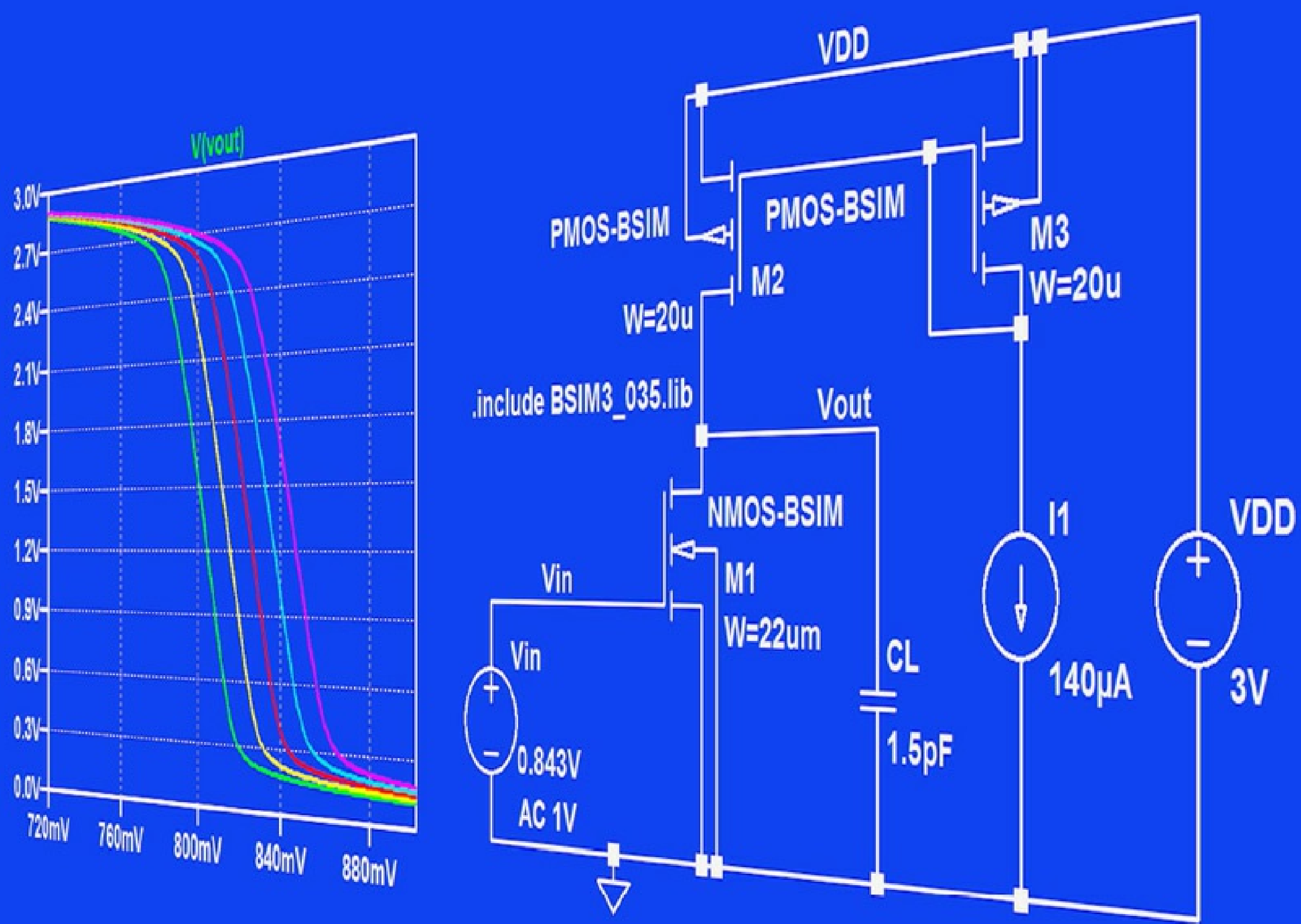
CMOS Integrated Circuit Simulation with LTspice IV

a Tutorial Guide

Erik Bruun



Erik Bruun

CMOS Integrated Circuit Simulation with LTspice IV – a Tutorial Guide

CMOS Integrated Circuit Simulation with LTspice IV – a Tutorial Guide
1st edition

ISBN 978-87-403-1059-7
Peer reviewed by Dennis Øland Larsen, IC design engineer, GN ReSound

Contents

Preface

This book is about circuit simulation with 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.

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. common source, common drain, common gate and 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 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.

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 files may be copied directly from the electronic version of this book into a text editor.

Acknowledgements: The author would like to acknowledge the many students who have contributed with comments and suggestion for the book. Also, a particular acknowledgement goes to my colleague Dennis Øland Larsen who reviewed the entire manuscript and provided many useful comments and corrections during the final phase of writing.

Getting started

The program LTspice is freely available from Linear Technology,
http://www.linear.com/designtools/software/.
Just click ‘Download LTspice’ and follow the instructions. You may register for an account with Linear Technology, but you do not have to. You may just click ‘No thanks, just download the software’ and choose ‘Run’ in the dialogue box which appears.

A ‘Getting started guide’ is available from
http://cds.linear.com/docs/en/software-and-simulation/LTspiceGettingStartedGuide.pdf.
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.

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). 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', 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 130 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, the following page provides a list of some of the differences which may initially cause confusion.

- The toolbar shown in fig. 1.2 on page 14 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 'Net Names' as explained on page 15.
- 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'.
- The 'Simulate' command shown in fig. 1.2 on page 14 and described on page 16 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 on page 18 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 '.step' commands.
- The result of a 'DC operating point' simulation ('.op') is not automatically displayed in a window like shown in fig. 1.6 on page 18. 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 on page 23. 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. 1.21 on page 32. 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 as described on page 23. It must be done manually.
- 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 plot 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:

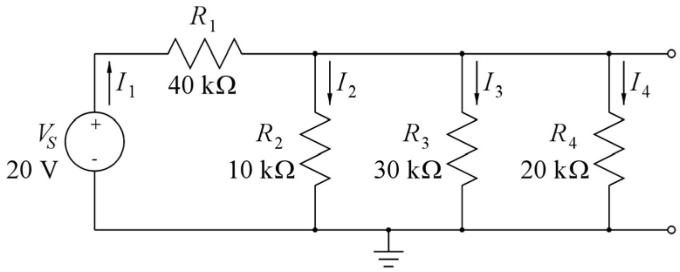


Figure 1.1: Circuit for first simulation.

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.

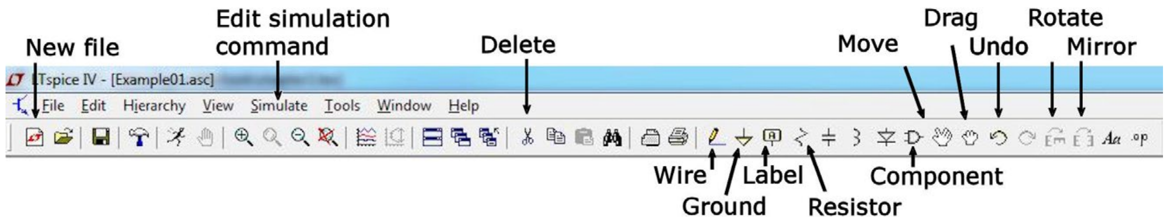


Figure 1.2: Some toolbar symbols.

Click (left mouse click) 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 number (the reference designator, e.g. R1). 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 a dialogue box where you can enter the new reference designator. Likewise, the value of the resistor is edited by right clicking R.

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 designation is 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.

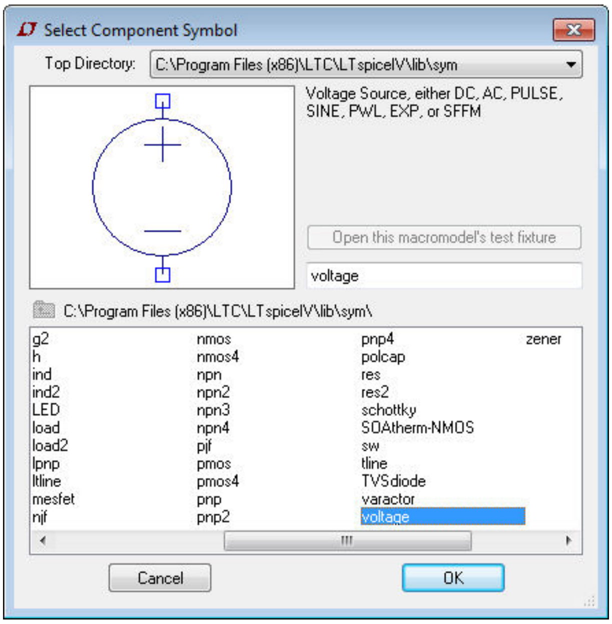


Figure 1.3: Component selection box.

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 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 → Hotkeys'.

The resulting schematic may look like the schematic shown in fig. 1.4. When the schematic is 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 on page 18. Each tab provides help for the basic simulation modes in LTspice. These are:

Transient: Perform a non-linear 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 function of one (or more) signals varying in magnitude, e.g. the output voltage of an amplifier as a funtion 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 which can now be placed on the schematic by the cursor. Insert the command by a left mouse click or by hitting 'Return'.

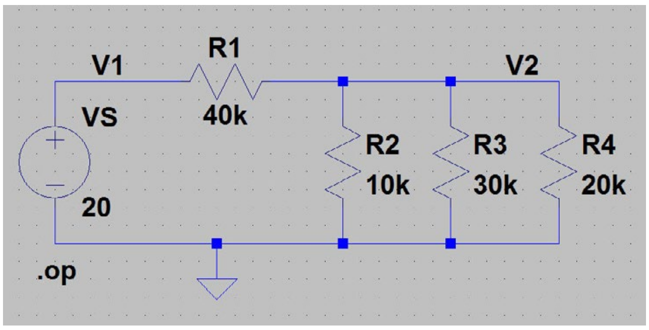


Figure 1.4: Schematic from LTspice.

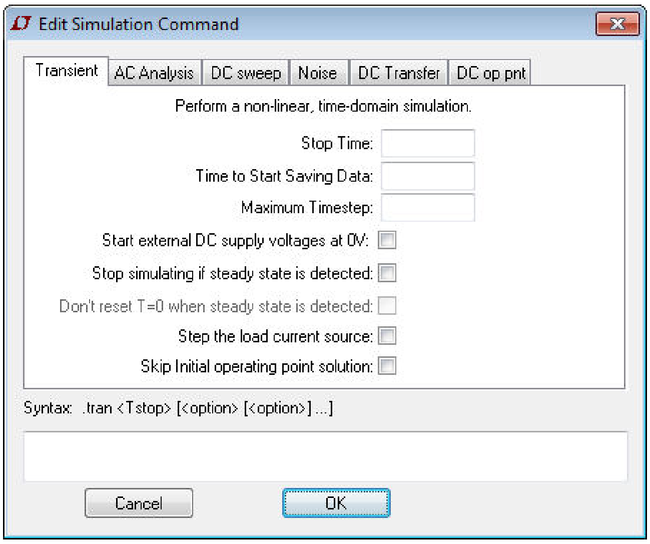


Figure 1.5: Help window for editing the simulation commands.

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(R2):	0.00024	device_current
I(R3):	8e-005	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. If there are no errors in the schematic, the simulation results in a new window being opened 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 'milli' is m (or M – LTspice is case insensitive) while the suffix for 'Mega' is Meg (or meg).

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'. 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.

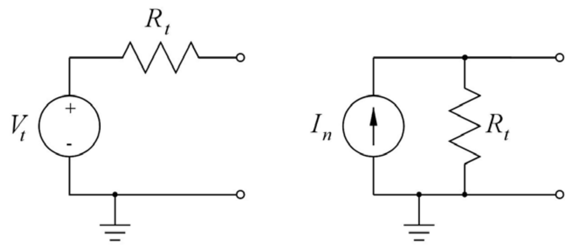


Figure 1.7: Thévenin equivalent (left) and Norton equivalent (right).

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. 1.7 (Hambley 2014). 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 seen from 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 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. Alternatively, you can model the short-circuit by a voltage source with a value of 0 V. 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 $I_n = 0.5$ mA, and you can calculate R_t from $R_t = V_t/I_n = 4.8$ kΩ. Alternatively, R_t can be found by simulation: Insert a current source I_1 between the two rightmost terminals 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 found as V_2/I_1, so if 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.

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. 1.8. Notice that in this figure the font size has been specified to 1.0 when inserting the text (the default is 1.5), and the background colour has been changed to white using the command 'Tools → Color Preferences' which opens a 'Color Palette Editor' for specifying the colours being used for schematics, netlists and waveforms.

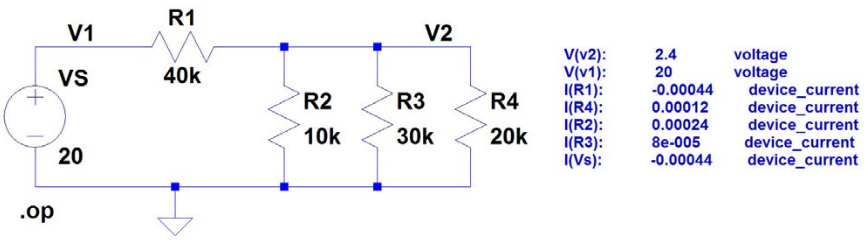


Figure 1.8: The circuit from fig. 1.4 with the results of the '.op' simulation shown as text.

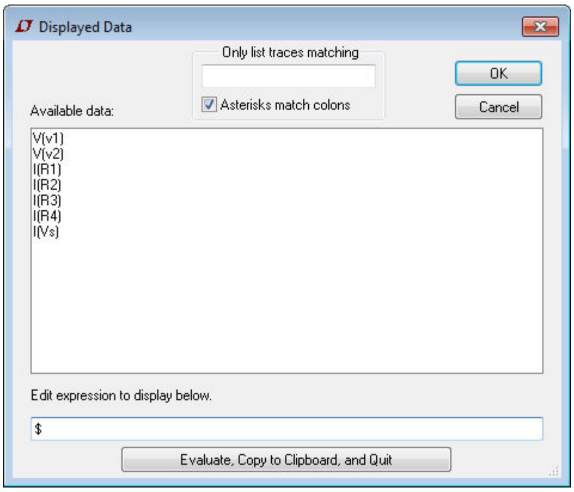
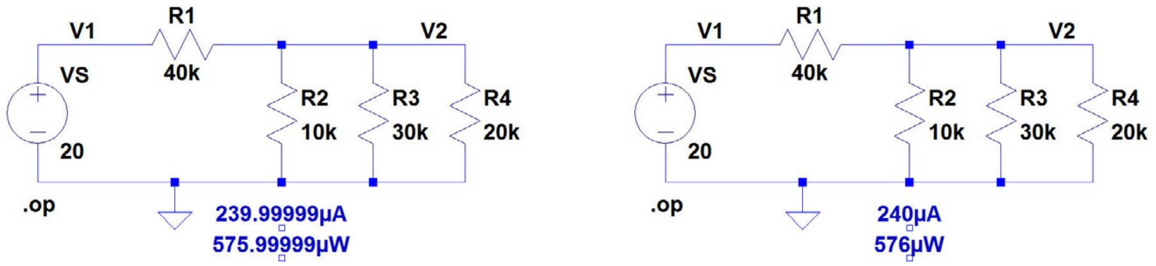


Figure 1.9: Dialogue box for entering simulation results on the schematic.

An alternative way to display specific simulation results is as follows: After having run the simulation, right click in an empty space on the schematic. This opens a command selection menu. Select the command 'View → Place .op Data Label'. This opens a text box which can be placed at an appropriate location on the schematic. The text box just contains three question marks, `???`. (An alternative way of opening this text box is to left click on a net label in the schematic ('`V1`' or '`V2`' in fig. 1.8).) When you right click on the question marks, the dialogue box shown in fig. 1.9 opens. 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. 1.9, by replacing the $ sign in the bottom line with



(a) (b)

Figure 1.10: The circuit from fig. 1.4 with the current and the power for R_2 shown on the schematic.

'`I(R2)`'. Adding a new text box in the same way lets you specify the expression '`I(R2)*V(v2)`' which will calculate the power in R_2. The resulting schematic may look like shown in fig. 1.10(a). You may find that the current and power need rounding off to integer µA and µW. This can be achieved 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 the resulting schematic looks like shown in fig. 1.10(b).

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 in the system.

You can calculate the voltages and currents for other values of components 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 mouse click or by typing '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.

Assuming that 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 mouse 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 can 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 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.

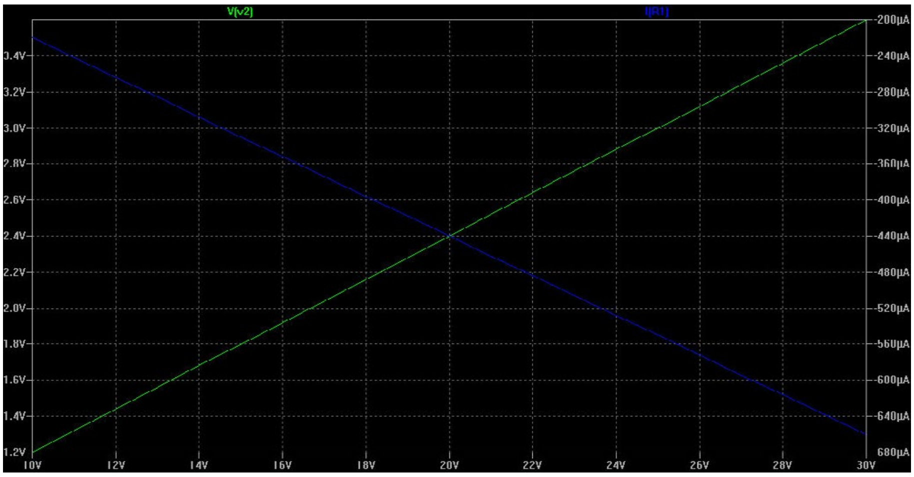


Figure 1.11: Plot of DC sweep simulation for circuit example from fig. 1.4 using the LTspice default setup of colours.

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. 1.11. You may find that the blue trace ('`I(R1)`') is difficult to see on the black background. You can change the colour of the trace by pointing to the trace name above the plot and right clicking. This opens a window where you can select another colour. Alternatively, you may change the background colour of the plot pane by the command 'Tools → Color Preferences' which opens a dialogue window where you can specify colours 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 current 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' 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 →

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 1.12: Table with results of DC sweep simulation for circuit example from fig. 1.4.

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. 1.12.

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. 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 your circuit does not have any errors, 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. Fig. 1.13 shows the schematic from fig. 1.4 with the '.step' command inserted, and it shows the resulting waveform plot of V_2. Here, the colour preferences of the waveform plot and the schematics 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

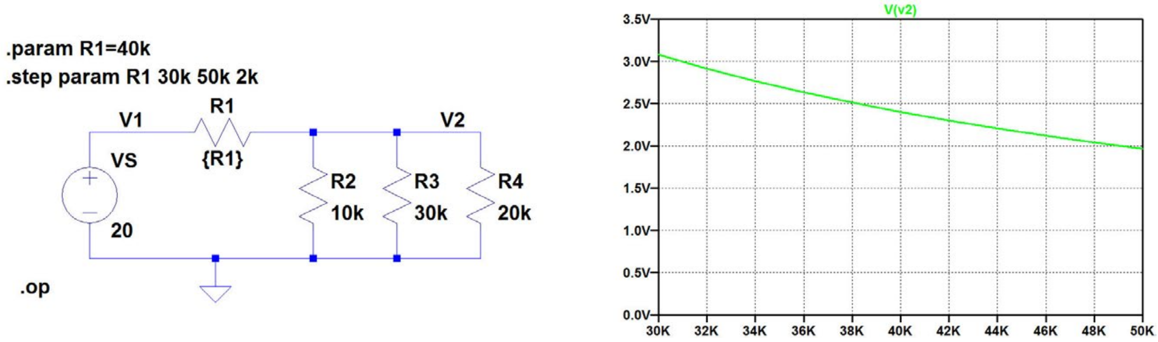


Figure 1.13: Simulation of sweep of resistor R_1 from fig. 1.1.

'Plot settings → Manual Limits' or by moving the mouse cursor over the axis and left clicking. In fig. 1.13 (and in subsequent figures showing simulation plots), the font size of the labels on the axes has been increased using the 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'.

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.

If you want to run a simulation with just one value for a variable parameter (R1 in fig. 1.13), then instead of the '.step' command, you can specify the value of R1 using a '.param' command: Insert the SPICE Directive '`.param R1=40k`' to run a simulation with $R_1 = 40\ \mathrm{k\Omega}$ and delete the '.step' command or edit it into a comment by inserting an asterix (*) as the first character or by ticking 'Comment' in the editing window. If you do not disable the '.step' command, the simulation will run this command regardless of the '.param' specification.

The '.step param' command 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. Problems 1.2 on page 38 and 1.5 on page 39 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. 1.14. 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. 1.14, a load resistor R_L and a signal source V_S with a source resistance R_S is connected to the amplifier.

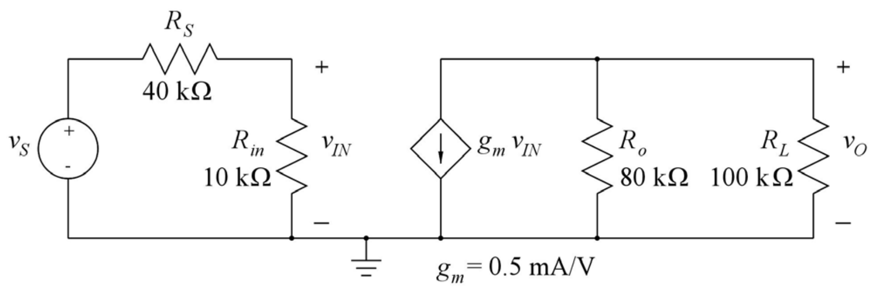


Figure 1.14: An inverting transconductance amplifier.

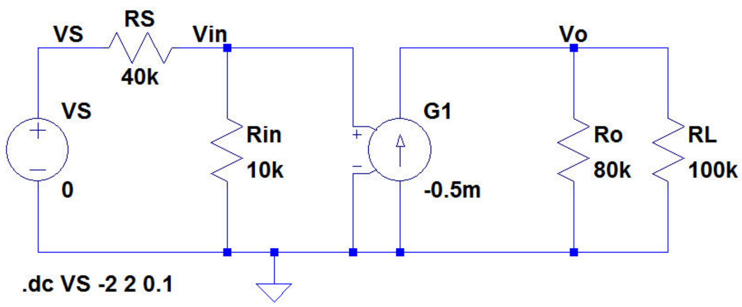


Figure 1.15: LTspice schematic for the inverting transconductance amplifier.

In this circuit, there is a new type of component, the 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 schematic drawn in LTspice is shown in fig. 1.15. The LTspice symbol for the controlled current source explicitly shows the controlling voltage as input terminals to the component symbol. The controlled current source is edited by right clicking on the symbol. This opens a 'Component Attribute Editor' as shown in fig. 1.16. By double clicking on the values for 'InstName' and 'Value', the values can be changed to the values shown in fig. 1.15. Alternatively, just right click on the device number (e.g. G1) and the value G to edit them

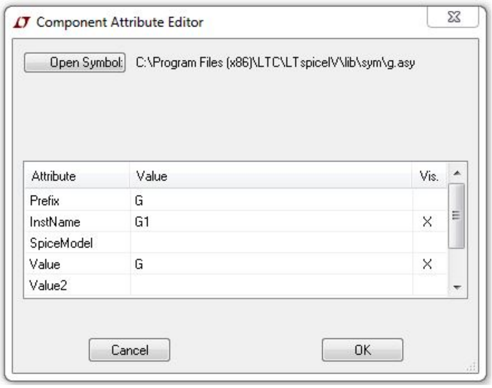


Figure 1.16: The window for editing the specifications of the voltage controlled current source.

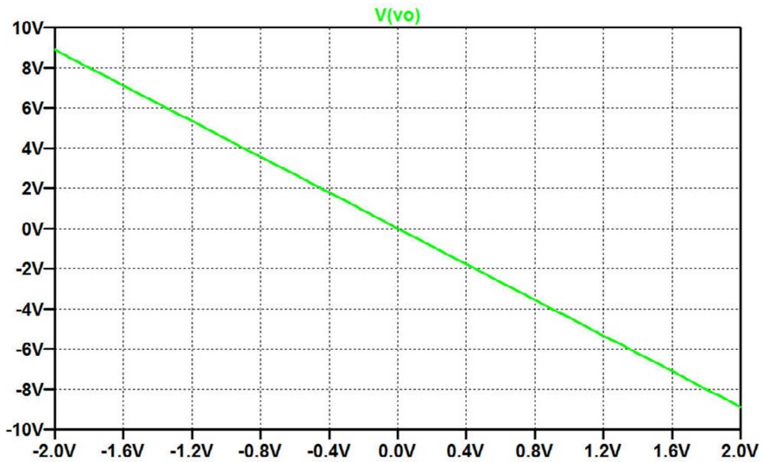


Figure 1.17: Plot of v_O versus v_S for the inverting amplifier.

to the desired values in the same way as editing the value of a resistor or a DC current source. By editing the 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. 1.17.

The arbitrary behavioural source: LTspice also provides an alternative to the voltage controlled current source. This is an 'Arbitrary behavioural 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. Fig. 1.18 shows the circuit from fig. 1.14 redrawn with the 'bi' symbol. Notice the definition 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 $*$ is the character indicating multiplication.

In fig. 1.18, the symbol for the controlled current source is a circle, exactly like the symbol for an

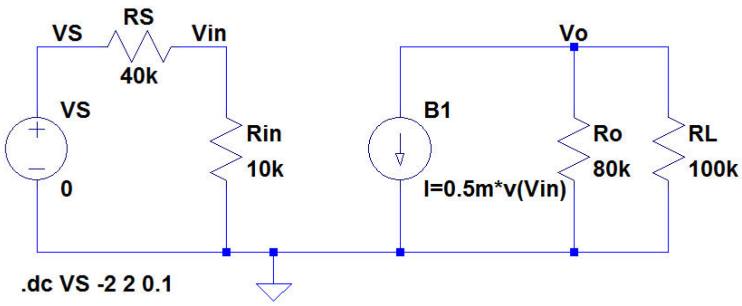


Figure 1.18: LTspice schematic for the inverting transconductance amplifier using an arbitrary behavioural current source.

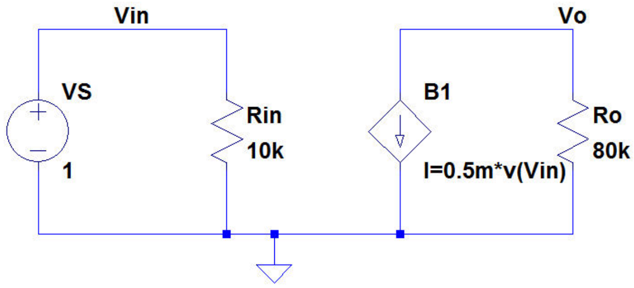


Figure 1.19: The inverting amplifier with a diamond shaped symbol for the arbitrary behavioural current source.

independent current source. Often in the literature, controlled sources are represented by a diamond shaped symbol to distinguish them from the independent sources (Hambley 2014; Sedra & Smith 2011). 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. 1.16), you click 'Open Symbol' to enter the symbol editor where you can redraw the shape of the symbol.

In fig. 1.19, 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 $A_{voc} = -g_m \times R_o = -40$ 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. 1.19 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. 1.20.

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. Obviously, the

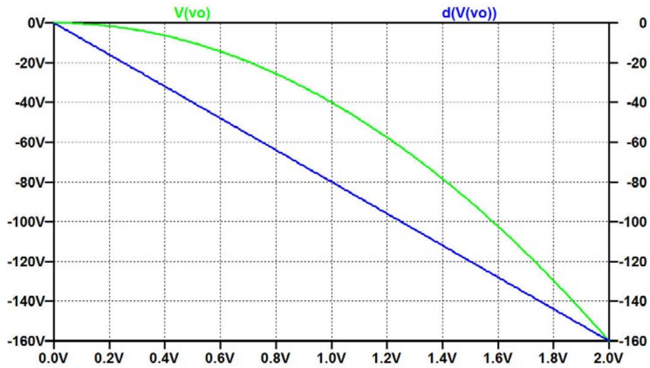


Figure 1.20: Plot of v_O versus v_S for the inverting amplifier with a nonlinear voltage controlled current source.

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 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), 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. 1.20 from which you can see that $A_{voc} = -80$ V/V as expected for $V_{IN} = 1$ V.

Output from DC Transfer simulation		
- - - Transfer Function - - -		
Transfer_function:	-80	transfer
vs#Input_impedance:	10000	impedance
output_impedance_at_V(vo):	80000	impedance

Figure 1.21: Output from '.tf' simulation of the circuit from fig. 1.19.

LTspice has another simulation command which will directly give 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. 1.19, 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. 1.21.

Example 1.3: A current amplifier.

The final example in this tutorial is a current amplifier as shown in fig. 1.22. 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_{isc} and an output resistance R_o.

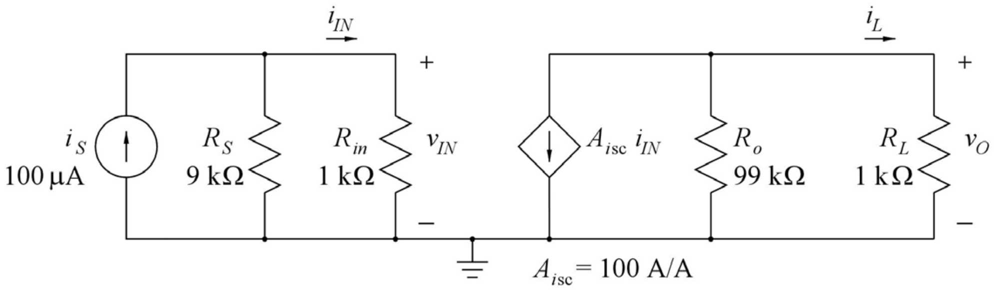


Figure 1.22: An inverting current amplifier.

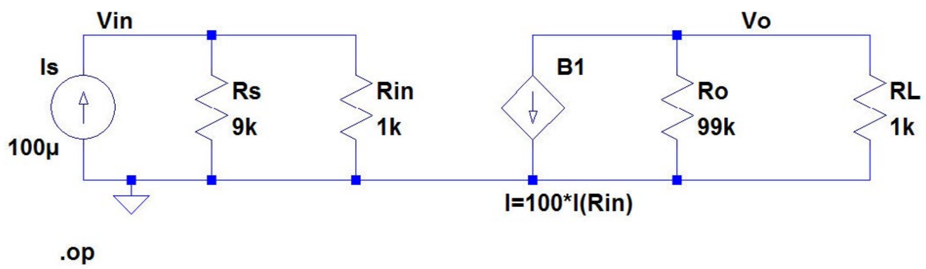


Figure 1.23: LTspice schematic for the inverting current amplifier.

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. 1.22, 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 behavioural current source', device type 'bi' in the component selection. Fig. 1.23 shows the schematic drawn with the arbitrary behavioural current source (using a diamond shaped symbol). Obviously, when examining fig. 1.22, 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. 1.23. Running a '.op' simulation indeed also results in the expected values of i_L and v_O.

But running a '.tf' simulation (see page 32) 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.

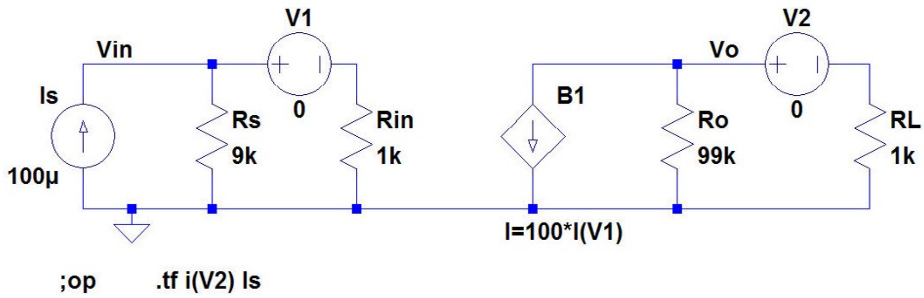


Figure 1.24: LTspice schematic for the inverting current amplifier with voltage sources in series with R_{in} and R_L and altered specification for B1.

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 as described on page 20.

Fig. 1.24 shows the circuit from fig. 1.23 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 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.

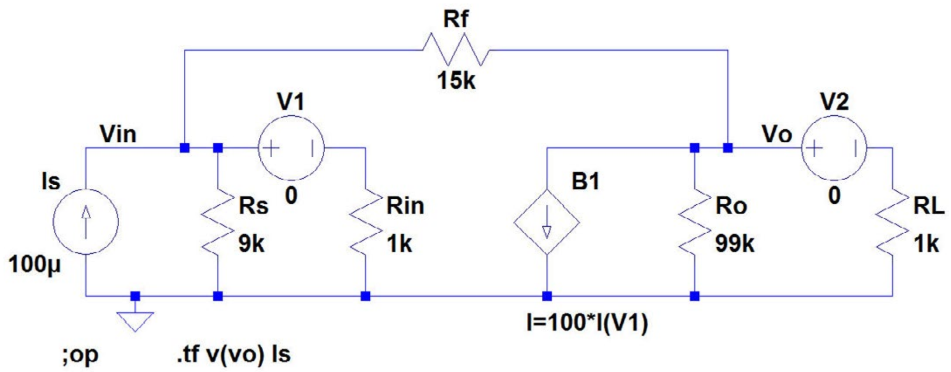


Figure 1.25: LTspice schematic for the inverting current amplifier with a feedback resistor R_f.

Now, let us connect a feedback resistor R_f of 15 kΩ between output and input as shown in fig. 1.25, shunt - shunt feedback (Sedra & Smith 2011). 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, 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 all 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. Examples of such circuits are given in problems P1.3 on page 38 and P1.4 on page 39.

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 a 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.

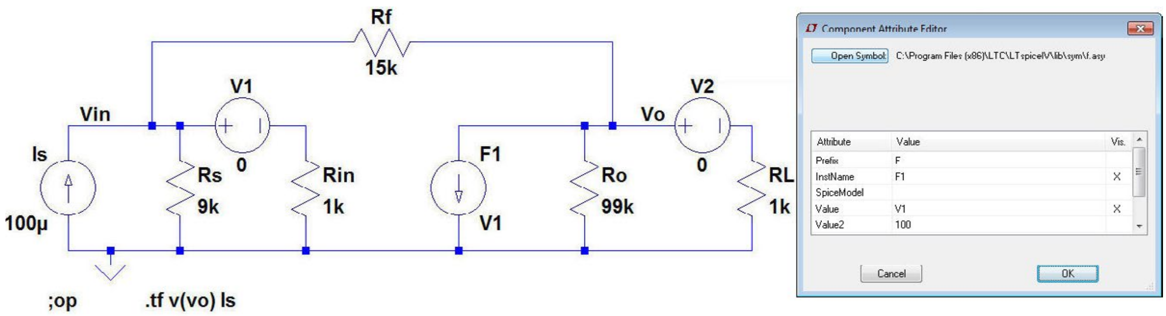


Figure 1.26: Circuit from fig. 1.25 redrawn with a current controlled current source instead of an arbitrary controlled current source.

This is also the way to specify a controlling current when using the current controlled current source with circuit designator F. Fig. 1.26 shows the circuit from fig. 1.25 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’.

By now you should be well prepared for analyzing resistive circuits with different kinds of voltage sources and current sources such as the examples given in the following problems.

Hints and pitfalls

- The suffix for ‘milli’ is ‘m’. The suffix for ‘Mega’ is ‘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’.
- 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’.
- 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 on page 41 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.
- Colour 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 .
- In a waveform plot, you can zoom in on details by clicking and dragging to define a box using the left mouse button.
- Text and other annotations (e.g. cursor position) can be placed in a simulation plot using the command ‘Plot Settings → Notes & Annotations’.
- 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 a 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. 2014, *Electrical Engineering, Principles and Applications*, Sixth Edition, Pearson Education Ltd., Harlow, UK.

Sedra, AS. & Smith, KC. 2011, *Microelectronic Circuits*, International Sixth 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

1.1

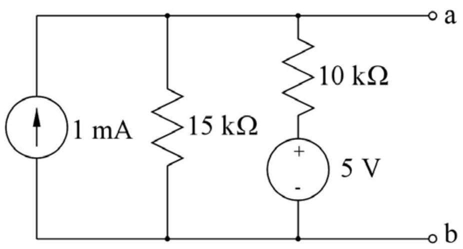


Figure P1.1

For the circuit shown in fig. P1.1, 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.

1.2

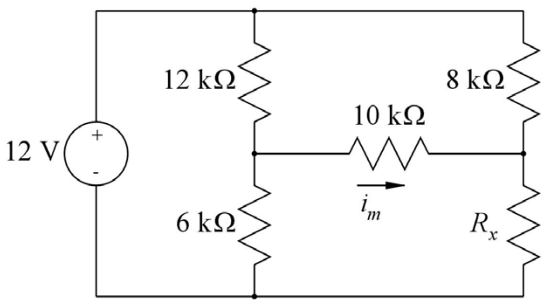


Figure P1.2

For the circuit shown in fig. P1.2, determine the value of resistor R_x so that the current i_m in the 10 kΩ resistor is 30 μA.

1.3

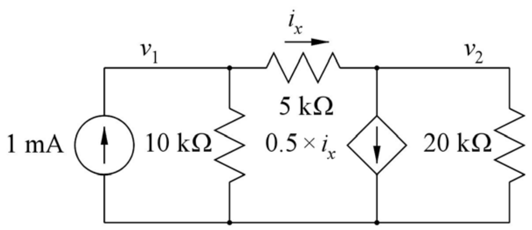


Figure P1.3

For the circuit shown in fig. P1.3, determine the value of the voltages v_1 and v_2 and the current i_x.

1.4

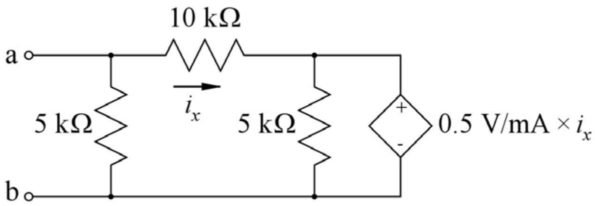


Figure P1.4

For the circuit shown in fig. P1.4, find the equivalent resistance looking into terminals a – b.

1.5

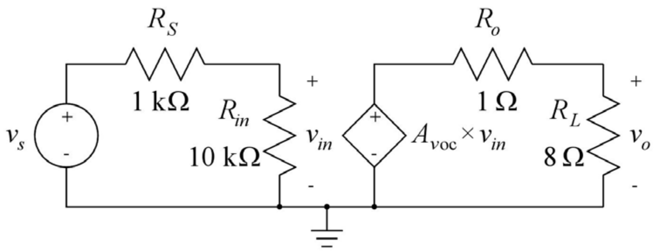


Figure P1.5

For the circuit shown in fig. P1.5, 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 input voltage for v_s in the range from 0 mV to 100 mV.

1.6

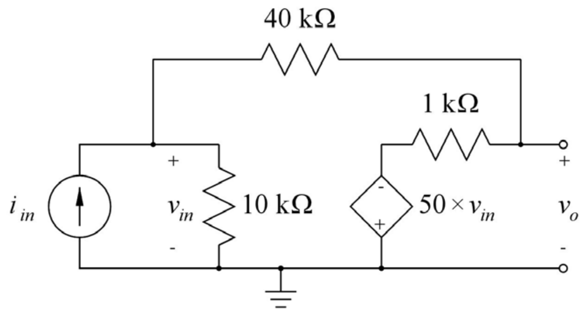


Figure P1.6

The circuit shown in fig. P1.6 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.

1.7

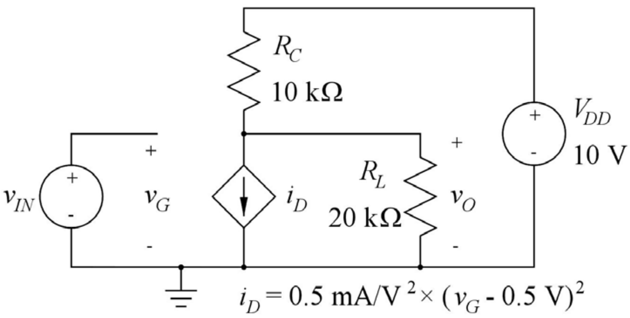


Figure P1.7

Fig. P1.7 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.

1.8

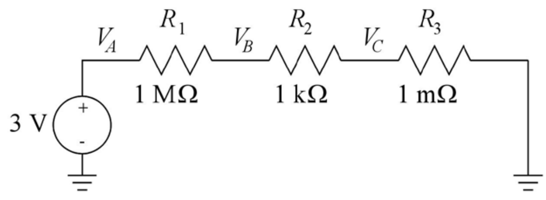


Figure P1.8

Fig. P1.8 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 leftmost ground symbol across the components to rightmost ground symbol.
(3): Insert the ground symbols, draw an unbroken wire between them, and then insert the component directly on top of the wire.
Observe how LTspice 'cleans up' the wiring.
Find the voltages V_A, V_B and V_C.

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: $A_{voc} = 70$ V/V.

1.6: $R_{moc} = -36.28$ 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.

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.

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:

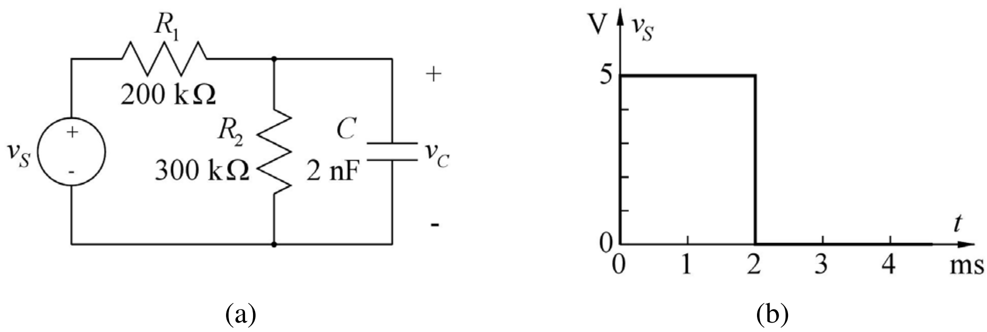


Figure 2.1: RC network (a) and input voltage v_S to the network (b).

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 time $t = 0$ 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 mouse click opens a window for specifying the voltage source. In this window, left click on 'Advanced'. This opens a dialogue box 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 from fig. 2.1(b) have to have a certain slope. Corresponding to the timing in fig. 2.1(b), you may enter time1=0s, value1=0V, time2=0.1us, value2=5V, time3=2ms, value3=5V, time4=2.0001ms, value4=0V. In this way, the voltage changes from 0 V to 5 V in 0.1 μs. An alternative to specifying time4 as 2.0001ms is to specify time4={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.

Next, you should specify the simulation. Use the command 'Simulate → Edit Simulation Cmd' and open the tab 'Transient'. For a simulation of the charging and discharging of the capacitor, you can run the simulation starting at time $t = 0$ s and stopping at the time specified in the box for 'Stop Time'. 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 will open a plot window with a time axis and by pointing to '`VC`' on the schematic (the red pointer,), the capacitor voltage v_C will be shown in the plot window. Fig. 2.2 shows both the schematic drawn in LTspice and the resulting plot window.

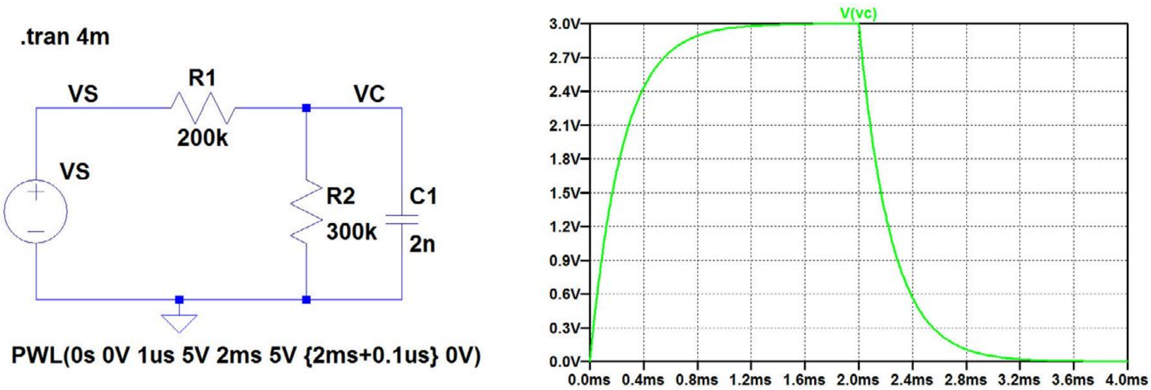


Figure 2.2: LTspice schematic and simulation results for the circuit in fig. 2.1.

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

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

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

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

The time constant can also be found from the simulation of the charging. From $v_C = V_0 \times (1 - \exp(-t/\tau))$, 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 will be 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. In the bottom line, you can enter 'Expression(s) to add'. In this case, enter '`V(vc)/3/(1-1/e)`' and click

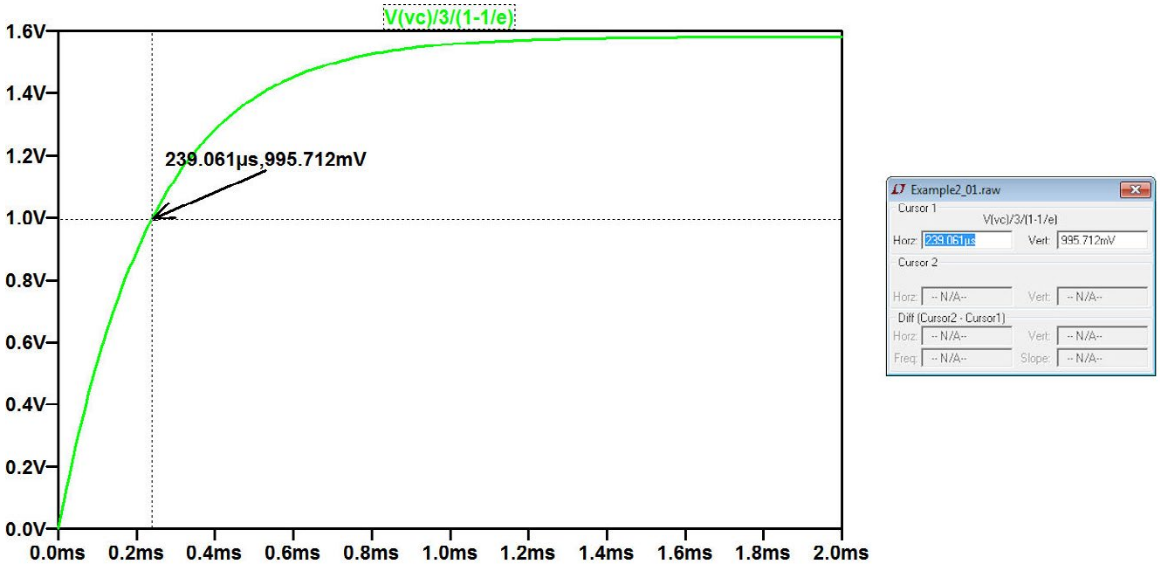


Figure 2.3: Plot window with scaled output for finding the time constant.

'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.

Fig. 2.3 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. Also, the grid has been turned off by the command 'Plot Settings → Grid' (or 'Ctrl-G') in order to make the cursor more visible. The figure also shows the window with the coordinates for the position of the cursor. This window can be copied to the clipboard using the standard 'Print screen' function ('Alt-PrtScn'). Using the command 'Plot Settings → Notes & Annotations', you can also directly insert the cursor position in the plot as shown in fig. 2.3.

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. 2.4. The switch is open for $t < 0$, closes when $t = 0$ 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.

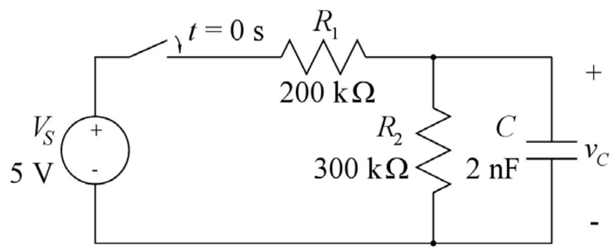


Figure 2.4: The RC network from fig. 2.1 with a switch between the voltage source and the resistor R_1.

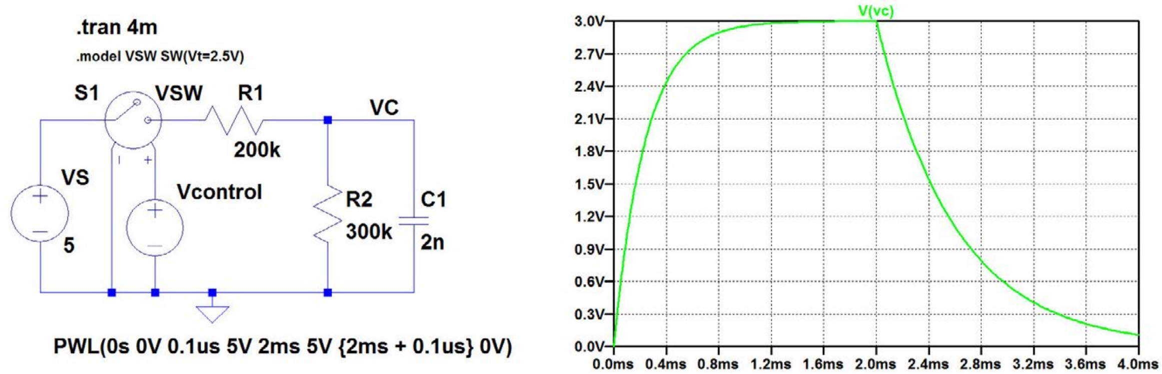


Figure 2.5: LTspice schematic and simulation results for the circuit in fig. 2.4.

When you insert the switch in the schematic, it appears with a reference to a default switch model SW. You may use the default model without inserting a '.model' specification. If you wish to change some parameters of the switch, you must include a '.model' specification and it is a good idea to use another name for the model, rather than the default SW.

Fig. 2.5 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. 2.2 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. 2.2. Also shown in fig. 2.5 is the simulation result for v_C, compare fig. 2.2.

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 $\tau = R_2 \times C = 600$ µs. By scaling the voltage v_C in the same way as in fig. 2.3, 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, see fig. 2.6.

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 $\omega_0 = 1/\tau = 1/((R_1 \parallel R_2) \times C)) = 4.17 \times 10^3$ 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

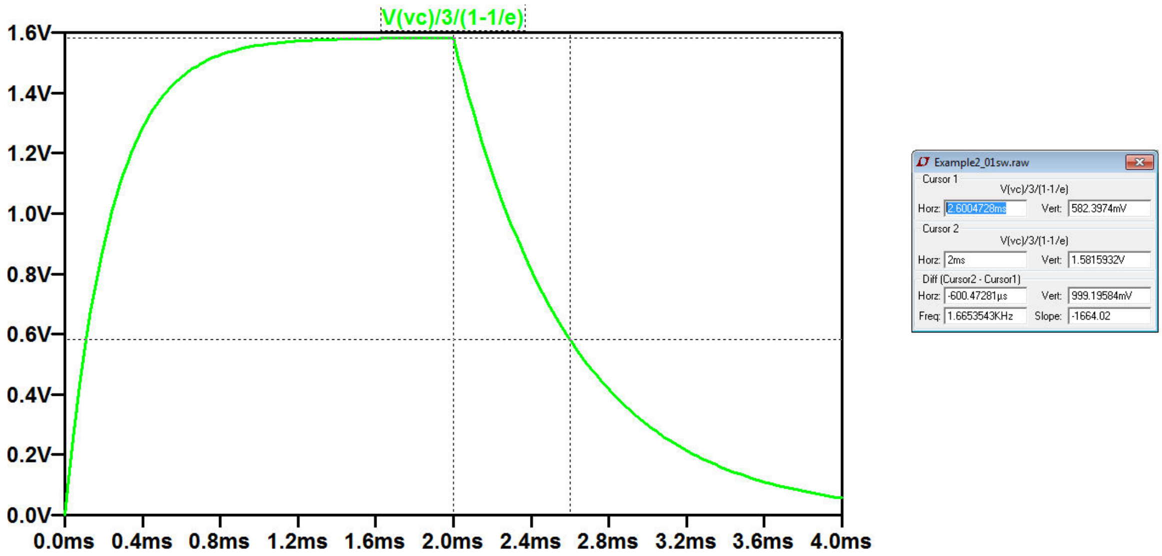


Figure 2.6: Plot window with scaled output for finding the time constant.

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 mouse 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 will have to 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 remove the time varying 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_c as the trace to show, both an amplitude plot and a phase plot appear as shown in fig. 2.7. In this plot, the colour 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 $-90°$.

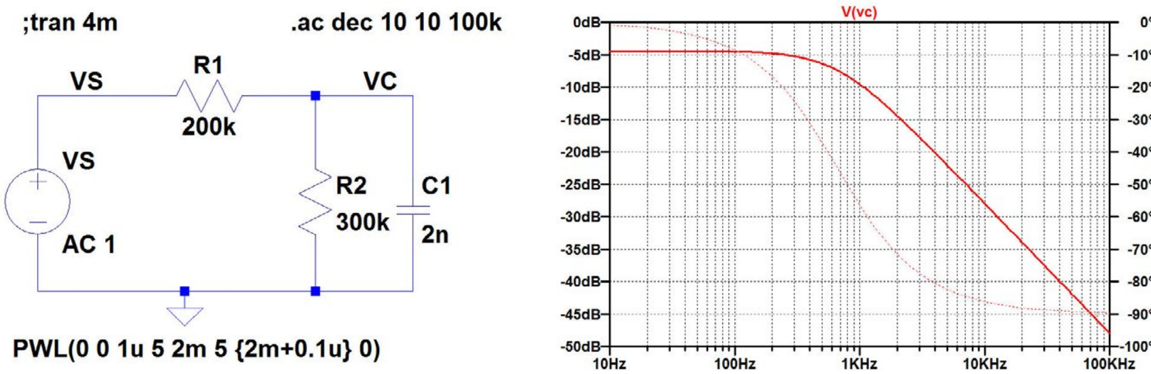


Figure 2.7: LTspice schematic and AC simulation results for the lowpass filter from fig. 2.1.

The -3 dB frequency is found using the cursors in a way similar to what was done for finding time constants. You may either 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 non-linear 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.

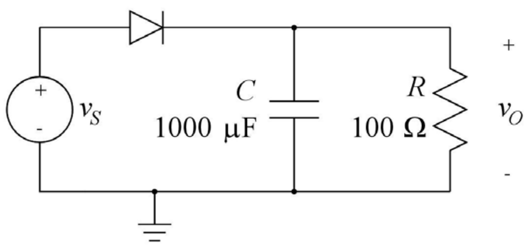


Figure 2.8: A half-wave rectifier with a smoothing filter.

