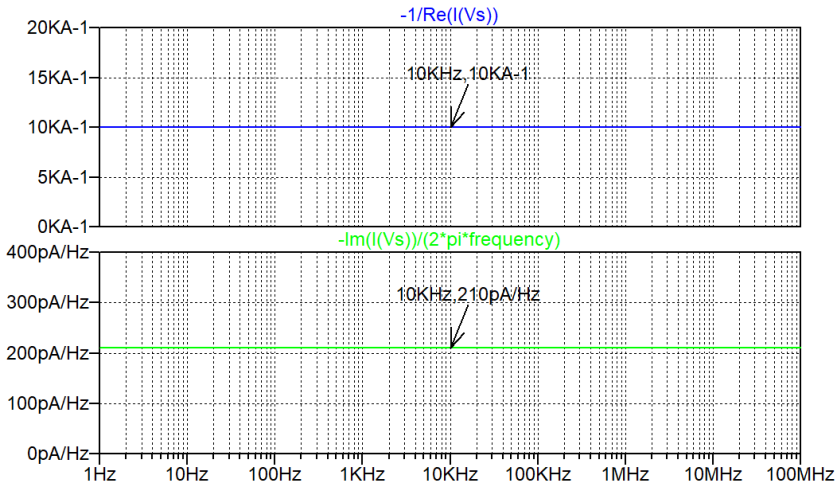


Figure 2.29: Plot of input resistance (top) and input capacitance (bottom) as found from the simulation specified in Fig. 2.28.

$(1+|A|)C$ where $A = -20$ V/V is the gain of the inverting amplifier. The large input capacitance is caused by the Miller effect on the capacitor C (Sedra & Smith 2016).

Example 2.7: A switch-mode dc to dc converter.

Switch-mode converters are frequently used for converting a high dc voltage to a low dc voltage which can be used as the supply voltage for a CMOS integrated circuit. As an example, a dc voltage of 10 V may need to be converted to a dc voltage of 1.8 V for a circuit designed in a 0.18 µm CMOS process.

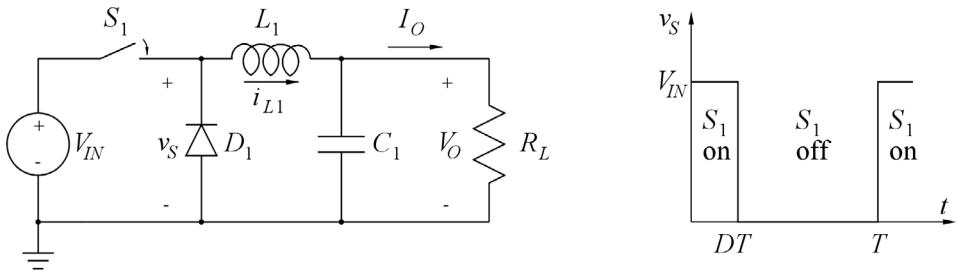


Figure 2.30: A switch-mode dc to dc converter

Figure 2.30 shows a buck converter for converting a dc input voltage V_{IN} to a lower dc output voltage V_O (Erickson & Maksimovic 2001). The converter uses a switch S_1 to switch the inductor current $i_{L1}(t)$ either from the input voltage source V_{IN} or from the diode D_1. With an ideal diode D_1, the voltage $v_S(t)$ switches between V_{IN} and 0 as shown in Fig. 2.30, and provided the converter operates in continuous conduction mode (CCM), i.e., the inductor current $i_{L1}(t) > 0$ at all times, the output voltage is the average of the voltage $v_S(t)$. With a switching period T and a duty cycle D for the control voltage to the switch, this results in an output voltage $V_O = DV_{IN}$.

For the converter, we assume an input voltage of 10 V, an output voltage of 1.8 V and an output current of 100 mA. This corresponds to a load resistor of $R_L = 18\ \Omega$. Also, we assume that the diode is ideal and that the switching frequency is 1 MHz, corresponding to $T = 1\ \mu$s. Thus the on-time for the switch is $T_{\text{on}} = 0.18\ \mu$s. Assuming a nearly constant V_O, we find that during the on-time for S_1, the current $i_{L1}(t)$ increases linearly with time, see Eq. (2.5), resulting in a change $\Delta i_{L1} = DT(V_{IN} - V_O)/L_1$. The average of this increase is $\Delta i_{L1}/2$, so for the converter to run in continuous conduction mode (CCM), we must require

$$I_O > \Delta i_{L1}/2 = DT(V_{IN} - V_O)/(2L_1) \Rightarrow L_1 > DT(V_{IN} - V_O)/(2I_O) = 7.38\ \mu\text{H} \tag{2.27}$$

For the simulation, we select $L_1 = 10\ \mu$H. The waveform for $i_{L1}(t)$ is shown in Fig. 2.31

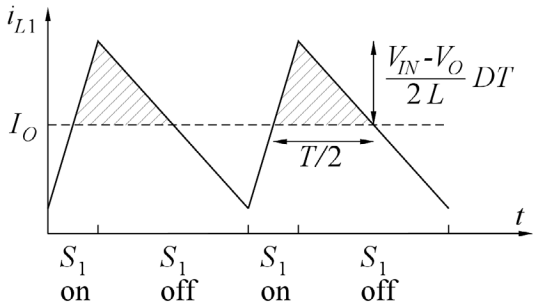


Figure 2.31: The inductor current in the switch-mode dc to dc converter.

The capacitor C_1 is used to reduce the ripple at the output voltage. From Kirchhoff's current law, we find the current flowing in C_1 as $i_{C1} = i_{L1} - I_O$, so when i_{L1} is larger than I_O, the capacitor is charged and when i_{L1} is smaller than I_O, the capacitor is discharged. In Fig. 2.31, showing two cycles of i_{L1}, the hatched areas where $i_{L1} > I_O$ illustrate the charging of C_1. The charge Q_{ripple} transferred to C_1 during a charging cycle is the integral of the current to C_1 during the charging, i.e., the area of the hatched triangle, so

$$Q_{\text{ripple}} = \frac{1}{2}\frac{V_{IN} - V_O}{2L_1}DT\frac{T}{2} \tag{2.28}$$

From $Q_{\text{ripple}} = C_1 V_{O\text{ripple}}$, we then find the ripple (peak-to-peak) to be

$$V_{O\text{ripple}} = \frac{1}{8}\frac{V_{IN} - V_O}{L_1 C_1}DT^2 = \frac{1}{8}V_{IN}(1-D)D\frac{T^2}{L_1 C_1} \tag{2.29}$$

Assuming that we want a ripple of less than 10 mV_{pp}, this implies

$$C_1 > \frac{1}{8}V_{IN}(1-D)D\frac{T^2}{L_1 V_{O\text{ripple}}} = 1.845\ \mu\text{F} \tag{2.30}$$

For the simulation, we select $C_1 = 2\ \mu$F.

Figure 2.32 shows an LTspice schematic corresponding to Fig. 2.30. For the switch, we just use the switch model '`SW`' with the default parameters. In order to avoid warnings and error messages about a missing model for the switch, we include the directive '`.model SW SW`'. The switch is controlled by the

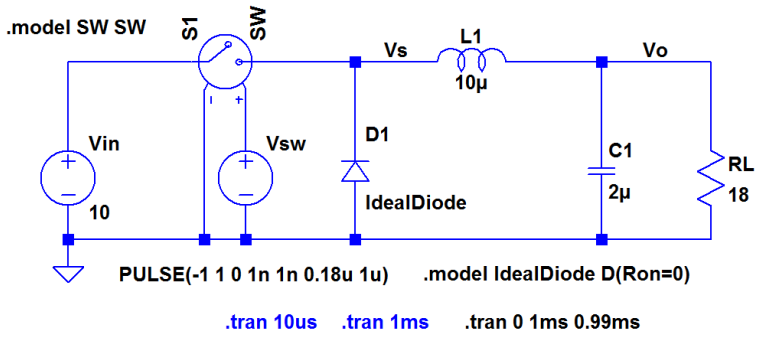


Figure 2.32: LTspice schematic of the switch-mode dc to dc converter.

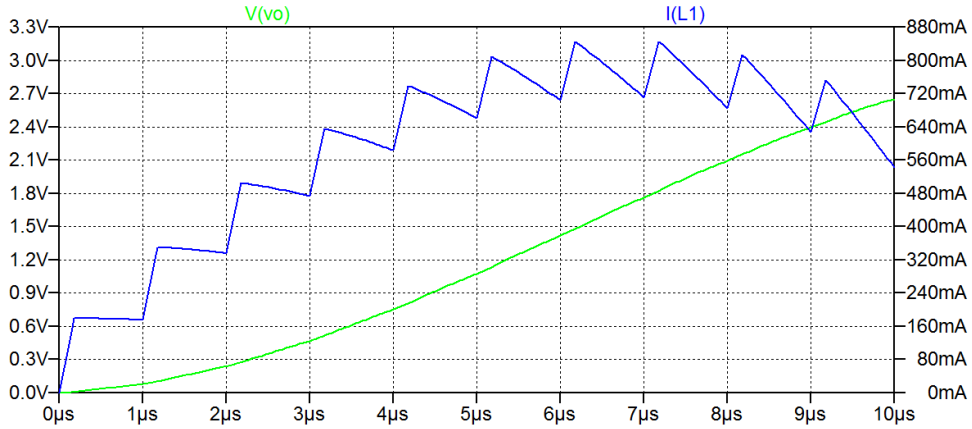


Figure 2.33: Plot of output voltage and inductor current for the dc to dc converter for the first 10 µs.

voltage source '`Vsw`' which has been defined as a pulse. You may observe that when the inductor was inserted, it has been rotated three times such that the positive direction of current in the inductor is from left to right, corresponding to the definition given in Fig. 2.30.

The diode has been specified by a '.model' directive just like the ideal diode in Example 2.2.

Now let us try to simulate the converter. We run a transient simulation, and for the first attempt, we just specify a simulation time of 10 µs, corresponding to 10 cycles of the switching voltage. The resulting plot of V_O and i_{L1} is shown in Fig. 2.33. We see that the output voltage does not settle at 1.8 V, and we also see that the inductor current is very different from the current shown in Fig. 2.31. Apparently, there is an initial transient response which is much longer than 10 µs, so we change the simulation time to 1 ms. The resulting output is shown in Fig. 2.34.

Apparently, the initial transient response is quite long compared to the switch period T, several hundreds of µs. However, from the plot shown in Fig. 2.34, we cannot see the details of the response when the converter has reached its steady state. We may show this by zooming in on the x-axis in the plot (right-click on the axis and define suitable axis limits) but if we are not interested in the initial transient response, an alternative method is to define the '`Time to start saving data`' when specifying the

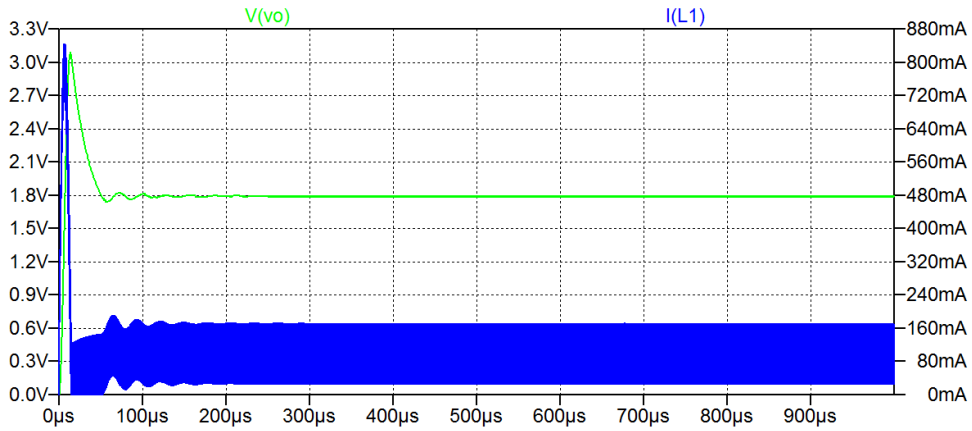


Figure 2.34: Plot of output voltage and inductor current for the dc to dc converter for 1 ms.

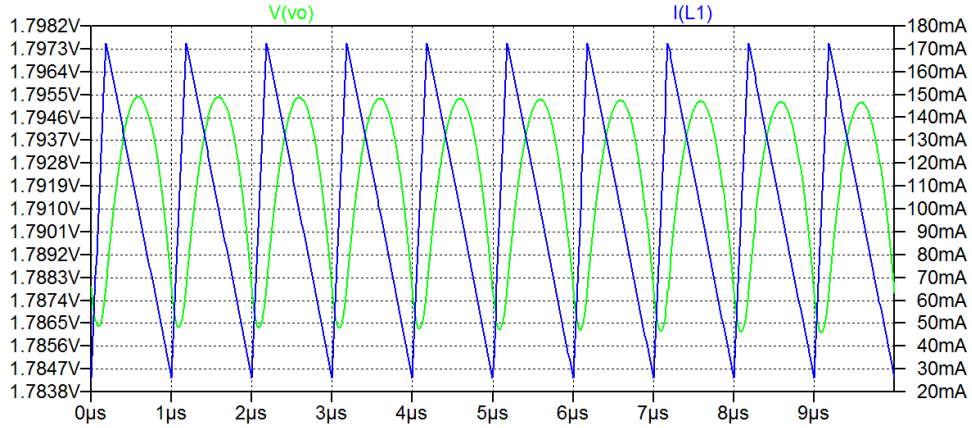


Figure 2.35: Plot of output voltage and inductor current for the dc to dc converter for the last 10 µs of a 1 ms simulation.

transient simulation. Specifying this time to be 0.99 ms results in the directive '`.tran 0 1ms 0.99ms`' shown in Fig. 2.32 and the plot shown in Fig. 2.35.

We note that the minimum value of i_{L1} is about 27.5 mA, so the converter is operating in continuous conduction mode. We also see that the output voltage varies between 1.786 V and 1.795 V, so the ripple is less than 10 mV and the mean value is close to the design target of 1.8 V.

We may find the average value of V_O by using 'Ctrl-left-click' on the label above the plot. This opens a window where we find an average value of 1.7917 V for V_O. Likewise, a 'Ctrl-left-click' on '`I(L1)`' shows an average inductor current of 99.494 mA. These values of voltage and current are slightly different from the ideal values of 1.8 V and 100 mA. The differences are caused by a non-zero value of the on resistance of the switch and by the finite rise time and fall time of the voltage controlling the switch.

Of course, the use of a nearly ideal switch and an ideal diode is a gross simplification of the circuit. As an exercise, you may see what happens if the diode is replaced by the Shockley default diode model, see Problem 2.9.

Hints and pitfalls

- Capacitors may give rise to floating nodes for dc voltages.
- The voltage of a floating node may be controlled by connecting a very large resistor from a non-floating node to the floating node.
- Alternatively, the initial value of a floating node may be specified using a '.ic' SPICE directive.
- Likewise, the initial value of the current in an inductor can be specified by a '.ic' SPICE directive.
- You cannot independently specify a dc voltage and a time-varying signal for a voltage source or current source.
- It can be a good idea to insert a signal source (voltage or current) as a combination of a dc bias source and a time-varying signal source so that the dc bias value and the time-varying waveform can be specified separately.
- In many textbooks, including Hambley (2018) and Sedra & Smith (2016), dc values, time-varying signals and small-signal values are distinguished by appropriate combinations of uppercase and lowercase letters and subscripts. LTspice is case insensitive. Thus, in LTspice, you cannot use uppercase and lowercase letters and subscripts to distinguish between dc values, time-varying signals and small-signal values.
- The SPICE Error Log ('Ctrl-L') provides warnings about floating nodes.
- When specifying component values in the schematic, 'e' (or 'E') is used for specifying the suffix, e.g., 'e6' for 'Mega'.
- When specifying mathematical expressions in the waveform viewer, 'e' (or 'E') is the base of the natural logarithm.
- Text (comments) can be placed in a schematic using the command 'Edit → Text', toolbar symbol  or hotkey 'T'.
- A SPICE directive can be changed into a comment by a 'Ctrl-right-click' on the directive. This opens a window where you can specify if the text should be interpreted as a directive or as a comment.
- Text and other annotations (e.g., cursor position) can be placed in a simulation plot using the command 'Plot Settings → Notes & Annotations'.
- In a simulation plot, additional plot panes can be opened using the command 'Plot Settings → Add Plot Pane'.
- Curly brackets `{ }` can be used to indicate expressions to be calculated by LTspice when specifying values, for instance, '`{2ms + 0.2us}`' is equivalent to '`2.0002ms`'.
- The SPICE directive '.measure' is very useful for calculating design parameters from simulations.
- The results of a '.measure' SPICE directive are found in the error log file ('Ctrl-L').

References

Brocard, G. 2013, *The LTspice IV Simulator – Manual, Methods and Applications*, First Edition, Swiridoff Verlag, Künzelsau, Germany.

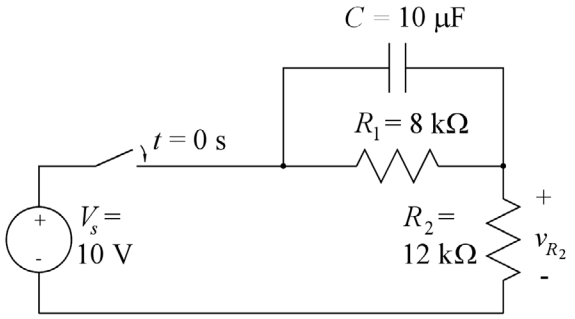
Erickson, RW. & Maksimovic, D. 2001: *Fundamentals of Power Electronics*, Second Edition, Kluwer Academic Publishers, New York, USA.

Hambley, AR. 2018, *Electrical Engineering, Principles and Applications*, Seventh Edition, Pearson Education Ltd., Harlow, UK.

Sedra, AS. & Smith, KC. 2016, *Microelectronic Circuits*, International Seventh Edition, Oxford University Press, New York, USA.

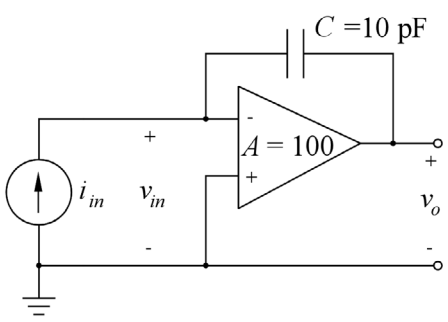
Problems

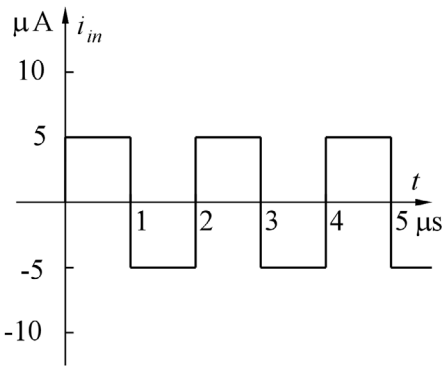
Problem 2.1



For the circuit shown above, assume that the switch is closed at time $t = 0$ and re-opened at time $t =$ 100 ms. Find the value of the voltage v_{R_2} immediately after the switch is closed. Find the value of v_{R_2} immediately before the switch is re-opened. Find the value of v_{R_2} immediately after the switch is re-opened. Plot v_{R_2} versus time for $0 \leq t \leq 200$ ms. Plot the capacitor voltage versus time and find the time constants for the charging and discharging of the capacitor C.

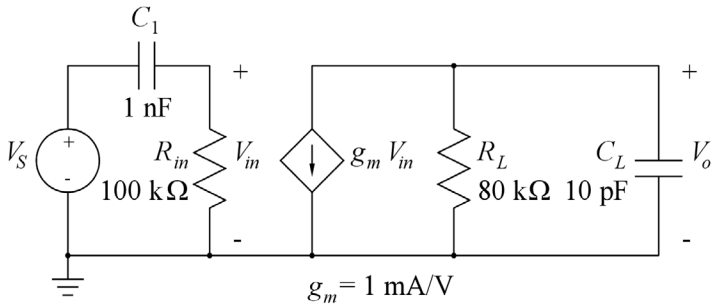
Problem 2.2





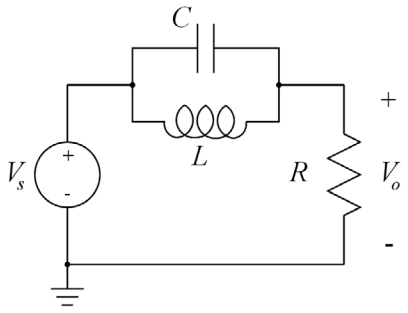
For the circuit shown above, plot the output voltage v_o versus time t for $0 \leq t \leq 5$ μs. You may assume that the amplifier has infinite input resistance and zero output resistance. Also, assume that the initial value of the input and output voltage at $t = 0$ is 0 V and that the input current pulses have a rise time and fall time of 1 ns. Which initial value of the input voltage v_{in} will result in a mean value of 0 V for the output voltage v_o?

Problem 2.3



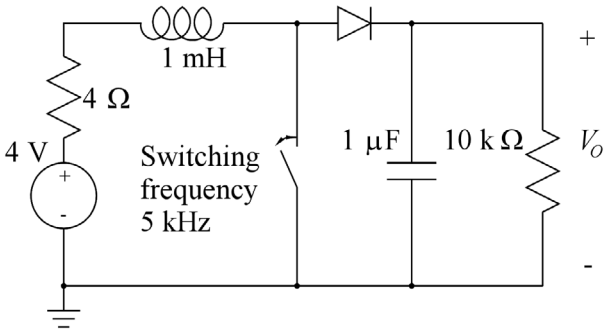
For the circuit shown above, find the midband gain, the upper and lower half-power (−3 dB) frequencies and the 3-dB bandwidth. Plot the output voltage V_o versus frequency in a Bode plot covering a frequency range which extends from approximately one decade below the lower half-power frequency to approximately one decade above the upper half-power frequency.

Problem 2.4



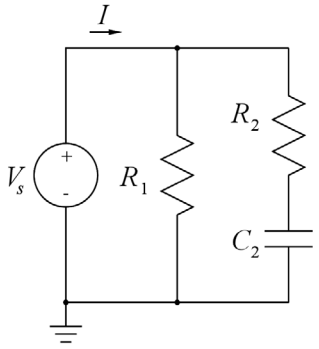
For the notch filter shown above, plot V_o versus frequency in a frequency range showing the notch and the 3-dB bandwidth. Assume $L = 1$ µH, $C = 5$ pF and $R = 10$ kΩ. From the plot, find the notch frequency, the bandwidth and the quality factor Q.

Problem 2.5



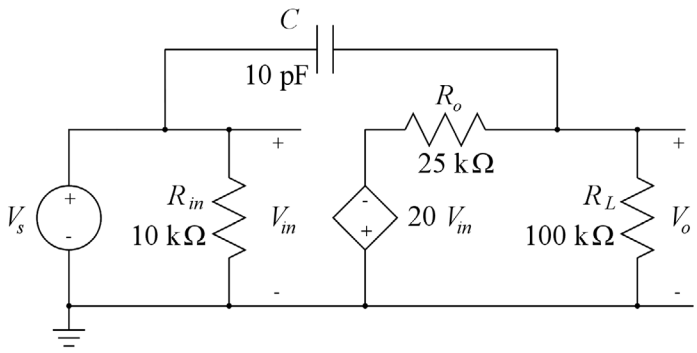
The circuit shown above is a dc-dc converter which converts a dc voltage of 4 V into a high voltage V_O. The switch is an electronic switch which opens and closes with a frequency of 5 kHz and a duty cycle of 50%, starting at time $t = 0$. The diode can be assumed to be modeled by the default Shockley diode model. Initially, the current in the inductor is 0 and the output voltage V_O is 0. Find the dc output voltage for $t \to \infty$ and find the time required for V_O to reach 90% of the final value.

