Schematic and Layout Design Simulation Using Cadence Spectre

Marym Shahangian, James E. Stine

Oklahoma State University

VLSI Computer Architecture Reaserch Group

The Cadence suite integrates various tools for schematic capture, simulation, layout, and post-layout verification, specifically tailored for analog design workflows and capable of handling a large number of devices. In this tutorial, Composer is utilized for schematic entry and test bench creation. The Analog Design Environment (ADE) is then configured to run simulations using Spectre, a powerful circuit simulator renowned for its accuracy and efficiency in analog and mixed-signal simulations.

Spectre performs detailed analysis by solving the circuit equations across a wide range of operating conditions, allowing for precise characterization of circuit behavior. It is particularly effective for tasks such as transient analysis, DC operating point analysis, and AC small-signal analysis, providing valuable insights into the circuit's performance.

The process detailed in this tutorial includes capturing the schematic of a CMOS inverter, creating its corresponding symbol, and setting up a test bench to evaluate its performance. Device models for the FETs are crafted using a text editor, after which simulations are configured and executed in ADE with Spectre, and results are plotted for analysis. To use a circuit simulator effectively, a thorough understanding of the circuit is essential. The simulator excels in handling numerous operating points but does not inherently grasp the circuit's behavior. Thus, it is crucial to interpret simulation data to identify trends and insights, rather than expecting exact matches to manual calculations.

This document aims to equip users with practical skills and insights for leveraging Cadence tools in circuit design and simulation. Additionally, the Calibre tool is employed for Design Rule Checking (DRC), Layout Versus Schematic (LVS) verification, and parasitic extraction, ensuring the accuracy and reliability of the design.

James E. Stine, Jr., Ph.D.
Edward Joullian Endowed Chair and Professor in Engineering
Oklahoma State University
Department of Electrical and Computer Engineering
VLSI Computer Architecture Research Group
Stillwater, OK 74078 USA
E-mail: james.stine@okstate.edu

Simulation with Spectre

Spectre is a highly-regarded SPICE-class circuit simulator developed and distributed by Cadence Design Systems, a leader in electronic design automation. It offers a comprehensive suite of fundamental SPICE analyses and supports a wide range of component models. Additionally, Spectre is compatible with the Verilog-A modeling language, providing flexibility for advanced circuit modeling. With Spectre, engineers can perform a variety of detailed circuit analyses, including DC, AC, Transient, and Noise Analysis, enabling precise evaluation of circuit performance.

Here are some key points about using Spectre in Cadence:

1. **Setting Up Your Environment**:
   - **Create a New Library and Cell**: Start by creating a new library and cell in Cadence Virtuoso.
   - **Draw the Schematic**: Use the Virtuoso Schematic Editor to draw your circuit.
2. **Configuring the Simulation**:
   - **Add Simulation Parameters**: Set up the simulation parameters such as analysis type (e.g., DC, AC, transient).
   - **Include Models**: Ensure that the correct device models are included in your simulation setup.
3. **Running the Simulation**:
   - **Launch Spectre**: Use the Analog Design Environment (ADE) to configure and run the Spectre simulation.
   - **Analyze Results**: After the simulation, use the waveform viewer to analyze the results.

Start:

Start the Cadence Document Server from the command line by typing:

```
cdsdoc &
```

Alternatively, you can use:

```
source /path/to/your/eda-tools/tower.sh   > virtuoso
```

- Note: Replace **`/path/to/your/eda-tools/`** with the actual path where your EDA tools are installed. Ensure that **`virtuoso`** is the command that starts Cadence in your environment.

In the Command Interface Window (CIW), follow these steps to create a library for our Cadence work.:

- Click on **`Files > New > Library`**.
- Change to directory **`~/cds/libs`**. If necessary, create it in unix.
- Choose a library name.
- Click OK.

Create the Inverter Schematic and Symbols

➢ *Schematic*

- Click on **`Tools`** in the menu.
- Select **`Library Manager`**.
- Go to **`File > New > CellView.`**
- Enter the desired **`name`** for the new cell view.

❖ Note: Cadence has many keyboard shortcuts. Some useful commands for later use include:

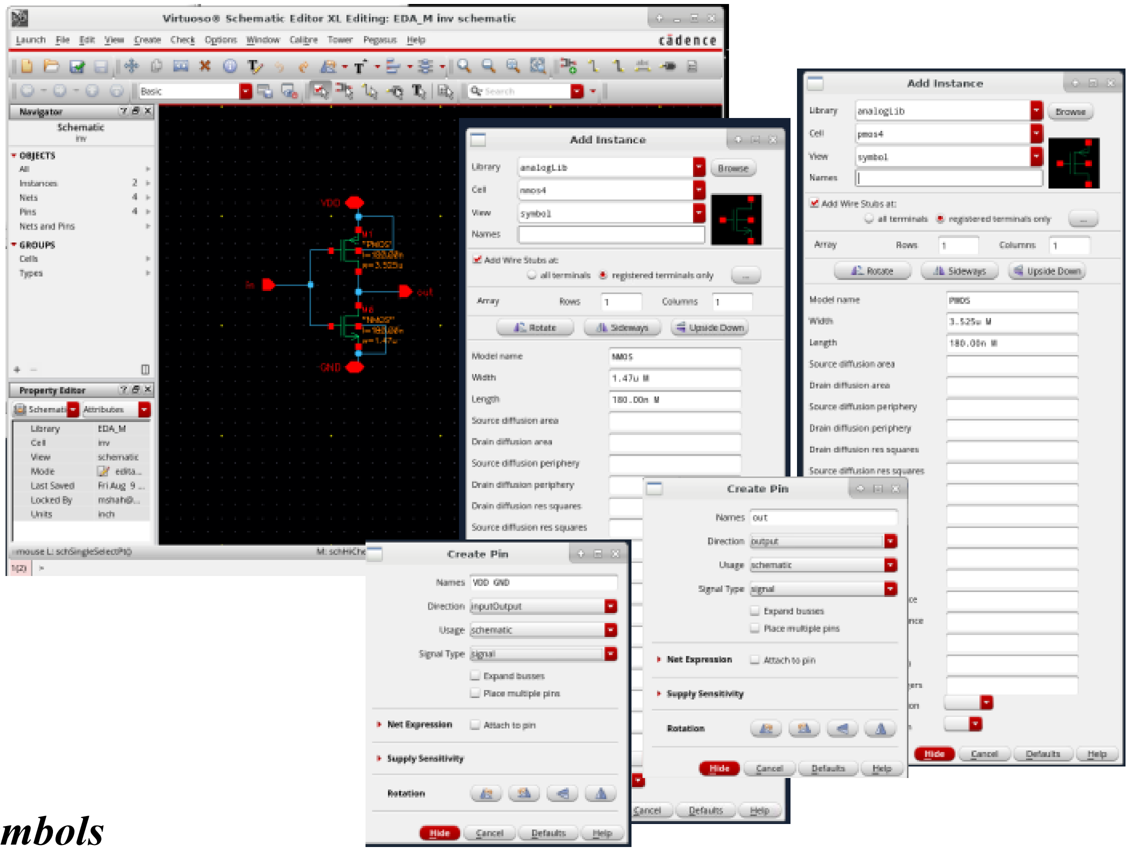
- **`i (instantiate)`**: Creates a new object in the schematic.
- **`p (pin)`**: Creates pins to allow signals to leave the schematic.
- **`c (copy)`**: Copies objects.
- **`m (move)`**: Moves objects and drags their wiring with them. Use a lowercase 'm' for this action.
- **`M (Move)`**: Moves objects after disconnecting their wires. Use an uppercase 'M' for this action.
- **`l (label)`**: Creates labels for wires to avoid default names.
- **`q (query)`**: Opens a dialog box showing object properties.
- **`w (wire)`**: Creates wires. Type 'w', left-click to start a wire, left-click to make a corner, and double left-click to end the wire.

some commands, such as copy, move, and wire, have additional options available through dialog boxes. Press **`(F3)`** to display these dialog boxes. Additionally, Cadence commands use the bottom line of the window as an information bar. To create the inverter schematic, follow these steps:

- Press **`i`** to initiate the process.
- Click on **`Browse`**. Navigate to the **`analogLib`** library and select the **`nmos4`** cell. Click on the **`symbol view`**, then click **`Close`**. The **`nmos4`** is a 4-terminal device, meaning the bulk/well terminal is explicitly shown on the symbol.
- A form will appear displaying all the options for the **`nmos4`**.

In the same way, the process is repeated for PMOS, and the table is filled according to the target technology, which in this case is 180nm.

- Press **p** to initiate the process for the **VDD** and **GND** pins, setting them as inputOutput, and for the **in** and **out** pins as input and output, respectively.
- Press **w** to create wires to wire everything together.
- If you later want to inspect or modify those values of a device, select the device and press **`(q)`**. This action opens the **`Edit Instance Properties`** window, where the parameters of the device can be adjusted as needed.
- To save the schematic, press **X**. The inverter schematic is now complete.



