Digital Design: Cadence Tutorial

Weijie Jiang, Ce Ma, Wim Dehaene

January 16, 2025

Welcome to the Cadence tutorial! Let's create a simple inverter in Cadence Virtuoso. If this is familiar to you, you can just read the instructions in bold. If you do not know how to proceed, we suggest you to read the entire text.

1. **Open Virtuoso** as explained in the main document. You should see the Virtuoso window popping up in a few seconds. Click Tools/Library Manager to open up the library manager (Figure 1).

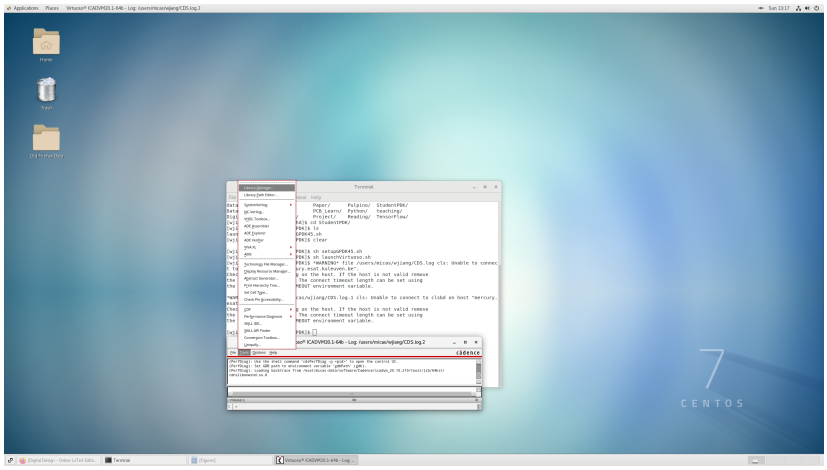

Figure 1: Cadence start page

2. In the library manager, **create a new library** by clicking on File/New/Library. Enter the name of your library. In this case **Digital_lib** is used. After that, check the attaching to existing technology library box and select the gpdk045 as the attached technology library.

3. Then click on the created library and **create a cellview** by clicking on File/New/Cell View. Enter "invertor" in the Cell box and make sure that the type is "schematic" (Figure 3). Click OK, and Virtuoso schematic editor will show. **Design an invertor** by first place a PMOS and NMOS pair. Click on the small transistor icon on the top (or press i) to create an instance. Change the library to gpdk045 and look for both pmos1v and nmos1v, place them in any place on the schematic editor (Figure 4. Only the nmos1v example is shown. Do the same for pmos1v.). Leave all parameters as default.

4. Then we **make connection of an invertor.** This is done by wire drawing (short-cut w). Connect the gate of pmos and nmos; source and bulk of pmos; source and bulk of nmos; and drain of pmos and nmos. Label them with good names and put pins that correspond to the net names (Figure 5). Useful shortcuts: "w" for drawing wires; "l" for labelling; "p" for pin placement. Note that for the output (in the example "ZN") needs to be specified