Problem 2.6



For the circuit shown above, assume that the voltage $V_s(j\omega)$ is real and has an ac amplitude of 1. Derive expressions for finding R_1, R_2 and C_2 from a '.ac' simulation over a suitable range of the frequency f, using asymptotic values for $f \to 0$ and $f \to \infty$. Verify your results by simulating the circuit with $R_1 = 300$ kΩ, $R_2 = 200$ kΩ and $C_2 = 2$ nF.

Problem 2.7



The figure above shows an inverting amplifier similar to the amplifier from Fig. 2.28 but with a finite output resistance $R_o = 25$ kΩ for the controlled voltage source. Assuming that the input impedance has a topology as shown in Problem 2.6, find the values of R_1, R_2 and C_2 in the input impedance.

Problem 2.8

For the amplifier shown in Problem 2.7, use a '.ac' simulation to find the output impedance by resetting the input voltage and replacing the load resistor R_L with an ac current source. Assume that the output impedance is a parallel connection of a resistor and a capacitor.

Problem 2.9

Re-simulate the circuit from Fig. 2.32 using the default Shockley diode model instead of the ideal diode model. Find the resulting dc output voltage. Modify the duty cycle of the switching signal '`Vsw`' so that a dc output voltage of 1.8 V is obtained. Find the ripple on the output voltage.

Answers

2.1: $v_{R_2}(t=0) = 10\,\text{V}$; $v_{R_2}(t=100\,\text{ms}^-) = 6.5\,\text{V}$; $v_{R_2}(t=100\,\text{ms}^+) = 0\,\text{V}$; $\tau^+ = 48\,\text{ms}$; $\tau^- = 80\,\text{ms}$.

2.2: $v_{in} = -2.474\,\text{mV}$.

2.3: Midband gain: 38 dB. Lower half-power frequency: 1.58 kHz.
Upper half-power frequency: 201.7 kHz. 3-dB bandwidth: 200.1 kHz.

2.4: Notch frequency: 71.18 MHz, bandwidth: 3.19 MHz, quality factor $Q = 22.31$.

2.5: $V_O(t \to \infty) = 51.4\,\text{V}$. Rise time: 7.5 ms.

2.6: $R_1 = V_s/\text{Re}\,(i) =$ `-1/Re(I(Vs))` for $f \to 0$.
$R_{\text{eq}} = R_1 \parallel R_2 = V_s/\text{Re}\,(i) =$ `-1/Re(I(Vs))` for $f \to \infty$. From this, $R_2 = R_1 R_{\text{eq}}/(R_1 - R_{\text{eq}})$.
$C_2 = \text{Im}\,(i)/(\omega V_s) =$ `-Im(I(Vs))/(2*pi*frequency)` for $f \to 0$.

2.7: $R_1 = 10\,\text{k}\Omega$; $R_2 = 1.177\,\text{k}\Omega$; $C = 170\,\text{pF}$.

2.8: $R_o = 25\,\text{k}\Omega$; $C_o = 10\,\text{pF}$.

2.9: Dc output voltage $V_O = 1.31\,\text{V}$; Duty cycle $D = 0.24$; $V_{O\,\text{ripple}} = 13.4\,\text{mV}_{\text{pp}}$.

Tutorial 3 – MOS Transistors

This tutorial introduces the fundamentals of MOS transistor modeling in LTspice. After having completed the tutorial, you should be able to

- specify MOS transistors using the basic Shichman-Hodges transistor model.
- use advanced transistor models obtained from textbooks or process foundries.
- simulate transistor input characteristics and output characteristics.
- find transistor small-signal parameters from a dc operating point ('.op') analysis.
- estimate basic transistor parameters from a dc operating point ('.op') analysis.
- simulate transistor small-signal parameters using a dc sweep analysis ('.dc') simulation.
- simulate transistor small-signal parameters using a dc transfer ('.tf') simulation.
- simulate transistor small-signal capacitances using an ac ('.ac') simulation.

Example 3.1: Different MOS transistor symbols and models in LTspice.

In LTspice, several symbols are available for a MOS transistor. They are all inserted using the command 'Edit → Component' (or toolbar symbol or hotkey 'F2') which opens the component selection box. Here you find the components 'nmos', 'nmos4', 'pmos' and 'pmos4' which are shown in Fig. 3.1. The MOS transistor is a device requiring a '.model' statement for the specification.

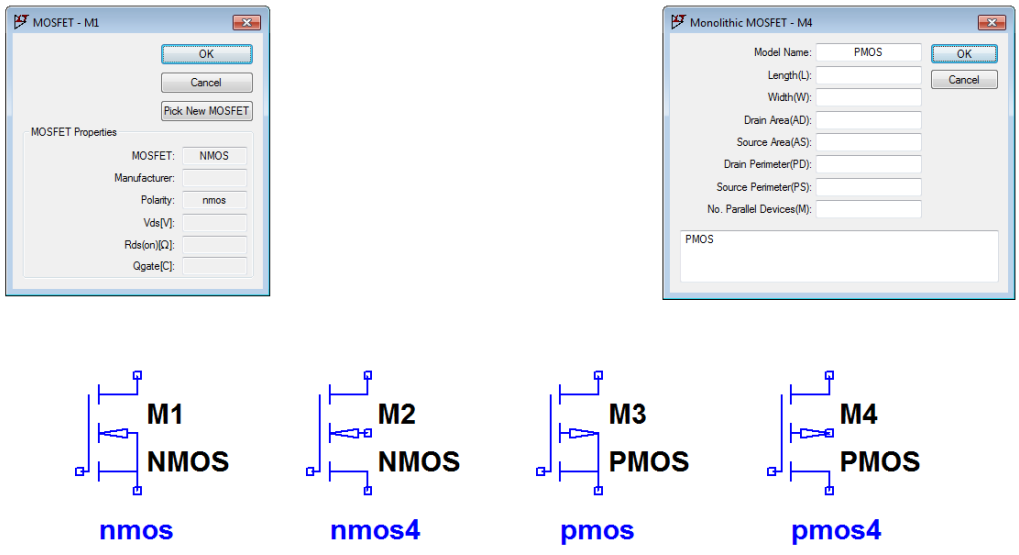


Figure 3.1: MOS transistor symbols and specification windows in LTspice.

Models for different discrete type MOS transistors are included with LTspice and are contained in a library file. Discrete type MOS transistors normally have their source and bulk contact connected, and for these components, you use the symbols 'nmos' and 'pmos' with only three terminals. When specifying the component, you point to the centre of the transistor symbol. This turns the cursor into a hand ☜. A right-click opens a window as shown in the top left part of Fig. 3.1. By clicking 'Pick New MOSFET', you open a window with a selection of standard component MOS transistors. Selecting a transistor and clicking 'OK' will insert the transistor name on the schematic and insert a link to the appropriate '.model' statement in the LTspice netlist file.

Models for MOS transistors in integrated circuits are not included with LTspice. For MOS transistors in integrated circuit design there is flexibility with respect to the bulk connection, so here you should use the symbols 'nmos4' and 'pmos4' with four terminals. When right-clicking on these symbols, a specification window as shown in the top right part of Fig. 3.1 opens. Notice that the specification window explicitly defines a 'Monolithic MOSFET'. The specification window contains entries for a model name and for the layout parameters of the MOSFET. LTspice has possibilities for specifying MOS models of different complexity, including the most basic Shichman-Hodges model (Shichman & Hodges 1968) and several advanced models such as the BSIM models (Sheu et al. 1987) and the EKV model (Enz & Vittoz 2006). The 'Help' function in LTspice provides an overview of the models. A shortcut (hotkey) to the 'Help' function is 'F1' which opens the 'LTspiceHelp'. Look for 'M. MOSFET' in the index and click 'Display' to find the description of MOS transistors. Notice the netlist syntax for a MOS transistor:
'`Mxx drain-node gate-node source-node bulk-node model-name layout-parameters`'.
This syntax is generated from the specifications entered in the schematic using the specification window for a 'Monolithic MOSFET'. A '.model' statement must also be included with the parameters for the specific MOS transistor model to be used for the simulation.

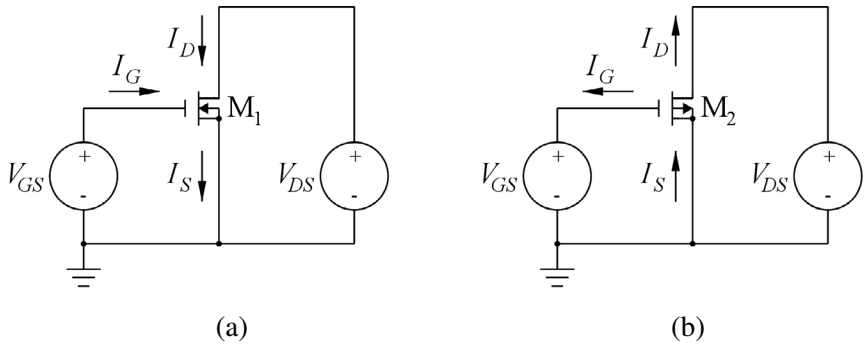


Figure 3.2: Normal textbook definitions of sign conventions for transistor currents and voltages. (a) NMOS transistor. (b) PMOS transistor.

We will start by considering the very simple circuits shown in Fig. 3.2. This is just to illustrate the sign conventions for voltages and currents and to illustrate how the '.model' statements can be included.

Standard textbook conventions: Most textbooks use the sign conventions for the transistor currents shown in Fig. 3.2. This assures that all currents are positive (or zero) in the normal operating regions of the transistors for both n-channel transistors and p-channel transistors. Using the Shichman-Hodges model for an n-channel transistor, we find

$$I_D = 0;\ V_{GS} \le V_t \tag{3.1}$$

$$I_D = \mu_n C_{ox}\left(\frac{W}{L}\right)[(V_{GS}-V_t)V_{DS} - V_{DS}^2/2](1+\lambda V_{DS});\ 0 \le V_{DS} \le V_{GS}-V_t \tag{3.2}$$

$$I_D = \frac{\mu_n C_{ox}}{2}\left(\frac{W}{L}\right)(V_{GS}-V_t)^2(1+\lambda V_{DS});\ 0 \le V_{GS}-V_t \le V_{DS} \tag{3.3}$$

where μ_n is the electron mobility, C_{ox} is the gate oxide capacitance per unit area, W and L are the channel width and length, respectively, V_{GS} is the gate-source voltage, V_{DS} is the drain-source voltage, V_t is the threshold voltage and λ is the channel-length modulation parameter. The threshold voltage V_t depends on the source-bulk voltage V_{SB} and is found from

$$V_t = V_{to} + \gamma(\sqrt{V_{SB}+|2\Phi_F|} - \sqrt{|2\Phi_F|}) \tag{3.4}$$

where V_{to} is the threshold voltage with $V_{SB}=0$, γ is the bulk threshold parameter (or body effect constant) and $|\Phi_F|$ is the Fermi potential of the body.

For this simple model, the process parameters specifying the transistor model are $\mu_n C_{ox}$, V_{to}, λ, γ and $|2\Phi_F|$. The lay-out parameters required for the transistor model are channel length L and width W.

The three regions of operation are termed cut-off ($V_{GS} \le V_t$), triode region, also called the linear region ($0 \le V_{DS} \le V_{GS}-V_t$) and active region ($0 \le V_{GS}-V_t \le V_{DS}$), and for the n-channel transistor (NMOS transistor), both V_{GS}, V_{DS} and V_t (assuming an enhancement NMOS transistor) are positive in the triode region and in the active region. The active region is also called the saturation region, and the minimum value of V_{DS} for which the transistor is in the saturation region is called $V_{DS\mathrm{sat}}$.

For a PMOS transistor, the voltages V_{GS}, V_{DS} and V_t and the three regions are defined as follows:

Cut-off region: $V_t \le V_{GS}$ (or $|V_{GS}| \le |V_t|$). (3.5)

Triode region: $V_{GS}-V_t \le V_{DS} \le 0$ (or $0 \le |V_{DS}| \le |V_{GS}-V_t|$). (3.6)

Active region: $V_{DS} \le V_{GS}-V_t \le 0$ (or $0 \le |V_{GS}-V_t| \le |V_{DS}|$). (3.7)

With the sign conventions for current given in Fig. 3.2, the equations given above for the currents can be used for both NMOS transistors and PMOS transistors provided the absolute values of V_{GS}, V_{DS} and V_t are used.

LTspice conventions: Figure 3.3 shows the transistors from Fig. 3.2 redrawn in LTspice with the symbols 'nmos4' and 'pmos4'. Also shown in the figure are model statements specifying the process parameters for each of the transistors. The model used is the simple Shichman-Hodges model and only the parameters corresponding to $\mu_n C_{ox}$, V_{to}, λ, γ and $|2\Phi_F|$ are specified. In LTspice, they are named '`Kp`',

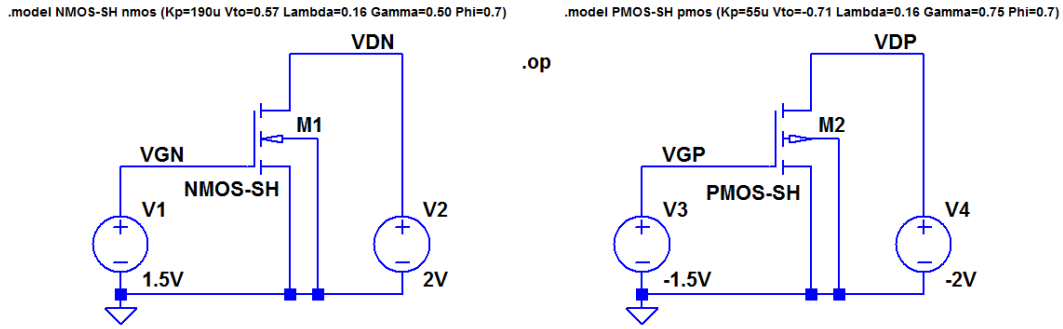


Figure 3.3: LTspice schematic for the circuits from Fig. 3.2.

'Vto', 'Lambda', 'Gamma' and 'Phi', respectively. These names are used for both NMOS transistors and PMOS transistors as shown in Fig. 3.3. The models are named NMOS-SH and PMOS-SH, respectively, and the parameters are representative for a 0.35 µm CMOS process (Chan Carusone, Johns & Martin 2012). The value of λ has been calculated for $L = 1$ µm. For the Shichman-Hodges model, λ is assumed to be inversely proportional to the channel length L. A dc operating point analysis is specified by the '.op' directive. The transistor geometries are specified using the specification window shown in Fig. 3.1, and for both transistors, L is 1 µm and W is 10 µm. This specification is not by default visible on the schematic. In Tutorial 4, Example 4.1, we show how to make it visible on the schematic. Here, to see the specification, we use the command 'View → SPICE Netlist' which brings up the window shown in the left part of Fig. 3.4. Notice that in addition to the circuit specification, the netlist includes references to the standard LTspice MOS models and the library file for standard MOS transistors. This file ('standard.mos') has been placed in a default folder during the installation of LTspice.

SPICE Netlist

```
* M:\LTspicebook_3rd_Edition\Tutorial03\Fig3_03.asc
M1 VDN VGN 0 0 NMOS-SH l=1um w=10um
V1 VGN 0 1.5V
V2 VDN 0 2V
V3 VGP 0 -1.5V
M2 VDP VGP 0 0 PMOS-SH l=1u w=10u
V4 VDP 0 -2V
.model NMOS NMOS
.model PMOS PMOS
.lib C:\Users\erbr\Documents\LTspiceXVII\lib\cmp\standard.mos
.model NMOS-SH nmos (Kp=190u Vto=0.57 Lambda=0.16 Gamma=0.50 Phi=0.7)
.op
.model PMOS-SH pmos (Kp=55u Vto=-0.71 Lambda=0.16 Gamma=0.75 Phi=0.7)
.backanno
.end
```

Output file

```
--- Operating Point ---

V(vdn):          2              voltage
V(vgn):          1.5            voltage
V(vgp):          -1.5           voltage
V(vdp):          -2             voltage
Id(M2):          -0.000226548   device_current
Ig(M2):          -0             device_current
Ib(M2):          2.01e-012      device_current
Is(M2):          0.000226548    device_current
Id(M1):          0.00108458     device_current
Ig(M1):          0              device_current
Ib(M1):          -2.01e-012     device_current
Is(M1):          -0.00108458    device_current
I(V4):           0.000226548    device_current
I(V3):           0              device_current
I(V2):           -0.00108458    device_current
I(V1):           0              device_current
```

Figure 3.4: Netlist file and output file for the circuit from Fig. 3.3.

Running the '.op' simulation produces the output file shown in right part of Fig. 3.4. From this, you see that the drain current is positive for the NMOS transistor and negative for the PMOS transistor. Also, the source current is negative for the NMOS transistor and positive for the PMOS transistor. The reason for this is that LTspice uses the convention that the positive direction of current flow is *into* the transistor, regardless of transistor type and transistor terminal.

Small-signal transistor parameters: Very important in the design of analog CMOS circuits are the small-signal properties of the transistors. At low frequencies, a small-signal transistor model can be derived from the nonlinear large-signal model by differentiation. For the Shichman-Hodges model for the NMOS transistor in the active region, see Eqs. (3.3) and (3.4), we find the following small-signal parameters:

$$g_m = \frac{\partial i_D}{\partial v_{GS}} = \mu_n C_{ox}\left(\frac{W}{L}\right)(V_{GS}-V_t)(1+\lambda V_{DS}) = \frac{2I_D}{V_{GS}-V_t}$$
$$= \sqrt{2\mu_n C_{ox}\left(\frac{W}{L}\right) I_D(1+\lambda V_{DS})} \simeq \sqrt{2\mu_n C_{ox}\left(\frac{W}{L}\right) I_D} \tag{3.8}$$

$$g_{ds} = 1/r_{ds} = \frac{\partial i_D}{\partial v_{DS}} = \lambda\frac{\mu_n C_{ox}}{2}\left(\frac{W}{L}\right)(V_{GS}-V_t)^2 = \frac{\lambda I_D}{1+\lambda V_{DS}} \simeq \lambda I_D \tag{3.9}$$

$$g_{mb} = \frac{\partial i_D}{\partial v_{BS}} = \frac{\partial i_D}{\partial V_t}\frac{\partial V_t}{\partial v_{BS}} = \frac{\gamma g_m}{2\sqrt{V_{SB}+|2\Phi_F|}} = \chi g_m \tag{3.10}$$

Corresponding to Eqs. (3.8) - (3.10), we have the small-signal model shown in Fig. 3.5. This model also applies to PMOS transistors, and g_m, g_{ds} and g_{mb} are positive for both NMOS transistors and PMOS transistors.

The small-signal parameters are always calculated assuming a specific bias point (operating point) for the transistor. In Spice, a calculation of the small-signal parameters is carried out with an operating point analysis, the '.op' simulation. LTspice does not show the small-signal parameters in the output file. However, the 'SPICE Error Log' provides the small-signal parameters. So using the command 'View → SPICE Error Log' or the hotkey 'Ctrl-L', you can open the error log with the small-signal parameters. Doing so for the circuit from Fig. 3.3 results in the error log shown in Fig. 3.6. Notice that the error log gives both the values of bias voltages and currents for the transistors and the values of the small-signal parameters. Also note that the error log provides warnings that the transistor dimensions are smaller than what is recommended for the Spice transistor models used for the simulation. This is an indication that the simple Shichman-Hodges model is a rather inaccurate transistor model for sub-micron transistor technologies. The main reason for using it here is its simplicity and also the fact that it is a transistor model often used for initial manual analysis of transistor circuits.

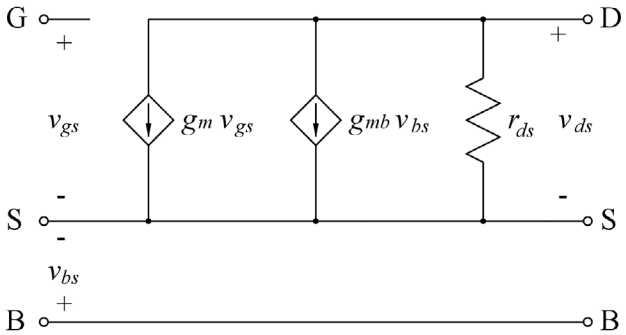


Figure 3.5: Low-frequency small-signal MOS transistor model.

SPICE Error Log

```
Circuit: * M:\LTspicebook_3rd_Edition\Tutorial03\Fig3_03.asc

Instance "m2": Length shorter than recommended for a level 1 MOSFET.
Instance "m1": Length shorter than recommended for a level 1 MOSFET.
Direct Newton iteration for .op point succeeded.
Semiconductor Device Operating Points:
                          --- MOSFET Transistors ---
Name:          m2          m1
Model:       pmos-sh     nmos-sh
Id:        -2.27e-04    1.08e-03
Vgs:       -1.50e+00    1.50e+00
Vds:       -2.00e+00    2.00e+00
Vbs:        0.00e+00    0.00e+00
Vth:       -7.10e-01    5.70e-01
Vdsat:     -7.90e-01    9.30e-01
Gm:         5.74e-04    2.33e-03
Gds:        2.75e-05    1.31e-04
Gmb:        2.57e-04    6.97e-04
Cbd:        0.00e+00    0.00e+00
Cbs:        0.00e+00    0.00e+00
Cgsov:      0.00e+00    0.00e+00
Cgdov:      0.00e+00    0.00e+00
Cgbov:      0.00e+00    0.00e+00
Cgs:        0.00e+00    0.00e+00
Cgd:        0.00e+00    0.00e+00
Cgb:        0.00e+00    0.00e+00
```

Figure 3.6: SPICE Error Log with bias point values and small-signal parameters from the simulation of the circuit from Fig. 3.3.

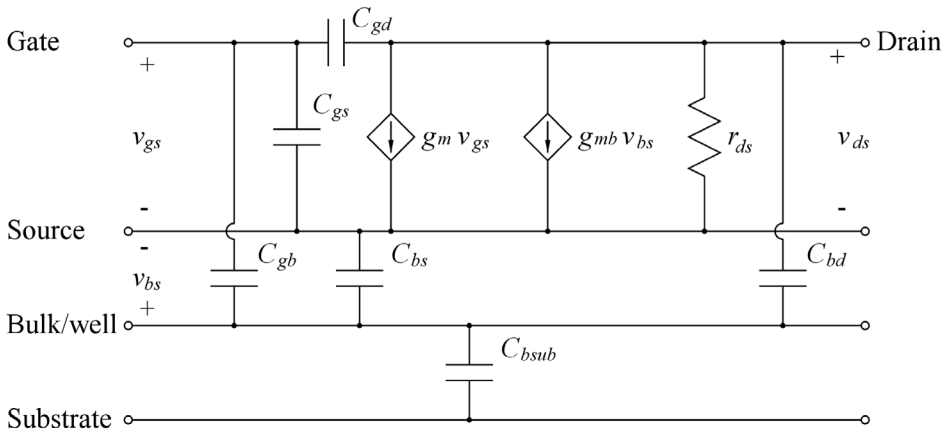


Figure 3.7: High-frequency small-signal MOS transistor model.

