

Netlist to GDS

Inputs required:

- ✓ Synthesized Netlist
- ✓ Sdc file along with netlist
- ✓ Qrc Tech file
- ✓ Library file that we have used for synthesis.
- ✓ Lef files

Output: **Layout**

FLOW:

Typical Backend Flow:

Synthesis -> Floorplan -> Power plan -> Placement -> CTS -> Routing -> verification -> Layout

STEP BY STEP PROCESS:

Here, I've used an ALU block that I created to demonstrate the Innovus flow.
I'll just explain the approach I used step by step.

- Before everything, create a folder ALU_Layout in which it has again three folders in it.
 - Inputs
 - Work
 - Outputs

/home/users/mtech/venkata/alu_layout/		
Name	Size (KB)	Last modified
..		
outputs		2023-10-14 21:02
inputs		2023-10-14 22:52
work		2023-10-16 03:49

- In inputs folder, make sure that following required files are there:

/home/users/mtech/venkata/alu_layout/inputs/		
Name	Size (KB)	Last modified
..		
ccopt_spec_file.tcl	1	2023-10-14 23:01
alu.sdc	7	2023-10-14 20:49
MMMC.tcl	1	2023-10-14 20:49
alu_init.tcl	1	2023-10-14 20:36
alu.dc.v	61	2023-10-14 20:23

The dc.v (netlist) and sdc files were obtained from the synthesis output, while the other three '.tcl' files were created and appropriate scripting instructions were entered into them.

mmmc.tcl:

Here, in the following script,

In 1st and 2nd lines, i.e., path of qrcTechfile and

In 3rd and 4th lines, i.e., path of library files

```

1  create_rc_corner -name rc_worst -qx_tech_file {/home/users/mtech/venkata/alu_layout2/inputs/qrcTechFile}
2  create_rc_corner -name rc_best -qx_tech_file {/home/users/mtech/venkata/alu_layout2/inputs/qrcTechFile}
3  create_library_set -name fast_libs -timing {/home/users/mtech/venkata/UMC65/UMK65LSCLLMVBBR_B03_TAPEOUTKIT/synopsys/uk65lscllmvbbbr_110c-40_bc.lib}
4  create_library_set -name slow_libs -timing {/home/users/mtech/venkata/UMC65/UMK65LSCLLMVBBR_B03_TAPEOUTKIT/synopsys/uk65lscllmvbbbr_108c125_wc.lib}
5  create_constraint_mode -name func_mode -sdc_files {../inputs/alu.sdc}
6  create_delay_corner -name slow_max -library_set {slow_libs} -rc_corner {rc_worst}
7  create_delay_corner -name fast_min -library_set {fast_libs} -rc_corner {rc_best}
8  create_analysis_view -name func_max -constraint_mode {func_mode} -delay_corner {slow_max}
9  create_analysis_view -name func_min -constraint_mode {func_mode} -delay_corner {fast_min}
10 set_analysis_view -setup {func_max} -hold {func_min}

```

alu_init.tcl:

Here, line no 17, i.e., lef files path

```

1  global init_design_uniquify
2  global init_import_mode
3  global init_verilog
4  global init_design_netlisttype
5  global init_design_Settop
6  global init_top_cell
7  global init_lef_file
8  global init_mmmc_file
9  global init_pwr_net
10 global init_gnd_net
11 global init_io_file
12 set init_design_uniquify 1
13 set init_import_mode {-treatUndefinedCellAsBbox 0 -keepEmptyModule 1}
14 set init_verilog "../inputs/alu_dc.v"
15 set init_design_netlisttype Verilog
16 set init_design_Settop 1
17 set init_lef_file "/home/users/mtech/venkata/UMC65/UMK65LSCLLMVBBR_B03_TAPEOUTKIT/lef/tf/uk65lscllmvbbbr_7m2t0f.lef /home/users/mtech/venkata/UMC65/UMK65LSCLLMVBBR_B03_TAPEOUTKIT/lef/uk65lscllmvbbbr.lef"
18 set init_mmmc_file "../inputs/MMMC.tol"
19 set init_pwr_net {VDD}
20 set init_gnd_net {VSS}
21

```

Cc_opt_spec.tcl:

This one will be useful during CTS step

```

1  set_ccopt_mode -integration native -cts_inverter_cells {CKINVM8R CKINVM12R} -cts_buffer_cells {CKBUFM8R CKBUFM12R} -cts_use_inverters true
2  set_ccopt_mode -cts_target_skew 0.1
3  set_ccopt_mode -cts_target_slew 0.2
4  set_ccopt_mode -cts_target_nonleaf_slew 0.25
5  create_ccopt_clock_tree_spec -filename ctscocopt.spec
6  source ctscocopt.spec

```

And for the time being, the work and output folders are empty. Eventually, they will receive certain files along the procedure.

Now that we have the necessary inputs and tcl scripts, the real procedure can begin:

- Go to work folder and invoke the innovus along with alu_init.tcl

type the following terminal commands one by one as they are:

- cd alu_layout
- cd_work
- csh
- source ../../cshrc
- innovus -64 -log ../logs/alu_FP.log -win -init ../inputs/alu_init.tcl
- init_design

- innovus will be opened in new window
 - ignore that for now, and perform the followings checks, before you start floorplanning.
- type the following terminal commands one by one as they are:

- checkDesign -all -noHtml -outfile ../outputs/checkdesign.rpt
- check_timing -verbose > ../outputs/checktiming.rpt
- checkNetlist -outfile ../outputs/checknetlist.rpt
- checkUnique -verbose
- timedesign -prePlace

Check the outputs folder, and if all of the reports are OK, we may proceed to the next stage.

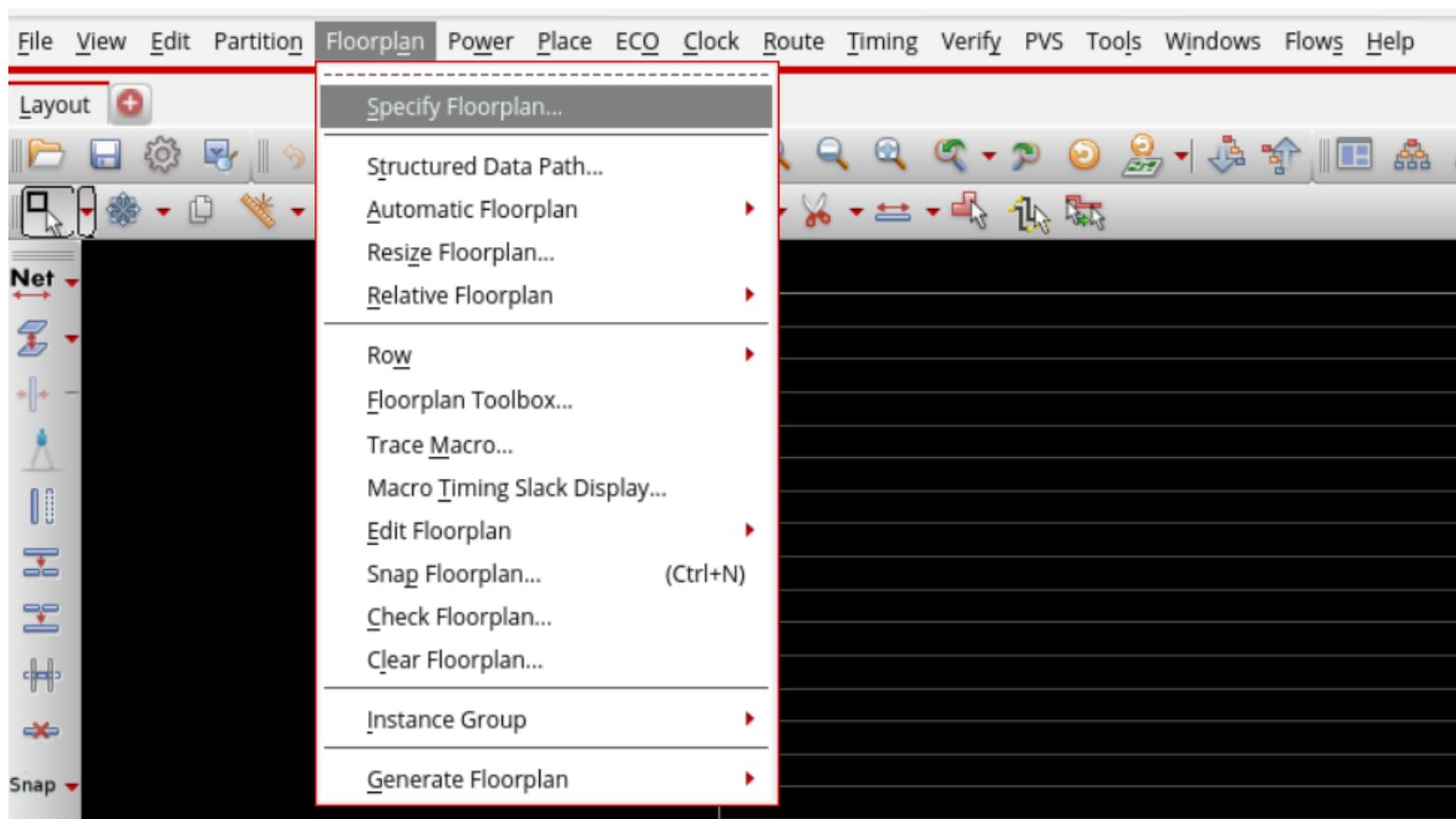
Now that the formal inspections have been completed and everything is in order, we can begin our actual **netlist to layout** procedure.

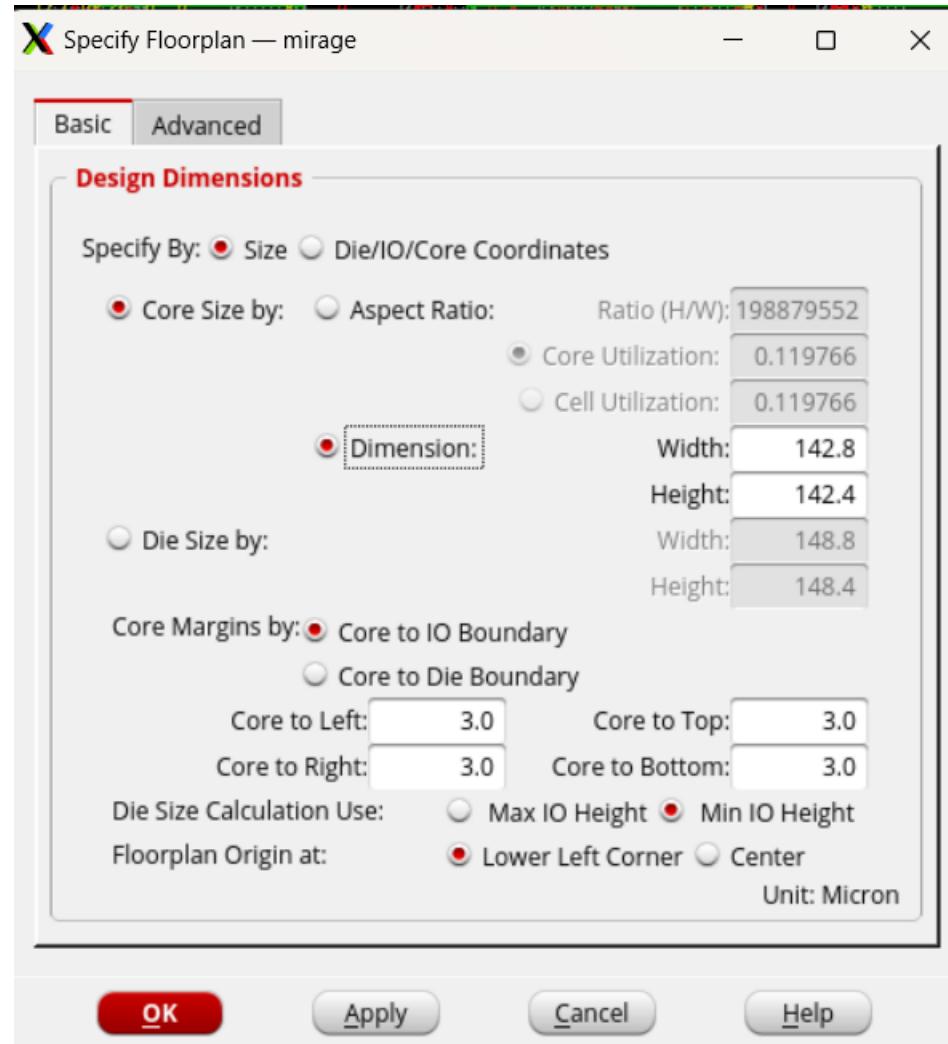
Cadence Innovus Flow:

- Floor plan
- Pin editor
- Power plan
- Connect global nets
- Add end cap
- Special route
- Placement
- Pre CTS
- CTS
- Post CTS
- Route
- Post route
- Layout

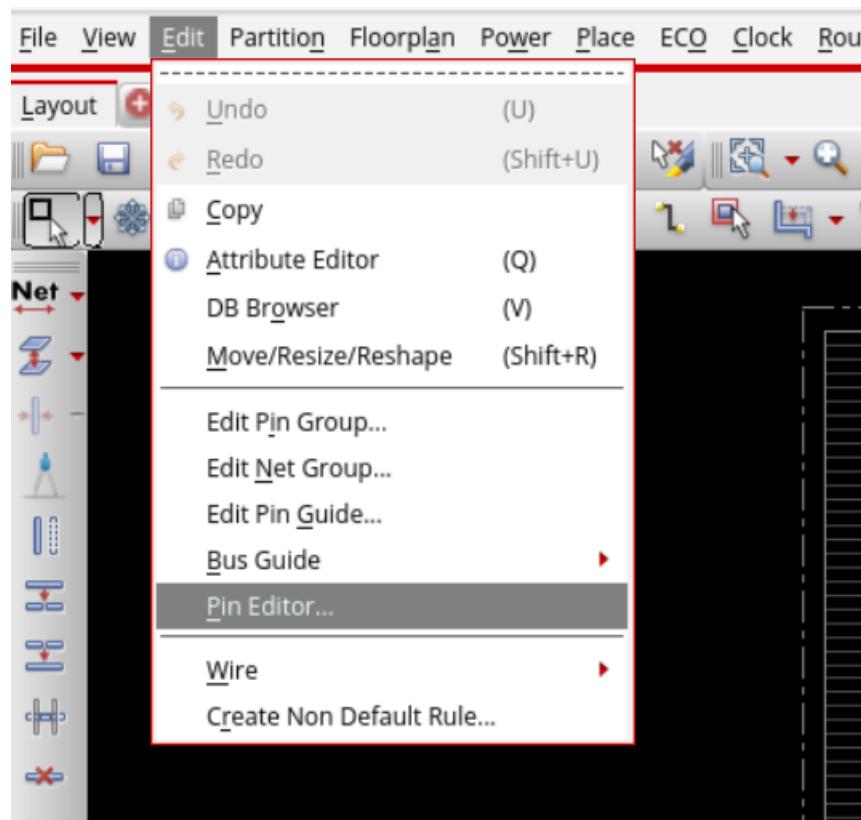
Follow the order of the photographs here from now on, one by one, step by step.