The next example is a half-wave rectifier as shown in fig. 2.8. When drawing the schematic, the diode symbol is selected from the component selection box (). Like the switch in fig. 2.5, the diode is modeled by a '.model' specification. If you omit the '.model' specification, the diode defaults to the standard Shockley model $i_D = I_S[\exp(v_D/(nV_T)) - 1]$ (Hambley 2014) 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 turns the cursor into a hand . A right mouse click opens a window as shown in fig. 2.9. 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.

Figure 2.9: Window for selecting diode model.

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.

Fig. 2.10 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. 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 found in (Sedra and Smith 2011, pp. 193-194): $V_r = V_p/(fCR)$ 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 sine wave input voltage, respectively.

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

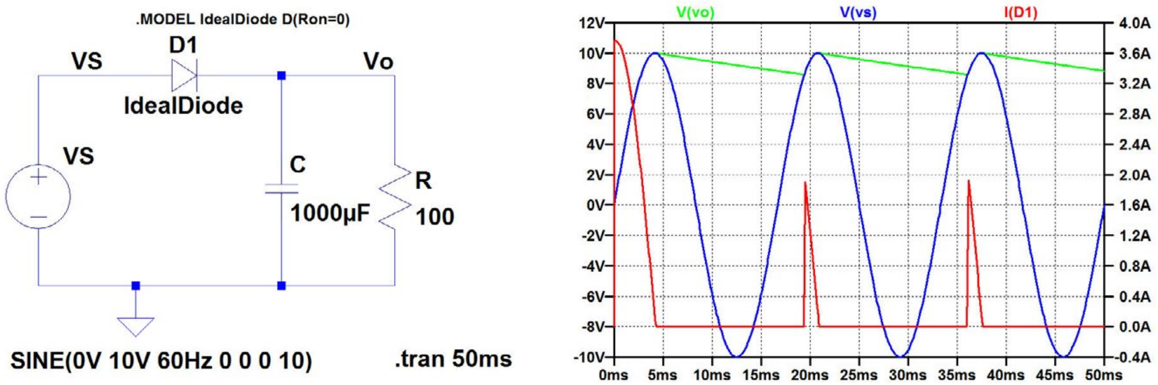


Figure 2.10: LTspice schematic and transient simulation results for the half-wave rectifier from fig. 2.8.

Example 2.3: An amplifier with capacitive feedback network.

The third example is a non-inverting amplifier with capacitive feedback, see fig. 2.11. 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

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

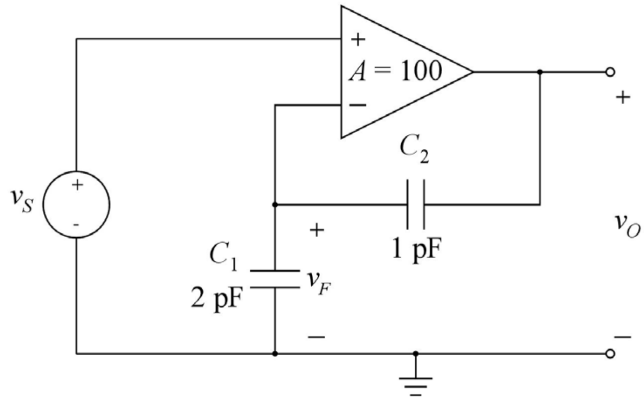


Figure 2.11: An amplifier with capacitive feedback.

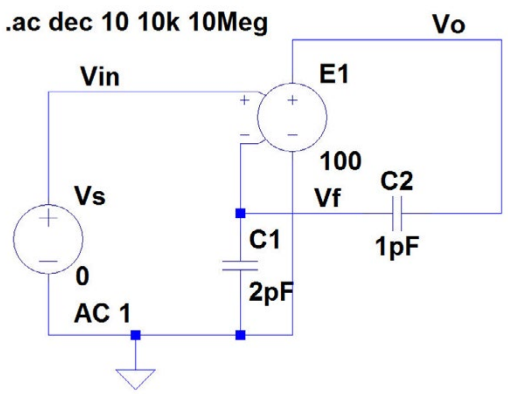
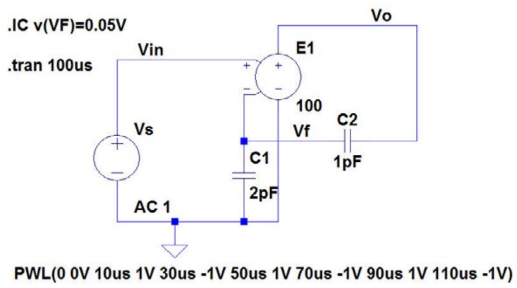


Figure 2.12: LTspice schematic for the circuit shown in fig. 2.11.

Be aware that ω has to be positive. For $\omega = 0$, the impedance of the capacitors is infinite and the gain is not defined. Fig. 2.12 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 (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. 2.12, LTspice assumes bias values of 0 V. After running the simulation, you may view the error log by using the command 'View $\rightarrow$ SPICE Error Log' (or 'Ctrl-L') and you will see that you get a warning: 'Node VF is floating'. In fig. 2.12, 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.

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 that 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. 2.12, a very large resistor (many gigaohms) may be connected in parallel with one of the capacitors. If connected in parallel with C_1, the DC gain is the open loop gain of the amplifier, i.e. $A = 100$ V/V. If connected in parallel with C_2, the DC gain of the amplifier is $A/(1+A) = 0.9901$ V/V.

For transient simulations, 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. 2.11 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 $v_O = -A \times v_F = -5$ V. Assume also that the input voltage v_S is a periodic triangular voltage with period of 40 µs, an amplitude of 1 V, a mean value of 0 V, and a start value of 0 V at time $t = 0$. Fig. 2.13 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. 2.13 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.



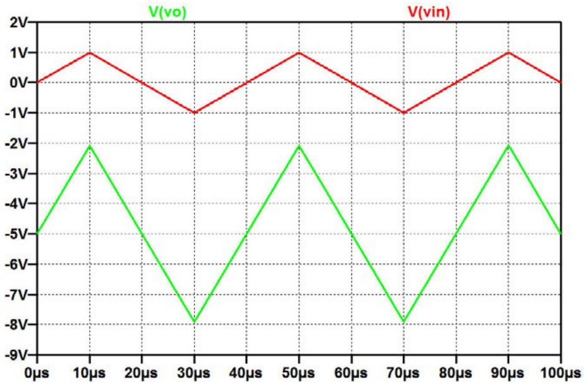


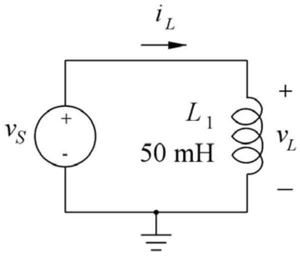
Figure 2.13: LTspice schematic for the circuit shown in fig. 2.11 with specifications for an initial value of v_F and the simulation result for a transient simulation.

Observe that the '.ic' directive affects only a transient simulation. It does not affect an AC simulation, nor an operating point analysis. If you wish to find the bias point for some specified initial value of the voltage at a floating node, you may do so by running a transient simulation with the independent signal source specified as a DC voltage (or current). In the dialogue box for specifying the independent source, just select 'Functions $\rightarrow$ (None)' and specify the appropriate DC value for the independent source. Then run a transient simulation with an arbitrary (small) value of the stop time.

Example 2.4: An ideal inductor.

Consider an inductor L_1 connected directly to a voltage source v_S as shown in fig. 2.14. The relation between current $i_L(t)$ and voltage $v_L(t)$ for the inductor is given by the equation shown in fig. 2.14 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 $t \rightarrow \infty$ which is clearly not acceptable.

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



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

Figure 2.14: An ideal inductor connected to a voltage source.

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 $i_L(t_0)$. Notice that the '.ic' directive for the inductor current works with the '.op' command whereas the '.ic' directive for a capacitor voltage does not work with '.op' command.

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. Fig. 2.15 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. 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. 2.15 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.

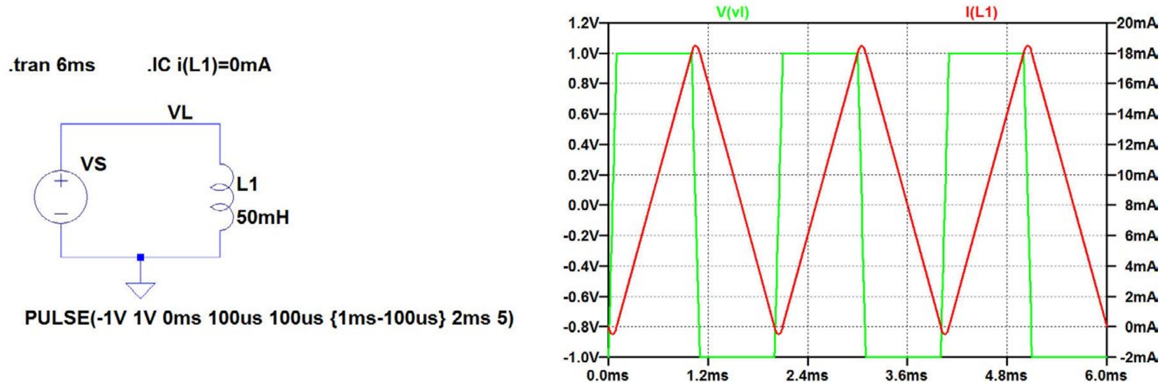


Figure 2.15: LTspice schematic for the circuit shown in fig. 2.14 with specifications for an initial value of i_{L_1} and the simulation result for a transient simulation with a pulse input.

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. 2.2 on page 45 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. 2.16. 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. 2.17.

Also the circuit from fig. 2.5 on page 47 can be simulated without the switch. The simulation of the charging of C is the same as shown in fig. 2.16. 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. 2.18.

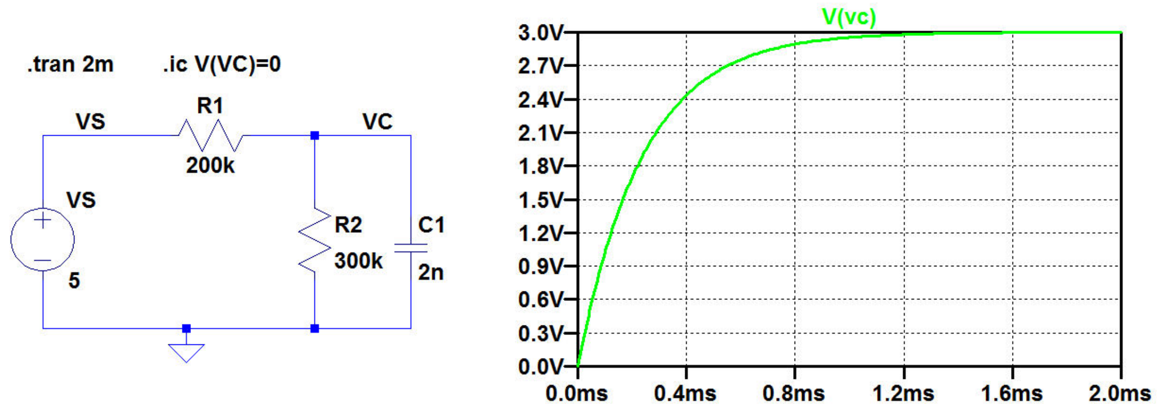


Figure 2.16: Simulation of capacitor charging using a DC voltage source and a specification of the initial capacitor voltage.

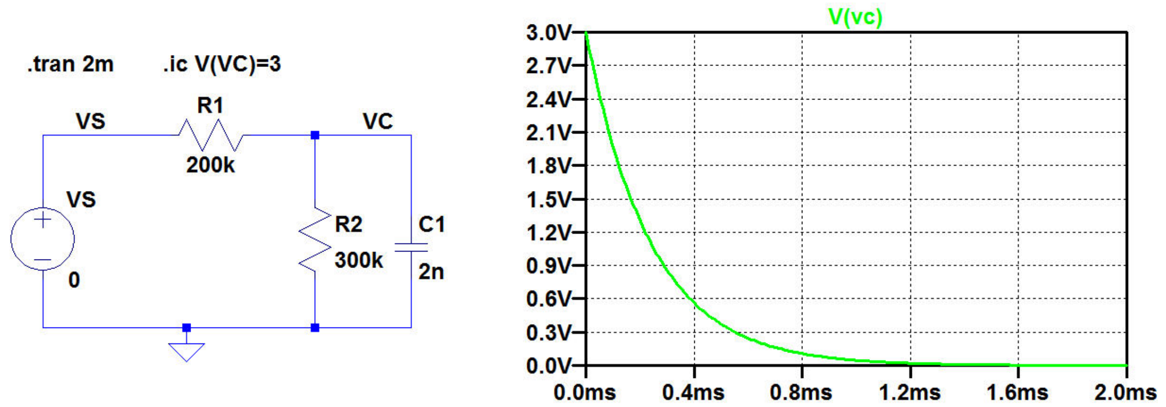


Figure 2.17: Simulation of capacitor discharging using a DC voltage source of $0\ \mathrm{V}$ and a specification of the initial capacitor voltage.

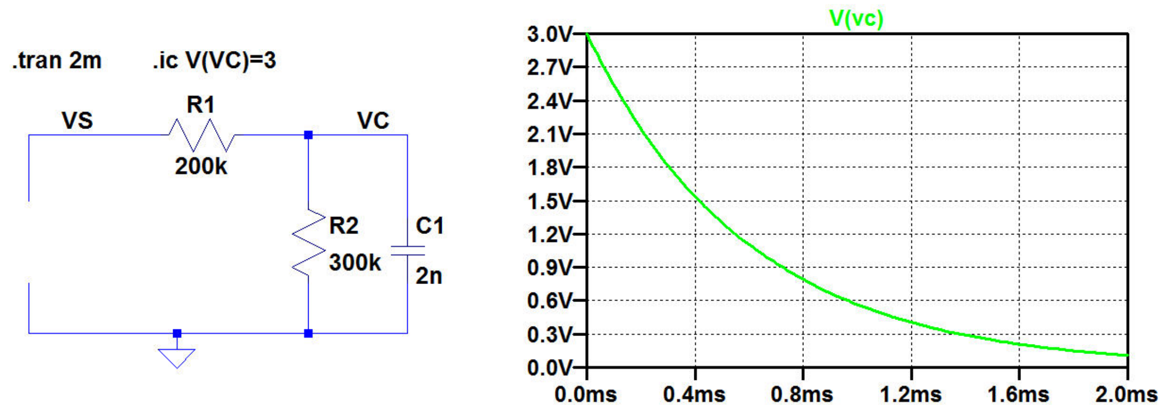


Figure 2.18: Simulation of capacitor discharging in the circuit from fig. 2.5 on page 47.

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.

As an alternative to the simple inspection of the charge and discharge using the waveform plot and the cursor as shown in fig. 2.3, you can use the SPICE Directive '.meas' or '.measure'. With this command, 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. 2.16, this is achieved by the SPICE Directive:

'`.meas tau targ 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 '`targ v(VC)=3*(1-1/e)`'. After having run the simulation, the result of the '.meas' command is found in the error log file ('Ctrl-L'). For the circuit in fig. 2.16, the result is given in the error log file as '`tau=0.000240194 FROM 0 TO 0.000240194`'.

Likewise, for the discharge time constant in figs. 2.17 and 2.18 you can use the directive:

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

For fig. 2.17, the error log file reports '`tau=0.000239805 FROM 0 TO 0.000239805`' and for fig. 2.18, the error log file reports '`tau=0.000600911 FROM 1.95312e-011 TO 0.000600911`'.

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

Hints and pitfalls

– Capacitors may give rise to floating nodes for DC voltages.

– For transient simulations, the value of a floating node may be specified using a '.ic' SPICE Directive.

– For AC simulations and simulations of operating point ('.op'), a floating node may be controlled by connecting a very large resistor from a non-floating node to the floating node.

– 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.

– LTspice is case insensitive. Thus, 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.

– The initial value for a '.op' simulation or a transient simulation of the current in an inductor can be specified by a '.ic' SPICE Directive.

– 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 and other annotations (e.g. cursor position) can be placed in a simulation plot using the command 'Plot Settings → Notes & Annotations'.

– Curly brackets `{ }` can be used to indicate expressions to be calculated by LTspice when specifying values, e.g. '`{2ms + 0.2us}`' is equivalent to '`2.0002ms`'.

References

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

Hambley, AR. 2014, *Electrical Engineering, Principles and Applications*, Sixth Edition, Pearson Education Ltd., Harlow, UK.

Sedra, AS. & Smith, KC. 2011, *Microelectronic Circuits*, International Sixth Edition, Oxford University Press, New York, USA.

Problems

2.1

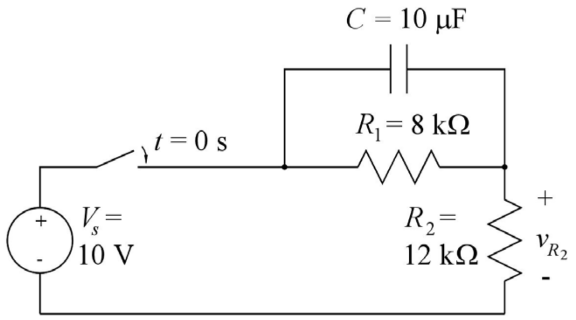


Figure P2.1

For the circuit shown in fig. P2.1, 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.

2.2

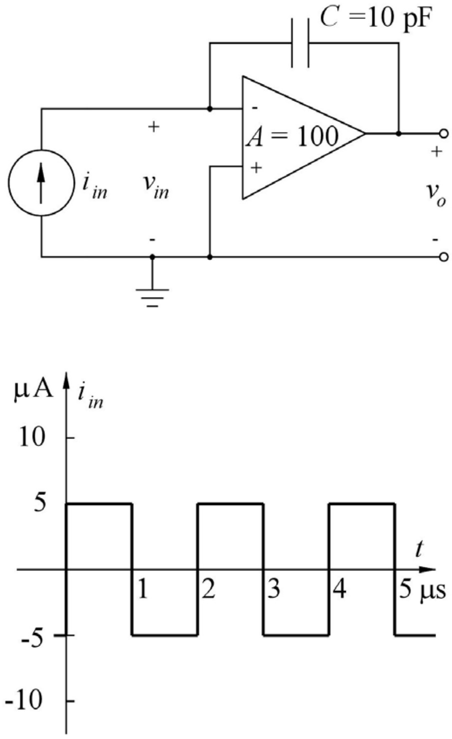


Figure P2.2

For the circuit shown in fig. P2.2, 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. Which initial value of the input voltage v_{in} will result in a mean value of 0 V for the output voltage v_o?

2.3

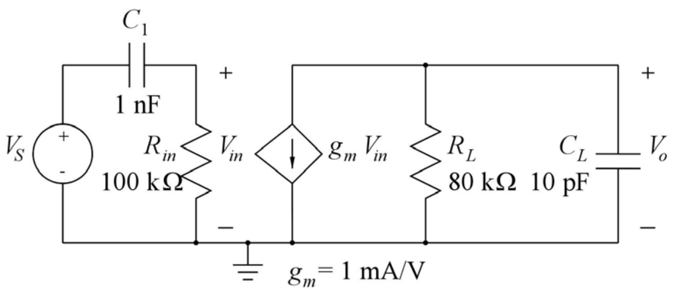


Figure P2.3

For the circuit shown in fig. P2.3, 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.

2.4

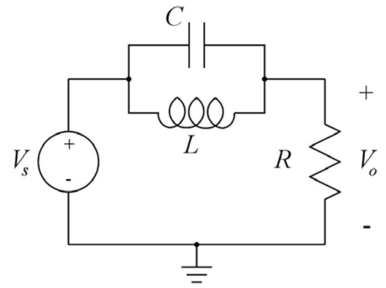


Figure P2.4

For the notch filter shown in fig. P2.4, plot V_o versus frequency in a frequency range showing the notch and the 3-dB bandwidth. Assume $L = 1\ \mu\text{H}$, $C = 5$ pF and $R = 10\ \text{k}\Omega$. From the plot, find the notch frequency, the bandwidth and the quality factor Q.

2.5

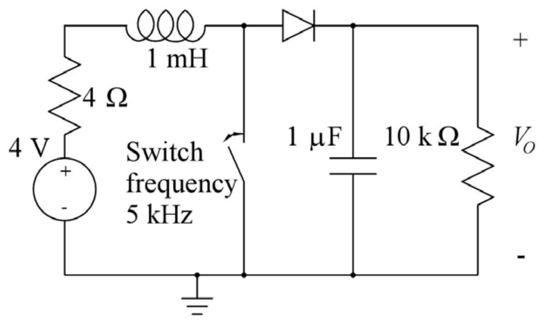


Figure P2.5

The circuit shown in fig. P2.5 is a DC–DC converter which converts 4 V DC 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.

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: $v_{in} = -2.5$ mV.

2.3: Midband gain: 38 dB. Lower half-power frequency: 1.59 kHz. Upper half-power frequency: 199 kHz. 3-dB bandwidth: 197.4 kHz.

2.4: Notch frequency: 71.2 MHz, bandwidth: 3.18 MHz, quality factor $Q = 22.4$.

2.5: $V_O(t \to \infty) = 51$ V. Rise time: 7.5 ms.

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 analysis.
- estimate basic transistor parameters from a DC operating point analysis.
- Simulate transistor small signal parameters using a DC Transfer ('.tf') 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.

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 mouse 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.

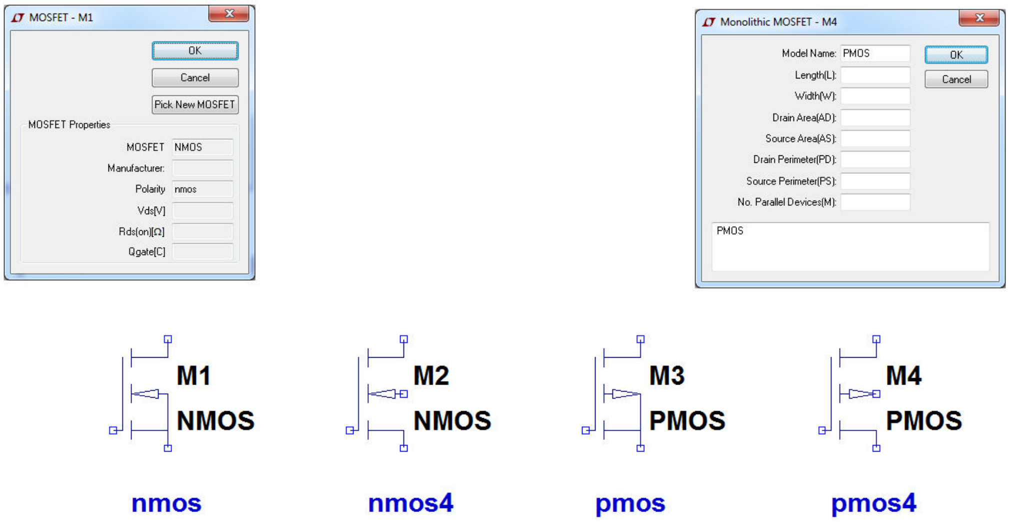


Figure 3.1: MOS transistor symbols and specification windows in LTspice.

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

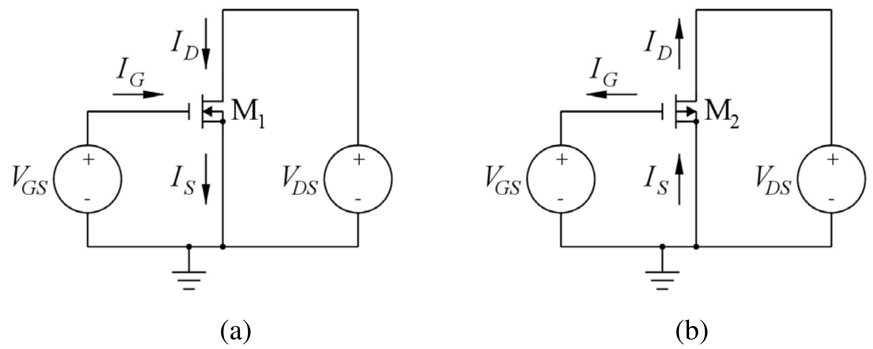


Figure 3.2: Normal textbook definitions of sign conventions for transistor currents and voltages. (a) NMOS transistor. (b) PMOS transistor.

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 model (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 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.

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

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

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

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

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

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

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 ($V_{GS} \leq V_t$), triode region ($0 \leq V_{DS} \leq V_{GS} - V_t$) and active region ($0 \leq V_{GS} - V_t \leq V_{DS}$), 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.

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

$$\text{Cut-off region:} \quad V_t \leq V_{GS} \ \ (\text{or } |V_{GS}| \leq |V_t|). \qquad (3.5)$$

$$\text{Triode region:} \quad V_{GS} - V_t \leq V_{DS} \leq 0 \ \ (\text{or } 0 \leq |V_{DS}| \leq |V_{GS} - V_t|). \qquad (3.6)$$

$$\text{Active region:} \quad V_{DS} \leq V_{GS} - V_t \leq 0 \ \ (\text{or } 0 \leq |V_{GS} - V_t| \leq |V_{DS}|). \qquad (3.7)$$

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 numeric values of V_{GS}, V_{DS} and V_t are used.

LTspice conventions: Fig. 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`, `Vto`, `Lambda`, `Gamma` and `Phi`, respectively. The models are named NMOS-SH and PMOS-SH, respectively, and the parameters are representative for a 0.35 μm CMOS process (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

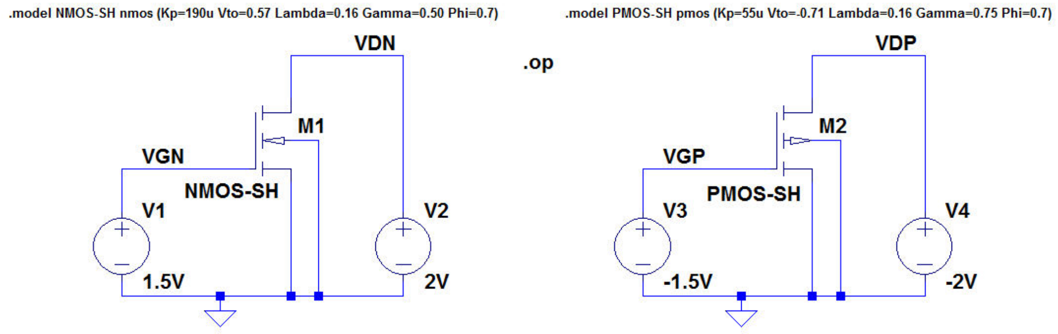


Figure 3.3: LTspice schematic for the circuits from fig. 3.2.

operating point analysis is specified by the '.op' command. 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 visible on the schematic. To see this specification, 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.

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.

```
SPICE Netlist
* M:\LTspice\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 PMOS035 l=1u w=10u
V4 VDP 0 -2V
.model NMOS NMOS
.model PMOS PMOS
.lib C:\PROGRA~\LTC\LTSPIC~1\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.

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 (3.3) and (3.4), we find the following small signal parameters:

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

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

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

Corresponding to (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

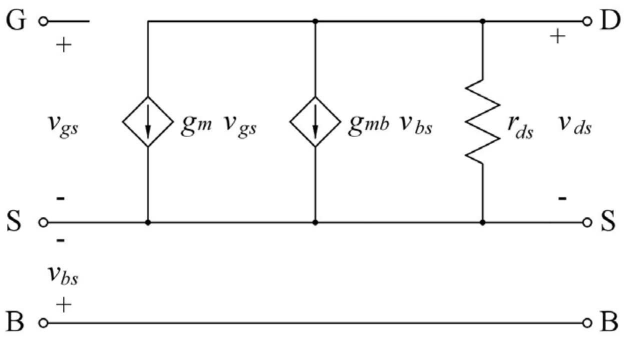


Figure 3.5: Low frequency small-signal MOS transistor model.

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.

```
SPICE Error Log

Circuit:  * M:\LTspice\Tutorial03\Fig3_06.asc

Instance "m2":  Length shorter than recommended for a level 1 MOSFET.
Instance "m2":  Width narrower than recommended for a level 1 MOSFET.
Instance "m1":  Length shorter than recommended for a level 1 MOSFET.
Instance "m1":  Width narrower 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.

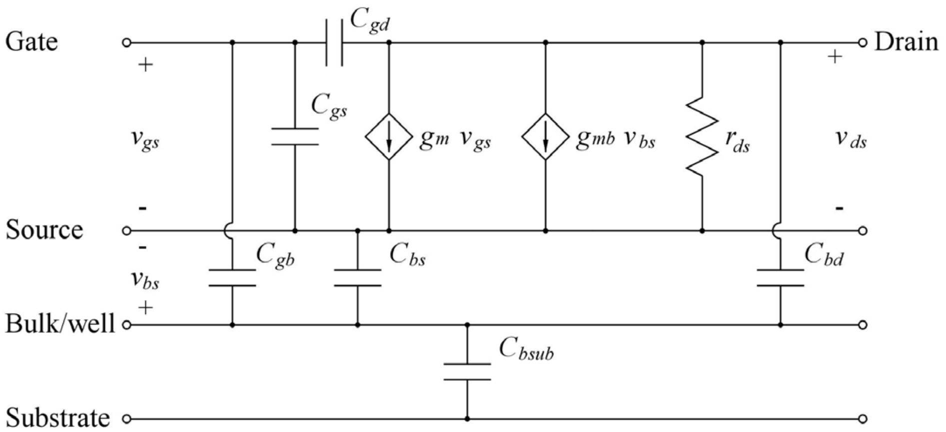


Figure 3.7: High frequency small-signal MOS transistor model.

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

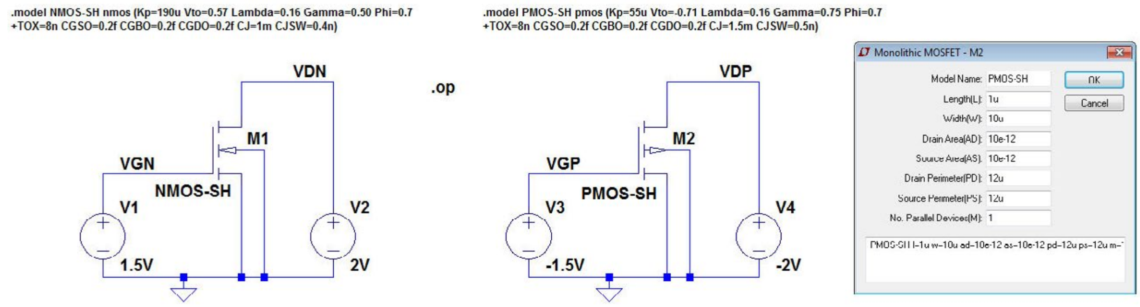


Figure 3.8: LTspice schematic including model parameters for capacitances for the circuits from fig. 3.2.

'LTspice Help', 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 (Carusone, Johns & Martin 2012, chapter 1.5).

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 2011, 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

```
SPICE Netlist

* M:\LTspice\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:\PROGRA~\LTC\LTSPIC~1\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=0.2f CGBO=0.2f CGDO=0.2f 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=0.2f CGBO=0.2f CGDO=0.2f CJ=1.5m CJSW=0.5n)
.backanno
.end
```

```
SPICE Error Log

Circuit:  * M:\LTspice\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.
Instance "m2":  Width narrower 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.
Instance "m1":  Width narrower 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:            2.00e-21    2.00e-21
Cgdov:            2.00e-21    2.00e-21
Cgbov:            2.00e-22    2.00e-22
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 must be applied. Generic models are available together with several textbooks such as (Sedra & Smith 2011), (Carusone, Johns & Martin 2012) and (Baker 2010). For submicron processes BSIM3 or BSIM4 models are often the preferred choice.

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 2014). Most foundries require non-disclosure agreements in order to provide detailed design information, but MOSIS does provide model files extracted from specific wafer lots, e.g. (The MOSIS Service: Wafer Electrical Test Data and SPICE Model Parameters 2014).

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 (Carusone, Johns & Martin 2014).

As an example, fig. 3.10 shows BSIM3 models derived from (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 the folder with the LTspice program, '<LTspiceIV> \lib\sub'. Alternatively, the full path to the file may be specified. The file shown in fig. 3.10 is adapted from (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'.

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. 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. 3.11 which can be compared to fig. 3.9 on page 76.

You may observe that there are significant differences between the small-signal parameters shown in fig. 3.9 and fig. 3.11 and it can be difficult to reach a reasonable degree of compliance between simulated results and results based on a calculation from the Shichman-Hodges model. For comparison, (3.8) and (3.9) gives $g_m = 2.33$ 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. 3.11.

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:\LTspice\Tutorial03\Fig3_11.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:\PROGRA~\LTC\LTSPIC~1\lib\cmp\standard.mos
.op
include BSIM3_035.lib
.backanno
.end
```

```
SPICE Error Log

Circuit:  * M:\LTspice\Tutorial03\Fig3_11.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 3.11: 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. Fig. 3.12 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 on page 68, but as this specification does not by default appear on the schematic, it has been inserted using the command 'Edit → Text', toolbar symbol *Aa* or hotkey 'T'.

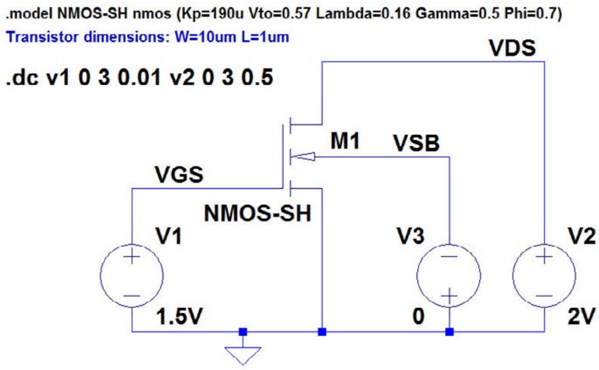


Figure 3.12: LTspice schematic for simulation of transistor characteristics.

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 transistor. In circuits with more transistors, it is highly advisable to show the transistor specification in a way that annotates to the netlist. On page 106, 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.

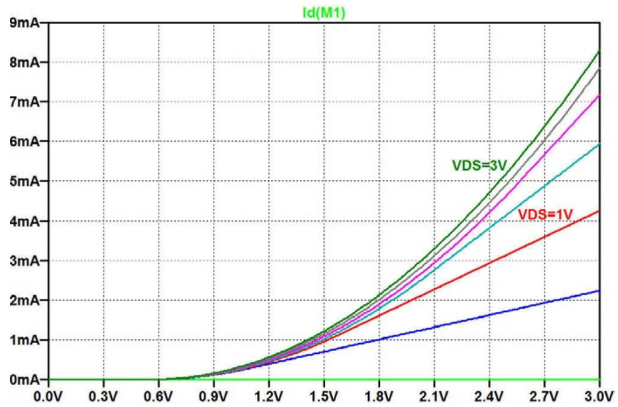


Figure 3.13: Plot of I_D versus V_{GS} for different values of V_{DS} and $V_{SB} = 0$ V.

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' command 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. 3.13.

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 $\rightarrow$ Notes & Annotations $\rightarrow$ Place Text'. 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. 2.3 on page 46, a cursor can be activated by a left mouse 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 mouse click when the mouse is positioned over the cursor will open 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. 2.3 (page 46).

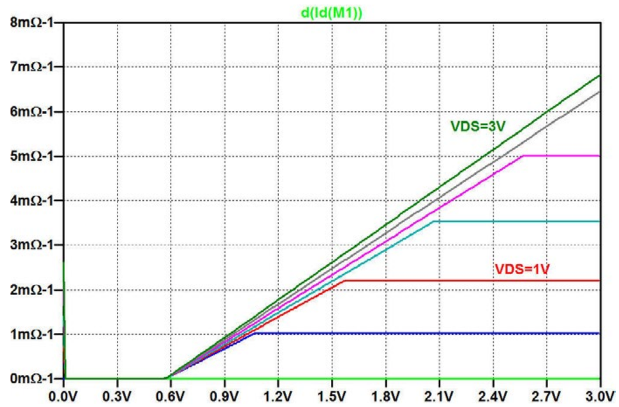


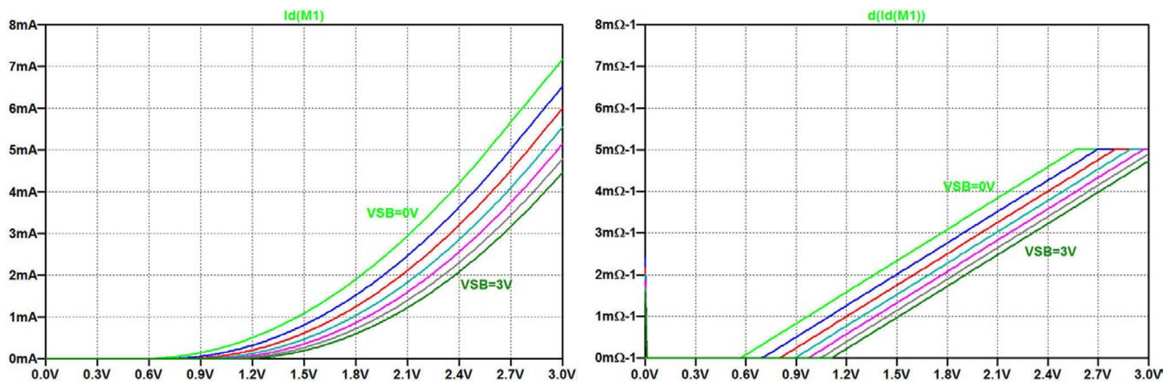
Figure 3.14: Plot of $\partial i_D / \partial v_{GS}$ versus V_{GS} for different values of V_{DS} and $V_{SB} = 0$ V.

Next, fig. 3.14 shows the derivative of I_D versus V_{GS}. This is obtained from the same simulation, just by editing the variable to be plotted: A right mouse 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. 3.13, this is '`Id(M1)`'. By modifying this to '`d(Id(M1))`' and clicking 'OK', the plot window shown in fig. 3.14 appears. The function '`d()`' computes a difference based derivative, see the LTspice 'Help' function (Waveform Arithmetic).

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

$$g_m = \mu_n C_{ox} \left(\frac{W}{L} \right) V_{DS}(1 + \lambda V_{DS}) \tag{3.11}$$

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. 3.14.

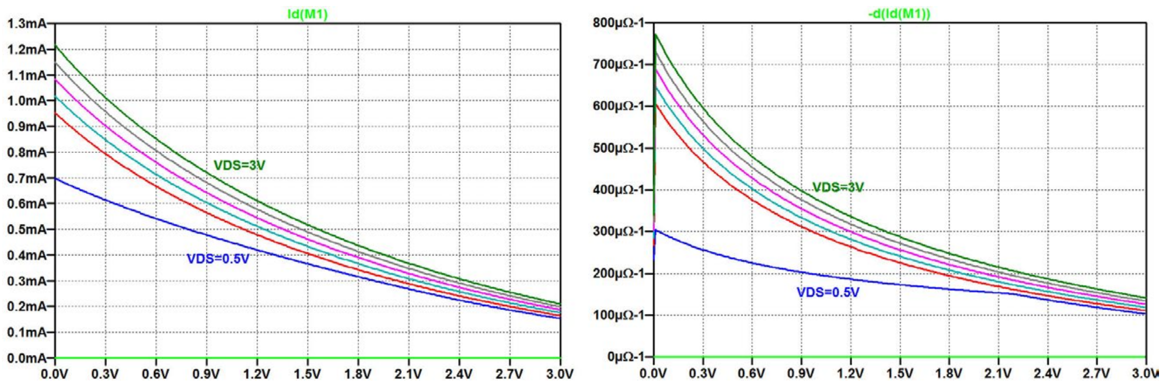


(a) (b)

Figure 3.15: 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.

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 (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. Fig. 3.15 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 obvious.

As indicated by (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}



(a) (b)

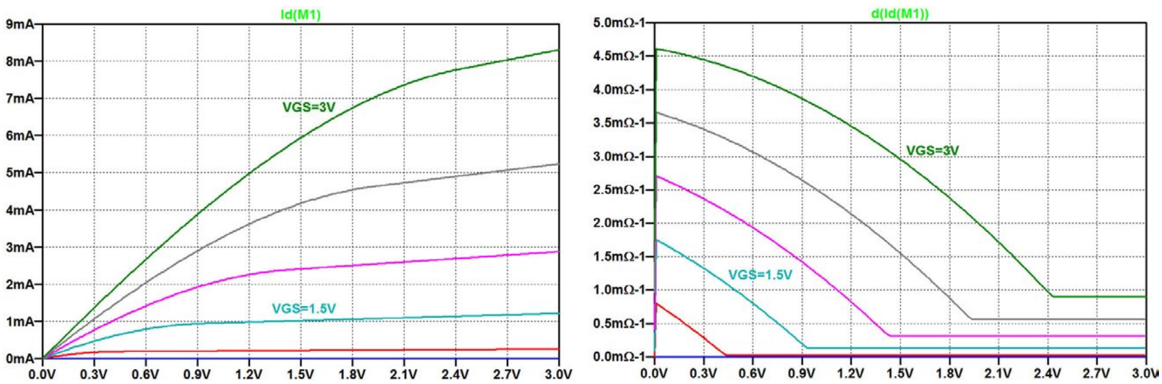
Figure 3.16: Plot I_D (a) and $\partial i_D/\partial v_{BS}$ (b) versus V_{SB} for different values of V_{DS} and $V_{GS} = 1.5$ V.

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. Fig. 3.16 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 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 page 72.

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. 3.12 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. 3.17(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



(a) (b)

Figure 3.17: Output characteristics: plot 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.

slope of the characteristics in the active region caused by the channel length modulation specified by the non-zero value of λ, see (3.3). This causes a non-zero value of g_{ds} which may be plotted as `d(Id(M1))`, see fig. 3.17(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 expected from (3.9).

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. Fig. 3.18 shows the schematic for simulating the characteristics of a PMOS transistor. Observe that the voltages have been reversed compared to fig. 3.12. Fig. 3.19 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))`.

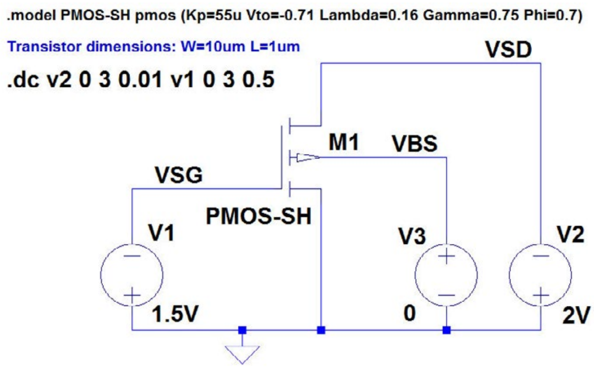
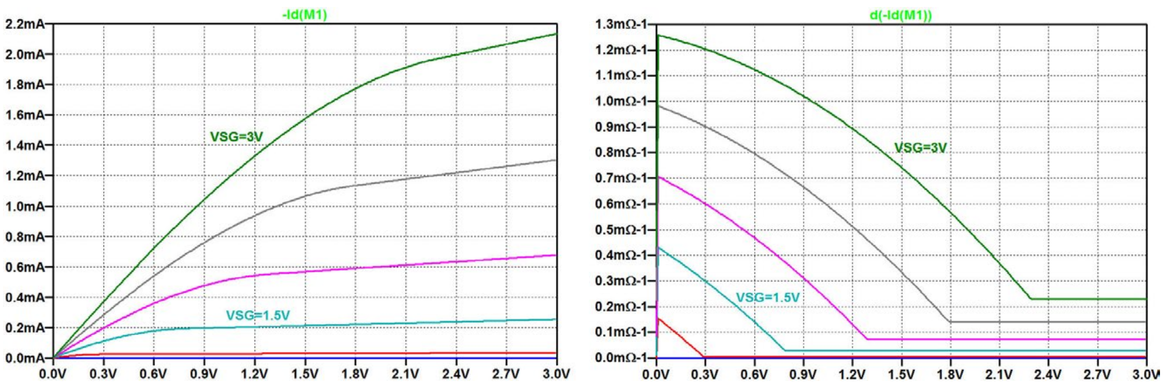


Figure 3.18: LTspice schematic for simulation of PMOS transistor characteristics.



(a) (b)

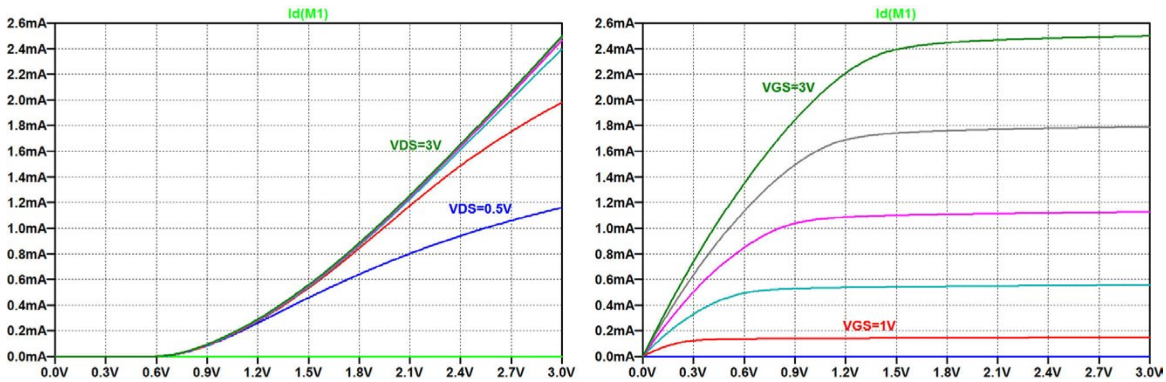
Figure 3.19: Plot I_D (a) and $\partial i_D/\partial v_{SD}$ (b) versus V_{SD} for different values of V_{SG} and $V_{BS} = 0$ V for the PMOS transistor from fig. 3.18.

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. 3.20.

Clearly, there are major differences between the curves in fig. 3.20 and figs. 3.13 and 3.17. 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 $V_{GS} - V_t$ (Carusone, Johns & Martin 2012). The mobility degradation causes the drain current to increase almost linearly with $V_{GS} - V_t$, rather than the square-law relation predicted by (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. 3.20 resembles the shape in fig. 3.17, but apparently with a significantly higher output resistance in the active region, corresponding to a smaller value of the channel length parameter λ.



(a) (b)

Figure 3.20: 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.

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. 3.20 is obtained, leading to somewhat better results when performing hand calculations during a circuit design.

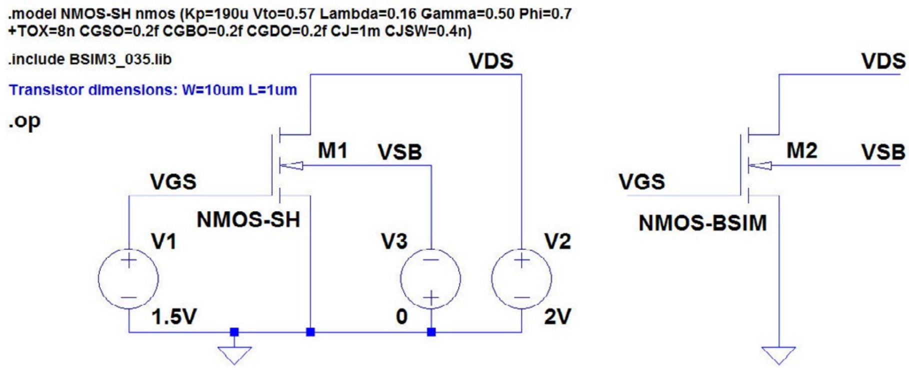


Figure 3.21: LTspice schematic for simulation of transistor characteristics for two different transistor models.

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. 3.21. This figure shows the circuit from fig. 3.12 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.

Matching small signal parameters: If the transistor is known to operate with small signal variations from a specific bias point, a simple approach to find 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. 3.21, 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. 3.22, and the error log file displayed when entering 'Ctrl-L' provides the information about the small signal parameters also shown in fig. 3.22.

```
Output file:
       - - - Operating point - - -

V(vds):      2              voltage
V(vgs):      1.5            voltage
V(vsb):      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:       2.00e-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 3.22: 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. 3.21.

From the small signal parameters, the transistor parameters may be calculated as follows, using (3.9), (3.8) and (3.3):

$$\lambda = \frac{g_{ds}}{I_D - g_{ds}V_{DS}} \tag{3.12}$$

$$V_t = V_{GS} - \frac{2\,I_D}{g_m} \tag{3.13}$$

$$K_p = \frac{g_m}{(W/L)(V_{GS}-V_t)(1+\lambda V_{DS})} = \left(\frac{g_m}{I_D}\right)^2 \times \frac{I_D - g_{ds}V_{DS}}{2(W/L)} \tag{3.14}$$

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

$$\gamma \;=\; 2\sqrt{|2\Phi_F|} \times \frac{g_{mb}}{g_m} = 1.67 \times \frac{g_{mb}}{g_m} \tag{3.15}$$

For the simulation results from fig. 3.22, this results in $\lambda = 0.0195\ \mathrm{V}^{-1}$, $V_t = 0.395$ V, $K_p = 86.4$ μA/V^2 and $\gamma = 0.44\ \sqrt{\mathrm{V}}$.

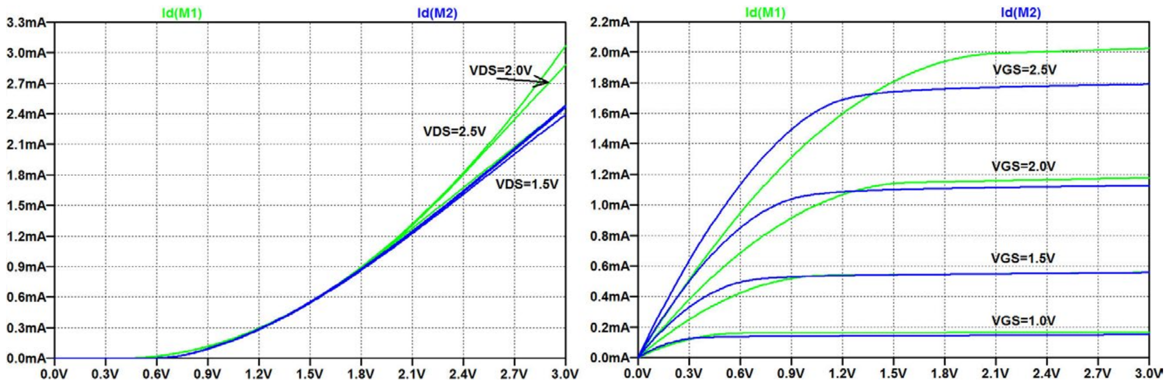
With these model parameters inserted for the model NMOS-SH, the '.op' analysis gives the operating point information shown in fig. 3.23. Obviously, there is a close match between the bias point information and small signal parameters for `M1` and `M2`.

