Tutorial – Cadence Virtuoso

Dae Hyun Kim

EECS
Washington State University

Cadence Virtuoso

- Used for full-custom (manual) physical design
- Download the following file.
  - wget https://eecs.wsu.edu/~ee434/Labs/tutorial_virtuoso.tar.gz
- Unzip it.
  - tar xvfz tutorial_virtuoso.tar.gz

Files

- Pre- and post-layout simulation
  - synopsys.sh, license-add.sh: used to run Synopsys HSpice
  - 45nm_PTM_HP_v2.1.pm: Transistor models for HSpice
  - myInv_X1_LVS.sp: An inverter netlist for pre-layout simulation, LVS, and PEX.
  - myInv_X1_pre.sp: An HSpice file for pre-layout simulation.
  - myInv_X1_post.sp: An HSpice file for post-layout simulation.

- Layout
  - cadence_ic618.sh: used to run Cadence Virtuoso
  - common_bindkeys.il, leBindKeys.il, schBindKeys.il: Key binding
  - display.drf: Display resource file
  - tech_ng45nm.tf: Tech file

- DRC, LVS, PEX
  - mentor_calibre_17.sh: used to run Mentor Calibre
  - rules/*.rul: DRC, LVS, PEX rule files

What We Will Do

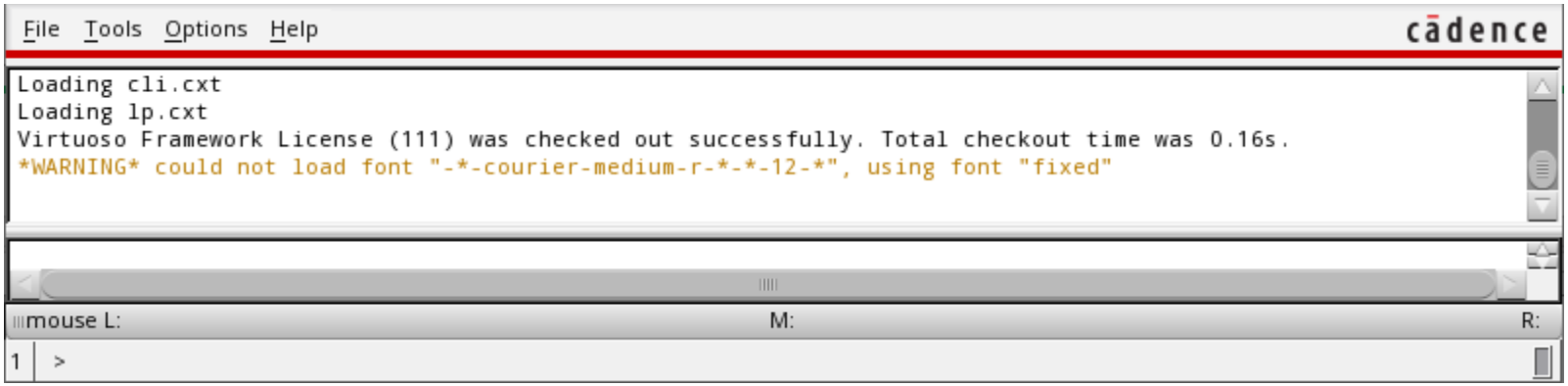
- Layout
- DRC (Design Rule Check)
- LVS (Layout vs. Schematic)
- xRC (PEX, parasitic extraction, RC extraction)
- SPICE simulation
  - Although we need to do pre-layout simulation before drawing a layout, we will just to it after PEX just to compare pre- and post-layout simulation results.

How to Launch Virtuoso

- Source the following files.
  - source ictools_generic.sh
  - source cadeice_ic618.sh

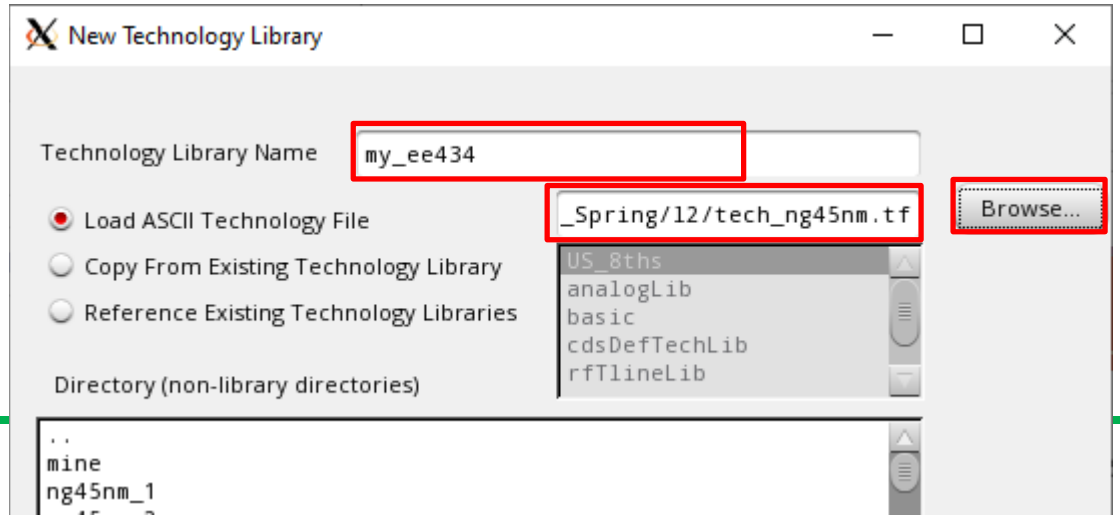
- Run Virtuoso.
  - virtuoso

- You will see the following window (called Command Interpreter Window, CIW).



Create a Library

- In CIW
  - “Tools” → “Technology File Manager...”
- In the Technology File Manager window
  - “New...”
- In the New Technology Library window
  - Enter a library name you want.
  - Click “Browse...” and select tech_ng45nm.tf.
  - Click OK.
- You will see a message box saying “... loaded successfully.”

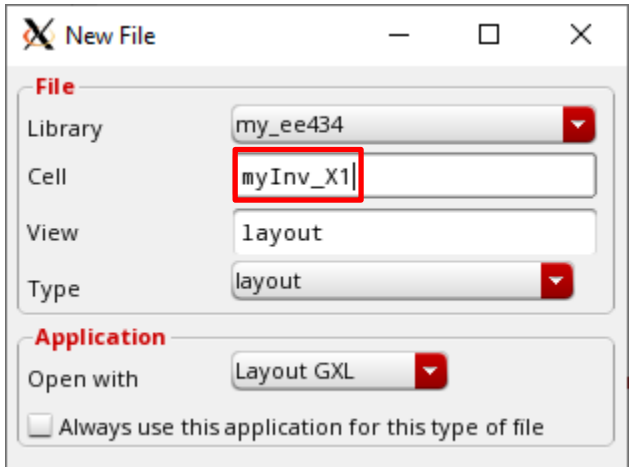


Create a Library

- Close the Technology File Manager window.
- In CIW, “Tools” → “Library Manager...”
- In the leftmost column, you will see your library name.

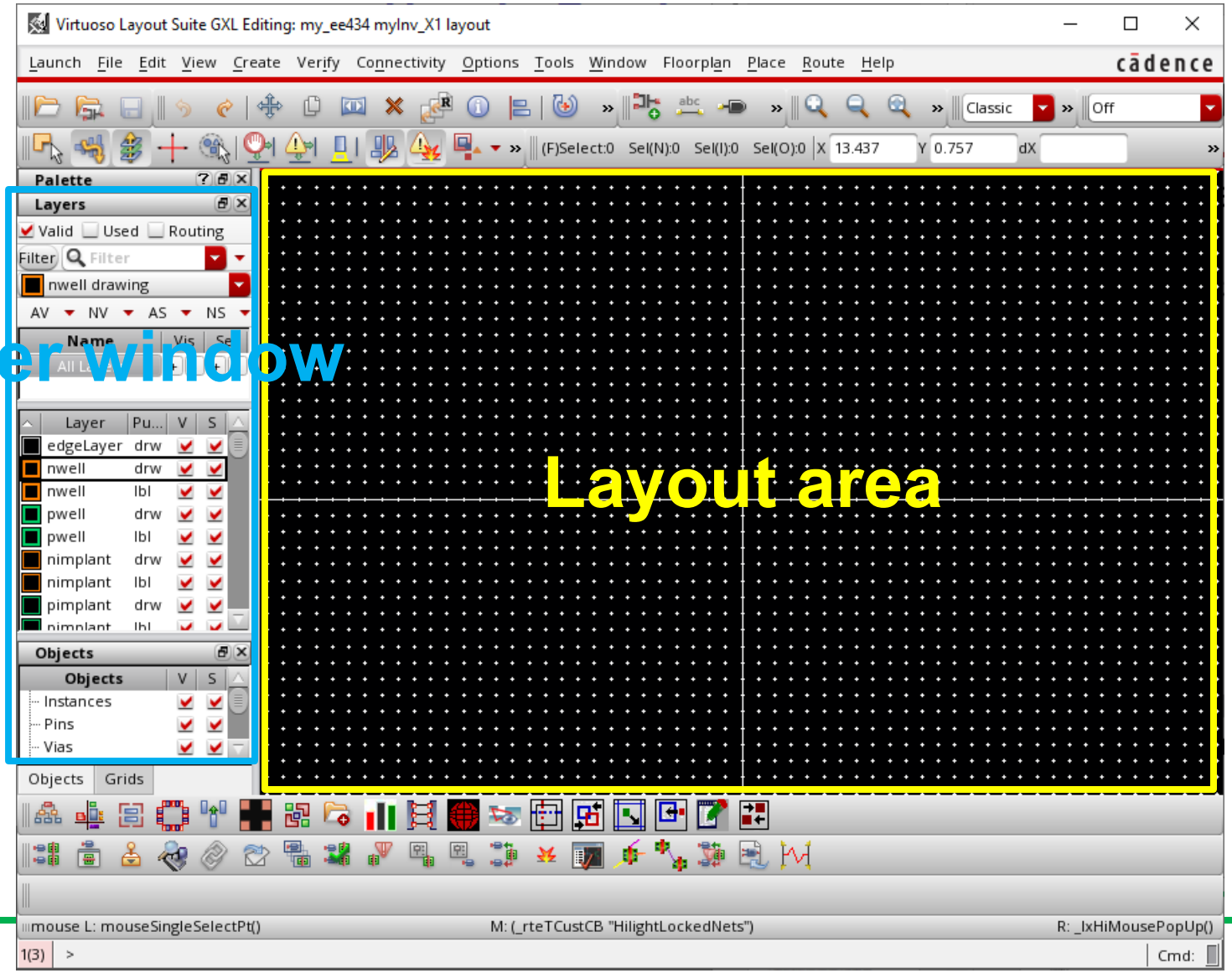
How to Create a Cell (Gate)

- In the Library Manager window
  - Select your library name.
  - Click “File” → “New” → “Cell View...”.
  - Enter a cell name and click OK.



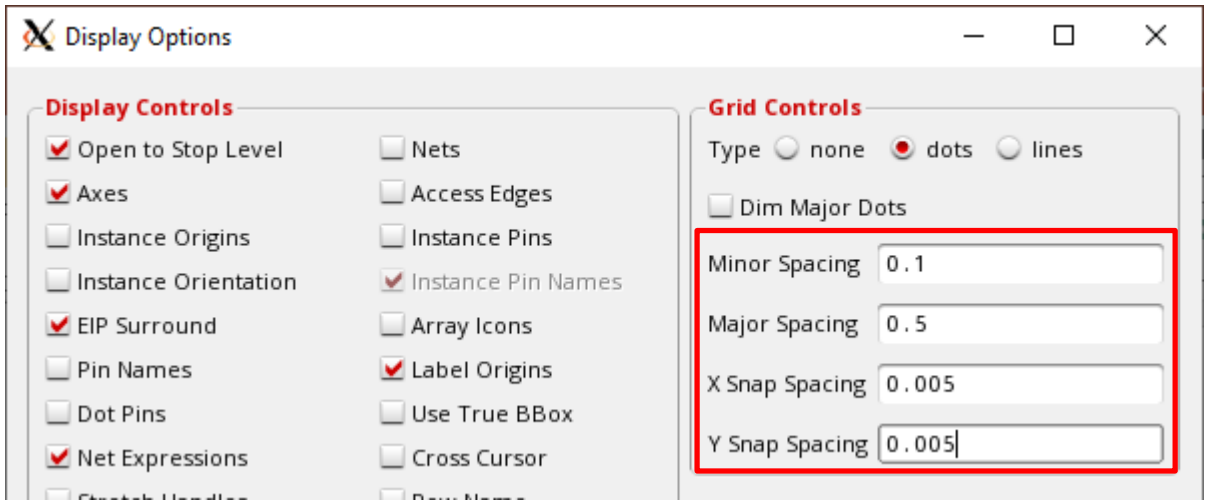
  - Then, it will add a new cell into your library and automatically open a layout window.

Layout Window



Editor Setup

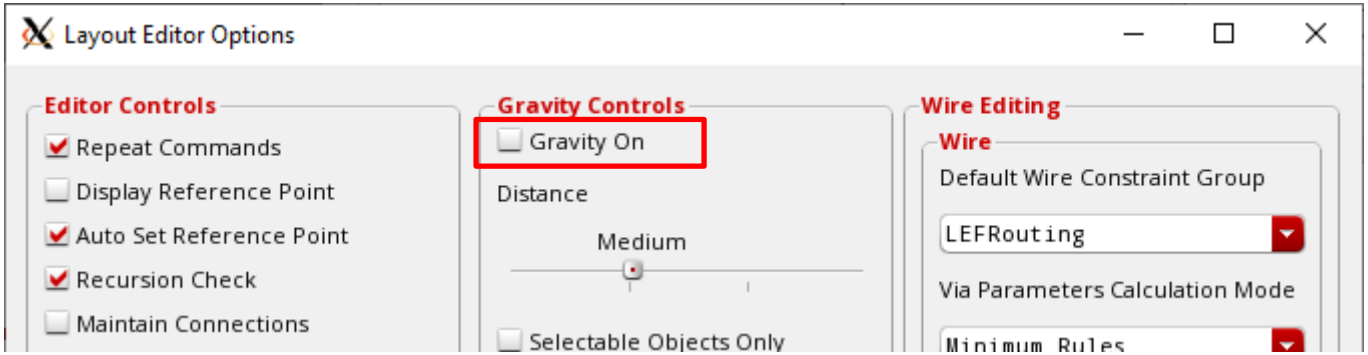
- Press “e” or Click “Options” → “Display...” and use the following setting.



