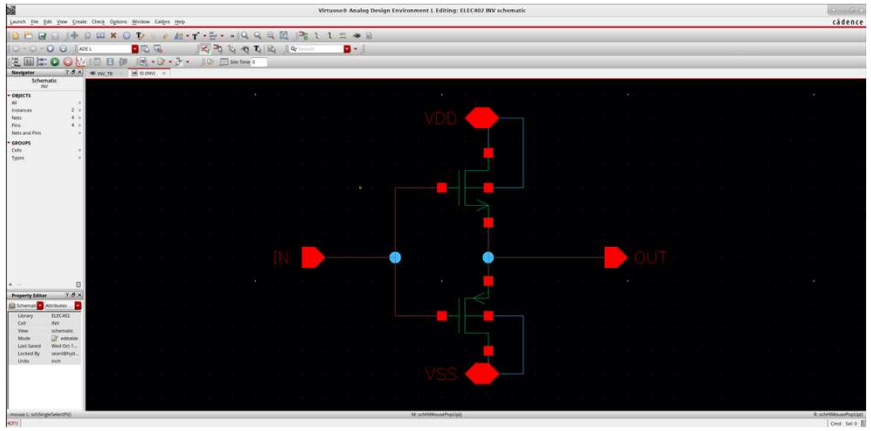
Tutorial 4
Cadence Virtuoso Layout

Sean Lam
seanlm@student.ubc.ca

Slides Courtesy of Omid Esmaeeli

Outline

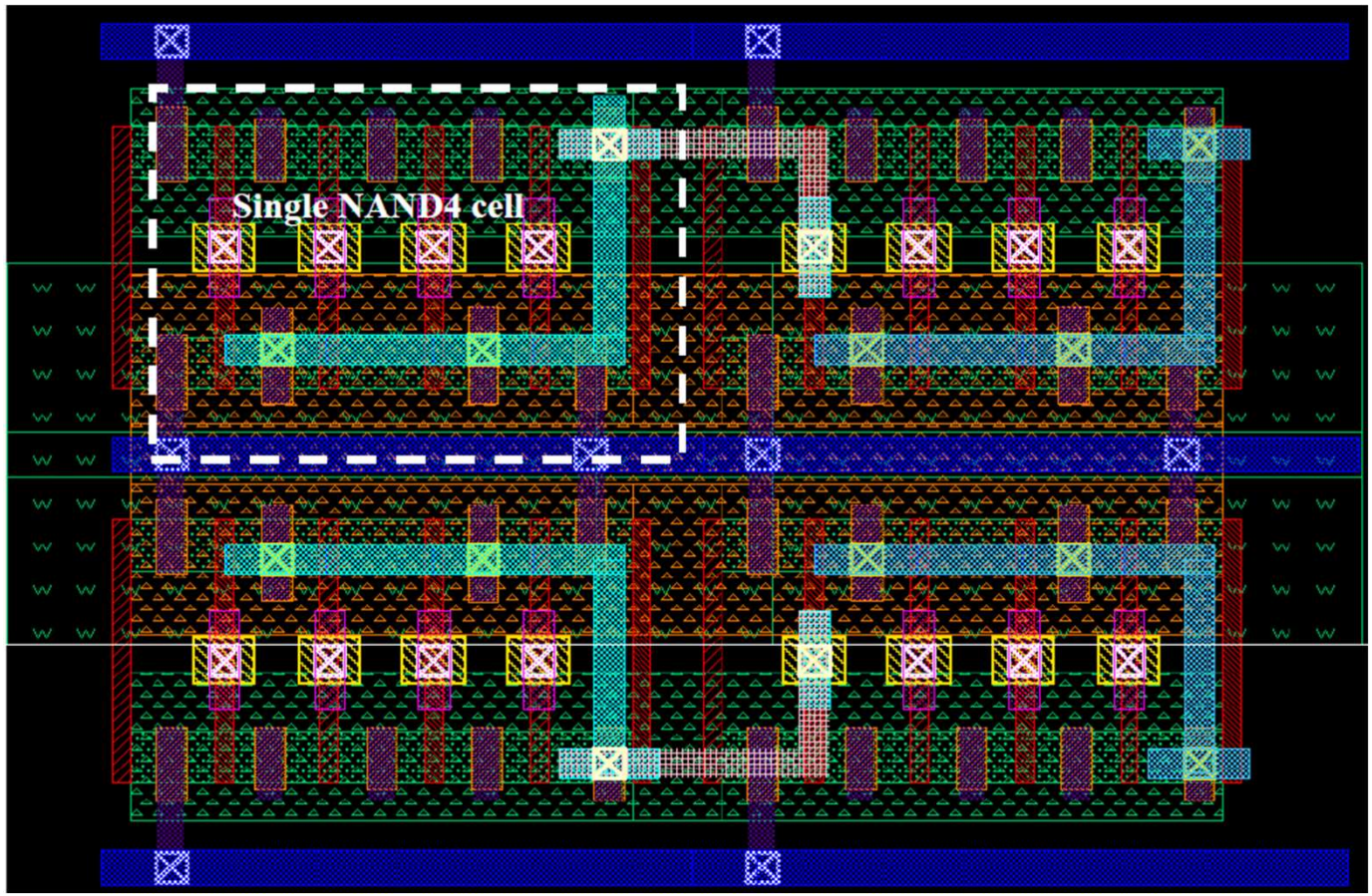
Goal: Create a layout, run DRC (Design Rule Check), run LVS (Layout vs. Schematic), and run PEX (Parasitic Extraction)



1. Understanding the process and technology node
2. Creating a layout
3. Run DRC, LVS, and PEX

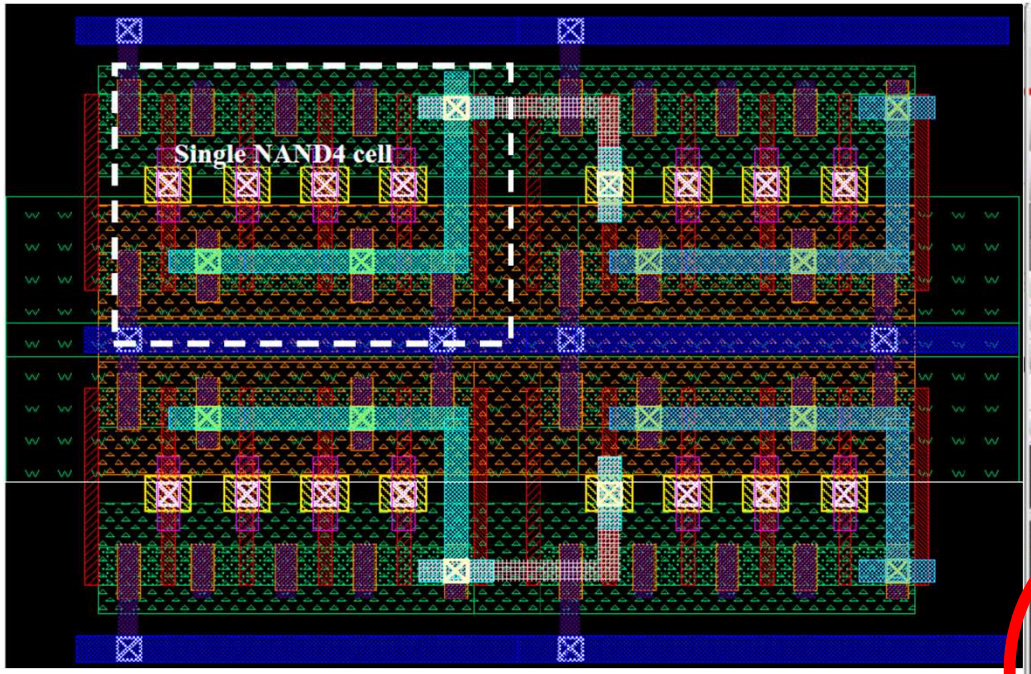
Introduction to Layout

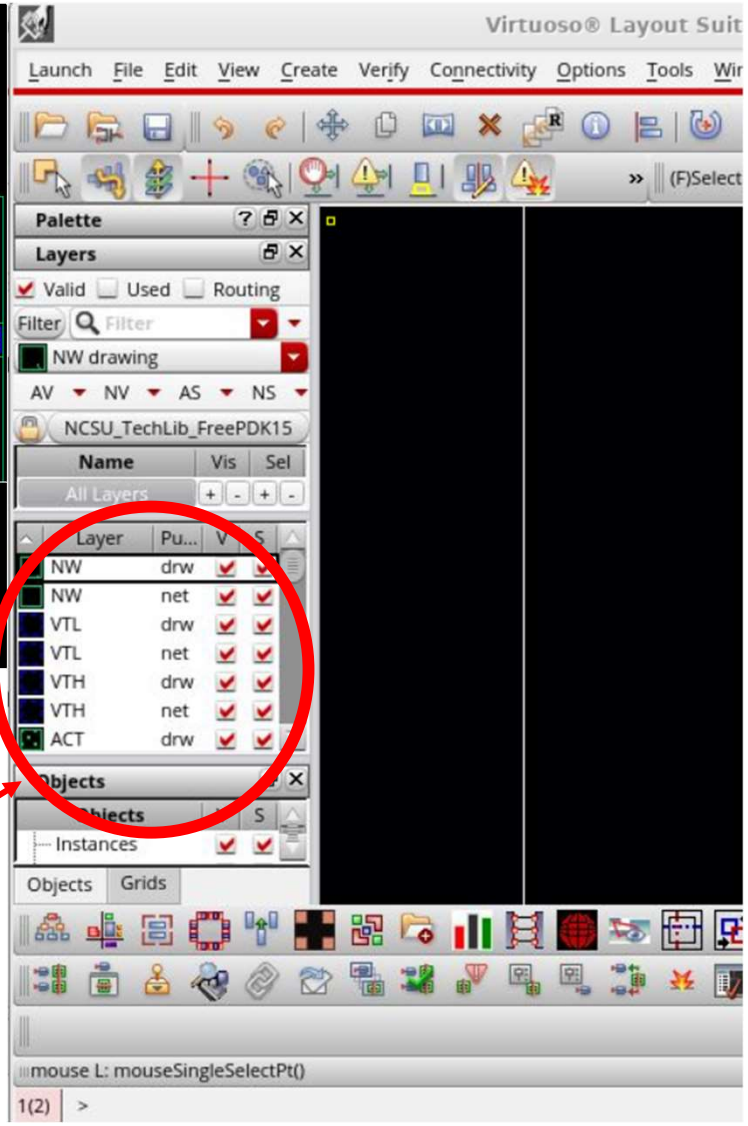
Layout is describing transistor and circuit functionality in 2D/3D



Ref:
https://www.eda.ncsu.edu/wiki/FreePDK15:Contents

Introduction to Layout



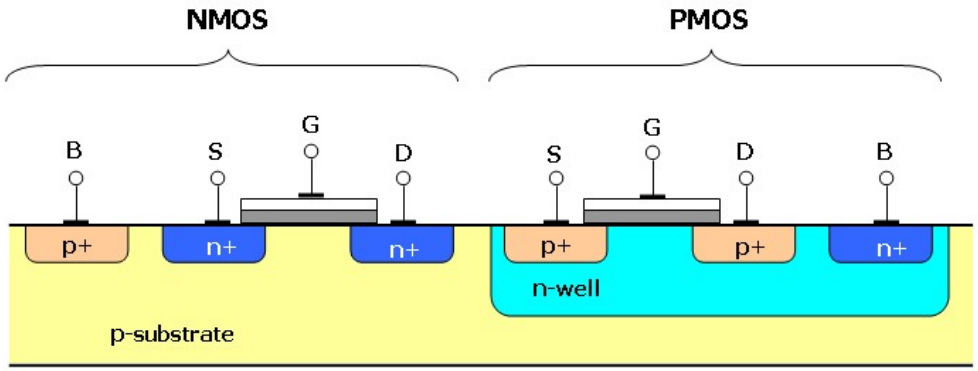


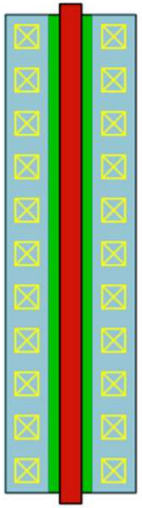
Layout File Format:
GDS II

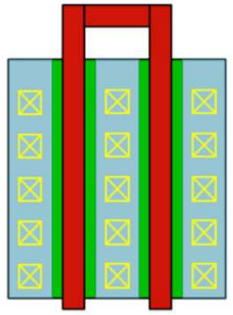
- 2D geometry drawn using polygons, paths, etc.
- Layer table required to differentiate layers
- Layer thicknesses not captured here

Ref:
https://www.eda.ncsu.edu/wiki/FreePDK15:Contents

Introduction to Layout







Less diffusion capacitance

Planar Transistors:

- 2 Parameters → Width + Length
- 3 Parameters → Width + Length + Fingers

Design parameters affect 2D geometry of the layout

Ref:
https://courses.engr.illinois.edu/ece110/sp2021/content/courseNotes/files/?MOSFETs

Introduction to Layout

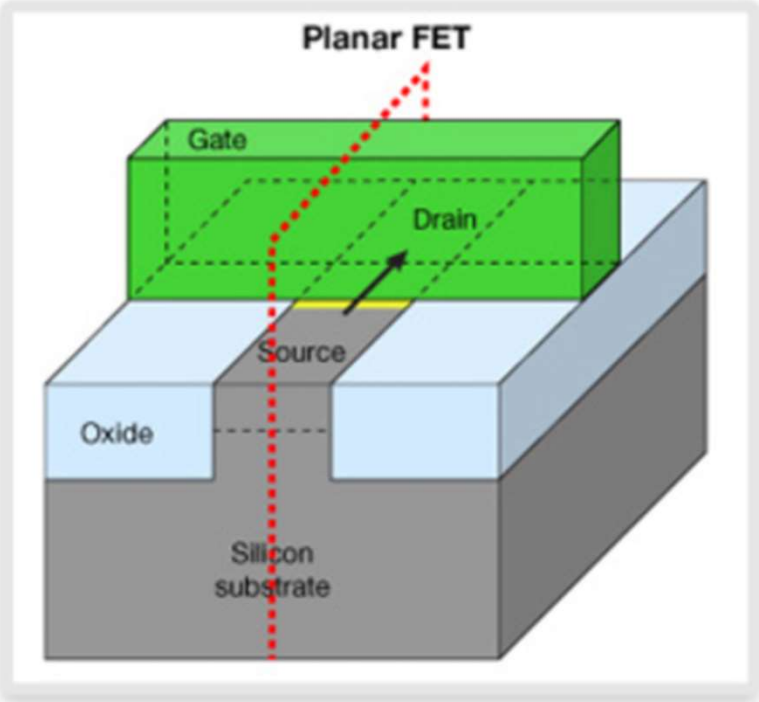


Figure 1: Planar FET

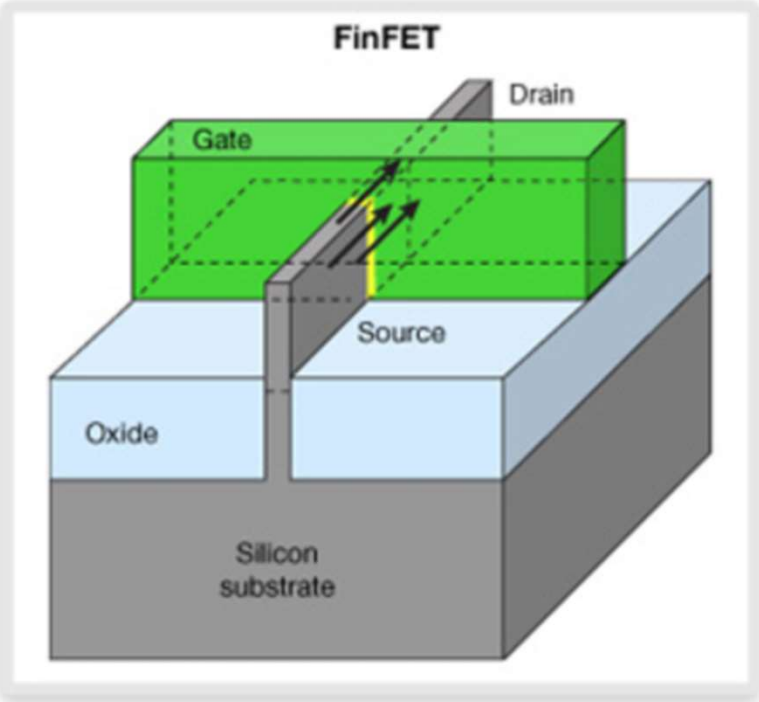


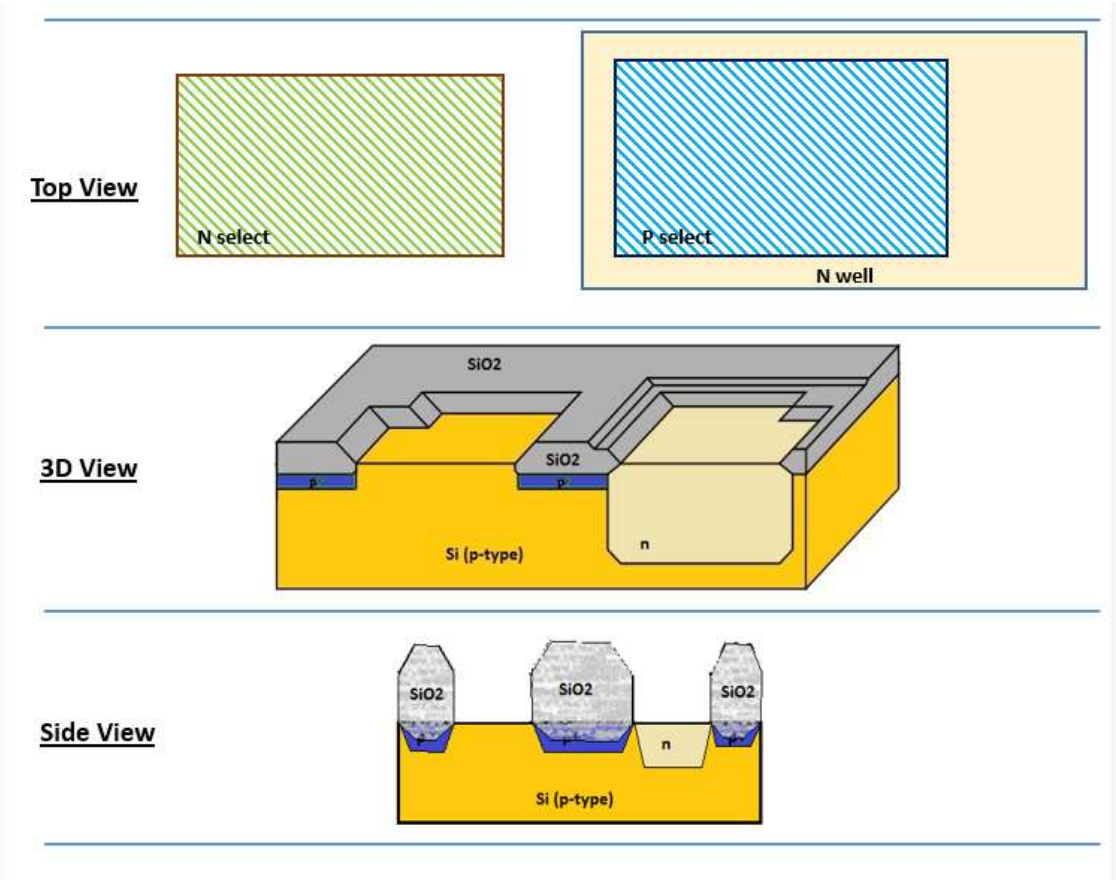
Figure 2: FinFET

More design parameters (i.e. # of fins), but still 2D layout

Ref:
https://www.synopsys.com/designware-ip/technical-bulletin/finfet-design.html

