Design and DC Simulation of an Inverter

For information on how to setup your account and Cadence environment, and on how to start running Cadence with NCSU cadence design kit, please refer the previous lab material, Cadence Tutorial-1.

For a full custom cadence design (as opposed to a semi-custom or coded/synthesized design), the design process starts by creating a schematic design. The schematic is then simulated to verify operation and optimize performance. As the next step, a layout design of the schematic is generated and checked for physical Design Rule Check (DRC). A netlist is then extracted from the layout design, and a Layout vs. Schematic (LVS) comparison is run to ensure the layout matches the schematic. Finally with the extracted netlist from the layout, the layout design is simulated to observe the effect of *parasitics*. This is called *post layout simulation*, and it reflects more close performance of the design when it comes in a fabricated chip.

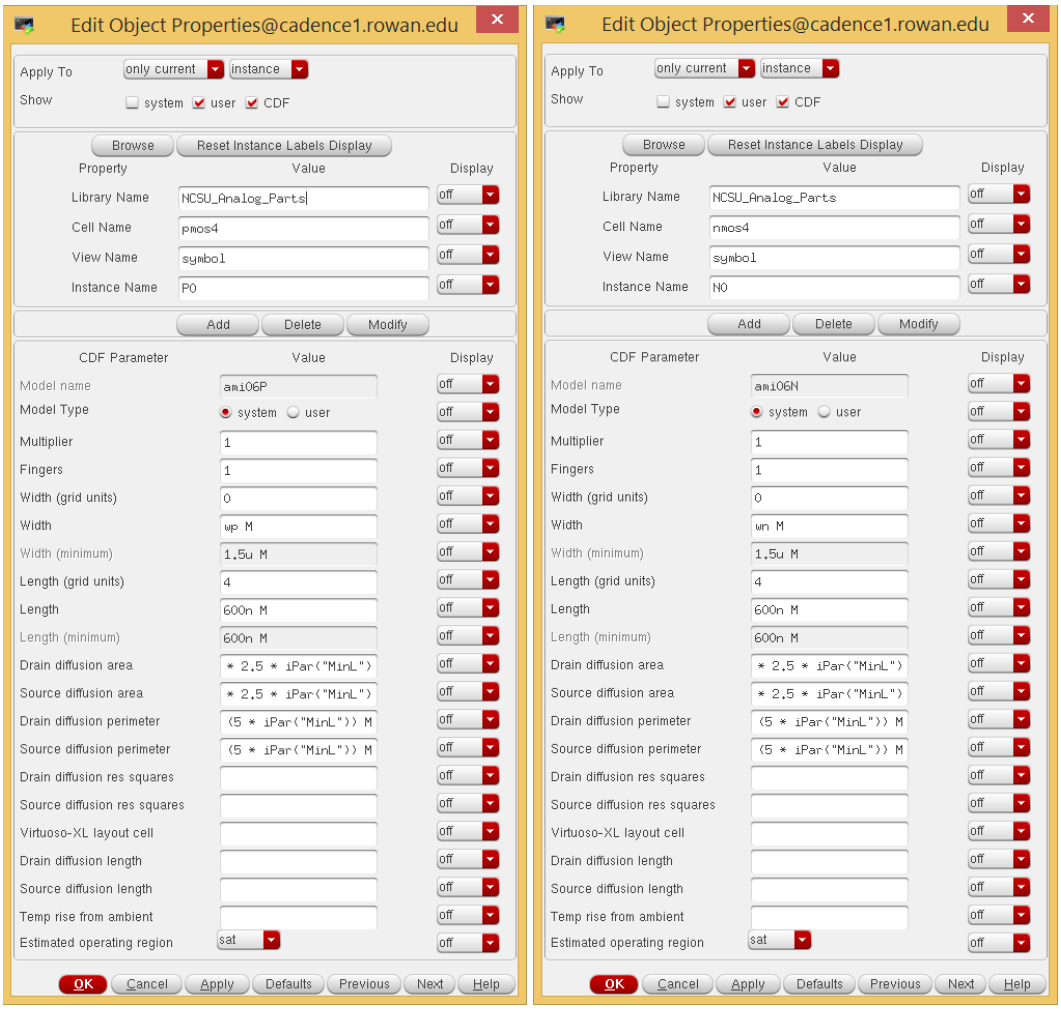
As the first step for a full custom cadence design, this lab will deal with a schematic design for the simplest CMOS logic gate, inverter, and perform DC simulations on the schematic to verify its functionality and Voltage Transfer Curve (VTC).

