
EE434
ASIC & Digital Systems

Automatic Layout Generation
(Encounter)

Spring 2019
Dae Hyun Kim
daehyun@eecs.wsu.edu

Preparation for Lab3

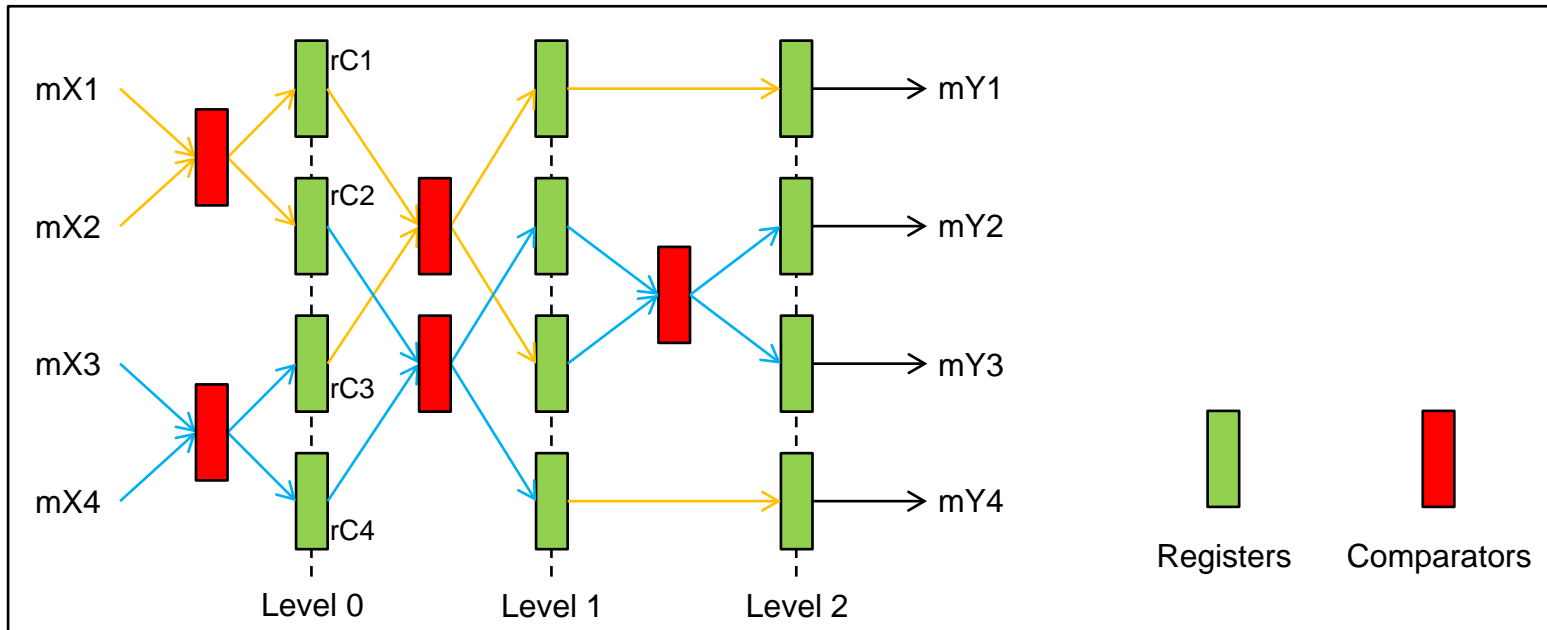
- Download the following file into your working directory.
 - `wget http://eecs.wsu.edu/~ee434/Labs/lab3.tar.gz`
- Unzip it.
 - `tar xvfz lab3.tar.gz`

What We Are Going To Do

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
9. Fill insertion

1. Chip Outlining

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

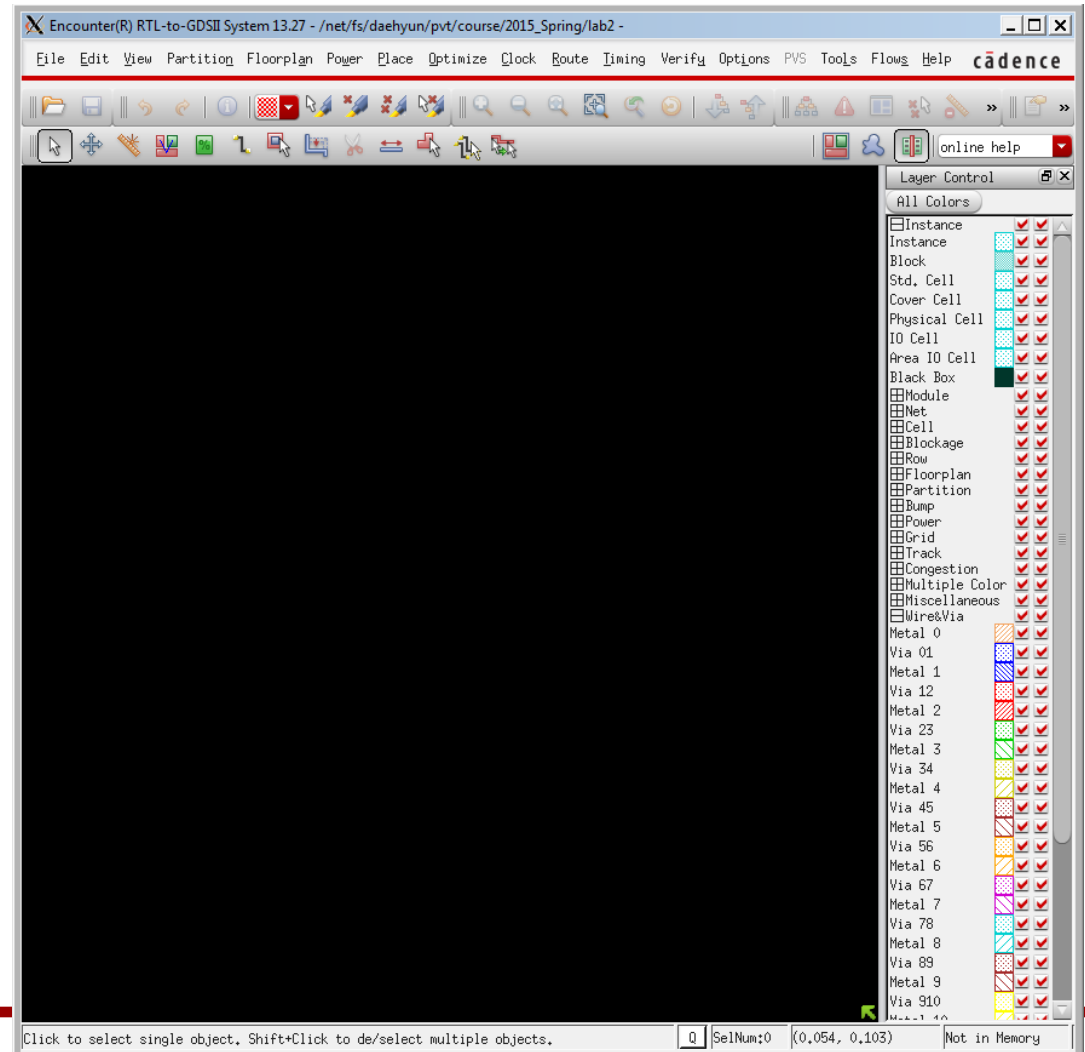


1. Chip Outlining

- VQS64_4_fm.globals
 - init_pwr_net: Power nets.
 - init_gnd_net: Ground nets.
 - init_lef_file: Physical library files.
 - init_mmmc_file: Analysis view files.
 - mmmc: Multi-mode multi-corner
 - init_verilog: Verilog netlists.
- VQS64_4_fm.view
 - create_rc_corner: Capacitance table + RC analysis corner
 - create_library_set: Library files
 - create_constraint_mode: Constraint files
 - create_delay_corner: Library + RC corner
 - create_analysis_view: Analysis view
 - set_analysis_view: Setup and hold analysis view