Layout Process 1

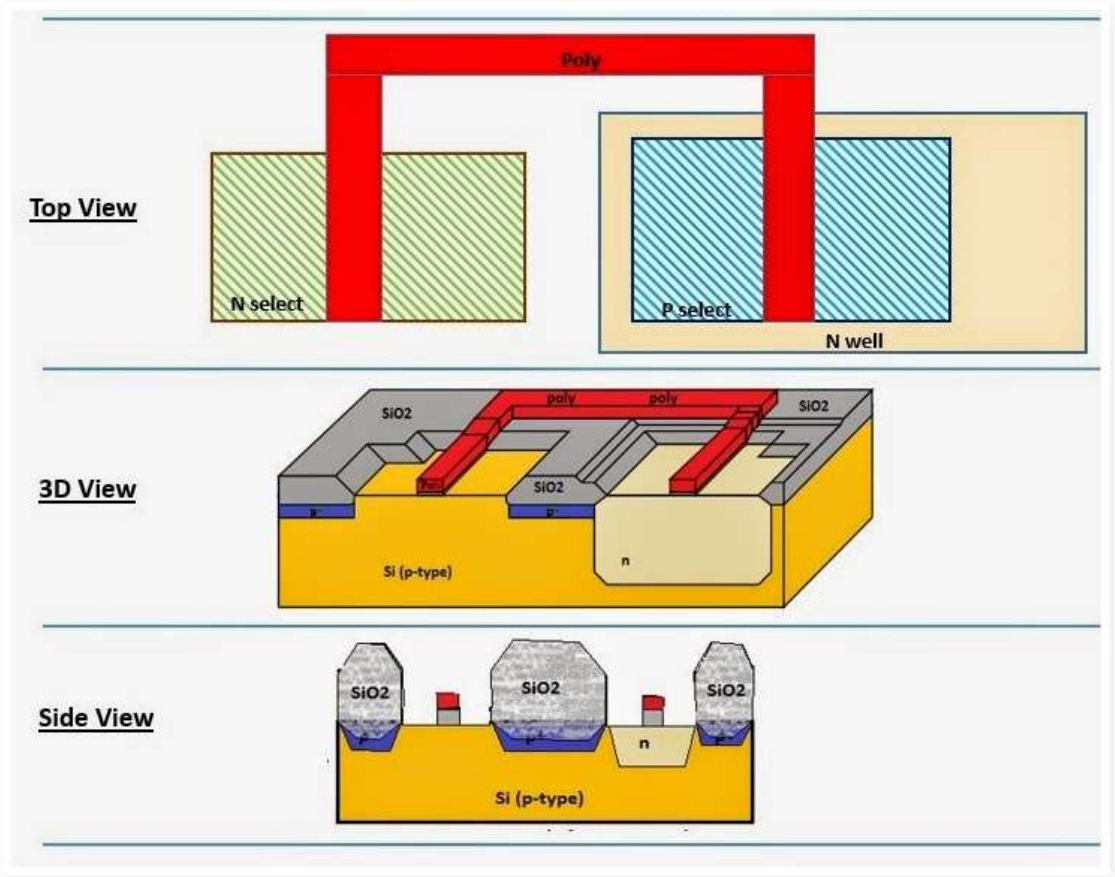
Draw NWELL and PWELL (Transistor Regions)



Ref:
http://users.ece.utexas.edu
https://inst.eecs.berkeley.edu
http://www.vlsi-expert.com

Layout Process 2

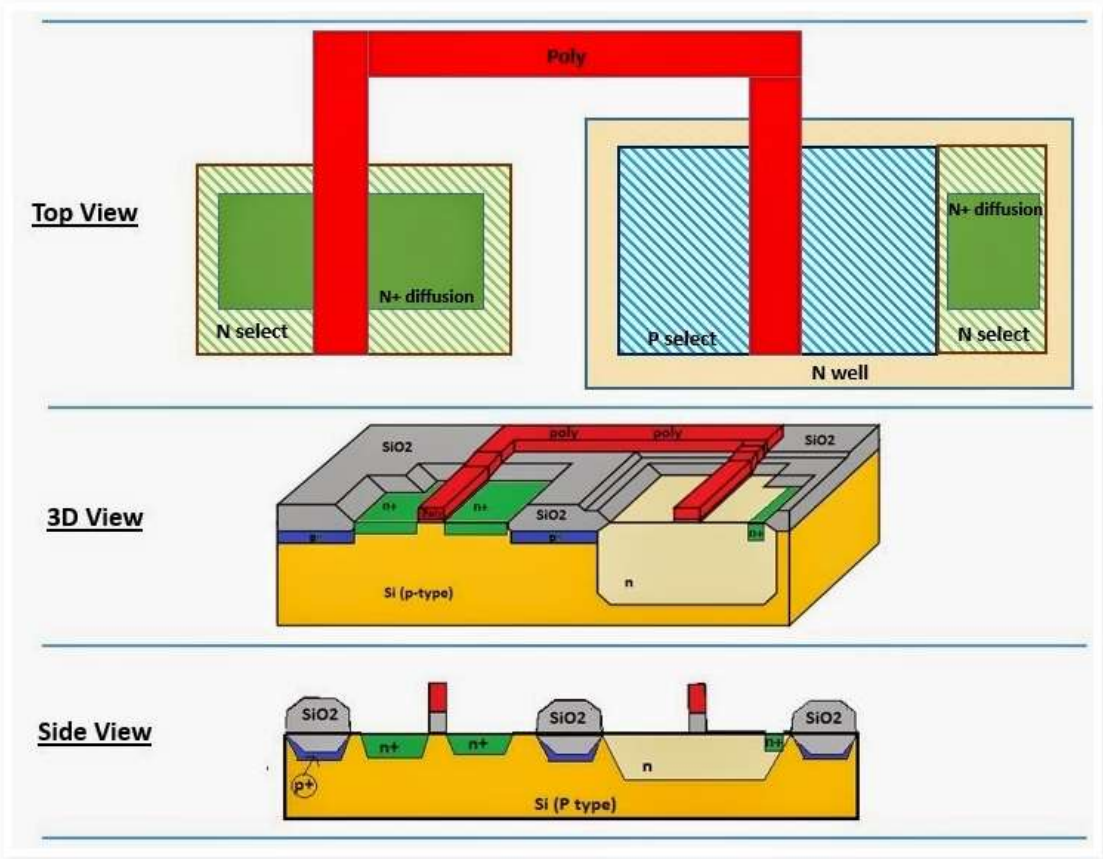
Draw POLY layer (Transistor Gates)



Ref:
http://users.ece.utexas.edu
https://inst.eecs.berkeley.edu
http://www.vlsi-expert.com

Layout Process 3

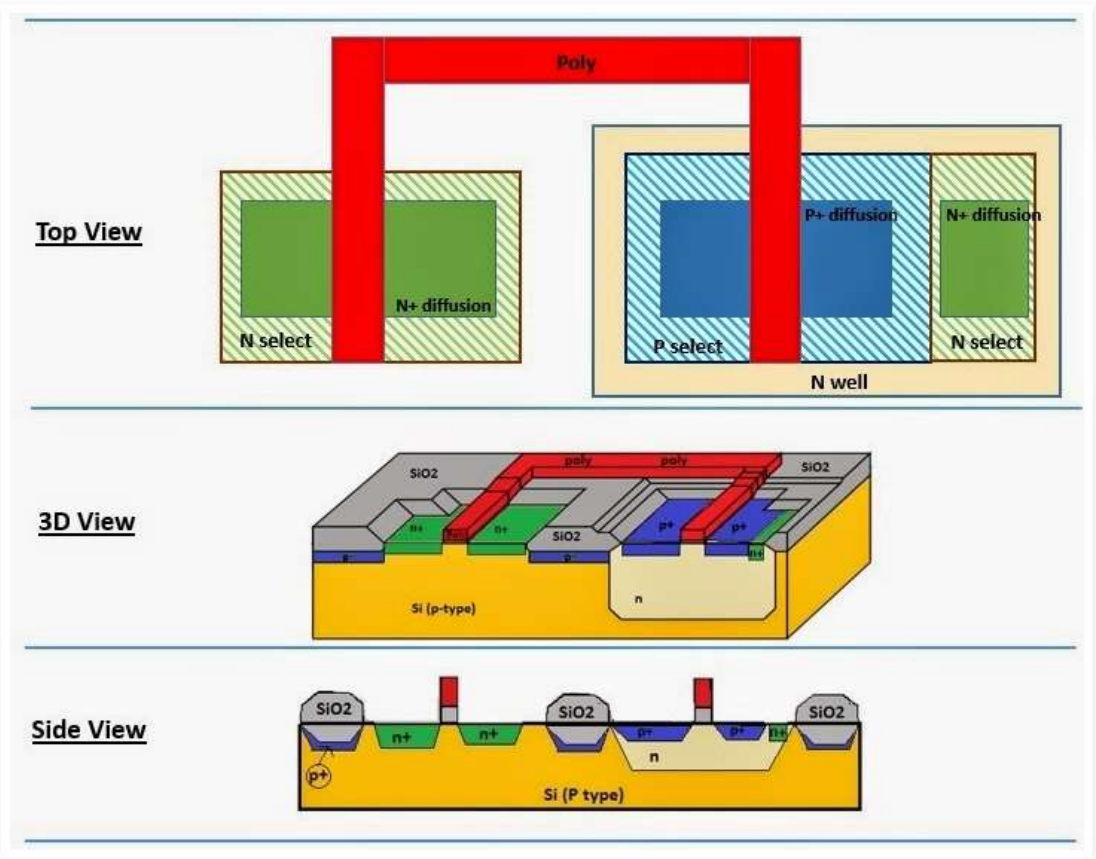
Draw N+ diffusion for NMOS drain/source.
Draw N+ diffusion for PMOS body.



Ref:
http://users.ece.utexas.edu
https://inst.eecs.berkeley.edu
http://www.vlsi-expert.com

Layout Process 4

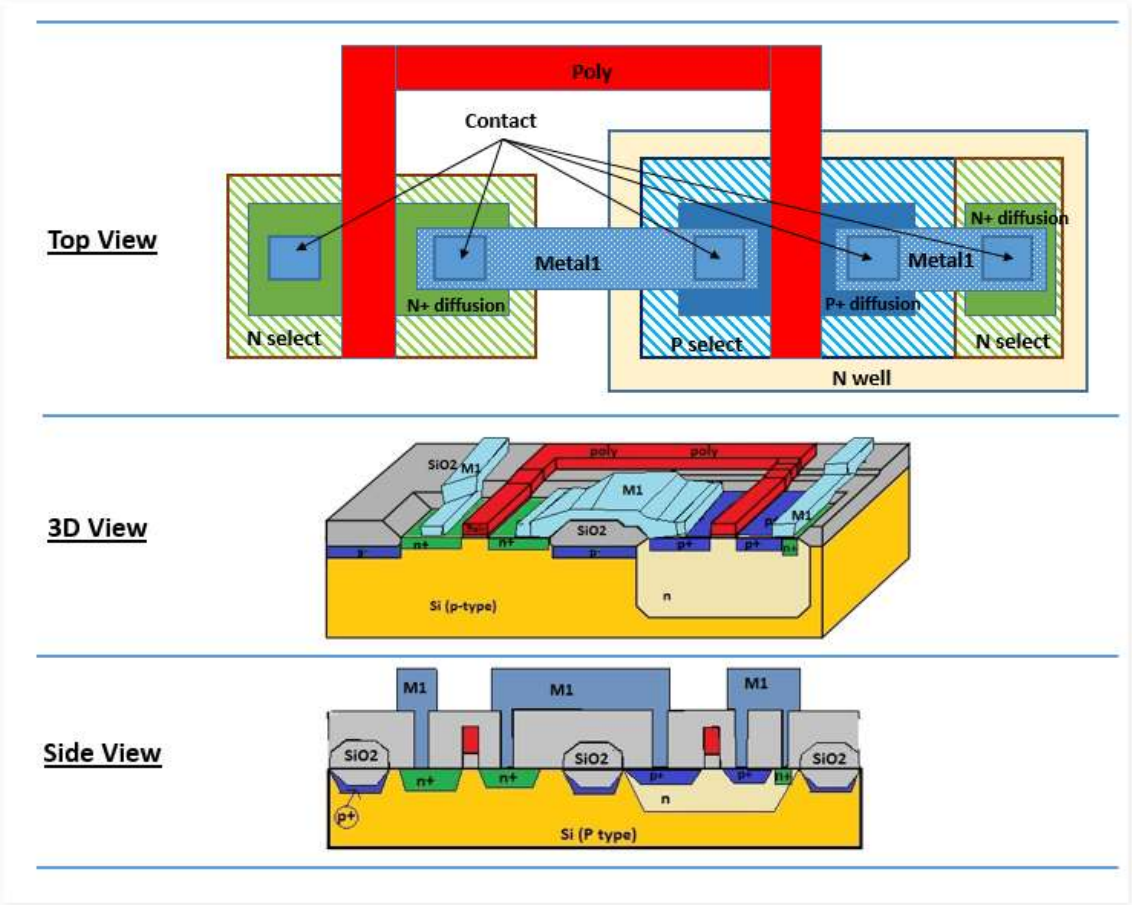
Draw P+ diffusion for PMOS drain/source.



Ref:
http://users.ece.utexas.edu
https://inst.eecs.berkeley.edu
http://www.vlsi-expert.com

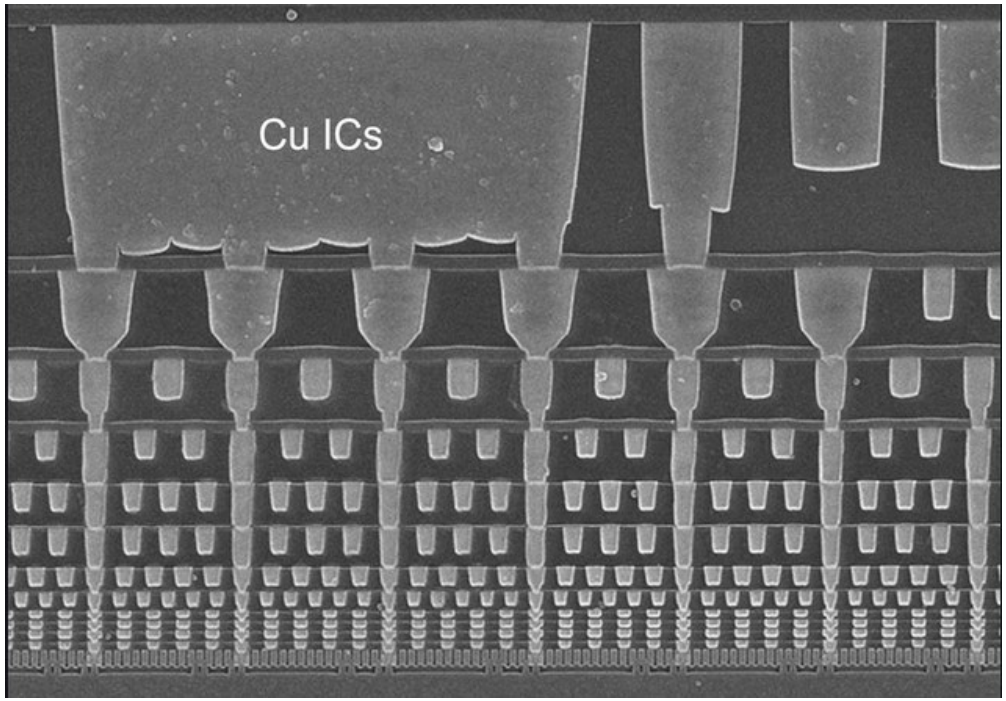
Layout Process 5

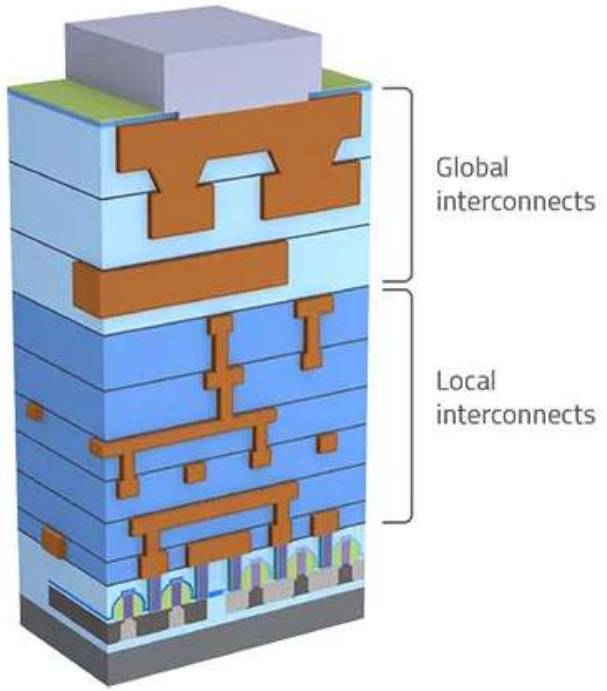
Draw metal layers.



