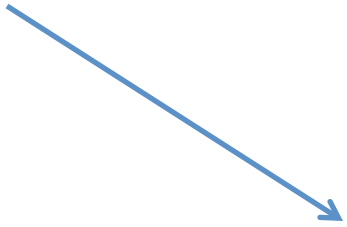
Cadence SOC Encounter Tutorial
for a logic block using
the University Of Utah Standard cell Libraries
In ON Semiconductor 0.5 µ C5N CMOS

1. Create a folder called encounter inside of your project folder
2. Copy the following files in to your encounter folder
   - From your synthesis folder copy
     - Yourfile_rtl.v
     - Yourfile_rtl.sdc
   - Download and copy
     - generic.conf (this file can be renamed to yourfile.conf)
     - cds.lib (**Do Not** modify the name of this file)
3. Open the generic.conf file and enter the name of your files
   - Change example_RTL.v to yourfile_rtl.v
   - Change example to the name of the top module in your example_RTL.v file
   - Change example_RTL.sdc to yourfile_rtl.sdc



```
#########################################################
# Here are the parts you need to update for your design
#########################################################
set rda_Input(ui_netlist)        {example_RTL.v}
set rda_Input(ui_topcell)        {example}
set rda_Input(ui_timingcon_file) {example_RTL.sdc}
#
```

Part II

Open a command line and navigate to your encounter directory. You can verify that you are in the correct directory by typing pwd. Do not move on until you are in the directory with your files

```
[142]cd002_eepc6> pwd
/users/class/ee5369/cd002/Project/encounter
[143]cd002_eepc6>
```

Once your are in the correct directory enter the following commands

1. source /export/cadence/linux/virtuoso.cshrc.UofU
2. encounter -win

```
[148]cd002_eepc6> source /export/cadence/linux/virtuoso.cshrc.UofU
[149]cd002_eepc6> encounter -win
```

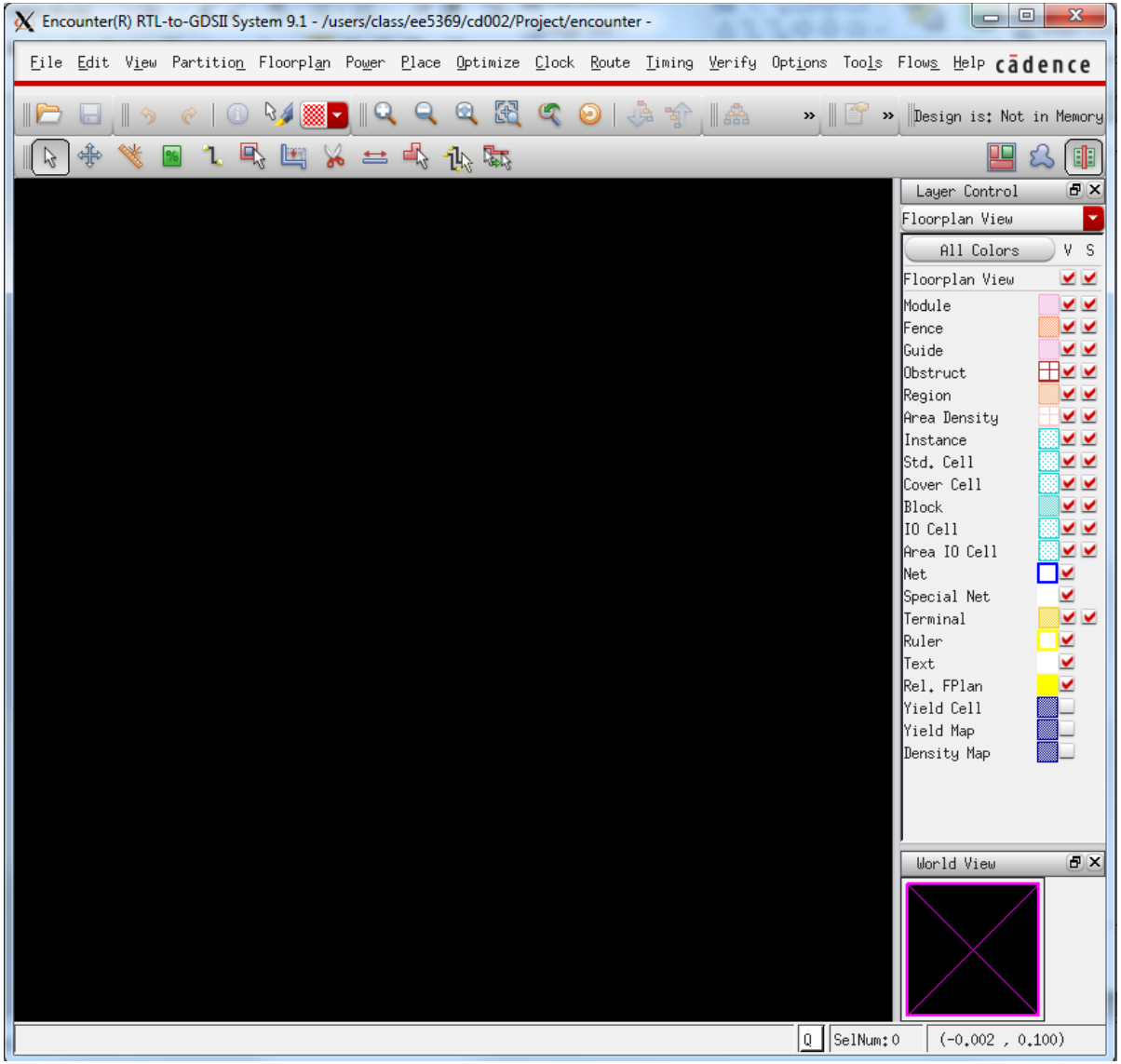
Note: Occasionally encounter will not open completely and you will need to force it to shut down and restart it. You can use one of the two following methods.

1. pkill encounter
2. The second method requires two steps
    1. ps (this will give you the programs pid)
    2. kill pid#

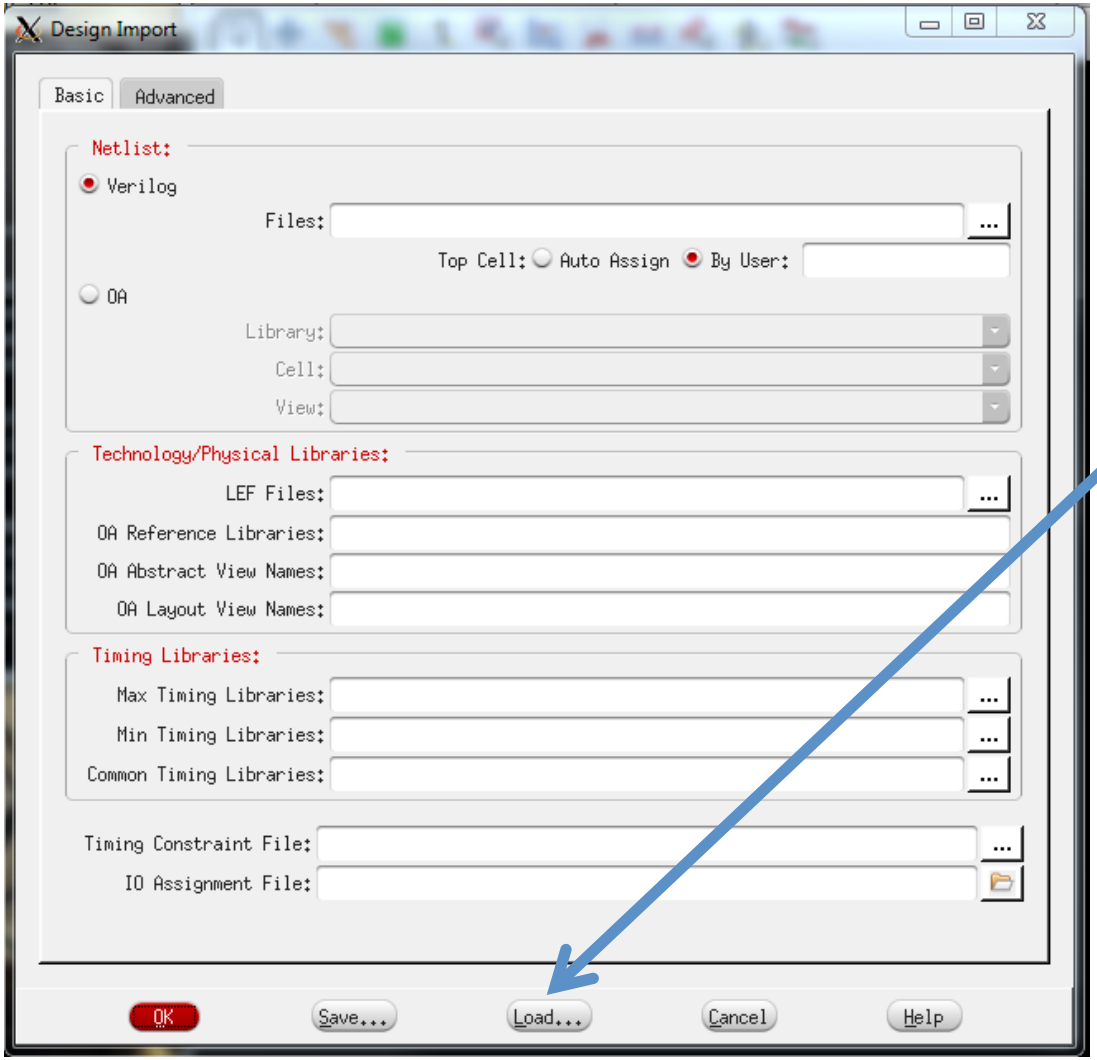
```
encounter 3> pid
20815
encounter 4> kill 20815
```

The following screen will appear. Click on

File → Import Design



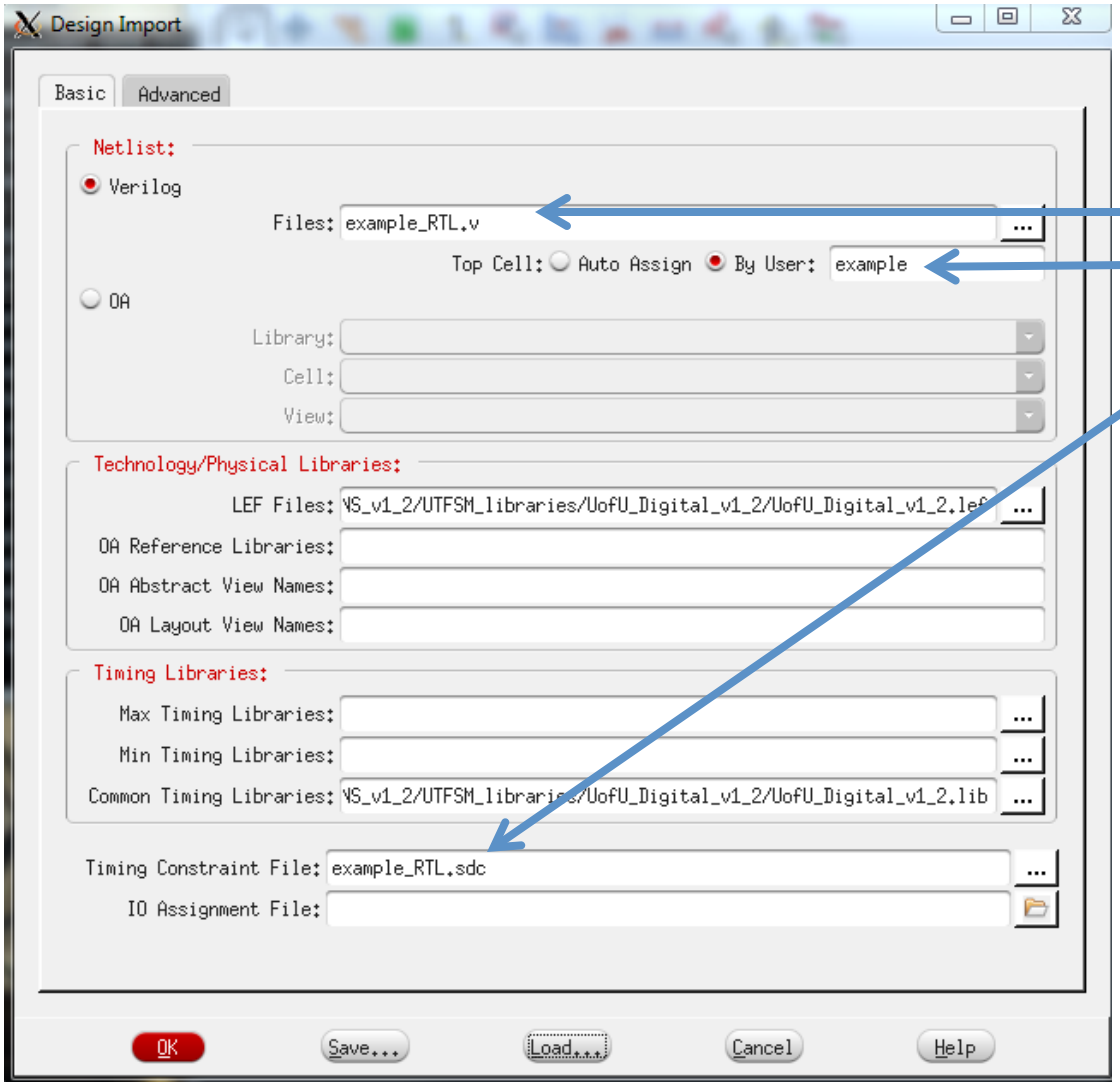
This screen will be used to load your .conf file. If you didn't change your name you will use the generic.conf file.



Click on load and select your .conf file and click open

Note: If you do not see your .conf file immediately after hitting load your did not open encounter in the correct directory. You can either navigate to the correct folder or exit the program go to the correct folder and restart encounter.

Notice that some of the files are now populated.



