ESE 566A Modern System-on-Chip Design, Spring 2017
Tutorial for Cadence Encounter

Electrical and System Engineering, Washington University

Revision: 01-25-2016

1. Introduction

We use Cadence Encounter for placing and routing standard cells, but also for power routing and clock tree synthesis. The Verilog gate-level netlist generated by Synopsys DC has no physical information: it is just a netlist, so the Cadence Encounter will first try and do a rough placement of all of the gates into rows on the chip. Cadence Encounter will then do some preliminary routing, and iterate between more and more detailed placement and routing until it reaches the target cycle time (or gives up). Cadence Encounter will also route all of the power and ground rails in a grid and connect this grid to the power and ground pins of each standard cell, and Cadence Encounter will automatically generate a clock tree to distribute the clock to all sequential state elements with hopefully low skew. The automated flow for place-and-route is much more sophisticated compared to what we did in the previous tutorial.

2. Login to the Linux Lab server

Detailed explanation is in *ese566-linux-tutorial.pdf*

3. Getting prepared using *Design Compiler*

In the terminal, change path to your project directory, and compile your code using *design compiler.*

(If you don't know how to use *design compiler*, look at *ese566-dc-tutorial.pdf*)

4. Encounter

4.1 Start Encounter

Download the "*Default.view*" and modify it according to your design for MMMC view definition file later. Please specify the name of the *.sdc* file used in your synthesis script. It should be the *.sdc* file generated by *Design Compiler* in the previous design step.

```
1 # Version:1.0 MMMC View Definition Fi
2 # Do Not Remove Above Line
3 create_library_set -name vtvt_tsmc180 -timing {/project/linuxlab/cadence/vendors/VTVT/vtvt_tsmc180/Synopsys_Libraries/libs/vtvt_tsmc180.lib}
4 create_constraint_mode -name constraint_rule -sdc_files {Counter.sdc}
5 create_delay_corner -name vtvt_tsmc180 -library_set {vtvt_tsmc180}
6 create_analysis_view -name constraint_rule -delay_corner vtvt_tsmc180 -constraint_mode {constraint_rule}
7 set_analysis_view -setup {constraint_rule} -hold {constraint_rule}
8
```

This is a file generated by Design Compiler.
(You need to change it according to your design)

Figure 4.1 Default.view

Then execute the following command in the shell to start *encounter*

% encounter

You will get the following printout in the terminal and a window pops up which looks like the one as below. This window will lead to the design window.

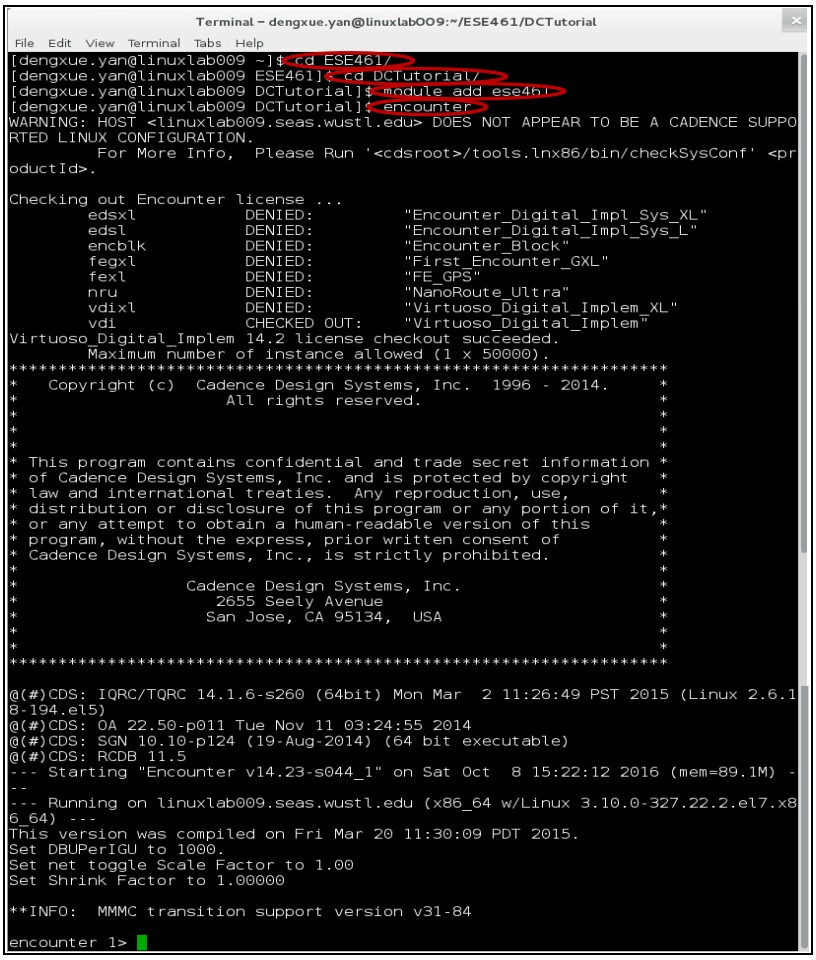

Figure 4.2 Command and the printout of “encounter”

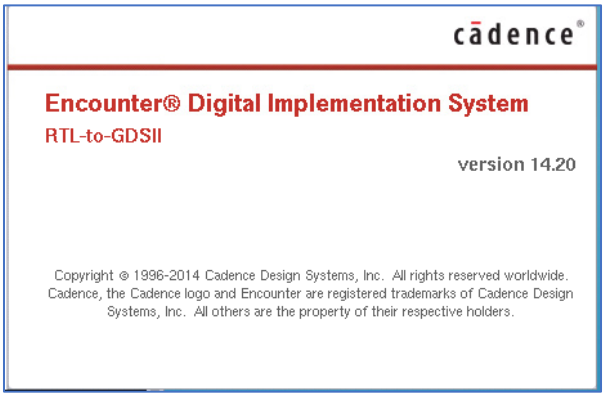

Figure 4.3 The window popped out when execute “*encounter*”