Ref:
http://users.ece.utexas.edu
https://inst.eecs.berkeley.edu
http://www.vlsi-expert.com

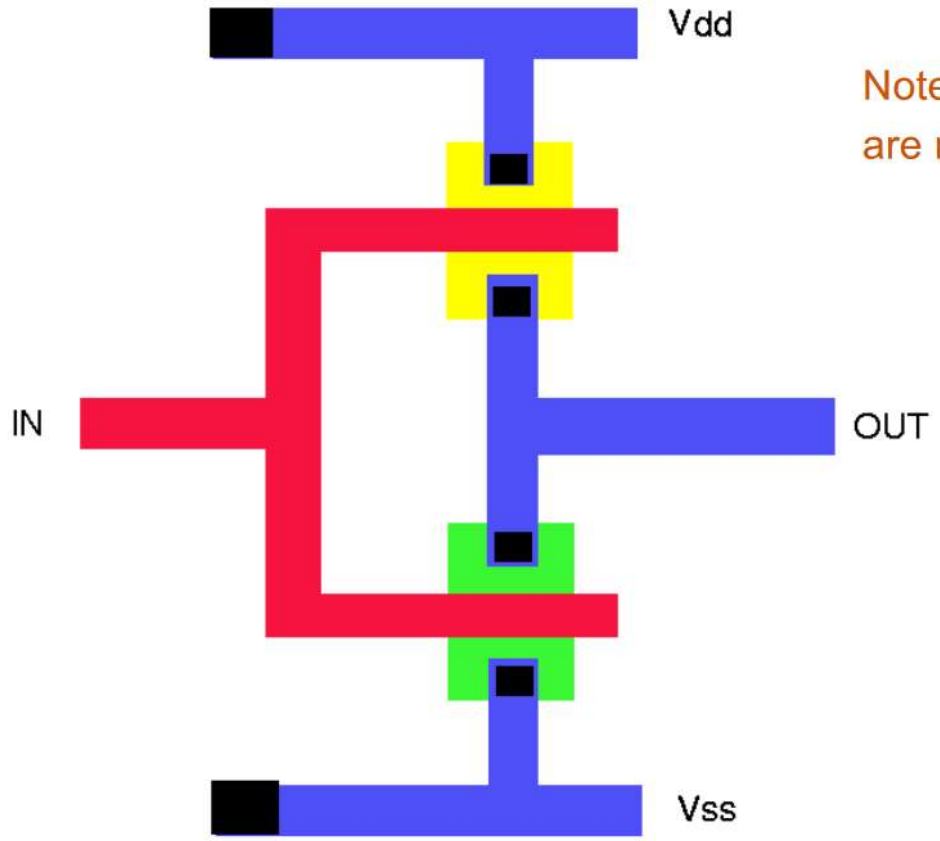
Metal Layers



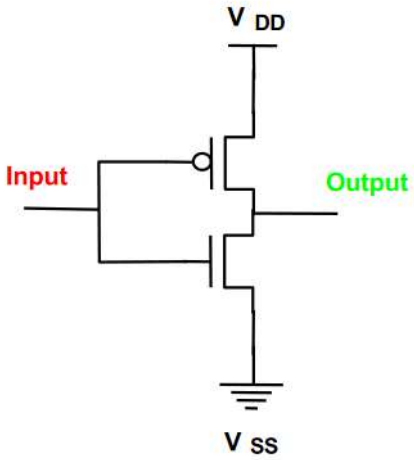


Ref:
http://users.ece.utexas.edu
https://inst.eecs.berkeley.edu
http://www.vlsi-expert.com

Inverter Layout



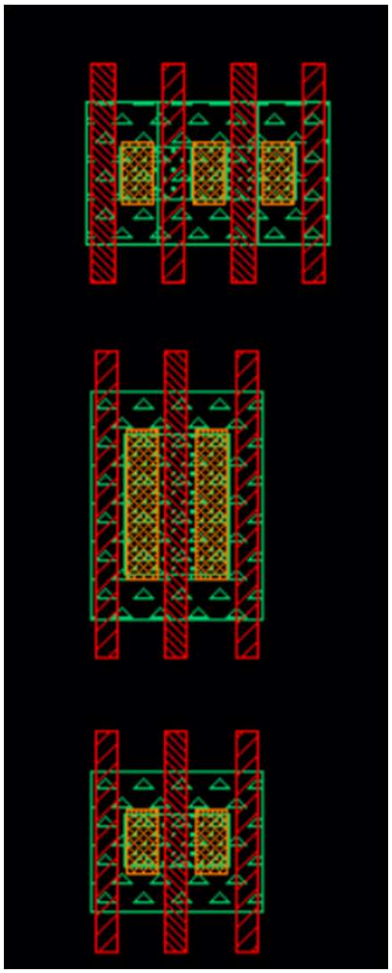
Note: the N- and P- wells are not shown here



Ref:
http://users.ece.utexas.edu
https://inst.eecs.berkeley.edu
http://www.vlsi-expert.com

NCSU 15nm FreePDK

Top View

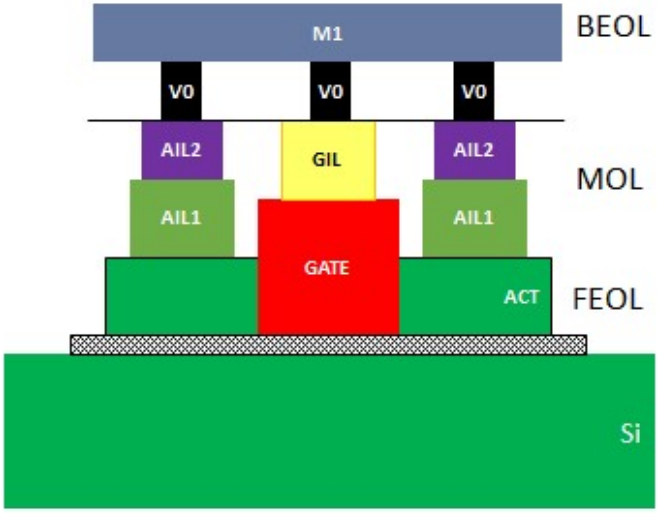


NMOS
1 Fin
2 Fingers

NMOS
2 Fins
1 Finger

NMOS
1 Fin
1 Finger

Cross Section View



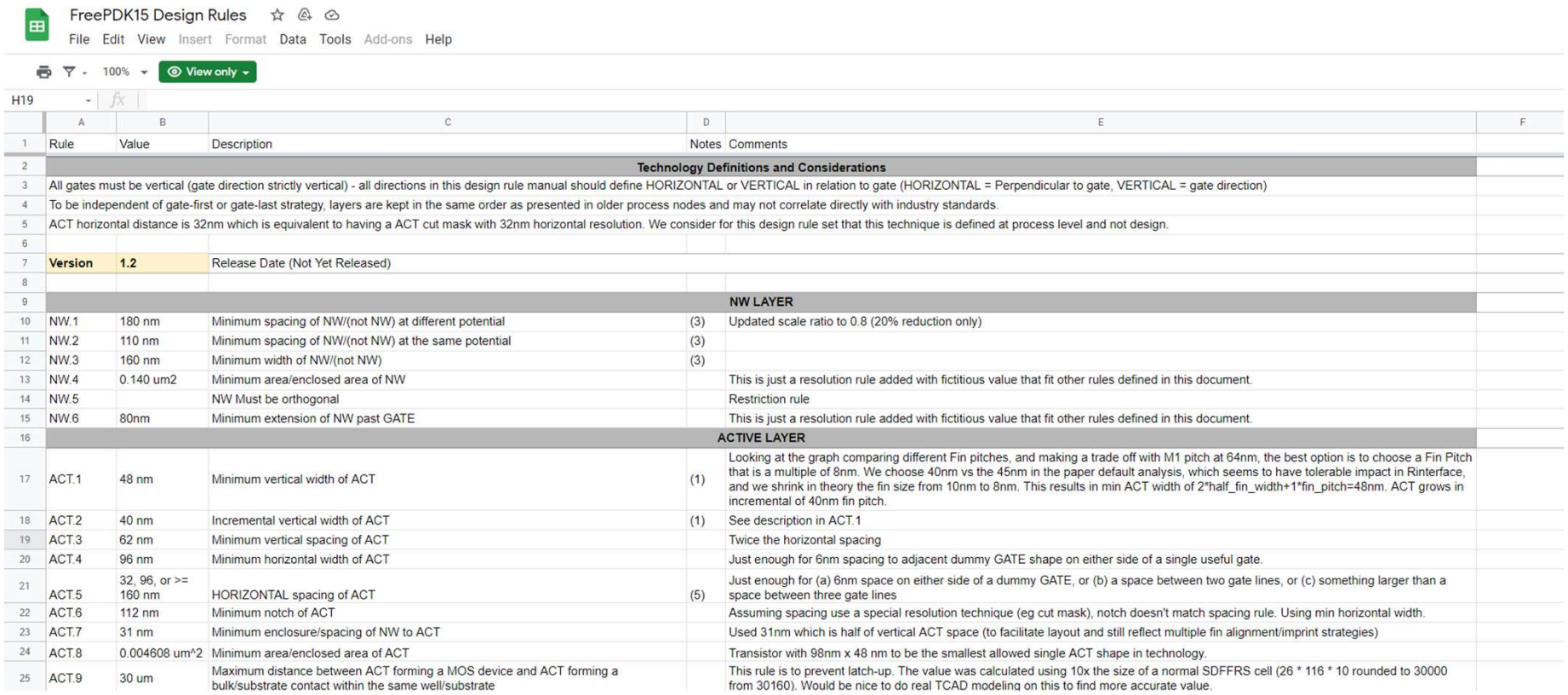
Back End of Line
(Metal Interconnects)

Middle of Line
(Connection Between Metal and Transistors)

Front End of Line
(Transistor Patterning)

Ref:
https://www.eda.ncsu.edu/wiki/FreePDK15:Contents

FreePDK15 Design Rules



FreePDK15 Design Rules

File Edit View Insert Format Data Tools Add-ons Help

100% View only

H19

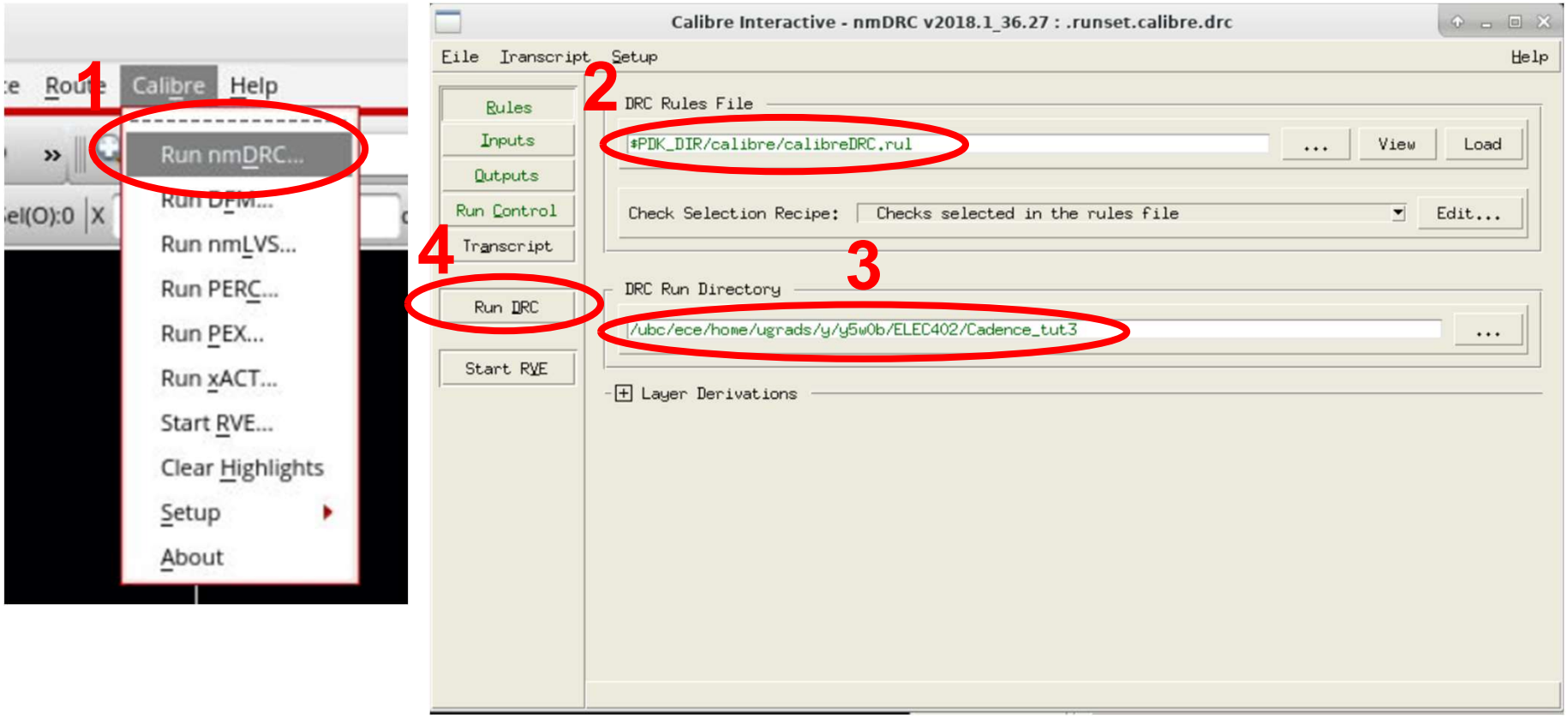
	A	B	C	D	E	F
1	Rule	Value	Description	Notes	Comments	
2			**Technology Definitions and Considerations**			
3	All gates must be vertical (gate direction strictly vertical) - all directions in this design rule manual should define HORIZONTAL or VERTICAL in relation to gate (HORIZONTAL = Perpendicular to gate, VERTICAL = gate direction)					
4	To be independent of gate-first or gate-last strategy, layers are kept in the same order as presented in older process nodes and may not correlate directly with industry standards.					
5	ACT horizontal distance is 32nm which is equivalent to having a ACT cut mask with 32nm horizontal resolution. We consider for this design rule set that this technique is defined at process level and not design.					
6						
7	**Version**	**1.2**	Release Date (Not Yet Released)			
8						
9					**NW LAYER**	
10	NW.1	180 nm	Minimum spacing of NW/(not NW) at different potential	(3)	Updated scale ratio to 0.8 (20% reduction only)	
11	NW.2	110 nm	Minimum spacing of NW/(not NW) at the same potential	(3)		
12	NW.3	160 nm	Minimum width of NW/(not NW)	(3)		
13	NW.4	0.140 um2	Minimum area/enclosed area of NW		This is just a resolution rule added with fictitious value that fit other rules defined in this document.	
14	NW.5		NW Must be orthogonal		Restriction rule	
15	NW.6	80nm	Minimum extension of NW past GATE		This is just a resolution rule added with fictitious value that fit other rules defined in this document.	
16					**ACTIVE LAYER**	
17	ACT.1	48 nm	Minimum vertical width of ACT	(1)	Looking at the graph comparing different Fin pitches, and making a trade off with M1 pitch at 64nm, the best option is to choose a Fin Pitch that is a multiple of 8nm. We choose 40nm vs the 45nm in the paper default analysis, which seems to have tolerable impact in Rinterface, and we shrink in theory the fin size from 10nm to 8nm. This results in min ACT width of 2*half_fin_width+1*fin_pitch=48nm. ACT grows in incremental of 40nm fin pitch.	
18	ACT.2	40 nm	Incremental vertical width of ACT	(1)	See description in ACT.1	
19	ACT.3	62 nm	Minimum vertical spacing of ACT		Twice the horizontal spacing	
20	ACT.4	96 nm	Minimum horizontal width of ACT		Just enough for 6nm spacing to adjacent dummy GATE shape on either side of a single useful gate.	
21	ACT.5	32, 96, or >= 160 nm	HORIZONTAL spacing of ACT	(5)	Just enough for (a) 6nm space on either side of a dummy GATE, or (b) a space between two gate lines, or (c) something larger than a space between three gate lines	
22	ACT.6	112 nm	Minimum notch of ACT		Assuming spacing use a special resolution technique (eg cut mask), notch doesn't match spacing rule. Using min horizontal width.	
23	ACT.7	31 nm	Minimum enclosure/spacing of NW to ACT		Used 31nm which is half of vertical ACT space (to facilitate layout and still reflect multiple fin alignment/imprint strategies)	
24	ACT.8	0.004608 um^2	Minimum area/enclosed area of ACT		Transistor with 98nm x 48 nm to be the smallest allowed single ACT shape in technology.	
25	ACT.9	30 um	Maximum distance between ACT forming a MOS device and ACT forming a bulk/substrate contact within the same well/substrate		This rule is to prevent latch-up. The value was calculated using 10x the size of a normal SDFFRS cell (26 * 116 * 10 rounded to 30000 from 30160). Would be nice to do real TCAD modeling on this to find more accurate value.	

