

Layout Synthesis Using Automatic Place-and-Route Tools - Part 2

Jaeha Kim, Sung-Joon Lee, and Eunseo Kim
Mixed-Signal IC and System Group
Seoul National University
May 20. 2016

Pre-Routing Critical Wires

- ▶ Pre-routing critical signals may be required for
 - ▶ Routing timing-critical signals
 - ▶ Routing sensitive analog signals (e.g. that require shielding)
 - ▶ Routing multi-phase clock signals that require delay-matching
- ▶ **Using Custom Designer is necessary to route critical wires**

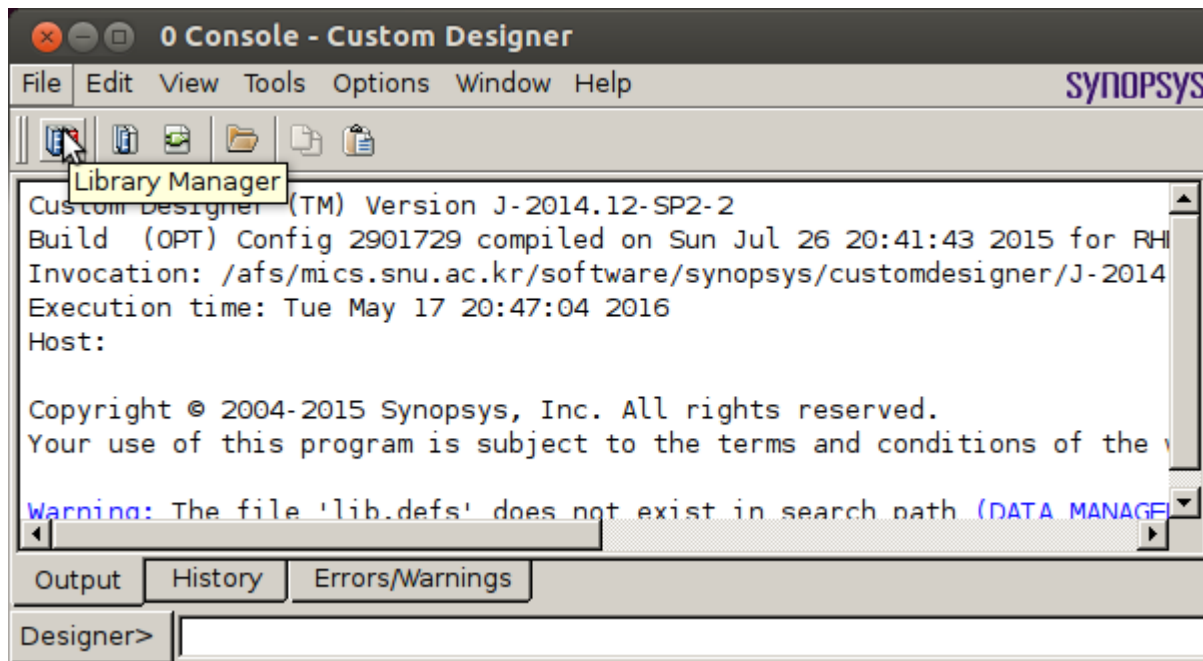
Pre-Routing with Custom Designer

- ▶ Synopsys provides a unified solution for cell-based and custom designs : **Custom Designer**
- ▶ You can transfer designs from IC Compiler to Custom Designer and perform shape-based editing (e.g. custom routing)
 - ▶ In fact, you can do this at any point during floorplanning, placement, or power/ground routing and clock tree synthesis
- ▶ Also, Custom Designer can read the Cadence Virtuoso OpenAccess database

Pre-Routing with Custom Designer

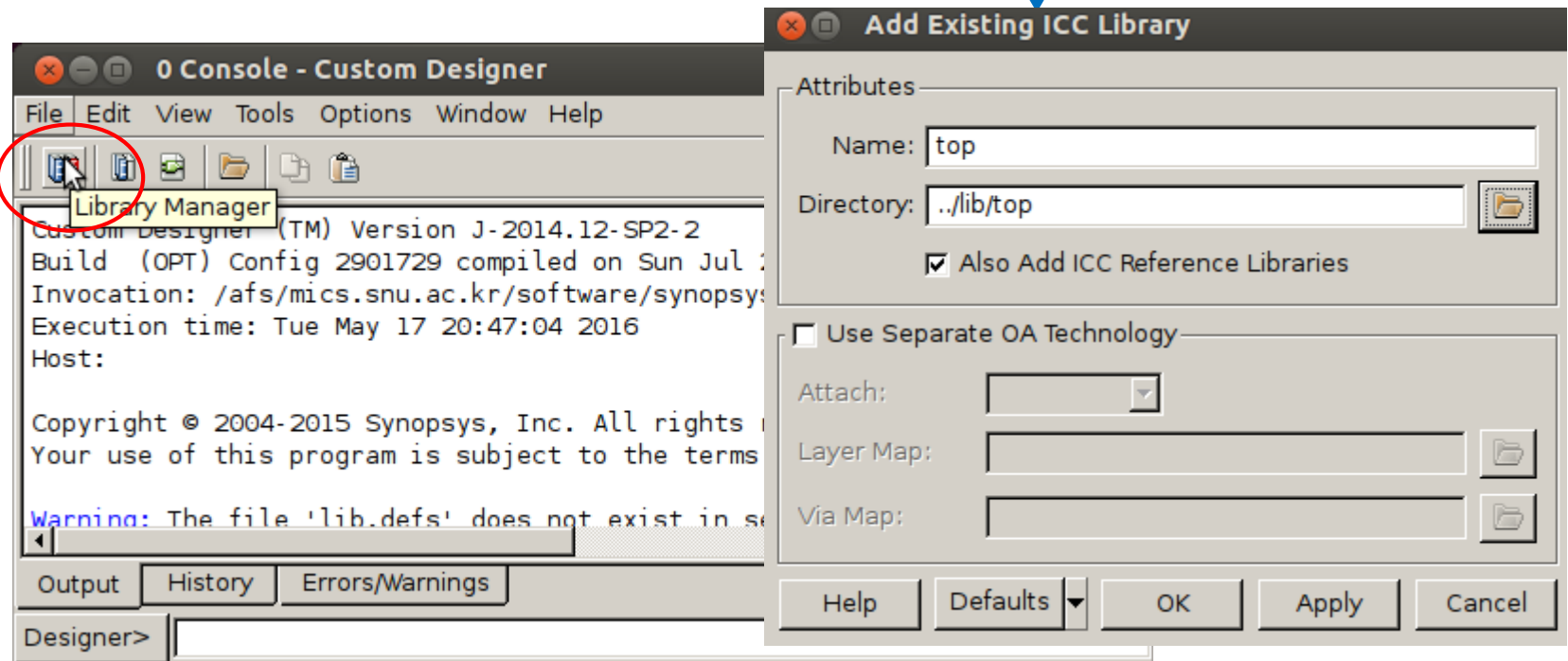
- ▶ To execute Custom Designer, type as below

```
$~/IDEC_CBDF/Place_Route/script> cdesigner &
```



