Prepare Peakview Devices for LVS Check for GlobalFoundries 22nm PDK

LORENTZ SOLUTION

April. 2019

LD 14765

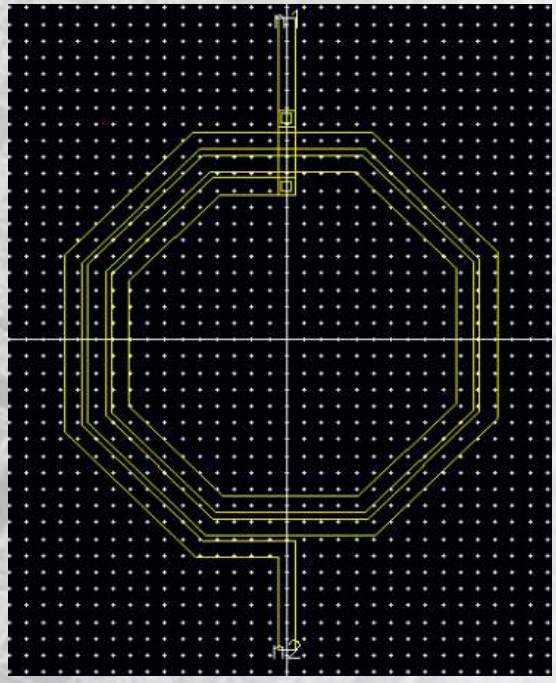
Background

- **Objective: solve the LVS issue for circuits with customized/PDK devices that are modeled by Peakview.**
- **Common LVS concerns with customized devices**
  - Original PDK rules may not be able to recognize customized devices.
  - If circuits with customized devices fail to pass LVS, post layout simulation would be impossible.
- **Prepare Peakview devices for LVS**
  - LVS box (black box) method
- **This method is only verified in this environment**
  - Device must be generated by Peakview
  - Only applies to Mentor Calibre
  - Tested on GlobalFoundaries 22nm

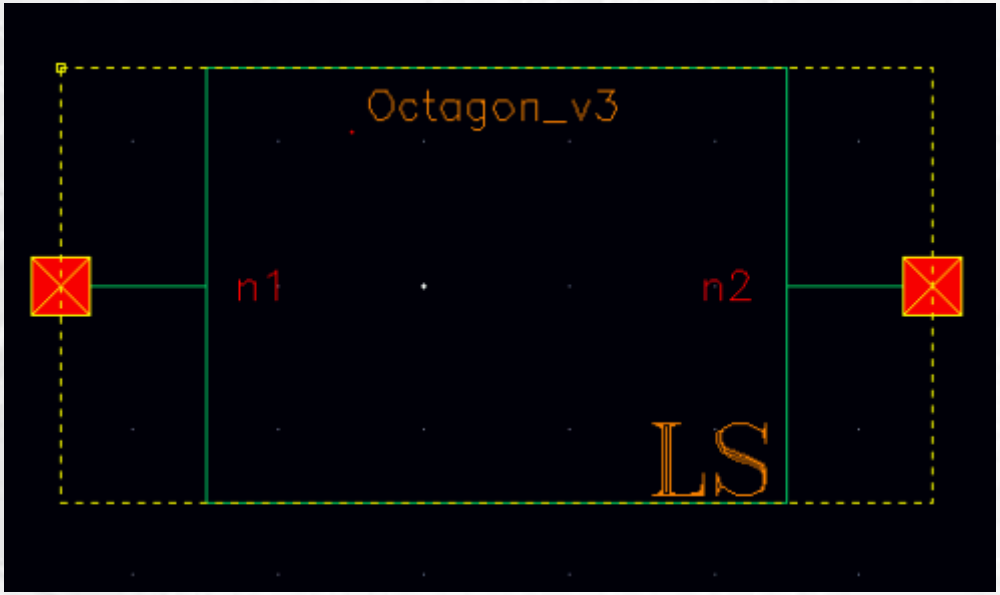
Basic Steps to Pass LVS

- **Generate the Peakview device cell**
  - Launch Peakview EM synthesis/LEM and simulate devices, then sync with Cadence.
- **Prepare devices for the LVS purpose**
  - Layout modification for LVS
  - Simple LVS rule file preparation
  - We mainly show Black Box method LVSTest LVS on Peakview device
  - Testbench preparation
    - Create the schematic and layout views in test bench cell containing Peakview device instance
  - Generate the CDL/netlist file
  - Run the LVS

