EE-103 VLSI Design
Custom Compiler Tutorial

Kaiyuan Zhang
kzhang11@tufts.edu

Tufts UNIVERSITY
SCHOOL OF ENGINEERING
Electrical and Computer Engineering

Outline

- Introduction
- Setup Design Workspace
- Creating an Inverter Netlist

Introduction

This tutorial will teach you how to draw a transistor-level schematic using the Synopsys Custom Compiler and output the design into a netlist for HSPICE simulation.

- It is not feasible to directly write HSPCIE netlist for a large circuit.
- Custom Compiler has debugging capability
- You can use Custom Compiler to generate other types of netlist as well.
- You can use Custom Compiler to generate layout and do post-layout analysis.
- In fact, you can use Custom Compiler to do full custom IC design.

Setup Design Workspace

1. Make sure load modules properly
   module use /usr/cots/modulefiles
   module load licenses/synopsys tools/hspice/K-2015.06-3
2. load the synopsys tool list
   use synopsys
3. Pop up the GUI interface of Custom Compiler
   cdesigner

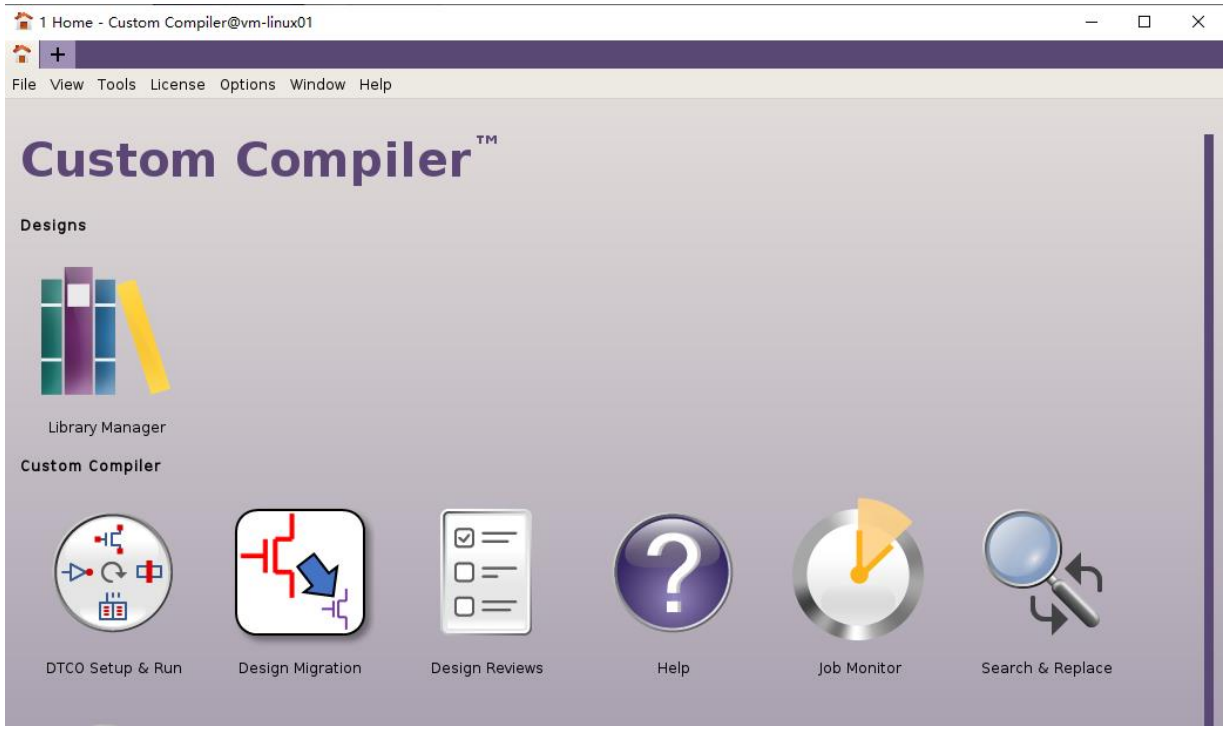
4. Download “SAED32nm_PDK_02_2024” from canvas and save the file into your home directory. (/h/YOUR_USERNAME/)

Setup Design Workspace

To run an instance of Custom Compiler, simply type “cdesigner”. Your Custom Compiler should look like the one show below.

OR

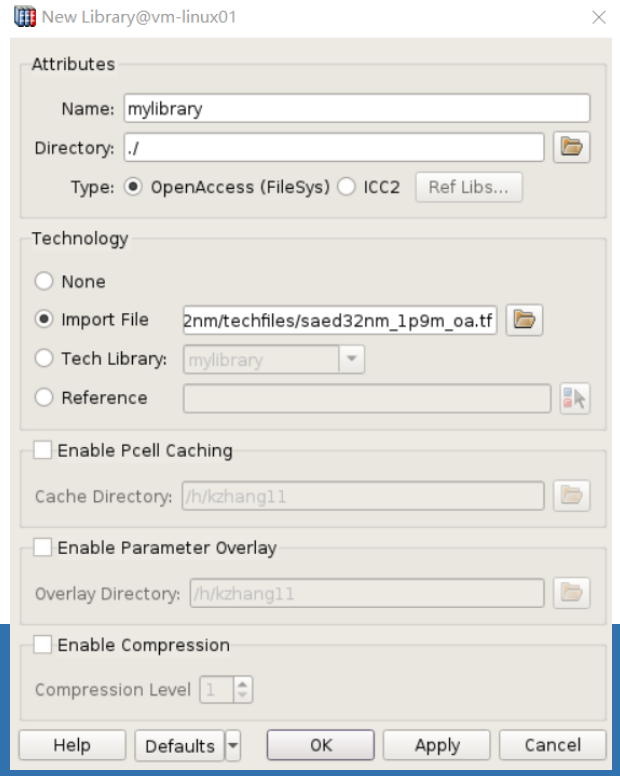
```
cd SAED32nm_PDK_02_2024
cdesigner &
```



Creating ‘mylibrary’

In the **Library Manager**, use **File > New > Library** to create a new library.