- You can click “Save To” to save the current setting.

Editor Setup

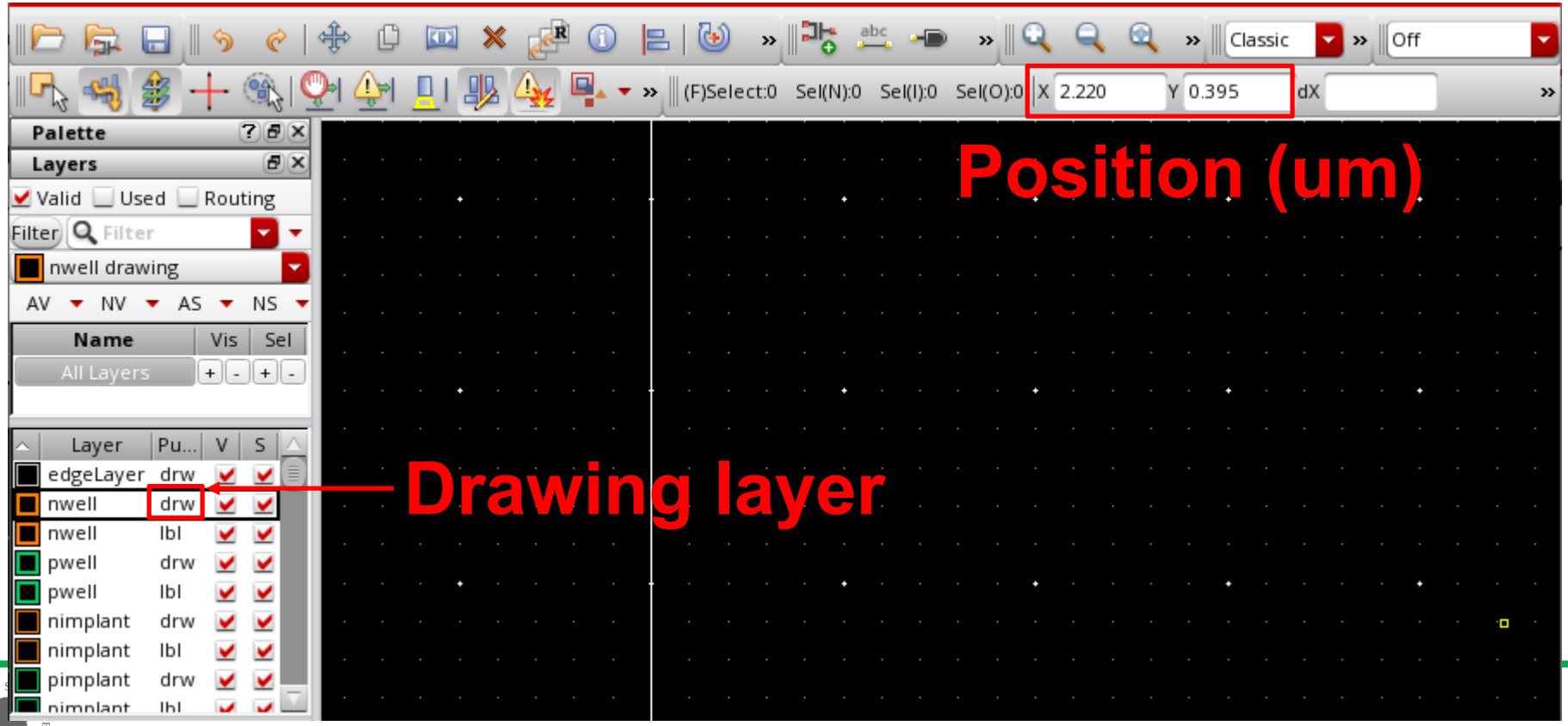
- Press “Shift+e” or Click “Options” → “Editor...” and use the following setting.



- Leave it unchecked.
- You can click “Save To” to save the current setting.

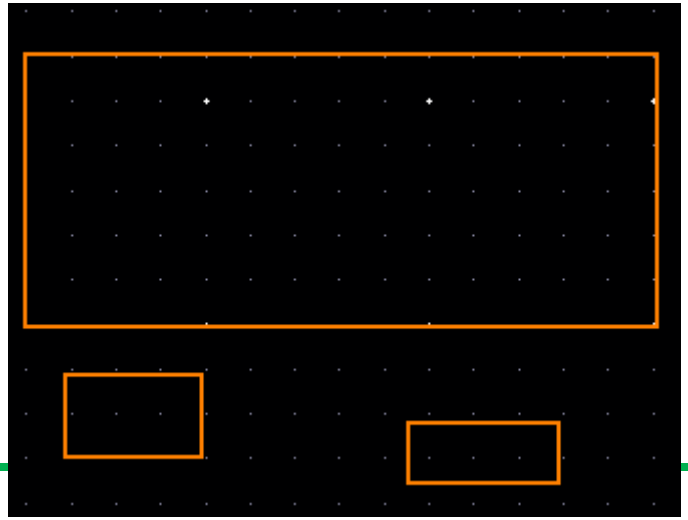
How to Draw Objects

- Length unit: um
- Use the right mouse button to zoom in (Shift+z: zoom out).
- Choose a layer you want to draw in the layer window.
  - Choose “drawing” for the “Purpose”.



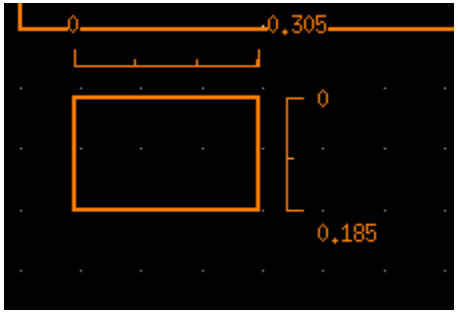
How to Draw Objects

- Now, let's draw a rectangle.
- Click "Create" → "Shape" → "Rectangle" or just press "r". Now you are ready to draw a rectangle of the selected layer.
- Draw a rectangle by clicking the left mouse button, move the cursor, and click the left mouse button again.
- After that, you are still in drawing mode, i.e., you can continue to draw more rectangles. Draw a few more rectangles.
- To get out of drawing mode, press "ESC".



How to Draw Objects

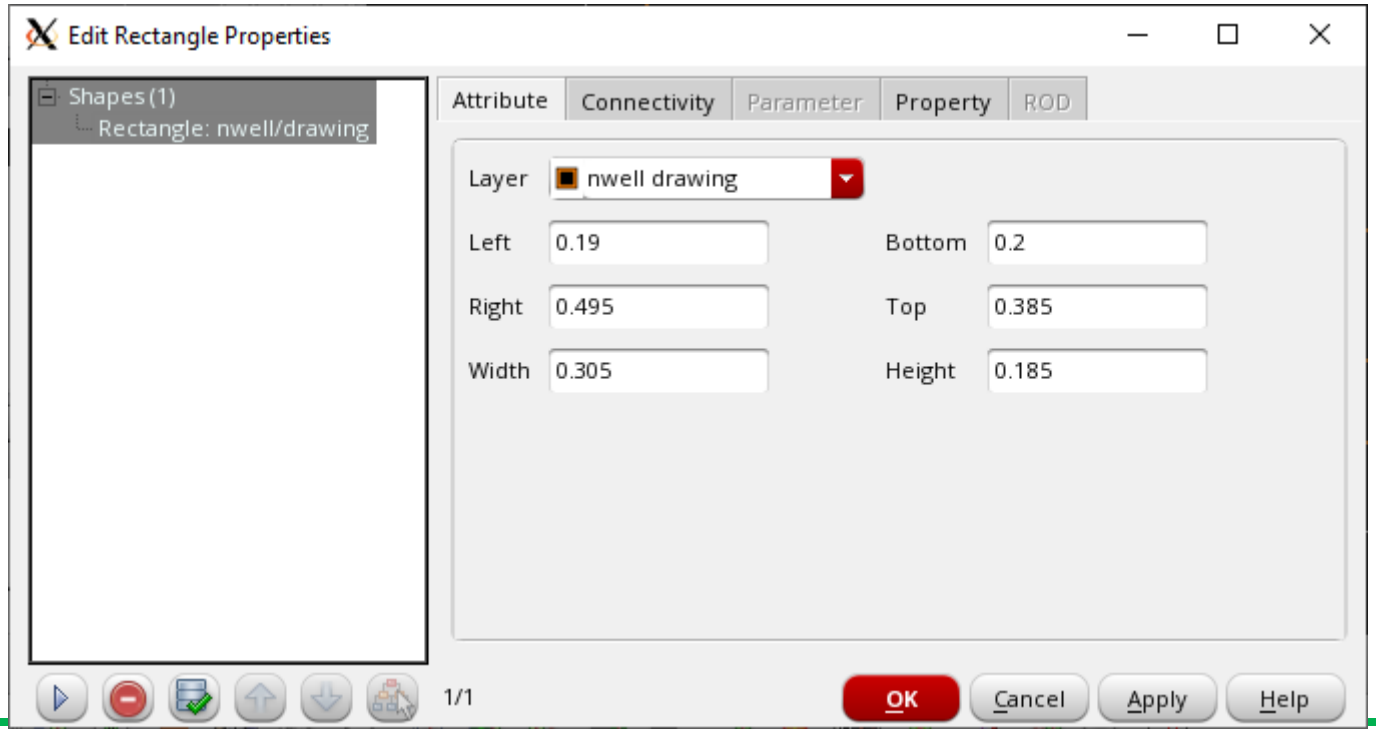
- If you want to delete an object, select the object (make sure you are not in drawing mode by pressing “ESC”) and press the delete key.
  - u: undo
- Now, let’s measure the length of the rectangle you drew.
  - k: ruler



  - Notice that you should get out of measuring mode by pressing “ESC”.
  - Shift+k: delete all measurements.

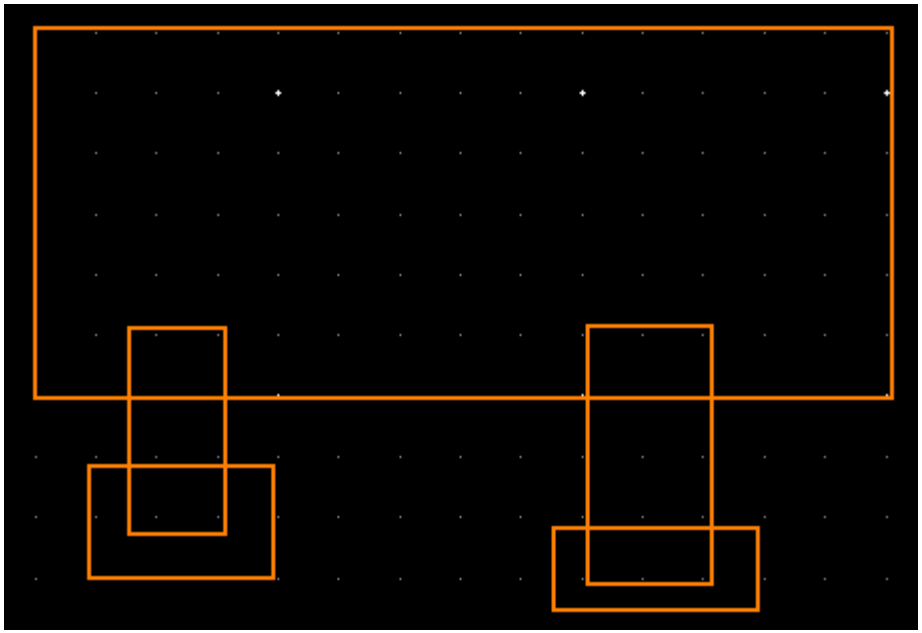
How to Draw Objects

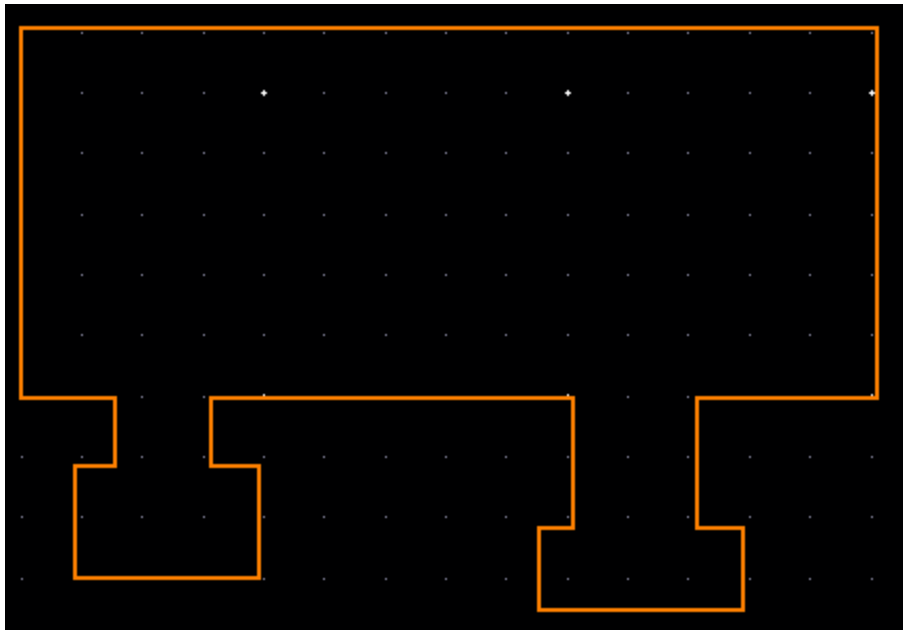
- Select a rectangle.
- Press “q” to see the property of the rectangle.
- You can fine-control the coordinates in this window.
  - Notice that the snap size is 0.005um (5nm), so don’t enter unaligned numbers.