✓ Step1 - Specify Floor plan

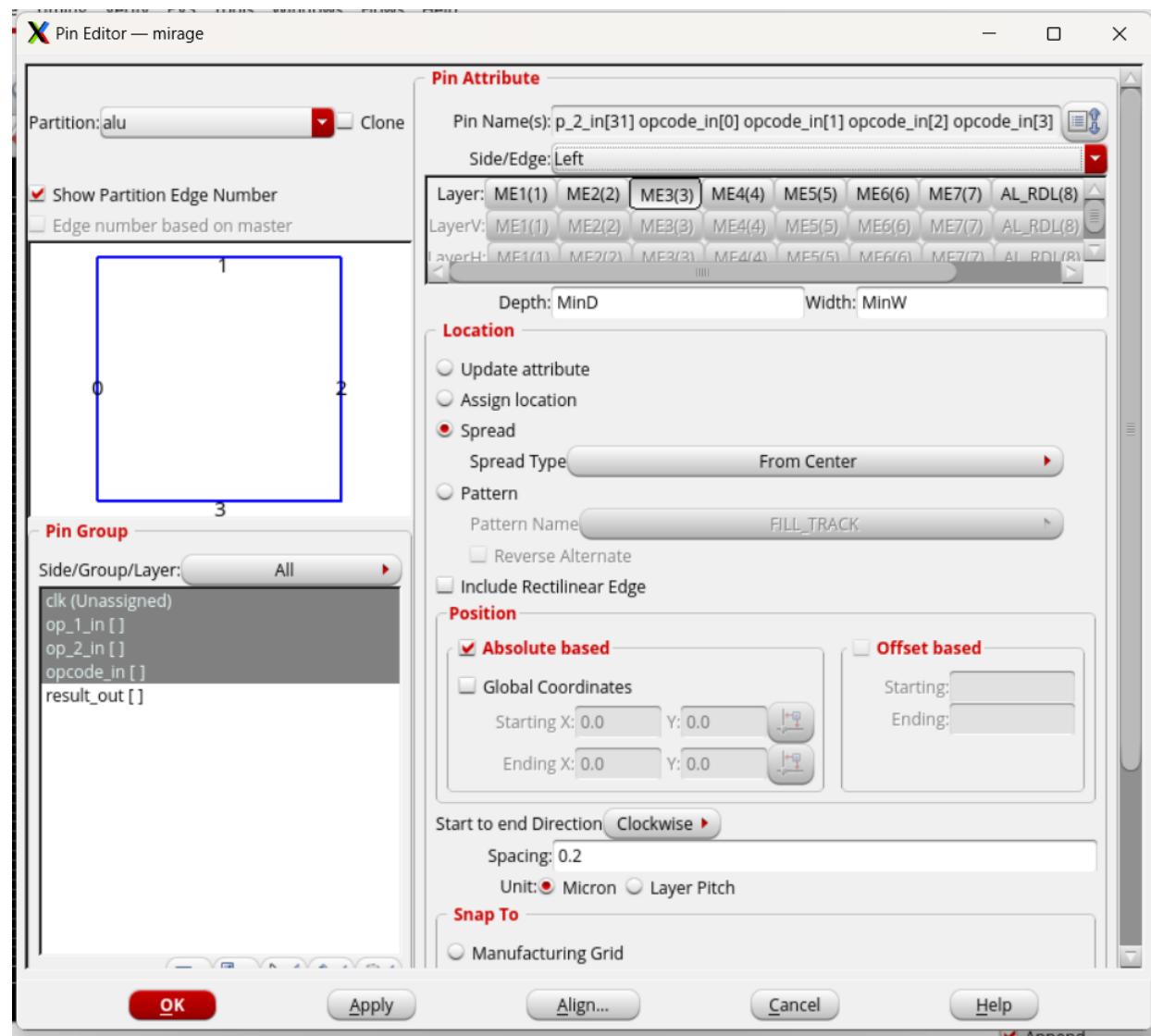




✓ Step2 - Pin Editor



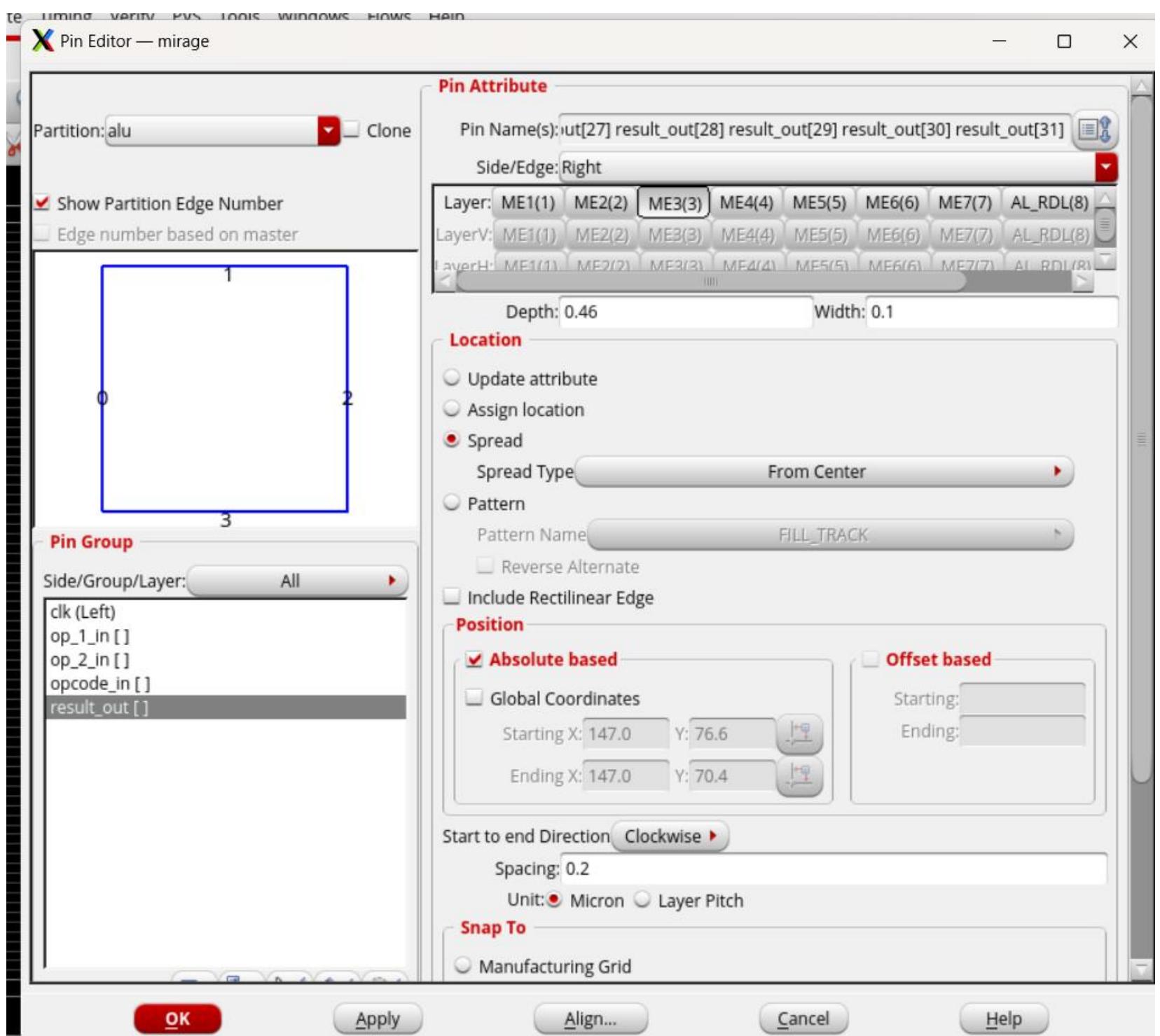
- Select inputs first



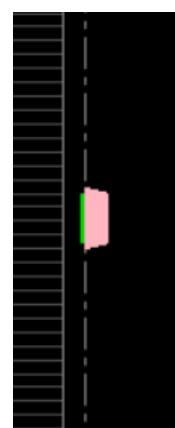
It will create pins like this



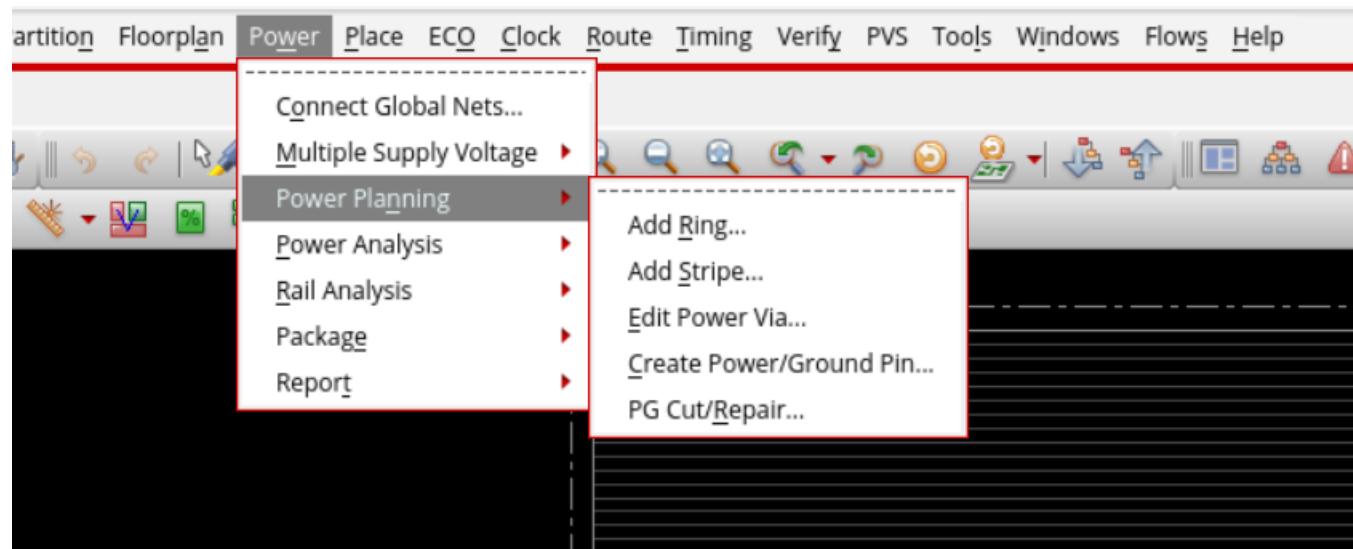
Now, output pins:

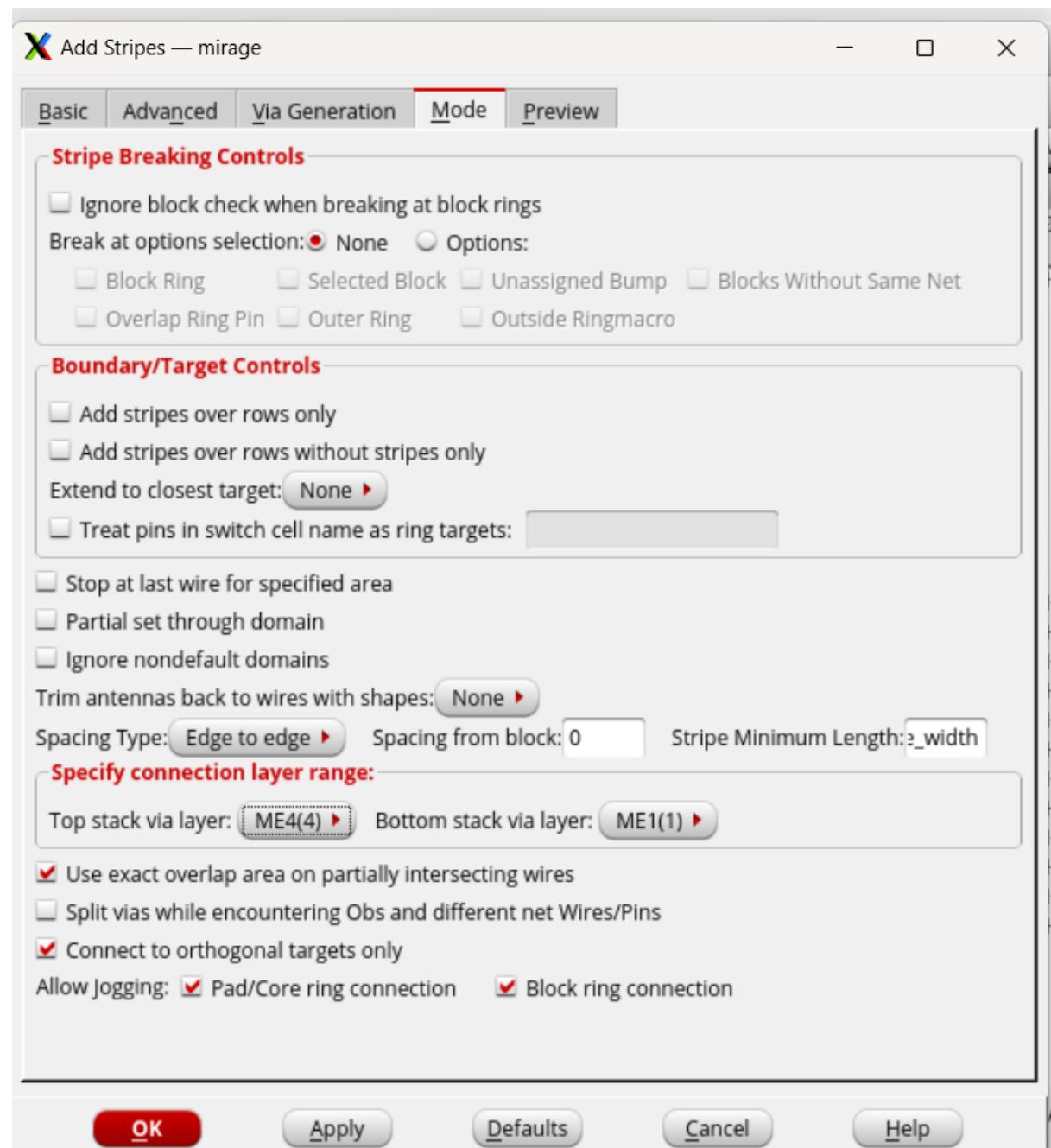


It will create output pins like this:

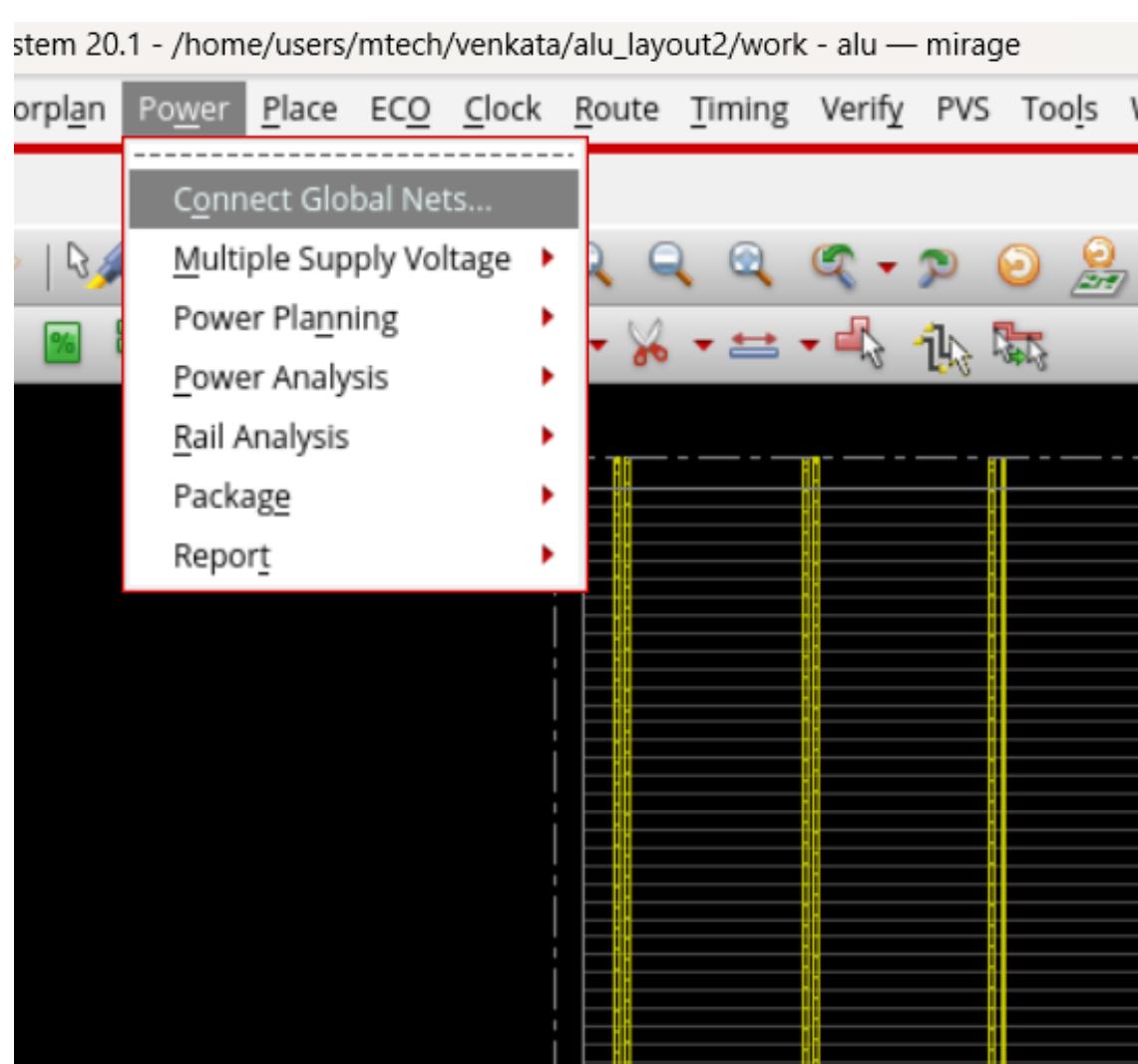


✓ Step3 - Power plan

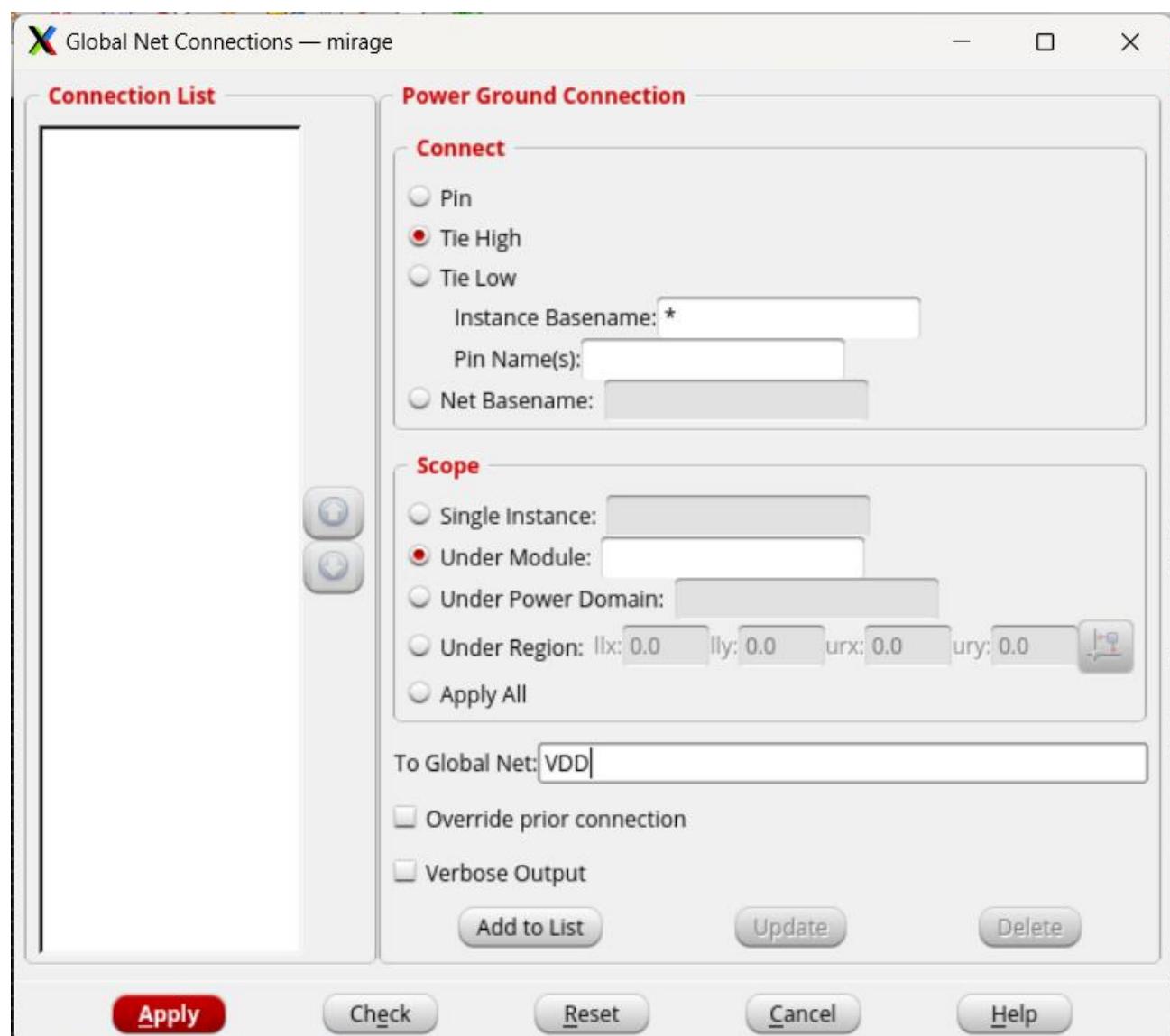




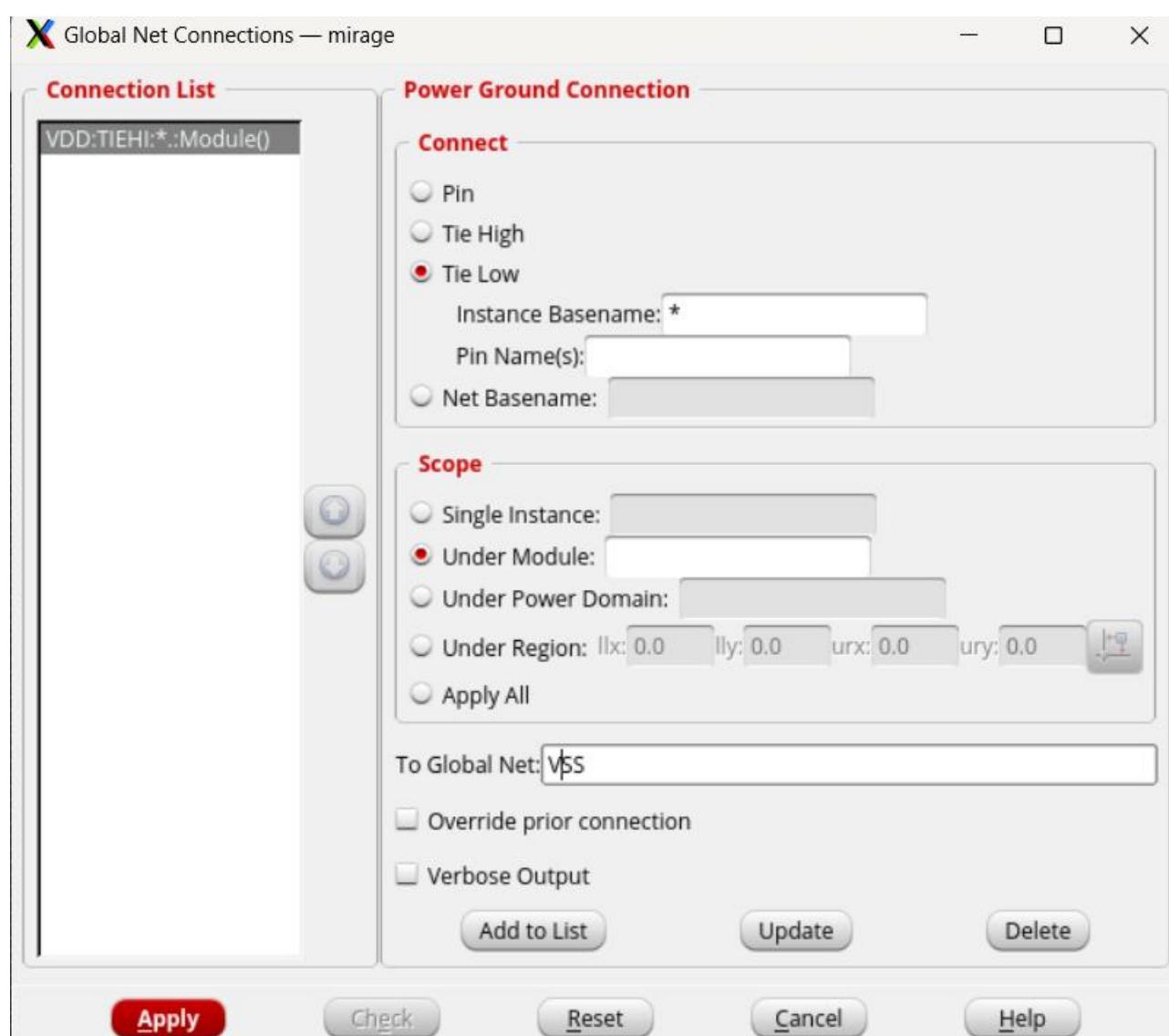
✓ Step4 - Connect Global nets



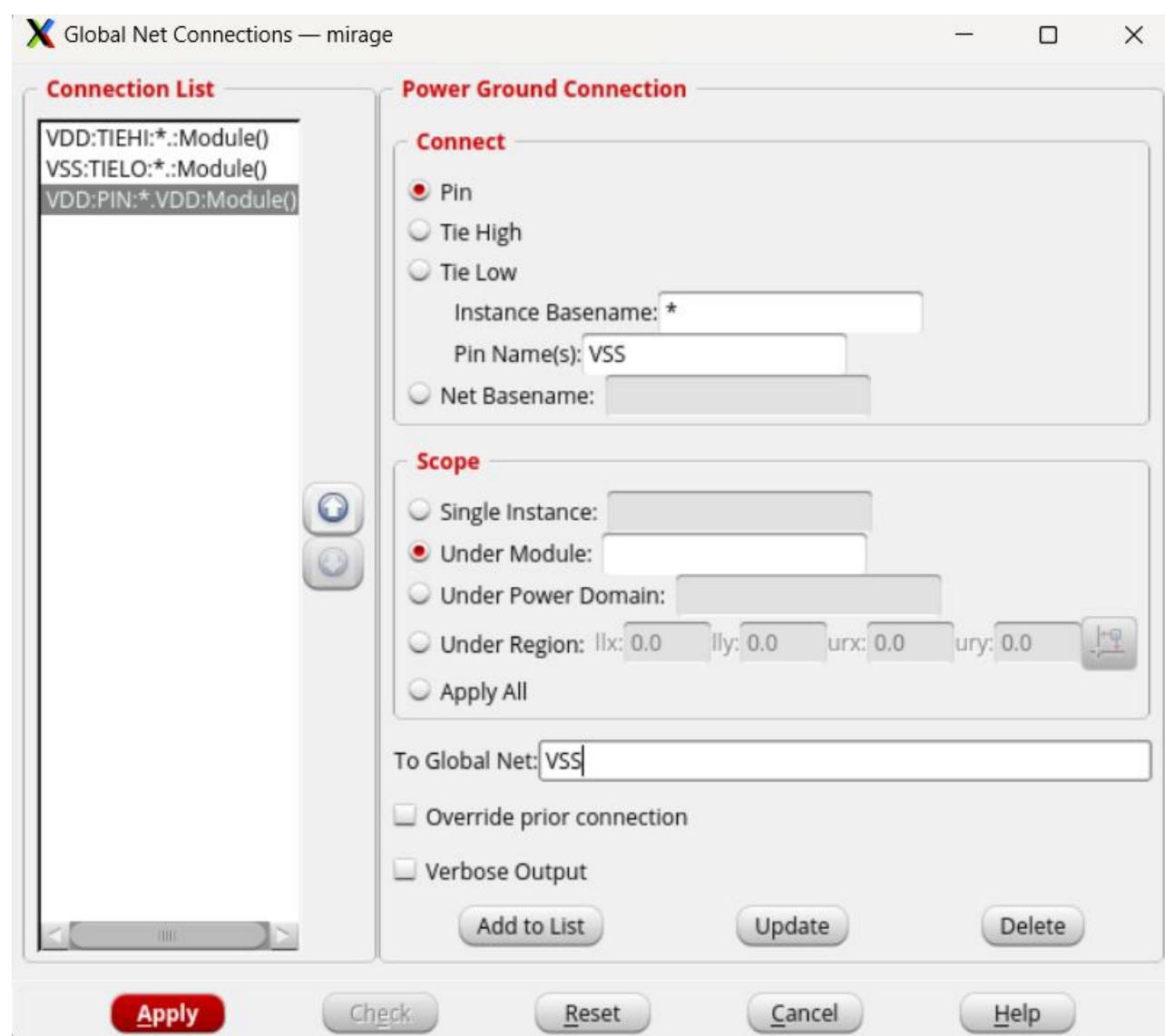
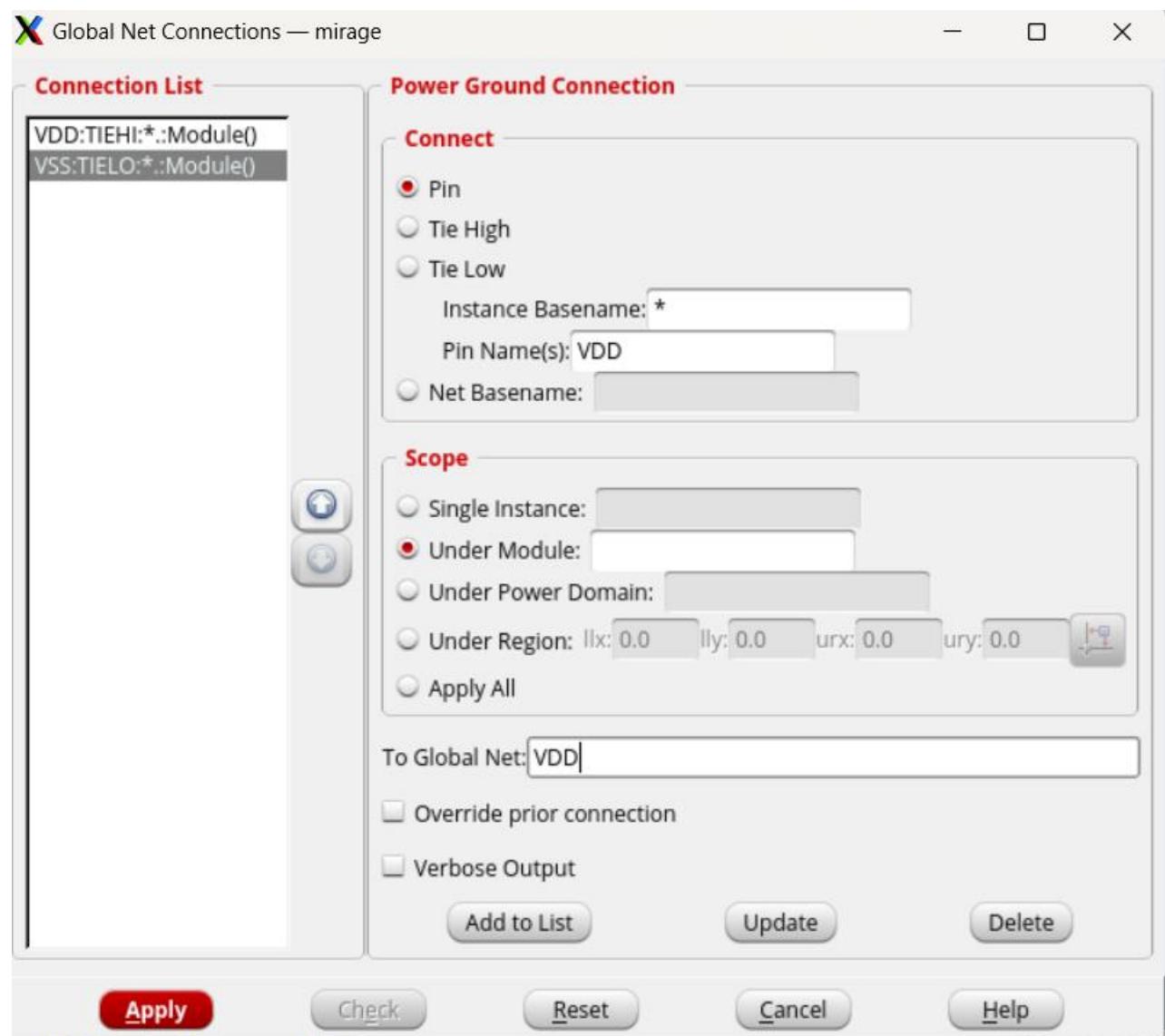
- Click on Tie High, to global net: VDD and click on Add to list



