UE 4002 Labs

Revision Number :	1.1
Owner:	Kevin O'Sullivan
Document Date :	09 Feb. 2012
Document Number :	TP200207ALS

Contents

1 Introduction

These Labs cover the basics of analog CAD analog using Cadence design framework, Composer schematic entry editor and the Spectre simulator through Analog Design Environment. The goals of the labs are to familiarise the student with the Cadence design framework, with the Spectre simulator, and with the concepts of process models and simulation types through simulating some of the circuits from the UE4002 course. Comparison of results derived from hand calculations with those obtained from simulations is also highlighted.

Lab 1 and Lab2 go through the basic features of Cadence design framework. These labs show how to create a library, attach this library to a technology, use the schematic entry tool, how to make hierarchical design and how to perform basic DC, AC and transient simulations. These exercises are performed on a simple circuit from the course - a common-source NMOS stage with a resistive load.

Lab3 introduces parametric analysis for both DC and AC simulations. The vehicles used for this Lab are the common-source NMOS stage, and simple and cascoded current mirrors.

Lab4 deals with the simulation and frequency compensation of a negative feedback circuit. The stability of the circuit is assessed by the method of opening the loop, and also using the Cadence stability analysis tool. The impact of stability on transient behaviour is also investigated.

Some notes on notation:

In general accessing menus, dialogue windows and commands is indicated using forward slashes e.g. CIW/File/New/Library

Notation for various windows is given in Table 1

Notation	Window
CIW	Command Interpreter Window
LM	Library Manager
SC	Schematic Composer
ADE	Analog Design Environment
PA	Parametric Analysis
WW	Waveform Window

Table 1: Window Notation

Commands which are to be entered from the shell command line are denoted by the ‘greater than’ symbol.

>

In some cases menus shown in the figures may differ slightly from those in your set-up.

For each lab the relevant worksheet in the Appendix needs to be filled in and handed up along with simulation output plots as requested in the worksheet.

1.1 Revision History

Revision	Date	Description	Responsible
1.0	09 Feb. 2012	First version	Kevin O'Sullivan
1.1	16 Feb 2010	Updated to use Save State to Cellview Option	Kevin O'Sullivan

Table 2: Revision History

2 Lab 1 - Cadence set-up and Spectre simulations

2.1 Cadence design framework (df2) basics

This section shows how to set the necessary environment for Cadence and introduces the basic Cadence design framework concepts. This Cadence design framework is also known as 'integrated circuit front-to-back' tool (icfb) as it has the necessary functionality to design a circuit from start (schematic) to finish (layout).

2.1.1 Start-up

Log on to a workstation.

Open a terminal. (right-click on workspace and open Tools/Terminal...)

Enter the following command on the unix command-line:

> cadsetup

Observe the various choices and select '1' (i.e. type '1' and press 'Return') for Cadence, then '5' for Europractice AMS, then x to exit (or type 1,5,x and then press 'Return', see Figure 1).

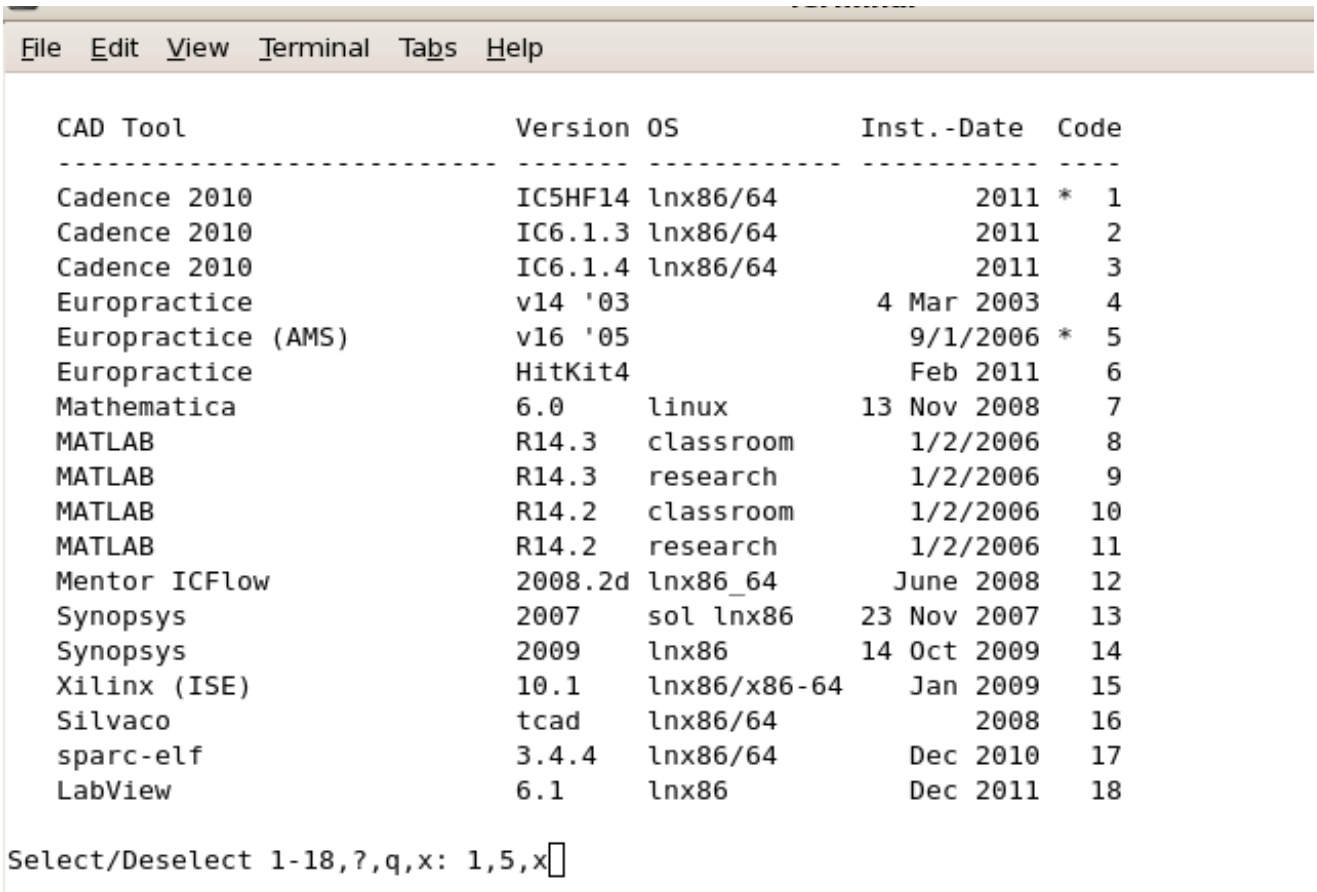


Figure 1: cadsetup options.

You will then need to logout and back on for the cadsetup changes to take effect

When you are logged back in, from the terminal command line make a sub-directory for the labs and descend into this sub-directory e.g.:

> mkdir UE4002_labs

> cd UE4002_labs

2.1.2 Opening Cadence icfb

Enter the following command:

> ams_cds -tech c35b4 -mode fb

This will start up Cadence icfb and give access to the Austria MicroSystems 0.35um technology.

OK the Select process Option window if it appears.

The first window which appears is the **CIW** (Command Interpreter Window)

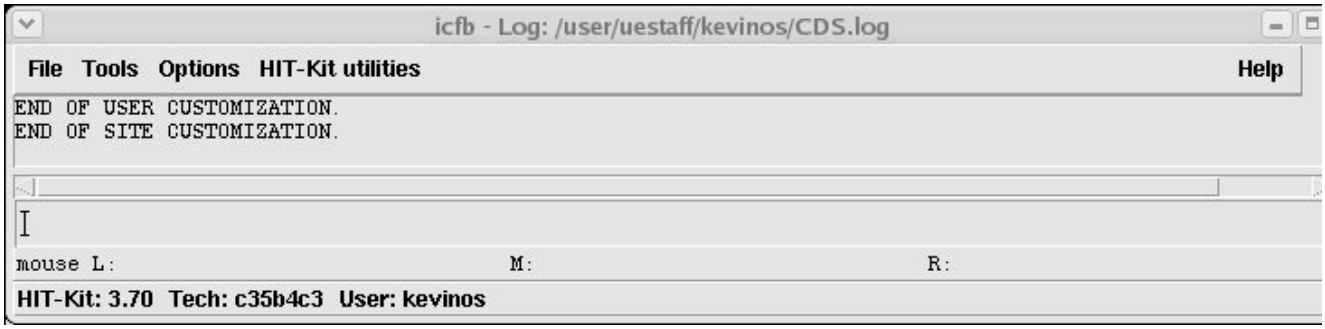


Figure 2: CIW

The CIW is the main entry into the various Cadence sub-programs.
The upper line contains the menu line. Left- or right- clicking on each of the menus will open submenus.

The text scroll window below the menu line contains output messages from Cadence.

The command input line accepts skill commands (skill is the internal Cadence programming language). Normally you will not need to use the command input line during the labs

2.1.3 Data structure

Cadence design data is organised in libraries, cells, and views.

2.1.4.1 Libraries

Libraries are collections of cells and views of these cells. Libraries can be classified in two types:

[1] **Reference** libraries. These contain tested cells with read-only status. These cells are reference components for the creation of new designs. Examples are:
PRIMLIB: technology reference library containing e.g. transistors from the AMS technology
analogLib: library containing Spectre simulation cells.
basic: library containing input, output pins etc.

[2] **Design** libraries. These contain the user's designs and can be edited.

2.1.5.2 Cells

A cell is the fundamental element, upon which each design is based. It is a logical unit, that contains several views of an object.

Examples of cells in a reference library are NMOS, or PMOS transistors.

A cell in a design library could be for example an inverter or an amplifier or a filter. These design cells can be comprised of cells from reference or design libraries, i.e. there is a hierarchy associated with each design. For example a filter could contain instantiations of an amplifier which in turn would contain instantiations of transistors from the PRIMLIB library.

2.1.6.3 Views

Views are representations of the cell at different levels of abstraction. Schematic, symbol and layout are examples of views. In these labs we will be concerned mainly with schematic and symbol views.

Some views are used for the graphic editor, but others are used for behavioural description (e.g vhdl, verilog-ams) of a block and so on.

2.1.7 Library Manager

The Library Manager (LM) is a central controlling item of the development environment. It provides access to the hierarchy of the libraries for opening and editing.

The Library Manager (Figure 3) can be activated by clicking on CIW/Tools/Library Manager...

The upper line contains the menu line. Left- or right- clicking on each of the menus will open submenus.

The main part of the window is split into three columns: Library, Cell, View.

A library, cell, or view can be selected by clicking with the left mouse button in the appropriate column.
When a library is selected the cells in that library appear in the cells column.
When a cell is selected the views of that cell appear in the View column.

Library cells can optionally be arranged in categories according to their purpose. Clicking on the 'Show Categories' button will make these visible in an additional Categories column.

Exercise 1: Browsing a library

Activate the Library Manager and open the library PRIMLIB and view the various components available.

Open the analogLib library and browse the various components and sources. If you click on Show Categories an extra column will be displayed showing the various categories of components.

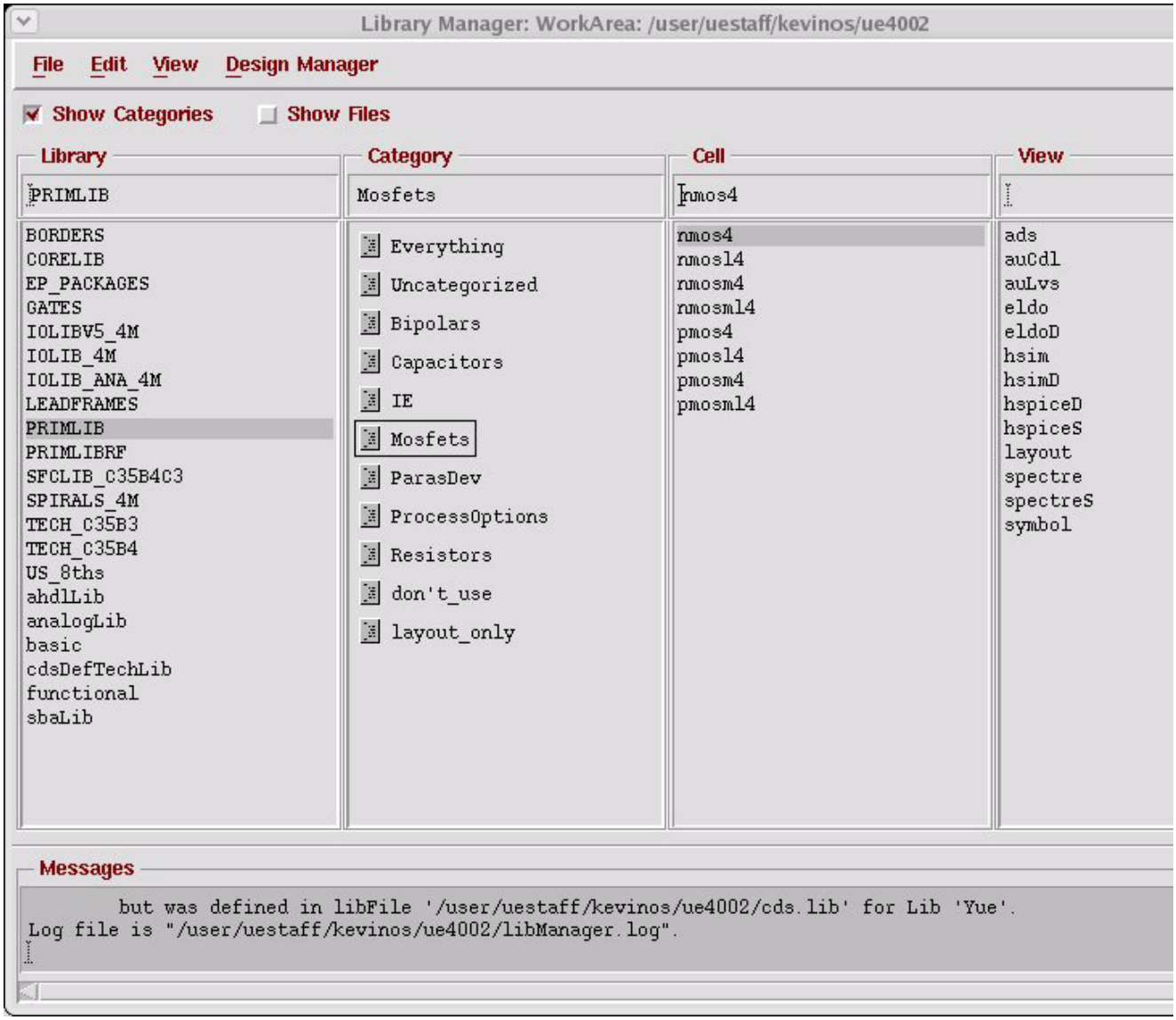


Figure 3: Library Manager

2.2 Creating a design

This section deals with creating a design, in this case a schematic view and a symbol view of a simple circuit. A design library will also need to be created to hold the cells and views.

2.2.1 Creating a library

First of all a design library needs to be generated, and this needs to be attached to a technology library (in this case TECH_C35B4).

Exercise 2: Creating a library

[1] Click on CIW/File/New/Library...

[2] In the 'New Library' window which appears (Figure 4), enter the name of the library in the Name field. Suggestion: Lab1Lib.
In the Technology file section of the window check that the option 'Attach to an existing techfile' is selected.
Check also that the directory field points to the directory you created for the labs i.e. ~/UE4002_labs
Click OK

[3] In the 'Attach Design Library to Technology' window which appears (Figure 5), set the technology library to TECH_C35B4.
Click OK.

The CIW scroll window should report that the library has been successfully attached to the technology library.

If you right click the library in the LM, and select Properties in the sub-menu, the 'Library Property Window' will appear (Figure 6), and show the library settings e.g. the unix path to the library, the technology to which the library is attached etc.

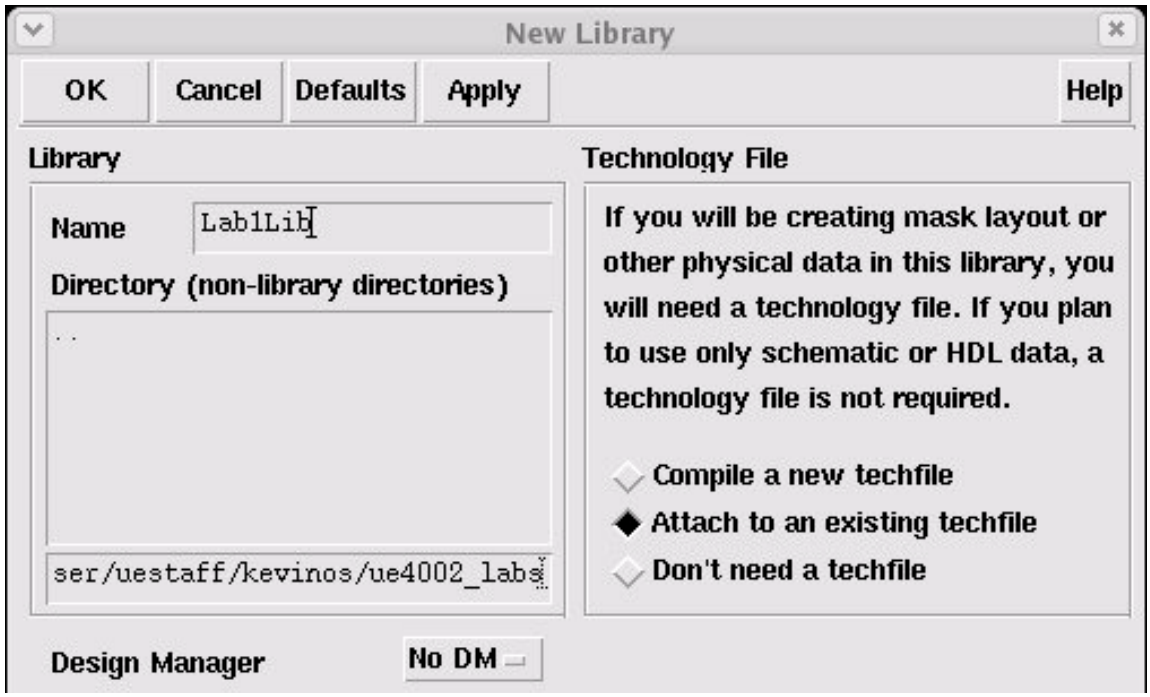


Figure 4: New Library menu

Figure 5: Attach Design Library to Technology File menu

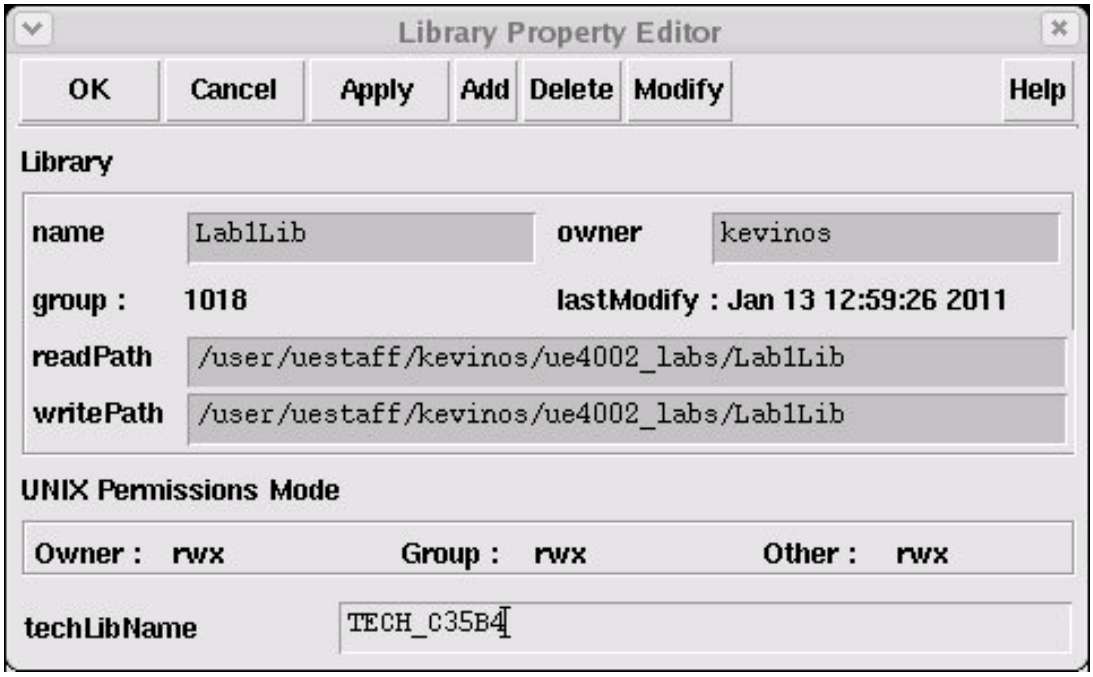


Figure 6: Library Property Editor

2.2.2 Creating a schematic cellview

Schematic entry is the basis of CAD for analog circuits. The following exercise is to generate a schematic cellview for an NMOS common-source stage with a resistive load.

Exercise 3: Creating a schematic cellview

[1] Click on CIW/File/New/Cellview

[2] In the 'Create New File' window which appears (Figure 7):
Set the library name to the library you have generated.
Set the cell name to CS_nmos_resload.
Set the View Name to Composer Schematic.

[3] Click on OK. The schematic composer window will appear.

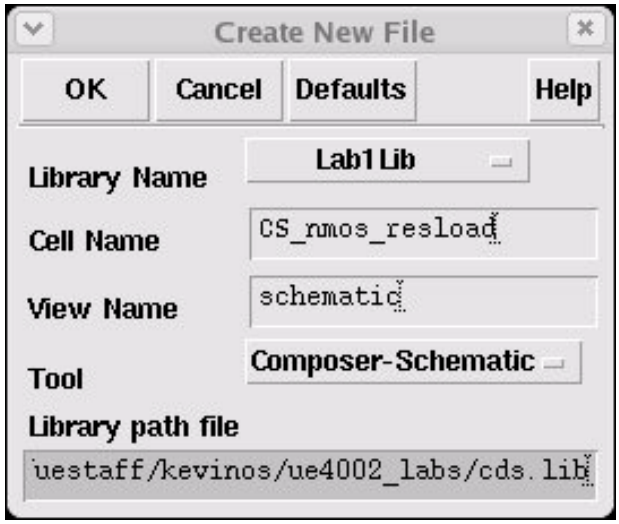


Figure 7: Create New File window

2.2.3 Schematics Composer Window (SC)

The schematics window essentially consists of the area in which a design schematic is edited. Along the top is a menu line. Left- or right- clicking on each of the menus will open sub-menus. Along the left-hand side (LHS) are buttons representing the most important design functions which can be executed by left-clicking the appropriate button.

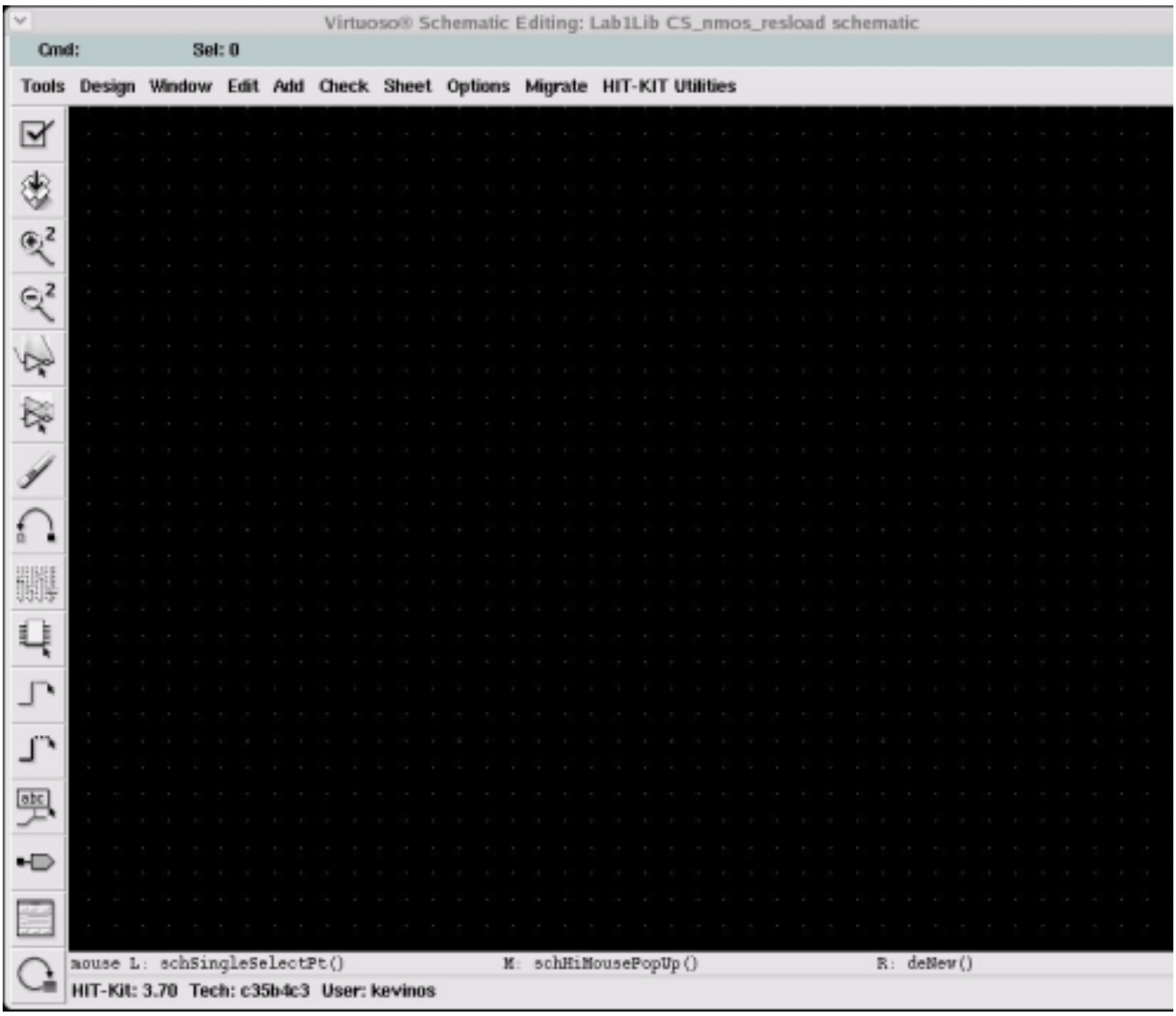


Figure 8: Schematic Composer window

In general schematic edit functions can be executed from the menus and sub-menus on top, the buttons on the left-hand side or using keyboard short-cuts. When available, keyboard short-cuts are indicated in the sub-menus.