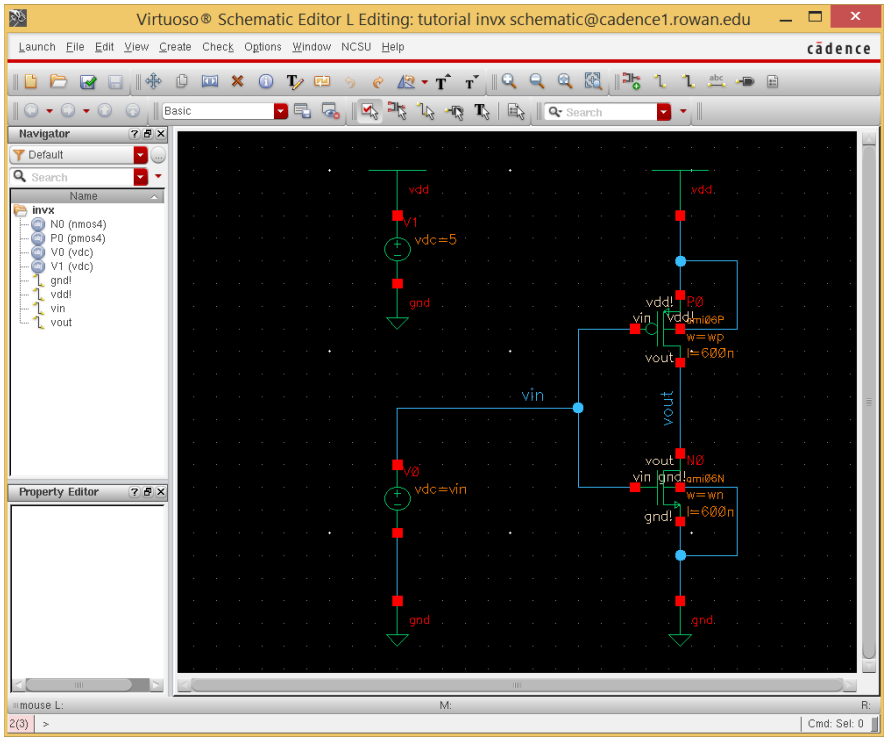
1) **Launch Cadence**:
    a. Connect to the cadence server (cadence1.rowan.edu), using your Rowan network ID and password (refer previous lab material for more info.).
    b. After logging in, set the Cadence environment by sourcing cds.setup file:
    >> source cds.setup
    c. **Move to your Cadence work directory** and then **start Cadence** there:
    >> virtuoso &
    d. You will get two windows, **CIW** and **Library Manager**, and the Library Manager should list three NCSU libraries and the one you created, e.g., **tutorial**, during the last lab.
2) **Draw a Schematic** for a CMOS inverter:
    a. From the **Library Manager**, create a new **Schematic** view, with a name **invx**, *under your work library*. Choose **Schematics XL** for the *Application/Open with*, and **check** for the *checkbox.*

b. From the menu bar, choose *Create* -> *Instance*, or type its hotkey ***i***. Then navigate for the components you want to place, specify its property, and then place it onto the *Schematic Editor*.

Use the components **nmos4** and **pmos4** from the library **NCSU_Analog_parts** for nMOS and pMOS transistors, and choose other ideal components from the library **analogLib**.

Below are example properties chosen for transistors with variables ***wn*** & ***wp*** for widths of nMOS & pMOS, respectively. (Note that the minimum transistor width for the AMI 0.6um CMOS is 1.5um which is equivalent to 10 grid units that is 5λ, not 4λ.)