- Fill out the name filed with a library name, in this case, ‘mylibrary’.
- For the “Import File” filed, click the dot and click the folder button so you can select the file. The technology file is in the directory: SAED32nm_PDK_02_2024/techfiles/saed32nm_1p9m_oa.tf
- After you have filled out the name, directory and import file, click **OK.**



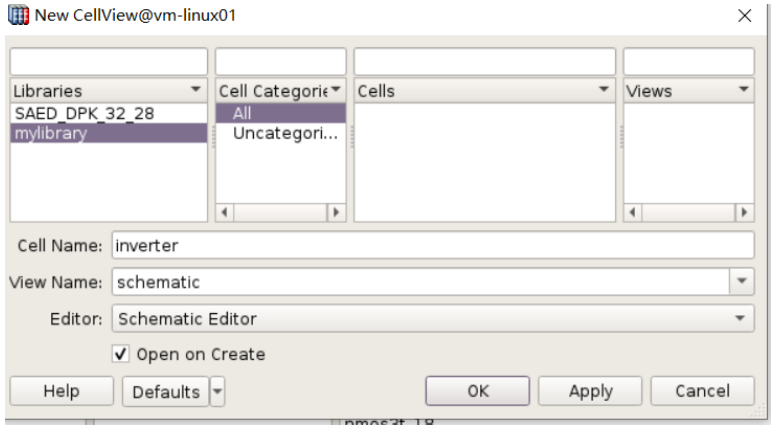
Creating CellView

In the **Library Manager**, use **File > New > CellView** to create a new Cell View under the library that you create in last step.

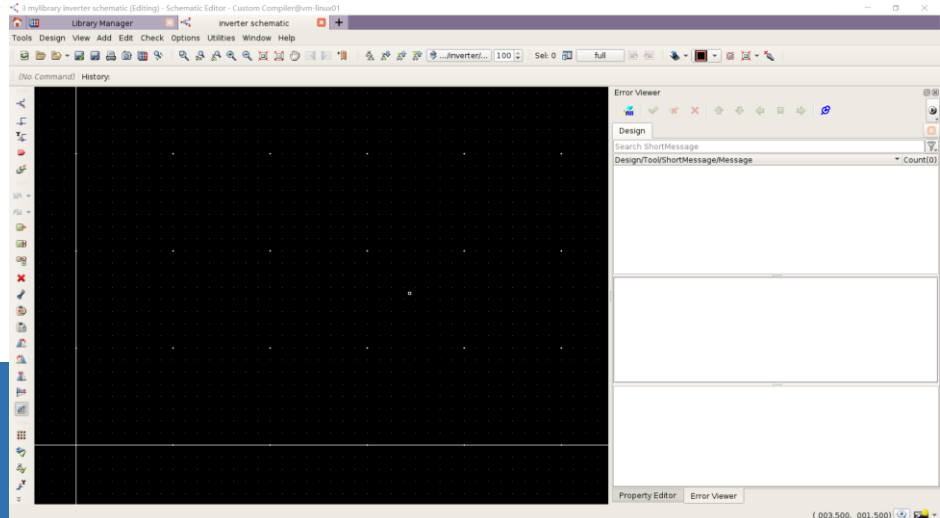
Enter a name for ‘Cell Name’ and choose schematic for ‘View Name’.

For ‘editor’ choose Schematic Editor.

Click OK.



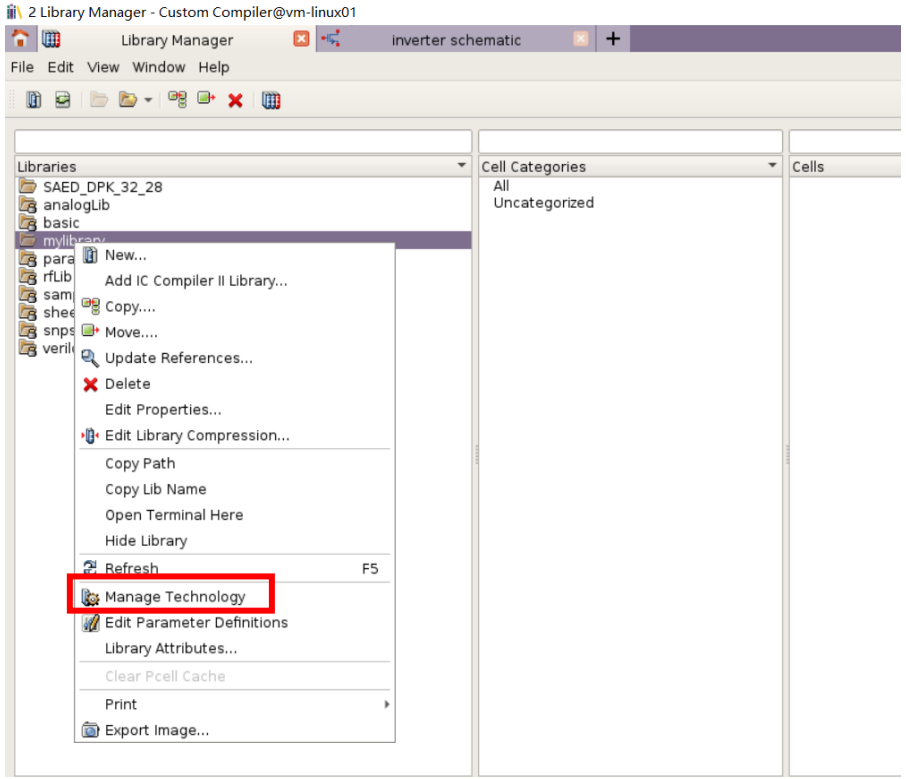
After you click OK, a new window will open up called schematic editor.