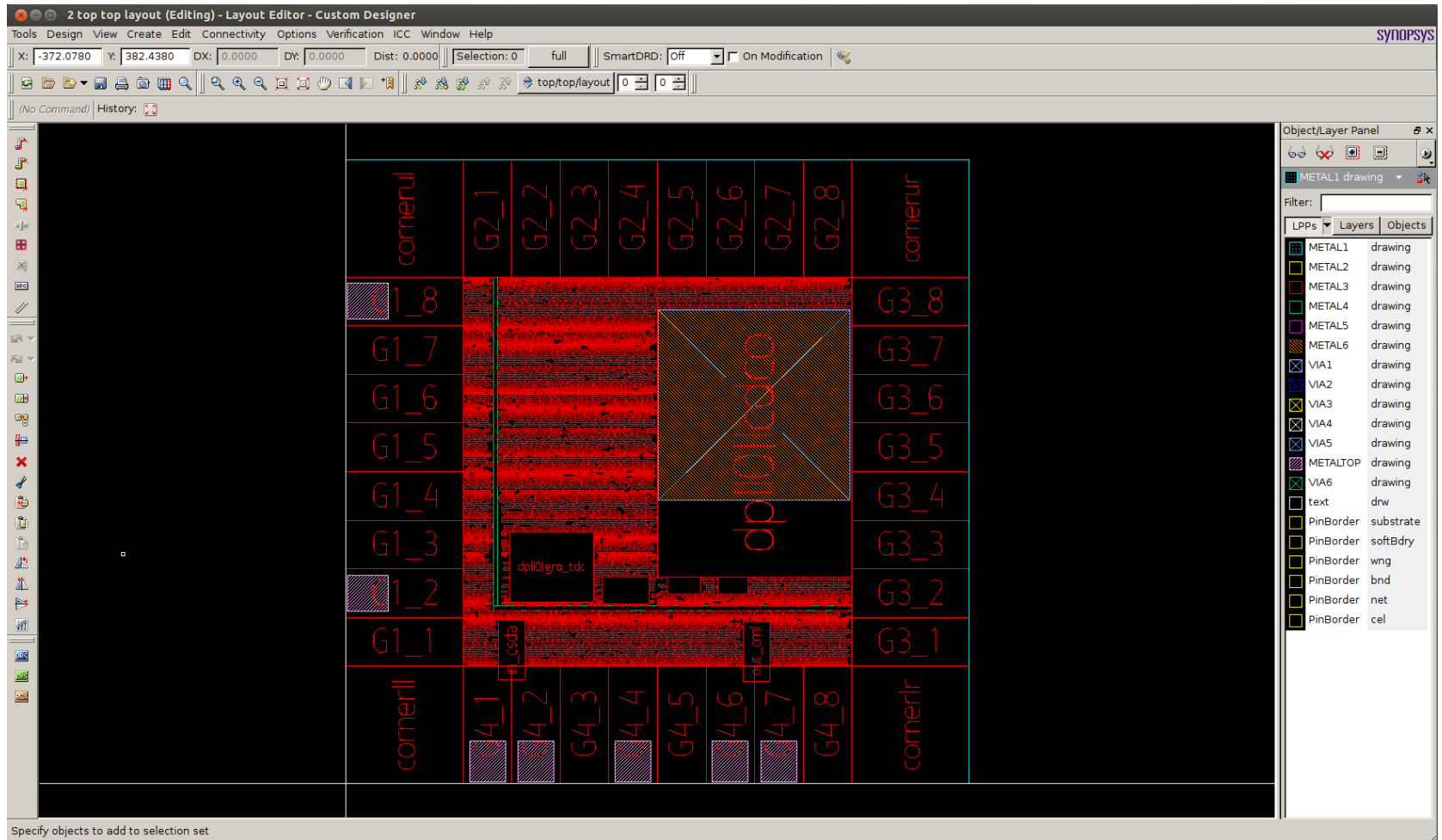
Pre-Routing with Custom Designer

- ▶ To load TOP MilkyWay Library
 - ▶ Tools → Library Manager (or click Library Manager button)
 - ▶ On Library Manager, click File → Add ICC Library ...
 - ▶ Next, set attributes as following