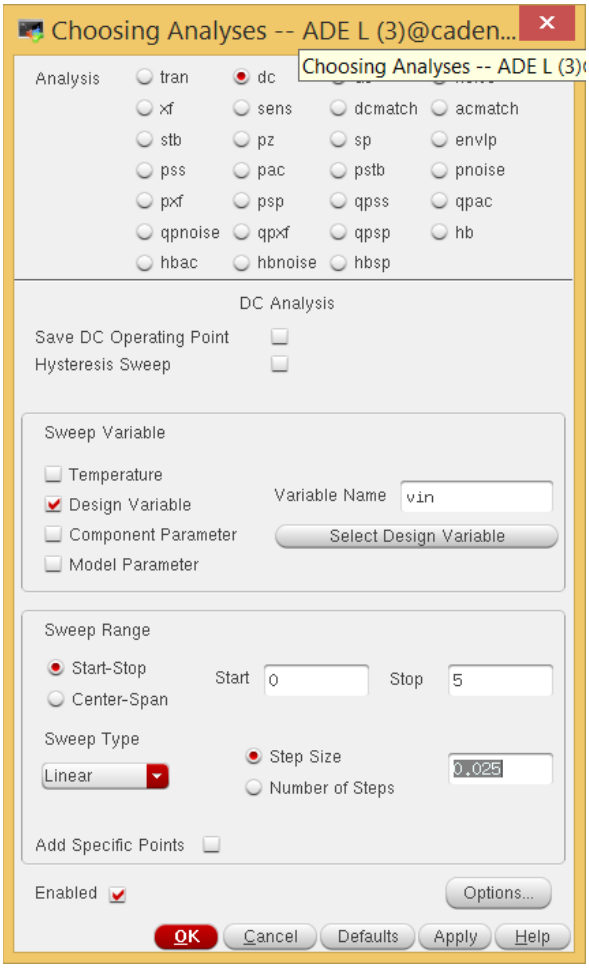
c. Don't forget to do **Check and Save** the schematic view, and make sure there is no error.

Useful Editing Hot Keys:

The following "hot keys" are available for the schematic editing tool.

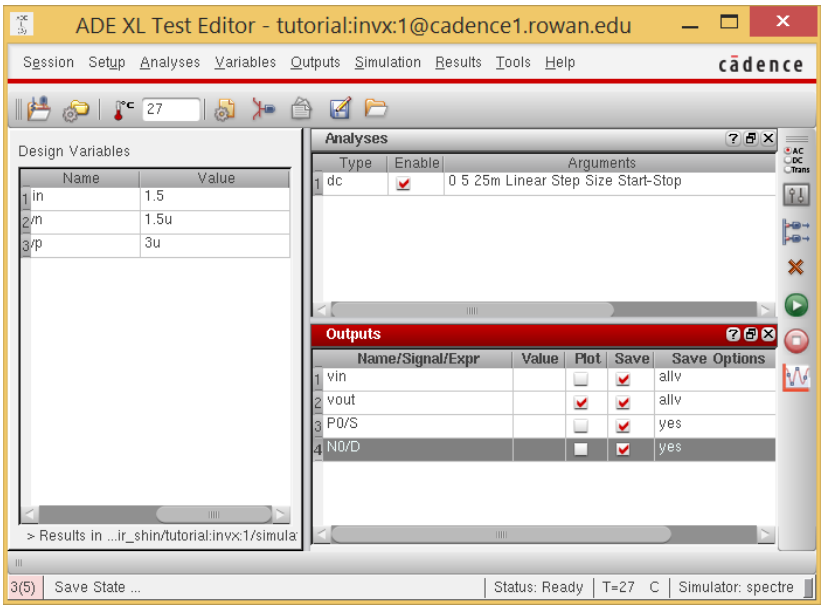
- Press '**i**' to add instances (components)
- Press '**q**' on the device/instance selected to edit properties of the device
- Press '**w**' to add wires
- Press '**f**' fit the schematic in your schematic window
- Press '**l**' to label a wire
- Press '**u**' to undo an operation in the schematic window
- Press '**m**' to move objects
- Press '**DEL**' key to delete objects
- Press '**p**' to add pins
- Press '**z**' to zoom in the window
- Press '**shift+z**' to zoom out the window
- Press '**Up**' and '**Down**' arrow keys to move up and down within a schematic
- Press '**ESC**' key to terminate an operation in the schematic window