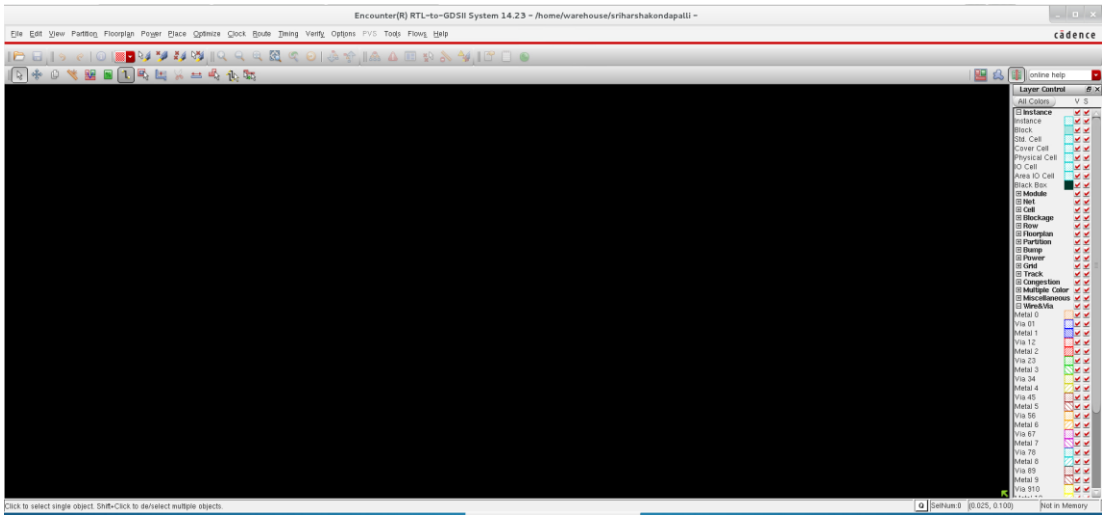

Figure 4.4 The design window of "*encounter*"

4.2 Import Design

Go to "*File->Import Design*" and add files to import your *netlist* file, and you have to:

- Provide synthesized Verilog (*.syn.v*) which is the output netlist file generated by Design Compiler in the previous design step.
- Provide path for the LEF file from technology library. Set path to *"/project/linuxlab/cadence/vendors/VTVT/vtvt_tsmc180/vtvt_tsmc180_lef*" and select the "*vtvt_tsmc180.lef*" file from the folder.
- Provide the supply nets as *vdd!* and *gnd!* in the appropriate boxes as below. These represent the global power supply and ground net in your design.
- Load the "*Default.view*" edited above for MMMC view definition file

Click "OK" to finish

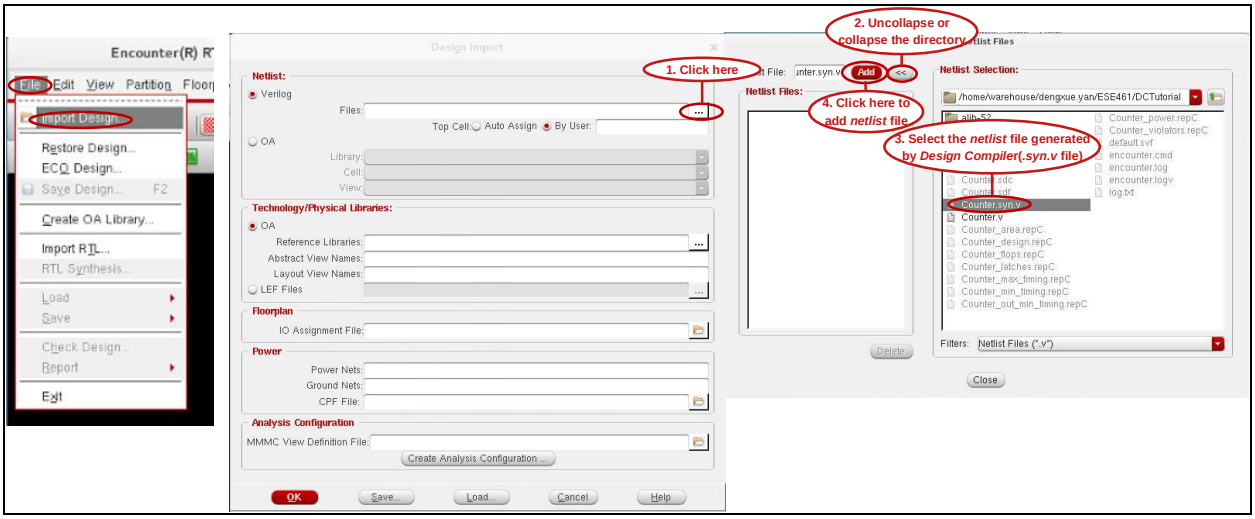

Figure 4.4 Import netlist

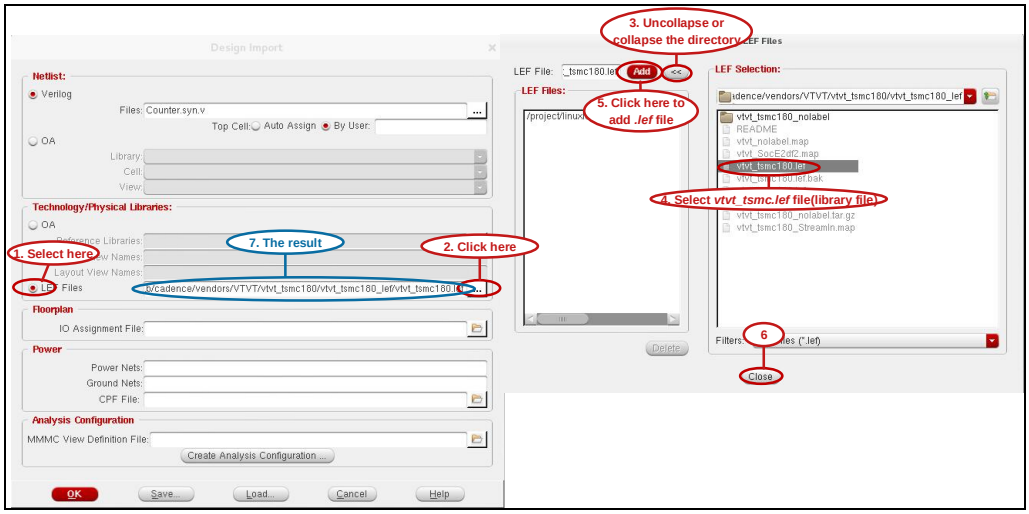

Figure 4.5 Import LEF file

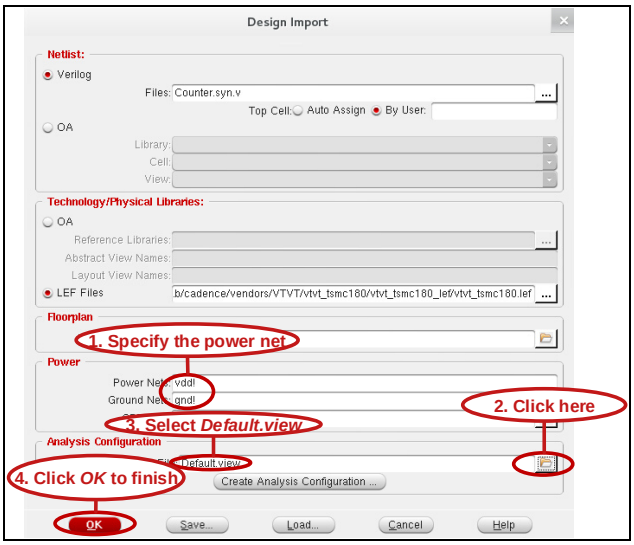

Figure 4.6 Add power net and MMMC view file

Then, you will be seeing a blank screen with horizontal lines on your main window (layout window). Press “f” for fit screen view of layout.

