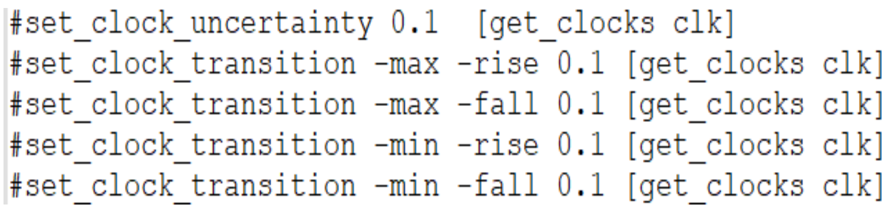
NCTU-EE IC LAB – Spring 2024

Lab12 Exercise: Innovus - From RTL to GDSII

Due to the main goal of this course is to teach students on the front-end design methodology, the APR of final project will be modified to pin design rather than pad design so it's easier.

1. Data Preparation

1. Complete rtl simulation, synthesis, and gate level simulation.
2. Change the directory to **05_APR**
3. Prepare chip design netlist:
   - **a.** Open **CHIP_SHELL.v** and **modify it** to your design.
     - ➢ The top module name is **CHIP**.
     - ➢ This **CHIP** contains the module **CAD;** and **I/O**, **I/O power**, **core power** pad.
     - ➢ Please calculate how many I/O and core power pads you need, and complete the netlist of pad cells (**size** and **action**).
     - ➢ After defining the pad cells, please run %**./00_combine** to combine the CAD_SYN.v (copy from 02_SYN/Netlist to here) with CHIP_SHELL.v to be CHIP_SYN.v
4. Prepare I/O pad location file:
   - **a.** Please see **CHIP.io** to know how to assign I/O pad location.
   - **b.** Open **CHIP.io** and complete the I/O pad location assignment according to the netlist of pad cells in **CHIP_SYN.v** (**size** and **action).**
5. Prepare timing/driving/loading constraint file **CHIP.sdc**:
   - **a.** cp ../02_SYN/Netlist/CAD_SYN.sdc **CHIP.sdc**:
   - **b.** Comment out the **set_clock_uncertainty** & **set_clock_transition** instruction in **CHIP.sdc**



```
#set_clock_uncertainty 0.1  [get_clocks clk]
#set_clock_transition -max -rise 0.1 [get_clocks clk]
#set_clock_transition -max -fall 0.1 [get_clocks clk]
#set_clock_transition -min -rise 0.1 [get_clocks clk]
#set_clock_transition -min -fall 0.1 [get_clocks clk]
```

    c. Modify the contents to the desired constraint

6. Prepare linked library files:

    a. **Timing libraries (LIB)**

        - **fsa0m_a_generic_core_ss1p62v125c.lib**
        - **fsa0m_a_generic_core_ff1p98vm40c.lib**
        - **fsa0m_a_t33_generic_io_ss1p62v125c.lib**
        - **fsa0m_a_t33_generic_io_ff1p98vm40c.lib**
        - **Memory_WC.lib**
        - **Memory_BC.lib**

    b. **Physical libraries (LEF)**

        - **header6_V55_20ka_cic.lef**
        - **fsa0m_a_generic_core.lef**
        - **FSA0M_A_GENERIC_CORE_ANT_V55.lef**
        - **fsa0m_a_t33_generic_io.lef**
        - **FSA0M_A_T33_GENERIC_IO_ANT_V55.lef**
        - **BONDPAD.lef**
        - **Memory.lef**

    c. **RC extraction table/files**

        - **u18_Faraday.CapTbl**
        - **icecaps.tch**

    d. **CeltIC libraries**

        - **u18_ss.cdb**
        - **u18_ff.cdb**

    e. **GDSII layout** (Will not stream out in this course)

        - **u18.gds2**
        - **u18io3v5v_6lm.gds2**
        - **Memory.gds2** (not provided in this compiler version)

2. Reading Cell Library information and Netlist for APR

1. If you are running on ee servers, type below command on terminal to set

OA_HOME.

$ setenv OA_HOME /RAID2/cad/cadence/INNOVUS/INNOVUS_20.15.000/share/oa

2. Start Innovus in the directory **05_APR**

   % innovus (no &, do NOT use background execution)

3. Set uniquify global variable to 1

   % set init_design_uniquify 1

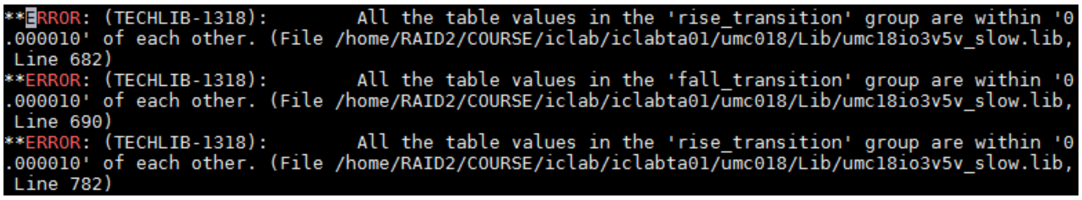
4. Change design mode to 0.18um process

   % setDesignMode -process 180

5. Skip error messages like the following figure when reading lef files

   % suppressMessage TECHLIB 1318

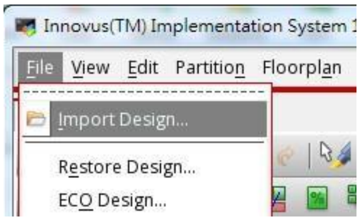
   % suppressMessage ENCEXT-2799



```
**ERROR: (TECHLIB-1318):        All the table values in the 'rise_transition' group are within '0
.000010' of each other. (File /home/RAID2/COURSE/iclab/iclabta01/umc018/Lib/umc18io3v5v_slow.lib,
 Line 682)
**ERROR: (TECHLIB-1318):        All the table values in the 'fall_transition' group are within '0
.000010' of each other. (File /home/RAID2/COURSE/iclab/iclabta01/umc018/Lib/umc18io3v5v_slow.lib,
 Line 690)
**ERROR: (TECHLIB-1318):        All the table values in the 'rise_transition' group are within '0
.000010' of each other. (File /home/RAID2/COURSE/iclab/iclabta01/umc018/Lib/umc18io3v5v_slow.lib,
 Line 782)
```

This error message occurs since innovus assumes the values of rise / fall transition time should differ larger than 0.00001 ns in **the table** depending on the supply Voltage or the Temperature **in .lef file**. The error message can be ignored if the given .lef file is ensured to be correct. (Note: only suppress error when the reason is clearly verified instead of suppressing all errors)

6. In innovus menu, open ***File → Import Design***



7. Fill the following field:

   a. Netlist

   ◆ Verilog

   ➢ Files: **CHIP_SYN.v**

   ➢ Top Cell: ◆ By User : **CHIP**

   b. Technology / Physical Libraries

   ➢ LEF Files: **header6_V55_20ka_cic.lef**

fsa0m_a_generic_core.lef

FSA0M_A_GENERIC_CORE_ANT_V55.lef

fsa0m_a_t33_generic_io.lef

FSA0M_A_T33_GENERIC_IO_ANT_V55.lef

BONDPAD.lef

Memory.lef

Note that (1) the standard cell lef file should be put in the first place since in contains the major metals / vias / polys technology definitions. Also (2) if there are antenna lef files, they should be put after the corresponded lef files without describing antenna layers. For example: umc18_6lm.lef should be put in the first place; umc18_6lm_antenna.lef should be put after umc18_6lm.lef.

c. Floorplan

- IO Assigment Files: **CHIP.io**

d. Power

- Power Nets: **VCC**
- Ground Nets: **GND**

e. Analysis Configuration

Press "Create Analysis Configuration" tab

- Double click **Library Sets** and include the max and min delay library :

Max delay:

Name: lib_max

Timing Library:

fsa0m_a_generic_core_ss1p62v125c.lib

fsa0m_a_t33_generic_io_ss1p62v125c.lib

Memory_WC.lib

SI Library: **u18_ss.cdb**

Min delay:

Name: lib_min

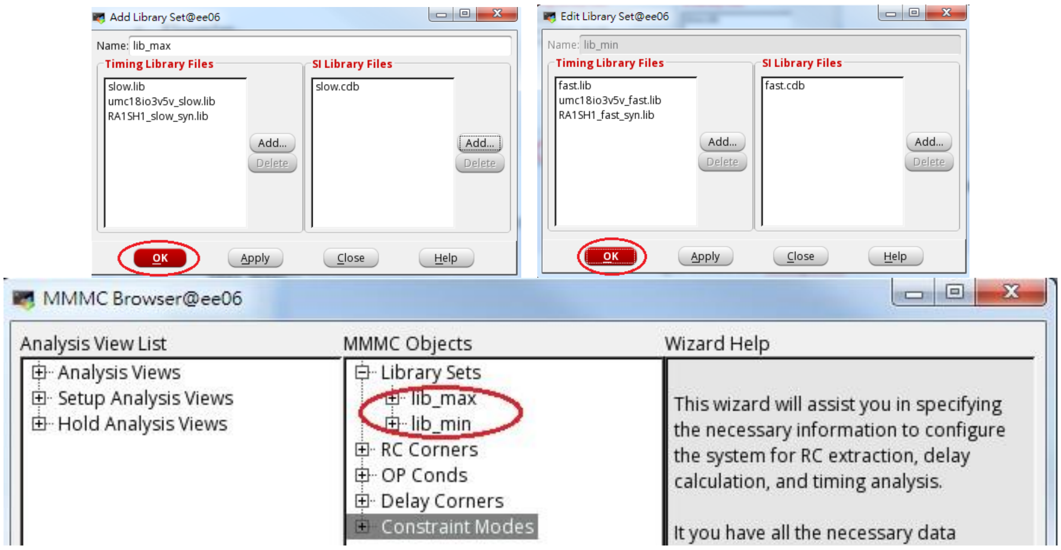
Timing Library:

fsa0m_a_generic_core_ff1p98vm40c.lib

fsa0m_a_t33_generic_io_ff1p98vm40c.lib

Memory_BC.lib

SI Library: **u18_ff.cdb**



Add Library Set@ee06

Name: lib_max

Timing Library Files:
slow.lib
umc18io3v5v_slow.lib
RA1SH1_slow_syn.lib

SI Library Files:
slow.cdb

Edit Library Set@ee06

Name: lib_min

Timing Library Files:
fast.lib
umc18io3v5v_fast.lib
RA1SH1_fast_syn.lib

SI Library Files:
fast.cdb

MMMC Browser@ee06

Analysis View List
- Analysis Views
- Setup Analysis Views
- Hold Analysis Views

MMMC Objects
- Library Sets
  - lib_max
  - lib_min
- RC Corners
- OP Conds
- Delay Corners
- Constraint Modes

Wizard Help

This wizard will assist you in specifying the necessary information to configure the system for RC extraction, delay calculation, and timing analysis.

It you have all the necessary data

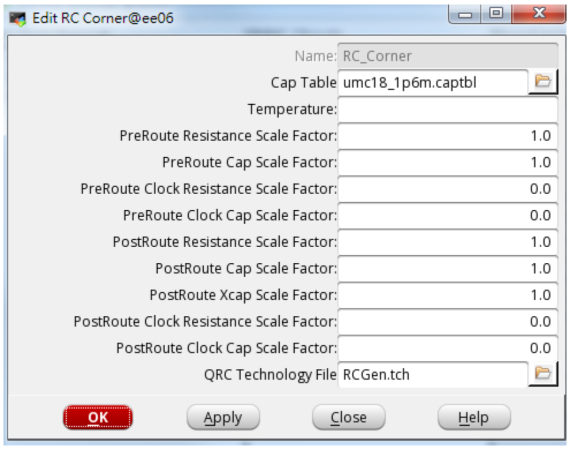
- ➢ Double click **RC Corners** to include the RC corner library :

Name: RC_Corner

Cap Table:

u18_Faraday.CapTbl

QRC Tech File: **icecaps.tch**



Edit RC Corner@ee06

Name: RC_Corner
Cap Table: umc18_1p6m.captbl
Temperature:
PreRoute Resistance Scale Factor: 1.0
PreRoute Cap Scale Factor: 1.0
PreRoute Clock Resistance Scale Factor: 0.0
PreRoute Clock Cap Scale Factor: 0.0
PostRoute Resistance Scale Factor: 1.0
PostRoute Cap Scale Factor: 1.0
PostRoute Xcap Scale Factor: 1.0
PostRoute Clock Resistance Scale Factor: 0.0
PostRoute Clock Cap Scale Factor: 0.0
QRC Technology File: RCGen.tch

- ➢ Double click **Delay Corners** and create max and min delay constraints :

Max delay:

Name: Delay_Corner_max

RC Corner: RC_Corner

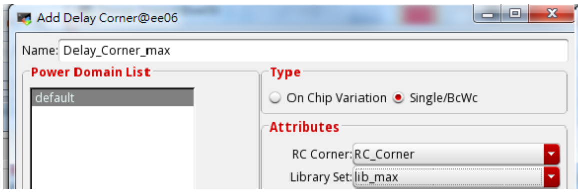
Lib Set: lib_max

Min delay:

Name: Delay_Corner_min

RC Corner: RC_Corner

Lib Set: lib_min


Add Delay Corner@ee06

Name: Delay_Corner_max

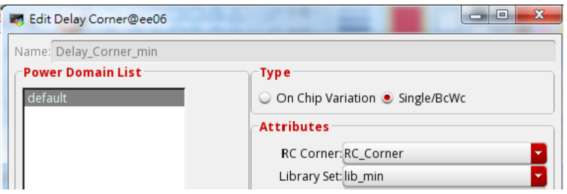
Power Domain List: default

Type: ○ On Chip Variation ● Single/BcWc

Attributes

RC Corner: RC_Corner

Library Set: lib_max


Edit Delay Corner@ee06

Name: Delay_Corner_min

Power Domain List: default

Type: ○ On Chip Variation ● Single/BcWc

Attributes

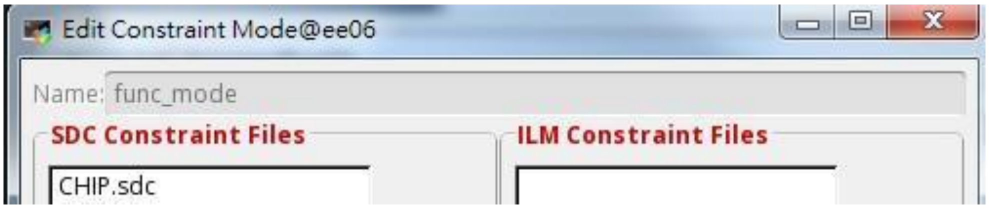
RC Corner: RC_Corner

Library Set: lib_min

- ➢ Double click **Constraints Mode** and create a function mode to place CHIP_SYC.sdc :