At high frequencies, the small-signal model must be augmented with the internal capacitors of the transistor as shown in Fig. 3.7. The size of the capacitors is calculated from the transistor dimensions, including the channel length, the channel width and the dimensions of source diffusion and drain diffusion. Also gate overlap is considered. You will see that in Fig. 3.6, all the capacitors have a value of 0. In order to make it possible for LTspice to calculate the capacitances, the transistor models must include parameters describing junction capacitances and oxide capacitance per unit area and also overlap capacitances per unit length. The parameters are defined as shown in the 'LTspiceHelp', and Fig. 3.8 shows Fig. 3.3 redrawn with '.model' specifications for the capacitances. The '.model' specifications extend over more than one line with a '+' to indicate that a line is a continuation of the specification. The specifications are adapted from Chan Carusone, Johns & Martin (2012, Chapter 1.5).

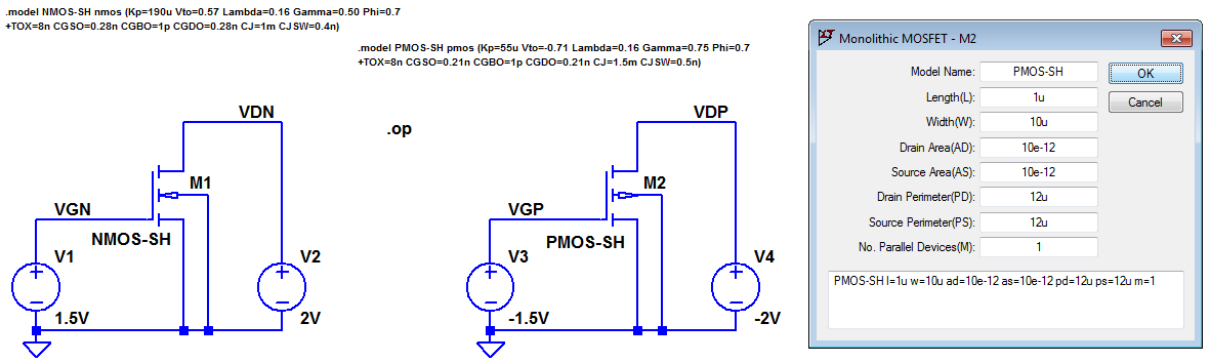


Figure 3.8: LTspice schematic including model parameters for capacitances for the circuits from Fig. 3.2.

Also, the transistor specification must include areas and perimeters of source and drain diffusions. This is done using the specification window shown in Fig. 3.8. The areas of source and drain diffusion will typically have a minimum dimension of approximately W times 2.75 times the minimum length which is 0.35 µm for the process assumed for Fig. 3.8. The perimeter of the drain and source diffusion will typically have a minimum dimension of W plus 5.5 times the minimum length (Sedra & Smith 2016, Appendix B). The multiplier M in the specification window is used to specify multiple devices in parallel.

The capacitance from well to substrate (C_{bsub} in Fig. 3.7) is not part of the transistor model in LTspice since a well may be common to several transistors, so C_{bsub} must be inserted separately if it is needed in the circuit analysis.

The netlist corresponding to Fig. 3.8 and the error log with the operating point information and small-signal parameters are shown in Fig. 3.9. Comparing to Fig. 3.6, it is seen that g_m, g_{ds} and g_{mb} remain unchanged, but now the small-signal capacitances are computed. Note that each of these capacitances may have an overlap component (e.g., '`Cgsov`') and a junction capacitance or gate oxide capacitance component (e.g., '`Cgs`').

Example 3.2: Advanced transistor models.

The basic Shichman-Hodges model presented in the previous example is primarily used for manual calculations and for establishing simple overviews of the relation between transistor parameters and circuit performance. For simulations required to provide accurate results, more complex transistor models must be applied. Generic models are available together with several textbooks such as Sedra & Smith (2016), Chan Carusone, Johns & Martin (2012) and Baker (2010). Also Predictive Technology Model Website (2012) developed at Arizona State University (Cao 2011) provides a large selection of generic models for CMOS processes.

SPICE Netlist

```
* M:\LTspicebook_3rd_Edition\Tutorial03\Fig3_08.asc
M1 VDN VGN 0 0 NMOS-SH l=1um w=10um ad=10e-12 as=10e-12 pd=12u ps=12u m=1
V1 VGN 0 1.5V
V2 VDN 0 2V
V3 VGP 0 -1.5V
M2 VDP VGP 0 0 PMOS-SH l=1u w=10u ad=10e-12 as=10e-12 pd=12u ps=12u m=1
V4 VDP 0 -2V
.model NMOS NMOS
.model PMOS PMOS
.lib C:\Users\erbr\Documents\LTspiceXVII\lib\cmp\standard.mos
.model NMOS-SH nmos (Kp=190u Vto=0.57 Lambda=0.16 Gamma=0.50 Phi=0.7
+TOX=8n CGSO=0.28n CGBO=1p CGDO=0.28n CJ=1m CJSW=0.4n)
.op
.model PMOS-SH pmos (Kp=55u Vto=-0.71 Lambda=0.16 Gamma=0.75 Phi=0.7
+TOX=8n CGSO=0.21n CGBO=1p CGDO=0.21n CJ=1.5m CJSW=0.5n)
.backanno
.end
```

Spice Error Log

```
Circuit: * M:\LTspicebook_3rd_Edition\Tutorial03\Fig3_08.asc

Model "pmos-sh": Oxide thickness thinner than recommended for a level 1 MOSFET.
Instance "m2": Length shorter than recommended for a level 1 MOSFET.
Model "nmos-sh": Oxide thickness thinner than recommended for a level 1 MOSFET.
Instance "m1": Length shorter than recommended for a level 1 MOSFET.
Direct Newton iteration for .op point succeeded.
Semiconductor Device Operating Points:
                          --- MOSFET Transistors ---
Name:           m2           m1
Model:       pmos-sh      nmos-sh
Id:         -2.27e-04     1.08e-03
Vgs:        -1.50e+00     1.50e+00
Vds:        -2.00e+00     2.00e+00
Vbs:         0.00e+00     0.00e+00
Vth:        -7.10e-01     5.70e-01
Vdsat:      -7.90e-01     9.30e-01
Gm:          5.74e-04     2.33e-03
Gds:         2.75e-05     1.31e-04
Gmb:         2.57e-04     6.97e-04
Cbd:         1.12e-14     7.91e-15
Cbs:         2.10e-14     1.48e-14
Cgsov:       2.10e-15     2.80e-15
Cgdov:       2.10e-15     2.80e-15
Cgbov:       1.00e-18     1.00e-18
Cgs:         2.88e-14     2.88e-14
Cgd:         0.00e+00     0.00e+00
Cgb:         0.00e+00     0.00e+00
```

Figure 3.9: Netlist file and error log file for the circuit from Fig. 3.8.

Models for specific CMOS processes can be obtained from foundries or from MPW (Multi-Project Wafer) service providers such as MOSIS (MPW) Integrated Circuit (IC) Fabrication Service Provider (The MOSIS Service 2019). However, most foundries require non-disclosure agreements in order to provide detailed design information.

As an example, Fig. 3.10 shows BSIM3 models derived from Chan Carusone, Johns & Martin (2014) for a generic 0.35 µm process. The two models NMOS-BSIM and PMOS-BSIM are contained in a single file named BSIM3_035.lib. This makes it possible to include the models in the schematic simply by giving a reference to this file. The SPICE directive for including the library file is '`.include BSIM3_035.lib`' (or '`.inc BSIM3_035.lib`'). The library file should be placed in the same folder as the circuit schematic file or in a folder dedicated specifically to library files. If the file is placed in a folder dedicated to library files (e.g., 'MOS models'), you must define a search path to this folder using the command 'Tools → Control Panel → Sym. & Lib. Search Paths', see Fig. 3.11. Alternatively, the full path to the file may be specified.

Generic BSIM3 model for 0.35 µm CMOS process. Adapted from Carusone, Johns & Martin (2014).

```
*BSIM3_035.lib
.MODEL NMOS-BSIM NMOS LEVEL = 49
+VERSION = 3.1 TNOM = 27 TOX = 7.8E-9
+XJ = 1E-07 NCH = 2.18E+17 VTH0 = 0.48
+K1 = 6.07E-01 K2 = 1.24E-03 K3 = 9.68E+01
+K3B = -9.84E+00 W0 = 2.02E-05 NLX = 1.62E-07
+DVT0W = 0 DVT1W = 0 DVT2W = 0
+DVT0 = 2.87E+00 DVT1 = 5.86E-01 DVT2 = -1.26E-01
+U0 = 360 UA = -8.48E-10 UB = 2.27E-18
+UC = 3.27E-11 VSAT = 1.87E+05 A0 = 1.22E+00
+AGS = 2.06E-01 B0 = 9.60E-07 B1 = 4.95E-06
+KETA = -1.67E-04 A1 = 0 A2 = 3.49E-01
+RDSW = 8.18E+02 PRWG = 2.35E-02 PRWB = -8.12E-02
+WR = 9.98E-01 WINT = 1.55E-07 LINT = 4.51E-10
+DWG = -4.27E-09
+DWB = 4.07E-09 VOFF = -4.14E-02 NFACTOR = 1.61E+00
+CIT = 0 CDSC = 2.39E-04 CDSCD = 0.00E+00
+CDSCB = 0 ETA0 = 1 ETAB = -1.99E-01
+DSUB = 1 PCLM = 1.32E+00 PDIBLC1 = 2.42E-04
+PDIBLC2 = 8.27E-03 PDIBLCB = -9.99E-04 DROUT = 9.72E-04
+PSCBE1 = 7.24E+08 PSCBE2 = 9.96E-04 PVAG = 1.00E-02
+DELTA = 1.01E-02 RSH = 3.33E+00 MOBMOD = 1
+PRT = 0 UTE = -1.5 KT1 = -1.11E-01
+KT1L = 0 KT2 = 2.22E-02 UA1 = 4.34E-09
+UB1 = -7.56E-18 UC1 = -5.62E-11 AT = 3.31E+04
+WL = 0 WLN = 9.95E-01 WW = 0
+WWN = 1.00E+00 WWL = 0 LL = 0
+LLN = 1 LW = 0 LWN = 1
+LWL = 0 CAPMOD = 2 XPART = 0.5
+CGDO = 2.76E-10 CGSO = 2.76E-10 CGBO = 1.00E-12
+CJ = 9e-4 PB = 7.95E-01 MJ = 3.53E-01
+CJSW = 2.8e-10 PBSW = 7.98E-01 MJSW = 1.73E-01
+CJSWG = 1.81E-10 PBSWG = 7.96E-01 MJSWG = 1.74E-01
+CF = 0 PVTH0 = -1.80E-02 PRDSW = -7.56E+01
+PK2 = 4.48E-05 WKETA = -1.33E-03 LKETA = -8.91E-03

.MODEL PMOS-BSIM PMOS LEVEL = 49
+VERSION = 3.1 TNOM = 2.69E+01 TOX = 7.8E-9
+XJ = 1.00E-07 NCH = 8.44E+16 VTH0 = -0.6
+K1 = 4.82E-01 K2 = -2.13E-02 K3 = 8.27E+01
+K3B = -5 W0 = 5.24E-06 NLX = 2.49E-07
+DVT0W = 0.00E+00 DVT1W = 0 DVT2W = 0
+DVT0 = 3.54E-01 DVT1 = 7.52E-01 DVT2 = -2.98E-01
+U0 = 150 UA = 1E-10 UB = 1.75E-18
+UC = -2.27E-11 VSAT = 2.01E+05 A0 = 1.04E+00
+AGS = 2.90E-01 B0 = 1.94E-06 B1 = 5.01E-06
+KETA = -3.85E-03 A1 = 4.20E-03 A2 = 1.00E+00
+RDSW = 4000 PRWG = -9.54E-02 PRWB = -1.92E-03
+WR = 1 WINT = 1.47E-07 LINT = 1.04E-10
+DWG = -1.09E-08
+DWB = 1.14E-08 VOFF = -1.29E-01 NFACTOR = 2.01E+00
+CIT = 0 CDSC = 2.40E-04 CDSCD = 0
+CDSCB = 0 ETA0 = 4.07E-02 ETAB = 6.84E-03
+DSUB = 3.21E-01 PCLM = 5.96E+00 PDIBLC1 = 2.89E-03
+PDIBLC2 = -1.45E-06 PDIBLCB = -1E-03 DROUT = 9.93E-04
+PSCBE1 = 7.88E+10 PSCBE2 = 5E-10 PVAG = 15
+DELTA = 9.96E-03 RSH = 2.6 MOBMOD = 1
+PRT = 0 UTE = -1.5 KT1 = -1.09E-01
+KT1L = 0 KT2 = 2.19E-02 UA1 = 4.34E-09
+UB1 = -7.62E-18 UC1 = -5.63E-11 AT = 3.28E+04
+WL = 0 WLN = 1 WW = 0
+WWN = 1.00E+00 WWL = 0 LL = 0
+LLN = 1 LW = 0 LWN = 1
+LWL = 0 CAPMOD = 2.01E+00 XPART = 0.5
+CGDO = 2.10E-10 CGSO = 2.12E-10 CGBO = 1.00E-12
+CJ = 14e-4 PB = 9.83E-01 MJ = 5.79E-01
+CJSW = 3.2e-10 PBSW = 9.92E-01 MJSW = 3.60E-01
+CJSWG = 4.41E-11 PBSWG = 9.85E-01 MJSWG = 3.58E-01
+CF = 0 PVTH0 = 2.58E-02 PRDSW = -3.98E+01
+PK2 = 2.02E-03 WKETA = 2.72E-03 LKETA = -7.14E-03
```

Figure 3.10: Library file with BSIM3 models for a generic 0.35 µm CMOS process, adapted from Chan Carusone, Johns & Martin (2014).

The file shown in Fig. 3.10 is adapted from Chan Carusone, Johns & Martin (2014). You may get the input to the file BSIM3_035.lib from this reference. You can open (and edit) simple text files such as BSIM3_035.lib in LTspice using the command 'Files → Open' and specify 'Files of type → All Files'. Appendix B shows how to generate LTspice-compatible library files from the BSIM files in Chan Carusone, Johns & Martin (2014).

You can find a link to a file 'transistormodels.zip' containing the BSIM models used in this book on the webpage of the book, (https://bookboon.com/en/cmos-integrated-circuit-simulation-with-ltspice-ebook).

Using more advanced models will generally give more precise simulation results, and often it is useful to compare the results obtained from a simple model with the results from an advanced model in order to explain deviations in circuit behavior from the manual calculations based on the Shichman-Hodges model.

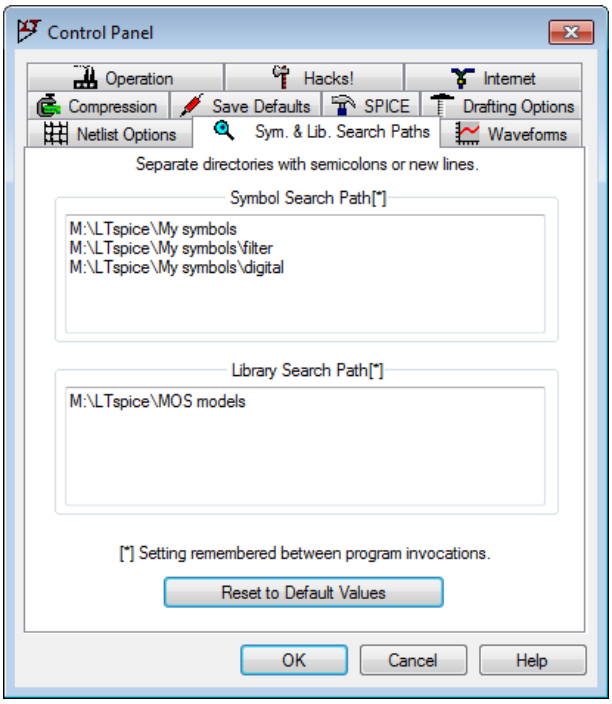


Figure 3.11: Window for defining search paths to symbol folders and library folders.

As a very simple example illustrating the differences, the circuit from Fig. 3.8 may be re-simulated with the models from the library file shown in Fig. 3.10. This results in the netlist and error log file shown in Fig. 3.12 which can be compared to Fig. 3.9, showing the results using the Shichman-Hodges model.

You may observe that there are quite significant differences between the small-signal parameters shown in Figs. 3.9 and 3.12, and it can be difficult to reach a reasonable degree of agreement between simulated results and results based on a calculation from the Shichman-Hodges model. For comparison, Eqs. (3.8) and (3.9) result in $g_m = 2.33$ mA/V and $g_{ds} = 131$ µA/V for the NMOS transistor and $g_m = 0.574$ mA/V and $g_{ds} = 27.5$ µA/V for the PMOS transistor, corresponding exactly to the simulation results from Fig. 3.9 but rather different from the values shown in Fig. 3.12.

This illustrates a need for deriving useful Shichman-Hodges parameters from a transistor simulation based on advanced models in order to be able to calculate analytical results which are useful for circuit design. Also note that the BSIM3 model does not result in values for the small-signal capacitances in the same way as the Shichman-Hodges model. Rather, derivatives of charge with respect to signal voltages are specified. For a definition of the corresponding capacitances, see for instance Tsividis & McAndrew (2010).

SPICE Netlist

```
* M:\LTspicebook_3rd_Edition\Tutorial03\Fig3_12.asc
M1 VDN VGN 0 0 NMOS-BSIM l=1um w=10um ad=10e-12 as=10e-12 pd=12u ps=12u m=1
V1 VGN 0 1.5V
V2 VDN 0 2V
V3 VGP 0 -1.5V
M2 VDP VGP 0 0 PMOS-BSIM l=1u w=10u ad=10e-12 as=10e-12 pd=12u ps=12u m=1
V4 VDP 0 -2V
.model NMOS NMOS
.model PMOS PMOS
.lib C:\Users\erbr\Documents\LTspiceXVII\lib\cmp\standard.mos
.op
.include BSIM3_035.lib
.backanno
.end
```

Spice Error Log

```
Circuit: * M:\LTspicebook_3rd_Edition\Tutorial03\Fig3_12.asc

Direct Newton iteration for .op point succeeded.
Semiconductor Device Operating Points:
                          --- BSIM3 MOSFETS ---
Name:         m2           m1
Model:     pmos-bsim    nmos-bsim
Id:       -1.60e-04     5.48e-04
Vgs:      -1.50e+00     1.50e+00
Vds:      -2.00e+00     2.00e+00
Vbs:       0.00e+00     0.00e+00
Vth:      -6.79e-01     5.43e-01
Vdsat:    -6.96e-01     6.21e-01
Gm:        3.39e-04     9.92e-04
Gds:       7.31e-06     1.03e-05
Gmb        7.54e-05     2.63e-04
Cbd:       8.14e-15     7.70e-15
Cbs:       1.52e-14     1.14e-14
Cgsov:     2.06e-15     2.67e-15
Cgdov:     2.04e-15     2.67e-15
Cgbov:     1.00e-18     9.99e-19
dQgdVgb:   3.81e-14     3.99e-14
dQgdVdb:  -1.99e-15    -2.68e-15
dQgdVsb:  -3.43e-14    -3.59e-14
dQddVgb:  -1.64e-14    -1.70e-14
dQddVdb:   1.02e-14     1.04e-14
dQddVsb:   1.79e-14     1.90e-14
dQbdVgb:  -5.36e-15    -6.01e-15
dQbdVdb:  -8.14e-15    -7.71e-15
dQbdVsb:  -1.88e-14    -1.62e-14
```

Figure 3.12: Netlist file and error log file for the circuit from Fig. 3.8 simulated with the device models included in BSIM3_035.lib.

Example 3.3: MOS transistor input characteristics.

The input characteristics of a MOS transistor show the drain current versus the gate-source voltage for fixed values of drain-source voltage and source-bulk voltage. Figure 3.13 shows an NMOS transistor drawn with LTspice and with the relevant voltage sources connected to the transistor. The transistor is defined by a simple Shichman-Hodges model for a 0.35 µm process with typical parameters, compare Fig. 3.3. The transistor dimensions are shown in the figure in blue. They have been defined in the specification window for transistor M1, compare Fig. 3.1, but as this specification does not by default appear on the schematic, it has been inserted using the command 'Edit → Text', toolbar symbol A_a or hotkey 'T'.

In the LTspice netlist, the text appears as a comment, i.e., a line starting with an asterix (∗). This is a 'quick and dirty' way of showing the transistor dimensions because there is nothing to ensure that the comment matches the transistor specification, but it will do for now when we are considering just a single

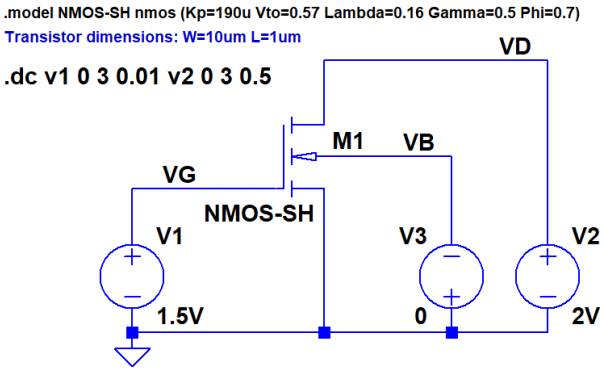


Figure 3.13: LTspice schematic for simulation of transistor characteristics.

transistor. In circuits with more transistors, it is highly advisable to show the transistor specification in a way that annotates to the netlist. In Tutorial 4, Example 4.1, it is explained how to do this.

When simulating the transistor characteristics for a transistor to be applied in a circuit design, it is important to use transistor dimensions, in particular channel length, corresponding to the dimensions planned to be used for the circuit design, and when modifying the channel length, the value of λ should also be re-calculated since λ is inversely proportional to the channel length L.

First, the drain current is simulated versus the gate-source voltage with $V_{SB} = 0$ and with V_{DS} varying in steps of 0.5 V from 0 to 3 V. The simulation command for this is the '.dc' directive with the voltage source '`V1`' as the first source and '`V2`' as the second source. The voltage '`V1`' is specified to sweep from 0 to 3 V with an increment of 0.01 V in order to obtain a reasonably smooth curve showing I_D and the derivative of I_D with respect to V_{GS}. The resulting plot of I_D versus V_{GS} is shown in Fig. 3.14.

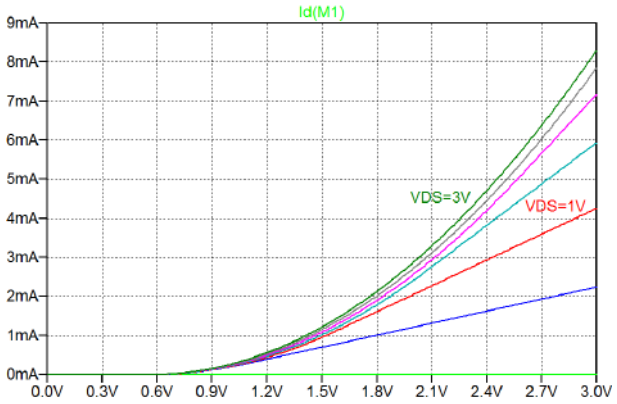


Figure 3.14: Plot of I_D versus V_{GS} for different values of V_{DS} and $V_{SB} = 0$ V.