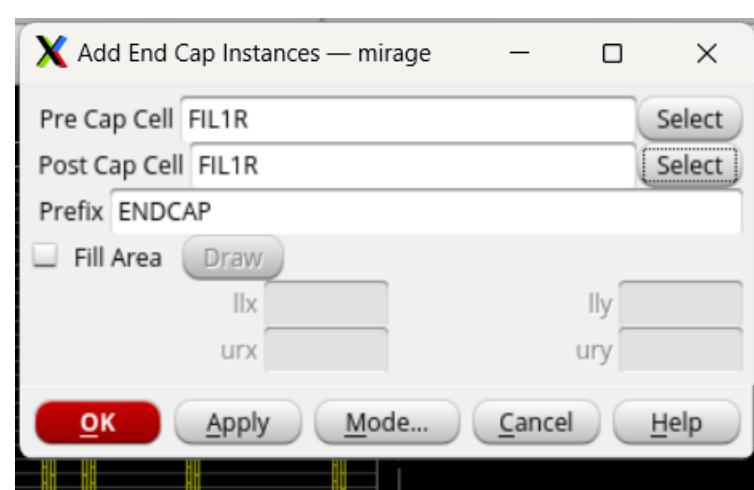
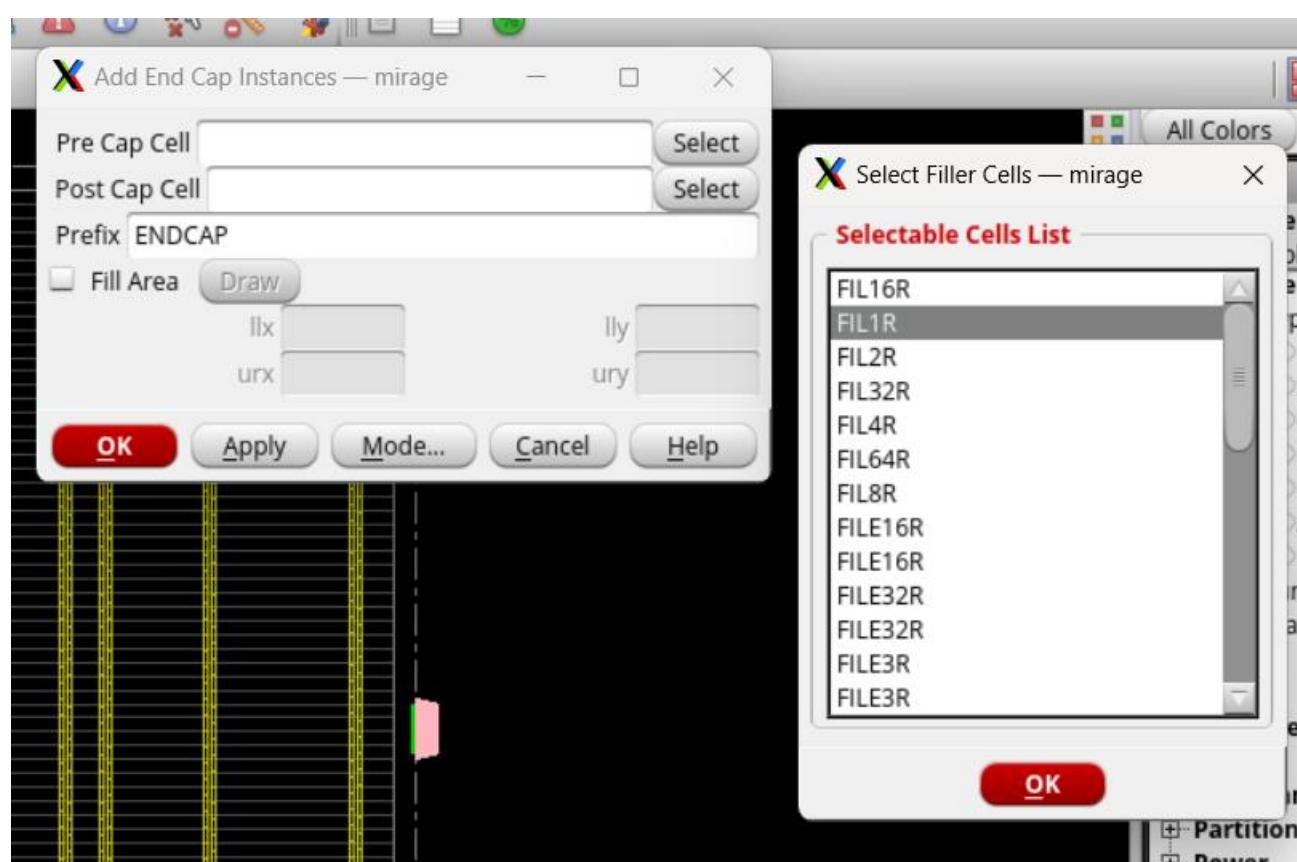
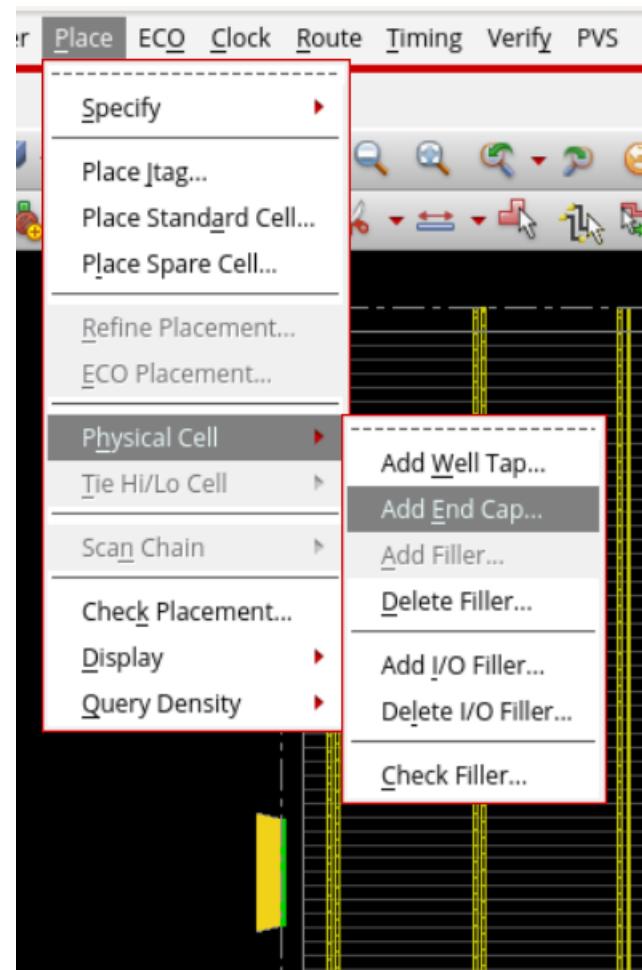
- Next, VSS.