Name: func_mode

SDC Constraint Files: CHIP.sdc


Edit Constraint Mode@ee06

Name: func_mode

SDC Constraint Files

CHIP.sdc

ILM Constraint Files

- ➢ Double click **Analysis Views** to create max and min delay analysis

Max delay:

Name: av_func_mode_max

Constraint Mode: func_mode

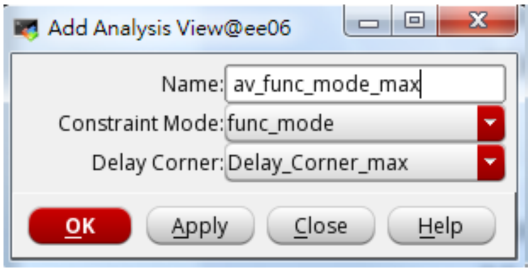
Delay Corner: Delay_Corner_max

Min delay:

Name: av_func_mode_min

Constraint Mode: func_mode

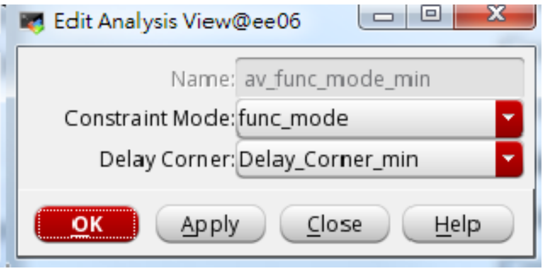
Delay Corner: Delay_Corner_min


Add Analysis View@ee06

Name: av_func_mode_max

Constraint Mode: func_mode

Delay Corner: Delay_Corner_max


Edit Analysis View@ee06

Name: av_func_mode_min

Constraint Mode: func_mode

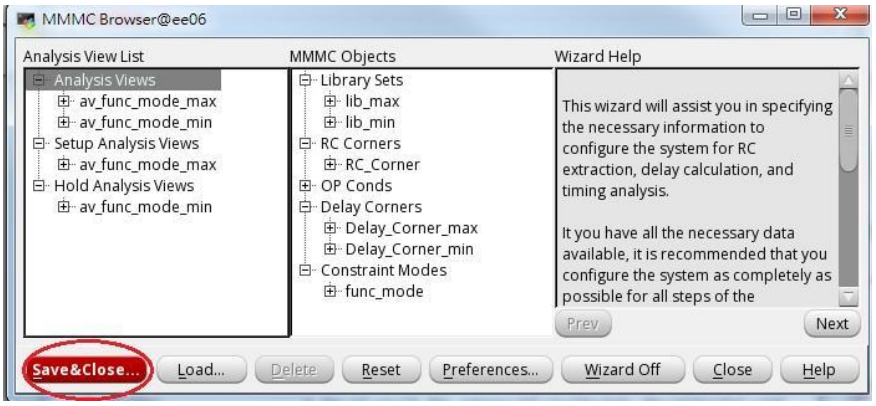
Delay Corner: Delay_Corner_min

- Click **Setup Analysis View** and specify the max analysis mode

  Choose: av_func_mode_max

- Click **Hold Analysis View** and specify the min analysis mode

  Choose: av_func_mode_min


MMMC Browser@ee06

Analysis View List
- Analysis Views
  - av_func_mode_max
  - av_func_mode_min
- Setup Analysis Views
  - av_func_mode_max
- Hold Analysis Views
  - av_func_mode_min

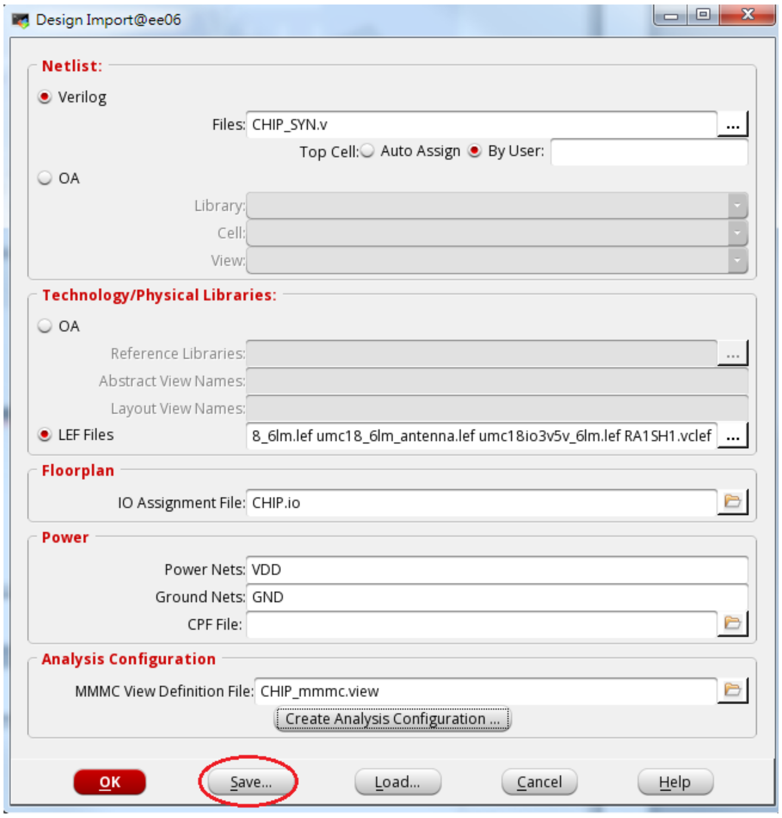
MMMC Objects
- Library Sets
  - lib_max
  - lib_min
- RC Corners
  - RC_Corner
- OP Conds
- Delay Corners
  - Delay_Corner_max
  - Delay_Corner_min
- Constraint Modes
  - func_mode

Wizard Help

This wizard will assist you in specifying the necessary information to configure the system for RC extraction, delay calculation, and timing analysis.

It you have all the necessary data available, it is recommended that you configure the system as completely as possible for all steps of the

- ➢ Save as “CHIP_mmmc.view”



Design Import@ee06

Netlist:

● Verilog

Files: CHIP_SYN.v

Top Cell: ○ Auto Assign ● By User:

○ OA

Library:

Cell:

View:

Technology/Physical Libraries:

○ OA

Reference Libraries:

Abstract View Names:

Layout View Names:

● LEF Files 8_6lm.lef umc18_6lm_antenna.lef umc18io3v5v_6lm.lef RA1SH1.vclef

Floorplan

IO Assignment File: CHIP.io

Power

Power Nets: VDD

Ground Nets: GND

CPF File:

Analysis Configuration

MMMC View Definition File: CHIP_mmmc.view

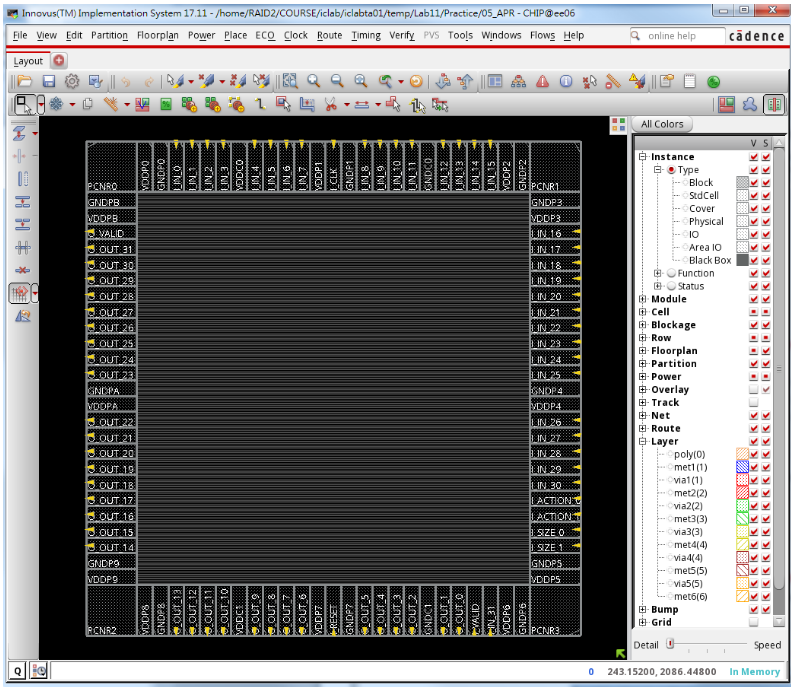
8. Save current settings:

   a. Click **Save…** button

   - ➢ File name: **CHIP.globals**

   If you want to reload the settings, you can just press Load… button

9. Click **OK** button


Innovus(TM) Implementation System 17.11 - /home/RAID2/COURSE/iclab/iclabta01/temp/Lab11/Practice/05_APR - CHIP@ee06
File
View
Edit
Partition
Floorplan
Power
Place
ECO
Clock
Route
Timing
Verify
PVS
Tools
Windows
Flows
Help
online help
cādence
Layout
PCNR0
VDDP0
GNDP0
I_IN_0
I_IN_1
I_IN_2
I_IN_3
VDDC0
I_IN_4
I_IN_5
I_IN_6
I_IN_7
VDDP1
I_CLK
GNDP1
I_IN_8
I_IN_9
I_IN_10
I_IN_11
GNDC0
I_IN_12
I_IN_13
I_IN_14
I_IN_15
VDDP2
GNDP2
PCNR1
GNDPB
VDDPB
O_VALID
O_OUT_31
O_OUT_30
O_OUT_29
O_OUT_28
O_OUT_27
O_OUT_26
O_OUT_25
O_OUT_24
O_OUT_23
GNDPA
VDDPA
O_OUT_22
O_OUT_21
O_OUT_20
O_OUT_19
O_OUT_18
O_OUT_17
O_OUT_16
O_OUT_15
O_OUT_14
GNDP9
VDDP9
GNDP3
VDDP3
I_IN_16
I_IN_17
I_IN_18
I_IN_19
I_IN_20
I_IN_21
I_IN_22
I_IN_23
I_IN_24
I_IN_25
GNDP4
VDDP4
I_IN_26
I_IN_27
I_IN_28
I_IN_29
I_IN_30
I_ACTION_0
I_ACTION_1
I_SIZE_0
I_SIZE_1
GNDP5
VDDP5
PCNR2
VDDP8
GNDP8
O_OUT_13
O_OUT_12
O_OUT_11
O_OUT_10
VDDC1
O_OUT_9
O_OUT_8
O_OUT_7
O_OUT_6
VDDP7
I_RESET
GNDP7
O_OUT_5
O_OUT_4
O_OUT_3
O_OUT_2
GNDC1
O_OUT_1
O_OUT_0
I_VALID
I_IN_31
VDDP6
GNDP6
PCNR3
All Colors
V
S
Instance
Type
Block
StdCell
Cover
Physical
IO
Area IO
Black Box
Function
Status
Module
Cell
Blockage
Row
Floorplan
Partition
Power
Overlay
Track
Net
Route
Layer
poly(0)
met1(1)
via1(1)
met2(2)
via2(2)
met3(3)
via3(3)
met4(4)
via4(4)
met5(5)
via5(5)
met6(6)
Bump
Grid
Detail
Speed
Q
0
243.15200, 2086.44800
In Memory

3. Specify Chip Floorplan

1. In innovus menu, open ***Floorplan → Specify Floorplan***

2. Specify core size:

   a. Core Utilization: Set any as your wish (0.7~0.8 is suggested)

3. Specify core margin:

   b. ◆ Core to IO Boundary

   Core to Left: **250**

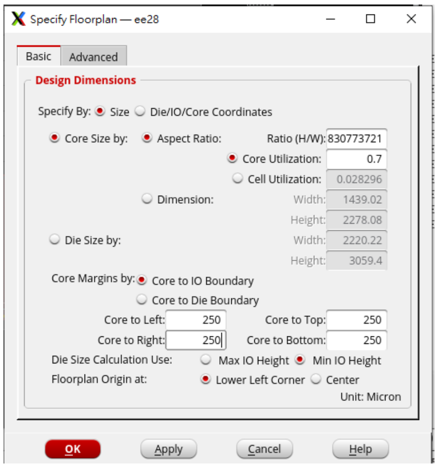
   Core to Right: 250

   Core to Top: 250

   Core to Bottom: 250

4. Apply the specification:

   c. Click **OK** button


Specify Floorplan — ee28

Basic | Advanced

Design Dimensions

Specify By: ● Size ○ Die/IO/Core Coordinates

● Core Size by: ● Aspect Ratio: Ratio (H/W): 830773721

● Core Utilization: 0.7

○ Cell Utilization: 0.028296

○ Dimension: Width: 1439.02

Height: 2278.08

○ Die Size by: Width: 2220.22

Height: 3059.4

Core Margins by: ● Core to IO Boundary

○ Core to Die Boundary

Core to Left: 250 Core to Top: 250

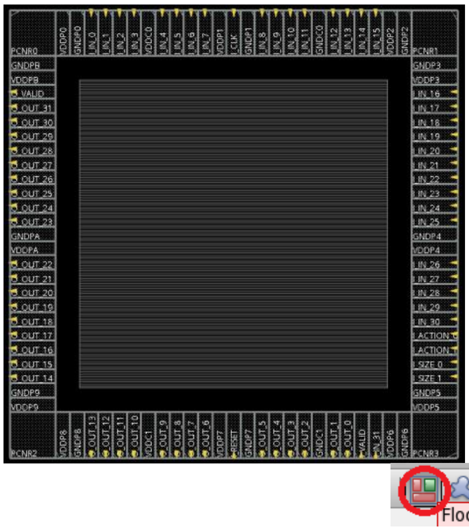
Core to Right: 250 Core to Bottom: 250

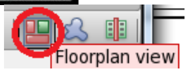
Die Size Calculation Use: ○ Max IO Height ● Min IO Height