The plot shows seven curves corresponding to $V_{DS} = 0,\ 0.5,\ 1.0,\ 1.5,\ 2.0,\ 2.5$ and 3.0 V, respectively. In the plot, labels have been inserted for two of the curves using the command 'Plot Settings → Notes & Annotations → Place Text'. You may notice that the plot does not give specific information about which quantity is used for the horizontal axis. By default, the horizontal axis is the first parameter stepped by

the '.dc' directive, i.e., 'V1' (or V_{GS}) for the plot in Fig. 3.14. If you wish to indicate this on the plot, you may either insert the information using the command 'Plot Settings → Notes & Annotations → Place Text', or you may change the 'Quantity Plotted' from 'V1' to 'v(VGS)' by moving the cursor to the x-axis and applying a right-click on the mouse, compare Fig. 1.21.

For the two curves corresponding to $V_{DS} = 2.5$ V and 3.0 V, the transistor is in the active region for the simulated range of V_{GS}, and the curves show a parabolic relation between I_D and V_{GS}. For the other values of V_{DS}, the transistor is in the triode region for large values of V_{GS}, leading to a linear relation between I_D and V_{GS}.

As explained for Fig. 2.4, a cursor can be activated by a left-click on the trace label above the plot window. The cursor opens with a horizontal value in the middle of the plot (i.e., $V_{GS} = 1.5$ V) and is attached to the first trace, i.e., the trace corresponding to $V_{DS} = 0$ V. The cursor may be moved to the other traces by positioning the mouse over the cursor (a '1' will appear on the plot) and entering the 'up' arrow key on the keyboard. By using the arrow keys 'up' and 'down', the cursor can be moved between the traces. A right-click when the mouse is positioned over the cursor opens a window with cursor step information, i.e., information about the trace where the cursor is attached. Also, when a cursor is included in the plot, a window is open with information about the position of the cursor, compare Fig. 2.4.

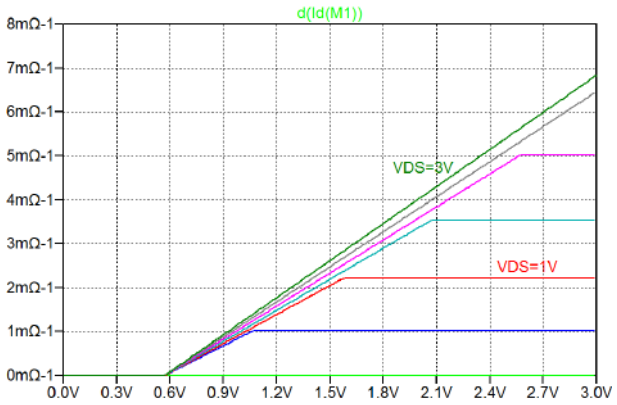


Figure 3.15: Plot of $\partial i_D/\partial v_{GS}$ versus V_{GS} for different values of V_{DS} and $V_{SB} = 0$ V.

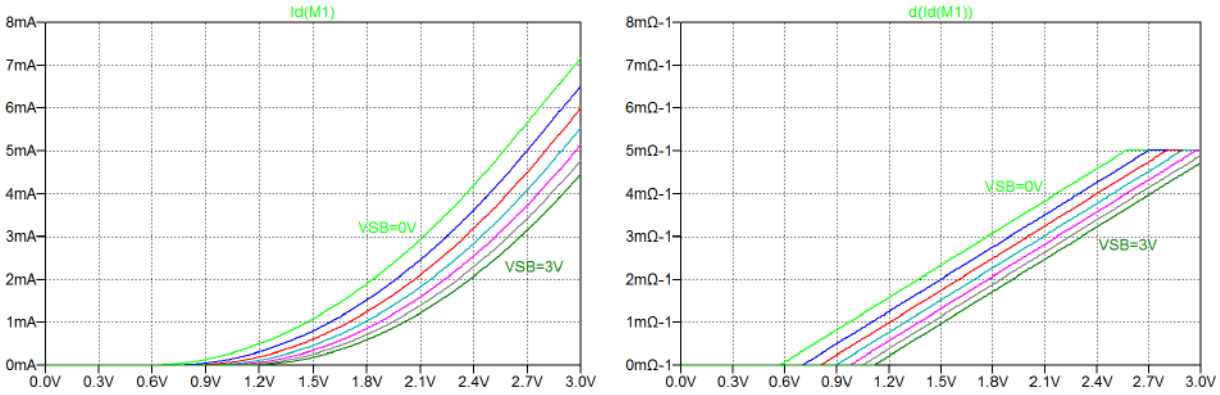
Next, Fig. 3.15 shows the derivative of I_D with respect to V_{GS} versus V_{GS}. This is obtained from the same simulation, just by editing the variable to be plotted: A right-click on the trace label above the plot opens a window with the 'Expression Editor'. The window shows the current expression being plotted. For Fig. 3.14, this is 'Id(M1)'. By modifying this to 'd(Id(M1))' and clicking 'OK', the plot window shown in Fig. 3.15 appears. The function 'd()' computes a difference-based derivative, see the LTspice 'Help' function (Waveform Arithmetic).

The derivative of I_D with respect to V_{GS} is the transconductance g_m. In the active region, g_m increases linearly with V_{GS} as found from Eq. (3.8), and in the triode region, g_m is found from Eq. (3.2) as

$$g_m = \mu_n C_{ox} \left(\frac{W}{L}\right) V_{DS}(1 + \lambda V_{DS})$$

This relation shows that the Shichman-Hodges transistor model leads to a constant value of g_m in the triode region as also clearly seen in Fig. 3.15.

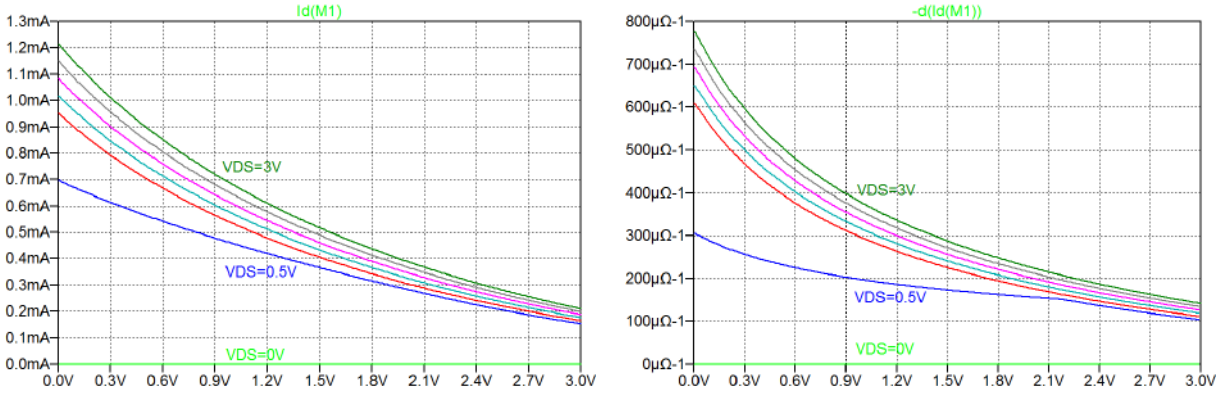
Next, we may investigate the influence of the source-bulk voltage V_{SB}. For the previous simulations, V_{SB} was equal to 0 but as indicated by Eq. (3.4), a positive value of V_{SB}, reverse biasing the channel-bulk junction, will lead to an increase in the threshold voltage V_t. Figure 3.16 shows a plot of I_D and $\partial i_D/\partial v_{GS}$ versus V_{GS} with a fixed value of 2 V for V_{DS} and V_{SB} swept from 0 to 3 V in steps of 0.5 V. The shift of the curves caused by the increase of the threshold voltage is evident.



(a) (b)

Figure 3.16: Plot of I_D (a) and $\partial i_D/\partial v_{GS}$ (b) versus V_{GS} for different values of V_{SB} and $V_{DS} = 2$ V.

As indicated by Eq. (3.10), the bulk terminal may also be used as the input signal to the transistor instead of the gate terminal. An increase in V_{SB} causes a decrease in I_D for a fixed value of V_{GS} and V_{DS}. The small-signal parameter g_{mb} is given by $g_{mb} = \partial i_D/\partial v_{BS} = -\partial i_D/\partial v_{SB}$, and you may find input characteristics for the transistor with the bulk as the input terminal in a similar way as the input characteristics with the gate as the input terminal. Figure 3.17 shows a plot of I_D versus V_{SB} for $V_{GS} =$ 1.5 V and V_{DS} swept from 0 to 3 V in steps of 0.5 V. Also shown is the bulk transconductance g_{mb} as a



(a) (b)

Figure 3.17: Plot of I_D (a) and $\partial i_D/\partial v_{BS}$ (b) versus V_{SB} for different values of V_{DS} and $V_{GS} = 1.5$ V.

plot of '`-d(iD(M1))`'. Notice that for $V_{DS} = 0$ (the green traces), the transistor is off, and for $V_{DS} = 0.5$ V (the blue traces), the transistor is in the triode region for small values of V_{SB}. Otherwise, the transistor is in the active region.

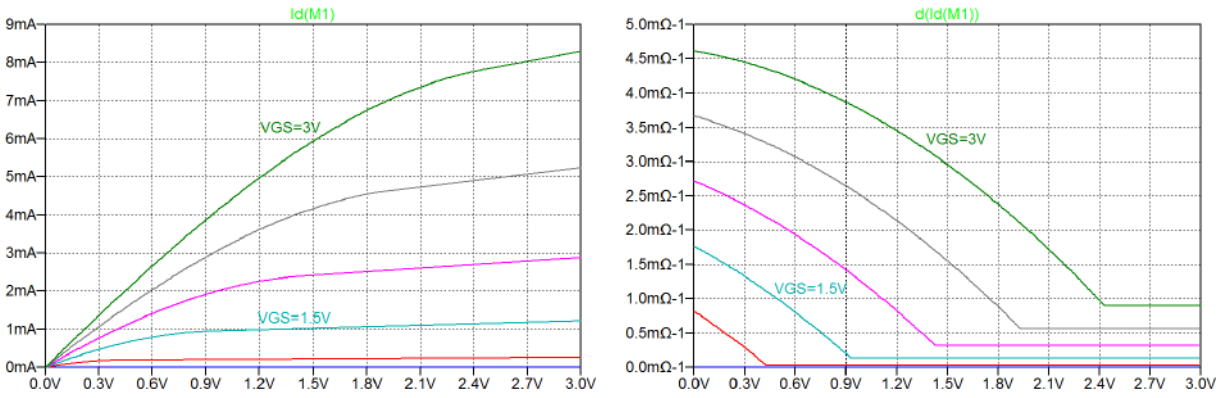
PMOS transistors: The input characteristics shown above were all simulated for an NMOS transistor. Of course, similar input characteristics may be simulated for a PMOS transistor. Some care is needed in order to avoid confusion of signs for the PMOS transistor currents and voltages. Recall that LTspice assumes the positive direction of current to be into the transistor, so in LTspice notation, the drain current of a PMOS transistor is negative as described earlier in this tutorial, see Example 3.1.

Example 3.4: MOS transistor output characteristics.

The output characteristics of a MOS transistor show the drain current versus the drain-source voltage for fixed values of gate-source voltage and source-bulk voltage. The output characteristics are obtained from the circuit from Fig. 3.13 by a dc sweep with '`V2`' as the first source (increment 0.01 V) and '`V1`' as the second source (increment 0.5 V). The resulting plot of I_D versus V_{DS} is shown in Fig. 3.18(a).

Notice that for the first two traces corresponding to $V_{GS} = 0$ V and $V_{GS} = 0.5$ V, respectively, the transistor is off and $I_D = 0$. The following five traces show the transistor in the triode region for small values of V_{DS} and in the active region for large values of V_{DS}.

Observe the slope of the characteristics in the active region caused by the channel-length modulation specified by the non-zero value of λ, see Eq. (3.3). This causes a non-zero value of g_{ds} which may be plotted as '`d(Id(M1))`', see Fig. 3.18(b). You will see that g_{ds} saturates at a constant level for the transistor in the active region and that it increases with increasing value of V_{GS} as shown by Eq. (3.9).



(a) (b)

Figure 3.18: Output characteristics: plot of I_D (a) and $\partial i_D/\partial v_{DS}$ (b) versus V_{DS} for different values of V_{GS} with $V_{SB} = 0$ V.

PMOS transistors: As for the input characteristics, some care is needed in order to avoid confusion of signs when simulation the characteristics for a PMOS transistor. Figure 3.19 shows the schematic for simulating the characteristics of a PMOS transistor. Observe that the voltages have been reversed compared to Fig. 3.13.

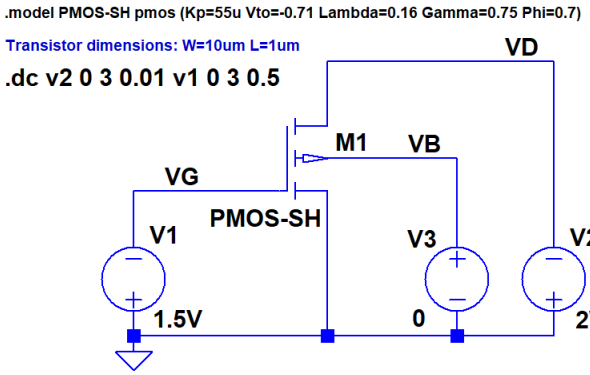


Figure 3.19: LTspice schematic for simulation of PMOS transistor characteristics.

Figure 3.20 shows the output characteristics of the PMOS transistor. In order to comply with the normal textbook sign definition for the drain current, the plot shows '`-Id(M1)`' and '`d(-Id(M1))`' rather than '`Id(M1)`' and '`d(Id(M1))`'.

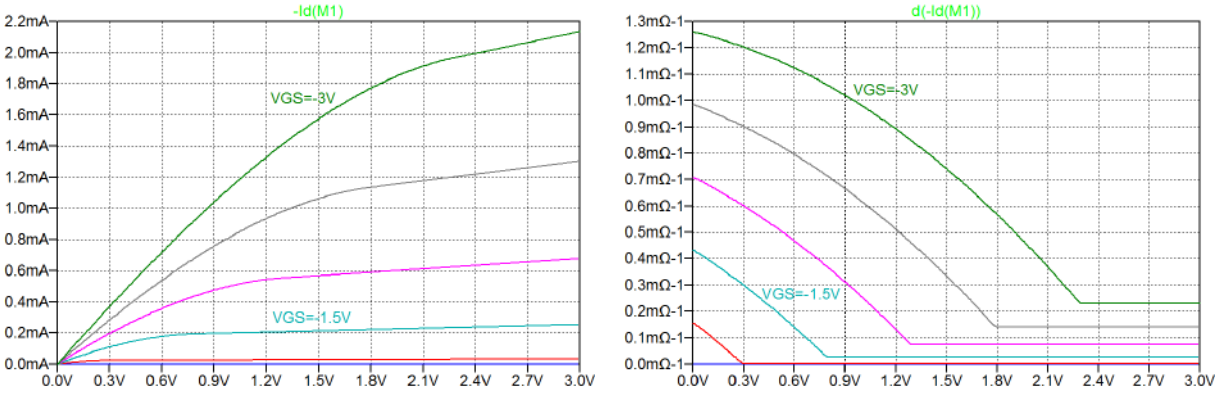


Figure 3.20: Plot of I_D (a) and $\partial i_D/\partial v_{SD}$ (b) versus V_{SD} for different values of V_{SG} and $V_{BS} = 0$ V for the PMOS transistor from Fig. 3.19.

Example 3.5: Deriving transistor parameters from input and output characteristics.

The characteristics simulated in the previous examples were all based on the simple Shichman-Hodges model. Using more accurate models may result in major discrepancies from the characteristics in the previous examples. Replacing the Shichman-Hodges transistor models with the models from Fig. 3.10 results in input characteristics and output characteristics for the NMOS transistor as shown in Fig. 3.21.

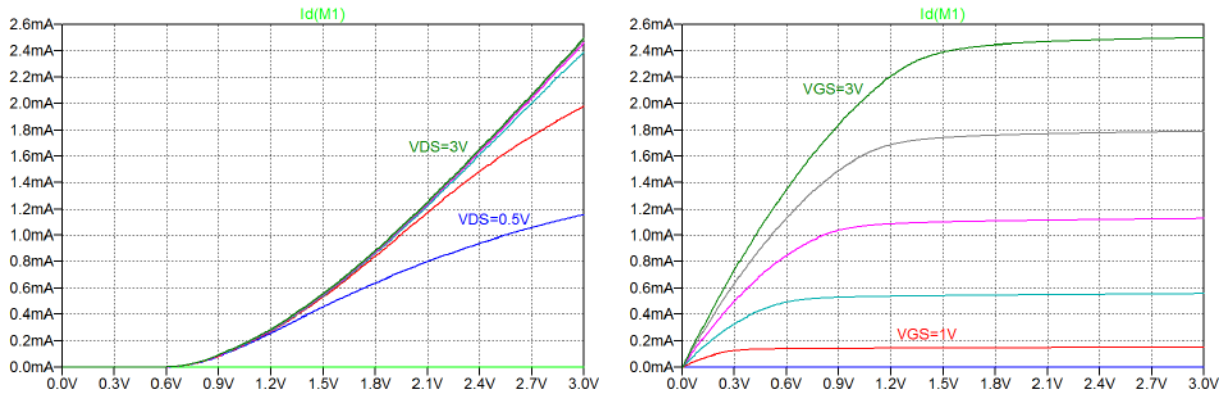


Figure 3.21: Input characteristics, I_D versus V_{GS}, (a), and output characteristics, I_D versus V_{DS}, (b), simulated for an NMOS transistor using the BSIM3 transistor model from Fig. 3.10.

Clearly, there are major differences between the curves in Fig. 3.21 and Figs. 3.14 and 3.18. Not only is the scale of the y-axis different (by a factor of more than 3). Also the shape of the input characteristics is different. The major reason for this difference is the mobility degradation, causing μ_n to decrease for large values of $V_{GS} - V_t$ (Chan Carusone, Johns & Martin 2012). The mobility degradation causes the drain current to increase almost linearly with $V_{GS} - V_t$, rather than the quadratic relation predicted by Eq. (3.3), and this also implies that the transconductance in the active region does not increase linearly with $V_{GS} - V_t$ but tends to saturate. The shape of output characteristics in Fig. 3.21 resembles the shape in Fig. 3.18, but apparently with a significantly higher output resistance in the active region, corresponding to a smaller value of the channel length parameter λ.

In this example, we will show how the LTspice model parameters for the Shichman-Hodges model can be modified so that a somewhat better match of the input and output characteristics to those shown in Fig. 3.21 is obtained, leading to somewhat better results when performing hand calculations using the Shichman-Hodges model during a circuit design.

We will use a simple, heuristic approach. More advanced methods (e.g., using linear regression techniques) can be found in the literature, e.g., Allen & Holberg (2012, Appendix B), but with the large discrepancies which cannot be avoided, a few simple methods may be sufficient in many cases.

In order to be able to compare the simulation results for the Shichman-Hodges model and an advanced model, it is useful to simulate the two models in parallel using the circuit configuration shown in Fig. 3.22. This figure shows the circuit from Fig. 3.13 augmented with an extra transistor '`M2`' connected in parallel with '`M1`' and specified with the model '`NMOS-BSIM`' from the library BSIM3_035.lib. This parallel connection is achieved without drawing wires between the two transistor. The labels defining the names of the nodes is enough to establish the correct connections in the LTspice netlist.

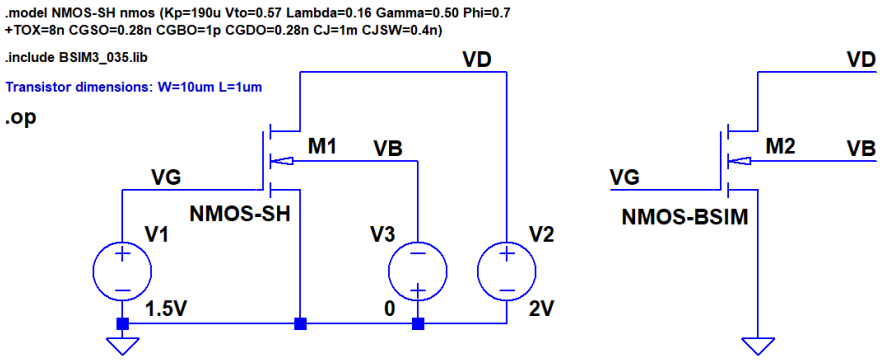


Figure 3.22: LTspice schematic for simulation of transistor characteristics for two different transistor models.

Matching small-signal parameters: If the transistor is known to operate with small signal variations from a specific bias point, a simple approach to finding values for the Shichman-Hodges parameters is to derive them from the information about the operating point and the small-signal parameters in the operating point provided by a '.op' simulation in LTspice. For the transistors in Fig. 3.22, the dc values specified for '`V1`', '`V2`' and '`V3`' define an operating point $V_{GS} = 1.5$ V, $V_{DS} = 2.0$ V and $V_{SB} = 0$ V with the transistor in the active region. Running a '.op' simulation results in an output file with the operating point information shown in Fig. 3.23, and the error log file displayed when entering 'Ctrl-L' provides the information about the small-signal parameters also shown in Fig. 3.23.

Output file

```
        --- Operating Point ---

V(vd):          2               voltage
V(vg):          1.5             voltage
V(vb):          0               voltage
Id(M2):         0.000548031     device_current
Ig(M2):         0               device_current
Ib(M2):         -2.001e-012     device_current
Is(M2):         -0.000548031    device_current
Id(M1):         0.00108458      device_current
Ig(M1):         0               device_current
Ib(M1):         -2.01e-012      device_current
Is(M1):         -0.00108458     device_current
I(V3):          -4.011e-012     device_current
I(V2):          -0.00163262     device_current
I(V1):          0               device_current
```

SPICE Error Log

```
                Semiconductor Device Operating Points:

   --- MOSFET Transistors ---          --- BSIM3 MOSFETS ---
Name:           m1                  Name:           m2
Model:          nmos-sh             Model:          nmos-bsim
Id:             1.08e-03            Id:             5.48e-04
Vgs:            1.50e+00            Vgs:            1.50e+00
Vds:            2.00e+00            Vds:            2.00e+00
Vbs:            0.00e+00            Vbs:            0.00e+00
Vth:            5.70e-01            Vth:            5.43e-01
Vdsat:          9.30e-01            Vdsat:          6.21e-01
Gm:             2.33e-03            Gm:             9.92e-04
Gds:            1.31e-04            Gds:            1.03e-05
Gmb:            6.97e-04            Gmb             2.63e-04
Cbd:            7.91e-15            Cbd:            7.70e-15
Cbs:            1.48e-14            Cbs:            1.14e-14
Cgsov:          2.80e-15            Cgsov:          2.67e-15
Cgdov:          2.80e-15            Cgdov:          2.67e-15
Cgbov:          1.00e-18            Cgbov:          9.99e-19
Cgs:            2.88e-14            dQgdVgb:        3.99e-14
Cgd:            0.00e+00            dQgdVdb:       -2.68e-15
Cgb:            0.00e+00            dQgdVsb:       -3.59e-14
                                    dQddVgb:       -1.70e-14
                                    dQddVdb:        1.04e-14
                                    dQddVsb:        1.90e-14
                                    dQbdVgb:       -6.01e-15
                                    dQbdVdb:       -7.71e-15
                                    dQbdVsb:       -1.62e-14
```

Figure 3.23: Operating point information from output file and from SPICE Error Log, Shichman-Hodges model (transistor M1) and BSIM3 model (transistor M2) for the circuit from Fig. 3.22.