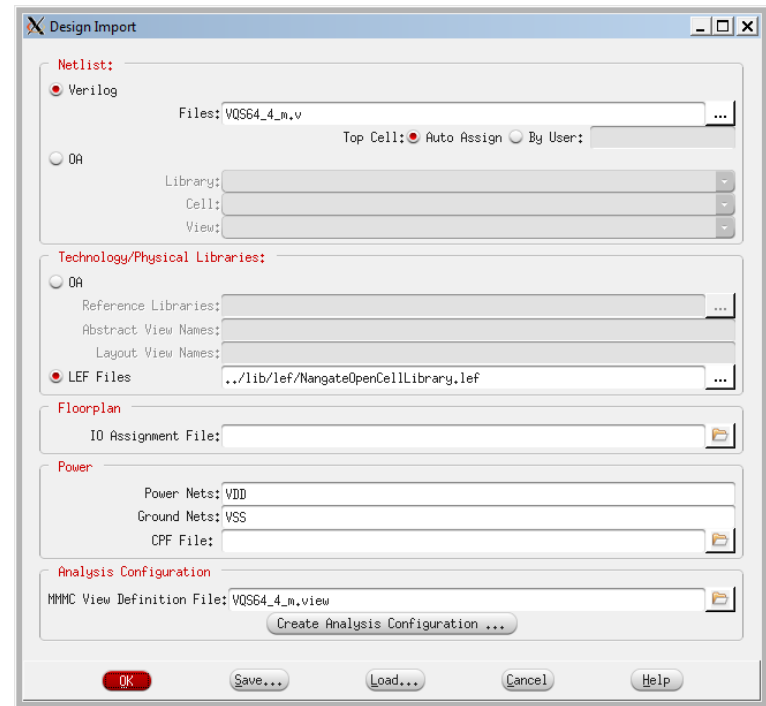
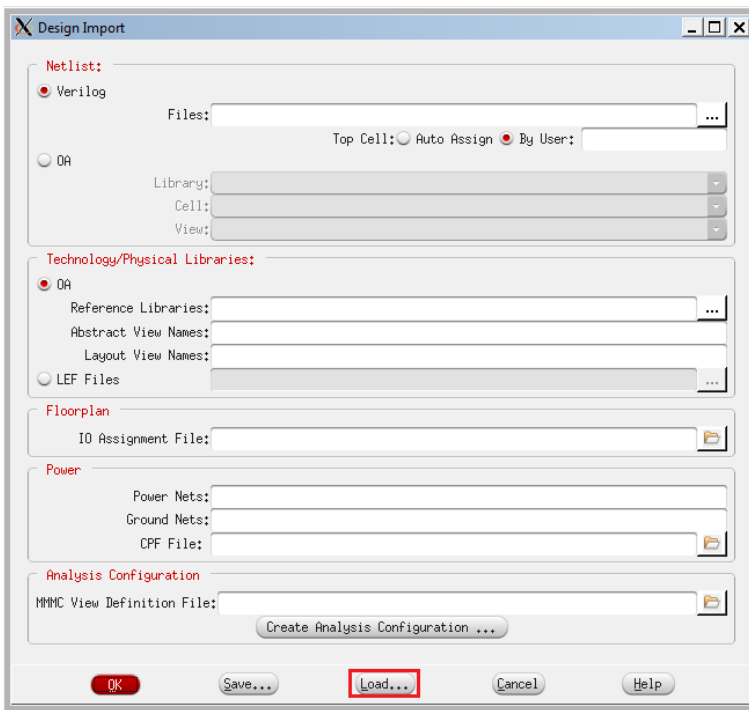
1. Chip Outlining

- Source “edi.sh”.
 - % source edi.sh
- Run Encounter.
 - % encounter



1. Chip Outlining

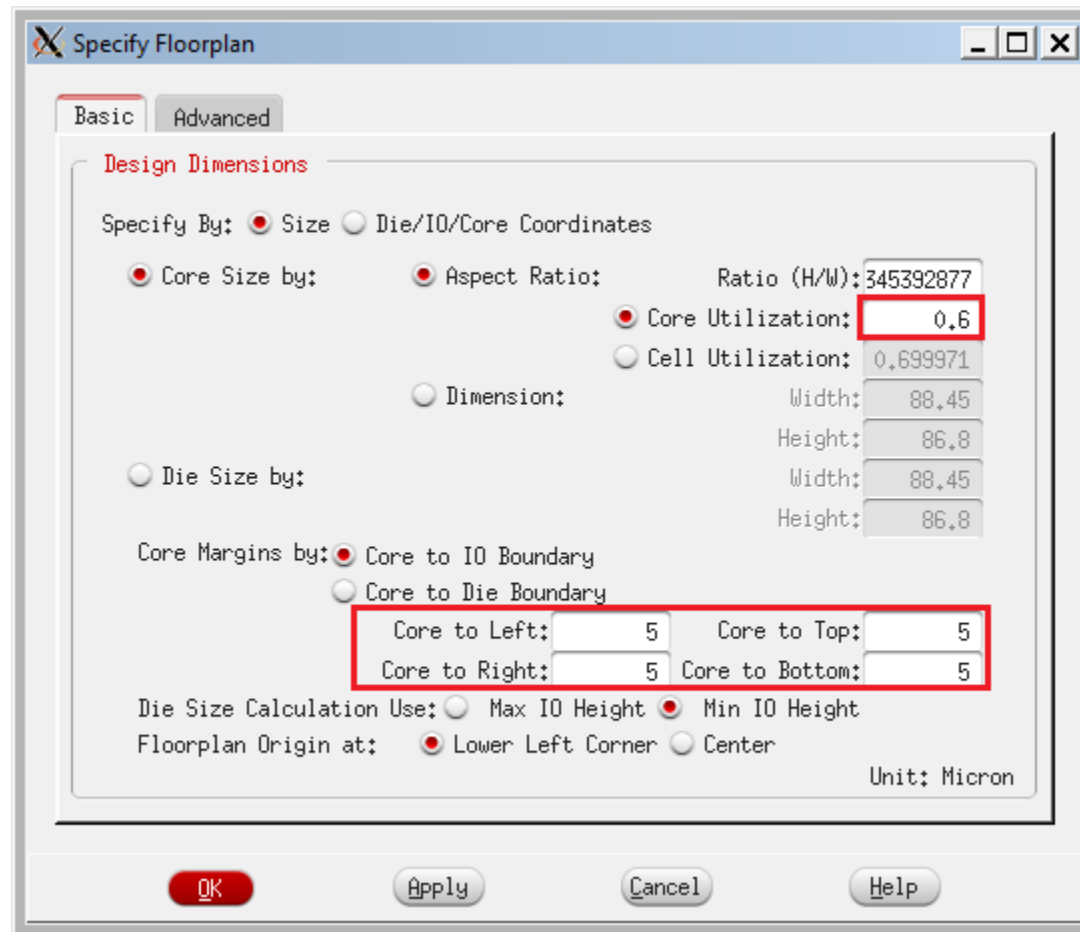
- Click “File” → “Import Design...”.
- In the “Design Import” window, click “Load...” and choose “VQS64_4_m.globals”. This will automatically fill up the settings. Then, click “OK”.



1. Chip Outlining

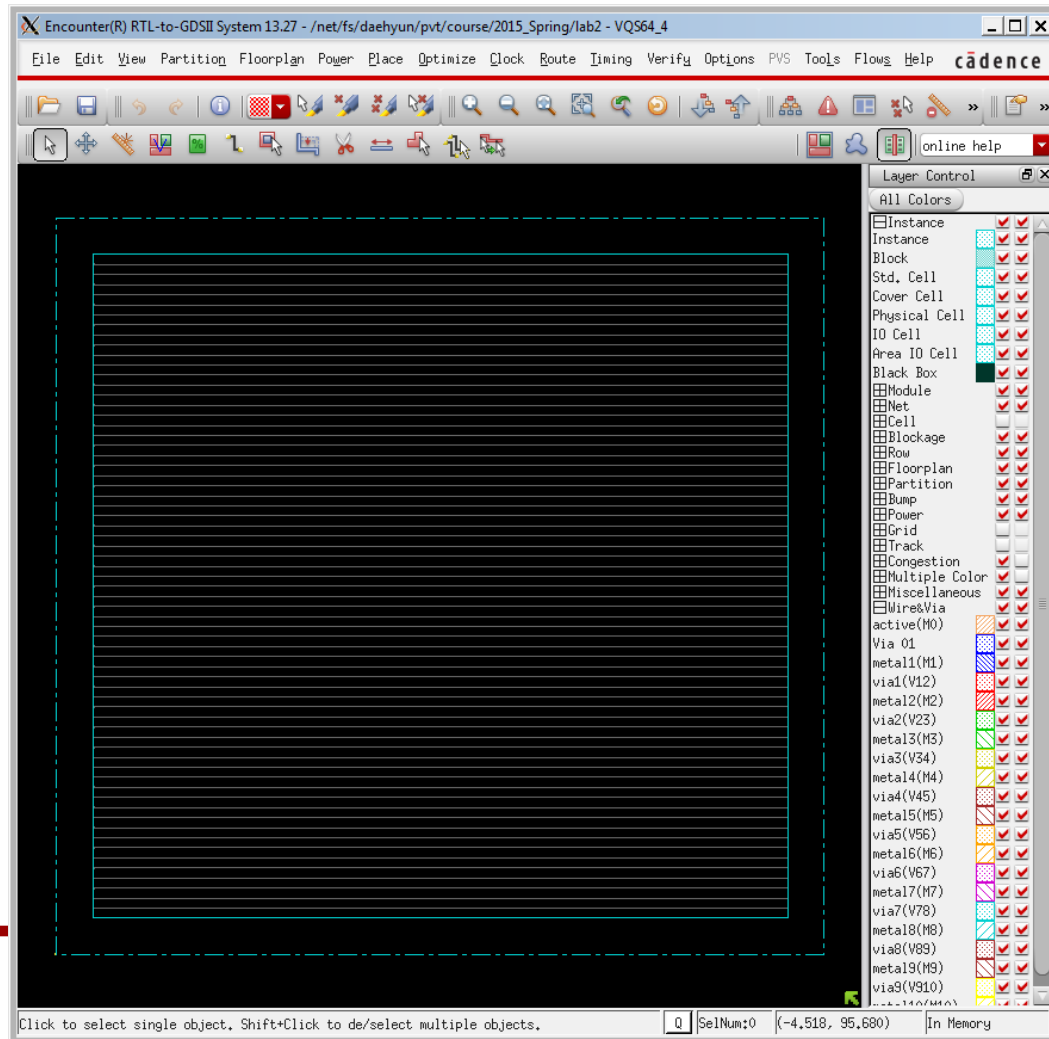
- See the terminal for Encounter messages. There might be some Error or Warning messages.
- In the Encounter main window, press “f” to see the outline of the layout.
- Encounter automatically computes and prepares the layout area.
- In the main window, click “Floorplan” → “Specify Floorplan...”.
- Set 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.