Add Instance
Library: analogLib
Cell: nmos4
View: symbol
☑ Add Wire Stubs at: ○ all terminals ● registered terminals only
Array: Rows 1, Columns 1
Model name: NMOS
Width: 1.47u M
Length: 180.00n M

Add Instance
Library: analogLib
Cell: pmos4
View: symbol
☑ Add Wire Stubs at: ○ all terminals ● registered terminals only
Array: Rows 1, Columns 1
Model name: PMOS
Width: 3.525u M
Length: 180.00n M

Create Pin
Names: VDD GND
Direction: inputOutput
Usage: schematic
Signal Type: signal
□ Expand busses
□ Place multiple pins

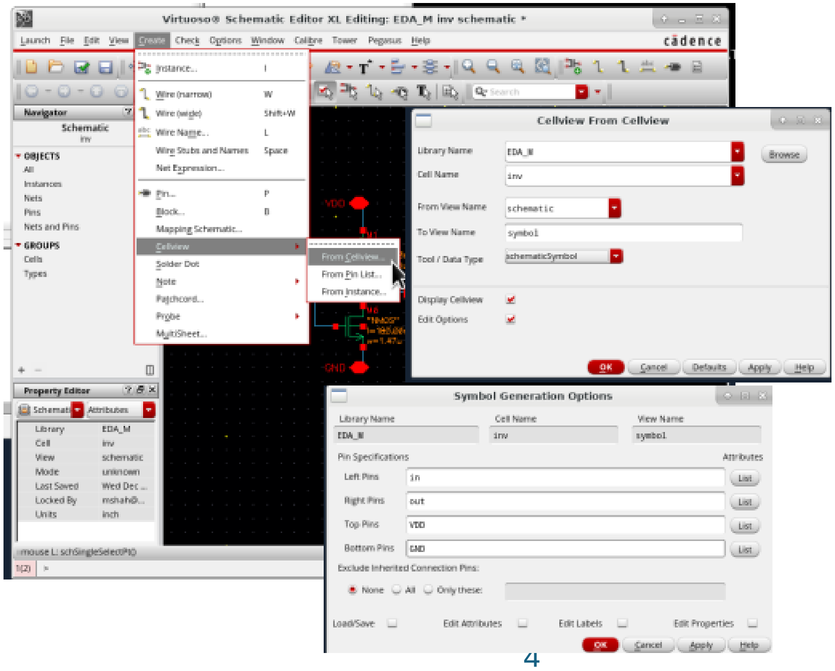
Create Pin
Names: out
Direction: output
Usage: schematic
Signal Type: signal
□ Expand busses
□ Place multiple pins

➢ *Symbols*

A symbol will be created to allow the inverter to be instantiated in other schematics without copying the entire schematic each time.

To proceed, follow these steps:

- Click on **`Creat > Cellview > From Cellview`**, then click OK.

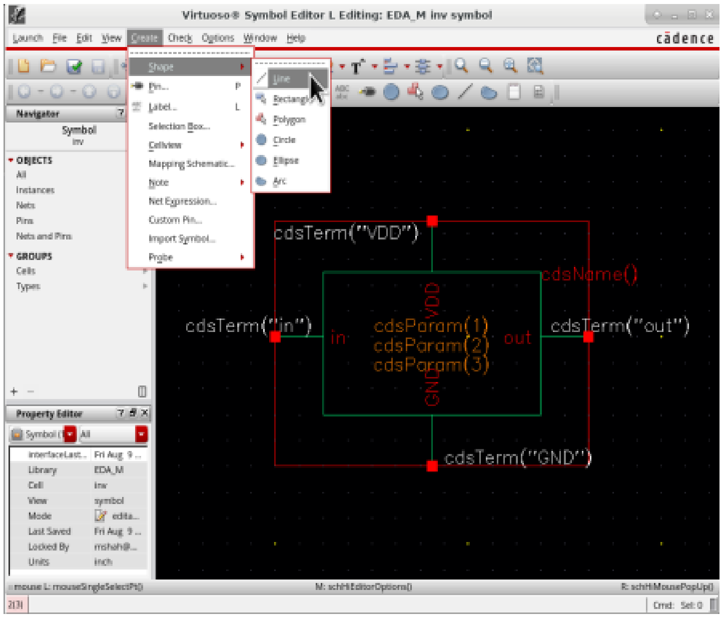


Cellview From Cellview
Library Name: EDA_M
Cell Name: inv
From View Name: schematic
To View Name: symbol
Tool / Data Type: schematicSymbol
Display Cellview: ☑
Edit Options: ☑

Symbol Generation Options
Library Name: EDA_M
Cell Name: inv
View Name: symbol
Pin Specifications
Left Pins: in
Right Pins: out
Top Pins: VDD
Bottom Pins: GND
Exclude Inherited Connection Pins: ● None ○ All ○ Only these:
Load/Save □ Edit Attributes □ Edit Labels □ Edit Properties □

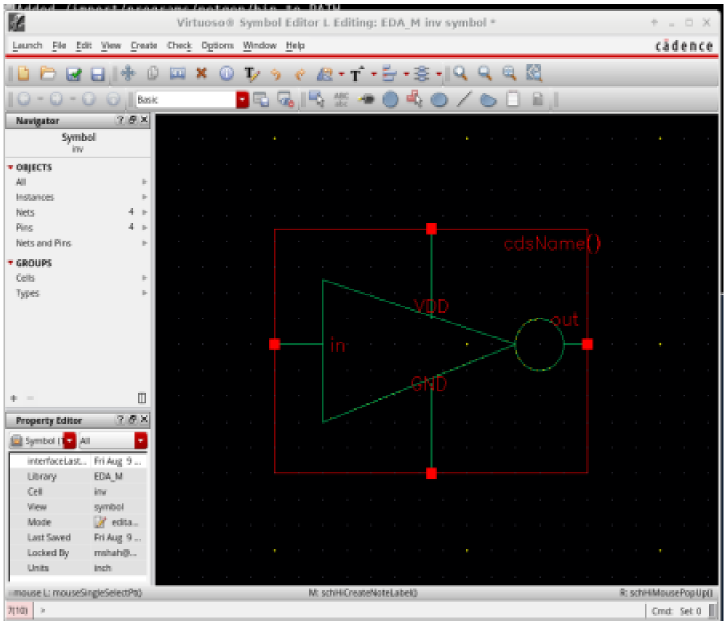
A symbol editor window will now open. The default symbol is sufficient, but if enhancement is desired, use the following commands:

- `Create > Shape > Polygon`
- `Create > Shape > Line`
- `Create > Shape > Circle`



interfaceLast... Fri Aug 9 ...
Library EDA_M
Cell inv
View symbol
Mode edita...
Last Saved Fri Aug 9 ...
Locked By mshah@...
Units inch

To create the symbol shown below, the original green box will need to be deleted. To do this, grab the edge, stretch it, click in the middle, and then delete it.Keep in mind the red bounding box defines the clickable area in higher-level schematics for selecting your inverter, so be careful not to delete it. Ensure the symbol is saved before exiting.

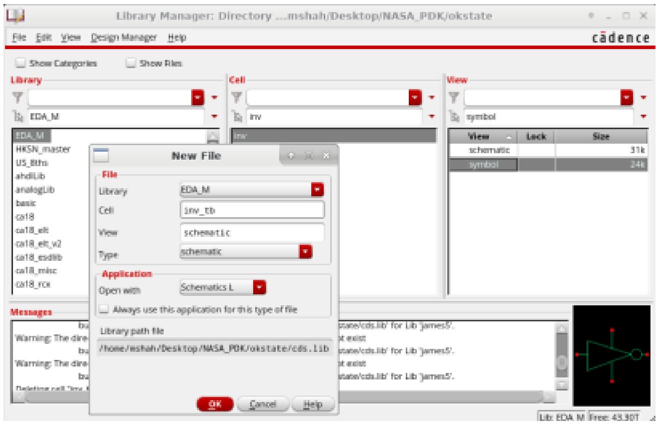


interfaceLast... Fri Aug 9 ...
Library EDA_M
Cell inv
View symbol
Mode edita...
Last Saved Fri Aug 9 ...
Locked By mshah@...
Units inch

Create Test Bench

Now, it's time to test the inverter by creating a test bench to measure its DC transfer curve. For this, the necessary components need to be set up: a VDD supply for the inverter, an input signal, and an appropriate output load.

- Create a schematic named **`test_inverter or (inv_tb)`** within our library. In the Window (CIW), Click on **`Files > New > CellView`**



Library Manager: Directory ....mshah/Desktop/NASA_PDK/okstate

New File

File

Library: EDA_M

Cell: inv_tb

View: schematic

Type: schematic

Application

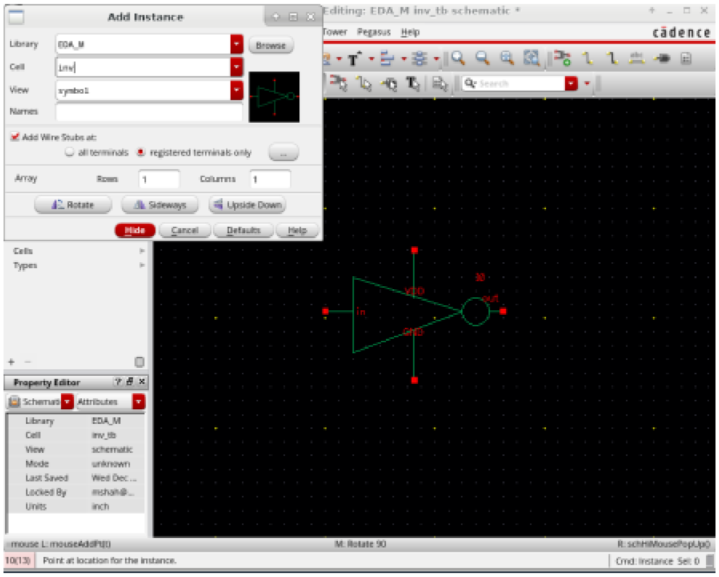
Open with: Schematics L

☐ Always use this application for this type of file

Library path file: /home/mshah/Desktop/NASA_PDK/okstate/cds.lib

View	Lock	Size
schematic		31k
symbol		24k

- In the Schematic Window, press **`(i)`**. Instead of working with analog parts, go to the library that we just created, then navigate to the cell where we made a symbol, and place this symbol inside the schematic.