From the small-signal parameters, the transistor parameters are calculated as follows, using Eqs. (3.9), (3.8) and (3.3):

$$\lambda = \frac{g_{ds}}{I_D - g_{ds}V_{DS}} \tag{3.11}$$

$$V_t = V_{GS} - \frac{2\,I_D}{g_m} \tag{3.12}$$

$$K_p = \frac{g_m}{(W/L)(V_{GS}-V_t)(1+\lambda V_{DS})} = \left(\frac{g_m}{I_D}\right)^2 \left(\frac{I_D - g_{ds}V_{DS}}{2(W/L)}\right) \tag{3.13}$$

Assuming a value of 0.7 V for $|2\Phi_F|$, the bulk effect parameter γ may be calculated using Eq. (3.10) with $V_{SB} = 0$:

$$\gamma = 2\sqrt{|2\Phi_F|}\left(\frac{g_{mb}}{g_m}\right) = 1.67 \times \frac{g_{mb}}{g_m} \tag{3.14}$$

For the simulation results from Fig. 3.23, this results in $\lambda = 0.0195\ \mathrm{V}^{-1}$, $V_t = 0.395$ V, $K_p = 86.4\ \mu\mathrm{A/V}^2$ and $\gamma = 0.44\ \sqrt{\mathrm{V}}$.

With these model parameters inserted for the model NMOS-SH, the '.op' analysis gives the operating point information shown in Fig. 3.24. Obviously, there is a close match between the bias point information and small-signal parameters for '`M1`' and '`M2`'.

Output file

```
        --- Operating Point ---

V(vd):          2               voltage
V(vg):          1.5             voltage
V(vb):          0               voltage
Id(M2):         0.000548031     device_current
Ig(M2):         0               device_current
Ib(M2):         -2.001e-012     device_current
Is(M2):         -0.000548031    device_current
Id(M1):         0.000548055     device_current
Ig(M1):         0               device_current
Ib(M1):         -2.01e-012      device_current
Is(M1):         -0.000548055    device_current
I(V3):          -4.011e-012     device_current
I(V2):          -0.00109609     device_current
I(V1):          0               device_current
```

SPICE Error Log

```
                Semiconductor Device Operating Points:

   --- MOSFET Transistors ---          --- BSIM3 MOSFETS ---
Name:            m1                    Name:          m2
Model:        nmos035                  Model:      nmos-bsim
Id:          5.48e-04                  Id:          5.48e-04
Vgs:         1.50e+00                  Vgs:         1.50e+00
Vds:         2.00e+00                  Vds:         2.00e+00
Vbs:         0.00e+00                  Vbs:         0.00e+00
Vth:         3.95e-01                  Vth:         5.43e-01
Vdsat:       1.11e+00                  Vdsat:       6.21e-01
Gm:          9.92e-04                  Gm:          9.92e-04
Gds:         1.03e-05                  Gds:         1.03e-05
Gmb:         2.61e-04                  Gmb          2.63e-04
Cbd:         7.91e-15                  Cbd:         7.70e-15
Cbs:         1.48e-14                  Cbs:         1.14e-14
Cgsov:       2.00e-21                  Cgsov:       2.67e-15
Cgdov:       2.00e-21                  Cgdov:       2.67e-15
Cgbov:       2.00e-22                  Cgbov:       9.99e-19
Cgs:         2.88e-14                  dQgdVgb:     3.99e-14
Cgd:         0.00e+00                  dQgdVdb:    -2.68e-15
Cgb:         0.00e+00                  dQgdVsb:    -3.59e-14
                                       dQddVgb:    -1.70e-14
                                       dQddVdb:     1.04e-14
                                       dQddVsb:     1.90e-14
                                       dQbdVgb:    -6.01e-15
                                       dQbdVdb:    -7.71e-15
                                       dQbdVsb:    -1.62e-14
```

Figure 3.24: Operating point information from output file and from SPICE Error Log with adjusted model parameters for the Shichman-Hodges model (transistor M1).

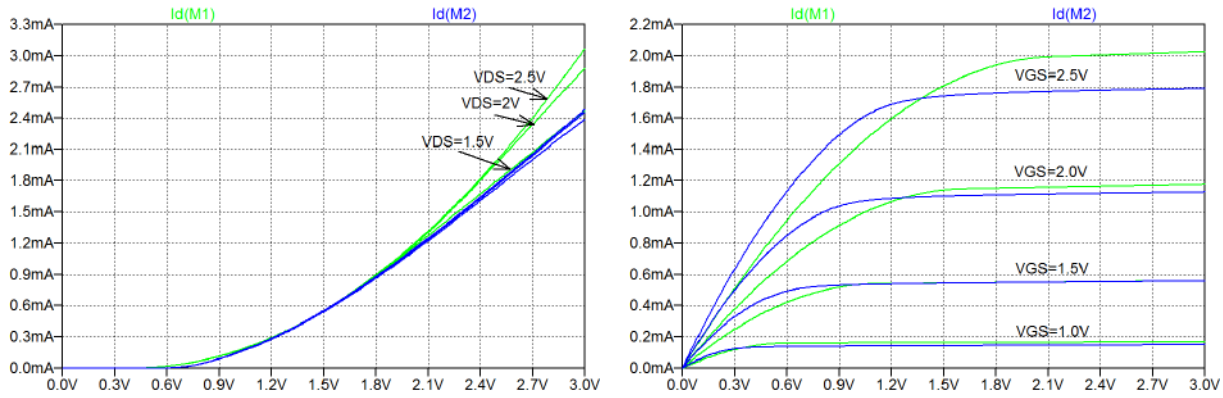


Figure 3.25: Input characteristics, I_D versus V_{GS}, (a), and output characteristics, I_D versus V_{DS}, (b), simulated for both Shichman-Hodges model with adjusted parameters (green traces) and BSIM3 model (blue traces).

While the transistor parameters have been adjusted to match in the bias point, this does not necessarily ensures a good match over a range of variations. This is illustrated in Fig. 3.25, showing simulated input characteristics and output characteristics for both transistors in the same plot. For simplicity, the input characteristics are shown only for $V_{DS} = 1.5$, 2.0 and 2.5 V, and the output characteristics are shown only for $V_{GS} = 1.0$, 1.5, 2.0 and 2.5 V. Evidently, the match between the two transistors is reasonable for variations around the bias point but is less good when the transistors operate in the triode region and for large values of V_{GS} and V_{DS} where the mobility degradation is important.

Example 3.6: Simulating small-signal parameters using '.dc' simulations and '.measure' directives.

In the previous examples, the small-signal parameters g_m and g_{ds} have been simulated directly as derivatives of the drain current, Figs. 3.15 and 3.18, and they have been shown as functions of V_{GS} and V_{DS}, respectively. However, when designing CMOS circuits, it may be useful to have the small-signal parameters also versus the bias current and versus the layout parameters channel width W and channel length L. For the Shichman-Hodges model, the small-signal parameters can be expected to rescale according to Eqs. (3.8) and (3.9), but for the BSIM models, these equations are only fairly rough approximations. In this example, we show how the small-signal parameters g_m and g_{ds} can be simulated using '.dc' simulations and how '.measure' directives can be used in order to obtain plots of g_m and g_{ds} versus the bias current I_D or the channel width W.

The starting point is the simple circuit shown in Fig. 3.26 where we use the BSIM transistor model from Fig. 3.10. For finding g_m, we run a '.dc' simulation where '`V1`' is swept from 0 to 3 V and '`V2`' is fixed at 2.5 V. With a threshold voltage V_t of about 0.5 V, this implies that the transistor is in the active region for the simulated range of V_G when $V_G > V_t$. We have selected a fixed channel length L of 1 µm but the channel width W is defined using a '.step' directive. For the circuit shown, W is stepped through the values 10, 20, 30, 40 and 50 µm. From the '.dc' simulation, we may plot i_D and g_m versus V_{GS} as

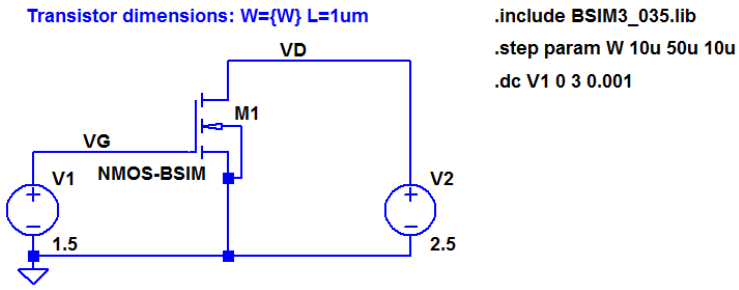


Figure 3.26: LTspice schematic for simulation of g_m for a BSIM n-channel transistor.

shown in Fig. 3.27. In this plot, we have specifically labeled the x-axis '`v(VG)`' by moving the cursor to the x-axis, right-clicking and specifying the 'Quantity Plotted' to be '`v(VG)`' rather than the default '`V1`'. Also, we have used separate panes for showing I_D and g_m ('Plot Settings → Add Plot Pane'). We notice that the maximum drain current is about 2.5 mA for $W = 10$ µm and about 12.5 mA for $W = 50$ µm.

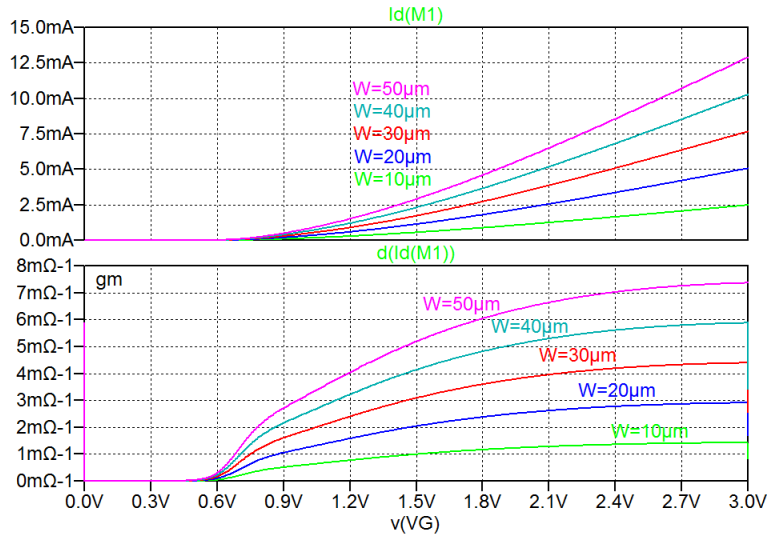


Figure 3.27: Simulation plot of I_D and g_m versus V_G for the BSIM n-channel transistor.

In order to find g_m versus the bias current I_D, we move the cursor to the x-axis and right-click. In the specification window, we change the 'Quantity Plotted' to be '`Id(M1)`'. This changes the plot window to the plot shown in Fig. 3.28. We notice that the top pane just shows I_D versus I_D so we may delete this pane. The lower pane shows g_m versus I_D. Obviously, each of the curves are plotted only for the range of I_D which is obtained with $0 < V_G < 3$ V, compare to the top pane in Fig. 3.27.

By deleting the top pane and modifying the range for the x-axis and the y-axis, we obtain the plot in Fig. 3.29, showing g_m versus I_D for different values of W.

In order to show g_m versus W, we cannot just change the x-axis quantity to W since W is not a function of the variable '`V1`' which is swept in the '.dc' simulation. Instead, we may use a '.measure' (or '.meas')

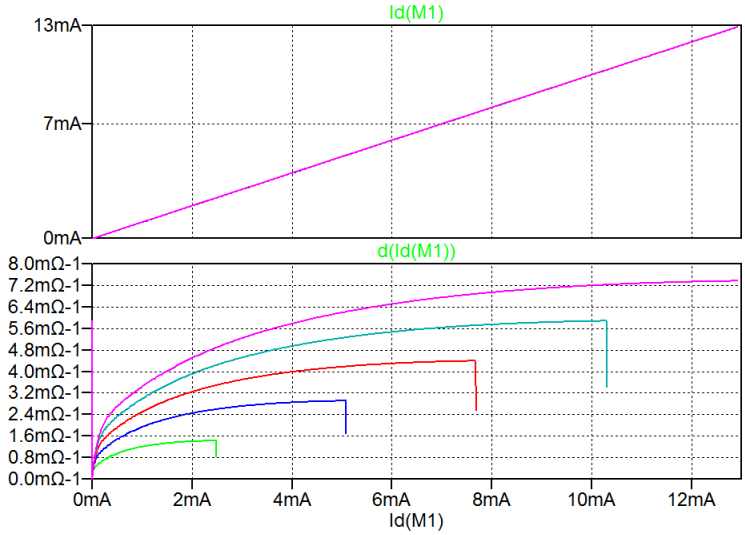


Figure 3.28: Simulation plot of I_D and g_m versus I_D for the BSIM n-channel transistor.

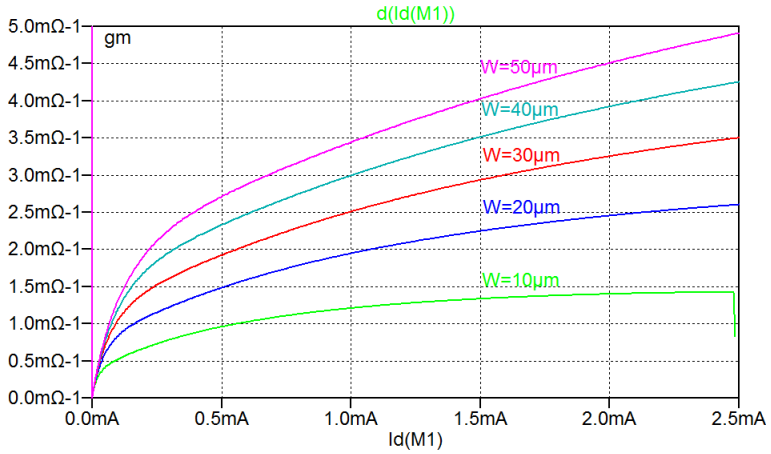


Figure 3.29: Simulation plot of g_m versus I_D for different values of W.

directive to find g_m for specified values of the bias drain current I_D or specified values of the bias gate voltage V_G. Figure 3.30 shows the schematic modified to include a '.step' directive for the bias current and a '.meas' directive which calculates g_m for the different values of the bias current. The results of the '.meas' directive are found in the error log file ('Ctrl-L'). When right-clicking in the error log file, a small dialogue box opens and you can select 'Plot .step'ed .meas data' which opens a window in the waveform viewer showing g_m versus W for the specified values of the bias current, see Fig. 3.31.

For finding g_{ds}, we must change the voltage to be swept from '`V1`' to '`V2`', i.e., the drain voltage in Fig. 3.26. For this simulation, the value of the gate voltage is $V_G = 1.5$ V. The resulting plot of I_D and g_{ds} is shown in Fig. 3.32. Evidently, the transistor is in the triode region for small values of v_{DS}.

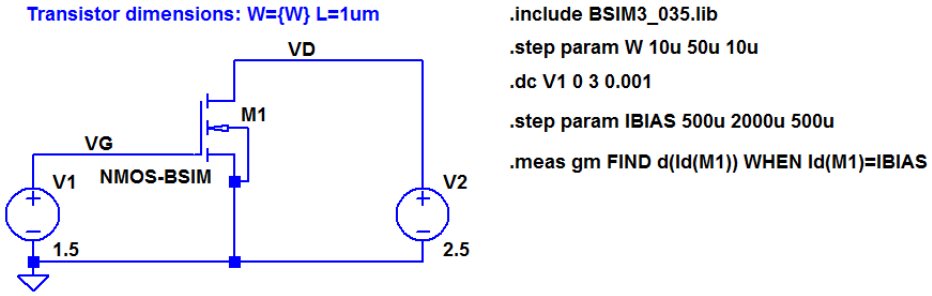


Figure 3.30: LTspice schematic for simulation of g_m using a '.meas' directive.

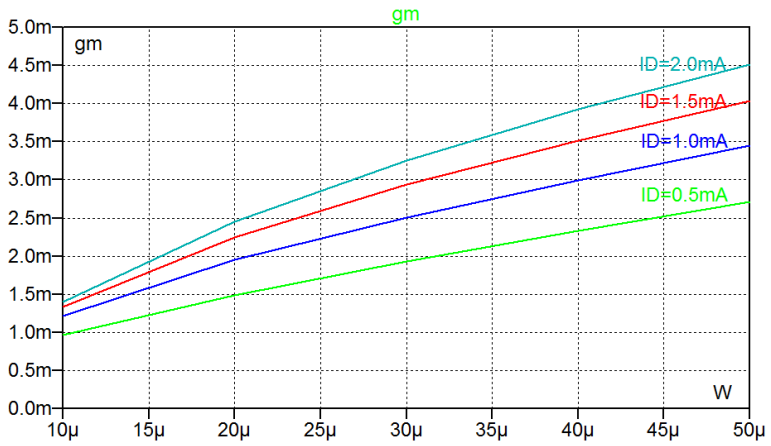


Figure 3.31: Simulation plot of g_m versus W for different values of I_D.

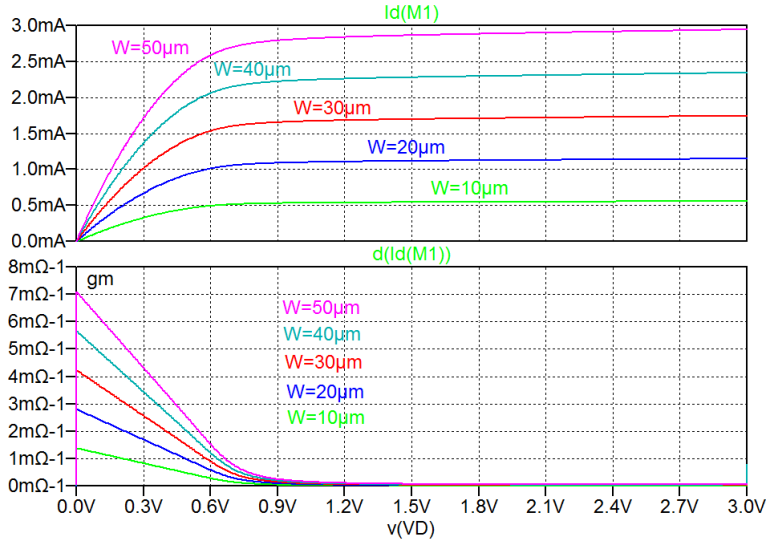


Figure 3.32: Simulation plot of I_D and g_{ds} versus V_D with $V_G = 1.5$ V.

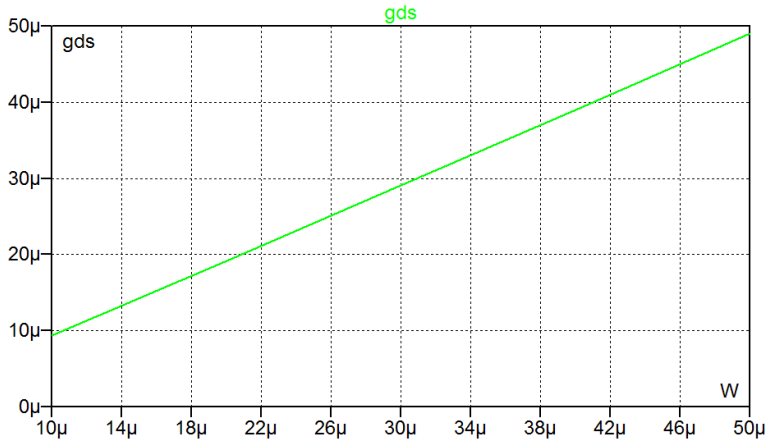


Figure 3.33: Simulation plot of g_{ds} versus W with $V_G = 1.5$ V and $V_D = 2.5$ V.

In order to show g_{ds} versus W for fixed values of the bias voltages, $V_G = 1.5$ V and $V_D = 2.5$ V, we may use a '.meas' directive: '`meas gds find d(ID(M1)) when v(VD)=2.5`'. From the error log file, we find the results of the '.meas' calculations and the plot in Fig. 3.33 shows g_{ds} versus W for $V_G = 1.5$ V and $V_D = 2.5$ V.

For showing g_{ds} versus W for different values V_G, we may include a '.step' directive for V_G. However, it may also be desirable to show g_{ds} for different values of the bias current I_D, and this cannot be achieved simply by a '.step' directive because I_D depends primarily on V_{GS} which is not changed during the simulation. An alternative approach to vary I_D is to insert a current mirror transistor as shown in Fig. 3.34. In this way, the input current can be stepped through different values and with both transistors in the active region, the drain bias current for M_1 is almost the same as the input current I_{BIAS} to M_2. Figure 3.35 shows g_{ds} versus W for four different values of the bias current. It should be noted that this approach is only directly applicable when both M_1 and M_2 are in the active region. In the next example, we show an alternative approach which can also be used for the transistors in the triode region.

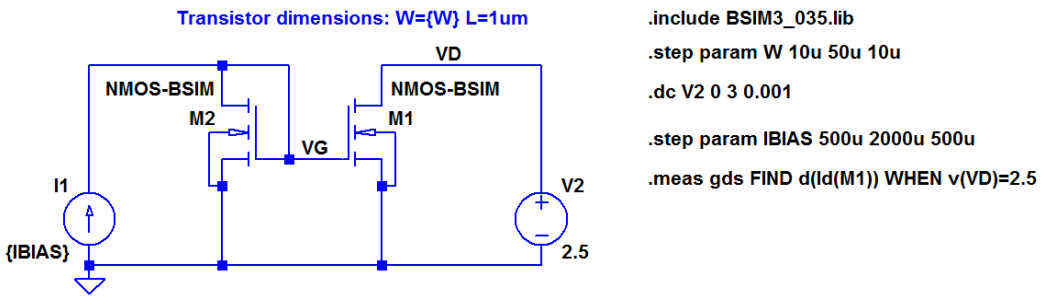


Figure 3.34: LTspice schematic for simulation of g_{ds} for different values of bias current using a '.meas' directive.

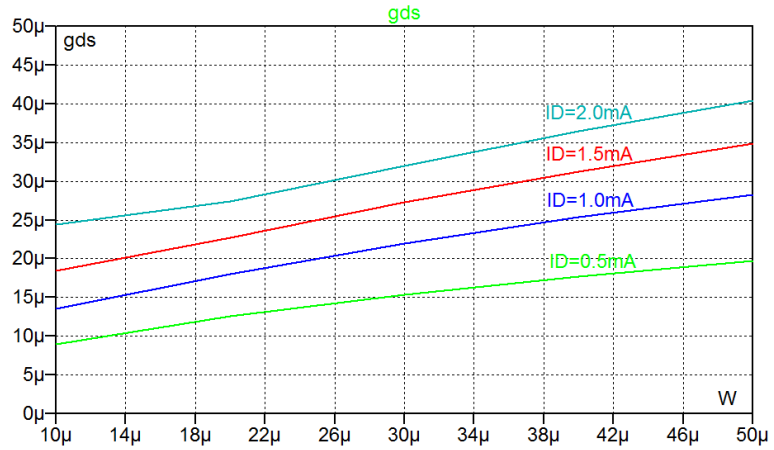


Figure 3.35: Simulation plot of g_{ds} versus W with for $V_D = 2.5$ V and different values of bias current.