3) Launch **Spectre simulator and setting** it up:
   a. In the **Schematic Editor**, go to Toolbar, and Launch **ADE GXL**.
   b. Open **ADE XL Test Editor**, by clicking **Tests** in the *Data View* panel.
   c. Load and define variables:
      From **ADE Editor**, choose *Variables -> Copy From Cellview*. It will get all the variables you defined in the Schematic Editor, ***vin*** , ***wp***, and ***wn***. Set their default values, e.g., vin=1.5V, wp=3u, and wn=1.5u.
   d. Simulation setup:
      From **ADE Editor**, choose *Analyses -> Choose*. You can choose a type of simulation and set conditions. Below is for a DC simulation over a range of variable sweep (***vin***=0:0.025:5).



   e. Set outputs:
      From **ADE Editor**, choose *Outputs ->To Be Plotted -> Select On Design*. This will activate the Schematic Window allowing you to pick which signals (nets/wires/pins) you would like to have plotted during the simulation. In the

Schematic Window click on the wire and/or pin that are the input and the output of the inverter. Typing the **ESC** key will complete the output selection.
Below setting is for a case where the nets **vin** & **vout** and the pins **P0/S** (source of pMOS) & **N0/D** (drain of nMOS) are chosen to save the input & output voltages and the currents through pMOS & nMOS, respectively. Among the outputs to be saved, the voltage ***vout*** is chosen to be plotted after each simulation.



f. **Save the simulation setup**:
The ADE state can be saved so that the settings do not need to be manually entered each time you want to simulate the design.

- To save the setting, in the **ADE Editor**, select *Session -> Save State*.
  The "Save As" field does NOT have to be a unique name for all designs, since the state is saved within the directory of the current cell view.
- To load a saved setting, in the **ADE Editor**, select *Session -> Load State*.
  You can load any saved state, by browsing and choosing the one you want to load. You will like this to save your time later on.

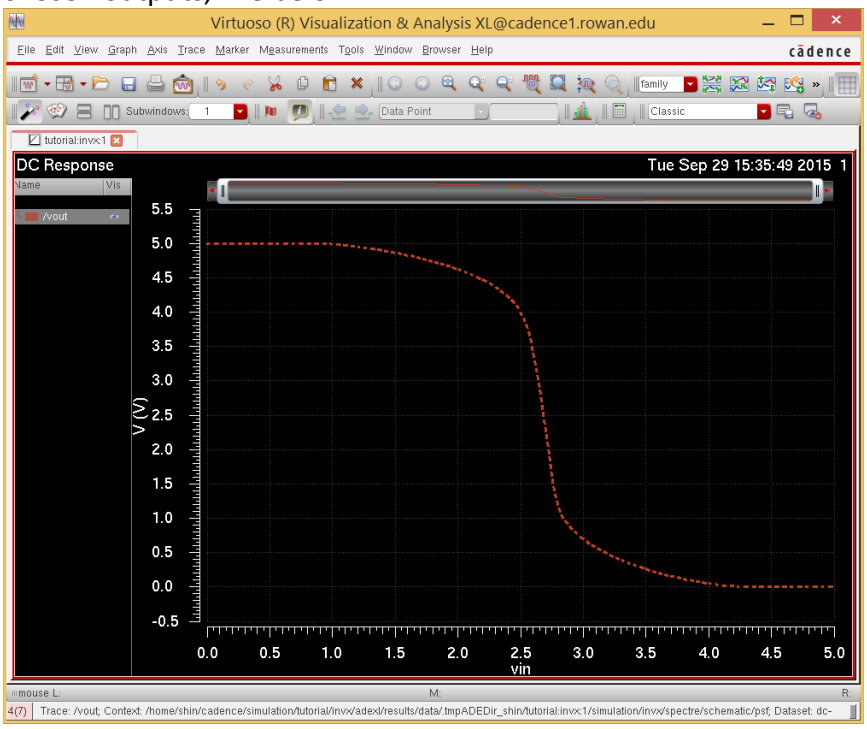
4) Run the simulation
   a. Extract netlist:

Before performing the simulation, you should create a netlist of your circuit, and it can be done in **ADE Editor**, by choosing *Simulation -> Netlist -> Create*. It will pop a window up that lists the netlist. You can close the netlist window.

b. Run simulation:

For a single simulation, you can simply push the ***green run button*** in the **ADE Editor** window. (Tip: The green run button is for 'netlist and run' which means clicking on the button firstly extracts the netlist and then runs the defined simulation, so you may skip the previous step of netlist extraction.)