Lots of design rules… Look through briefly…
Layout design and run DRC then check back regularly

Ref:
https://docs.google.com/spreadsheets/d/1oj-vyi7ofQjcoeXqKm_lEymi5vxDVPC07XwCqP48ABU/edit#gid=999622809

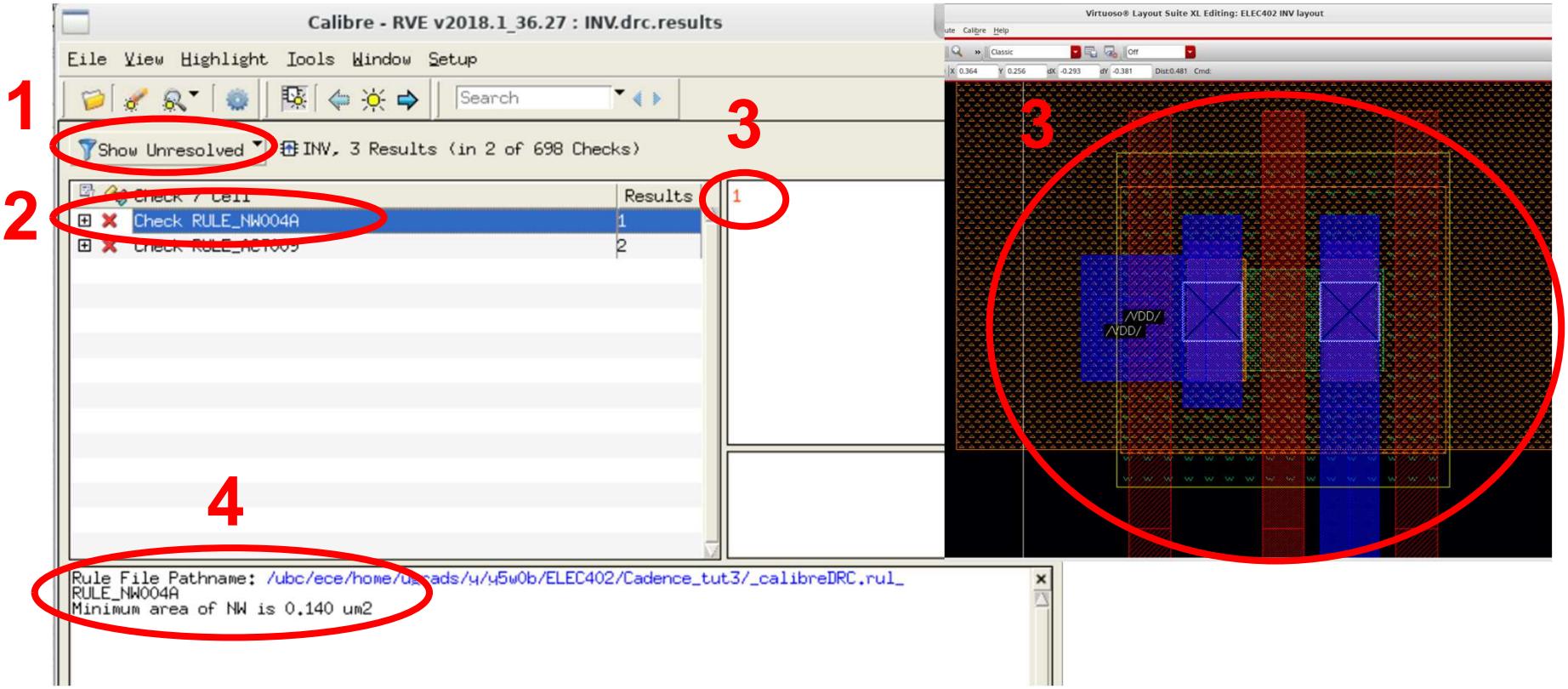
Design Rule Check (DRC)

DRC: Checks for geometries that violate manufacturing capabilities



1. Open Calibre DRC
2. Check that the design rule deck is loaded properly by clicking “View”
3. Set the run directory (use the project directory)
4. Run DRC

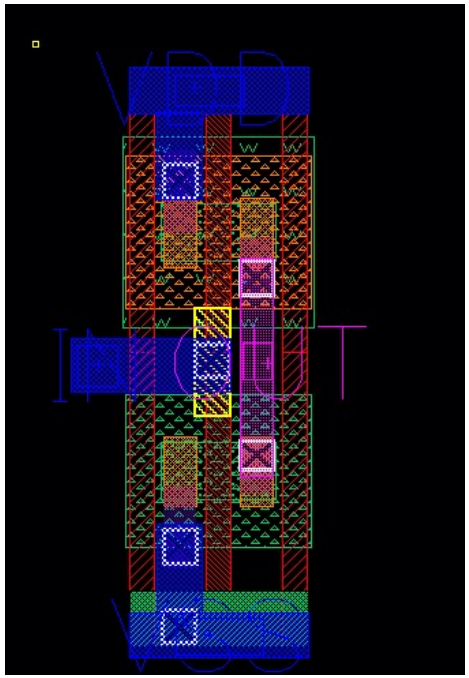
Design Rule Check (DRC)



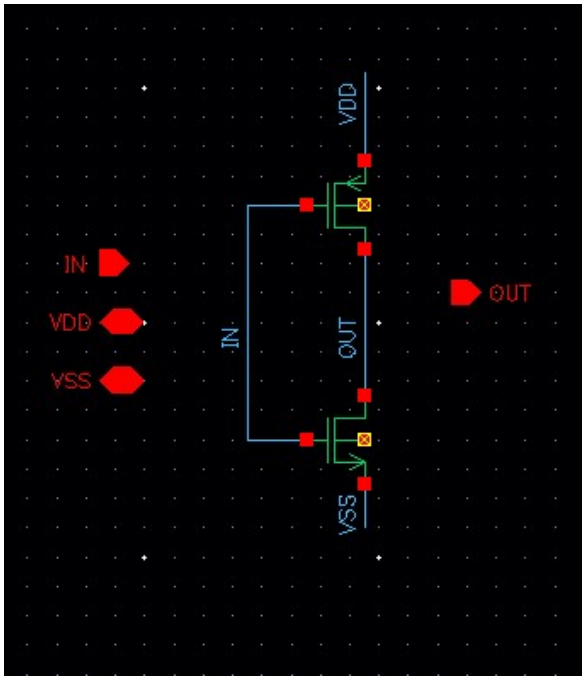
1. Set the filter to “Show Unresolved” to show unresolved DRC issues
2. Select the rule to view and expand to look at cells that have DRC issues
3. Double click the number to zoom into the DRC issue (shown in yellow highlight)
4. Description of the DRC issue

Layout vs Schematic (LVS)

LVS: Checks for connectivity differences between layout and schematic.

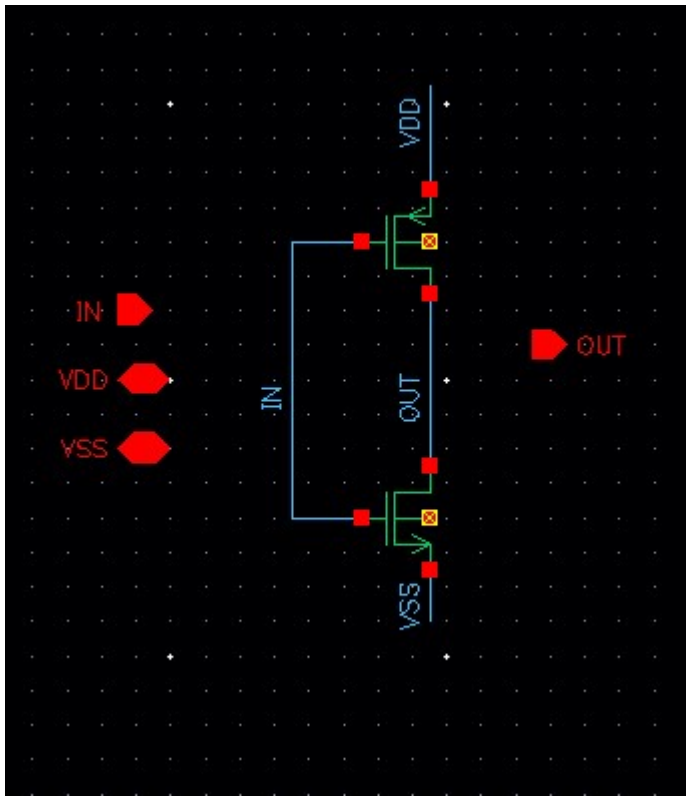


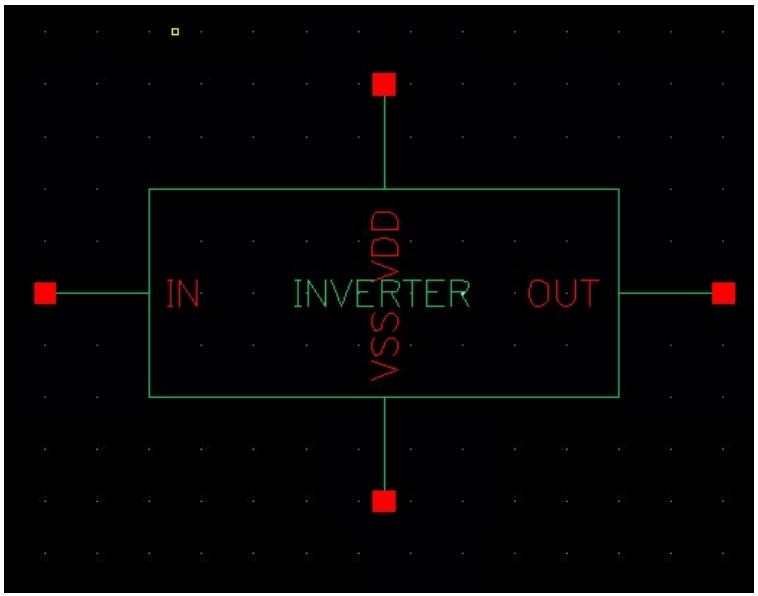
=



Layout vs Schematic (LVS)

Create schematic view and create a symbol.
Do not connect bulks, leave them open.

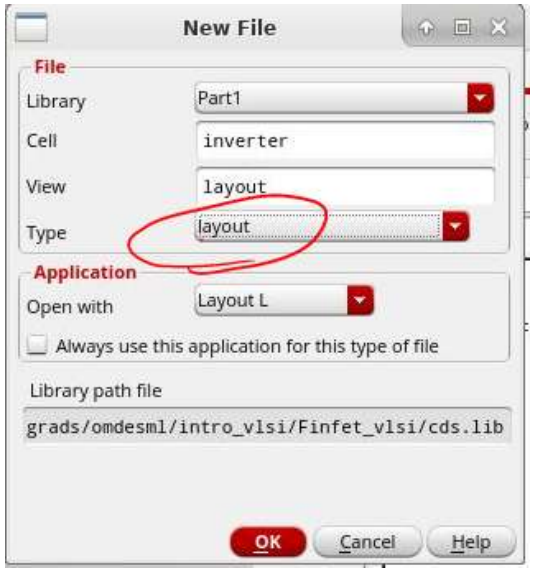


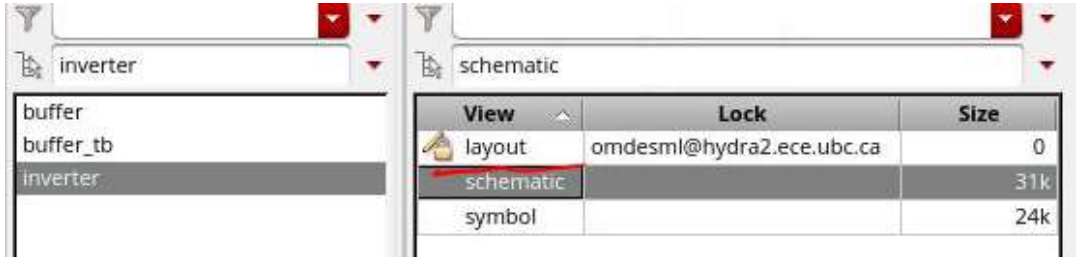


Layout vs Schematic (LVS)

For the same cell, create layout view:

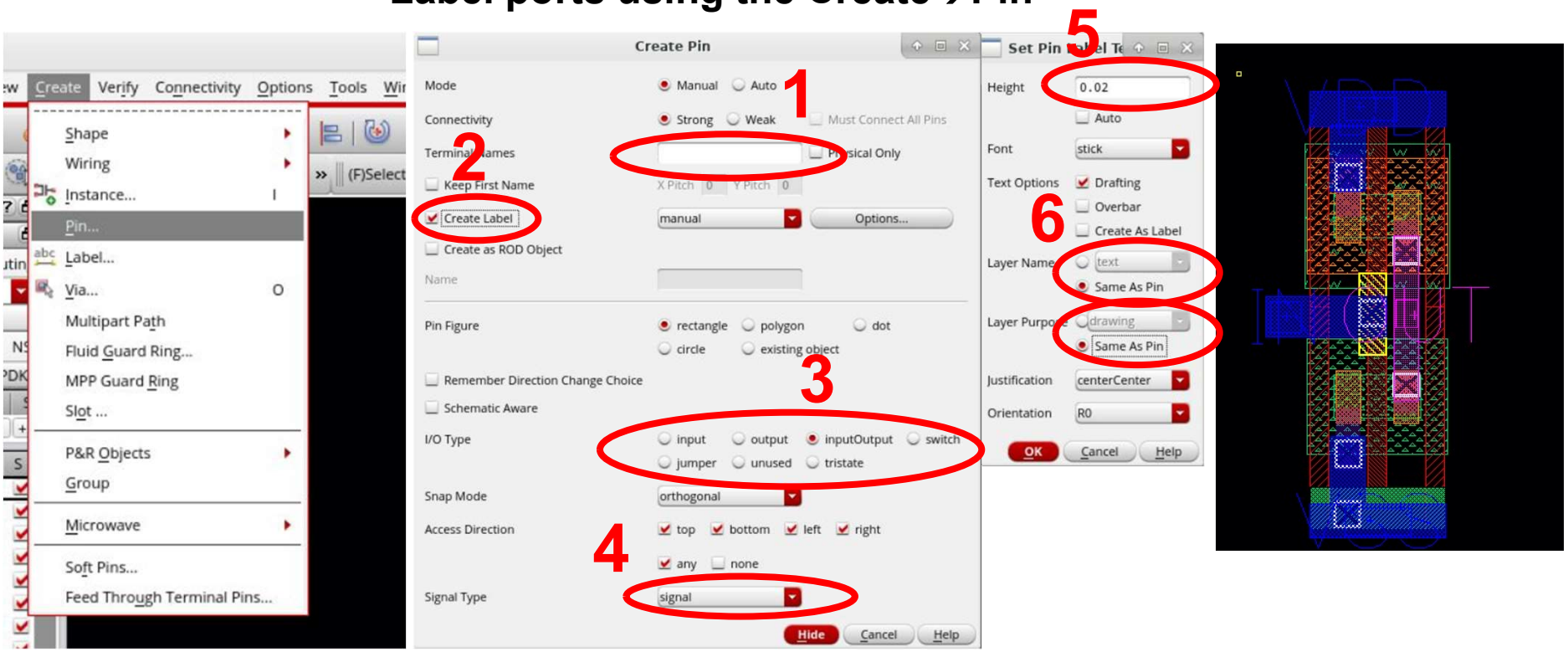
- **In “Library Manager”, File→New→Cell View**
- **Choose “Type” as “layout” then click “OK”**





Layout vs Schematic (LVS)

Layout design
- Label ports using the Create→Pin



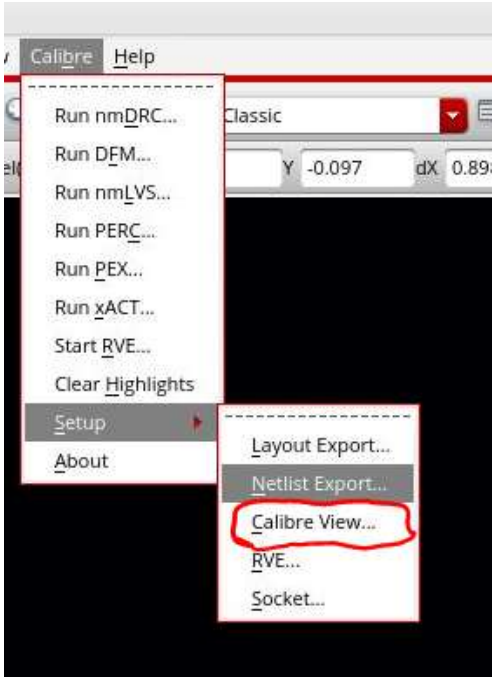
1. Set terminal name (match with schematic terminals)
2. Check “Create Label” (creates text label on layout)
3. Check I/O Type (match with I/O type in schematic)
4. Select Signal Type (match with signal type in schematic)
5. Set text height
6. Check “Same as Pin” for both “Layer Name” and “Layer Purpose” (ensures text is on same layer as pin)

Layout vs Schematic (LVS)

LVS extracts netlist from layout connections and schematic connections. Then, it compares them to ensure layout follows schematic.

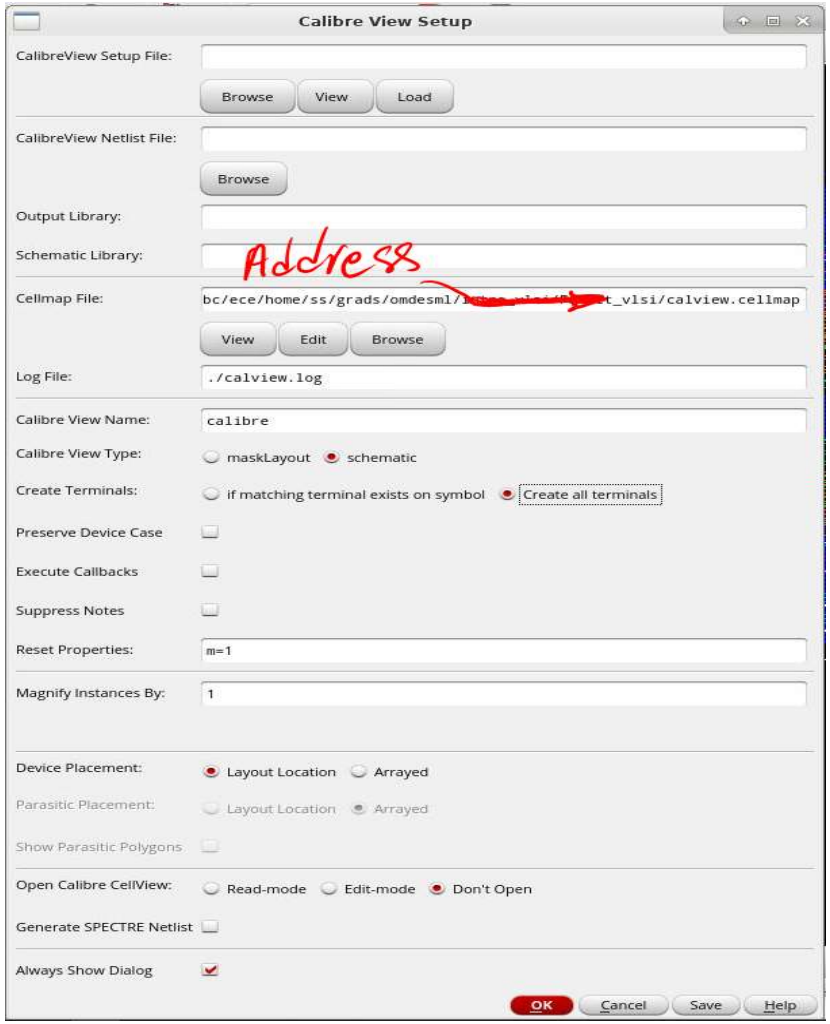
Need to load Calibre with library file called “calview.cellmap”

To do so, go to Calibre→Setup→Calibre View



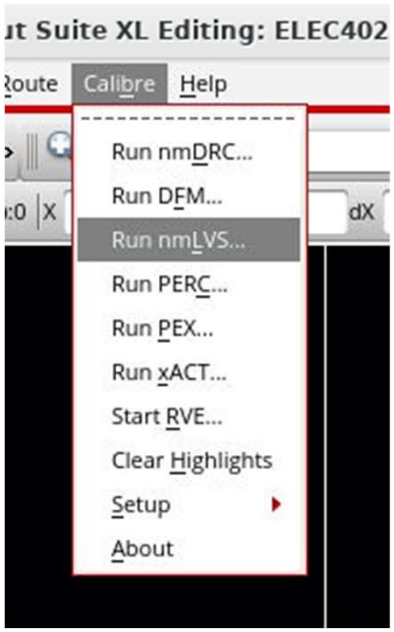
Layout vs Schematic (LVS)