Verify that these 3 fields are correct

1. Your_RTL.v
2. The name of your main module
3. Your_RTL.sdc

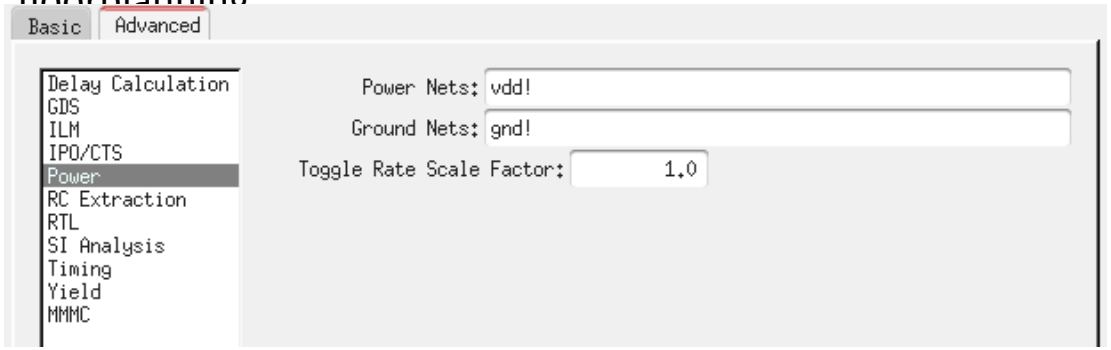
If you click on the **Advanced** tab you can see that the following changes have also been made

CTS (Clock Tree Synthesis) Cell List specification has changed to INVX1

This lets SOC Encounter know which inverter cells it can use during Optimization.

Power Nets specification has changed to vdd! and Ground Nets specification has changed to gnd!

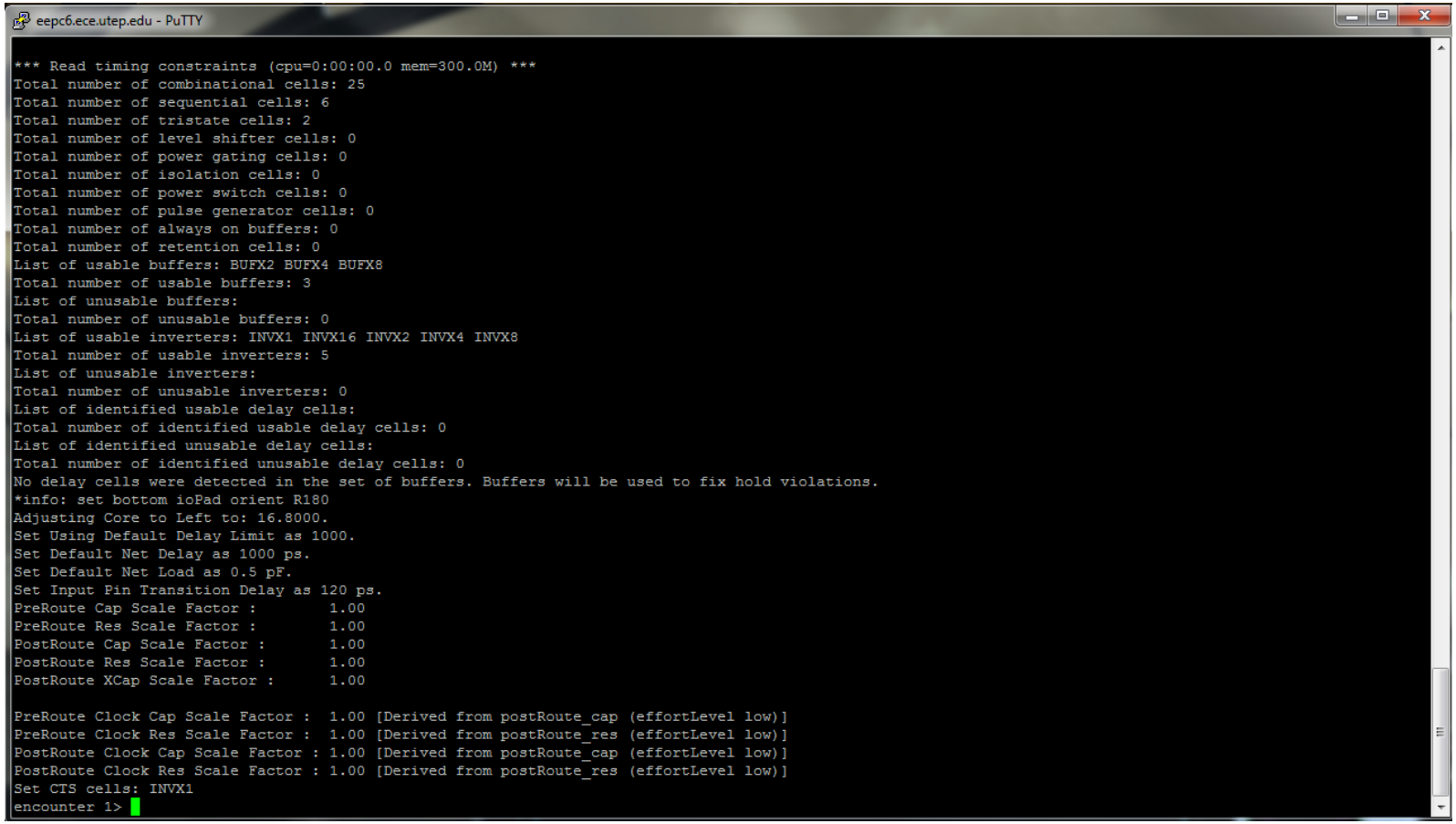
These are the names that Encounter will give to your ground and power nets during floorplanning



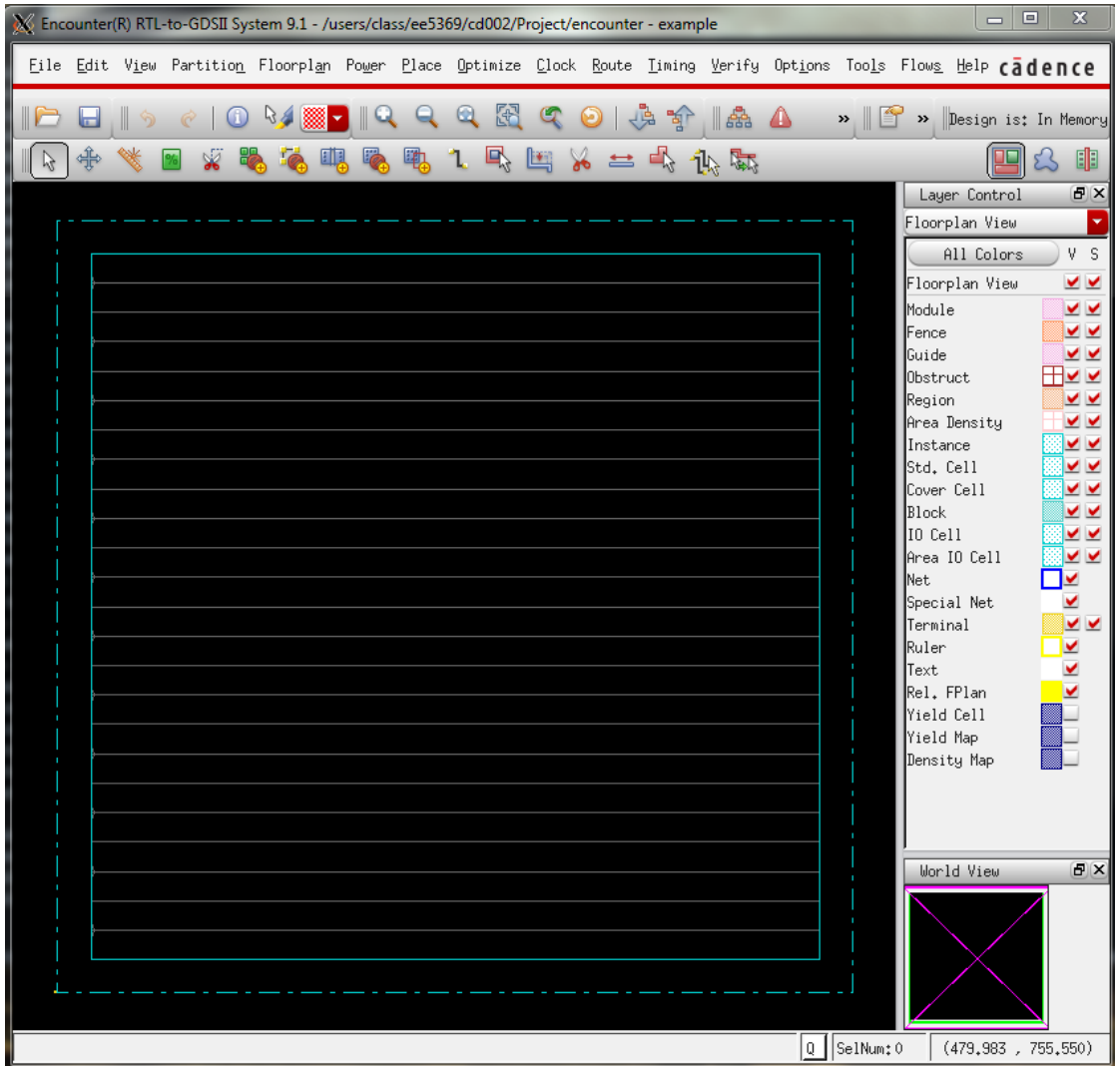
Click **ok** to continue

Once you click ok take a look at the shell window that you used to invoke SOC Encounter. It will have comments (and potential warning and errors). You should read these comments carefully and make sure that there were no error and that the warnings are ok to ignore.

You can also find these comments in the encounter.log file that was created in your working directory. If you followed the conventions set in these instructions the ecounter.log file will be in your encounter folder.

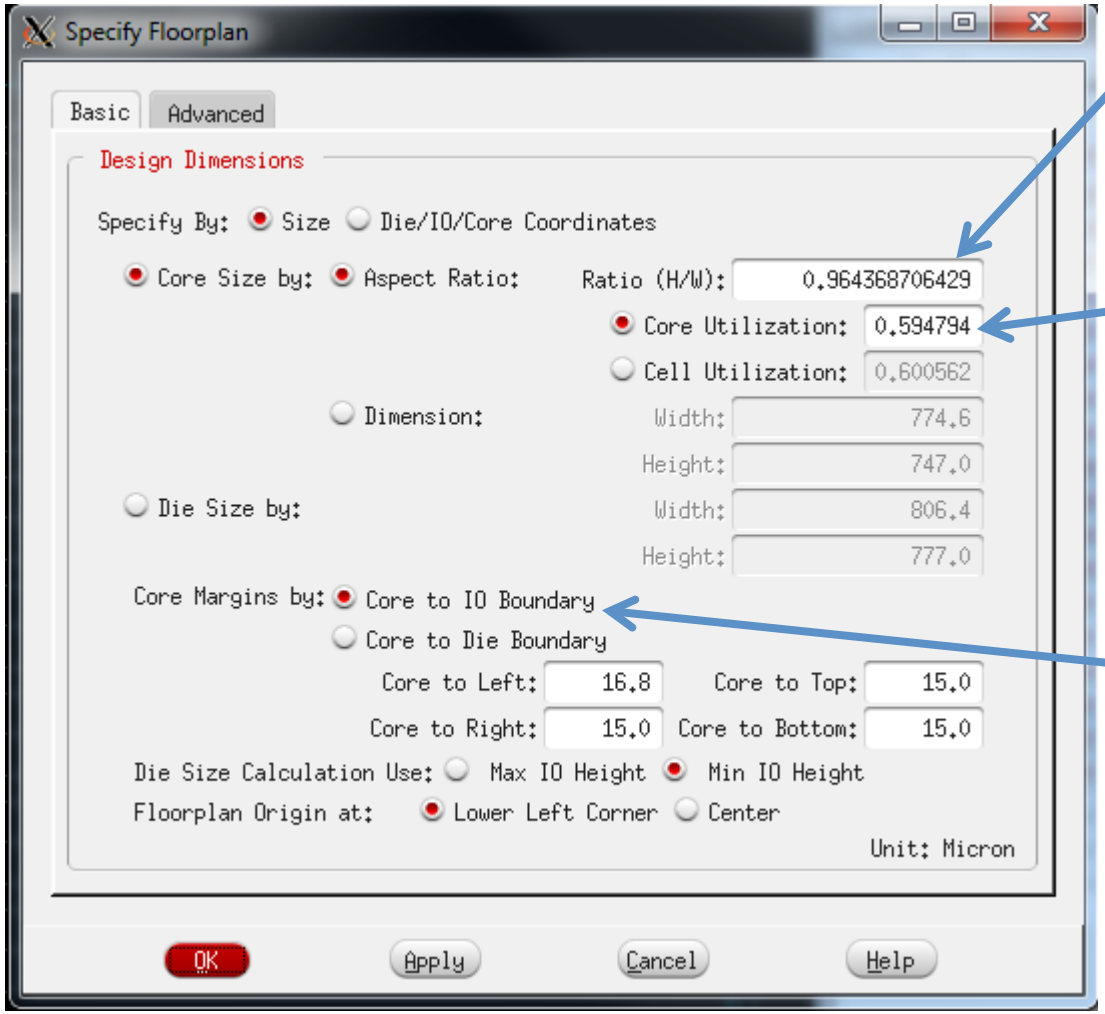


After clicking ok and making sure that everything imported ok take a look at the new window that appeared. The figure should be similar to the figure below. The set of rows are where the standard cells will be placed.



Click on **Floorplan → Specify Floorplan**

A screen will appear. The values that need to be set should have been set by your generic.conf file but it is a good idea to verify all the values.

