



VLSI Crash Course

Automatic Place and Route (APR)

姚云瀚、盧真玄

Energy-Efficient Circuit and System Lab

2018.07.17



About

◆ Goal

- How to run APR
- What the tool (wants to) do in each step (brief)

◆ Ref

- CIC (C106) Cell-Based IC Physical Design and Verification with ~~SOC Encounter~~ Training Manual, July-2016

Innovus



Cell-Based IC Physical Design and Verification - Encounter Digital Implementation

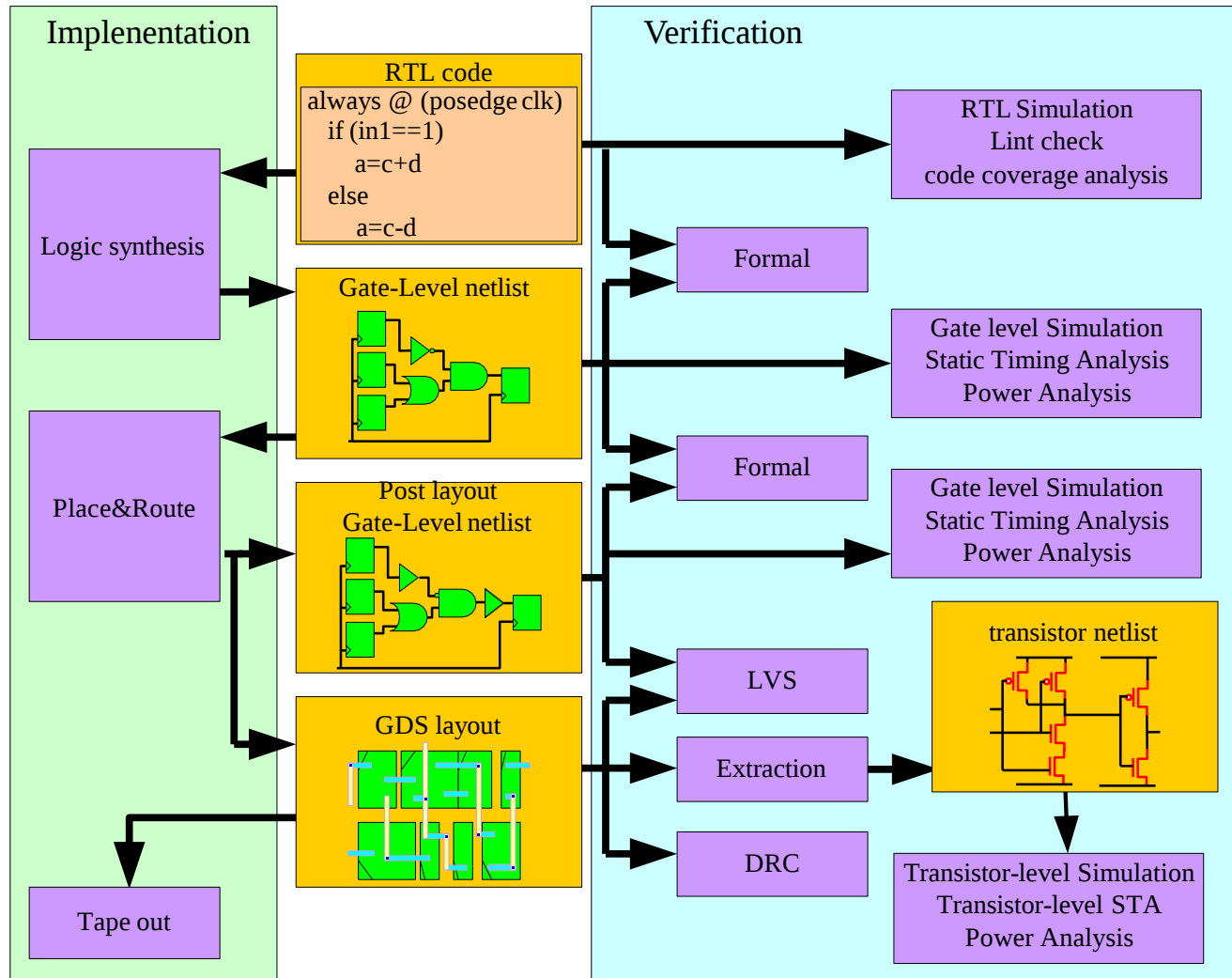
Class Schedule

- Day1
 - Design Flow Over View
 - Prepare Data
 - Getting Started
 - Importing Design
 - Specify Floorplan
 - Power Planning
 - Placement
- Day2
 - Synthesize Clock Tree
 - ~~– Timing Analysis~~
 - ~~– Trial Route~~
- ~~• Power Analysis~~
 - SRoute
 - NanoRoute
 - Fill Filler
 - Output Data
- Day3
 - DRC
 - LVS
 - ~~• extraction/nanosim~~
 - ~~• Foundation flow~~

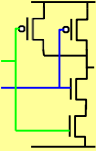
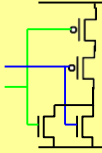
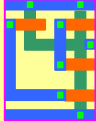
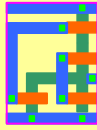
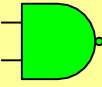


OVERVIEW

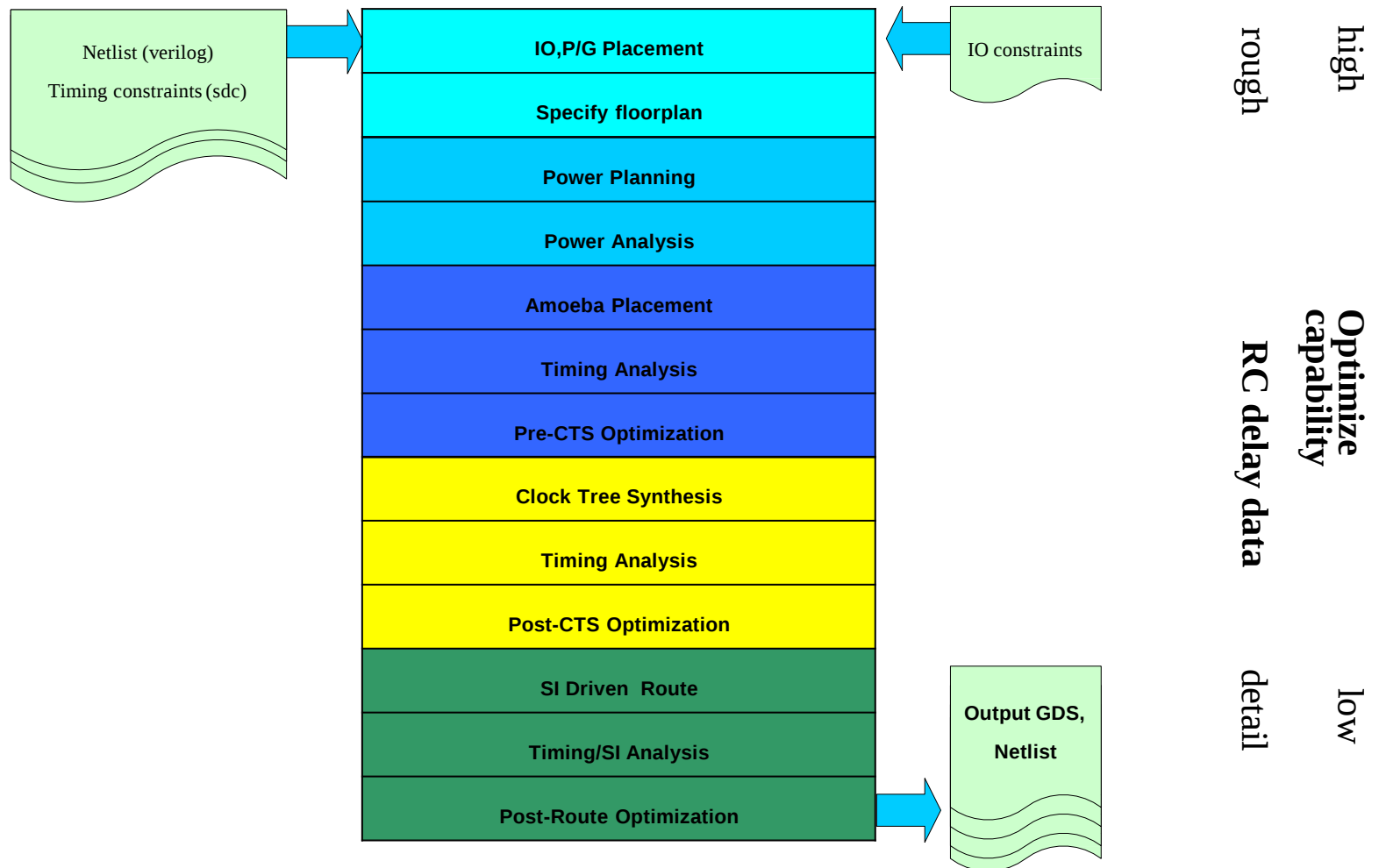
Cell-Based Design Flow



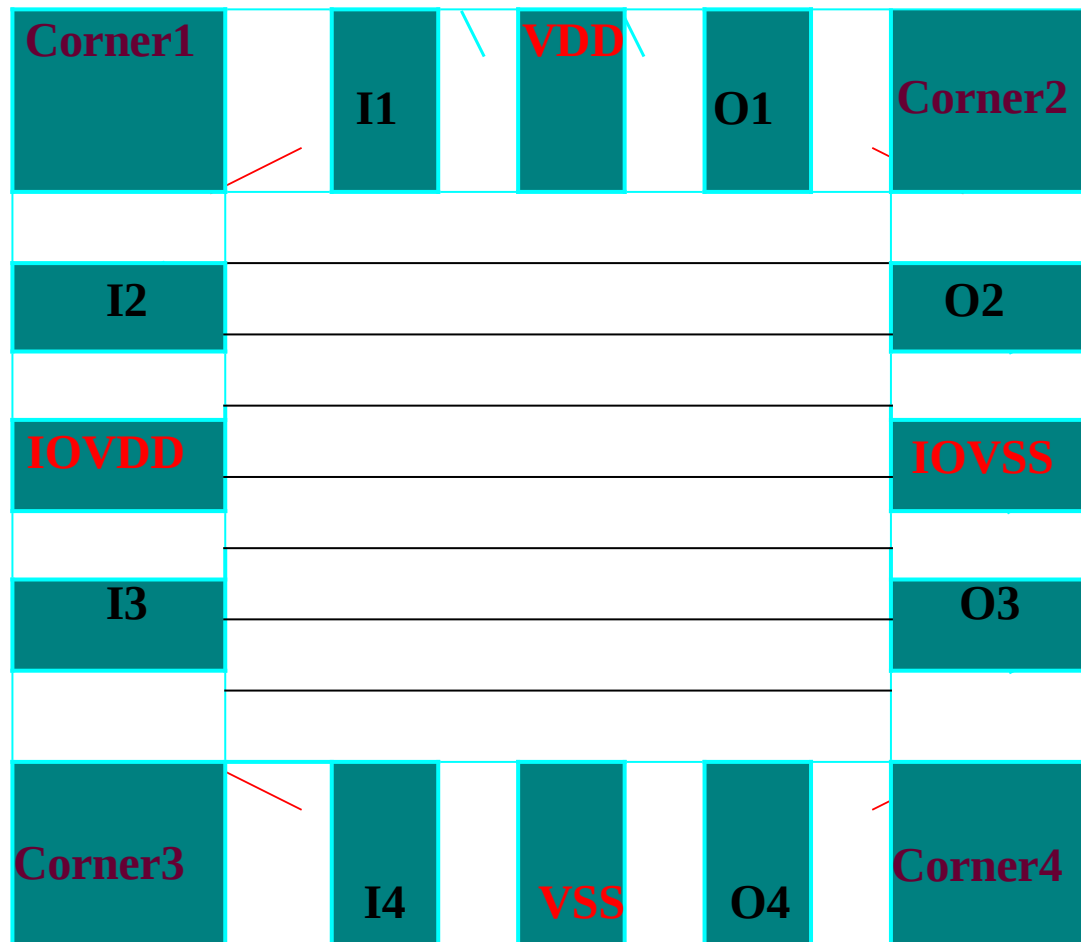
Cell Library

function	NAND	NOR	XOR	INV	ADD	FF	
schematic							
layout							
symble							
timing	A1→O 0.1ns A2→O 0.2ns	A1→O 0.1ns A2→O 0.2ns					
power	A1→O 0.1pw A2→O 0.2pw	A1→O 0.1pw A2→O 0.2pw					
abstract							