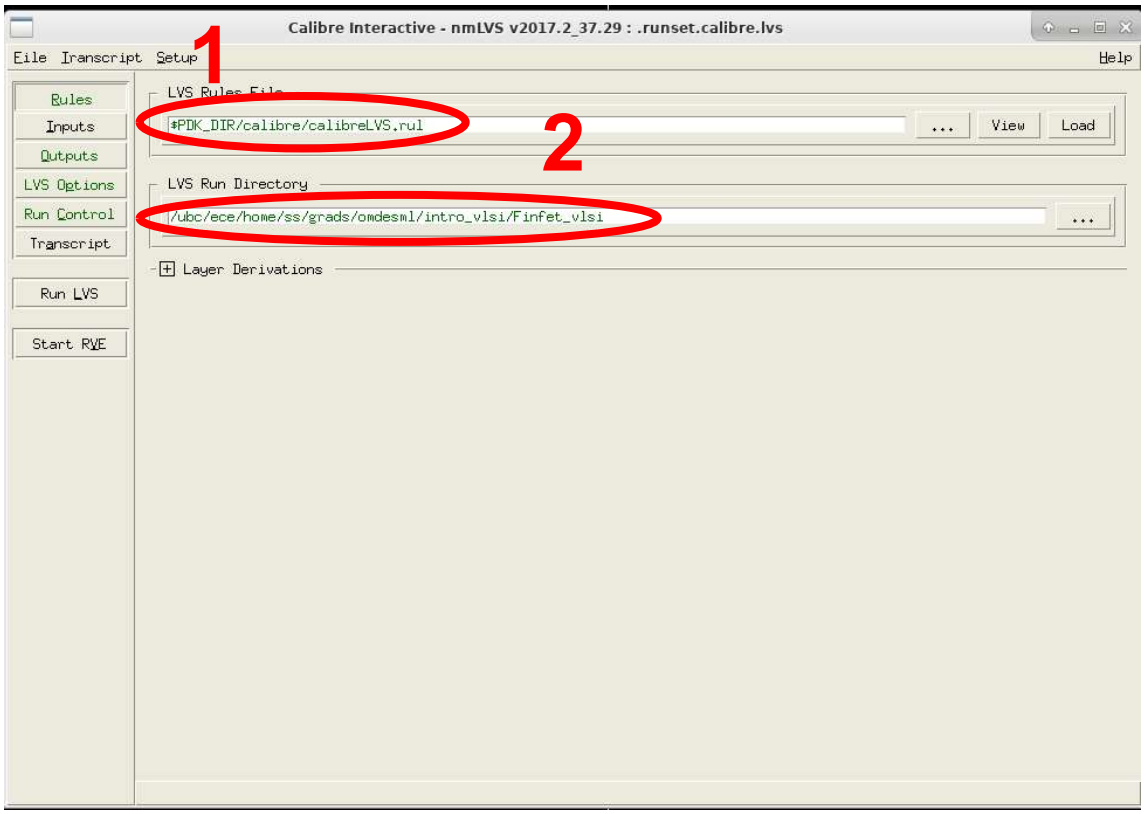
Set similar to the following and press "Save"



Layout vs Schematic (LVS)

Go to Calibre→Run nmLVS

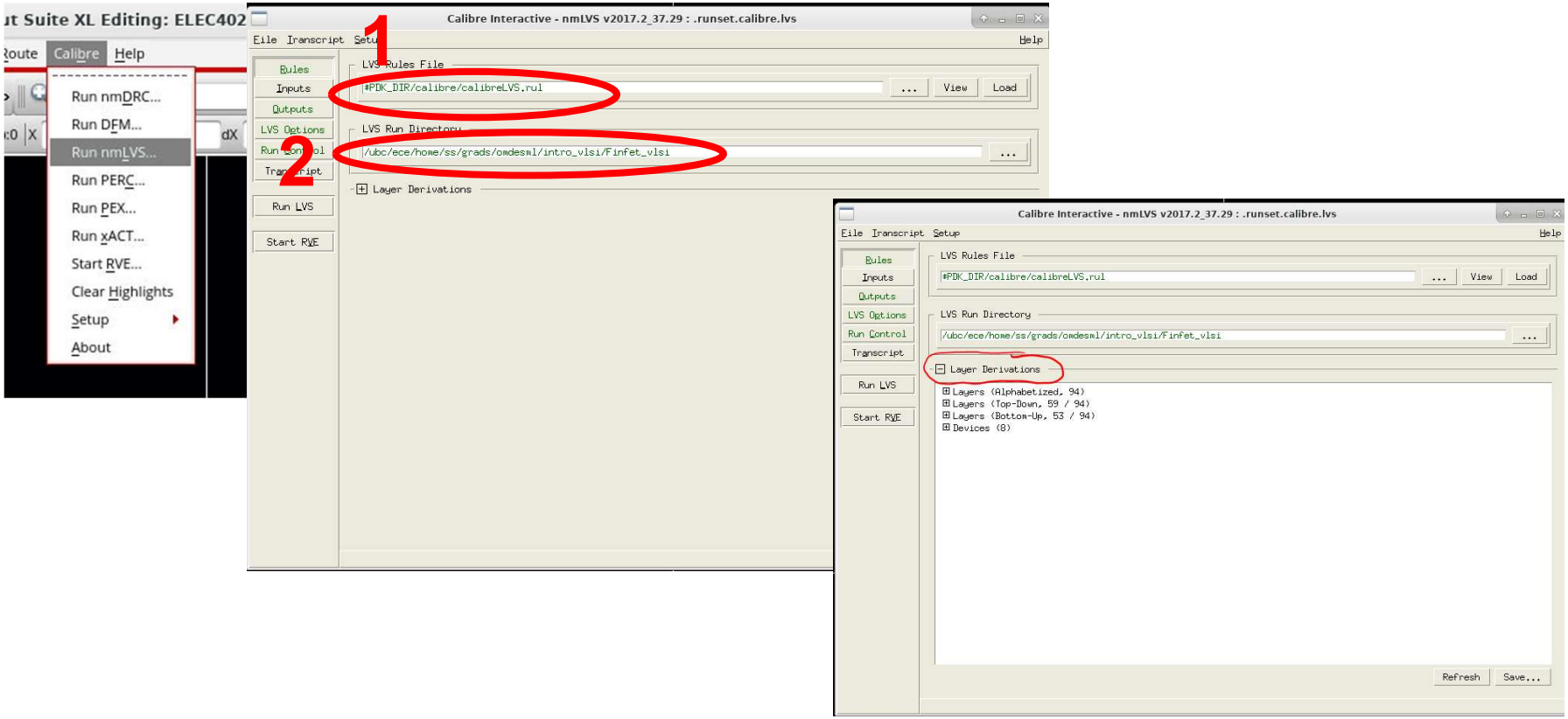




1. Check that the LVS rule file is loaded properly (/CMC/kits/ncsu_pdk/FreePDK15/calibre/calibreLVS.rul)
2. Set run directory to your working directory

Layout vs Schematic (LVS)

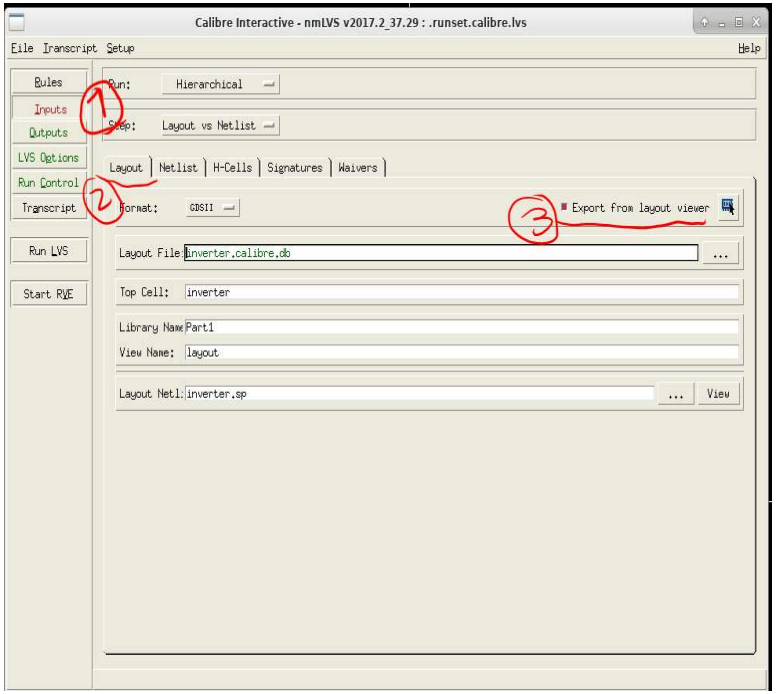
Go to Calibre→Run nmLVS

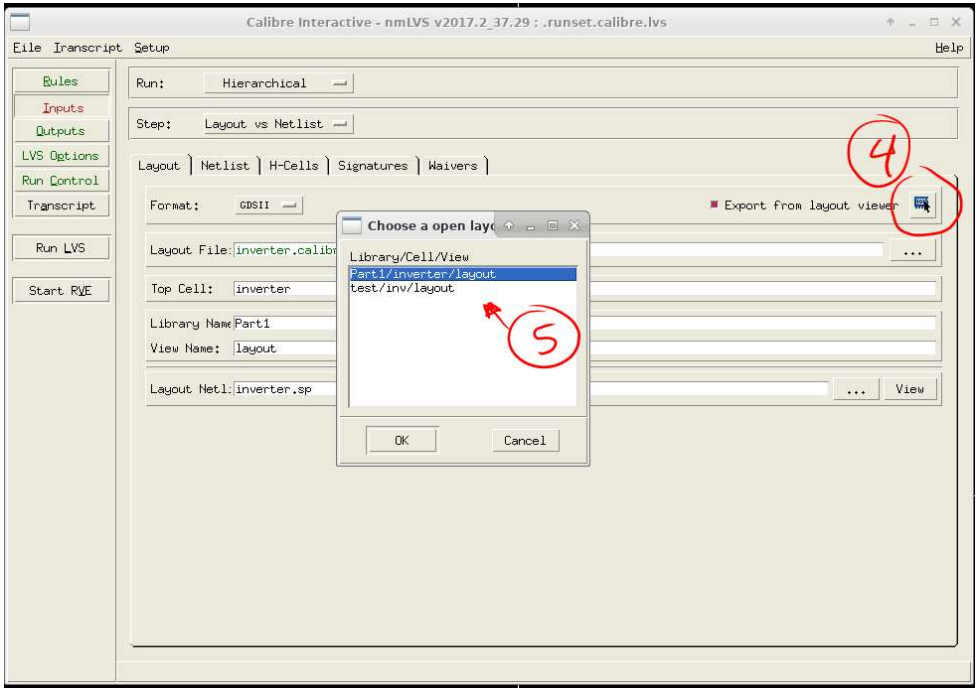


1. Check that the LVS rule file is loaded properly (/CMC/kits/ncsu_pdk/FreePDK15/calibre/calibreLVS.rul). Click on + Layer Derivations to expand; if it loads properly, you should see layers.
2. Set run directory to your working directory

Layout vs Schematic (LVS)

Set Inputs to export layout and schematic files.
Make sure schematic and layout are open.
If multiple windows are open, choose the correct one.

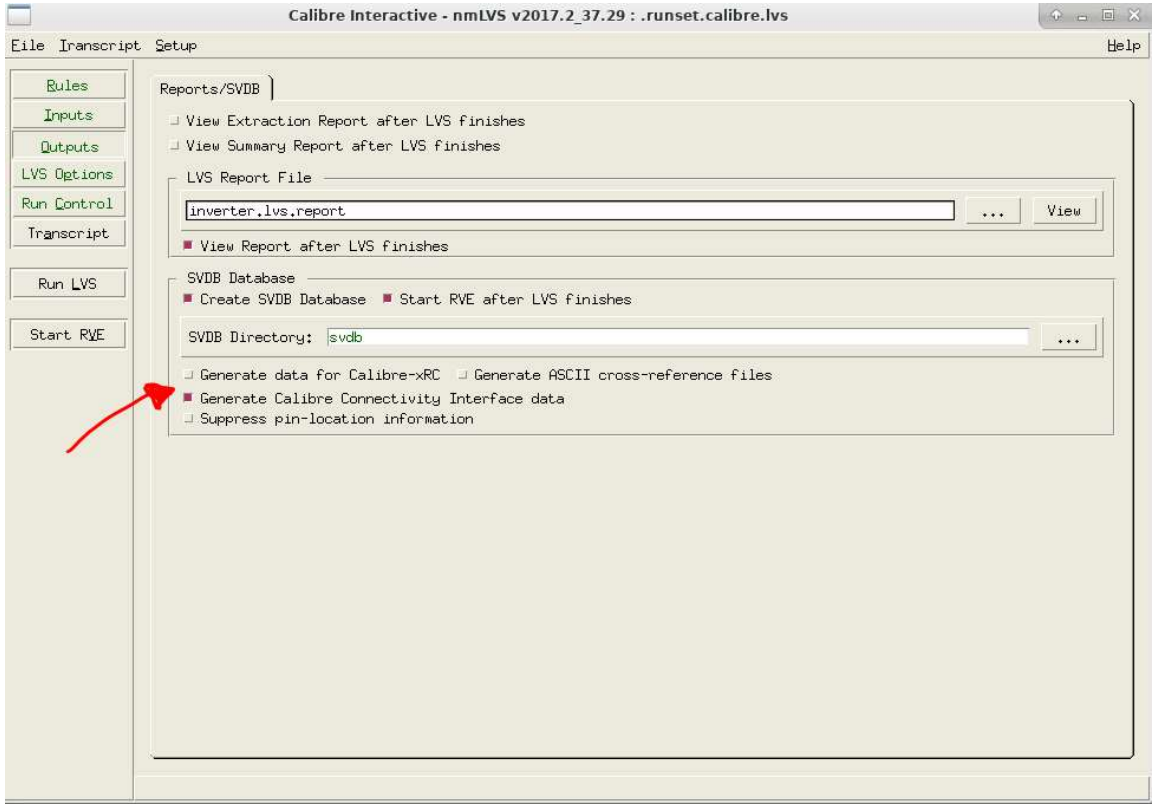




Input Tab→Layout→Export from layout viewer
Input Tab→Netlist→Export from schematic viewer

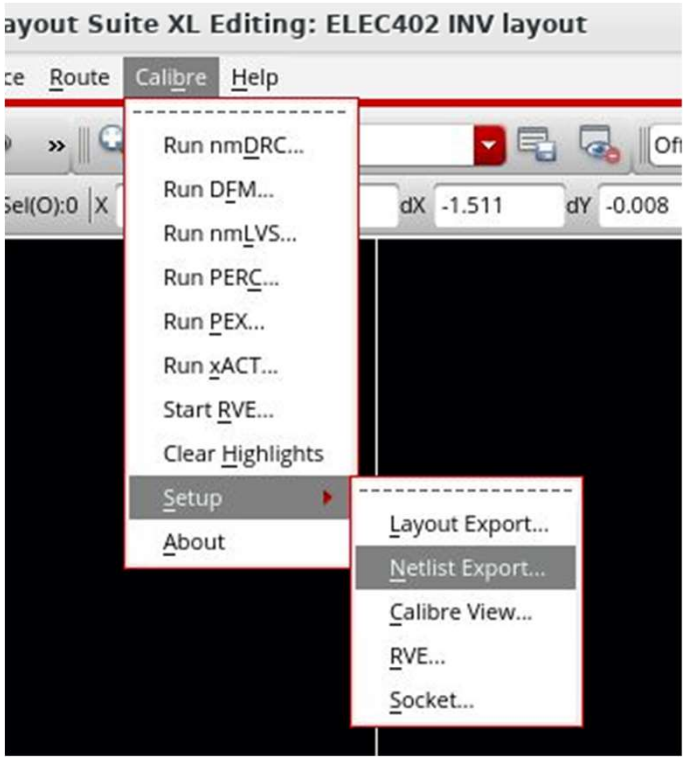
Layout vs Schematic (LVS)

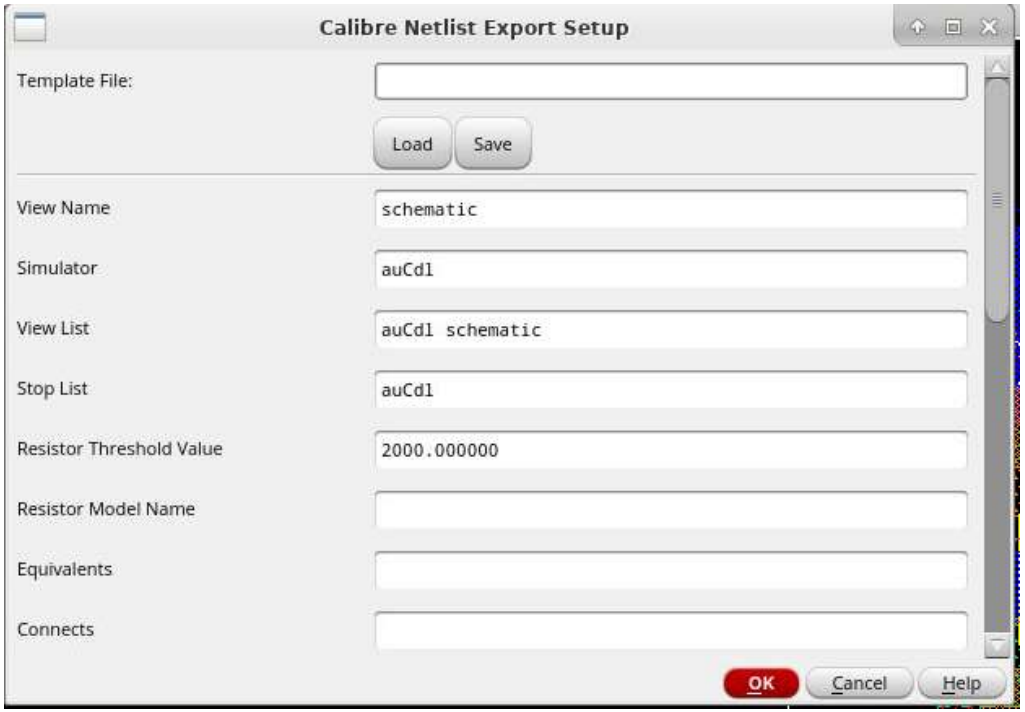
Go to Outputs and set “Generate Calibre Connectivity Interface Data”