Step 1. Generate the Peakview Device Cell



Customized device layout



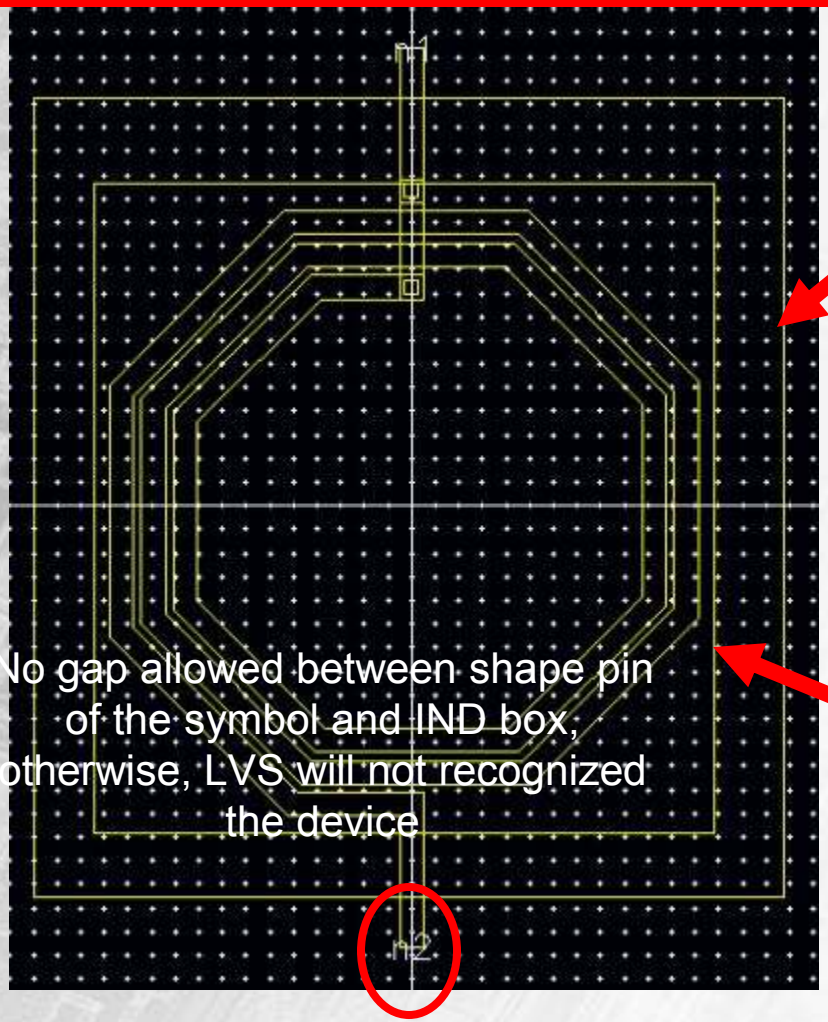
Customized device layout

Launch Peakview EM synthesis and simulate divices, then sync with Cadence. Corresponding cells would appear in the library(in this case, Octagon_v3), containing the device layout and symbol generated by PeakView

Step 2: Prepare for LVS

- **Only LVS/PEX Box method**
  - Layout modification for LVS
  - Simple LVS rule file preparation
- **LVS/PEX Box method**
  - The device layout is from PeakView or designer and the EM model is generated by PeakView.
  - LVS box device.
- **Difference**
  - RC extraction do not touch inside of the boxed cells, its netlist is from schematic instead of LPE.
  - We recommend the box method for its simplicity.