Floorplan Origin at: ● Lower Left Corner ○ Center

Unit: Micron

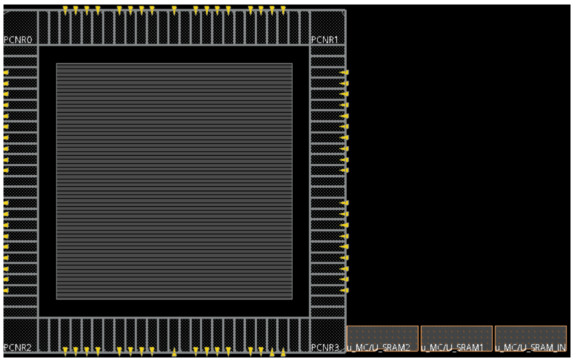
OK | Apply | Cancel | Help



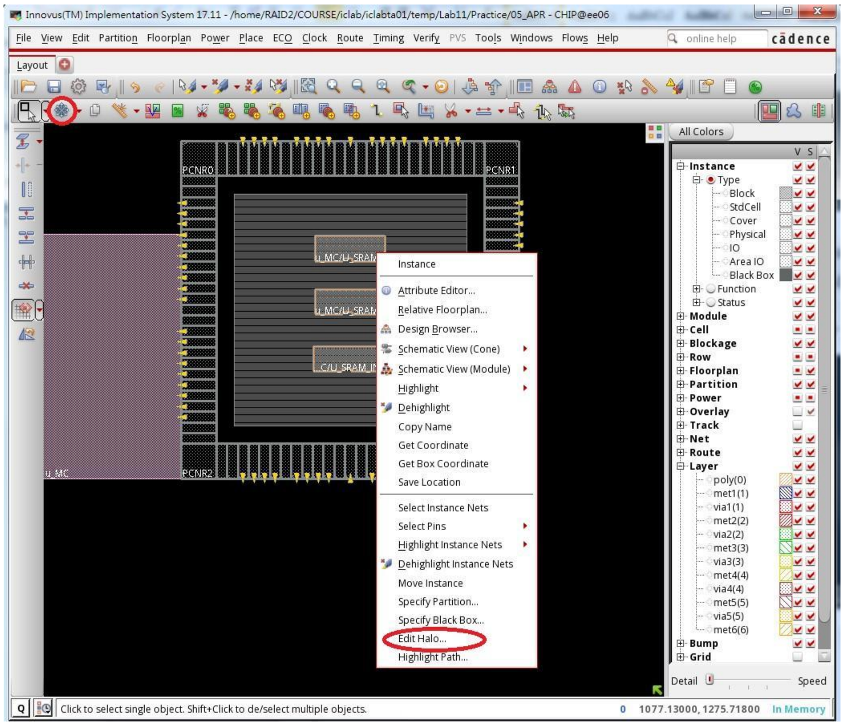


5. You can view designs in three ways; change to floorplan view:

   The memory macro should appear (Besides zoom in and zoom out button 🔍 🔍 🔍, you can use right click to zoom the view)



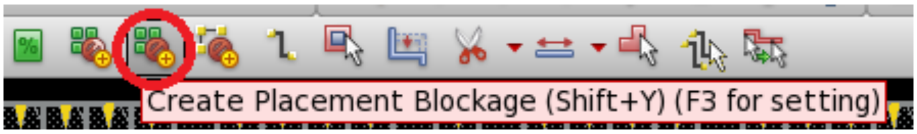
6. Move them to the desired places and make placement blockage:
   right click on one macro, Choose “**Edit Halo**”



7. Specify Halo For: ◆ All Macros

   Add/Update Halo: Top, Bottom, Left, Right: 15um

8. You can also create the placement blockage yourself with the button:



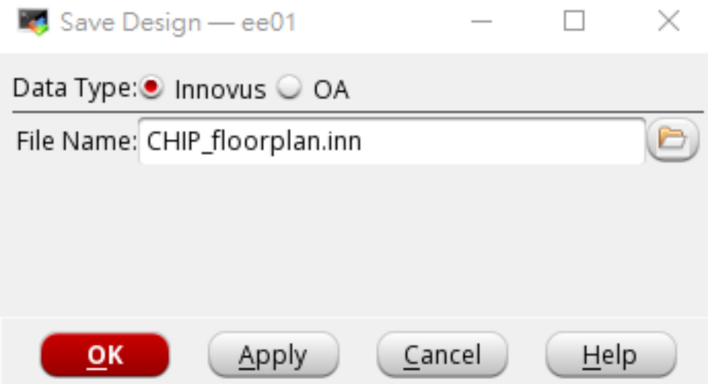
9. Save the floorplan design:

   a. File → Save Design

   b. Data Type ◆ Innovus

   File Name: CHIP_floorplan.inn

   c. Click **OK** button



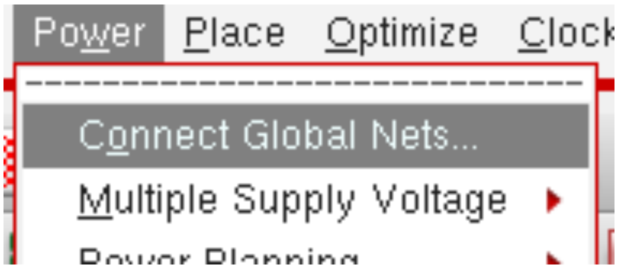
Data Type: ● Innovus ○ OA

File Name: CHIP_floorplan.inn

If you do the wrong things in the following steps such as power planning, you can restore the design from the previous steps. However if the input files such as verilog file / io file / timing constraints are changed, you have to rerun all the APR steps.

4. Connect/Define Global Net

1. In innovus menu, open **Power → Connect Global Net**



2. Add all VDD pins to Connection List:

   a. Connect ◆ Pin

   Instance Basename: *

   Pin Name(s): **VCC**

   b. Scope ◆ Apply All

   c. To Global Nets: VCC

   d. Click Add to List button

3. Add all VDD nets to Connection List:

   a. Connect ◆ Net Basename: **VCC**

   b. Scope ◆ Apply All

   c. To Global Nets: VCC

   d. Click Add to List button

4. Add all Tie High pins to Connection List:

   a. Connect ◆ Tie High

   b. Scope ◆ Apply All

   c. To Global Nets: VCC

   d. Click Add to List button

5. Add all GND pins to Connection List:

   a. Connect ◆ Pin

   Instance Basename: *

   Pin Name(s): **GND**

   b. Scope ◆ Apply All

   c. To Global Nets: GND

   d. Click Add to List button

6. Add all GND nets to Connection List:

   a. Connect ◆ Net Basename: **GND**

   b. Scope ◆ Apply All

   c. To Global Nets: GND

   d. Click Add to List button

7. Add all Tie Low pins to Connection List:

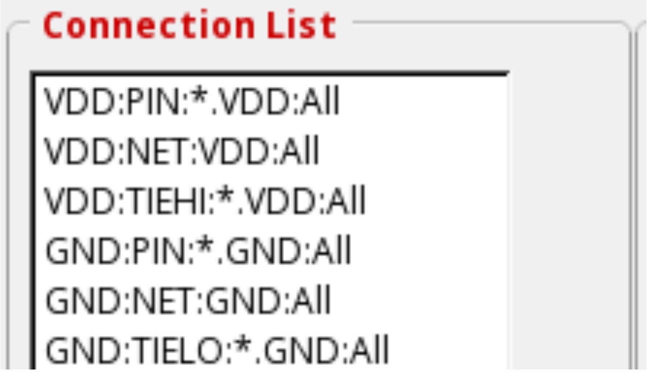
    a. Connect ◆ Tie Low

    b. Scope ◆ Apply All

    c. To Global Nets: GND

d. Click Add to List button

e. Click Add to List button



Connection List

VDD:PIN:*.VDD:All
VDD:NET:VDD:All
VDD:TIEHI:*.VDD:All
GND:PIN:*.GND:All
GND:NET:GND:All
GND:TIELO:*.GND:All

8. Apply the connection list and check:

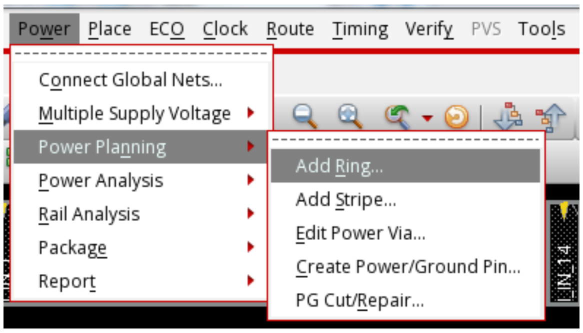
e. Click **Apply** button

f. Click **Check** button

g. Click **Cancel** button

5. Power Planning (Add Core Power Rings)

1. In innovus menu, open ***Power → Power Planning → Add Rings…***



2. In the **Basic** tab, fill the following field:

a. Net(s): GND VCC

b. Ring Type: Core ring(s) contouring

c. Specify metal layers and width

➢ Top Layer: **metal3 H** Width: **9**

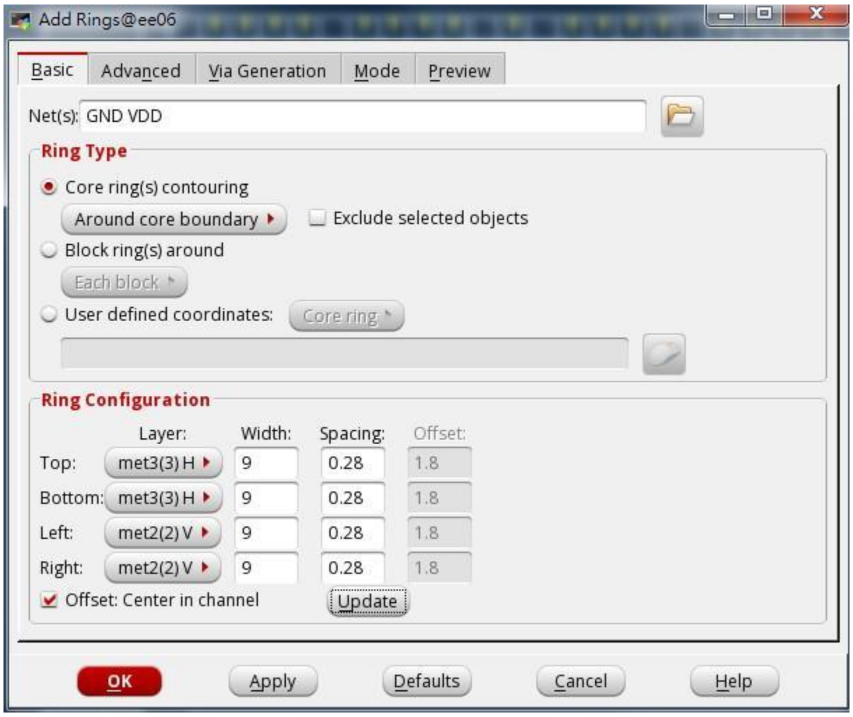
➢ Bottom Layer: **metal3 H** Width: **9**

➢ Left Layer: **metal2 V** Width: **9**

➢ Right Layer: **metal2 V** Width: **9**

➢ ◆ Offset: Center in channel

➢ Click **Update** button



Add Rings@ee06

Basic | Advanced | Via Generation | Mode | Preview

Net(s): GND VDD

Ring Type

● Core ring(s) contouring

Around core boundary ▸ □ Exclude selected objects

○ Block ring(s) around

Each block ▸

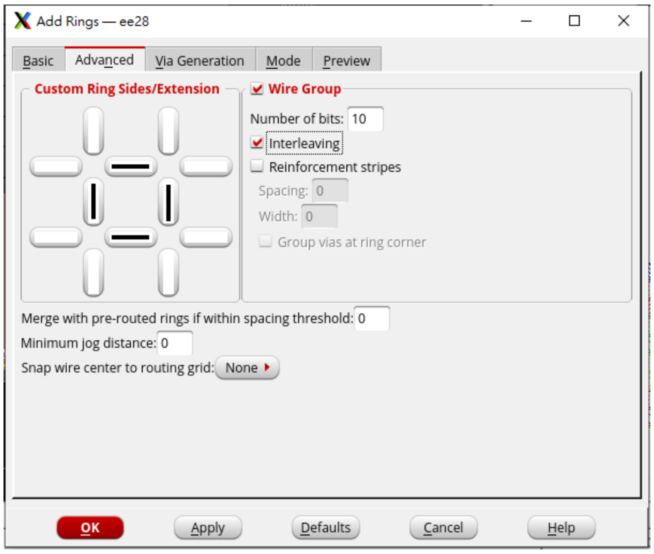
○ User defined coordinates: Core ring ▸

Ring Configuration

	Layer:	Width:	Spacing:	Offset:
Top:	met3(3) H ▸	9	0.28	1.8
Bottom:	met3(3) H ▸	9	0.28	1.8
Left:	met2(2) V ▸	9	0.28	1.8
Right:	met2(2) V ▸	9	0.28	1.8

☑ Offset: Center in channel Update

3. In the **Advanced** tab, fill the following field:
   - **a.** ◆ Wire Group
   - **b.** Number of bits: 10
   - **c.** ◆ Interleaving
4. Click **OK** button



