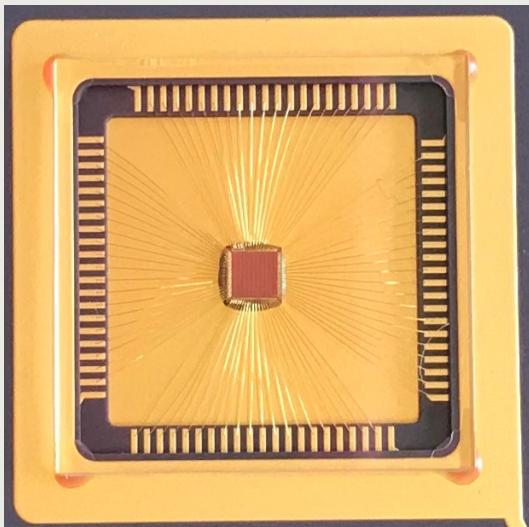


Fachhochschule Dortmund

University of Applied Sciences and Arts



Tutorial First Steps With Cadence

Author:

Hossein Tavakoli

hossein.tavakoli@fh-dortmund.de

Contents

1	Introduction	5
2	First steps	6
2.1	Preparatory configuration steps	7
2.2	Start of Cadence	8
3	Create a library	9
4	The circuit diagram editor	12
4.1	The editor	12
4.2	Toolbar elements in the circuit diagram editor	13
5	Schematic and Cellview	14
5.1	Construction of the first circuit	14
5.2	Dealing with faults in the circuit	17
5.3	Editing component properties and creating variables	18
5.4	Use of labels	20
5.5	Complete closure of the libraries used	21
6	Setting up a simulation	22
6.1	The ADE Explorer main window	22
6.2	Importing variables from the schematic	23
6.3	DC simulation to display the operating point	24
6.4	Display of the operating point	26
6.5	Creating simulation outputs from the schematic	29
6.6	Creating outputs using the calculator	31
6.7	Saving and loading ADE Explorer states	33
7	DC analysis in ADE Explorer	35
7.1	Gate-source voltage sweep	35
7.2	Variation of transistor width	38
7.3	Variation of channel length	40

7.4 Sweep of the drain-source voltage	41
7.5 Variation of channel width and channel length of the transistor	43
7.6 Consideration of the threshold voltage and the body effect	45
7.7 Temperature influence on threshold voltage and charge carrier mobility	48
8 Settings in the Visualisation & Analysis window	50
8.1 Use of markers in the simulation window	50
8.2 CSV-Exports generation	51
8.3 Export of plots in image form	51
9 Structure of an amplifier	54
9.1 The inverting amplifier	54
9.2 Creation of pins in the schematic	55
10 Use of symbols	57
10.1 Creating a symbol	57
10.2 Use of a symbol in a schematic	60
10.3 Display/edit a symbol in the schematic	61
11 Transient analysis in ADE Explorer	62
12 AC analysis in ADE Explorer	64
12.1 Preparing the circuit for an AC analysis	64
12.2 Configuration of the AC analysis	64
13 DC analysis in ADE Assembler	71
13.1 Starting ADE Assembler	71
13.2 Optimization of the simulation speed	72
13.3 Conversion of the circuit to an inverter	75
13.4 Configuring a DC analysis in ADE Assembler	75
13.5 Sweeps via global variables in ADE Assembler	80
14 The corner simulation	83
14.1 Setting up a corner simulation	83

14.2 Specs in a corner simulation	85
15 The Monte Carlo simulation	87
15.1 Setting up a Monte Carlo simulation	87
15.2 Limit values (specs) in a Monte Carlo simulation	90
16 Layout	92
16.1 Introduction	92
16.2 Elements of the Layout	92
16.3 The Virtuoso Layout Editor	94
16.4 The Layer Window	95
16.5 Test Run	95
16.5.1 Preparation	95
16.6 Insert Instances	97
16.7 Early Design Rule Check	100
16.8 Routing and labels	104
16.9 Layout-vs.-Schematic	105
17 Appendix	107
17.1 The ADE-L main window	107
17.1.1 Importing variables from the schematic	108
17.1.2 DC simulation to display the operating point	109
17.1.3 Display of the operating point	111
17.1.4 Creating simulation outputs from the schematic	114
17.1.5 Creating outputs using the calculator	116
17.1.6 Saving and loading ADE-L states	119
17.2 DC analysis in ADE-L	120
17.2.1 Gate-source voltage sweep	120
17.2.2 Variation of transistor width	124
17.2.3 Variation of channel length	126
17.2.4 Sweep of the drain-source voltage	127
17.2.5 Variation of channel width and channel length of the transistor	129
17.2.6 Consideration of the threshold voltage and the body effect	131

17.2.7	Temperature influence on threshold voltage and charge carrier mobility	134
17.3	Transient analysis in ADE-L	136
17.4	AC analysis in ADE-L	138
17.4.1	Preparing the circuit for an AC analysis	138
17.4.2	Configuration of the AC analysis	138
17.5	DC analysis in ADE-XL	144
17.5.1	Starting ADE-XL	144
17.5.2	Optimization of the simulation speed	145
17.5.3	Conversion of the circuit to an inverter	148
17.5.4	Configuring a DC analysis in ADE-XL	149
17.5.5	Sweeps via global variables in ADE-XL	154

1 Introduction

The following tutorial is intended to help interested students at Dortmund University of Applied Sciences and Arts get started with the Cadence Virtuoso Analogue Design Environment (Cadence for short) program from Cadence, which has established itself as the industrial standard for designing integrated circuits. The aim is to provide students with fundamental basic knowledge. The course covers the start of the software, the creation of new and the use of existing libraries and components, through to the design and analysis of own circuits. Using a simple transistor circuit, all the steps required to design and analyse the circuit in Cadence are run through.

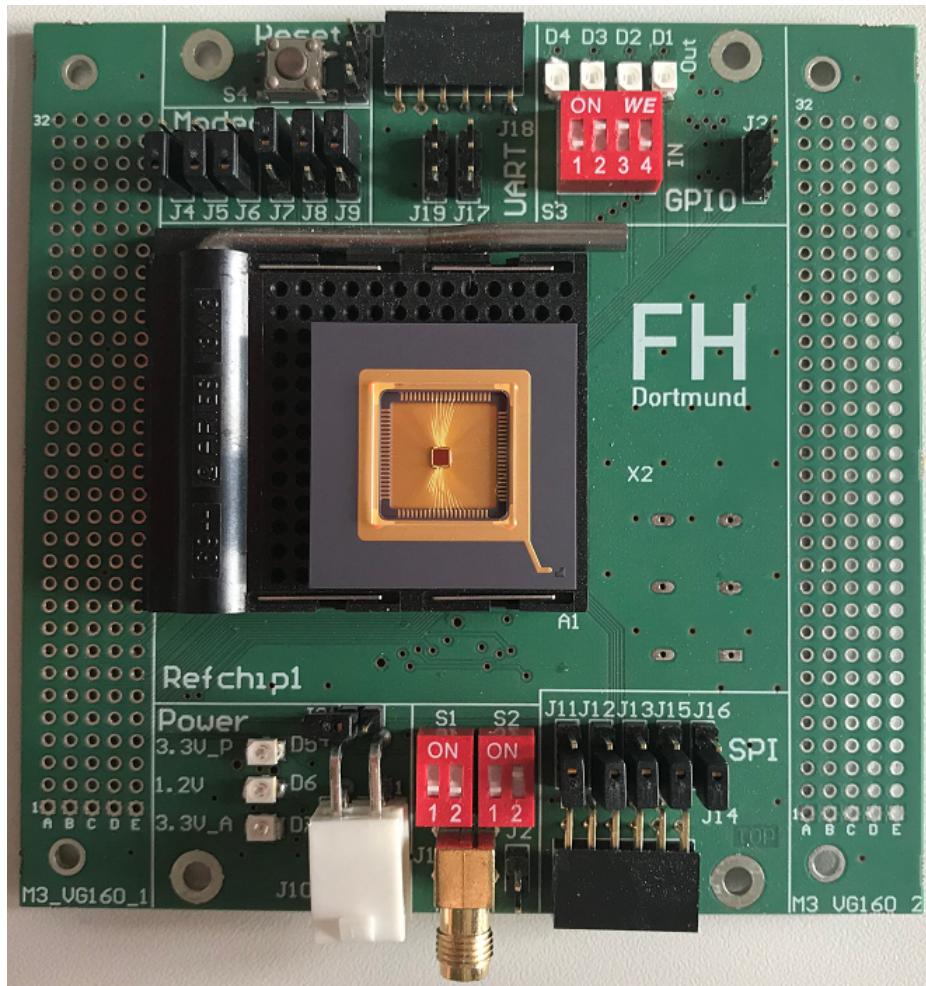


Figure 1: System-on-Chip

2 First steps

The Cadence software is a Linux application. This means that there are basically two ways to use it. One option is to log in from a Windows computer via a graphical terminal programme. Another option is to use the software locally directly under Linux. This section describes how the Cadence software can be used locally under Linux. In contrast to Windows, most programmes under Linux cannot be started via a desktop icon. Instead, a sequence of commands must be entered in the terminal each time the programme is restarted.

Applications > System tools > Terminal.

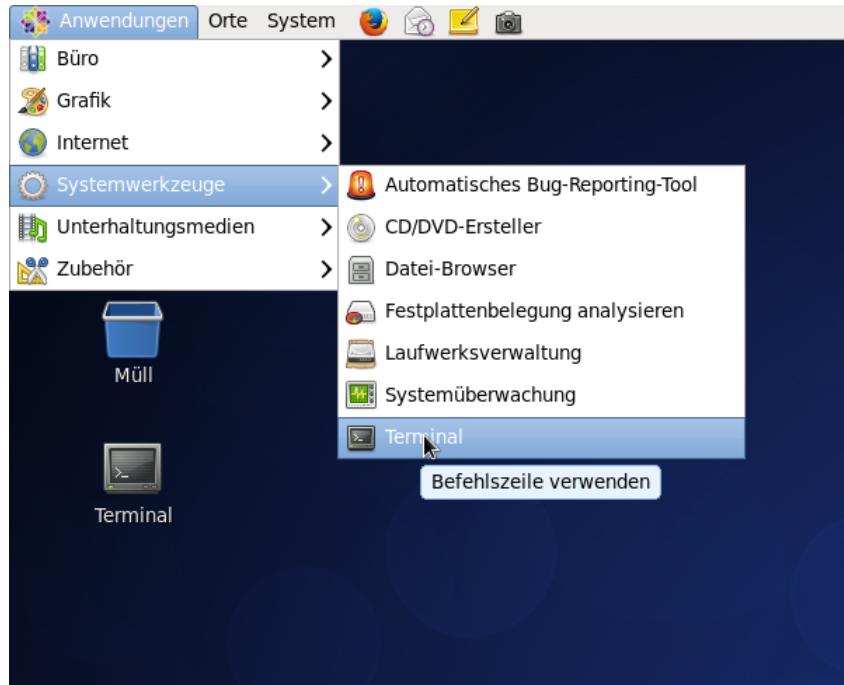


Figure 2: Start of the terminal

2.1 Preparatory configuration steps

Before Cadence can be used, the required directory structure and all personal configuration files must be created once in the home directory. The first step is to create the subdirectory *cadence*. Further subdirectories are created in this directory for each CMOS technology used. The command *mkdir* (make directory) is used to create new folders.

- **mkdir cadence**
- **ls -al**

The command *ls -al* can be used to list the contents of the current directory. The output of the command should contain the newly created library *cadence*. The command *cd* (change directory) makes it possible to change to the desired (sub)directories.

- **cd cadence**
- **mkdir umc180**
- **cd umc180**
- **pwd**

The command *pwd* shows the current directory path you are in. Make sure that the path is as follows */user/„username“/cadence*, where „username“ corresponds to your Linux user name.

Before the software can be started, configuration files and environment variables must be set. Among other things, the path to the executable files of the design software must be included in the environment variable *PATH*. Cadence also requires other environment variables, such as a variable containing the address of the licence server. An environment module is used to define environment variables, which must be loaded before each start of the software. The following command is used for this.

- **module load Cadence/2025**

The command *module avail* can be used to display all environment modules installed on the server. It is advisable to always use the latest version of the design software and load the corresponding environment module.

As a final step, technology-specific configuration files are created once and before the software is started for the first time. In these files, for example, the path to the library with the circuit symbols of the components of the technology used is defined. To simplify this, a configuration script is available that automatically generates all the required files. The following command is executed once in the directory *umc180*.

- **umc_l180**

2.2 Start of Cadence

While the steps from the previous section only need to be carried out once, the following instructions must be carried out each time the design software is started. If you have not already done so, open the terminal window, change to the directory `umc180` and load the environment package.

- `cd cadence`
- `cd umc180`
- `module load Cadence/2025`

You can then start the actual software with the command `virtuoso`.

- `virtuoso &`

The ampersand (&) after the command triggers the execution of the programme in the background. This makes it possible to enter further commands in the same terminal window even after starting the software. Without the ampersand, the terminal blocks and further command entries are no longer possible. An example command output is shown in the following illustration:

A screenshot of a terminal window titled "jzorn@matare:~/cadence/umc180". The window shows a series of terminal commands being entered and executed. Red arrows point to specific command lines: [jzorn@matare ~]\$ cd cadence, [jzorn@matare cadence]\$ cd umc180, [jzorn@matare umc180]\$ module load Cadence/2016_2017_PVS, and [jzorn@matare umc180]\$ virtuoso&. The terminal also displays some initial configuration messages and a prompt for the user to copy files to their home directory before calling up virtuoso. The bottom of the terminal shows the command history with [1] 3479 and [jzorn@matare umc180]\$.

Figure 3: Command sequence for starting Cadence

Note: To avoid having to enter these command lines in full in the console every time you start the programme, you can press the **[up arrow]** key to call up commands that have already been entered. Once you have reached the desired command, this command can be sent again by pressing **[Return]** or **[Enter]**.

3 Create a library

Entering a command in the terminal opens the *Virtuoso* console, which is the central operating element of Cadence. Other tools can be opened via the console. If, for example, the *Library Manager* does not appear automatically during startup or has been closed unintentionally in the meantime, it can be opened via the *Tools* menu > *Library Manager*. In addition, individual commands or scripts can also be entered and executed in the command prompt marked in red (see *Figure 4*). Among other things, libraries in which circuit designs and components are organised can be managed in the *Library Manager*.

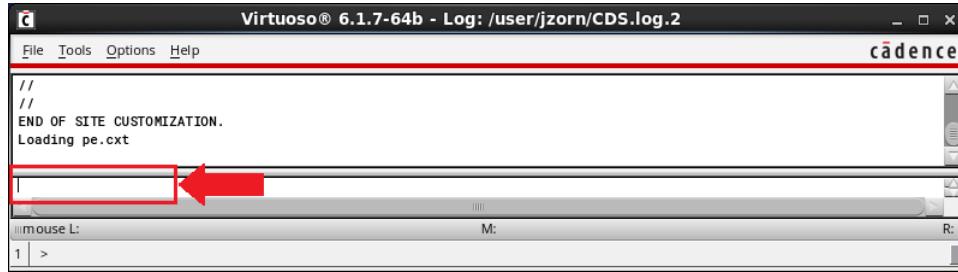


Figure 4: The Virtuoso-Window

Particularly important libraries include the *analogLib* and the *UMC_18_CMOS* library. The *analogLib* contains technology-independent and idealised components such as voltage and current sources, which can be useful for simulation. The *UMC_18_CMOS* library contains all components that can be realised in the UMC 180nm CMOS technology, such as MOS transistors of various types, but also passive components such as resistors and capacitors. After starting *Virtuoso*, you should first create a personal library in which you can create your own circuit designs. This library can be created in the *Library Manager* (see *Figure 5*).

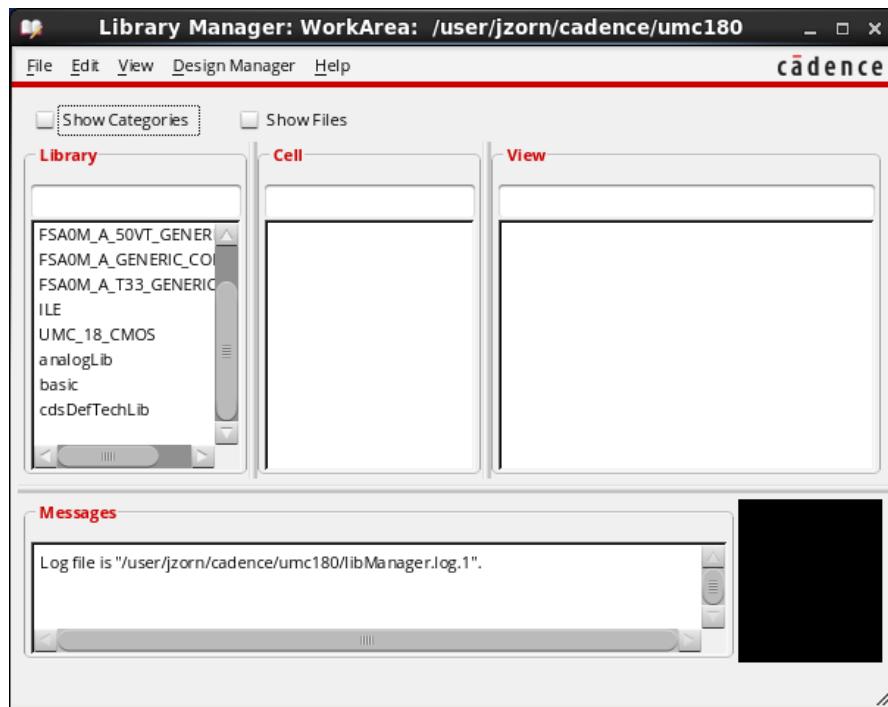


Figure 5: The Library Manager

A new library can be created in the *Library Manager* under the menu item *File > New > Library*.

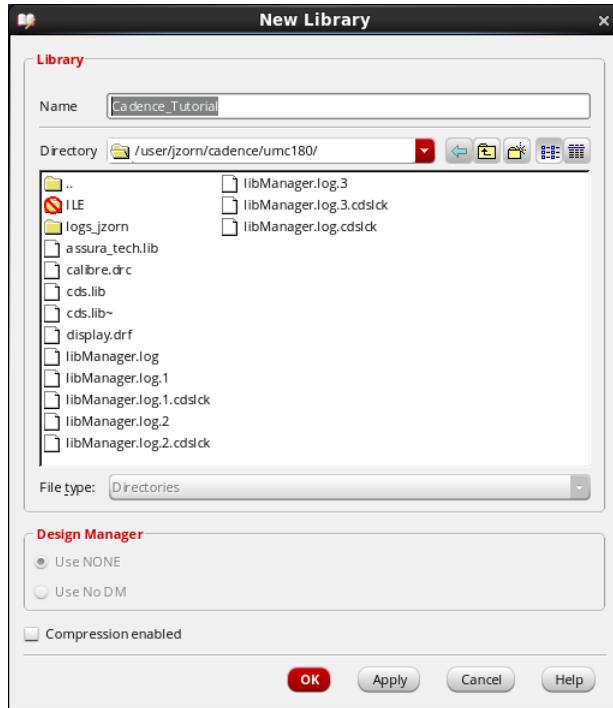


Figure 6: Create a new library

In the window that opens (see Figure 6), you can specify the name and the storage path. Give the library a meaningful name. By default, the library should be created in the *umc180* folder in which the software was started. After confirming with *OK*, you must specify that this library will contain design data for the *UMC 180nm* technology. To do this, select the option „*Attach to an existing technology library*“ in the window that opens and confirm with *OK*.

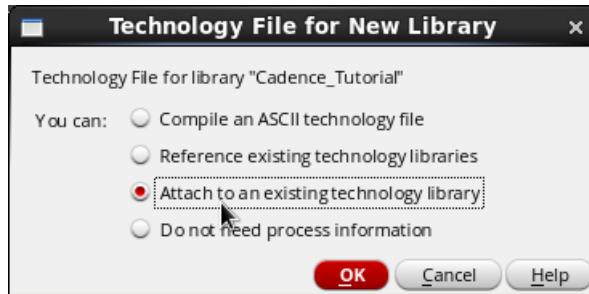


Figure 7: Linking the files

Note: Please note that this window may open in the background and be hidden by other windows. It may be necessary to move all other windows to the background in order to see the window shown in *Figure 7* for the technology assignment.

Finally, the UMC_18_CMOS library must be selected as *Technology Library*.

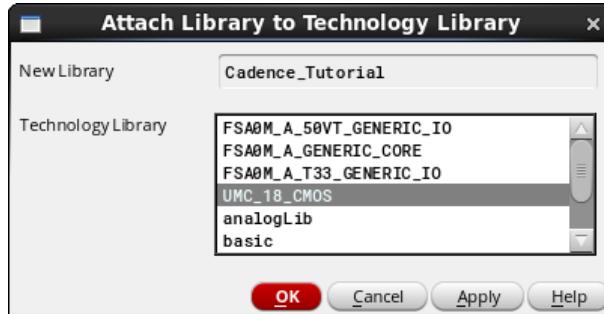


Figure 8: Adding the library

Now that you have your own library, you can also create a new cell (*Cell View*). A cell has different representations, so-called *Views*. A circuit diagram corresponds to the *Schematic View*. There is also the *Symbol View*, in which the schematic symbol for instantiating the cell in a hierarchical schematic is stored, and the *Layout View*, which contains information for mask production.

A cell is created in the *Library Manager* under *File > New > Cell View*. In the *New File* window (see *Figure 9*), the name of the library created should be entered and the name of the cell selected. As only one circuit diagram is to be created initially, both under *View* and under *Type schematic* are selected or entered. Select a meaningful name and confirm with *OK*.

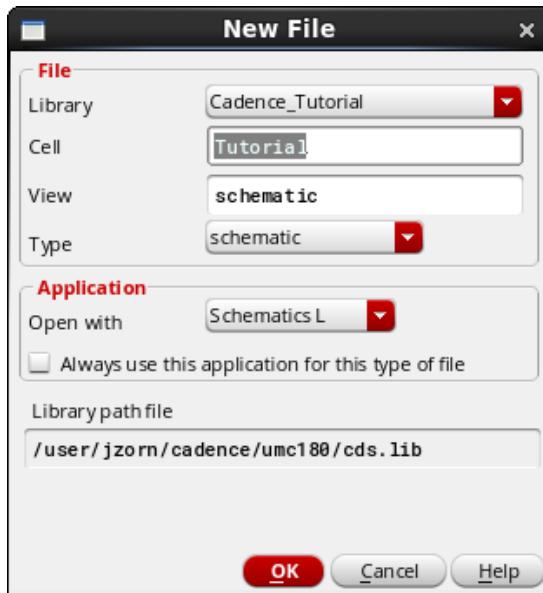


Figure 9: Create a new File

4 The circuit diagram editor

4.1 The editor

You are now in the main window of the *Schematic Editor* (Figure 10). Components are placed and connected here. The toolbar is located at the top of the window and contains various functions and tools for quick access. The toolbar and its functions are explained in detail in the following section.

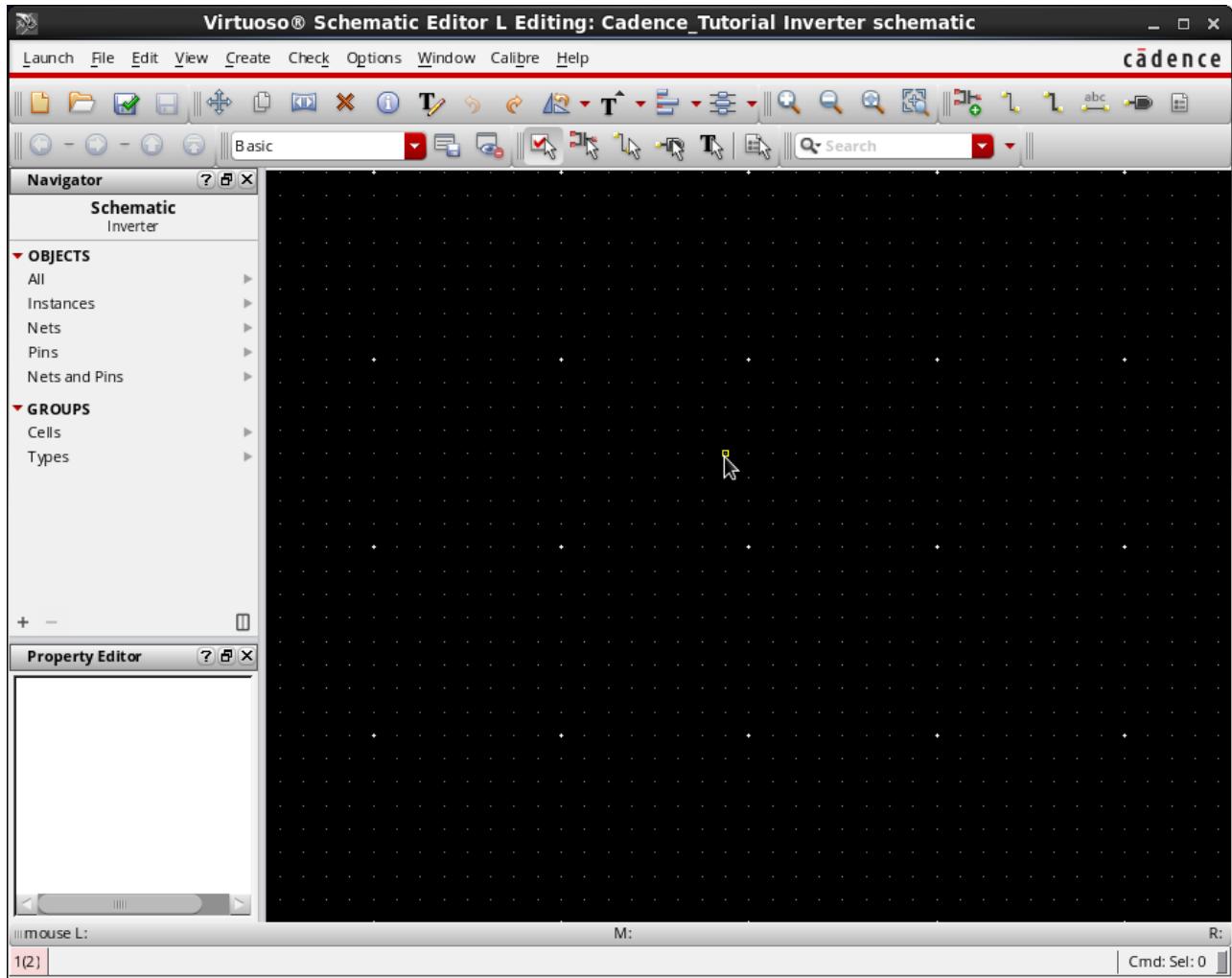


Figure 10: The circuit diagram editor

4.2 Toolbar elements in the circuit diagram editor

The following list describes the function of the pictograms, starting from left to right. So-called *Short-Cuts* are listed in square brackets and marked in bold, which allow the corresponding command to be executed using a key combination.



Figure 11: The toolbar of the circuit diagram editor

1. New
2. Open
3. Check and Save [**Shift+X**] -> *Schematic* check before a simulation
4. Save
5. Move [**Shift+M**] -> Move selected objects without connections
6. Copy [**C**]
7. Stretch [**M**] -> Move selected objects with connections
8. Delete [**Del**]
9. Edit Properties [**Q**] -> Edit the parameters of a selected component
10. Direct Text Edit [**T**]
11. Undo [**U**] -> Undo the last work step
12. Redo [**Shift+U**] -> Restore a work step that has been undone
13. Rotate [**R**] -> Rotate the symbol of a component
14. Change Text Size
15. Align
16. Distribute
17. Zoom In [**MouseIn**]
18. Zoom Out [**MouseOut**]
19. Zoom To Fit [**F**] -> Centre the circuit in the *Schematic*
20. Zoom To Selected -> Centre the selected objects
21. Create Instance [**I**] -> Place a new component in the *Schematic*
22. Create Narrow Wire [**W**] -> Create a conductive connection between components
23. Create Wide Wire [**Shift+W**]
24. Create Wire Name [**L**] -> Name a wire
25. Create Pin [**P**] -> Place input/output pins in the *Schematic*
26. Command Options [**B**]

5 Schematic and Cellview

5.1 Construction of the first circuit

In this section, the first example circuit to be simulated in the tutorial will be created. As shown in *Figure 12*, this circuit will consist of an NMOS transistor with adjustable voltage sources at the drain, gate and bulk terminals. First, the individual components are placed and then wired together.

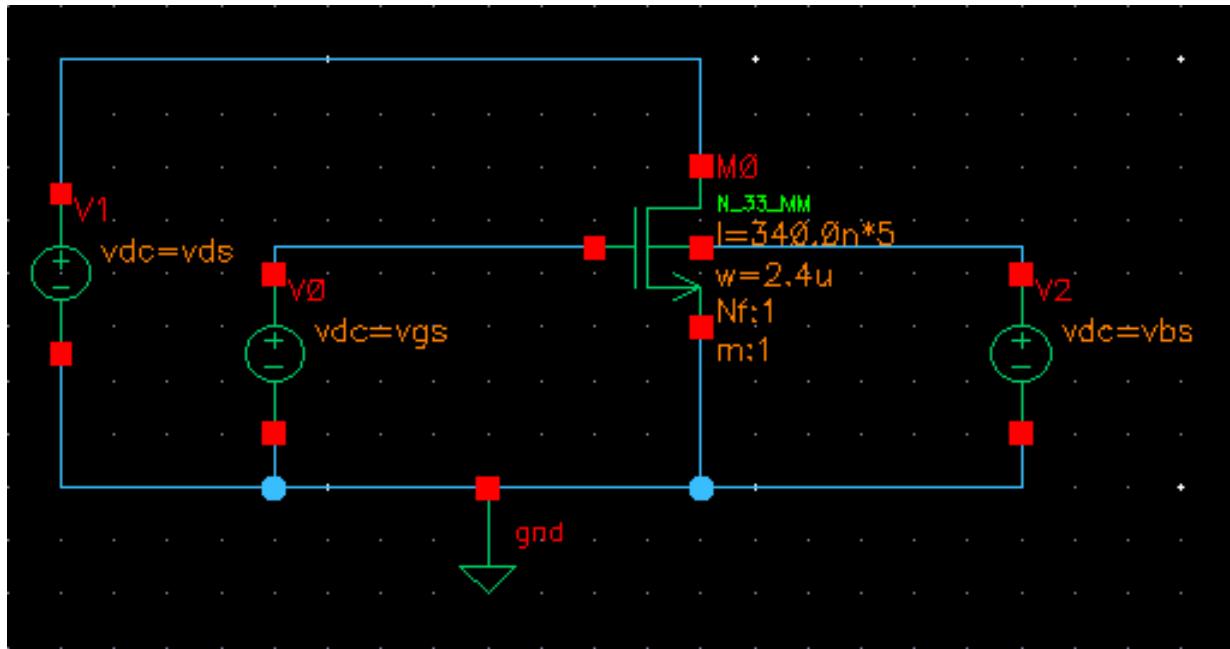


Figure 12: The first circuit of the tutorial

Build the circuit as shown in the *Figure 12* above. Use the instructions on the following pages to build the circuit.

The required components can be selected via the toolbar element *Create Instance* or via the [I] button. This opens the initially empty *Add Instance* window (see *Figure 13*). The *Browse* button in the top right-hand corner opens the *Library Browser*, which can be used to select the desired component. The selected component in the *Library Browser* is automatically transferred to the corresponding fields in the *Add Instance* window. Make sure that *symbol* is selected in the *View* field. If the *Add Instance* window is an obstacle when placing the component, e.g. because it covers the circuit diagram, it can be moved to the background via *Hide* button.

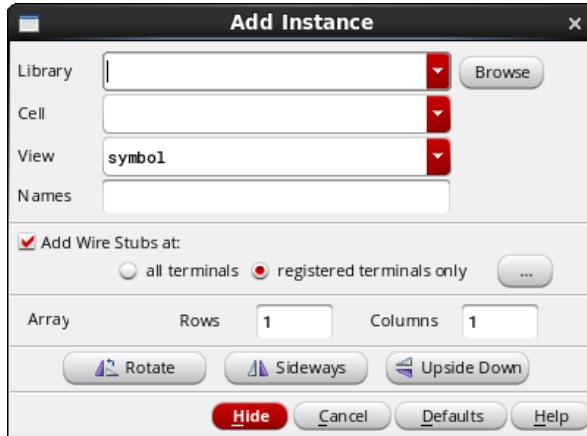


Figure 13: Adding a component

For the circuit, place a *NMOS* transistor (cell name *N_33_MM*), three voltage sources (cell name *vdc*) and a ground connection (cell name *gnd*) in the *Schematic*. The voltage sources and the ground are located in the *analogLib* library. The transistor is located (as shown in *Figure 14*) in the *UMC_18_CMOS* library.

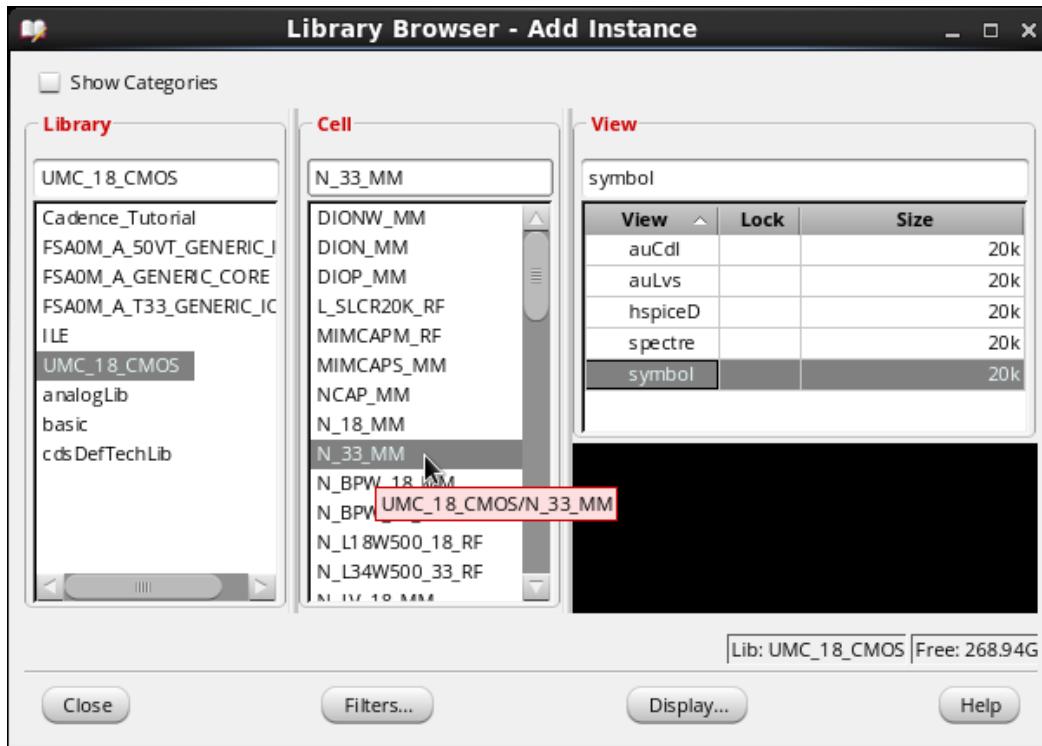


Figure 14: Listing of components in the libraries

If the symbol of a desired component has been selected in the *Library Browser*, the properties of the com-

ponent can be changed in the *Add Instance* window before it is placed. The values can either be set using variables or defined using fixed numerical values. This will be explained in more detail in *Chapter 5.3*.

Note: After completing an action, such as placing components in this case, this action should be ended each time by pressing the **[ESC]** key so that the action is not repeated unintentionally and, for example, further components of the same type are placed by mistake. The same applies to all other actions performed in Cadence.

Finally, all components must be connected with simple wires. Either the toolbar element *Create Narrow Wire* or the keyboard shortcut **[W]** can be used for this. When wiring, first click on the point where the wire should start and then on the point where the wire should end. As soon as the wiring is complete, end this action with the **[ESC]** key.

After completing the circuit, the circuit can be checked for errors and saved by clicking on *Check and Save*. Please note that the circuit can only be simulated successfully if the *Check and Save* action is executed after a change to the circuit diagram.

5.2 Dealing with faults in the circuit

If the *Check and Save* window opens during the *Schematic Check* action, there are errors in the circuit.

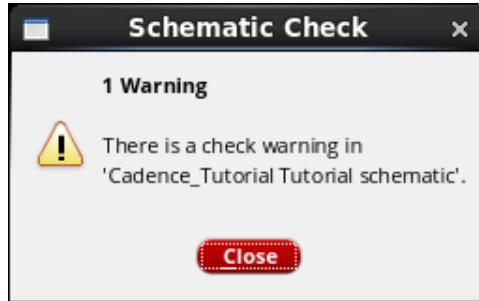


Figure 15: Message about errors in the circuit

Pressing the [G] button opens the *Find Marker* window, in which you can display these errors. The following Figure 16 shows this window with an example error message. In the case shown, the message is displayed that more than three lines are connected to a node, which is not permitted in Cadence.

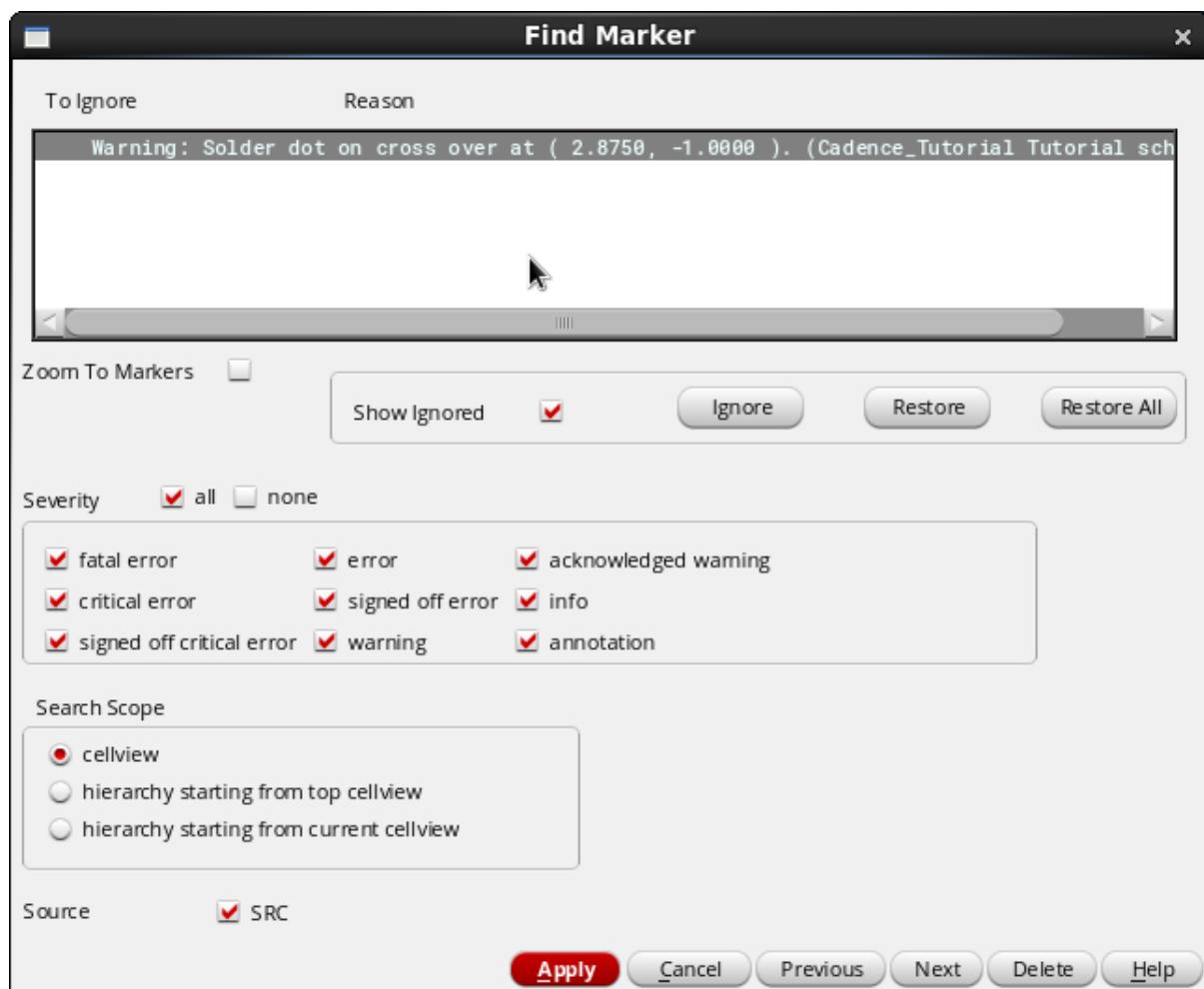


Figure 16: Window for displaying possible errors

5.3 Editing component properties and creating variables

Component parameters can be defined directly during instantiation in the *Add Instance* window (Figure 17). The parameters can also be changed after placement in the circuit diagram. To do this, select the desired component in *Schematic* and then click on the toolbar element *Edit Properties* or press the button [Q] (Figure 18) opens a window with the component properties.

For the dimensioning of the transistor, a tenfold minimum width of $W = 240 \text{ nm} \times 10 = 2.4 \mu\text{m}$ should first be selected. By clicking on *Apply*, the finger width (*Finger Width*) is automatically adjusted to the total width. For the transistor length, five times the minimum length of $L = 340 \text{ nm} \times 5 = 1.7 \mu\text{m}$ should be selected in order to reduce the influence of the channel length modulation.

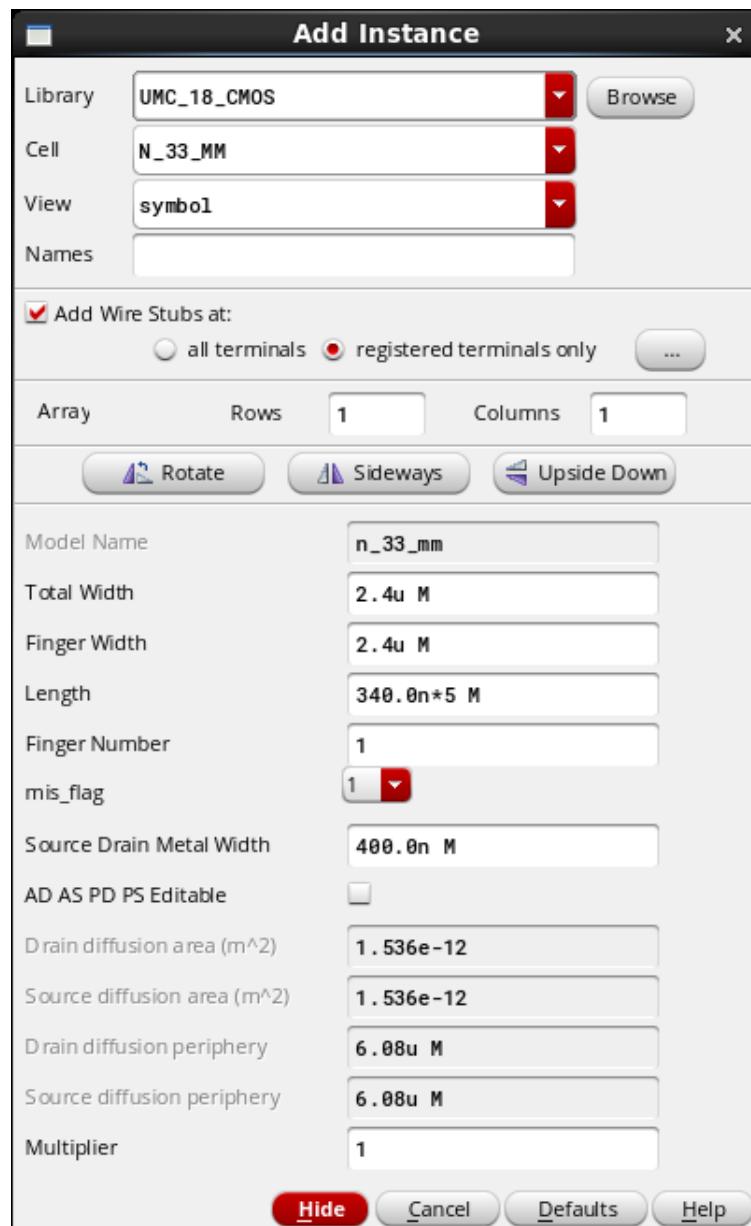


Figure 17: The *Add Instance*-Window

Changing component parameters can be simplified by using variables. In this tutorial, it makes sense to use variables for all existing voltages in the circuit. These can be used in simulations for parameterized variations. In this circuit diagram, the variables vds are used for the drain-source voltage, vgs for the gate-source voltage and vbs for the bulk-source voltage. Please enter these variables in the parameter field *DC voltage* of the respective source. The exact values of the voltage variables are defined in the respective analyses in *Chapter 7*.

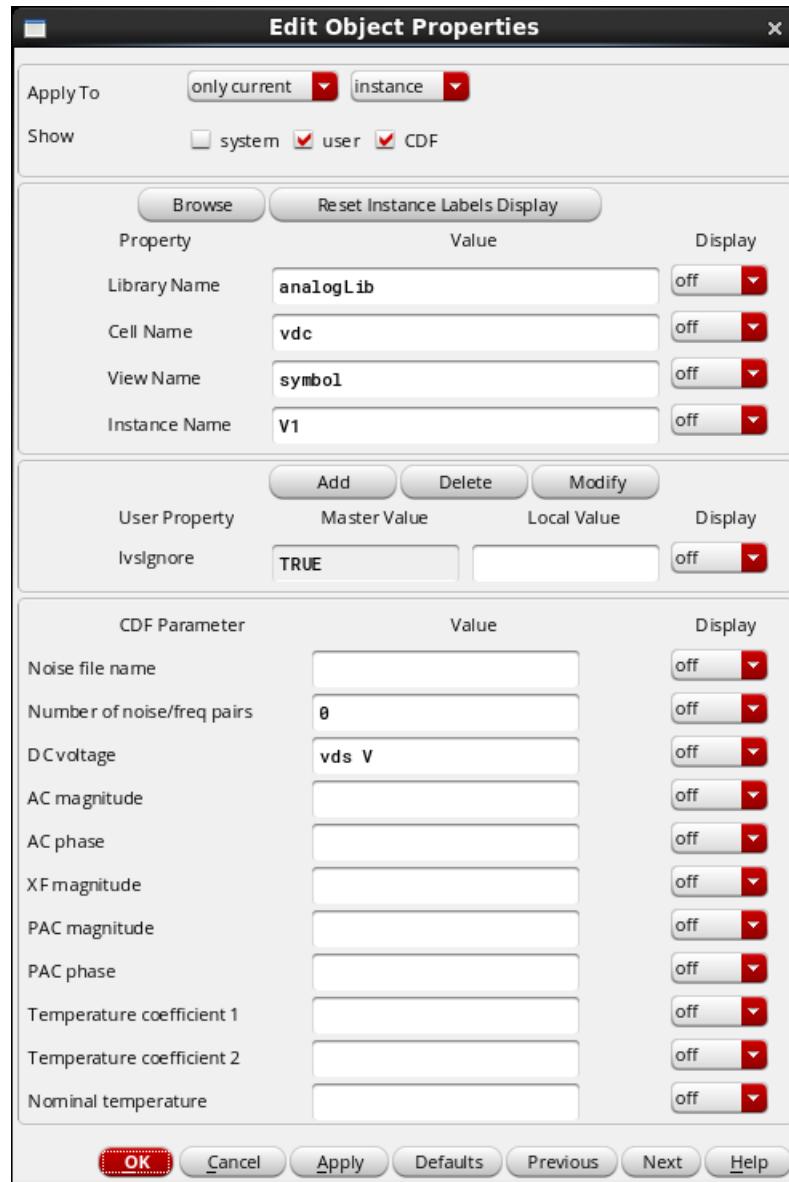


Figure 18: The *Edit Object Properties*-Windows

Note: Only letters or sequences of letters that are not abbreviations for existing units or prefixes should be used as variables.

5.4 Use of labels

In a larger circuit, it can happen that clarity is lost due to the large number of nets. To avoid long lines in the *Schematic*, it is possible to provide them with *Labels*, which creates a connection even if the lines are not visibly connected to each other. In addition, *Labels* can be used to display a desired name in analyses instead of an internal network name. However, the use of *Labels* is purely optional in this case. To do this, press the *Create Wire Name* toolbar element in the *Schematic* window or press the [L] button.



Figure 19: Fenster zum Erstellen von *Labels*

Choose a desired name and click *Hide*. You are now back in the *Schematic* window and can place the *Label* by clicking on a line. In order for two lines to be connected to each other, the same names must be chosen.

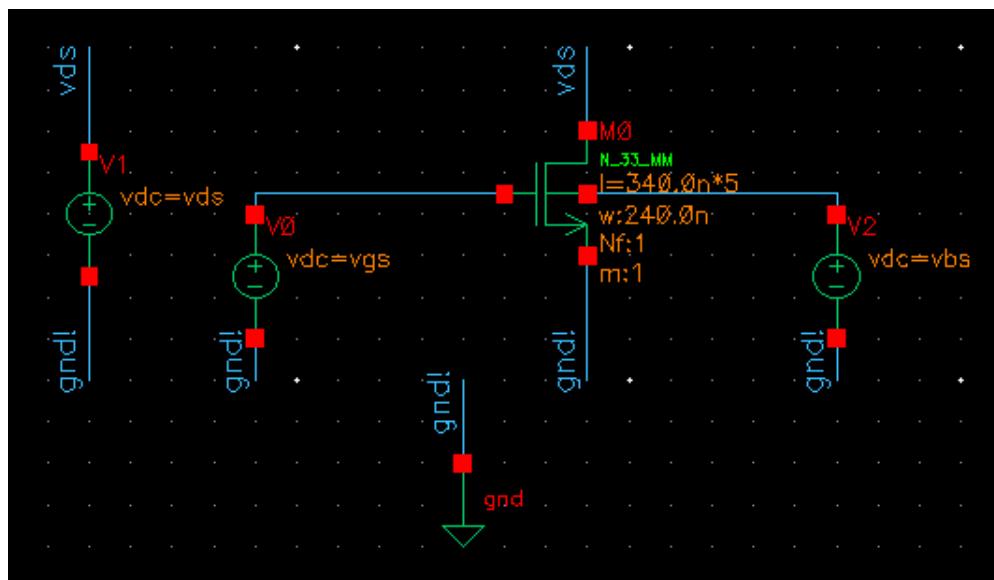


Figure 20: The circuit is simplified as an example by using *Labels*

5.5 Complete closure of the libraries used

When using Cadence over a longer period of time, you may not always be able to work on the same computer in the lab. For this reason, it is important to completely close all used cells after the work is done so that they are deleted from the virtual memory and released for use on other computers. There is a separate window for this, which can be accessed in the *Virtuoso* window (Figure 21) via the menu item *File > Close Data*.

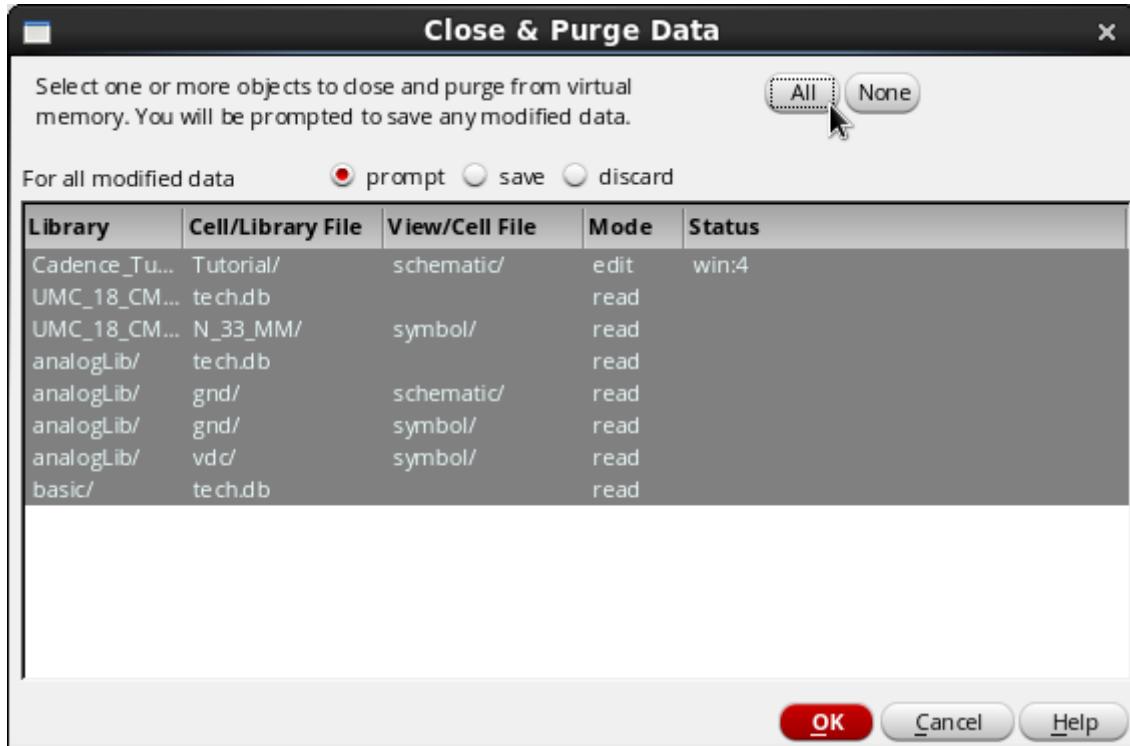


Figure 21: Window to close the data in use

Select all libraries here by clicking on *All* and then confirm the action by clicking on *OK*.

6 Setting up a simulation

In Cadence Virtuoso, the simulation of designed circuits in the *Schematic Editor* is performed in different ways. This tutorial describes the use of simulation environments tools, namely **Analog Design Environment Explorer (ADE Explorer)** and **Analog Design Environment Assembler (ADE Assembler)**. Moreover, in the Appendix chapter the usage of old tools, ADE-L and ADE-XL for further study is described.

6.1 The ADE Explorer main window

This simulation tool is accessible from the *Schematic Editor* menu *Launch > ADE Explorer*. Figure 22 shows the empty main window of the ADE Explorer which opens as an additional tab next to the schematic tab. Various simulations can be configured and executed in this window, including DC analyses to determine the operating point, transient analyses to simulate time-varying signals, and AC analyses to perform a frequency simulation in the small-signal circuit. Simulation results can also be displayed graphically using the ADE Explorer.

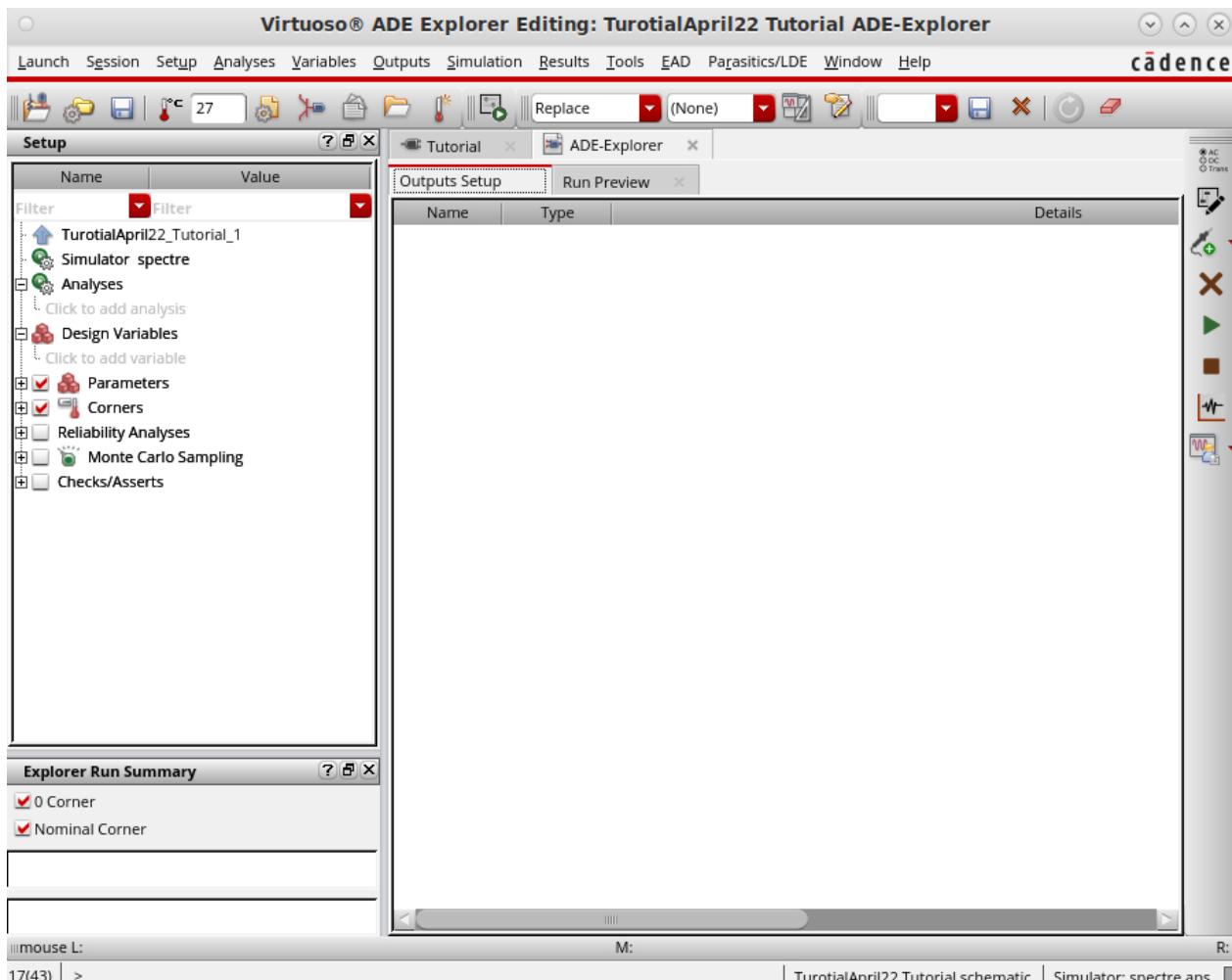


Figure 22: The main window of ADE Explorer

6.2 Importing variables from the schematic

Before a simulation can be run, the variables specified in the *Schematic* must be imported into ADE Explorer. This is done via the *Design Variables* area on the left side of the ADE Explorer main window by clicking on *Click to add variables*.

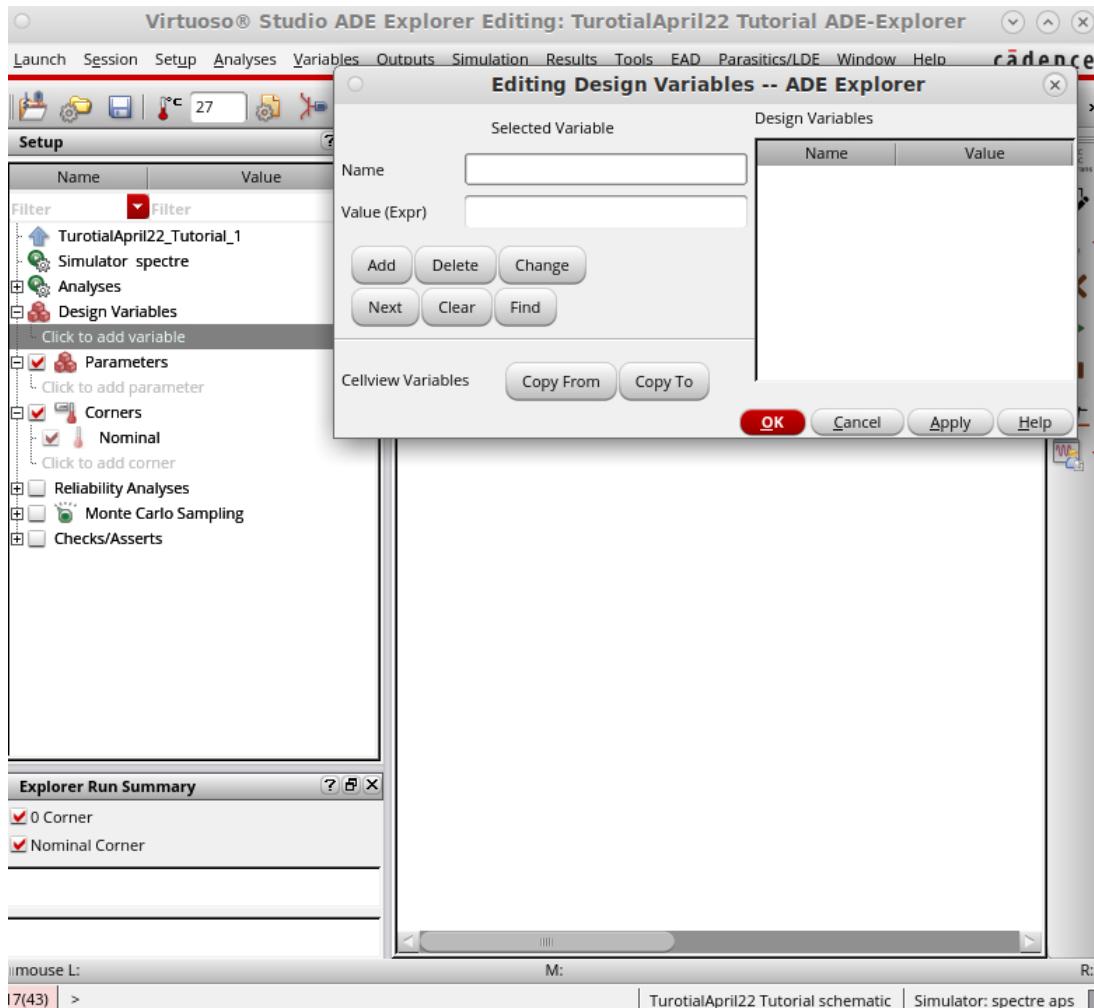


Figure 23: Variables from the *Schematic* import

By selecting the menu item *Copy From* all variables created in *Schematic* are imported. The values can be assigned to the variables by simply clicking on the corresponding field in the *Value* column. After importing the variables *vbs*, *vds* and *vgs* should be listed in the *Design Variables* window. First, set the variables to the following numeric values:

- $v_{bs} = 0 \text{ V}$
- $v_{ds} = 3.3 \text{ V}$
- $v_{gs} = 2.3 \text{ V}$

The values set here are used to calculate the operating point for a sweep analyses in the test.

6.3 DC simulation to display the operating point

The first simulation is a DC analysis, which determines the operating point of the transistor and displays it in the *Schematic*. This simulation can be started in the ADE Explorer by double clicking on the Analyses item in the vertical toolbar on the left side of the window or with the menu item *Analyses > Choose*. All adjustable simulation types are depicted in *Figure 24*.

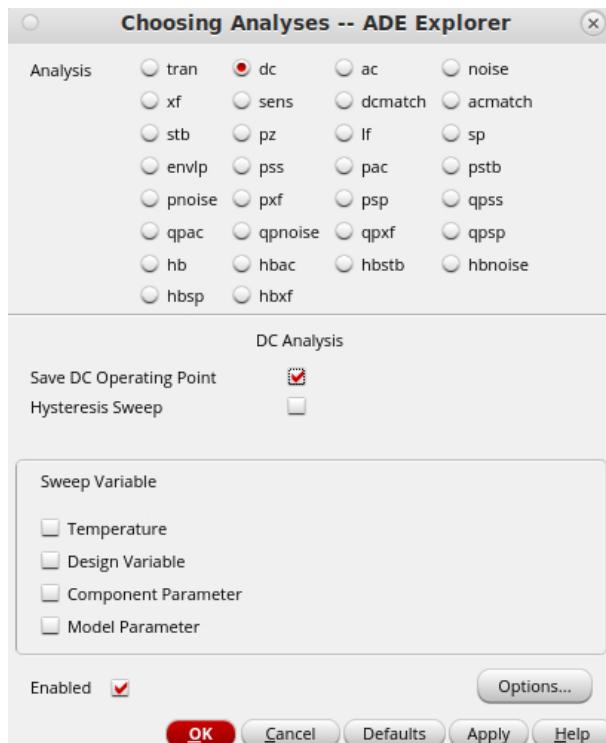


Figure 24: Window for setting up a simulation

The first thing to do here is to set up a simple DC analysis without using any variables or sweeps. Select DC analysis in this window, if not already done. In the middle area of the window, the *Save DC Operating Point* field must be checked so that the operating point can be saved and displayed on the transistor in the circuit. Clicking OK completes the simulation configuration, which is then displayed in the *Analyses* section of the ADE Explorer window (see *Figure 25*).

