

LABORATORY 0

INTRODUCTION TO VLSI DESIGN FLOW

OBJECTIVES

No.	Objective	Requirements
1	Setting Linux environment	<ul style="list-style-type: none"> ▪ Use VMware Workstation to open a virtual machine and successfully run Cadence Virtuoso; ▪ Learn basic commands.
2	Create a schematic in Cadence Virtuoso	<ul style="list-style-type: none"> ▪ Create a new library for students; ▪ Design an ideal inverter using Verilog-A and create its symbol; ▪ Create an inverter schematic using the Virtuoso Schematic Editor; ▪ Draw a symbol for the students' design using the Virtuoso Symbol Editor.
3	Pre-layout simulation	<ul style="list-style-type: none"> ▪ DC analysis using ADE; ▪ Transient simulation using ADE-L.
4	Measurements	<ul style="list-style-type: none"> ▪ Using Calculator or Point Marker to measure timing parameters (rise/fall time, rising/falling propagation delay); ▪ Power consumption.
5	Layout	<ul style="list-style-type: none"> ▪ Draw layout using Virtuoso Layout Editor; ▪ Check DRC, LVS, and PEX.
6	Post-layout simulation (optional)	<ul style="list-style-type: none"> ▪ Re-measure timing parameters and power consumption; ▪ Compare the measurement results from pre-layout and post-layout simulations.
7	Appendices	<ul style="list-style-type: none"> ▪ Design Rules of FreePDK45; ▪ Physical Layers in FreePDK45; ▪ Single VIA, VIA array, stacked VIA.; ▪ Bindkeys in Cadence Virtuoso; ▪ Enable Shared folder for a virtual machine.



PART 1**Objectives:**

- Run the virtual machine and open Cadence Virtuoso successfully.

Instructions:

Students download the virtual machine (via LMS), and VMware Workstation, then follow the instructions:

First, please install VMware on students' PCs/Macs...

Second, extract the virtual machine folder just downloaded. Students' laptops need ~88GB of space to extract (22.9GB when compressed, and 64GB when extracted).

Third, please configure the memory and processor settings to match your computer's specifications.

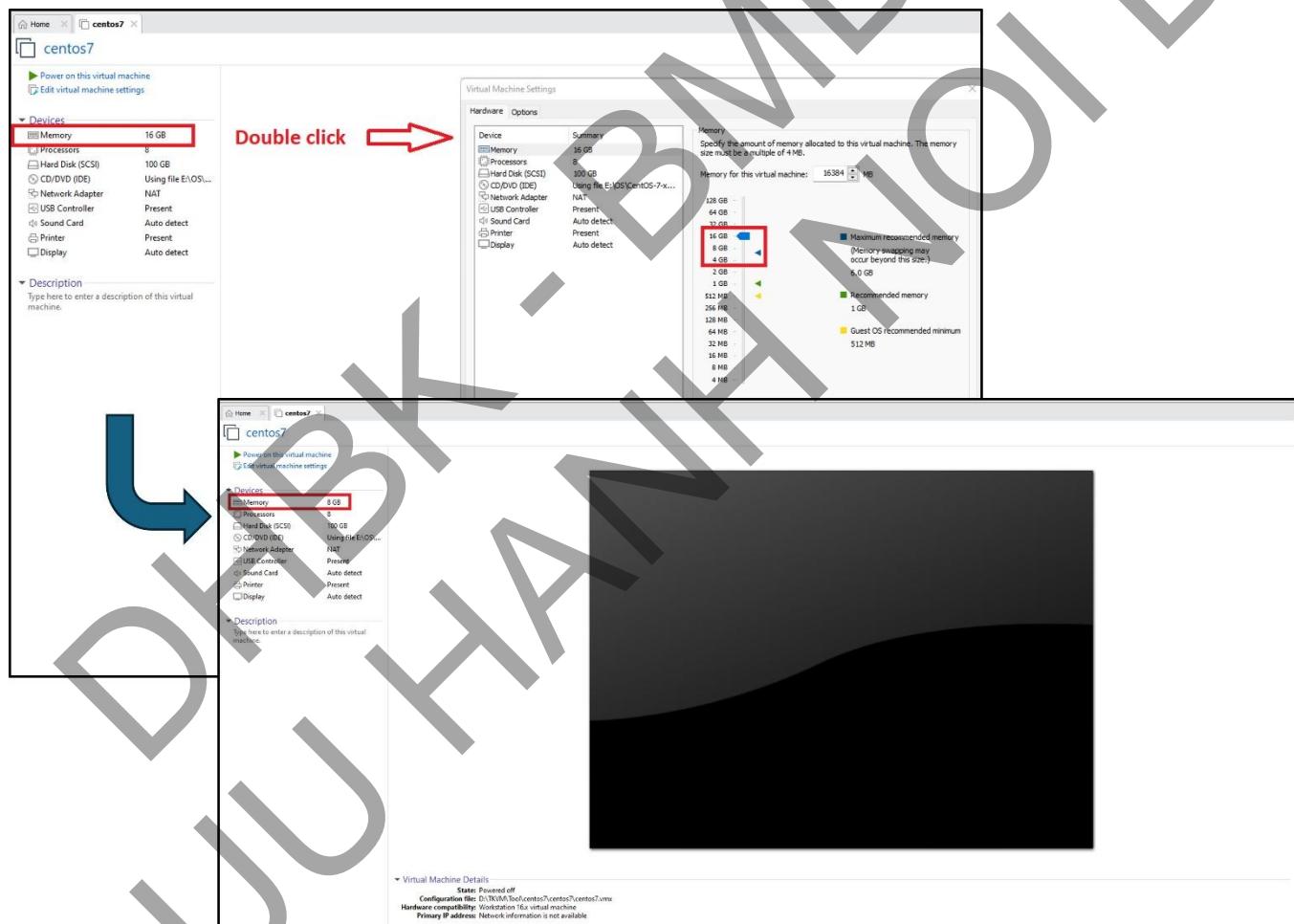


Figure 1 Configure the memory of the virtual machine.

Moreover, while using the virtual machine, students must establish a shared folder link between their PC and the virtual machine. This precaution helps prevent the loss of unsaved work if the virtual machine encounters issues. Virtual machines save data only when properly powered off; unexpected crashes or accidental shutdowns of VMWare can result in complete data loss. Students can refer to **Appendix E**.

Fourth, open the virtual machine as follows. Please enter the password in **Figure 5** to sign in.



Figure 2 Power on the virtual machine.

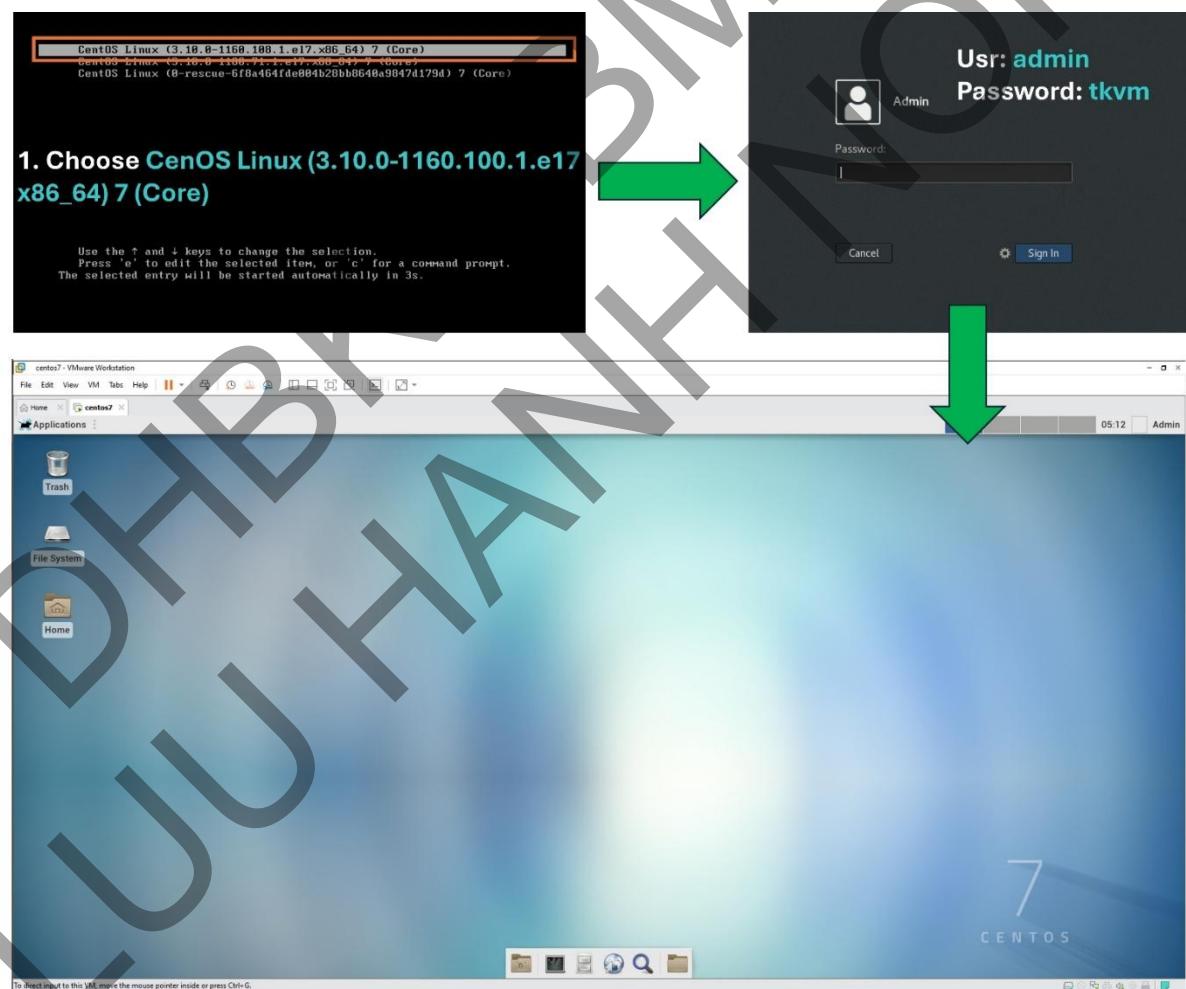


Figure 3 Sign in to the virtual machine.

Fourth, open the terminal and run Cadence Virtuoso.

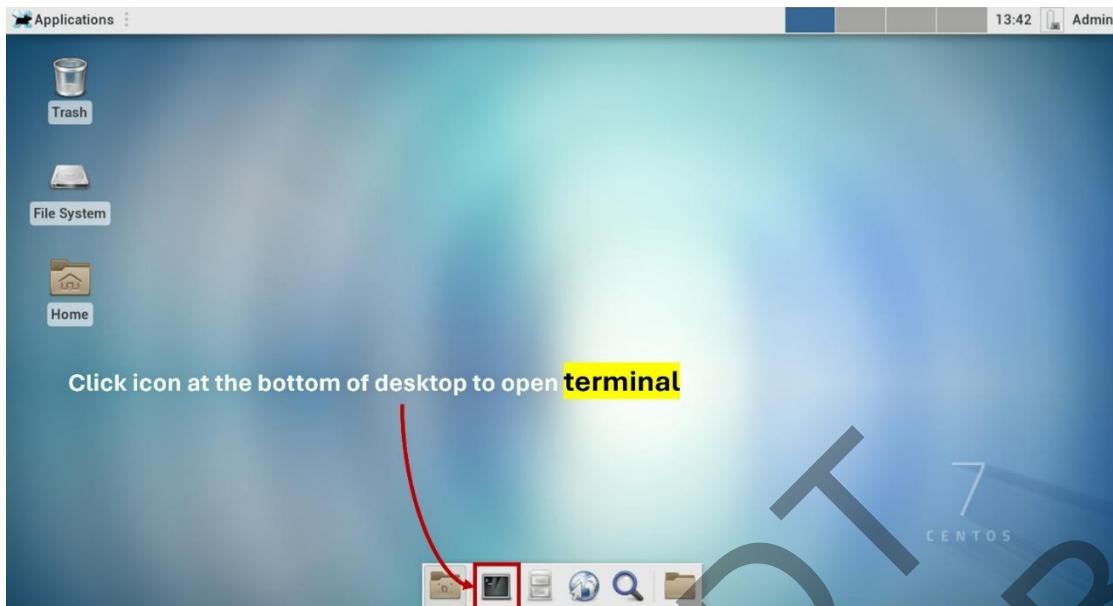


Figure 4 Open Terminal.

```
total 4.5K
drwxr-xr-x. 2 admin admin 22 Apr 19 09:35 Desktop
drwxr-xr-x. 6 admin admin 73 Jun 23 13:55 Documents
drwxr-xr-x. 2 admin admin 6 Mar 2 05:11 Downloads
drwxr-xr-x. 2 admin admin 6 Mar 2 02:57 Music
drwxr-xr-x. 2 admin admin 6 Mar 2 02:57 Pictures
drwxr-xr-x. 3 admin admin 17 Apr 16 06:17 Public
drwxr-xr-x. 2 admin admin 6 Mar 2 02:57 Templates
drwxr-xr-x. 2 admin admin 6 Mar 2 02:57 Videos
dr-xr-xr-x. 1 admin admin 4.1K Jun 24 02:47 shared
[admin@centos7 ~]$ cd Documents/virtuoso/
```

```
total 244K
-rw-rw-r--. 1 admin admin 537 Jun 24 02:52 cds.lib
-rwxrwxr-x. 1 admin admin 240K Mar 13 11:23 display.drf
drwxrwxr-x. 5 admin admin 151 Jun 24 02:38 sample
[admin@centos7 virtuoso]$ virtuoso &
```

1. Go to directory **Documents/virtuoso:**
cd Documents/virtuoso

2. Open Cadence Virtuoso:
virtuoso &

The next time when students open Cadence Virtuoso, please use two commands as follows:

cd ~/Documents/virtuoso

virtuoso &

Figure 5 Open Cadence Virtuoso.

Using the virtual machine requires Linux/UNIX scripting skills. Students must be familiar with command-line operations for creating and removing folders, accessing and modifying design files, and executing software through commands or scripts.

Command	Description
pwd	Print the working directory.
ls [path]	List file and folder.
ll	List directory contents with permission information.
cd <path>	Go to <path>.
cd ..	Traverse up one directory.
cd ../../	Traverse up two directories.
cd ~ or cd	Go to my home directory.
du -sh	Show directory/file size.
du -sh *	Show all of file/directory size.
touch <file>	Create a new temp file.
cp <file/path> <path>	To copy.
mv <file> <file>	To rename.
mv <file> <path>	To move file(s).
mkdir	Create a directory.
rmdir	Delete an empty directory.
rm <file>	Delete file (recommend: rm -i <file>).
rm -rf	Force delete.
man <command>	To read manual of command.
tree -l <level> -c	List contents of directories in a tree with level and color.
vi <file_name>	To edit or create ordinary file.

Table 1 Linux/UNIX basic commands.

Before proceeding to Part 2, students will be introduced to the Circuit Design Flow and the tools used at each stage. Reviewing this section will provide an overview of the lab. In this lab, all circuit design tasks will be performed exclusively through the Graphical User Interface (GUI). Unlike in industry, where scripts are commonly used for running simulations and collecting data, students will not be using scripts in this lab.



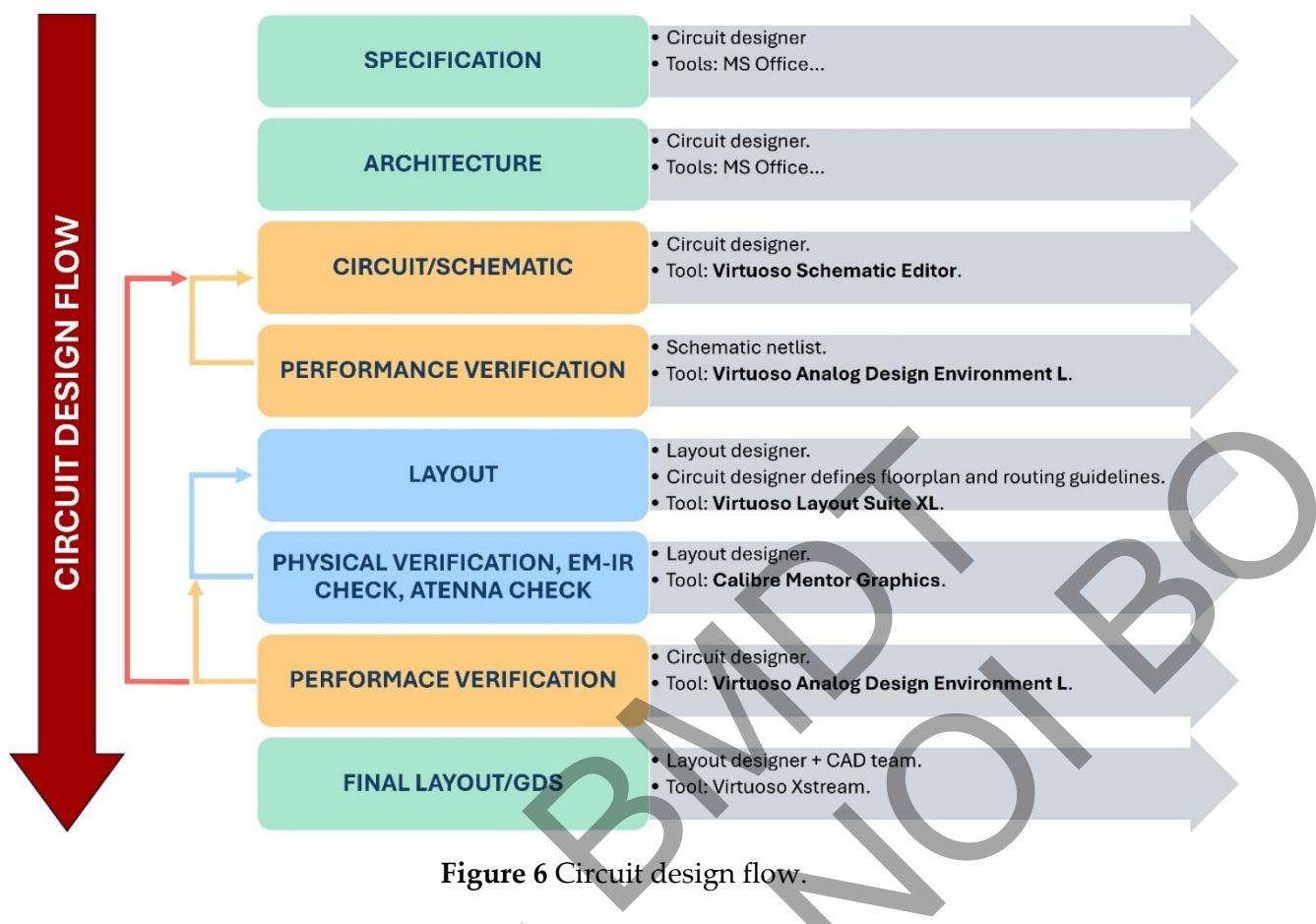


Figure 6 Circuit design flow.

PART 2**Objective:**

- Create an ideal inverter using Verilog-A and a CMOS-based inverter.

Requirements:

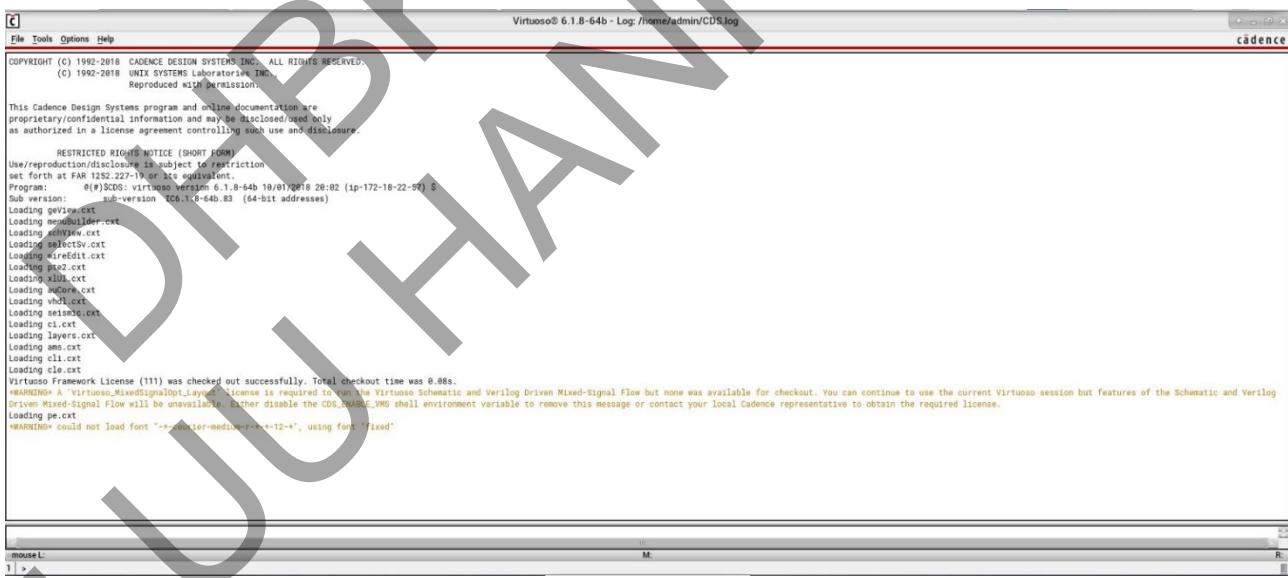
An example of designing a **logic inverter gate** is presented to illustrate the use of **Cadence Virtuoso** and the circuit design process. Each step, along with commonly used tools in digital and analog circuit design—such as simulation tools, circuit drawing, and modeling languages—will be explained sequentially. The instructions are divided into four steps:

1. Create a new library.
2. Design an ideal model using Verilog-A.
3. Develop the inverter schematic.
4. Generate the inverter symbol.

Instruction:

- 1) *Add new Create a new library:*

Command Interpreter Window (CIW) is displayed after opening Cadence Virtuoso.



The image shows a screenshot of the Cadence Virtuoso Command Interpreter Window (CIW). The window title is "Virtuoso® 6.1.8-64b - Log: /home/admin/CDS.log". The menu bar includes File, Tools, Options, Help, and a Cadence logo icon. The main area displays a log of commands and their execution. The log starts with copyright information for Cadence Design Systems, Inc., followed by a restricted rights notice. It then lists several files being loaded, such as "Loading genviwex", "Loading neopubBuilder", "Loading xmlView", "Loading selectSv", "Loading wireedit", "Loading netlist", "Loading xlls", "Loading acore", "Loading vhdl", "Loading seimic", "Loading layers", "Loading sens", "Loading cll", and "Loading cle". The log concludes with a warning about a missing font file and a note about the Virtuoso Framework License.

```

File Tools Options Help
Virtuoso® 6.1.8-64b - Log: /home/admin/CDS.log
cadence

COPRIGHT (C) 1992-2010 CADENCE DESIGN SYSTEMS INC. ALL RIGHTS RESERVED.
(C) 1992-2010 UNIT SYSTEMS Laboratories INC.
Reproduced With Permission.

This Cadence Design System program and online documentation are
proprietary/confidential information and may be disclosed/used only
as authorized in a license agreement controlling such use and disclosure.

RESTRICTED RIGHTS NOTICE (SHORT FORM)
Use/reproduction/disclosure is subject to restriction
set forth at FAR 1252.227-19 or its equivalent.
Program: #(#)CDS: virtuoso version 6.1.8-64b 10/01/2018 20:02 (ip-172-18-22-97)
Sub Version: sub-version IC6.1.8-64b.83 (64-bit addresses)

Loading genviwex
Loading neopubBuilder
Loading xmlView
Loading selectSv
Loading wireedit
Loading netlist
Loading xlls
Loading acore
Loading vhdl
Loading seimic
Loading layers
Loading sens
Loading cll
Loading cle

Virtuoso Framework License (1111) was checked out successfully. Total checkout time was 0.08s.
Information: Virtuoso Mixed-Signal Layout License is required to use Virtuoso Schematic and Verilog Driven Mixed-Signal Flow but none was available for checkout. You can continue to use the current Virtuoso session but features of the Schematic and Verilog Driven Mixed-Signal Flow will be unavailable. Either disable the CDS_VNML_VMS shell environment variable to remove this message or contact your local Cadence representative to obtain the required license.
Loading pe.cxt
#WARNING: could not load font "++courier-medium-r--12--", using font "fixed"

mouse L: M: R:
1 >

```

Figure 7 Command Interpreter Window (CIW) of Cadence Virtuoso

The Command Interpreter Window (CIW) functions similarly to a *.log* file, displaying all activities throughout the circuit design process in Cadence EDA. Designers can use the CIW to access any



integrated tool within the Cadence software suite by entering SKILL commands. Access Library Manager to create a new library by choosing **Tools > Library Manager**

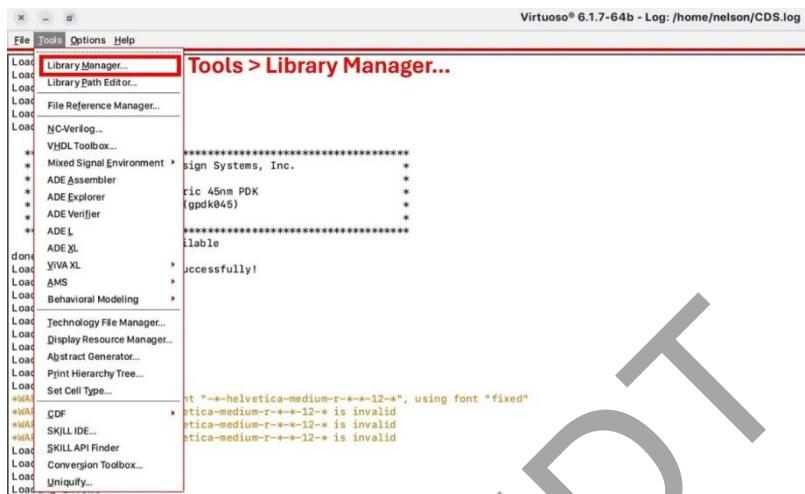


Figure 8 Access Library Manager by choosing Tools > Library Manager.

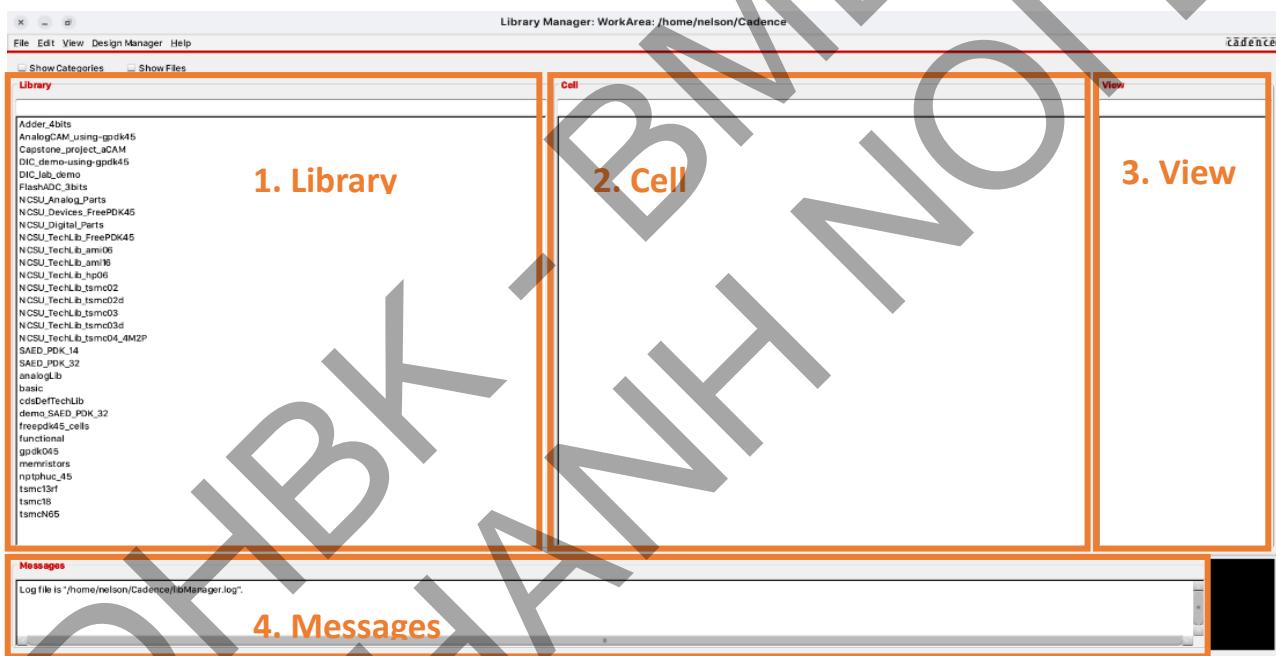


Figure 9 Library Manager window.

No	Name	Description
1	Library	The design libraries, including the reference Process Design Kit (PDK), are all displayed in this area.
2	Cell	The design blocks are displayed in this area to inform the designer about their location.
3	View	The design blocks are typically directories that contain design data. These data files are displayed in the View, such as the Schematic and Symbol.
4	Message	The *.log file monitors the activities of the Library Manager to inform the designer about the locations of files for error checking in the design.

Table 2 Dialogues shown in Library Manager

Follow these instructions to create a new library in Library Manager: **File > New > Library**. The “New Library” window will appear.

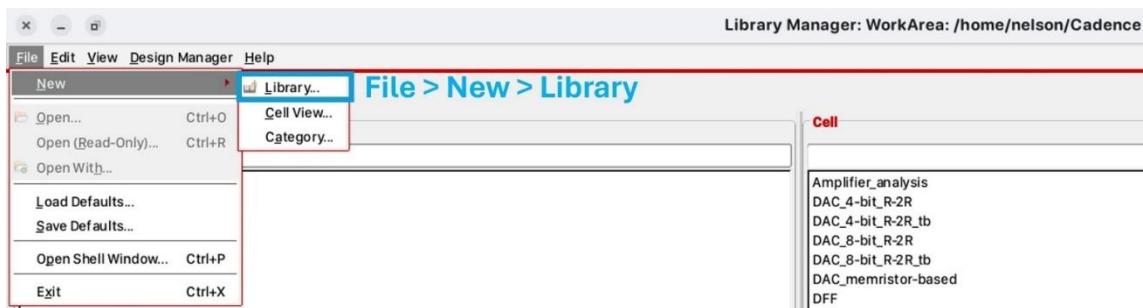


Figure 10 Open the New Library window by choosing **File > New > Library**.



Figure 11 New Library window and step to perform.

Enter a name for the new library and click **OK**. To design within a library, students must reference a **Process Design Kit (PDK)** that includes semiconductor components. In this case, the PDK used is **FreePDK45**.

Select the following options: “Attach to an existing technology library” > **NCSU_Techlib_FreePDK45** > Press “OK”.

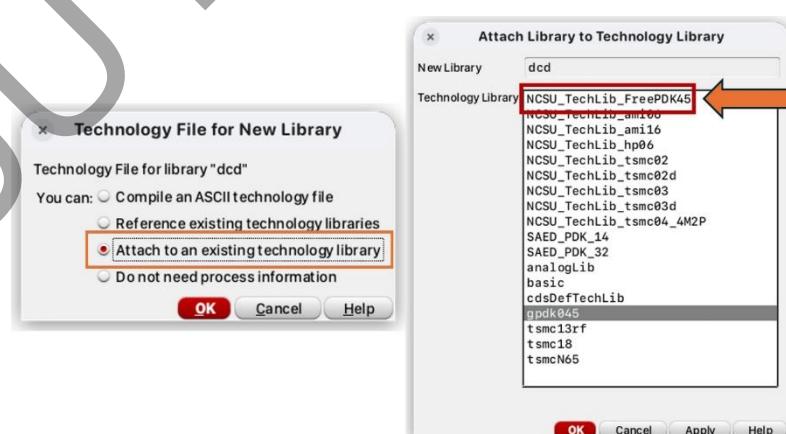


Figure 12 Add Technology Library to students New Library.



2) Create an ideal model using Verilog-A

Next, to create a new schematic circuit design, in the Library Manager window, select the newly created library, and choose **File > New > Cell View...**



Figure 13 Open the New File window by choosing File > New > Library.

A window for creating a new VerilogA view will appear as follows:

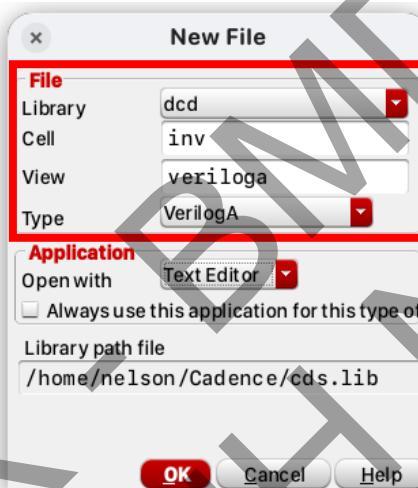


Figure 14 Create a new file at the library just created.

Enter the cell name, then select the design view and type. The selection menu provides various options. First, create an **ideal model** or **macro** for the **INVERTER gate circuit** using **Verilog-A**¹.

¹ Verilog-A is a high-level hardware description language (HDL) used for modeling analog and mixed-signal systems. As an extension of Verilog, it provides a continuous-time modeling approach, making it well-suited for describing circuit behavior at a higher level of abstraction. Unlike traditional Verilog, which primarily focuses on digital logic design, Verilog-A allows designers to incorporate electrical properties such as voltage, current, and time-dependent behaviors. This capability makes it a powerful tool for simulating analog components like amplifiers, filters, and custom semiconductor devices. Verilog-A is widely used in circuit simulation, particularly within **Cadence Virtuoso** and other electronic design automation (EDA) tools, to create behavioral models for verification before physical implementation.

A text editor window will appear:

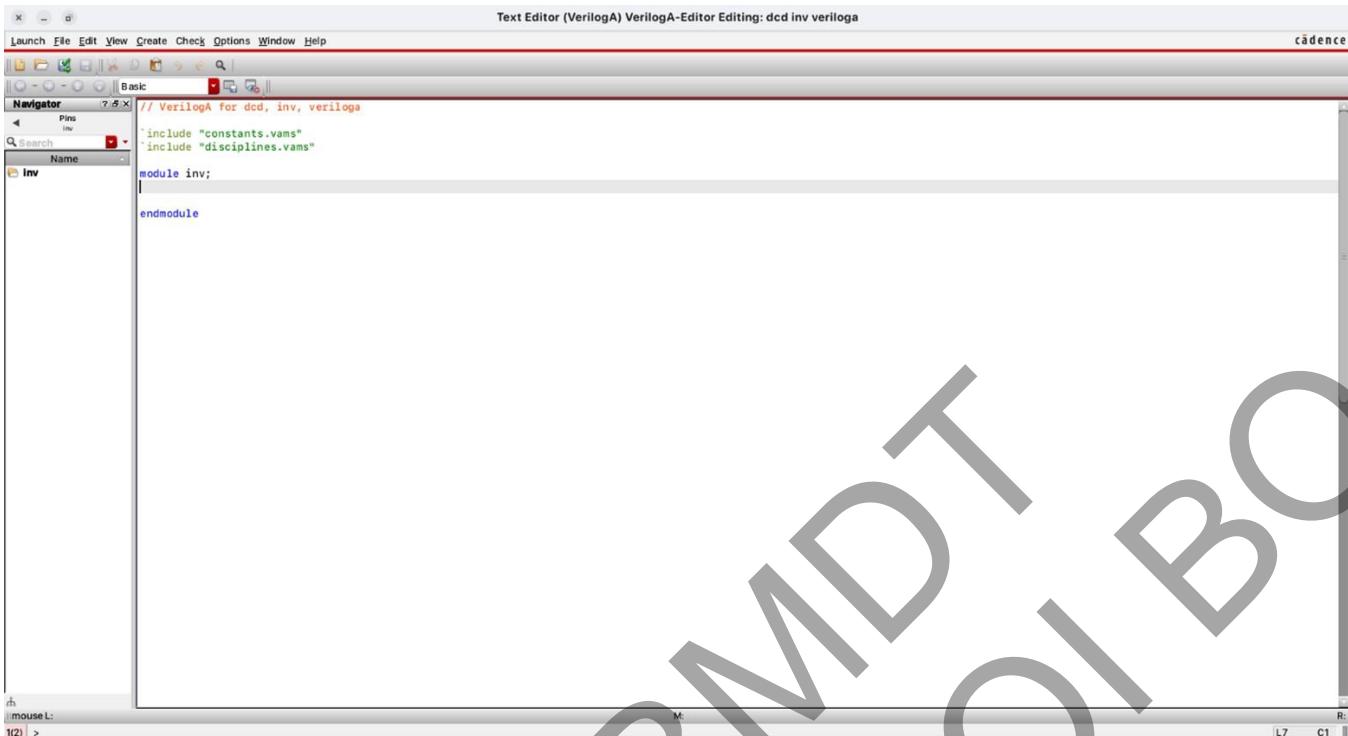


Figure 15 Text Editor VerilogA-Editor window.

Type the Verilog-A code for the Inverter_Idea gate as follows:

```
// VerilogA for nptphuc, inv_ideal, schematic

`include "constants.vams"
`include "disciplines.vams"

module inv_ideal(in, out );
    input          in;
    output         out;
    electrical    in, out;
    integer        d_out, vdd;

    parameter real tdelay = 0;
    parameter real trise = 1p;
    parameter real tfall = 1p;

    analog begin
        vdd=1;

        @(cross(V(in)-0.5*vdd));
        if (V(in)>0.5*vdd) d_out=0;
        else d_out=1;

        V(out) <+ transition(d_out, tdelay, trise, tfall)*vdd;
    end
endmodule
```

Figure 16 The Verilog-A code for the inverter idea gate

Like Verilog, **Verilog-A** includes similar declaration components used in Verilog-based design. However, in **Verilog-A**, students must declare not only voltage levels but also time parameters, including

rise time, fall time, and propagation delay of the logic gate.

After creating an **ideal inverter model** using **Verilog-A**, the next step is **compilation**. Save and close the editor, then select "Build a database of instances, nets, and pins found in files." A dialogue box will then appear.

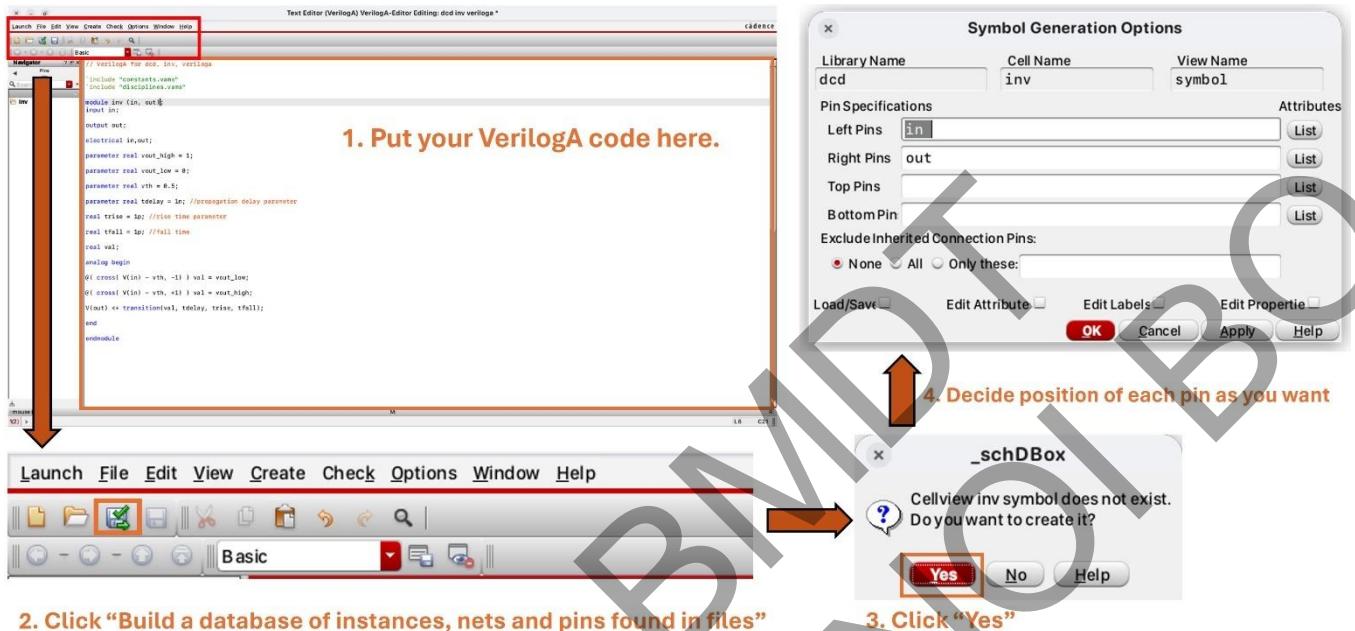
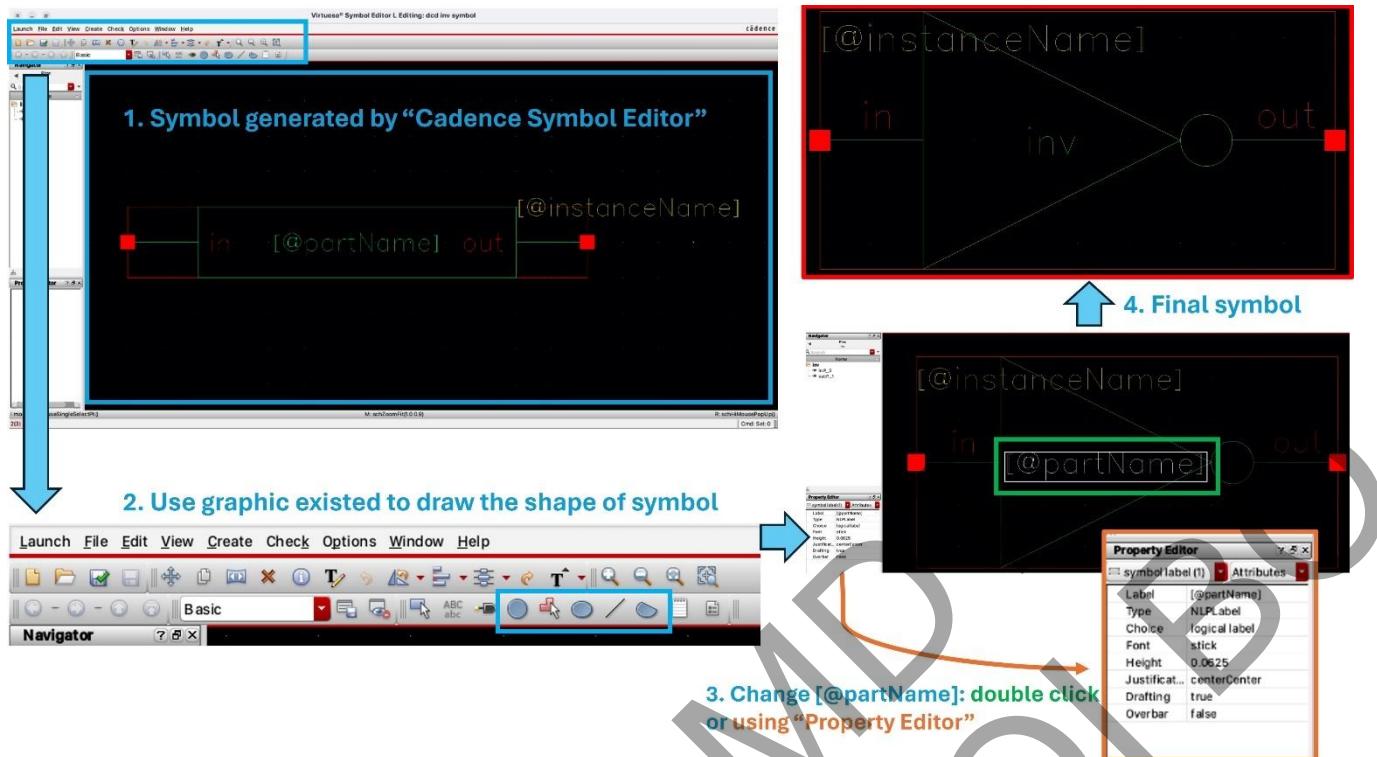


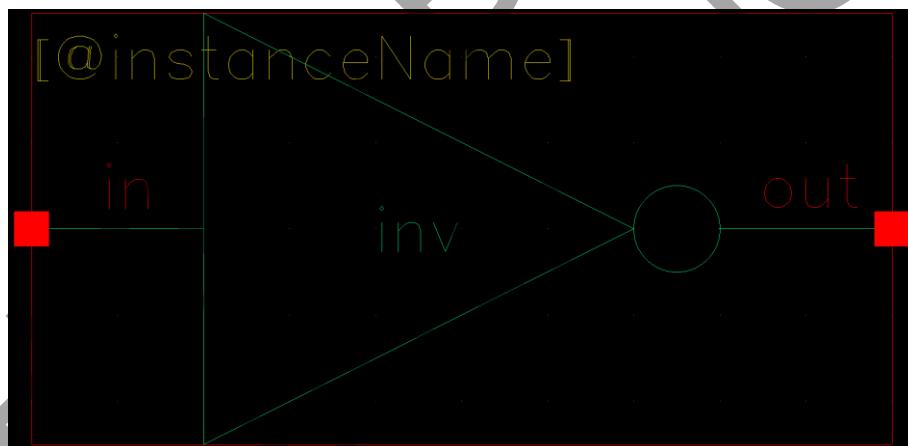
Figure 17 Step to create a symbol of `inv_ideal`.

Selecting **OK** takes students to the workspace where they can draw symbols for the logic gate. The Virtuoso Symbol Editor (VSE) provides an intuitive interface for creating symbols that represent the functionality of the inverter gate and other circuits. The inverter's symbol is created using graphical tools, as shown in **Figure 18**.

Students can use the existing graphical elements in **Virtuoso Symbol Editor (VSE)** to design symbols for their components. For example, the standard symbol for an inverter is shown below. If students experience issues when renaming `[@partName]`, such as lag or difficulty selecting it, they should use the **Properties Editor** instead. Additionally, students can customize the font and height as needed.

**Figure 18** Step to draw inv_idea's symbol.

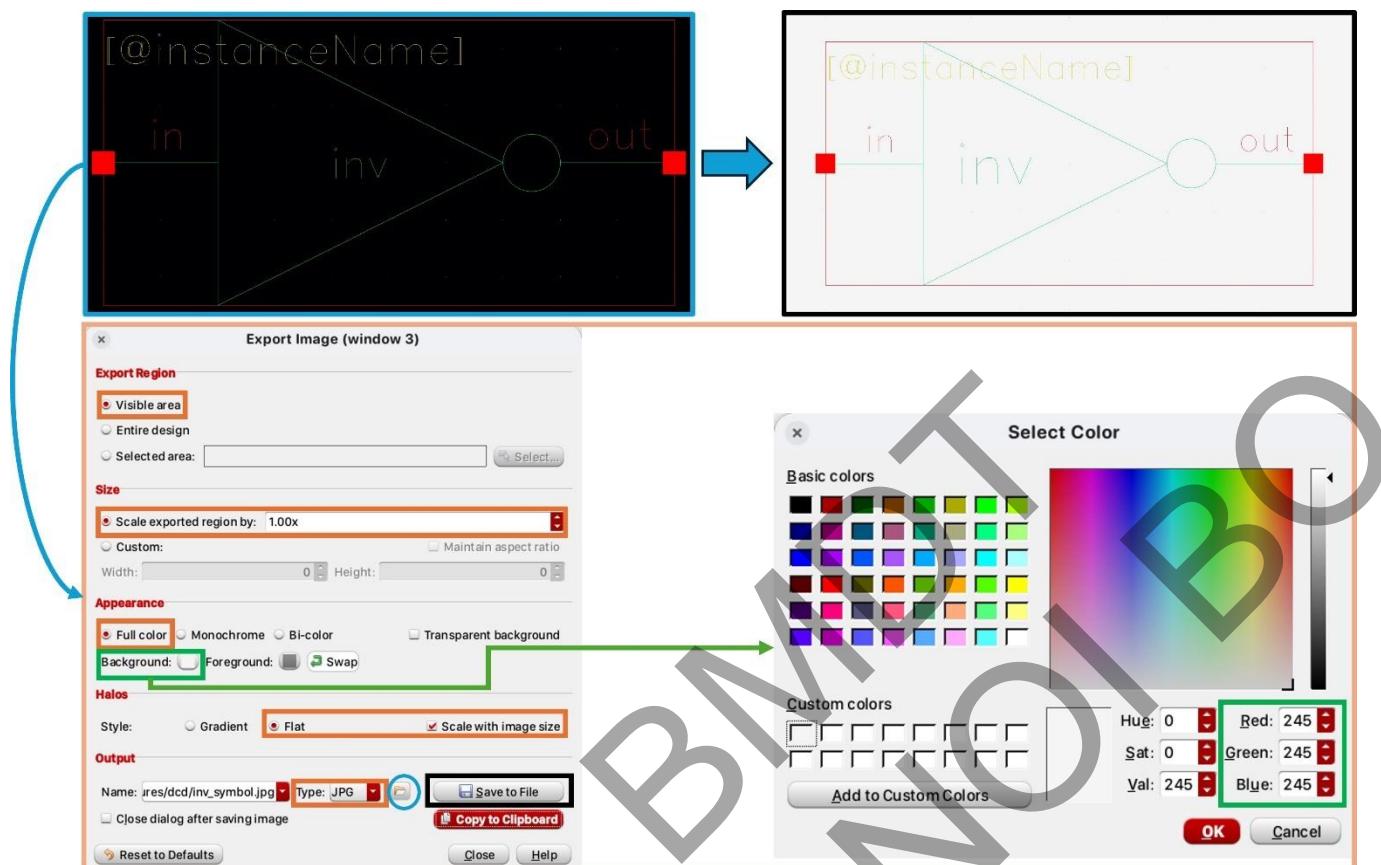
Please choose **Check** and **Save** after drawing completely. The complete symbol looks like this:

**Figure 19** inv_dea's symbol.

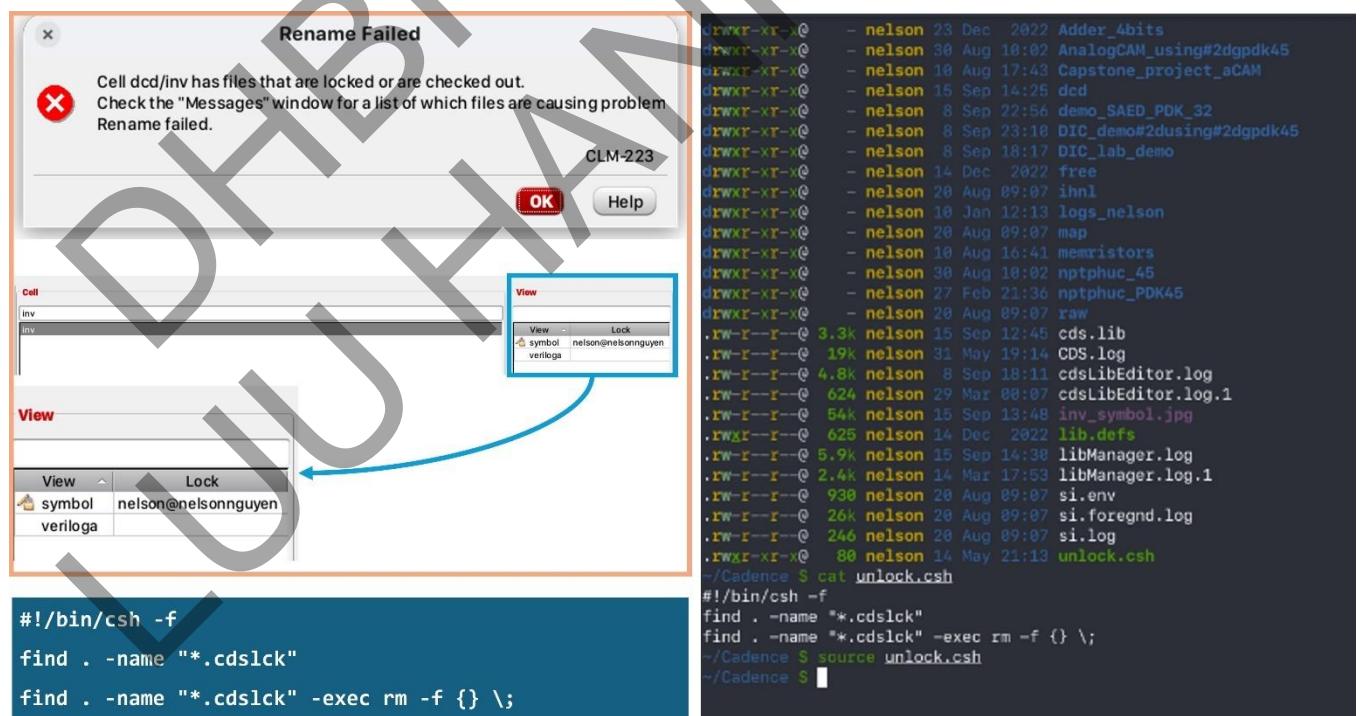
The newly drawn symbol may be difficult to see against the **black background**, especially when inserted into reports or presentations. To improve visibility, students can change the background color by exporting the image in **Virtuoso Symbol Editor**:

1. Navigate to **File > Export Image** in the **Virtuoso Symbol Editor** window.
2. Set the background color (e.g., $R = G = B = 245$ for a light gray background).

after completing these steps, students can capture the image and insert it into their reports. Additionally, students also can capture directly from **Virtuoso Symbol Editor** then students set the transparent color by using Microsoft Office.

**Figure 20** Change the background color when exporting an image.

Additionally, if students need to rename a newly created cell but encounter a "Rename Failed" dialogue, they can run the following script to resolve the issue.

**Figure 21** Solution when a cell has files that are locked.

```
#!/bin/csh -f
find . -name "*cdslck"
find . -name "*cdslck" -exec rm -f {} \;
```

Figure 22 Script to unlock files.

3) *Create an inverter's schematic:*

The next step is to create a **schematic** for the circuit, designing it at the **transistor level**. In the **Library Manager** window, follow these steps:

1. Go to **File > New > Cellview**.
2. A window for creating a new circuit will appear, as shown below:

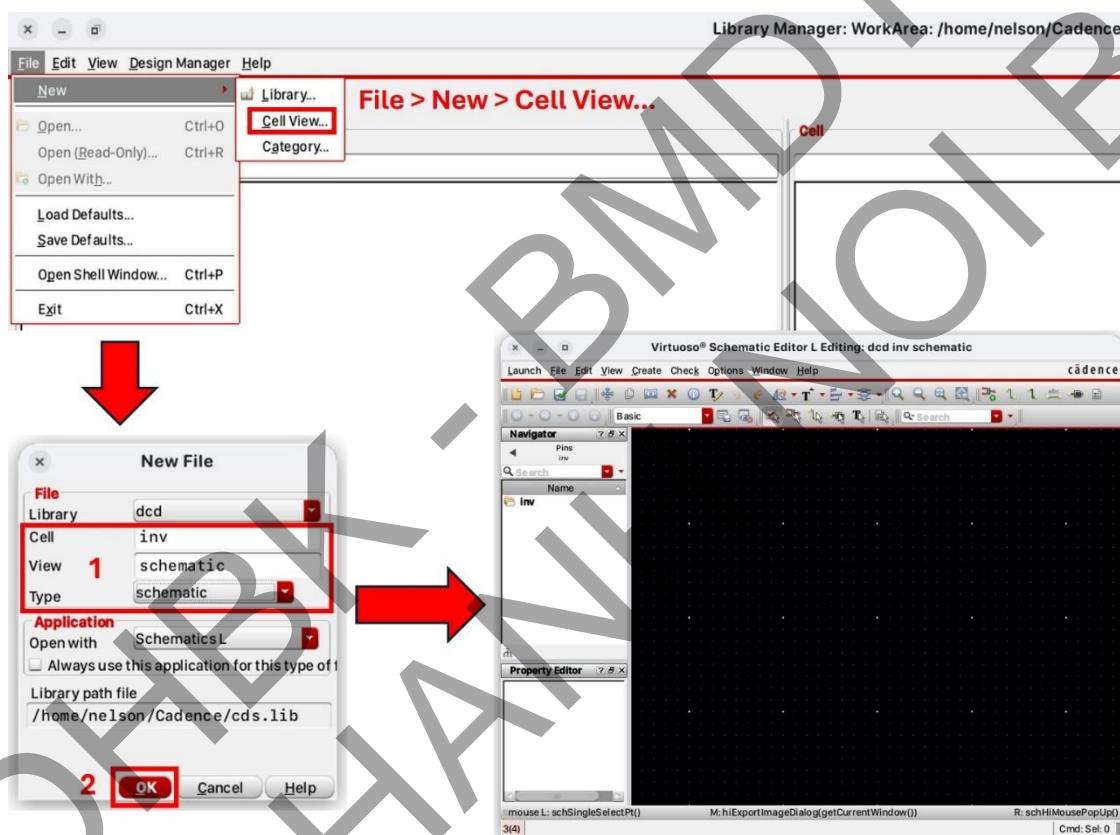


Figure 23 Open Virtuoso Schematic Editor.

Click **OK**, and the **Virtuoso Schematic Editor (VSE)** window will open. This window contains four key components, as listed in Table 3.