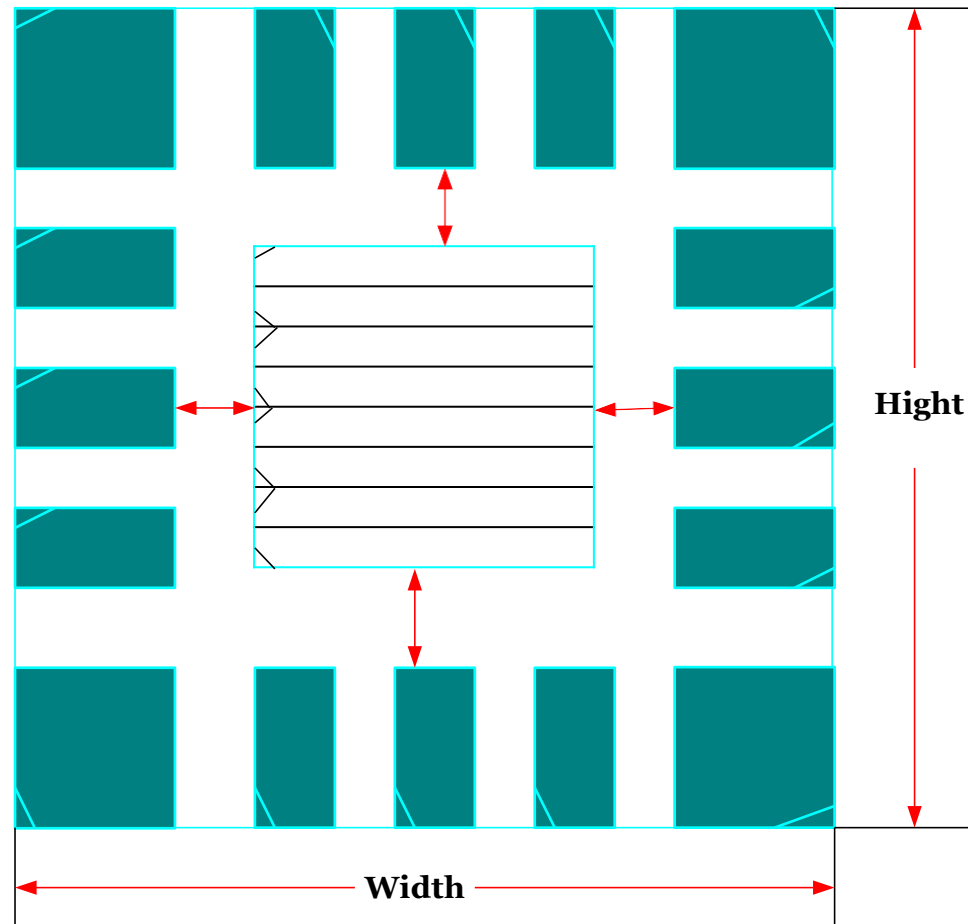
Innovus P&R flow



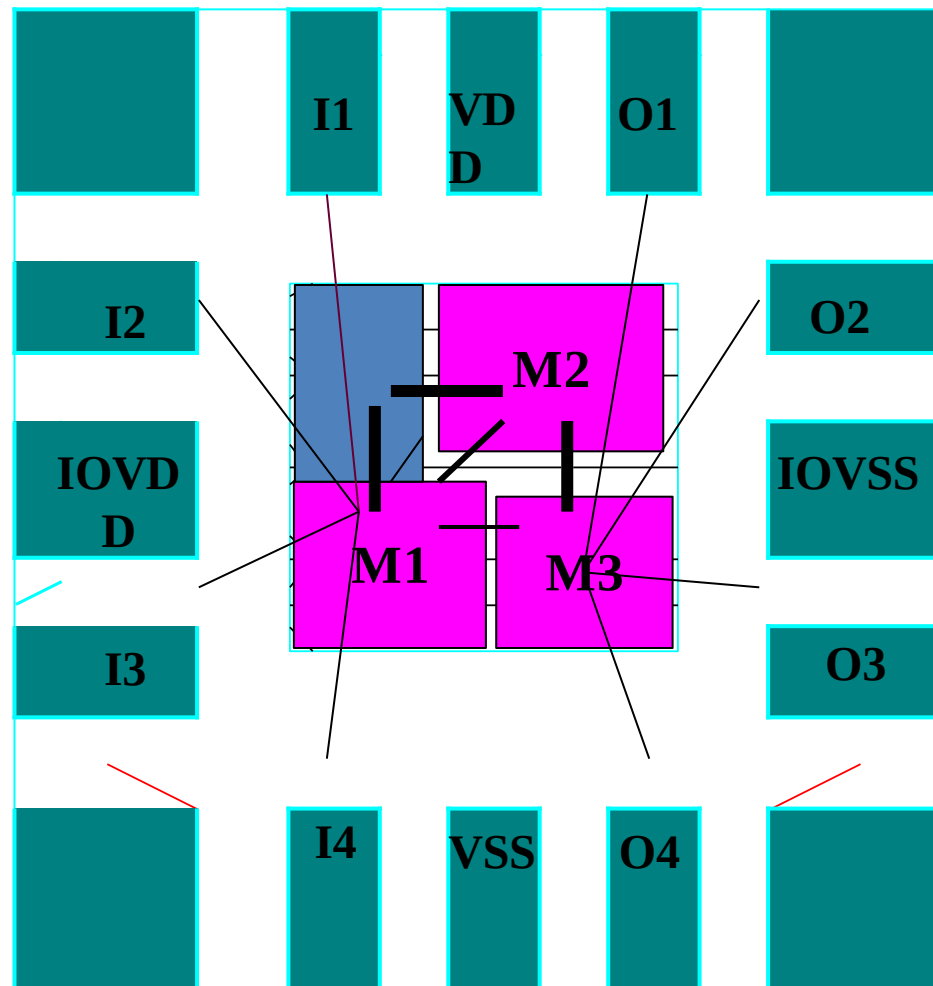
IO, P/G Placement



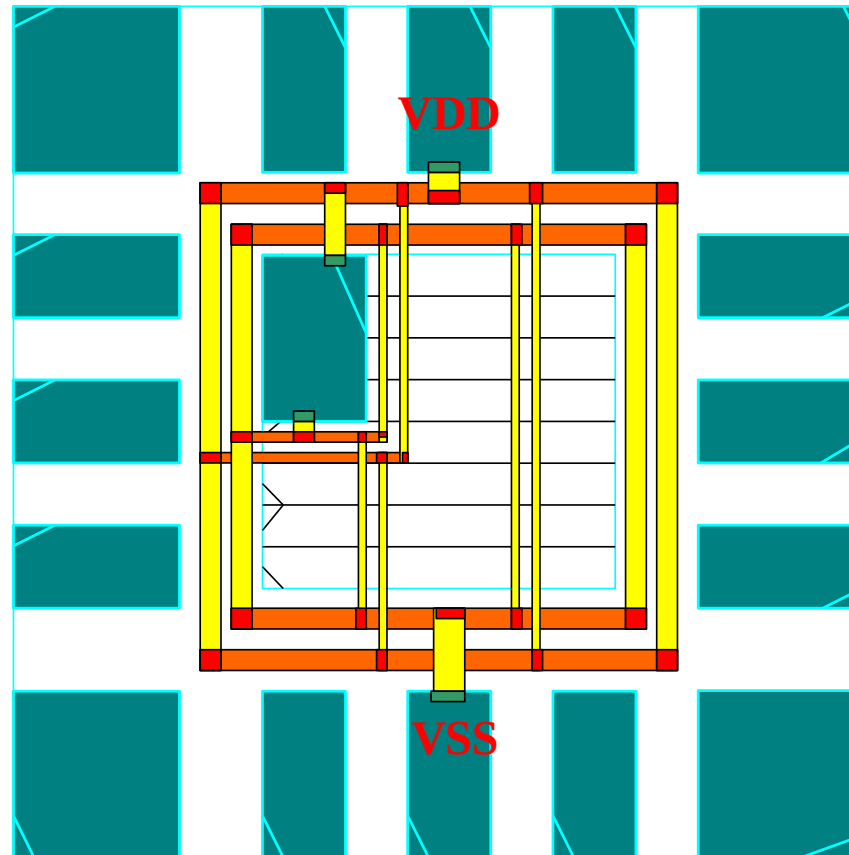
Specify Floorplan



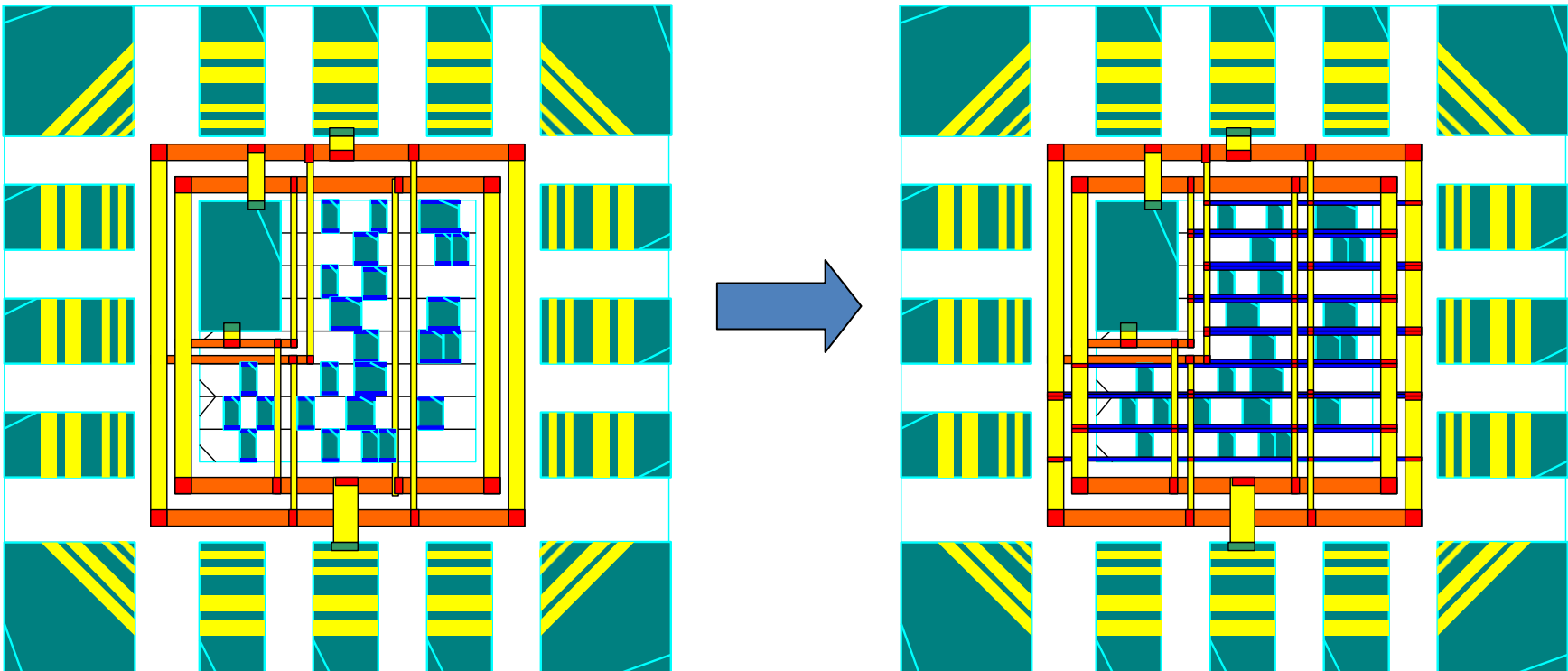
Floorplan



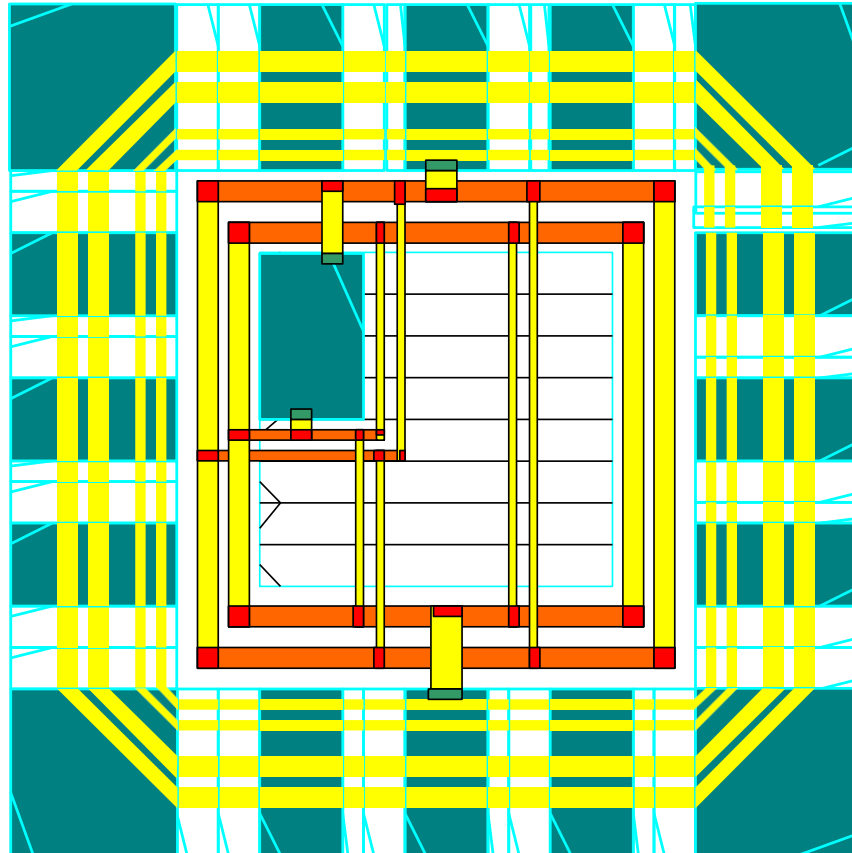
Power Planning



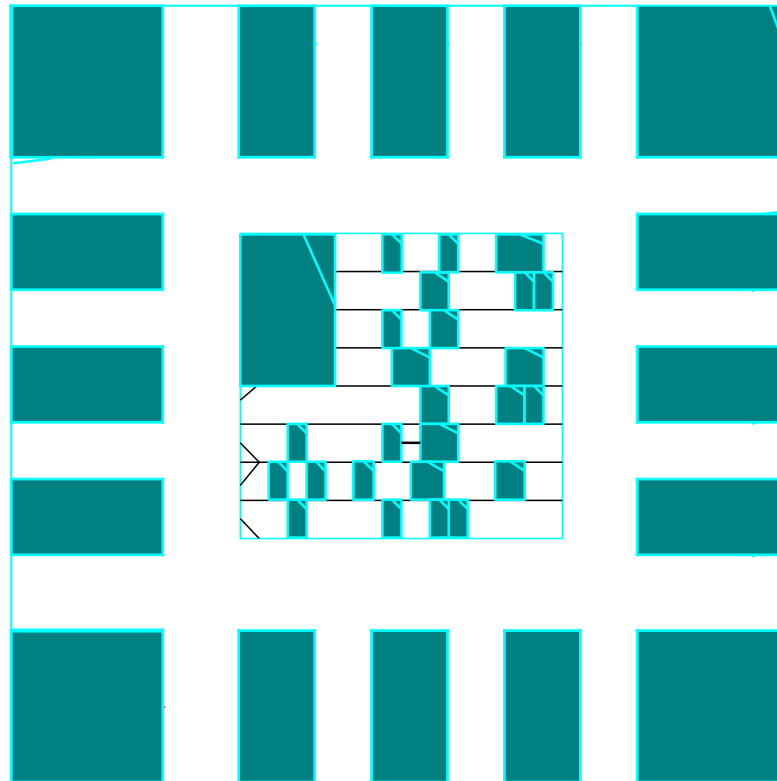
Power Route



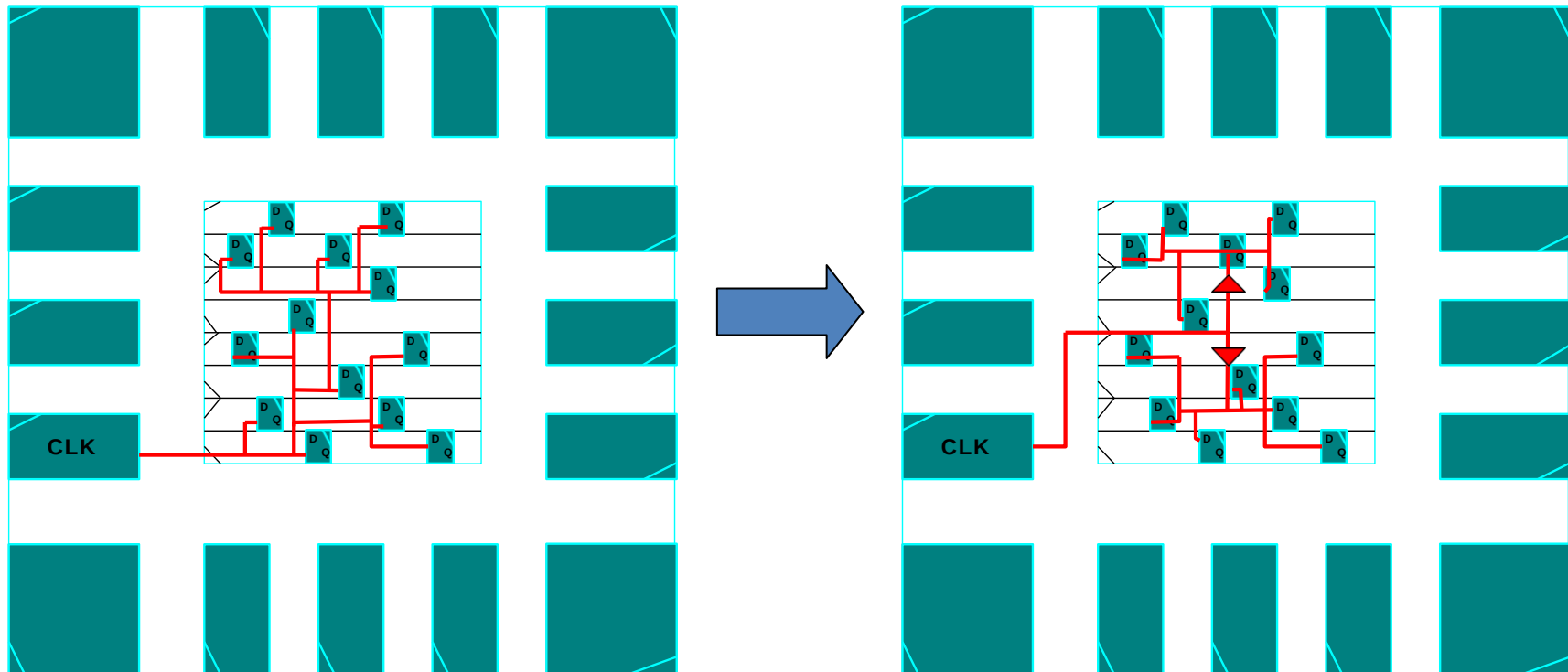
Add IO Filler



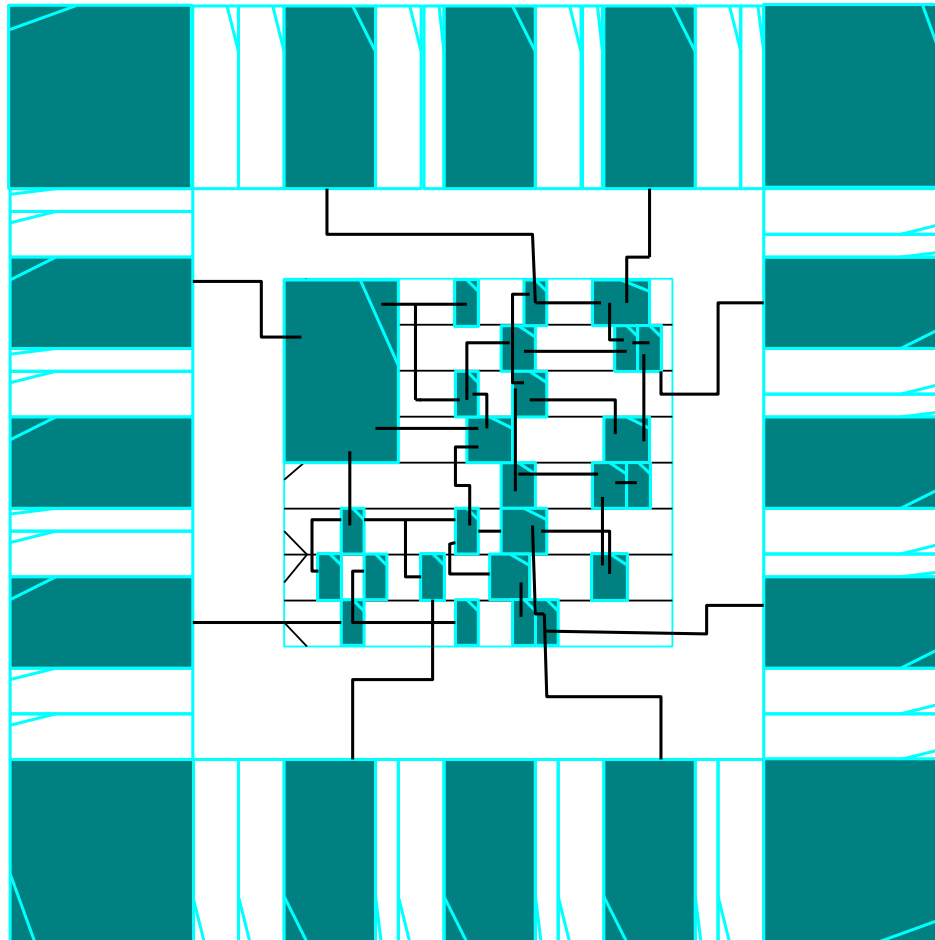
Placement



Clock Tree Synthesis



Routing





BEFORE APR

Prepare Data

- Library
 - Physical Library (LEF)
 - Timing Library (LIB)
 - Capacitance Table
 - Celtic Library Noise model
- User Data
 - Gate-Level Netlist (verilog)
 - SDC Constraints
 - IO Constraints
 - Scan Def

LEF Format Process Technology

Layers

 POLY

 Contact

 Metal1

 Via1

 Metal2

Design Rule

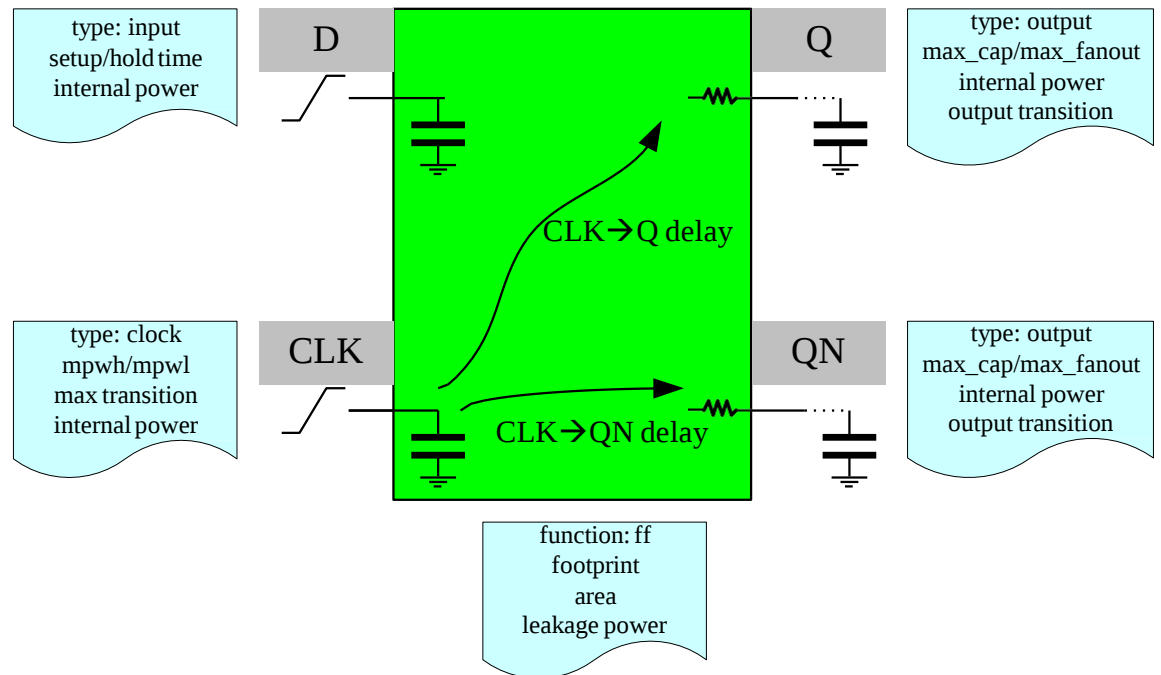
Net width
Net spacing
Area
Enclosure
Wide metal
Slot
Antenna
Current density

Parasitic

Resistance
Capacitance

LIB Format

- Operating condition
 - slow, fast, typical
- Pin type
 - input/output/inout
 - function
 - data/clock
 - capacitance
- Path delay/transition
- Internal power
- Timing constraint
 - setup, hold, mpwh, mpwl, recovery, removal ...



Gate-level Netlist

- Designing a chip , **IO pads** should be added before the netlist is imported.

```

module CHIP (CLK, HALT, RESET_, DoDCT, X, Z, Mode, SCAN_IN, SCAN_OUT, SCAN_EN);
input CLK, HALT, RESET_, DoDCT;
input [11:0] X;
input Mode;
input SCAN_IN, SCAN_EN;
output SCAN_OUT;
output [11:0] Z;

wire i_CLK, i_HALT, i_RESET_, i_DoDCT;
wire [11:0] i_X;
wire i_Mode;
wire [11:0] i_Z;
wire i_SCAN_IN, i_SCAN_EN;
wire i_SCAN_OUT;

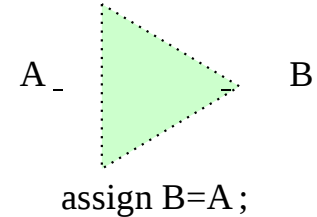
DCT DCT(.CLK(i_CLK), .HALT(i_HALT), .RESET_(i_RESET_), .DoDCT(i_DoDCT), .X(i_X), .Z(i_Z), .Mode(i_Mode), .test_si(i_SCAN_IN), .test_so(i_SCAN_OUT), .test_se(i_SCAN_EN));

PDIDGZ ipad_CLK    (.PAD(CLK), .C(i_CLK));
PDIDGZ ipad_HALT    (.PAD(HALT), .C(i_HALT));
PDIDGZ ipad_RESET_  (.PAD(RESET_), .C(i_RESET_));
PDIDGZ ipad_DoDCT   (.PAD(DoDCT), .C(i_DoDCT));
PDIDGZ ipad_Mode    (.PAD(Mode), .C(i_Mode));
PDIDGZ ipad_X0      (.PAD(X[0]), .C(i_X[0]));
PDIDGZ ipad_X1      (.PAD(X[1]), .C(i_X[1]));
PDIDGZ ipad_X2      (.PAD(X[2]), .C(i_X[2]));
PDIDGZ ipad_X3      (.PAD(X[3]), .C(i_X[3]));
PDIDGZ ipad_X4      (.PAD(X[4]), .C(i_X[4]));
PDIDGZ ipad_X5      (.PAD(X[5]), .C(i_X[5]));
PDIDGZ ipad_X6      (.PAD(X[6]), .C(i_X[6]));
PDIDGZ ipad_X7      (.PAD(X[7]), .C(i_X[7]));
PDIDGZ ipad_X8      (.PAD(X[8]), .C(i_X[8]));
PDIDGZ ipad_X9      (.PAD(X[9]), .C(i_X[9]));
PDIDGZ ipad_X10     (.PAD(X[10]), .C(i_X[10]));
PDIDGZ ipad_X11     (.PAD(X[11]), .C(i_X[11]));
PDIDGZ ipad_SCAN_IN (.PAD(SCAN_IN), .C(i_SCAN_IN));
PDIDGZ ipad_SCAN_EN (.PAD(SCAN_EN), .C(i_SCAN_EN));

```

Gate-level Netlist

- Remove “**assign**” statement before APR.
 - The assign statement can be removed in Encounter
Encounter> **setDoAssign -buffer buf_name on**
- Make sure that there is **no** “***cell***” net name in the netlist.
 - Use the synthesis commands (DC) below to remove “*cell*” cell name
dc_shell> **define_name_rules name_rule –map {{*cell* cell}}**
dc_shell> **change_names –hierarchy –rules name_rule**
- Ensure the names of all instantiated cell types are unique
unix> **uniquifyNetlist –top TOP output_netlist input_netlist**



SDC Constraint basic

create_clock

set_clock_latency

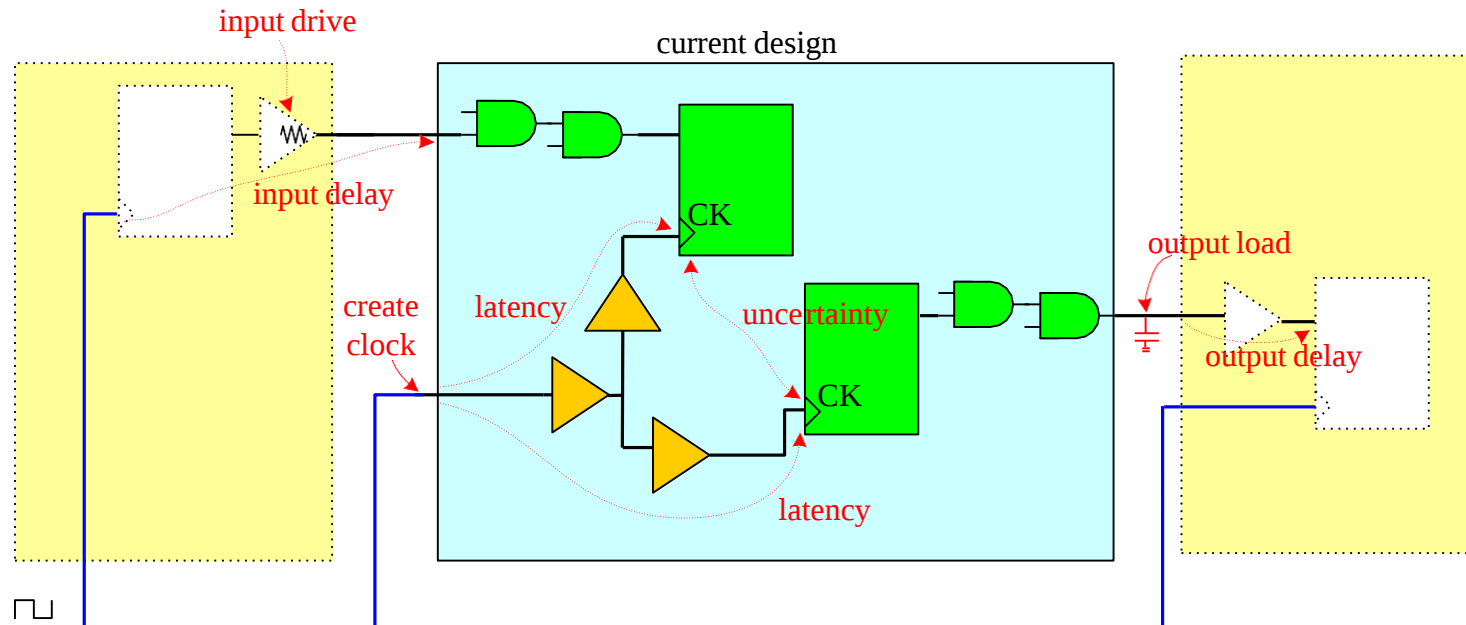
set_clock_uncertainty

set_input_delay

set_output_delay

set_drive

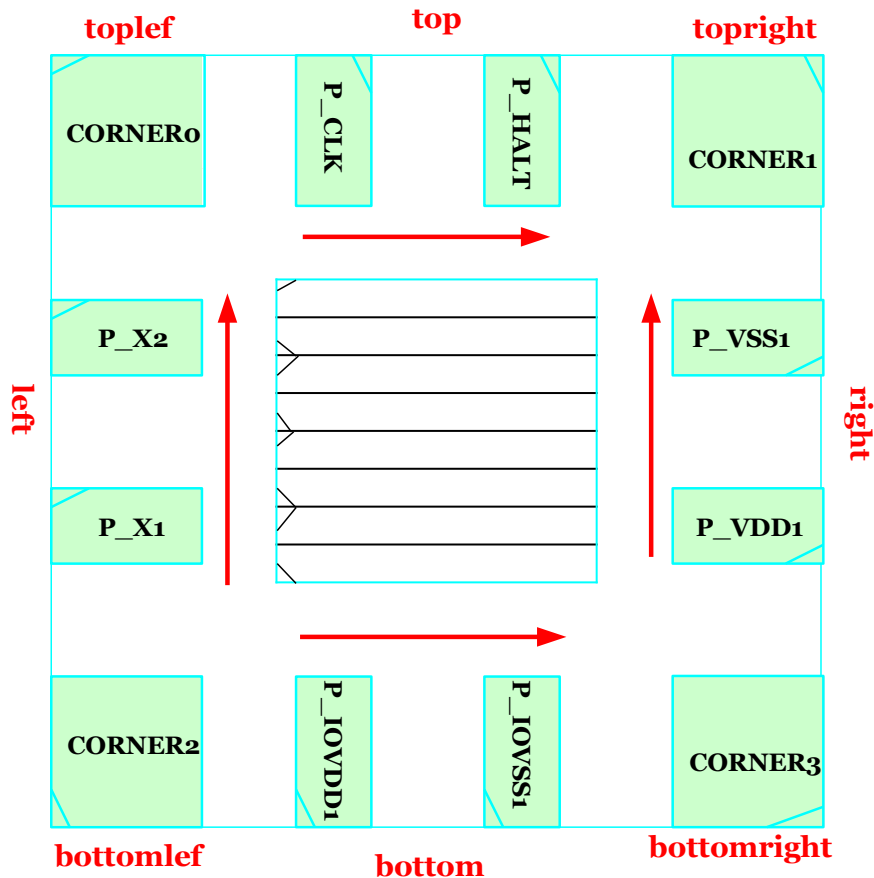
set_load



Basic Sdc File

```
create_clock [get_ports {CLK}] -name CLK -period 8 -waveform {0 4}
set_clock_latency 2 [get_clocks {CLK}]
set_clock_uncertainty 1 [get_clocks {CLK}]
set_input_delay 2 [remove_from_collection [all_inputs] [get_ports CLK]]
set_output_delay 2 -clock CLK [all_outputs]
set_drive 0.1 [all_inputs]
set_load -pin_load 20 [all_outputs]
```

IO Constraint



```
(globals
  version = 3
  io_order = default
)
(iopad
  (top
    (inst name="P_CLK")
    (inst name="P_HALT")
  )
  (right
    (inst name = "P_VDD1" cell="PVDD1DGZ")
    (inst name = "P_VSS1" cell="PVSS1DGZ")
  )
  (left
    (inst name="P_X1")
    (inst name="P_X2")
  )
  (bottom
    (inst name="P_IOVDD1" cell="PVDD2DGZ")
    (inst name="P_IOVSS1" cell="PVSS2DGZ")
  )
  (topright
    (inst name="CORNER0" cell="PCORNER")
  )
  (topright/topleft/bottomright/bottomleft
    .....
  )
)
```

IO Constraint version2

Version: 2

Pad: CORNER0 NW PCORNER

Pad: PAD_CLK N

Pad: PAD_HALT N

Pad: CORNER1 NE PCORNER

Pad: PAD_X1 W

Pad: PAD_X2 W

Pad: CORNER2 SW PCORNER

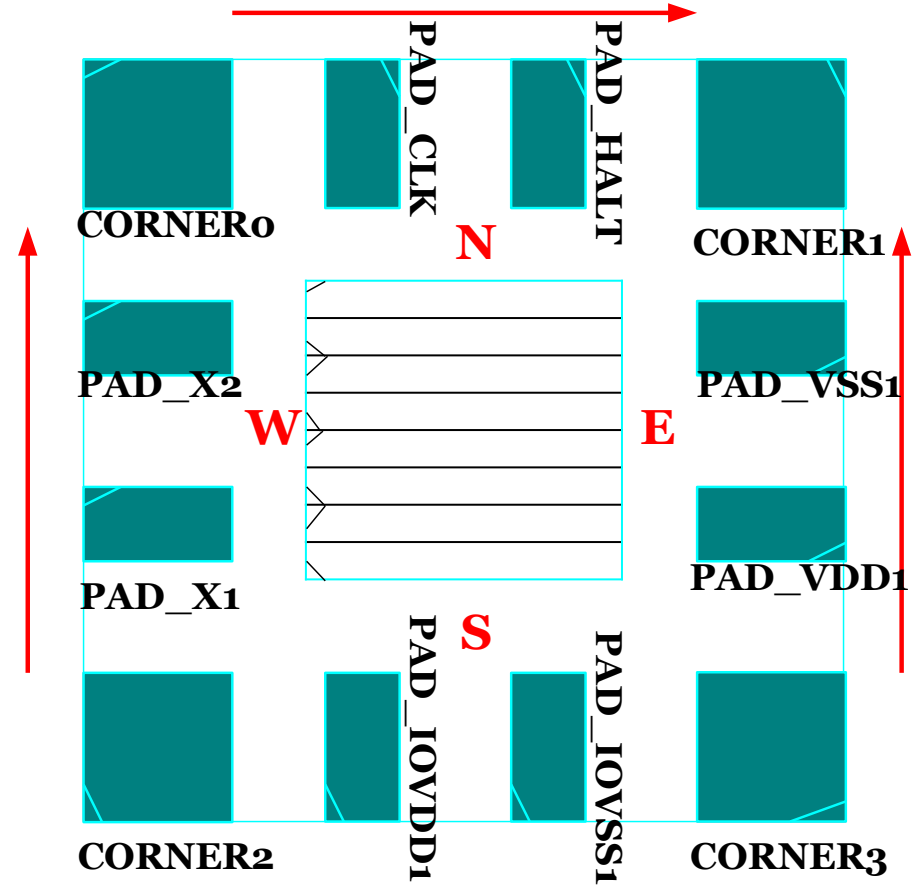
Pad: PAD_IOVDD1 S PVDD2DGZ

Pad: PAD_IOVSS1 S PVSS2DGZ

Pad: CORNER3 SE PCORNER

Pad: PAD_VDD1 E PVDD1DGZ

Pad: PAD_VSS1 E PVSS2DGZ



Power Pad Issues

- SSO
 - Simultaneously Switch Outputs
- DI
 - Maximum number of copies of an I/O cell switching from high to low simultaneously without making the voltage on the quiet output “0” higher than a threshold value “ V_{il} ” when a single ground cell is applied.
- DF
 - Drive Factor, $DF = 1/DI$
- SDF
 - Sum of Drive Factor

Ground pad	Output pad	Ground bounce
1	DI	$< V_{il}$
DF	1	$< V_{il}$

Power Pad Issues cont.