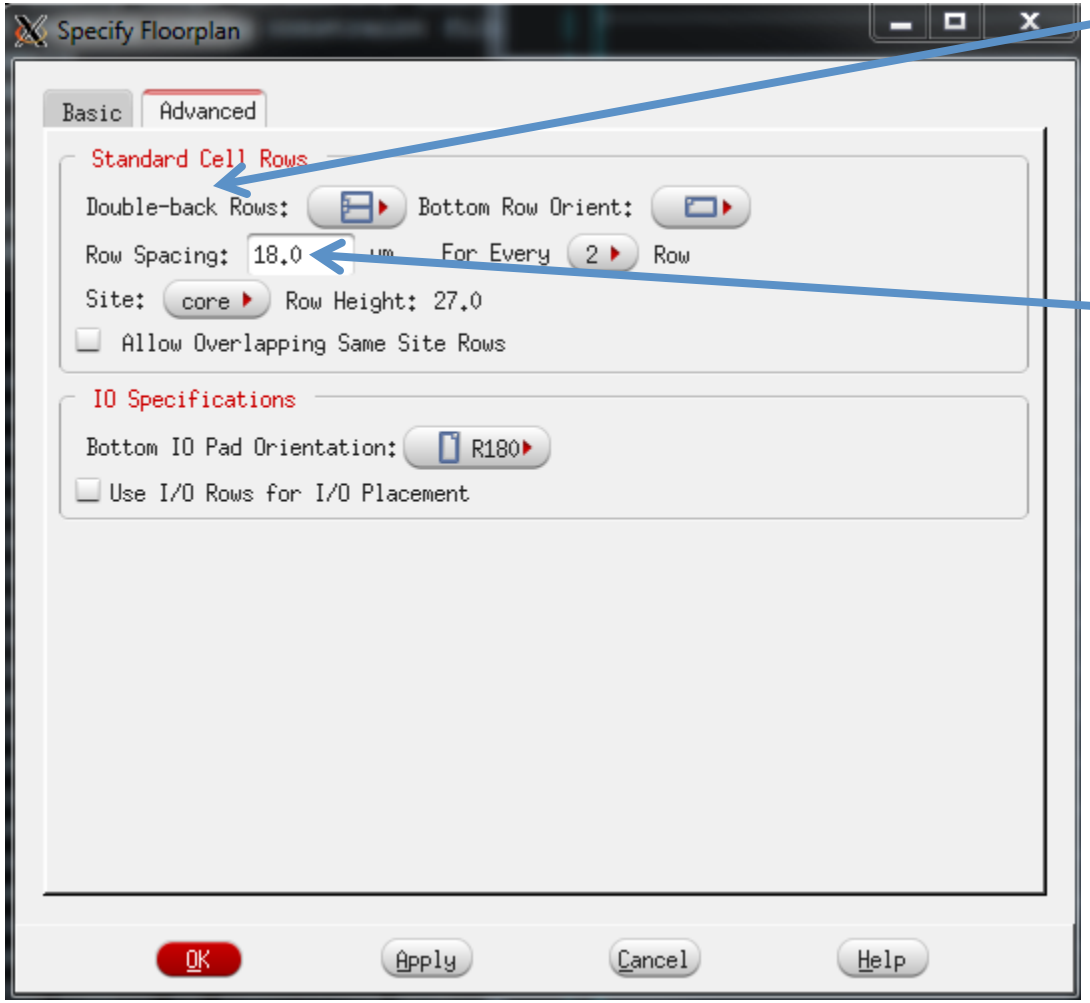


Ratio:
A ratio of close to **1** will set the shape of the cell to square. The conf file sets the ratio to 1 but SOC Encounter will adjust this number a bit based on the anticipated cell sizes.

Core Utilization :
Lets Encounter know how densely packed the core should be with standard cells .The conf file changes this value from the default .7 to **.6**
For a more complex design you may have to reduce the utilization percentage further.

Core Margins :
The margins are to leave room for the power and ground rings that will be generated around the cell. The conf file sets all the margins to **15** but Encounter changes them a little according to its measurements

Next click on the the **Advanced** tab. Note that Row Spacing will need to be changed



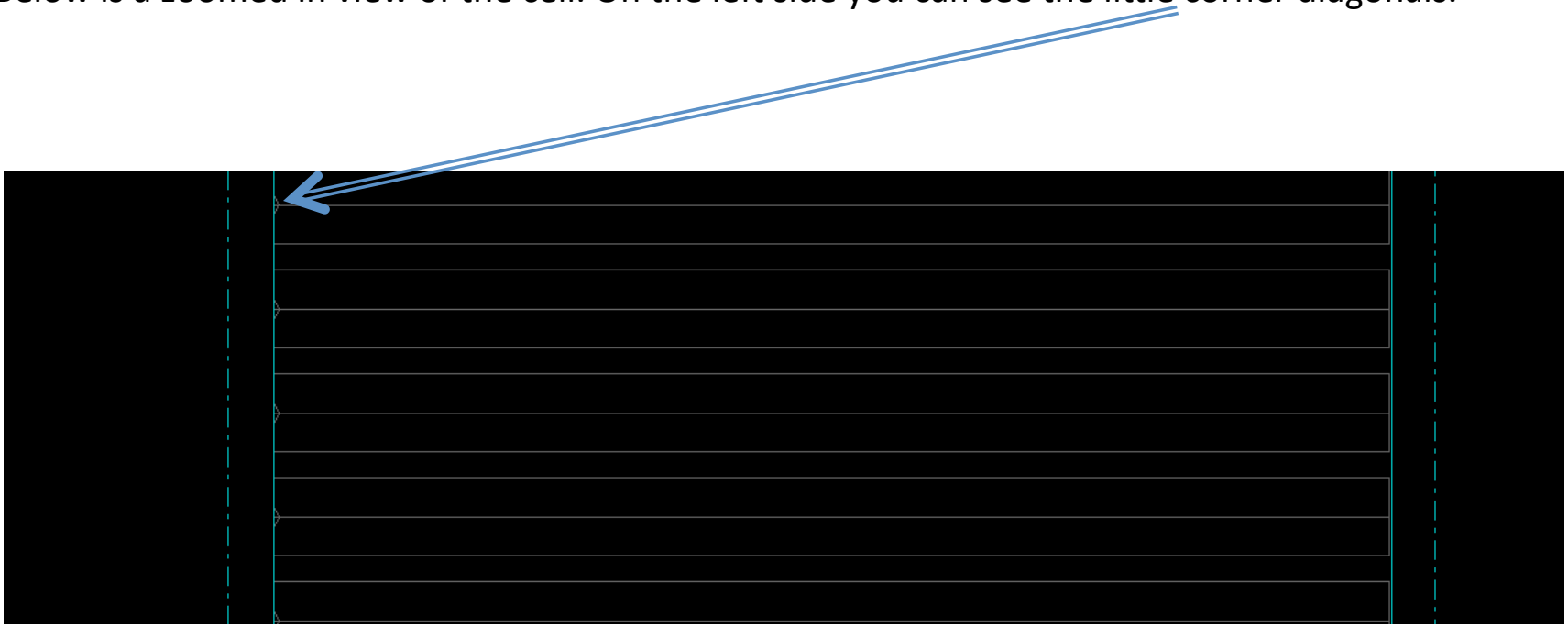
Rows:
Verify that rows is set to Double-back

Row Spacing:
Change Row Spacing to **18.0.**

Because we only have 3 layers it is necessary to leave room between cells so that SOC Encounter has more room for routing

After adjusting the floorplan the window will look similar to the window below. The rows in which cells will be placed are in the center, with little corner diagonals showing how the cells in those rows will be flipped. The dotted line is the outer dimension of the final cell. The power and ground rings will go between the cells and the outer boundary.

Below is a zoomed in view of the cell. On the left side you can see the little corner diagonals.



Part III- Power Planning

Before continuing to the power planning portion you should save the current design. From this point forward there are steps that cannot be undone. If you need to try something different you can simply reload your design.

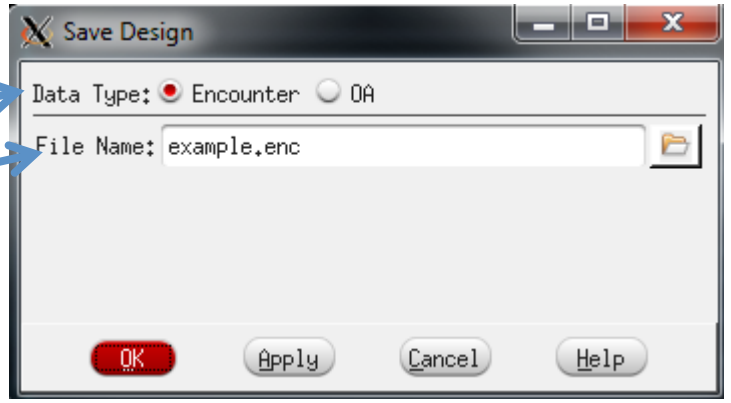
Saving the Design:

Click on **File → Save Design**

For Data Type select Encounter

Select a file name. Your_file.enc

Click ok

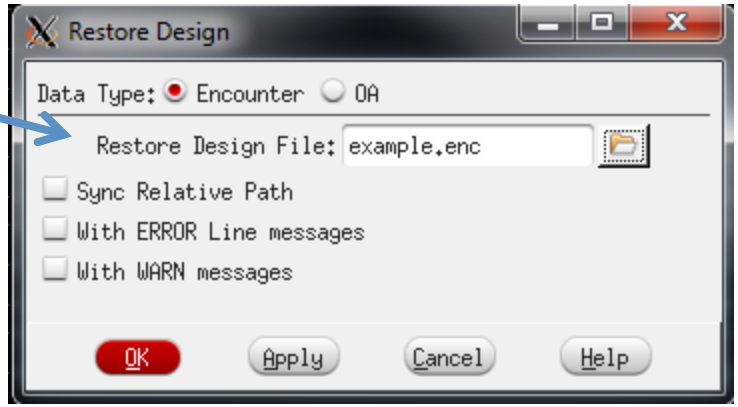


Restoring the Design:

Click on **File → Restore Design**

Select the file you want to restore

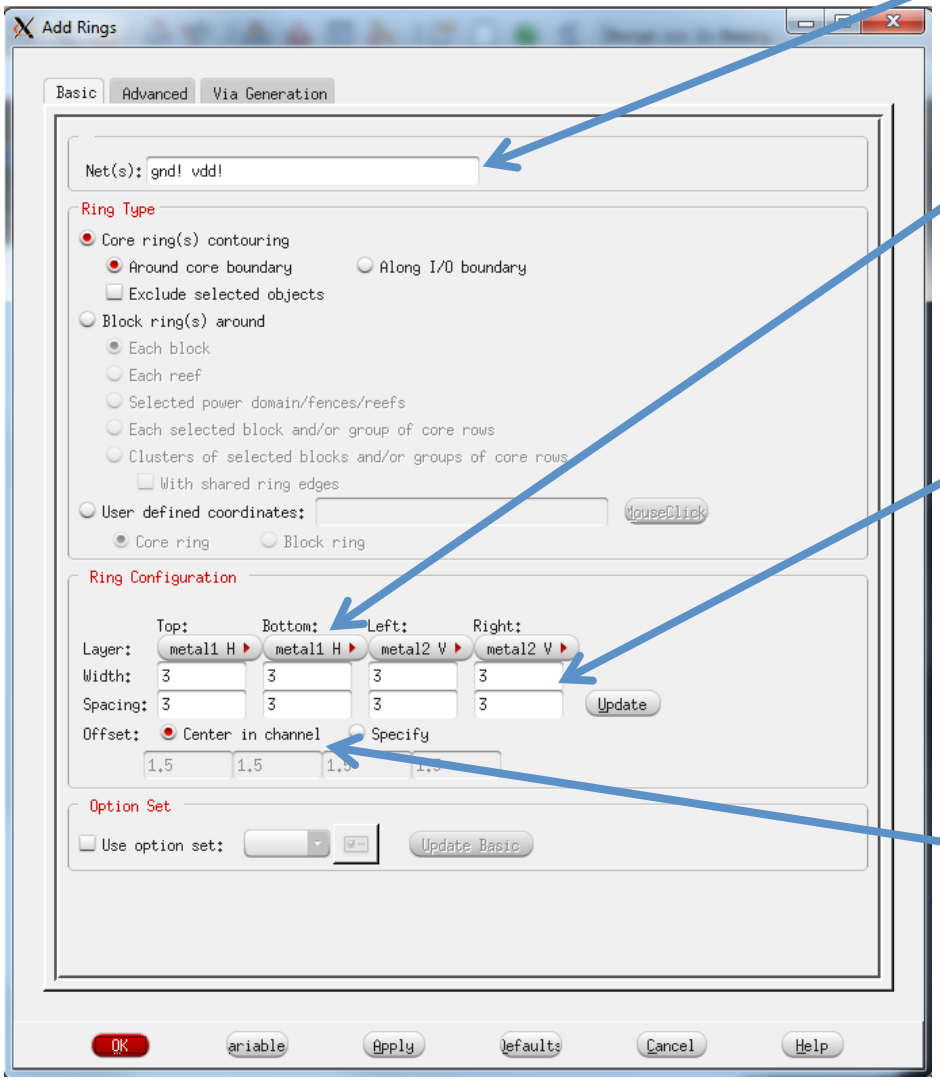
Click ok



Note: if your file is not there you may have opened SOC Encounter in the wrong directory. Go back and make sure you are in the correct working directory before invoking the encounter –win command.