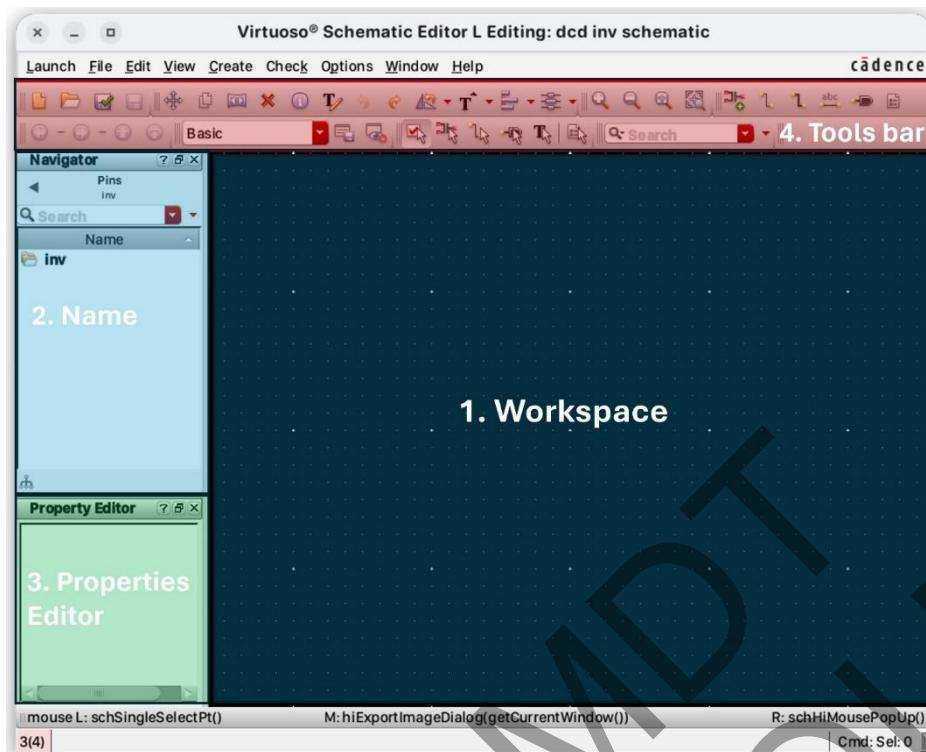


Figure 24 Virtuoso Schematic Editor window

No	Name	Description
1	Workspace	Schematics are drawn in this space. There also include other electronic components coming from PDK.
2	Name	Connections between components are displayed in this space.
3	Properties Editor	Display property of each component chosen. During design period using Virtuoso Schematic Editor (VSA), students can edit parameters faster.
4	Tools Bar	Tools Bar shows us some basic functions for designers. However, for more convenience, users use bindkeys instead of Tool Bars (click each symbol one by one).

Table 3 Description of VSE window

For greater convenience when using **Virtuoso Schematic Design**, students should refer to **Appendix D**, which provides a summary of the most useful features and commands in the **Virtuoso Schematic Editor**, **Virtuoso Symbol Editor**, and **Virtuoso Layout Editor**. To add components to the **Virtuoso Schematic Editor (VSE)** workspace, students can use the **Cadence Editing Shortcut** by pressing the **bindkey "I"**. Additionally, component parameters can be modified at this stage as needed. Students can manually select the **library**, **cell**, and **view** of the desired component or click **Browse** to open the **Library Browser – Add Instance** window.

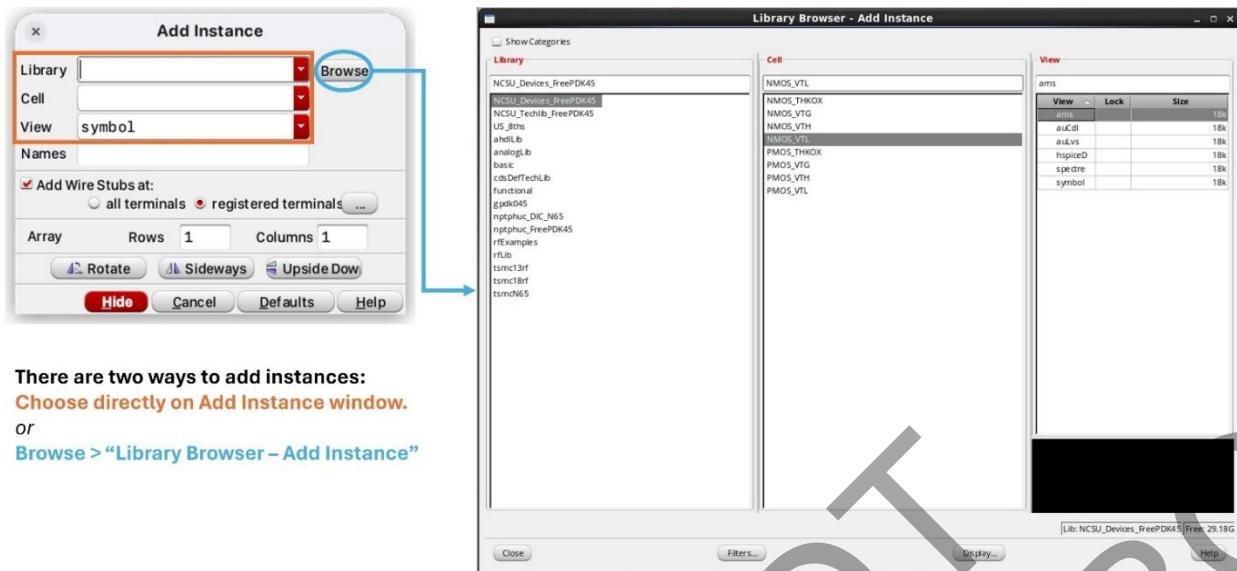


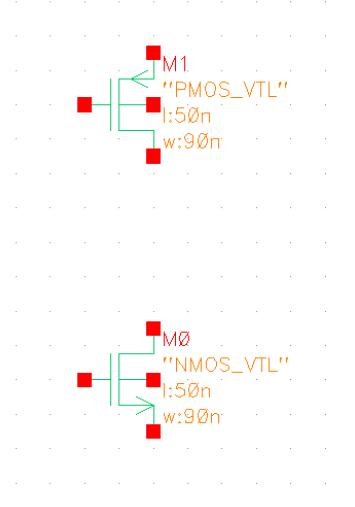
Figure 25 Two ways to add instances.

Four types of devices are supported, corresponding to the low, general, and high threshold voltage devices (VTL, VTG, and VTH). There is also a thick-oxide device (THK0X) for high-voltage off-chip IO. Simulation of these models with a supply voltage of 1.0V yields the following currents:

	Nominal	VTL			VTG			VTH		
		Ion (uA/um)	Ioff (nA/um)	Igate (A/cm ²)	Ion (uA/um)	Ioff (nA/um)	Igate (A/cm ²)	Ion (uA/um)	Ioff (nA/um)	Igate (A/cm ²)
FF Corner					VTL	VTG	VTH	VTL	VTG	VTH
Ion (uA/um)	1246	975.5	570		-801	-650.3	-379.2			
Ioff (nA/um)	100	10	0.2		-100	-10	-0.2			
Igate (A/cm ²)	15.3	6.2	0.8		-14.4	-8.0	-0.6			
SS Corner					VTL	VTG	VTH	VTL	VTG	VTH
Ion (uA/um)	1325	1040	618		-854	-699	-408			
Ioff (nA/um)	229	21.8	0.38		-205.4	-23.6	-0.39			
Igate (A/cm ²)	32	13.1	1.5		-28.9	-15.9	-1.1			
					VTL	VTG	VTH	VTL	VTG	VTH
Ion (uA/um)	1161	905	521		-750	-604	-351			
Ioff (nA/um)	43.1	4.6	0.1		-43	-4.4	-0.1			
Igate (A/cm ²)	6.4	3.3	0.4		-7.1	-4.0	-0.3			

Figure 26 Currents of each model @ a supply voltage of 1.0V.

NMOS_VTL and PMOS_VTL are two components chosen to build students' inverter.

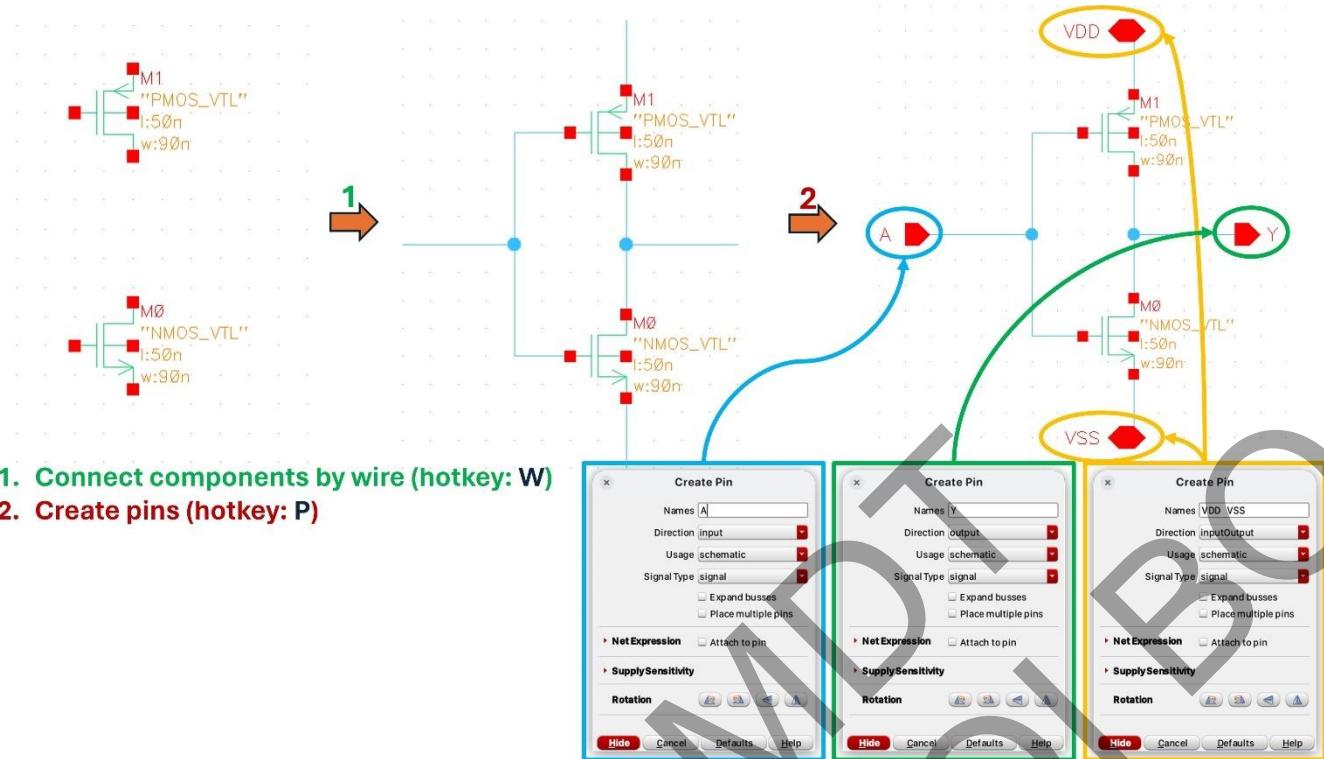
**Figure 27** Take two components on workspace.

Pin name	Direction
A	input
Y	output
VDD	inputOutput
VSS	inputOutput

Table 4 Direction of pins inserted into schematic.

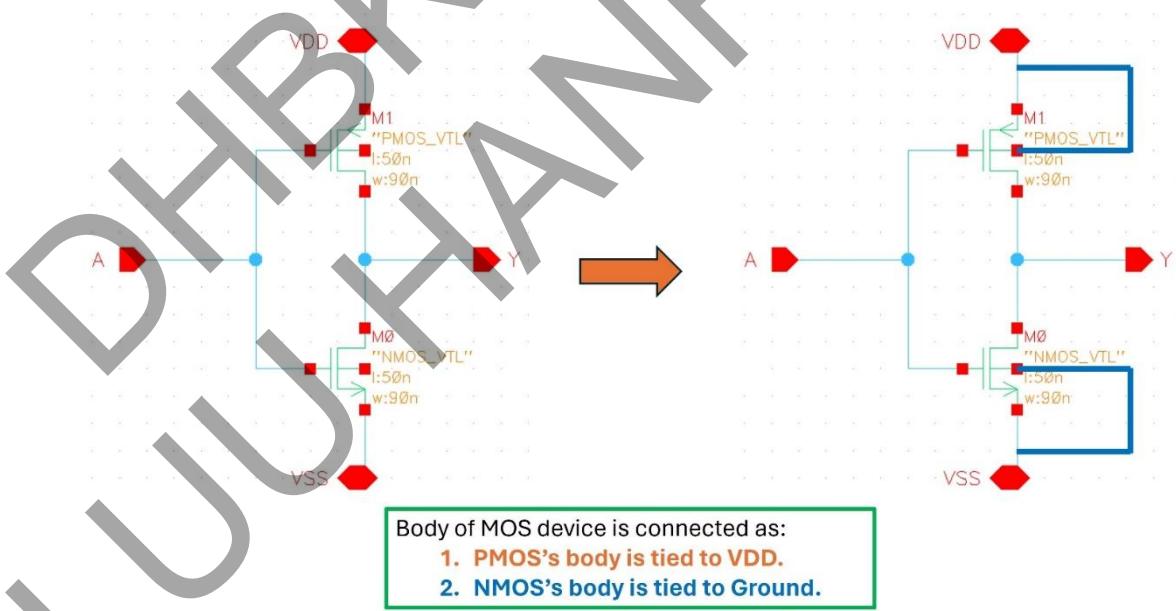
To connect components, use **wires** by pressing the **bindkey "W"**, then complete the **inverter gate diagram**. Next, press the **shortcut key "P"** to create **connection pins** for the circuit. Sequentially create the following pins: **IN**, **OUT**, **VDD**, and **VSS**. The **direction** of each pin is specified in the "**Direction**" field, with types as shown in **Table 5**.

DHBYK
LUUHAN

**Figure 28** Connect components and add pins to the schematic.

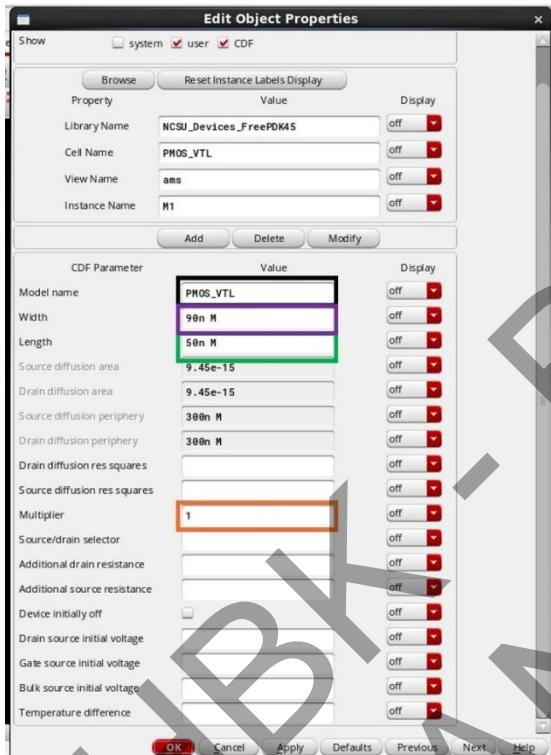
Note that the **bulk** (also known as body or substrate) of NMOS and PMOS transistors must be connected as follows:

- **NMOS bulk** is tied to **ground (VSS)**.
- **PMOS bulk** is tied to **VDD**.

**Figure 29** The complete schematic of the inverter.

Once the schematic is complete, select "Check and Save" or use the shortcut **Shift+X** or press **bindkey F8**.

To modify the attributes of any circuit component, press "Q", then select the desired component. The **attribute table** provides specific **model parameters**, particularly those calculated by **SPECTRE simulation software** based on fundamental **geometric parameters**. **Cadence** extracts values such as **parasitic capacitance and parasitic resistance** from these geometric parameters, which are then applied in **SPECTRE simulations**. For components like **resistors, capacitors, MOSFETs, and BJTs**, model parameters vary based on the **Process Design Kit (PDK)** and depend on the **mathematical model extraction and parasitic calculations** for each component. This document summarizes key component parameters relevant to **design and experimentation**.



1. **Model Name** – spectre model name (non-editable)
2. **Width (M)** – gate width in meters.
3. **Length (M)** – gate length in meters.
4. **Multiplier** – number of Parallel MOS devices.

Figure 30 Edit properties of PMOS_VTL.

No.	Component	CDF Parameter	Description
1	MOSFET	Model Name	spectre model name (non-editable)
		Multiplier	number of Parallel MOS devices
		Length (M)	gate length in meters
		Multiplier	Number of parallel MOS devices.
2	Resistor	Model Name	Spectre model name (non-editable)
		Segments	number of series or parallel segments for a resistor
		Segments Connection	cyclic field used for series or parallel segments
		Calculated Parameter	radio button that determines whether resistance or Length is the calculated value when instantiating a new resistor device
		Resistance	total resistance value equal to the sum of body resistance, contact resistance, end resistance, and grain resistance
		Segment Width	resistor segment width in meters
		Segment Length	resistor segment length in meters

		Effective Width	effective resistor segment width in meters
		Effective Length	effective resistor segment length in meters
3	Moscap	Model Name	spectre model name (non-editable)
		Multiplier	number of Parallel MOS devices
		Calculated Parameter	Calculated parameter cyclic (capacitance, length, width)
		Capacitance	total capacitance
		Length (M)	gate length in meters
		Total Width (M)	gate width in meters (sum of all fingers)
		Finger Width	width of each gate finger/stripe
		Fingers	number of poly gate fingers/stripes used in layout
4	Bipolar	Model Name	Model name used in simulation
		Device Area	Emitter area in microns squared (non-editable)
		Emitter Width	Emitter width microns (non-editable)
		Multiplier	Number of Parallel Bipolar devices
5	Diode	Model Name	Model used for simulation name
		Calculated Parameter	Choices are 'area', 'width', or 'length'
		Device Area	Calculated junction area in meters squared (non-editable)
		Length (M)	Diode length in meters
		Width (M)	Diode width in meters
		Multiplier	Number of Parallel Diode devices

Table 5 Some important CDF parameters of components in library.

4) Create a symbol of inverter:

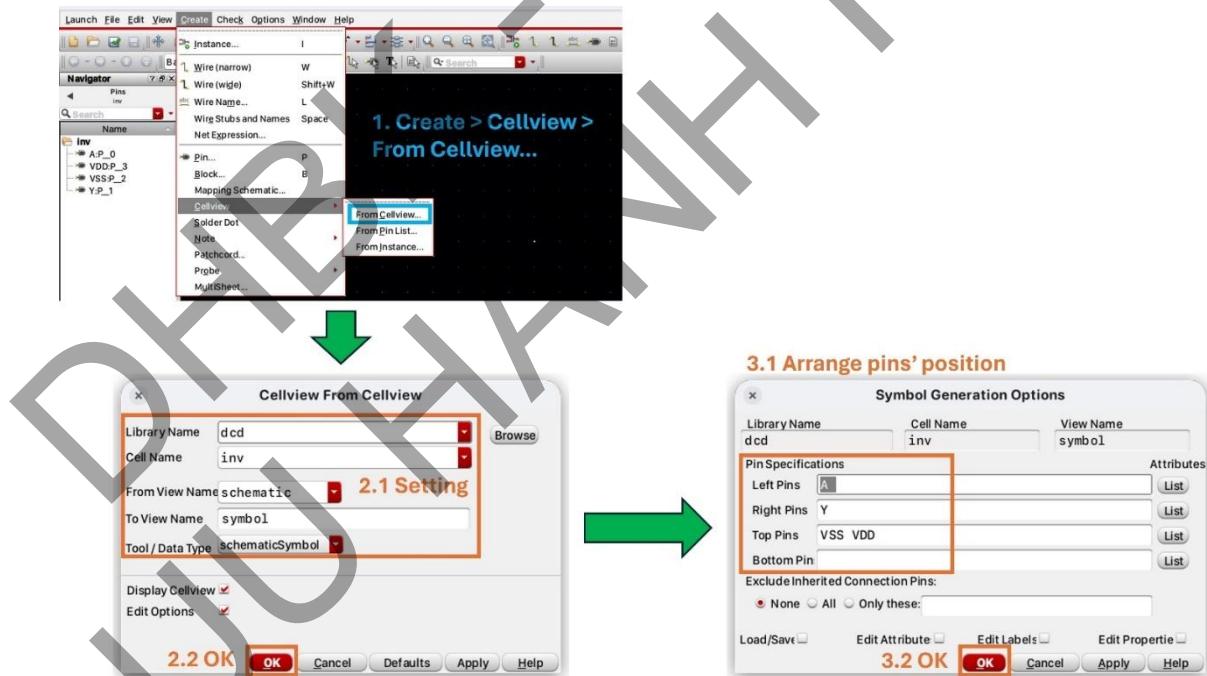


Figure 31 Prepare to create an inverter's symbol.

The screen now switches to the workspace for **drawing symbols** for logic gates. This tool provides an intuitive way to create **representations of an inverter gate and other circuits**. From a graphical standpoint, **Cadence** offers all the necessary tools for designing an **inverter gate**, which is drawn using

the same graphical tools as before.

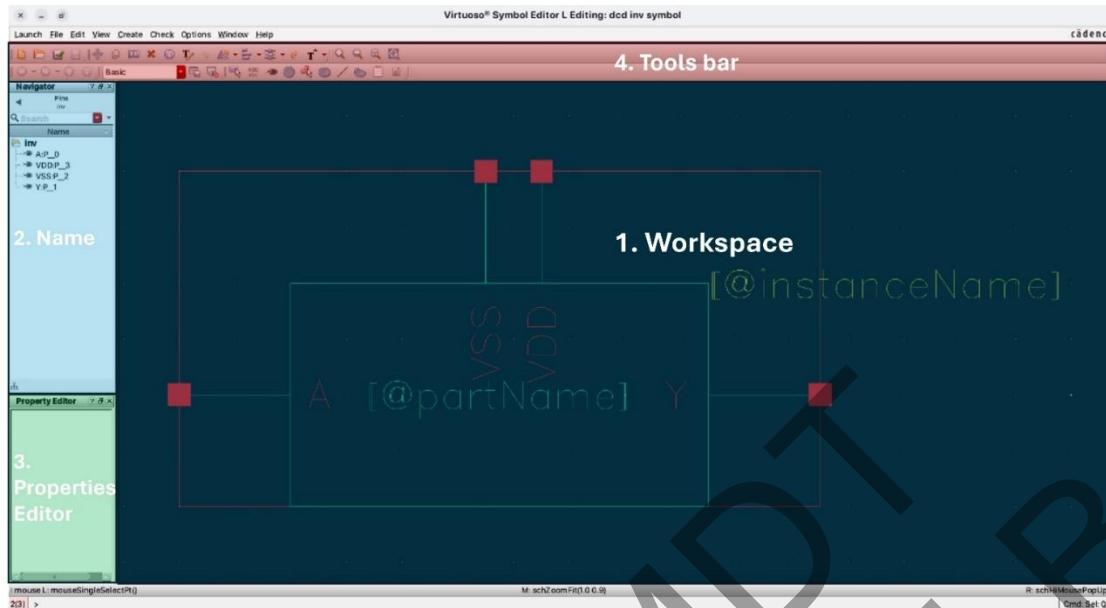


Figure 32 Virtuoso Symbol Editor window.

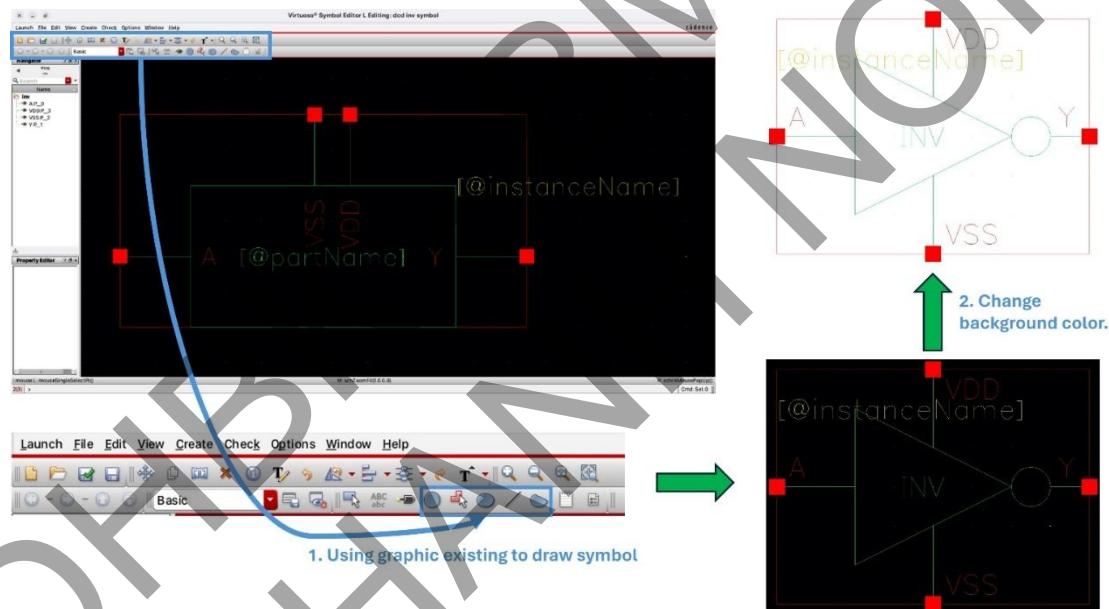


Figure 33 Re-draw inverter's symbol and change the background color.

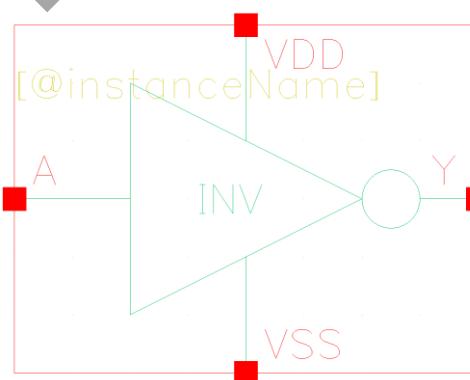


Figure 34 The complete symbol of the inverter.

PART 3

Objective: Pre-layout simulation.

Requirements: Simulate DC analysis, transient by using SPECTRE for the inverter.

Instructions:

Transient simulation is a **time-domain analysis** method used to examine signal waveform variations due to **noise or distortion**, such as **harmonic distortion**. These effects cannot be analyzed using **AC analysis**.

DC analysis, similar to **SPICE analysis**, allows designers to evaluate a circuit's behavior in the **DC domain**, providing insight into the **operating points** of individual components. For instance, it can verify whether all transistors are functioning within the **desired active region**. This analysis helps determine optimal operating conditions, such as identifying the appropriate **bias current** for an amplifier to ensure all transistors reach **deep saturation**, maximizing **signal gain**.

In this section, students will learn how to create a **testbench circuit** (inverter schematic vs. ideal inverter) and perform **transient simulation** followed by **DC analysis**.

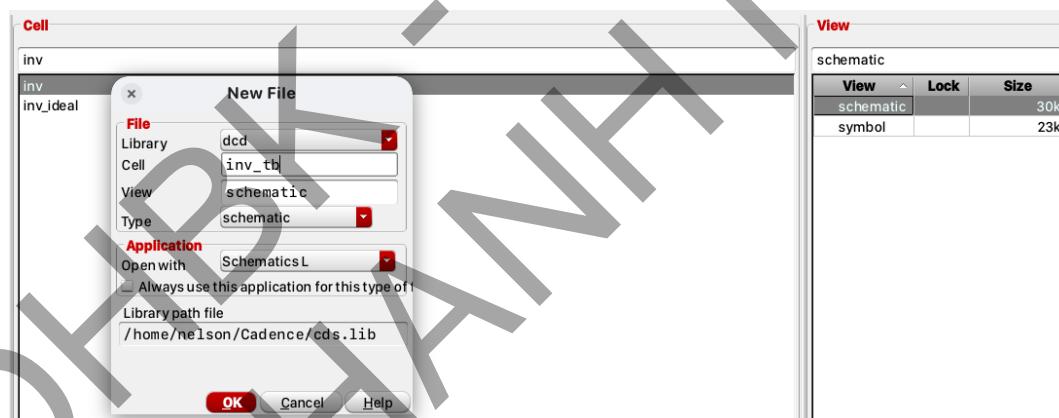


Figure 35 Create a testbench for the inverter.

During the component selection process, certain components are frequently used and play a crucial role in simulations or test model designs for logic circuits. The most commonly used elements in experiments are listed in the table below.

Component	Type	Description
vdd	Global components	▪ Have no component parameters.
gnd		
vdc	Independent components	▪ Independent voltage source. ▪ A constant vsource.
idc		▪ Independent DC current source. ▪ a constant isource.

vsin		<ul style="list-style-type: none"> ▪ Independent sinusoidal voltage source. ▪ A sin wave vsource.
isin		<ul style="list-style-type: none"> ▪ Independent sinusoidal current source. ▪ A sin wave isource.
vpulse		<ul style="list-style-type: none"> ▪ Independent pulse voltage source. ▪ A Square wave varying vsource.
ipulse		<ul style="list-style-type: none"> ▪ Independent pulse current source. ▪ A Square wave varying isource.
vcvs	Dependent component	<ul style="list-style-type: none"> ▪ Linear voltage-controlled voltage source. ▪ Current through the voltage source is calculated and is defined to be positive if it flows from the positive terminal, through the source, to the negative terminal.
res		<ul style="list-style-type: none"> ▪ Two terminal resistors. ▪ Students can give the resistance explicitly or allow it to be computed from the physical length and width of the resistor. In either case, the resistance can be a function of temperature or applied voltage.
cap	Passive component	<ul style="list-style-type: none"> ▪ Two terminal capacitors. ▪ Students can assign the capacitance or let Spectre compute it from the physical length and width of the capacitor. In either case, the capacitance can be a function of temperature or applied voltage.
ind		<ul style="list-style-type: none"> ▪ Two terminal inductors. ▪ The inductance of this component can be a function of temperature or branch current. If students do not specify the inductance in the instance statement, it is taken from the model.

Table 6 Some components are used frequently in analogLib.

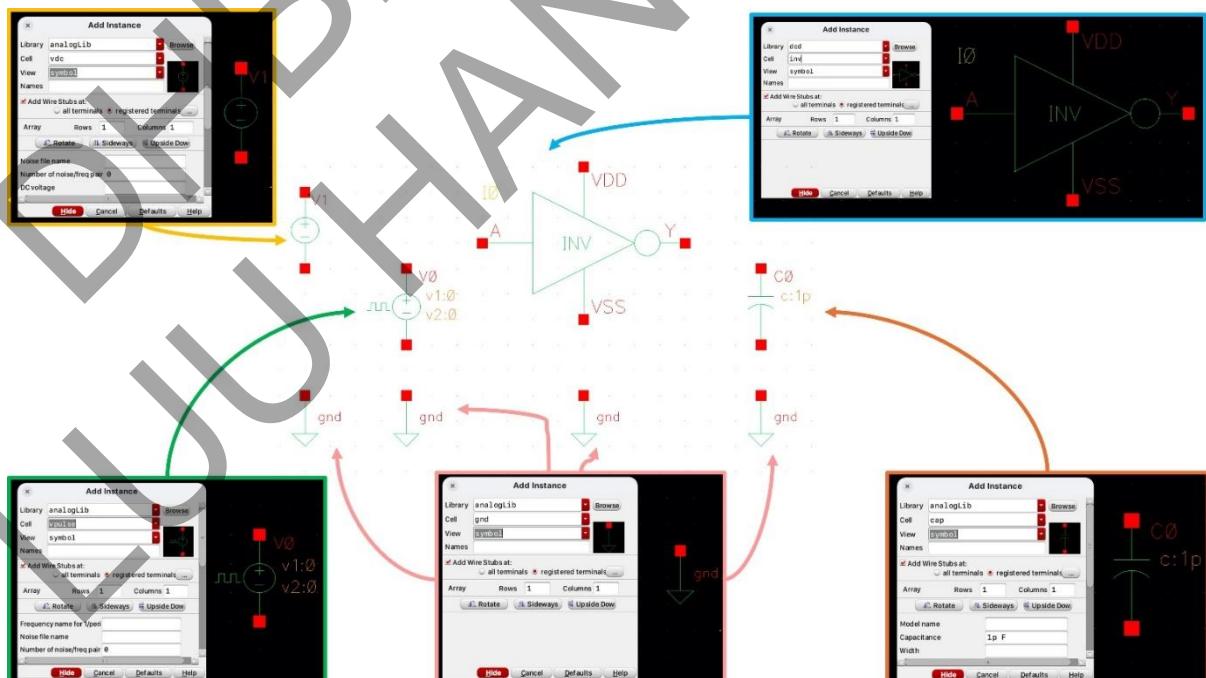


Figure 36 Arrange components for inverter testbench.



Press "W" to connect components and complete the testbench.

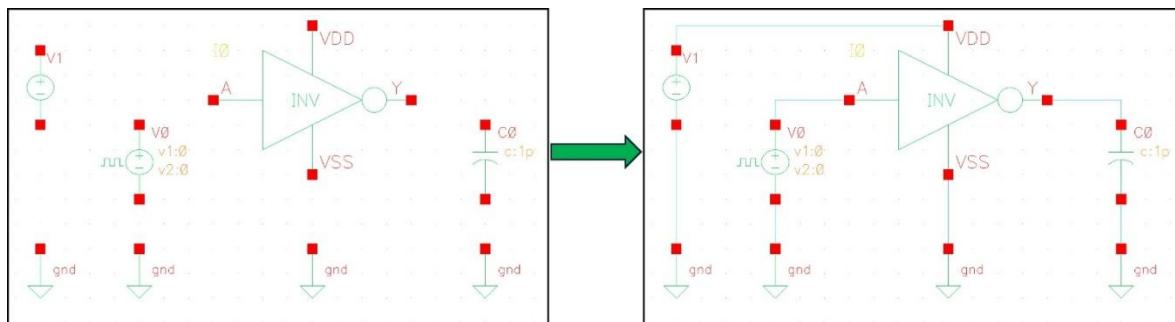


Figure 37 Connect components using wire (bindkey “W”).

Students need to set values for each component before running the simulation. These parameters are shown in Table 5

Component	CDF Parameter	Description	Value
vdc	DC voltage		1
cap	Capacitance		1fF
vpulse	Voltage 1	Initial voltage	1
	Voltage 2	Peak voltage	0
	Period	Period	4n
	Delay time	t_d – initial delay time	1.75n
	Rise time	t_r – rise time	1p
	Fall time	t_f – fall time	1p
	Pulse width	pw – pulse width	1.8n

Table 7 CDF parameters description with values.



Figure 38 Set value for vdc.

The image depicts a **pulsed waveform**, commonly used in **digital circuit simulations** and **signal analysis**. This waveform alternates between two voltage levels: **v1 (low)** and **v2 (high)**, with well-defined timing characteristics.

- The signal starts with an initial **time delay (td)** before transitioning from **v1 to v2**.
- This transition occurs over a specified **rise time (tr)**.
- The signal remains at **v2** for a duration known as the **pulse width (pw)**.
- It then returns to **v1** over a defined **fall time (tf)**.
- The entire cycle repeats with a total duration known as the **period (per)**.

these parameters are essential in **digital circuit design**, ensuring proper timing and functionality in **integrated circuit simulations**. Students should pay close attention to the values of **voltage 1 (v1)** and **voltage 2 (v2)** when analyzing the waveform.

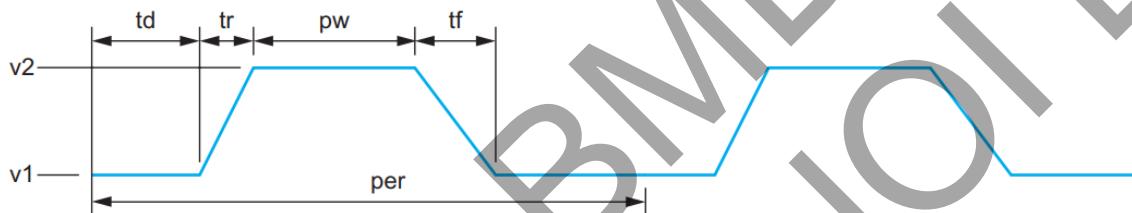


Figure 39 Pulse waveform.

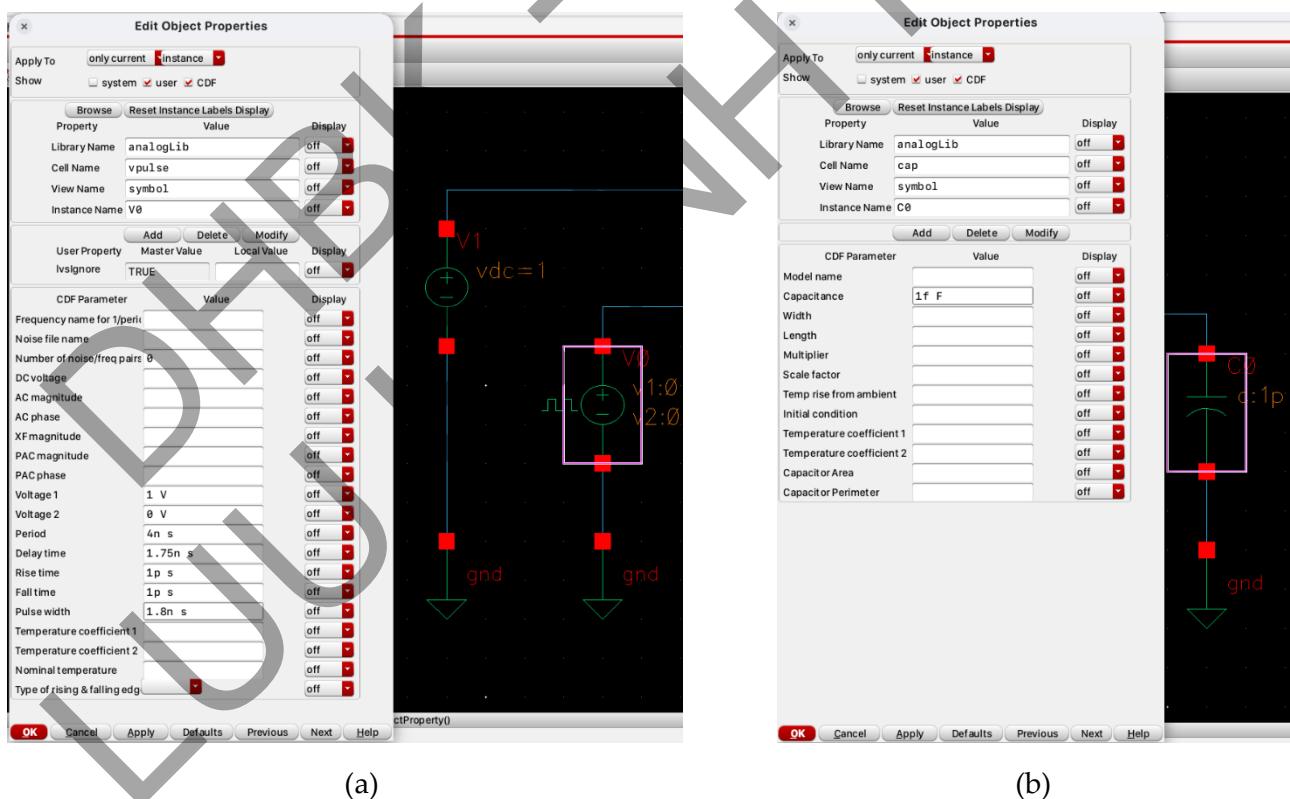


Figure 40 (a) Set value for vpulse (b) Set value for cap.

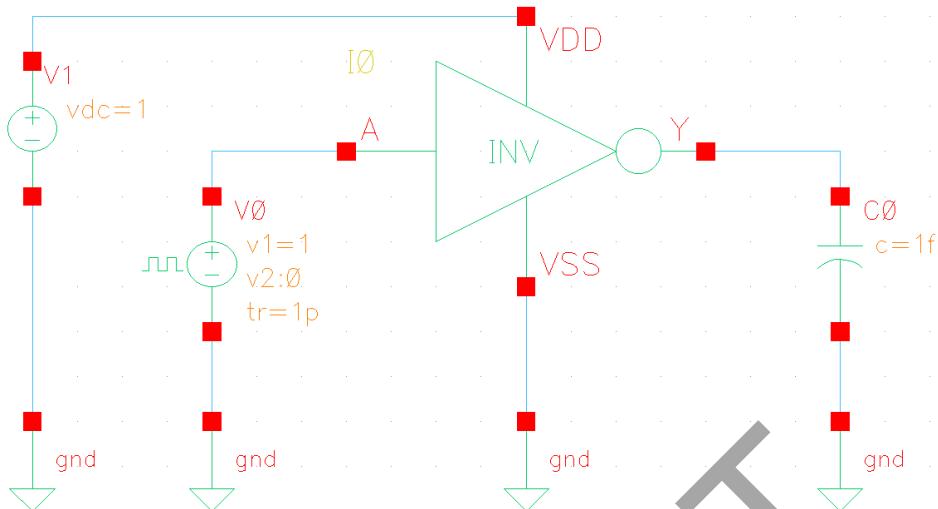


Figure 41 Inverter testbench without labeling for transient simulation.

To label components, press the "L" shortcut key to open the **wire naming window**. Enter the component names, such as "**input**" and "**output**," as shown in the diagram below. Students can adjust the **size, color, and font style** based on their design objectives, ensuring clarity and readability.

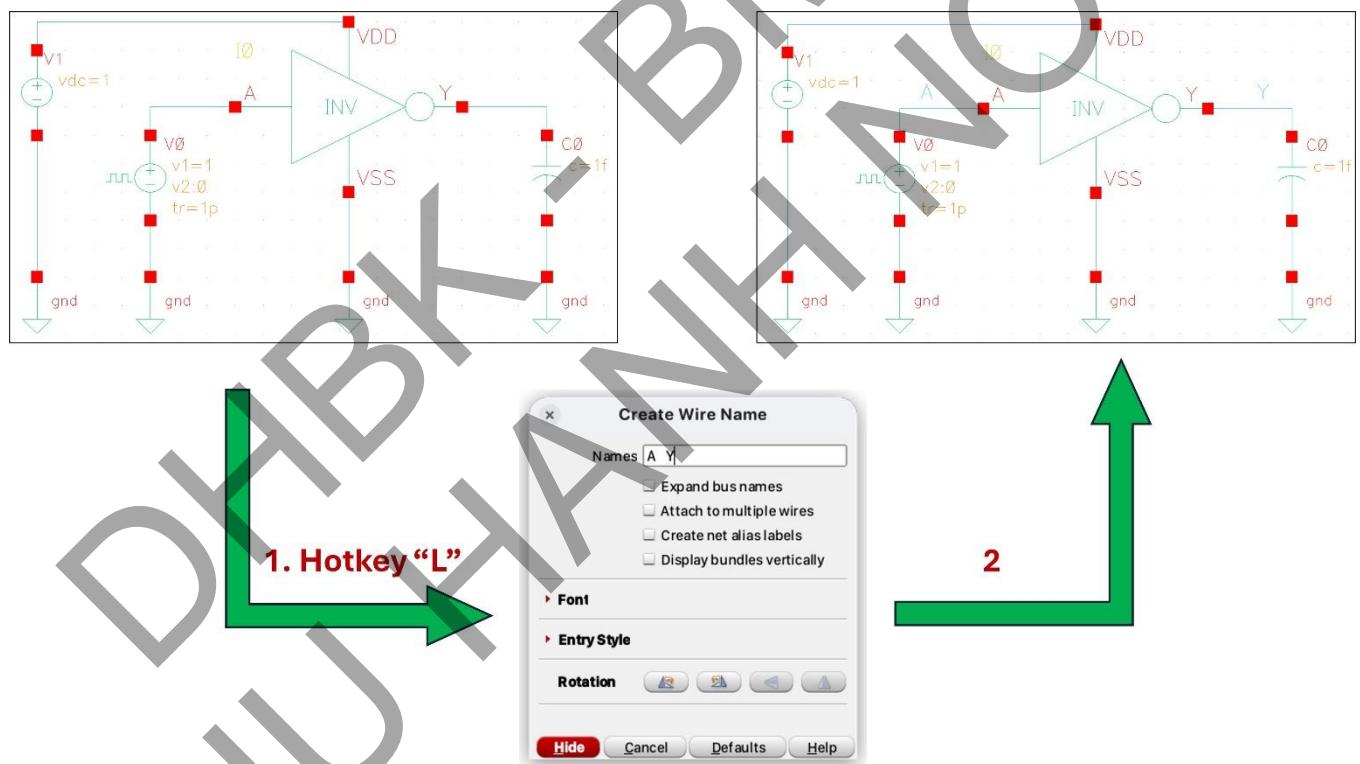


Figure 42 Inverter testbench after labeling for transient simulation.

Students can also arrange components as shown below. Using labels enhances the professionalism of the schematic.

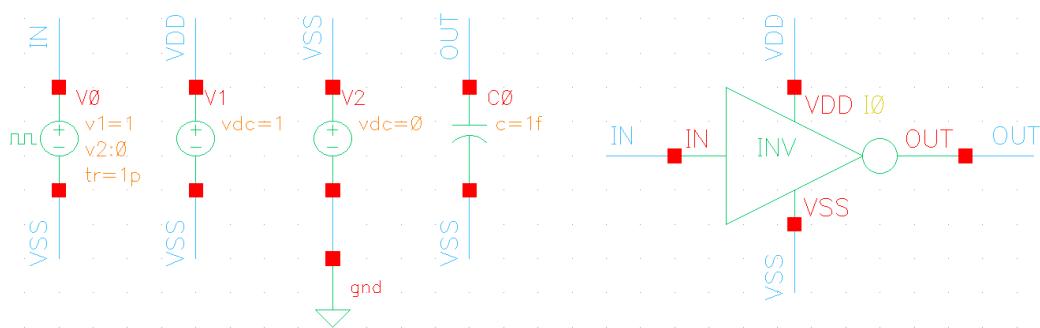


Figure 43 Inverter testbench after labeling for transient simulation.

The circuit is divided into two sections: one for testing the operation of the designed CMOS-based inverter and the other for simulating its ideal waveform. In the Cadence Virtuoso environment, simulations are conducted using the **Analog Design Environment (ADE)**, a graphical interface for circuit analysis.

To launch ADE, follow these steps in the schematic design window: **Launch > ADE L**.

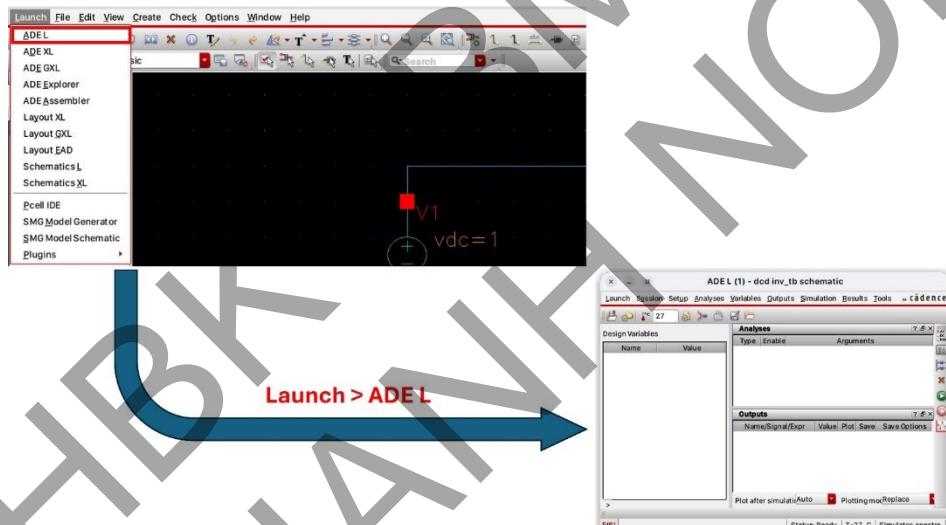


Figure 44 Open ADE L.

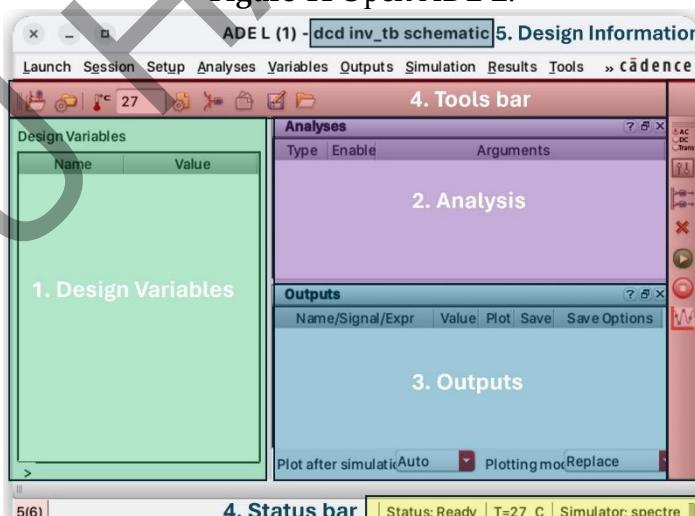


Figure 45 Analog Design Environment L (ADE L).

No.	Name	Description
1	Design variables	Declare existing variables in the principle circuit design. The use of variables in the design is crucial because based on these variables, the designer can select design values to ensure the circuit operates under optimal conditions.
2	Analysis	The region contains and displays the analysis modes, which can be DC, AC, time-domain (transient), etc. This area is also used to adjust the configured modes.
3	Outputs	This area contains values that the designer wants to calculate or display as waveforms to verify circuit operation. It is possible to display or save the simulation results of these values in this table. However, adding too many outputs can slow down the software's execution speed.

Table 8 Description of Analog Design Environment (ADE) L.

In this simulation environment, the first step is to choose the appropriate simulation software. Cadence offers various integrated simulation tools, each with distinct features suited for different applications. The choice of a simulation tool depends on the circuit's application and the specific simulation objectives, which can be categorized into analysis methods, simulation speed, and accuracy in complex systems.

Common analysis methods include **DC analysis, small-signal equivalent modeling, transient analysis, noise analysis, harmonic analysis, and sensitivity/stability analysis**. In this experimental course, the focus is on fundamental techniques used in analog circuit design, specifically **DC, AC, and transient analysis**. A typical circuit simulation setup consists of four key steps:

1. **Setting up the model library**
2. **Selecting the analysis mode**
3. **Configuring the output signals**
4. **Defining the design variables**
 - a) **Setting up the model library:**

At this step, students should select “**Attach to an existing technology file**” in **Part 2**. Since this option is already chosen, no further action is needed. However, if students are using a different library, they should click **Browse** and verify their selected library.



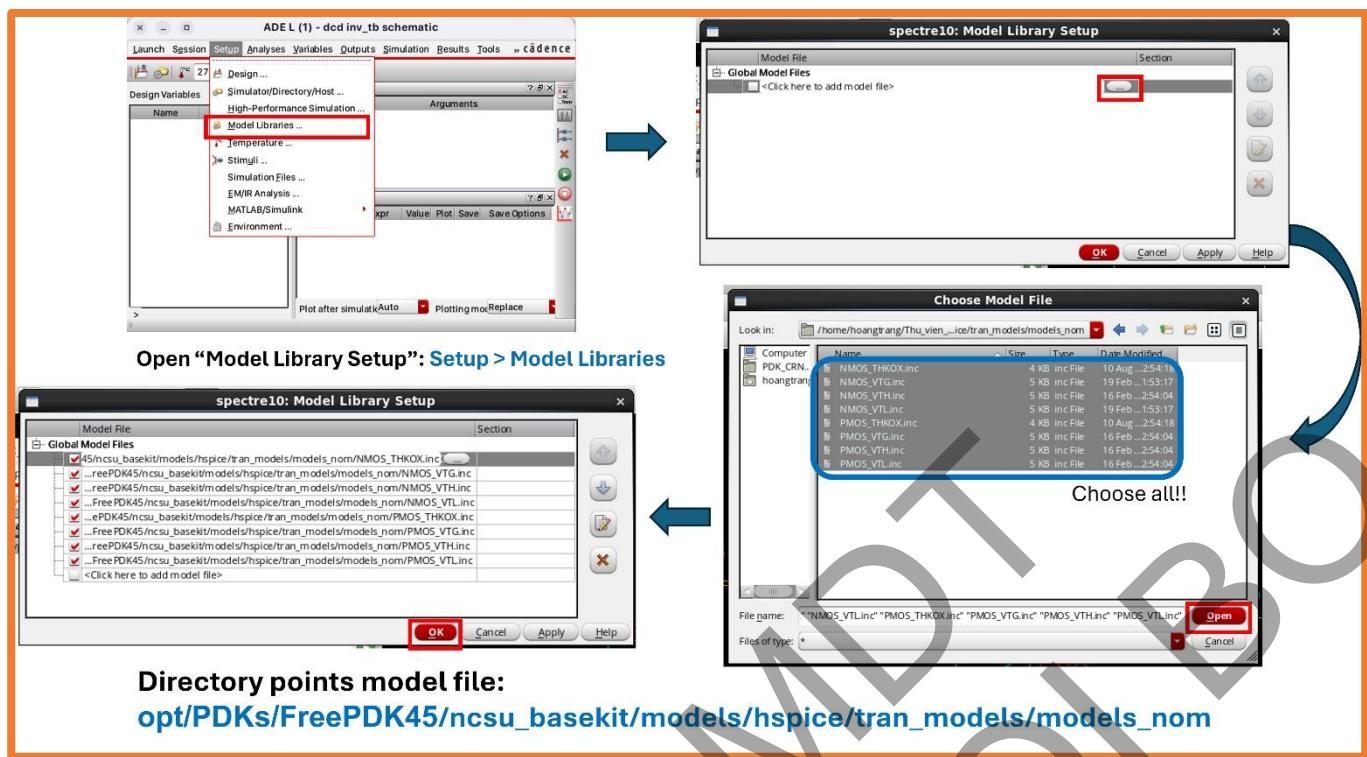


Figure 46 Setup Model library before simulating.

b) Setting the analysis mode:

Next, select the analysis mode by navigating to **Analyses > Choose**.

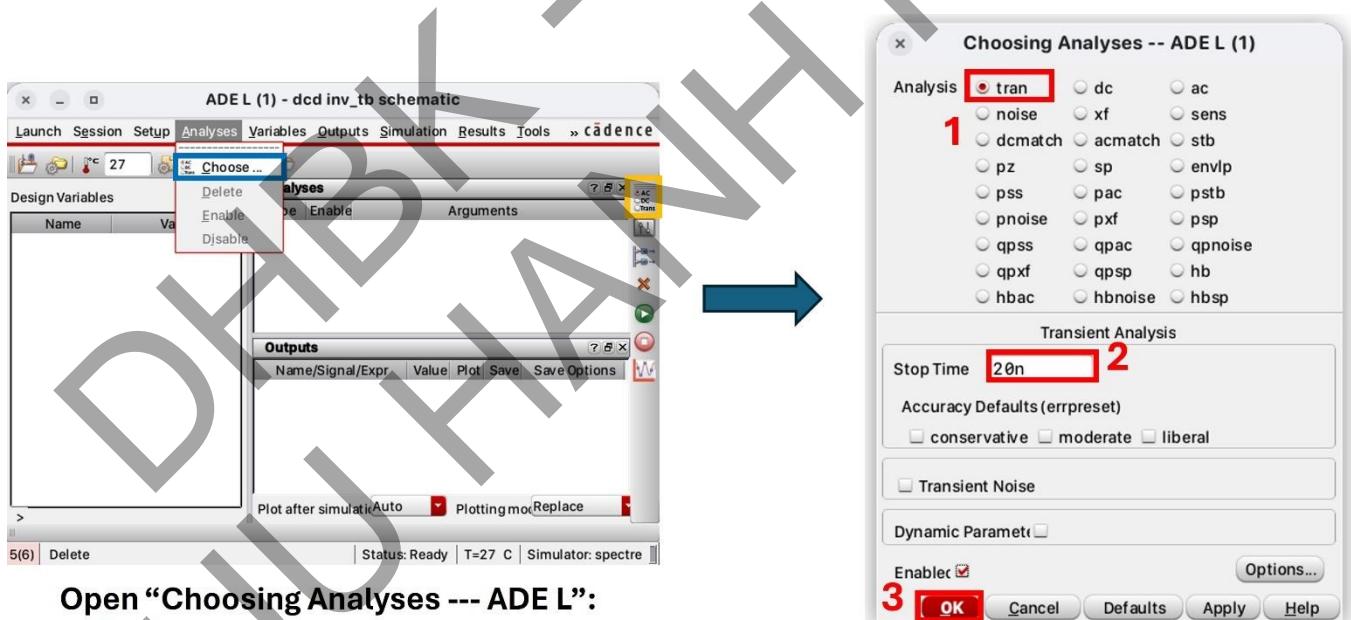


Figure 47 Choose analysis type for simulation.

c) Configuring the output signals:

Select the **output signals** to observe. This can be done in two ways: either by **clicking the icon** or selecting the option from the **taskbar**.

