

CMP305

Very Large Scale Integrated Circuit

Lab 4

Place & Route (Part 2)

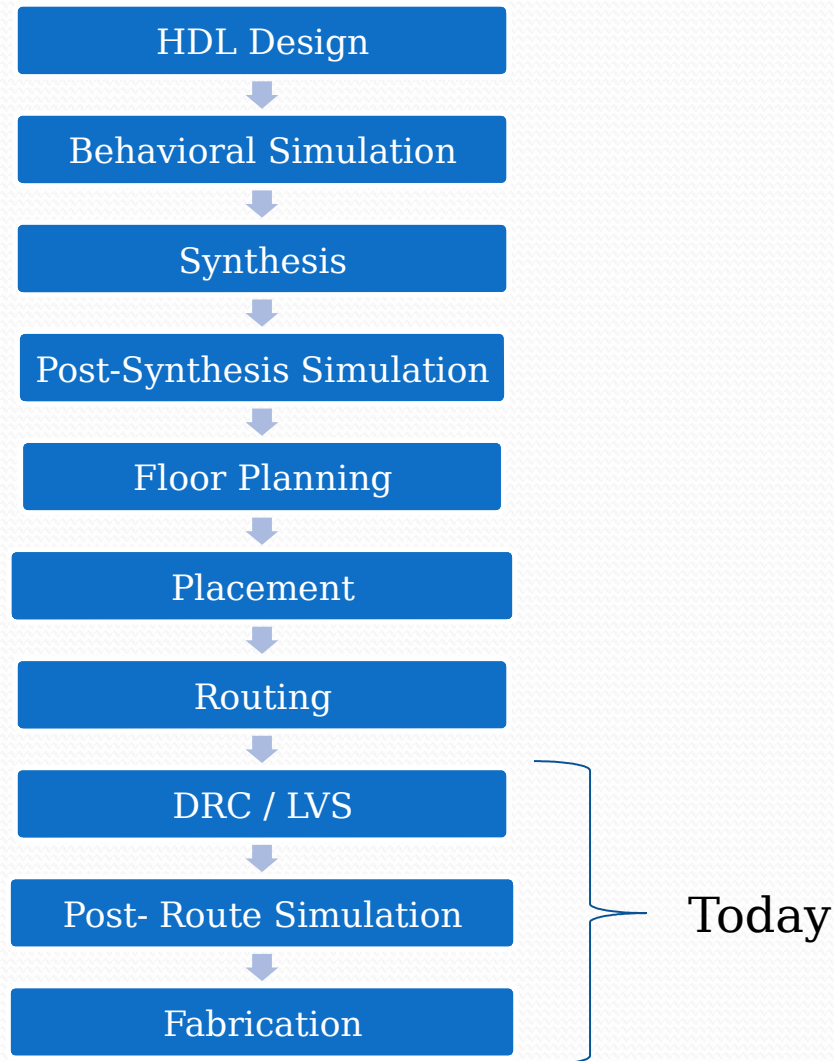
Last 3 labs

- We Discussed ASIC Design Flow
- We applied Synthesis Step
- We Learned about Timing Analysis
- We applied design constraints
- We applied Place & Route Step
- We used Tcl script instead of GUI
- We performed post-synthesis simulation

Objectives

- Understand:
 - DRC
 - LVS
- Learn:
 - Use Pyxis & Calibre
 - Post-Layout Simulation

ASIC Design Flow



Verification

- To make the layout ready for fabrication we need to:
 1. Check that all system requirements are met
 - Timing Analysis
 - Constraint Report
 2. Check that all the fab requirements are met
 - DRC
 - LVS
 3. Check that the functionality is still valid
 - Post-Layout Simulation

System Requirement

- As we explained last lab for the tools shortage
 - Pyxis doesn't provide a tool to generate timing analysis
 - So 3rd party tools are used
- For real world applications, we check:
 - Timing Constraints
 - Area Constraints
 - Power Consumption Constraints

Functionality Check

- We need to check that the functionality is still valid
 - After Synthesis
 - Simulate the netlist
 - After Place & Route
 - Extract the netlist from the Layout (Not Applicable in Pyxis)

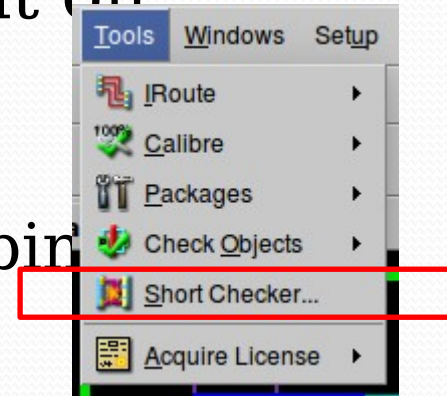
Fab Requirement

- The Fab announce their rules, you have to meet them
 - “TSMC03” we used is created by the fab
- In some cases, ASIC library is created by a company, & the Fab is owned by another company
 - An additional mapping step is needed (From ASIC Library to Fab library)
- We check for both:
 - DRC
 - LVS

Checking Short Circuit

- It checks for any short circuit on the diagram

- Wires in Pyxis are labeled according to the netlist mapping



- Select from "Tools"
 - Select "Short Checker"
 - Then click "OK"

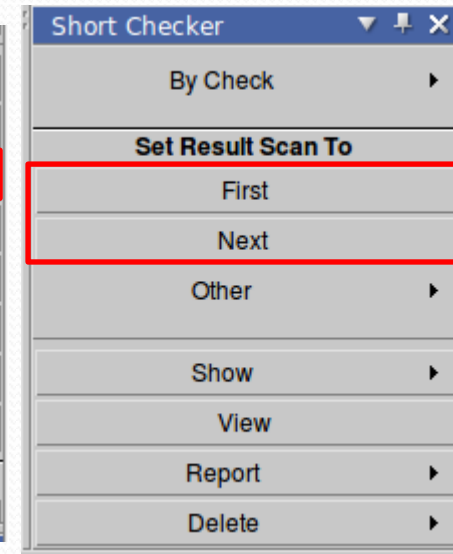
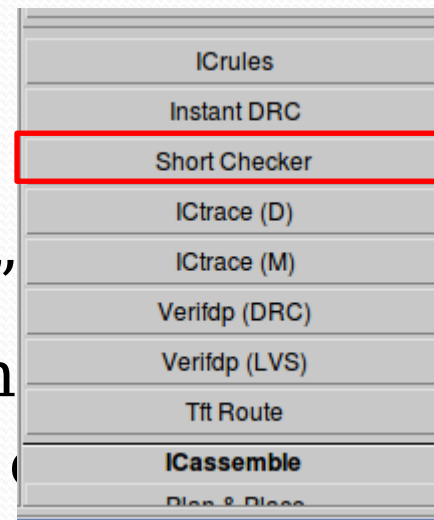