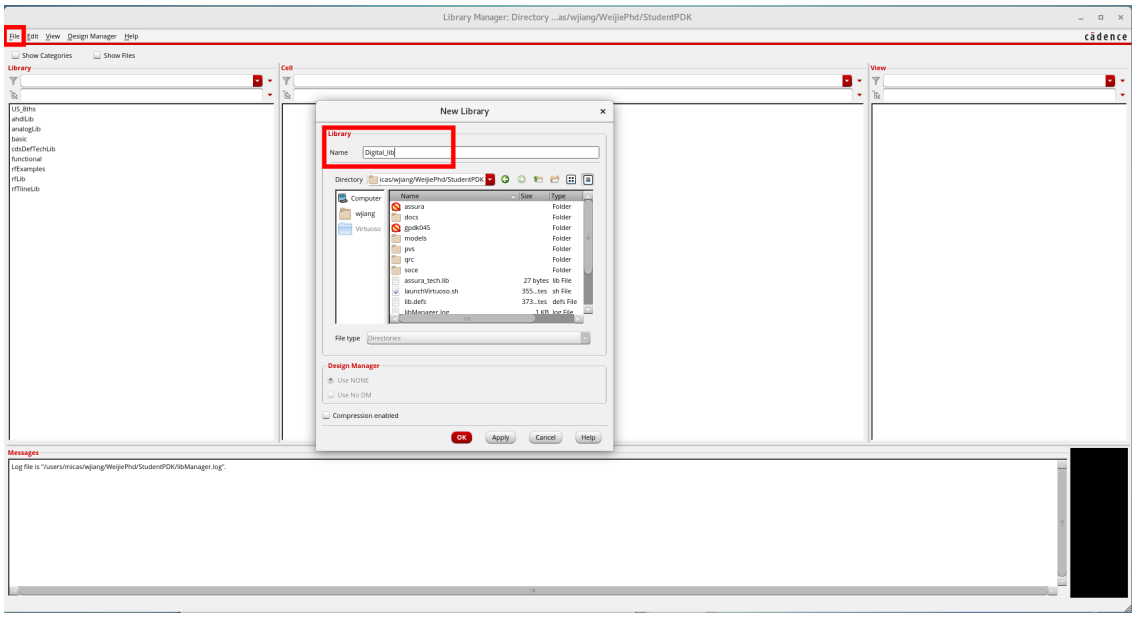

Figure 2: Create a new library

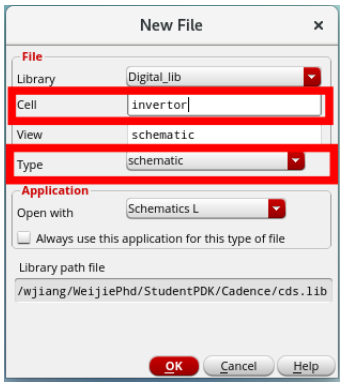

Figure 3: Create a new schematic

as output on the pin direction. When finished, click on the "check and save" button on the top left (the third one). If everything is fine, you should see no warnings or errors showing up.

5. Next, we will **simulate our circuit using ADE explorer**. The circuit will be simulated using transient simulation. To do that, we need to set the stimuli for all the pins. Click on the "launch/ADE explorer" to start the software (Figure 6). Then choose to "create a new session". Leave the name as default and click "session" if anything related to license shows up. The ADE explorer will open in place of the schematic viewer.

6. To perform the simulation, first **add an analysis**. Click on the "click to add analysis" and choose "tran" analysis. Enter some time for simulation (in the example 5 nano seconds). Useful units: "n" for nano, "f" for femto, "u" for micro. Change the accuracy to "conservative" and make sure the "enable" button on the bottom left is checked. If finished, leave everything else as default and click "ok". (Figure 7)

7. Next, we need to **specify a few signals to be examined.** Click on the "outputs/to be plotted/ select on design" (Figure 8). This will direct you back to the schematic viewer page. Click on the input and output pins of the design to add them into output lists.