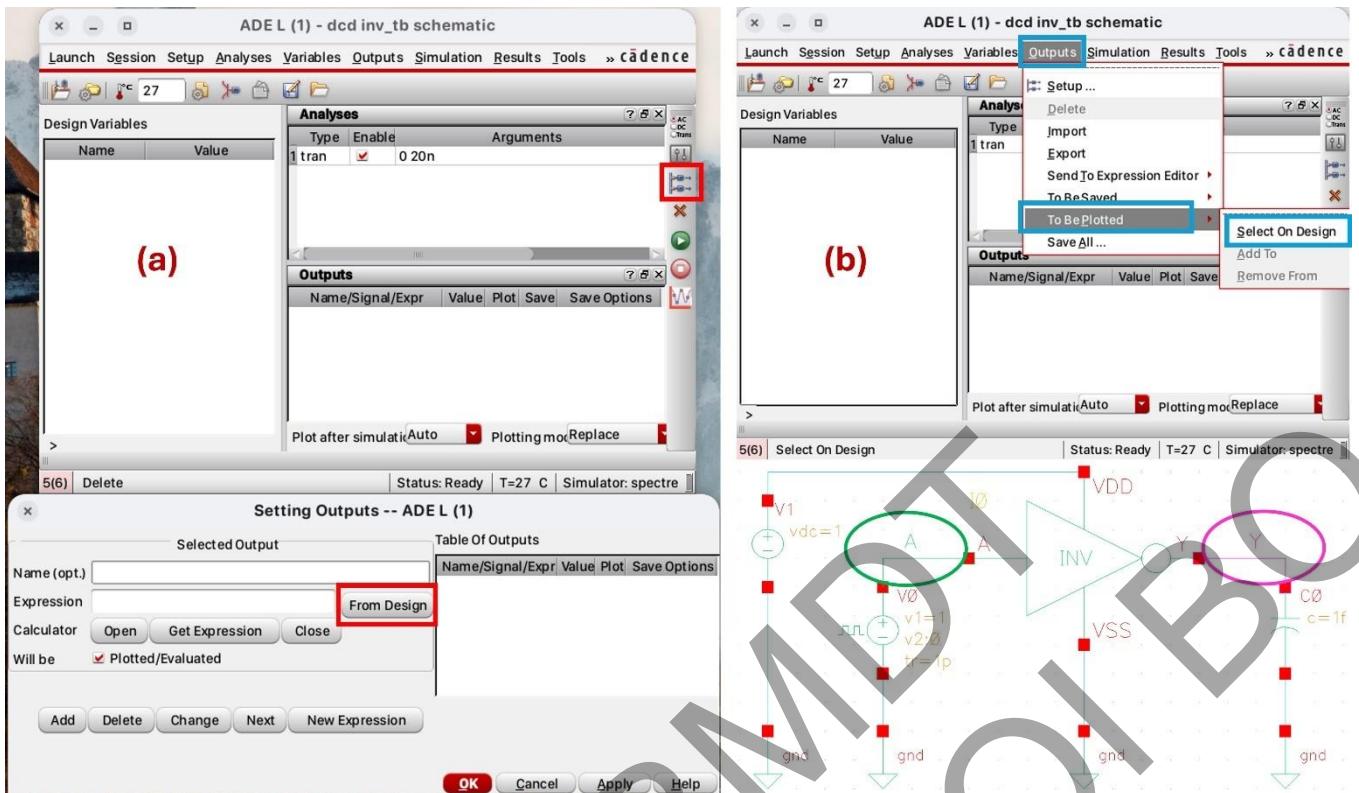


Figure 48 Two methods to select plotted output signals (a) click the icon > From Design or (b) navigate to Outputs > To Be Plotted > Select On Design.

With all preparations complete, click the green icon to start the transient simulation.

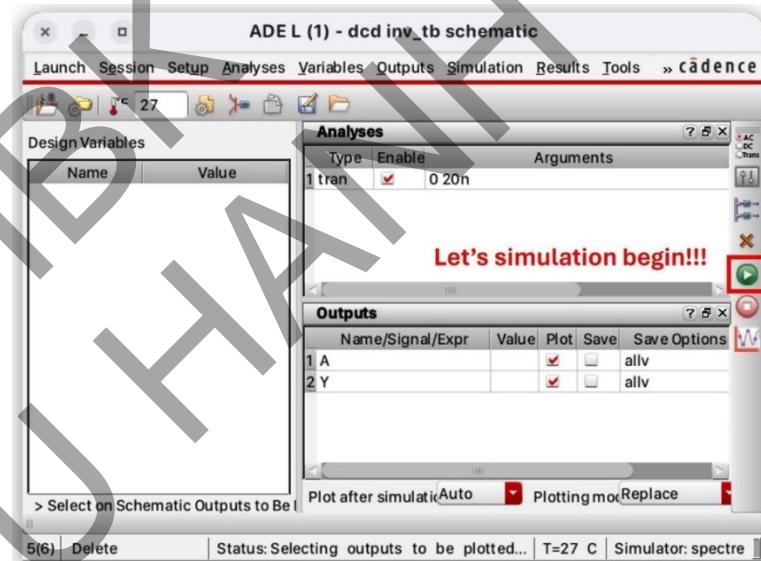


Figure 49 Setting complete and begin the transient simulation.



Figure 50 The result of the inverter’s transient simulation.

This waveform may be difficult to observe, especially when printed. In **Part 2**, students learned how to capture images directly and set a transparent color using **MS Office 365**. Additionally, this document will explore an alternative method to change the background color in **Virtuoso Visualization & Analysis XL**.

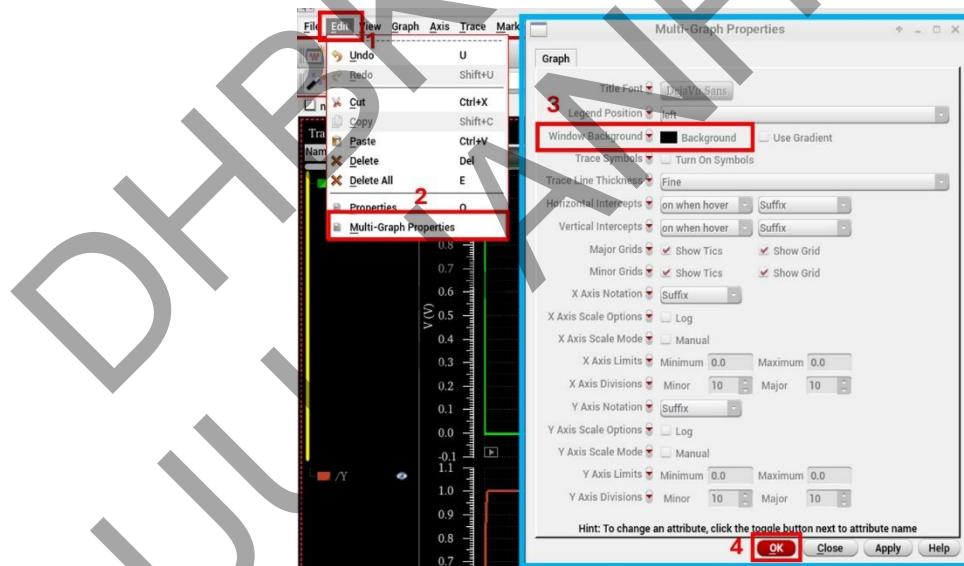
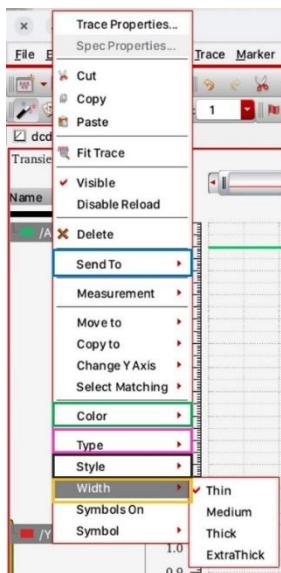


Figure 51 Change the background color of the output waveform.

If the output waveforms appear too thin to observe, students can adjust them by selecting the desired waveform, **right-clicking**, and choosing **Width** from the available options: **thin**, **medium**, **thick**, or **extra thick**. Additionally, students can customize the **color**, **line type** (e.g., continuous, points, histogram, bar,

spectral, or sample hold), and **line style** (e.g., solid, dot, dash, dash-dot, or dash-dot-dot).



1. Send to [define that signal to calculator]
2. Color
3. Type: continuous line, points, histogram, bar, spectral, sample hold.
4. Style: solid, dot, dash, dashdot, dashdotdot.
5. Width: thin, medium, thick, extrathick.

Figure 52 Some signal editing options.

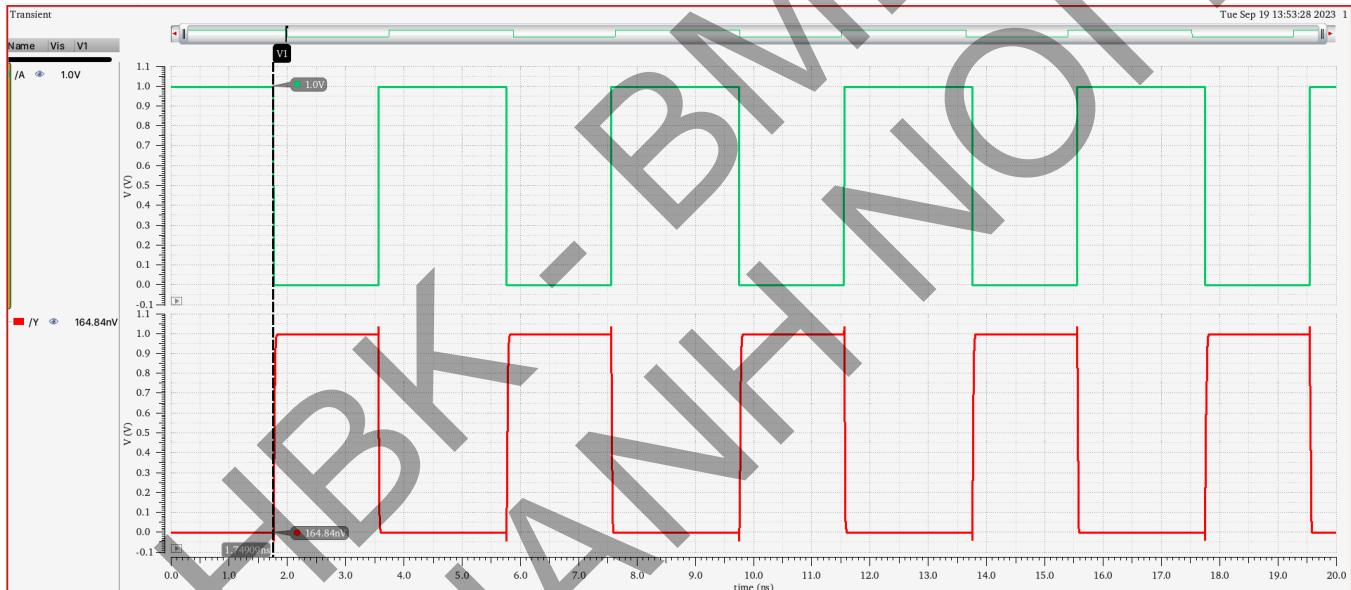


Figure 53 The waveform after changing the background.

Similarly, students should configure a simulation to compare the **ideal inverter** implemented in **Verilog-A** (from the beginning of Part 2) with the **CMOS-based inverter** by analyzing their output signals.

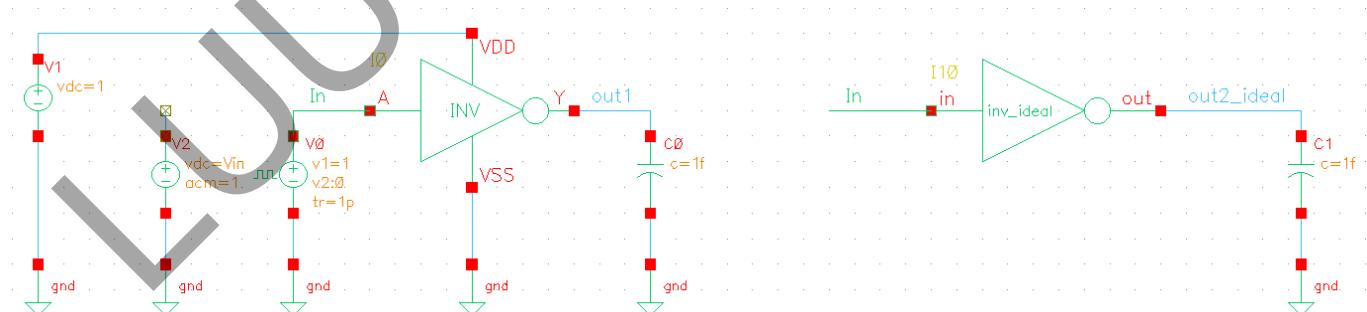
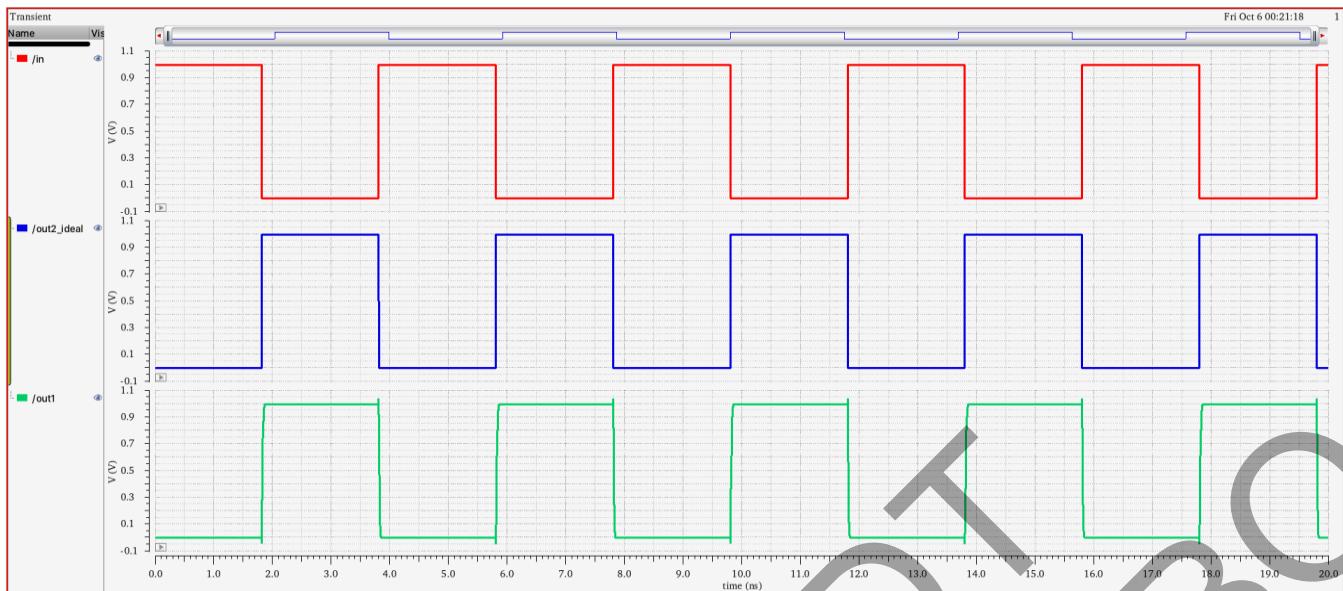
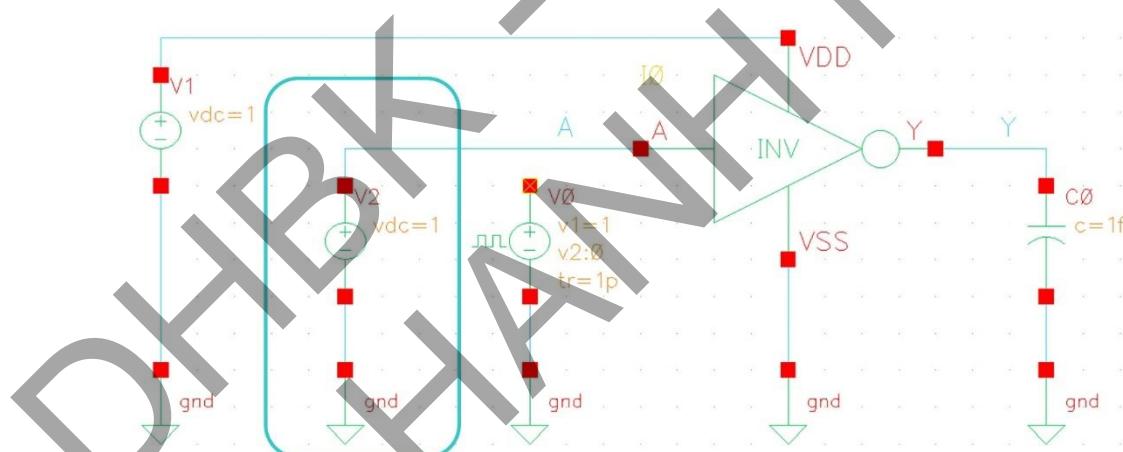


Figure 54 Comparison between ideal inverter and schematic of inverter.

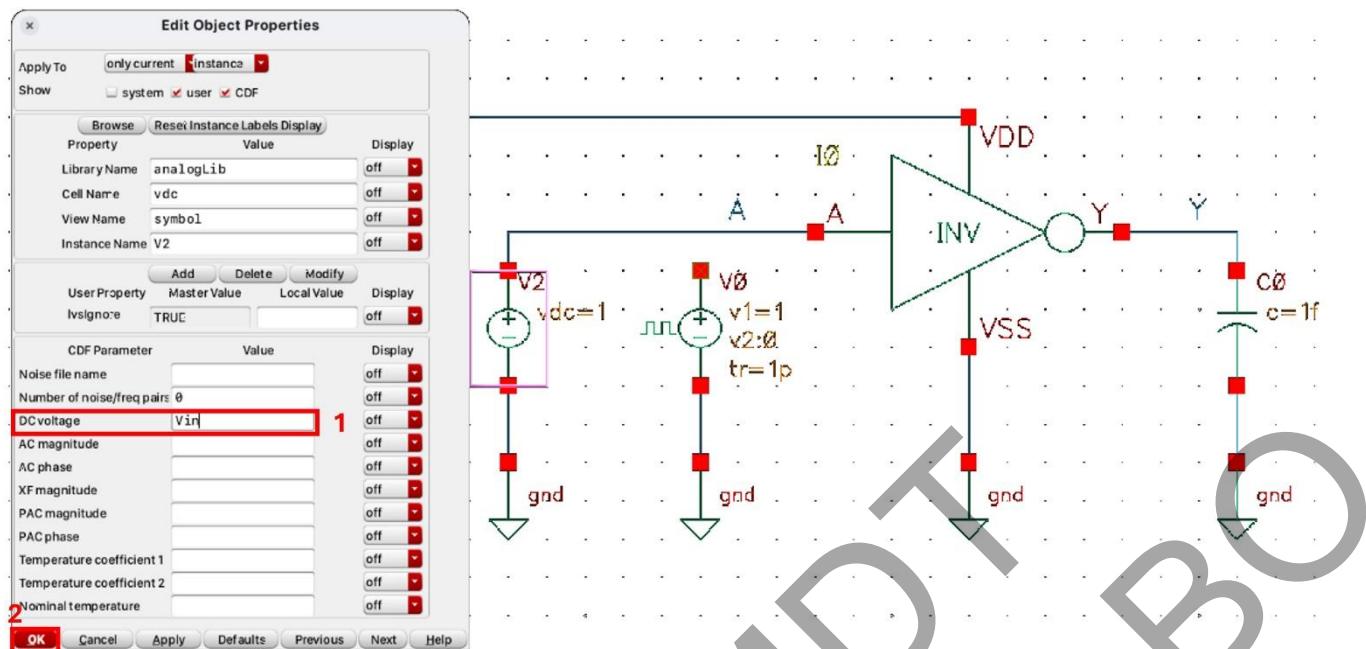
**Figure 55** The waveform of each component.

Please observe the difference between the two output waveforms and find an explanation for this problem.

For DC analysis, students can reuse the inverter testbench from the transient simulation by replacing vpulse with vdc.

**Figure 56** Replacing the vpulse with the vdc to simulate DC analysis.

The value of that VDC will be a variable to sweep input data.

Figure 57 Set value for v_{dc} .

Before running the simulation, students should follow these steps:

1. Add design variables from the existing design by navigating to **Variables > Copy from Cellview** or by right-clicking in the **Design Variables** dialog box. Assign an initial value, such as **1V**.
2. Select **Analyses > Choose** to configure the simulation settings.

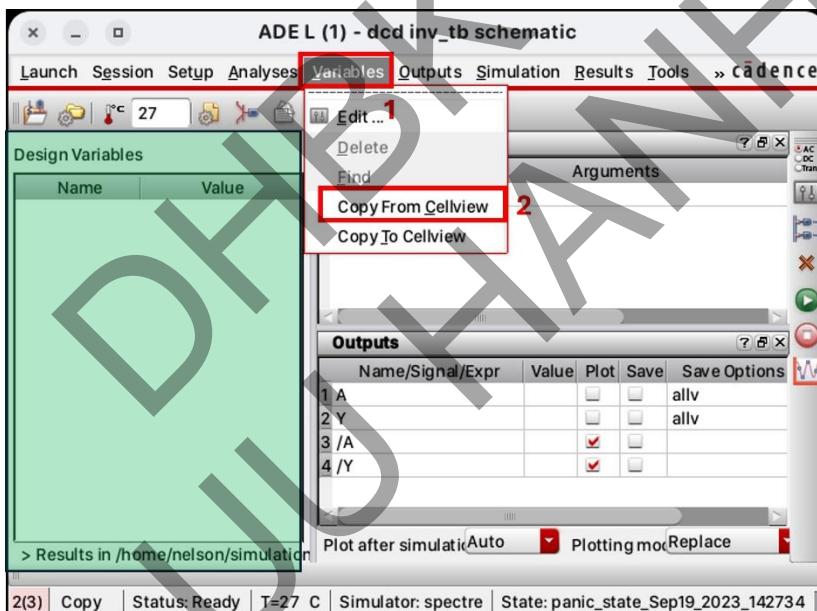


Figure 58 Setting up the design variables.

Setting for DC analysis as Figure below:

Update Design Variables from Schematic:

1. Choose **Variables > Copy From Cellview** or
2. Click right mouse at Design Variables and choose **Copy from Cellview**.

Initialize for this variable.

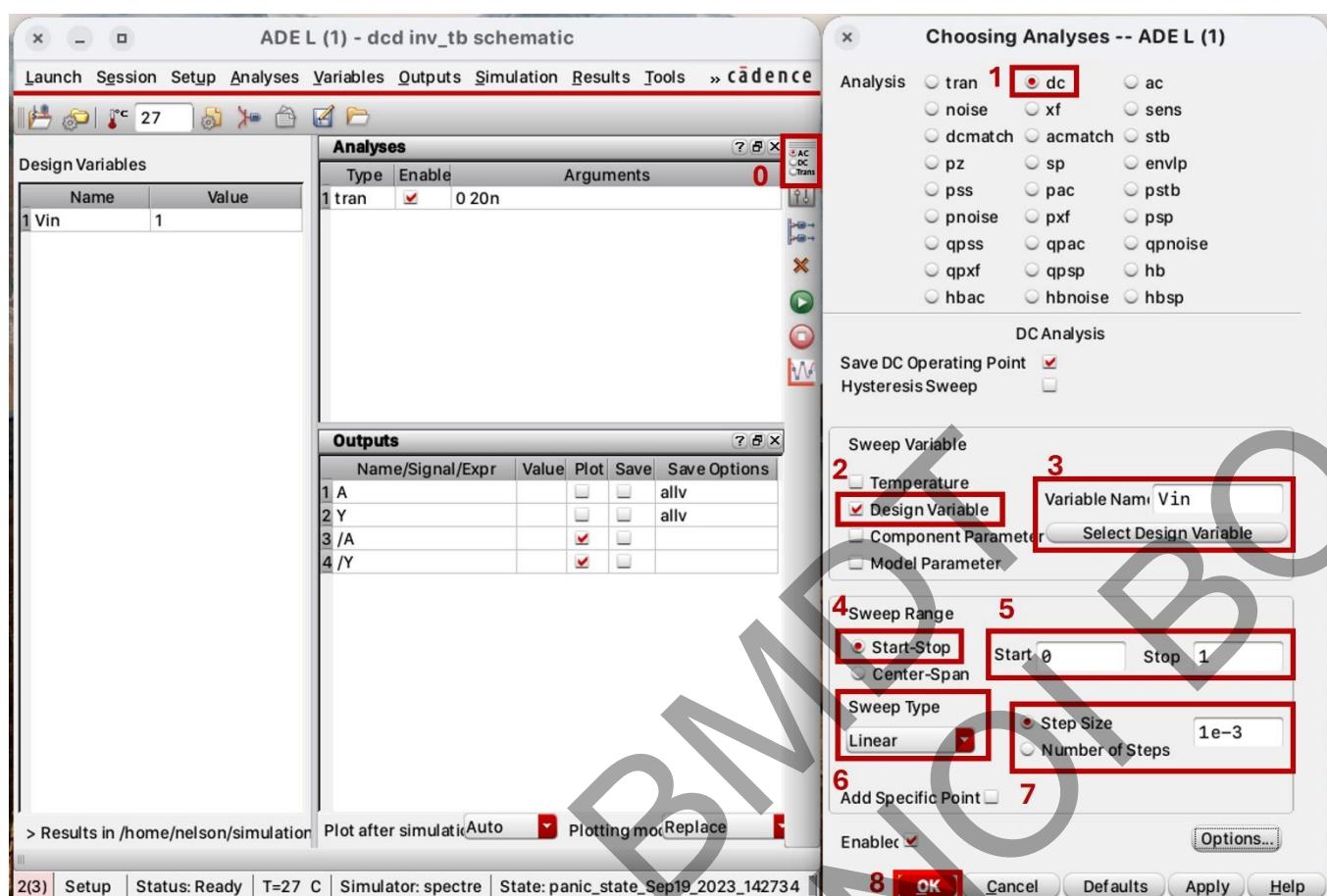


Figure 59 DC analysis setting instructions.

After preparation is completed, let's begin the DC analysis.

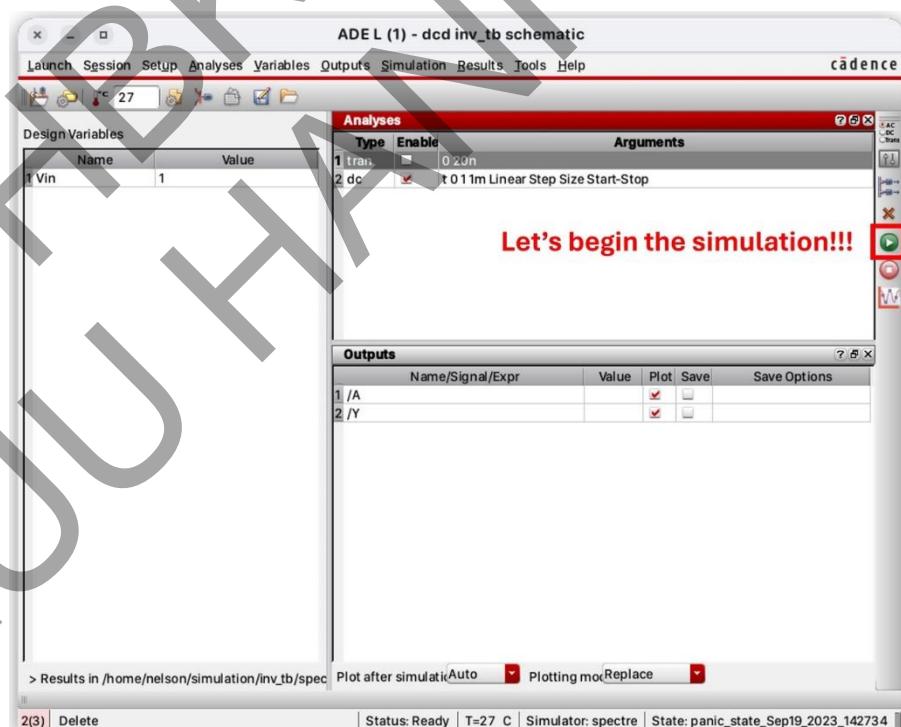


Figure 60 Setting complete and begin the DC analysis.

The result after running the DC analysis:

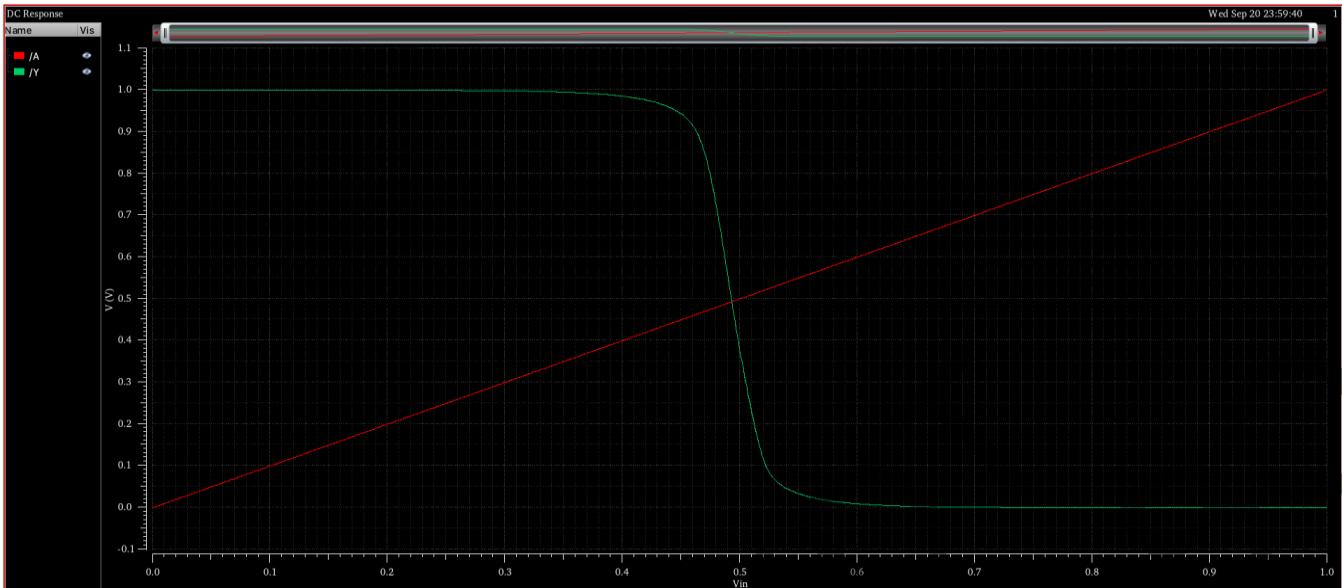


Figure 61 The output waveform of inverter's DC analysis.

Following the instructions to adjust the background color and curve thickness, the waveform is modified accordingly:

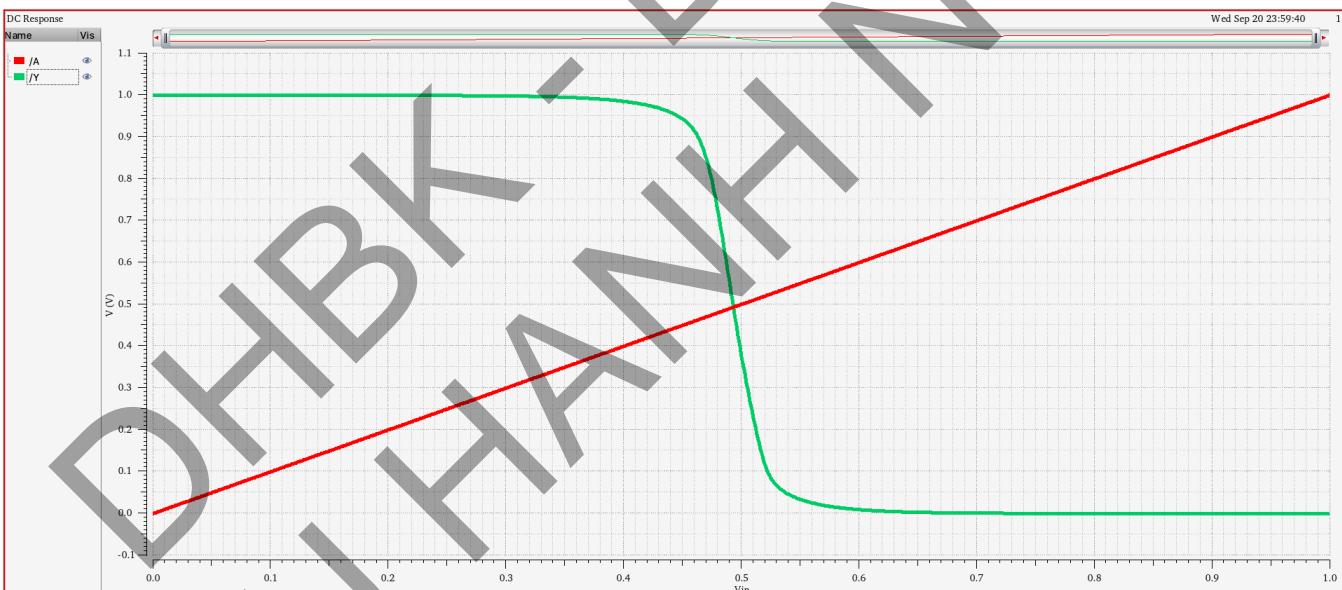


Figure 62 The output waveform with background color edited.

Additionally, students can check **DC Operating Points** and **DC Node Voltages**. First, they need to set up the inverter testbench. In the ADE L window, navigate to Choose Analyses – ADE L.

- Under **Transient (trans)** analysis, set the **stop time** to $10 \times \text{Input_period}$.
- Under **DC (dc)** analysis, select “**Save DC Operating Points**”.

Click **Apply**, then **OK** to confirm the settings.



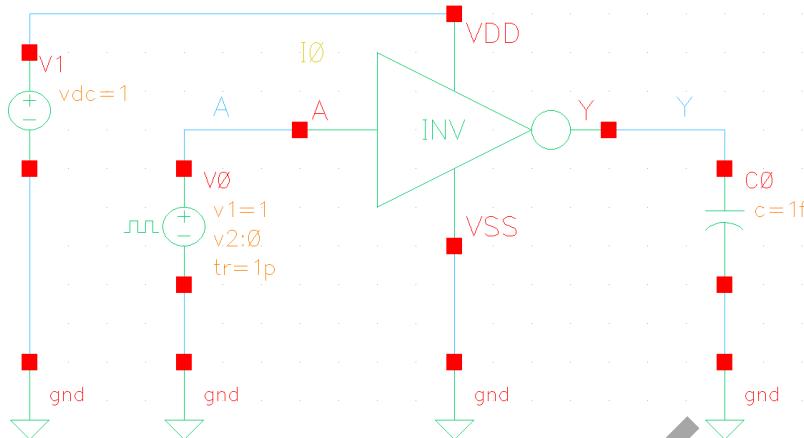


Figure 63 The circuit for running transient simulation.

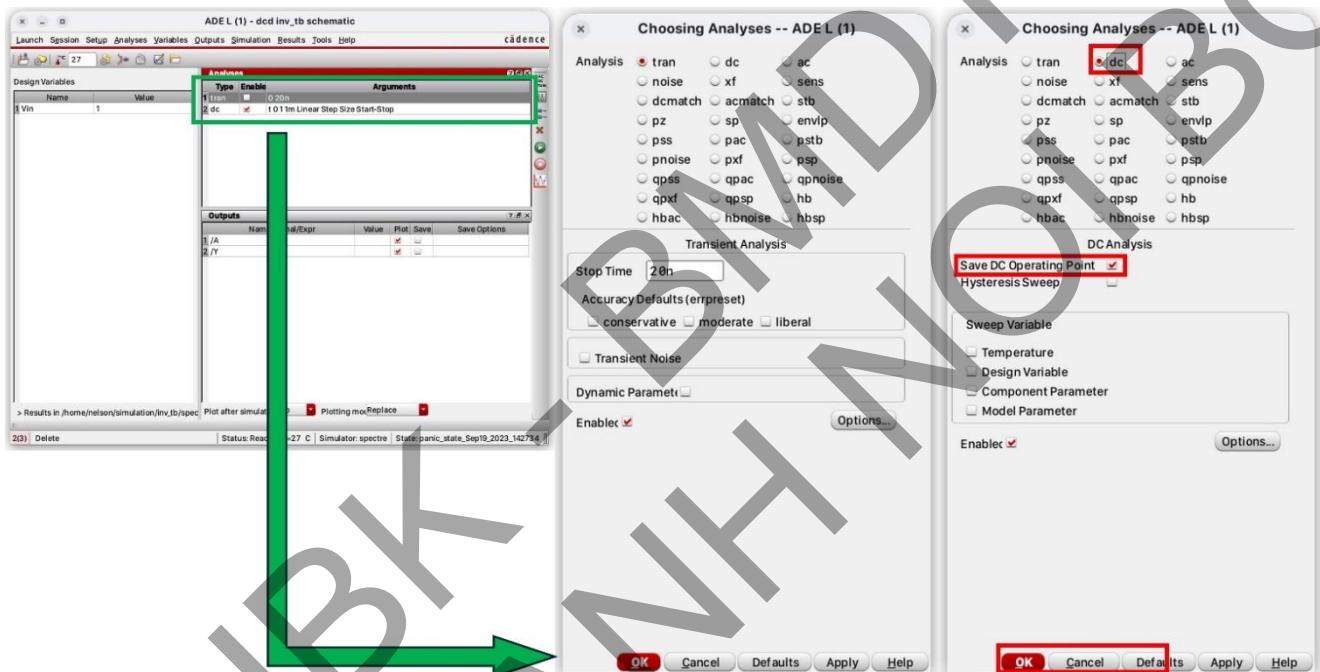


Figure 64 Analysis mode selection.

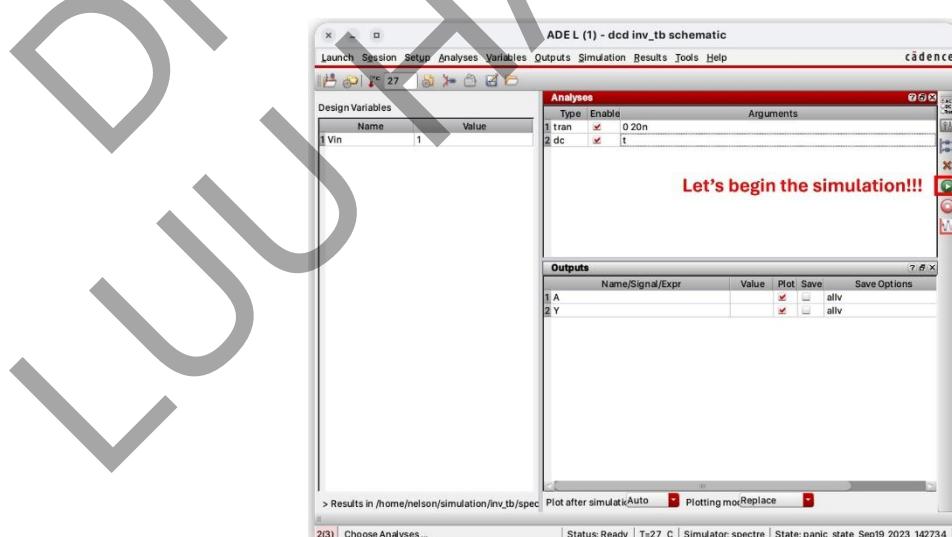


Figure 65 Setting Complete and begin the simulation.

After the simulation runs successfully, go to **Results > Annotate > DC Operating Points**. The **DC operating values** will then be displayed on the right side of each component.

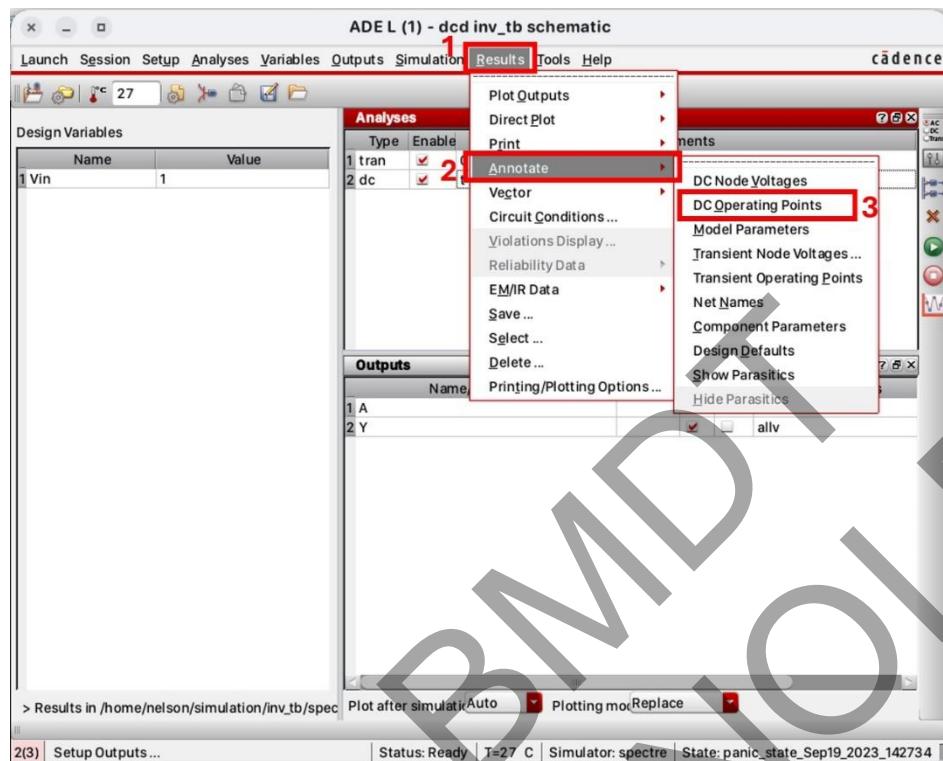


Figure 66 Show DC operating points.

Double-click the inverter's symbol, then click **OK** in the **Descend** window to move down one level. The **operating values** of each transistor will then be displayed.

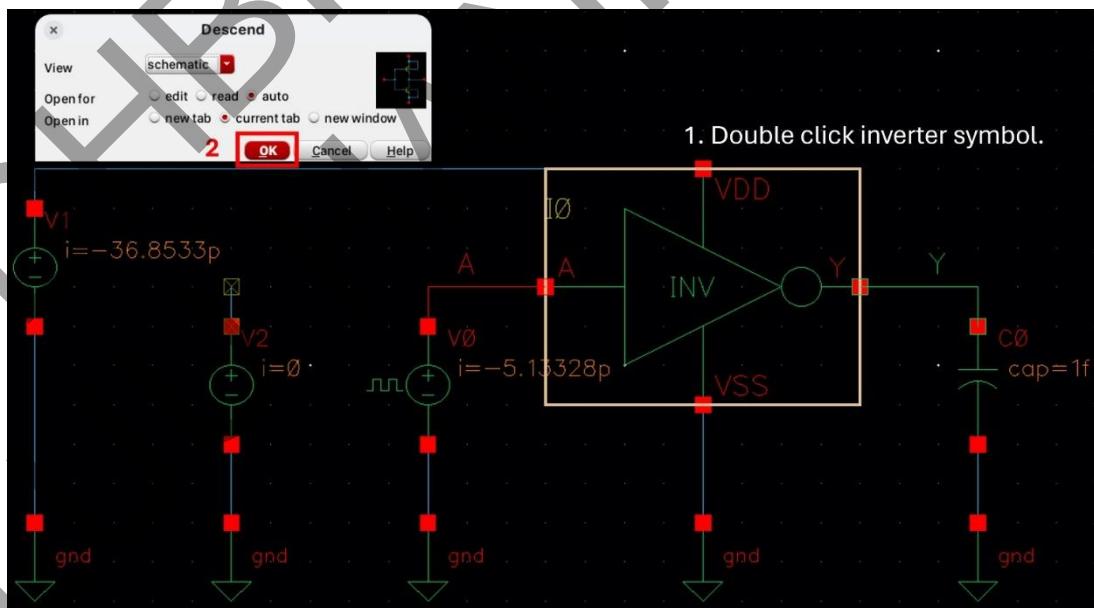


Figure 67 Go down one level.

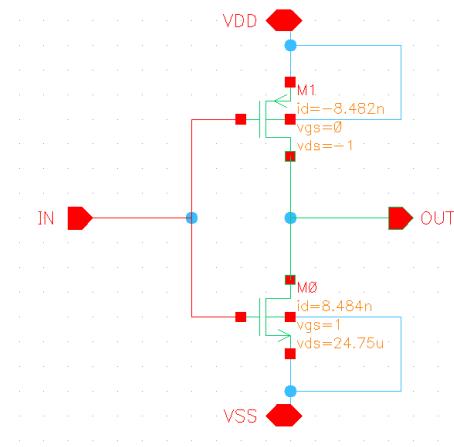


Figure 68 Inverter's schematic with DC operating points.

To observe the **operating values** of each component in more detail, students can print them in an **independent window** by following these steps in the ADE L window:

1. Navigate to **Results > Print > DC Operating Points**.
2. Click on the desired **device**.
3. The **Results Display Window** will appear, showing detailed operating values.

signal	OP("/I0/M1" "???")
beff	377.7n
betaeff	1.126m
cbb	10.81a
cbd	211.8y
cbdbo	211.8y
cbg	-10.81a
cbgbo	-10.23a
cbs	-2.831z
cbso	-2.831z
cdb	-1.269z
cdd	21.15a
cddbo	-466.9y
cdg	-21.15a
cdgbo	-5.265z
cds	7.001z
cdsbo	7.001z
egb	-18.81a
egd	-21.15a
cgbbo	955.3y
cgg	64.55a
cggbo	18.24a
cgs	-32.6a
cgsbo	-14.67z
cjd	122.7a
cjs	136.7a
cov1gb	576z
cov1gd	21.15a
cov1gs	32.59a
csb	-1.903z
csd	-700.2y
csg	-32.59a
css	32.6a
fug	545.2M
gbd	1p
gbs	1p
gds	23.85n
gm	221.1n
gmb	49.95n
gms	49.95n
gmoverid	26.27
ibe	1.001p
ibulk	680.7a
id	-8.482n
idb	1p

Figure 69 Results Display Window of PMOS.



PART 4

Objective: Performing key measurements.

Requirements: Define key inverter parameters, including **timing parameters, and power consumption**, measured either **directly from the waveform** or using the **Calculator**.

Instruction:

In this section, students will use **Virtuoso Visualization & Analysis XL** to **observe output waveforms** and perform **key measurements**.

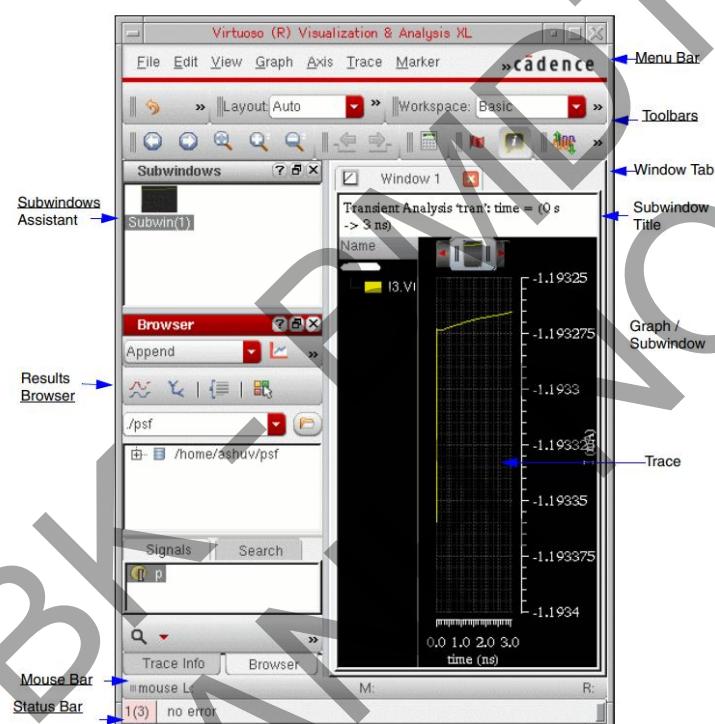
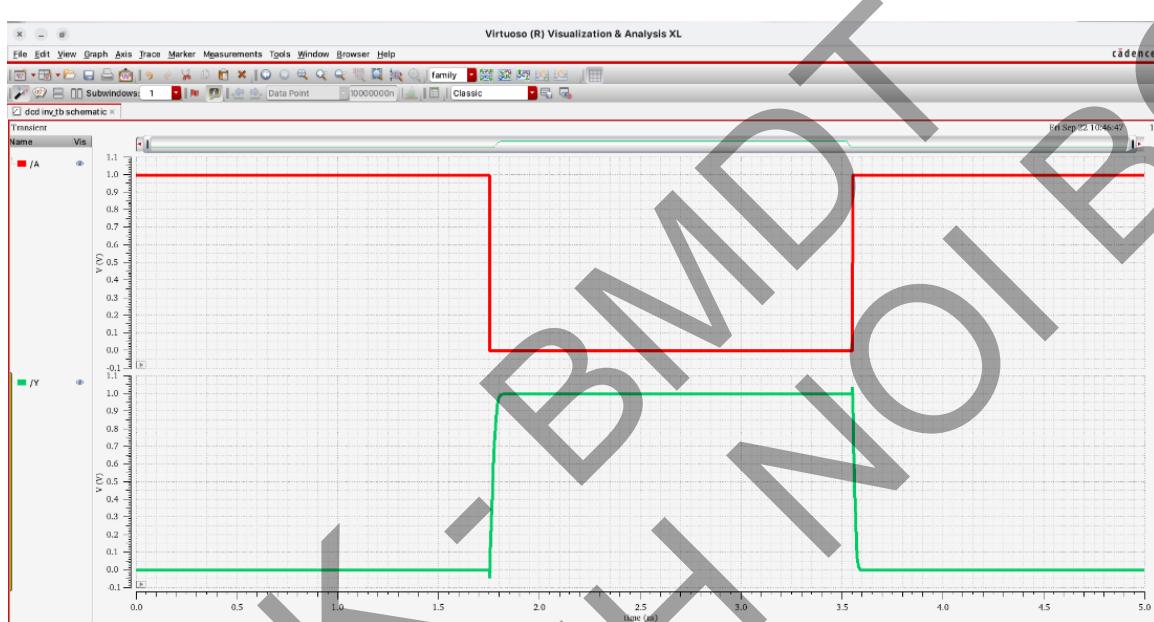
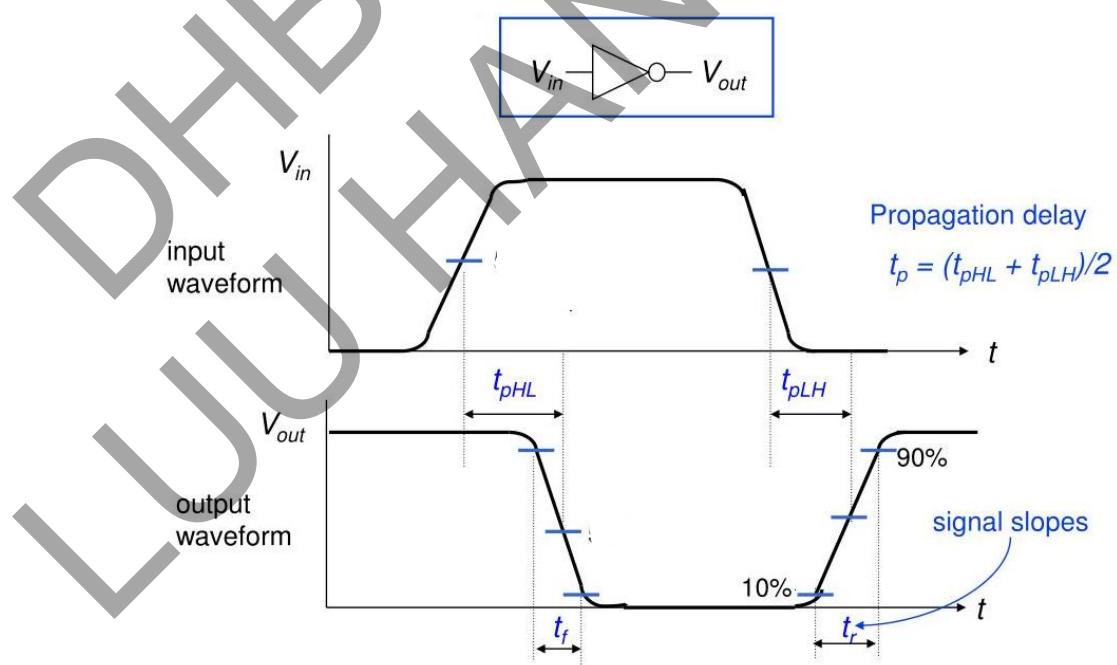


Figure 70 Virtuoso Visualization & Analysis XL window.

Students will learn how to measure the parameters listed in **Table 9**. In **Virtuoso Visualization & Analysis XL**, measurements can be performed using either **markers** or the **Calculator**. According to **Table 9**, the parameters will be measured in the following order:

1. Rising propagation delay
2. Falling propagation delay
3. Rise time
4. Fall time
5. Power consumption, including both static and dynamic power

Parameters	Value
t_{pdr} - rising propagation delay (90% – 50%)	
t_{pdf} - falling propagation delay (10% – 50%)	
t_p - average propagation delay (50% – 50%)	
t_{rise} - rising time (10% – 90%)	
t_{fall} - falling time (90% – 10%)	
Static power	
Total power consumption	
Dynamic power	

Table 9 Timing parameters and power of inverter.**Figure 71** The waveform of the inverter's transient simulation.**Figure 72** Review delay definitions.

Rise time is the duration a signal takes to transition from a specified low value to a high value. It is commonly defined as the time required for the waveform to rise from 10% to 90% of its final value.

In contrast, *fall time* is the duration a signal takes to decrease from a specified high value to a low value, typically measured from 90% to 10% of the peak value.

Rising propagation delay is the time taken for a signal to transition from low to high through a gate or net. It is measured from the 90% point of the input waveform to the 50% point of the output waveform.

Falling propagation delay is the time taken for a signal to transition from high to low through a gate or net. It is measured from the 10% point of the input waveform to the 50% point of the output waveform.

Propagation delay represents the total time required for a signal to travel through a gate or net, indicating how long it takes for a signal to change state.

★ Timing

- ❖ Measure rising propagation delay:
- Directly from waveform using **point marker** and **delta marker**:

In this method, students will be introduced to **Markers**, which attach descriptions to specific points on the graph. By default, a marker displays the **X and Y coordinates** of its intersection with the trace—if attached—or the coordinates of its placement.

To add a marker using **bindkeys**, follow these steps:

1. Click on the graph at the desired location for the marker.
2. Press one of the following keys:
 - **M** – Adds a **point marker**.
 - **H** – Adds a **horizontal marker**.
 - **V** – Adds a **vertical marker**.

The marker will be placed on the trace based on the selected **bindkey**.



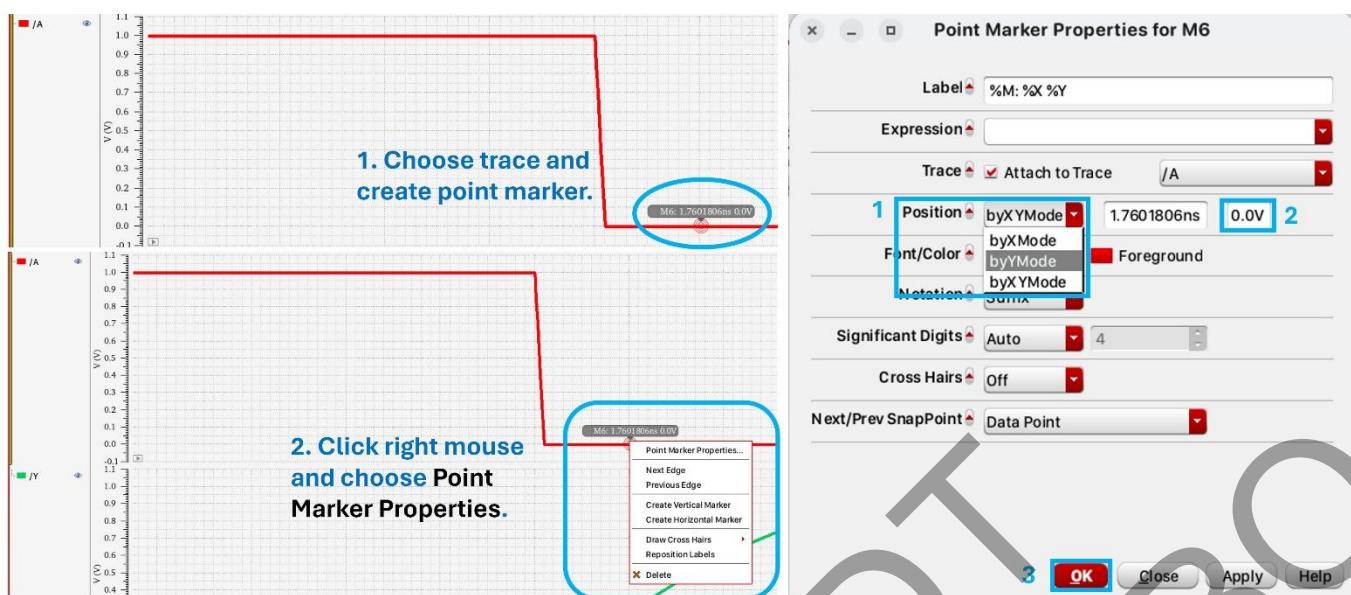


Figure 73 Changepoint marker properties.

Delta markers are used to measure the difference between two points on a graph. A delta marker connects two **point markers**, which can be on the same or different traces. To place a delta marker, students must first create or select a **point marker**. Delta markers can be moved or deleted independently, and either end of a delta marker can be adjusted, with the **X and Y coordinates updated accordingly**. These markers are useful for measuring **delays** or, when combined with **min and max functions**, for determining **peak-to-peak values**. *Creating Multiple Delta Markers on a Trace* is performed as follows:

1. **Create a point marker.** The selected marker remains active.
2. Move the mouse pointer to the trace where the second marker should be placed. Delta markers can also be added across multiple traces.
3. Press bindkey "D" to create a new point marker at the selected location. A **delta marker** is automatically generated between the new marker and the previously selected marker.

The newly created marker is of the **same type** as the original one. For example, if the first marker was a **point marker**, the new one will also be a **point marker**.



Figure 74 The result of rising propagation delay using delta marker.