The most important edit functions are described below:

[i] **Instantiating objects:**
Select SC/Add/Instance or click the Instance button on the LHS or use the short cut i.
In the Add instance Window which appears, click on the Browse button to open a LM and select the object you want to insert into your design.

Exercise 4: Entering a schematic

[1] Instantiate the symbol view of the nmos4 transistor from the PRIMLIB library using any of the means described above. An Add Instance window will appear. You can type in Library, Cell and View as shown in Figure 9, or select Browse and navigate to the nmos4 symbol view using the Add Instance Library Manager which then appears (Figure 10).
Instantiate the NMOS in the SC window by left-clicking

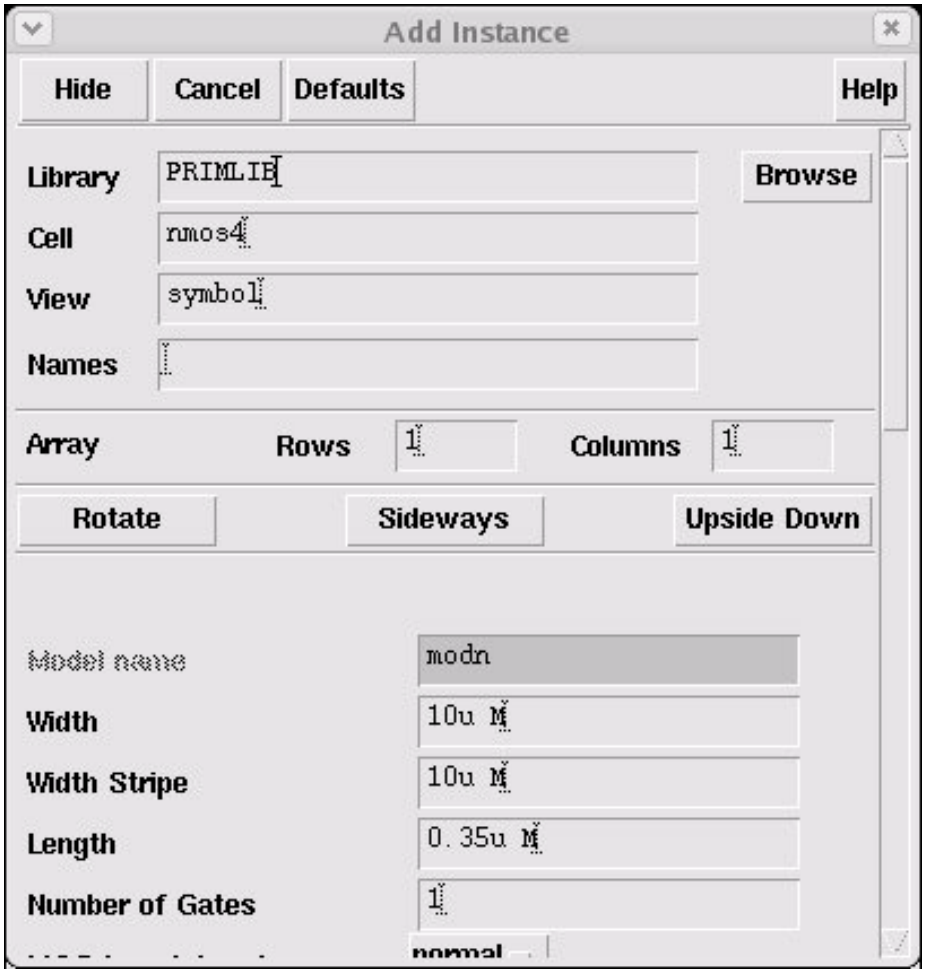


Figure 9: Add Instance window

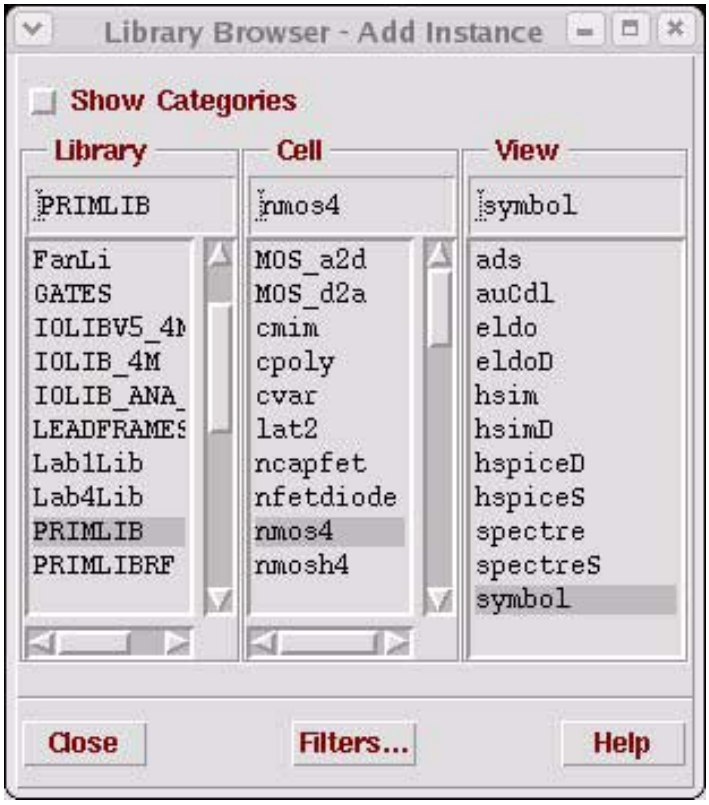


Figure 10: Add Instance Library Browser

[ii] **Zooming**

There are two buttons available on the LHS for zooming in and zooming out.
You can also draw a rectangle with the right mouse button for increasing the zoom factor
You can also use the SC/Window/Zoom menu option
To fit the design to the design window click on SC/Window/Fit or use the shortcut 'f'

[2] Instantiate a resistor (symbol view) from the analogLib library (cell: res, view:symbol) and place it above the NMOS transistor.

[iii] **Selecting Objects**

To select a single object move the cursor over the object and left-click to select it
You can also draw a rectangle with the left mouse button to select an object or group of objects.
<ctrl>a will select all objects.
Note: selecting a new object will unselect previously selected objects.
To select additional objects without unselecting previously selected objects use <ctrl> left-click.
Note: the status line above the menu line indicates how many objects are selected.

[iv] UnSelecting Objects

Left-clicking an empty area will unselect all objects
Left-clicking an object will select that object and unselect previously selected objects.
You can also draw a rectangle with the <ctrl>left mouse button to unselect select an object or group of objects

[v] Moving objects.
To move an object first select it.
Then use SC/Edit/Move or the short-cut M to activate the move command.
Left click in the SC window to provide the reference point for the move.
Left-click again to drop the moved object.

[vi] Stretching objects
Stretching is moving but retaining the connectivity.
To stretch an object first select it.
Then use SC/Edit/Stretch or the short-cut m to activate the stretch command.
Left click in the SC window to provide the reference point for the stretch.
Left-click again to drop the stretched object.

[vii] **Copying objects**.
To copy an object first select it.
Then use SC/Edit/Copy or use the short-cut c to activate the copy command.
Left click in the SC window to provide the reference point for the copy.
Left-click again to drop the copied object.

[viii]**Creating wires**
Use SC/Add/wire narrow or use the short-cut w or use the wire button on the LHS
Left-click to start the wire.
A second left-click will define a corner of the wire.
A double left-click will end the wire.
The Esc button will also end the wire.
A wire will also end if it is routed to a device terminal or onto another wire

[3] Create a wire from the drain of the NMOS to one terminal of the resistor.

[ix] **Creating pins**
Use SC/Add/pin or short-cut p or the Pin button on the LHS
Fill in the name of the pin in the Pin Name field, and set the direction.

[4] Create four pins: in (direction input), out (direction output), vdd (direction inputOutput), vss (direction inputOutput).

[x] **Adding net names (wire names)**
Cadence will assign arbitrary net names to nodes that are not explicitly named e.g. net105.
Nets that are attached to pins are assigned the name of the pin.
To assign a net name (wire name) to a node
Use SC/Add/ Wire Name or the shortcut l or the Wire Name button on the LHS. Then fill in the net name in the menu and left-click on the wire to be named.

[xi] **Deleting objects**
Select the objects to be deleted.
Use SC/Edit/Delete or the ‘delete’ key short-cut or the delete button on the LHS.

[xii] **Undoing actions**
Use SC/Edit/undo or the u short-cut

[xiii]**Exiting a command**
Use the Escape key to break-off a command

[5] Complete the schematic of the NMOS common-source stage with resistive load (see Figure 12).

[xiv]**Modifying Object Properties**

Select the object.
Use SC/Edit/Properties/Objects or the q short-cut or the Properties button on the LHS.
An Edit Object Properties menu will appear with various fields to modify the object properties.
OK the form to effect the property changes. (Apply will also effect the changes, but the menu will remain).

[6] Modify the NMOS transistor to give it a width of 10 and a length of 1.
Modify the resistor to give it a value of 10 k.

[xv] **Saving the design**

Use SC/Design/Check and Save or the X short-cut or the Check and Save button on the LHS.

With this option, the syntax of the schematic is checked. If errors or warnings are found, then these will be marked. The CIW scroll window will give explanations for errors/warnings.
You can also investigate the meaning and location of these markers with SC/Check/Markers/Find Markers and SC/Check/Markers/Explain Markers.

[7] Check and Save the schematic. If there are errors or warnings these should be fixed and the design again checked and saved.

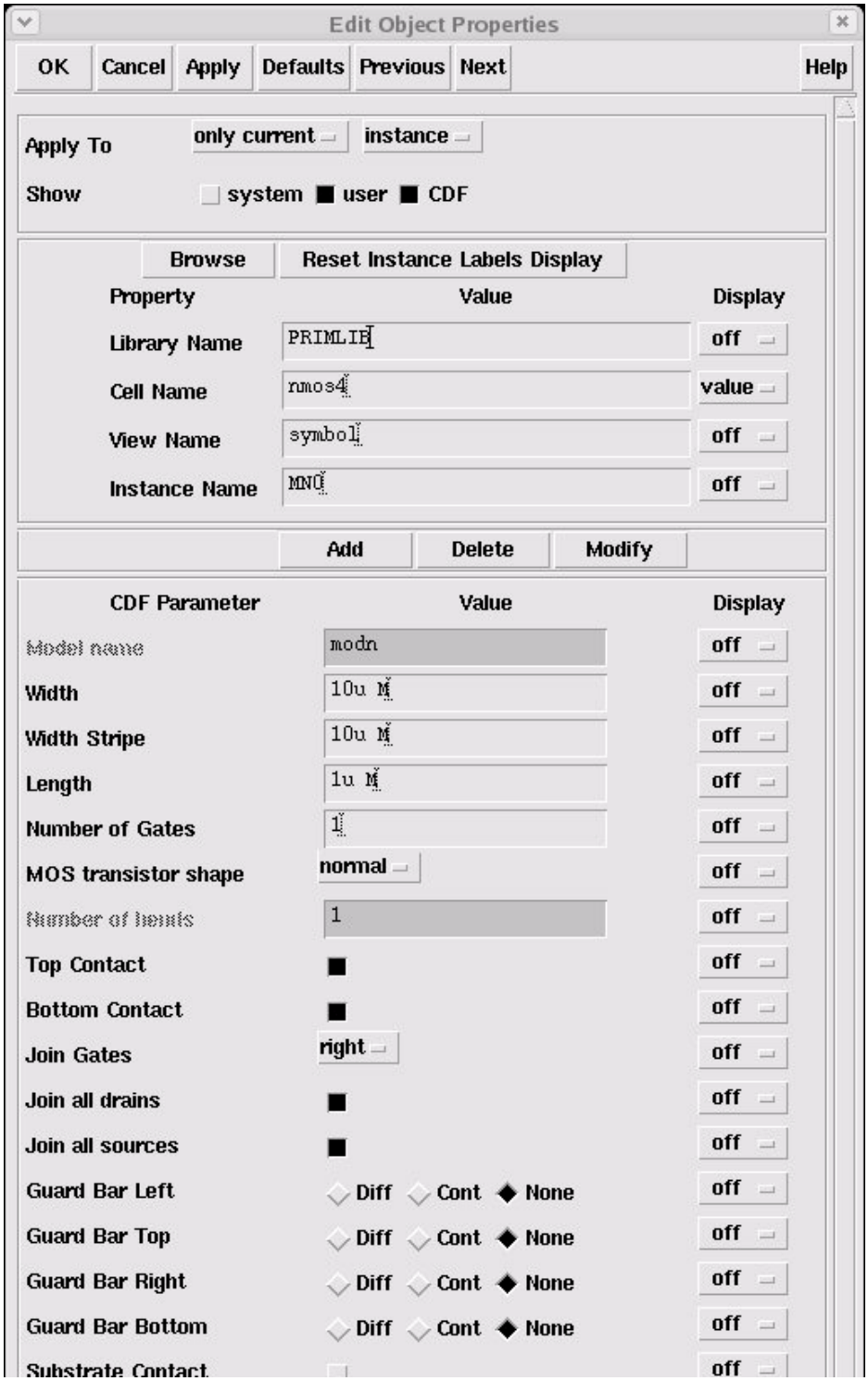


Figure 11: Edit Object Properties window.

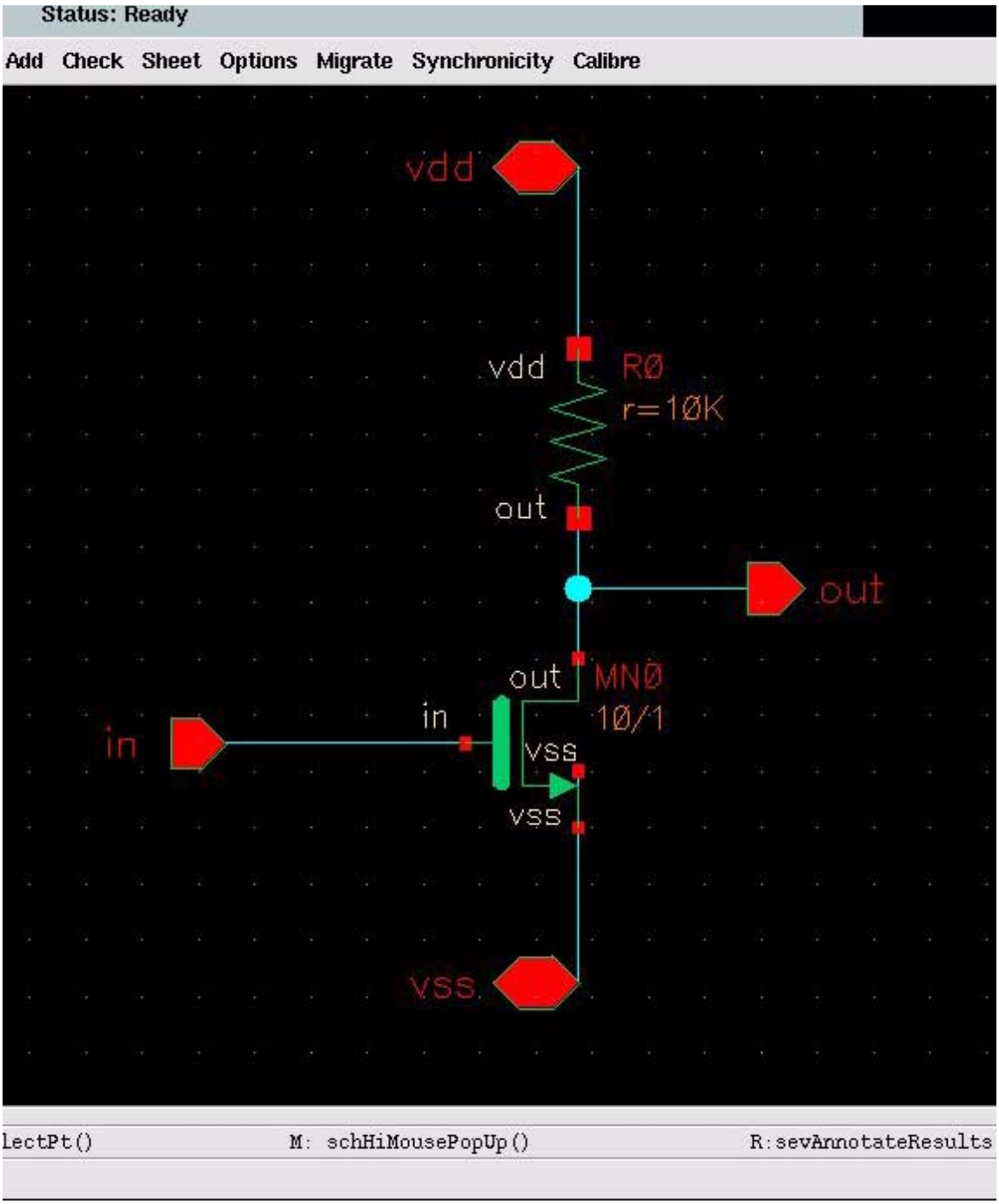


Figure 12: CS_nmos_resload schematic.

2.2.4 Creating a symbol cellview

In order to re-use a design in another design i.e. instantiate it at a higher hierarchical level you need to generate a symbol view of the schematic design.

Exercise 5: Creating a symbol cellview

[1] Open the schematic view of the cell CS_nmos_resload.

[2] Select SC/Design/Create Cellview/from Cellview... and click on OK

[3] The Cellview from Cellview window (Figure 13) appears, which should indicate the library and schematic name and that you are generating a symbol from a schematic.
OK this form.

[4] The Symbol Generation Options form (Figure 14) appears. This allows pin positions to be changed. Make any changes you require (not necessary but you may want the vss at the bottom) and OK the form.

[5] The Symbol View Editing window opens. The symbol can be modified here if wished: the functionality of this edit window is similar to the schematic composer window (an example symbol is shown in Figure 15).

[6] Save the symbol using the Design/Check and Save menu option

[7] Close the window using the Window/Close menu option.

Note: To re-open the design, or any existing cellview, right click on the view in the LM, and execute Open...

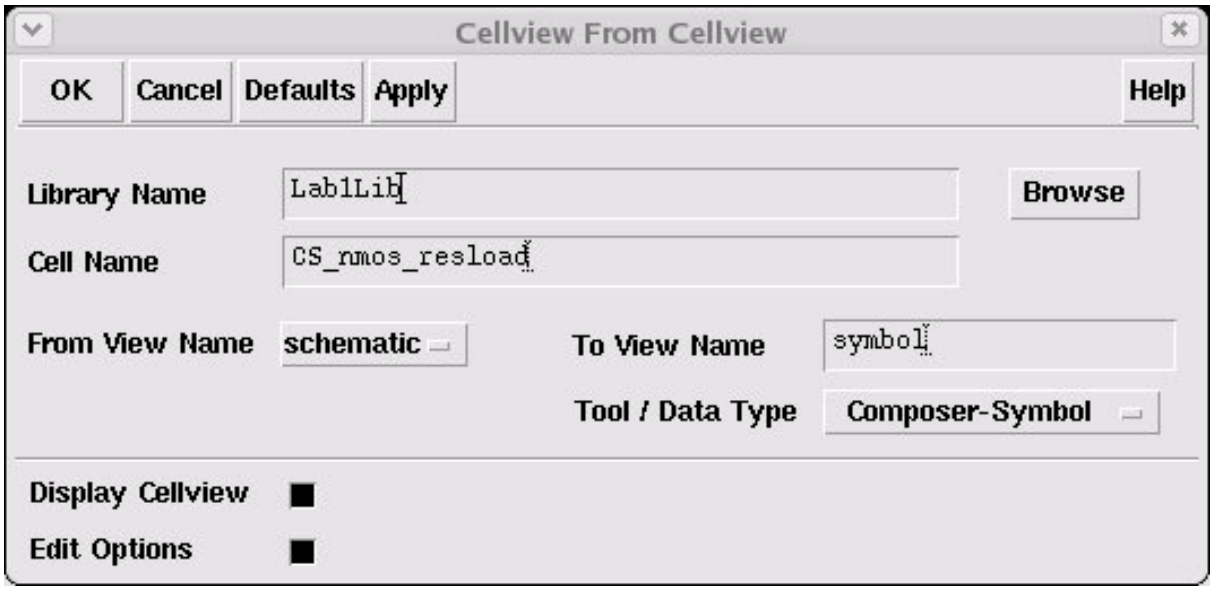


Figure 13: Cellview From Cellview window.

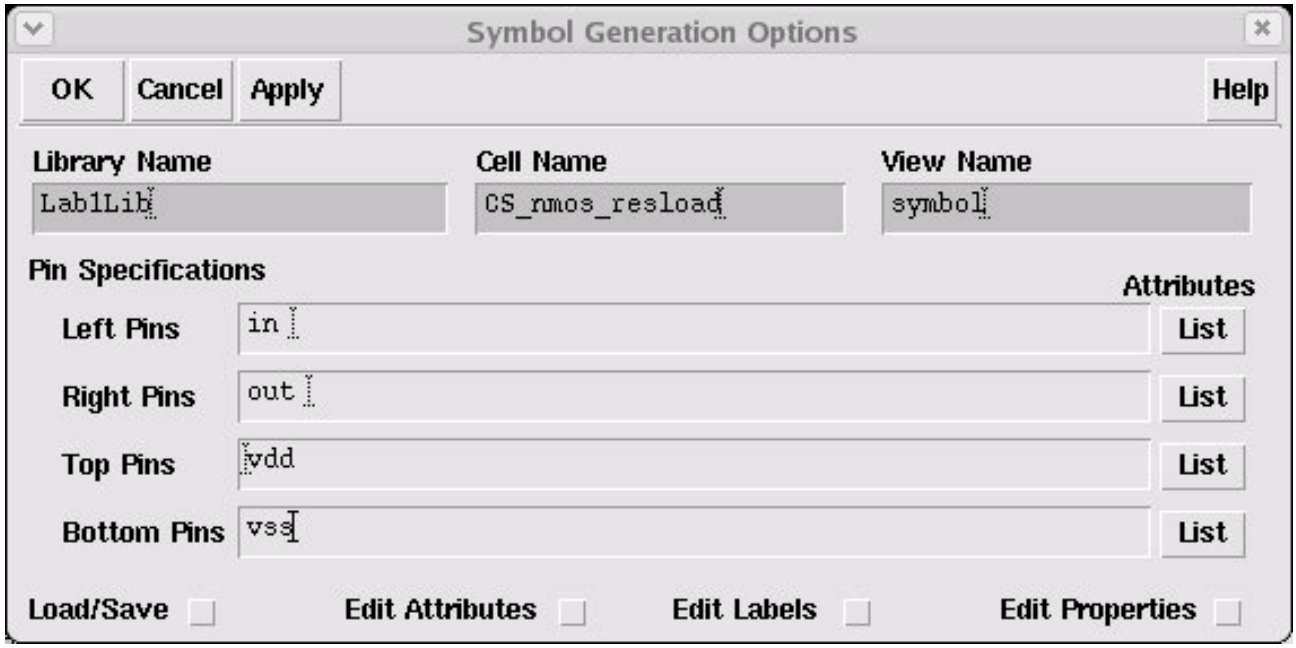


Figure 14: Symbol Generations Options window.

2.3 Creating a testbench and running simulations

The goal of this section is to create a testbench and run simulations on the design generated in Section 2.2.

2.3.1 Creating a testbench

First you need to generate a testbench. This testbench is a schematic view with an instantiation of the design to be tested, and various test stimuli.

Exercise 6: Generating a testbench

[1] Generate a schematic cellview in your work library (using CIW/File/New/Cellview) entitled test_CS_nmos_resload.