Checking Short Circuit

- After performing the check, a summary report appears in the Transcript

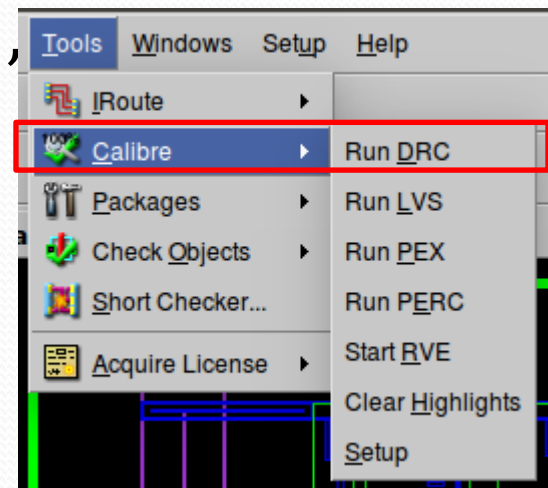
```
$check_shorts_all(@replace);  
// Note: License for ICshortchecker has been acquired. (from: Ic/ui/ic_license_mgr 81)  
// Note: Loadable Module "short_checker": 0 short(s) found. (from: Ic/modules/general 82)
```

- If any short is found we could see it from the Pyxis Palette
 - Select “Short Checker”
 - “First” shows the 1st short
 - “Next” shows the next short



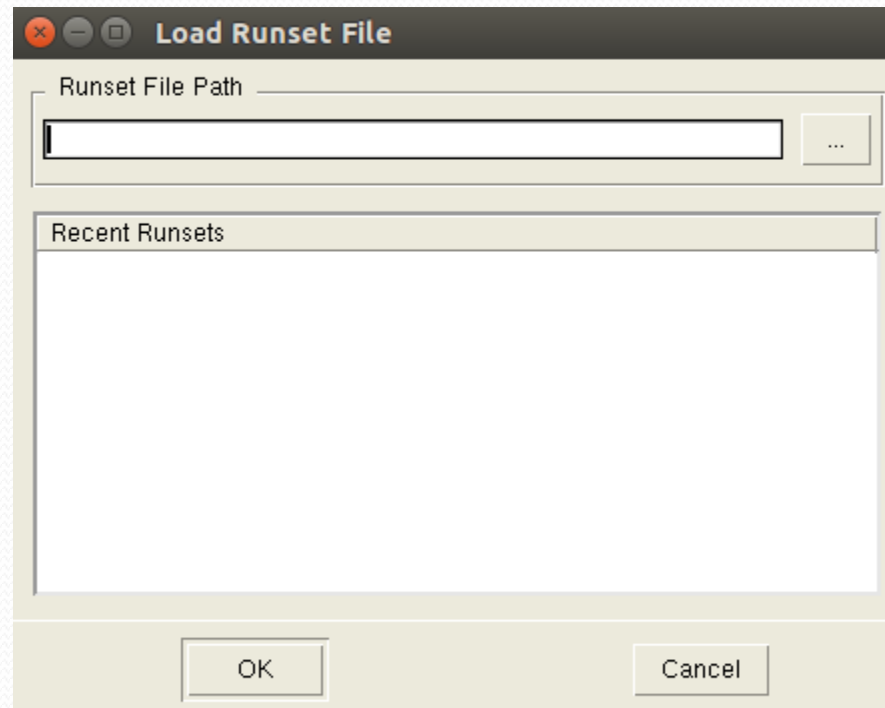
DRC

- We will run Calibre DRC from Pyxis
- From “Tools”, select “Calibre”
 - Then select “Run DRC”



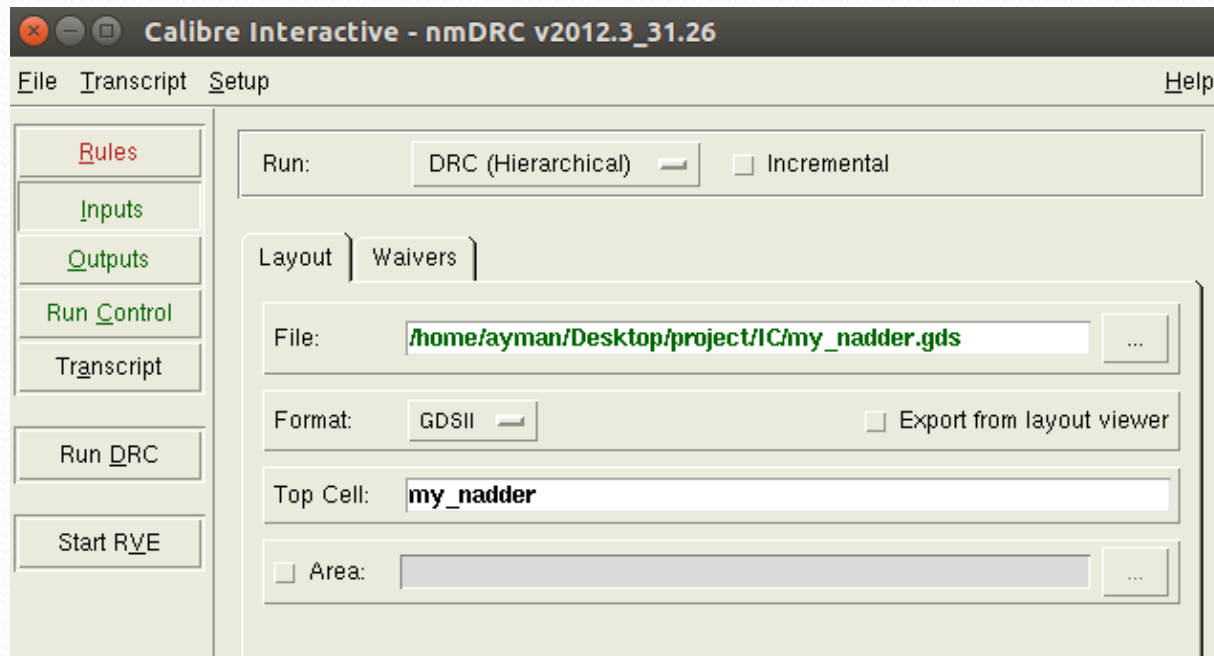
DRC

- Cancel “Load Runset File”



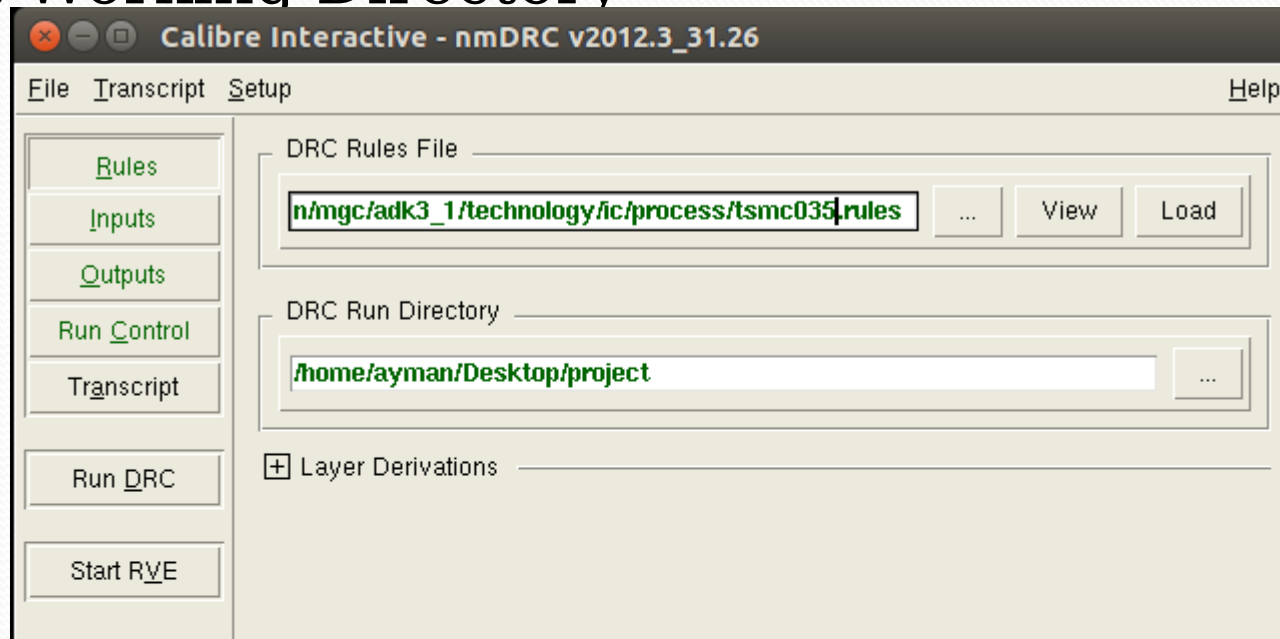
DRC

- In “Inputs” Tab:
 - Select DRC (Hierarchical)
 - Select the generated gds file
 - Unselect “Export from Layout Viewer”