1. Chip Outlining



1. Chip Outlining

- Now, you will see the following window.

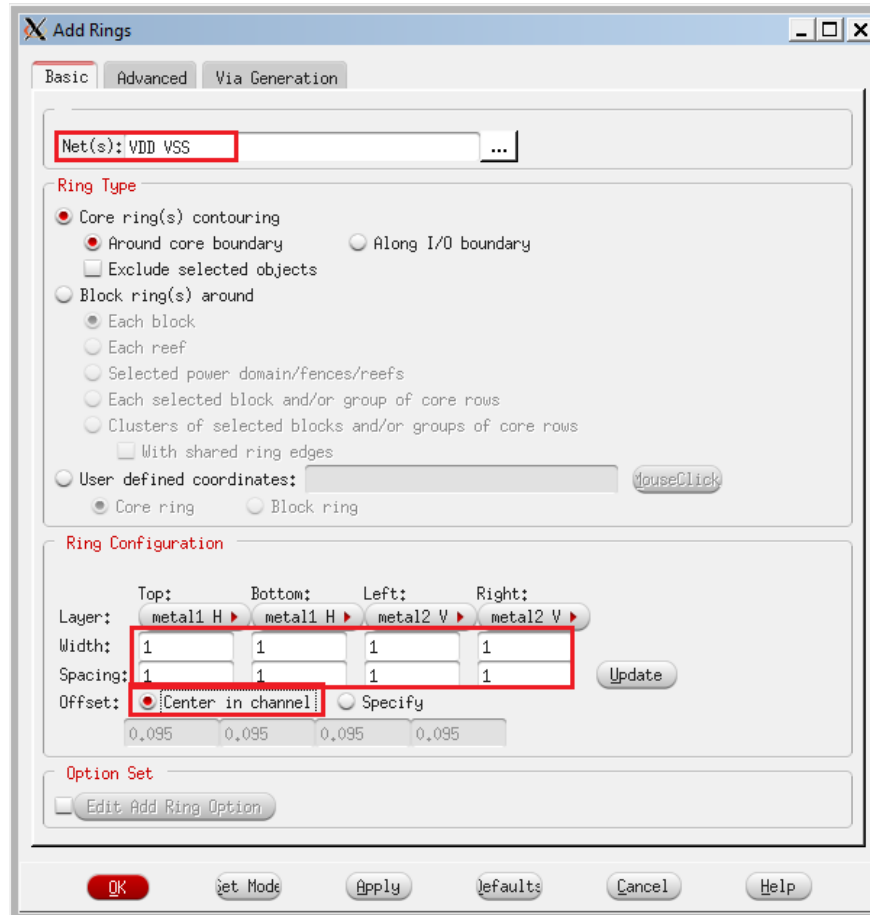


Save

- Let's save the current design.
- In the terminal, run the following command to save the current design into "test_01_floorplan.enc".
 encounter #> saveDesign test_01_floorplan.enc
- Later on, you can load the design as follows.
 - When you launch Encounter, add the following option to load the specified design.
 - `encounter -init test_01_floorplan.enc`
 - or, after you launch Encounter, run the following command.
 - `source test_01_floorplan.enc`