Add Instance

Library: EDA_M

Cell: inv

View: symbol

Names:

☑ Add Wire Stubs at: ○ all terminals ● registered terminals only

Array Rows 1 Columns 1

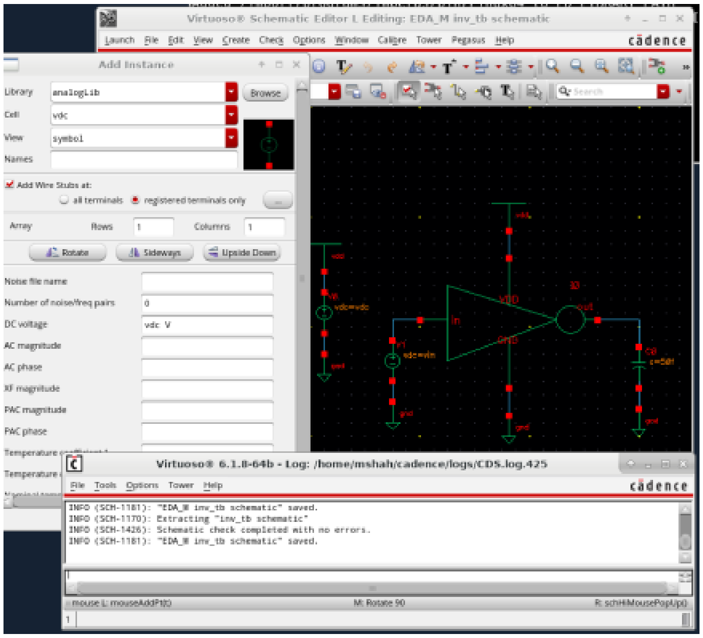
Editing: EDA_M inv_tb schematic *

Property Editor	
Library	EDA_M
Cell	inv_tb
View	schematic
Mode	unknown
Last Saved	Wed Dec ...
Locked By	mshah@...
Units	inch

mouse L: mouseAddPt() M: Rotate 90 R: schHiMousePopUp()

10(13) Point at location for the instance. Cmd: Instance Sel: 0

- Use **`(q)`** to set the voltage of the power supply to **`vdc`**, the input supply to **`vin`**, and the load capacitor to the appropriate value. Setting the power supply and input supply to the variables **`vdc`** and **`vin`** means that they will be configured in the simulator. This is a powerful technique to simulate many scenarios without changing your schematic each time. It can be used with most fields of most symbols.


Virtuoso® Schematic Editor L Editing: EDA_M inv_tb schematic
Launch File Edit View Create Check Options Window Calibre Tower Pegasus Help
cādence
Add Instance
Browse
Rotate
Sideways
Upside Down
vdd
gnd
in
out
VDD
vdc=vdc
vdc=vin
c=50f

Library: analogLib
Cell: vdc
View: symbol
Names:
☑ Add Wire Stubs at:
○ all terminals ● registered terminals only
Array Rows: 1 Columns: 1
Noise file name:
Number of noise/freq pairs: 0
DC voltage: vdc V
AC magnitude:
AC phase:
XF magnitude:
PAC magnitude:
PAC phase:

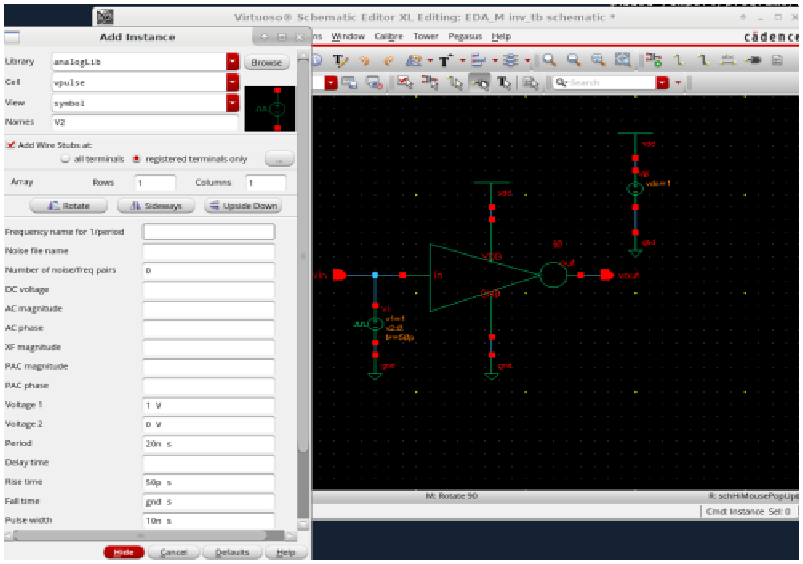
Virtuoso® 6.1.8-64b - Log: /home/mshah/cadence/logs/CDS.log.425

```
INFO (SCH-1181): "EDA_M inv_tb schematic" saved.
INFO (SCH-1170): Extracting "inv_tb schematic"
INFO (SCH-1426): Schematic check completed with no errors.
INFO (SCH-1181): "EDA_M inv_tb schematic" saved.
```

mouse L: mouseAddPt(t) M: Rotate 90 R: schHiMousePopUp()

- To create a pulse input, set the input signal parameters by selecting the instance and pressing **`(q)`**. In the **`'Voltage 1'`** and **`'Voltage 2'`** fields, input the desired low and high voltage levels for the pulse. Set the **`'Rise Time'`**, **`'Fall Time'`**, **`'Pulse Width'`**, and **`'Period'`** to define the shape and timing of the pulse. This setup will allow you to generate a time-varying input signal for your simulation.
- Label the wires using the **`(l)`** command, if needed.
- Check and save **`(x)`**.

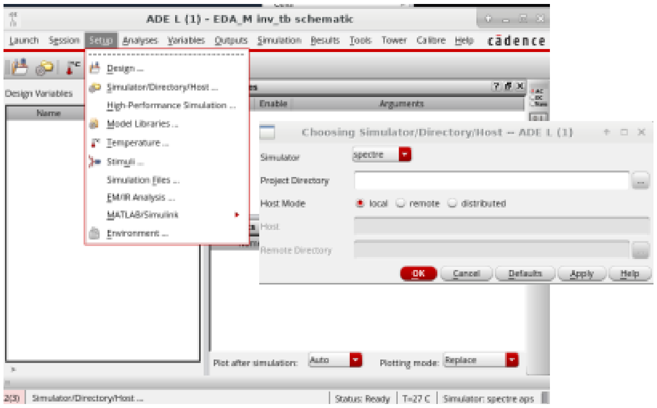
Additionally, note that Cadence's built-in scripting language, Skill, allows defining complex equations and automating processes, providing significant flexibility in simulations.Once completed, the schematic should resemble the example provided.



Library: analogLib
Cell: vpulse
View: symbol
Names: V2
☑ Add Wire Stubs at:
○ all terminals ● registered terminals only
Array Rows: 1 Columns: 1
Frequency name for 1/period:
Noise file name:
Number of noise/freq pairs: 0
DC voltage:
AC magnitude:
AC phase:
XF magnitude:
PAC magnitude:
PAC phase:
Voltage 1: 1 V
Voltage 2: 0 V
Period: 20n s
Delay time:
Rise time: 50p s
Fall time: gnd s
Pulse width: 10n s

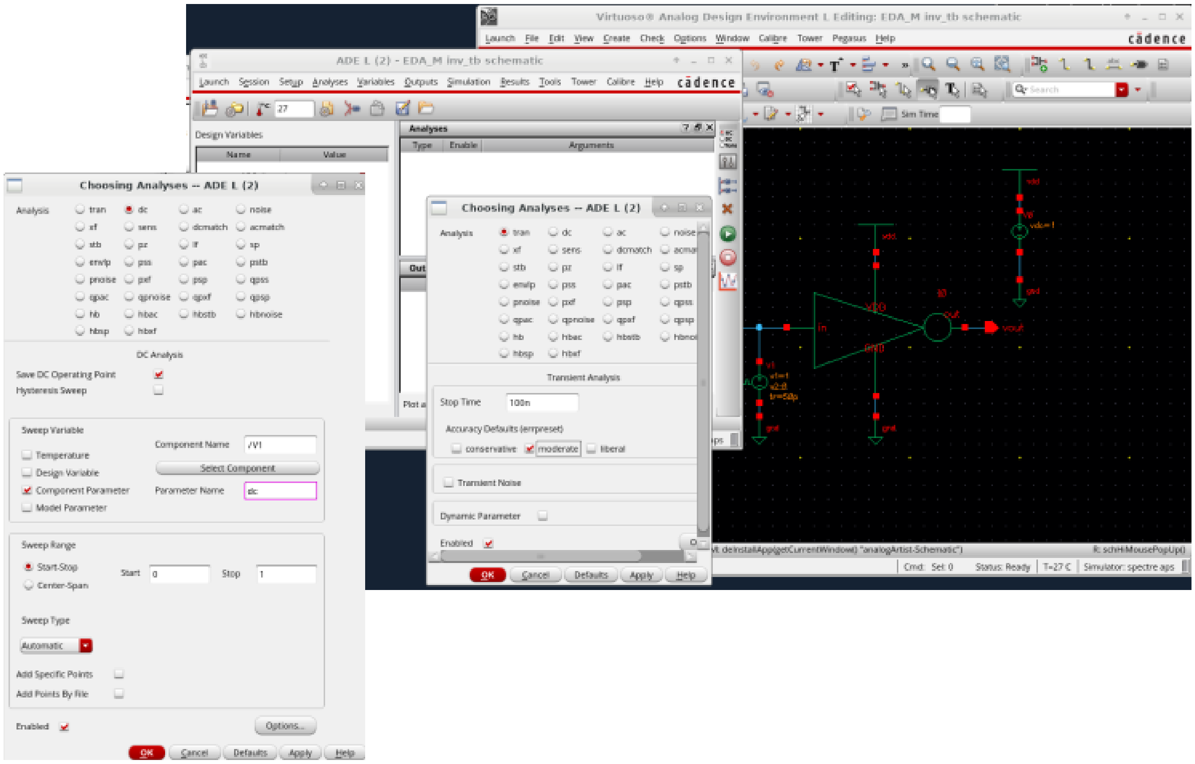
Running Simulations with Spectre

Now, to simulate using **`Spectre`**, select **`Setup`** to configure your simulation environment. Under **`Setup`**, configure the necessary parameters such as analysis type, output variables, and any additional settings required for your simulation. After configuring, run the simulation and then use the **`Results`** tab to plot the output signal and analyze the performance of your circuit