Add Rings — ee28

Basic | Advanced | Via Generation | Mode | Preview

Custom Ring Sides/Extension

☑ Wire Group

Number of bits: 10

☑ Interleaving

□ Reinforcement stripes

Spacing: 0

Width: 0

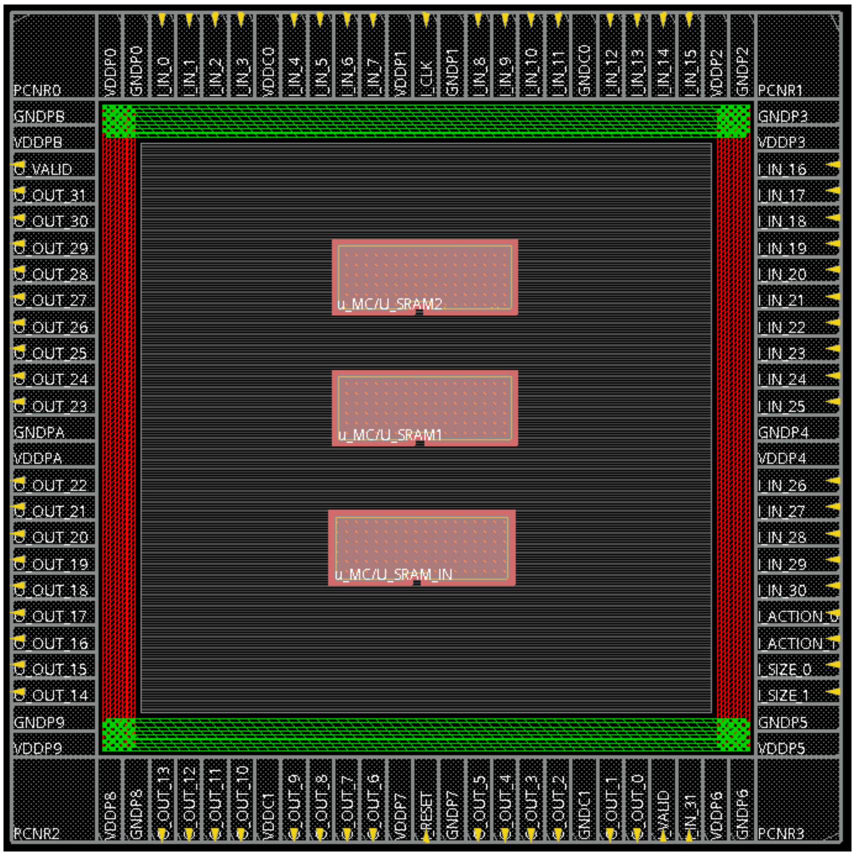
□ Group vias at ring corner

Merge with pre-routed rings if within spacing threshold: 0

Minimum jog distance: 0

Snap wire center to routing grid: None ▸

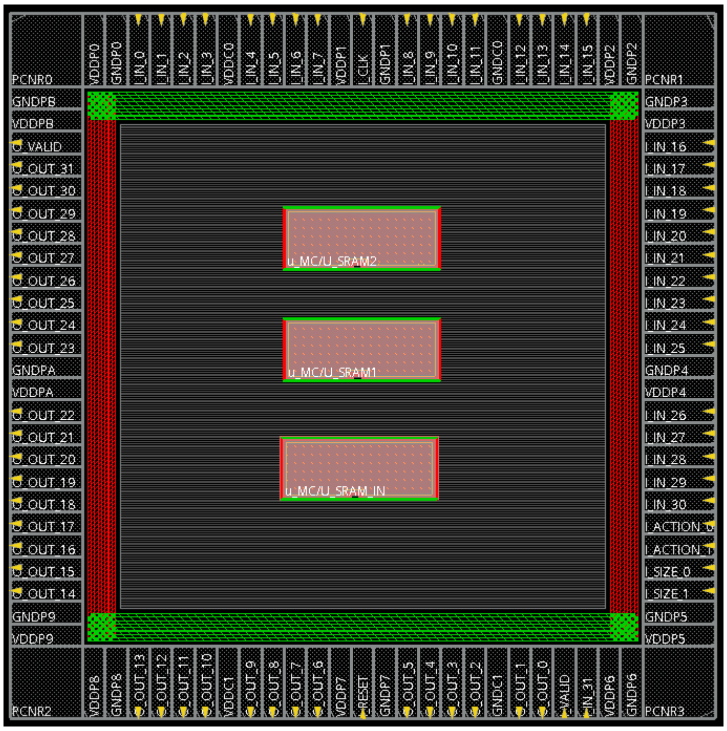
5. Check if the ring is correctly created. If not, click **undo** button ↶ and repeat step 2~4 again.



6. Power Planning (Add Block Rings)

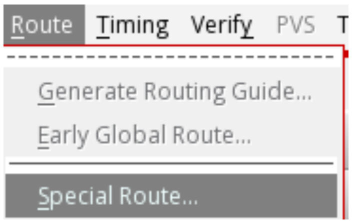
1. In innovus menu, open ***Power → Power Planning → Add Rings…***

2. In the **Basic** tab, fill the following field:
   a. Net(s): GND VCC
   b. Ring Type: Block ring(s) around
   c. Specify metal layers and width
   - ➢ Top Layer: **met3(3) H** Width: **2**
   - ➢ Bottom Layer: **met3(3) H** Width: **2**
   - ➢ Left Layer: **met2(2) V** Width: **2**
   - ➢ Right Layer: **met2(2) V** Width: **2**
   - ➢ ◇ Offset: Center in channel
   - ➢ Click **Update** button
3. In the **Advanced** tab, disable the wire group:
   a. ◇ Wire Group
4. Click **OK** button



7. Connect Core Power Pin

1. In innovus menu, open ***Route → Special Route…***



2. Connect core power:

In **Basic** tab,

a. Net(s): GND VCC

b. Set the following configuration

SRoute:

➢ ◆ Block pins

➢ ◇ Pad rings

➢ ◇ Floating Stripes

➢ ◆ Pad pins

➢ ◇ Follow pins

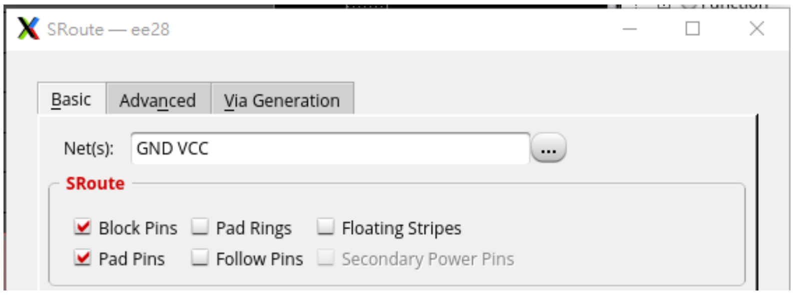
In **Via Generation** tab,

c. Set the following configuration

➢ ◆ Core Ring

- ➢ ◆ Block Ring

d. Click **OK** button



SRoute — ee28

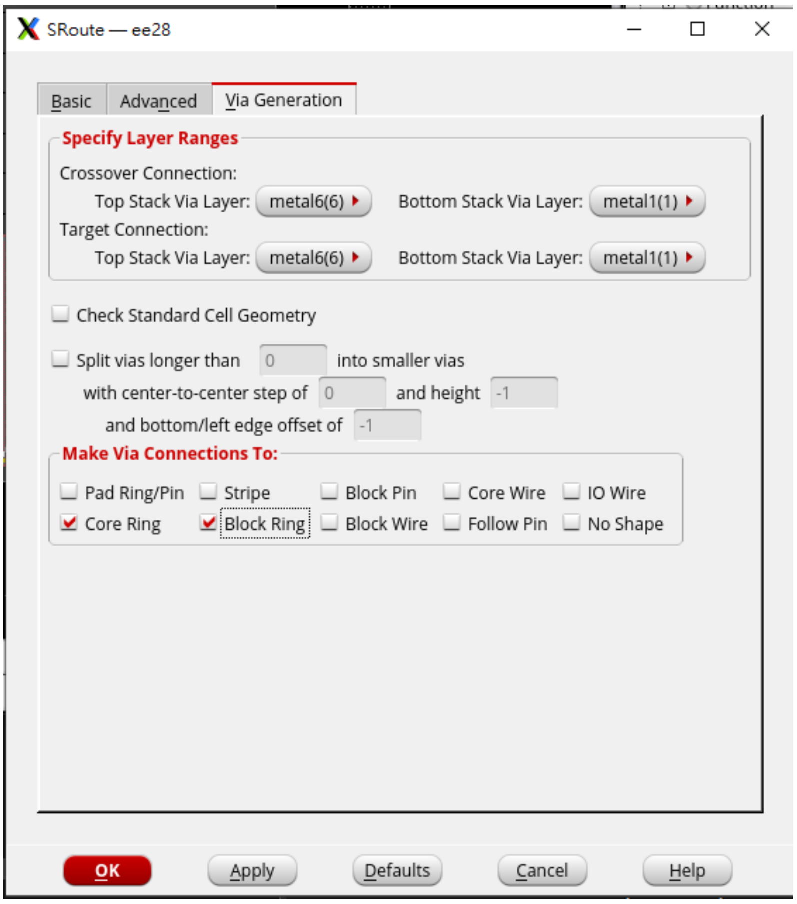
Basic | Advanced | Via Generation

Net(s): GND VCC

SRoute

☑ Block Pins ☐ Pad Rings ☐ Floating Stripes

☑ Pad Pins ☐ Follow Pins ☐ Secondary Power Pins



SRoute — ee28

Basic | Advanced | Via Generation

Specify Layer Ranges

Crossover Connection:

Top Stack Via Layer: metal6(6) Bottom Stack Via Layer: metal1(1)

Target Connection:

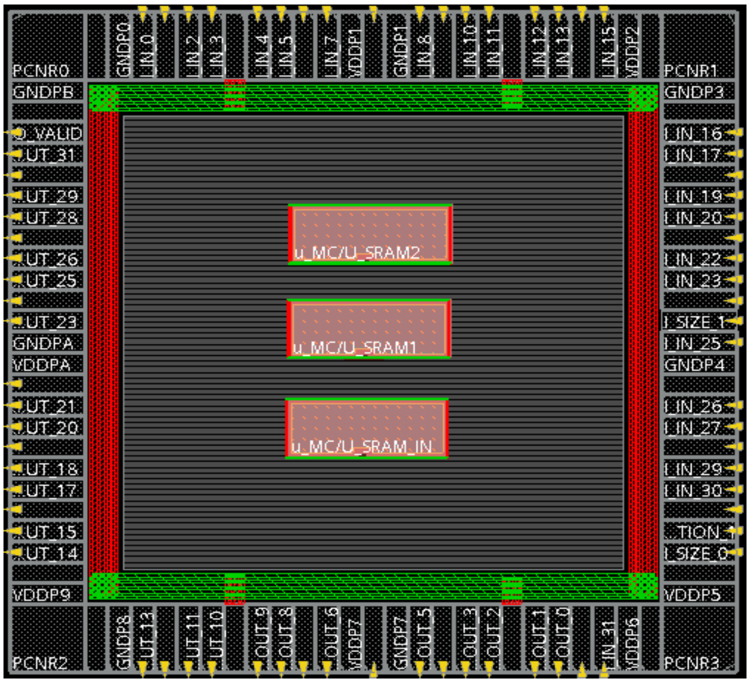
Top Stack Via Layer: metal6(6) Bottom Stack Via Layer: metal1(1)

□ Check Standard Cell Geometry

□ Split vias longer than 0 into smaller vias
with center-to-center step of 0 and height -1
and bottom/left edge offset of -1

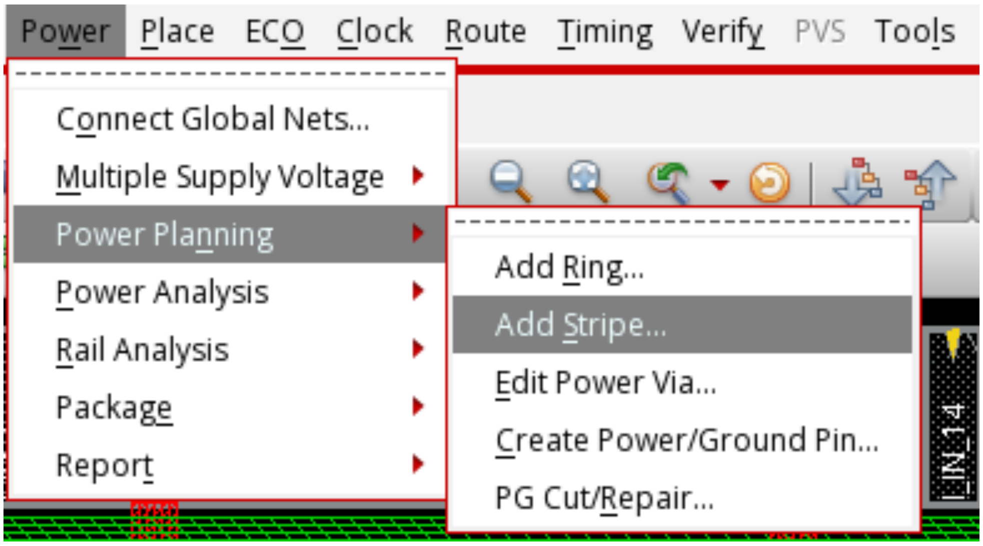
Make Via Connections To:

□ Pad Ring/Pin □ Stripe □ Block Pin □ Core Wire □ IO Wire
☑ Core Ring ☑ Block Ring □ Block Wire □ Follow Pin □ No Shape



8. Power Planning (Add Stripes)

1. In innovus menu, open ***Power → Power Planning → Add Stripes…***



2. Create vertical power stripes:

   a. Set Configuration

   - ➢ Net(s): GND VCC
   - ➢ Layer: **met2(2)**
   - ➢ Directions: ◆ Vertical
   - ➢ Width: **4**
   - ➢ Click **Update** Button