- Using the *delay0* function in Calculator:

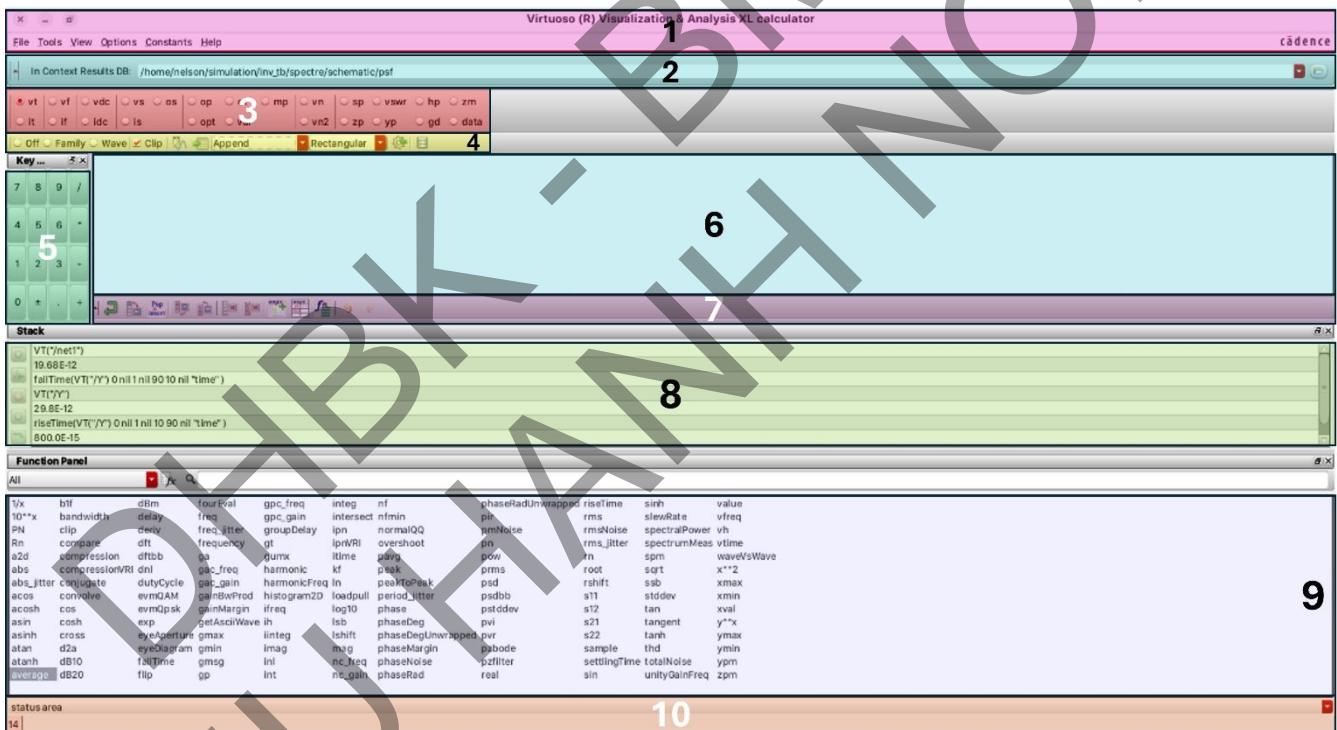


Figure 75 Calculator Graphical User Interface (Calculator GUI).

The **Calculator Graphical User Interface (GUI)** includes multiple toolbars, a menu bar, a **Buffer**, a status bar, and several dockable assistants—**Expression Editor**, **Function Panel**, **Stack**, **Keypad**, and **Memory Editor**—which students use to construct and evaluate expressions.

No.	Name	Description
1	Menu Bar	The menu bar displays the various menus that contain commands for working



		with the Calculator.
2	Result Toolbar	The Result toolbar displays the In Context Results DB field.
3	Schematic Selection Toolbar	This toolbar displays a set of function buttons, such as vt (voltage transient), that students can apply to schematic objects to build expressions.
4	Selection Toolbar	Students can use the Selection toolbar to select a signal from the Results Browser or the graph window. The selected signal is displayed in the Buffer. Students then use the tools available on the Selection toolbar to evaluate the selected signal and to define the simulation output format.
5	Keypad	The Keypad contains buttons for numbers and simple arithmetic functions. If students define any function buttons, they too are displayed on the Keypad.
6	Buffer	<p>The Buffer provides an area where students can create or edit expressions that are used to analyze the output data or signals generated after a simulation is run. Buffer is a fixed assistant and is displayed next to the Keypad. When students select a signal from the Results Browser, the signal appears in the Buffer. Students can now use this signal to build an expression.</p> <p>While creating or editing expressions in the Buffer, students can use the Keypad, Stack, Function Panel, and Expression Editor assistants or use the keyboard. After students have created or evaluated the expressions, students can save the expressions by moving them from the Buffer to the Stack, Expression Editor, or Memory Editor.</p>
7	Buffer and Stack Toolbar	This toolbar contains the buttons that help to perform operations on the Buffer and Stack contents. By default this toolbar is displayed below the Buffer and next to the keypad.
8	Stack	Stack is an area where students can store the expressions created in the Buffer.
9	Function Panel	This assistant displays a list of functions. The functions displayed are determined by the function category selected in the Function Panel drop-down list box.
10	Status Bar	

Table 10 Describe components of Calculator GUI.

To select **Signal 1** and **Signal 2**, follow the instructions in the next two figures. Click on the



desired **net**, and its name will appear in the **Calculator Buffer**. Press **Ctrl+C** to copy the name and **Ctrl+V** to paste it into **Signal 1**. Additionally, if a waveform already exists, students can use the **wave** option instead.

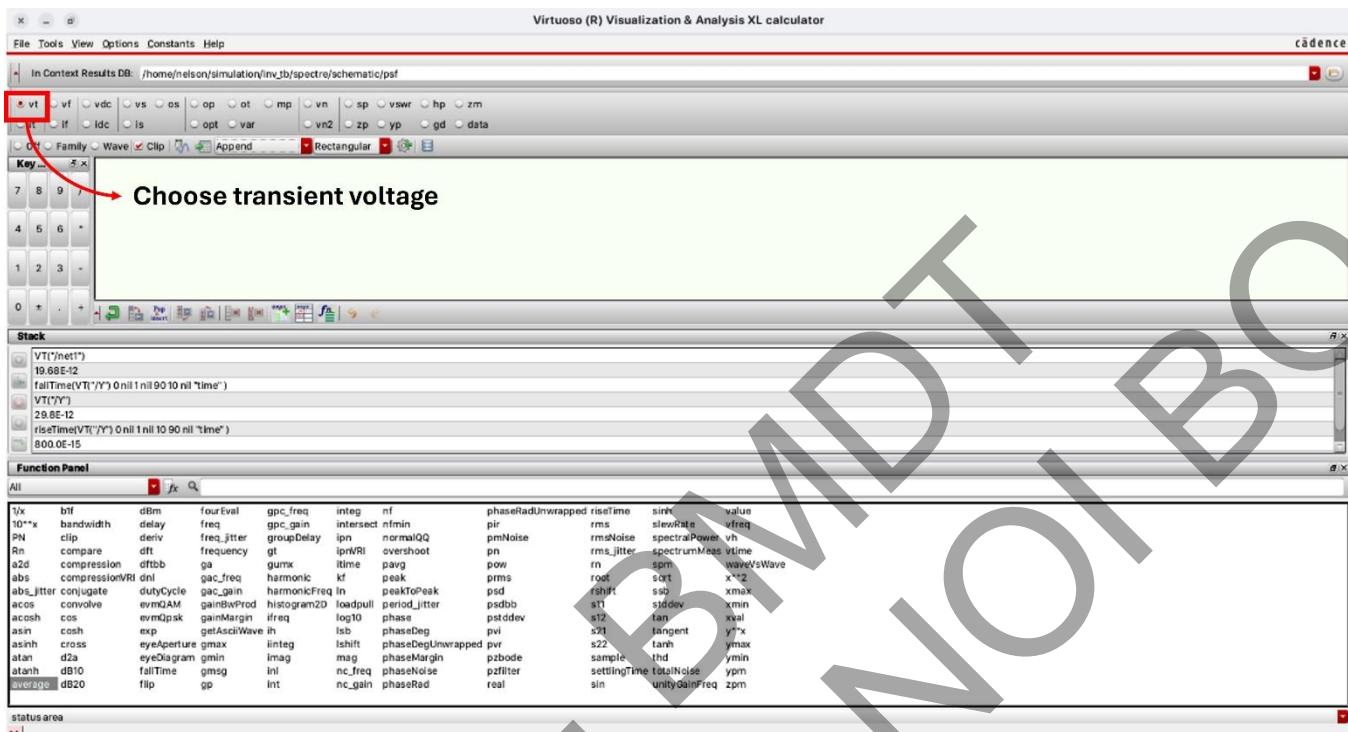


Figure 76 Define the net name of transient voltage in the schematic.

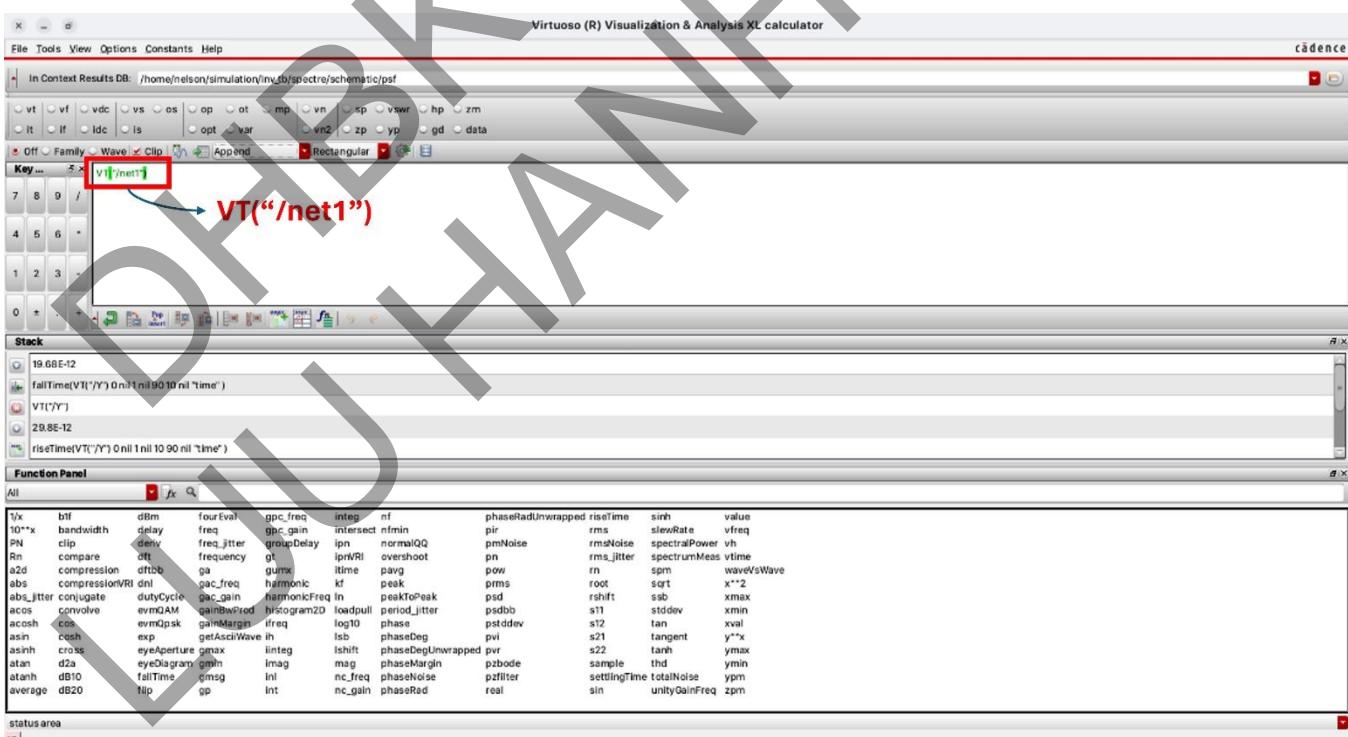


Figure 77 The net connecting between the power supply and VDD of the inverter is defined.

Using the instructions above, students define signal 1 and signal 2, then set values shown in Figure

below:

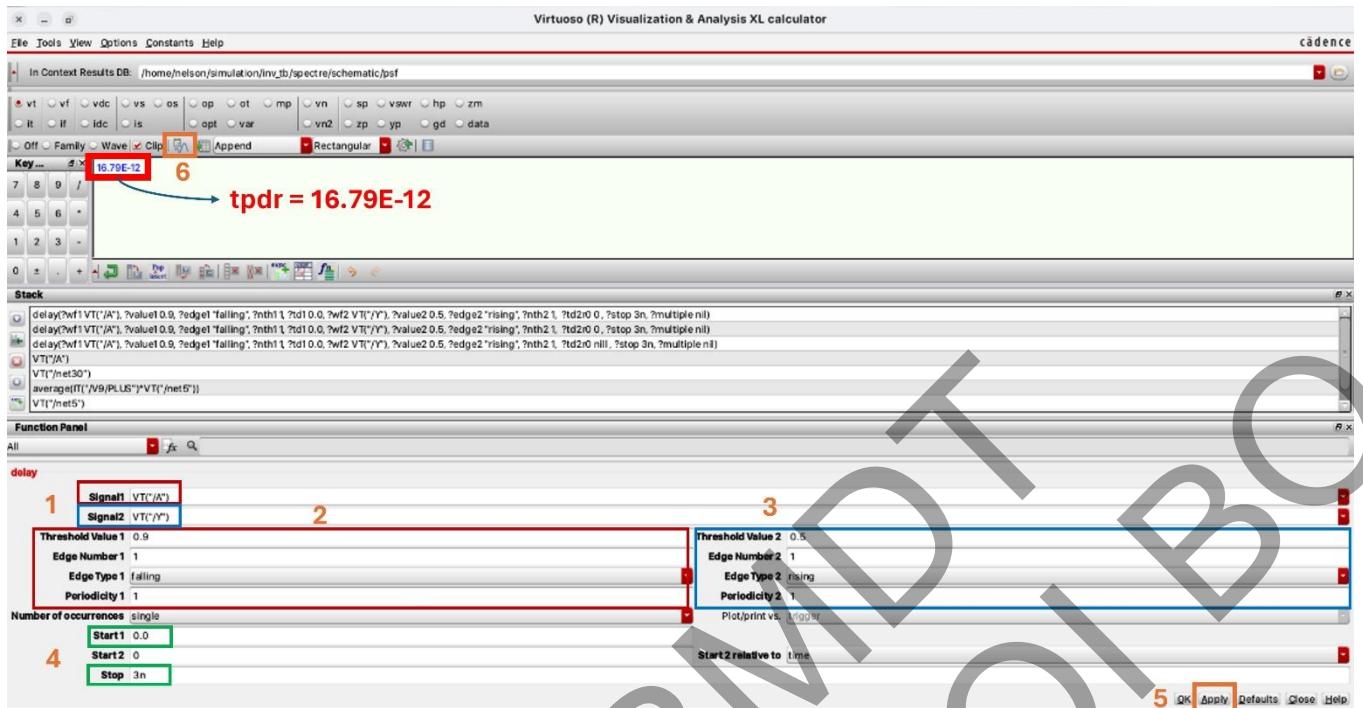


Figure 78 The result of rising propagation delay using Calculator.

Rising propagation delay: $t_{pdr} = 16.79e - 12$

❖ Measure falling propagation delay:

- Directly from waveform using point marker and delta marker:

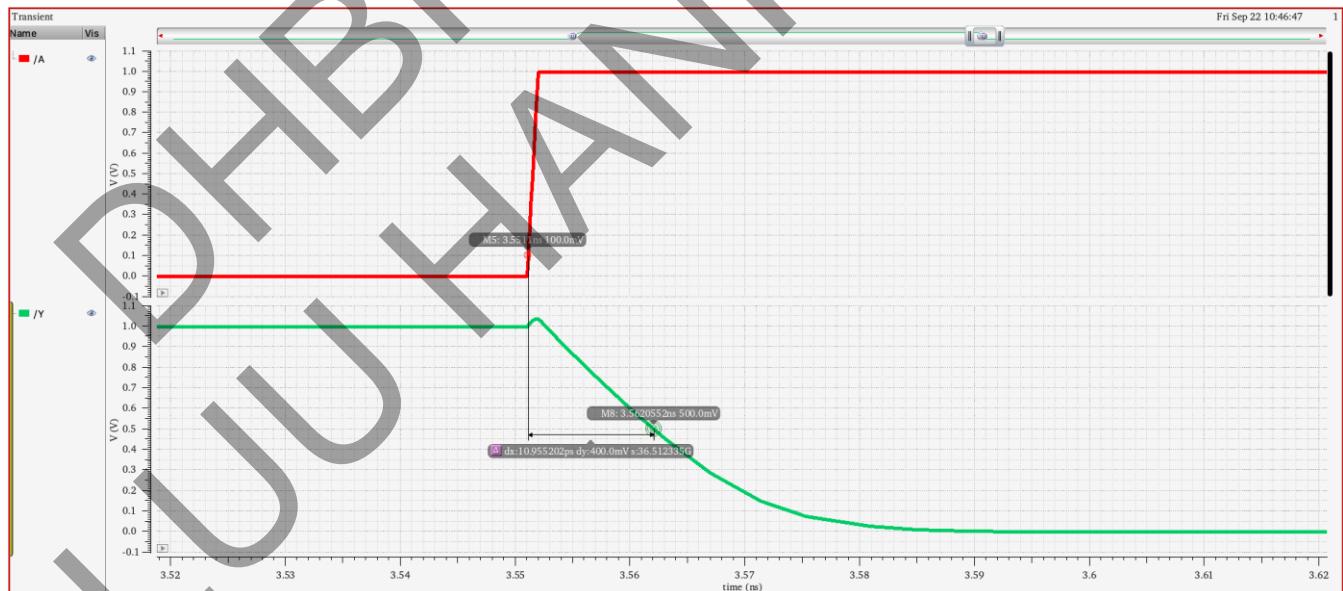


Figure 79 The result of falling propagation delay using delta marker.

- Using the **delay()** function in Calculator:

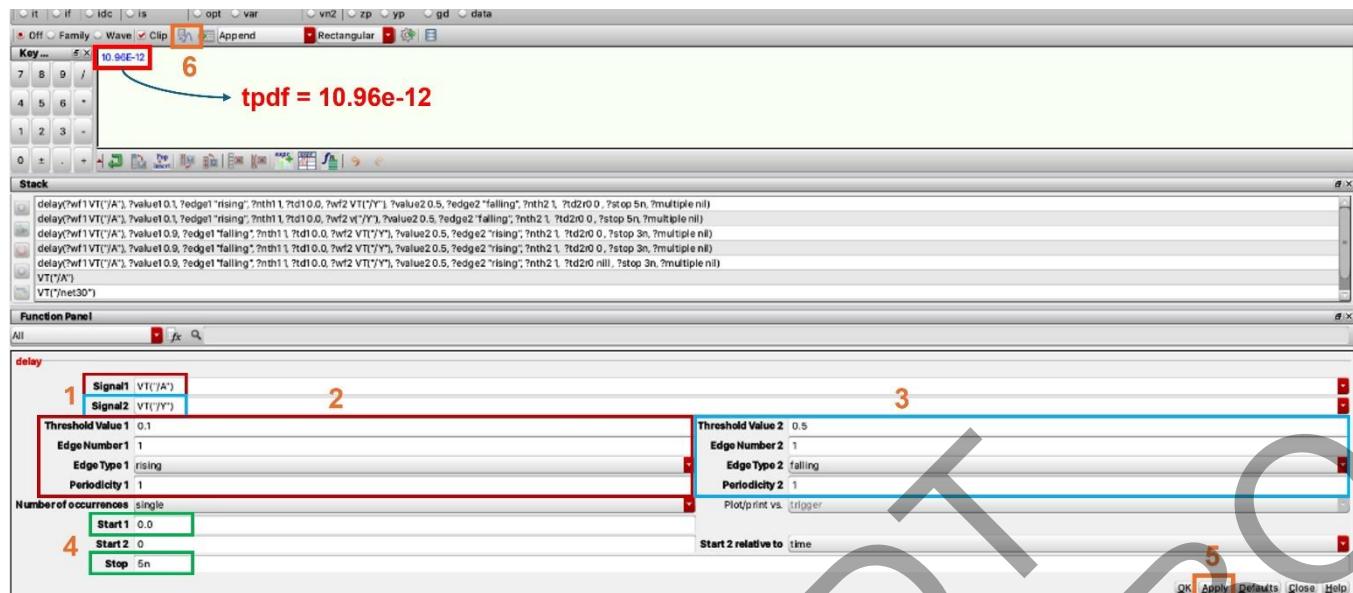


Figure 80 The result of falling propagation delay using Calculator.

$$t_{pdf} = 10.9552\text{ns}$$

❖ **Measure rising time:**

- Directly from waveform using point marker and delta marker:

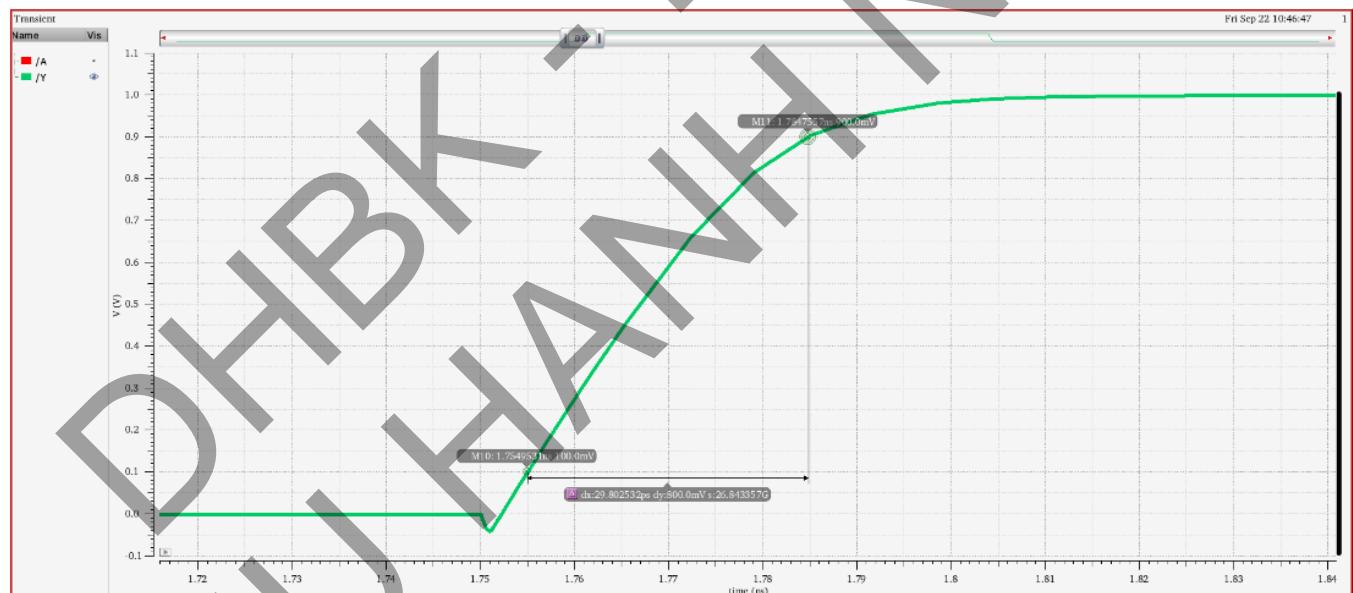


Figure 81 The result of rise time using delta marker.

- Using the `riseTime()` function in Calculator:

Perform the instructions shown in the figure below to measure the rise time of the inverter:

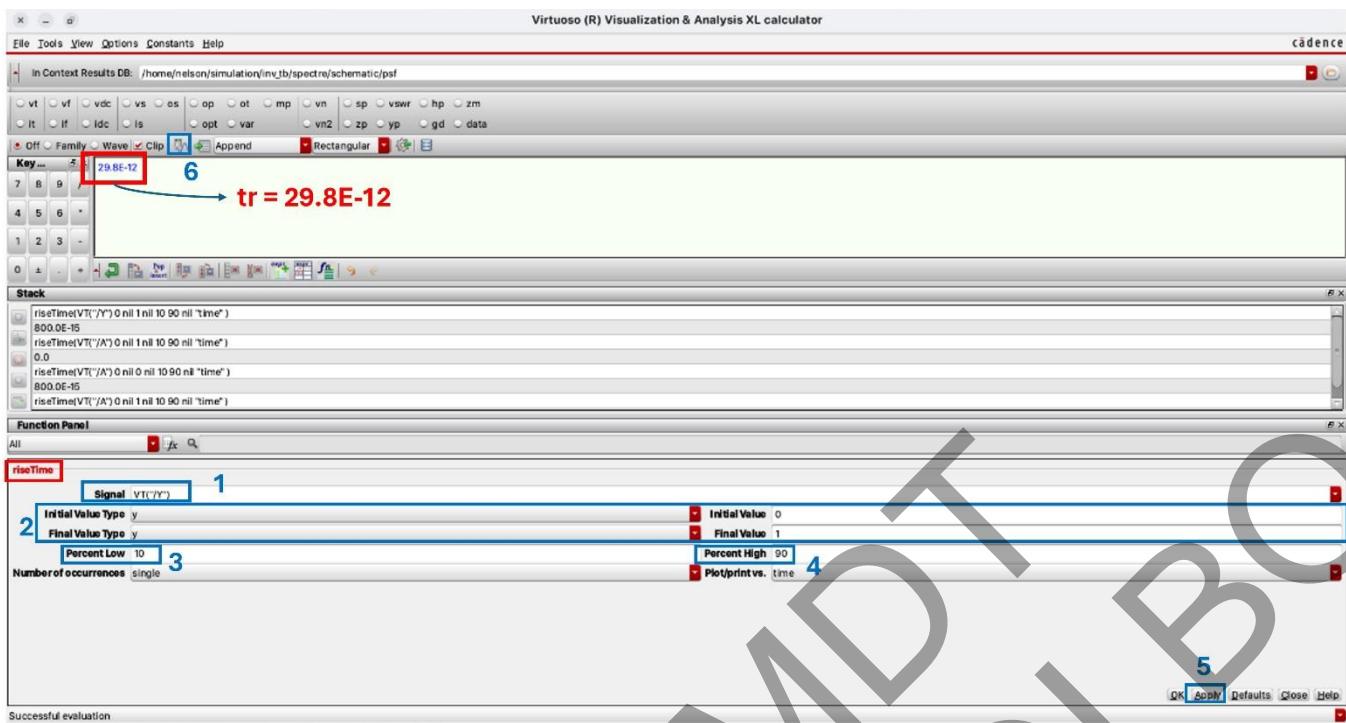


Figure 82 The result of rise time using Calculator.

Rise time: $t_r = 29.802532\text{ns}$

❖ Measure fall time:

- Directly from waveform using point marker and delta marker:

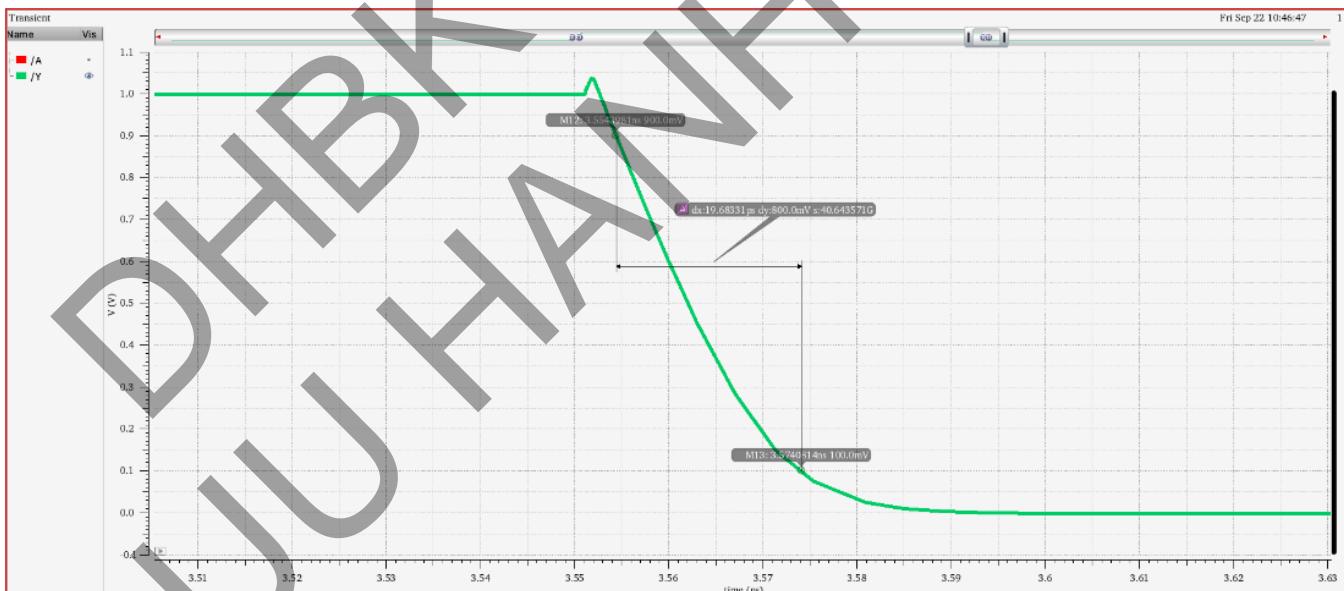


Figure 83 The result of fall time using delta marker.

- Using the **fallTime()** function in Calculator:

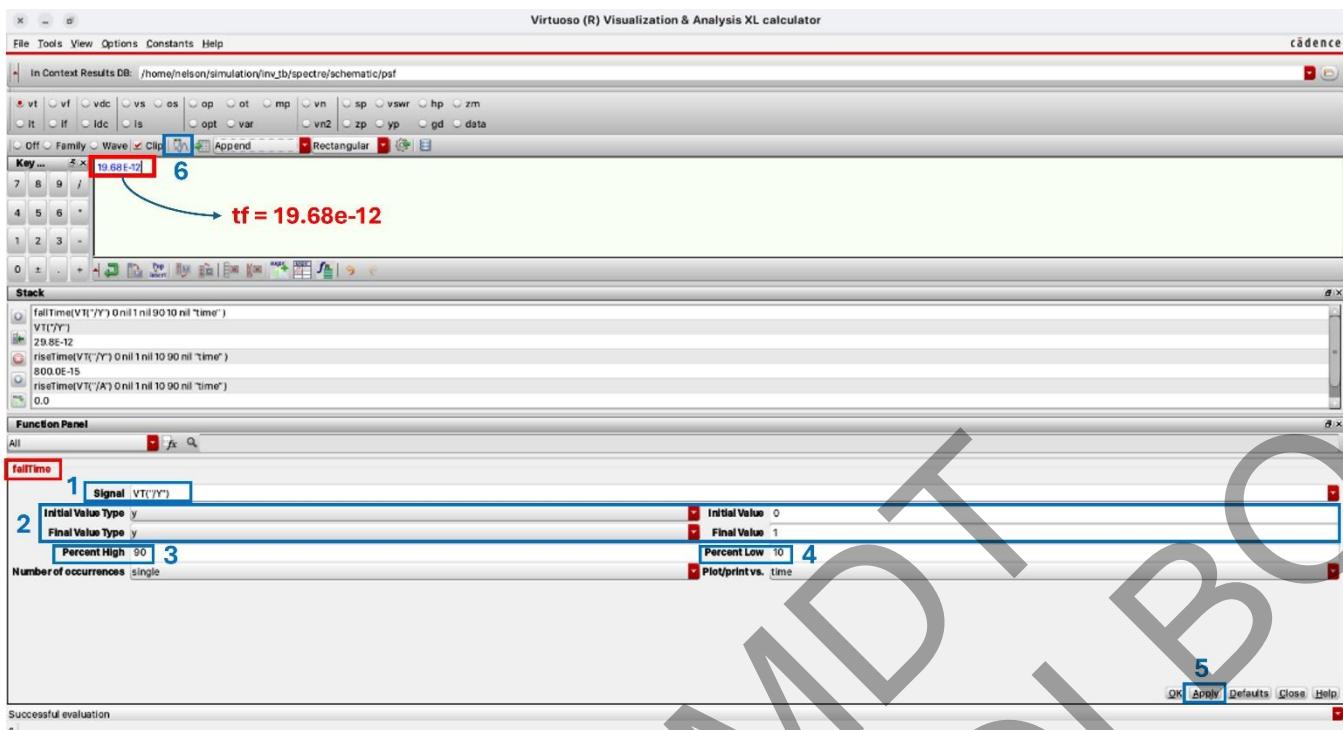


Figure 84 The result of fall time using Calculator.

Fall time: $t_f = 19.68\text{ns}$

★ Power consumption

$$\begin{aligned} P_{dynamic} &= C_L V_{DD}^2 f \\ P_{static} &= I_{static} V_{DD} \\ P &= P_{dynamic} + P_{static} \end{aligned}$$

The **total power dissipation** of a circuit consists of both **dynamic** and **static** components, which can be difficult to separate in simulations. Consider the **CMOS inverter** shown below. **Dynamic power** arises from **switching currents**, which are needed to charge and discharge output loads, as well as **short-circuit currents** that briefly flow between the **PMOS** and **NMOS** transistors during input transitions. In contrast, **static power** results from **leakage sources** within the transistors, including **subthreshold conduction** between the source and drain, as well as **reverse-bias pn-junction leakage** between the source/drain and substrate. These leakage mechanisms are illustrated in the transistor diagram below.

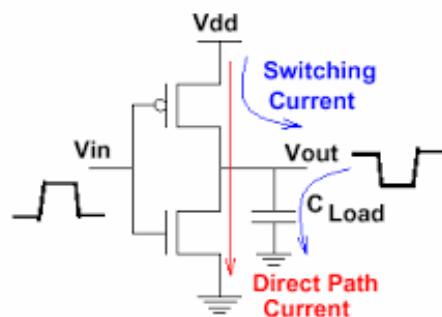


Figure 85 Dynamic power consumption.

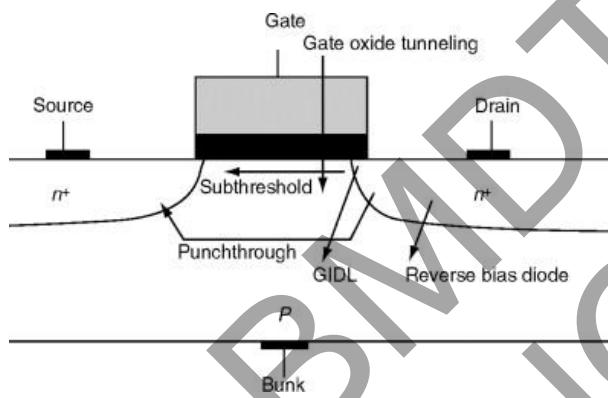


Figure 86 CMOS static leakage sources.

To measure static power dissipation, a DC input signal is applied to ensure no switching occurs.

In digital circuits, this means setting the input to either **high (VDD)** or **low (ground)**, which typically turns off one side of the circuit and prevents static **short-circuit current** through the transistors. For **analog circuits**, the input should be set to an appropriate **DC operating point**, usually between **ground** and **VDD**. Since many transistors remain turned on in their static state, analog circuits tend to **consume more static power** than digital ones.

Example: Consider a **CMOS inverter** with the following parameters: **PMOS: 1.5μm/0.6μm, NMOS: 1.5μm/0.6μm, Load capacitance: 5pF**

Using a **simple measurement method**, we can determine static power dissipation:

- $V_{in} = 3.3V \rightarrow P_{static} = 10.94 \text{ pW}$
- $V_{in} = 0V \rightarrow P_{static} = 10.93 \text{ pW}$

As expected, the power consumption is nearly the same in both cases. The slight difference is due to variations in **leakage currents** of PMOS and NMOS transistors when biased at different voltages. These differences could be more pronounced if one transistor were significantly larger than the other.

To measure total power dissipation, the circuit must **switch** between logic states, causing the output node to **charge and discharge**. In digital circuits, this requires applying a **pulse input signal**.

Example: For the same CMOS inverter, applying a **pulse input** (**pulse width = 5μs, period = 10μs**) results in **P_total = 5.49 μW**

If the **load capacitor is removed**, the **switching current** is reduced, leaving only the current required to charge/discharge the **parasitic capacitance** at the output. As a result, the measured power will be much closer to the **static power dissipation** value.

To estimate dynamic power dissipation, subtract **static power** from **total power** to estimate the contribution of dynamic sources:

$$P_{\text{total}} = P_{\text{dynamic}} + P_{\text{static}} \rightarrow P_{\text{dynamic}} = \dots$$

❖ Static power measurement:

There are two methods to define the static power of an inverter: based on NMOS (when $V_{in} = 1$) or based on PMOS (when $V_{in} = 0$). I will connect VDC at the input of the inverter with variable value - V_{in} .

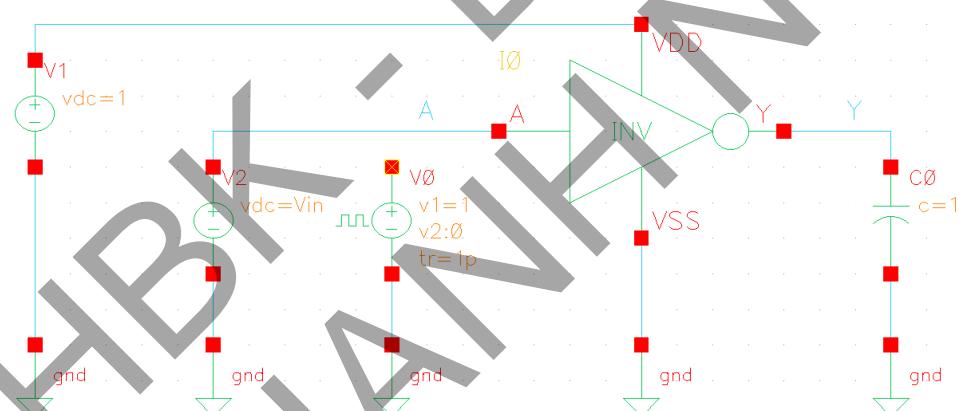


Figure 87 Schematic to measure static power.

- **Static power based on NMOS – when $V_{in} = 1$:**

First, perform **Outputs > Save All**, setup **Save Options** window as instructions:

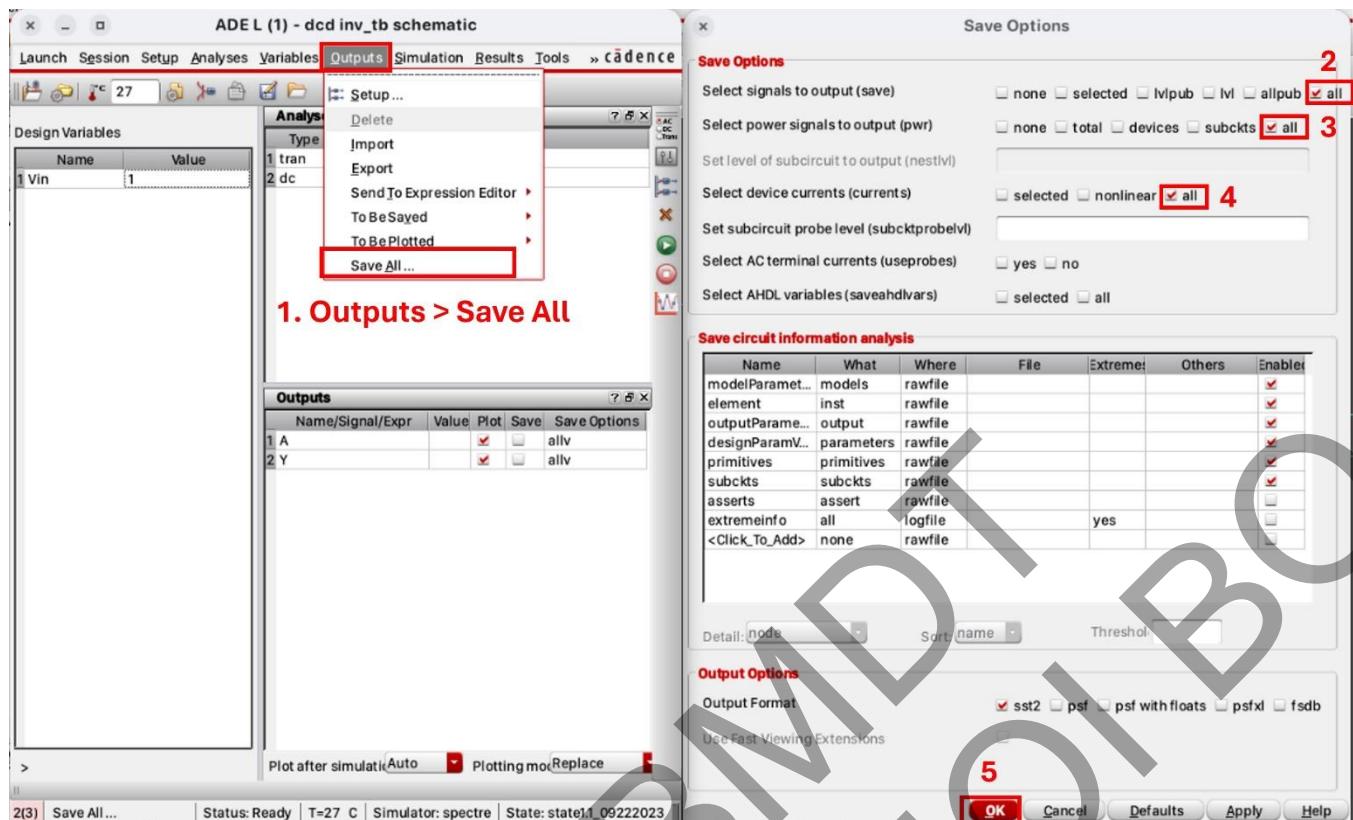


Figure 88 Save signals/power signal to output and device currents.

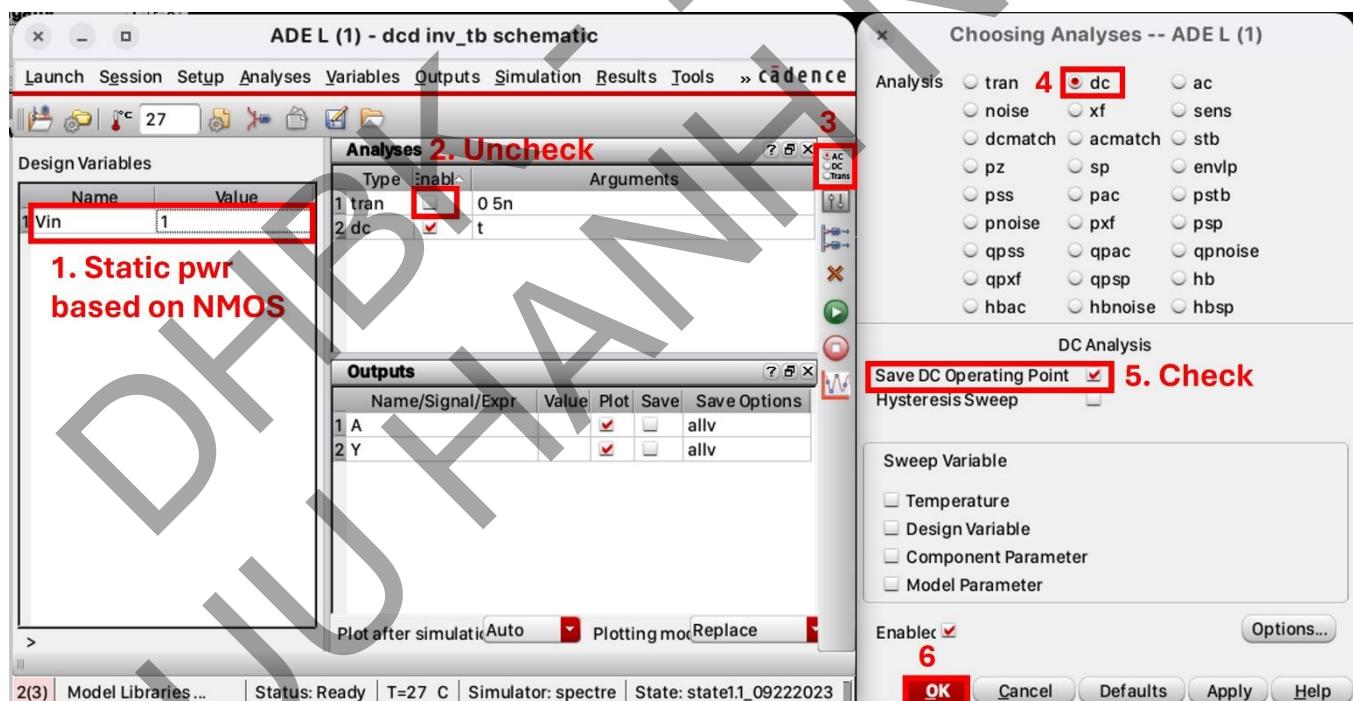


Figure 89 Analysis mode selection.

After the simulation runs successfully, choose **Tools > Results Browser > dcOpinfo > choose the name of power supply**.

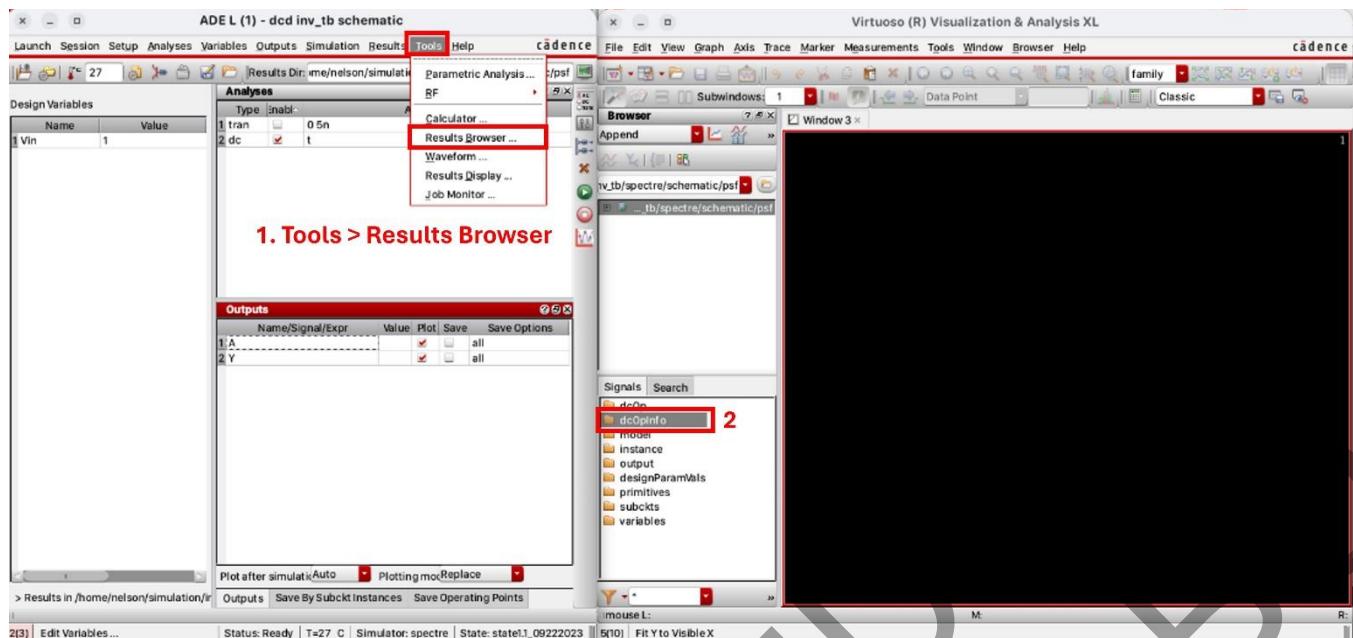


Figure 90 Open dcOpinfo.

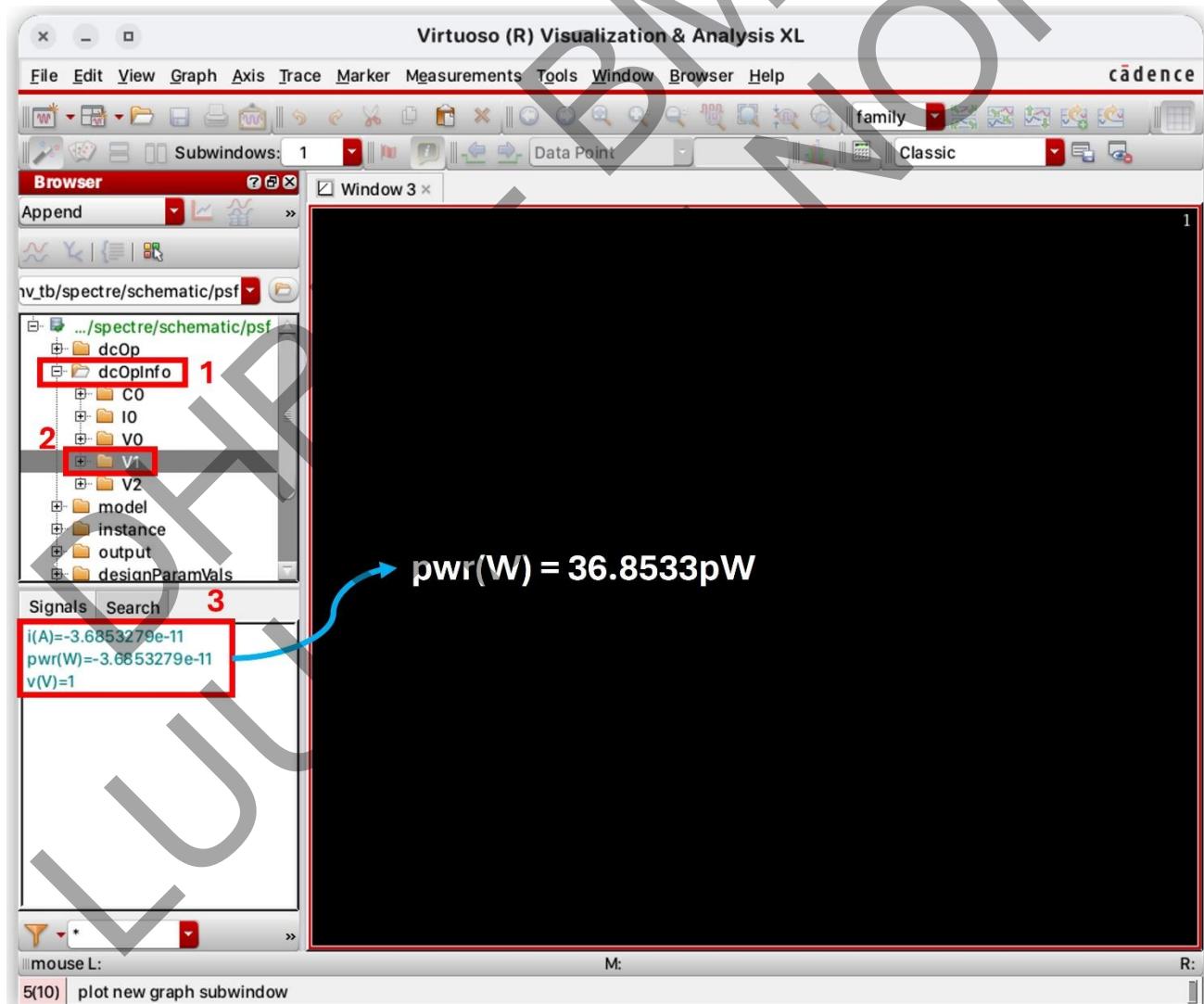


Figure 91 The result of static power based on NMOS.



Due to static power measurement result: $P_{static(NMOS)} = 36.8533\text{pW}$

Due to static power expression:

$$P_{static} = I_{static}V_{DD} = 3.68533 \times 10^{-11} \times 1 = 36.8533\text{pW}$$

- Static power based on PMOS – when $V_{in} = 0$:

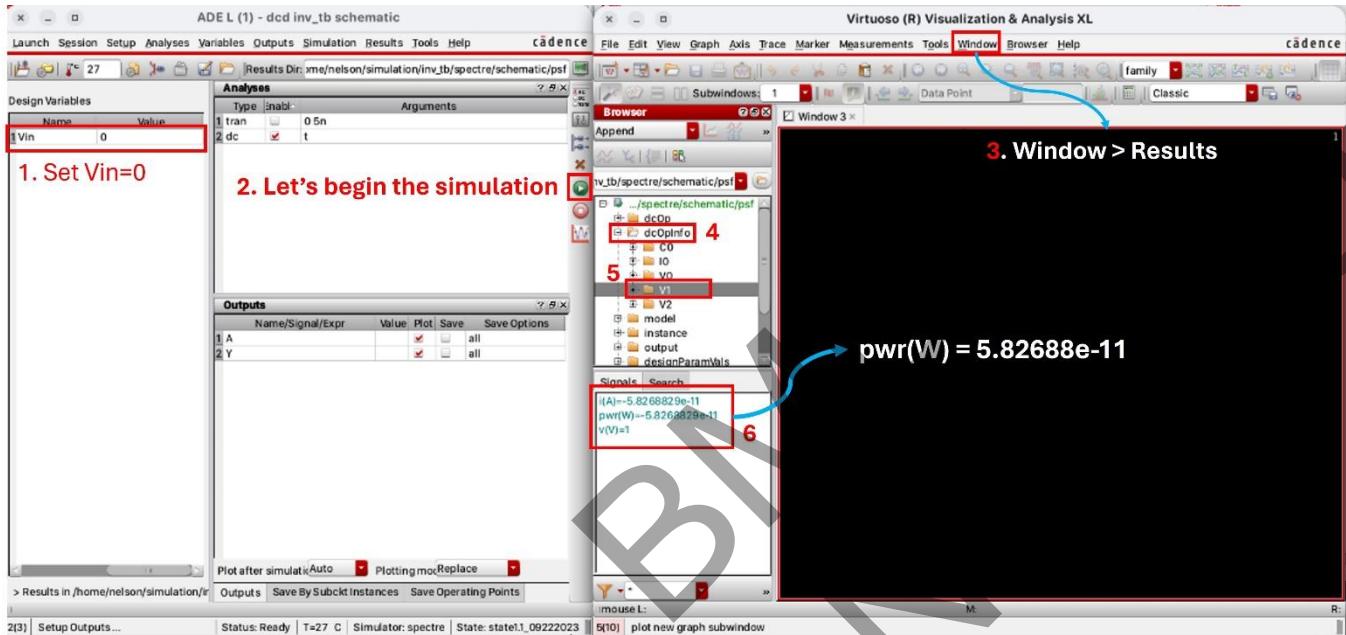


Figure 92 The result of static power based on PMOS.

$$P_{static(PMOS)} = 58.2688\text{pW}$$

Verify the result:

$$P_{static} = I_{static}V_{DD} = 5.82688 \times 10^{-11} \times 1 = 58.2688\text{pW}$$

- ❖ Total power measurement:

Before starting measurement, students need to edit the value of input voltage and load capacitance.

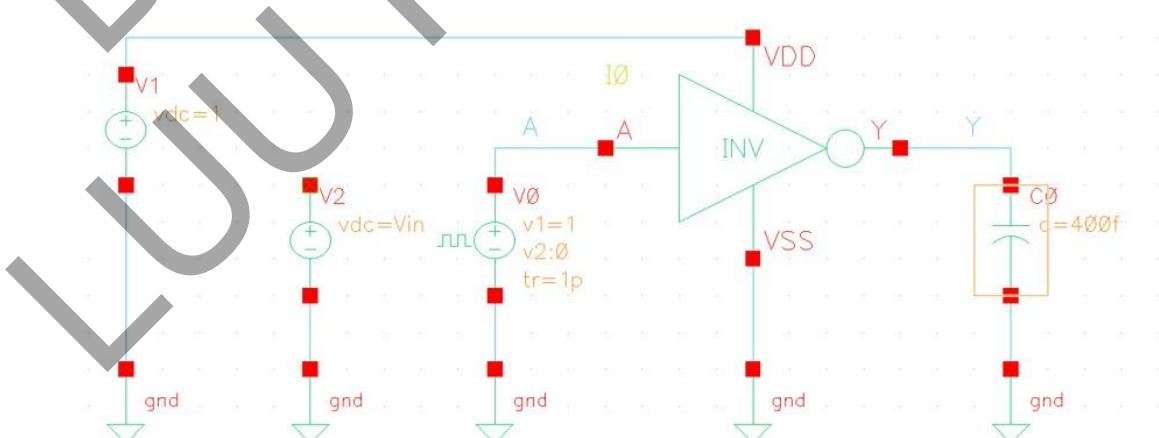


Figure 93 The schematic for total power measurement.

According to the table below, the set value required for vdc, cap, and vpulse.

Parameters	Value	Component
V_{DD}	1V	vdc
C_{load}	400 fF	cap
Voltage 1	0V	
Voltage 2	1V	
Rise time	400 ps	
Fall time	400 ps	vpulse
Delay	0 ns	
Pulse width	10 ns	
Period	20 ns	

Table 11 The value for each component.

To determine total power consumption, students must apply a pulse input signal, requiring a transient simulation instead of a static power measurement. First, save all signals to output, power signals to output, and device current. Then, proceed to analysis mode.