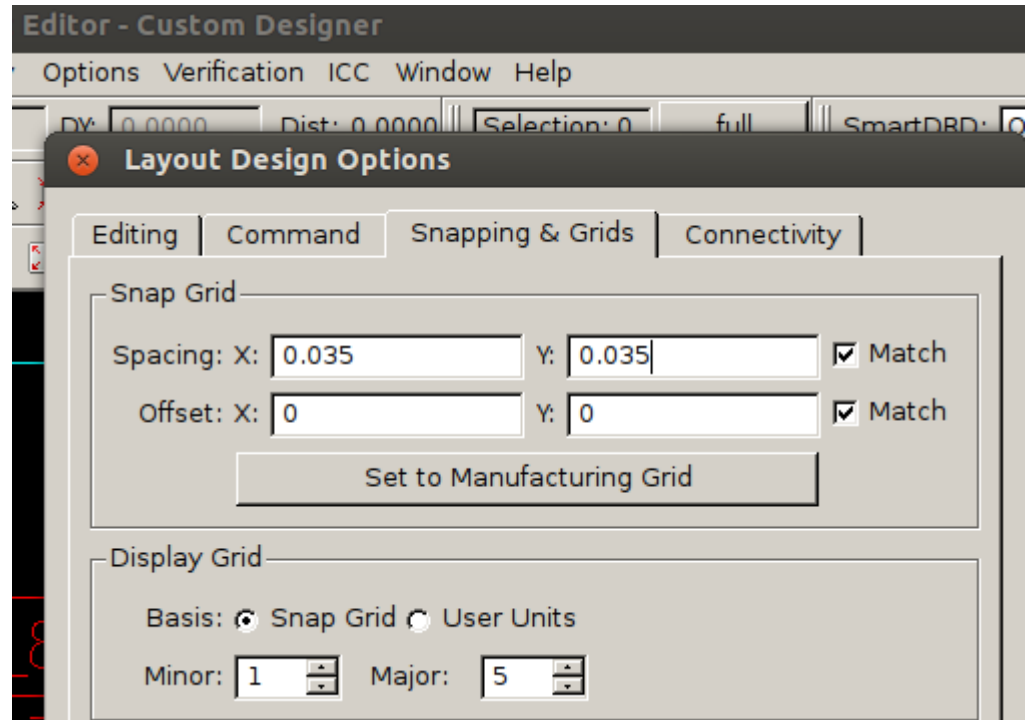
Pre-Routing with Custom Designer

► Open Custom Designer Layout Editor



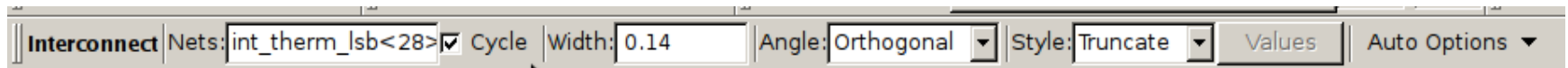
Pre-Routing with Custom Designer

- ▶ Before beginning the pre-routing, let's setup the grid for our convenience
- ▶ Click options > Designs > Snapping & Grids
- ▶ Set spacing X & Y with 0.035

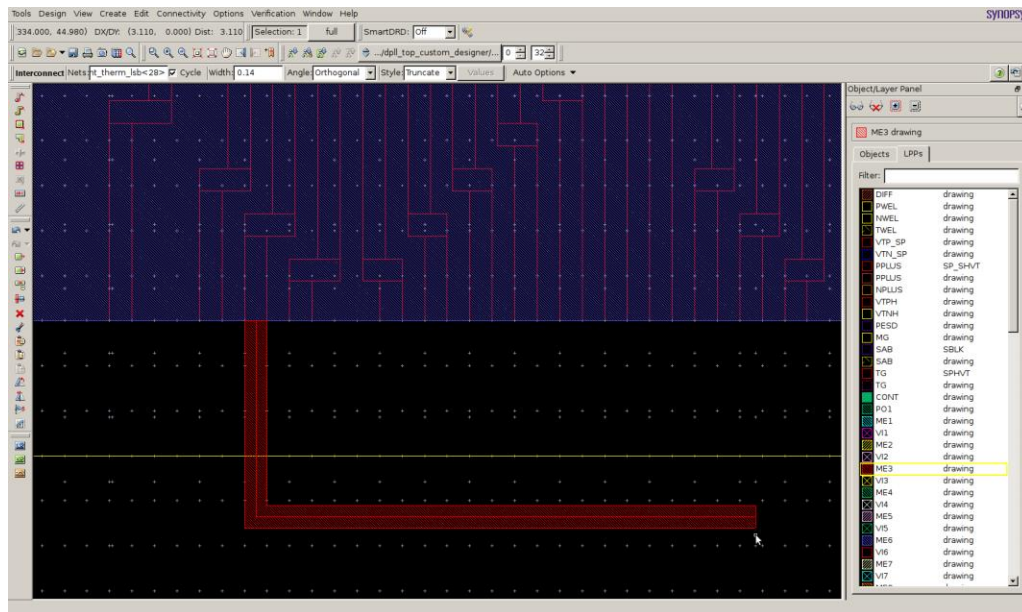


Pre-Routing with Custom Designer

- ▶ In Custom Designer Layout Editor,
 - ▶ Create → Interconnect , or press P
 - ▶ Next, Type Nets and Width



- ▶ Draw custom nets that you want preroute



Exercise

- ▶ Route nets {clk_ref_p, clk_ref_n, clk_outp, clk_outn} using METAL₄, the net width is 0.14 (=4-lambda)
- ▶ clk_ref_p, clk_ref_n : connection between in_csda & empty pads (G4_1, G4_2)
 - ▶ Note. G4_1 for 'clk_ref_n', G4_2 for 'clk_ref_p'
- ▶ clk_outp, clk_outn : connection between out_cml & empty pads (G4_6, G4_7)
 - ▶ Note. G4_6 for 'clk_outn', G4_7 for 'clk_outp'