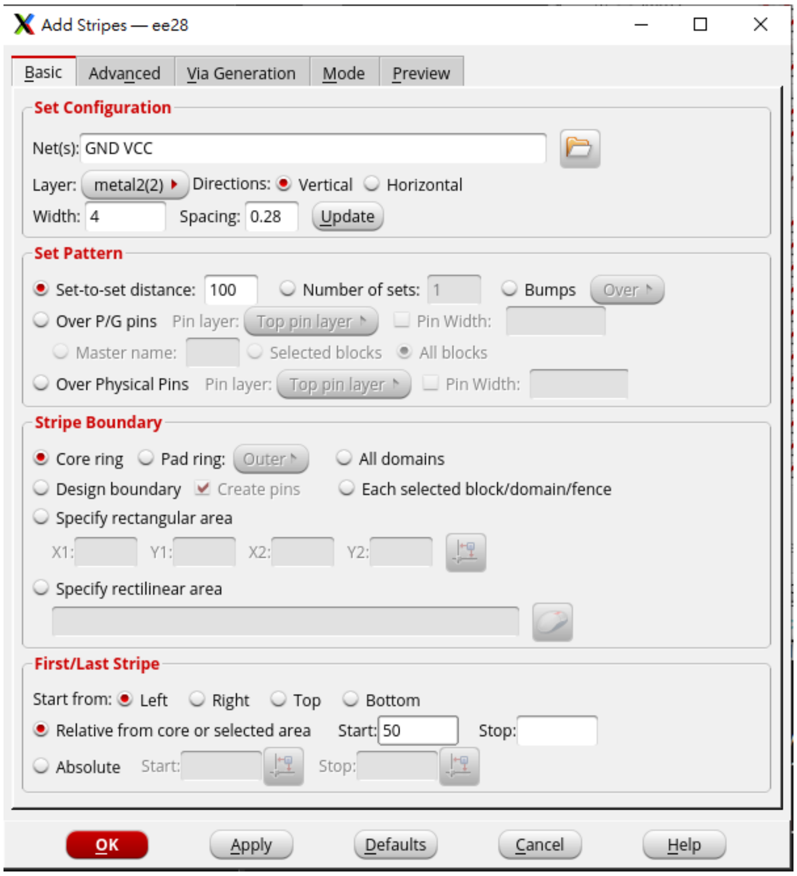
   b. Set Pattern

   - ➢ ◆ Set-to-set distance: **100**

   c. First/Last Stripe

   - ➢ Start from: ◆ Left
   - ➢ ◆ Relative from core or selected area
   - ➢ Start: 50

   d. Click **OK** button



Add Stripes — ee28

Basic | Advanced | Via Generation | Mode | Preview

Set Configuration

Net(s): GND VCC

Layer: metal2(2) Directions: ● Vertical ○ Horizontal

Width: 4 Spacing: 0.28 Update

Set Pattern

● Set-to-set distance: 100 ○ Number of sets: 1 ○ Bumps Over

○ Over P/G pins Pin layer: Top pin layer □ Pin Width:

○ Master name: ○ Selected blocks ● All blocks

○ Over Physical Pins Pin layer: Top pin layer □ Pin Width:

Stripe Boundary

● Core ring ○ Pad ring: Outer ○ All domains

○ Design boundary ☑ Create pins ○ Each selected block/domain/fence

○ Specify rectangular area

X1: Y1: X2: Y2:

○ Specify rectilinear area

First/Last Stripe

Start from: ● Left ○ Right ○ Top ○ Bottom

● Relative from core or selected area Start: 50 Stop:

○ Absolute Start: Stop:

3. Check if the stripes are correctly created. If not, click **undo** button ↶ and

repeat step a~d again.

4. Create horizontal stripes:

   a. Set Configuration

   - Net(s): GND VCC
   - Layer: **met3(3)**
   - Directions: ◆ Horizontal
   - Width: **4**
   - Click **Update** Button

   b. Set Pattern

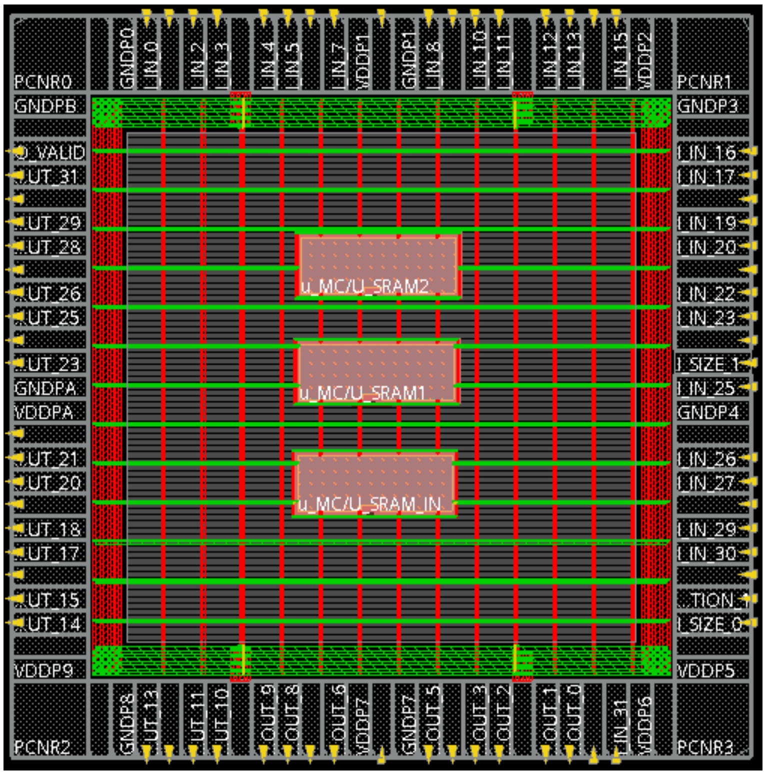
   - ◆ Set-to-set distance: **100**

   c. First/Last Stripe

   - Start from: ◆ Bottom
   - ◆ Relative from core or selected area
   - Start: 50

   d. Click **OK** button

5. Check if the stripes are correctly created. If not, click **undo** button ↶ and repeat step a~d again.



9. Connect Standard Cell Power Line

1. Make sure hard macro blockage has been added in floorplan.
2. In innovus menu, open ***Route → Special Route…***
3. Connect core power:

   In **Basic** tab,

   a. Net(s): GND VCC

   b. Set the following configuration

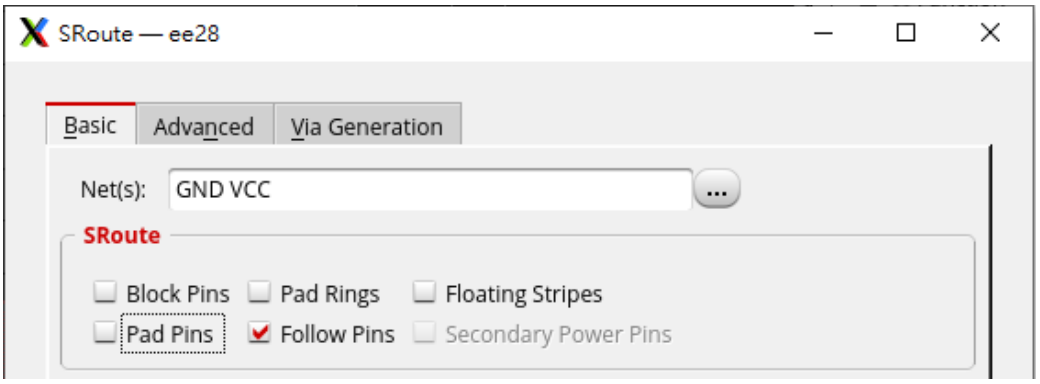
   - ➢ ◇ Block pins
   - ➢ ◇ Pad rings
   - ➢ ◇ Floating Stripes
   - ➢ ◇ Pad pins
   - ➢ ◆ Follow pins

   In **Via Generation** tab,

   c. Set the following configuration

   - ➢ ◆ Stripe
   - ➢ ◆ Core Ring
   - ➢ ◆ Block Ring

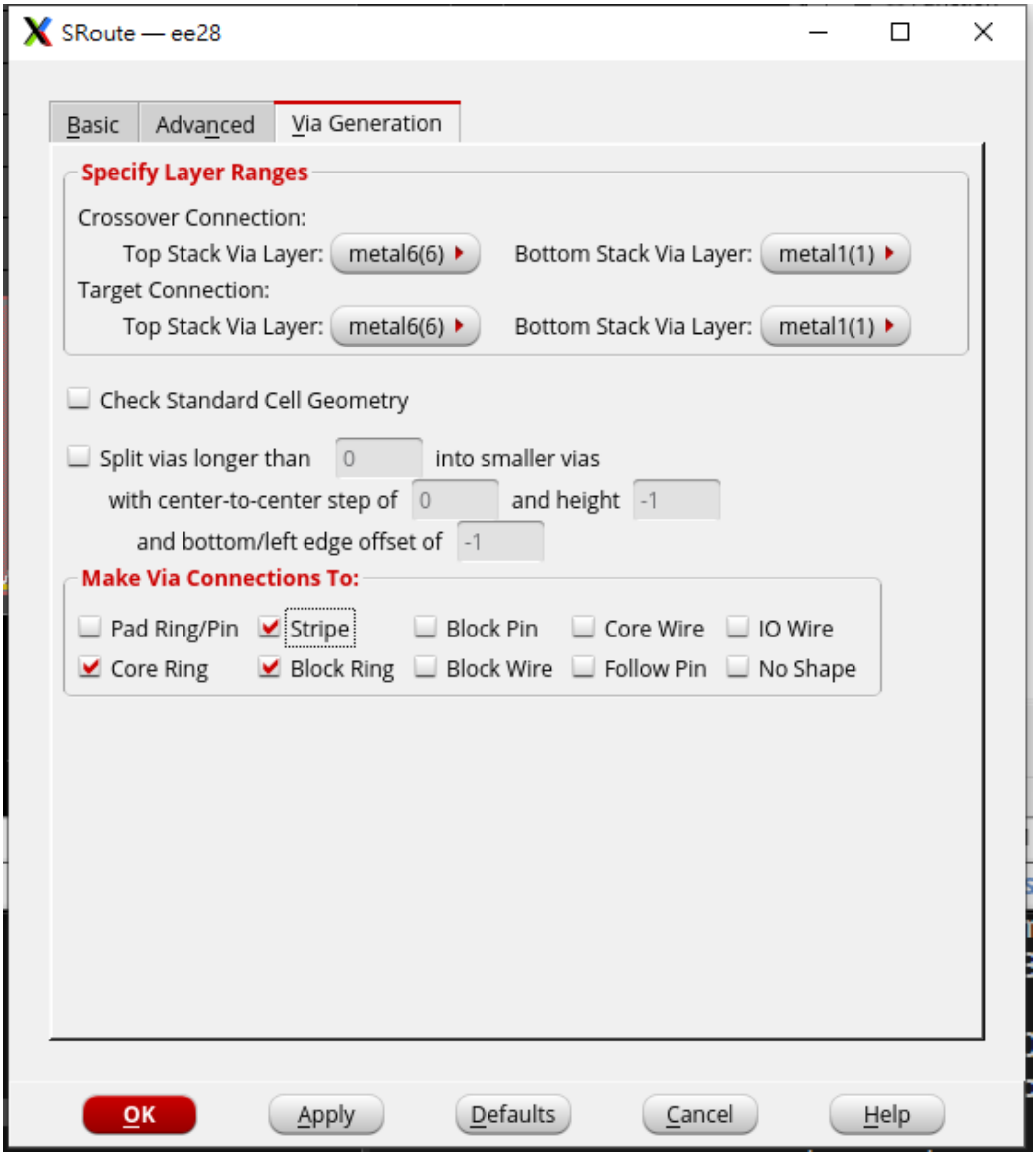
   d. Click **OK** button



Net(s): GND VCC

SRoute

□ Block Pins □ Pad Rings □ Floating Stripes
□ Pad Pins ☑ Follow Pins □ Secondary Power Pins



Specify Layer Ranges

Crossover Connection:

Top Stack Via Layer: metal6(6)   Bottom Stack Via Layer: metal1(1)

Target Connection:

Top Stack Via Layer: metal6(6)   Bottom Stack Via Layer: metal1(1)

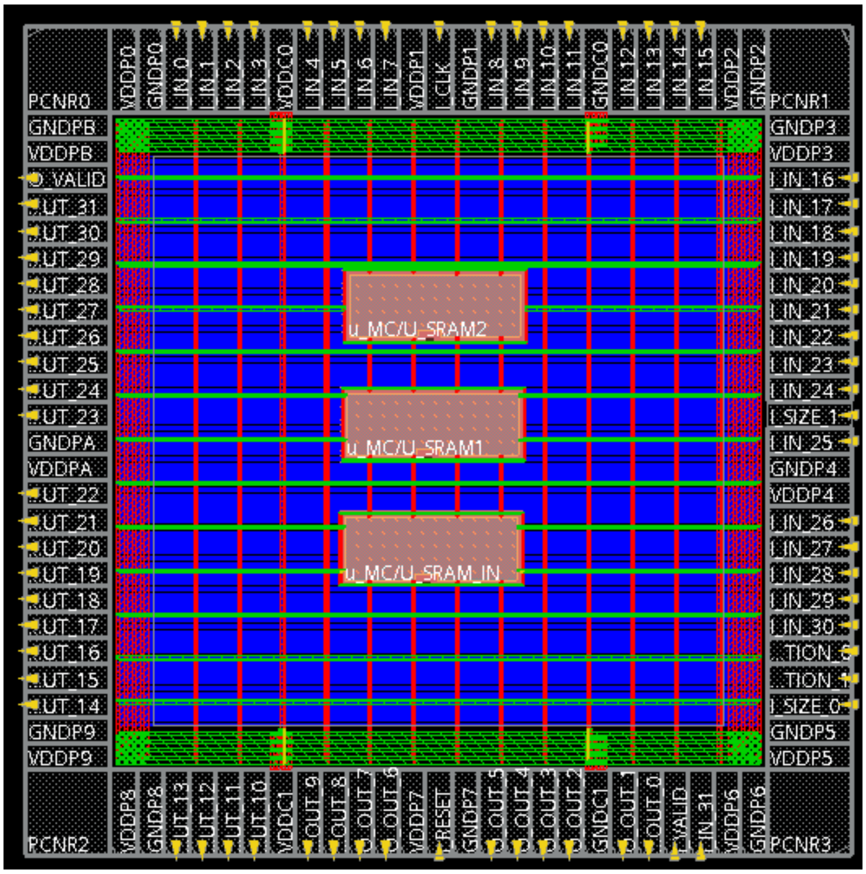
□ Check Standard Cell Geometry

□ Split vias longer than 0 into smaller vias
with center-to-center step of 0 and height -1
and bottom/left edge offset of -1

Make Via Connections To:

□ Pad Ring/Pin ☑ Stripe □ Block Pin □ Core Wire □ IO Wire

☑ Core Ring ☑ Block Ring □ Block Wire □ Follow Pin □ No Shape

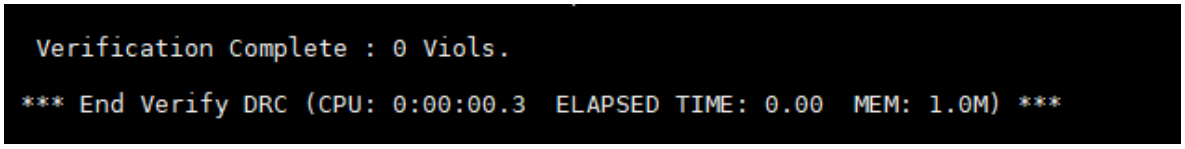


10. Verify DRC and LVS

1. In innovus menu, open ***Verify*** → ***DRC***

   ➢ Click **OK** button

   ➢ Check routing for DRC error



```
 Verification Complete : 0 Viols.
*** End Verify DRC (CPU: 0:00:00.3  ELAPSED TIME: 0.00  MEM: 1.0M) ***
```

Use ***Tools*** → ***Violation Browser*** to help finding Violations locations.