ADE L (1) - EDA_M inv_tb schematic
Launch Session Setup Analyses Variables Outputs Simulation Results Tools Tower Calibre Help cadence
Design ...
Simulator/Directory/Host ...
High-Performance Simulation ...
Model Libraries ...
Temperature ...
Stimuli ...
Simulation Files ...
EM/IR Analysis ...
MATLAB/Simulink
Environment ...
Design Variables
Name
Enable
Arguments
Choosing Simulator/Directory/Host -- ADE L (1)
OK
Cancel
Defaults
Apply
Help

Simulator: spectre
Project Directory:
Host Mode: ● local ○ remote ○ distributed
Host:
Remote Directory:
Plot after simulation: Auto
Plotting mode: Replace
Simulator/Directory/Host ...
Status: Ready | T=27 C | Simulator: spectre aps

- Click **`Launch > ADE L >  Setup > Simulator > Spectre > Apply`**
- Click **`Setup > Temperature`** and change the temperature to **`25 degrees`** **>** OK.


Virtuoso® Analog Design Environment L Editing: EDA_M inv_tb schematic
Launch File Edit View Create Check Options Window Calibre Tower Pegasus Help cadence
ADE L (2) - EDA_M inv_tb schematic
Launch Session Setup Analyses Variables Outputs Simulation Results Tools Tower Calibre Help cadence
Design Variables
Name
Value
Analyses
Type
Enable
Arguments
Choosing Analyses -- ADE L (2)
Analysis
tran
dc
ac
noise
xf
sens
dcmatch
acmatch
stb
pz
lf
sp
envlp
pss
pac
pstb
pnoise
pxf
psp
qpss
qpac
qpnoise
qpxf
qpsp
hb
hbac
hbstb
hbnoise
hbsp
hbxf
DC Analysis
Transient Analysis
Sweep Variable
Sweep Range
Sweep Type
Select Component
Options...
OK
Cancel
Defaults
Apply
Help
vdd
vin
vout
gnd
Sim Time

27

Analysis: ● dc
Save DC Operating Point ☑
Hysteresis Sweep □
□ Temperature
□ Design Variable
☑ Component Parameter
□ Model Parameter
Component Name /V1
Parameter Name dc
● Start-Stop ○ Center-Span
Start 0 Stop 1
Sweep Type Automatic
Add Specific Points □
Add Points By File □
Enabled ☑

Analysis: ● tran
Stop Time 100n
Accuracy Defaults (errpreset)
□ conservative ☑ moderate □ liberal
□ Transient Noise
Dynamic Parameter □
Enabled ☑

Cmd: Sel: 0 Status: Ready | T=27 C | Simulator: spectre aps

- right-click on the **`Analyses`** window and select **`Edit. (or Analyses > choose)`**
- Click on **`tran`**, select **`moderate`**, and then click **`Apply`**. You should see one analysis loaded.

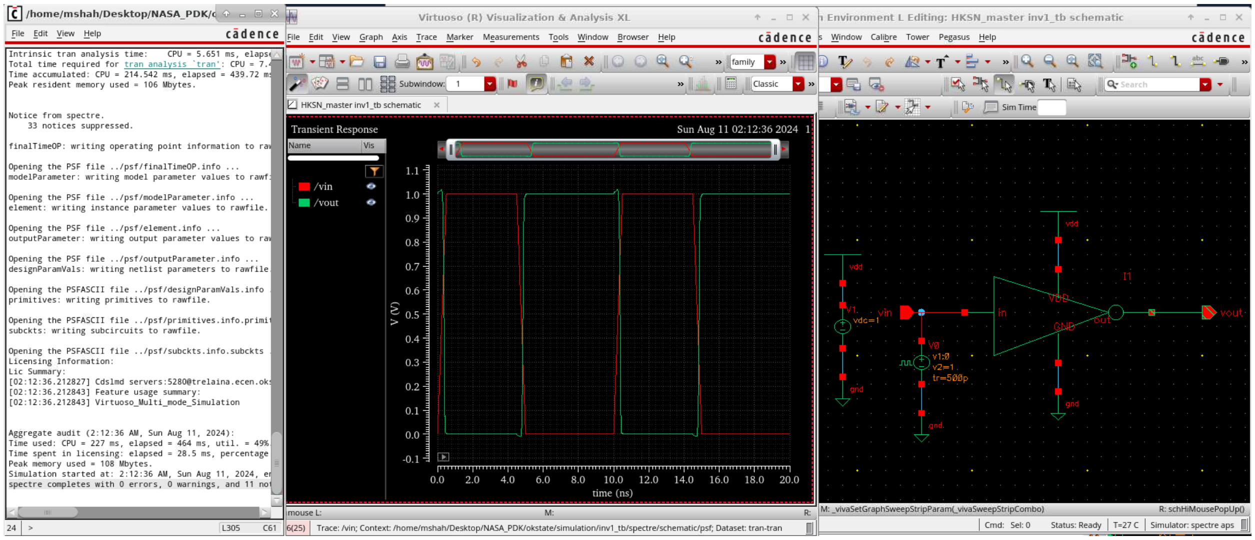
Next, we will set up a DC simulation where the input voltage is swept from 0 to 5 volts.

- Click on **`dc`**, check **`Save DC Operating Point`**, then go to **`Component Parameter`** and click **`Select Component`**.
- Manually create the Netlist. You usually only need to do this step if something odd is happening or you need to hand tweak the netlist. Click **`Simulation > Netlist > Create`**. Look at the netlist. (If you are happy with your simulation setup then save it.)
- Click **`output > to be plotted > select on schematic.(vin,vout)`**
- Click **`Simulation > Netlist and Run`**. Or Click the green stoplight.

Transient Analysis (tran): This type of analysis is used to examine the time-dependent behavior of a circuit. In other words, it shows how the circuit responds to a time-varying input (such as a pulse or a sinusoidal signal). Transient analysis helps you observe the circuit's behavior over time and see how voltages and currents change at every moment.

DC Analysis (dc): This analysis is used to examine the operating points and the circuit's behavior in a steady state. For example, DC analysis allows you to check the voltages and currents in the circuit under fixed DC conditions (such as a DC power supply). It's also used to evaluate the operating parameters of components (like transistors) at different operating points.

Combining these two types of analysis allows you to examine both the transient behavior of the circuit and its behavior under fixed DC conditions, providing important and complementary information about the overall performance of the circuit.



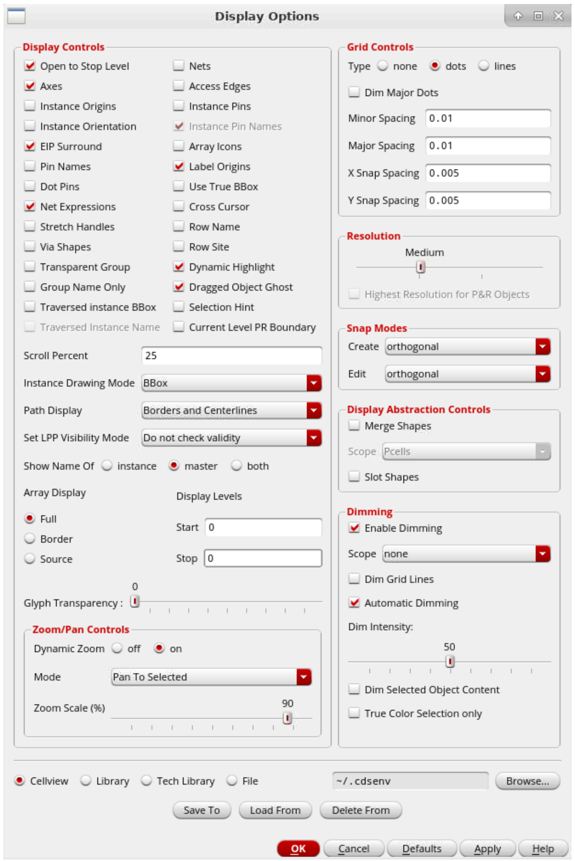
```
Intrinsic tran analysis time:    CPU = 5.651 ms, elapse
Total time required for tran analysis `tran': CPU = 7.4
Time accumulated: CPU = 214.542 ms, elapsed = 439.72 ms
Peak resident memory used = 106 Mbytes.


Notice from spectre.
    33 notices suppressed.

finalTimeOP: writing operating point information to raw

Opening the PSF file ../psf/finalTimeOP.info ...
modelParameter: writing model parameter values to rawfi

Opening the PSF file ../psf/modelParameter.info ...
element: writing instance parameter values to rawfile.

Opening the PSF file ../psf/element.info ...
outputParameter: writing output parameter values to raw

Opening the PSF file ../psf/outputParameter.info ...
designParamVals: writing netlist parameters to rawfile.

Opening the PSFASCII file ../psf/designParamVals.info .
primitives: writing primitives to rawfile.

Opening the PSFASCII file ../psf/primitives.info.primit
subckts: writing subcircuits to rawfile.

Opening the PSFASCII file ../psf/subckts.info.subckts .
Licensing Information:
Lic Summary:
[02:12:36.212827] Cdslmd servers:5280@trelaina.ecen.oks
[02:12:36.212843] Feature usage summary:
[02:12:36.212843] Virtuoso_Multi_mode_Simulation


Aggregate audit (2:12:36 AM, Sun Aug 11, 2024):
Time used: CPU = 227 ms, elapsed = 464 ms, util. = 49%.
Time spent in licensing: elapsed = 28.5 ms, percentage
Peak memory used = 108 Mbytes.
Simulation started at: 2:12:36 AM, Sun Aug 11, 2024, en
spectre completes with 0 errors, 0 warnings, and 11 not
```