Creating an Inverter Netlist

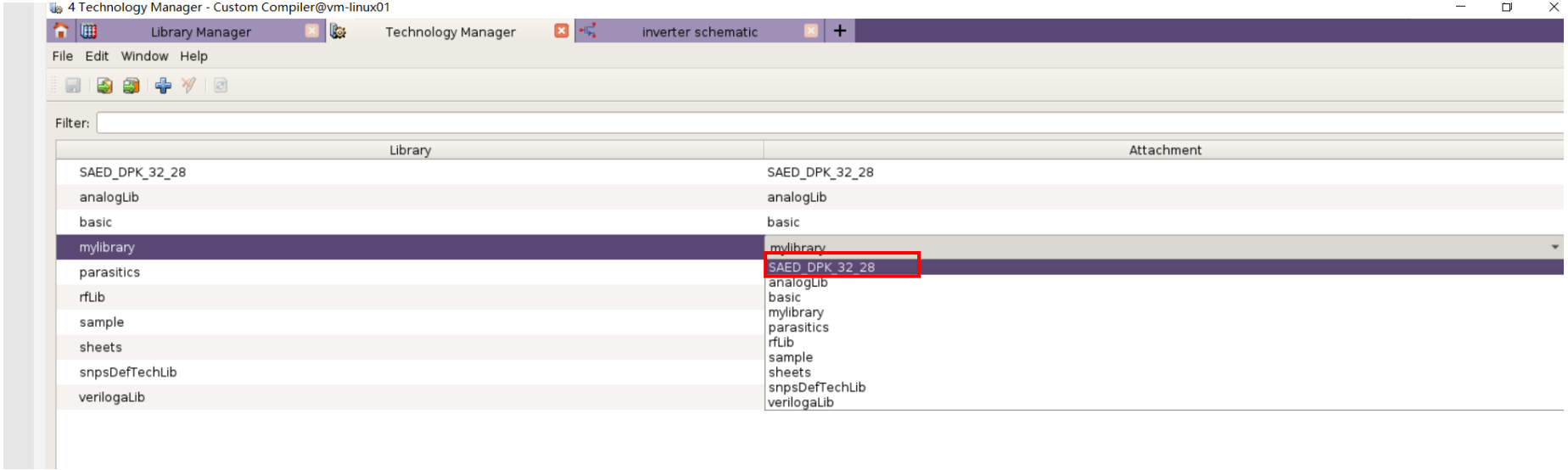
Here we need to make sure that we are in the correct library.

From Library Manager open the Manage Technology for the created library, mylibrary.

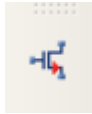


Creating an Inverter Netlist

Make sure the attachment of your library is ‘SAED_PDK_32_28’.



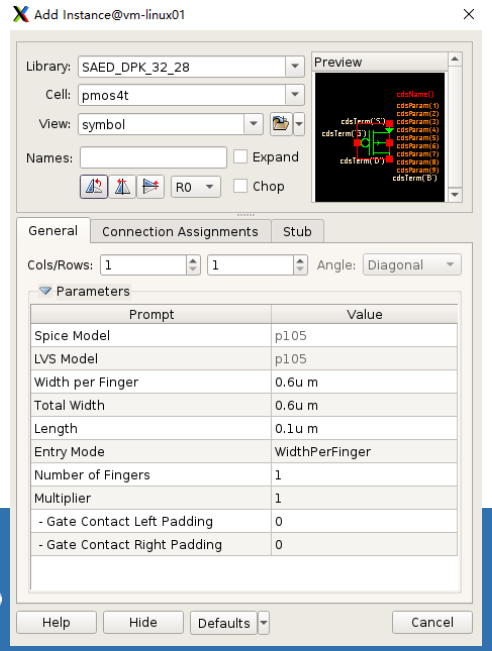
Creating an Inverter Netlist

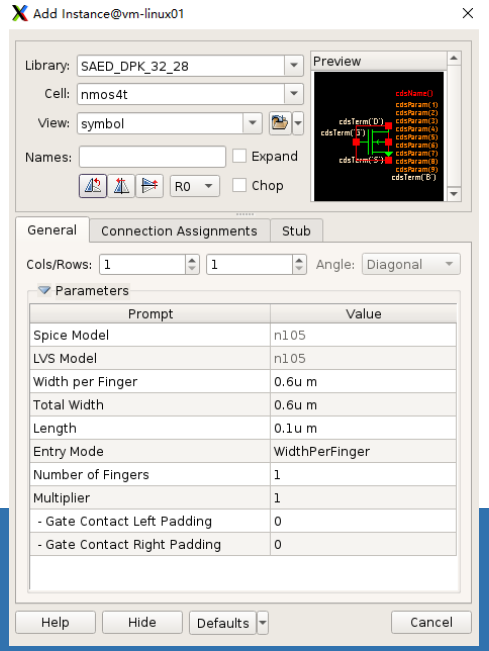
Now back to Schematic Editor, click Add > Instance or click on this icon 