- Parameter of DF
 - Operating condition
 - Package inductance
 - Slew-rate control IO
 - IO type with different drivestrength
- In SSO case
 - Required number of ground pads = SDF
 - Required number of power pads = $SDF/1.1$
- Non SSO case (suggest)
 - Required number of ground pads = $SDF/1.5$
 - Required number of power pads = $SDF/1.6$

Power Pad Issues Example

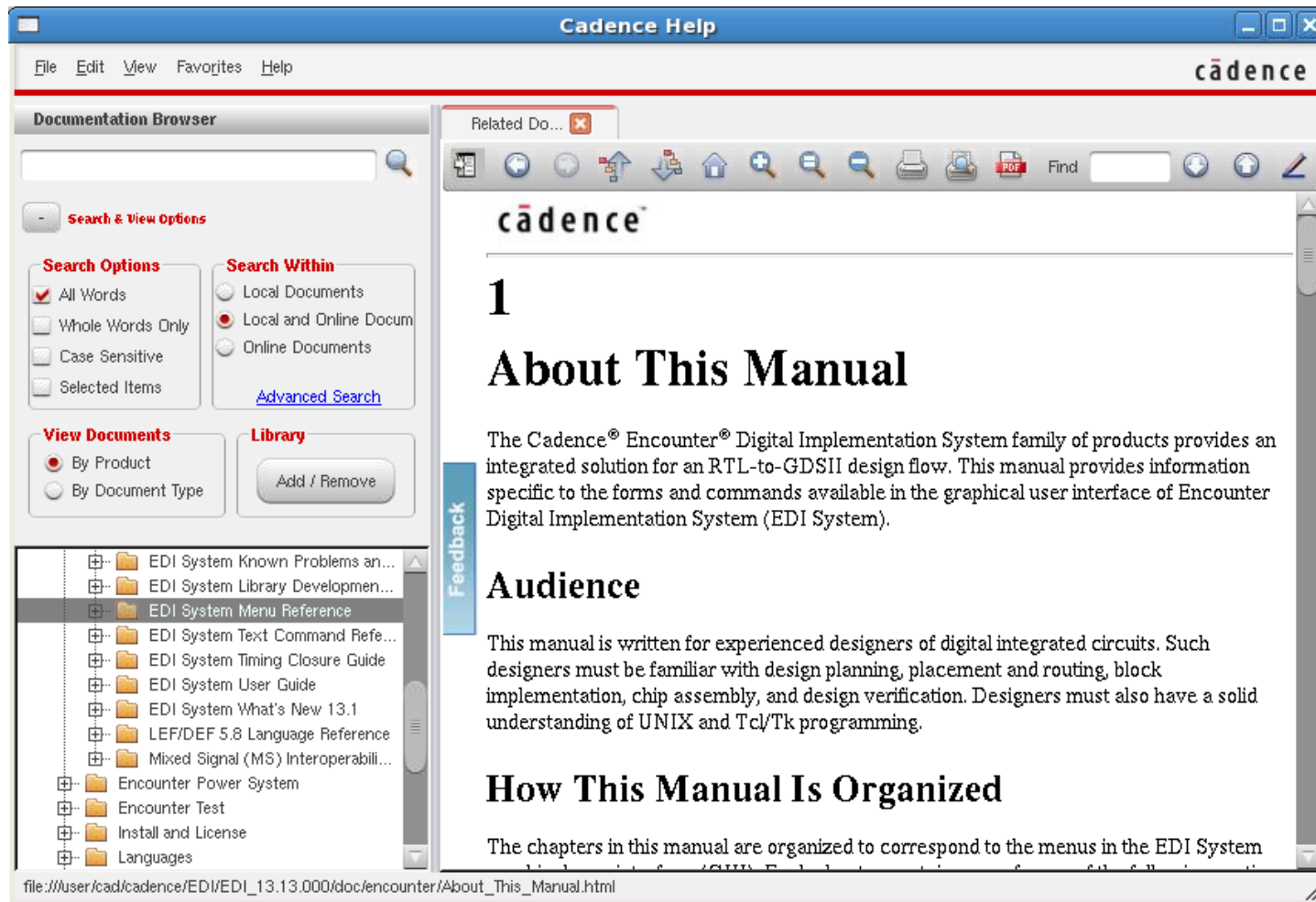
IO Type	2mA	4mA	8mA	12mA	16mA	24mA
DF Value	0.02	0.03	0.09	0.18	0.3	0.56

Drive Factor

- If a design has 20 PDB02DGZ(2mA), 10 PDD16DGZ(16mA). then
- $SDF = 20 \times 0.02 + 10 \times 0.3 = 3.4$
- In SSOcase,
 - number of VSSpad = 3.4 → 4
 - number of VDD pad = $3.4/1.1 = 3.09 \rightarrow 4$

Cadence On-Line Document : cdnshelp

/usr/cad/cadence/EDI/cur/tools/bin/cdnshelp



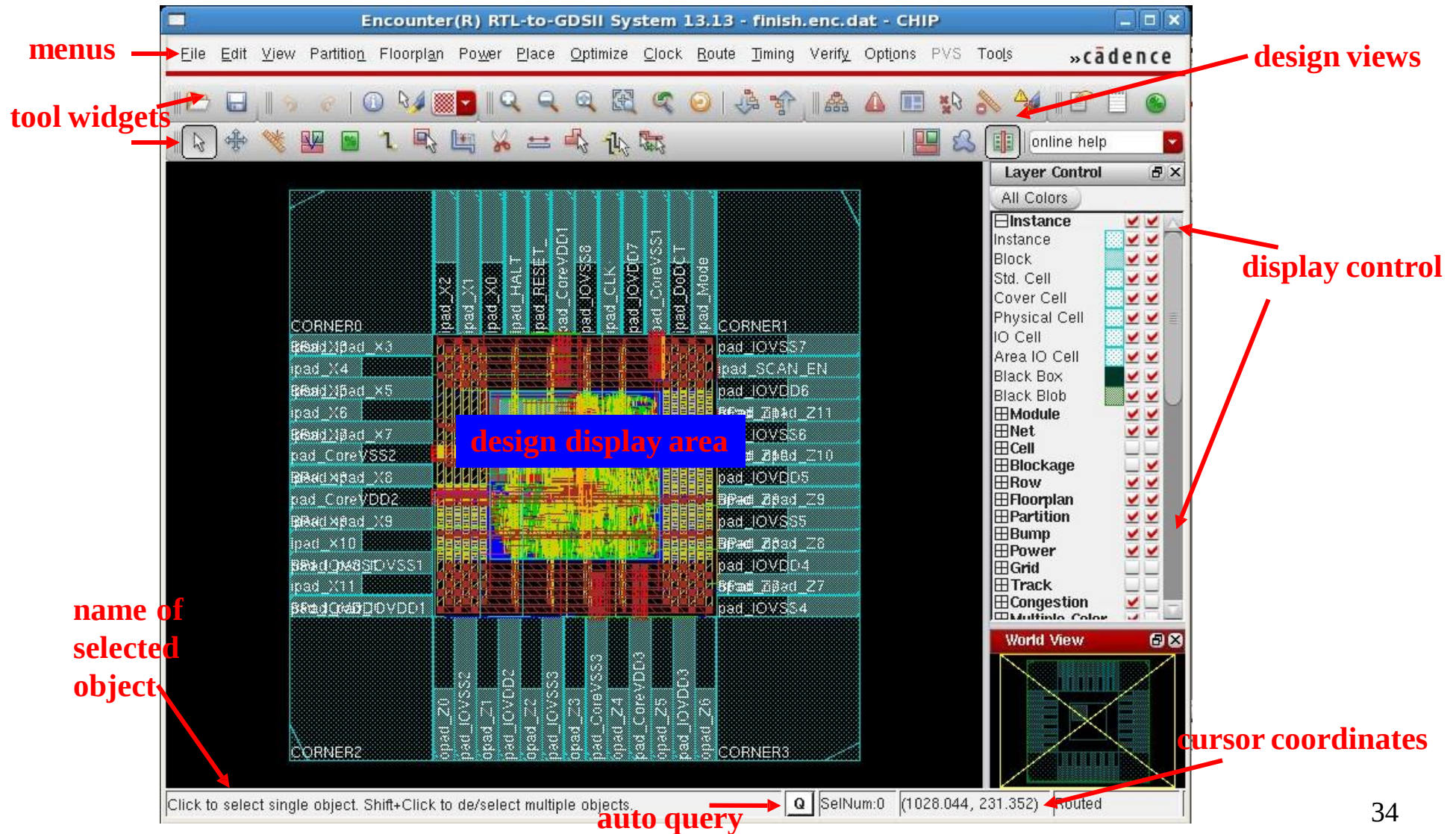


INTERFACE

Getting Started

- Source the encounter environment:
~~unix% source /usr/cad/cadence/CI/edi.cshrc~~
unix% **source /usr/cad/cadence/CI/innovus.cshrc**
- Invoke soc encounter :
~~unix% encounter~~
unix% **innovus**
- Do **not** run in background mode. Because the terminal become the interface of command input while running socencounter.
- Log file:
 - innovus.log*
 - innovus.cmd*

GUI



Display Control

The image displays four panels of PCB display control options, each with a list of items and corresponding checkboxes and color swatches.

Panel 1 (Left):

- All Colors** (button)
- Instance** (checkbox): Instance, Block, Std. Cell, Cover Cell, Physical Cell, IO Cell, Area IO Cell, Black Box, Black Blob.
- Module** (checkbox): Module, Guide, Fence, Region, Partition.
- Net** (checkbox): Net, Special Net, P/G, Metal Fill.
- Cell** (checkbox): Pin Shapes, Cell Blockage, Cell Layout.
- Blockage** (checkbox): Obstruct, Area Density, Macro Blkg, Routing Blkg, Block Halo, Routing Halo, Blockage Link.

Panel 2 (Middle):

- Row** (checkbox): Standard Row, IO Row, Row Pattern, Macro Pattern.
- Floorplan** (checkbox): Rel. FPlan, SDP Group, SDP Connect, SizeBlkg, S. Resize Line, E. Resize Line, MP CongTag, IO Cluster, Overlap Macro, Overlap Guide, Overlap Block, Datapath.
- Partition** (checkbox): Pin Guide, Ptn Pin Blk, Ptn Feedthru.
- Bump** (checkbox): Bump(Normal), Bump(Back), Bump Connect.
- Power** (checkbox): Power Domain, Power Graph, Sub. Noise, IR Drop & EM.

Panel 3 (Right):

- Grid** (checkbox): Manufacture, Placement, User-defined, GCell.
- Track** (checkbox): Pref Track, NPref Track.
- Congestion** (checkbox): Cong. Label, Congestion, GC Overflow, Channel Cong., H. Congest, V. Congest.
- Multiple Color** (checkbox): Density Map, Clock Tree, Thermal, MP Checker, FL Congest, GTD Object, Double Pattern.
- Miscellaneous** (checkbox): Terminal, Violation, Bus Guide, Select, Text, Pin Text, Channel, Flight Line, REDUCED, Port Number, Grid Resistor.

Panel 4 (Far Right):

- Wire&Via** (checkbox): active(M0), Via 01, metal1(M1), via1(V12), metal2(M2), via2(V23), metal3(M3), via3(V34), metal4(M4), via4(V45), metal5(M5), via5(V56), metal6(M6), via6(V67), metal7(M7), via7(V78), metal8(M8), via8(V89), metal9(M9), via9(V910), metal10(M10).

Display Select

Common Used Bindkeys

Key	Action
q	Edit attribute
f	Fits display
z	Zoom in
Z	Zoom out
Arrows	pans design area in the direction of the arrow
Escape	Cancel
K	Removes all rulers

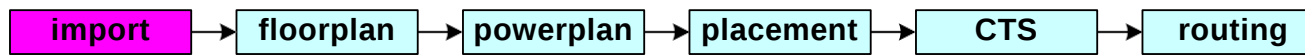
Key	Action
space	Select Next
e	popup Edit
T	editTrim
0-9	toggle layer[0-9] visibility
h/H	hierarchy up/down
x	clear Drc
N	next via

Looking for more bindkey:

Options → Set Preference, Binding Key



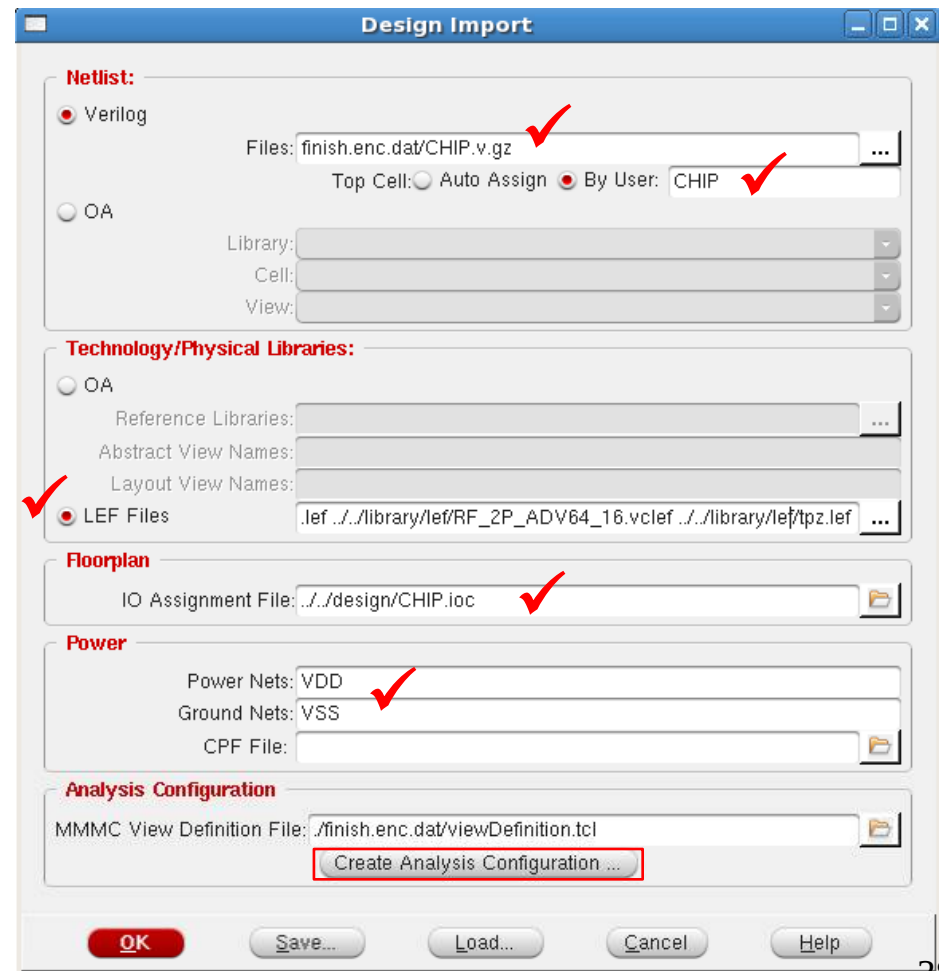
APR FLOW



Import Design

File → Design Import...

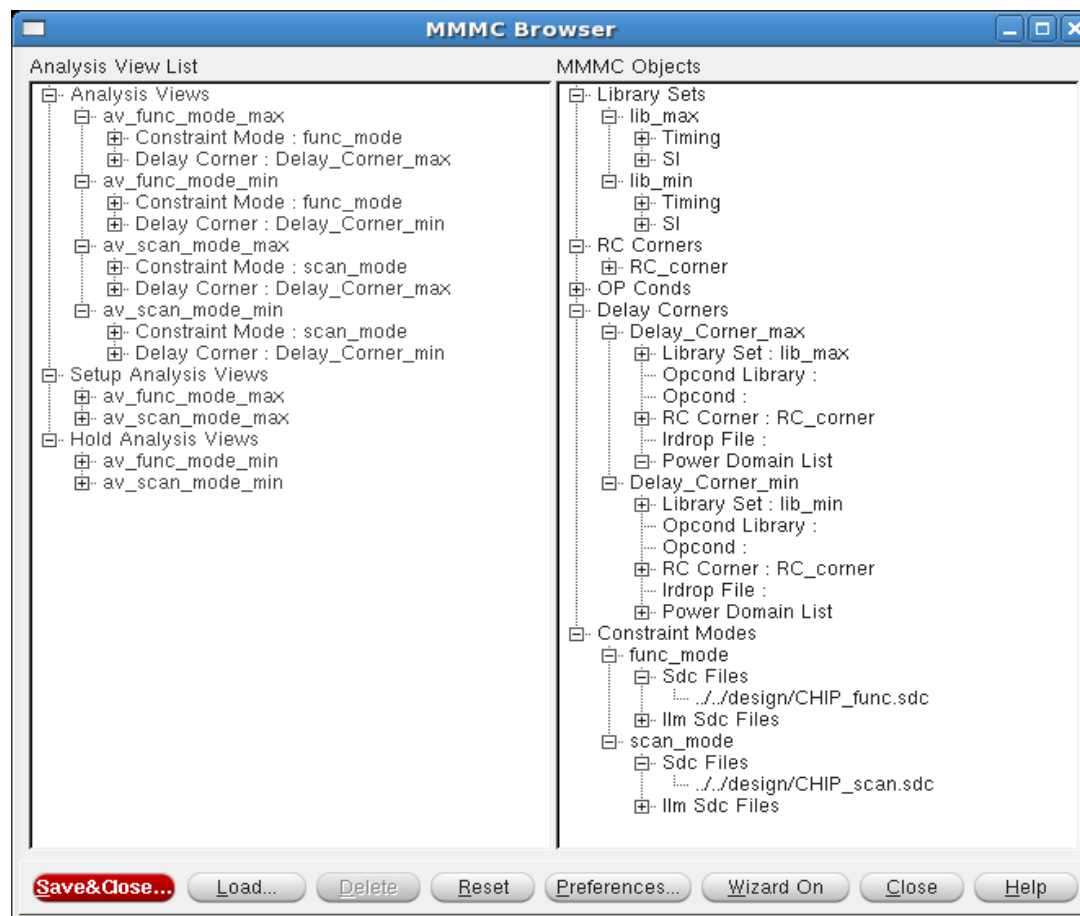
- ◆ Import LEF in the order:
 - technology first
 - geometry lef for cell/block
 - antenna lef for cell/block
- ◆ IO Assignment File:
 - get a IO assignment template:
Design → Save → I/O File...



MMMC Browser

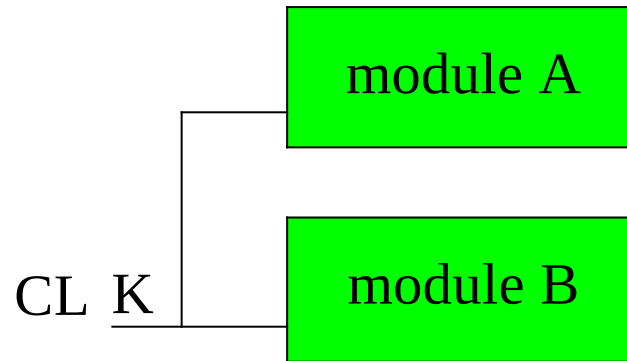
File → Design Import

Create Analysis Configuration ...



Why MMMC

Case1



Operation Mode1 : moduleA runs on 100MHz
moduleB not use

Operation Mode2 : moduleA runs on 50MHz
moduleB runs on 50MHz

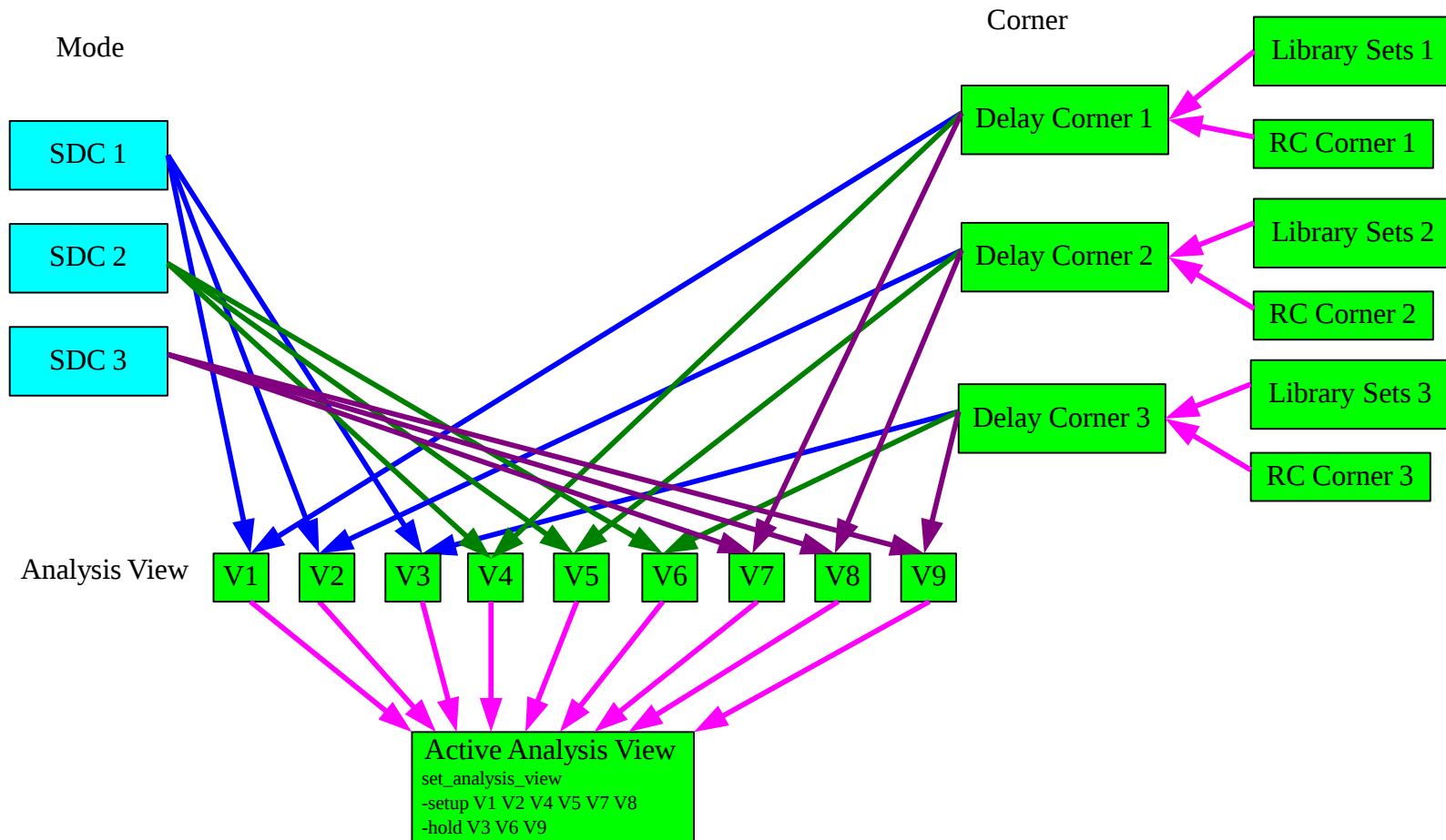
Why MMMC

Case2

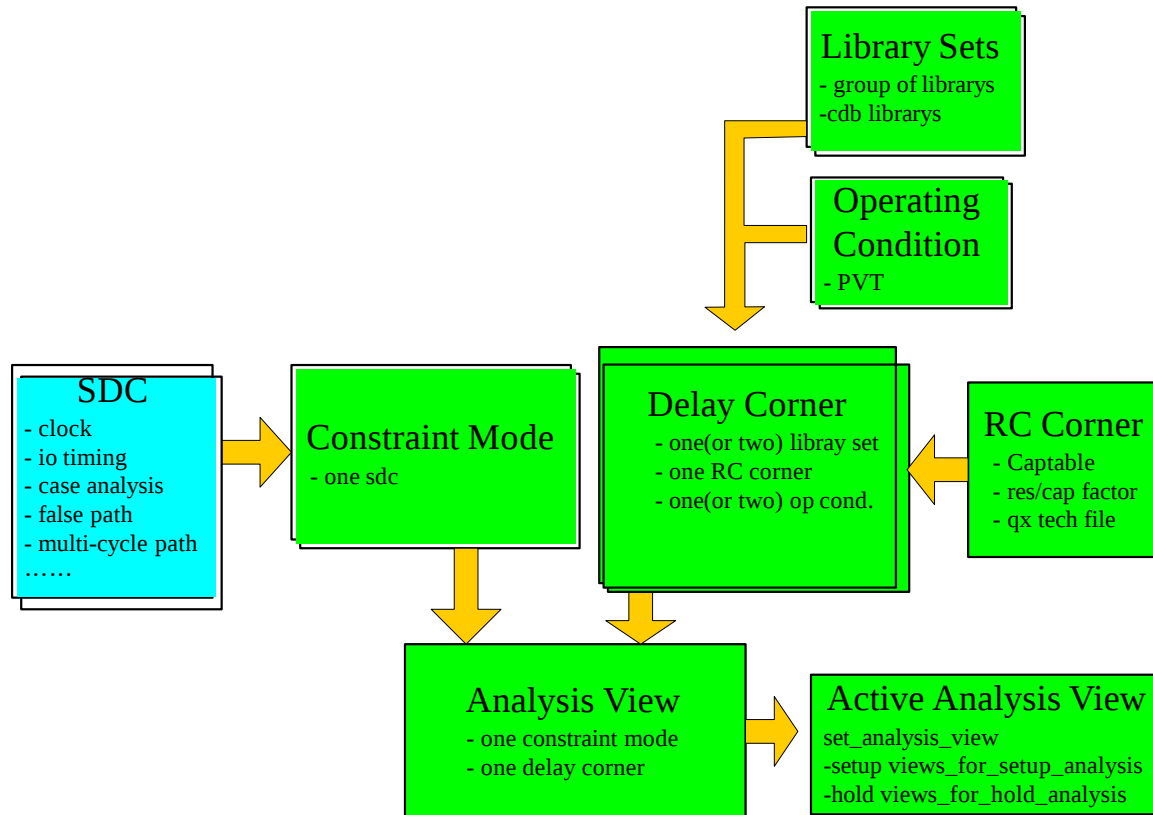
- The design is required to meet 3 operating corner
 - Corner1 : 1.1V , 0°C
 - Corner2 : 0.9V , 100°C
 - Corner3 : 1.1V , 100°C

Multi-Mode Multi Corner **NAR Labs**

expand view



Multi-Mode Multi Corner



MMMC Example

Library Set :

lib_max	Timing	../..../library/lib/NangateOpenCellLibrary_slow.lib ../..../library/lib/RF_2P_ADV64_16_ss_0.9_125.0_syn.lib ../..../library/lib/tpz_slow.lib
	SI	../..../library/celtic/slow.cdb

lib_min	Timing	../..../library/lib/NangateOpenCellLibrary_fast.lib ../..../library/lib/RF_2P_ADV64_16_ff_1.1_-40.0_syn.lib ../..../library/lib/tpz_fast.lib
	SI	../..../library/celtic/fast.cdb

lib_typ	Timing	../..../library/lib/NangateOpenCellLibrary_typical.lib ../..../library/lib/RF_2P_ADV64_16_tt_1.0_25.0_syn.lib ../..../library/lib/tpz_typ.lib
	SI	../..../library/celtic/typical.cdb

Analysis_Views :

AV_func_max	constraint mode	CM_func
	delay corner	DC_max

AV_func_min	constraint mode	CM_func
	delay corner	DC_min

AV_func_typ	constraint mode	CM_func
	delay corner	DC_typ

RC Corners :

RC_worst	Cap Table	../..../library/fireice/worst.captbl
	qx tech file	../..../library/fireice/worst.tch

RC_best	Cap Table	../..../library/fireice/best.captbl
	qx tech file	../..../library/fireice/best.tch

RC_typ	Cap Table	../..../library/fireice/typical.captbl
	qx tech file	../..../library/fireice/typical.tch

Constraint Modes :

CM_func	sdc file	../..../design/CHIP_func.sdc
---------	----------	------------------------------

CM_scan	sdc file	../..../design/CHIP_scan.sdc
---------	----------	------------------------------

AV_scan_max	constraint mode	CM_scan
	delay corner	DC_max

AV_scan_min	constraint mode	CM_scan
	delay corner	DC_min

AV_scan_typ	constraint mode	CM_scan
	delay corner	DC_typ

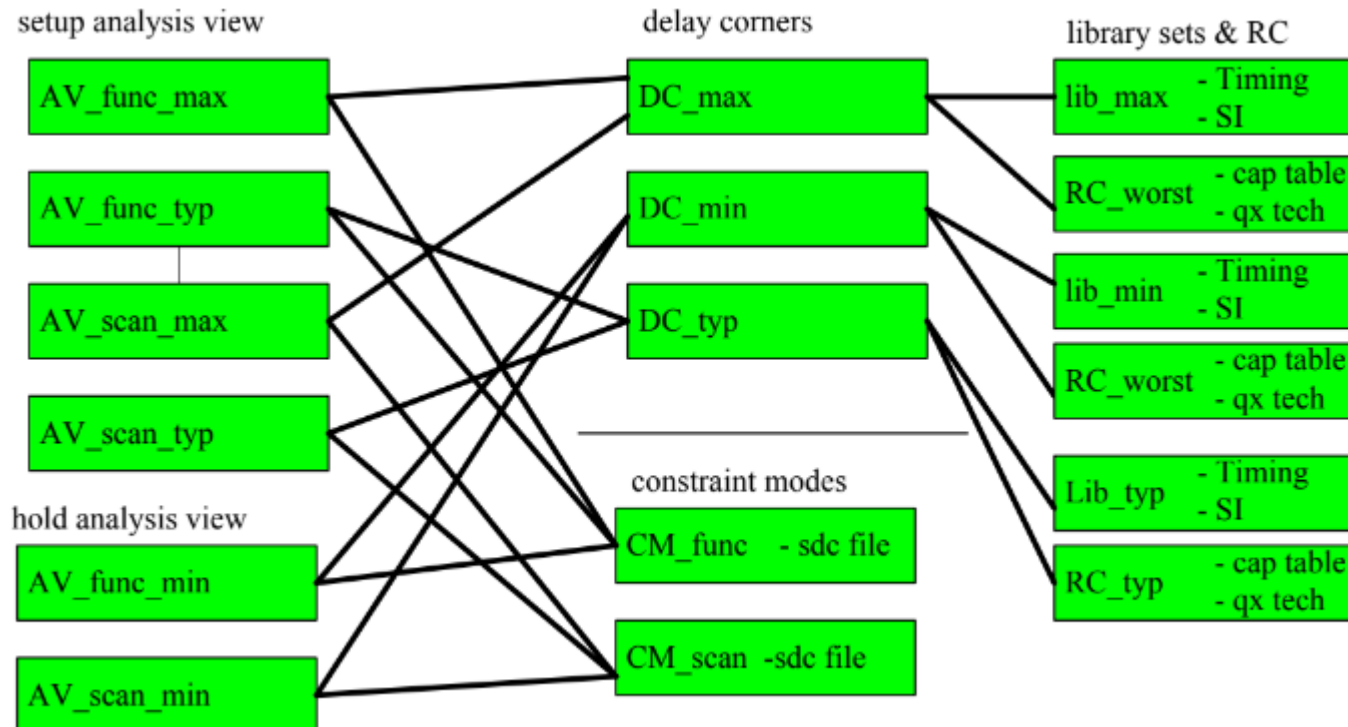
MMMC Example

Setup Analysis Views :