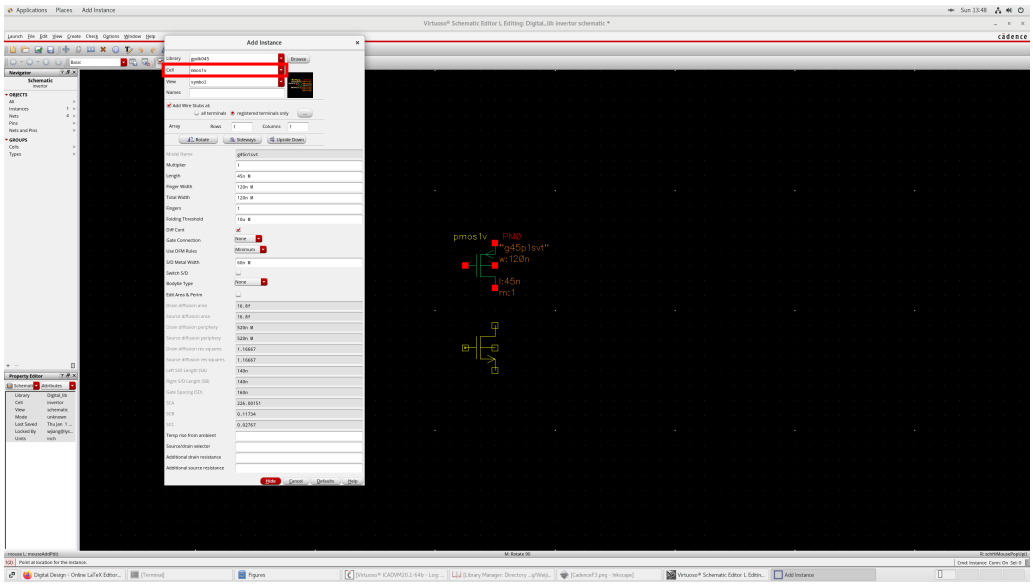

Figure 4: Place nmos and pmos

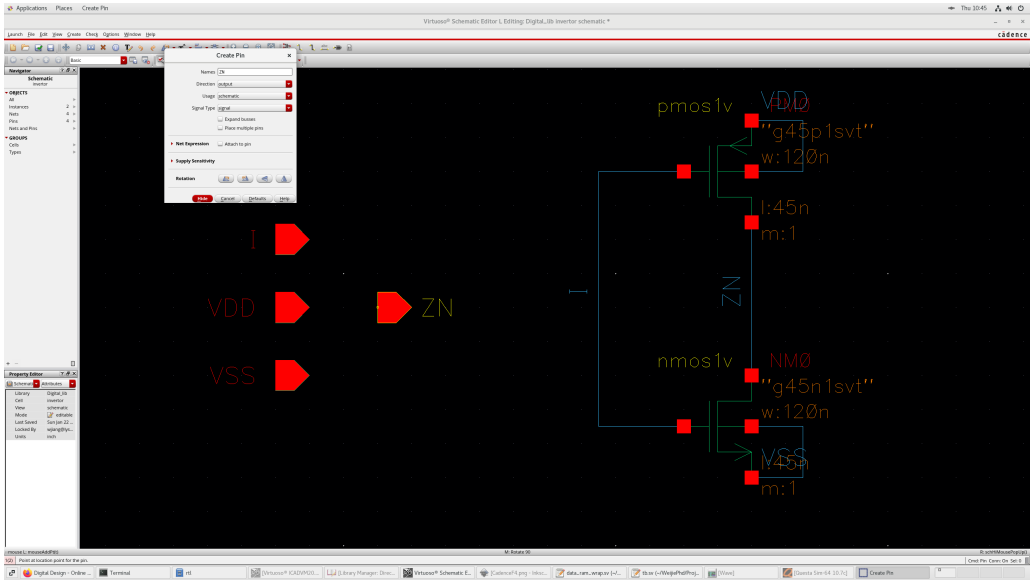

Figure 5: Completing wirings and pins

This allows us to see the current on those nodes. We also want to see the current of the invertor. To do that, click on the source pin of the PMOS (Figure 9). To check whether the type of each signals are entered correctly, go back to the ADE explorer by clicking on the "maestro" on the top of the schematic viewer. On the output page you should see two signal type and one signal (I) type being added to the list (Figure 10).

