

Tutorial – Cadence Innovus

Dae Hyun Kim

EECS
Washington State University

Cadence Innovus

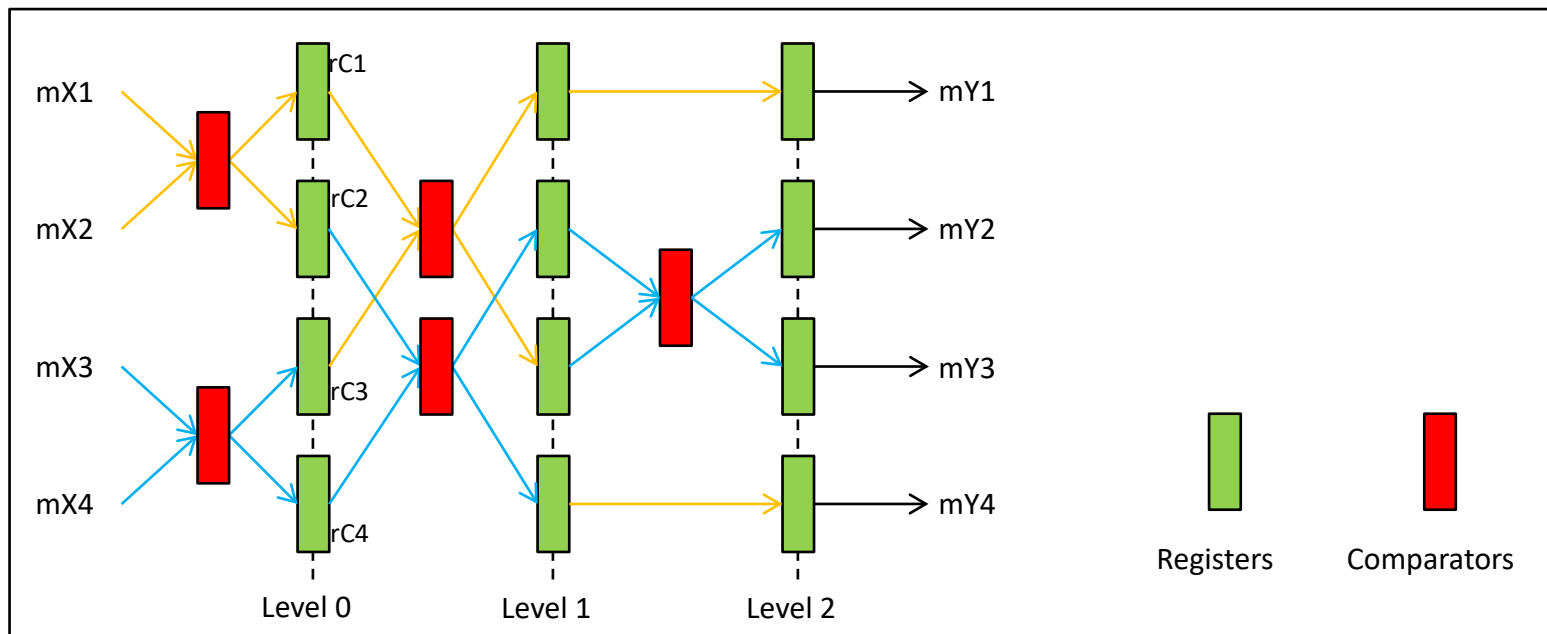
- Used for automatic physical design (P&R)
- Performs
 - Floorplanning
 - Placement
 - Clock-tree synthesis (CTS)
 - Routing
 - Optimization
 - Analysis (timing, power, ...)
 - ...

Tutorial

- Download the following file into your work directory.
 - `wget http://eecs.wsu.edu/~daehyun/teaching/2023_EE434/lab/tutorial_innovus.zip`
- Unzip it.
- Source
 - `source ictools_generic.sh`
 - `source cadence_innovus.sh`
- How to run Innovus
 - Innovus
- GUI: We will use GUI, so you should enable it.

Benchmark

- VQS64_4 (four-input 64-bit pipelined quick sort)
 - input [63:0] mX1, mX2, mX3, mX4; // four input data
 - input mCLK; // clock
 - output [63:0] mY1, mY2, mY3, mY4; // four output data

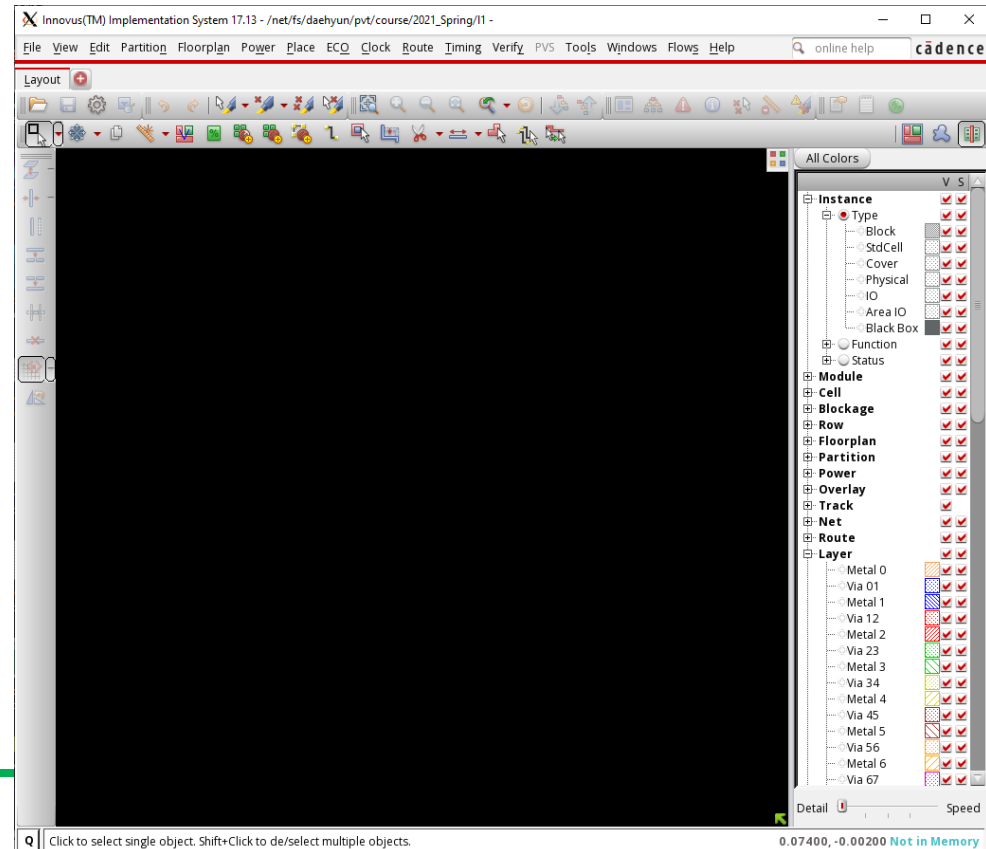


Physical

1. Chip outlining
2. P/G network design
3. Placement
4. Pre-CTS optimization
5. CTS
6. Post-CTS optimization
7. Routing
8. Post-routing optimization

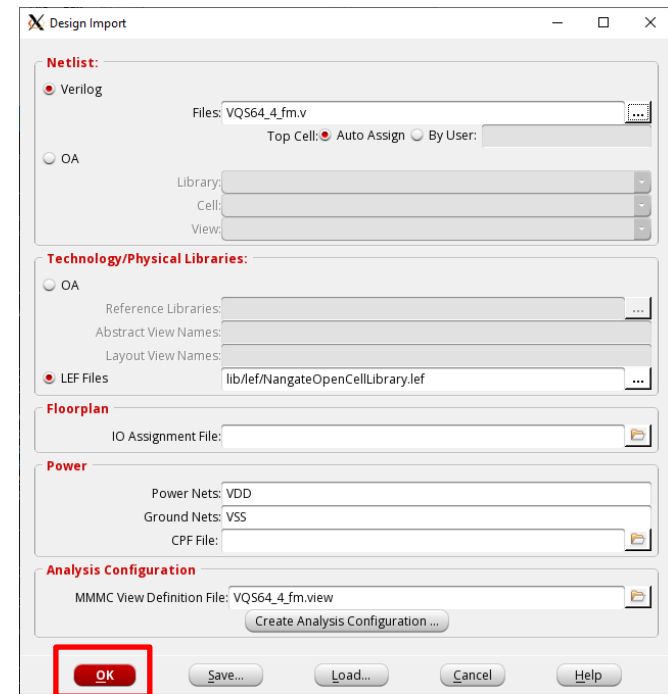
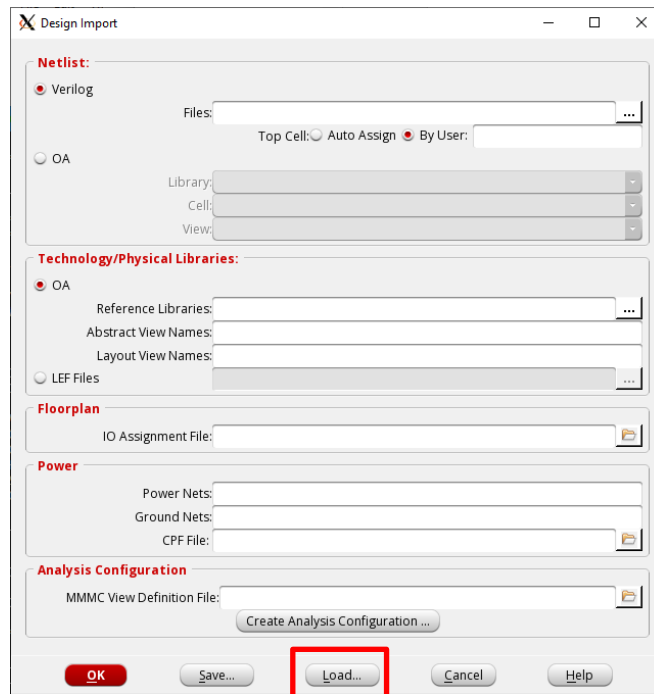
1. Chip Outlining

- Run Innovus.
 - innovus
- See the terminal window. It's similar to Synopsys DC. You can use the GUI or the terminal to do something.



1. Chip Outlining

- Click “File” → “Import Design...”.
 - (The GUI might be a bit slow, so be patient.)
- In the “Design Import” window, click “Load...” and choose “VQS64_4_m.globals”. This will automatically fill up the settings. Then, click “OK”.