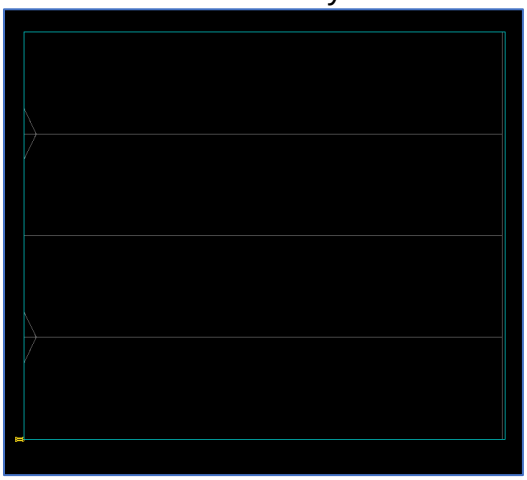

Figure 4.7 The initial layout window

Specify "*floorplan*" options in the "*Floorplan->Specify* Floorplan" and set the values as figure 4.8.

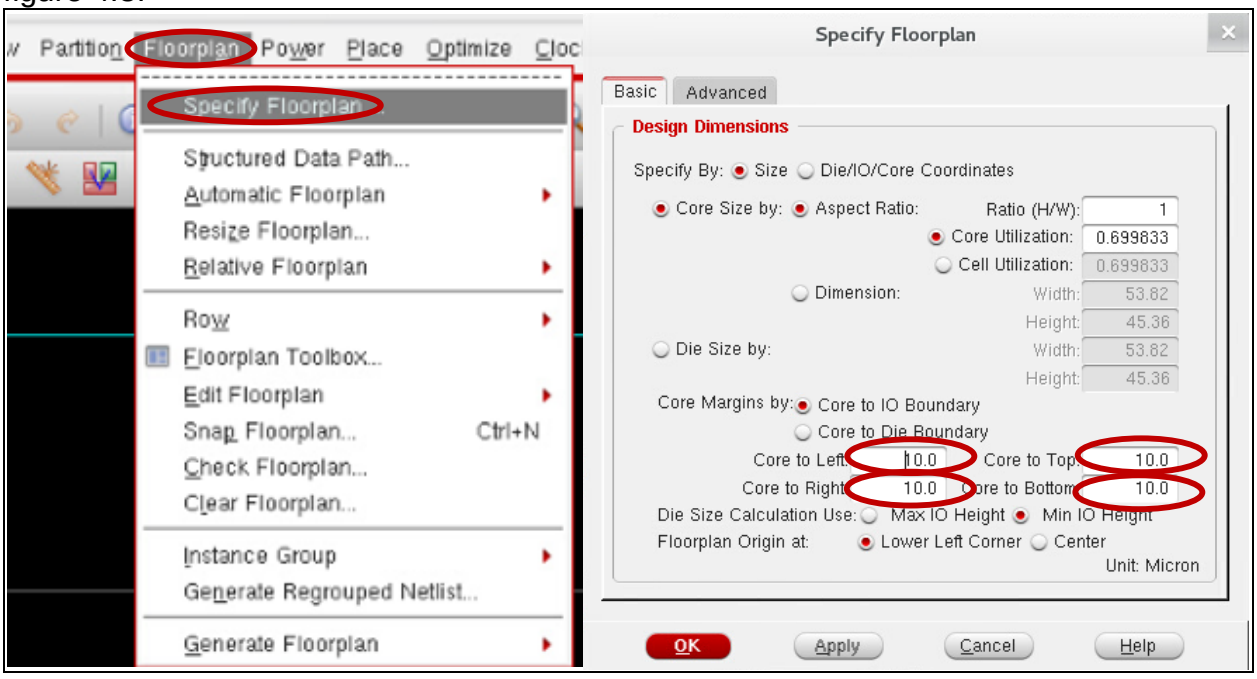

Figure 4.8 Specify Floorplan

Set the global nets *vdd!* and *gnd!* at "*Power->Connect Global Nets*".

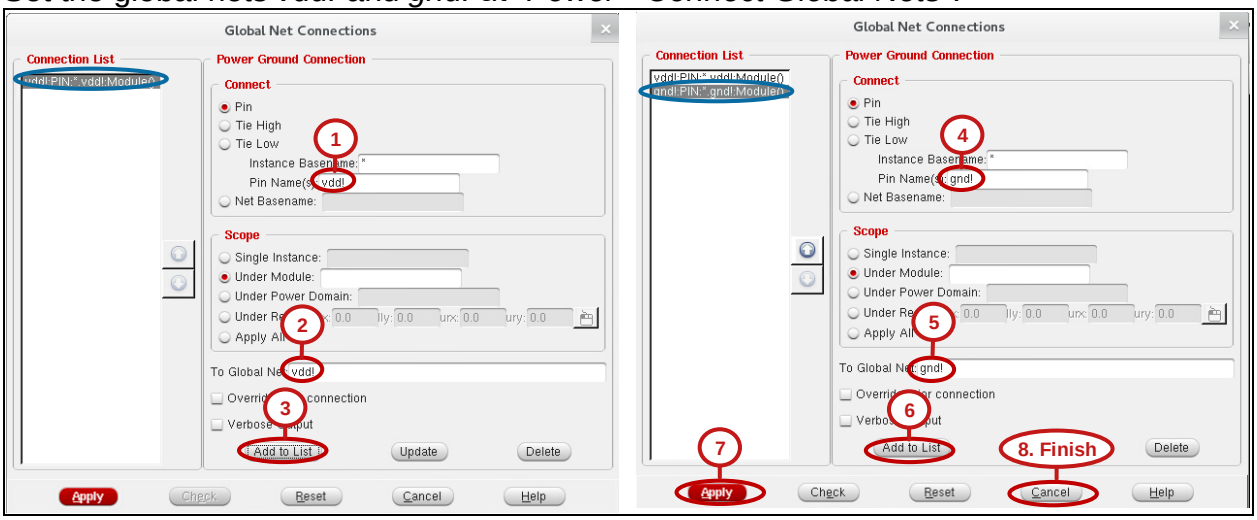

Figure 4.9 Set the global nets

Go to "*Power->Add Ring*" and set Net pins to *vdd!*, *gnd!*. And you change width and spacing of power ring.