8. Then we need to **give stimuli to the circuit.** To do that, click on the "setup/stimuli" (Figure 11). A window for stimuli assignment will show up. First, we add constant source for the power pins. Specify the stimuli type to be dc. Enter some name for the VDD and VSS stimuli. Set 1 on dc voltage for VDD, and 0 on dc voltage for VSS. When finished, click on apply to add the stimuli in reservation. Note that for each different stimuli, a different name needs to be specified. The name can be change right next to the type selection drop down. Then specify the stimuli type to be pulse and create a square wave for the input. Use 0 for voltage 1 and and 1 for voltage 2. Enter the waveform as you

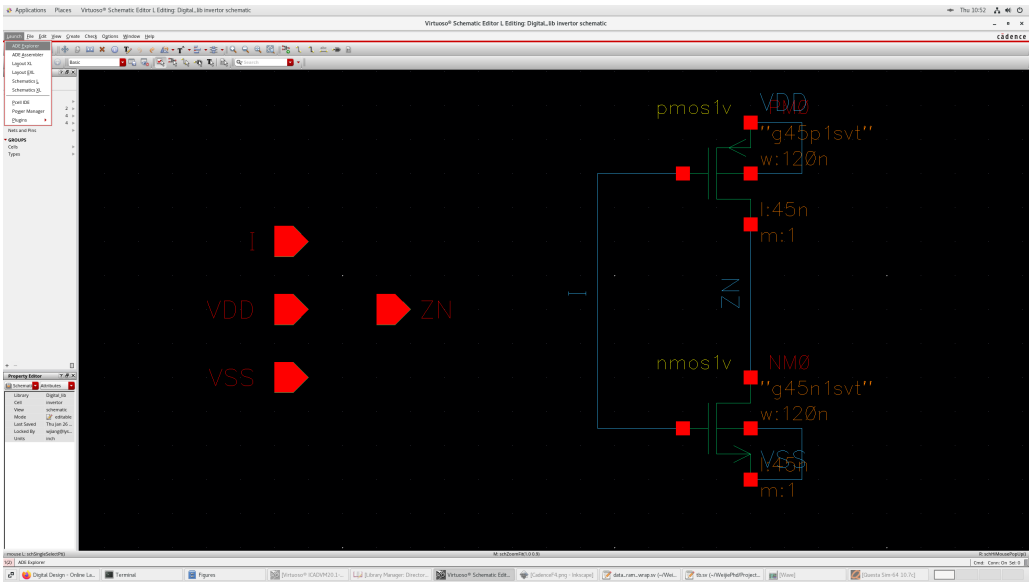

Figure 6: Launching ADE explorer

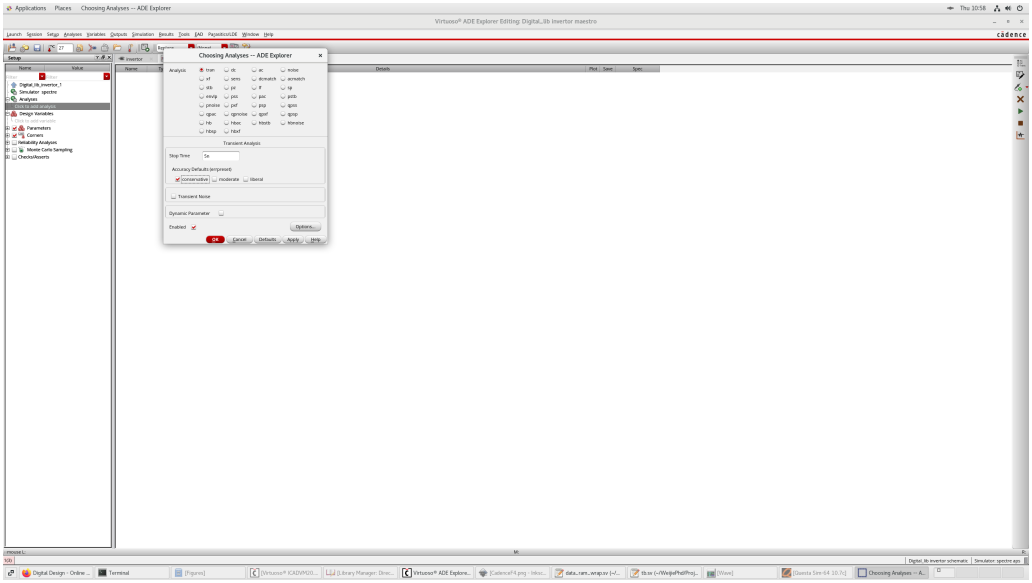

Figure 7: Adding analysis

like. Tips: period, total width of your periodic signal; delay time, delay time before the first pulse arrive; rise and fall edge, rising and fall time for the signal to change between voltage 1 and voltage 2; pulse width, total pulse length, which defines the duty cycle of your square wave. Type the name and apply. Then associate the stimuli with input pins. Change the authorizing to "off". Click on the stimuli and corresponding pin, then click on the associate button to associate them. The end results should look similar to the last figure. When done, click ok.

9. Next, **run the simulation** by clicking on "Simulation/netlist and run". The waveform window will show. Split the waveform by clicking on "split all strips" (Figure 13). Then we go measuring the rising delay of our invertor. Zoom in to the rising edge. Place the cursor at around half of VDD on the input curve and press "A" on the keyboard. This will create a marker on the input waveform. Then place the cursor at around half of VDD on the output curve and press "B" on the keyboard. Delay will display automatically as