Step 2a. Layout Modification for LVS

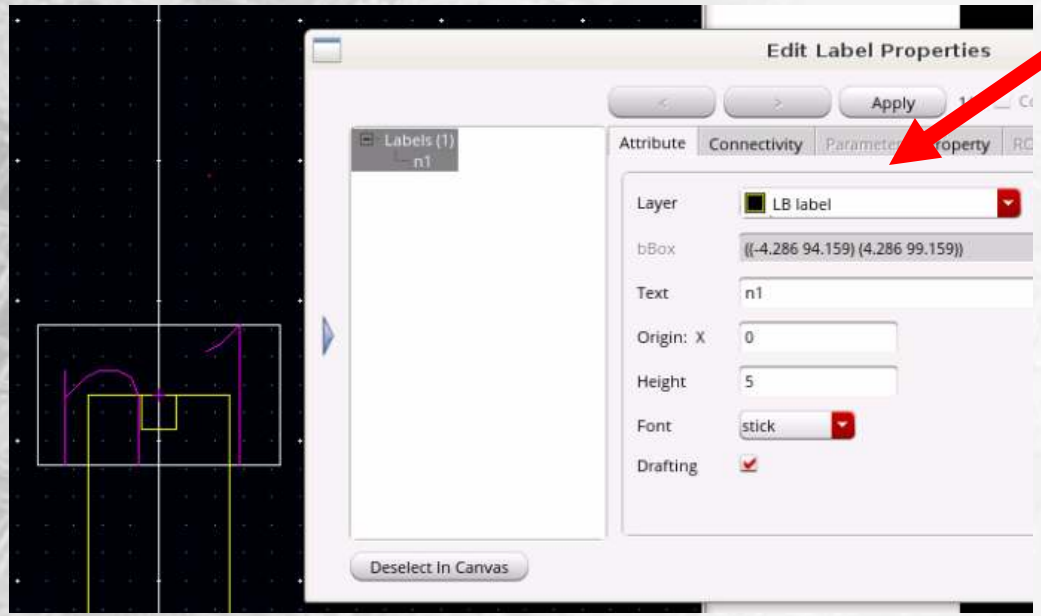


In the layout of a Peakview device cell, define an inductor with IND (drw) layer.
The IND layer must not interact with other layers outside the shape, use polygon if necessary.

Within the IND (drw) layer, define the dimension of the inductor with LVS (dr5). The LVS layer must not interact with the inductor nor the IND layer.

Step 2b. Change the Purpose of Layers of the Pins

Change the purpose of layers of n1 and n2 from LB drw to LB label.



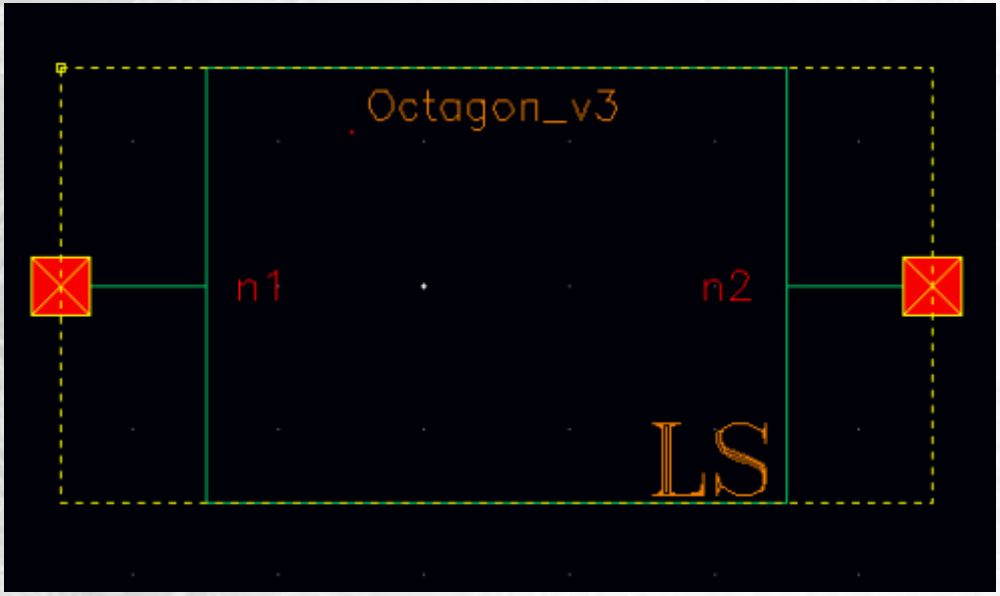
Step 2c. LVS/PEX BOX method: Create <mylvsrul> File