1. Chip Outlining

- See the terminal for Innovus messages. There might be some Warning messages. You can ignore them.
- In the Innovus main window, press “f” to see the outline of the layout.
- Innovus automatically computes and prepares the layout area.
- Let’s modify the layout area.
- In the main window, click “Floorplan” → “Specify Floorplan...”.
- Set the core utilization to 0.6.
- Set the core-to-left, core-to-top, core-to-right, and core-to-bottom to 5.0.
- Then, click OK. (See the next page)

1. Chip Outlining

Specify Floorplan

Basic Advanced

Design Dimensions

Specify By: ☒ Size ☐ Die/IO/Core Coordinates

☒ Core Size by: ☒ Aspect Ratio: Ratio (H/W): 1345392877

☒ Core Utilization: 0.6

☐ Cell Utilization: 0.699971

☐ Dimension: Width: 88.45 Height: 86.8

☐ Die Size by: Width: 88.45 Height: 86.8

Core Margins by: ☒ Core to IO Boundary

☐ Core to Die Boundary

Core to Left: 5 Core to Top: 5

Core to Right: 5 Core to Bottom: 5

Die Size Calculation Use: ☐ Max IO Height ☒ Min IO Height

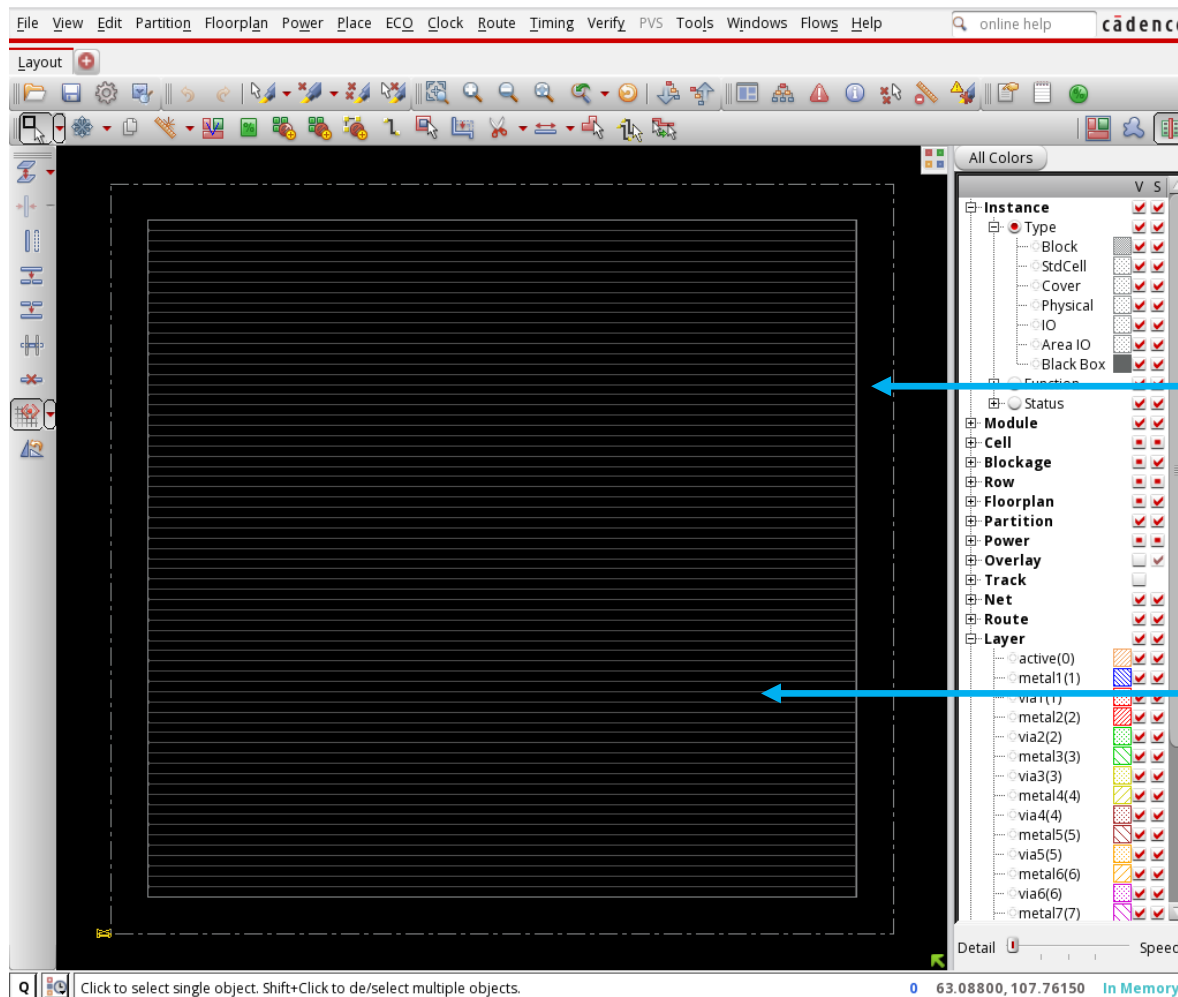
Floorplan Origin at: ☒ Lower Left Corner ☐ Center

Unit: Micron

OK Apply Cancel Help

1. Chip Outlining

- Now, you will see the following window.



Power/Ground rings will be laid out in this area.

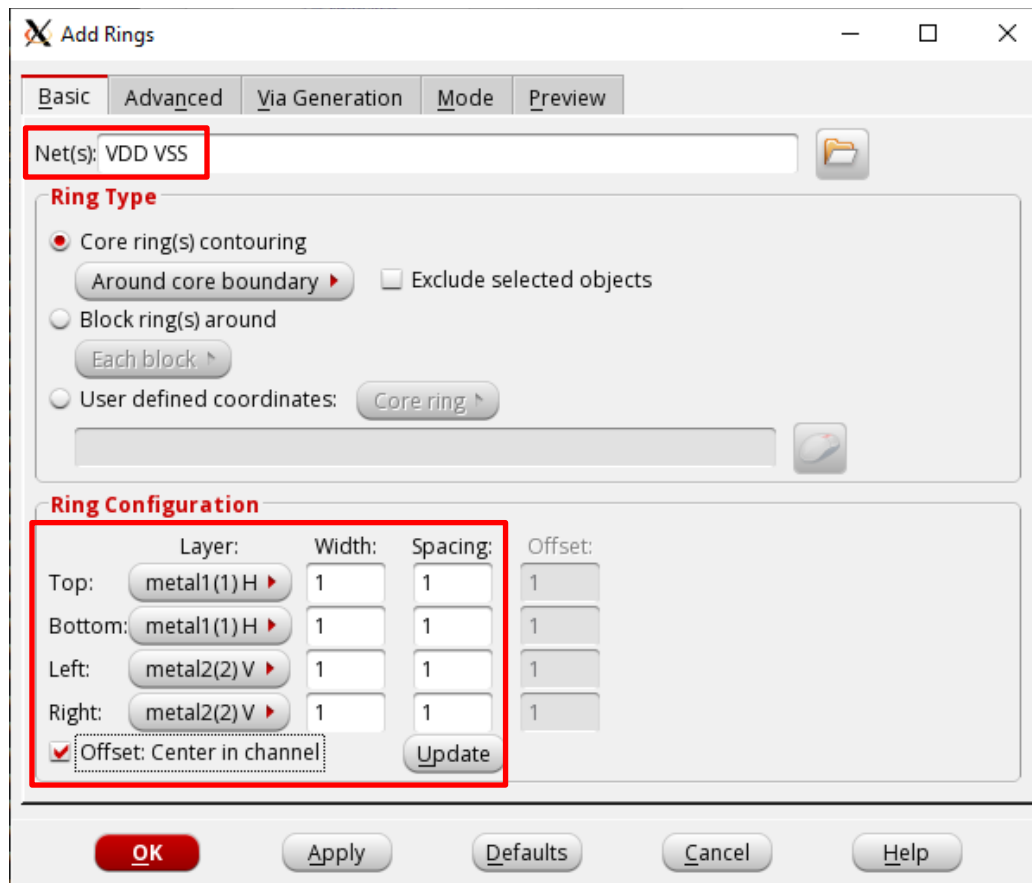
Std. cells will be placed in this core area.

How to Save the Design

- Let's save the current design.
- In the terminal, run the following command to save the current design into “test_01_floorplan.enc”.
`innovus #> saveDesign test_01_floorplan.enc`
- Later on, you can load the design as follows.
 - Run Innovus, click File → Restore Design → Data Type: Innovus → select the .enc file.

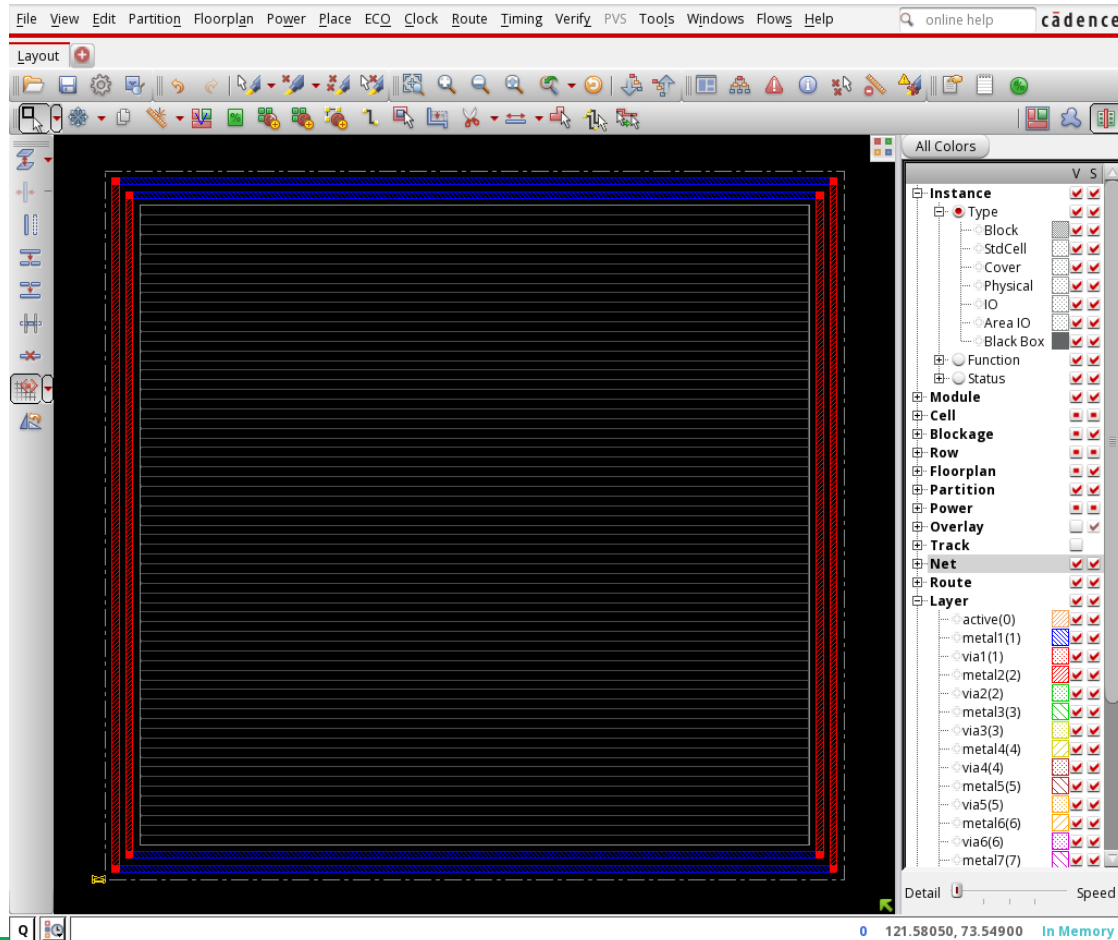
2. P/G Network Design

- Click “Power” → “Power Planning” → “Add Rings...”.