Layout vs Schematic (LVS)

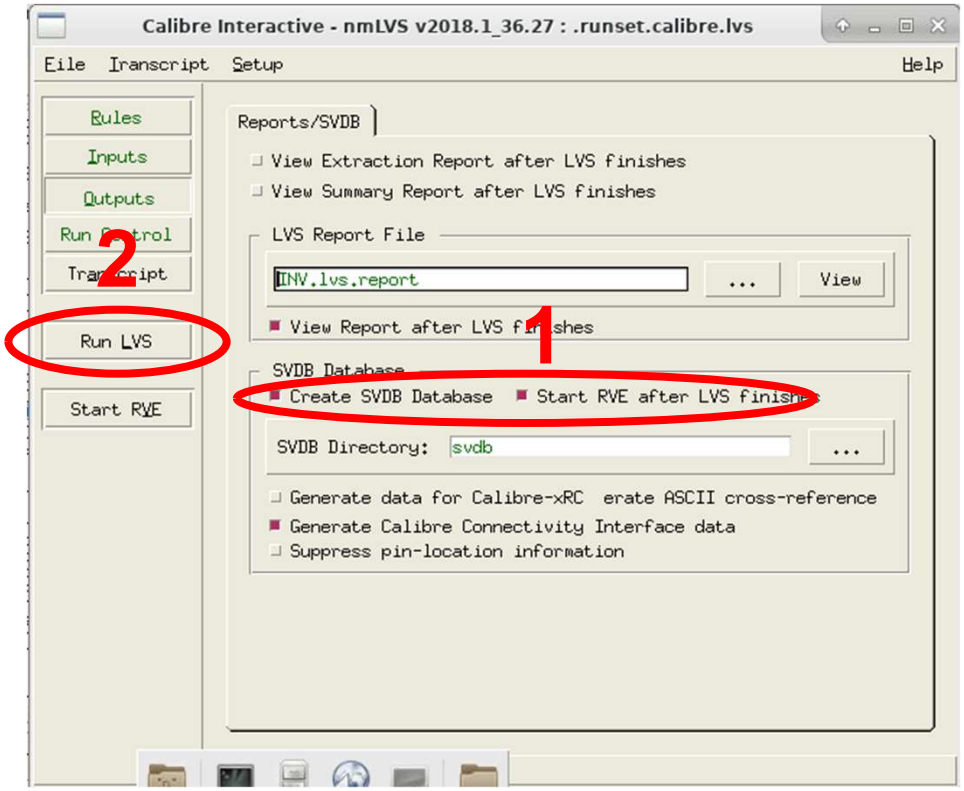
In Layout window, go to Calibre→Setup→Netlist export.
Change all "cdl" to "auCdl".





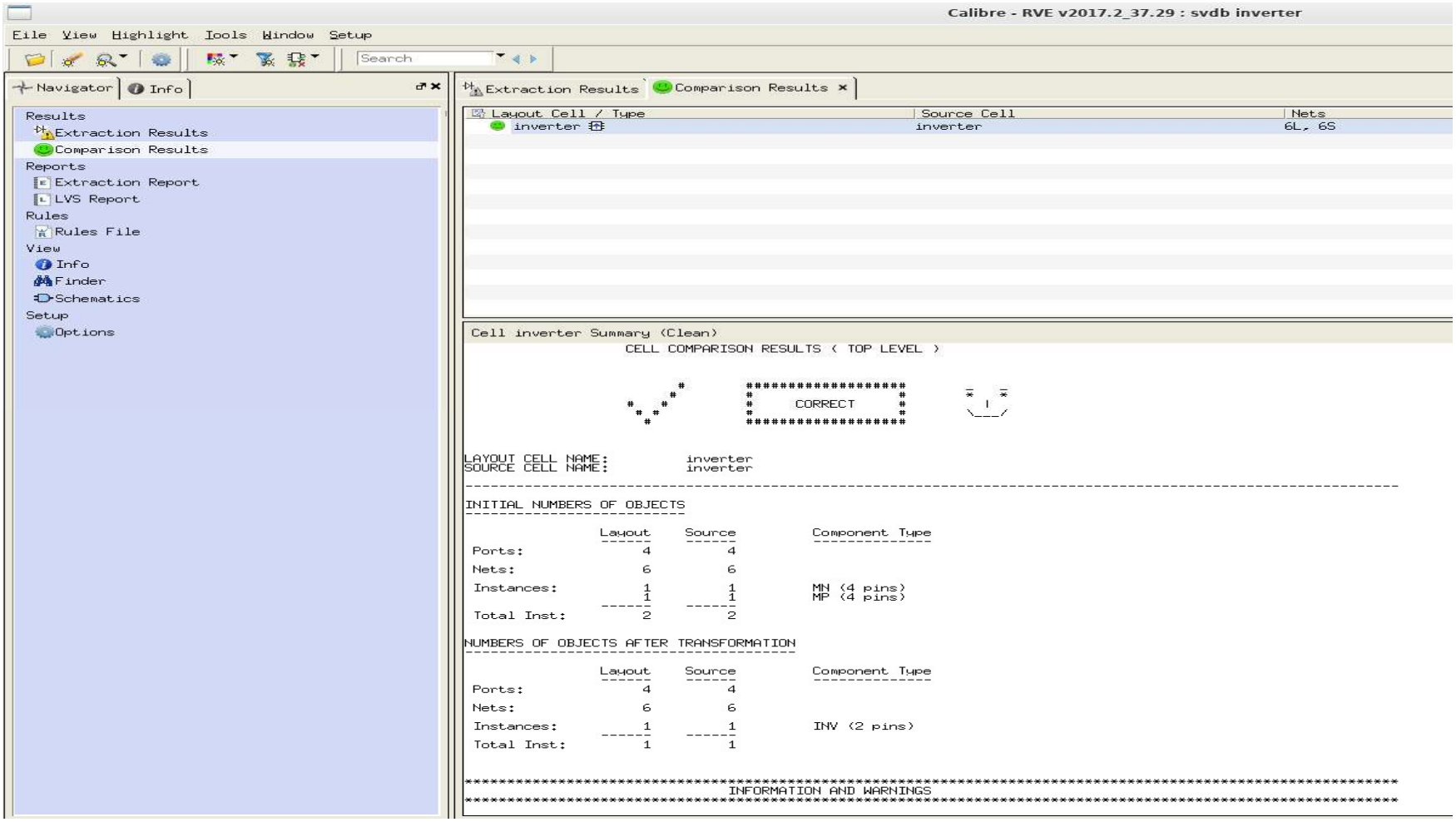
Layout vs Schematic (LVS)

Go back to LVS interactive menu and hit “Run LVS”



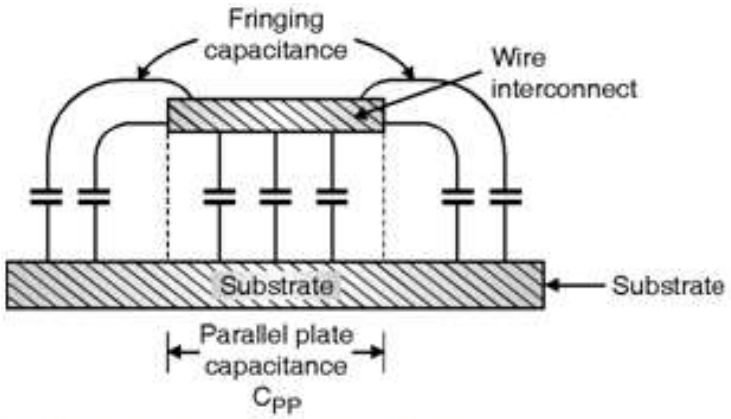
1. Ensure “Create SVDB Database” and “Start RVE after LVS finishes” are checked
2. Run LVS

Layout vs Schematic (LVS)



Parasitic Extraction (PEX)

Extracting R, C, and sometimes L parasitics from layout.



Fringing field on interconnect

Set up 15nm PDK
>> source /CMC/scripts/kit.ncsu_pdk15.csh

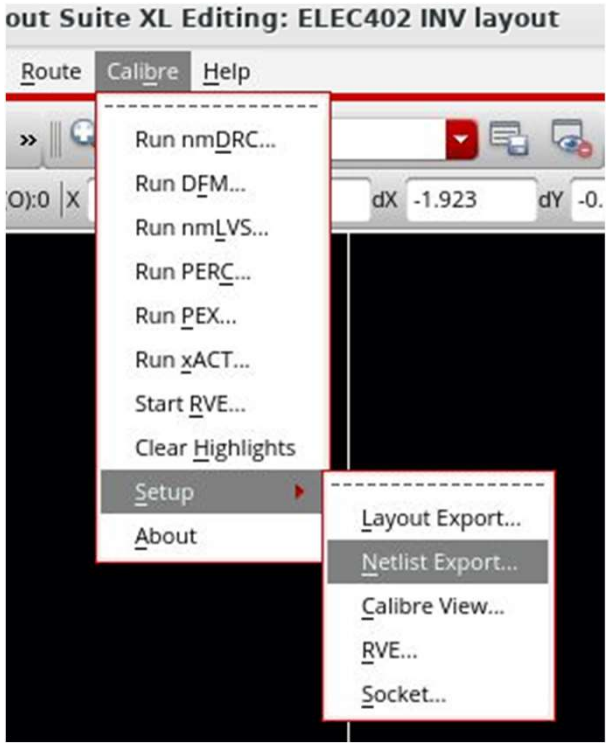
Set up Spectre simulation engine
>> source /CMC/scripts/cadence.spectre18.10.235.csh

Open Virtuoso
>> virtuoso &

Ref:
https://www.electronics-tutorial.net/Analog-CMOS-Design/MOSFET-Parasitics/Interconnect-Capacitance/

Parasitic Extraction (PEX)

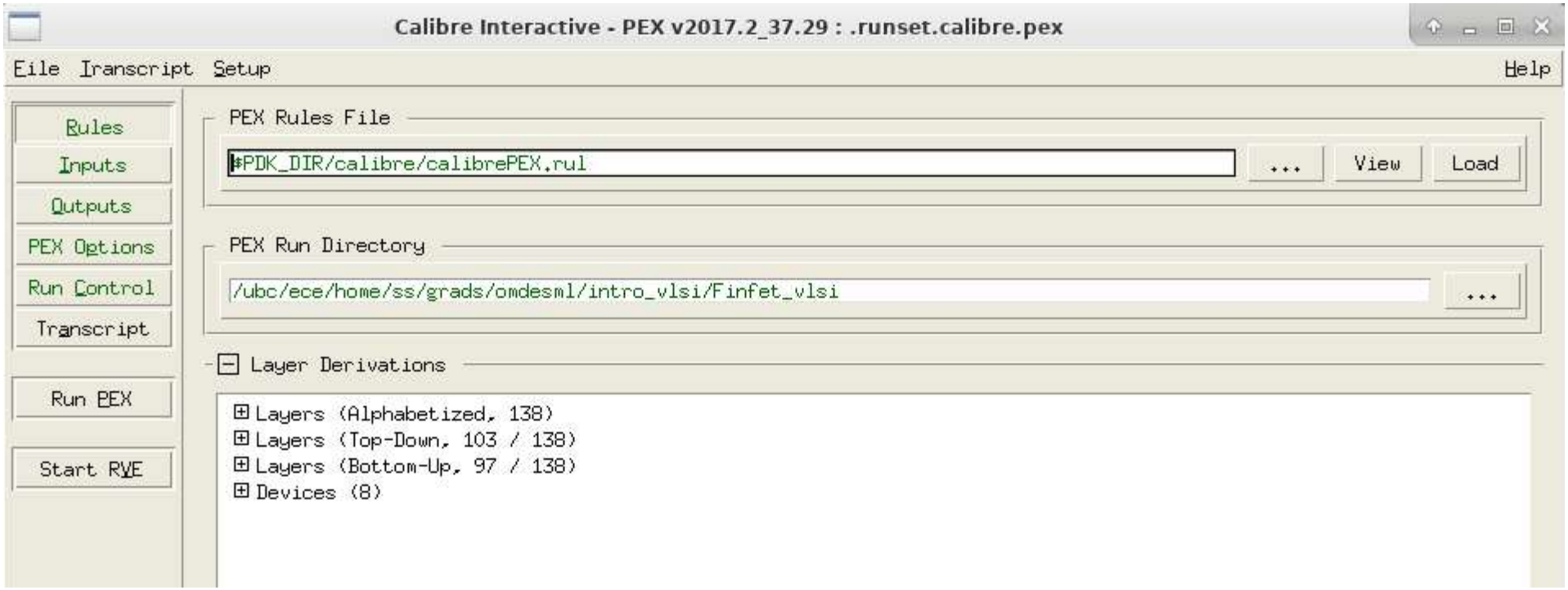
Go through LVS tutorial steps.
Ensure “Calibre View” and “Netlist Export” setup is same as in LVS tutorial.
Launch PEX by clicking under Calibre→Run PEX



Parasitic Extraction (PEX)

Select run directory (preferably the cell directory since the parasitics are associated with this cell).

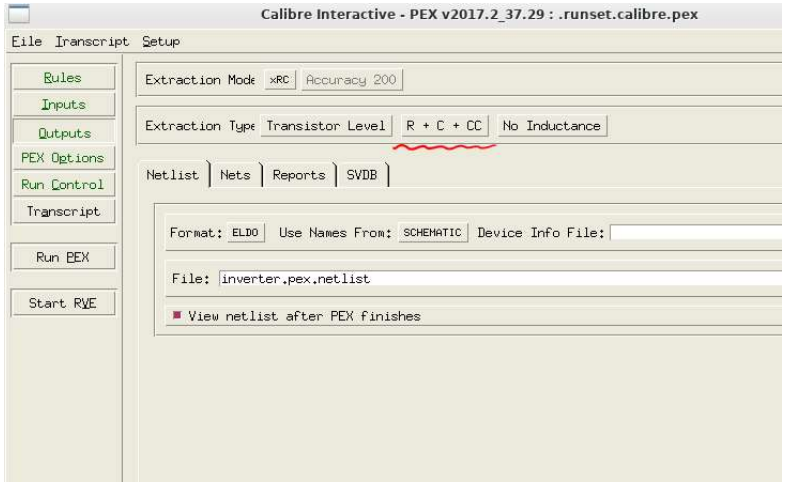
Browse PEX rule file:
/CMC/kits/ncsu_pdk/FreePDK15/ /calibre/calibrePEX.rul



Parasitic Extraction (PEX)

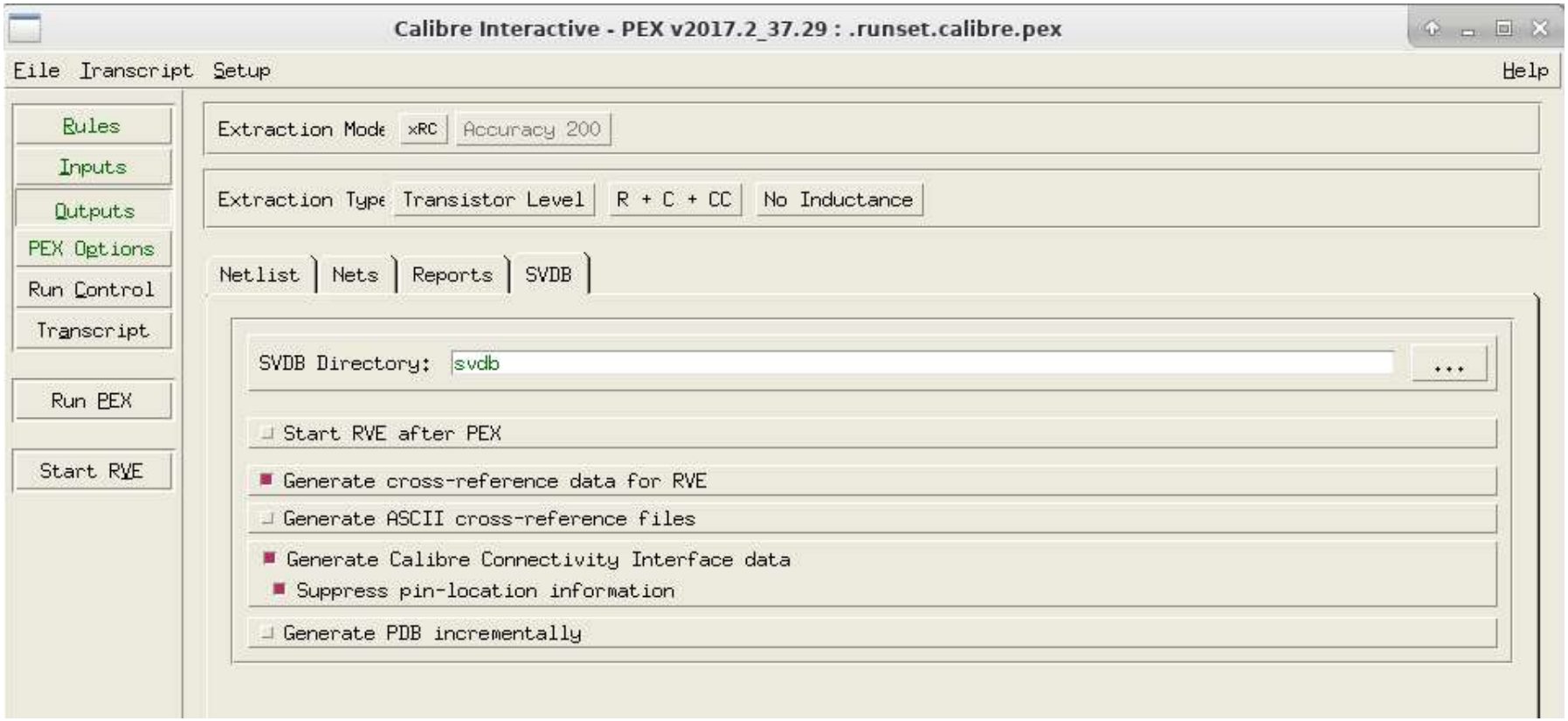
Follow same steps for input as in LVS tutorial.

Under Outputs tab, ensure R+C+CC is set as the Extraction Type.