[2] Edit this schematic as follows: (see Figure 15).
Instantiate the symbol view of CS_nmos_resload.
Instantiate a dc voltage source from library analogLib (cell vdc, view symbol).
Modify the object properties of the voltage source and set the DC Voltage field to 3.3
Connect the positive terminal of the voltage source to the vdd terminal on the CS_nmos_resload symbol.
Instantiate a ground symbol from library analogLib (cell gnd, view symbol).
Connect the negative terminal of the voltage source to the ground symbol.
Instantiate a second voltage source, set the DC Voltage to 1, and connect it between the in terminal on CS_nmos_resload and ground. (Note you can also copy and modify the existing source).
Create an output pin out and connect it to the out terminal on CS_nmos_resload.
Instantiate a capacitor (cell:cap) from the analogLib library. Set the value of the capacitor to 1pF. Connect the capacitor between the output node and ground.
Label the vdd wire vdd and the input wire in using SC/Add/wire name or the l shortcut
Check and save the design.

2.3.2 Navigating hierarchy

You have now created a hierarchical design. You can descend into a lower level of hierarchy e.g. to view simulation results or make modifications to designs.

Exercise 7: Navigating hierarchy

[1] In the test_CS_nmos_resload schematic, select the CS_nmos_resload symbol.

[2] Descend into the CS_nmos_resload schematic using SC/Design/Hierarchy/Descend Read or the e short-cut and OKing the form which appears.

[3] You are now in the CS_nmos_resload schematic view in read-only mode. (You can make it editable using SC/Design/Make Editable).

[4] To return to the testbench level use SC/Design/Hierarchy/Return or the <ctrl>-e short-cut.

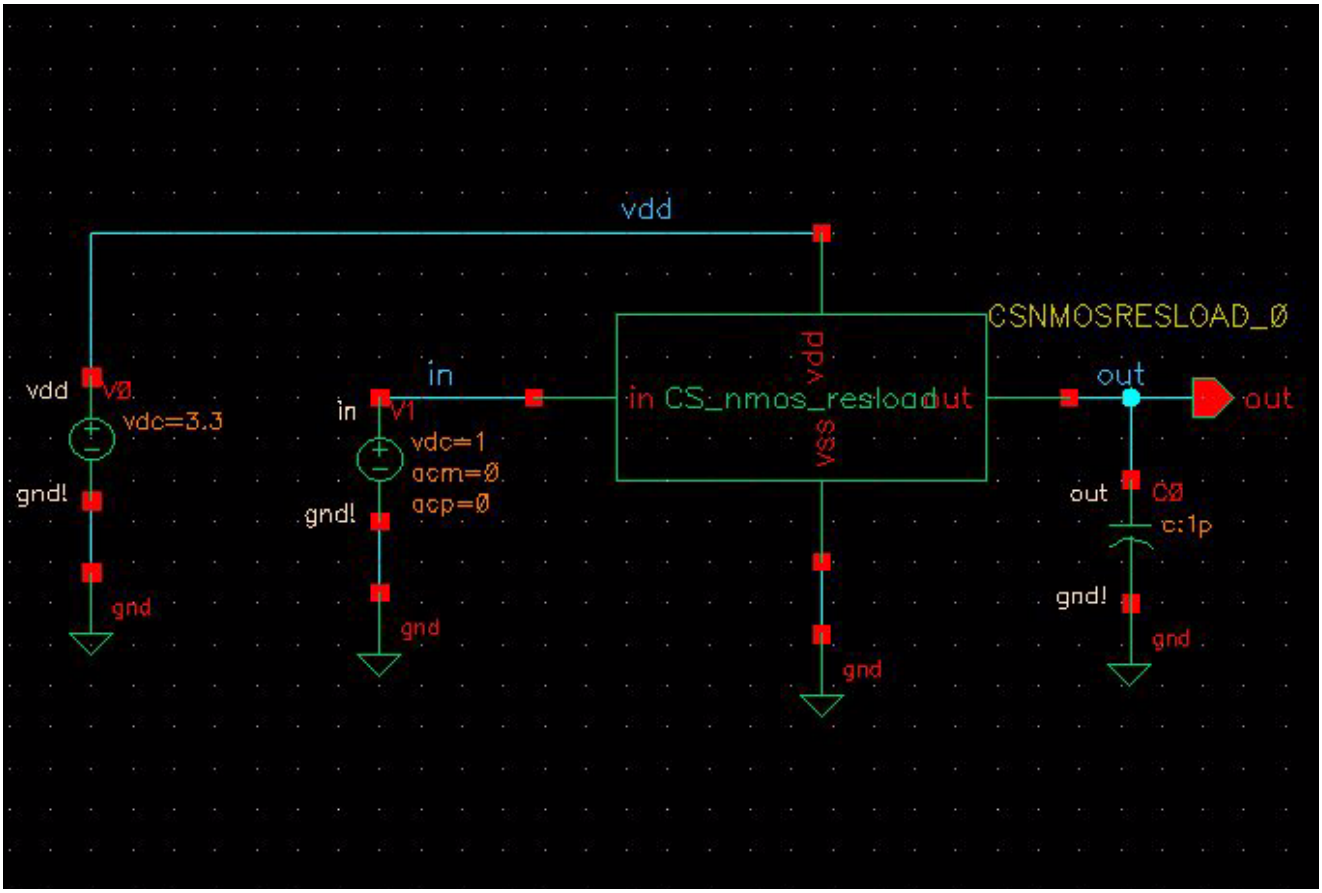


Figure 15: test_CS_nmos_resload schematic

2.3.3 Running a DC simulation

The first analysis to run is a DC analysis. The simulator will try to solve the voltage at each node and the current through each branch based on the DC biasing we have set up. Once the simulation has run we can print out the voltages at each node and look at the operating points of each component.

Exercise 8: Running a DC simulation

[1] Open the analog simulator environment with SC/Tools/Analog Environment. The Analog Design Environment (ADE) window will appear (Figure 16)

[2] In the menu line check ADE/Setup/Simulator that Spectre is set as simulator. In ADE/Setup/Model libraries check that the models for this technology are set.

[3] Select ADE/Analyses/Choose..

[4] In the Choosing Analyses window (Figure 17) click on DC.
Then turn on the buttons Save DC Operating Point and Enabled.
OK this form

[5] Select ADE/Simulation/Netlist and Run.
An output log window(Figure 18) appears, giving information on the progress of the simulation.
The Status line of the ADE window indicates Simulating.
When the simulation is complete, the Status line indicates Ready.

[6] Check the dc voltages on each node. To do this use ADE/Results/DC Node Voltages
Then left-click on each node and press ESC.
The voltages will appear in the Results Display Window which appears.
Alternatively you can use ADE/Results/Annotate?DC Node Voltages, and the DC voltages will be annotated to the nodes in the schematic.

[7] Check the DC operating point of the NMOS transistor with ADE/Results/Print/DC Operating Points. You will need to navigate into CS_nmos_resload and select the transistor. The output appears in the Results Display Window.
***Refer to the Exercise worksheet in Appendix A and fill in the required values in Table 3 and Table 4**.*
Press ESC when you need to exit the Show Operating Points mode.

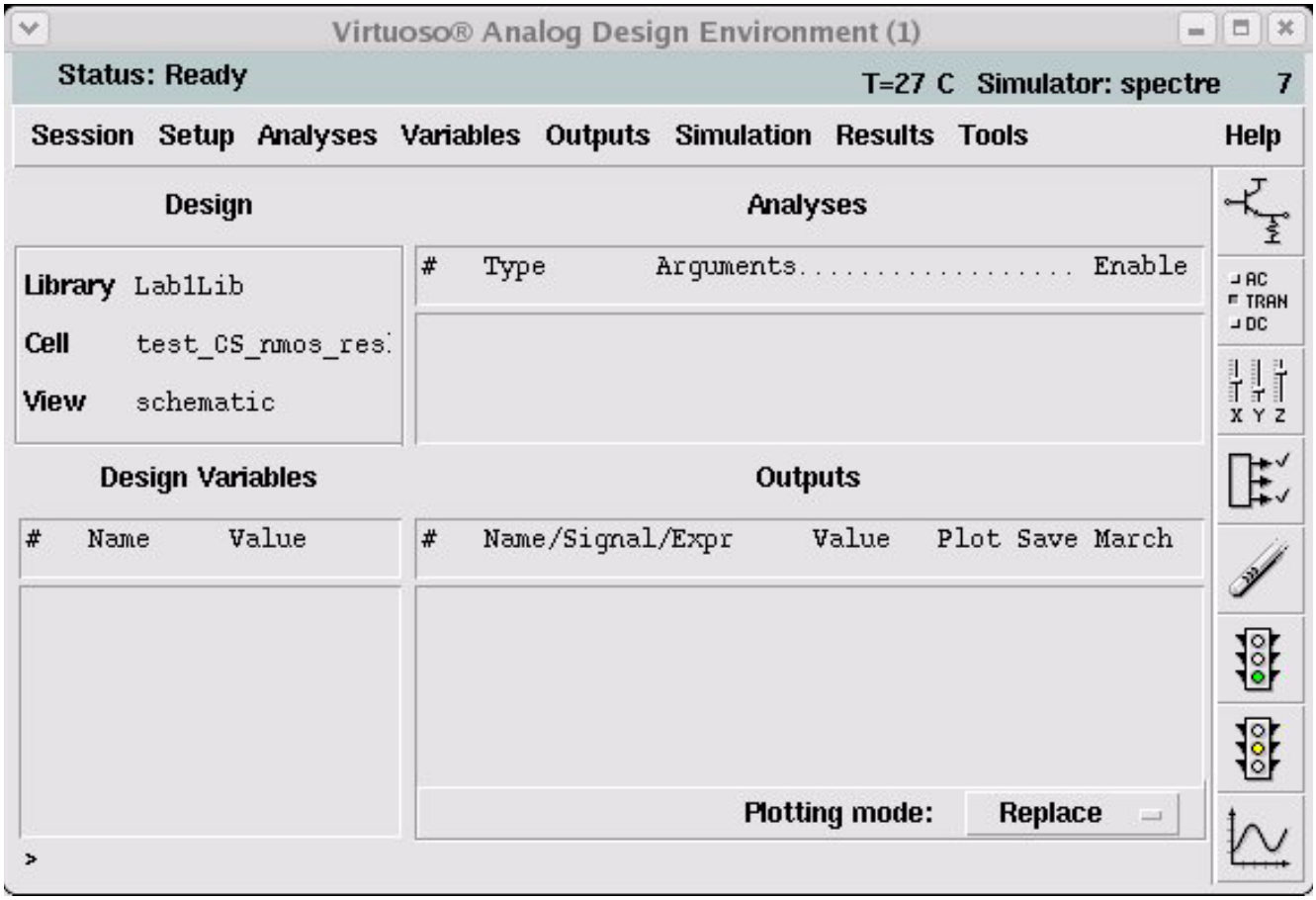


Figure 16: Analog Design Environment Window

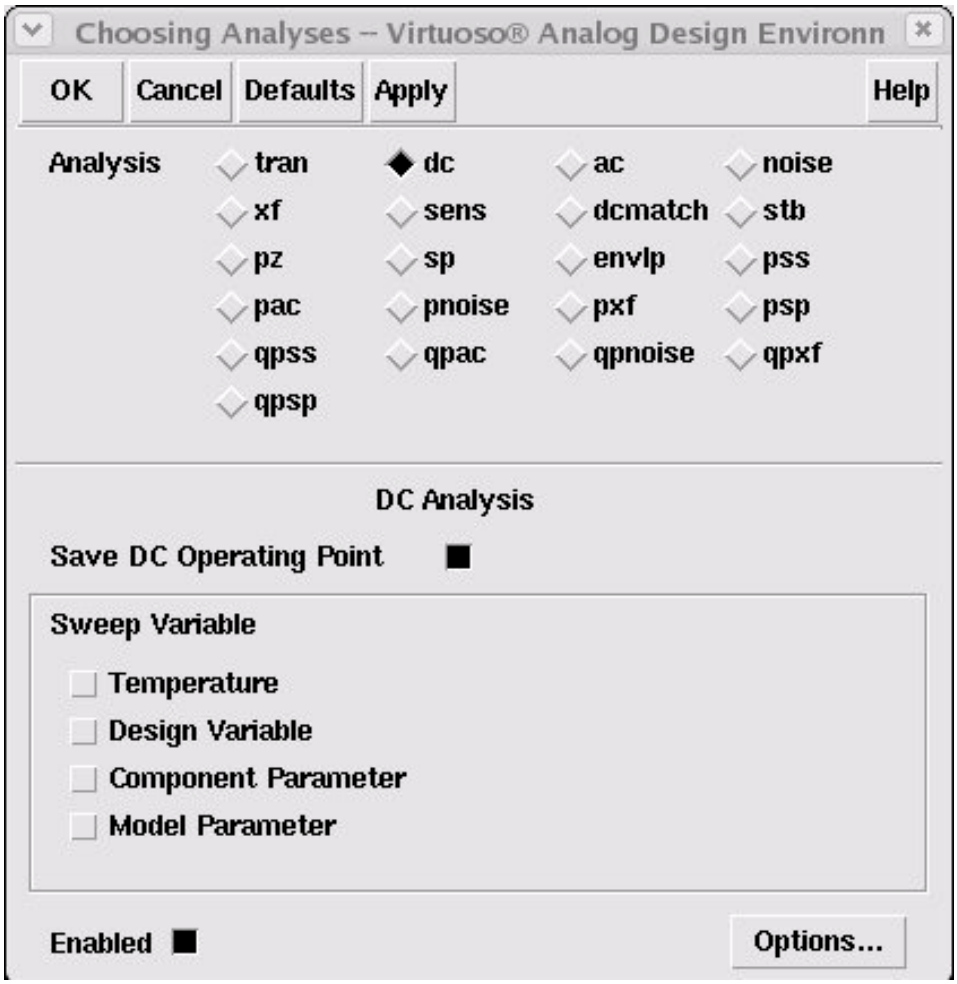


Figure 17: DC Analysis window.

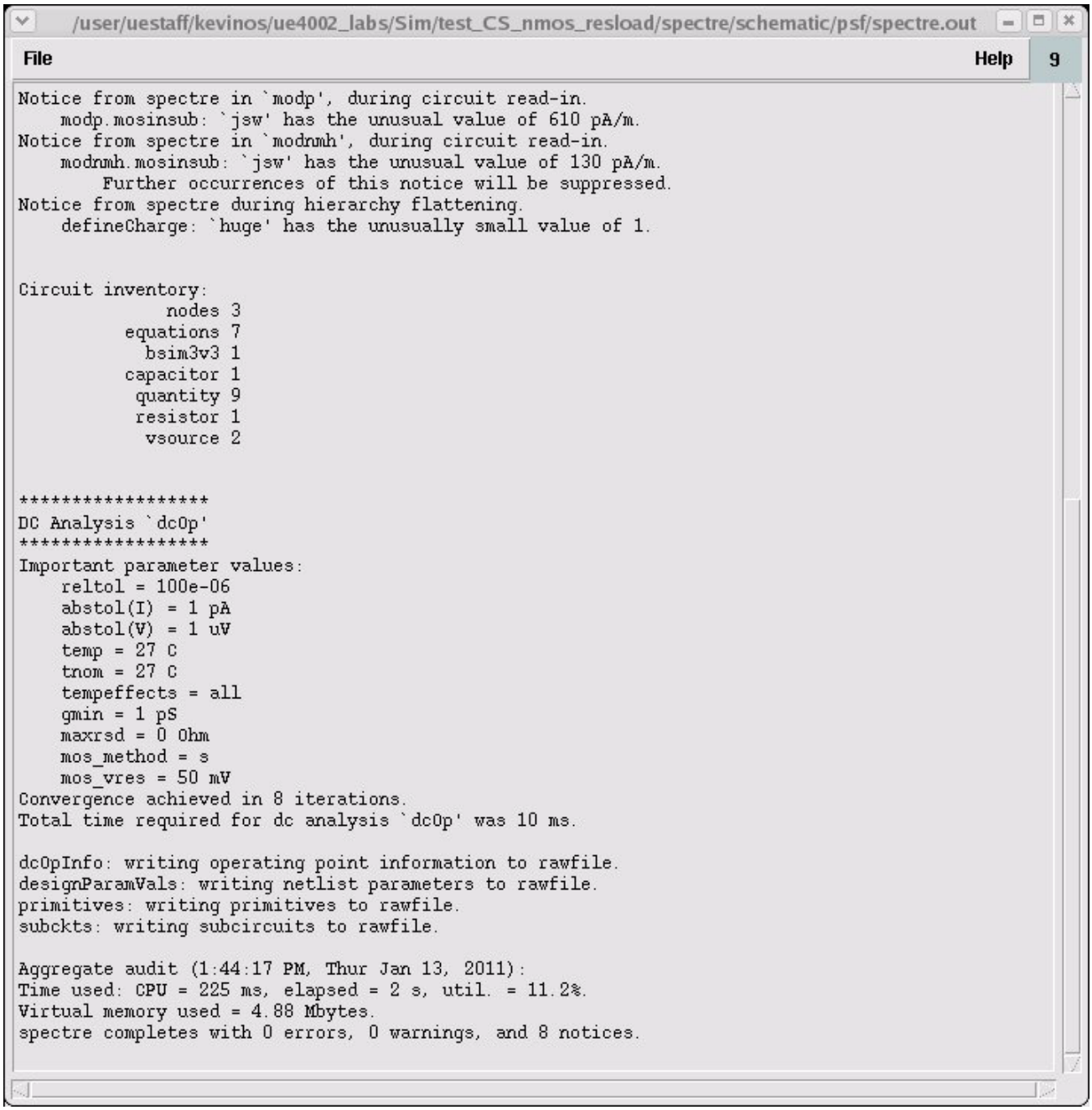


Figure 18: Spectre output log window

2.3.4 Running an AC simulation

An AC simulation is a small-signal analysis, using a circuit linearised around the dc operating point. (This is an important point - it does not matter what AC magnitude you set for the input stimulus, it will not affect the gain/ phase results). The AC simulation will usually involve a sweep over frequency to get the high-frequency response of the circuit.

Exercise 9: Running an AC simulation

[1] Instantiate a vsin element from the analogLib library and place it in series with the dc input voltage source (see Figure 19)

[2] Modify the object properties of the vsin component and set the AC magnitude to 1 and the AC Phase to 0. Save the schematic using SC/Design/ Check and Save or the X short cut.. (Note that Spectre won't run unless the schematic has been checked and saved).

[3] Select ADE/Analyses/Choose.. In the Choosing Analyses window click on AC and set the options as shown in Figure 20 (Note the Stop frequency is 1G).

[4] Select ADE/Simulation/Netlist and Run.

[5] Select ADE/Results/Direct Plot/ AC Magnitude and Phase, left-click on the output node and press Esc.

[6] From the AC plot (Figure 21) check that the low-frequency gain is in line with that predicted from the DC analysis.
Check also that the pole frequency is in line with expectations.
Note: The graphical waveform analysis tool is called 'Wavescan', WS for short.
Use WS/Trace/Trace Cursor or the short-cut c which enables you to read values from the graph.
Use WS/Trace/Delta Cursor or the short-cut d to measure differences on the graph.
Use WS/Axis/Strips to toggle between single and separate plots (separate plots shown in Figure 21).
Use WS/Graph/Color Schemes/Default tio set the graph background colour to white if preferred.

[7] ***Refer to the Exercise worksheet in Appendix A and fill in the required values in Table 5.***

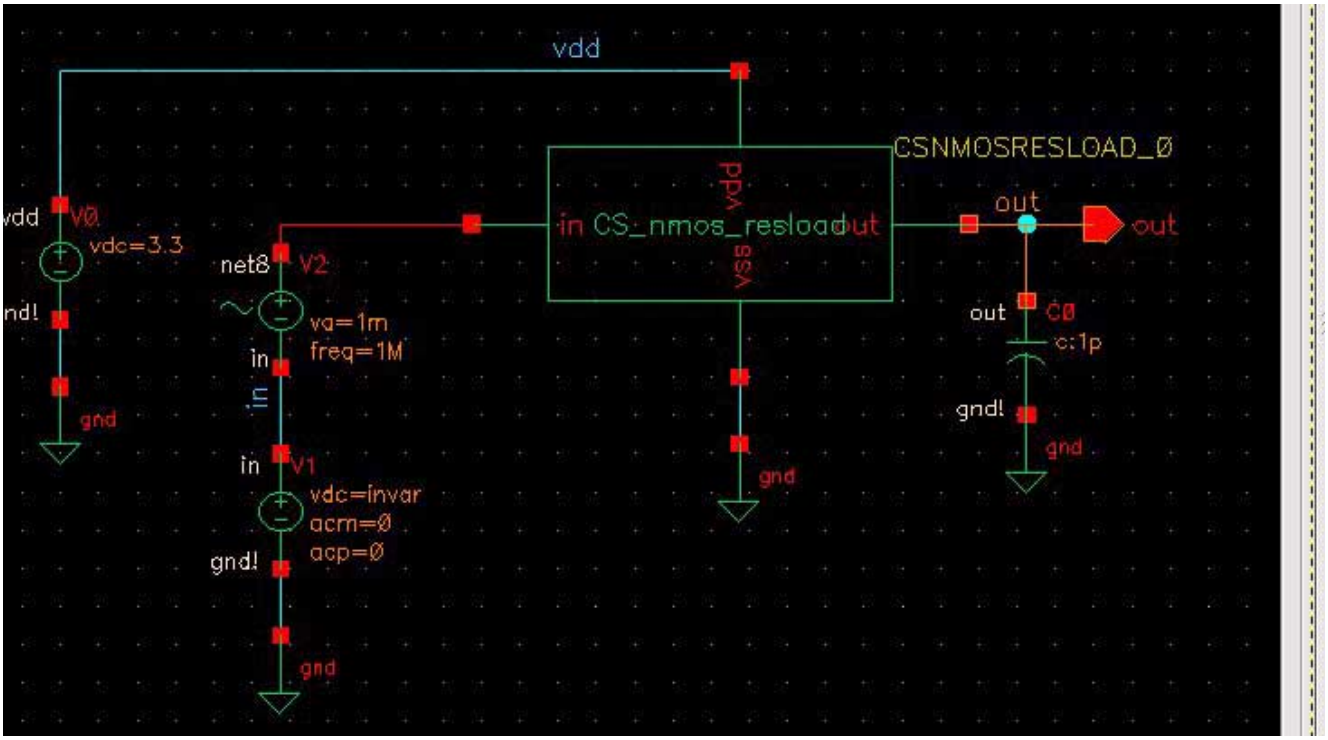


Figure 19: test_CS_nmos_resload with vsin source inserted.

.