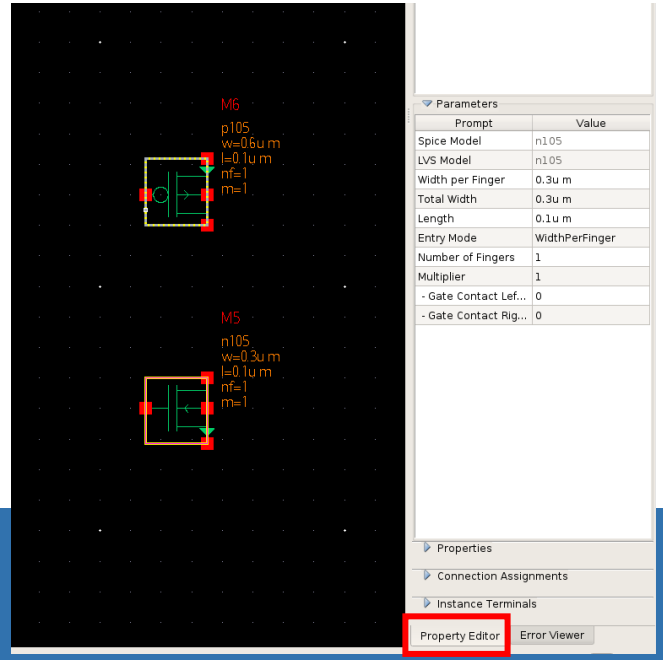
Select ‘SAED_PDK_32_28’ for the library, select ‘pmos4t’ and ‘nmos4t’ for the cell while placing their respective parts on the schematic. Assign 0.6um for pmos4t width, 0.3um for nmos4t width, and assign 0.1um for both length.

You can do this by modifying the value at the ‘Width per Finger’ box and the ‘Length’ box.

You can also modify these properties later using the property editor by going to **Edit > Properties > Assistant** and selecting the component you want to modify in schematic view.







Creating an Inverter Netlist

After placing the PMOS and NMOS transistors, the schematic should look like the figure below.

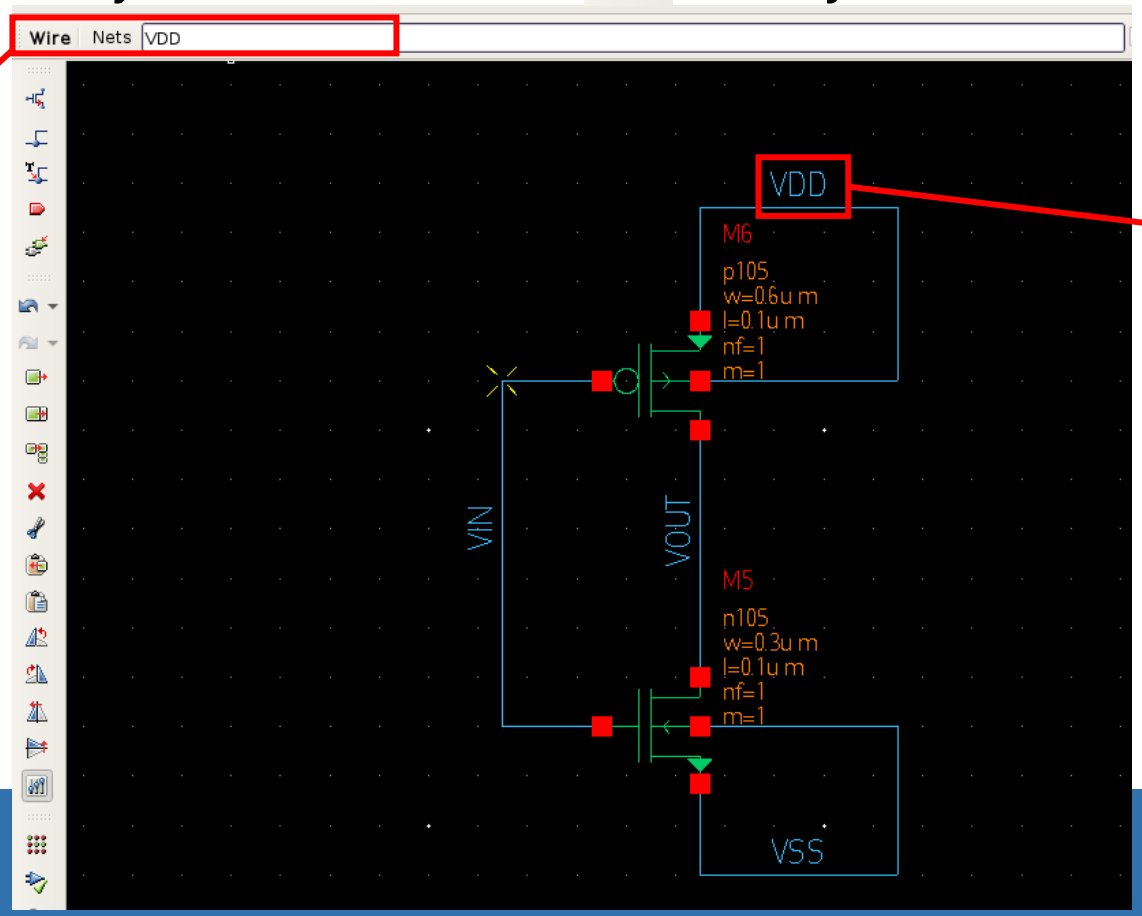
Hint: Press “ESC” to finish an action

Creating an Inverter Netlist

Next, add wires to the schematic. Click  and draw wires to the circuit using the mouse pointer.