2. P/G Network Design

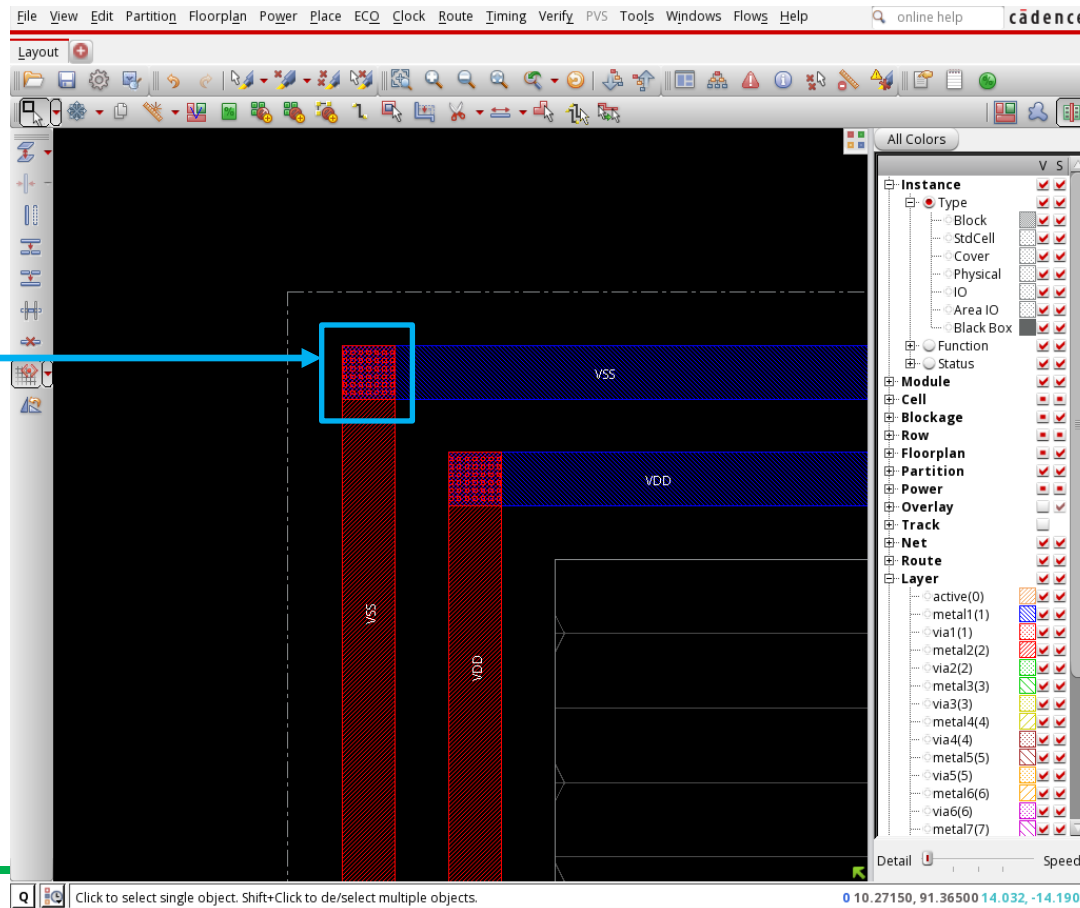
- Fill in the input boxes as shown in the previous page and click OK. Now you can see the power and ground rings.



2. P/G Network Design

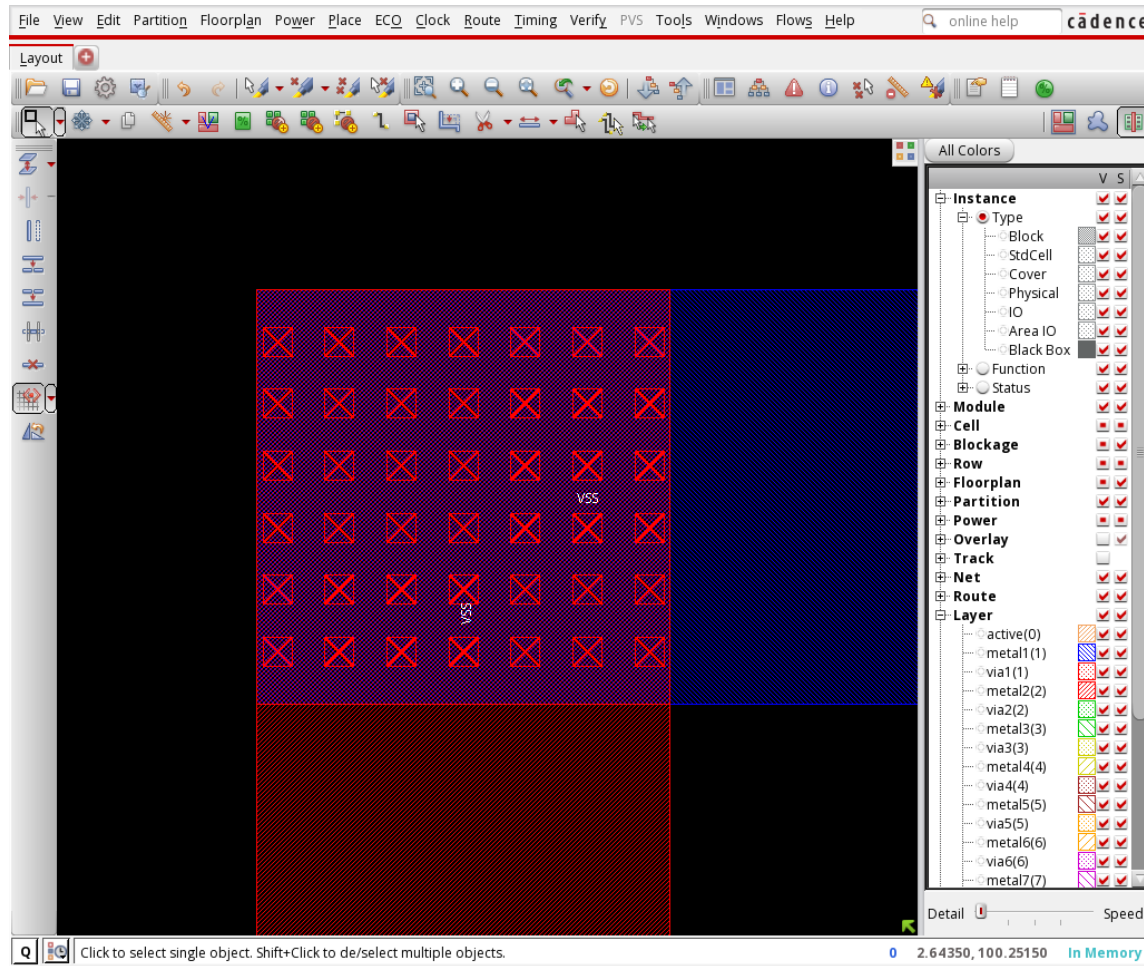
- Zoom in the top-left corner (Mouse right click – hold – drag – release). As shown below, the outer ring is VSS and the inner ring is VDD. Blue: Metal 1. Red: Metal 2. Zoom in the via arrays.

Zoom in



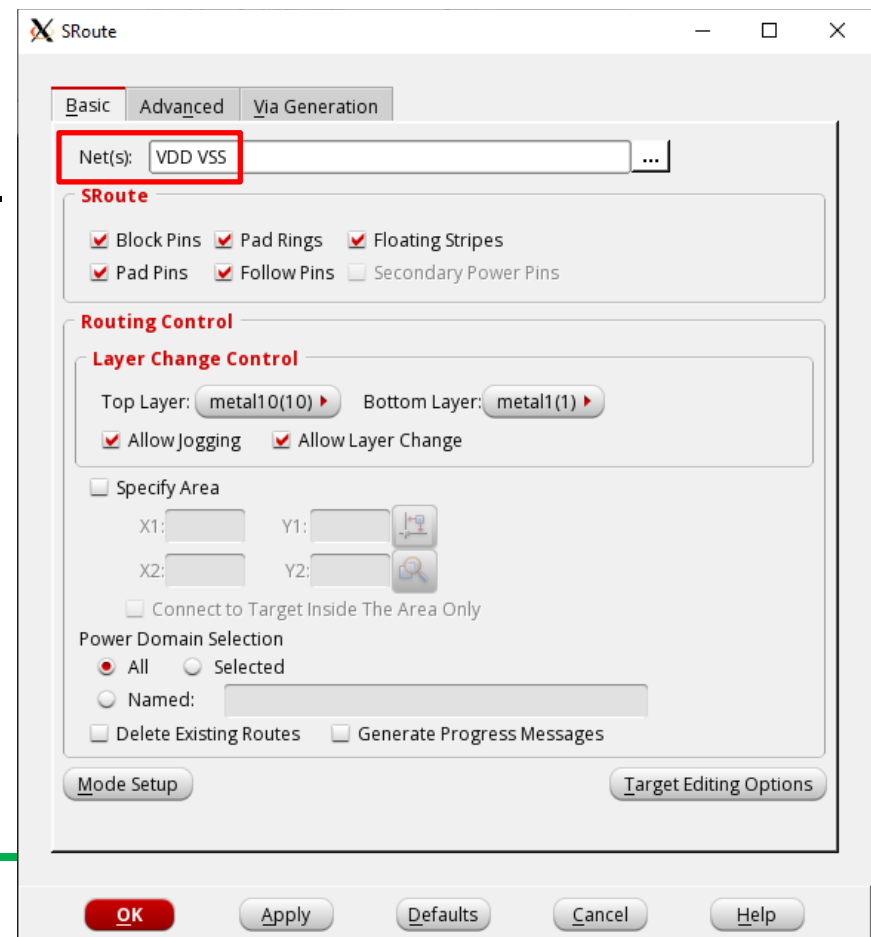
2. P/G Network Design

- The “X” squares are vias connecting the M1 and M2 wires.



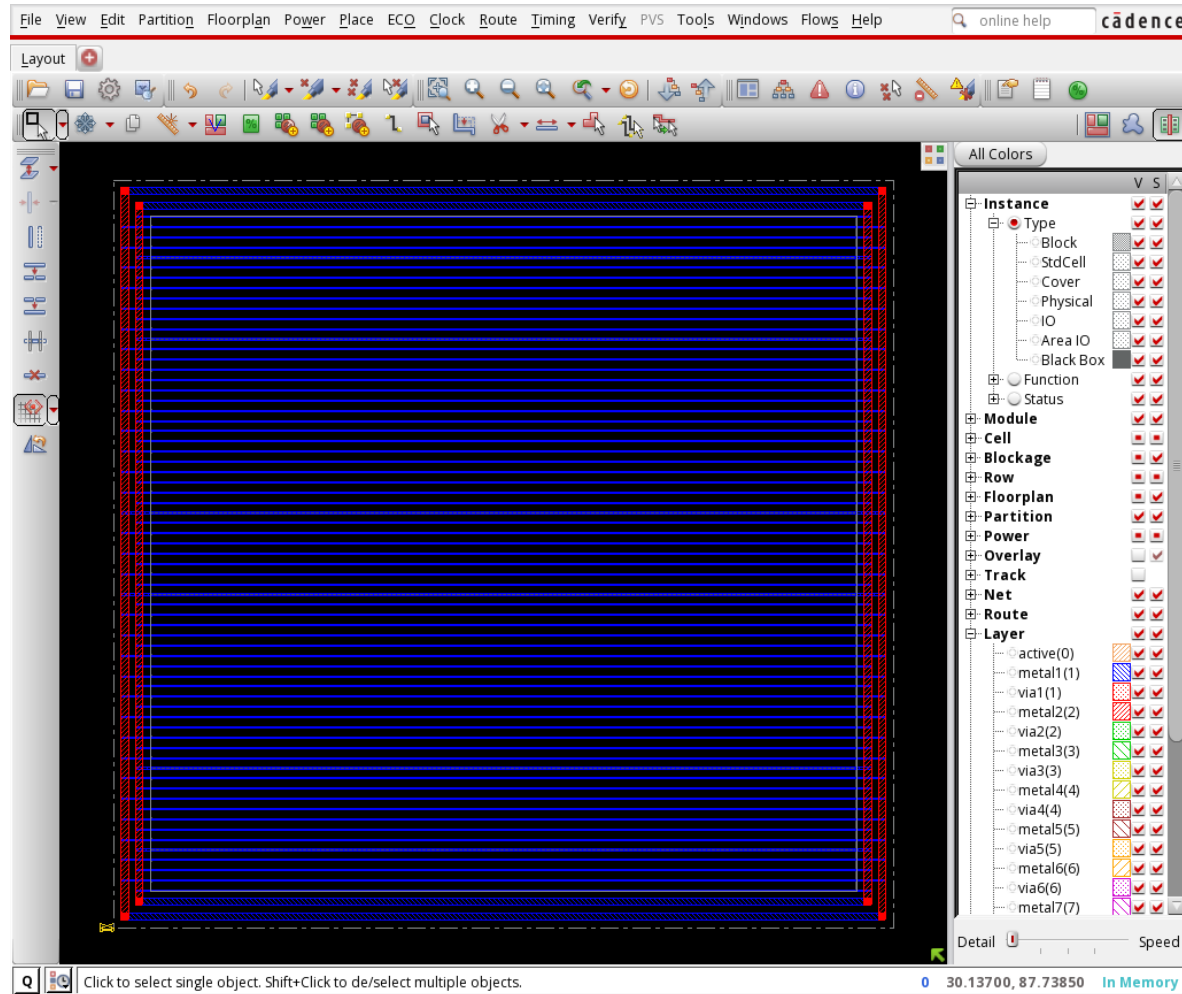
2. P/G Network Design

- Press “f” to zoom out to the full design.
- Now, we will draw power/ground stripes to connect the P/G rings to standard cells.
- Click “Route” → “Special Route...”.