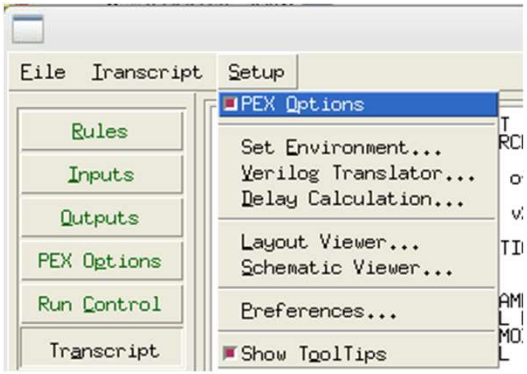
Parasitic Extraction (PEX)

In the SVDB tab, check "Generate Calibre Connectivity Interface Data" and "Suppress pin-location information"



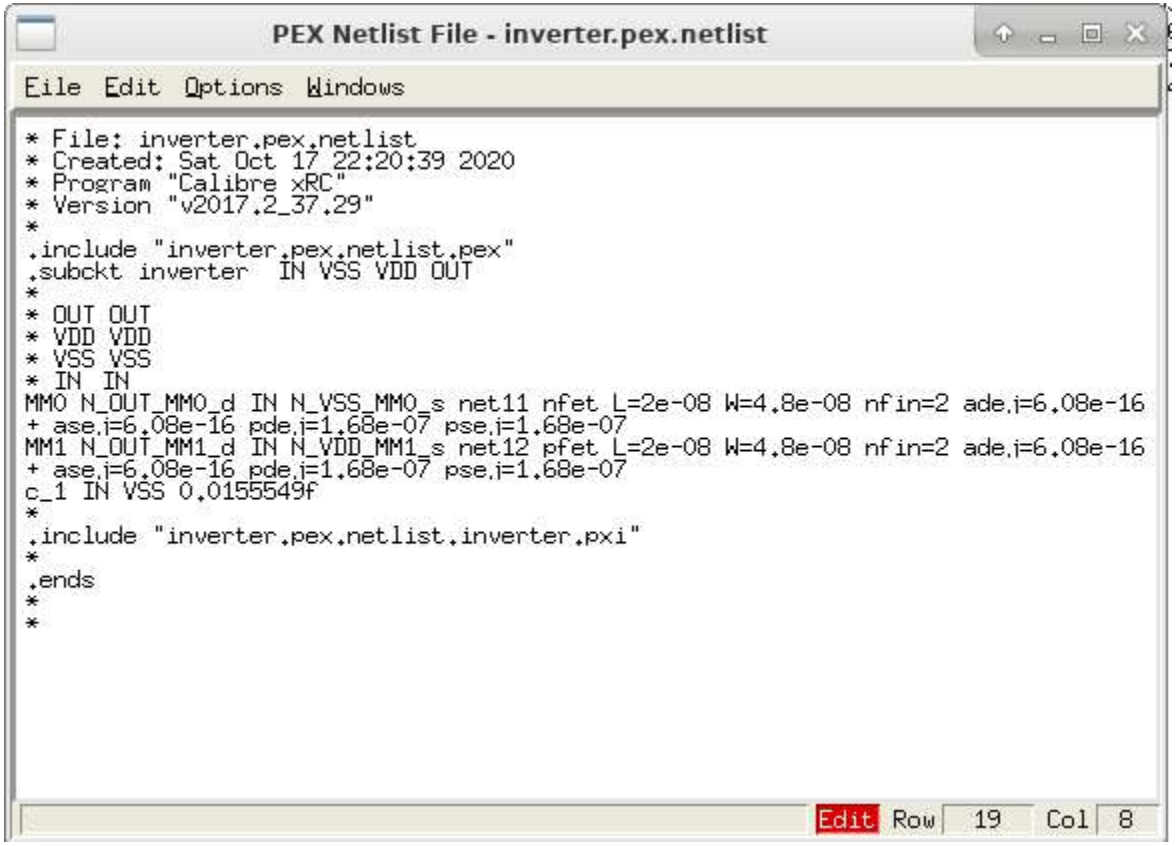
Parasitic Extraction (PEX)

In PEX window, go to Setup and click on “PEX Options”
Then, PEX options menu appears on the left toolbar.
Then, on that menu, define your ground net.
PEX needs a reference for parasitic calculation.



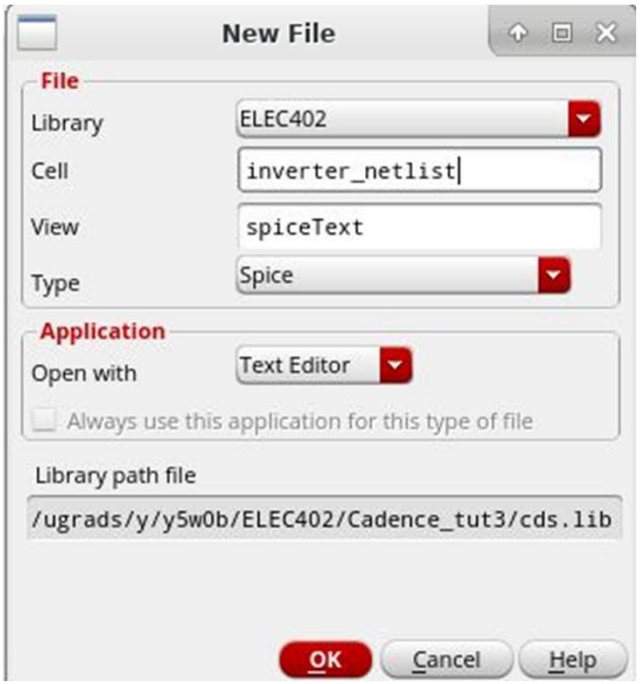
Parasitic Extraction (PEX)

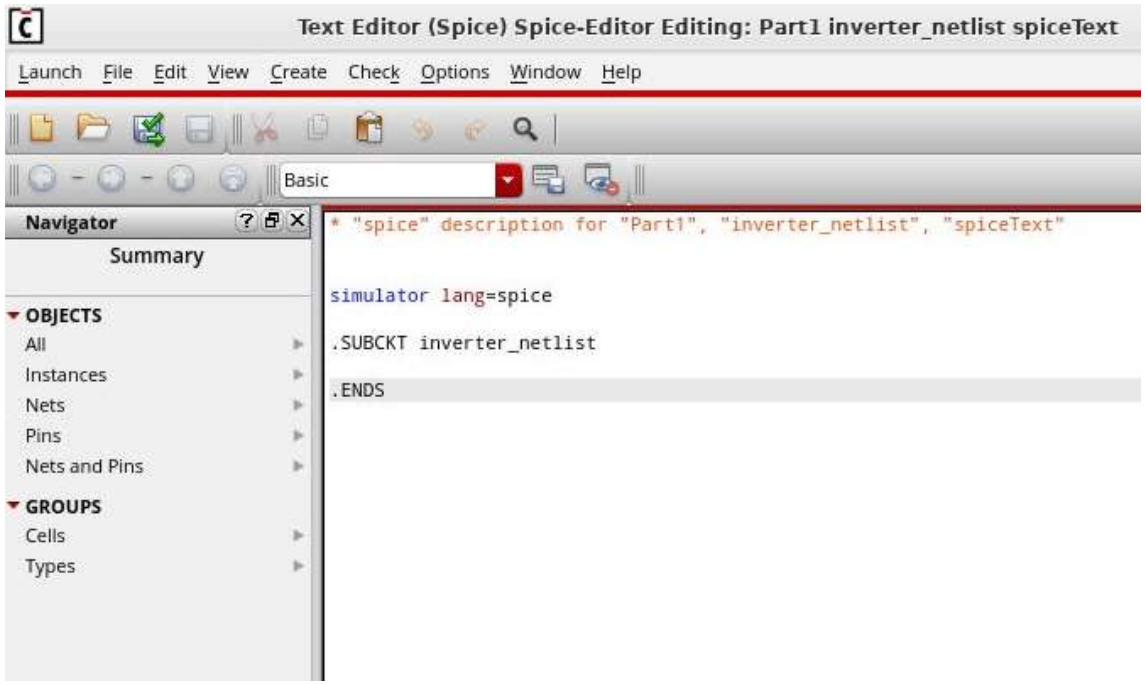
Run PEX.
Netlist is extracted.



Parasitic Extraction (PEX)

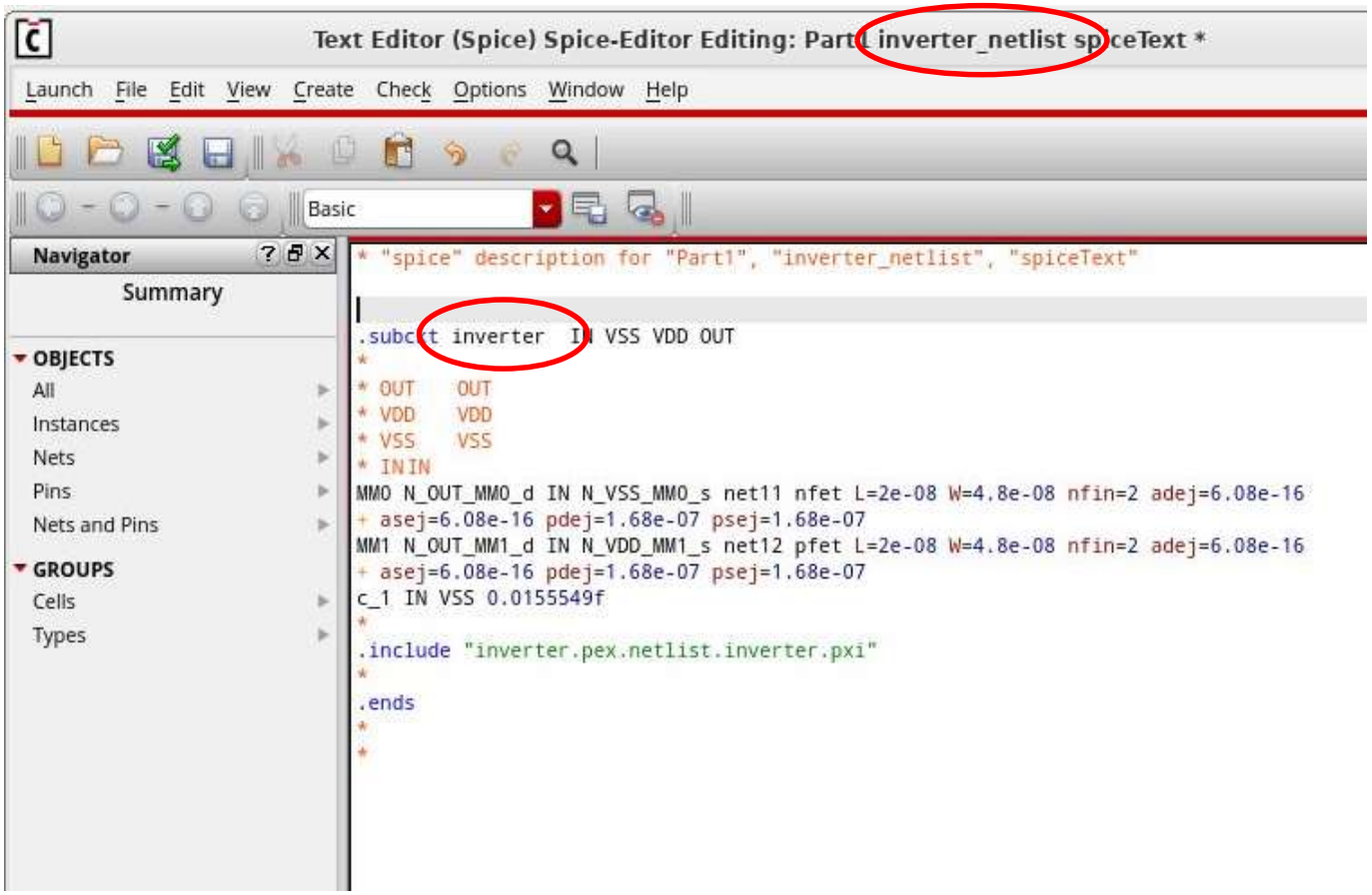
In Library Manager, create a new cell view and set the Type to “Spice”. This generates a spicetext which you can import your netlist.





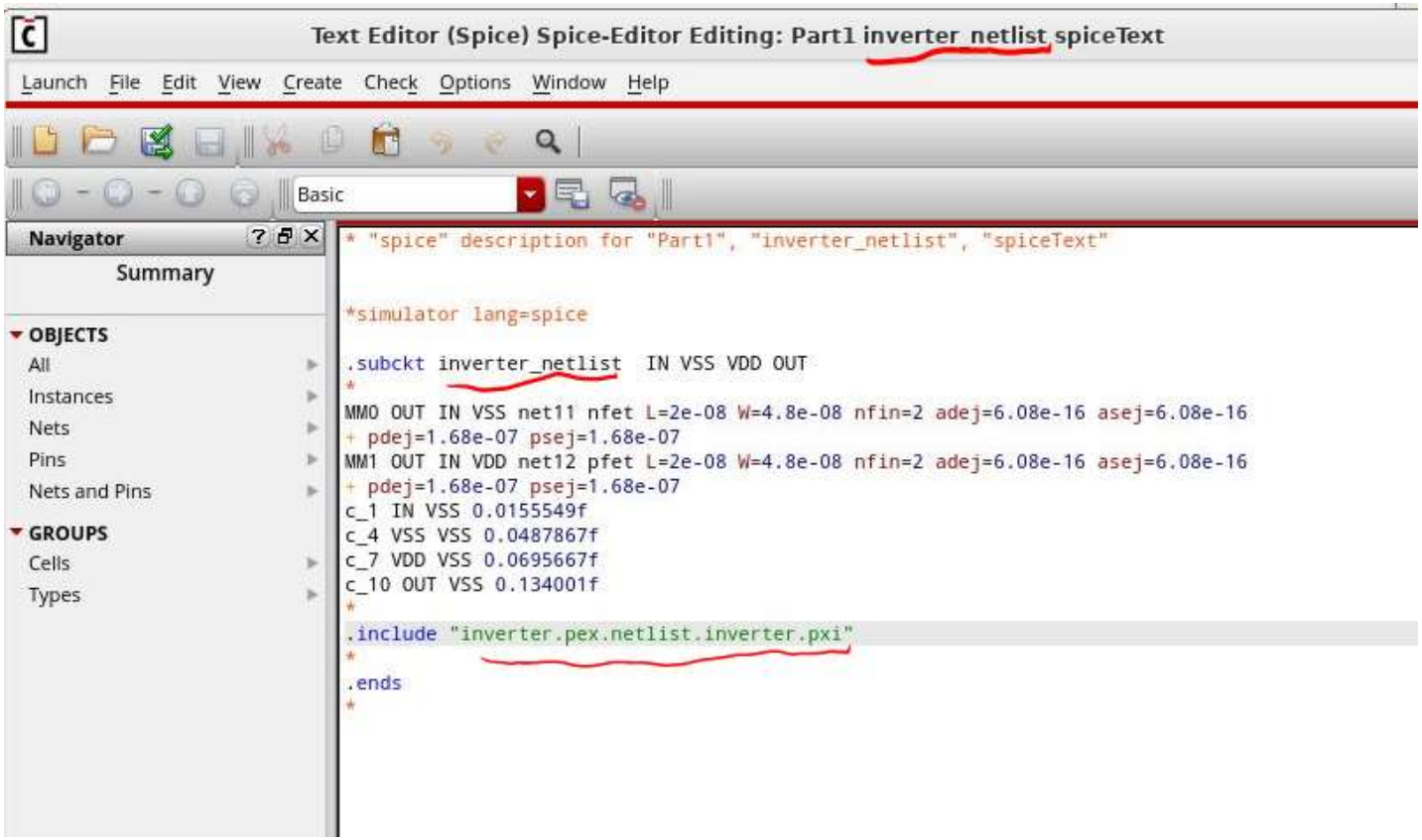
Parasitic Extraction (PEX)

Copy and paste the netlist.
You can delete “simulator lang=spice”.
NOTE: the name of the cell you are defining in the netlist should follow the cell name. In this example, it runs into errors.



Parasitic Extraction (PEX)

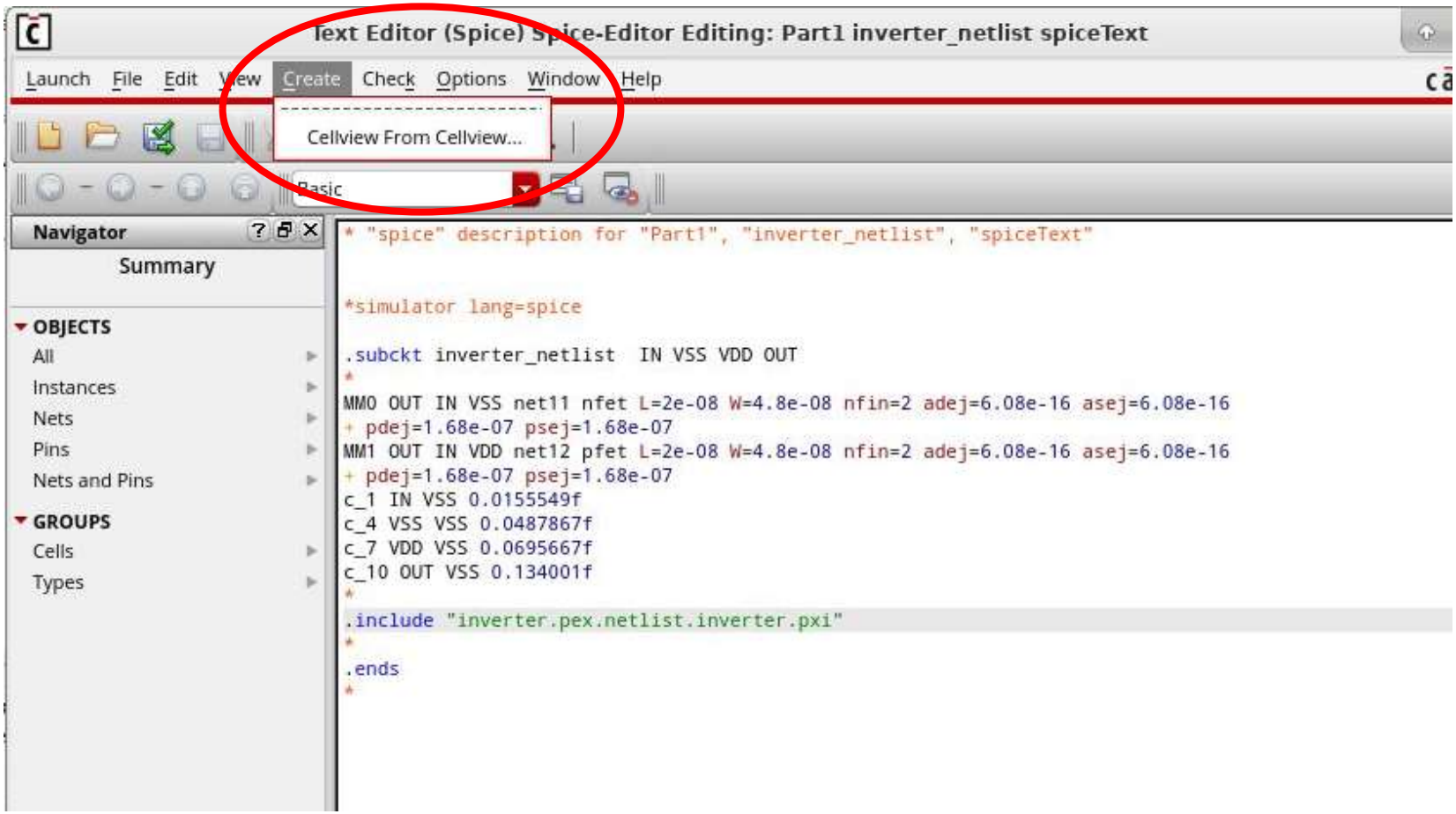
Corrected form:



Also PEX generates a netlist file, cellname.pex.netlist.cellname.pxi, which is included in the pex netlist. Copy this file to the spice cell folder so that can be included. Then check and save the spice code. *Copy and Paste all the files included in the netlist.