How to Draw Objects

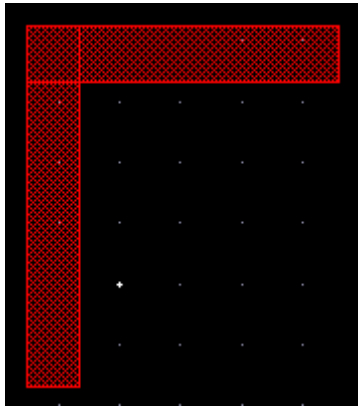
- If you want to unselect the currently-selected object, just click on a white space.
- How to merge objects.
  - Draw them, select them all, then Shift+m.
  - This is very useful if you want to merge objects into a polygon.

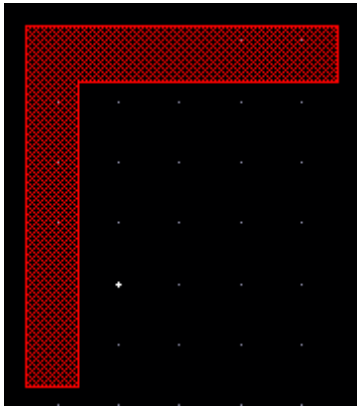




How to Draw Objects

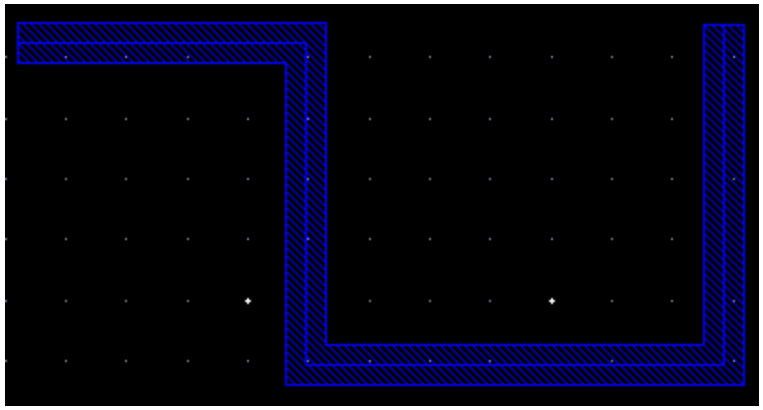
- Suppose you have two overlapped objects. Are they electrically connected? Yes, they are. Thus, you don't need to merge them.
- However, if you merge them, they form a polygon, which is a single object. In this case, if you want to move it, for example, you can just move a single object. If you have multiple, separate (unmerged) objects, you should select and move all of them.





How to Draw Objects

- You can draw wires using rectangles, but aligning is sometimes quite painful.
- You can use “path” for that.
- “Create” → “Shape” → “Path”
- Click, click, ..., double-click.



How to Draw Objects

- Unselect objects by clicking on white space.

- Move: Press “m”, select an object, move it, press ESC.
- Copy: Press “c”, select an object, move it, click, then ESC.
- Stretch: Press “s”, click on a boundary, stretch it, click, then ESC.
- Ruler: Press “k”.
  - Shift + k: Clear all the rulers.
- Merge: Select objects, then Shift+m.
- Save: F2.
- Zoom in: Mouse right-click, hold, drag, and release.
- Zoom out: Shift+z
- Zoom to fit: “f”

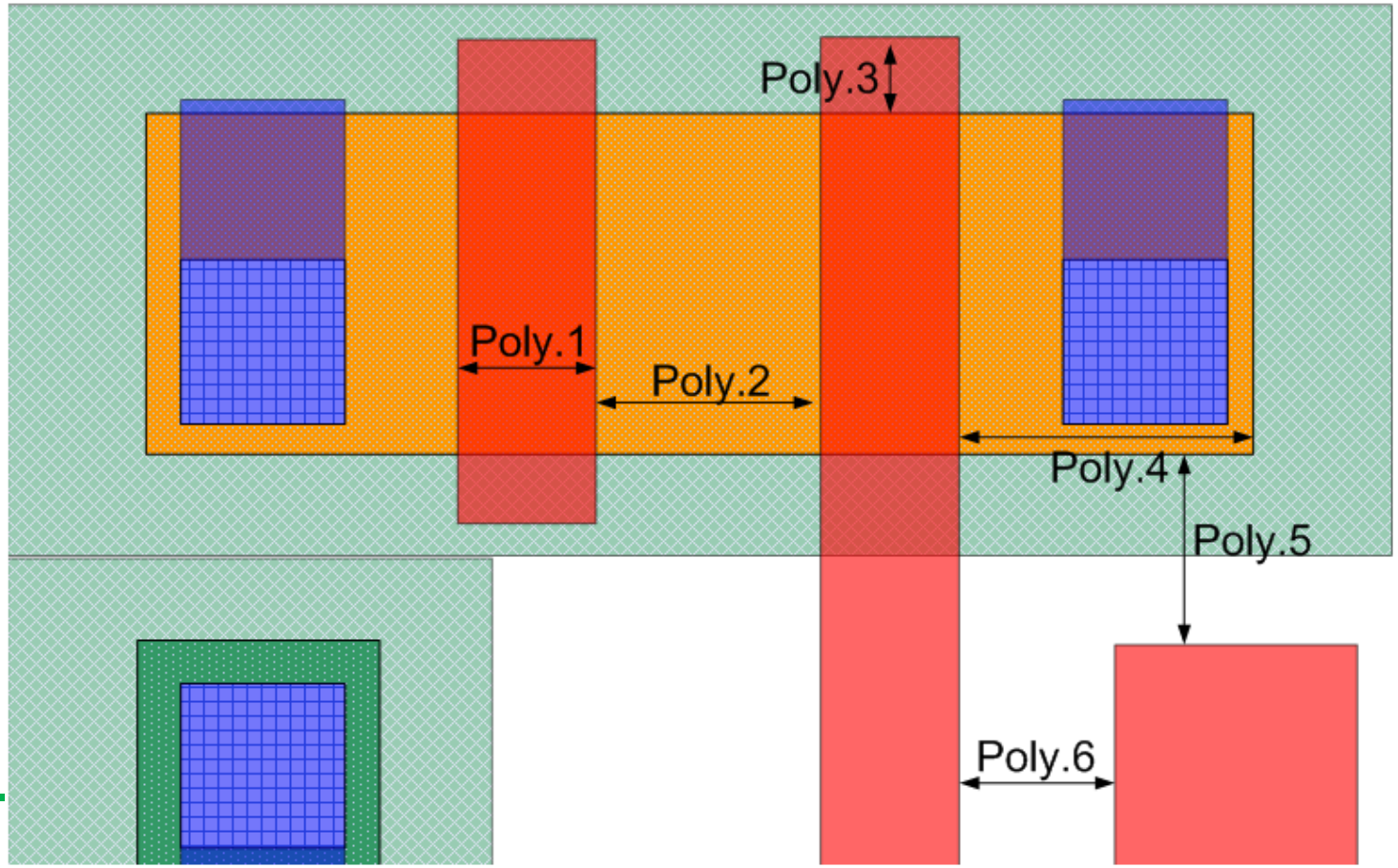
Design Rules

- See the following page.
  - http://www.eda.ncsu.edu/wiki/FreePDK45:Contents

- Click each layer under “Design Rules”.

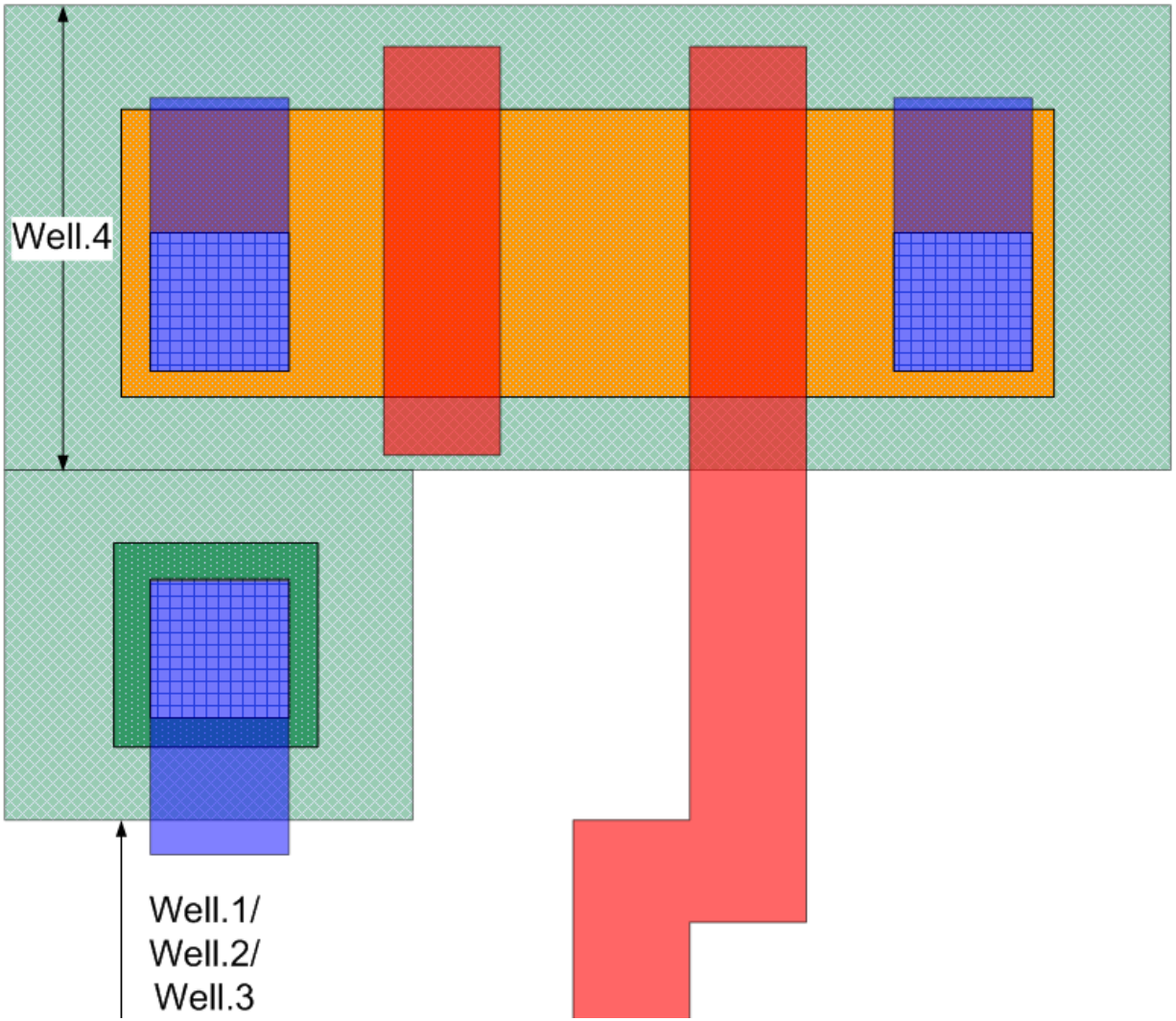
FreePDK45:PolyRules

Rule	Value	Description
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 Minimum spacing of field poly