AV_func_max
AV_func_typ
AV_scan_max
AV_scan_typ

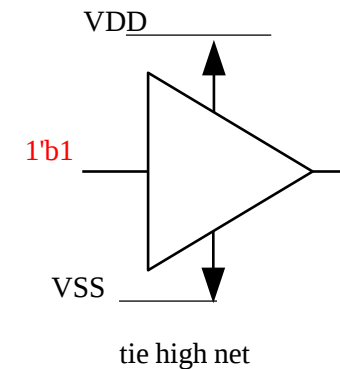
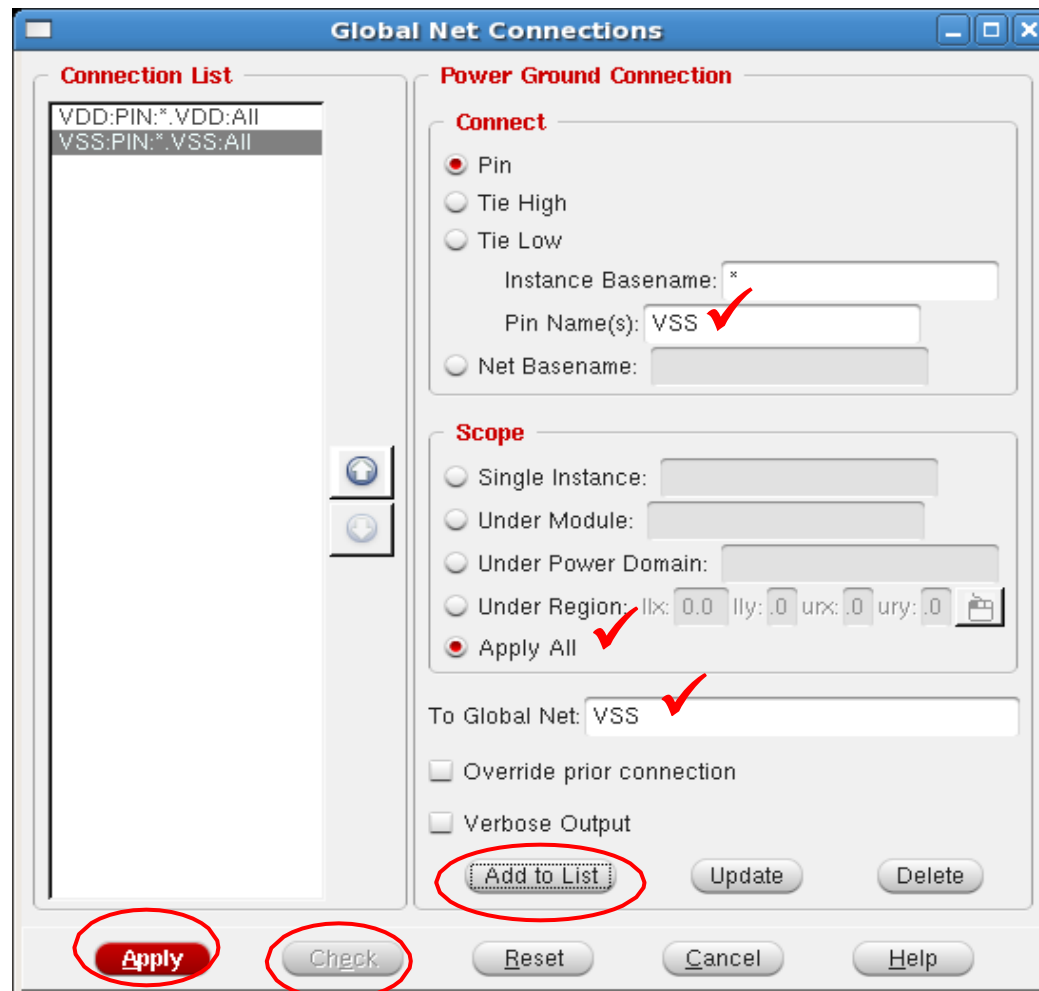
Hold Analysis Views :

AV_func_min
AV_scan_min

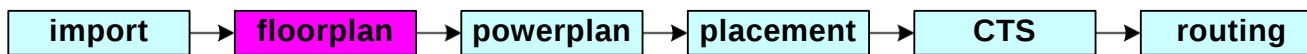


Global Net Connection

Power → Connections Gloval Nets ...

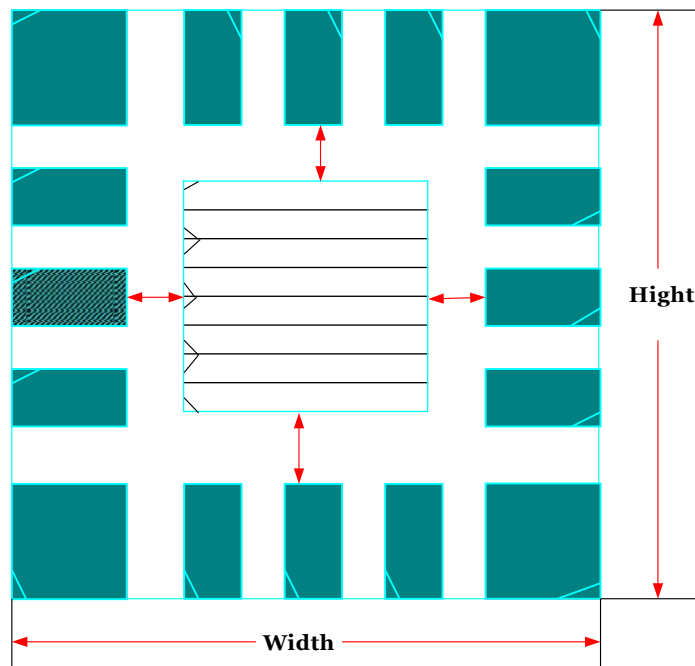


INV inv1(.I(1'b1), .O(o));



Specify Floorplan

Floorplan → Specify Floorplan ...



Specify Floorplan

Basic Advanced

Design Dimensions

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

☒ Core Size by: ☒ Aspect Ratio: Ratio (H/W): ☒ 1

☒ Core Utilization: ☒ 0.7

☐ Cell Utilization: 0.707175

☐ Dimension: Width: 672.12

Height: 670.32

☐ Die Size by: Width: 1042.58

Height: 1040.68

Core Margins by: ☒ Core to IO Boundary

☐ Core to Die Boundary

Core to Left: ☒ 80 Core to Top: ☒ 80

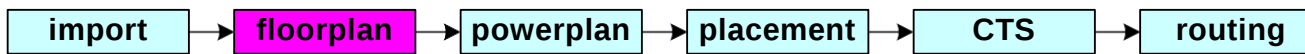
Core to Right: ☒ 80 Core to Bottom: ☒ 80

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

Floorplan Origin at: ☒ Lower Left Corner ☐ Center

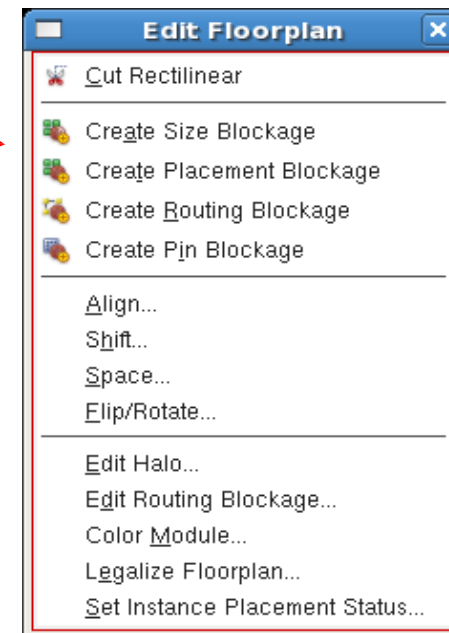
Unit: Micron

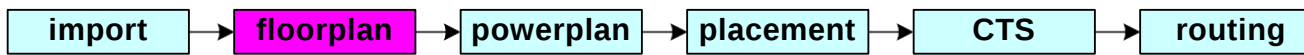
OK Apply Cancel Help



Place Block

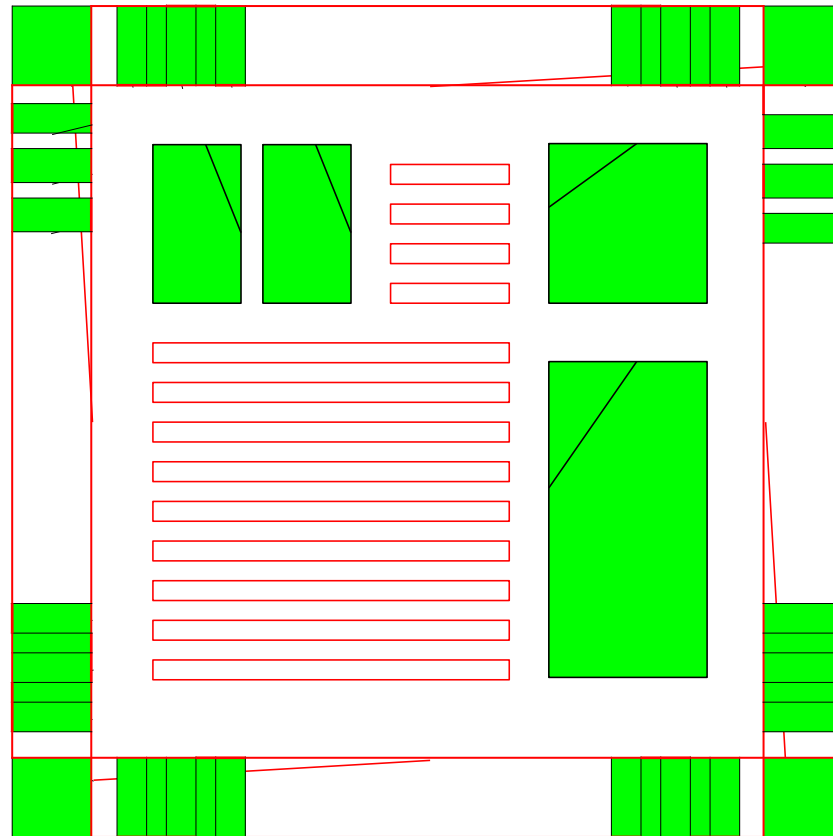
- ◆ Floorplan → Automatic Floorplan → Plan Design...
 - Automatic generate a quick, initial floorplan.
- ◆  Move/Resize/Reshape floorplan object.
- ◆ edit floorplan by functions in :
Floorplan → Edit Floorplan

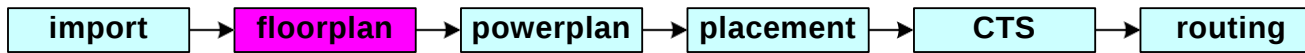




Block Placement

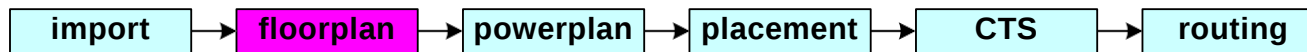
- Block place issue
 - power issue
 - noise issue
 - route issue





Blockage

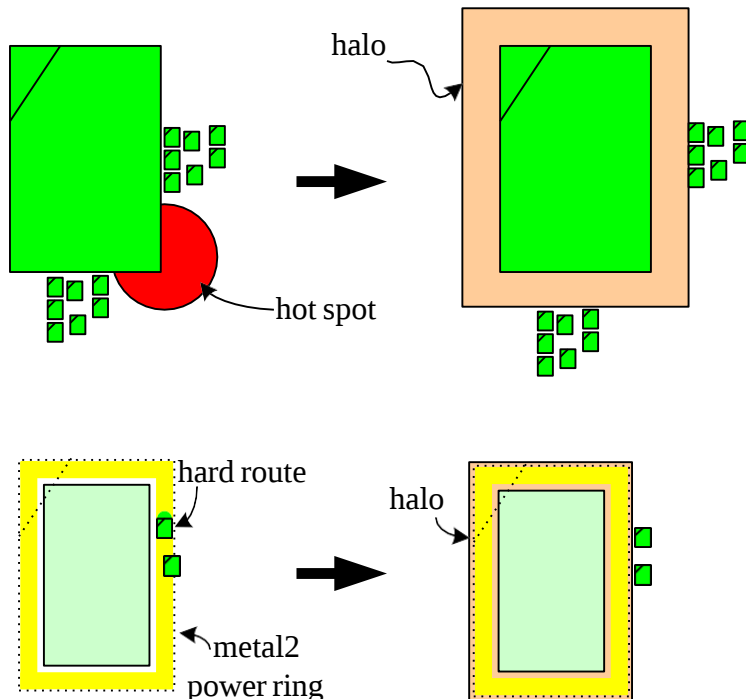
- Placement Blockage 
 - Hard
 - The initial placement should not use the area, but later phases, such as optimization of CTS can use the blockage area.
 - Soft
 - The initial placement should not use more than maxDensity percentage of the blockage area.
- Routing Blockage 
 - Blockage on given routing layers

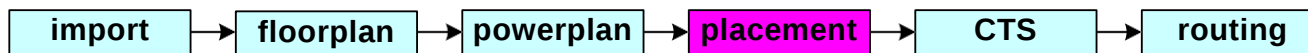


Add Halo to Block

Floorplan → Edit Floorplan → Edit Halos...

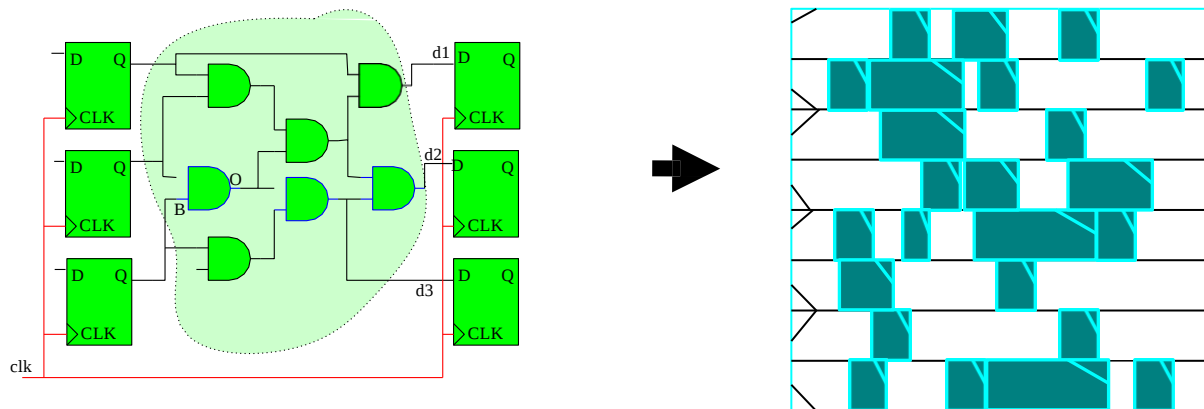
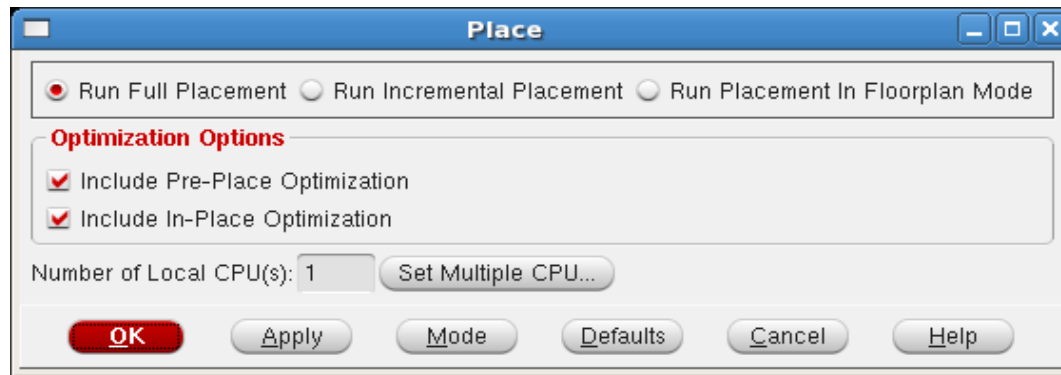
- Prevent the placement of blocks and standard cells in order to reduce congestion around a block.

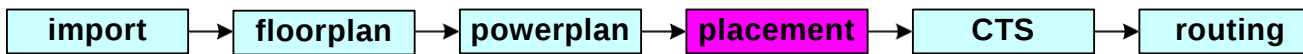




Placement

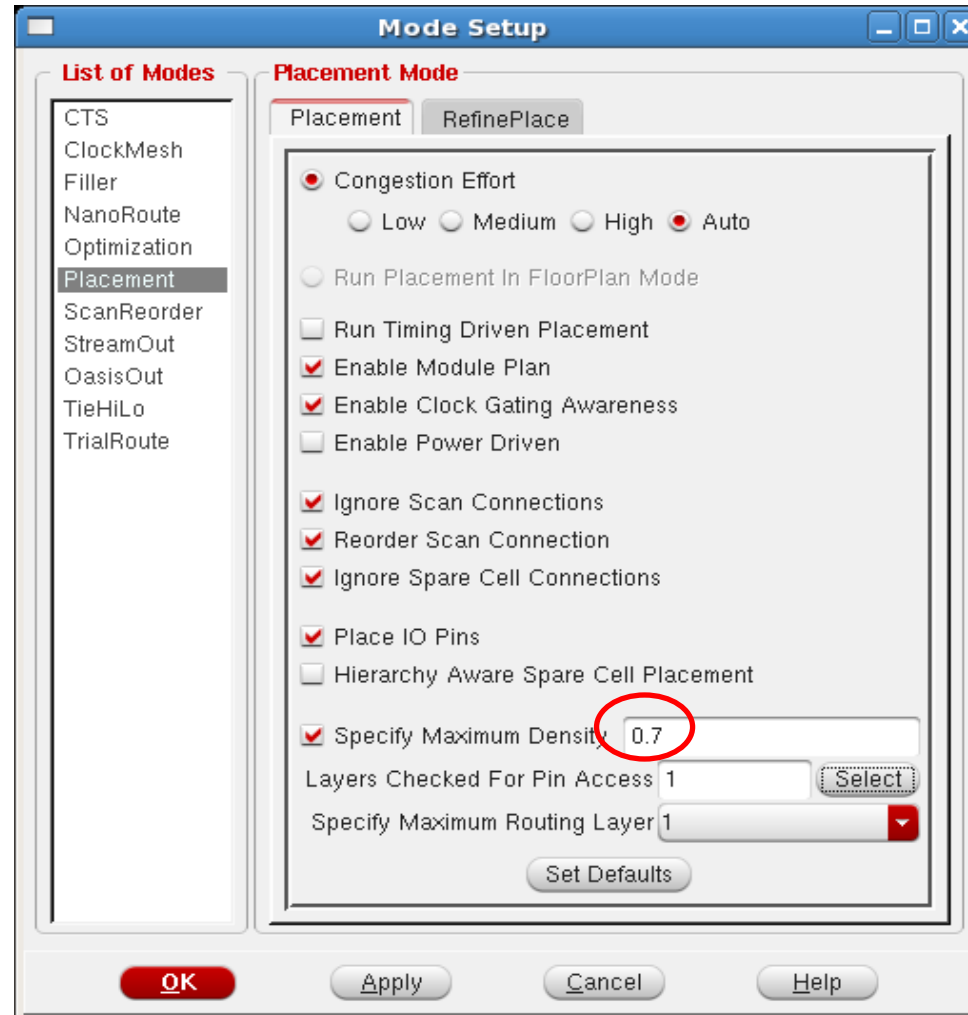
Place → Place Standard Cells ...



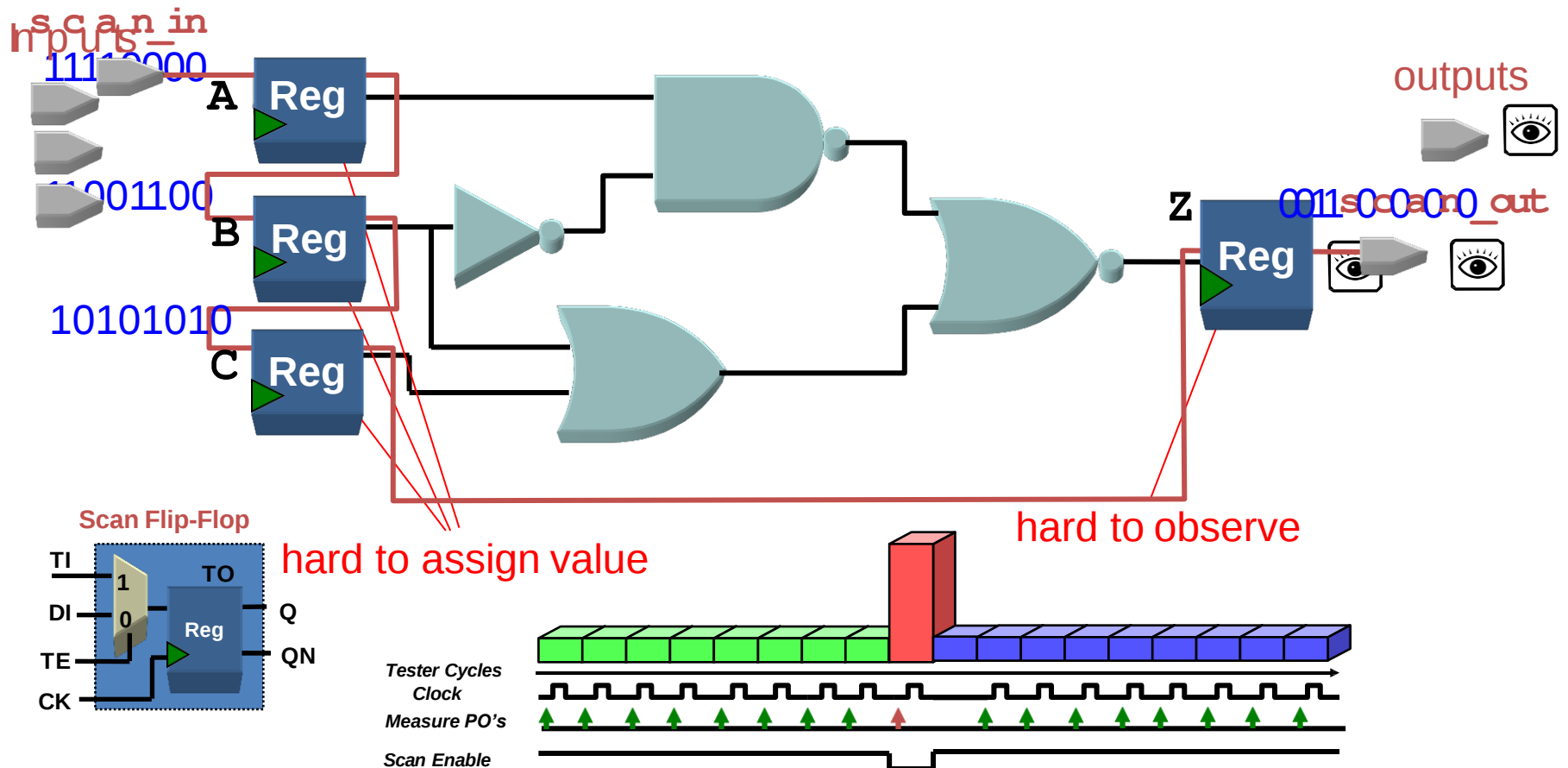


Mode Setup -- Placement

Options → Set Mode → Mode Setup...



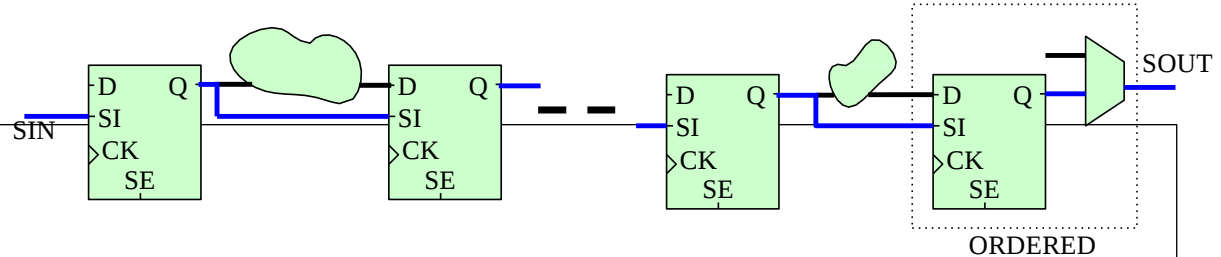
Scan Chain



Specify Scan Chain with scan def

File → Load → DEF...

scan_def



```
SCANCHAINS 1 ;
```

```
- scan1
```

```
+ START SIN
```

```
+ FLOATING
```

```
DCT_tposemem_Bisted_RF_2P_ADV64x16_BistCtrl_i0/S44/State_reg[2] ( IN SI ) ( OUT QN )
```

```
DCT_tposemem_Bisted_RF_2P_ADV64x16_BistCtrl_i0/S44/State_reg[1] ( IN SI ) ( OUT Q )
```

```
DCT_tposemem_Bisted_RF_2P_ADV64x16_BistCtrl_i0/S44/State_reg[3] ( IN SI ) ( OUT Q )
```

```
DCT_tposemem_Bisted_RF_2P_ADV64x16_BistCtrl_i0/S44/State_reg[0] ( IN SI ) ( OUT QN )
```

```
.....
```

```
+ ORDERED
```

```
DCT/tposemem_Bisted_RF2SH64x16_BistCtrl_i0_ST_MAL_i0_S17_reg ( IN SI ) ( OUT QN )
```

```
DFT_shared_out_mux_3 ( IN B ) ( OUT Y )
```

```
+ STOP SOUT
```

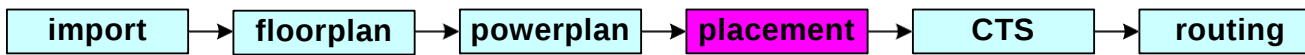
```
;
```

```
END SCANCHAINS
```

```
END DESIGN
```

Generate Scan Def

- Design Vision
 - *write_scan_def -o scan.def*
- RTL compiler
 - *write_scandef > scan.def*



Specify Scan Chain

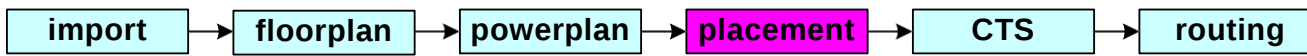
with `specifyScanChain` command

encounter > *`specifyScanChain scanChainName`*

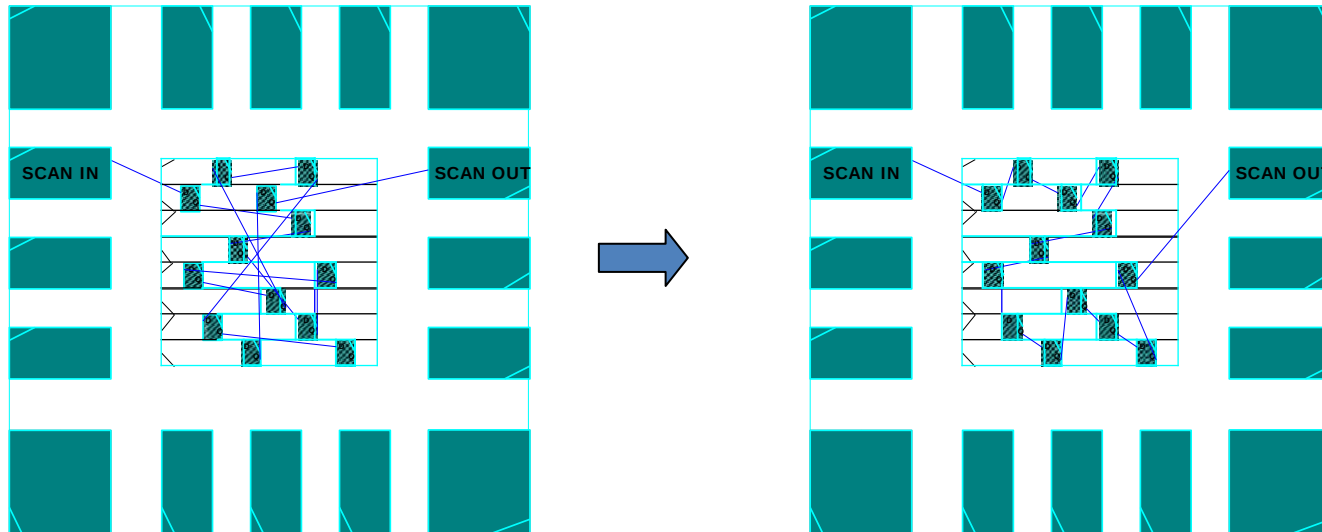
 – *`start {ftname | instPinName}`*

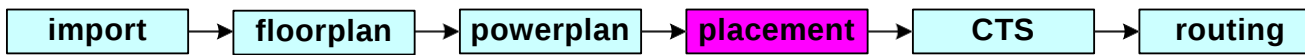
 – *`stop {ftname | instPinName}`*

- `specifyScanChain`
 - `ftname`
 - The design input/output pin name
 - `instPinName`
 - The design instance input/output pin name
- Specifies a scan chain in a design. The actual tracing of the scan chain is performed by the ***scanTrace*** or ***scanReorder*** command
- Enables `scanTrace` trace through multiple input logic gates in scan path
 - *`setScanReorderMode -compLogic`*



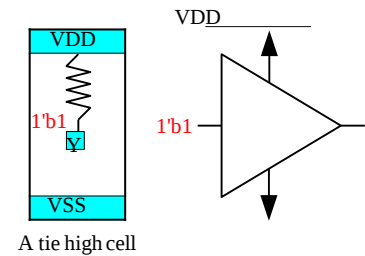
Scan Chain Reorder





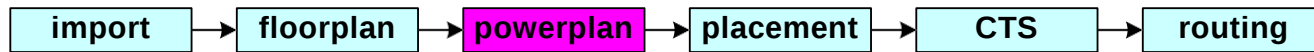
Add Tiehi/Tielo cell

- ◆ Tiehi/Tielo cell connect tiehi/tielo net to supply voltage or ground with resister
- ◆ Tiehi/Tielo cell is added for ESD protection.