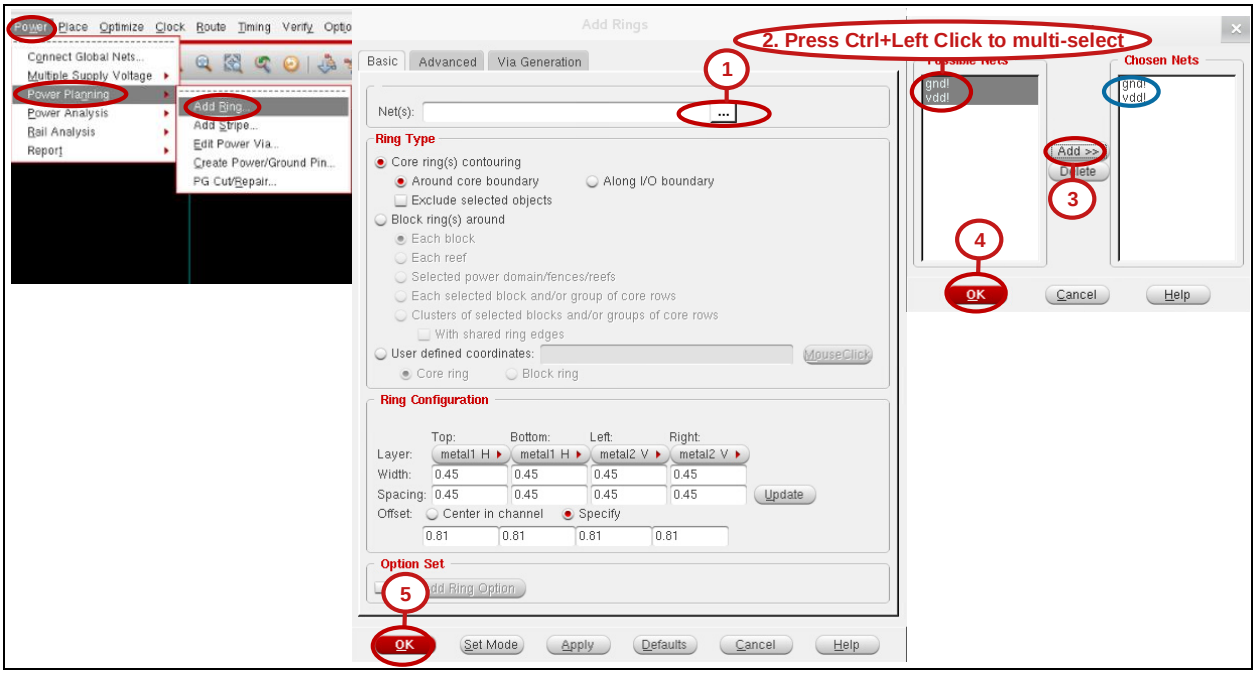

Figure 4.10 Add power ring

We can also edit the pin placement in our design. Go to "*Edit->Pin Editor*". Group some pins and you can place them on left or right side of the design. If you don't specify *Encounter* will be taking appropriate placement during optimization. Once done click "*OK*"

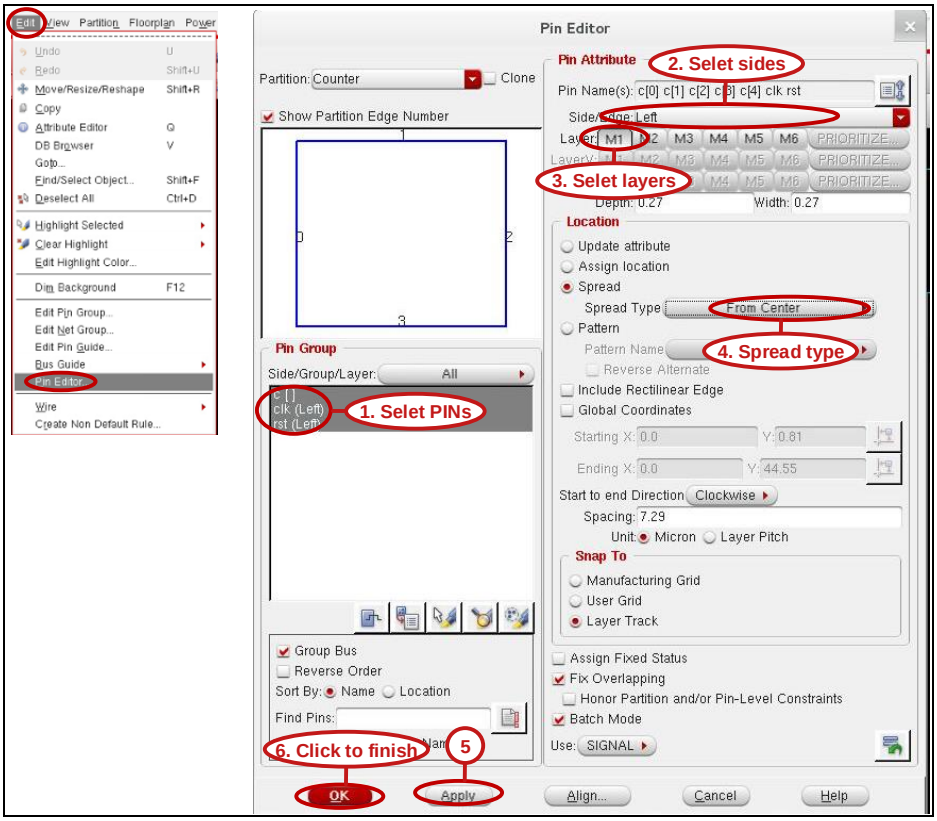

Figure 4.11 Pin Editor

Watch how your layout is changing while you are setting the parameters in the GUI.