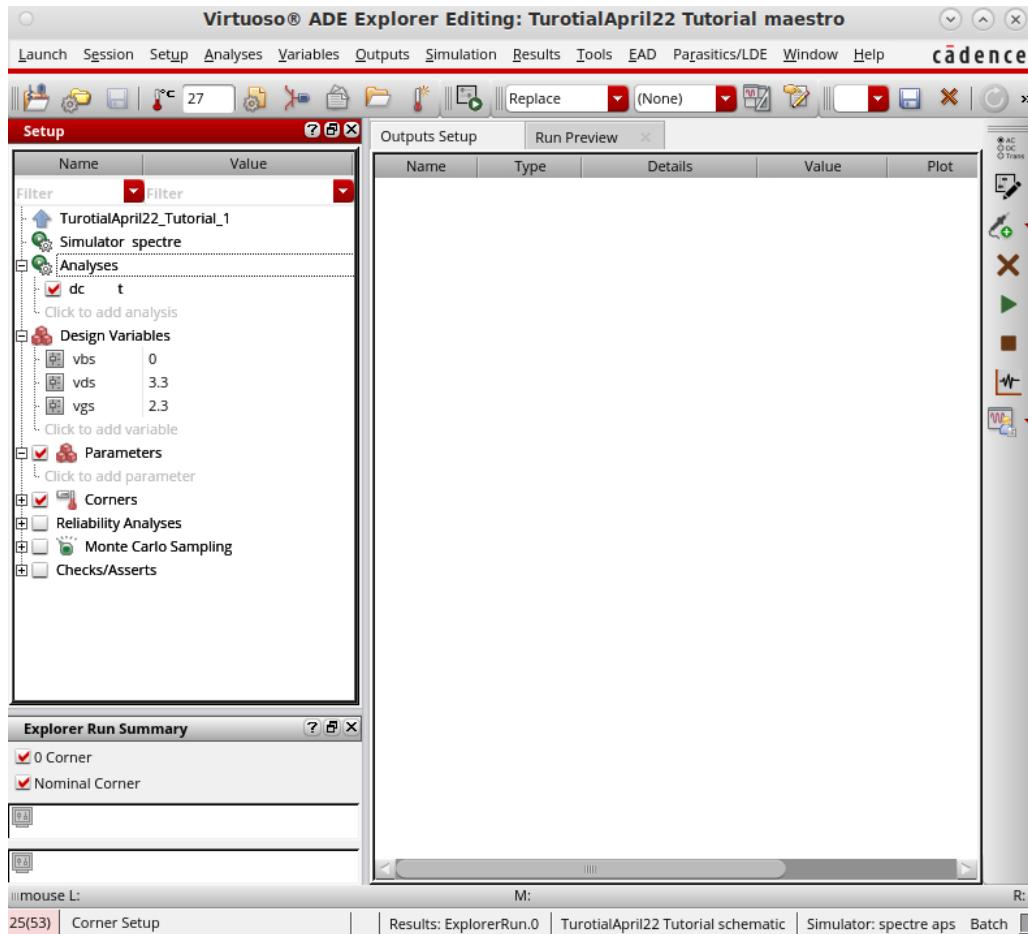


Figure 25: The simulation set up in the ADE Explorer window

The simulation is started by clicking on the green icon *Netlist and Run* on the right toolbar or via the menu item *Simulation > Netlist and Run*.

A netlist is automatically generated from the schematic and passed to the simulator. At runtime, the simulator generates messages that are logged in a new window. The simulation has run successfully if the message *Convergence Achieved* can be found in the log files.

If the log window does not open, the generation of the netlist may have failed. A common cause for this is the failure to perform the *Check and Save* action after editing the Schematic. Therefore, always remember to perform the *Check and Save* action before simulating a modified schematic.

6.4 Display of the operating point

After successfully running this DC simulation, it is possible to display all voltages on the circuit lines and the operating point of all components used. The operating point is displayed in the ADE Explorer window via the menu items *Results > Annotate > DC Operating Points*, or the mains voltages via *Results > Annotate > DC Node Voltages* (see Figure 26). Another way to activate this display is in the Schematic window under *View > Annotations > DC Operating Points* or *DC Voltages* (see Figure 27 on the next page).

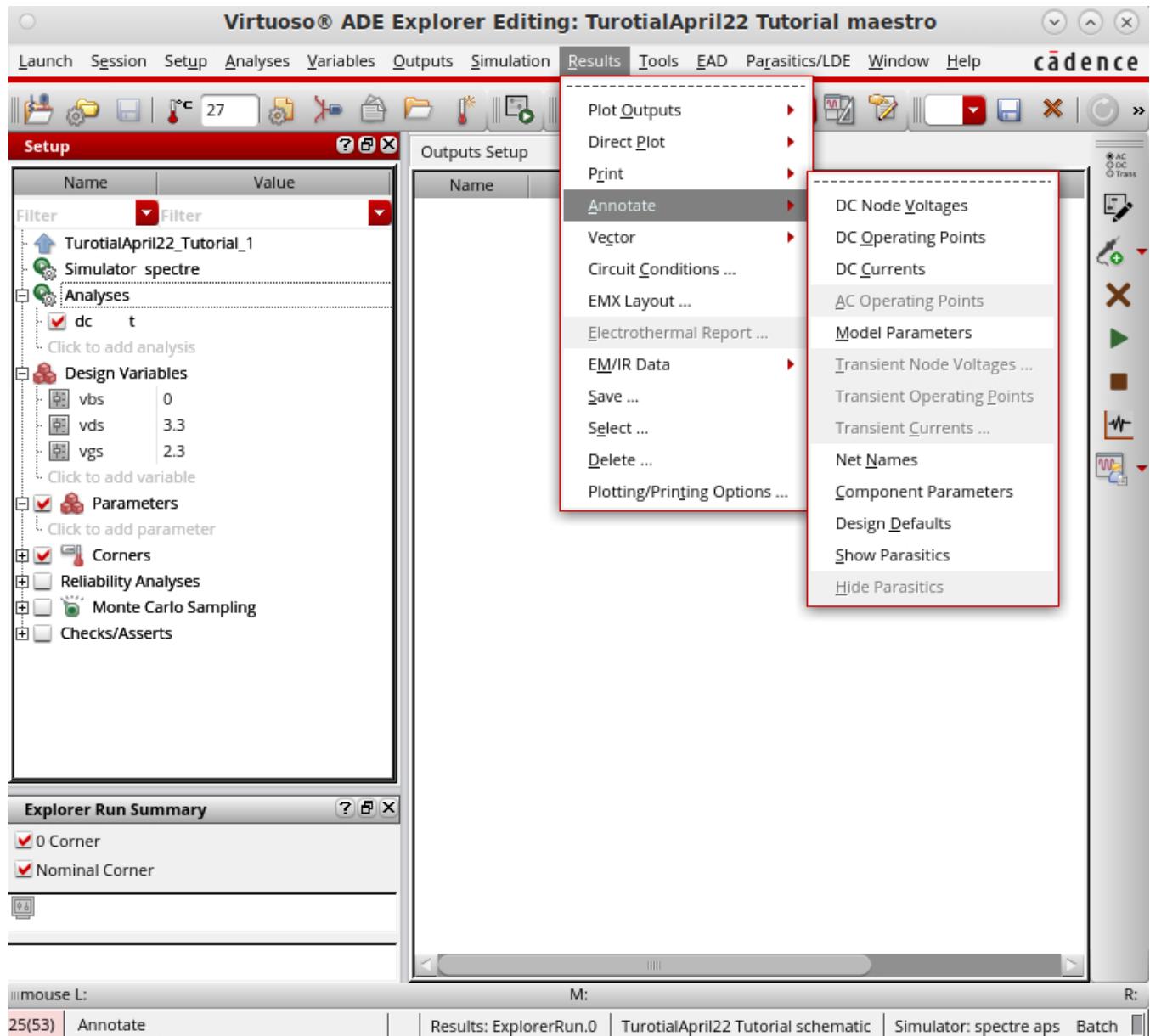


Figure 26: Display options in the ADE Explorer window

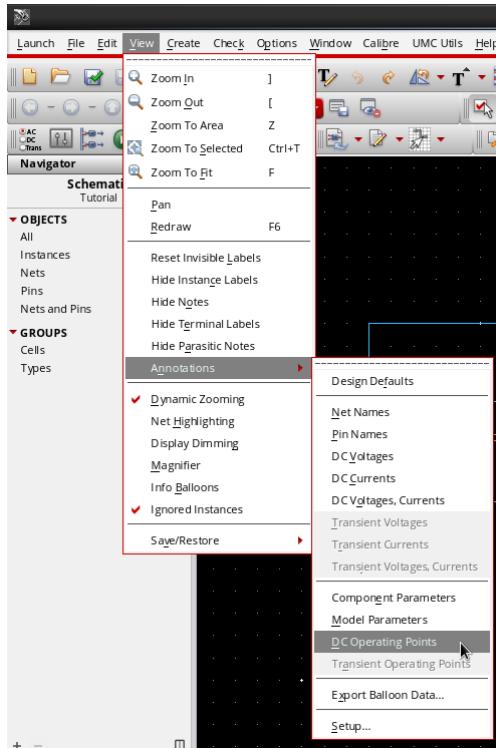


Figure 27: View the DC-Operating Points

Also under the View menu or through the context menu that opens when you right-click on the transistor, you can access the Annotation Setup window via Annotations > Setup.

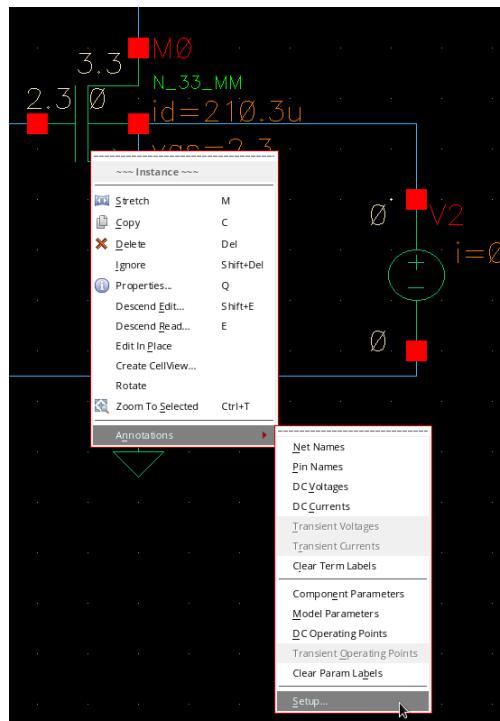


Figure 28: Call up the setup to set the Annotations

In the Annotation Setup window, the NMOS transistor in the Cell area must be selected. As illustrated in Figure 29, additional operating point data, including saturation voltage (v_{dsat}), can be designated within one of the available expression fields. In order to apply changes to the operating point display to both the selected transistor and all other transistors of the same type in the circuit, the star symbol can be selected under *Instance* instead of an instance name. The window is then closed by clicking on OK.

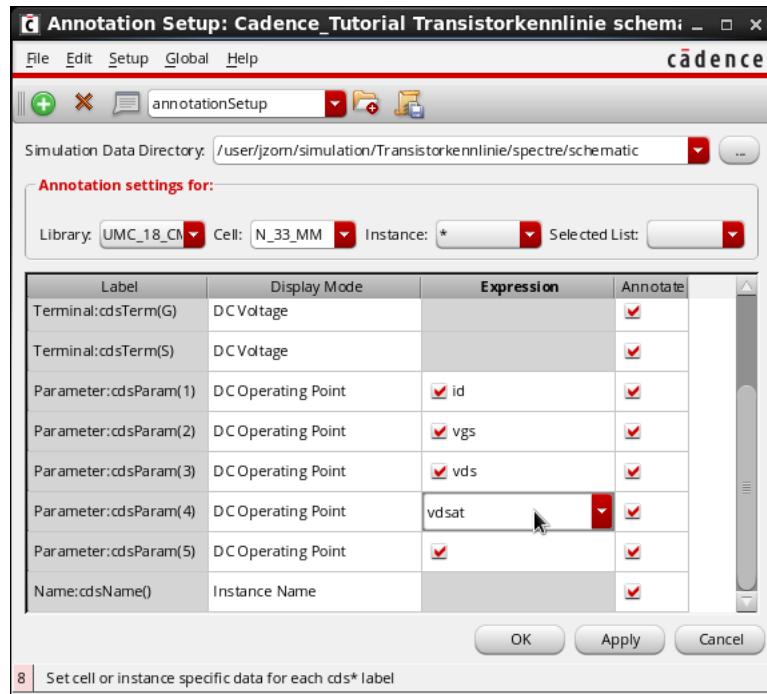


Figure 29: Display the saturation voltage of the transistor

Note: If there are other transistor types in your circuit, such as PMOS transistors, they must also be selected in the *Cell* field and the above steps must be carried out again.

When the star symbol has been selected in the *Instance* popout, after confirming with OK, another window (Figure 30) will appear asking if the settings should be applied to all selected elements of the same type. By clicking on *Apply * settings to all* and further confirming with OK, the saturation voltage of the transistor should now be displayed next to the component in the Schematic (compare Figure 32).

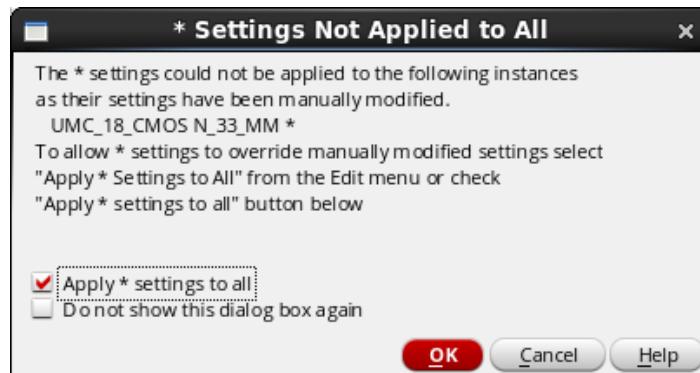


Figure 30: Confirm display settings

6.5 Creating simulation outputs from the schematic

Before an analysis is carried out, it is necessary to define so-called *outputs*, which are automatically displayed after the simulation has been carried out. Either voltages on individual lines of the circuit or connections of components (e.g. the drain of a transistor) can be considered. To set up an *Output*, click on the icon **Add Outputs** in the vertical bar on the right of the ADE Explorer main window, or alternatively select *Outputs* in the menu bar.

This opens the *Outputs* window, which offers various options for defining *Outputs*. The *Output* to be simulated is specified in the text field *Expression*. This can be the name of a network or port, as well as a mathematical expression. The use of mathematical expressions with the help of the *Calculator* tool is discussed in detail in the following section. First, an *Output* should be set up by a direct selection in the *Schematic*. To do this, click on the *To Be Plotted > Select on Design* option.

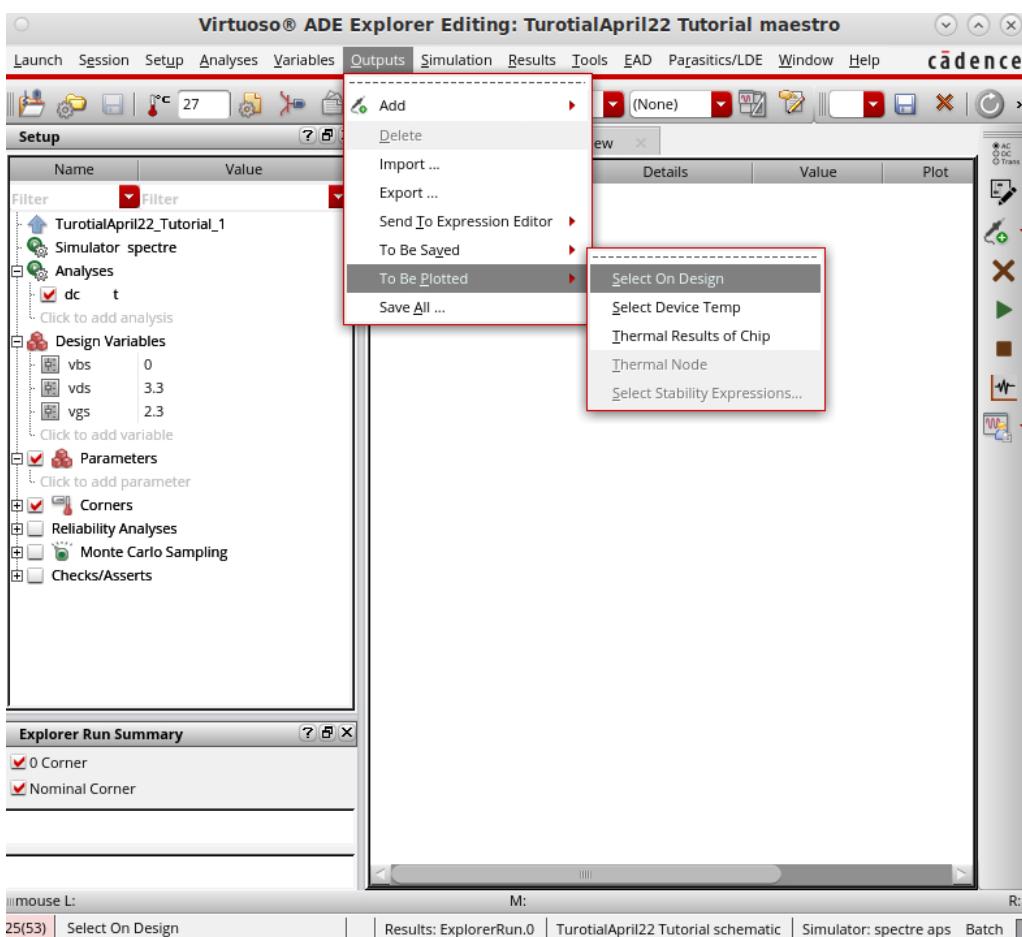


Figure 31: Window for setting up *Outputs*

You now get to the *Schematic* view and click to select a point in the circuit that you want to specifically look at in the simulation. There are various options to choose from. If you click on a net with the mouse, the voltage of this net is selected as *Output*. If you click on an individual connection contact of a component, the current flow at this point is created as *Output*. By clicking on the component, the current from all connection contacts of this component are selected as outputs.

It is also possible to select several nets, connections, and components at the same time. The drain connection of the transistor was selected here as an example.

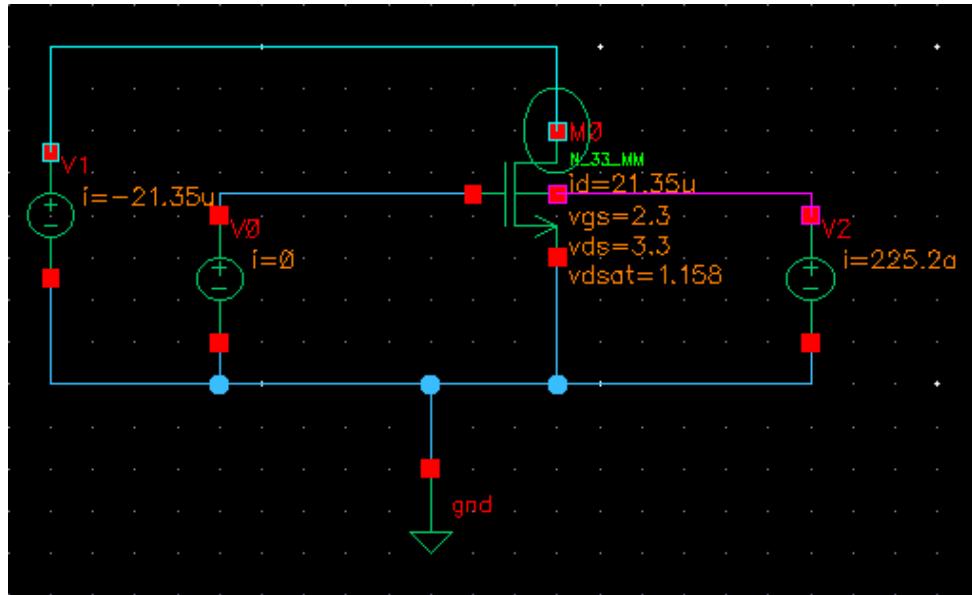


Figure 32: Colored markings inside *Schematic* mark the *Outputs*

The selected line or component connection is marked in color, and also listed in the *Table of Outputs* area of the *Setting Outputs* window. By default, the software uses a combination of the instance name of the component and the name of the selected port or line as the name. This expression can be given a meaningful name in the *Name* field.

6.6 Creating outputs using the calculator

Another way to create *Outputs* is the so-called *Calculator*. Figure 33 shows the main view of the *Calculator* tool. This tool can be opened in the ADE Explorer window under *Tools > Calculator*. The *Calculator* offers extensive options for numerical analysis and evaluation of simulation results using special functions and generic mathematical expressions. The *Calculator* stores results in a stack according to the *Last-In-First-Out* (LIFO) principle.

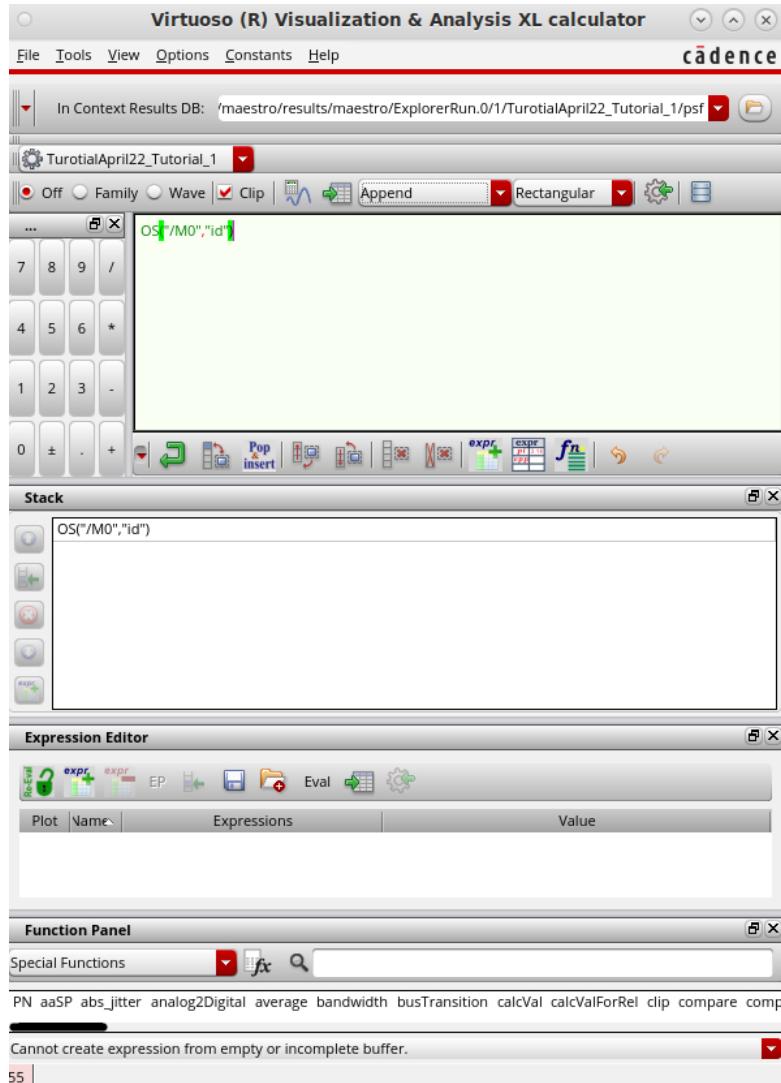


Figure 33: Main view of the *Calculators*

With the help of the buttons in the upper third of the *Calculator*, expressions for operating points, sizes or even any signals can be created for all common types of simulation (DC, AC, Trans) with just a click select from the *Schematic*. These expressions are sorted from left to right in blocks according to the following scheme: *Trans*, *AC*, *DC*, *Sweeps*, *Operating Points*. For better understanding, the current flow of the drain connection of the transistor should also be selected as an example.

Clicking on the *os* button opens a window in which an *instance* can be selected from the *Schematic*. Please note that the window shown in *Figure 34* may open in the background and may need to be brought to the foreground by selecting it.

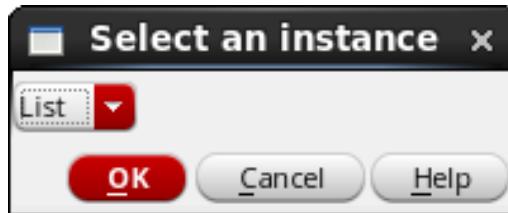


Figure 34: Window for selecting a component

Clicking on the transistor in the circuit diagram opens a selection list with the available operating point sizes of the selected transistor. Select the drain current *id* and press *OK*. (Notice: For the operating point information to be available in this list you need to run a DC simulation first.)

The selected quantity is now displayed as an *expression* in the stack window of the *Calculator* (see *Figure 33*). The expression follows a fixed syntax, which can in principle also be entered manually in the stack area. In this case, the expression `OS("\M0",""id")` describes that the parameter *id* of the transistor M0, which corresponds to the drain current, extracted from the results of the operating point simulation.

Note on using the *Calculator* stack: The entry in the input area can be brought to the top of the *Calculator* stack by clicking on the *Enter* button (green arrow at the bottom of the input field). Alternatively, this also works with the **[Enter]** key. Expressions on the stack can also be dragged and dropped back into the input area and linked to other expressions there.

In order to be able to use the expression for the simulation as *Output*, it must be transferred to the ADE Explorer window. To do this, after creating the *Expression* in the *Calculator*, click on the cogwheel icon with a green arrow on it (send buffer expression to ADE outputs) in the right area of the *Calculator* window (see *Figure 35*). After importing, the *Output* can be named as desired in the *Name* field.

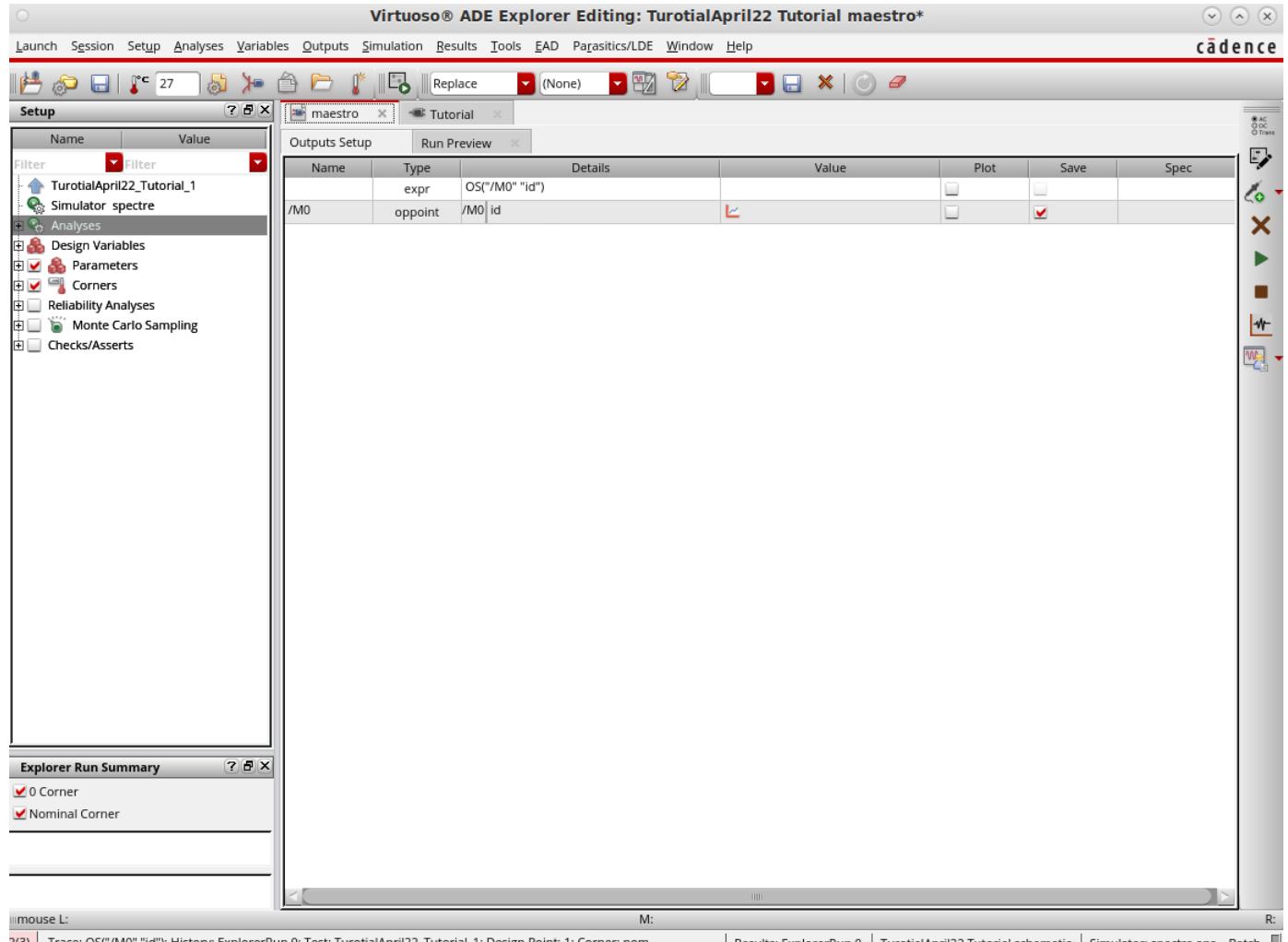


Figure 35: Import from *Calculator* as Outputs

6.7 Saving and loading ADE Explorer states

ADE Explorer views are saved for future reuse by pressing the disk symbol or choosing from menu item *Session > Save A Copy*. The *Save A Copy* window opens (Figure 36). The ADE Explorer state is then stored in a view *maestro*. The state can be reopened by clicking on the *maestro* view in the Library Manager or by launching the ADE Explorer from the menu command in the respective schematic and choosing the stored *maestro* view.

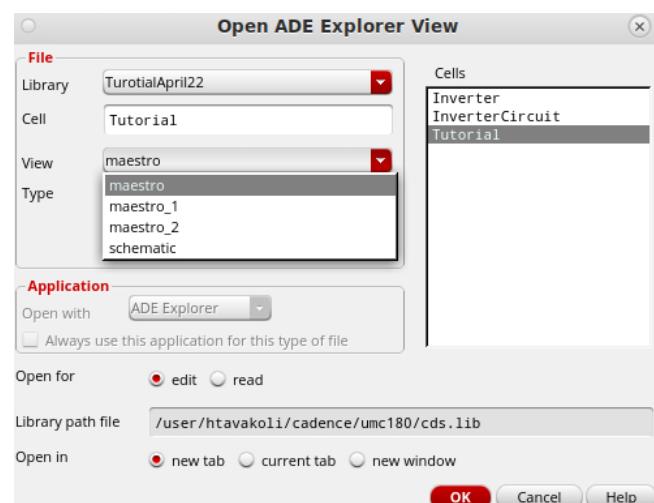
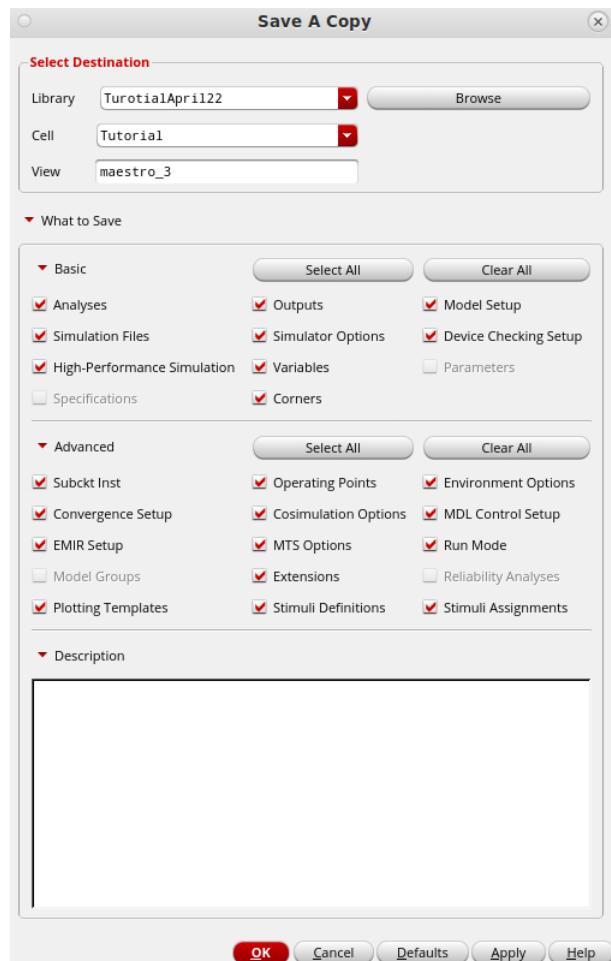


Figure 36: Storing ADE Explorer views

7 DC analysis in ADE Explorer

7.1 Gate-source voltage sweep

DC simulation offers the possibility of determining operating points as a function of a variable parameter. Such simulation series are called *Sweep* in Cadence. The temperature, design variables, component parameters, and individual parameters of a simulation model can be changed. To do this, open the *Choosing Analyses* window in the *Analyses > Choose* area of the ADE Explorer window by double-clicking on the existing DC simulation to calculate the operating point.

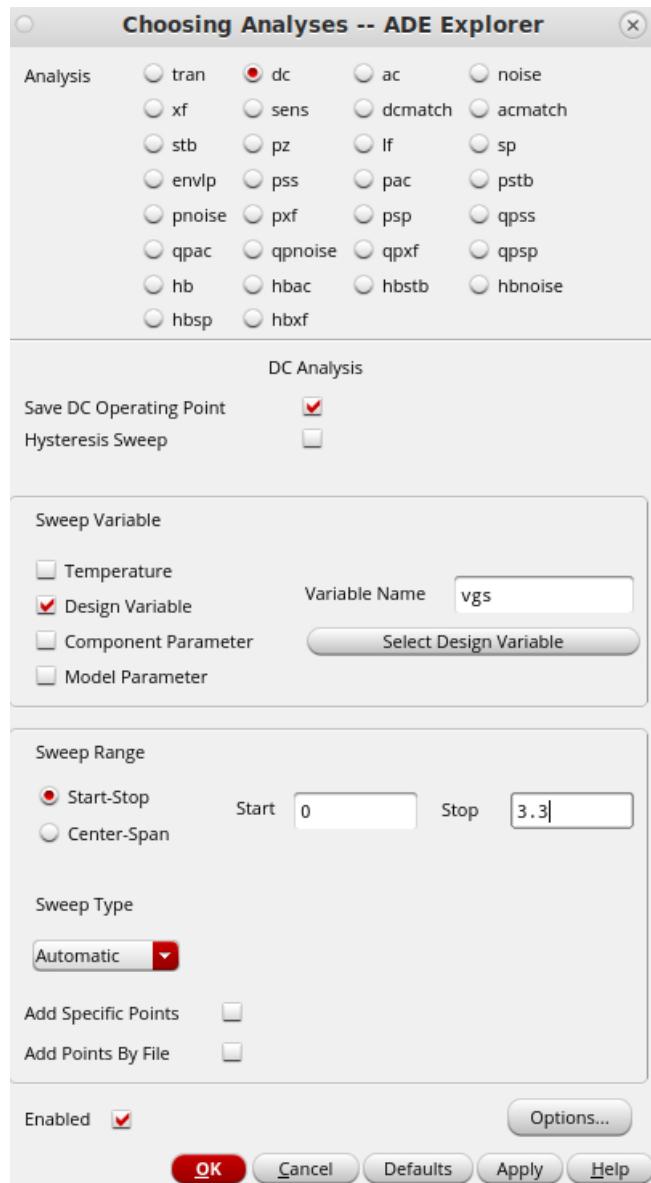


Figure 37: Simulation selection window

In the *Sweep Variable* area, a variable is selected that should be passed through. In addition to a fixed variable, such as the temperature, this can also be a variable created in the *Schematic (Design Variable)*, or a parameter of a component (*Component Parameter*). In the lowest window section called *Sweep Range*, the interval to be run through, as well as the type of *Sweep (Sweep Type)* can be specified. You can choose from *Automatic*, *Linear* and *Logarithmic*.

The following simulation corresponds to a DC analysis with a sweep of the gate-source voltage of the NMOS transistor. Select the checkbox *Design Variable*. The previously introduced variable *vgs* is used for this *sweep*. This can either be entered directly or selected by clicking on *Select Design Variable* in the window that opens (Figure 38).

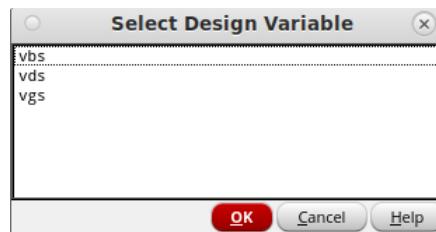


Figure 38: Selection of the created variables

A *Sweep* of type *Automatic* should be carried out from 0 V to 3.3 V. Enter these values in the *Sweep Range* area in the *Start* and *Stop* fields. The values of the variables for the drain-source voltage (*vds*) and the source-bulk voltage (*vbs*) initially remain constant at the values previously set in the ADE Explorer window section *Design Variables* defined values. The variable *vgs* is varied as a parameter during the *sweep*, but the originally set value in the ADE Explorer window of 2.3 V decides which gate-source voltage the operating point is for of the transistor is saved.

In this analysis, the current flow in the drain of transistor M0 should be defined as *Output*. How an *Output* is created from the *Schematic* is described before. If the creation was successful, it should appear as M0/D in the *Output* area (see Figure 39).

After selecting the *Outputs*, the simulation can now be carried out. To do this, all possible changes must first be saved in the *Schematic* using *Check and Save*. The simulation can then be started by clicking on the green *Run Simulation* button in the menu bar on the right edge of the window.

The *Output* shown in Figure 39 should be both saved and displayed graphically. To do this, both checkboxes (*Save and Plot*) are selected in the *Outputs* area of the ADE Explorer window. Figure 40 shows the dependence of the drain current on the gate-source voltage.

The characteristic corresponds to the theoretically known relationship between channel current and gate-source voltage. A larger gate-source voltage *vgs* leads to a greater current flow through the transistor. However, a significant current flow only occurs as soon as the gate-source voltage has exceeded the threshold voltage of the transistor. Mathematically, the current in saturation is described by the following relationship:

$$I_{DSat} = \frac{1}{2} \cdot C_{ox} \cdot \mu_n \cdot \frac{W}{L} (V_{GS} - V_{th})^2 \cdot (1 + \lambda \cdot V_{DS}) \quad (1)$$

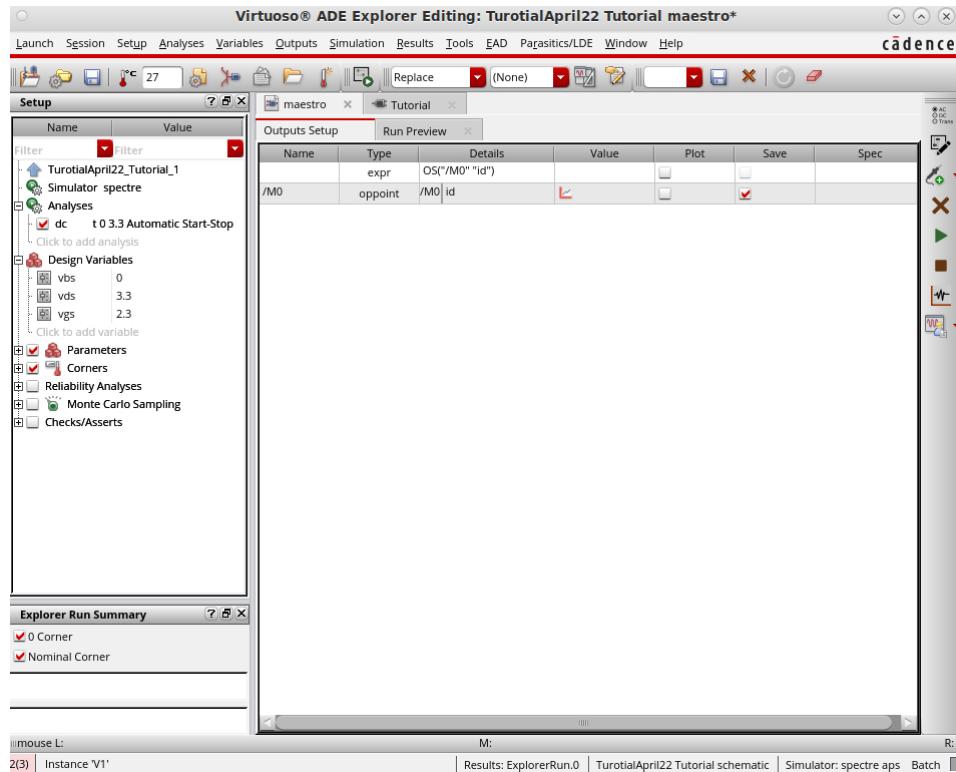


Figure 39: The ADE Explorer window set up for analysis

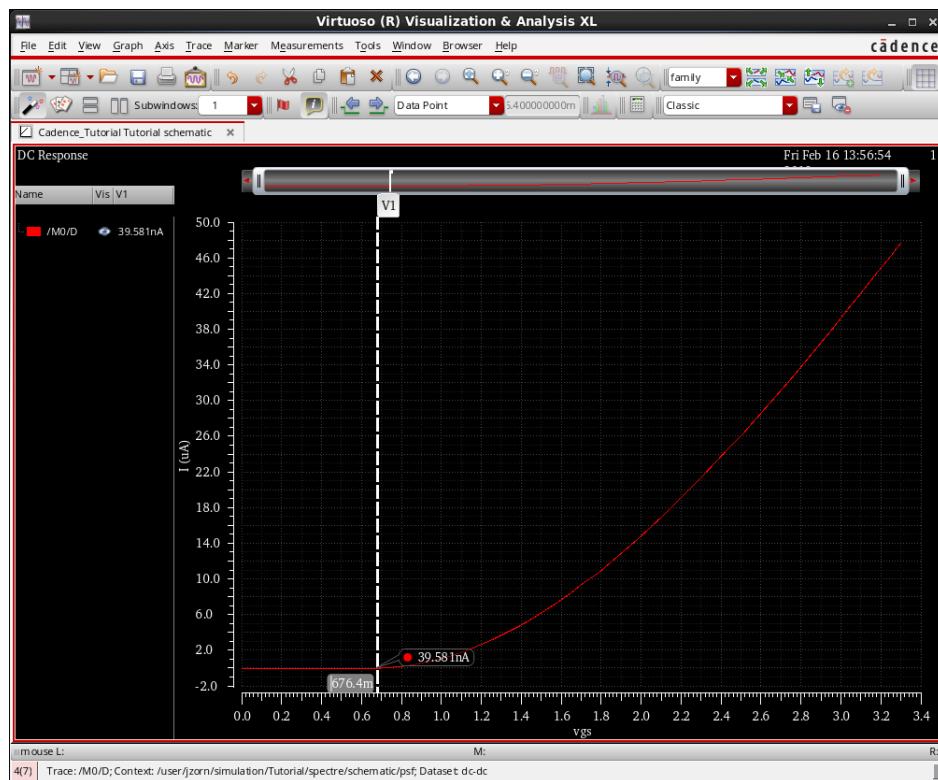


Figure 40: DC simulation with Sweep the gate-source voltage v_{gs}

7.2 Variation of transistor width

Next, the influence of the transistor width on the drain current will be examined. For this purpose, in addition to varying the gate-source voltage via a sweep of the variable v_{gs} , the transistor width should also be changed within a specified value range. Since the DC simulation only allows sweeps over a single size, the *Parameterize* function of the ADE Explorer simulation environment should be used for this simulation, which allows variations of several parameters.

To allow the width of the transistor to be changed parametrically, a variable W is introduced, which is entered in the component properties of the instantiated transistor. In this case, the variable should act as a factor of the minimum width. In the transistor properties, $240\text{ nm} \times W$ must therefore be entered in the *Total Width* and *Finger Width* fields. After clicking on *Check and Save*, import the variable into the ADE Explorer window and set the value to 1.

To define parameterization of the transistor width select the introduced variable W in the left side of the ADE Explorer window. Then a button with three dots appears as shown in Figure 41. By pressing on this button a new window called *Parameterize* is opened as shown in Figure 42.

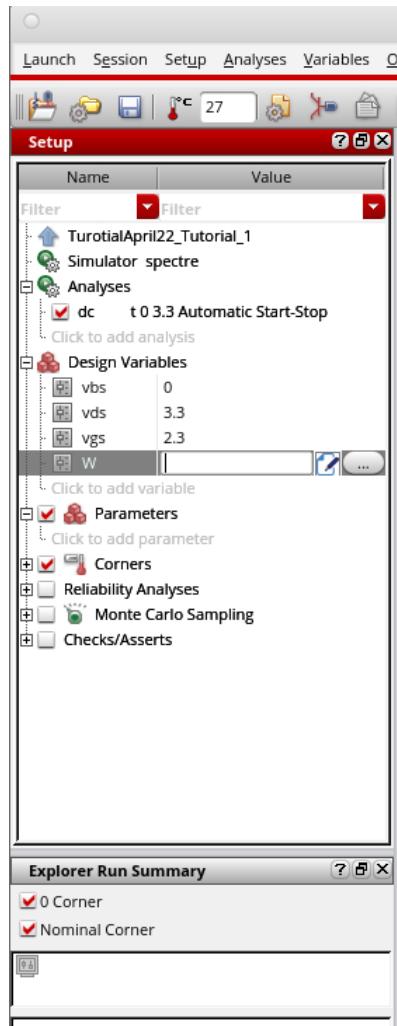


Figure 41: Path to setting up a Parameterize analysis

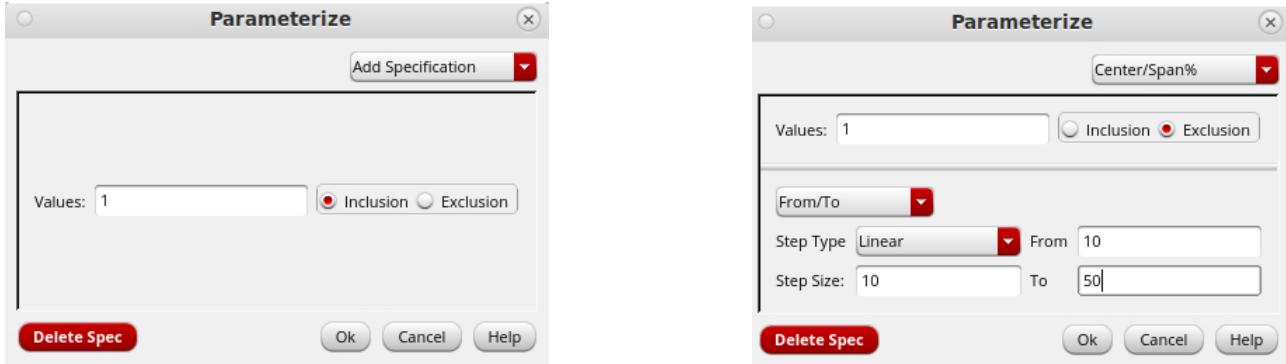


Figure 42: Parameterization setup window

In the *Add Specification* box choose the *From/To* option. The variable should take the values between 10 and 50 in five steps with a step size of 10. For this purpose, the value 10 is entered in *From*, the value 50 in *To* and the value 10 in *Step Size*. Select *Linear* as *Step Type*. Then click on the green *Run* button to start the simulation.

The result of the simulation should be similar to the curves shown in *Figure 43*. According to *Formula (1)* we expect an increase in drain current for larger transistor.

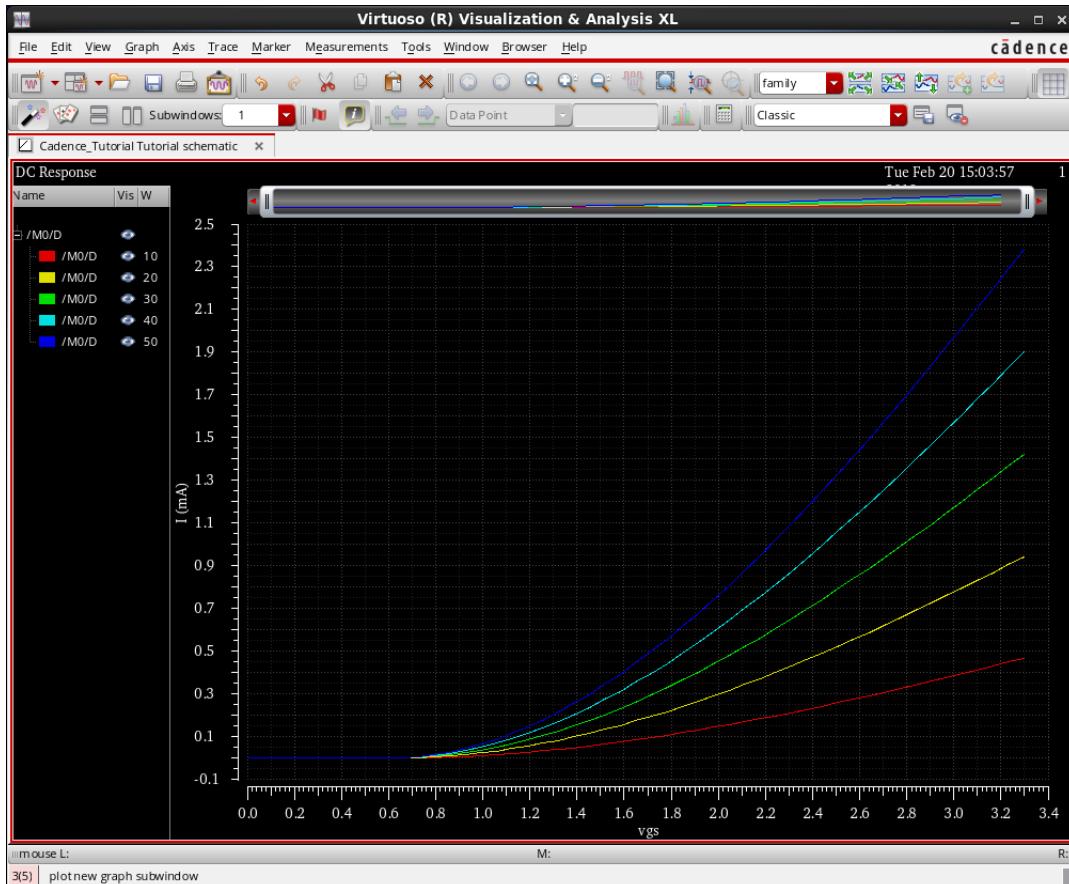


Figure 43: Drain current with variable gate-source voltage and variable transistor widths

7.3 Variation of channel length

Instead of the transistor width, the transistor length should now be varied in the next simulation. Based on *Formula (1)* it can be expected that the drain current decreases with increasing channel length L . The parameterized simulation of the transistor channel length is set up analogously to the procedure for varying the transistor width. In the transistor properties, the factor L is entered in the *Length* field as a multiplier of the minimum length ($340\text{ nm} \times L$). After clicking on *Check and Save*, the variable is imported into ADE Explorer and set to the default value to 1.

In the next step, a new linear sweep is created in the *Parameterized* window analogous to the transistor width from 1 to 9 with a step size of 2. The variation of the transistor width W is deactivated by pressing the *Delete Spec* red button in the parameterize setup window (see again *Figure 42*). The transistor width W thereby receives the standard value ($W = 1$) defined in the ADE Explorer window by clicking on inclusion button. The simulation can then be started by clicking on the run button in the window. *Figure 44* shows the curves for channel length factors L from 1 to 9. As expected, the channel current drops as the transistor length increases.

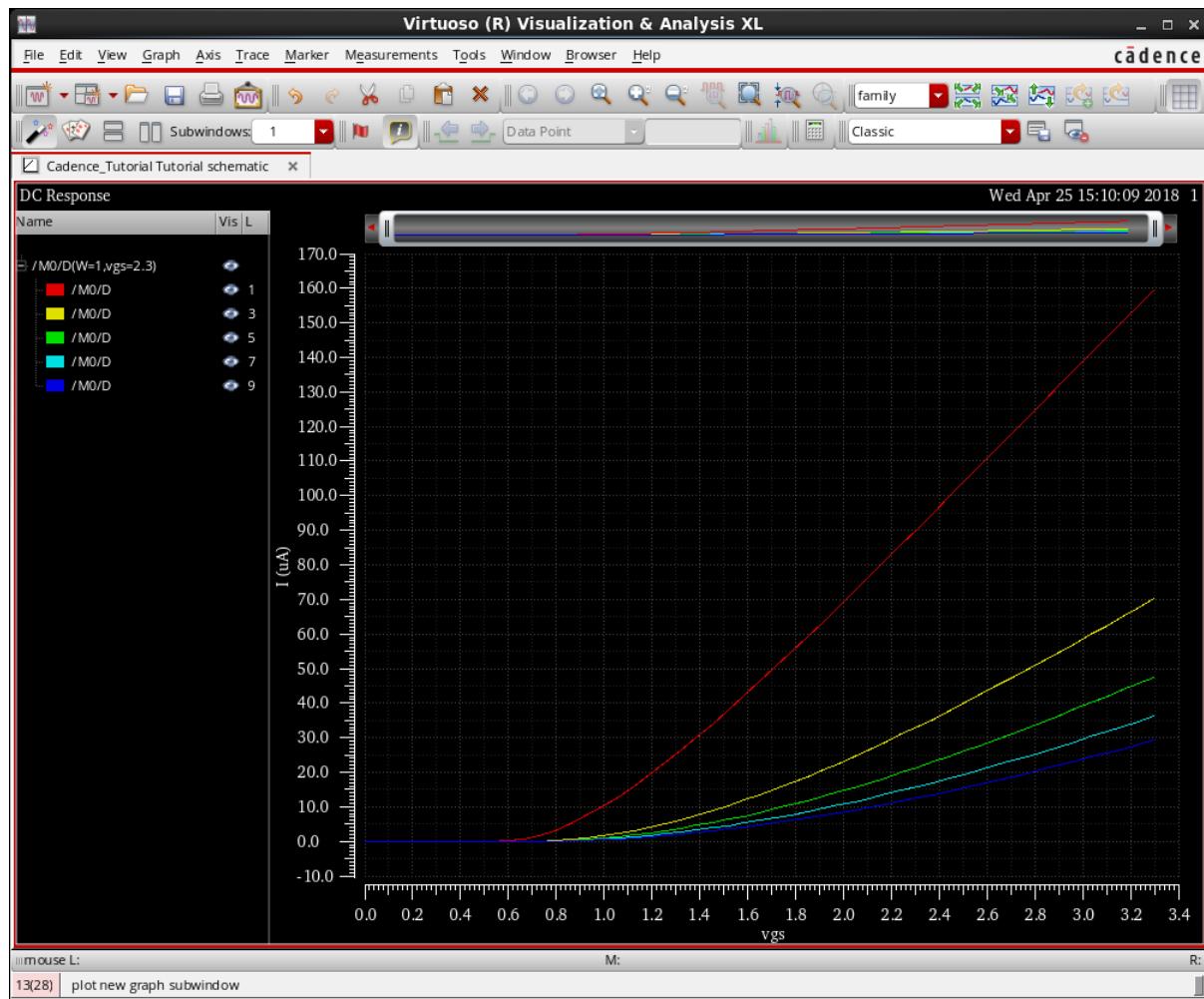


Figure 44: Drain current with variable gate-source voltage and variable transistor lengths

7.4 Sweep of the drain-source voltage

Another interesting simulation that provides an insight into how a MOS transistor works is the so-called output characteristic curve. Here, the drain current is represented as a function of the drain-source voltage (v_{ds}). As the nature of the output characteristic curve is also influenced by the gate-source voltage (v_{gs}), this voltage should also be varied as a parameter. To do this, a sweep is first set up for v_{ds} in the DC simulation in ADE Explorer. Double-click on the DC simulation in the ADE Explorer window to open the *Choosing Analysis* window. In the *Sweep Variable* area, the variable v_{gs} is replaced by the variable v_{ds} . As before, the value interval should be between 0 V and 3.3 V and *Automatic* should continue to be used as *Sweep Type*.

In the *Parameterize* window, the L parameterization has to be deactivated. The additional *Sweep* for the variable v_{gs} should be inserted (see *Figure 45*).

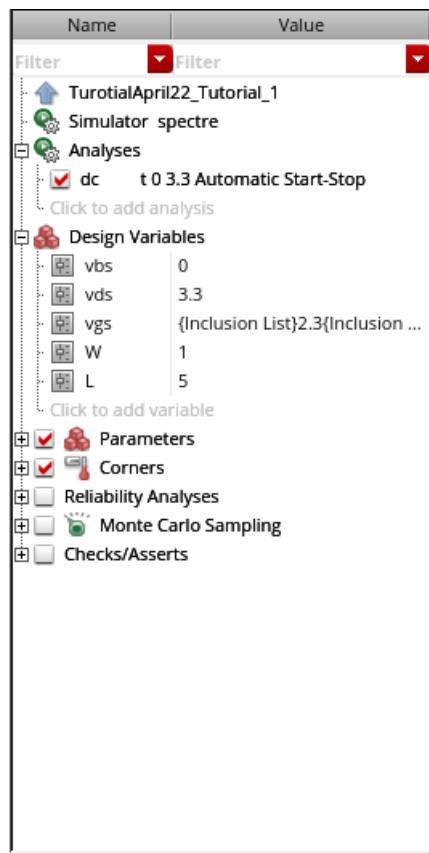


Figure 45: Parameterized with variable gate-source voltages

The gate-source voltage should be run through the interval from 1.3 V to 3.3 V with a *Step Size* of 0.5 V. It is not a good idea to start the simulation at a gate-source voltage of 0 V, as in this case there is only little current flow and only a straight-horizontal line results.

When you run the parameterized simulation, five curves are generated that correspond to DC sweeps over the defined drain-source voltage range with different gate-source voltages as coulter parameters. According to *Formula (1)*, we also expect a higher current through the transistor with increasing gate-source voltages. Figure 46 shows the five simulated curves.

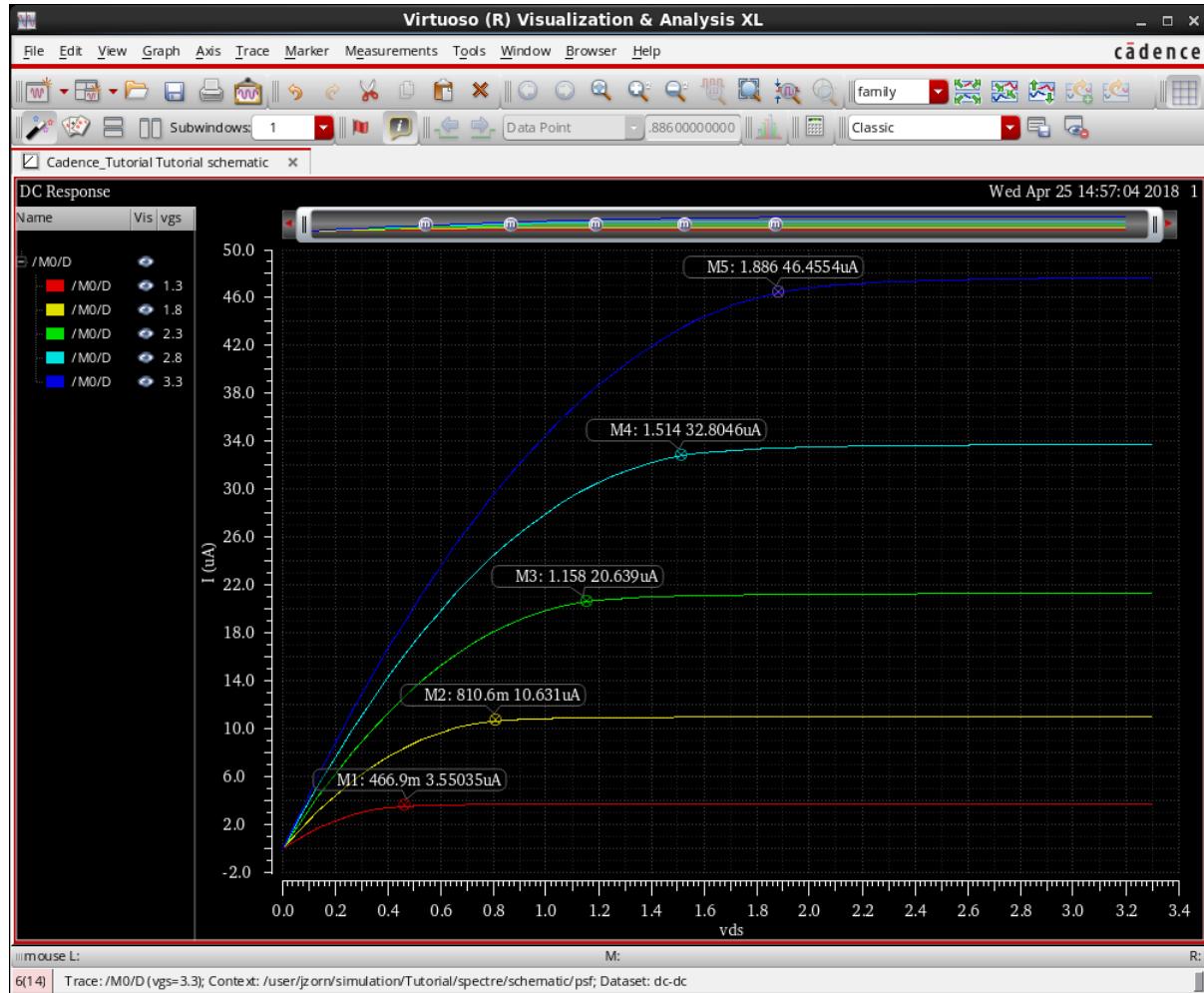


Figure 46: Characteristic curve of drain-source voltages with variable gate-source voltages

The simulation shows that the saturation voltage shifts to higher values for increased gate-source voltages. The saturation voltage V_{DSAT} defines the transition of the current curve from a downward-opening parabola to a constant. It can be calculated using the following formula:

$$V_{DSAT} = V_{GS} - V_{TH} \quad (2)$$

7.5 Variation of channel width and channel length of the transistor

As a final simulation to analyse the influence of the transistor geometry on the current characteristic, the channel width W and channel length L are now to be varied simultaneously at a constant W/L ratio. For this purpose, a factor K is used in the transistor properties (Figure 47) as a multiplier of the minimum width and minimum length.

To proceed, insert the expression $240 \text{ nm} \times K$ into the field *Total Width* and *Finger Width* and the expression $340 \text{ nm} \times K$ into the field *Length*. In ADE Explorer, set the value for $K = 5$.

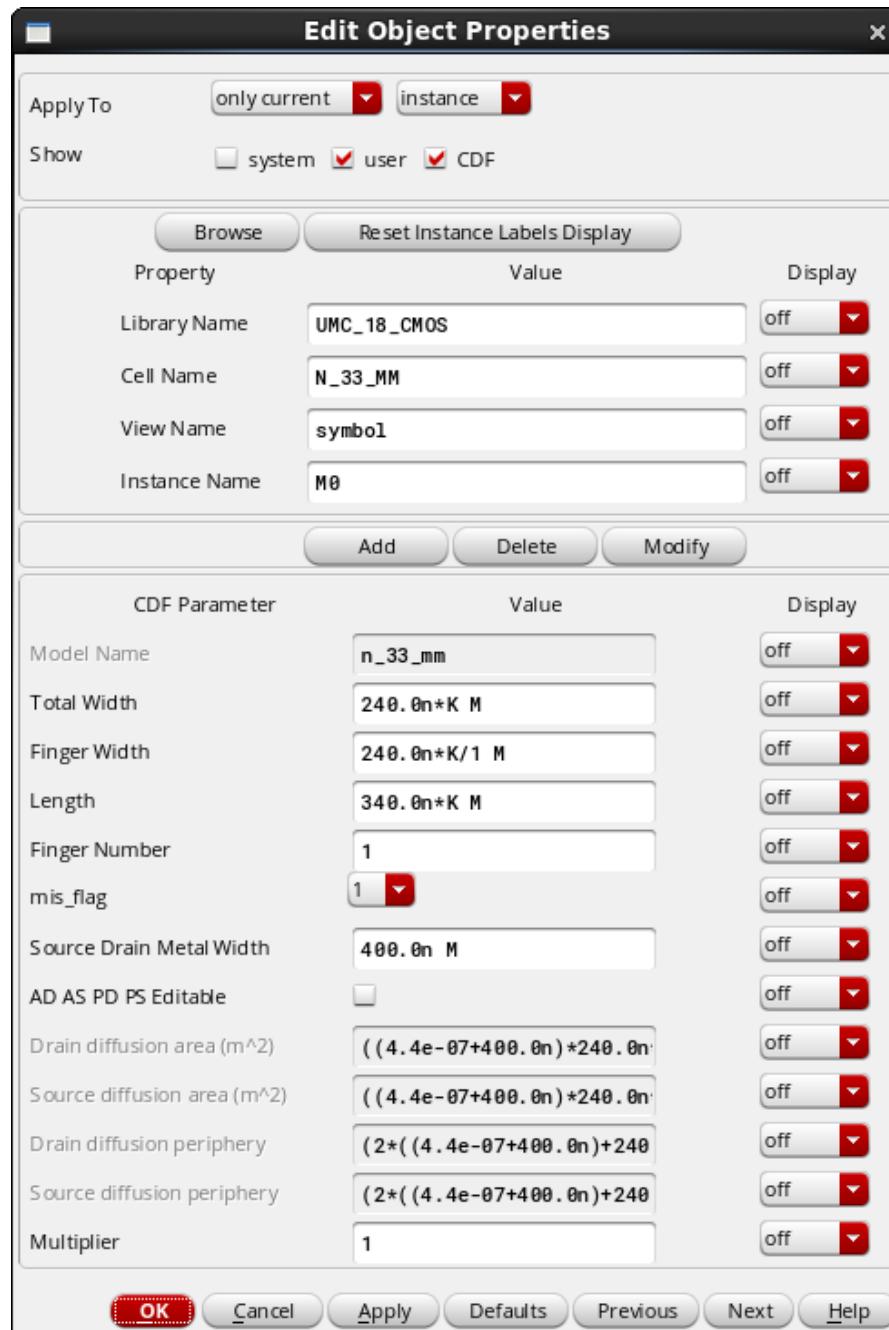


Figure 47: Multiplication of the transistor minimum dimensions

After you have performed a *Check and Save*, deselect the *Sweep* for the channel length L in the *Parameterized* window add a new *Parameterization* for the factor K instead, from 1 to 10 and a step size of 1, and deactivate the sweep for the L parameter as described in previous section.