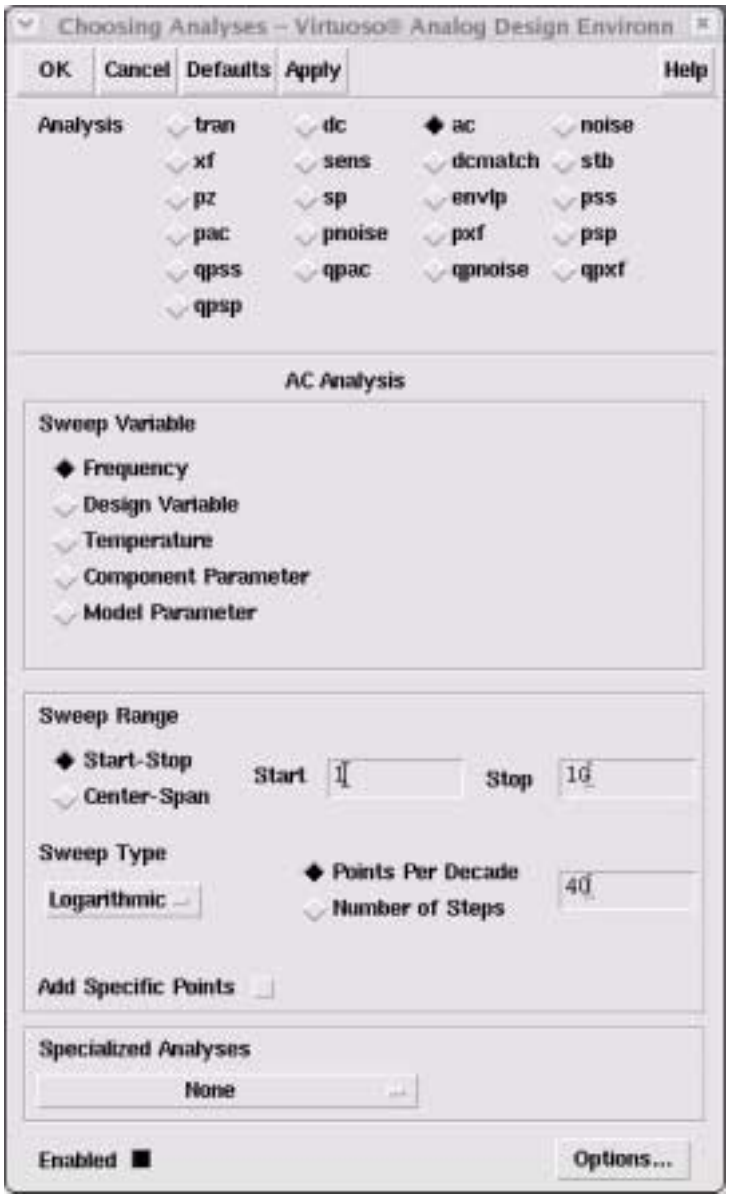


Figure 20: AC Analysis window

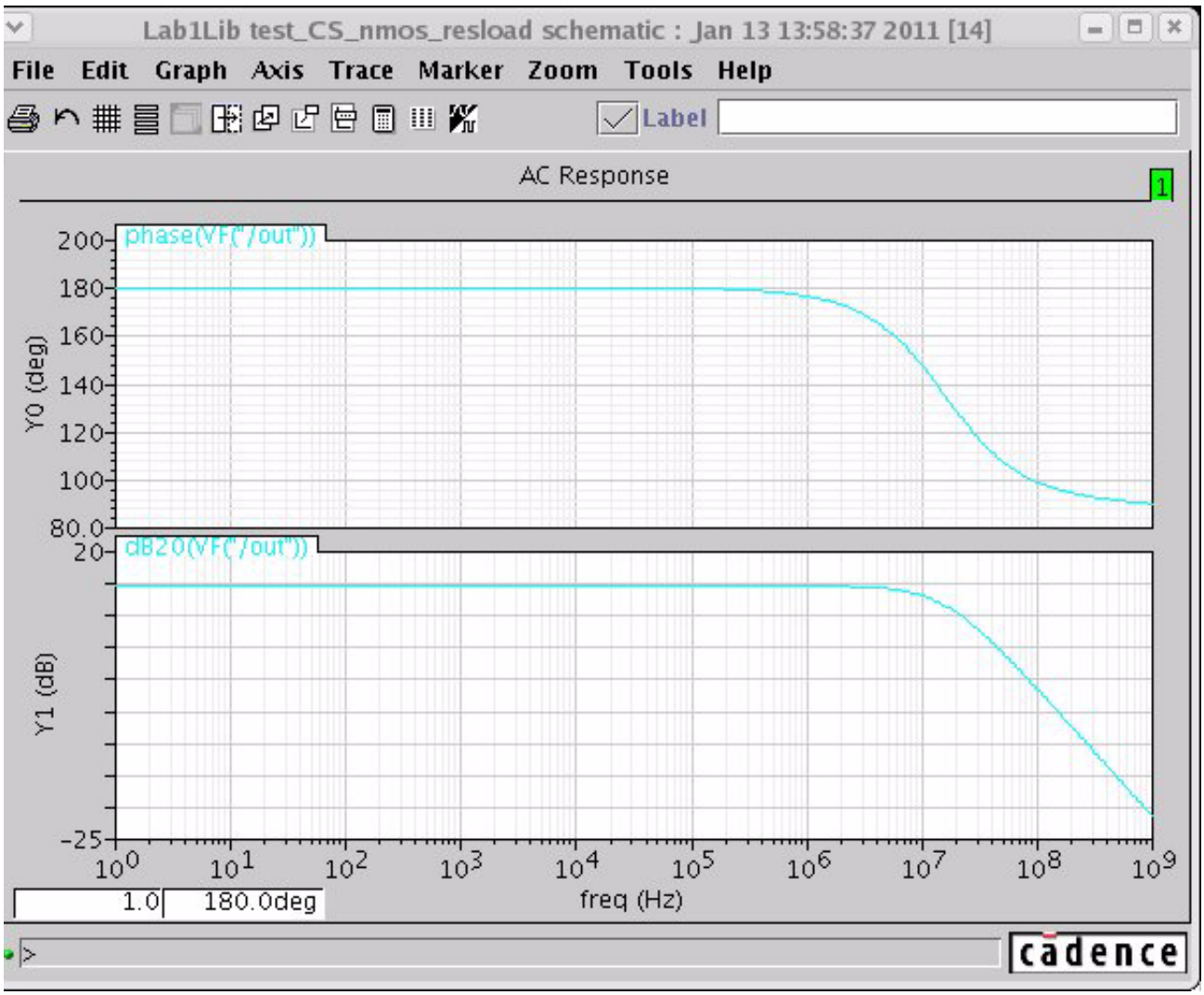


Figure 21: Wavescan AC simulation plot

Exercise 10: Setting Outputs

[1] Select ADE/Outputs/Setup...
In the Setting outputs window fill in the values as shown in Figure 22.
This will plot the AC gain.

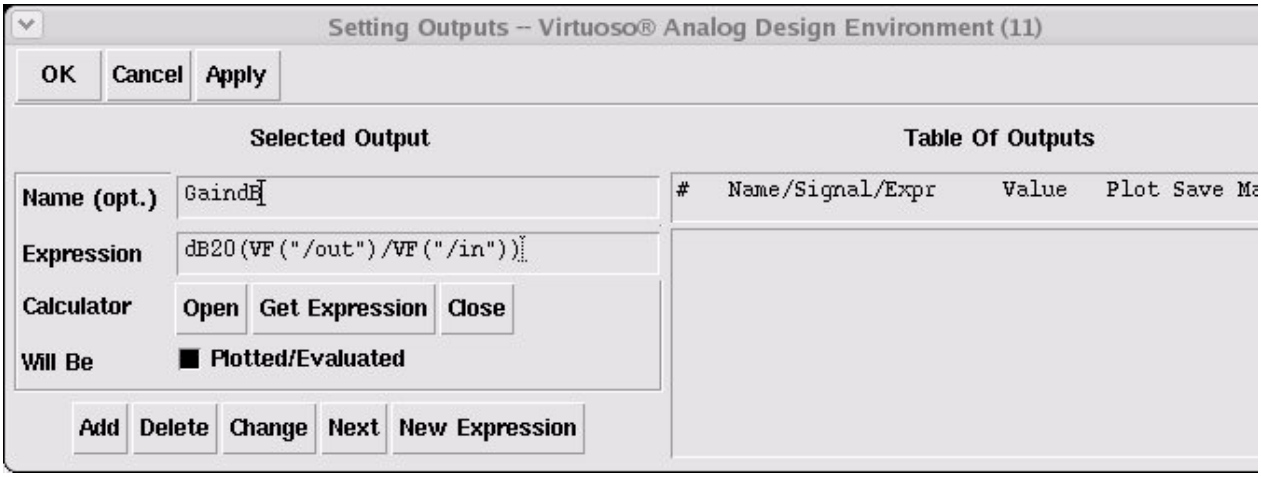


Figure 22: Setting outputs window, GaindB

[2] Select ADE/Outputs/Setup...
In the Setting outputs window fill in the values as shown in Figure 23.
This will write the value of the 3dB bandwidth to the ADE Outputs window.

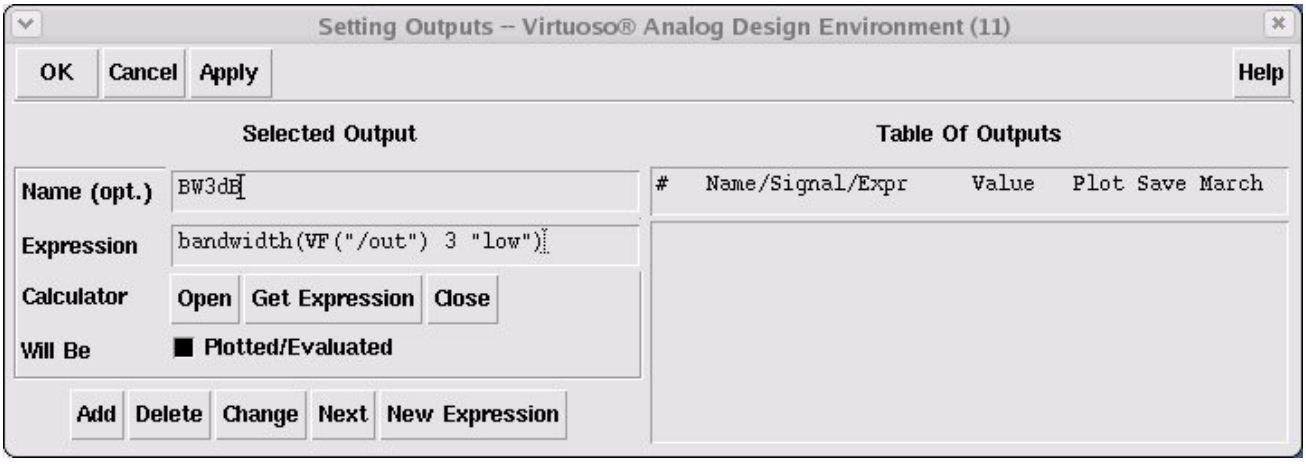


Figure 23: Setting outputs window, 3dB bandwidth

[3] Select ADE/Outputs/Setup...
In the Setting outputs window fill in the values as shown in Figure 24.
This will plot the phase.

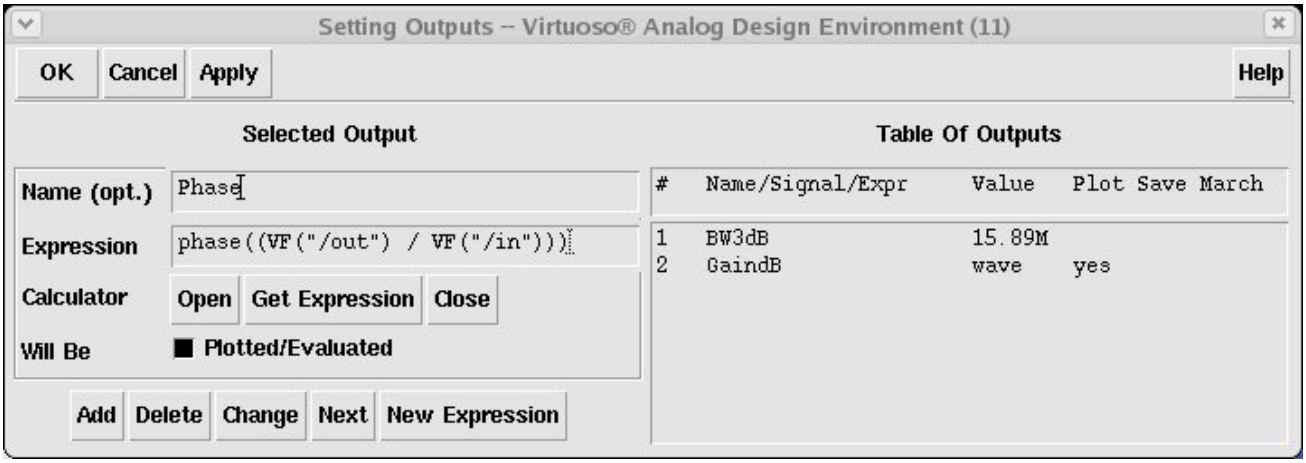


Figure 24: Setting outputs window, Phase

[4] Select ADE/Outputs/Setup...
In the Setting outputs window fill in the values as shown in Figure 25.
Note the Expression field entry is: value(dB20(VF("/out")) 1)
This will write the value of the gain at 1Hz (i.e. DC gain) to the ADE Outputs window

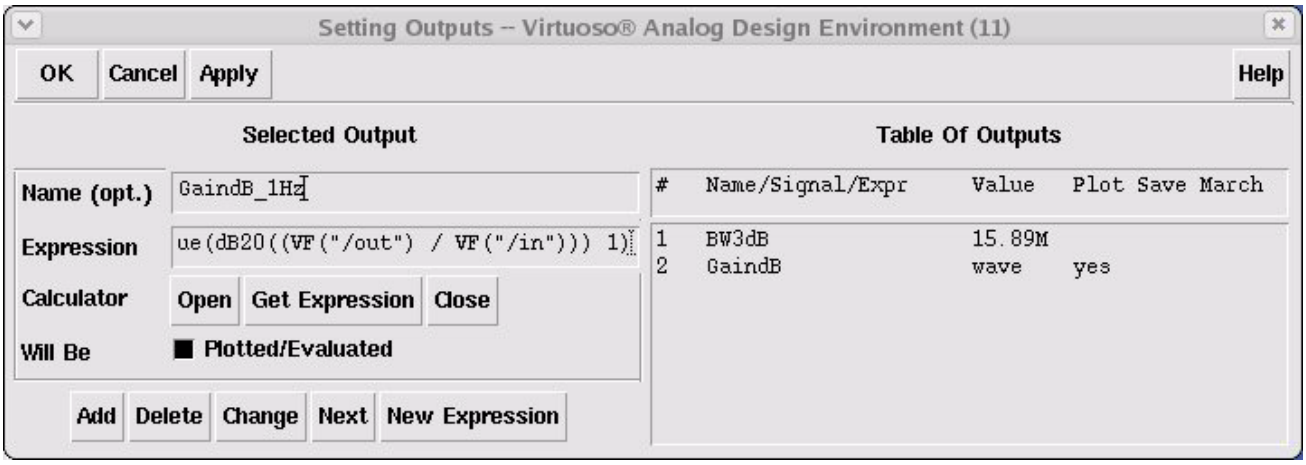


Figure 25: Setting outputs window, Gain in dB at 1Hz.

[5] Select ADE/Analyses/Choose.. In the Choosing Analyses window click on AC and set the stop frequency to 1T.

[6] Select ADE/Netlist and Run.

[7] The gain plot now appears automatically, and the ADE window outputs section shows the values for the 3dB bandwidth and the low-frequency gain i.e. the outputs of the functions defined.
Refer to the Exercise worksheet in Appendix A and fill in the required values in Table 6.
Print out the plot which should be handed up with the worksheet in Appendix A

2.3.5 Running a transient simulation

A transient simulation is a simulation in the time domain. The simulator performs an initial DC analysis and then an incremental timestep analysis.

Exercise 11: Running a transient simulation

[1] Modify the object properties of the vsin element and set the Amplitude to 1m and the Frequency to 1M.

[2] Select ADE/Analyses/Choose.. In the Choosing Analyses window, click on tran and set the options as shown in Figure 26 (i.e. set the stop time to 10u, set the accuracy to conservative).

[3] Select ADE/Netlist and Run.

[4] Select ADE/Results/Direct Plot/ Transient Signal, left-click on the input and output nodes, and press Esc..

[5] From the transient plot check the input and output sine wave amplitudes and check that the gain is in line with that predicted from the DC and AC analyses.
Refer to the Exercise worksheet in Appendix A and fill in the required values in Table 7.
Print out the plot which should be handed up with the worksheet in Appendix A.

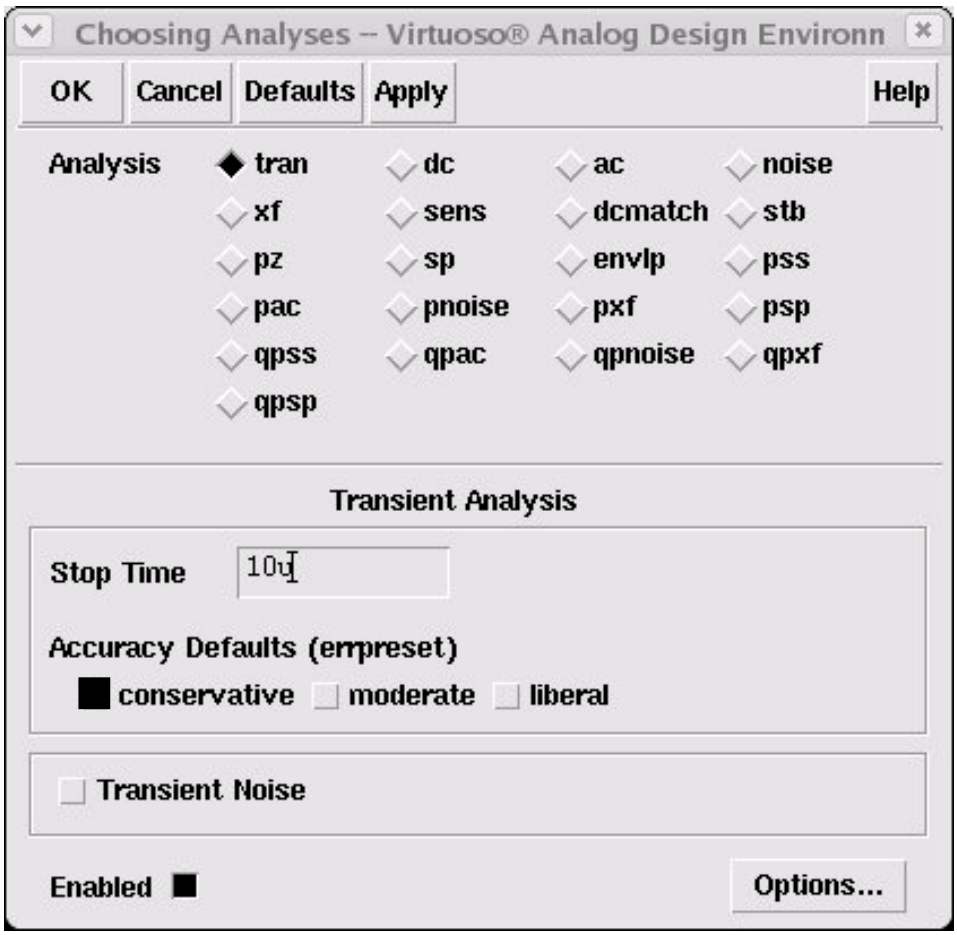


Figure 26: TR Analysis window

Exercise 12: Transient simulation with large input signal

[1] Modify the object properties of the vsin element and set the Amplitude to 500m and the Frequency to 1M.

[2] Select ADE/Outputs/Setup...
In the Setting outputs window fill in VT("/out")as shown in Figure 27., and press Add.
Repeat for VT("/in")
This will plot the transient input and output signals automatically after a simulation has run.

[3] Select ADE/Netlist and Run.

[4] The transient input and output signals should now appear automatically.

[5] From the transient plot check the input and output sine wave amplitudes. Note the strong non-linearity on the output signal as it clips at the top and bottom. As the signal is outside the linear range of the circuit the gain will be much less than that predicted from the DC and AC analyses.
Refer to the Exercise worksheet in Appendix A and fill in the required values in Table 8.

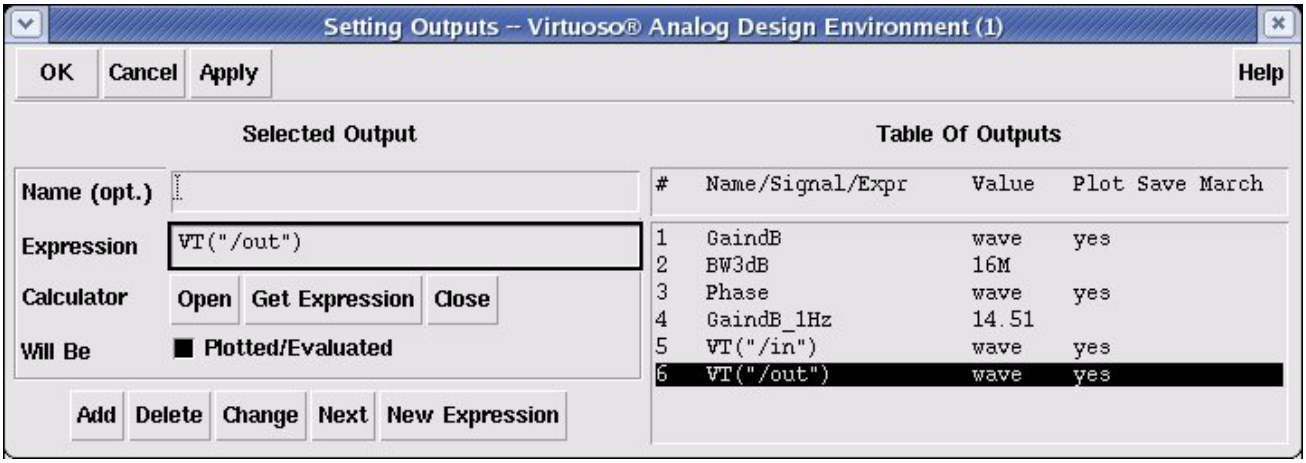


Figure 27: Setting outputs window

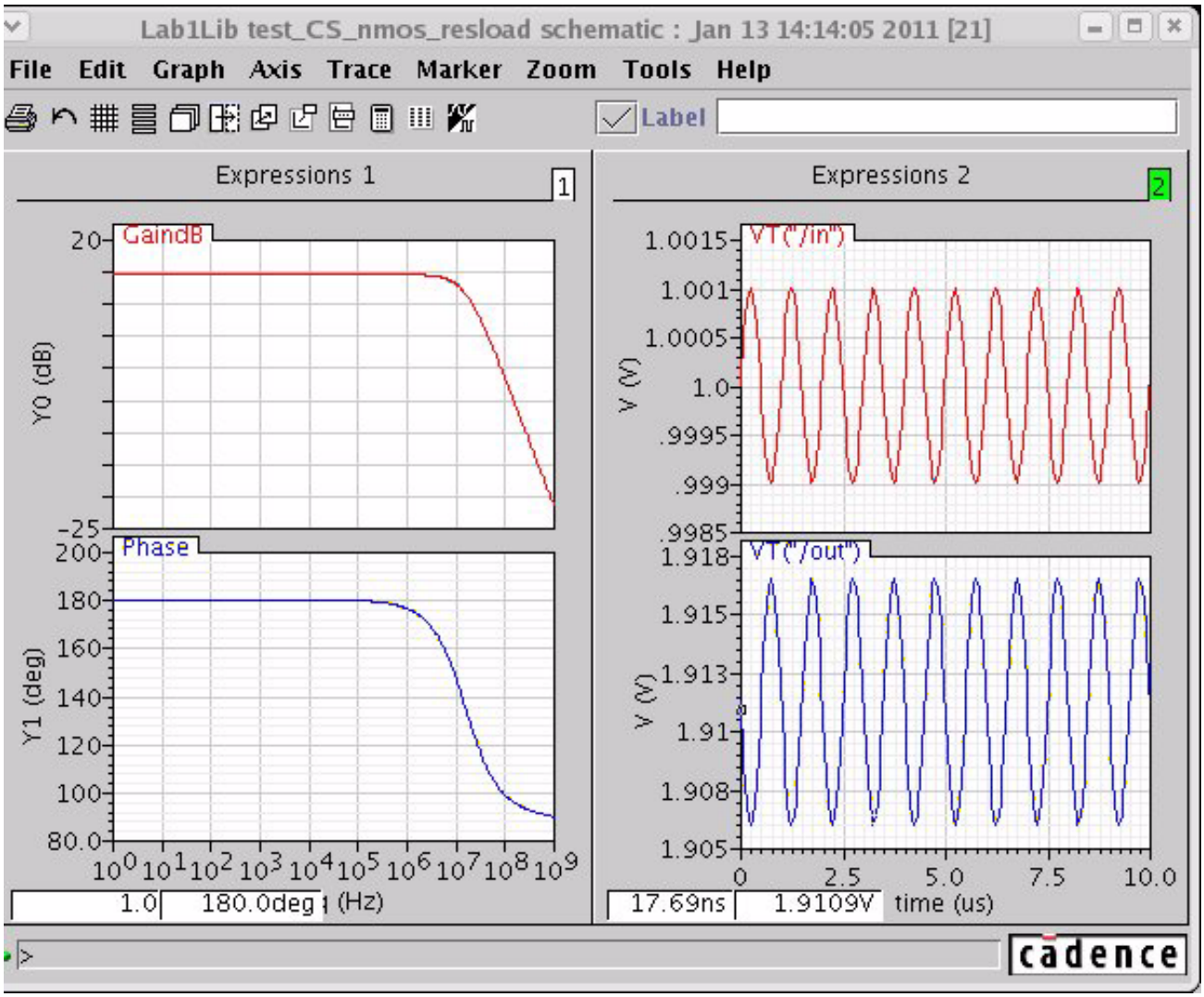


Figure 28: Wavescan window showing AC and Transient plots

2.3.6 Saving states

It is convenient to save the state of a simulation i.e. the settings including the model files, the temperature, and particularly the output functions. These can be re-loaded next time you access ADE.

Exercise 13: Saving a State

[1] In the ADE/Session/Save State... menu, choose Save State Option: Cellview, enter a suitable name for the state in the Cellview Options State field, and then press OK (see Figure 30). Here the state has been named spectre_acdctr, but any suitable name is OK. The artist state is stored as a cellview alongside the schematic cellview (see Figure 29).

To close ADE use ADE/Session/Quit.

