
Cadence Tutorial

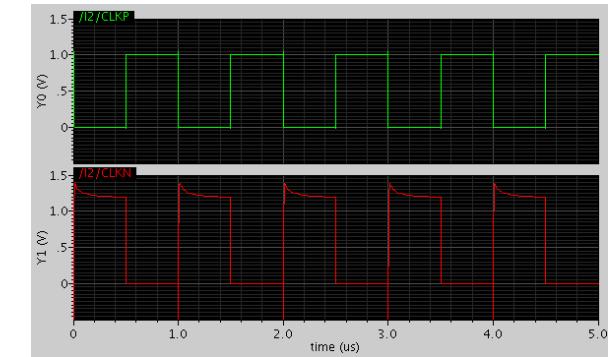
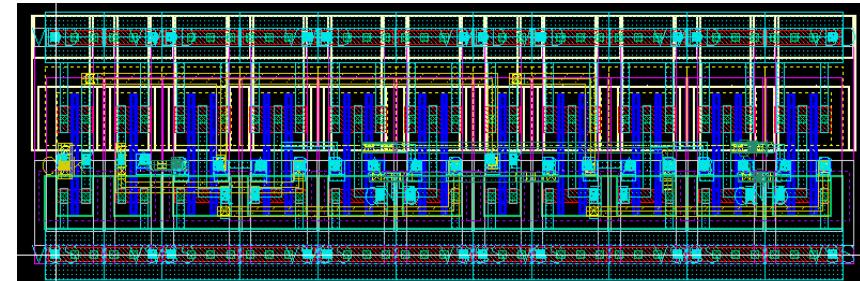
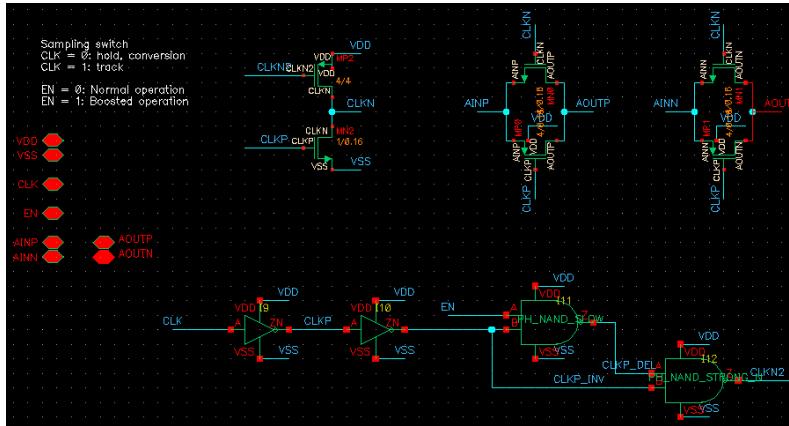
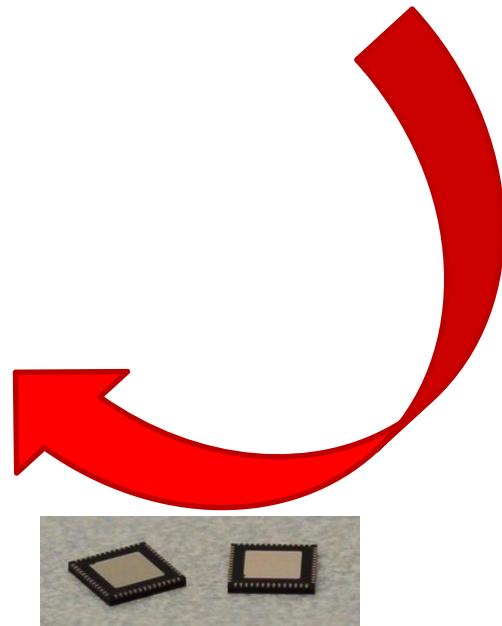
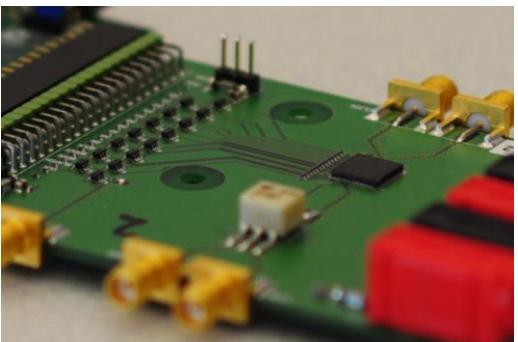
Cadence GPDK045

05/03/2025

Pieter Harpe

IC Design Flow

- Schematic design
- Simulation
- Layout design
- Layout verification
- Post layout simulations
- Chip fabrication
- Measurements



Technology and Tools

- Technology: 45nm CMOS process (Cadence GPK045)
- Software: Cadence Virtuoso
 - For circuit design, simulations, and layout
 - The software runs on an IC-group server, which you can access from your own PC
- Google and YouTube (etc.) are VERY handy to answer your Cadence questions!

Outline

- Installation of the software
- Starting Cadence
- Basic use of Cadence
 - Schematic design & Simulations
 - Layout

Installation of the Software

- Make a VPN connection to the TU/e
- Follow the procedure on the following website to install a client on your laptop so you can connect to the server with Cadence:
<https://sambava.ics.ele.tue.nl/tiki-index.php?page=Software%20-%20NX-SSH>
 - You will use the Daraina server
 - Install the X2GO client
 - Carefully follow all steps for installation and apply the correct settings

Starting Connection

- Make a VPN connection to the TU/e if you did not do so yet
- Use the X2GO client to connect to the server
- Once on the server, you can start using Cadence
 - All your designs and simulation results will be stored on the server
- **Important notes:**
 - There is a disk quota for each account of 3GB. If you exceed this (for instance due to simulation data), you may not be able to write additional information, but you will first have to delete something.
 - Please TERMINATE your session after working and do not SUSPEND it. Suspending leaves the memory and licenses in use, which can at some point cause problems.
 - Simulations which take up more than 24hours are considered wrongly set and thus can be terminated by the administrator.

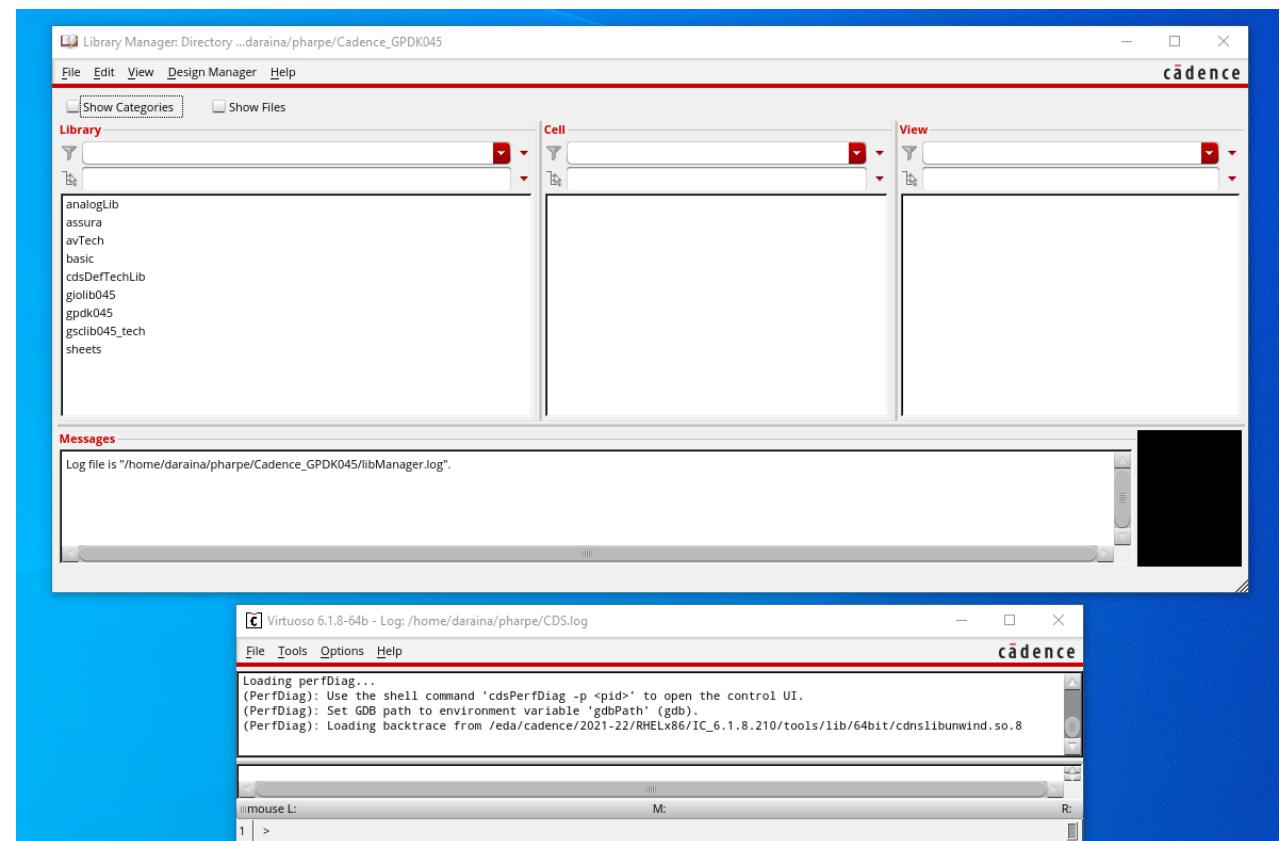
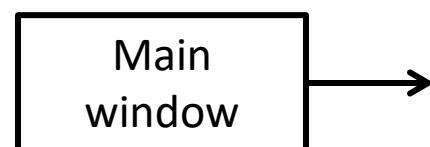
Starting Cadence

- Click the left mouse button on the UNIX desktop and choose “Cadence Virtuoso”
 - If there are multiple icons, choose “GPDK045”
- Cadence should now start



Starting Cadence

- You should see 2 windows now: the main window and the library manager.
- If the library manager did not open, you can start it from the main window from the menu “Tools” → “Library manager”



Useful Libraries

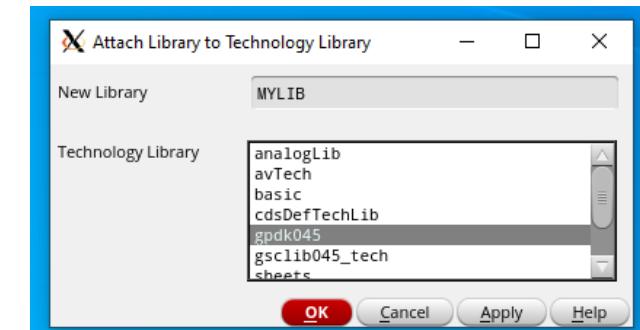
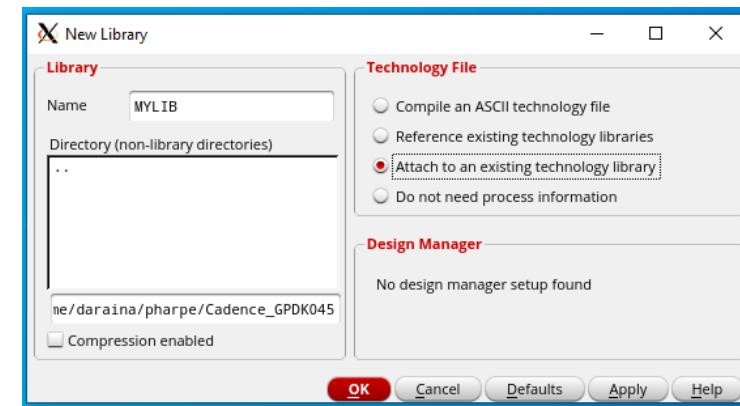
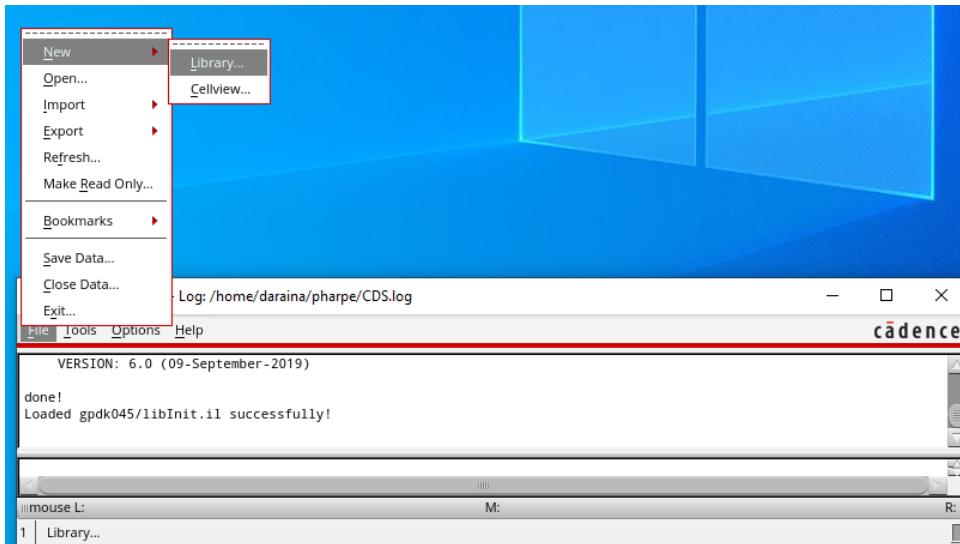
- In the library manager you can find several useful components in the libraries. The most important ones are:
- Library: analogLib
 - Contains ideal components, such as: resistor (res), capacitor (cap), ground terminal (gnd), voltage sources (vdc, vpulse, vpwl, vsource), current sources (idc), etc.
- Library: gpdk045
 - Contains NMOS/PMOS transistors, resistors, capacitors and other components with realistic models. Some of the most relevant components are:
 - NMOS transistors: nmos1v, nmos1v_lvt, nmos1v_hvt, nmos2v
 - PMOS transistors: pmos1v, pmos1v_lvt, pmos1v_hvt, pmos2v
 - The “1v” devices are *core* devices and only supposed to operate at max. 1.1V.
 - The “2v” devices are *IO* devices and can operate at max. 1.8V.
 - The addition of “lvt” or “hvt” means that the transistor has a lower or higher threshold voltage as compared to the baseline transistor.

Naming Conventions – Tips

- When you make a design in Cadence, you will typically make your own **library**, your own **schematics/symbols/layouts**, and you will need to assign **labels** and **pins** to some of your nets.
- Because some of the Cadence-related tools have problems with special symbols (like +, -, %, \$, #, etc.) or they might mix up capitals (ABC) versus lowercase (abc), **it is recommended that you use CAPITAL LETTERS and the “_” symbol only for library, schematic/symbol/layout, label and pin names.**

Making Your Own Library

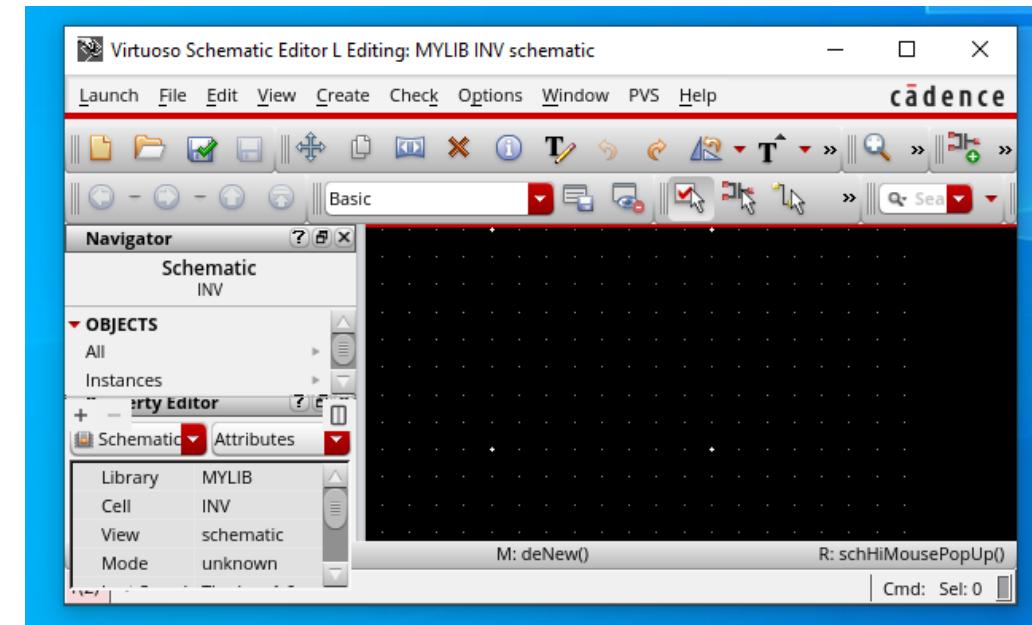
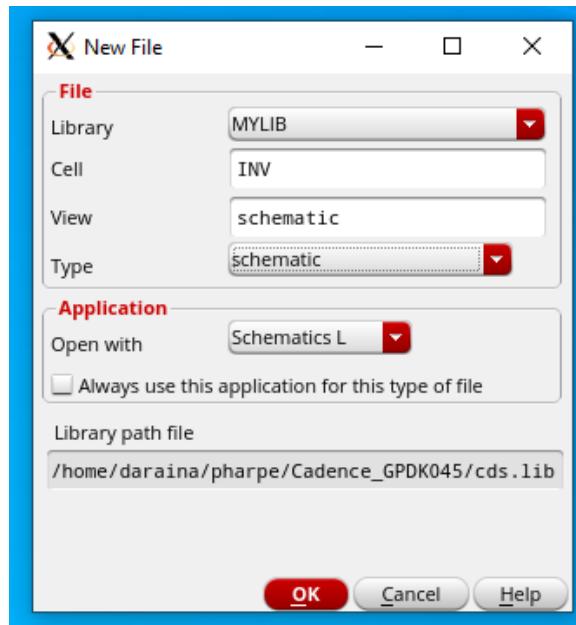
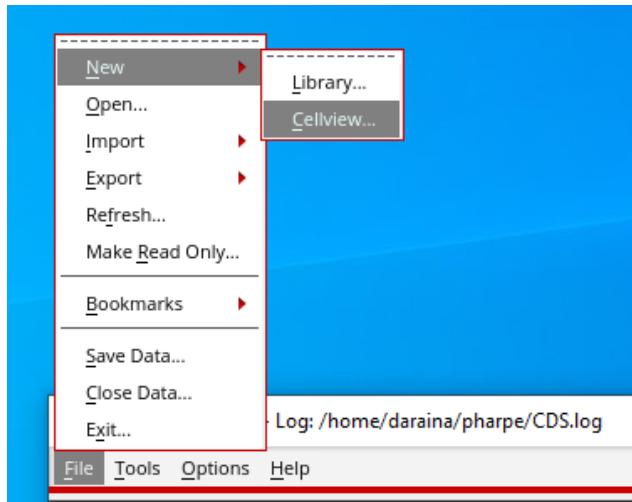
- To store your own designs, you first have to make a new library in which you can save all your schematics, layouts, symbols, etc. To make a library, choose the menu “File” → “New” → “Library” from the main window.
- Type a name for your new library and enable the option “Attach to...”
- In the next window, choose “GPKD045” as technology library



- You should now have a new library in the library manager

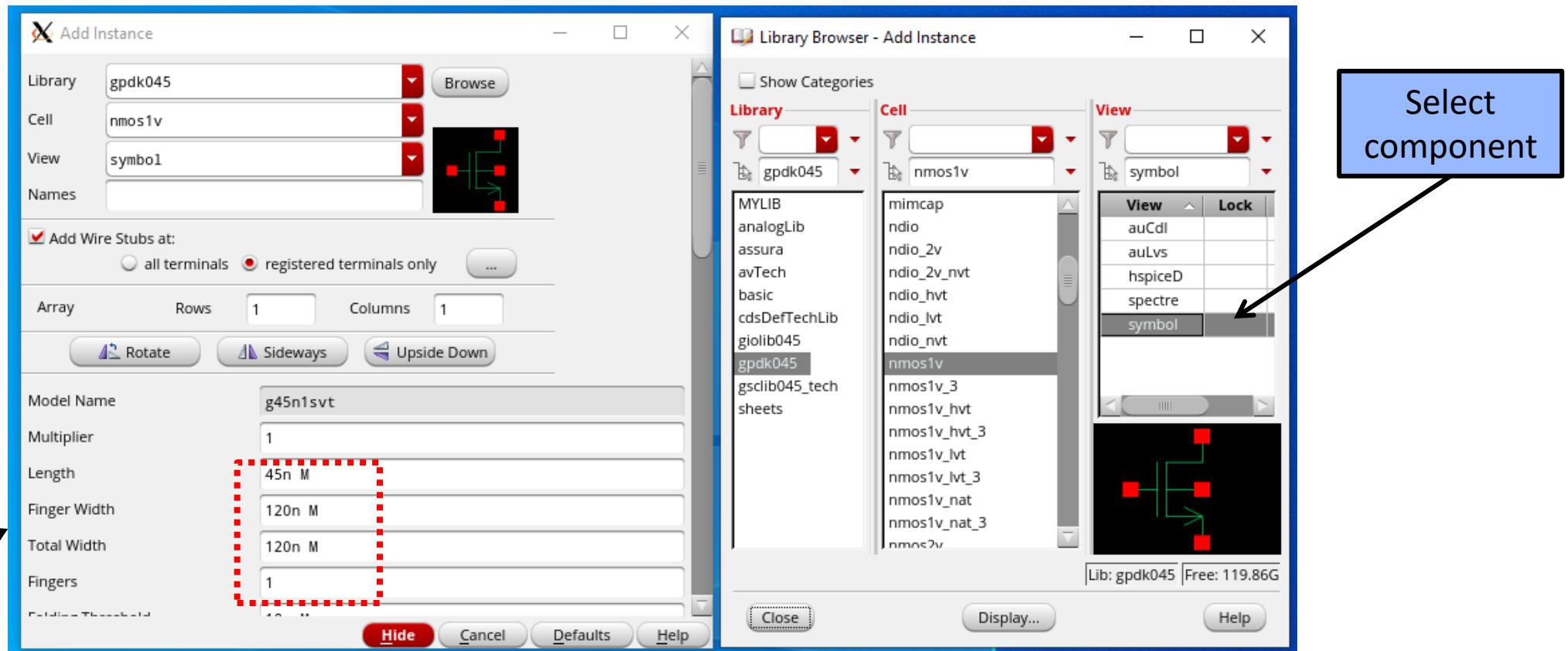
Starting a New Schematic

- Choose the option “File” → “New” → “Cellview” from the main window.
- Select your library, type a new name for the cell (“INV” in this example), and set the type to “schematic”
- You will now get a new empty window where you can draw your schematic



Inserting Components to Schematic

- Press the shortcut “i” in the schematic window to insert a new component. Use “browse” to find the desired component in the library. Select the component and the “symbol” view. You can change parameters (e.g. resistor value or transistor size) in the menu. Then place it in the schematic.

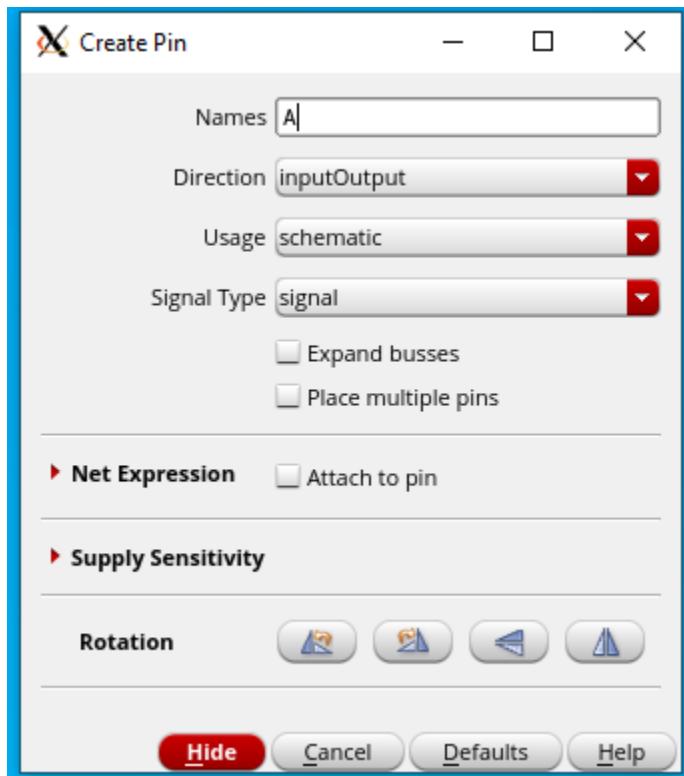


Choose
parameters

Select
component

Adding Pins to a Schematic

- Pins are used to define connections of a schematic to the outside world. Use the shortcut “p” to create a new pin. Type the desired name and set the direction to “inputOutput”, then place the new pin.