To deselect wire adding, press ‘ESC’.

Now you need to give names to your wires. Click on  ,then you will see a wire name space on top to enter wire names.



Creating an Inverter Netlist

Go to **ADD > Pin** or click the  to add pins for input (VIN,VDD,VSS) and output (VOUT) for your schematic. You can type in a name for the pin and select whether the pin is an input or output port.

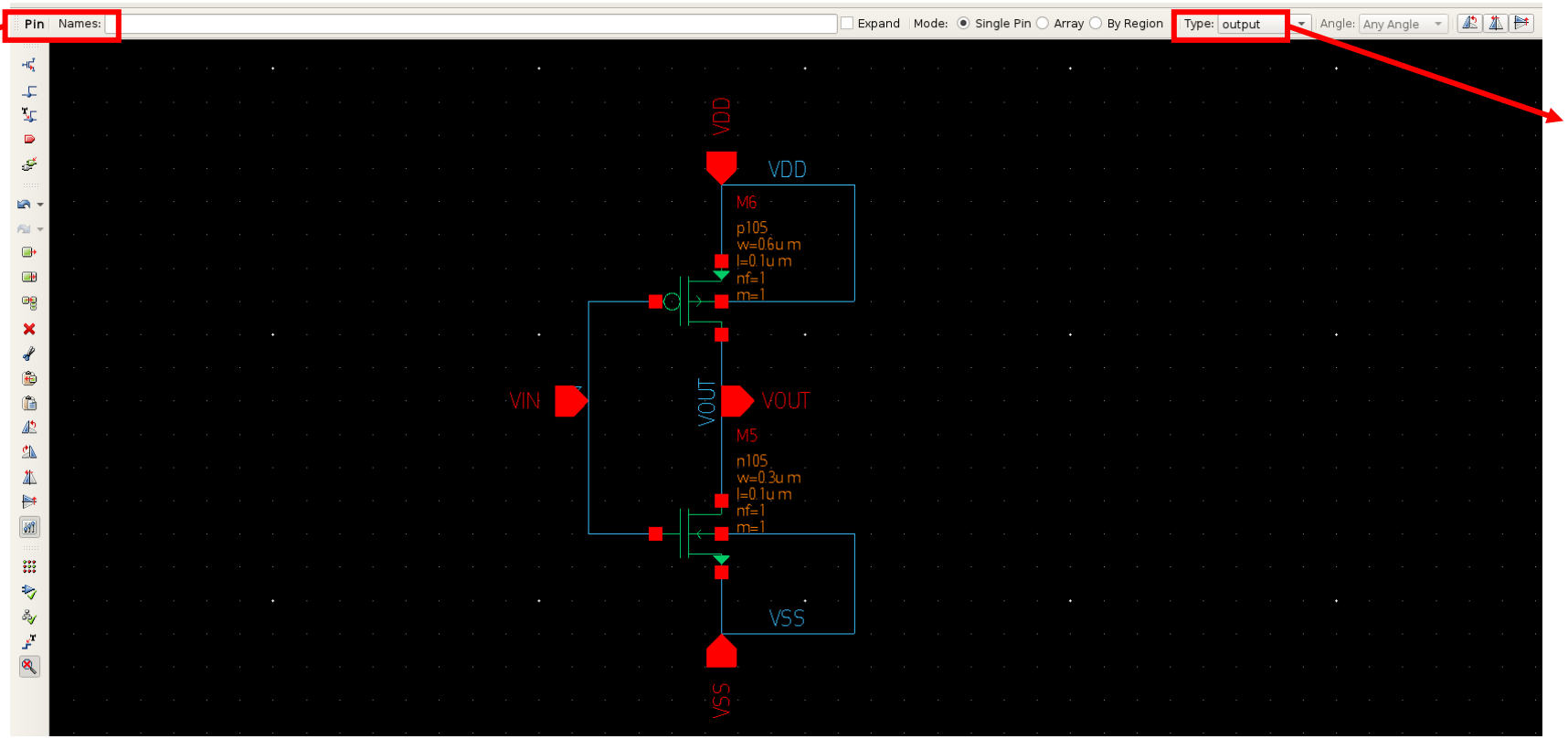
Then place the pin on a wire or if the wire in schematic view already has a name you can click on the wire and the pin will get the name of the wire.

You can right click the mouse and the symbol of the pin will rotate.

As a general convention, use uppercase letters for naming pins instead of lowercase.

Creating an Inverter Netlist

Give Names to Pins



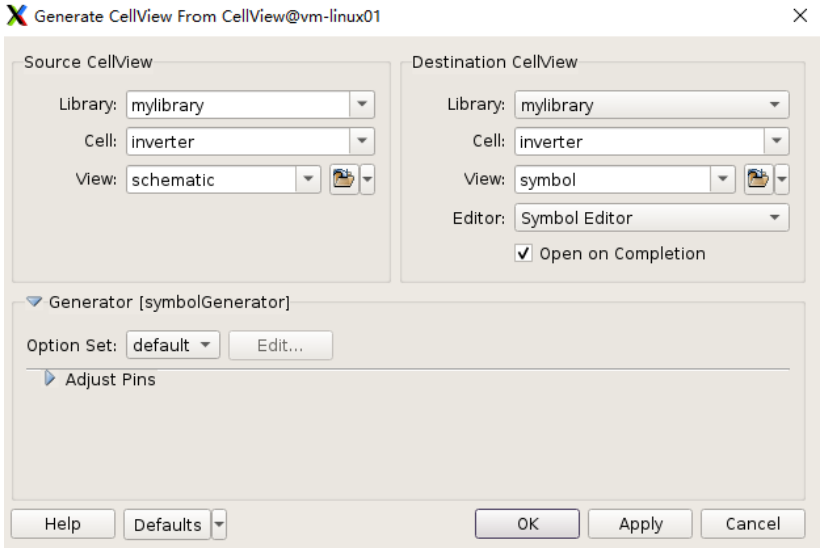
Select Input nodes or output nodes before placing