Exercise

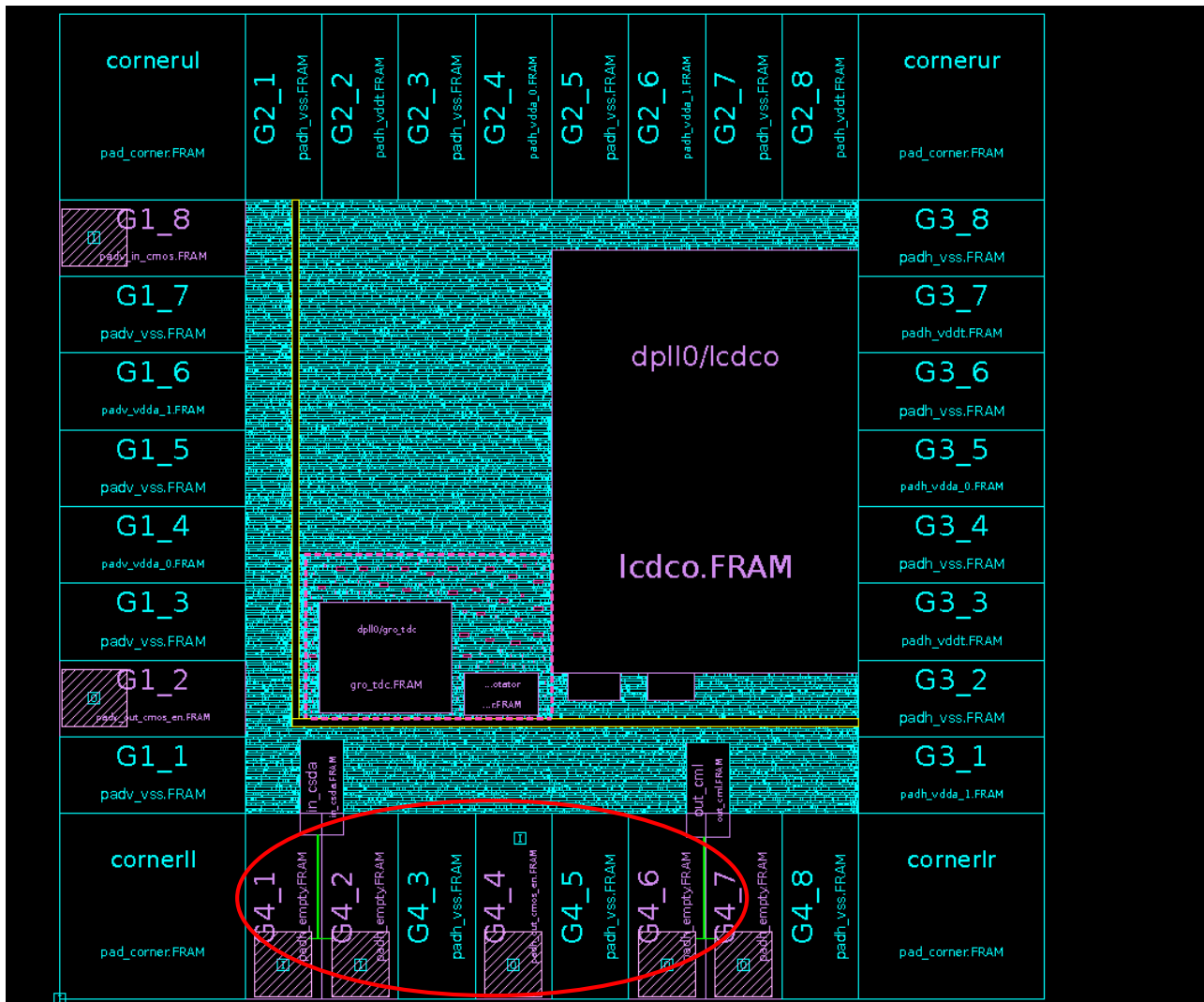
- ▶ *Useful Keyboard Shortcut
 - ▶ m : move
 - ▶ c : Copy
 - ▶ s: stretch
 - ▶ f : zoom fit
 - ▶ Shift+f : Reveal the layout of cells
 - ▶ Ctrl+f : Hide the layout of cells

Save and Quit the Custom Designer

- ▶ After finishing your routing, click Design > Save, or press F2 to save your design
- ▶ Close Custom Designer and let's open it through IC Compiler

```
$~/IDEC_CBDF/Place_Route/script> make open_lib
```