Power Planning

Click on **Power→ Power Planning → Add Rings**



Should already say gnd! vdd! If it does not go back and make sure your project is loaded

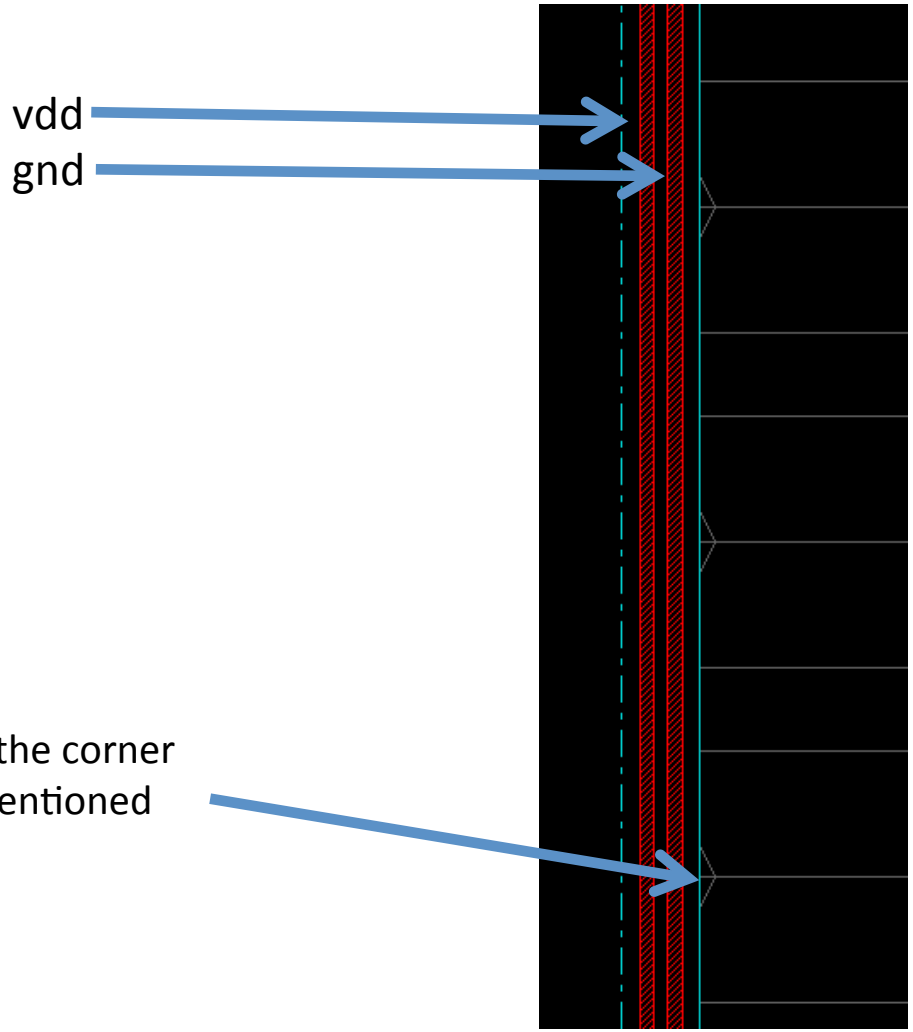
You can select the metal layers you want to use for the ring. I left the defaults. Metal 1 for the left and right vertical and metal 2 for the top and bottom horizontal

This is the width of the metal that will be used for the vdd and gnd lines. Spacing is the space between the lines. For larger designs a much thicker metal is used. Thicker metal means that we need a larger spacing between vdd and gnd.

Change Offset to center in Channel. This will center the lines in the available channel we made earlier.

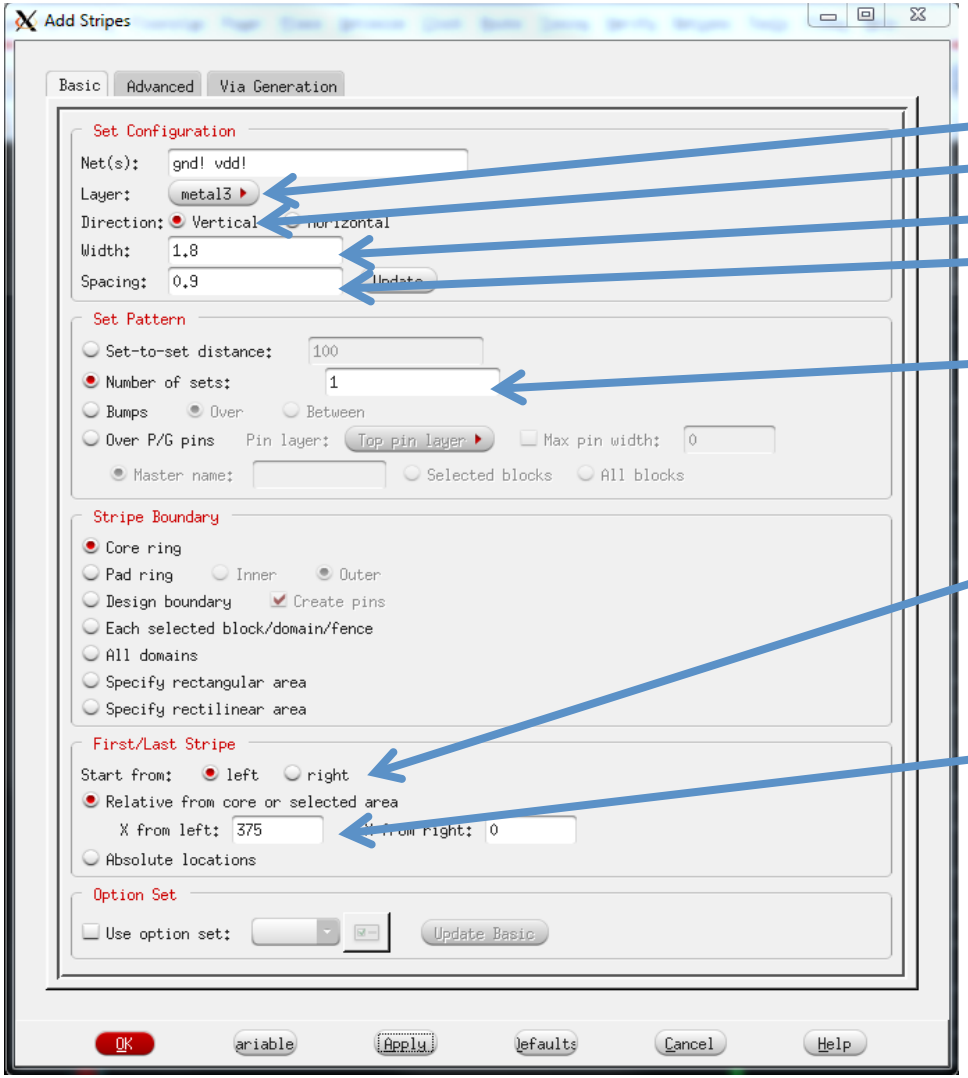
Click ok. The window should look similar to the window below.

If you zoom in to the new lines you can see that they are actually two lines. The line on the outside is your vdd the line on the inside is your gnd. Red is metal 2 and blue is metal 1



Click on **Power → Power Planning → Add Stripes**

This command creates additional power network robustness but at the expense of routability later.



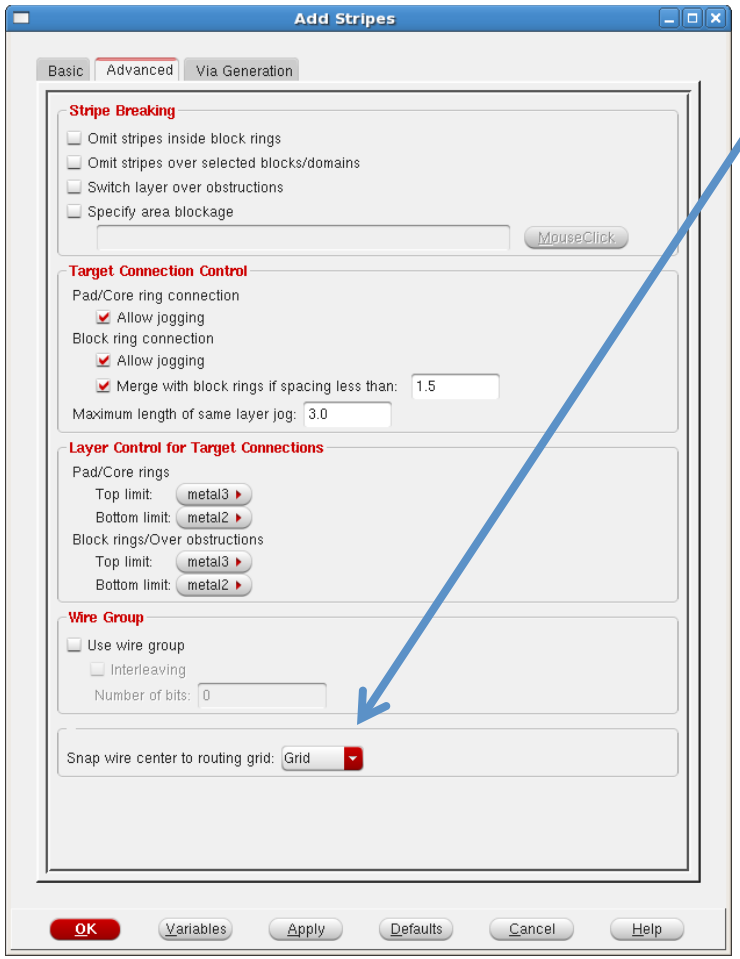
Set the following properties
-layer to metal3
-Direction to vertical
-width to 1.8
-spacing to 0.9

-select Number of sets radio button and change its value to 1

-select start from left
-select relative from core or selected area

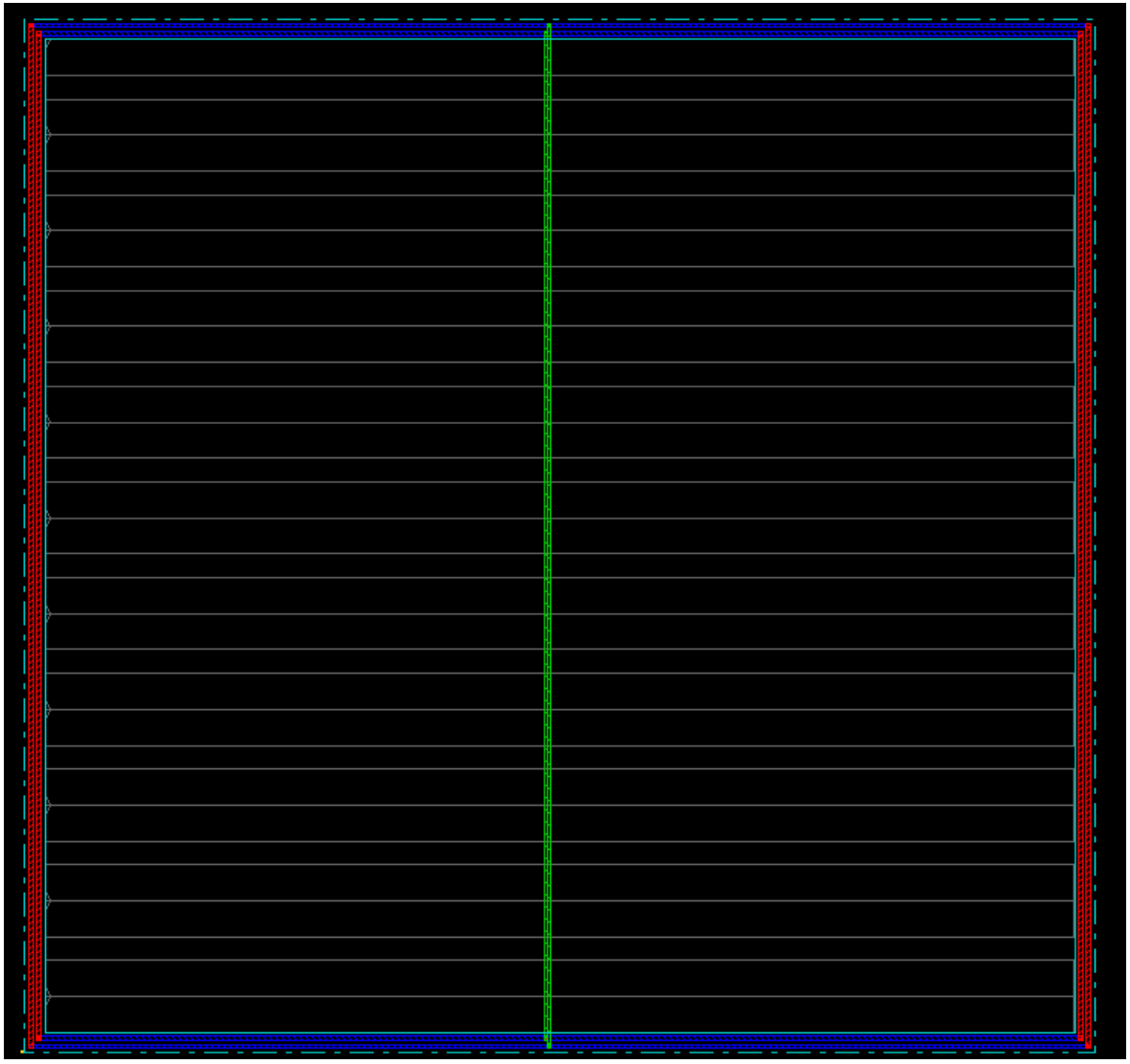
-This value will change depending on the size of your core. Enter 375 and click on apply. The line power line will appear in your core. If you would like you can delete it by clicking on it and hitting delete key. Increasing the number will move it further right. Decreasing the number will move it further right.