Example 3.7: Simulating small-signal parameters using the '.tf' simulation.

An alternative to the use of '.dc' simulations and '.measure' directives for finding g_m and g_{ds} is the '.tf' simulation. This simulation provides small-signal input resistances and output resistances and by using suitable circuit configurations, these parameters can be used for finding the small-signal parameters. In this example, we show some simple circuits which make it possible to simulate the small-signal parameters using the '.tf' simulation. By defining a design variable as a parameter in the circuit and sweeping the parameter using a '.step' directive, the small-signal parameter can be plotted versus the design variable. As a starting point, we consider a diode-connected transistor as shown in Fig. 3.36. With $V_{GS} = V_{DS}$, the transistor is in the active region, and from the small-signal diagram, we find an input resistance of $(g_m + g_{ds})^{-1}$. Simulating the small-signal transfer function v_{gs}/i_d results in both the transfer function, the input resistance and the output resistance being equal to $(g_m + g_{ds})^{-1}$. With $g_{ds} \ll g_m$, this can be used as an approximate value for $1/g_m$.

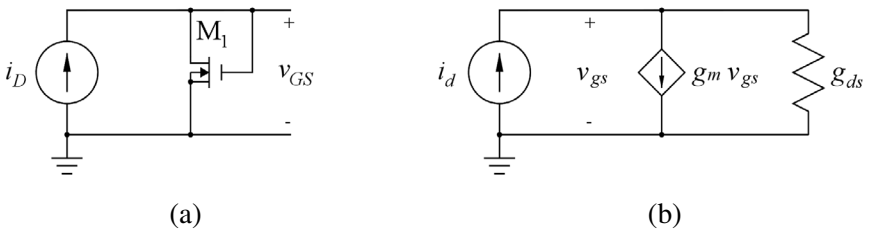


Figure 3.36: Diode-connected NMOS transistor (a) and small-signal diagram for the diode-connected transistor (b).

For finding g_{ds}, we may connect an additional transistor to form a current mirror. Figure 3.37 shows the LTspice schematic with the current mirror output transistor '`M2`'. This is connected to a controlled current source ensuring that $I_{D1} = I_{D2}$. With identical transistors, this implies that $V_{DS1} = V_{DS2}$, so from this configuration both g_m and g_{ds} can be found from a single '.tf' simulation with '`v(VD2)`' as the output and '`I1`' as the source. With this simulation, $g_m + g_{ds}$ is found as the reciprocal of the input impedance and g_{ds} is the reciprocal of the output impedance.

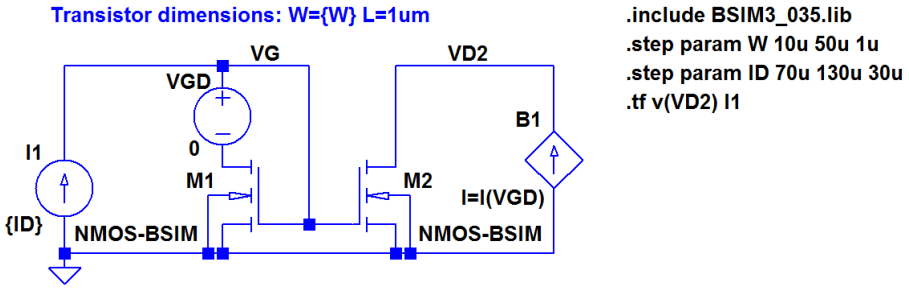
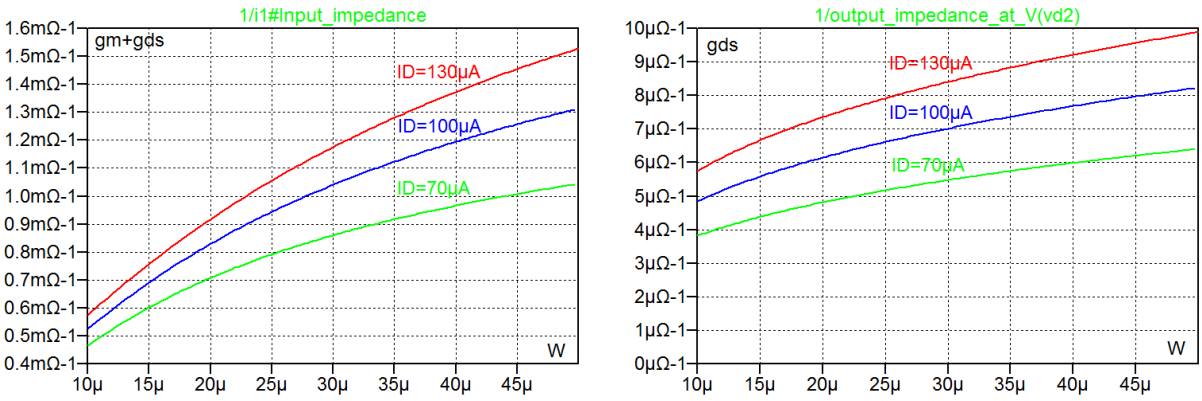


Figure 3.37: NMOS current mirror for simulating both g_{ds} and $g_m + g_{ds}$.

For the circuit, you may specify several design variables as parameters and you may include multiple '.step' directives. The '.step' directives are executed in the sequence in which they appear in the SPICE Netlist ('View → SPICE Netlist'). They appear in the netlist in the same sequence as inserted in the schematic, and the first '.step' directive determines the default x-axis in the plot window. In Fig. 3.37, both transistor channel width W and bias current I_D are specified as parameters, and the '.step' directive for W is inserted first.

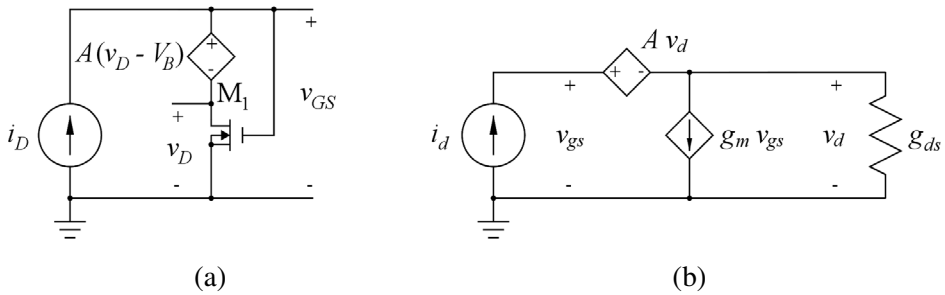


(a) (b)

Figure 3.38: Simulation of $g_m + g_{ds}$ (a) and g_{ds} (b) versus W for different levels of bias current.

Figure 3.38 shows the results of a '.tf' simulation for finding $g_m + g_{ds}$ and g_{ds}. The traces are added using 'Plot Settings → Add trace' where you click on a parameter (e.g., '`i1#Input_impedance`') and use the text box for editing to modify it (e.g., to '`1/i1#Input_impedance`'). We see that g_{ds} is indeed much smaller than g_m, so it is reasonable to approximate g_m by $g_m + g_{ds}$. If this had not been the case, we might have plotted the reciprocal of the input impedance minus the reciprocal of the output impedance instead.

We may specify the value of '`VGD`' to be different from 0. In Fig. 3.37, a value of '`VGD`' larger than V_t will bias the transistor in the triode region while a negative value will increase V_{DS} and bias the transistor deeper into the active region. However, in this circuit configuration, the drain bias voltage depends on both the transistor layout parameters and on the drain bias current.



(a) (b)

Figure 3.39: Diode-connected NMOS transistor with a voltage buffer for the drain voltage (a) and small-signal diagram for the diode-connected transistor (b).

In order to obtain a constant drain voltage, the diode-connected transistor may be modified as shown in Fig. 3.39. With the gain A of the voltage-controlled voltage source $\gg 1$, the bias value of the drain voltage is (almost) equal to the dc bias voltage V_B used as input to the voltage-controlled voltage source. A small-signal analysis gives the small-signal input impedance $v_{gs}/i_d = (g_m + g_{ds}/(1+A))^{-1} \simeq 1/g_m$ for $Ag_m \gg g_{ds}$. Figure 3.40 shows the LTspice schematic corresponding to the schematic from Fig. 3.37 augmented with the voltage buffer shown in Fig. 3.39.

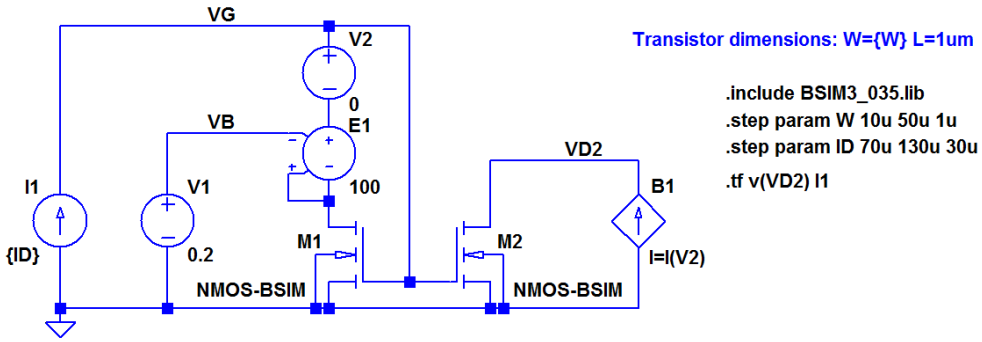
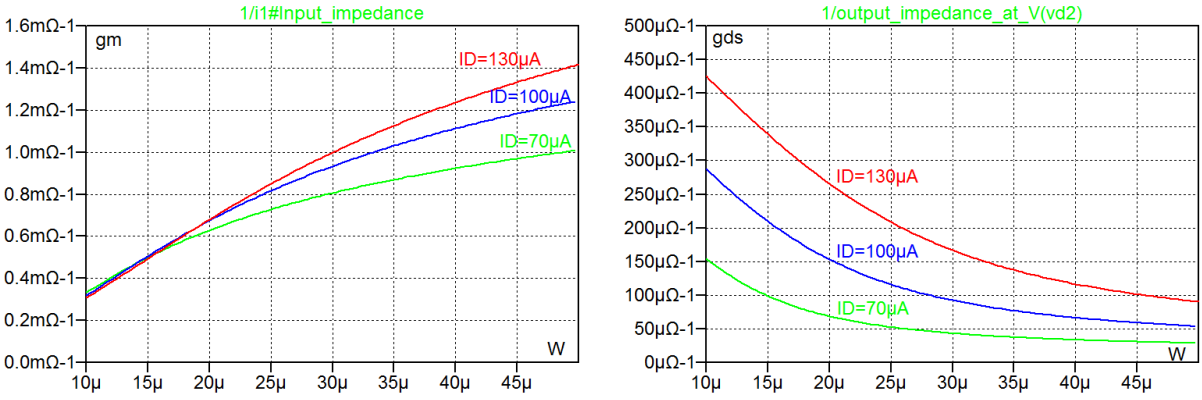


Figure 3.40: NMOS current mirror with voltage buffer for the drain voltage for simulating both g_{ds} and g_m.



(a) (b)

Figure 3.41: Simulation of g_m (a) and g_{ds} (b) versus W for different levels of bias current with the transistor in the triode region.

As an example, Fig. 3.41 shows a simulation of g_m and g_{ds} with a small value of V_B ($V_B = 0.2$ V) for which the transistors are in the triode region. Clearly, in this situation g_{ds} cannot be neglected compared to g_m, but for the configuration with the voltage buffer, we find g_m directly as the reciprocal of the input impedance (for $A \gg 1$).

Example 3.8: Simulating small-signal transistor capacitances using the '.ac' simulation.

While the small-signal parameters g_m and g_{ds} are easily simulated using the '.tf' simulation, the small-signal capacitances cannot be simulated by this command. They are listed in the error log resulting from a '.op' simulation as shown in Fig. 3.9, but in order to obtain the capacitances as functions of design parameters such as bias current or bias voltages, a '.ac' simulation may be used.

Following the approach described in Tutorial 2, Example 2.6, the capacitances can be found from $C = \mathrm{Im}(I)/(\omega V) = \mathrm{Im}(I)/(2\pi f V)$ where V is the voltage applied to the capacitor and I is the current in the capacitor, compare Eq. (2.17).

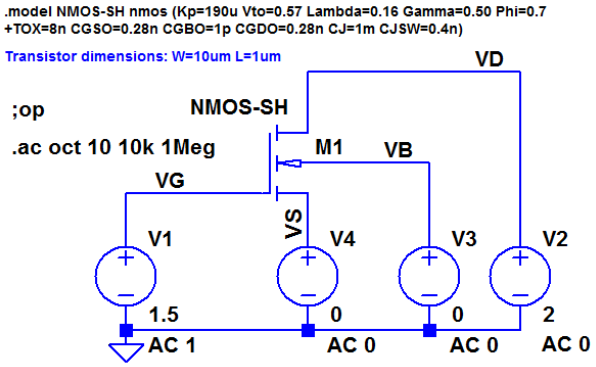


Figure 3.42: LTspice schematic for simulating small-signal transistor capacitances for an NMOS transistor.

Figure 3.42 shows an LTspice schematic for simulating the small-signal capacitances for an NMOS transistor with the same Shichman-Hodges model and the same dimensions and bias conditions as the NMOS transistor shown in Fig. 3.8. A '.op' simulation results in the capacitance values listed in the log file shown in Fig. 3.9. When applying an ac voltage with an amplitude of 1 to the gate, we can find the gate-source capacitance from the current in the source terminal, i.e., the current in '`V4`'. Also, we can find the gate-drain capacitance from the current in '`V2`'. Figure 3.43 shows the simulation plot (with a linear y-axis), and we find a gate-source capacitance of 31.6 fF and a gate-drain capacitance of 2.8 fF. This may be compared to the values from Fig. 3.9 where we have C_{gs} = `Cgs + Cgsov` = 31.6 fF and C_{gd} = `Cgdov` = 2.8 fF.

For finding the bulk-drain capacitance we may apply an ac amplitude of 1 to drain, gate and source ('`V2`', '`V1`' and '`V4`') while the bulk ('`V3`') is reset to 0 V. The bulk-drain capacitance is then found from the current in '`V2`'. The simulation plot is shown in Fig. 3.44, and we find a bulk-drain capacitance of 7.91 fF, i.e., the same as found from the '.op' simulation.

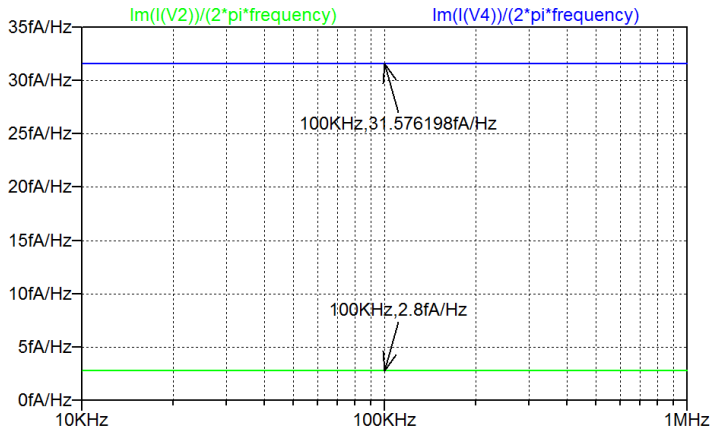


Figure 3.43: Simulation of C_{gs} and C_{gd}.

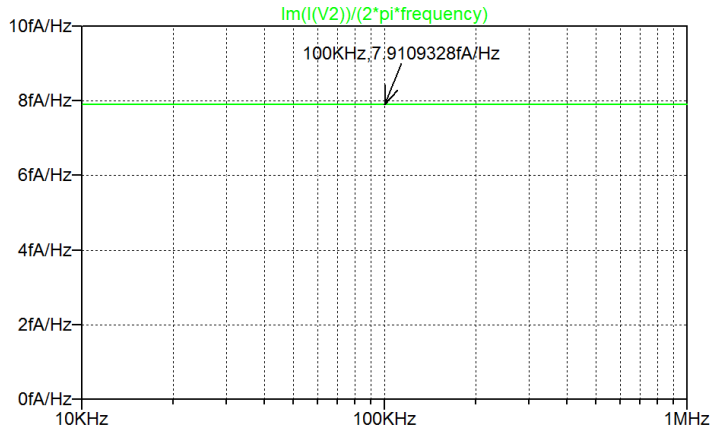


Figure 3.44: Simulation of C_{bd}.

The bulk-drain capacitance is a junction capacitance, so it is dependent on the reverse bias voltage of the junction, i.e., V_{DB}. In order to investigate this, we define the dc value of '`V2`' as a parameter '`{VDB}`' and introduce a '.step' directive in schematic in Fig. 3.42: '`.step param VDB 0 3 0.3`'. This causes V_{DB} to be stepped from 0 V to 3 V in increments of 0.3 V, so the resulting plot of the bulk-drain capacitance

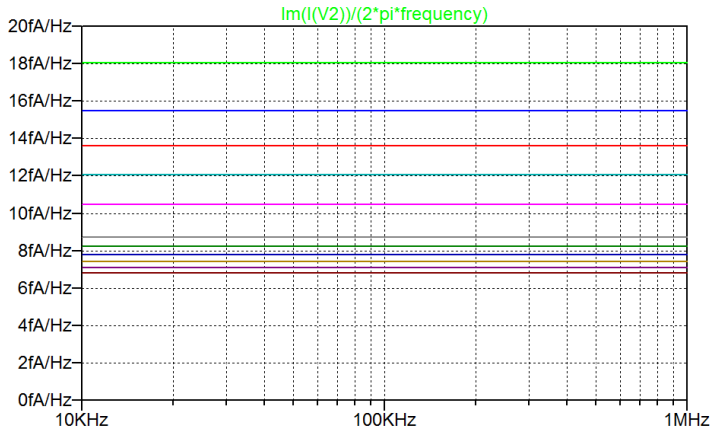


Figure 3.45: Simulation of C_{bd} for different values of the bias voltage V_{DB}.

has 11 curves as shown in Fig. 3.45. We see that the bulk-drain capacitance varies from 18.0 fF to 6.8 fF when the bias voltage varies from 0 V to 3 V.

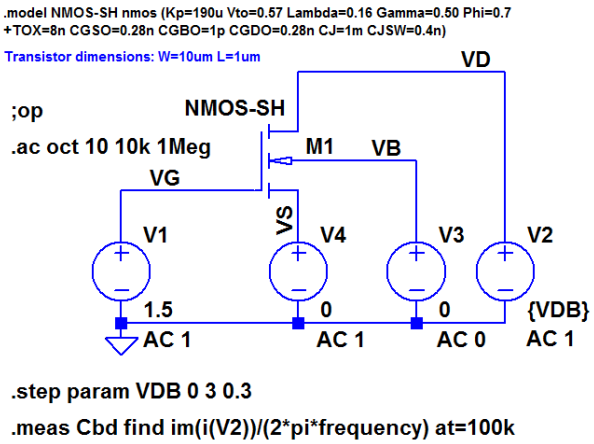


Figure 3.46: LTspice schematic for simulating small-signal transistor capacitances for an NMOS transistor, including a '.meas' directive for calculating C_{bd}.

Using a '.meas' directive, the capacitance may be calculated and plotted versus the bias voltage: Insert a '.meas' directive as shown in Fig. 3.46 and run the '.ac' simulation. Then open the error log file ('Ctrl-L'). In this file, the results of the '.meas' directive are given. When right-clicking in the error log file, a small dialogue box opens and you can select 'Plot .step'ed .meas data' which opens a window in the waveform viewer showing C_{bd} as a function of the bias voltage V_{DB}. Figure 3.47 shows this plot where the (left) y-axis has been selected to be linear and the phase is not shown. The variation of C_{bd} versus V_{DB} is evident.

In a similar way, the input capacitance to a transistor may be simulated. This is left as an exercise to the reader, see Problem 3.8.

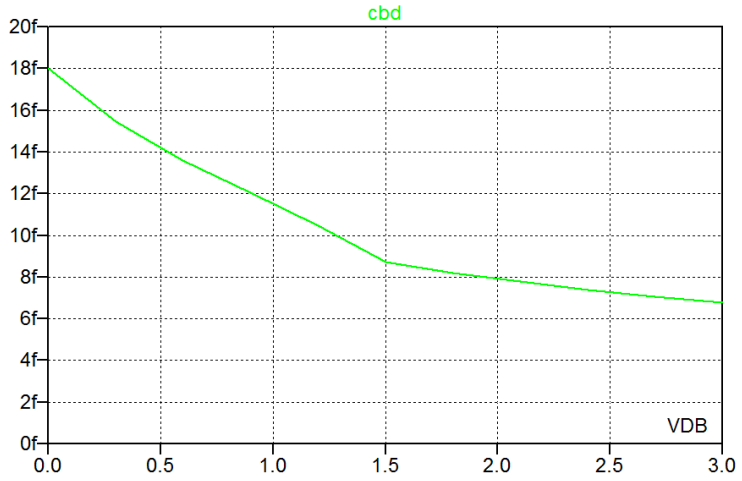


Figure 3.47: Simulation of C_{bd} versus the bias voltage V_{DB}.

Hints and pitfalls

- LTspice defines the positive direction of current flow *into* the transistor, so the drain current in a PMOS transistor is normally negative (or zero).
- The output file from an operating point simulation ('.op') provides information about node voltages and device currents.
- The SPICE Error Log (hotkey 'Ctrl-L') from an operating point simulation ('.op') provides information about bias points and small-signal parameters for the transistors in a circuit.
- Simple transistor models may be entered directly in the schematic using a '.model' SPICE directive. Advanced transistor models are included using model files and a '.include' SPICE directive.
- A model file can be opened and viewed in LTspice using the command 'Files → Open' and specifying 'Files of type → All Files'.
- The default model names for 'nmos4' and 'pmos4' transistors are NMOS and PMOS. If your '.model' directives or model file use model names different from this, remember to change the model name when specifying the transistor parameters for each transistor (see Fig. 3.1). Otherwise, the simulation will run with a default Shichman-Hodges transistor model with '`Kp=2e-5`', '`Vto=0`', '`Lambda=0`', '`Gamma=0`', and '`Phi=0.6`'.
- Remember to re-calculate the channel-length modulation parameter '`Lambda`' in the Shichman-Hodges transistor model when changing the channel length of a transistor. It is inversely proportional to the channel length.
- When using the Shichman-Hodges model, separate transistor models are required for transistors with different channel lengths in order to specify different values for '`Lambda`'.
- When having multiple traces in a simulation plot (e.g., output characteristics for different values of V_{GS}), one or two cursors may be attached to the traces and moved from one trace to another by the up-arrow key and the down-arrow key.
- The information about a trace followed by a cursor is displayed by right-clicking on the cursor number.
- When using two '.step' directives in a schematic, the x-axis in the simulation plot is determined by the '.step' directive appearing first in the netlist. This is the command inserted first in the schematic.
- When using '.measure' directives in combination with '.step' directives, the resulting tables in the error log file can be presented in the waveform viewer by using a right-click on the mouse and selecting 'Plot .step'ed .meas data'.