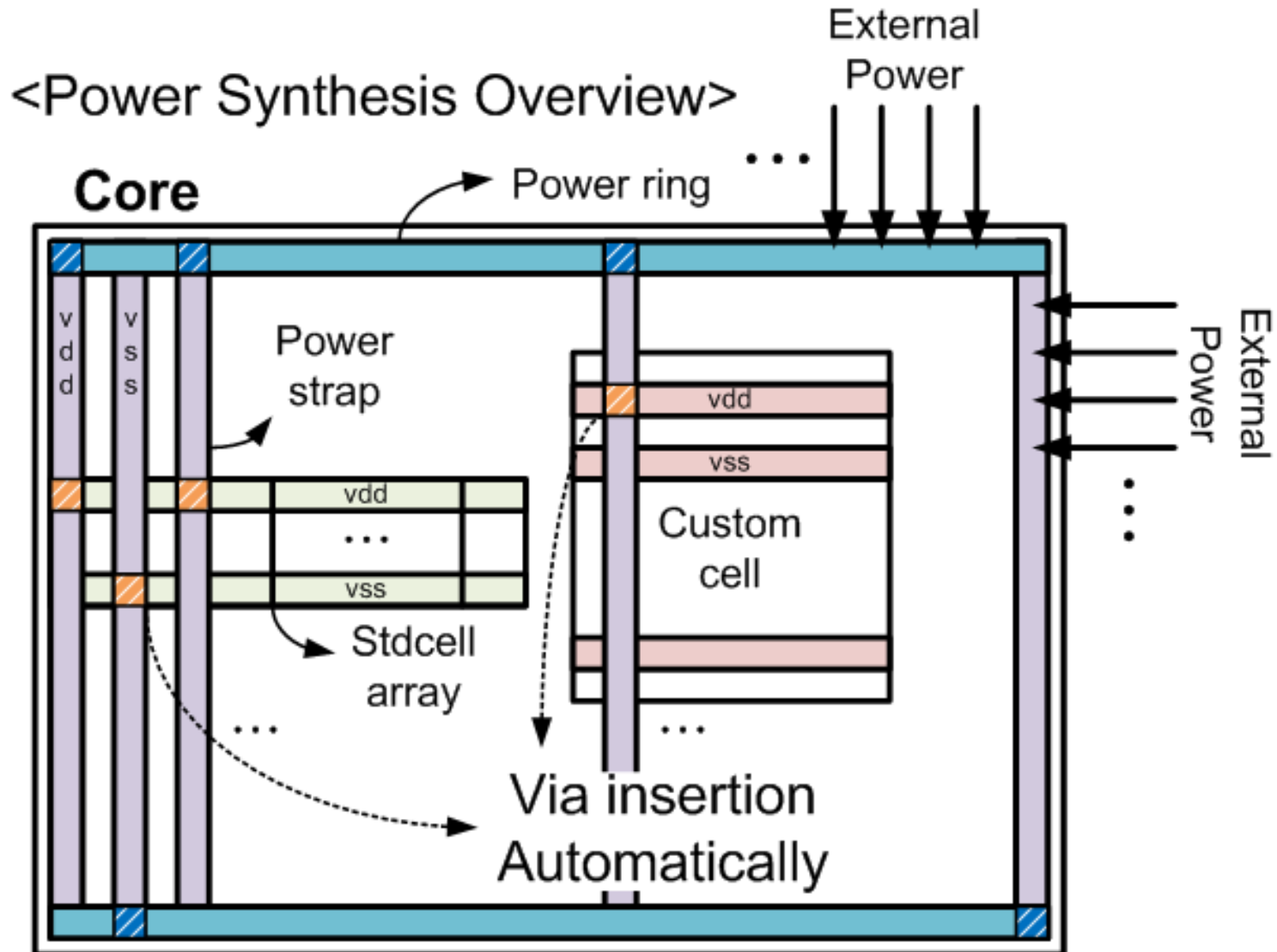
Pre-Route Result



Power Synthesis

- ▶ Create power strap in overall chip, automatically connect to the power pin of the cell to the corresponding power strap

Power Synthesis



Power Synthesis

► \$~/IDEC_CBDF/Place_Route/script> vi icc_scripts/power_syn.tcl

```
Change selection [get_cells -all -hierarchical -filter \
```

```
...
```

```
set_object_fixed_edit [get_selection] 1
```

Fixing placement of
digital cells

```
create_power_straps \
```

Creating Vertical
Power Straps

```
...
```

```
-direction vertical -layer ${POWER_STRAP_LAYER_V} \
```

```
-step $STRAP_INCREMENT_V -nets "$DPLL_POWER_NET ..." \
```