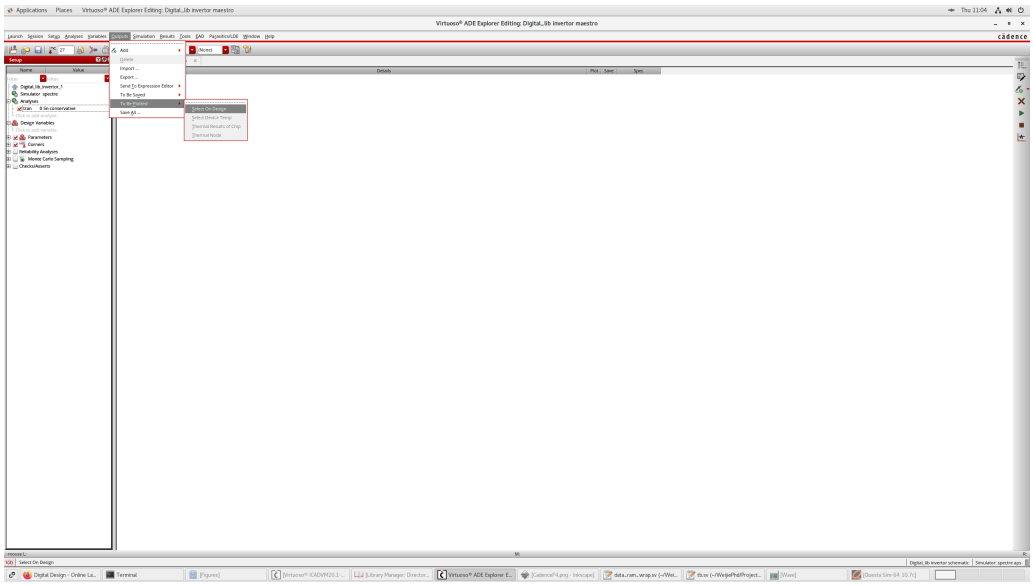

Figure 8: Adding outputs

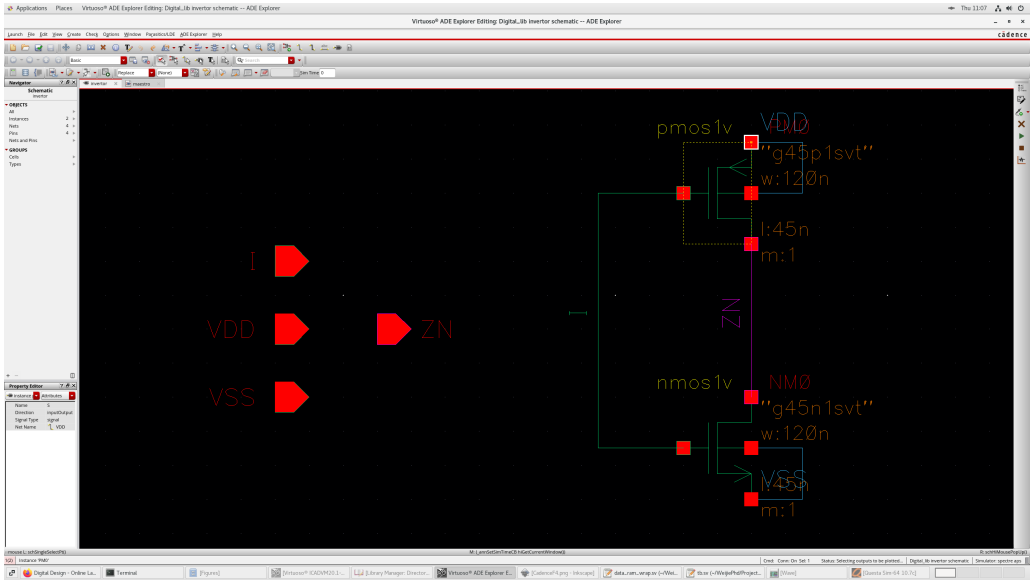

Figure 9: Selecting outputs

"delta x" (Figure 14). Do the same to measure the falling edge.

10. To get familiar with what happens in the background, **view the netlist** in the ADE-L window, by clicking Simulation/Netlist/Display. A new window opens, which shows the netlist. **Read it, so that you get familiar with the different parts**: the transistors (MP0 and MN0), the net names, the simulation statement(tran),... When running a simulation in cadence virtuoso, first a netlist is generated from the schematic, then the spectre simulator (which is similar to Hspice) is launched separately and finally the results are collected.

Working with Virtuoso and ADE is straightforward. However, it involves a lot of clicking and GUI operations, which is inefficient for large scale digital circuits. Meanwhile, many analog features are not used by digital designers. To do more efficient design, we will move to more scripted version of this exercise.