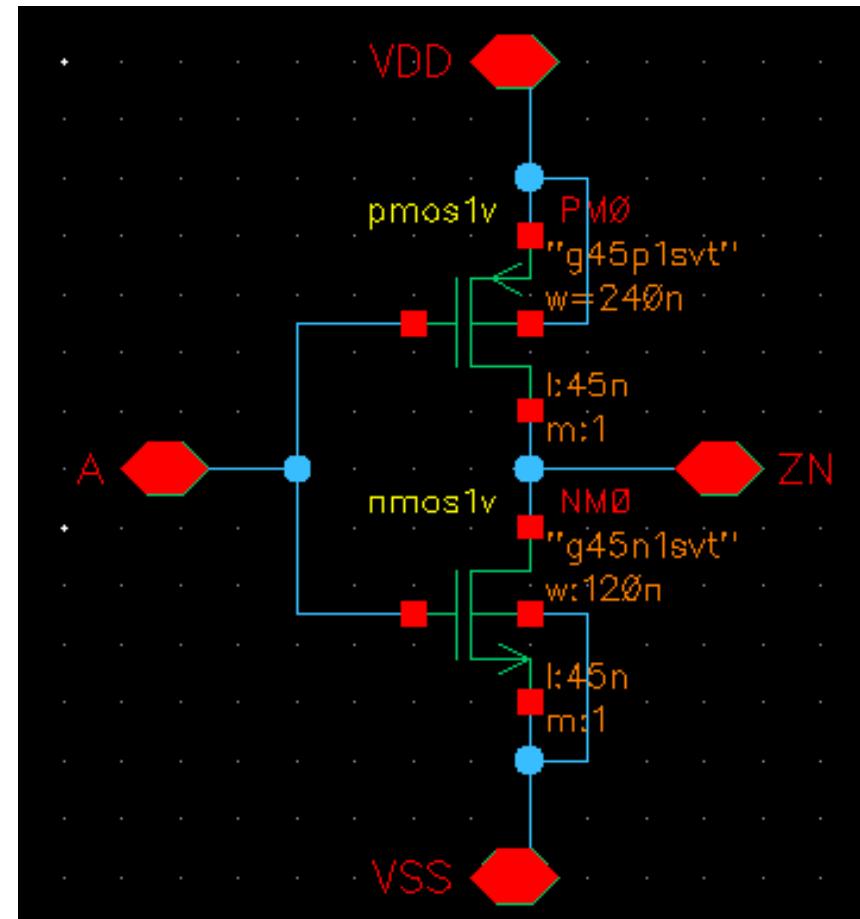
Shortcuts for Drawing a Schematic

- “f”: fit to screen, “[“ and ”]” to zoom out and in
- “i”: insert a component from the library
- “l”: add label to a net
- “p”: add a pin to a schematic
- “q”: edit properties of a component already placed in the schematic
- “w”: draw wire
- Shift + “x”: save schematic
- “e”: descend into selected sub-circuit
- Ctrl + “e”: return to top circuit

Create a Test Schematic

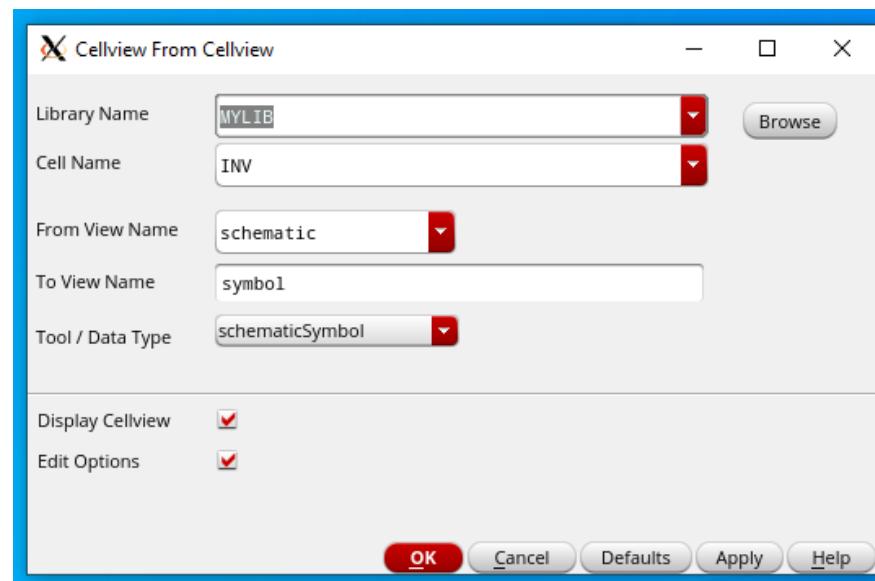
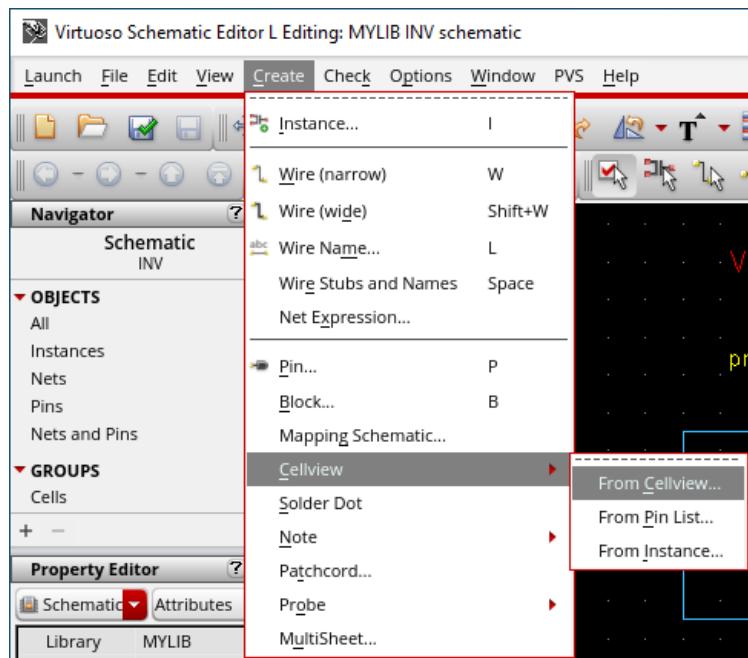
- With the explanation above, create the following schematic with the schematic name “INV”.
- Use 2 transistors:
 - nmos1v with $W = 120n$, $L = 45n$
 - pmos1v with $W = 240n$, $L = 45n$
- Use 4 pins (A, ZN, VDD, VSS)
- Save the schematic

Note: as long as you keep multiplier and fingers set to 1, the total width is equal to the finger width and you can set either value (the other one follows).



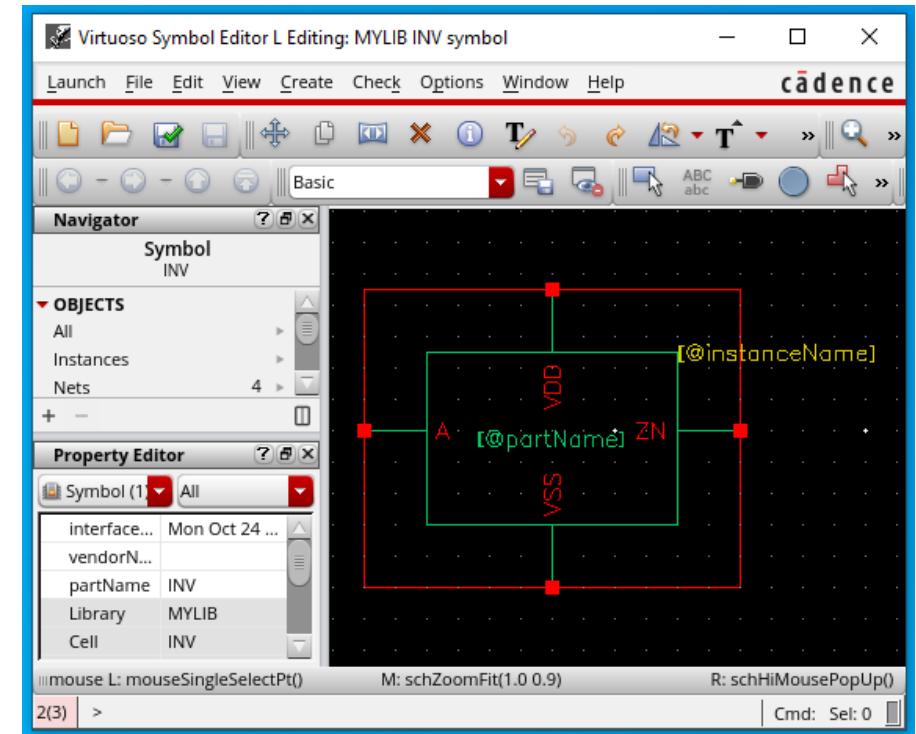
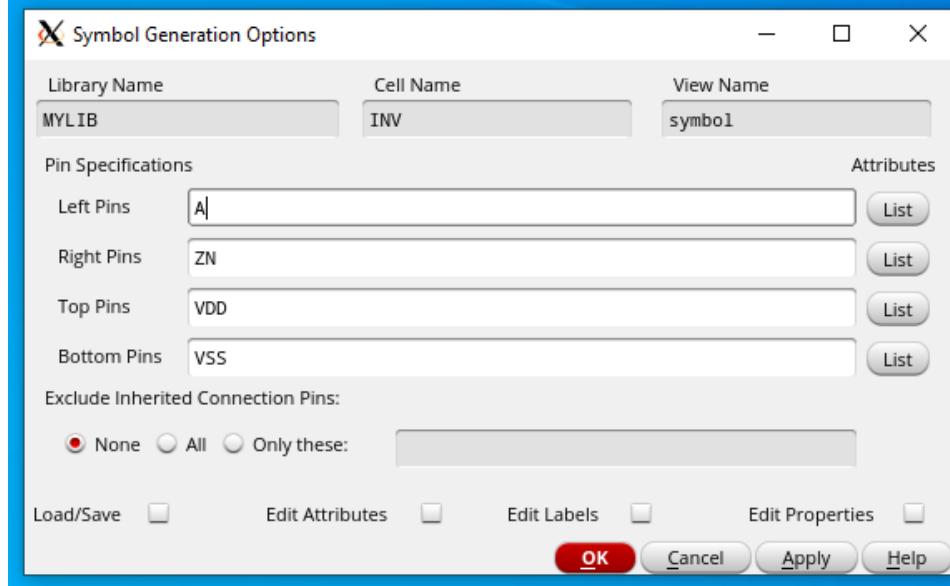
Create a New Symbol (1)

- To create a new symbol for your schematic, choose the menu “Create” → “Cellview” → “From Cellview” from the schematic window.
- In the window that appears, accept the default settings and press “OK”.



Create a New Symbol (2)

- In the next window you can change the Pin Specifications. It is a good habit to place inputs on the left, outputs on the right, and supplies on top and bottom. Make the shown adjustments and press “OK”.
- You should now get the symbol as shown.
- If you change the pins in your schematic later on, you also need to update your symbol. To do so, just follow the same procedure for creating a symbol, and indicate (when asked) that you want to modify the existing symbol.

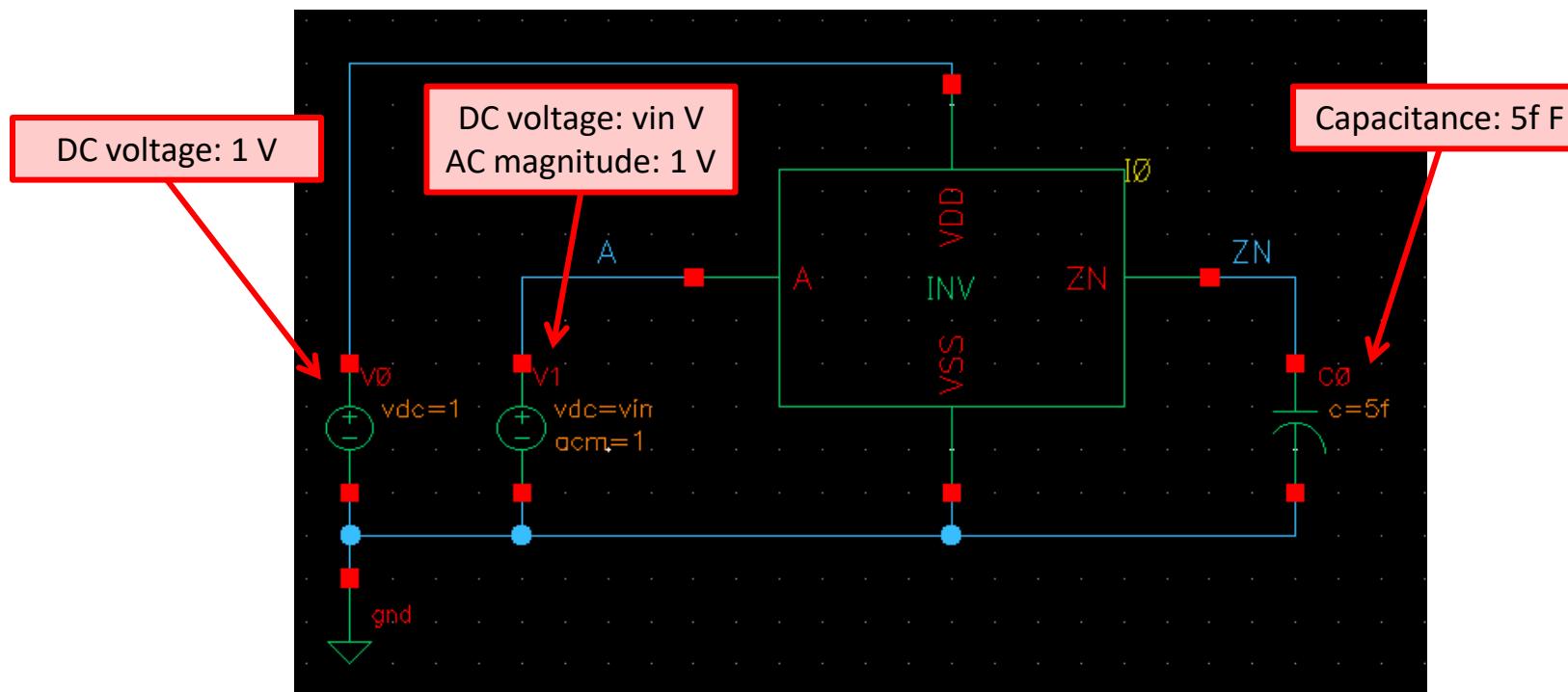


Create a New Symbol (3)

- The symbol (shown in the previous page) has two strange names (@partName and @instanceName). Do not change these! These are kind of variables and will be replaced by the actual name of the schematic (INV) and an identification number once you place a symbol in the schematic.
- Now you have a symbol view of your schematic, you can insert the symbol in a new schematic to build a hierarchy. In this way, if you need multiple inverters, you only need to draw the schematic once, and you simply place multiple symbols, referring to that same schematic.
- If you modify your schematic, any place where you use the symbol will automatically refer to the new schematic as well!

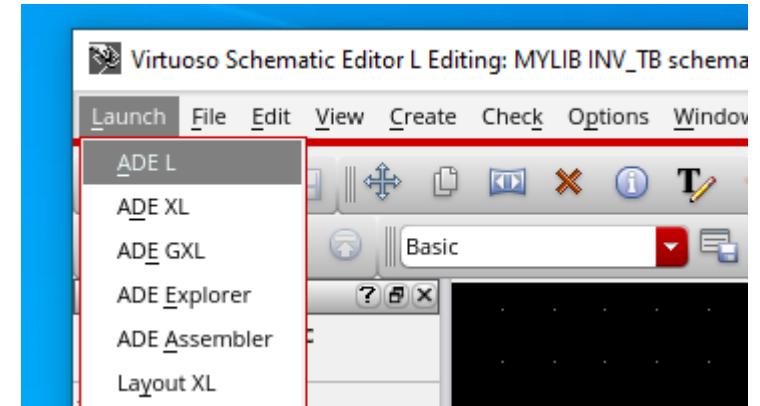
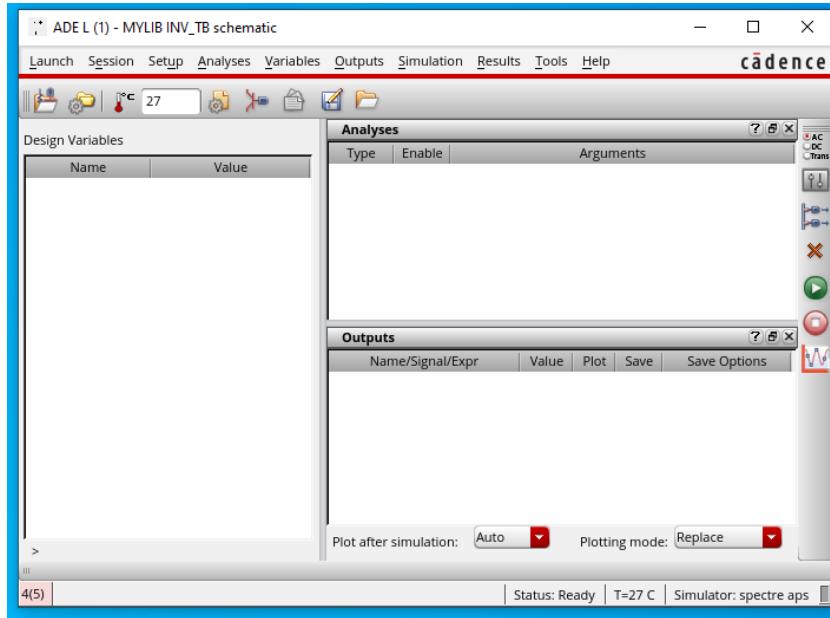
Creating a Test-Bench

- A test-bench is a schematic used to simulate a circuit. Make a new schematic with the name “INV_TB” in your library that will become the test-bench for your inverter.
- Create the schematic below using components “gnd”, “vdc” and “cap” from analogLib, and using the symbol of your “INV” from your own library.
- Set the indicated parameters, add labels “A” and “ZN”, and save the schematic.



Starting the Simulation Tool

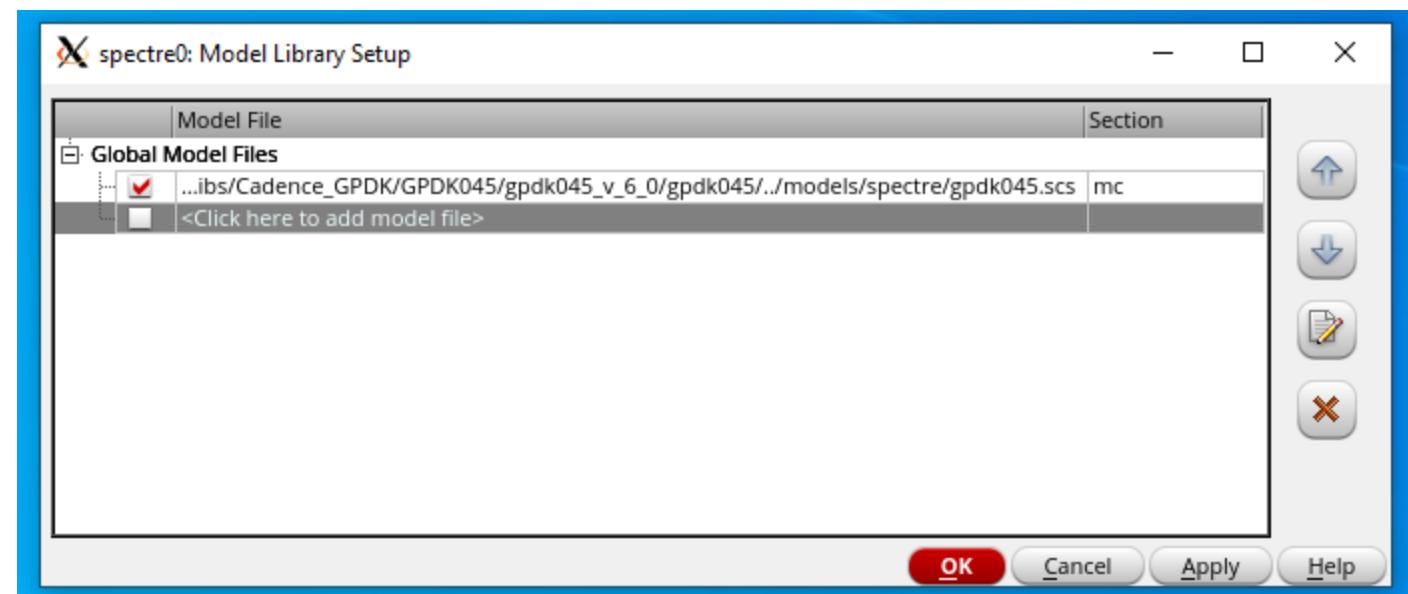
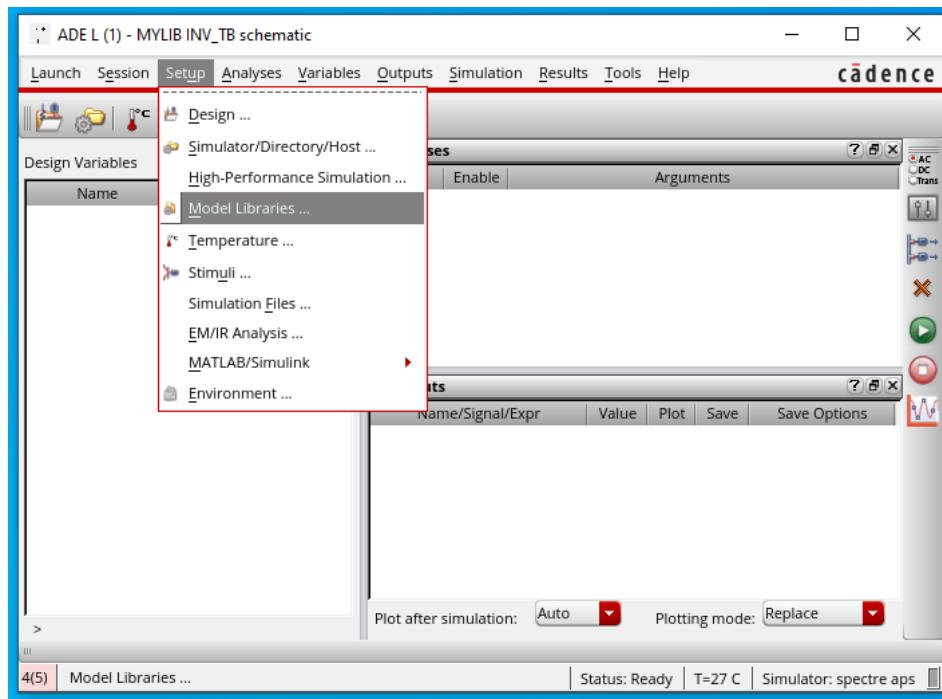
- In the window with the test-bench, use the menu “Launch” → “ADE L” to start the simulator. It will open a new window with the simulator tool.



- Before you can run simulations, you need to set up several things: the transistor models, the variables, the simulations that you want to perform, and the signals that you want to observe. This is explained next.