Create the <mylvsrul> file and write the follwing commands

```
LVS BOX Octagon1
LVS PRESERVE BOX CELLS YES
```

Add your cell by using the following format:
LVS BOX “CELL_NAME1”
LVS BOX “CELL_NAME2”
……
LVS PRESERVE BOX CELLS YES
Replace red fonts with your own design name

Step 3a. Create A testbench Schematic View



Customized device symbols

Create a schematic view in a test bench cell whose name differs from the cell from Peakview. Add an instance using the symbol from Peakview. Check and save.

Step 3b. LVS/PEX BOX method: Edit <.simrc> File

```
;Setting this variable to nil eliminates Path Display errors for
;schematics where there are terminals labelled differently than
the
;wires they connect to. However, this slows it down
significantly.
;vtiWireLabelToTermNameFlag = nil

; STL
; simDiffInputFile = "simdiff.stl"
auCdlPrintEmptySUBCKT = t
```

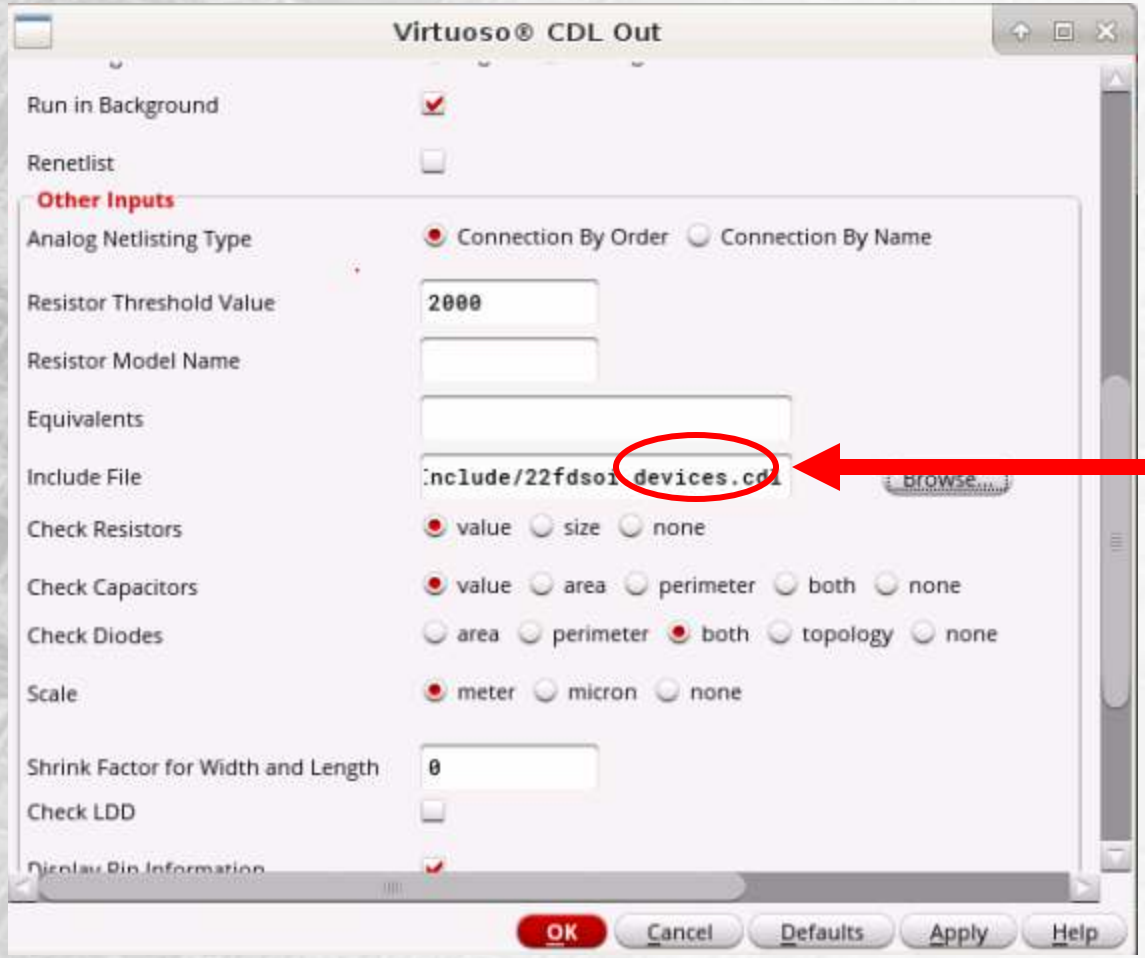
Edit <.simrc> file by adding one command line:

auCdlPrintEmptySUBCKT = t

##<.simrc> file is under your working directory. If you cannot find it, you can try to manually create one.

LORENTZ SOLUTION

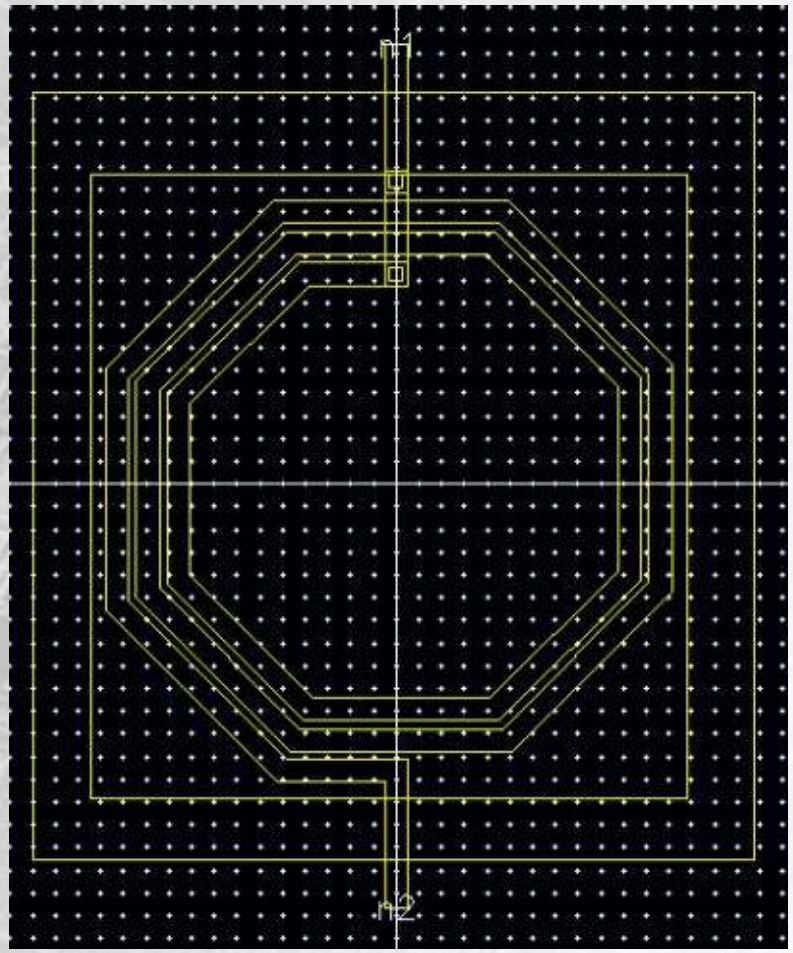
Step 3c. Generate CDL of Top Schematic



When export the CDL in Cadence, the proper Calibre include file should be included.