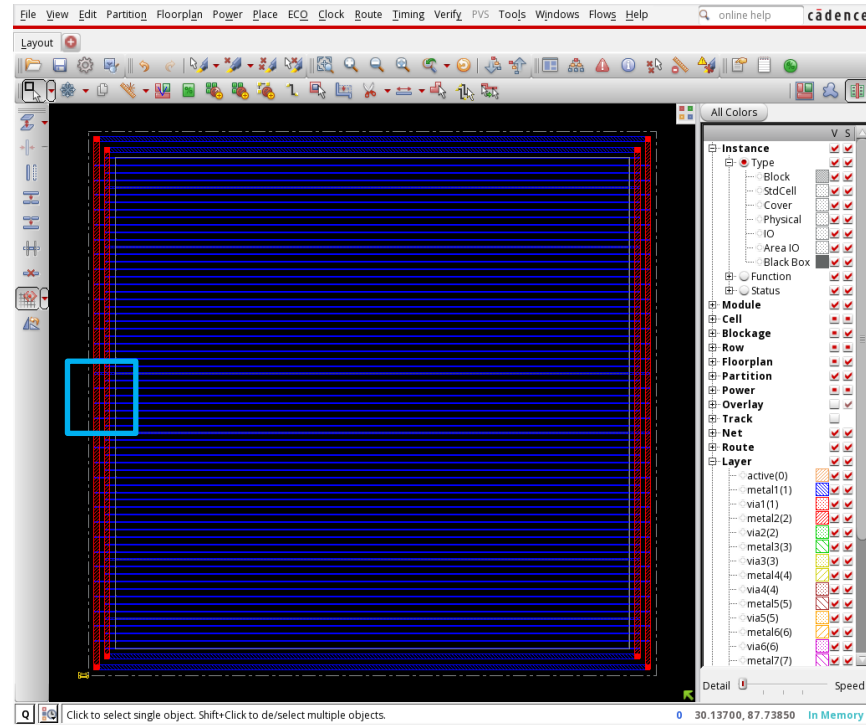
2. P/G Network Design

- P/G network



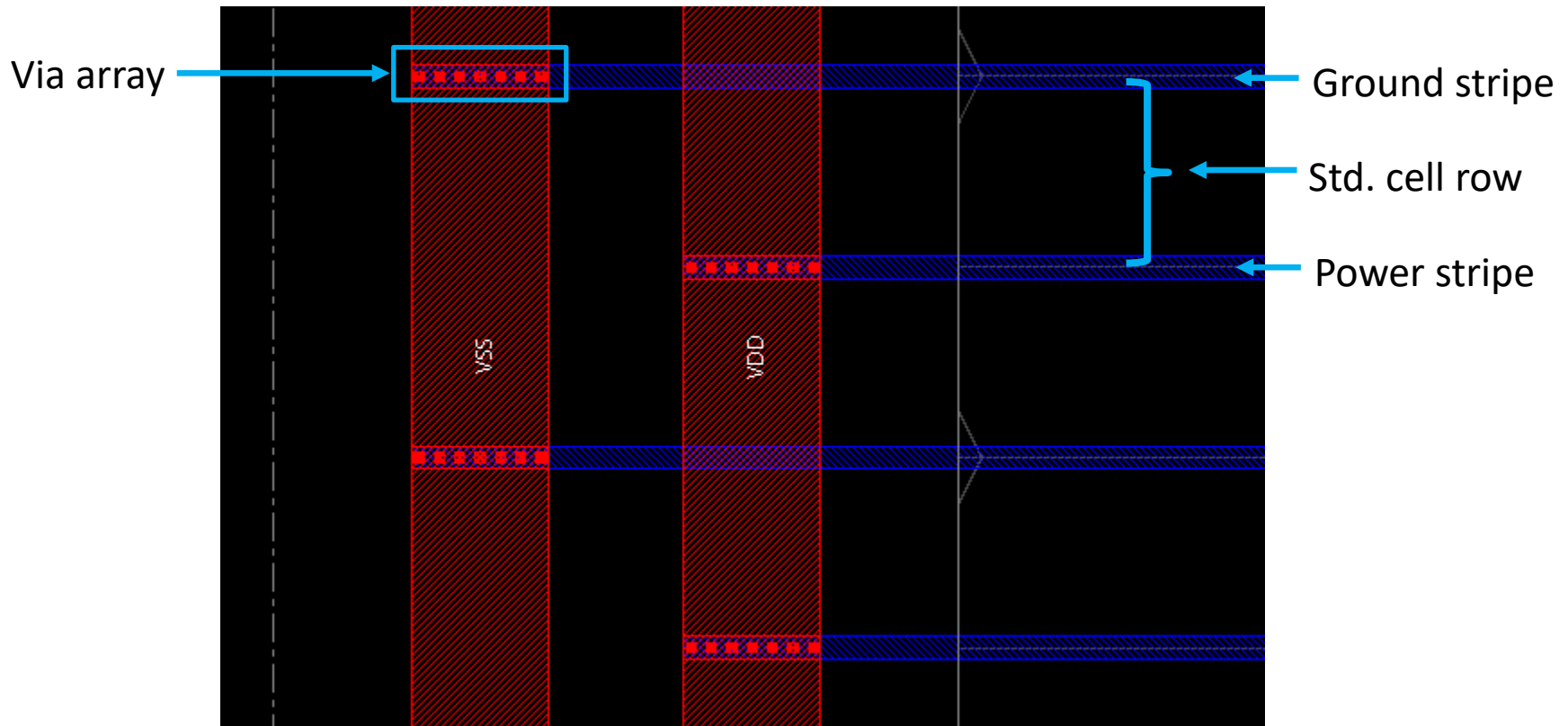
2. P/G Network Design

- saveDesign test_02_pg.enc
- Zoom in the following area.



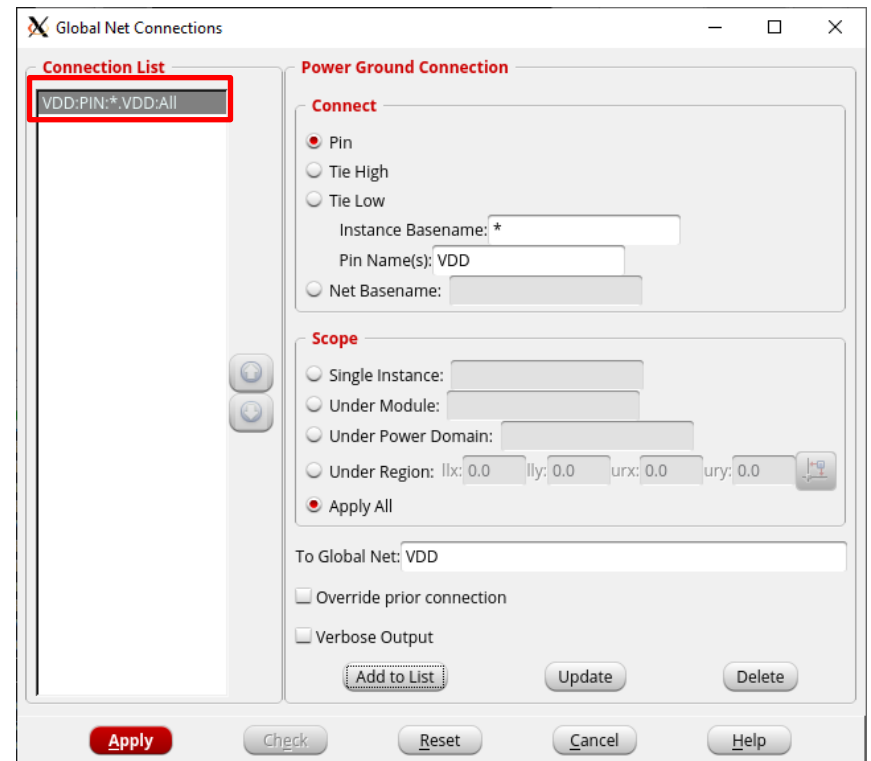
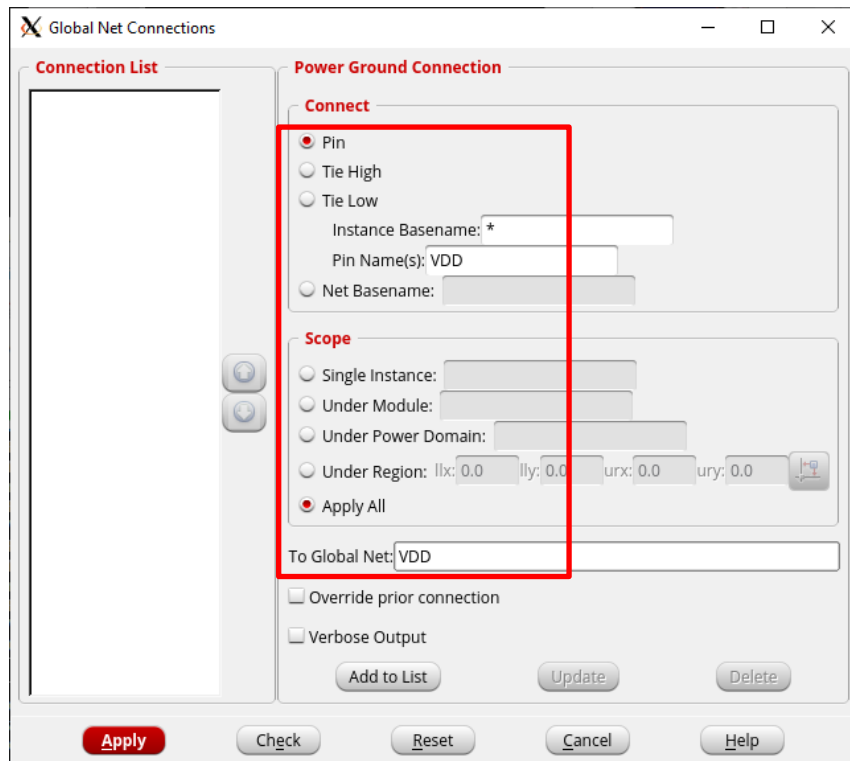
2. P/G Network Design

- As you see, the P/G stripes are alternating between VDD and VSS. See the vias.