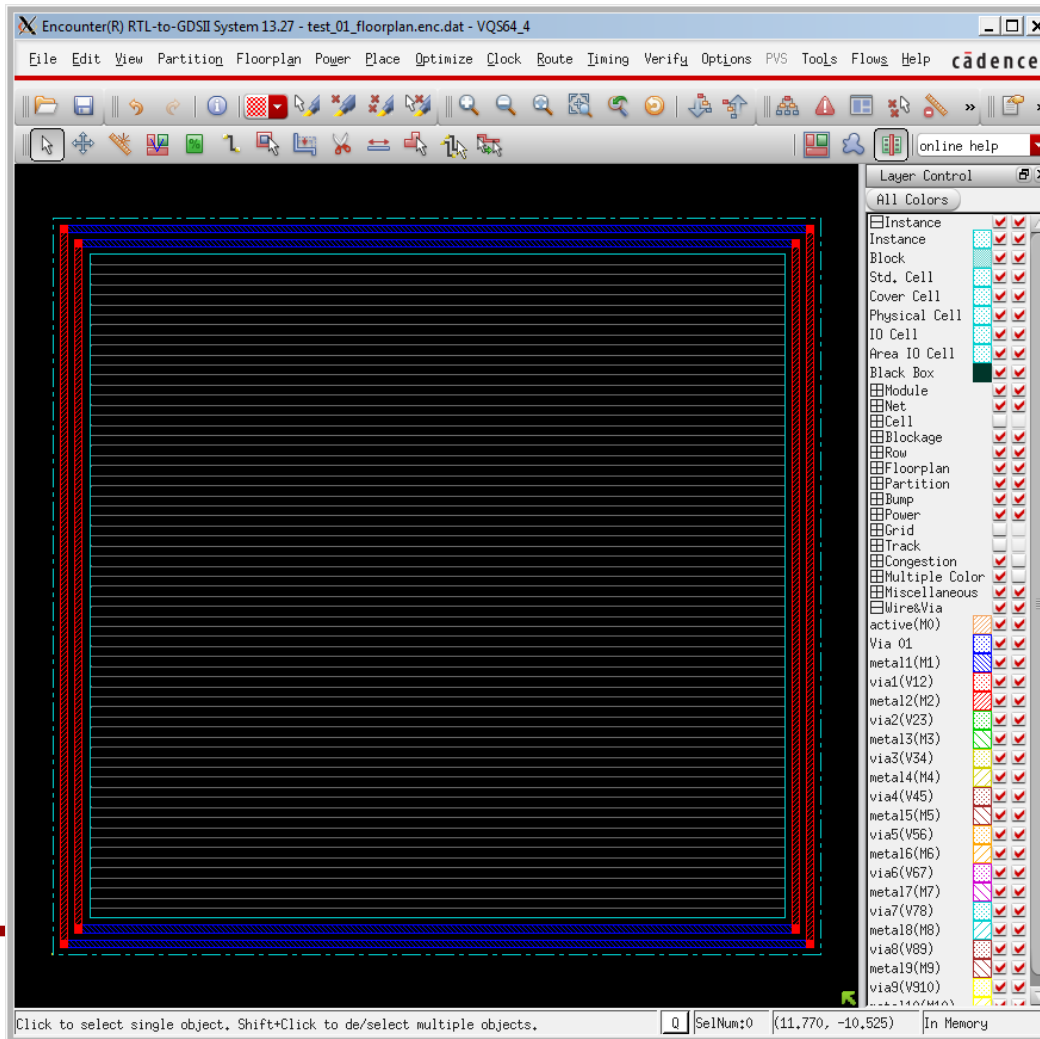
2. P/G Network Design

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



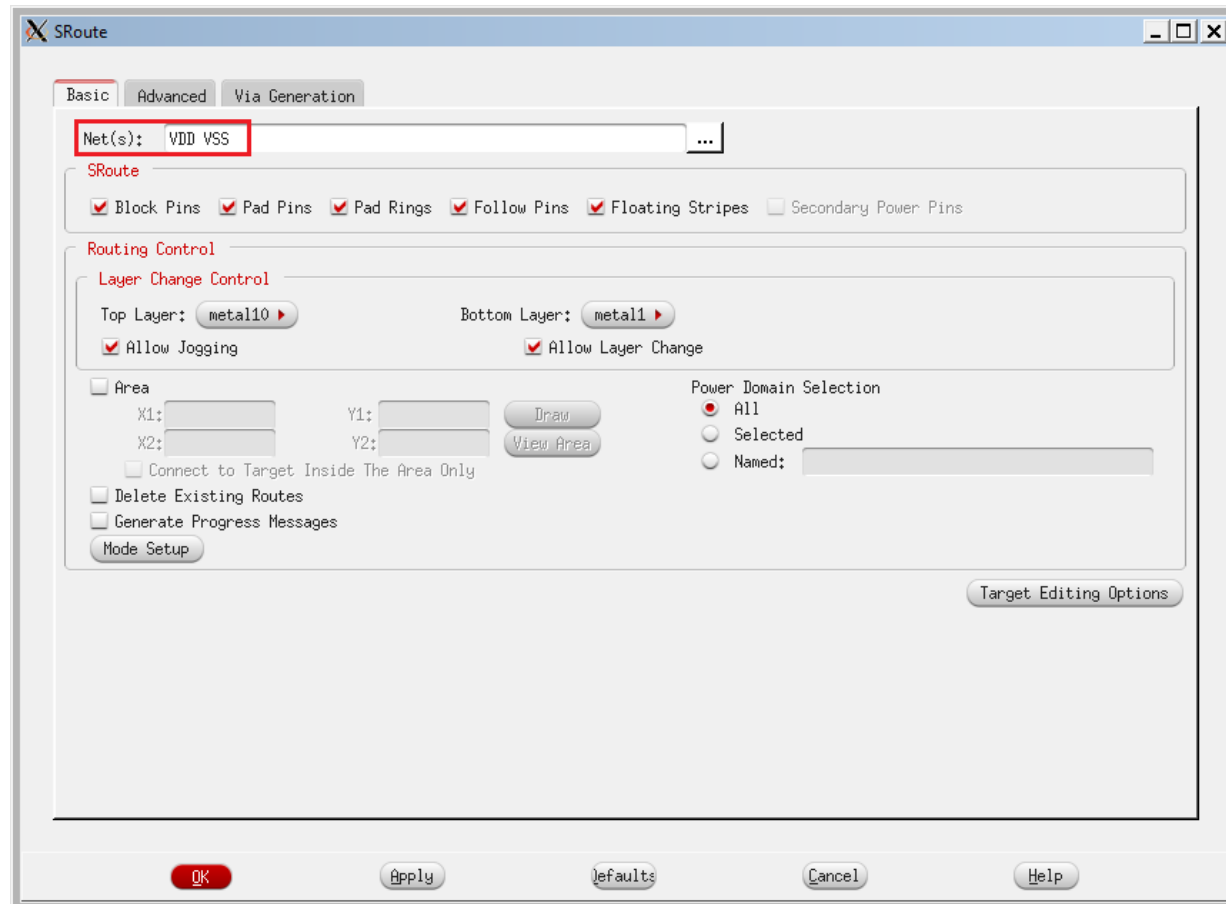
2. P/G Network Design

- Fill in the input boxes as shown in the previous page and click OK.



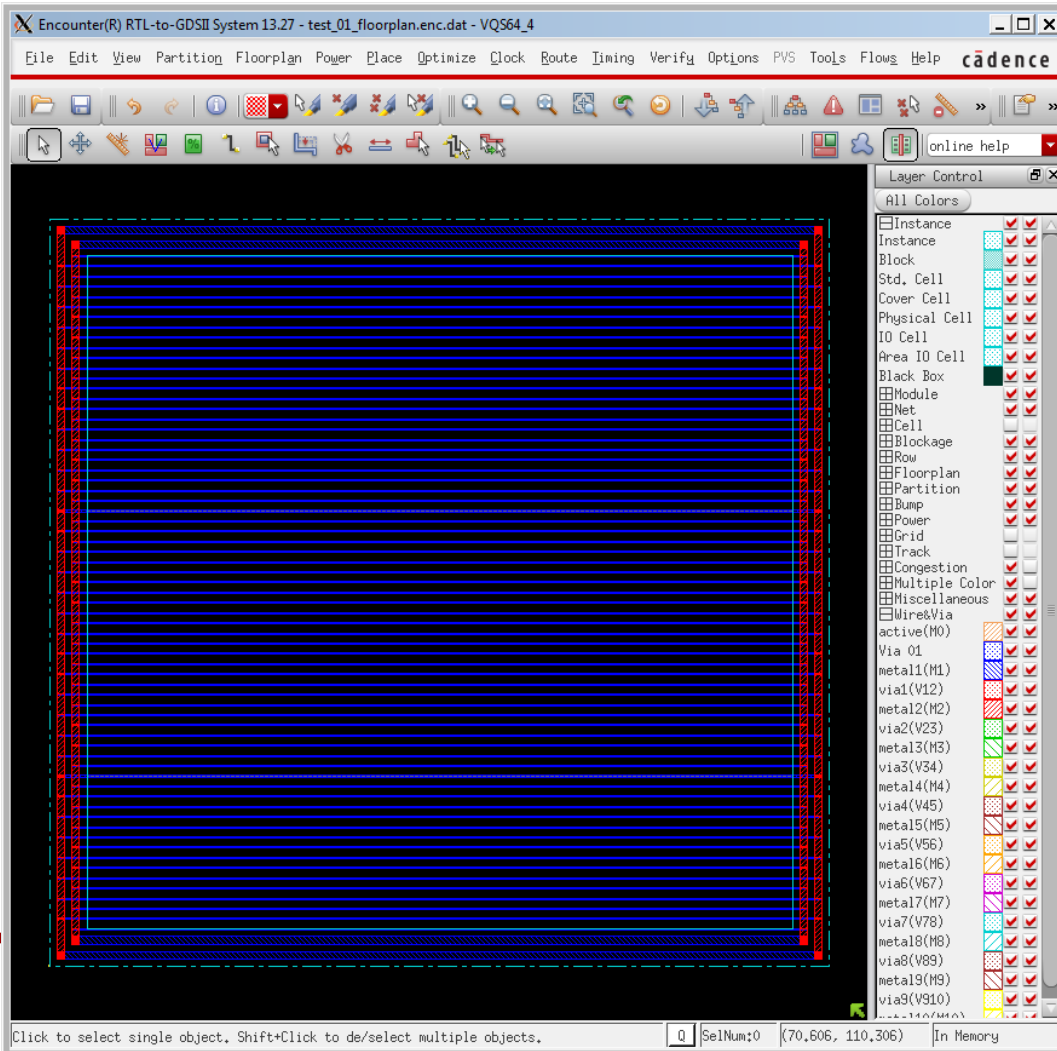
2. P/G Network Design

- Click “Route” → “Special Route...”.



2. P/G Network Design

- P/G network

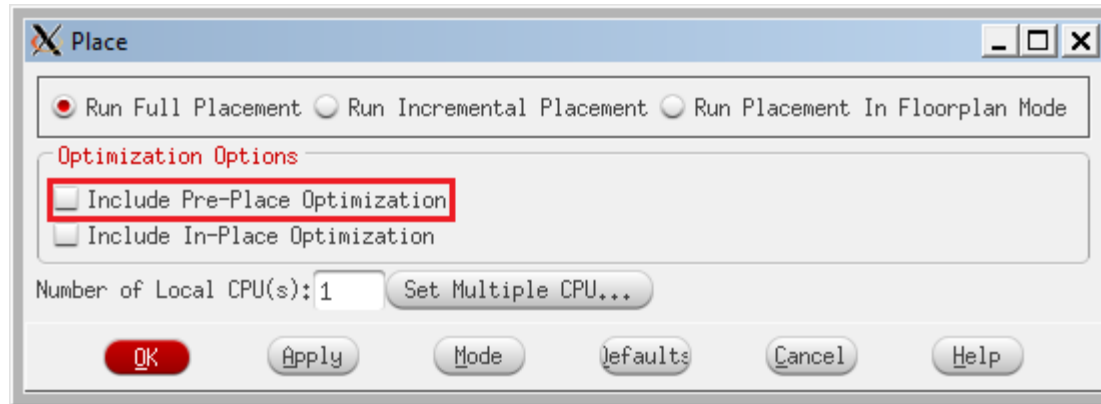


2. P/G Network Design

- saveDesign test_02_pg.enc

3. Placement

- Let's place the instances (cells).
- In the main window, click "Place" → "Place Standard Cell".
- In the following window, turn off "Include Pre-Place Optimization".