Next click on the advanced tab. Change **Snap wire center to routing grid** from None to **Grid** click apply



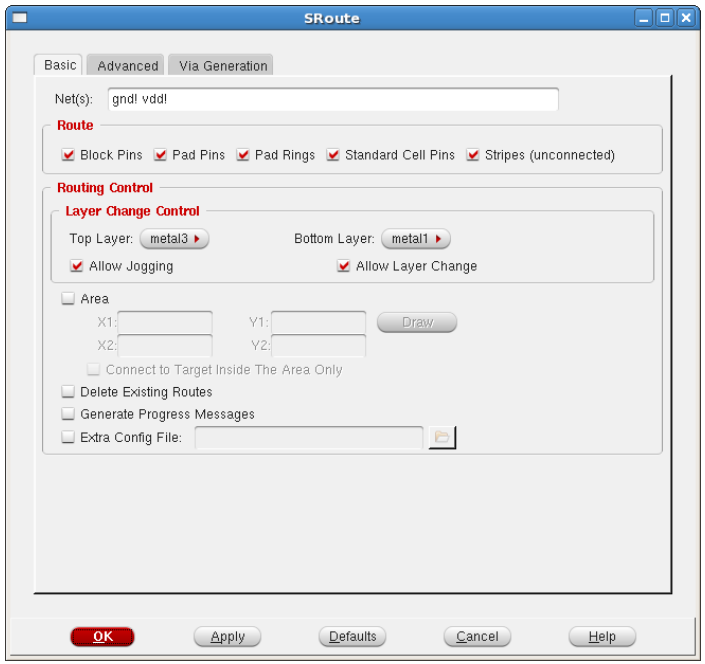
If you don't center the power stripes on the grid, a cell could be placed right next to a power grid and cause a metal spacing DRC violation when the metal of the stripe is only 0.6µ from the metal in the cell. Centering the stripe on a grid keeps this from happening by placing the metal of the stripe in a position so that the next legal cell position isn't directly abutting with the power stripe

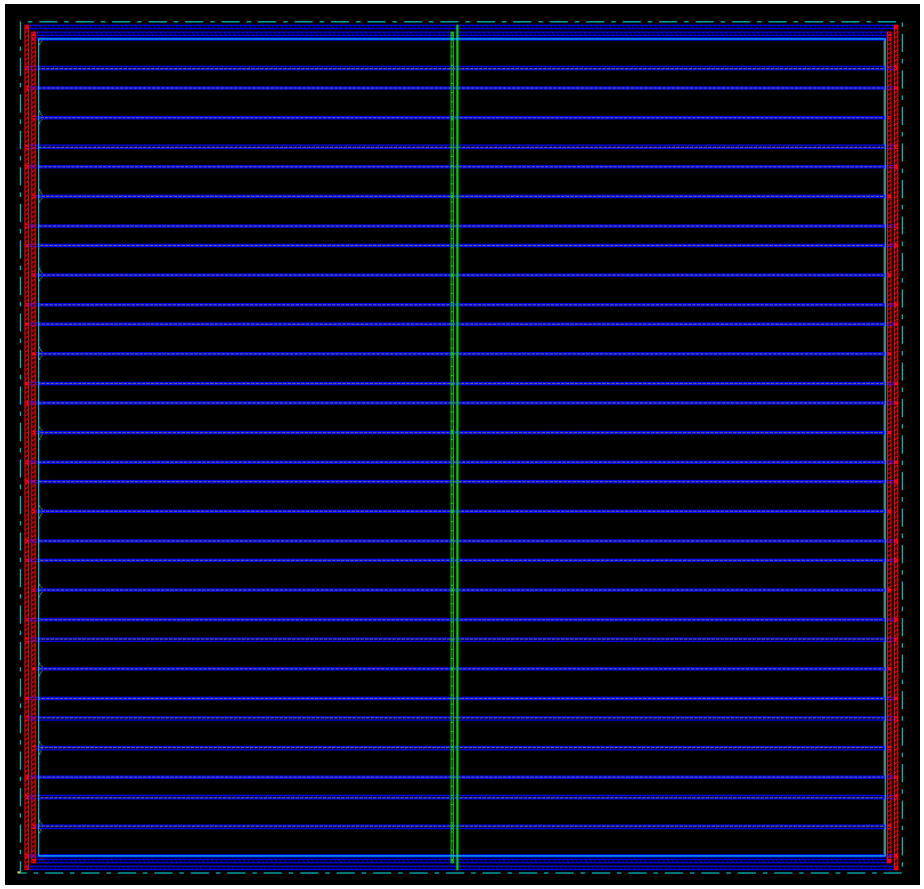
Clicking **Apply** will apply the stripes to your design. If you don't like the way the stripe looks you can select the stripe and hit the delete key to get rid of it and you can try again with new parameters. Once the line is aligned properly hit ok and your screen will look like the one below. Green represents metal3.



The green metal 3 is the power strap that will help tie each of the circuit rows to ground and vdd with less resistance. The additional power network robustness come at the expense of routability

Once you have the stripes place, you can connect power to the rows where the cells will be placed. Select **Route → Special Route** to route the power lines. Make sure your power supplies (vdd! and gnd!) are listed in the nets. Leave all the other defaults and click **OK** and you will see the rows have their power connections made to the ring/stripe grid as shows in the figure below

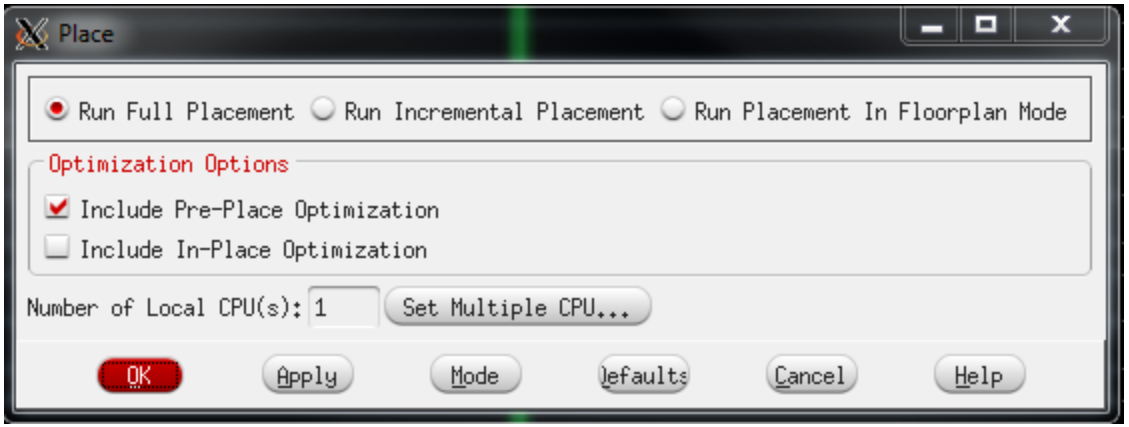




Part III – Placing Standard Cells

This is a good place to create another backup. Next we will be placing the cells. If there are errors not involving the power network you can restore to this point and resume from here.

Click on **Place** → **Place Standard Cell**

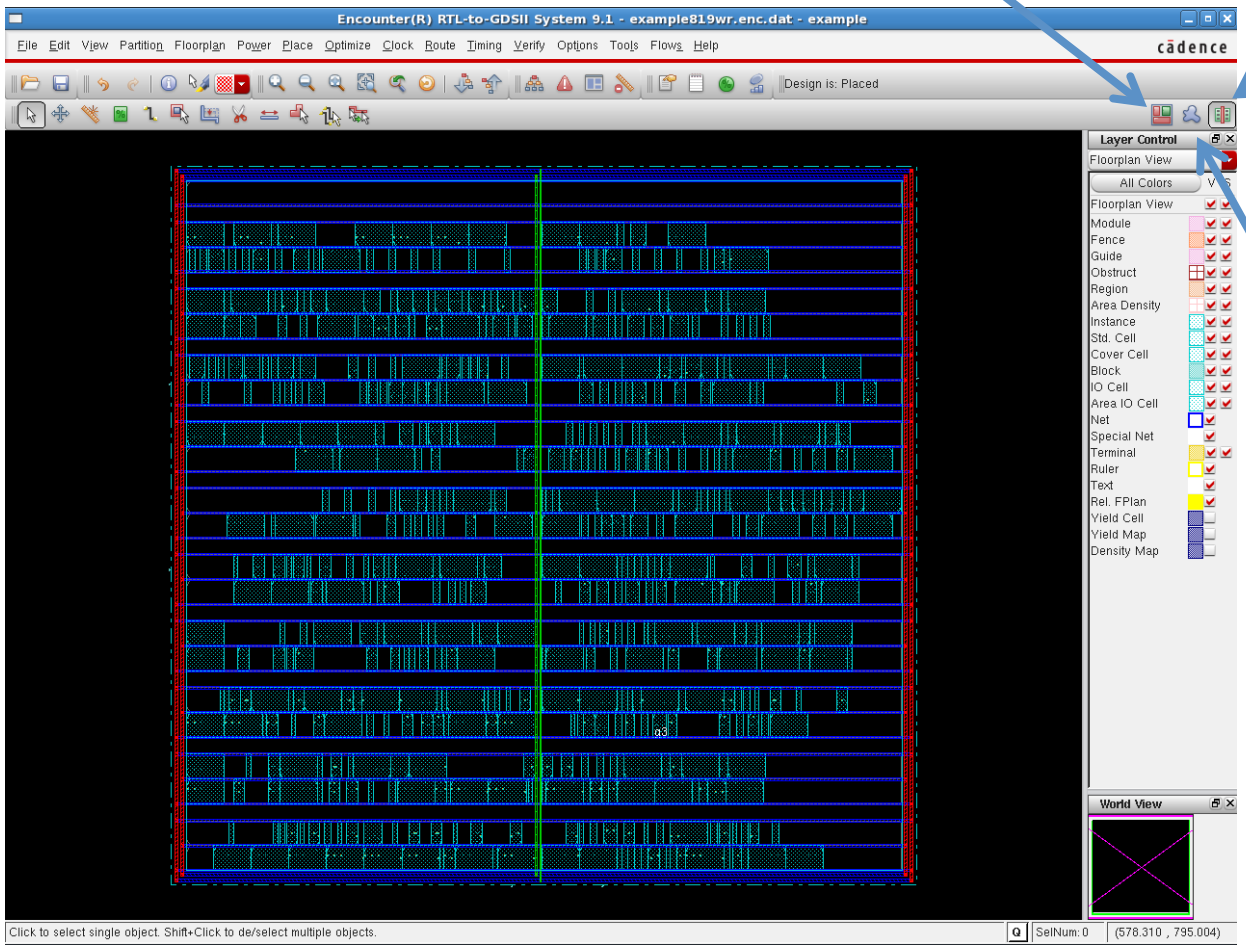


Leave the defaults in the dialog box, as seen in the figure above. You want to Run Full Placement and Include Pre-Place Optimization.

After pressing **Ok**,your cells will be placed in the rows on the floorplan. This might take a while for a large design. When it's finished, the screen won't look any different until you change the view. The **physical view** is shown on the next slide.

-Floorplan View

-Click on this Icon to change to Physical view

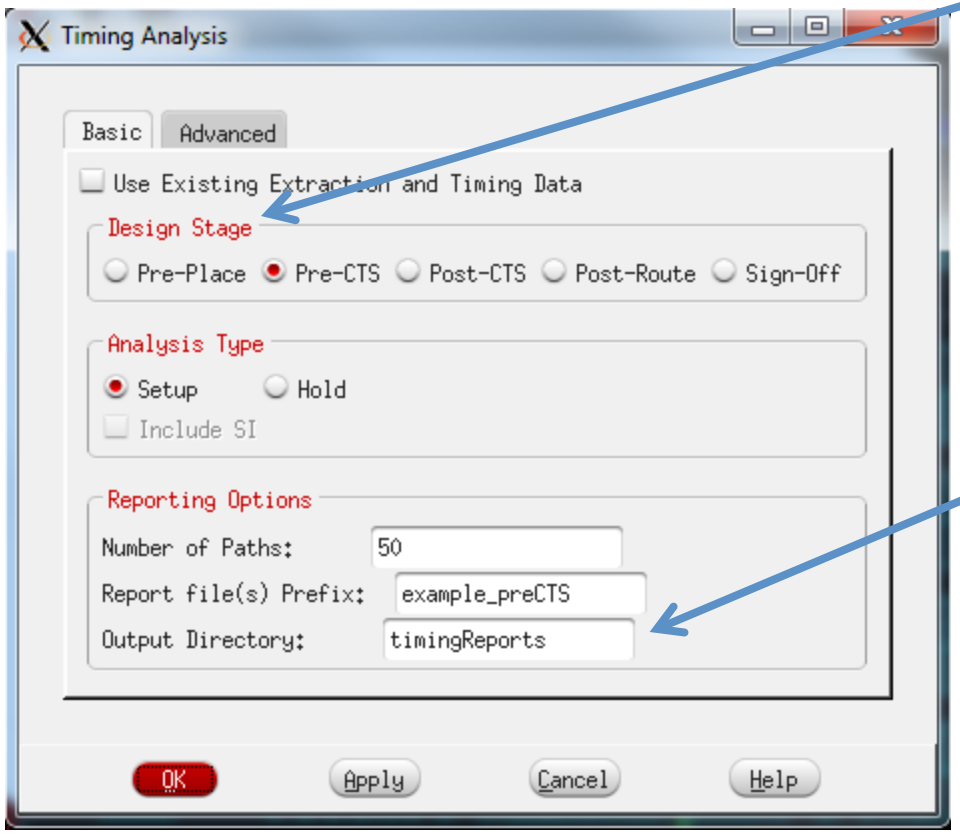


-Amoeba View

First Optimization Phase

At this stage of the process there are no wires, so the timing step will do a trial route to estimate the wiring. This is a low-effort, not necessarily correct routing of the circuit used just for estimation.