◆ ***Place → Tie Hi/Lo Cell → Add***





Power Planning: Add Rings

Power → Power Planning → Add Rings

Add Rings

Basic Advanced Via Generation

Net(s): VDD VSS

Ring Type

- ☒ Core ring(s) contouring
 - ☒ Around core boundary
 - ☐ Along I/O boundary
 - ☐ Exclude selected objects
- ☐ Block ring(s) around

Ring Configuration

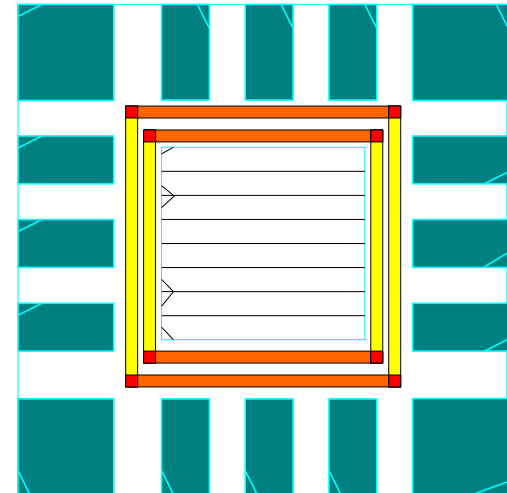
	Top:	Bottom:	Left:	Right:
Layer:	METAL5 H	METAL5 H	METAL4 V	METAL4 V
Width:	8	8	8	8
Spacing:	0.28	0.28	0.28	0.28
Offset:	☐ Center in channel ☒ Specify			
	0.56	0.56	0.56	0.56

Update

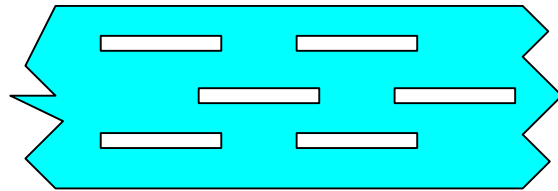
Option Set

☐ Use option set: [dropdown] [icon] Update Basic

OK Variables Apply Defaults Cancel Help

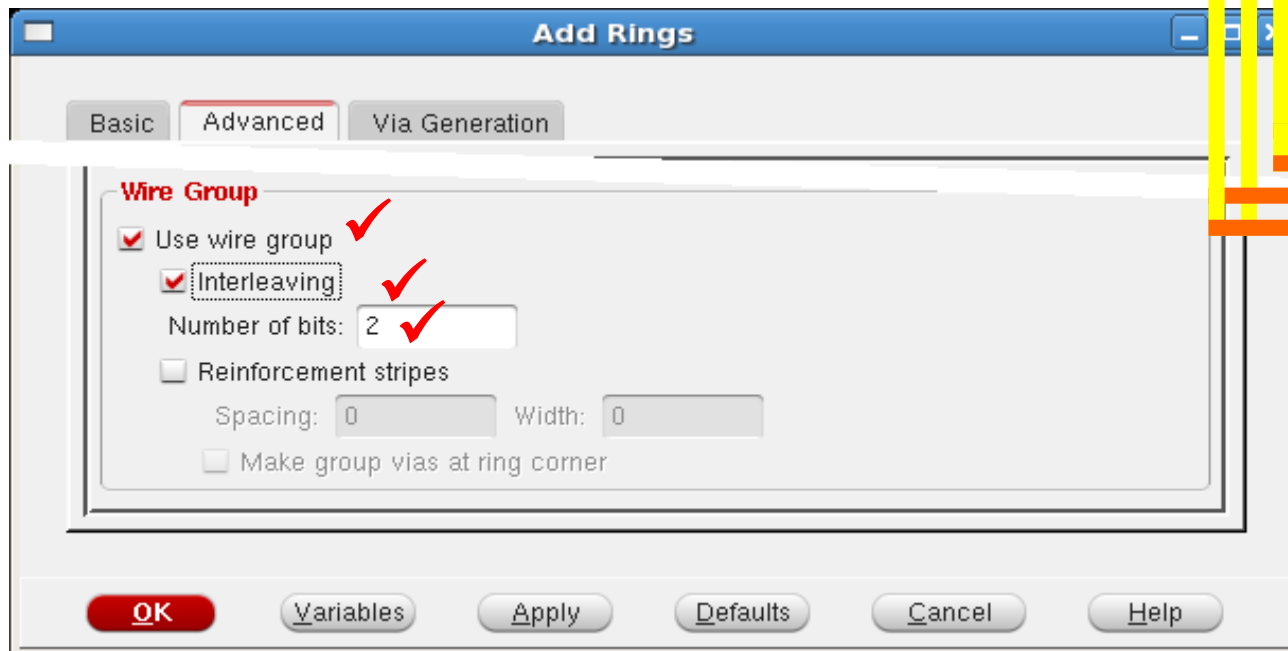
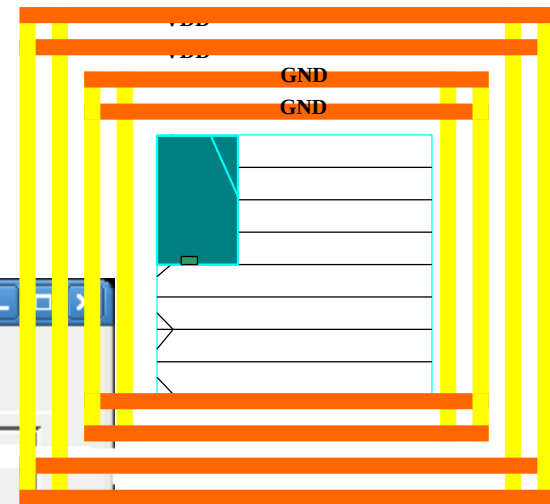


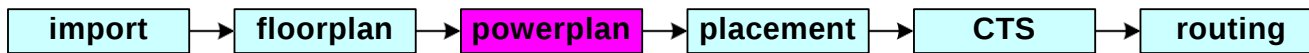
Power Planning: Add Rings



metal slot

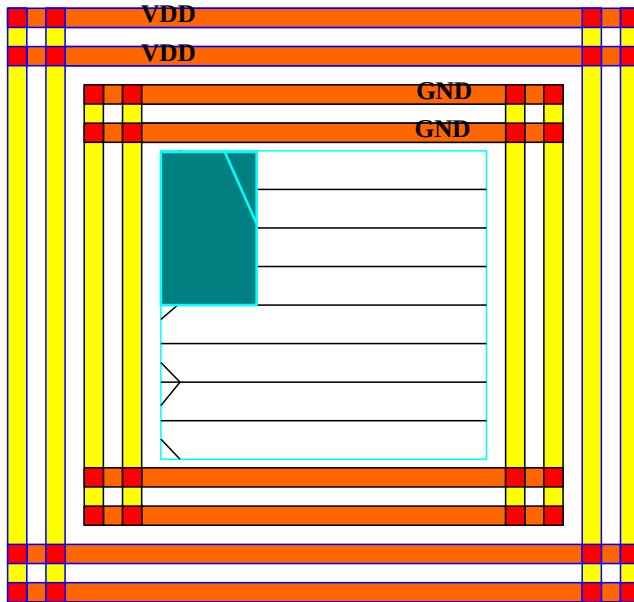
Use wire group to avoid slot DRC error



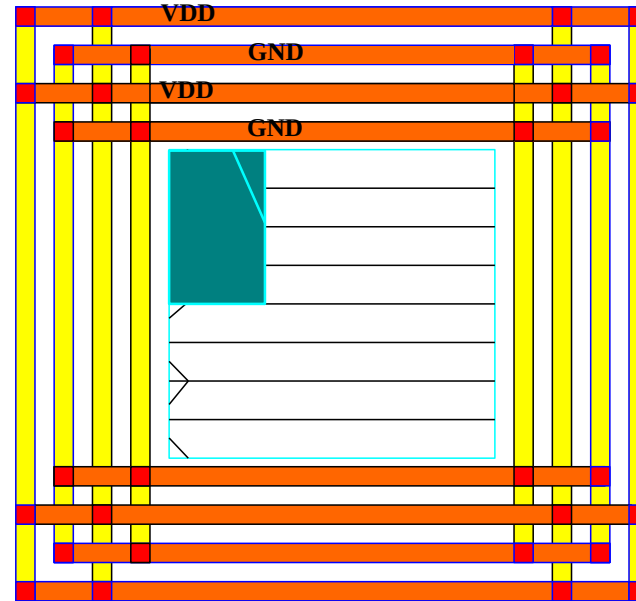


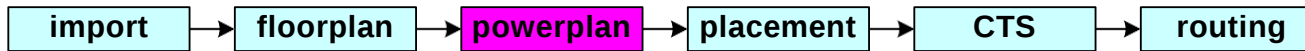
Power Planning: Wire Group

- ✓ Use wire group
- no interleaving
- ✓ number of bits = 2



- ✓ Use wire group
- ✓ interleaving
- ✓ number of bits = 2





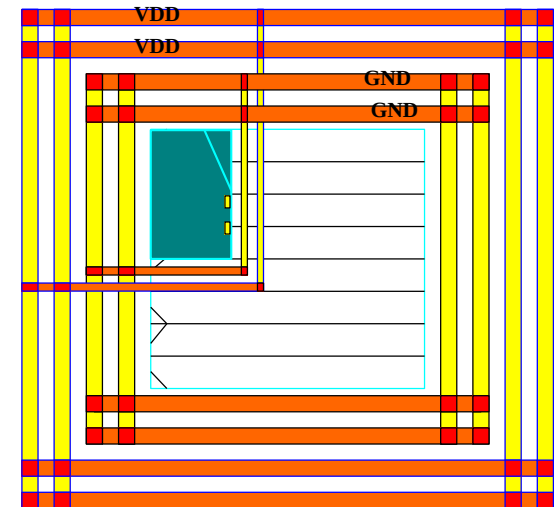
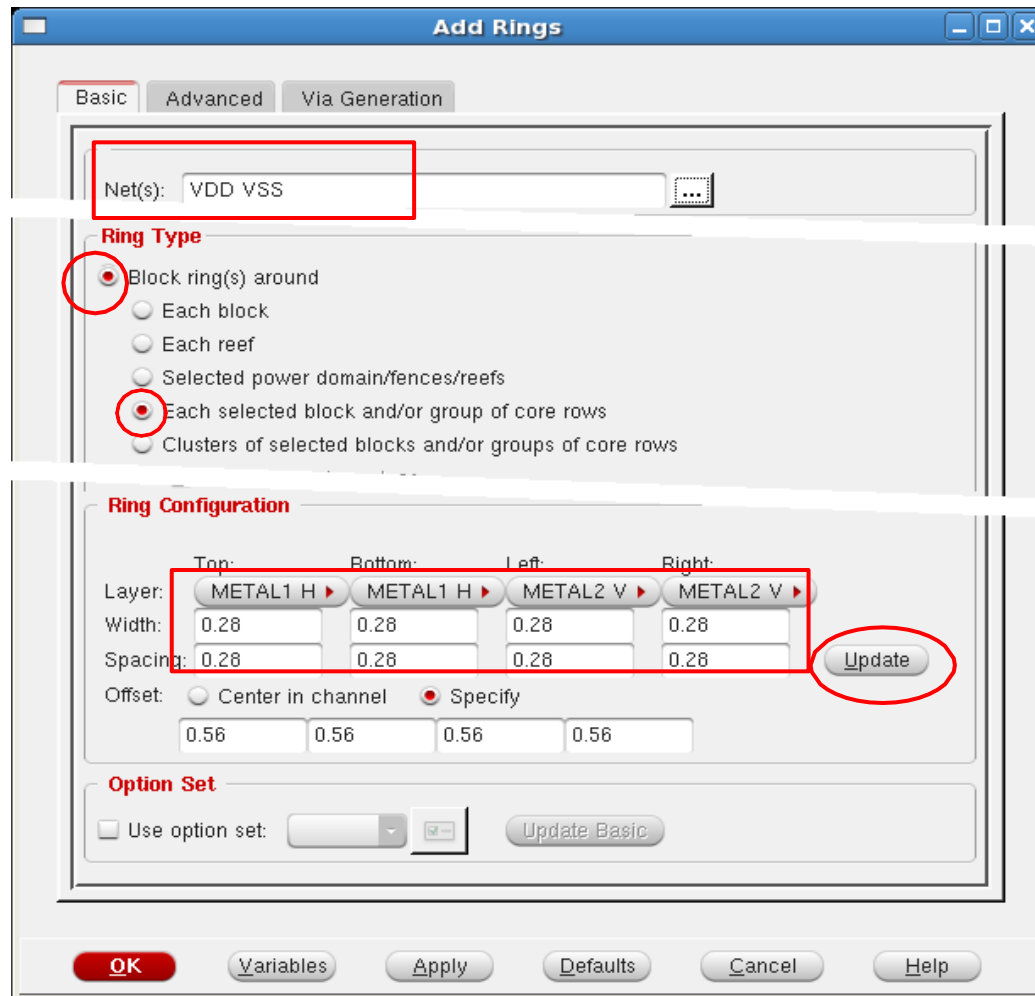
Max Density Rule

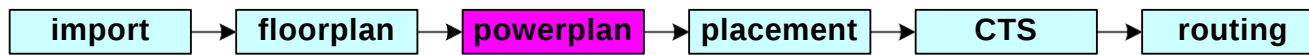
- Max density violation usually happened on power ring
- Max density rule : metal area coverage must < 70%

$$\text{density} = \frac{2x}{2x+x} = 66\%$$

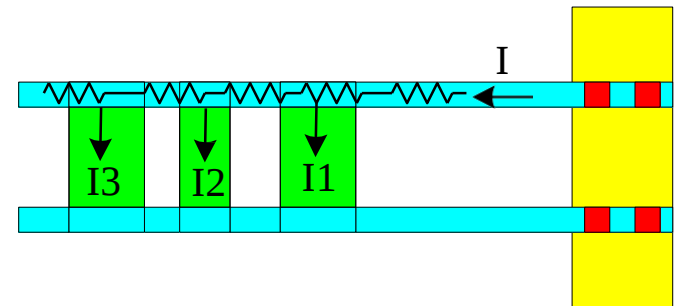
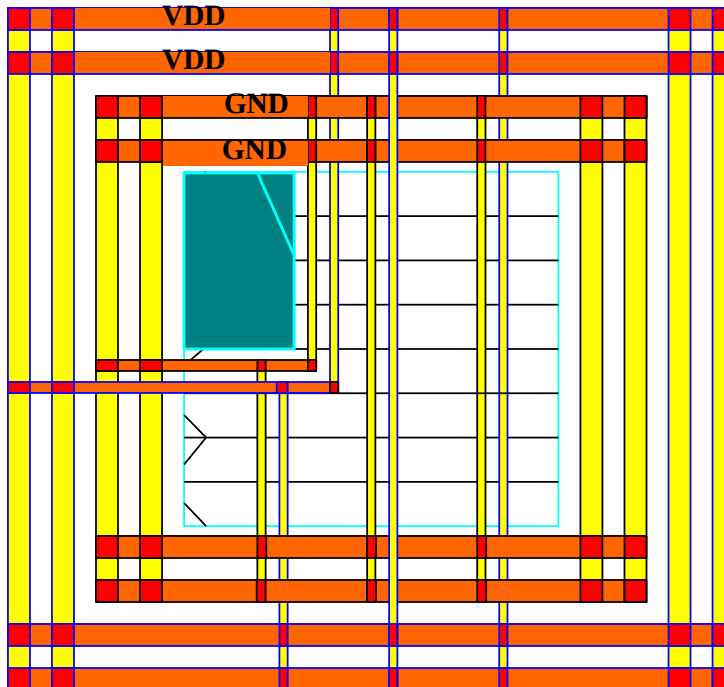


Power Planning: Block Ring



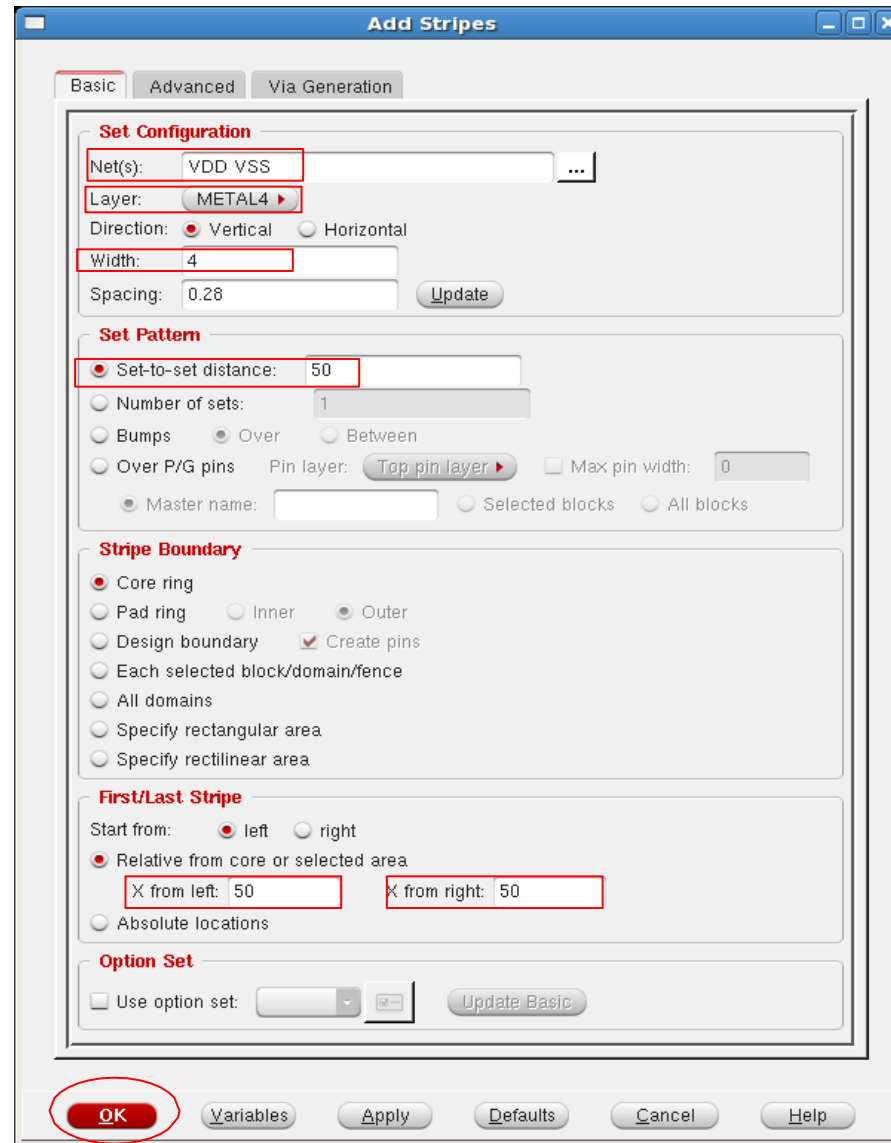


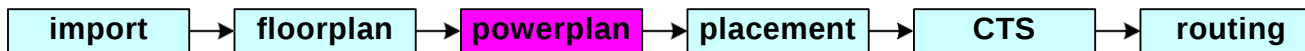
Power Planning: Add Stripes



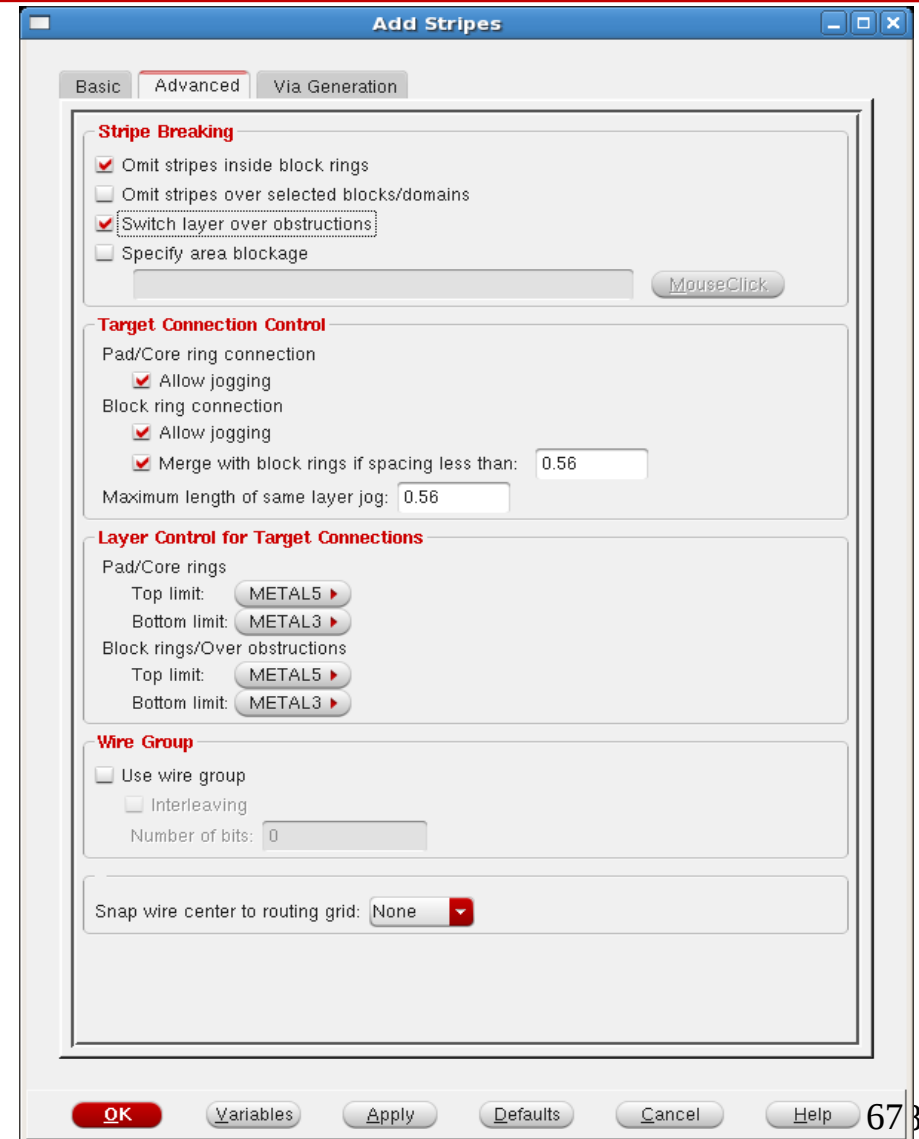
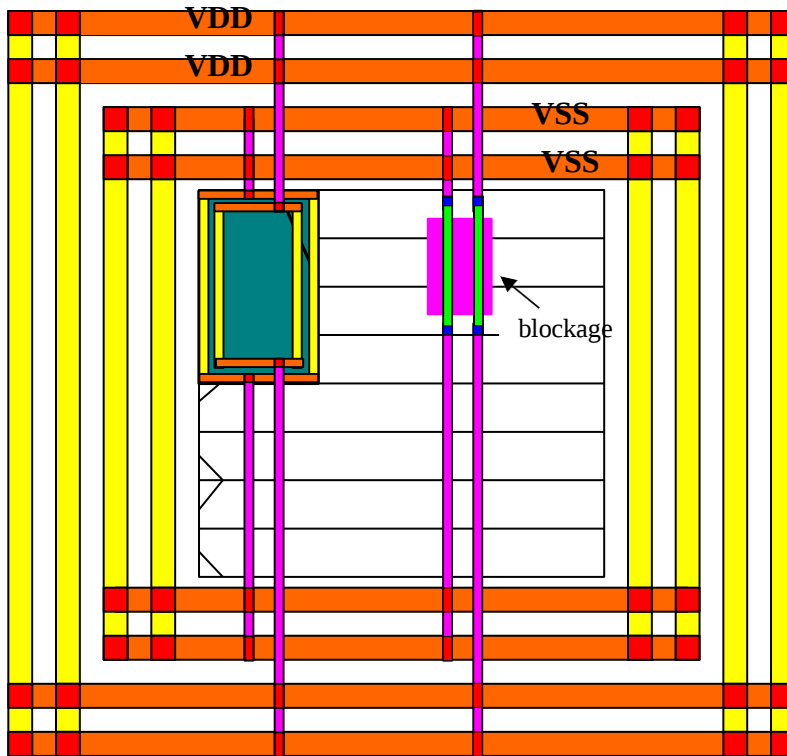
IR drop