LORENTZ SOLUTION

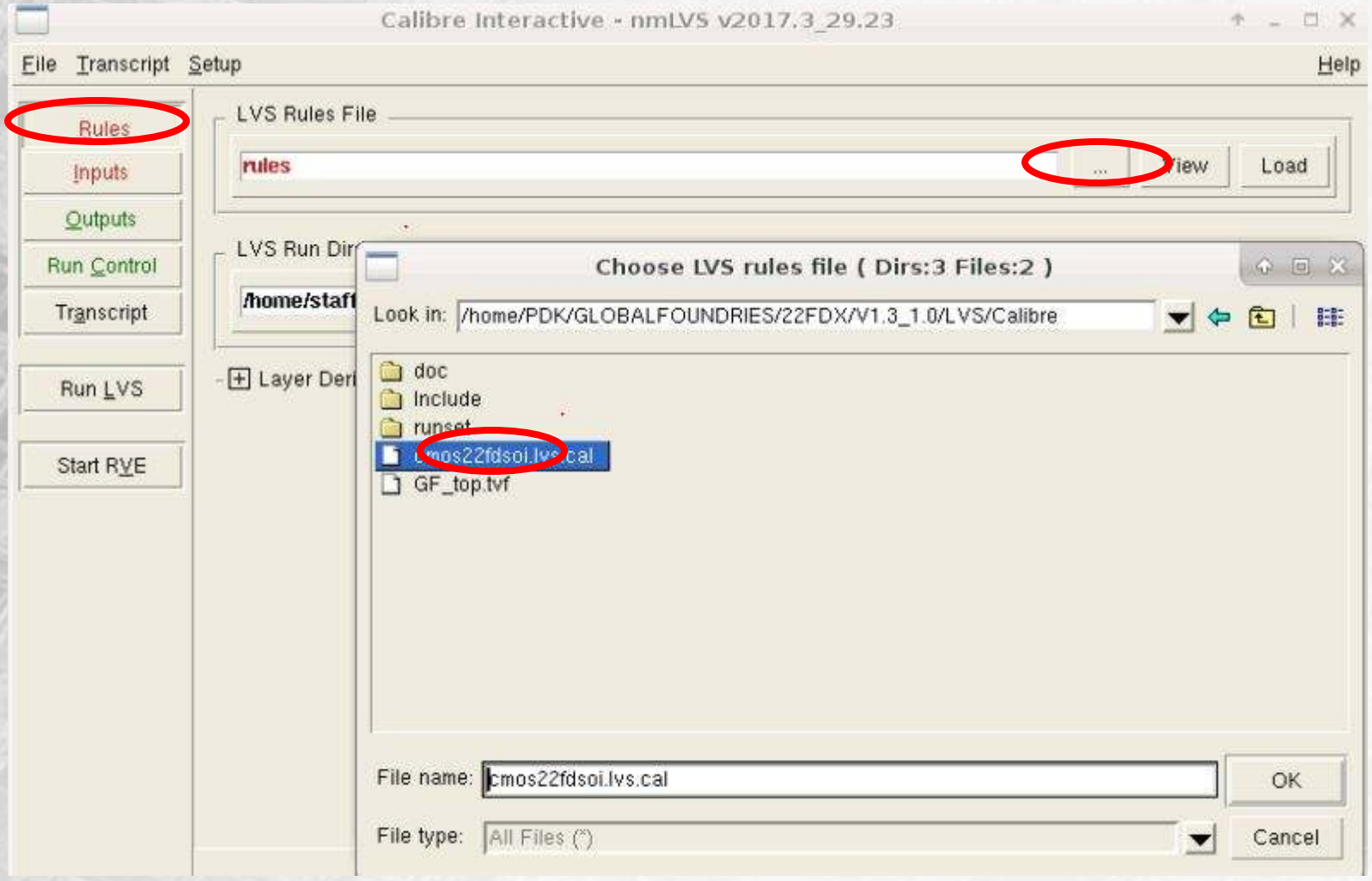
Step 3d. Create A New Layout View



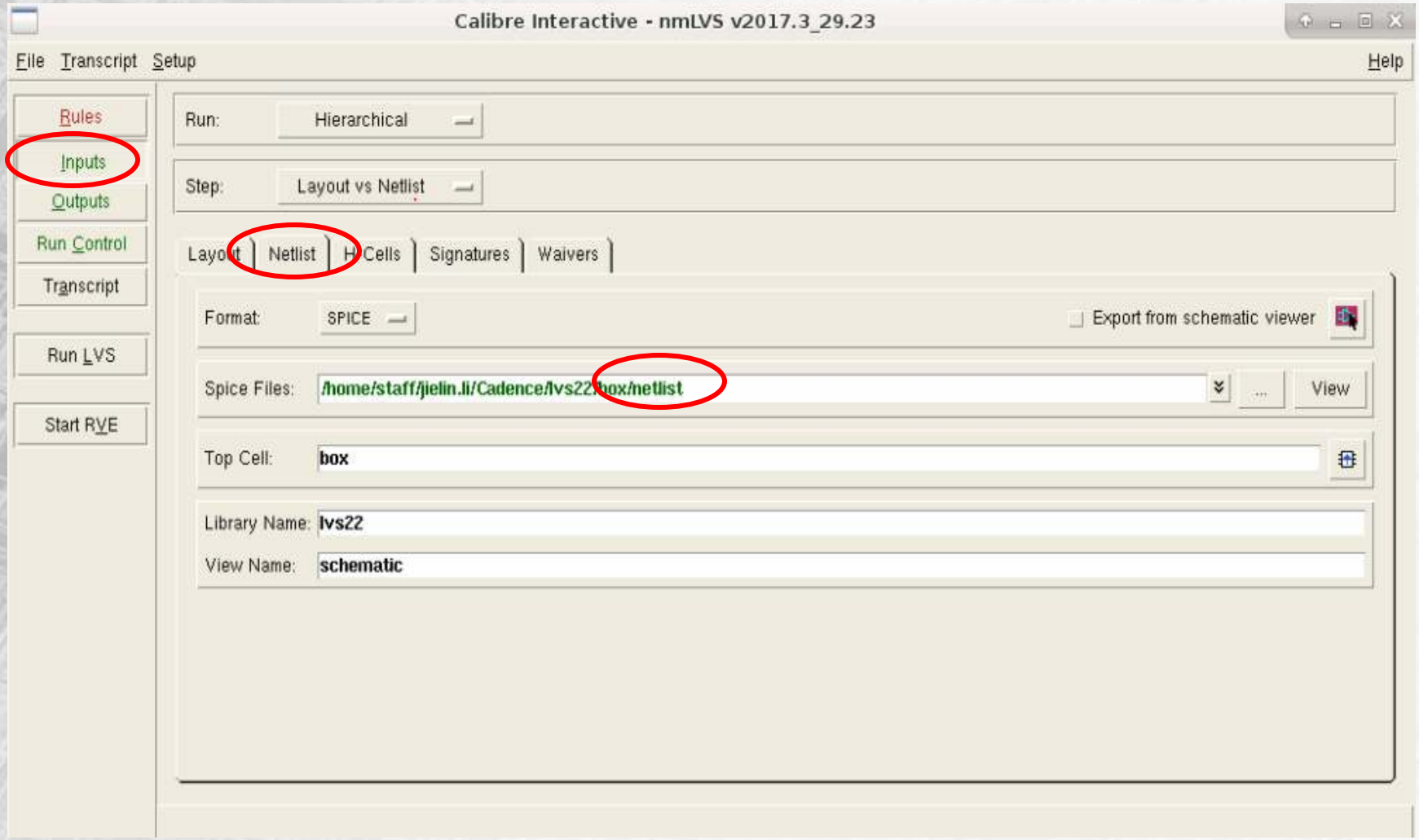
Customized devices

Create a layout view in the test bench cell. Add an instance using the device from Peakview. Save.