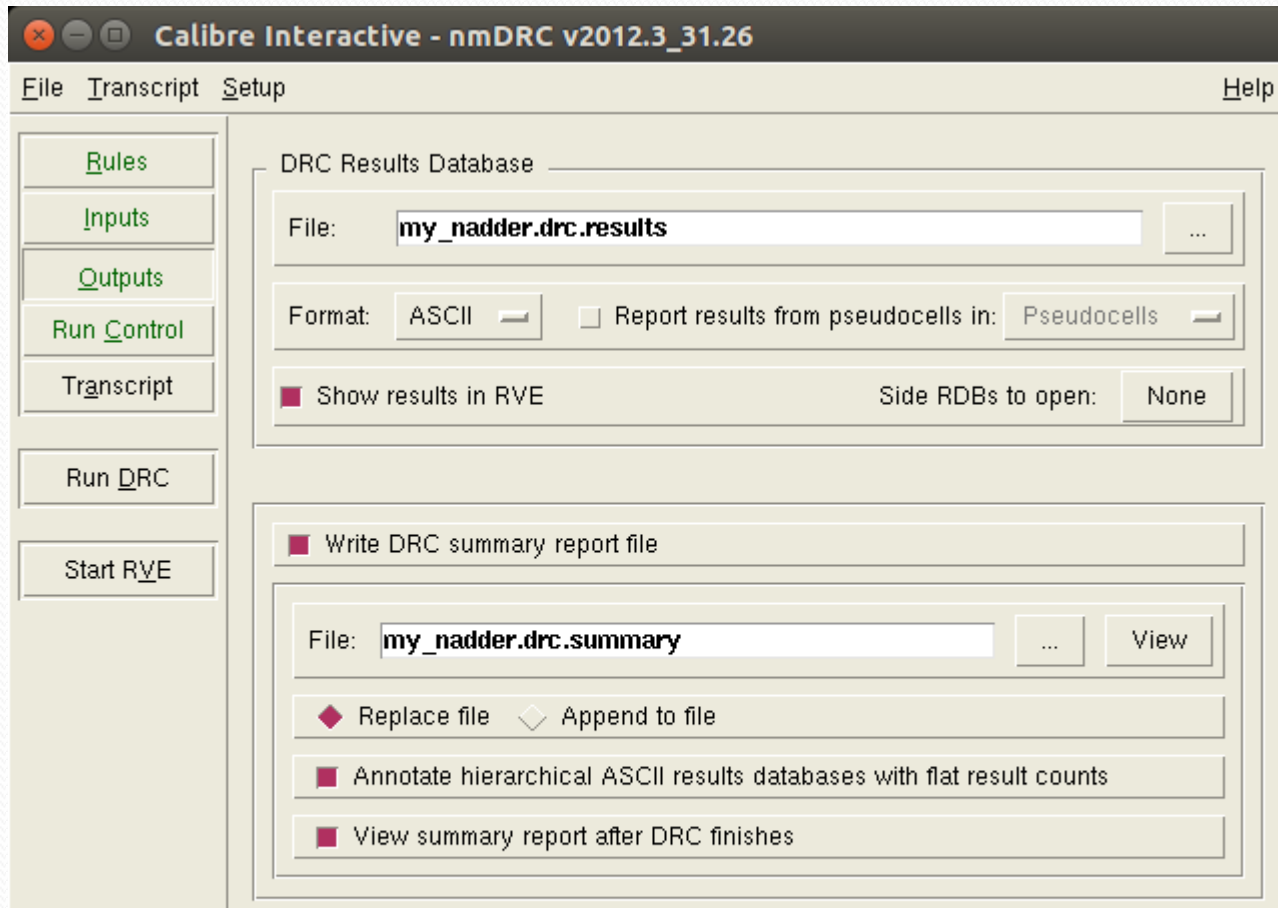
DRC

- In “Rules” Tab:
 - Set Rule file to “technology/ic/process/ami05.rules”
 - Set Working Directory