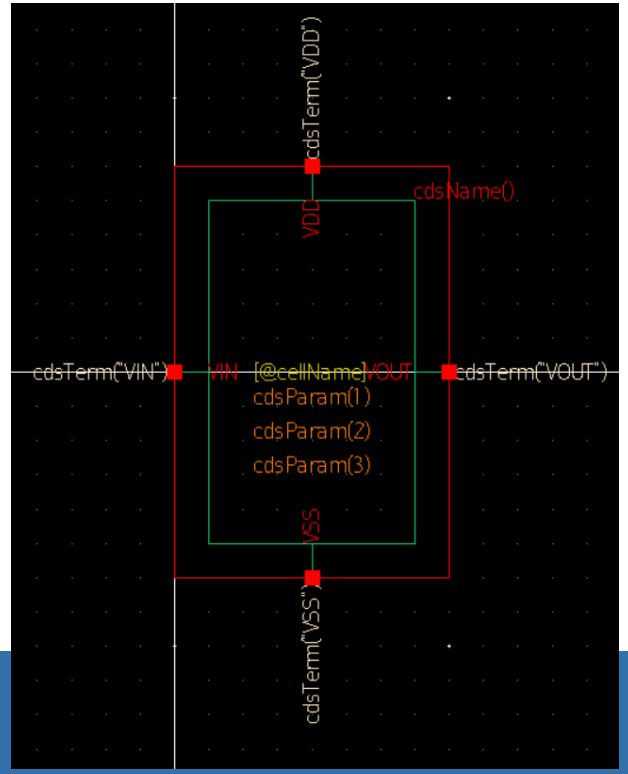
Save your schematic by clicking  or go to **Design > Save**.

Creating an Inverter Netlist

Now we want to create symbol for the inverter schematic to use for future designs instead of redrawing it every time. To create a symbol for the inverter, go to **Design > New CellView > From CellView**. Make sure library and cell names match and click **OK**.

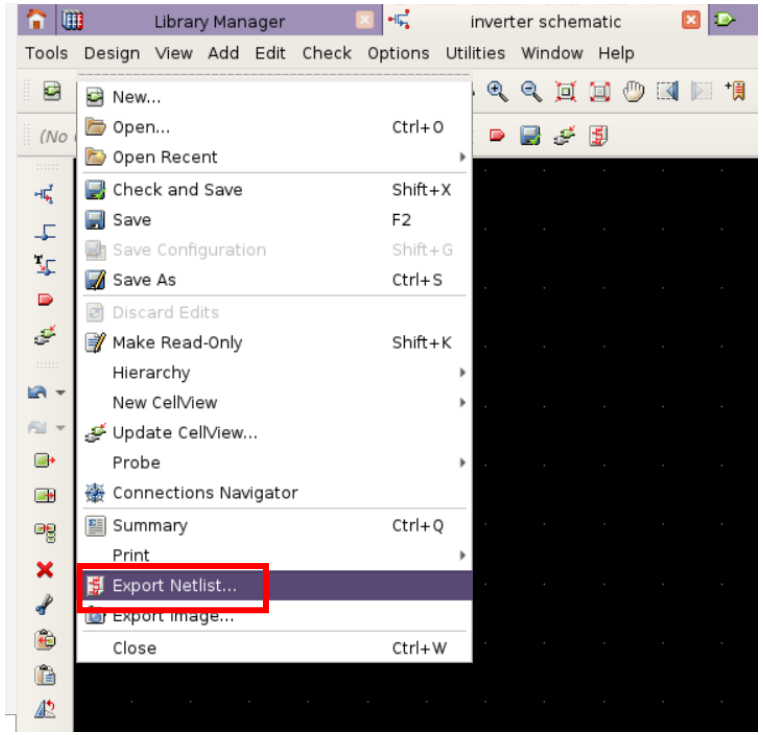
Now we have transistor level model of an inverter (symbol).

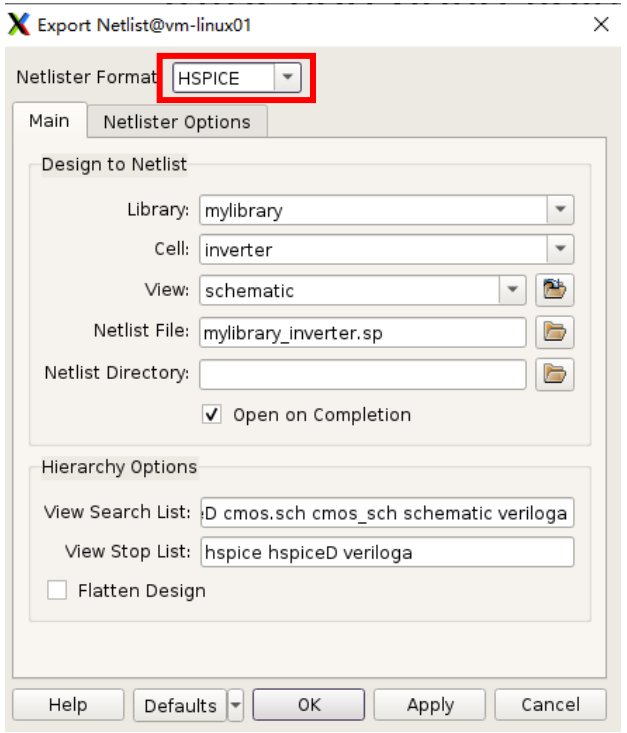




Creating an Inverter Netlist

After you have finished your schematic design you need to output it to a netlist in order to do HSPICE simulation. To do this, go to **Design > Export Netlist**.