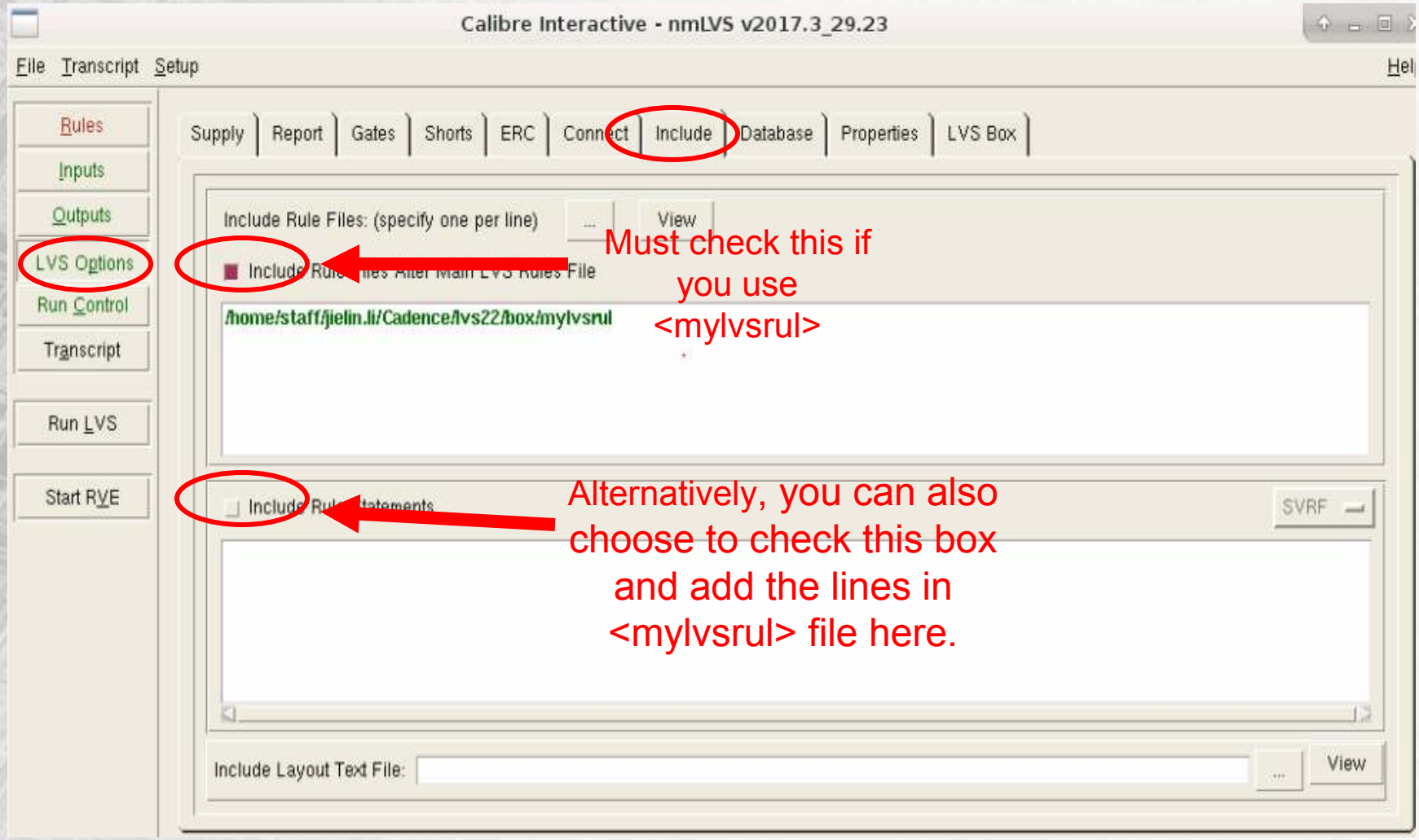
Step 4a: Run LVS including main rule file



Step 4b: Run LVS including netlist



Step 4c: Run LVS including mylvsrul



Finally: LVS Results