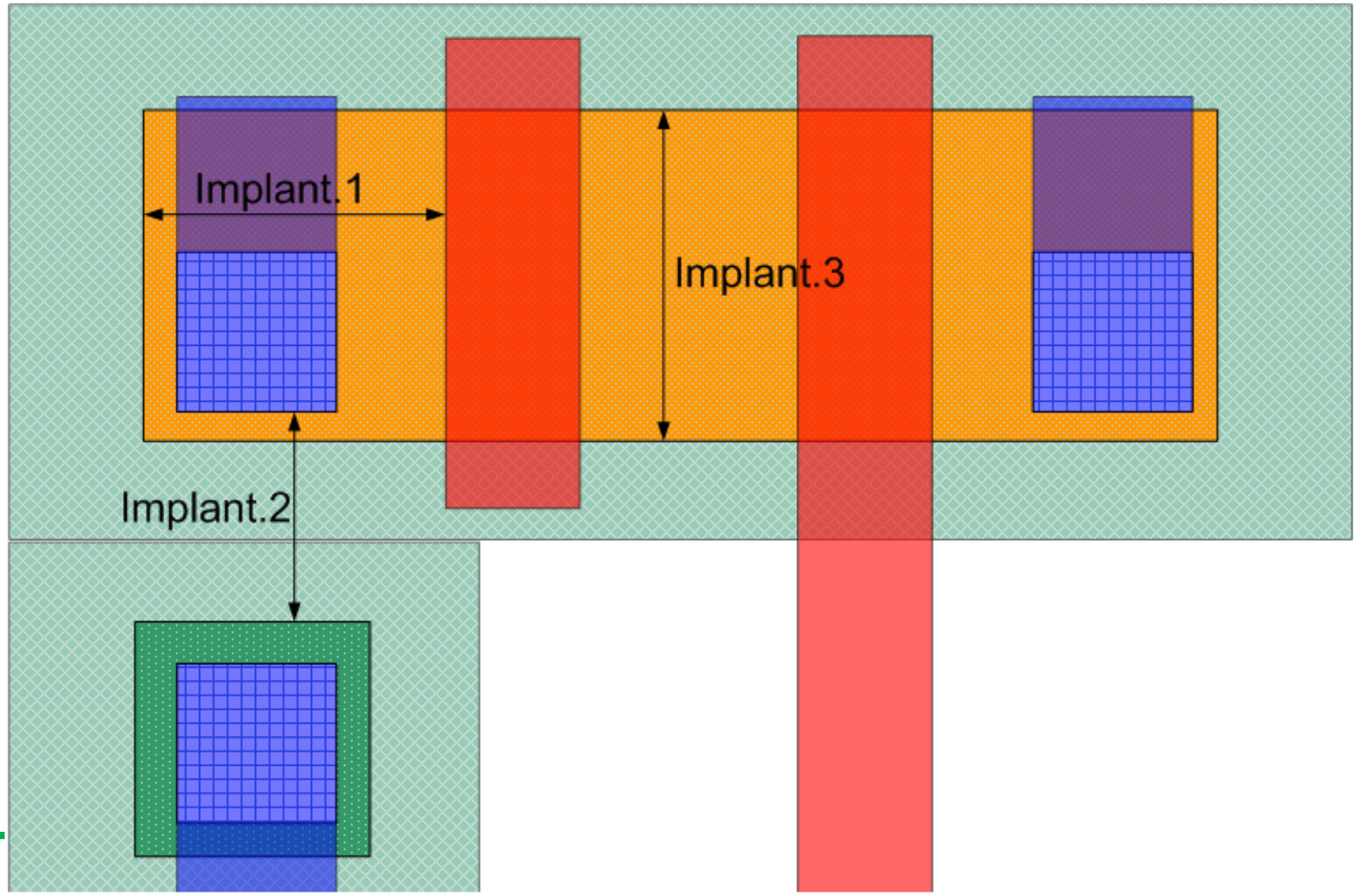
FreePDK45:WellRules

Rule	Value	Description
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



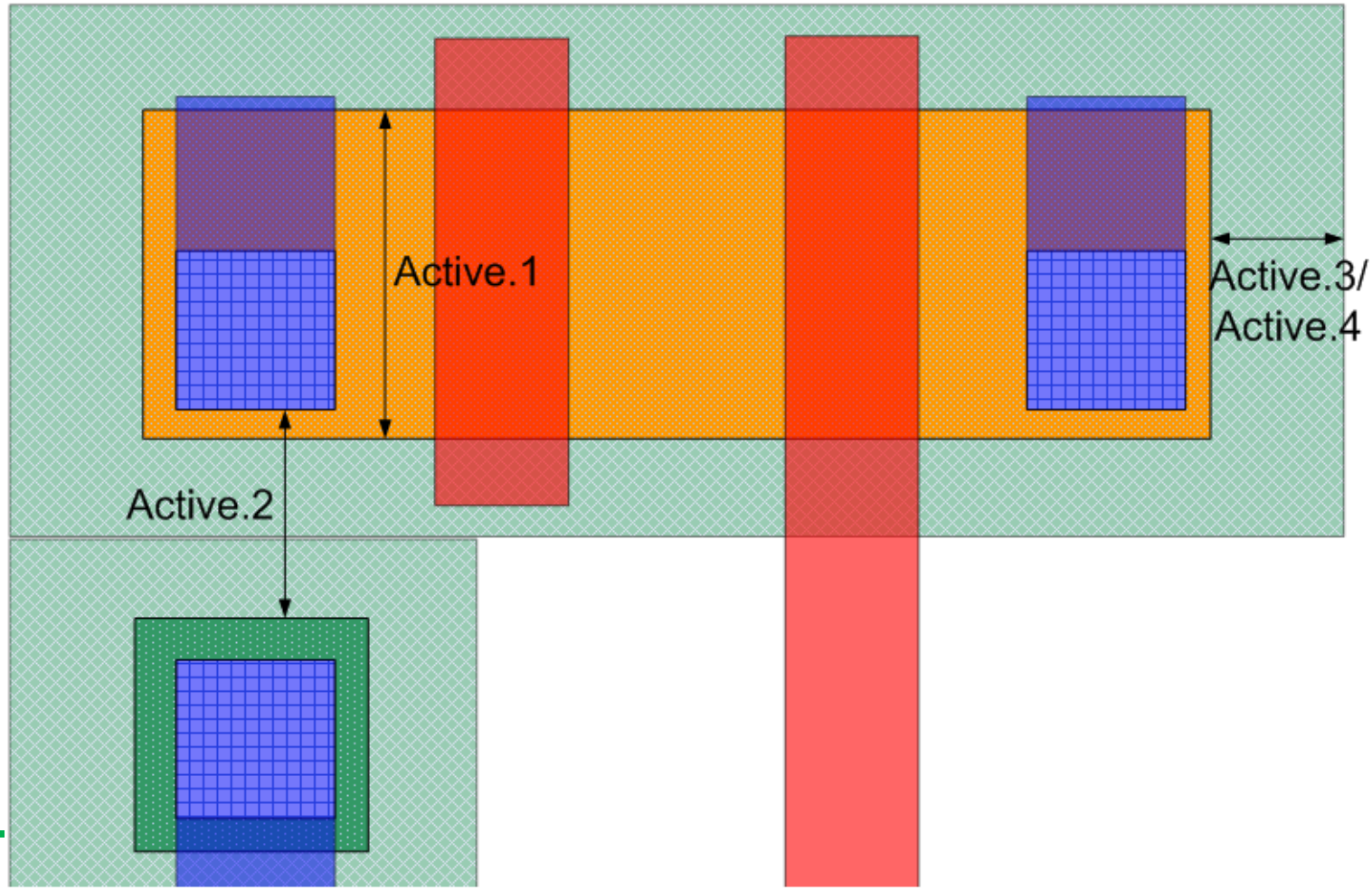
FreePDK45:ImplantRules

Rule	Value	Description
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 pimplant must not overlap