Select **Timing → Report timing**



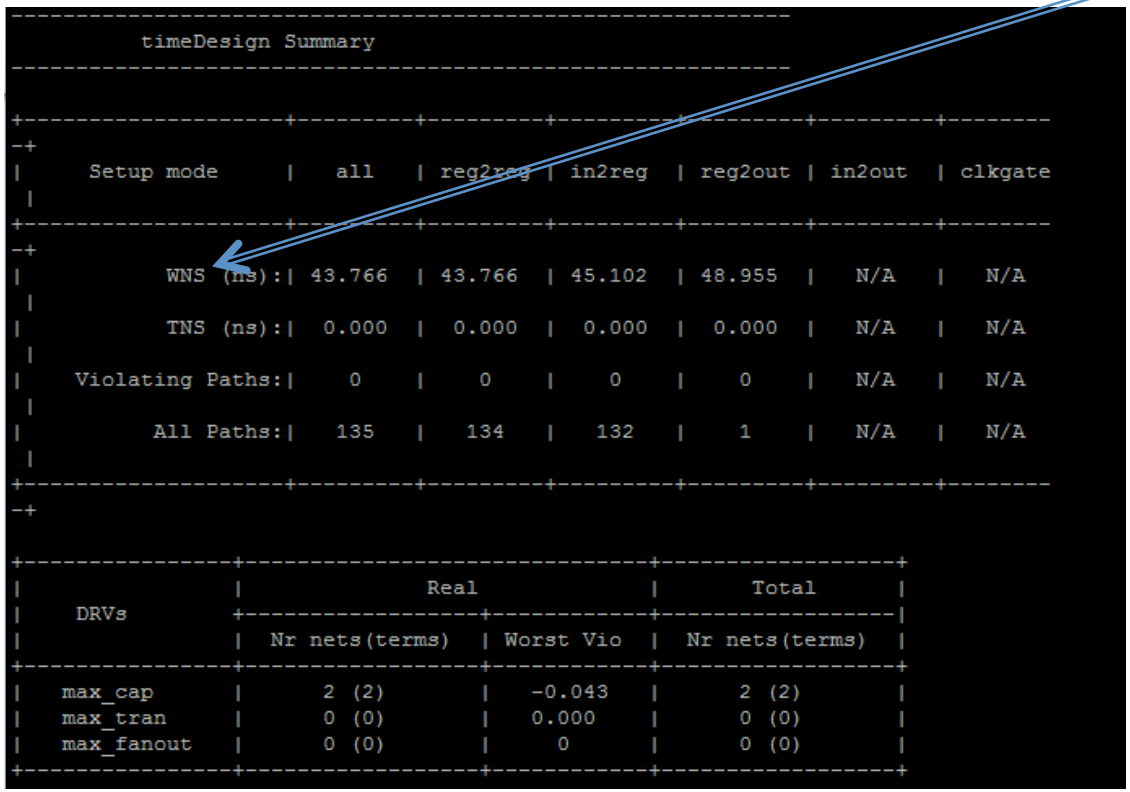
Under Design State select Pre-CTS to indicate that this analysis is before any clock tree has been generated. You're also doing analysis only on the setup time of the circuit.

Click **OK**

Creates a folder in your encounter folder. The timing report that is in your shell can also be found in this folder. It should be named YourFile_preCTS.summary

The summary is shown on the next slide

The “timeDesign Summary” report can also be found in your newly created timingReports folder.



```
------------------------------------------------------
          timeDesign Summary
------------------------------------------------------

+--------------------+---------+---------+---------+---------+---------+--------
-+
|     Setup mode     |   all   | reg2reg | in2reg  | reg2out | in2out  | clkgate
 |
+--------------------+---------+---------+---------+---------+---------+--------
-+
|           WNS (ns):| 43.766  | 43.766  | 45.102  | 48.955  |   N/A   |   N/A
 |
|           TNS (ns):|  0.000  |  0.000  |  0.000  |  0.000  |   N/A   |   N/A
 |
|    Violating Paths:|    0    |    0    |    0    |    0    |   N/A   |   N/A
 |
|          All Paths:|   135   |   134   |   132   |    1    |   N/A   |   N/A
 |
+--------------------+---------+---------+---------+---------+---------+--------
-+

+----------------+-------------------------------+-------------------+
|                |              Real             |       Total       |
|    DRVs        +------------------+------------+-------------------|
|                |  Nr nets(terms)  | Worst Vio  | Nr nets(terms)    |
+----------------+------------------+------------+-------------------+
|   max_cap      |      2 (2)       |   -0.043   |      2 (2)        |
|   max_tran     |      0 (0)       |   0.000    |      0 (0)        |
|   max_fanout   |      0 (0)       |     0      |      0 (0)        |
+----------------+------------------+------------+-------------------+
```

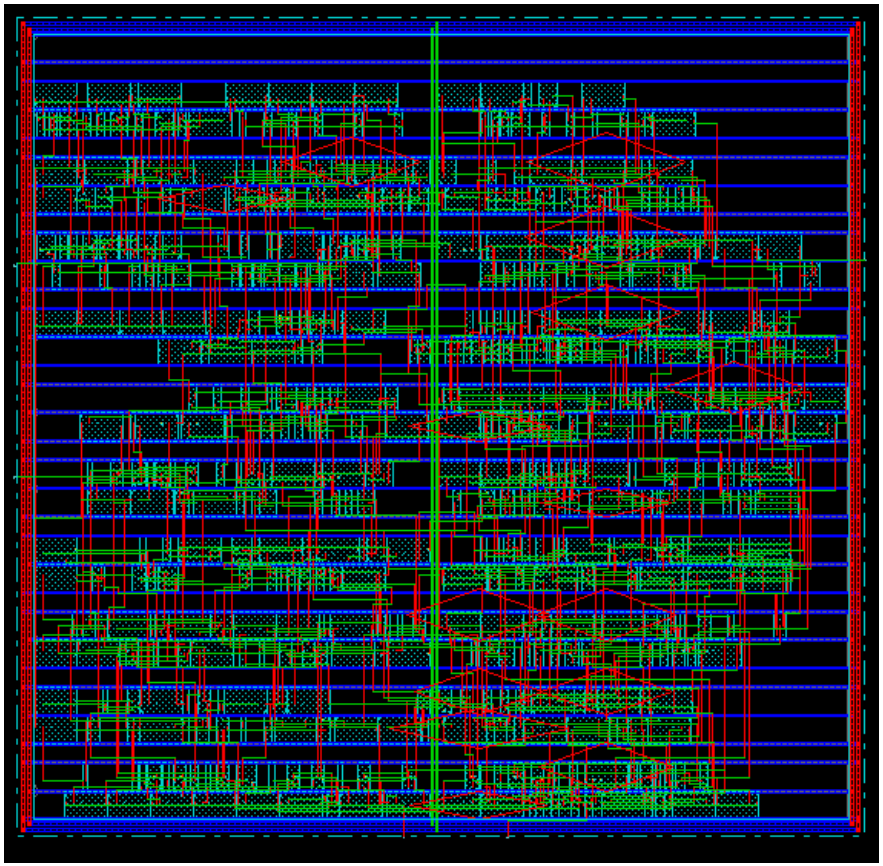
The timing report shows WNS(worst negative slack) and the TNS (total negative slack. The TNS is the sum of all the negative slacks in all violation paths.

Positive slack is ok. Negative slack means you have a problem(clock skew). Recall that a negative slack means that the signal will not reach its destination in time resulting in setup time violations. As a rule of thumb you should keep clock skew less than 200 ps.

This example does not have clock skew. There is only positive slack

Note: There are still several steps that we will take to optimize the timing. If your slack is a bit high it may be fixed after optimization. If your slack is unreasonably high you may want to adjust your timing constraints and start over with your new .sdc file.

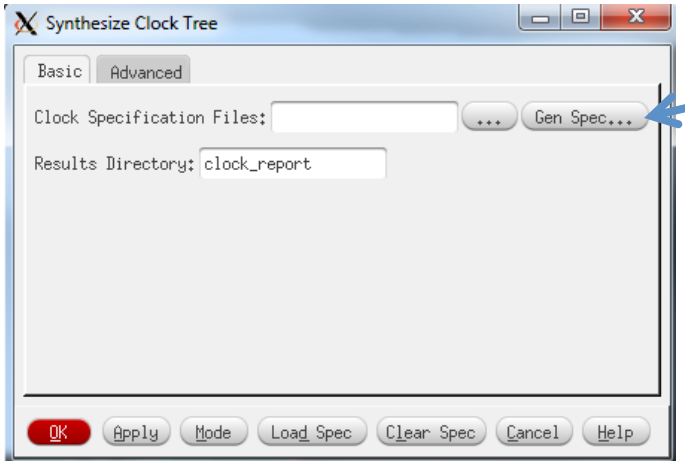
Looking at your SOC Encounter GUI you can see that it looks like the circuit has been routed. This is just a trial route, and these wires will be replaces with real routing wires later.



Note: This is a good place to save your design. Make sure to select the correct design stage to reflect where you are in the design process. For example at this stage I named the design pre-cts.enc

Clock Tree Synthesis

Select **Clock → Synthesize Clock Tree**



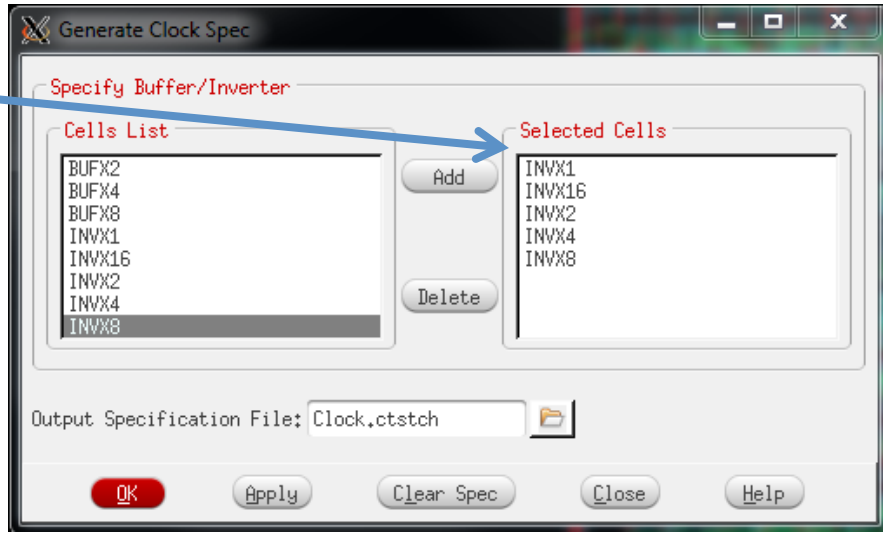
Because we do not have a Clock Specification File yet; you can use the **Gen Spec…** button to create one.

The following window appears after clicking Gen Spec

Select the cells indicated in the figure

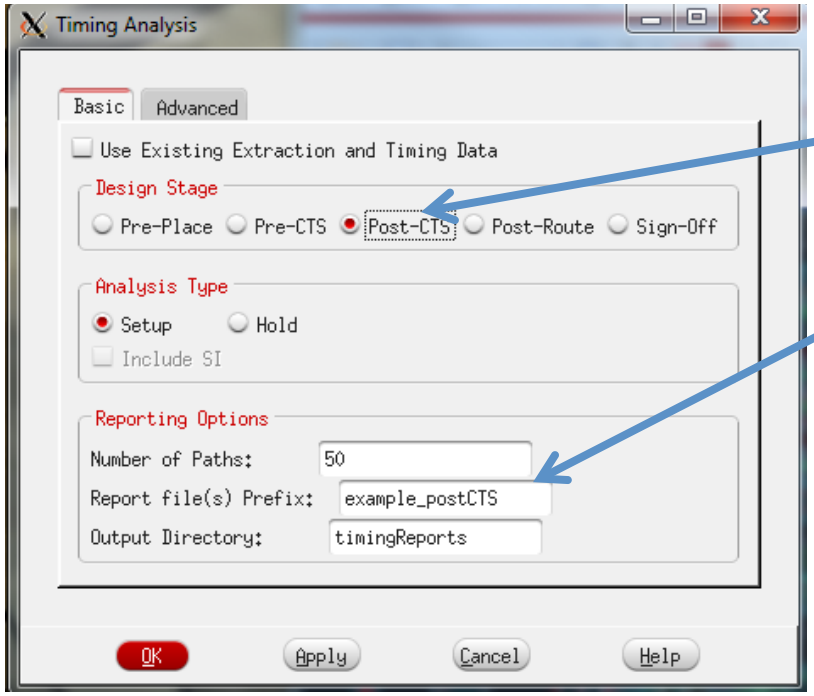
Click **Ok** in the Generate Clock spec window
Click **Ok** in the Synthesize Clock Tree window.

Once you click Ok in the Synthesize Clock Tree window a folder in created in your default folder as well as a file named Clock.ctstch



After you have generated the clock tree, you can do another phase of timing optimization.