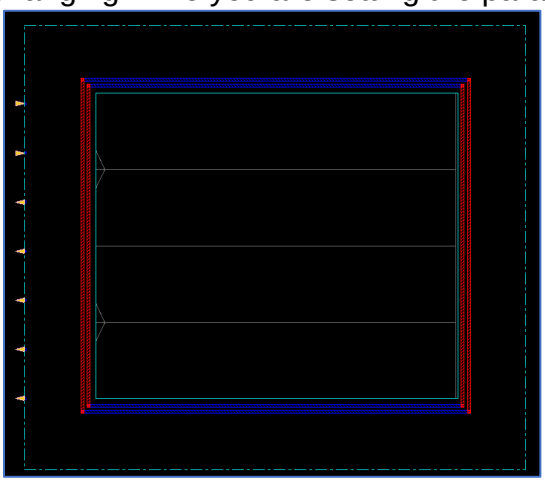

Figure 4.12 The layout view after edit pins

Place the *vdd!* and *gnd!* in the design. Go to "*Route->Special Route*"

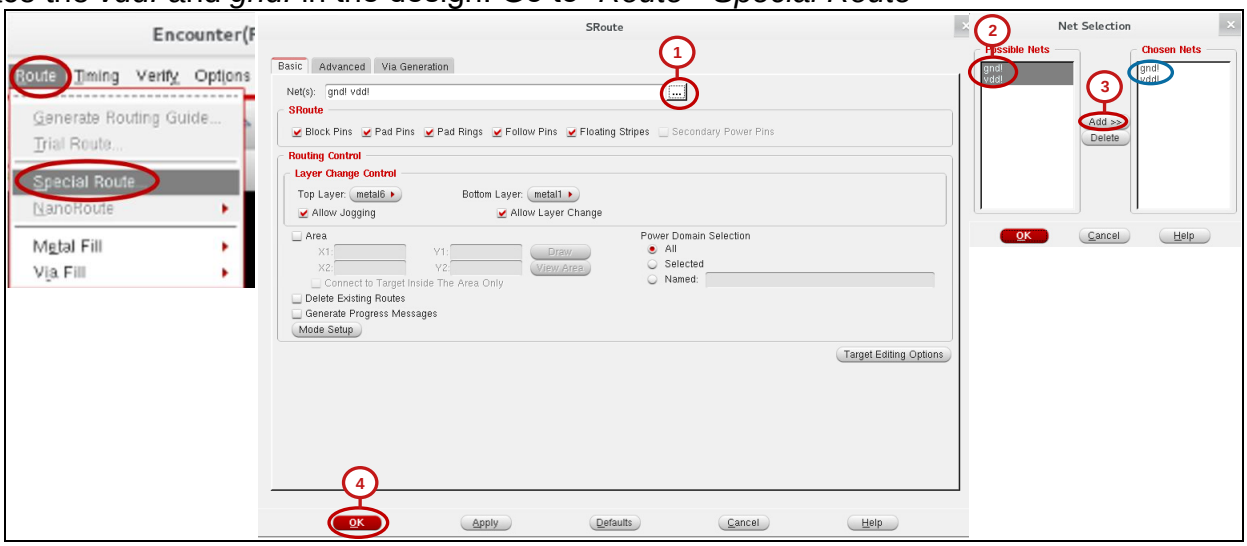

Figure 4.13 Place vdd! and gnd!

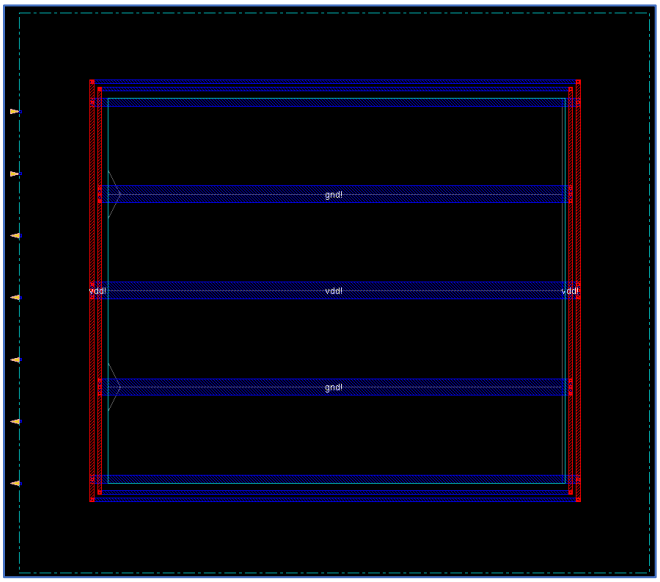

Figure 4.14 The layout after place vdd! and gnd!

Now we are all set to place the design. And then we go to "*Place->Place Standard Cell*". Click OK.