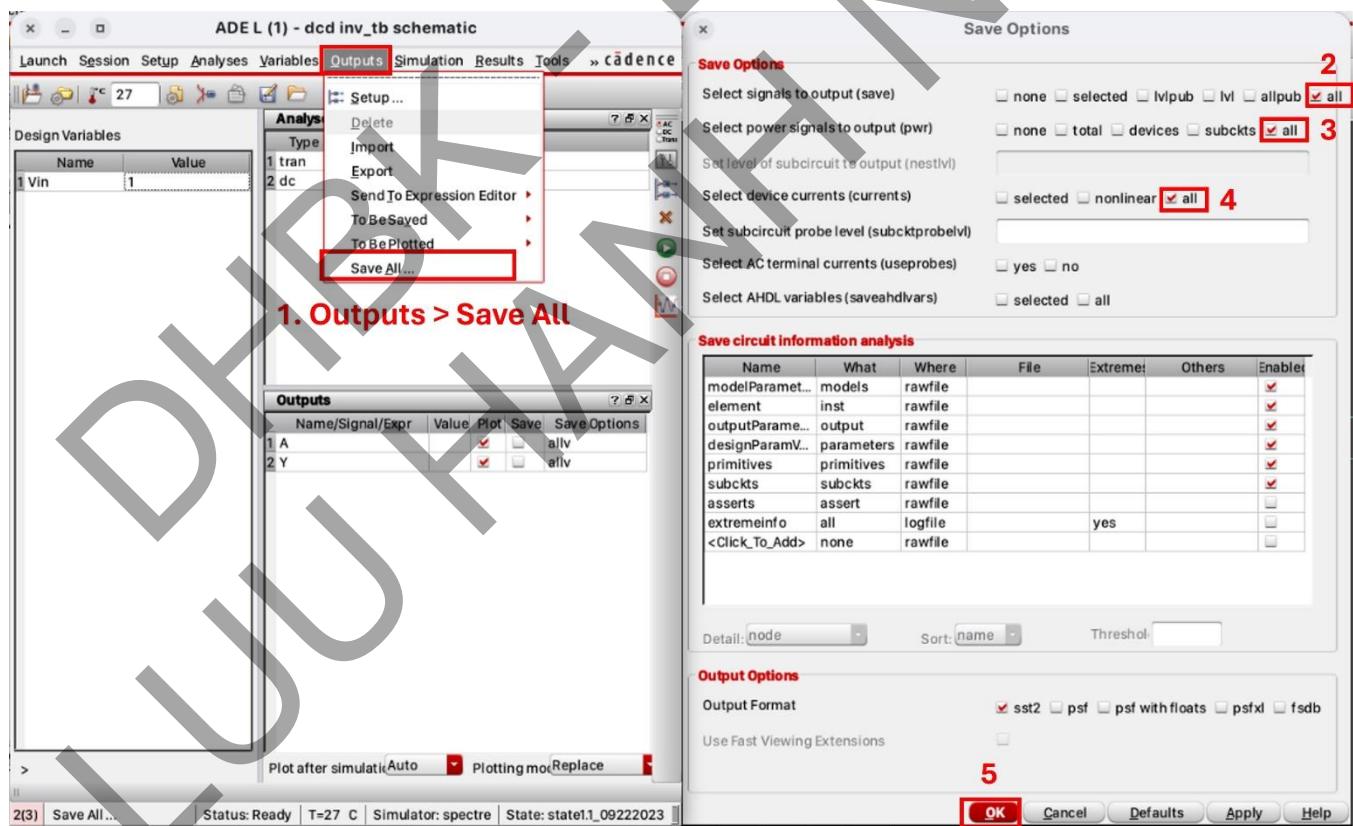


Figure 94 Save signals/power signal to output and device currents.

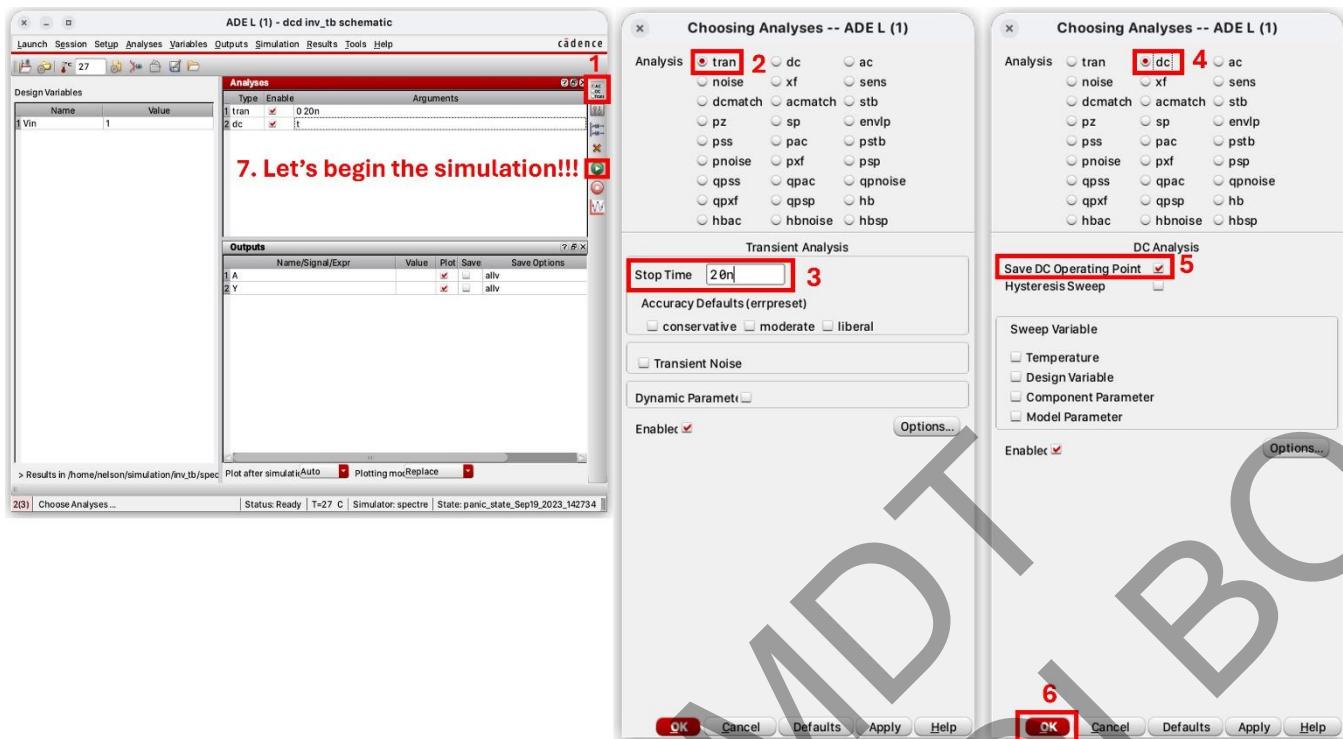


Figure 95 Analysis mode selection for total power measurement.

After the simulation runs successfully, open Calculator to perform our work (**Tools > Calculator**). Define nets used to calculate total power.

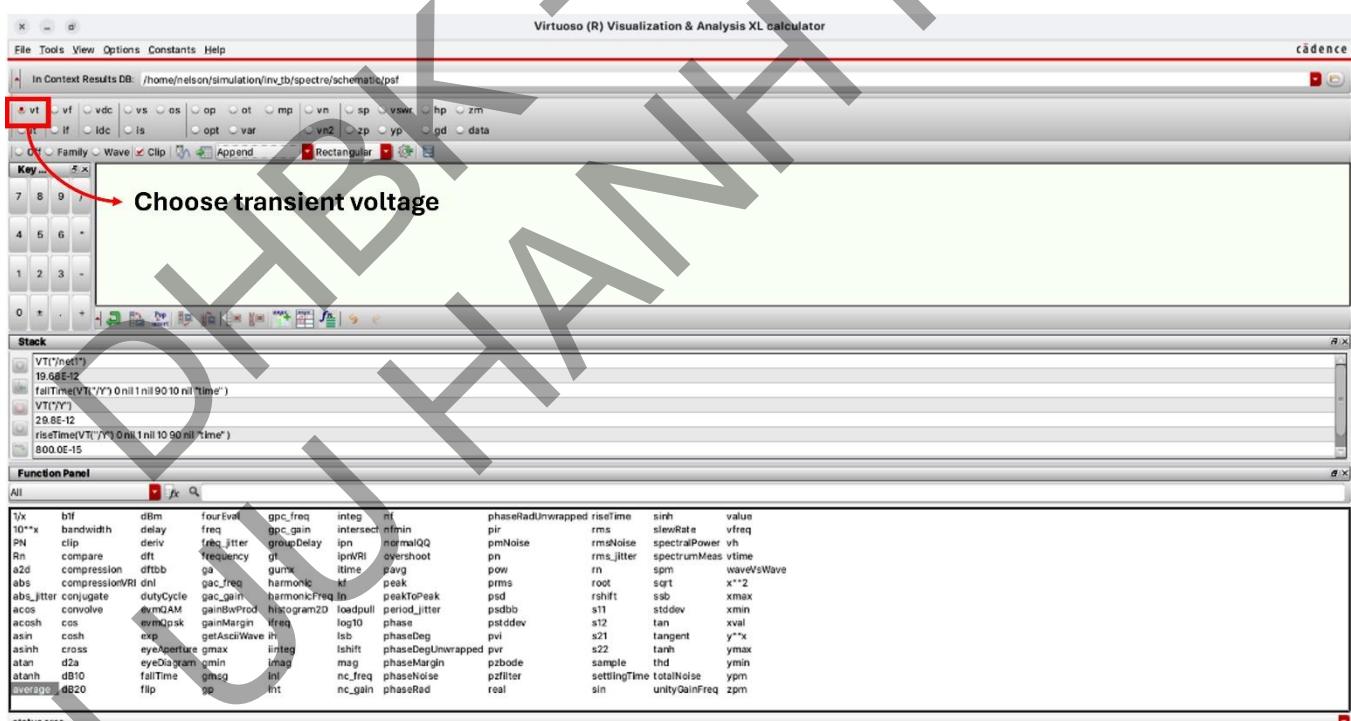


Figure 96 Define the net name of transient voltage in the schematic.

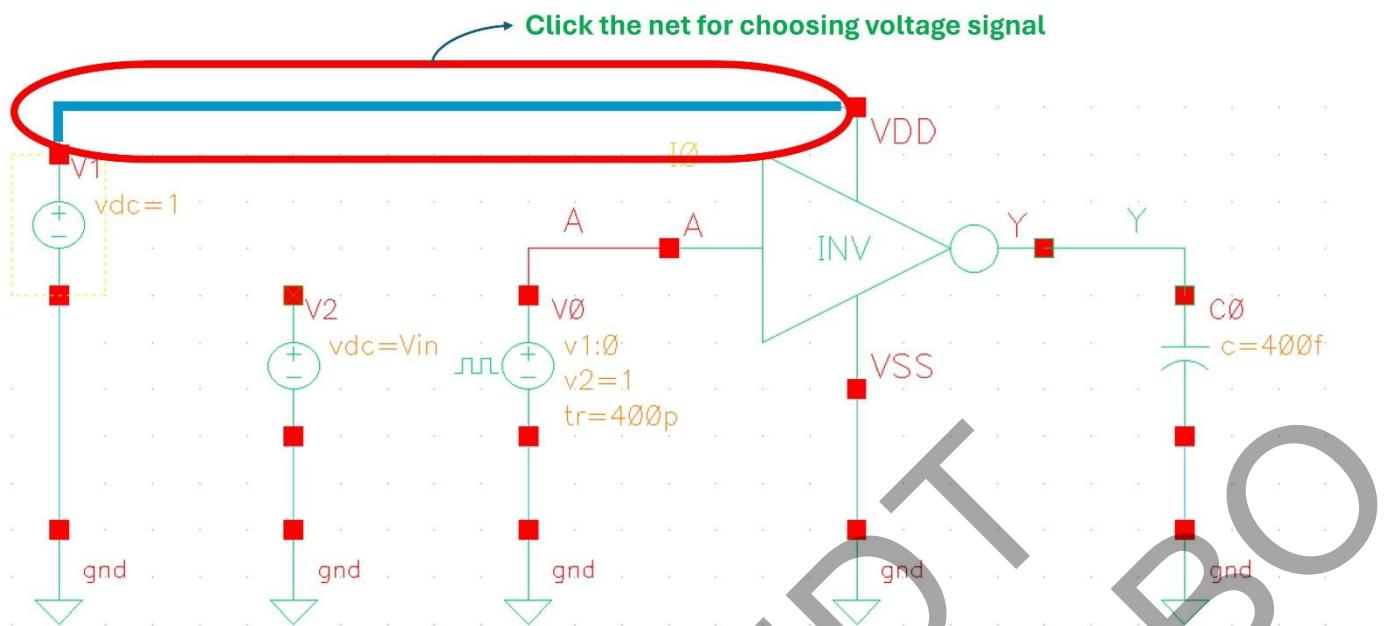


Figure 97 Choose the net students want to define.

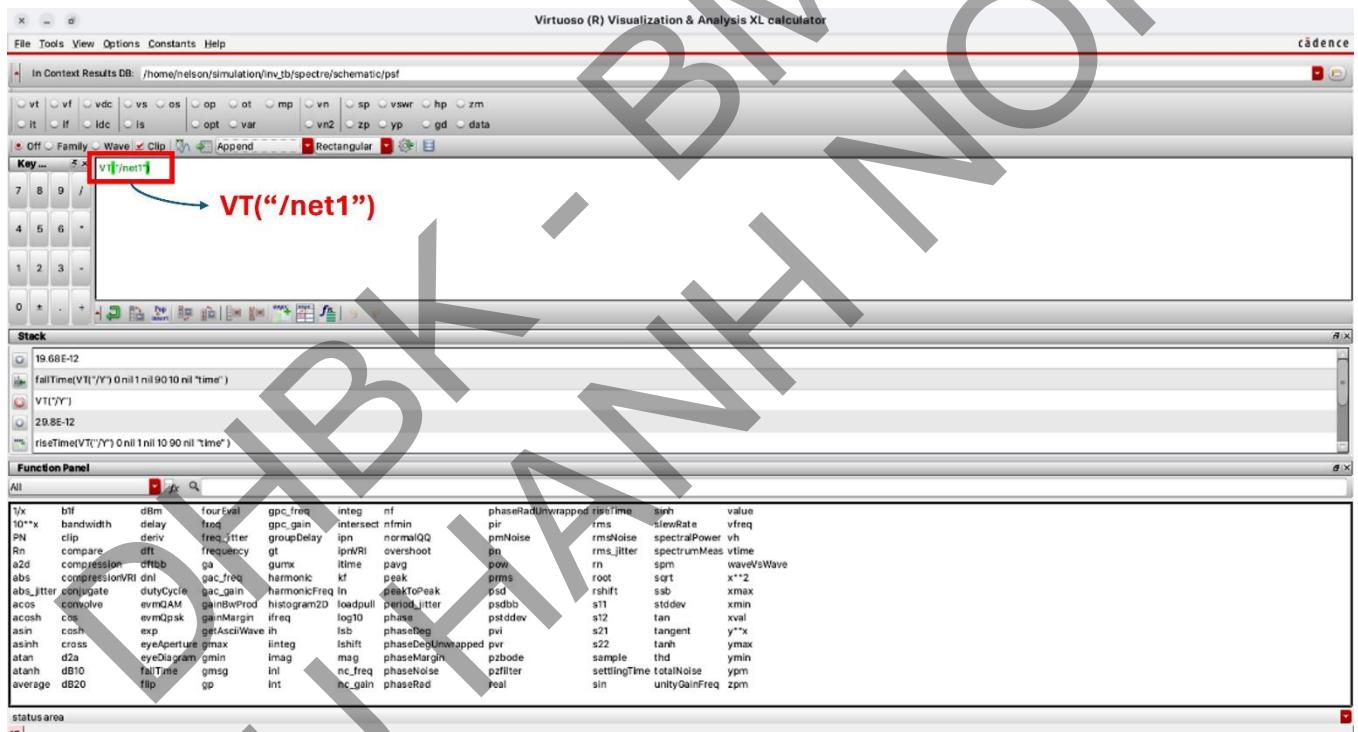


Figure 98 The name of the transient voltage is printed.

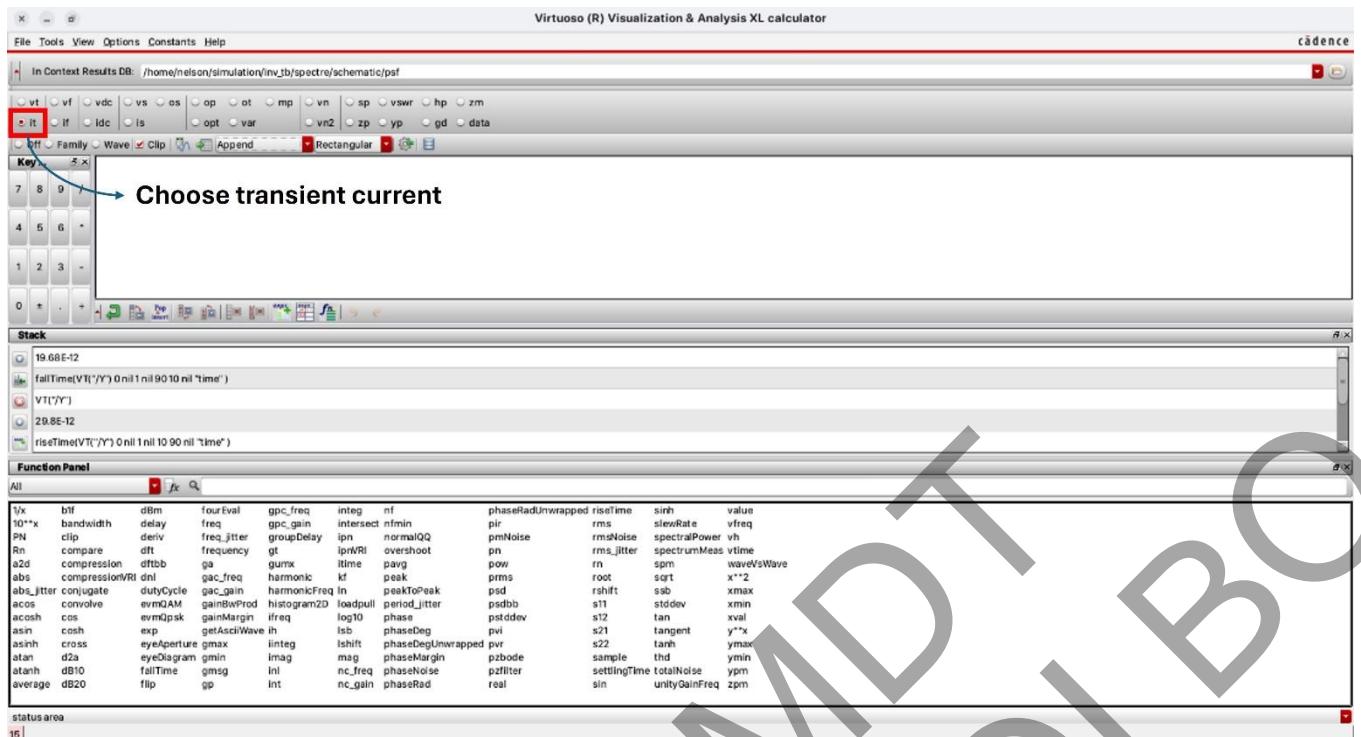


Figure 99 Define the net name of the transient current in the schematic.

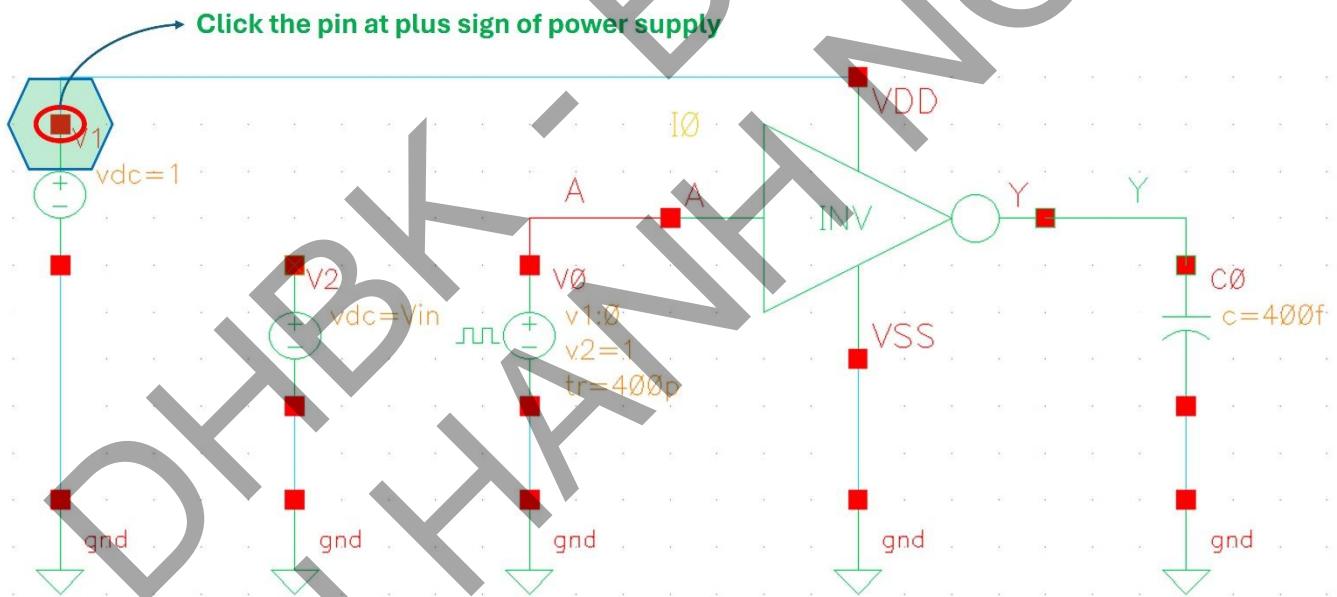


Figure 100 Choose terminal to define transient current.

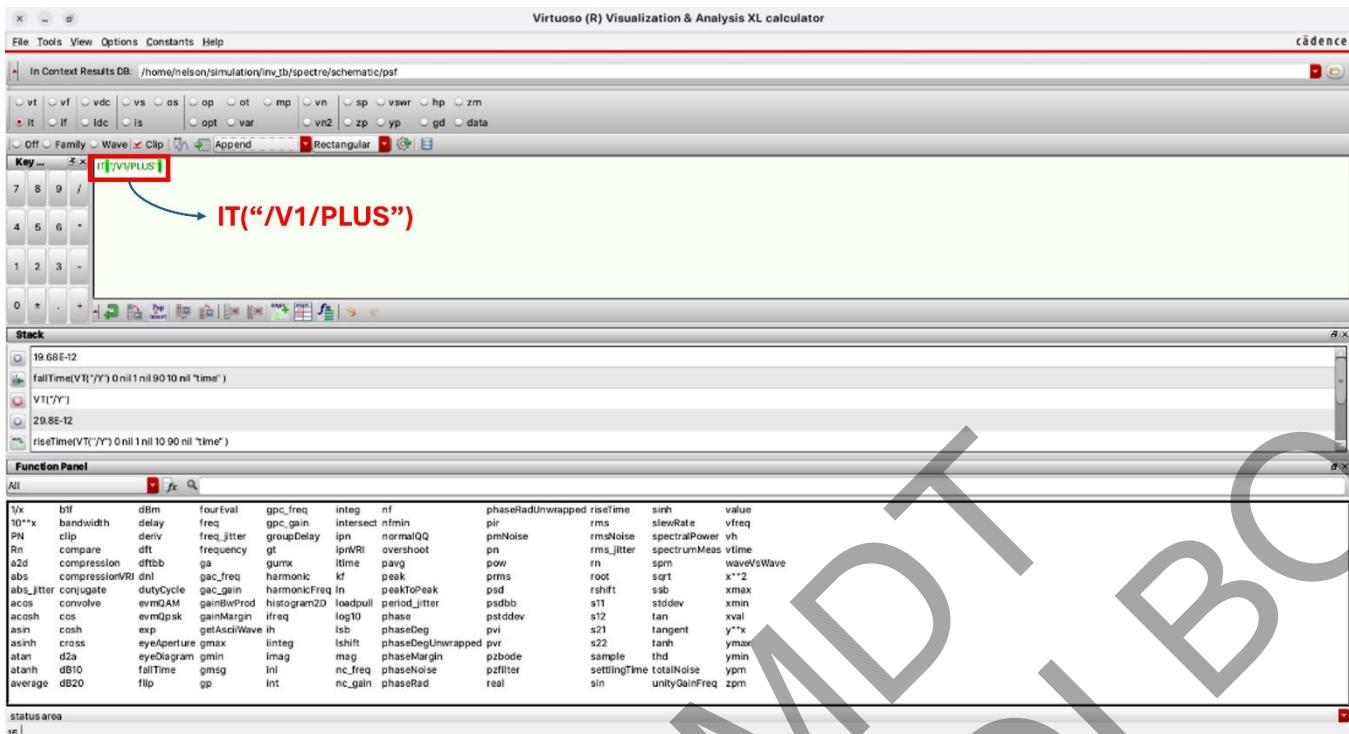


Figure 101 The name of the transient current is printed.

After defining them, students copy and paste them to complete the expression calculating total power consumption: $P_{total} = \text{average}(u(t) \times i(t))$

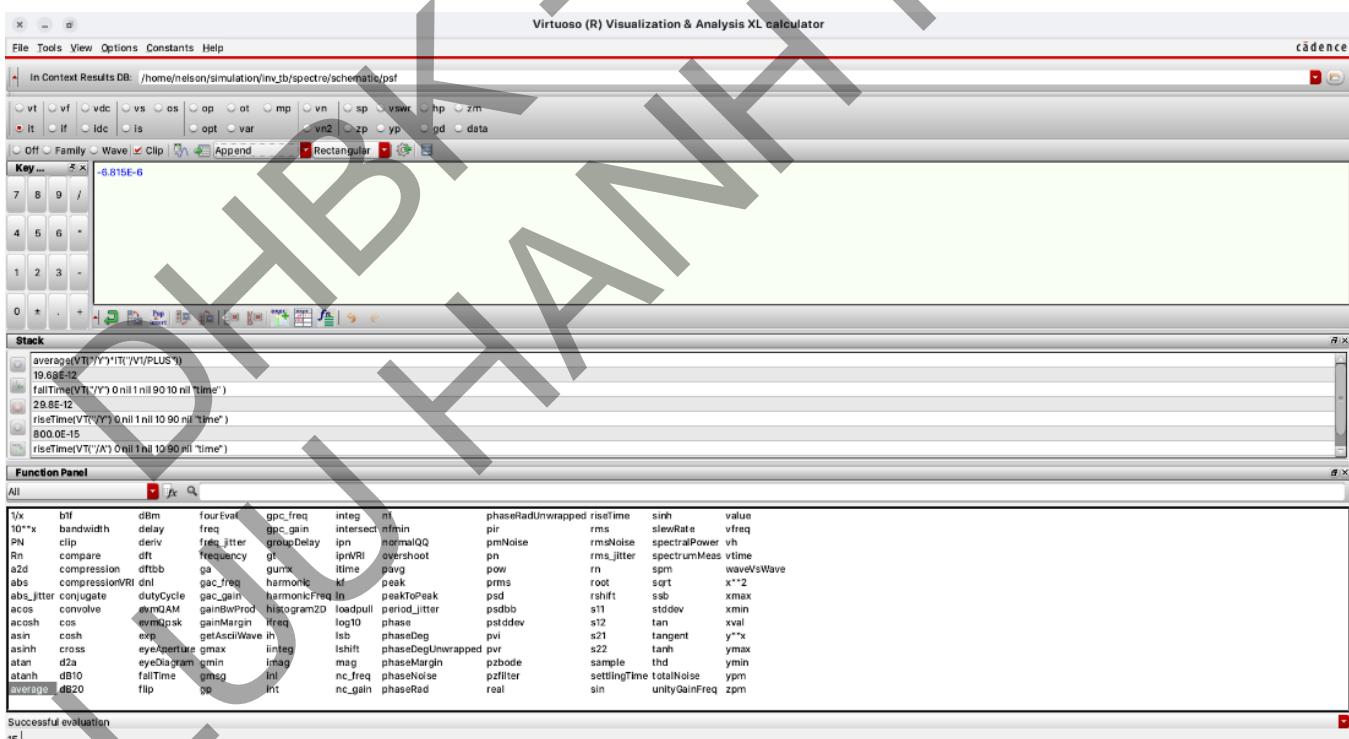


Figure 102 The result of total power consumption using Calculator.

$$P_{total} = 16.15 \mu W$$

❖ Dynamic power measurement:



Due to the dynamic power formula: $P_{dynamic} = C_L V_{DD}^2 f = 400 \times 10^{-15} \times 1 \times \frac{1}{20 \times 10^{-9}} = 20 \mu W$

Due to static and total power measurement results:

$$P_{dynamic} = P_{total} - P_{static} = 16.15 \times 10^{-6} - 58.2688 \times 10^{-12} = 16.14 \mu W$$

In addition, students can also add expressions to the Outputs of the ADE L window.

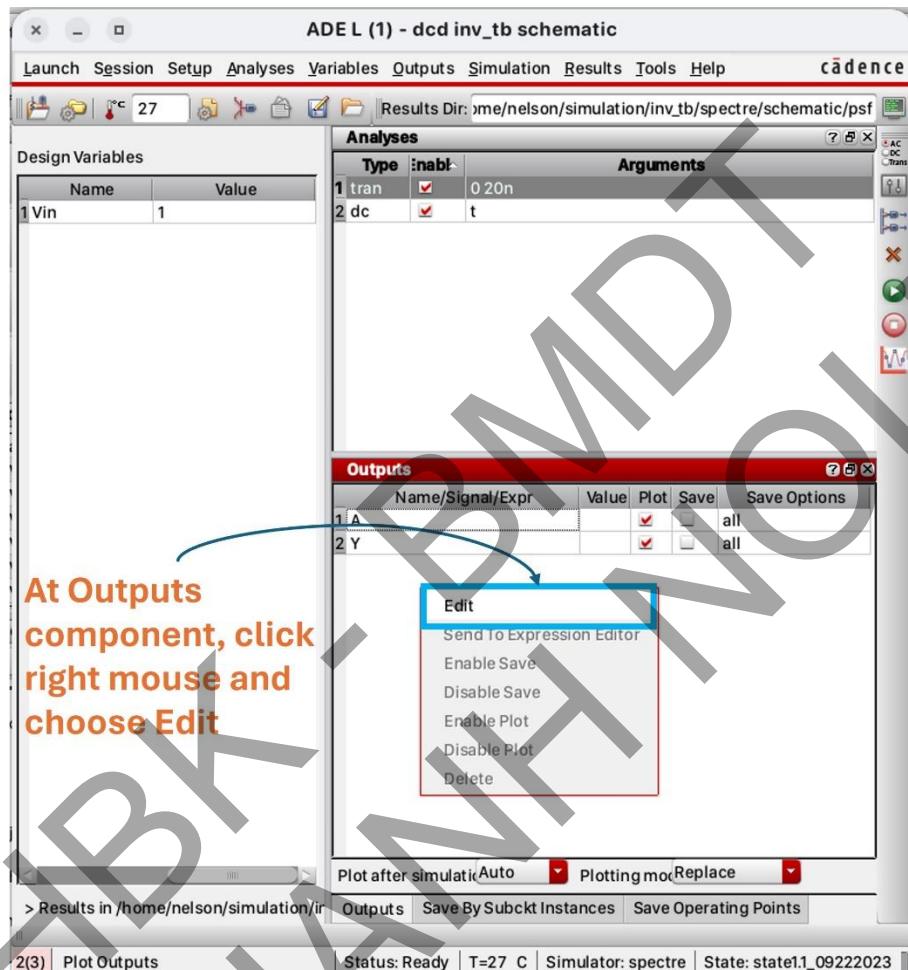


Figure 103 Another way to open the “Setting Outputs” window.

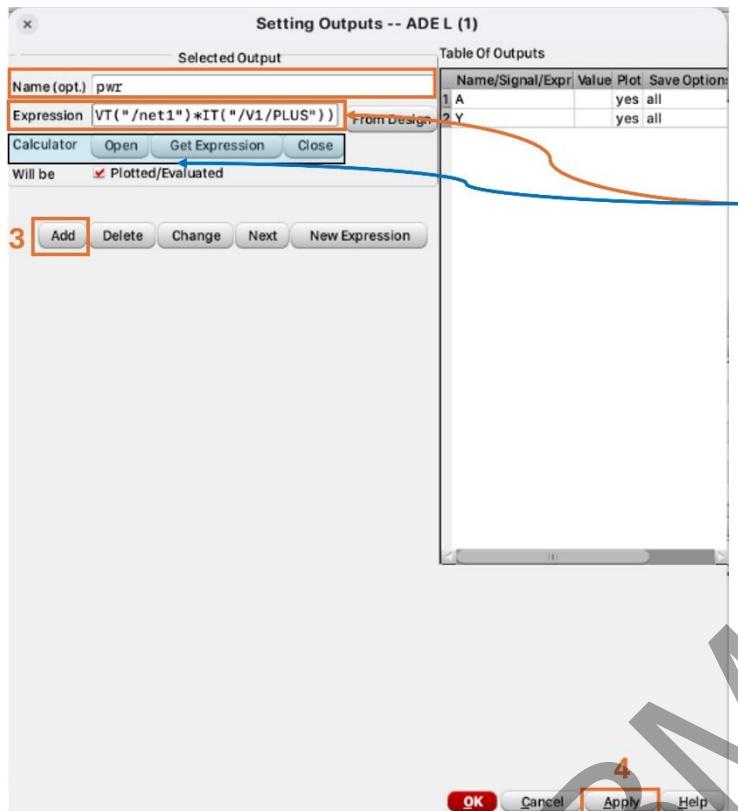


Figure 104 Add an expression to output.

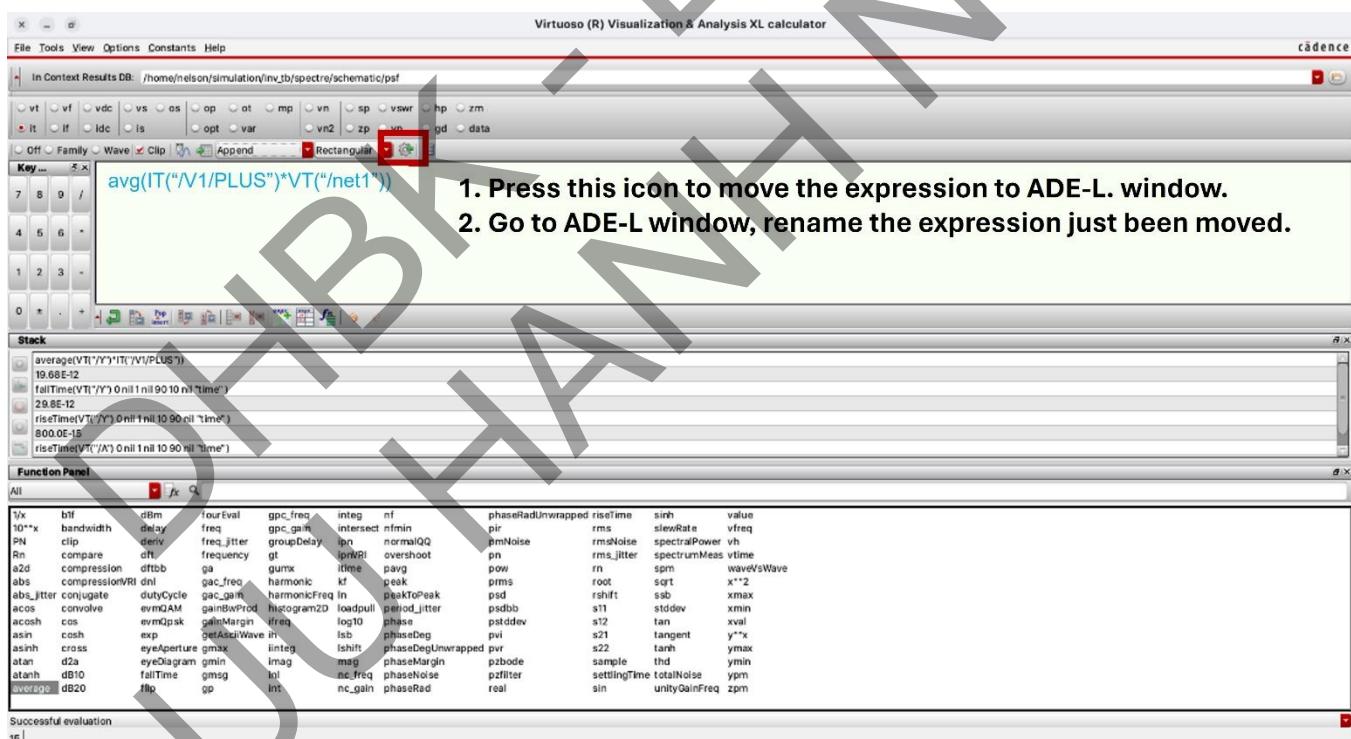


Figure 105 Another way to add expression to output.

After simulating successfully, at outputs dialogue, the value of total power consumption will be shown.

Perform these steps to add output:

1. Give a name for this output
2. Copy the expression from Calculator or you can use “Get Expression” button.
3. Add this output.
4. Apply.
5. You can repeat these steps to add more outputs as you want.
6. OK.

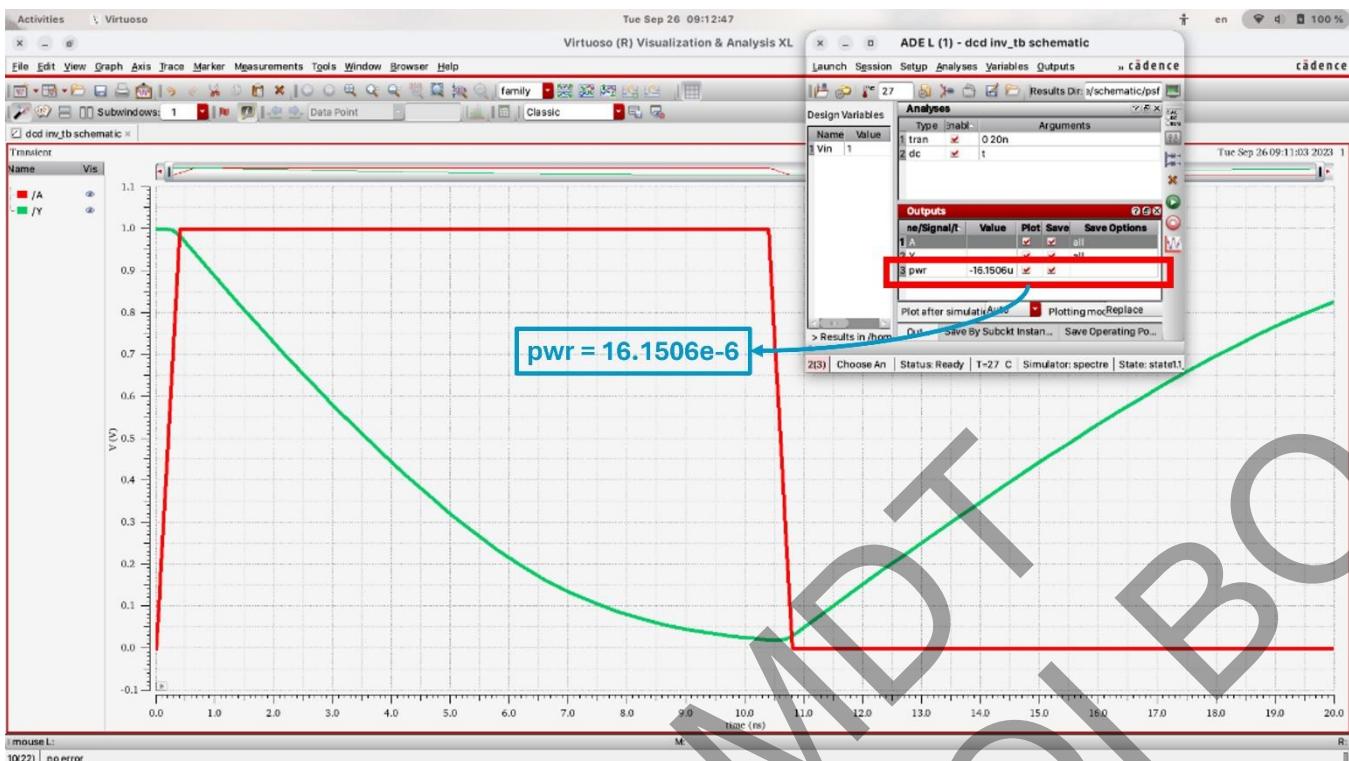


Figure 106 The value of total power consumption is printed in the Outputs dialogue.

DHQBK - LUU HANH - NOIBO



PART 5

Objective: Draw a layout of a CMOS-based inverter.

Requirements: Use the **Virtuoso Layout Editor** to create the inverter layout. Then, open Calibre to perform DRC, LVS, and PEX checks.

Instructions:

The instructions are divided into four key parts:

1. **Creating the Inverter Layout** – Students will follow a step-by-step process to build the layers of an inverter.
2. **Design Rule Check (DRC)** – Students will verify that the layout complies with design rules.
3. **Layout vs. Schematic (LVS) Check** – Students will ensure that the layout matches the schematic.
4. **Parasitic Extraction (PEX)** – Students will analyze parasitic effects in the layout.

Additionally, understanding the **physical implementation of circuits** is crucial, as it directly impacts **performance, power, and cost**. This is best visualized by examining both **top-view and cross-sectional** representations of a wafer:

- The **top view** is obtained by looking down on the wafer.
- The **cross-section** is obtained by slicing the wafer through the middle of a transistor and viewing it edge-wise.

Students should start by examining the **cross-section of a CMOS inverter**, then analyze the **top view** to define the **mask set** used in fabrication.

Now, let's begin the journey—**creating the CMOS inverter layout!**

1) How to create an inverter layout?

From the Virtuoso Schematic Editor window, perform **Launch > Layout XL**.



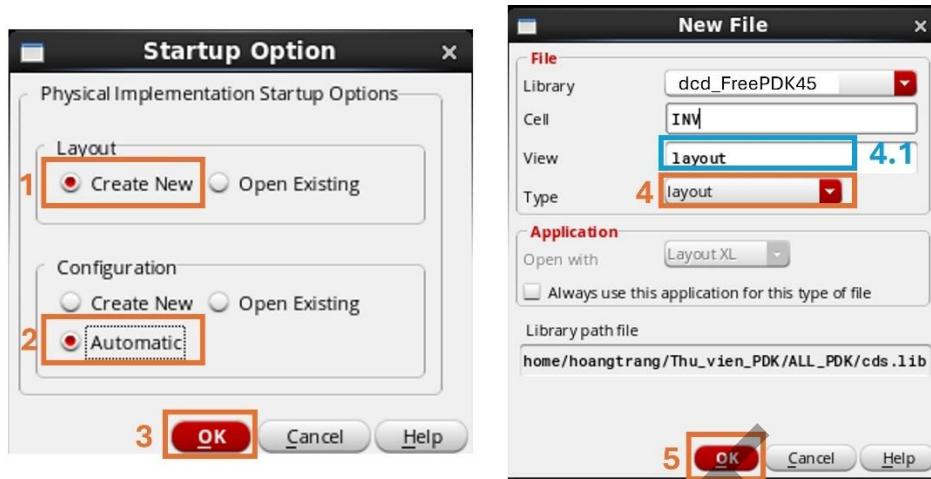


Figure 107 Startup menu and New File before opening Virtuoso Layout Editor.

The selection startup menu appears as shown in the image above. To create an entirely new design, select “**Create New**” and leave the automatic setup mode in the **Configuration**. The design creation window for the physical layout – Virtuoso Layout Suite XL – will appear as shown in Figure after pressing OK.

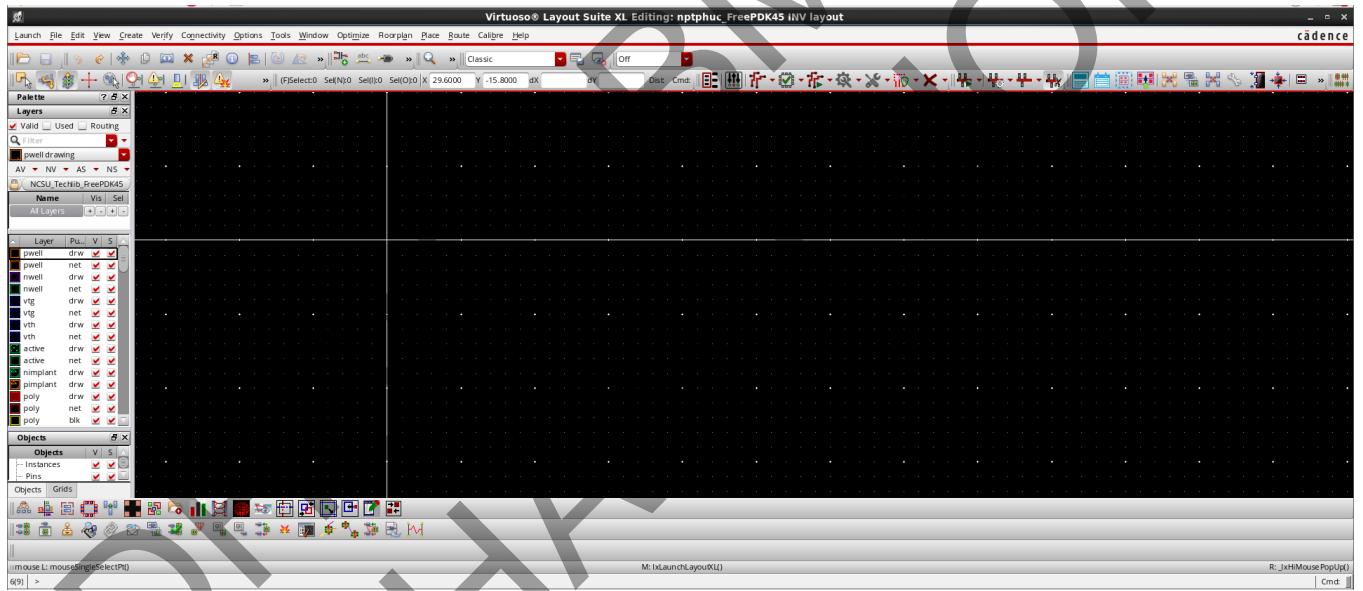


Figure 108 Virtuoso Layout Suite XL window.

Since a layout is essentially a huge number of rectangular shapes with specific sizes, the first thing we need to do is to setup the grid dimensions that help make drawings easy. Click **Options > Display** to bring out the *Display Options* dialog. The options students need to change are in the *Grid Control* panel in the top-right corner.

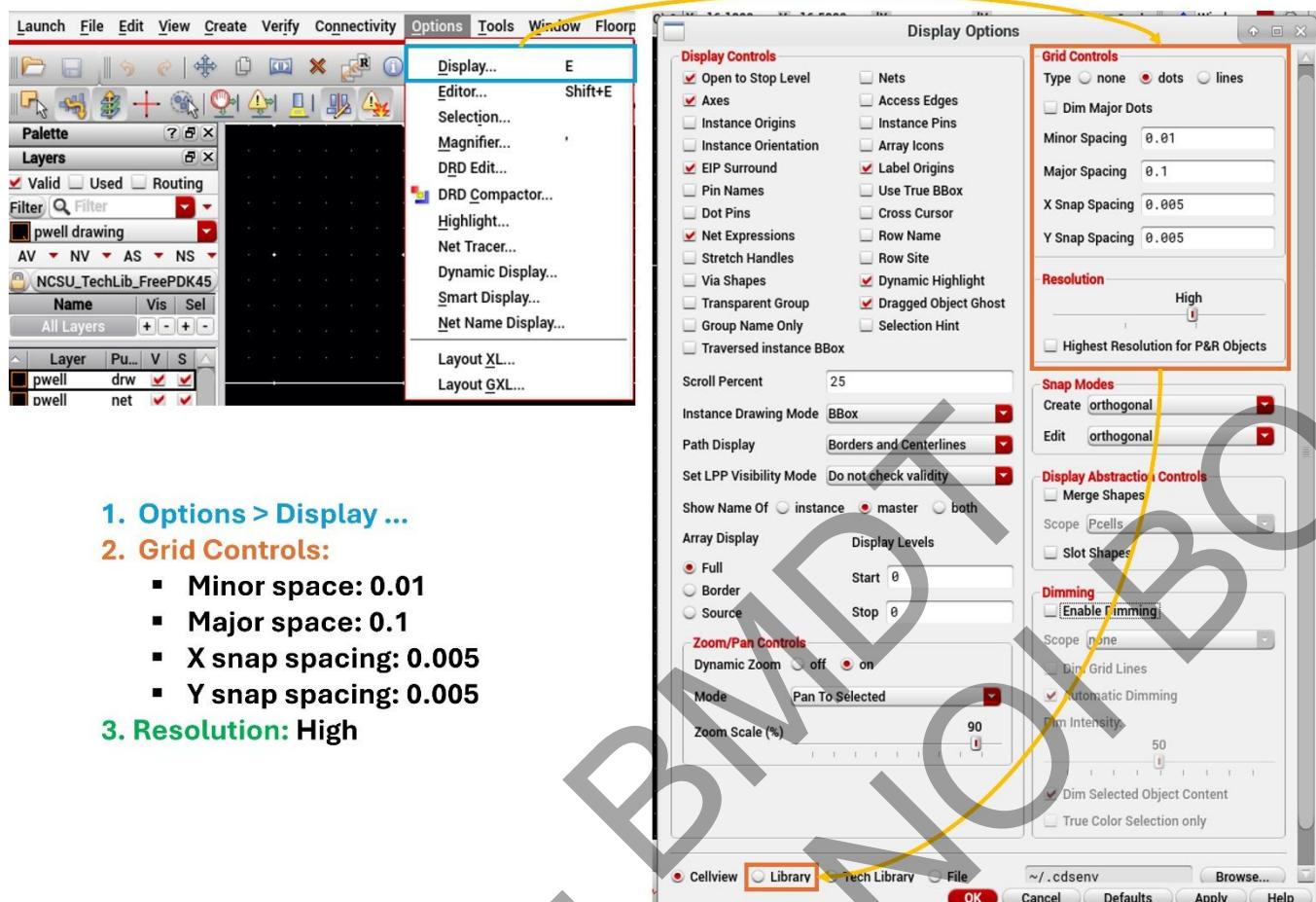


Figure 109 Change snap spacing.

Save the settings by choosing “Library” from the bottom row and clicking the “Save To” button. Click OK to dismiss the Display Options dialog. Press **Ctrl+G** to refresh the grid. Students can move the cursor around to see how the ‘X’ and ‘Y’ change in the toolbar. Moreover, students can use dimming and unable grid to observe easily when drawing.

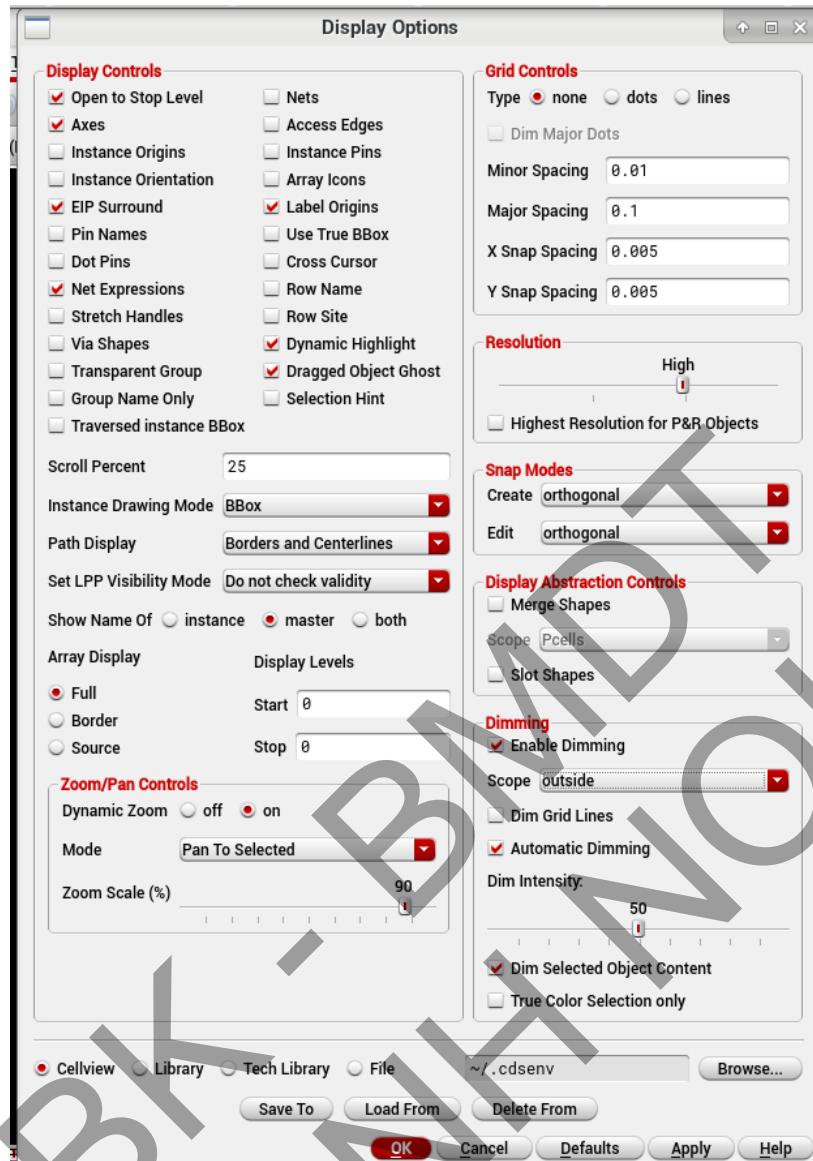


Figure 110 Dimming options and settings.

For more convenience when drawing, please read bindkeys summarized in part 2. However, students can read some common commands used frequently.

Bindkey	Function
K	Create Ruler
Shift + K	Remove all rulers
R	Rectangle
Shift + P	Polygon
S	Stretch
M	Move
C	Copy
P	Path

Shift +M	Merge
Shift + C	Chop
I	Instance Cell
O	Contact
L	Label
Ctrl + S	Split
U	Undo
F	Fit all

Table 12 Some bindkeys for drawing layout.

1.1. Step 1: Create NMOS

To draw a transistor layout, we first need to decide the dimension of the active region. Since we plan to place the poly gate vertically, the transistor width will decide the height of the active region. For the NMOS transistor, it is 90nm. Note that this is the minimum width as required by [ACTIVE.1](#).

It is much more complicated to decide the width of the active area since we need to accommodate the source/drain plus contacts to metal1. The following layout rules should be observed:

- [CONTACT.4](#) spacing between boundary of active and contact, round up to 10nm.
- [CONTACT.1](#) width of contact, round up to 70nm.
- [CONTACT.6](#) spacing between contact and poly, round up to 40nm.
- [POLY.1](#) minimum width of poly, 50nm, which is also the transistor length throughout all labs.

Putting everything together, the active region should have a width of:

$$10 + 70 + 40 + 50 + 40 + 70 + 10 = 290\text{nm}$$

Choose “*active|draw*” from the LSW window on the left. Press **R** and drag a rectangle measuring $290\text{nm} \times 90\text{nm}$. This will be the active region for the NMOS transistor.



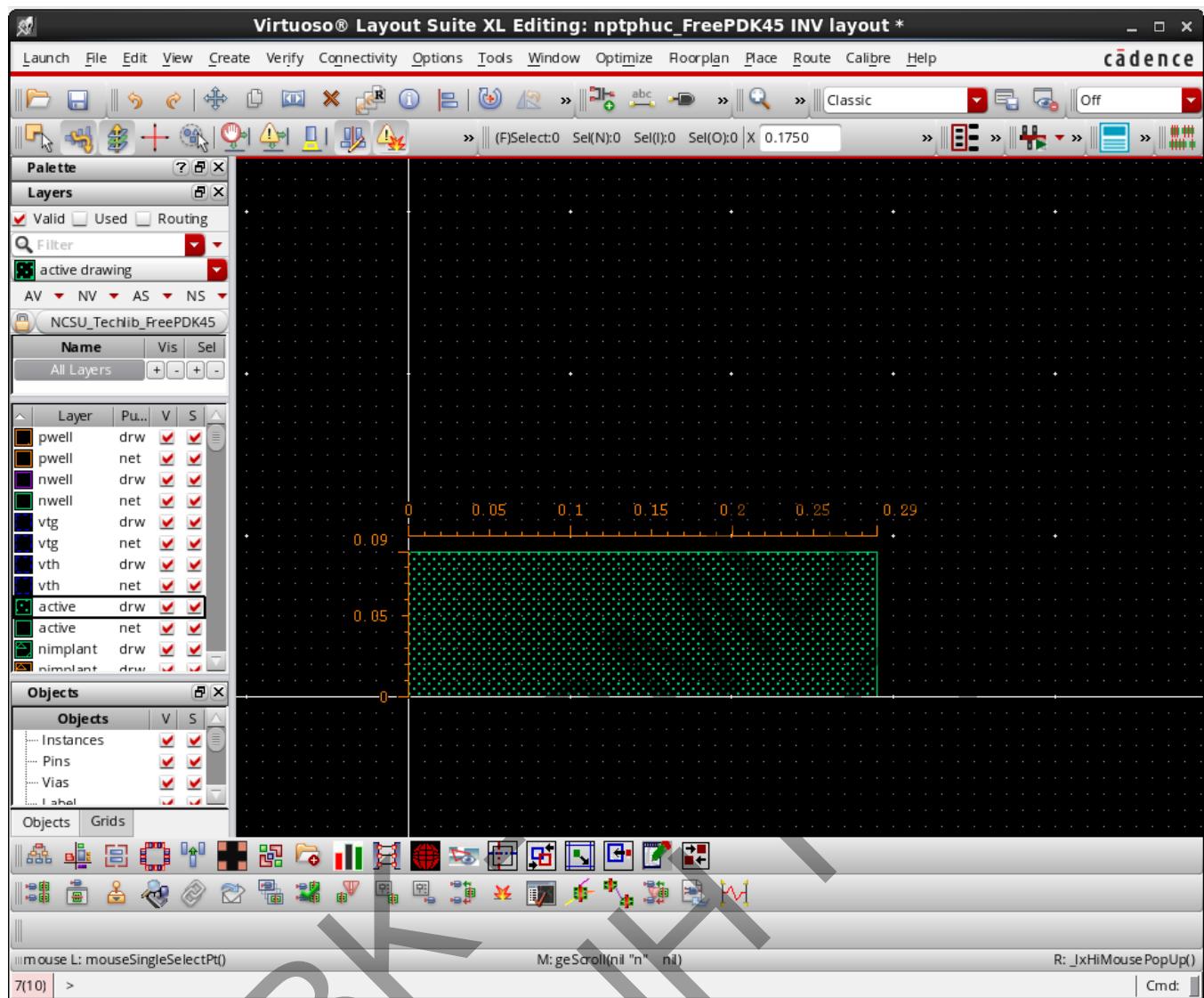


Figure 111 The ACTIVE region for the NMOS transistor.

Now, add two $70nm \times 70nm$ CONTACT by choosing “*contact|drw*” and then a POLY rectangle $50nm$ wide crossing the active region by choosing “*poly|drw*”. Students should space them according to the above layout rules.