References

Allen, PE., & Holberg, DR. 2012, *CMOS Analog Integrated Circuit Design*, Third Edition, Oxford University Press, New York, USA.

Baker, RJ. 2010, *CMOS Circuit Design, Layout, and Simulation*, Third Edition, IEEE Press, Piscataway, USA.

Cao, Y. 2011, *Predictive Technology Model for Robust Nanoelectronic Design*, Springer, New York, USA.

Chan Carusone, T., Johns, D. & Martin, K. 2012, *Analog Integrated Circuit Design*, Second Edition, John Wiley & Sons, Inc., Hoboken, USA.

Chan Carusone, T., Johns, D. & Martin, K. 2014, *Analog Integrated Circuit Design, Netlist and model files*. Retrieved from http://analogicdesign.com/students/netlists-models/

Enz, CC. & Vittoz, EA. 2006, *Charge-Based MOS Transistor Modeling: The EKV Model for Low-Power and RF Design*, John Wiley & Sons, Ltd., Chichester, UK.

Predictive Technology Model Website 2012. Retrieved from http://ptm.asu.edu/

Sedra, AS. & Smith, KC. 2016, *Microelectronic Circuits*, International Seventh Edition, Oxford University Press, New York, USA.

Sheu, BJ., Sharfetter, DL., Ko, P-K. & Jeng, M-C. 1987, 'BSIM: Berkeley Short-Channel IGFET Model for MOS Transistors', *IEEE J. Solid-State Circuit*, vol. SC-22, No. 4, pp. 558-566.

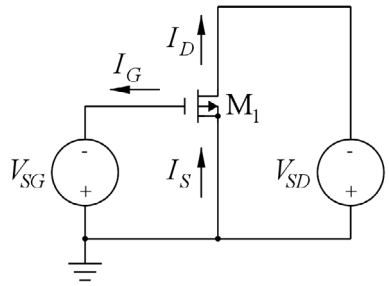
Shichman, H. & Hodges, DA. 1968, 'Modeling and Simulation of Insulated-Gate Field-Effect Transistor Switching Circuits', *IEEE J. Solid-State Circuit*, vol. SC-3, No. 3, pp. 285-289.

The MOSIS Service 2019. Retrieved from http://www.mosis.com/

Tsividis, Y. & McAndrew, C. 2010, *Operation and Modeling of the MOS Transistor*, Third Edition, Oxford University Press, New York, USA.

Problems

Problem 3.1



$W = 10$ µm, $L = 1$ µm,
$K_p = 55$ µA/V^2, $V_{to} = -0.71$ V, $\lambda = 0.16$ V^{-1},
$\gamma = 0.75\ \sqrt{\text{V}}$, $|2\Phi_F| = 0.7$ V.

For the PMOS transistor shown above, simulate and plot the input characteristics I_D versus V_{SG} and $\partial i_D/\partial v_{SG}$ for $V_{SD} = 0$, 0.5, 1.0, 1.5, 2.0, 2.5 and 3.0 V. Use the model parameters and transistor dimensions shown above. Find the bias current I_D and the small-signal parameters g_m, g_{mb} and g_{ds} for the bias point of $V_{SG} = 1.5$ V and $V_{SD} = 2.0$ V.

Problem 3.2

```
.MODEL NMOS-BSIM NMOS LEVEL = 49
+VERSION = 3.1 TNOM = 27 TOX = 7.8E-9
+XJ = 1E-07 NCH = 2.18E+17 VTH0 = 0.48
+K1 = 6.07E-01 K2 = 1.24E-03 K3 = 9.68E+01
+K3B = -9.84E+00 W0 = 2.02E-05 NLX = 1.62E-07
+DVT0W = 0 DVT1W = 0 DVT2W = 0
+DVT0 = 2.87E+00 DVT1 = 5.86E-01 DVT2 = -1.26E-01
+U0 = 360 UA = -8.48E-10 UB = 2.27E-18
+UC = 3.27E-11 VSAT = 1.87E+05 A0 = 1.22E+00
+AGS = 2.06E-01 B0 = 9.60E-07 B1 = 4.95E-06
+KETA = -1.67E-04 A1 = 0 A2 = 3.49E-01
+RDSW = 8.18E+02 PRWG = 2.35E-02 PRWB = -8.12E-02
+WR = 9.98E-01 WINT = 1.55E-07 LINT = 4.51E-10
+DWG = -4.27E-09
+DWB = 4.07E-09 VOFF = -4.14E-02 NFACTOR = 1.61E+00
+CIT = 0 CDSC = 2.39E-04 CDSCD = 0.00E+00
+CDSCB = 0 ETA0 = 1 ETAB = -1.99E-01
+DSUB = 1 PCLM = 1.32E+00 PDIBLC1 = 2.42E-04
+PDIBLC2 = 8.27E-03 PDIBLCB = -9.99E-04 DROUT = 9 72E-04
+PSCBE1 = 7.24E+08 PSCBE2 = 9.96E-04 PVAG = 1.00E-02
+DELTA = 1.01E-02 RSH = 3.33E+00 MOBMOD = 1
+PRT = 0 UTE = -1.5 KT1 = -1.11E-01
+KT1L = 0 KT2 = 2.22E-02 UA1 = 4.34E-09
+UB1 = -7.56E-18 UC1 = -5.62E-11 AT = 3.31E+04
+WL = 0 WLN = 9.95E-01 WW = 0
+WWN = 1.00E+00 WWL = 0 LL = 0
+LLN = 1 LW = 0 LWN = 1
+LWL = 0 CAPMOD = 2 XPART = 0.5
+CGDO = 2 76E-10 CGSO = 2.76E-10 CGBO = 1.00E-12
+CJ = 9e-4 PB = 7.95E-01 MJ = 3.53E-01
+CJSW = 2.8e-10 PBSW = 7.98E-01 MJSW = 1.73E-01
+CJSWG = 1.81E-10 PBSWG = 7.96E-01 MJSWG = 1.74E-01
+CF = 0 PVTH0 = -1.80E-02 PRDSW = -7.56E+01
+PK2 = 4.48E-05 WKETA = -1.33E-03 LKETA = -8.91E-03
```

For an NMOS transistor with the transistor model shown above (BSIM3 0.35 µm model, Fig. 3.10) and channel width $W = 10$ µm, simulate and plot I_D versus the channel length L in the interval 1 µm $< L <$ 10 µm for a bias point of $V_{GS} = 1.5$ V, $V_{DS} = 2.0$ V and $V_{SB} = 0$ V. Find the bias current I_D and the small-signal parameters g_m, g_{mb} and g_{ds} for $L = 1$ µm and for $L = 5$ µm in the bias point.
Hint: Define L as a parameter and step L from 1 µm to 10 µm.

Problem 3.3

```
.MODEL PMOS-BSIM PMOS LEVEL = 49
+VERSION = 3.1 TNOM = 2.69E+01 TOX = 7.8E-9
+XJ = 1.00E-07 NCH = 8.44E+16 VTH0 = -0.6
+K1 = 4.82E-01 K2 = -2.13E-02 K3 = 8.27E+01
+K3B = -5 W0 = 5.24E-06 NLX = 2.49E-07
+DVT0W = 0.00E+00 DVT1W = 0 DVT2W = 0
+DVT0 = 3.54E-01 DVT1 = 7.52E-01 DVT2 = -2.98E-01
+U0 = 150 UA = 1E-10 UB = 1.75E-18
+UC = -2.27E-11 VSAT = 2.01E+05 A0 = 1.04E+00
+AGS = 2.90E-01 B0 = 1.94E-06 B1 = 5.01E-06
+KETA = -3.85E-03 A1 = 4.20E-03 A2 = 1.00E+00
+RDSW = 4000 PRWG = -9.54E-02 PRWB = -1.92E-03
+WR = 1 WINT = 1.47E-07 LINT = 1.04E-10
+DWG = -1.09E-08
+DWB = 1.14E-08 VOFF = -1.29E-01 NFACTOR = 2.01E+00
+CIT = 0 CDSC = 2.40E-04 CDSCD = 0
+CDSCB = 0 ETA0 = 4.07E-02 ETAB = 6.84E-03
+DSUB = 3.21E-01 PCLM = 5.96E+00 PDIBLC1 = 2.89E-03
+PDIBLC2 = -1.45E-06 PDIBLCB = -1E-03 DROUT = 9.93E-04
+PSCBE1 = 7.88E+10 PSCBE2 = 5E-10 PVAG = 15
+DELTA = 9.96E-03 RSH = 2.6 MOBMOD = 1
+PRT = 0 UTE = -1.5 KT1 = -1.09E-01
+KT1L = 0 KT2 = 2.19E-02 UA1 = 4.34E-09
+UB1 = -7.62E-18 UC1 = -5.63E-11 AT = 3.28E+04
+WL = 0 WLN = 1 WW = 0
+WWN = 1.00E+00 WWL = 0 LL = 0
+LLN = 1 LW = 0 LWN = 1
+LWL = 0 CAPMOD = 2.01E+00 XPART = 0.5
+CGDO = 2 10E-10 CGSO = 2.12E-10 CGBO = 1.00E-12
+CJ = 14e-4 PB = 9.83E-01 MJ = 5.79E-01
+CJSW = 3.2e-10 PBSW = 9.92E-01 MJSW = 3.60E-01
+CJSWG = 4.41E-11 PBSWG = 9.85E-01 MJSWG = 3.58E-01
+CF = 0 PVTH0 = 2.58E-02 PRDSW = -3.98E+01
+PK2 = 2.02E-03 WKETA = 2.72E-03 LKETA = -7.14E-03
```

For a PMOS transistor with the transistor model shown above (BSIM3 0.35 μm model, Fig. 3.10) and channel width $W = 10$ μm and channel length $L = 1$ μm, simulate and plot the input characteristics I_D versus V_{SG} for $V_{SD} = 0$, 0.5, 1.0, 1.5, 2.0, 2.5 and 3.0 V. Assume $V_{BS} = 0$ V. Also simulate and plot the output characteristics I_D versus V_{SD} for $V_{SG} = 0$, 0.5, 1.0, 1.5, 2.0, 2.5 and 3.0 V. Use the cursors to find I_D and $\partial i_D/\partial v_{SD}$ for $V_{SG} = 1.5$ V, $V_{BS} = 0$ V and $V_{SD} = 2.0$ V.

Problem 3.4

```
.MODEL PMOS-BSIM PMOS LEVEL = 49
+VERSION = 3.1 TNOM = 2.69E+01 TOX = 7.8E-9
+XJ = 1.00E-07 NCH = 8.44E+16 VTH0 = -0.6
+K1 = 4.82E-01 K2 = -2.13E-02 K3 = 8.27E+01
+K3B = -5 W0 = 5.24E-06 NLX = 2.49E-07
+DVT0W = 0.00E+00 DVT1W = 0 DVT2W = 0
+DVT0 = 3.54E-01 DVT1 = 7.52E-01 DVT2 = -2.98E-01
+U0 = 150 UA = 1E-10 UB = 1.75E-18
+UC = -2.27E-11 VSAT = 2.01E+05 A0 = 1.04E+00
+AGS = 2.90E-01 B0 = 1.94E-06 B1 = 5.01E-06
+KETA = -3.85E-03 A1 = 4.20E-03 A2 = 1.00E+00
+RDSW = 4000 PRWG = -9.54E-02 PRWB = -1.92E-03
+WR = 1 WINT = 1.47E-07 LINT = 1.04E-10
+DWG = -1.09E-08
+DWB = 1.14E-08 VOFF = -1.29E-01 NFACTOR = 2.01E+00
+CIT = 0 CDSC = 2.40E-04 CDSCD = 0
+CDSCB = 0 ETA0 = 4.07E-02 ETAB = 6.84E-03
+DSUB = 3.21E-01 PCLM = 5.96E+00 PDIBLC1 = 2.89E-03
+PDIBLC2 = -1.45E-06 PDIBLCB = -1E-03 DROUT = 9.93E-04
+PSCBE1 = 7.88E+10 PSCBE2 = 5E-10 PVAG = 15
+DELTA = 9.96E-03 RSH = 2.6 MOBMOD = 1
+PRT = 0 UTE = -1.5 KT1 = -1.09E-01
+KT1L = 0 KT2 = 2.19E-02 UA1 = 4.34E-09
+UB1 = -7.62E-18 UC1 = -5.63E-11 AT = 3.28E+04
+WL = 0 WLN = 1 WW = 0
+WWN = 1.00E+00 WWL = 0 LL = 0
+LLN = 1 LW = 0 LWN = 1
+LWL = 0 CAPMOD = 2.01E+00 XPART = 0.5
+CGDO = 2 10E-10 CGSO = 2.12E-10 CGBO = 1.00E-12
+CJ = 14e-4 PB = 9.83E-01 MJ = 5.79E-01
+CJSW = 3.2e-10 PBSW = 9.92E-01 MJSW = 3.60E-01
+CJSWG = 4.41E-11 PBSWG = 9.85E-01 MJSWG = 3.58E-01
+CF = 0 PVTH0 = 2.58E-02 PRDSW = -3.98E+01
+PK2 = 2.02E-03 WKETA = 2.72E-03 LKETA = -7.14E-03
```

For a PMOS transistor with the transistor model shown above (BSIM3 0.35 µm model, Fig. 3.10) and channel width $W = 10$ µm and channel length $L = 1$ µm, find the bias current I_D and the small-signal parameters g_m, g_{mb} and g_{ds} in a bias point of $V_{SG} = 1.5$ V, $V_{BS} = 0$ V and $V_{SD} = 2.0$ V. From these small-signal parameters and the bias current, estimate parameters for a Shichman-Hodges model for the transistor. Assume $|2\Phi_F| = 0.7$ V.

Simulate and plot the input characteristics (I_D versus V_{SG}) and output characteristics (I_D versus V_{SD}) using both the BSIM model and the Shichman-Hodges model with the parameters estimated from the simulation of small-signal parameters in the bias point.

Problem 3.5

```
.MODEL NMOS-BSIM NMOS LEVEL = 49
+VERSION = 3.1 TNOM = 27 TOX = 7.8E-9
+XJ = 1E-07 NCH = 2.18E+17 VTH0 = 0.48
+K1 = 6.07E-01 K2 = 1.24E-03 K3 = 9.68E+01
+K3B = -9.84E+00 W0 = 2.02E-05 NLX = 1.62E-07
+DVT0W = 0 DVT1W = 0 DVT2W = 0
+DVT0 = 2.87E+00 DVT1 = 5.86E-01 DVT2 = -1.26E-01
+U0 = 360 UA = -8.48E-10 UB = 2.27E-18
+UC = 3.27E-11 VSAT = 1.87E+05 A0 = 1.22E+00
+AGS = 2.06E-01 B0 = 9.60E-07 B1 = 4.95E-06
+KETA = -1.67E-04 A1 = 0 A2 = 3.49E-01
+RDSW = 8.18E+02 PRWG = 2.35E-02 PRWB = -8.12E-02
+WR = 9.98E-01 WINT = 1.55E-07 LINT = 4.51E-10
+DWG = -4.27E-09
+DWB = 4.07E-09 VOFF = -4.14E-02 NFACTOR = 1.61E+00
+CIT = 0 CDSC = 2.39E-04 CDSCD = 0.00E+00
+CDSCB = 0 ETA0 = 1 ETAB = -1.99E-01
+DSUB = 1 PCLM = 1.32E+00 PDIBLC1 = 2.42E-04
+PDIBLC2 = 8.27E-03 PDIBLCB = -9.99E-04 DROUT = 9.72E-04
+PSCBE1 = 7.24E+08 PSCBE2 = 9.96E-04 PVAG = 1.00E-02
+DELTA = 1.01E-02 RSH = 3.33E+00 MOBMOD = 1
+PRT = 0 UTE = -1.5 KT1 = -1.11E-01
+KT1L = 0 KT2 = 2.22E-02 UA1 = 4.34E-09
+UB1 = -7.56E-18 UC1 = -5.62E-11 AT = 3.31E+04
+WL = 0 WLN = 9.95E-01 WW = 0
+WWN = 1.00E+00 WWL = 0 LL = 0
+LLN = 1 LW = 0 LWN = 1
+LWL = 0 CAPMOD = 2 XPART = 0.5
+CGDO = 2.76E-10 CGSO = 2.76E-10 CGBO = 1.00E-12
+CJ = 9e-4 PB = 7.95E-01 MJ = 3.53E-01
+CJSW = 2.8e-10 PBSW = 7.98E-01 MJSW = 1.73E-01
+CJSWG = 1.81E-10 PBSWG = 7.96E-01 MJSWG = 1.74E-01
+CF = 0 PVTH0 = -1.80E-02 PRDSW = -7.56E+01
+PK2 = 4.48E-05 WKETA = -1.33E-03 LKETA = -8.91E-03
```

For an NMOS transistor with the transistor model shown above (BSIM3 0.35 µm model, Fig. 3.10), a channel width $W = 10$ µm and channel length $L = 1$ µm, assume a bias point specified by $V_{GS} = V_{DS}$, $V_{SB} = 0$ V and $I_D = 140$ µA. Find g_m, g_{mb} and g_{ds} from a '.op' simulation and estimate parameters K_p, V_{to}, λ and γ for a Shichman-Hodges model for the transistor. Assume $|2\Phi_F| = 0.7$ V.

Problem 3.6

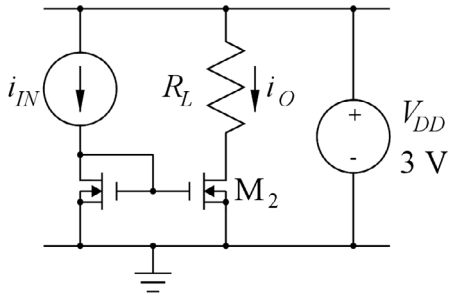
An NMOS transistor is assumed to have the following Shichman-Hodges parameters: $K_p = 190$ µA/V^2, $V_{to} = 0.57$ V, $\lambda = 0.16$ V^{-1}, $\gamma = 0.5\ \sqrt{\text{V}}$ and $|2\Phi_F| = 0.7$ V. The channel length is $L = 1$ µm. Simulate and plot g_m and g_{ds} versus the drain current I_D for $W = 10$ µm, $W = 30$ µm and $W = 50$ µm, and $0 < I_D < 10$ mA. Assume a drain-source voltage of $V_{DS} = 1.2$ V.

From the plots of g_m and g_{ds}, find the maximum drain current for which the transistor is in the active region for each of the three values of channel width.

Problem 3.7

An NMOS transistor is assumed to have the following Shichman-Hodges parameters: $K_p = 190\ \mu A/V^2$, $V_{to} = 0.57$ V, $\lambda = 0.16\ V^{-1}$, $\gamma = 0.5\ \sqrt{V}$ and $|2\Phi_F| = 0.7$ V. The channel length is $L = 1\ \mu m$ and the channel width is $W = 20\ \mu m$. Simulate and plot g_m/g_{ds} versus the drain current I_D for $1\ \mu A < I_D < 100\ \mu A$. Assume a drain-source voltage of $V_{DS} = 1.5$ V.
Also plot g_m/g_{ds} versus $1/\sqrt{I_D}$. Find g_m/g_{ds} for $I_D = 1\ \mu A$ and $I_D = 100\ \mu A$.

Problem 3.8



$W_1 = W_2 = 20\ \mu m$, $L_1 = L_2 = 1\ \mu m$,
$AD_1 = AD_2 = 20 \times 10^{-12}\ m^2$
$PD_1 = PD_2 = 22\ \mu m$.

For the current mirror shown above, use a '.ac' simulation to find the input resistance and the input capacitance. Assume that the input impedance can be approximated by a parallel connection of a capacitance and a resistance so that Eqs. (2.22) and (2.23) can be used. Assume a dc value of 100 µA for the input current and use the transistor dimensions shown above. Also, use the BSIM3 transistor model from Problem 3.5 (BSIM3 0.35 µm model, Fig. 3.10).
Find the input resistance and input capacitance for two different values of the load resistor: $R_L = 0\ \Omega$ and $R_L = 10\ k\Omega$.

Answers

3.1: $I_D = 227$ µA; $g_m = 0.574$ mA/V; $g_{mb} = 0.257$ mA/V; $g_{ds} = 27.5$ µA/V

3.2: $L = 1$ µm: $I_D = 0.548$ mA; $g_m = 0.992$ mA/V; $g_{mb} = 0.263$ mA/V; $g_{ds} = 10.3$ µA/V.
$L = 5$ µm: $I_D = 0.127$ mA; $g_m = 0.245$ mA/V; $g_{mb} = 0.065$ mA/V; $g_{ds} = 1.94$ µA/V.

3.3: $I_D = 160$ µA; $\partial i_D / \partial v_{SD} = 7.32$ µA/V.

3.4: $I_D = 160$ µA; $g_m = 339$ µA/V; $g_{mb} = 75.4$ µA/V; $g_{ds} = 7.31$ µA/V;
$\lambda = 0.05$ V^{-1}; $V_{to} = -0.556$ V; $K_p = 32.6$ µA/V^2; $\gamma = 0.37$ $\sqrt{\text{V}}$; $|2\Phi_F| = 0.7$ V.

3.5: $g_m = 0.584$ mA/V; $g_{mb} = 0.167$ mA/V; $g_{ds} = 6.04$ µA/V;
$\lambda = 0.045$ V^{-1}; $V_{to} = 0.520$ V; $K_p = 117$ µA/V^2; $\gamma = 0.48$ $\sqrt{\text{V}}$; $|2\Phi_F| = 0.7$ V.

3.6: $W = 10$ µm: $I_{D\max} = 1.65$ mA; $W = 30$ µm: $I_{D\max} = 4.95$ mA; $W = 50$ µm: $I_{D\max} = 8.23$ mA.

3.7: $I_D = 1$ µA: $g_m/g_{ds} = 751$; $I_D = 100$ µA: $g_m/g_{ds} = 75.1$.

3.8: $R_L = 0\ \Omega$: $R_{in} = 1.206$ kΩ; $C_{in} = 0.146$ pF; $R_L = 10$ kΩ: $R_{in} = 1.206$ kΩ; $C_{in} = 0.190$ pF.

Tutorial 4 – Basic Gain Stages

This tutorial introduces the basic CMOS gain stages and some of the issues arising when simulating the stages. The basic gain stages include the common-source stage, the common-drain stage, the common-gate stage and the differential input pair. After having completed the tutorial, you should be able to

- find bias currents and voltages for the standard configurations of basic gain stages.
- simulate the low-frequency transfer function and the signal swings on the input and output of a gain stage.
- find small-signal parameters for the transistors in a gain stage.
- simulate the frequency response of a gain stage.
- perform design iterations from simple Shichman-Hodges transistor models to advanced Spice models.
- simulate common-mode rejection ratio and power-supply rejection ratio of a gain stage.
- simulate input impedances and output impedances of a gain stage.
- simulate the noise properties of a gain stage.

Example 4.1: The common-source amplifier (inverting amplifier).

The simplest form of a common-source stage is just an NMOS transistor with a resistor to provide the bias current as shown in Fig. 4.1. This configuration is rarely used in integrated circuit design, but it provides a good introduction to the common-source stage and to the steps in design iterations involving different transistor models. Hence, we will start by analyzing this configuration and subsequently examine a common-source configuration with an active load.