Ex.

Cut_Short via4 →delete the via4 which make this violation

Cut_Spacing via4 →delete the via4 which make this violation

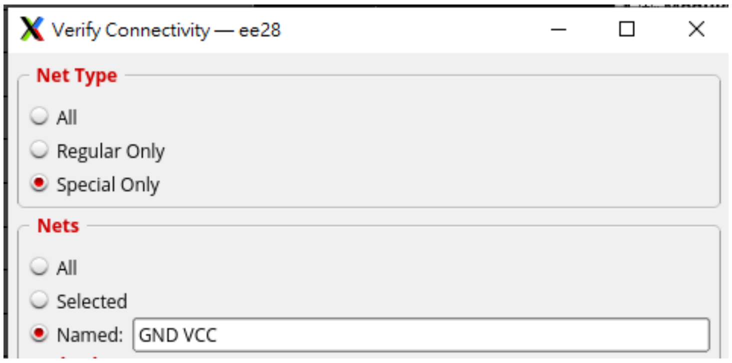
2. In innovus menu, open ***Verify*** → ***Connectivity***

   ➢ Net Type: ◆ Special Only

   ➢ Nets: ◆ Named: GND VCC

   ➢ Click **OK** button

   ➢ Check routing for LVS error



Verify Connectivity — ee28

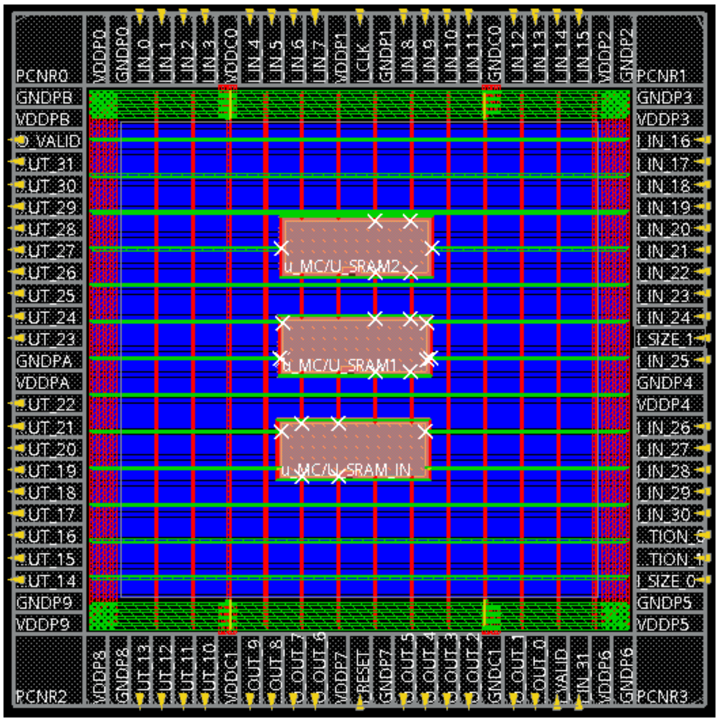
Net Type

○ All
○ Regular Only
● Special Only

Nets

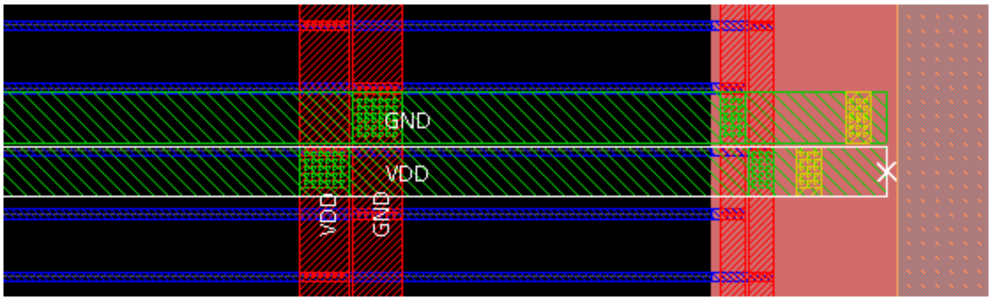
○ All
○ Selected
● Named: GND VCC

3. If the problem is the dangling wire (floating wire) around memory, like:

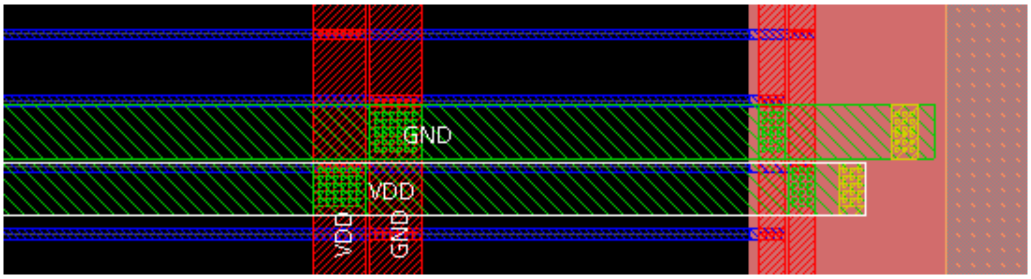


Zoom in and left click to select the highlighted wire:

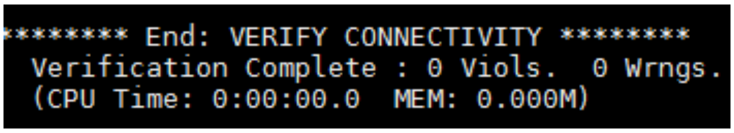
(In innovus menu, ***Tools → Violation Browser*** can help finding locations)



Press “shift+t” , the dangling wire will be adjusted. (Do not delete those wires or else the power cannot be averagely distributed on the chip)



4. After fixing all the dangling wires, run step 2. Again to ensure the connectivity correctness.

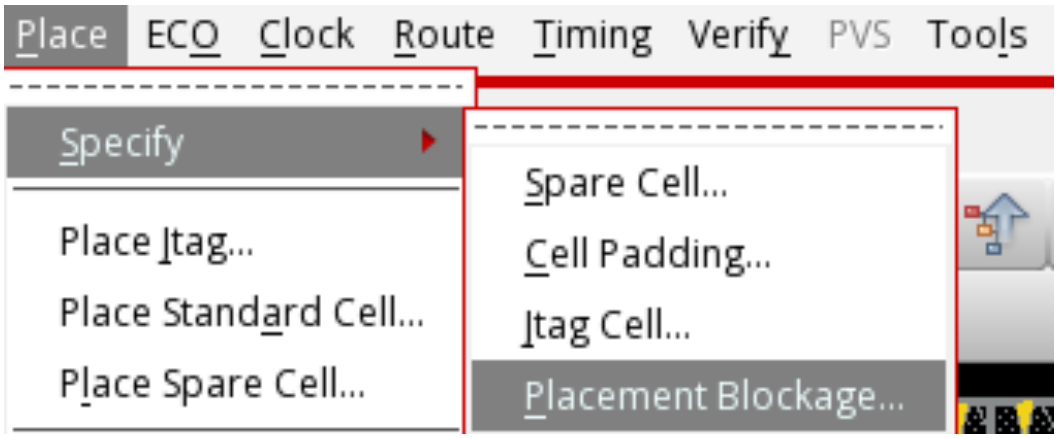


```
******** End: VERIFY CONNECTIVITY ********
  Verification Complete : 0 Viols.  0 Wrngs.
  (CPU Time: 0:00:00.0  MEM: 0.000M)
```

5. Save design as CHIP_powerplan.inn

11. Place Standard Cells

1. In innovus menu, open ***Place → Specify → Placement Blockage***

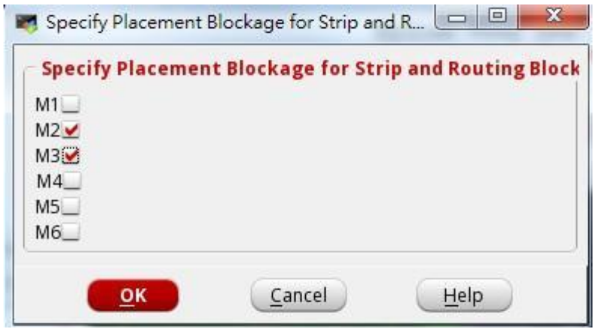


2. Specify placement blockage for stripes

a. Specify placement blockage under metal2 and metal3

- ➢ ◇ M1
- ➢ ◆ M2
- ➢ ◆ M3
- ➢ ◇ M4
- ➢ ◇ M5
- ➢ ◇ M6

b. Click **OK** button



Specify Placement Blockage for Strip and R...

Specify Placement Blockage for Strip and Routing Block

M1 □
M2 ☑
M3 ☑
M4 □
M5 □
M6 □

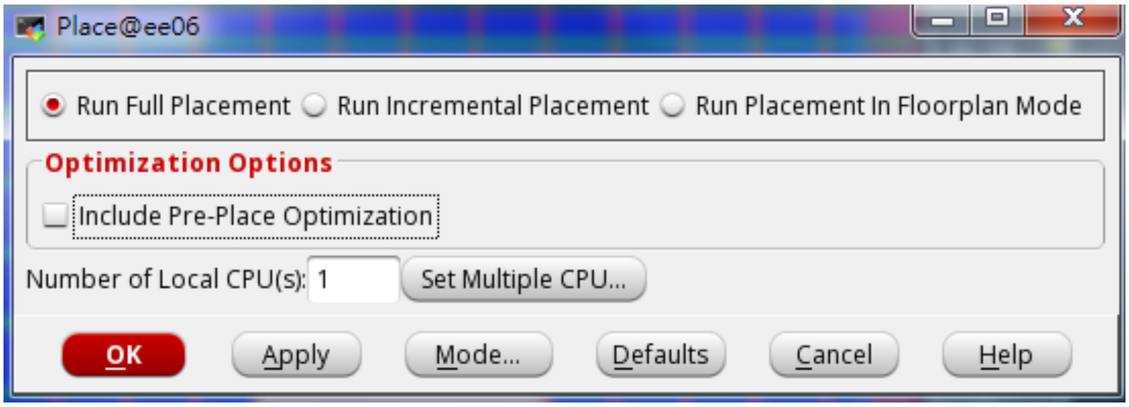
3. In innovus menu, open ***Place → Place Standard Cells…***

- ➢ ◆ Run Full Placement

Optimization Options

- ➢ ◇ Include Pre-Place Optimization

Click **OK** button



Place@ee06

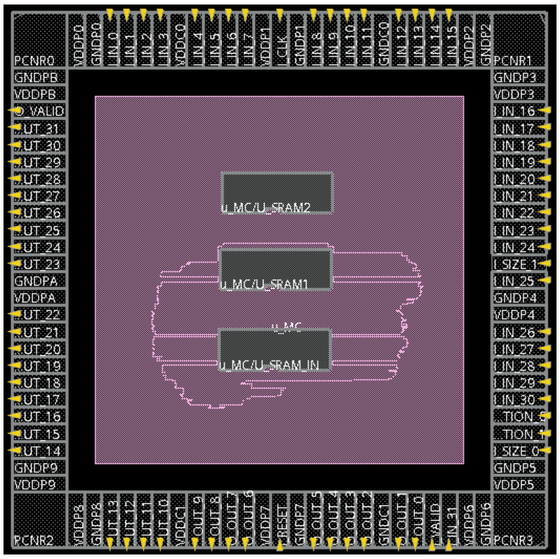
● Run Full Placement ○ Run Incremental Placement ○ Run Placement In Floorplan Mode

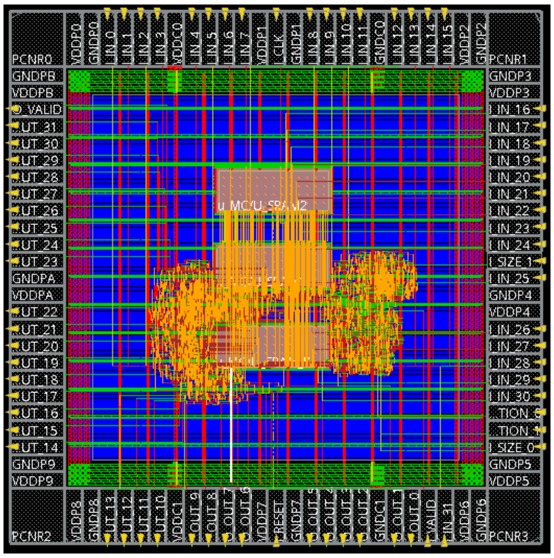
Optimization Options

□ Include Pre-Place Optimization

Number of Local CPU(s): 1

The following figures show the results of Amoeba view and Physical view



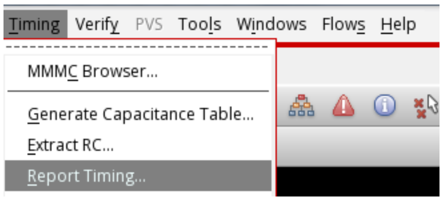


4. Save design as CHIP_placement.inn

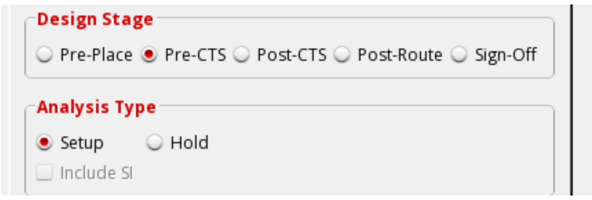
12. **In-Place Optimization (IPO)**

- Before Clock Tree Synthesis

1. In innovus menu, open ***Timing → Report Timing…***



2. Perform trial route to model the interconnection RC effects
   - **a.** Design Stage ◆ pre-CTS
   - **b.** Analysis Type ◆ Setup
   - **c.** Click **OK** button