Select **Timing → Report timing** but this time select Post-CTS



Post-CTS

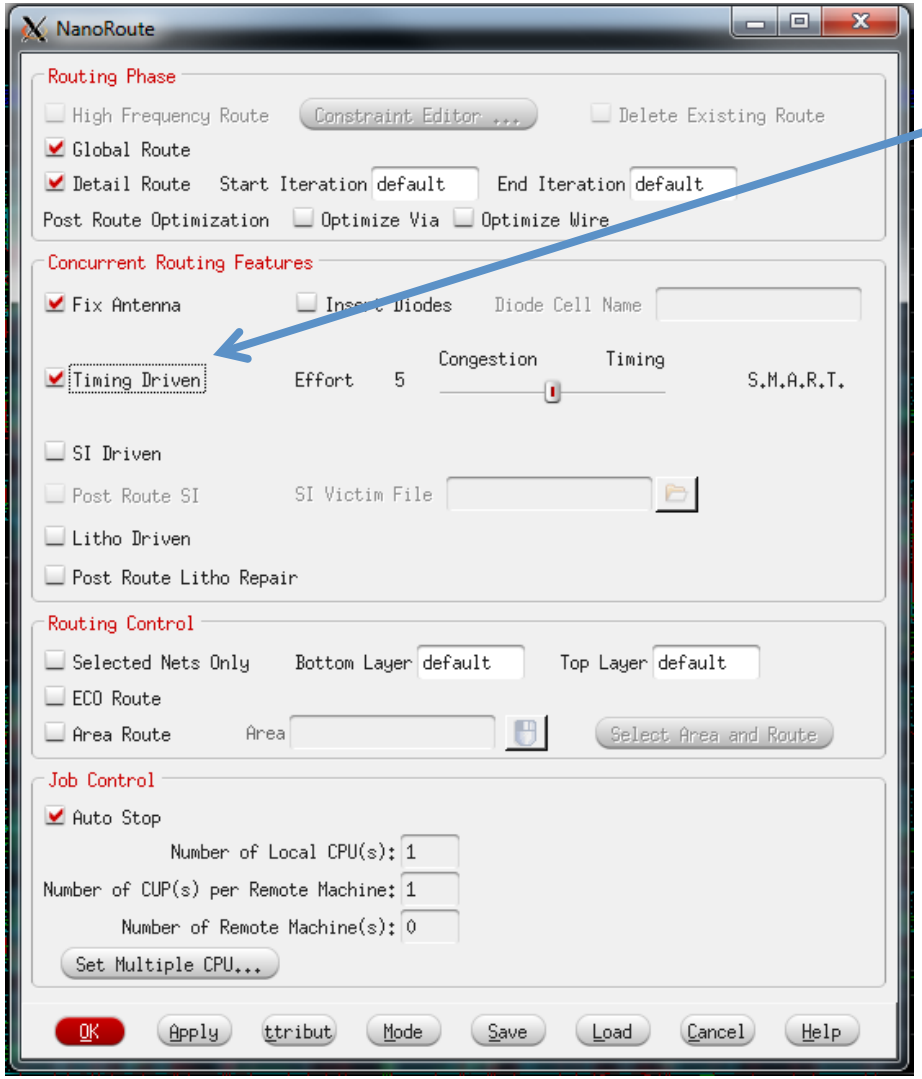
Note: if you look at your shell or at the postCTS log that was created in your timingReports directory you should see some improvement in WNS. In a large design the change will be more dramatic.

```
------------------------------------------------------------
          timeDesign Summary
------------------------------------------------------------

+--------------------+---------+---------+---------+---------+---------+--------
--+
|     Setup mode     |   all   | reg2reg | in2reg  | reg2out | in2out  | clkgate
 |
+--------------------+---------+---------+---------+---------+---------+--------
--+
|           WNS (ns):| 43.766  | 43.766  | 45.102  | 48.955  |   N/A   |   N/A
 |
|           TNS (ns):|  0.000  |  0.000  |  0.000  |  0.000  |   N/A   |   N/A
 |
|    Violating Paths:|    0    |    0    |    0    |    0    |   N/A   |   N/A
 |
|          All Paths:|   135   |   134   |   132   |    1    |   N/A   |   N/A
 |
+--------------------+---------+---------+---------+---------+---------+--------
--+

+----------------+-------------------------------+------------------+
|                |              Real             |       Total      |
|    DRVs        +------------------+------------+------------------|
|                |  Nr nets(terms)  | Worst Vio  |  Nr nets(terms)  |
+----------------+------------------+------------+------------------+
|   max_cap      |      2 (2)       |   -0.043   |      2 (2)       |
|   max_tran     |      0 (0)       |   0.000    |      0 (0)       |
|   max_fanout   |      0 (0)       |     0      |      0 (0)       |
+----------------+------------------+------------+------------------+

Density: 60.039%
Routing Overflow: 0.00% H and 12.19% V
------------------------------------------------------------
```

In this example there was no change to the TNS or WNS because there was no negative slack

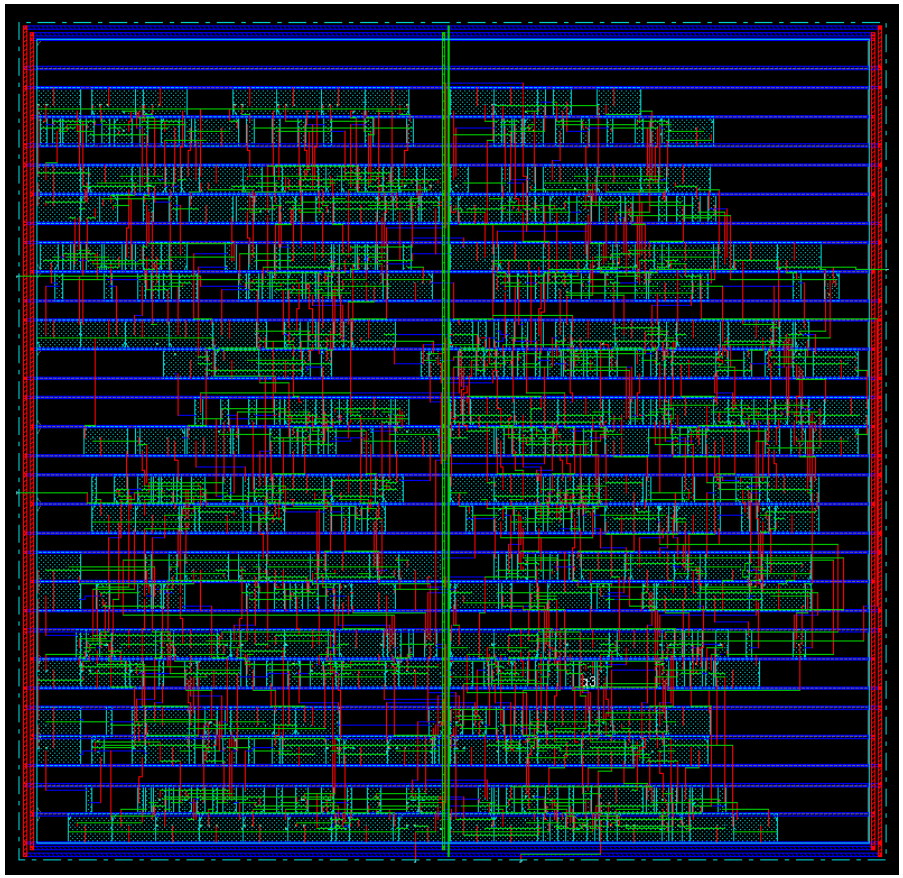
Now that the design has a clock tree you can perform final routing of the design.

Select **Route → NanoRoute → Route** to invoke the



Select Timing Driven
-you can adjust the **Effort** slider to tell the tool how much effort to spend on meeting timing and how much to spend on reducing congestion.

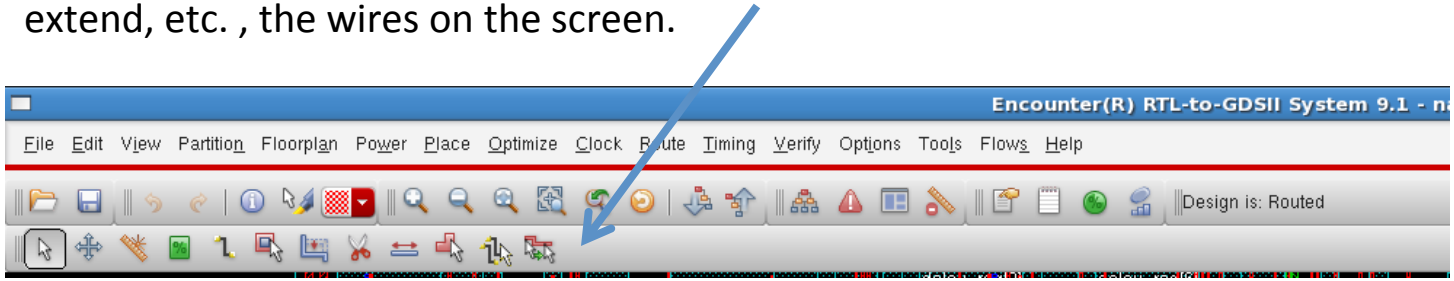
When the tool is done your design will look like the design below. Also take a look at your shell and make sure there are 0 violations and 0 fails. If you want to see your report after closing the shell you can find it in the default directory. The encounter.log file maintains a record of everything that appears in the shell.



```
#Complete Detail Routing.
#Total wire length = 71925 um.
#Total half perimeter of net bounding box = 65445 um.
#Total wire length on LAYER metal1 = 5328 um.
#Total wire length on LAYER metal2 = 34288 um.
#Total wire length on LAYER metal3 = 32309 um.
#Total number of vias = 2399
#Up-Via Summary (total 2399):
#
#-----------------------
#  Metal 1          370
#  Metal 2         2029
#-----------------------
#                  2399
#
#Total number of DRC violations = 0
#Total number of violations on LAYER metal1 = 0
#Total number of violations on LAYER metal2 = 0
#Total number of violations on LAYER metal3 = 0
#detailRoute Statistics:
#Cpu time = 00:00:01
#Elapsed time = 00:00:01
#Increased memory = 0.00 (Mb)
#Total memory = 326.00 (Mb)
#Peak memory = 359.00 (Mb)
#
```

It is possible you might have an error associated with a pin after routing. You can fix it by starting over and routing again to see whether you still have the error, or you can zoom into the error and move things around to fix the error. If you want to do this, you can zoom into the correct part of the layout using the mouse wheel.

Once you are there you can use the **Tool Widget** in the upper left of the screen to move, add extend, etc. , the wires on the screen.

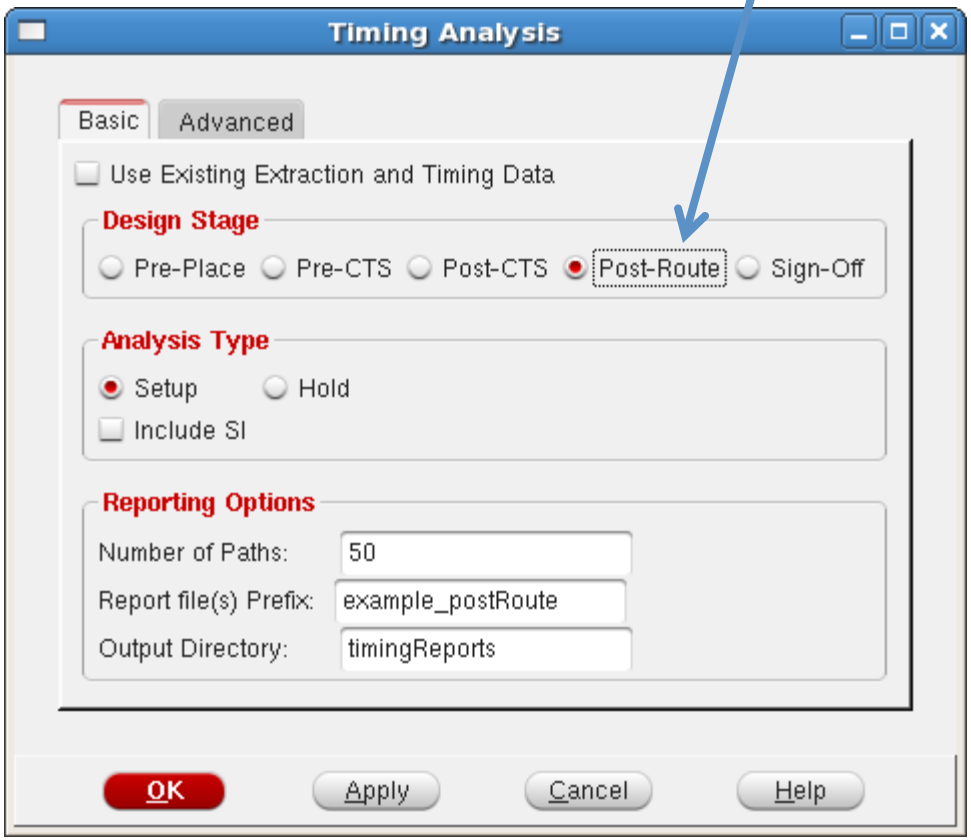


Alternatively you can wait to fix things in Virtuoso after you've finished with SOC Encounter.

Post-Route Optimization.

You can now run one or more optimization steps. In this case we will run Timing Optimization one more time but this time we will use post route optimization.

Select **Timing → Report Timing** Select **Post Route** and click **OK**



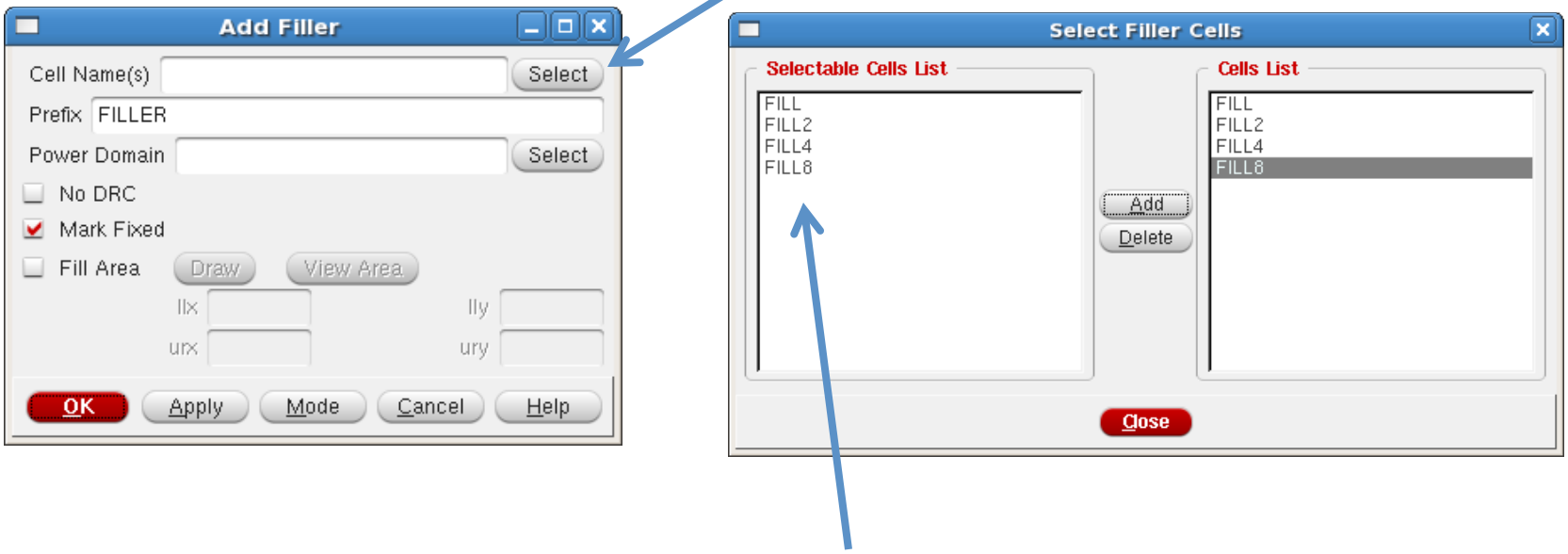
Note: This is the final stage of optimization. If at this point your design still has more than 200 ps of clock skew you should start over at the synthesis stage and modify the period of your clock.