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. 3.24, showing simulated input characteristics and output characteristics for both transistors in the same plot. For simplicity, the input characteristics are shown only for $V_{DS} = 1.5$ V, 2.0 V and 2.5 V, and the output characteristics are shown only for $V_{GS} = 1.0$ V, 1.5 V, 2.0 V 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.

```
Output file:

        - - - Operating point - - -

V(vds):      2              voltage
V(vgs):      1.5            voltage
V(vsb):      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:       nmos-sh                  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 3.23: Operating point information from output file and from SPICE Error Log with adjusted model parameters for the Shichman-Hodges model (transistor M1).



(a) (b)

Figure 3.24: 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).

Example 3.6: Simulating small signal parameters using the '.tf' simulation.

In the previous examples, the small signal parameters g_m and g_{ds} have been simulated directly as derivatives of the drain current, figs. 3.14 and 3.17. 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 (3.8) and (3.9), but for the BSIM models,

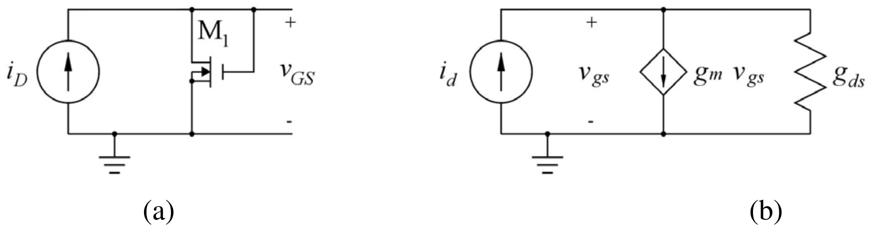


Figure 3.25: Diode connected NMOS transistor (a) and small signal diagram for the diode connected transistor (b).

these equations are only fairly rough approximations. 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' command, 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. 3.25. 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 $1/g_m$.

For finding g_{ds}, we may connect an additional transistor to form a current mirror. Fig. 3.26 shows the LTspice schematic with the current mirror output transistor `M2`. This is connected to a controlled

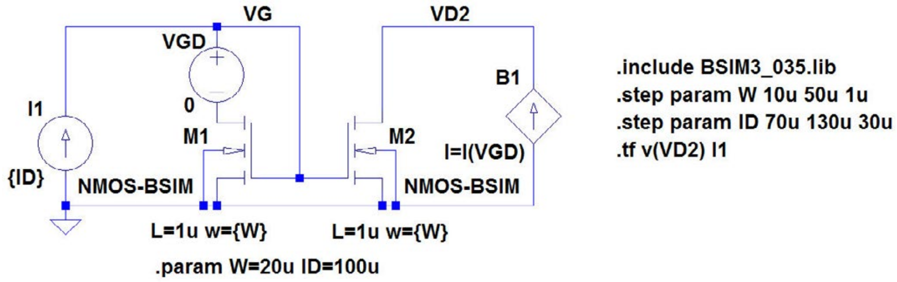
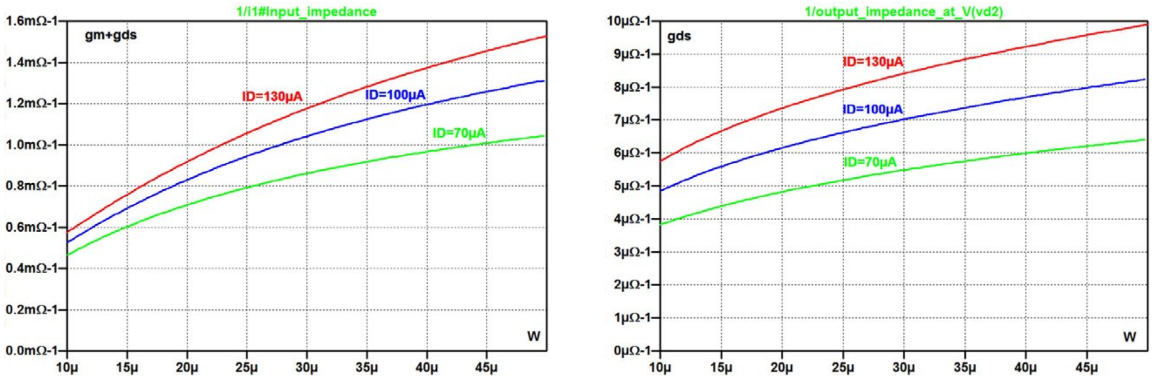


Figure 3.26: NMOS current mirror for simulating both g_{ds} and $g_m + g_{ds}$.

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.

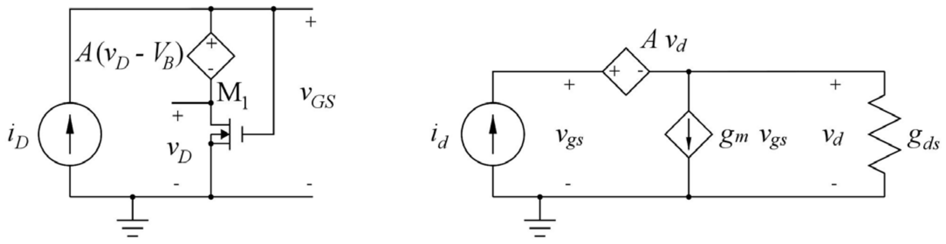


(a) (b)

Figure 3.27: Simulation of $g_m + g_{ds}$ (a) and g_{ds} (b) versus W for different levels of bias current.

For the circuit, you may specify several design variables as parameters and you may include multiple '.step' commands. The '.step' commands 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' command determines the x-axis in the plot window. In fig. 3.26, both transistor channel width W and bias current I_D are specified as parameters, and the '.step' command for W is inserted first. Fig. 3.27 shows the results of a '.tf' simulation for finding $g_m + g_{ds}$ and g_{ds}.

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.



(a) (b)

Figure 3.28: Diode connected NMOS transistor with a voltage buffer for the drain voltage (a) and small signal diagram for the diode connected transistor (b).

We may specify the value of ‘VGD’ to be different from 0. In fig. 3.26, a value of ‘VGD’ larger than V_t will bias the transistor in the linear 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. In order to obtain a constant drain voltage, the diode connected transistor may be modified as shown in fig. 3.28. 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 $Ag_m \gg g_{ds}$.

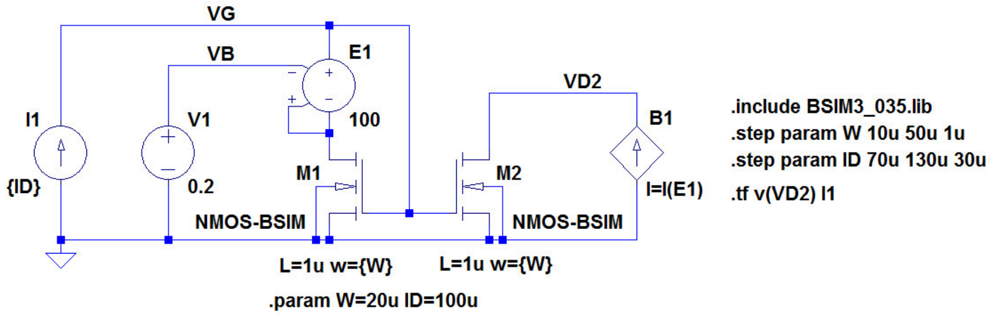
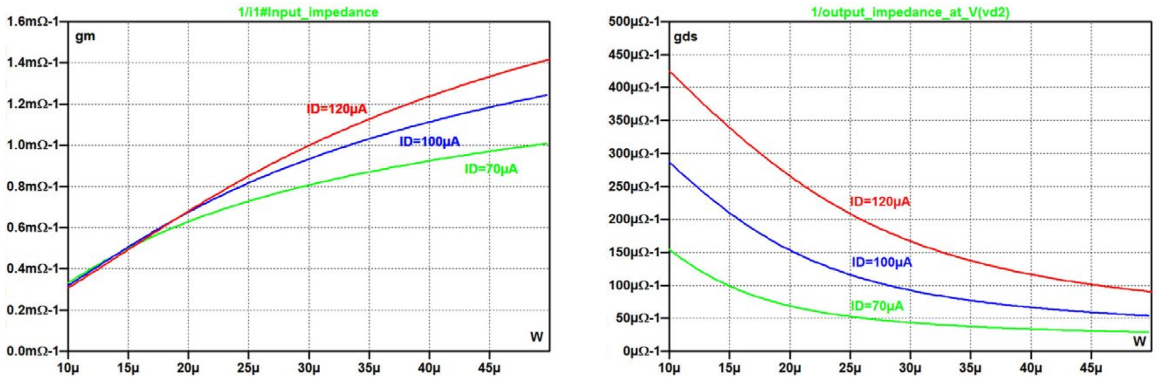


Figure 3.29: NMOS current mirror with voltage buffer for the drain voltage for simulating both g_{ds} and g_m.

Fig. 3.29 shows the LTspice schematic corresponding to the schematic from fig. 3.26 augmented with the voltage buffer shown in fig. 3.28.

As an example, fig. 3.30 shows a simulation of g_m and g_{ds} with a small value of V_B 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 directly g_m as the reciprocal of the input impedance (for $A \gg 1$).



(a) (b)

Figure 3.30: Simulation of g_m (a) and g_{ds} (b) versus W for different levels of bias current with the transistor in the triode region.

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 on page 68). 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’ commands in a schematic, the x-axis in the simulation plot is determined by the ‘.step’ command appearing first in the netlist. This is the command inserted first in the schematic.
- Text (comments) can be placed in a schematic using the command ‘Edit → Text’, toolbar symbol Aa or hotkey ‘T’.
- Text and other annotations, (e.g. cursor position) can be placed in a simulation plot using the command ‘Plot Settings → Notes & Annotations’.

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.

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

Carusone, TC., 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.

Sedra, AS. & Smith, KC. 2011, *Microelectronic Circuits*, International Sixth 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.

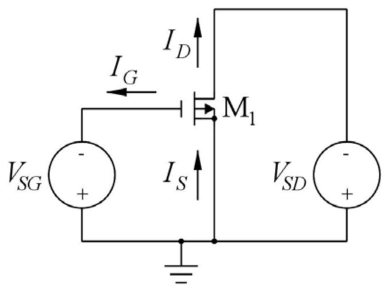
The MOSIS Service 2014. Retrieved from http://www.mosis.com/

The MOSIS Service: Wafer Electrical Test Data and SPICE Model Parameters 2014. Retrieved from https://www.mosis.com/pages/Technical/Testdata/tsmc-035-prm

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

Problems

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.

Figure P3.1

For the PMOS transistor shown in fig. P3.1, 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 in the figure. 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.

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
```

Figure P3.2

For an NMOS transistor with the transistor model shown in fig. P3.2 (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, compare page 26.

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
```

Figure P3.3

For a PMOS transistor with the transistor model shown in fig. P3.3 (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.

3.4

For a PMOS transistor with the transistor model shown in fig. P3.3 (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}).

3.5

For an NMOS transistor with the transistor model shown in fig. P3.2 (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.

3.6

$K_p = 190$ μA/V^2, $V_{to} = 0.57$ V, $\lambda = 0.16$ V^{-1},
$\gamma = 0.5\ \sqrt{\text{V}}$, $|2\Phi_F| = 0.7$ V.

Figure P3.6

For an NMOS transistor with the Shichman-Hodges parameters shown in fig. P3.6 and a channel length $L = 1$ 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.

Answers

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

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

3.3: $I_D = 0.16$ mA; $\partial i_D/\partial v_{SD} = 7.3\ \mu$A/V.

3.4: $I_D = 0.16$ mA; $g_m = 0.339$ mA/V; $g_{mb} = 0.0754$ mA/V; $g_{ds} = 7.31\ \mu$A/V;
$\lambda = 0.05\ \text{V}^{-1}$; $V_{to} = -0.557$ V; $K_p = 32.6\ \mu\text{A/V}^2$; $\gamma = 0.37\ \sqrt{\text{V}}$; $|2\Phi_F| = 0.7$ V.

3.5: $g_m = 0.584$ mA/V; $g_{mb} = 0.167$ mA/V; $g_{ds} = 6.04\ \mu$A/V; $\lambda = 0.045\ \text{V}^{-1}$;
$V_{to} = 0.51$ V; $K_p = 117\ \mu\text{A/V}^2$; $\gamma = 0.48\ \sqrt{\text{V}}$; $|2\Phi_F| = 0.7$ V.

3.6: $W = 10\ \mu$m: $I_{D\max} = 1.62$ mA; $W = 30\ \mu$m: $I_{D\max} = 4.89$ mA; $W = 50\ \mu$m: $I_{D\max} = 8.13$ mA.

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 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.

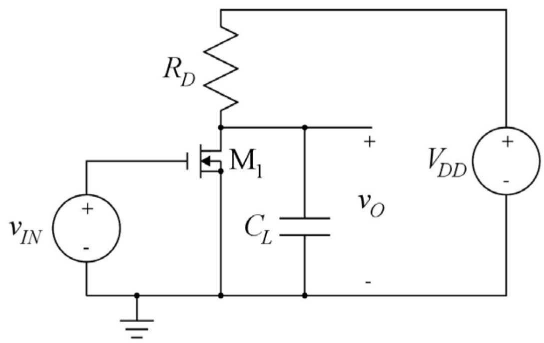


Figure 4.1: NMOS common source amplifier with drain resistor.

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 on page 75. 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:

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

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}, this results in

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

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 $f_0 = 1/(2\pi R_D C_L) \geq$

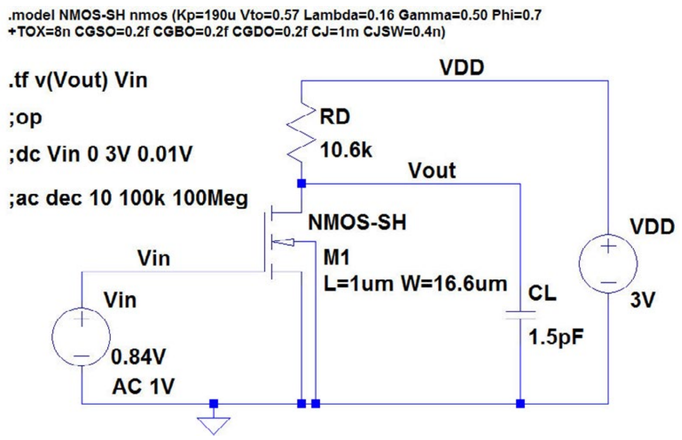


Figure 4.2: LTspice schematic for simulating the common source stage with the Shichman-Hodges transistor model.

10 MHz. From this,

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

With this value of R_D, the bias current I_D is $I_D = V_{DD}/(2R_D) = 142\ \mu\mathrm{A}$ and the transistor transconductance is $g_m = 0.94$ 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. Fig. 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 on page 75.

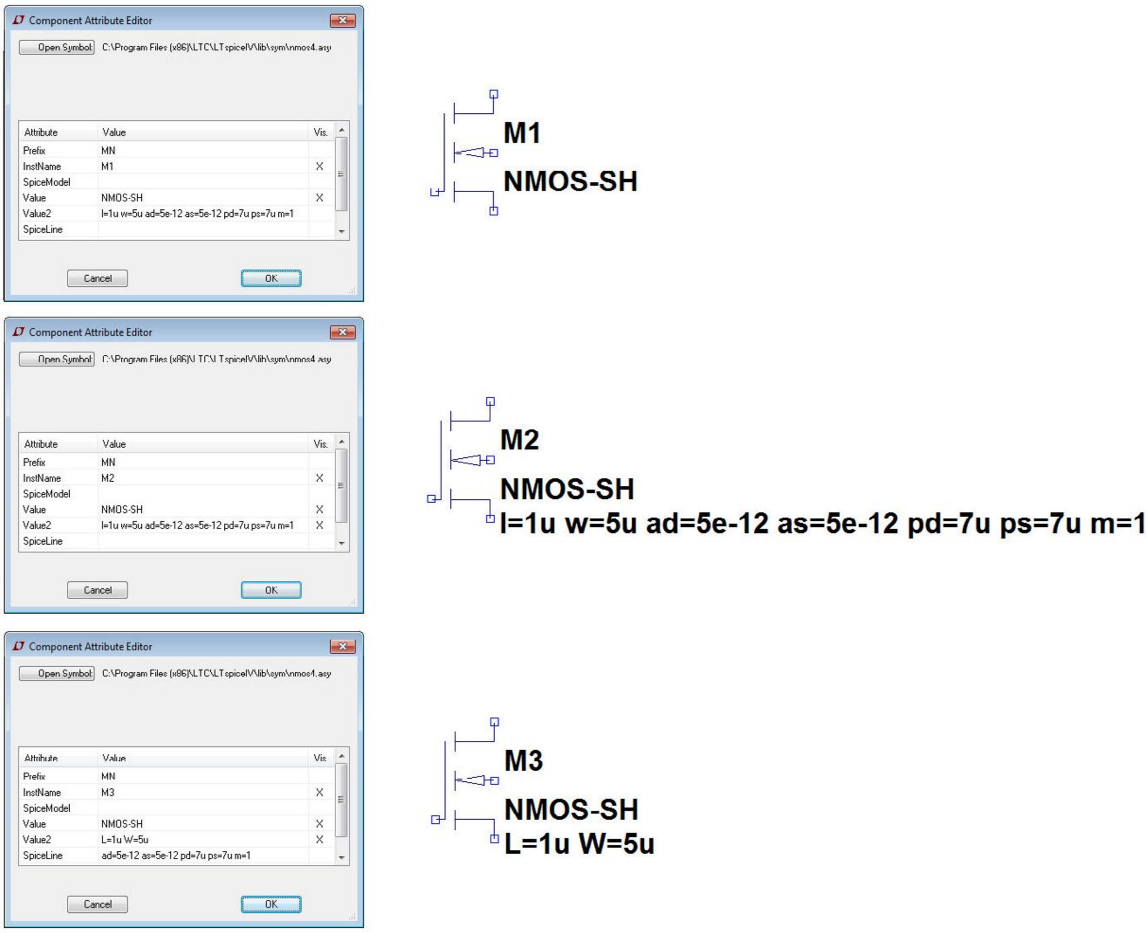


Figure 4.3: LTspice transistor symbols with different numbers of visible transistor parameters.

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 in 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 explained on page 75, 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. You may notice that the transistor specification in the netlist is the same for the three transistors ('View → SPICE Netlist').

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.

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 are then edited just by moving the cursor over the text and right clicking. If you use the specification window shown in fig. 3.8 (page 75), all the parameters which are entered in the specification window are placed in the line 'Value2' and will be visible in the schematic.

Iterativ design of the transistor channel width: In order to verify the calculations from (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 in order to have 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 lowers 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 fulfil 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 (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 fulfils 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 on page 77. 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 signal 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 (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 iteration 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. Here, a PMOS current mirror

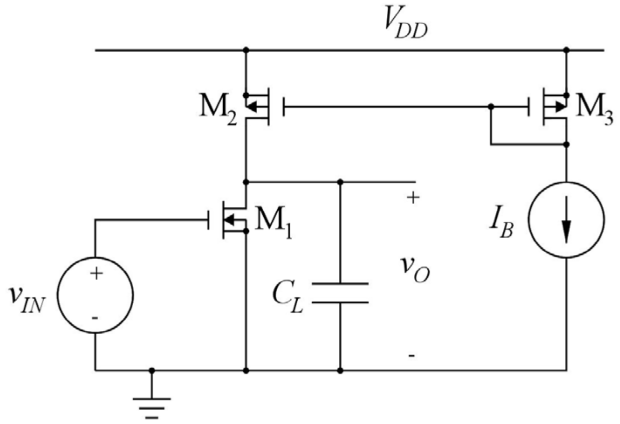


Figure 4.4: NMOS common source amplifier with PMOS load.

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 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. Fig. 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. Fig. 4.7 shows the error log files with the transistor bias voltages and currents and small signal parameters corresponding to both fig. 4.5 and fig. 4.6. The small signal parameters are identical, but the gate-source voltages and drain source voltages are different since drain and source are interchanged. In general, the drawing standard of fig. 4.5 is recommended, even though it requires more manipulations in terms of rotation and mirroring since it gives output results which are directly comparable to the results obtained from standard hand calculations.

You should also notice that just drawing a wire directly from the bulk terminal of a transistor to the source terminal (the way it is done in fig. 4.1 and fig. 4.4) does not establish a connection between bulk and source. This is why the bulk connections in figs. 4.5 and 4.6 are drawn explicitly to ground and V_{DD}. Fig. 4.8 shows a few ways to draw connections between bulk and source. Pay attention to the incorrect drawings of M_4, M_5, M_9 and M_{10}. Also see the resulting netlist specification shown to the right in the figure. You can always check the connections by examining the netlist ('View →

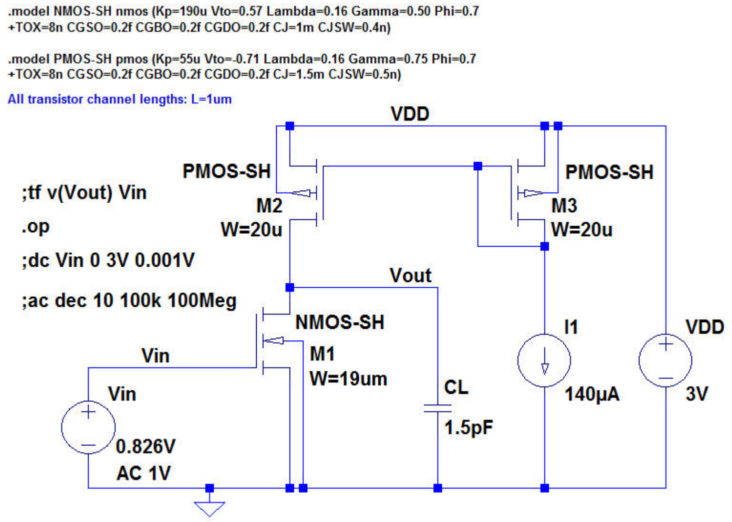


Figure 4.5: LTspice schematic for simulating the common source stage with a PMOS active load, drawn with correct polarity of PMOS transistors.

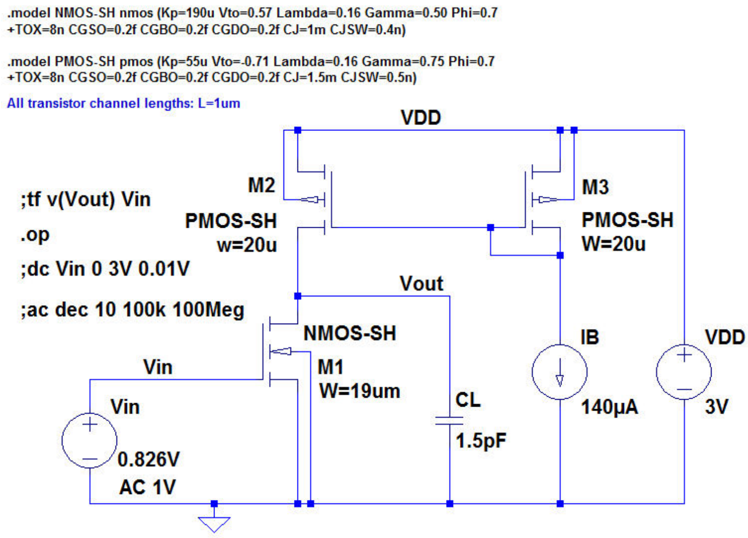


Figure 4.6: LTspice schematic for simulating the common source stage with a PMOS active load, drawn with inverted polarity of PMOS transistors.

SPICE Netlist') where the syntax for a mos transistor is:
'`Mxx drain_node gate_node source_node bulk_node model_name layout_parameters`'.

For fig. 4.5, the size of M_1 has been selected to be 19 µm as for the configuration with a drain resistor (and simulated with the Shichman-Hodges transistor model) in order to compare the performance of the two circuits. Also, the bias current has been selected to be the same. The PMOS transistors

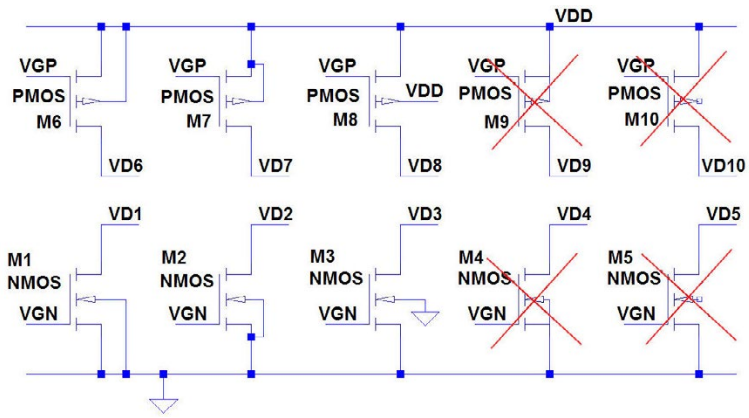
```
SPICE Error Log

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    1.62e-14
Cbs:           4.10e-14    4.10e-14    2.74e-14
Cgsov:         4.00e-21    4.00e-21    3.80e-21
Cgdov:         4.00e-21    4.00e-21    3.80e-21
Cgbov:         2.00e-22    2.00e-22    2.00e-22
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:         4.00e-21    4.00e-21    3.80e-21
Cgdov:         4.00e-21    4.00e-21    3.80e-21
Cgbov:         2.00e-22    2.00e-22    2.00e-22
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
```

(b): Inverted polarity of PMOS transistors

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




```
SPICE Netlist

M1 VD1 VGN 0 0 NMOS
M2 VD2 VGN 0 0 NMOS
M3 VD3 VGN 0 0 NMOS
M4 VD4 VGN 0 N001 NMOS
M5 VD5 VGN 0 NC_01 NMOS
M6 VD6 VGP VDD VDD PMOS
M7 VD7 VGP VDD VDD PMOS
M8 VD8 VGP VDD VDD PMOS
M9 VD9 VGP VDD N002 PMOS
M10 VD10 VGP VDD NC_02 PMOS
```

Figure 4.8: Drawing connections between bulk and source for NMOS and PMOS transistors.

have (somewhat arbitrarily) been selected to have a width of 20 µm. A large value of W results in a small overdrive voltage $|V_{GS} - V_{to}|$ giving a large output voltage range for the gain stage. With $W = 20$ µm, the overdrive voltage for M_2 and M_3 is (neglecting the channel length modulation)

$$|V_{GS} - V_{to}| = \sqrt{\frac{2I_B}{\mu_p C_{ox}(W/L)}} = 0.5 \text{ V} \tag{4.6}$$

giving a maximum output voltage of 2.5 V with M_2 in saturation.

Comparing the simulation results of the circuit in fig. 4.5 with the simulations results of the circuit in fig. 4.2 with W adjusted to 19 µm, you will recognize that now the gain is larger and the output resistance is higher. The '.tf' simulation results in $A_{voc} = -30.3$ V/V and $r_o = 26,5$ kΩ. The reason for this is that in the expression for the gain, R_D is now replaced by the small signal output resistance of M_2, and the total small signal output resistance is the parallel combination of the output resistance of M_1 and M_2. They are both caused by the transistor channel length modulation which is not a well controlled transistor parameter. This implies that for the common source configuration with an active load, a design target for the 3 dB bandwidth cannot be met with a reasonable precision. The '.ac' simulation of fig. 4.5 shows a 3 dB bandwidth of 3.9 MHz rather than 11.6 MHz as found on page 108. Instead, it is the product of gain and bandwidth which is a relevant design target. It is given by $2\pi \times GBW = g_{m1}/C_L$, and both g_{m1} and C_L are reasonably well controlled. For the common source stage with a drain resistor as in fig. 4.2, the gain bandwidth product is 10×11.6 MHz = 116 MHz. For the common source stage with an active load as in fig. 4.5, the gain bandwidth product is 30.3×3.9 MHz=118 MHz, i.e. very close to the value obtained for the circuit with a drain resistor.

Next, the circuit may be simulated using the BSIM3 transistor models. With the same transistor geometries, the simulations result in a gain of -60.3 V/V and a bandwidth of 1.57 MHz, giving a gain bandwidth product of 95 MHz which is somewhat smaller than with the Shichman-Hodges model. As for the configuration with a drain resistor, the design may be modified by increasing W_1, and using the same value for W_1 as for the configuration with a drain resistor (i.e. $W_1 = 22$ µm), we find a gain of -65.2 V/V and a bandwidth of 1.60 MHz, resulting in a gain bandwidth product of 104 MHz. For the configuration with resistive load and the same geometry for M_1 (i.e. $W_1 = 22$ μm), we found a gain bandwidth product of 10.0×10.6 MHz = 106 MHz, so the gain bandwidth product is the same for the two designs.

A word of caution: For the simulations described here, it is important to start with a DC sweep in order to find the bias value for V_{IN} resulting in an output voltage of 1.5 V. Due to the large gain of the common source configuration with an active load, even a very small error in the DC value specified for V_{IN} will result in an operating point which is way off from the desired operating point. Always check the operating point by a '.op' simulation to see that bias voltages and currents and small signal parameters have reasonable values before running '.ac' simulations and '.tf' simulations.

Example 4.2: The common drain amplifier (source follower).

The common drain stage (or source follower) is shown in fig 4.9 with an NMOS transistor as the common drain transistor and an NMOS current mirror to bias the common drain stage. The source follower is normally used as a buffer stage. It has a voltage gain close to one and a fairly small output resistance, implying that it can drive a resistive load. For the source follower, it may not be possible to connect the bulk to the source, so the bulk effect will have an impact on the performance, both the small signal gain, the output resistance and the output voltage swing. In this example, we will compare a source follower with source and bulk connected, fig 4.9(a), to a source follower with the bulk connected to the negative supply rail, fig 4.9(b).

For this design, we assume that we need a buffer capable of driving a resistive load of 5 kΩ with a voltage swing from -1.0 V to + 0.5 V. The supply voltages are $V_{DD} = V_{SS} = 1.5$ V. The source follower cannot drive the output very high (assuming a maximum input voltage equal to V_{DD}) because

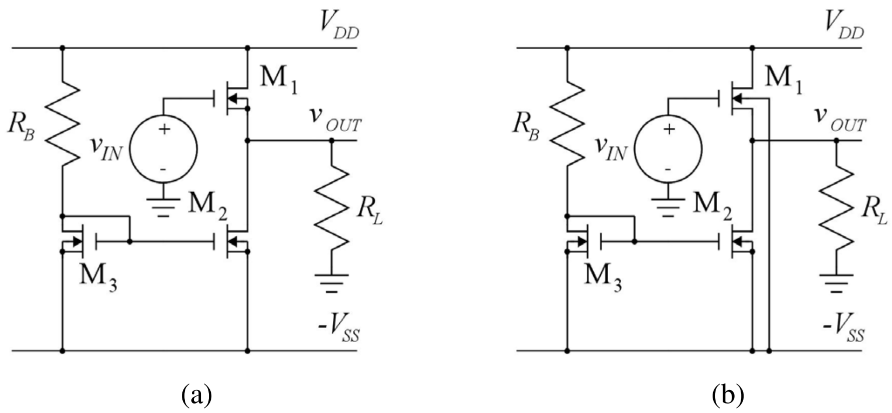


Figure 4.9: Common drain stage. (a) Common drain NMOS with source and bulk connected. (b) Common drain NMOS with bulk connected to negative supply rail.

a gate-source voltage is needed between the input voltage and the output voltage. First, we design the buffer shown in fig. 4.9(a) assuming a Shichman-Hodges transistor model with the parameters from fig. 3.8 on page 75. Next, we take the bulk effect into consideration (fig. 4.9(b)), and finally we use the BSIM3 transistor model from fig. 3.10 on page 77 for an extra design iteration.

The basic design requirements arise from the output swing. In order to have a negative output swing of -1.0 V with a load resistor of 5 kΩ, we need a bias current in M_2 of at least 1 V/5 kΩ = 0.2 mA. As M_2 is configured as a current source, it should be in the active area for this bias current, implying that the overdrive voltage $V_{GS2}-V_{to}$ must be at most 0.5 V in order ensure $V_{DS2} \geq V_{GS2}-V_{to}$. Neglecting the channel length modulation, we may calculate a minimum value of (W_2/L_2) from

$$\frac{W_2}{L_2} = \frac{2\,I_{D2}}{\mu_n C_{ox}(V_{GS2}-V_{to})^2} = 8.4 \tag{4.7}$$

For an initial design, we select $L_2 = 1$ µm and $W_2 = 10$ µm. For M_3, we may use a smaller width, scaling down the current in M_3 and R_B compared to the current in M_3 in order to reduce the current consumption of the bias circuit. Using a scale factor of 10, we select $L_3 = W_3 = 1$ µm. The resistor R_B is calculated from V_{GS3}, the supply voltage and the current:

$$R_B = \frac{V_{DD}+V_{SS}-V_{GS3}}{I_{D3}} = \frac{V_{DD}+V_{SS}-V_{to}-\sqrt{\dfrac{2\,I_{D3}}{\mu_n C_{ox}(W_3/L_3)}}}{I_{D3}} = 98.5\ \text{k}\Omega \tag{4.8}$$

Finally, M_1 must be designed so that it can supply 200 µA to M_2 plus 100 µA to R_L (giving an output voltage of 0.5 V) when the gate voltage is the maximum input voltage which is assumed to be V_{DD}. This implies (compare (4.7))

$$\frac{W_1}{L_1} = \frac{2\times 300\ \mu\text{A}}{190\ \mu\text{A}/\text{V}^2(1.5\ \text{V}-0.5\ \text{V}-0.57\ \text{V})^2} = 17 \tag{4.9}$$

With $L_1 = 1$ µA, we select $W_1 = 17$ µm.

Fig. 4.10 shows the schematic corresponding to fig 4.9(a) and with the Shichman-Hodges transistor model for the specification of M_1, M_2 and M_3. For an initial simulation, it may be a good idea to use $\lambda = 0$ in the model description as shown in the figure. In this way, the hand calculations can be verified directly. For verifying the lower output voltage limit, a '.op' simulation with the input at -1.5 V may be run, and for verifying the upper output limit, a '.op' simulation with the input at +1.5 V may be run. Fig. 4.11 shows the output files and the error log files from these simulations. Notice that labels have been placed on all the nodes in the schematic in fig. 4.10, using the command 'Edit → Label Net' (or toolbar symbol). This makes it easier to read the output file from the simulation.

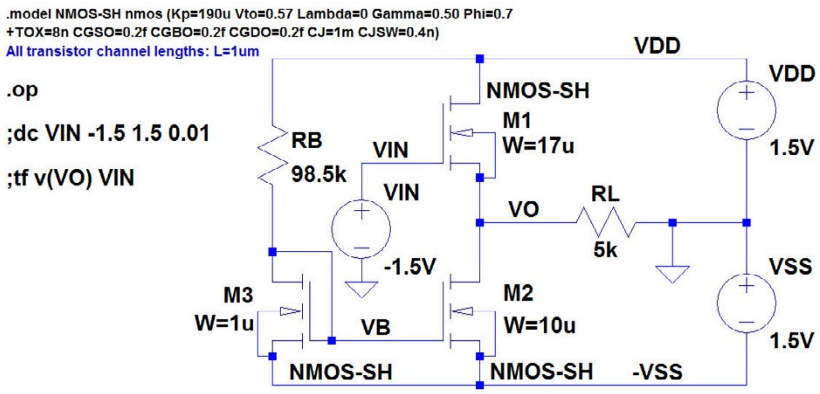


Figure 4.10: LTspice schematic for the common drain stage without bulk effect and with a Shichman-Hodges transistor model with $\lambda = 0$.

Fig. 4.11(a) shows the simulation results for $V_{IN} = -1.5$ V. From the output file, we find $V_O = -1.00$ as required, and from the error log file, we find `Vdsat` = 0.459 V < `Vds` = 0.499 V for M_2, so M_2 is in the active region as required.

Fig. 4.11(b) shows the simulation results for $V_{IN} = +1.5$ V. From the output file, we find $V_O = 0.499$ which is sufficiently close to the calculated value to confirm the hand calculations.

```
Output file:

       - - - Operating point - - -

V(vdd):      1.5            voltage
V(vin):      -1.5           voltage
V(vo):       -1.00053       voltage
V(vb):       -0.471047      voltage
V(-vss):     -1.5           voltage
Id(M3):      2.00106e-005   device_current
Ig(M3):      0              device_current
Ib(M3):      -1.03895e-012  device_current
Is(M3):      -2.00106e-005  device_current
Id(M2):      0.000200106    device_current
Ig(M2):      0              device_current
Ib(M2):      -5.09469e-013  device_current
Is(M2):      -0.000200106   device_current
Id(M1):      5.01106e-012   device_current
Ig(M1):      0              device_current
Ib(M1):      -2.51053e-012  device_current
Is(M1):      -2.50053e-012  device_current
I(Rl):       0.000200106    device_current
I(Rb):       2.00106e-005   device_current
I(Vin):      0              device_current
I(Vss):      -0.000220117   device_current
I(Vdd):      -2.00106e-005  device_current
```

```
SPICE Error Log

Semiconductor Device Operating Points:
                           - - - MOSFET Transistors - - -
Name:              m3          m2          m1
Model:           nmos-sh     nmos-sh     nmos-sh
Id:               2.00e-05    2.00e-04    5.01e-12
Vgs:             1.03e+00    1.03e+00   -4.99e-01
Vds:             1.03e+00    4.99e-01    2.50e+00
Vbs:             0.00e+00    0.00e+00    0.00e+00
Vth:             5.70e-01    5.70e-01    5.70e-01
Vdsat:           4.59e-01    4.59e-01    0.00e+00
Gm:              8.72e-05    8.72e-04    0.00e+00
Gds:             0.00e+00    0.00e+00    0.00e+00
Gmb:             2.61e-05    2.61e-04    0.00e+00
Cbd:             1.46e-15    1.16e-14    1.21e-14
Cbs:             2.20e-15    1.48e-14    2.46e-14
Cgsov:           2.00e-22    2.00e-21    3.40e-21
Cgdov:           2.00e-22    2.00e-21    3.40e-21
Cgbov:           2.00e-22    2.00e-22    2.00e-22
Cgs:             2.88e-15    2.88e-14    0.00e+00
Cgd:             0.00e+00    0.00e+00    0.00e+00
Cgb:             0.00e+00    0.00e+00    7.34e-14
```

(a): output file and error log file with $V_{IN} = -1.5$ V.

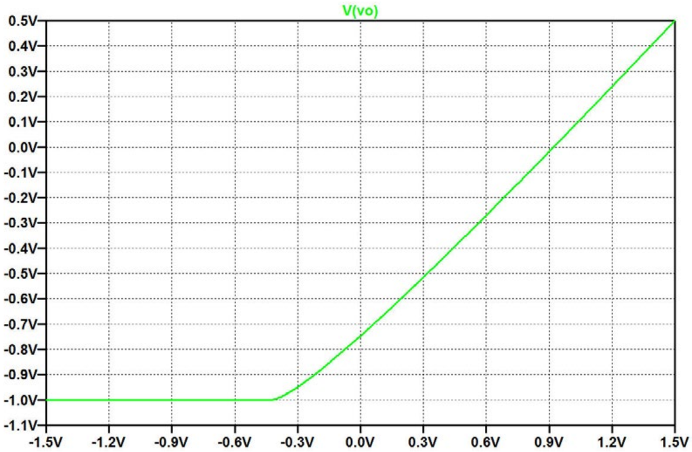
```
Output file:

       - - - Operating point - - -

V(vdd):      1.5            voltage
V(vin):      1.5            voltage
V(vo):       0.499061       voltage
V(vb):       -0.471047      voltage
V(-vss):     -1.5           voltage
Id(M3):      2.00106e-005   device_current
Ig(M3):      0              device_current
Ib(M3):      -1.03895e-012  device_current
Is(M3):      -2.00106e-005  device_current
Id(M2):      0.000200106    device_current
Ig(M2):      0              device_current
Ib(M2):      -2.00906e-012  device_current
Is(M2):      -0.000200106   device_current
Id(M1):      0.000299919    device_current
Ig(M1):      0              device_current
Ib(M1):      -1.01094e-012  device_current
Is(M1):      -0.000299919   device_current
I(Rl):       -9.98123e-005  device_current
I(Rb):       2.00106e-005   device_current
I(Vin):      0              device_current
I(Vss):      -0.000220117   device_current
I(Vdd):      -0.000319929   device_current
```

```
SPICE Error Log

Semiconductor Device Operating Points:
                           - - - MOSFET Transistors - - -
Name:              m3          m2          m1
Model:           nmos-sh     nmos-sh     nmos-sh
Id:              2.00e-05    2.00e-04    3.00e-04
Vgs:             1.03e+00    1.03e+00    1.00e+00
Vds:             1.03e+00     2.00e+00   1.00e+00
Vbs:              0.00e+00   0.00e+00    0.00e+00
Vth:             5.70e-01    5.70e-01    5.70e-01
Vdsat:           4.59e-01    4.59e-01    4.31e-01
Gm:              8.72e-05    8.72e-04    1.39e-03
Gds:              0.00e+00   0.00e+00    0.00e+00
Gmb:             2.61e-05    2.61e-04    4.16e-04
Cbd:             1.46e-15    7.91e-15    1.64e-14
Cbs:             2.20e-15     1.48e-14   2.46e-14
Cgsov:           2.00e-22    2.00e-21    3.40e-21
Cgdov:           2.00e-22    2.00e-21    3.40e-21
Cgbov:           2.00e-22    2.00e-22    2.00e-22
Cgs:             2.88e-15    2.88e-14    4.89e-14
Cgd:              0.00e+00   0.00e+00    0.00e+00
Cgb:             0.00e+00    0.00e+00    0.00e+00
```

(b): output file and error log file with $V_{IN} = +1.5$ V.

Figure 4.11: Output files and error log files for '.op' simulations of the common drain stage from fig. 4.10

For the schematic, you may also run a DC sweep with the input voltage swept from -1.5 V to +1.5 V. The result of this is shown in fig. 4.12, and it is evident that the source follower provides an output voltage in the range -1 V to +0.5 V. The input voltage required to obtain an output voltage of 0 V is found to be $V_{IN} = 0.92$ V. With this value of the input bias voltage, also the result of a '.tf' simulation is shown in fig. 4.12. The small signal gain is found to be 0.85 V/V. Also an output resistance of 749 Ω is listed, but this is not the output resistance of the source follower since the



Output from DC Transfer simulation		
--- Transfer Function ---		
Transfer_function:	0.850299	transfer
vs#Input_impedance:	1e+020	impedance
output_impedance_at_V(vo):	748.505	impedance

Figure 4.12: Result for a DC sweep and from a '.tf' simulation of the circuit from fig. 4.10.

simulation has included the load resistor R_L. In order to find the output resistance r_o and the open circuit voltage gain A_{voc}, the '.tf' simulation must be run without R_L (or with R_L set to a very high value, e.g. 5 GΩ. Running this '.tf' simulation, you will find $A_{voc} = 1$ and $r_o = 880$ Ω. Running a '.op' simulation with $V_{IN} = 0.92$ V (and $R_L = 5$ GΩ), the error log file yields `gm1` = 1.14 mA/V, confirming that $r_o = 1/g_{m1}$ for the source follower when the bulk effect and the channel length modulation is not taken into account.

A re-simulation of the circuit, taking the channel length modulation into account (using $\lambda = 0.16$ V^{-1}) shows that the circuit no longer quite fulfils the requirements concerning the minimum value of the output voltage. This is hardly surprising since M_2 has a smaller drain-source voltage than M_3, so with the channel length modulation taken into account, the scaling of the current mirror $M_3 - M_2$ is smaller than 10. A simple way to compensate for this is to reduce the value of R_B. A new value for R_B may be found by a simple iteration or by running a simulation where the value of R_B is swept over a suitable range, e.g. 90 kΩ to 100 kΩ. A new value of $R_B = 92$ kΩ gives an output voltage range from -1 V to +0.5 V with an input bias voltage of 0.9 V required for an output bias voltage of 0 V. A re-simulation to find r_o and A_{voc} results in $A_{voc} = 0.96$ V/V and $r_o = 705$ Ω, so as expected, the channel length modulation affects both the gain and the output resistance (Carusone, Johns & Martin 2012).

The next step is to include the bulk effect by connecting the bulk of M_1 to the negative supply rail $-V_{SS}$. This will result in an increased value of V_t for M_1, and also the bulk transconductance g_{mb} will affect the gain and the output resistance. The new value of V_t may be calculated from (3.4) on page 70, and g_{mb} may be calculated from (3.10) on page 72. Inserting $\gamma = 0.5\ \sqrt{\mathrm{V}}$ and $|2\Phi_F| = 0.7$ V, we find that the threshold voltage is increased by about 0.4 V for $V_{SB} = 2$ V, so the bulk effect has a strong impact on the upper limit of the output range. A DC sweep simulation shows that the output voltage range is now from −1 V to +0.2 V. The maximum output voltage can be increased by increasing the width of M_1, but a few iterations with larger values of W_1 show that an output voltage of 0.5 V cannot be obtained with realistic values of W_1.

We can examine this situation in more detail by simulating the current in just a single NMOS transistor with the appropriate voltages connected to drain, gate, source and bulk. Fig. 4.13 shows the LTspice schematic for this simulation. In the same way as in fig. 3.21 on page 87, both a Shichman-Hodges transistor (`M1`) and BSIM transistors (`M2` and `M3`) are inserted in order to compare the transistors. For the BSIM transistors, two transistors with different channel length are simulated. The reason for this is that the threshold voltage decreases with decreasing channel length and increasing channel width (Tsividis & McAndrew 2010). Also shown in fig. 4.13 is the result of a '.dc' simulation with the source voltage swept from 0.4 V to 0.6 V in order to analyze the transistor for common drain stage output voltages in this range.

It is obvious from the simulation that the transistors with $L = 1$ µm cannot deliver any output current at an output voltage of 0.5 V, whereas the threshold voltage reduction for a transistor with $L = 0.35$ µm does enable the transistor `M3` to deliver an output current. In order to relax the requirements for output current, the bias current in the source follower should be reduced to the minimum value required to provide the minimum output voltage of −1 V, i.e. 200 µA. With the value of R_B deter-

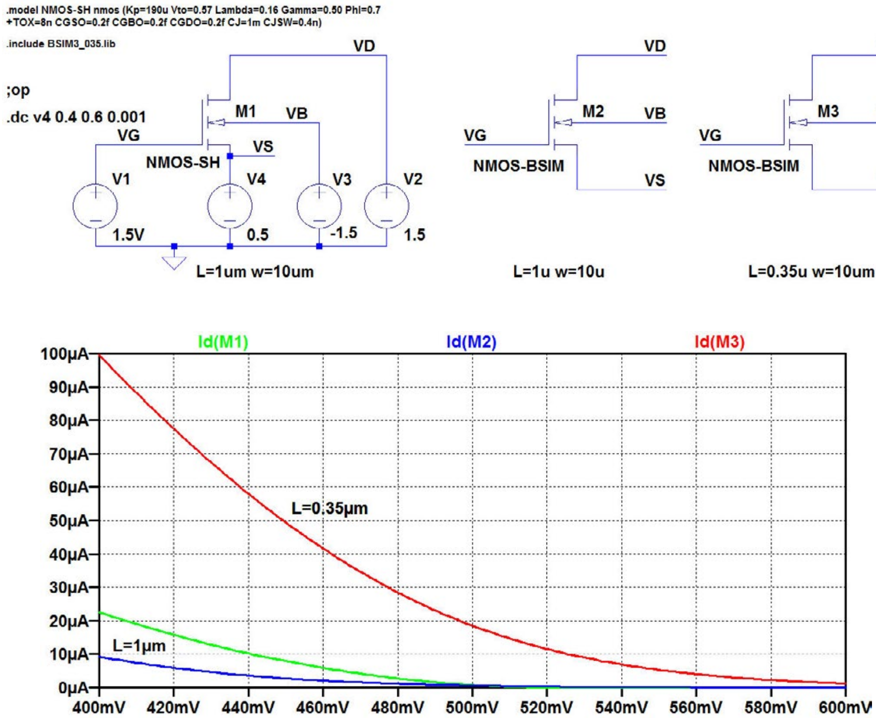


Figure 4.13: LTspice schematic for simulation of the common drain output transistor and simulation plot for output current.

mined from the Shichman-Hodges model (with $\lambda = 0$), the bias current is somewhat higher due to the finite output resistance of the transistors. Thus, to match the design requirements R_B may be increased so that the current in M_2 is reduced to 200 µA.

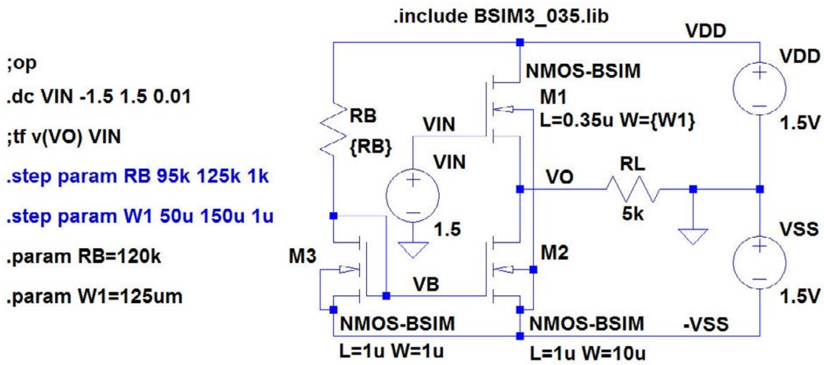
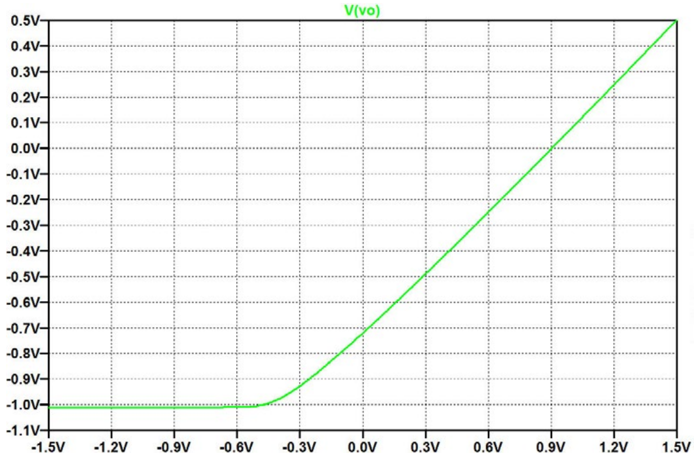


Figure 4.14: LTspice schematic for the common drain stage with bulk effect and with BSIM3 transistor models.

Fig. 4.14 shows the source follower with a minimum channel length transistor for M_1 and with commands for sweeping R_B and W_1. When sweeping R_B, we run a '.op' simulation with a bias value of the input voltage of -1.5 V. This simulation shows that $R_B = 120$ kΩ results in an output voltage



Output from DC Transfer simulation		
- - - Transfer Function - - -		
Transfer_function:	0.862102	transfer
vs#Input_impedance:	1e+020	impedance
output_impedance_at_V(vo):	201.313	impedance

Figure 4.15: Result of a DC sweep with $R_L = 5$ kΩ and from a '.tf' simulation with $R_L = 5$ GΩ of the circuit from fig. 4.14.

of -1.0 V for an input voltage of -1.5 V. Subsequently, W_1 may be swept over a suitable range for a '.op' simulation with an input bias voltage of +1.5 V. This simulation shows that an output voltage of 0.5 V is obtained with $W_1 = 125$ µm. With these component values, a DC sweep shows the required output range, and to get an output bias voltage of 0 V, an input bias of 0.9 V is required. With this input bias, a '.tf' simulation with R_L infinite shows an open circuit voltage gain of $A_{voc} = 0.86$ V/V and an output resistance of $r_o = 201$ Ω. Fig. 4.15 shows the DC sweep with $R_L = 5$ kΩ and the result of the '.tf' simulation with $R_L = 5$ GΩ.

You may notice that one of the reasons why a very wide source follower transistor is needed for the circuit of fig. 4.14 is that it must be able to supply not only current to R_L but also to the biasing transistor M_2. By designing a class AB source follower buffer, the current to a bias transistor can be avoided, see problem 4.3 on page 142. Also the offset between input and output is avoided, and the small signal open circuit voltage gain is 1, even in the presence of the bulk effect.

Example 4.3: The common gate amplifier.

The third basic configuration is the common gate amplifier where the input is applied to the source and the output is taken from the drain while the gate is connected to a fixed bias voltage (small signal ground). In this example, we show an NMOS transistor biased with ideal current sources and with an input signal source with a source resistance R_S and a resistive load R_L at the output, see fig. 4.16. In the figure, source and bulk are connected, implying that there is no bulk effect. The common gain stage may be considered as a current buffer with a current gain of 1, a low input resistance and a high output resistance. One of the tricky properties of the common gate configuration is that it is not a unilateral amplifier, not even at low frequencies. The load resistor affects the input resistance of the stage, and the source resistor affects the output resistance of the stage. From (Carusone, Johns & Martin 2012, p. 126) we have the following expressions for small signal voltage gain, input resistance and output resistance:

$$A_v = \frac{v_o}{v_{in}} = \frac{(g_m + g_{ds})R_L}{1 + R_L g_{ds} + (g_m + g_{ds})R_S} \tag{4.10}$$

$$r_{in} = \frac{1 + R_L g_{ds}}{g_m + g_{ds}} \tag{4.11}$$

$$r_o = 1/g_{ds} + R_S(1 + g_m/g_{ds}) \tag{4.12}$$

where g_m is the transconductance and g_{ds} is the output conductance of the common gate transistor. If the common gate transistor has its bulk contact connected to a fixed voltage (e.g $-V_{SS}$), g_m should be replaced by $g_m + g_{mb}$ in these equations.

With $R_L \to \infty$, the input resistance approaches infinity, and with $R_S \to \infty$, the output resistance approaches infinity. The voltage gain approaches a maximum of $1 + g_m/g_{ds} \simeq g_m/g_{ds}$ for $R_S = 0$ and $R_L \to \infty$.

These relations may be illustrated from a simulation of the circuit which is shown as an LTspice schematic in fig. 4.17. For the simulations, we are assuming a Shichman-Hodges transistor model and transistor dimensions and bias currents as shown in the figure. In the schematic, both R_L and R_S are defined as parameters which can be swept in order to find the input resistance and the output

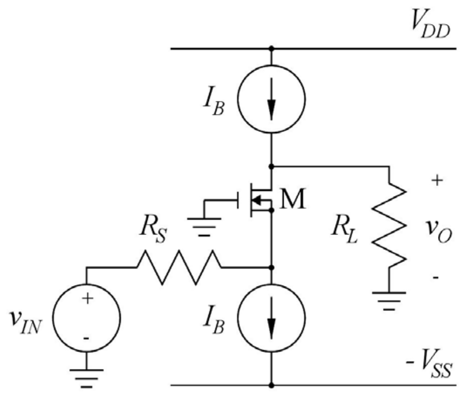


Figure 4.16: Common gate stage biased from ideal DC current sources.

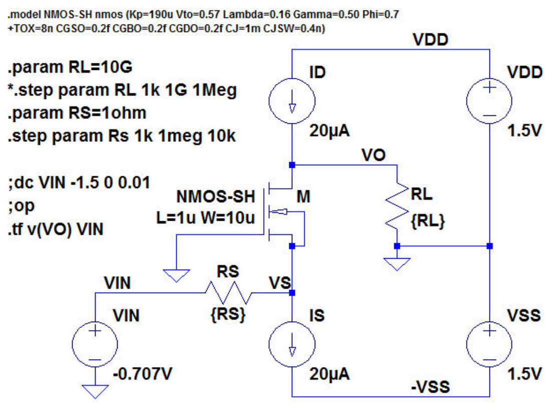


Figure 4.17: LTspice schematic for the common gate stage from fig. 4.16.

resistance as functions of R_L and R_S. Default values for R_L and R_S have been defined. First, a DC sweep of `VIN` is required to find the DC bias value for V_{IN} which gives an output voltage of 0 V. The result is the DC value of `VIN` shown in fig. 4.17. Next, a '.op' simulation is run to verify that input and output voltages are as expected and to find the transistor small signal parameters from the error log file ('Ctrl-L'). This results in $g_m = 290$ µA/V and $g_{ds} = 2.85$ µA/V. Then, the input and output resistance can be simulated using a DC transfer ('.tf') simulation with `v(VO)` as the output and `VIN` as the input. In order to find the input resistance, R_S is specified to a small value, e.g. 1 Ω, and R_L is stepped over a suitable range, e.g. 1 kΩ to 1000 MΩ as shown in fig. 4.17. In order to find the output resistance, R_L is specified to a very large value, e.g. 10 GΩ, and R_S is stepped over a suitable range, e.g. 1 kΩ to 1 MΩ as shown in fig. 4.17. The results of these simulations are shown in fig. 4.18. The traces to be plotted are selected using the command 'Plot Settings $\rightarrow$ Visible Traces' in the plot window. From this figure, you find that the output resistance is about $100 \times R_S$ and the input resistance is about $R_L/100$. This is as expected from (4.11) and (4.12)

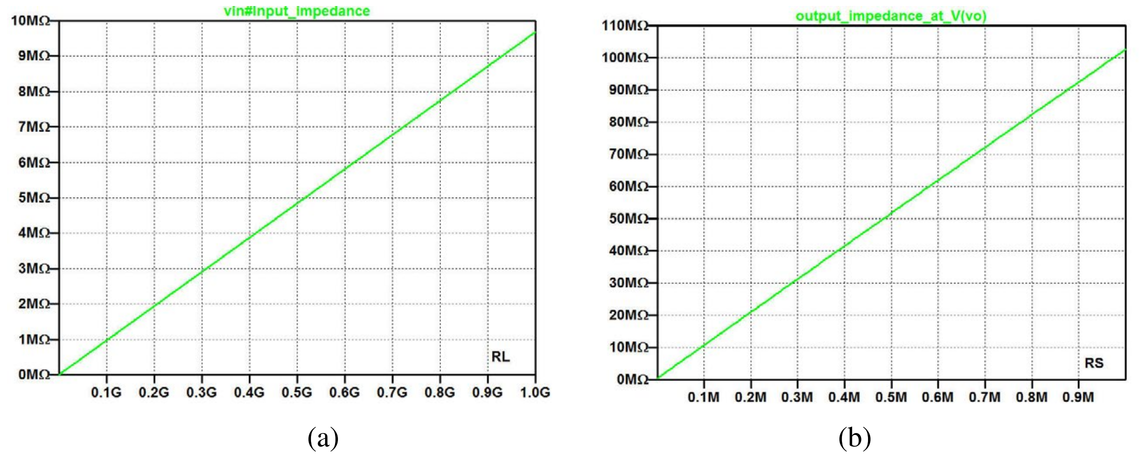


Figure 4.18: Simulations of input and output resistance of common gate stage. (a) Input resistance versus load resistance. (b) Output resistance versus source resistance.

and $g_m/g_{ds} = 290/2.85 = 101.7$. Also, the transfer function may be plotted from the simulation, showing a value of 103 V/V for R_S small (1 kΩ) and R_L large (10 GΩ) as expected from (4.10).

The cascode stage: The fact that the common gate stage transforms the resistance level from a comparatively low level at the input to a high level at the output is used in the frequently encountered cascode stage, combining a common source stage and a common drain stage as shown in fig. 4.19. Two versions of the cascode are shown: the telescopic cascode using two transistors of the same type (fig. 4.19(a)), and the folded cascode using a combination of an NMOS transistor and a PMOS transistor (fig. 4.19(b)). Provided that the bias current source I_B (I_{BN} for the folded cascode) is an ideal current source (or a current source with a very high output resistance), the resistance level at the output of the cascode is very high, resulting in a very high small signal voltage gain from input to output of the cascode. From (4.12), the output resistance with an ideal bias current source I_B is found as

$$r_o = r_{ds2} + r_{ds1}(1 + (g_{m2} + g_{mb2})/g_{ds2}) \simeq r_{ds1}(g_{m2} + g_{mb2})r_{ds2} \tag{4.13}$$

Assuming for instance the transistor parameters corresponding to the simple Shichman-Hodges model from fig. 3.3 on page 71 with $L_1 = L_2 = 1$ µm, $W_1 = W_2 = 10$ µm and $I_B = 20$ µA, we find for the telescopic cascode in fig. 4.19(a) r_o on the order of 40 MΩ, giving a small signal gain $v_0/v_{in} = -g_{m1} \times r_o$ on the order of 80 dB. This means that an input signal swing on the order of 100 µV results in an output voltage swing on the order of volts, so a DC sweep simulation over a small range of input voltage and with an increment of e.g. 10 µV is necessary to find a proper value for the DC bias value of V_{IN} to ensure an output bias voltage which is a reasonable fraction of V_{DD}. Once a DC bias value for V_{IN} is found, the operating point and the small signal transistor parameters in the

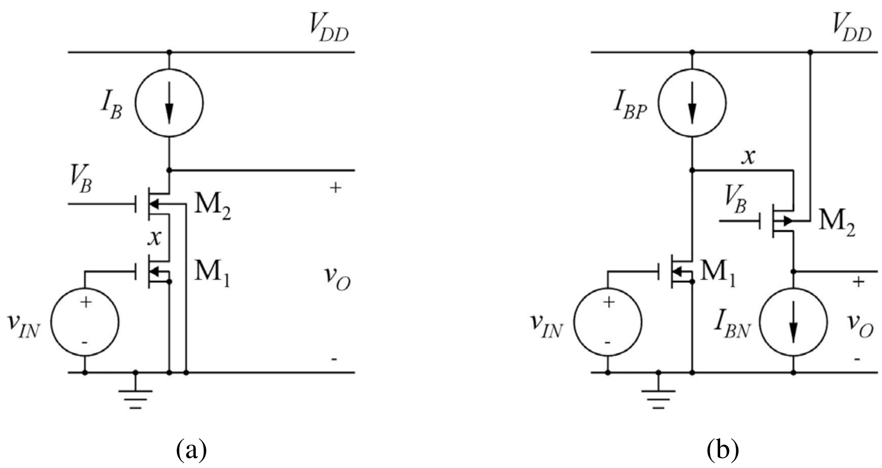


Figure 4.19: Cascode stage. (a) Telescopic cascode. (b) Folded cascode.

operating point can be found from a '.op' simulation, and gain and output resistance can be found from a '.tf' simulation. Also the resistance level in the intermediate node x can be found from a '.tf' simulation with `V(vx)` defined as the output. The detailed simulations for the circuits from fig. 4.19 are left for the reader, see problems 4.4 and 4.5 on page 143 and 144.

Example 4.4: The differential pair.

The last configuration to be examined in this tutorial is the differential pair. Fig. 4.20 shows a differential pair with PMOS input transistors and an active load with NMOS transistors. Also shown is a simple bias circuit consisting of R_B, M_5 and M_6 which generates a constant bias current, I_{D5}, for the differential pair. We will use this circuit for showing how to configure different simulations and will not go into details concerning the design of a differential stage. For simplicity, the stage is simulated with all transistors having the same size: $L = 1$ μm, $W = 10$ μm, $AD = AS = 10$ (μm)^2, $PD = PS = 12$ μm. The transistor models used for the simulations are the BSIM3 models adapted from (Carusone, Johns & Martin 2014), see fig. 3.10 on page 77. The bias current is set by the resistor R_B to a level of about 20 μA for the current mirror M_5 and M_6 and 10 μA for the other transistors.

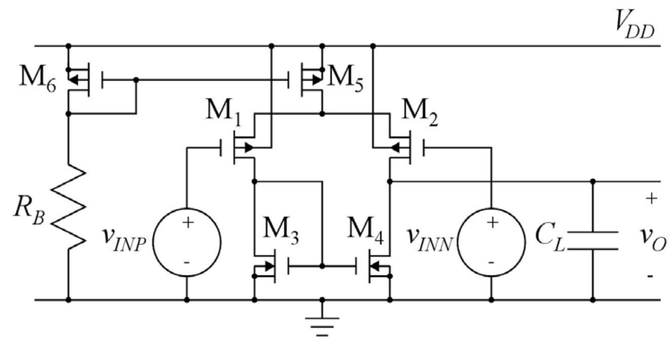


Figure 4.20: PMOS differential pair with NMOS active load.

For a differential stage, the input voltages are normally split into a differential input voltage and a common mode input voltage with $v_{IN+} = V_{CM} + v_{in}/2$ and $v_{IN-} = V_{CM} - v_{in}/2$ where V_{CM} is the common mode input voltage and v_{in} is the differential input voltage. In LTspice, this may be achieved by connecting a voltage controlled voltage source to each of the two inputs as shown in fig. 4.21. Alternatively, the connection shown in problem P4.6 on page 144 can be used. Fig. 4.21 includes most of the simulation commands and SPICE Directives described in the following.

Finding the correct bias point: As a starting point for all the small signal simulations of the circuit, a suitable bias point must be established. The circuit shown in fig. 4.21 is designed to operate with input gate voltages for M_1 and M_2 in a range extending from slightly below ground to an upper limit determined by the supply voltage, the drain-source saturation voltage of M_5 and the gate-source voltage of M_1 and M_2, i.e. around 2 V. Thus, a reasonable common mode input bias voltage would be 1 V. For the differential input voltage, a reasonable bias value would be 0 V, and the offset voltage V_{off} is also 0 V since the circuit has been designed to be fully symmetrical and the

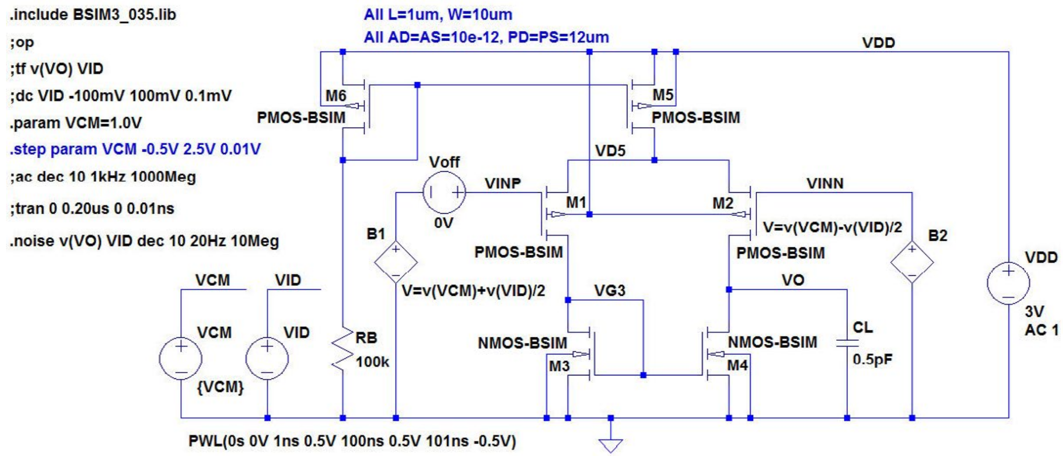


Figure 4.21: LTspice schematic for simulations of the circuit from fig. 4.20.

transistors are assumed to match perfectly. The expected output voltage for this input bias would be the same as the gate-source voltage of M_3 and M_4. Running a '.op' simulation actually confirms that these input bias conditions are reasonable and provide an output voltage of 0.69 V. Moreover, the error log file shows that the current levels are as expected, that all transistors are in saturation and that g_m of the input transistors is slightly smaller than 0.1 mA/V.

Notice that in fig. 4.21, the input common mode voltage has been defined as a parameter `VCM` which can be stepped in combination with other analyses, e.g. a '.tf' simulation. The default value for `VCM` has been set to 1.0 V. The DC values for the differential input voltage and the offset voltage have both been set to 0 V because the design is perfectly symmetrical, so there is no systematic offset. Had there been a mismatch, e.g. between M_1 and M_2, a DC sweep of `Voff` can be used to find the offset voltage which would be needed to give the correct output voltage, see problem 4.7 on page 145. Alternatively, the offset voltage can be found from a '.op' simulation with the output connected back to the inverting input and the common mode voltage specified to the desired output voltage. With `Voff` and `VID` set to 0, the feedback ensures that the voltage difference between the non-inverting input and the inverting input is equal to the offset voltage of the differential stage.
Fig. 4.22 shows a DC sweep of the differential input voltage from -100 mV to $+100$ mV. Obviously, the output voltage range extends from 0.2 V to 1.8 V. The output voltage for a differential input voltage of 0 V is equal to the DC bias voltage `VG3`.

Differential low frequency gain: Once the bias point has been verified, a quick simulation of the low frequency gain can be obtained by a '.tf' simulation with `v(VO)` as the output and `VID` as the input. This will show a gain of $A_{voc} = 52$ V/V or 34 dB. The '.tf' simulation also shows an output resistance of $r_o = 535$ kΩ as expected from g_{ds} for M_2 and M_4 found from the '.op' simulation.

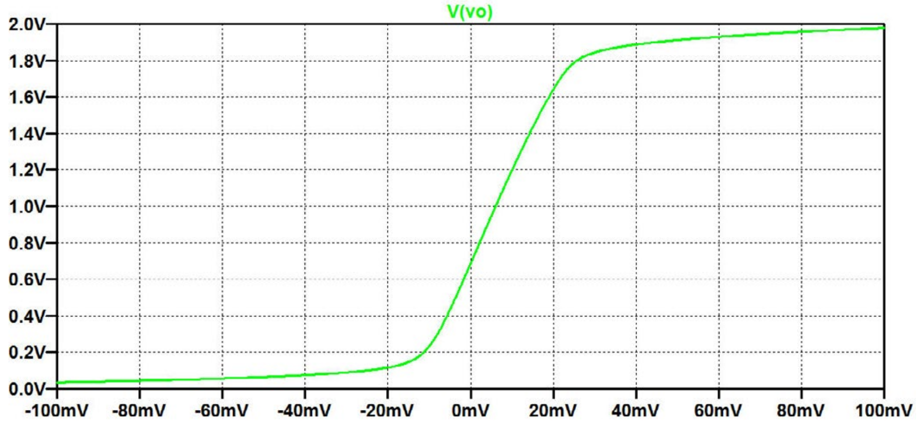


Figure 4.22: Plot of output voltage versus differential input voltage for a common mode input voltage of 1 V for the circuit of fig. 4.21.

Input common mode voltage range: The input common mode range is the voltage range for V_{CM} where the differential stage has all transistors operating in the active region so that the small signal gain is almost constant over this range of V_{CM}. For the circuit of fig. 4.20, the upper limit is defined by M_5 which enters the triode region when V_{CM} increases, implying that M_5 can no longer supply the required bias current to M_1 and M_2. The lower limit is defined by M_1 which will enter the triode region when V_{CM} is so low that there is no longer headroom for the gate-source voltage of M_3. From the error log file from the '.op' simulation, you may find $|V_{GS}| - |V_t| = |V_{dsat}|$ to be 0.25 V for M_5 and 0.19 V for M_1. In order to check the voltage levels for the drain-source voltages of M_5 and M_1 we run a DC sweep from -1 V to $+3$ V with V_{CM} as the source.

Fig. 4.23 shows the plot of $|V_{DS5}|$ (green trace) and $|V_{DS1}|$ (blue trace), respectively. From this, we find a common mode input range from about -0.2 V to about $+1.8$ V. In this voltage range, the differential gain is expected to remain around 34 dB as found from the '.tf' simulation in the bias point.

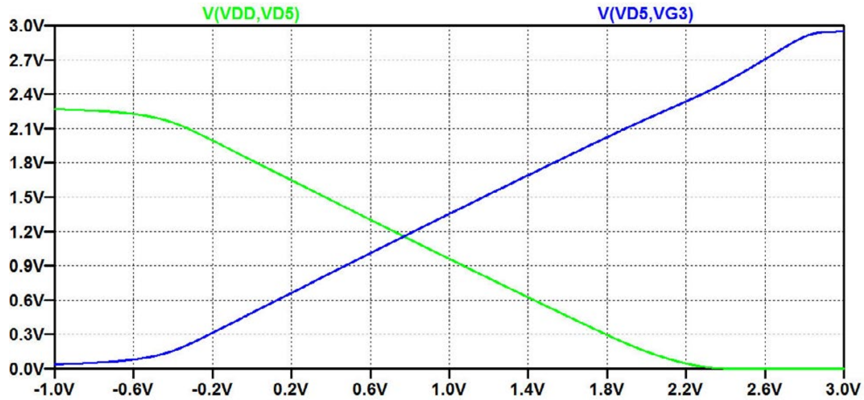


Figure 4.23: Plot of drain-source voltage for M_1 (blue trace) and M_5 (green trace) versus the common mode input voltage for the circuit of fig. 4.21.

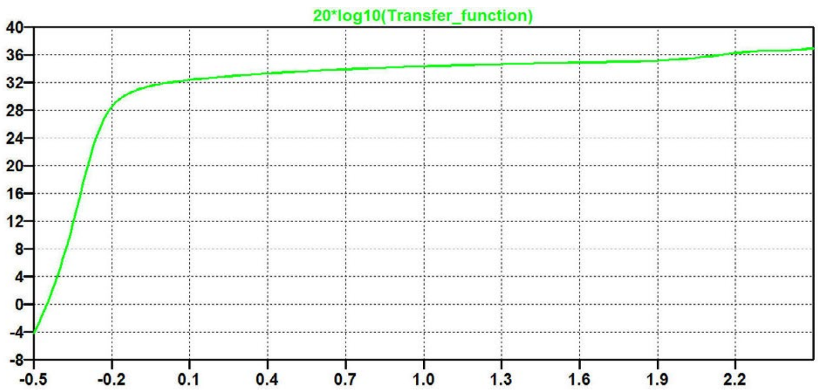


Figure 4.24: Plot of low frequency small signal differential gain versus the common mode input voltage for the circuit of fig. 4.21.

This can be verified by a '.tf' simulation where `VCM` is stepped from -0.5 V to $+2.5$ V. Fig. 4.24 shows a plot of the transfer function. Obviously, the gain falls off for smaller values of V_{CM}. The reason for this is that g_{ds2} increases as M_2 approaches the triode region, but still, the useful range extends down to the negative rail with a drop in gain of less than 3 dB. You may introduce a small negative offset voltage (e.g. −5 mV) to reduce the bias value of the output voltage and thus bring M_2 deeper into the active region. With $V_{\text{off}} = -5$ mV you will find a gain of more than 31 dB for a common mode input voltage down to −0.22 V.

For V_{CM} in the range from 1.8 V to 2.5 V, the gain increases somewhat. The reason for this is that M_5 enters the triode region, so the bias current for the differential pair is reduced, leading to smaller values of g_m and g_{ds} with g_{ds} being reduced more than g_m. This range of operation will normally not be considered as useful because the smaller bias current also means smaller slew rate and smaller unity gain frequency.

Differential frequency response: When the differential stage is loaded by a capacitance C_L which is much larger than the internal transistor capacitances, the frequency response is expected to have a dominant pole at $\omega_p = (r_o C_L)^{-1}$ and a gain bandwidth product of $2\pi \times GBW = A_{\text{voc}}\omega_p = g_{m1}/C_L$. From the error log file from a '.op' simulation, we see that the internal transistor capacitances are on the order of 10^{-14} F, i.e. more than one order of magnitude smaller than C_L, so the condition for a dominant pole is fulfilled, but non-dominant poles can be expected at frequencies comparable to the gain bandwidth product. An AC simulation with V_{id} as the input signal results in a Bode plot for the output voltage as shown in fig. 4.25. V_{id} is defined as the input signal by specifying the AC

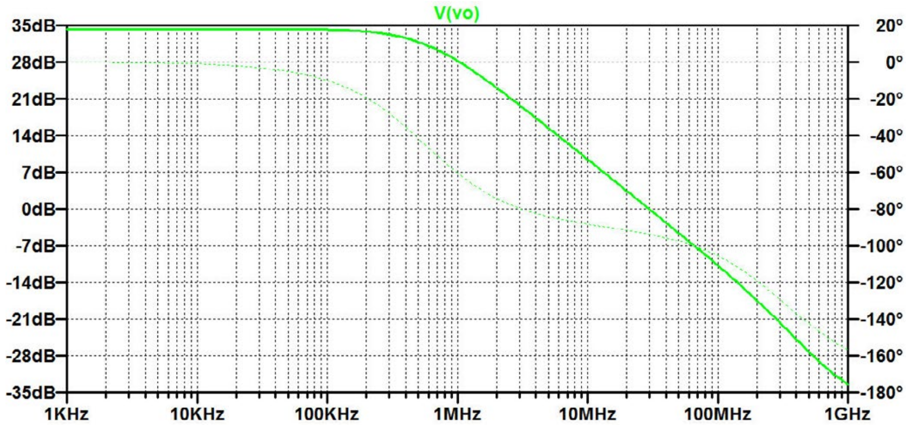


Figure 4.25: Small signal frequency response for the differential gain for the circuit of fig. 4.21.

amplitude of V_{id} to be 1 V, see page 49. From the magnitude plot, we find a unity gain frequency of 30 MHz, i.e. close to the value expected from the g_m values found from the '.op' simulation. Also, the Bode plot shows that at frequencies above 10 MHz, the phase response indicates the presence of higher order poles and zeros.

Common mode gain: With a perfectly matched differential pair, the common mode gain is expected to be very small, ideally 0 at low frequencies, but at high frequencies the common mode rejection is smaller due to differences in the capacitive loading in the two sides of the differential pair. The common mode gain can be simulated with V_{CM} as the input signal for an AC simulation. Fig. 4.26 shows the simulation plot. With a mismatch in the differential pair, a significantly higher common mode gain can be expected, see problem 4.7 on page 145.

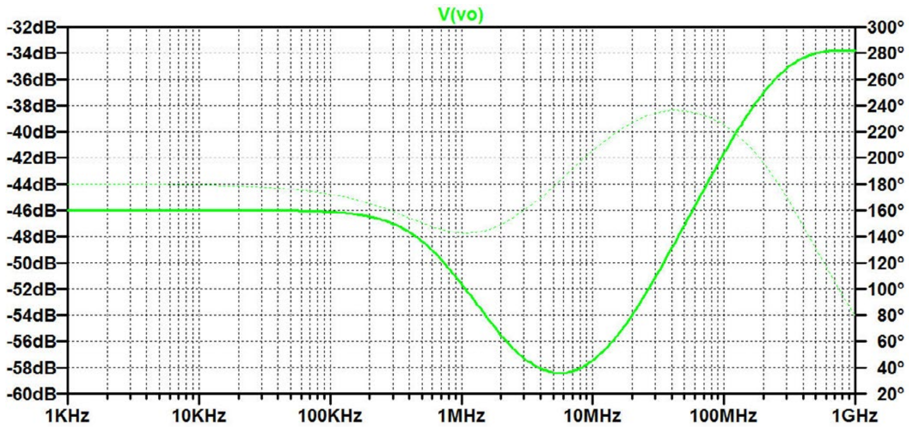


Figure 4.26: Small signal frequency response for the common mode gain for the circuit of fig. 4.21.

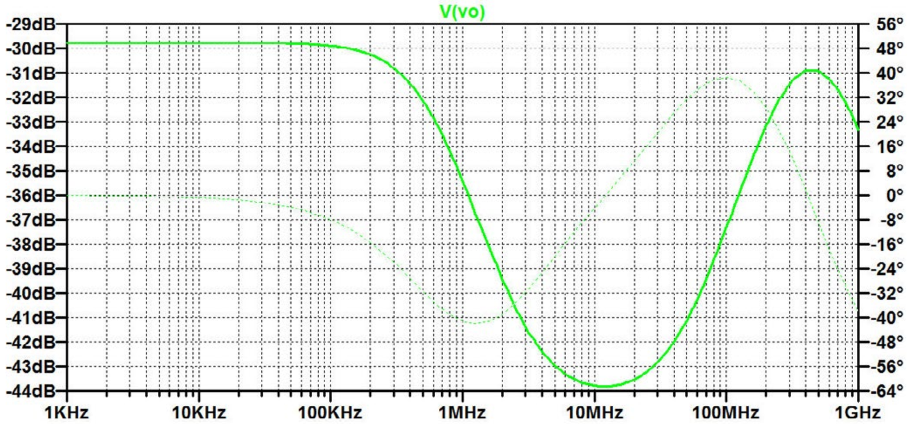


Figure 4.27: Power supply rejection for the circuit of fig. 4.21.

Power supply rejection: Also the power supply rejection can be simulated with an AC simulation. For this, the supply voltage V_{DD} must be specified with an AC amplitude of 1 V (as shown in fig. 4.21) while the other voltage sources should have AC amplitudes of 0. Fig. 4.27 shows the simulation plot of the power supply rejection. The power supply rejection ratio is found by dividing the differential small signal gain by the small signal gain from V_{DD} to v_O (Sedra & Smith 2011).

Common mode rejection ratio (CMRR) and power supply rejection ratio (PSRR): Common mode gain and power supply rejection are often characterized by the parameters CMRR and PSRR which are defined as the ratio between the differential gain and the common mode gain or power supply rejection, respectively (Sedra & Smith 2011). For simulating CMRR and PSRR, we need both the differential gain, the common mode gain and the power supply rejection in one simulation. We can achieve this by defining a parameter M with a value of -1 for common mode gain, 0 for differential gain and $+1$ for power supply rejection.

When simulating the common mode gain, the AC amplitude of V_{CM} should be 1 while the AC amplitude of V_{id} and V_{DD} should be 0. Likewise, for simulating the differential gain, the AC amplitude of V_{id} should be 1 while the AC amplitude of V_{CM} and V_{DD} should be 0, and for simulating the power supply rejection, the AC amplitude of V_{DD} should be 1 while the AC amplitude of V_{id} and V_{CM} should be 0. This can be achieved by defining the amplitudes as parameters ‘`VCM_AC`’, ‘`VID_AC`’ and ‘`VDD_AC`’, respectively, and stepping through the three needed combinations using a ‘step’ parameter ‘`M`’. The parameters ‘`VCM_AC`’, ‘`VID_AC`’ and ‘`VDD_AC`’ are defined by table specifications as

$$.\mathrm{param}\ \mathtt{VCM\_AC} = \mathtt{table}(\mathtt{M},1,1,2,0,3,0) \tag{4.14}$$

$$.\mathrm{param}\ \mathtt{VID\_AC} = \mathtt{table}(\mathtt{M},1,0,2,1,3,0) \tag{4.15}$$

$$.\mathrm{param}\ \mathtt{VDD\_AC} = \mathtt{table}(\mathtt{M},1,0,2,0,3,1) \tag{4.16}$$

In (4.14) - (4.16), the first table entry is the step number M, and this is followed by pairs of M-values (counting from 1 to 3) and amplitude values.

Fig. 4.28 shows the LTspice schematic with these definitions of the AC amplitudes and the '.ac' simulation command. Also included is the '.step' command, stepping M through the values 1, 2 and 3 in order to achieve the common mode gain, the differential gain and the power supply rejection in one simulation.

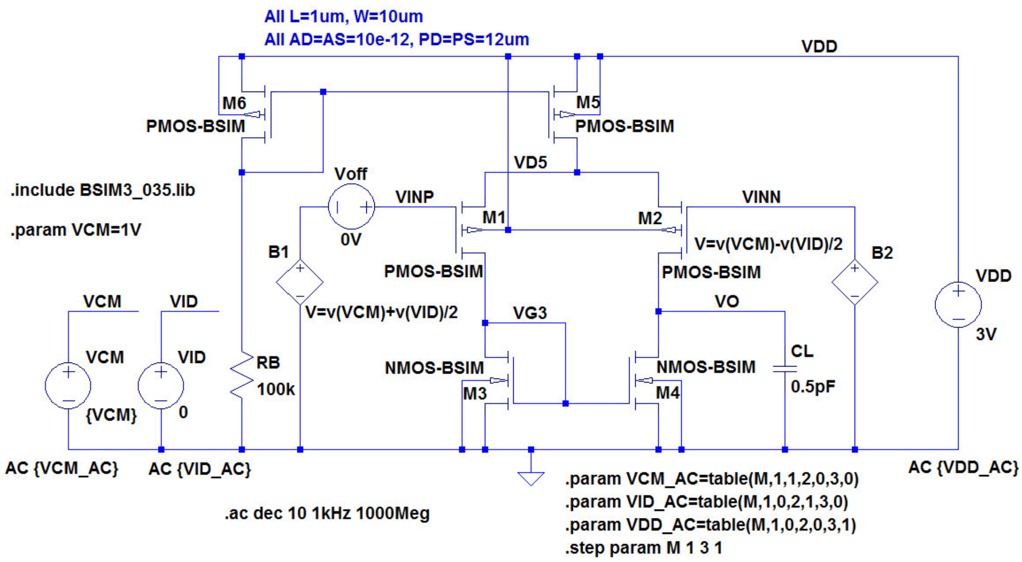


Figure 4.28: LTspice schematic for simulating both common mode gain, differential gain and power supply rejection.

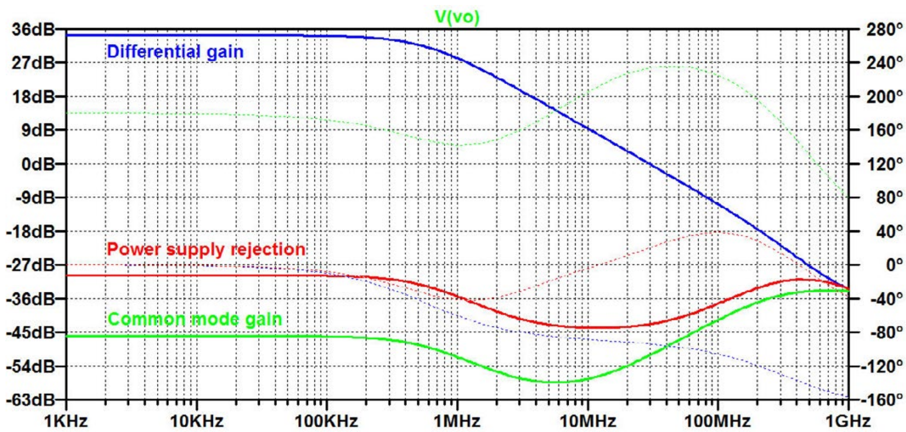


Figure 4.29: Common mode gain (green trace), differential gain (blue trace) and power supply rejection (red trace) for the circuit of fig. 4.21.

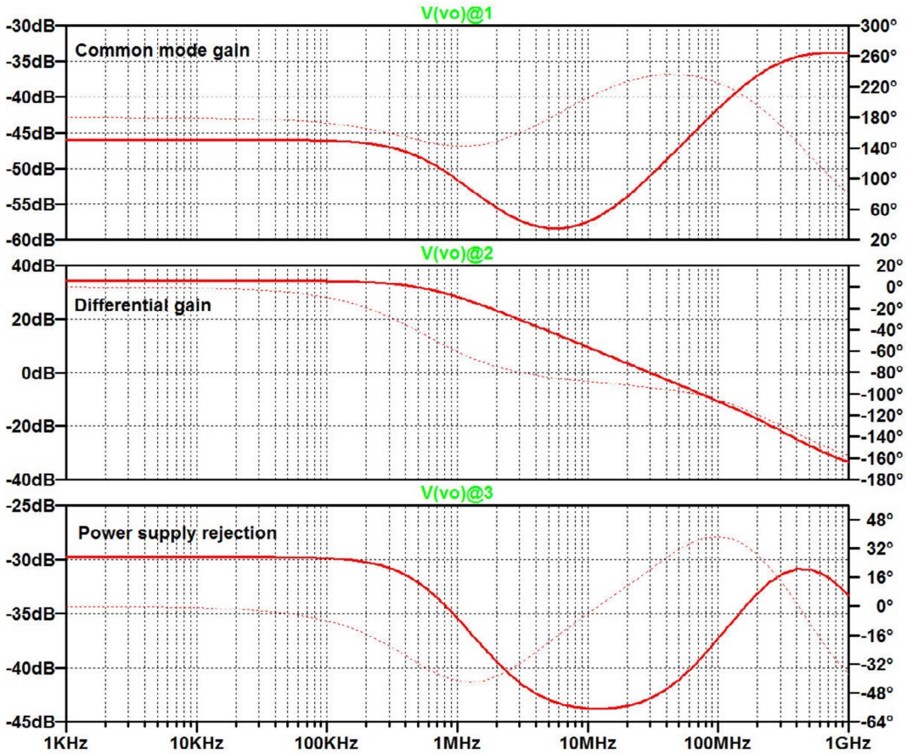


Figure 4.30: Simulation traces from fig. 4.30 separated into different variables.

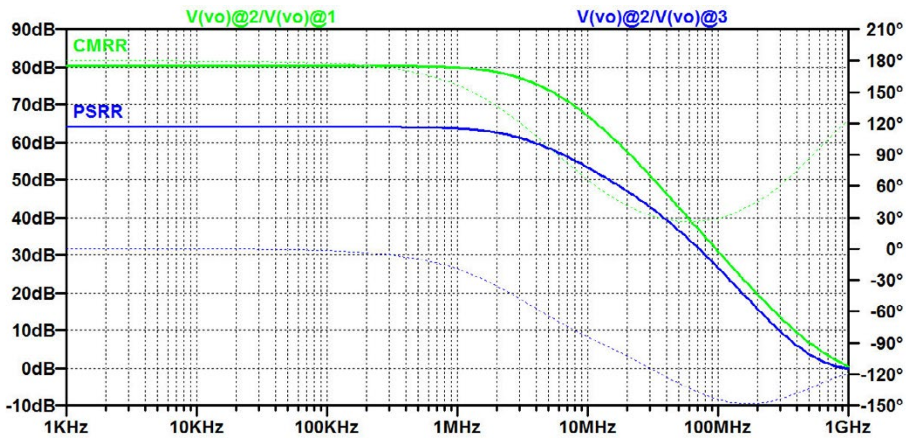


Figure 4.31: Common mode rejection ratio (CMRR) (green trace) and power supply rejection ratio PSRR (blue trace) for the circuit of fig. 4.21.

Fig. 4.29 shows the resulting simulation plot. The green trace (trace 1) shows the common mode gain, the blue trace (trace 2) is the differential gain, and the red trace (trace 3) is the power supply rejection. You may compare the traces to figs. 4.25, 4.26 and 4.27. Notice that the three traces are plotted as the result of showing just one variable, ‘`V(vo)`’. For finding CMRR and PSRR, we need to separate the three traces into different variables. This is achieved by adding the operand @ to the variable. Thus, ‘`V(vo)@1`’ shows only the common mode gain (trace 1), ‘`V(vo)@2`’ shows only the differential gain (trace 2), and ‘`V(vo)@3`’ shows only the power supply rejection (trace 3). Fig. 4.30 shows the three different traces in separate plot panes (achieved by the command ‘Plot Settings $\rightarrow$ Add Plot Pane’) so that different scaling of the Y-axes can be used.

Finally, fig. 4.31 shows CMRR and PSRR calculated as ‘`V(vo)@2/V(vo)@1`’ and ‘`V(vo)@2/V(vo)@3`’, respectively.

Slew rate: The slew rate SR of the output voltage is limited by the current I available to charge and discharge the load capacitor C_L. From $SR = I/C_L$ and with a maximum current I limited by M_5, a slew rate of about 40 V/µs can be expected. The slew rate can be simulated by a transient simulation with square wave pulses applied to the input, so that the output switches between the minimum and maximum available output. Fig. 4.32 shows the simulation plot from a transient simulation. The plot shows the differential input voltage and the output voltage. From the slope of the output voltage, a slew rate of about 40 V/µs is found as expected with $I = 20$ µA and $C_L = 0.5$ pF.

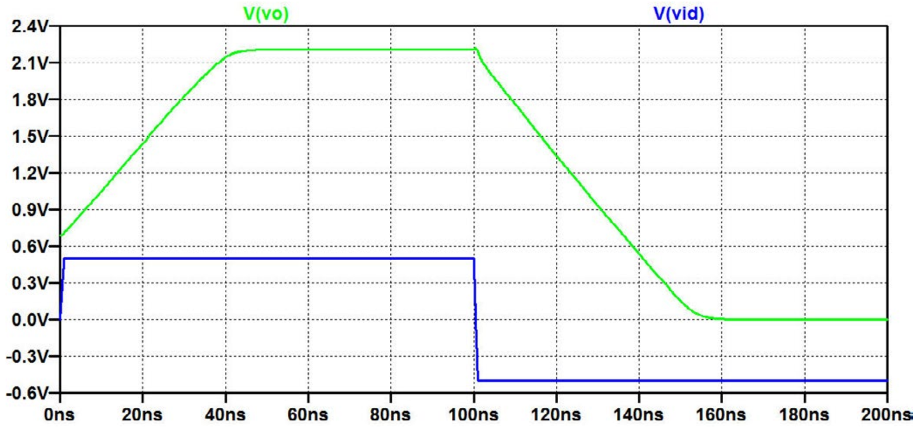


Figure 4.32: Transient simulation of output voltage for finding the slew rate of the circuit of fig. 4.21.

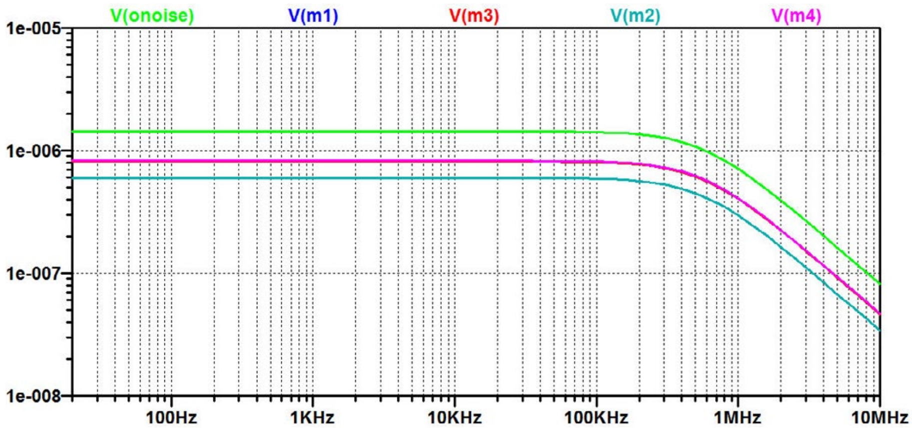


Figure 4.33: Noise simulation of the circuit from fig. 4.21.

Noise: The final simulation considered here is a noise simulation. The specification of a noise simulation is similar to the specification of an AC simulation. The noise simulation is selected by the command 'Simulate → Edit Simulation Command' which opens the window with tabs for the different simulations (fig. 1.5 on page 18). When selecting 'Noise', a tab opens for specifying the output voltage, input voltage, frequency range and sweep characteristics. As an example, a noise simulation of the circuit from fig. 4.21 on page 127 is performed over a frequency range from 20 Hz to 10 MHz. After running the simulation, the noise spectral density of the output voltage can be plotted by pointing to the output node in the schematic. Also, the noise contribution from each component can be plotted by pointing to the component and left clicking. The cursor turns into a red pointer ( ) when moved to a component. Fig. 4.33 shows a plot from the noise simulation with traces for the output noise spectral density and for the noise contributions from transistors M_1 to M_4. Observe that a logarithmic scale is applied to both axes. The traces for `V(m1)` and `V(m3)` (noise from M_1 and M_3) cannot be seen as they are hidden beneath the identical noise contributions from M_2 and M_4. All the noise contributions are noise at the specified output terminal. Obviously, for the circuit of fig. 4.21, the active load (M_3 and M_4) is the main noise contributor. It should be noted that the circuit of fig. 4.21 is in no way optimized for noise performance. For details on how to optimize this, the analysis from (Carusone, Johns & Martin 2012, pp. 392-394) may be applied.

A useful feature is that by pointing to a trace label and using 'Ctrl-left click', the total noise voltage integrated over the specified frequency range is calculated and displayed in a separate window. Thus, for the circuit of fig. 4.21, a 'Ctrl-left click' on the trace label shows a total output noise RMS voltage of 1.33 mV.

Also the input referred spectral noise density may be plotted. It is selected from the schematic using the command 'View → Visible Traces' and selecting '`V(inoise)`'.

Note that the transistor models used for this simulation do not include flicker noise. This is apparent since the noise spectral density is flat all the way down to very low frequencies. The noise parameters for a transistor may be included in the transistor model, but for the model taken from (Carusone, Johns & Martin 2014), no flicker noise parameters are specified. The most important parameter describing the flicker noise is the flicker noise coefficient KF (Baker 2010; Allen 2012). The modeling of flicker noise is beyond the scope of this tutorial. Problem 4.8 on page 145 illustrates the simulation of flicker noise with noise parameters for a 0.35 µm CMOS process.

Hints and pitfalls

- Show relevant transistor parameters in the schematic by using the ‘Component Attribute Editor’, see page 106.
- Use the ‘.op’ (DC op pnt) analysis for checking voltages and currents in the bias point of your circuit.
- Check small signal parameters and operating region of transistors using the error log file (‘Ctrl-L’) from the ‘.op’ simulation.
- Use the ‘.op’ analysis and the ‘.tf’ (DC Transfer) analysis for checking basic properties at low frequencies of your circuit.
- Perform DC sweeps to find proper bias values for input signal sources.
- Define component values as variable parameters and run ‘.op’ and/or ‘.tf’ simulations with parameter sweeps (‘.step parameter ...’) for the design of component values.
- Set parameters to proper default values and remove ‘.step param’ commands before running AC Analysis, DC sweep, Transient or Noise simulations (unless you really want simulation results for several values of the parameters).
- When using a ‘.step’ command in combination with an AC Analysis, DC sweep, Transient or Noise simulation, the plot of a variable results in several traces, one for each step. You can extract the trace for a specific step by the operand @<step number> after the name of the variable when displaying plot traces, see example on page 134.
- Assure correct bias voltages and currents before running DC Transfer, AC Analysis or Noise simulations.
- If the results of a DC Transfer simulation, AC Analysis or Noise simulation are very different from what you expected, check that your bias point is what you expected (using the ‘.op’ analysis and checking both the output file and the error log file from the ‘.op’ analysis.)
- If the results of an AC analysis are very different from what you expected, also check that you have only one AC source with an amplitude of 1 specified in your circuit.

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.

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

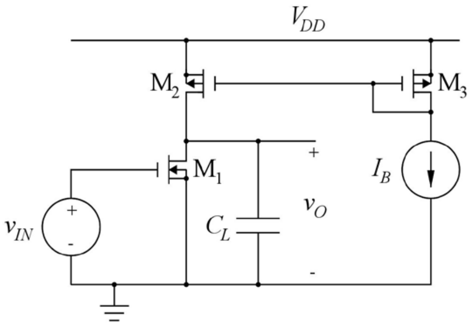
Carusone, TC., Johns, D. & Martin, K. 2014, *Analog Integrated Circuit Design, Netlist and model files*. Retrieved from http://analogicdesign.com/students/netlists-models/

Sedra, AS. & Smith, KC. 2011, *Microelectronic Circuits*, International Sixth Edition, Oxford University Press, New York, USA.

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.

Problems

4.1



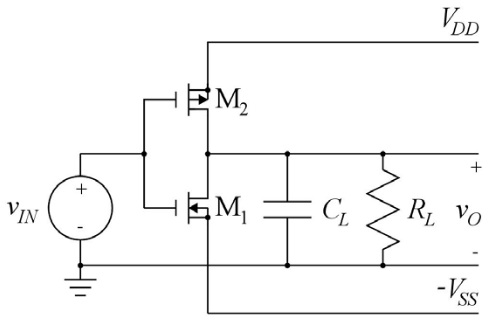
$W_2 = W_3 = 25$ µm, $L_1 = L_2 = L_3 = 1$ µm

$C_L = 3.2$ pF, $V_{DD} = 3$ V, $I_B = 100$ µA.

Figure P4.1

For the common source amplifier shown in fig. P4.1, design M_1 so that the gain bandwidth product of the stage is 50 MHz. Assume a transistor model as specified in fig. P3.2 on page 98 and fig. P3.3 on page 99 and use a channel length of $L_1 =$ 1 µm. Use a channel width for M_1 which is a multiple of 1 µm.
Hint: Design M_1 to have the required g_m for the gain bandwidth product with $I_D = 100$ µA. Find g_m versus I_D using the method shown in example 3.5 on page 90.
Find the DC bias value of the input voltage for which the output voltage is 1.5 V and find the small signal voltage gain A_v at low frequencies.

4.2



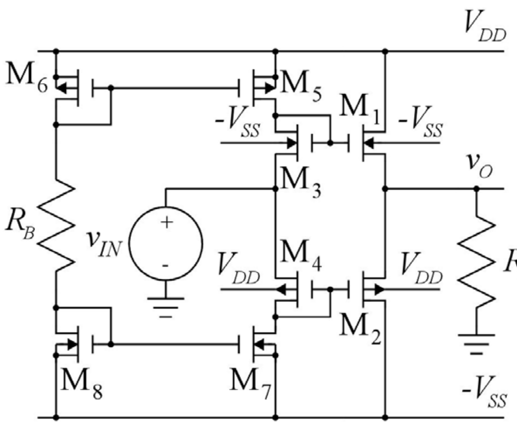
$V_{DD} = V_{SS} = 1.5$ V, $L_1 = L_2 = 1$ μm

$C_L = 0.5$ pF, $R_L = 10$ kΩ

Figure P4.2

For the inverting amplifier shown in fig. P4.2, design M_1 and M_2 so that the DC bias value of the output voltage is within the range ±100 mV with an input DC bias voltage of 0 V and so that the low frequency small signal gain with an input DC bias voltage of 0 V is −10 V/V. Assume transistor models as specified in fig. P3.2 on page 98 and fig. P3.3 on page 99 and use a channel length of $L_1 = L_2 = 1$ μm. Use channel widths for M_1 and M_2 which are multiples of 0.5 μm. What is the low frequency small signal gain if the load resistor R_L is omitted? What is the gain bandwidth product of the amplifier for $R_L = 10$ kΩ and for $R_L = \infty$?

4.3

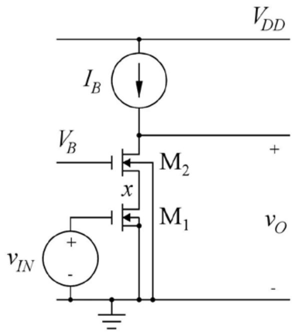


$L_1 = L_2 = L_3 = L_4 = L_5 = L_6 = L_7 = L_8 = 0.35$ μm
$R_L = 10$ kΩ, $V_{DD} = V_{SS} = 1.5$ V.

Figure P4.3

Fig. P4.3 shows a class AB buffer amplifier. Design the output transistors M_1 and M_2 so that the amplifier can deliver an output voltage swing of ± 0.5 V with a load resistor of 10 kΩ. Assume that the gate voltage of M_1 and M_2 can reach the positive and negative supply voltages, respectively. Select values of the channel widths which are multiples of 10 μm. Use transistor models as specified in fig. P3.2 on page 98 and fig. P3.3 on page 99. Design the bias network $M_3 - M_8$ and R_B to provide a bias current of 1 μA for $M_3 - M_8$. $M_5 - M_8$ should be designed to have a saturation voltage $|V_{GS} - V_t|$ of less than 50 mV, and the channel widths should be multiples of 10 μm. M_3 and M_4 should be scaled to channel widths of 0.1 times the channel widths of M_1 and M_2, respectively. Plot the output voltage versus the input voltage for -1.5 V < v_{IN} < 1.5 V. Find the open circuit voltage gain and the output resistance of the buffer for an input bias voltage of 0 V. Find the bias current in M_1 and M_2 for an output bias voltage of $V_O = 0$ V. Why is the current scaling in $M_1 - M_2$ / $M_3 - M_4$ different from the channel width scaling?

4.4



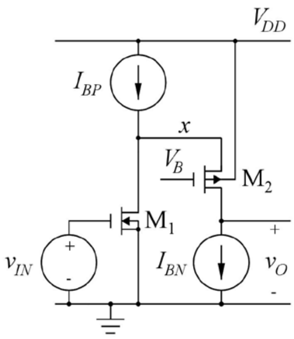
$L_1 = L_2 = 1$ µm, $W_1 = W_2 = 10$ µm
$I_B = 20$ µA, $V_B = 1.5$ V, $V_{DD} = 3$ V.

.MODEL NMOS-SH nmos (Kp=190u Vto=0.57
+Lambda=0.16 Gamma=0.50 Phi=0.7 TOX=8n
+CGSO=0.2f CGBO=0.2f CGDO=0.2f CJ=1m)
+CJSW=0.4n)

Figure P4.4

For the telescopic cascode shown in fig. P4.4, find the bias value of V_{IN} required to give an output voltage of 2 V. Also find the small signal gain A_{voc} and output resistance r_o at low frequencies. Find the small signal resistance r_x to ground from the node x between the source of M_2 and the drain of M_1. Assume a transistor model as shown in fig. P4.4.

4.5



$L_1 = L_2 = 1$ µm, $W_1 = 10$ µm, $W_2 = 30$ µm
$I_{BP} = 40$ µA, $I_{BN} = 20$ µA, $V_B = 1.5$ V, $V_{DD} = 3$ V.

```
.MODEL PMOS-SH pmos (Kp=55u Vto=-0.71
+Lambda=0.16 Gamma=0.75 Phi=0.7 TOX=8n
+CGSO=0.2f CGBO=0.2f CGDO=0.2f CJ=1.5m)
+CJSW=0.4n)
```

Figure P4.5

For the folded cascode shown in fig. P4.5, find the bias value of V_{IN} required to give an output voltage of 1 V. Also find the small signal gain A_{voc} and output resistance r_o at low frequencies. Find the small signal resistance r_x to ground from the node x between the source of M_2 and the drain of M_1. Assume transistor models as shown in fig. P4.4 and P4.5.

4.6

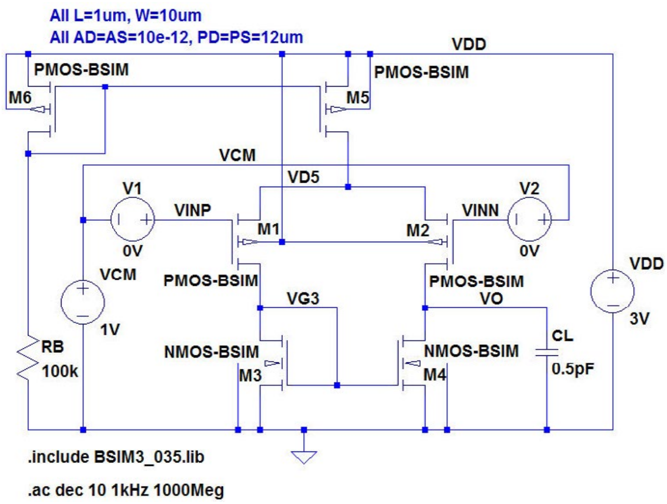
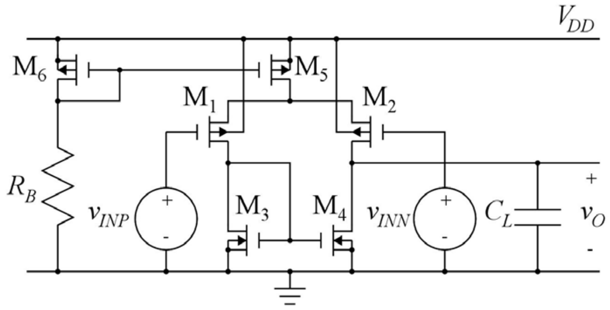


Figure P4.6

Fig. P4.6 shows an alternative version of the LTspice schematic from fig. 4.21 with a different arrangement for the input voltages. Define the AC amplitudes of `VCM`, `V1`, `V2` and `VDD` such that the '.ac' simulation shows the differential gain and compare your simulation to fig. 4.25. Next, define the AC amplitudes of `VCM`, `V1`, `V2` and `VDD` such that the '.ac' simulation shows the common mode gain and compare your simulation to fig. 4.26. Finally, define the AC amplitudes of `VCM`, `V1`, `V2` and `VDD` such that the '.ac' simulation shows the power supply rejection and compare your simulation to fig. 4.27.

4.7



$L_1 = L_2 = L_3 = L_4 = L_5 = 1$ μm

$W_1 = 30$ μm, $W_2 = 33$ μm, $W_3 = W_4 = W_5 = W_6 = 10$ μm

$AD_1 = AS_1 = AD_2 = AS_2 = 30$ (μm)^2

$AD_3 = AS_3 = AD_4 = AS_4 = AD_5 = AS_5 = AD_6 = AS_6 = 10$ (μm)^2

$PD_1 = PS_1 = PD_2 = PS_2 = 32$ μm

$PD_3 = PS_3 = PD_4 = PS_4 = PD_5 = PS_5 = PD_6 = PS_6 = 12$ μm

$R_B = 100$ kΩ, $C_L = 0.5$ pF, $V_{DD} = 3.0$ V, $V_{CM} = 1$ V

Figure P4.7

For the differential pair shown in fig. P4.7, we assume that a layout error has resulted in a mismatch between M_1 and M_2 such that $W_1 = 30$ μm and $W_2 = 33$ μm. Find the input offset voltage caused by this error for a common mode input voltage of $V_{CM} = 1$ V and an output voltage of 0.7 V. Use the Shichman-Hodges transistor model from fig. P4.4 and P4.5. Next, plot the differential gain and the common mode gain versus frequency. Find the gain bandwidth product and calculate the common mode rejection ratio at low frequencies. Also plot the gain from the power supply to the output and calculate the power supply rejection ratio at low frequencies.

4.8