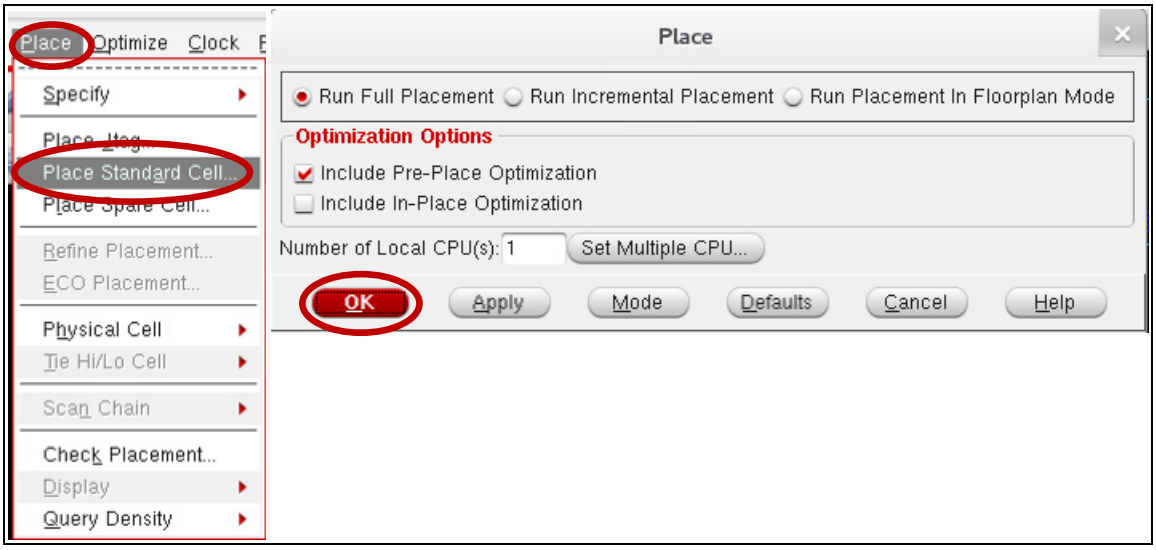

Figure 4.15 Place standard cell

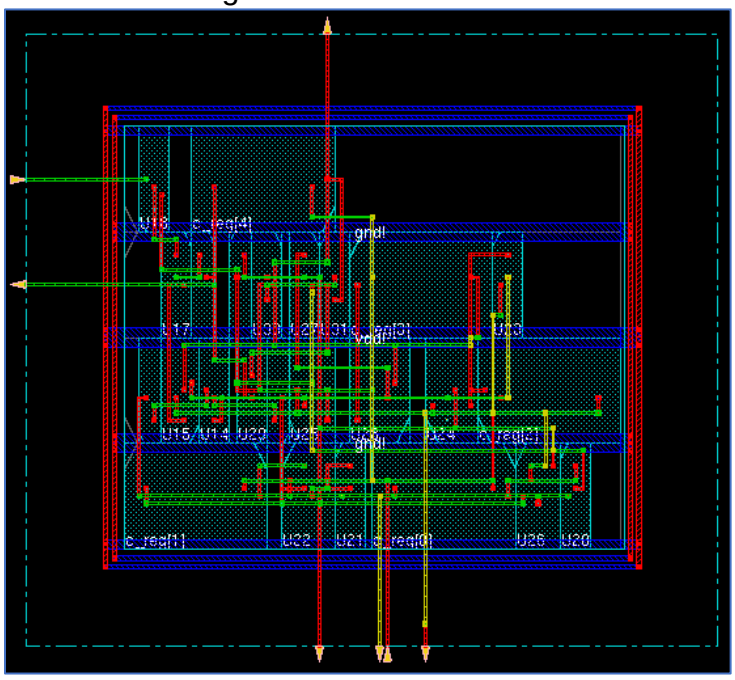

Figure 4.16 The result after place standard cell

4.3 Clock tree synthesis

- Download *Clock.ctstch* and place it in working directory.
- Download *Clock.tcl* and place it in working directory.
- Open *Clock.tcl* using *gedit* and copy the content line by line to the *encounter terminal* and execute it.

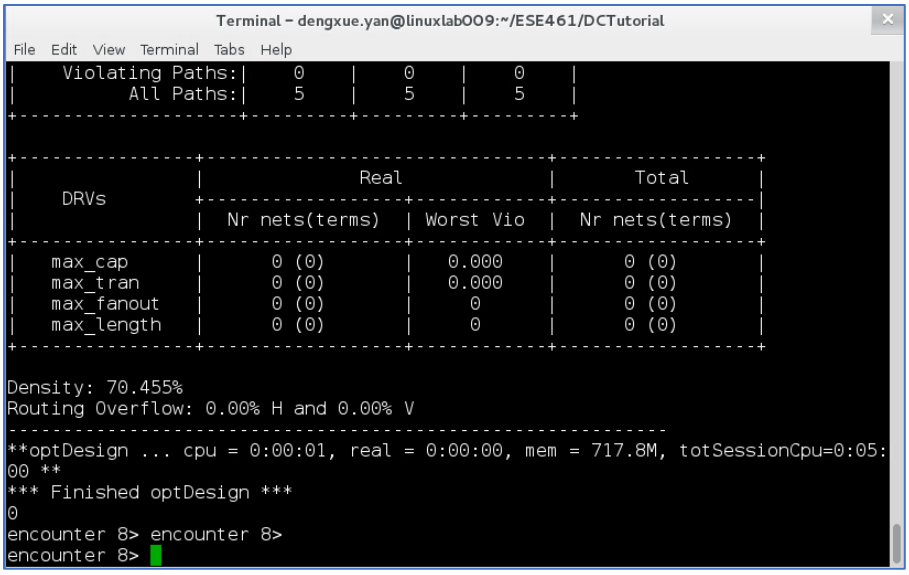

Figure 4.17 Partial of clock tree synthesis commands and results

4.4 NanoRoute

As part of routing go to "*Route->NanoRoute->Route*". And click OK.

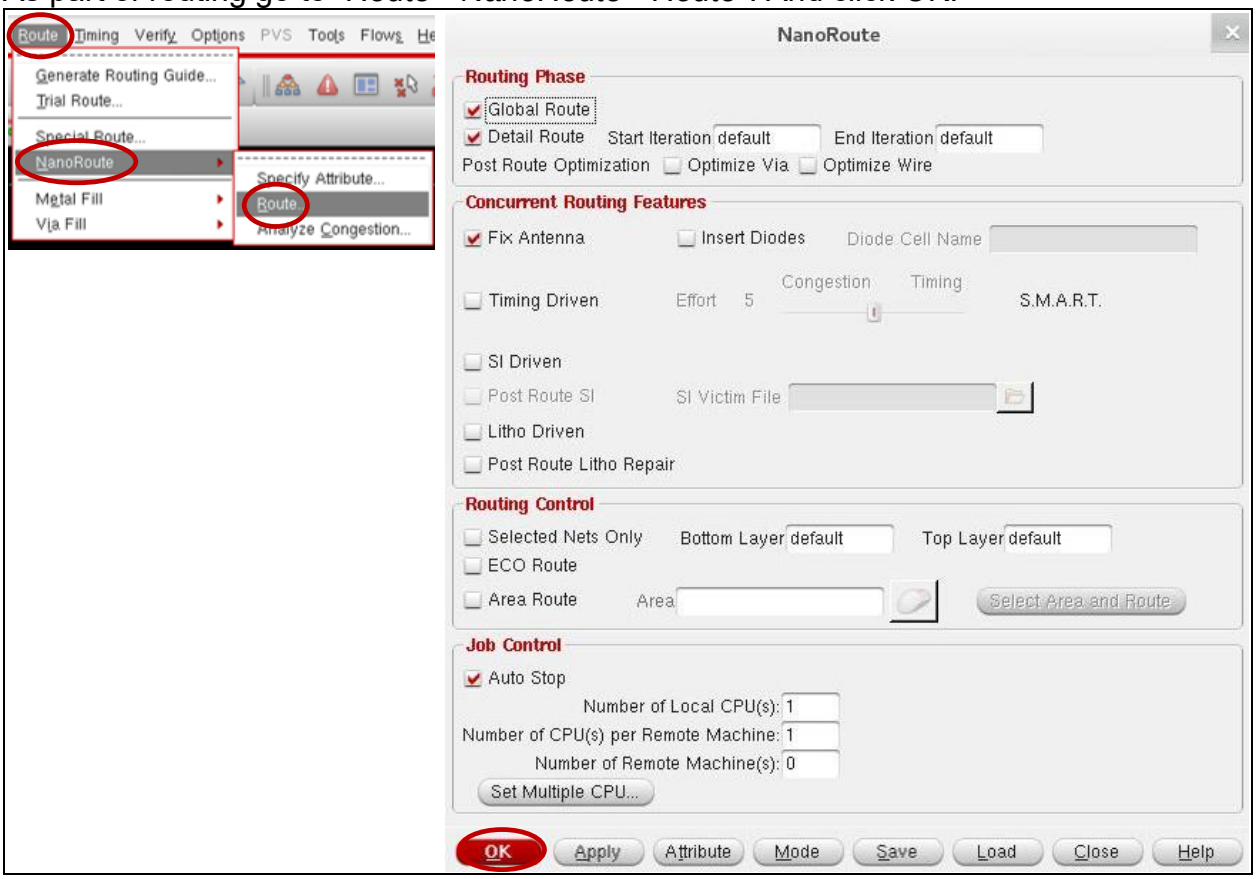

Figure 4.18 Nano route

You will see that your design got placed in the layout and all the interconnections are done.