When the simulation is completed, it will open up a result window for the chosen outputs, like below:



5) **Analysis with Calculator** (Noise Margin):

We're going to use the Calculator to compute the noise margin from the Voltage Transfer Curve (VTC) of the inverter. To open up with the Calculator tool, from the **waveform window** choose *Tools-> Calculator*. *Tips*: The Calculator works with a "stack" in which you first input one or more operands (waveforms) and then you perform an operation on them, which implies that in order to compute '3-2' you should stack '2' first then '3', and then compute '3-2' by pressing '-'.

To calculate the *Noise Margin*, you will need to handle the VTC. (For details about the definitions of noise margins, please refer the lecture material.)

Find and select ***vs*** from the Calculator, and then click the ***vout*** net from the schematic window. That will display **VS("/vout")** in the Calculator. Then search for a special function ***deriv*** and click on it, that will display **deriv(VS("/vout"))**. To calculate V_{IL}, V_{IH}, V_{OH}, and V_{OL}, following expressions can be used:

V_{IL}: *cross(deriv(VS("/vout")) -1 1 "either" nil nil)*

⇨ Finds the lower input voltage at which the VTC slope equals -1

V_{IH}: *cross(deriv(VS("/vout")) -1 2 "either" nil nil)*

⇨ Finds the upper input voltage at which the VTC slope equals -1

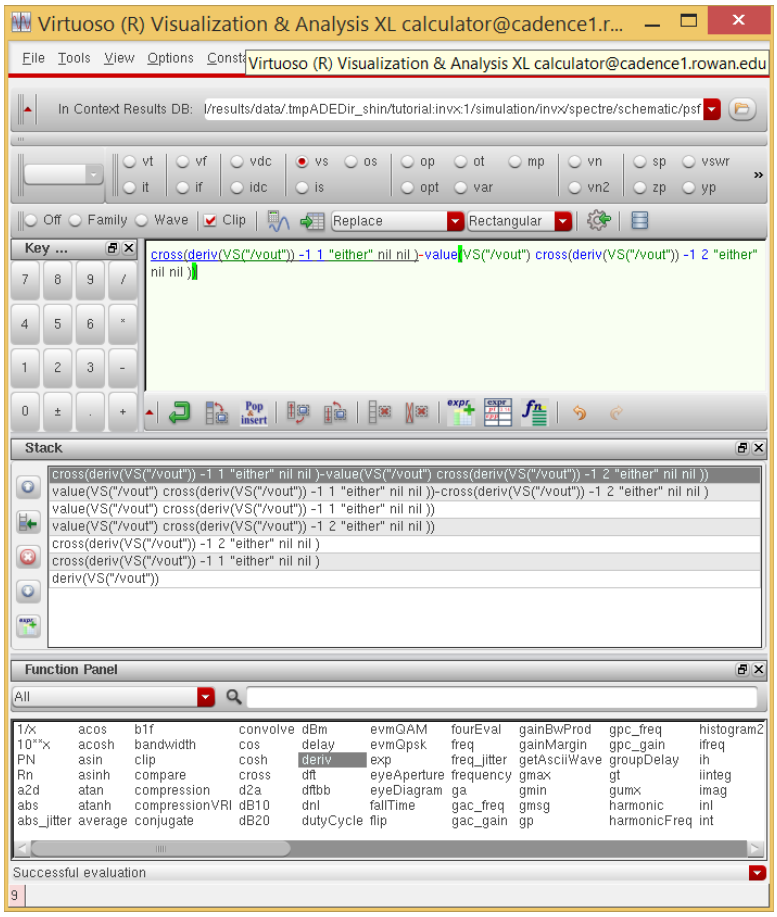
V_{OH}: *value(VS("/vout") cross(deriv(VS("/vout")) -1 1 "either" nil nil))*