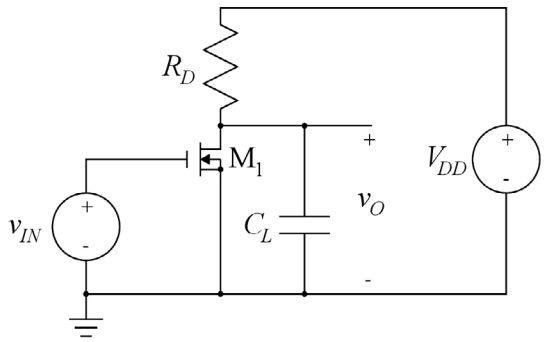


Figure 4.1: NMOS common-source amplifier with drain resistor.

Common-source stage with a drain resistor: The design specification for such an amplifier stage may comprise specifications of small-signal gain, output resistance, supply voltage, output-voltage range, input-voltage range, supply current, frequency response, etc. For this example, we assume that the supply voltage is specified to be $V_{DD} = 3$ V and that the open-circuit small-signal gain should be $A_{voc} = -10$ V/V, corresponding to 20 dB. The quiescent value of the output voltage (with no dc load connected to the output) should be $V_{DD}/2 = 1.5$ V in order to allow a large voltage swing at the output. Also, let us assume that the -3 dB frequency f_0 should be at least 10 MHz with a load capacitance of 1.5 pF and that the current consumption should be as small as possible. The design parameters for this stage are the value of R_D and the transistor dimensions W and L. In order to have a starting point for the simulation of the stage, we will calculate values for these parameters using the simple Shichman-Hodges transistor model (Shichman & Hodges 1968). We are assuming a 0.35 µm CMOS process and use the transistor parameters from Fig. 3.8. For initial calculations by hand, it may be acceptable to ignore the channel-length modulation (i.e., assume $\lambda = 0$), and with a load capacitance of 1.5 pF, it is also reasonable to neglect the internal transistor capacitances.

The design equations corresponding to the design requirements are as follows:

Gain requirements:

$$A_{voc} = -R_D g_m = -R_D \frac{2 I_D}{V_{GS} - V_{to}} \tag{4.1}$$

With the bias current I_D and the resistor R_D selected to provide a bias value of the output voltage of half the supply voltage V_{DD}, we have $R_D I_D = V_{DD}/2$, resulting in

$$A_{voc} = -\frac{V_{DD}}{V_{GS} - V_{to}} \Rightarrow V_{GS} = V_{to} + \frac{V_{DD}}{|A_{voc}|} = 0.87 \text{ V} \tag{4.2}$$

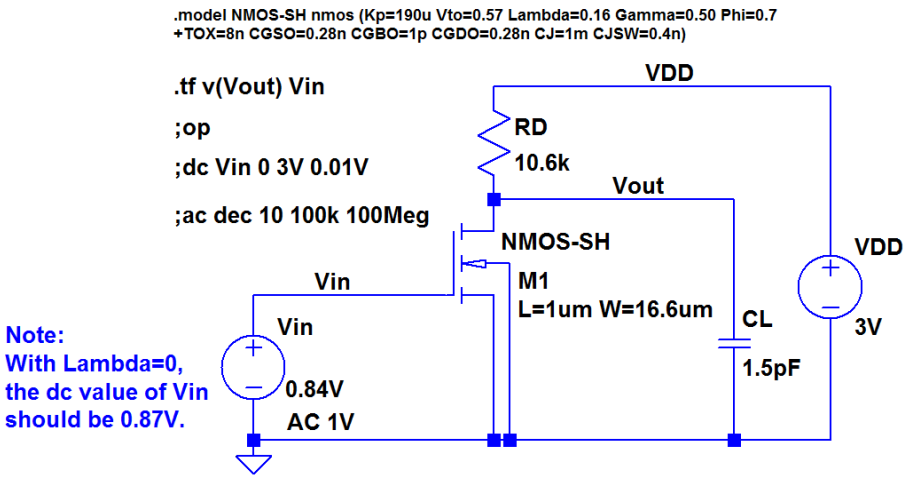


Figure 4.2: LTspice schematic for simulating the common-source stage with the Shichman-Hodges transistor model and with $\lambda = 0.16$ V^{-1} and the corresponding value of input bias voltage, i.e., a dc value of 0.84 V for V_{IN}.

Frequency response requirements: With a minimum requirement for f_0 and a requirement for small current consumption, R_D should be selected as large as possible while fulfilling $f_0 = 1/(2\pi R_D C_L) \geq$ 10 MHz. From this,

$$R_D = 1/(2\pi f_0 C_L) = 10.6\ \text{k}\Omega \tag{4.3}$$

With this value of R_D, the bias current I_D is $I_D = V_{DD}/(2R_D) = 142$ µA and the transistor transconductance is $g_m = |A_{voc}|/R_D = 0.94$ mA/V. The transistor dimensions can be calculated from

$$\frac{W}{L} = \frac{2\,I_D}{\mu_n C_{ox}(V_{GS} - V_{to})^2} = 16.6 \tag{4.4}$$

Selecting $L = 1$ µm, we find $W = 16.6$ µm.

These values are used in the following for simulating the circuit. Figure 4.2 shows the LTspice schematic for the circuit where also the transistor drain and source areas and perimeters have been specified using source and drain areas of 3 times W times the minimum length and perimeters of W plus 6 times the minimum length, i.e., slightly larger than the minimum sizes indicated in Example 3.1 (Sedra & Smith 2016, Appendix B).

Specifying transistor parameters: At this point, it may be useful to demonstrate how the transistor specifications can be shown in the schematic in a way that ensures back annotation from the schematic to the netlist, rather than the 'quick and dirty' way used throughout Tutorial 3. Consider Fig. 4.3 with three transistor symbols. The transistors have been specified to be identical by right-clicking on the transistor symbol and inserting the transistor parameters in the specification window as shown in Fig. 3.8.

For all three transistors, the following values have been entered: $L = 1$ µm, $W = 5$ µm, $AD = AS = 5 \times 10^{-12}$ m^2, $PD = PS = 7$ µm, $M = 1$. The topmost transistor (M_1) shows only the name ('`M1`') and the transistor model ('`NMOS-SH`'), and this is the LTspice default way of showing the transistor. When you 'Ctrl-right-click' on the transistor symbol, the 'Component Attribute Editor' shown to the left of the symbol opens. In this, you will see the model name ('`NMOS-SH`') listed as 'Value' and the transistor specification parameters listed as 'Value2'. Notice the column heading 'Vis.' (Visible). It has an X in the line for 'InstName' and 'Value' but not for 'Value2'. This specifies that only the name ('`M1`') and the transistor model ('`NMOS-SH`') are visible in the schematic.

The middle transistor (M_2) shows all the transistor parameters on the schematic. This is achieved by inserting an X in the 'Component Attribute Editor' for the line with 'Value2' as shown to the left of the transistor.

The bottom transistor (M_3) shows only L and W in addition to the name ('`M3`') and the transistor model ('`NMOS-SH`'). To the left of the transistor symbol is shown the 'Component Attribute Editor' for achieving this. The parameters which should be visible are remaining in the line 'Value2' which is still marked as visible, whereas the other parameters have been moved to the next line, 'SpiceLine', which is not marked as visible. Also, the letters for channel length and width have been changed to capital letters. The transistor specification in the netlist is the same for the three transistors ('View → SPICE Netlist').

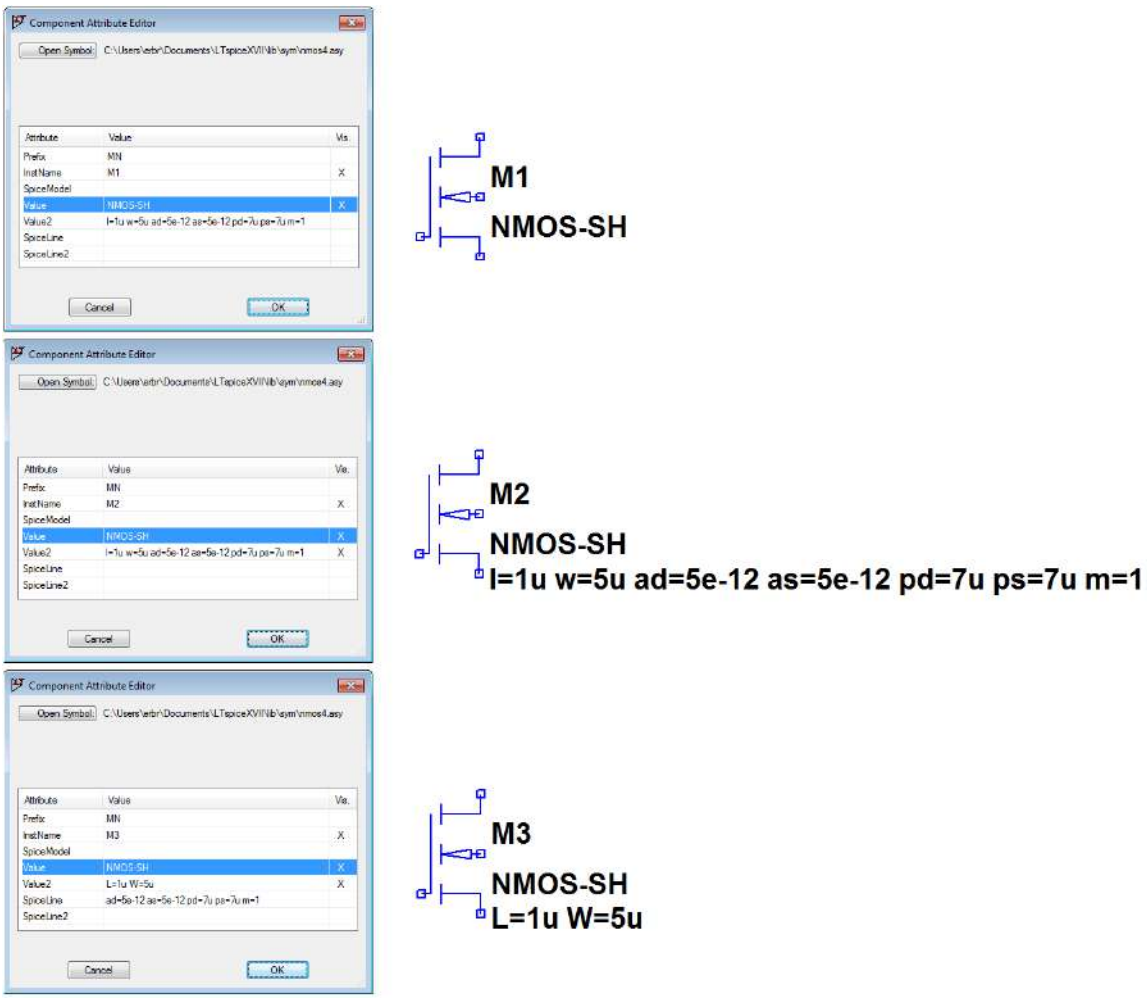


Figure 4.3: LTspice transistor symbols with different numbers of visible transistor parameters.

Often in a CMOS circuit, you would use the same channel length for all transistor, so it may not be necessary to show L in the schematic. If this is the case, L can just be moved to 'SpiceLine'. The order of the parameters has no influence on the simulation. Having only W visible saves some drawing space.

Actually, you may also use the 'Component Attribute Editor' for the transistor symbols 'nmos' and 'pmos' with direct connections between source and bulk. Rather than specifying the transistors as shown in Fig. 3.1, you 'Ctrl-right-click' on the symbol and enter the transistor parameters directly in the 'Component Attribute Editor' as shown in Fig. 4.3.

Clearly, this editing of the transistor is more involved than just the simple default specification. However, you can specify just one NMOS transistor and one PMOS transistor to show the parameters of interest, and then you can draw additional transistors using the duplicate command, 'Edit → Duplicate', 'F6', or toolbar symbol . An additional advantage is that the visible parameters can then be edited just by moving the cursor over the text and right-clicking.

Iterativ design of the transistor channel width: In order to verify the calculations from Eqs. (4.1) to (4.4), the circuit of Fig. 4.2 may first be simulated using $\lambda = 0$. Running a dc sweep, you will find that an input voltage of 0.87 V indeed results in an output voltage of 1.5 V, and from an ac analysis, you will find that the gain is indeed 20 dB with a -3 dB frequency of 10 MHz.

Changing λ to 0.16 V^{-1}, a re-simulation of the dc sweep shows that the input bias voltage must be changed to 0.84 V as shown in Fig. 4.2 in order to get an output voltage of 1.5 V. With this value of input bias voltage, a '.tf' simulation shows a low-frequency gain of -9.35 V/V and an output impedance of 8.87 kΩ. Obviously, the small-signal output resistance r_{ds} of the transistor has some influence. Running a '.op' simulation and analyzing the transistor small-signal parameters ('Ctrl-L' for viewing the error log), you find g_{ds} to be 18.4 μA/V, corresponding to $r_{ds} = 54$ kΩ. This is not quite negligible compared to R_D, and it will result in a smaller output resistance and a smaller gain. In order to obtain a larger gain, the value of R_D may be increased or the transistor width may be increased. Increasing R_D increases the output resistance and reduces the -3 dB frequency. Since λ is generally not a very well-controlled parameter, it is not advisable to increase R_D as this may cause the output resistance to be too high to fulfill the bandwidth requirement. Rather, the design may be modified by increasing the transistor width which (for the same bias current I_D) gives a larger value of g_m and a larger value of A_{voc}. From Eq. (3.8), we have

$$g_m = \sqrt{2\mu_n C_{ox} \frac{W}{L} I_D (1 + \lambda V_{DS})} \tag{4.5}$$

showing that g_m is proportional to the square root of W/L.

Thus, since g_m needs to be increased by about 7%, the value of W/L should be increased by about 14%, i.e., to about 19 μm. Again, a dc sweep is needed in order to find a new value for the input bias voltage. It is now $V_{GS} = 0.82$ V for an output voltage of 1.5 V. With this value of V_{GS}, an ac analysis results in a gain very close to 20 dB and a -3 dB frequency of 11.6 MHz which fulfills the specifications.

As we learned in Tutorial 3, there might be significant discrepancies between a simple Shichman-Hodges model and a more realistic, advanced transistor model. The circuit of Fig. 4.2 may be re-simulated using the BSIM3 transistor model from Fig. 3.10. Running a dc sweep with this model (and $W =$ 19 μm), you will find that the input bias voltage should be changed to 0.89 V in order to get an output voltage of 1.5V, and a '.tf' simulation results in a low-frequency gain of -9.05 V/V, i.e., about 9.5% too small. The bandwidth may be found from a '.ac' simulation to be 10.6 MHz. Hence, assuming that g_m follows Eq. (4.5), W should be increased by about 20%, giving a new value for W of 22.8 μm. With this value of W, the input bias voltage should be changed to 0.84 V, and a '.tf' simulation shows a low-frequency gain of -10.3 V/V while a '.ac' simulation shows a bandwidth of 10.5 MHz. Reducing W to 22 μm results in a gain of -10 V/V and a bandwidth of 10.6 MHz, so with just a few simple iterations, we have achieved an acceptable design.

Common-source stage with an active load: The common-source stage with resistive load is rarely used in integrated circuit design, partly because resistors typically take up more silicon area than transistors, partly because the gain is limited by the value of R_D. Instead, a standard configuration is the common-source stage with an active load as shown in Fig. 4.4.

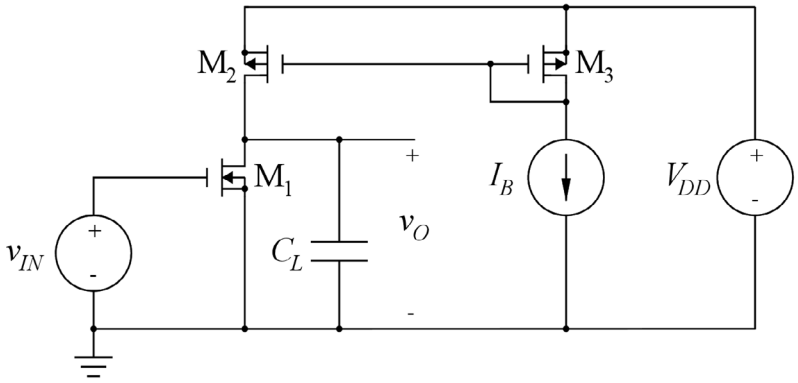


Figure 4.4: NMOS common-source amplifier with PMOS load.

Here, a PMOS current mirror implements the load to the NMOS common-source transistor M_1. In this way, the bias current to the common-source transistor is controlled by the current mirror, and the small-signal load to the common-source transistor is the parallel combination of r_{ds} for the NMOS common-source transistor and r_{ds} for the PMOS active load operating as a current source.

Observe that the PMOS transistors have their source upwards and drain downwards. This means that when drawing the schematic in LTspice, you need to rotate ('Ctrl-R') and mirror ('Ctrl-E') the transistor

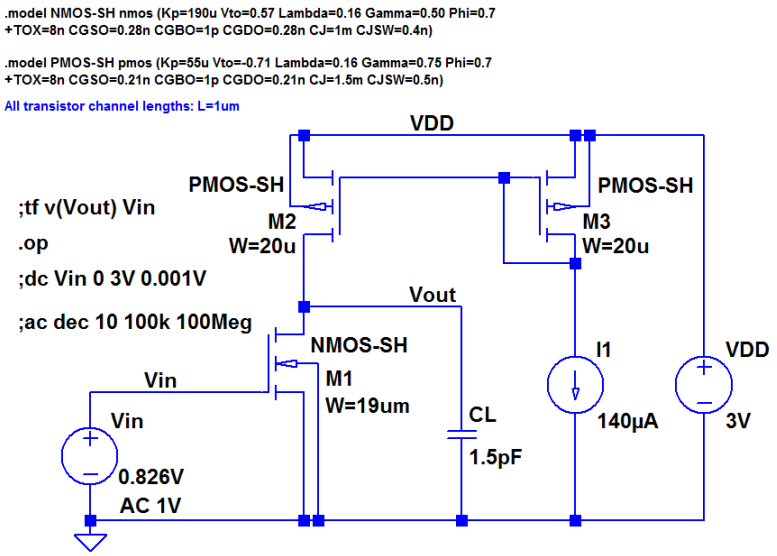


Figure 4.5: LTspice schematic for simulating the common-source stage with a PMOS active load, drawn with correct polarity of PMOS transistors.

symbols appropriately in order to get the correct schematic as shown in Fig. 4.5 where the connection to the PMOS gates is at the upper end of the gate electrode. You may have realized that the MOS transistor (with equal dimensions of drain and source diffusions) is a symmetrical device, so even if you do not perform the rotation and mirroring as in Fig. 4.5, you would expect the same performance of the circuit.

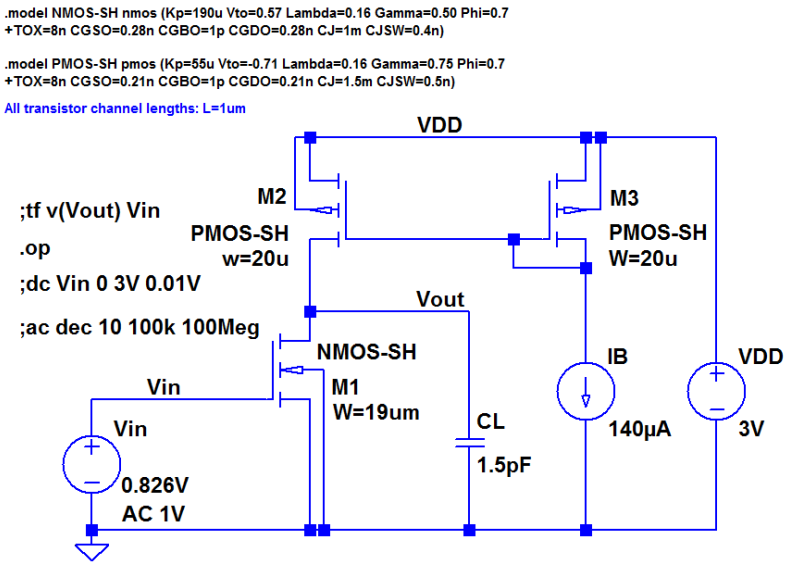


Figure 4.6: LTspice schematic for simulating the common-source stage with a PMOS active load, drawn with inverted polarity of PMOS transistors.

Figure 4.6 shows the schematic drawn without rotation of the PMOS transistors. The simulation results for the output voltage is the same for the two schematics, but with the inverted drain and source terminals in Fig. 4.6, the output files from a '.op' simulation will be somewhat confusing.

SPICE Error Log

```
Circuit: * M:\LTspice\Tutorial04\fig4_05.asc

Semiconductor Device Operating Points:
                          --- MOSFET Transistors ---
Name:            m3           m2           m1
Model:        pmos-sh      pmos-sh      nmos-sh
Id:        -1.40e-04    -1.46e-04     1.46e-04
Vgs:       -1.17e+00    -1.17e+00     8.26e-01
Vds:       -1.17e+00    -1.51e+00     1.49e+00
Vbs:        0.00e+00     0.00e+00     0.00e+00
Vth:       -7.10e-01    -7.10e-01     5.70e-01
Vdsat:     -4.63e-01    -4.63e-01     2.56e-01
Gm:         6.05e-04     6.33e-04     1.14e-03
Gds:        1.89e-05     1.89e-05     1.89e-05
Gmb:        2.71e-04     2.84e-04     3.42e-04
Cbd:        2.61e-14     2.41e-14     8.75e-15
Cbs:        4.10e-14     4.10e-14     1.48e-14
Cgsov:      4.20e-15     4.20e-15     5.32e-15
Cgdov:      4.20e-15     4.20e-15     5.32e-15
Cgbov:      1.00e-18     1.00e-18     1.00e-18
Cgs:        5.76e-14     5.76e-14     5.47e-14
Cgd:        0.00e+00     0.00e+00     0.00e+00
Cgb:        0.00e+00     0.00e+00     0.00e+00
```

(a): Correct polarity of PMOS transistors.

SPICE Error Log

```
Circuit: * M:\LTspice\Tutorial04\fig4_06.asc

Semiconductor Device Operating Points:
                          --- MOSFET Transistors ---
Name:            m3           m2           m1
Model:        pmos-sh      pmos-sh      nmos-sh
Id:         1.40e-04     1.46e-04     1.46e-04
Vgs:        0.00e+00     3.41e-01     8.26e-01
Vds:        1.17e+00     1.51e+00     1.49e+00
Vbs:        1.17e+00     1.51e+00     0.00e+00
Vth:       -7.10e-01    -7.10e-01     5.70e-01
Vdsat:     -4.63e-01    -4.63e-01     2.56e-01
Gm:         6.05e-04     6.33e-04     1.14e-03
Gds:        1.89e-05     1.89e-05     1.89e-05
Gmb:        2.71e-04     2.84e-04     3.42e-04
Cbd:        4.10e-14     4.10e-14     1.62e-14
Cbs:        2.61e-14     2.41e-14     2.74e-14
Cgsov:      4.20e-15     4.20e-15     5.32e-15
Cgdov:      4.20e-15     4.20e-15     5.32e-15
Cgbov:      1.00e-18     1.00e-18     1.00e-18
Cgs:        0.00e+00     0.00e+00     5.47e-14
Cgd:        5.76e-14     5.76e-14     0.00e+00
Cgb:        0.00e+00     0.00e+00     0.00e+00
```

(b): Inverted polarity of PMOS transistors.

Figure 4.7: Error log output files corresponding to Fig. 4.5 and Fig. 4.6, respectively.