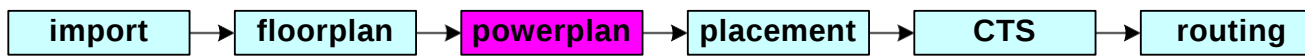
Power Planning: Add Stripes





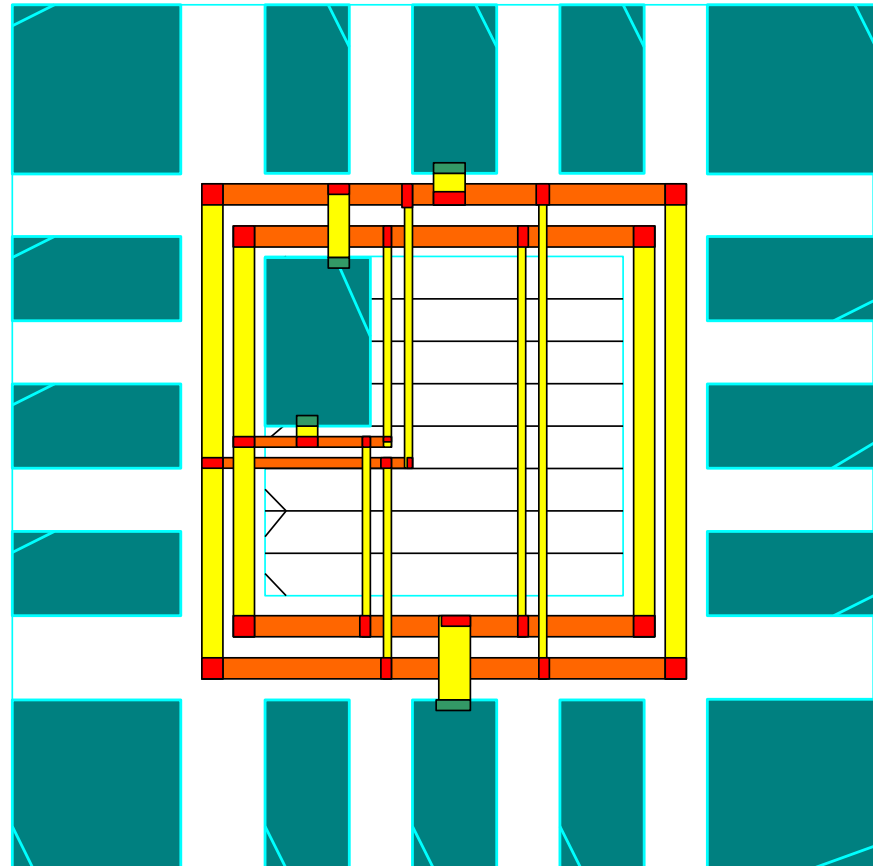
Power Planning: Add Stripes

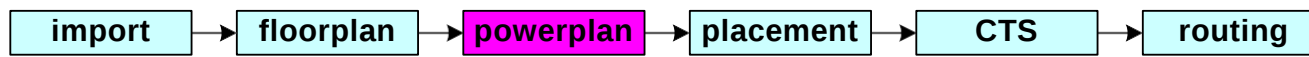




SRoute

- ***Route → Special Route***
- Route Special Net (power/ground net)
 - Block pins
 - Pad pins
 - Follow pins
 - Floating Stripes
 - Secondary Power Pins

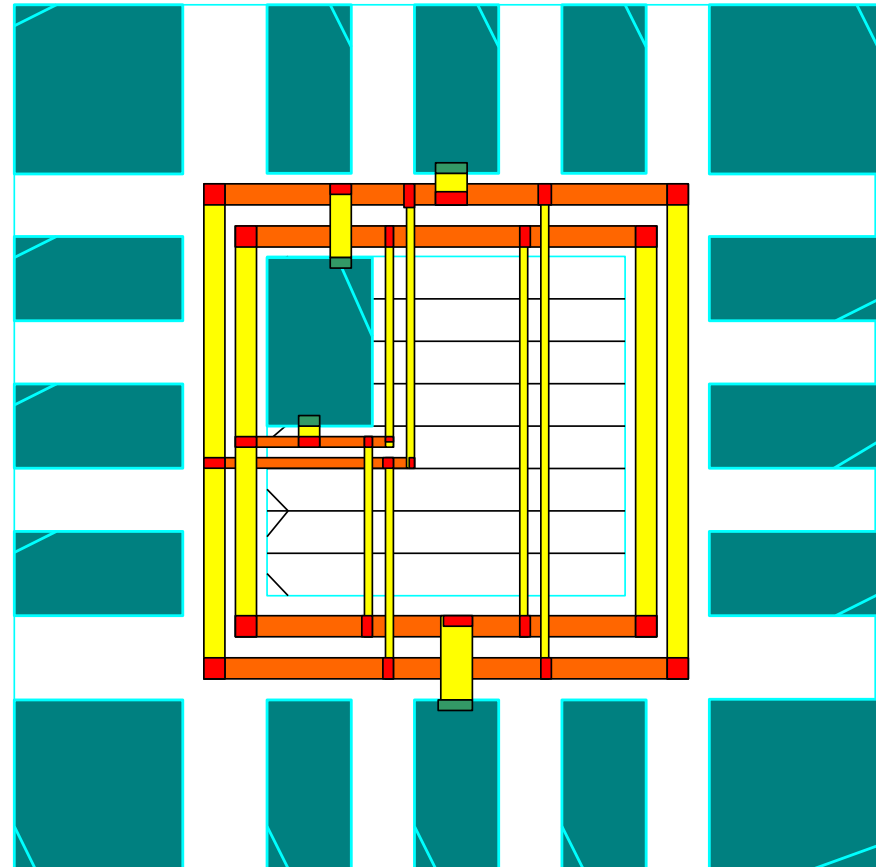


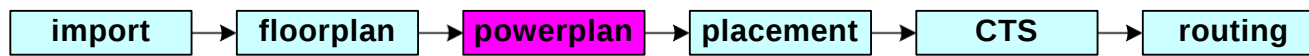


PowerPlan Order

hint: connect wider nets prior then narrow ones.

1. create power ring
2. connect pad pin
3. create block ring
4. connect block pin
5. create stripe
6. connect follow pin

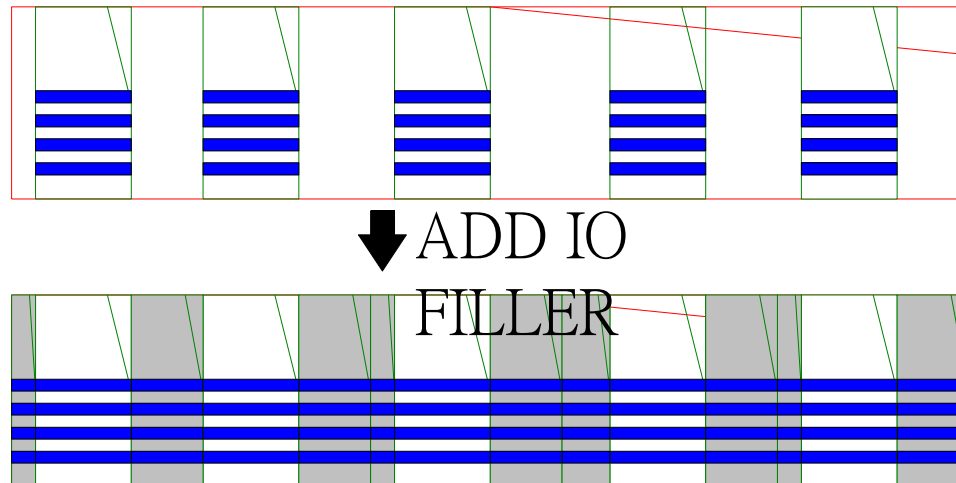




Add IO Filler

addIoFiller **-cell** <fillerCellName>
[**-prefix** <prdfix>]
[**-side** { n|w|s|e }]
[**-fillAnyGap**]

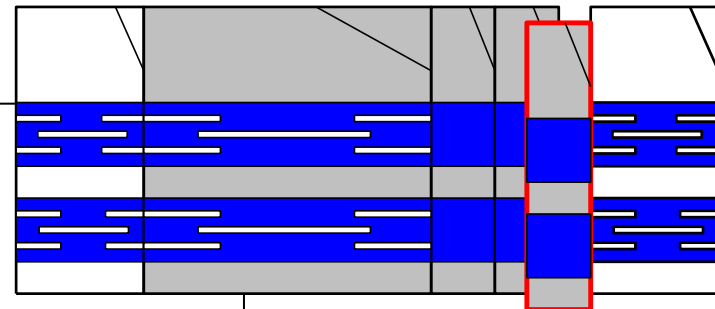
- Connect IO pad power bus by inserting IO filler.
- Add from wider filler to narrower filler.



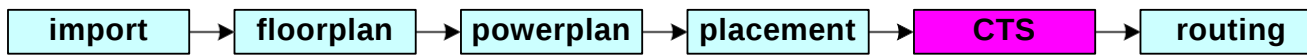
Add IO Filler cont.

- In order to avoid DRCerror
 - The sequence of placing fillers must be from wider fillers to narrower ones.
 - Only the smallest filler can use *-fillAnyGap* option.
- Use addIoFiller.cmd provided in CICdesign kit
 - *source addIoFiller.cmd*

A gap that can place any filler

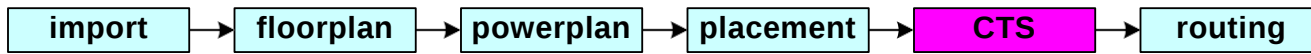


```
addIoFiller -cell PADFILLER20 -prefix IOFILLER
addIoFiller -cell PADFILLER10 -prefix IOFILLER
addIoFiller -cell PADFILLER5 -prefix IOFILLER
addIoFiller -cell PADFILLER1 -prefix IOFILLER
addIoFiller -cell PADFILLER05 -prefix IOFILLER
addIoFiller -cell PADFILLER0005 -prefix IOFILLER -fillAnyGap
```

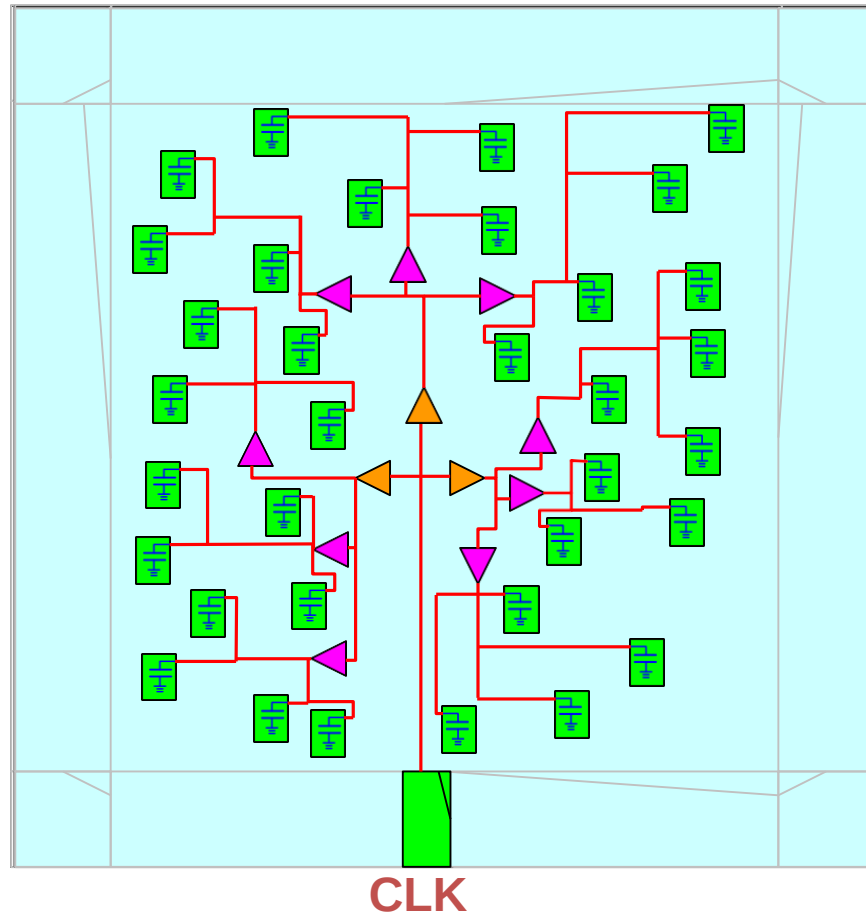


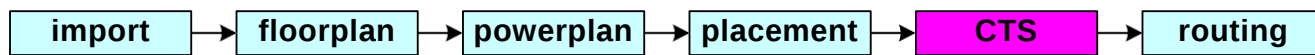
Clock Problem

- Clock problem
 - Heavy clock net loading
 - Long clock insertion delay
 - Clock skew
 - Skew across clocks
 - Clock to signal coupling effect
 - Clock is power hungry

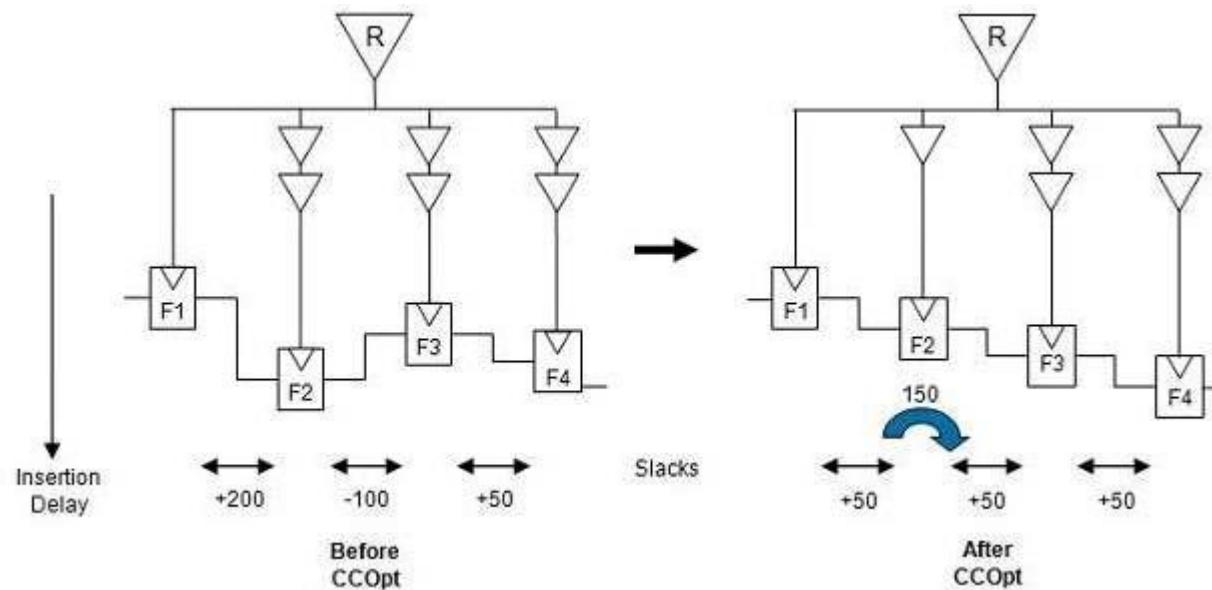


Clock Tree Topology





Clock Concurrent Optimization



Create CCOpt Clock Tree Spec

Create a clock tree specification by analyzing the timing graph structure of all active setup and hold analysis views

`create_ccopt_clock_tree_spec`

or written to a file for inspection and then loaded

`create_ccopt_clock_tree_spec -file ccopt.spec`

`source ccopt.spec`

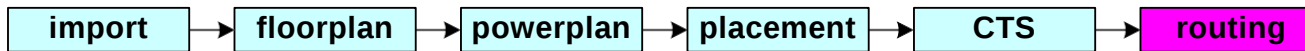
A clock tree specification contains `clock_tree`, `skew_group`, and property settings.

Optimization

Optimize → Optimize Design...

- Optimization
 - setup time
 - hold time
 - DRV (Design Rule Violation)





NanoRoute

Route → NanoRoute → Route

NanoRoute

Routing Phase

- ☒ Global Route
- ☒ Detail Route Start Iteration 0 End Iteration default
- Post Route Optimization ☒ Optimize Via ☒ Optimize Wire

Concurrent Routing Features

- ☒ Fix Antenna ☒ Insert Diodes Diode Cell Name ANTENNA
- ☒ Timing Driven Effort 5 Congestion Timing S.M.A.R.T.
- ☒ SI Driven
- ☒ Post Route SI SI Victim File
- ☐ Litho Driven
- ☐ Post Route Litho Repair

Routing Control

- ☐ Selected Nets Only Bottom Layer default Top Layer default
- ☐ ECO Route
- ☐ Area Route Area Select Area and Route

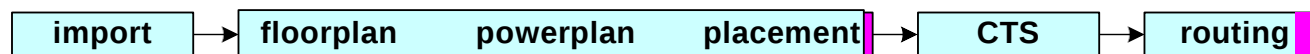
Job Control

- ☒ Auto Stop Number of Local CPU(s): 1
- Number of CPU(s) per Remote Machine: 1
- Number of Remote Machine(s): 0
- Set Multiple CPU...

OK Apply Attribute Mode Save Load Cancel Help

Optimize Via

Optimize Wire



Verify Geometry

Verify → *Verify Geometry*

Verify Geometry

Basic | Advanced

Verification Area

☒ Entire area
☐ Specify Draw View Area

X1: 0 Y1: 0
X2: 0 Y2: 0

Check

☒ Minimum Width	☒ Minimum Spacing
☒ Minimum Area	☒ Same Net Spacing
☒ Short	☐ Geometry Antenna
☒ Cell Overlap	☐ Off Routing Grid
☒ Insufficient Metal Overlap	☒ Off Manufacturing Grid
☒ MinHole	☒ Implant Check
☒ Minimum Cut	☒ MinStep
☒ Via Enclosure	

Allow

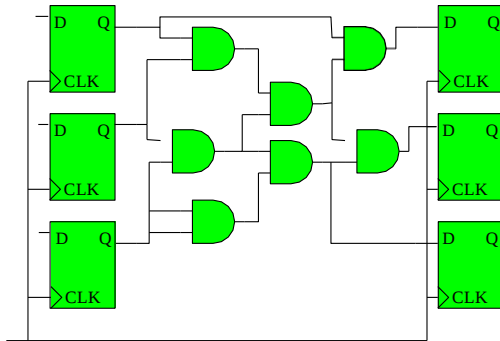
☒ Pin In Blockage
☒ Same Cell Violations
☐ Different Cell Violations
☐ Overlap of Pad Filler Cells
☐ Overlap of Routing Blockages And Pins
☐ Overlap of Routing Blockage And Cell Blockage

OK Apply Reset Cancel Help

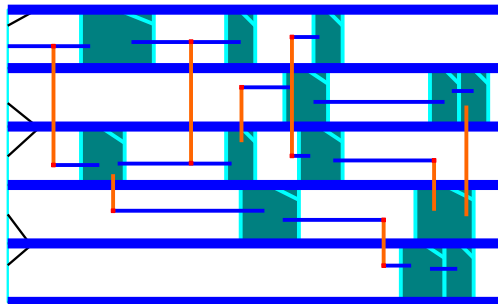
Verify Connectivity

Verify → Verify Connectivity

connectivity
database



layout
database



Verify Connectivity

Net Type

☒ All
 ☐ Regular Only
 ☐ Special Only

Nets

☒ All
 ☐ Selected
 ☐ Named:

Check

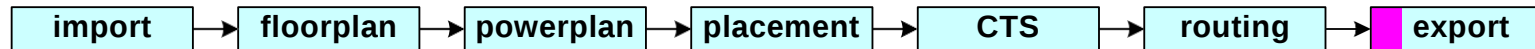
☒ Open
 ☒ UnConnected Pin
 ☒ Unrouted Net
 ☐ Connectivity Loop
 ☒ DanglingWire (Antenna)
 ☒ Weakly Connected Pin
 ☐ Geometry Loop
 ☐ Geometry Connectivity
 ☐ Keep Previous Results
 ☐ TSV Die Abstract File

Verify Connectivity Report:
 CHIP.conn.rpt

Report Limits

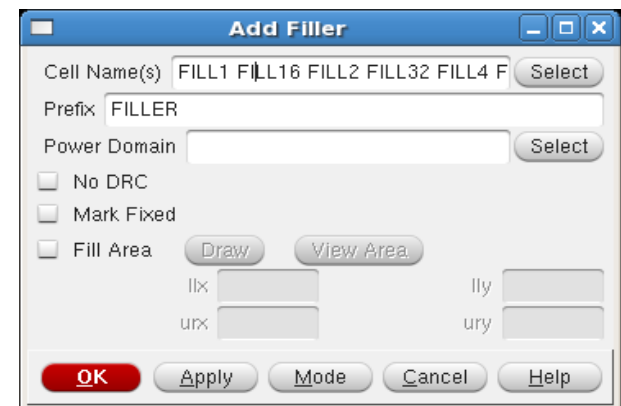
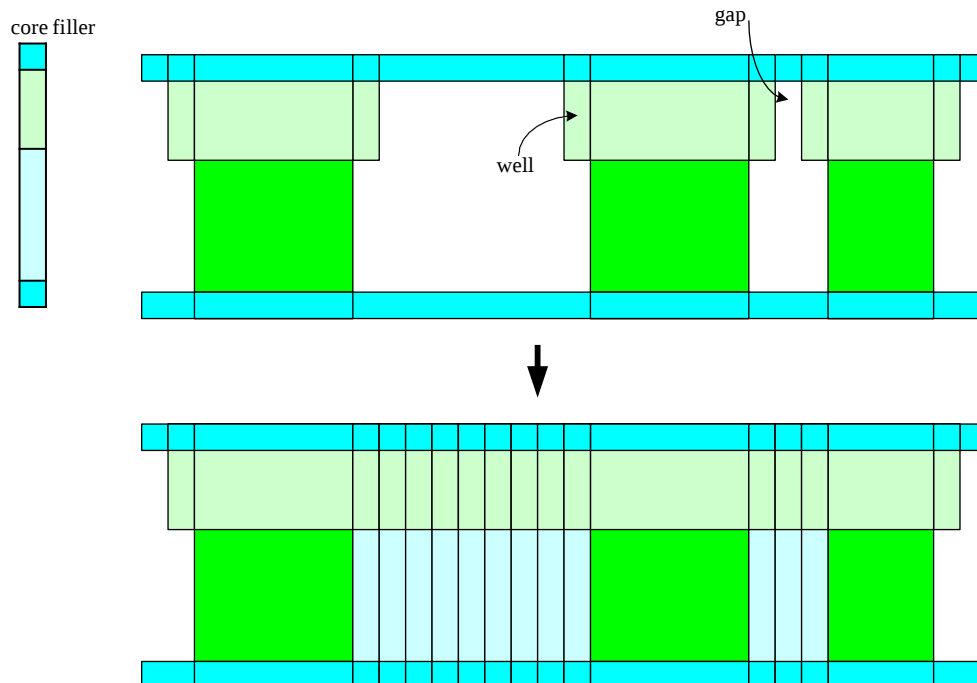
Error: 1000
 Warning: 50

Add Core Filler

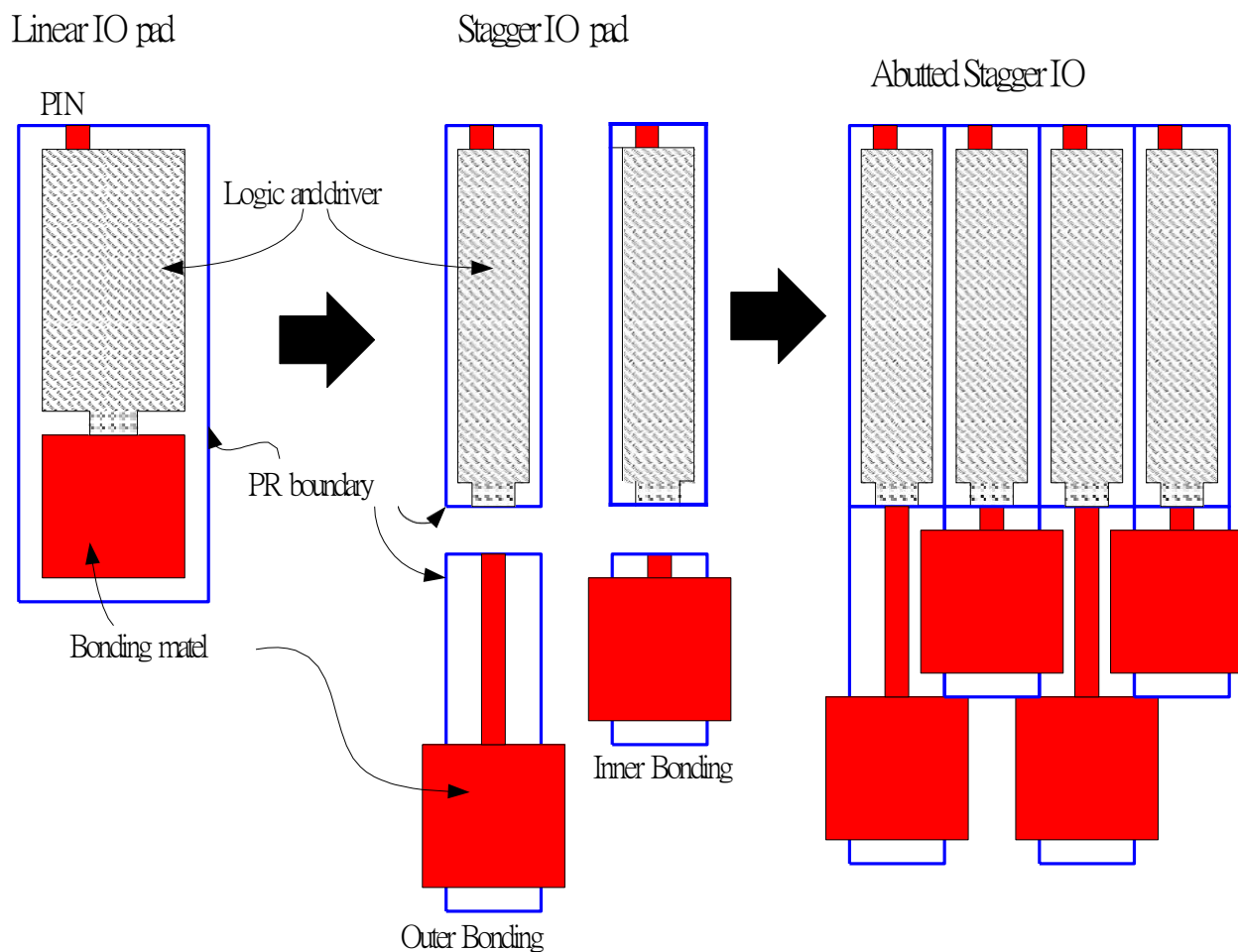
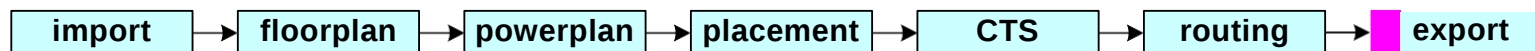


Place → Filler → Add Filler...

- Connect the NWELL/PWELL layer in core rows.
- Insert Well contact.
- Add from wider filler to narrower filler.