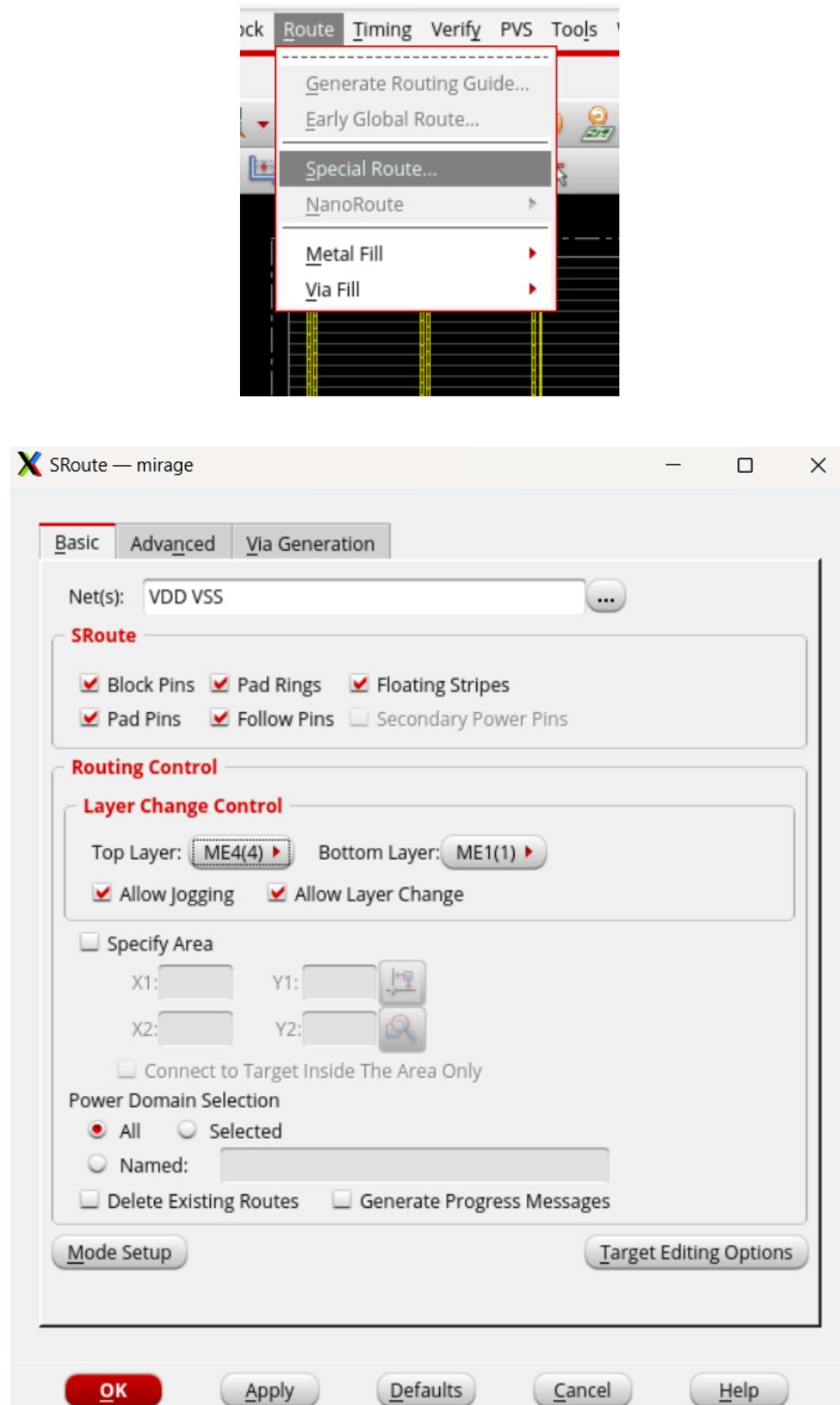
- o Next, pins: VDD and VSS



✓ Step5 - Add End Cap

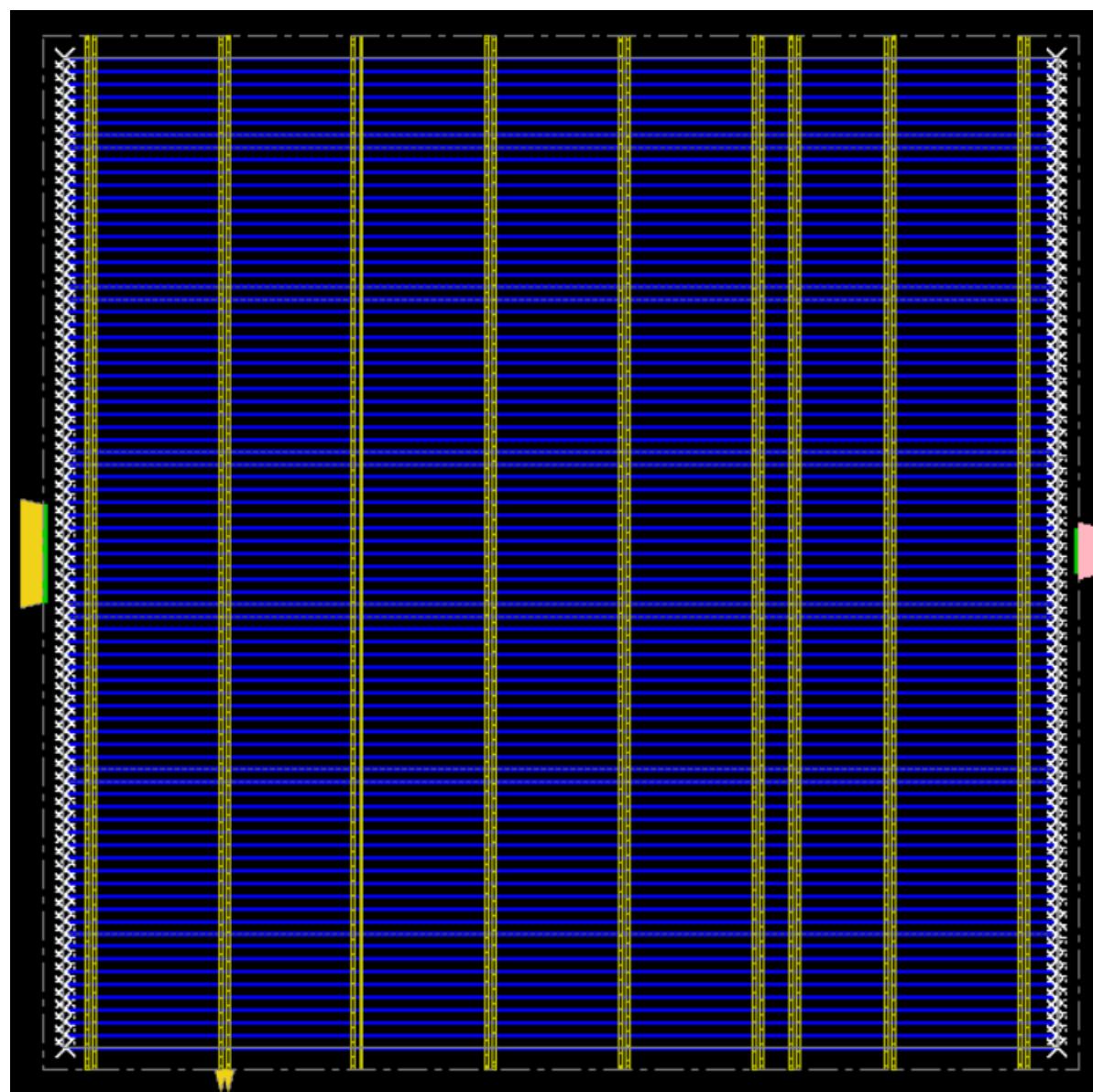


✓ Step6 - Special route

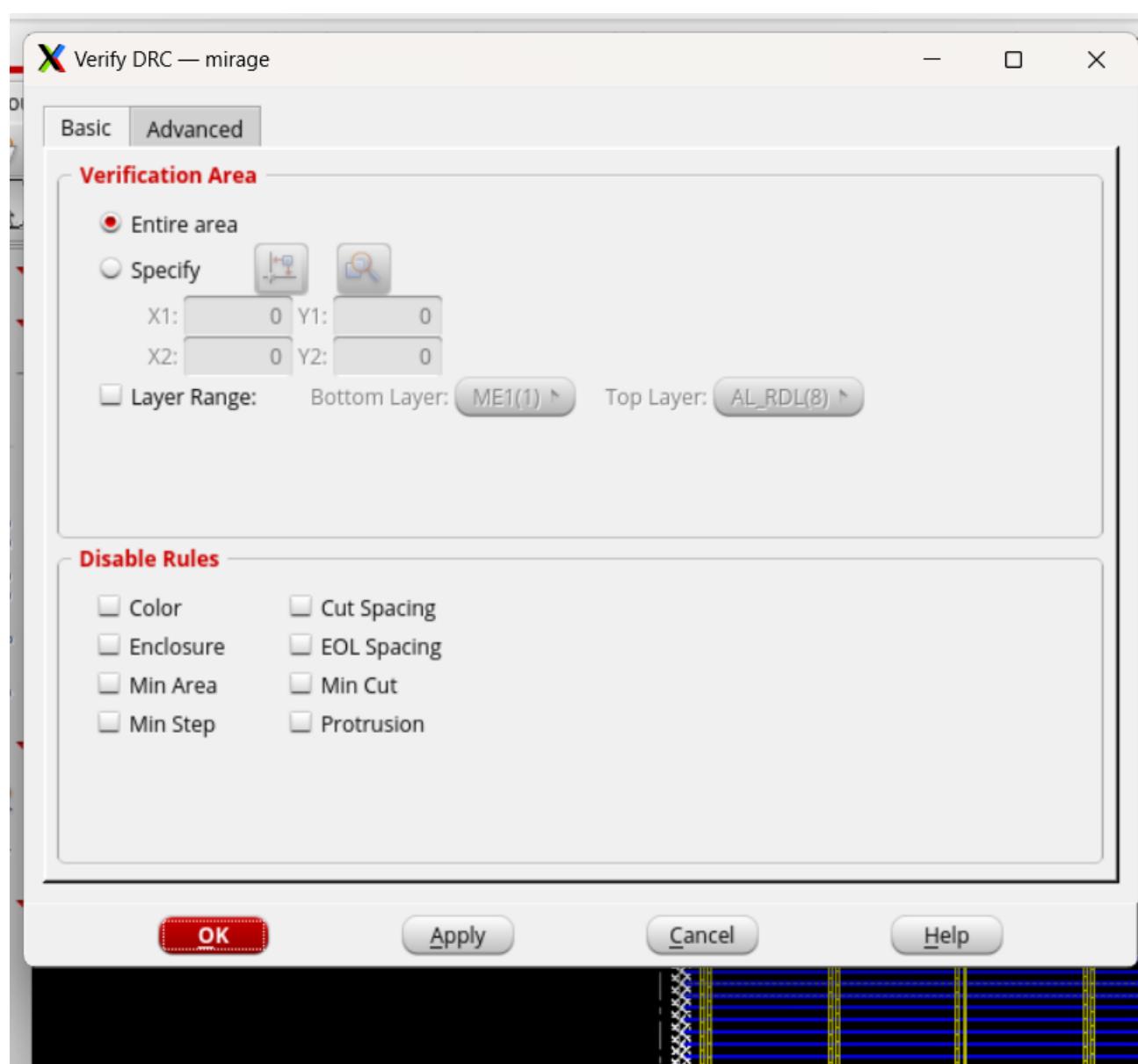
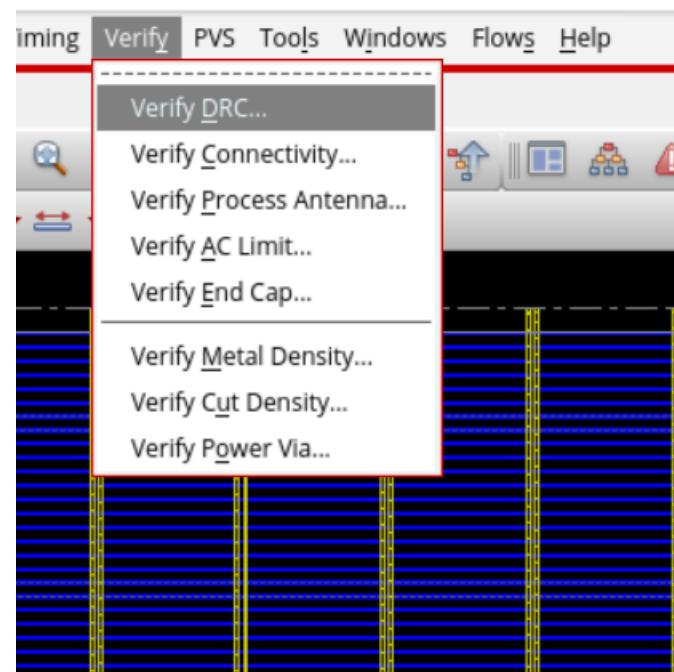




- After S-route, it will be like this:



✓ Step7 - Verify



It should come zero DRC violations

```
VERIFY DRC ..... Sub-Area : 4 complete 0 Viols.  
Verification Complete : 0 Viols.  
*** End Verify DRC (CPU: 0:00:00.0 ELAPSED TIME: 0.00 MEM: 0.0M) ***
```

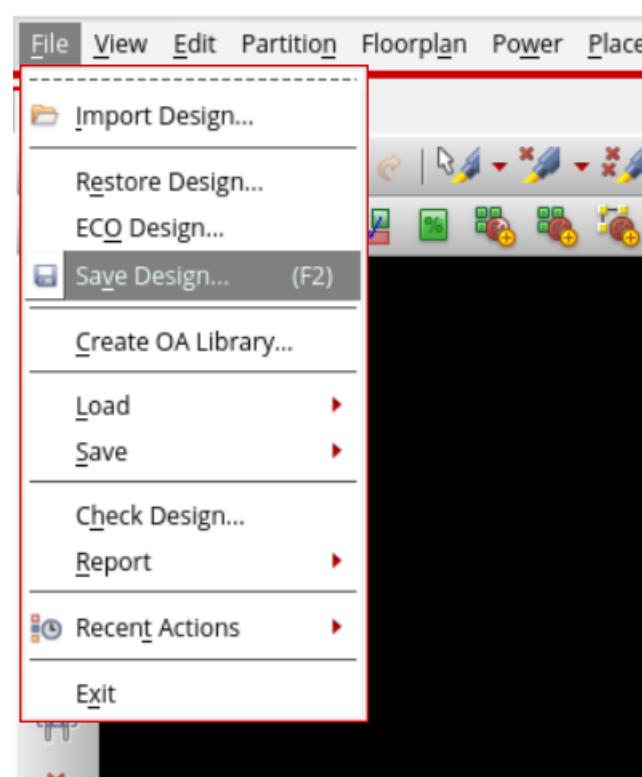
Next, verify connectivity also:

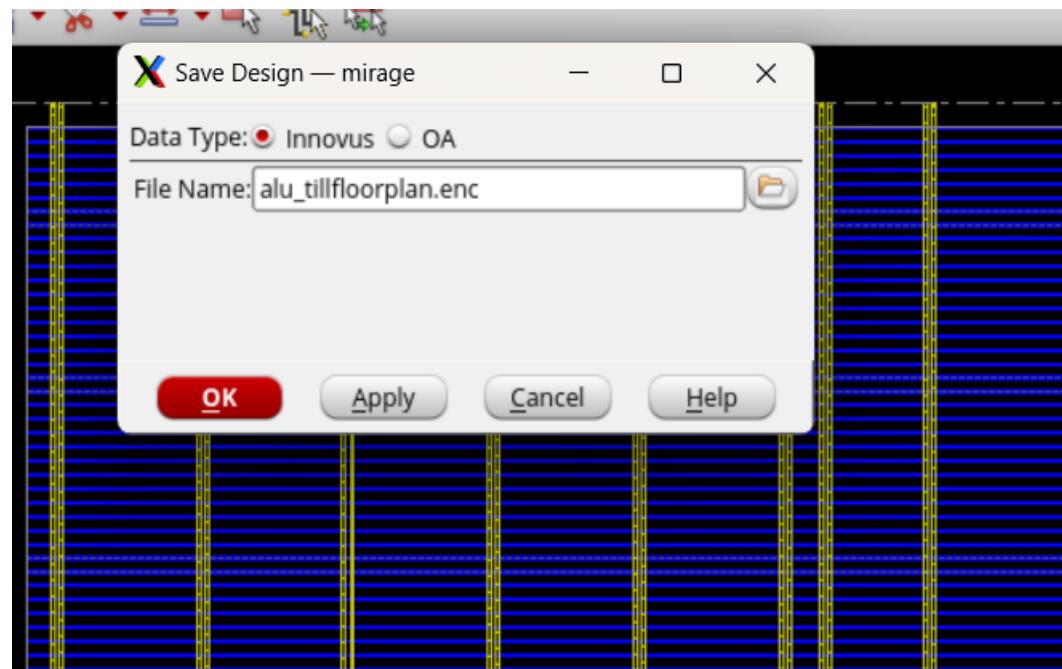


If any violations are there, start from beginning (floorplan step) with different values of height, width, power plan stripes etc.

If there are no violations, Go Ahead.

You can save your work till floorplan, like this:





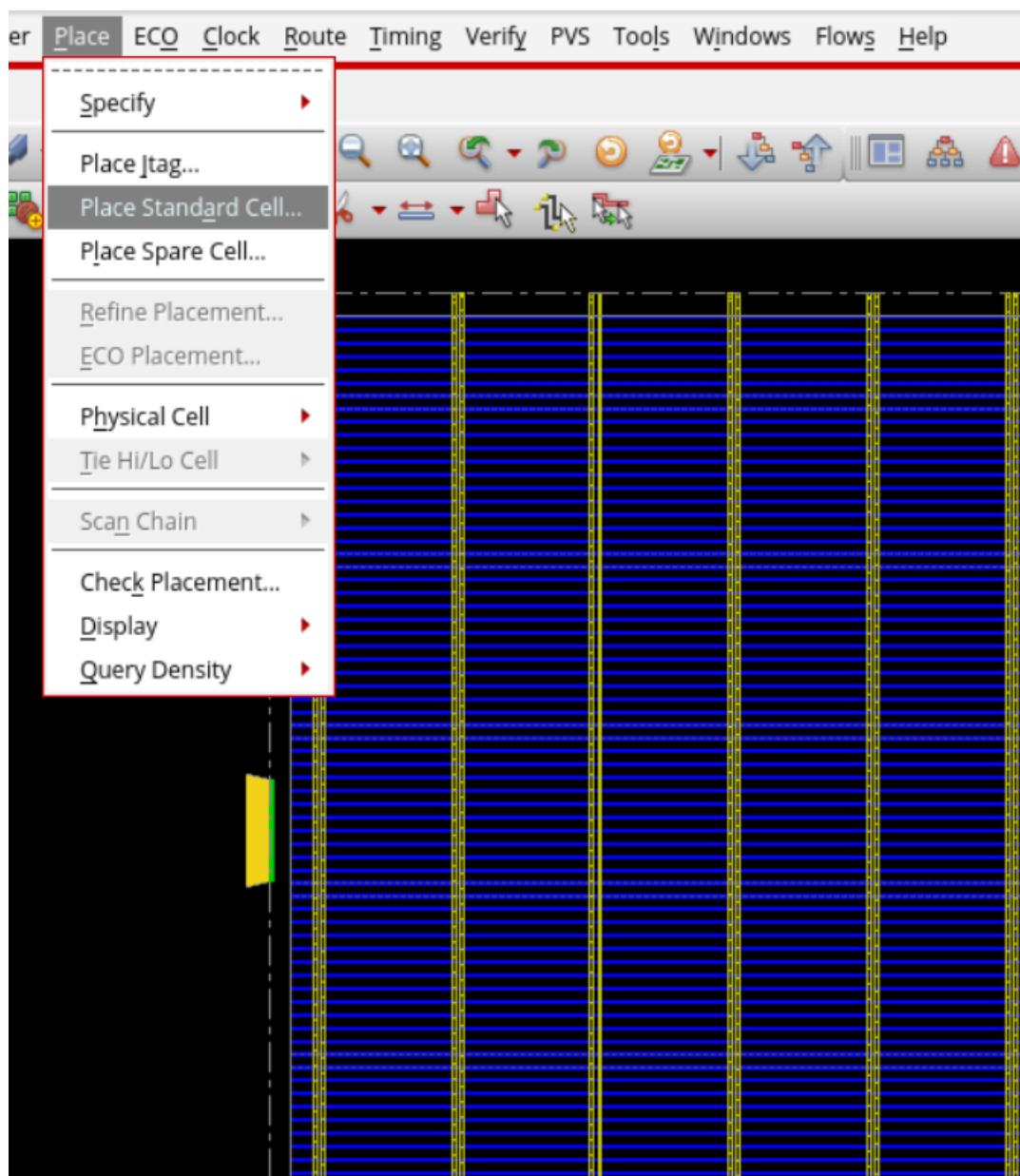
We can save work after each step, like floorplan, pre-cts, post-cts, post-route etc.