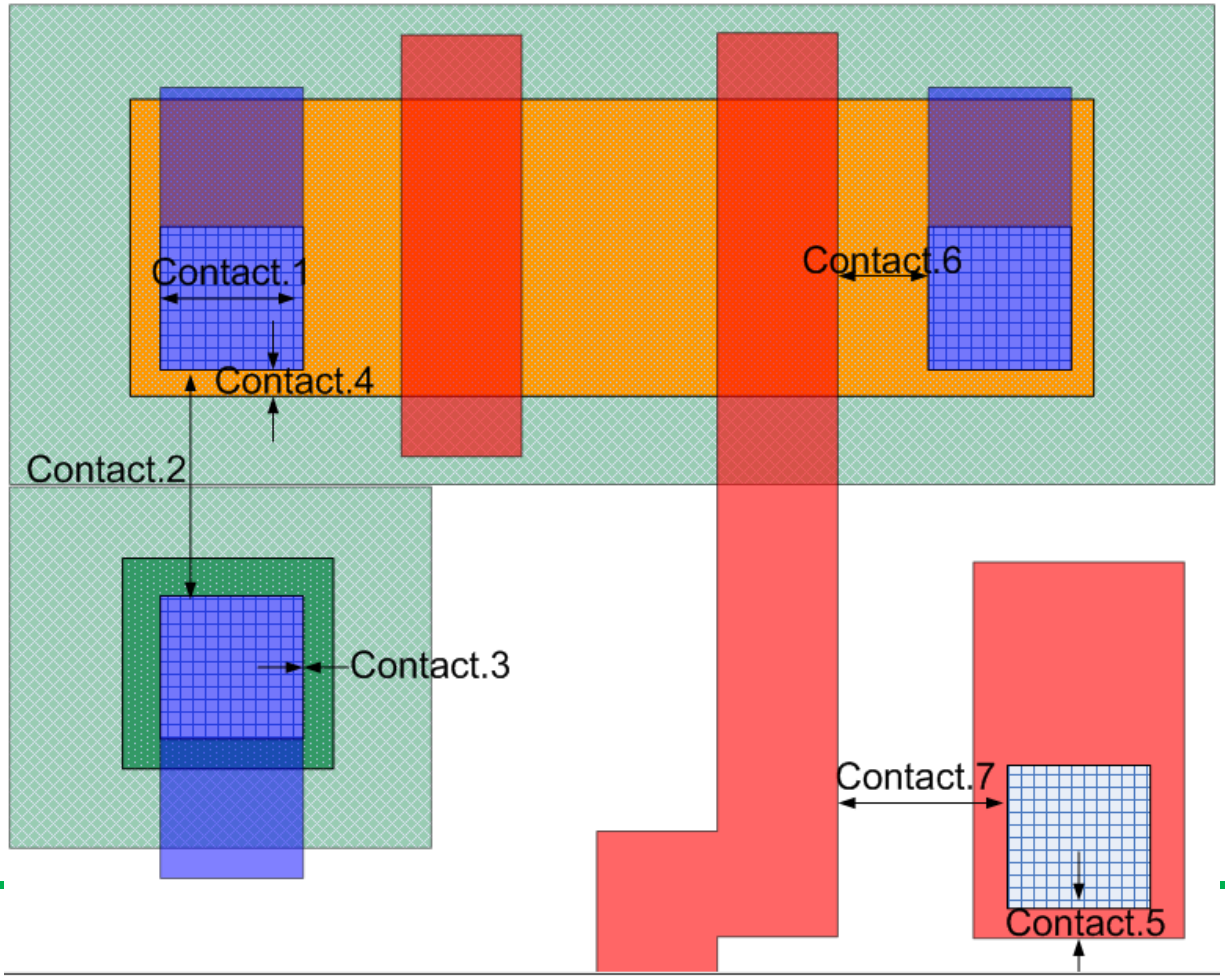
FreePDK45:ActiveRules

Rule	Value	Description
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



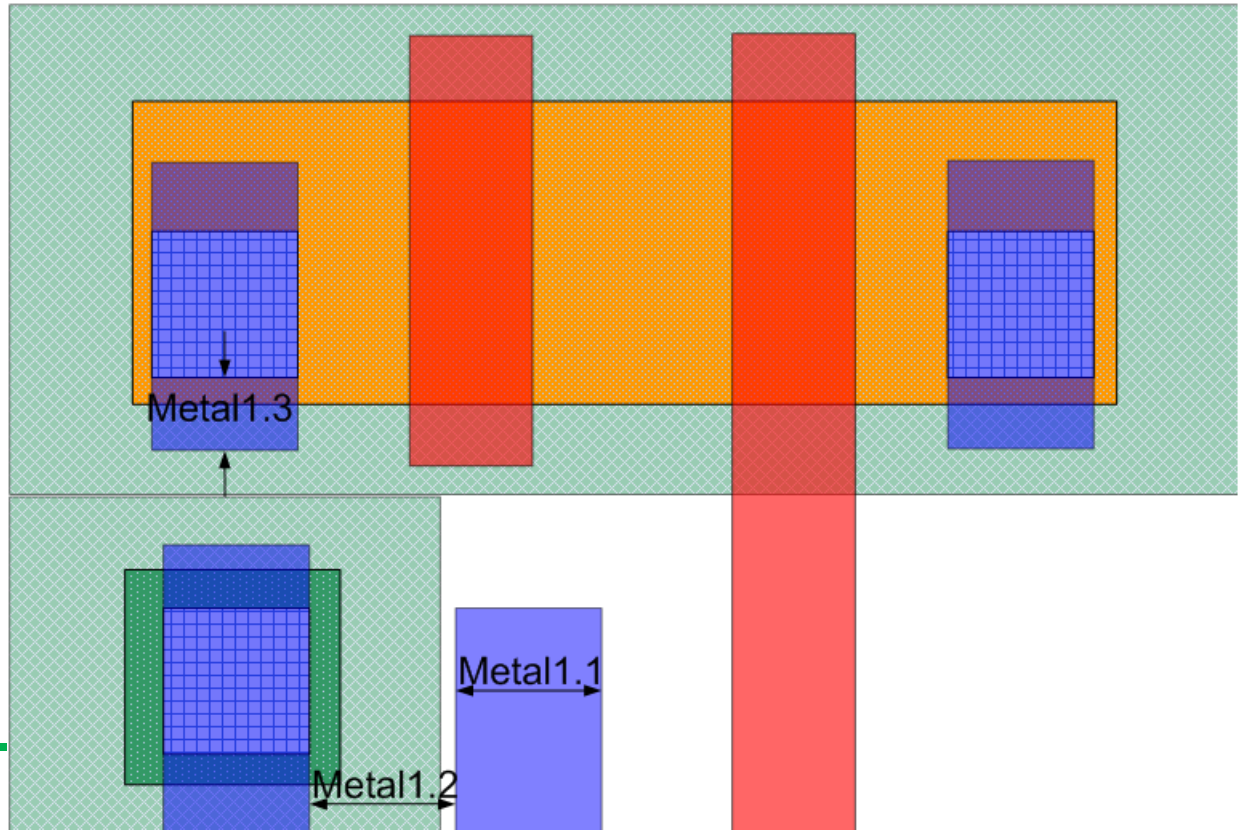
FreePDK45:ContactRules

Rule	Value	Description
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



FreePDK45:Metal1Rules

Rule	Value	Description
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

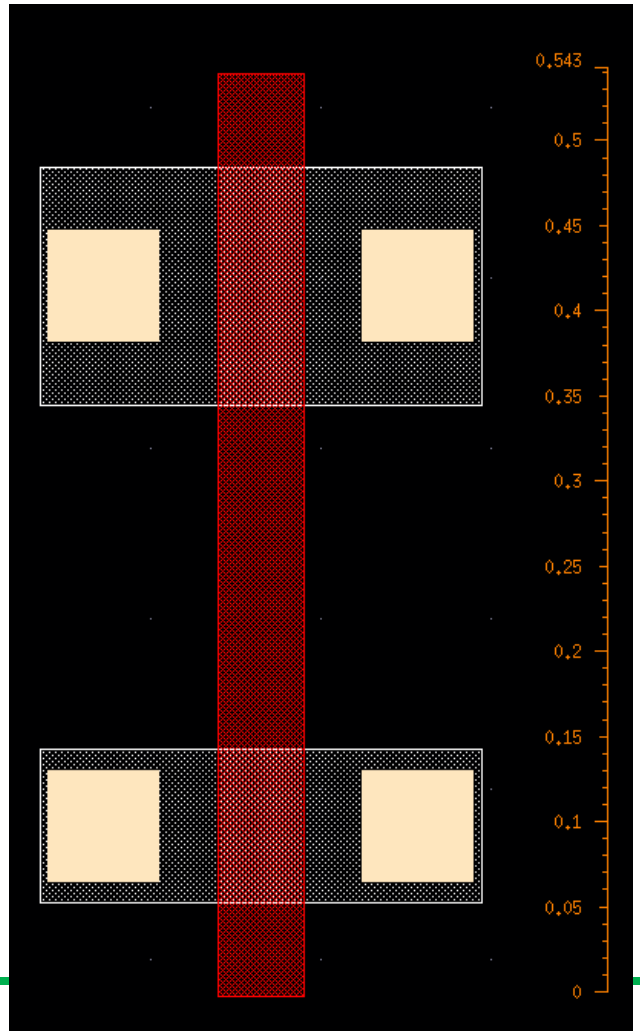


FreePDK45:Via1Rules

Rule	Value	Description
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

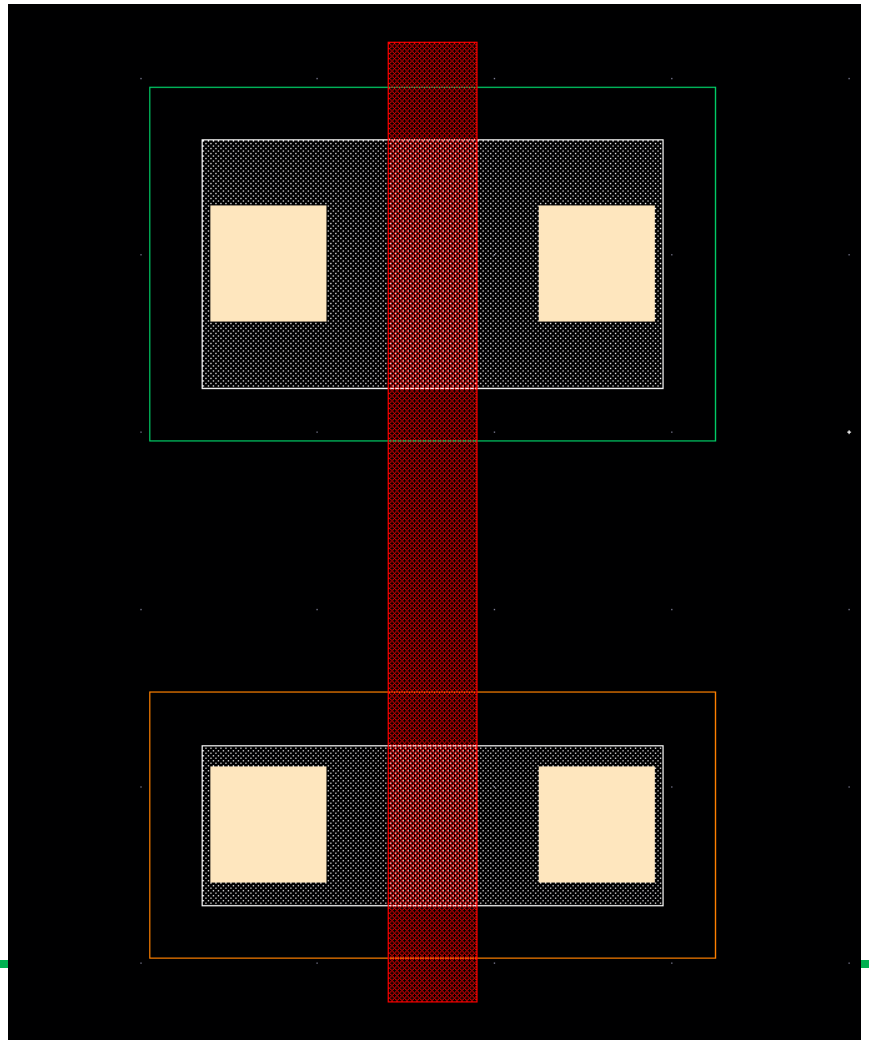
Example – Inverter

1. Poly, Active, and Contact (Ln=50nm, Wn=90nm, Lp=50nm, Wp=140nm)



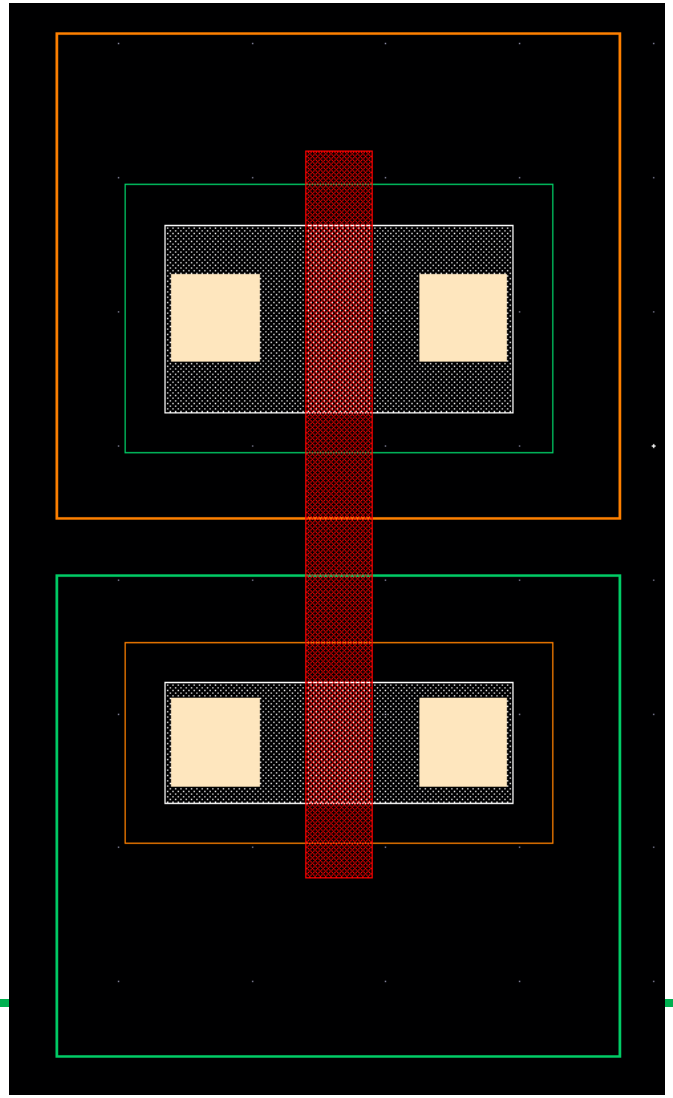
Example – Inverter

2. pimplant and nimplant



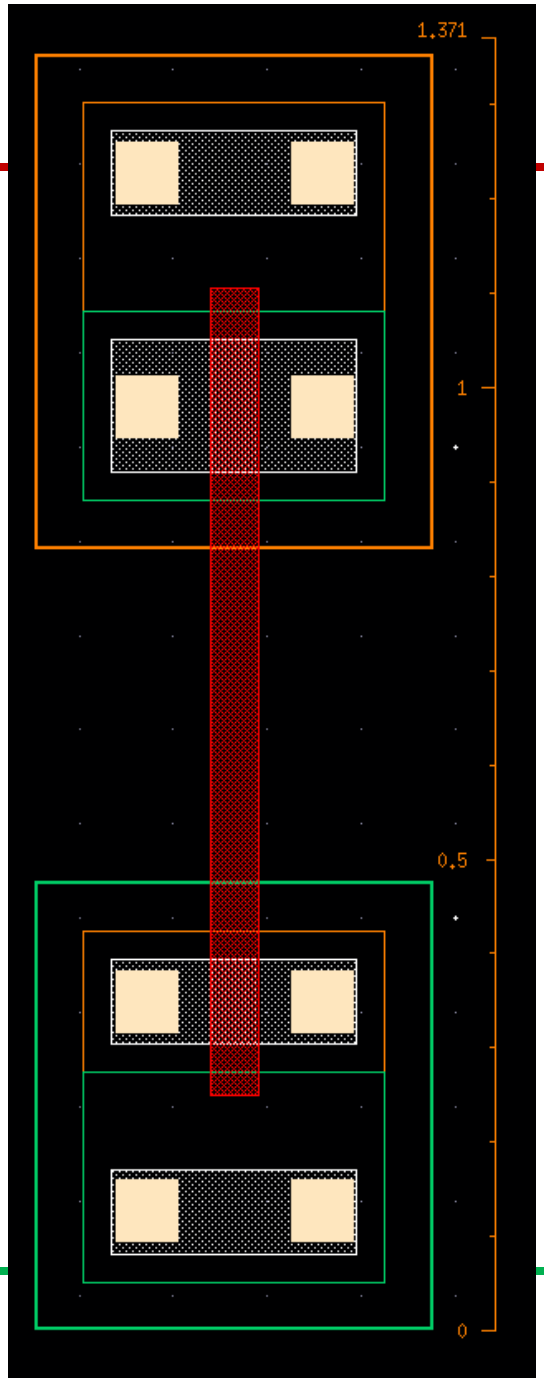
Example – Inverter

3. pwell and nwell



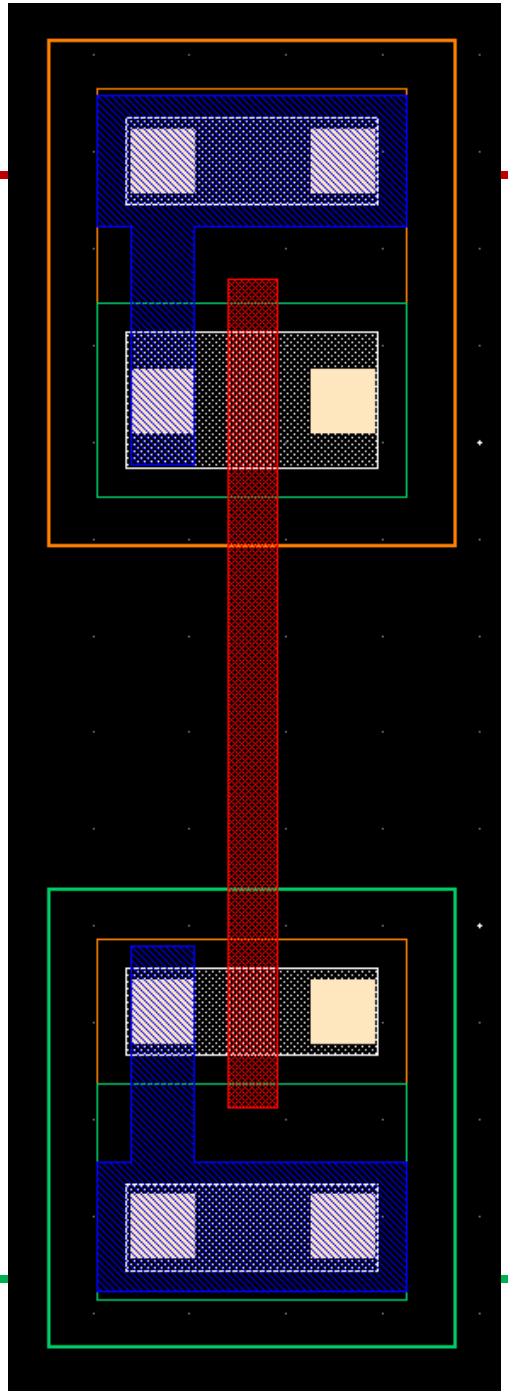
Example – Inverter

4. Body contacts



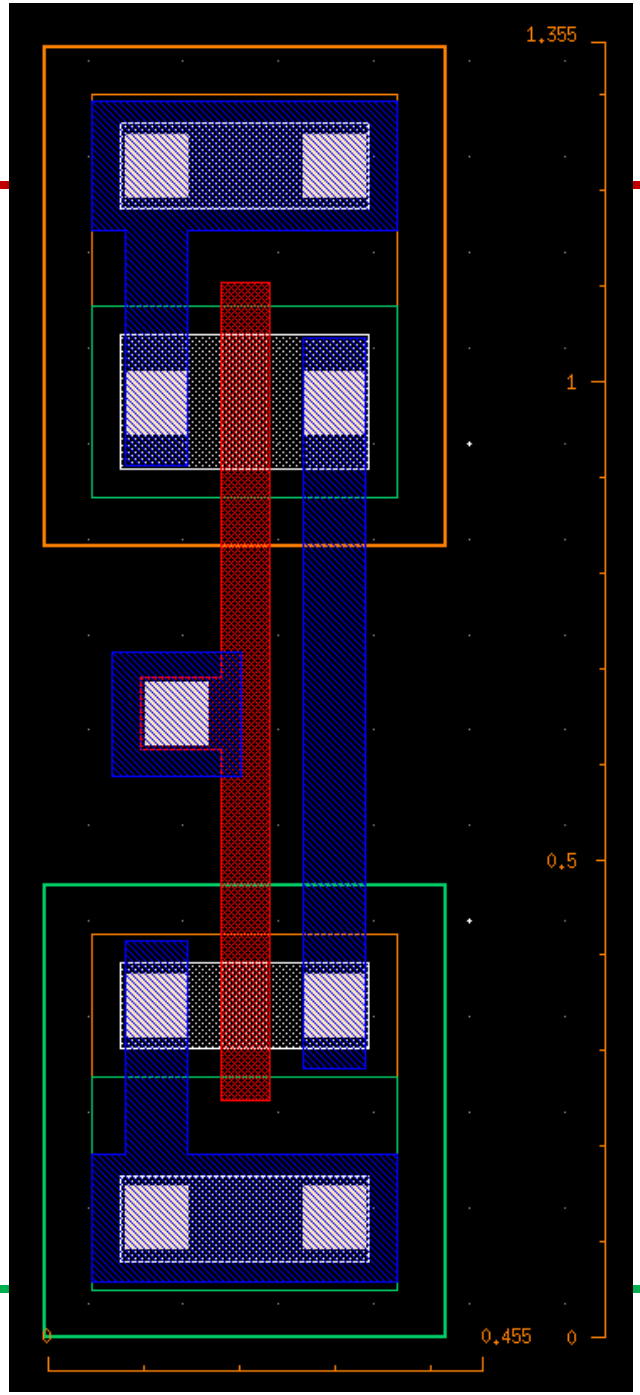
Example – Inverter

5. VDD and VSS (GND)
   - Metal 1 wires



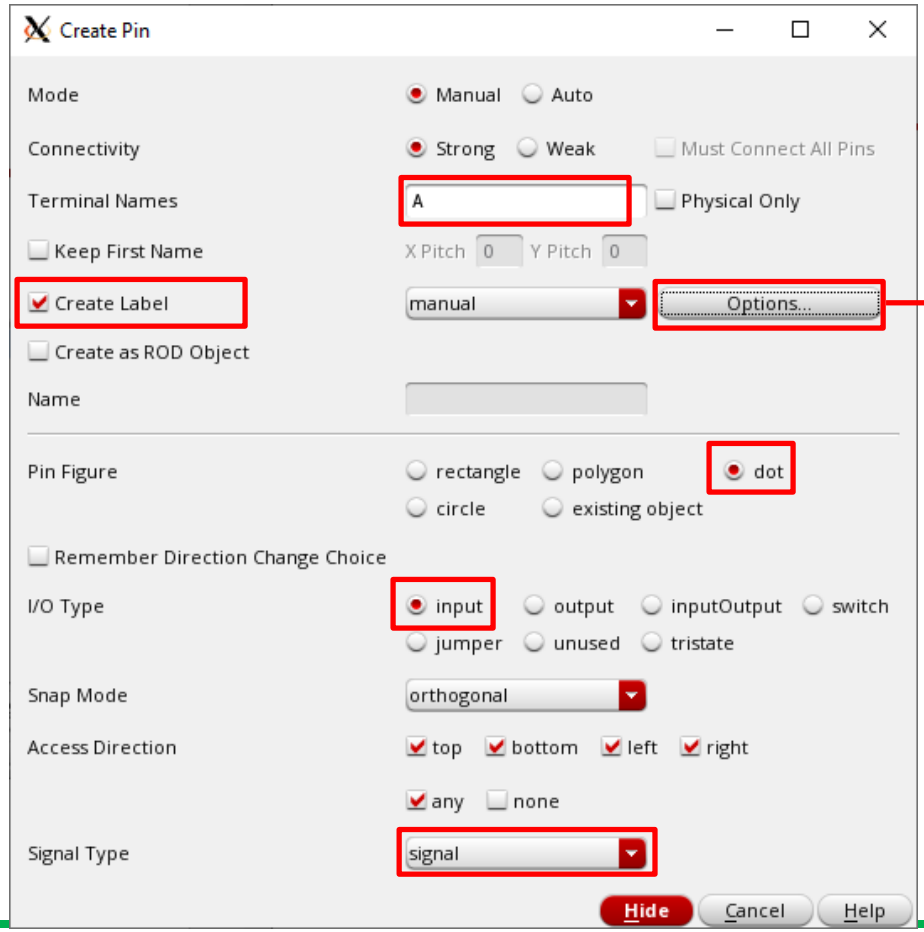
Example – Inverter

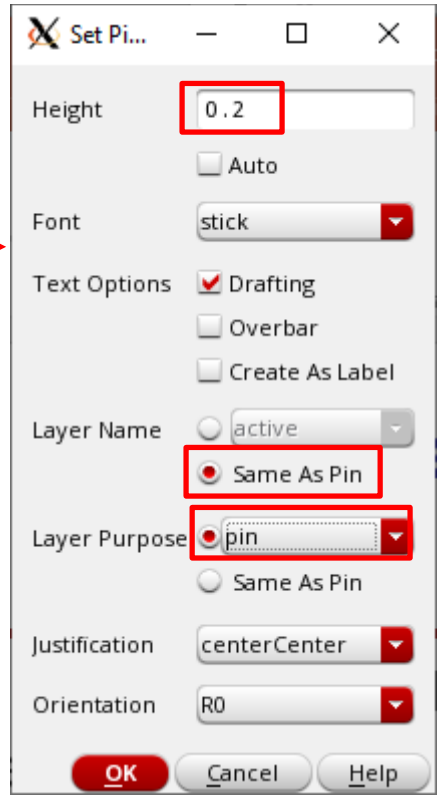
6. Input and output ports