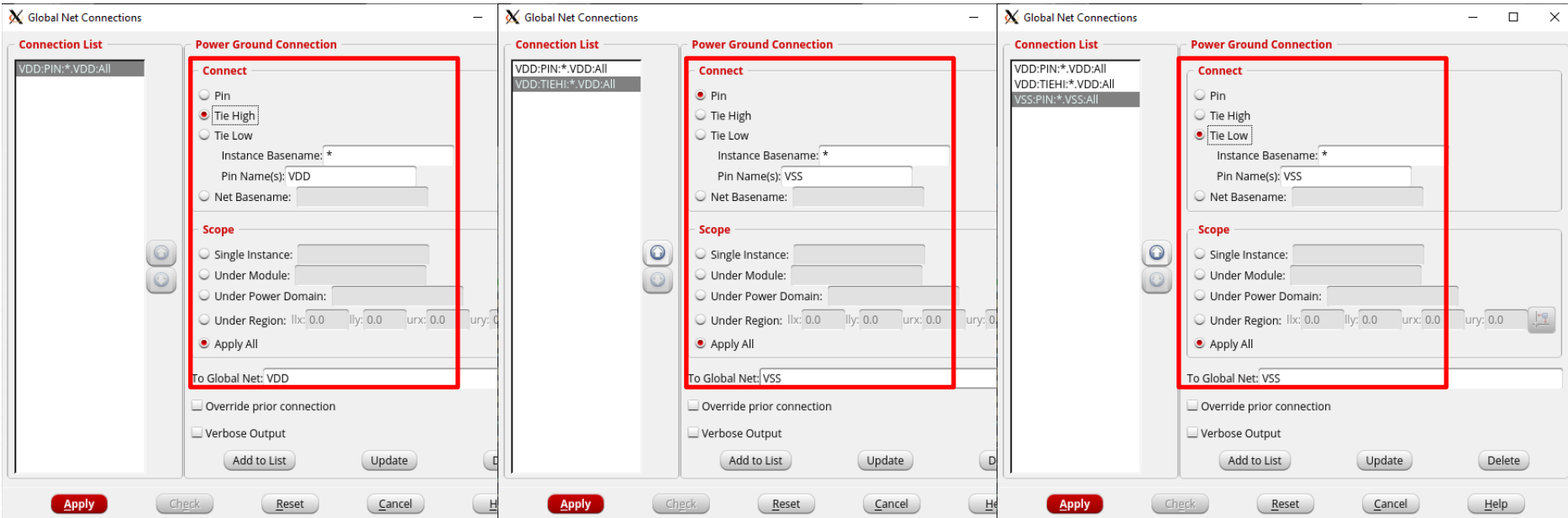
2. P/G Network Design

- Let's add some more settings to the design for P/G network.
- “Power” → “Connect Global Nets...”
- Fill in the blanks and click “Add to List”.



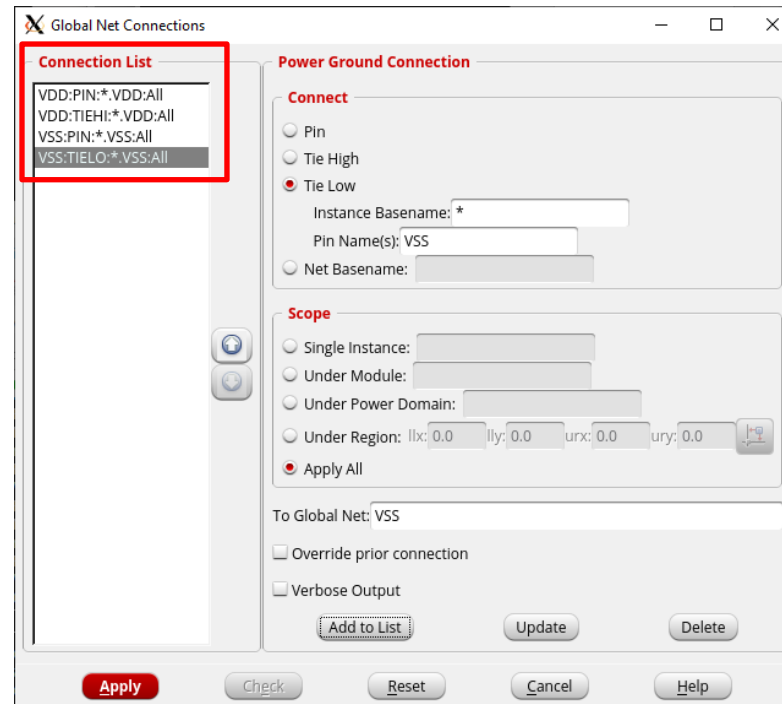
2. P/G Network Design

- Repeat it for the followings.



2. P/G Network Design

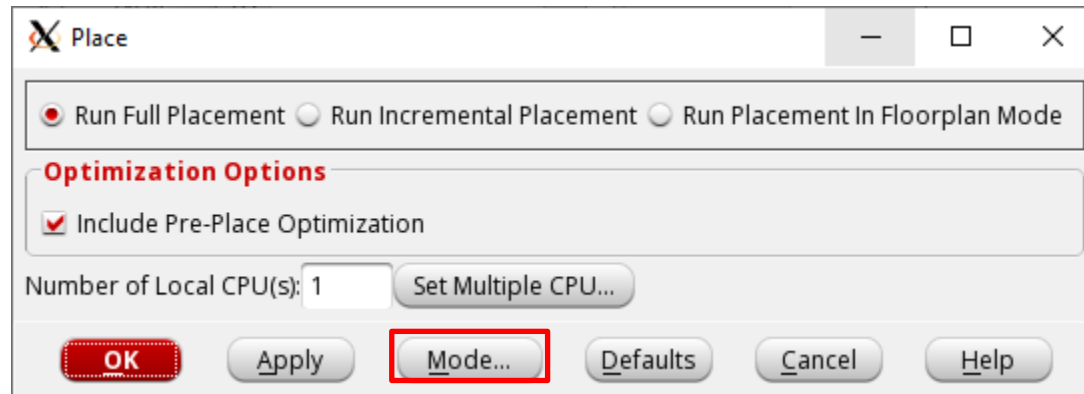
- After that, it should look like this.



- Now, click “Apply”.
- Then close the window (click “Cancel” or “X”).

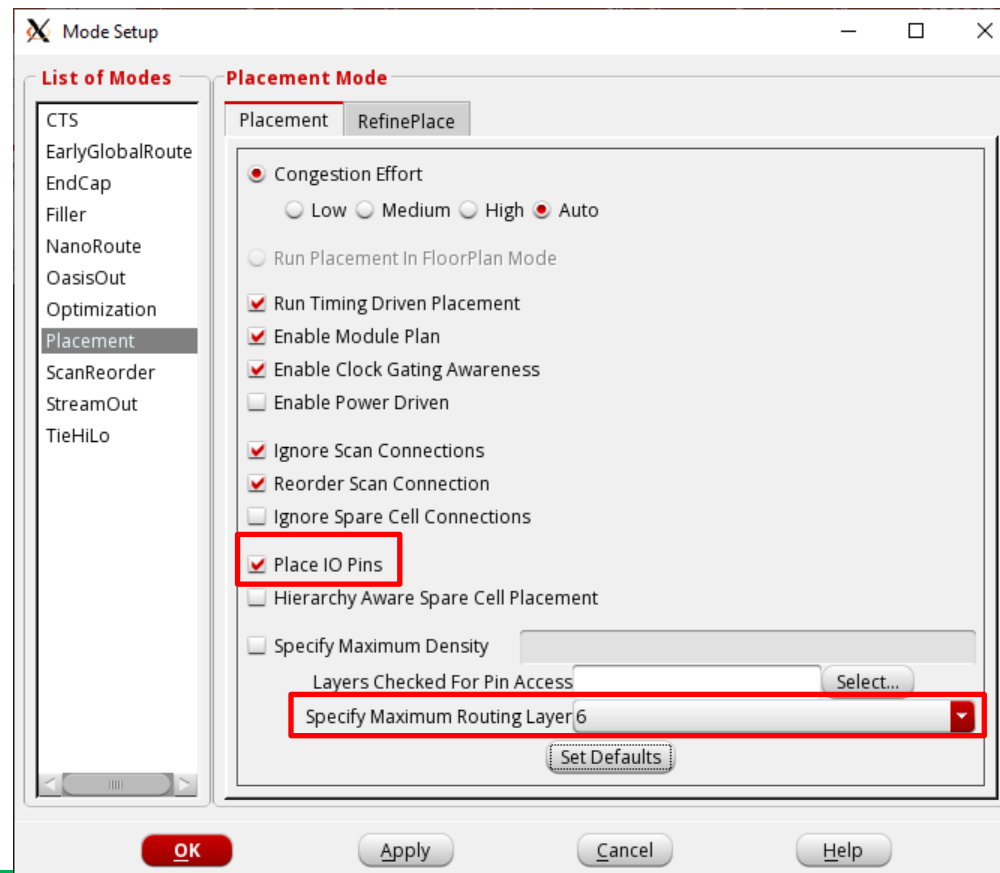
3. Placement

- Let's place the instances (cells).
- In the main window, click “Place” → “Place Standard Cell”.
- In the “Place” window, click “Mode”.



3. Placement

- Turn on “Place IO Pins”. Set the “Specify Maximum Routing Layer” to 6. We will use only six metal layers. Click OK. In the “Place” window, click OK.

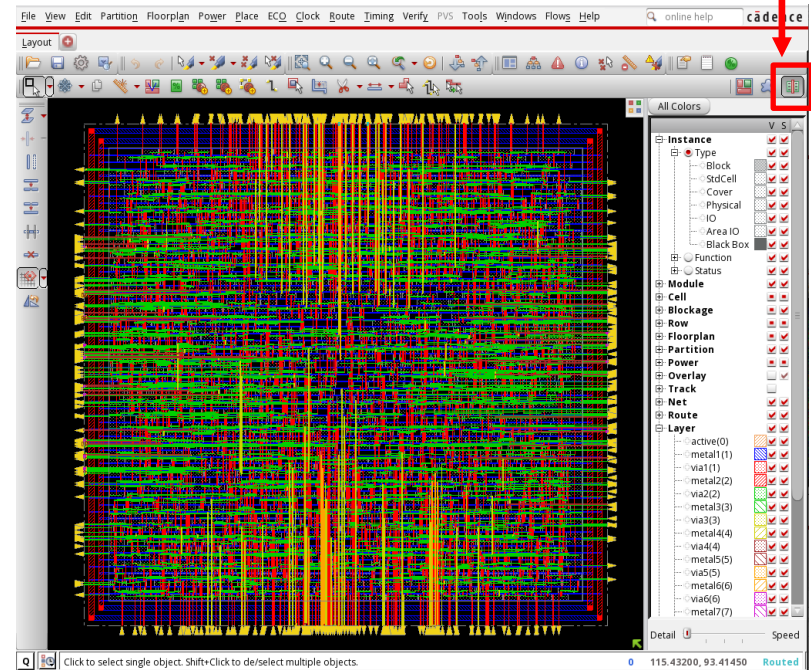


3. Placement

- It shows placement and trialRoute results. trialRoute is just a quick routing for an estimation of some design metrics.
- See the terminal. It shows some more information.
 - Total wire length: 32,540.21um

Click this button
if you don't see the layout.