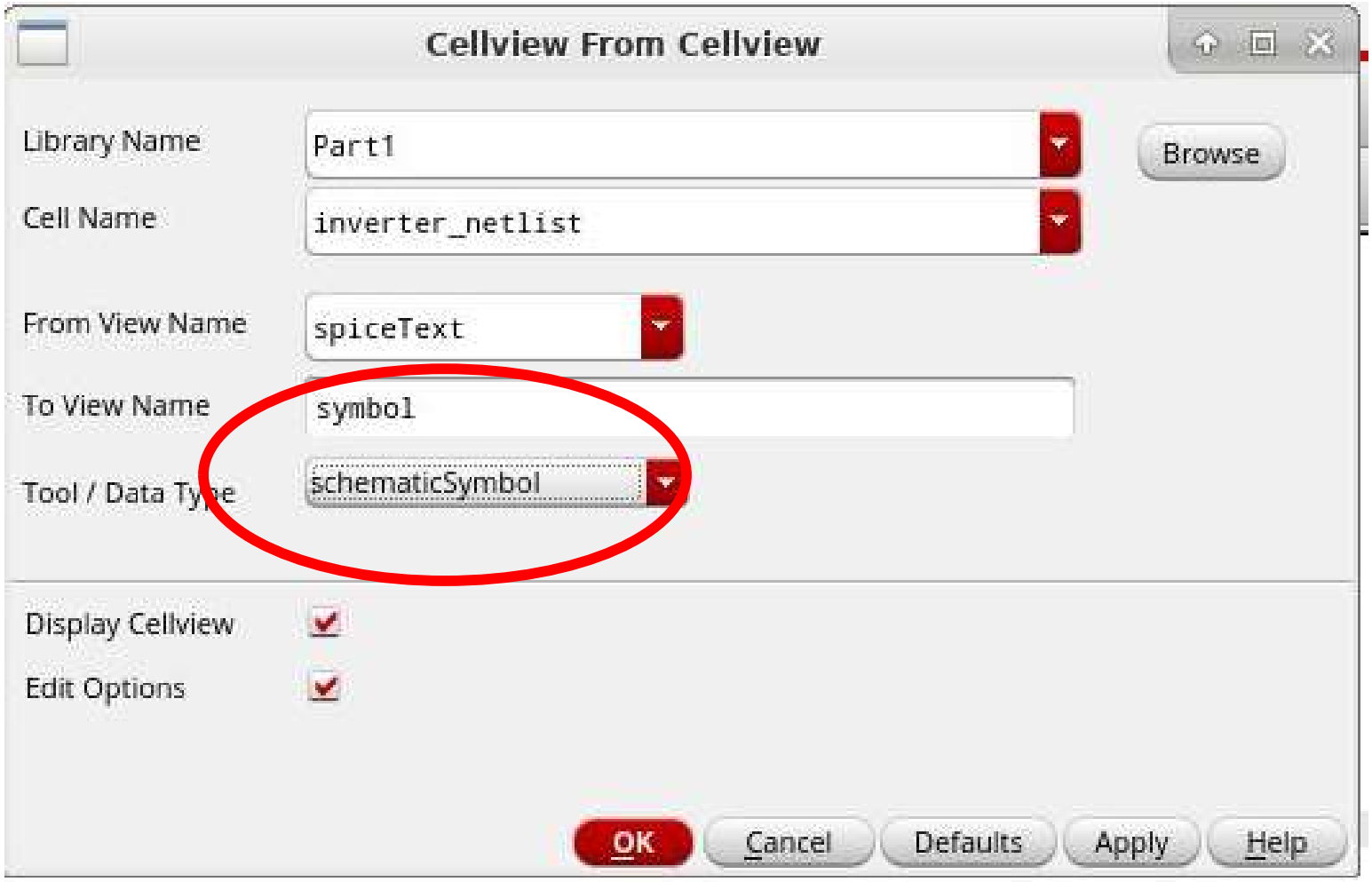
Parasitic Extraction (PEX)

Create symbol for spice netlist



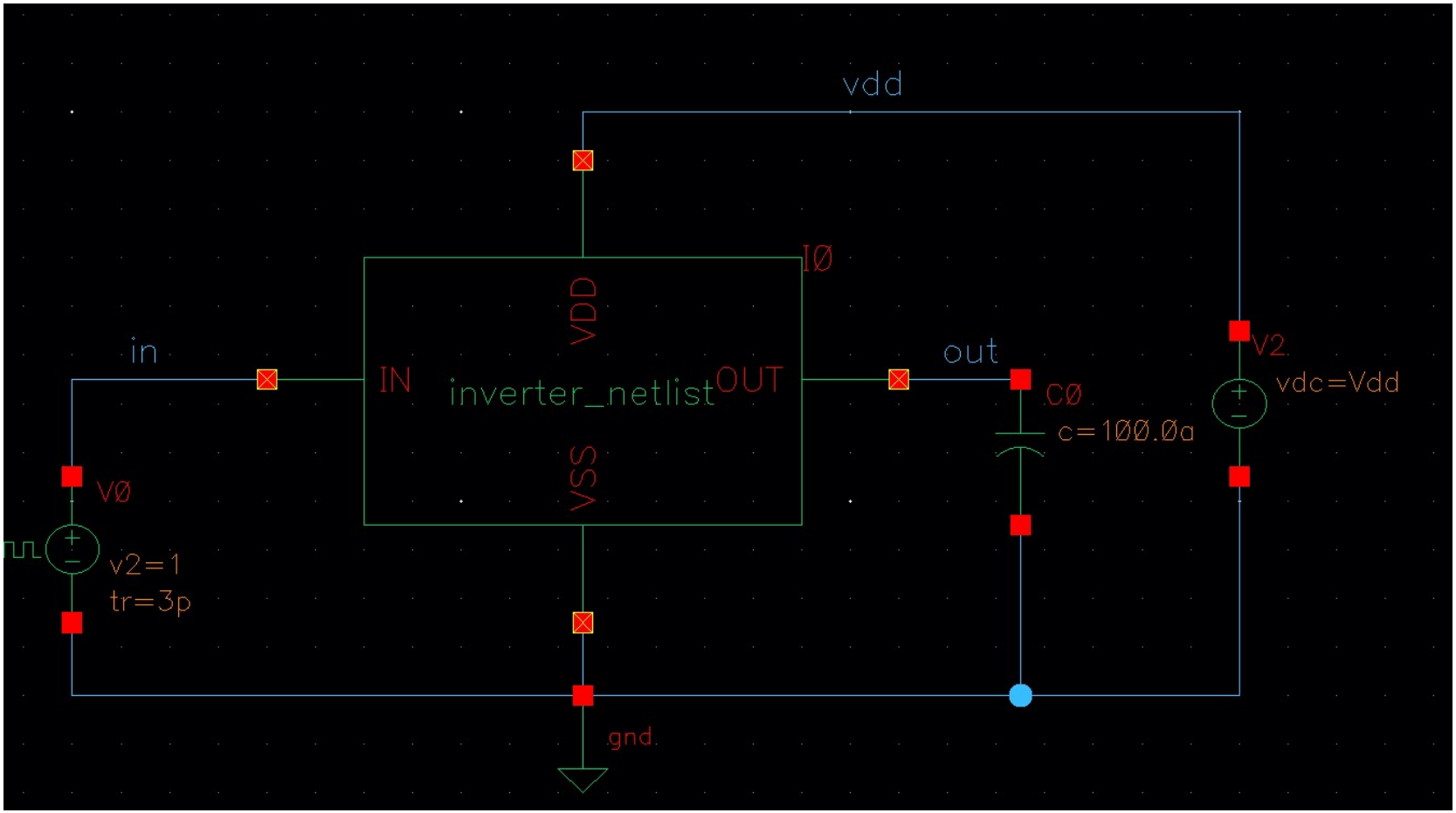
Parasitic Extraction (PEX)

Create symbol for spice netlist



Parasitic Extraction (PEX)

Create testbench and instantiate spice symbol.



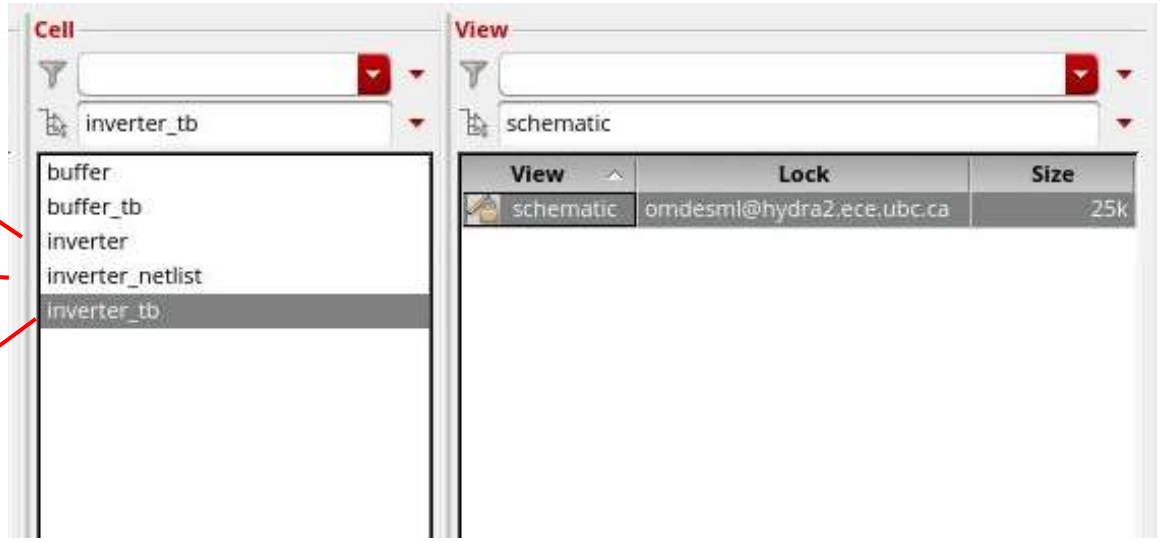
Parasitic Extraction (PEX)

Now we need a config cell for our testbench schematic to load the netlist within the symbol. This doesn't happen initially. Config cell gives you the ability to load different views for an imported instance (symbol).

My inverter schematic and layout

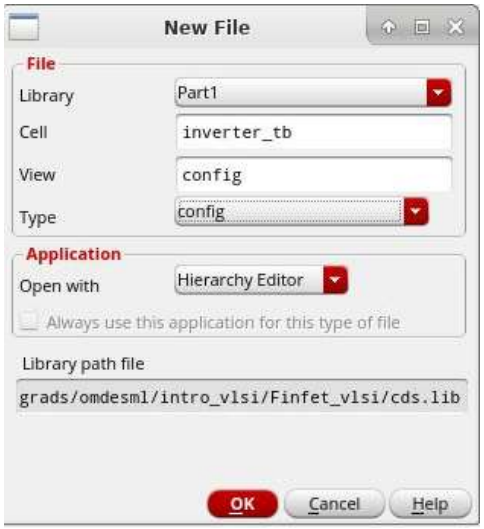
My post layout netlist with its symbol

My testbench which needs a config cell

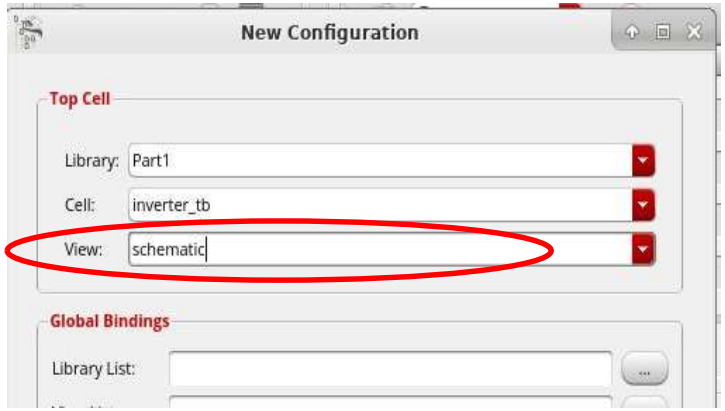


Parasitic Extraction (PEX)

1-New Cell

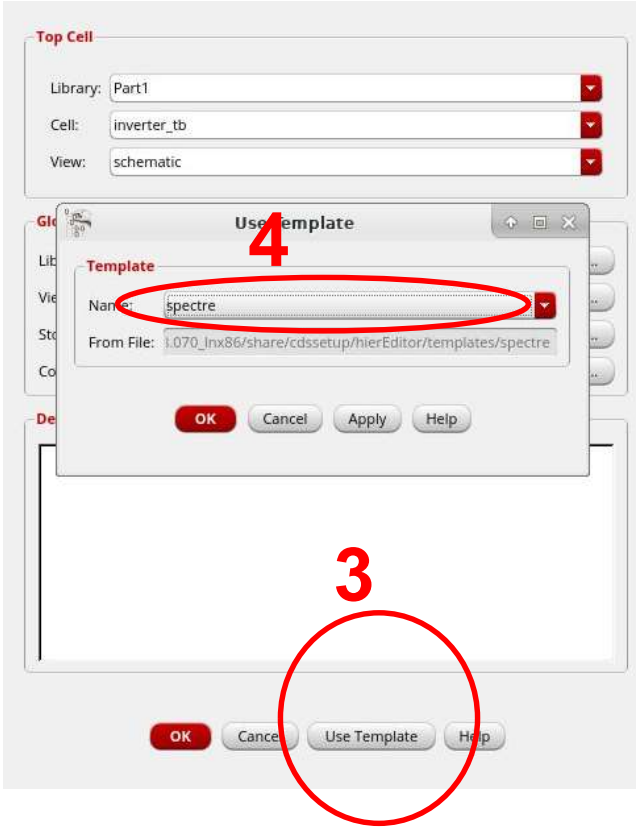


2-Choose the schematic



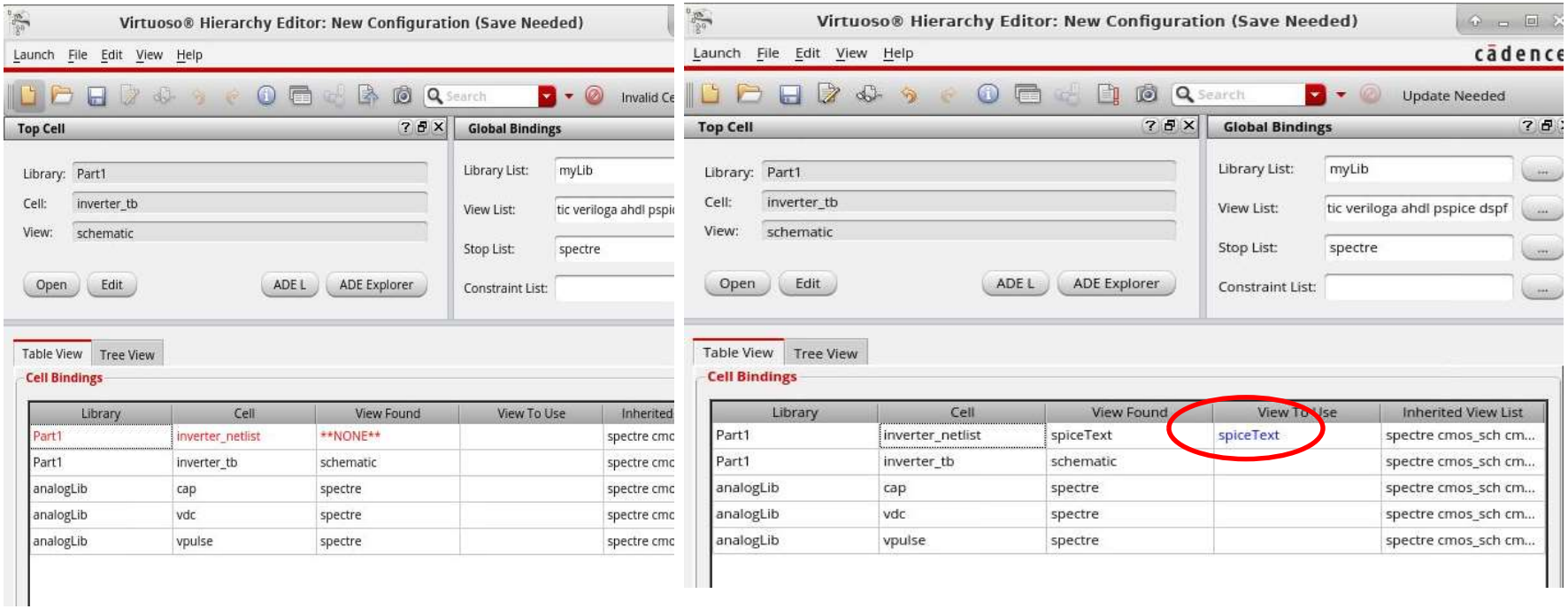
This indicates which cell needs a configuration setup (i.e. testbench)

2-Spectre as the template



Parasitic Extraction (PEX)

Right click on cell and choose the spice as the view to use in config window



Parasitic Extraction (PEX)

1. Launch ADE from configuration menu.
2. Set up simulator as spectre.
3. Set model libraries to:
→ /CMC/kits/ncsu_pdk/FreePDK15/hspice/models/fet.inc
→ CMG
4. Create simulation settings and simulate post-layout simulations