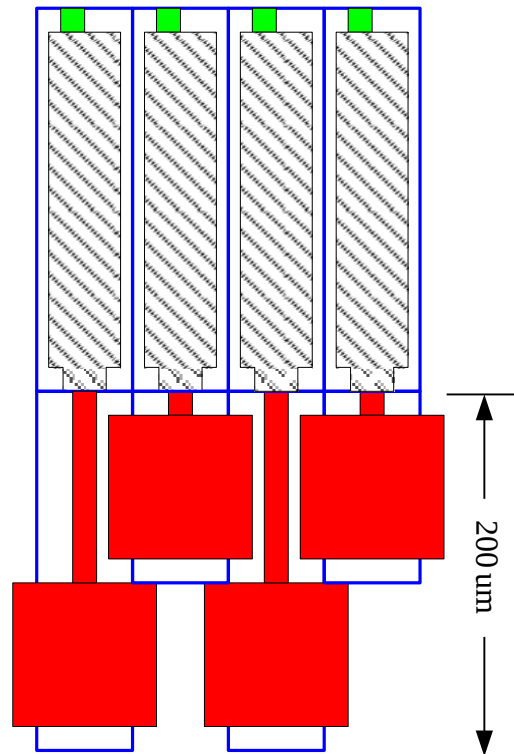


Add Bonding Pads

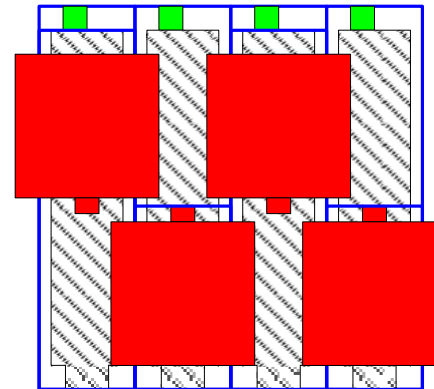


Circuit under Pad

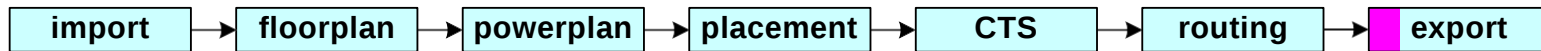
traditional bonding pad



CUP bonding pad



Add Bonding Pads

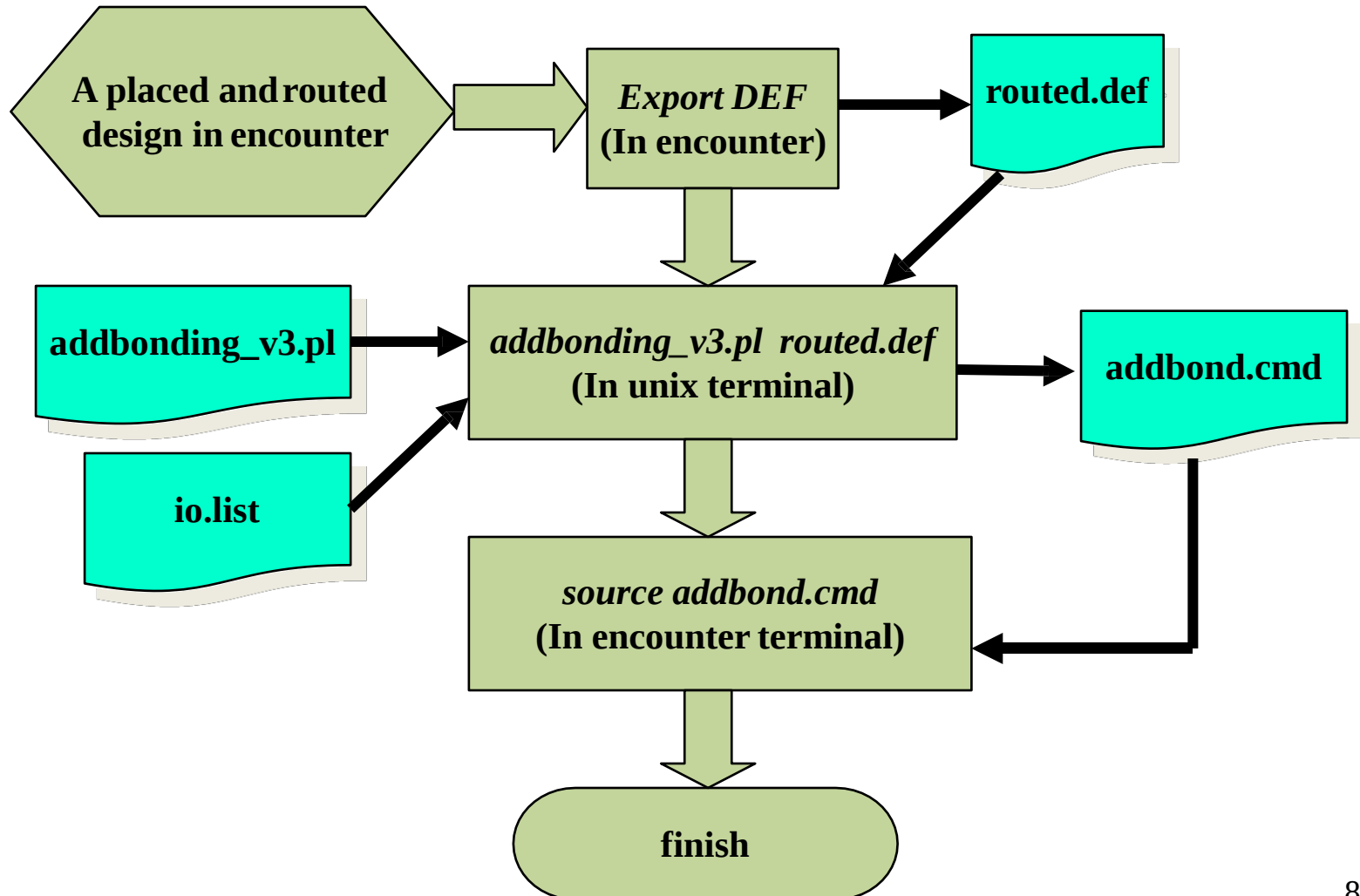
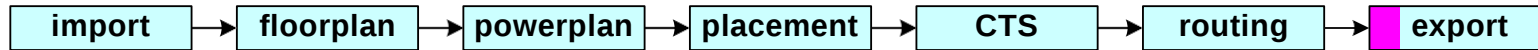


- For the limitation of bonding wire technique , the stagger IO pads are used in order to reduce IOpad width.
- We have to add the bonding pads after APR is finished if stagger IO pads is used. But Encounter does not provide a built-in function for add bonding pads, ClC reaches this purpose by the way of importing DEF.
- ClC provides a perl script to calculate the bonding pad location. The full flow is described in next page

Add Bonding Pads Flow

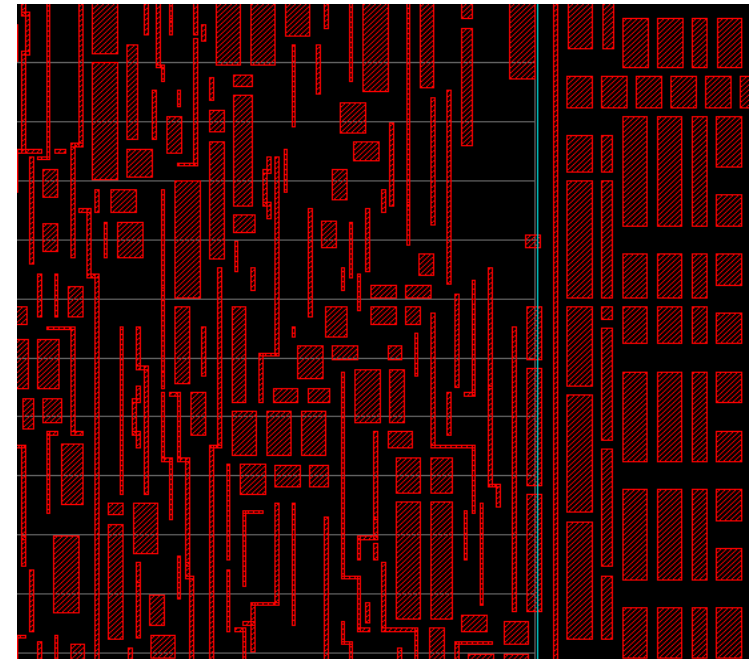
(stagger IO pads only)

NARLabs



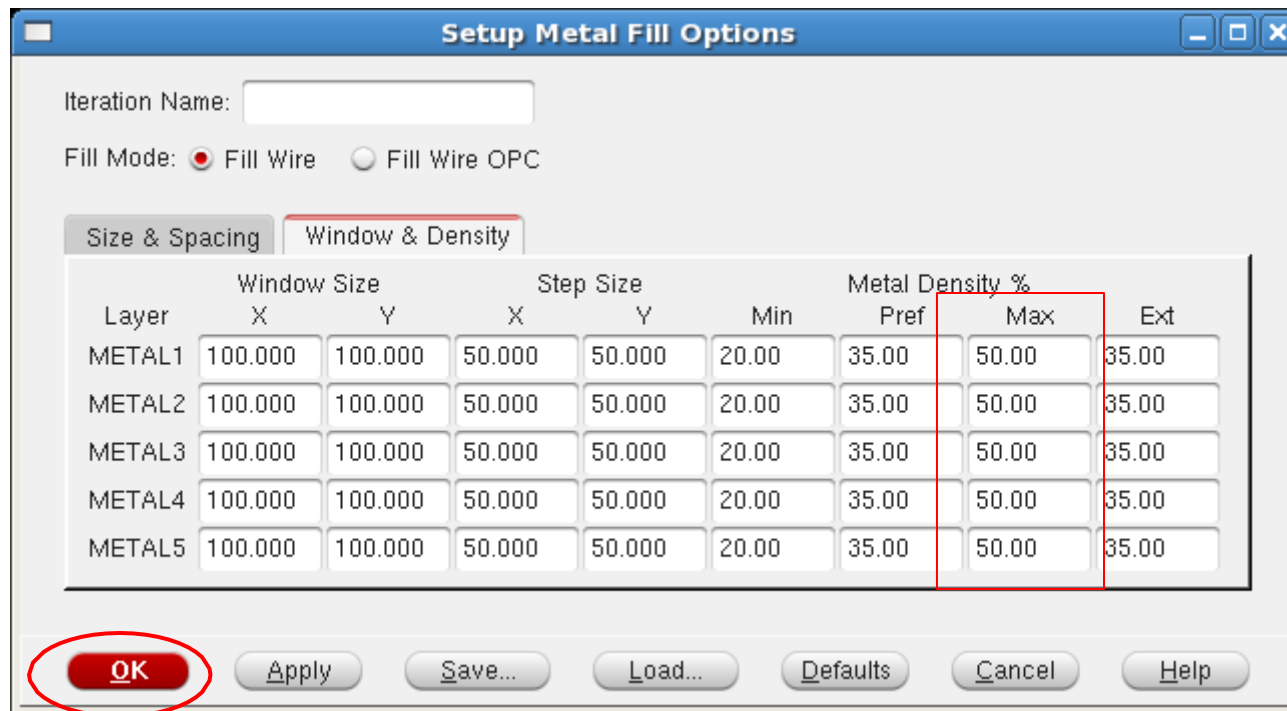
Add Dummy Metal

- Why add dummy
 - meet minimize metal density rule
 - prevent over etching
 - prevent sagging in local area
 - improve yield
 - reduce on chip variation
- Better connect dummy metal to VSS
- Side effect
 - introduce parasitic to signal line



Add Dummy Metal

Route → Metal Fill → Setup...



The dialog box titled "Setup Metal Fill Options" contains the following elements:

- Iteration Name:
- Fill Mode: ☒ Fill Wire ☐ Fill Wire OPC
- Two tabs: "Size & Spacing" (selected) and "Window & Density".
- A table with 9 columns: Layer, Window Size X, Window Size Y, Step Size X, Step Size Y, Min, Metal Density % Pref, Metal Density % Max, and Ext.
- Five rows of data for METAL1 through METAL5.
- A red rectangle highlights the "Metal Density % Max" column.
- Buttons at the bottom: OK (circled in red), Apply, Save..., Load..., Defaults, Cancel, and Help.

Layer	Window Size		Step Size		Min	Metal Density %		Ext
	X	Y	X	Y		Pref	Max	
METAL1	100.000	100.000	50.000	50.000	20.00	35.00	50.00	35.00
METAL2	100.000	100.000	50.000	50.000	20.00	35.00	50.00	35.00
METAL3	100.000	100.000	50.000	50.000	20.00	35.00	50.00	35.00
METAL4	100.000	100.000	50.000	50.000	20.00	35.00	50.00	35.00
METAL5	100.000	100.000	50.000	50.000	20.00	35.00	50.00	35.00

Add Dummy Metal

Route → Metal Fill → Add...

Add Metal Fill

Number of Local CPU(s): 1 [Set Multiple CPU...](#)

Iteration Name List:

Model Selection

Shape: ☒ Rectangle ☐ Square

Connection: ☐ Tie High/Low to Net(s):

Connection Shape: ☒ Tree ☐ Mesh

☒ Keep Unconnected Metal Fill(s)

☐ Square Shape

☐ Use Generated Vias Only

Exclude Vias and Via Rules:

☐ Allow Fill on Cells

Incremental Control

☐ Delete Metal Fill before Creating New Metal Fill

☒ FillWire ☒ FillWireOPC

Layer Selection

☒ METAL1 ☒ METAL2 ☒ METAL3 ☒ METAL4 ☒ METAL5 [All Layers](#)

☒ Timing Aware

☒ Critical Nets from Timing Analysis

Slack Threshold:

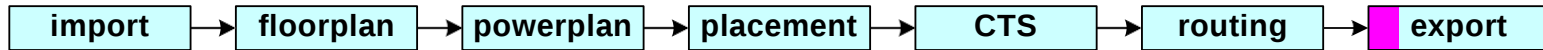
☐ Area [Draw](#)

X1: Y1:

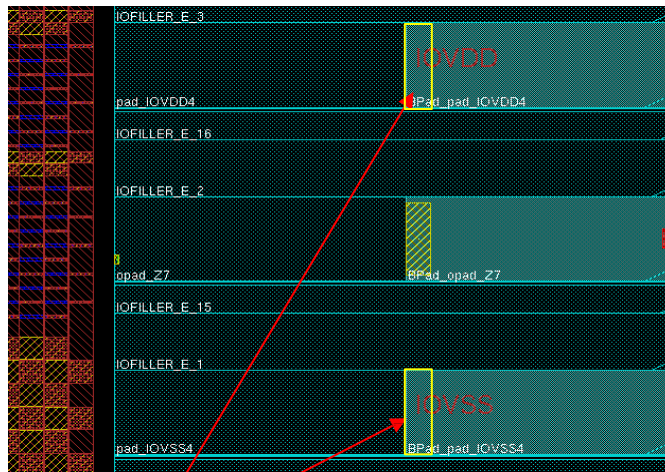
X2: Y2:

OK [Apply](#) [Defaults](#) [Cancel](#) [Help](#)

Add Text IOVDD & IOVSS **NAR Labs**

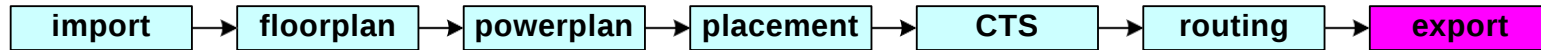


add_text -layer METAL5 -label IOVSS -pt 1365 1095 -height 10



add text location

Output Data



Design → Save → GDS...

Design → Save → Netlist...

write_sdf

Design → Save → DEF

- Export GDS for DRC, LVS, LPE, and tape out.
- Export Netlist for LVS and simulation.
- Export Netlist and sdf for post layout simulation
- Export DEF for reordered scan chain.

Export Sdf

◆ source savesdf.cmd

savesdf.cmd

```
setAnalysisMode -analysisType bcwc
write_sdf      -max_view  av_func_mode_max \
               -typ_view   av_func_mode_typ \
               -min_view   av_func_mode_min \
               -edges noedge \
               -splitsetuphold \
               -remashold \
               -splitrecrem \
               -min_period_edges none \
               CHIP.sdf
```

CHIP.sdf

```
(CELL
  (CELLTYPE "INVXL")
  (INSTANCE DFT_shared_out_mux_6)
  SLAY
  (DE SOLUTE
  (ABATH A Y (0.14:0.29:0.32)(0.08:0.17:0.23))
  (IOP
  )
  )
)
```

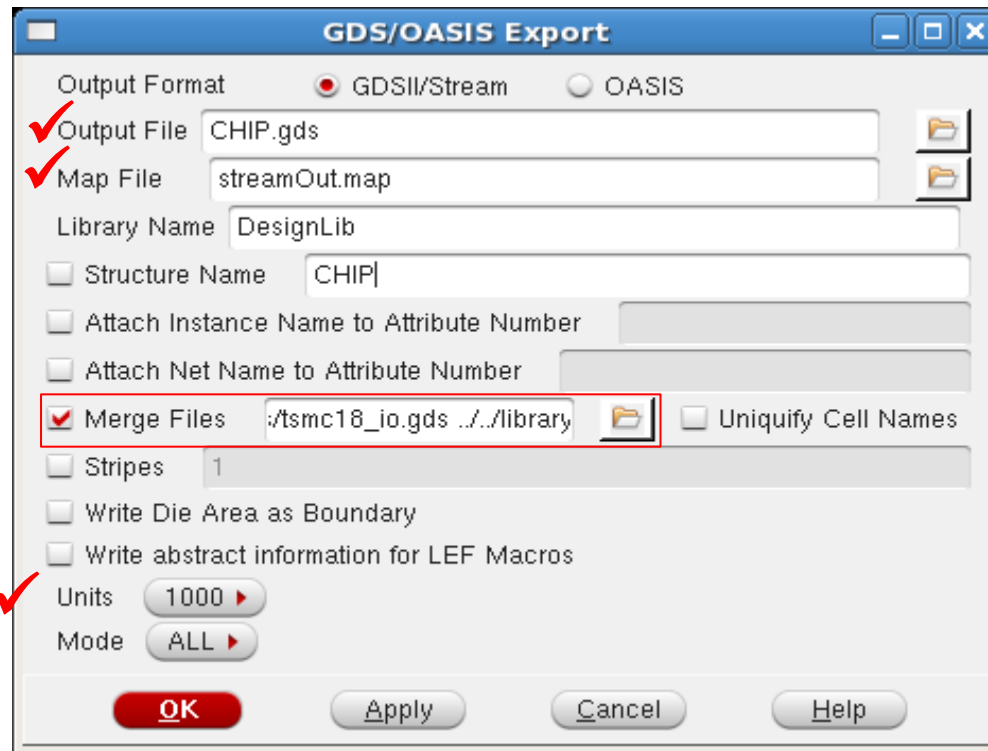
Stream Out

Edit → Save → GDS/OASIS...

- source savegds.cmd

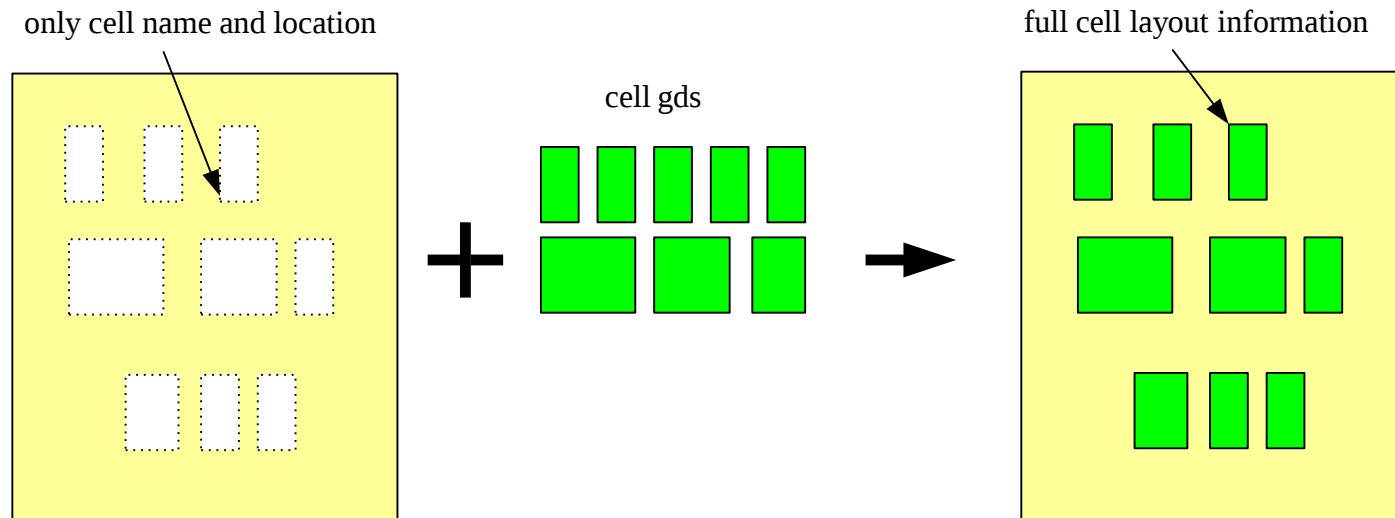
savegds.cmd

```
streamOut CHIP.gds \
    -mapFile streamOut.map \
    -merge { gds/RF2SH64x16.gds \
            gds/tpb973gv.gds \
            gds/tsmc18_core.gds \
            gds/tsmc18_io.gds } \
    -stripes 1 -units 1000 -mode ALL
```



Stream Out

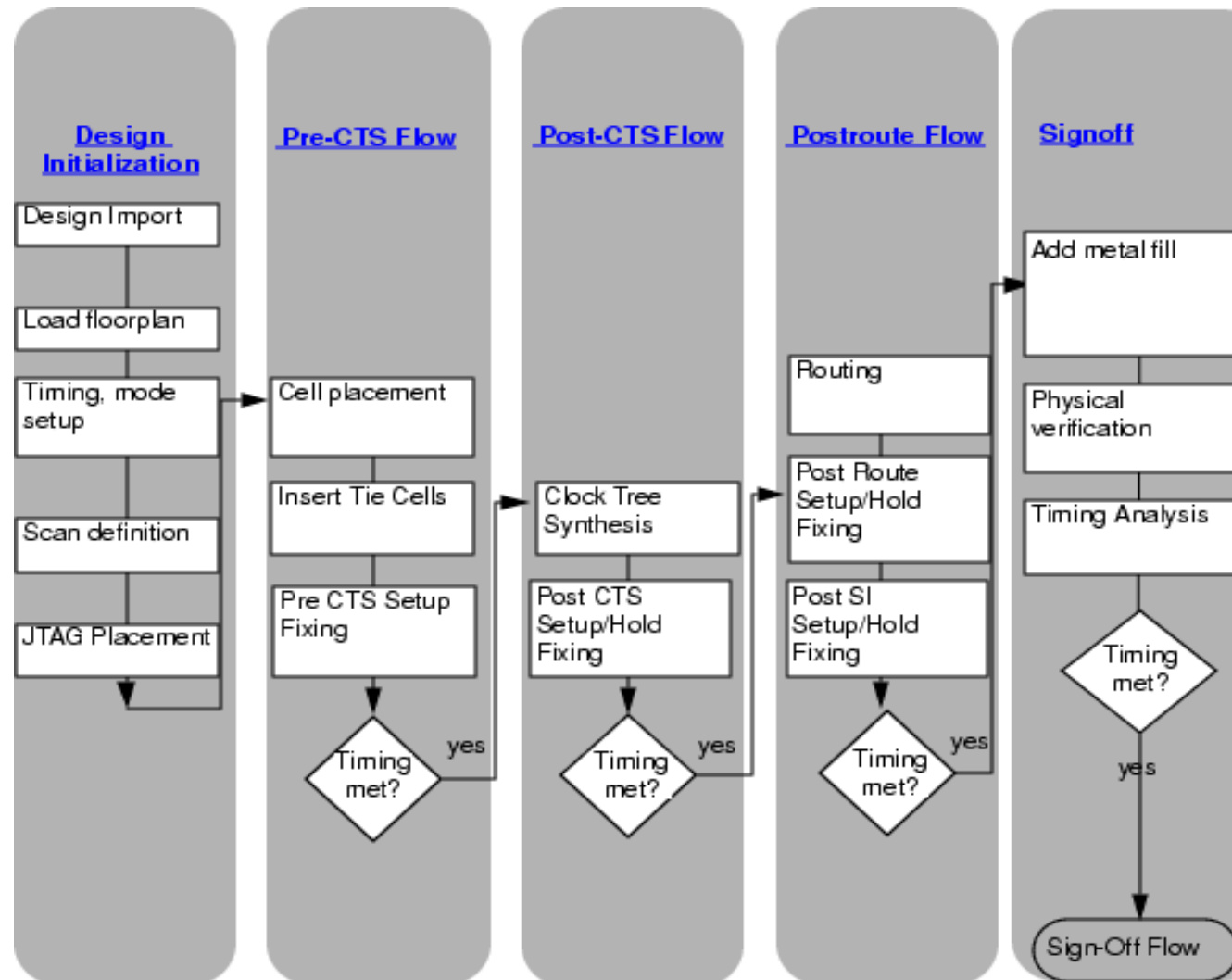
- Merge gds





FOUNDATION FLOW

Flow Step



Create Flow Environment

1. Create Flow template

- *Flows → Create Foundation Flow Template → Save*
- *writeFlowTemplate* ← or

2. Prepare setup file

- *Flows → Foundation Flow Wizard...*
- *SCRIPTS/gen_edl_setup.tcl* ← or

3. Generate script

- *SCRIPTS/gen_edl_flow.tcl*

Foundation Flow Wizard

- *Flows → Foundation Flow Wizard...*

Foundation Flow Wizard

EDI System

Welcome to Foundation Flow Wizard

This wizard will take you through the Foundation Flow. Once you make all your selections the wizard will generate a script which you can save for your use. On each page you can click "Tips" to get a brief overview of either what is to be selected or why a selection is required.

*How do you want to start?

☐ Start from scratch

☒ Load the design setup from memory

☐ Load previously saved script list

*Foundation flow install directory: /

Save foundation flow database at: DBS

Save foundation flow reports at: RPT

*Indicates mandatory fields

Continue >



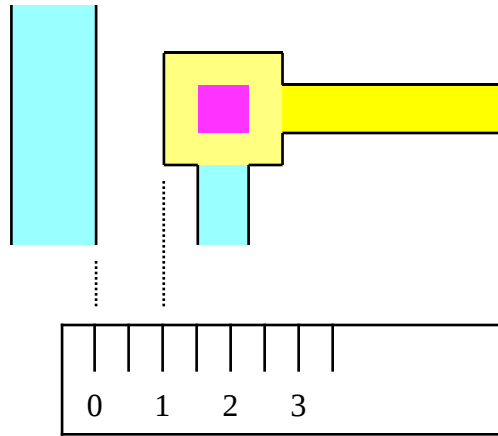
POST-LAYOUT VERIFICATION

Post-Layout Verification Overview

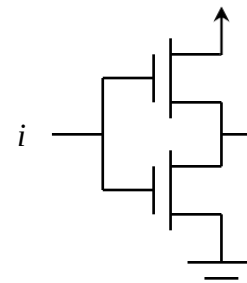
- Post-Layout Verification do the following things : (by Mentor Calibre)
 - DRC (Design Rule Check)
 - LVS (Layout versus Schematic)
 - ERC (Electrical Rule Check)
 - LPE/PRE (Layout Parasitic Extraction / Parasitic Resistance Extraction) and Post-Layout Simulation.