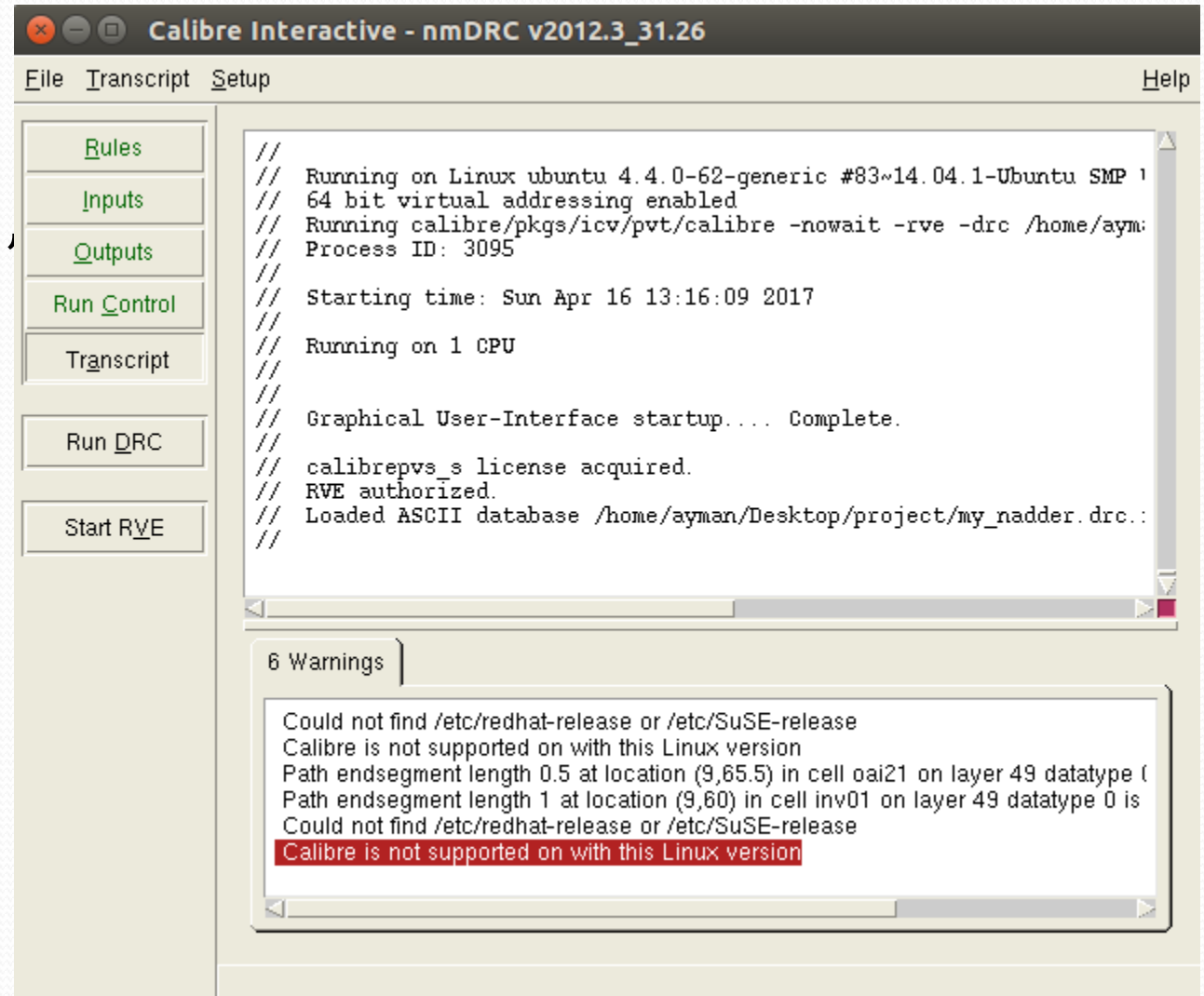
DRC

- In “Output” Tab:

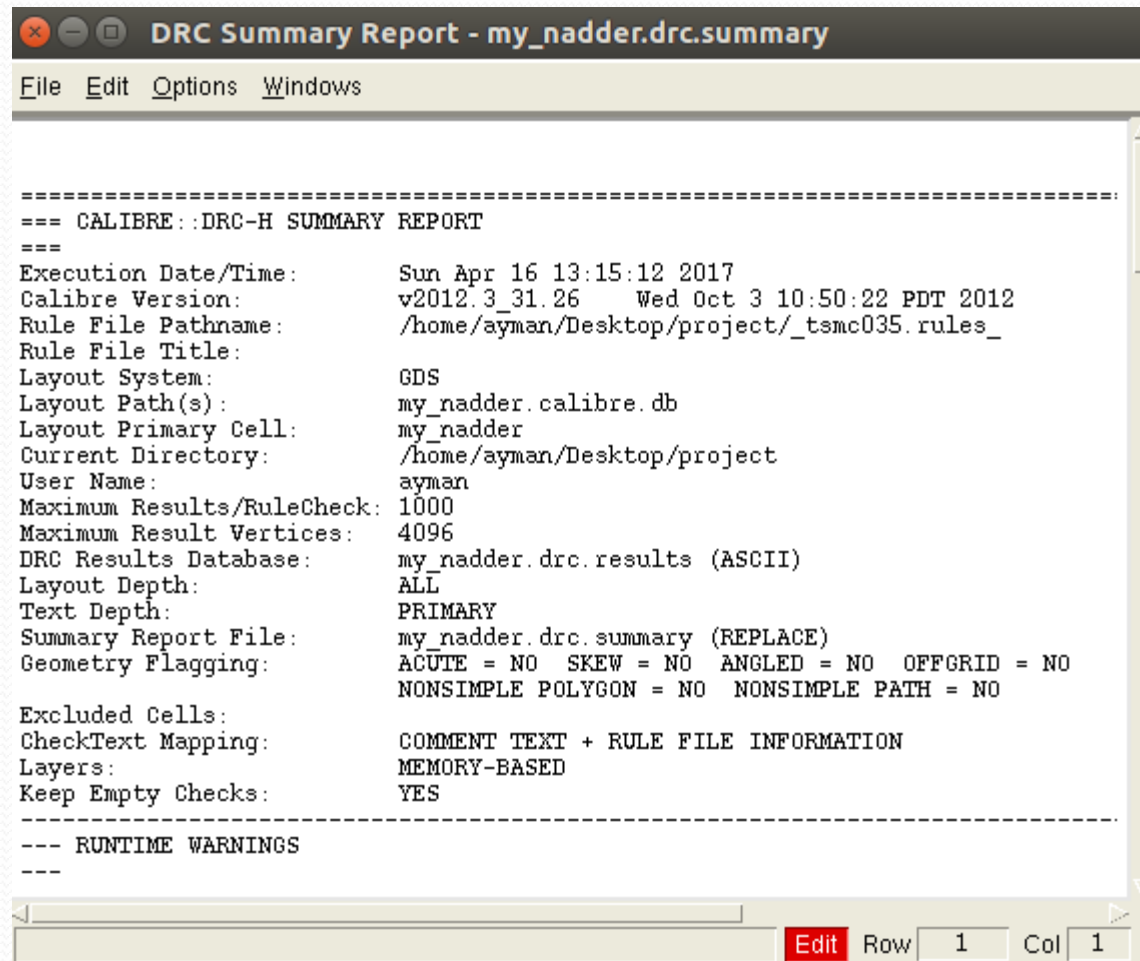


DRC

- Click on “Run DRC”



DRC



```
=====
=== CALIBRE::DRC-H SUMMARY REPORT
===
Execution Date/Time:      Sun Apr 16 13:15:12 2017
Calibre Version:          v2012.3_31.26      Wed Oct 3 10:50:22 PDT 2012
Rule File Pathname:       /home/ayman/Desktop/project/_tsmc035.rules_
Rule File Title:
Layout System:            GDS
Layout Path(s):           my_nadder.calibre.db
Layout Primary Cell:      my_nadder
Current Directory:        /home/ayman/Desktop/project
User Name:                ayman
Maximum Results/RuleCheck: 1000
Maximum Result Vertices:  4096
DRC Results Database:     my_nadder.drc.results (ASCII)
Layout Depth:             ALL
Text Depth:               PRIMARY
Summary Report File:      my_nadder.drc.summary (REPLACE)
Geometry Flagging:        ACUTE = NO  SKEW = NO  ANGLED = NO  OFFGRID = NO
                           NONSIMPLE POLYGON = NO  NONSIMPLE PATH = NO

Excluded Cells:
CheckText Mapping:        COMMENT TEXT + RULE FILE INFORMATION
Layers:                   MEMORY-BASED
Keep Empty Checks:        YES
=====
--- RUNTIME WARNINGS
---
```

Edit Row 1 Col 1

DRC

- Calibre RVE Shows which checks passed & which failed
- Failed checks (red 'x') Rule description is below
Click on number on right

The screenshot displays the Calibre RVE v2012.3_31.26 interface. The title bar reads "Calibre - RVE v2012.3_31.26 : my_nadder.drc.results". The menu bar includes File, View, Highlight, Tools, Window, and Setup. Below the menu is a toolbar with icons for file operations, search, and navigation. The main window shows "Topcell my_nadder, 10 Results (in 4 of 81 Checks)" and a "Show All" button. A table lists the DRC checks and their results:

Check / Cell	Results
✓ Check DRC13_4	0
✓ Check DRC13_5	0
✓ Check DRC14_1	0
✓ Check DRC14_2	0
✓ Check DRC14_3	0
✓ Check DRC15_1	0
✗ Check DRC15_2	1
✓ Check DRC15_3	0
✓ Check DRC15_4	0
✓ Check DRC21_1	0
✓ Check DRC21_2	0
✓ Check DRC21_3	0
✗ Check DRC22_1	1
✓ Check DRC22_2	0
✓ Check DRC22_3	0
✓ Check DRC22_4	0