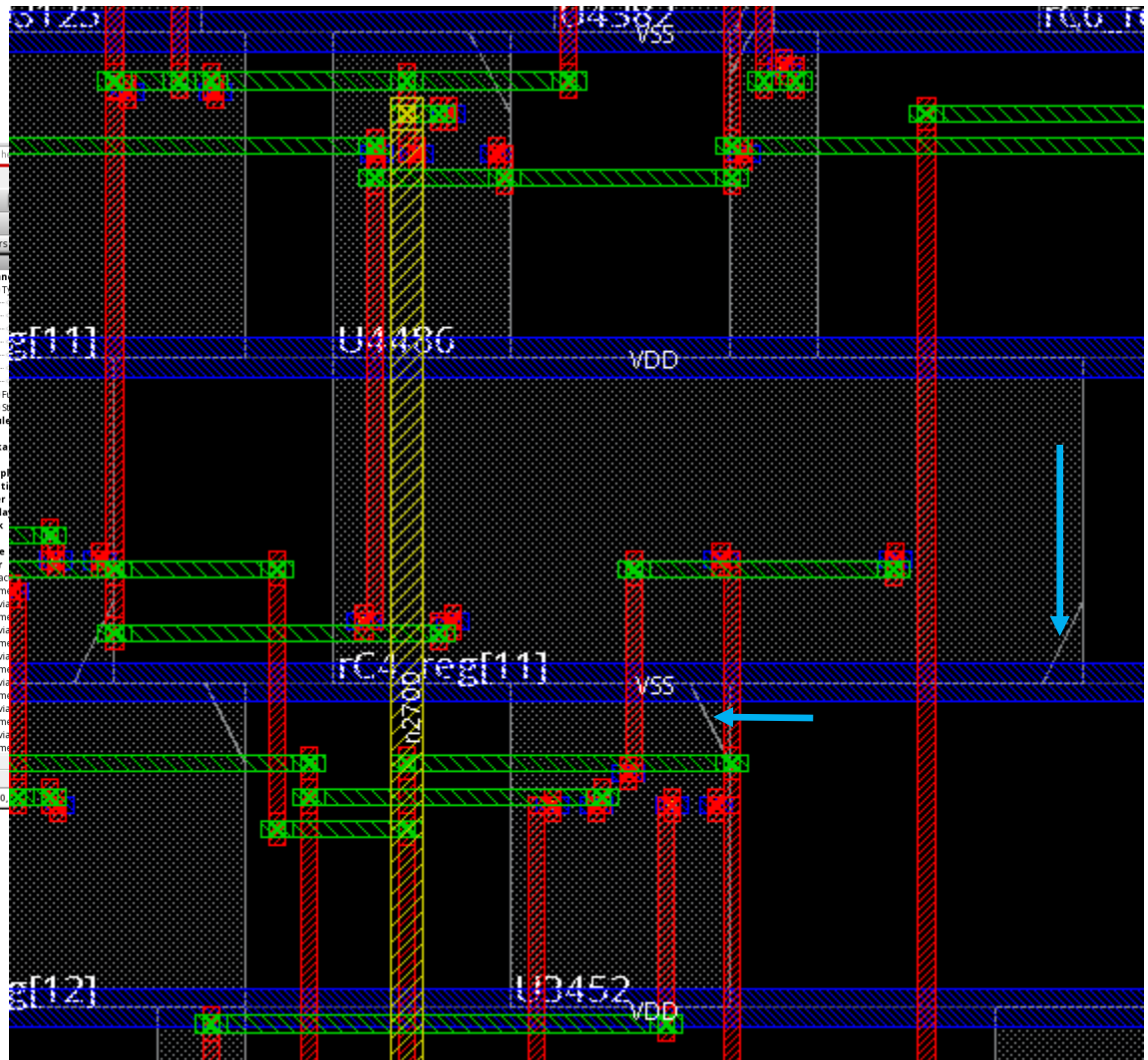
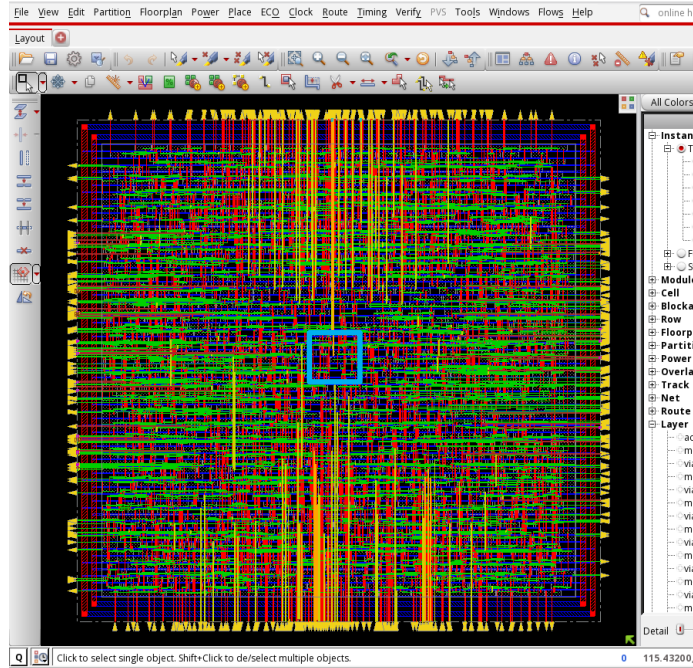
```
Layer1(metal1)(F) length: 0.000000e+00um, number of vias: 8752
Layer2(metal2)(V) length: 1.563008e+04um, number of vias: 12901
Layer3(metal3)(H) length: 1.402659e+04um, number of vias: 355
Layer4(metal4)(V) length: 1.713939e+03um, number of vias: 108
Layer5(metal5)(H) length: 8.258645e+02um, number of vias: 46
Layer6(metal6)(V) length: 3.437350e+02um, number of vias: 19
Total length: 3.254021e+04um, number of vias: 22181
```



- Save it.
 - saveDesign test_03_pl.enc

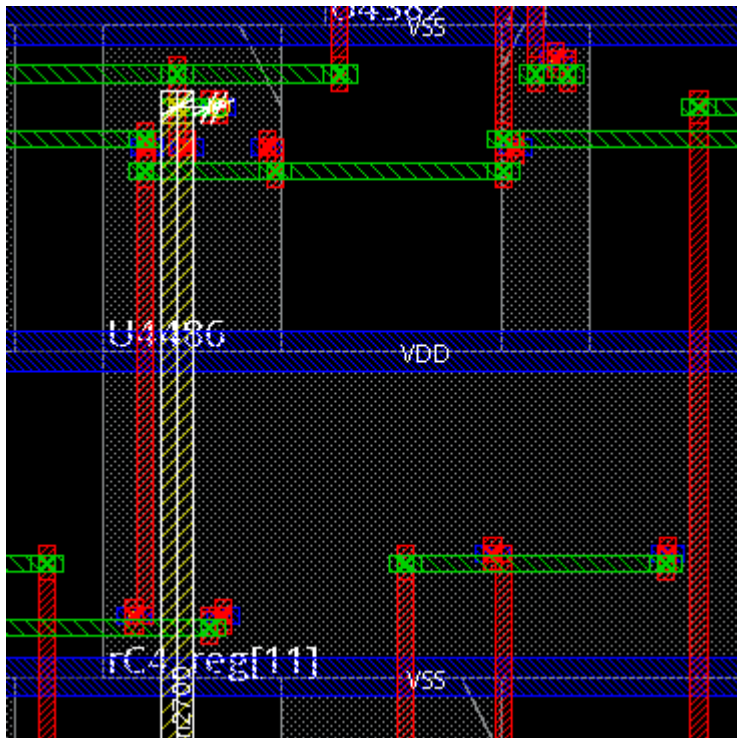
3. Placement

- Let's zoom in.



3. Placement

- Click a wire and press 'q'. You will see a property window.



Attribute Editor

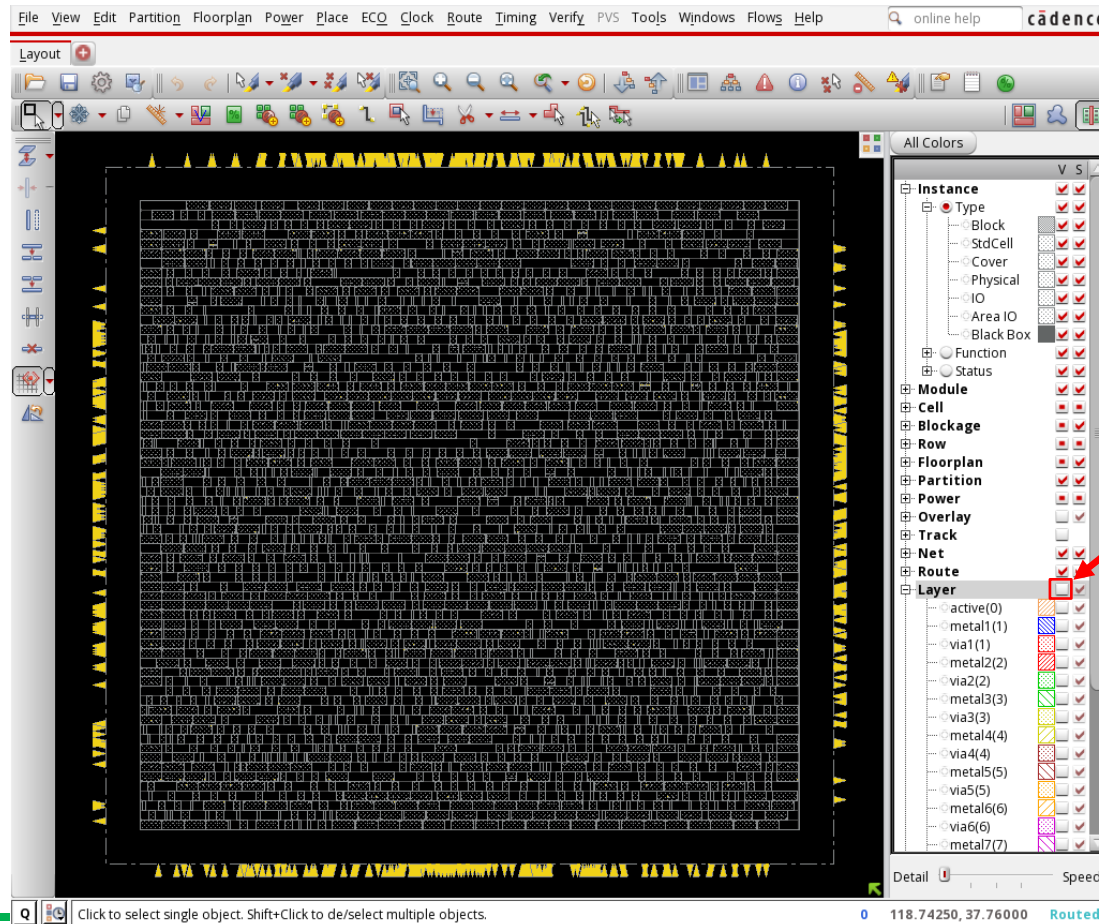
Object Type: Regular Wire

Name	Value	Type
Net Name	n2700	String
Wire Direction	dbcWireN ▾	Enumerate
Bounding Box	{45.945 31.22} {46.085 56.56}	Box
Routing Layer	metal4(4) ▾	Layer
Wire Status	Unknown ▾	Enumerate
Rule	default ▾	Enumerate
Mask	0	Integer

OK Apply Add Prop Delete Prop Close Help

Visibility

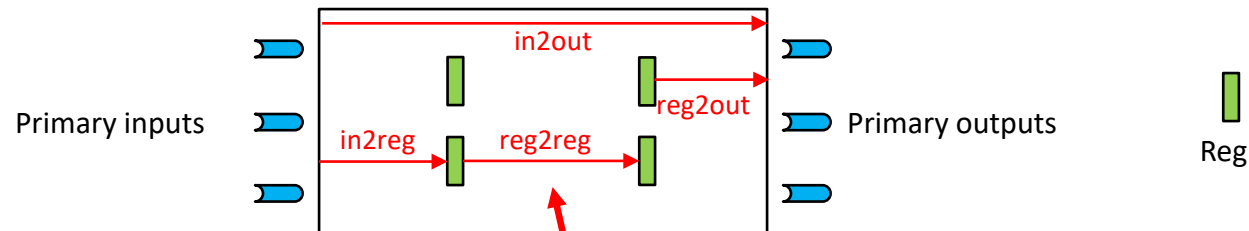
- Let's see the placement result only.
- Turn off the following check-box to turn off the visibility of the wires.



Static Timing Analysis

- Run the following command to turn off SI-awareness.
 - `innovus #> setDelayCalMode -siAware false`
- Then, run the following command to analyze setup time.
 - `innovus #> timeDesign -preCTS`
 - preCTS means “before Clock-Tree-Synthesis (CTS)”. A clock tree is designed after placement.
- It will show the following summary:

Static Timing Analysis



timeDesign Summary			
Setup views included: VView1			
Setup mode	all	reg2reg	default
WNS (ns):	-3.012	-1.624	-3.012
TNS (ns):	-1249.8	-508.627	-741.206
Violating Paths:	635	381	254
All Paths:	768	512	256

DRVs	Real		Total
	Nr nets(terms)	Worst Vio	Nr nets(terms)
max_cap	10 (10)	-0.234	10 (10)
max_tran	10 (1226)	-1.062	10 (1226)
max_fanout	0 (0)	0	0 (0)
max_length	0 (0)	0	0 (0)

Density: 57.409%
Routing Overflow: 0.00% H and 0.00% V

Setup time analysis

Negative WNS

Design Rule
Violations

Layout density

Static Timing Analysis

- Run the following command to check the longest path.
 - innovus #> report_timing
 - The clock frequency is 1GHz.

```
Path 1: VIOLATED Setup Check with Pin rc2_reg[62]/CK
Endpoint: rc2_reg[62]/D (^) checked with leading edge of 'myCLK'
Beginpoint: mX2[1] (^) triggered by leading edge of '@'
Path Groups: {myCLK}
Analysis View: VView1
Other End Arrival Time 0.000
- Setup 0.048
+ Phase Shift 1.000
= Required Time 0.952
- Arrival Time 3.965
= Slack Time -3.012
```

Annotations:

- F/F setup time (48ps) points to Setup (0.048)
- Clock period points to Phase Shift (1.000)
- RT ($T_{CLK} - T_s$) points to Required Time (0.952)
- AT points to Arrival Time (3.965)
- Slack (=RT - AT) points to Slack Time (-3.012)

Instance	Arc	Cell	Delay	Arrival Time	Required Time
	mX2[1] ^			0.000	-3.012

4. Pre-CTS Optimization

- Now, since the design violates the timing constraints, let's optimize it. (Notice that we can still try to optimize it to reduce power even if it satisfies the timing constraints.)
- Run the following command to optimize the design before CTS.
 - `innovus #> optDesign -preCTS`
- (This will take some time, up to several minutes depending on the machine you are working in).
- After Pre-CTS optimization is done, you will see the following result:

4. Pre-CTS Optimization

- Pre-CTS optimization

Positive WNS!!!

optDesign Final Summary

Setup views included:
VView1

Setup mode	all	reg2reg	default
WNS (ns):	0.005	0.010	0.005
TNS (ns):	0.000	0.000	0.000
Violating Paths:	0	0	0
All Paths:	768	512	256

DRVs	Real		Total
	Nr nets(terms)	Worst Vio	Nr nets(terms)
max_cap	0 (0)	0.000	0 (0)
max_tran	0 (0)	0.000	0 (0)
max_fanout	0 (0)	0	0 (0)
max_length	0 (0)	0	0 (0)

Density: 60.796%

The density increased from 57% to 61%.

Routing Overflow: 0.00% H and 0.00% V

The density increased from 57% to 61%.

4. Pre-CTS Optimization

- saveDesign test_04_prectsopt.enc

5. Clock Tree Synthesis (CTS)

- Run the following command to run CTS.
 - innovus #> create_ccopt_clock_tree_spec
 - innovus #> get_ccopt_clock_trees *
 - myCLK (You will see this.)
 - innovus #> set_ccopt_property target_max_trans 0.05
 - Max. transition time at a clock pin is 50ps.
 - innovus #> set_ccopt_property target_skew 0.02
 - Clock skew is 20ps.
 - innovus #> ccopt_design

5. Clock Tree Synthesis (CTS)

- saveDesign test_05_cts.enc

Timing Analysis

- Run the following command to check timing.
 - timeDesign -postCTS

optDesign Final Summary				
Setup views included: VView1				
Setup mode	all	reg2reg	default	
WNS (ns):	0.005	0.010	0.005	
TNS (ns):	0.000	0.000	0.000	
Violating Paths:	0	0	0	
All Paths:	768	512	256	
DRVs	Real		Total	
	Nr nets(terms)	Worst Vio	Nr nets(terms)	
max_cap	0 (0)	0.000	0 (0)	
max_tran	0 (0)	0.000	0 (0)	
max_fanout	0 (0)	0	0 (0)	
max_length	0 (0)	0	0 (0)	
Density: 60.796%				
Routing Overflow: 0.00% H and 0.00% V				

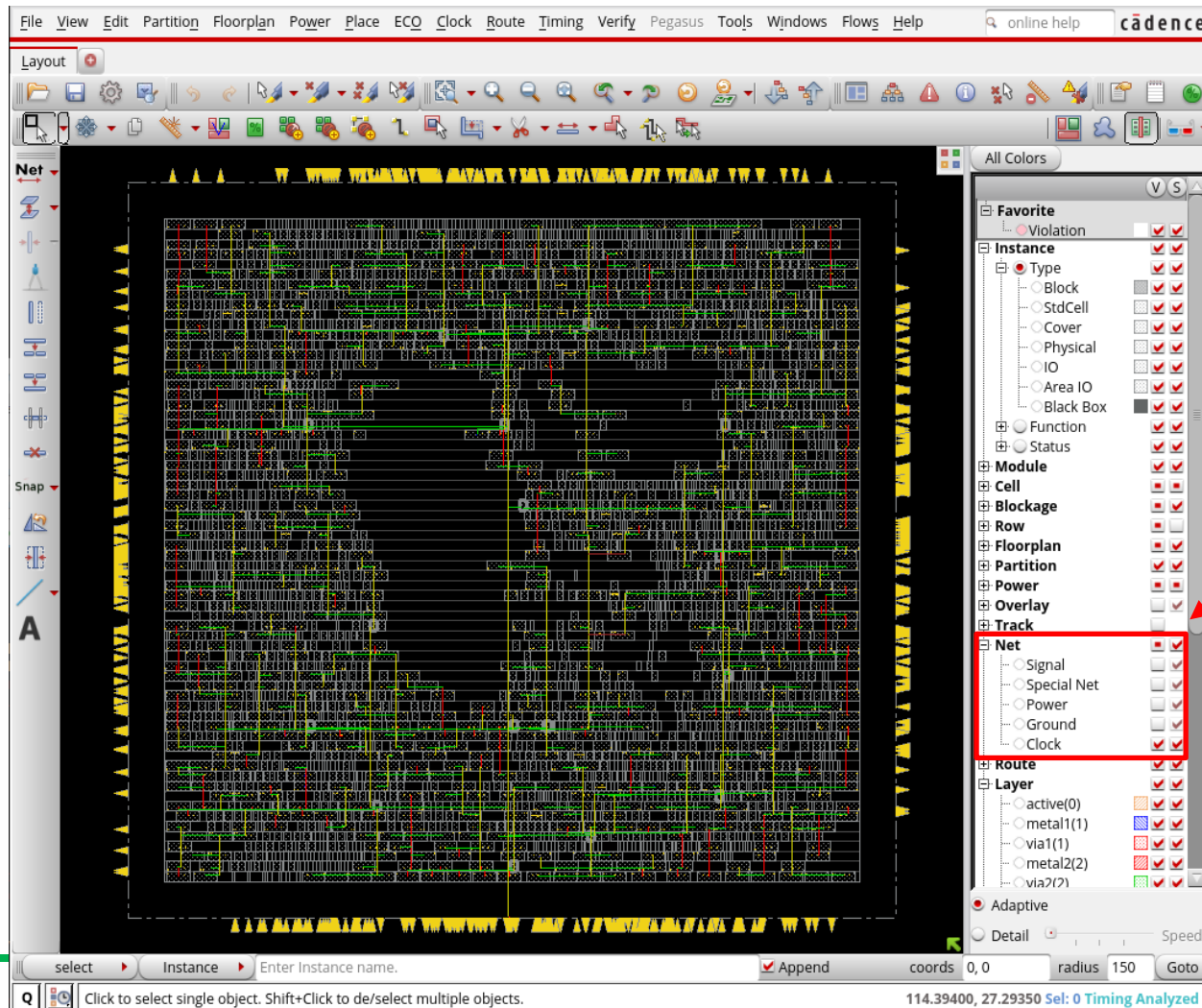
Before CTS

timeDesign Summary				
Setup views included: VView1				
Setup mode	all	reg2reg	default	
WNS (ns):	0.003	0.004	0.003	
TNS (ns):	0.000	0.000	0.000	
Violating Paths:	0	0	0	
All Paths:	768	512	256	
DRVs	Real		Total	
	Nr nets(terms)	Worst Vio	Nr nets(terms)	
max_cap	0 (0)	0.000	0 (0)	
max_tran	0 (0)	0.000	0 (0)	
max_fanout	0 (0)	0	0 (0)	
max_length	0 (0)	0	0 (0)	
Density: 61.072%				
Routing Overflow: 0.00% H and 0.00% V				