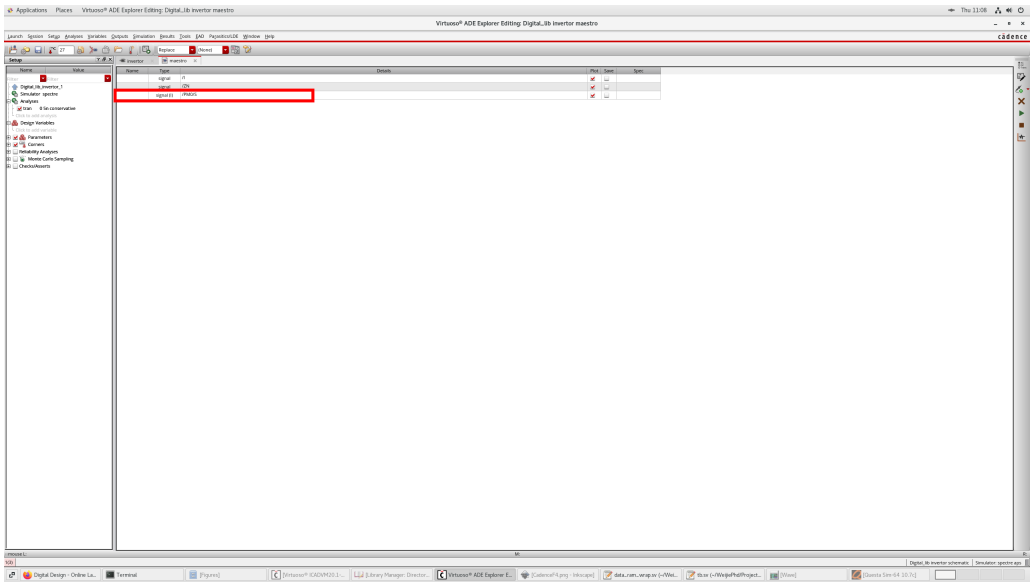

Figure 10: Check outputs

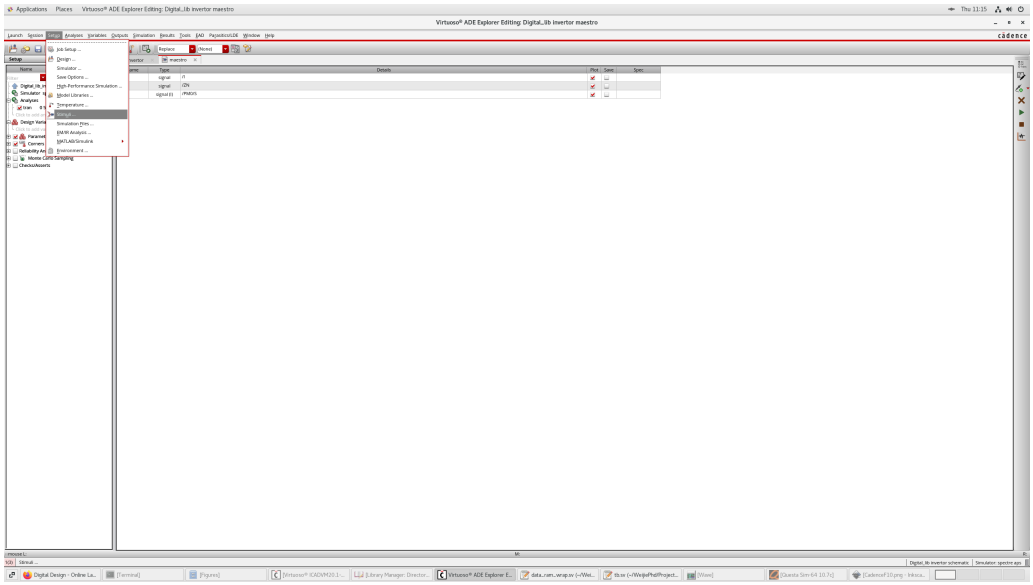