   - Metal 1 wires



Example – Inverter

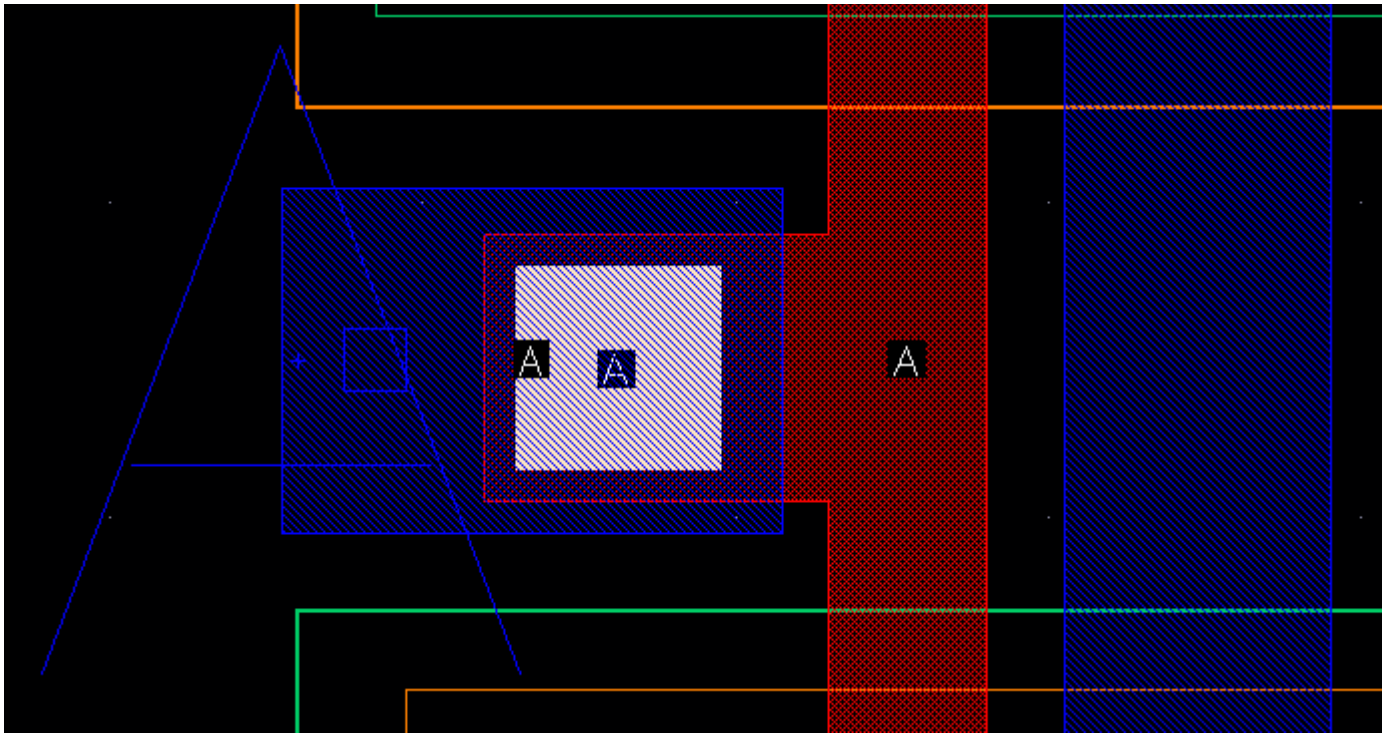
7. Create pins (A, ZN, VDD, VSS)
   - “Create” → “Pin...” →





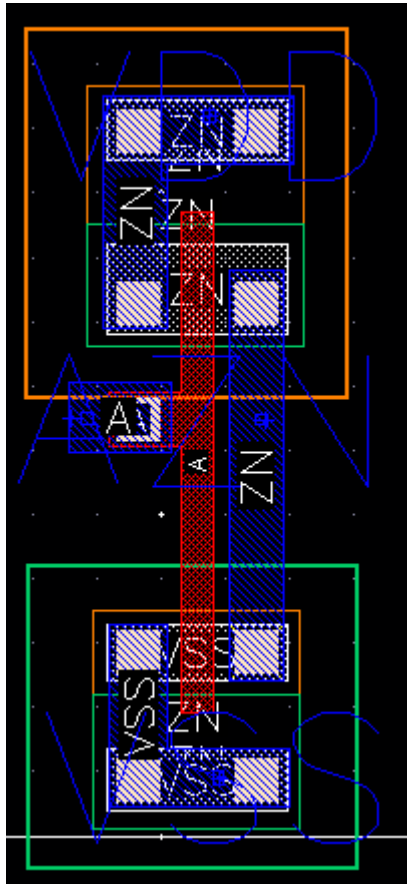
Example – Inverter

7. Create pins (A, ZN, VDD, VSS)
   - Then, create a small rectangle inside the target pin.
   - Make sure that the + mark of the pin is placed inside the wire object.



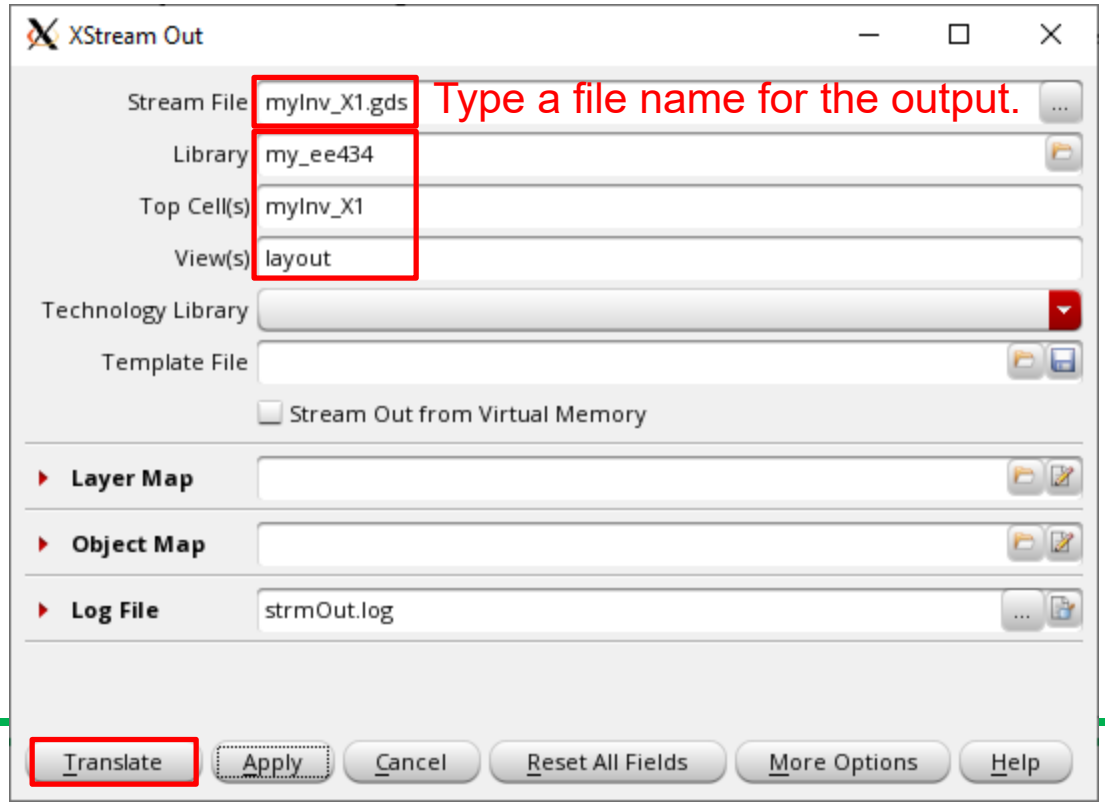
Example – Inverter

7. Create pins (A, ZN, VDD, VSS)
   - A
     - I/O type: input
     - Signal type: signal
   - ZN
     - I/O type: output
     - Signal type: signal
   - VDD
     - I/O type: inputOutput
     - Signal type: power
   - VSS
     - I/O type: inputOutput
     - Signal type: ground



Example – Inverter

8. Save the design.
9. Now, we should run DRC, LVS, and PEX. For that, we should export the layout into a GDSII file.
   - In CIW, Click “File” → “Export” → “Stream...”.
   - Click “Translate”.



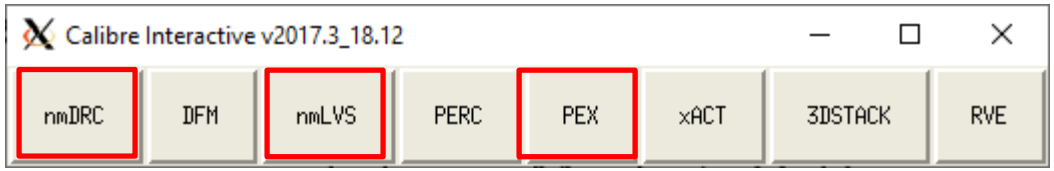
Example – Inverter

- If you export several times, it will ask you whether you want to overwrite the file. Select “Yes” and you might see this window. You can ignore the warning.