After CTS

5. Clock Tree Synthesis (CTS)

- How to show the clock tree



6. Post-CTS Optimization

- Although we already satisfied the timing without any further optimization after CTS, we will run post-CTS optimization.
 - innovus #> optDesign -postCTS

```
-----
optDesign Final Summary
-----

Setup views included:
VView1

+-----+-----+-----+
| Setup mode | all | reg2reg | default |
+-----+-----+-----+
| WNS (ns): | 0.004 | 0.004 | 0.004 |
| TNS (ns): | 0.000 | 0.000 | 0.000 |
| Violating Paths: | 0 | 0 | 0 |
| All Paths: | 768 | 512 | 256 |
+-----+-----+-----+

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

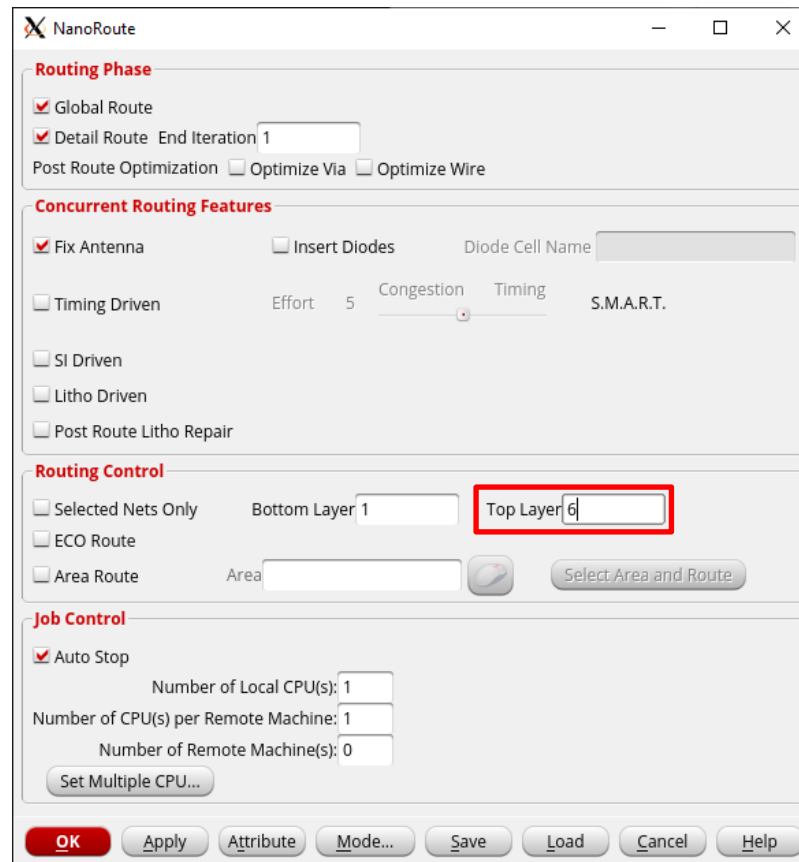
Density: 61.070%
Routing Overflow: 0.00% H and 0.00% V
```

6. Post-CTS Optimization

- saveDesign test_06_postctsopt.enc
- So far, we have done
 - Placement
 - CTS
- Now we will route the nets.

7. Routing

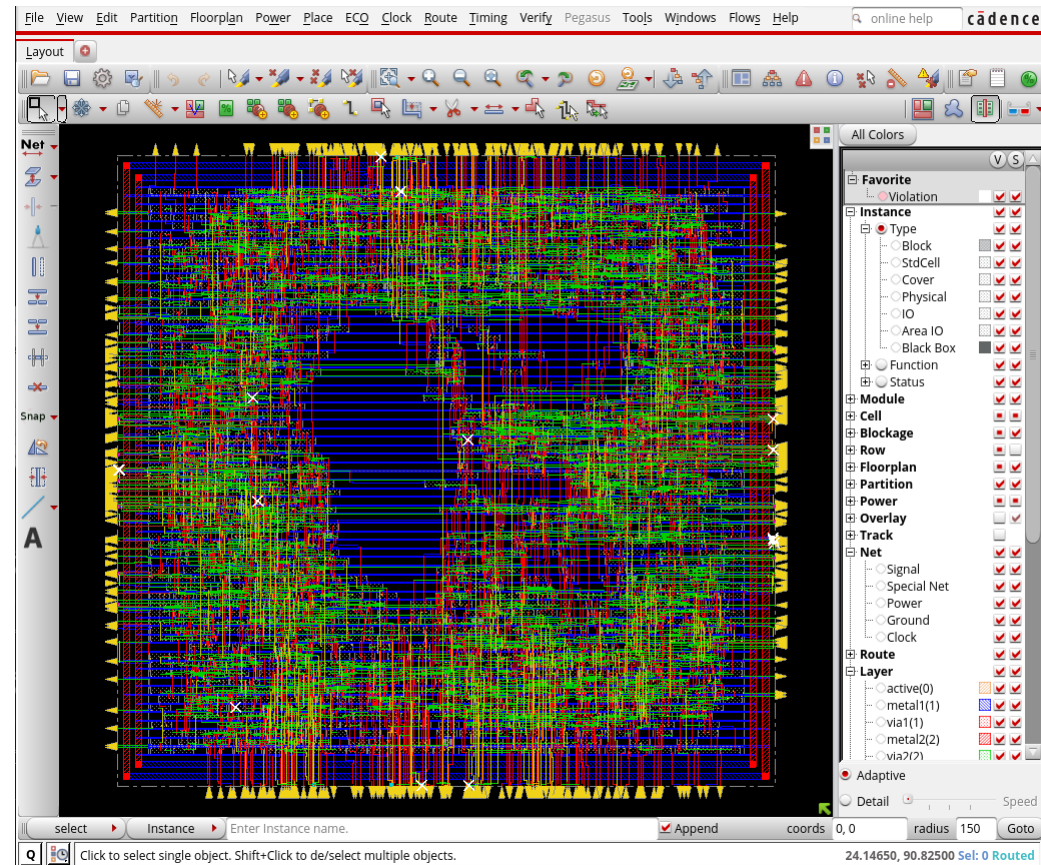
- Click “Route” → “NanoRoute” → “Route...”.
- Make sure that the top layer is “6”. If not, set it to 6.
- Click OK.



7. Routing

- Routing result.
- See the log.
 - WL: 35,172um

```
#Post Route wire spread is done.  
#Total number of nets with non-default rule or having  
#Total wire length = 31549 um.  
#Total half perimeter of net bounding box = 28181 um.  
#Total wire length on LAYER metal1 = 1299 um.  
#Total wire length on LAYER metal2 = 13043 um.  
#Total wire length on LAYER metal3 = 12676 um.  
#Total wire length on LAYER metal4 = 3314 um.  
#Total wire length on LAYER metal5 = 910 um.  
#Total wire length on LAYER metal6 = 307 um.  
#Total wire length on LAYER metal7 = 0 um.  
#Total wire length on LAYER metal8 = 0 um.  
#Total wire length on LAYER metal9 = 0 um.  
#Total wire length on LAYER metal10 = 0 um.  
#Total number of vias = 19082  
#Up-Via Summary (total 19082):
```



Timing Analysis

- Run the following command to check timing.
 - timeDesign -postRoute

timeDesign Summary			
Setup views included: VView1			
Setup mode	all	reg2reg	default
WNS (ns):	0.005	0.021	0.005
TNS (ns):	0.000	0.000	0.000
Violating Paths:	0	0	0
All Paths:	768	512	256
DRVs	Real		Total
	Nr nets(terms)	Worst Vio	Nr nets(terms)
max_cap	0 (0)	0.000	0 (0)
max_tran	0 (0)	0.000	0 (0)
max_fanout	0 (0)	0	0 (0)
max_length	0 (0)	0	0 (0)
Density: 61.070%			

8. Post-Route Optimization

- Although we've already satisfied the timing without any further optimization after routing, we will run post-routing optimization.
- Well, let's get PPA before that.
 - innovus #> report_area

```
innovus 18> report_area
```

Hinst Name	Module Name	Inst Count	Total Area
VQS64_4		2663	5467.896

- innovus #> report_power

```
*      Power Units = 1mW
*
*      Time Units = 1e-09 secs
*
Total Power
```

Total Internal Power:	6.30258075	65.7817%
Total Switching Power:	3.16562666	33.0405%
Total Leakage Power:	0.11284297	1.1778%
Total Power:	9.58105038	

- Now, let's optimize the design.
 - innovus #> optDesign -postRoute

8. Post-Route Optimization

timeDesign Summary			
Setup views included: VView1			
Setup mode	all	reg2reg	default
WNS (ns):	0.005	0.021	0.005
TNS (ns):	0.000	0.000	0.000
Violating Paths:	0	0	0
All Paths:	768	512	256
DRVs	Real		Total
	Nr nets(terms)	Worst Vio	Nr nets(terms)
max_cap	0 (0)	0.000	0 (0)
max_tran	0 (0)	0.000	0 (0)
max_fanout	0 (0)	0	0 (0)
max_length	0 (0)	0	0 (0)
Density: 61.070%			

Before postRoute opt.

optDesign Final Non-SI Timing Summary			
Setup views included: VView1			
Setup mode	all	reg2reg	default
WNS (ns):	0.005	0.022	0.005
TNS (ns):	0.000	0.000	0.000
Violating Paths:	0	0	0
All Paths:	768	512	256
DRVs	Real		Total
	Nr nets(terms)	Worst Vio	Nr nets(terms)
max_cap	0 (0)	0.000	0 (0)
max_tran	0 (0)	0.000	0 (0)
max_fanout	0 (0)	0	0 (0)
max_length	0 (0)	0	0 (0)
Density: 61.070%			

After postRoute opt.

8. Post-Route Optimization

- Area and power (before)

```
innovus 18> report_area
  Hinst Name      Module Name      Inst Count      Total Area
-----
VQS64_4          2663          5467.896

*      Power Units = 1mW
*
*      Time Units = 1e-09 secs
*
Total Power
-----
Total Internal Power:      6.30258075      65.7817%
Total Switching Power:     3.16562666      33.0405%
Total Leakage Power:       0.11284297      1.1778%
Total Power:               9.58105038
```

- Area and power (after)

```
innovus 22> report_area
  Hinst Name      Module Name      Inst Count      Total Area
-----
VQS64_4          2663          5467.896

Total Power
-----
Total Internal Power:      6.30263487      65.7820%
Total Switching Power:     3.16561527      33.0402%
Total Leakage Power:       0.11284297      1.1778%
Total Power:               9.58109311
```

8. Post-Route Optimization

- saveDesign test_08_postrouteopt.enc
- Done

9. Verification



9. Verification

- In the main menu, Verify → Very DRC. Click OK. (DRC: Design Rule Check)

```
VERIFY DRC ..... Starting Verification
VERIFY DRC ..... Initializing
VERIFY DRC ..... Deleting Existing Violations
VERIFY DRC ..... Creating Sub-Areas
VERIFY DRC ..... Using new threading
VERIFY DRC ..... Sub-Area: {0.000 0.000 54.000 51.840} 1 of 4
VERIFY DRC ..... Sub-Area : 1 complete 4 Viols.
VERIFY DRC ..... Sub-Area: {54.000 0.000 107.100 51.840} 2 of 4
VERIFY DRC ..... Sub-Area : 2 complete 2 Viols.
VERIFY DRC ..... Sub-Area: {0.000 51.840 54.000 102.480} 3 of 4
VERIFY DRC ..... Sub-Area : 3 complete 1 Viols.
VERIFY DRC ..... Sub-Area: {54.000 51.840 107.100 102.480} 4 of 4
VERIFY DRC ..... Sub-Area : 4 complete 7 Viols.

Verification Complete : 14 Viols.

Violation Summary By Layer and Type:
```

	Short	MetSpc	CutSpc	Totals
metal2	1	0	0	1
metal6	4	2	0	6
via6	0	0	3	3
metal7	3	1	0	4
Totals	8	3	3	14

- There are some violations, but we will ignore them.
- (In the real world, we run DRC with the most accurate design rule files.)

9. Verification

- In the main menu, Verify → Very Connectivity. Click OK.

```
innovus 28> innovus 28> VERIFY_CONNECTIVITY use new engine.  
  
***** Start: VERIFY CONNECTIVITY *****  
Start Time: Sun Mar 29 15:08:03 2020  
  
Design Name: VQS64_4  
Database Units: 2000  
Design Boundary: (0.0000, 0.0000) (107.1000, 102.4800)  
Error Limit = 1000; Warning Limit = 50  
Check all nets  
  
Begin Summary  
  Found no problems or warnings.  
End Summary  
  
End Time: Sun Mar 29 15:08:03 2020  
Time Elapsed: 0:00:00.0  
  
***** End: VERIFY CONNECTIVITY *****  
Verification Complete : 0 Viols. 0 Wrngs.
```