/home/mshah/Desktop/NASA_PDK/c
File Edit View Help
cādence
24 >
L305 C61
Virtuoso (R) Visualization & Analysis XL
File Edit View Graph Axis Trace Marker Measurements Tools Window Browser Help
Subwindow: 1
family
Classic
HKSN_master inv1_tb schematic
Transient Response
Sun Aug 11 02:12:36 2024 1
Name Vis
/vin
/vout
V (V)
1.1
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
-0.1
0.0 2.0 4.0 6.0 8.0 10.0 12.0 14.0 16.0 18.0 20.0
time (ns)
mouse L:
M:
R:
6(25)
Trace: /vin; Context: /home/mshah/Desktop/NASA_PDK/okstate/simulation/inv1_tb/spectre/schematic/psf; Dataset: tran-tran
Environment L Editing: HKSN_master inv1_tb schematic
Window Calibre Tower Pegasus Help
Search
Sim Time
vdd
V1
vdc=1
gnd
vin
V0
v1:0
v2=1
tr=500p
in
VDD
GND
out
I1
vout
M: _vivaSetGraphSweepStripParam(_vivaSweepStripCombo)
R: schHiMousePopUp()
Cmd: Sel: 0
Status: Ready
T=27 C
Simulator: spectre aps

Create the Inverter Layout

In the schematic window, select **`Launch > Layout`** to begin designing the layout for the inverter. Choose the option to create a new layout automatically, or manually create a new one, then press **OK** to open the layout window.

Once the layout window is open, this is where the design process begins. Before starting, it is important to adjust the grid snapping resolution, which controls how the mouse snaps from one grid point to another. To set the resolution, press **(e)** or navigate to **`Options > Display`**. Set the **`X Snap Spacing`** and **`Y Snap Spacing`** to 0.005 for finer control, and press **OK** to confirm.



Display Options

Display Controls

☑ Open to Stop Level
☑ Axes
□ Instance Origins
□ Instance Orientation
☑ EIP Surround
□ Pin Names
□ Dot Pins
☑ Net Expressions
□ Stretch Handles
□ Via Shapes
□ Transparent Group
□ Group Name Only
□ Traversed instance BBox
□ Traversed Instance Name

□ Nets
□ Access Edges
□ Instance Pins
☑ Instance Pin Names
□ Array Icons
☑ Label Origins
□ Use True BBox
□ Cross Cursor
□ Row Name
□ Row Site
☑ Dynamic Highlight
☑ Dragged Object Ghost
□ Selection Hint
□ Current Level PR Boundary

Scroll Percent: 25
Instance Drawing Mode: BBox
Path Display: Borders and Centerlines
Set LPP Visibility Mode: Do not check validity
Show Name Of ○ instance ● master ○ both

Array Display
● Full
○ Border
○ Source

Display Levels
Start: 0
Stop: 0

Glyph Transparency : 0

Zoom/Pan Controls
Dynamic Zoom ○ off ● on
Mode: Pan To Selected
Zoom Scale (%): 90

Grid Controls
Type ○ none ● dots ○ lines
□ Dim Major Dots
Minor Spacing: 0.01
Major Spacing: 0.01
X Snap Spacing: 0.005
Y Snap Spacing: 0.005

Resolution
Medium
□ Highest Resolution for P&R Objects

Snap Modes
Create: orthogonal
Edit: orthogonal

Display Abstraction Controls
□ Merge Shapes
Scope: Pcells
□ Slot Shapes

Dimming
☑ Enable Dimming
Scope: none
□ Dim Grid Lines
☑ Automatic Dimming
Dim Intensity: 50
□ Dim Selected Object Content
□ True Color Selection only

● Cellview ○ Library ○ Tech Library ○ File
~/.cdsenv

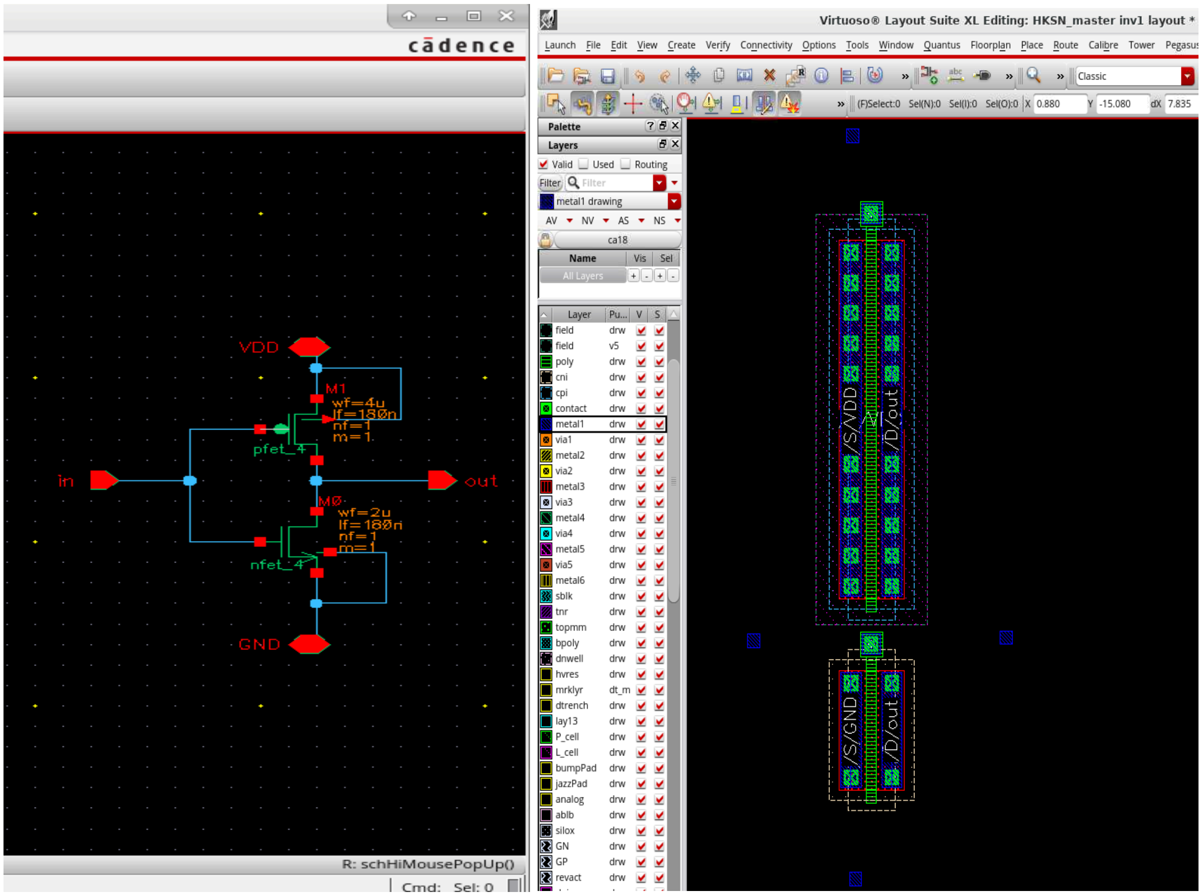
In the layout window, **`Connectivity > Generate > All From Source`** command in is used. This command ensures that connections and details designed in the schematic are automatically reflected in the layout. It generates pins, connections, and other related elements directly from the schematic to the layout.

The main advantages of using this command include:

- **Automatic Information Transfer:** Pins and connections defined in the schematic are seamlessly transferred to the layout, reducing the risk of manual errors.
- **Schematic-Layout Synchronization:** Ensures that the layout stays in sync with the schematic, reflecting any changes made and preventing issues in later design stages.
- **Time Efficiency:** Significantly reduces the time required for layout design by automating many steps.
- **Error Prevention:** Lowers the chances of DRC (Design Rule Check) and LVS (Layout vs. Schematic) errors by ensuring that the layout is correctly generated from the schematic.

These commands are essential for precise and efficient synchronization between schematic and layout, optimizing the design process.