To close composer use SC/Window/Close

To exit icfb use CIW/File/Exit

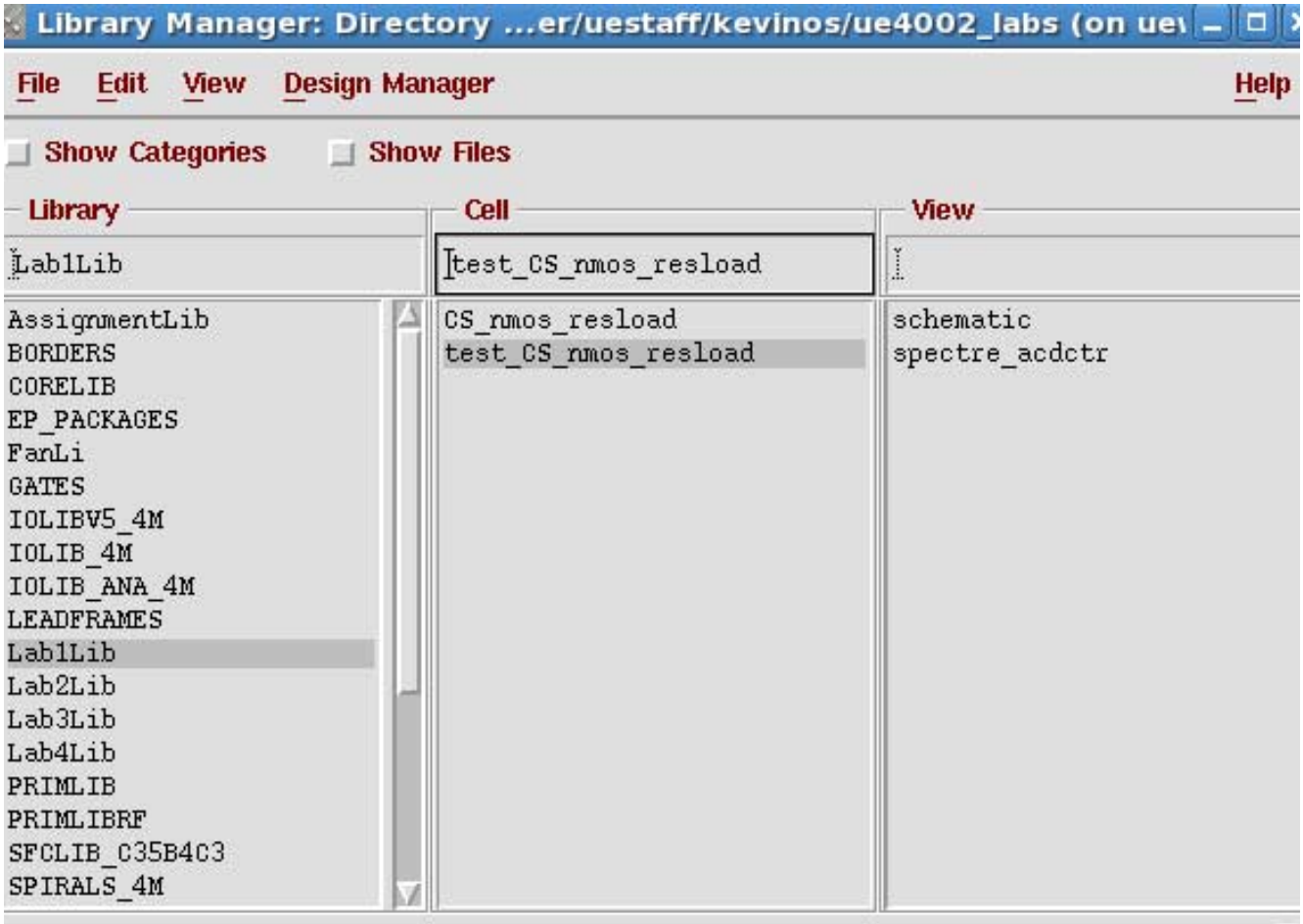


Figure 29: Library manager showing artist state stored as cellview

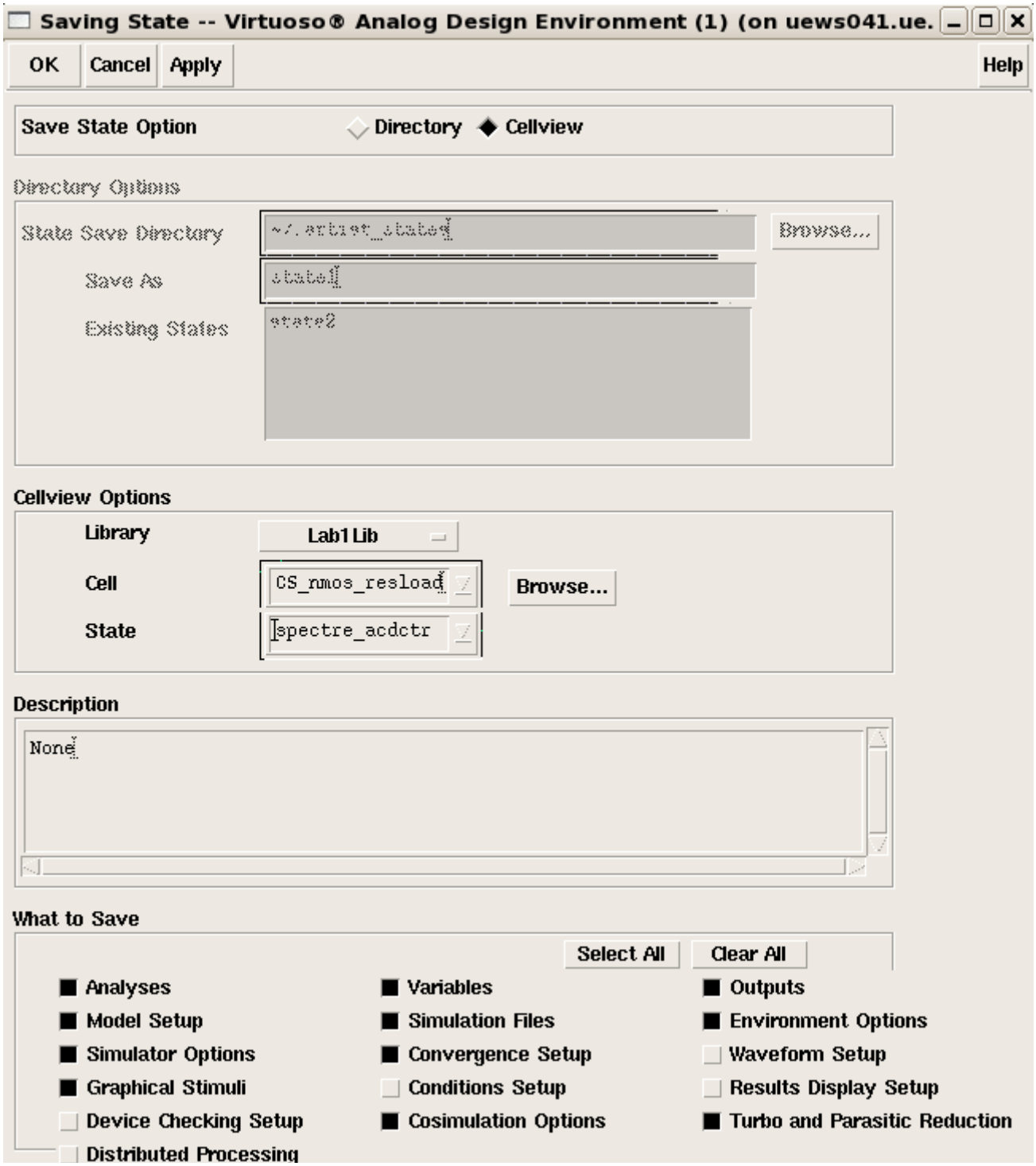


Figure 30: Saving State Menu

3 Lab 2 - Further Spectre analyses and using the ADE calculator

This lab deals firstly with further analyses sub-types (sweeps), and using the calculator tools. Using these, various aspects of transistor behaviour, and deviations from the simple model used in the course notes, are highlighted.

Log on to a terminal.

Open a terminal. (right-click on workspace and open Tools/Terminal...)

Navigate to your 4002 lab directory

> cd UE4002_labs

.To start cadence type
> ams_cds -mode fb

3.1 Further Analyses

This section looks at using the calculator to set-up outputs in ADE, running DC sweeps and running AC analyses sweeping variables other than frequency.

3.1.1 Running A DC Sweep (Parametric Analysis)

A DC sweep allows you to get simulation outputs as a function of a design variable and can be an effective way of investigating the limitations of a circuit. In ADE the DC sweep can be run using the dc analysis directly or using a separate Parametric Analysis (PA) tool. Operating points can only be output with the PA tool. In the sections below, the Calculator is first used to set up a number of outputs, which will then automatically be plotted when the parametric analysis completes.

Exercise 14: Setting up DC Operating Point (OP) outputs using the calculator.

As seen in Lab 1 it is possible to view the DC Operating Points of components using Results/Print/DC Operating Points. It is also possible to set up individual operating points as outputs in ADE. This can be done using ADE/Outputs/Setup... and entering the outputs manually as previously, or by using the Calculator tool. In this exercise we use the calculator tool.

[1] Open the test_CS_nmos_resload schematic from the previous lab.

[2] Open the Analog Design Environment (SC/Tools/Analog Environment)

[3] Reload the simulation state you saved at the end of Lab1 with ADE/Session/Load State.
In the Loading State menu select Load State Option: Cellview, and select the state you saved in the Cellview Options State field (Figure 32).

[4] Select ADE/Outputs/Setup...

[5] In the Setting Outputs menu which appears, click on Calculator Open.

[6] In the calculator window which appears select info and then select op (Figure 31). A Select instance window appears.

[7] Navigate into the CS_nmos_resload schematic and select the nmos transistor.
The small Select Instance window changes name to Op parameters for MN0.
Click on List and select gm.
In the Calculator the expression
OP("/CSNMOSRESLOAD_0/MN0","gm")

(or similar, depending on your instance name), now appears in the buffer field (Figure 31).
Note: If the operating points do not appear, try running a DC simulation and try again.

[8] In the Setting outputs window click on Get expression to grab the expression which appears in the expression field. You can optionally give a name e.g. gm in the Name field. Then click on Add. The output should then appear in the ADE Outputs section.

[9] Add additional expressions to output ids and gds. You can either do this by repeating the above steps and selecting the appropriate OP parameter, or by double-clicking on the gm output in ADE, modifying the Expression and Name fields in the Setting outputs window and clicking on Add.

[10] In the Setting outputs window manually enter the following:
In the name field: vsat
In the Expression field: headroom = (OP("/I0/MN0" "vds") - OP("/I0/MN0" "vdsat"))
vds is the drain-source voltage and vdsat the minimum required vdsat (approximately equal to vgs-vt in the BSIM3 model). So the vsat output will calculate how much saturation headroom a transistor has. If the headroom becomes negative then the transistor is out of saturation.

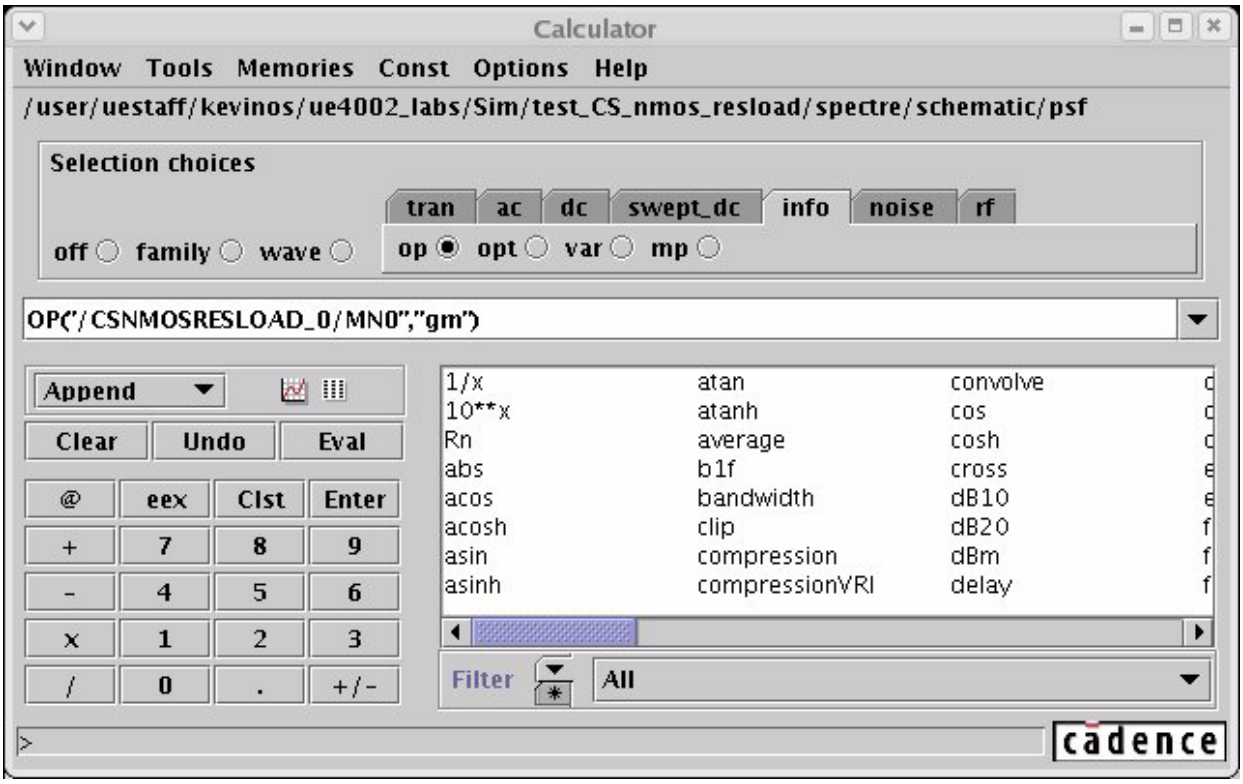


Figure 31: Calculator window

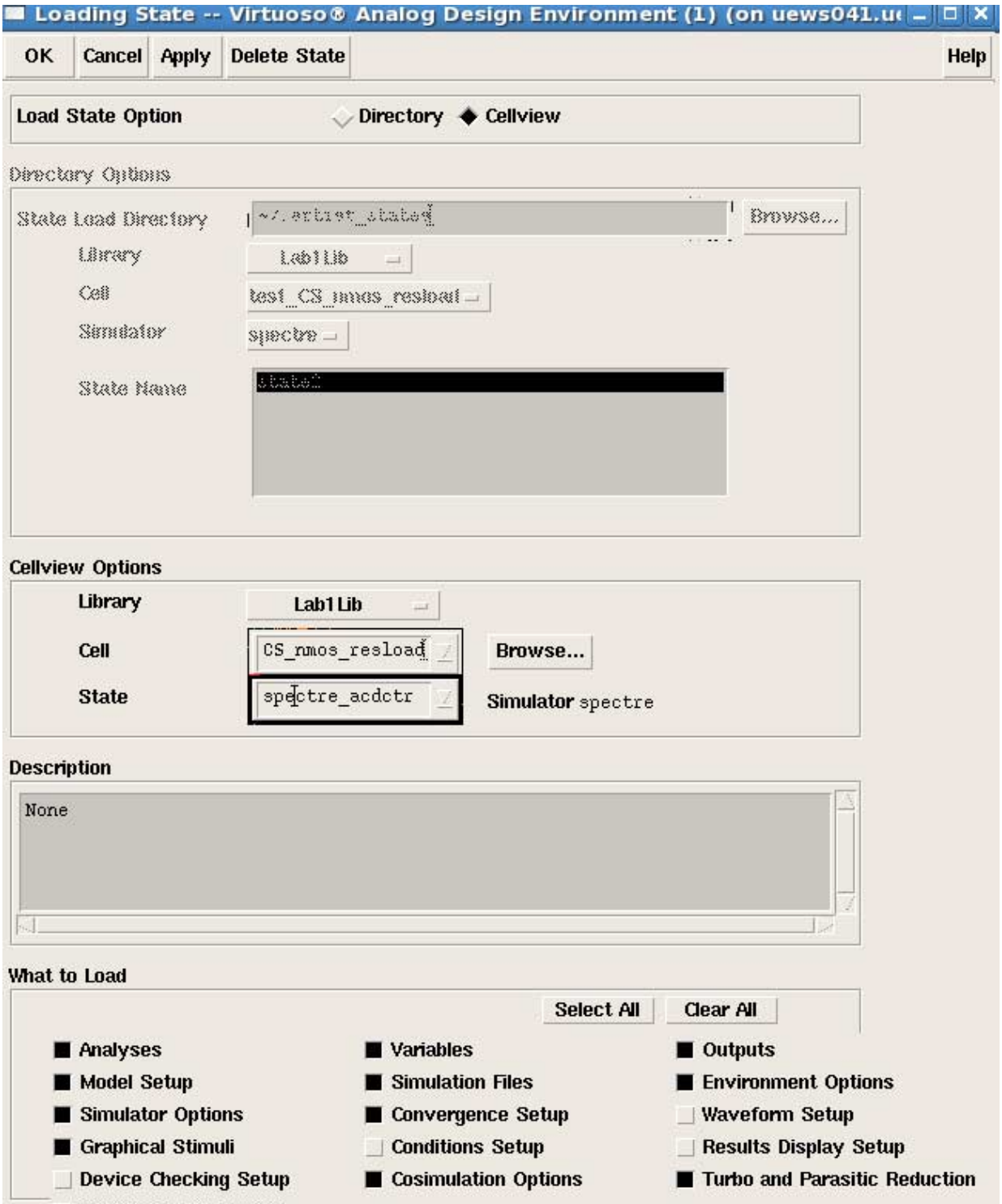


Figure 32: Loading State Menu

Exercise 15: Setting up a dc voltage output using the calculator.

In this exercise, the calculator is used to set up a dc output voltage and its derivative.

[1] Open ADE. Open ADE/Outputs/Setup...

[2] In the Setting Outputs menu which appears click on Calculator Open.

[3] In the calculator window which appears select vdc.

[4] Select the output node in the test_CS_nmos_resload schematic. In the Calculator the expression VDC("/out") now appears in the buffer field.

[5] In the Setting outputs window click on Get expression to grab the expression which appears in the expression field. You can optionally give a name e.g. VDC_out in the Name field. Then click on Add. The output should then appear in the ADE Outputs section.

[6] In the Calculator with VDC("/out") still in the buffer, select Special Functions/deriv. Edit to -deriv(VDC("/out")). See Figure 33. Grab the expression -deriv(VDC("/out")) in the Setting outputs window, and click on Add to add this output to the ADE Outputs section.

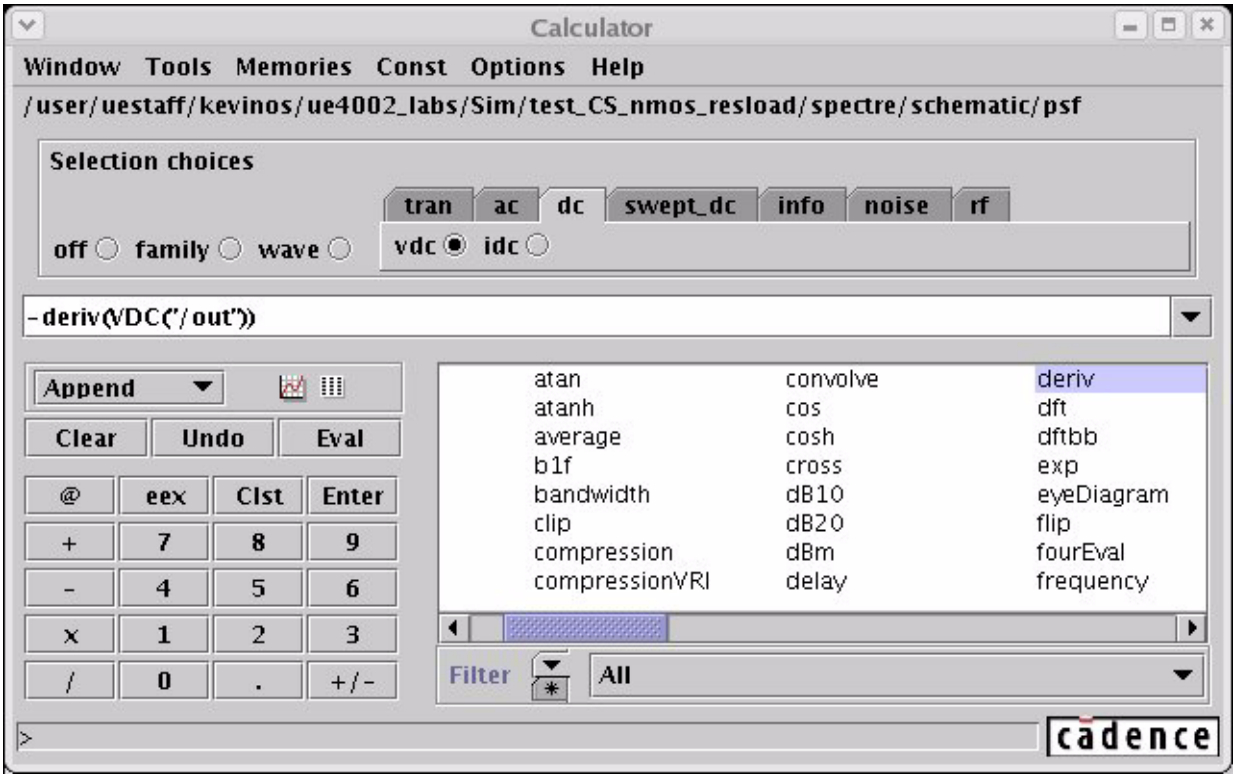


Figure 33: Calculator window with deriv function

Exercise 16: Using variables in an analysis.

Up to now we have entered fixed component values. For a swept analysis we need to assign design variables to components.

[1] Modify the object properties of the DC voltage source in test_CS_nmos_resload and change DC Voltage to some variable name e.g. invar. Check and Save the schematic.

[2] Modify the object properties of the resistor in CS_nmos_resload and change the value of the resistance to some variable name e.g. rload. Check and Save the schematic.

[3] Select ADE/Variables/Copy From Cellview. The Variables will appear in the Design Variables section of the ADE.

[4] Double left-click on each variable and supply values e.g. 1 for invar and 10k for rload.

[5] Run the dc analysis with ADE/Netlist and Run.

[6] Record the values which appear in the ADE Output section for gm,gds, ids and the DC output voltage.
Refer to the Exercise worksheet in Appendix B and fill in the required values in Table 9.

Exercise 17: Running a DC sweep analysis (1).

In this exercise the DC bias voltage of the NMOS transistor is swept using the Parametric Analysis tool.

[1] Disable any other simulations (e.g. ac, tran) by turning off the Enabled switches in the respective menus. (You can double click on the simulation in the ADE/Analyses field to pop up the simulation menu).

[2] Select ADE/Tools/Parametric Analysis...

[3] In the Parametric Analysis (PA) window which appears enter the Variable Name (invar), the Range (0.9 to 1.4) and the Steps (set Step Control to Linear Steps, set the Step Size to 0.01). (See Figure 34).

[4] Select PA/Analysis/Start.

[5] When the simulation completes, the outputs you have set-up in ADE will be plotted in the Waveform Window (WW) as a function of the variable invar, the DC bias voltage.
You may need to set WW/Axes/To Strip, and maximise the WW to see each parameter separately.
Note the current characteristic and how gm and the derivative of the output voltage peak around where the transistor goes out of saturation.
Note also (not apparent from the graph unless you change the y axis range) that the value of gds is not constant while the transistor is in saturation. It is strongly dependant on the value of vds i.e. on how far in saturation the transistor is.
Refer to the Exercise worksheet in Appendix B and fill in the required values in Table 10.
Recall that WW/Trace/Trace Cursor or shortcut c will show numeric values as you navigate along a plot.
WW/Marker/Place/Vertical Marker or shortcut v will place a vertical marker and show the numeric values at the marker location of all plots.

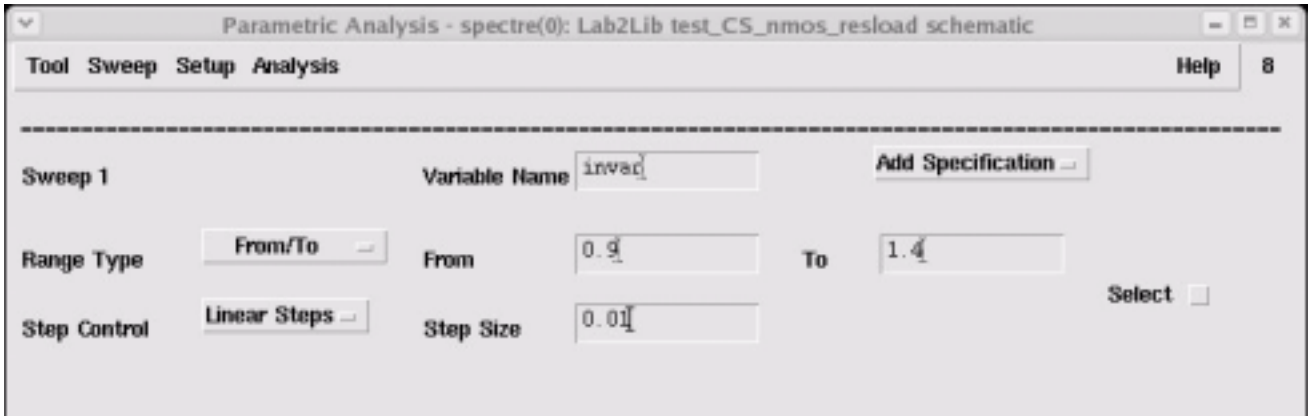


Figure 34: Parametric Analysis window.

Exercise 18: Running a DC sweep analysis (2)

In this exercise the DC analysis sweep is used.

[1] Select ADE/Analyses/Choose and select dc.

[2] In the Choosing Analyses form
Select Design Variable, and enter the parameter you used for the DC bias voltage (e.g. invar).
Set the Sweep Range to Start-Stop, Start to 0, and Stop to 2.
Set the Sweep Type to Linear and the Step Size to 0.01.
Set the Enabled switch on if it is not already on
OK the form (which should look like Figure 35).
You may wish to disable other simulations (e.g. ac, tran) by turning off the Enabled switches in the respective menus.

[3] Set up swept outputs: (for the DC analysis the calculator functions are VS, IS instead of VDC, IDC).
Double left-click on VDC("/out") in the ADE Output section.
In the Setting Outputs Window, change to VS("/out") and click on Add.
Double left-click on -deriv(VDC("/out")) in the ADE Output section.
In the Setting Outputs Window, change to -deriv(VS("/out")) and click on Add
Alternatively you can set up these output functions using ADE/Outputs/Setup and selecting VS in the calculator.

[4] Run the simulation with Select ADE/Simulation/Run

[5] When the simulation completes, the output voltage and its derivative will be plotted as a function of the input voltage.
Note that this sweep simulation runs much faster than the previous method. Unfortunately it is not possible to output the small-signal parameters using this method.
Refer to the Exercise worksheet in Appendix B and fill in the required values in Table 11
Print out the plot which should be handed up with the worksheet in Appendix B.