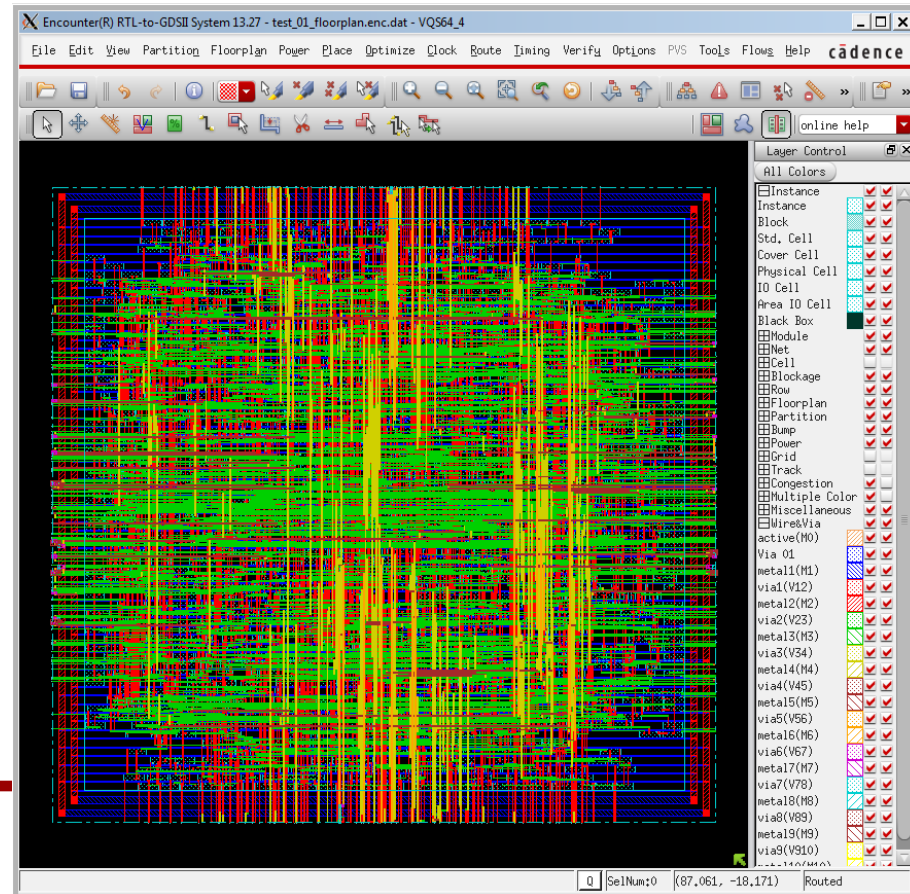


- Then, click "OK" to run Placement.

3. Placement

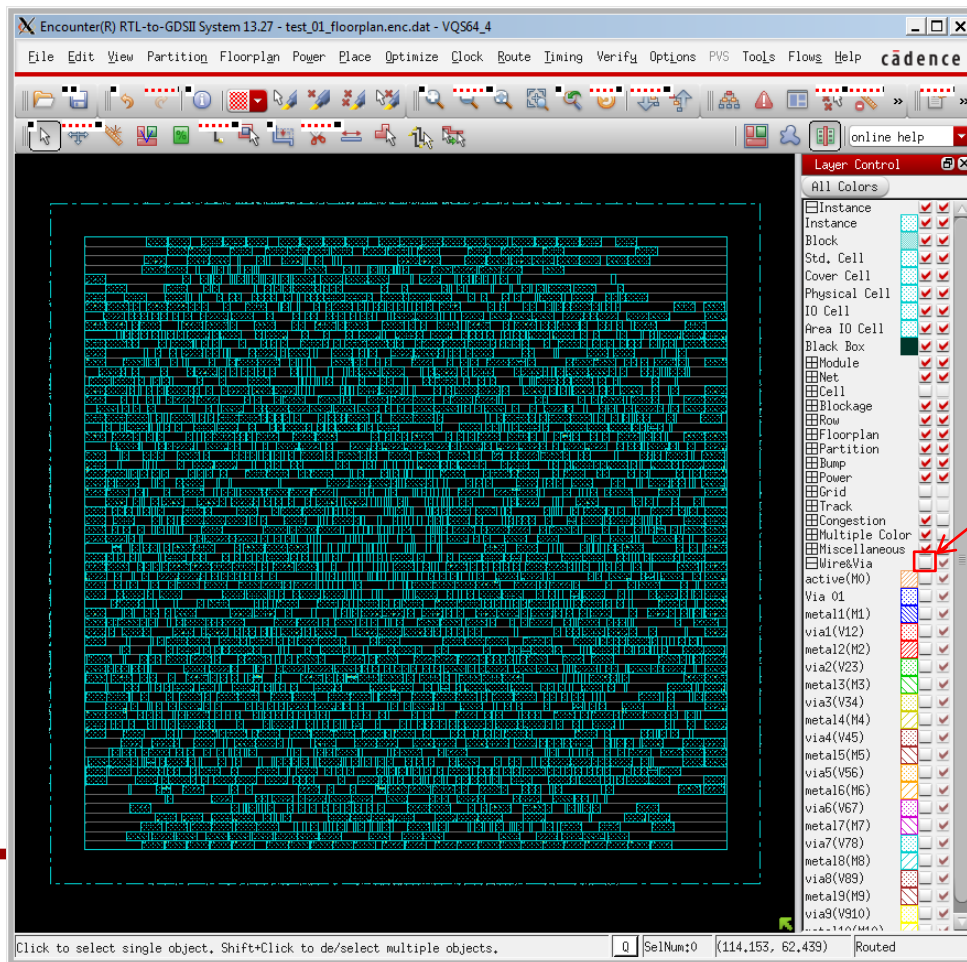
- It shows placement and trialRoute results.
- See the terminal. It shows some more information.
 - Total wire length: 46,920um
- Save it.
 - saveDesign test_03_pl.enc

```
Total length: 4.692e+04um, number of vias: 17032
M1(H) length: 1.187e+03um, number of vias: 9371
M2(V) length: 2.059e+04um, number of vias: 6881
M3(H) length: 1.976e+04um, number of vias: 506
M4(V) length: 3.390e+03um, number of vias: 167
M5(H) length: 1.261e+03um, number of vias: 68
M6(V) length: 7.066e+02um, number of vias: 25
M7(H) length: 9.350e+00um, number of vias: 8
M8(V) length: 6.480e+00um, number of vias: 6
M9(H) length: 4.140e+00um, number of vias: 0
M10(V) length: 0.000e+00um
```



Visibility

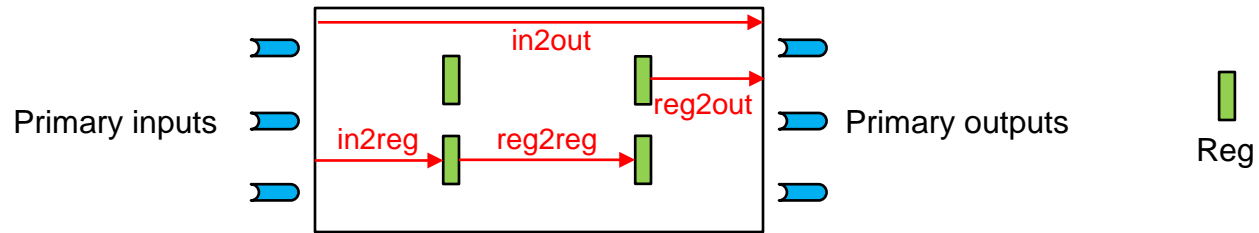
- Let's see the placement result only.
- Turn off the following check-box to turn off the visibility of the wires.