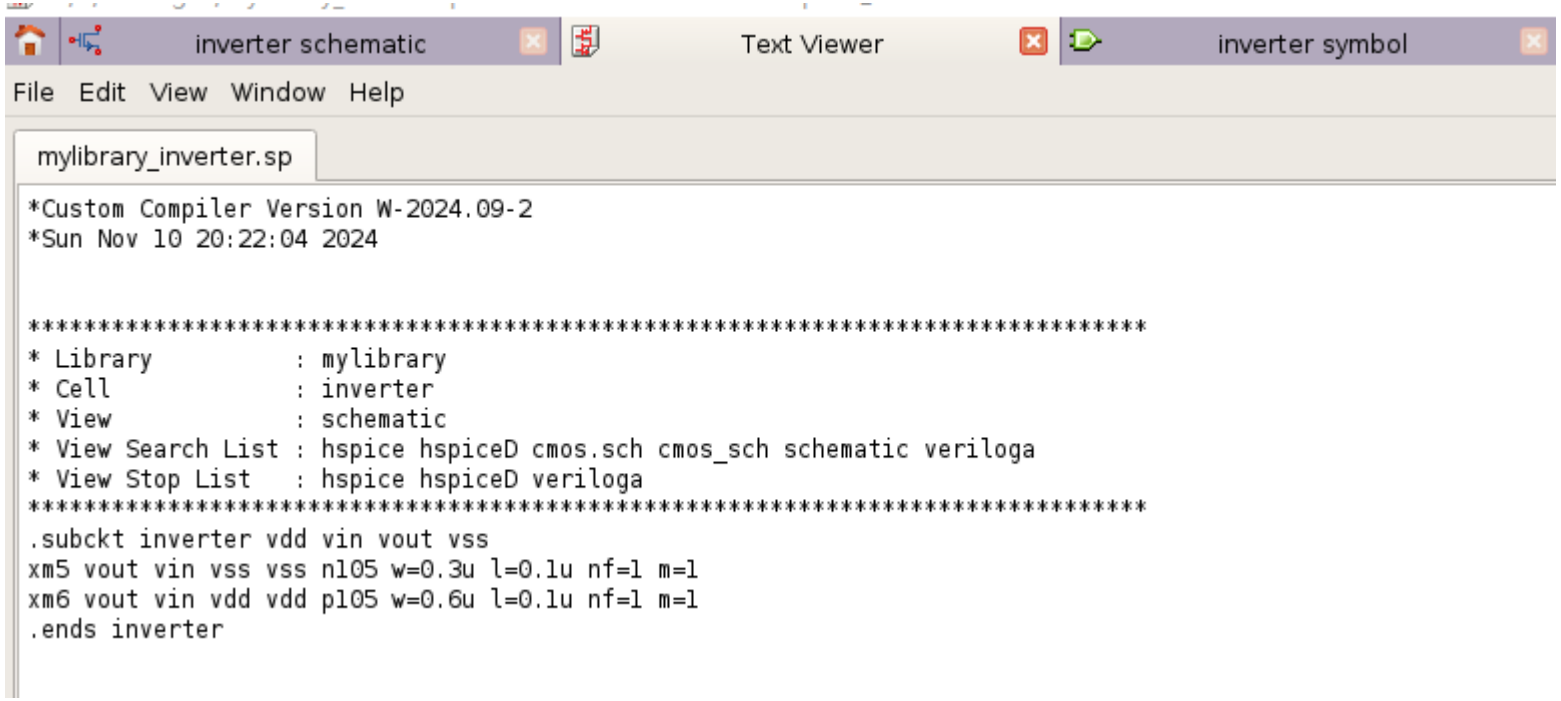




Creating an Inverter Netlist

Then a netlist file will be generated. You can find the output file (mylibrary_inverter.sp in this case) under the directory where you opened your Custom Compiler.

You can write a simple HSPICE script to test the netlist.



Creating an Inverter Netlist

Notice in this generated file, it used ‘p105’ and ‘n105’ to represent the PMOS and NMOS, which are not the formal model representations. Thus, don't forget to define them in the HSPICE simulation script shown below:

.model n105 nmos level=54

.model p105 pmos level=54

```
.lib '/usr/cots/synopsys/UniversityLibrary/SAED32_EDK/tech/hspice/saed32nm.lib' TT
.include '/usr/cots/synopsys/UniversityLibrary/SAED32_EDK/lib/stdcell_rvt/hspice/saed32nm_rvt.spf'

*post the results
.option post
.global vdd gnd

*define model
.model n105 nmos level=54
.model p105 pmos level=54

*source declaration
vvdd vdd 0 vdd   *syntax:vname pos_node neg_node voltage_value
vgnd gnd 0 0v

*M1 vo vi gnd gnd n105 W=300n L=1
*M2 vo vi vdd vdd p105 W=300n L=1

*.subckt inverter vdd vi vo gnd
*xm5 vo vi gnd gnd n105 w=0.3u l=0.1u nf=1 m=1
*xm6 vo vi vdd vdd p105 w=0.6u l=0.1u nf=1 m=1
*.ends inverter

xinv0 vi vo inverter
```

Delay measurement of your inverter

Copy and paste the subcircuit into your HSPICE code. Don't forget to instantiate it!