If the placed devices are not immediately visible, press **`Shift + F`** to refresh the view and make the devices visible.

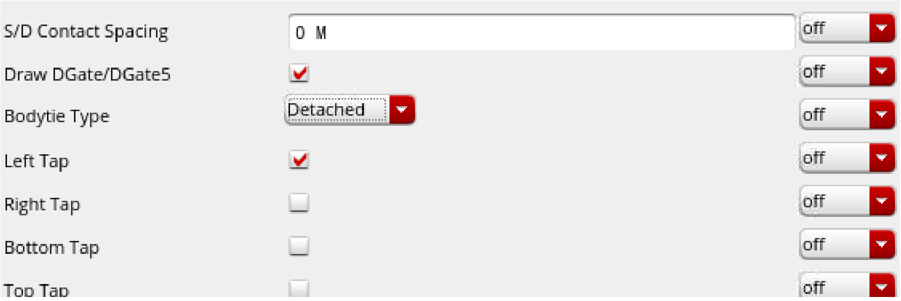

cādence
VDD
M1
wf=4u
lf=180n
nf=1
m=1
pfet_4
in
out
M0
wf=2u
lf=180n
nf=1
m=1
nfet_4
GND
R: schHiMouseePopUp()
Cmd: Sel: 0
Virtuoso® Layout Suite XL Editing: HKSN_master inv1 layout *
Launch File Edit View Create Verify Connectivity Options Tools Window Quantus Floorplan Place Route Calibre Tower Pegasus
Classic
(F)Select:0 Sel(N):0 Sel(I):0 Sel(O):0 X 0.880 Y -15.080 dX 7.835
Palette
Layers
Valid Used Routing
Filter
metal1 drawing
AV NV AS NS
ca18
Name Vis Sel
All Layers
/S/VDD
/D/out
/S/GND
/D/out

Layer	Pu...	V	S
field	drw	✓	✓
field	v5	✓	✓
poly	drw	✓	✓
cni	drw	✓	✓
cpi	drw	✓	✓
contact	drw	✓	✓
metal1	drw	✓	✓
via1	drw	✓	✓
metal2	drw	✓	✓
via2	drw	✓	✓
metal3	drw	✓	✓
via3	drw	✓	✓
metal4	drw	✓	✓
via4	drw	✓	✓
metal5	drw	✓	✓
via5	drw	✓	✓
metal6	drw	✓	✓
sblk	drw	✓	✓
tnr	drw	✓	✓
topmm	drw	✓	✓
bpoly	drw	✓	✓
dnwell	drw	✓	✓
hvres	drw	✓	✓
mrklyr	dt_m	✓	✓
dtrench	drw	✓	✓
lay13	drw	✓	✓
P_cell	drw	✓	✓
L_cell	drw	✓	✓
bumpPad	drw	✓	✓
jazzPad	drw	✓	✓
analog	drw	✓	✓
ablb	drw	✓	✓
silox	drw	✓	✓
GN	drw	✓	✓
GP	drw	✓	✓
revact	drw	✓	✓

At this point, it's necessary to briefly return to the schematic window, select the transistor, press **`(q)`**, and change the **`Bodytie Type`** from "none" to "detached" if it's not already set. This adjustment is essential because, in MOSFETs, the body is typically connected to a reference potential to prevent unwanted effects.

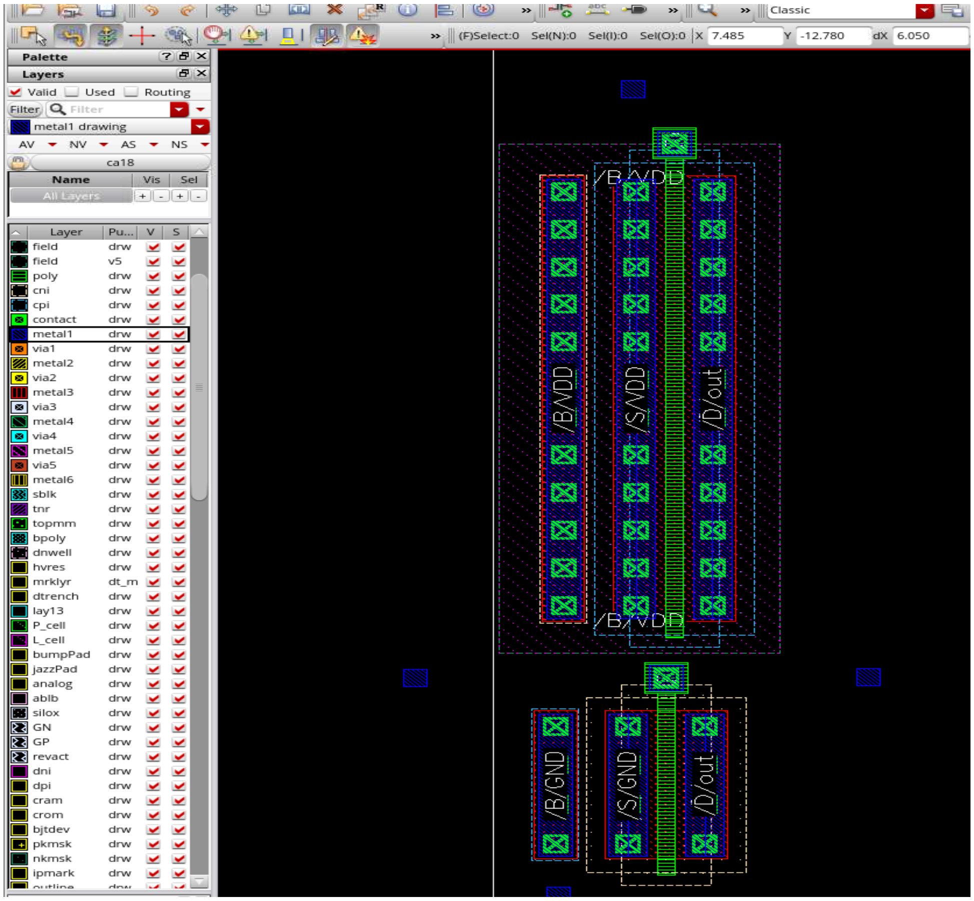
- **`"none"`**: When this option is selected, the transistor's body is not connected to any reference potential. This setting is generally used in circuits where body connection isn't required or is managed externally in higher layers.

- **`"detached":`** Switching the body to "detached" allows for manual connection to a specific potential, such as ground or VDD, enhancing transistor performance and mitigating unwanted effects like leakage currents. Ultimately, this change is made to improve control and precision in circuit design.



S/D Contact Spacing	0 M	off
Draw DGate/DGate5	☑	off
Bodytie Type	Detached	off
Left Tap	☑	off
Right Tap	□	off
Bottom Tap	□	off
Top Tap	□	off

Then, repeat the steps **Connectivity > Generate > All From Source**.



Layer	Pu...	V	S
field	drw	✓	✓
field	v5	✓	✓
poly	drw	✓	✓
cni	drw	✓	✓
cpi	drw	✓	✓
contact	drw	✓	✓
metal1	drw	✓	✓
via1	drw	✓	✓
metal2	drw	✓	✓
via2	drw	✓	✓
metal3	drw	✓	✓
via3	drw	✓	✓
metal4	drw	✓	✓
via4	drw	✓	✓
metal5	drw	✓	✓
via5	drw	✓	✓
metal6	drw	✓	✓
sblk	drw	✓	✓
tnr	drw	✓	✓
topmm	drw	✓	✓
bpoly	drw	✓	✓
dnwell	drw	✓	✓
hvres	drw	✓	✓
mrklyr	dt_m	✓	✓
dtrench	drw	✓	✓
lay13	drw	✓	✓
P_cell	drw	✓	✓
L_cell	drw	✓	✓
bumpPad	drw	✓	✓
jazzPad	drw	✓	✓
analog	drw	✓	✓
ablb	drw	✓	✓
silox	drw	✓	✓
GN	drw	✓	✓
GP	drw	✓	✓
revact	drw	✓	✓
dni	drw	✓	✓
dpi	drw	✓	✓
cram	drw	✓	✓
crom	drw	✓	✓
bjtdev	drw	✓	✓
pkmsk	drw	✓	✓
nkmsk	drw	✓	✓
ipmark	drw	✓	✓
outline	drw	✓	✓

- **Connecting Gates and Adjusting Layout Size**

To connect the gates of two devices, select the `poly` layer, which is the polysilicon layer. Then, press `(r)` to draw a rectangular shape or (create > shape > Rectangle). As you zoom in, you'll notice that the mouse snaps to grid points. Start drawing the layer by typing `(r)` and then pressing `Escape (ESC)` to exit the drawing mode. In some environments of Cadence Virtuoso, the `P` key is used to create a "Wire" or "Path". By pressing the `P` key, you can draw a route or wiring between two points in your circuit. Additionally, while drawing the path, you can adjust parameters such as "Wire Width, Layer, and Orientation".

After drawing the rectangle, you might observe that it is slightly larger than the gate of the devices. This discrepancy could lead to a **DRC (Design Rule Check)** error. To match the rectangle to the gate size (e.g., 180 nanometers), a slight adjustment or stretching is required.