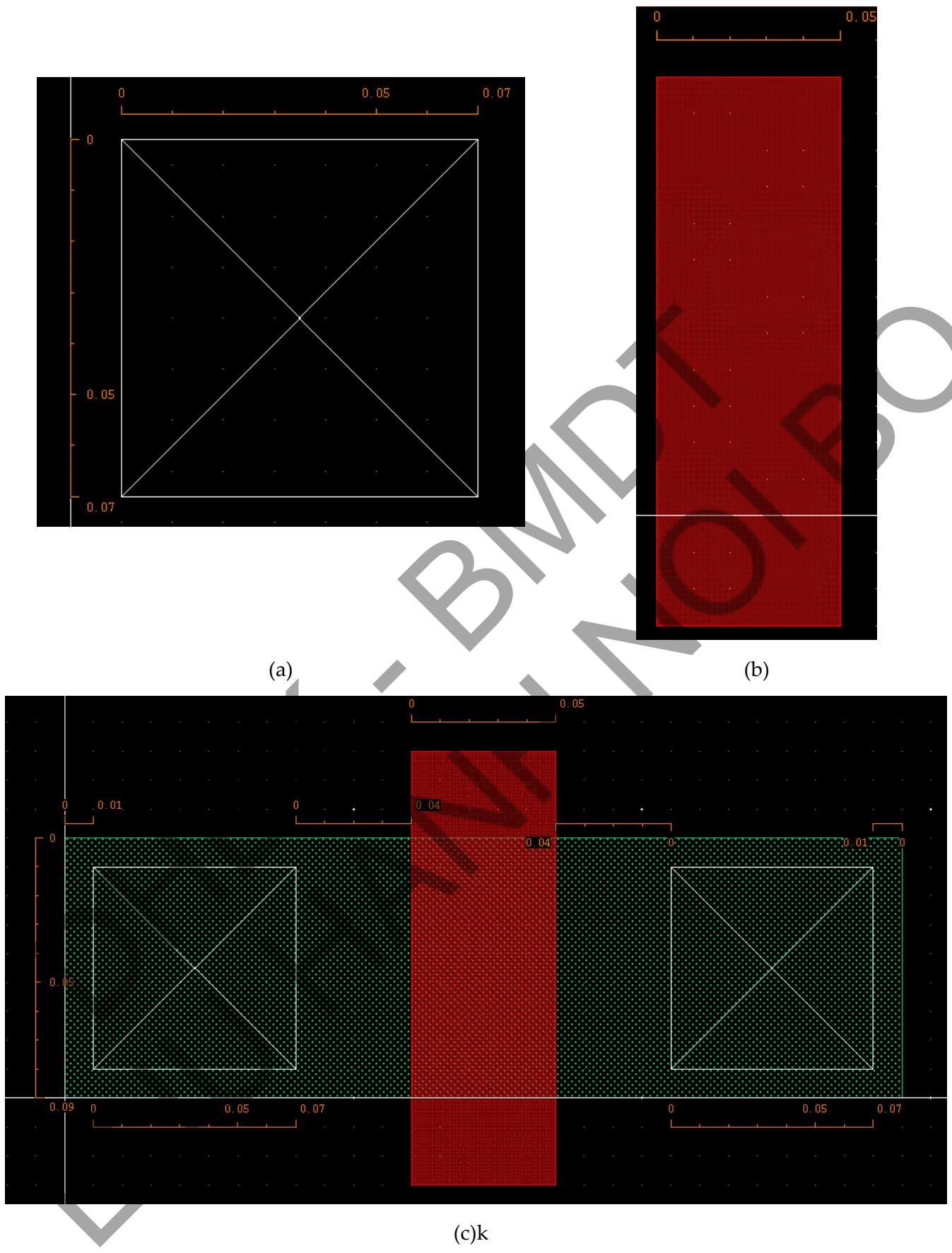


Figure 112 (a) Create CONTACT 70nm × 70nm (b) Create a POLY rectangle 50nm wide (c) Add CONTACT and POLY to the ACTIVE region of NMOS.

We need to make a well-tap to connect the body of the NMOS transistor to GND. This would require a separate active region that can accommodate a contact. Based on the rules:

- **CONTACT.4** and **CONTACT.1**, the dimension should be $90nm \times 90nm$.
- The rule **ACTIVE.2** further demands a spacing $80nm$ between the two active regions.

As an alternative to draw a new contact, students can make a copy of the existing contact by first press **C** and then click on the existing contact.

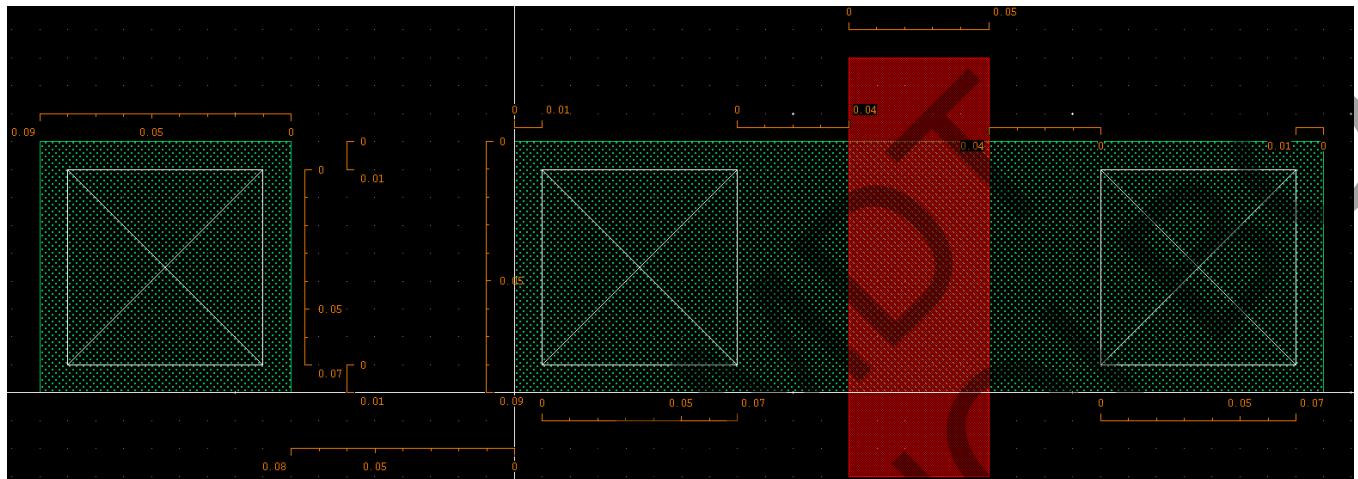


Figure 113 Create well-tap of NMOS.

We are ready to draw the pwell to hold all the previous drawing:

- The rule **ACTIVE.3** demands a enclosure of $60nm$ (round up from $55nm$).

Choose “**pwell|drw**” from LSW and draw a rectangle measuring $580nm \times 210nm$ that encloses the active areas.

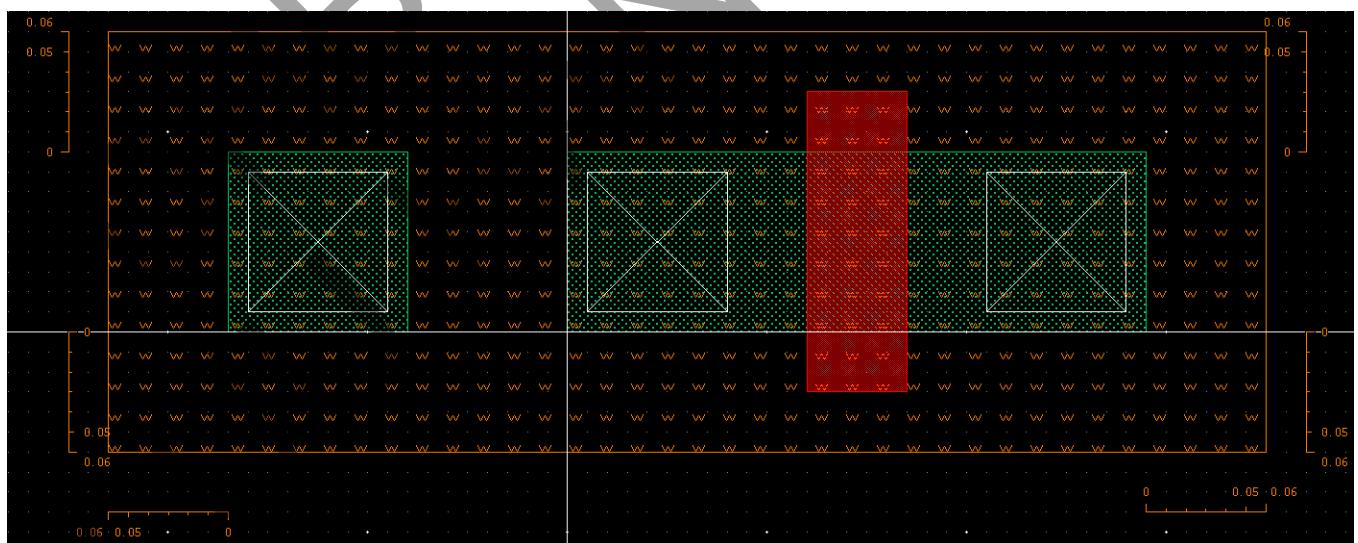


Figure 114 Create pwell.

Don't forget to overlap the two active regions with the correct types of implants:

- The active region for the NMOS transistors should use “**nimplant|drw**”

- The active region for the well-tap should use “*pimplant|drw*”.

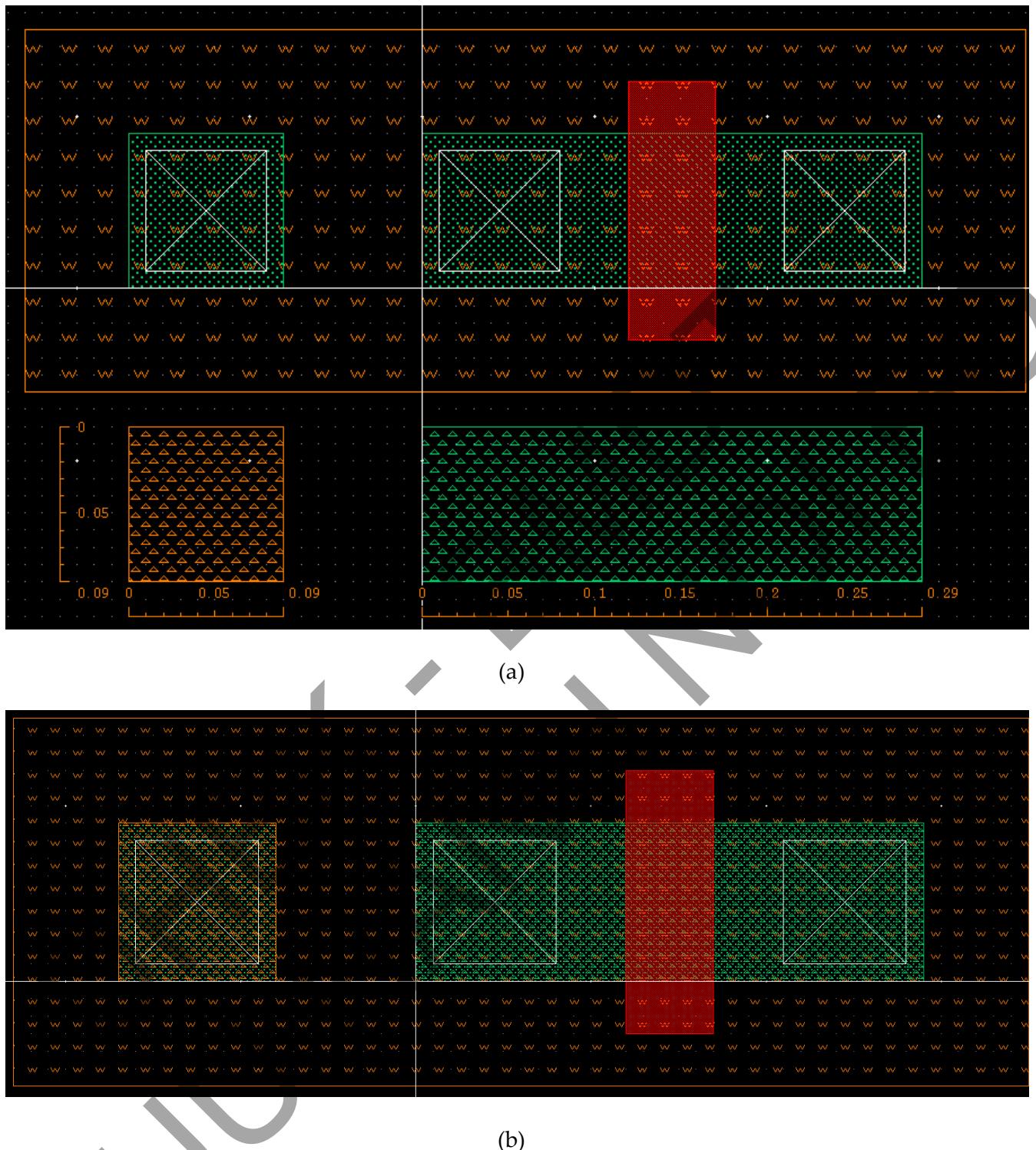
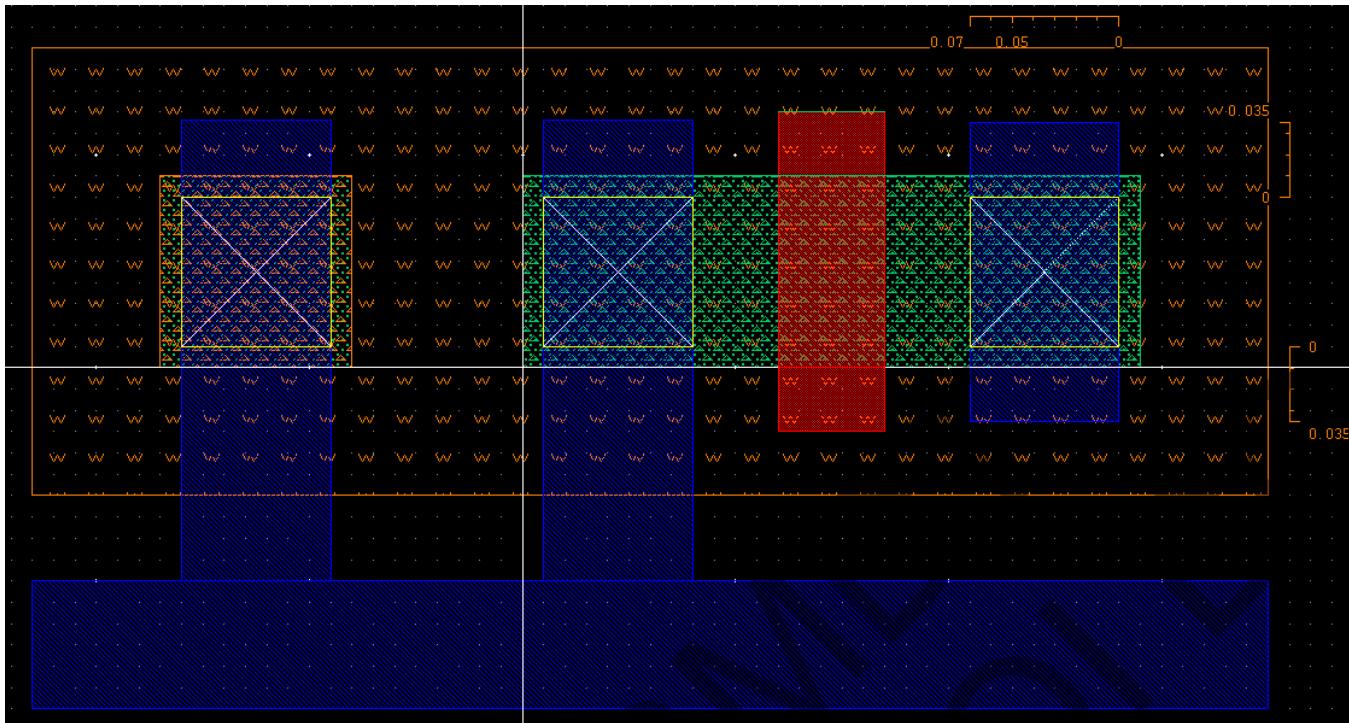


Figure 115 (a) Create NIMPLANT and PIMPLANT (b) Overlap implants to the active regions.

Finally, we create the GND rail and make connections using **metal1**. The following layout rules should be observed:

- METAL1.1** minimum metal1 width, round up to 70nm.
- METAL1.2** minimum metal1 spacing, round up to 70nm.

- METAL1.3 enclosure between metal1 and contact, round up to 40nm.



1.2. Step 2: Check DRC for NMOS creation in step 1

Due to the overwhelming number of layout rules, students will likely miss some during students layout design. It will be very difficult for students to modify student's layout once students have everything there. Therefore, students need to locate the errors and correct them as early as possible. This goal is achieved by using a design rule checking (DRC) tool to check students' designs frequently.

The DRC tool we are going to use is part of *Calibre from Mentor Graphics*. When Virtuoso loads the FreePDK45 library, *Calibre* is integrated into the Virtuoso framework and the layout rules are imported. We can therefore go directly to use it without any further configuration.

Perform in Virtuoso Layout Suite XL: **Calibre > Run mmDRC**. Because this is the first DRC run, students can choose "Cancel" when asked about the "**Runset File**." A runset file will save all our DRC run settings.

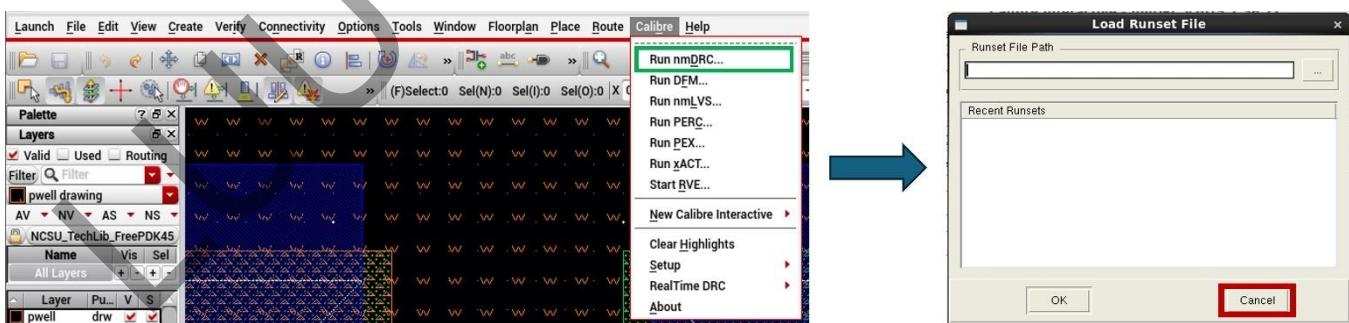


Figure 116 Open Calibre for running DRC and cancel load runset file.

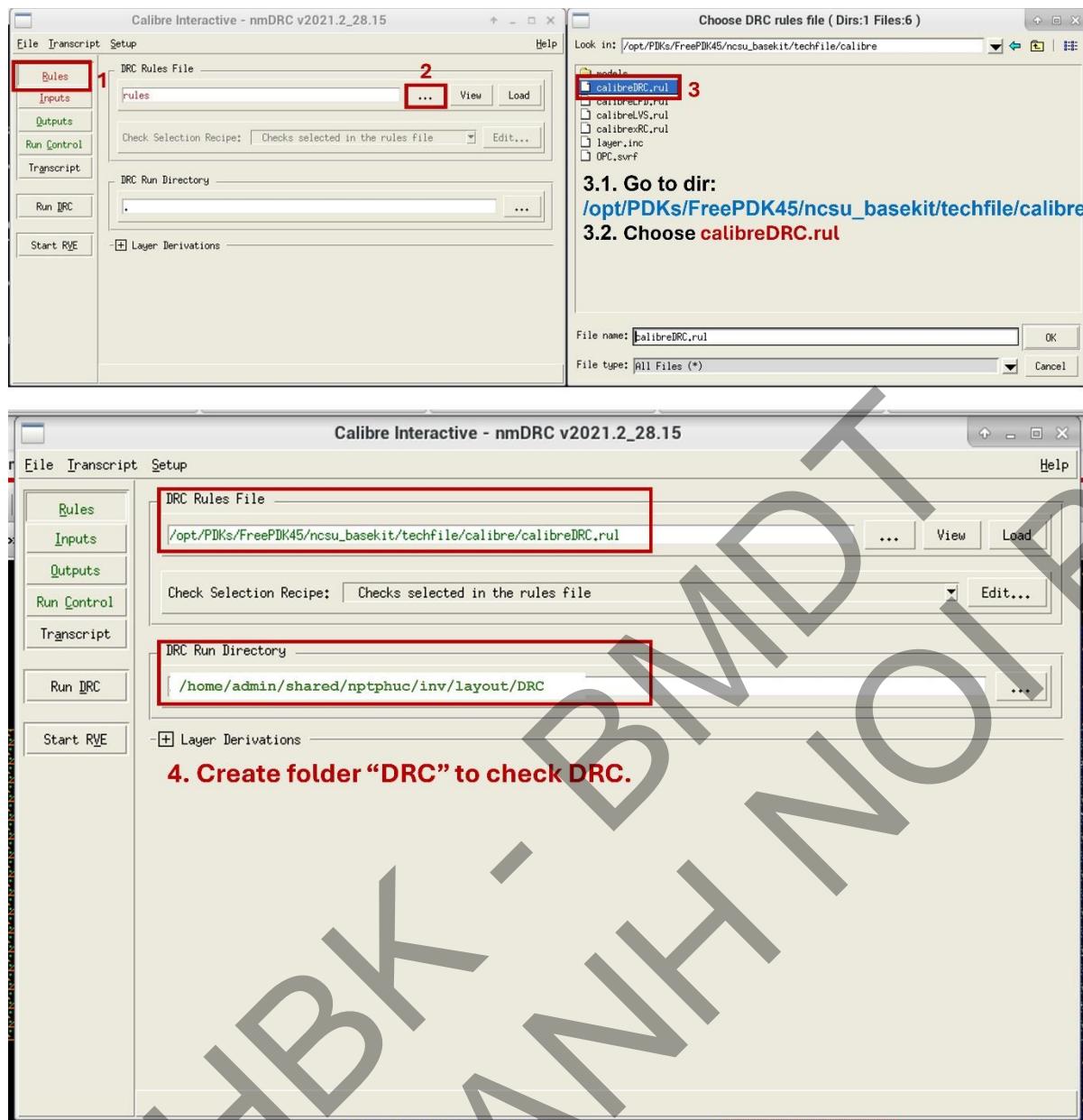


Figure 117 Choose the Design Rules file to run DRC.

Next, in the “Inputs” tab, please check “Export from layout viewer”. The setting in the “Outputs” tab is shown.

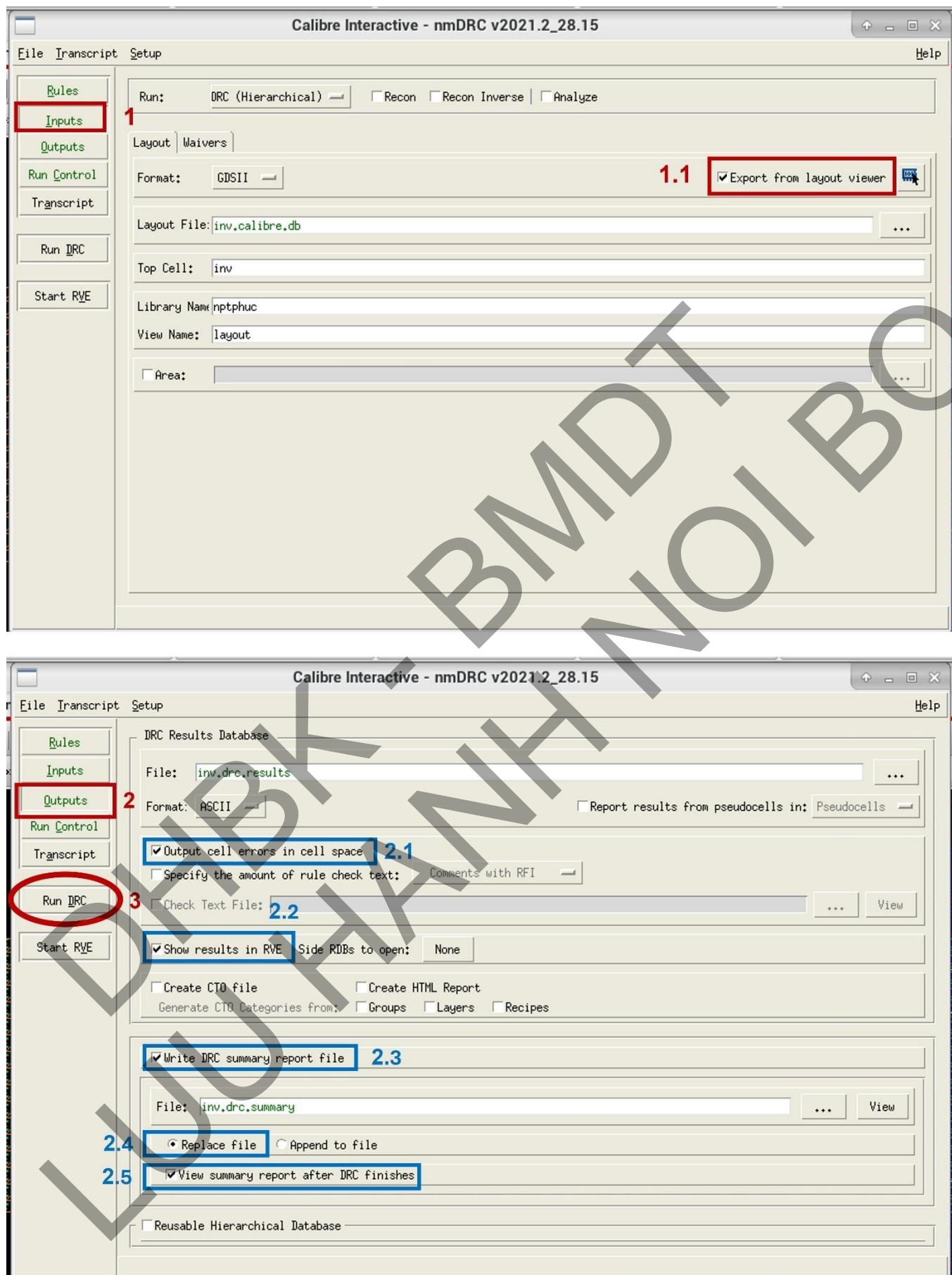


Figure 118 Setting inputs and outputs before running DRC.

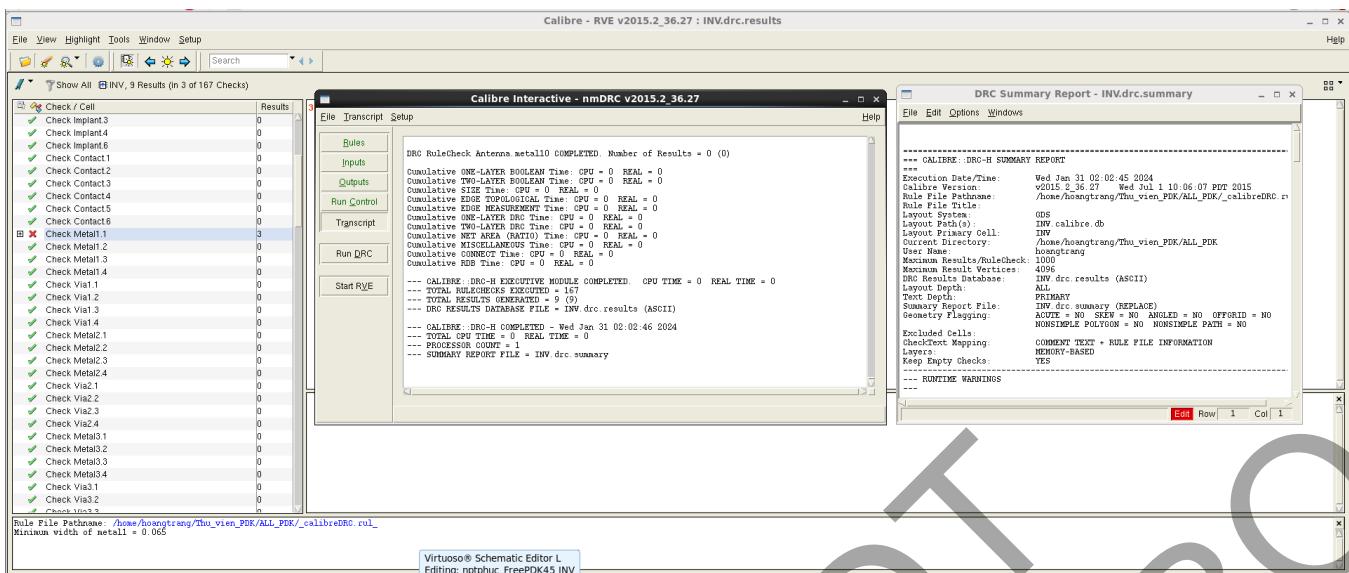


Figure 119 The result of the first run DRC.

The error occurs at Check POLY3 and METAL1.1. Students can observe the error by highlighting:

Highlight > Highlight all

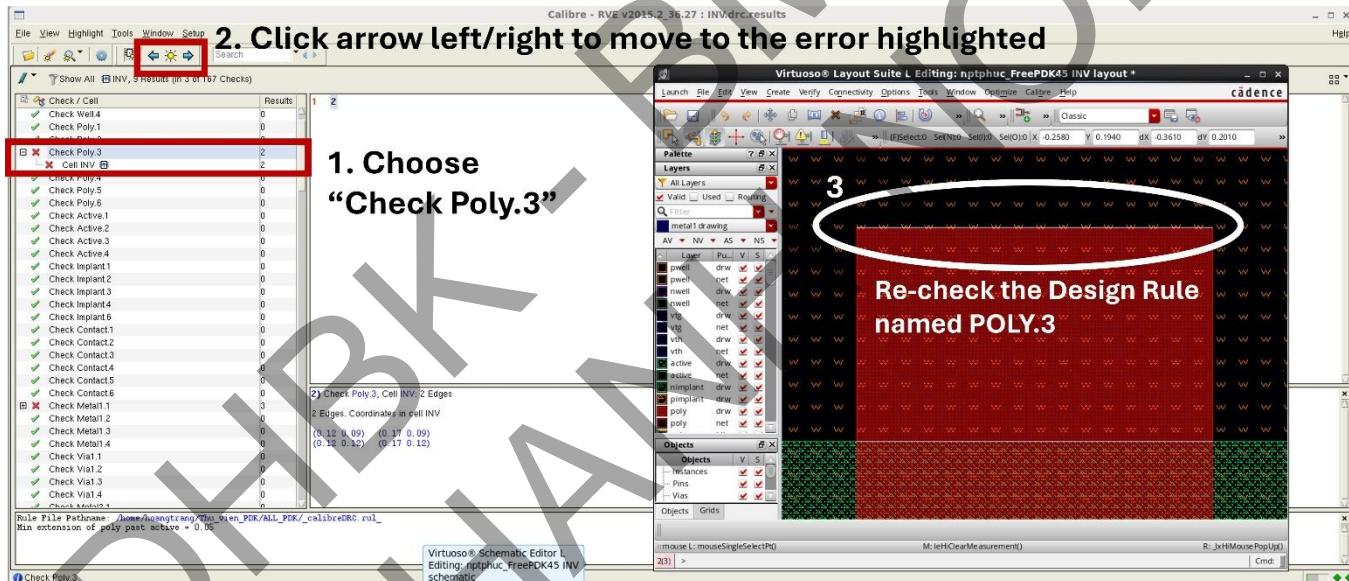


Figure 120 Check POLY.3 error.



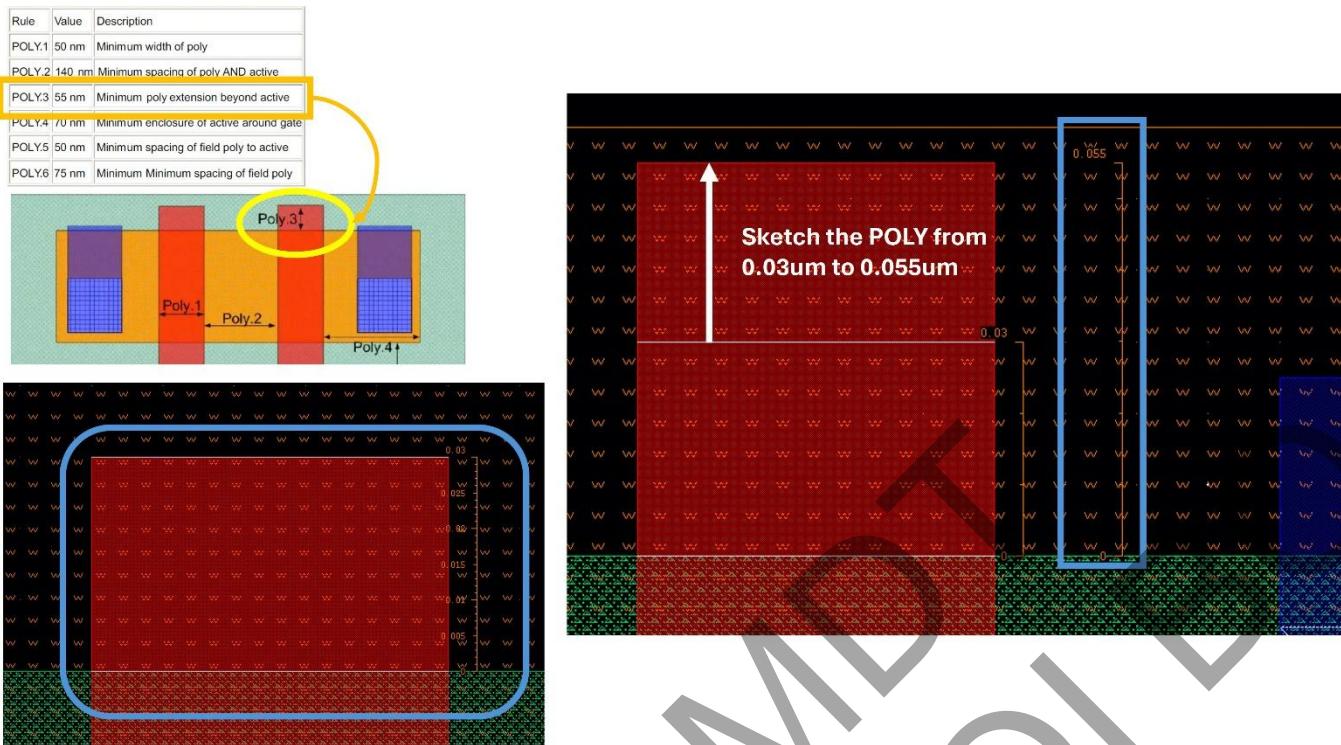


Figure 121 Check rule named POLY.3 and solution.

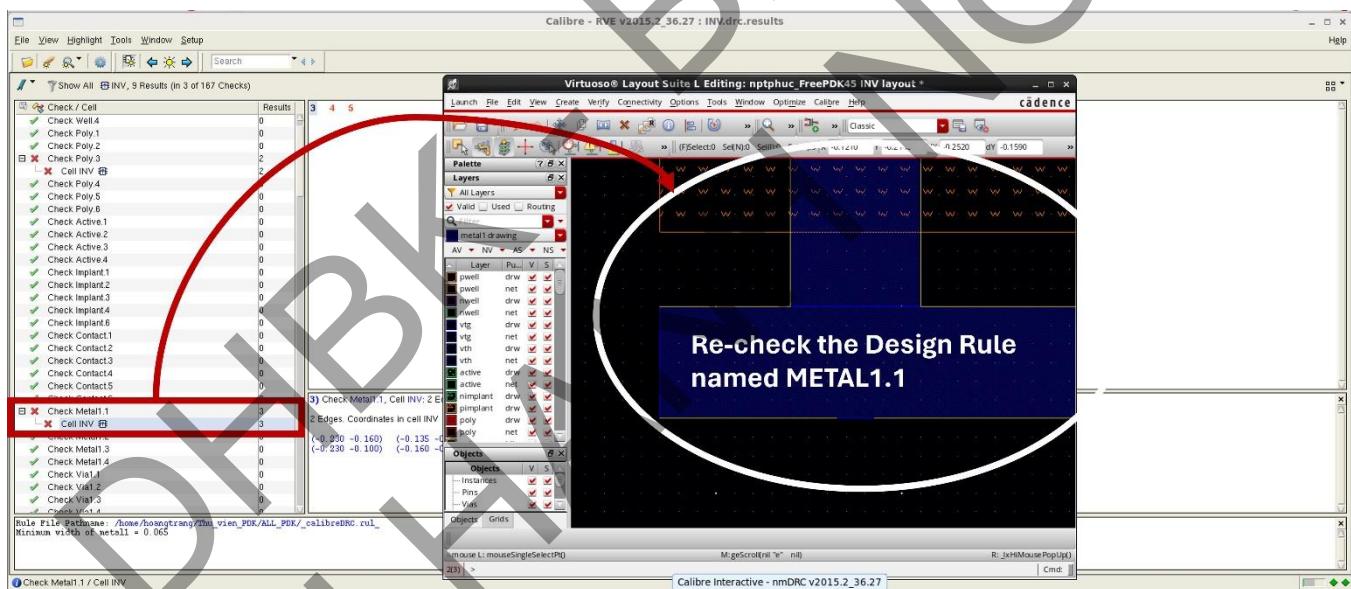


Figure 122 Check METAL1.1 error.

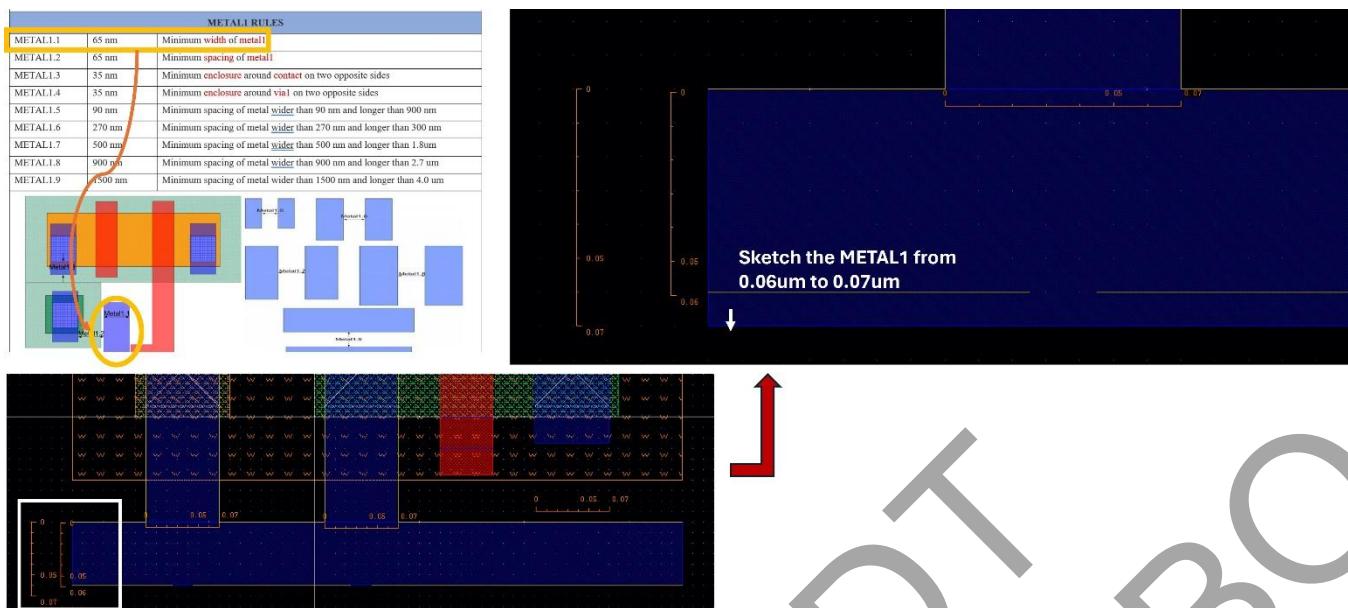


Figure 123 Check rule named METAL1.1 and solution.

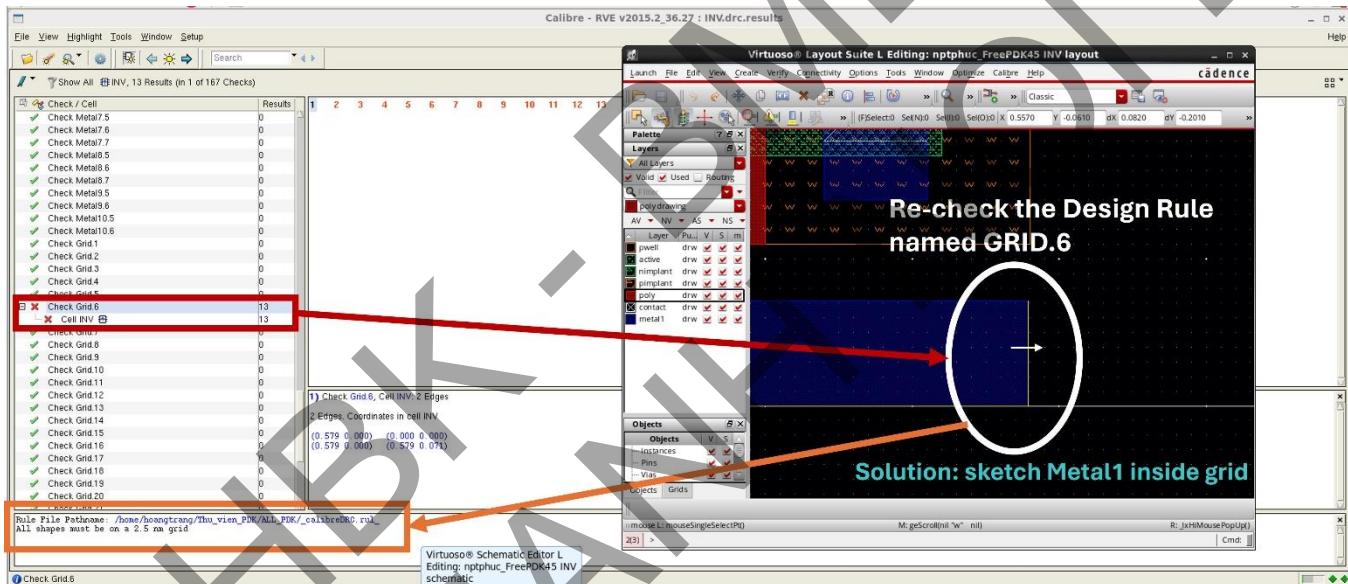


Figure 124 Check Grid.6 error and solution.

All the solutions are shown. Run DRC again. There should be no error, and vice versa, feel free to fix all the errors.

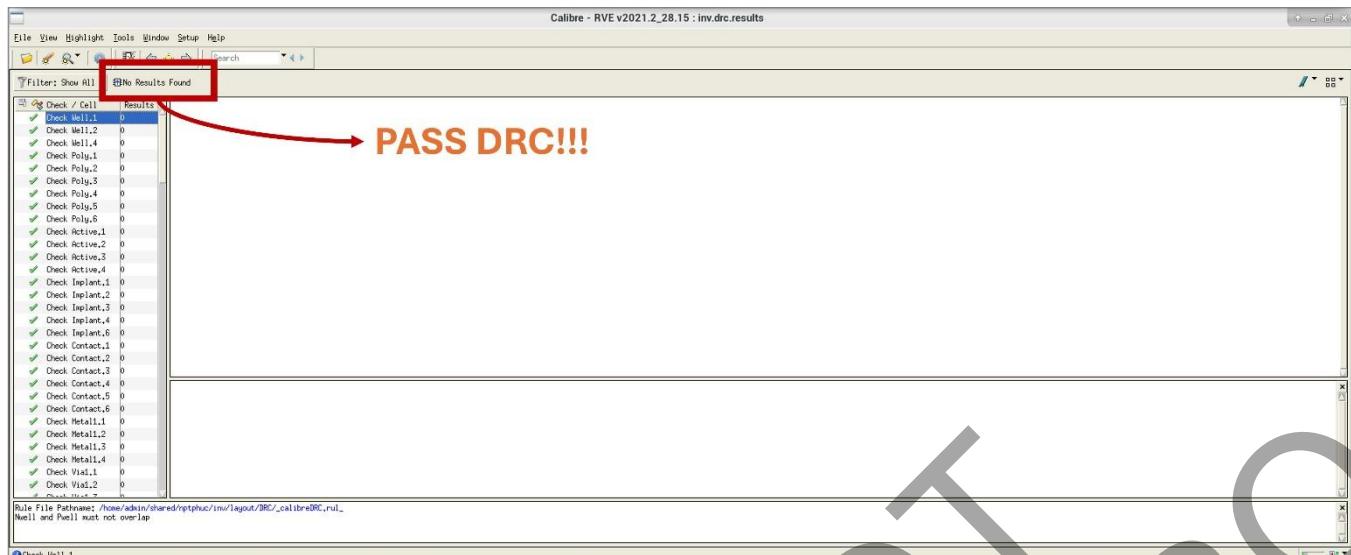
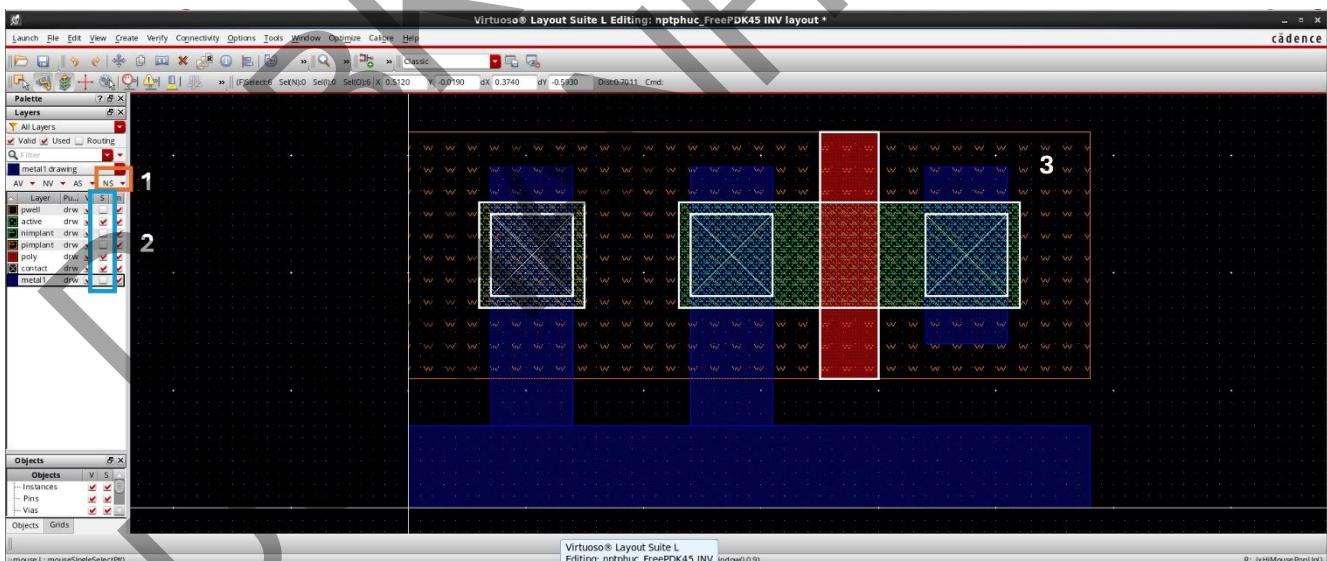


Figure 125 All errors are corrected.

Once students correct all errors, students may close the report and the Calibre RVE window. Students can leave the Calibre Interactive DRC window open so the layout can be checked frequently.

1.3. Step 3: Create PMOS

We can draw the layout of the PMOS transistor using that of the NMOS transistor as a template. Click the 'NS' button in the LSW window, which will make all layers not selectable. Then right click 'active|drw', 'contact|drw', and 'poly|drw'. Drag a big box around the NMOS transistor and students will find only the above three layers are selected.



1. Click the **NS** button in the *Layers* window.
2. Right click **active|drw**, **contact|drw**, and **poly|drw**
3. Drag a big box around the NMOS transistor → the above three layers are selected.
4. Press **C** and click on any part of the selection to make a copy.

(a)



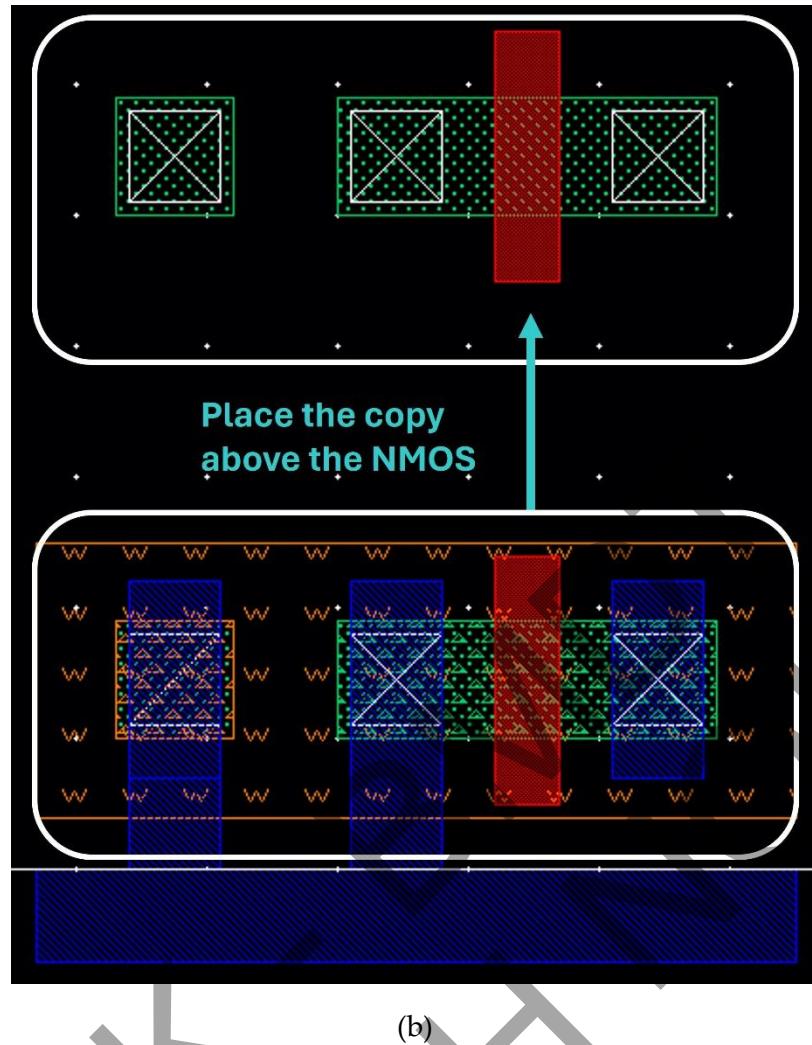


Figure 126 (a) Copy the selected layers (b) Place the copy to create PMOS.

Since the width of the PMOS transistor is 180nm, we need to stretch (press 's') the active region to have a height of 180nm. The poly gate should be stretched as well. On the other hand, it is not necessary to change the widths and horizontal spacings, which saves us a lot of work. After adding the correct types of implants and drawing a nwell, students will have the PMOS transistor ready as follows.

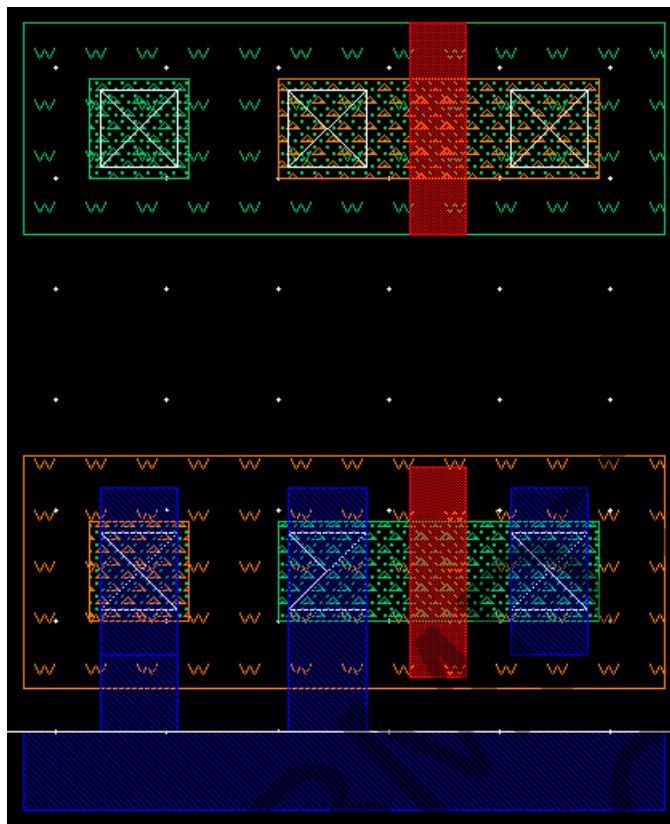


Figure 127 PMOS creates complete.

Run DRC again. This time it shows 2 errors.

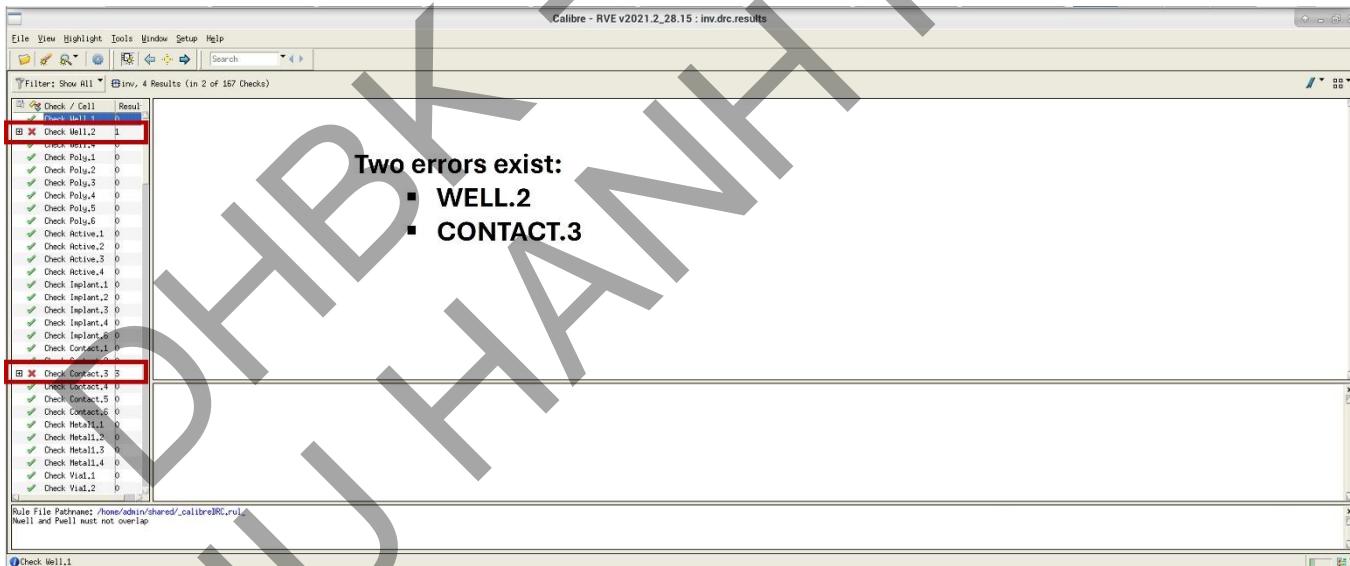


Figure 128 DRC results.

Expand 'Check Well.2' and then double-click the '1' in the top-right panel. Students will notice the bottom of the nwell and the top of the pwell are highlighted in the student's layout -- that's where the rule Well.2 fails. Students can read the rule from the bottom-right panel without going online -- pwell and nwell should be separated by at least 225nm.



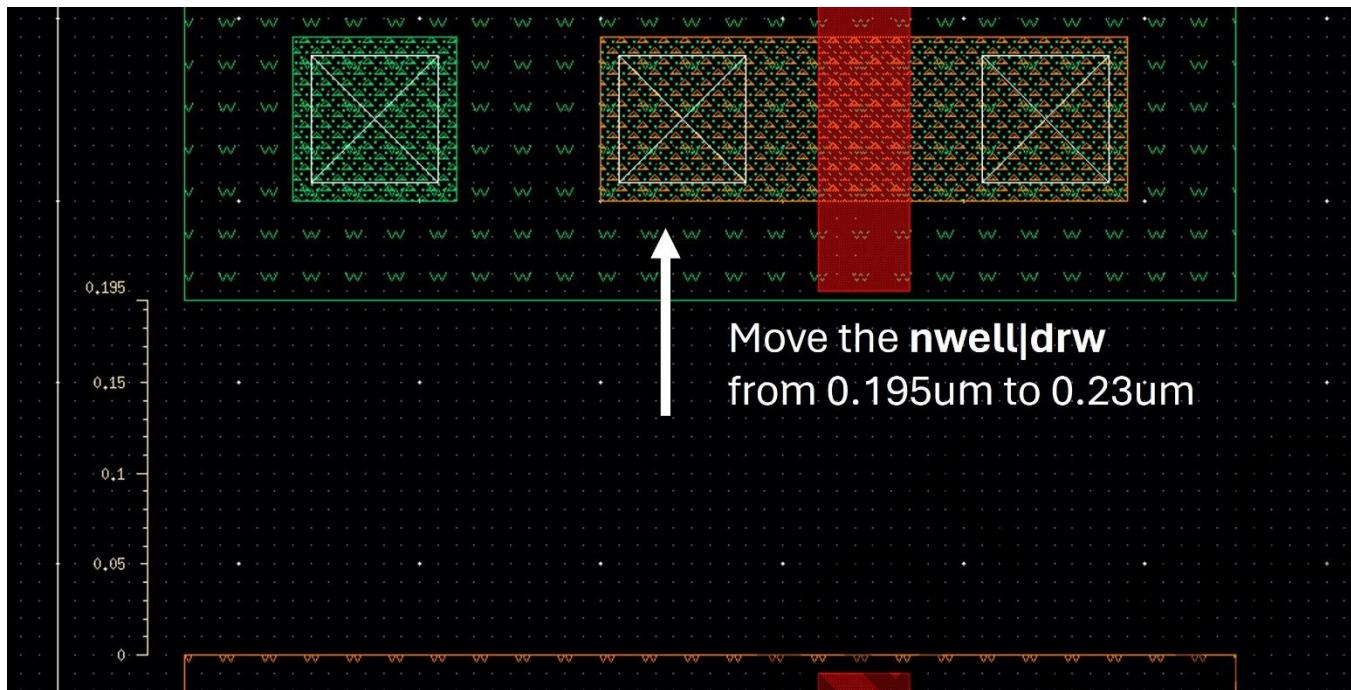


Figure 129 Move PMOS to fix WELL.2 error.