4.5 Add Filler.

"*Place->Physical Cell->Add Filler*". Select "filler" cell name and click OK.

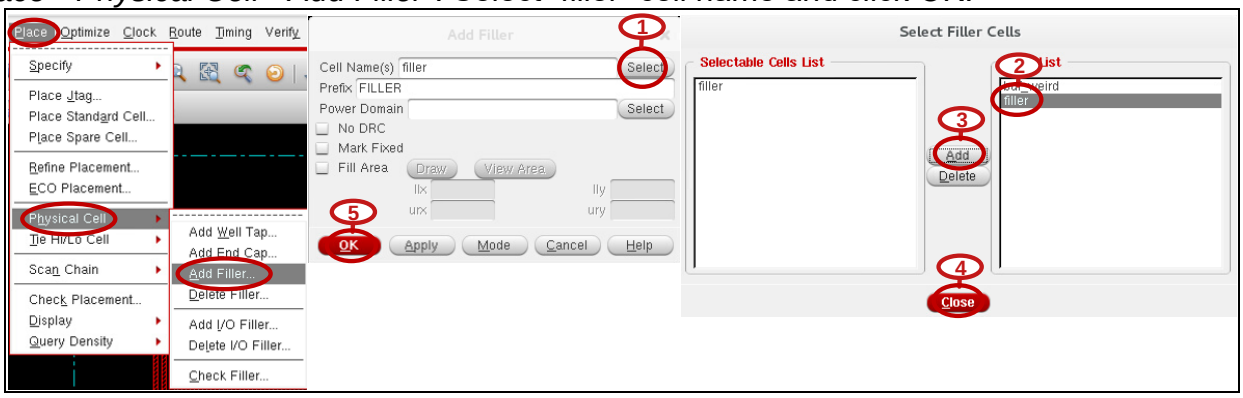

Figure 4.19 Add Filler

4.6 Verification

Once Place and Route is done:

- Go to "Verify->Verify Geometry". Click OK.
- Go to "Verify->DRC". Click OK
- Go to "Verify-> Connectivity". Click OK

Check the terminal every time for any violations and warnings.