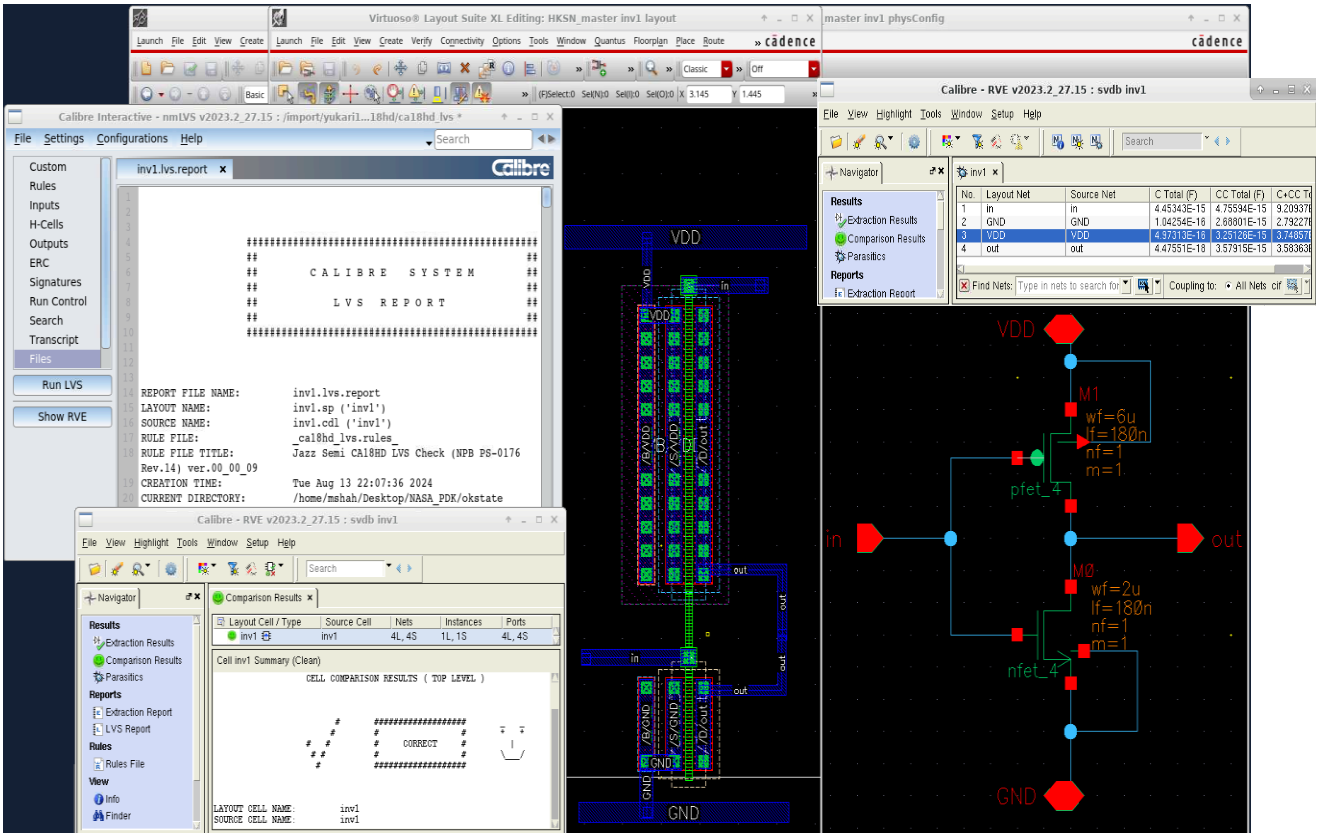
To stretch the rectangle, click on a blank area to ensure no objects are selected. Then, right-click and choose **Stretch** from the context menu, or simply press **(s)** for the shortcut. When using the stretch, hover near the edge of the rectangle you've drawn. The edge will highlight, indicating it is ready to be modified. Click on the highlighted edge and move the mouse to adjust the size, aligning it with the gate of the PMOS device.

For zooming into a specific area, right-click and drag to focus on the desired section. To frame the entire view, press **(f)**. These shortcut options are highly useful for efficient layout design.

- **Connecting PMOS and NMOS Drains**

In this step, we'll connect the drain of the PMOS device to the drain of the NMOS device using the **Metal 1** layer. To do this, select **Metal 1** and press **(r)** to draw the connection. After positioning the connection, press **(ESC)** to exit the drawing mode. Once the connection is completed, both drains will be properly linked.

After completing all the connections, save the design and check it with DRC (Design Rule Check) and LVS (Layout Versus Schematic). Begin by running these checks to ensure the design adheres to the required fabrication standards.



Virtuoso® Layout Suite XL Editing: HKSN_master inv1 layout
master inv1 physConfig
Launch File Edit View Create Verify Connectivity Options Tools Window Quantus Floorplan Place Route
cādence
Classic
Off
Basic
(F)Select:0 Sel(N):0 Sel(I):0 Sel(O):0 X 3.145 Y 1.445
Calibre Interactive - nmLVS v2023.2_27.15 : /import/yukari1...18hd/ca18hd_lvs *
File Settings Configurations Help
Search
Custom
Rules
Inputs
H-Cells
Outputs
ERC
Signatures
Run Control
Search
Transcript
Files
Run LVS
Show RVE
inv1.lvs.report
Calibre
VDD
in
out
GND
Calibre - RVE v2023.2_27.15 : svdb inv1
File View Highlight Tools Window Setup Help
Navigator
Results
Extraction Results
Comparison Results
Parasitics
Reports
Extraction Report
LVS Report
Rules
Rules File
View
Info
Finder
Comparison Results
inv1
Find Nets: Type in nets to search for
Coupling to: All Nets cif
VDD
M1
wf=6u
lf=180n
nf=1
m=1
pfet_4
in
out
M0
wf=2u
lf=180n
nf=1
m=1
nfet_4
GND

```
##################################################
##                                              ##
##          C A L I B R E   S Y S T E M         ##
##                                              ##
##             L V S   R E P O R T              ##
##                                              ##
##################################################



REPORT FILE NAME:         inv1.lvs.report
LAYOUT NAME:              inv1.sp ('inv1')
SOURCE NAME:              inv1.cdl ('inv1')
RULE FILE:                _ca18hd_lvs.rules_
RULE FILE TITLE:          Jazz Semi CA18HD LVS Check (NPB PS-0176
Rev.14) ver.00_00_09
CREATION TIME:            Tue Aug 13 22:07:36 2024
CURRENT DIRECTORY:        /home/mshah/Desktop/NASA_PDK/okstate
```

Layout Cell / Type	Source Cell	Nets	Instances	Ports
inv1	inv1	4L, 4S	1L, 1S	4L, 4S

Cell inv1 Summary (Clean)

```
              CELL COMPARISON RESULTS ( TOP LEVEL )



                 #         ###################
                #          #                 #
            #   #          #     CORRECT     #
             # #           #                 #
              #            ###################


LAYOUT CELL NAME:         inv1
SOURCE CELL NAME:         inv1
```

No.	Layout Net	Source Net	C Total (F)	CC Total (F)	C+CC To
1	in	in	4.45343E-15	4.75594E-15	9.20937E
2	GND	GND	1.04254E-16	2.68801E-15	2.79227E
3	VDD	VDD	4.97313E-16	3.25126E-15	3.74857E
4	out	out	4.47551E-18	3.57915E-15	3.58363E

Design Rule Check (DRC):

DRC verifies that the physical layout of the design meets all the manufacturing constraints set by the foundry, such as minimum metal widths, spacing between components, and layer densities. Any violation of these rules could result in a non-functional or difficult-to-manufacture chip.

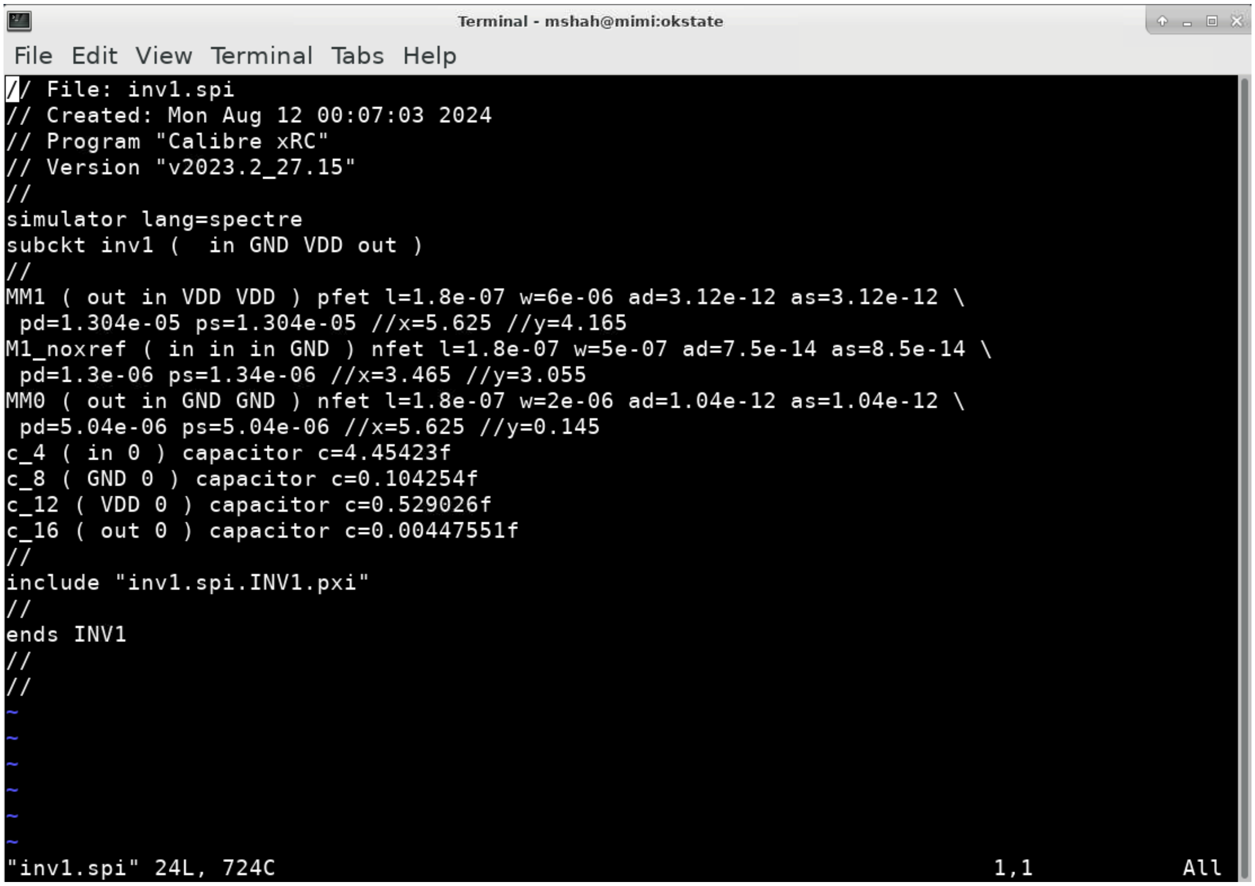
Layout Versus Schematic (LVS):