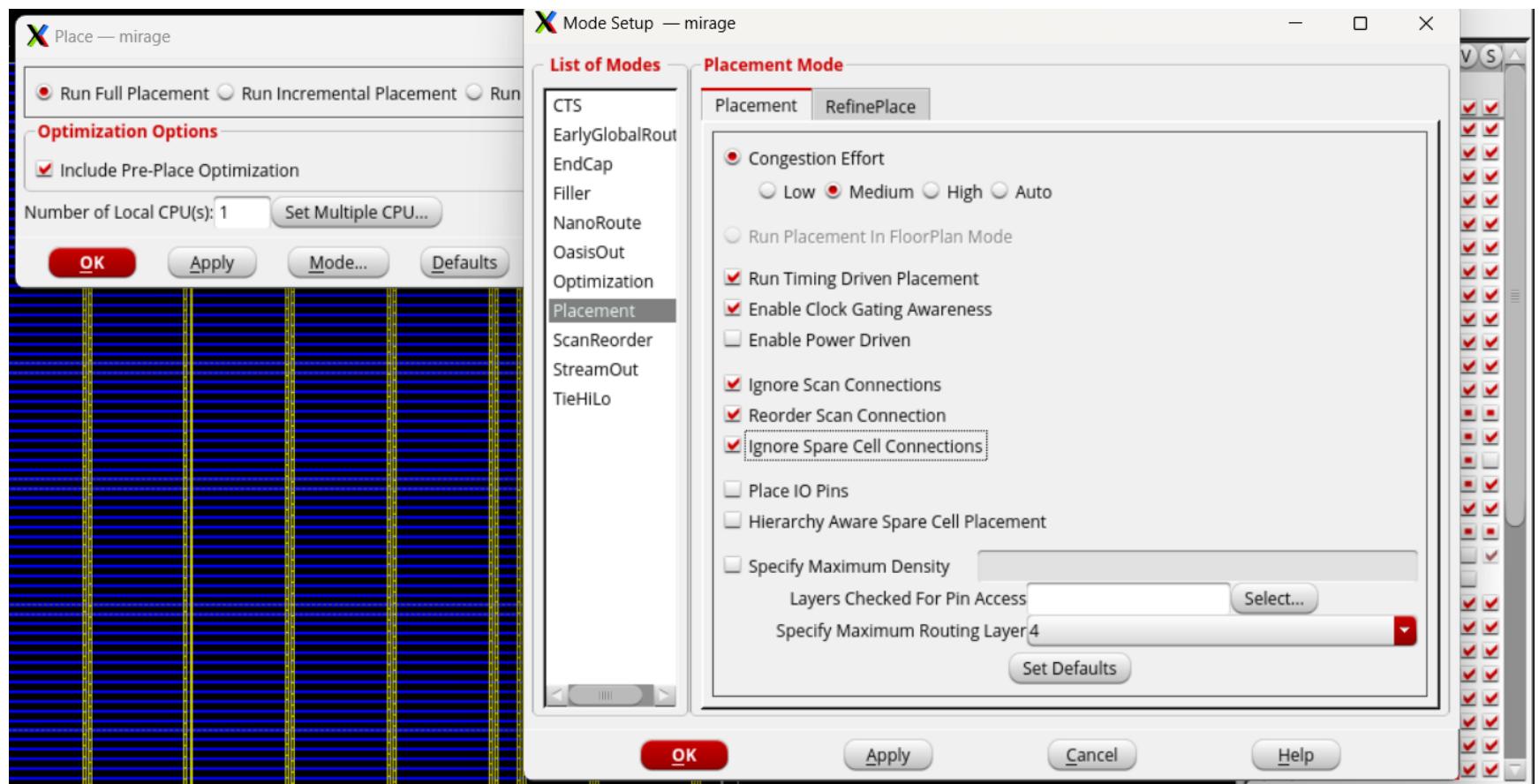
To open your saved work, after invoking in terminal, give this command:

```
innovus 1> source alu_tillfloorplan.enc
```

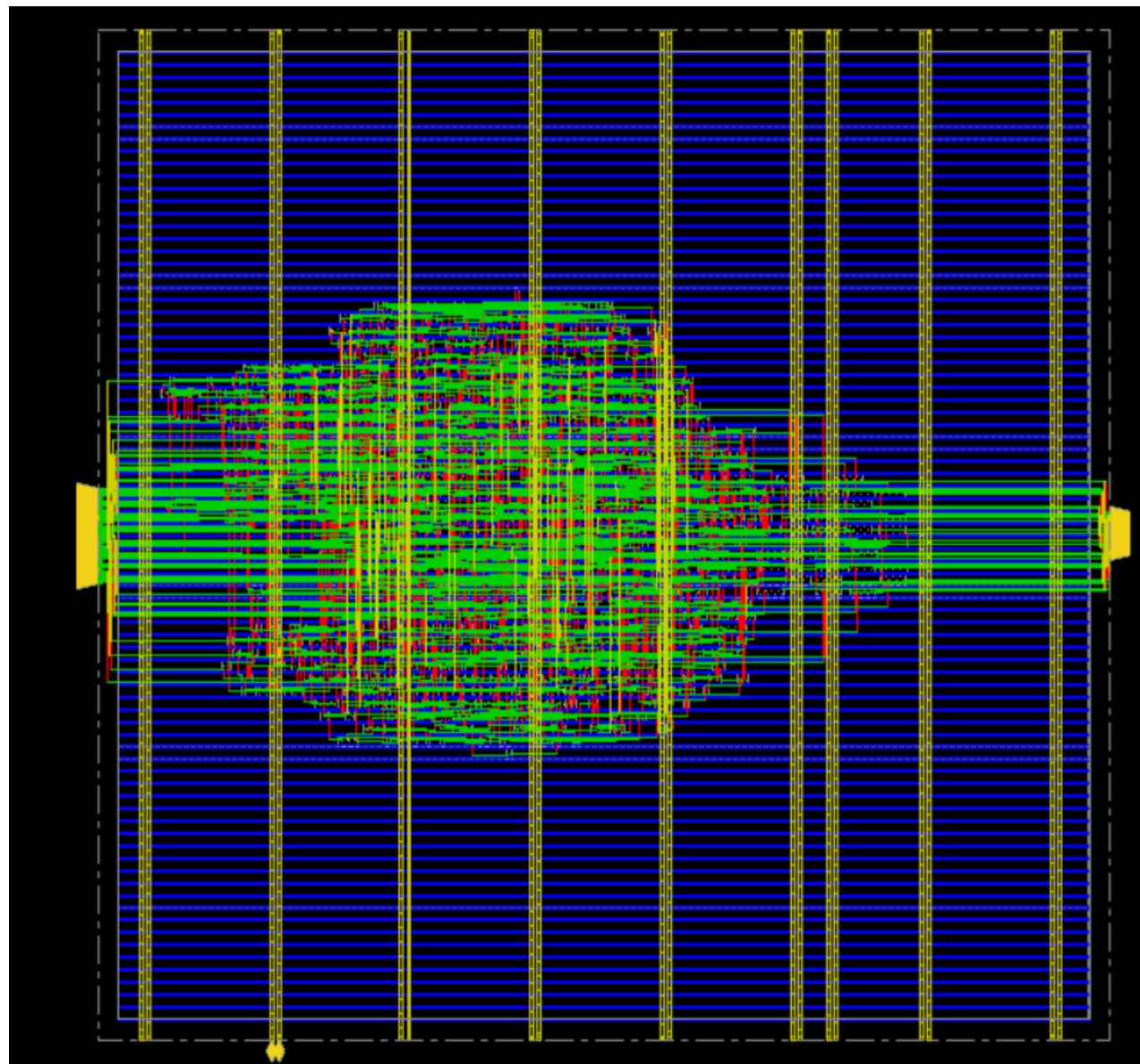
Till now, pre-placements steps are done.

✓ Step8 – Placement



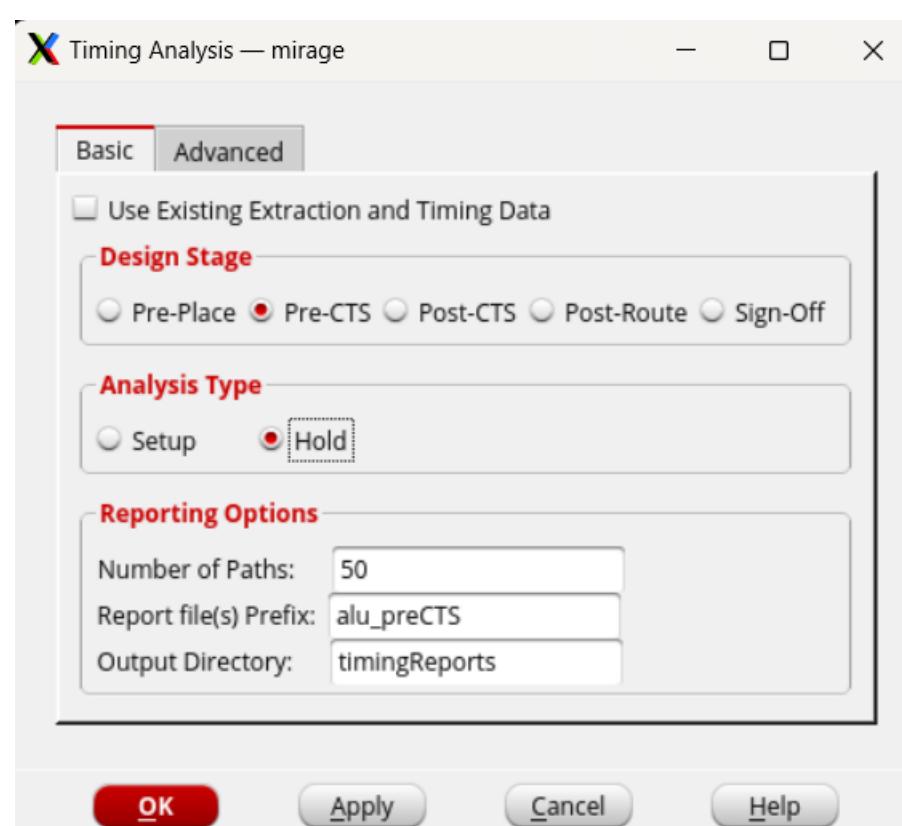
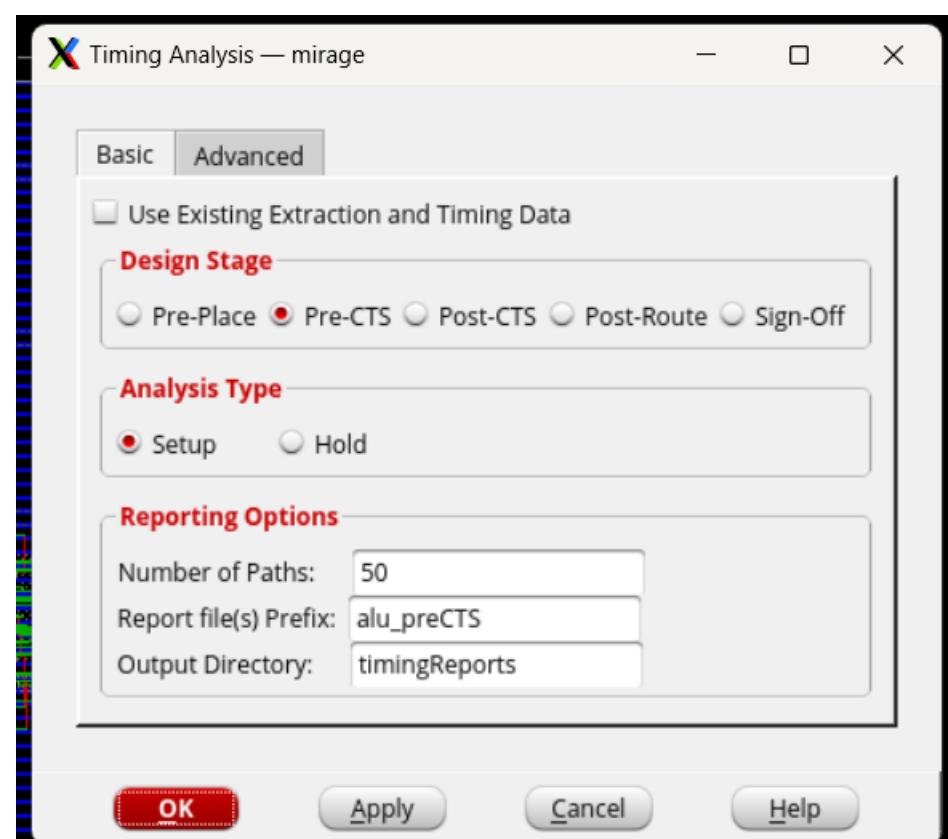
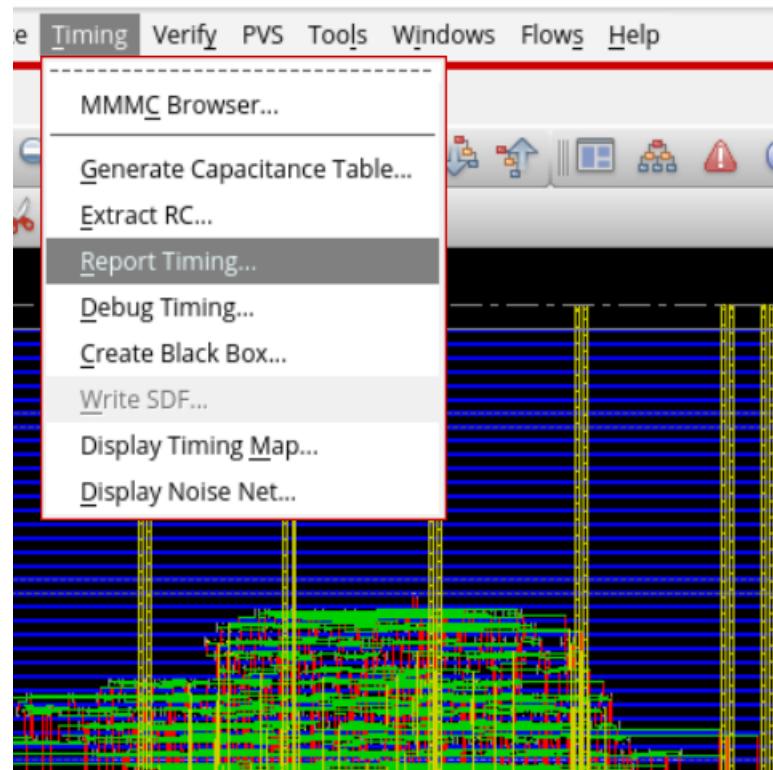


Placement output will be:



Perform pre-CTS timing:

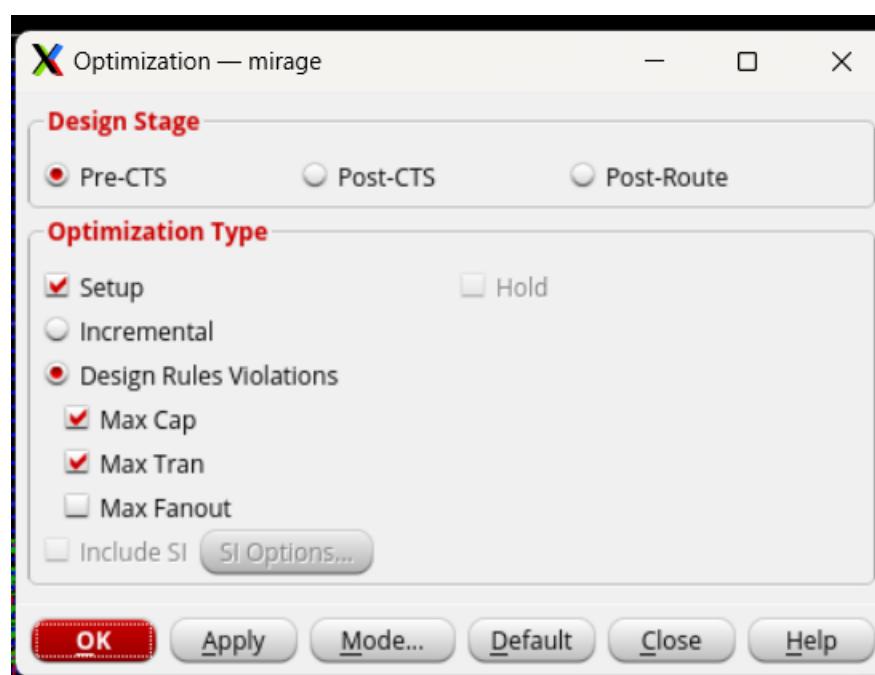
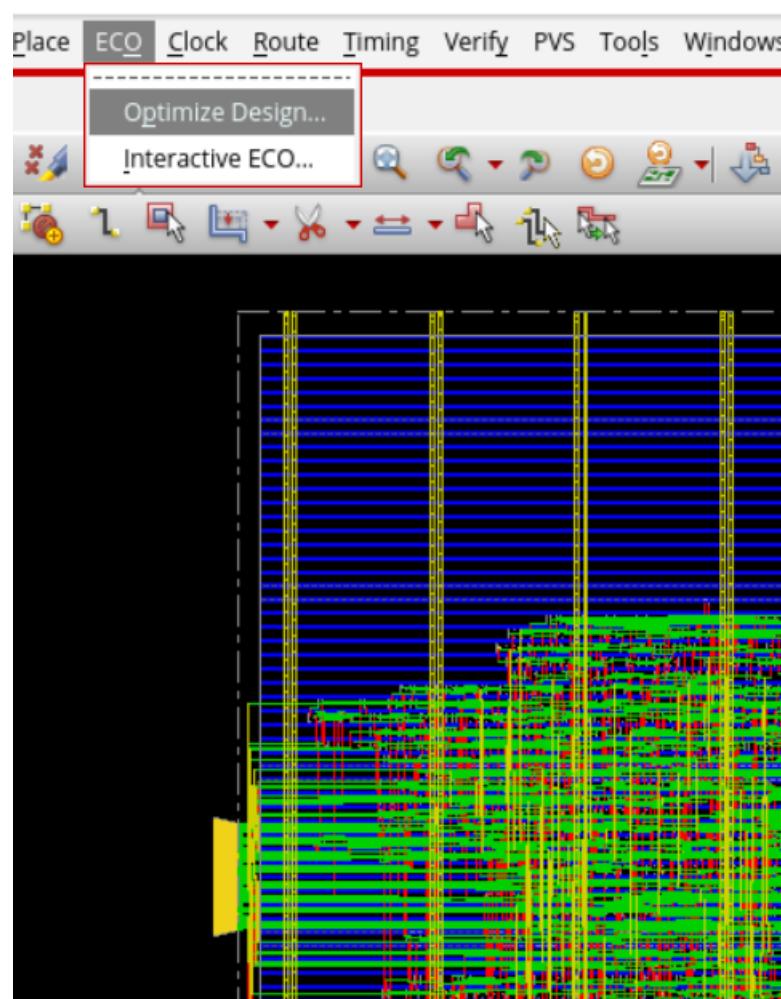
✓ Step8 - Pre- CTS



Check for setup and hold timings, if no any timing violations go for CTS.

If there's any timing violation, then optimize.

Pre- CTS Optimization:



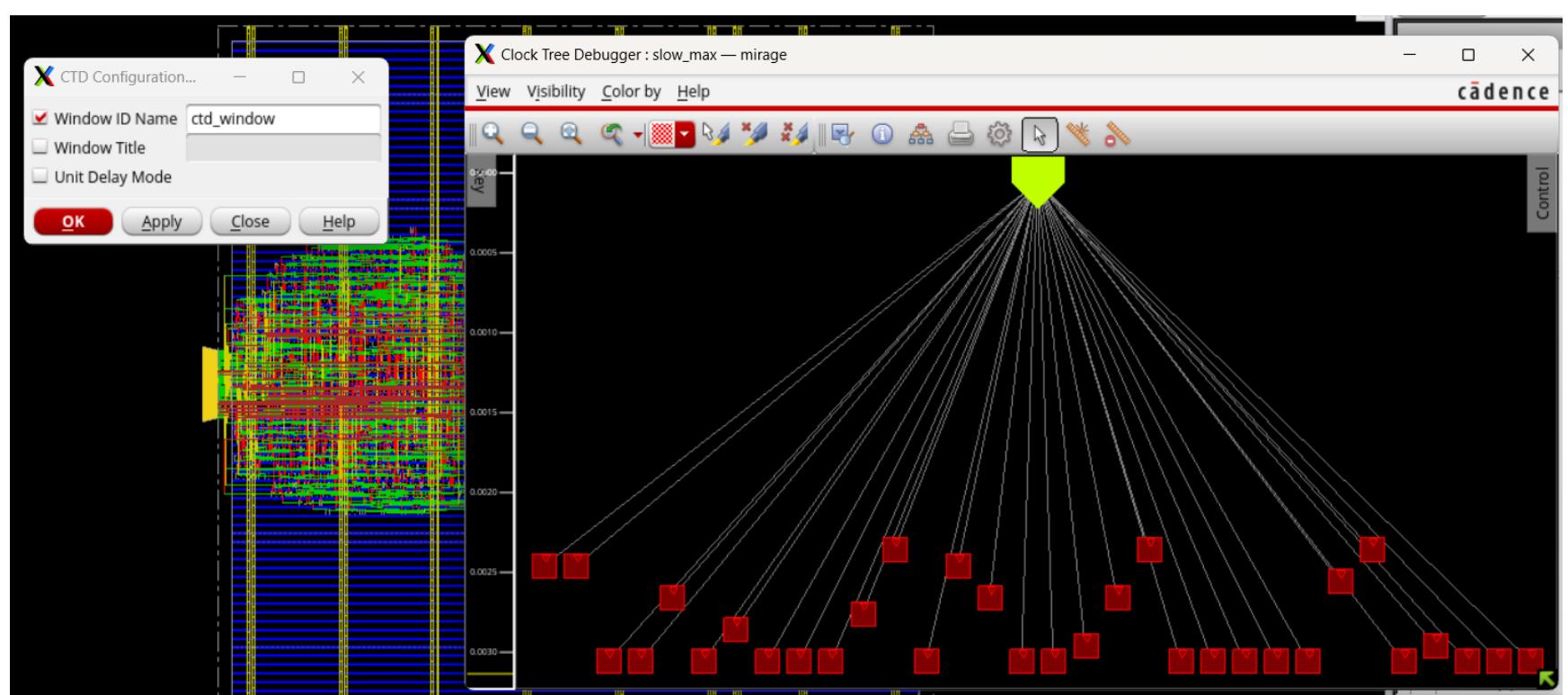
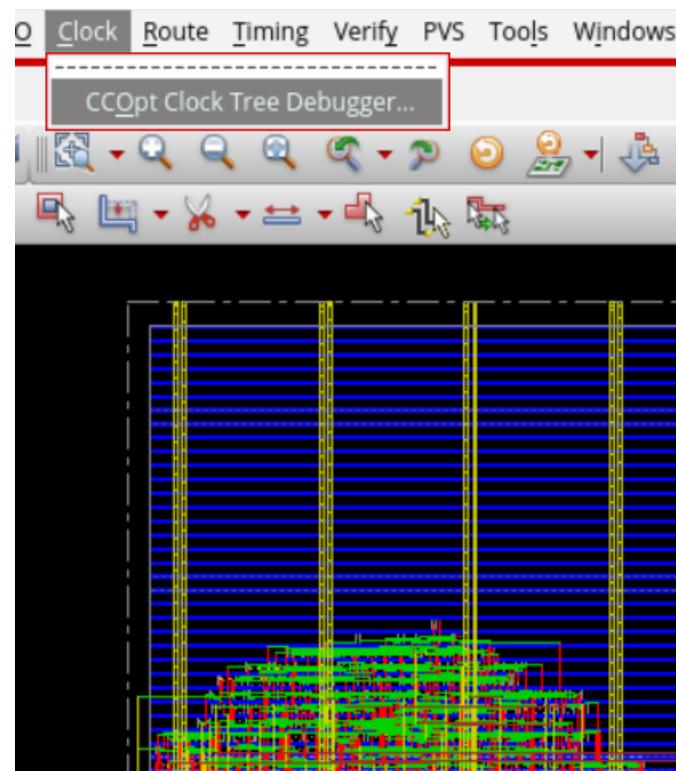
With clicking Ok, there will changes you can observe some changes happening and after done, do report timing step again.
If no violations, go for CTS.

✓ Step9 - CTS:

```
innovus 2>
innovus 2> source ../inputs/ccopt_spec_file.tcl
```

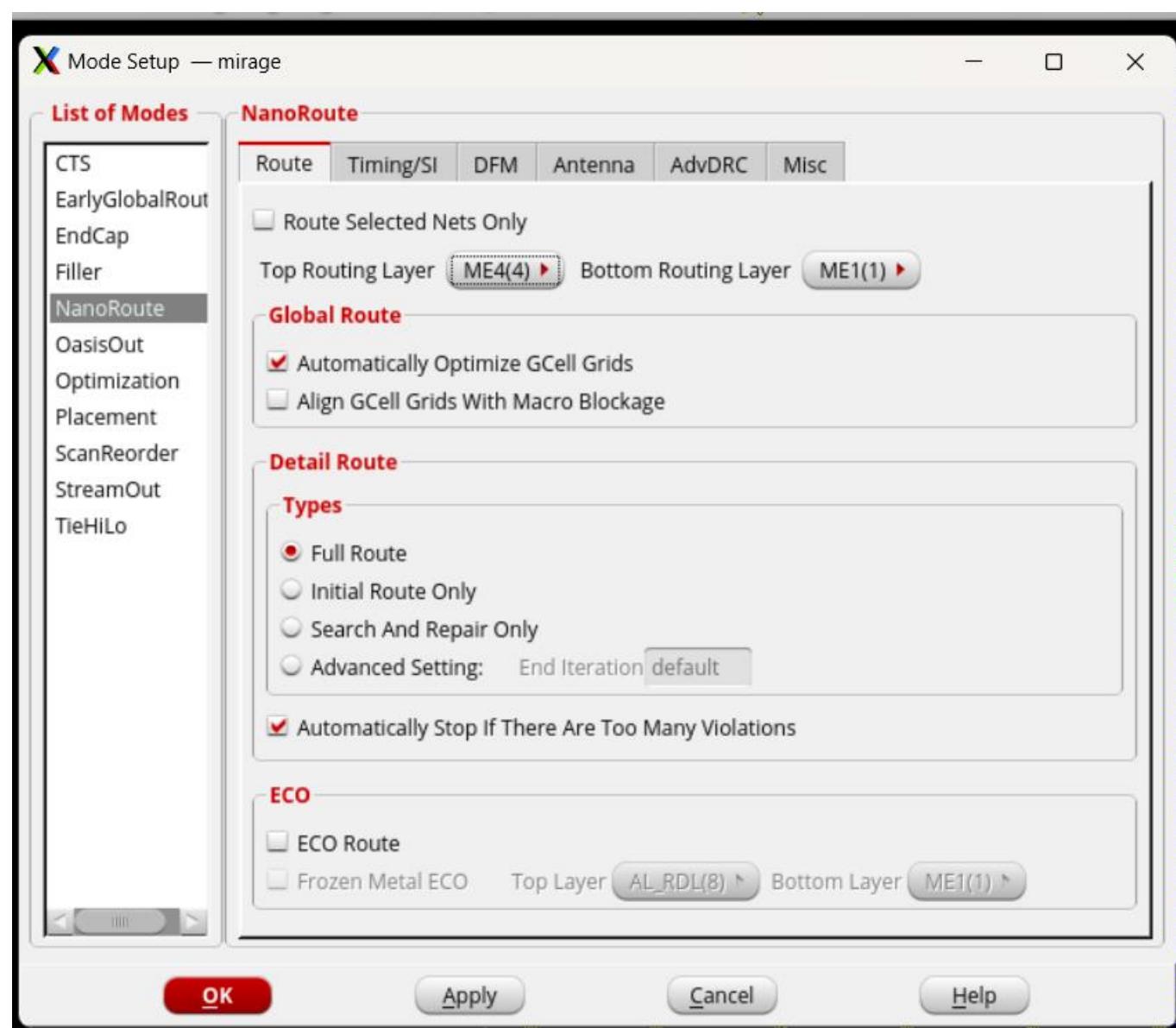
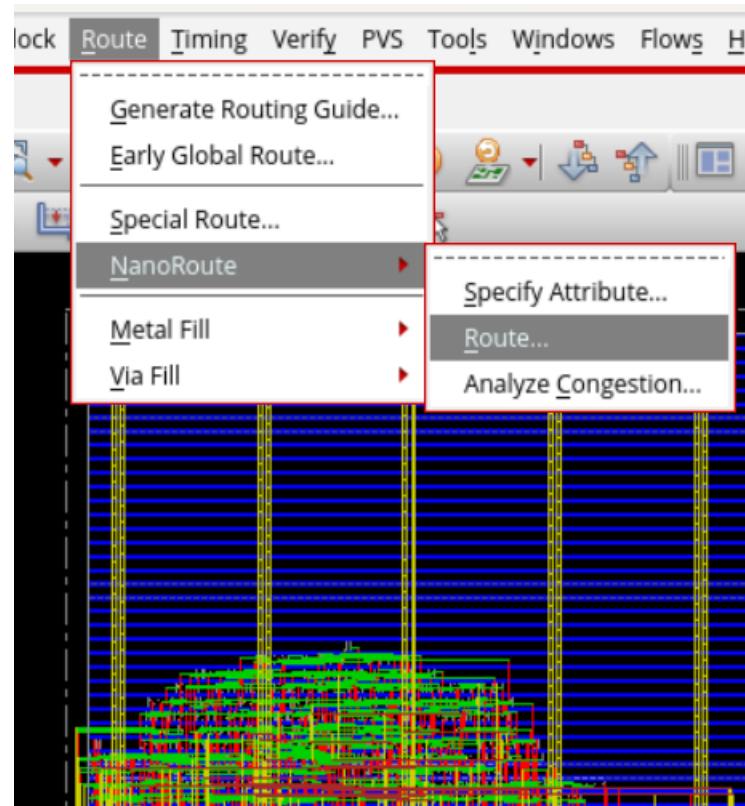
```
Checking clock tree convergence...
Checking clock tree convergence done.
innovus 3> ccopt_design -cts
```