1.4. Step 4: Arrangement and connection

The other three errors are with the contacts. The rule says contacts should be inside metal1 and active or poly. This will not be a concern once we connect them to metal1. Using metal1 to create VDD rail and make necessary connections. Students should also connect the polys for the input and connect it to the metal1 layer to make hierarchical design easier.

- This work is also known as creating VIA – a connection between POLY and METAL1 (Port In will use them). Notice that a **VIA is performed by three layers: CONT, POLY, and METAL1 stacked**. The VIA of M1-POLY is shown as follows, please pay attention to the rules named **CONTACmT.1**, and **CONTACT.5** in the appendix.

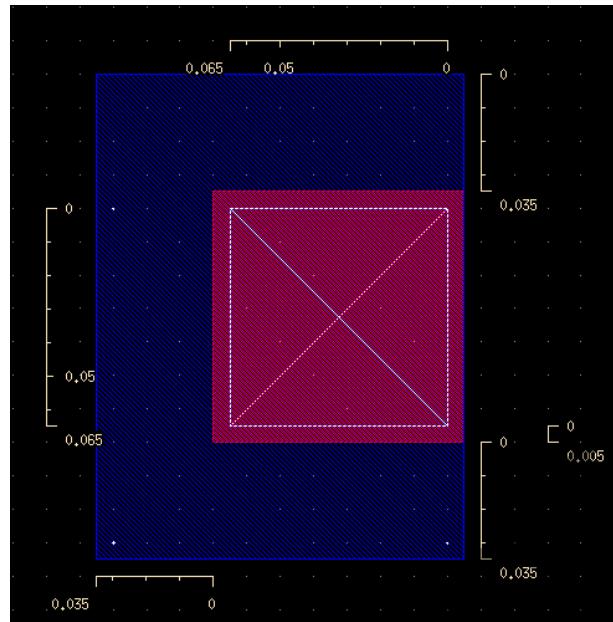
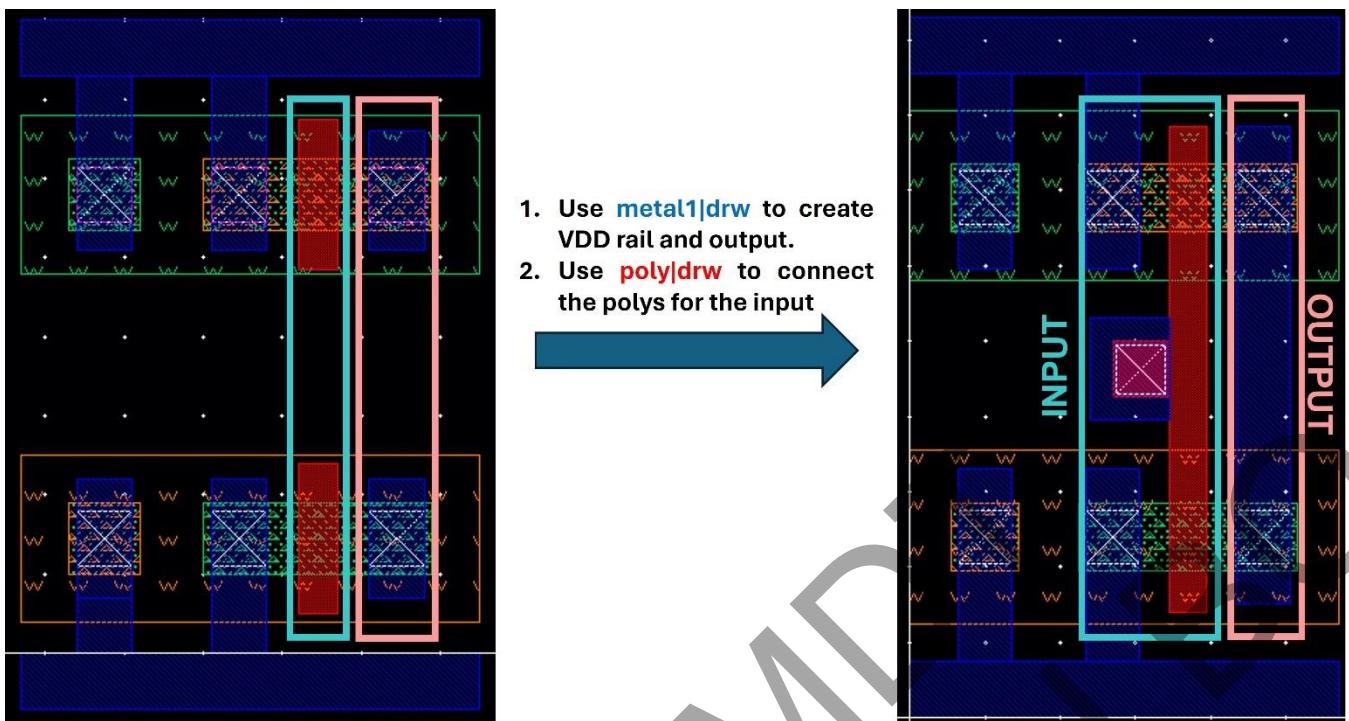


Figure 130 The VIA of M1-POLY.

Students are not allowed to use VIAs already existing in any library.

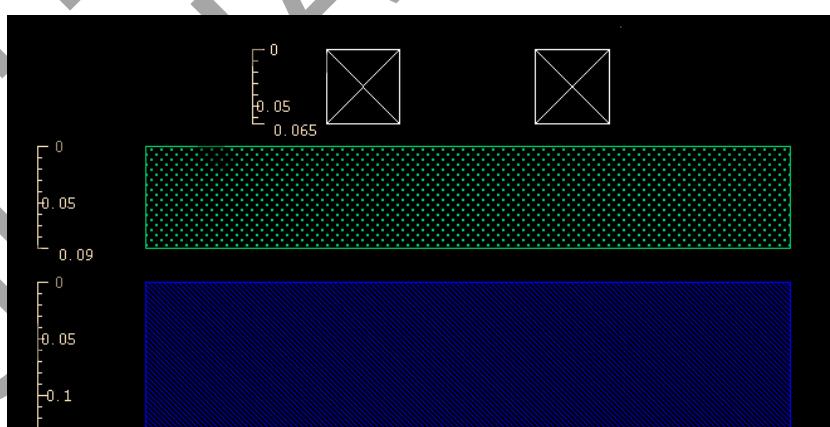
- Students connect the two G terminals of the NMOS and PMOS transistors using the POLY1 layer, connect the D terminal of the PMOS to the D terminal of the NMOS using the METAL1 layer, connect the S, and B terminal of the PMOS to the power rail (VDD), and connect the S, and B terminal of the NMOS to the ground (VSS) using the METAL1 layer.
- The **POLY** connecting the **gate terminals** of both PMOS and NMOS transistors must be **straight from top to bottom**, without any bends or corners. This design constraint aims to avoid the phenomenon of **IR drop**, which refers to a voltage drop occurring at each bend.

The whole inverter layout is shown as follows.

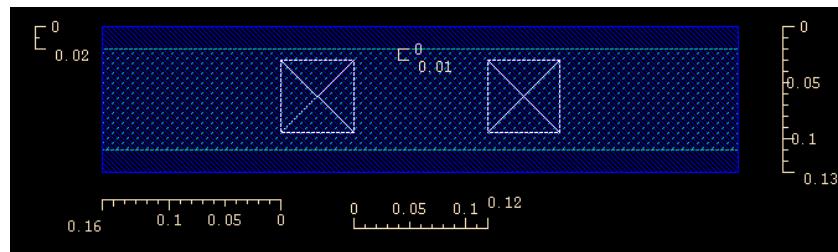
**Figure 131** Create connections to complete inverter layout.

In more complex designs, it is advisable to introduce an additional **METAL2 layer**. This choice is driven by the fact that the parasitic capacitance between the **POLY1** layer and **METAL2** layer is smaller than that between the **POLY1** layer and **METAL1** layer. As a result, incorporating the **METAL2** layer reduces parasitic effects in the circuit, leading to more accurate results during **post-layout simulation**. Students can observe the 3D planar inverter to understand the inverter layout that has just been made easily, please access this link: [planar inverter](#).

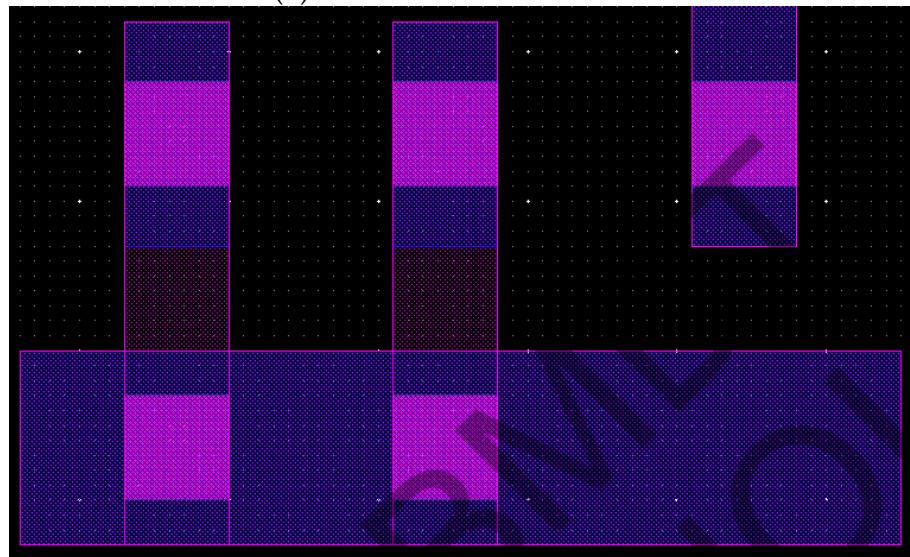
Students will create two types of connection between physical layers: METAL1 contact to active, METAL2 contact to METAL1.



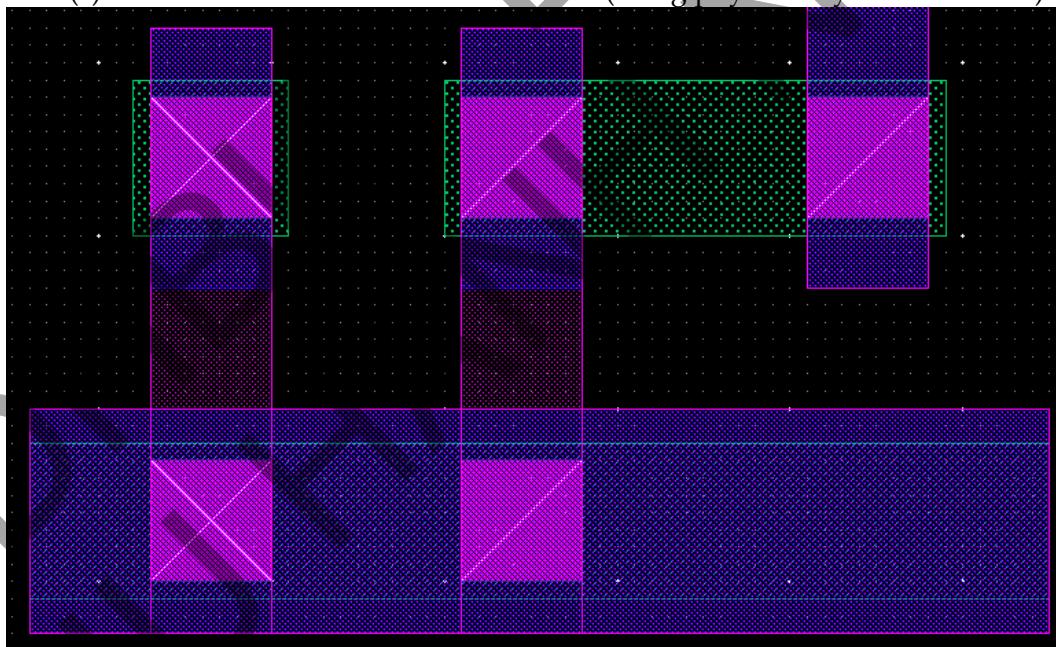
(a) Prepare creating connection between METAL1 and active



(b) METAL1 contact to active



(c) VIA between METAL1 and METAL2 (using physical layer named via1).



(d) VIAs are used in inverter layout.

Figure 132 Create power rail.

The complete layout is shown in the figure below.

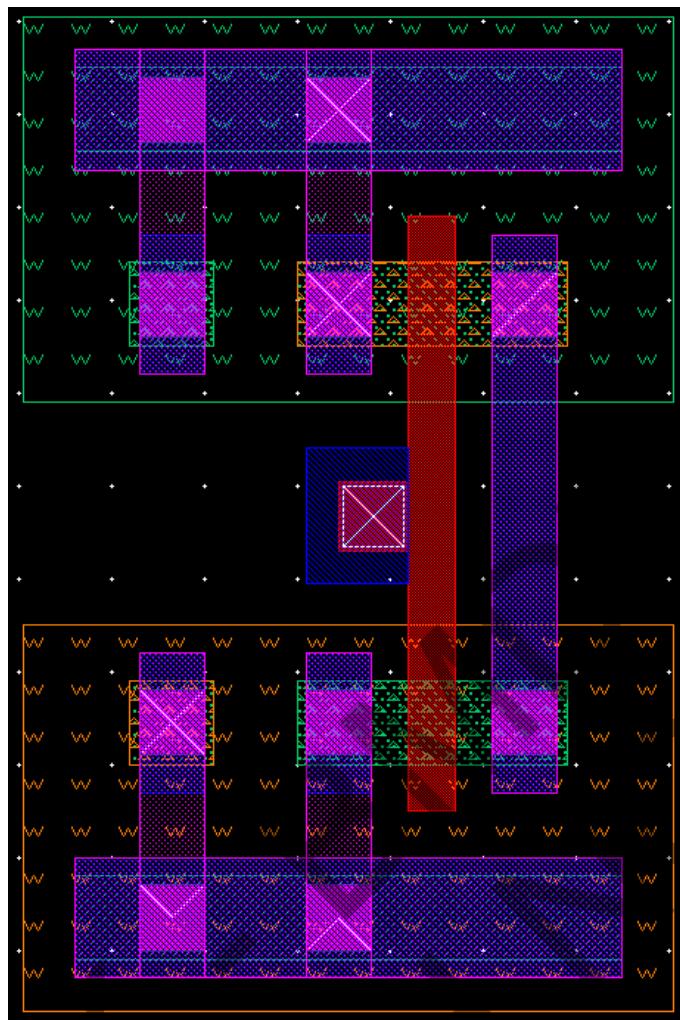
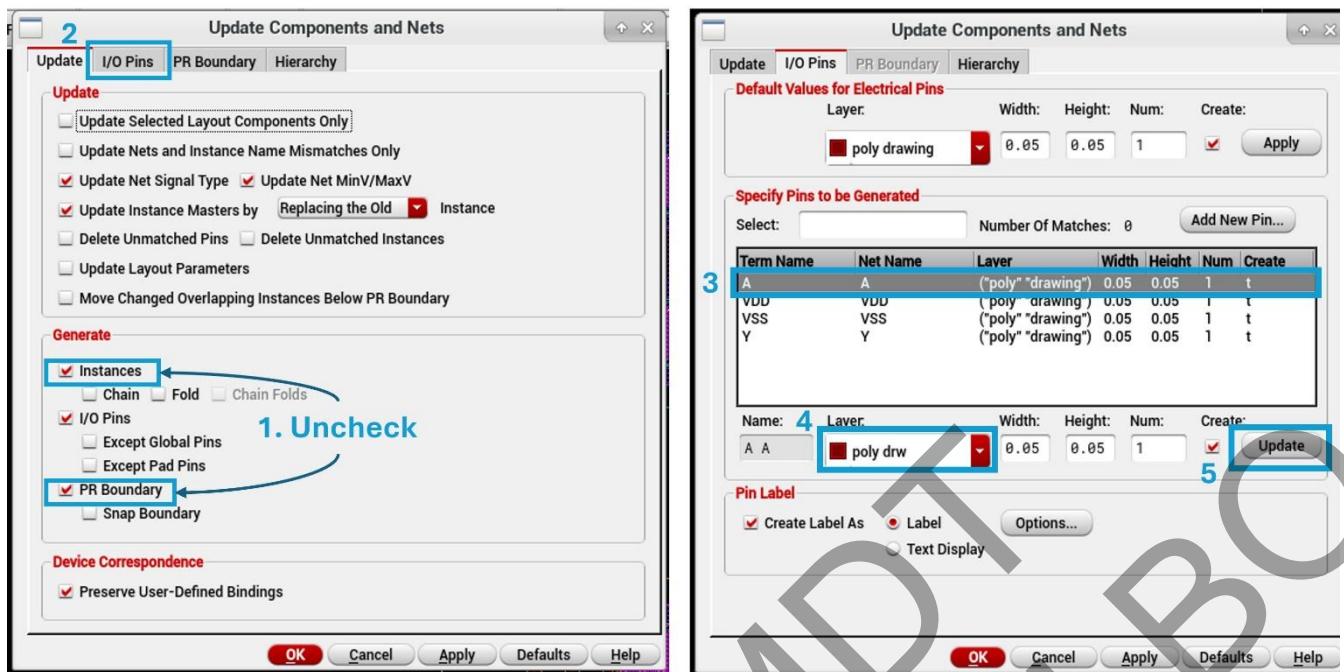


Figure 133 The complete inverter layout (without pins).

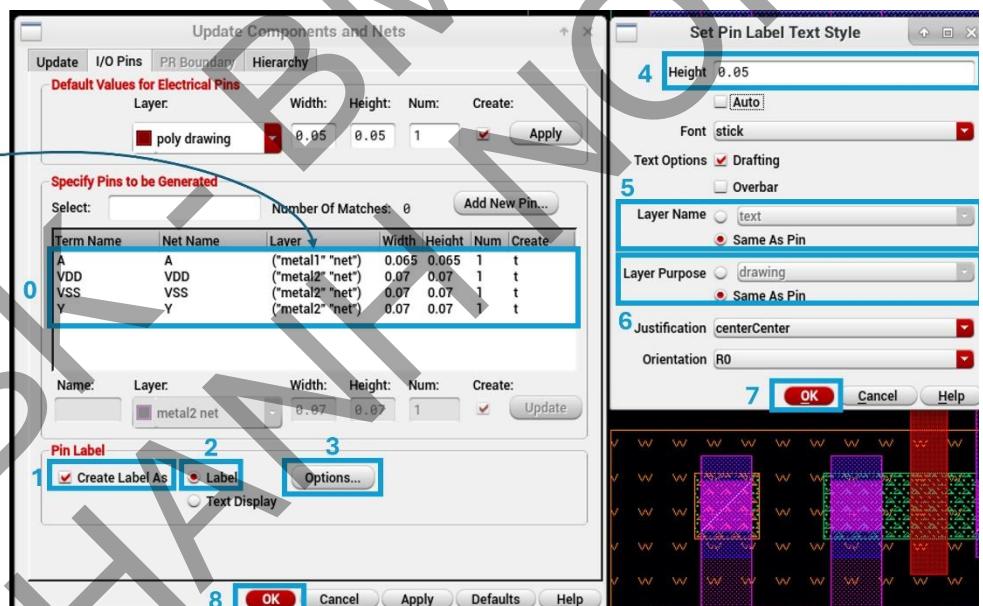
Then rerun DRC. There should be no error, and vice versa, feel free to fix those errors.

1.5. Step 5: Put I/O Pins, VDD, and GND

Click [Connectivity > Update > Components and Nets](#) to bring out the **Update Components and Nets** dialog box.

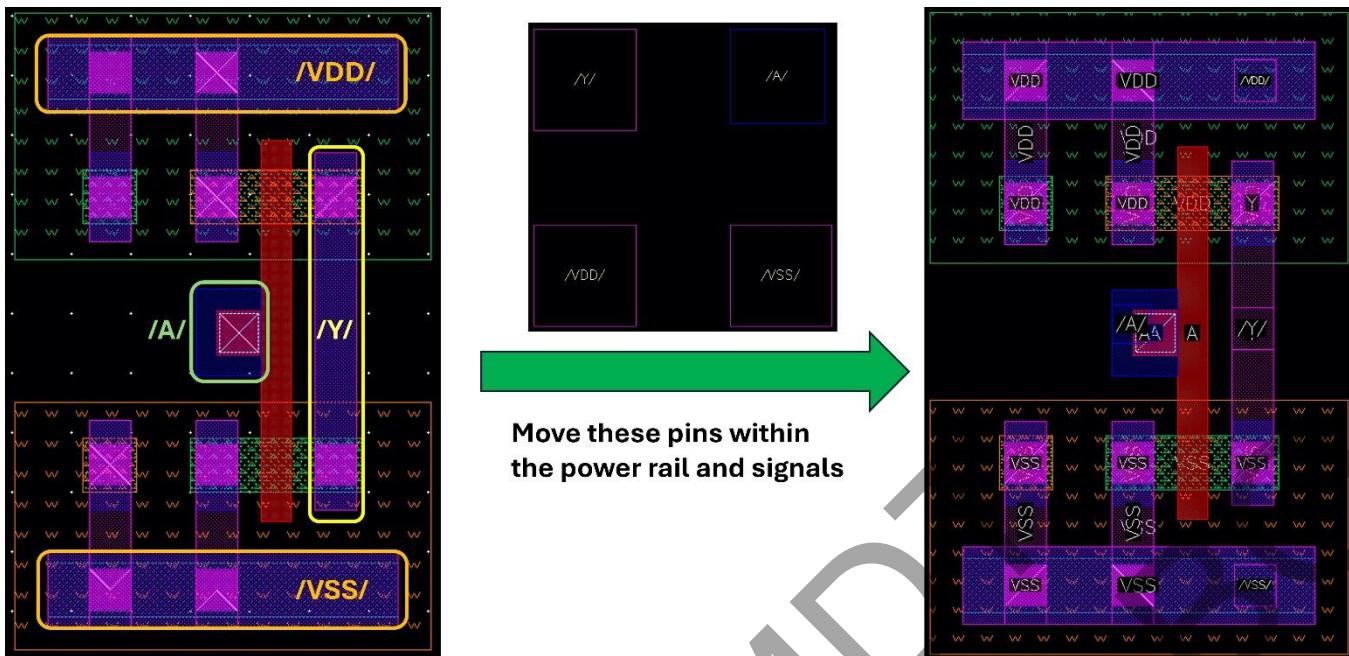


(a)

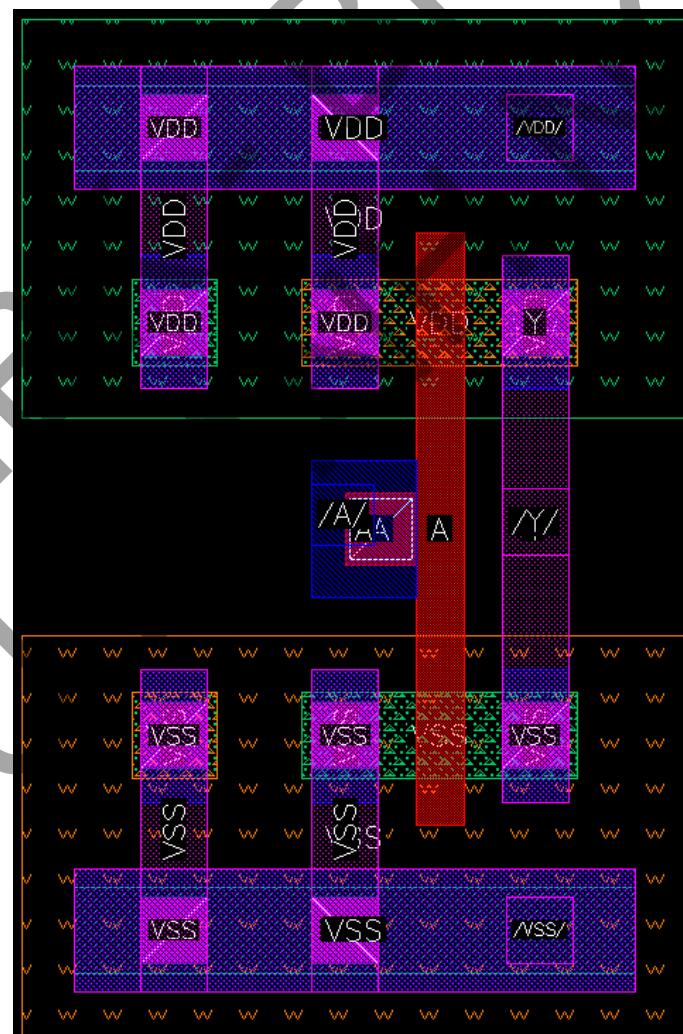


(b)

Figure 134 Settings for generating pins in the layout of the inverter.

**Figure 135** Move pins to the layout.

The inverter layout is complete. Please remember to save after any updates.

**Figure 136** The complete layout.

2) Check LVS.

While DRC ensures there will be no violation of layout rules, it does not guarantee students' layout has the correct functionality. In other words, we need to verify students' inverter layout will work as an inverter. This is done by using the layout-vs-schematic (LVS) tool that is also part of Calibre. The LVS tool will take a schematic and a layout as the inputs and output whether the two are constructed with the same set of transistors and interconnections.

Perform in Virtuoso Layout Suite XL: Calibre > Run mmDRC

Because this is the first DRC run, students can choose "Cancel" when asked about the "Runset File." A runset file will save all our LVS run settings.

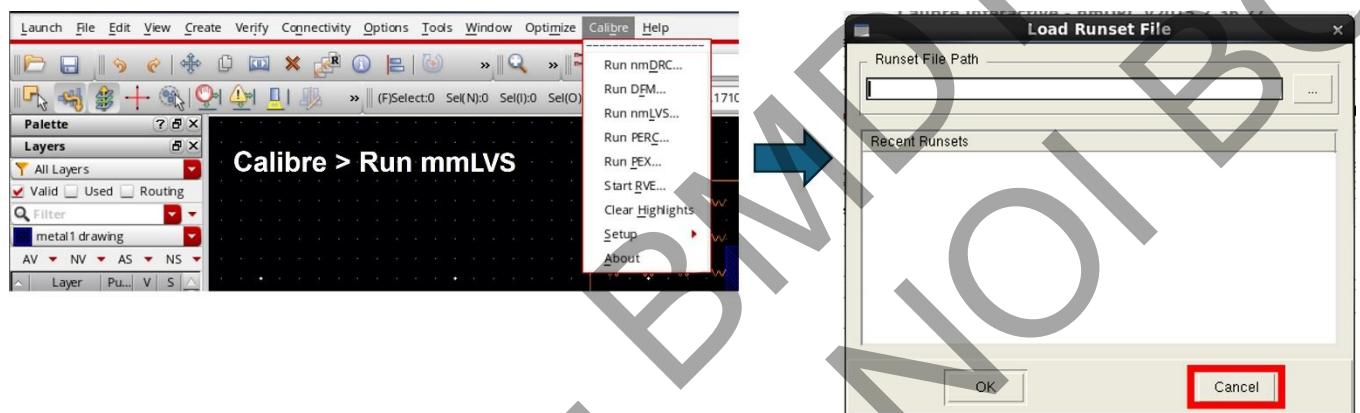


Figure 137 Open Calibre for running LVS.

At the left of the window, choose "Rules" and put the directory including the LVS Rules File. This step looks like this:

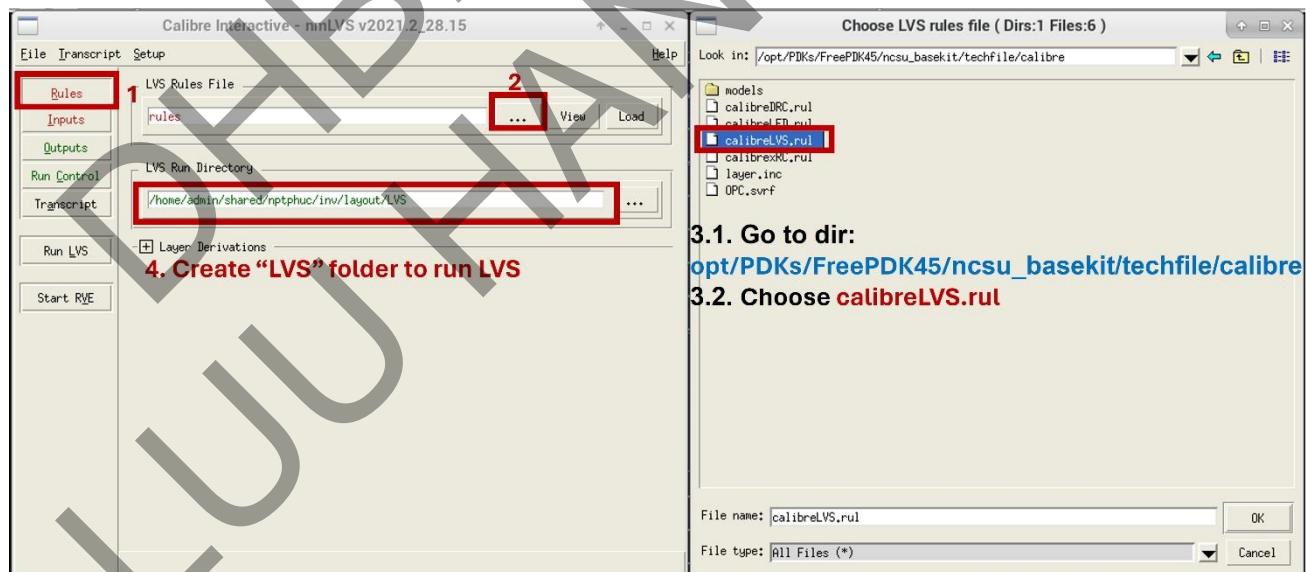


Figure 138 Choose the LVS rules file to run LVS.

Next, in the "Inputs" tab, please check "Export from layout viewer". Switch to the 'Netlist' tab. This is where the schematic should come from. Make sure it is 'Export from schematic viewer'. Students may

save some LVS running time later by unchecking that option if students know the file 'inv.src.net' is up to date.

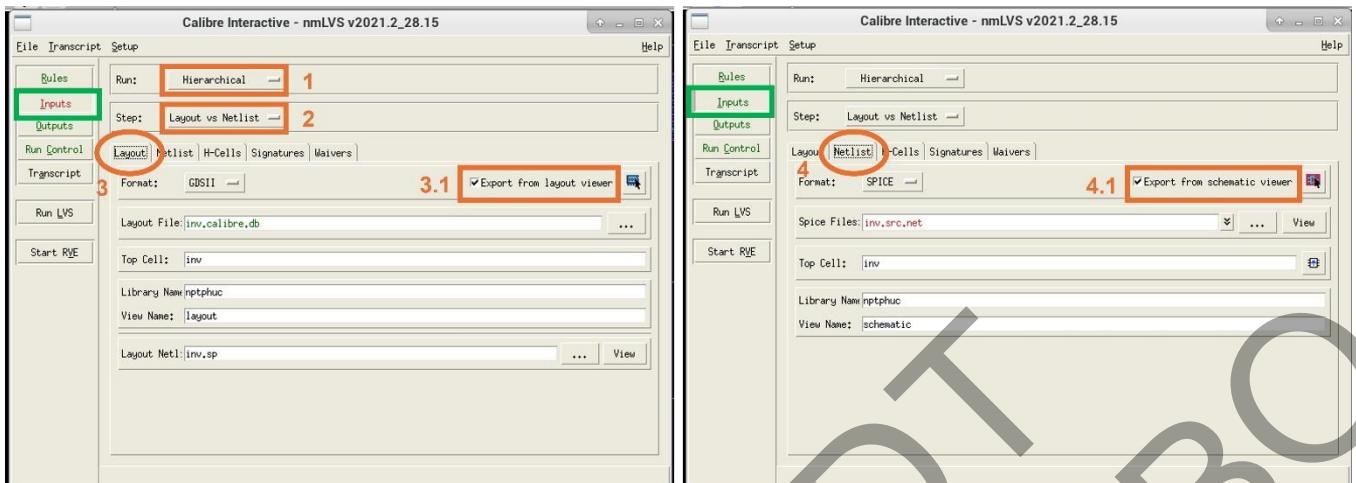


Figure 139 Setting inputs before running LVS.

The setting in the “Outputs” tab is shown:

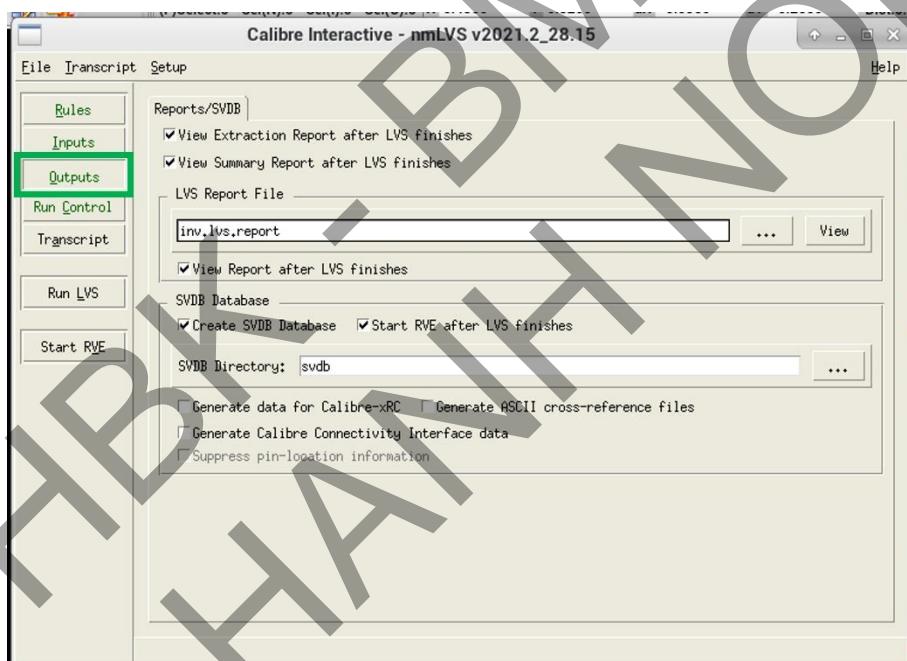


Figure 140 Setting outputs before running LVS.

Now that LVS (Layout vs. Schematic) has been successfully set, select "Run LVS" on the left. If any pop-up windows appear, check "Don't show this dialog again" and select OK.

If everything is correct, we should see a Check Mark and a Happy Face, like the image below. When LVS completes running, the Calibre-RVE window will appear as an LVS Report file.

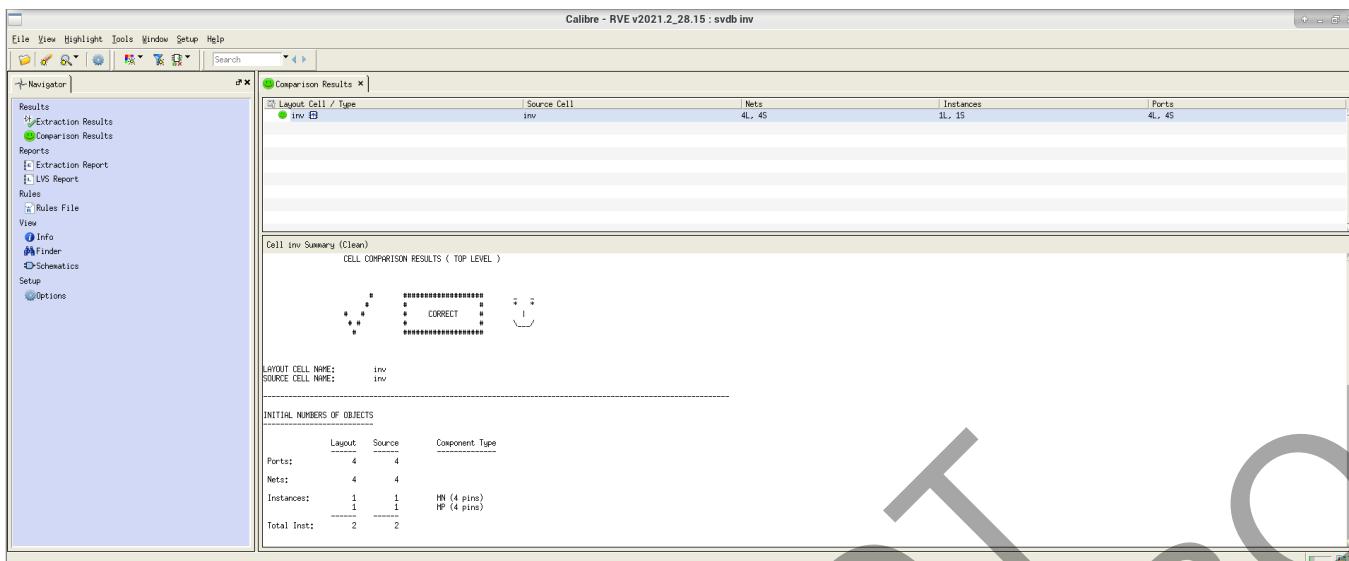


Figure 141 LVS report with happy face.

Common errors include errors caused by the Width (W) and Length (L) of the NMOS and PMOS transistors in our layout not matching the library, so we need to go back to the schematic and adjust W and L to match.

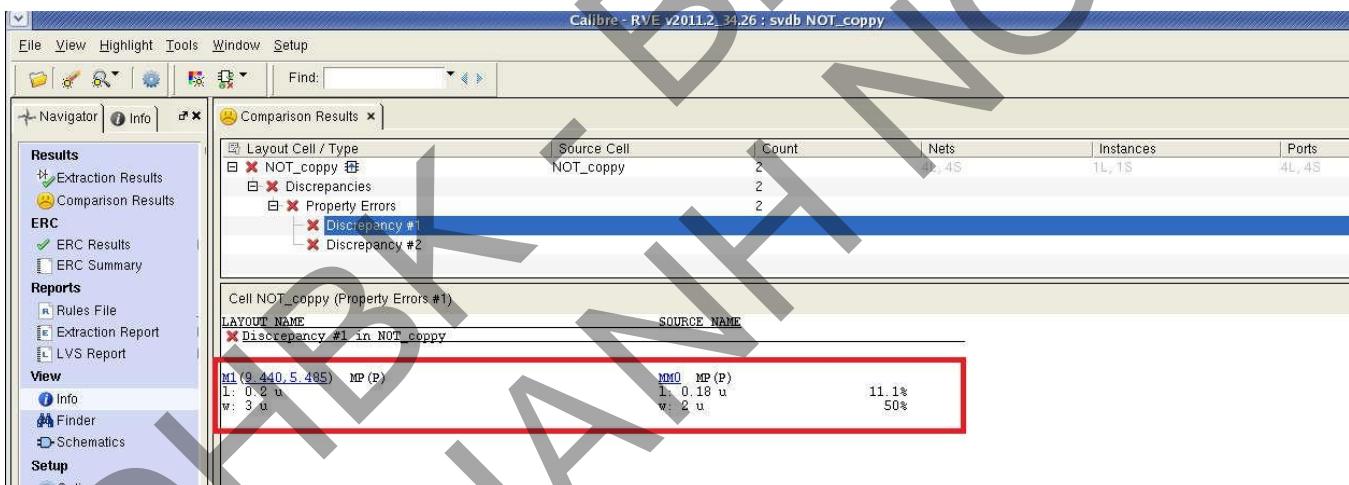


Figure 142 LVS report with discrepancy error, i.e. device's size is not match.

Select "Schematics" on the left to view the schematic image of the layout just created and the original schematic.

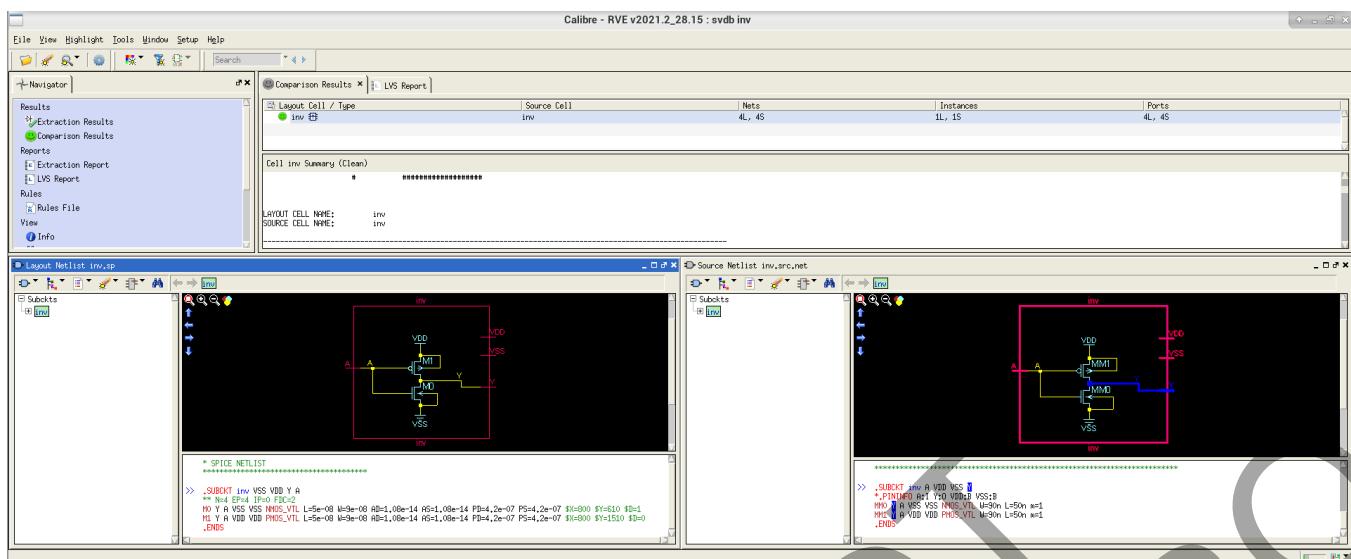


Figure 143 Schematic image of the layout and netlist file – INV.sp

3) Check PEX.

This section describes how to extract a netlist from students' layout that include parasitic resistances and capacitances. Students will then be able to re-simulate their design with extracted parasitic. Before starting, we must ensure that Calibre LVS has been run correctly with 0 errors and 0 warnings. Any warning at this stage may lead to issues during the subsequent Postlastudentst simulation.

To run Calibre PEX, select the **Calibre > Run PEX**.

Since this is the first PEX run, students can choose "cancel" when asked about the "runset file." A runset file saves all our PEX run settings. After saving the runset file, students can choose OK in subsequent runs. On the PEX form, go to the Rules tab, and next to PEX Rules File, click on "...". Browse to the folder containing the Rules file.

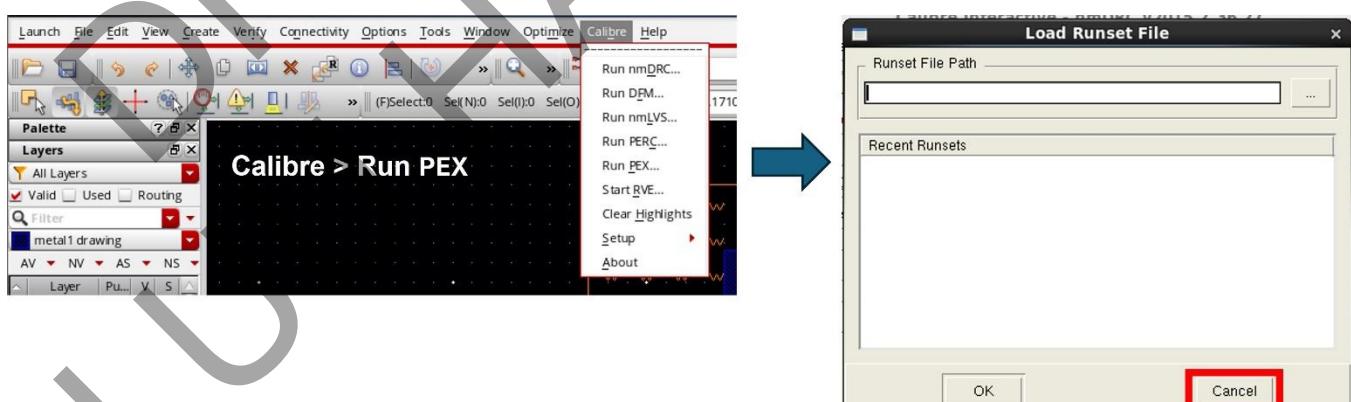
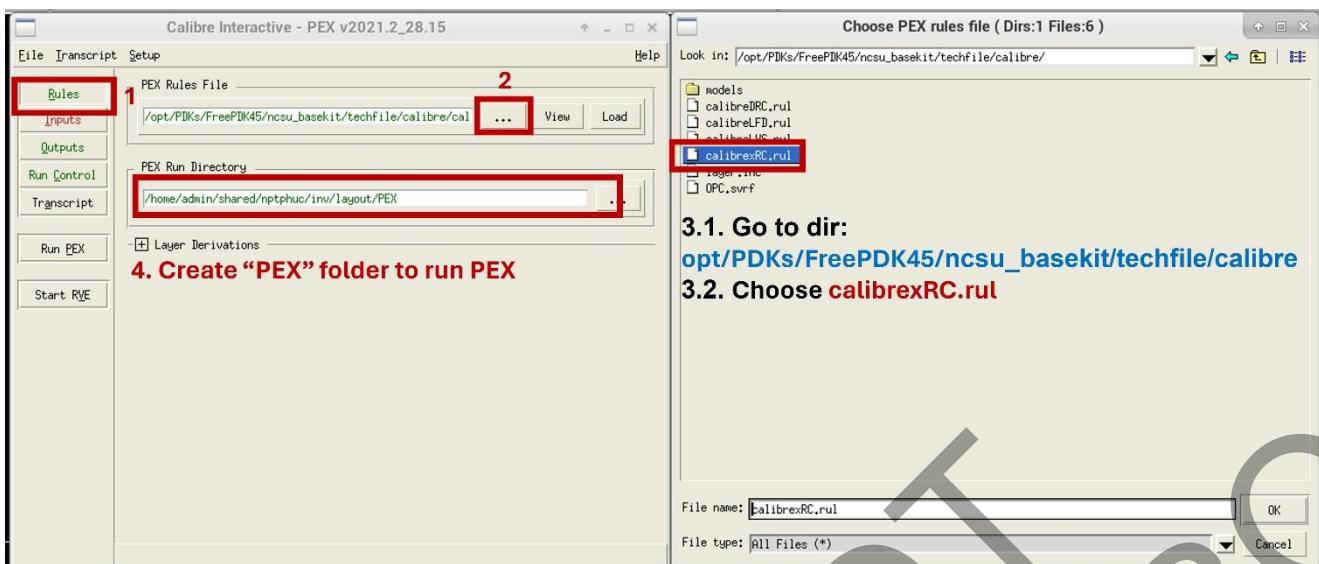


Figure 144 Open Calibre for running PEX.

Please put a directory including RC rules and choose a run directory:



Next, in the “Inputs” tab, please check “Export from layout viewer”. Go to the Output tab, change the Format from "ELDO" to “CALIBREVIEW” Change “Use Names From” to “SCHEMATIC.” Set the Extraction Type to "R+C+CC" to extract both parasitic resistances and parasitic capacitances.

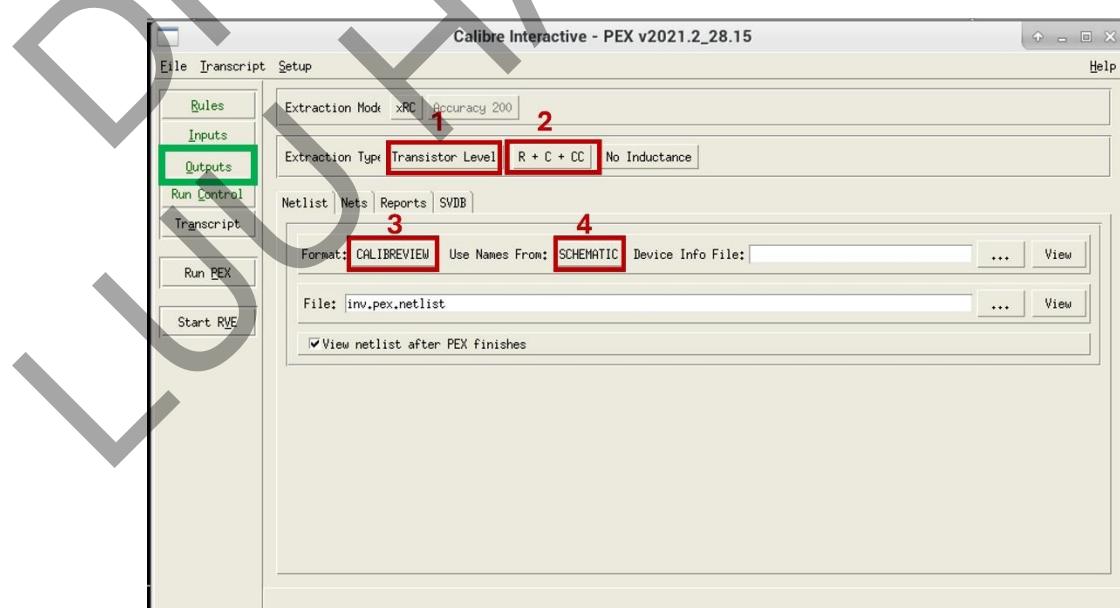
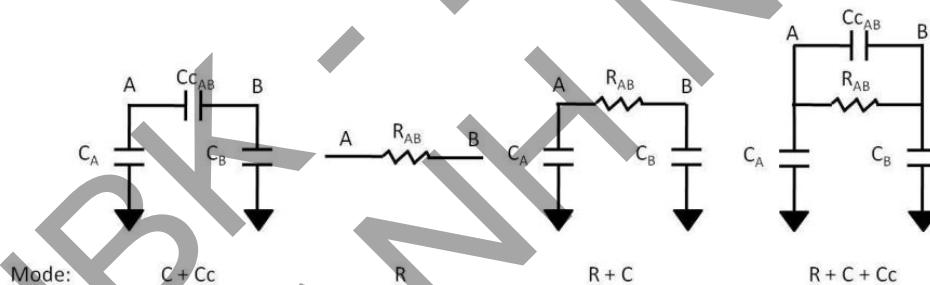


Figure 147 Setting outputs before running PEX.

Typically, students will work at the **transistor level** without extracting **inductance**. However, different forms of **RC extraction** are commonly used depending on the type of simulation being performed. Consider two nodes, **A** and **B**. The figure below illustrates the extracted **resistances (R)** and **capacitances (C, Cc)** for different extraction options:

- **R:** Extracts resistances between all nodes.
- **C:** Extracts a **lumped parasitic capacitance** to ground. This means all parasitic capacitances on a node are summed together and connected to **ground**, simplifying the extracted netlist. This method is useful when many **DC bias lines (AC ground)** exist around a node, and students are primarily interested in the effect of capacitance.
- **Cc:** Extracts **coupling capacitances** between every pair of nodes, making it the most detailed and complex extraction method.

PEX (Parasitic Extraction) provides additional options, such as extracting parasitics for selected nodes instead of all nodes and modifying the **lumped reference node** (e.g., connecting CA and CB to a different node instead of ground).

**Figure 148** Different types of RC extraction.

Now, students can save all the settings by selecting File > Save Runset. Give it a name, such as "PEX_cal," and click OK in the next window. This will allow us to reuse the same settings for the next PEX run by loading this runset file.

Select Run PEX. Then "Start RVE"

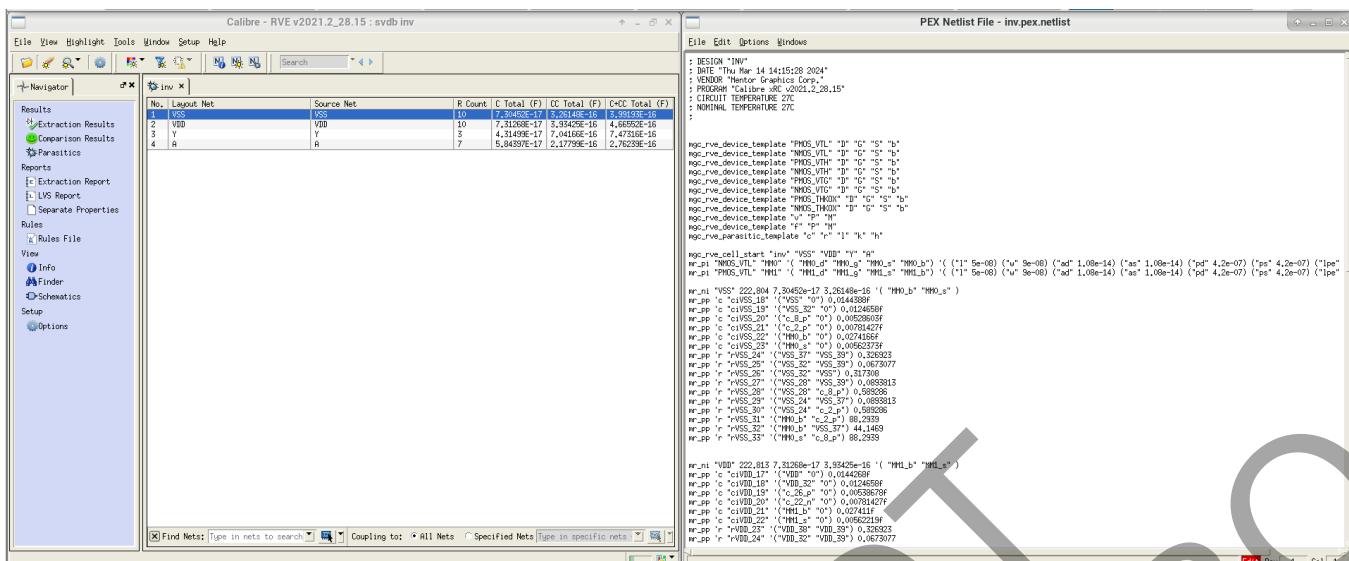


Figure 149 The PEX netlist and RVE result.

Students can change the output format to HSPICE, and then use the command **vimdiff** to observe the difference before and after extracting parasitic capacitances.

```

File Edit View Terminal Tabs Help
* SPICE NETLIST
*****
*.SUBCKT inv VSS VDD Y A
** N=4 EP=4 IP=0 FDC=2
M0 Y A VSS VDD NMOS_VTL L=5e-08 W=9e-08 AD=1.08e-14 AS=1.08e-14 PD=4.2e-07 PS=4.2e-07 SX=800
M1 Y A VDD VDD PMOS_VTL L=5e-08 W=9e-08 AD=1.08e-14 AS=1.08e-14 PD=4.2e-07 PS=4.2e-07 SX=800
.ENDS
*****

```

```

File: inv.pex.netlist
Created: Fri Mar 15 21:10:27 2024
Program: Calibre xRC
Version "v2021.2_28.15"
include "inv.pex.netlist.pex"
subckt inv VSS VDD Y A
*
* A A
* Y Y
* VDD VDD
* VSS VSS
M0 M_N_MM0_d N_A_MM0_g N_VSS_MM0_s N_VSS_MM0_b NMOS_VTL L=5e-08 W=9e-08
+ AD=1.08e-14 AS=1.08e-14 PD=4.2e-07 PS=4.2e-07
M1 N_Y_MM1_d N_A_MM1_g N_VDD_MM1_s N_VDD_MM1_b PMOS_VTL L=5e-08 W=9e-08
+ AD=1.08e-14 AS=1.08e-14 PD=4.2e-07 PS=4.2e-07
*
.include "inv.pex.netlist.INV.pxi"
*
.ends
*

```

Figure 150 The netlist difference between before and after extracting parasitics capacitance.

```

File Edit View Terminal Tabs Help
* File: inv.pex.netlist.INV.pxi
* Created: Fri Mar 15 21:10:27 2024
*
x_PM_INV%VSS N_VSS_MM0_s N_VSS_MM0_b N_VSS_c_2_p N_VSS_c_8_p VSS PM_INV%VSS
x_PM_INV%VDD N_VDD_MM1_s N_VDD_MM1_b N_VDD_c_22_n N_VDD_c_26_p VDD PM_INV%VDD
x_PM_INV%Y N_Y_MM0_d N_Y_MM1_d N_Y_c_39_n Y PM_INV%Y
x_PM_INV%A N_A_MM0_g N_A_MM1_g A N_A_c_54_n PM_INV%A
cc_1 N_VSS_MM0_b N_VDD_MM1_b 0.00103849f
cc_2 N_VSS_c_2_p N_VDD_MM1_b 4.11781e-19
cc_3 VSS N_VDD_MM1_b 2.6378e-19
cc_4 N_VSS_MM0_b N_VDD_c_22_n 4.11781e-19
cc_5 N_VSS_MM0_b VDD 2.63664e-19
cc_6 N_VSS_MM0_s N_Y_MM0_d 0.0335003f
cc_7 N_VSS_MM0_b N_Y_MM0_d 0.215948f
cc_8 N_VSS_c_8_p N_Y_MM0_d 3.93856e-19
cc_9 N_VSS_MM0_s N_Y_c_39_n 2.28114e-19
cc_10 N_VSS_c_8_p N_Y_c_39_n 0.00556577f
cc_11 VSS N_Y_c_39_n 0.00456939f
cc_12 N_VSS_MM0_s N_A_MM0_g 0.00237285f
cc_13 N_VSS_MM0_b N_A_MM0_g 0.001387f
cc_14 N_VSS_c_8_p N_A_MM0_g 2.80056e-19
cc_15 N_VSS_MM0_s N_A_c_54_n 0.057933f
cc_16 N_VSS_MM0_b N_A_c_54_n 6.4178e-19
cc_17 N_VSS_c_8_p N_A_c_54_n 6.82041e-19
cc_18 VSS N_A_c_54_n 2.56383e-19
cc_19 N_VDD_MM1_s N_Y_MM0_d 0.0335003f
cc_20 N_VDD_MM1_b N_Y_MM0_d 0.295564f
cc_21 N_VDD_c_26_p N_Y_MM0_d 4.35292e-19
cc_22 N_VDD_MM1_s N_Y_c_39_n 2.28114e-19
cc_23 N_VDD_c_26_p N_Y_c_39_n 0.00575651f
cc_24 VDD N_Y_c_39_n 0.00489139f
cc_25 N_VDD_MM1_s N_A_MM1_g 0.00237285f
cc_26 N_VDD_MM1_b N_A_MM1_g 0.00135841f
cc_27 N_VDD_MM1_s N_A_c_54_n 0.0453727f
cc_28 N_VDD_MM1_b N_A_c_54_n 6.24917e-19
cc_29 N_VDD_c_26_p N_A_c_54_n 6.77238e-19
cc_30 VDD N_A_c_54_n 2.53972e-19
cc_31 N_Y_MM0_d N_A_MM0_g 0.0136266f
cc_32 N_Y_MM0_d N_A_c_54_n 0.088787f
cc_33 N_Y_c_39_n N_A_c_54_n 0.00117214f

```

Figure 151

Where

*****.sp** is the circuit netlist before extracting parasitic capacitance and resistance.

*****.pex** is the circuit netlist after extracting parasitic capacitance and resistance.

*****.pxi** is the connection relationship between the circuit netlist and the parasitic parameters.

PART 6

Objective: Post-layout simulation

Requirements: Re-measure some parameters of the inverter (timing parameters, power consumption) and compare post-layout simulation results to pre-layout simulation results.