Figure 11: Adding stimuli

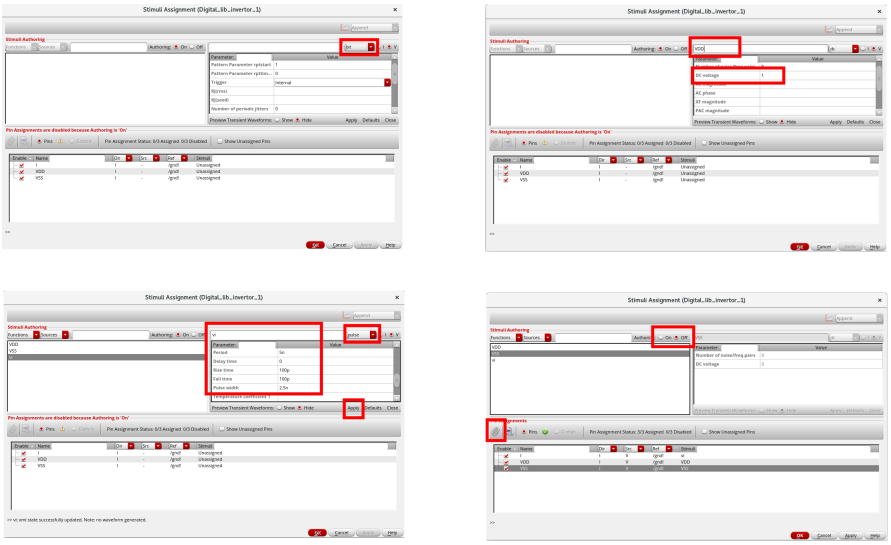

Figure 12: Adding stimuli (more details)