Setting up the Models

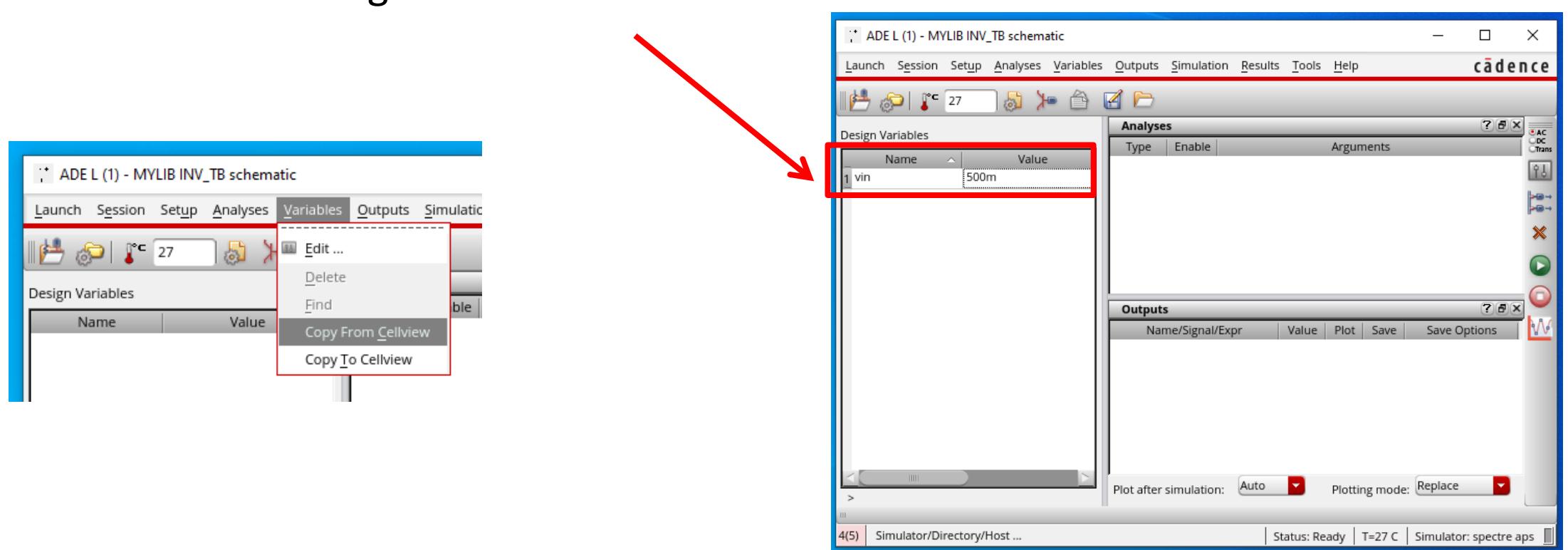
- Go to “Setup” → “Model libraries”. In the new window, the right model file and section should be selected. Usually, the default is OK. The model file should be set to something like this:
`/cadappl_sde/iclibs/Cadence_GPDK/GPDK045/gpdk045_v_6_0/gpdk045/..../models/spectre/gpdk045.scs` with section mc



- Press “OK” to close the window.

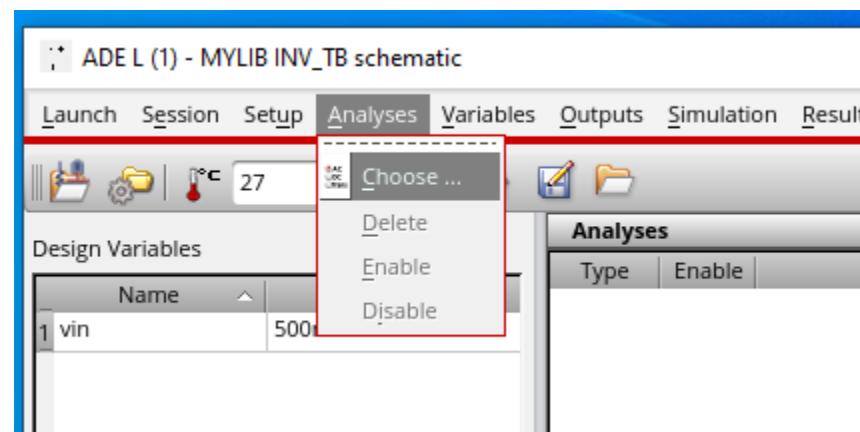
Setting up Variables

- In our schematic test-bench, we defined a variable “vin” (see slide 20). Click the menu “Variables” → “Copy from cellview” to copy it to the simulator. It now appears in the simulator window.
- Double click it to change the value and set it to 0.5V



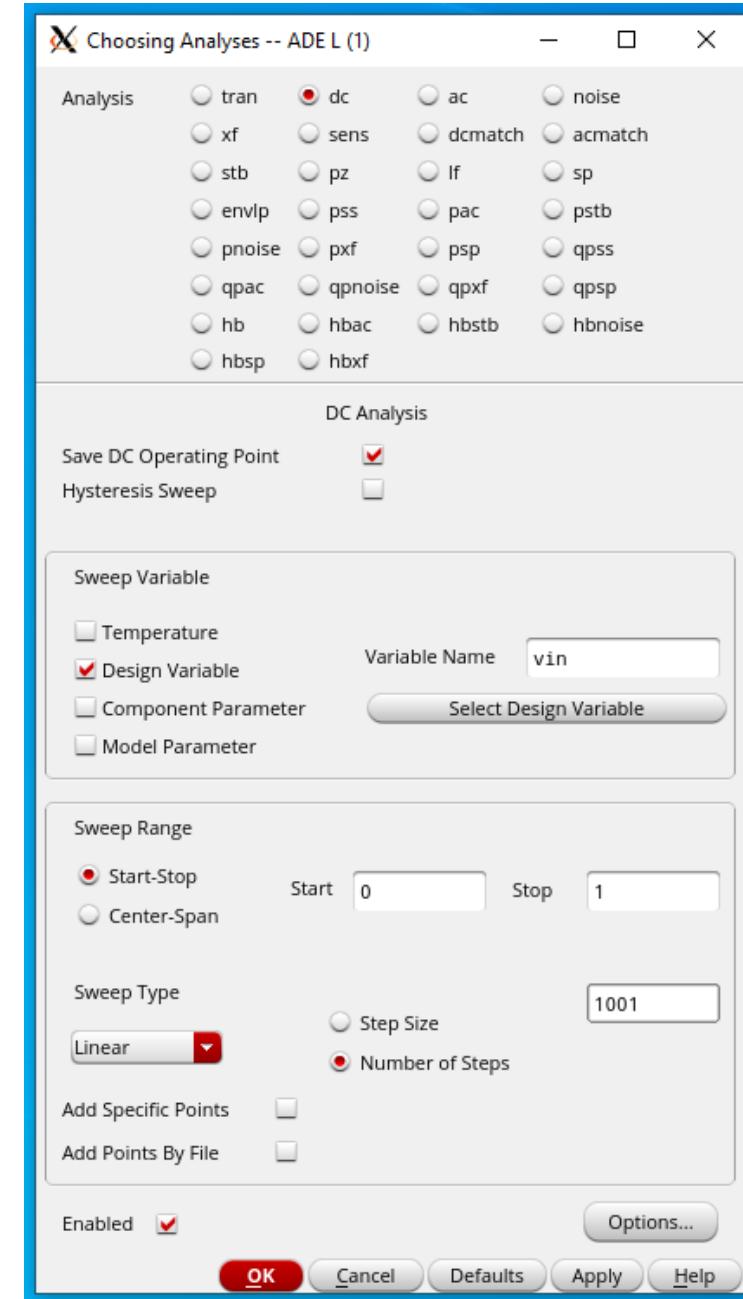
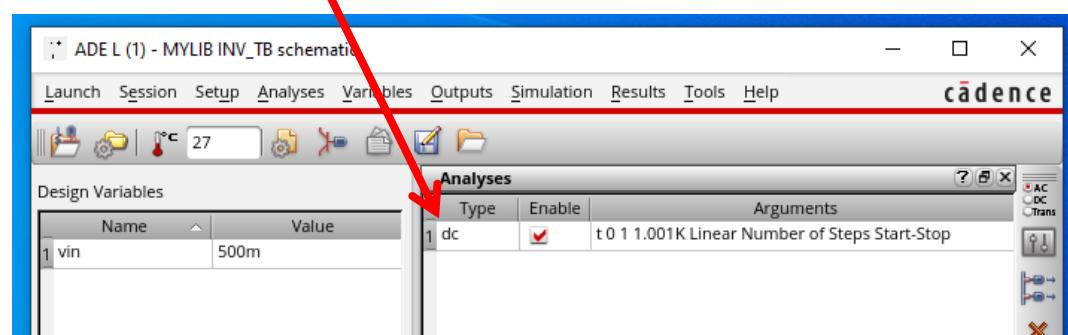
Defining Simulations

- Many types of simulations can be performed. To start, we use a DC and an AC simulation. Other basic simulations are for instance transient and noise simulations. To define a new simulation, use the menu “Analyses” → “Choose”



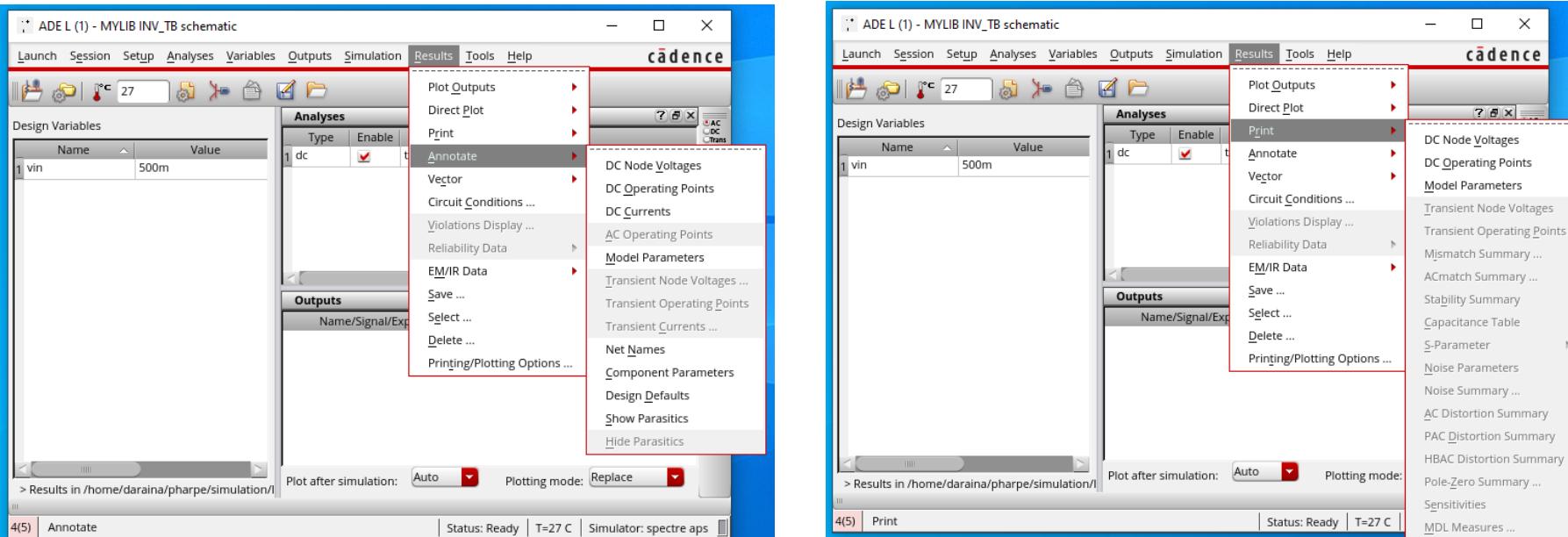
DC Simulation (1)

- Fill out the following settings to define a DC simulation.
- In this case, we will sweep the input voltage “vin” from 0 to 1V, in this way, we can obtain the transfer curve (“vout” versus “vin”) of the inverter.
- Note that the variable vin will be swept from 0 to 1V, so this simulation will OVERRULE the value vin = 0.5V that we defined earlier!
- Press “OK” to close the window, the DC simulation now appears in the list:



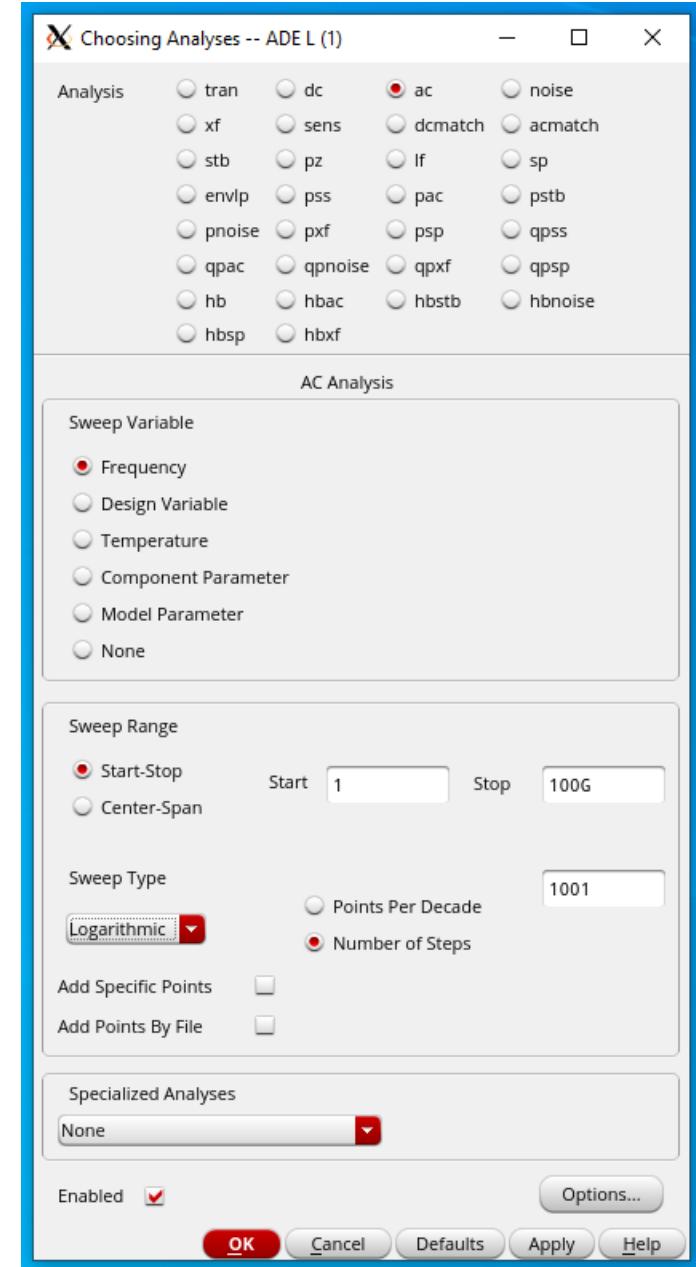
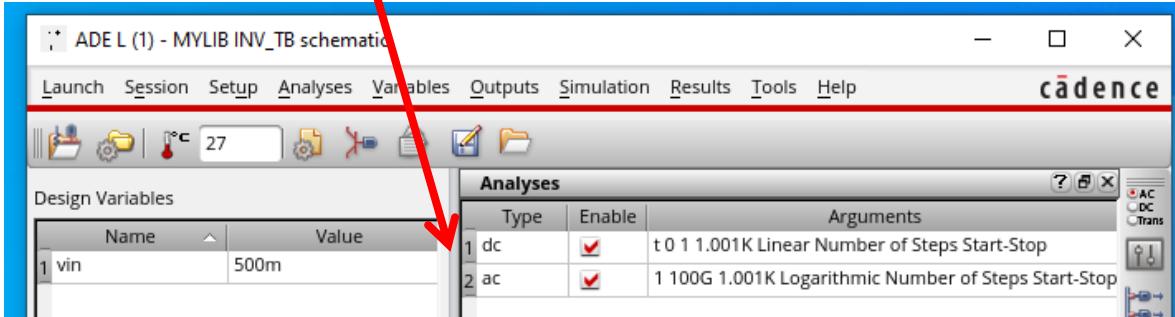
DC Simulation (2)

- If you have ticked the box “Save DC Operating Point” as shown on the previous slide, you can do a few nice things after you ran the simulation:
 - 1) In the “Results → Annotate” menu, you can use “DC Node voltages” and “DC operating points” to show the DC voltages at each node in the schematic, and to show some DC details of each device (like V_{GS} , I_{DS} , etc.). The information will show up in your schematic.
 - 2) In the “Results → Print” menu, you can click “DC operating points” or “Model Parameters”, and then click on a transistor in your schematic, to show a long list of details about that device in its DC point, like the g_m , capacitances, etc.



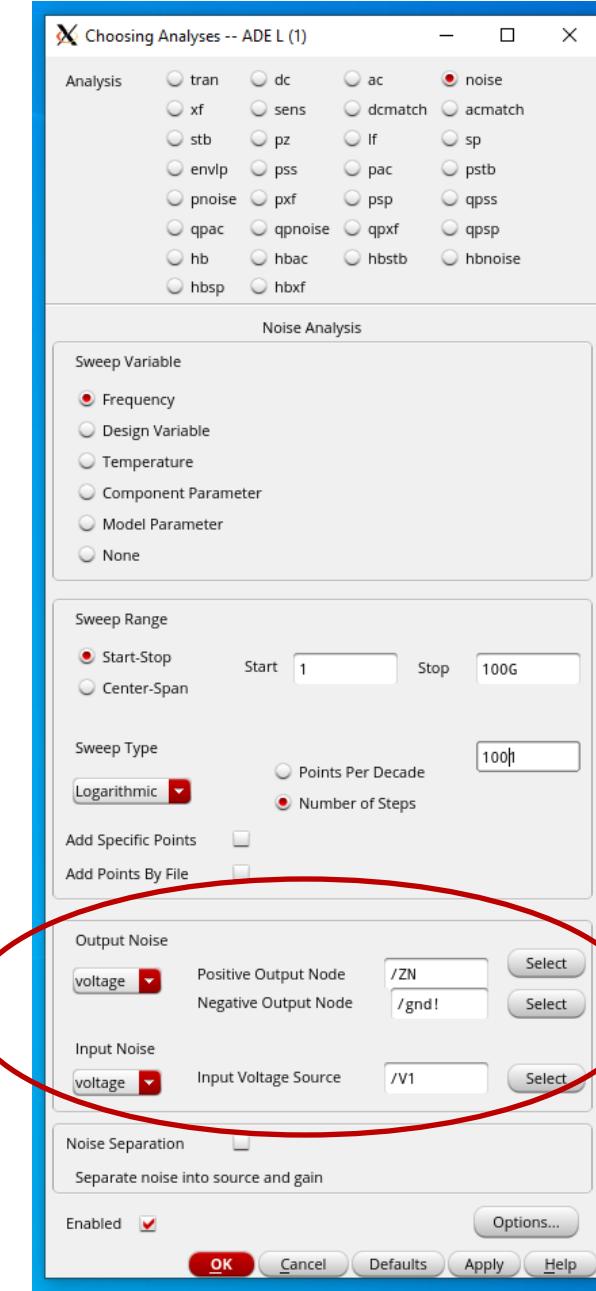
AC Simulation

- Use the menu “Analyses” → “Choose” again, and now use the following settings to define an AC simulation.
- In this case, we will sweep the frequency of the input signal to see the gain and bandwidth of the inverter at a bias point of $v_{in} = 0.5V$
- Note that the input source (slide 20) is now a combination (superposition) of:
 - A DC value of $v_{in} = 0.5V$ (the value set before)
 - An AC amplitude of 1V
- Press “OK” to close the window, the AC simulation now appears in the list as well:



Noise Simulation (1)

- Use the menu “Analyses” → “Choose”.
- Use the following settings to define a noise simulation.
- In this example, we will sweep the frequency from 1Hz to 100GHz with 1001 steps to see the noise in that bandwidth.
- To be able to see the output-referred noise and input-referred noise of the circuit, you need to define where the output of your circuit is, and where the input is.
 - The output is the difference between two nodes. If your circuit is single-ended, you take the output vs ground
 - The input is indicated by a voltage source that connects to the input of your circuit.
 - You can either type the names of the nets and sources or select them from your schematic with the “Select” buttons.
 - If you use the schematic from slide 20, the positive output node would be named “/ZN”, the negative output node “/gnd!”, and the input source would be “/V1”. **Adjust accordingly if you gave different names to the nets, or if the voltage source has a different number.**

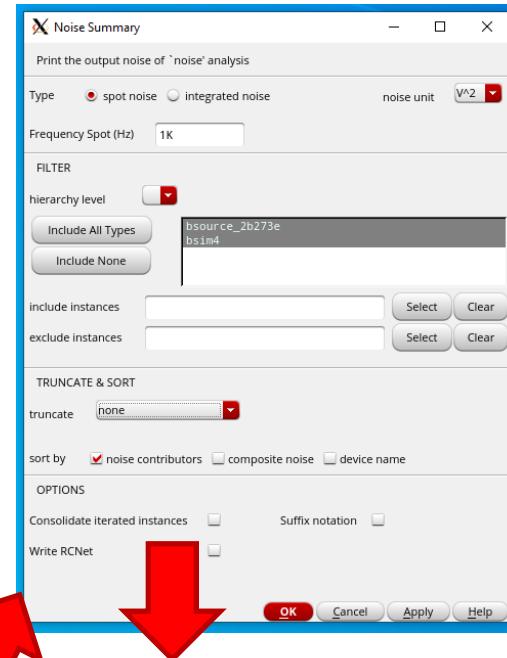
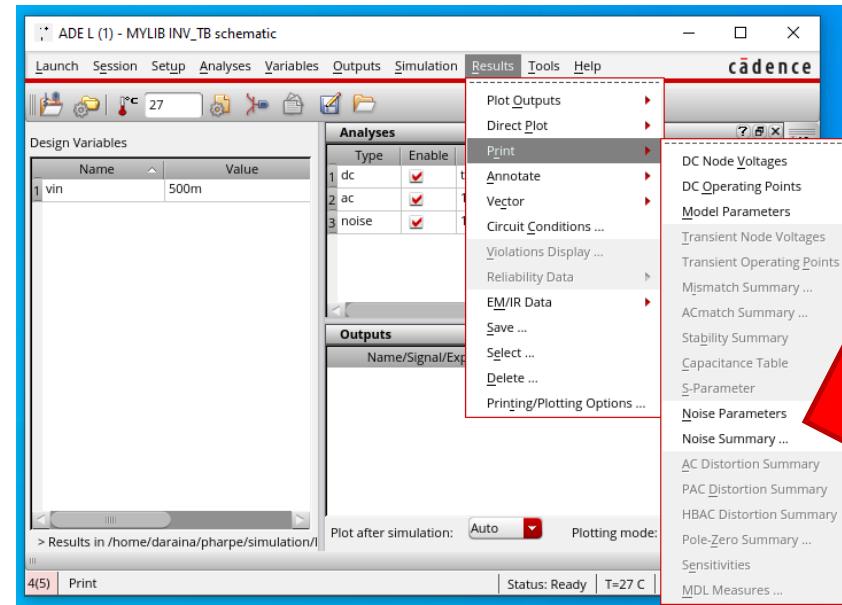
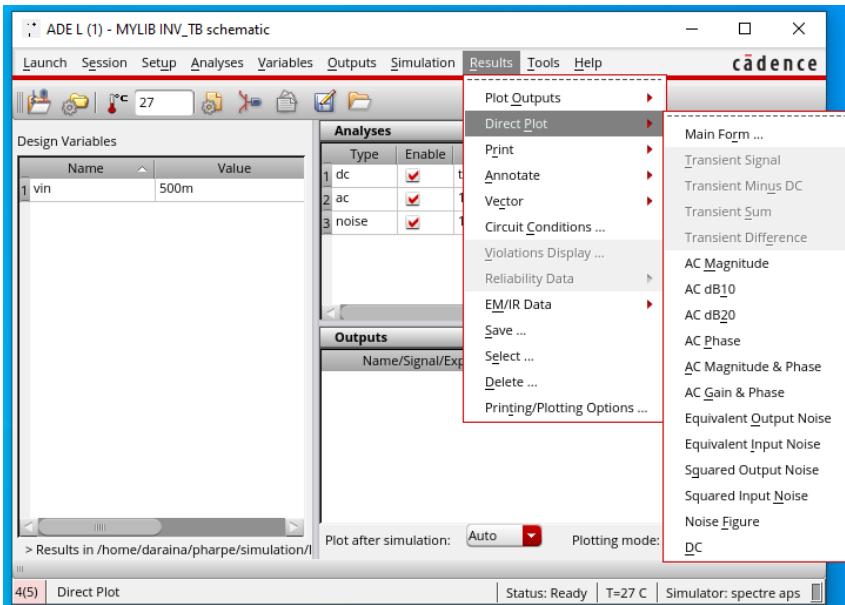


Noise Simulation (2)

- After running the noise simulation, you can do a few things:

1) In the “Results → Direct plot” menu, you can click equivalent input/output noise and squared input/output noise to see the either the amplitude or power spectral density of the noise, referred to either the input or output. To get the integrated noise power over a certain bandwidth, you still need to integrate the squared noise over this BW.

2) You can click on the menu “Results → Print → Noise summary”, which shows a pop-up window. If you take there for instance spot noise at 1k, it will show the noise PSD at 1kHz in a list, where the individual contributions from each component and for each type of noise are specified.