Instruction:

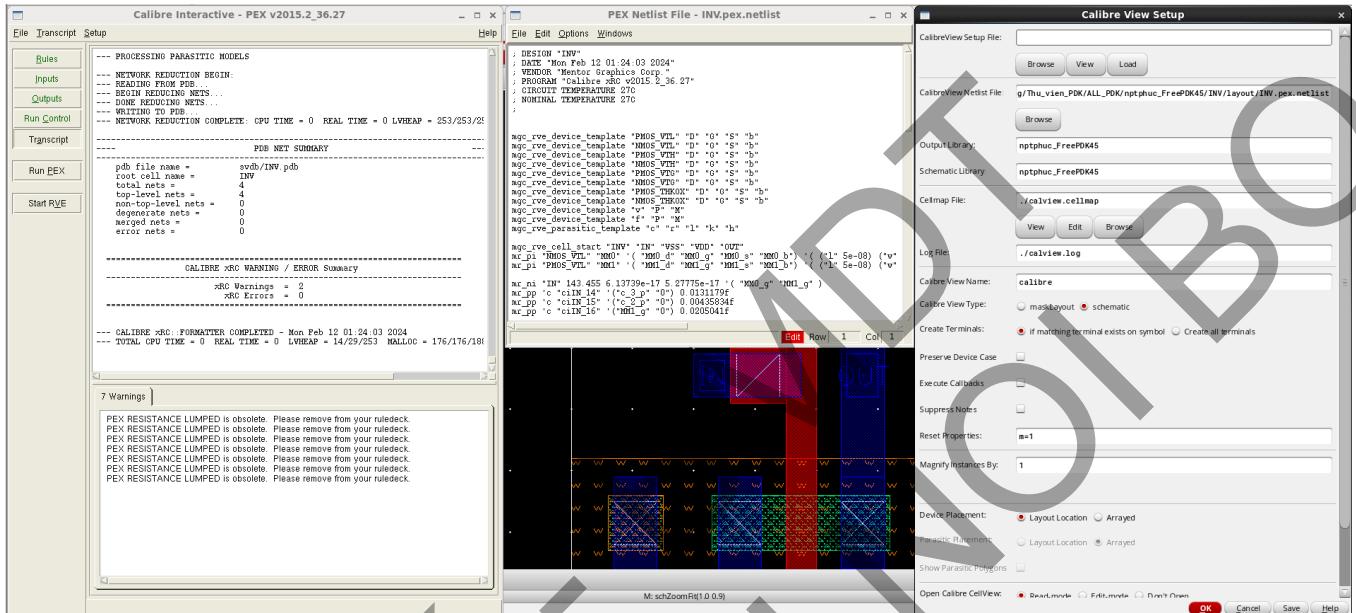


Figure 152 The Calibre View Setup window.

After running on the "Calibre View Setup" form that pops up:

- Make sure that students 'Cellmap File' points to the directory.
- Change "Calibre View Type" to "schematic"
- Change "Create terminals" to "Create all terminals"
- Change "Device Placement" to "Arrayed"
- Change "Open Calibre Cellview" to "read-mode".

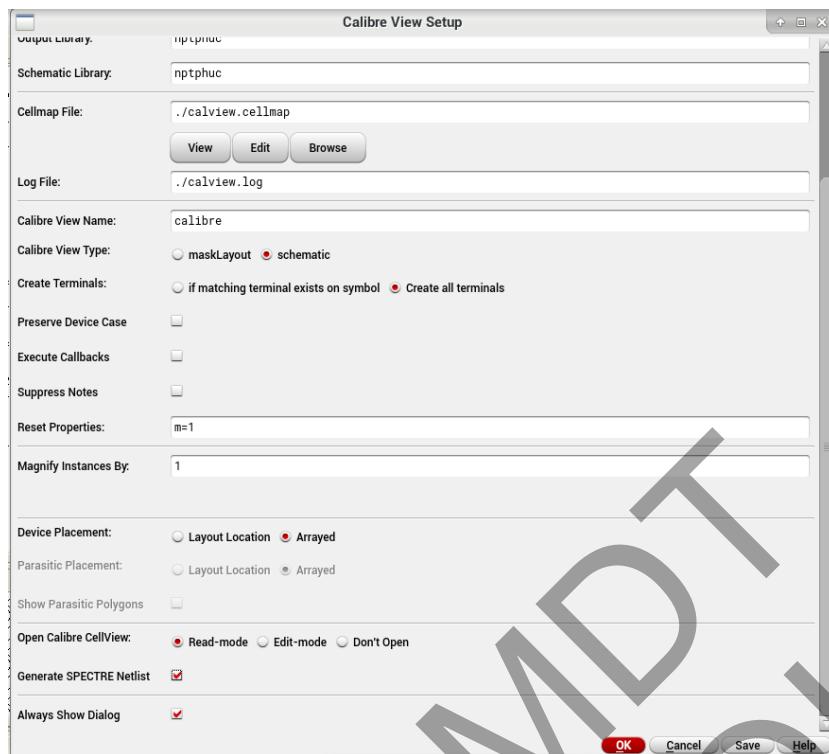
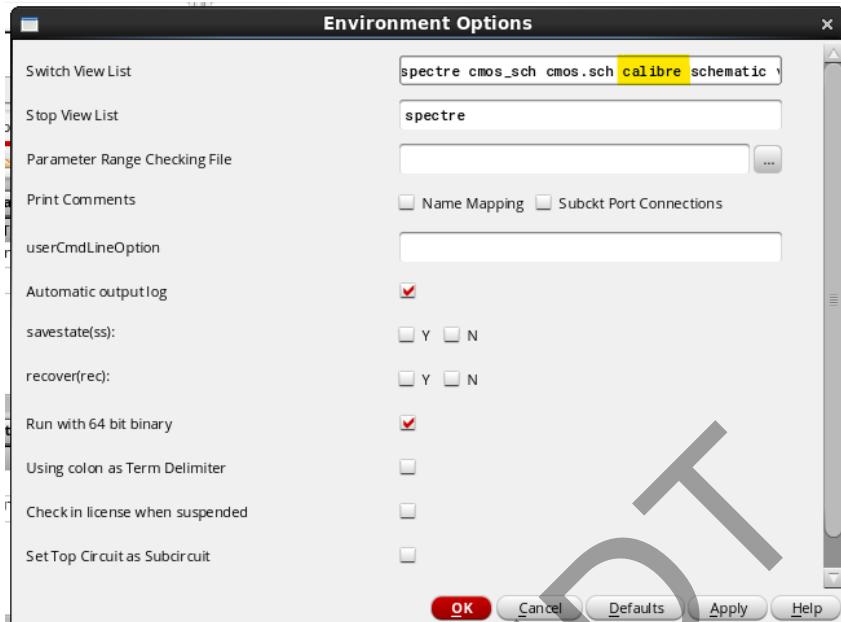


Figure 153 Calibre View Setup.

This will then create a "calibre" view Cadence that is a schematic in Virtuoso, from which students can simulate. Students can locate this together with students' schematic, layout, and symbol files in Library Manager.

Now it's time to simulate what we extracted. Open the students 'inverter_test' schematic again. Open student's spectre view by doing Launch -> ADE L. Load students **saved the state** from before which includes library data, temperature, simulation data, etc. If students have not saved students spectre environment, follow the link above to go through the settings again. After that, we need only change one thing to simulate our extraction rather than our schematic. Go to **Setup -> Environment** and at the topmost line, enter **calibre** before **schematic**, with a space separating each word. The window should look like this:



Finally, hit "Netlist and run". Display the student's input and output transient results, and students will see a very similar result from the schematic simulation results. It is because a simple inverter does not require complicated modeling. **Try comparing the two output waveforms and delay times.**

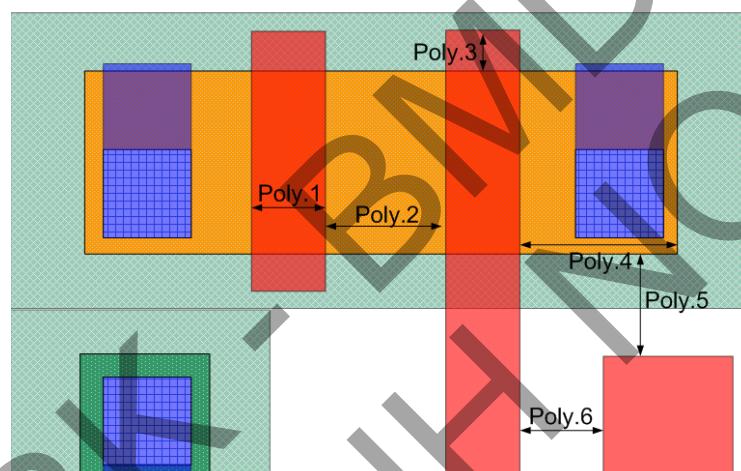
Students can also use **hspice** to run this simulation instead of CalibreView.

LUU HANH - DHBK

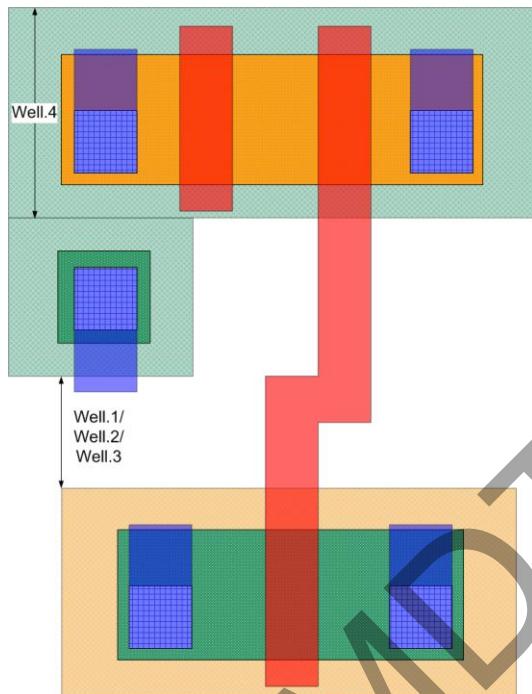
APPENDIX A:

DESIGN RULES OF FreePDK45

Rule	Value	Description
POLY RULES		
POLY.1	50 nm	Minimum width of poly
POLY.2	140 nm	Minimum spacing of poly AND active
POLY.3	55 nm	Minimum poly extension beyond active
POLY.4	70 nm	Minimum enclosure of active around gate
POLY.5	50 nm	Minimum spacing of field poly to active
POLY.6	75 nm	Minimum spacing of field poly
WELL RULES		
WELL.1	none	saveDerived: nwell/pwell must not overlap
WELL.2	225 nm	Minimum spacing of nwell/pwell at different potential
WELL.3	135 nm	Minimum spacing of nwell/pwell at the same potential
WELL.4	200 nm	Minimum width of nwell/pwell

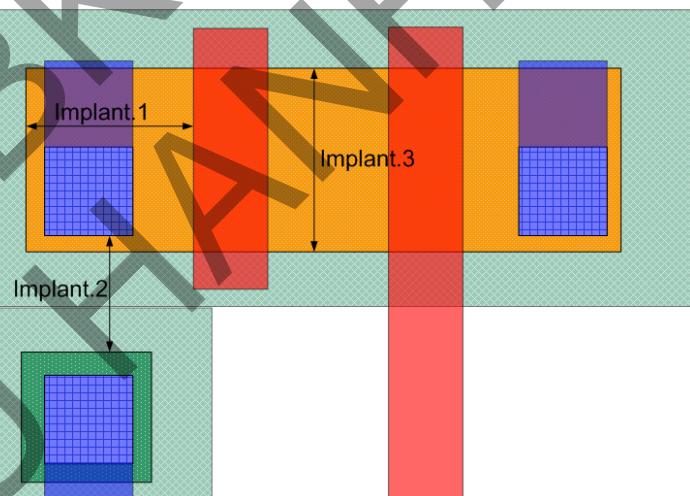


NOTE : POLY.5 rule is for poly lines which interact with any active regions, in other words, which are extensions of gate poly lines. If a poly line is used as a print assist feature (which does not interact with any active regions) then minimum spacing of field poly to active is 35nm which is stated in POLY.7 rule.



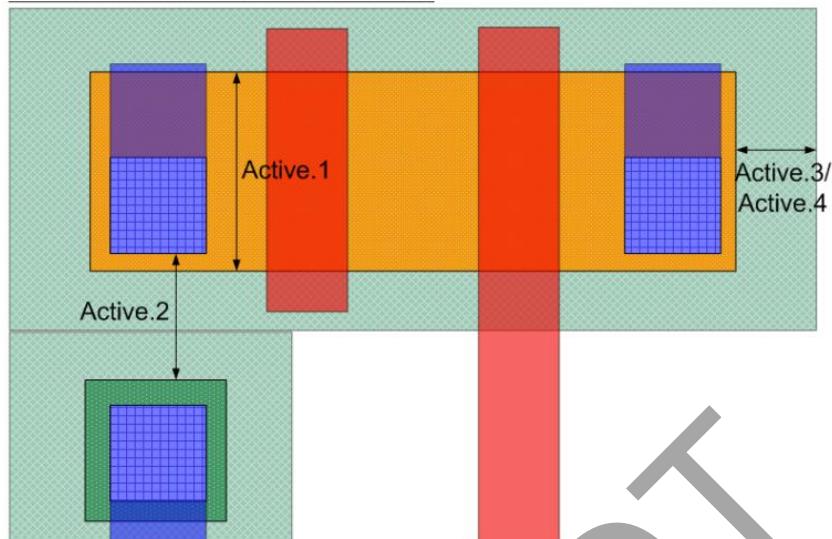
IMPLANT RULES

IMPLANT.1	70 nm	Minimum spacing of nimplant/ pimplant to channel
IMPLANT.2	25 nm	Minimum spacing of nimplant/ pimplant to contact
IMPLANT.3/4	45 nm	Minimum width/ spacing of nimplant/ pimplant
IMPLANT.5	none	Nimplant and implant must not overlap



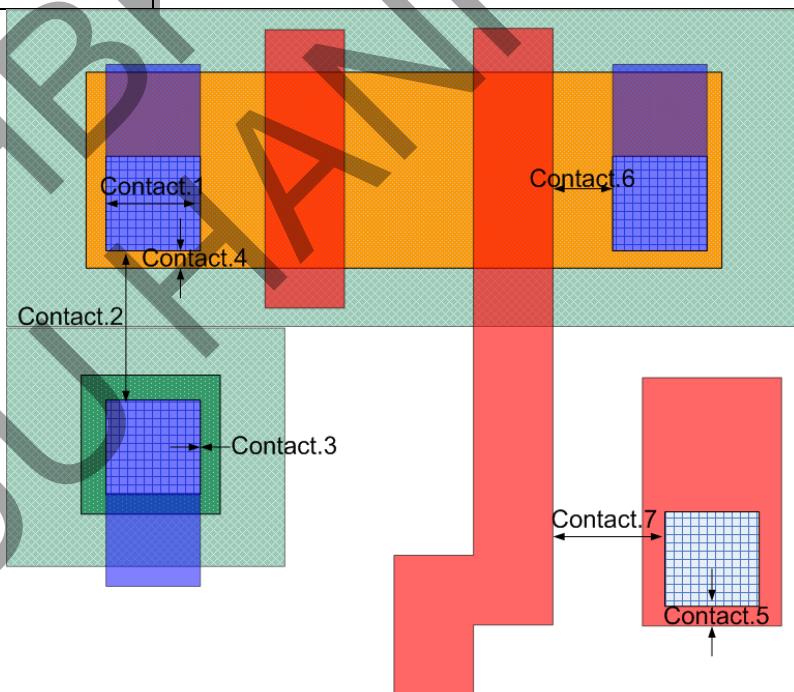
ACTIVE RULES

ACTIVE.1	90 nm	Minimum width of active
ACTIVE.2	80 nm	Minimum spacing of active
ACTIVE.3	55 nm	Minimum enclosure/spacing of nwell/pwell to active
ACTIVE.4	none	saveDerived: active must be inside nwell or pwell



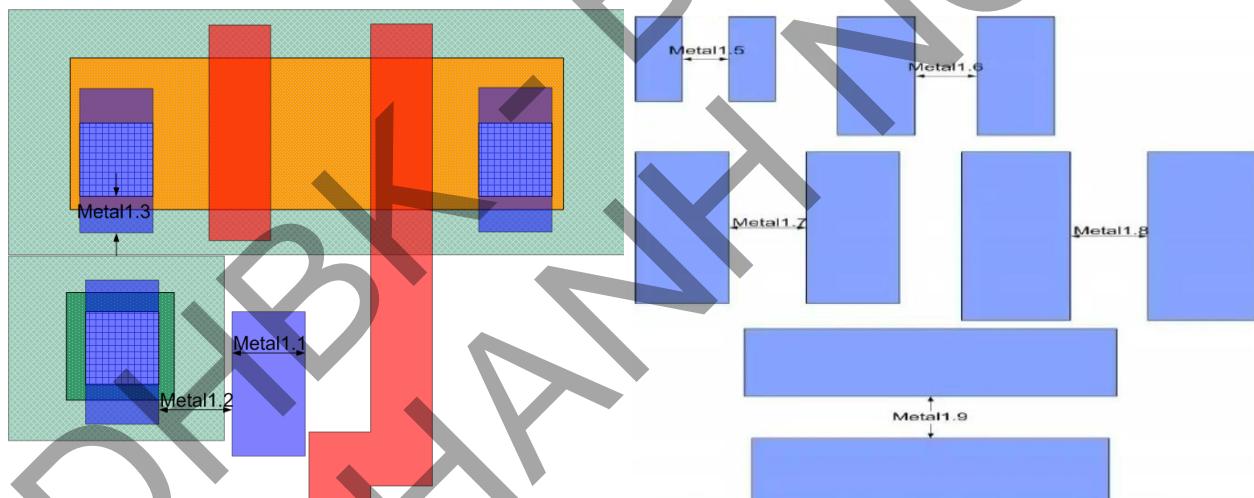
CONTACT RULES

CONTACT.1	65 nm	Minimum width of contact
CONTACT.2	75 nm	Minimum spacing of contact
CONTACT.3	none	saveDerived: contact must be inside active or poly or metal1
CONTACT.4	5 nm	Minimum enclosure of active around contact
CONTACT.5	5 nm	Minimum enclosure of poly around contact
CONTACT.6	35 nm	Minimum spacing of contact and gate
CONTACT.7	90 nm	Minimum spacing of contact and poly



METAL1 RULES

METAL1.1	65 nm	Minimum width of metal1
METAL1.2	65 nm	Minimum spacing of metal1
METAL1.3	35 nm	Minimum enclosure around contact on two opposite sides
METAL1.4	35 nm	Minimum enclosure around via1 on two opposite sides
METAL1.5	90 nm	Minimum spacing of metal wider than 90 nm and longer than 900 nm
METAL1.6	270 nm	Minimum spacing of metal wider than 270 nm and longer than 300 nm
METAL1.7	500 nm	Minimum spacing of metal wider than 500 nm and longer than 1.8um
METAL1.8	900 nm	Minimum spacing of metal wider than 900 nm and longer than 2.7 um
METAL1.9	1500 nm	Minimum spacing of metal wider than 1500 nm and longer than 4.0 um

**VIA1 RULES**

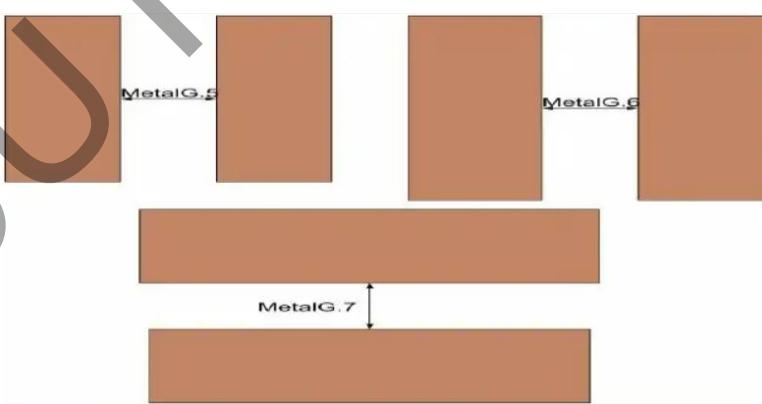
VIA1.1	65 nm	Minimum width of via1
VIA1.2	75 nm	Minimum spacing of via1
VIA1.3	none	saveDerived: via1 must be inside metal1
VIA1.4	none	saveDerived: via1 must be inside metal2

METALLINT RULES

METALLINT.1	70 nm	Minimum width of intermediate metal
METALLINT.2	70 nm	Minimum spacing of intermediate metal

METALINT.3	35 nm	Minimum enclosure around via1 on two opposite sides
METALINT.4	35 nm	Minimum enclosure around via[2-3] on two opposite sides
METALINT.5	90 nm	Minimum spacing of metal wider than 90 nm and longer than 900 nm
METALINT.6	270 nm	Minimum spacing of metal wider than 270 nm and longer than 300 nm
METALINT.7	500 nm	Minimum spacing of metal wider than 500 nm and longer than 1.8um
METALINT.8	900 nm	Minimum spacing of metal wider than 900 nm and longer than 2.7 um
METALINT.9	1500 nm	Minimum spacing of metal wider than 1500 nm and longer than 4.0 um
VIA2-3 RULES		
VIA[2-3].1	70 nm	Minimum width of Via[2-3]
VIA[2-3].2	85 nm	Minimum spacing of Via[2-3]
VIA[2-3].3	none	saveDerived: Via[2-3] must be inside metal[2-3]

VIA[2-3].4	none	saveDerived: Via[2-3] must be inside metal[3-4]
METALSMG RULES		
METALSMG.1	140 nm	Minimum width of semi-global metal
METALSMG.2	140 nm	Minimum spacing of semi-global metal
METALSMG.3	0 nm	Minimum enclosure around via[3-6] on two opposite sides
METALSMG.6	270 nm	Minimum spacing of metal wider than 270 nm and longer than 300 nm
METALSMG.7	500 nm	Minimum spacing of metal wider than 500 nm and longer than 1.8um
METALSMG.8	900 nm	Minimum spacing of metal wider than 900 nm and longer than 2.7 um
VIA[4-6] RULES		
VIA[4-6].1	140 nm	Minimum width of Via[4-6]
VIA[4-6].2	160nm	Minimum spacing of Via[4-6]
VIA[4-6].3	none	saveDerived: Via[4-6] must be inside metal[4-6]
VIA[4-6].4	none	saveDerived: Via[4-6] must be inside metal[5-7]

METALTNG RULES		
METALTNG.1	400 nm	Minimum width of metalTNG
METALTNG.2	400 nm	Minimum spacing of metalTNG
METALTNG.3	0 nm	Minimum enclosure around via[6-8] on two opposite sides
METALTNG.4	20000 nm^2	Minimum area of metalTNG straddling via[6-8]
METALTNG.5	90 nm	Minimum spacing of metal wider than 90 nm and longer than 900 nm
METALTNG.6	270 nm	Minimum spacing of metal wider than 270 nm and longer than 300 nm
VIA[7-8] RULES		
VIA[7-8].1	400 nm	Minimum width of via[7-8]
VIA[7-8].2	440 nm	Minimum spacing of via[7-8]
VIA[7-8].3	none	saveDerived: via[7-8] must be inside metal[7-8]
VIA[7-8].4	none	saveDerived: via[7-8] must be inside metal[8-9]
METALG RULES		
METALG.1	800 nm	Minimum width of global metal
METALG.2	800 nm	Minimum spacing of global metal
METALG.3	0 nm	Minimum enclosure around via[8-9] on two opposite sides
METALG.8	900 nm	Minimum spacing of metal wider than 900 nm and longer than 2.7 um
METALG.9	1500 nm	Minimum spacing of metal wider than 1500 nm and longer than 4.0 um
		
VIA9 RULES		

VIA[9].1	800	Minimum width of via9
VIA[9].2	880	Minimum spacing of via9
VIA[9].3	none	saveDerived: via9 must be inside metal9
VIA[9].4	none	saveDerived: via9 must be inside metal10
GRID RULES		
GRID.[1-26]	2.5nm	Shapes on all layers must be on a 2.5 nm grid
ATENNA RULES		
ANTENNA.1	300:1	Ratio of Maximum Allowed (Field poly area or Metal Layer Area) to transistor gate area

DHBK - LUU HANH NOI BO



APPENDIX B:
PHYSICAL LAYERS IN FREEPDK45

Name	Description	Color (R ,G ,B #Hex)
active	Active Area	0,204,102 #00CC66
nwell	n-type well implant	0,204,102 #00CC66
pwell	p-type well implant	255,128,0 #FF8000
nimplant	n-type source/drain doping	0,204,102 #00CC66
pimplant	p-type source/drain doping	255,128,0 #FF8000
sblock	Salicide block	0,0,255 #0000FF
vthl	Low threshold implant	
vthg	General use threshold implant	
vthh	High threshold implant	
thkox	Thick oxide	
poly	Poly-silicon	255,0,0 #FF0000
contact	Metal 1 contact to poly or active	128,38,38 #802626
metal1	Metal 1	0,0,255 #0000FF
via1	Metal 2 contact Metal 1	–
metal2	Metal 2 (intermediate wires)	255,,0,255 #FF00FF
via2	Metal 3 contact Metal 2	–
metal3	Metal 3 (intermediate wiring)	0,255,255 #00FFFF
via3	Metal 4 contact Metal 3	–
metal4	Metal 4 (semi-global wiring)	255,255,204 #FFFFCC
via4	Metal 5 contact Metal 4	–
metal5	Metal 5 (semi-global wiring)	57,191,255 #39BFFF
via5	Metal 6 contact to Metal 5	–
metal6	Metal 6 (semi-global wiring)	217,204,0 #D9CC00
via6	Metal 7 contact Metal 6	–
metal7	Metal 7 (thin-global wiring)	–
via7	Metal 8 contact to Metal 7	–



metal8	Metal 8 (thin-global wiring)	-
via8	Metal 9 contact to Metal 8	-
metal9	Metal 9 (global wiring)	-
via9	Metal 10 contact to Metal 9	-
metal10	Metal 10 (global wiring)	-

DHBK - BMNDT
LƯU HÀNH NỘI BO



APPENDIX C:

SINGLE VIA, VIA ARRAY, STACKED VIA

A VIA forms a connection between overlapping geometries on different layers through a cut layer, and it is formed by geometries on all three layers. Three types of VIAs:

A single VIA;

An array VIA;

A stacked VIA.

1. Single VIA:

The below diagram helps students to understand how single VIA are placed between two metals and helps them to connect them.

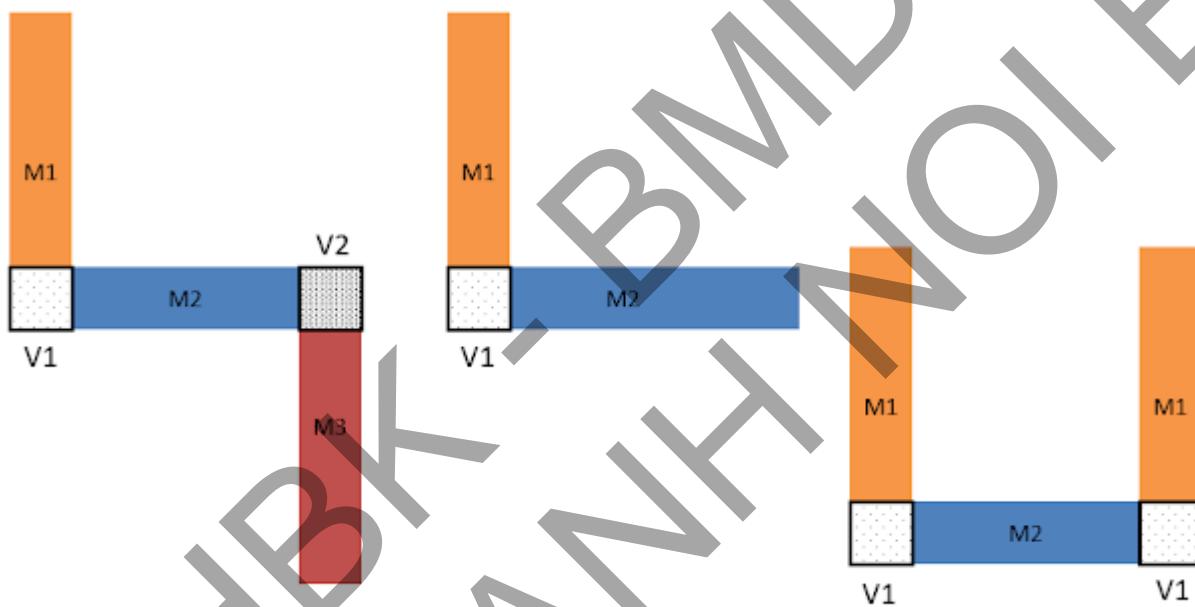


Figure 154 2D view/Top view with different arrangements (layout view).

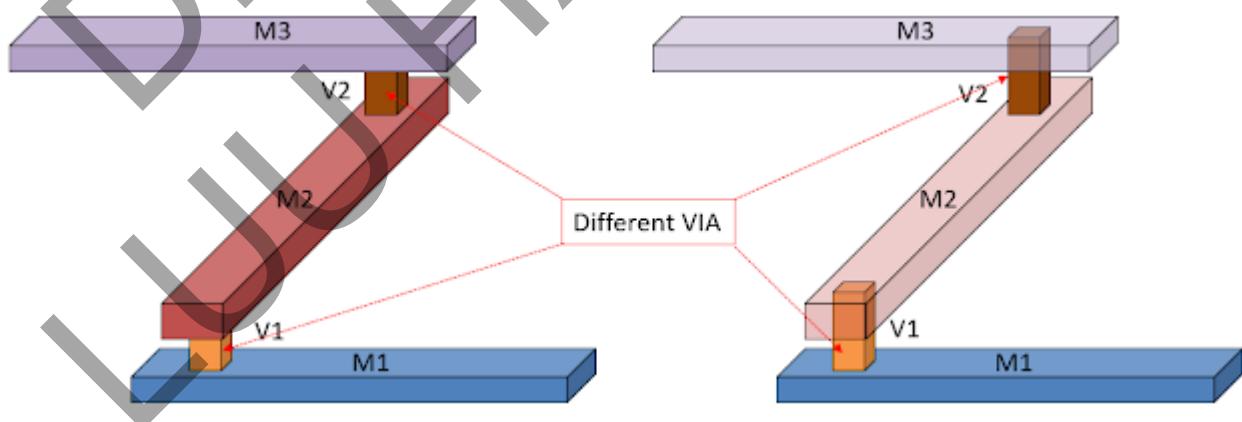
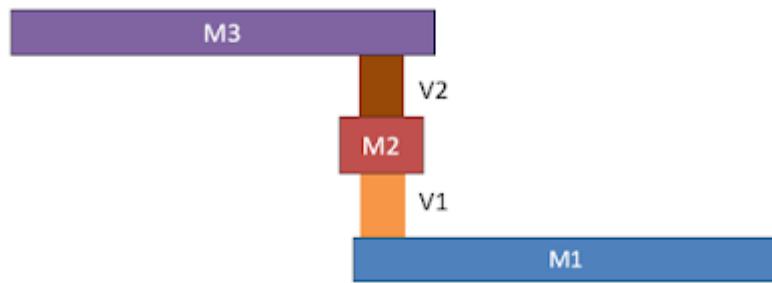
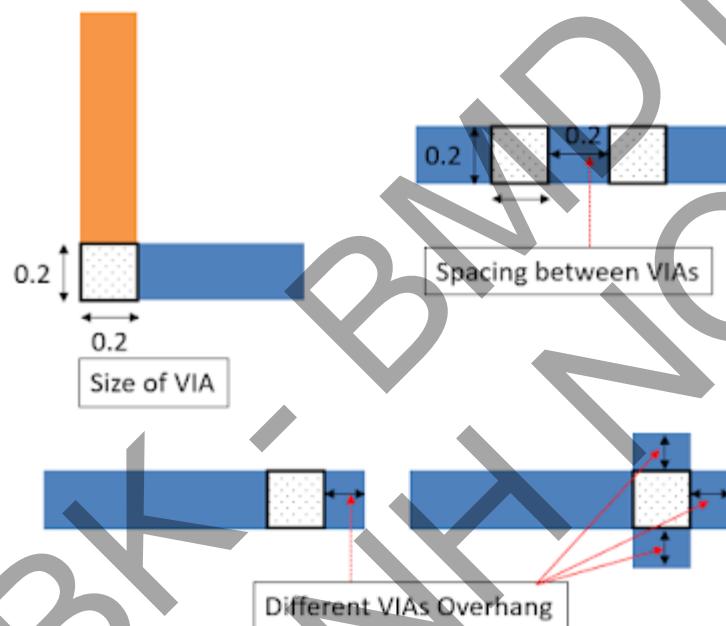


Figure 155 3D view (left) of VIA and metal connection (right) transparent view of connection.

**Figure 156** Side view of VIA and metal connection.

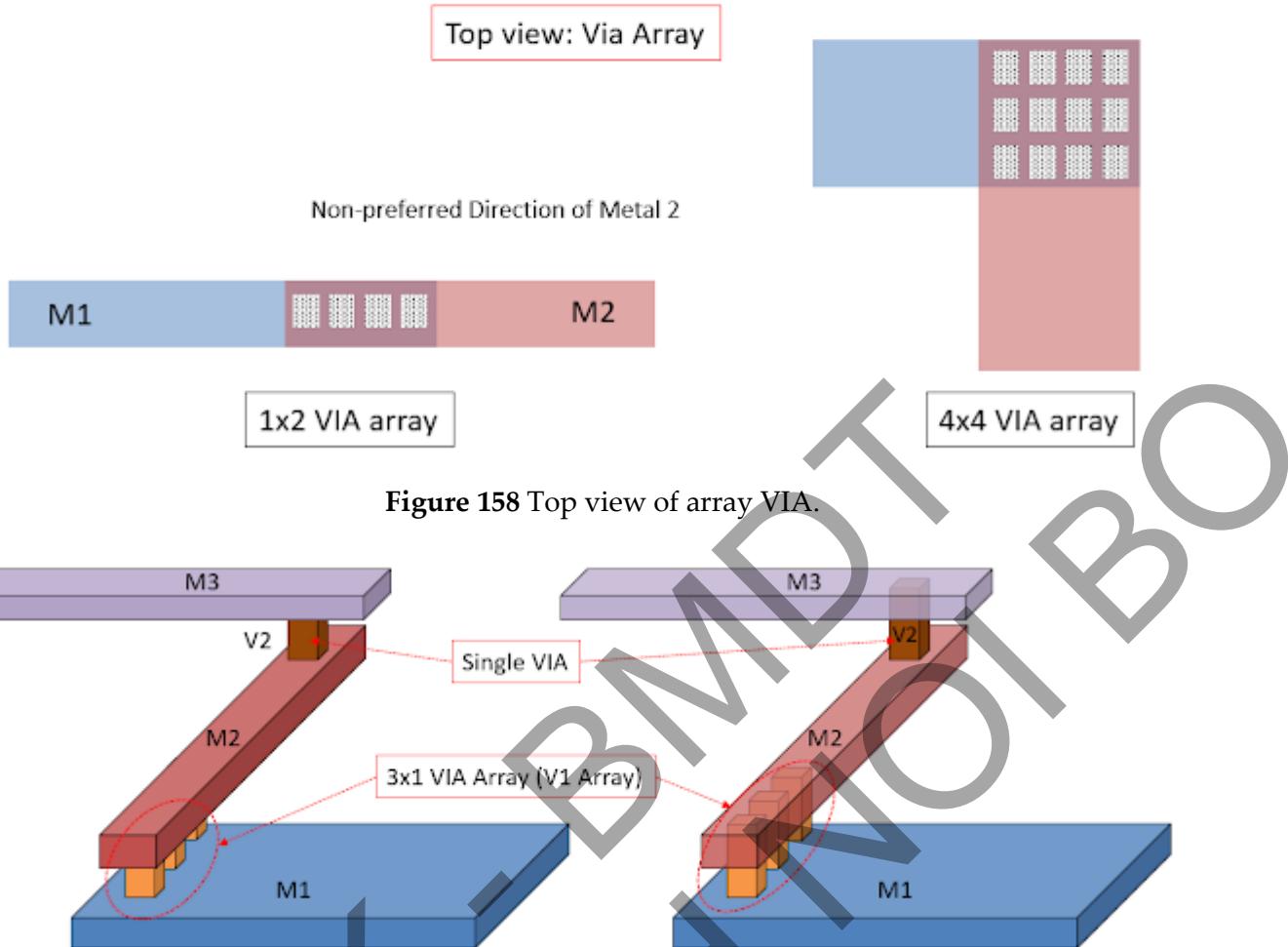
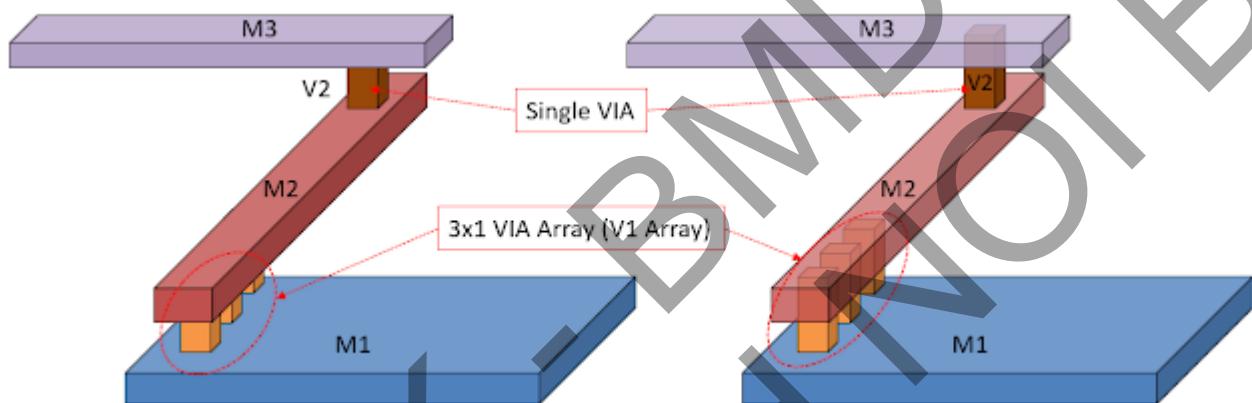
There are certain Design Rules for VIAs also. Students can observe the below figure to have a general idea.

**Figure 157** The example of VIA's design rules.

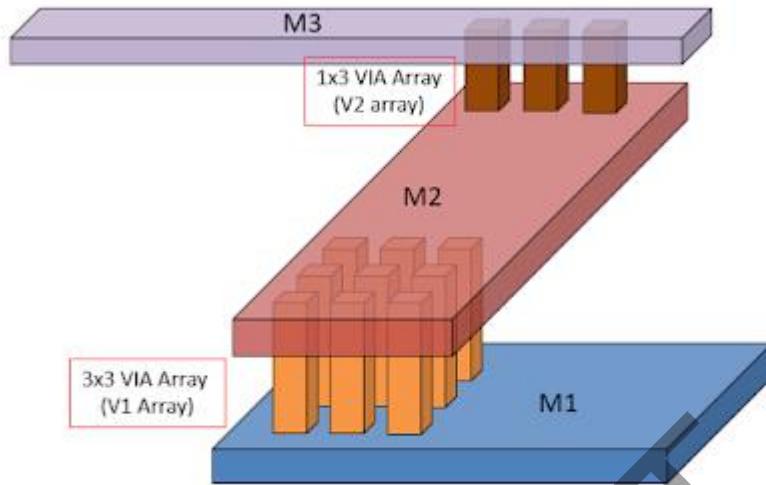
VIAs can be asymmetric, meaning the overhangs in the x and y directions are different. The overhang parameters refer to those of a VIA connecting preferred-direction wires. If the wires are in the nonpreferred direction, the VIA is rotated and the overhangs are reversed meaning that the extension in the x and y directions are given by the y overhang parameters.

2. VIA array:

Array VIAs are used for connecting wide wires where the required cut size would exceed the maximum cut size of the simple VIA. In an array VIA, the region of an intersection of the wires is filled by a regular array of small cuts of fixed size and separation.

**Figure 158** Top view of array VIA.**Figure 159** (left) 3D view of VIA array (right) transparent view of VIA array.

The 3D view helps students to understand how different layers relate to each other, students can also see that size is 3×1 between metal1 and metal2. When it comes to metal2 and metal3. A single VIA connects both wires. But do not think that it will always be the case. It depends on the design and a width of a metal wire. If students want to use a VIA array or single VIA, in the below figure students can see that metal1 and metal2 are using 3×3 VIA array. Between metal2 and metal3, it is 1×3 VIA array.

**Figure 160** 3D view of VIA and metal connection.

Why do they use a VIA array instead of a single-cut VIA?

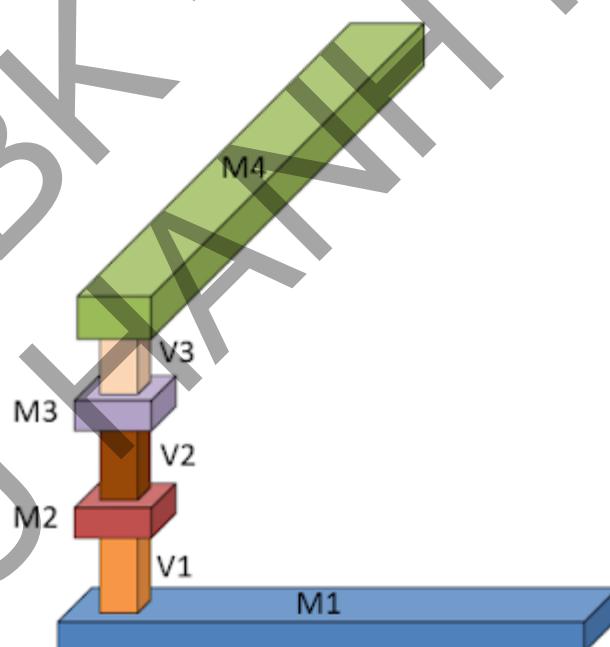
→ Usually at the VIA where two different types of metals meet and the formation of contact resistance.

If they go with multi-cut VIA, the contact resistance has fewer drops of voltage in the rails.

Rethinking the equation:

$$R = \rho \times \frac{l}{A}$$

3. Stacked VIA:

**Figure 161** 3D view of stacked VIA and metal connection.

Why do they choose upper metal layers to put power rail and output signal?

APPENDIX D:

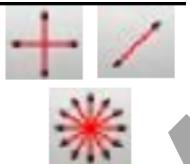
BINDKEYS IN CADENCE VIRTUOSO

For more convenience when using Virtuoso Schematic Design, the table below presents a quick summary of some of the most useful features and commands in the Virtuoso Schematic Editor, Virtuoso Symbol Editor, and Virtuoso Layout Editor as follows:

Function	Bindkey	Icon	Toolbar access	Relevant editor
WORKING WITH OTHER INSTANCES				
Place a schematic or layout instance (or layout mosaic)	I		Create > Instance	Schematic Layout
Edit properties of selected object	Q		Edit > Properties > Objects. Edit > Basic > Properties	Schematic Symbol Layout
Add Property Editor assistant to sidebar	-	-	Window > Assistants > Property Editor	Schematic Symbol Layout
Descend into design of selected instance	Read-only: E Edit: Shift + E In-place: X Descend: Shift + X	-	Edit > Hierarchy > Descend Edit/Read	Schematic Layout
Return to previous level of hierarchy	Ctrl + E Shift + B	-	Edit > Hierarchy > Return	Schematic Layout
Display hierarchy of current design	Shift + T	-	Edit > Hierarchy > Print Tree Edit > Hierarchy > Tree	Schematic Layout
DRAWING COMMANDS				
Undo/Redo	U or Shift+U		Edit > Undo/Redo	Schematic Symbol

				Layout
Check and save design	Shift + X		File > Check and Save	Schematic Symbol
Delete deleted objects	Delete		Edit > Delete	Schematic Symbol Layout
Copy selected objects	C		Edit > Copy	Schematic Symbol Layout
Draw a wire	W		Create > Wire (narrow)	Schematic
Draw a path	P		Create > Wiring > Wire	Layout
Draw a line	-		Create > Shape > Line	Symbol
Add a pin	P		Create > Pin	Schematic Symbol Layout
Add a label	L		Create > Wire Name	Schematic
			Create > Label	Symbol Layout
Move instance, keeping wire connections	M		Edit > Stretch	Schematic Symbol
Stretch shape	S			Layout
Move instance only (not maintaining wire connections)	Shift + M		Edit > Move	Schematic Symbol
	M			Layout
Rotate object (hot	R		Edit > Rotate	Schematic

keys apply to objects being moved; icon or menu applies to selected objects)	Shift + O		Edit > Rotate > (select option)	Symbol Layout
Mirror object	Shift + R		Edit > Rotate (press F3 for menu to mirror)	Schematic Symbol
	Ctrl + J		Edit > Flip > (select option)	Layout
Add labeled wire stubs to selected instance	Spacebar	-	Create > Wire Stubs and Names	Schematic
Align selected group of objects (use menu or drop-down on icon to select alignment type)			Edit > Align > (select option)	Schematic Symbol
			Edit > Advanced > Align	Layout
Open 'Create Via' window	O		Create > Via	Layout
Create bus (many parallel paths)	Ctrl + Shift + X		Create > Wiring > Bus	Layout
Draw a ruler	K	-	Tools > Create Measurement	Layout
Clear all rulers	Shift + K	-	Tools > Clear All Measurements	Layout
Toggle between Full / Partial select mode (controls what happens	-		Options > Selection > (Choose 'Partial' / 'Full' from drop-	Layout
		(click to toggle)		

when students click and drag a selection box over a shape)			down)	
Toggle between grouping / ungrouping vias stacked together			Options > Selection > (Select / deselect 'Via Stack')	Layout
Toggle between orthogonal / diagonal / any angle snap modes	-		Options > Display Options > (modify Snap Modes)	Layout
Chop rectangle out of selected polygon	Shift + C		Edit > Basic > Chop	Layout
Group / ungroup selected layout objects			Create > Group / Edit > Group > Ungroup	Layout
Toggle gravity (snaps cursor to edges, centers of shapes)	G		Options > Editor > (Select / deselect 'Gravity On')	Layout

DISPLAY CONFIGURATION

Zoom to fit everything in visible area	F		View > Zoom to fit	Schematic Symbol Layout
Zoom in	[or scroll		View > Zoom in	Schematic Symbol
	Ctrl + Z or scroll			Layout
Zoom out	[or scroll		View > Zoom out	Schematic Symbol

	Shift + Z or scroll			Layout
Draw box to specify zoom area	z or right click and drag	-	View > Zoom to area	Schematic Symbol Layout
Open 'Display Options' window (change snap settings, layout hierarchy visibility)	O	-	Options > Display	Schematic Symbol
	E			Layout
Open 'Display Editor' window (toggle display defaults, gravity)	Shift + O	-	Options > Display	Schematic Symbol
	Shift + E			Layout
Toggle visibility of certain instance text labels	-	-	View > Hide/Show Instance Labels, Notes, etc.	Schematic Symbol
Toggle visibility, select ability of certain drawing layers	-	-	'V' (visible) or 'S' (selectable) checkbox in Layers Panel of Palette Assistant	Layout
Automatic design rule checking (enforce / notify)	-	 	Options > DRD Edit > (Check 'Enforce' or 'Notify')	Layout

Table 13 Summarized Virtuoso bindkeys.

APPENDIX E:

ENABLE A SHARED FOLDER FOR A VIRTUAL MACHINE

Before using the virtual machine, students must establish a shared folder link between their PC and the virtual machine. This precaution helps prevent the loss of unsaved work if the virtual machine encounters issues. Virtual machines save data only when properly powered off; unexpected crashes or accidental shutdowns of VMWare can result in complete data loss. How does it work? For example, if you choose a folder on your PC like *D:\shared*, any files or packages placed in this folder will automatically appear in the "shared" folder on the virtual machine.

To share a folder from students' PCs to VMWare, first, click **VM > Settings > Options...**, and make sure "**Shared Folder**" is "**Always enabled**". Please follow the steps in **Figure 162** and complete this work.

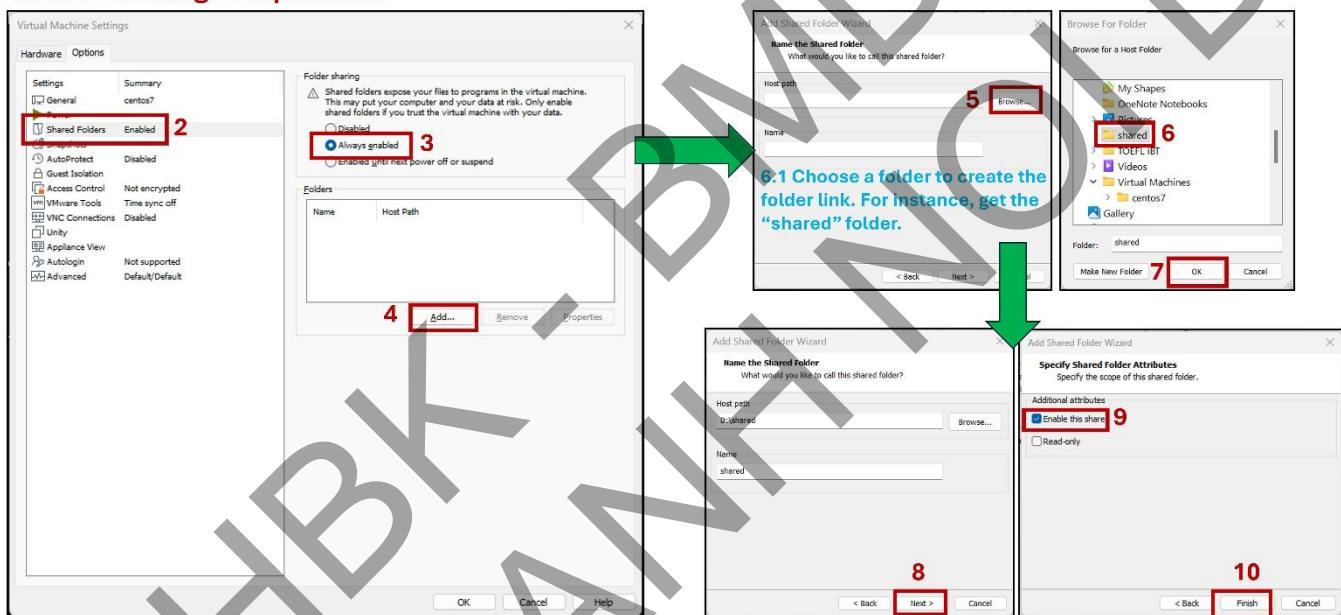
1. VM > Settings > Options...

Figure 162 Share a folder from PC to VMWare.

After configuring, students need to suspend and resume the machine for the configuration to be properly applied to the virtual machine. Add this command and compile your **.bashrc** file in the virtual machine.

```
alias mntsh="sudo /usr/bin/vmhgfs-fuse .host:/ /home/admin/shared -o subtype=vmhgfs-fuse,allow_other,nonempty,default_permissions,uid=$(id -u),gid=$(id -g)"
```

Students follow these steps.

```
total 4.5K
drwxr-xr-x. 2 admin admin 22 Apr 19 09:35 Desktop
drwxr-xr-x. 6 admin admin 73 Jun 23 13:55 Documents
drwxr-xr-x. 2 admin admin 6 Mar 2 05:11 Downloads
drwxr-xr-x. 2 admin admin 6 Mar 2 02:57 Music
drwxr-xr-x. 2 admin admin 6 Mar 2 02:57 Pictures
drwxr-xr-x. 3 admin admin 17 Apr 16 06:17 Public
drwxr-xr-x. 2 admin admin 6 Mar 2 02:57 Templates
drwxr-xr-x. 2 admin admin 6 Mar 2 02:57 Videos
dr-xr-xr-x. 1 admin admin 4.1K Jun 24 03:07 shared
[admin@centos7 ~]$ vi ~/.bashrc
```

(a)

```
# .bashrc

# Source global definitions
if [ -f /etc/bashrc ]; then
    . /etc/bashrc
fi

# Uncomment the following line if you don't like systemctl's auto-paging feature:
# export SYSTEMD_PAGER=

# User specific aliases and functions
alias vi="vim"
alias l="ls -lh"
alias ll="ls -lh"
alias mntsh="sudo /usr/bin/vmhgfs-fuse .host:/ /home/admin/shared -o subtype=vmhgfs-fuse,allow_other,nonempty,default_permissions,uid=$(id -u),gid=$(id -g)"
#alias mnt_shared="sudo mount -t fuse.vmhgfs-fuse -o allow_other .host:/shared ~/shared"
alias cl="clear; ll"
```

(b)

```
total 4.5K
drwxr-xr-x. 2 admin admin 22 Apr 19 09:35 Desktop
drwxr-xr-x. 6 admin admin 73 Jun 23 13:55 Documents
drwxr-xr-x. 2 admin admin 6 Mar 2 05:11 Downloads
drwxr-xr-x. 2 admin admin 6 Mar 2 02:57 Music
drwxr-xr-x. 2 admin admin 6 Mar 2 02:57 Pictures
drwxr-xr-x. 3 admin admin 17 Apr 16 06:17 Public
drwxr-xr-x. 2 admin admin 6 Mar 2 02:57 Templates
drwxr-xr-x. 2 admin admin 6 Mar 2 02:57 Videos
dr-xr-xr-x. 1 admin admin 4.1K Jun 24 03:07 shared
[admin@centos7 ~]$ vi ~/.bashrc
[admin@centos7 ~]$ source ~/.bashrc
```

(c)

Figure 163 (a) Open .bashrc to add the command (b) Switch to insert mode, then type or paste the command (or new alias) then press Esc and type :wq to save the changes and quit (c) Compile the

.bashrc file.

```
total 8.0K
-rw-rw-r--. 1 admin admin 7.4K Jun 19 12:34 CDS.log
drwxr-xr-x. 2 admin admin 22 Apr 28 09:46 Desktop
drwxr-xr-x. 3 admin admin 22 Jun 19 11:45 Documents
drwxr-xr-x. 2 admin admin 6 Mar 2 05:11 Downloads
drwxr-xr-x. 2 admin admin 6 Mar 2 02:57 Music
drwxr-xr-x. 2 admin admin 6 Mar 2 02:57 Pictures
drwxr-xr-x. 2 admin admin 6 Mar 2 02:57 Public
drwxr-xr-x. 2 admin admin 6 Mar 2 02:57 Templates
drwxr-xr-x. 2 admin admin 6 Mar 2 02:57 Videos
drwxrwxr-x. 15 admin admin 250 Apr 30 10:13 simulation
[admin@centos7 ~]$ mkdir shared
[admin@centos7 ~]$ mntsh
[sudo] password for admin:
```



```
total 12K
-rw-rw-r--. 1 admin admin 7.4K Jun 19 12:34 CDS.log
drwxr-xr-x. 2 admin admin 22 Apr 28 09:46 Desktop
drwxr-xr-x. 3 admin admin 22 Jun 19 11:45 Documents
drwxr-xr-x. 2 admin admin 6 Mar 2 05:11 Downloads
drwxr-xr-x. 2 admin admin 6 Mar 2 02:57 Music
drwxr-xr-x. 2 admin admin 6 Mar 2 02:57 Pictures
drwxr-xr-x. 2 admin admin 6 Mar 2 02:57 Public
drwxr-xr-x. 2 admin admin 6 Mar 2 02:57 Templates
drwxr-xr-x. 2 admin admin 6 Mar 2 02:57 Videos
dr-xr-xr-x. 1 root root 4.0K Jun 20 05:39 shared
drwxrwxr-x. 15 admin admin 250 Apr 30 10:13 simulation
[admin@centos7 ~]$ cp -rf Documents/virtuoso/<your_project> shared/shared/
```

Figure 164 Open Cadence Virtuoso.