LVS ensures that the layout accurately reflects the schematic, verifying that all electrical connections and component placements match the intended design. This check is crucial for confirming that the final fabricated chip will perform as expected.

These steps are essential to validate the design's integrity and manufacturability before proceeding to the fabrication process.

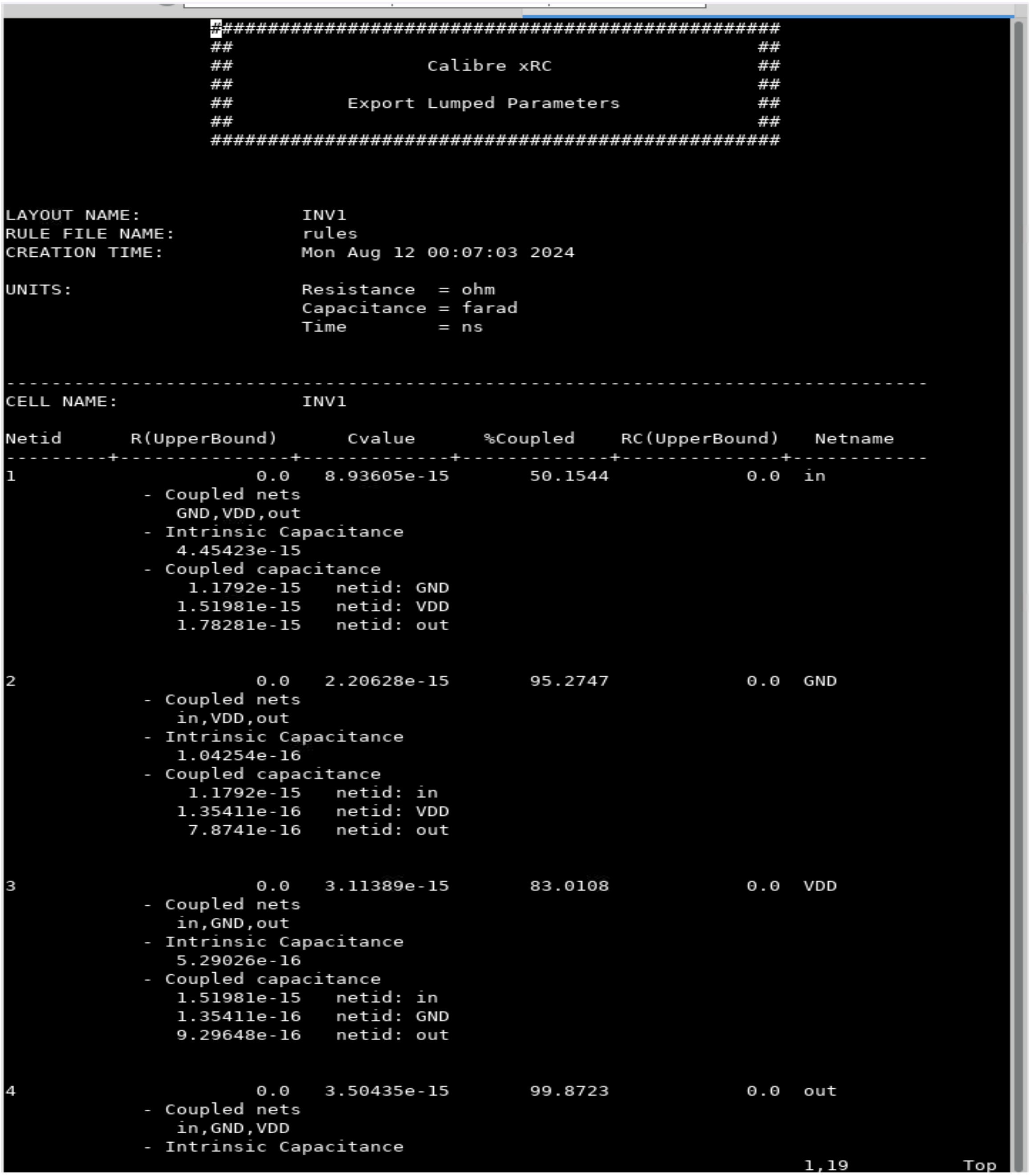
Extracted Resistance and Capacitance (xRC):

Now, the parasitic resistances and capacitances need to be extracted; therefore, using the **Calibre xRC** tool for this extraction.


Terminal - mshah@mimi:okstate
File Edit View Terminal Tabs Help

```
// File: inv1.spi
// Created: Mon Aug 12 00:07:03 2024
// Program "Calibre xRC"
// Version "v2023.2_27.15"
//
simulator lang=spectre
subckt inv1 (  in GND VDD out )
//
MM1 ( out in VDD VDD ) pfet l=1.8e-07 w=6e-06 ad=3.12e-12 as=3.12e-12 \
 pd=1.304e-05 ps=1.304e-05 //x=5.625 //y=4.165
M1_noxref ( in in in GND ) nfet l=1.8e-07 w=5e-07 ad=7.5e-14 as=8.5e-14 \
 pd=1.3e-06 ps=1.34e-06 //x=3.465 //y=3.055
MM0 ( out in GND GND ) nfet l=1.8e-07 w=2e-06 ad=1.04e-12 as=1.04e-12 \
 pd=5.04e-06 ps=5.04e-06 //x=5.625 //y=0.145
c_4 ( in 0 ) capacitor c=4.45423f
c_8 ( GND 0 ) capacitor c=0.104254f
c_12 ( VDD 0 ) capacitor c=0.529026f
c_16 ( out 0 ) capacitor c=0.00447551f
//
include "inv1.spi.INV1.pxi"
//
ends INV1
//
//
~
~
~
~
~
~
"inv1.spi" 24L, 724C                                          1,1          All
```

xRC plays a critical role in the post-layout verification process by accurately modeling parasitic resistances and capacitances that arise during the physical implementation of circuits. These parasitic elements can significantly affect the performance, timing, and power consumption of integrated circuits, making xRC extraction an essential step in ensuring that the final design meets all specifications. By incorporating xRC into the design flow, engineers can perform detailed analyses that account for these parasitics, leading to more reliable and efficient designs. The ability to accurately predict the behavior of circuits under real-world conditions helps avoid costly iterations and ensures that the final product performs as intended. In essence, xRC serves as a bridge between theoretical design and practical implementation, making it an indispensable tool in modern electronic design automation (EDA).



```
                  ##################################################
                  ##                                              ##
                  ##                 Calibre xRC                  ##
                  ##                                              ##
                  ##          Export Lumped Parameters            ##
                  ##                                              ##
                  ##################################################



LAYOUT NAME:              INV1
RULE FILE NAME:           rules
CREATION TIME:            Mon Aug 12 00:07:03 2024

UNITS:                    Resistance  = ohm
                          Capacitance = farad
                          Time        = ns


--------------------------------------------------------------------------------
CELL NAME:                INV1

Netid      R(UpperBound)      Cvalue      %Coupled    RC(UpperBound)   Netname
---------+---------------+-------------+-------------+--------------+------------
1                     0.0   8.93605e-15       50.1544            0.0  in
            - Coupled nets
               GND,VDD,out
            - Intrinsic Capacitance
               4.45423e-15
            - Coupled capacitance
                1.1792e-15   netid: GND
               1.51981e-15   netid: VDD
               1.78281e-15   netid: out


2                     0.0   2.20628e-15       95.2747            0.0  GND
            - Coupled nets
               in,VDD,out
            - Intrinsic Capacitance
               1.04254e-16
            - Coupled capacitance
                1.1792e-15   netid: in
               1.35411e-16   netid: VDD
                7.8741e-16   netid: out


3                     0.0   3.11389e-15       83.0108            0.0  VDD
            - Coupled nets
               in,GND,out
            - Intrinsic Capacitance
               5.29026e-16
            - Coupled capacitance
               1.51981e-15   netid: in
               1.35411e-16   netid: GND
               9.29648e-16   netid: out


4                     0.0   3.50435e-15       99.8723            0.0  out
            - Coupled nets
               in,GND,VDD
            - Intrinsic Capacitance
                                                                      1,19          Top
```