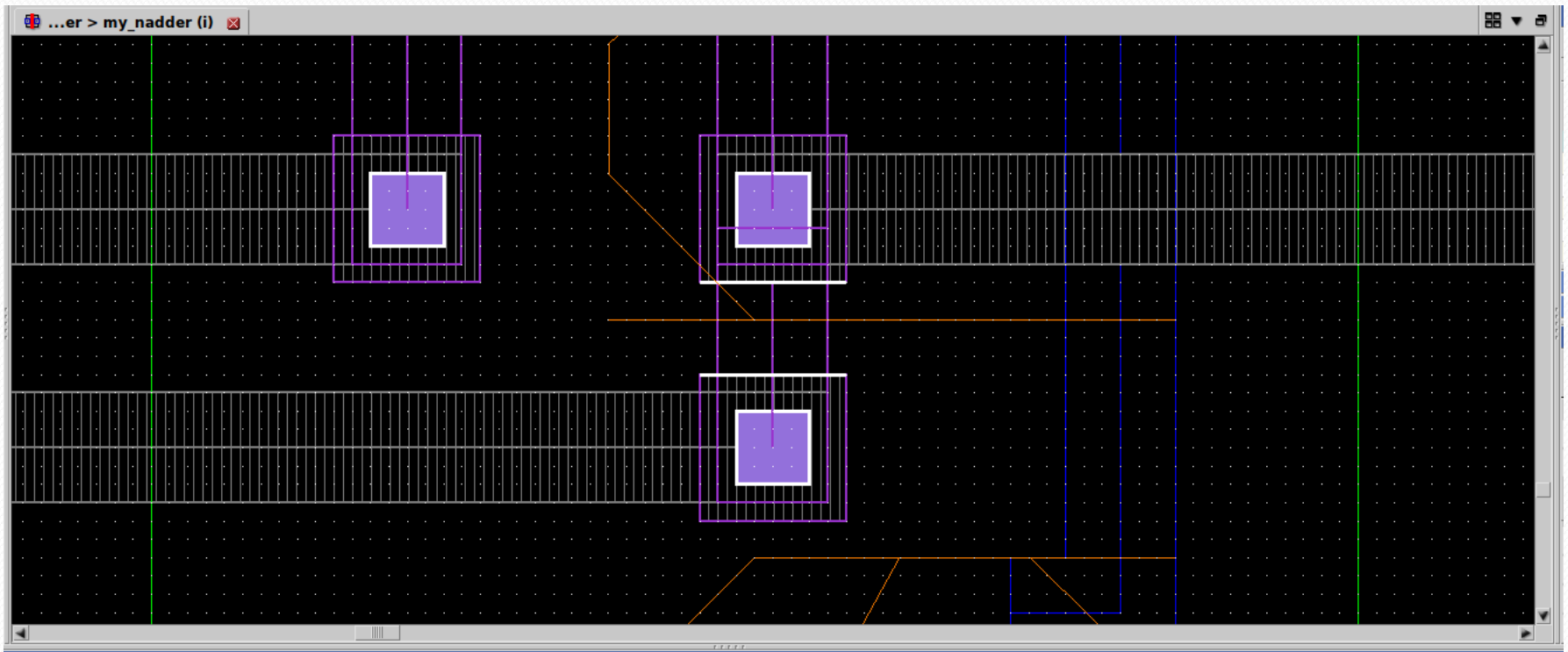
On the right, a detailed view for "9) Check DRC15_2, Cell my_nadder: 2 Edges" is shown. It states "2 Edges. Coordinates in cell my_nadder" and lists the coordinates:

(327.0 144.5) (331.0 144.5)
(327.0 147.0) (331.0 147.0)

At the bottom, the Rule File Pathname is shown as `/home/ayman/Desktop/project/_tsmc035.rules_` and Metal3 spacing is set to 3L.

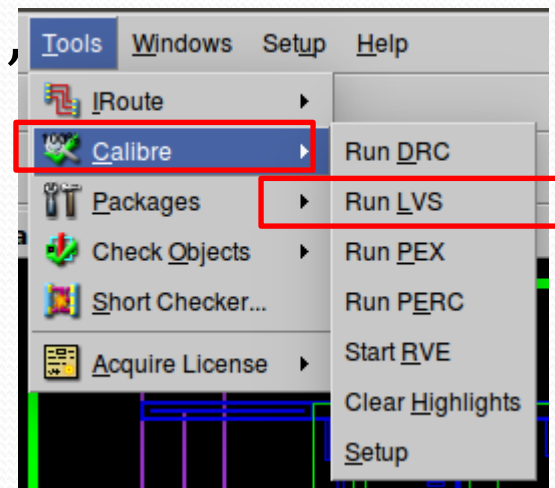
DRC

- The place, where DRC is violated, will appear on Pyxis



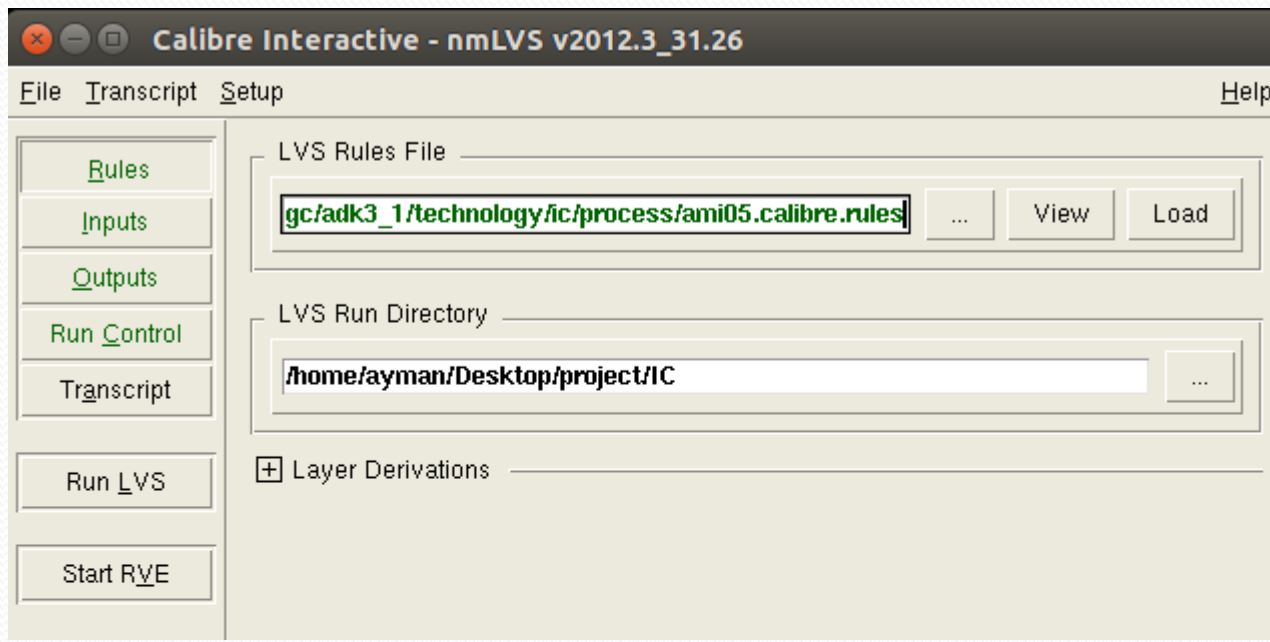
LVS

- We will run Calibre LVS from Pyxis also
- From “Tools”, select “Calibre”
 - Then select “Run LVS”