This is another good time to save your project

Adding Filler Cells

After the final optimization is complete, you need to fill the gaps in the cell row with filler cells. These are the cells in your library that have no active circuits in them, just power and ground wires and Nwell layers that match your cell template.

Select **Place → Physical Cell → Add Filler** click on select

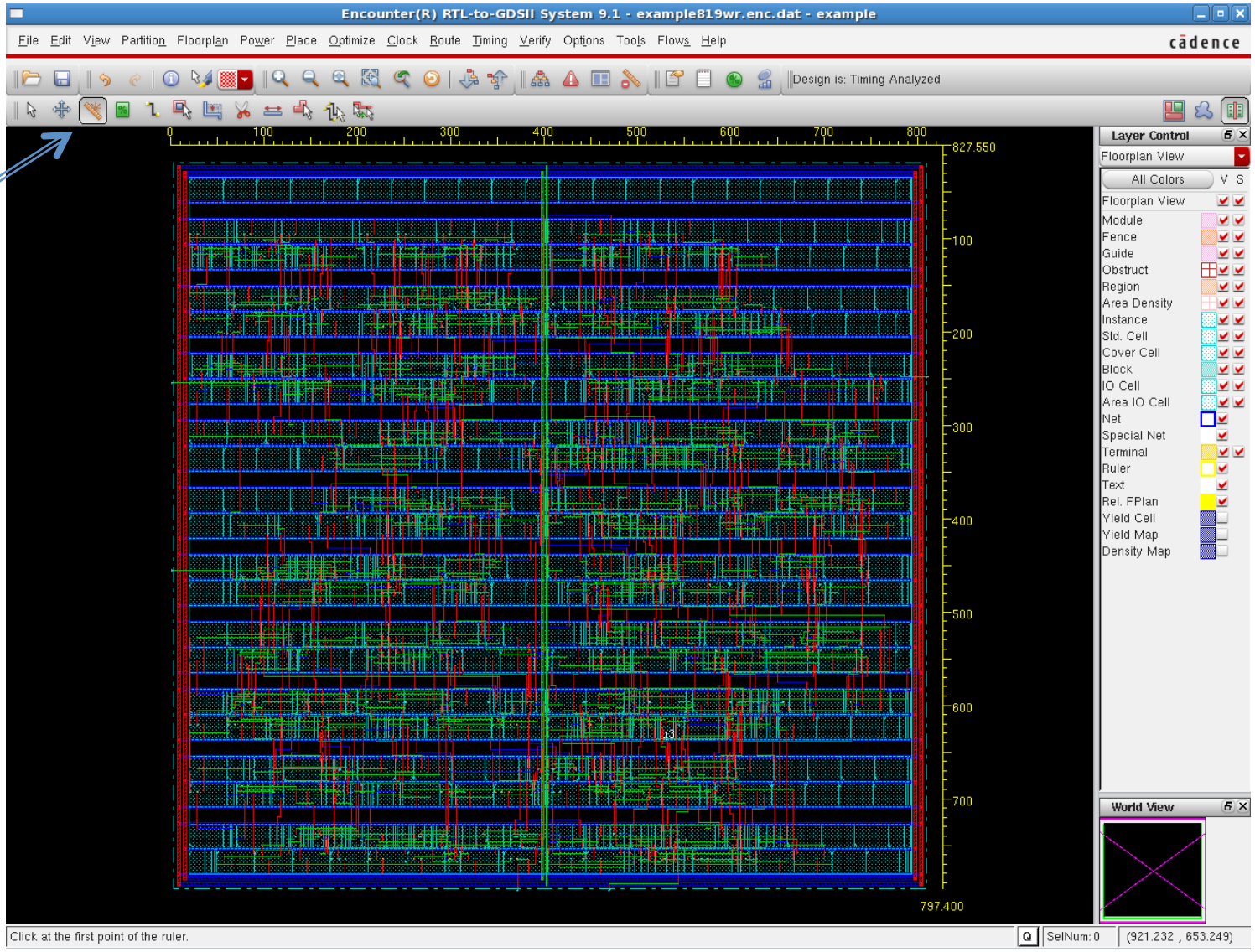


Select Fill, Fill2 , Fill 4, Fill 8. Click **Close**

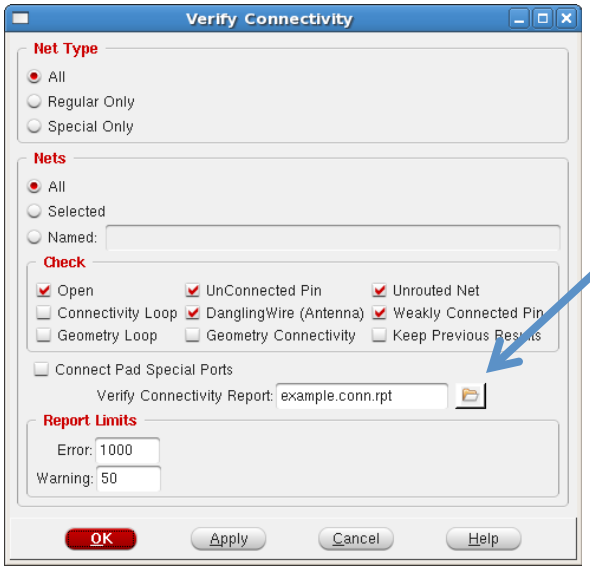
Once you click close the Names you selected will appear in the cell name(s) field. Make sure all the cells you selected are there and click **OK**

Your project should look similar to the one below. Use the **Ruler** in the tools widget to verify that your design meets size constraints**. Don't forget to save a copy of your completed project.**

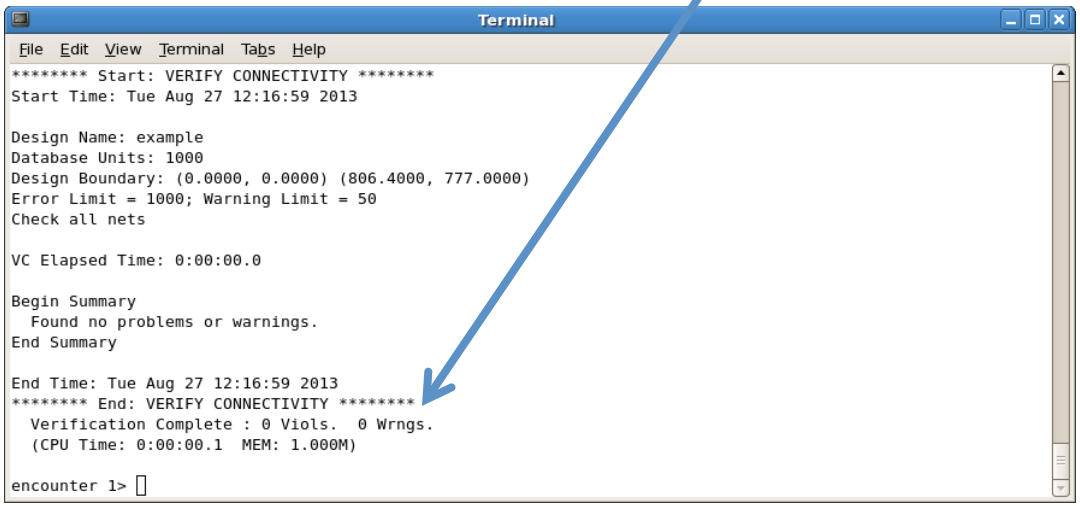
Ruler



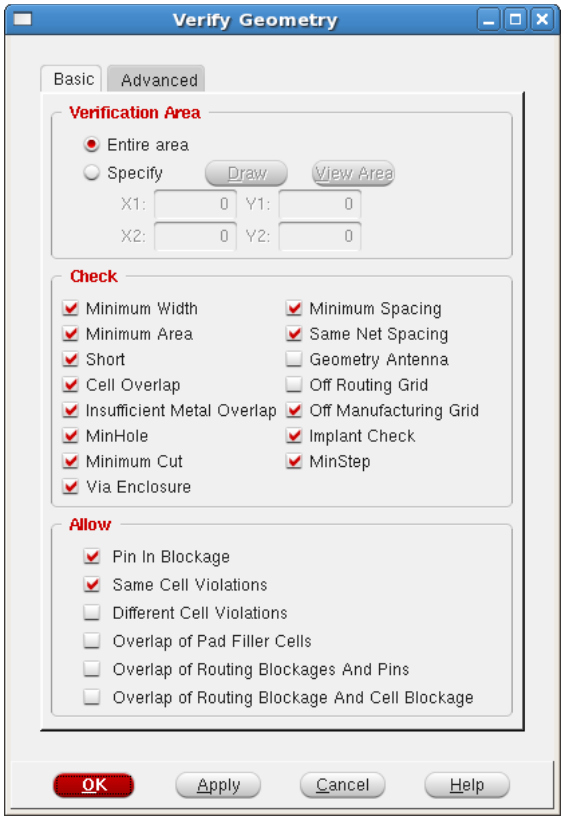
Verify Connectivity checks that all connections specified in your original structural netlist have been made successfully. To check this use **Verify → Verify Connectivity** menu choice. Note the name in the **Verify Connectivity Reports** this is where the results will be saved. You can leave the default or you can change it to whatever your wish as long as it has the .rpt extension.

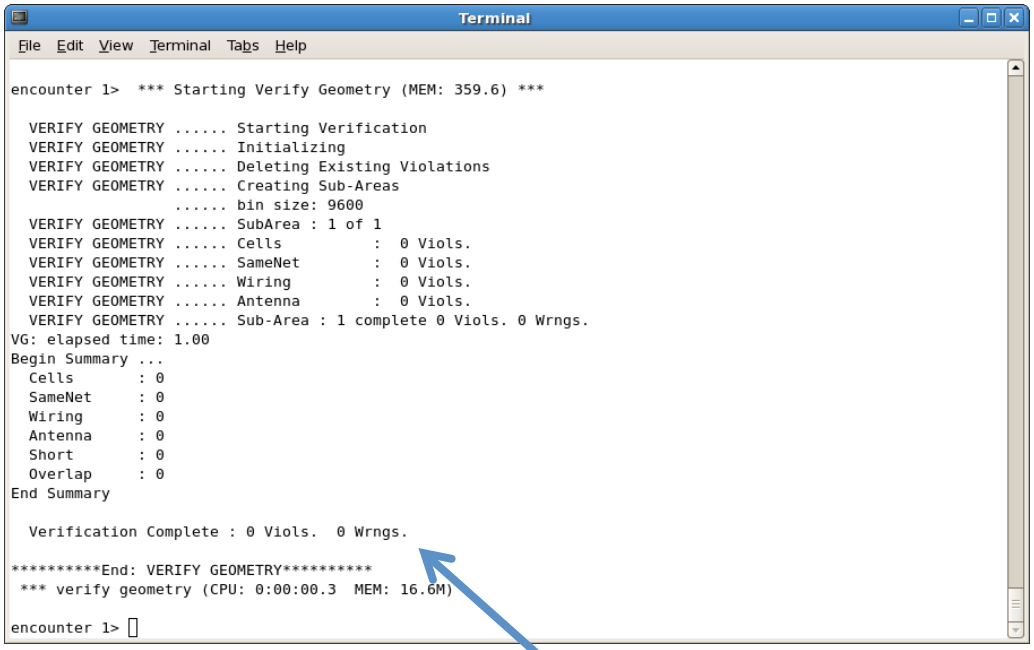


This should return a result in the shell(or in your report) that says that there are no problems or warnings related to connectivity. If there are problems, it's likely that routing congestion cause the problem. You could try a new route, or a new placement, or go all the way back to a new floorplan with a lower core utilization percentage or more space between rows for routing channel.



The **Verify Geometry** command will run a DRC check. DRC's are Design Rules Checks and basically tests that your spacing and widths are good. As well as standard cell library placement of rows.
To check this use **Verify → Verify Geometry.** Leave all of the defaults and click **OK**





There should be 0 Violations and 0 Warnings. If there are violations or warnings it is likely that a step was skipped. For example forgetting to run the power lines, add filler, or insert a necessary spacing will cause Violations.

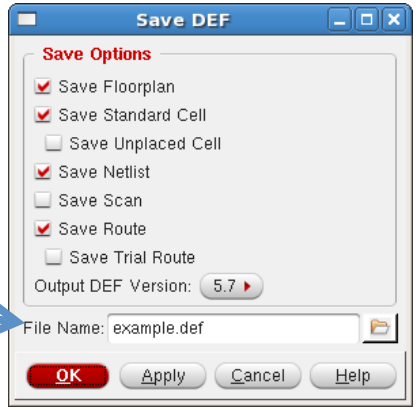
Note: This is not a substitute for a full DRC check in Virtuoso. This check only runs on the routing and abstract views, and has only a subset of the full rules that are checked in Virtuoso. But if you have errors here, you should try to correct them here before moving on to the next step.

Saving and Exporting the Placed and Routed Cells

Now that you have a completely placed, routed, optimized, and filled cell, you need to save it and export it.

Exporting a DEF file:

- **File → Save → DEF**
- You can rename the file or leave the default name. This file will be used to import your design in to Verilog



Exporting a structural Verilog File:

- **File → Save → Netlist**
- You can rename the file or leave the default.
- Because we had Encounter generate a clock tree and run a number of optimization steps the circuit that we've ended up with is not the same circuit that we started with. So, if you want to compare the layout to a schematic, you need the new structural file.