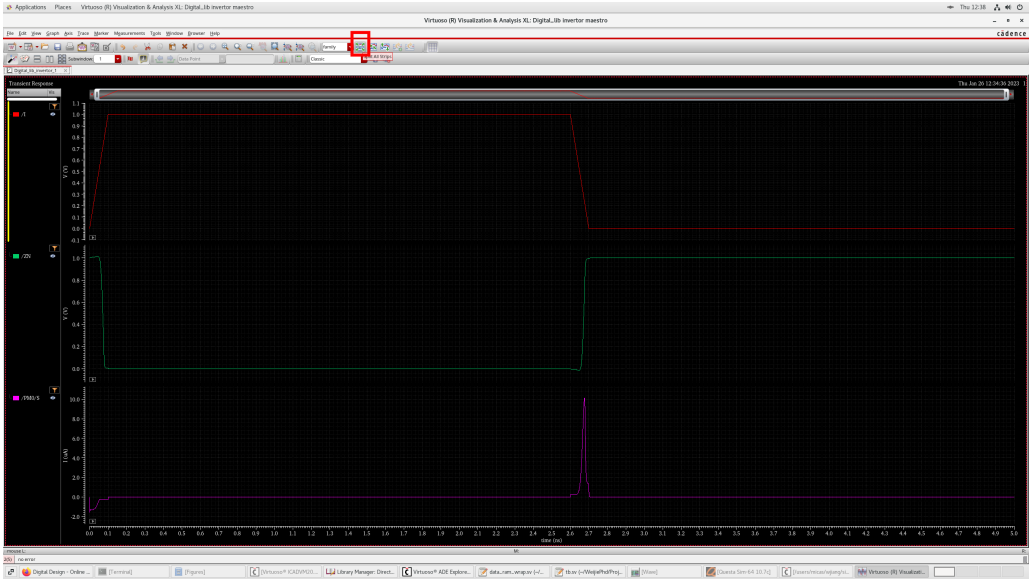

Figure 13: Split all strips

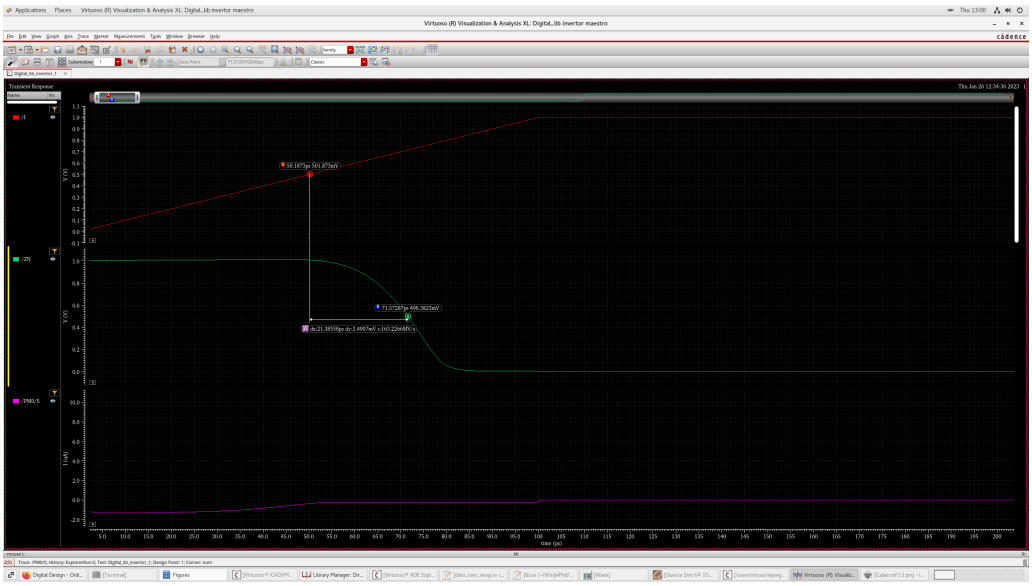

Figure 14: Measuring delay