Design Stage
○ Pre-Place ● Pre-CTS ○ Post-CTS ○ Post-Route ○ Sign-Off

Analysis Type
● Setup ○ Hold
□ Include SI

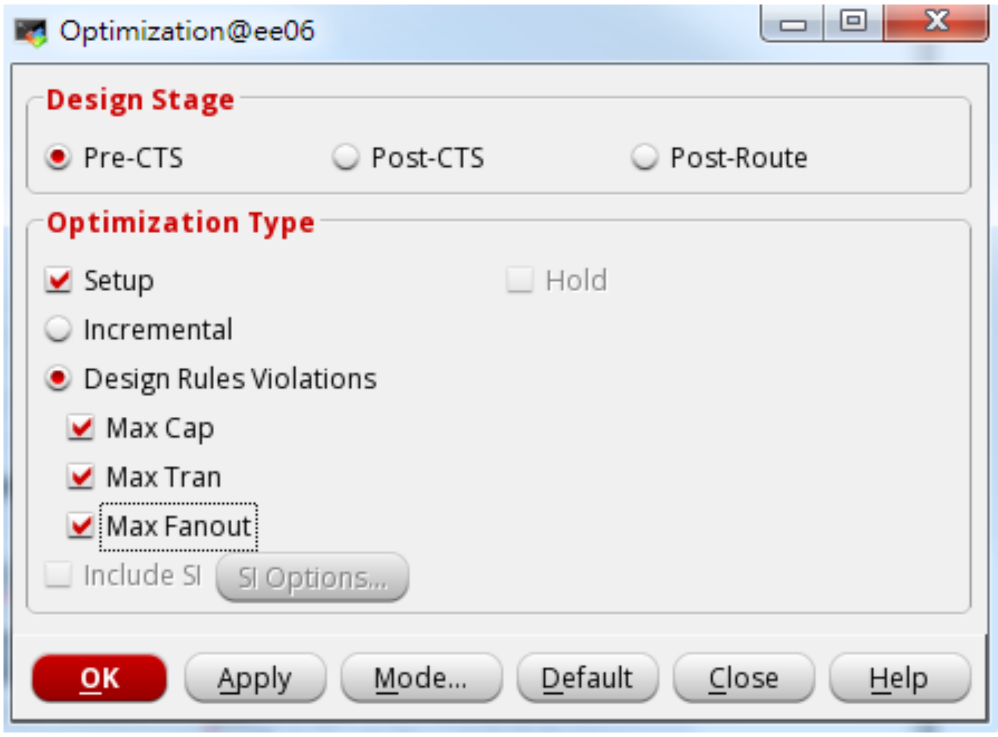
3. See timing reports in **timingReports/** directory, **CHIP_preCTS.slk** shows the timing analysis results. All slack values must be positive value in this file. Moreover, for detail path report, see **CHIP_preCTS_all.tarpt**. DRVs report files: ***.cap**, ***.fanout**, and ***.tran**.
4. If the timing slack is negative, or there are DRVs, open ***ECO → Optimize Design…*** in innovus menu
5. Perform pre-CTS IPO

a. Design Stage ◆ Pre-CTS

b. Optimization Type

- ➢ ◆ Setup
- ➢ ◆ Design Rule Violations
  - ◆ Max Cap
  - ◆ Max Tran
  - ◆ Max Fanout

c. Click **OK** button



Optimization@ee06

Design Stage

● Pre-CTS ○ Post-CTS ○ Post-Route

Optimization Type

☑ Setup ☐ Hold

○ Incremental

● Design Rules Violations

☑ Max Cap

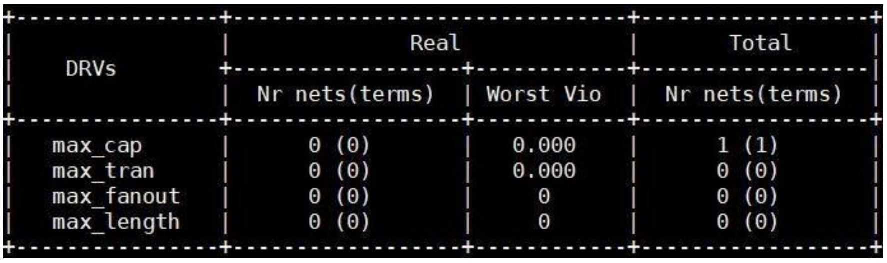
☑ Max Tran

☑ Max Fanout

☐ Include SI

6. Save design as CHIP_preCTS.inn

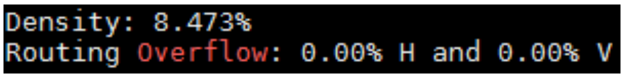
In preCTS timing report, the max cap violation of DRV may be caused by reset_n or clk, which connects to a lot of registers. After ECO, the rst_n DRV may be fixed by inserting buffers. Whereas the clk may not since the clk buffers will be inserted in the next step: Clock Tree Synthesis (CTS) stage. The remaining clk DRV will be shown like:



```
+---------------+--------------------------------+-----------------+
|               |              Real              |      Total      |
|     DRVs      +------------------+-------------+-----------------|
|               |  Nr nets(terms)  |  Worst Vio  |  Nr nets(terms) |
+---------------+------------------+-------------+-----------------+
|   max_cap     |      0 (0)       |    0.000    |      1 (1)      |
|   max_tran    |      0 (0)       |    0.000    |      0 (0)      |
|   max_fanout  |      0 (0)       |      0      |      0 (0)      |
|   max_length  |      0 (0)       |      0      |      0 (0)      |
+---------------+------------------+-------------+-----------------+
```

and can be view in the file ***timingReports/CHIP_preCTS.tran.gz***.This DRV should be fixed after synthesizing the clock tree, thus don't worry in this stage. You only have to worry if this DRV cannot be fixed after CTS.

You will also see **Density** and **Routing Overflow** terms at the bottom of the timing report.



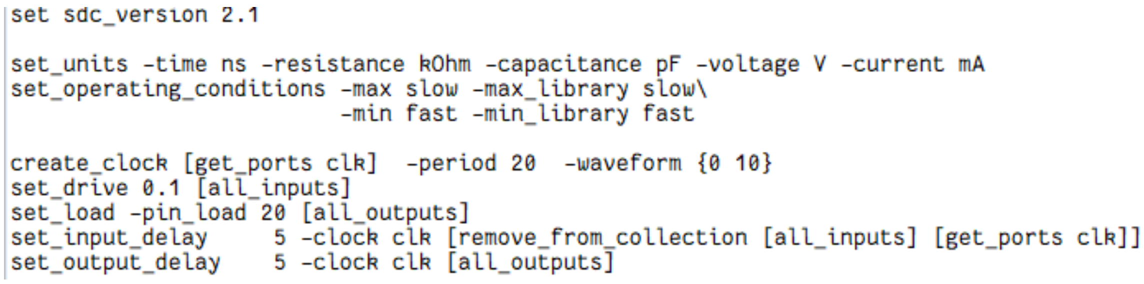
```
Density: 8.473%
Routing Overflow: 0.00% H and 0.00% V
```

The lower the density is, the lower the standard cells (excluding Hardmacros) are utilizing the core, which means it provides larger flexibility in further routing and optimization steps.

After the placement and in the preCTS stages, innovus performs **trial route** to give the rough delay calculation of path between every registers / input output ports so the preCTS can be done. Similar to the density which is the placement utilization, the routing overflow represents the wiring congestion level. The higher the routing overflow is, the harder of the coming CTS and detail routing (**nanoroute**). Usually when Horizontal and Vertical routing overflow are both < 0.5% ~ 1.0%, the further routing problem will be small. The routing overflow term will disappear after **nanorouting** since the **detail route** will be given to replace the **trial route**.

13. Clock Tree Synthesis (CTS)

Change the CIHP_cts.sdc (period, -waveform, input delay, output delay) according to your CHIP.sdc.



```
set sdc_version 2.1

set_units -time ns -resistance kOhm -capacitance pF -voltage V -current mA
set_operating_conditions -max slow -max_library slow\
                         -min fast -min_library fast

create_clock [get_ports clk]  -period 20  -waveform {0 10}
set_drive 0.1 [all_inputs]
set_load -pin_load 20 [all_outputs]
set_input_delay     5 -clock clk [remove_from_collection [all_inputs] [get_ports clk]]
set_output_delay    5 -clock clk [all_outputs]
```

➢ source ./cmd/ccopt.cmd

14. In-Place Optimization (IPO)

- After Clock Tree Synthesis

1. In innovus menu, open ***Timing → Report Timing…***

2. Perform trial route to model the interconnection RC effects

   a. Design Stage ◆ Post-CTS

   b. Analysis Type ◆ Setup

   c. Click **OK** button

3. See timing reports in **timingReports/** directory, **CHIP_postCTS.slk** shows the timing analysis results. All slack values must be positive value in this file. Moreover, for detail path report, see **CHIP_postCTS_all.tarpt**. DRVs report files: ***.cap**, ***.fanout**, and ***.tran**.

4. If the timing slack is negative, or there are DRVs, open ***ECO → Optimize Design…*** in innovus menu

5. Perform post-CTS IPO

   a. Design Stage ◆ Post-CTS

   b. Optimization Type

   ➢ ◆ Setup

   ➢ ◆ Design Rule Violations

   ◆ Max Cap

   ◆ Max Tran

   ◆ Max Fanout

   c. Click **OK** button

From post CTS steps, hold time checking is also required

6. In innovus menu, open ***Timing → Report Timing…***

7. Perform trial route to model the interconnection RC effects

   a. Design Stage ◆ Post-CTS

   b. Analysis Type ◆ Hold

   c. Click **OK** button

8. See timing reports in **timingReports/** directory, **CHIP_postCTS_hold.slk** shows the timing analysis results. All slack values must be positive value in this file. Moreover, for detail path report, see **CHIP_postCTS_all_hold.tarpt**. DRVs report files: ***.cap**, ***.fanout**, and ***.tran**.

9. If the timing slack is negative, or there are DRVs, open ***ECO → Optimize Design…*** in innovus menu

10. Perform Post-CTS IPO

    a. Design Stage ◆ Post-CTS

    b. Optimization Type

    ➢ ◆ Hold

    ➢ ◆ Design Rule Violations

    ◆ Max Cap

    ◆ Max Tran

◆ Max Fanout

c. Click **OK** button

11. Save design as CHIP_postCTS.inn

15. Add PAD Filler

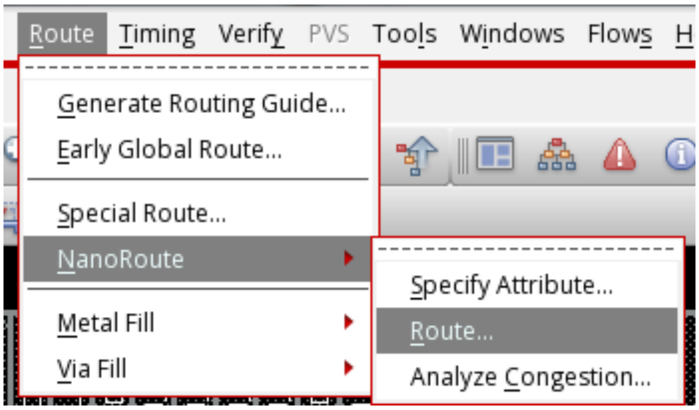
1. In innovus command prompt, execute the following commands:

➢ **source ./cmd/addIOFiller.cmd**

-fillAnyGap PAD filler must be added before detail route, or there may have some DRC/LVS violations after PAD filler insertion

16. SI-Prevention Detail Route (NanoRoute)

1. In innovus menu, open ***Route → NanoRoute → Route***



2. Nanoroute can prevent cross talk effects and fix antenna rule violations, also it routes design to meet timing constraints.

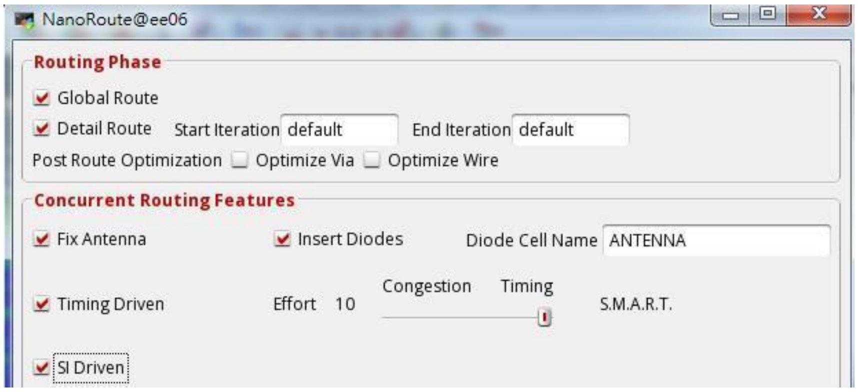
a. Configure routing features

➢ ◆ Fix Antenna

➢ ◆ Insert Diodes Diode Cell Name: ANTENNA

➢ ◆ Timing Driven Effort: 10

➢ ◆ SI Driven



NanoRoute@ee06

Routing Phase

☑ Global Route

☑ Detail Route Start Iteration default End Iteration default

Post Route Optimization □ Optimize Via □ Optimize Wire

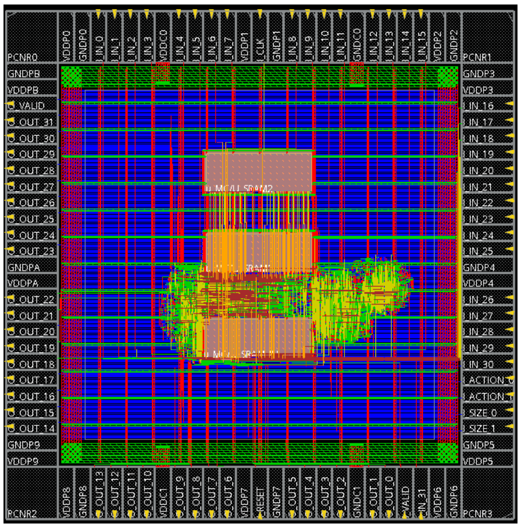
Concurrent Routing Features

☑ Fix Antenna ☑ Insert Diodes Diode Cell Name ANTENNA

☑ Timing Driven Effort 10 Congestion Timing S.M.A.R.T.