Example – Inverter

- Don’t close your layout window because you might (will!) have to fix some errors in your layout.
- Open another XWindow session.
- Source the following files.
  - source ictools_generic.sh
  - source mentor_calibre17.sh

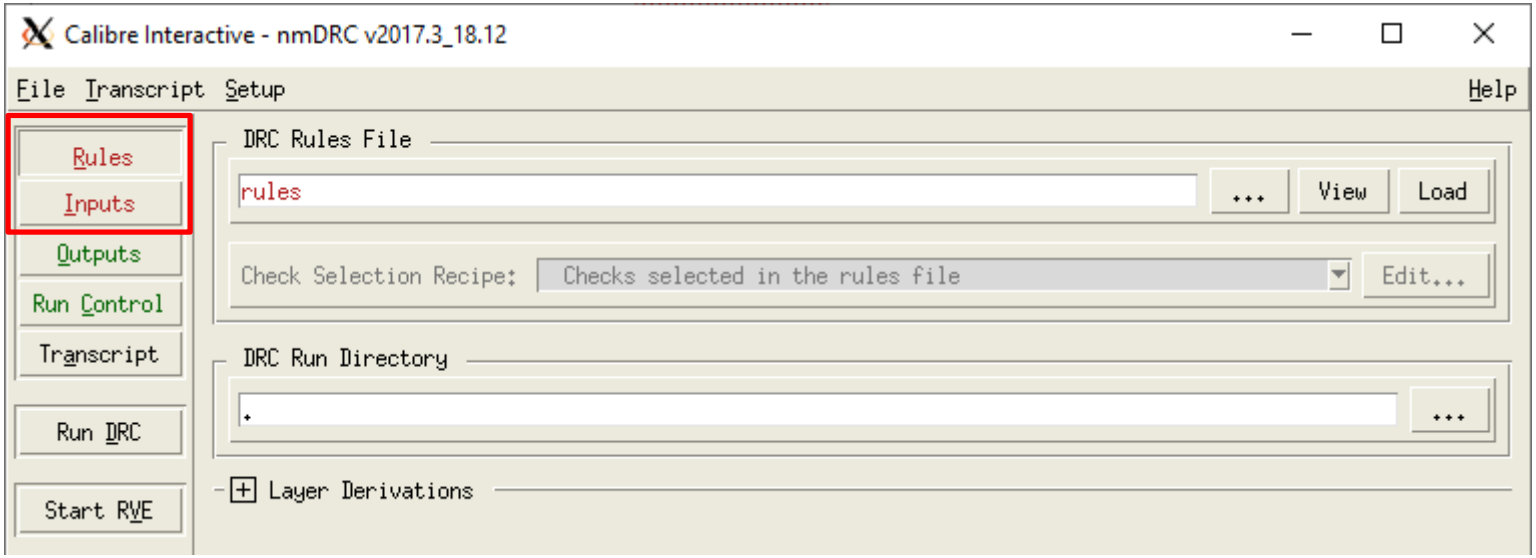
- Now you are ready to run Mentor’s Calibre.
  - calibre -gui



for DRC for LVS for PEX

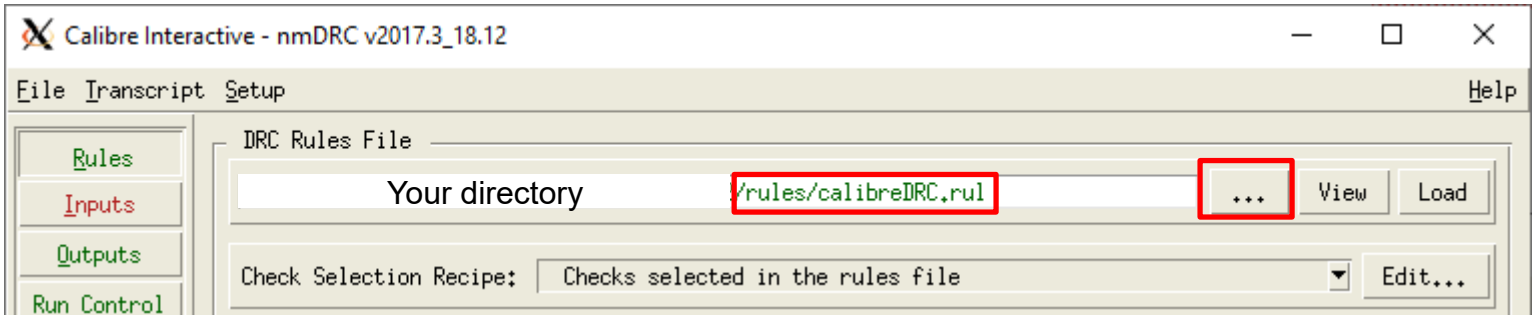
Example – Inverter

- Let's run DRC. Click "nmDRC".
- Close the "Load Runset File" window.
- The red texts mean that some inputs are missing.

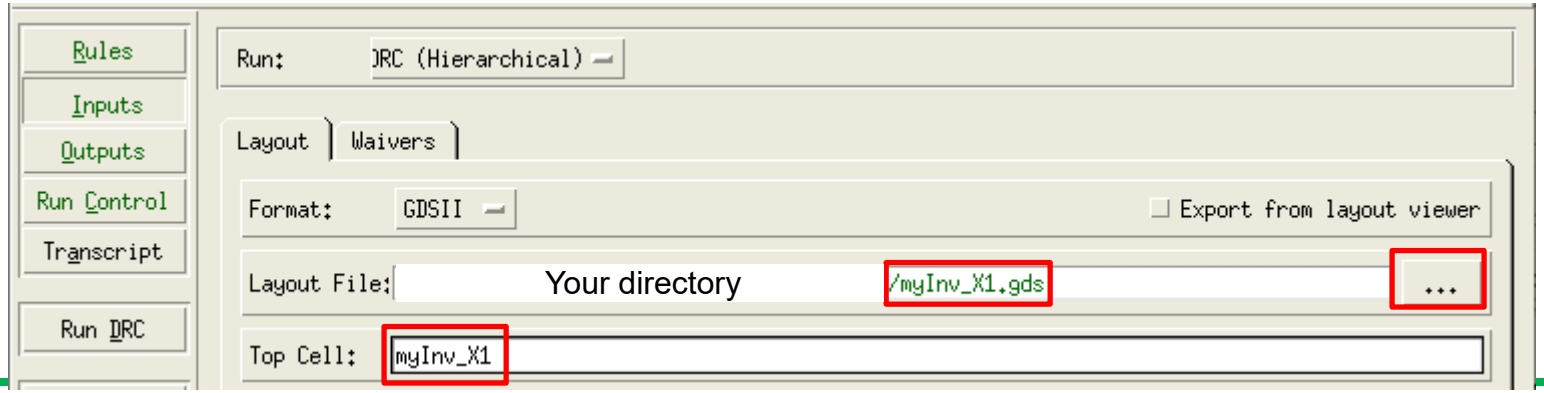


Example – Inverter

- Click “Rules” and “...” in “DRC Rules File” and select rules/calibreDRC.rul.

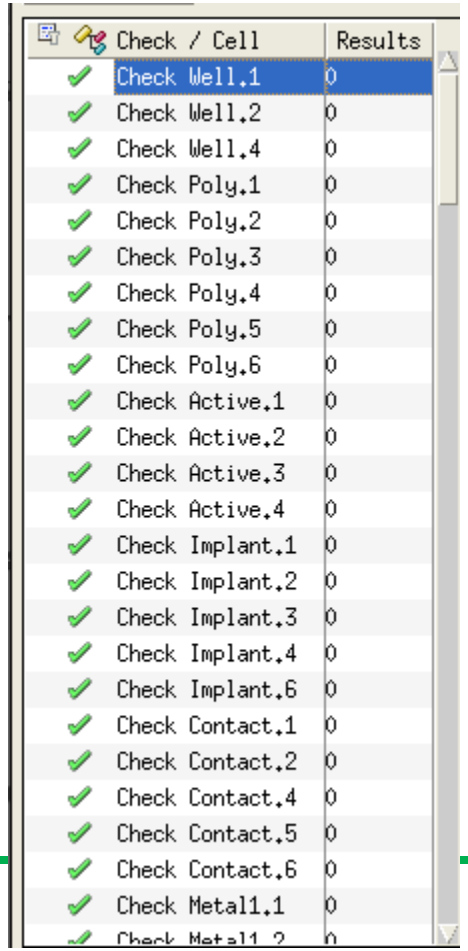


- Click “Inputs” and “...” in the “Layout File: ” and select the GDSII file.
- Enter the name of your cell.



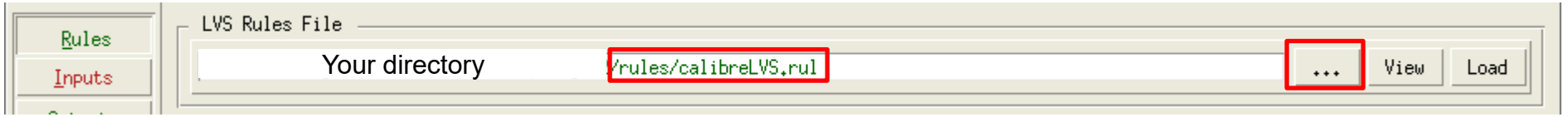
Example – Inverter

- Let's save it in a runset so that you don't need to re-type this again.
  - "File" → "Save Runset As" and just save it in a file.
  - Next time you open nmDRC, you can load the runset.
- Click "Run DRC".
- This will open two windows.
  - DRC Summary Report (just close it).
  - Calibre – RVE
    - You can see some errors. You should fix all of them.
    - Fix them in the layout window, export it, and rerun DRC.