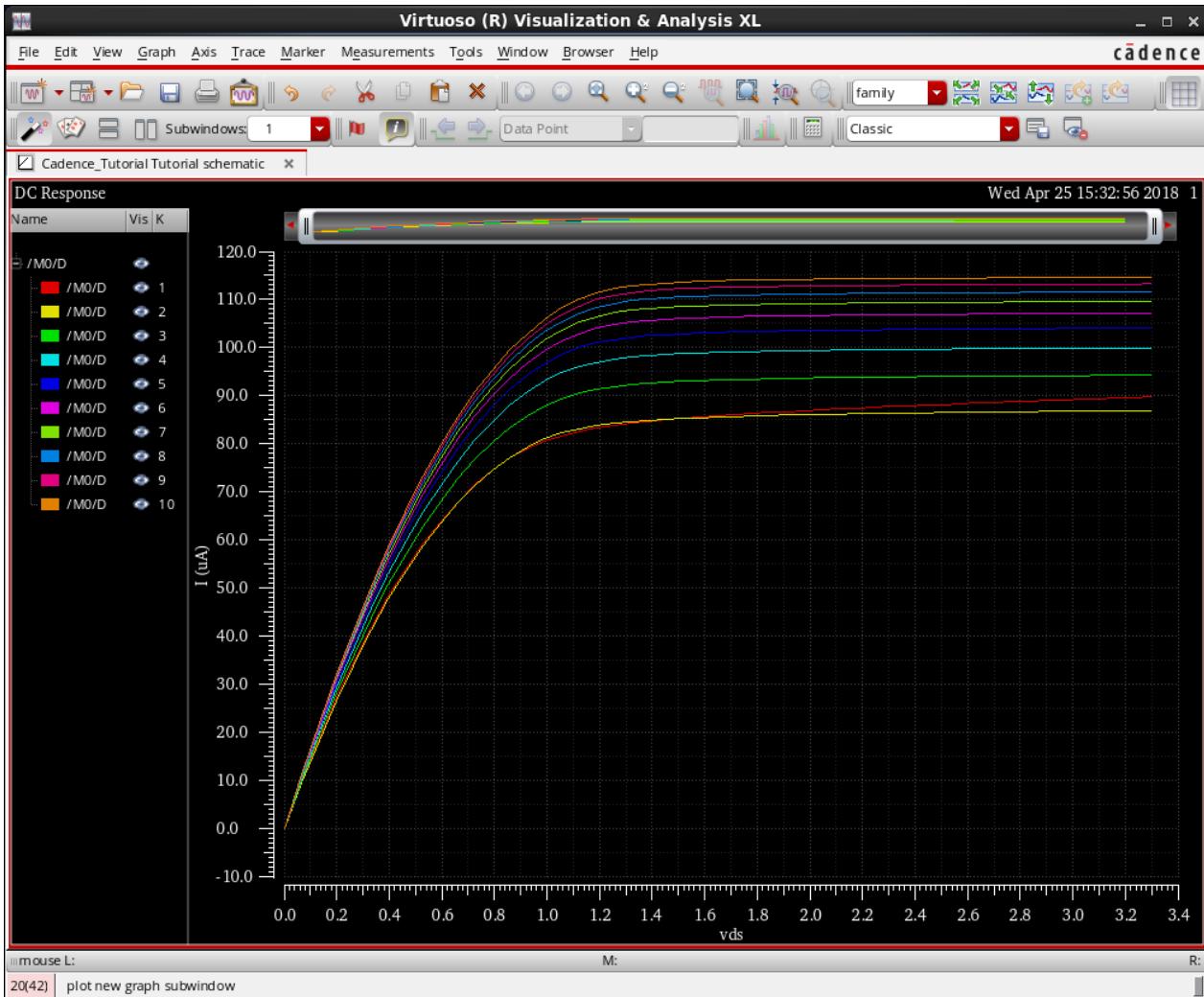


Figure 48: Characteristic curves with increased transistor widths and channel lengths

According to *Formula (1)*, a simultaneous change in the transistor width and length should not affect the transistor current at a constant W/L ratio. However, the simulation result shows that slightly higher currents flow for higher values of K . This is due to the fact that the threshold voltage is dependent on the channel length. Transistors with longer channels have lower threshold voltages than short-channel transistors, which in turn also leads to higher currents at a constant voltage v_{gs} .

Furthermore, it can be seen that the slope of the curves in the saturated range decreases with increasing factor K . The dependence of the channel current on the drain-source voltage v_{ds} in the saturated range is related to the effect of channel length modulation. The shorter the channel, the more the current increases with increasing drain-source voltage. For this reason, transistors with a minimum channel length are very rarely used in analogue circuit design. Transistors with channels that are three to five times longer than the technology minimum are usually used.

7.6 Consideration of the threshold voltage and the body effect

As a further parameter, the threshold voltage v_{th} of the transistor should now be considered in more detail. The threshold voltage can be displayed for the selected operating point in the circuit diagram or also shown graphically as a function of the temperature or other variables.

In this example, the value of the threshold voltage is displayed first. The bulk or *Body effect* is then triggered by varying the bulk source voltage via the variable v_{bs} .

To set up the analysis, the two *Calculator* functions OP and OS are important so that the *Operating Point* (OP) and an *Operating Point Sweep* (OS) can be displayed. How to use the *Calculator* has been described before.

The threshold voltage at the operating point is stored as a simulation result in the *DC operating point* and can be read out using the OP function. Select the threshold voltage v_{th} (see *Figure 49 (a)*) and press OK. Transfer the expression to the ADE Explorer window. To be able to display the threshold voltage even when a DC sweep is running, repeat the above steps, this time using the OS function (*Figure 49 (b)*). Give the expressions meaningful names.

Now activate only the OP expression (or v_{th_wert}) for the threshold voltage in the *Outputs* in the ADE Explorer window and start a DC simulation by clicking on the green button. After running the simulation, the value of the threshold voltage under nominal conditions, i.e. at room temperature ($T = 27^\circ\text{C}$), now appears in the *Outputs* area of the ADE Explorer window in the *Value* column.

Next, the influence of the body effect on the threshold voltage is to be simulated and displayed graphically. The body effect describes a change in the threshold voltage depending on the voltage between the bulk and source. The bulk-source voltage is introduced into the circuit by a voltage source whose voltage is defined by the variable v_{bs} . The bulk-source voltage is to be varied as a DC sweep in a range from -1 V to +1 V.

Open the DC simulation and configure an automatic *Sweep* for the variable v_{bs} in the corresponding value range as in *Chapter 7.1*. From theory, a decreasing threshold voltage is expected for an increasing bulk source voltage. Now also activate the expression OS (or v_{th_sweep}) for the threshold voltage in the *Outputs* of the ADE Explorer window and carry out the simulation.

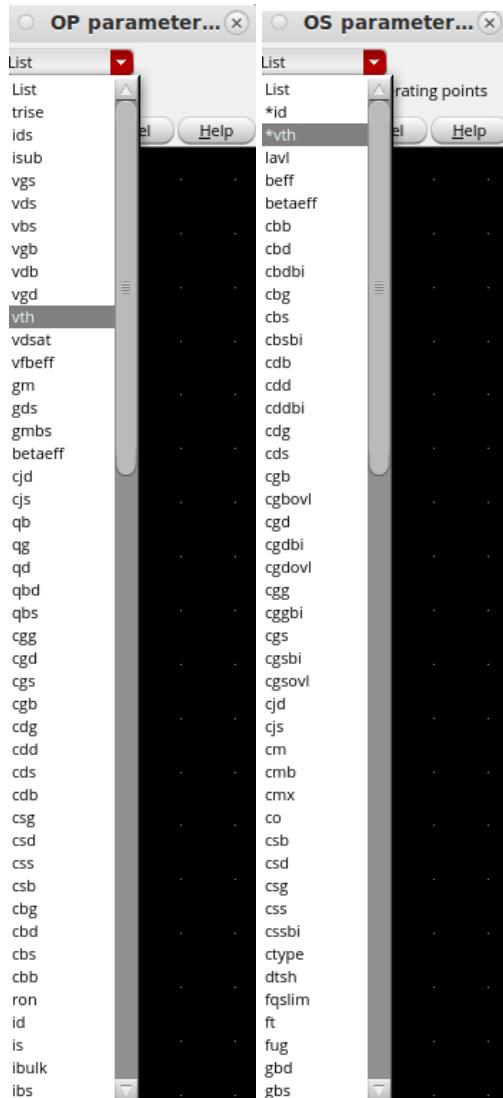


Figure 49: Selection of the threshold voltage via the operating point list (a) OP and (b) OS

The simulation result should correspond to the curve shown in *Figure 51*. Place a vertical line at a value of 0 V for the voltage v_{bs} and compare your values. You can find out how to set a marker in *chapter 8.1*.

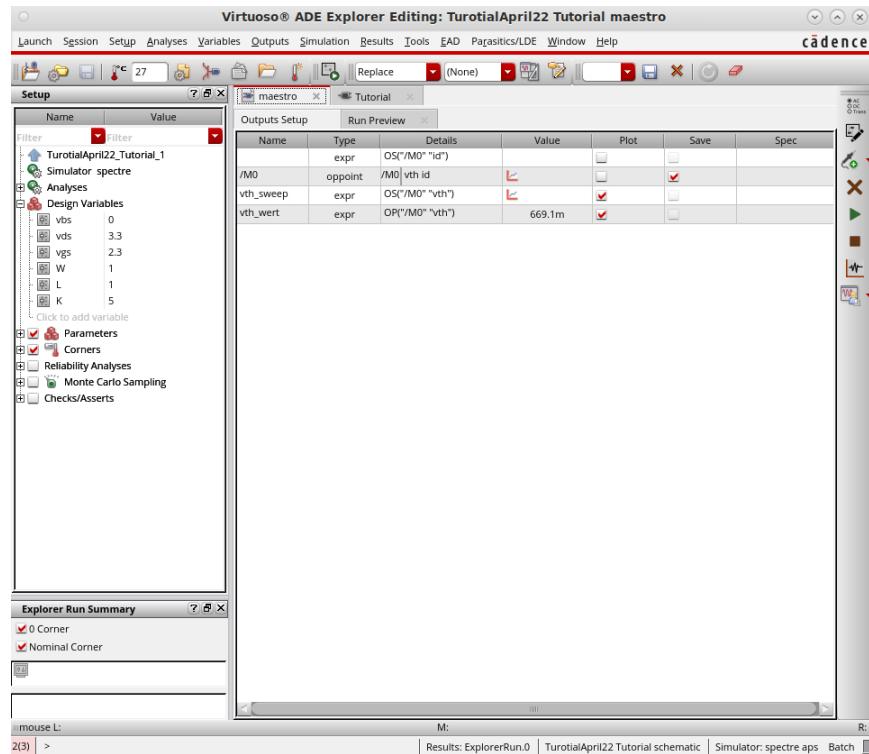


Figure 50: ADE Explorer main window with calculated threshold voltage

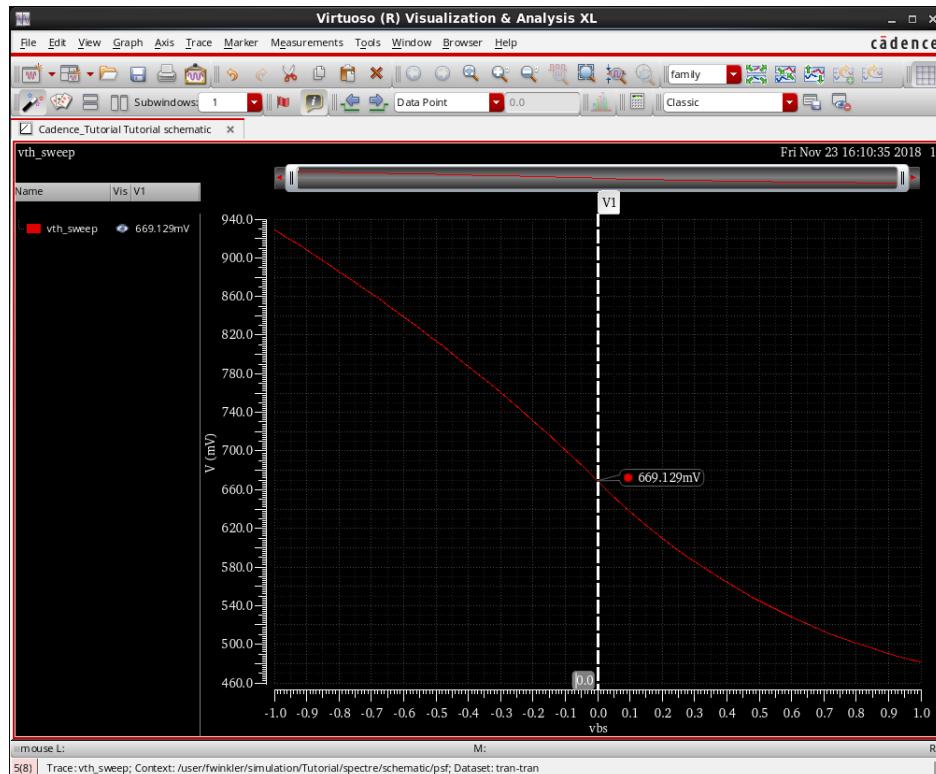


Figure 51: Graphic representation of the body effect with swept source-bulk voltage

7.7 Temperature influence on threshold voltage and charge carrier mobility

As the value of the threshold voltage is also temperature-dependent, this relationship should now be simulated and displayed graphically. In addition, the temperature dependence of the charge carrier mobility μ is also to be investigated. To do this, the temperature in the *Choosing Analyses* window is to be varied as a DC-Sweep. Then define a sweep interval in the range *Sweep Range* from -40 °C to 120 °C using the *Sweep Type Automatic*.

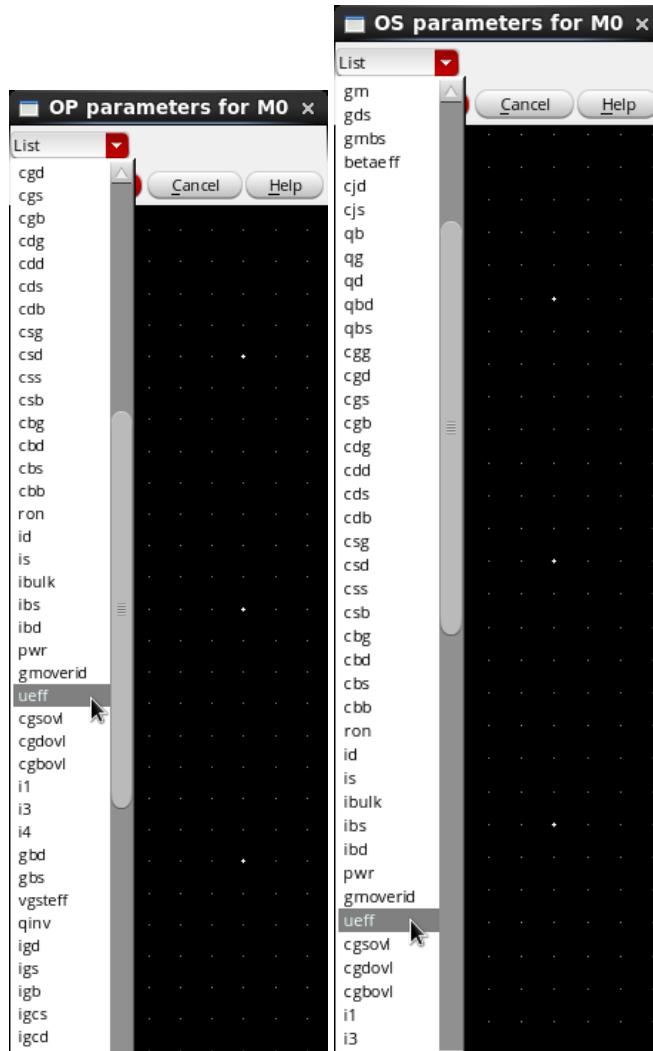


Figure 52: Selection of the load carrier via the operating point list (a) OP and (b) OS

Since the expressions for the threshold voltage variation have already been introduced as *Outputs*, only the corresponding expressions for the charge carrier mobility need to be added. To do this, open the *Calculator* and first click on *OP* and then on the transistor in *Schematic*. Then select the size *ueff* from the list shown in Figure 52. Repeat this step by first selecting the function *OS* in *Calculator*. Insert these two expressions into the *Outputs* of ADE Explorer and give them a meaningful name.

Now run the DC simulation by clicking on the green button in the ADE Explorer window. After running the simulation, the graphs of the charge carrier mobility and the threshold voltage should be displayed as a function of temperature as shown in Figure 53.

It can be seen that both the threshold voltage and the charge carrier mobility decrease with increasing temperature. The reason for the reduction in threshold voltage is stronger atomic movement in the crystal structure

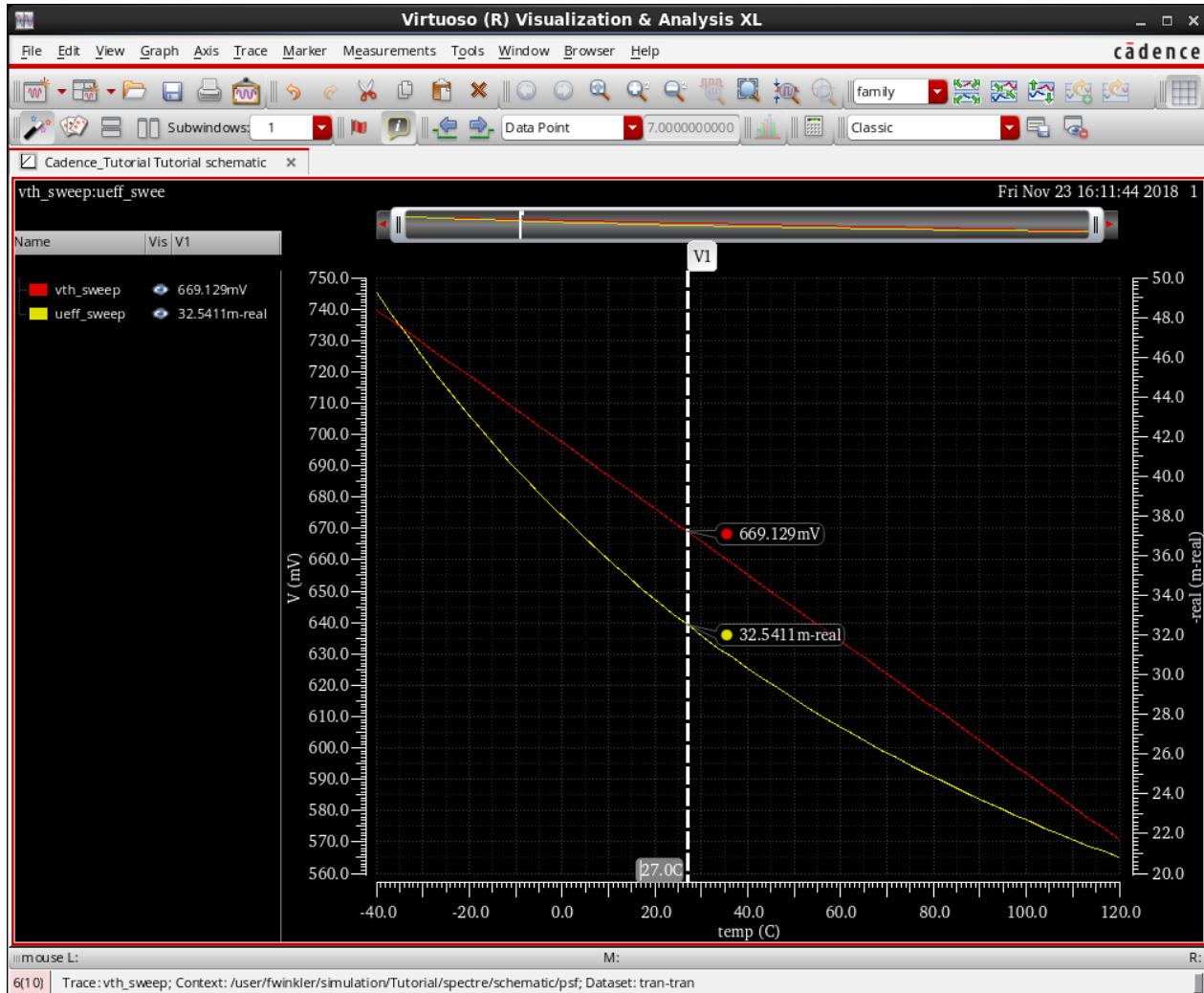


Figure 53: Threshold voltage and charge carrier behaviour as a function of temperature

of the semiconductor material, which means that charge carriers can be released from this structure more easily, resulting in higher conductivity of the channel due to a higher number of charge carriers. Similarly, the charge carrier mobility decreases with increasing temperature, as the stronger movement of the atoms in the crystal leads to more frequent collisions between the charge carriers and the atoms.

The influence on the transistor characteristic is complementary. According to formula (1), the decreasing threshold voltage leads to an increase, while the decreasing charge carrier mobility leads to a decrease in the transistor current. Which effect prevails depends on the specific operating point of the transistor.

8 Settings in the Visualisation & Analysis window

8.1 Use of markers in the simulation window

Markers can be set in the simulation window for precise evaluation. This makes it possible to compare the calculated values with the simulation result. Markers can be created either via the menu item *Marker > Create Marker* (see *Figure 54*) or via the keyboard shortcuts listed in *Table 1*.

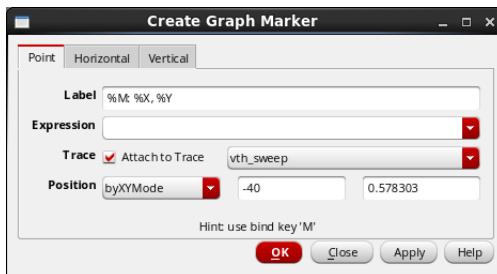


Figure 54: Window for creating markers

There is also the option of displaying differences between two points marked in a curve. To do this, generate the first marker using the corresponding shortcut for the desired marker type. Then generate a second marker using the shortcut **[D]**. Once the markers have been created, they can be moved and deleted as required.

Tastenkürzel	Function
[M]	Creates a point
[H]	Creates a horizontal straight line
[V]	Creates a vertical straight line
[D]	Creates a marker and displays the difference to the selected marker

Table 1: Keyboard shortcuts and their function in the simulation window

The vertical and horizontal lines can be moved to the desired positions by clicking on the marker and entering the value in the field that appears on the X or Y axis. For point markers, this setting can be called up by double-clicking on the marker.

The following *Figure 55* shows an example of the simulation carried out in *Chapter 7.7* with a selection of different markers.

The vertical line *V1* is created by pressing the **[V]** key. The second vertical line *V2* is generated using the keyboard shortcut **[D]** and shows the distance to the previously created vertical line *V1*. Alternatively, pressing the **[V]** key again would have created a vertical line independent of *V1* without calculating a difference.

The horizontal line *H1* is created using the keyboard shortcut **[H]**.

The points *M3* and *M4* are created by pressing the **[M]** key and then the **[D]** key. The difference between the two marker points is also displayed here.

If you were unable to place a marker at the exact desired value when creating it, click on it and enter the corresponding value in the field.

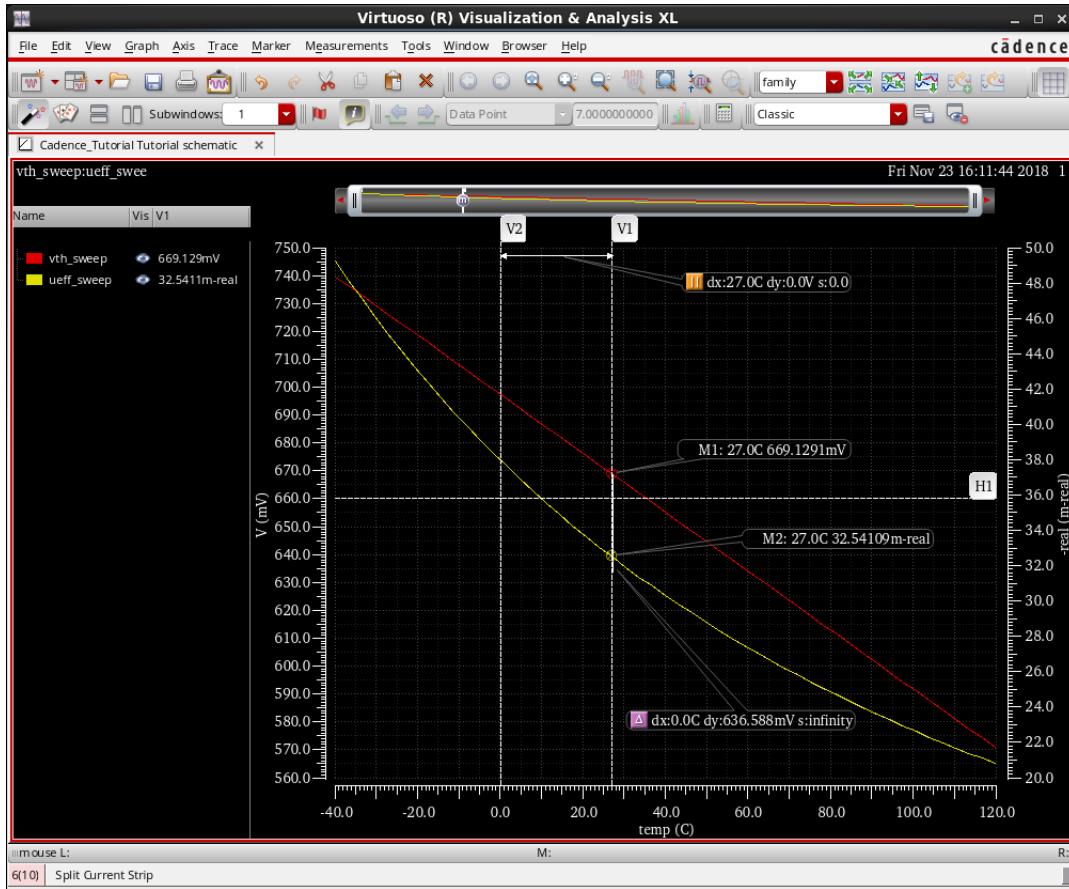


Figure 55: Created markers in simulation window

8.2 CSV-Exports generation

Cadence offers the option of exporting simulation results as a CSV file so that they can then be imported into Excel or similar programmes. To do this, click on *Trace* in the menu bar in the *Visualisation & Analysis* window after a completed simulation and then select *Export* from the selection menu that opens.

You can now select a storage location and a file type for your export in the *Export Waveforms* window. In this case, select the CSV setting and leave all other settings unchanged. These additional setting options allow you to export a curve range.

The exported file can then be further processed in spreadsheet software.

8.3 Export of plots in image form

To convert a plot into an image file, press *File > Save Image* in the *Visualisation & Analysis*. This opens the window of the same name.

Here you will find various settings for the format, the size of the image (section *Resize Image(s)*), the selection of multiple analyses in one window (section *Save*), as well as various color settings and display configurations (section *Graph Display*). The *Format Options* window (Figure 59) contains further settings to optimize the image for Microsoft Office and adjust the quality.

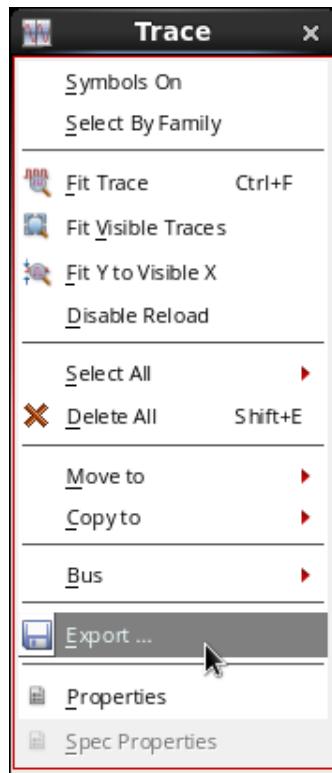


Figure 56: Menu item for exporting the simulation results



Figure 57: Window for saving Waveforms

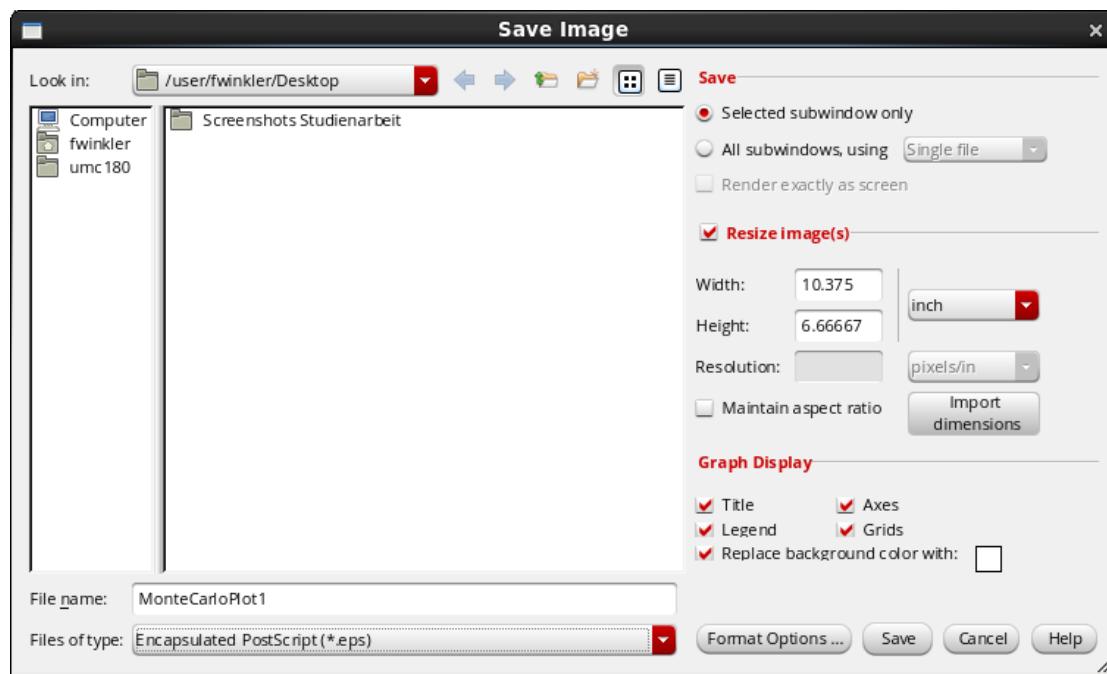


Figure 58: Window for saving a *Plots*

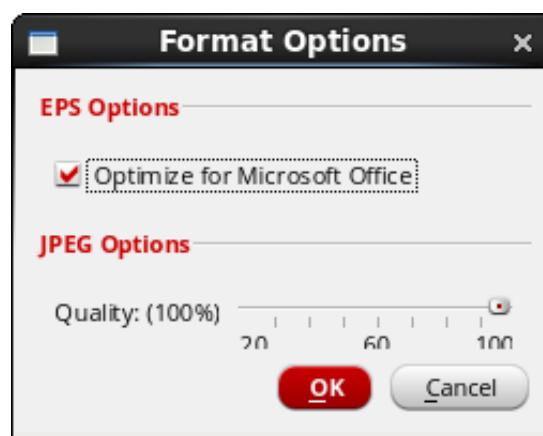


Figure 59: Option window for the *Plot-Export*

9 Structure of an amplifier

9.1 The inverting amplifier

In this section, we will now design a new circuit in which DC and transient analyses can be carried out. The circuit is to be constructed as shown in the following (see Figure 60).

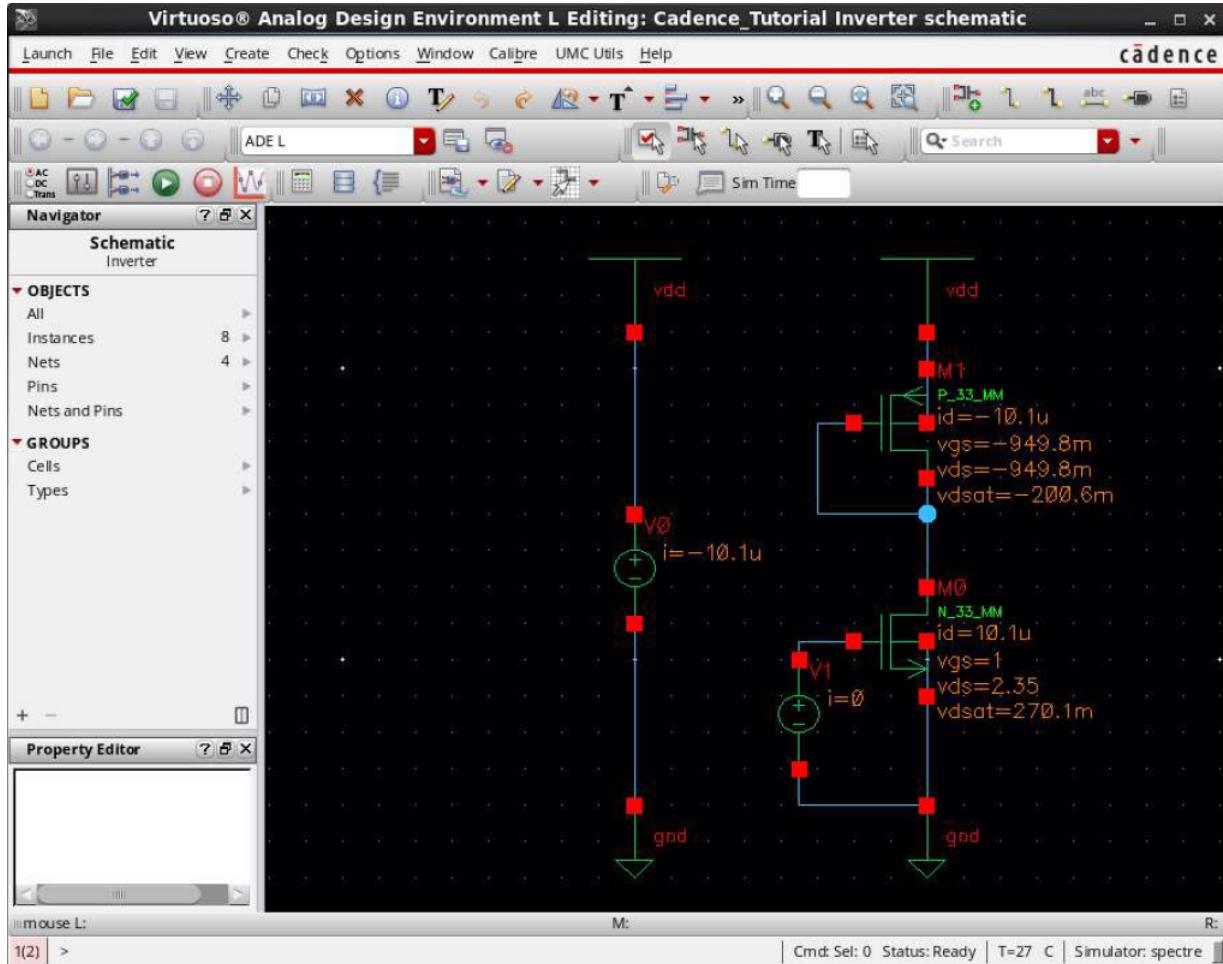


Figure 60: Structure of the inverter circuit with external voltage source

The circuit shown corresponds to a simple inverting amplifier consisting of a series connection of an NMOS and a PMOS transistor. This circuit amplifies the input signal and performs an inversion. Inversion means that the sign of the signal deflection is changed. If the input signal increases, the output signal decreases and vice versa, i.e. the sign of the deflection changes.

Create a new *Schematic* with the name „Inverter“. How new *Schematics* are created is explained in *Chapter 3*.

For the circuit, an NMOS transistor and a PMOS transistor of type *N_33_MM* or *P_33_M* from the *UMC_180_CMOS* library are required. In addition, voltage sources of type *vdc* and the cells *vdd* and *gnd* from the *analogLib* are used.

The power supply should be connected to the circuit via global networks. For this purpose, the voltage source is connected with the positive pole to the *vdd* cell and with the negative pole to the *gnd* cell. The *vdd* network corresponds to the positive supply voltage, while *gnd* represents the ground node. At all points where the *vdd* and *gnd* terminals are attached in the circuit, there is a connection to the supply voltage or to ground respectively.

Place the components and connect them as shown in *Figure 60*. How components are placed and wired is described in *chapter 5.1*. When wiring, make sure that the bulk and source of the transistors are short-circuited. To set the operating point, place the voltage source *vdc* between the gate and source of the NMOS transistor and set a voltage of 1 V for this. Give the voltage source a voltage of 3.3 V and set the channel lengths of both transistors to five times the minimum length ($L = 5 \times 340.0 \text{ nm} = 1.7 \mu\text{m}$).

Open ADE Explorer and carry out a simple DC simulation and display the DC operating points of the circuit as already described in *Chapter 6.4*. Display the current (*id*) and the saturation voltage (*vdsat*) on both transistors, as you will set these variables via the transistor dimensions.

The width of the NMOS transistor should now be set so that the current through both transistors is set to $id = 10 \mu\text{A}$. The transistor width of the PMOS transistor should then be set so that a saturation voltage of $vdsat = 300 \text{ mV}$ results in order to achieve an amplification greater than 1. As both transistors are connected in series, the current through both transistors is always the same. In addition, the PMOS transistor is always in saturation due to the gate-drain connection.

This results in the following transistor widths to be set:

- $W_{NMOS} = 2.1 \mu\text{m}$
- $W_{PMOS} = 7.5 \mu\text{m}$

After adjusting the transistor widths and all other variables, the voltage source between the gate and source of the NMOS transistor can be removed from the circuit again to set the operating points. Leave the connection at the gate of the NMOS transistor open for the time being. A pin should be connected here.

9.2 Creation of pins in the schematic

Pins are used for the implementation of hierarchical circuit diagrams and the creation of symbols and represent the connection contacts of the circuit. In the following step, the inverting amplifier is to be expanded to include pins for the input and output signal and the power supply. As a result, the remaining voltage source is no longer required and can be removed along with the *vdd* and *gnd* connections. The open lines are connected to pins instead. To create a pin, press *Create Pin* in the bar for symbol elements of the editor or alternatively use the keyboard shortcut **[P]**. A window for setting up pins opens (*Figure 61*).

In the *Names* field, the pin can be assigned a name. The *Direction* field determines whether the pin should be an input (*input*), output (*output*) or a bidirectional (*inputOutput*) pin.

Place the pins in the circuit as shown in *Figure 62*. To do this, you need a *inputOutput* pin for the power supply (*vdd*) and the ground connection of the inverting amplifier (*gnd*), as well as an *Input* pin at the gate connection of the NMOS transistor (*In*) and an *output* pin (*Out*), which is connected to the drains of the NMOS and PMOS transistors.

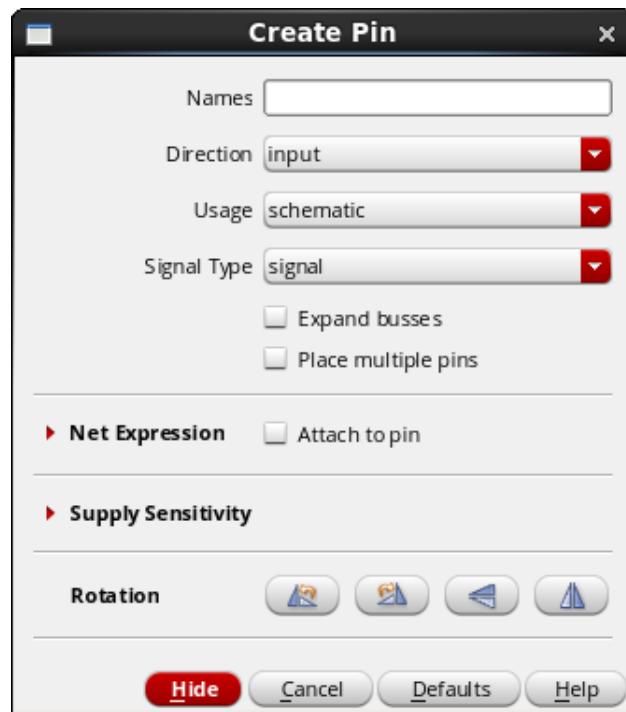


Figure 61: Window for creating pins

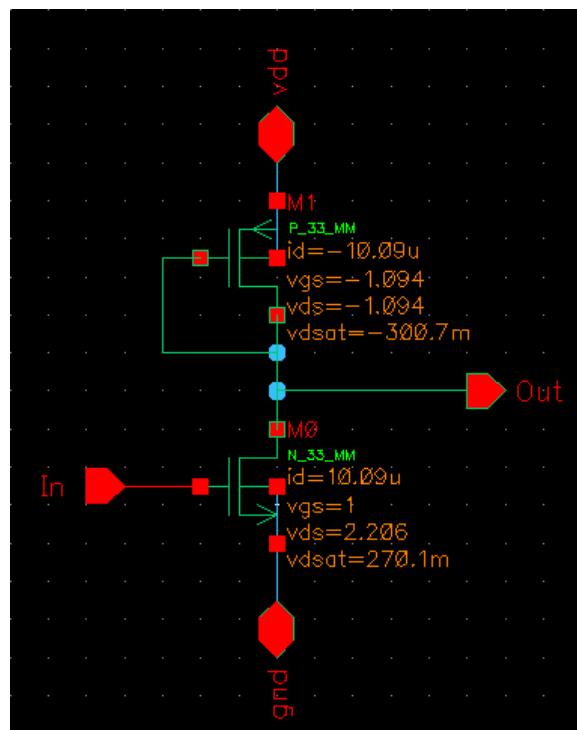


Figure 62: The inverting amplifier with pins on the connections

10 Use of symbols

10.1 Creating a symbol

Once the pins have been placed, the circuit is ready for symbol creation. To do this, click on the menu item *Create > Cellview > From Cellview* in the Schematic window (Figure 63).

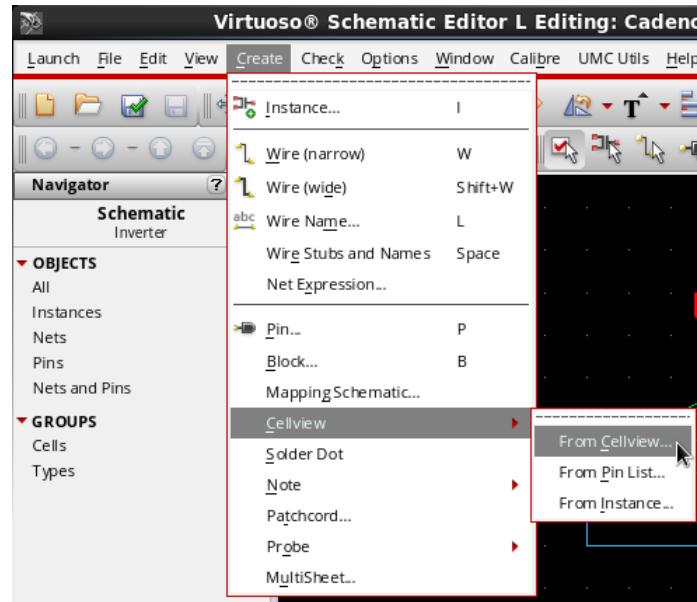


Figure 63: Menu items for creating a symbol

The name of the symbol can be specified in the window that opens. By default, the correct names of the *Library* and the *Cells* should already be specified here. Make sure the *To View Name* field contains the name *symbol* in order to follow the standard naming conventions for view names.

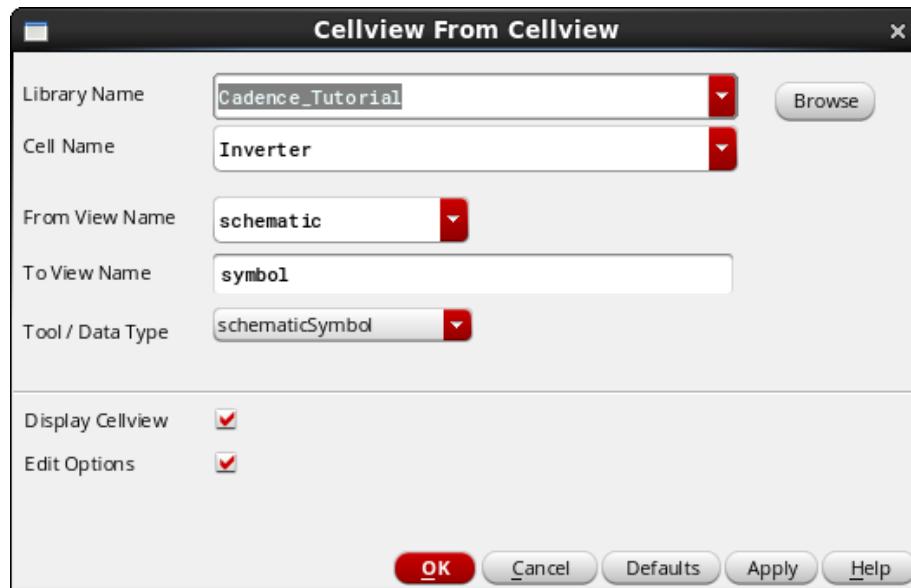


Figure 64: Options for creating a symbol

Click on OK to open a window in which the connection location of the pins in the symbol can be defined. Insert the pins into the fields as shown in the following *Figure 65* and then confirm the action by clicking on OK.

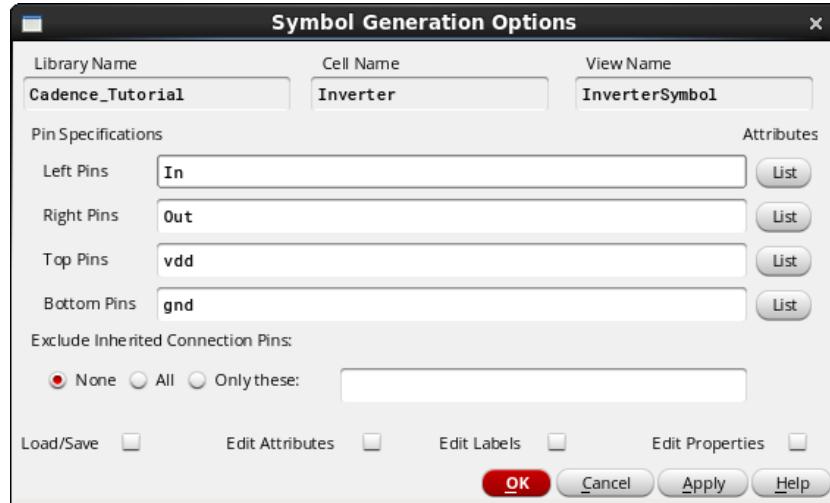


Figure 65: Definition of the pins in the symbol

A simple rectangular symbol is then automatically generated and opened in the Symbol Editor window for further editing (*Figure 66*).

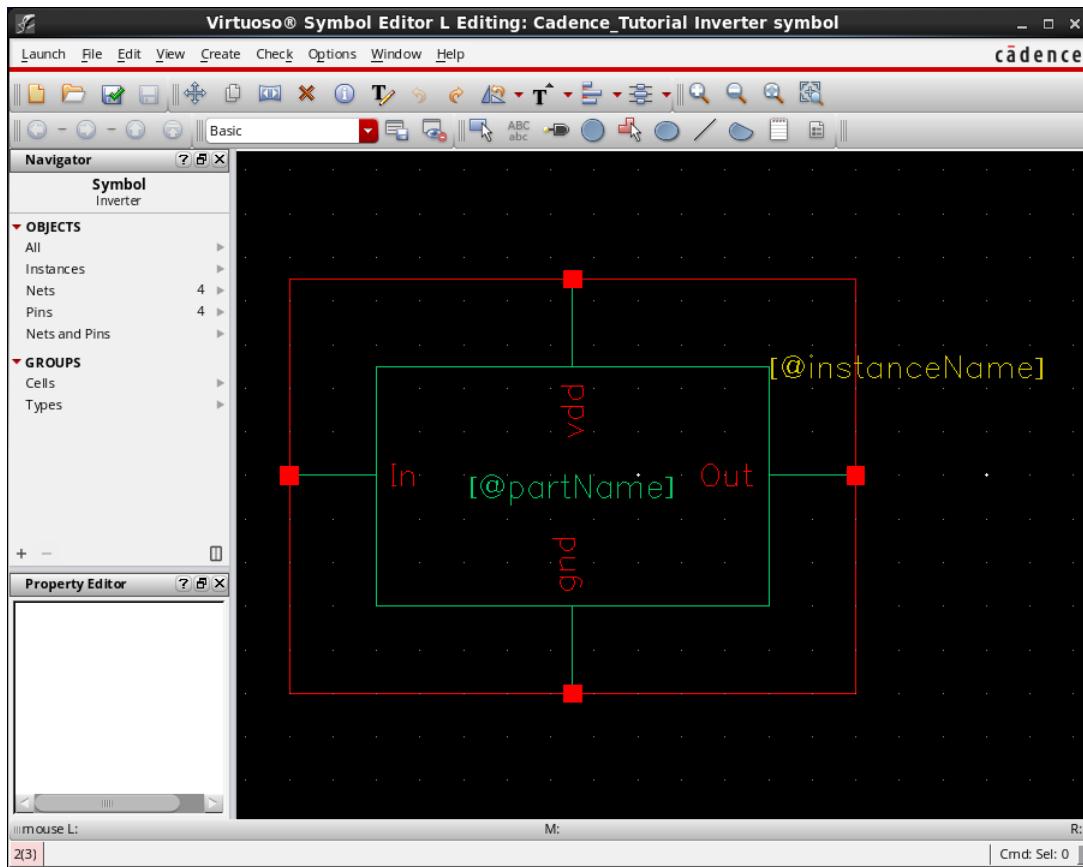


Figure 66: The Schematic-Window when creating a new symbol

You can see the symbol with a green border and small red square dots that correspond to the previously defined

pin assignment. The red border defines an area in the form of a rectangular box in which no overlapping with other symbols or wiring is permitted in the *Schematic*. This box should always correspond to the actual size of the symbol and include all symbol elements. The text fields `[@InstanceName]` and `[@PartName]` are display variables that represent the instance or cell name in the *Schematic*.

A triangle is now to be created as an amplifier symbol. First remove the green rectangle and pressing the **[Del]** key or selecting the corresponding entry in the *Edit* menu. There are various options for creating geometric figures in the toolbar of the *Schematic* window. You can choose between circles, ellipses, rectangles and lines. In principle, it is also possible to combine any number of these standard shapes in different sizes and orientations to create symbols.

With *Create Line* you can draw the individual segments of the triangle. To do this, start the first line by clicking on a point, e.g. the starting pin. Move the mouse to the end of the line and click again to mark the end of the line. Then draw the next two sides of the triangle in the same way. When drawing the last point that closes the triangle, double-click to indicate the end of the editing operation. The drawing process can also be interrupted beforehand by pressing the **[ESC]** key and drawn lines can be deleted or undone using the *Undo* function. If the lines do not fit exactly on the triangle, you can select and move or extend them. Then adjust the red border so that it encloses the symbol (see *Figure 67*).

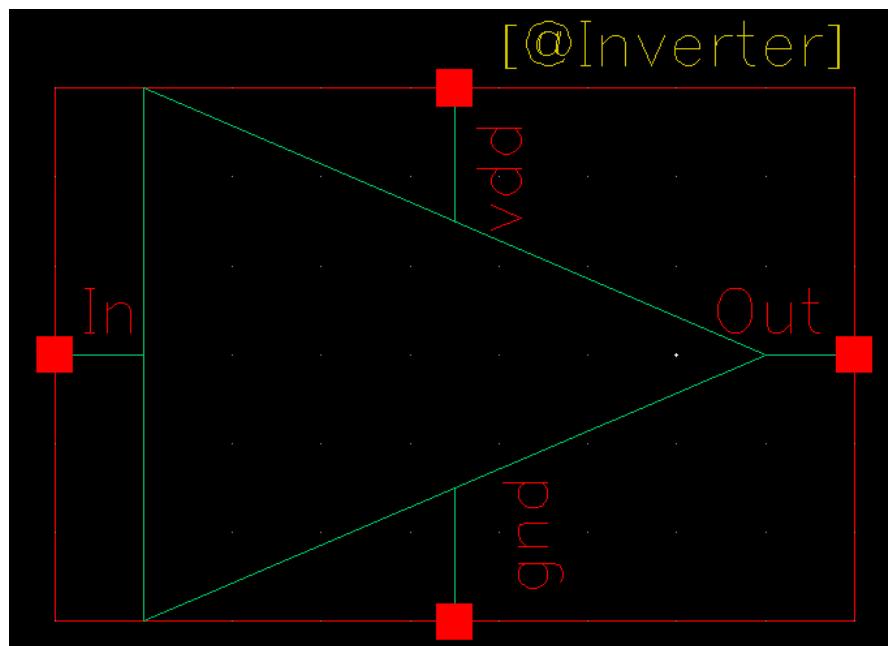


Figure 67: The finished switching symbol of the inverting amplifier

Finally, save the symbol by clicking on *Check and Save*.

10.2 Use of a symbol in a schematic

Now create (as described in *Chapter 3*) a *Schematic* in a new cell with the name „Inverting circuit“. The symbol of the inverting amplifier is to be used in this *Schematic*. The created symbol can be placed in the *Schematic* via *Create Instance* or the keyboard shortcut [I] after selection in the *Library Browser*, just like pre-defined or manufacturer-specific components. In this case, the view *symbol* must be selected in the library *Cadence_Tutorial* and the cell *Inverter* (*Figure 68*).

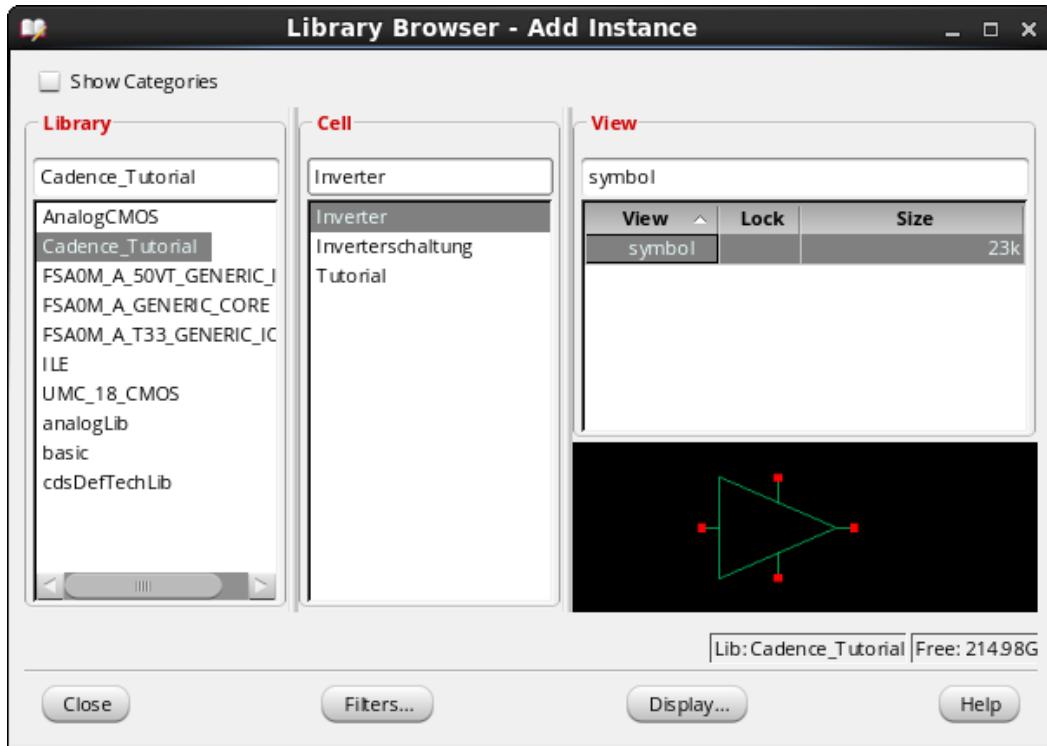


Figure 68: The icon created in the Library Browser

Place the symbol in the *Schematic*. To complete the circuit, you need a constant voltage source (*vdc*) as well as a pulsed voltage source (*vpulse*) and a ground connection (*gnd*). You need an open net for the output of the circuit. This is achieved by placing a *noConn* component which is located in the *basic* library. Place *Labels* (*In*) at the input and (*Out*) at the output. Build the circuit as shown in the *Figure 69* on the next page.

In the DC voltage source, enter the variable *vsup* in *DC voltage*. The following values must be set for the pulsed voltage source

In ADE Explorer, after clicking on *Check and Save*, set the value for *vdc* to 3.3 V and for *vin* to 1 V. In transient simulations, the pulsed voltage source generates a square-wave signal with a defined period duration, sampling width, rise and fall time. In a DC simulation, the value specified in the *DC Voltage* field is generated.

Parameter	Wert
DC voltage	$vin\ V$
AC magnitude	1V
AC phase	0
Voltage 1	1V
Voltage 2	1.2 V
Period	1m s
Rise time	1n s
Fall time	1n s
Pulse width	500u s

Table 2: Values of the pulsed voltage source

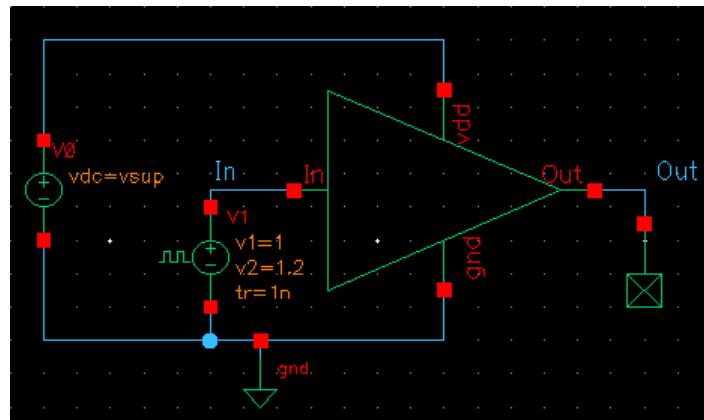


Figure 69: The complete test circuit

10.3 Display/edit a symbol in the schematic

After placing the symbol in the *Schematic*, it is possible to open the circuit behind it, which in this example corresponds to the inverting amplifier, and thus descend one hierarchy level in the *Schematic*. To do this, select the symbol in the *Schematic* by clicking on it and press the key combination **[Shift]+[E]** or just **[E]**. A window opens (Figure 70) in which you can select whether the circuit should be opened in the current window and tab, or in a new window. You also have the option of opening the circuit read-only (*read*) or editable (*edit*). To return from the inverter to the *Schematic* above, press the key combination **[CTRL]+[E]**.

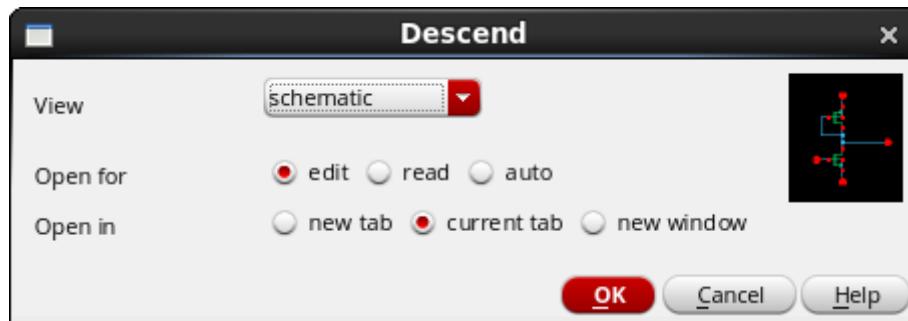


Figure 70: Descending in schematic hierarchy

11 Transient analysis in ADE Explorer

This chapter introduces transient analysis as a further simulation method. If you have not already done so, open the previously created circuit diagram of the inverter circuit and start ADE Explorer. To configure a transient analysis, open the *Choosing Analyses* window in the ADE Explorer simulation environment and select the *tran* field. A simulation time is entered in the *Stop Time* field, which in this case should be 3 ms.

When analysing transients, it is possible to improve the accuracy of the analysis at the expense of the duration of the simulation. The following three setting options are available in the *Accuracy Defaults (errpreset)* area.

- *conservative*: slower simulation speed, higher accuracy
- *moderate*: Moderate speed and accuracy
- *liberal*: Fast simulation speed, lower accuracy

Select the option *conservative*, as the simulation time is very short for a small circuit despite the high accuracy. If you do not select any of these fields, the simulation is carried out with the setting *moderate* by default. Then select the *Enabled* field and confirm the simulation setup by clicking on *OK*.

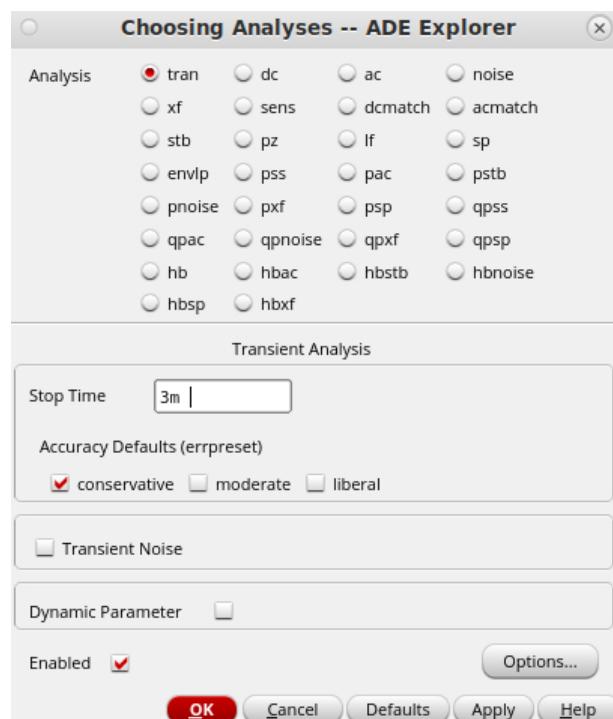


Figure 71: Setting up a transient analysis

As with previous simulations, you also need to define *Outputs* that you want to plot. Firstly, the input and output signal of the circuit should be chosen as *Output*. Open the *Setting Outputs* window known from Chapter 6.6. Click on *To Be Plotted > Select On Design* and select the input and output lines. Confirm by pressing [ESC]. Then start the transient analysis by clicking on the *Run Simulation* icon.

The analysis window opens in which the graphical curves of the input and output signal of the inverter are displayed. The green diagram curve represents the output signal, while the red curve corresponds to the input signal. To make the diagrams clearer, click on *Split Current Strip* in the menu bar of the *Visualisation & Analysis* window. This function shifts these two curves into separate horizontal image sections, so-called strips, and improves clarity by adjusting the scaling of the axes, among other things. The diagrams can be merged again by clicking on *Combine All Analogue Traces*.

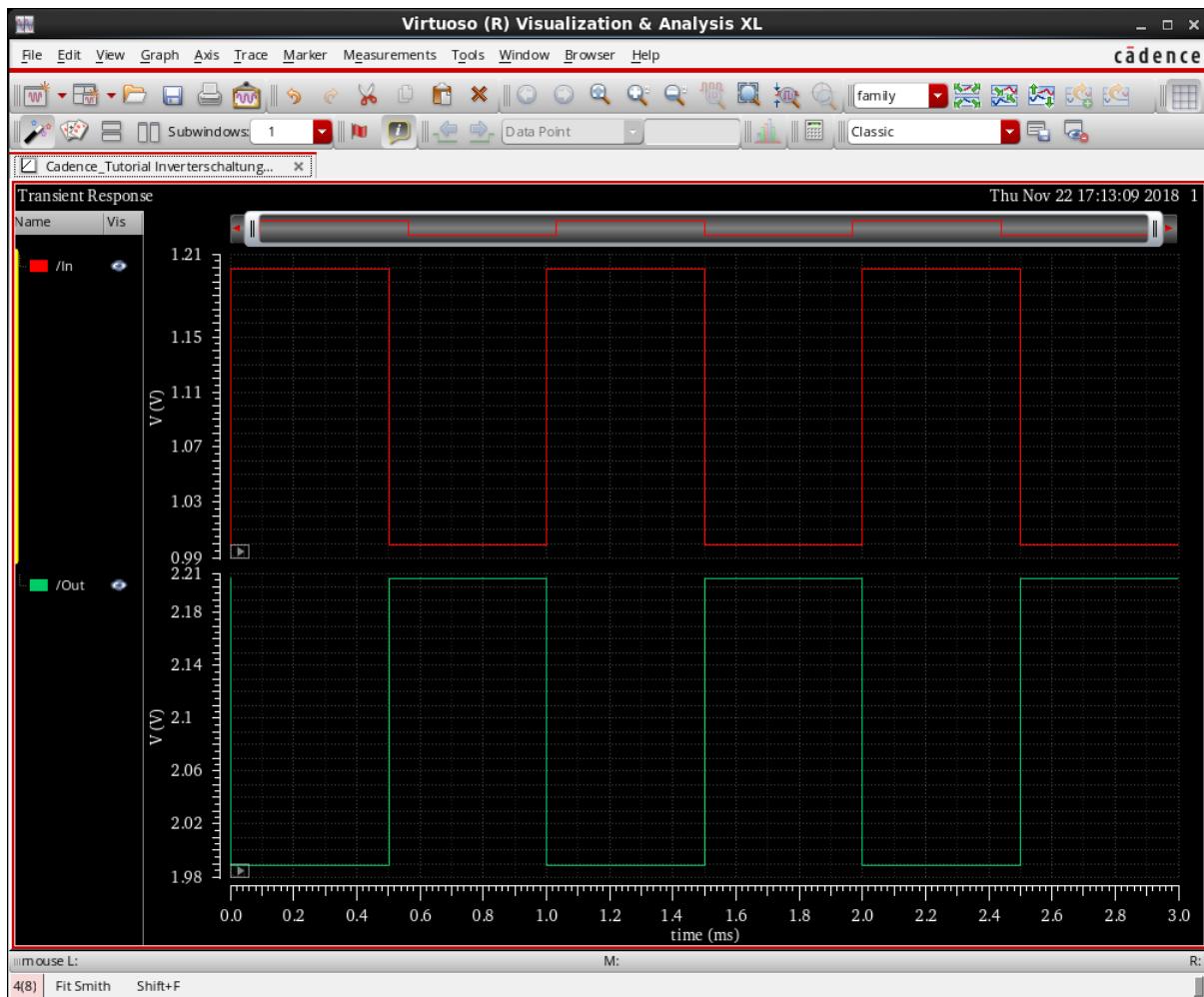


Figure 72: The result of the transient analysis

It can be clearly seen that the pulsed output signal is inverted compared to the input signal.

12 AC analysis in ADE Explorer

The next simulation presented in this tutorial is the AC analysis.

12.1 Preparing the circuit for an AC analysis

First delete the *noConn* connection at the circuit output. Connect a capacitor to the now open line of the inverting amplifier. To do this, press the [I] button in the circuit diagram editor and select the component *cap* from the *analogLib* library. After placing and connecting the capacitor as shown in Figure 73, press [Q] and set the parameter *Capacitance* to 1 nF . If you have not already done so, also set the parameter *AC magnitude* of the pulsed voltage source to 1 V .

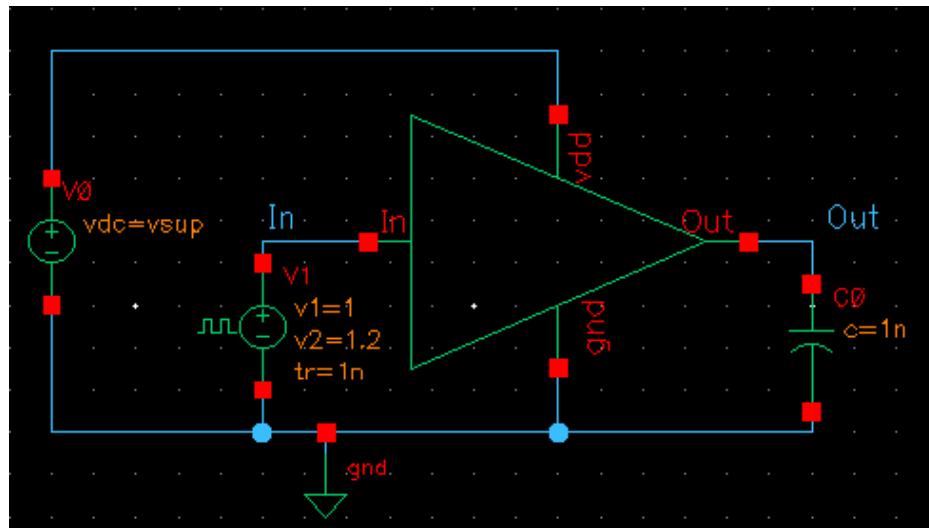


Figure 73: Amplifier circuit with capacitor at the output

12.2 Configuration of the AC analysis

Now call up ADE Explorer and set up an AC simulation via the *Choosing Analysis* window. Select *ac* as the simulation type with *Frequency* as *Sweep Variable* and enter the value 1 for *Start* and the value $100G$ for *Stop* to perform a simulation in a frequency range from 1 Hz to 100 GHz . Confirm the setup by clicking *OK*. Since the operating point parameters will also be used next, a simple DC simulation must also be configured as described in Chapter 6.3.

Here, also, you must select *Outputs* for the AC analysis. You should have already created these during the setup of the transient analysis. In the *Outputs* area of the ADE Explorer window, only keep the input (*In*) and the output (*Out*) and deactivate all other *Outputs* that you have previously created.

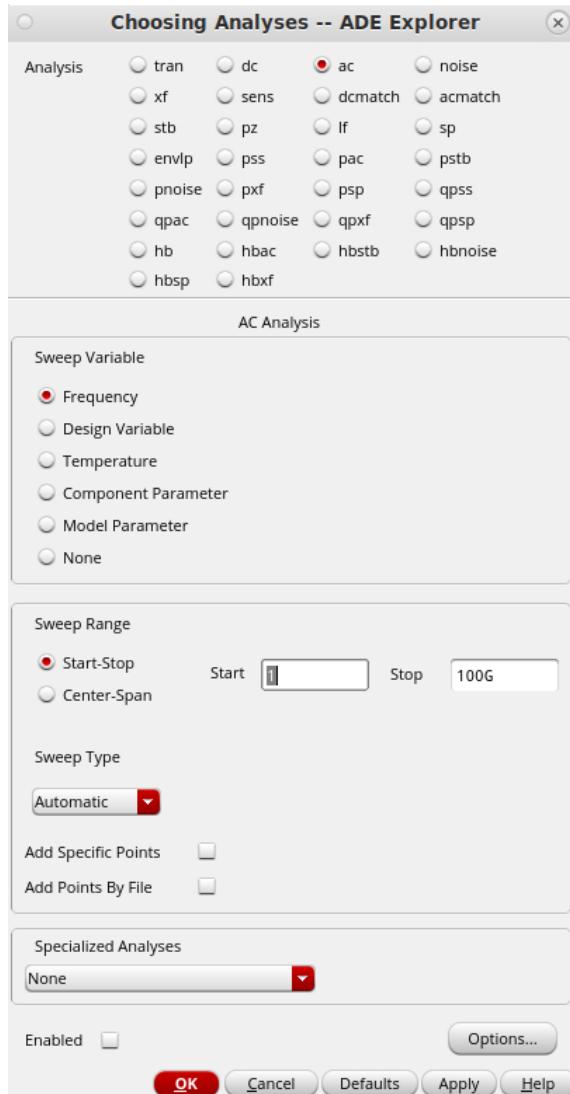


Figure 74: Setting up an AC analysis

You also want to check whether the expected value is reached in the AC simulation. To do this, create a *Calculator* expression that determines the gain from the small signal properties. The following expression applies to the inverting single-ended amplifier in the implemented version:

$$A = \frac{gm_n}{gm_p} \quad (3)$$

The values gm_n and gm_p correspond to the transconductance of the NMOS or PMOS transistor. To be able to extract these variables, the DC simulation must have been carried out in advance. Transfer this formula to the *Calculator* by selecting the transistor slope (gm), as described in *Chapter 6.7*, using the *op* operator.

Once you have completed the formula, send the complete expression to the ADE Explorer window. To do this, first define a new *Output* with the name *gain* in the *Setting Outputs* window and then click on *Get Expression* to get the expression from the *Calculator*.

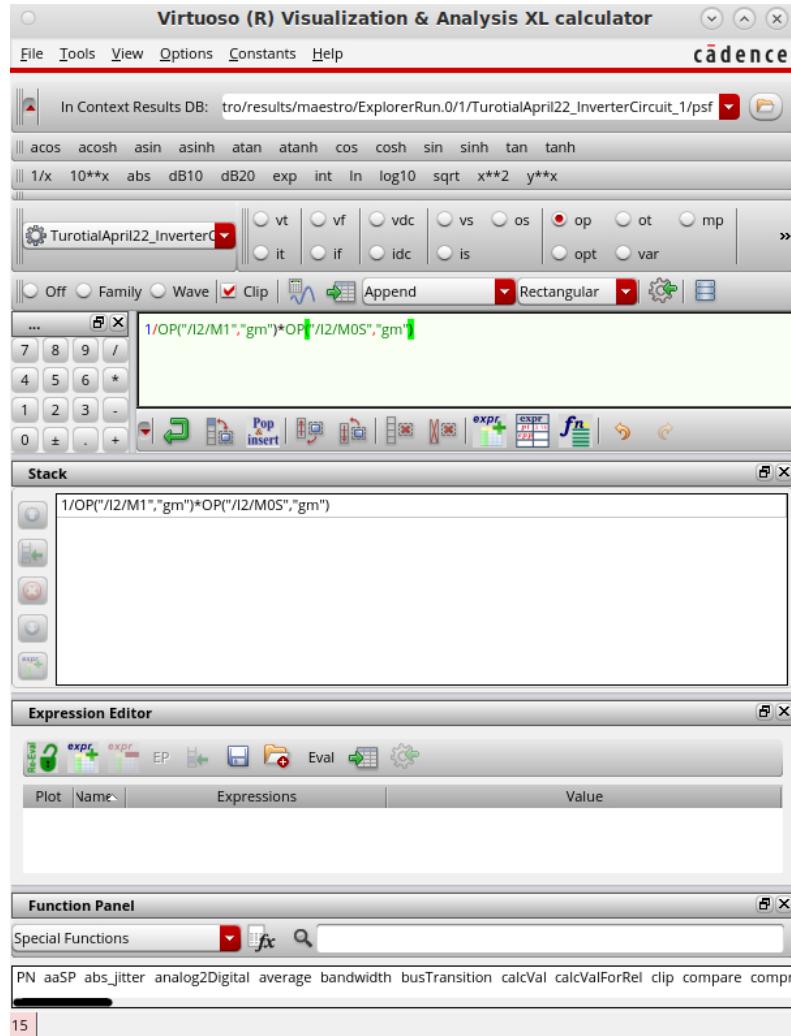


Figure 75: The *Calculator* window with formula (3) for calculating the gain

Then start a simulation and display the input and output voltage graphically.

It can be seen that the output voltage for a certain frequency range is higher than the input voltage by a factor of 1.08. This factor is the gain of the circuit. If you compare the gain value calculated from the transistor slopes with the value in the ADE Explorer window, you will see that the value is identical. It becomes also clear that the gain decreases with a certain frequency. The frequency range in which the gain of the circuit remains constant is called bandwidth.

Next, you will determine the bandwidth of the amplifier in various ways on the basis of the simulation results. To do this, you must first create a Bode diagram that shows the amplitude and phase response of the circuit.

To proceed, click on *Results* in the ADE Explorer menu bar and then on *Direct Plot*. In the next submenu, select *AC Magnitude & Phase* and click on *Schematic* to select the output of the circuit. Confirm the selection with **[ESC]**.

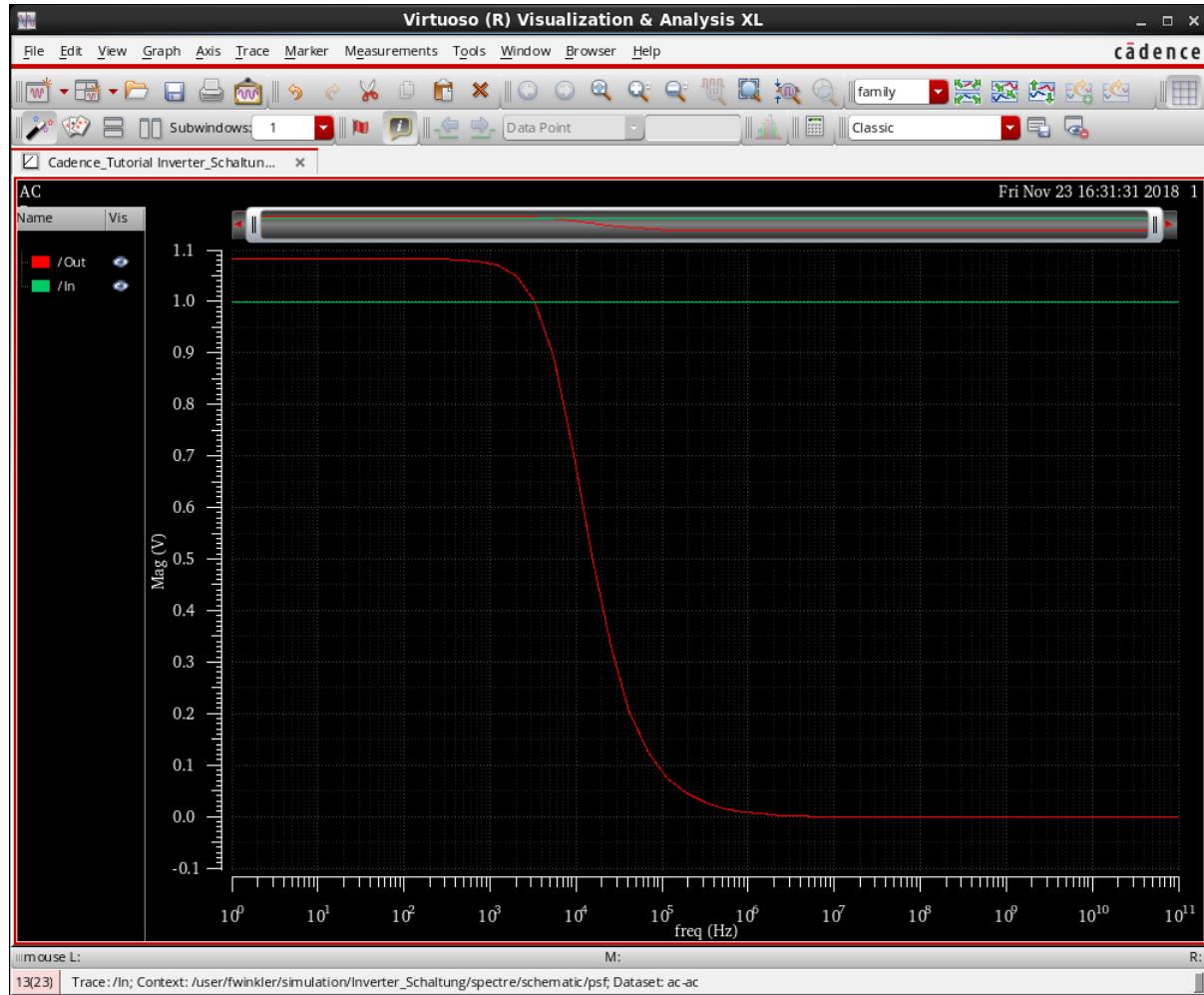


Figure 76: Plot of AC analysis with input and output voltage

A window now opens which displays the Bode diagram, consisting of amplitude (*magnitude*) and phase response (*phase*). For a better overview, separate the amplitude and phase response into separate strips using the function *Split all Strips*.

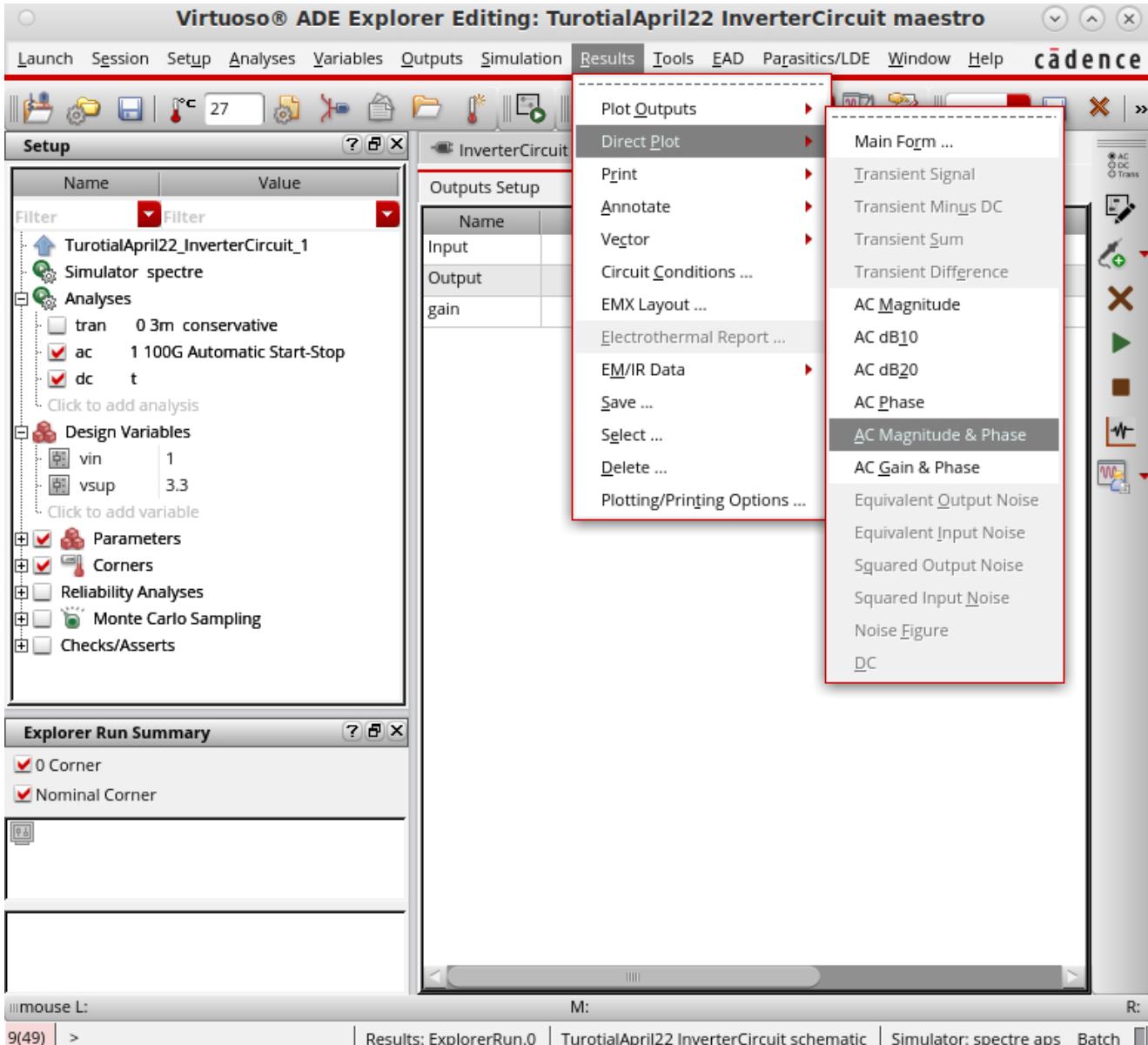


Figure 77: Calling up the display of *Magnitude and Phase*

In the *magnitude* graph, set a marker at 10^0Hz and another at the position where the amplitude characteristic has dropped by 3 dB . Alternatively, find the frequency at which the phase rotation has changed from 180° by 45° to 135° . The frequency at this point corresponds to the bandwidth of the amplifier.

The bandwidth should now be extracted from the simulation data using a *Calculator* function. To do this, start the *Calculator* again and search for the function *bandwidth* in the *Function Panel*. You can now specify various parameters for the function (see *Figure 79*). First select the signal of the output with the *VF* operator. This operator is used to select a signal from the *Schematic*, which generates an *Expression* for an AC voltage. In the *Db* field, the value is entered by which gain the Bode plot should drop to determine the bandwidth. Enter the value 3 dB in this field and press *OK*.

Send the finished expression directly from the *Calculator* to the *Output* area of ADE Explorer. The graphically determined bandwidth value of about 8.086 kHz should be almost identical to the calculated value.

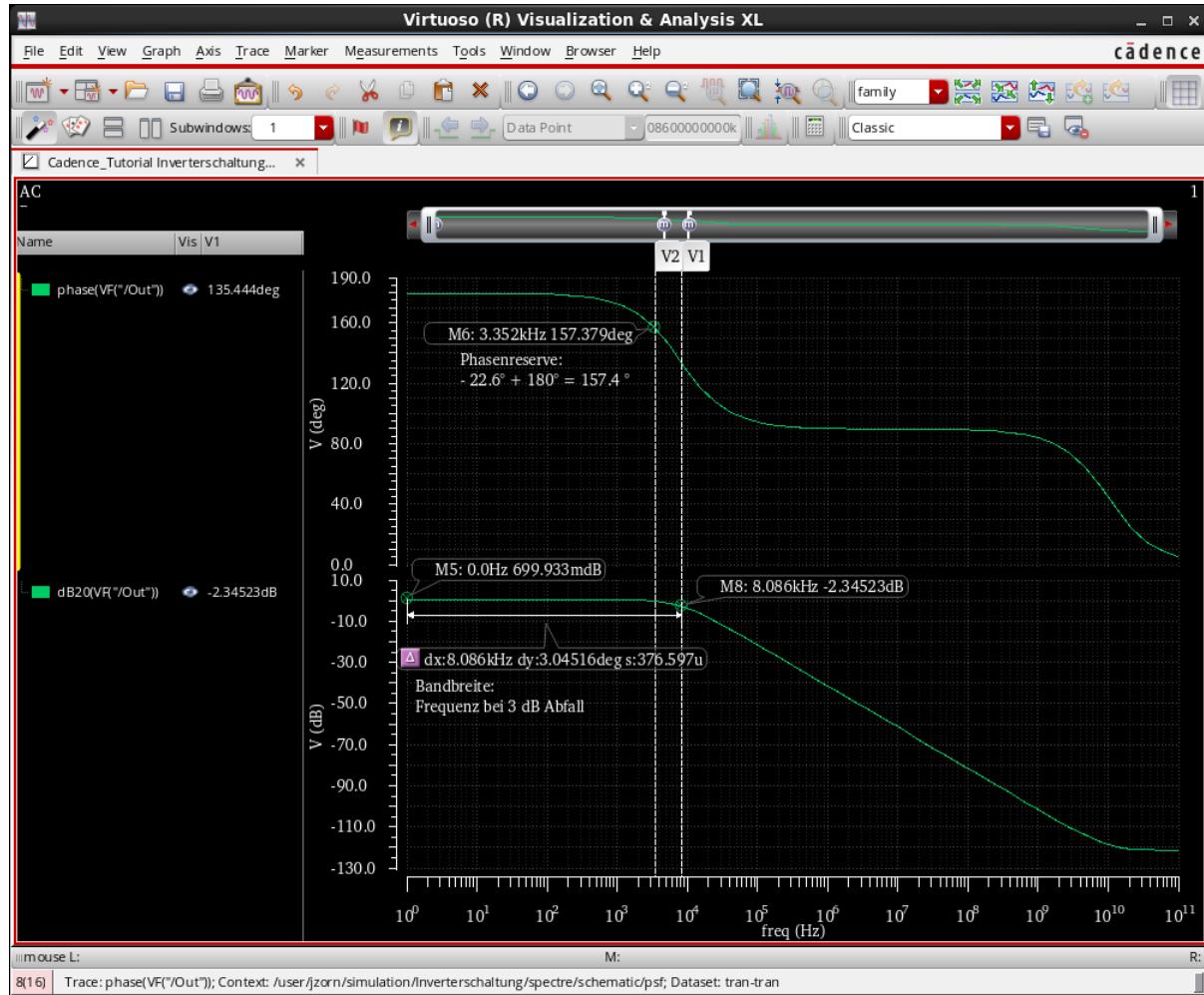


Figure 78: Bode diagram, consisting of frequency and amplitude response

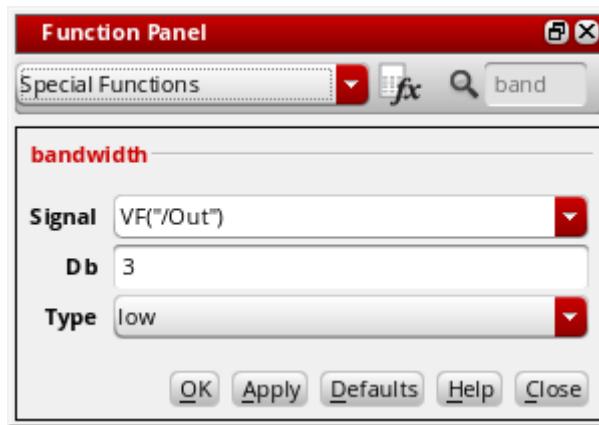


Figure 79: Parameters of the bandwidth function in the *Calculator*

The phase margin of the circuit should also be determined in the AC simulation. The phase margin is related to the phase rotation that occurs at an amplification of 0 dB. It corresponds to the difference between the actual phase rotation of the system and a phase rotation of 180° . The phase margin is a parameter that can be used to evaluate the stability of a control loop. Set another marker at the 0 dB line in the Bode diagram (compare

Figure 78). Then read the value in the phase margin and subtract it from 180° to determine the phase margin.

The phase margin should also be calculated using the *Calculator*. To do this, proceed in exactly the same way as for calculating the bandwidth and search for the function *Phase Margin* in the *Function Panel*. Then transfer the expression to the *Outputs* of ADE Explorer, simulate again and compare the results.

Outputs					
Name/Signal/Expr	Value	Plot	Save	Save Options	
1 Out		<input checked="" type="checkbox"/>	<input type="checkbox"/>	allv	
2 In		<input checked="" type="checkbox"/>	<input type="checkbox"/>	allv	
3 gain	1.087	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
4 phaseMargin(VF("/Out"))	-22.61	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
5 bandwidth(VF("/Out")) 3 "I...	8.086K	<input checked="" type="checkbox"/>	<input type="checkbox"/>		

Plot after simulation: Auto Plotting mode: Replace

Figure 80: The *Output-Area* of the ADE Explorer analysis window with all the expressions used

13 DC analysis in ADE Assembler

13.1 Starting ADE Assembler

The following section introduces the ADE Assembler tool. This simulation tool extends the familiar ADE Explorer environment with more advanced analysis options. A key feature of the ADE Assembler is the configuration of multiple simulations. These are referred to *Tests* and enable the simultaneous execution of different analyses (as known from ADE Explorer) with different variables. Different designs can also be simulated within an ADE Assembler environment. In addition, it is possible to create global variables that are valid in all *Tests* created and can be varied in a predefined range if required. All simulation results are displayed centrally in the main window of ADE Assembler. The following chapters will also deal with simulation types that are only possible in ADE Assembler. These are in particular the Corner and Monte Carlo simulations.

At the beginning a new *View* must be created for ADE Assembler. To do this, the tool, analogous to ADE Explorer, is started by the menu bar by clicking on *Launch > ADE Assembler*.

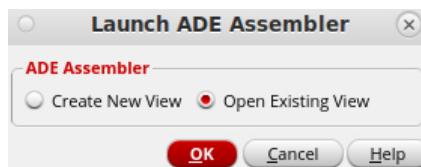


Figure 81: Selection of a new or existing ADE Assembler view

When you launch ADE Assembler for the first time, click on *Create New View* and in the next window on *OK*. If you have already worked with ADE Assembler and saved a *View*, you can open this *View* by selecting the setting *Open Existing View*.

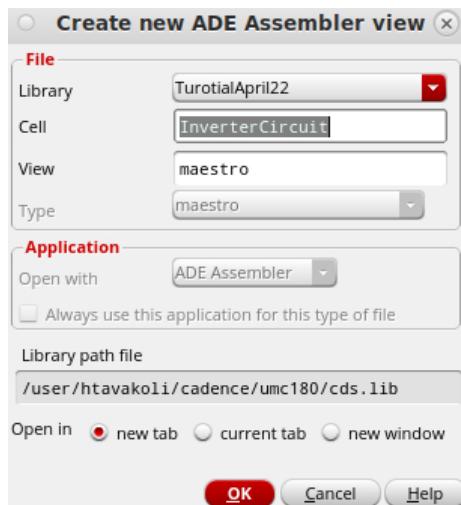


Figure 82: Configuring a new ADE Assembler View

The main window of ADE Assembler appears.

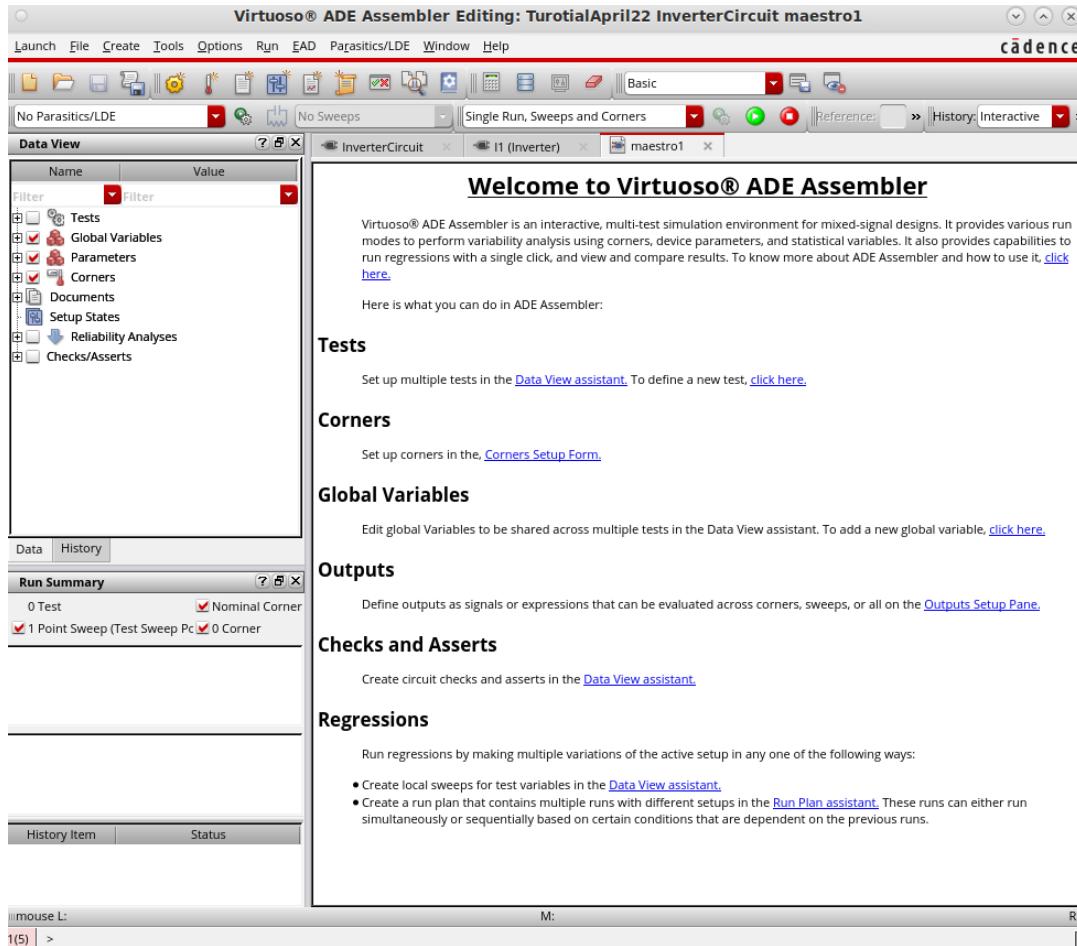


Figure 83: The main window of ADE Assembler

13.2 Optimization of the simulation speed

The simulation speed can be significantly accelerated by running several simulations in parallel. However, the optimised number of simulations running in parallel depends on the number of available processor cores installed in the available simulation computer. To determine the number of Cores available in your computer, click on *Terminal* in the Linux menu bar and write the syntax *lscpu*. It contains information regarding the installed operating system and the available hardware. In particular, the number of processor cores can be read here.

To configure the parallel execution of simulations in ADE Assembler, select the menu item *Run Options* in the ADE Assembler main window under *Options* and select the option *parallel* in the *Run in* area.

```
[htavakoli@shima ~]$ lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Address sizes:         46 bits physical, 48 bits virtual
Byte Order:            Little Endian
CPU(s):                48
On-line CPU(s) list:  0-47
Vendor ID:             GenuineIntel
Model name:            Intel(R) Xeon(R) Gold 6226 CPU @ 2.70GHz
CPU family:            6
Model:                 85
Thread(s) per core:   2
Core(s) per socket:   12
Socket(s):             2
Stepping:              7
CPU(s) scaling MHz:  98%
CPU max MHz:          3700.0000
CPU min MHz:          1200.0000
BogoMIPS:              5400.00
Flags:                 fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge m
ca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 s
s ht tm pbe syscall nx pdpe1gb rdtscp lm constant_tsc
art arch_perfmon pebs bts rep_good nopl xtopology nons
top_tsc cpuid aperfmpf perf pn1 pclmulqdq dtes64 monitor
ds cpl vmx smx est tm2 ssse3 sdbg fma cx16 xptr pdcm p
ci1 dca sse4_1 sse4_2 x2apic movbe popcnt tsc deadline
timer aes xsave avx f16c rdrand lahf lm abm 3dnowpref
etchn cpuid fault epb cat l3 cdpr_l3 intel_ppin ssbd mba
ibrs ibpb stibp ibrs enhanced_tpr shadow flexpriority
ept vpid ept ad fsgsbase tsc_adjust bm11 avx2 smep bm
i2 erms invpcid cqmq mpx rdt a avx512f avx512dq rdseed
```

Figure 84: Display of the system values in the system monitoring

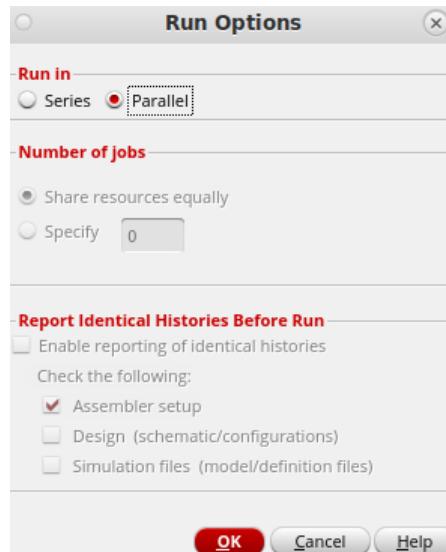


Figure 85: Setting the Run Options

Select the number of simulations running in parallel in the *Options* menu under the *Job Setup* menu item. In the *Max. Jobs* area, enter the number of CPU Cores that you have previously determined. In the *Run Summary* area of the ADE Assembler main window, you can check how many jobs are running in parallel during the simulation execution. Each running simulation corresponds to a small square. The number of simultaneously displayed squares corresponds to the number of simultaneously running simulations.

Note: These settings are not saved in Cadence Virtuoso and must be reset each time the Virtuoso is started.

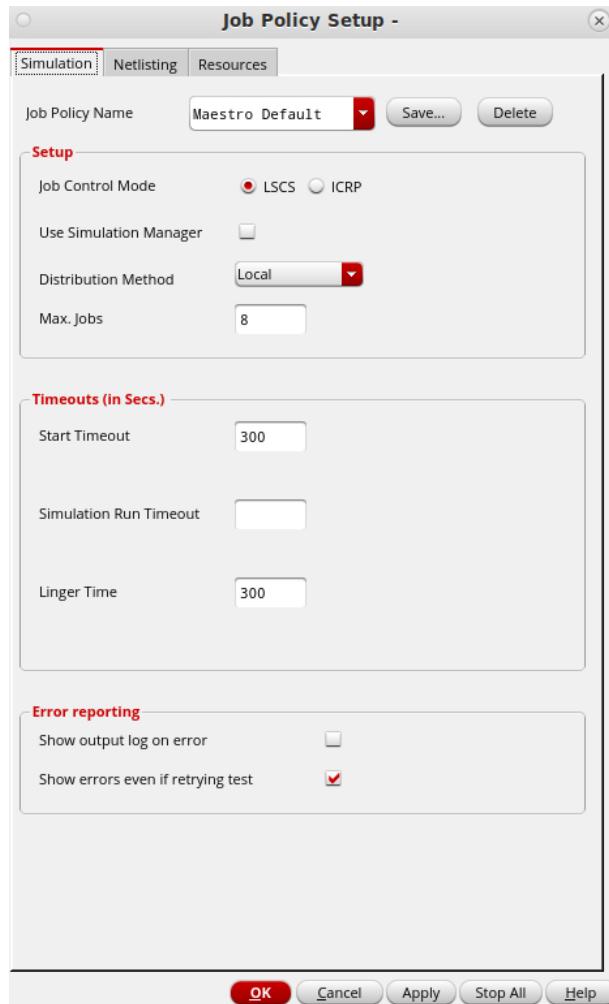


Figure 86: Enter the number of CPU-Cores

Furthermore, the settings are applied only to the currently selected processed ADE Assembler session.

13.3 Conversion of the circuit to an inverter

For a meaningful DC analysis, the inverting amplifier must be converted into an inverter. To do this, the gate-drain connection on the PMOS transistor is removed. The two gates of the transistors are short-circuited and connected to the input pin (*In*).

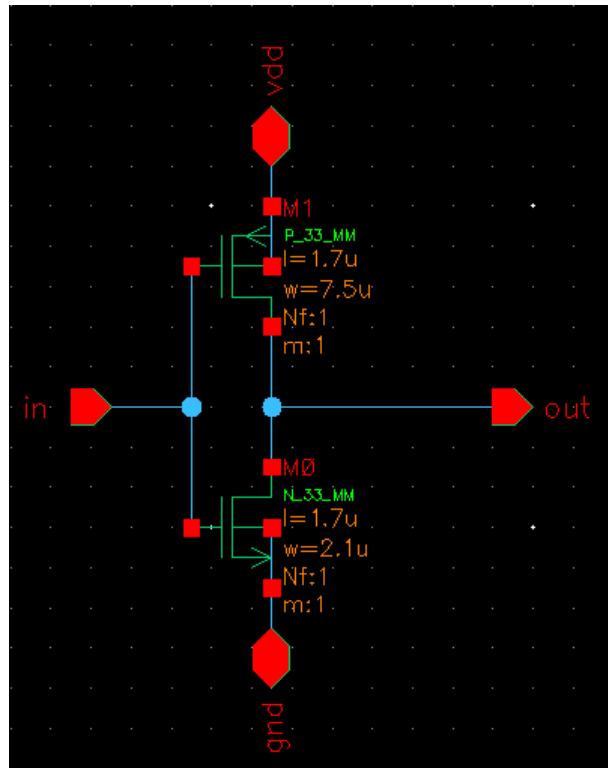


Figure 87: Inverter circuit in the *Schematic*

The transistor geometries remain unchanged.

13.4 Configuring a DC analysis in ADE Assembler

To begin, you must create a new test by clicking on the plus symbol next to *Tests* and selecting *Click to Add Test*. The ADE Assembler test editor window opens, which is very similar to the familiar ADE Explorer window. You will be asked to select the design for which you want to define a test. The cell preselected in this window always corresponds to the cell from which ADE Assembler was started. As you have started ADE Assembler from the *Schematic* of the inverter circuit, you only need to confirm with *OK*.

Tests configured in ADE Assembler are comparable with several independent ADE Explorer views. You have the option of using your existing *State* from ADE Explorer and deactivating unnecessary analyses. Alternatively, you can configure a new DC simulation directly within a test in ADE Assembler.

To configure a new DC simulation, proceed as follows:

1. If you have not yet created a test, create it as described in the previous paragraph.
2. Import all variables from the *Schematic* by right-clicking in the *Design Variables* area and selecting *Copy from Cellview*. The variables *vin* and *vsup* should then be visible. Set *vin* = 1 V and *vsup* = 3.3 V.

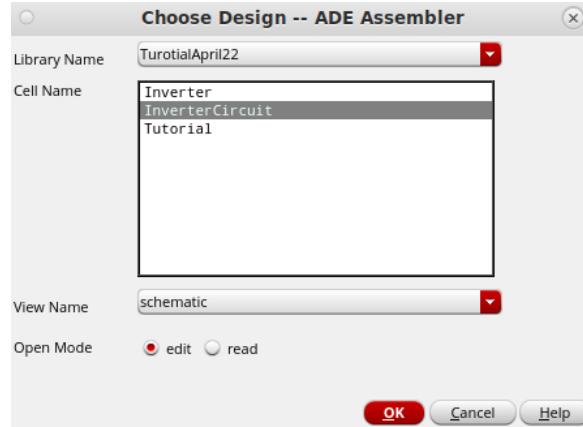


Figure 88: Configuration of a new test

3. As explained in *Chapter 7.1*, configure a DC simulation with *Sweep* by clicking on *Click to add analysis* in the *Analyses* area. Select *vin* as the *Sweep* variable. Also select *Save DC-Operating Point* and confirm with *OK*.
4. Finally, configure *Outputs* for the DC simulation. To do this, proceed as in *Chapter 6.5*. Record the output voltage *vout* as *Output*.

ADE Assembler differentiates between local and global variables. Local variables are only valid locally for a single test. Global variables are valid for all tests and also overwrite the values of local variables with the same name. Local variables from the test are automatically also available as global variables. However, the value can vary.

Now click on the plus symbol next to *Global Variables* and then on *Click to add variable*. Give the global variable *vsup* a value of 3.3 V. Also, remember that the variable names refer to the names in your *Schematic*!



Figure 89: Create a new global variable

As with ADE Explorer, ADE Assembler offers the option of using the *Calculator* tool. To start this tool, click on *Tools* in the main window of ADE Assembler and then on *Calculator*.

You should now formulate a *Calculator* expression that determines the value of *vin* at which an output voltage *vout* is reached that corresponds to half the supply voltage, i.e. 1.65 V. Use the *cross* function, which can be found in the *Function-Panel* of the *Calculator*. This function determines when a signal reaches a specified comparison or threshold value (*Threshold Value*).

Select the output signal of the inverter using the *Calculator* operator *vs*, as the simulation of the output voltage is a *Sweep*. Select half the supply voltage as the threshold value. To do this, you can use the *op* function to select the voltage of the supply voltage source. Click on this voltage source and select *v* from the list. This value must also be halved, which can be done using the notation *vdc/2* in the *Calculator*. The advantage of this

procedure is that any change in the supply voltage is transferred directly to the *Calculator* expression without further adjustment. Enter 1 as *Edge Number*. For *Edge Type*, select *falling*, as the simulation starts with an input voltage v_{in} of 0 V and the full supply voltage v_{dc} of 3.3 V is therefore applied to the output at the beginning. The threshold value is therefore determined on a falling edge. For *Number of occurrences*, click on *single*, as the threshold value is only reached once in this simulation.

Compare your expression with the expression from *Figure 90* on the next page. Then select the expression and copy it with [CTRL]+[C].

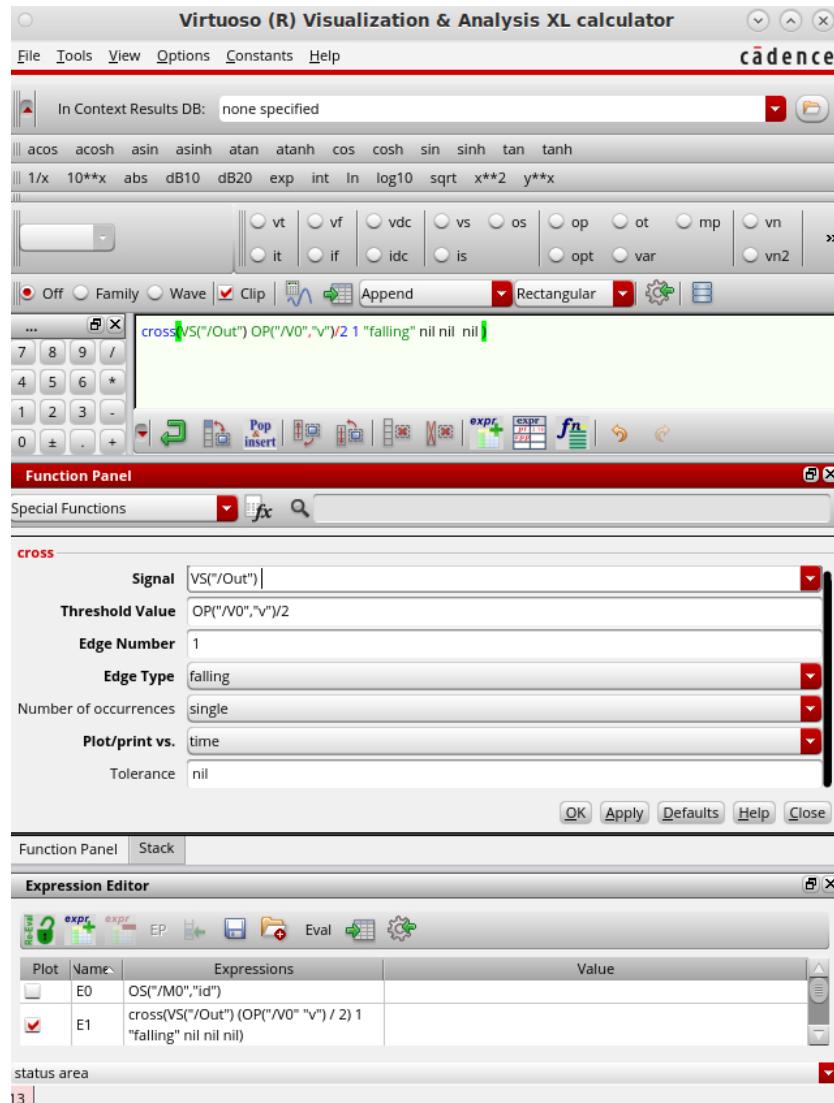


Figure 90: The complete cross-Expression in the Calculator

You can create a new *Output* directly in the main window of ADE Assembler and assign it to a test. To do this, click on the red arrow of the *Add new output* button in the *Outputs Setup* tab in the menu bar. A submenu appears in which you can specify the type of newly created *Outputs* (Figure 91).

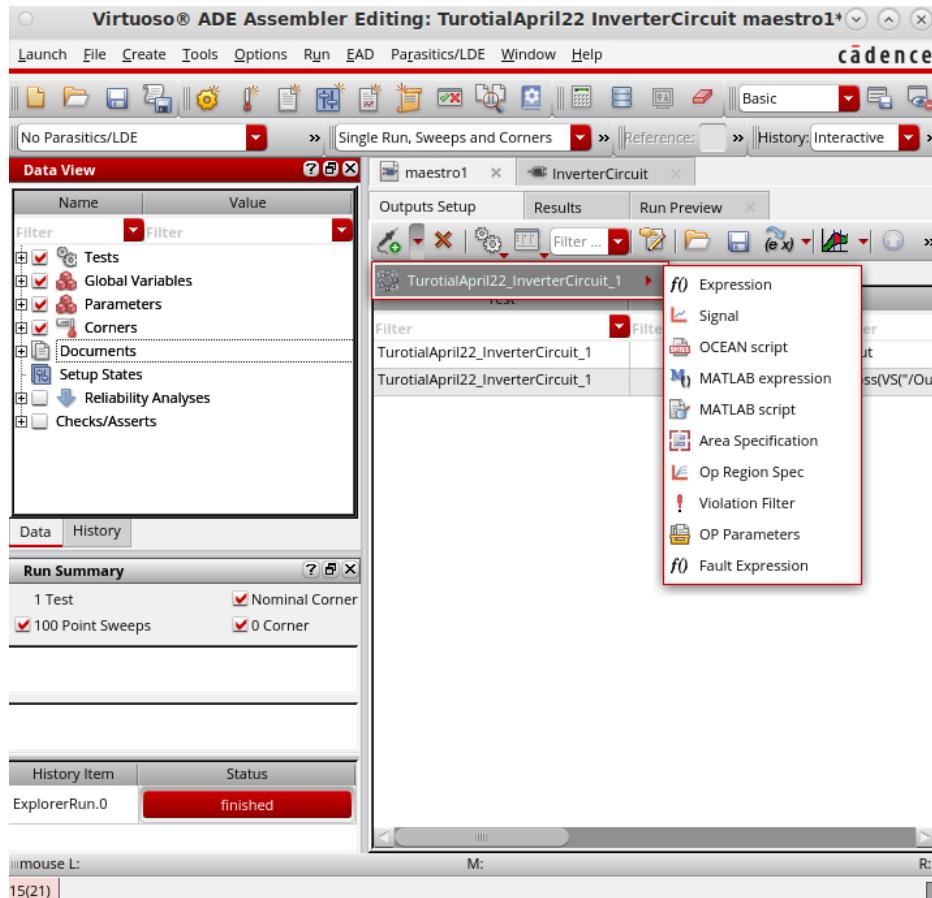


Figure 91: Configuration of *Outputs* in ADE Assembler

You should configure the *Calculator* expression created and copied in the previous paragraph as *Output*. Therefore, select *Expression* as the *Output* type. You will now see the newly created *Output* in the ADE Assembler main window (Figure 186). You can name the *Output* by double-clicking on *Name*. In the *Type* column, you can see the type of *Outputs*. Double-click in the *Details* cell. A text field and a button with three dots opens.

Test	Name	Type	Details	EvalType	Plot
TurotialApril22_InverterCircuit_1		signal	/Out	point	<input checked="" type="checkbox"/>
TurotialApril22_InverterCircuit_1		expr	cross(VS("Out") (OP("V0" "v") / ...	point	<input checked="" type="checkbox"/>

cross(VS("Out") (OP("V0" "v") / 2) 1 "falling" nil nil nil)

Figure 92: Inserting an expression as *Output* in ADE Assembler

Paste your already copied *Calculator* expression into the text field by clicking on it and pressing **[CTRL]+[V]**. You can also set up signals from the *Schematic* as *Output* in this way. To do this, proceed as described above and select *Signal* instead of *Expression* as the *Output* type. In this case, the button with the three dots opens the *Schematic*, where you can select signals by clicking on them in the same way as in *Chapter 6.6*.

Now perform the DC analysis and check the value determined by the *Calculator* expression.

maestro1		InverterCircuit				
Outputs Setup		Run Preview				
Name	Type	Details	Value	Plot	Save	Spec
	signal	/Out			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	expr	<code>cross(VS("/Out") (OP("V0" "v") / 2) 1 "falli..."</code>	1.557	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Figure 93: The calculated value of the cross-Function

Click on *Plot All* in the menu bar or on the graph symbol in the *Nominal* column to display the plot. In the diagram that opens automatically, you can now compare the calculated result with the graphical solution. To do this, place a marker in the diagram by pressing the [M] button. Double-click on the marker to open a settings window in which you first select *by XMode* under *Position* and then enter the value determined by *Calculator* in the field next to it. Confirm with *OK*. Check whether the value you read on the Y-axis corresponds to half the supply voltage (take into account possible rounding errors).

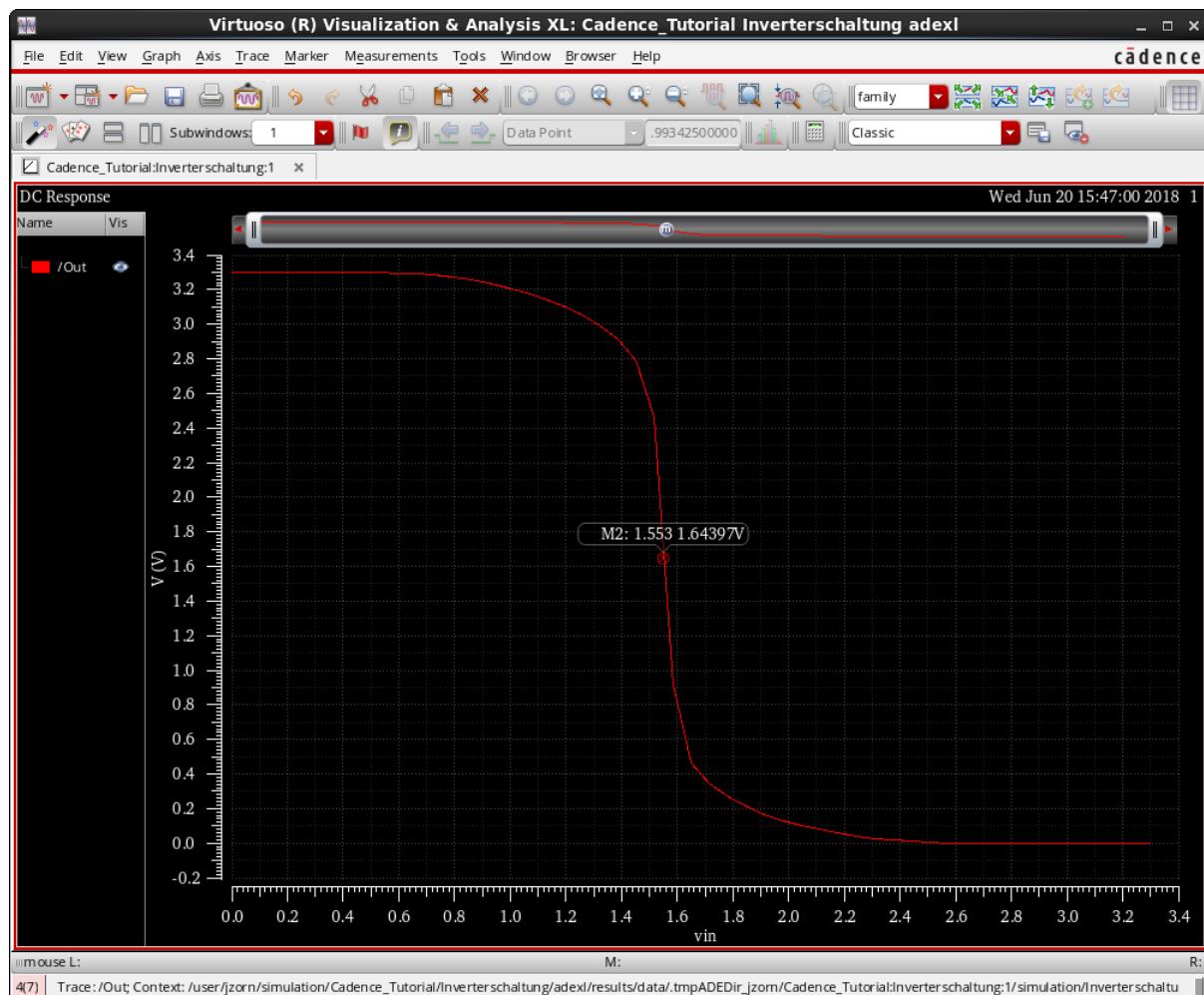


Figure 94: Diagram with marker at the value resulting from the cross-Function results

13.5 Sweeps via global variables in ADE Assembler

In ADE Assembler, you have two options for setting up *Sweeps*. You can define these using the test editor, which is similar to the procedure for ADE Explorer. Alternatively, you can also create a global variable and define a value range for this variable in ADE Assembler. A significant difference is that with a *Sweep* with a global variable, a working point is calculated for each variable value. With *Sweeps* configured within a test (e.g. as an option DC analysis) the operating point calculation only takes place for the starting point, while intermediate values are not saved.

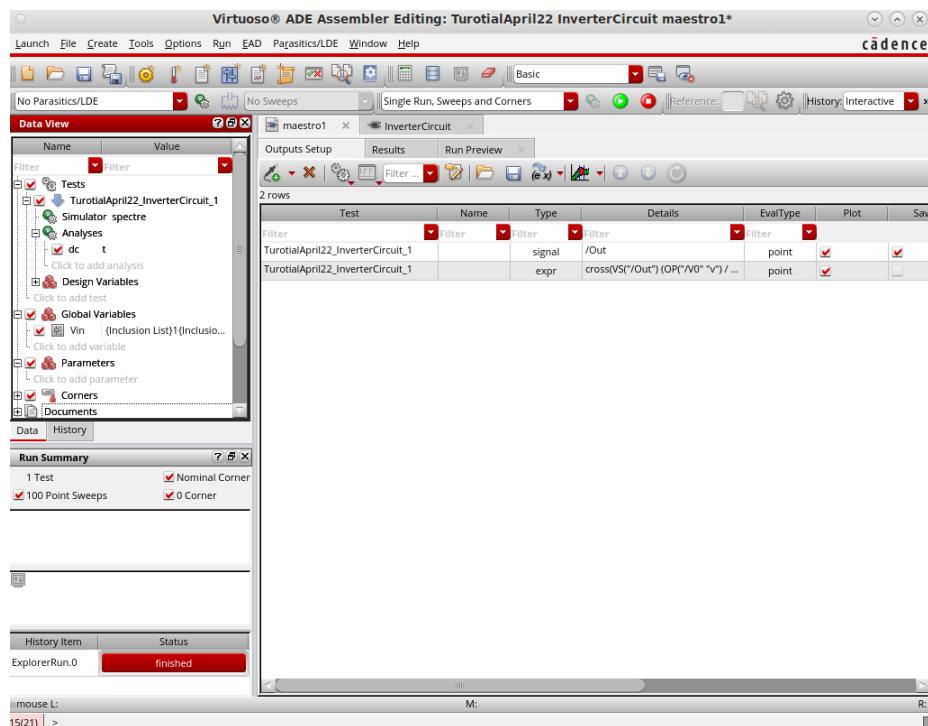


Figure 95: Configuring a test in ADE Assembler

First deactivate the *Sweep* within the test by clicking on the plus symbol below the test. Then open another submenu under *Analyses* and double-click on the DC simulation to open the *Choosing Analysis* window. Uncheck *Design Variable* in the *Sweep Variable* area. To be on the safe side, check at this point whether *Save DC Operating Point* is checked and confirm with *OK*.

If it does not already exist, create the global variable *vin* to define the *sweep* outside the test. Click on *click to add variable* in the *Global Variables* area of the ADE Assembler main window and create the variable there. Then double-click on the value of the variable and open the *Parameterize* window (Figure 190) by clicking on the button with three dots.

Here, you first delete the default setting by clicking on the *Delete Spec* button and then create a new *From/To* parameterization. Start at 0 V and go to 3.3 V in 100 steps. The more *Steps* you choose, the better accuracy the representation of the characteristic curve acquire. However, this also increases the simulation run time. Leave the *Step Type* at *Auto*. Then confirm with *OK* and return to the ADE Assembler main window, where the newly created global variable and its value range are now displayed.

Check that *vin* is crossed out in the area under *Design Variables*. As the global variable, *vin* now applies to all configured tests and overwrites a value previously defined within a test.

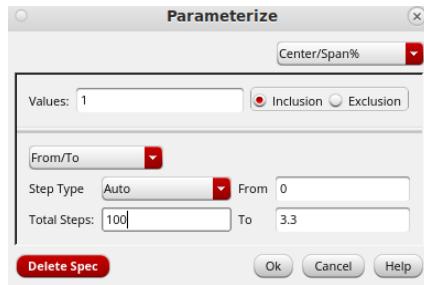


Figure 96: Sweep configuration of global variable *vin* with 100 steps

With a Sweep via a global variable, an operating point is calculated for each Sweep-Step and is also saved using the *Save DC Operating Points* setting. Therefore, you need to make a change to the configuration of the Outputs in order to be able to display the simulation results graphically. To do this, configure a new Output for the test of the DC simulation in ADE Assembler of type *Expression*.

To proceed open the *Calculator*. Then create an expression for a DC operating point by clicking *vdc* from the operators introduced in Chapter 6.7. The Schematic of the inverter circuit opens. Select the line of the output signal *vout* by clicking. Also disable the cross expression in the Outputs by removing the arrow in the *Plot* button. This is necessary because this expression was configured for the evaluation of a sweep configured within the test and a sweep over the global variable *vin* would lead to evaluation errors.

Test	Name	Type	Details	EvalType	Plot	Save
Filter	Filter	Filter	Filter	Filter	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
TurorialApril22_InverterCircuit_1		signal	/Out	point	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
TurorialApril22_InverterCircuit_1		expr	VDC(""/Out")	point	<input checked="" type="checkbox"/>	<input type="checkbox"/>
TurorialApril22_InverterCircuit_1		expr	cross(VS(""/Out") (OP(""\V0" "v") / ...	point	<input type="checkbox"/>	<input type="checkbox"/>

Figure 97: Disabling the cross expression for Sweep via global variables

Now start the simulation by clicking on the green arrow in the main window of ADE Assembler. After the simulation is completed, the results of the simulation run are listed under the *Results* tab. In this example, 100 sweep points should be listed. For better clarity, click on the arrow of the *Details* selection menu in the toolbar of the *Results* window and select *Detail - Transpose*.

This presentation is clearer, especially for simulations with many results. Plot the values by clicking the *Plot-all* button in the menu bar of the *Result* window. The resulting curve is the same course of the DC sweep shown in Figure 94. The difference to the previous simulation is that for each Sweep Step of operating point has been calculated, which can be displayed in the *Schematic*.

To do this, right-click on a simulation result in the *Results* tab and select *DC Operating Points* in the *Annotate* menu.

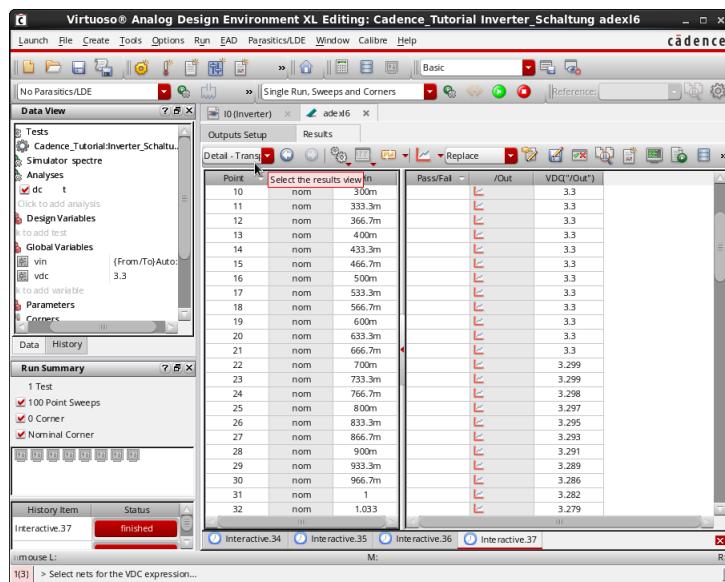


Figure 98: Transposed view of the Results window

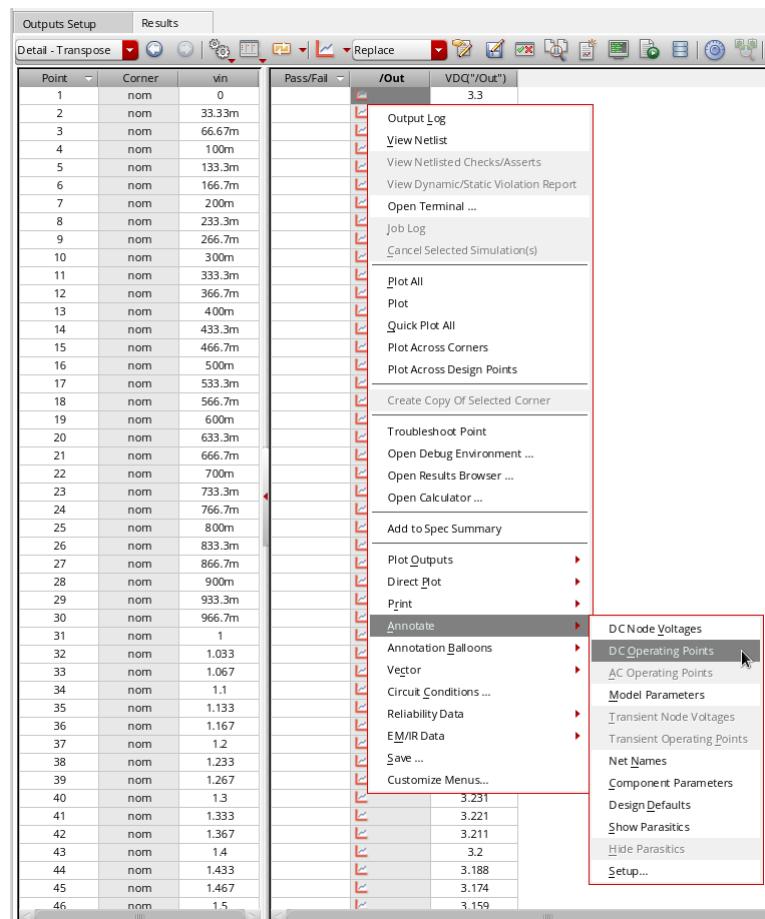


Figure 99: Display of specific Operating Points in Schematic

14 The corner simulation

14.1 Setting up a corner simulation

Corner simulations is used to examine the influence of process variations and temperature changes on the behavior of a circuit. In addition, variables can be set to special key values in *Corner* definitions, for example to introduce the influence of fluctuations in the supply voltages into the simulation. Expand the *Corners* list in the *Data View* area of the ADE Assembler window by clicking on the plus button next to it. Clicking on *Click to Add Corners* opens the *Corners Setup* window (Figure 100). By default there is already the column *Nominal*, which corresponds to the nominal simulation conditions. Each column in this window represents its own *Corner* definition, which can be named in the column header.

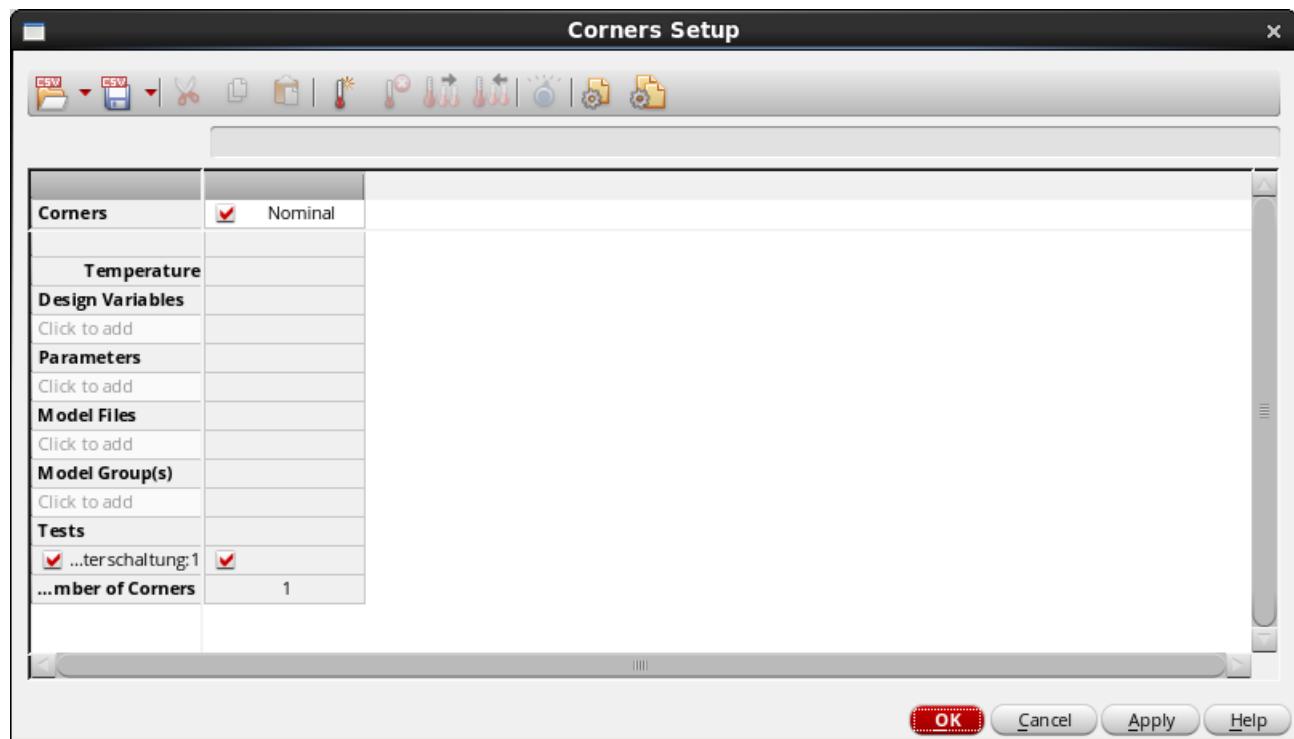


Figure 100: Window for creating Corner simulations

Before the setup can be carried out, the *Model Files* of the components used in the circuit must be loaded into ADE Assembler. Click on *Model Files > Click to Add* in the *Corners Setup* window, whereupon the *Add/Edit Model Files* window opens (Figure 101). Click on *Import from Tests* to load the simulation model files available in previously defined tests.

With the CMOS technology used, the simulation models are distributed across many different files. However, only transistors with a supply voltage of up to 3.3 V are used in the current circuit diagram. The simulation models for these transistors can be found in the file „mm180_reg33_v114.lib.scs“. All other *Model Files* can be marked and removed again for better clarity. However, if there are other components in the circuit diagram, such as resistors, capacitors, inductors, etc., additional simulation files must be included. After another click on *OK*, the selected model file is loaded into the *Corners Setup* window.

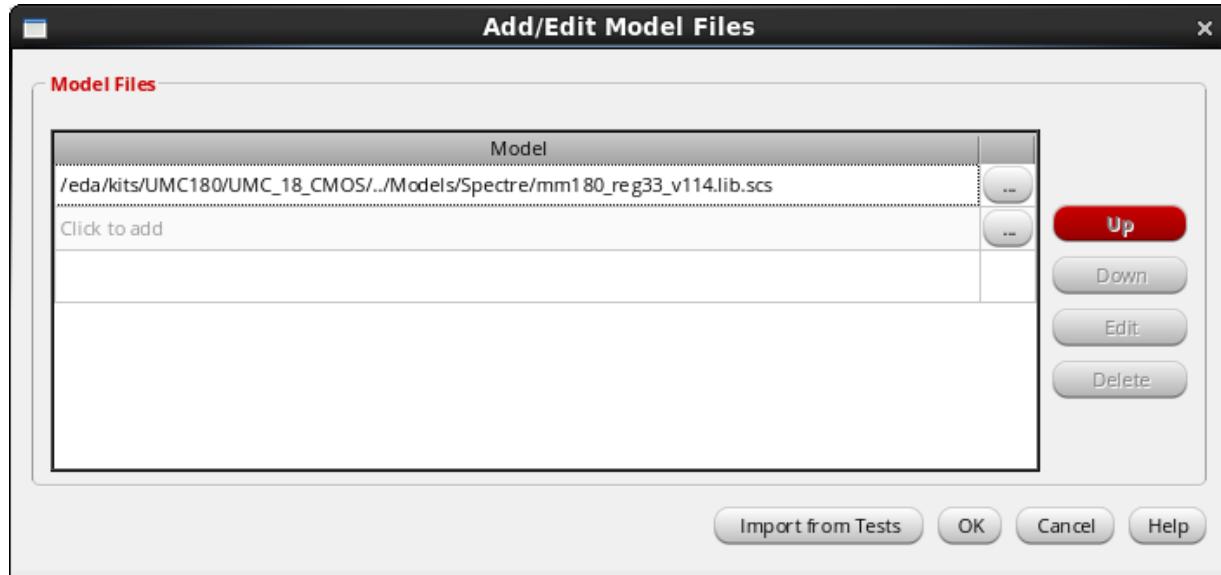


Figure 101: Inserting the model files of the Corner-Simulation

Now create a new *Corner* definition by clicking on the *Add new corner* button in the toolbar of the *Corners Setup* window. This creates another column in which you can enter *Process-Corners*, temperatures and variable values. In this simulation, the influence of the ambient temperature on the circuit is to be checked. To do this, enter the values -40 27 120 in the (*Temperature*) field.

In the UMC 180nm technology, a total of five *Process Corners* have been defined by the manufacturer, which take into account variations in technology parameters and their effect on the transistor properties. A distinction is typically made between the *typical* (normal), *fast* (fast) or *slow* (slow) corner. These designations are abbreviated with the initial letter and summarised in a combination of two or four letters. The first letter denotes the process *Corner* for the NMOS transistor, while the second letter defines the *Corner* of the PMOS transistor. The following table 3 lists all available *Corners* of the process.

NMOS	PMOS	Corner
Typical	Typical	tt
Fast	Fast	ff
Slow	Slow	ss
Fast	Slow	fnsP
Slow	Fast	snfp

Table 3: Process Corners in UMC180 nm

Now enter all five *Process Corners* mentioned above in the *Model Files* field and activate the field by ticking the adjacent box. The *Number of Corners* field should then display 15 (see Figure 100). Confirm the settings by clicking on *OK*.

The *Corner* settings now appear selected in the *Data View* area in ADE Assembler. Run the simulation with the settings from Chapter 13.4. Click on the *Run Simulation* button to run the simulation. The following diagram is displayed.

Each graph in the set of curves corresponds to a corner. In particular, the influence of the three different temperatures can be seen as the different curves cluster in three areas.

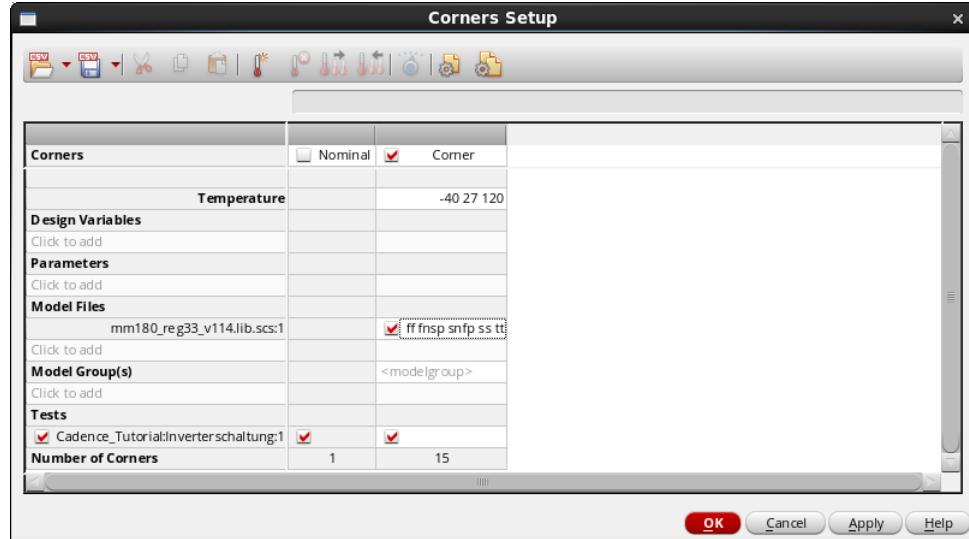


Figure 102: The complete setup of a Corner-Simulation

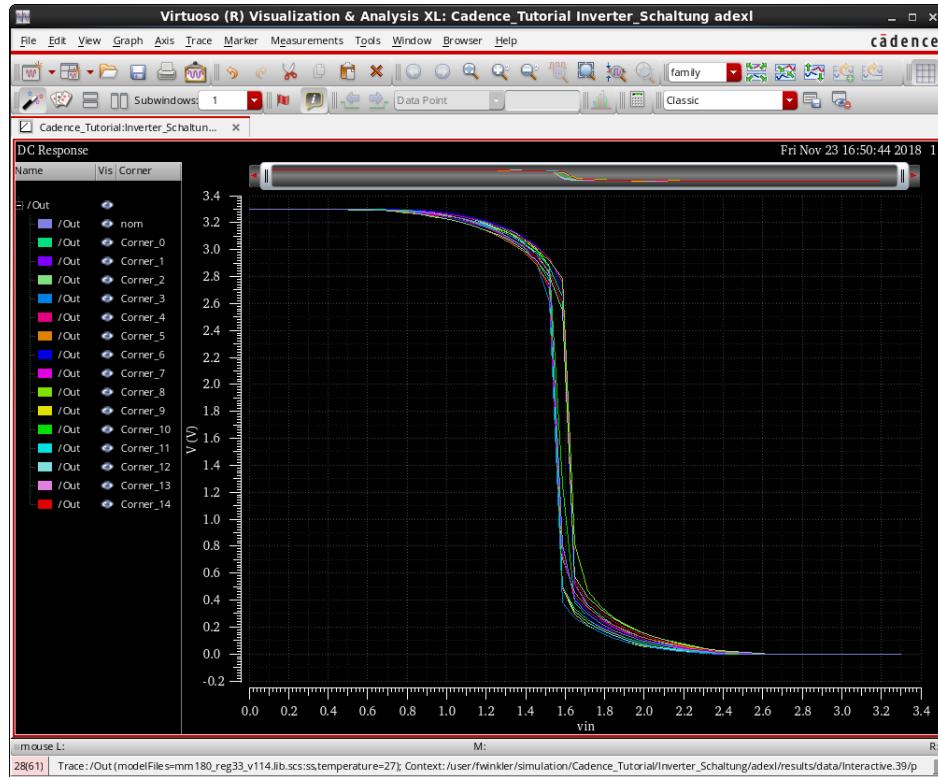


Figure 103: Diagram of the DC analysis including the temperature-Corners

14.2 Specs in a corner simulation

ADE Assembler offers the option of introducing specifications for defined *Outputs* and *Expressions*, with which the results of an analysis can be checked for selected criteria (larger, smaller, interval, minimum, maximum) and highlighted in colour. This is illustrated using the simulation parameters from Chapter 13. Activate the cross function in the *Outputs Setup* tab in ADE Assembler.

In the *Outputs Setup* tab of ADE Assembler, double-click in the *Spec* column of the a *Output Expression* which corresponds to the cross *Expression* to configure a desired value range. A selection menu opens from which you can select the criteria shown in Figure 104. Click on *range* and select a value range from 1.55 V to 1.56 V.

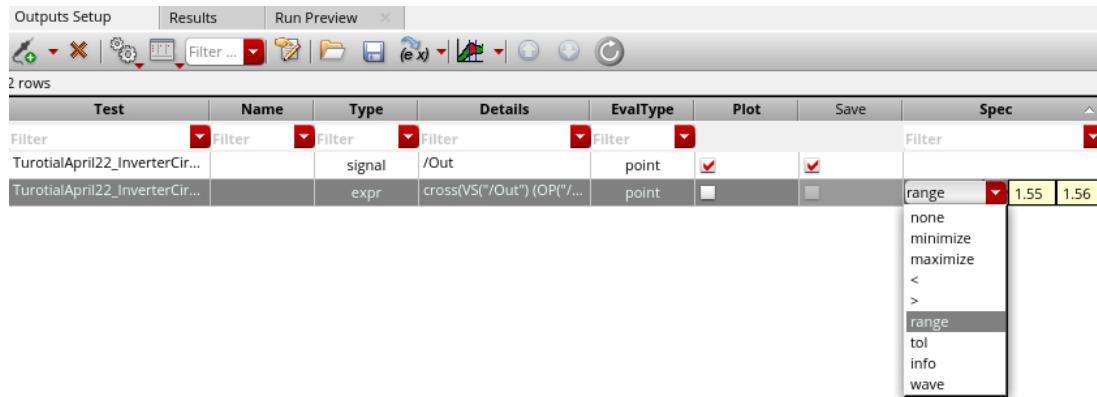


Figure 104: Specification of a *Spec* in the ADE Assembler output window

After the simulation run, you can check the *Spec* specification in the *Results* area of ADE Explorer. Values that match the specification are highlighted in green. Results that come close to the limit value are highlighted in yellow and values that are clearly outside the specification are shown in red.

The screenshot shows the ADE Explorer interface with the 'Results' tab selected. On the left, a table shows simulation corner data for 'mm180_reg33_v114.lib.scs'. On the right, a table lists results for the expression 'cross(VS("/Out") (OP(...'. The results table has three columns: 'Pass/Fail', '/Out', and 'cross(VS("/Out") (OP(...'. The values are color-coded: green for 'pass' (e.g., 1.557, 1.551, 1.555), yellow for 'near' values (e.g., 1.614, 1.617, 1.622), and red for values outside the range (e.g., 1.547, 1.55, 1.553, 1.614, 1.617, 1.622).

Figure 105: Simulation results with colour-coded values

15 The Monte Carlo simulation

15.1 Setting up a Monte Carlo simulation

With the help of Monte-Carlo simulations, it is possible to study the influence of random process fluctuations on the behaviour of the circuit and to determine statistical parameters for relevant circuit properties. While in the *Corner* simulation the properties of all transistors are influenced equally and set to a specific value of the expected process variance (*slow*, *typical*, *fast*), the Monte Carlo simulation enables the independent and random change of the properties of the transistors used within the expected variation spectrum of the process. The statistical distribution of the analysed circuit variable can then be taken from the simulation results and characteristics such as the mean value and standard deviation can be extracted.

The Monte Carlo simulation is also set up under Cadence in the *Corner Setup* window. With the *UMC-180nm* technology, it is necessary to import an additional *Model File*, which contains the statistical parameters of the process variation. To do this, click on *Add/Edit Model Files*. In the following window, click on *Click to add* in the *Model* area. The *Choose Model File* window opens (Figure 106). The *mm180_reg33_v114_mc_corner.lib.scs* file can be found under the path *"/eda/kits/UMC180/Models/Spectre/Monte_Carlo"*. Select this file and click on *Open*.

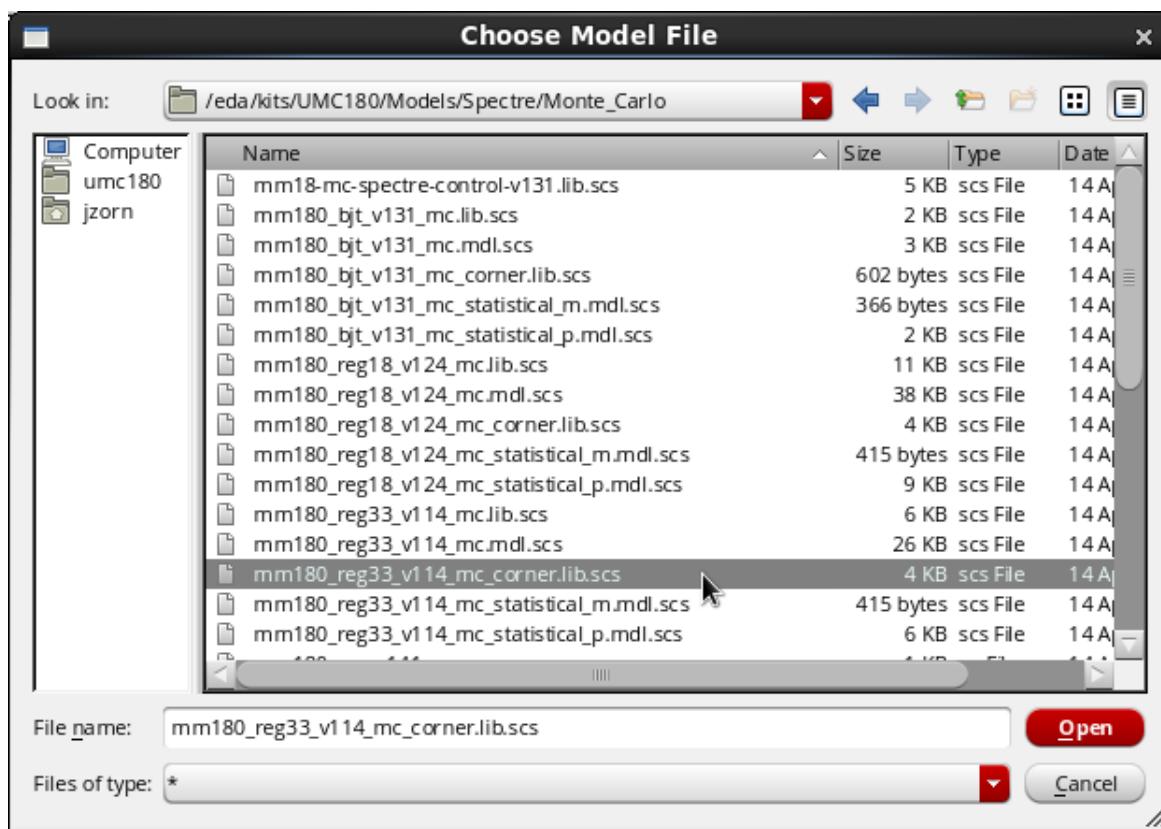


Figure 106: Inserting the model files of the Monte-Carlo-Simulation

The selected file is now in the *Model Files* area. Now create a new column to set up a special Corner for the Monte Carlo simulation. Instead of the *Process Corners* already introduced, now select the option *mc* for the Monte Carlo simulation (see *Figure 107*). Deselect the nominal and the previously created corner simulations and close the *Corners Setup* window by clicking on *OK*.

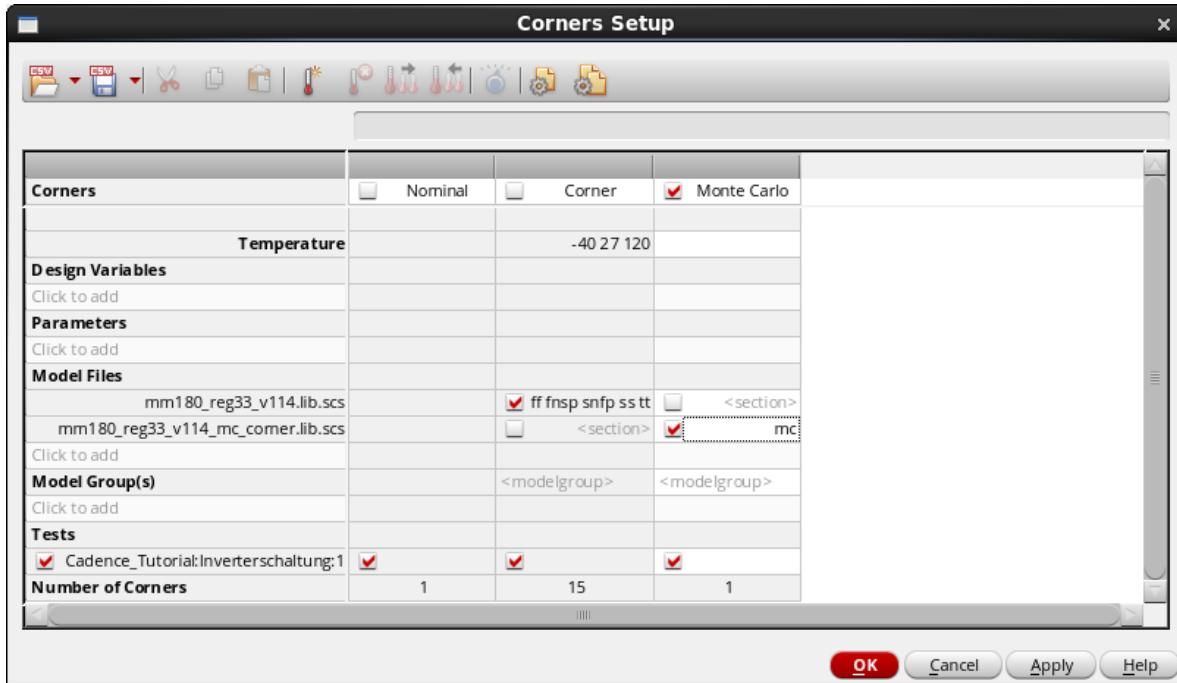


Figure 107: Setting up the Monte Carlo simulation

Also use the test created in *Chapter 13.4* for the Monte Carlo simulation. Before the simulation can be started, you must create a global variable *sigma* in ADE Assembler and assign it the value 3 (*Figure 108*). This variable describes the standard deviation of the statistical simulation.



Figure 108: Creating the global variable *sigma*

Now open the menu for selecting the analyses in the centre of the top toolbar of the ADE Assembler window and select the option *Monte Carlo Sampling* (compare *Figure 109*).

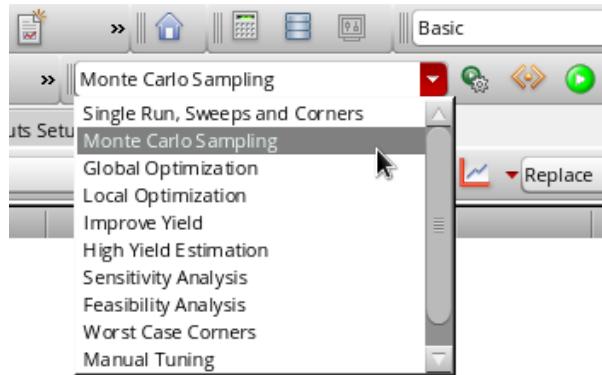


Figure 109: Selection of the various analysis options in ADE Assembler

On the right-hand side of the selection list is an icon with a green arrow and a cogwheel (*Simulation Options*). Clicking on this button opens a window in which further settings for the Monte-Carlo simulation can be made (see Figure 110). The number of points to be simulated can be set here in the *Sampling Method* area in the *Number of Points* field. By default, this is already set to *200 points*. A higher number can improve result accuracy, at the expense of simulation time.

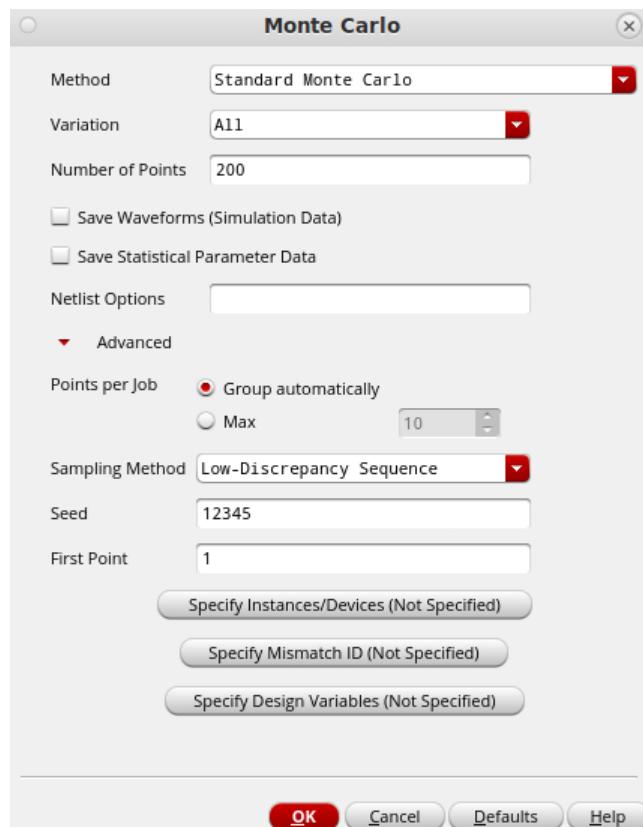


Figure 110: Further settings for Monte-Carlo-Simulation

The Monte-Carlo simulation can then be started by clicking on *Run Simulation*. Once all the previously configured simulation runs have been completed, a histogram opens showing the distribution of the switching voltage of the inverter.

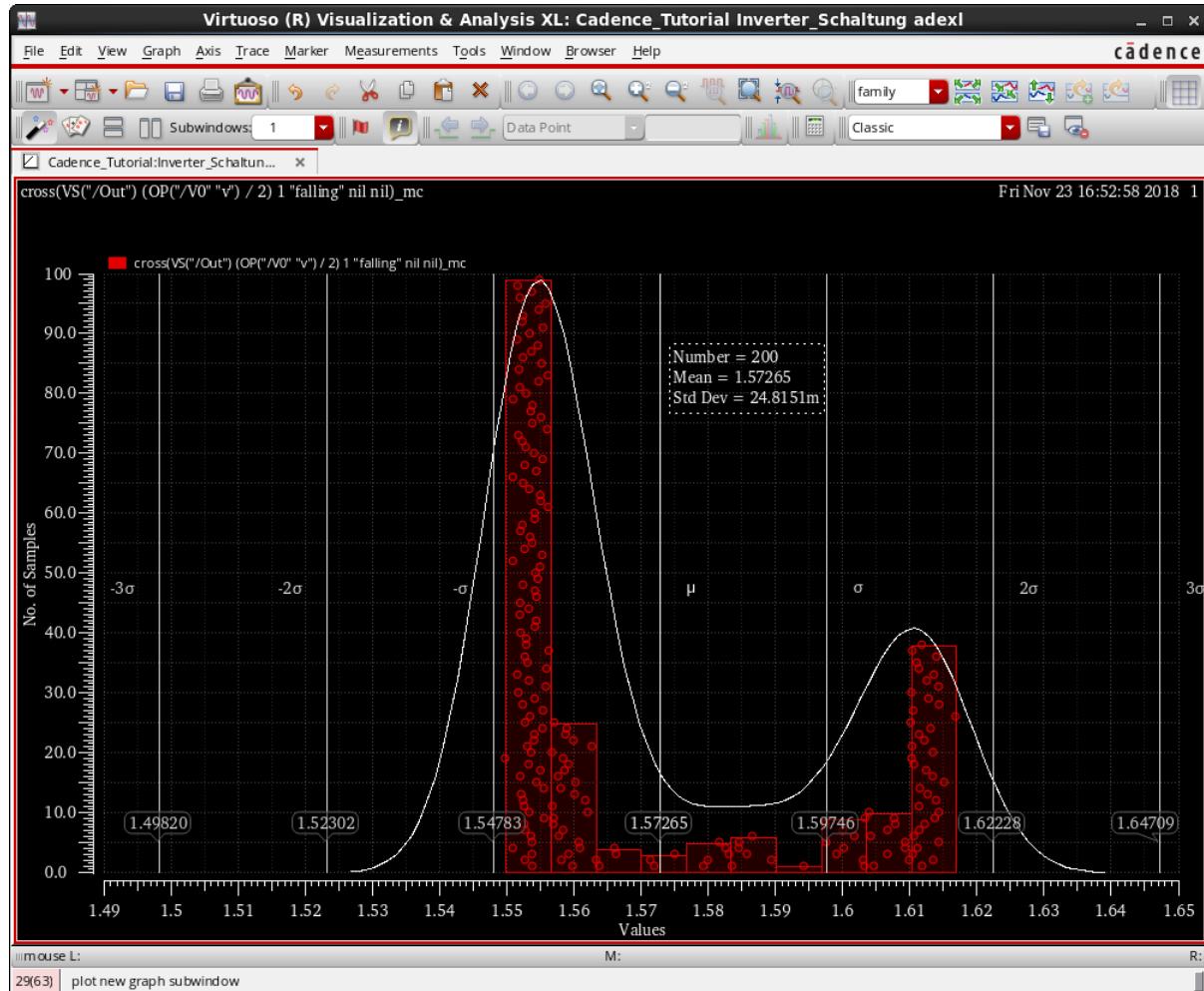


Figure 111: The distribution of 200 Values of the Monte-Carlo-Simulation

15.2 Limit values (specs) in a Monte Carlo simulation

Defined Specs are also taken into account in the histogrammed display of the Monte Carlo simulation results. To set up a Spec preset, proceed as described in Chapter 14.2. For this example, an interval of 1.55 V - 1.56 V was selected for the cross expression. An iteration count of 1,000 iterations was also set. The results outside the specifications (*fail*) are in the red section, while the matching (*pass*) values are marked in the green section.

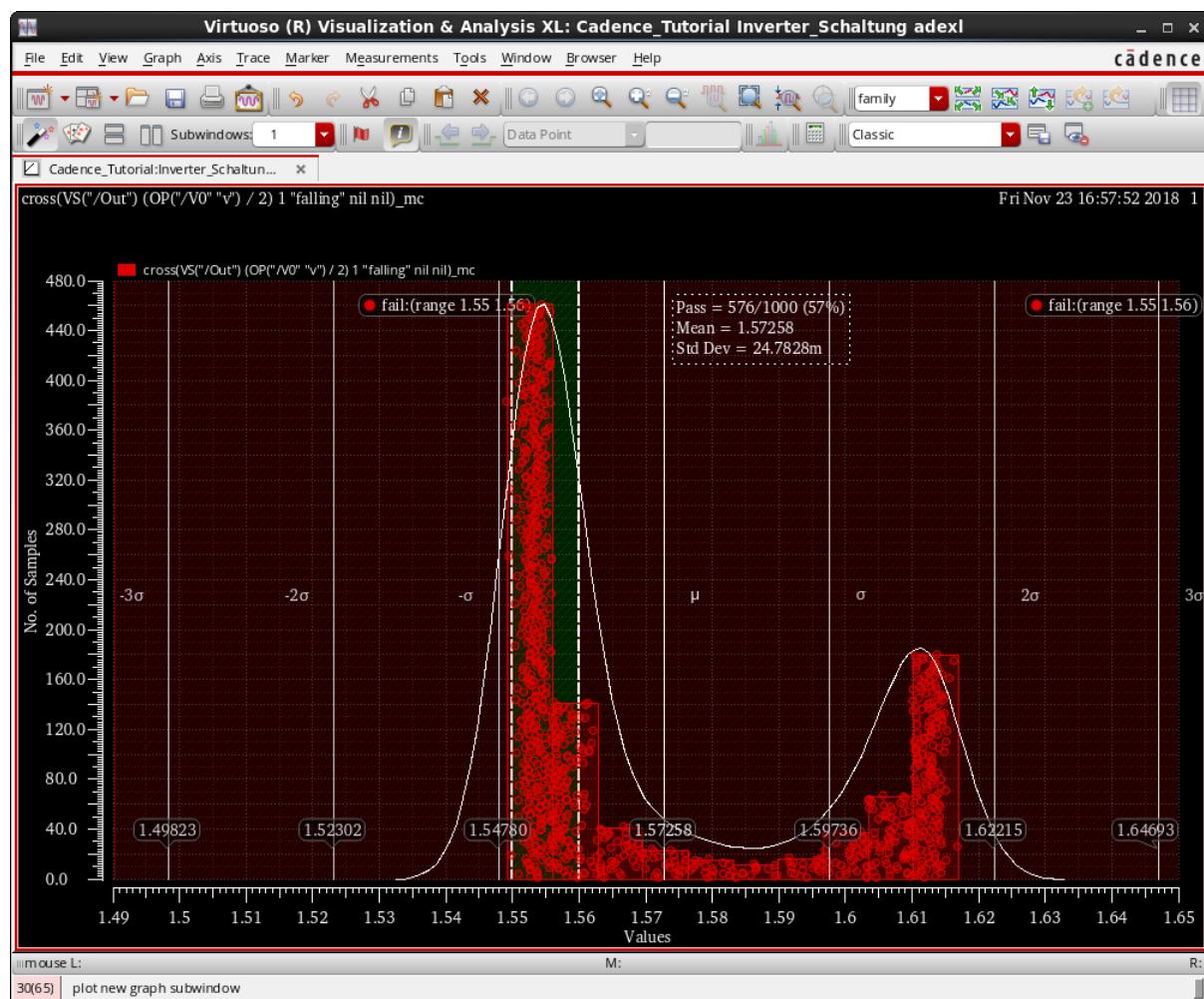


Figure 112: Plot of the Monte-Carlo-Simulation with defined limits

16 Layout

16.1 Introduction

This chapter provides an introduction to the methods and tools used to create an integrated circuit layout. This development step is still carried out manually, especially in analog circuit design and full-custom design of digital circuits, and is carried out after schematic creation and functional testing by extensive simulations. During layout creation, the degree of abstraction of the representation of an integrated circuit is reduced and transferred from a circuit diagram consisting of component symbols and their connection via nets to a two-dimensional view of the spatial arrangement of components and conductor paths.

The layout consists of geometric structures in different layers, which correspond to different manufacturing steps in chip production. These layers are displayed in the layout editor in different colors and patterns, either transparent or non-transparent. Individual layers are used in the production of integrated circuits for the production of masks, which are used in the individual photolithographic process steps.

When creating the layout, it must be taken into account that the components and materials used do not behave ideally in an electrical sense and therefore always cause parasitic effects, such as additional and previously unconsidered track resistances or capacitances. This is a conscious approach when creating a layout. For example, it makes little sense to route an expected current of several milliamperes through a conductor track that is less than half a micrometer wide. It must also be taken into account that small variations in component properties can occur during production. For components where matching is of great importance for the function of the circuit, special measures can be taken to reduce these variations to a minimum. For example, when creating the layout, care must be taken to ensure that the alignment of matching components is identical and the environment of these components is similar, which can often only be guaranteed by using dummy structures, i.e. additional components without an electrical function.

16.2 Elements of the Layout

This section explains important terms used in the layout process.

- **Substrate:** The basic material used in the manufacturing process are thin disc shaped slices (typically 0.3-0.5 mm thick), which are cut out of a silicon single crystal and are usually P-doped. These slices are referred to as wafers or substrates. In the layout editor, the substrate area, which is usually shown in black, is automatically displayed wherever no layer structures are used. It should be noted that a substrate connection is usually required in close proximity to the active components. This substrate connection is often placed around the active component as a closed ring, the so-called guard ring.
- **Well:** A well is a spatially limited area with a different doping from the substrate. The well extends a few μm into the depth of the substrate. Wells are used to create PN junctions and are therefore an important component of transistors. The wafer is doped with a homogeneous particle beam, which is applied to certain areas of the wafer by using a mask. In this way, all wells of one type on the wafer are created simultaneously in one step and are therefore combined into one layer.
- **N-Well:** An N-doped well is the most common type of well. As with all other layers, two touching N-wells are interpreted as a single well. If two N-wells do not touch, a safety distance must be maintained between the N-wells. Such distance rules apply to most layers and are therefore automatically checked by a special checking tool.

- **Diffusion:** This is an area of strong doping and very low resistance. The doping can either be P-type or N-type. Diffusion layers are usually used for the realization of substrate and well contacts but also for the source and drain electrodes of transistors. The name of the layer is derived from the fact that it is produced using a process in which the doping atoms are deposited on the surface of the wafer and diffuse into the semiconductor through heating.
- **Poly:** The term poly stands for polycrystalline silicon. The poly lies on a separate layer below the first metal layer. The poly layer is combined with the diffusion layer in order identify MOS transistors. Due to its high electrical conductivity, polysilicon can also be used as an additional layer for wiring. However, care must be taken to ensure that poly connections are not routed across diffusion zones, as the combination of diffusion zone and poly results in a transistor, as described above. It should also be noted that the poly has a lower conductivity compared to the metal layers, which can lead to problems, especially in applications which require fast switching.
- **Metal:** Metal layers create the electrical connection between components. Due to the large number of required connections, there are several metal layers in each technology. When using metallayers, it makes sense to assign a preferred direction to each layer. The first metal layer (M1) is usually used for connections in a horizontal direction, while the second metal layer (M2) is used for routing in a vertical direction. The third metal layer (M3) should then routed in the horizontal direction again, and so on. One advantage of this arrangement is that closed circuits are avoided on the respective metal layers, which simplifies routing and improves the clarity of the layout.
- **Contact/Via:** Contacts and vias create connections between different layers. In the literature, contacts that create a connection between metal layers are referred to as vias and only contacts between the lowest metal layer and a semiconductor layer, e.g. diffusion or poly, are regarded as actual contacts. You can create contacts/vias via the shortcut "o" or using the menu item Create/Contact. The names of the contact types are derived from the designation of the layers to be connected. For example, a contact that creates a connection between metal layers 1 and 2 can have the designation M1 M2. To contact an N-well, you need a contact that creates a connection from the first metal layer (M1) to an N-doped diffusion area. This contact can be found under the designation M1 NDIFF or under the designation M1 NWELL (M1 NDIF= Contact with additional N-well layer) As the substrate (wafer) is P-doped, it can be contacted accordingly using the contact with the designation M1 PDiff.
- **Pins/Labels:** Pins represent virtual ports within a project that consists of several layout cells. By using pins, Cadence makes it possible to analyze a project distributed over many cells as a whole and check it for correctness. A distinction is made between standard pins/net names and global net names. When using a global network (GND, VDD ...), it is important to ensure that the network name always begins with a exclamation mark and the designation always corresponds to the network name used in the circuit diagram. Example: The net vddd was used in the schematics, so a label must be set on the corresponding net in the layout, which is given the name vddd. It is important that the program is informed that the net name is defined by this label. This is done by using a special layer with the designation MX_CAD and the "Purpose" text. To place a lable to a metal layer of type Metal layer 1, the label command must be used with the M1_CAD Text layer. Proceed in the same way for other metal layers.

16.3 The Virtuoso Layout Editor

The Virtuoso Layout Editor corresponds in many aspects to the circuit diagram editor. However, some things are more complicated because the layout is topological representation of an IC on a wafer. While the position of a component in a circuit diagram can be arbitrarily chosen. The position of a component in the layout defines its actual position on the wafer. The function of the layout is explained using a detailed test run. This is followed by a few explanatory sections.

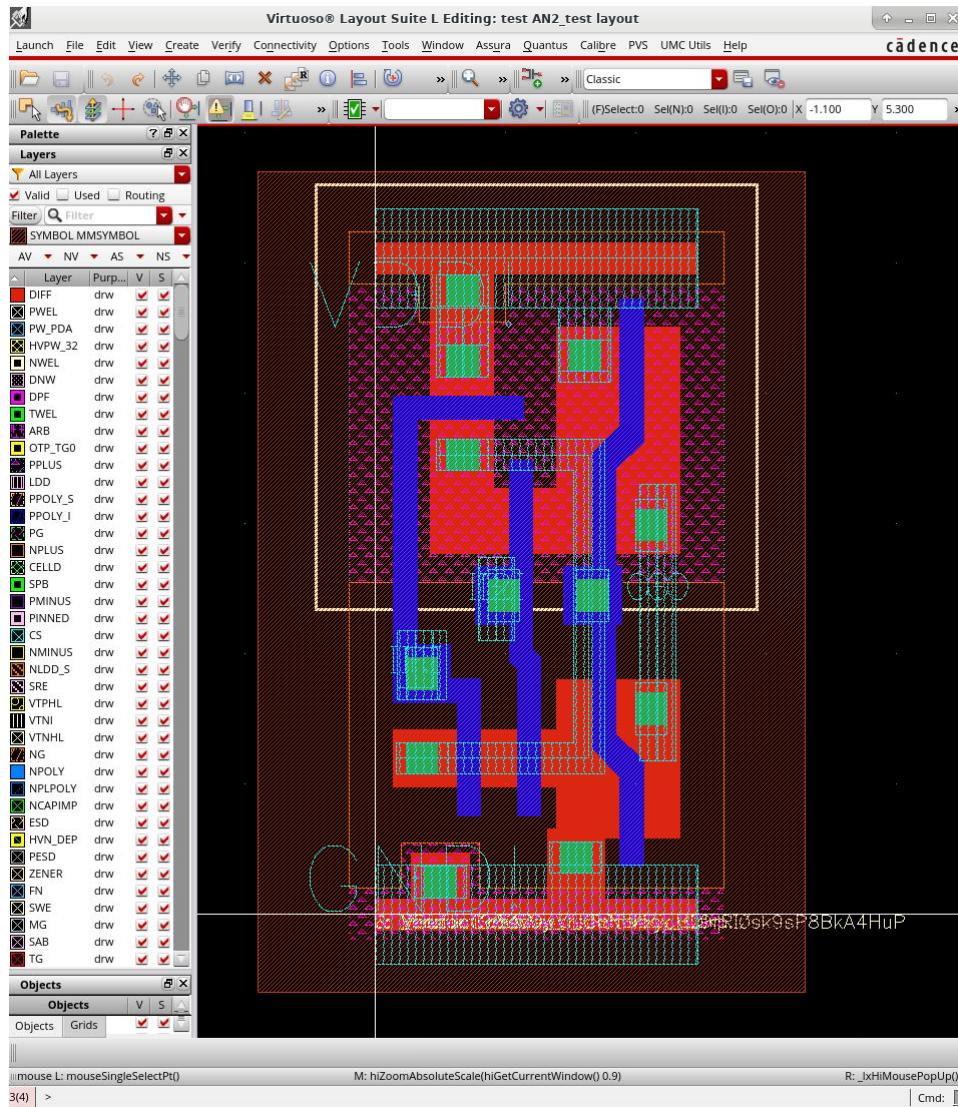


Figure 113: Demonstration of a layout

16.4 The Layer Window

All layers and the colors or patterns assigned to the layers are listed in the layer section. This window is the most important window next to the layout editor itself (see Figure 113, LSW on the left, layout editor on the right) and can therefore not be closed except by exiting Cadence. The LSW window controls the display of all layers and provides a legend for the colors and patterns of individual layers. The complex structure of an IC is created by skillfully superimposing layers in different areas. For example, the basis of a MOSFET is the deposition of n-doped areas (N-wells) on the P-doped wafer. It is therefore necessary to display a large number of specific layers simultaneously and edit them. The LWS window with its four buttons is used for this purpose:

- **AV=All Visible:** All layers are displayed
- **NV=None Visible:** None layers are displayed
- **AS=All Selected:** All layers can be selected. Selected areas can be edited, copied, deleted, moved, enlarged, etc.
- **NS=None Selected:** No layer can be selected.

In addition to the layers that correspond to processing steps, there are a number of symbolic layers that are provided for markers or e.g. for creating comments or designations and thus facilitate the documentation of the design.

16.5 Test Run

16.5.1 Preparation

The starting point for creating the layout is the schematic of the inverter circuit that has already been created in *Chapters 9 and 10*.

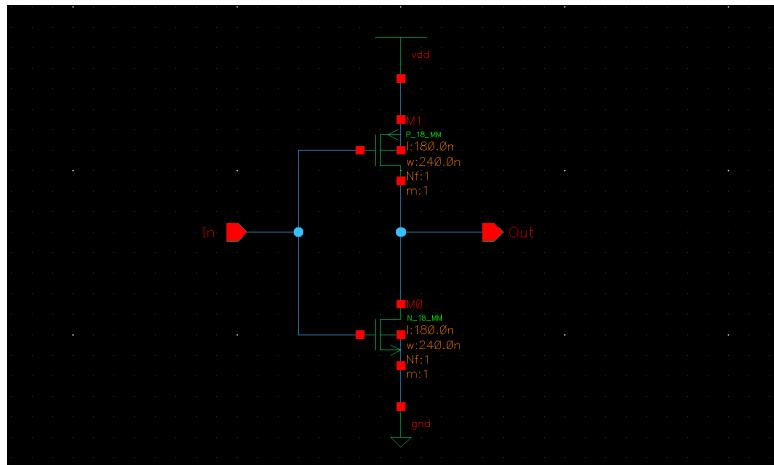


Figure 114: Schematic of an inverter for which the layout is to be drawn

Another cell view is added to the existing schematic. To do this, open the Library Manager in the Virtuoso window and select the existing cell. First, select the library in which the schematic is saved, and then the corresponding schematic. Once both have been selected, a new cell view can be created via *File → New → Cell View*, which is linked to the existing cell. The cell name is already entered in the window that opens (see Figure 115) and should not be changed. The only thing that needs to be adjusted at this point is the type of the new file *Layout* must be selected instead of *Schematic*.

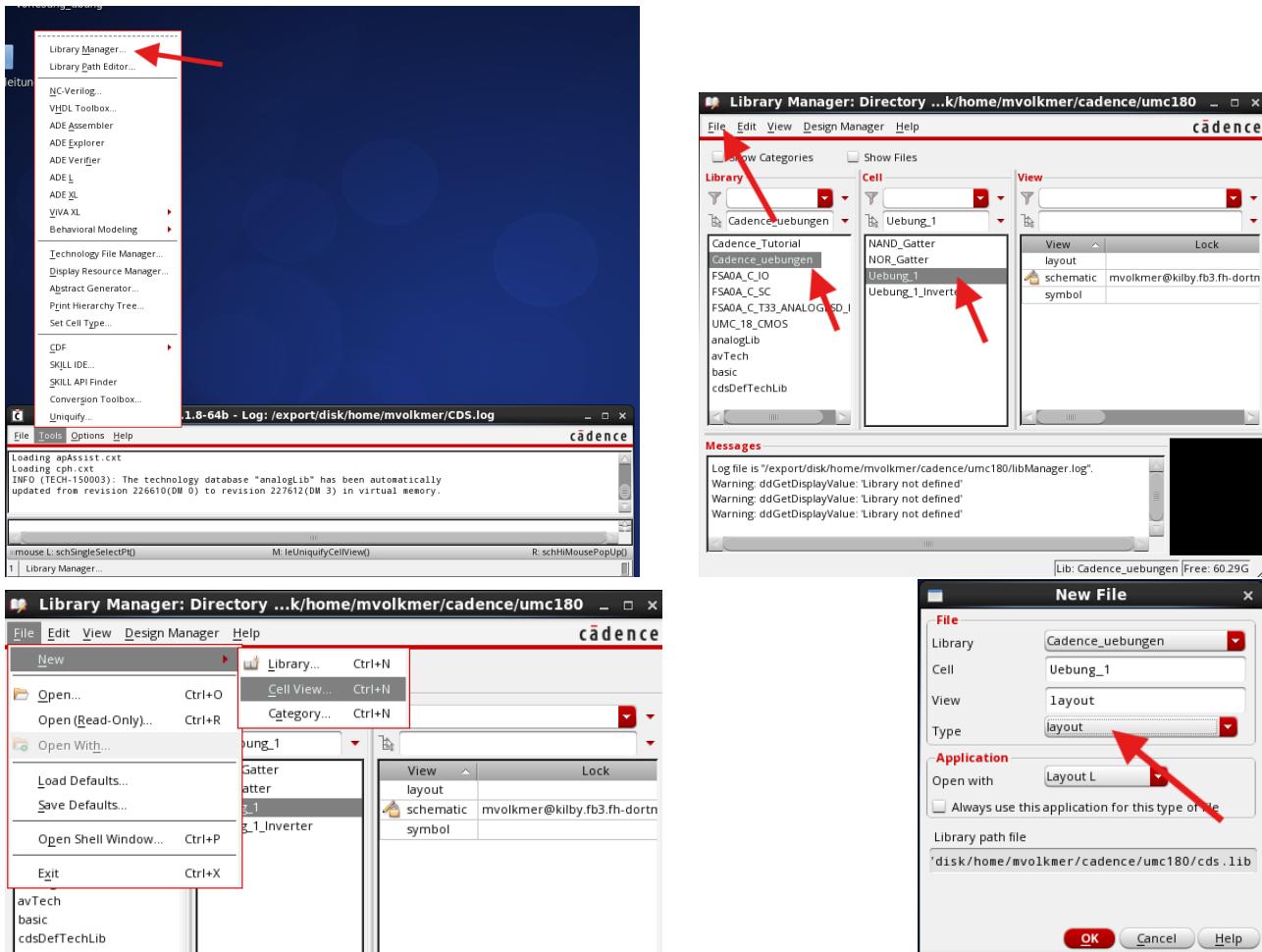


Figure 115: Adding another cell view

After opening the layout creation window, it is important to first define the technology grid. You can access the „Display Options“ through the *Options → Display* or the **E** button (Figure 116). The grid can be defined there using the *X Snap Spacing* and *Y Snap Spacing* parameters. In this technology, the grid size is $0.01 \mu\text{m}$ (Figure 117).

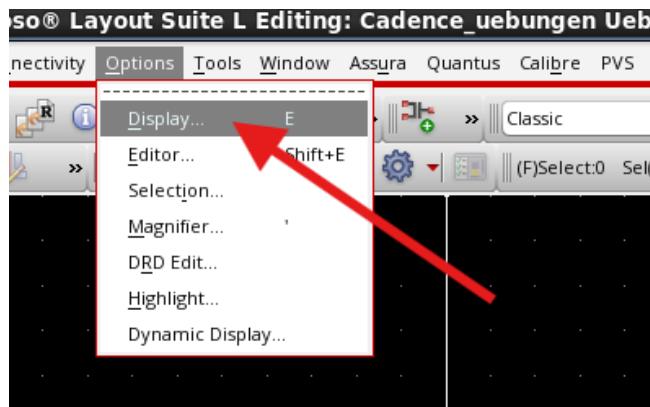


Figure 116: Path to the grid setting

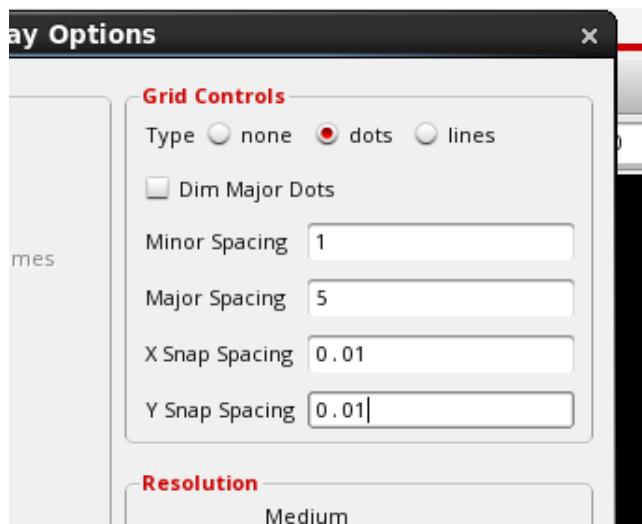


Figure 117: Grid settings

16.6 Insert Instances

The layout process begins with the addition of all the required components. These can be added via the *Create → Instance → Browse* or by pressing the **I** button. The transistors are located in the „UMC_18_CMOS“library. In order a component to be placed, the *layout* option must be selected in the "View" category (Figure118).

By default transistors and other layout instances are shown in a simplified manner as transparent red rectangles. to increase clarity and optimize execution speed (Figure 119a). To display the details of the component (Figure 119b), press **Shift + F**. To return to the previous view, use **ctrl + F**.

To move or rotate components, press the **M** button then click on the desired component and move it directly. Alternatively, you can press **Shift + O** for rotation (Figure 120). The direction of movement of the component can also be defined in the same window (snap mode).

Visibility V and **Selectivity S** can be used to control the visibility and selectability of a specific layers. If *Used* is checked, only the layers that have already been used are displayed, which provides a better overview. Just as the individual layers can be selected and made visible, individual objects can also be selected and made visible. In this example, only the diffusion layer of the PMOS is visible (Figure 121).

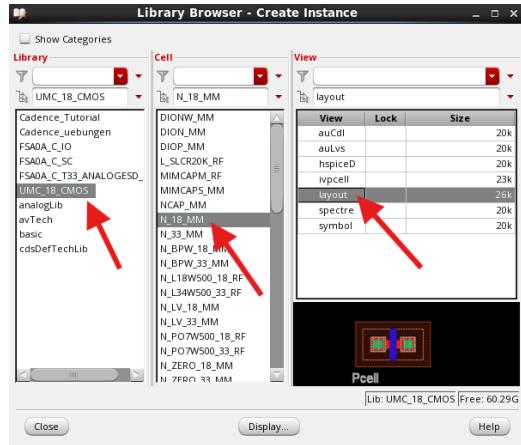


Figure 118: Selection of components

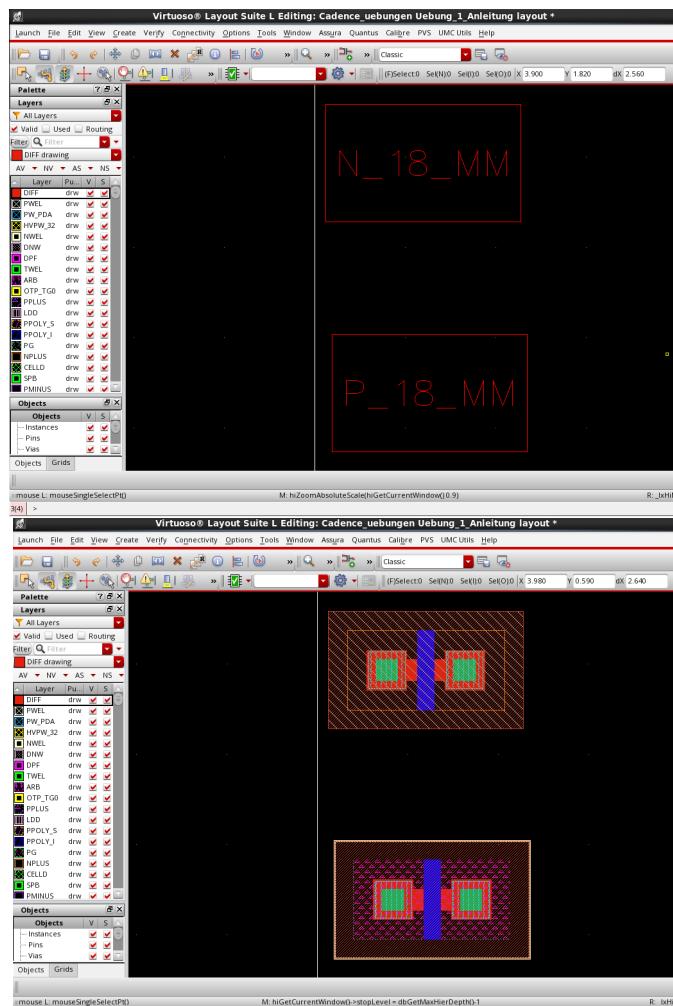


Figure 119: Changing the views: (a) Simplified view (b) Complex view

Vias can be created through the menu option *Create* → *Via* or by pressing the **O** button. In the window that opens (Figure 122), select the appropriate contacts under "Via Definition". For example „M1_NWEL“ is the

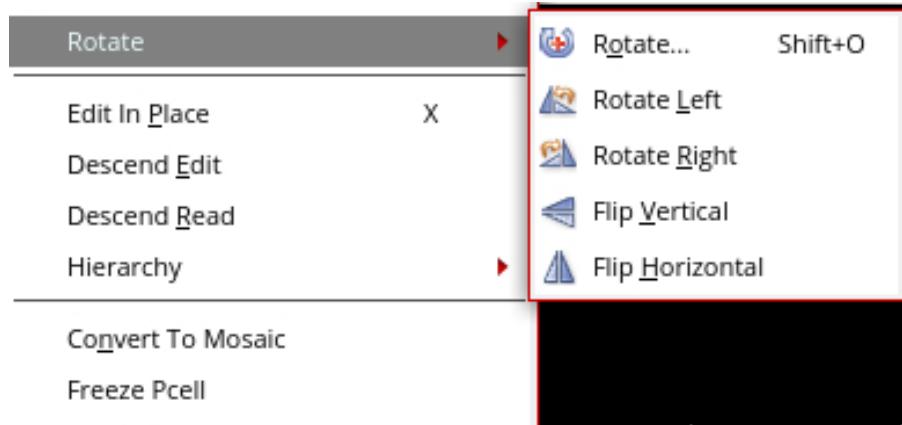


Figure 120: Rotating the component

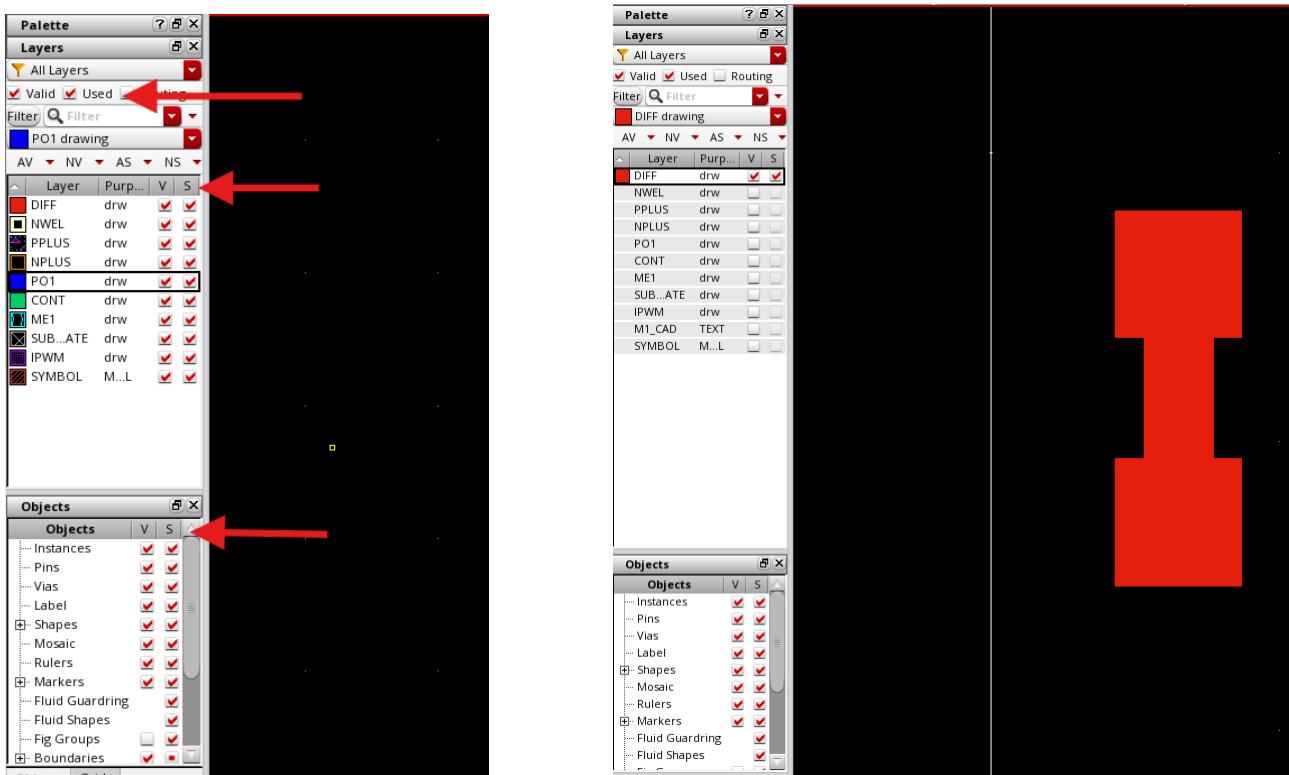


Figure 121: Selection of layers and objects and their visibility settings: (a) Display of the visibility and selection options (b) View of the diffusion layer

contact for connection between first metal level M1 with the NWELL.



Figure 122: Selection of contacts

16.7 Early Design Rule Check

The aim of the *Design Rule Checks* (DRC) is to ensure that the layout complies with the technological specifications and manufacturing restrictions. Without a DRC, errors in the design could be overlooked, leading to production downtime or reduced performance later on. During DRC, the layout is compared with a series of predefined rules that are specified by the manufacturing technology (e.g. CMOS process). These rules define minimum distances, sizes and overlaps for the various layers of the layout, such as track spacing, layer thicknesses and via positioning.

In this example, the DRC checks, among other things, the minimum distance between polysilicon structures that act as gates, as well as the correct placement and dimensioning of the source and drain areas. In addition, the position and size of the contact windows between polysilicon, diffusion and metal are checked.

It is a good strategy to perform the *Design Rule Check* before routing, as this allows errors such as incorrect spacing, layer violations or minimum size problems to be detected at an early stage. This reduces the workload as changes made more easily and quickly at an early stage. In addition cascade effects are avoided, where an error could have a negative impact on later layout steps. A clean state through early DRC checks eases planning and routing considerably easier, as problematic areas can be avoided from the outset. Overall, the layout process becomes more efficient as later debugging phases can focus on routing without being affected by fundamental design errors.

The DRCrun is started via *Calibre* → *Run nmDRC* (Figure123). First, a window opens in which you are asked to select a *Runsetfile*. The „*calibre.drc*“ file, located in the working directory from which you started virtuoso, contains all required settings and points to check the circuit. Select this file and confirm the selection with „OK“. Then you can start DRC via *Run DRC* (Figure124).

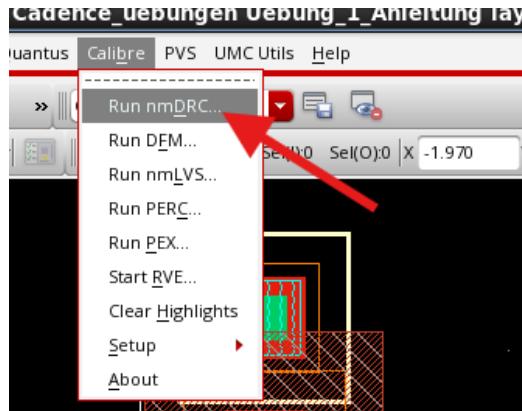


Figure 123: Run Calibre nmDRC

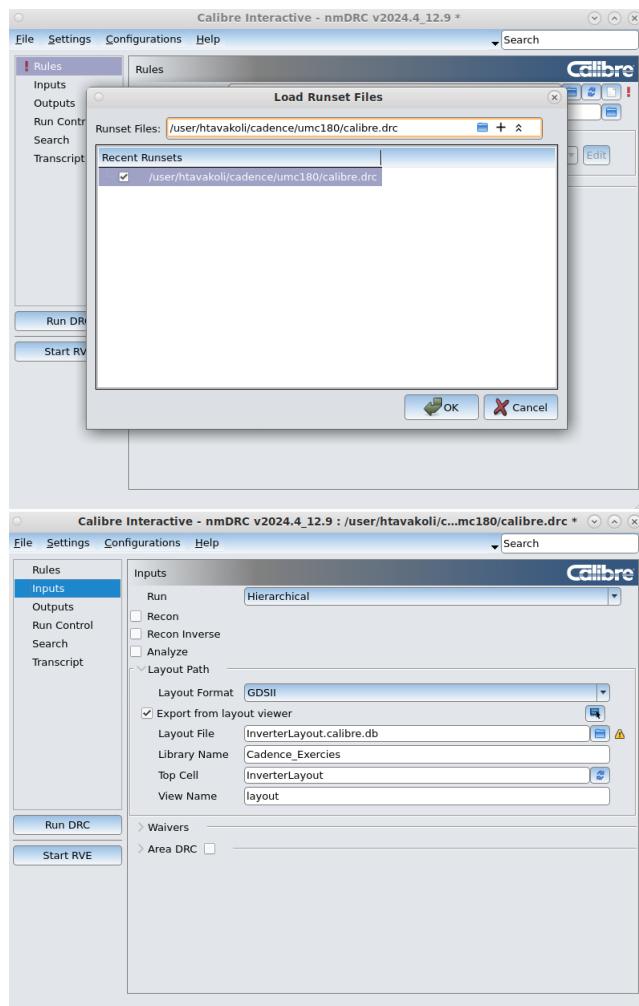


Figure 124: Design Rule Check

The window that is displayed after the DRC (Figure 125) shows a list of errors in the left-hand area grouped by violation types. If an error is selected, a number appears in the middle section of the window and a detailed error description in the lower section (highlighted in red). Clicking on the number opens another window that displays further details about the error. To visualize the error in the layout, it can be selected with a right-click by choosing „Highlight“. This highlights the error directly in the layout. By clicking on the displayed coordinates, the layout is zoomed in at the corresponding point so that the error can be examined more closely and, in the best case, rectified (Figure 126).

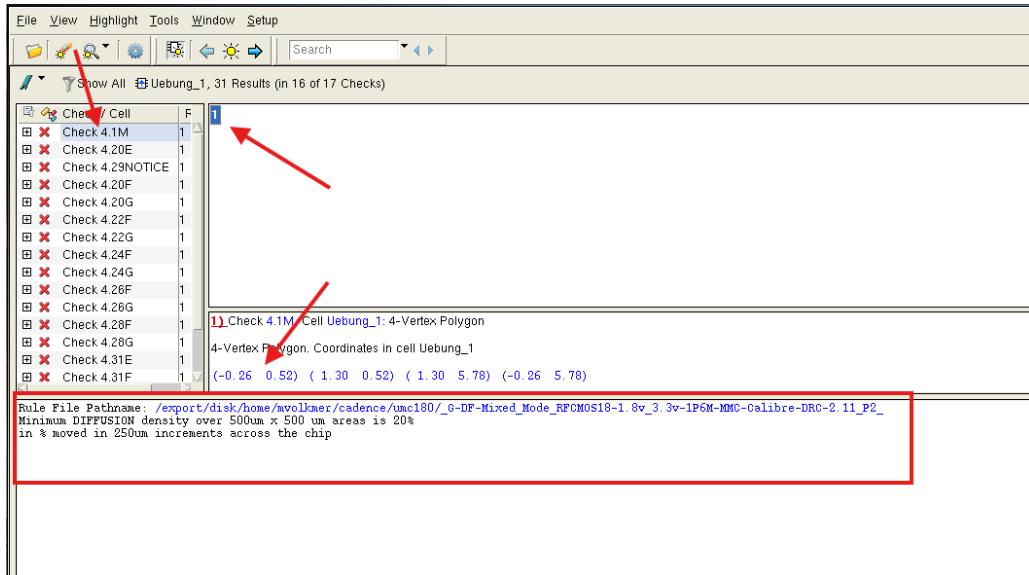


Figure 125: Error output

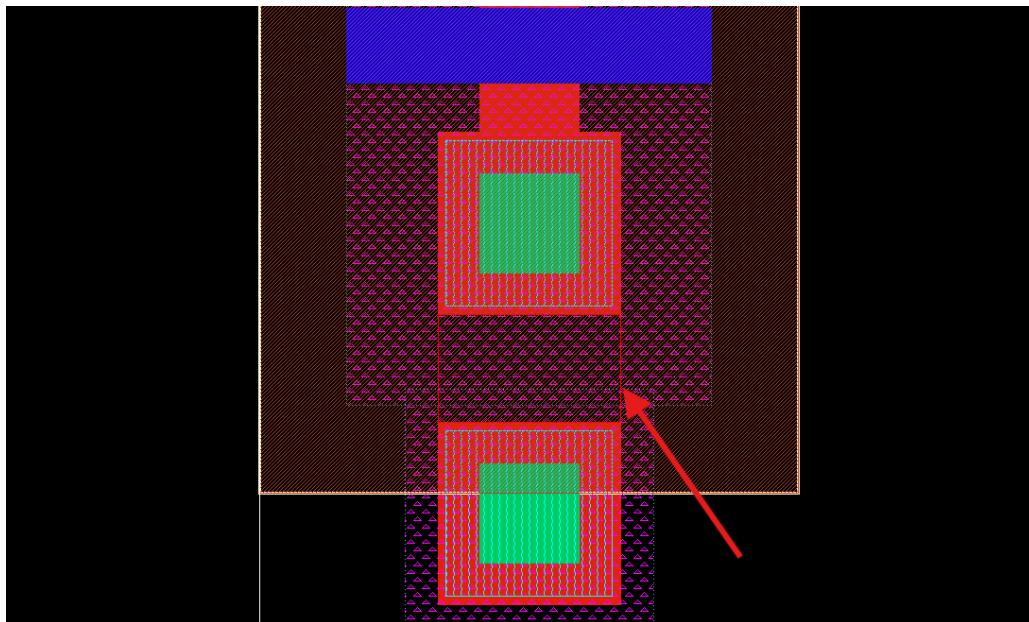


Figure 126: Display of the error

Many design errors are caused by insufficient spacing between certain layers. The ruler function is used to maintain these distances correctly. This can be activated either via *Tools → Create Measurement* (Figure127) or by pressing the **K** key. After selecting the ruler, the start and end point of the measurement can be defined with the mouse, whereby the measurement is displayed directly in the layout and remains there (Figure128).

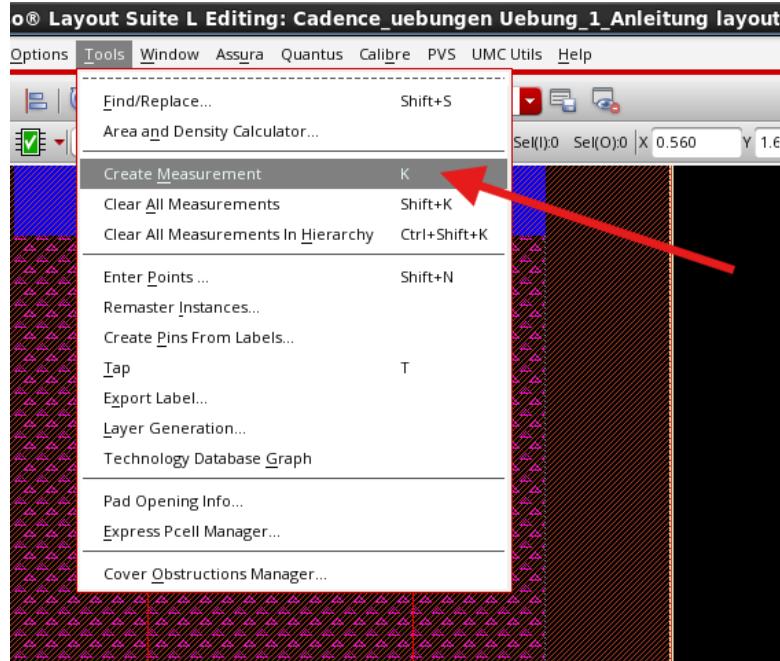


Figure 127: Path to the ruler

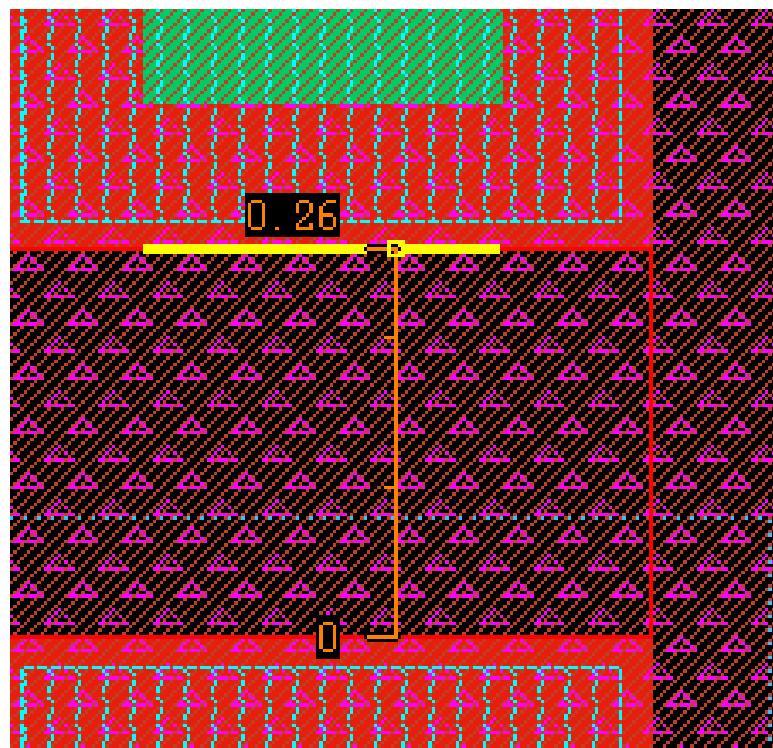


Figure 128: View of ruler

Spacing errors can also be avoided by expanding the layer so that there is no more spacing. To do this, select the layer to be expanded and either use *Create → Shape → Rectangle* or press the **R** button to draw a rectangle with the mouse that overlaps the selected layer and thus completely eliminates the spacing (Figure 129).

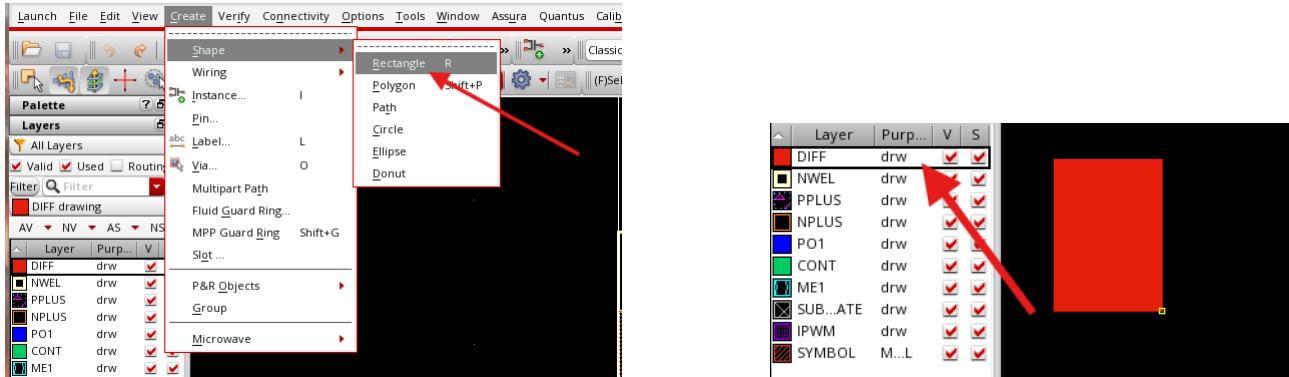


Figure 129: Creating a layer

Errors that affect the minimum area or coverage of a metal layer can be ignored at this point for the time being since additional metal area will be introduced during wiring. Once all other errors have been corrected, routing can begin.

16.8 Routing and labels

Routing is an essential step in circuit design in which the connections between the components are realized. Traces are arranged on the layout levels in such a way that the electrical signals are transmitted correctly and all design rules are adhered to.

The connections are created using *Paths*. To draw a trace, first select the corresponding layer and start the process via *Create → Wiring → Wire* or by pressing the **P** key (Figure 130). The first left mouse click starts the routing, further clicks change the direction of the wire and a double-click ends the routing of the current connection. If the connection area of the new line is selected, Cadence usually automatically adopts the existing width. Alternatively, the width of the line can be adjusted manually via the properties (**Q**). Connect the drains and also the gates of the PMOS and NMOS with Metal1 according to the connections available in the inverter schematic. For the gate connection also the placement of M1 to POLY contacts is required.

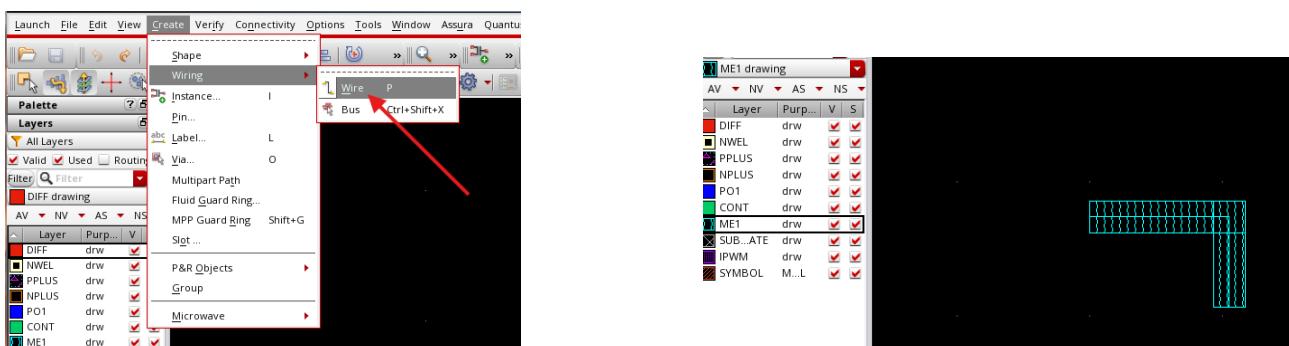


Figure 130: Creating a connection

A correct connection of all components is checked by the layout vs. schematic test (LVS) check discussed in

the next section. For this to be successful, the same connections must be defined in the layout as in the schematic. Labels for these connections can be created either via *Create → Label* or by pressing the **L** button. A window opens (Figure 131) in which the connection name is entered. A trailing exclamation mark indicates global variables. The text size in the layout can be adjusted using the *Height* field.

Once the settings have been completed, the label is placed in the layout, ideally in the middle of the corresponding layer. The process is ended by pressing the **Esc** button. Introduce labels for all pins available in the schematic and the respective position in layout. The layout should look like Figure 132. Rerun the DRC check and correct all reported errors.

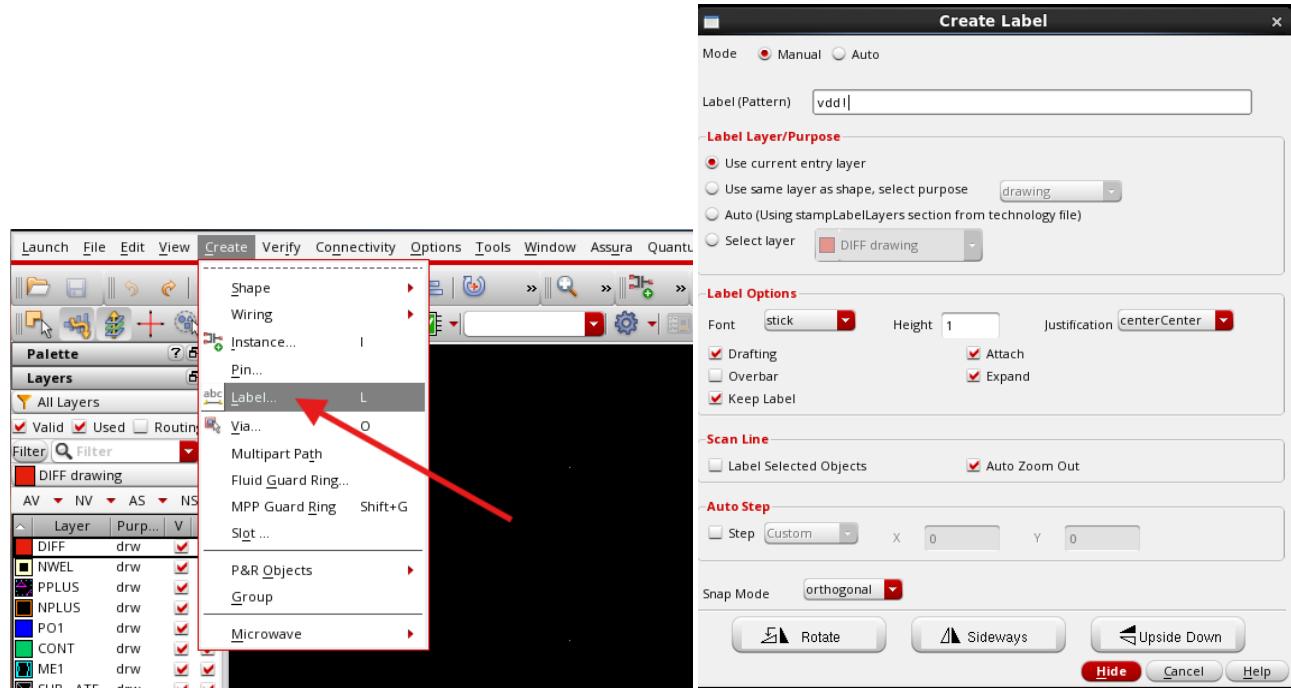


Figure 131: Creating a label

16.9 Layout-vs.-Schematic

The layout vs. schematic (LVS) checks ensure that the layout created corresponds functionally to the associated circuit diagram. It checks whether all connections, components and parameters defined in the circuit diagram have been correctly implemented in the layout. Errors, such as missing connections or incorrect dimensioning, can be detected and corrected at an early stage.

To start the LVS, select menu option *Calibre → Run nmLVS*. Similar to the DRC, a window opens in which you must read in a runset file. Proceed as with the DRC, but select the file „calibre.lvs“ instead. The rest of the process is identical to the DRC test.

If the layout vs. schematic test (LVS) has been successfully passed, this means that the layout fully matches the circuit diagram and all defined connections and components have been implemented correctly (Figure 132 and 133).

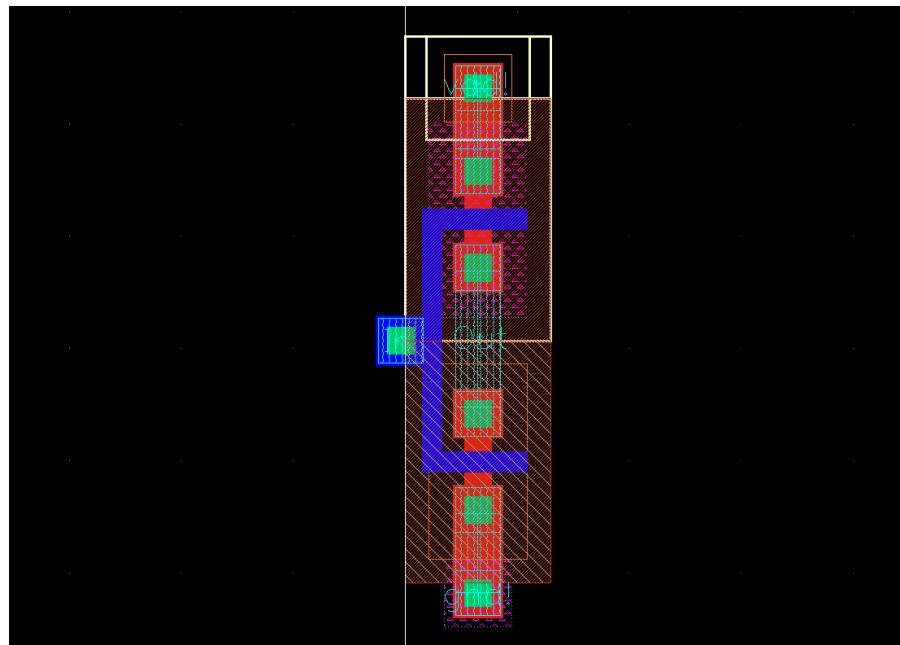


Figure 132: View of the finished layout

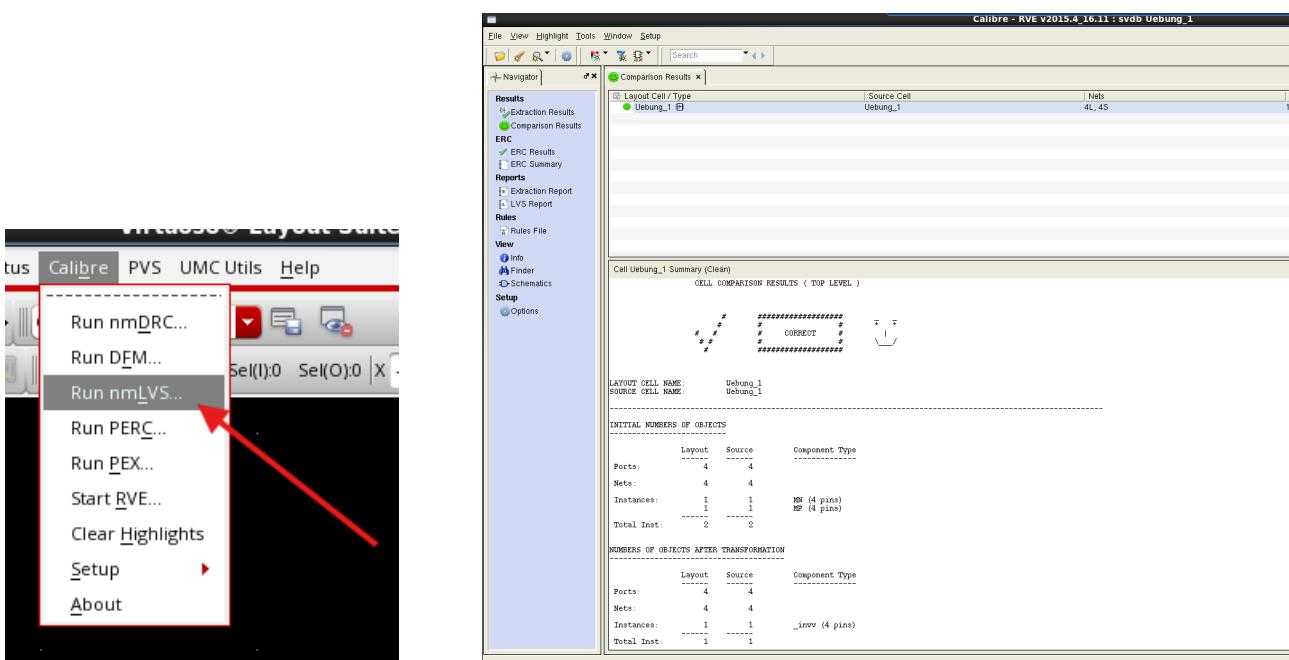


Figure 133: Layout vs. schematic

17 Appendix

17.1 The ADE-L main window

In Cadence Virtuoso, the circuits designed in the *Schematic Editor* can be simulated in various ways. One option is to use the **Analog Design Environment L(ADE-L)** simulation environment. This tool can be accessed in the *Schematic Editor* under the menu *Launch > ADE-L*. Figure 134 shows the still empty main window of ADE-L. Various simulations can be configured and executed in this window. These include, for example, DC analyzes to determine the operating point, transient analyzes for the simulation of time-varying signals and AC analyzes which carry out a frequency simulation in the small signal equivalent circuit. Furthermore, simulation results can also be displayed graphically via ADE-L.

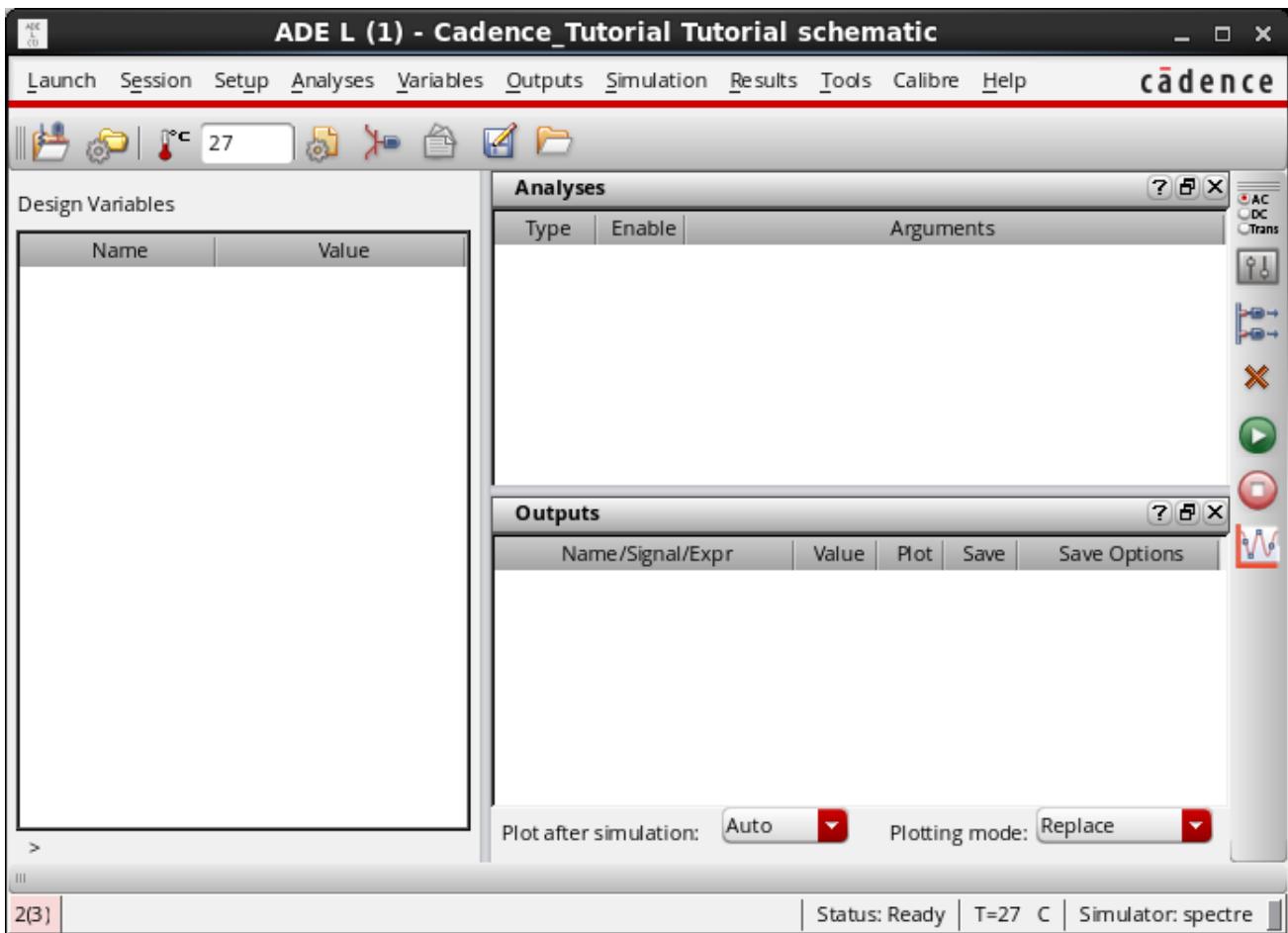


Figure 134: The main window of ADE-L

17.1.1 Importing variables from the schematic

Before a simulation can be carried out, the variables specified in the *Schematic* must be imported into ADE-L. This is accomplished via the *Design Variables* area on the left side of the ADE-L main window.

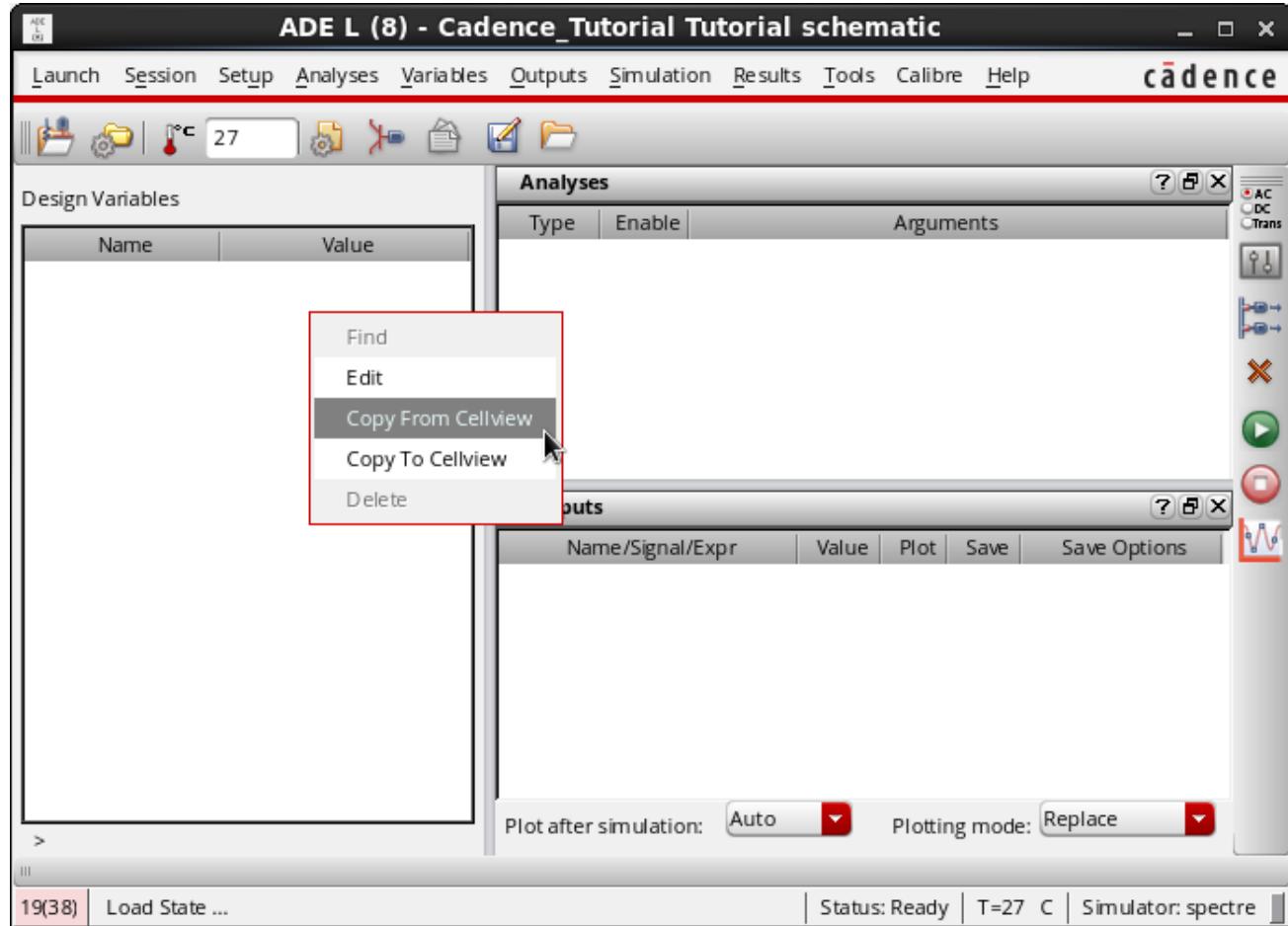


Figure 135: Variables from the Schematic import

Right-clicking in this area opens a context menu. By selecting the menu item *Copy From Cellview*, all variables created in the *Schematic* are imported. By simply clicking on the corresponding field in the *Value* column, values can be assigned to the variables. After importing, the variables *vbs*, *vds* and *vgs* should be listed in the *Design Variables* window. First set the variables to the following numerical values:

- $v_{bs} = 0 \text{ V}$
- $v_{ds} = 3.3 \text{ V}$
- $v_{gs} = 2.3 \text{ V}$

The values set here are the basis for the operating point calculation for a *sweep* in the test.

17.1.2 DC simulation to display the operating point

The first simulation is to carry out a DC analysis, which determines the operating point of the transistor and displays it in the *Schematic*. This simulation can be set up in ADE-L using the *Choose Analyses* menu option in the vertical toolbar on the right edge of the window or via the menu item *Analyses > Choose Analyses*. All adjustable simulation types are shown in *Figure 136*.

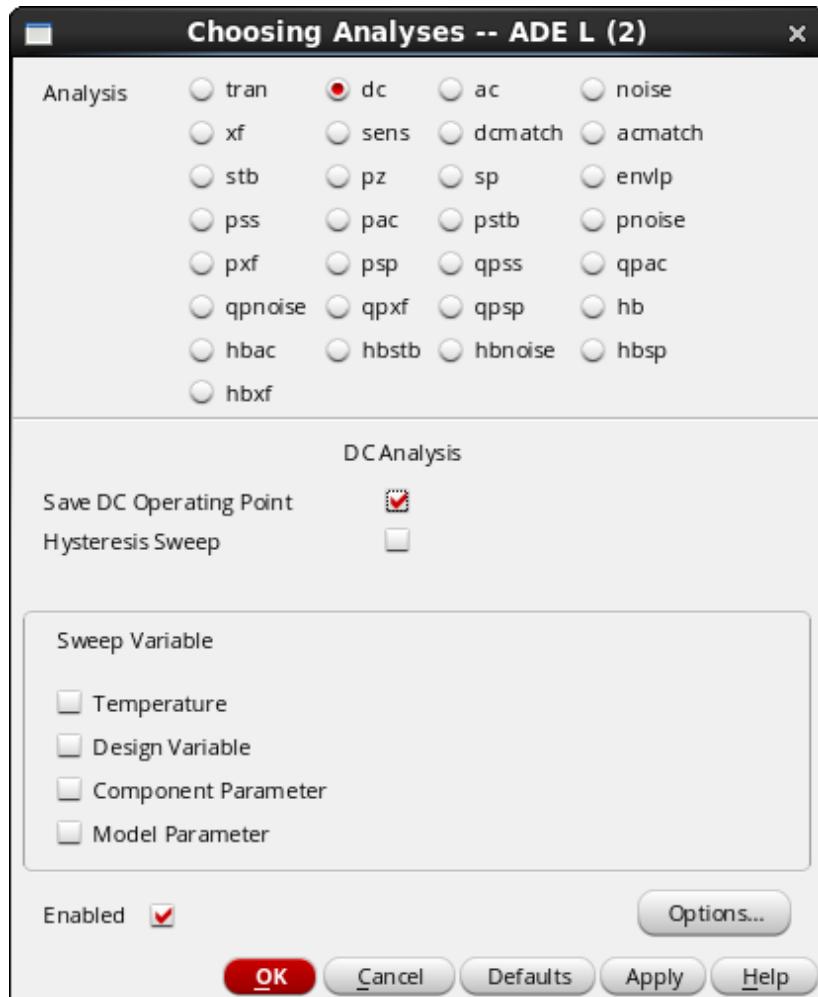


Figure 136: Window for setting up a simulation

A simple DC analysis should first be set up without the use of variables or sweeps. Select the DC analysis in this window if not already done. In the middle area of the window, a check mark must be placed in the *Save DC Operating Point* field so that the operating point can be saved and displayed in the circuit on the transistor. By clicking on *OK*, the configuration of the simulation is completed, which is then displayed in the *Analyses* area of the ADE-L window (see *Figure 137*).

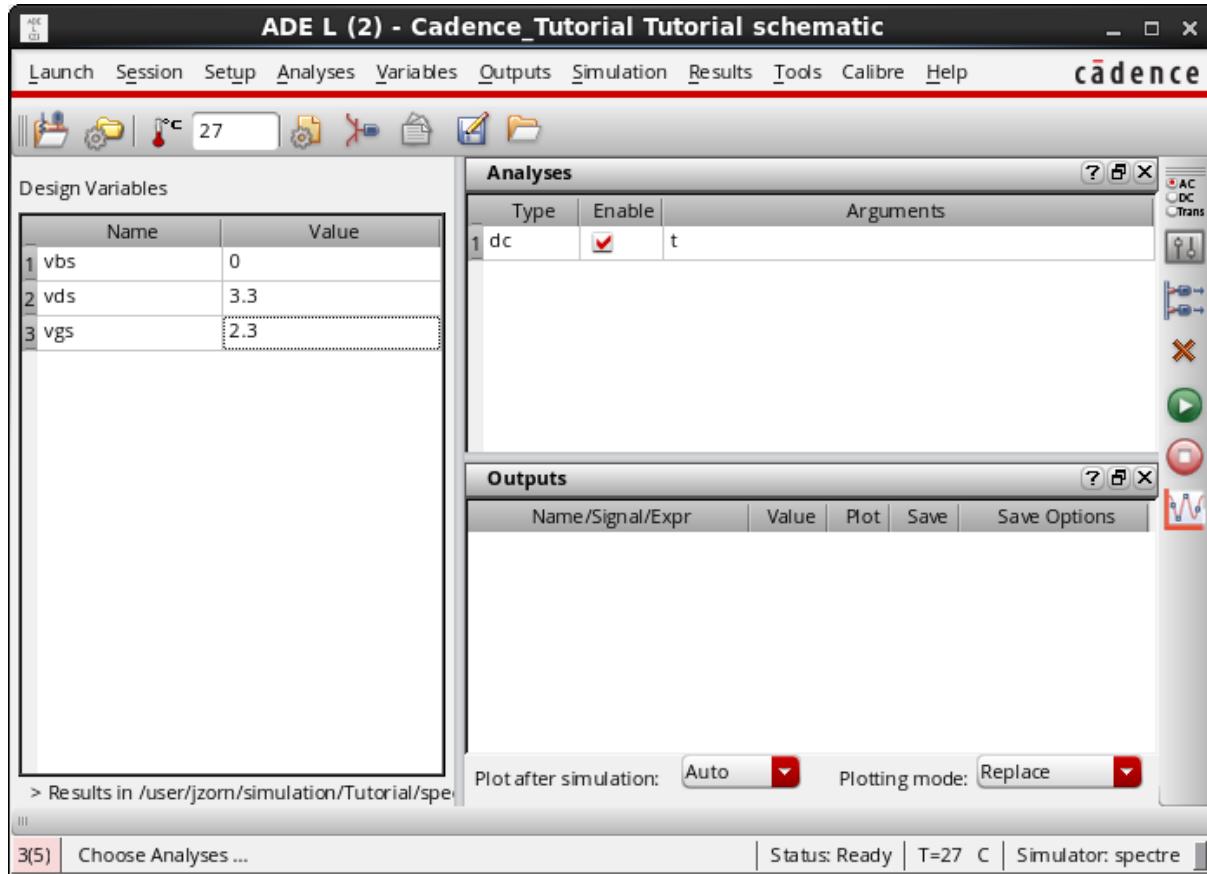


Figure 137: The simulation set up in the ADE-L window

The simulation is started by clicking on the green *Netlist and Run* symbol in the right toolbar or via the menu item *Simulation > Debug Test*.

For this purpose, a netlist is automatically generated from the circuit diagram and passed on to the simulator. During runtime, the simulator generates messages that are logged in a new window. The simulation has run successfully if the message *Convergence Achieved* can be found in the log files.

If the log window does not open, the creation of the netlist may have failed. A common cause of this is failure to perform the *Check and Save* action after editing the schematic. Therefore, always remember to carry out the *Check and Save* action before simulating a modified circuit diagram.

17.1.3 Display of the operating point

After successfully carrying out this DC simulation, it is possible to display all voltages on the circuit lines and the operating point of all components used. The operating point is displayed in the ADE-L window via the menu items *Results > Annotate > DC Operating Points*, or the mains voltages via *Results > Annotate > DC Node Voltages* is displayed (see Figure 138). Another way to activate this display is in the Schematic window under *View > Annotations > DC Operating Points* or *DC Voltages* (see Figure 139 on the next page).

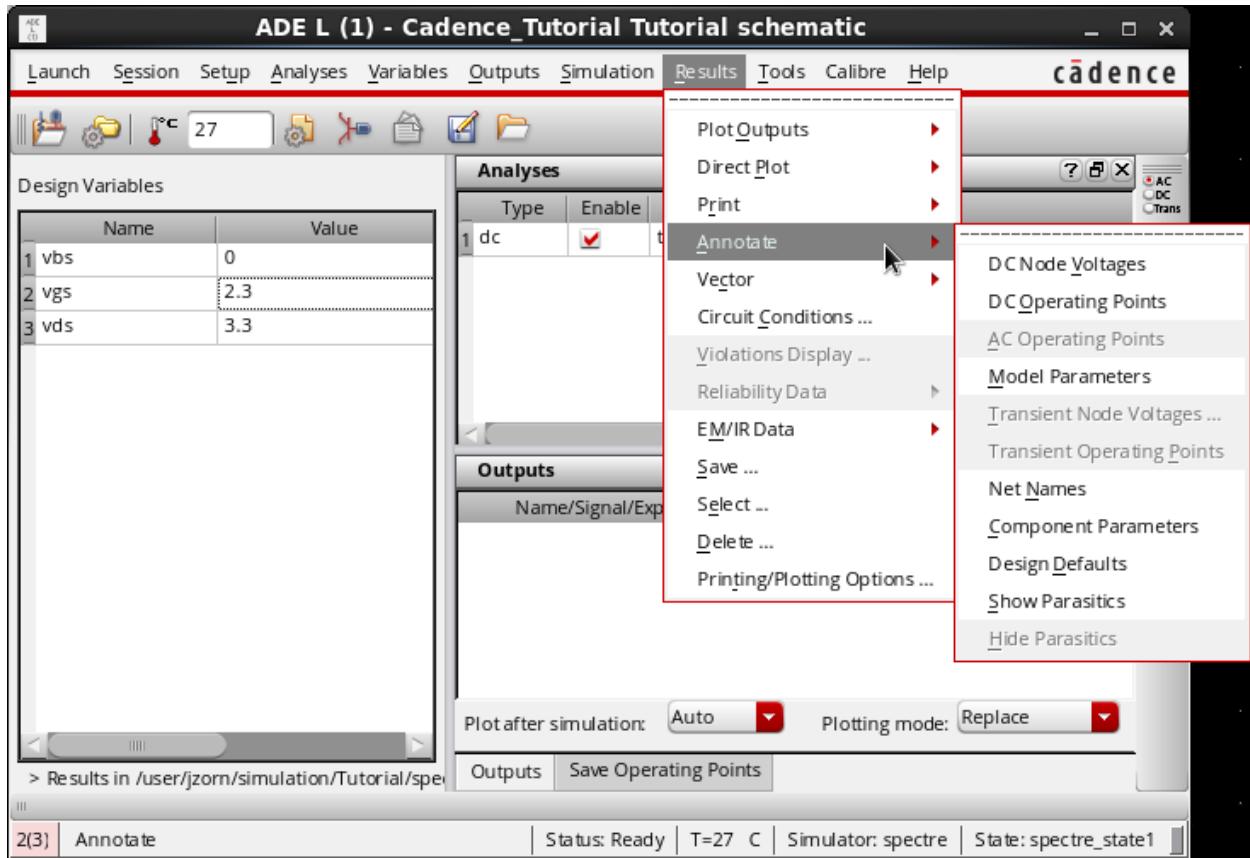


Figure 138: Display options in the ADE-L window

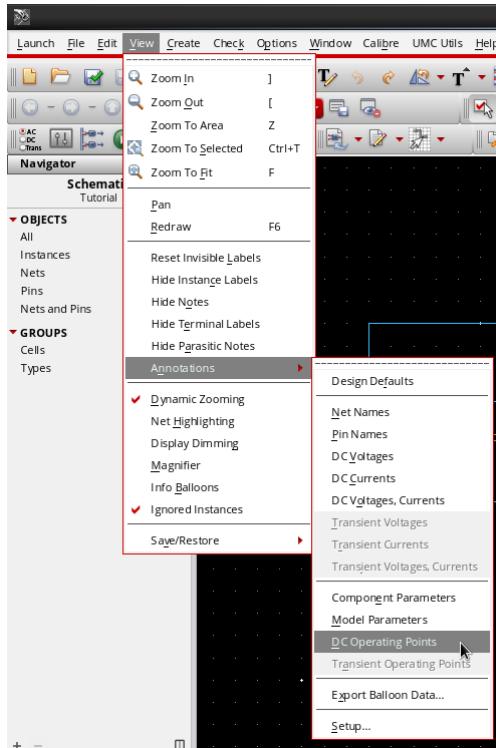


Figure 139: View the DC-Operating Points

Also under the View menu or through the context menu that opens when you right-click on the transistor, you can access the Annotation Setup window via Annotations > Setup.

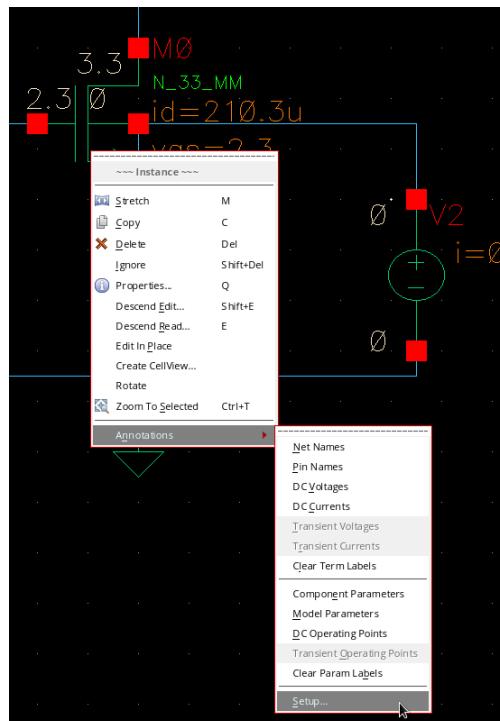


Figure 140: Call up the setup to set the Annotations

In the Annotation Setup window, select the NMOS transistor in the *Cell* area. As can be seen in Figure 141, additional operating point information such as the saturation voltage (*vdsat*) can be selected in one of the free *Expression* fields. If the changes to the operating point display are to be applied not only to the selected transistor, but also to all other transistors of the same type in the circuit, the star symbol can be selected under *Instance* instead of an instance name. The window is then closed by clicking on *OK*.

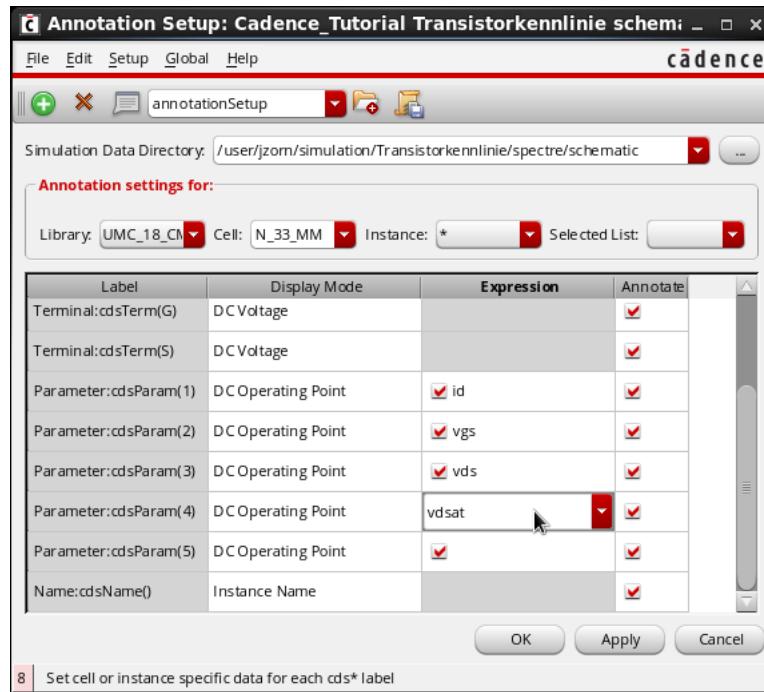


Figure 141: Display the saturation voltage of the transistor

Note: If there are other transistor types in your circuit, such as PMOS transistors, these must also be selected in the *Cell* field and the above steps must be carried out again.

If the star symbol has been selected in the *Instance* popout, after confirming with *OK*, another window (Figure 142) will appear asking whether the Settings should be applied to all selected elements of the same type. By clicking on *Apply * Settings to all* and further confirming with *OK*, the saturation voltage of the transistor should now be displayed next to the component in the *Schematic* (compare Figure 144).

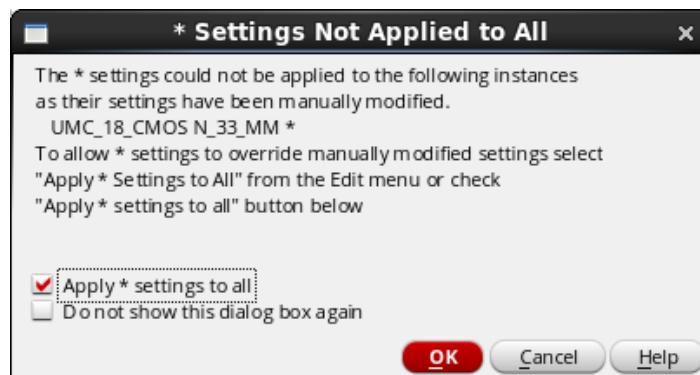


Figure 142: Confirm display settings

17.1.4 Creating simulation outputs from the schematic

Before an analysis is carried out, it is necessary to define so-called *outputs*, which are automatically displayed after the simulation has been carried out. Either voltages on individual lines of the circuit or connections of components (e.g. the drain of a transistor) can be considered. To set up an *Output*, right-click in the *Output* area of the ADE-L main window and select the *Edit* item. Alternatively, select *Outputs > Setup* in the menu bar.

This opens the *Setting Outputs* window, which offers various options for defining *Outputs*. The *Output* to be simulated is specified in the text field *Expression*. This can be the name of a network or a port, as well as a mathematical expression. The use of mathematical expressions with the help of the *Calculator* tool is discussed in detail in the following *Chapter 6.6*. First, an *Output* should be set up via a direct selection in the *Schematic*. To do this, click on the *From Design* button, which is located to the right of the *Expression* text field.

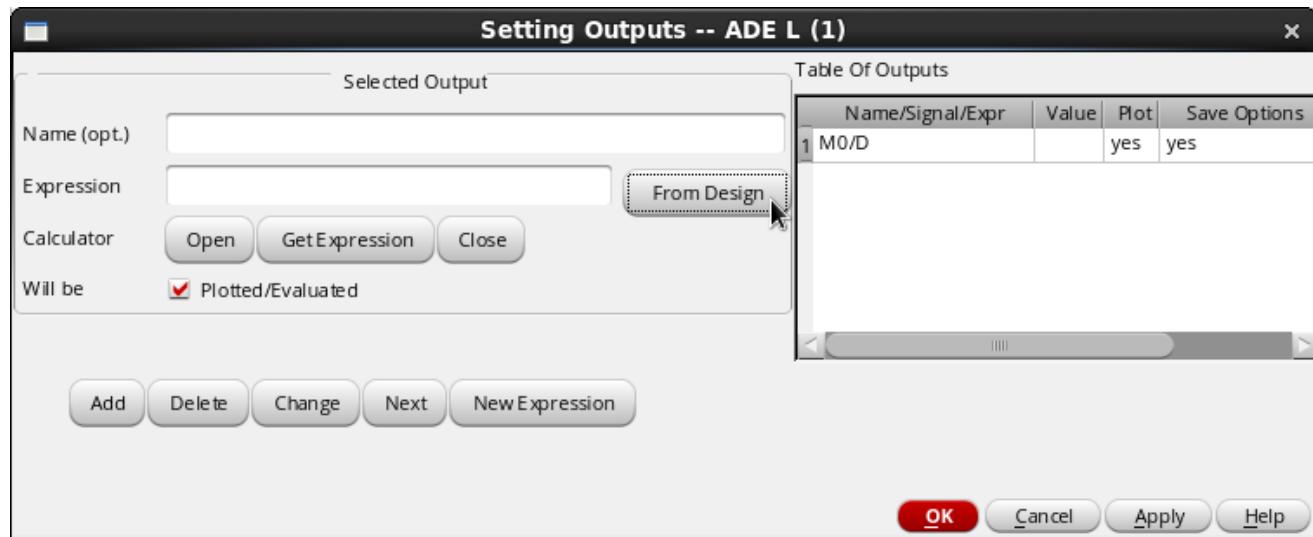


Figure 143: Window for setting up *Outputs*

You now get to the *Schematic* view and click to select a point in the circuit that you want to specifically look at in the simulation. There are various options to choose from. If you click on a net with the mouse, the voltage of this net is as *Output*. If you click on an individual connection contact of a component, the current flow at this point is created as *Output*. By clicking on the component, the current from all connection contacts of this component are selected as outputs.

It is also possible to select several nets, connections and components at once. The required *Outputs* are specified from *Chapter 7*. The drain connection of the transistor was selected here as an example.

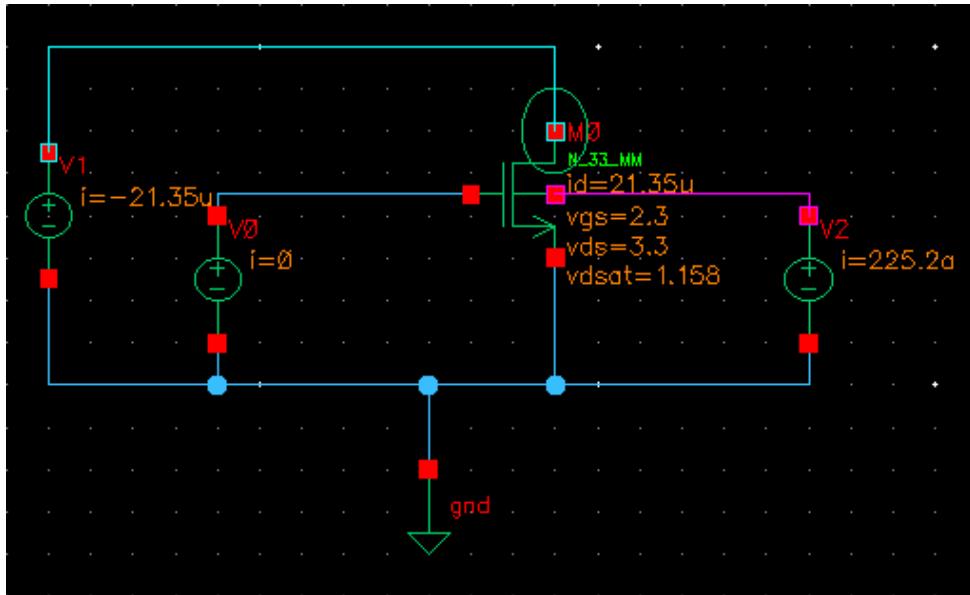


Figure 144: Colored markings inside Schematic mark the Outputs

The selected line or component connection is marked in color and also listed in the *Table of Outputs* area of the *Setting Outputs* window (see Figure 143). By default, the software uses a combination of the instance name of the component and the name of the selected port or line as the name. This expression can be given a meaningful name in the *Name (opt.)* field. By clicking on OK, the *Outputs* are transferred to the ADE-L main window.

17.1.5 Creating outputs using the calculator

Another way to create *Outputs* is the so-called *Calculator*. Figure 145 shows the main view of the *Calculator* tool. This tool can be opened in the ADE-L window under *Tools > Calculator*. Another way to open the *Calculator* is in the *Setting Outputs* window. There you click on the *Open* button in the *Calculator* area. The *Calculator* offers extensive options for numerical analysis and evaluation of the simulation results using special functions and generic mathematical expressions.

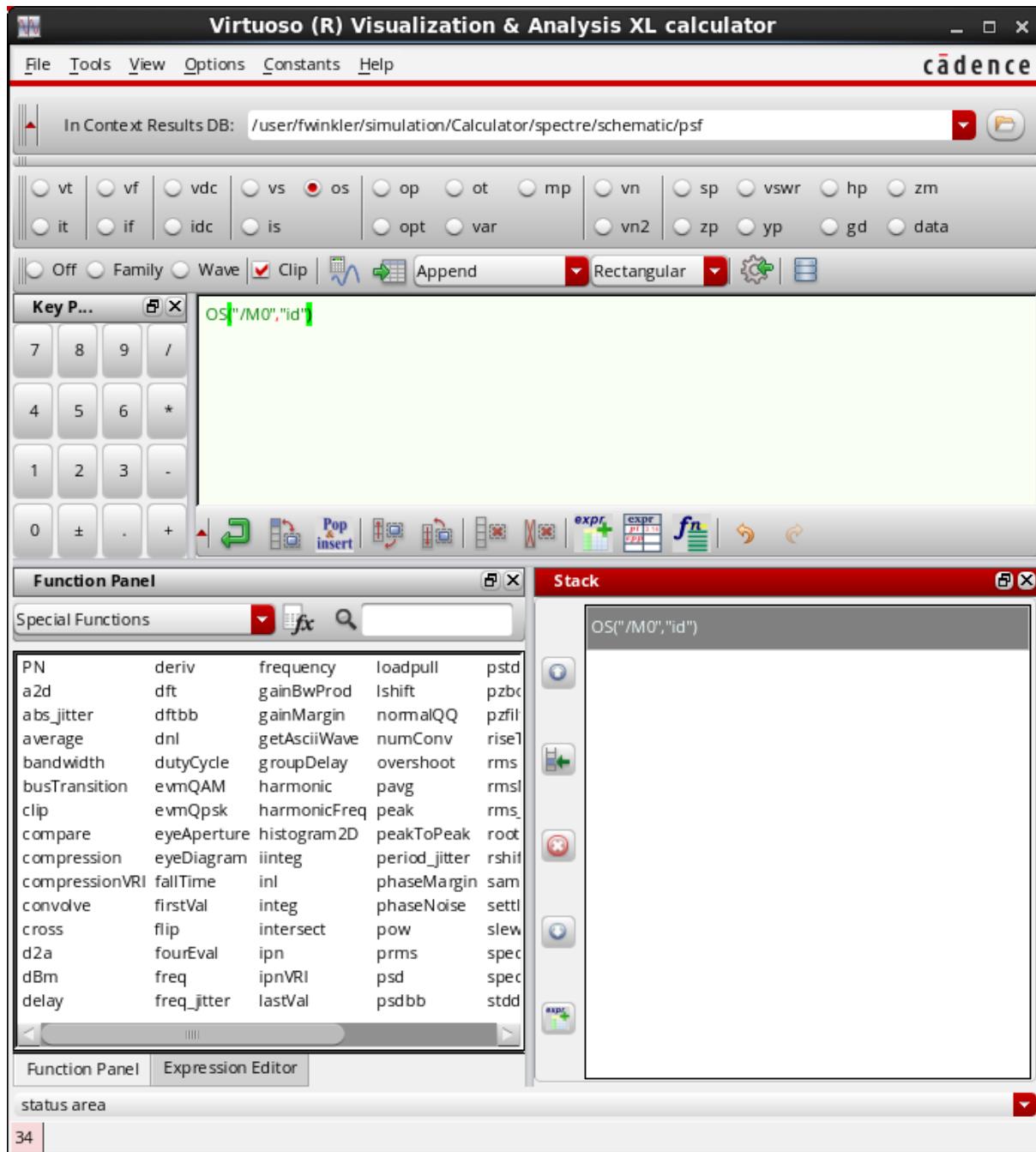


Figure 145: Main view of the Calculators

With the help of the buttons in the upper third of the *Calculator*, expressions for operating points, sizes or even any signals can be created for all common types of simulation (*DC*, *AC*, *Trans*) with just a click select from the *Schematic*. These expressions are sorted from left to right in blocks according to the following scheme: *Trans*, *AC*, *DC*, *Sweeps*, *Operating Points*. For easy understanding, the current flow of the drain connection of the transistor should also be selected as an example, analogous to *Chapter 6.5*.

Clicking on the *os* button opens a window with which an *instance* can be selected from the *Schematic*. Please note that the window shown in *Figure 146* may open in the background and may need to be brought to the foreground by selecting it.

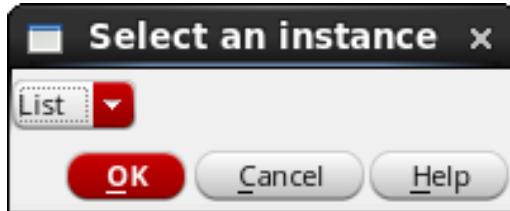


Figure 146: Window for selecting a component

Clicking on the transistor in the circuit diagram opens a selection list with the available operating point sizes of the selected transistor. Select the drain current *id* and press *OK*.

The selected quantity is now displayed as an *expression* in the stack window of the *Calculator* (see *Figure 145*). The expression follows a fixed syntax, which can in principle also be entered manually in the stack area. In this case, the expression *OS(""\M0"", "id")* describes that the parameter *id* of the transistor M0, which corresponds to the drain current, extracted from the results of the operating point simulation shall be.

Note on using the *Calculator* stack: The entry in the input area can be brought to the top of the *Calculator* stack by clicking on the *Enter* button (green arrow at the bottom of the input field). Alternatively, this also works with the **[Enter]** key. Expressions on the stack can also be dragged and dropped back into the input area and linked to other expressions there.

In order to be able to use the expression for the simulation as *Output*, it must be transferred to the ADE-L window. To do this, after creating the *Expression* in the *Calculator*, switch back to the ADE-L window and leave the *Calculator* window still open. In the ADE-L window, open the *Setting Outputs* window (as already described in Chapter 6.6). Using the *Get Expression* button (see Figure 147), the expression that is currently in the input area of the *Calculator* is imported as *Output*. After importing, the *Output* can be named as desired in the *Name (opt.)* field and transferred to the ADE-L main window by clicking on *OK*.

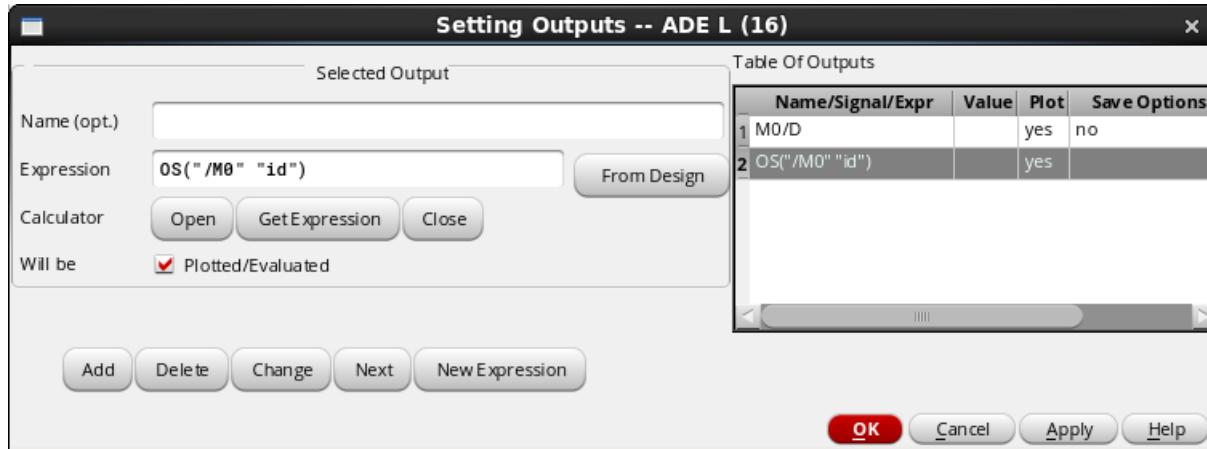


Figure 147: Import from *Calculator-Express* as *Outputs*

Alternatively, the small gear symbol (*Send buffer expression to ADE Outputs*) in the right area of the *Calculator* window can be used to send the expression directly to ADE-L. This can then be edited later by double-clicking.

17.1.6 Saving and loading ADE-L states

In order to be able to restore analyses, variables and *outputs* once created in ADE-L after restarting the software, a so-called simulation *State* must be created or saved in ADE-L. To do this, click on the menu item *Session > Save State* in the ADE-L window. The *Saving State* window opens (Figure 148). In the *Save State Option* area, select the *Cellview* field. The correct *Library*, *Cell* and a *State name* should now already be entered under the *Cellview Options*. By clicking on *OK* the *State* is saved. If necessary, this can then be loaded into ADE-L via the menu item *Session > Load State*. If you do not want to overwrite an existing *State*, you must enter an alternative name in the *State* field.

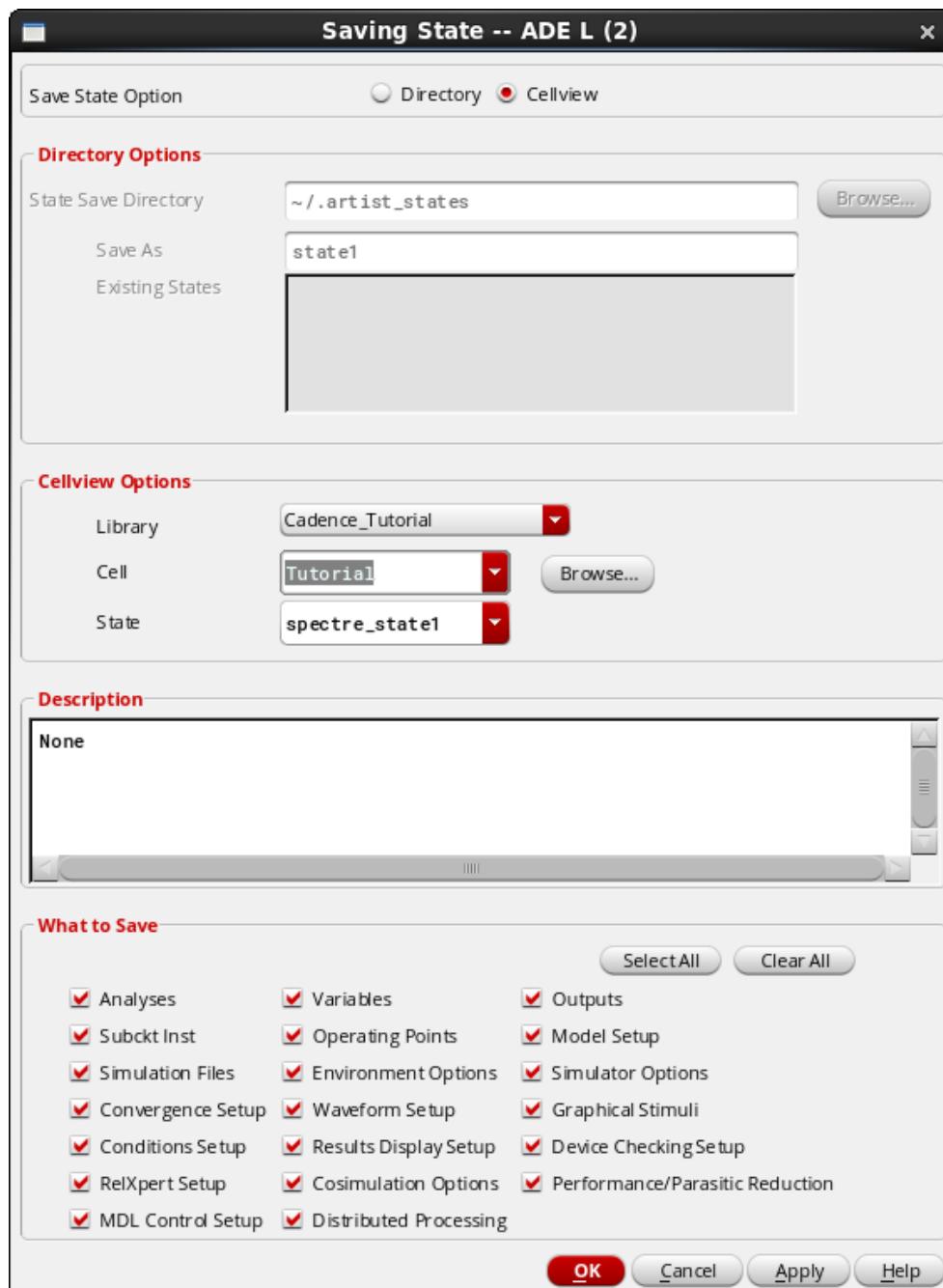


Figure 148: Storing ADE-L States

17.2 DC analysis in ADE-L

17.2.1 Gate-source voltage sweep

DC simulation offers the possibility of determining operating points as a function of a variable parameter. Such simulation series are called *Sweep* in Cadence. The temperature, design variables, component parameters and individual parameters of a simulation model can be changed. To do this, open the *Choosing Analyses* window in the *Analyses > Choose* area of the ADE-L window by double-clicking on the existing DC simulation to calculate the operating point.