Timing Analysis

- Run the following command to turn off SI-awareness.
 - `encounter #> setDelayCalMode –siAware false`
- Then, run the following command to analyze setup time.
 - `encounter #> timeDesign –preCTS`
- It will show the following summary:

Timing Analysis



timeDesign Summary

Setup time analysis

Setup mode	all	reg2reg	in2reg	reg2out	in2out	clkgate
WNS (ns):	-1.654	-0.142	-1.654	N/A	N/A	N/A
TNS (ns):	-408.823	-11.658	-397.164	N/A	N/A	N/A
Violating Paths:	426	172	254	N/A	N/A	N/A
All Paths:	768	512	256	N/A	N/A	N/A

Design Rule
Violations

DRVs	Real		Total
	Nr nets(terms)	Worst Vio	Nr nets(terms)
max_cap	2 (2)	-0.011	2 (2)
max_tran	0 (0)	0.000	0 (0)
max_fanout	0 (0)	0	0 (0)
max_length	0 (0)	0	0 (0)

Timing Analysis

- Run the following command to check the longest path.
 - encounter #> report_timing

```
Path 1: VIOLATED Setup Check with Pin rC1_reg[52]/CK
Endpoint: rC1_reg[52]/D (v) checked with leading edge of 'myCLK'
Beginpoint: mX2[1] (^) triggered by leading edge of '@'
Analysis View: NG_view_typ
Other End Arrival Time      0.000
- Setup                     0.047
+ Phase Shift               1.000
= Required Time             0.953
- Arrival Time              2.606
= Slack Time                -1.654

Clock Rise Edge             0.000
+ Input Delay               0.000
= Beginpoint Arrival Time   0.000
```

Instance	Arc	Cell	Delay	Arrival Time	Required Time
	mX2[1] ^			0.000	-1.654
U4103	A ^ -> ZN v	INV_X1	0.010	0.011	-1.643
U3853	C2 v -> ZN ^	OAI211_X1	0.034	0.045	-1.609
U3852	A ^ -> ZN v	OAI221_X1	0.034	0.079	-1.575
U3861	A v -> ZN ^	OAI221_X1	0.026	0.105	-1.549
U3860	A ^ -> ZN v	OAI221_X1	0.041	0.146	-1.508
U3859	A v -> ZN ^	OAI221_X1	0.028	0.174	-1.480
U3858	A ^ -> ZN v	OAI221_X1	0.040	0.214	-1.440
U3869	A v -> ZN ^	OAI221_X1	0.027	0.241	-1.413
U3868	A ^ -> ZN v	OAI221_X1	0.040	0.281	-1.372
U3867	A v -> ZN ^	OAI221_X1	0.029	0.311	-1.343
U3866	A ^ -> ZN v	OAI221_X1	0.041	0.352	-1.302
U3877	A v -> ZN ^	OAI221_X1	0.028	0.380	-1.274
U3876	A ^ -> ZN v	OAI221_X1	0.041	0.421	-1.233
U3875	A v -> ZN ^	OAI221_X1	0.028	0.449	-1.205
U3874	A ^ -> ZN v	OAI221_X1	0.041	0.490	-1.164
U3885	A v -> ZN ^	OAI221_X1	0.027	0.517	-1.137
U3884	A ^ -> ZN v	OAI221_X1	0.040	0.556	-1.097
U3883	A v -> ZN ^	OAI221_X1	0.027	0.584	-1.070
U3882	A ^ -> ZN v	OAI221_X1	0.040	0.624	-1.030
U3893	A v -> ZN ^	OAI221_X1	0.027	0.651	-1.003

U3946	A ^ -> ZN v	OAI221_X1	0.039	1.707	0.053
U3957	A v -> ZN ^	OAI221_X1	0.027	1.734	0.080
U3956	A ^ -> ZN v	OAI221_X1	0.040	1.774	0.120
U3955	A v -> ZN ^	OAI221_X1	0.029	1.803	0.149
U3954	A ^ -> ZN v	OAI221_X1	0.041	1.843	0.190
U3965	A v -> ZN ^	OAI221_X1	0.028	1.872	0.218
U3964	A ^ -> ZN v	OAI221_X1	0.042	1.914	0.260
U3963	A v -> ZN ^	OAI221_X1	0.029	1.943	0.289
U3962	A ^ -> ZN v	OAI221_X1	0.042	1.985	0.331
U3973	A v -> ZN ^	OAI221_X1	0.027	2.011	0.358
U3972	A ^ -> ZN v	OAI221_X1	0.040	2.051	0.397
U3971	A v -> ZN ^	OAI221_X1	0.026	2.077	0.424
U3970	A ^ -> ZN v	OAI21_X1	0.024	2.102	0.448
U4233	A v -> ZN ^	OAI21_X1	0.021	2.123	0.469
U4232	A ^ -> ZN v	OAI221_X1	0.034	2.157	0.503
U4231	A v -> ZN ^	OAI21_X1	0.113	2.270	0.616
U3648	A ^ -> ZN v	INV_X1	0.025	2.295	0.642
U3330	A v -> Z v	BUF_X1	0.050	2.346	0.692
U3273	A v -> Z v	BUF_X1	0.055	2.401	0.747
U3255	A v -> ZN ^	INV_X1	0.157	2.557	0.903
U3513	B2 ^ -> ZN v	OAI22_X1	0.049	2.606	0.953
rC1_reg[52]	D v	DFF_X1	0.000	2.606	0.953

4. Pre-CTS Optimization

- Run the following command to optimize the design before CTS.
 - `encounter #> optDesign -preCTS`
- (This will take some time, up to 20~30 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

```
optDesign Final Summary
```

Setup mode	all	reg2reg	in2reg	reg2out	in2out	clkgate
WNS (ns):	0.066	0.074	0.066	N/A	N/A	N/A
TNS (ns):	0.000	0.000	0.000	N/A	N/A	N/A
Violating Paths:	0	0	0	N/A	N/A	N/A
All Paths:	768	512	256	N/A	N/A	N/A

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: 74.899%

Positive WNS

The density increased from 60% to 75%.

4. Pre-CTS Optimization

- saveDesign test_04_prechtsopt.enc

5. Clock Tree Synthesis (CTS)

- Open “VQS64_4_fm.ctstch” in a text editor and see the spec.
- Run the following command to run CTS.
 - `encounter #> clockDesign –specFile VQS64_4_fm.ctstch –outDir clk_report`

5. Clock Tree Synthesis (CTS)

- CTS

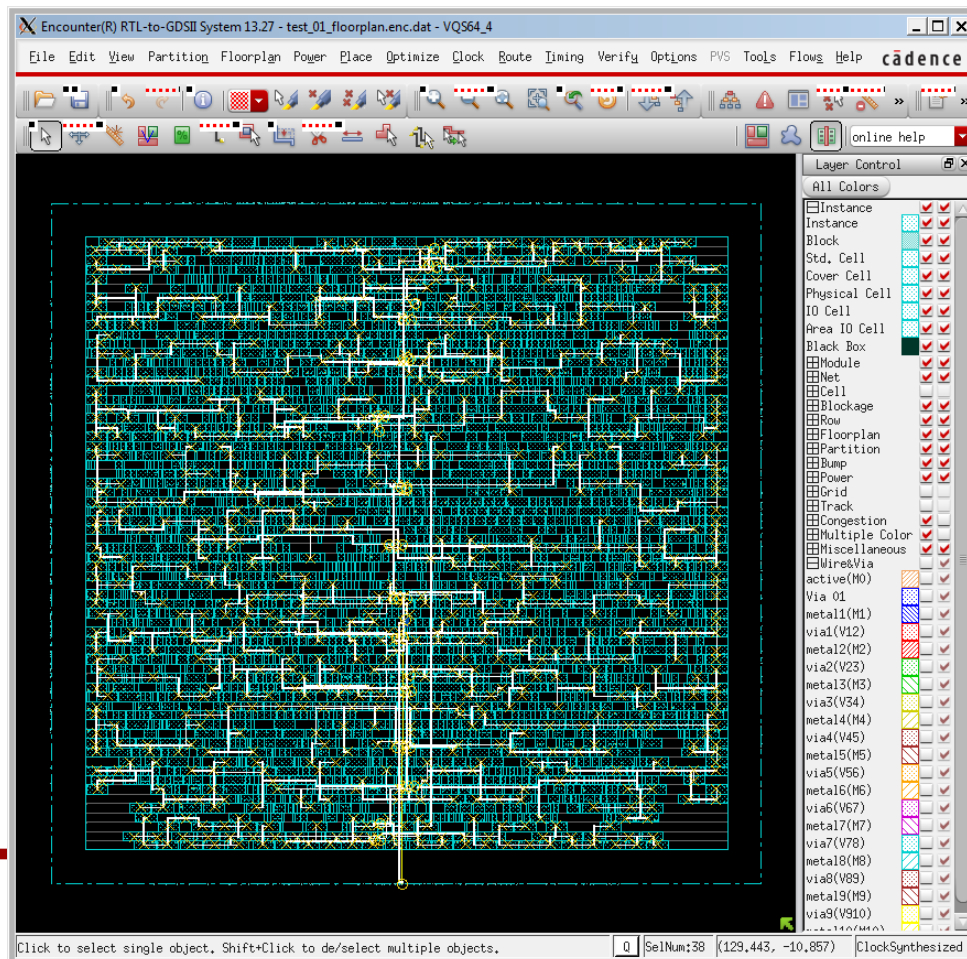
```
# Analysis View: NG_view_typ
***** Clock mCLK Post-CTS Timing Analysis *****
Nr. of Subtrees           : 1
Nr. of Sinks              : 768 ← # Sinks (64 F/Fs*12 groups = 768)
Nr. of Buffer             : 37 ← # buffers inserted
Nr. of Level (including gates) : 2 ← # levels
Root Rise Input Tran      : 100(ps)
Root Fall Input Tran     : 100(ps)
No Driving Cell Specified!
Max trig. edge delay at sink(R): rC2_reg[53]/CK 163.3(ps)
Min trig. edge delay at sink(R): mY2_reg[34]/CK 155.5(ps)

                (Actual)                (Required)
Rise Phase Delay       : 155.5~163.3(ps)    0~1000(ps)
Fall Phase Delay       : 168.2~175.6(ps)    0~1000(ps)
Trig. Edge Skew       : 7.8(ps)           20(ps)
Rise Skew              : 7.8(ps) ← Clock skew
Fall Skew              : 7.4(ps)
Max. Rise Buffer Tran.  : 49.7(ps)           100(ps)
Max. Fall Buffer Tran.  : 44.8(ps)           100(ps)
Max. Rise Sink Tran.   : 29.4(ps)           50(ps)
Max. Fall Sink Tran.   : 29.2(ps)           50(ps)
Min. Rise Buffer Tran.  : 49.7(ps)           0(ps)
Min. Fall Buffer Tran.  : 44.8(ps)           0(ps)
Min. Rise Sink Tran.   : 23.6(ps)           0(ps)
Min. Fall Sink Tran.   : 23.5(ps)           0(ps)

view NG_view_typ : skew = 7.8ps (required = 20ps)
```