Post-Layout Verification Overview_{cont.}

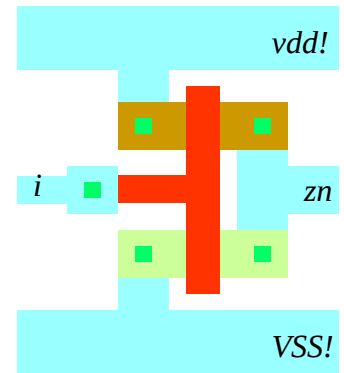
DRC



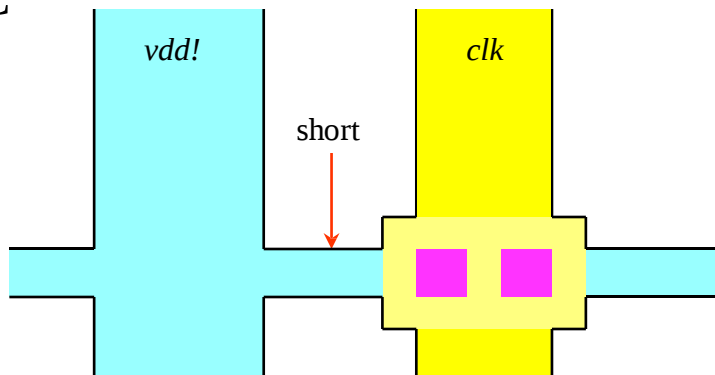
LVS



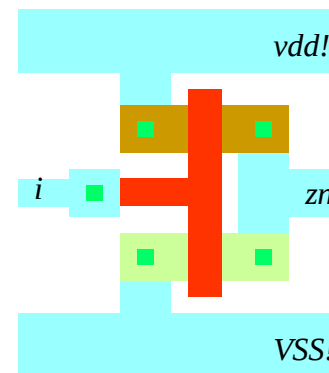
compare with



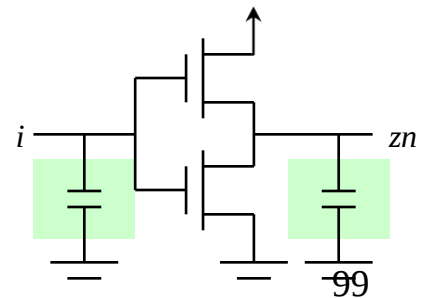
ERC



LPE/PRE



extract



DRC Flow

- Prepare Layout
- Prepare command file
- Run DRC
- View DRC error (DRC summary/RVE)

- Stream out with cell gds merged
- Be sure to use layer map file provided by CIC

Prepare Command File

- Prepare DRCCommand file:
 - **TSMC 90nm** (*CBDK_TSMC90G_Arm*) Calibre
 - CLN90S_3XTM_9M.22a1
 - **TSMC 0.18** (*CBDK018_TSMC_Artisan*) Calibre
 - CLM18_LM16_6M.28a_m.drc

Prepare Calibre Command file

- Edit runset file

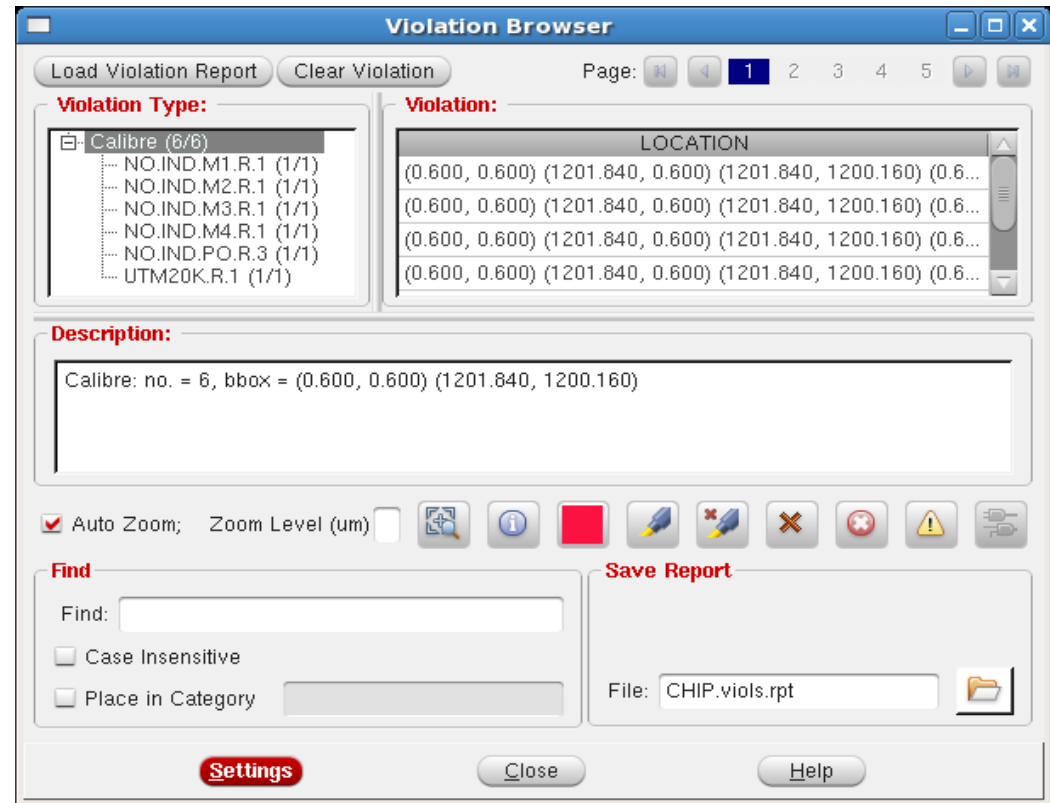
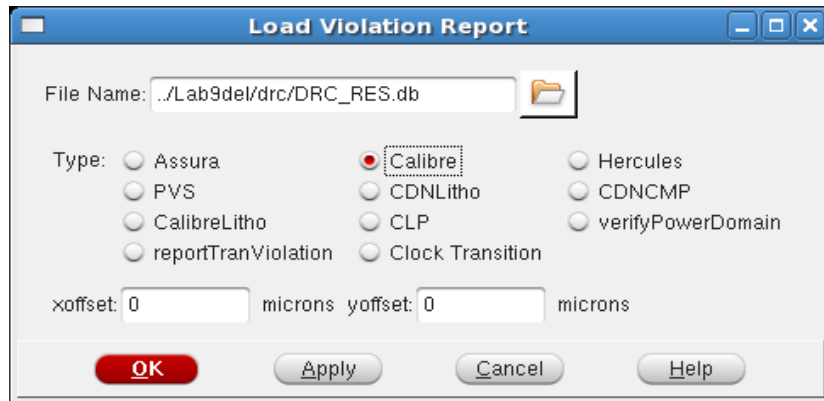
```
LAYOUT PATH "CHIP.gds2"  
LAYOUT PRIMARY "CHIP"  
LAYOUT SYSTEM GDSII  
...  
...  
...  
DRC SELECT CHECK  
    NW.W.1  
    NW.W.2  
    ...  
DRC UNSELECT CHECK  
    NW.S.1Y  
    NW.S.2Y  
    ...  
DRC ICSTATION YES  
INCLUDE "Calibre-drc-cur"
```

Submit Calibre Job

- Submit Calibre Job
 - `unix% calibre -drc CLM18_LM16_6M.28a_m.drc`
 - Result log
 - DRC.sum (ASCII result)
 - DRC.db (Graphic result)

View Calibre Result in SOCEncounter

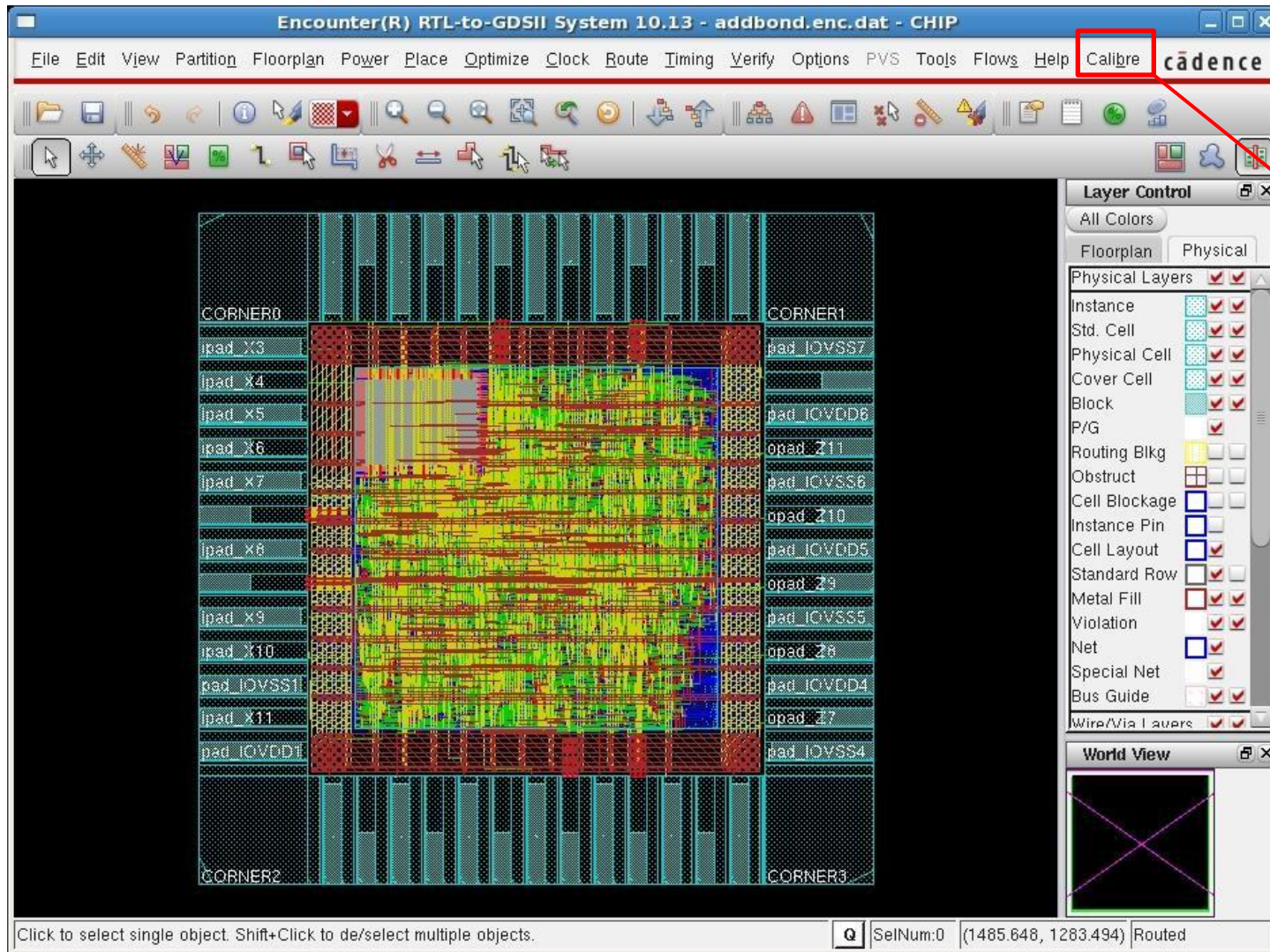
Tools → Violation Browser...



Calibre Interactive in Encounter

- STEP1
 - source calibre.cshrc
 - source edi.cshrc
 - exec encounter
 - In encounter terminal:
source /usr/cad/mentor/calibre/cur/lib/cal_enc.tcl

Calibre Menu in Encounter



Setup Streamout Options

- STEP2: *calibre* → *Setup* → *GDS Export*

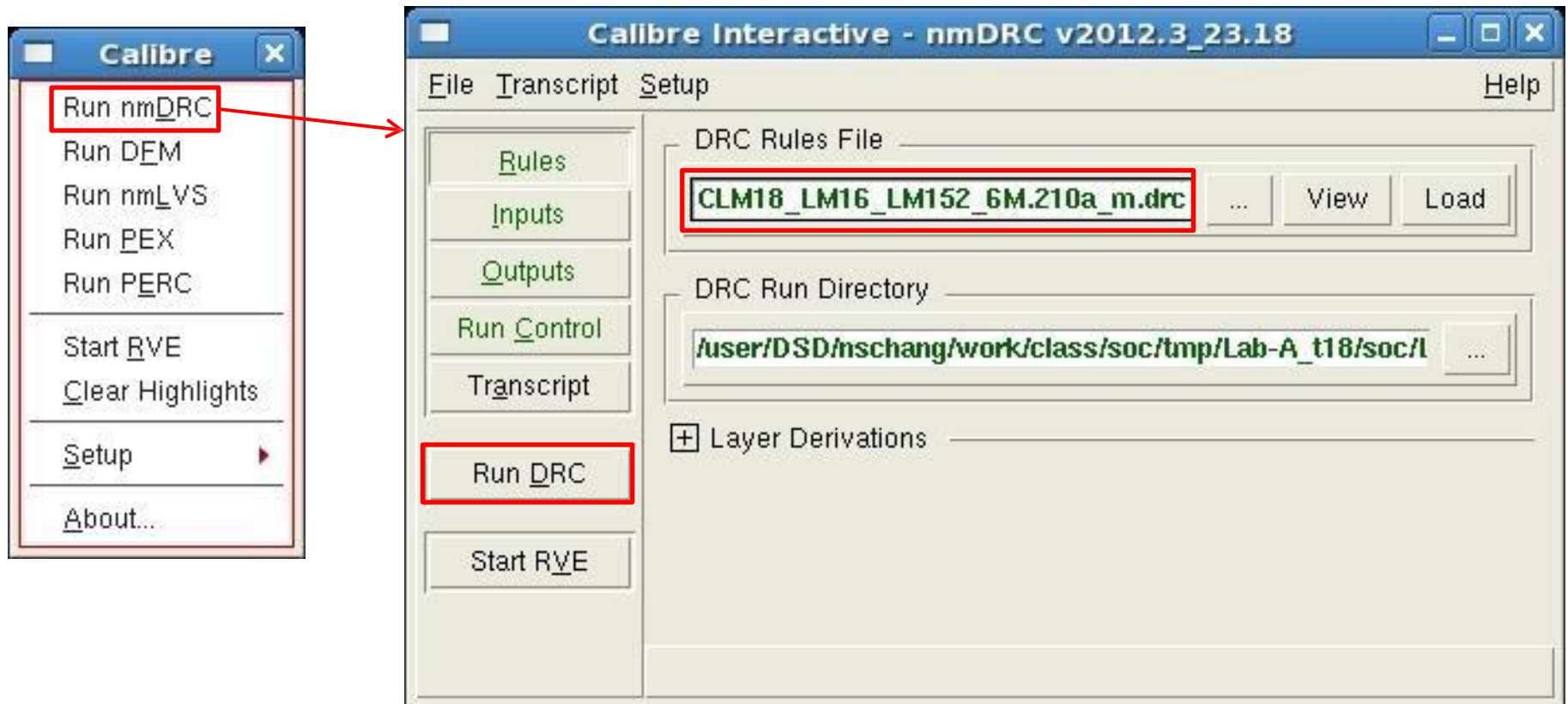


streamOut CHIP.gds \

```
-mapFile streamOut.map \
-merge { gds/RF2SH64x16.gds \
        gds/tpb973gv.gds \
        gds/tsmc18_core.gds \
        gds/tsmc18_io.gds } \
-stripes 1 -units 1000 -mode ALL
```

Calibre Interactive

- STEP3: *calibre* → *Run nmDRC*



Calibre RVE

The screenshot displays the Calibre RVE v2012.3_23.18 interface. The main window shows a physical layout of a chip with various colored regions representing different layers or materials. A sidebar on the left lists the results of a Design Rule Check (DRC) for the file 'CHIP.drc.results'.

Calibre - RVE v2012.3_23.18 : CHIP.drc.results

File View Highlight Tools Window Setup Help

Topcell CHIP, 11 Results (in 11 of 405 Checks) Show All

Check / Cell

- Check M3.S.1
- Check NO.IND.OD.
- Check NO.IND.PO.I
- Check NO.IND.M1.I
- Check NO.IND.M2.I
- Check NO.IND.M3.I
- Check NO.IND.M4.I
- Check NO.IND.M5.I
- Check UTM20K.R.1
- Check OD.R.1

1) Check M3.S.1, Cell CHIP: 6-V

```

M3.S.1 { @ Min. M3 space < 0.28
EXT M3 < 0.28 ABUT < 90 SINGULAR REGION
}

```

Check M3.S.1

Encounter(R) RTL-to-GDSII System 10.13 - addbond.enc.dat - CHIP

File Edit View Partition Floorplan Power Place Optimize Clock Route Timing Verify Options PVS Tools Flows Help Calibre cadence

Layer Control

All Colors

Floorplan Physical

Physical Layers

- Instance
- Std. Cell
- Physical Cell
- Cover Cell
- Block
- P/G
- Routing Blkg
- Obstruct
- Cell Blockage
- Instance Pin
- Cell Layout
- Standard Row
- Metal Fill
- Violation
- Net
- Special Net
- Bus Guide
- Wire/Via Layers

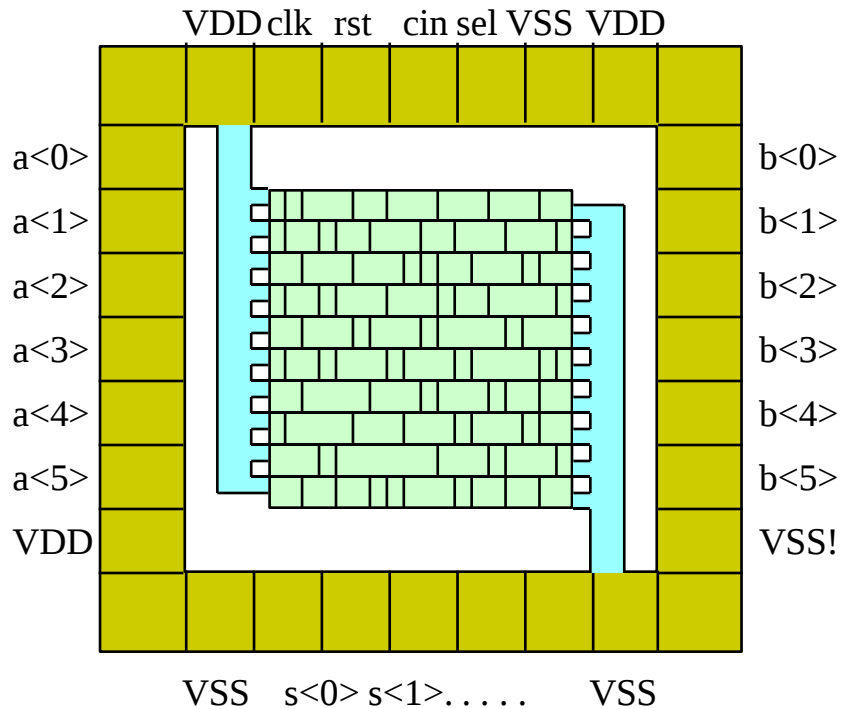
World View

Click to select single object. Shift+Click to de/select multiple objects.

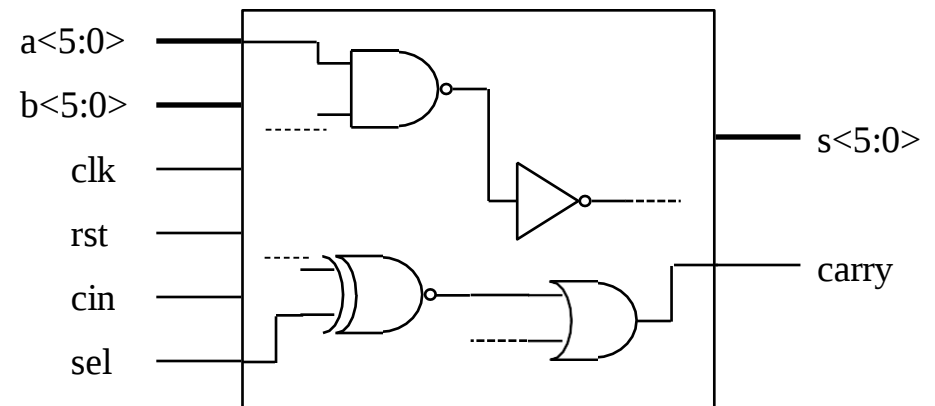
Q SelNum:0 (643.255, 742.698) Routed 110

LVS Overview

Layout Data



Schematic Netlist

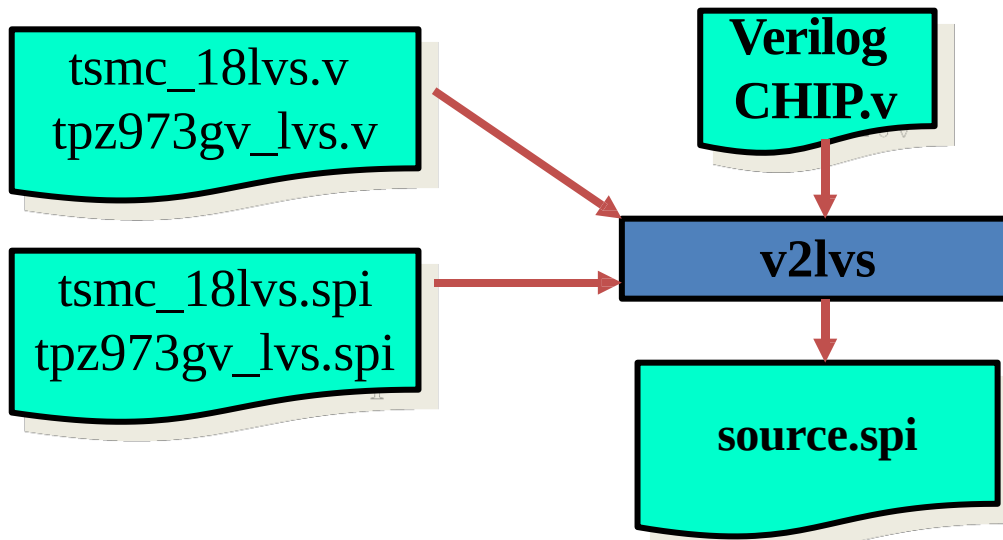


LVS Flow

- Prepare Layout
 - The same as DRCPrepare Layout
- Prepare Netlist
 - v2lvs
- Prepare calibre command file
- Run calibre LVS
- View LVSError (LVSSummary/RVE)

Prepare Netlist for Calibre LVS

Prepare Netlist



- `v2lvs -v CHIP.v -l tsmc18_lvs.v -l tpz973gv_lvs.v -s tsmc18_lvs.spi -s tpz973gv_lvs.spi -o source.spi -s1 VDD -s0 VSS`
- If a macro DRAM64x16 is used
- `v2lvs -v CHIP.v -l tsmc18_lvs.v -l tpz973gv_lvs.v -l DRAM64x16.v -s tsmc18_lvs.spi -s tpz973gv_lvs.spi -s DRAM64x16.spi -o source.spi -s1 VDD -s0 VSS`

CIC Supported Files (tsmc0.18)

◆ CIC supports the following files in our cell library design kit.

➤ Calibre LVS rule file

Calibre.lvs

➤ Black-box LVS relative files

✓ pseudo spice file

tsmc18_lvs.spi

tpz973gv_lvs.spi

✓ **pseudo verilog** file

tsmc18_lvs.v

tpz973gv_lvs.v

Generate Pseudo Verilog File

- Gen pseudo verilog for from simulation model, but leaving only header definition.
- Gen pseudo spice by run v2lvs on pseudo verilog

unix% **v2lvs -v RF2SH64x16.v**

```
.SUBCKT RF2SH64x16 QA[15] QA[14] QA[13] QA[12] QA[11] QA[10] QA[9] QA[8] QA[7]
+ QA[6] QA[5] QA[4] QA[3] QA[2] QA[1] QA[0] AA[5] AA[4] AA[3] AA[2] AA[1] AA[0]
+ CLKA CENAAB[5] AB[4] AB[3] AB[2] AB[1] AB[0] DB[15] DB[14] DB[13] DB[12]
+ DB[11] DB[10] DB[9] DB[8] DB[7] DB[6] DB[5] DB[4] DB[3] DB[2] DB[1] DB[0]
+ CLKB CENB
.ENDS
```

- **ADD VDD VSSport on pseudo spice**

```
.SUBCKT RF2SH64x16 QA[15] QA[14] QA[13] QA[12] QA[11] QA[10] QA[9] QA[8] QA[7]
+ QA[6] QA[5] QA[4] QA[3] QA[2] QA[1] QA[0] AA[5] AA[4] AA[3] AA[2] AA[1] AA[0]
+ CLKA CENAAB[5] AB[4] AB[3] AB[2] AB[1] AB[0] DB[15] DB[14] DB[13] DB[12]
+ DB[11] DB[10] DB[9] DB[8] DB[7] DB[6] DB[5] DB[4] DB[3] DB[2] DB[1] DB[0]
+ CLKB CENB VDD VSS
.ENDS
```

```
module RF2SH64x16 (
    QA,
    AA,
    CLKA,
    CENA,
    AB,
    DB,
    CLKB,
    CENB
);
    output [15:0] QA;
    input [5:0] AA;
    input CLKA;
    input CENA;
    input [5:0] AB;
    input [15:0] DB;
    input CLKB;
    input CENB;

endmodule
```

Prepare command file for Calibre LVS

- Edit Calibre LVSrunset

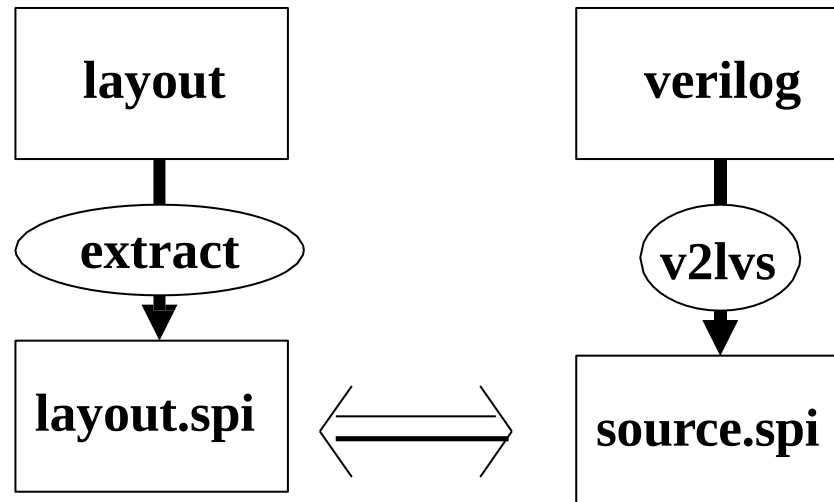
```
LAYOUT PATH "CHIP.calibre.gds"  
LAYOUT PIMARY "CHIP"  
LAYOUT SYSTEM GDSII  
SOURCE PATH "source.spi"  
SOURCE PRIMARY "CHIP"  
...  
...  
INCLUDE "/calibre/LVS/Calibre-lvs-cur"
```

- ◆ Edit Calibre LVS rule file

```
...  
...  
LVS BOX PVSSC  
LVS BOX PVSSR  
LVS BOX DRAM64x4s
```

Submit Calibre LVS

- *calibre -lvs -spice layout.spi -hier -auto Calibre.lvs > lvs.log*



Check Calibre LVSSummary

OVERALL COMPAISON RESULTS

OVERALL COMPARISON RESULTS

```
      # #####
      # #                               # * *
#  #  # CORRECT  # |
      # #        # \_/_/
      # #####
```



PRACTICE