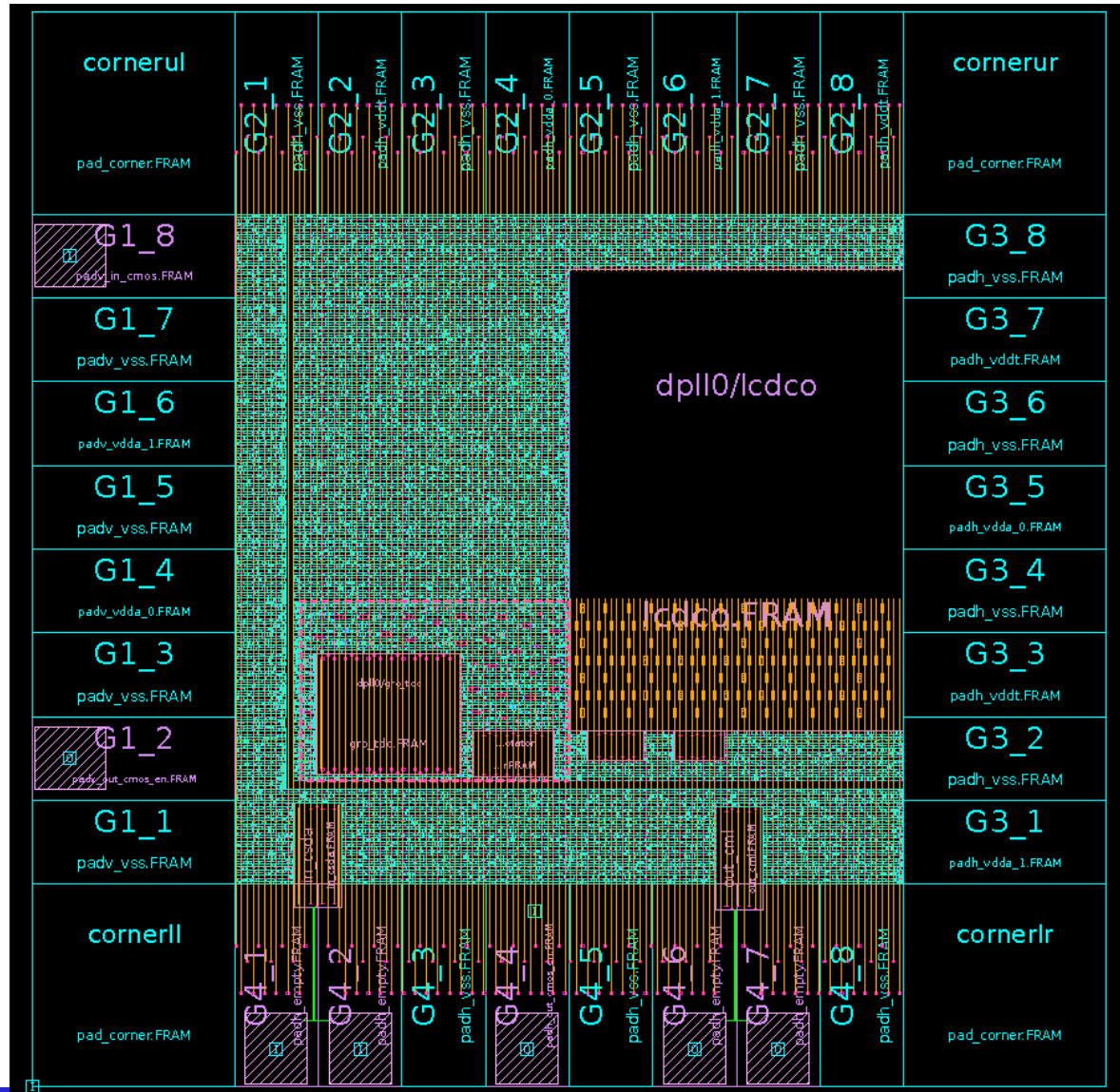
```
...
```

Power Synthesis

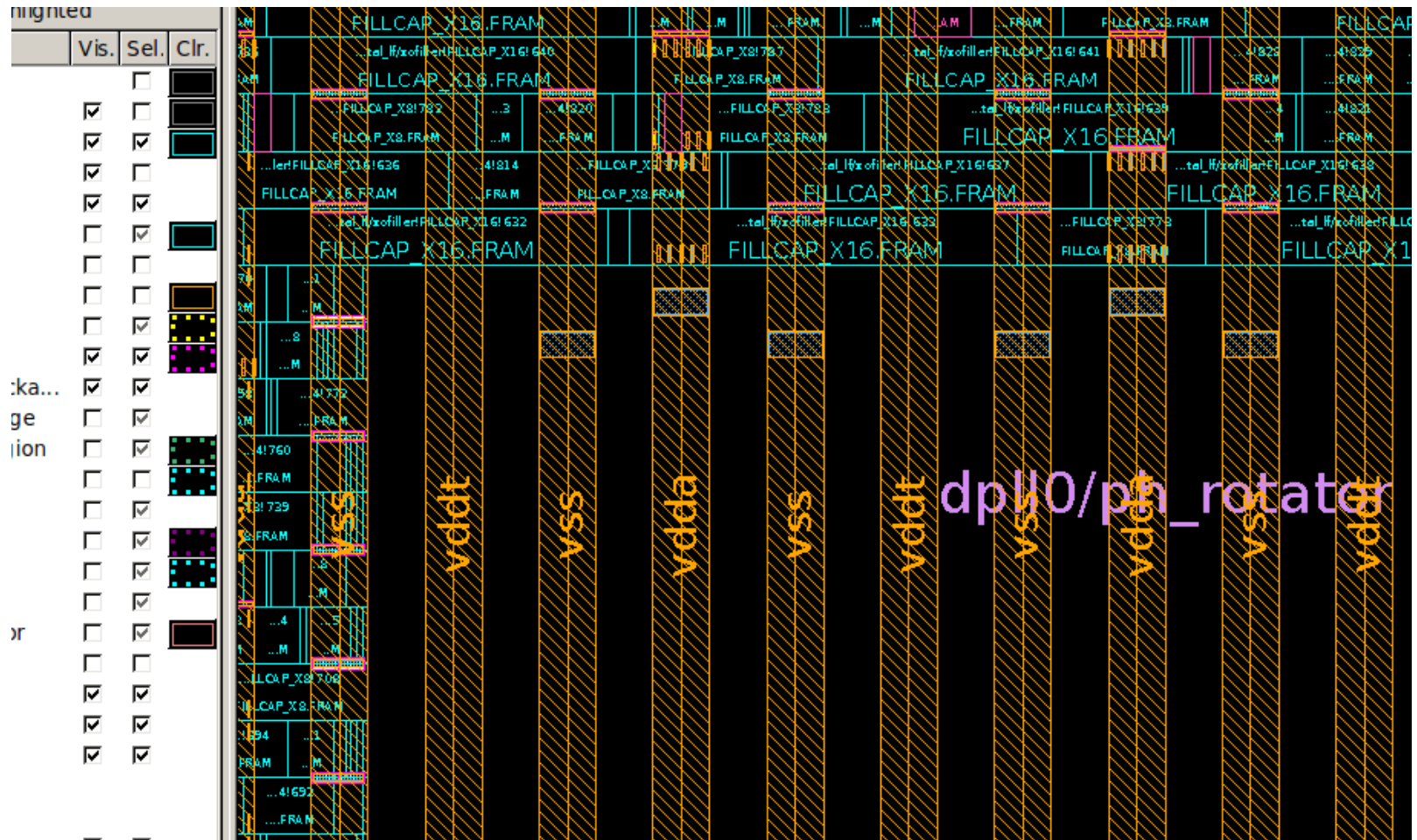
- For Power synthesis, run:

```
$~/IDEC_CBDF/Place_Route/script> make power_syn
```


Power Synthesis Result



Power Synthesis Result



Routing Optimization

- ▶ Routing optimization process includes :
 - ▶ Clock tree routing
 - ▶ Insert redundant via
 - ▶ Initial/detailed routing

Routing Optimization

► \$~/IDEC_CBDF/Place_Route/script> vi icc_scripts/route_optimizer.tcl

```
removed_preferred_routing_direction -layers {METAL1 ... METALTOP}
```

```
set_preferred_routing_direction -layers ... -direction vertical
```

Setting routing
direction

```
...
```

```
set_route_zrt_common_options -read_user_metal_blockage_layer true ...
```

Routing options to avoid DRC
including antenna rules

```
...
```

```
route_group -all_clock_nets -no_track -no_detail
```