5. Clock Tree Synthesis (CTS)

- You can see the clock tree by the following command:
 - encounter #> displayClockTree –clk mCLK –allLevel



5. Clock Tree Synthesis (CTS)

- saveDesign test_05_cts.enc

Timing Analysis

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

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

Setup mode	all	reg2reg	in2reg	reg2out	in2out	clkgate
WNS (ns):	0.071	0.071	0.207	N/A	N/A	N/A
TNS (ns):	0.000	0.000	0.000	N/A	N/A	N/A
Violating Paths:	0	0	0	N/A	N/A	N/A
All Paths:	768	512	256	N/A	N/A	N/A

```
-----
```

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: 75.449%

6. Post-CTS Optimization

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

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

Setup mode	all	reg2reg	in2reg	reg2out	in2out	clkgate
WNS (ns):	0.101	0.121	0.101	N/A	N/A	N/A
TNS (ns):	0.000	0.000	0.000	N/A	N/A	N/A
Violating Paths:	0	0	0	N/A	N/A	N/A
All Paths:	768	512	256	N/A	N/A	N/A

```
-----
```

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: 65.532%

6. Post-CTS Optimization

- saveDesign test_06_postctsopt.enc

Timing Analysis

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

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

Setup mode	all	reg2reg	in2reg	reg2out	in2out	clkgate
WNS (ns):	0.101	0.121	0.101	N/A	N/A	N/A
TNS (ns):	0.000	0.000	0.000	N/A	N/A	N/A
Violating Paths:	0	0	0	N/A	N/A	N/A
All Paths:	768	512	256	N/A	N/A	N/A

```
-----
```

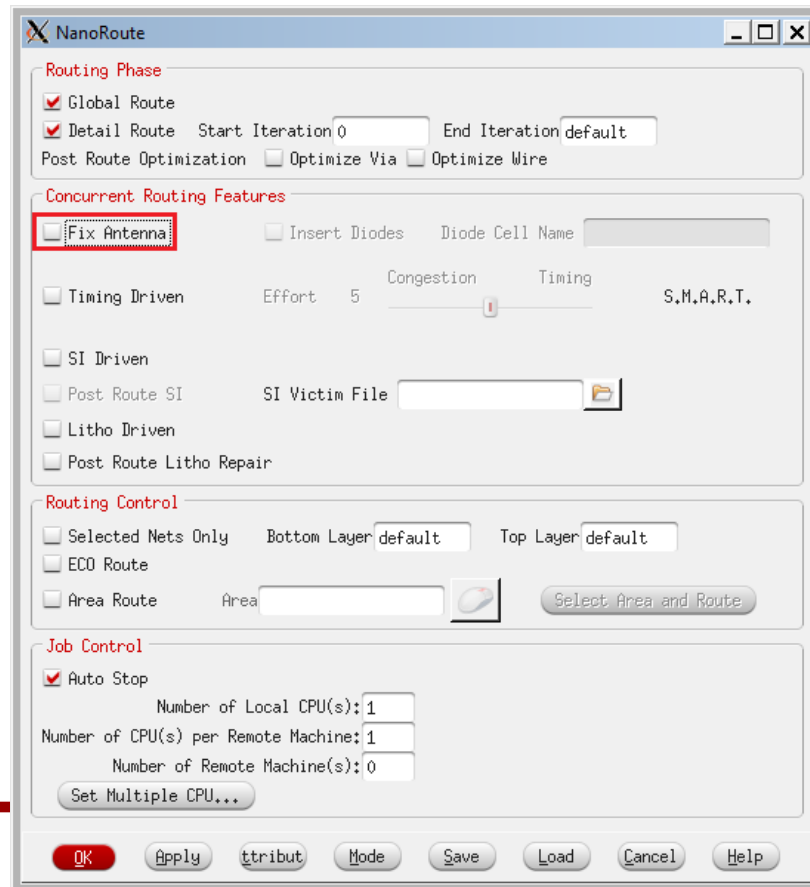
DRVs	Real		Total
	Nr nets(terms)	Worst Vio	Nr nets(terms)
max_cap	0 (0)	0.000	0 (0)
max_tran	0 (0)	0.000	0 (0)
max_fanout	0 (0)	0	0 (0)
max_length	0 (0)	0	0 (0)

```
-----
```