⇨ Finds the vout for the input vin=V_{IL}

V_{OL}: *value(VS("/vout") cross(deriv(VS("/vout")) -1 2 "either" nil nil))*

⇨ Finds the vout for the input vin=V_{IH}

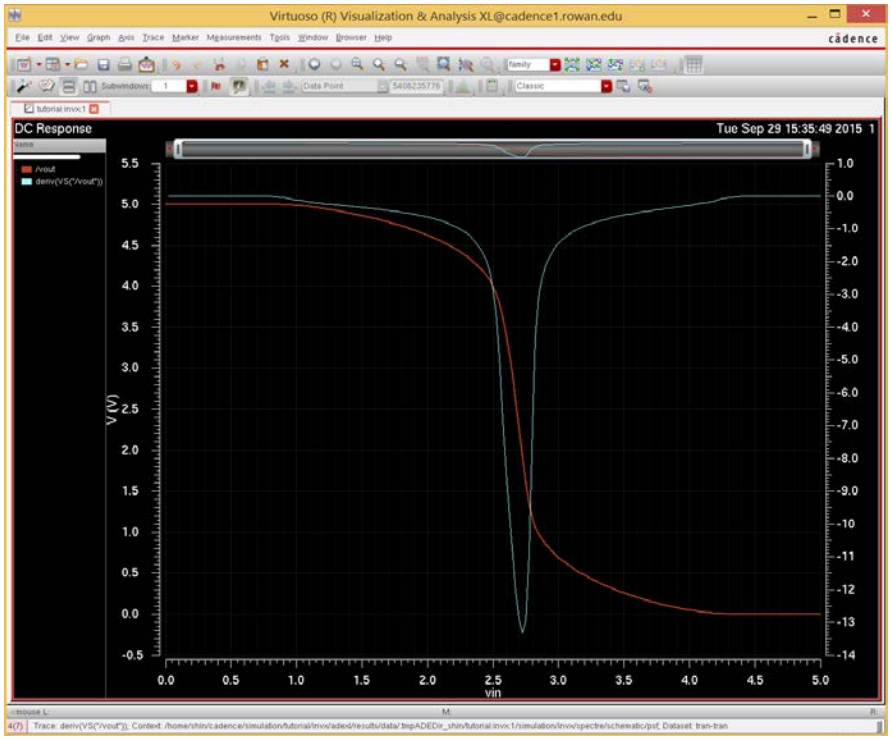
Then the noise margins can be calculated as: **$NM_L = V_{IL}-V_{OL}$, $NM_H = V_{OH}-V_{IH}$**



Computed results can be printed in the Calculator window or plotted in the waveform window. To plot the results of the expression displayed in the Calculator, from the Calculator menu, choose *Tools->Plot*. You can choose to append new

results to or replace the previous plots with new ones, by selecting its mode (Append, or Replace) from the Calculator.

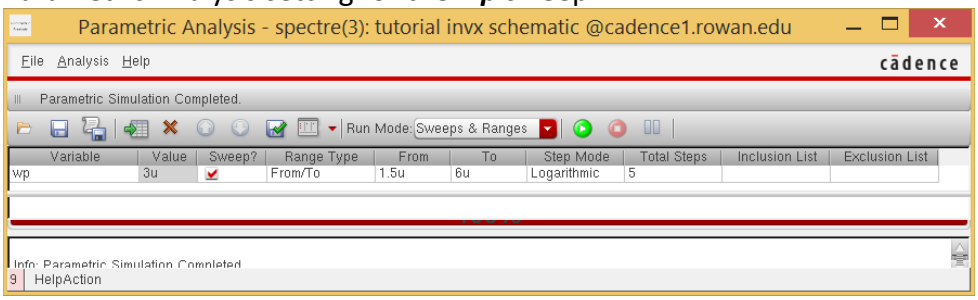
Below is a case where the results of *deriv(VT("/vout"))* is appended in blue to the *VT("/vout")* waveform in red.



Activity: Find the Noise Margins for logic levels LOW and HIGH.