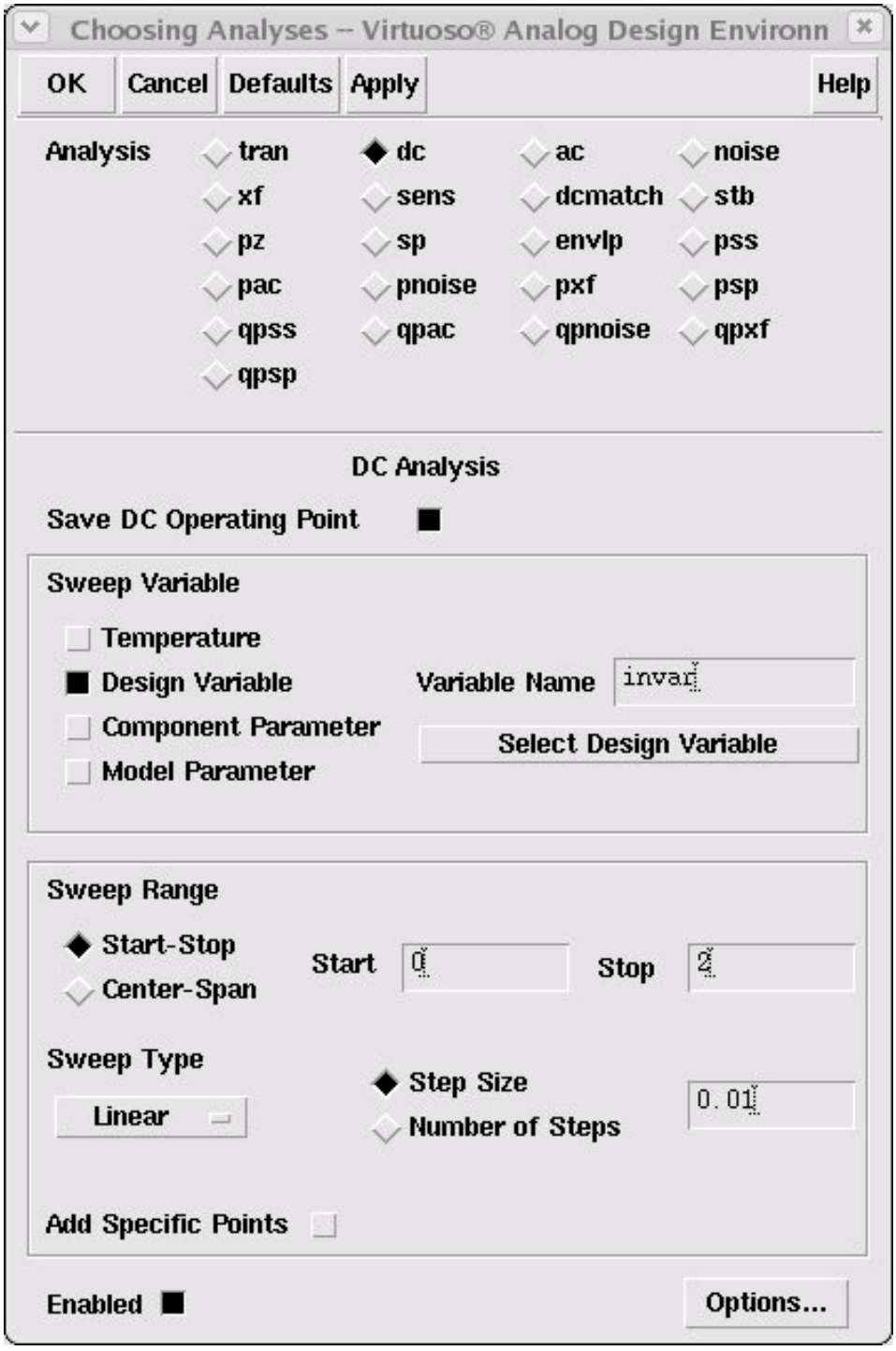


Figure 35: DC Sweep analysis input

3.1.2 AC Sweep Analysis

It is also possible to run the linearised AC analysis with the swept variable set to a design variable rather than frequency.

Exercise 19: Running an AC Sweep Analysis with a Design Variable as swept parameter

[1] Select ADE/Analyses/Choose and select ac

[2] Modify the analysis:
Set the Sweep Variable to Design Variable, and enter the parameter you used for the DC bias voltage (e.g. invar).
Enter 1 in the At Frequency field
Set the Sweep Range to Start-Stop, Start to 0, and Stop to 3.3.
Set the Sweep Type to Linear and the Step Size to 0.01.
Set the Enabled switch on if it is not already on
OK the form (which should look like Figure 36).
You may wish to disable other simulations (e.g. dc, tran) by turning off the Enabled switches in the respective menus.

[3] Select ADE/Simulation/Run

[4] When the simulation completes select WW/Results/Direct Plot/ AC dB20 and click on the output node.
The ac output voltage is now plotted as a function of the input dc voltage (invar).
As the magnitude of the ac input voltage is 1, the ac output voltage gives the gain of the circuit.. The variation of gain with input bias voltage can again be seen clearly.
Refer to the Exercise worksheet in Appendix B and fill in the required values in Table 12.

[5] Save the simulation state with ADE/Session/Save State.

Exercise 20: Running an AC Sweep Analysis with the load resistance Design Variable as swept parameter

[6] Repeat Exercise 19 sweeping not the DC bias voltage but the load resistance.
Refer to the Exercise worksheet in Appendix B and fill in the required values in Table 13.
Print out the plot which should be handed up with the worksheet in Appendix B.

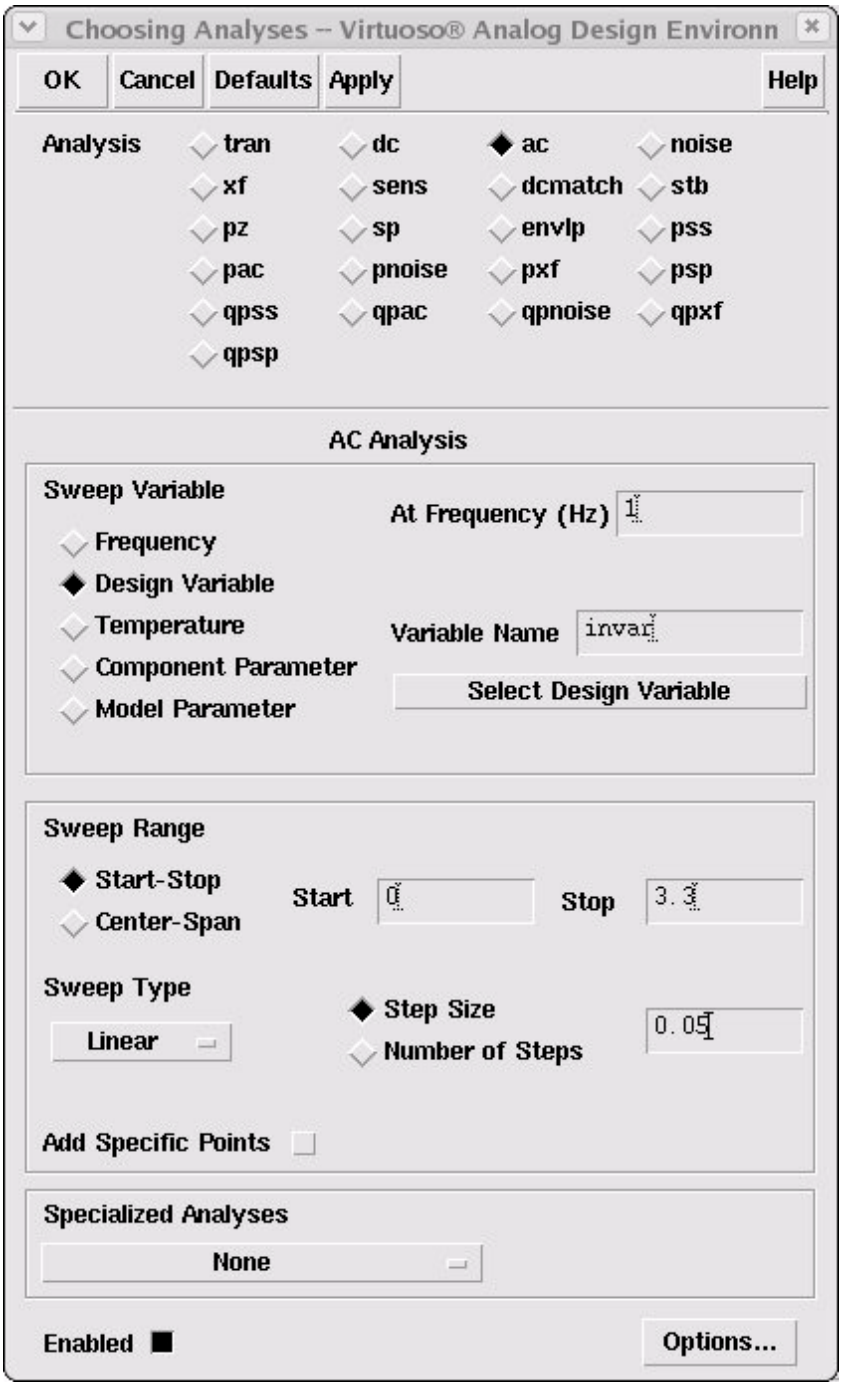


Figure 36: Choosing Analysis window for AC sweep with Design Variable.

4 Lab 3 - Design and Simulation of current mirrors

In this section a simple current mirror and a cascode mirror are designed and simulated[1].

This lab deals firstly with further analyses sub-types (sweeps), and using the calculator tools. Using these, various aspects of transistor behaviour, and deviations from the simple model used in the course notes, are highlighted.

.Log on to a Sun terminal.

Open a terminal. (right-click on workspace and open Tools/Terminal...)

Navigate to your 4002 lab directory

> cd UE4002_labs

.To start cadence type
> ams_cds -mode fb

4.1 Simple current mirror

This section looks at the design and simulation of a simple current mirror/

Exercise 21: Design a simple current mirror

[1] Create a schematic cellview test_currmir1 in your work library.

[2] Draw the schematic of a simple NMOS 1:1 current mirror, similar to the one shown in Figure 37 below. You may keep the transistor dimensions at 10/1 or use your own values. In addition to the current mirror transistors (library PRIMLIB, cellview nmos4), the following sources and grounds are required from the analogLib library:
supply voltage (vdc). Set the DC voltage to a variable name e.g. vddvar.
input current source (idc). Set the DC current to a variable name e.g. iinvar.
output voltage source (vdc) to set the voltage at the output current mirror. Set the DC voltage to a variable name e.g. vout.
ground symbol (gnd).
Add names to the wires (key short-cut l) vdd, in and out as shown).

[3] When finished Check and Save.

1. See Section 5 of class notes.

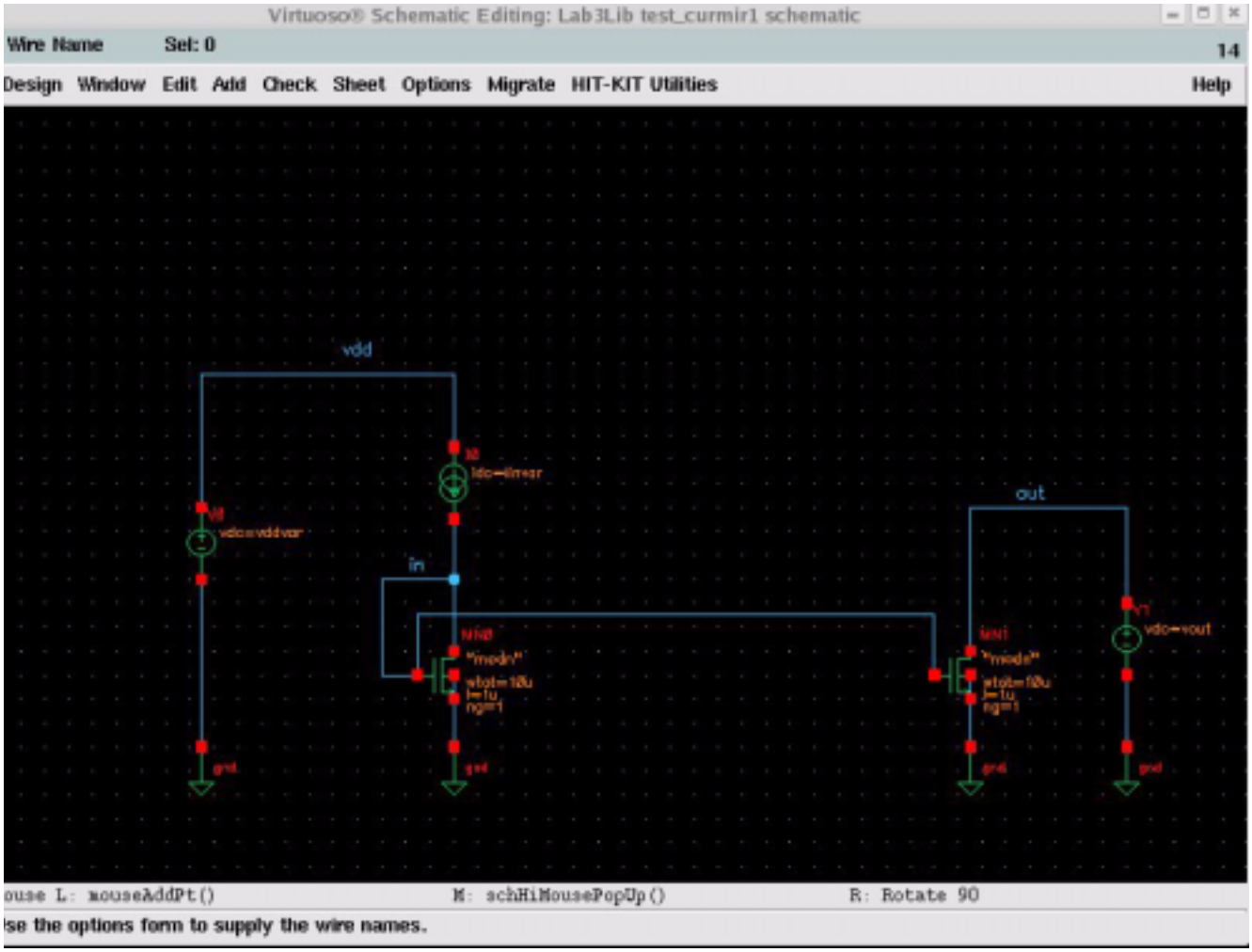


Figure 37: test_curmir1 schematic.

Exercise 22: DC Simulation of a simple current mirror

In this exercise the output current is added as output to the ADE window, as well as the DC voltage, and DC simulation are run examining the behaviour as function of output voltage and transistor dimensions.

[1] Open Analog design Environment with SC/Tools/Analog Environment

[2] Select ADE/Variables/Copy from Cellview, and fill in the following values for the variables (double left-click on them to bring up an edit window):
vddvar 3.3
iinvar 100u
vout 1.5V

[3] In order to get the output current (i.e. drain current of the output NMOS), the drain terminal first needs to be saved as an output:
Select ADE/Outputs/To Be Saved/Select on schematic.
Click on the drain terminal of the current mirror output NMOS. You need to click exactly on this terminal. If successful the terminal will be circled.
Click on Esc.

[4] Use the calculator to output the drain current:
Select ADE/Outputs/Setup.
In the Setting Outputs menu which appears click on Calculator Open.
In the calculator window which appears select IDC.
Click on the drain terminal of the current mirror output NMOS.
In the Calculator the expression IDC("/MN1/d") or similar depending on the instance name should appear.
In the Setting outputs window click on Get expression to grab the expression which appears in the expression field.
Then click on Add. The output should then appear in the ADE Outputs section.

[5] Use the calculator to output the dc voltage at the node in:
Select ADE/Outputs/Setup.
In the Setting Outputs menu which appears click on Calculator Open.
In the calculator window which appears select VDC. Click on the iin node of the current mirror.
In the Calculator the expression VDC("/in") should appear.
In the Setting outputs window click on Get expression to grab the expression which appears in the expression field. Then click on Add. The output should then appear in the ADE Outputs section which should then look similar to that shown in Figure 38

[6] Select ADE/Analyses/Choose and make sure the dc analysis is enabled.

[7] Select ADE/Simulation/Netlist and Run.

*[8] **Refer to the Exercise worksheet in Appendix C and record the following values in Table 14:***
Record the mirror input voltage and output current.
Calculate the mirror inaccuracy.
Record also the vdsat operating point parameter of the output NMOS (Select ADE/Results/Print/DC Operating Point and select the transistor).

*[9] Set vout to be equal to vdsat (Double-click on vout in the Design Variables win) and re-run the simulation. **Refer to the Exercise worksheet in Appendix C and record the following values in Table 14:**Record the output current and the mirror inaccuracy.*

*[10] Set vout to 1.5 again. Double the width and length of both transistors. Re-run the simulation. **Refer to the Exercise worksheet in Appendix C and record the following values in Table 14:** the output current and mirror inaccuracy. Has the accuracy improved by a factor two?*

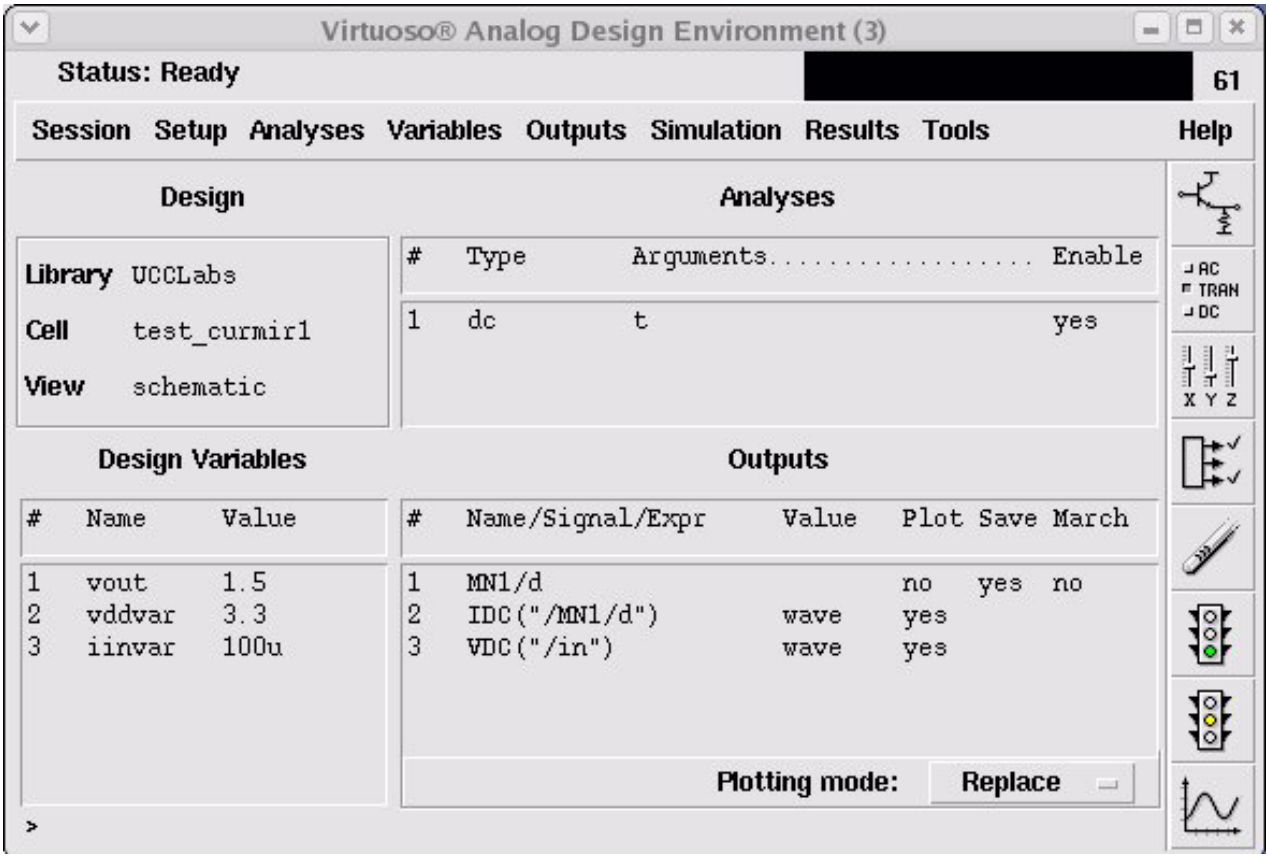


Figure 38: ADE window for current mirror simulation.

Exercise 23: DC sweep simulation of a simple current mirror

In this simulation the output current is examined as a function of the output voltage.

[1] Select ADE/Analyses/Choose and select dc.

[2] In the Choosing Analyses form
Select Design Variable, and enter the parameter you used for the output voltage (e.g. vout).
Set the Sweep Range to Start-Stop, Start to 0, and Stop to 2.
Set the Sweep Type to Linear and the Step Size to 0.01. (Figure 39)
Set the Enabled switch on if it is not already on

[3] Set up swept outputs:
Double left-click on IDC("/MN1/d")in the ADE Output section.
In the Setting Outputs Window, change to IS("/MN1/d") and click on Add.
Alternatively you can set up these output functions using ADE/Outputs/Setup and selecting IS in the calculator.

[4] Run the simulation with Select ADE/Simulation/Run

[5] When the simulation completes, note the characteristic of the output current. from the plot.
Refer to the Exercise worksheet in Appendix C and record the following values in Table 15.
Print out the plot which should be handed up with the worksheet in Appendix C.

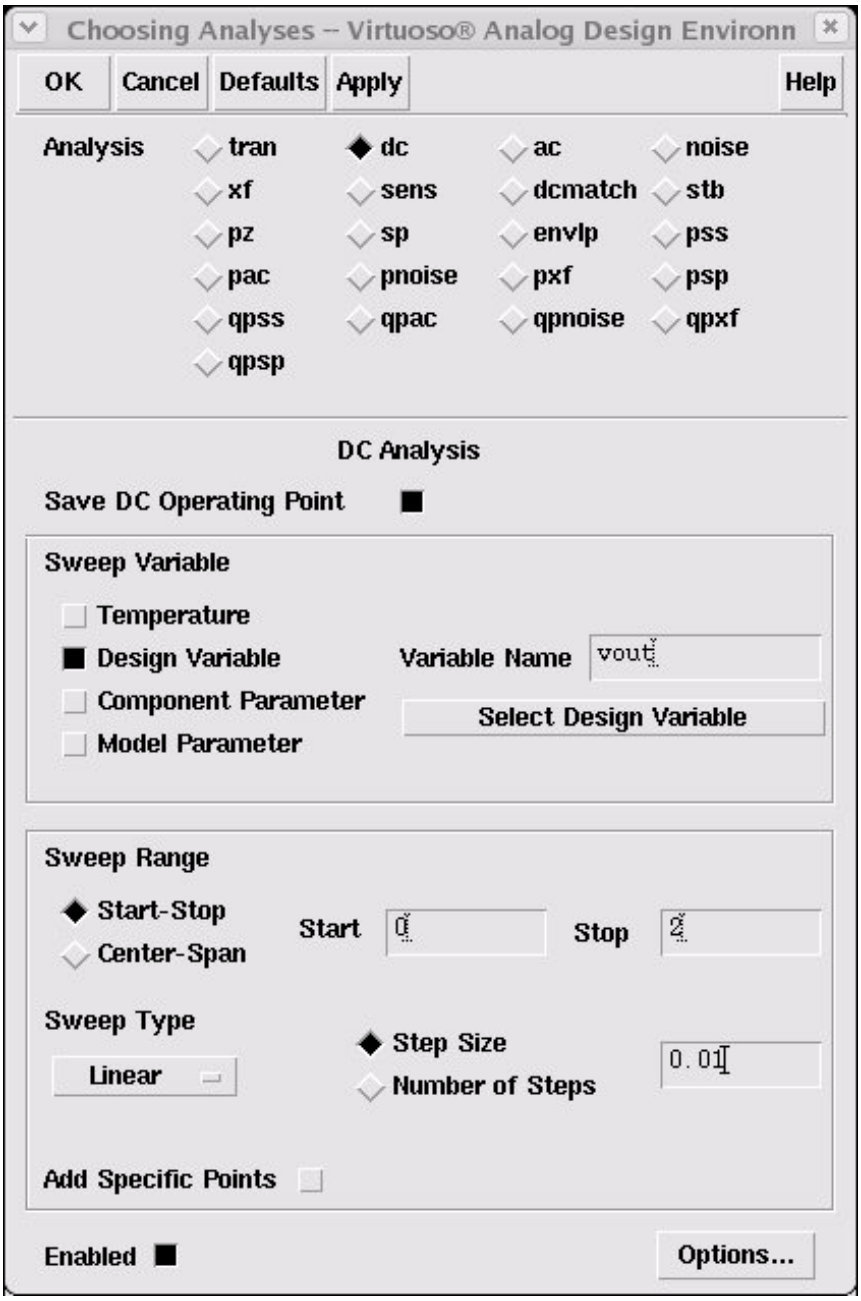


Figure 39: Settings for DC sweep analysis.

4.2 Cascode current mirror

In this exercise a low-voltage cascode mirror is designed and simulated.

Exercise 24: Design a cascode current mirror

[1] Copy the cell test_curmir1 to a new cell test_curmir2 in your work library. To do this:
Open the Library Manager (CIW/Tools/Library Manager...) ,
left click on your library
left click on test_curmir1 in the Cell column
with the right button held down on test_curmir1, select copy on the sub-menu which appears.
Fill out the Copy Cell form which appears, just changing test_curmir1 to test_curmir2 in to To Cell field. (Figure 40)

[2] Modify the test_curmir2 circuit to look like the schematic shown in Figure 40.
For the cascode transistors use the same width as the mirror transistors but use minimum length.
For the transistor which sets up the cascode voltage you need to set W/L such that the mirror transistors are in saturation. Try some initial value based on the guidelines in the course notes, check with a DC simulation that the transistors are in saturation, and iterate if necessary.
Make sure all NMOS bulk nodes are connected to ground.

[1] Open Analog design Environment with SC/Tools/Analog Environment

[2] Select ADE/Variables/Copy from Cellview, and fill in the following values:
vddvar 3.3
iinvar 100u
vout 1.0V

[3] You can use the calculator to output the drain current as in Exercise 22, or you can use the calculator to output the operating point parameter ids as shown in Exercise 14, or you can check the drain current manually each time with Select ADE/Results/Print/DC Operating Point.

[4] You can also use the calculator to output vdsat of the mirror output transistor and cascode, or you can check this manually each time with ADE/Results/Print/DC Operating Point.

[5] When the design is complete record vdsat of the mirror output transistor and cascode, and note the output current (should be 100u exactly)
Refer to the Exercise worksheet in Appendix C and fill in the required values in Table 16

Exercise 25: DC sweep simulation of a cascode current mirror

[1] Repeat the DC sweep simulation of Exercise 23 for this circuit.

[2] When the simulation completes, note the characteristic of the output current.
Refer to the Exercise worksheet in Appendix C and fill in the required values in Table 17
Print out the plot which should be handed up with the worksheet in Appendix C.

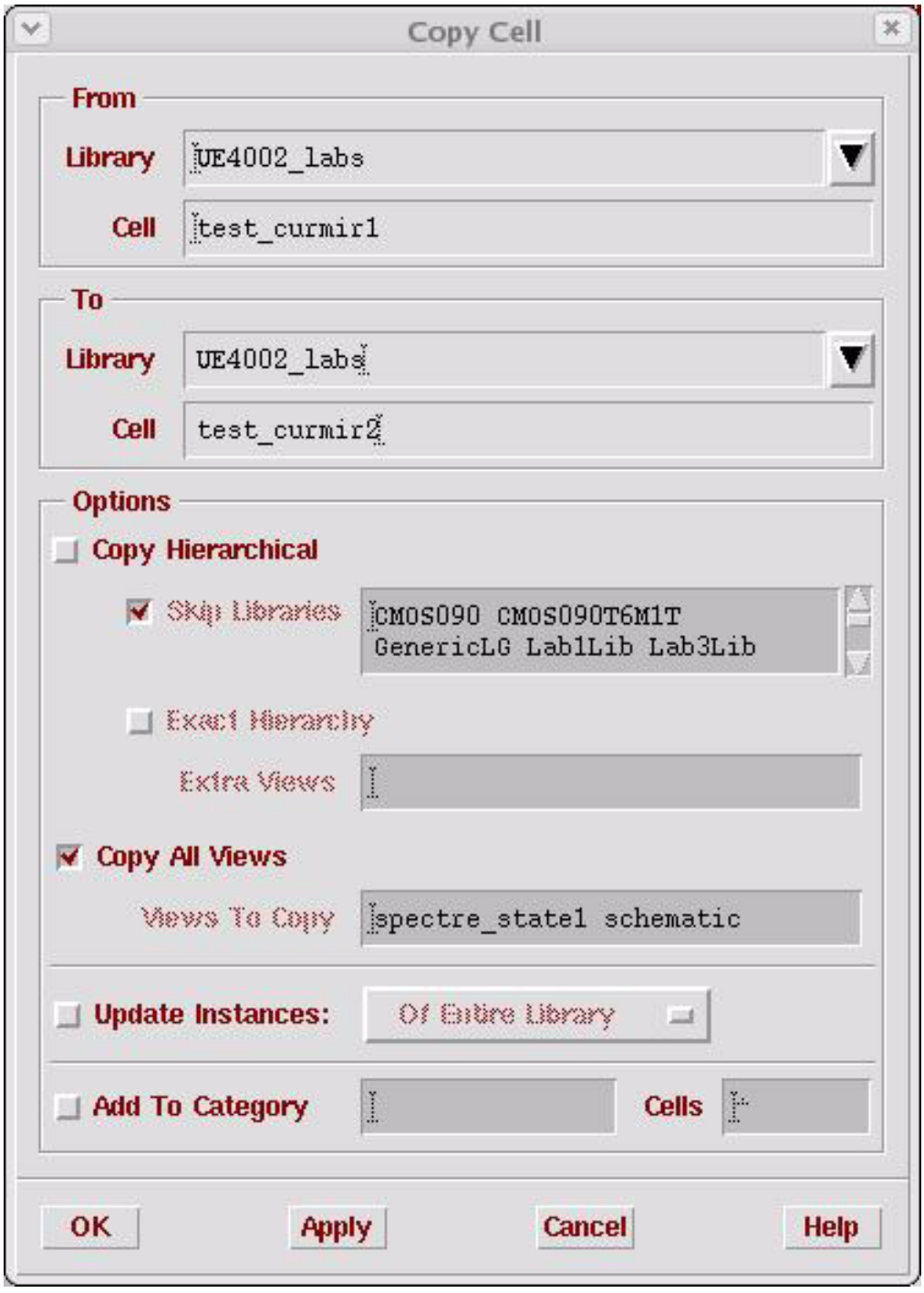


Figure 40: Copy Cell window

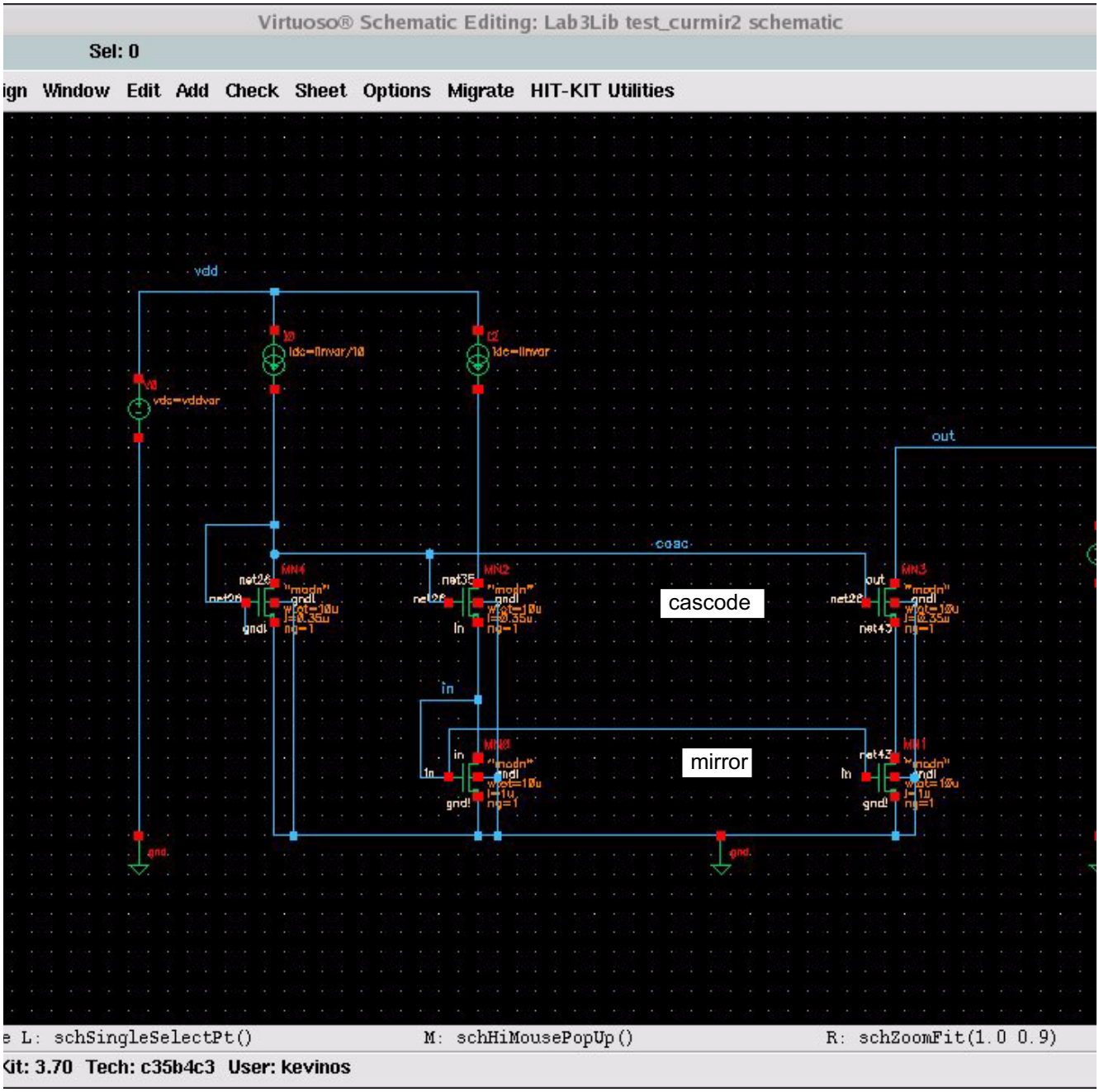


Figure 41: test_curmir2 schematic.

5 Lab 4 - Simulation and compensation of a negative feedback circuit.

This lab deals with the simulation and compensation of a negative feedback circuit.

Log on to a Sun terminal.

Open a terminal. (right-click on workspace and open Tools/Terminal...)

Navigate to your 4002 lab directory

> cd UE4002_labs

Exercise 26: Setting up the df2 library.

The design for Lab4 has been emailed to you in the form of a tar file. This needs to be extracted:

[1] cd to the directory in which you start cadence

[2] Enter the following commands to extract the cadence design library in your own user area:
> tar -xvf Lab4Lib.tar

[3] Edit your cds.lib file using a text editor and add a reference to the Lab4Lib library. This needs to be in the form similar to the other libraries defined in the cds.lib file i.e.
DEFINE <libname> <libpath>
*[**Alternatively**, after opening cadence use LM/Edit/Library Path... to open the Library Path editor and add the Library name and the Path (see Figure 42, use own username obviously). Save with File..Save]*

[4] Start up cadence with
> ams_cds -tech c35b4 -mode fb

[5] Check that Lab4Lib is visible in your library manager.
If it is not, check the path in your cds.lib file and re-load the cds.lib file from the library manager (LM/View/Refresh)

[6] In the Lab4Lib library open the schematic test_nmos_1stage_amp. Open ADE with SC/Tools/Analog Environment.
Open ADE/Session/Load State, select Cellview option and load the spectre_ac state.

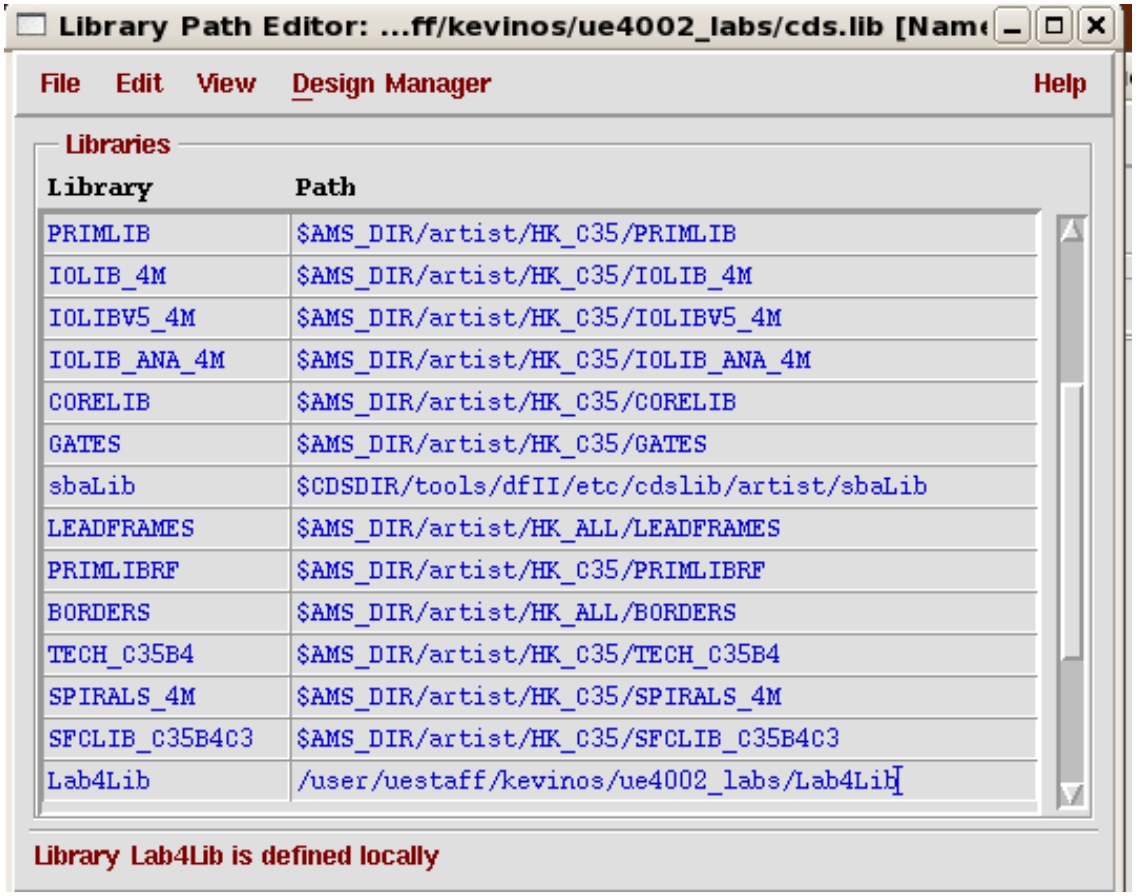


Figure 42: Library Path Editor

Some things to note about the design:

[i] The circuit (similar to one treated in the course, and shown in Figure 43) features a simple opamp (more correctly an operational transconductance amplifier or ota), and a source follower stage which uses negative feedback to copy a reference voltage (in this case 1.2V e.g. from a bandgap voltage reference) across a resistor Rext. The intention is to generate an accurate current of Vref/Rext which can then be mirrored to other circuits. MP6 is the input part of the PMOS current mirror which copies the reference current generated by the circuit.

[ii] In order to check the stability of the negative feedback loop, the loop has been 'cut' between the output and the feedback to the opamp. This is done using the sp1tswitch component from analogLib. If you check the properties of this component you will see that the switch is closed for DC, and open for AC. This means that the switch will act as a short-circuit for a DC analysis, and as an open circuit for an AC analysis.
The simulator can then reach a DC solution with the feedback in place, and using that DC solution can calculate the gain around the loop in the AC analysis with the feedback cut.
Note that a large inductor would perform the same function as the switch i.e. give zero impedance at DC and high impedance for high frequencies which would effectively cut the loop.

In order to measure the gain around the loop, an AC source is added in series with another sp1tswitch, which is open for DC but shorted for AC. This AC source injects signal into the loop at INN, and the loop gain is the gain from the INN node to the OUT node.
A large capacitor would perform the same function as the switch, i.e. give infinite impedance at DC, and low impedance at higher frequencies.

[iii] When the feedback is in place, the output node 'sees' the input capacitance of the opamp. When the loop is cut, this capacitance needs to be added to the output node to replicate the real closed loop situation. This is the function of the second dummy opamp.

[iv] All NMOS transistors have the bulk connected to ground as this technology does not have isolated Pwells as a standard feature. This of course affects the gain of the source follower.

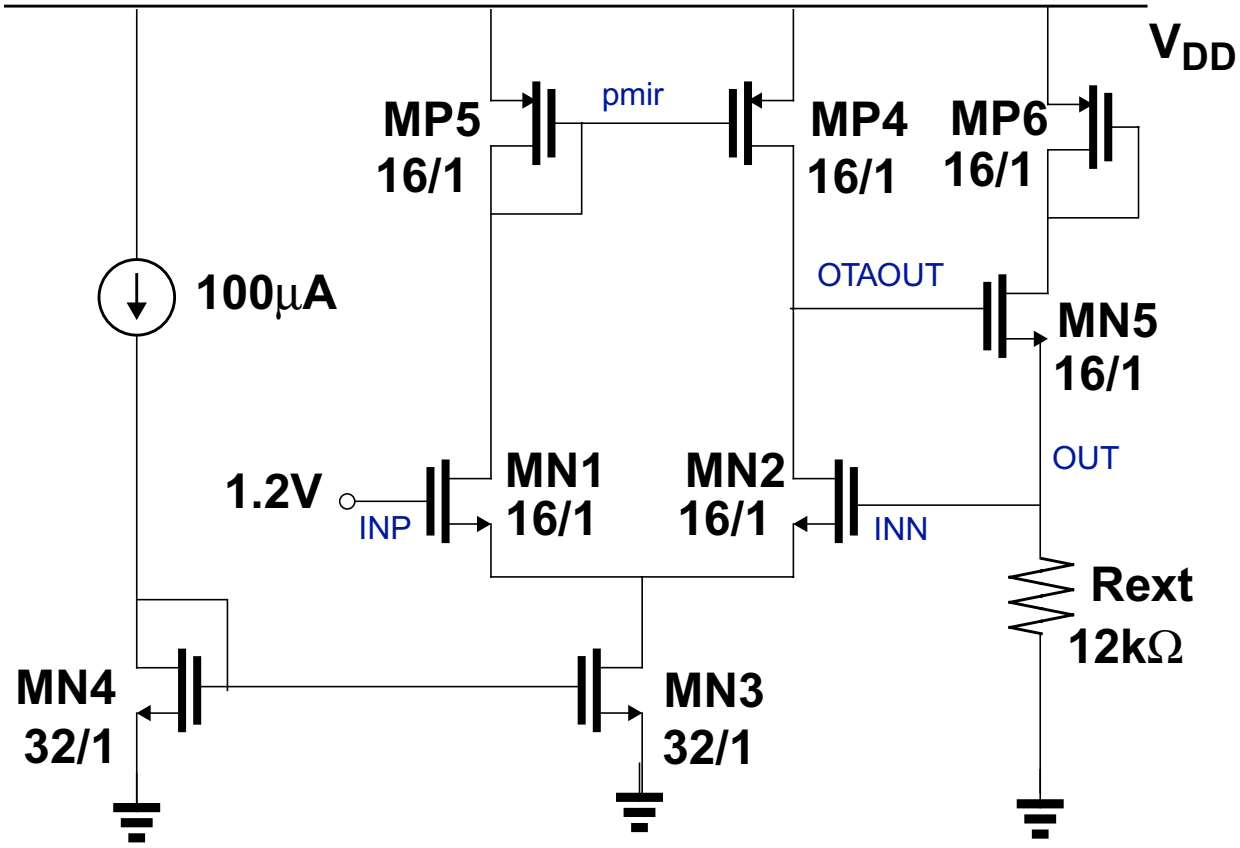


Figure 43: Circuit drawing for Lab 3. INN and OUT are the same node (but are split in the AC analysis when the loop is cut).

Exercise 27: Checking the dc solution and the loop gain.

In this exercise the gain of the ota and source follower are checked against the simple hand calculations using the operating parameters.

[1] Check in the ADE Design Variable section that vddVAR(3.3), VrefVAR (1.2), ibias (100u), cload (11p) and ccomp (1a) have been correctly set.
Note 1aF is a negligibly small value for ccomp. Setting the value of a component to zero may cause simulation problems.
If the analyses have not loaded correctly, then set up the DC and AC analyses:
Choose ADE/Analyses/Choose, In the Choosing Analyses form click on dc, Save Operating Point and Enabled and OK the form.
Choose ADE/Analyses/Choose, In the Choosing Analyses form click on ac, set Sweep Start to 1, Sweep Stop to 100T, set sweep type to logarithmic and set Points per Decade to 100.

[2] Run the simulation with ADE/Simulation/Netlist and Run.

[3] Using ADE/Results/Print/DC Operating points check that each of the transistors is in saturation (i.e. check that vds is larger than vdsat). Record in Appendix D which transistor has the least headroom.
Note: recall that mirror input transistors i.e. those with the gate and drain shorted are in saturation by construction and do not need to be checked.

[4] The ADE output section should give the operating points for various transistors from the DC analysis, and the low-frequency gain of the ota (A0_ota) and of the complete loop (A0) from the AC analysis. Calculate the expected gain from the ota and from the source-follower and compare with the AC results.
Record your answers in Appendix D.

[5] Check also the bandwidth (f3dB output function) and the unity gain frequency (fu).
Refer to the Exercise worksheet in Appendix D and fill in the required values in Table 18

Exercise 28: Checking the stability of the uncompensated amplifier.

Note that the load capacitance is set to 11pF. This value is derived from the following requirements:

[i] The 12k resistor is an accurate external resistor. Therefore the OUT node has to go off-chip. The bondpad, ESD structure and package pin will have a certain capacitance, which is taken here to be 1pF.

[ii] For Silicon evaluation and test it is desirable that the output node can be probed with a standard 10pF probe without making the opamp unstable.

[iii] The stability requirement is therefore that the circuit be stable (i.e. have a good phase margin, which we take here to be greater than 60^o) for a total capacitive load of 11pF, and also for 1pF.

[1] Check the phase margin from the Phasemargin output function and record this in Appendix D.
Confirm that this is the phasemargin by manually checking on the gain, phase plot that this is the value of phase when the gain just goes below 0dB. [1]

[2] Change cload to 1p, re-run the simulation and check the phase margin.
Refer to the Exercise worksheet in Appendix D and fill in the required values in Table 19.

1. Note that in the plot that phase starts at 180^o for low frequencies. So another 180^o phase shift (i.e. when the phase plot goes to 0^o) will cause instability.
Note also that the Spectre phaseMargin function (which is accessed in the calculator Special Functions section) assumes a phase which is 0 degrees at low frequencies. For this reason the PhaseMargin function in this artist window has been set up as follows phaseMargin(-VF("/OUT")).

Exercise 29: Compensating the circuit

First it is necessary to analyse the circuit and see where the poles are. There are three possibilities:

[i] In the ota, at the pmir node there is a pole at

$$f_{ppmir} \approx \frac{g_{mMP3}}{2 \times \pi \times C_{pmir}}$$

[ii] At the otaout node there is a pole at

$$f_{poutaout} \approx \frac{g_{dsMN2} + g_{dsMN4}}{2 \times \pi \times C_{otaout}}$$

[iii] At the out node there is a pole at

$$f_{pout} \approx \frac{g_{mMN5} + g_{mbMN5} + \frac{1}{R_{ext}}}{2 \times \pi \times C_{out}}$$

There are also a few zeroes in the circuit caused by the gate-drain over lap capacitance of MN2, and the gate-source capacitance of MN5.

Given the relative values of gm and gds, the dominant pole is likely to be at otaout. This is the best point at which to add capacitance. This will reduce the frequency of the dominant pole, and increase the phase margin by ensuring the gain reaches 0dB more before the total phase around the loop reaches 360^o (including the 180^o phase shift from the negative feedback).

Note that there is no opportunity to use the Miller effect to increase the effective value of the added capacitance.

[1] Set cload to 11pF. Increase the value of ccomp, re-run the simulation and check the phasemargin. Do this iteratively until the phase margin is somewhere between 60^o and 65^o. Record the phase margin and the value of the capacitance in Appendix D.

[2] Set cload to 1pF, re-run the simulation and check the phasemargin.
Refer to the Exercise worksheet in Appendix D and fill in the required values in Table 20
Print out the plot of gain and phase which should be handed up with the worksheet in Appendix D.

Exercise 30: Transient analysis

Here a small pulse signal is input to the circuit and the response and overshoot of the uncompensated and compensated amplifiers are checked.

[1] Open the test_nmos_1stage_amp_tr schematic. Open Analog Design Environment and load state_tr- (Open ADE/Session/Load State, select Cellview option and load the spectre_tr state).
If the analysis has not loaded correctly, then set up the transient analysis:
Choose ADE/Analyses/Choose, In the Choosing Analyses form click on tran, set the stop time to 10u, set Accuracy Defaults to Conservative click on Enabled and OK the form.

[2] Run the simulation, and note the ringing on the output signal.
Record in Appendix D the percentage overshoot of the output signal.

[3] Change ccomp to the value needed for a phase margin between 60^{o} and 65^{o}.
Re-run the simulation, and note the ringing on the output signal.
Refer to the Exercise worksheet in Appendix D and fill in the required values in Table 21
Print out the plot which should be handed up with the worksheet in Appendix D.

Exercise 31: Stability analysis (stb)

While the process of opening the loop and adding dummy loads is reasonably straightforward in this case, it becomes quite complicated in more complex amplifiers like fully-differential amplifiers with common-mode feedback where there are two loops and both common-mode and differential feedback loops needs to be stabilised.Furthermore the addition of the dummy load to faithfully reproduce the load seen at the point where the loop is cut can be troublesome.

To ease the task of stability analysis Cadence introduced a stability analysis tool (stb analysis). The theory behind the tool is not discussed here, but it has been proven to give accurate results and is much easier to set up than the traditional method. A disadvantage is that while it is easy to assess the gain and phase of the complete loop, the gain or phase of individual stages of the loop cannot be assessed.