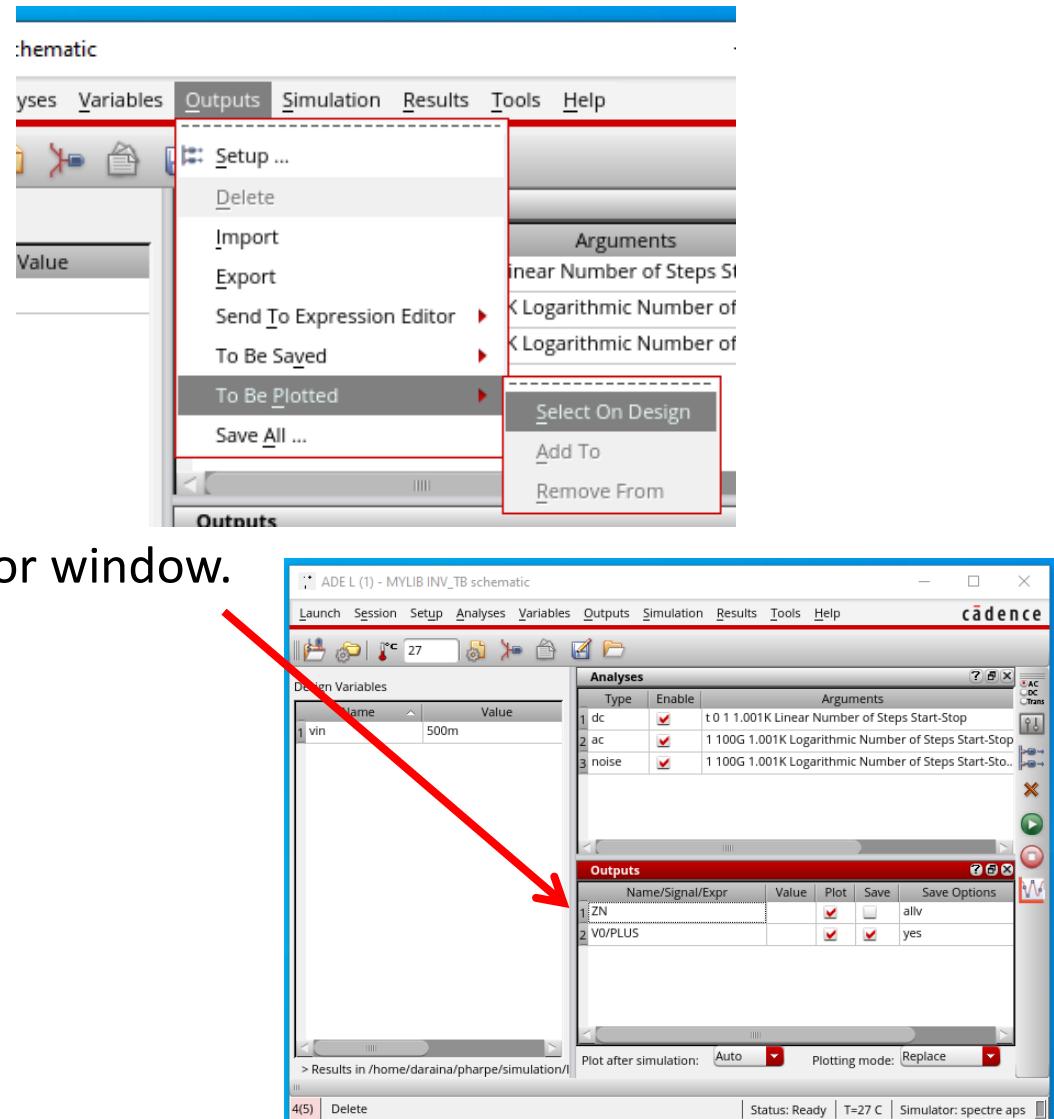
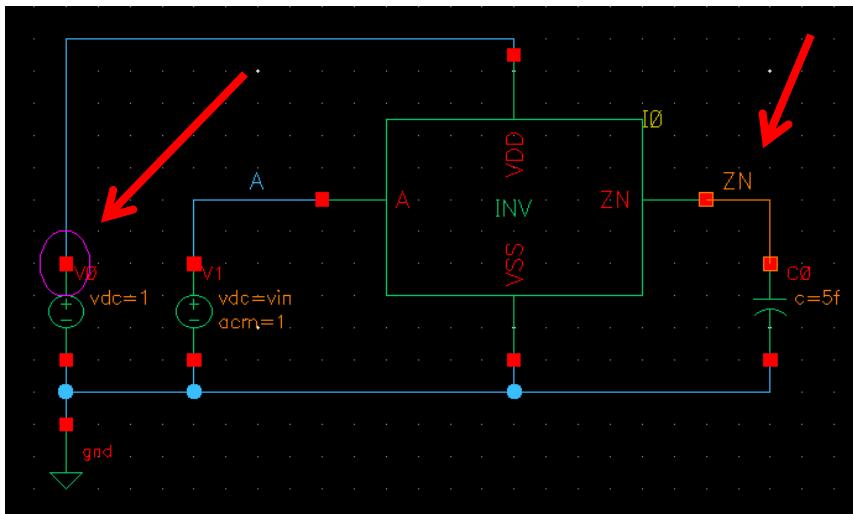


Device	Param	Noise Contribution	% of Total
/10/NMO	fn	6.06526e-09	92.60
/10/PMO	fn	4.24538e-10	7.40
/10/PMO	id	1.31361e-13	0.00
/10/NMO	id	9.4534e-14	0.00
10.NMO_xrg.r1	thermal_noise	4.54457e-17	0.00
10/NMO	rs	3.90534e-17	0.00
10.NMO_xrg.r1	thermal_noise	1.95684e-17	0.00
10/PMO	rs	1.37822e-17	0.00
10/PMO	igd	5.29483e-20	0.00
10/NMO	rd	4.27135e-20	0.00
10/PMO	rd	3.89069e-20	0.00
10/NMO	igd	6.79794e-21	0.00
10/NMO	is	3.48624e-26	0.00
10/NMO	is	4.41809e-27	0.00
10/PMO	igb	1.47205e-31	0.00
10/NMO	igb	3.04929e-35	0.00
10/PMO	rphs	0	0.00
10,NMO_xrg.r1	flicker_noise	0	0.00
10/PMO	rgb1	0	0.00
10,PMO_xrg.r1	flicker_noise	0	0.00
10/PMO	rbpd	0	0.00
10/NMO	rbpd	0	0.00
10/NMO	rbhp	0	0.00
10/NMO	rphs	0	0.00
10/NMO	rphd	0	0.00
10/NMO	rgb1	0	0.00
10/PMO	rbpb	0	0.00

Spot Noise Summary (in V²/Hz) at 1K Hz Sorted By Noise Contributors
Total Summarized Noise = 6.55102e-09
Total Netted Deferred Noise = 1.8802e-11

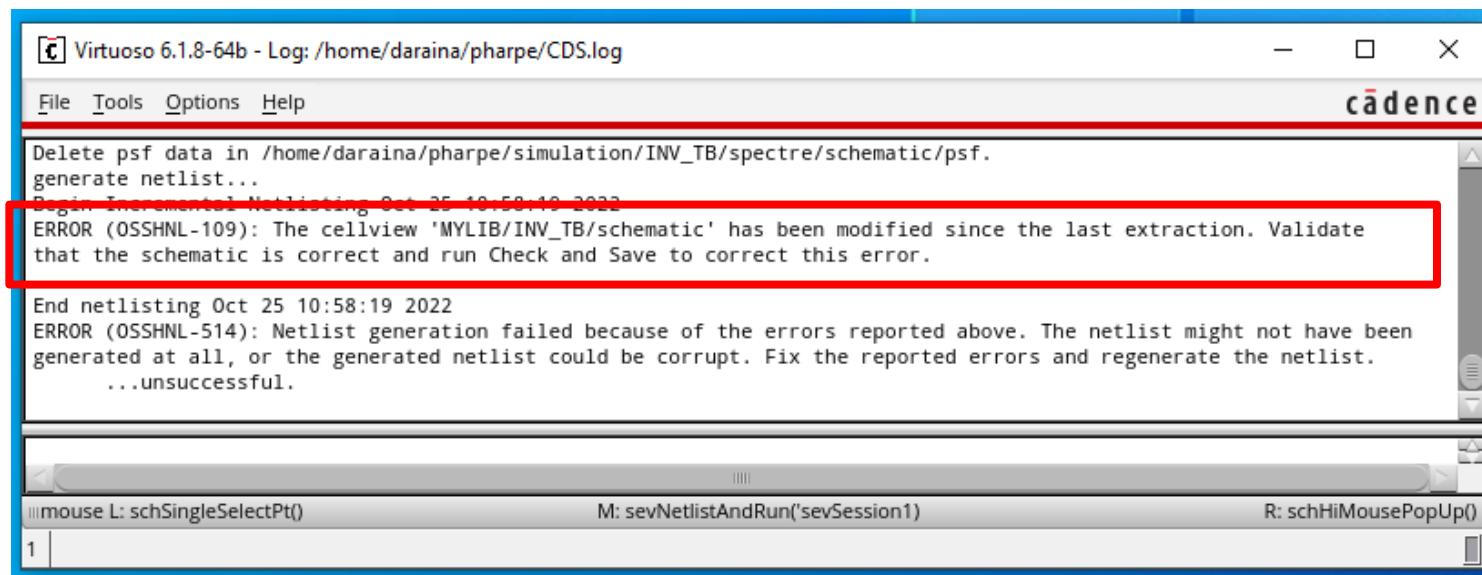
Defining Signals to be Observed

- To define which signals you like to see during your simulation, use the menu “Outputs” → “To be plotted” → “Select on design”
- Now go to the schematic and click on:
 - Net “ZN” to show the output voltage
 - The terminal of the supply to see the supply current
- Then press the “ESC” button to stop selecting signals.
The two selected signals are now shown in the simulator window.



Running Simulations

- Finally you can run the simulation! Choose “Simulation” → “Netlist and Run” or use the green button
- If it doesn’t work, check the main window for error messages, for instance in the example below, we have to “save” the schematic of “inv” again before being able to run the simulation

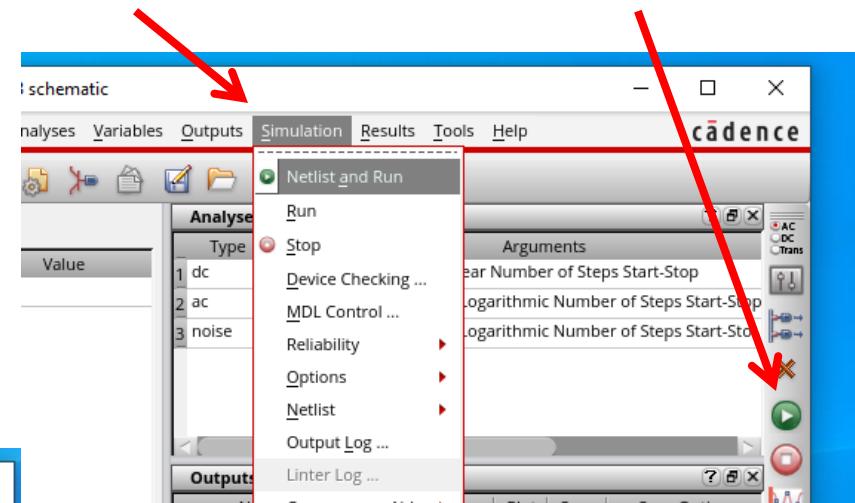


Virtuoso 6.1.8-64b - Log: /home/daraina/pharpe/CDS.log

```
Delete psf data in /home/daraina/pharpe/simulation/INV_TB/spectre/schematic/psf.  
generate netlist...  
Begin Incremental Netlisting Oct 25 10:58:10 2022  
ERROR (OSSHNL-109): The cellview 'MYLIB/INV_TB/schematic' has been modified since the last extraction. Validate  
that the schematic is correct and run Check and Save to correct this error.  
  
End netlisting Oct 25 10:58:19 2022  
ERROR (OSSHNL-514): Netlist generation failed because of the errors reported above. The netlist might not have been  
generated at all, or the generated netlist could be corrupt. Fix the reported errors and regenerate the netlist.  
...unsuccessful.
```

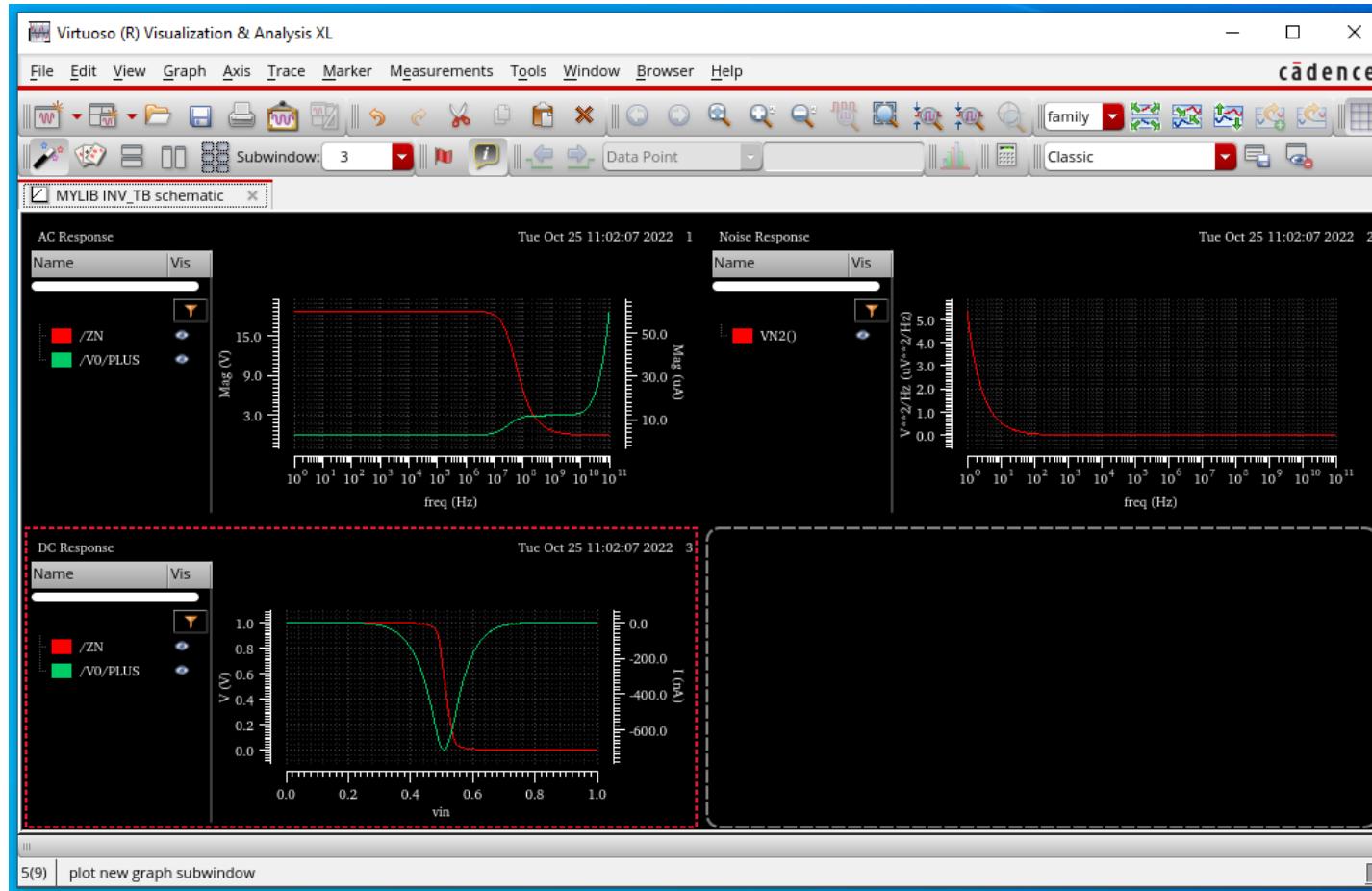
mouse L: schSingleSelectPt() M: sevNetlistAndRun('sevSession1') R: schHiMousePopUp()

1



Observing Results (1)

- Once the simulation works, you will see the outputs of DC, AC, and noise simulations displayed.
 - Note: you can enable/disable individual simulations, or just run all of them at the same time.



Observing Results (2)

- If you saved the DC operating point, you can check if a particular transistor is in weak inversion or not.

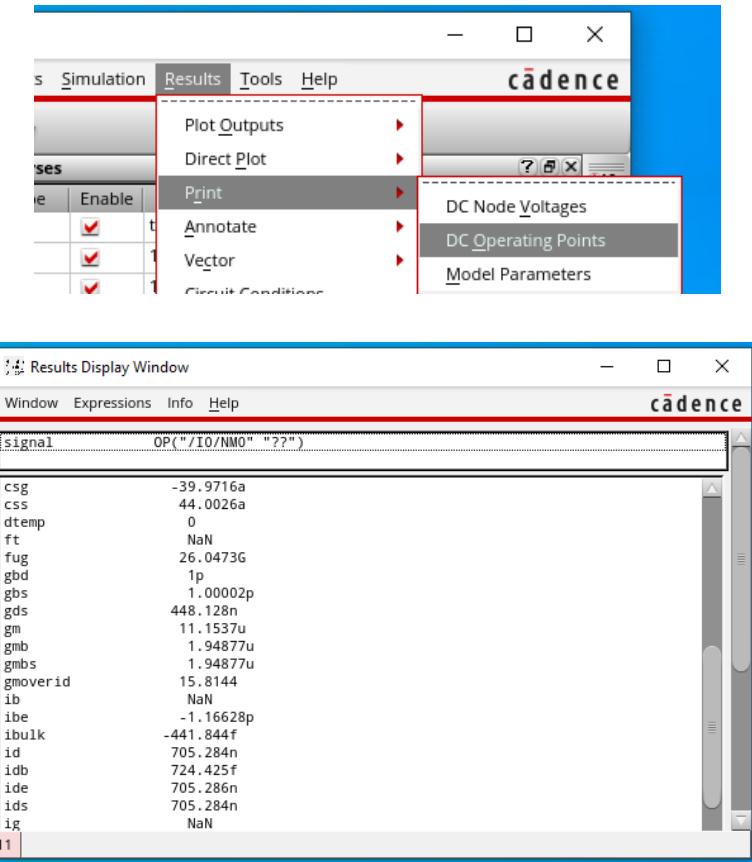
In the “Results → Print” menu, choose “DC operating points” and then click on the transistor in your schematic about which you’d like to get information.

It will print a list of parameters, useful ones are for instance:

- gm
- gmoverid (g_m / I_D)
- vgs
- vth

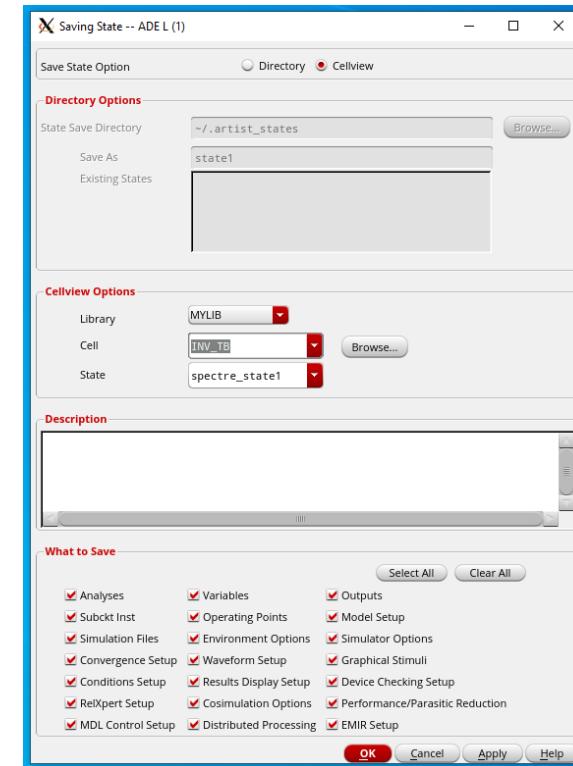
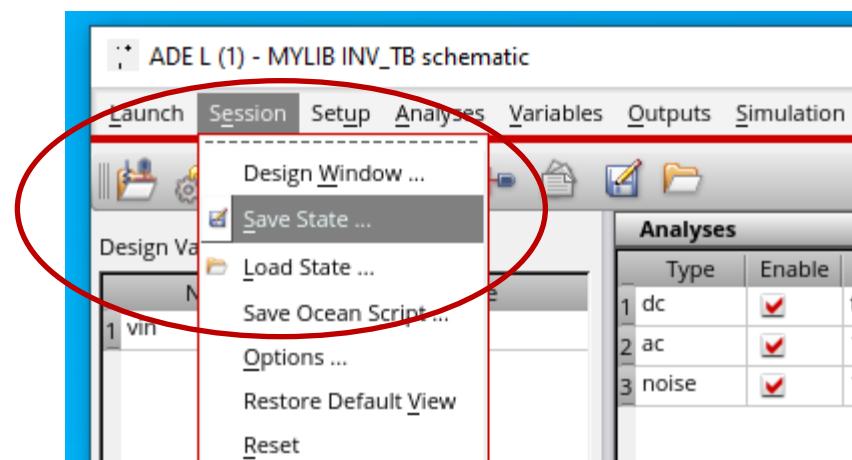
With gmoverid or with a comparison of vgs and vth you can estimate in which biasing range your transistor is.

Note: checking gmoverid is usually more accurate.



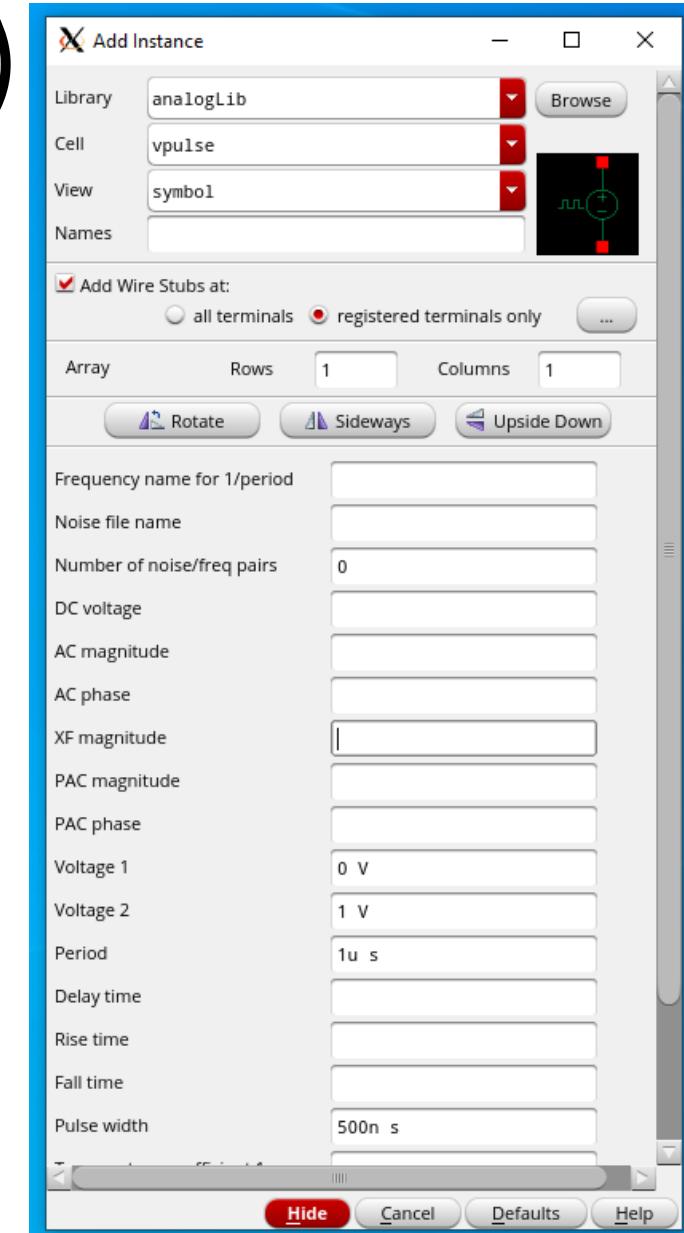
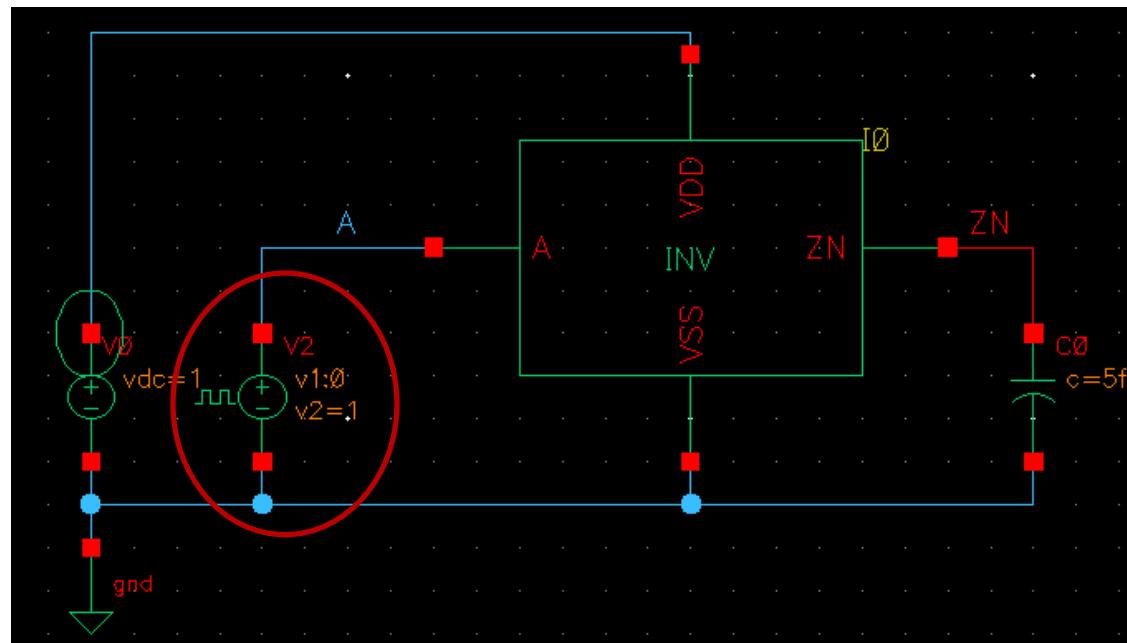
Saving and Loading State

- To avoid that you need to set all simulator details (models, variables, simulations, outputs) again every time you start Cadence, you can save and reload the settings in a “state”. To do so, use the options “Save state” and “Load state” from the “Session” menu. You can choose the name of the state (by default it is named “spectre_state1”). In this way you can save multiple states for a design.



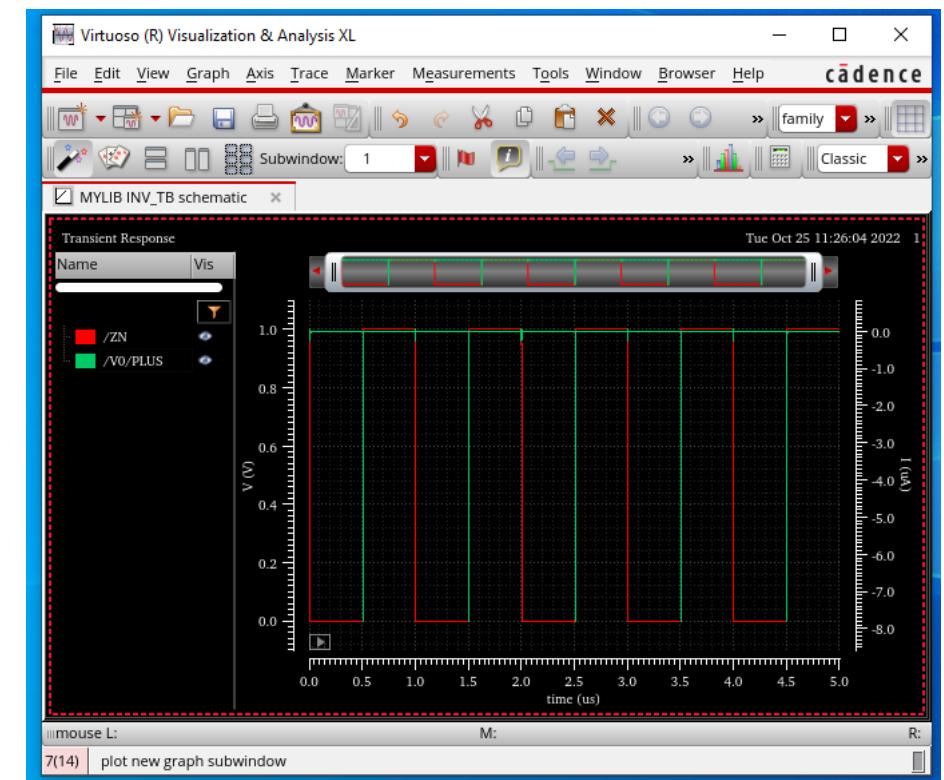
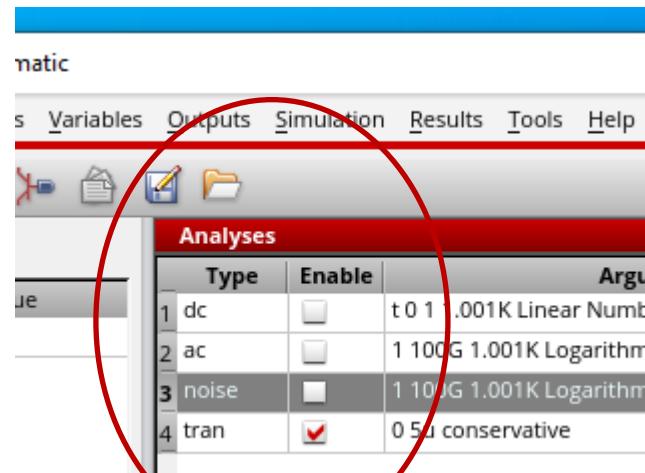
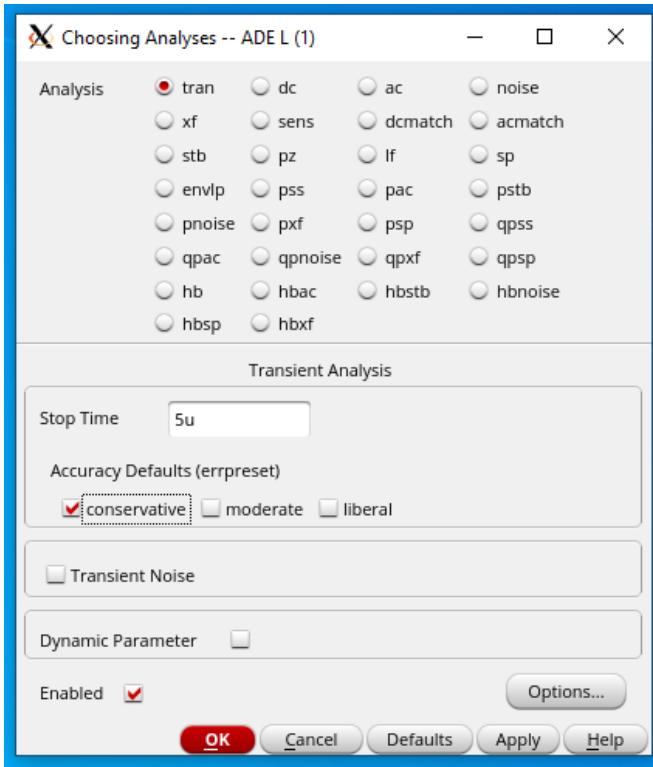
Transient Simulation (1)

- Another useful simulation for an inverter is a transient simulation. To do this, we should first change our input signal to (for instance) a square wave signal, as a DC signal is not very interesting in this case.
- Open your test-bench and replace the input signal source by a “vpulse” from analogLib. Set the properties as shown to make a 1MHz square wave.
- Save the test-bench again.



Transient Simulation (2)

- Define a transient simulation with the shown settings. Remove the AC, DC, and noise simulations or disable them, as you currently don't want to run them.
- Run the new simulation. You should get the results as shown.

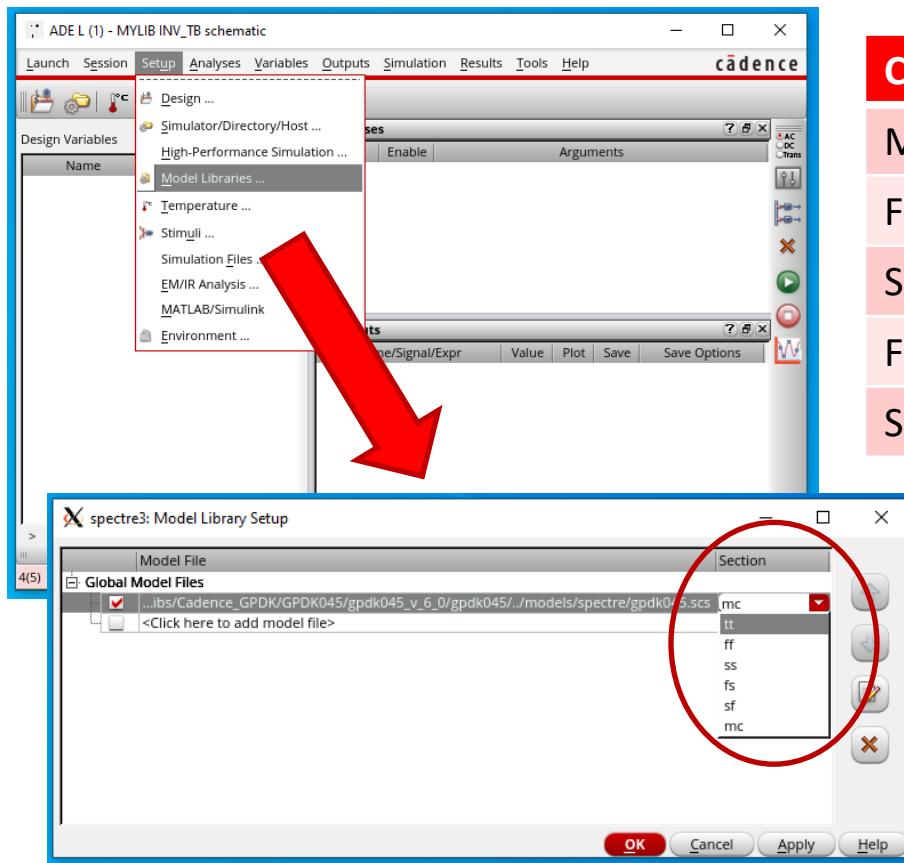


Advanced Tests: PVT Variations

- PVT variations are variations in:
 - P – Process: due to manufacturing tolerances and variations, the parameters of your circuit may shift. The worst-case variations are captured in the **process corners**.
 - V – Voltage: the actual supply voltage of your circuit may vary, e.g.: due to a non-ideal source like a battery, or due to voltage drops in the layout.
 - T – Temperature: the environmental temperature of your IC might change dependent on where/when you use it. Also, the IC could cause local heating if you dissipate power.
- PVT simulations are usually performed to check the robustness of a design against such variations.
 - Ideally, we should test all combinations of P+V+T variations simultaneously.
 - In practice, we often compromise and test P, V, and T separately, or we test a few cases where the combination of P, V, T is set to an extreme combination.

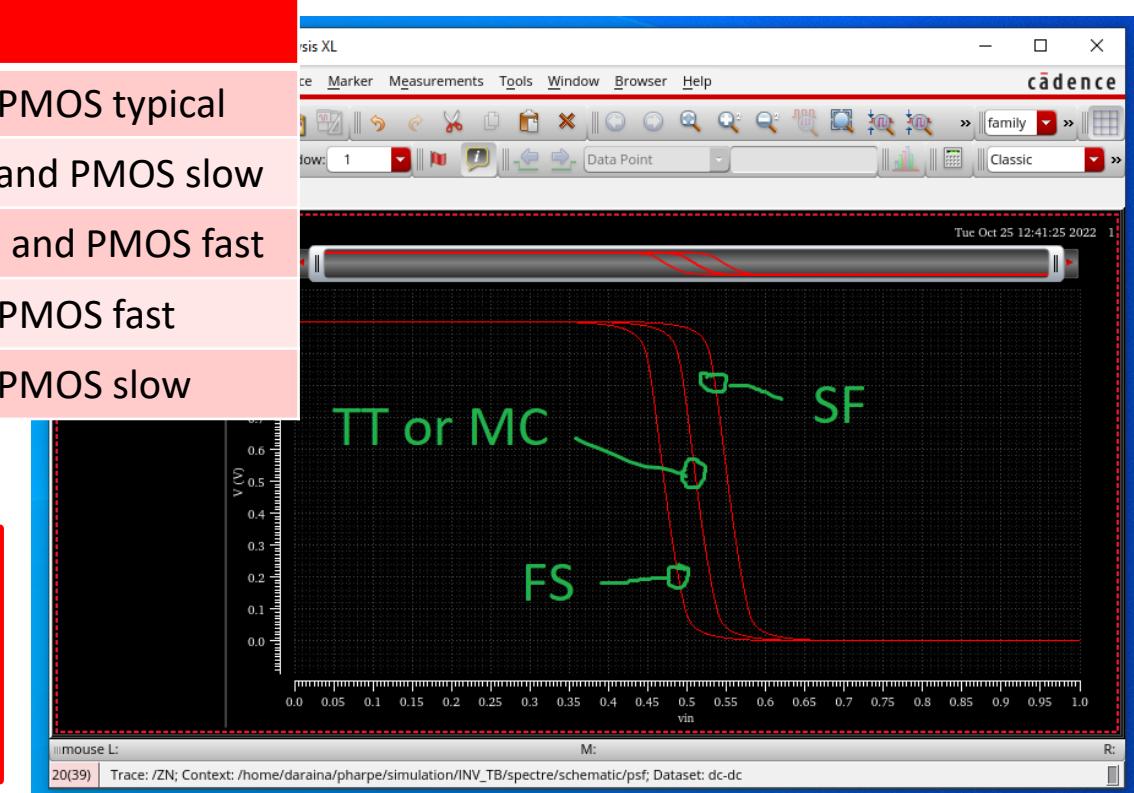
Process Corner Simulations (P)

- To perform process corner simulations, modify the “Section” of the model from the drop-down menu. The corners of NMOS and PMOS can be set to **Typical**, **Fast**, and **Slow**, each giving a combination of parameter shifts. E.g.: in slow mode, the threshold will go up (and other parameters change as well).



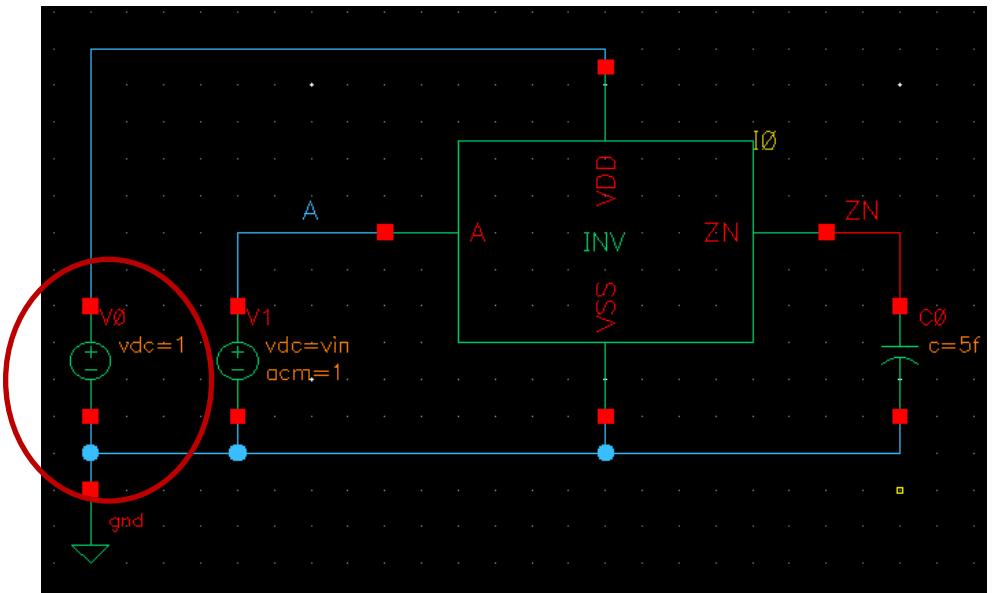
Corner	Meaning
MC or TT	NMOS and PMOS typical
FS	NMOS fast and PMOS slow
SF	NMOS slow and PMOS fast
FF	NMOS and PMOS fast
SS	NMOS and PMOS slow

Remember to set the model back to default afterwards!



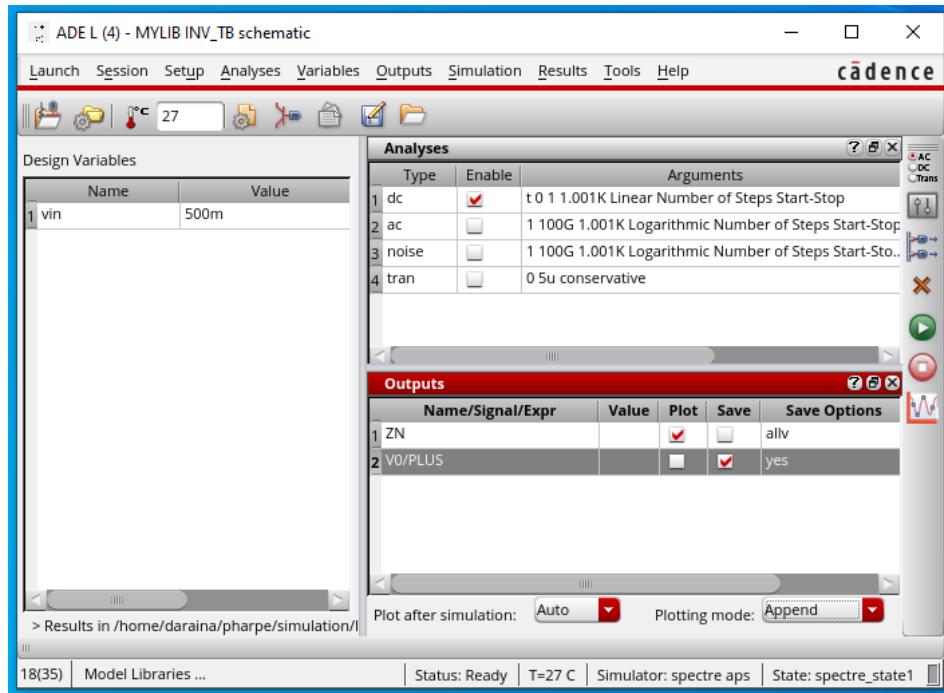
Supply Variation Simulations (V)

- To simulate supply voltage variations (or variations in biasing voltages/currents), we need to estimate how much variation is realistic and simulate accordingly.
 - As a rule of thumb (example), we could simulate at -10% and +10% of the nominal supply.
 - If the circuit performance changes drastically, it might imply it is not sufficiently robust.
 - To simulate, simply change the voltage source level in your schematic (or in ADE L if you defined the supply voltage with a variable).
 - Example: change V_0 (the supply source) with -10% and +10%, so:
 - Nominal case: 1V
 - Minimum VDD corner: 0.9V
 - Maximum VDD corner: 1.1V
- 



Temperature Corner Simulations (T)

- To simulate the effect of temperature changes, you can simply define the environmental temperature in ADE L before you run the simulation. We usually check nominal, min. and max. T.
 - Remember to set the temperature back to default (27°C) after running the temperature checks!
- Limitation: the same temperature setting is applied to all devices. Therefore, this simulation cannot check for local on-chip variations, nor does it include effects of self-heating due to power dissipation.



Grade *	Temperature range
Commercial	0°C to 70°C
Industrial	-40°C to 85°C
Automotive	-40°C to 125°C

* Different manufacturers specify different temperature ranges, so this is an example.

Even if you don't expect temperature variations in your application (e.g.: an IC for an implantable device), it is still good habit to check the impact of temperature variations to get an idea of the circuit robustness

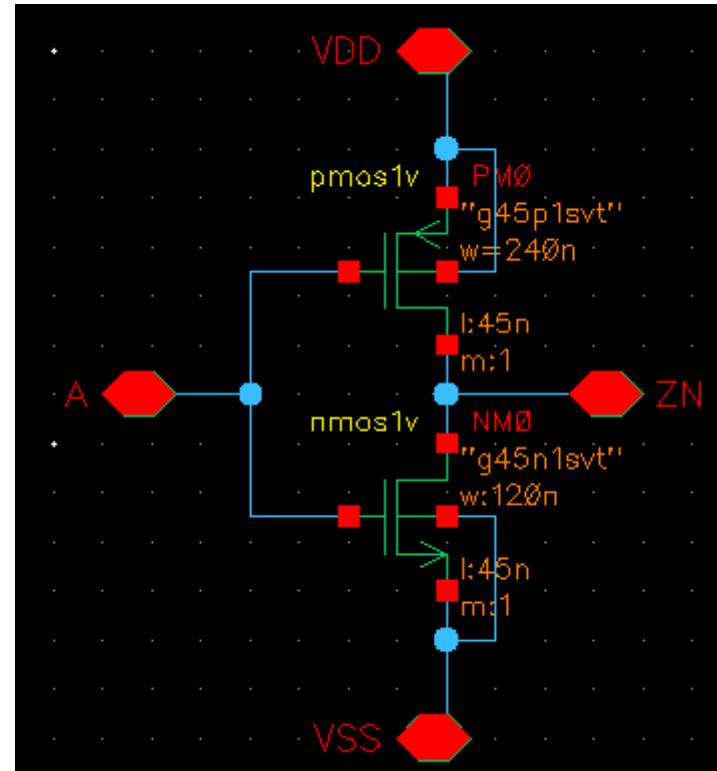
Layout

- Drawing Layouts
- Performing checks:
 - DRC – Design Rule Check
 - LVS – Layout Versus Schematic
 - RCX or PEX – Parasitic RC Extraction
- Performing post-layout simulations

Before You Start the Layout

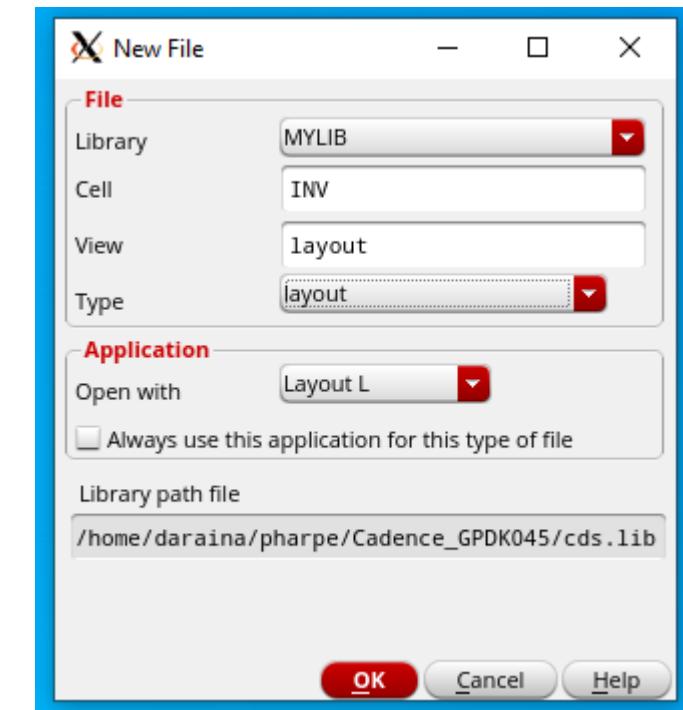
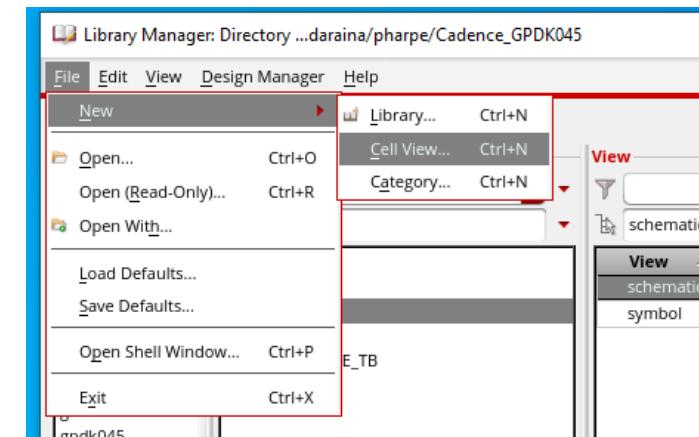
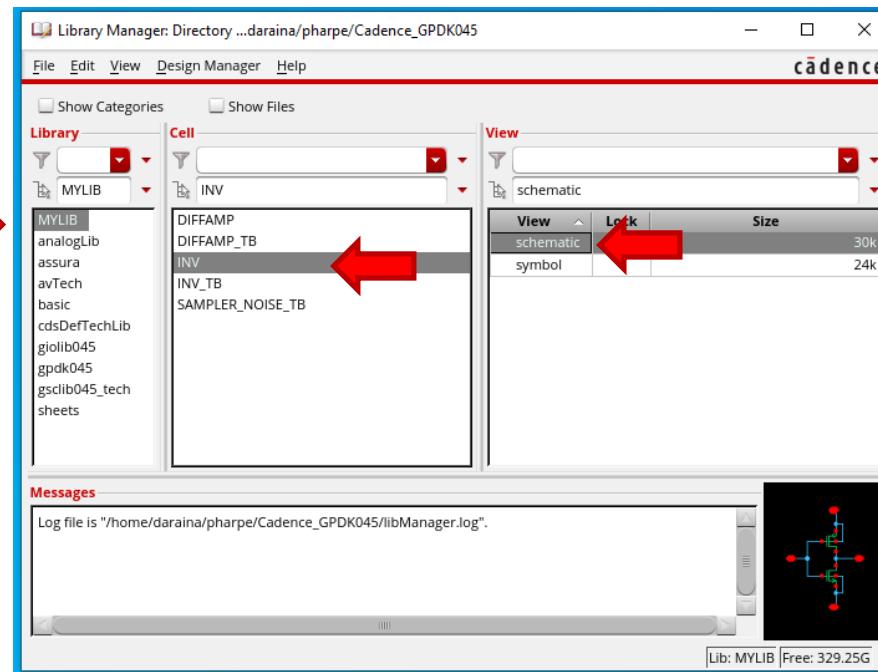
- It is assumed that you already have the schematic of an inverter in Cadence. If not, draw the inverter schematic and symbol as discussed before.
- Make sure to use the correct transistor types (nmos1v and pmos1v) with the given W and L, the correct connections, and the four pins (VDD, VSS, A, ZN).