Subcircuit generated by Custom Compiler

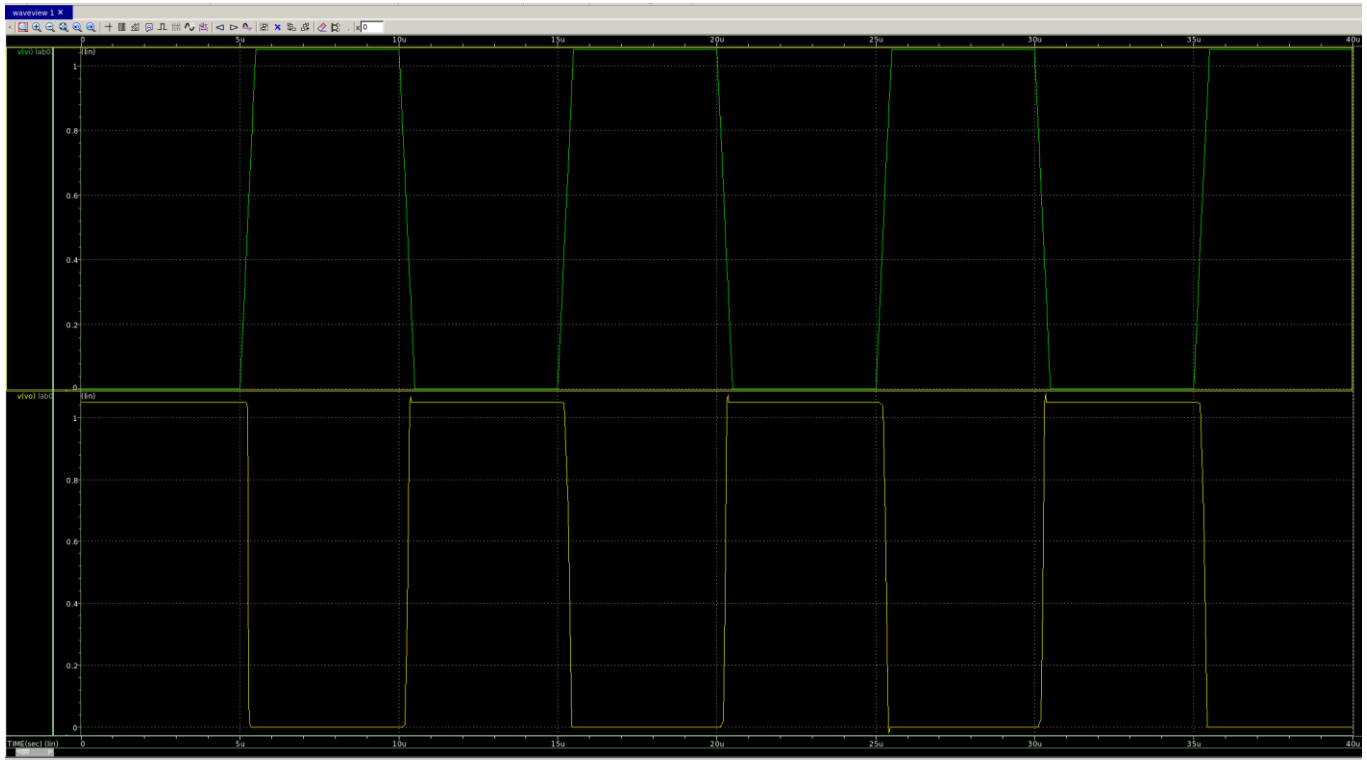
```
*library file
.lib '/usr/cots/synopsys/UniversityLibrary/SAED32_EDK/tech/hspice/saed32nm.lib' TT
.include '/usr/cots/synopsys/UniversityLibrary/SAED32_EDK/lib/stdcell_rvt/hspice/saed32nm_rvt.spf'

*post the results
.option post
.global vdd gnd

*define model
.model n105 nmos level=54
.model p105 pmos level=54

*source declaration
vvdd vdd 0 vdd   *syntax:vname pos_node neg_node voltage_value
vgnd gnd 0 0v

*M1 vo vi gnd gnd n105 W=300n L=1
*M2 vo vi vdd vdd p105 W=300n L=1

.subckt inverter vdd vin vout vss
xm5 vout vin vss vss n105 w=0.3u l=0.1u nf=1 m=1
xm6 vout vin vdd vdd p105 w=0.6u l=0.1u nf=1 m=1
.ends inverter

xinv0 vdd vi vo gnd inverter
```

Instantiate the subcircuit

Delay measurement of your inverter

Same input setting with Lab03 demo:

vinput vi gnd pulse 0 vdd 5u 0.5u 0.5u 4.5u 10u



```
****** transient analysis tnom=  25.000 temp=  25.000 ******
outrise= 100.3587n  targ=  20.3130u   trig=  20.2127u
outfall= 186.4116n  targ=  15.4278u   trig=  15.2414u
tphl= 106.4353n  targ=  15.3564u   trig=  15.2500u
tplh=  23.2822n  targ=  20.2733u   trig=  20.2500u
```

Tufts UNIVERSITY

Thank you!