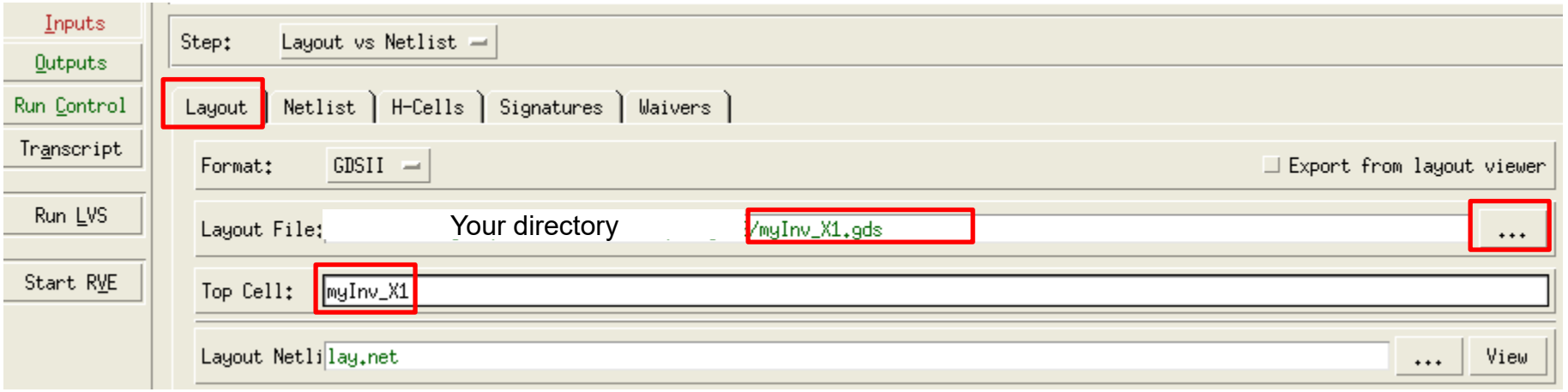


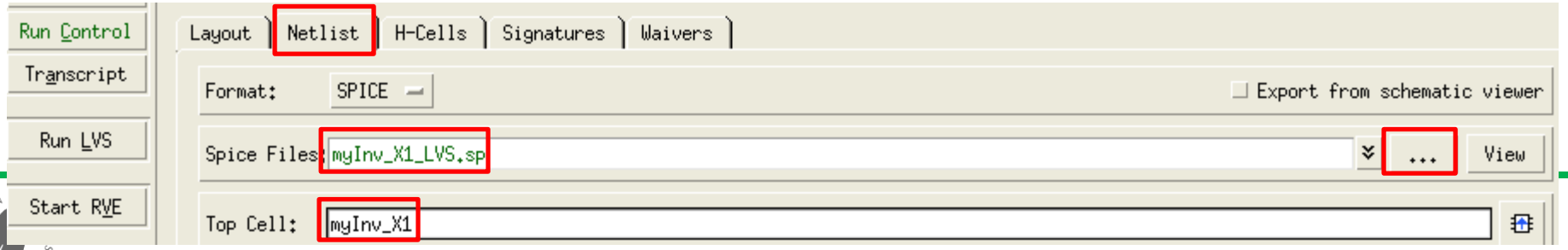
Example – Inverter

- Let's run LVS. Click "nmLVS".
- LVS rule: rules/calibreLVS.rul



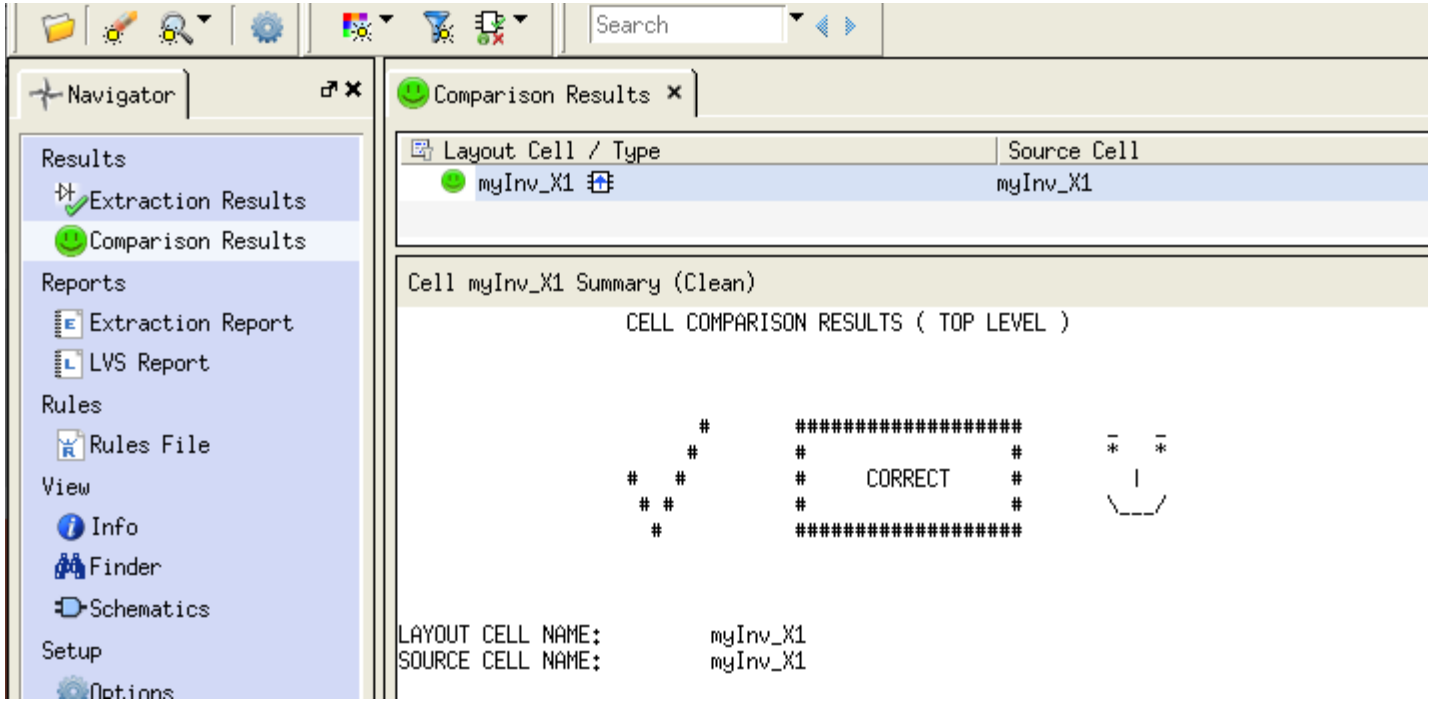
- Inputs





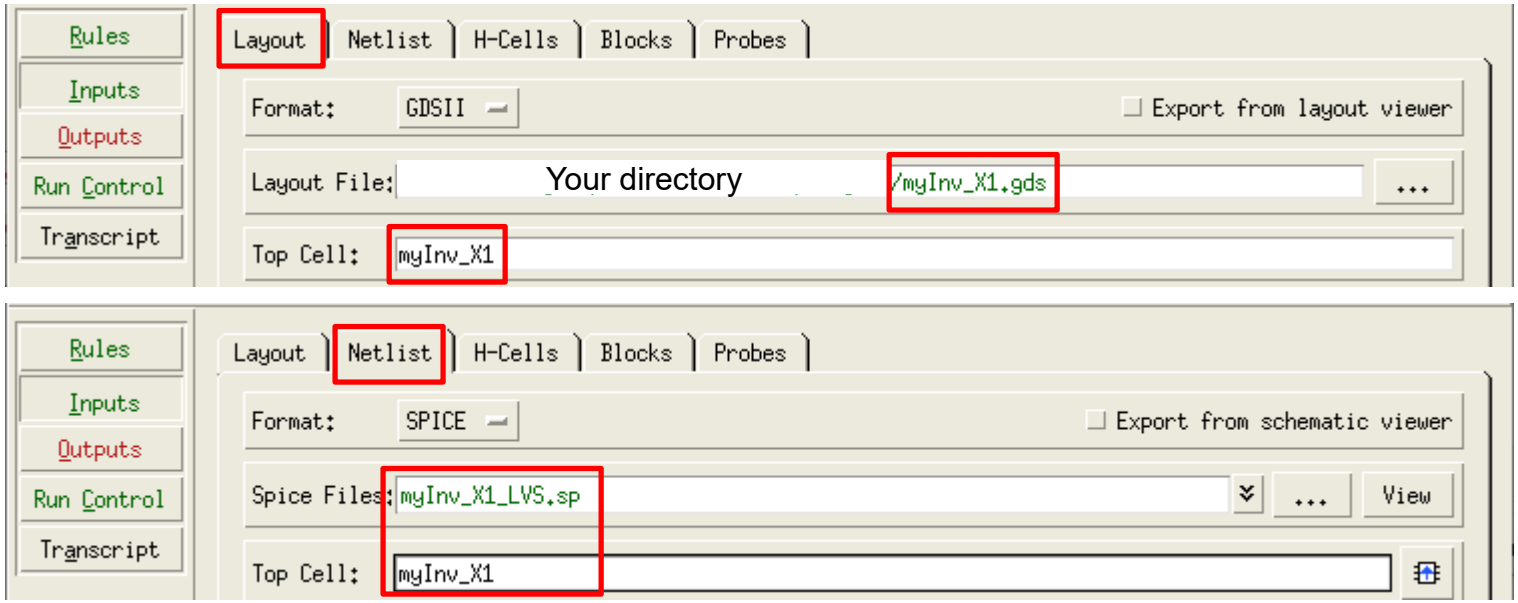
Example – Inverter

- You can save the setting in a runset file.
- Click “Run LVS”.
- Close the “LVS Report File” window.
- See the Calibre – RVE window. You should see a smile.

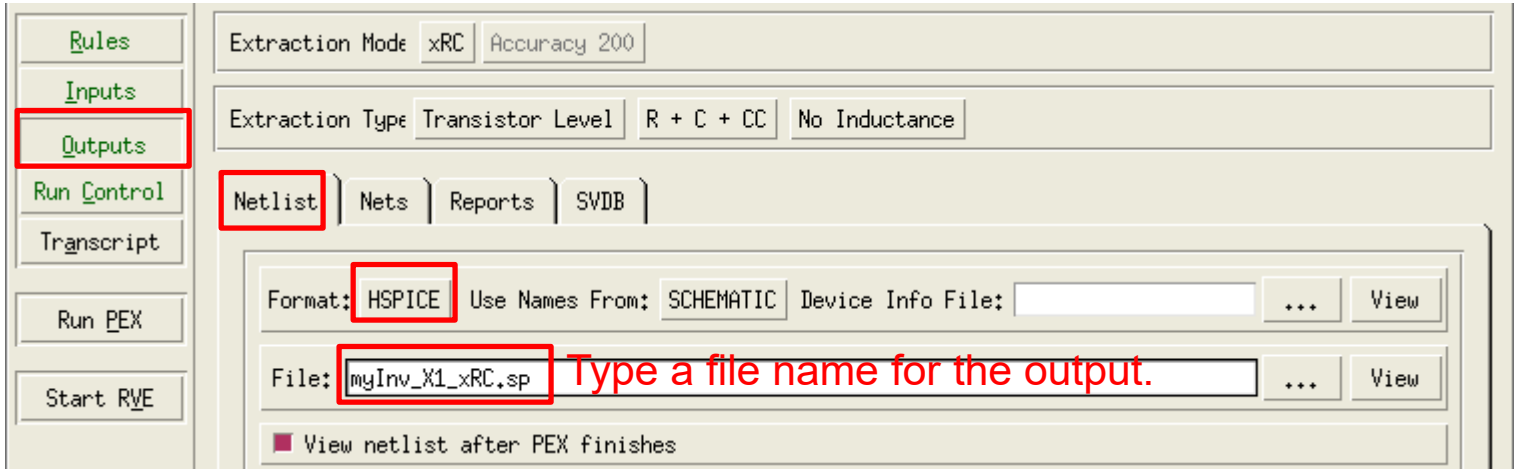


Example – Inverter

- Let's run PEX.
- Click "PEX" in the Calibre window.
  - Use rules/calibrexRC.rul for the rule file.



Example – Inverter



- Click “Run PEX”.
- Close the “PEX Netlist File” window.
- Now, you are done.

Example – Inverter

- Post-layout simulation
- The output of PEX is a new netlist with parasitic RC.
  - myInv_X1_xRC.sp is the top-level netlist.
    - myInv_X1_xRC.sp.pex has parasitic RC.

```
* File: myInv_X1_xRC.sp
* Created: Mon Mar  8 20:52:41 2021
* Program "Calibre xRC"
* Version "v2017.3_18.12"
*
.include "myInv_X1_xRC.sp.pex"
.subckt myInv_X1  A VSS VDD ZN
*
* ZN  ZN
* VDD VDD
* VSS VSS
* A A
mn1 N_ZN_mn1_d N_A_mn1_g N_VSS_mn1_s N_VSS_mn1_b NMOS_HP L=5e-08 W=9e-08
+ AD=1.035e-14 AS=1.035e-14 PD=4.1e-07 PS=4.1e-07
mp1 N_ZN_mp1_d N_A_mp1_g N_VDD_mp1_s N_VDD_mp1_b PMOS_HP L=5e-08 W=1.4e-07
+ AD=1.61e-14 AS=1.61e-14 PD=5.1e-07 PS=5.1e-07
*
.include "myInv_X1_xRC.sp.MYINV_X1.pxi"
*
.ends
*
```

This is a sub-circuit definition. You need to instantiate it to use it.
See the definition and the port names.

Example – Inverter

- Let's run HSPICE for the inverter.
  - hspice myInv_X1_pre.sp
  - hspice myInv_X1_post.sp

- The following table shows my result:

	Fall	Rise
Without RC	112.65ps	123.05ps
With RC	119.22ps	132.13ps
Difference	+6.57ps	+9.08ps

Troubleshooting

- For LVS, this is a general order of troubleshooting.
- First of all, match “Ports”, which are the primary input and output ports.
- Layout is from the GDSII file.
- Source is from the HSpice netlist, which is the golden for LVS.
- Thus, if there is any mismatch, it means your layout has problems.

```
LAYOUT CELL NAME:        myInv_X1
SOURCE CELL NAME:        myInv_X1

-------------------------------------------------------

INITIAL NUMBERS OF OBJECTS
--------------------------

                  Layout    Source        Component Type
                  ------    ------        --------------
 Ports:              4         4
 Nets:               4         4
 Instances:          1         1          MN (4 pins)
                     1         1          MP (4 pins)
                   ------    ------
 Total Inst:         2         2
```

Troubleshooting

- If # ports in your layout is 0, it is usually due to the port setting.

```
LAYOUT CELL NAME:         myInv_X1
SOURCE CELL NAME:         myInv_X1

------------------------------------------------------

INITIAL NUMBERS OF OBJECTS
--------------------------

                  Layout    Source        Component Type
                  ------    ------        --------------
 Ports:              4         4

 Nets:               4         4

 Instances:          1         1         MN (4 pins)
                     1         1         MP (4 pins)
                  ------    ------
 Total Inst:         2         2
```

- If the ports match, then, try to match the “Instances”.

```
LAYOUT CELL NAME:         myInv_X1
SOURCE CELL NAME:         myInv_X1

------------------------------------------------------

INITIAL NUMBERS OF OBJECTS
--------------------------

                  Layout    Source        Component Type
                  ------    ------        --------------
 Ports:              4         4

 Nets:               4         4

 Instances:          1         1         MN (4 pins)
                     1         1         MP (4 pins)
                  ------    ------
 Total Inst:         2         2
```

Troubleshooting

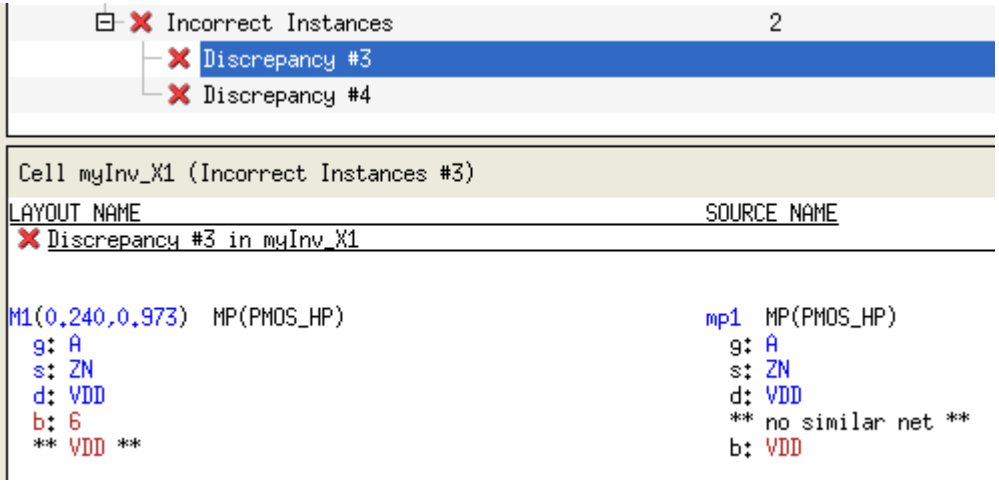
- If # instances (transistors) mismatch, check the transistors in your layout.

- If # instances match, try to match “Nets”.

```
LAYOUT CELL NAME:        myInv_X1
SOURCE CELL NAME:        myInv_X1

-------------------------------------------------------

INITIAL NUMBERS OF OBJECTS
--------------------------

                Layout    Source         Component Type
                ------    ------         --------------
 Ports:            4         4
 Nets:             4         4
 Instances:        1         1          MN (4 pins)
                   1         1          MP (4 pins)
                ------    ------
 Total Inst:       2         2
```

Troubleshooting

- If there are still some errors, click “Discrepancy #”



- For example, the above error says that the body of the transistor M1 in the layout is connected to net “6”, but that in the source (HSpice netlist) is connected to VDD. Since the source is always correct, the body of M1 in your layout should be connected to VDD.

Troubleshooting

- For subckts in HSpice,
  - Open myInv_X1_LVS.sp and see the definition.
  - Also, open myInv_X1_pre.sp and see how I use it.

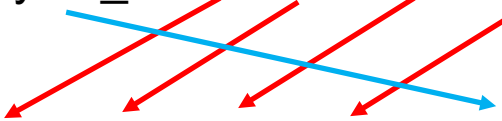
- When you run PEX, Calibre generates a new HSpice netlist with parasitic RC.
  - This netlist also has a subckt definition for your cell.
  - However, the order of the ports in the definition might different from that in myInv_X1_LVS.sp.
  - Thus, when you do post-layout simulation, you should make sure that your cell instantiation follows the order of the ports in the PEX output file.

Troubleshooting

- For example, for pre-layout simulation

```
.SUBCKT  myInv_X1  A  ZN  VDD  VSS
```

(definition in myInv_X1_LVS.sp)
This is made by you.



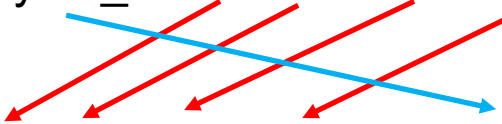
```
X1  nIn  nOut  nVdd  0  myInv_X1
```

(instantiation in myInv_X1_pre.sp)

- For post-layout simulation

```
.SUBCKT  myInv_X1  A  VSS  VDD  ZN
```

(definition in myInv_X1_xRC.sp)
This is generated by PEX.



```
X1  nIn  0  nVdd  nOut  myInv_X1
```

(instantiation in myInv_X1_post.sp)