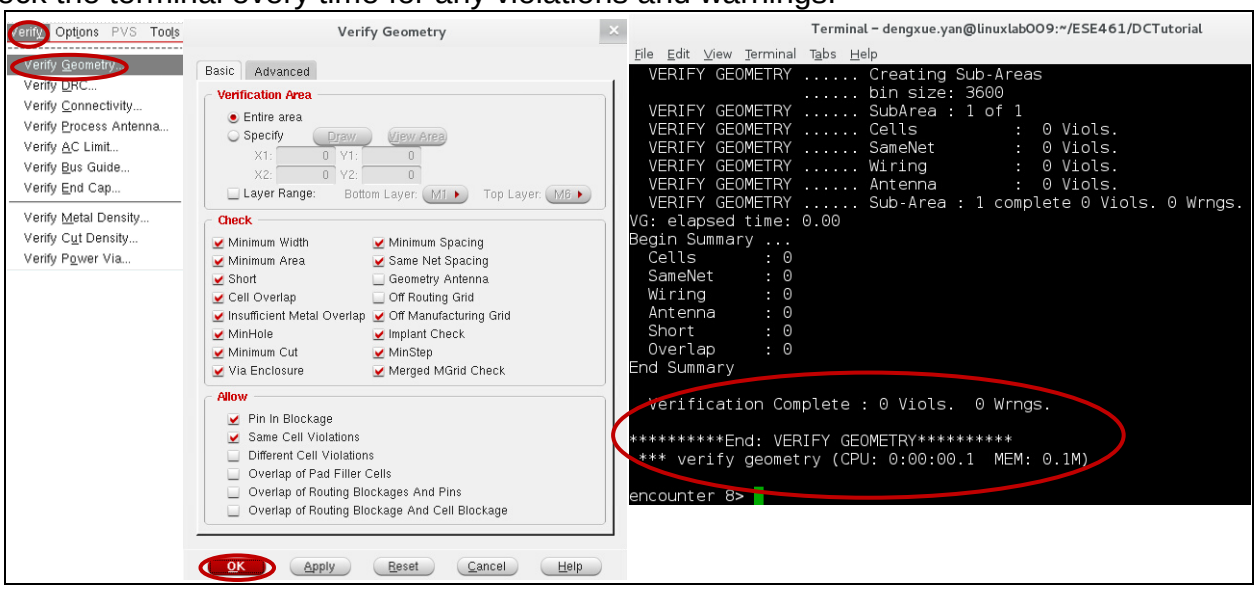

Figure 4.20 Verify Geometry

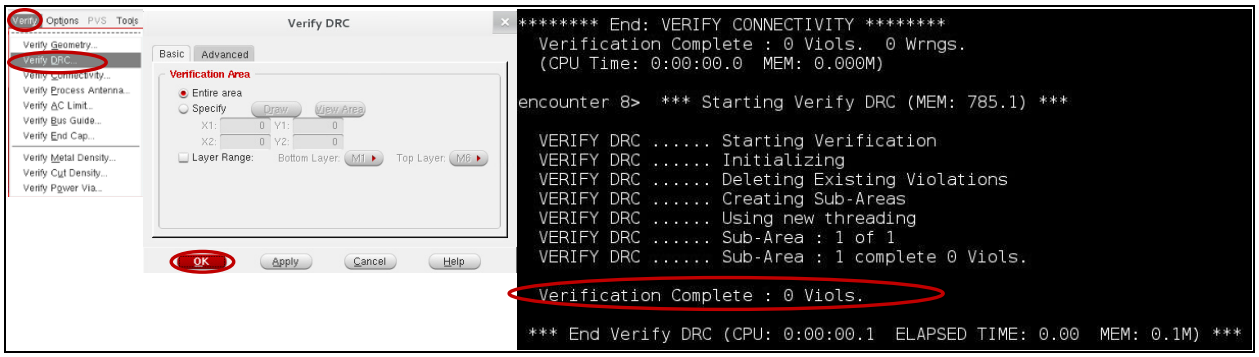

Figure 4.21 Verify DRC

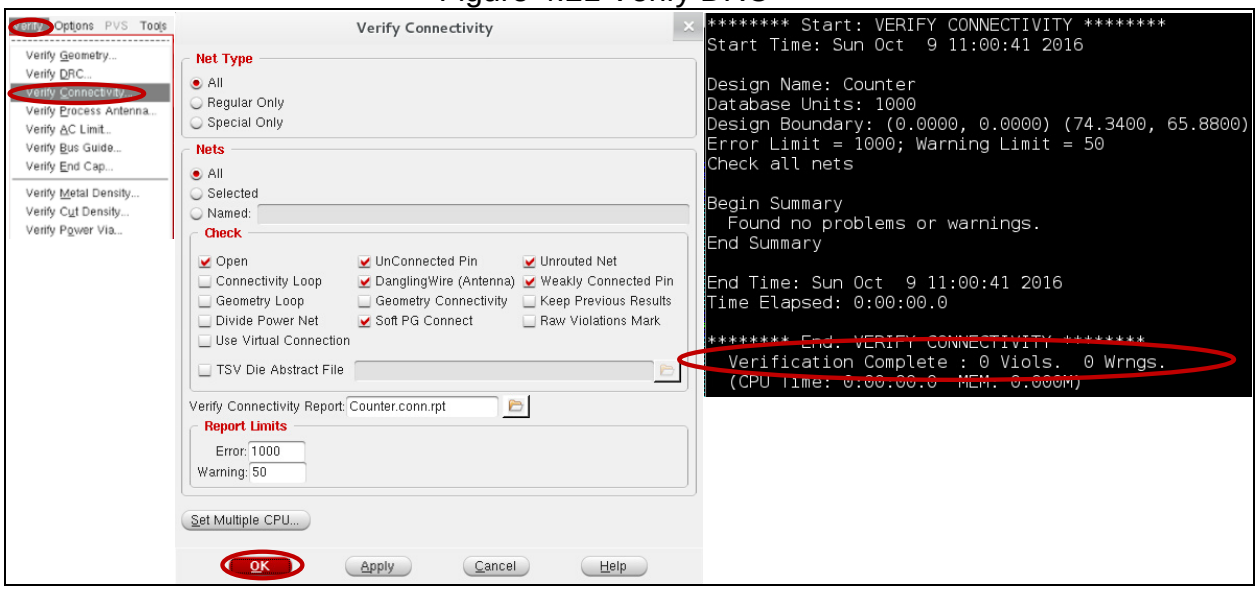

Figure 4.22 Verify Connectivity

4.7 Timing report

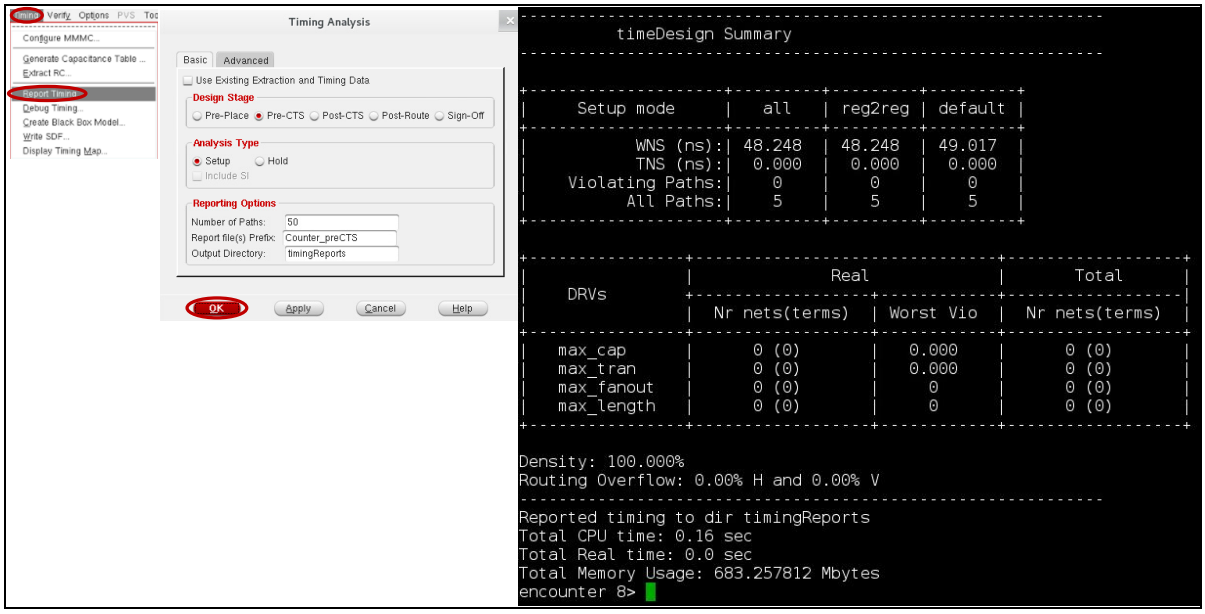

Figure 4.22 Generating timing report

4.8 Additional Information

Save your design for later use. "*File -> Save*". Give an output file name for each floorplan, Place, Netlist and DEF file so that you can save all the optimization options and can load it later.

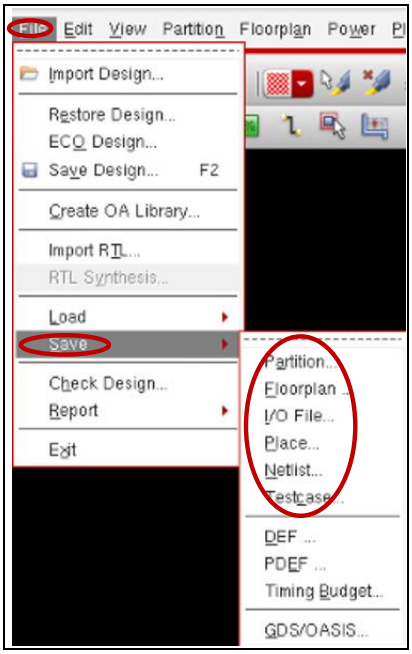

Figure 4.23 Save your project

5. Reference

[1] https://cornell-ece5745.github.io/ece5745-tut5-asic-tools/
[2] https://cornell-ece5745.github.io/ece5745-tut6-asic-flow/
[3] http://classes.engineering.wustl.edu/ese566/Tutorial/ese566-linux-tutorial.pdf
[4] http://classes.engineering.wustl.edu/ese566/Tutorial/ese566-vcs-tutorial.pdf
[5] http://classes.engineering.wustl.edu/ese566/Tutorial/ese566-dc-tutorial.pdf
[6] http://www.csl.cornell.edu/courses/ece5745/handouts.html