[1] Open the test_nmos_1stage_amp_stb schematic. Open Analog Design Environment and load state_dcstbtr
(Open ADE/Session/Load State, select Cellview option and load the spectre_dcstbtr state).
If the analyses have not loaded correctly, then set up the stb analyses:
Choose ADE/Analyses/Choose, In the Choosing Analyses form click on stb, set the options as shown in Figure 44 and OK the form.

Note the dc voltage source V6 in the feedback loop, with zero DC voltage (i.e. just a short). This is the only addition to the circuit required for stb analysis.

[2] Open the stability analysis form (Select ADE/Analyses/Choose... and click on stb). The form should appear as in Figure 44 (though stop frequency may be different). Note the settings.
For now you do not need to take any action but note that to enable the stb analysis you need to select the voltage source in the loop by clicking Select in the form and then selecting V6 as the Probe Instance in the schematic.
You will need to do this in the assignment.

[3] Run the simulation.

[4] When the simulation completes note that various outputs have been set up in the ADE Outputs window e.g. phase margin, gain margin and low-frequency loop gain (LGdB20_1). Record the values for the Phase Margin and Gain Margin and low-frequency loop gain (which can also be read from the Loop Gain 20dB plot). Note that the values are almost identical to those previously found in the ac loop analysis.

[5] Change the value of Ccomp to that previously found to give a phase margin between 60 and 65 degrees.

[6] Re-run the simulation and record the values for the Phase Margin and Gain Margin.

[7] ***Refer to the Exercise worksheet in Appendix D and fill in the required values in Table 22***

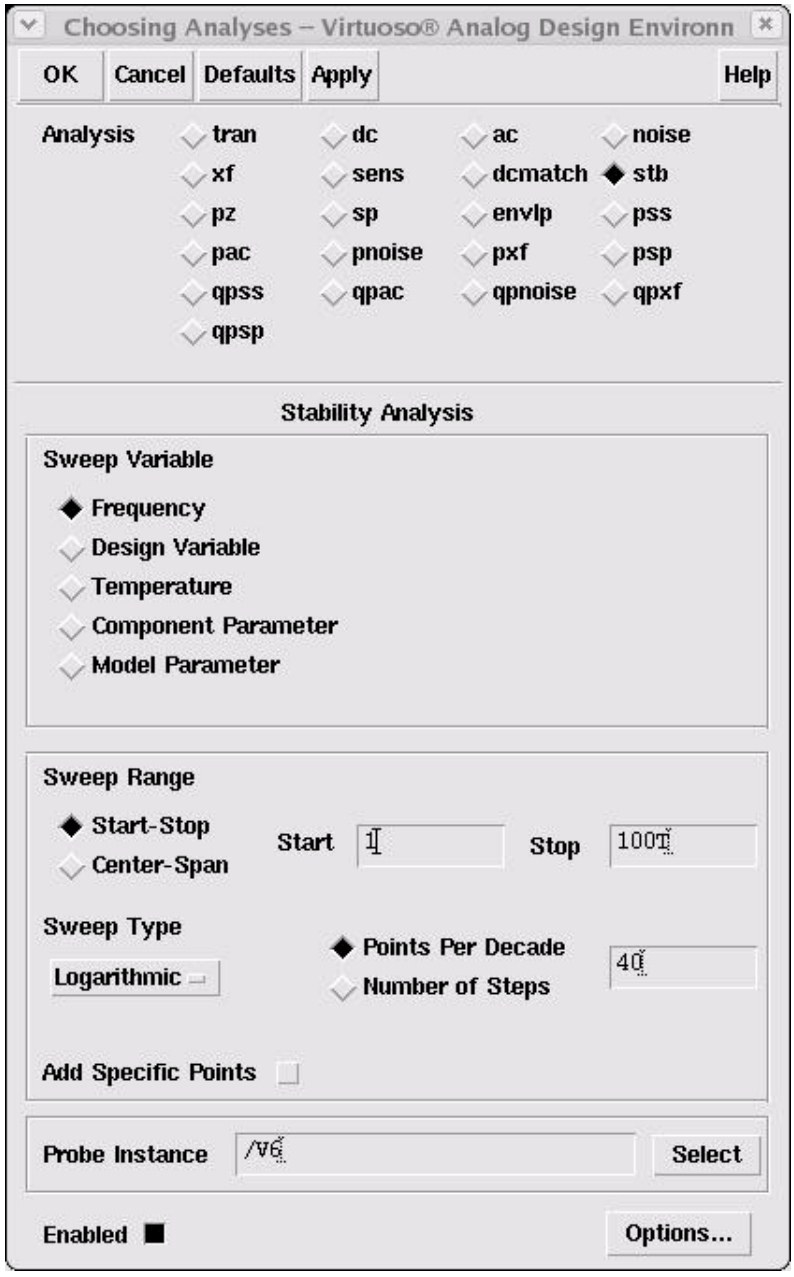


Figure 44: Stb analysis

The following may be of use in the assignment:

The outputs which were set-up in the artist state for the stb analysis have been done so using the Direct Plot form (Select ADE/Results/Direct Plot/Main Form...). A form similar to that shown in Figure 45 appears. If you click on phase margin you will see that the phase margin appears automatically. Similarly if you select Gain Margin or any of the other quantities. Clicking on Add to Outputs will add the measurement to the ADE Outputs window.

Use the Loop Gain option to set up gain and phase plots.

Clicking on Stability Summary in the Direct Plot form will show the values for the Phase Margin and Gain Margin

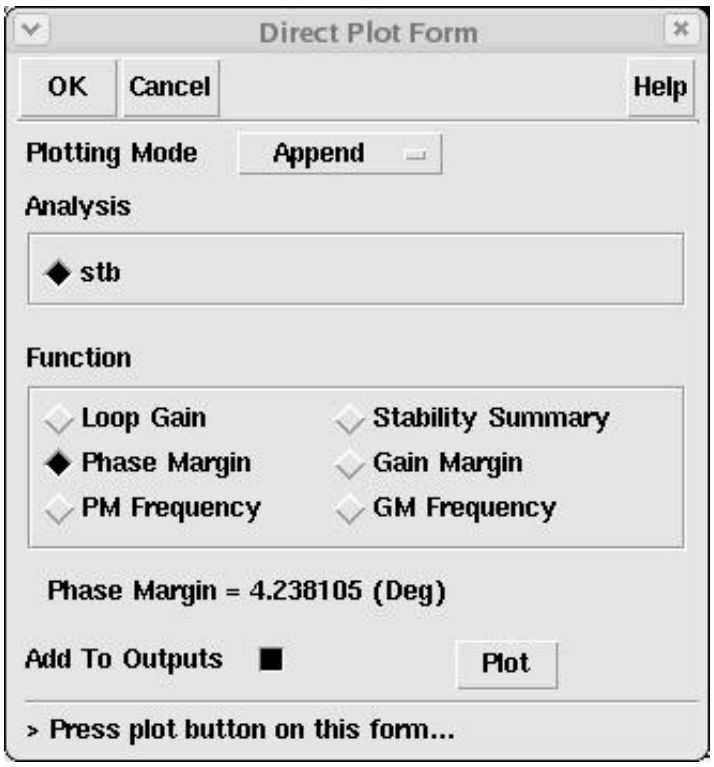


Figure 45: Direct Plot Form.

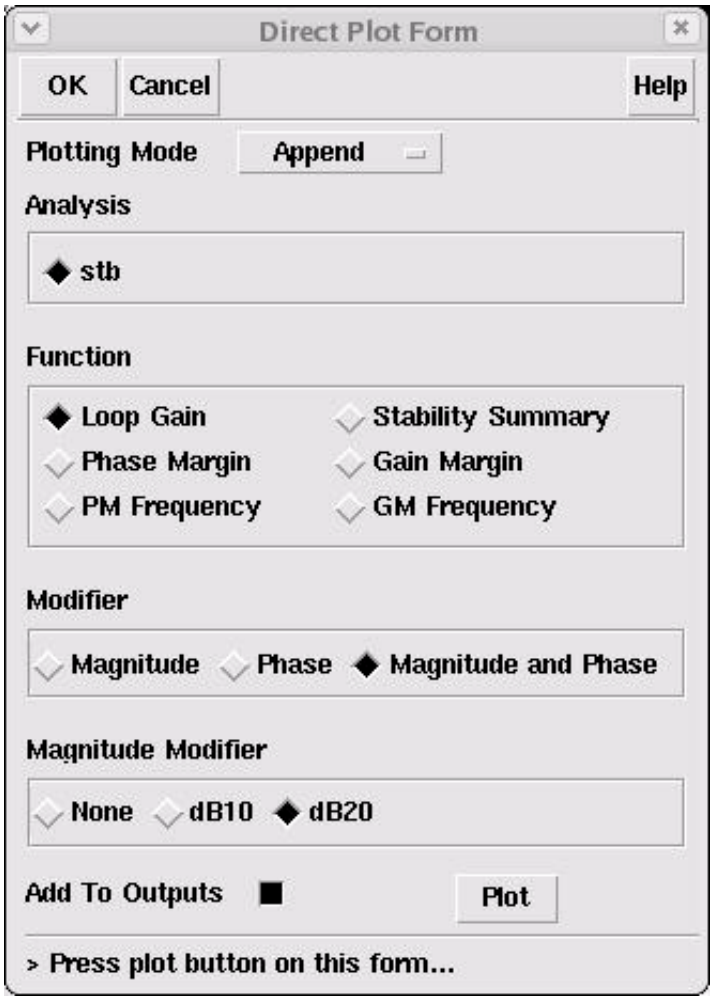


Figure 46: Direct Plot form - setting up the loop gain and phase plots.

Appendix A: Exercise Worksheet Lab 1

Fill in the values requested below for the various exercises.

Exercise 8: Running a DC simulation.

Record the following for the NMOS transistor:

Small-signal parameter	Note	Value
betaeff	corresponds to k_n' x W/L in course	
gm	transconductance	
gds	output conductance	
ids	drain-source current	
vgs	gate-source voltage	
vth	threshold voltage	
vds	drain-source voltage	
vdsat		

Table 3: Small signal parameters -1

Verify that the transistor is in saturation (i.e. show that vds>vgs-vth) and enter the headroom (headroom = vds-(vgs-vth)) in Table 4.

Calculate the current predicted by the equation below and enter this into Table 4.

$$I_D = \frac{betaeff}{2}(vgs - vth)^2$$

Calculate the low-frequency small-signal gain of the circuit using the equation below and enter this into Table 4.

$$Gain = \frac{g_m}{g_{ds} + \frac{1}{R_L}}$$

parameter	Note	Value
headroom		
I_D		
Gain		

Table 4: Small signal parameters 2

Exercise 9: Running an AC simulation

Note that the plot shows the frequency response contains 1 pole and 1 zero.

From the plot, read the value of the low-frequency gain (i.e. Gain at 1Hz), the gain at 1THz, the pole frequency (i.e. the frequency at which the gain has dropped 3dB) and the zero frequency (i.e. the frequency at which the gain is 3dB above the gain at 1THz) from the graph and enter these values in Table 5.

Calculate the total load capacitance and enter this into Table 5. Use the following equation

$$f_p = \frac{1}{2\pi} \cdot \frac{g_{ds} + \frac{1}{R_L}}{C_{load}}$$

Calculate the gate-drain capacitance of the nmos transistor and enter this into Table 5. Use the following equation

$$f_z = \frac{1}{2\pi} \cdot \frac{g_m}{C_{gd}}$$

parameter	Note	Value
Low-frequency gain	Gain at 1Hz	
Gain at 1THz		
Pole frequency	frequency at which the gain has dropped 3dB from the value at 1Hz.	
Zero frequency	frequency at which the gain is 3dB above the gain at 1THz	
Load capaci-tance		
Gate-drain capacitance		

Table 5: AC Plot calculations

Exercise 10: Setting Outputs

Record the value of the low-frequency gain and the bandwidth (as given by the output functions) in Table 6 below

Print out the plot which should be handed up with the worksheet in Appendix A.

parameter	Note	Value
GaindB		
BW3dB		

Table 6: AC Output values

Exercise 11: Running a transient simulation

Record the value of the output signal amplitude and of the gain as read from the transient plot.
Print out the plot which should be handed up with the worksheet in Appendix A.

parameter	Note	Value
Output signal amplitude		
Gain		

Table 7: Transient Output values

Exercise 12: Transient simulation with large input signal

Record the value of the output signal amplitude and of the gain as read from the transient plot.

parameter	Note	Value
Output signal amplitude		
Gain		

Table 8: Transient Output values

This page is intentionally left blank.

Appendix B: Exercise Worksheet Lab 2

Fill in the values requested below for the various exercises.

Exercise 16: Using variables in an analysis.

Record the following for the NMOS transistor:

Parameter	Note	Value
gm	transconductance	
gds	output conductance	
ids	drain-source current	
DC output voltage		

Table 9: Parameters from output function

Exercise 17: Running a DC sweep analysis (1).

Record the following from the plot:

Parameter	Note	Value
Input voltage at which headroom goes negative		
gm	at this voltage	
gds	at this voltage	
ids	at this voltage	
Maximum gain	from the derivative function	
Input voltage at which max. gain occurs		

Table 10: DC Sweep analysis values - 1

Exercise 18: Running a DC sweep analysis (2).

Record the following from the plot:

Parameter	Note	Value
Maximum gain	from the derivative function	
Input voltage at which max. gain occurs		

Table 11: DC Sweep analysis values - 2

Print out the plot which should be handed up with the worksheet in Appendix B.

Exercise 19: Running an AC Sweep Analysis with the input bias voltage Design Variable as swept parameter

Record the following:

Parameter	Note	Value
Maximum gain		
Input voltage at which max. gain occurs		

Table 12: AC Sweep analysis values - 1

Exercise 20: Running an AC Sweep Analysis with the resistive load Design Variable as swept parameter

Record the following:

Parameter	Note	Value
Maximum gain		
Resistive load which max. gain occurs		

Table 13: AC Sweep analysis values - 2

Print out the plot which should be handed up with the worksheet in Appendix B.

Appendix C: Exercise Worksheet Lab 3

Exercise 22: DC Simulation of a simple current mirror

Parameter	Note	Value
Mirror input voltage	with vout =1.5	
Mirror output current	with vout =1.5	
Percentage mirror inaccuracy	with vout =1.5	
vdsat operating point parameter of the output nmos	with vout =1.5	
Mirror input voltage	with vout = vdsat	
Mirror output current	with vout = vdsat	
Percentage mirror inaccuracy	with vout = vdsat	
Mirror input voltage	with vout =1.5; W and L doubled	
Mirror output current	with vout =1.5; W and L doubled	
Percentage mirror inaccuracy	with vout =1.5; W and L doubled	

Table 14: DC Sweep analysis values - 2

Exercise 23: DC sweep of a simple current mirror

Record the following:

Parameter	Note	Value
Minimum output voltage for which the output current is within 5% of the input current:		

Table 15: DC Sweep of a simple current mirror

Print out the plot which should be handed up with the worksheet in Appendix C.

Exercise 24: Design a cascode current mirror.

vdsat of the mirror output and cascode transistors:

Output current:

Record the following:

Parameter	Note	Value
Output current		
vdsat of the mirror output transistor		
vdsat of the output cascode transistor		
W and L of the transistor which sets up the cascode voltage	this is MN11 in Figure 40.	

Table 16: Design of a cascode current mirror.

Exercise 25: DC sweep of a cascode current mirror

Record the following:

Parameter	Note	Value
Minimum output voltage for which the output current is within 5% of the input current:		

Table 17: DC Sweep of a cascode output mirror

Print out the plot which should be handed up with the worksheet in Appendix C.

Appendix D: Exercise Worksheet Lab 4

Exercise 27: Checking the dc solution and the loop gain.

Check which transistor has the least headroom and enter in Table 18.

Calculate the expected low-frequency gain from the ota using the following expression and enter in Table 18.

$$A_{ota} \approx \frac{g_{mMN2}}{g_{dsMN2} + g_{dsMP4}}$$

Enter in Table 18 the low-frequency gain of the ota as given by the AC analysis

Calculate the expected gain from the source follower using the following expression and enter in Table 18.

$$A_{sf} \approx \frac{g_{mMN5} R_L}{1 + (g_{mMN5} + g_{mbMN5}) R_L}$$

Enter in Table 18 the low-frequency gain of the source follower as given by the AC analysis?

Enter in Table 18 the bandwidth and unity gain given by the AC analysis

What is the unity gain frequency (fu)?

Parameter	Note	Value
Transistor with least headroom	vdsat	
Low-frequency ota gain	from calculation	
Low-fequency ota gain	from AC analysis	
Low-frequency source-follower gain	from calculation	
Low-fequency source-follower gain	from AC analysis	
Bandwidth	f3dB output function	
Unity gain	fu output function	

Table 18: DC solution and loop gain

Exercise 28: Checking the stability of the uncompensated amplifier.

Enter the required values in Table 19

Parameter	Note	Value
phase margin	cload = 11pF	
phase margin	cload =1pF	

Table 19: Stability of uncompensated amplifier.

Exercise 29: Compensating the circuit

Enter the required values in Table 20

For the bandwidth calculation, check that the bandwidth corresponds roughly with the value of the pole given by

$$f_{poutaout} \approx \frac{g_{dsMN2} + g_{dsMP4}}{2 \times \pi \times C_{comp}}$$

Parameter	Note	Value
phase margin	cload = 11pF, phase margin should be between 60^o and 65^o	
ccomp	to achieve above phase margin	
bandwidth	under above conditions, from f3dB function.	
Bandwidth	from calculation	
phase margin	cload=1pF, ccomp as given above	

Table 20: Compensated amplifier.

Print out the plot of gain and phase which should be handed up with the worksheet in Appendix D.

Exercise 30: Transient analysis

Enter the required values in Table 21

Parameter	Note	Value
Percentage overshoot of the output signal	without ccomp increased	
Percentage overshoot of the output signal	with ccomp increased	

Table 21: Transient analysis of amplifier.

Print out the plot of overshoot of the output signal of the compensated amplifier which should be handed up with the worksheet in Appendix D.

Exercise 31: Checking the stability of the amplifier with stb analysis.

Enter the required values in Table 22

Parameter	Note	Value
phase margin	ccomp= 1a, cload = 11pF	
gain margin	ccomp= 1a, cload = 11pF	
low-frequency loop-gain	ccomp= 1a, cload = 11pF	
phase margin	ccomp= design value, cload = 11pF.	
gain margin	ccomp= design value, cload = 11pF.	

Table 22: Stability of amplifier with stb analysis.

This page is intentionally left blank.

Appendix E: Assignment

Design an opamp with the following required specifications:

[1] DC Loop gain: >80dB

[2] Phase Margin: > 60^o

[3] Gain Bandwidth product: > 15MHz

[4] Power Dissipation target :1mW (including the input bias current). Marks will be awarded for better efficiency.

The following are the external conditions under which the required specifications are to be met:

[1] Supply voltage: 3.3V.

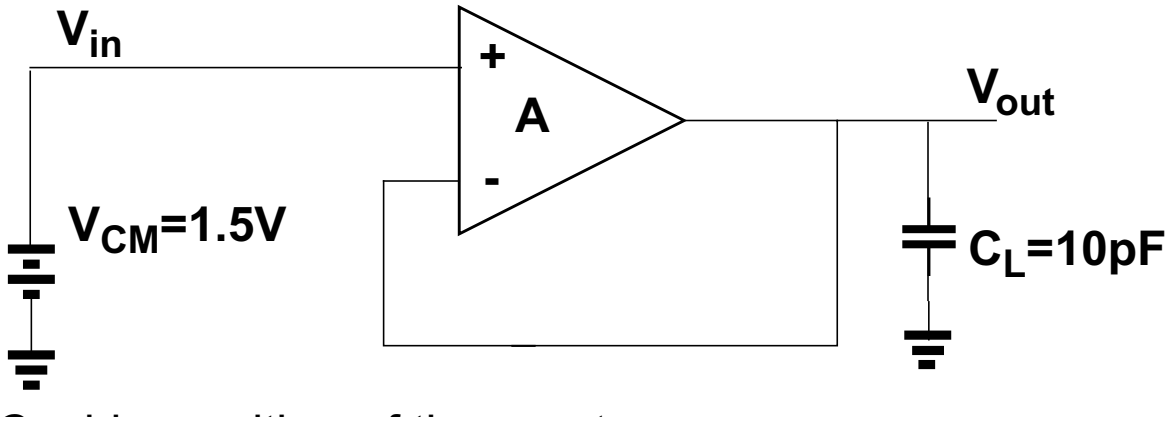
[2] Input bias current (into NMOS): 100uA.

[3] Common-mode reference voltage: 1.5V.

[4] Load: 10pF capacitance to ground.

[5] Process: c35b4 nominal (i.e. the default models).

No other supplies, reference voltages or current sources should be used.



The assignment needs to be completed individually.
The design should be completed in a library UE4002_<username>. This library should contain at least:

[1] schematic, symbol design of two-stage opamp.

[2] schematic of testbench to simulate dc operation and AC response of the opamp.
To simulate the AC response you may use either the AC analysis i.e. cutting the loop, or the stb analysis (recommended).

The artist state for this design should be stored as cellviews using theADE..Session..Save State menu (and select Cellview option in the Saving States menu). The artist state should run the dc analysis and the AC or stb analysis from 1Hz to 100GHz.
The dc analysis should include an output function to:

[i] Measure Power dissipation e.g. (IVSS = IDC("/NMOS2STAGEAMP_0/vss"))
Note: the vss terminal on the opamp will need to be saved using ADE/Outputs/To be saved/Select on Schematic. Then select the vss terminal and press Esc.
The current through vss should be measured rather than vdd so that the input bias current is included

If using the method of opening the loop along with an ac analysis, the ac analysis should include output functions to

[i] Plot gain response e.g. dB20(VF("/OUT"))

[ii] Plot phase response e.g. phase(VF("/OUT"))

[iii] Measure Phase Margin e.g. PhaseMargin= phaseMargin(-VF("/OUT"))

[iv] Measure Open-loop gain at 1Hz e.g. A0 = value(dB20(VF("/OUT")) 1)

[v] Measure Gain-Bandwidth product e.g. GBW = gainBwProd(VF("/OUT"))

If using the stb analysis you may obtain the plots and values of phasemargin and unity gain from the Direct Plot form (Select ADE/Results/Direct Plot/Main Form...)

[3] schematic of testbench to simulate the transient response of the opamp

The artist state for this design should run the transient analysis from 0 to 10us, with an 100mVinput square wave with a period of 2us as in Lab4 and include output functions to:

[i] Plot the transient response e.g. VT("/OUT")

[ii] Measure the overshoot e.g. overshoot(VT("/OUT") 2.5e-06 t 3.5e-06 t nil nil).

The results of your work should be documented in a short report containing:

[i] Motivation for the opamp architecture shown

[ii] Description of how the design was done and optimised

[iii] Schematic plots of the testbench(es).

[iv] Schematic plot of the opamp

[v] Plots showing the loop gain and phase

[vi] Plots showing the transient step response (show input and output)

[vii] Table with the following results: Current consumption, DC loop gain, loop gain 3dB bandwidth, phase margin, gain margin, Gain Bandwidth Product, Step response percentage overshoot.

The completion deadline for submission of the report is **TBD** (check with the lab instructor).

After the deadline a time may be arranged for each student to go through the design on a one-to-one basis, run the simulation and verify that the results meet the requirements, and correspond to the filled-in values.

Note on copying cadence libraries:

The easiest approach for generating the initial design and testbench schematics is to copy the Lab4 library Lab4Lib and use the designs and testbenches in this lab as a starting point for the assignment.
To do this right-click on Lib4Lab in the LM, and select Copy. A menu as in Figure 47 will appear. Fill in the name of the new library (change username obviously), and click **Update Instances**. OK the form

A New Library menu will appear (Figure 48). Fill in the unix location of the new library (/home/<username>/UE4002_<username>) and OK the form

After Cadence has copied the library a 'Technology File for New Library' menu appears. Select 'Attach to an existing techfile' and OK. An Attach Design Library to Technology menu appears - select TECH_C35B4 and OK this. The new library should now appear in the LM

Figure 47: Copy Library menu

Figure 48: New Library menu

Appendix F: Assignment Worksheet

The worksheet is intended as an aid only. Results should be documented in the report to be submitted to the lab instructor..

Assignment 1: Power dissipation of the amplifier designed for the assignment.

Enter the required values in Table 23

Parameter	Note	Value
Power dissipation.		

Table 23: Power dissipation of the amplifier from dc analysis.

Assignment 2: Stability of the amplifier designed for the assignment.

Enter the required values in Table 24

Parameter	Note	Value
Loop gain	If using stb analysis you may read this value from the Loop Gain 20dB plot. (Select ADE/Results/Direct Plot/ Main Form...) and then select stb, Loop gain and Plot.	
phase margin	If using stb analysis you may get this value from the stability summary (Select ADE/Results/Direct Plot/Main Form...) and then select stb and Phase Margin.	
Unity gain frequency or Gain-Bandwidth product	If using stb analysis you may get this value from the stability summary (Select ADE/Results/Direct Plot/Main Form...) and then select stb and PM frequency.	

Table 24: Stability of the amplifier from dc analysis.

Assignment 3: Transient analysis

Enter the required values in Table 25

Parameter	Note	Value
Percentage overshoot of the output signal		

Table 25: Transient analysis of amplifier.