```
.MODEL PMOS-SH pmos (Kp=55u Vto=-0.71
+Lambda=0.16 Gamma=0.75 Phi=0.7 TOX=8n
+CGSO=0.2f CGBO=0.2f CGDO=0.2f CJ=1.5m
+CJSW=0.4n KF=5e-26)
.MODEL NMOS-SH nmos (Kp=190u Vto=0.57
+Lambda=0.16 Gamma=0.50 Phi=0.7 TOX=8n
+CGSO=0.2f CGBO=0.2f CGDO=0.2f CJ=1m
+CJSW=0.4n KF=1e-25)
```

Figure P4.8

For the differential pair shown in fig. P4.7 with $W_1 = W_2 = 30$ μm, assume the transistor models shown in fig. P4.8 which include parameters for the flicker noise modelling. Plot the total noise spectral density of the output voltage and the noise contributions from M_1 and M_3 in the frequency range 20 kHz to 10 MHz, using logarithmic axes. Find the total RMS output noise voltage in this frequency range. Also plot the input referred noise spectral density and find the total RMS input referred noise voltage in this frequency range.

Answers

4.1: $W_1 = 28$ µm; $V_{IN} = 0.7735$ V; $A_v = 38$ dB.

4.2: $W_1 = 6.5$ µm; $W_2 = 21$ µm; $A_{voc} = -43$ V/V; $GBW_{(R_L=10\ \text{k}\Omega)} = GBW_{(R_L=\infty)} = 405$ MHz.

4.3: $W_1 = 30$ µm; $W_2 = 370$ µm; $W_3 = 3$ µm; $W_4 = 37$ µm; $W_5 = W_6 = 30$ µm; $W_7 = W_8 = 10$ µm;
$R_B = 1.84$ MΩ; $A_{voc} = 0.95$ V/V; $r_o = 1019$ Ω; $I_{D1} = I_{D2} = 18.3$ µA;
$|V_{DS}|$ larger for M_1-M_2 than for M_3-M_4 and $|V_{th}|$ smaller for M_1-M_2 than for M_3-M_4.

4.4: $V_{IN} = 708.21$ mV; $A_{voc} = 83$ dB, $r_o = 49$ MΩ, $r_m = 344$ kΩ.

4.5: $V_{IN} = 692.40$ mV; $A_{voc} = 87$ dB, $r_o = 66$ MΩ, $r_m = 439$ kΩ.

4.6: Differential gain: `VCM` = 0, `V1` = 0.5, `V2` = 0.5 and `VDD` = 0;
Common mode gain: `VCM` = 1, `V1` = 0, `V2` = 0 and `VDD` = 0;
Power supply rejection: `VCM` = 0, `V1` = 0, `V2` = 0 and `VDD` = 1.

4.7: $V_{\text{off}} = -4.2$ mV; $GBW = 60$ MHz; CMRR = 66 dB; PSRR = 57 dB.

4.8: $V_{on,\,\text{RMS}} = 1.3$ mV; $V_{in,\,\text{RMS}} = 48$ µV.

Tutorial 5 – Hierarchical design

This tutorial illustrates how complex circuits and systems can be built from basic circuit functions using a hierarchical description. The hierarchical description is useful for creating a better overview of a system design, and it is useful when specific circuit blocks are repeated several times in a system design. After having completed the tutorial, you should be able to

- define a circuit as a subcircuit to be used in a hierarchical circuit description.
- create and edit a symbol for a subcircuit.
- use subcircuits in a higher level schematic.
- apply the hierarchical design to a simple two stage CMOS opamp.
- apply more levels of hierarchy, e.g. in a design using digital gate functions.

Example 5.1: A two stage operational amplifier.

A two stage opamp is built from a differential input stage followed by an inverting gain stage as shown in fig. 5.1. The differential input stage may be a stage similar to the differential pair described in example 4.4 on page 126. The inverting gain stage may be a common source gain stage as described in example 4.1 on page 103. Normally, the opamp will also include some biasing circuit as shown in fig. 4.4 on page 109 and fig. 4.20 on page 126. The bias circuit may be common to the input stage and the inverting gain stage, so in a block diagram, it can be shown as a separate block. Fig. 5.1 shows a block diagram of a two stage operational amplifier. In addition to the blocks described here, the opamp will often include a compensation capacitor C_c to control the frequency response in order to achieve stability in a feedback system using the opamp.

Creating the subcircuits: Combining the schematics from fig. 4.4 and fig. 4.20, we can draw the complete schematic at transistor level as shown in fig. 5.2, and the resulting circuit is so simple that

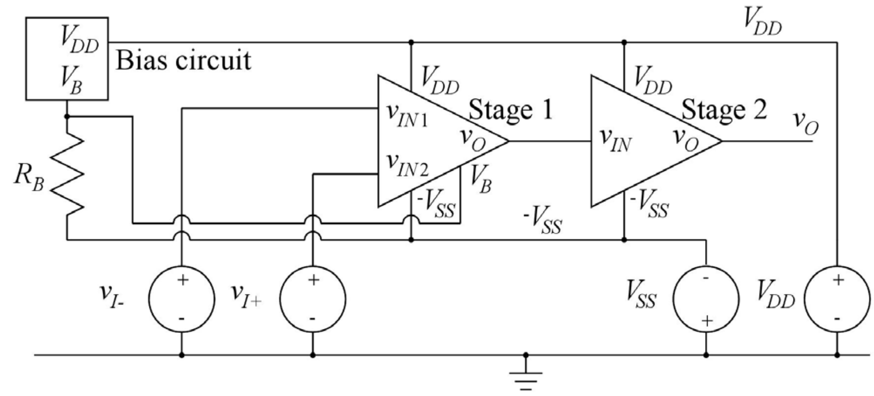


Figure 5.1: Schematic at block level of the two stage operational amplifier.

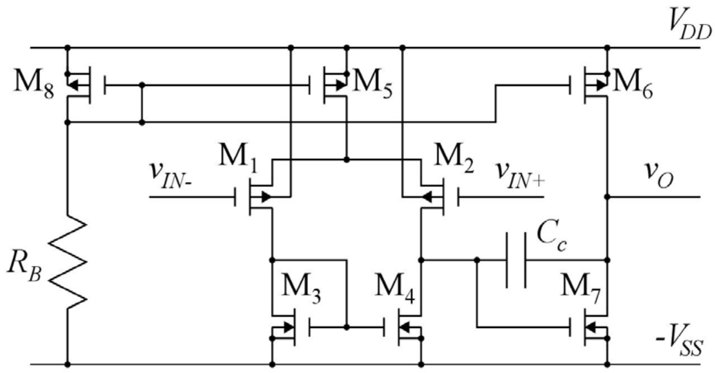


Figure 5.2: Transistor level schematic of two stage operational amplifier.

we could easily perform simulations directly on the entire circuit. However, in this example, we will show how the blocks defined in fig. 5.1 can be defined as subcircuits in LTspice and how the block diagram of fig. 5.1 can be simulated.

We start with the differential pair. In fig. 4.21 on page 127, this is shown as a schematic for LTspice. Now we need only the five transistors M_1 – M_5, but we need to specify which nodes in the circuit are the terminals corresponding to the six terminals shown in fig. 5.1. The six terminals are the two inputs, the output, the two supply voltages and the input for a bias voltage. This is done when labeling the nodes using the command 'Edit → Label Net' (or hotkey 'F4'). In the dialogue box for specifying the Net Name, you select the 'Port Type' to be either 'Input', 'Output' or 'Bi-Direct'. The resulting schematic is shown in fig. 5.3. For the transistors, the model references NMOS-BSIM and PMOS-BSIM to the BSIM3 model from fig. 3.10 on page 77 are used, since these models will be applied for the design of the amplifier. The transistor channel lengths and widths are shown using the approach described on page 106. The transistor drain and source areas and perimeters have been

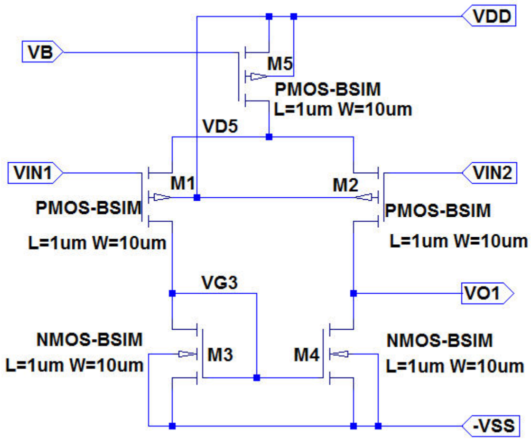


Figure 5.3: LTspice schematic for differential pair subcircuit.

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 on page 75. The dimensions have (arbitrarily) been chosen to the same as in fig. 4.21.

Now the subcircuit schematic is complete, and we need a symbol for it, so that it can be used at a block level as in fig. 5.1. This is achieved by the command 'Hierarchy → Open this Sheet's Symbol'. A new window opens with a message from LTspice: 'Couldn't find this sheet's symbol. Shall I try to

(a) (b)

Figure 5.4: LTspice autogenerated symbol (a) and edited symbol (b) for the differential input pair.

automatically generate one?' Answering 'Yes' results in a new sheet being opened with a symbol as shown in fig. 5.4(a). It is just a box with the output terminal on the right side and the other terminals on the left side. You may notice that the file name is the name of the circuit in fig. 5.3 with a file extension '.asy' to indicate that it is a symbol, and the sheet is open in the symbol editor mode of LTspice. You may wish to edit the symbol to resemble the triangular symbol shown in fig. 5.1. This can be done using the 'Edit' and 'Draw' commands in the symbol editor. Moving the terminals and drawing a triangular shape instead of the rectangular box, the symbol may be modified to look like shown in fig. 5.4(b). When you click 'File → Save', the symbol is saved in the same folder as the subcircuit schematic file.

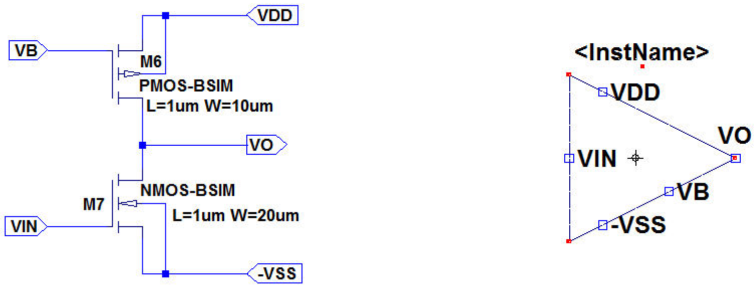
In the same way you can create subcircuit schematics and a symbols for the common source stage and the bias circuit. For the common source stage, the NMOS transistor is chosen to have a channel width of 20 µm (i.e. twice the width of the NMOS transistors in the differential stage) because the current in this transistor is twice the current of the NMOS transistor in the differential stage.

The bias circuit is simply a diode connected PMOS transistor, and the resistor R_B (see fig. 5.2) is considered as a separate component as shown in fig. 5.1.

Fig. 5.5 shows the common source stage subcircuit and the symbol for the common source stage.

Fig. 5.6 shows the bias subcircuit and the symbol for the bias circuit.

Simulating at block level: Using the symbols just created, we can draw the schematic at block level as shown in fig. 5.7. When inserting the subcircuits, remember to select the folder ('Top Directory') with your subcircuits in the component selection window, see fig. 1.3 on page 15. Notice that the model reference ('`.include BSIM3_035.lib`') to the transistor models NMOS-BSIM and PMOS-BSIM is given in the block level schematic, not in the separate subcircuit schematics. This ensures that the same transistor models are used for all subcircuits, so if the model specification is changed, e.g. by using a '.model' specification from a specific vendor rather than the generic BSIM3 model from (Carusone, Johns & Martin 2014), this will automatically affect all the subcircuits. If the model definitions are included as '.model' specifications in the subcircuits, the specifications in the subcircuit schematic will override the specifications at the block level.



(a) (b)

Figure 5.5: LTspice schematic (a) and symbol (b) for the common source stage.

(a) (b)

Figure 5.6: LTspice schematic (a) and symbol (b) for the bias circuit.

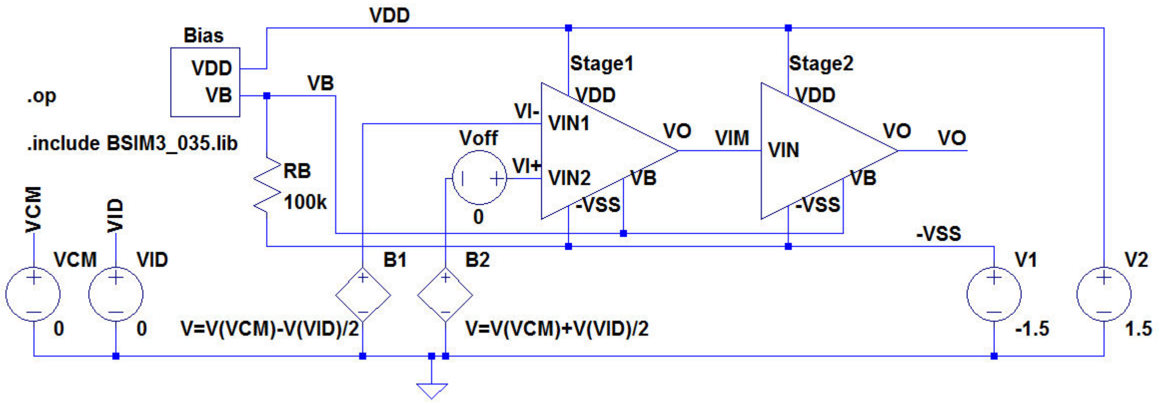


Figure 5.7: LTspice schematic at block level of the two stage operational amplifier.

Also, you may observe that when moving the mouse cursor over a subcircuit, the cursor turns into a hand . A right mouse click opens a dialogue box from which you can open either the symbol or the schematic for the subcircuit. Also, you can specify parameters for the subcircuit, see page 165.

In the schematic in fig. 5.7 we have just specified a '.op' simulation. Since the circuit has been designed with matching transistor geometries, we do not expect any systematic offset so the expected result from the '.op' simulation is a bias value of the output voltage which is within the useful output voltage range of the amplifier.

The output file from the simulation is shown in fig. 5.8.

When you have completed the '.op' simulation and closed the window with the output file, you can see currents and voltages in the circuit at block level 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. When you open a subcircuit schematic, you can also in this schematic point to nodes and see the voltages on the status bar but only for input and output nodes to the subcircuit. If you wish to see also internal node voltages and currents in the subcircuits, you must set up LTspice to save the subcircuit voltages and currents. This is done by the command 'Tools → Control Panel' where you select the tab 'Save Defaults'. Here you tick 'Save Subcircuit Node Voltages' and 'Save Subcircuit Device Currents'.

After running a new '.op' simulation, you will then be able to see also voltages and currents in the subcircuits, and the output file from the '.op' simulation will also include node voltages and device currents in the subcircuits.

In the error log file ('Ctrl-L'), you can see device currents and small signal parameters for all transistors in the circuit, even if you have not selected to save subcircuit node voltages and device currents.

```
Output file:

       - - - Operating point - - -

V(vim):                -0.81608        voltage
V(vi+):                0               voltage
V(vdd):                1.5             voltage
V(vi-):                0               voltage
V(vb):                 0.55281         voltage
V(-vss):               -1.5            voltage
V(vid):                0               voltage
V(vcm):                0               voltage
V(n001):               0               voltage
V(vo):                 -0.479621       voltage
I(B1):                 0               device_current
I(B2):                 0               device_current
I(RB):                 2.05281e-005    device_current
I(V2):                 -6.2341e-005    device_current
I(V1):                 6.2341e-005     device_current
I(Voff):               0               device_current
I(Vcm):                0               device_current
I(Vid):                0               device_current
Ix(stage1:-VSS):       -1.9599e-005    subckt_current
Ix(stage1:VB):         0               subckt_current
Ix(stage1:VDD):        1.9599e-005     subckt_current
Ix(stage1:VIN1):       0               subckt_current
Ix(stage1:VIN2):       0               subckt_current
Ix(stage1:VO):         -3.89635e-020   subckt_current
Ix(bias:VB):           -2.05281e-005   subckt_current
Ix(bias:VDD):          2.05281e-005    subckt_current
Ix(stage2:-VSS):       -2.22139e-005   subckt_current
Ix(stage2:VB):         0               subckt_current
Ix(stage2:VDD):        2.22139e-005    subckt_current
Ix(stage2:VIN):        0               subckt_current
Ix(stage2:VO):         -6.77626e-020   subckt_current
```

Figure 5.8: Output file from the '.op' simulation of the circuit in fig 5.7.

```
SPICE Error Log

Semiconductor Device Operating Points:
                          - - - BSIM3 MOSFETS - - -
Name:          m:stage2:1   m:stage1:3   m:stage1:4   m:stage2:2     m:bias:1
Model:          nmos-bsim    nmos-bsim    nmos-bsim    pmos-bsim    pmos-bsim
Id:              2.22e-05     9.80e-06     9.80e-06    -2.22e-05    -2.05e-05
Vgs:             6.84e-01     6.84e-01     6.84e-01    -9.47e-01    -9.47e-01
Vds:             1.02e+00     6.84e-01     6.84e-01    -1.98e+00    -9.47e-01
Vbs:             0.00e+00     0.00e+00     0.00e+00     0.00e+00     0.00e+00
Vth:             5.39e-01     5.43e-01     5.43e-01    -6.79e-01    -6.81e-01
Vdsat:           1.11e-01     1.07e-01     1.07e-01    -2.52e-01    -2.51e-01
Gm:              3.52e-04     1.60e-04     1.60e-04     1.46e-04     1.36e-04
Gds:             1.98e-06     9.78e-07     9.78e-07     1.41e-06     1.98e-06
Gmb:             1.04e-04     4.73e-05     4.73e-05     3.43e-05     3.21e-05
Cbd:             1.71e-14     9.38e-15     9.38e-15     8.17e-15     1.04e-14
Cbs:             2.22e-14     1.14e-14     1.14e-14     1.52e-14     1.52e-14
Cgsov:           5.43e-15     2.67e-15     2.67e-15     2.06e-15     2.06e-15
Cgdov:           5.43e-15     2.67e-15     2.67e-15     2.04e-15     2.04e-15
Cgbov:           9.99e-19     9.99e-19     9.99e-19     1.00e-18     1.00e-18
dQgdVgb:         8.01e-14     3.94e-14     3.94e-14     3.82e-14     3.82e-14
dQgdVdb:        -5.44e-15    -2.68e-15    -2.68e-15    -1.99e-15    -2.00e-15
dQgdVsb:        -6.89e-14    -3.39e-14    -3.39e-14    -3.35e-14    -3.35e-14
dQddVgb:        -3.38e-14    -1.66e-14    -1.66e-14    -1.64e-14    -1.64e-14
dQddVdb:         2.25e-14     1.21e-14     1.21e-14     1.02e-14     1.24e-14
dQddVsb:         3.77e-14     1.85e-14     1.85e-14     1.79e-14     1.79e-14
dQbdVgb:        -1.26e-14    -6.25e-15    -6.25e-15    -5.44e-15    -5.43e-15
dQbdVdb:        -1.71e-14    -9.39e-15    -9.39e-15    -8.17e-15    -1.04e-14
dQbdVsb:        -3.42e-14    -1.72e-14    -1.72e-14    -1.96e-14    -1.96e-14

Name:          m:stage1:5   m:stage1:1   m:stage1:2
Model:          pmos-bsim    pmos-bsim    pmos-bsim
Id:             -1.96e-05    -9.80e-06    -9.80e-06
Vgs:            -9.47e-01    -9.59e-01    -9.59e-01
Vds:            -5.41e-01    -1.78e+00    -1.78e+00
Vbs:             0.00e+00     5.41e-01     5.41e-01
Vth:            -6.82e-01    -7.96e-01    -7.96e-01
Vdsat:          -2.50e-01    -1.81e-01    -1.81e-01
Gm:              1.30e-04     9.62e-05     9.62e-05
Gds:             2.72e-06     7.66e-07     7.66e-07
Gmb:             3.06e-05     1.78e-05     1.78e-05
Cbd:             1.19e-14     7.70e-15     7.70e-15
Cbs:             1.52e-14     1.19e-14     1.19e-14
Cgsov:           2.06e-15     2.06e-15     2.06e-15
Cgdov:           2.04e-15     2.04e-15     2.04e-15
Cgbov:           1.00e-18     1.00e-18     1.00e-18
dQgdVgb:         3.82e-14     3.74e-14     3.74e-14
dQgdVdb:        -2.04e-15    -1.99e-15    -1.99e-15
dQgdVsb:        -3.35e-14    -3.28e-14    -3.28e-14
dQddVgb:        -1.64e-14    -1.62e-14    -1.62e-14
dQddVdb:         1.39e-14     9.71e-15     9.71e-15
dQddVsb:         1.79e-14     1.69e-14     1.69e-14
dQbdVgb:        -5.40e-15    -4.85e-15    -4.85e-15
dQbdVdb:        -1.19e-14    -7.69e-15    -7.69e-15
dQbdVsb:        -1.96e-14    -1.49e-14    -1.49e-14
```

Figure 5.9: Error log file from the '.op' simulation of the circuit in fig 5.7.

For the circuit from fig. 5.7, all transistors are in the subcircuits, and the error log file identifies the transistors by subcircuit name and device number in the subcircuit as shown in fig. 5.9.

Example 5.2: Designing the two stage opamp for an inverting feedback amplifier.

In this example, we show how the two stage opamp can be designed to a specific set of design requirements using a combination of analytical methods and LTspice simulations. For the design

example, we assume the following design requirements: The opamp is to be used in an inverting amplifier configuration with a capacitive feedback network as shown in fig. 5.10. With a high gain of the opamp at low frequencies, the midband gain of the amplifier is $A_v = V_o/V_{in} \simeq -C_1/C_2$.
The specifications to be met are the following:

Midband gain:	14 dB
Bandwidth:	20 MHz
Input capacitance:	1 pF
Load capacitance:	1.5 pF
Slew rate:	$\geq$ 30 V/µms
Phase margin:	$\geq 65^\circ$
Positive supply voltage:	1.5 V
Negative supply voltage:	-1.5 V
Technology:	0.35 µm CMOS process with BSIM3 transistor models, see fig. 3.10 on page 77

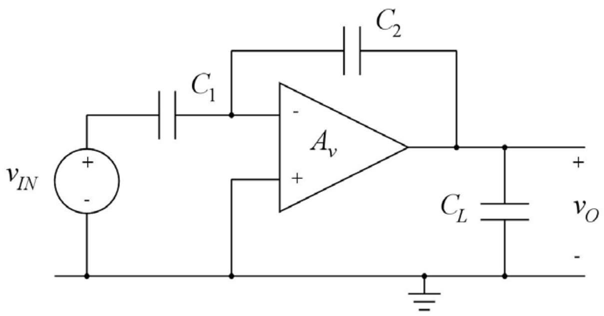


Figure 5.10: Inverting opamp configuration with capacitive feedback.

For the design, the starting point is the schematic shown in fig. 5.2. The design parameters are all transistor dimensions, the compensation capacitor C_c, and the bias current set by R_B. Obviously, this is a large number of design variables, and some analytical design methodology is needed. There is no way to find a reasonable set of transistor dimensions and bias currents simply through iterative simulation with sweeping of the transistor parameters.

We note that the requirement concerning the input capacitance and the midband gain leads to $C_1 = 1$ pF and $C_2 = 0.2$ pF. The other design requirements determine the design as follows:

– The bandwidth specification, *BW*, puts a constraint on the transconductance of the differential input transistors and the compensation capacitor C_c which determine the unity gain bandwidth of the opamp.

– The specification for the phase margin, *PM*, puts constraints on the location of non-dominant poles and zeros in the opamp.

– The slew rate specification, *SR*, puts constraints on the bias current for M_5 and M_6 in fig. 5.2. The bias currents must be large enough to slew the voltages on the load capacitance, the feedback capacitors and the compensation capacitor.

For the analytical approach, we base the design on the analysis of the two stage opamp presented in (Carusone, Johns & Martin 2012, chapter 6). Examining the feedback configuration, we find that the loop gain $L(s)$ is determined from the schematic shown in fig. 5.11 as $L(s) = V_r(s)/V_t(s) = A_v(s) \times C_2/(C_1+C_2)$. We also find that the total capacitive load at the output of the amplifier is the load capacitor C_L in parallel with the feedback network which is a series connection of C_1 and C_2, i.e. the total capacitive load is $C_L' = C_L + C_1C_2/(C_1+C_2)$.

Assuming that the transfer function $A_v(s)$ of the opamp has a low frequency gain A_0, a dominant pole at the frequency ω_{p1}, a non-dominant pole at the frequency ω_{p2} and a right half plane zero at

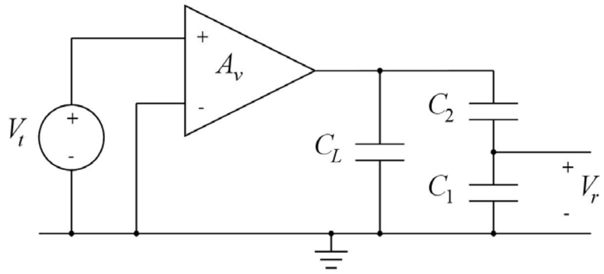


Figure 5.11: Open loop circuit for finding the loop gain $L(s) = V_r(s)/V_t(s)$.

the frequency ω_z, the loop gain is given by

$$L(s) = \frac{C_2}{C_1 + C_2} \times \frac{A_0(1 - s/\omega_z)}{(1 + s/\omega_{p1})(1 + s/\omega_{p2})} \tag{5.1}$$

In this expression, all higher order poles and zeros are neglected. With $\omega_{p2} \gg \omega_{p1}$, the loop gain can be approximated by

$$L(s) = \frac{C_2}{C_1 + C_2} \times \frac{A_0\omega_{p1}(1 - s/\omega_z)}{s(1 + s/\omega_{p2})} \tag{5.2}$$

for frequencies $\omega \gg \omega_{p1}$.

From (Carusone, Johns & Martin 2012) we find approximate expressions for A_0, ω_{p1}, ω_{p2} and ω_z as follows:

$$\begin{aligned}
A_0 &\simeq \frac{g_{m1}}{g_{ds2} + g_{ds4}} \times \frac{g_{m7}}{g_{ds6} + g_{ds7}} && (5.3)\\
\omega_{p1} &\simeq \frac{(g_{ds2} + g_{ds4})(g_{ds6} + g_{ds7})}{g_{m7}C_c} && (5.4)\\
\omega_{p2} &\simeq \frac{g_{m7}}{C'_L} && (5.5)\\
\omega_z &\simeq \frac{g_{m7}}{C_c} && (5.6)
\end{aligned}$$

where the numbering of the transistor small signal parameters refers to the transistor numbering from fig. 5.2.

The bandwidth BW of the amplifier (with feedback) can be estimated by $2\pi BW = \omega_{p1}(1 + L_0) \simeq \omega_{p1}L_0$ where $L_0 = A_oC_2/(C_1 + C_2)$ is the loop gain at low frequencies. From this, we find

$$2\pi BW = \frac{g_{m1}}{C_c} \times \frac{C_2}{C_1 + C_2} \tag{5.7}$$

With the loop gain expression given by (5.2), we find a phase margin PM of

$$PM = 180^\circ - 90^\circ - \arctan(\omega_t/\omega_z) - \arctan(\omega_t/\omega_{p2}) = 180^\circ + \angle L(j\omega_t) \tag{5.8}$$

where ω_t is the unity gain frequency for the loop gain.

We may start by selecting $\omega_z \gg \omega_t$ to reduce the influence of the zero. With $\omega_z \simeq 10 \times \omega_t$, the phase shift from the zero is about 6°. Also, additional high order poles will contribute to the phase shift. Assuming (somewhat arbitrarily) an additional phase shift from higher order poles of 4°, this leaves a phase shift of about 15° from the second pole for a phase margin of 65°. From this, we get

$$\arctan(\omega_t/\omega_{p2}) = 15^\circ \Rightarrow \omega_{p2} = 3.8 \times \omega_t \tag{5.9}$$

With the non-dominant pole and the zero well above the unity gain frequency ω_t of the loop gain, ω_t is approximately given by the gain bandwidth product of the loop gain, i.e.

$$\omega_t = L_0 \times \omega_{p1} = \frac{g_{m1}}{C_c} \times \frac{C_2}{C_1 + C_2} = 2\pi \cdot 20 \text{ MHz} \tag{5.10}$$

Note that the unity gain frequency of the opamp is

$$\omega_{ta} = A_0 \omega_{p1} = (1 + C_1/C_2) L_0 \omega_{p1} = 2\pi \cdot 120 \text{ MHz} \tag{5.11}$$

Using $\omega_z = 10 \times \omega_t = 2\pi \cdot 200$ MHz in combination with (5.6) and (5.10), we find

$$\frac{g_{m7}}{C_c} = 10 \times \frac{g_{m1}}{C_c} \times \frac{C_2}{C_1 + C_2} \Rightarrow g_{m7} = \frac{10 \times C_2}{C_1 + C_2} g_{m1} = 1.67 \times g_{m1} \tag{5.12}$$

From (5.5), (5.9), (5.10) and (5.12) we find

$$\frac{g_{m7}}{C_L'} = 3.8 \times \frac{g_{m1}}{C_c} \times \frac{C_2}{C_1 + C_2} \Rightarrow C_c = 0.38 \times C_L' = 0.63 \text{ pF} \tag{5.13}$$

Then, using (5.7) and (5.12) we find

$$g_{m1} = 2\pi BW \times C_c (1 + C_1/C_2) = 0.48 \text{ mA/V} \tag{5.14}$$

$$g_{m7} = 1.67 \times g_{m1} = 0.80 \text{ mA/V} \tag{5.15}$$

Next, we consider the slew rate specification. The bias current for M_5 is the maximum current available for charging and discharging of C_c, and the bias current for M_6 is the maximum current for for charging the output node when $v_{IN+} - v_{IN-}$ is positive. In this situation, I_{D6} charges C_c, C_L and the series connection of C_1 and C_2. From this, we find

$$\begin{aligned} SR &= \frac{I_{D5}}{C_c} \\ &\Rightarrow I_{D5} \geq SR \times C_c = 19\ \mu\text{A} \end{aligned} \tag{5.16}$$

$$\begin{aligned} SR &= \frac{I_{D6}}{C_c + C_L + C_1 C_2/(C_1 + C_2)} \\ &\Rightarrow I_{D6} \geq SR \times (C_c + C_L + C_1 C_2/(C_1 + C_2)) = 70\ \mu\text{A} \end{aligned} \tag{5.17}$$

These values of bias current are minimum values, and it may be a good idea to select somewhat larger bias current values in order to leave room for design iterations with increased values of C_c.

The bias current for M_1 and M_2 is obviously $I_{D5}/2$ and the bias current for M_7 is $I_{D7} = I_{D6}$, so for M_1, M_2 and M_7 we have found both the transconductances and the minimum bias currents, and using (3.8) on page 72, the ratio W/L may be found from the Shichman-Hodges transistor model. However, we noticed in tutorial 3 that it is not easy to obtain a good match over a wide range of variation in g_m between the Shichman-Hodges model and the BSIM3 transistor models specified for the opamp. So, rather than trying to adapt the Shichman-Hodges parameters to fit the BSIM model, we may use LTspice to find the transconductance g_m versus channel width W for the desired values of bias current. First, we need to select a value for the channel length, and for this design we chose (somewhat arbitrarily) L to be about three times the minimum dimensions specified for the process, i.e. $L = 1$ μm for all transistors in the circuit.

In order to simulate g_m versus W, we use the approach described in Example 3.6, page 90. As we need only approximate results and as g_m is much larger than g_{ds} for the BSIM3 transistors with $L = 1$ μm in the active region, we may just run a '.tf' simulation on a single diode connected transistor as shown in fig. 5.12. For the diode connected transistor, the input resistance is $(g_m + g_{ds})^{-1} \simeq 1/g_m$, so stepping W and plotting '`1/id#input_resistance`' directly shows g_m versus W.

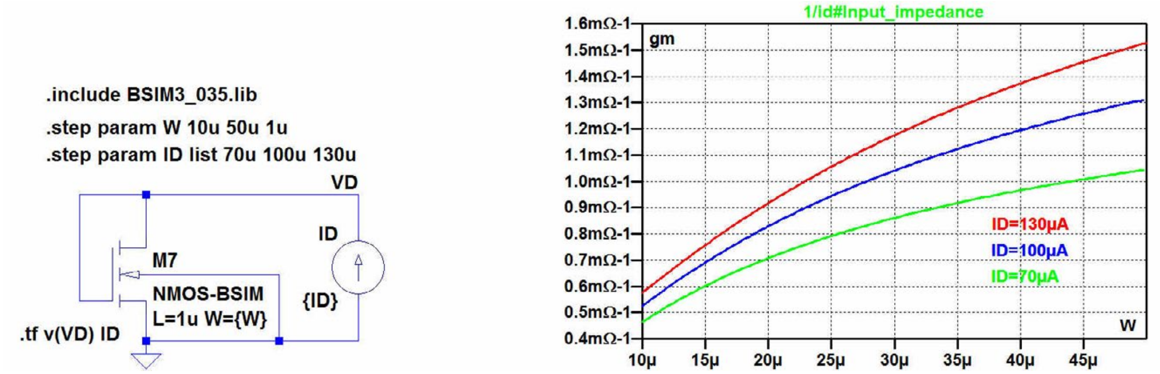


Figure 5.12: Simulation showing g_m versus W for an NMOS transistor with $L = 1$ µm using the BSIM3 transistor model.

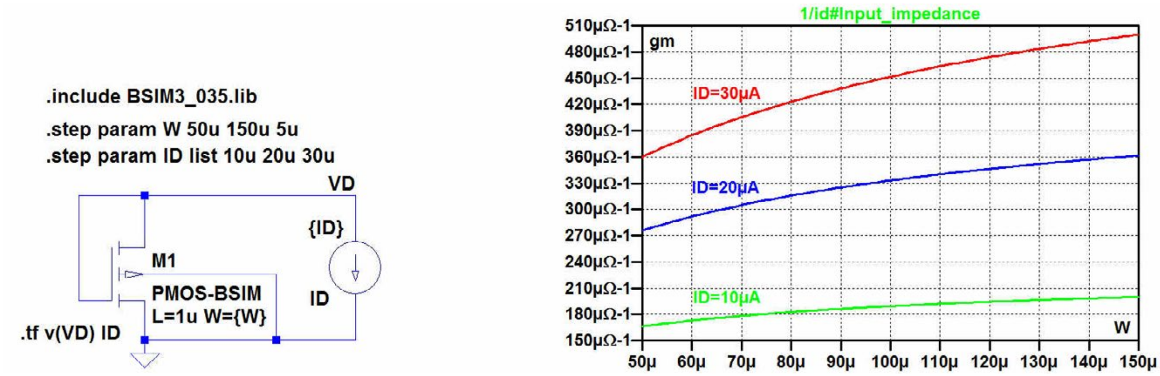


Figure 5.13: Simulation showing g_m versus W for a PMOS transistor with $L = 1$ µm using the BSIM3 transistor model.

In fig. 5.12, also the drain current is defined as a parameter, and g_m is shown versus W for three different values of the bias current I_D. For the design, we select $I_{D7} = 100$ µA in order to have some margin for the slew rate and for design iterations. With this value of I_{D7}, we select $W_7 = 20$ µm.

Fig. 5.13 shows a similar simulation for a PMOS transistor. From this, we see that the required value for g_{m1} cannot be obtained with reasonable transistor geometries with a bias current of 10 µA. Instead, we select a bias current of 30 µA, and with this bias current, we can select the channel width for M_1 and M_2 to be 130 µm.

In order to avoid systematic offset errors, the same current density $I_D/(W/L)$ should be selected for M_3, M_4 and M_7, so $W_3 = W_4 = W_7(I_{D3}/I_{D7}) = 6$ µm.

Also, M_8, M_5 and M_6 must have the same current density in order to avoid systematic offset errors. Thus, $W_6 = W_5(I_{D6}/I_{D5}) = W_5 \times 100/60$. Selecting the bias current in R_B and M_8 to be the same as the bias current in M_1 and M_2 (i.e. half the current in M_5), we find $W_8 = W_5/2$. For these transistors, a small value of $|V_{GS} - V_t|$ will give a large output voltage range and input voltage range, and a large

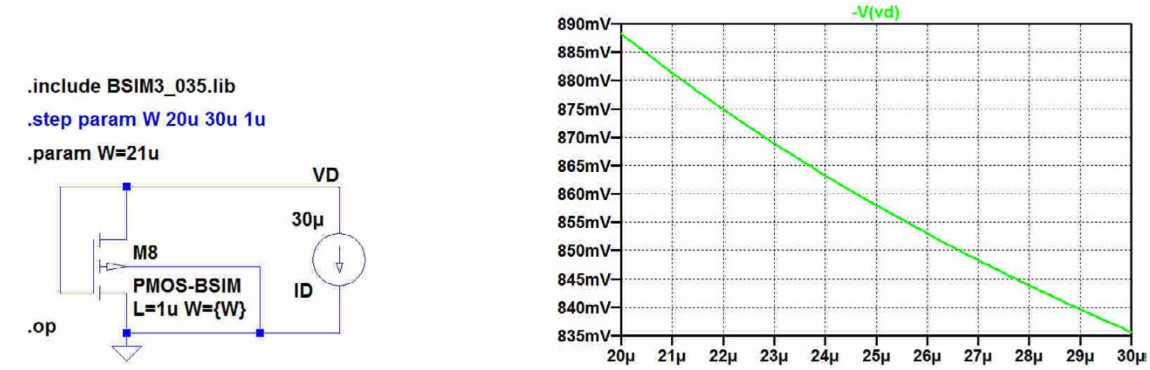


Figure 5.14: Simulation showing $|V_{GS}|$ versus W for a PMOS transistor with $L = 1$ µm and $I_D = 30$ µA using the BSIM3 transistor model.

value of $|V_{GS} - V_t|$ will improve the matching between the transistors. A reasonable compromise may be $|V_{GS} - V_t| \simeq 200$ mV. Fig. 5.14 shows a simulation of $|V_{GS}|$ versus W_8 for $I_{D8} = 30$ µA, and from this we select $W_8 = 21$ µm. The value of $|V_{GS} - V_t|$ is verified by a '.op' simulation with $W_8 = 21$ µm. From the error log file, we find $|V_{GS} - V_t| = 211$ mV which is acceptable. With $W_8 = 21$ µm, we find $W_5 = 42$ µm and $W_6 = 70$ µm.

With all transistor dimensions in place, the complete opamp is now ready for simulation. The transistor channel widths are summarized in the table below which also shows the calculated bias current for each transistor and g_m values for the transistors where g_m is included in the design equations.

Transistor number	1	2	3	4	5	6	7	8
Channel width	130 µm	130 µm	6 µm	6µm	42 µm	70 µm	20 µm	21 µm
Bias current	30 µA	30 µA	30 µA	30 µA	60 µA	100 µA	100 µA	30 µA
Transconductance	0.48 mA/V	0.48 mA/V					0.80 mA/V	

Table 5.1: Calculated transistor parameters for initial simulations.

Both the closed loop performance and the loop gain must be simulated for a verification of the design requirements. In order to be able to modify the design, all transistor channel widths are defined as parameters at the top level schematic so that relations between the widths can be used when defining the parameters, and assuming that the input capacitance of the opamp is much smaller than C_1 and C_2, two different test benches for the simulation of loop gain and closed loop gain can be defined as shown in figs. 5.15 and 5.16.

Also the drain and source areas and perimeters are calculated using the channel width as the input parameter. For the drain and source regions, we assume an area of $W \times 1$ µm and a perimeter of $W + 2$ µm, compare page 75. Rather than specifying a value of the resistor R_B, the bias current is

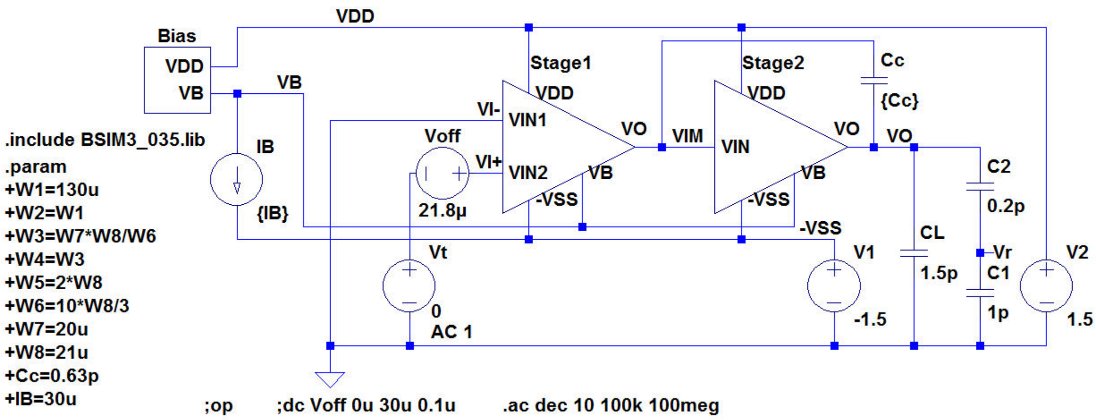


Figure 5.15: Test bench for simulation of loop gain.

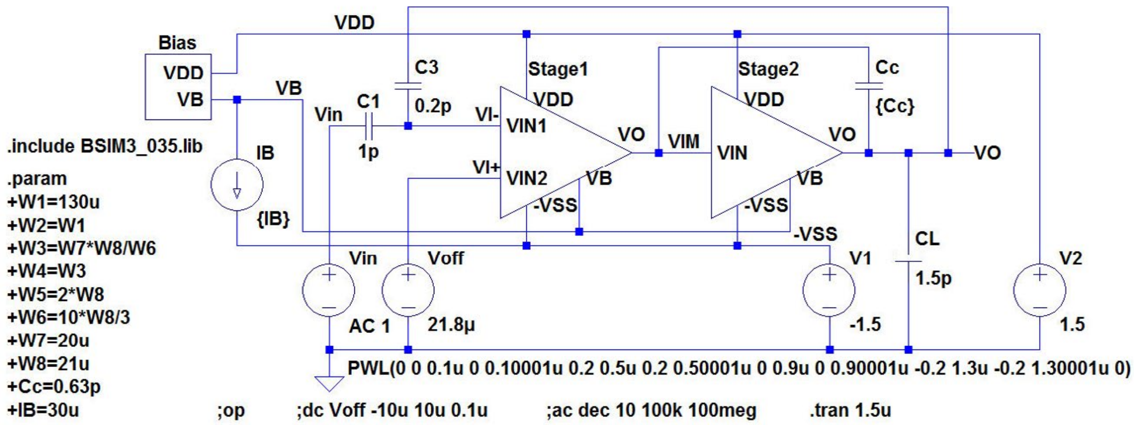


Figure 5.16: Test bench for simulations of closed loop response.

directly set by a current source I_B. For both schematics, a DC offset voltage is inserted in series with the input, and a DC sweep of the offset voltage is the first simulation to run in order to find an offset voltage resulting in an output voltage close to 0 V. This is important not only for the loop gain simulation but also for the closed loop simulations since there is no DC feedback to ensure a proper bias point. Subsequently, a '.op' simulation is run in order to verify the bias points of all transistors. The error log file from the '.op' simulation provides information about transistor bias currents and small signal parameters. The error log file also gives a warning that the common node for C_1 and C_2 is floating, but since the voltage calculated for this node is 0, the warning may be neglected, compare example 2.3 on page 53.

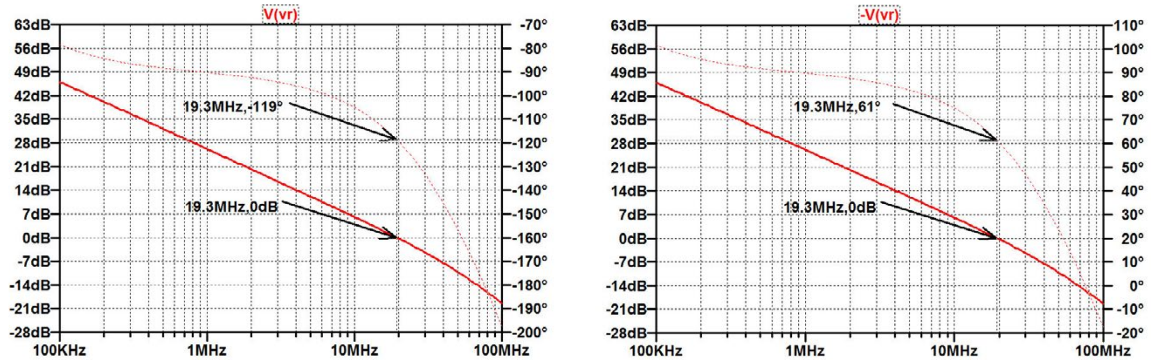
Table 5.2 shows the simulated transistor parameters. Comparing to the values from table 5.1, we find a reasonable match between calculated and simulated parameters. From the error log file, you can

Transistor number	1	2	3	4	5	6	7	8
Channel width	130 μm	130 μm	6 μm	6 μm	42 μm	70 μm	20 μm	21 μm
Bias current	31.80 μA	31.8 μA	31.8 μA	31.8 μA	63.5 μA	119 μA	119 μA	30 μA
Transconductance	0.51 mA/V	0.51 mA/V					0.89 mA/V	

Table 5.2: Simulated transistor parameters from the initial '.op' simulation of fig. 5.15.

also find the parasitic transistor capacitances calculated on basis of the transistor dimensions. They are all much smaller than C_1, C_2 and C_c but some of them on the order of 0.1pF can be expected to affect the position of the zero and the second pole and also introduce additional pole(s) from the input stage. This will cause an additional phase shift of the loop gain.

After having verified the bias point, a '.ac' simulation is run to find the loop gain and the phase margin. Fig. 5.17(a) shows the resulting gain and phase response of the loop gain $L(j\omega)$ or '`V(vr)`'. Fig. 5.17(b) shows a Bode plot of $-L(j\omega)$ or '`-V(vr)`'. Since $\angle(-L(j\omega)) = 180^\circ + \angle L(j\omega) = PM$ for $\omega = \omega_t$, this plot directly shows the phase margin at the frequency where the amplitude is 0 dB. From these plots, we find a phase margin of 61° which is slightly lower than the design specification. Before modifying the design, we also run a '.ac' simulation and a '.tran' simulation of the closed loop circuit, fig. 5.16, in order to find the bandwidth and the slew rate of the amplifier with feedback. The results of these simulations are shown in fig. 5.18, and we notice a bandwidth of 26.8 MHz and



(a) (b)

Figure 5.17: Simulated loop gain showing the phase margin, $C_c = 0.63$ pF. (a) shows the loop gain $L(j\omega)$. (b) shows $-L(j\omega)$ where the phase is equal to the phase margin for the frequency where the amplitude is 0 dB.

a slew rate well above the minimum specification of 30 V/µs. Also, we see that the rising edge slew rate is smaller than the falling edge slew rate. This is not surprising since the rising edge slew rate is limited by I_{D6} and the falling edge slew rate is limited by I_{D5}. In the design, the relative increase of the bias current over the calculated minimum levels is larger for I_{D5} than for I_{D6}.

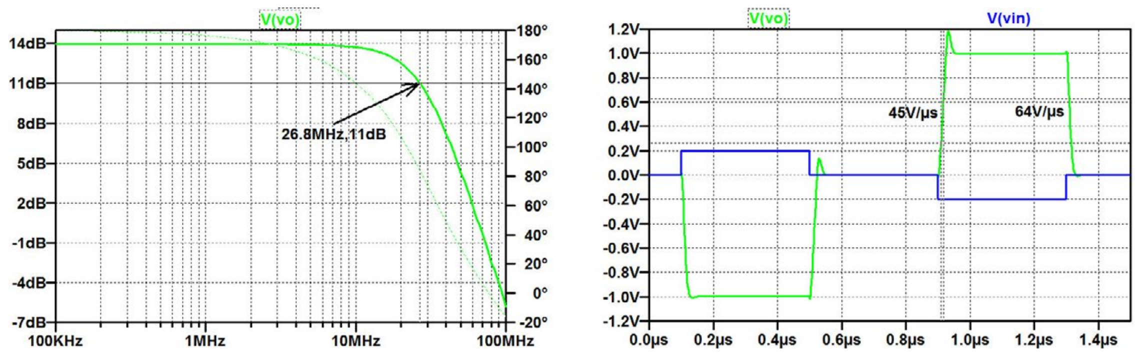


Figure 5.18: Simulated frequency response and transient response for the opamp with feedback, $C_c = 0.63$ pF.

With the values found for bandwidth and slew rate, an obvious modification of the design is an increase of the compensation capacitor C_c. This decreases the bandwidth and the slew rate. The bandwidth is approximate inversely proportional to C_c, so a new value of C_c should be about 0.84 pF in order to obtain a -3 dB frequency of 20 MHz. A few simulations with different values of C_c show that with $C_c = 0.80$ pF, we achieve the frequency response and transient response shown in fig. 5.19. A re-simulation of the loop gain with $C_c = 0.80$ pF results in the loop gain response shown in fig. 5.20. From figs. 5.19 and 5.20, we find that the design requirements for the opamp listed on page 155 are met, so this concludes the design of the opamp.

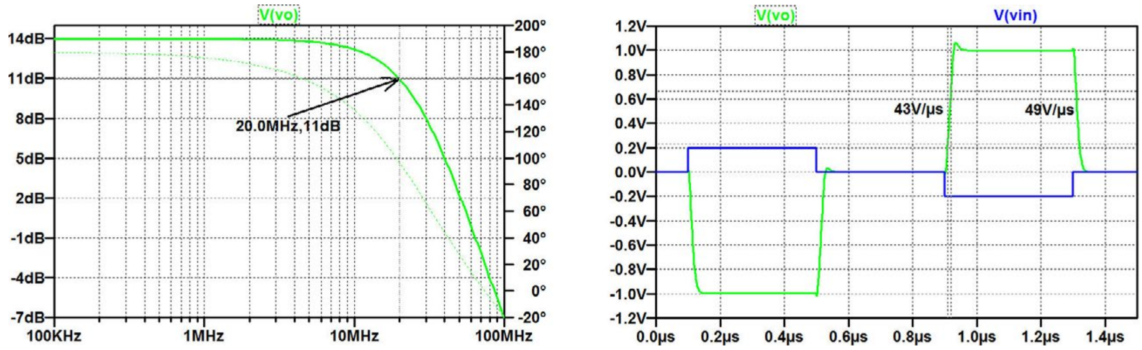
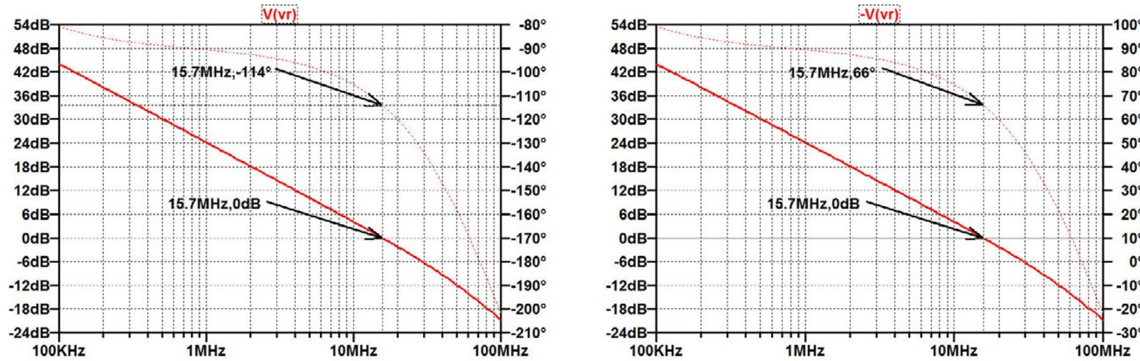


Figure 5.19: Simulated frequency response and transient response for the opamp with feedback, $C_c = 0.80$ pF.



(a) (b)

Figure 5.20: Simulated loop gain showing the phase margin, $C_c = 0.80$ pF. (a) shows the loop gain $L(j\omega)$. (b) shows $-L(j\omega)$ where the phase is equal to the phase margin for the frequency where the amplitude is 0 dB.

Example 5.3: Generic filter blocks.

In this example, we show a few generic filter blocks which can be used for simulation of filter responses. The blocks are designed so that the filter characteristics can be defined by specifying parameters for each filter block. The blocks are shown in table 5.3 where the transfer function in the frequency domain is specified for each block. Also, the schematic and the symbol for each block is shown. The resistor at the output in each of the filter blocks does not affect the transfer function but it prevents LTspice from producing a warning in the error log file (‘`WARNING: Less than two connections ...`’) concerning connections to the output node of a filter with no load connected at the output.

When using the filter blocks, you can obviously encounter situations with multiple use of the same subcircuit. In this case you must be able to specify different parameter values for the different instances of the subcircuit. This is done by a right mouse click on the subcircuit symbol. This opens a window as shown in fig. 5.21 where you can specify parameters for the subcircuit in the line

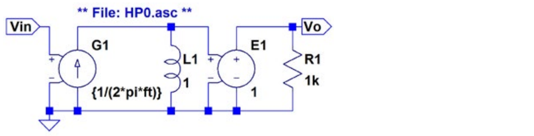
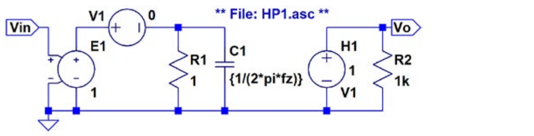
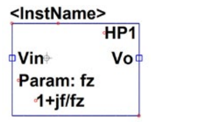
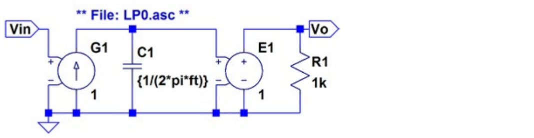
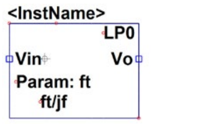
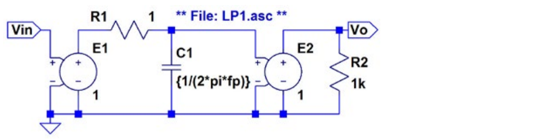
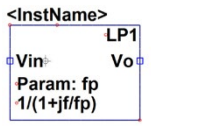
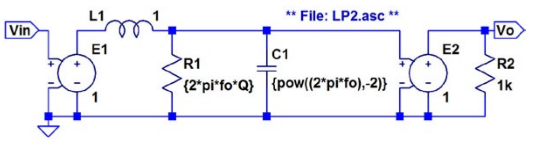
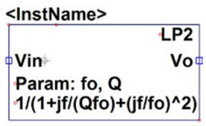
Transfer function	**Schematic**	**Symbol**	**Parameter**
Zero in 0 $T(jf) = jf/f_t$			Unity gain frequency f_t
Real zero $T(jf) =$ $1 + jf/f_z$			Zero frequency f_z
Pole in 0 $T(jf) = f_t/(jf)$			Unity gain frequency f_t
Real pole $T(jf) =$ $\frac{1}{1+jf/f_p}$			Pole frequency f_p
Biquad $T(jf) =$ $\frac{1}{(jf)^2+(jf)f_0/Q+f_0^2}$			Resonance frequency f_0 and quality factor Q

Table 5.3: Generic filter blocks defined as subcircuits.

'`PARAMS:`'. Also tick the box next to the specification line in order to make the parameters visible on the schematic. In the specification window shown in fig. 5.21, the parameter f_p (pole frequency) has been specified for a subcircuit '`LP1.asc`' (low pass filter with a single, real pole). From the specification window, you can also open the schematic and the symbol for the subcircuit.

A simple example of a circuit using subcircuits from table 5.3 is the audio amplifier shown in fig. 5.22 on page 168. The overall transfer function is

$$T(jf) = \frac{V_o(jf)}{V_{in}(jf)} = \frac{jf/f_t}{(1+jf/f_{p1})(1+jf/f_{p2})} \tag{5.18}$$

where $f_{p1} = 20$ Hz is the lower -3 dB frequency and $f_{p2} = 20$ kHz is the upper -3 dB frequency. The midband gain is $A = f_{p1}/f_t$, so with $f_t = 2$ Hz, the amplifier has a midband gain of 10 V/V (or 20 dB). Fig. 5.22 also shows the resulting frequency response from a '.ac' simulation.

The filter blocks shown in table 5.3 are also very suitable for the investigation of a feedback amplifier as designed in Example 5.2 (page 154). From (5.2), we find that the transfer function of the amplifier is given by

$$A_v(jf) = \frac{A_0 f_{p1}(1 - jf/f_z)}{jf(1 + jf/f_{p2})} = \frac{f_{ta}(1 - jf/f_z)}{jf(1 + jf/f_{p2})} \tag{5.19}$$

where $f_{ta} = 120$ MHz, $f_z = 200$ MHz and $f_{p2} = 3.8 \cdot 20$ MHz $= 76$ MHz. In order to model the right half plane zero, we just use a negative value of the zero frequency for a filter block of type HP1.asc, see table 5.3. Using the circuits from fig. 5.10 and 5.11 we can then find closed loop gain and loop gain with the approximate loop gain transfer function given by (5.2). This is an exercise left for the reader (Problem 5.2 on page 176).

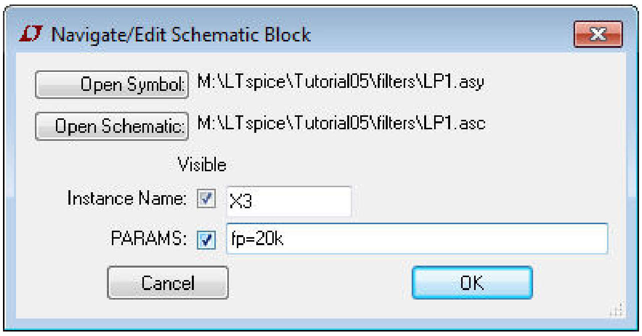


Figure 5.21: Window for parameter specification for subcircuit.

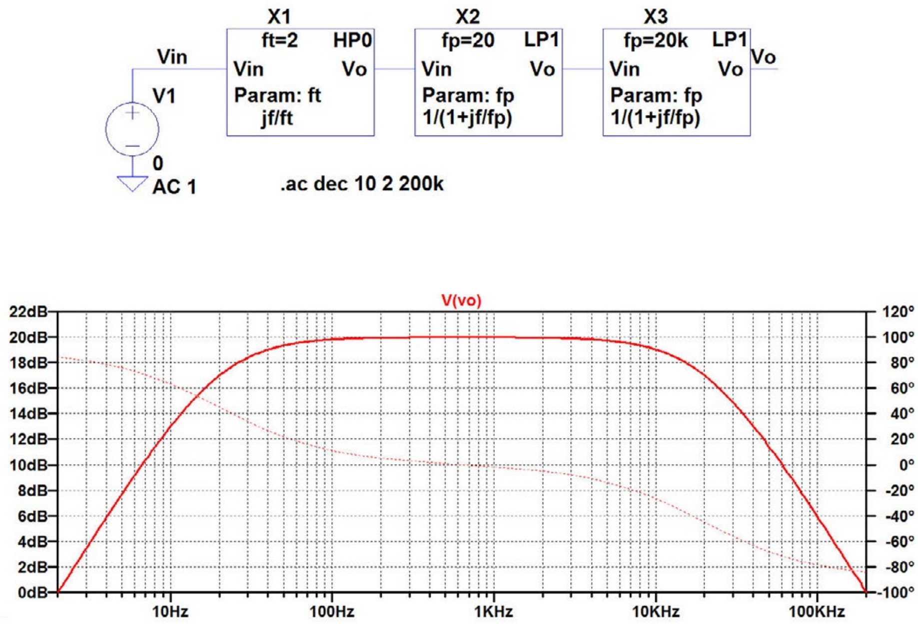


Figure 5.22: Audio amplifer (schematic and frequency response).

Example 5.4: A mixed analog/digital circuit.

The final example in this tutorial is a 3-bit resistor string D/A converter (Carusone, Johns & Martin 2012) with a decoder as shown in fig. 5.23. The circuit requires a digital 3 to 1-of-8 decoder, a resistor string and 8 analog switches. The decoder is built from inverters and NAND gates while the analog switches require transistors and inverters.

Thus, at transistor level we need a 3-input NAND gate and an inverter as subcircuits. These are shown in figs. 5.24 and 5.25 together with the associated symbols. The circuits have been designed like the gain stages in example 5.1 (page 147), only the 'Port Type' definition for V_{DD} (the supply voltage) has been left empty. This implies that V_{DD} does not appear as a terminal in the symbols for the subcircuits, and at the top level of the circuit hierarchy, V_{DD} must be declared as a global node using the SPICE Directive '`.global VDD`'. Also, for the digital subcircuits, the symbols have been edited to the conventional digital gate symbols, and the names of inputs (`X1` – `X3`) and output (`Y`) are not shown in the symbol. This is achieved by right clicking on the pin in the symbol editor and selecting 'Not Visible' in the 'Pin/Port Properties' window. For the transistor dimensions, we have used the dimensions shown in the schematics. Note that the channel length is specified as a parameter '`Lmin`' which must be defined at the top level of the circuit hierarchy, and so must the

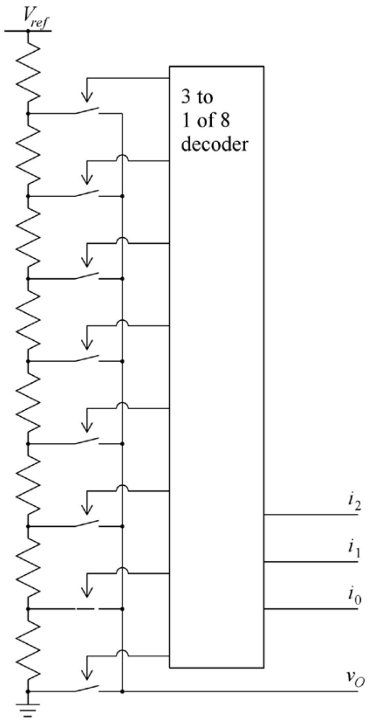


Figure 5.23: Resistor string D/A converter. The converter has the binary inputs i_0, i_1 and i_2 and the analog output voltage v_O. It is shown with a binary input value of 001.

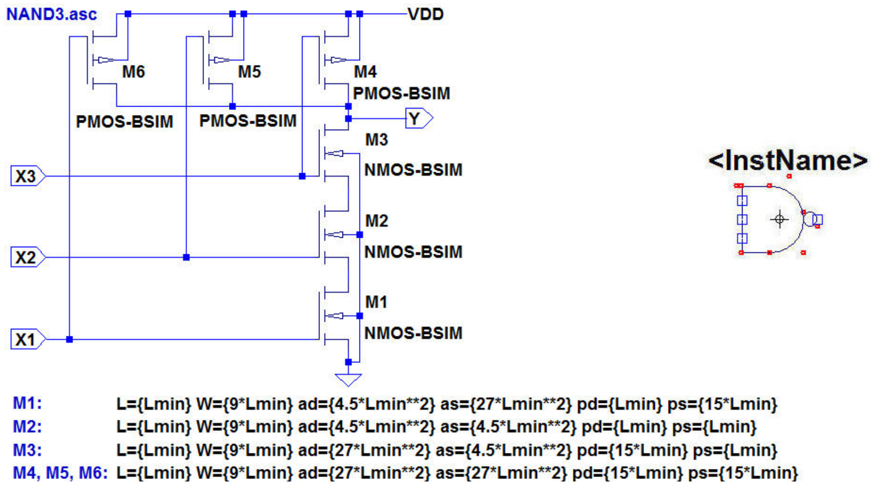


Figure 5.24: Three-input NAND gate, schematic and symbol.

transistor model file for '`NMOS-BSIM`' and '`PMOS-BSIM`'. With this approach, the same subcircuits can be used for different technologies with different minimum dimensions, e.g. a 0.35 µm process (`Lmin=0.35u`) or a 0.18 µm process (`Lmin=0.18u`), for which a BSIM3 model is also provided

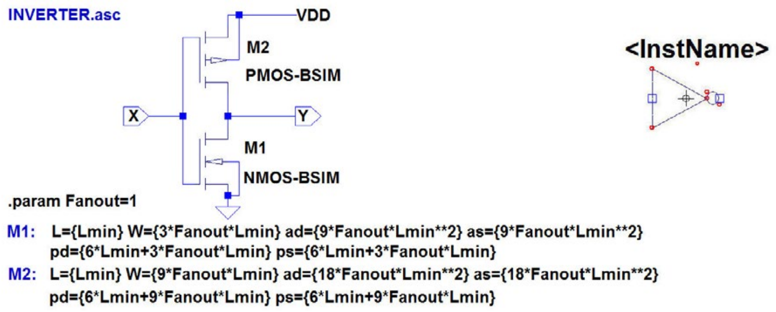


Figure 5.25: Inverter, schematic and symbol.

in (Carusone, Johns & Martin 2014). More examples of model files for different values of ‘Lmin’ can be found in (Baker 2010) and in (The MOSIS Service: Wafer Electrical Test Data and SPICE Model Parameters 2014). Additionally, transistor widths have been scaled for PMOS and NMOS transistors to compensate for the difference in hole mobility μ_p and electron mobility μ_n.

For the inverter, an additional parameter ‘Fanout’ has been used. With this parameter, the transistor dimensions in the inverter can be scaled when the inverter is used as a buffer driving a large number of inputs or a large capacitive load. The default parameter Fanout=1 is defined in the inverter subcircuit. When a different value of Fanout is required, it is specified when inserting the inverter in the higher level schematic using the specification window shown in fig. 5.21 on page 167. This specification will override the specification given in fig. 5.25.

With these subcircuits defined, the decoder and the analog switch are easily designed as shown in figs. 5.26 and 5.27. As the decoder has an active low output, the analog switch has been designed to have an active low input, indicated by a small circle in the symbols.

Finally, the complete D/A converter is shown in fig. 5.28. The circuit has three levels of hierarchy. The supply voltage is inserted as shown at the top level of the hierarchy with the command '`.global`

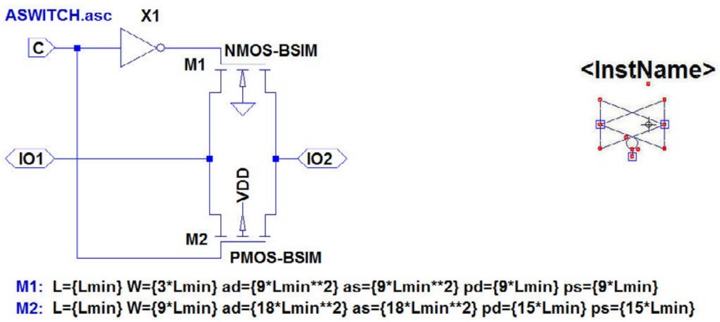


Figure 5.26: Analog switch, schematic and symbol.

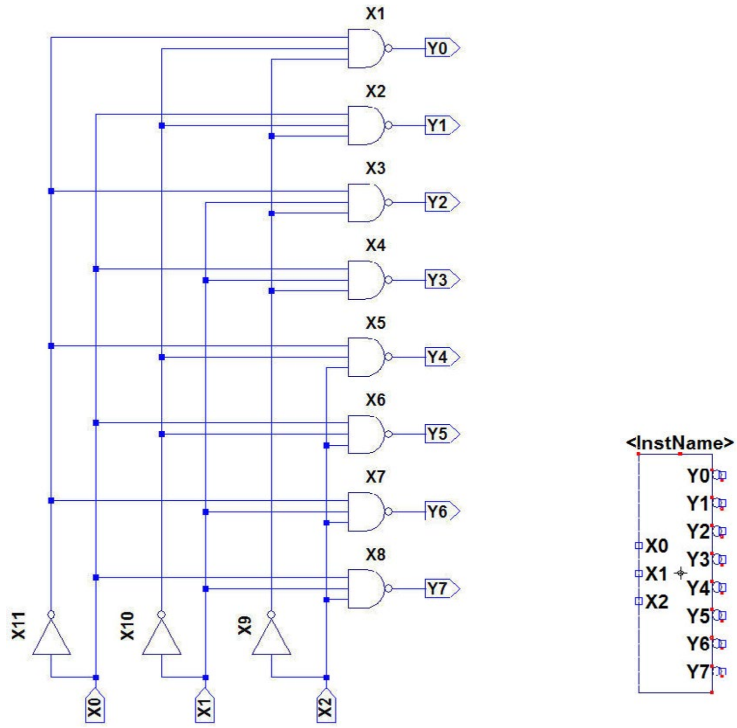


Figure 5.27: 1 to 1-of-8 decoder, schematic and symbol.

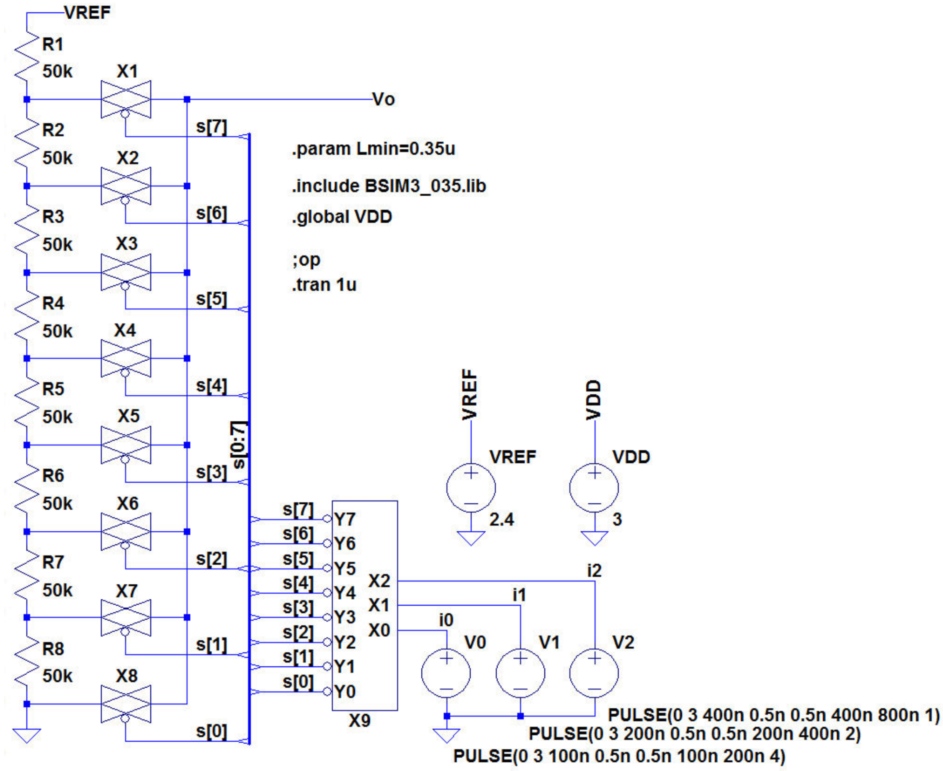


Figure 5.28: Complete D/A converter.

VDD', defining it as a common node to all subcircuits in the hierarchy, and also the transistor model specification is given at the top level. Observe that in this figure, the outputs from the decoder have been connected to the analog switch inputs using a bus wire. First, the bus wire is drawn as a normal wire, 'Edit → Draw Wire' (or 'F3'). Then bus taps are inserted using 'Edit → Place BUS tap', and the wires going to the bus are given the appropriate labels as shown in fig. 5.28 using 'Edit → Label Net' (or 'F4'). This labelling of the wires going to the bus is what ensures the correct netlist for the schematic. Finally, the bus wire can be labelled using 'Edit → Label Net'. The bus label must indicate the number of the first and last wire, separated by a colon (:) and using square brackets ([and]) which are also used for the individual wires. The specification of the bus wire turns the wire into a thick line to indicate that it is a bus.

As an example of a simulation of the D/A converter, fig. 5.29 shows the output from a transient simulation where the input signals are defined to switch through all 8 input combinations using the 'PULSE' specifications of the input signals shown in fig. 5.28.

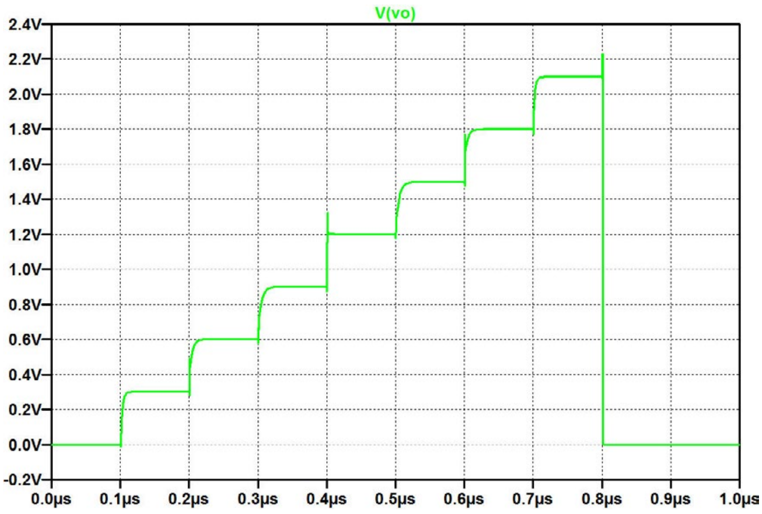


Figure 5.29: Simulation of the D/A converter from fig. 5.28.

Hints and pitfalls

- The easiest way to ensure correct correspondence between a subcircuit schematic and a subcircuit symbol is to let LTspice create the symbol from the schematic, see page 149.
- Define inputs and outputs in the subcircuit by using the ‘Port Type’ definition when labelling the inputs and output.
- A global node which should not be shown as a terminal in the subcircuit symbol must be declared as a global node by the command ‘.global <net label>’ or must be specified by a net label beginning with the characters ‘`$G_`’.
- After the symbol has been created by LTspice, you can modify the graphic appearance using the symbol editor.
- Save your subcircuits and symbols in an appropriate folder, the folder also used for your circuits. Do not use the same filename for a subcircuit and a circuit at a higher level in the circuit hierarchy.
- Parameters for subcircuits can be specified at subcircuit level, at top level of the circuit hierarchy, or by specifying the parameter when inserting and editing the subcircuit, see fig. 5.21 on page 167. A parameter defined in this way overrides a parameter specified at top level or subcircuit level. A parameter specified at subcircuit level overrides a parameter specified at top level.
- ‘.include’ statements (e.g. a library file) can be specified at subcircuit level or at top level. A specification at subcircuit level overrides a specification at top level.
- Internal node voltages and device currents in a subcircuit can be made visible by the command ‘Tools → Control Panel’ where you select the tab ‘Save Defaults’. Here you tick ‘Save Subcircuit Node Voltages’ and ‘Save Subcircuit Device Currents’.

References

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

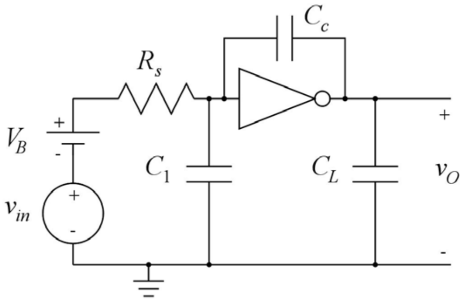
Carusone, TC., Johns, D. & Martin, K. 2012, *Analog Integrated Circuit Design*, Second Edition, John Wiley & Sons, Inc., Hoboken, USA.

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

The MOSIS Service: Wafer Electrical Test Data and SPICE Model Parameters, 2014. Retrieved from https://www.mosis.com/pages/Technical/Testdata/tsmc-018-prm

Problems

5.1



$R_s = 1\ \mathrm{M\Omega}$, $V_{DD} = 3$ V,

$C_1 = 0.2$ pF, $C_c = 0.7$ pF, $C_L = 1.5$ pF.

Figure P5.1

An inverter as shown in fig. 5.25 on page 170 may be used as an inverting amplifier. Design a test bench as shown in fig. P5.1 using a supply voltage of 3 V, a minimum length of `Lmin=0.35u`, a fanout of `Fanout=1` and the BSIM3 transistor model from fig. 3.10 on page 77. Find a bias input voltage V_B which gives a bias output voltage of 1.5 V. With this value of V_B, simulate the AC response and find the dominant pole. Also, use the Miller approximation (Carusone, Johns & Martin 2012) to calculate the dominant pole and compare to the simulated value.

5.2

$$A_v(s) = \frac{\omega_{ta}(1 - s/\omega_z)}{s(1 + s/\omega_{p2})}$$

$\omega_{ta} = 2\pi \cdot 120$ MHz,

$\omega_z = 2\pi \cdot 200$ MHz,

$\omega_{p2} = 2\pi \cdot 76$ MHz.

Figure P5.2

Simulate the AC response of the closed loop gain and the loop gain for the opamp shown in fig. 5.10 and 5.11 with $C_1 = 1$ pF, $C_2 = 0.2$ pF and $C_L = 1.5$ pF using the generic filter blocks from Table 5.3. Assume a transfer function for the opamp as specified in fig. P5.2.
Find the phase margin and the closed loop bandwidth and compare to the results found in Example 5.2

5.3

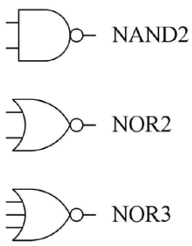


Figure P5.3

Design subcircuits for a two-input NAND gate and two-input and three-input NOR gates similar to the logic gate and inverter designs shown in Example 5.4. Scale the PMOS transistors relative to the NMOS transistors to compensate for the difference in electron mobility and hole mobility, assuming $\mu_n = 3 \times \mu_p$. Use the BSIM3 transistor models from fig. 3.10 on page 77 with a channel length of $L_{min} = 0.35$ µm and a minimum channel width of $3 \times L_{min}$ for NMOS transistors and $9 \times L_{min}$ for PMOS transistors.
What are the transistor channel widths used for the gates?

5.4

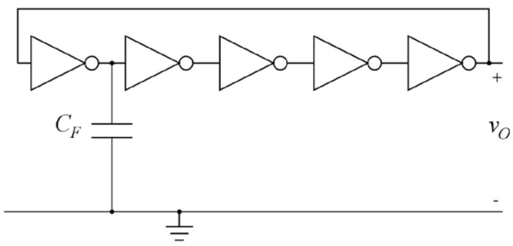


Figure P5.4

Use the inverter from fig. 5.25 on page 170 to design a ring oscillator as shown in fig. P5.4. Use the BSIM3 transistor models from fig. 3.10 on page 77 with a channel length of $L_{min} = 0.35$ µm and a supply voltage of 3 V. With `Fanout=1` and $C_F = 0.2$ pF, find the frequency of oscillation. Also find the inverter delay for an inverter loaded with an identical inverter. Repeat for `Fanout=5` for all of the inverters.

Hint: To start the oscillation, inject a short current pulse in the output node.

Answers

5.1: $V_B = 1.505$ V; low frequency gain: 24.9 dB; dominant pole, simulated: 12 kHz; dominant pole, calculated: 12 kHz.

5.2: Phase margin: 70°; Closed loop bandwidth: 30.7 MHz.

5.3: NOR2: $W_n = 3 \times L_{min}$, $W_p = 18 \times L_{min}$; NOR3: $W_n = 3 \times L_{min}$, $W_p = 27 \times L_{min}$. NAND2: $W_n = 6 \times L_{min}$, $W_p = 9 \times L_{min}$.

5.4: `Fanout=1`: $f_{osc} = 629$ MHz, $t_{delay} = 72$ ps; `Fanout=5`: $f_{osc} = 1076$ MHz, $t_{delay} = 64$ ps.

Tutorial 6 – Process and parameter variations

This tutorial illustrates how process variations and parameter variations can be handled with LTspice simulations. Normally, a design is required to work not only under some typical conditions but also over a range of parameter variations such as voltage and temperature variations. Also, tolerance in the manufacturing processes must be taken into account. This is a major challenge to the designer, involving extensive simulation work during the design process. After having completed the tutorial, you should be able to

– simulate a design in design corners with process, voltage and temperature (PVT) variations.
– design device models for slow, typical and fast process corners.
– evaluate target design parameters using the '.measure' SPICE Directive.
– perform a Monte Carlo simulation.

Often process and parameter variations are treated using a worst-case approach where the worst combinations of process parameters and operating conditions are simulated in addition to a simulation under typical conditions. For the process parameter variations, a standard method is to use device models for 'typical', 'fast' and 'slow' devices (Weste & Harris 2010). Thus, for each transistor type (NMOS and PMOS) we have three different models, a typical model, a fast model and a slow model. This leads to the process corners SS, FS, SF and FF where the first letter indicates the NMOS model and the second letter indicates the PMOS model (F for fast, S for slow). This is illustrated in fig. 6.1, showing the four process corners for a design target in addition to the typical set of parameters. Generally, it is assumed that for any combination of parameter variations, the design target will fall within the shaded area shown in the figure. Also for resistors and capacitors, typically fast and slow models or values must be considered.

The process parameter variations may be combined with simulations taking operating conditions such as supply voltage and temperature into account. This is often termed PVT variations (Carusone,

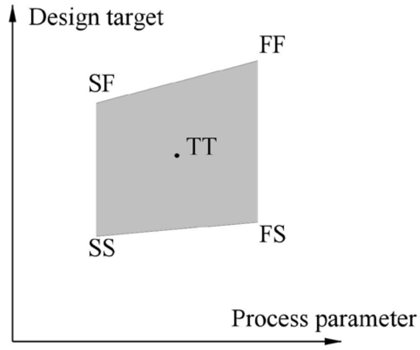


Figure 6.1: Design corners with slow and fast transistor models.

Johns & Martin 2012). This can lead to a large number of combinations and simulations, so - if possible - it is a good idea to identify the more critical combinations of process parameters and operating conditions in order to limit the number of simulations.

Using the design corners with respect to process variations and operating conditions leads to simulation of worst-case combinations so that a design can be made robust over a large span of variations. However, it does not take into account the probability that the circuit will actually fall in a worst case process corner.

Another approach is to apply statistical variations rather than worst case variations to some of the critical parameters of a design and use Monte Carlo simulation for determining the influence on the final design. This will be further investigated in Example 6.4.

Example 6.1: Model files for corner simulations.

An obvious way of handling model files for different process corners is to use different models for each simulation and run a separate simulation for each corner. Thus, for the BSIM transistor model used in Tutorials 3 - 5 you could define a file for each corner, i.e. '`BSIM3_035TT.lib`', '`BSIM3_035SS.lib`', '`BSIM3_035FF.lib`' etc. For the corners and the typical device parameters shown in fig. 6.1, this implies 5 different files, and you would just include the appropriate file for each simulation.

However, here we will show an approach where the corners are defined by a speed parameter '`S`' with a value of 0 for the typical condition, -1 for the slow condition and $+1$ for the fast condition. With this approach, we can easily show results for simulations with different values of '`S`' using the '.step' command and we can even interpolate the process parameters between the typical value and the corner values. For the transistors, we need a speed parameter '`SN`' for the NMOS transistor and another parameter '`SP`' for the PMOS transistor. Obviously, we also need the values for the different model parameters for typical, slow and fast transistors.

A general method: Assume that a certain parameter x is defined by a typical value x_t, a slow value x_s and a fast value x_f. We can define:

$$\Delta x_s = x_s - x_t \tag{6.1}$$

$$\Delta x_f = x_f - x_t \tag{6.2}$$

Using a speed parameter S (with $S = 0$ for the typical parameter, $S = -1$ for the slow parameter and $S = 1$ for the fast parameter), we notice that the function $(|S|+S)/2$ is equal to S for S positive and 0 for S negative. Similarly, the function $(|S|-S)/2$ is equal to $|S| = -S$ for S negative and 0 for S positive. Using S, we then have a general expression for x:

$$\begin{aligned} x &= x_t + \Delta x_f \times (|S|+S)/2 + \Delta x_s \times (|S|-S)/2 \\ &= x_t \times (1-|S|) + x_s \times (|S|-S)/2 + x_f \times (|S|+S)/2 \end{aligned} \tag{6.3}$$

With this expression, we achieve $x = x_t$ for $S = 0$, $x = x_s$ for $S = -1$ and $x = x_f$ for $S = 1$ and we have a linear interpolation between typical, fast and slow values for (non-integer) values of $|S| < 1$. In LTspice, $|x|$ is given by '`abs(x)`' and also the function '`uramp(x)`' is available. This function is defined as ‘$\texttt{uramp(x)} = (|\texttt{x}| + \texttt{x})/2$’ and using this, (6.3) would appear as

$$\texttt{X=\{Xt*(1-abs(S))+Xs*uramp(-S)+Xf*uramp(S)\}} \tag{6.4}$$

with the values for `Xt`, `Xs` and `Xf` inserted in the expression.

Specific examples: The approach described above is generally applicable when you have the model parameters given by specific values for typical, fast and slow conditions. Occasionally, the model parameters may be specified by a relative variation or an absolute variation instead. For example, a capacitor may be specified by a nominal value C_{nom} and a tolerance δ in percent, or a threshold voltage may be specified by a nominal value $V_{th,\text{nom}}$ and a tolerance ΔV_{th} in absolute value (volts). In such cases the introduction of the speed parameter S is even simpler:

$$C = C_{\text{nom}}(1 - S \times \delta/100) \tag{6.5}$$

assuming that a small value of C is the fast model, and

$$V_{th} = V_{th,\text{nom}} - S \times \Delta V_{th} \tag{6.6}$$

for an NMOS transistor which is faster for smaller values of V_{th}.

An examination of the BSIM3 models for the 0.35 μm process from (Carusone, Johns & Martin 2014) shows that only 5 parameters are different for typical, fast and slow transistor models. These are:

The threshold voltage: The threshold voltage parameter `VTH0` is given by a nominal value and a threshold voltage shift of ± 0.1 V, increasing the numeric value of the threshold voltage for slow transistors and decreasing it for fast transistors.

The oxide thickness: The parameter for gate oxide thickness `TOX` is given by a nominal value which is divided by 0.95 for the slow models and by 1.05 for the fast models.

The mobility: The mobility `U0` is given by a nominal value which is multiplied by $(0.95)^2$ for the slow models and by $(1.05)^2$ for the fast models.

The bulk junction bottom capacitance: The zero-bias bulk junction bottom capacitance per unit area `CJ` is given by a nominal value which is divided by 0.95 for the slow models and by 1.05 for the fast models.

The bulk junction sidewall capacitance: The zero-bias bulk junction bottom capacitance per unit perimeter `CJSW` is given by a nominal value which is divided by 0.95 for the slow models and by 1.05 for the fast models.

Using the speed parameters SN for NMOS transistors and SP for PMOS transistors, these relations can be modeled as follows (with the nominal values inserted):

NMOS transistors:

$$\mathtt{VTH0} = \{0.48 - \mathtt{SN}/10\} \tag{6.7}$$

$$\mathtt{TOX} = \{7.8E - 9/(1 + \mathtt{SN}/20)\} \tag{6.8}$$

$$\mathtt{U0} = \{360 * (1 + \mathtt{SN}/20) * *2\} \tag{6.9}$$

$$\mathtt{CJ} = \{9e - 4/(1 + \mathtt{SN}/20)\} \tag{6.10}$$

$$\mathtt{CJSW} = \{2.8e - 10/(1 + \mathtt{SN}/20)\} \tag{6.11}$$

PMOS transistors:

$$\mathtt{VTH0} = \{-0.6 + \mathtt{SP}/10\} \tag{6.12}$$

$$\mathtt{TOX} = \{7.8E - 9/(1 + \mathtt{SP}/20)\} \tag{6.13}$$

$$\mathtt{U0} = \{150 * (1 + \mathtt{SP}/20) * *2\} \tag{6.14}$$

$$\mathtt{CJ} = \{14e - 4/(1 + \mathtt{SP}/20)\} \tag{6.15}$$

$$\mathtt{CJSW} = \{3.2e - 10/(1 + \mathtt{SP}/20)\} \tag{6.16}$$

Inserting the expressions (6.7 - 6.16) in the models from fig. 3.10 on page 77, we have the library file BSIM3_035PVT.lib shown in fig. 6.2. This file can be directly copied into a text editor and saved as BSIM3_035PVT.lib for use with LTspice.

Combining the speed parameters into a single parameter: Sometimes it may be useful to combine a set of speed parameters and/or temperature parameters into a single parameter by which you can step through a desired sequence of combinations of the different parameters, rather than stepping each parameter individually. It reduces the number of '.step' commands (LTspice can handle a maximum of three levels of nested '.step' commands), and it reduces the total number of simulations to include only the PVT corners of interest. This can be done by using the '.step' command in combination with the command '.param <param_name>=table(N,a,b,c,d,...)'. Assume for instance that we want to step through the process corners shown in fig. 6.1 in the sequence TT, SS, FS, SF, FF. This corresponds to the sequence (SN,SP)=(0,0), (−1,−1), (1,−1), (−1,1), (1,1). Using a parameter 'N' as step number to count the steps from 1 to 5, this is accomplished by the following SPICE Directives:

```
.step param N 1 5 1
```

Generic BSIM3 model for 0.35 μm CMOS process with speed parameters SN and SP to define process variations.

```
*BSIM3_035PVT.lib
.MODEL NMOS-BSIM NMOS LEVEL = 49
*Speed parameter SN
+VERSION = 3.1 TNOM = 27 TOX = {7.8E-9/(1+SN/20)}
+XJ = 1E-07 NCH = 2.18E+17 VTH0 = {0.48-SN/10}
+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*(1+SN/20)**2} 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/(1+SN/20)} PB = 7.95E-01 MJ = 3.53E-01
+CJSW = {2.8e-10/(1+SN/20)} 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
*Speed parameter SP
+VERSION = 3.1 TNOM = 2.69E+01 TOX = {7.8E-9/(1+SP/20)}
+XJ = 1.00E-07 NCH = 8.44E+16 VTH0 = {-0.6+SP/10}
+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*(1+SP/20)**2} 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/(1+SP/20)} PB = 9.83E-01 MJ = 5.79E-01
+CJSW = {3.2e-10/(1+SP/20)} 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 6.2: Library file with BSIM3 models for a generic 0.35 μm CMOS process with speed parameters SN and SP to define process variations, adapted from (Carusone, Johns & Martin 2014). Speed parameter is 0 for typical model, –1 for slow model and +1 for fast model.

```
.param SN=table(N,1,0,2,-1,3,1,4,-1,5,1)
.param SP=table(N,1,0,2,-1,3,-1,4,1,5,1)
```

In the '.param' definitions, the first table entry is the step number N, and this is followed by pairs of N-values and SN/SP-parameter values.

Process and temperature variations for an NMOS transistor: As a very simple example of the use of the speed parameter we will examine the characteristics of an NMOS transistor. Fig. 6.3 shows the LTspice schematic for this, compare fig. 3.12 on page 80. Using the '.step' command, the parameter 'SN' is stepped through the values -1, 0 and $+1$, corresponding to the slow, typical and fast NMOS transistor model, respectively.

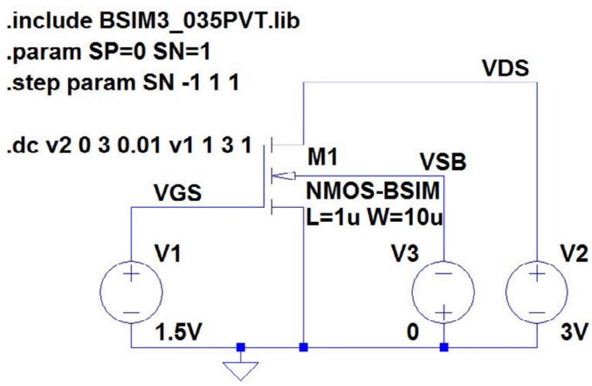
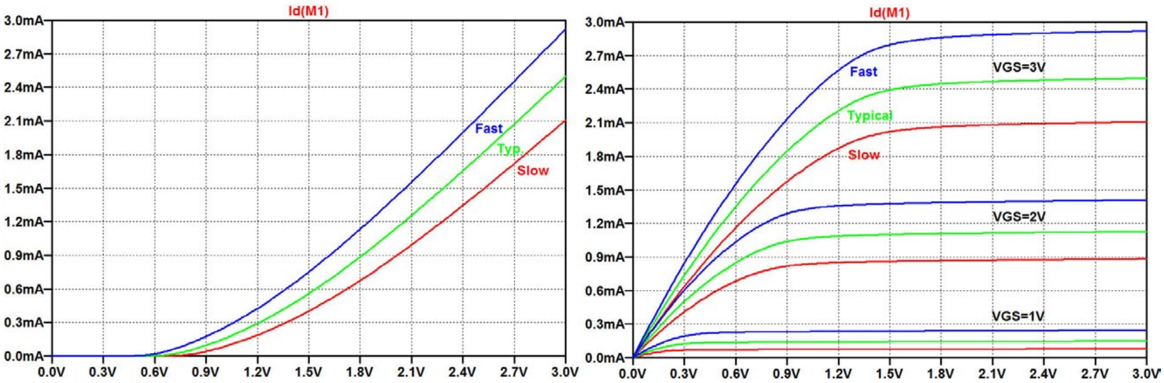


Figure 6.3: LTspice schematic for simulation of transistor characteristics with slow, fast and typical BSIM3 models.

Fig. 6.4 shows the simulated input characteristics and output characteristics. The input characteristics are simulated with $V_{DS} = 3.0$ V so that the transistor is in the active region. The output characteristics are simulated for $V_{GS} = 1$, 2 and 3 V. It is evident that there is a significant difference in drain current between the fast and slow model, about 50%. For the transistor, often a design target is a specific value of the transconductance g_m, see for instance the design procedure for transistors M_1 and M_7 in example 5.2 on page 154 - 165. The transconductance may be simulated as shown in fig. 5.12 on page 160. In order to illustrate the variation in g_m caused by process variations, we can simulate the transistor with $I_D = 100$ µA for three different values of the speed parameter 'SN', corresponding to slow, typical and fast parameters. The results of this simulation are shown in fig. 6.5



(a) (b)

Figure 6.4: Input characteristics (a) and output characteristics (b) for slow (red traces), fast (blue traces) and typical (green traces) BSIM3 models.

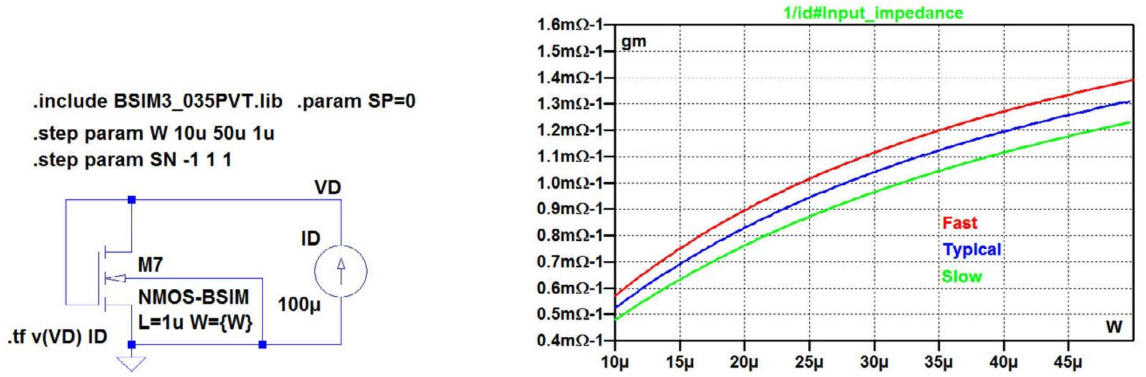


Figure 6.5: Simulation showing g_m versus W for an NMOS transistor with $L = 1$ µm and $I_D = 100$ µA using the BSIM3 transistor model with process variations (slow, typical and fast models).

which may be compared with fig. 5.12 on page 160. We notice that the process variations cause a variation in g_m of about $\pm 8\%$.

Another cause of variation in g_m is the temperature. The previous simulations are performed at a default temperature of 27°C. By using the SPICE Directive '.temp', different values for the temperature can be specified. The command '`.temp -40 27 85`' is equivalent to '`.step temp list -40 27 85`'. The parameter 'temp' is a parameter predefined in LTspice for temperature. Although it is possible, it is not advisable to use the name 'temp' for another parameter. You may observe that there is a difference between '`.step temp list -40 27 85`' and '`.step param temp list -40 27 85`'. The first command steps the temperature in three steps, -40°C, 27°C and 85°C. The second defines a new parameter, 'temp', which is stepped between the values -40, 27 and 85, not to be confused with the temperature.

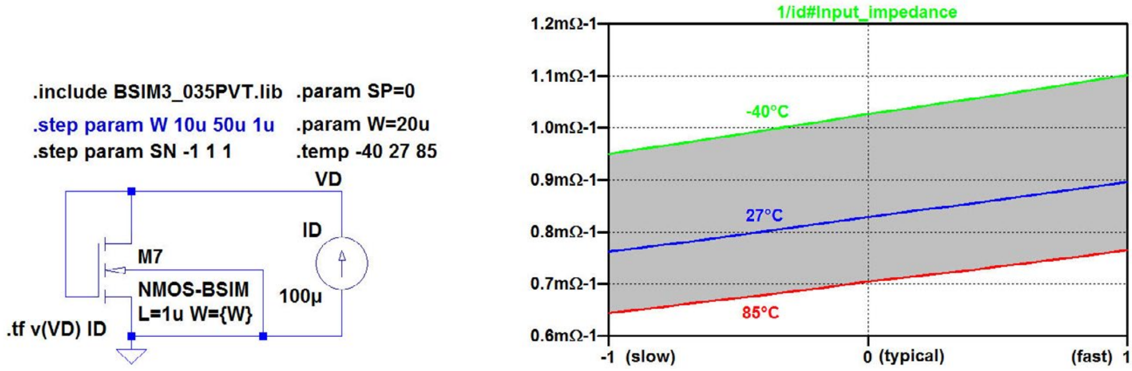


Figure 6.6: Simulation showing g_m versus process variations and temperature W for an NMOS transistor with $L = 1$ µm, $W = 20$ µm and $I_D = 100$ µA using the BSIM3 transistor model.

Fig. 6.6 shows a simulation of g_m with $W = 20$ µm, $I_D = 100$ µA and at -40°C, 27°C and 85°C, corresponding to an industrial temperature range. For this simulation, also 'SN' is stepped from -1 to $+1$, so the simulation shows the design corners for process and temperature variations, compare fig. 6.1. The shaded area in fig. 6.6 shows the range of g_m values for both process and temperature variations. We see that g_m decreases significantly with temperature. A small value of g_m is critical, so the worst case corner is the high temperature corner with a slow process. In this corner, g_m is only 0.65 mA/V whereas the design target in Example 5.2 was 0.80 mA/V. Clearly, a design iteration where the transistor channel width is increased to give $g_m = 0.80$ mA in the worst case corner would be an obvious improvement to the design. A simulation like the simulation shown in fig. 6.5 but at 85°C shows that W has to be increased to 30 µm in order to ensure $g_m \geq 0.80$ mA/V.

Example 6.2: An inverter.

An inverter as shown in fig. 5.25 on page 170 can be used both as a digital inverter and as an inverting amplifier (see problem 5.1 on page 175). For the inverter used as an amplifier, we would expect PVT variations to cause a variation in several design parameters, including low frequency gain, bandwidth, unity gain bandwidth, supply current, etc. For this example, we assume that the amplifier is capacitively loaded and is driven from a voltage source providing an AC signal and a DC bias voltage V_B, see fig. 6.7. With this configuration, the unity gain bandwidth is $GBW = (g_{m1} + g_{m2})/(2\pi \times C_L)$, the low frequency gain is $A_0 = (g_{m1} + g_{m2})/(g_{ds1} + g_{ds2})$ and the bandwidth is $BW = (g_{ds1} + g_{ds2})/(2\pi \times C_L)$. These parameters are all small signal parameters, and they depend on the bias point which in turn depends on the process parameters, supply voltage and temperature.

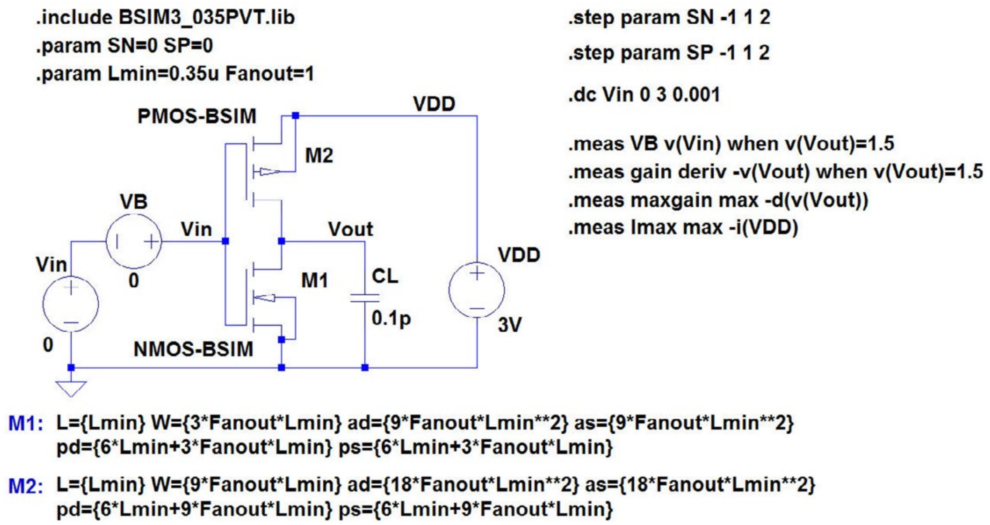


Figure 6.7: LTspice schematic of the inverting amplifier showing a selection of '.measure' commands.

DC Sweep: Also large signal properties such as DC transfer characteristics and peak supply current exhibit PVT variations. As the first investigation, we show the DC transfer characteristics for the process corners in fig. 6.8. We observe that a fast NMOS transistor pulls the transfer characteristics to the left whereas a fast PMOS transistor pulls the characteristics to the right. Obviously, the bias value V_B of the input voltage for an output quiescent value of about 1.5 V varies between 1.3 V and 1.7 V.

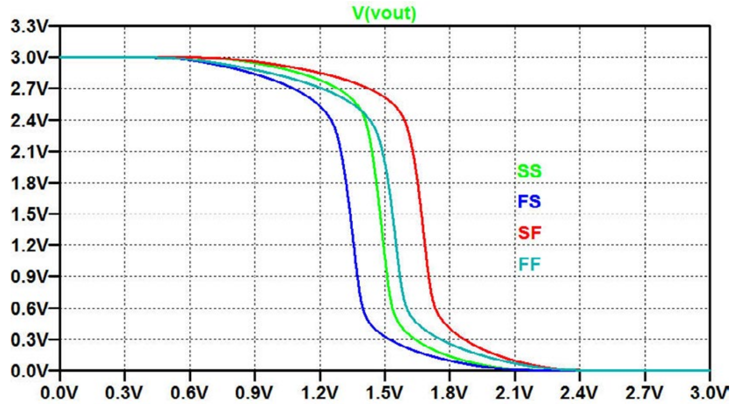


Figure 6.8: DC sweep, output voltage versus input voltage for the inverter from fig. 6.7.

From the DC sweep we can also find the low frequency gain and the supply current by plotting '`d(V(vout))`' and '`-I(Vdd)`', respectively, as shown in fig. 6.9.

By using the '.measure' (or '.meas') command we can analyze gain and supply current in the process corners. In fig. 6.7 four different '.measure' commands are shown.

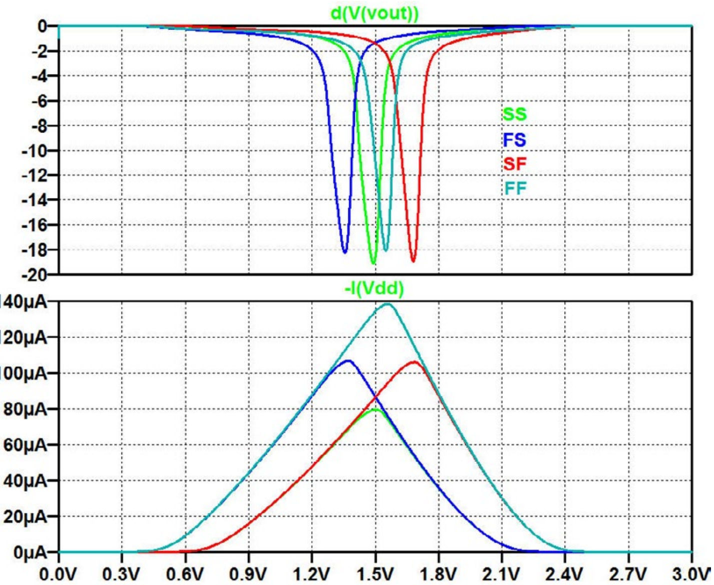


Figure 6.9: DC sweep, low frequency gain versus input voltage (top) and peak current consumption versus input voltage (bottom) for the inverter from fig. 6.7.

```
SPICE Error Log

Circuit:  * M:\LTspice\Tutorial06\Fig6_07.asc

.step sn=-1 sp=-1
.step sn=1 sp=-1
.step sn=-1 sp=1
.step sn=1 sp=1

Measurement:  vb
  step    v(vout)=1.5
     1    1.47817
     2    1.33947
     3    1.66893
     4    1.53505

Measurement:  gain
  step    D(-v(vout))        at
     1    18.3992            1.47817
     2    17.0934            1.33947
     3    18.1569            1.66893
     4    16.9492            1.53505

Measurement:  maxgain
  step    MAX(-d(v(vout)))    FROM      TO
     1    19.1021             0         3
     2    18.2456             0         3
     3    18.918              0         3
     4    18.0925             0         3

Measurement:  imax
  step    MAX(-i(vdd))        FROM      TO
     1    7.95096e-005        0         3
     2    0.000106626         0         3
     3    0.000105899         0         3
     4    0.000138436         0         3
```

Figure 6.10: Error log file with results of '.measure' commands.

– The first, '`.meas VB v(Vin) when v(Vout)=1.5`', finds the required value '`VB`' of the DC bias voltage in each process corner.

– The second, '`.meas gain deriv -v(Vout) when v(Vout)=1.5`', calculates the numeric value of the slope of the transfer characteristics, '`gain`', at an output voltage of 1.5 V.

– The third, '`.meas maxgain max -d(v(Vout))`', calculates the maximum numeric value '`maxgain`' of the gain.

– The fourth, '`.meas Imax max -i(VDD)`', finds the peak supply voltage '`Imax`'.

Each of the '.measure' commands are computed in the four process corners in the sequence SS, FS, SF, FF and the results are given in the error log file ('Ctrl-L'). Fig. 6.10 shows the results from the error log file. 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. Here, the traces to be displayed are selected using 'Plot Settings → Add trace' or simply using 'Pick Visible Traces', . Fig. 6.11 shows the plots of '`VB`', '`gain`', '`maxgain`' and '`Imax`'. In each of the plots, the X-axis is '`SN`' and the green line corresponds to '`SP`' = -1 (slow PMOS) while the blue line is for '`SP`' = 1 (fast PMOS).

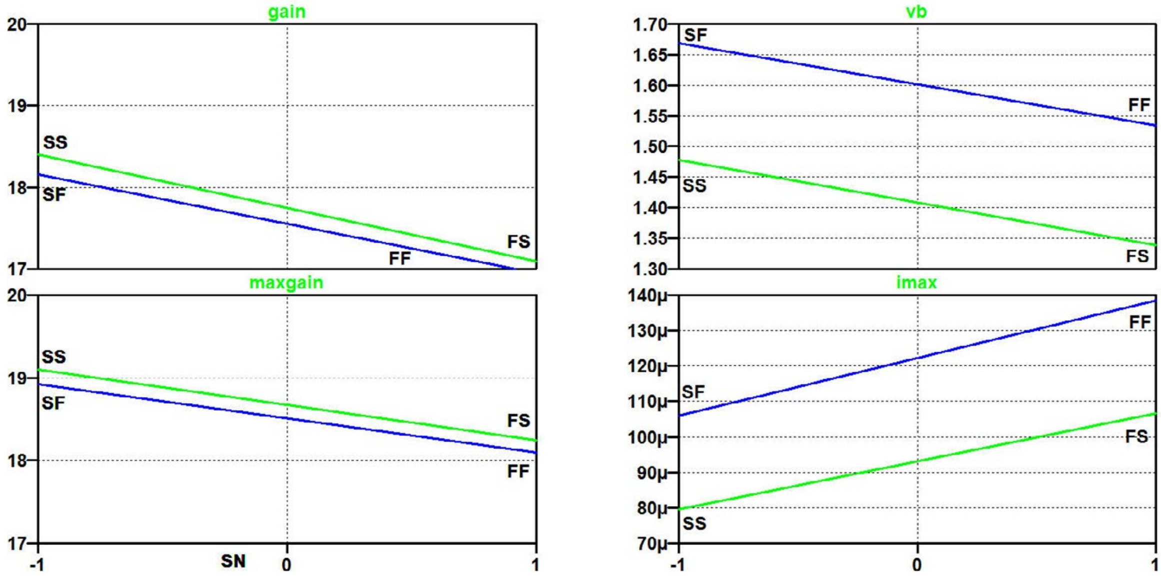


Figure 6.11: Gain, maximum gain, input bias voltage and peak supply current versus process variations.

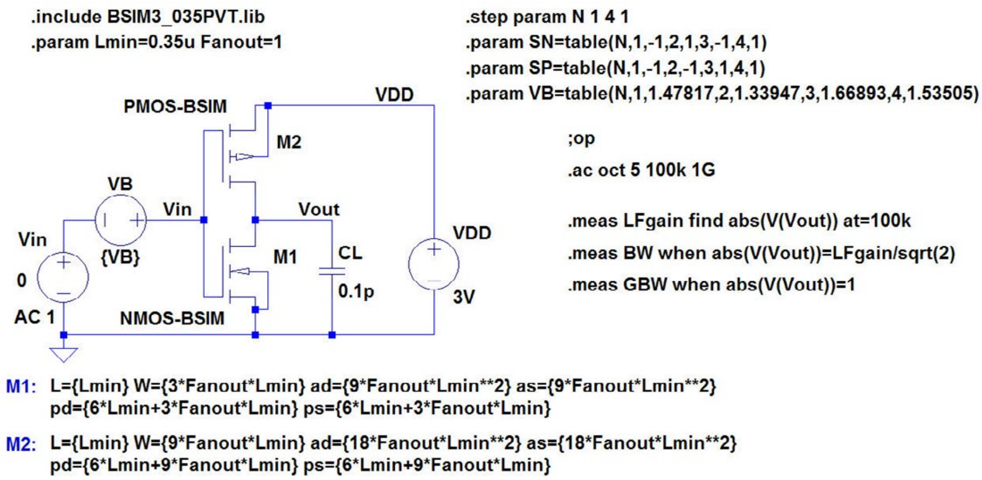


Figure 6.12: LTspice schematic of the inverting amplifier with '.step' and '.param' commands for stepping through the four process corners SS, FS, SF and FF.

AC Analysis: For simulating bandwidth and unity gain bandwidth, a '.ac' simulation is obvious. However, in order to get correct results from a '.ac' simulation, it is necessary to use the correct bias point. For the circuit in fig. 6.7, a suitable bias point would imply a DC value of V_B resulting in an output voltage of about 1.5 V when the input signal is $v_{IN} = 0$ V. From the '.dc' simulation, we found the values of V_B required for an output voltage of 1.5 V and they are listed in the error log file, fig. 6.10. Thus, a simple way of specifying V_B is to use a parameter 'N' to count through the four steps from the previous simulations as explained on page 183.

Fig. 6.12 shows the circuit from fig. 6.7 with the required parameter definitions and also with '.measure' commands for finding the low frequency gain 'LFgain', the 3-dB bandwidth 'BW' and the unity gain bandwidth 'GBW'. The bias values are easily checked by a '.op' simulation.

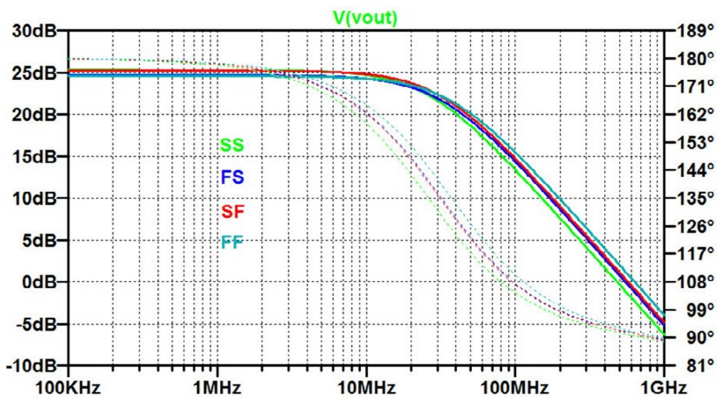


Figure 6.13: AC sweep, output voltage versus frequency for the inverter from fig. 6.12.

```
SPICE Error Log
Circuit:  * M:\LTspice\Tutorial06\Fig6_12.asc

.step n=1
.step n=2
.step n=3
.step n=4

Measurement:  gbw
  step     abs(v(vout))=1
     1     4.83727e+008
     2     5.47652e+008
     3     5.72209e+008
     4     6.33343e+008
```

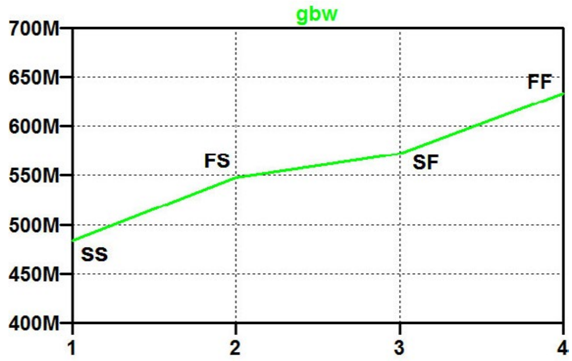


Figure 6.14: Error log file and plot of 'GBW' for the inverter from fig. 6.12.

The result of the '.ac' simulation is shown as a Bode plot of '`V(Vout)`' in fig. 6.13, and fig. 6.14 shows the *GBW* values from the error log file and the graphical plot of *GBW* (using a right mouse click in the error log file). With the use of the step number '`N`' as a parameter rather than separate step commands for '`SN`' and '`SP`', we do not obtain a plot directly showing the process corners as corners like the plots in fig. 6.11.