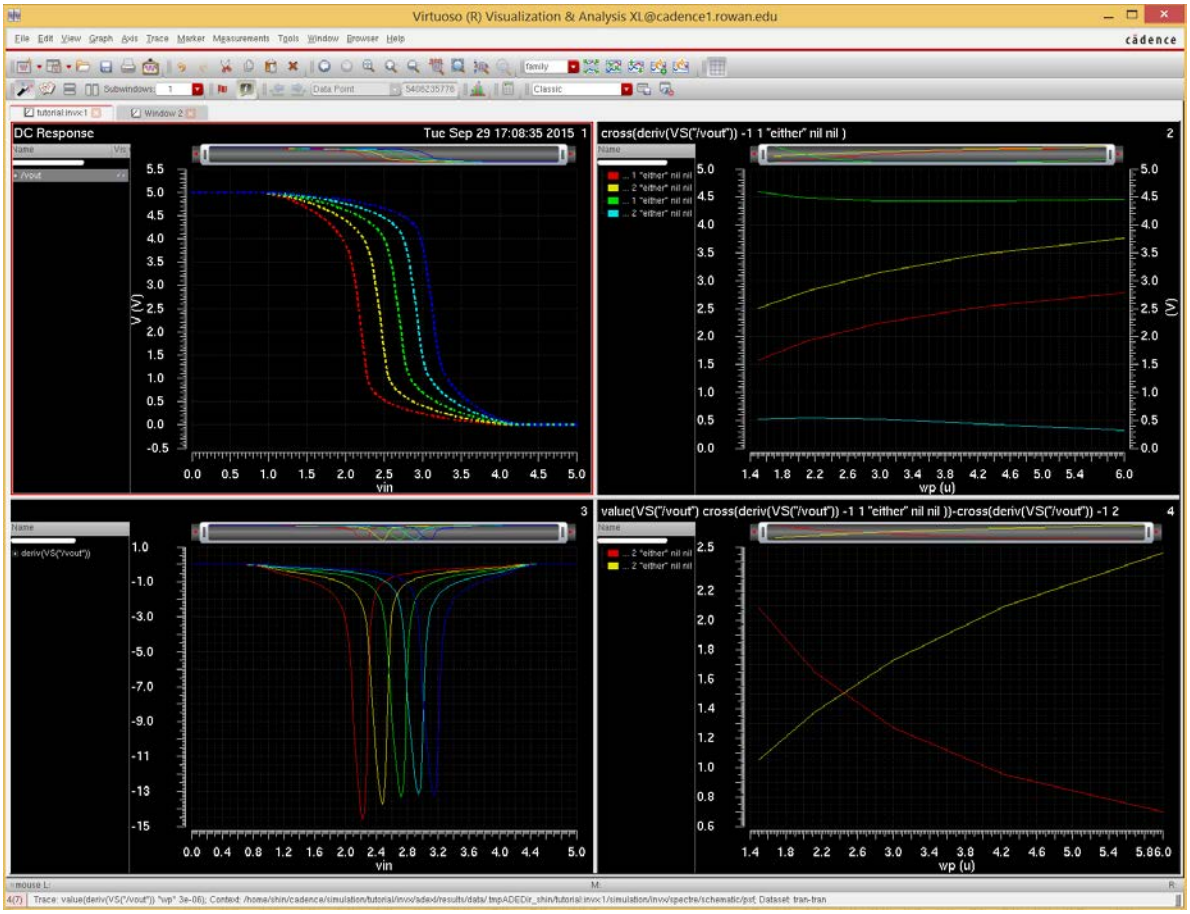
6) **Parametric Analysis**:

Now we want to see how the noise margin changes over a range of transistor size. To run the simulation with another parameter swept, we can use **Parametric Analysis** that can be chosen from **ADE Editor** by selecting *Tools-> Parametric Analysis*. Below is the Parametric Analysis setting for the ***wp*** sweep.



After creating netlist from the **ADE Editor** (*Simulation -> Netlist -> Create*), you can run the parametric simulation by clicking the ***green run button*** on the **Parametric Analysis**. Waveform window will be pop up when the analyses are completed.

With the expressions stacked in the Calculator, you can now draw the trend of noise margin over the range of transistor size. In the below waveform window, the first section displays the VTC over the sweep range, the 3rd panel shows the slopes of VTC (in other words, gain), the 2nd one shows the computed V_{IL}, V_{IH}, V_{OH}, and V_{OL}, and the 4th one shows the NM_H and NM_L.



If you successfully got the similar graphs showing the trends of noise margins, you're done for this lab. Please summarize what you've learned in this lab and put all relevant results into your report, and qualitatively compare the results you got against the theoretic expectations.