 - nmos1v with $W = 120n$, $L = 45n$
 - pmos1v with $W = 240n$, $L = 45n$



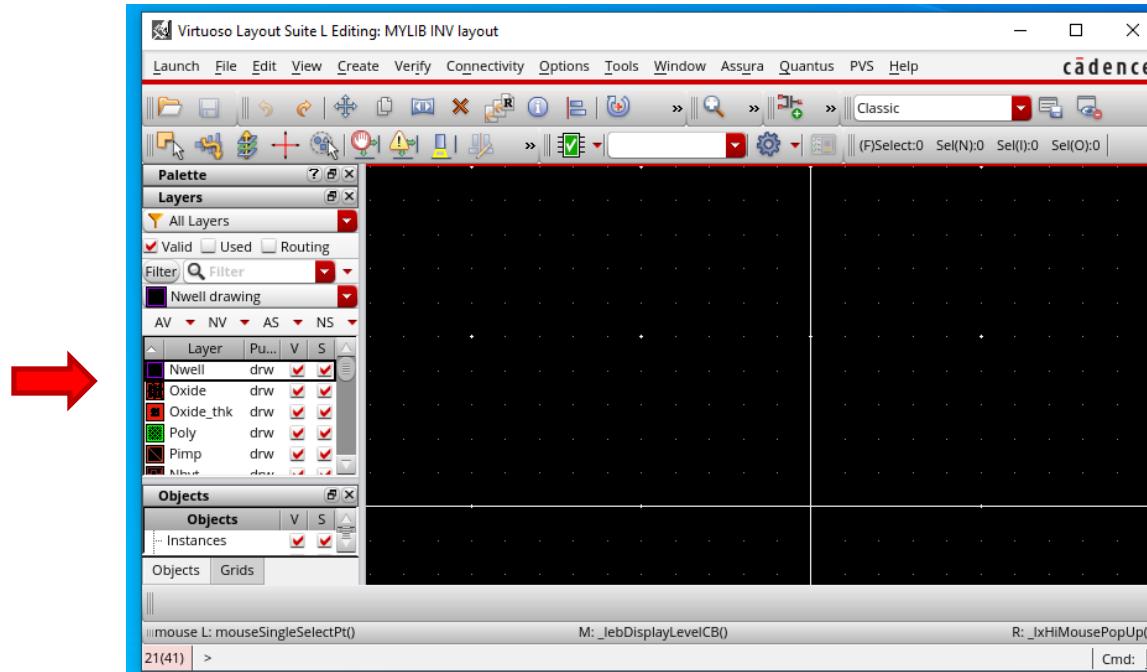
Creating a New Layout

- Check the name of the Library and the name of the Cell of your inverter schematic.
- Create a new cellview in the same Library, with the same Cell name, but this time a Layout view



The Layout Window

- You should now get a new window in which you can draw the layout. On the left side is a list of “layers” that you can use to draw metals, transistors, etc.



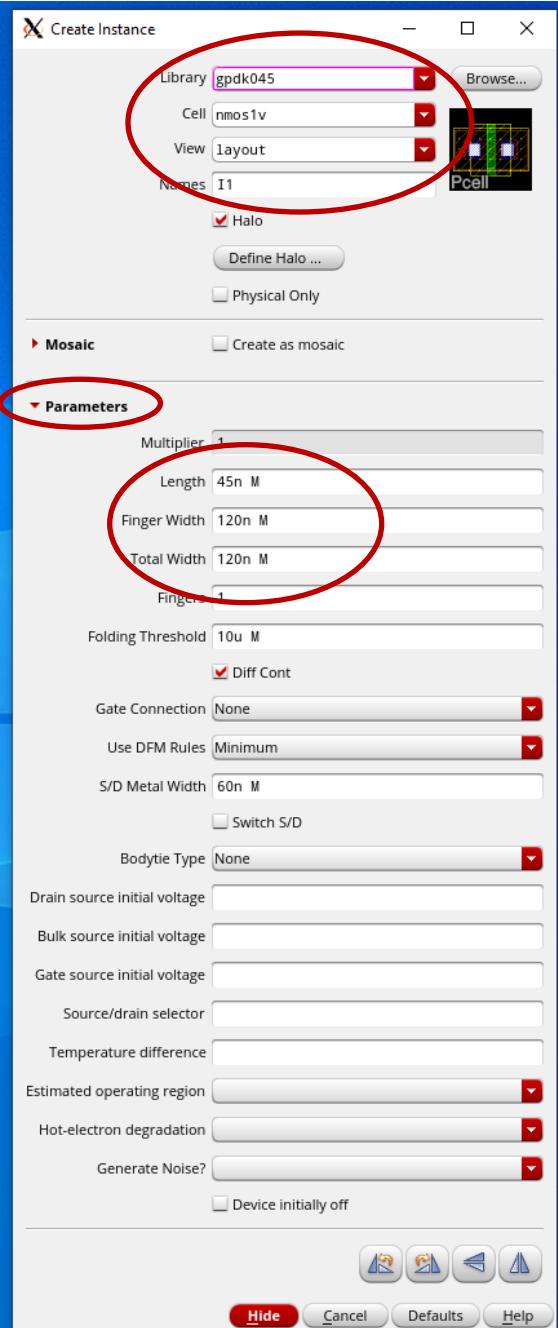
- You now have to make a layout (components, wire connections and pins) that is exactly the same as your schematic.

Shortcuts for Drawing a Layout

- “f”: fit to screen, Shift + “z” and Ctrl + “z” to zoom out and in
- “i”: insert a component from the library
- “l”: insert a label (needed to make a pin)
- “o”: insert a contact/via (connect multiple layers)
- “p”: draw a path in the active layer
- “q”: edit properties of a component already placed in the layout
- “r”: draw a rectangle in the active layer
- F2: save layout
- Shift + “x”: descend into selected sub-circuit
- Shift + “b”: return to top circuit
- Shift + “f” and Ctrl + “f”: change layout view (detailed or simplified)

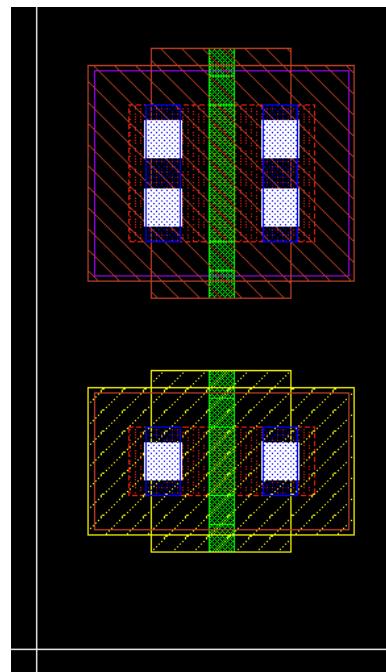
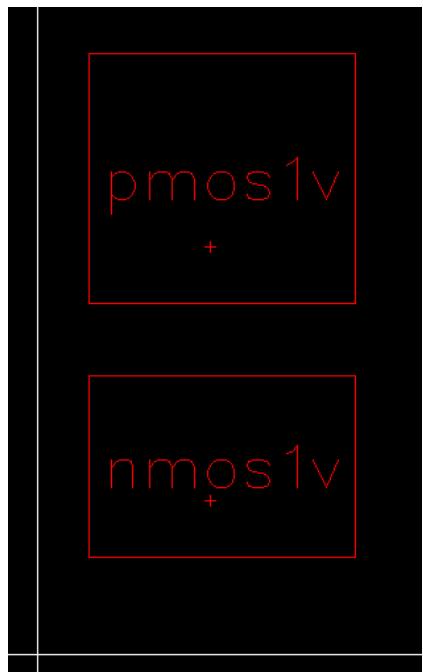
Adding Components

- As a first step: draw the two transistors that you have in the schematic.
- Use the shortcut “i” to insert new components into the layout. Select the transistor that you like to add and **use the “Layout” view**. If you scroll down in the window, you can find the W and L parameters. Adjust them to the same value as in the schematic. You can also change the parameters afterwards by editing the properties with the “q” shortcut.



Viewing the Layout

- After adding the two transistors, your layout may either show the layers of the transistors, or it may only show the outline.
 - Use the shortcuts Shift + “f” or Ctrl + “f” to switch between the two views.

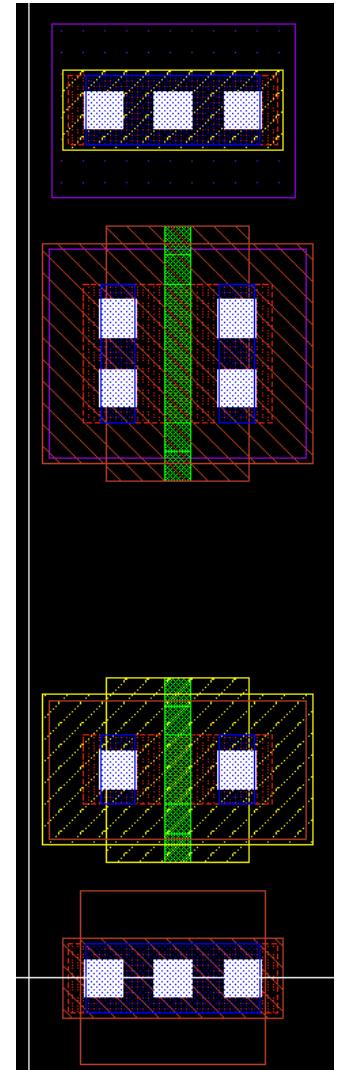
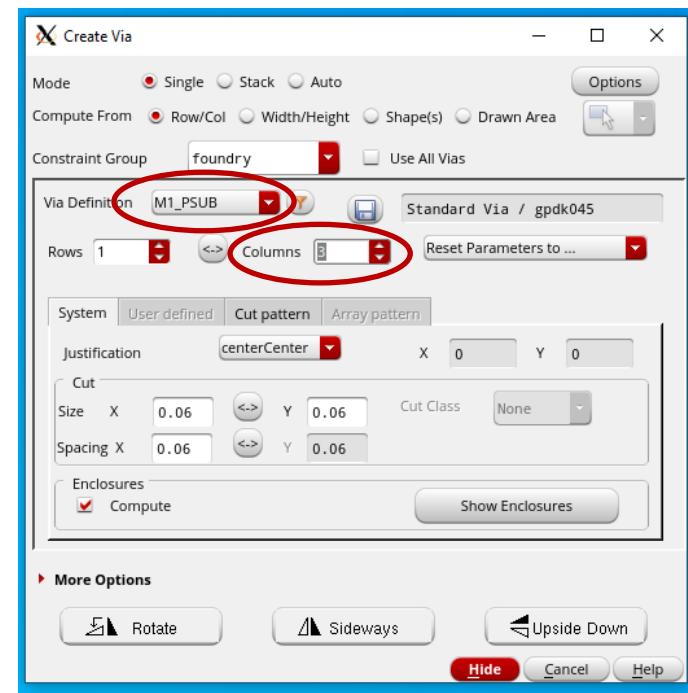


You can check which layers are used in the devices.
For instance:

- Green = Poly (Gate)
- Blue = Metal1 (Source and Drain connections)
- White = Contact (Metal1 to Diffusion connection)
- Yellow = Nimp (N implant to make an NMOS device)
- Brown = Pimp (P implant to make a PMOS device)
- Purple = Nwell (NWELL to make a PMOS device)

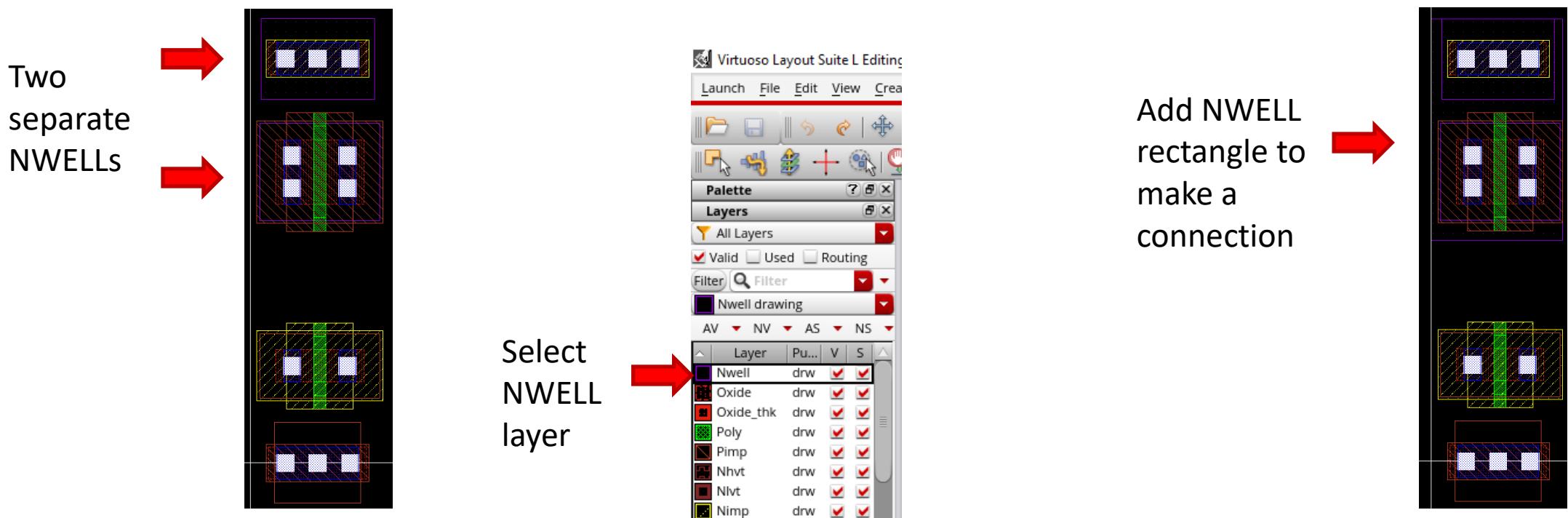
Adding NWELL/PSUB Contacts (1)

- As you know from the schematic, each transistor has four terminals: G, D, S and B (Bulk). By default, the layout has 3 connections (G, D, S). You need to add the bulk contacts (also called “vias”) manually:
 - For an NMOS, you need an M1_PSUB contact to connect the P substrate near the NMOS.
 - For a PMOS, you need an M1_NWELL contact to connect the NWELL surrounding the PMOS.
- To make a contact, press the shortcut “o”. Select the right via type and change the number of columns to 3.
 - Place the M1_PSUB and M1_NWELL contacts as shown in the layout view.



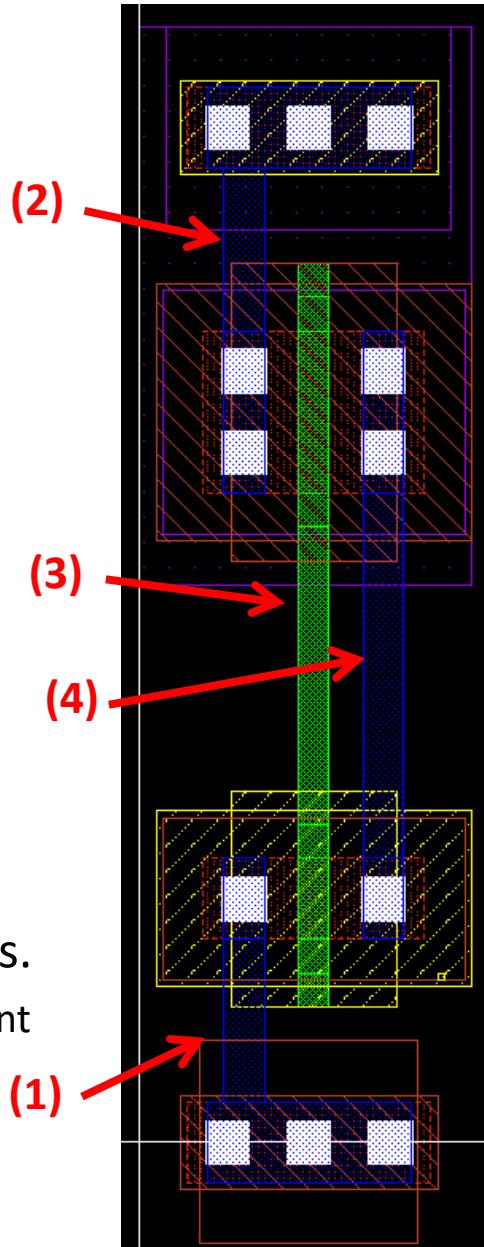
Adding NWELL/PSUB Contacts (2)

- P substrate is everywhere in your chip, so placing the M1_PSUB contact near the NMOS is sufficient to define the BULK voltage of the NMOS transistor.
- NWELLS are local, so if you want to connect the M1_NWELL contact to the NWELL of the PMOS, you may need to draw an extended rectangle of NWELL to cover (and connect) both parts.
- Select the NWELL layer, and use shortcut “r” to draw an NWELL rectangle to connect both NWELLS.



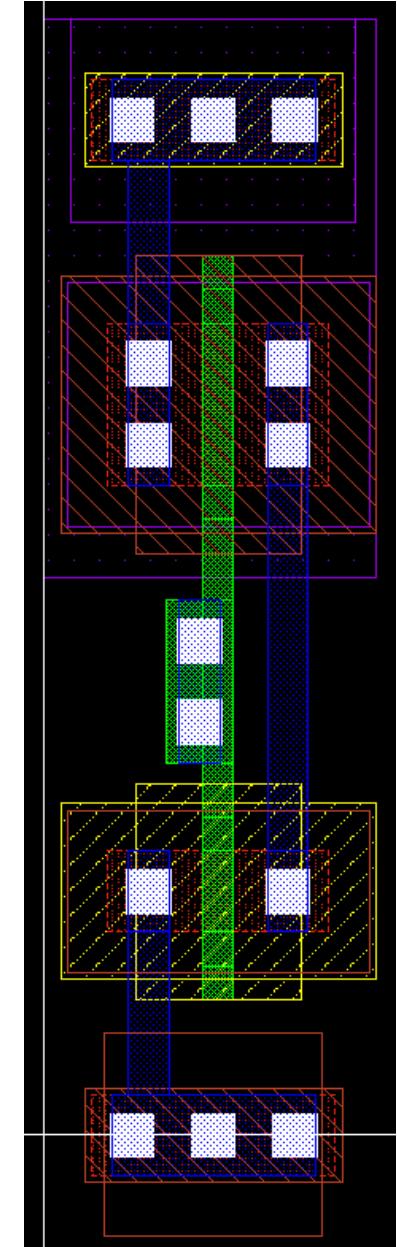
Making Connections

- The next step is to make the connections, like in the schematic:
 - The M1_PSUB and the NMOS source need to be connected (1)
 - The M1_NWELL and the PMOS source need to be connected (2)
 - The gates of PMOS and NMOS need to be connected (3)
 - The drains of PMOS and NMOS need to be connected (4)
- You can make connections by selecting the layer in which you like to make the connection, and then draw a rectangle with ‘r’ or a path with ‘p’.
- In this case, it is easiest to connect the gates with Poly, and to make the other connections with Metal1.
- Tip: if you align the NMOS and PMOS exactly, it is easier to make the connections.
- Select the appropriate layers and draw rectangles to make the required connections.
 - Note: for instance for “Metal1”, you can find multiple layers with the same name, but with a different purpose. Always select the “drw” (drawing) purpose.
- You should get a layout like the one shown here.



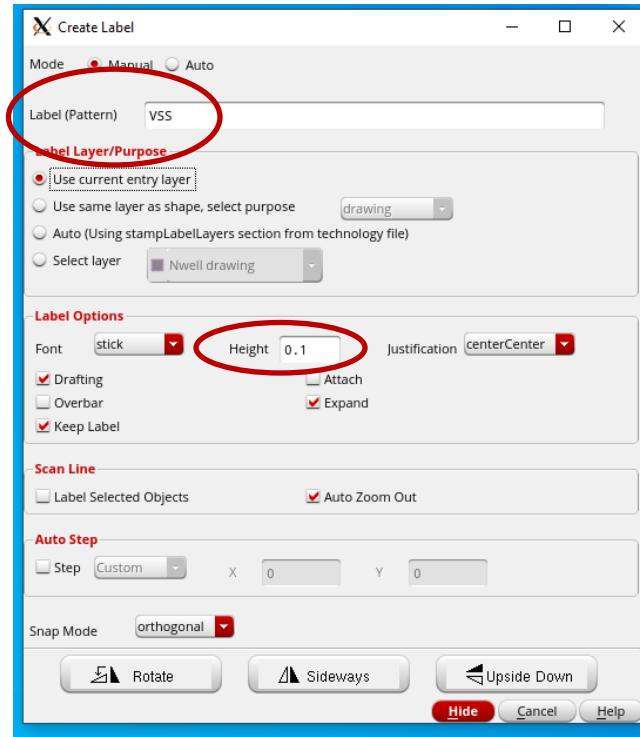
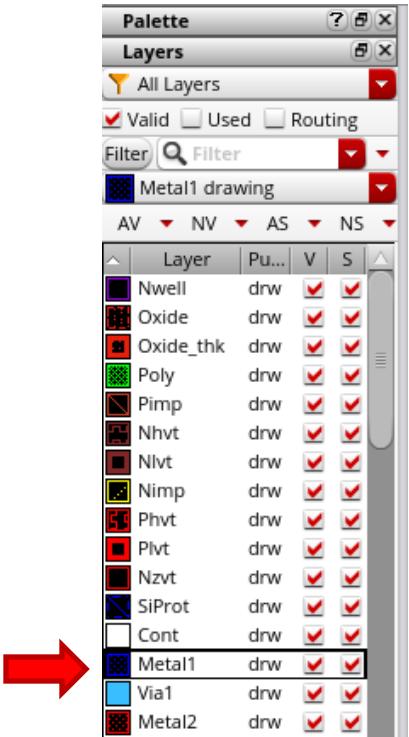
Adding Contacts

- The input gate of the inverter is now in the Poly layer. Maybe later on, you like to connect it to a metal layer (Metal1, Metal2, ...).
- To make connections from one layer to the other, you can use Contacts (also called Vias).
- Press ‘o’ to make a “M1_PO” contact with 2 rows and connect it to the Poly of the inverter’s input.

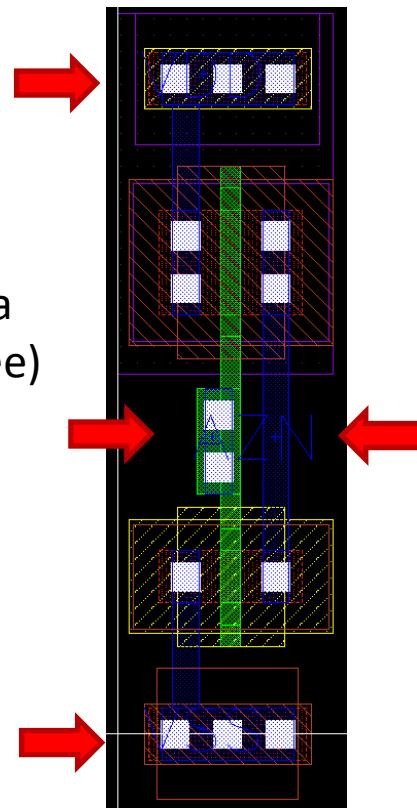


Adding Pins

- Now, we need to add the 4 pins that are also in the schematic.
- Since all 4 signals (VDD, VSS, A, ZN) are available on Metal1, we're going to make the pins on that specific layer.
- First, select the layer "Metal1". Then, press "I" to create a label (which creates the pin). Type the name of your pin and place the center of the label on top of the Metal1 structure that identifies that pin. Repeat for all 4 pins.



Placed pins (a bit hard to see)

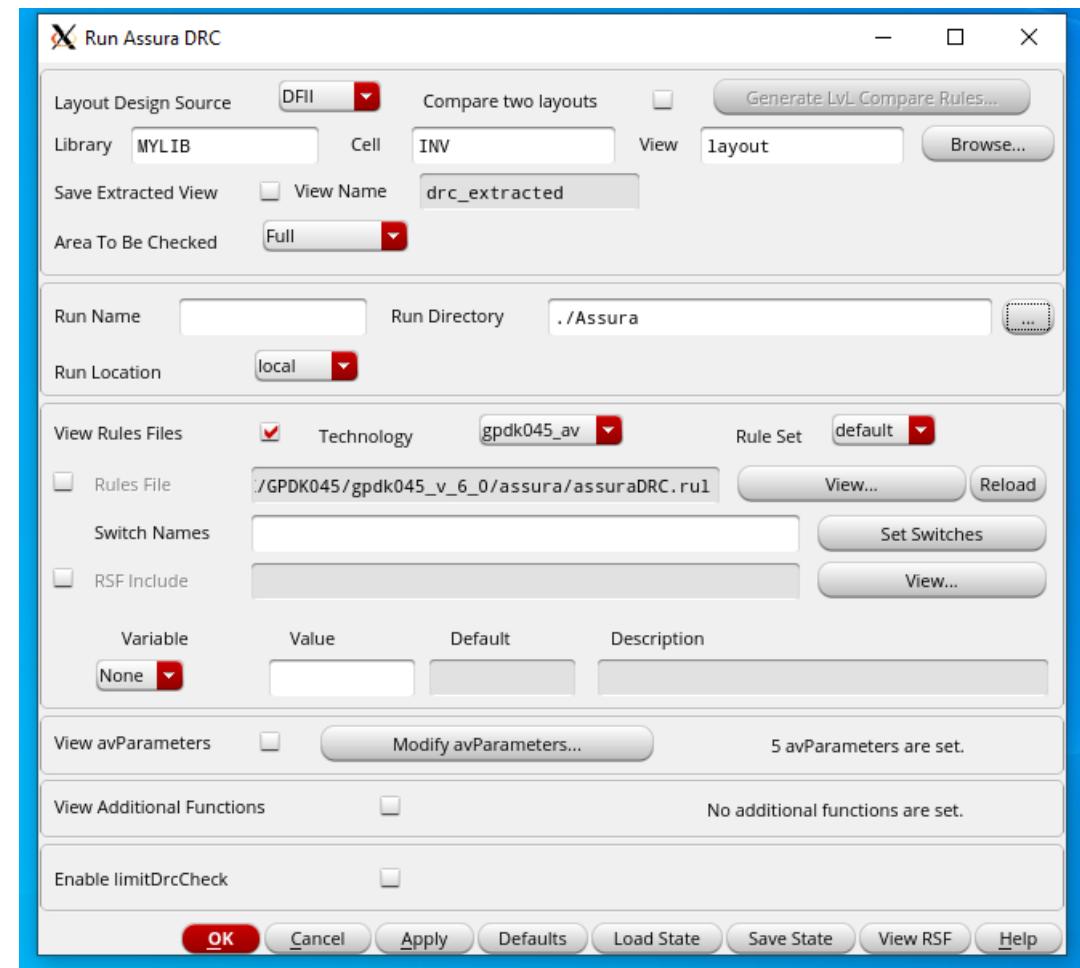
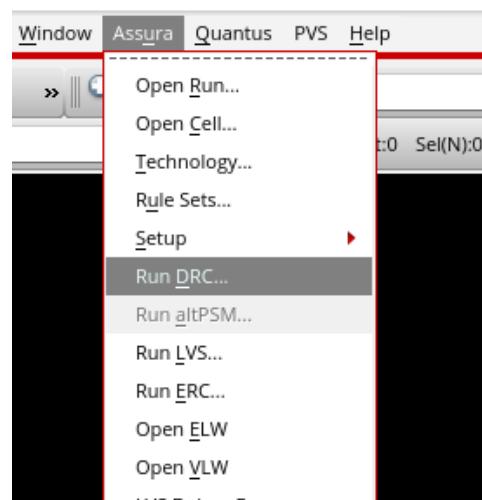


Checks

- Your design is now ready. Before you proceed, **SAVE** it.
The next step is to perform DRC and LVS checks:
- **DRC = Design Rule Check**
 - DRC checks if your layout can be manufactured (for instance: there are minimum space requirements, or distance requirements between different layers, etc., to guarantee correct production)
- **LVS = Layout Versus Schematic**
 - LVS checks if your layout is actually the same as your schematic
- If DRC and LVS are “clean”, it means your layout is correct and manufacturable!
- After that, you may do an RCX/PEX (Parasitic RC Extraction) to determine the layout parasitics.
 - After that, you can do a post-layout simulation, which is the same as your schematic simulation, except that the simulation will now include the impact of the extracted RCs from your layout.

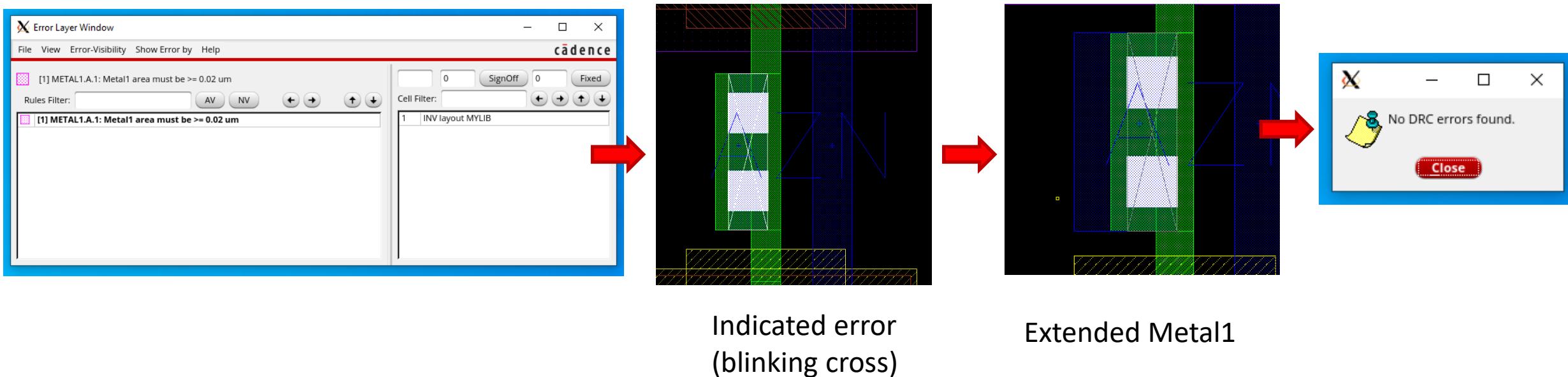
DRC (1)

- Use the menu “Assura” → “Run DRC” in the Layout window
- Check the settings in the DRC window
 - Set the run directory to “./Assura”
 - Set the technology to “gpdk045_av”
 - Usually, it remembers your settings correctly next time.
- Press the button “OK”
 - It may ask you to overwrite previous results, and to overrule a previous job, which is OK.
- When finished, it asks if you want to see the results.
 - Of course, so you say “Yes”



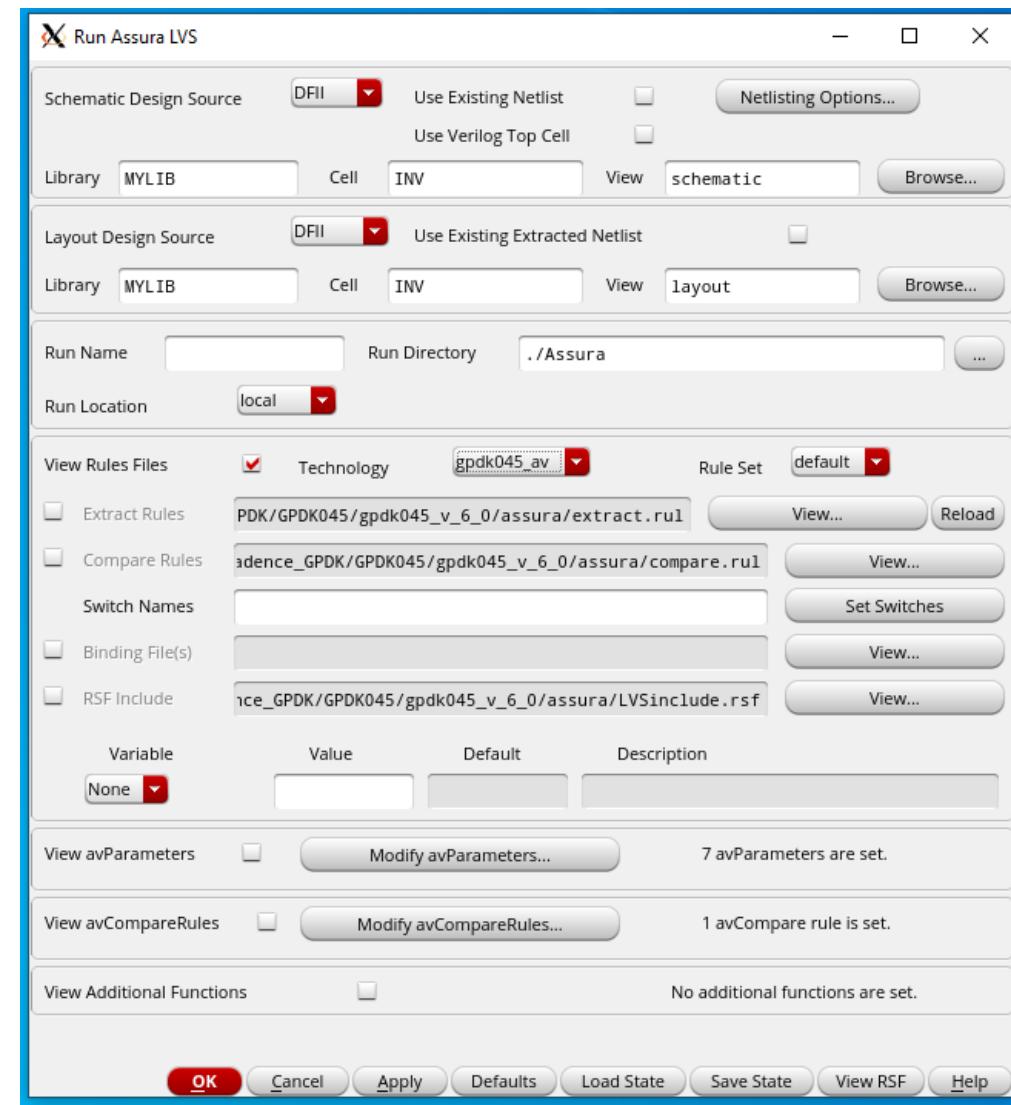
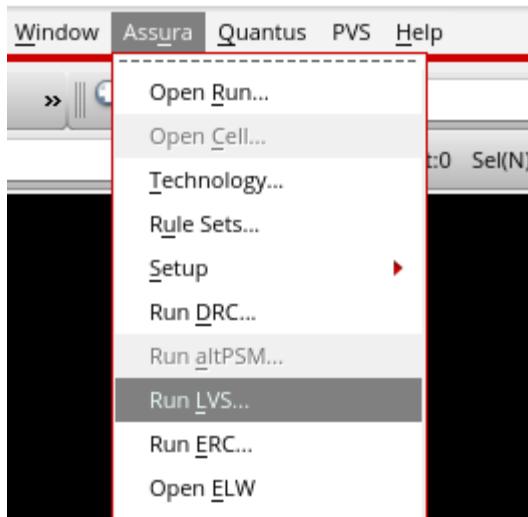
DRC (2)

- You will now get the DRC results window.
 - In this case there is 1 violation, namely the area of one piece of Metal1 is too small (below 0.02um^2). You can use the tool to highlight exactly where the violation happens in the layout, so you can easily find and solve the problem.
 - In this example, the issue is that the area for Metal1 for pin “A” is too small. If you enlarge that Metal1 by drawing a larger rectangle, it can be solved.
- If there are errors, try to solve them and repeat the DRC check until the layout is DRC-clean
 - If there are no violations, you’ll get a simple message “No DRC errors found”



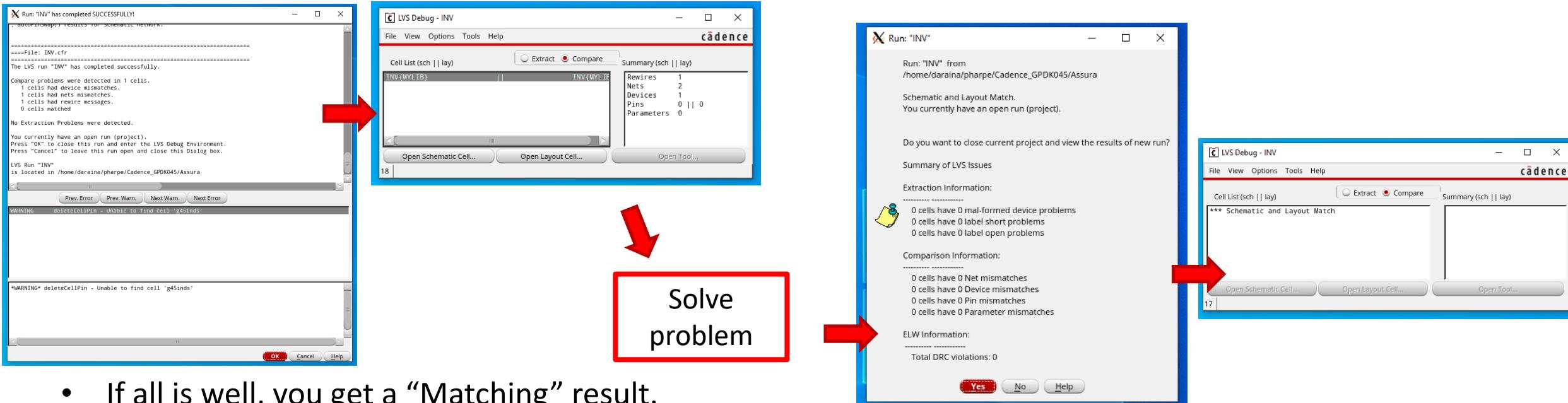
LVS (1)

- Use the menu “Assura” → “Run LVS” in the Layout window
- Check the settings in the LVS window
 - Set the run directory to “./Assura”
 - Set the technology to “gdk045_av”
 - Usually, it remembers your settings correctly next time.
- Press the button “OK”
 - It may ask you to overwrite previous results, and to overrule a previous job, which is OK.
- When finished, it asks if you want to see the results.
 - Of course, so you say “Yes”



LVS (2)

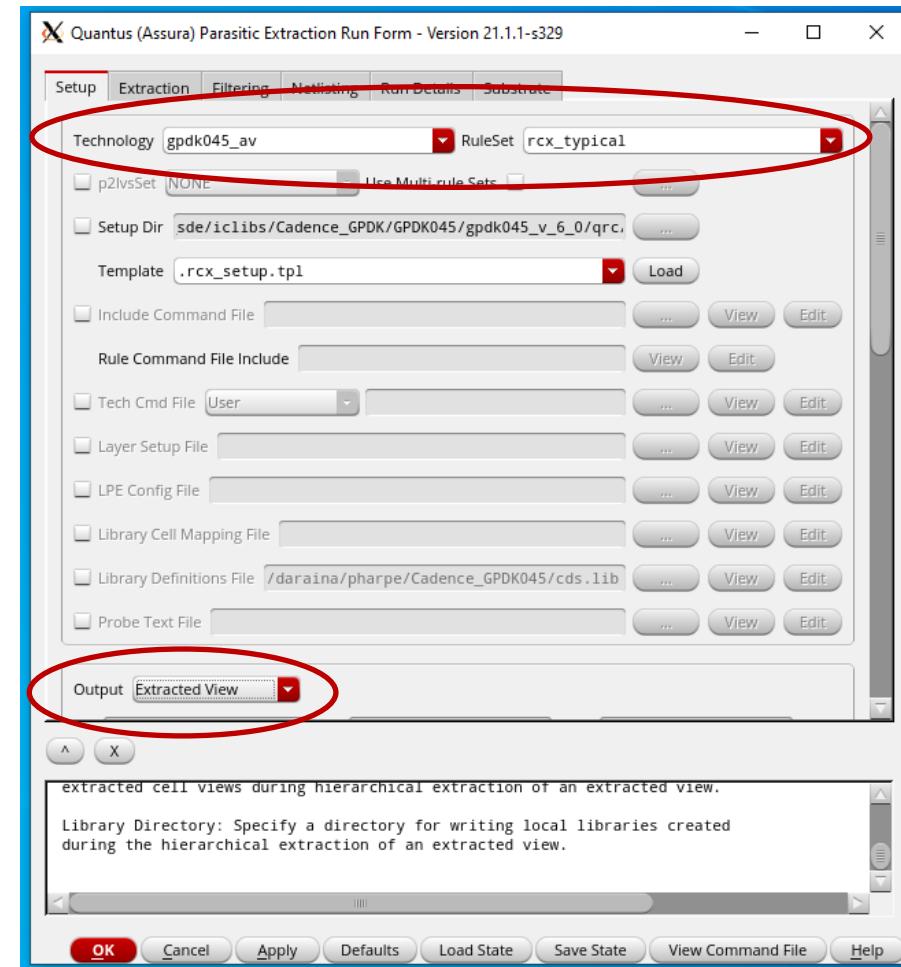
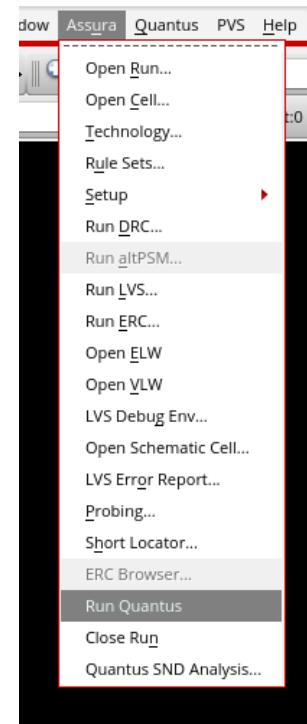
- If there is an LVS problem, you get a temporary window. If you click “OK”, you will get an LVS debug window to help you find the problem. However, if the Layout deviates too much from the Schematic, it may not show the Debug window.
 - Try to fix the problems one by one. Start with the Pins, and then the Nets or Devices.



- If all is well, you get a “Matching” result.
 - Always click “Yes” until you see the Debug window with the “Schematic and Layout Match” message!
 - Redo your DRC check if you had to change your layout due to LVS errors!

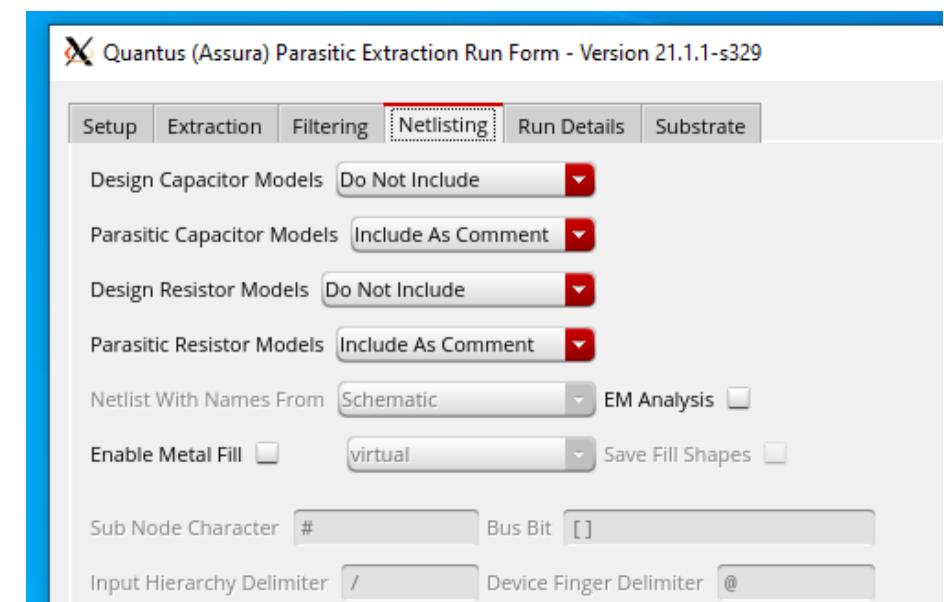
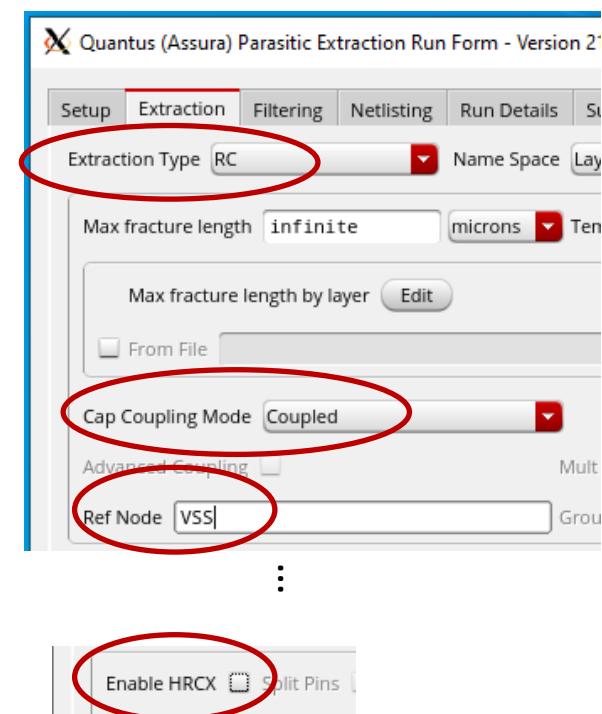
RCX (1)

- Make sure you pass the DRC and LVS first
- Use the menu “Assura” → “Run Quantus” in the Layout window
- Check the settings in the RCX window under the tab “Setup”
 - Set the technology to “gdk045_av”, Ruleset to “rcx_typical”, and the Output to “Extracted View”
- Continue with changing the settings under the “Extraction” and “Netlisting” tabs as explained on the next slide.



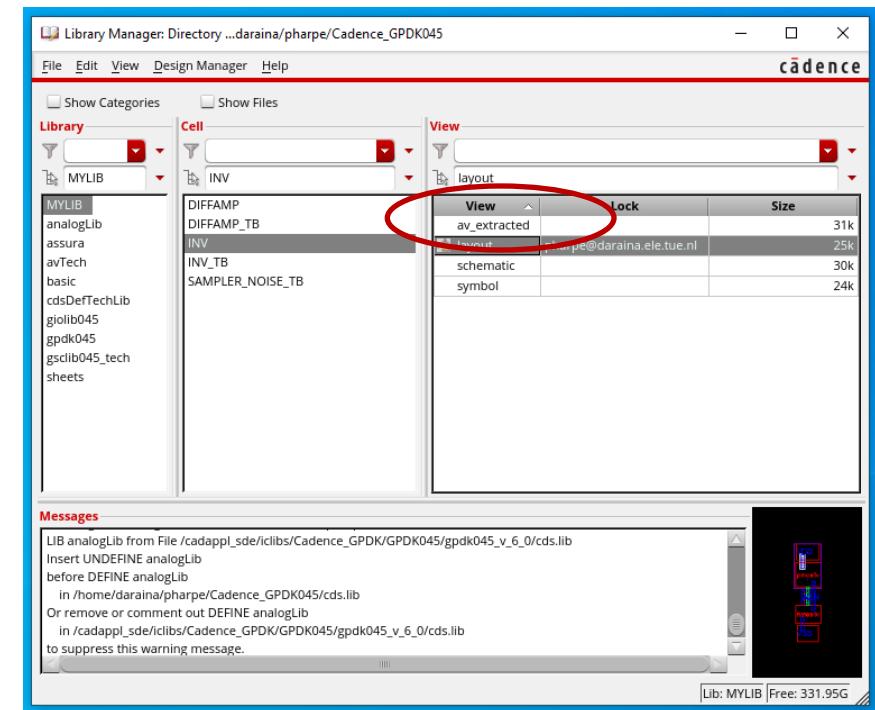
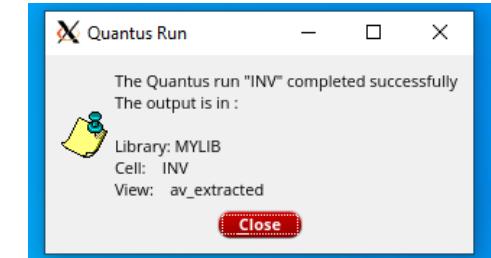
RCX (2)

- In the “Extraction” tab, set the Extraction Type to “RC”, the Cap Coupling Mode to “Coupled”, the Ref Node to “VSS”, and disable the “Enable HRCX” setting
- In the “Netlisting” tab, change the Parasitic Capacitor Models to “Include As Comment”
- Usually, it remembers your settings correctly next time.
- Press the button “OK”
 - It may ask you to overwrite previous results, and to overrule a previous job, which is OK.



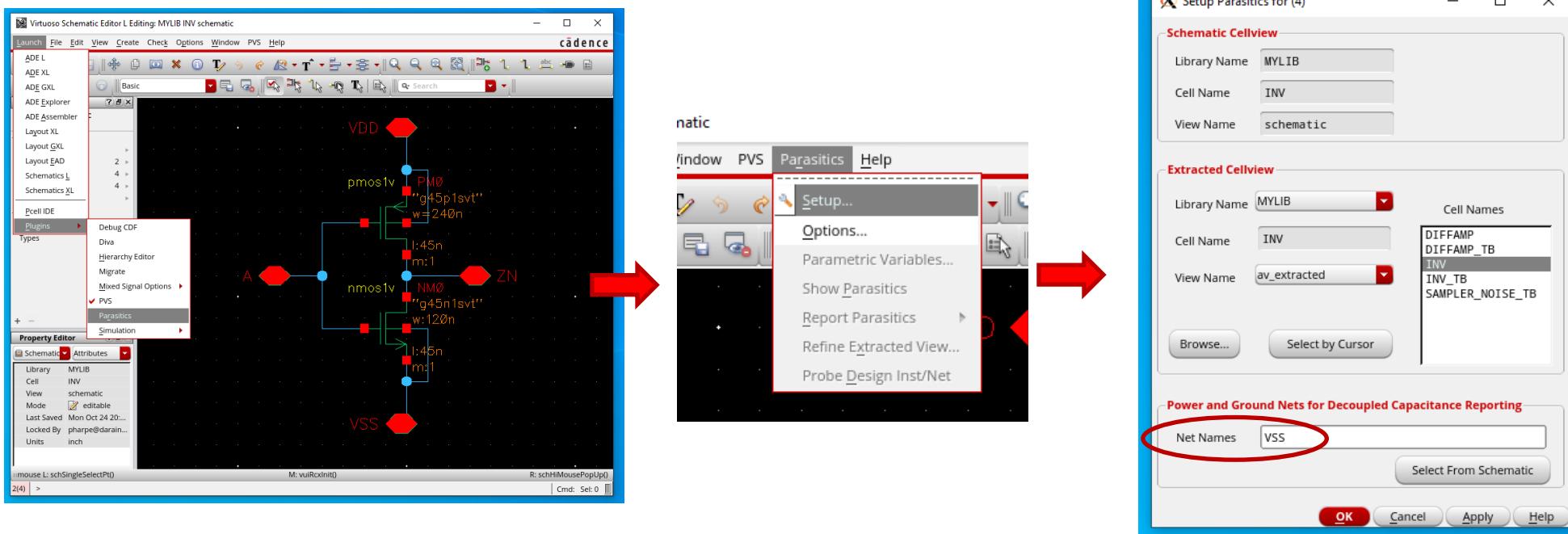
RCX (3)

- When finished, it may either:
 - Show you a Failure. In that case check the log file for errors.
 - Show successful results. In that case, press “close”.
There is now a new CellView “av_extracted” in the library that contains the parasitics.
 - Note: it is not very useful to open the av_extracted view. Instead, you can:
 - Open the corresponding schematic and display the parasitics inside the schematic.
 - Perform a simulation including schematic + parasitics



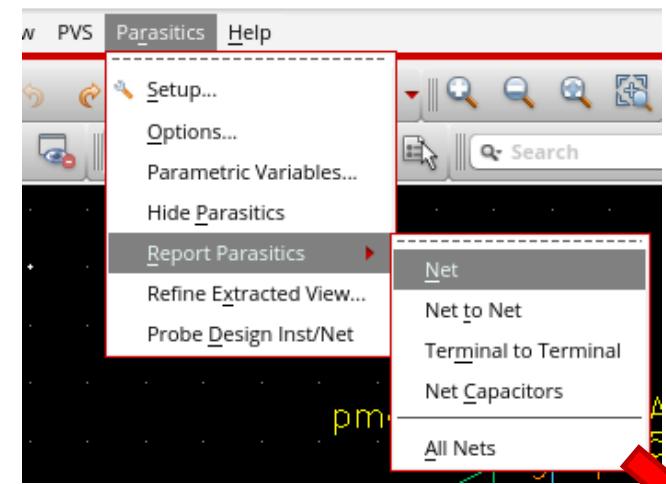
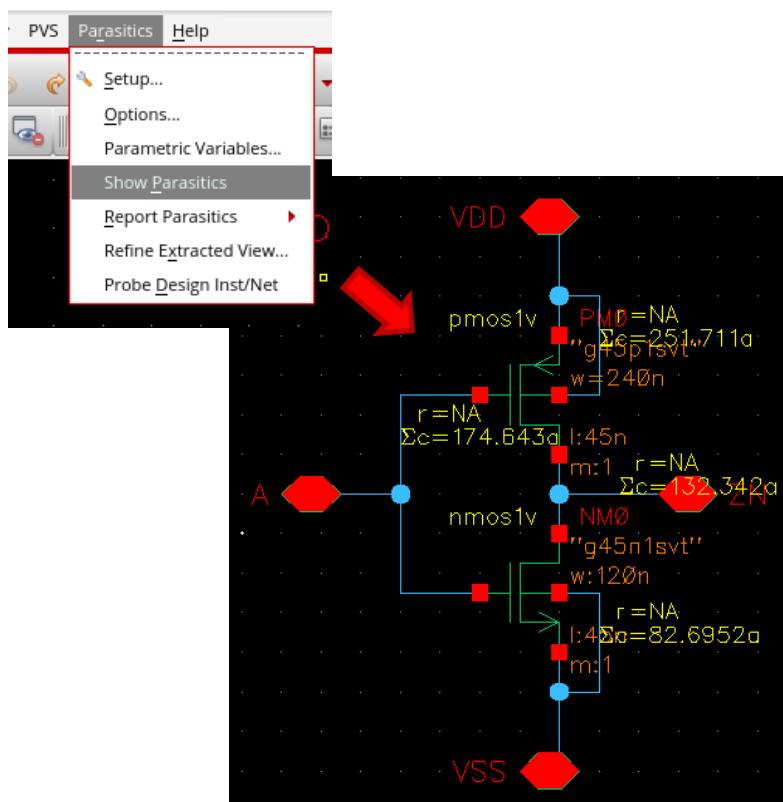
View Parasitics in the Schematic (1)

- Open the schematic for which you did the RCX
- Use the menu “Launch” → “Plugins” → “Parasitics”
- This will enable a new menu “Parasitics”, from which you pick the “Setup” option
- This should correctly show your related extraction file. Add “VSS” to the Net Names and click OK.



View Parasitics in the Schematic (2)

- You can now use “Parasitics” → “Show parasitics” to display the total parasitic capacitance per node.
- You can also use “Parasitics” → “Report parasitics” to list the parasitics at a net, or between nets/terminals. Click the desired net(s) or terminals in the schematic, and then a list with RCs is shown.

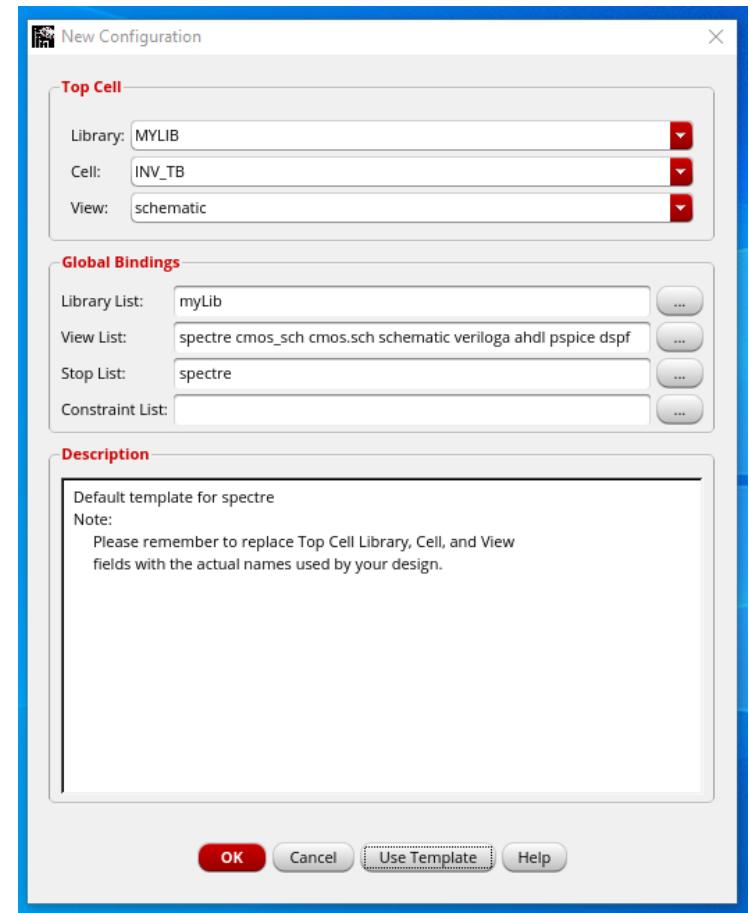
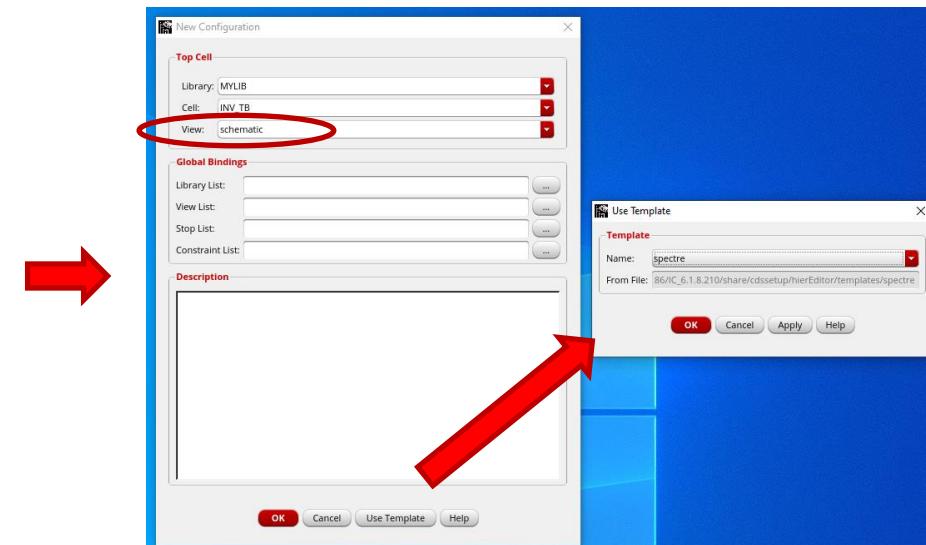
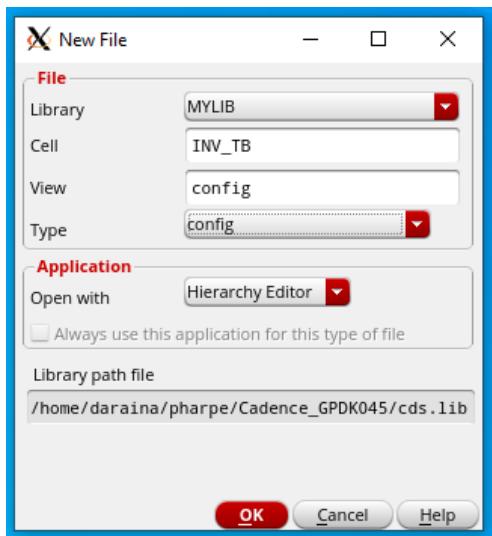


The screenshot shows the 'Parasitics for net /A' dialog box. It lists parasitic components for net /A, categorized by type: Resistors (R), Coupled Capacitors (coupled C), Decoupled Capacitors (decoupled C), Self Capacitors (self C), Inductors (L), and Kerr Inductances (K). The table includes columns for Instance, Type, Value, From, To, and additional totals at the bottom.

Instance	Type	Value	From	To
/r12	R	60.3072	/2:A	/3:A
/r1	R	75.6918	/1:A	/2:A
/rk1	R	22.504	/A	/2:A
/c9	C	5.64625a	/2:A	/ZN
/c8	C	10.9901a	/A	/1:VDD
/c7	C	10.022a	/A	/2:A
/c6	C	39.6076a	/A	/ZN
/c33	C	160.985z	/2:A	/VSS
/c32	C	4.75532a	/3:A	/VSS
/c31	C	2.49045a	/A	/VSS
/c3	C	4.91886a	/1:A	/VDD
/c26	C	1.34227a	/2:A	/2:VSS
/c24	C	1.28017a	/2:A	/1:ZN
/c23	C	6.3934a	/3:A	/2:VSS
/c22	C	1.37306a	/2:A	/2:VDD
/c21	C	4.33336a	/3:A	/1:ZN
/c20	C	1.0153a	/2:A	/2:ZN
/c19	C	25.5694a	/2:A	/1:VDD
/c18	C	7.9612a	/1:A	/2:VDD
/c17	C	13.7386a	/3:A	/1:VDD
/c16	C	6.46031a	/1:A	/2:ZN

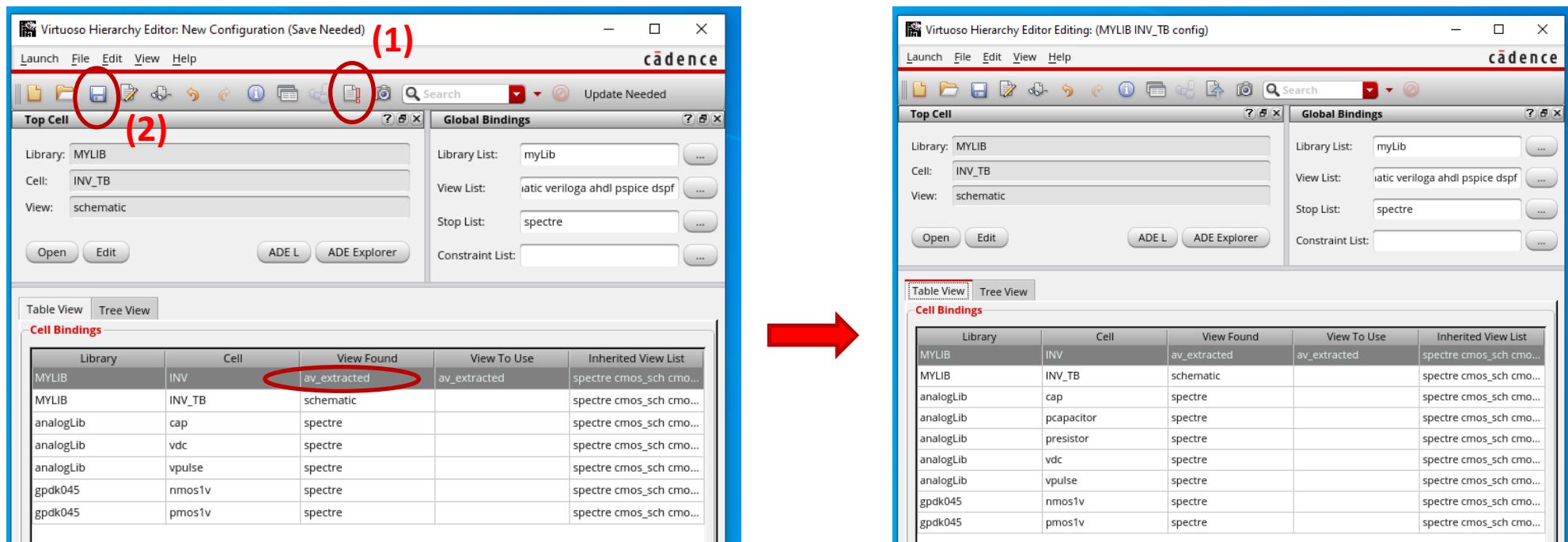
Post-Layout Simulations (1)

- First, we have to make a Config view for the test bench
 - With this Config view, you can easily change the configuration of what you simulate, i.e.: it enables you to change easily between schematic and post-layout simulations.
- Create a new Cellview for your testbench INV_TB of the config type.
- In the “New Configuration” window, set the “View” to “schematic”, then press the “Use Template” button and pick the “spectre” template, which will load the settings under “Global Bindings”.



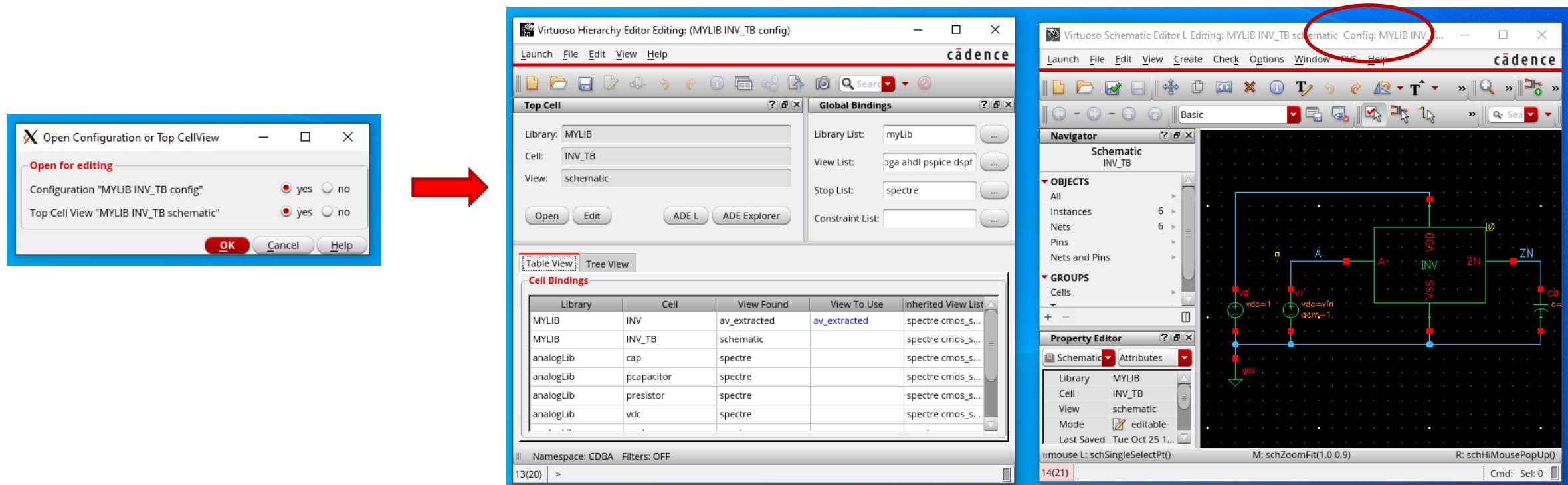
Post-Layout Simulations (2)

- The config view now shows which cells are (hierarchically) used for the simulation. By default, you use the INV_TB and INV schematics, plus spectre models for the gpdk045 and analogLib components
 - Right click on the “schematic” view of the INV cell to change it to “av_extracted”. This will enable post-layout simulation.
 - The testbench (INV_TB) remains at schematic level, because there is no layout for a test bench.
 - Next, press the “Recompute/Update” button (1), and then the “Save button” (2) to update the hierarchy.



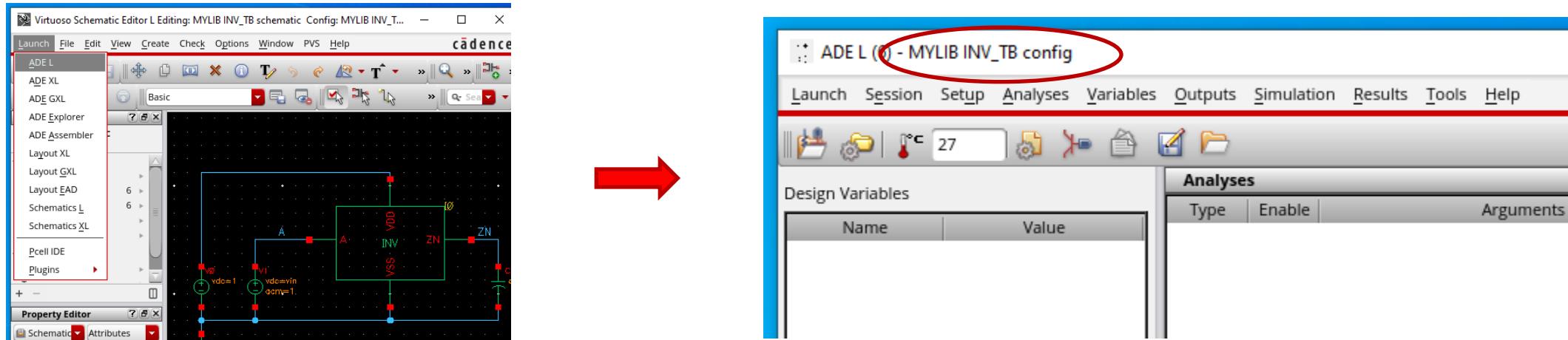
Post-Layout Simulations (3)

- Close the config view and go back to the library manager to open the config view of INV_TB.
- It will ask you what to view. If you say yes to both questions, you'll get to see the previously closed Config view again, and you'll get to see your schematic. However, the schematic is actually following the hierarchy as defined in the configuration, which we can see from the “Config” in the window name.



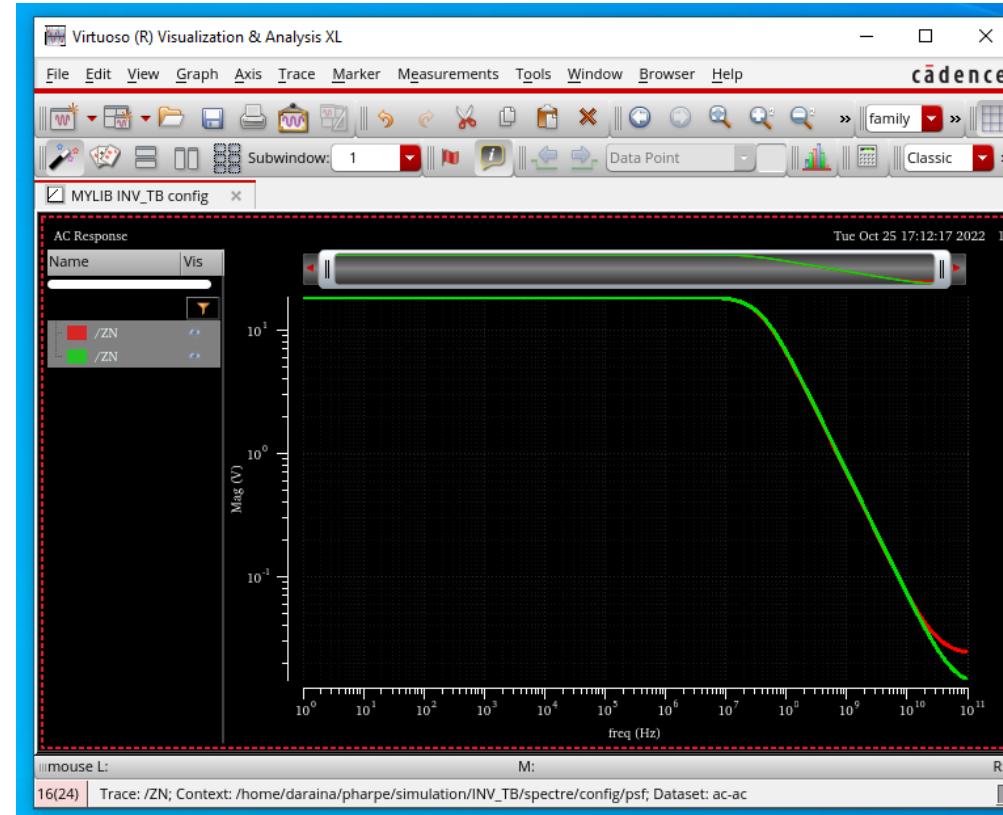
Post-Layout Simulations (4)

- You can now start the simulation tool from the schematic viewer showing the config.
- The simulation tool (ADE L) will look and work the same as for the schematic simulation.
 - If you carefully look to the ADE L window name, you can see it is actually going to simulate the hierarchy as defined in the CONFIG view (which could be either a schematic or a post-layout simulation, dependent on what you set in the Config view).
 - If you start ADE L from the INV_TB schematic, it will show that you will simulate the schematic.
- You can now load the state (from your schematic simulation) and run simulations.



Post-Layout Simulations (5)

- For comparison, the figure below shows the AC simulation of the inverter both schematic and post-layout.
- The differences are very small as the layout for such a small cell has limited impact.
- For larger circuits and/or higher frequencies, the layout can be highly dominant in final performance.



Summary

- You now know:
 - How to access Cadence
 - How to use Cadence
 - How to make schematics
 - How to make symbols
 - How to make a test bench
 - How to run simulations
 - How to draw layouts
 - How to check DRC, LVS, and RCX
 - How to run post-layout simulations