Often, a more flexible way of establishing the correct bias conditions for the '.ac' simulation is to provide a DC feedback from the output to the input. This will automatically adapt the input bias voltage when process parameters, voltage or temperature are changed. Fig. 6.15 shows two different ways of establishing a DC feedback. The basic concept is to provide a low pass feedback path from the output to the input. In fig. 6.15(a), this is achieved by an inductive feedback directly to the gates of M_1 and M_2 and a (highpass) AC coupling of the input voltage. This circuit ensures that both M_1 and M_2 are in the active region as $V_{GS} = V_{DS}$ but it does not ensure a bias value of the output voltage which is equal to $V_{DD}/2$.

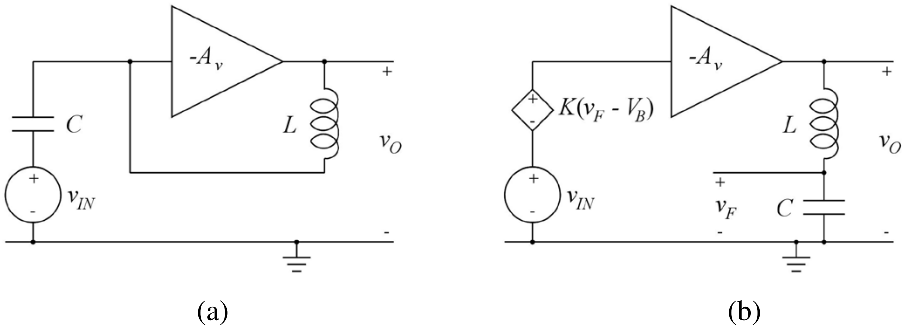


Figure 6.15: Examples of DC feedback to provide a suitable DC bias point for the inverter from fig. 6.12.

In the circuit in fig. 6.15(b), the fixed DC bias voltage shown in fig. 6.12 is replaced by a voltage controlled by the output voltage at very low frequencies so that the bias value of the output voltage is almost equal to V_B if the gain K is $\gg 1$. As an example, fig. 6.16 shows the LTspice schematic corresponding to fig. 6.15(b). Here the step number parameter '`N`' is replaced by separate step commands for '`SN`' and '`SP`', and also a temperature step is inserted using a step number '`T`' and a '`.temp = table(..)`' command with the three temperatures 27°C, −40°C and 85°C. The purpose of the '`T`' step number is to arrange the temperature steps with 27°C as the first step rather than −40 °C. Using just '`.step temp 27 -40 85`' results in steps with the lowest temperature first, even with 27°C listed as the first temperature.

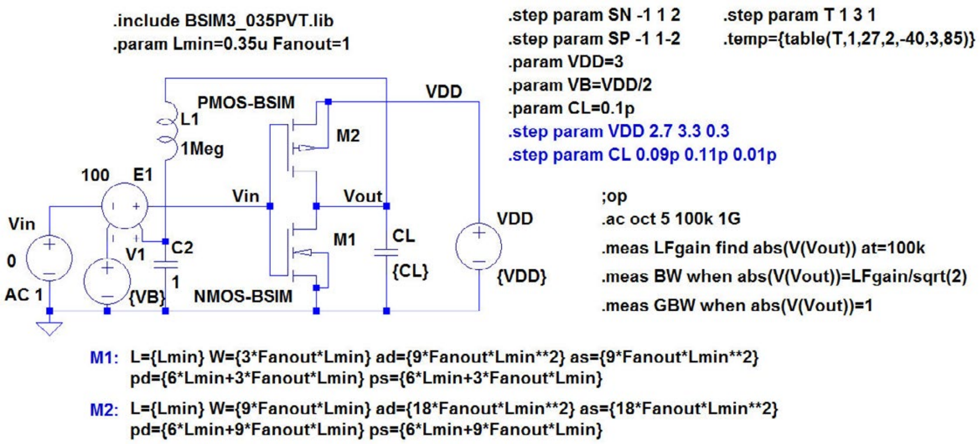


Figure 6.16: LTspice schematic of the inverting amplifier with low pass feedback to establish a suitable DC bias point.

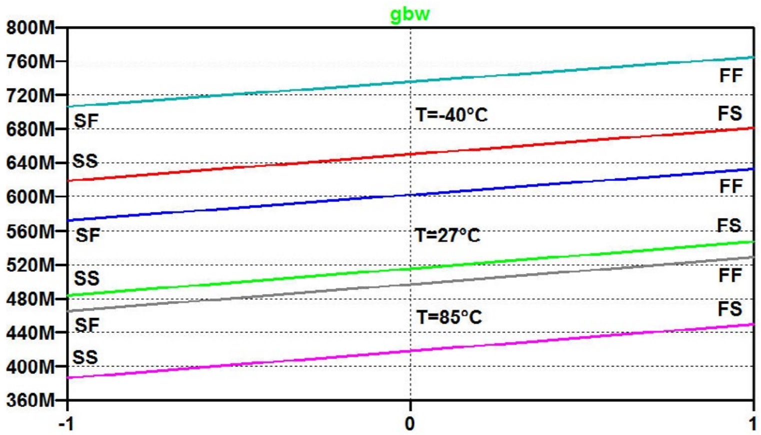


Figure 6.17: Plot of 'GBW' for the inverter from fig. 6.12 for different temperatures.

Fig. 6.17 shows the resulting plot of *GBW* for the three temperatures. For the temperature 27°C (green and blue trace), this plot may be compared to the plot of *GBW* shown in fig. 6.14.

In fig. 6.16, there is also a '.step' command for the supply voltage V_{DD} and for the capacitor C_L but they appear only as a comments.

LTspice does not support more than three levels of '.step' commands in one simulation, so trying to use the '.step' command for the supply voltage or capacitor in combination with the other three step commands in the schematic just results in an error message.

The digital inverter: The inverter shown in fig. 6.7 is basically a digital inverter, so it is also of interest to examine variations in its digital properties such as propagation delay, output rise time and fall time. Fig. 6.18 shows the digital inverter `X2` driven by an identical inverter and driving another inverter specified to have a fanout of 3 (see page 170). This implies that the inverter `X2` has a capacitive load corresponding to three standard inverters (or 2 NAND gate inputs (3-input NAND gates)).

Also shown in fig. 6.18 is the result of a transient simulation with typical parameters. From the waveforms, we can define a rise time and a fall time for '`V(vout)`' (the green trace) from 0.3 V to 2.7 V, and we can define a delay time from the rising edge of '`V(vin)`' to the falling edge of '`V(vout)`' and another delay time from the falling edge of '`V(vin)`' to the rising edge of '`V(vout)`'. The '.measure' commands for finding these are shown in the schematic.

When running a simulation with the speed parameters '`SN`' and '`SP`' varied through the process corners, we can find the process corners for the delay times and the rise and fall times. The result of this simulation is shown in fig. 6.19.

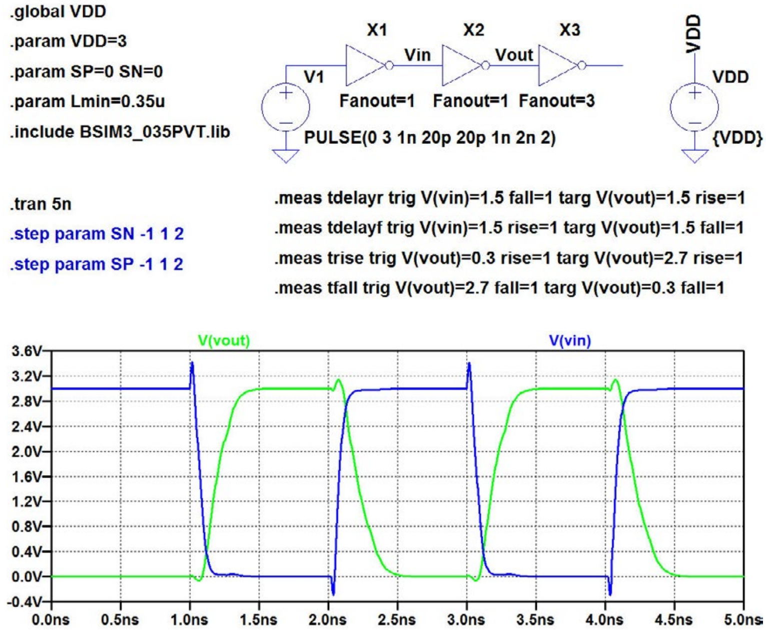


Figure 6.18: LTspice schematic of the inverter used as a digital inverter (top) and the result of a transient simulation (bottom).

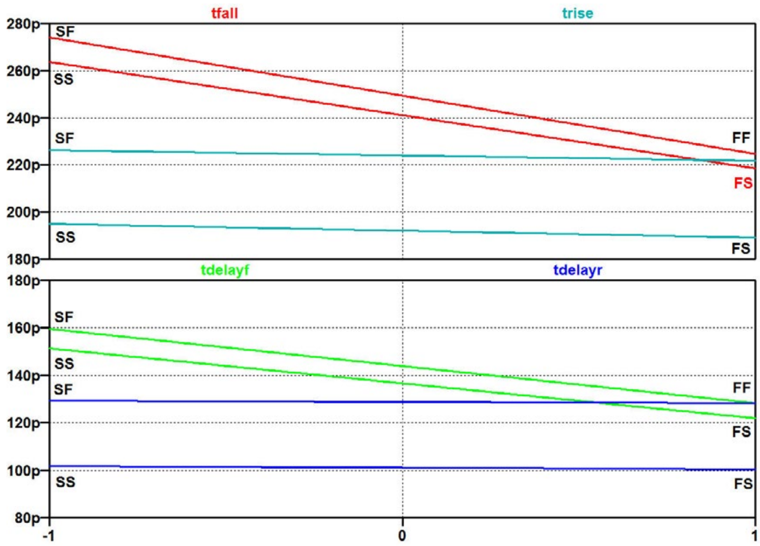


Figure 6.19: Process corner simulation for rise time, fall time and delays for the digital inverter from fig. 6.18 (top).

Example 6.3: A test bench for the two stage opamp.

When simulating PVT variations in the two stage opamp from Example 5.2 on page 154, it is necessary to define test benches which provide suitable bias conditions for the opamp, regardless of the PVT variations. This implies that we cannot rely on a fixed value of the offset voltage as in

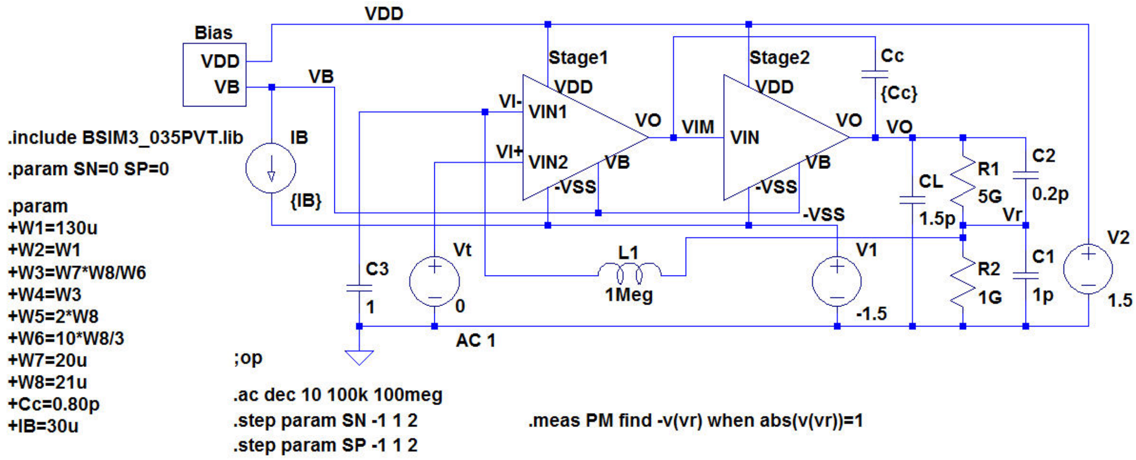


Figure 6.20: Test bench for simulation of loop gain of the two stage opamp with PVT variations.

figs. 5.15 and 5.16 (page 162). Rather, we must ensure a DC feedback path to the inverting opamp input similar to the DC feedback path used for the inverting amplifier in fig. 6.15.

Figs. 6.20 and 6.21 show the test benches from figs. 5.15 and 5.16 modified to include the required DC feedback. Also shown are '.measure' commands to find the phase margin and the bandwidth, respectively.

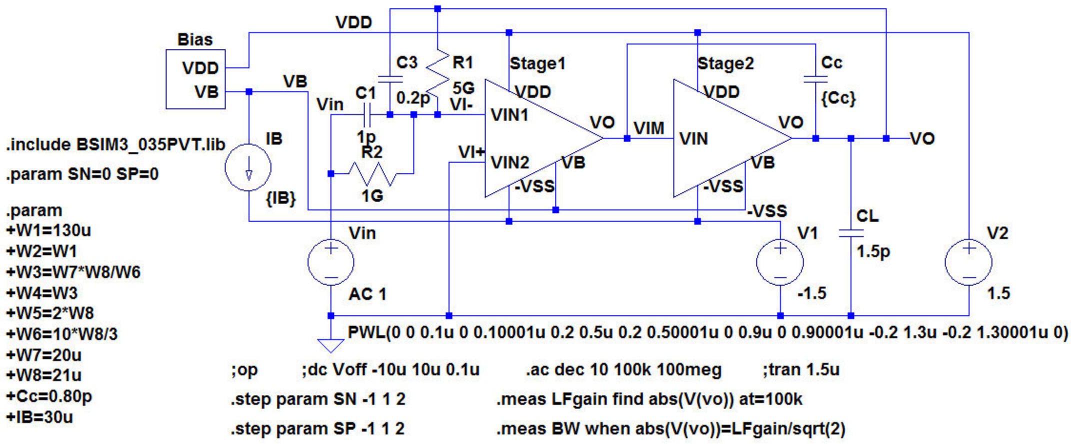


Figure 6.21: Test bench for simulations of closed loop response of the two stage opamp with PVT variations.

Fig. 6.22 shows the results of simulations of phase margin and bandwidth with process variations taken into account. Obviously, the amplifier does not fulfil the design requirements from page 155 in all process corners. The optimization of the two stage opamp for PVT variations is left as an exercise for the reader.

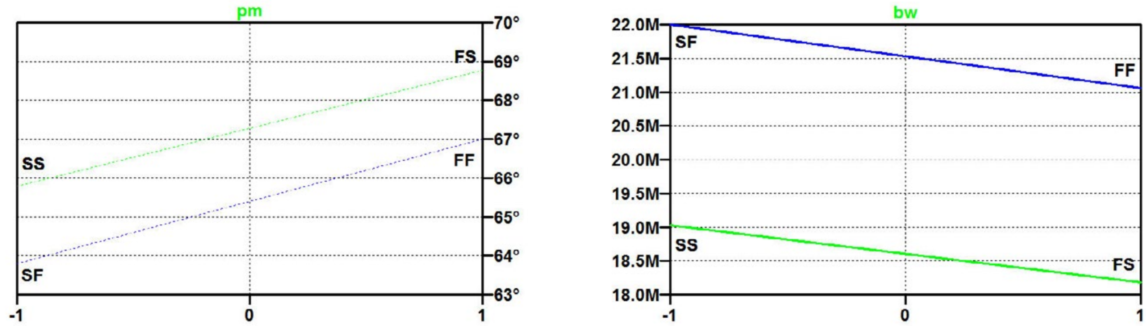


Figure 6.22: Phase margin and bandwidth versus process variations for the two stage opamp.

Example 6.4: Monte Carlo simulation.

The simulations shown in the previous examples are worst case simulations considering process, voltage and temperature limits. However, in practice the process variations are not described by fixed limits but rather by statistical variations of the parameter values. Often, a normal distribution around a nominal value is assumed. For investigating such statistical variations, Monte Carlo simulations may be used. In a Monte Carlo simulation, one or more device parameters are varied in a random fashion, and a number of simulations are performed with randomly selected device parameters. As an example, we will investigate the influence of variations in the transistor threshold voltage in a

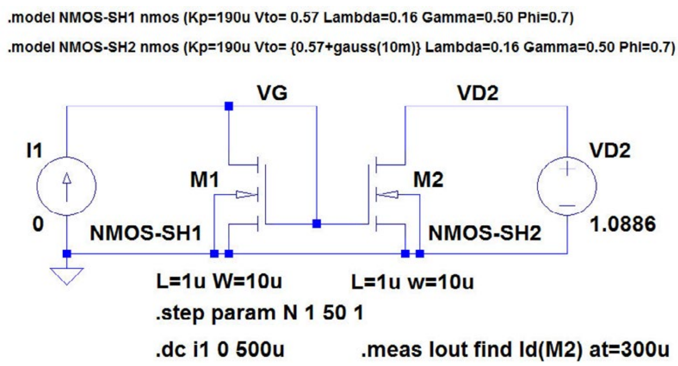


Figure 6.23: Current mirror for Monte Carlo simulation with threshold voltage mismatch.

simple current mirror. For simplicity, we assume a Shichman-Hodges transistor model. Fig. 6.23 shows the current mirror with the Shichman-Hodges models for transistor M_1 and M_2. For transistor M_1, the model from fig. 3.3 on page 71 is used, but for M_2, a statistical variation is added to the threshold voltage so that the two transistors no longer match perfectly.

For M_2, the threshold voltage is specified as the fixed value plus a statistical variation. LTspice has different possibilities for specifying a statistical variation:

– ‘`gauss(x)`’ generates a random number from a normal (Gaussian) distribution with a standard deviation of `x`.

– ‘`flat(x)`’ generates a random number between `-x` and `x` with a uniform distribution.

– ‘`mc(x,y)`’ generates a random number between `x*(1-y)` and `x*(1+y)` with a uniform distribution.

For the purpose of this simulation, we assume that the threshold voltage has the nominal value of 0.57 V as in fig. 3.3, but in addition to this, the threshold voltage of M_2 is given a statistical variation following a normal distribution with a standard deviation of 10 mV.

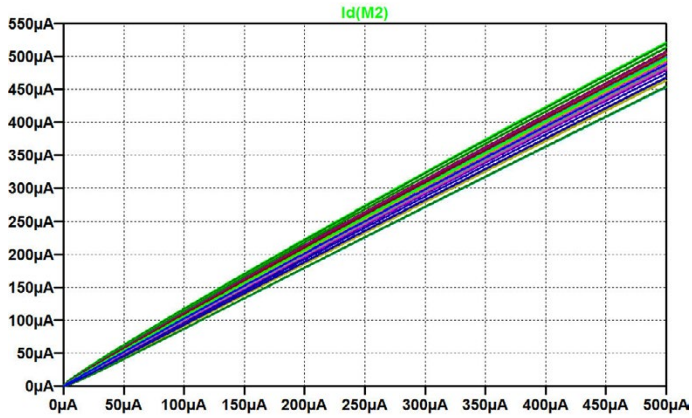


Figure 6.24: Output current versus input current with variations for the threshold voltage mismatch from fig. 6.23.

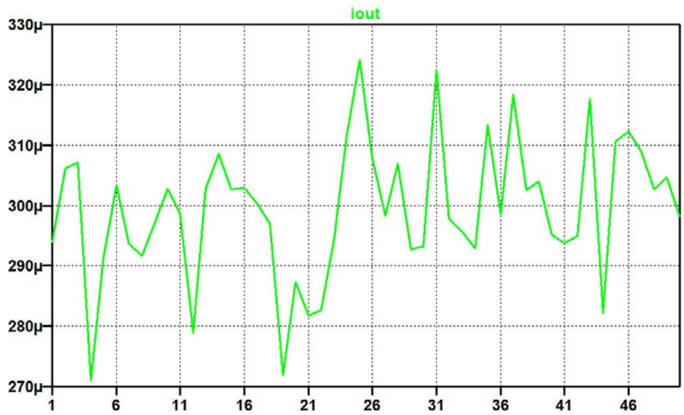


Figure 6.25: Output current variation for the 50 simulations with random values for the threshold voltage of M_2.

In fig. 6.23, also a parameter ‘`N`’ has been specified. This is the step count for the Monte Carlo simulation. With ‘`N`’ counting from 1 to 50, a total of 50 simulations are run where the value of the threshold voltage for M_2 is varied randomly between the simulations. The value of the drain voltage for M_2 has been selected to be equal to the gate voltage for a drain current of 300 µA, so for this value of the drain current, the current mirror provides a perfect match when the threshold voltages are identical. Also, a '.measure' command has been specified for finding the drain current of M_2 when the input current is 300 μA.

Fig. 6.24 shows the result of the '.dc' simulation. The spread of the output current is evident. In order to investigate the spread in more detail, the '.measure' command provides a table in the error log file with the output currents for each simulation and as in the previous examples, this can be displayed graphically using a right mouse click in the error log file. The resulting graph is shown in fig. 6.25. Obviously, the output current is about 300 μA but with variations of up to about ±25 μA. For further investigation of this result, it is a good idea to copy and paste the table with the output current into an Excel spreadsheet. Doing so, you find an average output current of 300 μA with a standard deviation of about 11 μA.

Statistical variation of the speed parameters: With the introduction of the speed parameters 'SN' and 'SP' providing interpolation of process parameters from slow to fast processes, it is easy to perform Monte Carlo simulations where 'SN' and 'SP' are varied randomly. We conclude this tutorial by revisiting the inverting amplifier from fig. 6.16 on page 194. This amplifier is shown again in fig. 6.26 with statistical specifications for the speed parameters. Two sets of specifications are given, one with a Gaussian distribution of 'SN' and 'SP' with a standard deviation of 0.4 and another (shown as a comment) with a flat distribution of 'SN' and 'SP' between −1 and 1.The number of simulations is specified by the parameter 'N' to be 50.

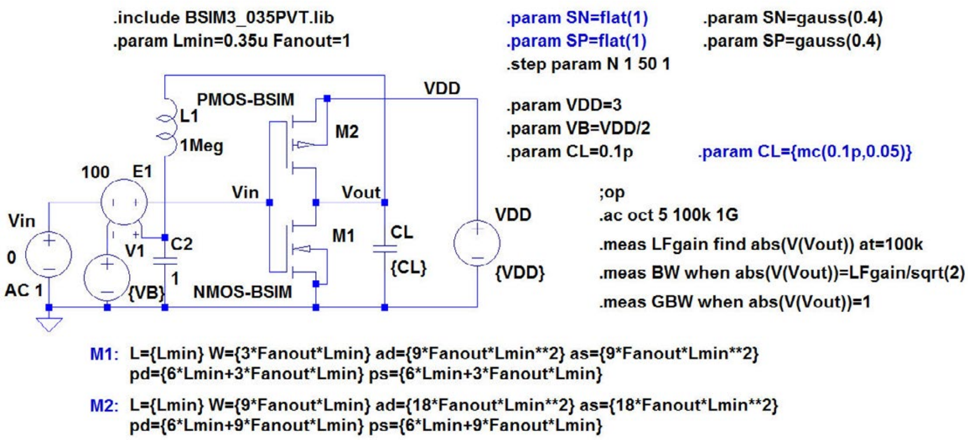


Figure 6.26: LTspice schematic of the inverting amplifier from fig. 6.16 with stochastic specification of the speed parameters for Monte Carlo simulation.

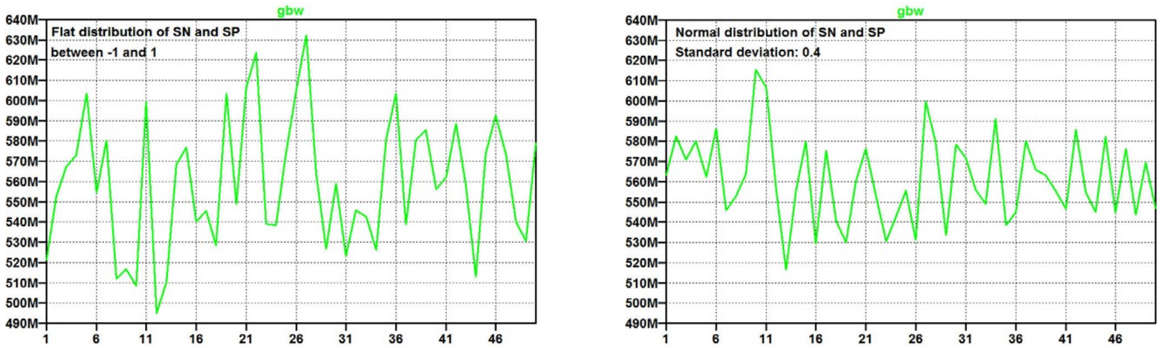
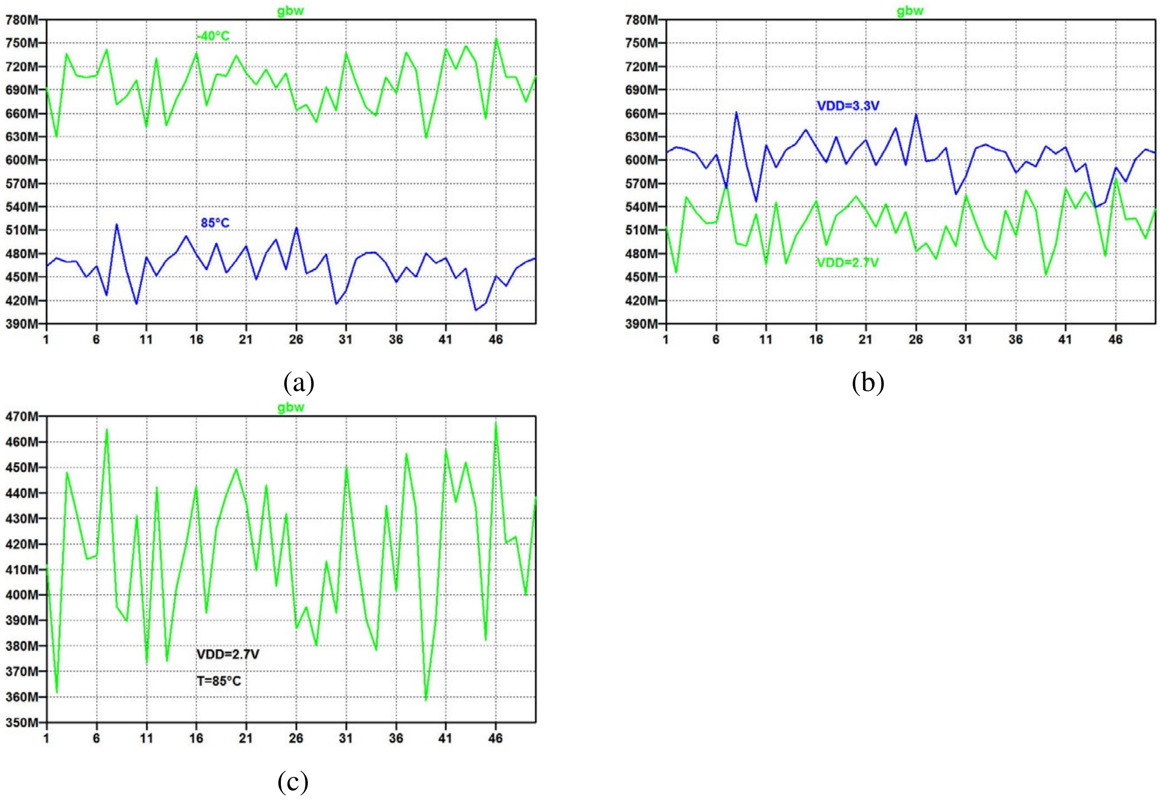


Figure 6.27: Monte Carlo simulation of the unity gain bandwidth of the inverting amplifier from fig. 6.26.

Fig. 6.27 shows the unity gain bandwidth both for the simulations with a flat distribution and with a Gaussian distribution. This figure may be compared to fig. 6.14 with the results of the worst case corner simulations. Obviously, assuming the Gaussian distribution, the SS corner and the FF corner are rather unlikely worst case situations.

Also shown in fig. 6.26 (as a comment) is a stochastic specification of the capacitor C_L, specifying a tolerance of $\pm 5\%$. An advantage of the Monte Carlo simulation is that varying values are specified without using '.step' commands but with stochastic variables instead. Thus, the limitation of three nested '.step' commands is relaxed.

The process parameters, including the capacitor variation, are normally to be considered as stochastic variables. Conversely, temperature range and supply voltage range are specified operating ranges for which the circuit should be designed. Therefore, it makes sense to run simulations with the process parameters specified as stochastic variables and temperature and/or supply voltage specified as minimum and maximum limits. Fig. 6.28 shows the results of Monte Carlo simulations for the inverting amplifier from fig. 6.26 for maximum and minimum temperature (6.28(a)) and maximum and minimum supply voltage (6.28(b)). Obviously, the worst case combination is a low supply voltage and a high temperature, so the optimization may proceed by investigating this combination. Fig. 6.28 (c) shows the result of a Monte Carlo simulation with this combination of supply voltage and temperature. The average bandwidth is found to be 417 MHz with a standard deviation of 28 MHz. For comparison, the bandwidth for the typical PVT combination is 560 MHz.



(a) (b) (c)

Figure 6.28: Monte Carlo simulation of the unity gain bandwidth of the inverting amplifier from fig. 6.26 with stochastic variation of process parameters and capacitor value and operating range limits for temperature (a) and supply voltage (b). Worst case combination is low supply voltage and high temperature (c).

Hints and pitfalls

- Process variations may be described by model files for slow, typical and fast components, leading to different process corners, temperature corners and voltage corners (PVT corners).
- In order to step between slow, typical and fast components in a single simulation, ‘speed’ parameters may be applied to distinguish the different process parameters.
- Stepping between different PVT corners can be achieved by stepping the relevant speed parameters.
- Stepping between selected PVT corners can be customized using a ‘step count’ parameter and a table specification of the selected corners, see example on page 192.
- Temperature is predefined as a parameter ‘`temp`’ in LTspice. Do not use this name for another parameter.
- LTspice supports up to three levels of nested ‘.step’ commands.
- ‘.step’ commands are executed in the sequence in which they appear in the SPICE Netlist. This corresponds to the sequence in which they are inserted in the schematic.
- ‘.measure’ commands are very useful for calculating design parameters from simulations.
- The results of ‘.measure’ commands are found in the error log file (‘Ctrl-L’).
- When using ‘.measure’ commands in combination with ‘.step’ commands, the resulting tables in the error log file can be presented in the waveform viewer by using a right mouse click and selecting ‘Plot .step’ed .meas data’.
- A Monte Carlo simulation is useful for simulating stochastic variations. With a Monte Carlo simulation, several design parameters can simultaneously be subject to variations in a single simulation run.
- Monte Carlo simulations require a fairly large number of simulations, implying that they might be slow or require a fast computer.

References

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

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

Weste, NHE. & Harris, DM. 2010, *CMOS VLSI Design, A Circuits and Systems Perspective*, Fourth Edition, Addison-Wesley, Boston, USA.

Problems

6.1

For an NMOS transistor, assume that typical, slow and fast models are given by the Shichman-Hodges model parameters shown in fig. P6.1. Design a transistor model which combines the three models into one using a speed parameter '`SN`' with `SN`$= -1$ for the slow model, 0 for the typical model and 1 for the fast model. Find the gate-source voltage, the transconductance and the output conductance for a transistor with $V_{GS} = V_{DS}$, $I_D = 0.4$ mA, $W = 20$ µm and $L = 1$ µm for typical model parameters and for slow and fast process corners.

Typical model:

```
Kp=190u, Vto=0.57, Lambda=0.16
Gamma=0.5, Phi=0.7
```

Slow model:

```
Kp=170u, Vto=0.65, Lambda=0.17
Gamma=0.5, Phi=0.7
```

Fast model:

```
Kp=220u, Vto=0.45, Lambda=0.14
Gamma=0.5, Phi=0.7
```

Figure P6.1

6.2

Design a PMOS transistor to provide a g_m of at least 0.48 mA/V with $V_{GS} = V_{DS}$ and $I_D = 30$ μA using a worst case combination of temperature and process variations. Assume the BSIM3 model shown in fig. 6.2 on page 184 and a temperature range from -40°C to 85°C. Use a channel length of $L = 1$ μm.

6.3

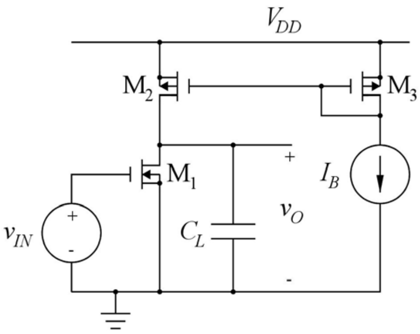


Figure P6.3

For the common source stage shown in fig. P6.3, assume $L_1 = L_2 = L_3 = 1$ µm, $W_1 = 22$ µm, $W_2 = W_3 = 20$ µm, $I_B = 140$ µA and $V_{DD} = 3$ V. Assume the BSIM3 models shown in fig. 6.2 on page 184 and a temperature of 27°C. Also assume that process variations cause C_L to have a value in the range of 1.3 pF to 1.7 pF. Find the process corners for the unity gain frequency.

6.4

For the digital inverter `X2` in fig. 6.18 on page 196, find the worst case delay time considering both process variations, temperature variations from −40 °C to 85°C and supply voltage variations from 2.7 V to 3.3 V.

6.5

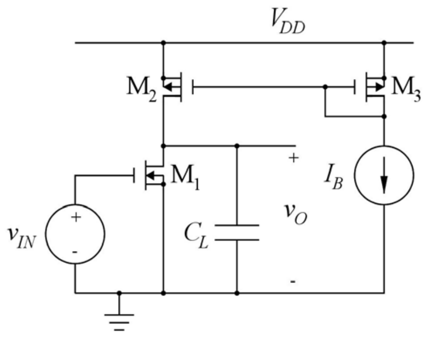


Figure P6.5

For the common source stage shown in fig. P6.5, find the worst case corner for unity gain bandwidth (lowest unity gain bandwidth) for temperature variations and supply voltage variations. Assume $C_L = 1.5$ pF, $L_1 = L_2 = L_3 = 1$ μm, $W_1 = 22$ μm, $W_2 = W_3 = 20$ μm, $I_B = (V_{DD} - 0.9\text{ V})/(15\text{ k}\Omega)$ and $V_{DD} = 3$ V. Assume the BSIM3 models shown in fig. 6.2 on page 184 with typical process parameters, temperature variations from -40°C to 85°C and supply voltage variations from 2.7 V to 3.3 V. Run a Monte Carlo simulation for the worst case combination of temperature and supply voltage with stochastic variations of the process parameters for the transistors and the capacitor C_L. Assume a standard deviation of 0.4 for the process speed parameters and a capacitor tolerance of $\pm 5\%$. Estimate mean value and standard deviation of the unity gain frequency for the worst case combination of temperature and supply voltage.

Answers

6.1: Typical: $V_{GS} = 0.996$ V; $g_m = 1.88$ mA/V; $g_{ds} = 55.2$ μA/V.
Slow: $V_{GS} = 1.10$ V; $g_m = 1.80$ mA/V; $g_{ds} = 57.3$ μA/V.
Fast: $V_{GS} = 0.853$ V; $g_m = 1.99$ mA/V; $g_{ds} = 50.0$ μA/V.

6.2: Worst case corner: slow process, high temperature. $W = 280$ μm.

6.3: Process corners: $(SN, SP, C_L) = (-1, -1, 1.7$ pF): $GBW = 85.5$ MHz.
$(SN, SP, C_L) = (1, -1, 1.7$ pF): $GBW = 100.8$ MHz.
$(SN, SP, C_L) = (-1, 1, 1.3$ pF): $GBW = 112.3$ MHz.
$(SN, SP, C_L) = (1, 1, 1.3$ pF): $GBW = 133.2$ MHz.

6.4: Worst case: falling output, low supply voltage, high temperature, slow processes: $t_{delay} = 191$ ps.

6.5: Mean value of GBW: 85.6 MHz. Standard deviation: 3.6 MHz.

Tutorial 7 – Importing and exporting files

This tutorial shows a few ways to import files to LTspice and export files from LTspice. The main file format for this is the SPICE Netlist which describes the circuit to the simulator. But also output files from simulations are available, see for instance fig. 1.12 on page 25, and we have seen already in Tutorial 3 how model files can be imported to LTspice. After having completed the tutorial, you should be able to

–	import a netlist file to LTspice and run simulations directly from the netlist.
–	use a netlist input to define a subcircuit and create a symbol for the subcircuit.
–	export output netlist files from schematics.
–	export output files from simulations.

Example 7.1: Importing a netlist file describing a current conveyor.

A current conveyor is a generic combination of a voltage follower and a current follower or current inverter. It is a three-terminal device with one input terminal, Y, one output terminal, Z, and one combined input-output terminal, X, see fig. 7.1. The more popular form of the current conveyor is the second generation current conveyor referred to as CCII (Sedra & Smith 1970). This device is described by the terminal relations

$$\left\{\begin{array}{c} i_Y \\ v_X \\ i_Z \end{array}\right\} = \left\{\begin{array}{ccc} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & \pm 1 & 0 \end{array}\right\} \left\{\begin{array}{c} v_Y \\ i_X \\ v_Z \end{array}\right\} \tag{7.1}$$

The definition incorporates a positive version and a negative version, corresponding to $i_Z = i_X$ and $i_Z = -i_X$, respectively.

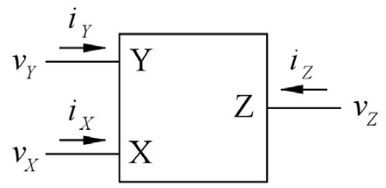


Figure 7.1: Current conveyor terminal definition.

A simple CMOS implementation of a CCII+ current conveyor (with each node labeled by a number) is shown in fig. 7.2 (Bruun 1994), and a netlist corresponding to this schematic is shown in fig. 7.3.

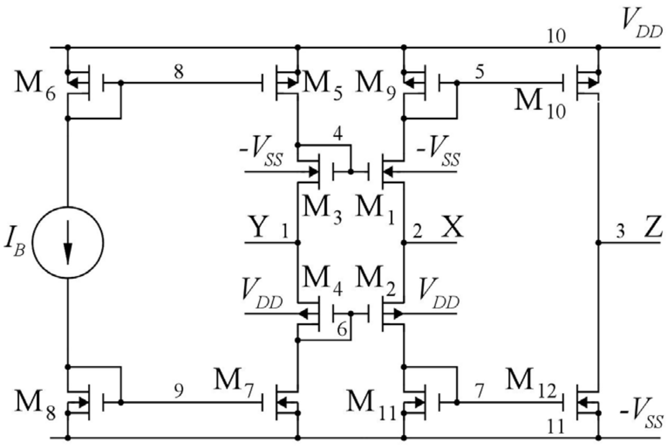


Figure 7.2: CMOS current conveyor, CCII+.

Current conveyor netlist description.

```
*CMOS Second Generation Current Conveyor
*Y-input: Node 1
*X-input: Node 2
*Z-output: Node 3
*Positive supply voltage: Node 10
*Negative supply voltage: Node 11
*Bias current: IB, from node 8 to node 9

*Circuit description
M1 5 4 2 11 NMOS-BSIM L=1.0u W=30u AD=30e-12 AS=30e-12 PD=32u PS=32u
M2 7 6 2 10 PMOS-BSIM L=1.0u W=90u AD=90e-12 AS=90e-12 PD=92u PS=92u
M3 4 4 1 11 NMOS-BSIM L=1.0u W=30u AD=30e-12 AS=30e-12 PD=32u PS=32u
M4 6 6 1 10 PMOS-BSIM L=1.0u W=90u AD=90e-12 AS=90e-12 PD=92u PS=92u
M5 4 8 10 10 PMOS-BSIM L=1.0u W=90u AD=90e-12 AS=90e-12 PD=92u PS=92u
M6 8 8 10 10 PMOS-BSIM L=1.0u W=90u AD=90e-12 AS=90e-12 PD=92u PS=92u
M7 6 9 11 11 NMOS-BSIM L=1.0u W=30u AD=30e-12 AS=30e-12 PD=32u PS=32u
M8 9 9 11 11 NMOS-BSIM L=1.0u W=30u AD=30e-12 AS=30e-12 PD=32u PS=32u
M9 5 5 10 10 PMOS-BSIM L=1.0u W=90u AD=90e-12 AS=90e-12 PD=92u PS=92u
M10 7 7 11 11 NMOS-BSIM L=1.0u W=30u AD=30e-12 AS=30e-12 PD=32u PS=32u
M11 3 5 10 10 PMOS-BSIM L=1.0u W=90u AD=90e-12 AS=90e-12 PD=92u PS=92u
M12 3 7 11 11 NMOS-BSIM L=1.0u W=30u AD=30e-12 AS=30e-12 PD=32u PS=32u
```

Figure 7.3: Netlist description of the current conveyor from fig. 7.2.

The netlist file is named ‘ccii.net’ and can be opened in LTspice with the command ‘File → Open’ (or on the toolbar) using ‘Files of type: Netlists’. In LTspice, a simulation can be executed directly from the netlist file, but first the simulation must be specified. This is done by directly inserting the required SPICE Directive in the netlist file. Also, the transistor model file (‘bsim3_035.lib’) must be included, and it should be in the same folder as the netlist file unless the full path name is specified.

Fig. 7.4 shows the netlist file edited in LTspice to include model file, supply voltages, bias current, input signals v_X and i_Y, output load R_L and a simulation command for a DC sweep of the input current i_X. Also included is a ‘.end’ directive to mark the end of the netlist file. This is not required in order to run a simulation, but it is good practice to mark the end of the file, and when LTspice generates netlist files from a schematic, the ‘.end’ directive is also automatically inserted.

SPICE Netlist from the LTspice file editor.

```
*CMOS Second Generation Current Conveyor
*Y-input: Node 1
*X-input: Node 2
*Z-output: Node 3
*Positive supply voltage: Node 10
*Negative supply voltage: Node 11
*Bias current: IB, from node 8 to node 9

*Circuit description
M1 5 4 2 11 NMOS-BSIM L=1.0u W=30u AD=30e-12 AS=30e-12 PD=32u PS=32u
M2 7 6 2 10 PMOS-BSIM L=1.0u W=90u AD=90e-12 AS=90e-12 PD=92u PS=92u
M3 4 4 1 11 NMOS-BSIM L=1.0u W=30u AD=30e-12 AS=30e-12 PD=32u PS=32u
M4 6 6 1 10 PMOS-BSIM L=1.0u W=90u AD=90e-12 AS=90e-12 PD=92u PS=92u
M5 4 8 10 10 PMOS-BSIM L=1.0u W=90u AD=90e-12 AS=90e-12 PD=92u PS=92u
M6 8 8 10 10 PMOS-BSIM L=1.0u W=90u AD=90e-12 AS=90e-12 PD=92u PS=92u
M7 6 9 11 11 NMOS-BSIM L=1.0u W=30u AD=30e-12 AS=30e-12 PD=32u PS=32u
M8 9 9 11 11 NMOS-BSIM L=1.0u W=30u AD=30e-12 AS=30e-12 PD=32u PS=32u
M9 5 5 10 10 PMOS-BSIM L=1.0u W=90u AD=90e-12 AS=90e-12 PD=92u PS=92u
M10 7 7 11 11 NMOS-BSIM L=1.0u W=30u AD=30e-12 AS=30e-12 PD=32u PS=32u
M11 3 5 10 10 PMOS-BSIM L=1.0u W=90u AD=90e-12 AS=90e-12 PD=92u PS=92u
M12 3 7 11 11 NMOS-BSIM L=1.0u W=30u AD=30e-12 AS=30e-12 PD=32u PS=32u

*Transistor models
.include bsim3_035.lib

*Supply voltages and bias current
VDD 10 0 1.5V
VSS 0 11 1.5V
IB 8 9 10u

*Input signals
VY 1 0 0
IX 0 2 0

*Output load
RL 3 0 10k

*Analysis
.dc ix -100u 100u 1u

.end
```

Figure 7.4: SPICE Netlist for running a ‘.dc’ simulation of the current conveyor from fig. 7.2.

The simulation is run directly from the SPICE Netlist using the command 'Simulate → Run' or using the 'Run'-symbol on the toolbar. Fig. 7.5 shows the result of the simulation. The traces to be displayed are selected in the plot window using the command 'Plot Settings → Add trace' or the command 'Plot Settings → Visible Traces'. The command 'Visible Traces' is also available with the netlist window active ('View → Visible Traces') and on the toolbar, symbol , also see page 23. Fig. 7.5 shows the input current i_X and the output current $i_Z = -I(R_L)$. Obviously, the output current is almost the same as the input current as expected from (7.1).

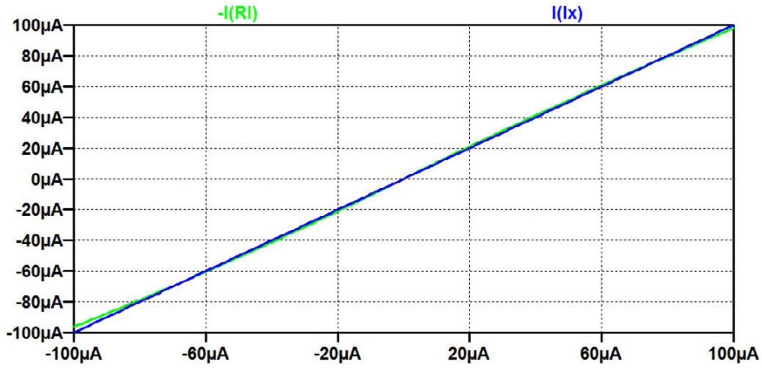


Figure 7.5: Simulated output current $i_Z = -I(R_L)$ for the current conveyor from the SPICE Netlist shown in fig. 7.4.

Example 7.2: Creating a subcircuit from a netlist.

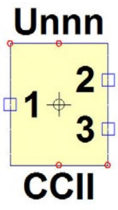
The current conveyor is a generic building block in analog circuit design, so it is of interest to have it as a generic subcircuit with a symbol resembling the symbol shown in fig. 7.1. For this, the netlist from fig. 7.4 must be modified so that it starts with a '.subckt' directive and ends with a '.ends' directive. Fig. 7.6 shows the netlist with a minimum of changes required to turn it into a subcircuit specification. The '.subckt' directive and the '.ends' directive have been inserted and the simulation command has been removed, but the file still contains both model specifications and specifications of supply voltages and bias current. The '.subckt' directive specifies the name of the subcircuit and the order of the terminals.

SPICE Netlist, current conveyor subcircuit.

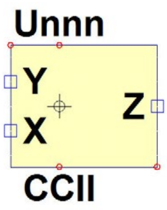
```
*CMOS Second Generation Current Conveyor
*Y-input: Node 1
*X-input: Node 2
*Z-output: Node 3
*Positive supply voltage: Node 10
*Negative supply voltage: Node 11
*Bias current: IB, from node 8 to node 9

.subckt CCII 1 2 3

*Circuit description
M1 5 4 2 11 NMOS-BSIM L=1.0u W=30u AD=30e-12 AS=30e-12 PD=32u PS=32u
M2 7 6 2 10 PMOS-BSIM L=1.0u W=90u AD=90e-12 AS=90e-12 PD=92u PS=92u
M3 4 4 1 11 NMOS-BSIM L=1.0u W=30u AD=30e-12 AS=30e-12 PD=32u PS=32u
M4 6 6 1 10 PMOS-BSIM L=1.0u W=90u AD=90e-12 AS=90e-12 PD=92u PS=92u
M5 4 8 10 10 PMOS-BSIM L=1.0u W=90u AD=90e-12 AS=90e-12 PD=92u PS=92u
M6 8 8 10 10 PMOS-BSIM L=1.0u W=90u AD=90e-12 AS=90e-12 PD=92u PS=92u
M7 6 9 11 11 NMOS-BSIM L=1.0u W=30u AD=30e-12 AS=30e-12 PD=32u PS=32u
M8 9 9 11 11 NMOS-BSIM L=1.0u W=30u AD=30e-12 AS=30e-12 PD=32u PS=32u
M9 5 5 10 10 PMOS-BSIM L=1.0u W=90u AD=90e-12 AS=90e-12 PD=92u PS=92u
M10 7 7 11 11 NMOS-BSIM L=1.0u W=30u AD=30e-12 AS=30e-12 PD=32u PS=32u
M11 3 5 10 10 PMOS-BSIM L=1.0u W=90u AD=90e-12 AS=90e-12 PD=92u PS=92u
M12 3 7 11 11 NMOS-BSIM L=1.0u W=30u AD=30e-12 AS=30e-12 PD=32u PS=32u

*Transistor models
.include bsim3_035.lib

*Supply voltages and bias current
VDD 10 0 1.5V
VSS 0 11 1.5V
IB 8 9 10u

.ends
```

Figure 7.6: SPICE Netlist specifying the current conveyor as a subcircuit including transistor models, supply voltages and bias current.

Next, we need a symbol for the subcircuit. This is achieved from the netlist editor by placing the cursor in the line '.subckt CCII 1 2 3' and right clicking. This opens a window where you can select 'Create Symbol' and answer 'Yes' in the dialogue window which opens. The auto-generated symbol is shown in fig. 7.7(a). Using the symbol editor as described on page 150, this is easily modified



(a)



(b)

Figure 7.7: LTspice symbol for the current conveyor. Autogenerated from the netlist (a). Edited from the autogenerated symbol (b).

into the symbol shown in fig. 7.7(b). When you save the symbol CCII.asy, it is by default saved in the folder with the LTspice program, '<LTspiceIV> \lib\sym\Autogenerated', so when inserting the symbol in a schematic using 'Edit → Component' or hotkey 'F2', you must select the folder '[Autogenerated]' in the component selection window, see fig. 1.3 on page 15. You may also select to save the symbol in the same folder as your circuits using the symbol.

Fig. 7.8 shows a schematic with the current conveyor symbol and input signals and load resistor. From this schematic, we can run the same simulation as the simulation run from the netlist file in fig. 7.4, and the result is as shown in fig. 7.5.

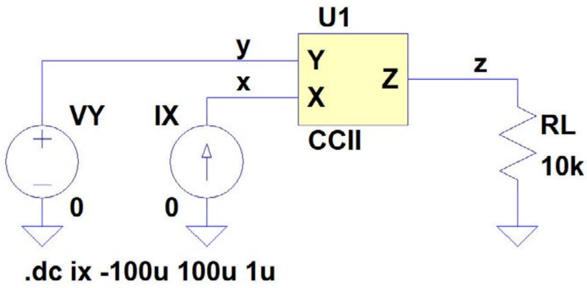


Figure 7.8: LTspice schematic for running the same simulation as specified in the netlist in fig. 7.4.

The subcircuit defined in fig. 7.6 includes both the transistor model file, the supply voltages and the bias current. For added flexibility, it may be desirable to have the specification of transistor models, supply voltages and bias current at the top level of the circuit hierarchy, similar to the examples 5.1, 5.2 and 5.4. An easy way to achieve this is to specify the supply voltages and the bias current as parameters and omit the transistor model specification in the subcircuit. Fig. 7.9 shows the netlist file for this. In this netlist file, default values for the supply voltages and the bias current are specified. In order to override the default values, you must specify the parameters VDD, VSS and IB for the current

SPICE Netlist, current conveyor subcircuit.

```
*CMOS Second Generation Current Conveyor
*Y-input: Node 1
*X-input: Node 2
*Z-output: Node 3
*Positive supply voltage: Node 10
*Negative supply voltage: Node 11
*Bias current: IB, from node 8 to node 9

.subckt CCII 1 2 3

*Circuit description
M1 5 4 2 11 NMOS-BSIM L=1.0u W=30u AD=30e-12 AS=30e-12 PD=32u PS=32u
M2 7 6 2 10 PMOS-BSIM L=1.0u W=90u AD=90e-12 AS=90e-12 PD=92u PS=92u
M3 4 4 1 11 NMOS-BSIM L=1.0u W=30u AD=30e-12 AS=30e-12 PD=32u PS=32u
M4 6 6 1 10 PMOS-BSIM L=1.0u W=90u AD=90e-12 AS=90e-12 PD=92u PS=92u
M5 4 8 10 10 PMOS-BSIM L=1.0u W=90u AD=90e-12 AS=90e-12 PD=92u PS=92u
M6 8 8 10 10 PMOS-BSIM L=1.0u W=90u AD=90e-12 AS=90e-12 PD=92u PS=92u
M7 6 9 11 11 NMOS-BSIM L=1.0u W=30u AD=30e-12 AS=30e-12 PD=32u PS=32u
M8 9 9 11 11 NMOS-BSIM L=1.0u W=30u AD=30e-12 AS=30e-12 PD=32u PS=32u
M9 5 5 10 10 PMOS-BSIM L=1.0u W=90u AD=90e-12 AS=90e-12 PD=92u PS=92u
M10 7 7 11 11 NMOS-BSIM L=1.0u W=30u AD=30e-12 AS=30e-12 PD=32u PS=32u
M11 3 5 10 10 PMOS-BSIM L=1.0u W=90u AD=90e-12 AS=90e-12 PD=92u PS=92u
M12 3 7 11 11 NMOS-BSIM L=1.0u W=30u AD=30e-12 AS=30e-12 PD=32u PS=32u

*Supply voltages and bias current
VDD 10 0 {VDD}
VSS 0 11 {VSS}
IB 8 9 {IB}
.param VDD=1.5 VSS=1.5 IB=10u

.ends
```

Figure 7.9: SPICE Netlist specifying the current conveyor as a subcircuit with supply voltages and bias current specified as parameters with default values.

conveyor symbol, rather than using a '.param' SPICE Directive. This is done by a 'Ctrl-right click' on the current conveyor symbol. This opens the 'Component Attribute Editor' shown in fig. 7.10, and new values for `VDD`, `VSS` and `IB` can be inserted in the 'Spiceline' as shown in fig. 7.10. Also mark the 'Spiceline' to be visible in the schematic by inserting a 'x' in the rightmost column in the 'Component Attribute Editor'.

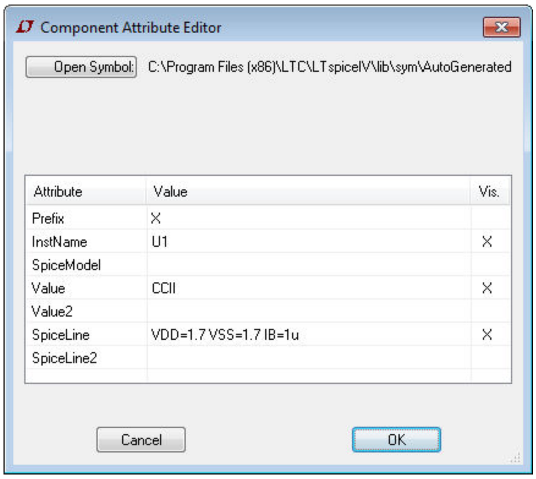


Figure 7.10: Component Attribute Editor for specifying parameters for the current conveyor subcircuit.