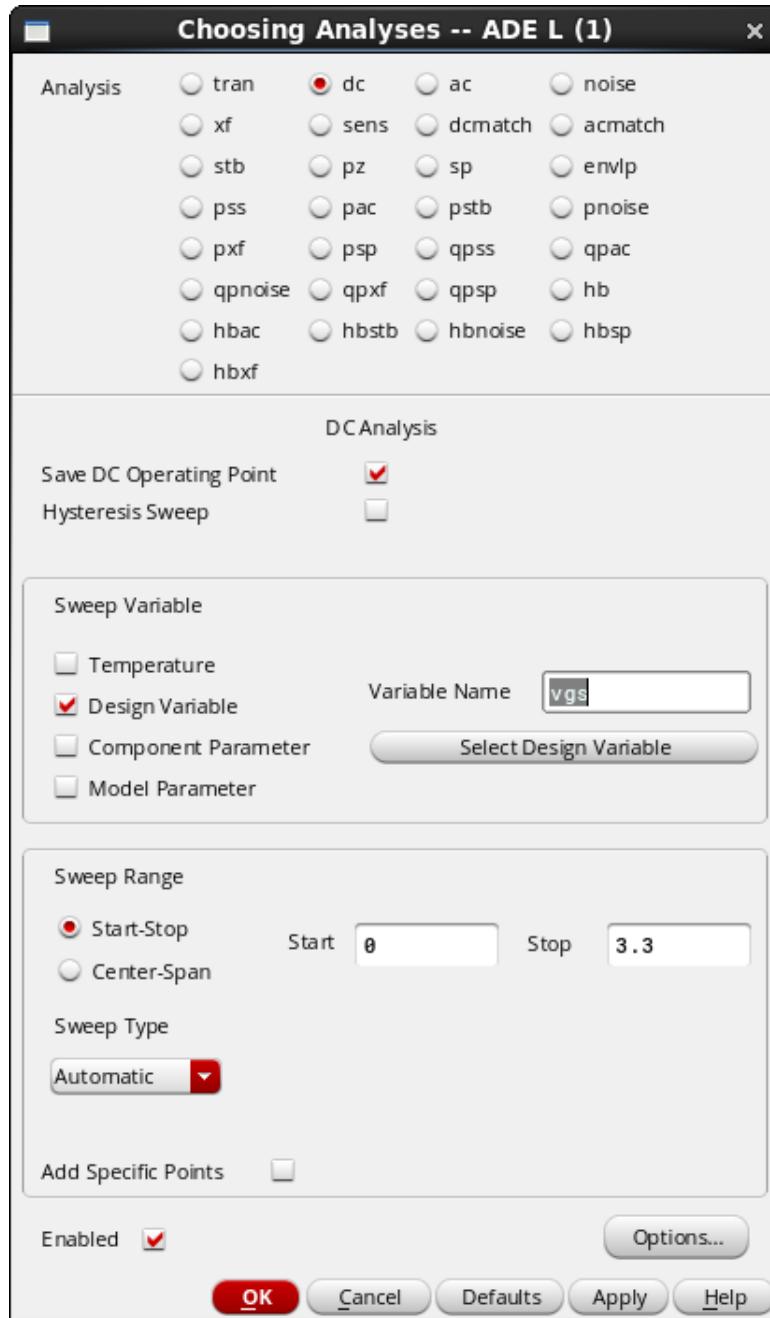


Figure 149: Simulation selection window

In the *Sweep Variable* area, a variable is selected that should be passed through. In addition to a fixed variable, such as the temperature, this can also be a variable created in the *Schematic (Design Variable)*, or a parameter of a component (*Component Parameter*). In the lowest window section called *Sweep Range*, the interval to be run through as well as the type of *Sweep (Sweep Type)* can be specified. You can choose from *Automatic*, *Linear* and *Logarithmic*.

The following simulation corresponds to a DC analysis with a sweep of the gate-source voltage of the NMOS transistor. Select the checkbox *Design Variable*. The previously introduced variable *vgs* is used for this *sweep*. This can either be entered directly or selected by clicking on *Select Design Variable* in the window that opens (Figure 150).



Figure 150: Selection of the created variables

A *Sweep* of type *Automatic* should be carried out from 0 V to 3.3 V. Enter these values in the *Sweep Range* area in the *Start* and *Stop* fields. The values of the variables for the drain-source voltage (*vds*) and the source-bulk voltage (*vbs*) initially remain constant at the values previously set in the ADE-L window section *Design Variables* defined values. The variable *vgs* is varied as a parameter during the *sweep*, but the originally set value in the ADE-L window of 2.3 V decides which gate-source voltage the operating point is for of the transistor is saved.

In this analysis, the current flow in the drain of transistor M0 should be defined as *Output*. How an *Output* is created from the *Schematic* is described in *Chapter 6.5*. If the creation was successful, it should appear as M0/D in the *Output* area (see Figure 151).

After selecting the *Outputs*, the simulation can now be carried out. To do this, all possible changes must first be saved in the *Schematic* using *Check and Save*. The simulation can then be started by clicking on the green *Run Simulation* button in the menu bar on the right edge of the window.

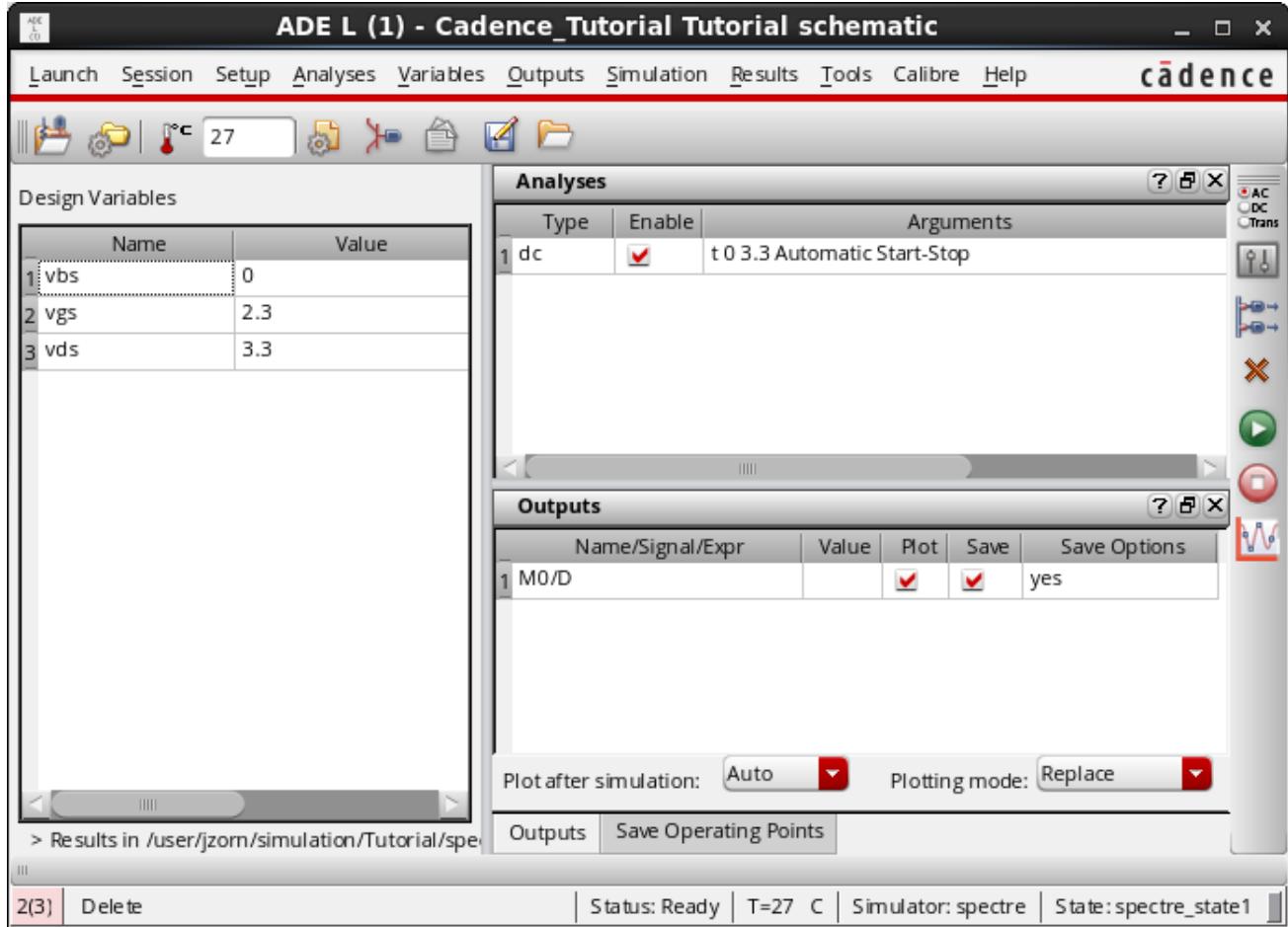


Figure 151: The ADE-L window set up for analysis

The *Output* shown in *Figure 151* should be both saved and displayed graphically. To do this, both checkboxes (*Save* and *Plot*) are selected in the *Outputs* area of the ADE-L window. *Figure 152* shows the dependence of the drain current on the gate-source voltage.

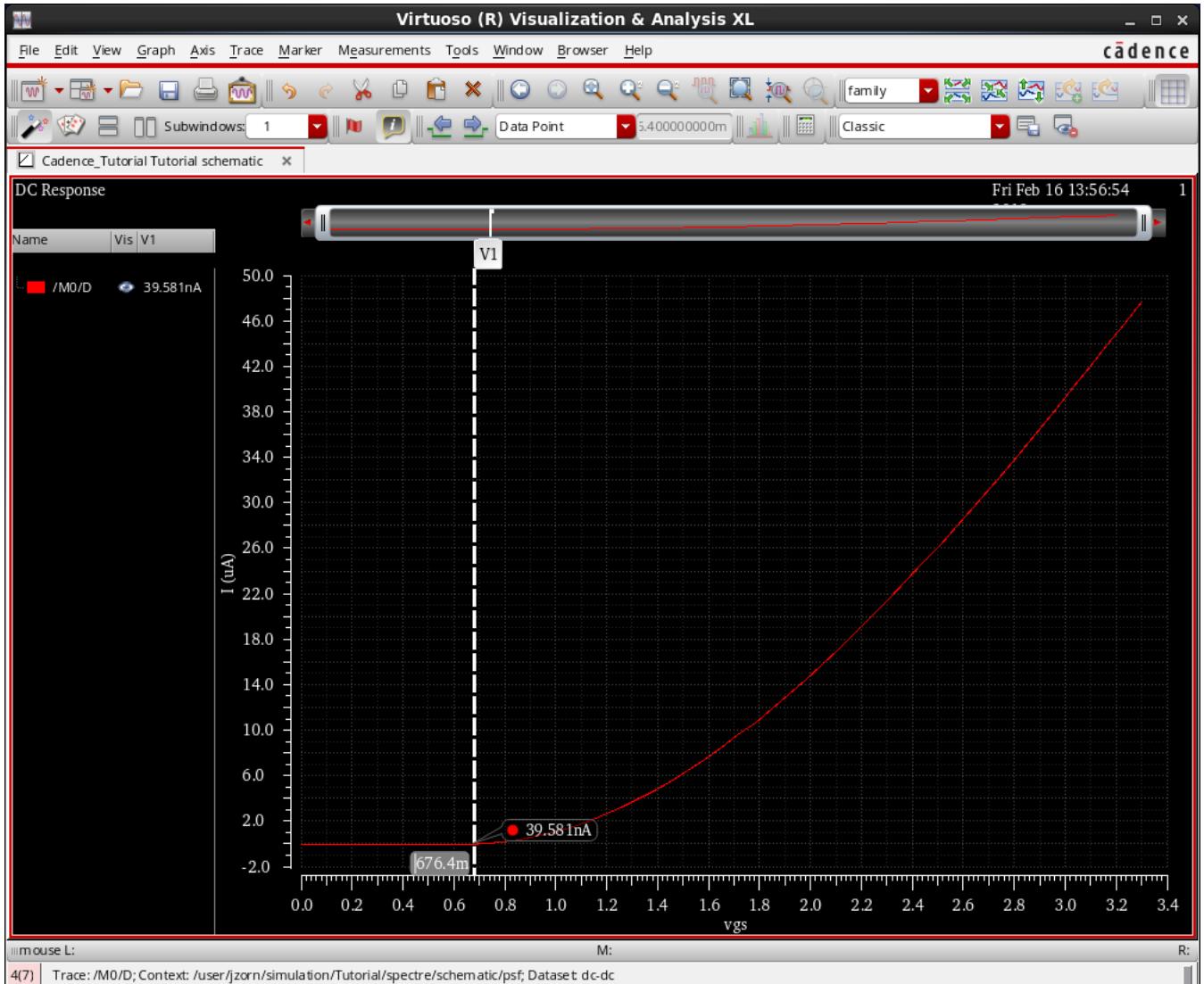


Figure 152: DC simulation with Sweep the gate-source voltage v_{gs}

The characteristic corresponds to the theoretically known relationship between channel current and gate-source voltage. A larger gate-source voltage v_{gs} leads to a larger current flow through the transistor. However, a significant current flow only occurs as soon as the gate-source voltage has exceeded the threshold voltage of the transistor. Mathematically, the current in saturation is described by the following relationship:

$$I_{DSat} = \frac{1}{2} \cdot C_{OX} \cdot \mu_n \cdot \frac{W}{L} (V_{GS} - V_{th})^2 \cdot (1 + \lambda \cdot V_{DS}) \quad (4)$$

17.2.2 Variation of transistor width

Next, the influence of the transistor width on the drain current will be examined. For this purpose, in addition to varying the gate-source voltage via a sweep of the variable v_{gs} , the transistor width should also be changed within a specified value range. Since the DC simulation only allows sweeps over a single size, the *Parametric Analysis* of the ADE-L simulation environment should be used for this simulation, which allows variations of several parameters.

To allow the width of the transistor to be changed parametrically, a variable W is introduced which is entered into the component properties of the instantiated transistor. In this case, the variable should act as a factor of the minimum width. In the transistor properties, $240\text{ nm} \times W$ must therefore be entered in the *Total Width* and *Finger Width* fields. After clicking on *Check and Save*, import the variable into the ADE-L window and set the value to 1.

Open the configuration window for the *Parametric Analysis* shown in *Figure 153* under the menu item *Tools* of the ADE-L menu bar.

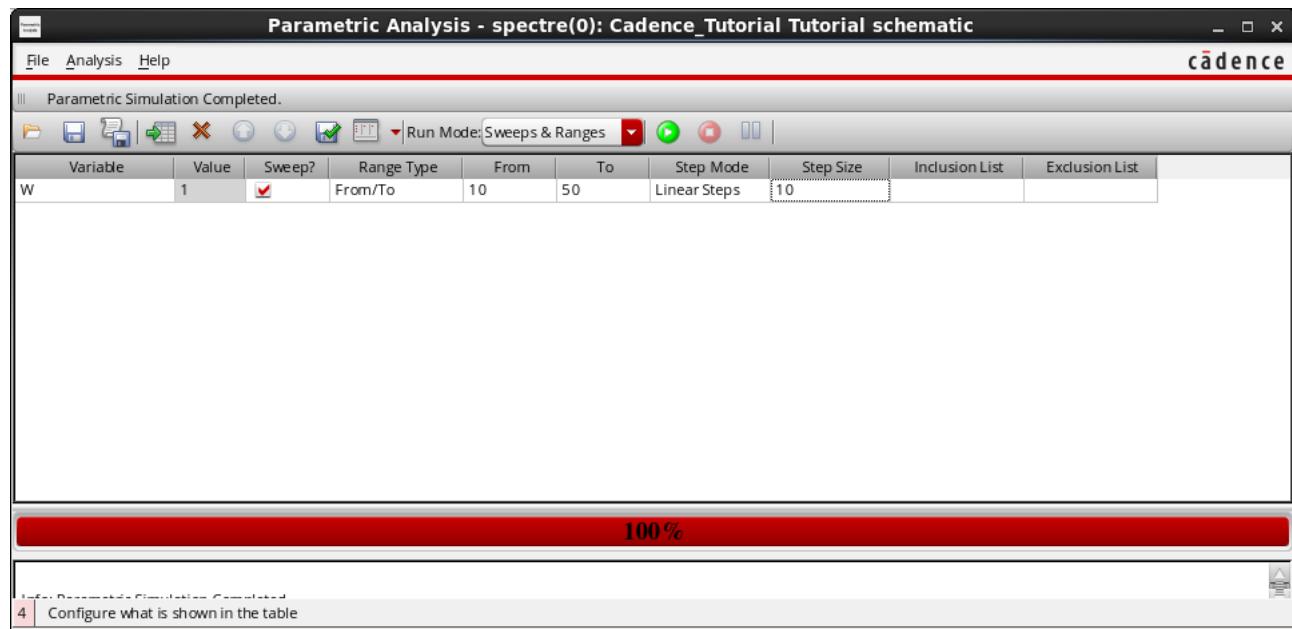


Figure 153: Window for setting up a parametric analysis

The parameter for the transistor width W is inserted as a variable in this window. The variable should take on the values between 10 and 50 in five steps with a step size of 10. For this purpose, the option *From/To* is selected in the *Range Type* field. The value 10 is entered in *From*, the value 50 in *To* and the value 10 in *Step Size*. Select *Linear Steps* as *Step Mode*. Make sure that the checkbox in the *Sweep* column is selected. Then click on the green *Run Selected Sweep* button to start the simulation.

The result of the simulation should be the curves shown in *Figure 154*. According to *Formula (1)* from *Chapter 7.1* we expect an increasing drain current for larger transistor.

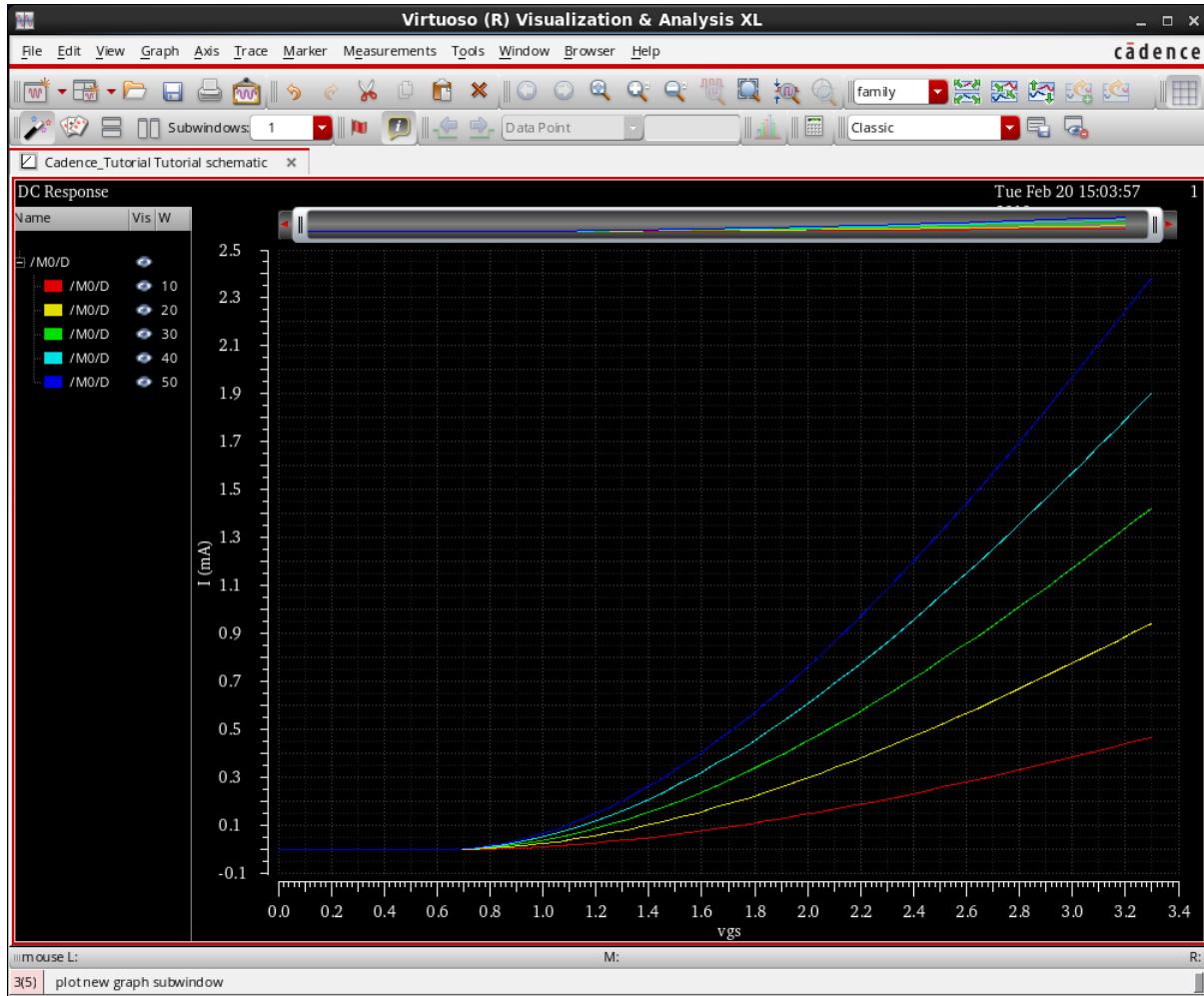


Figure 154: Drain current with variable gate-source voltage and variable transistor widths

17.2.3 Variation of channel length

Instead of the transistor width, the transistor length should now be varied in the next simulation. Based on *Formula (1)* it can be expected that the drain current decreases with increasing channel length L . The parametric simulation of the transistor channel length is set up analogously to the procedure for varying the transistor width. In the transistor properties, the factor L is entered in the *Length* field as a multiplier of the minimum length ($340\text{ nm} \times L$). After clicking on *Check and Save*, the variable is imported into ADE-L and set to the default value 1.

In the next step, a new linear sweep is created in the *Parametric Analyzes* window by clicking on the *Add New Row* button from 1 to 9 with a step size of 2. The variation of the transistor width W is deactivated by deselecting the check mark in the *Sweep* field. The transistor width W thereby receives the standard value ($W = 1$) defined in the ADE-L window. The simulation can then be started by clicking on the green button in the *Parametric Analysis* window. *Figure 155* shows the curves for channel length factors L from 1 to 9.

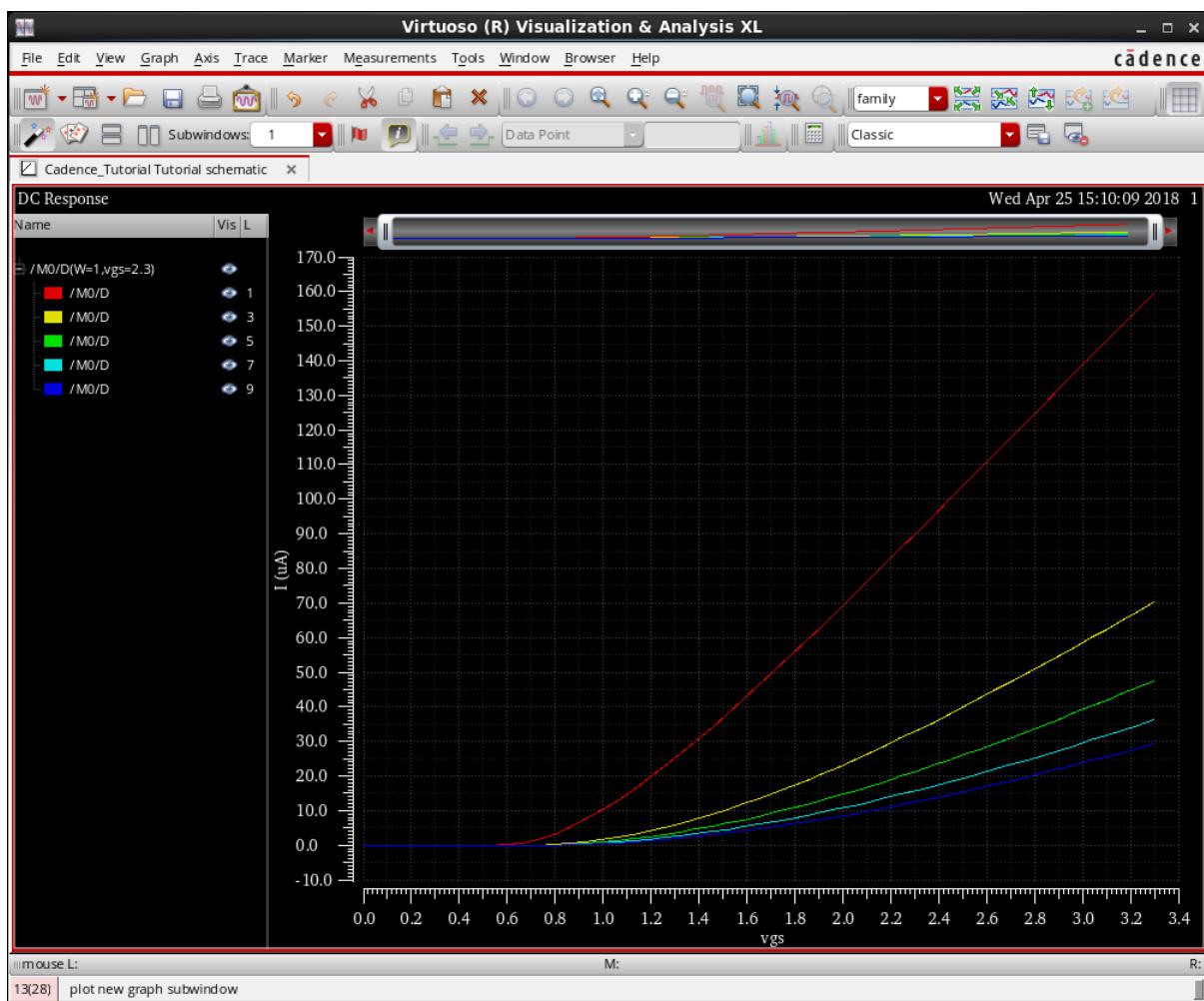


Figure 155: Drain current with variable gate-source voltage and variable transistor lengths

As expected, the channel current drops as the transistor length increases.

17.2.4 Sweep of the drain-source voltage

Another interesting simulation that provides an insight into how a MOS transistor works is the so-called output characteristic curve. Here, the drain current is represented as a function of the drain-source voltage vds . As the nature of the output characteristic curve is also influenced by the gate-source voltage vgs , this voltage should also be varied as a coulter parameter. To do this, a *sweep* is first set up for vds in the DC simulation in ADE-L. Double-click on the DC simulation in the ADE-L window to open the *Choosing Analysis* window. In the *Sweep Variable* area, the variable vgs is replaced by the variable vds . As before, the value interval should be between 0 V and 3.3 V and *Automatic* should continue to be used as *Sweep Type*.

In the *Parametric Analysis* window, a click on the *Add New Row* button inserts an additional *Sweep* for the variable vgs and deactivates the existing *Sweeps* for W and L (see *Figure 156*).

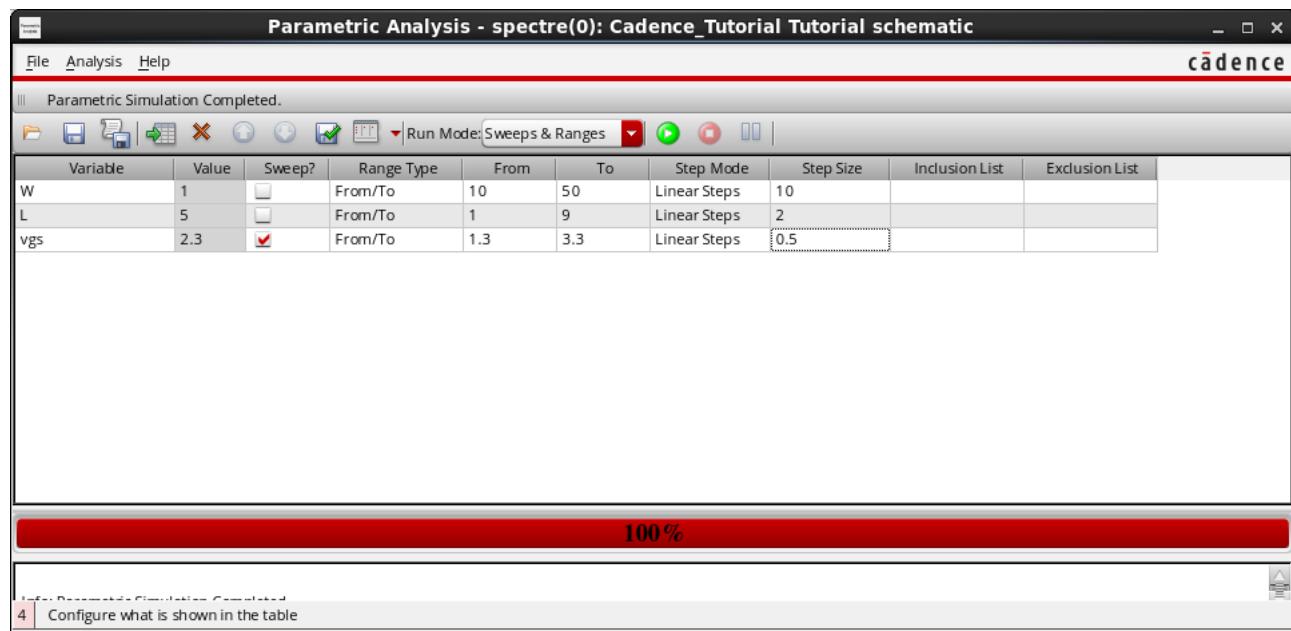


Figure 156: Parametric analysis with variable gate-source voltages

The gate-source voltage should be run through in the interval from 1.3 V to 3.3 V with a *Step Size* of 0.5 V. It does not make sense to start the simulation at a gate-source voltage of 0 V, as in this case there is only little current flow and only a horizontal straight line results.

When you run the parameterized simulation, five curves are generated that correspond to DC sweeps over the defined drain-source voltage range with different gate-source voltages as coulter parameters. According to Formula (1), we also expect a higher current through the transistor for increasing gate-source voltages. Figure 157 shows the five simulated curves.

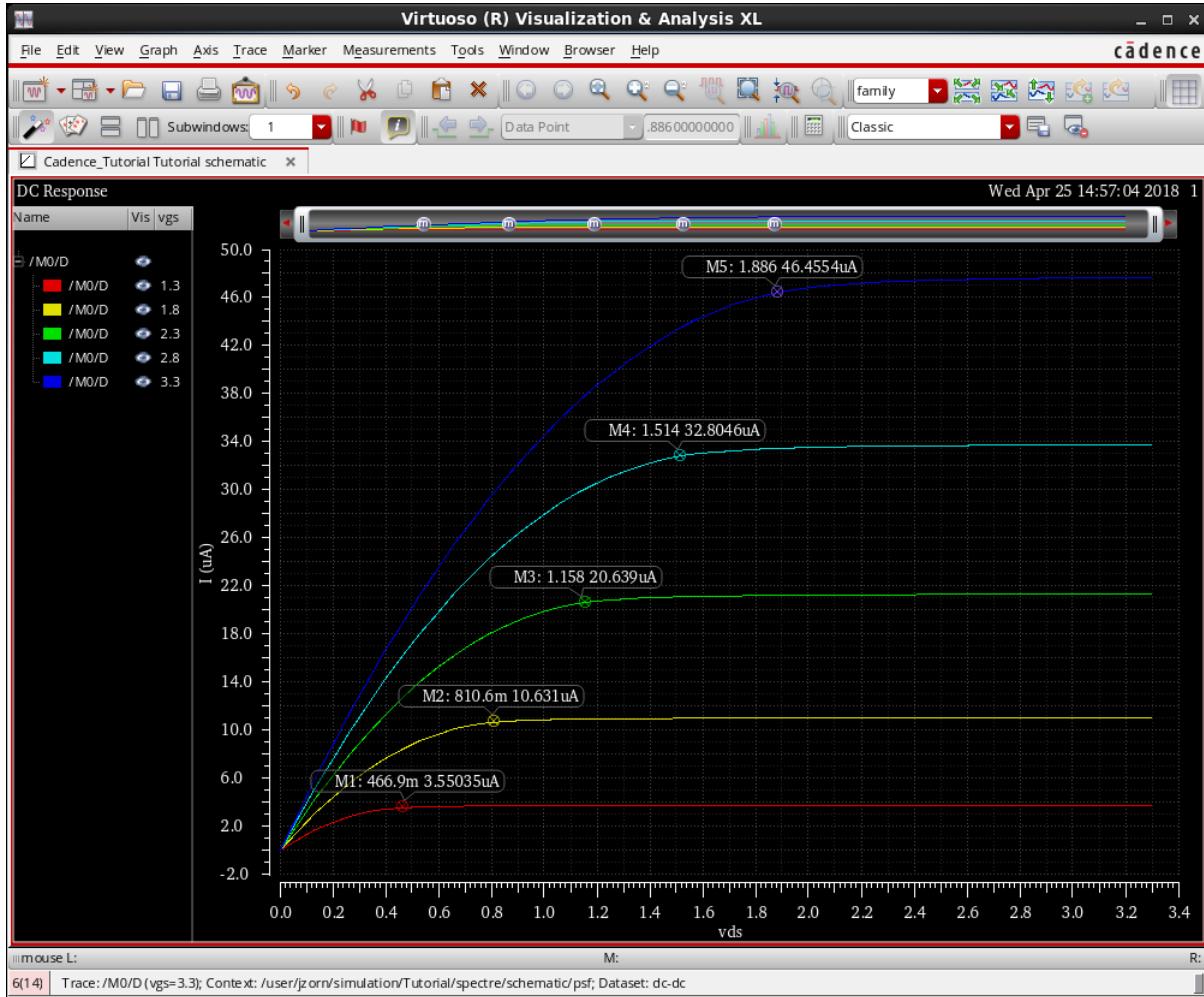


Figure 157: Characteristic curve of drain-source voltages with variable gate-source voltages

The simulation shows that the saturation voltage shifts to higher values for increasing gate-source voltages. The saturation voltage v_{dsat} defines the transition of the current curve from a downward-opening parabola to a constant. It can be calculated using the following formula

$$V_{DSAT} = V_{GS} - V_{TH} \quad (5)$$

17.2.5 Variation of channel width and channel length of the transistor

As a final simulation to analyse the influence of the transistor geometry on the current characteristic, the channel width W and channel length L are now to be varied simultaneously at a constant W/L ratio. To do this, the factor K is used in the transistor properties (Figure 158) as a multiplier of the minimum width and minimum length.

To do this, insert the expression $240 \text{ nm} \times K$ into the field *Total Width* and *Finger Width* and the expression $340 \text{ nm} \times K$ into the field *Length*. In ADE-L, set the value for $K = 5$.

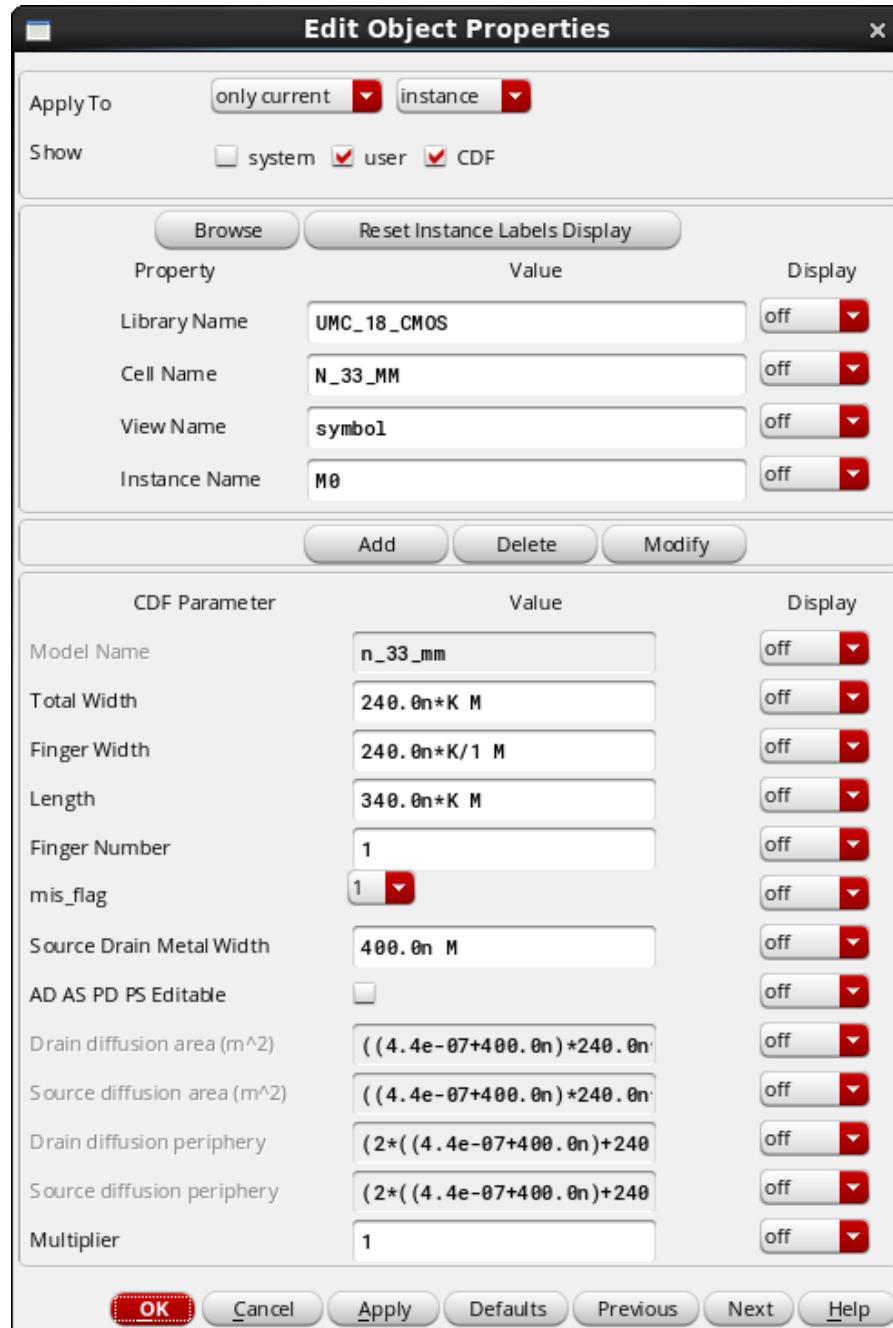


Figure 158: Multiplication of the transistor minimum dimensions

After you have performed a *Check and Save*, deselect the *Sweep* for the channel length L in the *Parametric Analyses* window if you have not already done so, the channel width W and the voltage v_{gs} and add a new *Sweep* for the factor K instead, from 1 to 10 and a step size of 1.

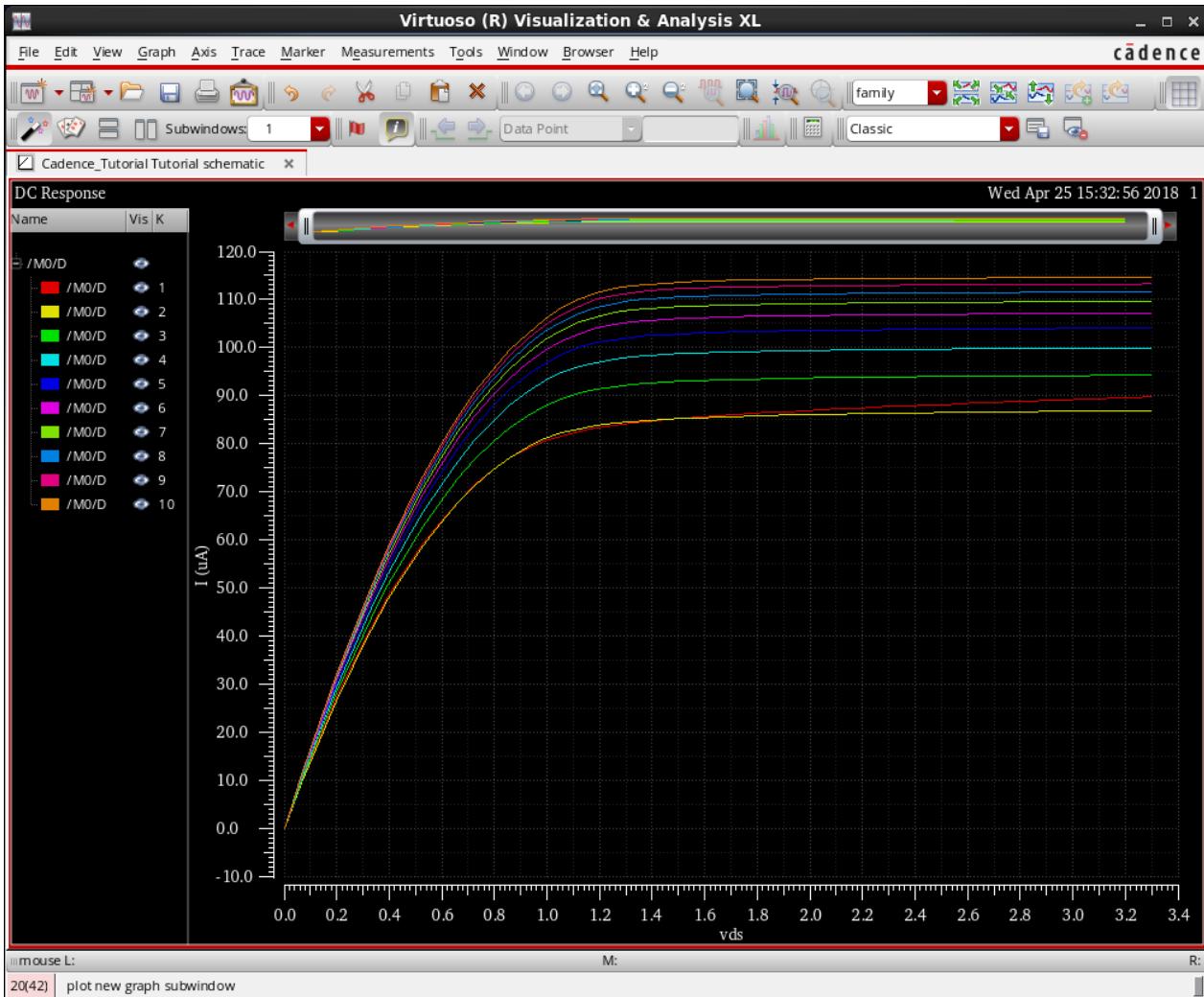


Figure 159: Characteristic curves with increased transistor widths and channel lengths

According to *Formula (1)*, a simultaneous change in the transistor width and length should have no influence on the transistor current at a constant W/L ratio. However, the simulation result shows that slightly higher currents flow for higher values of K . This is due to the fact that the threshold voltage is dependent on the channel length. Transistors with longer channels have lower threshold voltages than short-channel transistors, which in turn also leads to higher currents at a constant voltage v_{gs} .

Furthermore, it can be seen that the slope of the curves in the saturated range decreases with increasing factor K . The dependence of the channel current on the drain-source voltage v_{ds} in the saturated range is related to the effect of channel length modulation. The shorter the channel, the more the current increases with increasing drain-source voltage. For this reason, transistors with a minimum channel length are very rarely used in analogue circuit design. Transistors with channels that are three to five times longer than the technology minimum are usually used.

17.2.6 Consideration of the threshold voltage and the body effect

As a further parameter, the threshold voltage v_{th} of the transistor should now be considered in more detail. The threshold voltage can be displayed for the selected operating point in the circuit diagram or also shown graphically as a function of the temperature or other variables.

In this example, the value of the threshold voltage at the set operating point is displayed first. The bulk or body effect is then triggered by varying the bulk source voltage via the variable v_{bs} .

To set up the analysis, the two *Calculator* functions OP and OS are important so that the *Operating Point* (OP) and an *Operating Point Sweep* (OS) can be displayed.

The threshold voltage at the operating point is stored as a simulation result in the *DC operating point* and can be read out using the OP function. Select the threshold voltage v_{th} (see Figure 160 (a)) and press OK. Transfer the expression to the ADE-L window. To be able to display the threshold voltage even when a DC sweep is running, repeat the above steps, this time using the OS function (Figure 160 (b)). Give the expressions meaningful names.

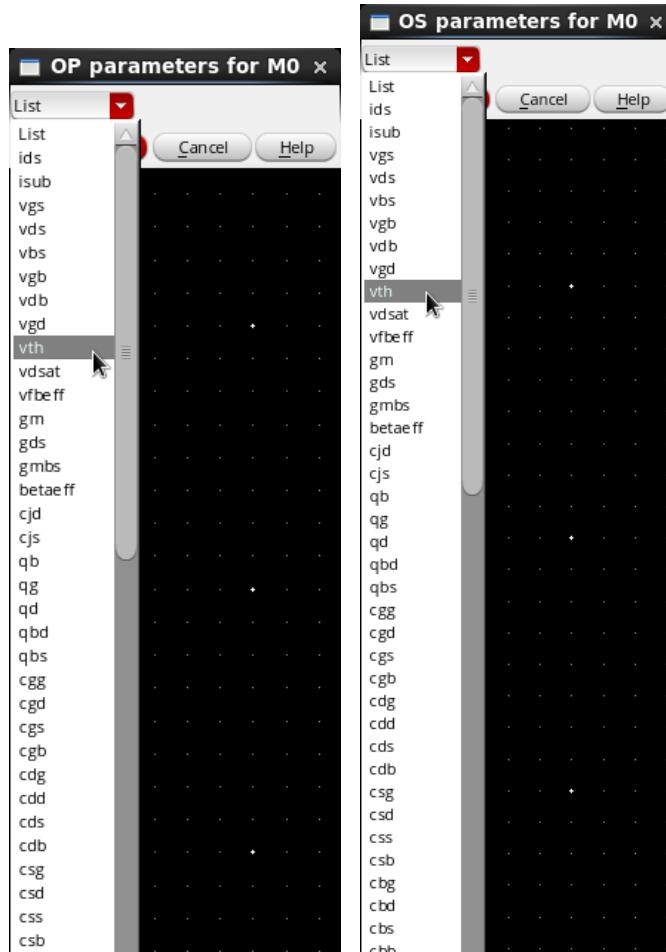


Figure 160: Selection of the load carrier via the operating point list (a) OP and (b) OS

Now activate only the OP expression (or v_{th_wert}) for the threshold voltage in the *Outputs* in the ADE-L window and start a DC simulation by clicking on the green button. After running the simulation, the value of the

threshold voltage under nominal conditions, i.e. at room temperature ($T = 27^\circ\text{C}$), now appears in the *Outputs* area of the ADE-L window in the *Value* column.

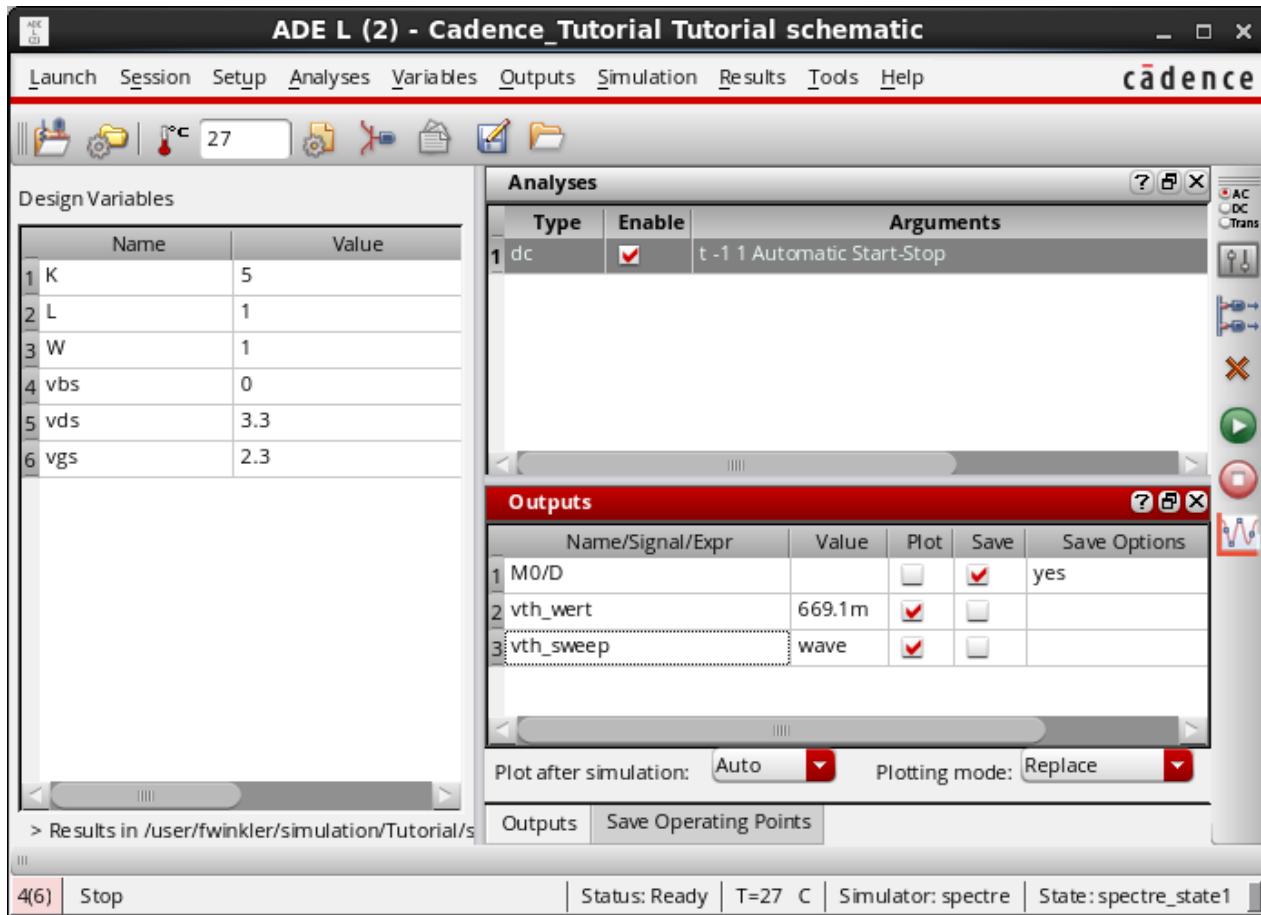


Figure 161: ADE-L main window with calculated threshold voltage

Next, the influence of the body effect on the threshold voltage is to be simulated and displayed graphically. The body effect describes a change in the threshold voltage depending on the voltage between the bulk and source. The bulk-source voltage is introduced into the circuit by a voltage source whose voltage is defined by the variable *vbs*. The bulk-source voltage is to be varied as a DC sweep in a range from -1 V to +1 V.

To do this, open the DC simulation and configure an automatic Sweep for the variable *vbs* in the corresponding value range. From theory, a decreasing threshold voltage is expected for an increasing bulk source voltage. Now also activate the expression OS (or *vth_sweep*) for the threshold voltage in the *Outputs* of the ADE-L window and carry out the simulation.

The simulation result should correspond to the curve shown in *Figure 162*. Place a vertical line at a value of 0 V for the voltage v_{bs} and compare your values. You can find out how to set a marker in *chapter 8.1*.

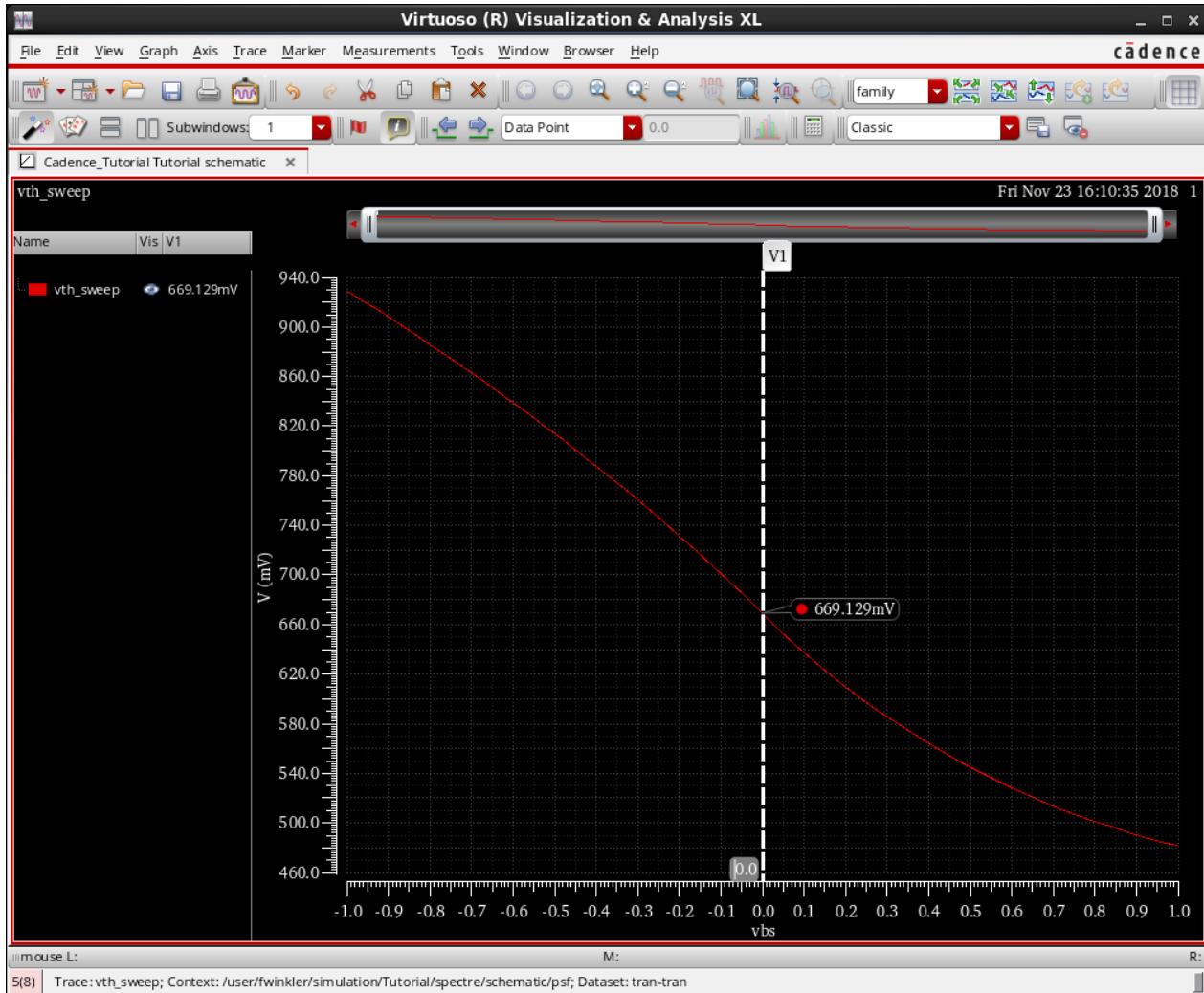


Figure 162: Graphic representation of the body effect with swept source-bulk voltage

17.2.7 Temperature influence on threshold voltage and charge carrier mobility

As the value of the threshold voltage is also temperature-dependent, this relationship should now be simulated and displayed graphically. In addition, the temperature dependence of the charge carrier mobility μ is also to be investigated. To do this, the temperature in the *Choosing Analyses* window is to be varied as DC-Sweep. Then define a sweep interval in the range *Sweep Range* from -40 °C to 120 °C using the *Sweep Type Automatic*.

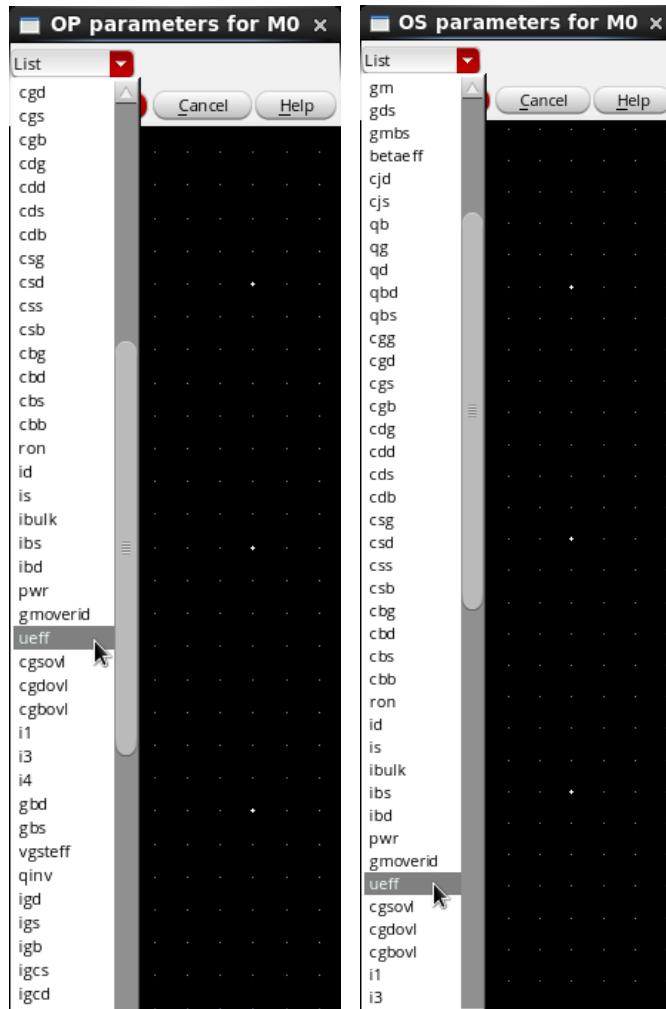


Figure 163: Selection of the load carrier via the operating point list (a) OP and (b) OS

Since the expressions for the threshold voltage variation have already been introduced as *Outputs*, only the corresponding expressions for the charge carrier mobility need to be added. To do this, open the *Calculator* and first click on *OP* and then on the transistor in *Schematic*. Then select the size *ueff* from the list shown in Figure 163. Repeat this step by first selecting the function *OS* in *Calculator*. Insert these two expressions into the *Outputs* of ADE-L and give them a meaningful name.

Now run the DC simulation by clicking on the green button in the ADE-L window. After running the simulation, the graphs of the charge carrier mobility and the threshold voltage should be displayed as a function of temperature as shown in *Figure 164*.

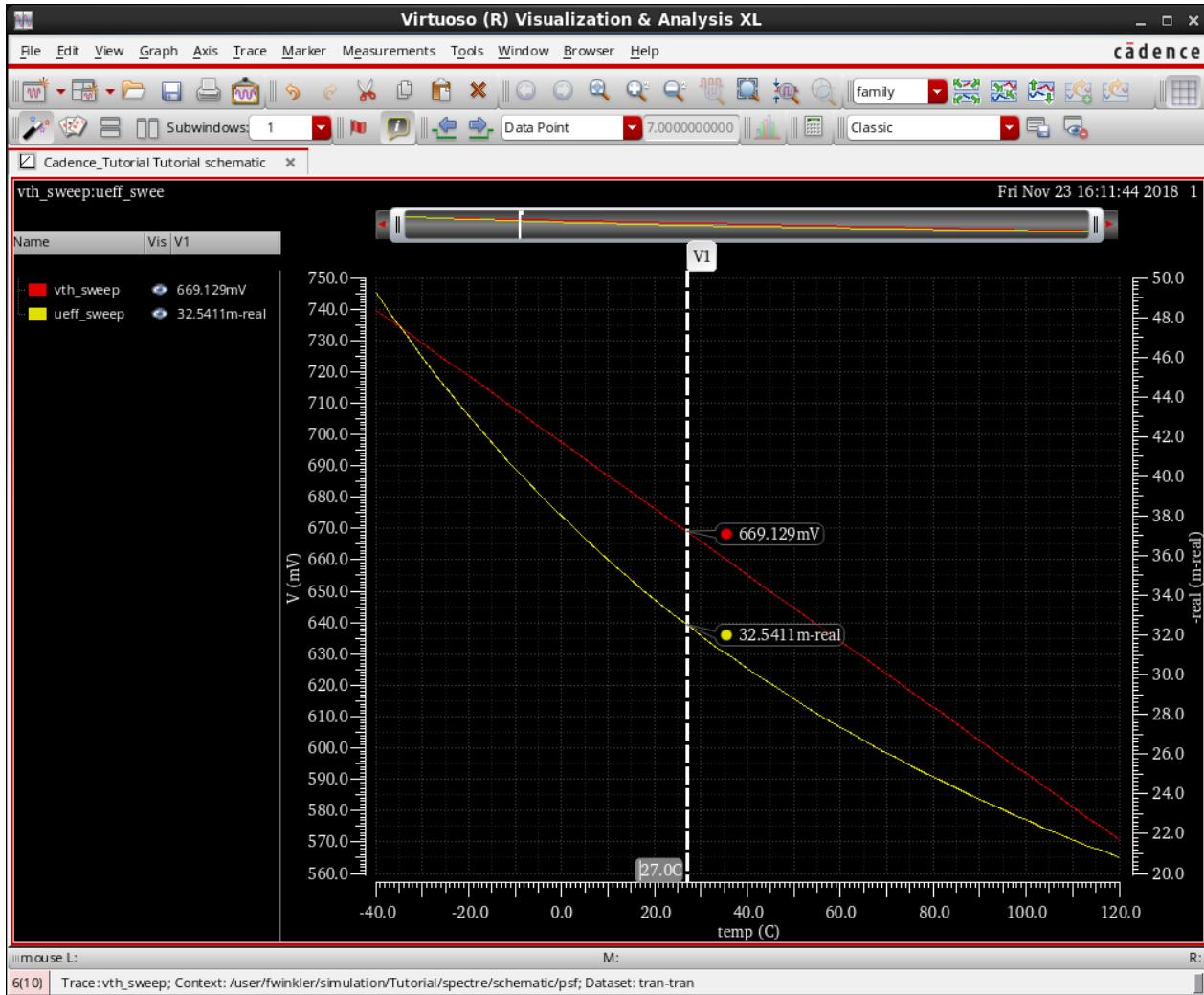


Figure 164: Threshold voltage and charge carrier behaviour as a function of temperature

It can be seen that both the threshold voltage and the charge carrier mobility decrease with increasing temperature. The reason for the reduction in threshold voltage is stronger atomic movement in the crystal structure of the semiconductor material, which means that charge carriers can be released from this structure more easily, resulting in higher conductivity of the channel due to a higher number of charge carriers. Similarly, the charge carrier mobility decreases with increasing temperature, as the stronger movement of the atoms in the crystal leads to more frequent collisions between the charge carriers and the atoms.

The influence on the transistor characteristic is complementary. According to formula (1), the decreasing threshold voltage leads to an increase, while the decreasing charge carrier mobility leads to a decrease in the transistor current. Which effect prevails depends on the specific operating point of the transistor.

17.3 Transient analysis in ADE-L

This chapter introduces transient analysis as a further simulation method. If you have not already done so, open the previously created circuit diagram of the inverter circuit and start ADE-L. To configure a transient analysis, open the *Choosing Analyses* window in the ADE-L simulation environment and select the *tran* field. A simulation time is entered in the *Stop Time* field, which in this case should be 3 ms .

When analysing transients, it is possible to improve the accuracy of the analysis at the expense of the duration of the simulation. The following three setting options are available in the *Accuracy Defaults (errpreset)* area.

- *conservative*: slower simulation speed, higher accuracy
- *moderate*: Moderate speed and accuracy
- *liberal*: Fast simulation speed, lower accuracy

Select the option *conservative*, as the simulation time is very short for a small circuit despite the high accuracy. If you do not select any of these fields, the simulation is carried out with the setting *moderate* by default. Then select the *Enabled* field and confirm the simulation setup by clicking on *OK*.