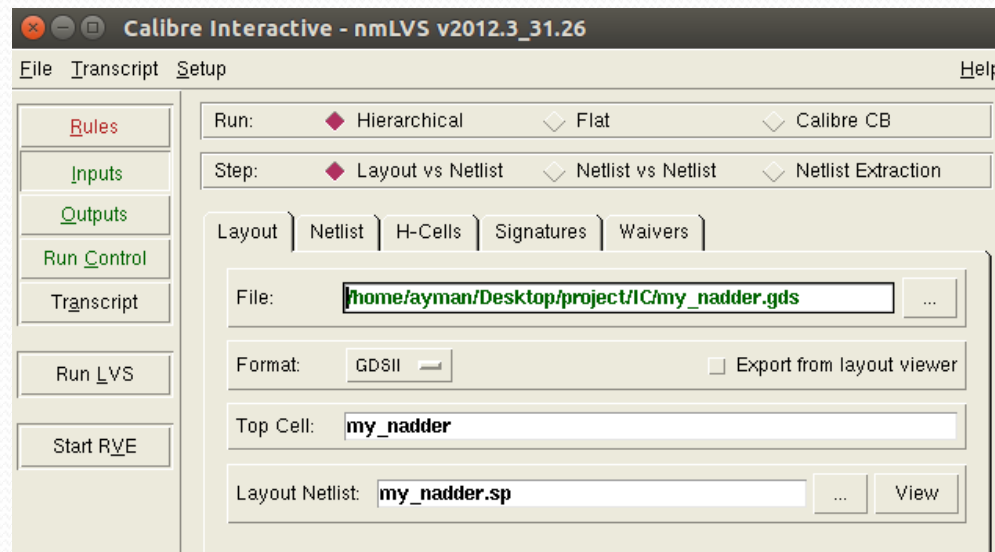
LVS

- In “Rules” Tab:
 - Set Rules to
“Technology/ic/process/ami05.calibre.rules”
 - Set Working Directory



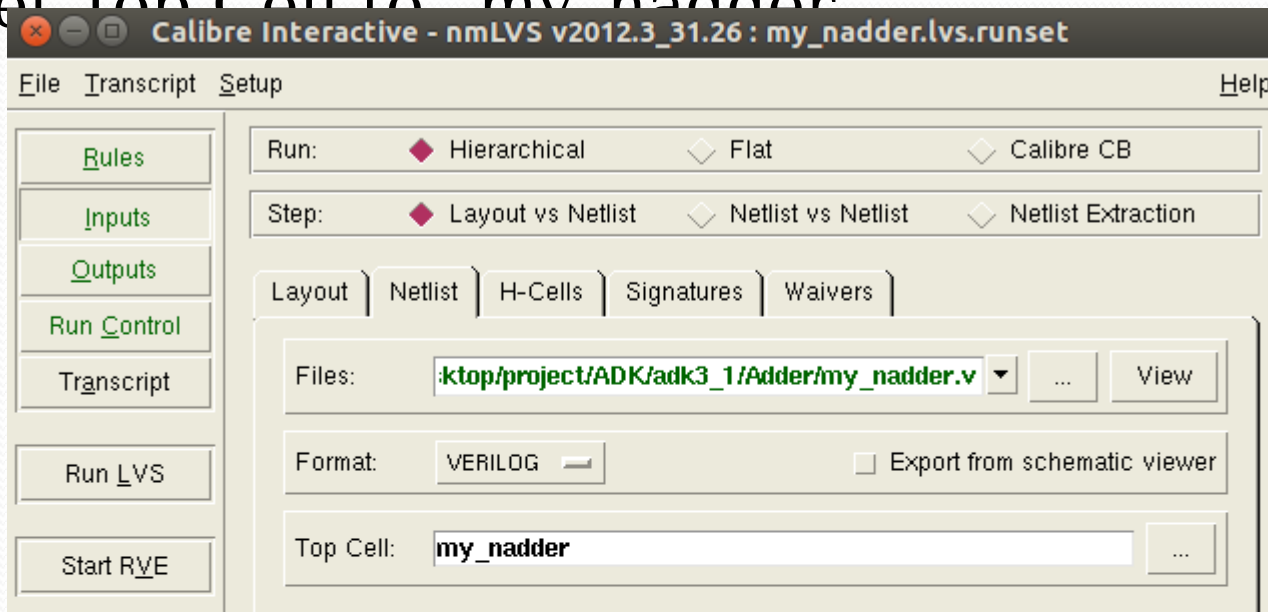
LVS

- In “Inputs” Tab (Layout sub-tab):
 - Uncheck “Export from Layout viewer”
 - Set File to the exported gds file
 - Set Top Cell to “my_nadder”
 - Set Layout Netlist to my_nadder.sp



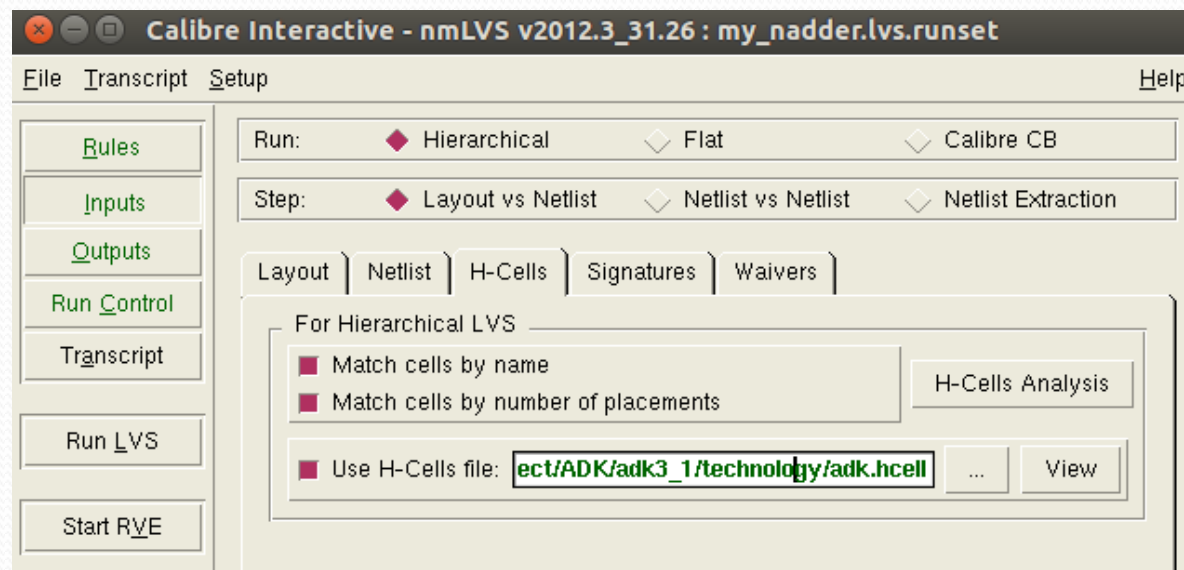
LVS

- In “Inputs” Tab (Netlist sub-tab):
 - Set Format to “VERILOG” & Uncheck “Export from ...”
 - Set file to the verilog netlist
 - Set Top Cell to “my_nadder”