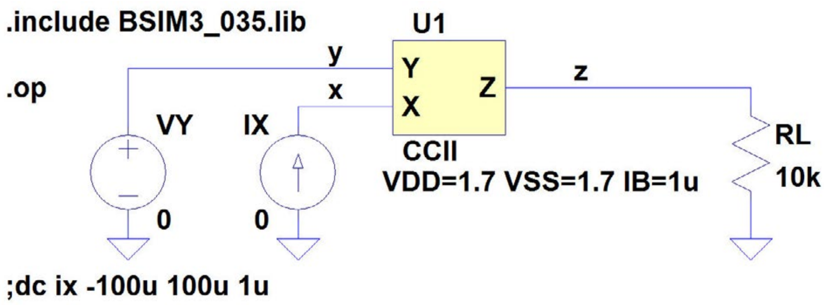


Figure 7.11: LTspice schematic for running simulations with specifications of subcircuit parameters.

Fig. 7.11 shows the LTspice schematic including overriding parameter specifications and including the transistor model file. Also shown is a '.op' simulation command. Running the '.op' simulation, the bias conditions for the current conveyor can be verified if LTspice has been set up to save the subcircuit voltages and currents. This is done by the command 'Tools → Control Panel' where you select the tab 'Save Defaults'. Here you tick 'Save Subcircuit Node Voltages' and 'Save Subcircuit Device Currents'.

In the specification of the supply voltages, you may use another parameter as shown in fig. 7.12. Here, the numeric value of the supply voltage is defined as a parameter '`Vsup`', and this parameter is stepped from 1.3 V to 1.7 V using a '.step param' directive. Also the value of the load resistor has been changed to 20 kΩ. With this value of R_L, the Z-output cannot deliver an output current of 100 µA because the output voltage exceeds the supply rails for $i_Z = \pm 100$ µA. This is shown in fig. 7.13 where the output current is plotted versus the input current for three different values of supply voltage.

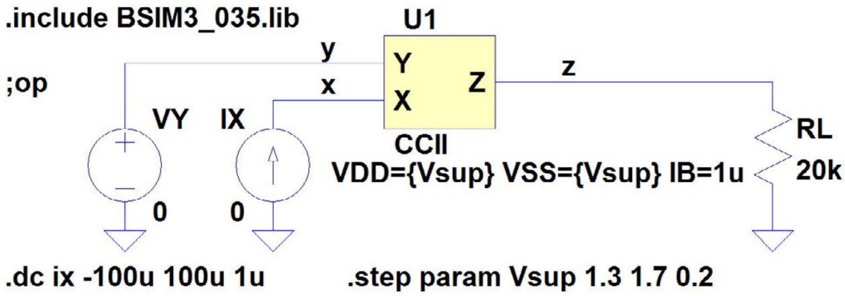


Figure 7.12: LTspice schematic for running simulations with different values of supply voltage.

An alternative way for specifying supply voltages and bias current at top level of the circuit hierarchy is by introducing terminals for the supply voltages and the bias current in the schematic symbol. This is achieved by first specifying the terminals in the netlist description as shown in fig. 7.14 and then designing a symbol including these terminals in the same way as the symbol from fig. 7.7.

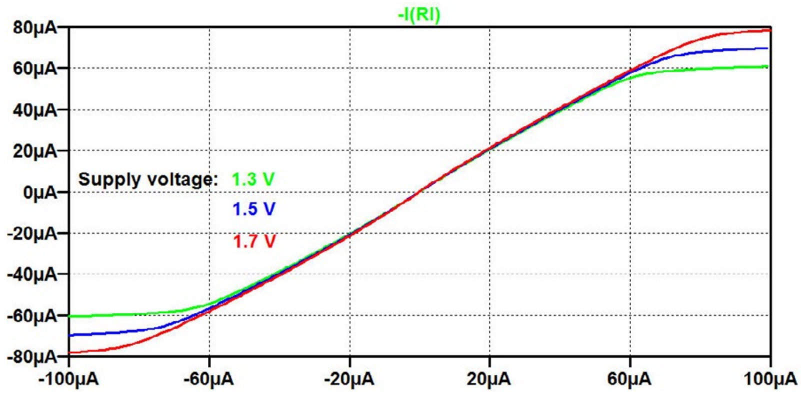


Figure 7.13: Simulation plot for the '.dc' simulation specified in fig. 7.12.

SPICE Netlist, current conveyor subcircuit with terminals for supply voltages and bias current.

```
*CMOS Second Generation Current Conveyor
*Y-input: Node 1
*X-input: Node 2
*Z-output: Node 3
*Positive supply voltage: Node 10
*Negative supply voltage: Node 11
*Bias current: IB, from node 8 to node 9

*Connections: Y-input
*             | X-input
*             | | Z-output
*             | | | Bias current out
*             | | | | Bias current in
*             | | | | | Positive supply voltage
*             | | | | | | Negative supply voltage
*             | | | | | | |
.subckt CCII  1 2 3 8 9 10 11

*Circuit description
M1 5 4 2 11 NMOS-BSIM L=1.0u W=30u AD=30e-12 AS=30e-12 PD=32u PS=32u
M2 7 6 2 10 PMOS-BSIM L=1.0u W=90u AD=90e-12 AS=90e-12 PD=92u PS=92u
M3 4 4 1 11 NMOS-BSIM L=1.0u W=30u AD=30e-12 AS=30e-12 PD=32u PS=32u
M4 6 6 1 10 PMOS-BSIM L=1.0u W=90u AD=90e-12 AS=90e-12 PD=92u PS=92u
M5 4 8 10 10 PMOS-BSIM L=1.0u W=90u AD=90e-12 AS=90e-12 PD=92u PS=92u
M6 8 8 10 10 PMOS-BSIM L=1.0u W=90u AD=90e-12 AS=90e-12 PD=92u PS=92u
M7 6 9 11 11 NMOS-BSIM L=1.0u W=30u AD=30e-12 AS=30e-12 PD=32u PS=32u
M8 9 9 11 11 NMOS-BSIM L=1.0u W=30u AD=30e-12 AS=30e-12 PD=32u PS=32u
M9 5 5 10 10 PMOS-BSIM L=1.0u W=90u AD=90e-12 AS=90e-12 PD=92u PS=92u
M10 7 7 11 11 NMOS-BSIM L=1.0u W=30u AD=30e-12 AS=30e-12 PD=32u PS=32u
M11 3 5 10 10 PMOS-BSIM L=1.0u W=90u AD=90e-12 AS=90e-12 PD=92u PS=92u
M12 3 7 11 11 NMOS-BSIM L=1.0u W=30u AD=30e-12 AS=30e-12 PD=32u PS=32u

.ends
```

Figure 7.14: SPICE Netlist specifying the current conveyor as a subcircuit with supply voltages and bias current connected to separate terminals.

Fig. 7.15 shows a schematic with this definition of the conveyor symbol and with the bias current source replaced by a resistor which controls the bias current for the conveyor. With the resistor shown in fig. 7.15, the bias current is about 10 µA for a supply voltage of ± 1.5 V.

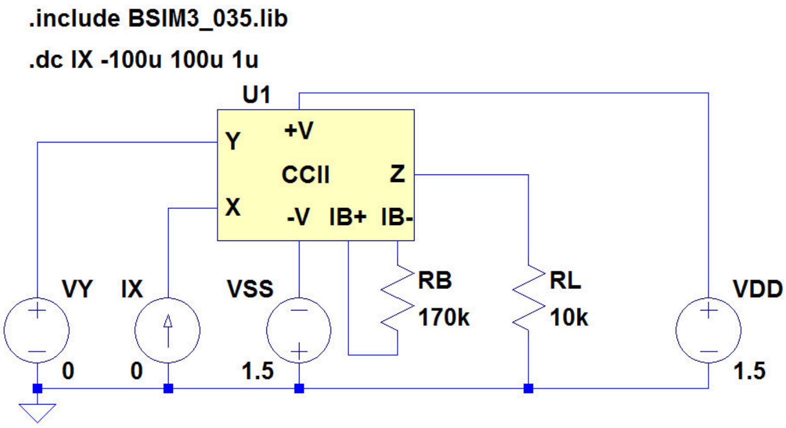


Figure 7.15: LTspice schematic using a current conveyor symbol with supply voltages and bias current connected to separate terminals.

Example 7.3: Exporting a netlist.

Sometimes you may wish to export a netlist to another design system for further processing. LTspice has a tool for netlist export, 'Tools → Export Netlist', by which a netlist for a schematic can be exported to a number of different file formats, typically for use by PCB layout editors. For integrated circuit design, this feature is not so useful. You would rather need to export to IC design tools such as Cadence, Synopsis or Tanner EDA. For this, you can often use the netlist which can be viewed in LTspice by the command 'View → SPICE Netlist'. This netlist file is not by default automatically saved by LTspice, but from the command 'Tools → Control Panel', you can select the tab 'Operation'. Here you find 'Automatically delete .net files [*]:' and change the selection from 'Yes' to 'No'. This will cause the netlist file to be saved automatically with the extension '.net'. The file format is the generic SPICE format for netlists (Vladimirescu 1994) with the addition of the command '.backanno' used by LTspice.

The '.net' file can also be opened and edited in LTspice with the command 'File → Open' (or  on the toolbar) using 'Files of type: Netlists'.

As an example, fig. 7.16 shows the netlist file for the schematic from in fig. 7.12. This is a netlist file containing both references to a transistor model file and to a subcircuit, so without these files it cannot be used by another Spice simulator. In order to obtain maximum portability to another system, an expanded netlist can be generated: Right click in the file and select 'Generate Expanded Listing' in the window which opens. The expanded netlist file is saved as a new file with extension '.sp'. Fig. 7.17 shows the expanded netlist file corresponding to the netlist from fig. 7.16. We notice that the subcircuit is expanded into the individual transistors, the supply voltages and the bias current. Also, the values for supply voltages are replaced by the parameter values specified

SPICE Netlist from schematic.

```
* M:\LTspice\Tutorial07\fig7_12.asc
RL z 0 20k
VY y 0 0
IX 0 x 0
XU1 y x z CCII VDD={Vsup} VSS={Vsup} IB=1u
.dc ix -100u 100u 1u
.include BSIM3_035.lib
;op
.step param Vsup 1.3 1.7 0.2
.lib M:\LTspice\Tutorial07\ccii.net
.backanno
.end
```

Figure 7.16: SPICE Netlist for the circuit from fig. 7.12.

as the first step in the '.step param' command. Further, the '.include BSIM3_035.lib' command is expanded into the two transistor models. Each of the models is collapsed into a single, very long line in the expanded netlist file, so in fig. 7.17 only the start of the model lines is shown. Finally, the '.backanno' command has been removed.

The expanded netlist also contains the simulation command, so a simulation can be run directly from the netlist, but the '.step param' command is not included. Instead, the parameter values for the first step are inserted, and the simulation will show only the first step. Fig. 7.18 shows the simulation run from the netlist. It may be compared to the simulation shown in fig. 7.13 run from the schematic.

Expanded SPICE Netlist from schematic.

```
* M:\LTspice\Tutorial07\fig7_12.asc
rl z 0 20k
vy y 0 0
ix 0 x 0
m:u1:1 u1:5 u1:4 x u1:11 nmos-bsim l=1.0u w=30u ad=30e-12 as=30e-12 pd=32u ps=32u
m:u1:2 u1:7 u1:6 x u1:10 pmos-bsim l=1.0u w=90u ad=90e-12 as=90e-12 pd=92u ps=92u
m:u1:3 u1:4 u1:4 y u1:11 nmos-bsim l=1.0u w=30u ad=30e-12 as=30e-12 pd=32u ps=32u
m:u1:4 u1:6 u1:6 y u1:10 pmos-bsim l=1.0u w=90u ad=90e-12 as=90e-12 pd=92u ps=92u
m:u1:5 u1:4 u1:8 u1:10 u1:10 pmos-bsim l=1.0u w=90u ad=90e-12 as=90e-12 pd=92u ps=92u
m:u1:6 u1:8 u1:8 u1:10 u1:10 pmos-bsim l=1.0u w=90u ad=90e-12 as=90e-12 pd=92u ps=92u
m:u1:7 u1:6 u1:9 u1:11 u1:11 nmos-bsim l=1.0u w=30u ad=30e-12 as=30e-12 pd=32u ps=32u
m:u1:8 u1:9 u1:9 u1:11 u1:11 nmos-bsim l=1.0u w=30u ad=30e-12 as=30e-12 pd=32u ps=32u
m:u1:9 u1:5 u1:5 u1:10 u1:10 pmos-bsim l=1.0u w=90u ad=90e-12 as=90e-12 pd=92u ps=92u
m:u1:10 u1:7 u1:7 u1:11 u1:11 nmos-bsim l=1.0u w=30u ad=30e-12 as=30e-12 pd=32u ps=32u
m:u1:11 z u1:5 u1:10 u1:10 pmos-bsim l=1.0u w=90u ad=90e-12 as=90e-12 pd=92u ps=92u
m:u1:12 z u1:7 u1:11 u1:11 nmos-bsim l=1.0u w=30u ad=30e-12 as=30e-12 pd=32u ps=32u
v:u1:dd u1:10 0 1.3
v:u1:ss 0 u1:11 1.3
i:u1:b u1:8 u1:9 1e-006
.model pmos-bsim pmos level=49 version=3.1 tnom=2.69e+01 tox=7.8e-9 xj=1.00e-07 nch=8.44e+
.model nmos-bsim nmos level=49 version=3.1 tnom=27 tox=7.8e-9 xj=1e-07 nch=2.18e+17 vth0=0.
.dc ix -100u 100u 1u
.end
```

Figure 7.17: Expanded SPICE Netlist for the circuit from fig. 7.12.

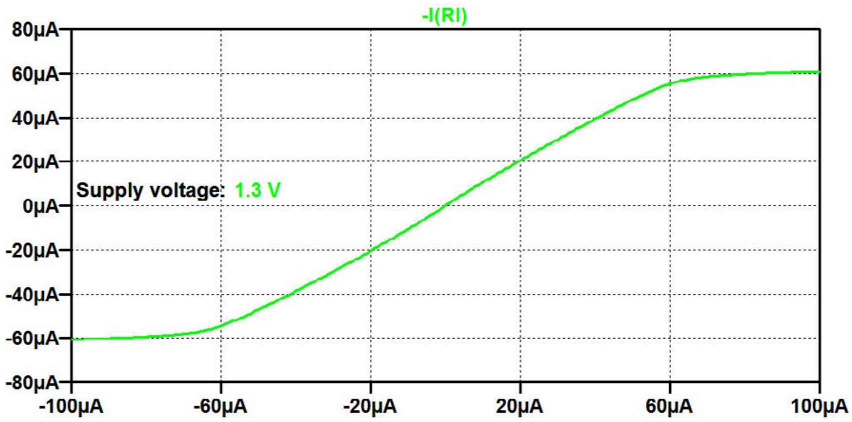


Figure 7.18: Simulation plot for the '.dc' simulation run from the expaned netlist shown in fig. 7.17.

Example 7.4: Exporting other files.

In this book, the command 'Tools → Copy bitmap to Clipboard' has been used extensively for copying schematics and simulation plots into a graphics file format. The 'Tools' command also provides the option of writing to a '.wmf' file. This is a vector file format resulting in fairly small file sizes but less flexible when used in combination with graphics editors.

For simulation plots, also the command 'File → Export' is available as explained on page 24. This is very useful for exporting simulation result to other programs such as Microsoft Excel or Matlab but may require some editing of the exported file in order to ensure compatible formats. Finally, some simulations provide results in the error log file generated from every simulation and saved with the extension '.log'. An important category is the bias point information for transistors available in the '.log' file after running a '.op' simulation. For '.op' simulations (without '.step param' commands), the simulation results (node voltages and device currents) appear directly in a window with the operating point voltages and currents (see page 18), and this information can be copied to the clipboard using 'Ctrl-A' and 'Ctrl-C'. Also for a '.tf' simulations, the simulation results appear directly in a window and can be copied to the clipboard.

It is beyond the scope of this book to go into details concerning how to proceed with the different output files in other programs.

Hints and pitfalls

- LTspice accepts Spice netlists which are compatible with the general syntax for Spice netlists.
- Simulations can be run directly from a netlist file.
- LTspice generates a Spice netlist format which is compatible with the general syntax for Spice netlists.
- The SPICE Netlist for a schematic is automatically saved with a '.net' file extension only if this option is selected in the 'Tools → Control Panel' tab for 'Operation'.
- An expanded netlist file (with file extension '.sp') including subcircuits and models can be generated from a netlist file by selecting 'Generate Expanded Listing' when 'right-clicking' in the netlist file.
- An expanded netlist file ensures maximum portability to other systems.
- A subcircuit may be generated from a netlist file by inserting a '.subckt' command and a '.ends' command.
- A subcircuit symbol may be generated automatically from a netlist by 'right-clicking' in the '.subckt' command line.
- A subcircuit symbol generated from a netlist is by default saved in the folder with the LTspice program, '<LTspiceIV> \lib\sym\Autogenerated'.
- A subcircuit symbol generated automatically from a schematic is by default saved in the same folder as the subcircuit schematic, see page 149.
- Do not use the same filename for a subcircuit and a circuit at a higher level in the circuit hierarchy.

References

Bruun, E. 1994, 'CMOS Current Conveyors', pp. 632-641 (Chapter 11.5) in: Toumazou, C., Battersby, N. and Porta, S., *Circuits & Systems Tutorials*, pre-conference tutorials at *IEEE International Symposium on Circuits and Systems*, London, May/June 1994.

Sedra, AS. & Smith, KC. 1970, 'A Second Generation Current Conveyor and its Applications', *IEEE Trans. Circuit Theory* vol. CT-17, No. 1, pp. 132-134.

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

Problems

7.1

```
*Differential NMOS pair with
*resistive load and differential output

*Noninverting input: Node 3
*Inverting input: Node 2
*Noninverting output: Node 4
*Inverting output: Node 5
*Supply voltage: Node 1
*Bias current: IB from node 6 to ground

*Circuit description
R1 1 4 2.7k
R2 1 5 2.7k
M1 4 2 6 0 NMOS-BSIM L=0.7u W=16u
M2 5 3 6 0 NMOS-BSIM L=0.7u W=16u
```

Figure P7.1

Fig. P7.1 shows a netlist for a differential NMOS pair with differential output. Create a netlist file for LTspice for simulating the circuit with a supply voltage of 3 V, a bias current of 250 μA and a common mode input voltage of 1.5 V. Use the BSIM3 transistor model from fig. 3.10 on page 77. Find the bias values of the output voltages and the small signal differential gain.

7.2

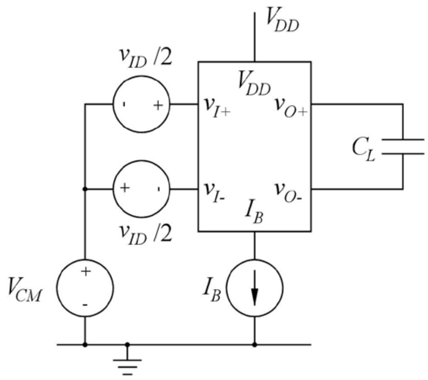


Figure P7.2

Use the netlist from fig. P7.1 to design a subcircuit and a subcircuit symbol for the differential pair. Use the BSIM3 transistor model from fig. 3.10 on page 77. Design the subcircuit to have separate terminals for the supply voltage and the bias current. Insert the differential pair in a test bench as shown in fig. P7.2 and find the −3 dB frequency for the differential gain. Use $V_{DD} = 3$ V, $I_B = 250$ μA, $V_{CM} = 1.5$ V and $C_L = 3$ pF.

7.3

Redefine the subcircuit from problem 7.2 to have the transistor channel width as a parameter which can be defined at top level.
Use the testbench from fig. P7.2 to find the low frequency gain as a function of the channel width for 5 μm$\leq W \leq$ 30 μm. What is the value of the low frequency gain for $W = 5$ μm and $W = 30$ μm?

Answers

7.1: Bias value of output voltages: 1.75 V; Gain: 9.12 V/V.

7.2: -3 dB frequency: 2.84 MHz.

7.3: $W = 5$ μm: Gain: 4.15 V/V; $W = 30$ μm: Gain: 12.9 V/V.

Moving on

LTspice is is an excellent SPICE simulator, easy to use and with the major advantage of being freely available from Linear Technology, http://www.linear.com/designtools/software/.

LTspice supports a large selection of models for standard components, and it supports output file formats for several PCB design systems.
Also, LTspice has a large user community which is extremely helpful whenever questions and problems related to LTspice occur: https://groups.yahoo.com/neo/groups/LTspice/info.

In this book, we have not covered all the possibilities of LTspice, far from. For instance, we have not mentioned FFT analysis, distortion analysis or the generation of Nyquist plots. Nor have we treated issues related to convergence problems in the simulations.

To move on with such issues, use the LTspice documentation from Linear Technology, the comprehensive documentation from (Brocard 2013) or get help from the LTspice user group.

LTspice is not an integrated design tool for CMOS design. It includes a schematic editor and a simulator, but it does not include a layout editor, a DRC program (Design Rule Check), an LVS program (Layout Versus Schematic), or a parasitic extraction program. Also, the selection of transistor models supported by LTspice is limited and design kits from the various foundries and MPW providers are not available.

So in order to move on to full CMOS design, including layout and wafer fabrication, you will need to turn to other EDA systems (Electronic Design Automation) such as Cadence or Tanner EDA.

But for educational purposes and for learning how CMOS circuits behave without having to invest time and money in a design system with complete support, LTspice is second to none.

References

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

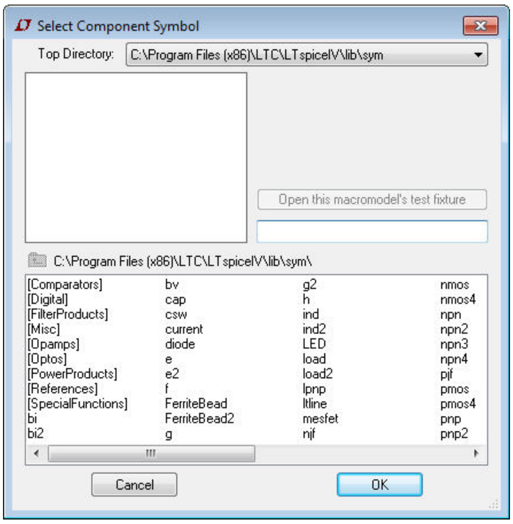
Appendix A –

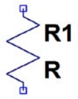
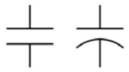
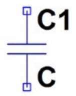
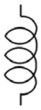
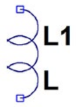
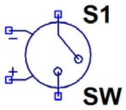
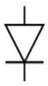
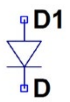
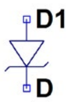
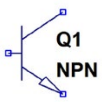
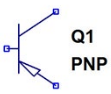
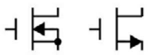
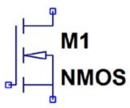
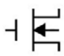
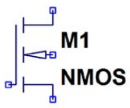
A beginner's guide to components and simulation commands in LTspice

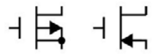
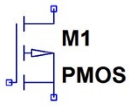
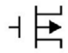
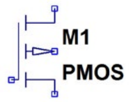
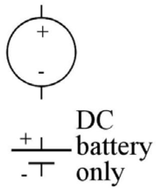
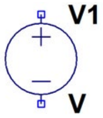
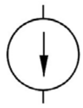
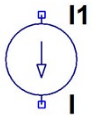
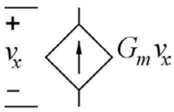
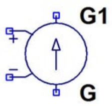
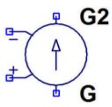
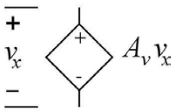
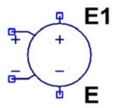
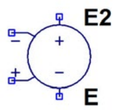
1. Component selection.

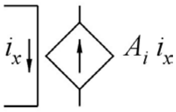
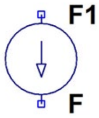
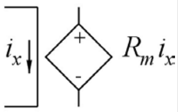
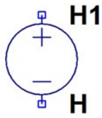
The tables on the following pages give an overview of components typically used in an introductory course in electronics or electrical engineering. In the tables, the LTspice name is the name used in the 'Select Component Symbol' window which is opened by the command 'Edit → Components', symbol on the toolbar, or hotkey 'F2'. (In the Mac version of LTspice by the command 'Draft → Components'.)

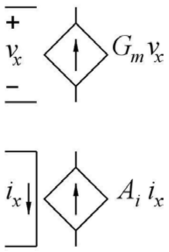
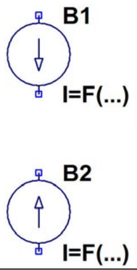
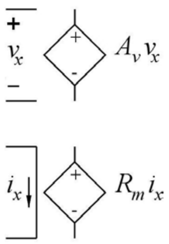
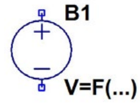
In the LTspice component symbols shown in the tables, the top letter and number is the name and number of the component while the bottom letter is the value of the component or a specification such as type number or reference to a '.model' description.



Name	Symbol		LTspice name and specification	
	Textbook	*LTspice*	*LTspice name and letter*	*Specification*
resistor, resistance			res, R	value in ohm, Ω, see page 14
capacitor, capacitance			cap, C	value in farad, F, compare specification of resistor, page 14
coil, inductor, inductance			ind, L	value in henry, H, compare page 14
switch			sw, S	.model specification, see page 47
diode			diode, D	diode type number or '.model' specification, see page 51
Zener diode			zener, D	diode type number or '.model' specification, see page 51
bipolar npn transistor			npn, Q	transistor type number or '.model' specification, compare page 51
bipolar pnp transistor			pnp, Q	transistor type number or '.model' specification, compare page 51
n-channel MOS transistor, discrete type			nmos, M	transistor type number, see page 67
n-channel MOS transistor, monolitic			nmos4, M	'.model' specification, see page 71

Name	Symbol		LTspice name and specification	
	Textbook	*LTspice*	*LTspice name and letter*	*Specification*
p-channel MOS transistor, discrete type			pmos, M	transistor type number, see page 67
p-channel MOS transistor, monolitic			pmos4, M	'.model' specification, see page 71
independent voltage source			voltage, V	DC value in volt, V, or AC value in volt, V, or time-varying voltage, see page 49 and 44
independent current source			current, I	DC value in ampere, A, or AC value in ampere, A, or time-varying current, compare specification of independent voltage
voltage controlled current source		 	g, G g2, G	value of transconductance in ampere per volt, A/V, see page 29
voltage controlled voltage source		 	e, E e2, E	value of voltage gain E in volt per volt, V/V, compare specification of voltage controlled current source, see page 29

Name	**Symbol**		**LTspice name and specification**	
	Textbook	*LTspice*	*LTspice name and letter*	*Specification*
current controlled current source			f, F	A right click on the symbol opens the 'Component Attribute Editor', compare fig. 1.16 on page 28 and fig. 1.26 on page 35. In 'Value' you specify the name of a voltage source through which the controlling current flows. In 'Value2' you specify the current gain in A/A.
current controlled voltage source			h, H	A right click on the symbol opens the 'Component Attribute Editor', compare fig. 1.16 on page 28 and fig. 1.26 on page 35. In 'Value' you specify the name of a voltage source through which the controlling current flows. In 'Value2' you specify the transresistance in V/A.

Name	**Symbol**		**LTspice name and specification**	
	Textbook	*LTspice*	*LTspice name and letter*	*Specification*
arbitrary controlled current source			bi, B bi2, B	mathematical expression for the current, compare fig. 1.18 on page 29 and fig. 1.24 on page 33.
arbitrary controlled voltage source			bv, B	mathematical expression for the voltage, e.g. `V={Av}*V(Vin)` where `{Av}` is a voltage gain, or `V={Rm}*I(V1)` where `{Rm}` is a transresistance and `I(V1)` is the current through a voltage source `V1`, compare fig. 1.24 on page 33.

2. Overview of basic simulation commands.

The basic simulation commands used for simple circuits with DC voltages and currents, time varying signals and AC signals are 'DC op pnt', 'DC sweep', 'Transient', 'DC Transfer' and 'AC Analysis', compare page 16. In the Windows version of LTspice, these commands are inserted using 'Simulate → Edit Simulation Cmd' which opens a window with help functions for each of the simulation commands, see fig. 1.5 on page 18. In the Mac version, the window shown in fig. 1.5 is opened using 'Draft → SPICE Directives' and a right click in the field for typing in the SPICE Directive.
A very useful command in combination with the simulation commands is the LTspice directive '.step param'. By this command, parameter values can be swept over a specified range, so the command is very useful for examining the properties of a circuit when component values are varied over a specified range.

DC op pnt: This command computes DC currents and voltages in a circuit with capacitors treated as open circuits and inductors as short circuits.
Syntax: .op
In the Windows version of LTspice, the simulation results in an output file with all node voltages and device currents in the circuit. In the Mac version, the simulation results in a plot window where voltages and currents can be displayed. Voltages and currents are selected by pointing at nodes or components in the schematic, compare page 23. Alternatively, voltages and currents are available in the Spice Error Log file which is shown by 'View → Spice Error Log' or hotkey '⌘L'.
The simulation will run correctly only if DC voltages are defined in all nodes. If a node is connected only to capacitors or inputs to ideal voltage controlled sources, the node voltage is not defined and the simulation may lead to erroneous results.

DC sweep: This command computes DC currents and voltages over a range of values for one, two or three independent current sources or voltage sources.
Syntax: .dc <srcnam> <Vstart> <Vstop> <Vincr>
where <srcname> is the name of an independent current source or voltage source, <Vstart> is the start value and <Vstop> is the end value of the range of variation for the source. The step between each simulation is specified by <Vincr>.
The command is good for finding for instance the output voltage versus the input voltage for an amplifier.
The simulation results in a plot window where voltages and currents can be displayed versus the (first) independent source which is stepped. Voltages and currents are selected by pointing at nodes or components in the schematic, see page 23.

Transient: This command computes currents and voltages as a function of time in a circuit with one or more sources specified as time varying sources, see page 44.
Syntax: `.tran <Tstop>`
where `<Tstop>` is the end time for the simulation. The command is good for analyzing for instance charging or discharging of a capacitor or for finding time varying output signals versus time varying input signals in a circuit with amplifiers, capacitors and/or inductors.
The command requires that the voltage is defined in all nodes at time $t = 0$. If a node is connected only to capacitors or inputs to ideal voltage controlled sources, the node voltage is not defined and it may be necessary to define the voltage by a '.ic' command, see for instance fig. 2.13 on page 55.
In a similar way, the initial current in an inductor can be defined by a '.ic' command, see for instance fig. 2.15 on page 57.
A '.ic' command is also very useful when analyzing charging or discharging of capacitors or inductors, see for instance Example 2.5 on page 57.
The simulation results in a plot window where voltages and current are shown versus time. Voltages and currents are selected by pointing at nodes or components in the schematic, see page 23.
For the transient simulation, a modifier `[startup]` may be specified after `<Tstop>`. This modifier causes all DC sources in the circuit to ramp up linearly to the specified DC value during 20 µs. Do not use this modifier unless you are absolutely sure of what you are doing. Rather, use a '.ic' command as shown in Example 2.5 on page 57 or a time varying voltage or current, see page 44.

DC Transfer: This command computes the small signal input resistance, output resistance and transfer function from an (independent) input source to an output at a frequency of 0 Hz.
Syntax: `.tf <V(vout)> <srcname>`
where `<V(vout)>` is the voltage in the node labeled `vout` and `<srcname>` is the name of an independent voltage source or current source serving as the input signal.
Alternatively, the output may be specified as `I(<voltage source>)` where `<voltage source>` is an independent voltage source through which the output current `I` flows.
In the Windows version of LTspice, the simulation results in an output file showing the transfer function, the input resistance and the output resistance, see fig. 1.21 on page 32. In the Mac version, the simulation results in a plot window where the transfer function, the input resistance and the output resistance can be selected using the command 'Add Traces'.
Just like the '.op' command, the '.tf' requires all DC node voltages to be defined.

AC Analysis: This command computes 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.

Syntax: `.ac <oct, dec, lin> <Nsteps> <StartFreq> <EndFreq>`
where `<StartFreq>` and `<EndFreq>` denote the start and the end of the frequency range being simulated while `<oct, dec, lin>` and `<Nsteps>` determine the number of steps. Normally, you would want a logarithmic frequency scale in which case you would use `<oct>` (the default) or `<dec>` and specify a suitable number of points per octave or decade of frequency.
For this simulation, at least one independent source must be specified with an AC value, see for instance fig. 2.7 on page 50.
The simulation results in a plot window where voltages and current are shown versus frequency. Voltages and currents are selected by pointing at nodes or components in the schematic, see page 23.

Parameter Sweep: Combined with the simulation commands above, the LTspice command '.step param' may be used for analyzing the properties of a circuit when component values are varied over a specified range.
Syntax: `.step <Param> <Vstart> <Vstop> <Vincr>`
where `<Param>` is the name of the parameter to be varied, `<Vstart>` is the start value of the parameter, `<Vstop>` is the end value and `<Vincr>` is the step size of the parameter value between each simulation. When specifying a component value as a parameter, curly brackets are used to indicate the name of the parameter, for instance `{R1}`, compare fig. 1.13 on page 26.
Up to three parameters may be stepped in one simulation. The syntax shown above is the simplest form of the '.step' command. Alternative '.step' specifications are shown in the LTspice 'help' function.
A simulation with a parameter sweep results in a plot window. For a '.op' or '.tf' simulation, the x-axis is the (first) parameter being stepped. For '.dc', '.tran' and '.ac' simulations, the x-axis is the independent source, the time and the frequency, respectively, and separate traces are shown for each value of the parameter which is stepped. Voltages and currents are selected by pointing at nodes or components in the schematic, see page 23.

Appendix B –

BSIM transistor models for use in LTspice

The following pages give tables with BSIM transistor models adapted from (Carusone 2014) to be compatible with LTspice. The models have been modified to include the speed parameters `SN` and `SP` defined in Example 6.1 on page 180. For typical parameters, use `SN` = 0 and `SP` = 0. The files include a command '`.param SN=0 SP=0`' but it is inserted as a comment. To use only the typical models, just un-comment this command by deleting the '*' in the beginning of the command line. Alternatively, include a command specifying `SN` and `SP` in your schematic. The use of the speed parameters is described in Tutorial 6, Example 6.1 on page 180.

References

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

Generic BSIM3 model for 0.35 µm CMOS process with speed parameters `SN` and `SP` to define process variations.
To use without speed parameters and only with typical process parameters, un-comment the line `*.PARAM SN=0 SP=0` by deleting the `*`.

```
*BSIM3_035.lib
*Speed parameters SN and SP
*.PARAM SN=0 SP=0; Un-comment this command for use with only typical models
.MODEL NMOS-BSIM NMOS LEVEL = 49
+VERSION = 3.1 TNOM = 27 TOX = {7.8E-9/(1+SN/20)}
+XJ = 1E-07 NCH = 2.18E+17 VTH0 = {0.48-SN/10}
+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*(1+SN/20)**2} 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/(1+SN/20)} PB = 7.95E-01 MJ = 3.53E-01
+CJSW = {2.8e-10/(1+SN/20)} 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
*Speed parameter SP
+VERSION = 3.1 TNOM = 2.69E+01 TOX = {7.8E-9/(1+SP/20)}
+XJ = 1.00E-07 NCH = 8.44E+16 VTH0 = {-0.6+SP/10}
+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*(1+SP/20)**2} 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/(1+SP/20)} PB = 9.83E-01 MJ = 5.79E-01
+CJSW = {3.2e-10/(1+SP/20)} 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
```

Generic BSIM3 model for 0.18 µm CMOS process with speed parameters `SN` and `SP` to define process variations.
To use without speed parameters and only with typical process parameters, un-comment the line `*.PARAM SN=0 SP=0` by deleting the `*`.

```
*BSIM3_018.lib
*Speed parameters SN and SP
*.PARAM SN=0 SP=0; Un-comment this command for use with only typical models
.MODEL NMOS-BSIM NMOS
+ VERSION = 3
+ LEVEL = 49 NOIMOD = 1 TNOM = 2.70E+01
+ TOX = {4.1E-9/(1+SN/20)} XJ = 1.00E-07 NCH = 2.33E+17
+ VTH0 = {0.36-SN/10} K1 = 5.84E-01 K2 = 4.14E-03
+ K3 = 1.01E-03 K3B = 2.20E+00 W0 = 1.00E-07
+ NLX = 1.81E-07 DVT0W = 0.00E+00 DVT1W = 0.00E+00
+ DVT2W = 0.00E+00 DVT0 = 1.73E+00 DVT1 = 4.38E-01
+ DVT2 = -3.70E-04 U0 ={260*(1+SN/20)**2} UA = -1.38E-09
+ UB = 2.26E-18 UC = 5.46E-11 VSAT = 1.03E+05
+ A0 = 1.92E+00 AGS = 4.20E-01 B0 = -1.52E-09
+ B1 = -9.92E-08 KETA = -7.16E-03 A1 = 6.61E-04
+ A2 = 8.89E-01 RDSW = 1.12E+02 PRWG = 4.92E-01
+ PRWB = -2.02E-01 WR = 1.00E+00
+ WINT = 7.12E-09 LINT = 1.12E-08
+ DWG = -3.82E-09 DWB = 8.63E-09 VOFF = -8.82E-02
+ NFACTOR = 2.30E+00 CIT = 0.00E+00 CDSC = 2.40E-04
+ CDSCD = 0.00E+00 CDSCB = 0.00E+00 ETA0 = 3.13E-03
+ ETAB = 1.00E+00 DSUB = 2.25E-02 PCLM = 7.20E-01
+ PDIBLC1 = 2.15E-01 PDIBLC2 = 2.23E-03 PDIBLCB = 1.00E-01
+ DROUT = 8.01E-01 PSCBE1 = 5.44E+08 PSCBE2 =1.00E-03
+ PVAG = 1.00E-12 DELTA = 1.00E-02 RSH = 6.78E+00
+ MOBMOD = 1.00E+00 PRT = 0.00E+00 UTE = -1.50E+00
+ KT1 = -1.10E-01 KT1L = 0.00E+00 KT2 = 2.19E-02
+ UA1 = 4.28E-09 UB1 = -7.62E-18 UC1 = -5.57E-11
+ AT = 3.30E+04 WL = 0.00E+00 WLN = 1.00E+00
+ WW = 0.00E+00 WWN = 1.00E+00 WWL = 0.00E+00
+ LL = 0.00E+00 LLN = 1.00E+00 LW = 0.00E+00
+ LWN = 1.00E+00 LWL = 0.00E+00 CAPMOD = 2.00E+00
+ XPART = 5.00E-01 CGDO = 6.98E-10 CGSO = 7.03E-10
+ CGBO = 1.00E-12 CJ = {9.8e-4/(1+SN/20)} PB = 7.34E-01
+ MJ = 3.63E-01 CJSW = {2.4e-10/(1+SN/20)} PBSW = 4.71E-01
+ MJSW = 1.00E-01 CJSWG = 3.29E-10 PBSWG = 4.66E-01
+ MJSWG = 1.00E-01 CF = 0.00E+00 PVTH0 = -7.16E-04
+ PRDSW = -6.66E-01 PK2 = 5.92E-04 WKETA = 2.14E-04
+ LKETA = -1.51E-02 PU0 = 3.36E+00 PUA = -1.31E-11
+ PUB = 0.00E+00 PVSAT = 1.25E+03 PETA0 = 1.00E-04
+ PKETA = 6.45E-04 KF = 4.46E-29
```

```
.MODEL PMOS-BSIM PMOS
+ VERSION = 3.1
+ LEVEL = 49 NOIMOD = 1
+ TNOM = 2.70E+01 TOX = {4.1E-9/(1+SP/20)}  XJ = 1.00E-07
+ NCH = 4.12E+17 VTH0 = {-0.39+SP/10} K1 = 5.50E-01
+ K2 = 3.50E-02 K3 = 0.00E+00 K3B = 1.20E+01
+ W0 = 1.00E-06 NLX = 1.25E-07 DVT0W = 0.00E+00
+ DVT1W = 0.00E+00 DVT2W = 0.00E+00 DVT0 = 5.53E-01
+ DVT1 = 2.46E-01 DVT2 = 1.00E-01 U0 = {110*(1+SP/20)**2}
+ UA = 1.44E-09 UB = 2.29E-21 UC = -1.00E-10
+ VSAT = 1.95E+05 A0 = 1.72E+00 AGS = 3.80E-01
+ B0 = 5.87E-07 B1 = 1.44E-06 KETA = 2.21E-02
+ A1 = 4.66E-01 A2 = 3.00E-01 RDSW = 3.11E+02
+ PRWG = 5.00E-01 PRWB = 1.64E-02 WR = 1.00E+00
+ WINT = 0.00E+00 LINT = 2.00E-08
+ DWG = -3.49E-08 DWB = 1.22E-09
+ VOFF = -9.80E-02 NFACTOR = 2.00E+00 CIT = 0.00E+00
+ CDSC = 2.40E-04 CDSCD = 0.00E+00 CDSCB = 0.00E+00
+ ETA0 = 1.12E-03 ETAB = -4.79E-04 DSUB = 1.60E-03
+ PCLM = 1.50E+00 PDIBLC1 = 3.00E-02 PDIBLC2 = -1.01E-05
+ PDIBLCB = 1.00E-01 DROUT = 1.56E-03 PSCBE1 = 4.91E+09
+ PSCBE2 = 1.64E-09 PVAG = 3.48E+00 DELTA = 1.00E-02
+ RSH = 7.69E+00 MOBMOD = 1.00E+00 PRT = 0.00E+00
+ UTE = -1.49E+00 KT1 = -1.09E-01KT1L = 0.00E+00
+ KT2 = 2.18E-02 UA1 = 4.27E-09 UB1 = -7.68E-18
+ UC1 = -5.57E-11 AT = 3.31E+04 WL = 0.00E+00
+ WLN = 1.00E+00 WW = 0.00E+00 WWN = 1.00E+00
+ WWL = 0.00E+00 LL = 0.00E+00  LLN = 1.00E+00
+ LW = 0.00E+00 LWN = 1.00E+00  LWL = 0.00E+00
+ CAPMOD = 2.00E+00 XPART = 5.00E-01 CGDO = 6.88E-10
+ CGSO = 6.85E-10 CGBO = 1.00E-12 CJ = {1.2e-3/(1+SP/20)}
+ PB = 8.70E-01 MJ = 4.20E-01 CJSW = {2.4e-10/(1+SP/20)}
+ PBSW = 8.00E-01 MJSW = 3.57E-01 CJSWG = 4.24E-10
+ PBSWG = 8.00E-01 MJSWG = 3.56E-01 CF = 0.00E+00
+ PVTH0 = 3.53E-03 PRDSW = 1.02E+01 PK2 = 3.35E-03
+ WKETA = 3.52E-02 LKETA = -2.06E-03 PU0 = -2.19E+00
+ PUA = -7.63E-11 PUB = 9.91E-22 PVSAT = 5.00E+01
+ PKETA = -6.41E-03 KF = 1.29E-29 PETA0 = 7.31E-05
```

Generic BSIM4 model for 45 nm CMOS process with speed parameters `SN` and `SP` to define process variations.
To use without speed parameters and only with typical process parameters, un-comment the line `*.PARAM SN=0 SP=0` by deleting the `*`.

This file is derived from (Carusone 2014) subject to Apache License version 2.0, http://apache.org/licenses/LICENSE-2.0

```
*BSIM4_0045.lib
*Speed parameters SN and SP
*.PARAM SN=0 SP=0; Un-comment this command for use with only typical models
.model  NMOS-BSIM  nmos  level = 54
+version = 4.0    binunit = 1    paramchk= 1    mobmod  = 0
+capmod  = 2     igcmod  = 1    igbmod  = 1    geomod  = 1
+diomod  = 1     rdsmod  = 0    rbodymod= 1    rgatemod= 1
+permod  = 1     acnqsmod= 0    trnqsmod= 0
* parameters related to the technology node
+tnom = 27    epsrox = 3.9
+eta0 = 0.0049    nfactor = 2.1    wint = 5e-09
+cgso = 1.1e-10    cgdo = 1.1e-10    xl = -2e-08
* parameters customized by the user
+toxe = 1.75e-09    toxp = 1.1e-09    toxm = 1.75e-09    toxref = 1.75e-09
+lint = {3.75e-09*(1-abs(SN))+2.875e-09*uramp(-SN)+4.625e-09*uramp(SN)}
+vth0 = {0.471*(1-abs(SN))+0.5*uramp(-SN)+0.44*uramp(SN)}
+k1 = {0.53*(1-abs(SN))+0.555*uramp(-SN)+0.503*uramp(SN)}
+u0 = {0.04359*(1-abs(SN))+0.04163*uramp(-SN)+0.04581*uramp(SN)}
+vsat = 147390    rdsw = 155    ndep = 3.3e+18
+xj = {1.4e-08*(1-abs(SN))+1.54e-08*uramp(-SN)+1.26e-08*uramp(SN)}
* secondary parameters
+ll      = 0          wl      = 0          lln     = 1          wln     = 1
+lw      = 0          ww      = 0          lwn     = 1          wwn     = 1
+lwl     = 0          wwl     = 0          xpart   = 0
+k2      = 0.01       k3      = 0
+k3b     = 0          w0      = 2.5e-006   dvt0    = 1          dvt1    = 2
+dvt2    = -0.032     dvt0w   = 0          dvt1w   = 0          dvt2w   = 0
+dsub    = 0.1        minv    = 0.05       voffl   = 0          dvtp0   = 1.0e-009
+dvtp1   = 0.1        lpe0    = 0          lpeb    = 0
+ngate   = 2e+020     nsd     = 2e+020     phin    = 0
+cdsc    = 0.000      cdscb   = 0          cdscd   = 0          cit     = 0
+voff    = -0.13      etab    = 0
+vfb     = -0.55      ua      = 6e-010     ub      = 1.2e-018
+uc      = 0          a0      = 1.0        ags     = 1e-020
+a1      = 0          a2      = 1.0        b0      = 0          b1      = 0
+keta    = 0.04       dwg     = 0          dwb     = 0          pclm    = 0.04
+pdiblc1 = 0.001      pdiblc2 = 0.001      pdiblcb = -0.005     drout   = 0.5
+pvag    = 1e-020     delta   = 0.01       pscbe1  = 8.14e+008  pscbe2  = 1e-007
+fprout  = 0.2        pdits   = 0.08       pditsd  = 0.23       pditsl  = 2.3e+006
+rsh     = 5          rsw     = 85         rdw     = 85
+rdswmin = 0          rdwmin  = 0          rswmin  = 0          prwg    = 0
+prwb    = 6.8e-011   wr      = 1          alpha0  = 0.074      alpha1  = 0.005
+beta0   = 30         agidl   = 0.0002     bgidl   = 2.1e+009   cgidl   = 0.0002
+egidl   = 0.8
+aigbacc = 0.012      bigbacc = 0.0028     cigbacc = 0.002
+nigbacc = 1          aigbinv = 0.014      bigbinv = 0.004      cigbinv = 0.004
+eigbinv = 1.1        nigbinv = 3          aigc    = 0.012      bigc    = 0.0028
+cigc    = 0.002      aigsd   = 0.012      bigsd   = 0.0028     cigsd   = 0.002
+nigc    = 1          poxedge = 1          pigcd   = 1          ntox    = 1
+xrcrg1  = 12         xrcrg2  = 5
+cgbo    = 2.56e-011  cgdl    = 2.653e-10
+cgsl    = 2.653e-10  ckappas = 0.03       ckappad = 0.03       acde    = 1
+moin    = 15         noff    = 0.9        voffcv  = 0.02
+kt1     = -0.11      kt1l    = 0          kt2     = 0.022      ute     = -1.5
+ua1     = 4.31e-009  ub1     = 7.61e-018  uc1     = -5.6e-011  prt     = 0
+at      = 33000
+fnoimod = 1          tnoimod = 0
+jss     = 0.0001     jsws    = 1e-011     jswgs   = 1e-010     njs     = 1
+ijthsfwd= 0.01       ijthsrev= 0.001      bvs     = 10         xjbvs   = 1
+jsd     = 0.0001     jswd    = 1e-011     jswgd   = 1e-010     njd     = 1
+ijthdfwd= 0.01       ijthdrev= 0.001      bvd     = 10         xjbvd   = 1
+pbs     = 1          cjs     = 0.0005     mjs     = 0.5        pbsws   = 1
+cjsws   = 5e-010     mjsws   = 0.33       pbswgs  = 1          cjswgs  = 3e-010
+mjswgs  = 0.33       pbd     = 1          cjd     = 0.0005     mjd     = 0.5
+pbswd   = 1          cjswd   = 5e-010     mjswd   = 0.33       pbswgd  = 1
+cjswgd  = 5e-010     mjswgd  = 0.33       tpb     = 0.005      tcj     = 0.001
+tpbsw   = 0.005      tcjsw   = 0.001      tpbswg  = 0.005      tcjswg  = 0.001
+xtis    = 3          xtid    = 3
+dmcg    = 0e-006     dmci    = 0e-006     dmdg    = 0e-006     dmcgt   = 0e-007
+dwj     = 0.0e-008   xgw     = 0e-007     xgl     = 0e-008
+rshg    = 0.4        gbmin   = 1e-010     rbpb    = 5          rbpd    = 15
+rbps    = 15         rbdb    = 15         rbsb    = 15         ngcon   = 1
```

```
.model  PMOS-BSIM  pmos  level = 54
+version = 4.0    binunit = 1    paramchk= 1    mobmod  = 0
+capmod  = 2     igcmod  = 1    igbmod  = 1    geomod  = 1
+diomod  = 1     rdsmod  = 0    rbodymod= 1    rgatemod= 1
+permod  = 1     acnqsmod= 0    trnqsmod= 0
* parameters related to the technology node
+tnom = 27    epsrox = 3.9
+eta0 = 0.0049    nfactor = 2.1    wint = 5e-09
+cgso = 1.1e-10    cgdo = 1.1e-10    xl = -2e-08
* parameters customized by the user
+toxe = 1.85e-09    toxp = 1.1e-09    toxm = 1.85e-09    toxref = 1.85e-09
+lint = {3.75e-09*(1-abs(SP))+2.875e-09*uramp(-SP)+4.625e-09*uramp(SP)}
+vth0 ={-0.423*(1-abs(SP))-0.452*uramp(-SP)-0.392*uramp(SP)}
+k1 = {0.491*(1-abs(SP))+0.517*uramp(-SP)+0.465*uramp(SP)}
+u0 = {0.00432*(1-abs(SP))+0.00389*uramp(-SP)+0.00482*uramp(SP)}
+vsat = 70000  rdsw = 155    ndep = 2.54e+18
+xj = {1.4e-08*(1-abs(SP))+1.54e-08*uramp(-SP)+1.26e-08*uramp(SP)}
*secondary parameters
+ll      = 0          wl      = 0          lln     = 1          wln     = 1
+lw      = 0          ww      = 0          lwn     = 1          wwn     = 1
+lwl     = 0          wwl     = 0          xpart   = 0
+k2      = -0.01      k3      = 0
+k3b     = 0          w0      = 2.5e-006   dvt0    = 1          dvt1    = 2
+dvt2    = -0.032     dvt0w   = 0          dvt1w   = 0          dvt2w   = 0
+dsub    = 0.1        minv    = 0.05       voffl   = 0          dvtp0   = 1e-009
+dvtp1   = 0.05       lpe0    = 0          lpeb    = 0
+ngate   = 2e+020     nsd     = 2e+020     phin    = 0
+cdsc    = 0.000      cdscb   = 0          cdscd   = 0          cit     = 0
+voff    = -0.126     etab    = 0
+vfb     = 0.55       ua      = 2.0e-009   ub      = 0.5e-018
+uc      = 0          a0      = 1.0        ags     = 1e-020
+a1      = 0          a2      = 1          b0      = -1e-020    b1      = 0
+keta    = -0.047     dwg     = 0          dwb     = 0          pclm    = 0.12
+pdiblc1 = 0.001      pdiblc2 = 0.001      pdiblcb = 3.4e-008   drout   = 0.56
+pvag    = 1e-020     delta   = 0.01       pscbe1  = 8.14e+008  pscbe2  = 9.58e-007
+fprout  = 0.2        pdits   = 0.08       pditsd  = 0.23       pditsl  = 2.3e+006
+rsh     = 5          rsw     = 85         rdw     = 85
+rdswmin = 0          rdwmin  = 0          rswmin  = 0          prwg    = 3.22e-008
+prwb    = 6.8e-011   wr      = 1          alpha0  = 0.074      alpha1  = 0.005
+beta0   = 30         agidl   = 0.0002     bgidl   = 2.1e+009   cgidl   = 0.0002
+egidl   = 0.8
+aigbacc = 0.012      bigbacc = 0.0028     cigbacc = 0.002
+nigbacc = 1          aigbinv = 0.014      bigbinv = 0.004      cigbinv = 0.004
+eigbinv = 1.1        nigbinv = 3          aigc    = 0.69       bigc    = 0.0012
+cigc    = 0.0008     aigsd   = 0.0087     bigsd   = 0.0012     cigsd   = 0.0008
+nigc    = 1          poxedge = 1          pigcd   = 1          ntox    = 1
+xrcrg1  = 12         xrcrg2  = 5
+cgbo    = 2.56e-011  cgdl    = 2.653e-10
+cgsl    = 2.653e-10  ckappas = 0.03       ckappad = 0.03       acde    = 1
+moin    = 15         noff    = 0.9        voffcv  = 0.02
+kt1     = -0.11      kt1l    = 0          kt2     = 0.022      ute     = -1.5
+ua1     = 4.31e-009  ub1     = 7.61e-018  uc1     = -5.6e-011  prt     = 0
+at      = 33000
+fnoimod = 1          tnoimod = 0
+jss     = 0.0001     jsws    = 1e-011     jswgs   = 1e-010     njs     = 1
+ijthsfwd= 0.01       ijthsrev= 0.001      bvs     = 10         xjbvs   = 1
+jsd     = 0.0001     jswd    = 1e-011     jswgd   = 1e-010     njd     = 1
+ijthdfwd= 0.01       ijthdrev= 0.001      bvd     = 10         xjbvd   = 1
+pbs     = 1          cjs     = 0.0005     mjs     = 0.5        pbsws   = 1
+cjsws   = 5e-010     mjsws   = 0.33       pbswgs  = 1          cjswgs  = 3e-010
+mjswgs  = 0.33       pbd     = 1          cjd     = 0.0005     mjd     = 0.5
+pbswd   = 1          cjswd   = 5e-010     mjswd   = 0.33       pbswgd  = 1
+cjswgd  = 5e-010     mjswgd  = 0.33       tpb     = 0.005      tcj     = 0.001
+tpbsw   = 0.005      tcjsw   = 0.001      tpbswg  = 0.005      tcjswg  = 0.001
+xtis    = 3          xtid    = 3
+dmcg    = 0e-006     dmci    = 0e-006     dmdg    = 0e-006     dmcgt   = 0e-007
+dwj     = 0.0e-008   xgw     = 0e-007     xgl     = 0e-008
+rshg    = 0.4        gbmin   = 1e-010     rbpb    = 5          rbpd    = 15
+rbps    = 15         rbdb    = 15         rbsb    = 15         ngcon   = 1
```

Index