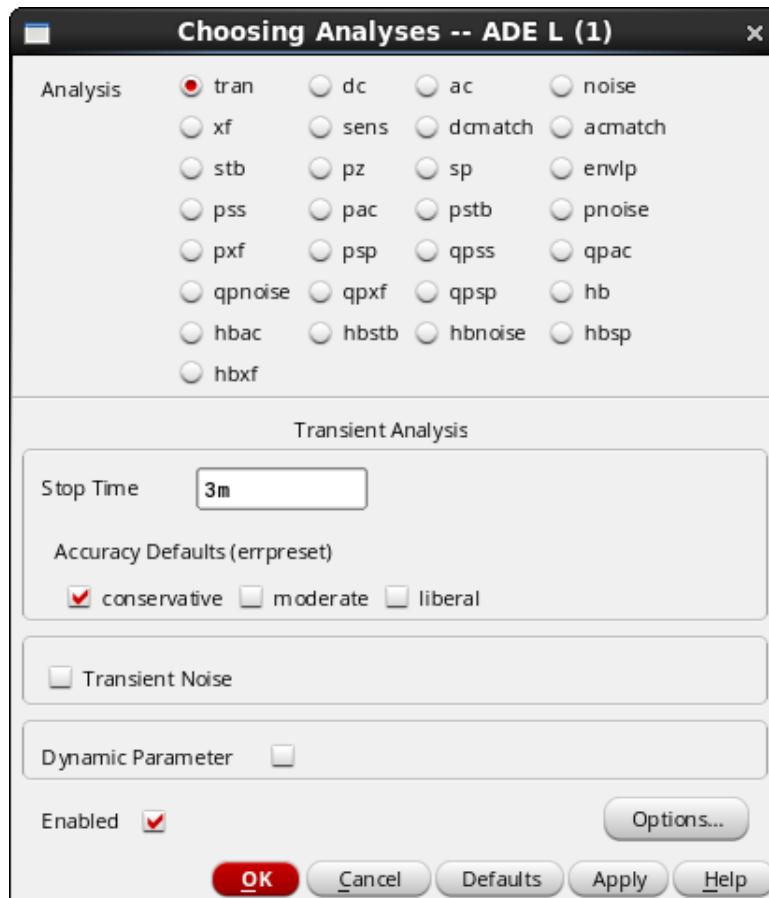


Figure 165: Setting up a transient analysis

You also need to define *Outputs* that you want to plot. Firstly, the input and output signal of the circuit should be chosen as *Output*. Open the *Setting Outputs* window known from *Chapter 6.6*. Click on *From Design* and select the input and output lines. Confirm by pressing **[ESC]**. Then start the transient analysis by clicking on the green arrow.

The analysis window opens, in which the graphical curves of the input and output signal of the inverter are displayed. The green diagram curve represents the input signal, while the red curve corresponds to the output signal. To make the diagrams clearer, click on *Split Current Strip* in the menu bar of the *Visualisation & Analysis* window. This function shifts these two curves into separate horizontal image sections, so-called strips, and improves clarity by adjusting the scaling of the axes, among other things. The diagrams can be merged again by clicking on *Combine All Analogue Traces*.

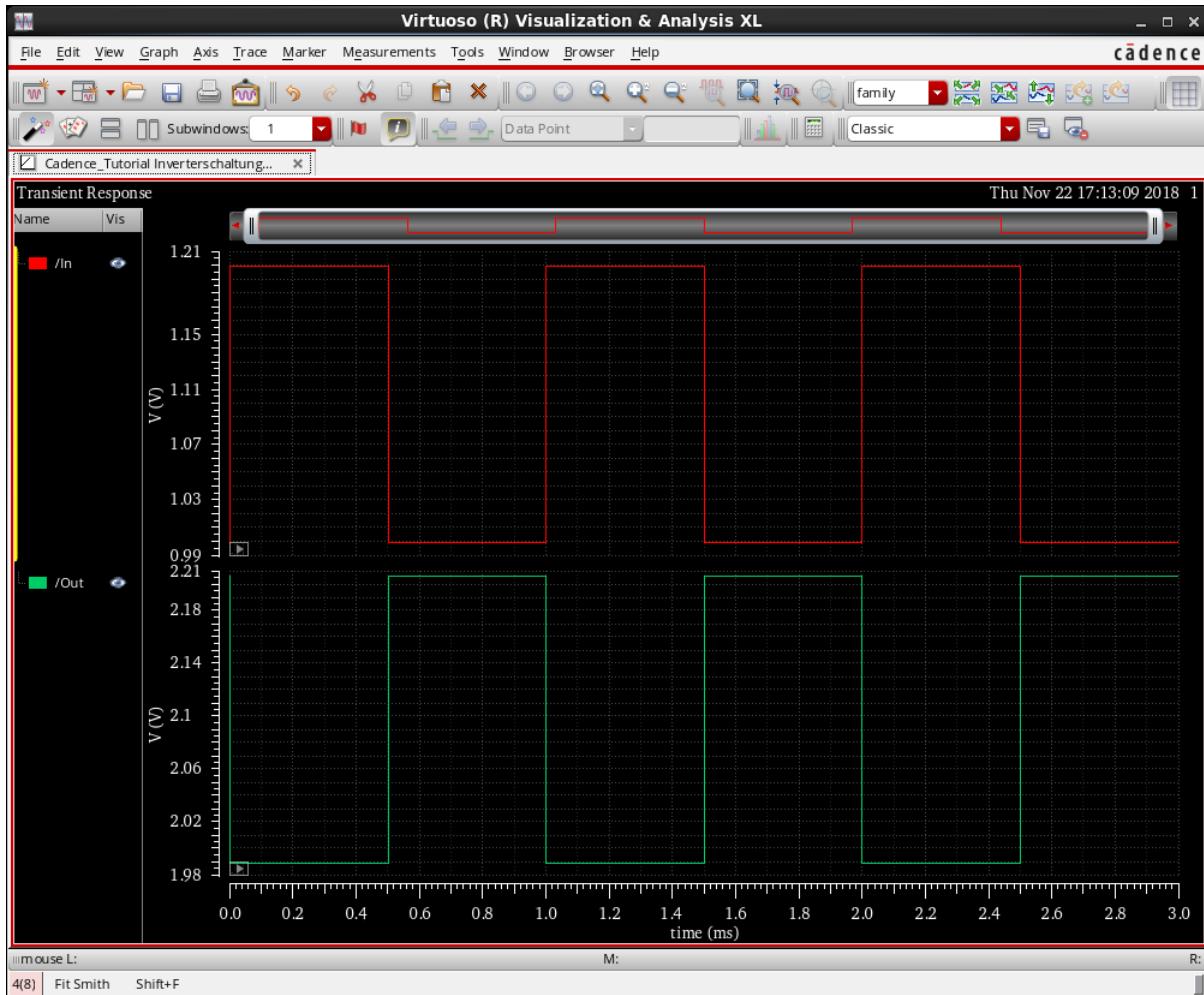


Figure 166: The result of the transient analysis

It can be clearly seen that the pulsed output signal is inverted compared to the input signal.

17.4 AC analysis in ADE-L

17.4.1 Preparing the circuit for an AC analysis

First delete the *noConn* connection at the output of the circuit. Connect a capacitor to the now open line of the inverting amplifier. To do this, press the [I] button in the circuit diagram editor and select the component *cap* from the *analogLib* library. After you have placed and connected the capacitor as shown in Figure 167, press [Q] and set the parameter *Capacitance* to 1nF . If you have not already done so, also set the parameter *AC magnitude* of the pulsed voltage source to 1V .

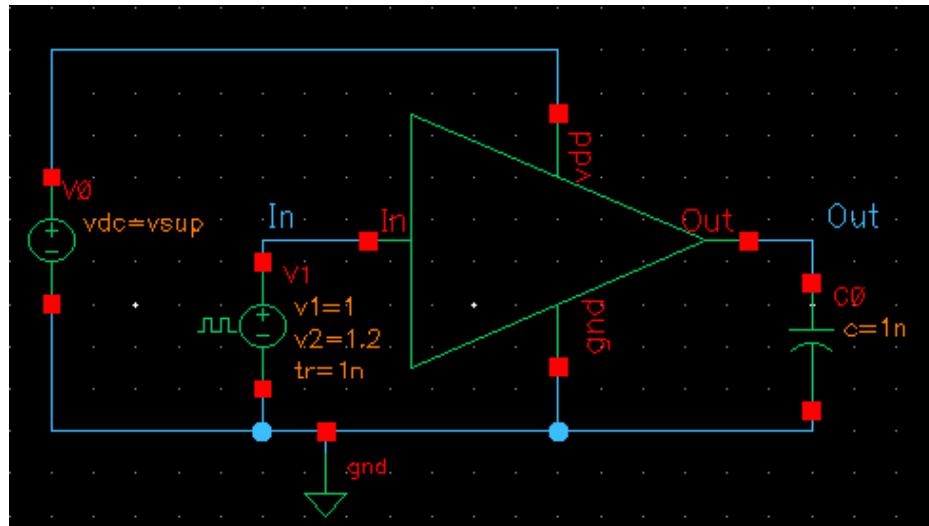


Figure 167: Amplifier circuit with capacitor at the output

17.4.2 Configuration of the AC analysis

Now call up ADE-L and set up an AC simulation via the *Choosing Analysis* window. Select *ac* as the simulation type with *Frequency* as *Sweep Variable* and enter the value 1 for *Start* and the value 100G for *Stop* to perform a simulation in the frequency range from 1 Hz to 100 GHz . Confirm the setup by clicking *OK*. A simple DC simulation must also be configured, as the parameters of the operating points determined there are used in the AC analysis.

You must select *Outputs* for the AC analysis. You should have already created these in the transient analysis. In the *Outputs* area of the ADE-L window, activate the input (*In*) and the output (*Out*) and deactivate all other *Outputs* that you have previously created.



Figure 168: Setting up an AC analysis

You also want to check the expected values in the AC simulation. Therefore, create a *Calculator* expression that determines the gain from the small signal variables. The following expression applies to the inverting single-ended amplifier in the implemented version.

$$A = \frac{gm_n}{gm_p} \quad (6)$$

The values gm_n and gm_p correspond to the transconductance of the NMOS or PMOS transistor. To be able to

extract these properties, the DC simulation must have been carried out beforehand. Transfer this formula to the *Calculator* by selecting the transistor slope (*gm*) using the *op* operator.

Once you have completed the formula, send the complete expression to the ADE-L window. To do this, first define a new *Output* with the name *gain* in the *Setting Outputs* window and then click on *Get Expression* to get the expression from the *Calculator*.

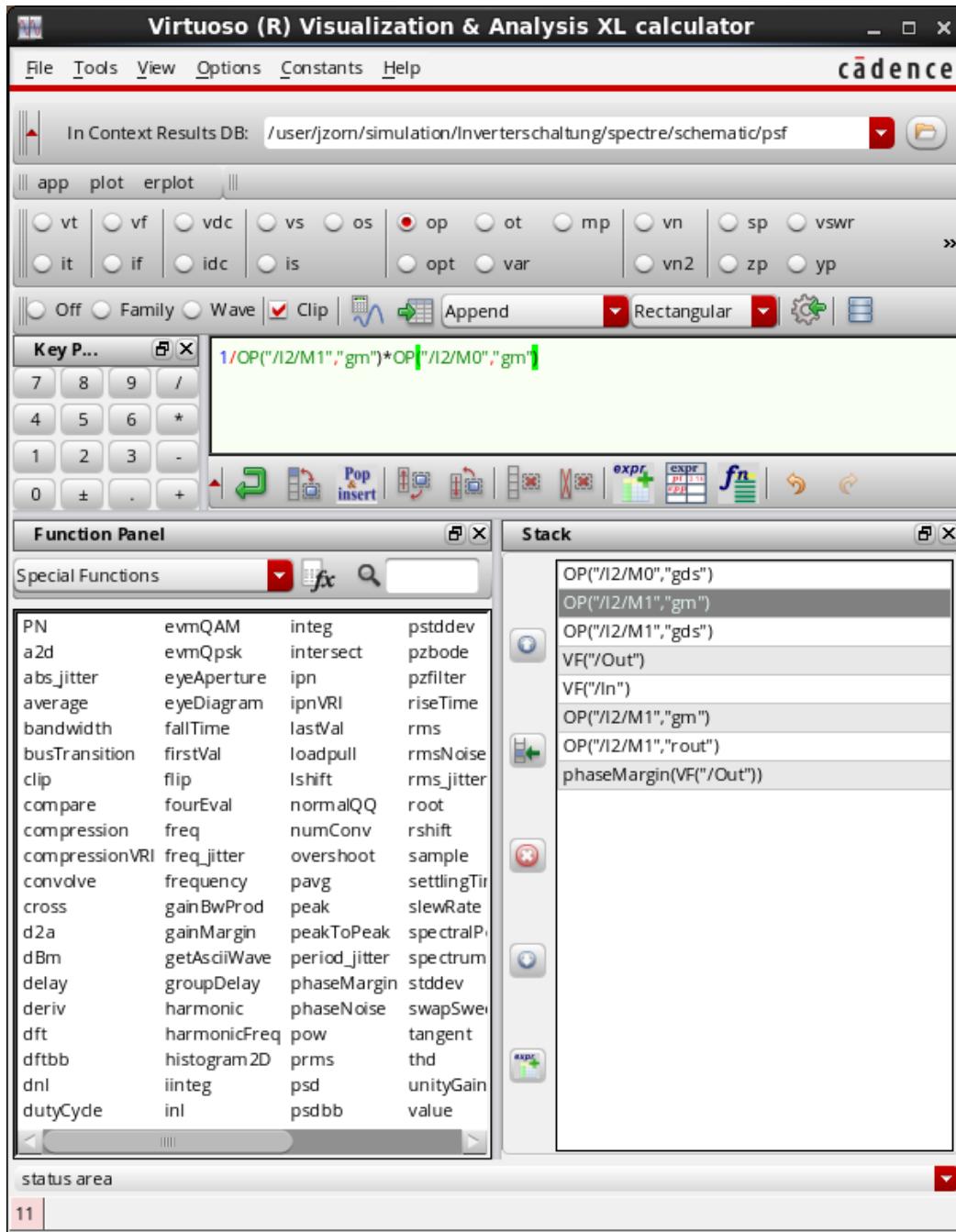


Figure 169: The *Calculator* window with formula (6) for calculating the gain

Then start a simulation and display the input and output voltage graphically.

It can be seen that the output voltage for a certain frequency range is higher than the input voltage by a factor

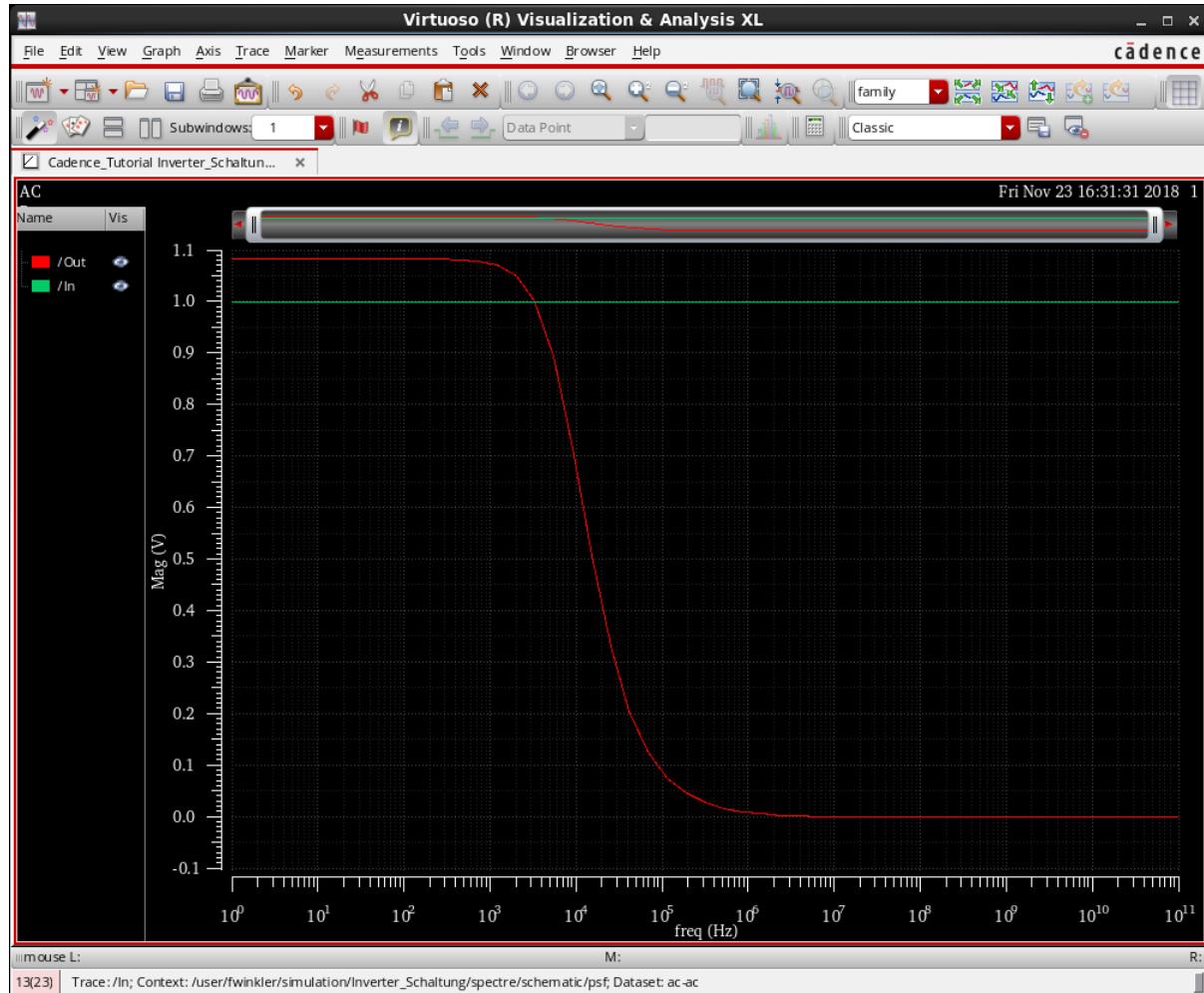


Figure 170: Plot of AC analysis with input and output voltage

of 1.08. This factor is the gain of the circuit. If you compare the gain value calculated from the transistor slopes with the value in the ADE-L window, you will see that the value is identical. It also becomes clear that the gain decreases from a certain frequency. The frequency range in which the gain of the circuit remains constant is called the bandwidth.

Next, you will determine the bandwidth of the amplifier in various ways based on the simulation results. To do this, you must first create a Bode diagram that shows the amplitude and phase response of the circuit.

To do this, click on *Results* in the ADE-L menu bar and then on *Direct Plot*. In the next submenu, select *AC Magnitude & Phase* and click on *Schematic* to select the output of the circuit. Confirm the selection with **[ESC]**.

A window now opens which displays the Bode diagram, consisting of amplitude (*magnitude*) and phase response (*phase*). For a better overview, separate the amplitude and phase response into separate strips using the function *Split all Strips*.

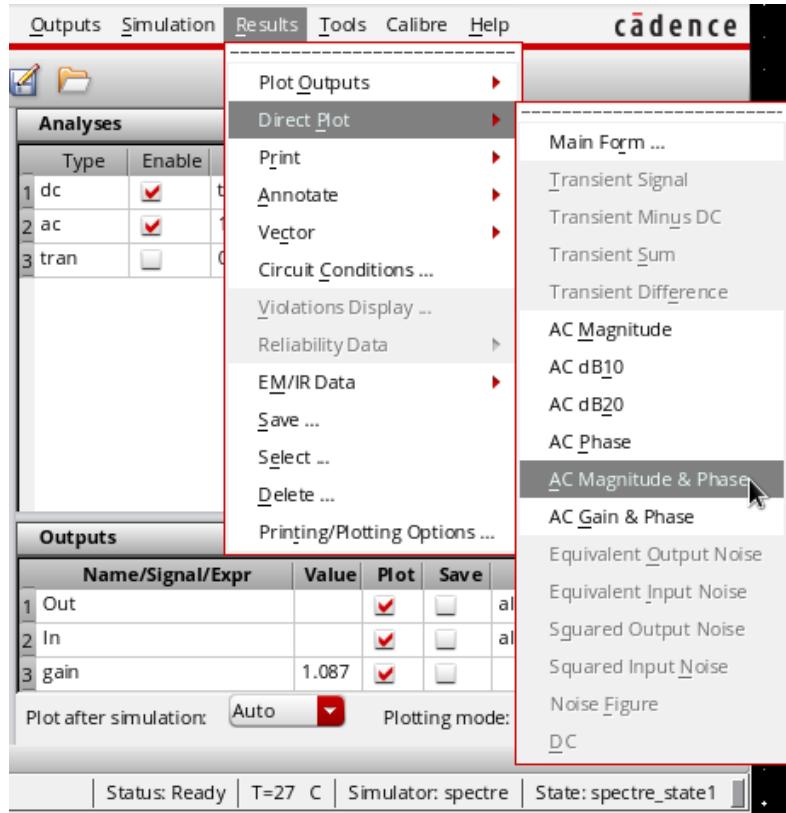


Figure 171: Calling up the display of *Magnitude and Phase*

In the *magnitude* graph, set a marker at 10^0Hz and another at the position where the amplitude characteristic has dropped by 3 dB . Alternatively, find the frequency at which the phase rotation has changed from 180° by 45° to 135° . The frequency at this point corresponds to the bandwidth of the amplifier.

The bandwidth should now be extracted from the simulation data using a *Calculator* function. To do this, start the *Calculator* again and search for the function *bandwidth* in the *Function Panel*. You can now specify various parameters for the function (see Figure 173). First select the signal of the output with the *VF* operator. This operator is used to select a signal from the *Schematic*, which generates an *Expression* for an AC voltage. In the *Db* field, the value is entered by which gain the Bode plot should drop to determine the bandwidth. Enter the value 3 dB in this field and press *OK*.

Send the finished expression directly from the *Calculator* to the *Output* area of ADE-L. The graphically determined bandwidth value of about 8.086 kHz should be almost identical to the calculated value.

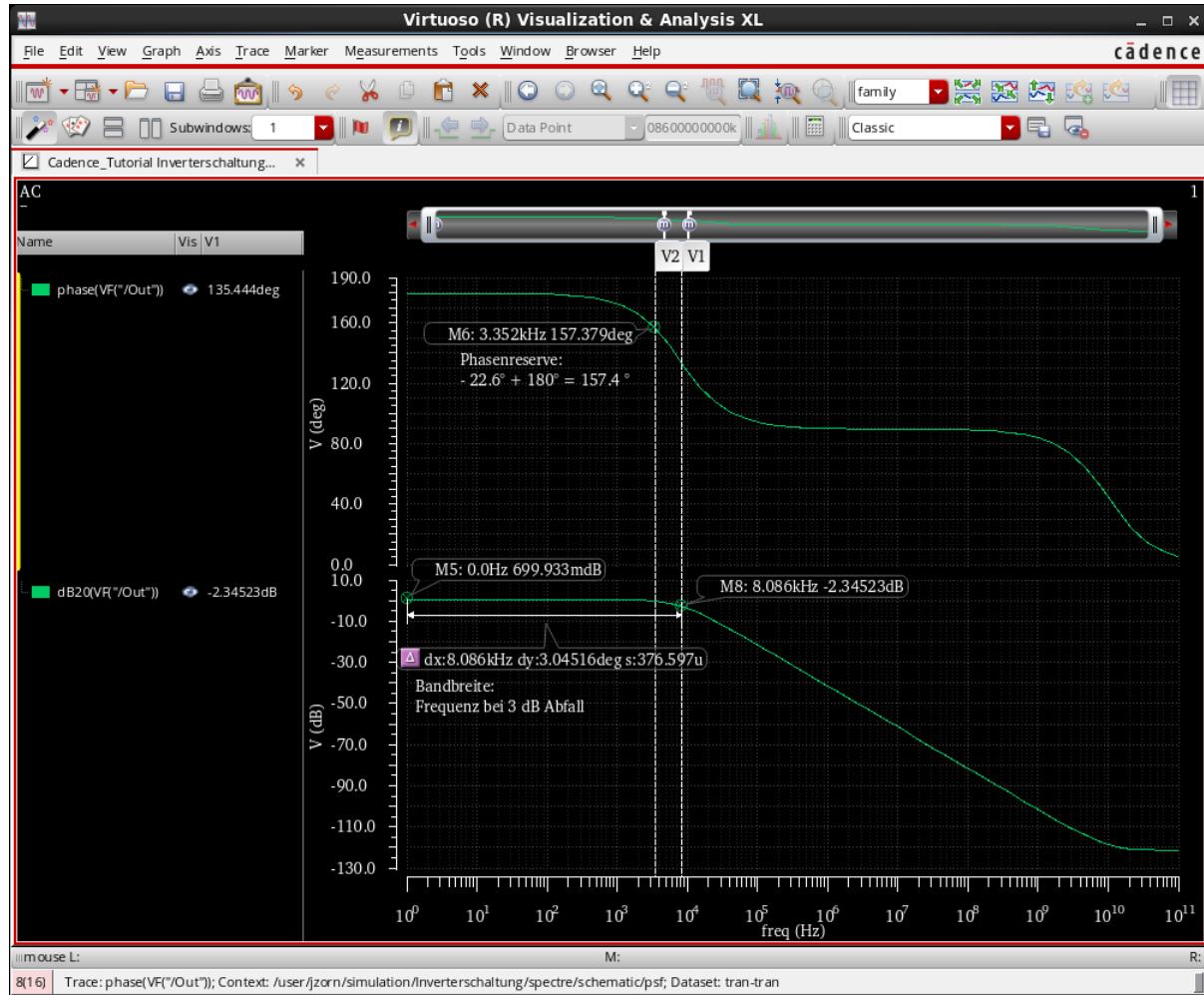


Figure 172: Bode diagram, consisting of frequency and amplitude response

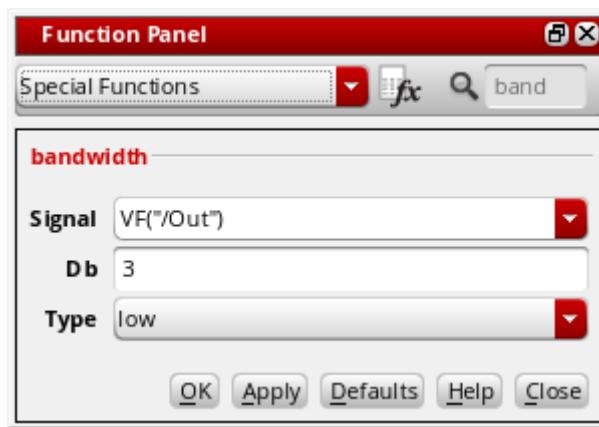


Figure 173: Parameters of the bandwidth function in the *Calculator*

The phase margin of the circuit should also be determined in the AC simulation. Set another marker at the 0 dB line in the Bode diagram (compare *Figure 172*). Then read the value in the phase margin and subtract it from 180° to determine the phase margin.

The phase margin should also be calculated using the *Calculator*. Proceed in exactly the same way as for calculating the bandwidth and search for the function *Phase Margin* in the *Function Panel*. Then transfer the expression to the *Outputs* of ADE-L, simulate again and compare the results.

Name/Signal/Expr	Value	Plot	Save	Save Options
1 Out		<input checked="" type="checkbox"/>	<input type="checkbox"/>	allv
2 In		<input checked="" type="checkbox"/>	<input type="checkbox"/>	allv
3 gain	1.087	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
4 phaseMargin(VF"/Out")	-22.61	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
5 bandwidth(VF"/Out") 3 "I...	8.086K	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Plot after simulation: Auto Plotting mode: Replace

Figure 174: The *Output-Area* of the ADE Explorer analysis window with all the expressions used

17.5 DC analysis in ADE-XL

17.5.1 Starting ADE-XL

The following section introduces the ADE-XL tool. This simulation tool extends the familiar ADE-L environment with more advanced analysis options. A key feature of ADE-XL is the configuration of multiple simulations. These are referred to as *Tests* and enable the simultaneous execution of different analyses (as known from ADE-L) with different variables. Different designs can also be simulated within an ADE-XL environment. In addition, it is possible to create global variables that are valid in all *Tests* created and can be varied in a predefined range if required. All simulation results are displayed centrally in the main window of ADE-XL. The following chapters will also deal with simulation types that are only possible in ADE-XL. These are in particular the Corner and Monte Carlo simulation.

At the beginning a new *View* must be created for ADE-XL. To do this, the tool, analogous to ADE-L, is started via the menu bar by clicking on *Launch > ADE-XL*.

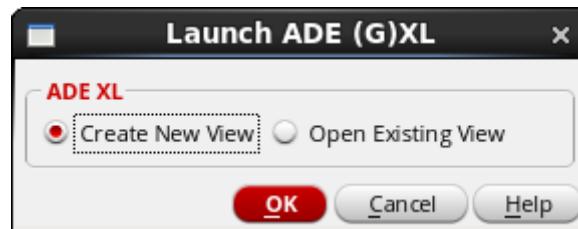


Figure 175: Selection of a new or existing ADE-XL view

When you start ADE-XL for the first time, click on *Create New View* and in the next window on *OK*. If you have already worked with ADE-XL and saved a *View*, you can open this *View* by selecting the setting *Open Existing View*.

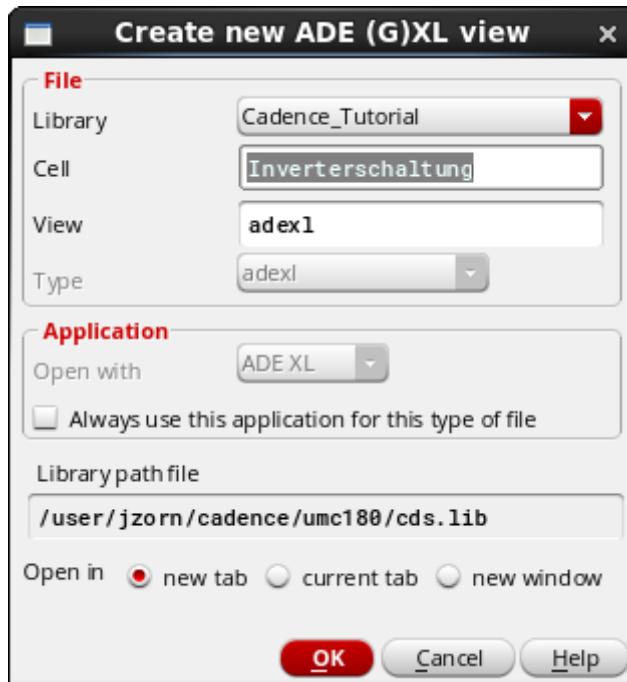


Figure 176: Configuring a new ADE-XL-View

The main window of ADE-XL appears.

17.5.2 Optimization of the simulation speed

The simulation speed can be significantly accelerated by running several simulations in parallel. However, the optimum number of simulations running in parallel depends on the number of available processor cores installed in the available simulation computer. To determine the number of *Cores* available in your computer, click on *System > About this computer* in the Linux menu bar. The *System monitoring* window that opens contains information about the installed operating system and the available hardware. In particular, the number of processor cores can be read here.

To configure the parallel execution of simulations in ADE-XL, select the menu item *Run Options* in the ADE-XL main window under *Options* and select the option *parallel* in the *Run in* area.

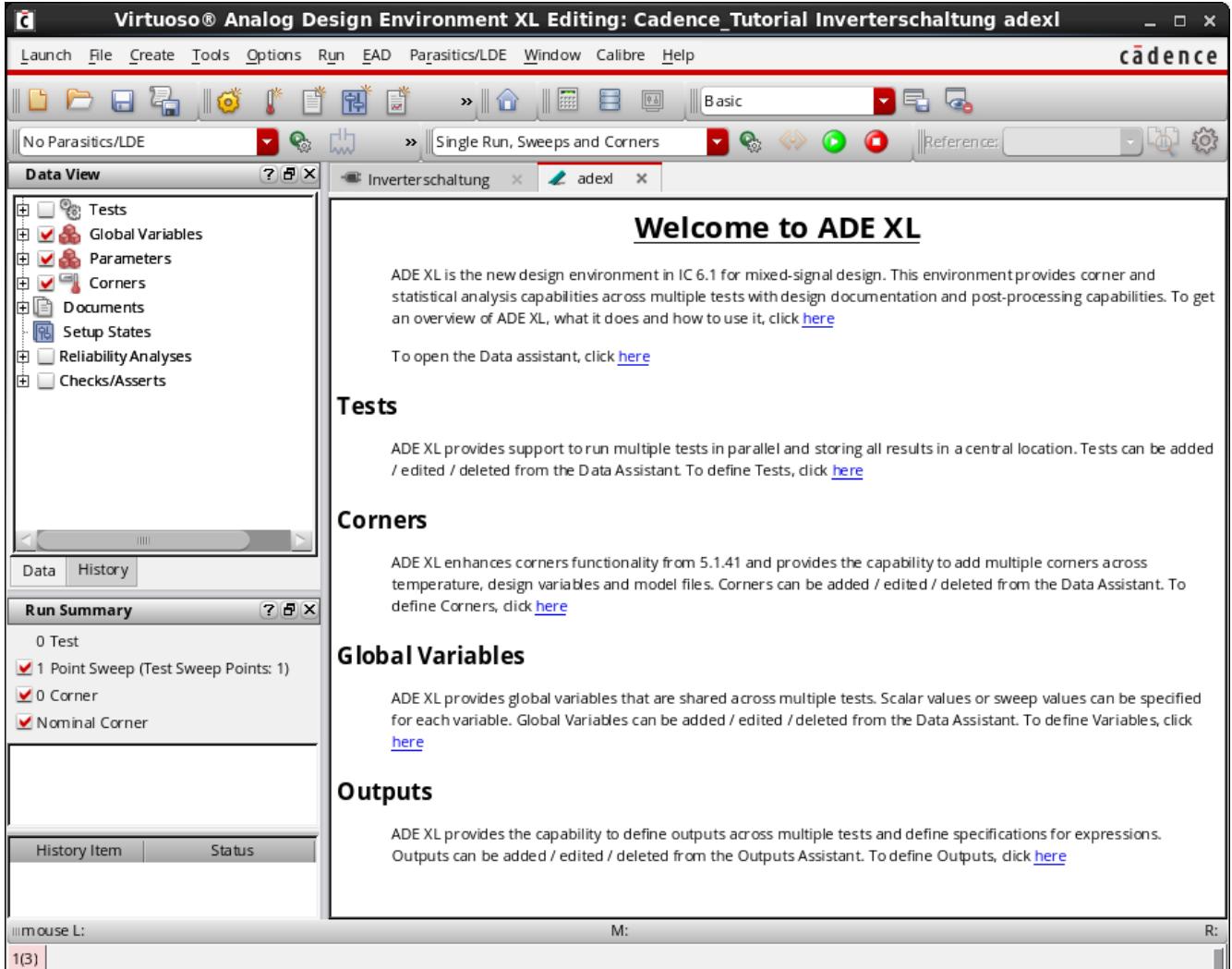


Figure 177: The main window of ADE-XL

Select the number of simulations running in parallel in the *Options* menu under the *Job Setup* menu item. In the *Max. Jobs* area, enter the number of CPU Cores that you have previously determined. In the *Run Summary* area of the ADE-XL main window, you can check how many jobs are running in parallel during the simulation execution. Each running simulation corresponds to a small square. The number of simultaneously displayed squares corresponds to the number of simultaneously running simulations.

Note: These settings are not saved in Cadence Virtuoso and must be reset each time Virtuoso is started. Furthermore, the settings only apply to the currently processed ADE-XL session.



Figure 178: Display of the system values in the system monitoring



Figure 179: Setting the Run Options

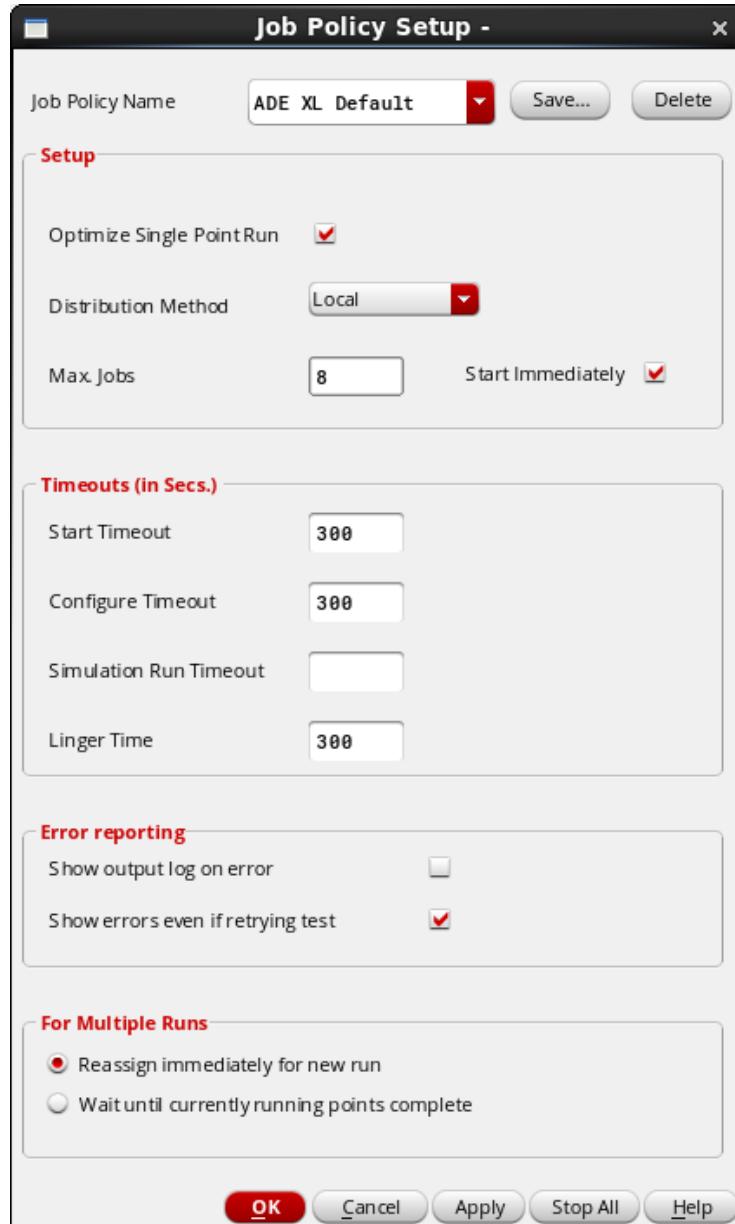


Figure 180: Enter the number of CPU-Cores

17.5.3 Conversion of the circuit to an inverter

For a meaningful DC analysis, the inverting amplifier must be converted into an inverter. To do this, the gate-drain connection on the PMOS transistor is removed. The two gates of the transistors are short-circuited and connected to the input pin (*In*).

The transistor geometries remain unchanged.

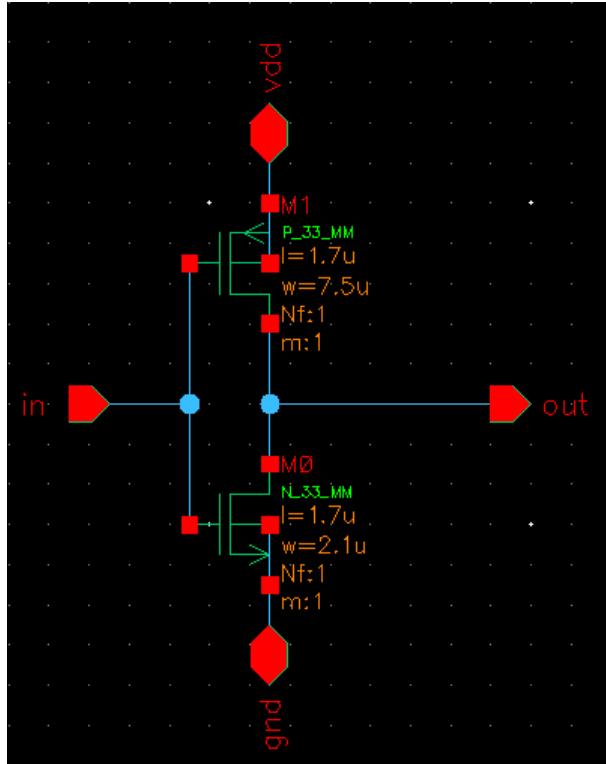


Figure 181: Inverter circuit in the Schematic

17.5.4 Configuring a DC analysis in ADE-XL

To begin, you must create a new test by clicking on the plus symbol next to *Tests* and selecting *Click to Add Test*. The ADE XL test editor window opens, which is very similar to the familiar ADE L window. You will be asked to select the design for which you want to define a test. The cell preselected in this window always corresponds to the cell from which ADE-XL was started. As you have started ADE-XL from the *Schematic* of the inverter circuit, you only need to confirm with *OK*.

Here you can now import an existing ADE-L-State by clicking on *Session* in the menu bar and then on *Load State*. Tests configured in ADE-XL are comparable with several independent ADE-L-States. You have the option of using your existing *State* from ADE-L and deactivating unnecessary analyses. Alternatively, you can configure a new DC simulation directly within a test in ADE-XL.

To configure a new DC simulation, proceed as follows

1. If you have not yet created a test, create it as described in the previous paragraph.
2. Import all variables from the *Schematic* by right-clicking in the *Design Variables* area and selecting *Copy from Cellview*. The variables *vin* and *vsup* should then be visible. Set *vin* = 1 V and *vsup* = 3.3 V.
3. As explained in *Chapter 7.1*, configure a DC simulation with *Sweep* by clicking on *Click to add analysis* in the *Analyses* area. Select *vin* as the *Sweep* variable. Also select *Save DC-Operating Point* and confirm with *OK*.
4. Finally, configure *Outputs* for the DC simulation. To do this, proceed as in *Chapter 6.5*. Record the output voltage *vout* as *Output*.

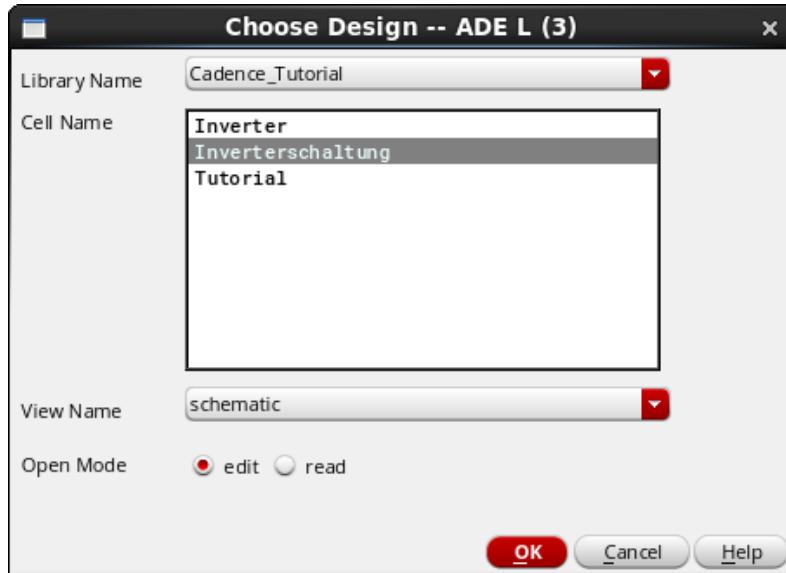


Figure 182: Configuration of a new test

ADE-XL differentiates between local and global variables. Local variables are only valid locally for a single test. Global variables are valid for all tests and also overwrite the values of local variables with the same name. Local variables from the test are automatically also available as global variables. However, the value can vary.

Now click on the plus symbol next to *Global Variables* and then on *Create Global Variable*. Give the global variable *v_{sup}* a value of 3.3 V. Also remember that the variable names refer to the names in your Schematic!



Figure 183: Create a new global variable

As with ADE-L, ADE-XL offers the option of using the *Calculator* tool. To start this tool, click on *Tools* in the ADE-XL main window and then on *Calculator*.

You should now formulate a *Calculator* expression that determines the value of *vin* at which an output voltage *vout* is reached that corresponds to half the supply voltage, i.e. 1.65 V. Use the *cross* function, which can be found in the *Function-Panel* of the *Calculator*. This function determines when a *signal* reaches a specified comparison or threshold value (*Threshold Value*).

Select the output signal of the inverter using the *Calculator* operator *vs*, as the simulation of the output voltage is a *Sweep*. Select half the supply voltage as the threshold value. To do this, you can use the *op* function to select the voltage of the supply voltage source. Click on this voltage source and select *v* from the list. This value must also be halved, which can be done using the notation *vdc/2* in the *Calculator*. The advantage of this procedure is that any change in the supply voltage is transferred directly to the *Calculator* expression without

further adjustment. Enter 1 as *Edge Number*. For *Edge Type*, select *falling*, as the simulation starts with an input voltage v_{in} of 0 V and the full supply voltage v_{dc} of 3.3 V is therefore applied to the output at the beginning. The threshold value is therefore determined on a falling edge. For *Number of occurrences*, click on *single*, as the threshold value is only reached once in this simulation.

Compare your expression with the expression from *Figure 184* on the next page. Then select the expression and copy it with [CTRL]+[C].

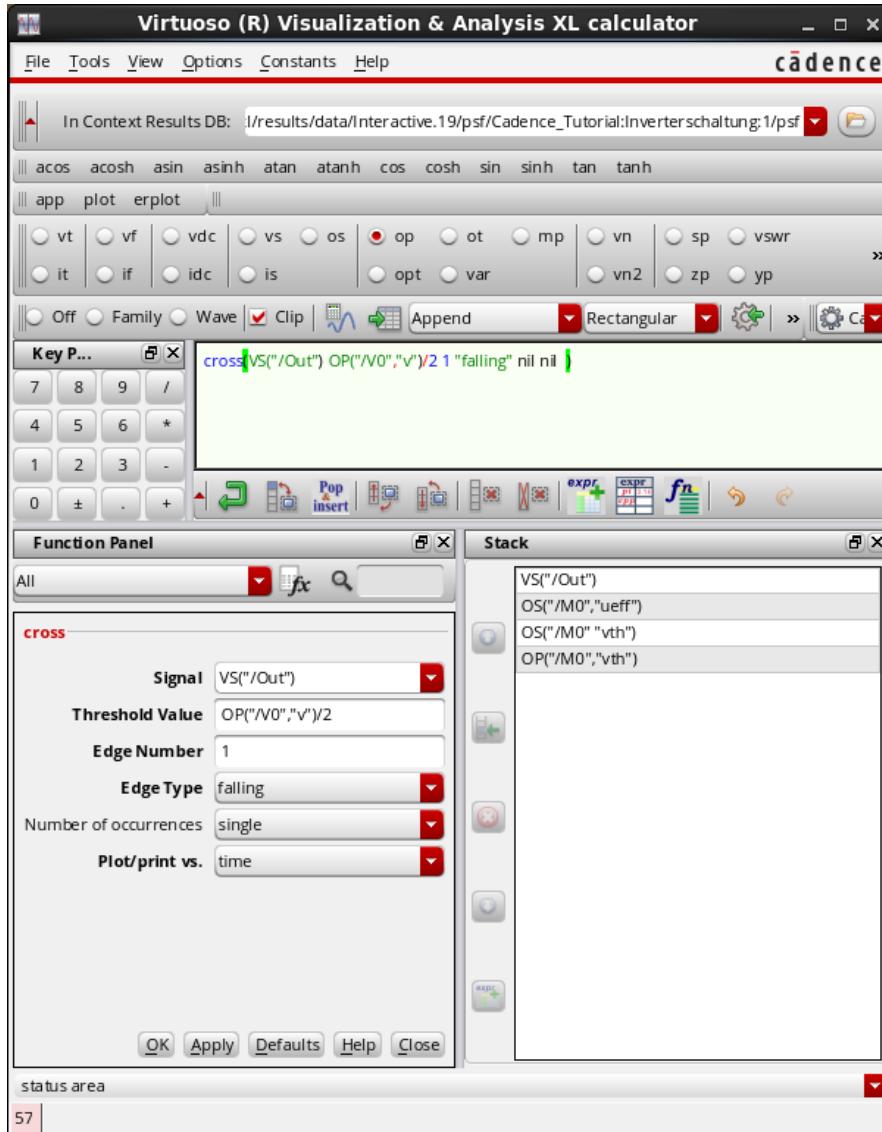


Figure 184: The complete cross-Expression in the Calculator

You can create a new *Output* directly in the main window of ADE-XL and assign it to a test. To do this, click on the red arrow of the *Add new output* button in the *Outputs Setup* tab in the menu bar. A submenu appears in which you can specify the type of newly created *Outputs* (Figure 185).

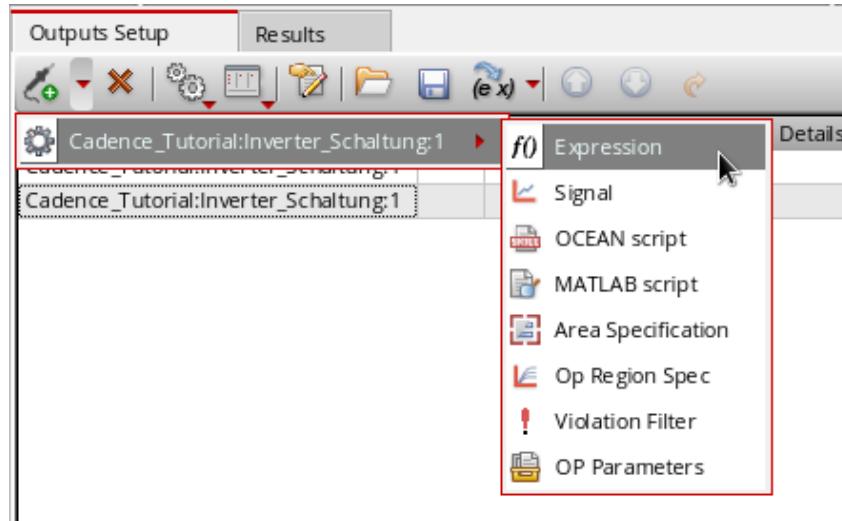


Figure 185: Configuration of *Outputs* in ADE-XL

You should configure the *Calculator* expression created and copied in the previous paragraph as *Output*. Therefore, select *Expression* as the *Output* type. You will now see the newly created *Output* in the ADE-XL main window (Figure 186). You can name the *Output* by double-clicking on *Name*. In the *Type* column, you can see the type of *Outputs*. Double-click in the *Details* cell. A text field and a button with three dots opens.

Test	Name	Type	Details	EvalType	Plot
Cadence_Tutorial:Inverter_Schaltung:1		signal	/Out	point	<input checked="" type="checkbox"/>
Cadence_Tutorial:Inverter_Schaltung:1		expr	<input (op(\"="" 1="" 2)="" \"falling\"="" \"v\"))="" nil="" nil"="" out\")="" type="text" v0\"="" value="cross(VS(\"/> ...	point	<input checked="" type="checkbox"/>

Figure 186: Inserting an expression as *Output* in ADE-XL

Paste your already copied *Calculator* expression into the text field by clicking on it and pressing [CTRL]+[V]. Clicking on the button with the three dots opens the *Calculator* for a *Output* of type *expr*. You can also set up signals from the *Schematic* as *Output* in this way. To do this, proceed as described above and select *Signal* instead of *Expression* as the *Output* type. In this case, the button with the three dots opens the *Schematic*, where you select signals by clicking on them in the same way as in Chapter 6.6.

Now carry out the DC analysis and check the value determined by the *Calculator* expression.

Test	Output	Nominal
Cadence_Tutorial:Inverter_Schaltung:1	/Out	
Cadence_Tutorial:Inverter_Schaltung:1	cross(VS("/Out") (OP("/V0" "v") / ...	1.557

Figure 187: The calculated value of the cross-Function

Click on *Plot All* in the menu bar or on the graph symbol in the *Nominal* column to display the plot. In the diagram that opens automatically, you can now compare the calculated result with the graphical solution. To do this, place a marker in the diagram by pressing the [M] button. Double-click on the marker to open a settings window in which you first select *by XMode* under *Position* and then enter the value determined by *Calculator* in the field next to it. Confirm with *OK*. Check whether the value you read on the Y-axis corresponds to half the supply voltage (take into account possible rounding errors).

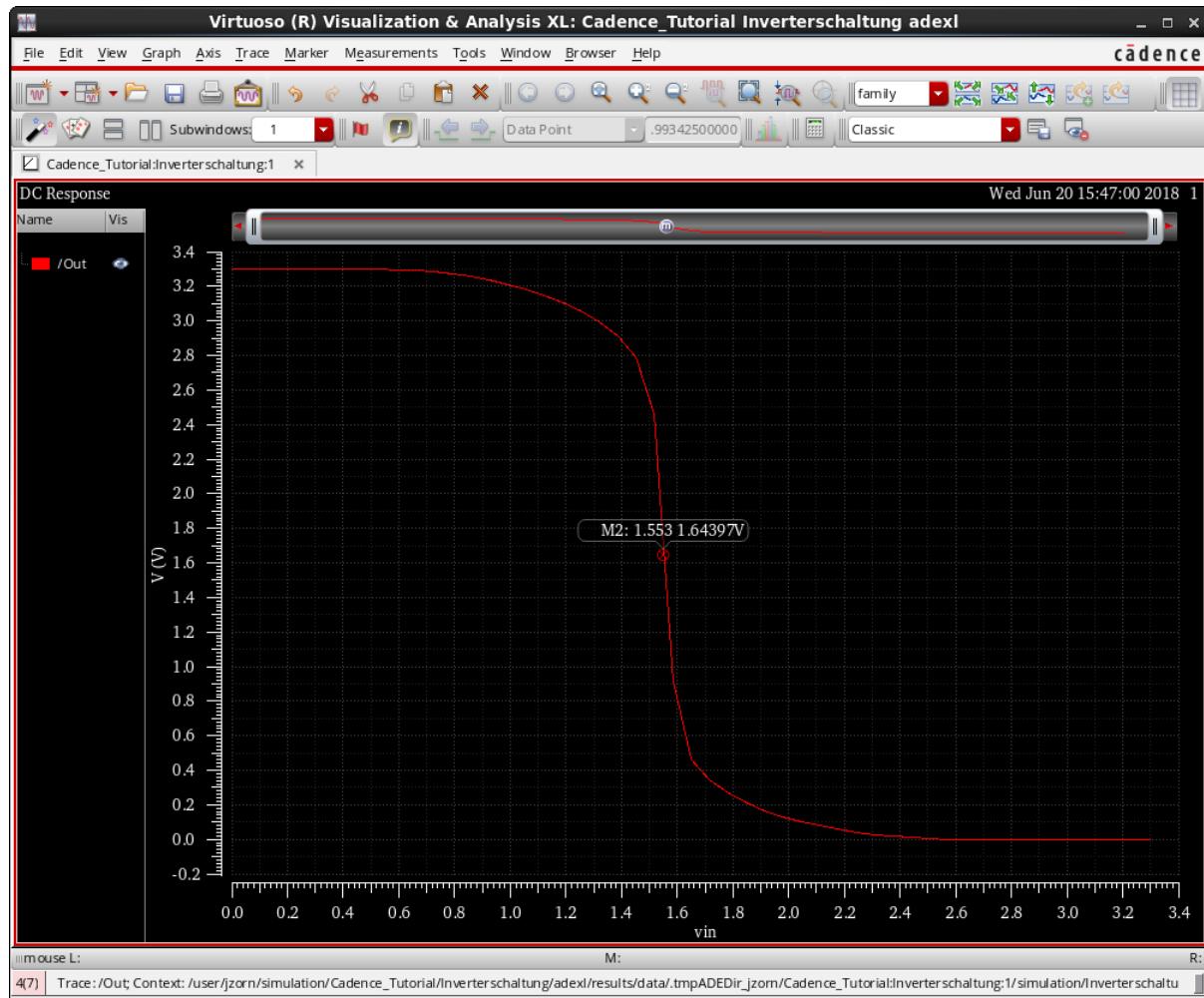


Figure 188: Diagram with marker at the value resulting from the cross-Function results

17.5.5 Sweeps via global variables in ADE-XL

In ADE-XL, you have two options for setting up Sweeps. You can define these using the test editor, which is similar to the procedure for ADE-L. Alternatively, you can also create a global variable and define a value range for this variable in ADE-XL. A significant difference is that with a Sweep with a global variable, a working point is calculated for each variable value. With Sweeps configured within a test, the operating point calculation only takes place for the starting point, while intermediate values are not saved.

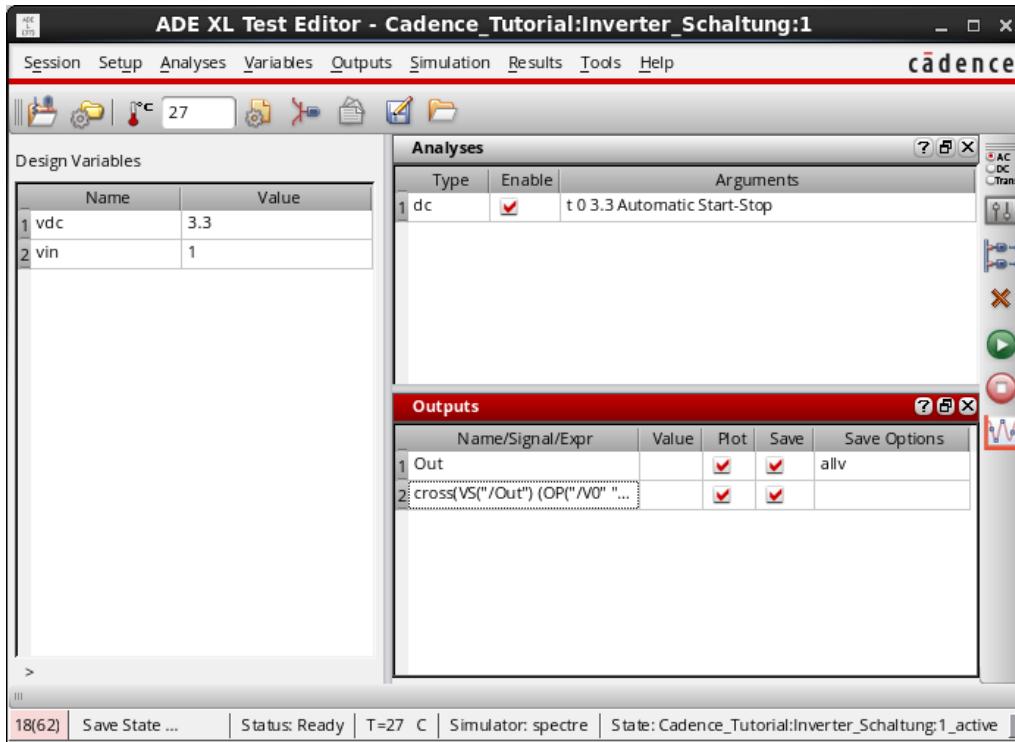


Figure 189: Configuring a test in ADE-XL

First deactivate the Sweep within the test by clicking on the plus symbol below the test. Then open another submenu under Analyses and double-click on the DC simulation to open the Choosing Analysis window. Uncheck Design Variable in the Sweep Variable area. To be on the safe side, check at this point whether Save DC Operating Point is checked and confirm with OK.

If it does not already exist, create the global variable *vin* to define the sweep outside the test. To do this, click on *click to add variable* in the Global Variables area of the ADE-XL main window and create the variable there. Then double-click on the value of the variable and open the Parameterize window (Figure 190) by clicking on the button with three dots.

Here you first delete the default setting by clicking on the Delete Spec button and then create a new From/To parameterization. Start at 0 V and go to 3.3 V in 100 steps. The more Steps you choose, the more precise the representation of the characteristic curve is. However, this also increases the simulation time. Leave the Step Type at Auto. Then confirm with OK and return to the ADE-XL main window, where the newly created global variable and its value range are now displayed.

Check that *vin* is crossed out in the area under *Design Variables*. As a global variable, *vin* now applies to all configured tests and overwrites a value previously defined within a test.

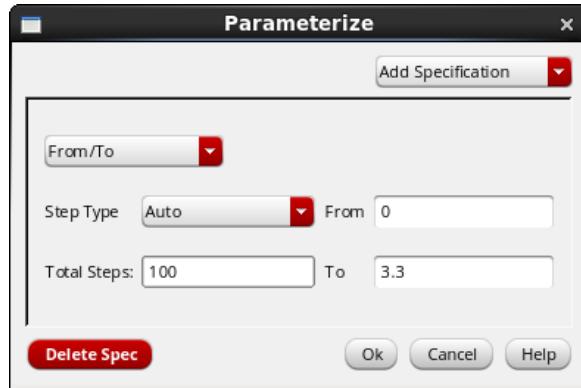


Figure 190: Setting one Sweep mit 100 Steps the global variable *vin*

With a Sweep via a global variable, an operating point is calculated for each Sweep-Step and is also saved using the Save DC Operating Points setting. Therefore, you need to make a change to the configuration of the Outputs in order to be able to display the simulation results graphically. To do this, configure a new Output for the test of the DC simulation in ADE-XL of type Expression.

Then click on the button with the three dots in the Details cell to open the Calculator. Create an expression for a DC operating point by clicking *vdc*. The Schematic of the inverter circuit opens. Select the net of the output signal *vout* by clicking. Also disable the cross expression in the Outputs by removing the arrow in the Plot button. This is necessary because this expression was configured for the evaluation of a sweep configured within the test. A sweep over the global variable *vin* would lead to evaluation errors.

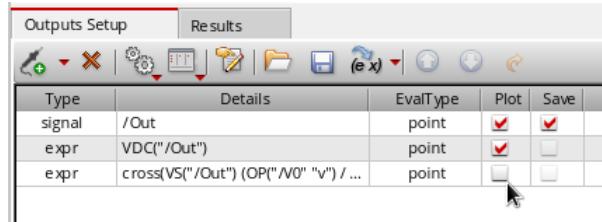


Figure 191: Disabling the cross expression for Sweep via global variables

Now start the simulation by clicking on the green arrow in the main window of ADE-XL. After the simulation is completed, the results of the simulation run are listed under the Results tab. In this example 100 sweep points should be listed. For better clarity, click on the arrow of the Details selection menu in the toolbar of the Results window and select Detail - Transpose.

This illustration is clearer, especially for simulations with many results. Plot the values by clicking the Plot-all button in the menu bar of the Result window. The result is the same course of the DC sweep as in Figure 188. The difference to the previous simulation is that for each Sweep-Step an operating point has been calculated, which can be displayed in the Schematic.

To do this, right-click on a simulation result in the Results tab and select DC Operating Points in the Annotate menu.

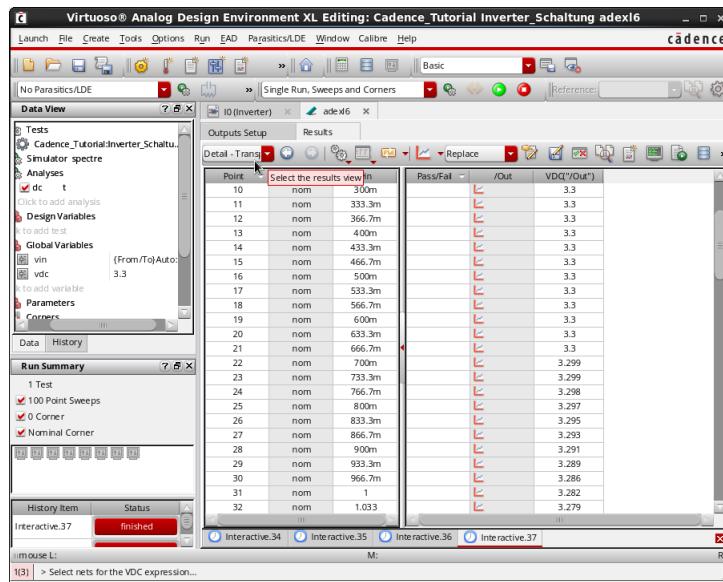


Figure 192: Transposed view of the Results window

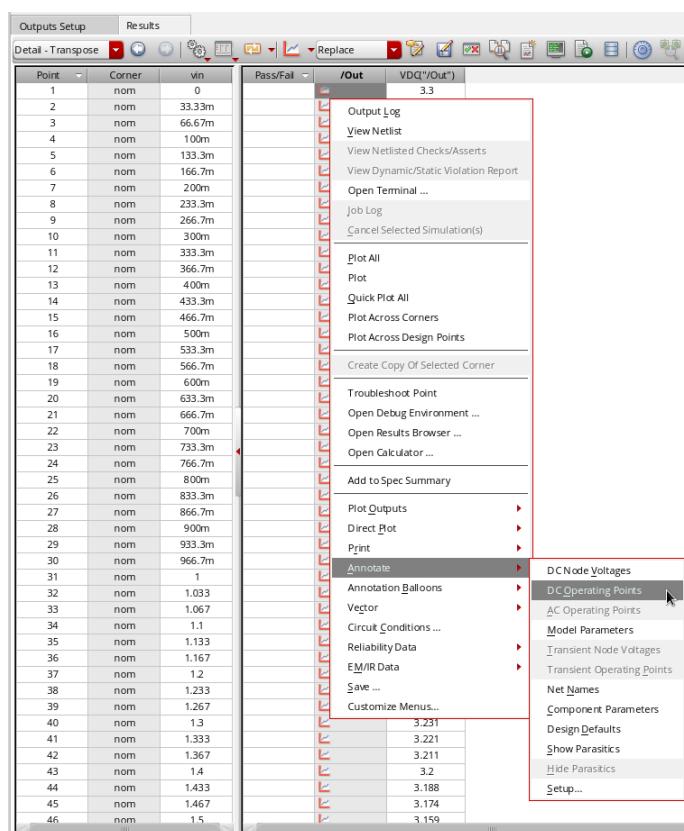


Figure 193: Display of specific Operating Points in Schematic