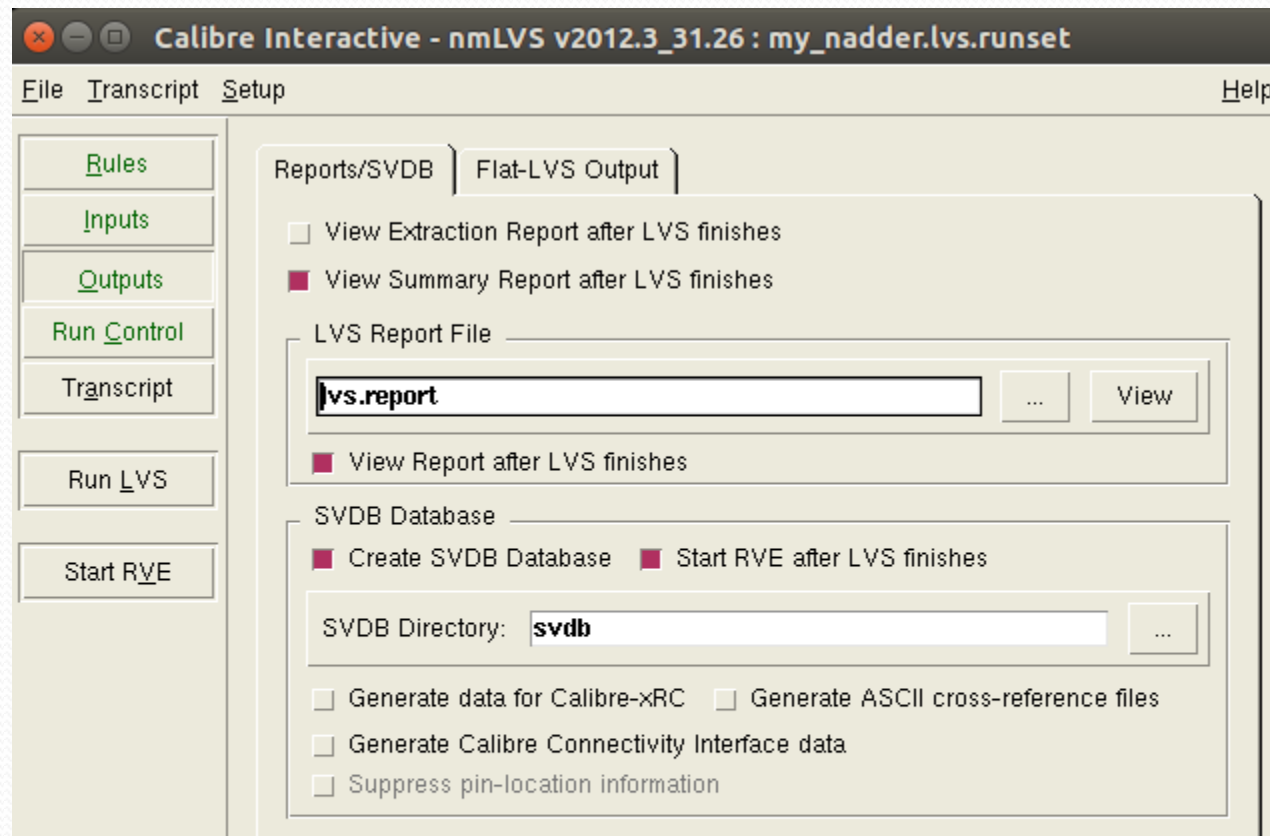
LVS

- In “Inputs” Tab (H-Cells sub-tab):
 - Check “Match cells by name”
 - Check “Match cells by number of placement”
 - Check “Use H-Cells file”
 - Set H-Cells file to “technology/adk.hcell”



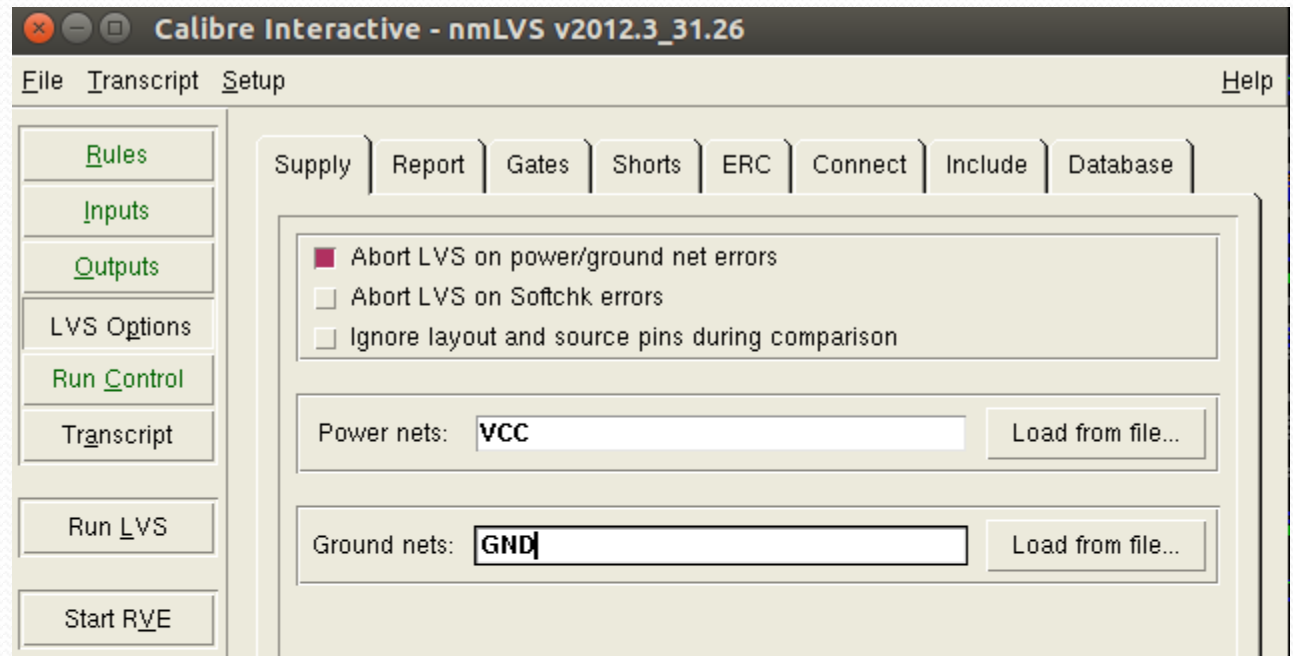
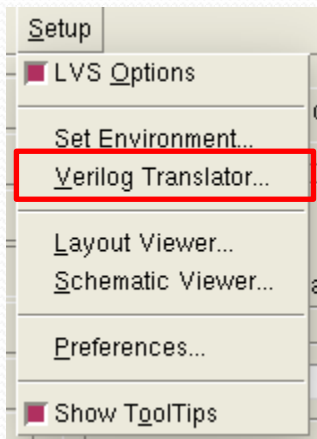
LVS

- In “Outputs” Tab:



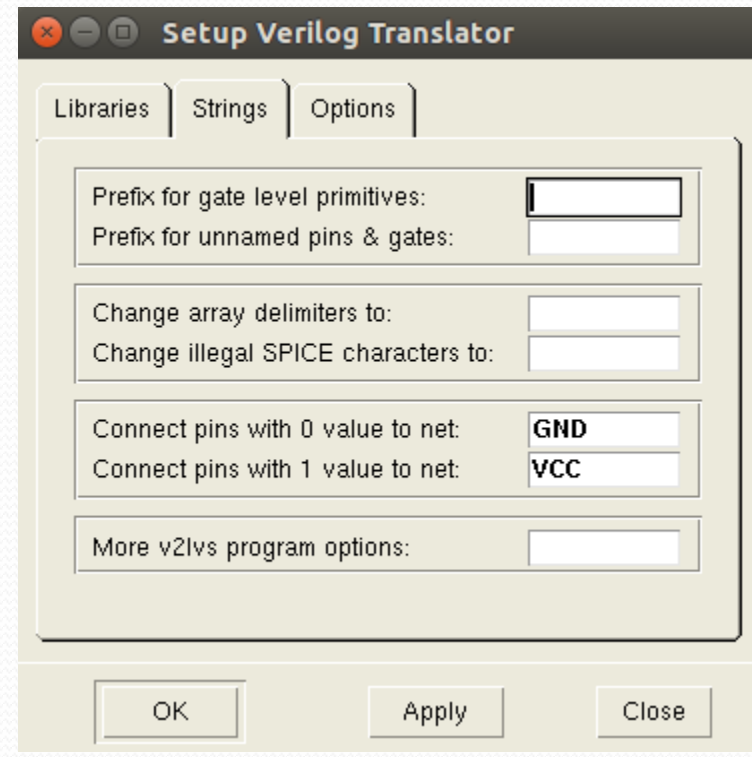
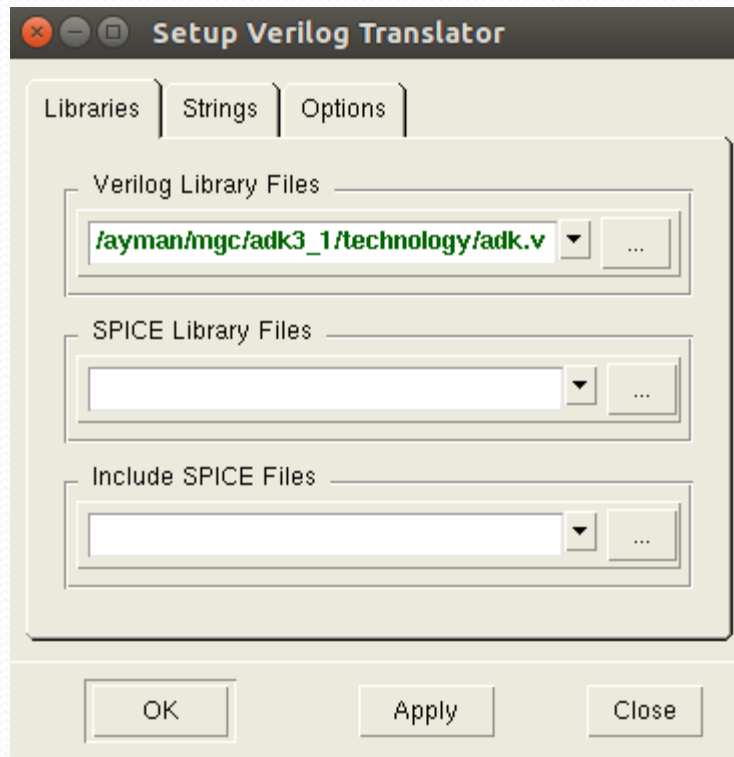
LVS

- From “Setup” Menu
 - Check “LVS Options”
 - Set Power to “VCC” & Ground to “GND”
 - Select Verilog Translator



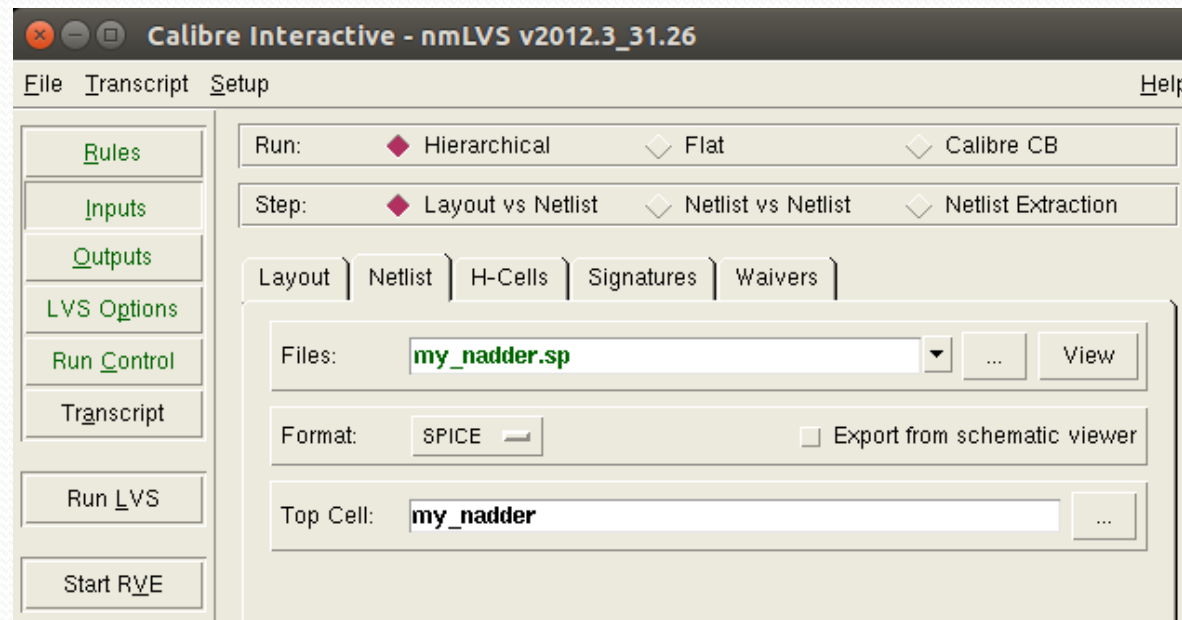
LVS

- In “Verilog Translator” window
 - Set Verilog Library to “technology/adk.v”
 - Set pins with 0 “GND” & pins with 1 “VCC”



LVS

- After running the LVS for 1st time, Spice Netlist is Generated
- Go back to Inputs & Read “my_nadder.sp”
- Re-run LVS



LVS

- RVE Report shows which comparison failed & why

The screenshot displays the Calibre RVE v2012.3_31.26: svdb my_nadder interface. The main window shows the 'Comparison Results' tab, which contains a table of comparison data and a detailed view of the 'Cell aoi22 Summary (Clean)'.

Comparison Results Table:

Layout Cell / Type	Source Cell	Nets	Instances	Ports
aoi22	aoi22	7L, 7S	3L, 3S	7L, 7S
buf02	buf02	4L, 4S	1L, 1S	3L, 3S
oai22	oai22	7L, 7S	3L, 3S	4L, 4S
xnor2	xnor2	6L, 6S	4L, 4S	5L, 5S
my_nadder	my_nadder	66L, 66S	37L, 37S	30L, 30S

Cell aoi22 Summary (Clean):

CELL COMPARISON RESULTS

CORRECT #

#####

LAYOUT CELL NAME: aoi22
SOURCE CELL NAME: aoi22

INITIAL NUMBERS OF OBJECTS

Lavout Source Component Type