Density: 65.532%

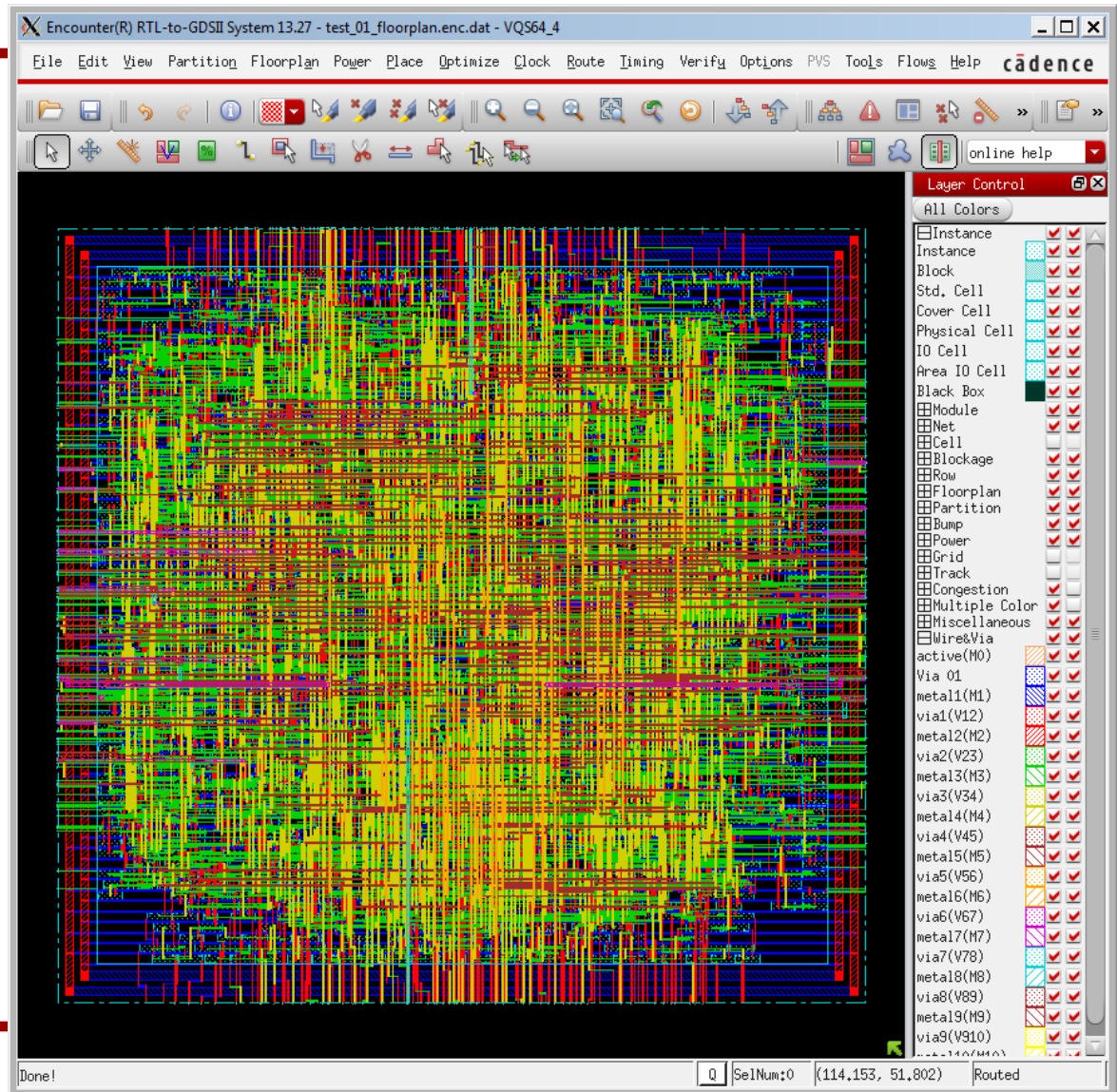
7. Routing

- Click “Route” → “NanoRoute” → “Route...”.
- Turn off “Fix Antenna” and click OK to run routing.



7. Routing

- Routing result.



7. Routing

- Routing result.
- Wirelength: 52,077um
- No DRC violations.

```
#Complete Detail Routing.
#Total number of nets with non-default rule or having extra spacing = 38
#Total wire length = 52077 um
#Total half perimeter of net bounding box = 48628 um.
#Total wire length on LAYER metal1 = 1604 um.
#Total wire length on LAYER metal2 = 15448 um.
#Total wire length on LAYER metal3 = 19261 um.
#Total wire length on LAYER metal4 = 9446 um.
#Total wire length on LAYER metal5 = 4644 um.
#Total wire length on LAYER metal6 = 1314 um.
#Total wire length on LAYER metal7 = 236 um.
#Total wire length on LAYER metal8 = 68 um.
#Total wire length on LAYER metal9 = 56 um.
#Total wire length on LAYER metal10 = 0 um.
#Total number of vias = 24082
#Up-Via Summary (total 24082):
#
#-----
# Metal 1      10605
# Metal 2      10028
# Metal 3       2748
# Metal 4       515
# Metal 5       147
# Metal 6        25
# Metal 7         8
# Metal 8         6
#-----
#              24082
#
#Total number of DRC violations = 0
```

7. Routing

- saveDesign test_07_route.enc

Timing Analysis

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

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

Setup mode	all	reg2reg	in2reg	reg2out	in2out	clkgate
WNS (ns):	0.102	0.139	0.102	N/A	N/A	N/A
TNS (ns):	0.000	0.000	0.000	N/A	N/A	N/A
Violating Paths:	0	0	0	N/A	N/A	N/A
All Paths:	768	512	256	N/A	N/A	N/A

```
-----
```

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: 65.532%

8. Post-Routing Optimization

- Although we already satisfied the timing without any further optimization after routing, we will run post-routing optimization.
 - encounter #> optDesign –postRoute

```
-----
optDesign Final Non-SI Timing Summary
-----
```

Setup mode	all	reg2reg	in2reg	reg2out	in2out	clkgate
WNS (ns):	0.102	0.139	0.102	N/A	N/A	N/A
TNS (ns):	0.000	0.000	0.000	N/A	N/A	N/A
Violating Paths:	0	0	0	N/A	N/A	N/A
All Paths:	768	512	256	N/A	N/A	N/A

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: 65.532%

8. Post-Routing Optimization

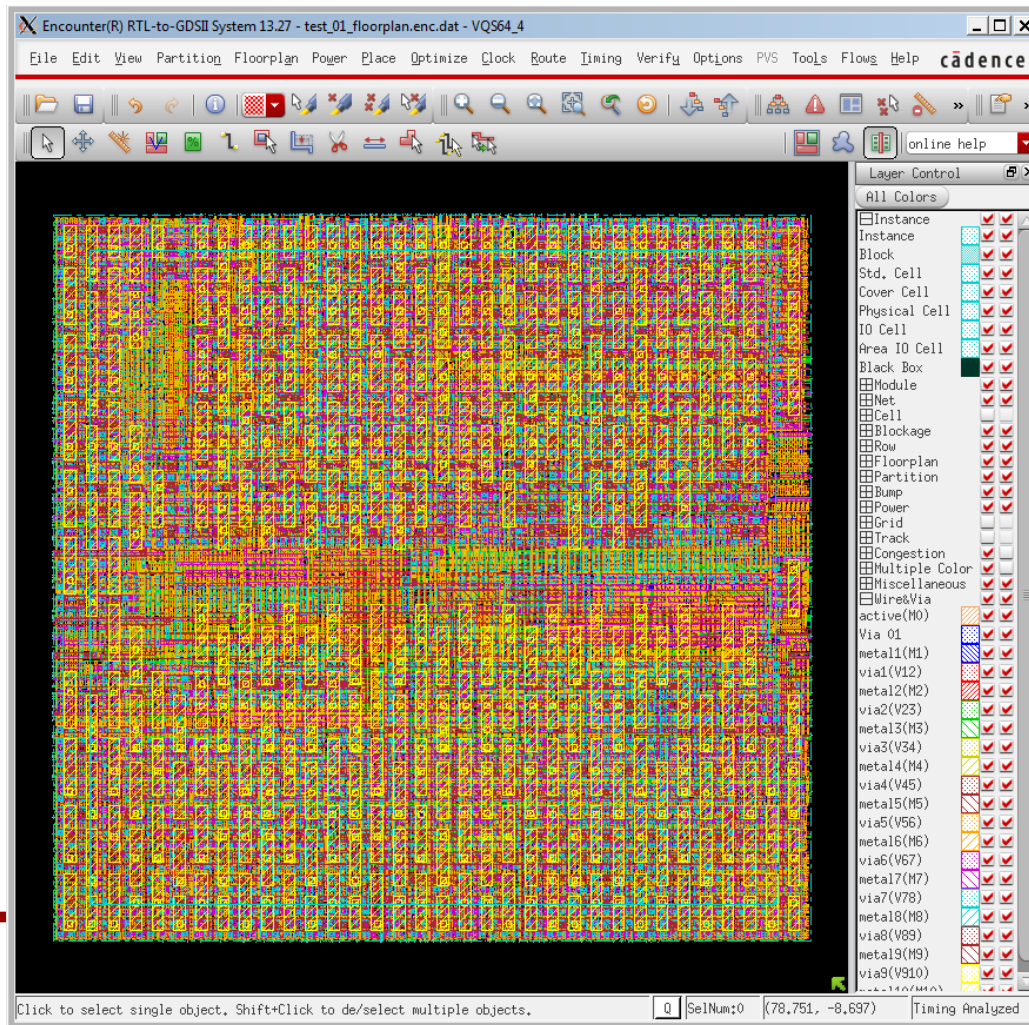
- saveDesign test_08_postrouteopt.enc

9. Fill Insertion

- Click “Route” → “Metal Fill” → “Setup...”.
- Click “Load” and choose “metalfill.cmd” to load the setting I made.
- Click OK.
- Click “Route” → “Metal Fill” → “Add”.
- Click OK to insert metal fills.

9. Fill Insertion

- The following shows my fill insertion result.



Timing Analysis

- Run the following command to analyze timing.
 - `encounter #> timeDesign -postRoute`

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

Setup mode	all	reg2reg	in2reg	reg2out	in2out	clkgate
WNS (ns):	0.102	0.139	0.102	N/A	N/A	N/A
TNS (ns):	0.000	0.000	0.000	N/A	N/A	N/A
Violating Paths:	0	0	0	N/A	N/A	N/A
All Paths:	768	512	256	N/A	N/A	N/A

```
-----
```

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: 65.532%

9. Fill Insertion

- saveDesign test_09_fill.enc