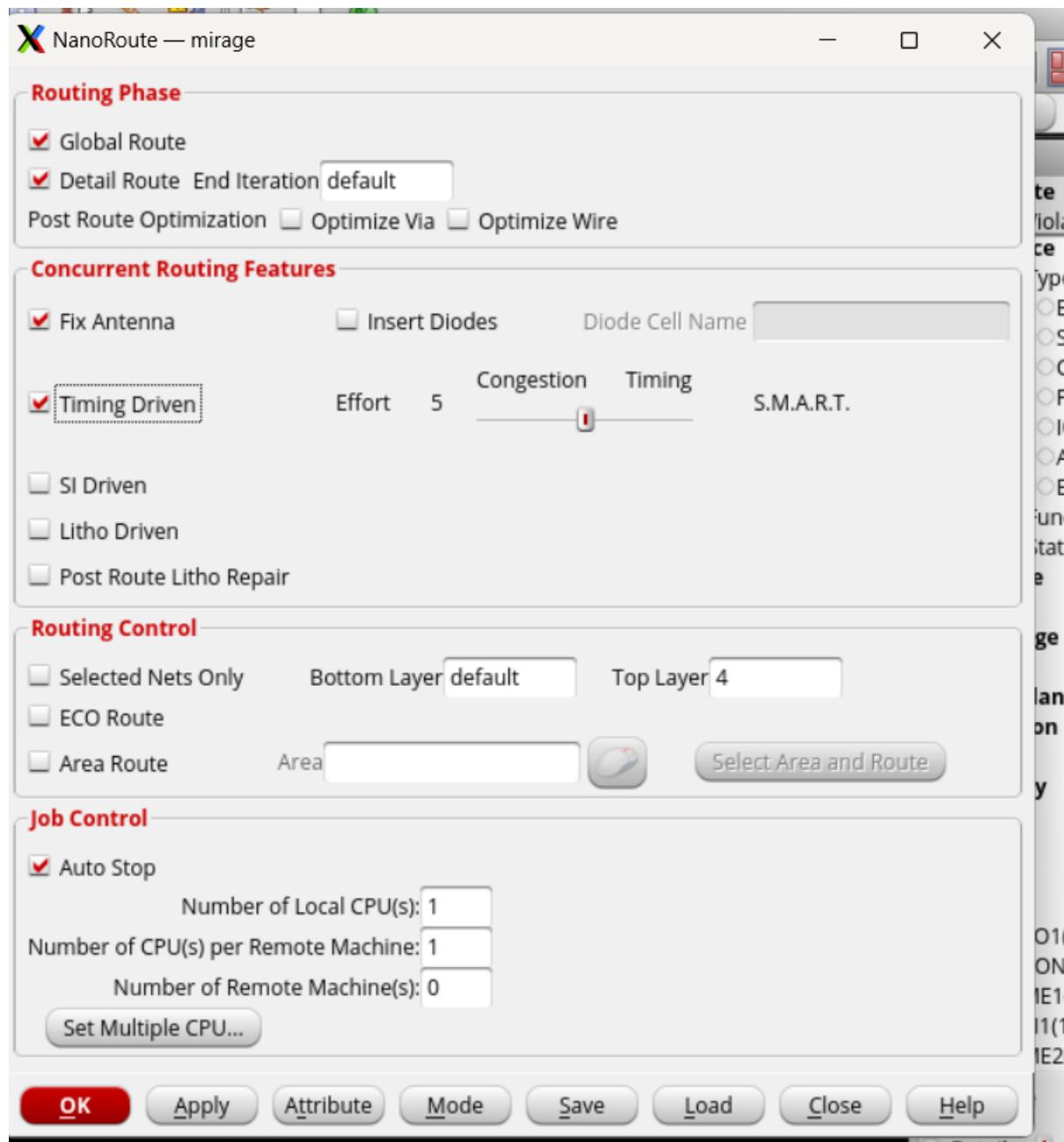
After entering those two commands, check whether clock tree has built or not:



Now, do report timing step again, but this time, select post-CTS.
If there are any timing violations, do ECO optimization by selecting post-CTS and repeat the report-timing step.
If no violations, we can go for next step: **ROUTE**.

✓ Step10 - Route:





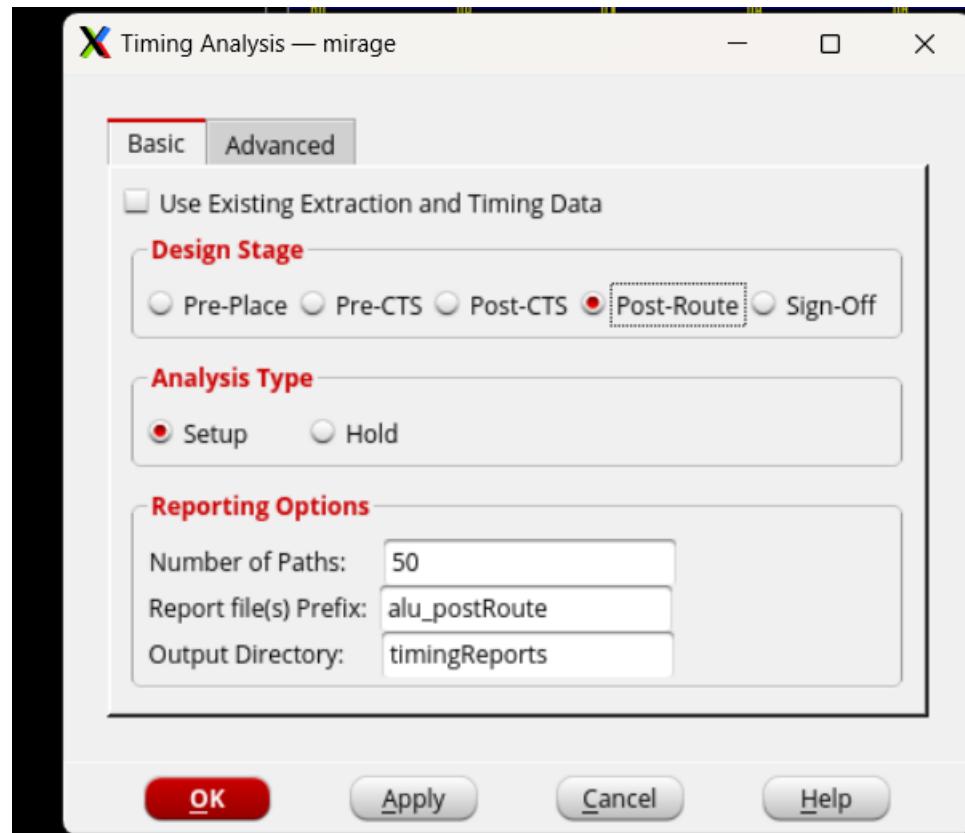
Clicking OK, routing will be done.

✓ Step11 - Post-route:

Before checking for timing, do this:

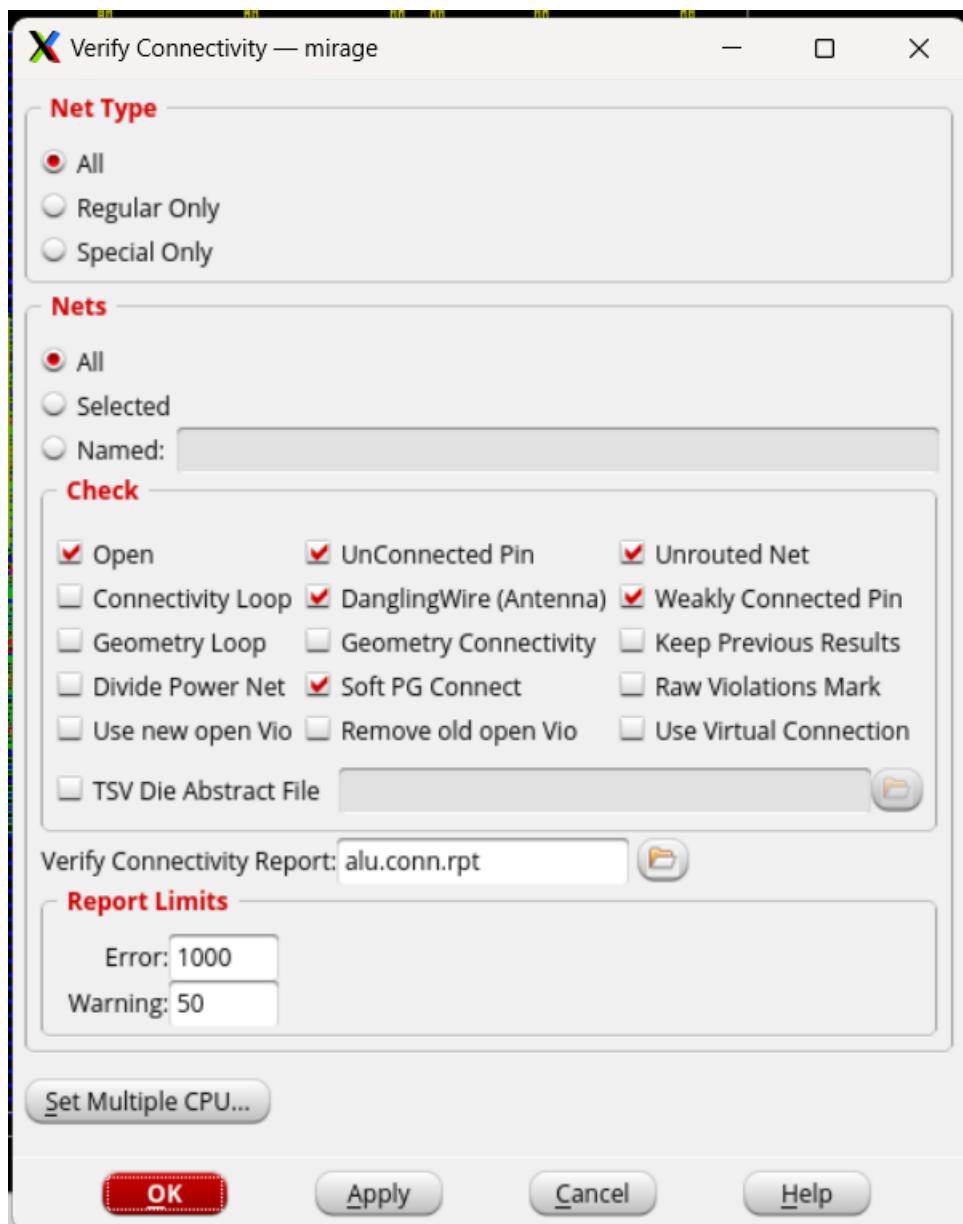
```
innovus 4> setAnalysisMode -analysisType onChipVariation
```

Then, do report timing again



Do it for Both for setup and hold.
If there are any violations, do **eco-optimization** step.

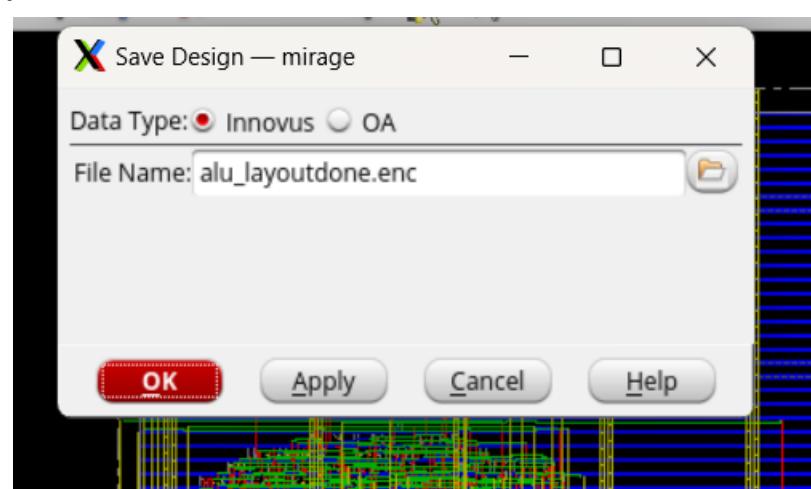
Verify DRC and Connectivity



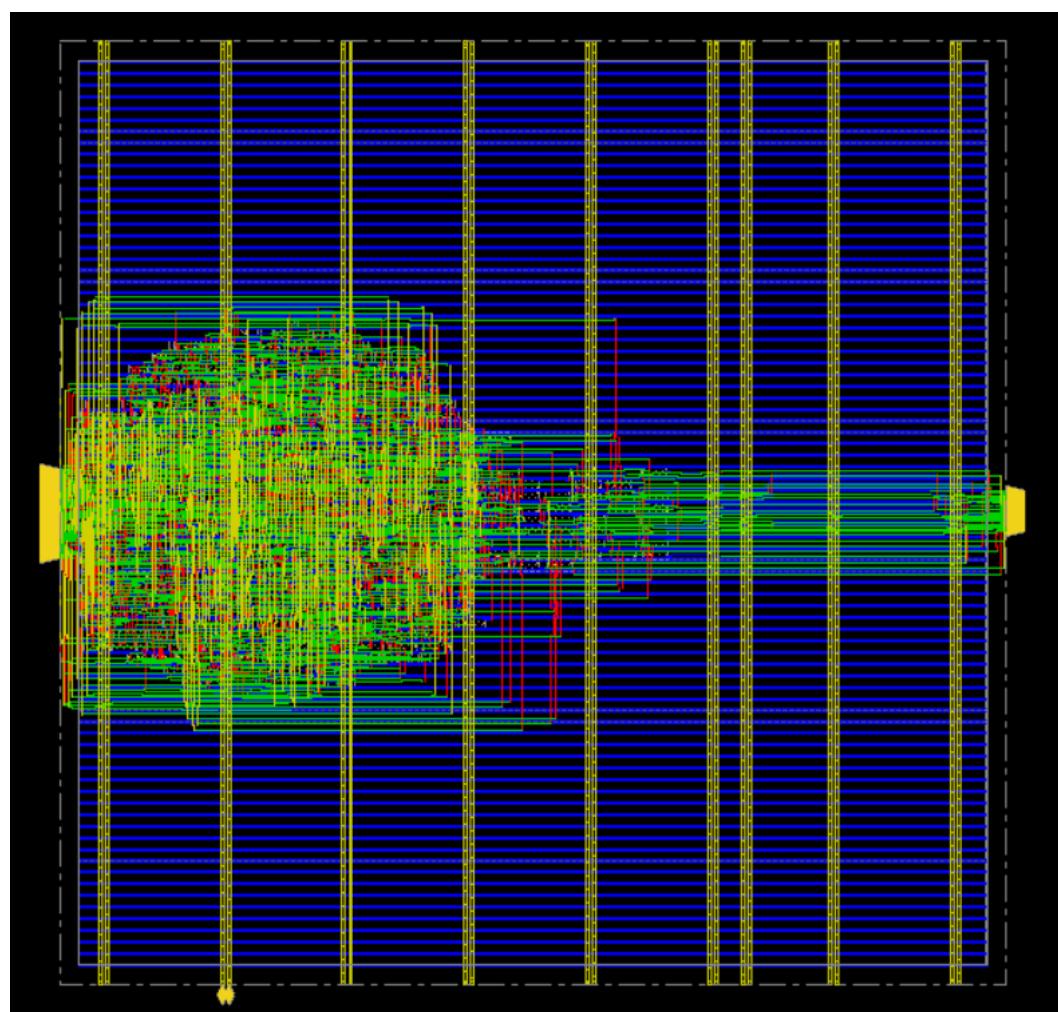
If timing, DRC and connectivity and everything is fine with 0 violations,

Layout is done.

Save layout work:



✓ Step12 - Layout:



Zoom in:



✓ Step13 – Layout outputs:

- For the final layout outputs, execute the following instructions sequentially.
 - streamOut alu_layout.gds
 - saveNetlist alu_postlayoutnetlist.v
 - extractRC
 - rcOut -spef alu_postlayoutspeffile.spef
 - write_sdf -recompute_parallel_arcs alu_postlayoutsdffile.sdf
 - defOut alu_postlayoutdeffile.def

you'll have all above outputs in work folder:

