EE-320 ANALOG IC DESIGN TP-2024/2025
PRACTICAL EXERCISE SESSION No. 1

Note: For students connecting virtually, the details of the Zoom Meeting for the TP sessions are given below (The same link, meeting ID, and passcode are valid for all 4 TP sessions)

Meeting link:
https://epfl.zoom.us/j/63742124234?pwd=jO1wa65VGLrFCFrMfsVDBXreRpY318.1

Meeting ID: 637 4212 4234
Passcode: 405170

Calendar Event: Zoom Calendar Event for EE-320: Analog IC Design TP Sessions

Accessing Servers

Connecting to a VM

We use Ubuntu virtual machines (VM) to connect to the servers that host the EDA tools. You can use a computer room or use your own computer to access a VM.

Working in computer rooms

The computer rooms **BC 07-08** (that is where we will be), CO 260, and MXF 014 are equipped with Dell thin clients providing access to virtual machines (VM). Each thin client has a display, a keyboard, a mouse, and a small Wyse network box located on the rear side of the display. You can use any seat in the room and **login with your GASPAR account**. Do not store personal files in the Ubuntu VM.

To use a Dell thin client:

1. If required, turn the display on.
2. If the client is down, start it by pressing the start button (on the left side of the Wyse box). Wait until you get the start button blue-lighted.
3. At the appearance of the login window, **enter your GASPAR username and password.**
4. After a successful login, you see a list of available virtual machines on the left of the screen. Select the Linux virtual machine **STI-EDA-LABS-RTX**.

Working with your own computer

The **STI-EDA-LABS-RTX** virtual machine can be used from a personal computer as follows (note that if you work from home, it is recommended not to use the EPFL VPN):

Using a web browser: Login to **vdi.epfl.ch** using your GASPAR credentials.

Connecting to a server

To ensure that the workload is evenly spread across the two servers, please follow the allocation listed in the table below.

edauserN	Server
edauser1 – edauser135	selsrv1
edauser135 – edauser242	selsrv2

You want your simulations to run faster, don't you? ☺

1. Once the Ubuntu desktop is displayed (it may take some time), open a terminal window (Ctrl-Alt-T or right-click in the desktop area and select Open Terminal).
2. From the VM terminal window, log in to one of the servers selsrv1, or selsrv2 with your edauser*N* account (password: `24user_eda!`).

```
>> ssh -X edauserN@selsrv1.epfl.ch

OR

>> ssh -X edauserN@selsrv2.epfl.ch
```

Do not forget the -X (capital X) option, otherwise, the display won't be redirected properly to the local screen.

You can change your password with the yppasswd command. We recommend that you do so to keep your files private!

Do not confuse the terminal windows running locally on the Ubuntu VM and the terminal running on the remote Linux server. EDA tools are *not* available on the Ubuntu VM.

When you are done working, quit running tools, log out from the selsrv1 or selsrv2 server (type *exit* in the terminal), and then disconnect from the Ubuntu VM. There are several ways to quit the Ubuntu VM:

- In the top **Options** menu, select **Disconnect** and **Log Off** (selecting Disconnect only disconnects the VM, but keeps the session active).
- In the top right menu, select your name and then select **Log Out**.

Running Cadence Virtuoso

Setting up the environment

Once you are connected to the one of the selsrv# servers, run the following commands in your terminal:

```
>>cd
>>gtar -xvf /softs/classroom/tutorials/2024-2025/EE320/EE320-2024.tar.gz
```

This will create a directory named EE320-2024 which contains subdirectories for various tools. **We will always work in the subdirectory for Cadence Virtuoso (CDS_VISO).**

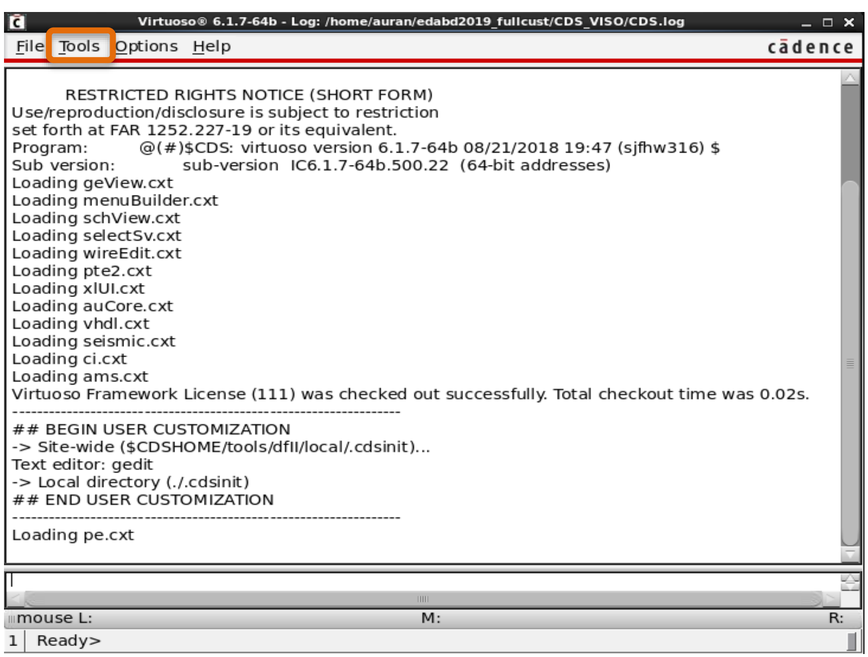
<u>You will have to do this only once.</u> You will use this environment for all exercise sessions.

Starting Cadence Virtuoso

Go to your EE320-2024/CDS_VISO directory and run the virtuoso command followed by an ampersand.

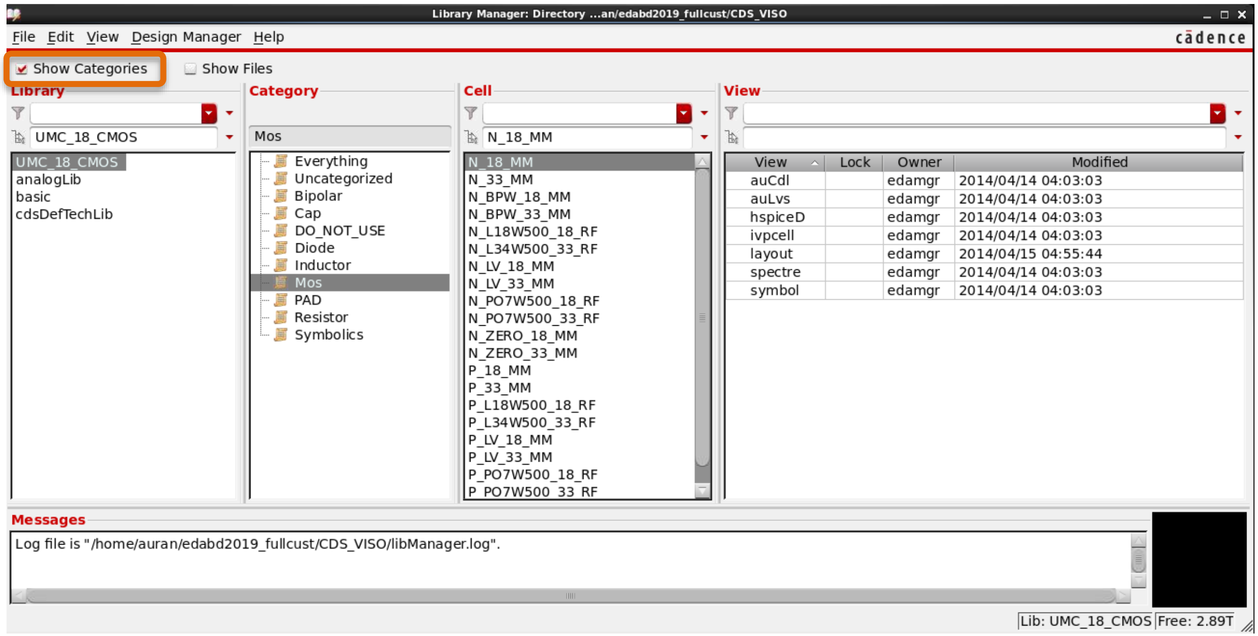
```
>> cd
>> cd EE320-2024/CDS_VISO
>> virtuoso &
```

You will see the main command window open. This window is called CIW – Command Interpreter Window. This is where you can access various tools and settings within the Virtuoso Suite, and where you can exit Virtuoso.



The Library Manager

In the CIW window, click **Tools>Library Manager**. The library manager is where you will find all the technology libraries and where you will create your own custom-design libraries.

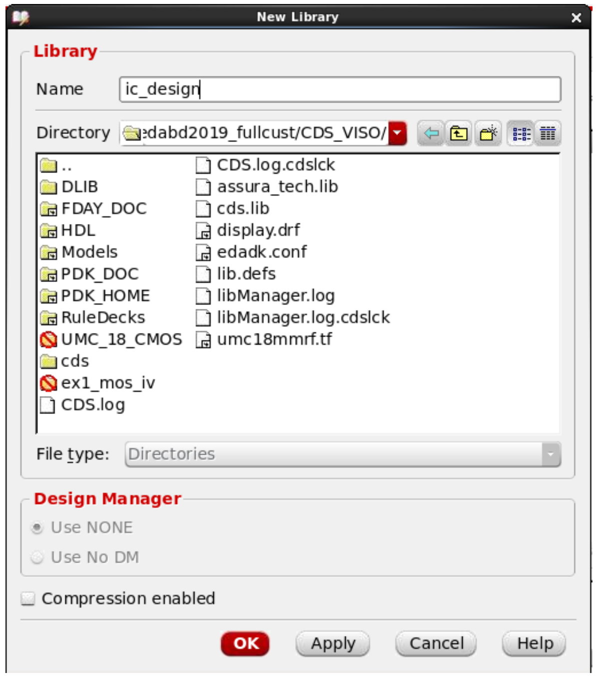


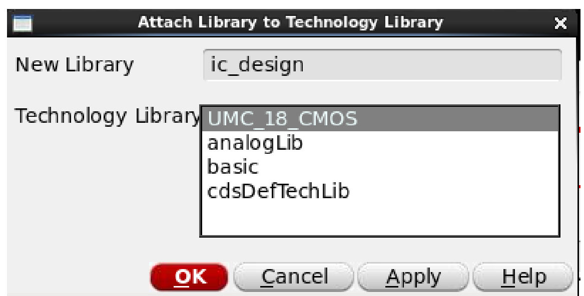
In your library manager, click on the **"Show categories"** button at the top of the window. Find the **"UMC_18_CMOS"** library and observe its contents. You will see circuit elements such as transistors (Mos and Bipolar), capacitors (Cap), resistors (Resistor), and so on. These are created and characterized by the foundry (UMC), so they are ready to be simulated and manufactured.

Find the **"analogLib"** library and observe its contents. You will see generic passive components and dependent/independent sources. These are created by Cadence to help your simulations. These cannot be manufactured.

You need a dedicated library to store your own work. This library will have a collection of custom circuit blocks called cell views. Click **File>New>Library**. Give a meaningful name to your library. We suggest "ic_design". Click OK.

Do not forget to choose the "Attach to an existing technology library option" and select UMC_18_CMOS!

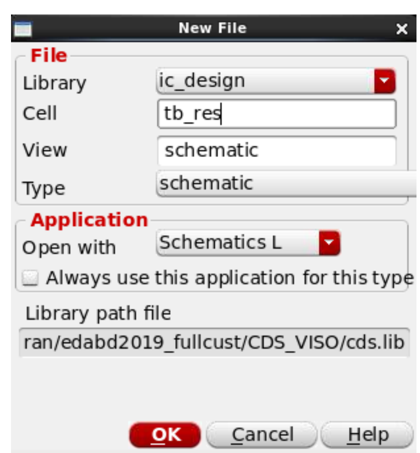




1. Warm-up example: a resistive voltage divider test bench

Creating a schematic

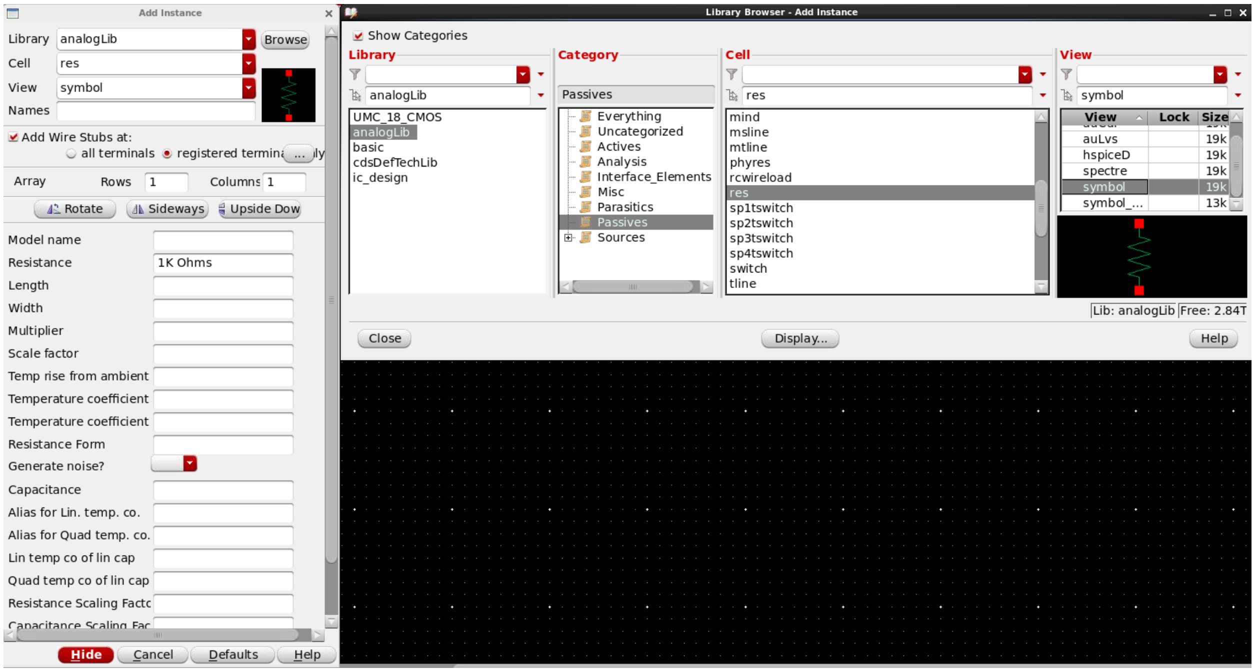
In your library manager, select your new library and click **File>New>Cellview**. Select **"schematic"** as the type of your cellview. Name your schematic as "tb_res". Click OK.



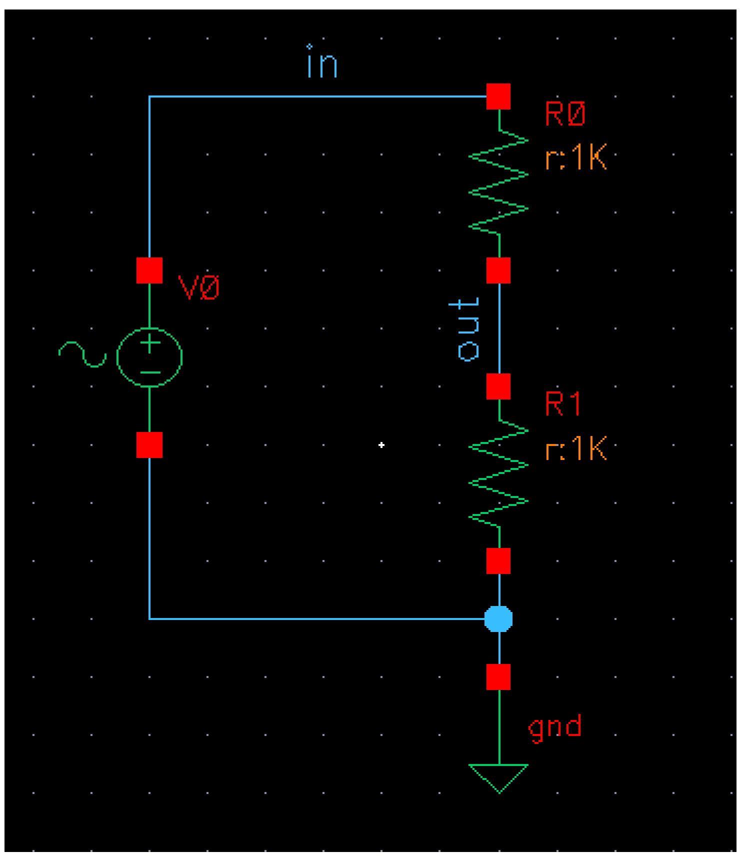
You will see an empty black canvas. Go to **Options>Editor** and change the "**Add instance browser type"** setting to **"library"**, then click OK.

- Return to your canvas and press the key "**i**" on your keyboard, which will open the **"Add Instance"** dialogue.

Click browse, select the library analogLib, and select “res”.



Place the resistor on your canvas. Then browse to analogLib and place the independent voltage sources and grounds (vsin and gnd) as shown in the schematic below. (Note: “**u**” key is for undo)



We use wires and wire labels to create connections between component pins.

- Press "**w**" on your window to bring the wiring tool. Connect all components as shown in the schematic.

The default name for each wire is "netXX", which is not descriptive at all. It is good practice to give meaningful names to your wires.

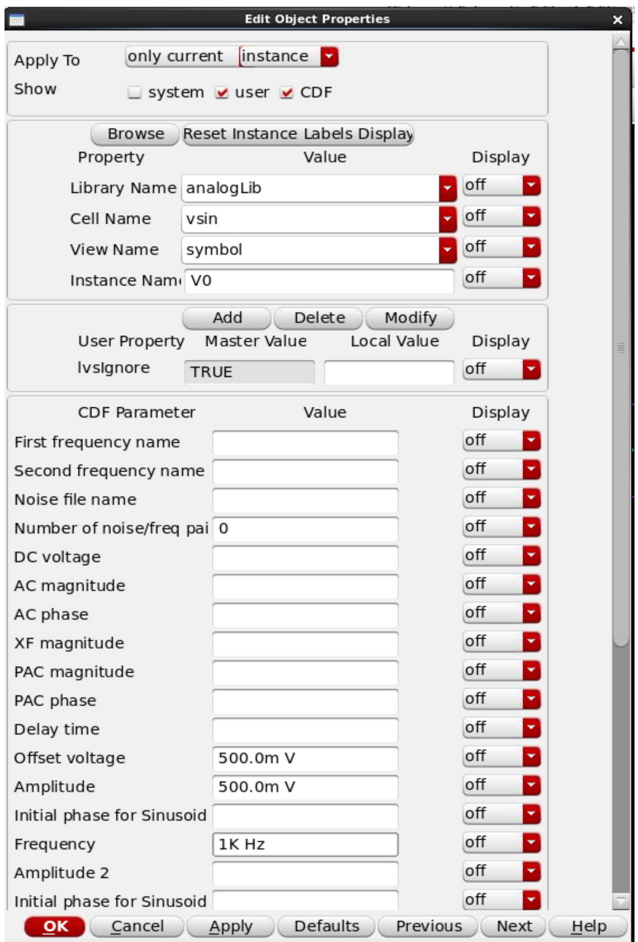
- Press "**l (this is lowercase L)**" on your keyboard to bring the wire label dialogue. Name the "in" and "out" as shown in the figure. All ground nets connected to a gnd instance are named **globally as "gnd!" by default**, so you don't need to put a label on them.
- Once you made all the connections and labeling, select the voltage source (vsin) and press "**q**" which will bring the **"Properties"** dialogue.

Set the values so that the applied waveform has 0.5V mean, 0.5V amplitude, and 1 kHz frequency.

We are now ready to simulate this schematic.

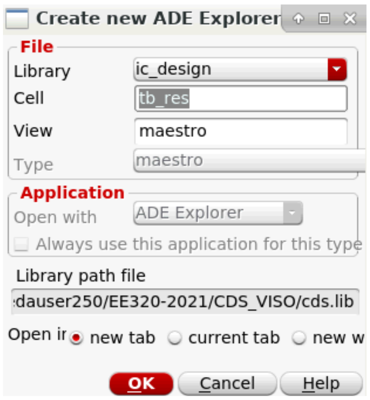
- Press "**Shift+x**" to check and save your schematic. This will check the schematic for any floating wires, open terminals etc. Note that the undo history is lost after you save.

- If there are any warnings or errors, you can view them by pressing "**g**".

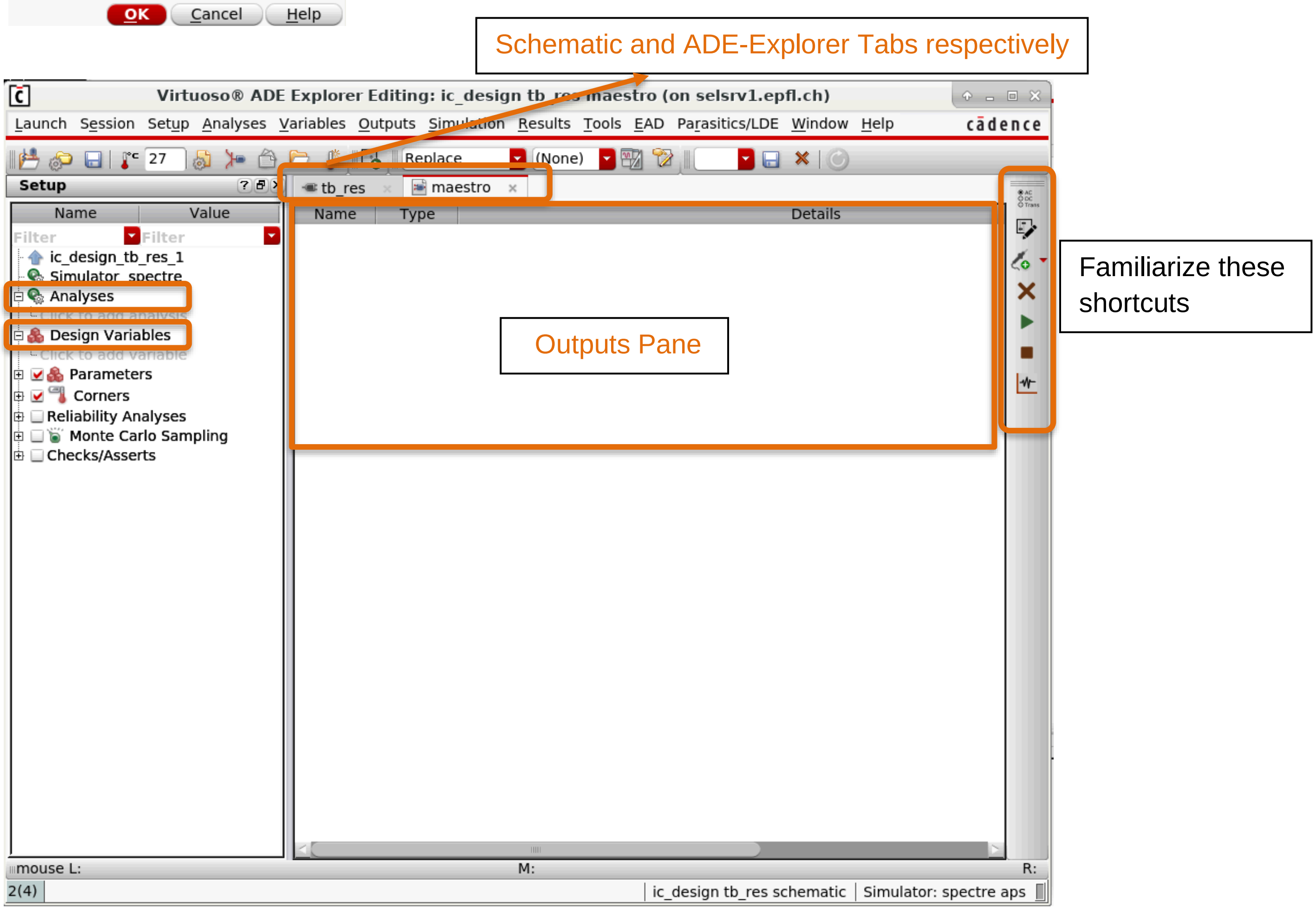


Transient simulation

On your schematic window, click **Launch>ADE Explorer**. In the next prompt, select **Create New View**. Keep the View as **maestro**, select **Open in new tab**, and press OK.

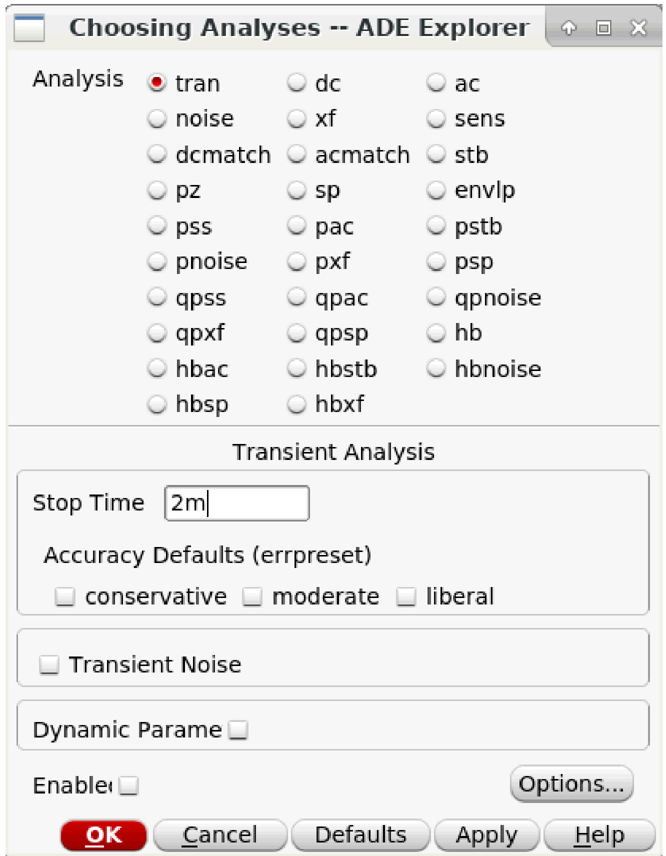


ADE stands for “Analog Design Environment”. One could also use ADE-L, ADE-XL, but of late, Cadence is asking its users to migrate to ADE Explorer/ADE Assembler (the support for ADE-L and ADE-XL might end in the near future!). For our simulations, ADE Explorer is sufficient. ADE Assembler can be used when running multiple tests on a design.

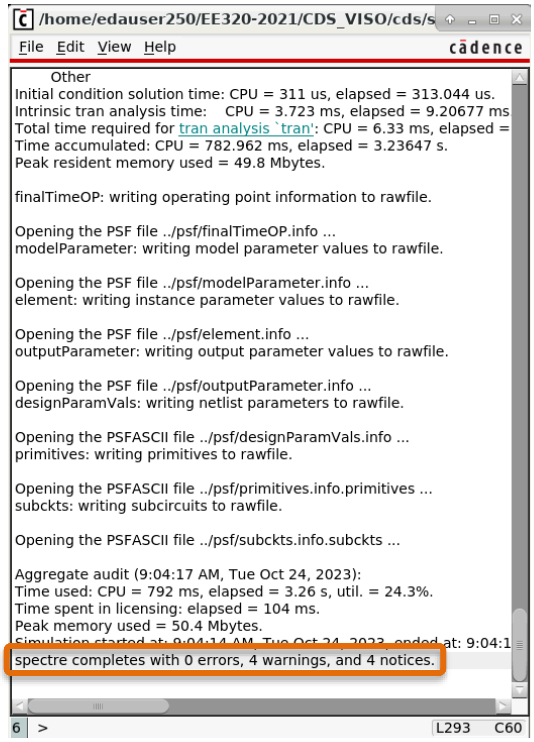


In the above window, familiarize with the **Analyses**, **Design Variables**, and **Outputs Pane**.

Now to start off our simulations, we first need to add an Analysis type. For this click on **Click to Add Analysis**. It should draw up a window like the one shown below. Here, you will see many analyses types, but for this exercise, we will only perform **"tran"** (transient) analysis. Set a stop time that will allow you to see two periods of the sine waveform. Click **OK**.

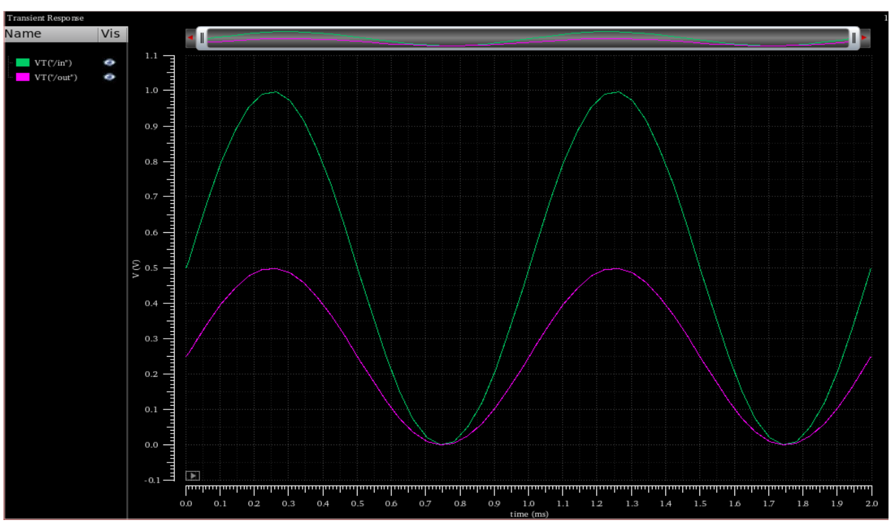


Now, click on the **green run play button (**  **)** in the shortcuts to run the simulation. Once you run the simulation, an output log (spectre.out) will pop up on your screen. Once the simulation is over, check the bottom of the output log to ensure that there are no errors. Warnings and notices are okay, but feel free to review them.



Displaying transient results

Click **Results>Direct Plot>Transient Signal**. Click on “in” and “out” nets in your schematic. Hit **“Esc”**. This will show the selected net voltages varying over time. When running in ADE-Explorer for the first time, you may have to undock your Viva waveform window (follow the instructions on the prompt).

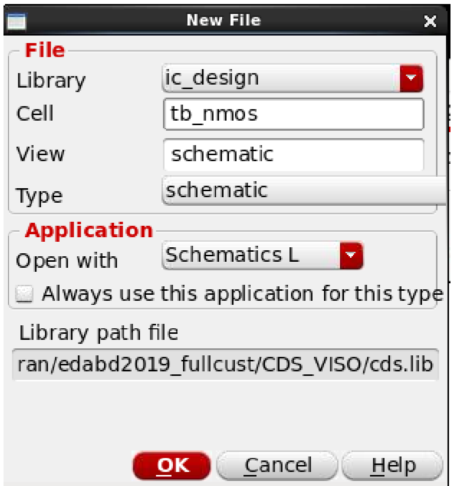


Change the resistor values to have a division factor of 3, then repeat the simulation to verify it.

2. MOS I-V curves and small-signal parameters

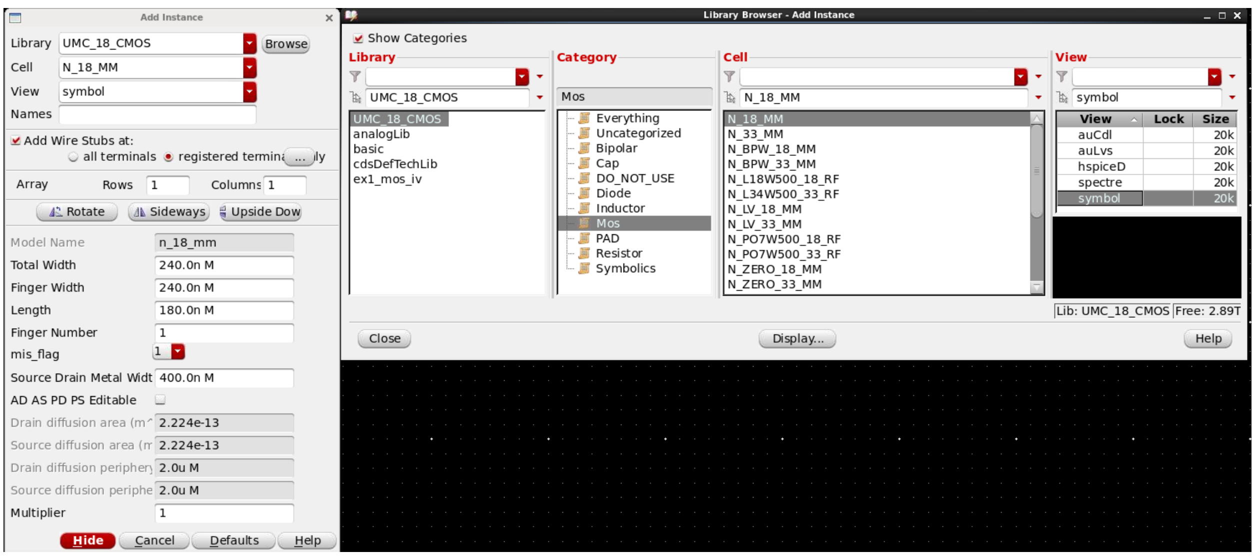
Creating a test bench schematic for characterizing an NMOS

In your library manager, select your new library and click **File>New>Cellview**. Select **"schematic"** as the type of your cellview. Give a meaningful name to your schematic. We suggest "tb_nmos". Click OK.

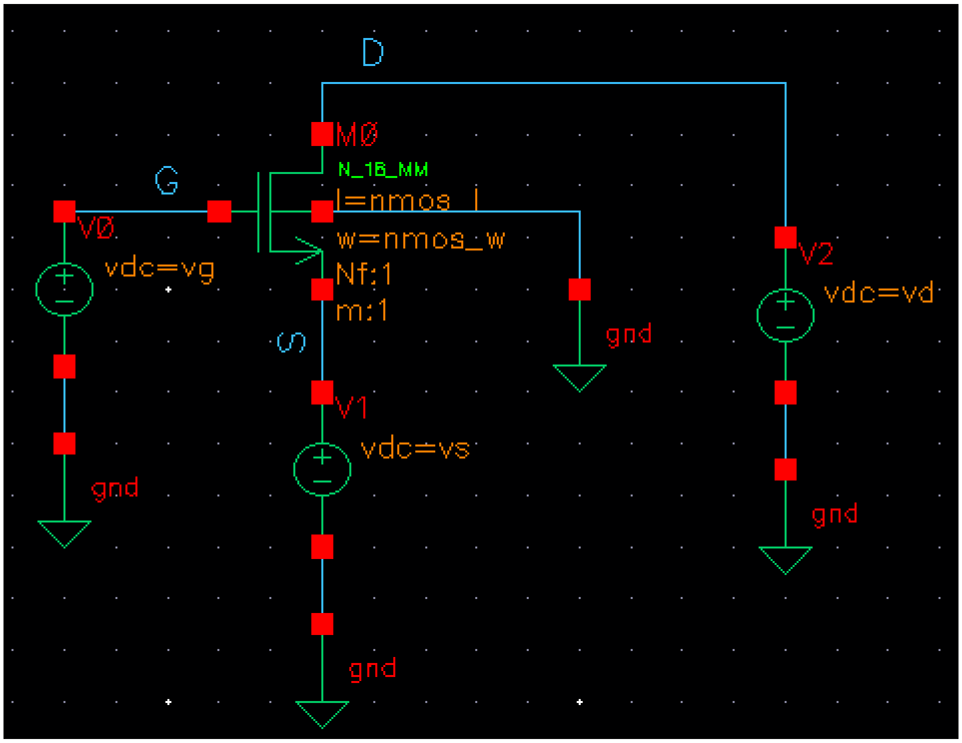


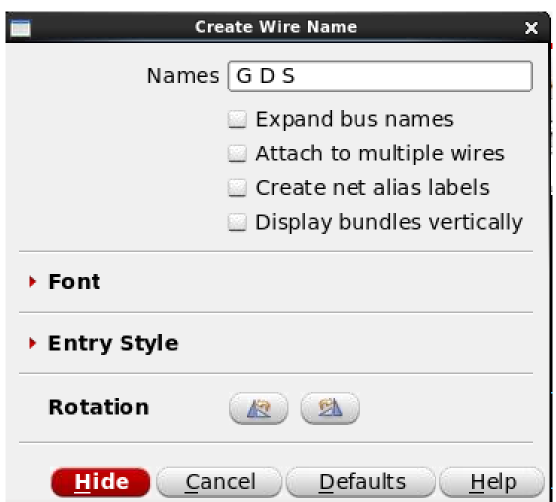
- Press the key "**i**" on your keyboard, which will open the **"Add Instance"** dialogue.

Click browse, select the library UMC_18_CMOS and select **"N_18_MM".** (You will notice different flavors of transistors. For our transistor "N_18_MM", "N" denotes NMOS. "18" denotes that the transistor can withstand a maximum of 1.8 V on its terminals. "MM" means that the transistor is to be used in low frequency (mixed mode digital and analog circuits). "RF" transistors are characterized for high frequencies.)



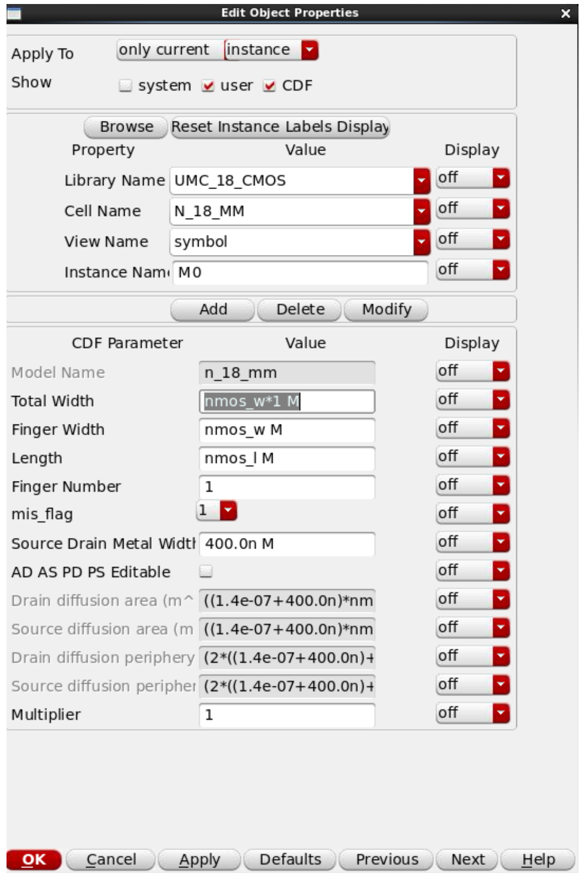
Place the transistor on your canvas. Then browse to analogLib and place the independent voltage sources and grounds (vdc and gnd) as shown in the schematic below. Again, press l (it is lowercase L) to create labels for wires.





- Once you made all the connections and labeling, select the transistor and press "**q**" which will bring the **"Properties"** dialogue.

Name the instance as M0. Here you can see the parameters that you can control as the designer, which are mainly **"Finger Width"** and **"Length".** You can enter real numbers like 2µ or 180n, or you can set them as custom parameters like **"nmos_w"** and **"nmos_l"**. These parameters can be changed in simulation without having to go back to the schematic each time during an iteration. Do the same for the voltage sources and set their dc voltage to parameters to “vg”, “vd”, and “vs”.



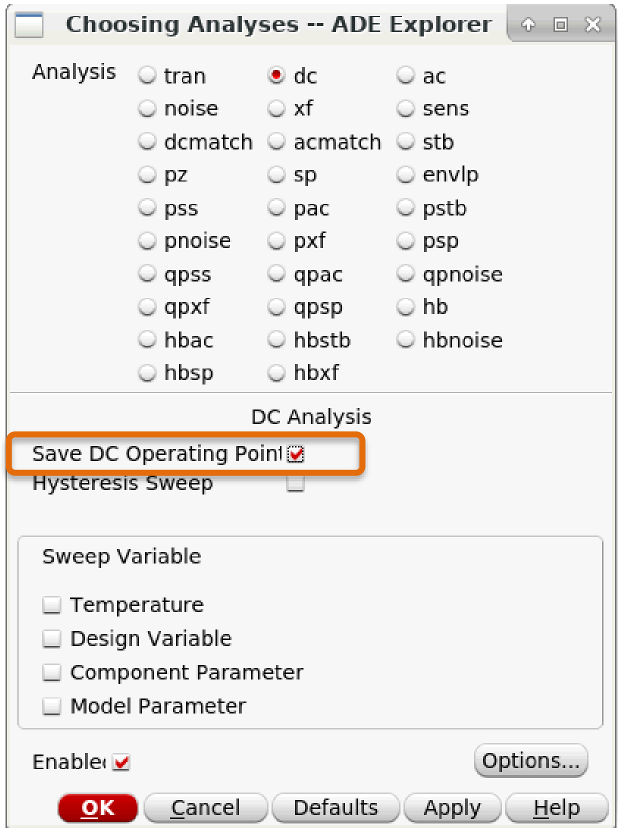
We are now ready to simulate this schematic.

- Press “**Shift+x**” to check and save your schematic.

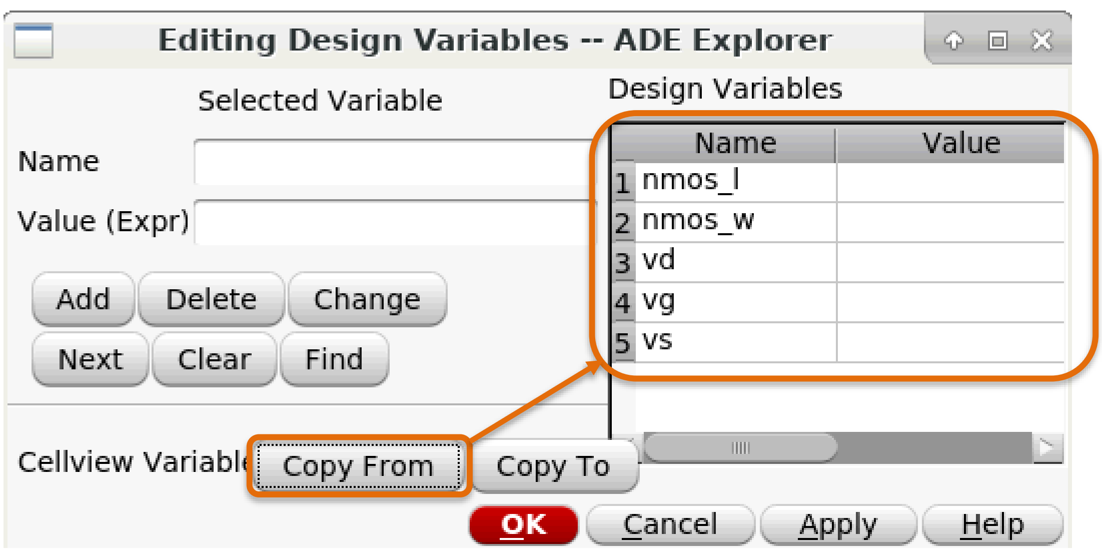
DC simulation

On your schematic window, click **Launch>ADE Explorer**. In the next prompt, select **Create New View**. Keep the View as **maestro**, select **Open in new tab**, and press OK.

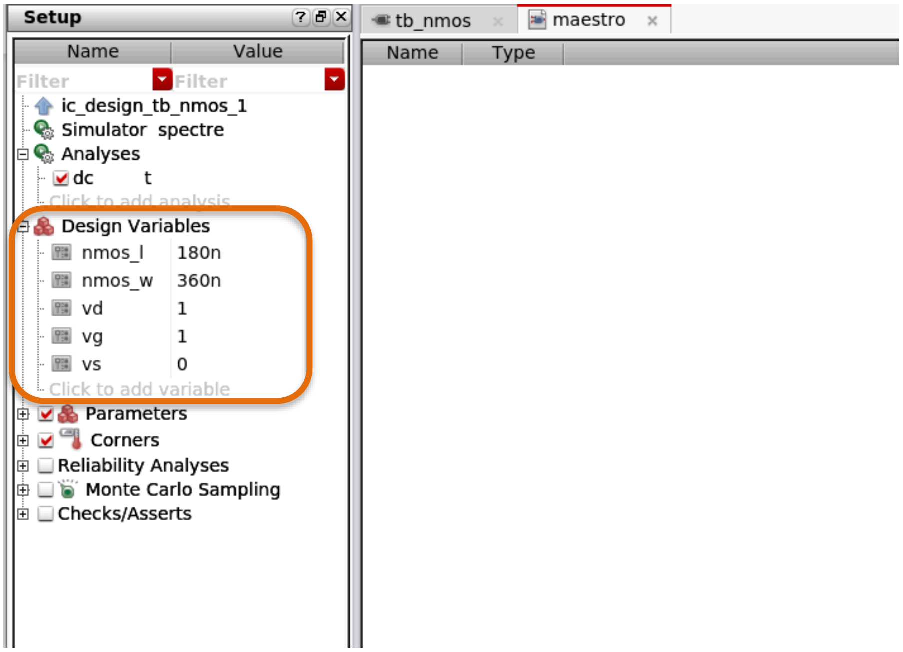
Click on **Click to add analysis**. As you have seen, there are many analyses available, but for this exercise, we will only perform **“dc”** analysis. **Don’t forget to check the “Save DC Operating Point” box!**



Below the Design Variables, click on **Click to add variable**. Select **"Copy from Cellview"**.



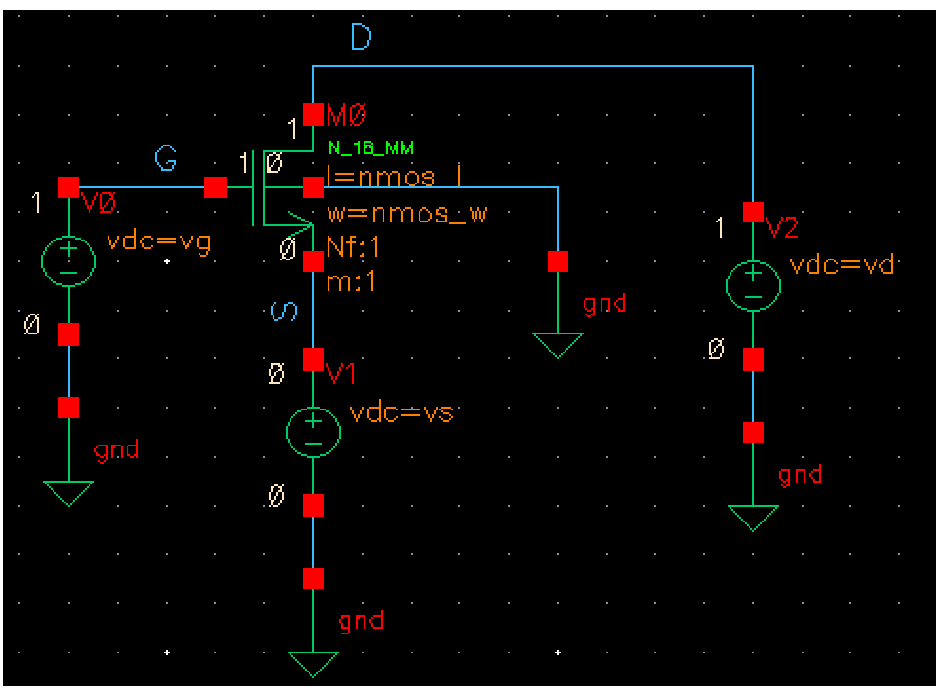
This will load the component parameters you defined earlier. Set the values for your variables as shown below.



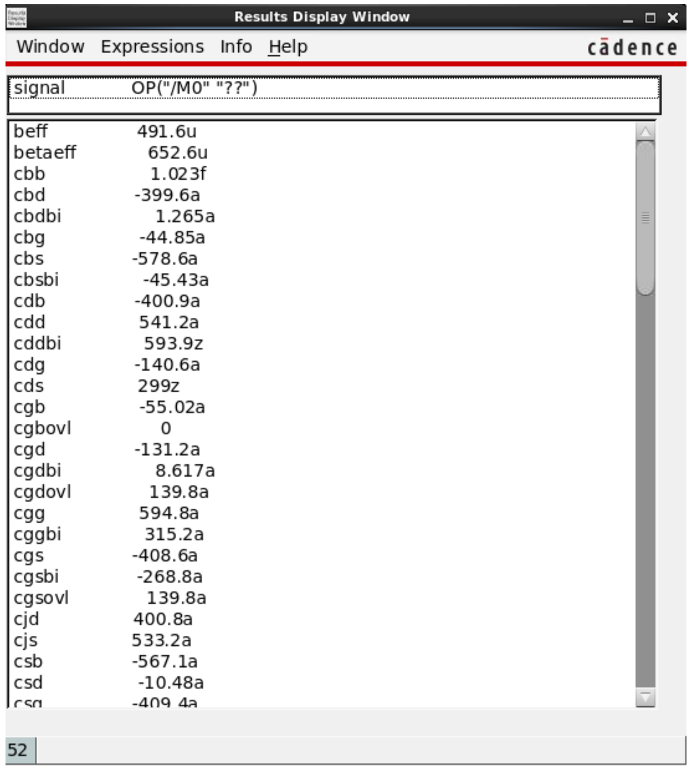
Click on the **green play button (** **)** to run the simulation.

Analyzing DC results

Click **Results>Annotate>DC node voltages**. This will annotate DC node voltages on your schematic. Make sure that the voltage on each terminal is correct.



Click **Results>Print>DC Operating point**, then click on the transistor M0. In this list, you will find the small signal parameters that are derived from the transistor model and the bias point.



Find and note down the following parameters:

- beff, id, vth, gm

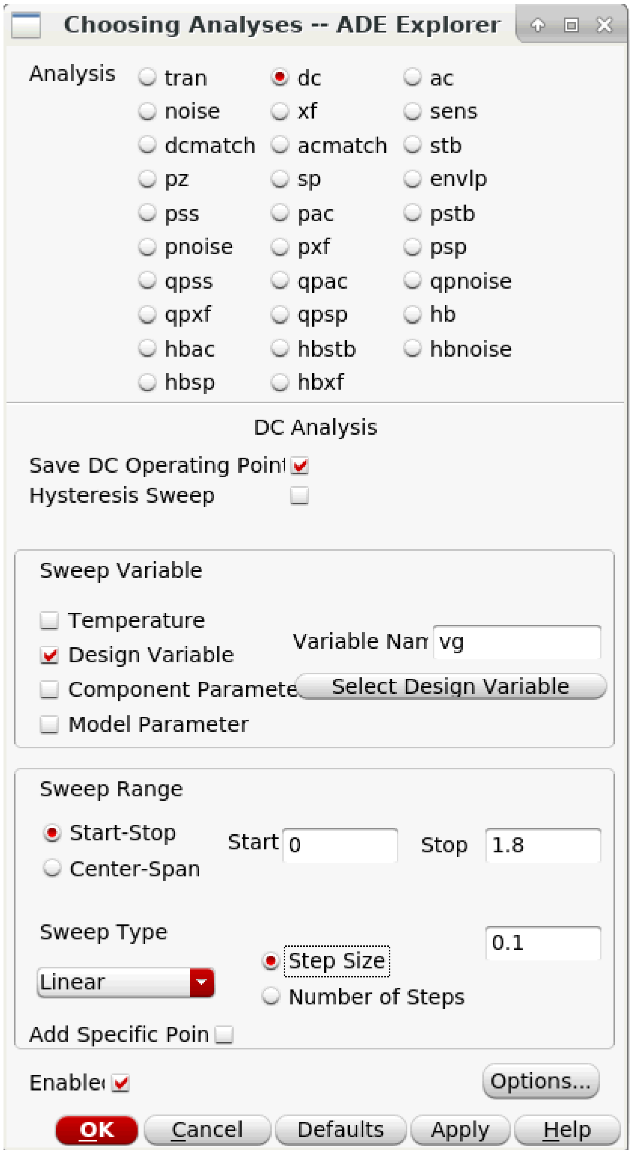
"beff" is the effective beta parameter which is equivalent to µCox(W/L).

Calculate "id" with the simple MOS equation and compare it with the id you obtained from the simulator.

DC sweeps

MOS regions

Double-click the dc analysis in the ADE Explorer Analyses box. Create the setup below for a DC sweep of "vg".

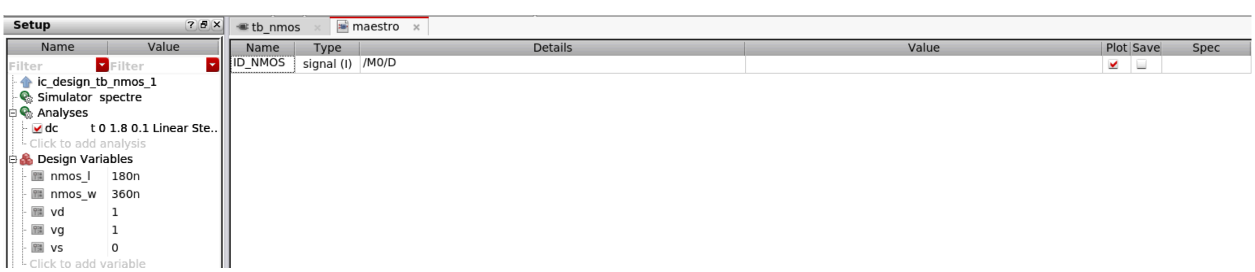


This lets you sweep the gate to source voltage from 0V to 1.8V in steps of 0.1V (since the source voltage, vs is fixed)

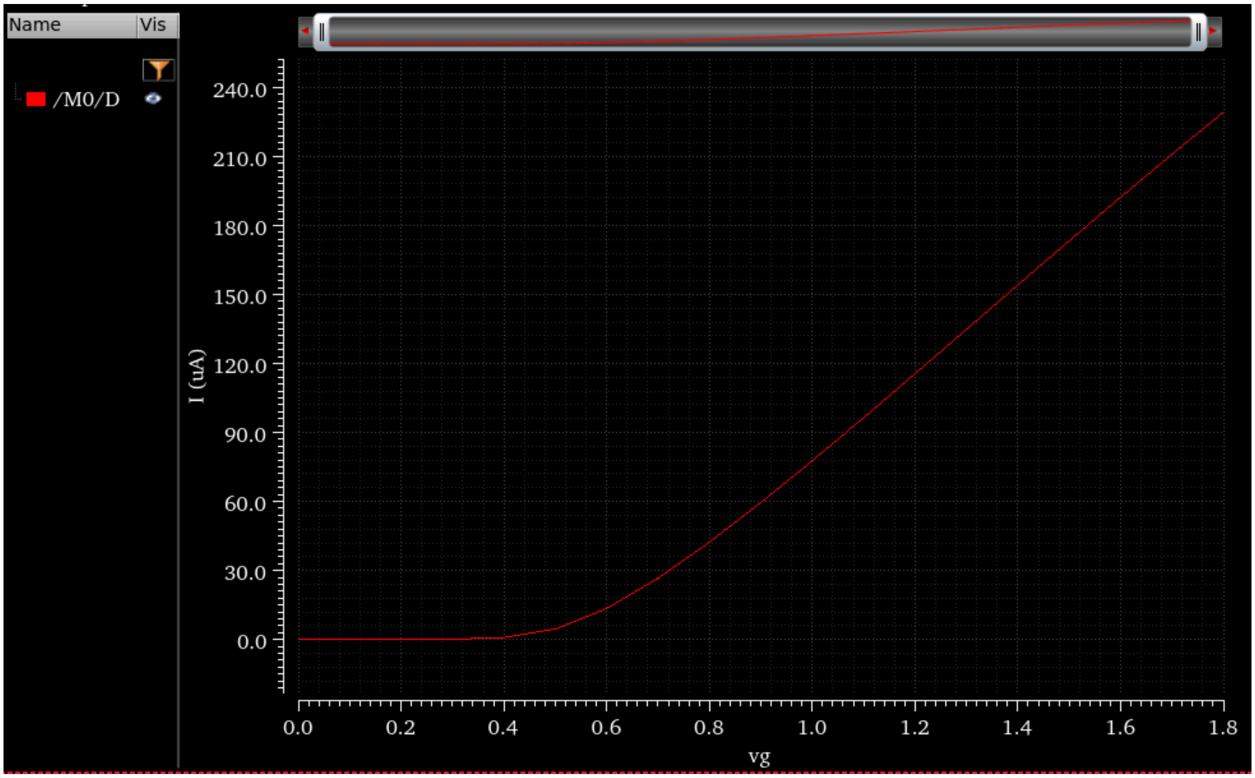
Why are we doing this?

Helps you get an idea of the operating region, current, transconductance, output impedance etc. as a function of the vgs.

Click **Outputs>To Be Plotted > Select on Design**. Select the drain terminal (the square red terminal) of the transistor M0, then hit **"Esc".** This creates an output in the ADE Explorer outputs box as shown below. You can also add a name like **"ID_NMOS"** if you want.



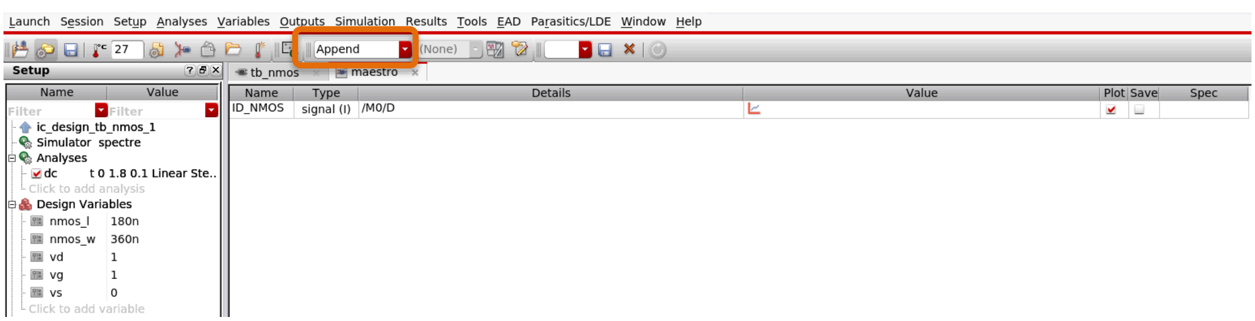
Run the simulation and the output will be plotted when the simulation ends.

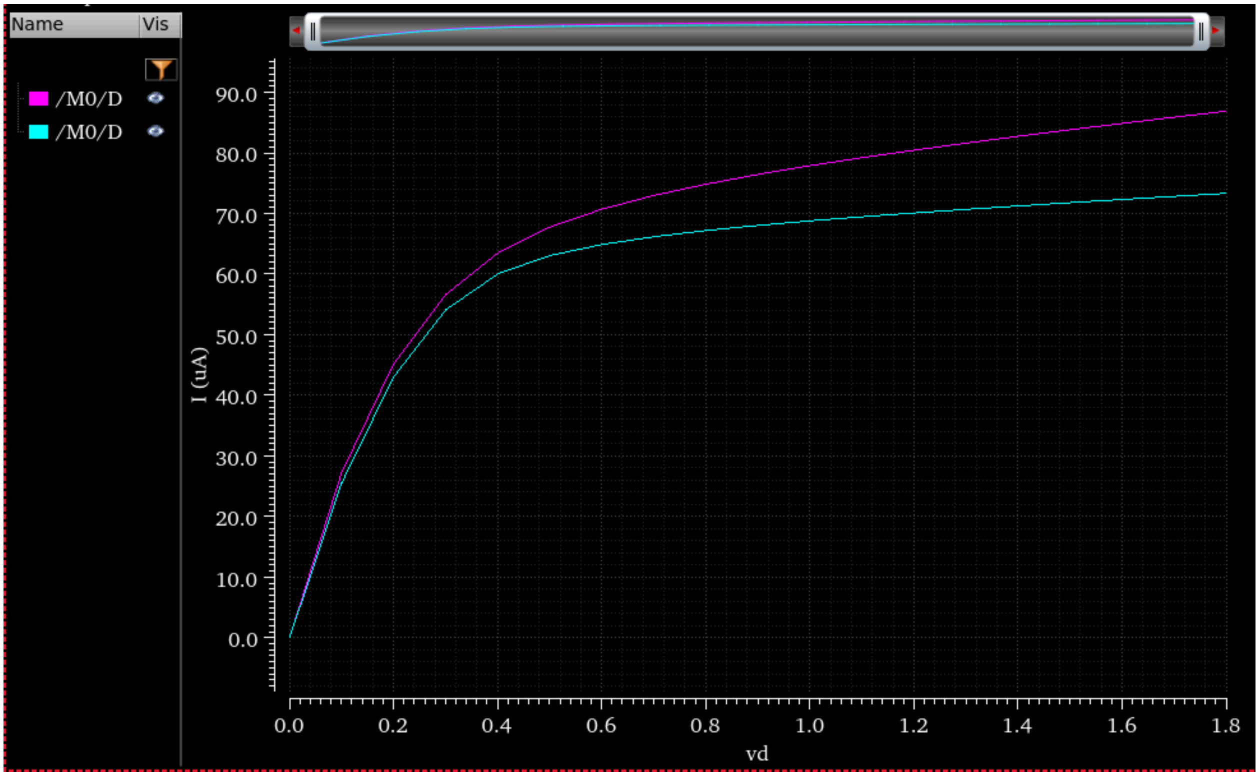


Identify the MOS operating regions in the plot you obtained.

Effect of drain voltage: **Repeat this exercise by varying just the drain voltage "vd" from 0V to 1.8V this time keeping all other terminals to a constant value.**

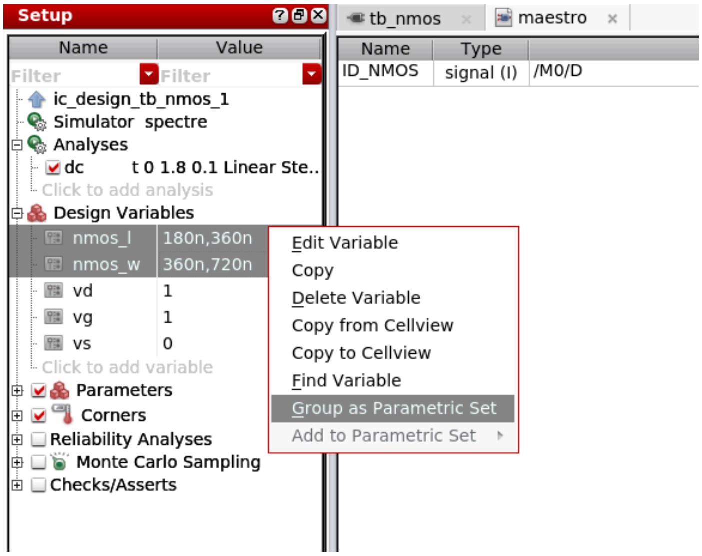
Now, **double the transistor width and length (keeping the W/L a constant)**, change the plotting mode to **"Append"** as shown below, and **repeat the simulation.**





Comment on the differences between the two plots. Provide your reasoning for the same.

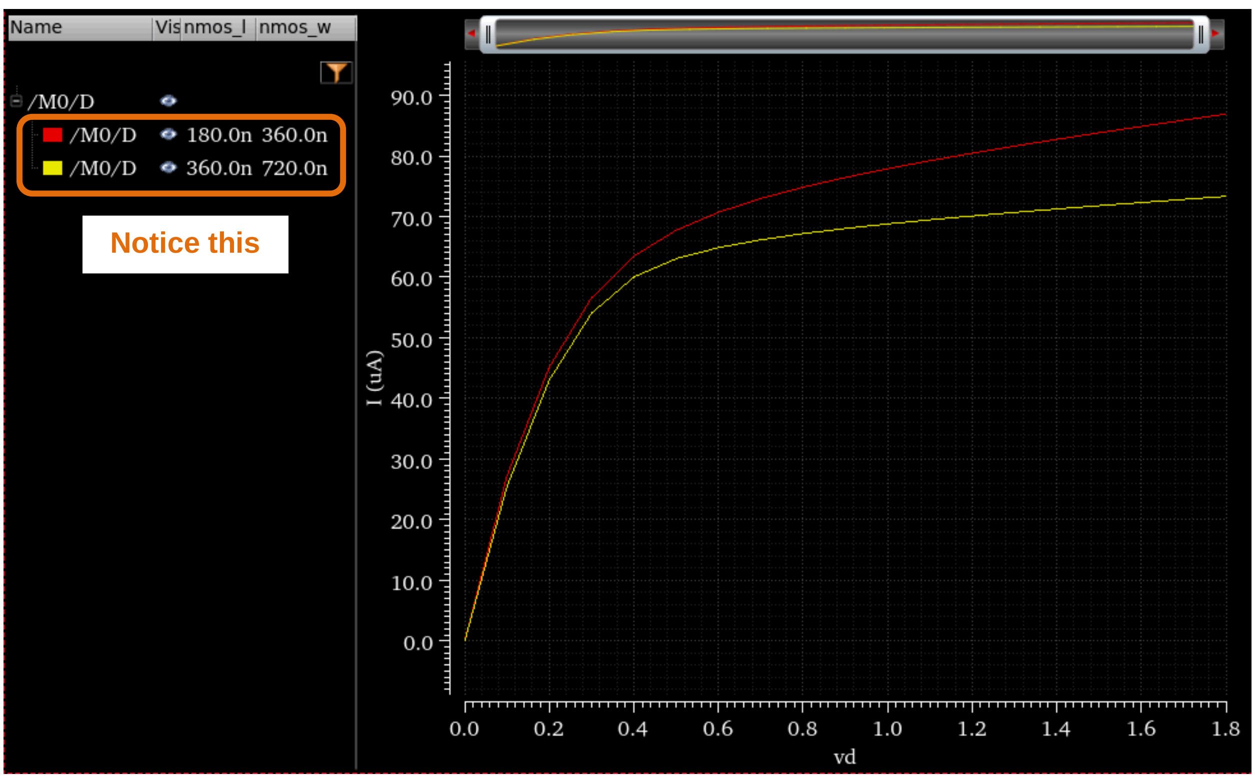
You can also do both of these cases as one single simulation. For this enter the two values of **“nmos_l”** as “180n,360n” and **“nmos_w”** as “360n,720n”. If you run this simulation, it will generate four combinations (2 values of nmos_l x 2 values of nmos_w). However, we are interested only in 2 of them. So, select both **“nmos_l”** and **“nmos_w”** by holding down the Shift key and clicking on them, then right click and select “**Group as Parametric Set**”. This will run the first value of “**nmos_l”** against the first value of “**nmos_w**”. The same is true for rest of the values.

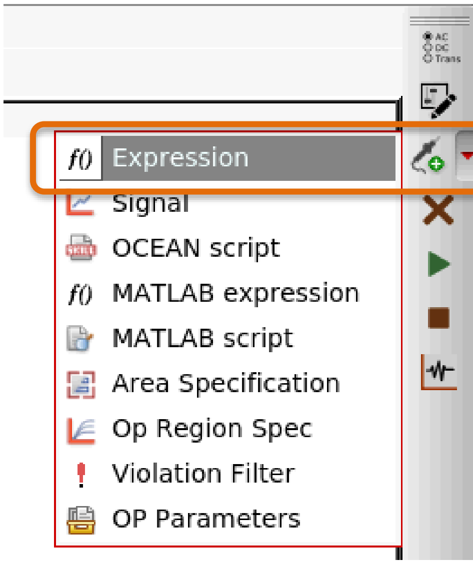


Once you do this, notice that both of these variable’s rows change color

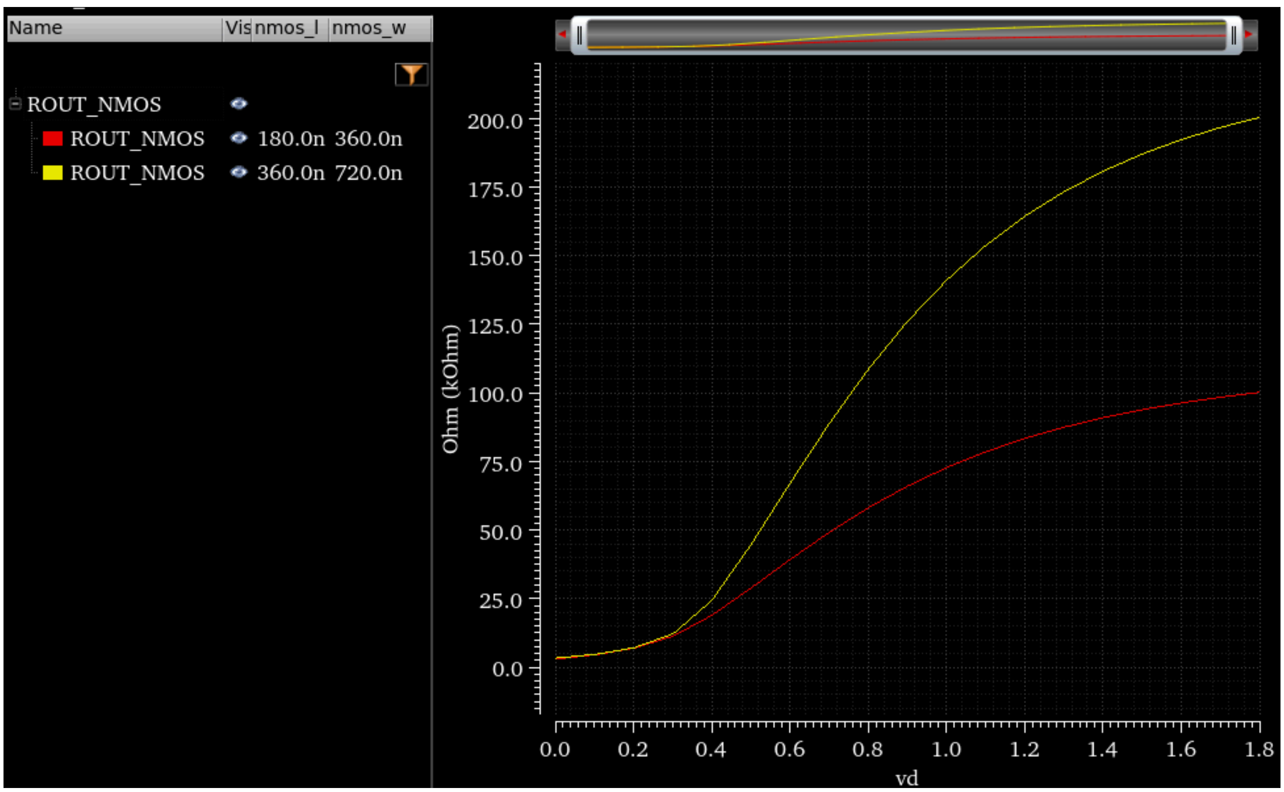
Now, again run the simulation.

You will notice that two run instances are now launched, and the waveform pops up as shown below.





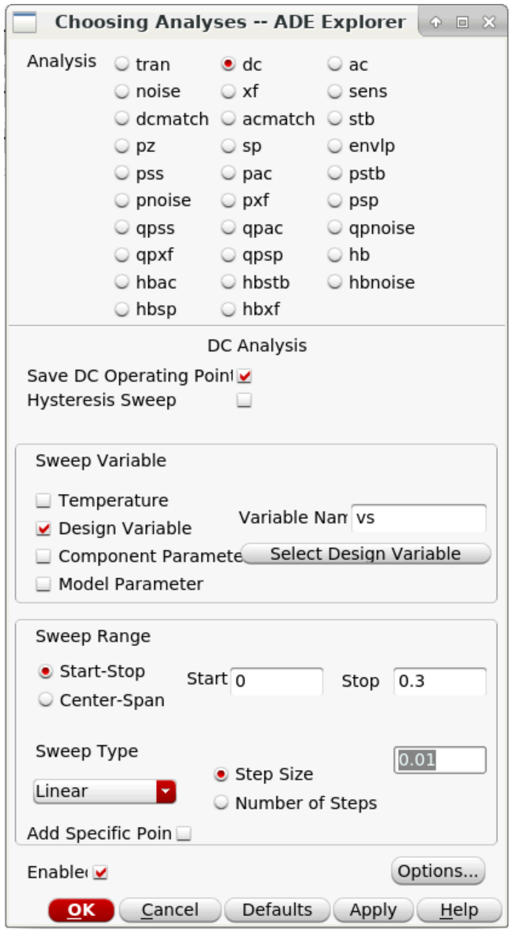
Measuring rout as a function of drain voltage: Right click and delete the output you created. Now select the **"Add Outputs"** option on the shortcut menu on the right and select **"Expression".** Now, type the expression **OS("/M0" "rout")** and hit **Enter.** Here, OS stands for Operating Sweep. You can also do this using the calculator. To know how, grab hold of your nearest TA (not literally! 😊). Now, run the simulation. This will plot the parameter "rout" with respect to the swept dc voltage when you simulate.



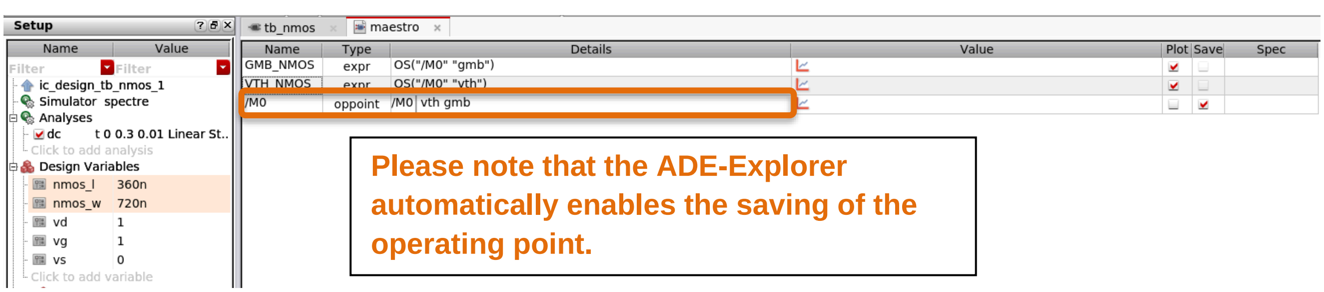
Compare the plots and provide your reasoning for this difference.

Effect of source voltage

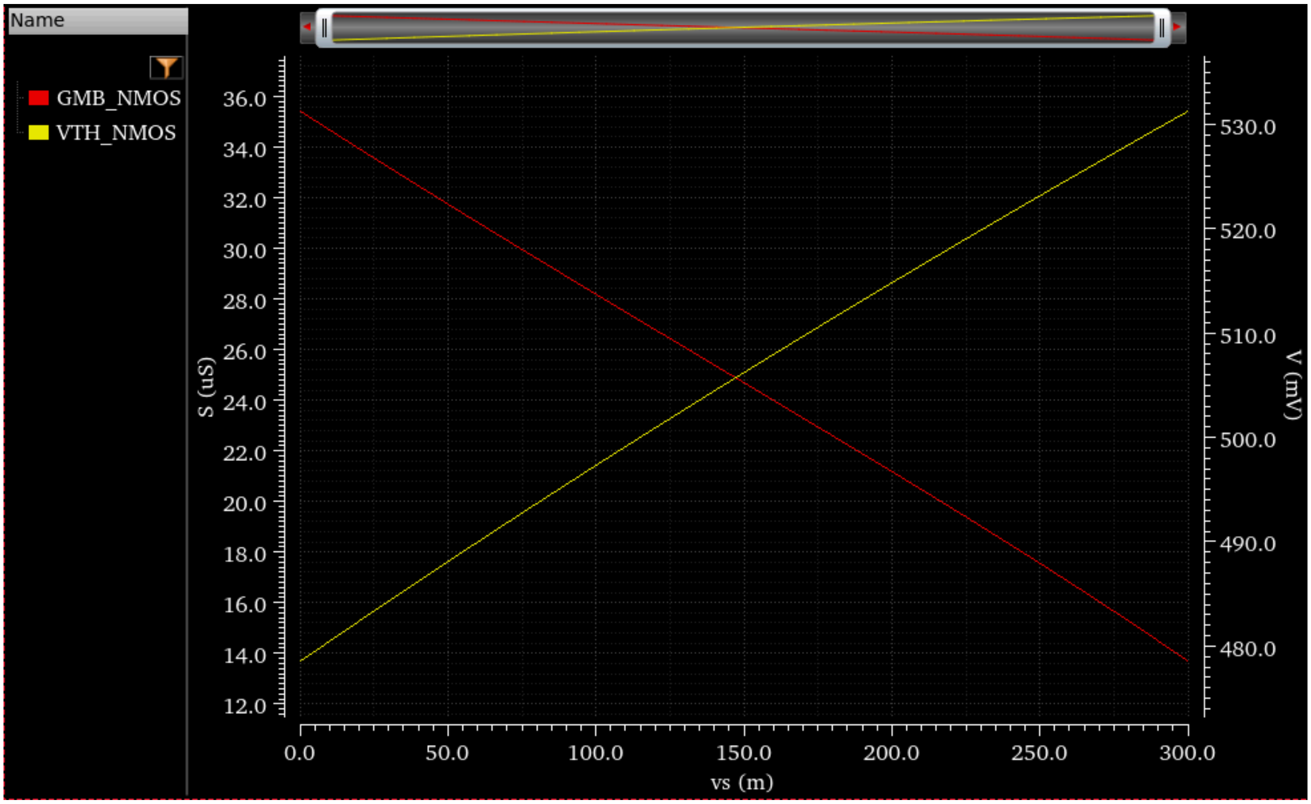
Change the DC analysis to the setting below for a sweep of “vs”.



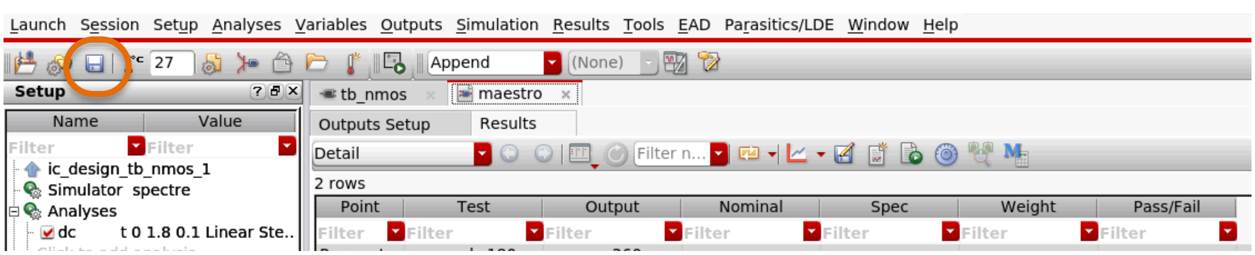
Delete all outputs and add **OS(“/M0” “vth”)** and **OS(“/M0” “gmb”)** as the new outputs. (Again, this is easier to do from the calculator. Haven’t caught hold of a TA yet? Catch them, catch them all!). Also run it just for **“nmos_l” = 360n** and **“nmos_w” = 720n.**



Run the simulation and comment on the plots you obtain.

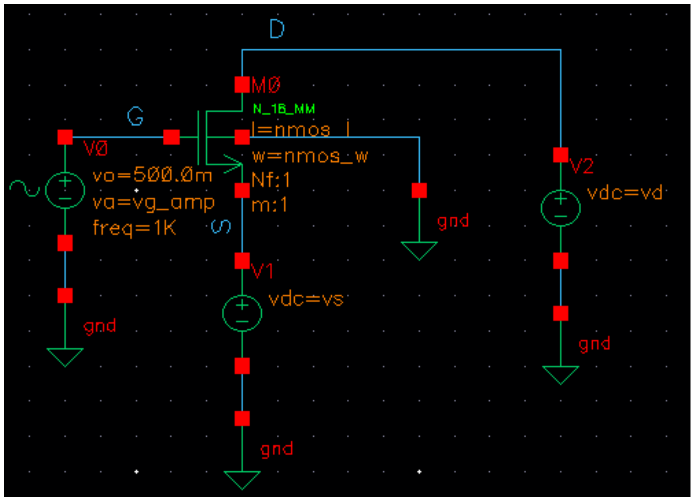


It is always a good idea to save your maestro window by clicking the save button as shown below. (Yeah right, the apostrophe in DIDN'T is not in the right place)

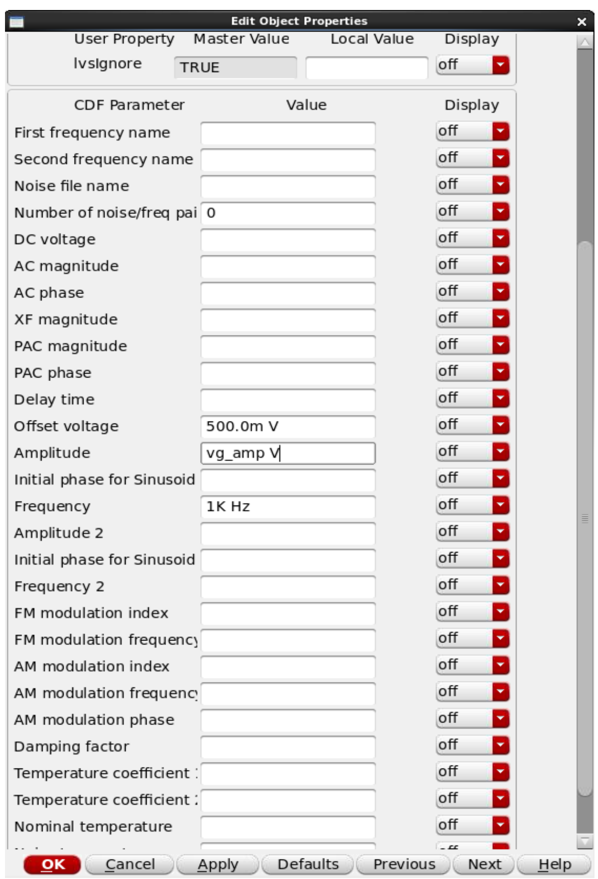


3. Bias dependency of small-signal parameters

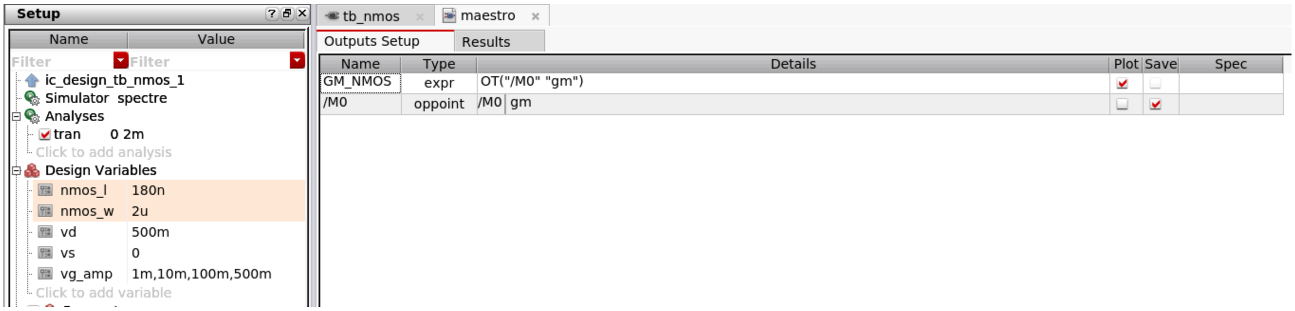
In the schematic window, replace the dc gate voltage source by a sine generator (vsin) as shown in the figure.



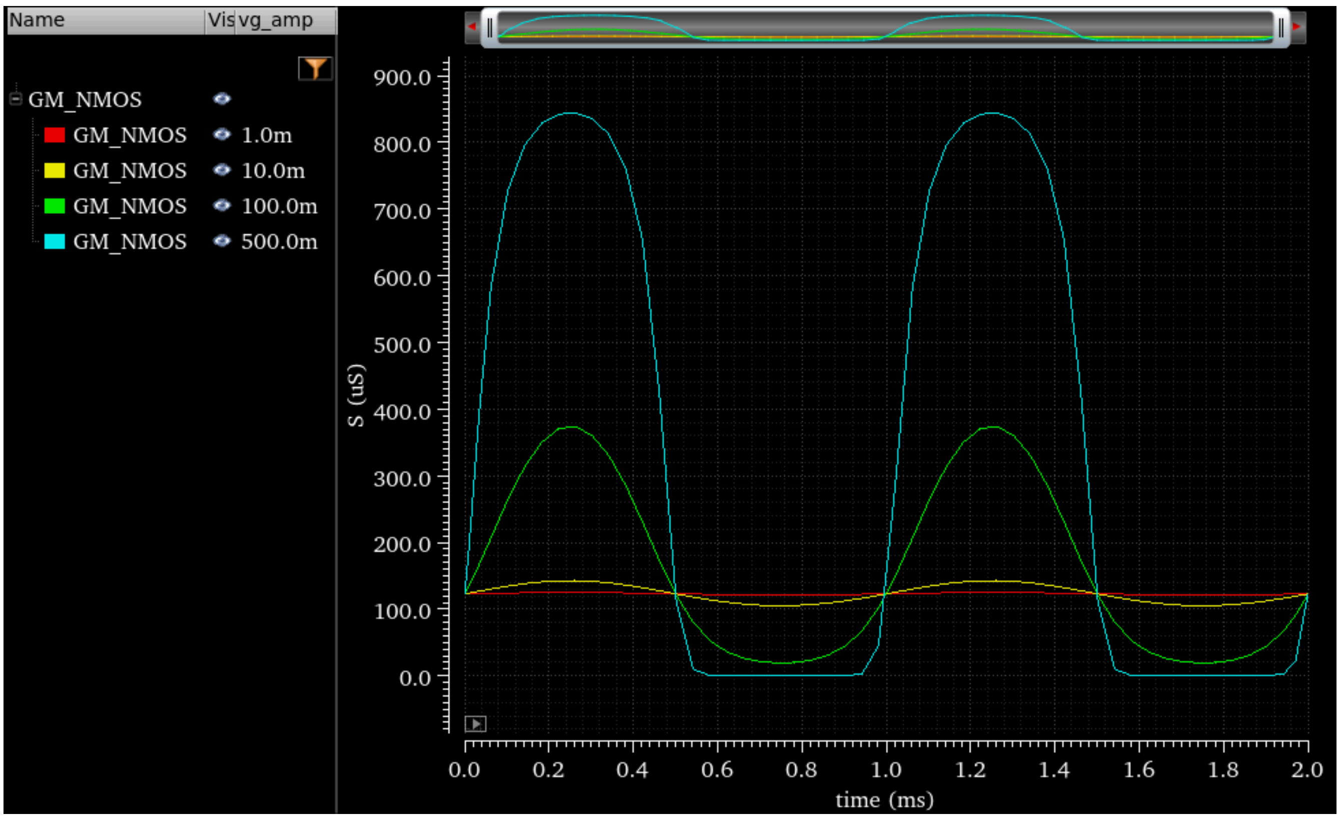
Set the properties of the sine generator as shown below. Check and save by pressing **“Shift + x”**.



Switch to the ADE-Explorer window and set up a transient analysis with a 2ms stop time. Copy the variables from cellview and set them to the values as shown below. Add OT("/M0" "gm") as the output. Here, OT stands for Operating Transient. Set the other Design Variables as shown below.



Simulate with vg_amp set to 1m, 10m, 100m and 500m. **Comment on the plots you obtain**.



4. Take-home exercises (optional, but you better do it! ☺)

In this first tutorial, we used some basic but important features of Cadence Virtuoso to see the textbook concepts on a real transistor. Please do the following exercises to get more comfortable with Virtuoso before the next session.

a) Repeat this tutorial for a PMOS device (P_18_MM in UMC_18_CMOS library).
b) For the same transistor length and drain current, simulate, and verify that a PMOS has a higher r_{out} than an NMOS transistor. Try to reason why.
c) For an NMOS transistor in the saturation region, increase the transistor width W by 5% (this is emulating the change in the MOS geometry post-fabrication) and simulate the change in drain current for the same V_{GS} and V_{DS}. How can the drain current be made less sensitive to an unavoidable change in the width of the MOSFET?