Clock tree routing

```
route_opt -initial_route_only
```

Initial routing

```
insert_zrt_redundant_vias -effort high
```

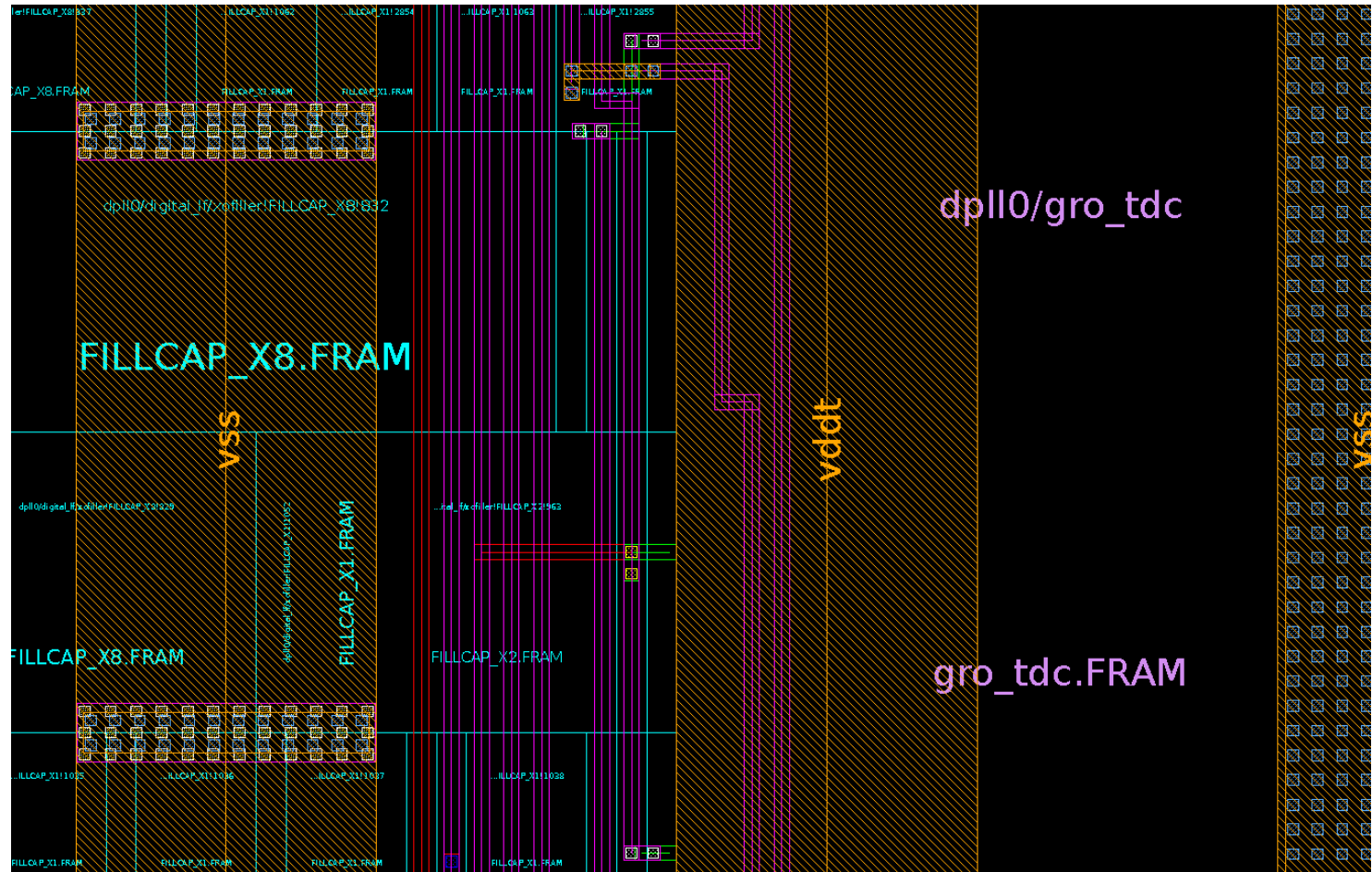
Inserting redundant vias

Routing Optimization

- For Routing optimization, run:

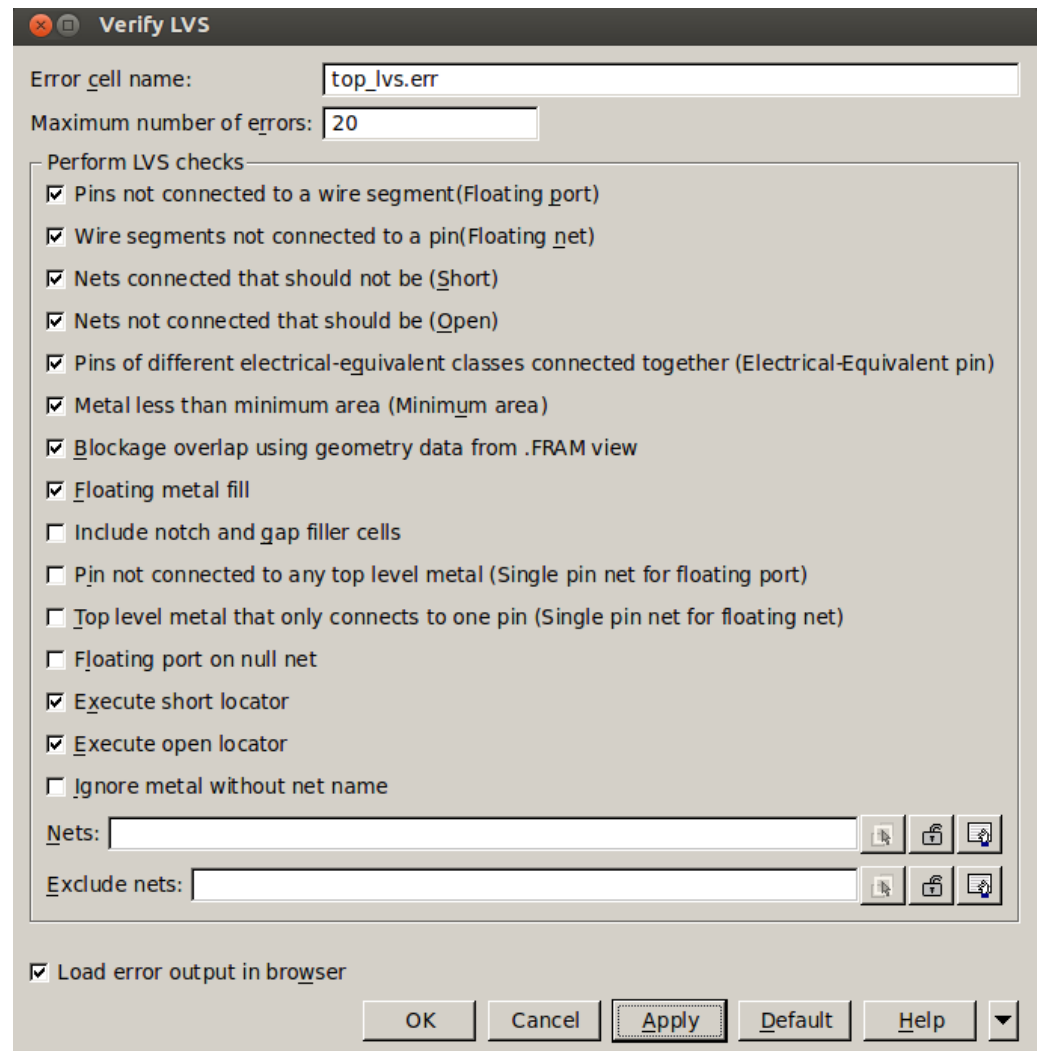
```
$~/IDEC_CBDF/Place_Route/script> make route_optimizer
```

Routing Optimization Result



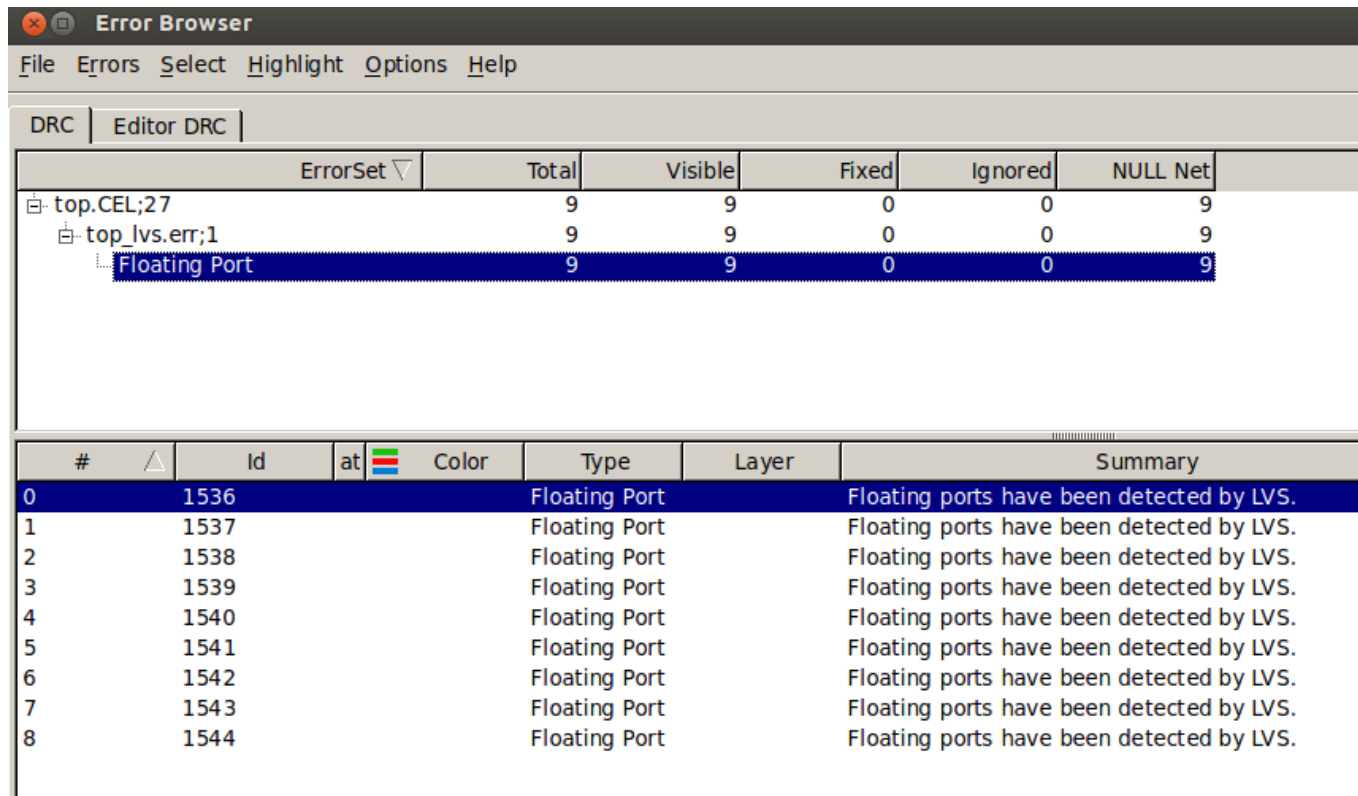
DRC/LVS in IC Compiler

- ▶ In ICC GUI, open the design, and click verification > LVS
- ▶ Check the “Execute short/open locator” to detect error easily



DRC/LVS Result

- ▶ Floating port error : port that doesn't used in design, so it is not an actual error



ErrorSet ▾	Total	Visible	Fixed	Ignored	NULL Net
top.CEL;27	9	9	0	0	9
top_lvs.err;1	9	9	0	0	9
Floating Port	9	9	0	0	9

#	△	Id	at	Color	Type	Layer	Summary
0		1536			Floating Port		Floating ports have been detected by LVS.
1		1537			Floating Port		Floating ports have been detected by LVS.
2		1538			Floating Port		Floating ports have been detected by LVS.
3		1539			Floating Port		Floating ports have been detected by LVS.
4		1540			Floating Port		Floating ports have been detected by LVS.
5		1541			Floating Port		Floating ports have been detected by LVS.
6		1542			Floating Port		Floating ports have been detected by LVS.
7		1543			Floating Port		Floating ports have been detected by LVS.
8		1544			Floating Port		Floating ports have been detected by LVS.

How to fix DRC/LVS errors in IC Compiler

- ▶ Most of errors occur due to following reasons :
 - ▶ Bad floorplanning
 - ▶ Human mistake in pre-routing
 - ▶ Bad pin placement in cells
- ▶ Fixing above issues is as easy as pie, compared to fixing errors in chip designed manually
- ▶ **Cell-based Design flow enables designer to do “construct by correction” easily**