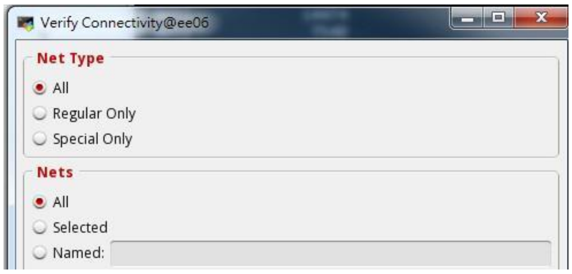
☑ SI Driven

b. Click **OK** button



3. In innovus menu, open ***Verify → Connectivity***

   - ➢ Net Type: ◆ All
   - ➢ Nets: ◆ All
   - ➢ Click **OK** button
   - ➢ Check routing for LVS error


Verify Connectivity@ee06

Net Type

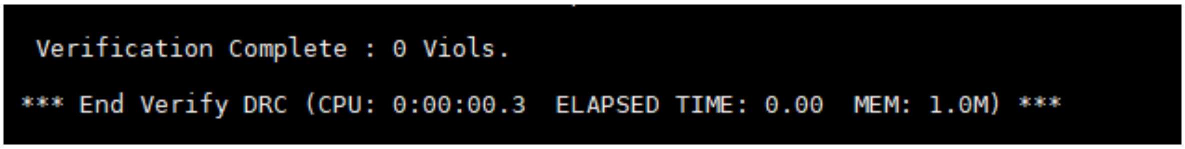
● All
○ Regular Only
○ Special Only

Nets

● All
○ Selected
○ Named:

4. In innovus menu, open ***Verify → DRC***

   - ➢ Click **OK** button
   - ➢ Check routing for DRC error



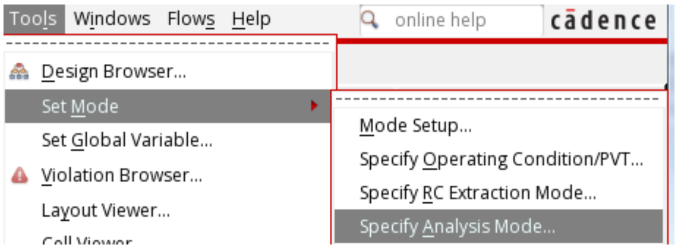
```
 Verification Complete : 0 Viols.
*** End Verify DRC (CPU: 0:00:00.3  ELAPSED TIME: 0.00  MEM: 1.0M) ***
```

5. Save design as CHIP_nanoRoute.inn

17. In-Place Optimization (consider crosstalk effects)

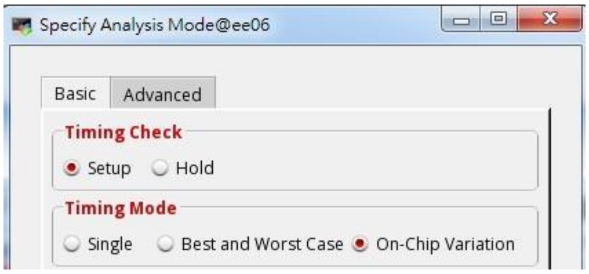
- After Detail Route

1. In innovus menu, open ***Tools → Set Mode → Specify Analysis Mode…***



a. Timing Mode: ◆ On-Chip Variation

b. Click **OK** button



Specify Analysis Mode@ee06

Basic | Advanced

Timing Check

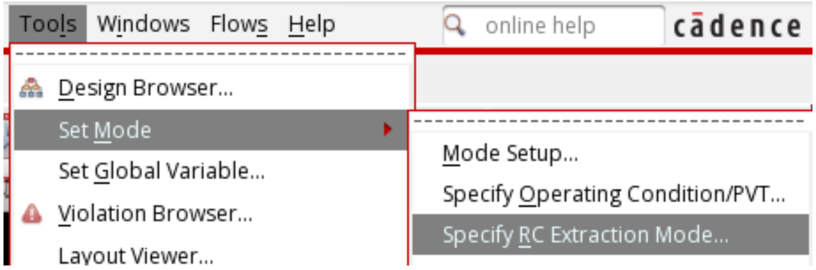
● Setup ○ Hold

Timing Mode

○ Single ○ Best and Worst Case ● On-Chip Variation

2. In innovus menu, open

Tools → Set Mode → Specify RC Extraction Mode…



a. ◆ PostRoute

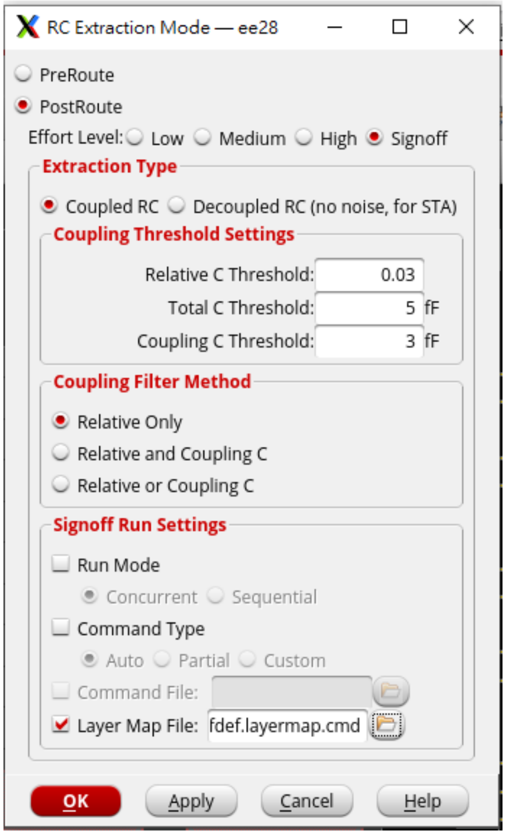
b. Effort Level: ◆ Signoff

c. Extraction Type: ◆ Coupled RC

d. Signoff Run Settings:

◆ Layer Map File: lefdef.layermap.cmd

e. Click **OK** button



RC Extraction Mode — ee28

○ PreRoute

● PostRoute

Effort Level: ○ Low ○ Medium ○ High ● Signoff

Extraction Type

● Coupled RC ○ Decoupled RC (no noise, for STA)

Coupling Threshold Settings

Relative C Threshold: 0.03

Total C Threshold: 5 fF

Coupling C Threshold: 3 fF

Coupling Filter Method

● Relative Only

○ Relative and Coupling C

○ Relative or Coupling C

Signoff Run Settings

□ Run Mode

● Concurrent ○ Sequential

□ Command Type

● Auto ○ Partial ○ Custom

□ Command File:

☑ Layer Map File: fdef.layermap.cmd

3. Setting rc and si

➢ innovus > set_db extract_rc_engine post_route

➢ innovus > set_db extract_rc_effort_level high

   ➢ innovus > set_db delaycal_enable_si true

4. In innovus menu, open ***Timing → Report Timing…***

5. Perform trial route to model the interconnection RC effects
   - **a.** Design Stage ◆ Post-Route
   - **b.** Analysis Type ◆ Setup
   - **c.** Click **OK** button

6. See timing reports in **timingReports/** directory, **CHIP_postRoute.slk** shows the timing analysis results. All slack values must be positive value in this file. Moreover, for detail path report, see **CHIP_postRoute_all.tarpt**. DRVs report files: ***.cap**, ***.fanout**, and ***.tran**.

**** You can only run postRoute and signoff timing analysis on ee servers ****

7. If the timing slack is negative, or there are DRVs, open ***ECO→Optimize Design…*** in innovus menu

8. Perform post-Route IPO
   - **a.** Design Stage ◆ post-Route
   - **b.** Optimization Type
     - ➢ ◆ Setup
     - ➢ ◆ Design Rule Violations
       - ◆ Max Cap
       - ◆ Max Tran
       - ◆ Max Fanout
   - **c.** Click **OK** button

9. In innovus menu, open ***Timing → Report Timing…***

10. Perform trial route to model the interconnection RC effects
    - **a.** Design Stage ◆ Post-Route
    - **b.** Analysis Type ◆ Hold
    - **c.** Click **OK** button

11. See timing reports in **timingReports/** directory, **CHIP_postRoute_hold.slk** shows the timing analysis results. All slack values must be positive value in this file. Moreover, for detail path report, see **CHIP_postRoute_all_hold.tarpt**. DRVs report files: ***.cap**, ***.fanout**, and ***.tran**.

12. If the timing slack is negative, or there are DRVs, open ***ECO→Optimize Design…*** in innovus menu

13. Perform post-Route IPO

    a. Design Stage ◆ post-Route

    b. Optimization Type

    - ◆ Hold
    - ◆ Design Rule Violations

        ◆ Max Cap

        ◆ Max Tran

        ◆ Max Fanout

    c. Click **OK** button

14. Save design as CHIP_postRoute.inn

18. Timing Analysis (Signoff)

- Optional in this Practice

Signoff timing analysis is similar to postRoute timing analysis. In addition to invoke Quantus QRC Extraction tool to calculate the RC delay as postRoute, **it also consider the SI affect with the signoff RC**.

Currently, the delay considering the SI affect with signoff RC can be only used for timing analysis, but cannot be used for timing optimization. If you want to perform timing analysis or timing optimization for postRoute again after signoff timing analysis, you need to reset the SI mode by the command: innovus > **setDelayCalMode -siMode signoff**

1. In innovus menu, open ***Timing → Report Timing…***

2. Perform trial route to model the interconnection RC effects

    a. Design Stage ◆ Sign-Off

    b. Analysis Type ◆ Setup

    c. Click **OK** button

3. See timing reports in **timingReports/** directory, **CHIP_signOff.slk** shows the timing analysis results. All slack values must be positive value in this file. Moreover, for detail path report, see **CHIP_signOff_all.tarpt**. DRVs report files: ***.cap**, ***.fanout**, and ***.tran**.

4. In innovus menu, open ***Timing → Report Timing…***

5. Perform trial route to model the interconnection RC effects

    a. Design Stage ◆ Sign-Off

    b. Analysis Type ◆ Hold

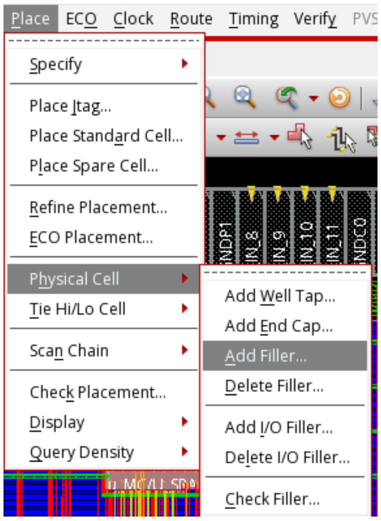
   c. Click **OK** button

6. See timing reports in **timingReports/** directory, **CHIP_signOff_hold.slk** shows the timing analysis results. All slack values must be positive value in this file. Moreover, for detail path report, see **CHIP_signOff_all_hold.tarpt**. DRVs report files: ***.cap**, ***.fanout**, and ***.tran**.

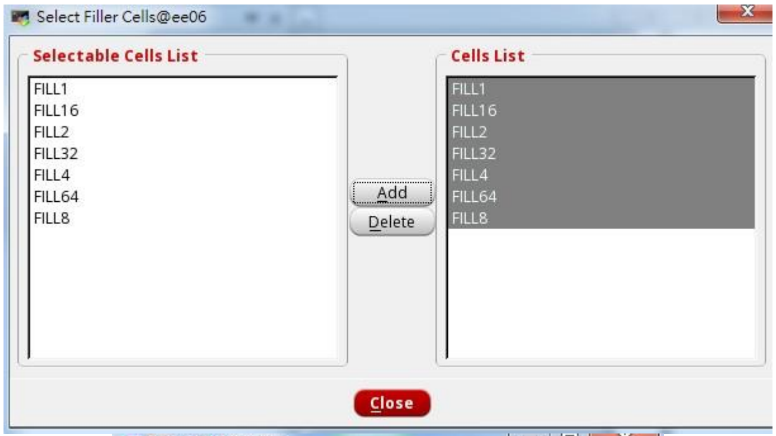
You can perform the above steps to see if there is any difference between the results between the signoff timing analysis and the postRoute timing analysis. **In most experiences and in this practice, signoff timing analysis result is very closed the postRoute timing analysis**. Thus in this practice you can choose to perform this step or not.

19. Add CORE Filler Cells

1. In innovus menu, open ***Place → Physical Cell → Add Filler***



2. Add core filler to improve electric effects of NWELL and PWELL:
   a. Click **Select** button next to Cell Name(s)
   b. Select core filler cells
   c. Click **Add** button
   d. Click **Close** button
   e. Click **OK** button

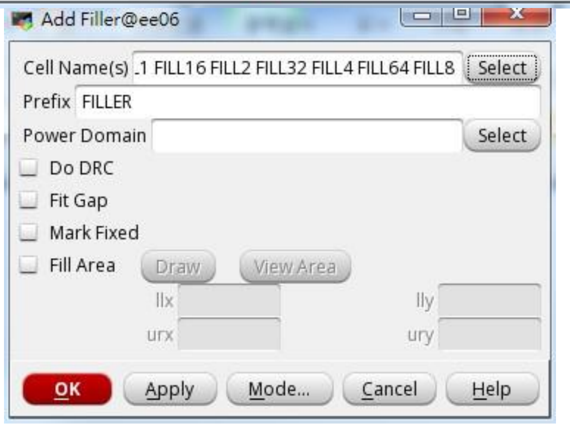

Select Filler Cells@ee06

Selectable Cells List

FILL1
FILL16
FILL2
FILL32
FILL4
FILL64
FILL8

Cells List

FILL1
FILL16
FILL2
FILL32
FILL4
FILL64
FILL8


Add Filler@ee06

Cell Name(s) .1 FILL16 FILL2 FILL32 FILL4 FILL64 FILL8

Prefix FILLER

Power Domain

□ Do DRC

□ Fit Gap

□ Mark Fixed

□ Fill Area

llx lly

urx ury

20. Stream Out and Write Netlist

1. Save design as CHIP.inn
2. Write CHIP.sdf

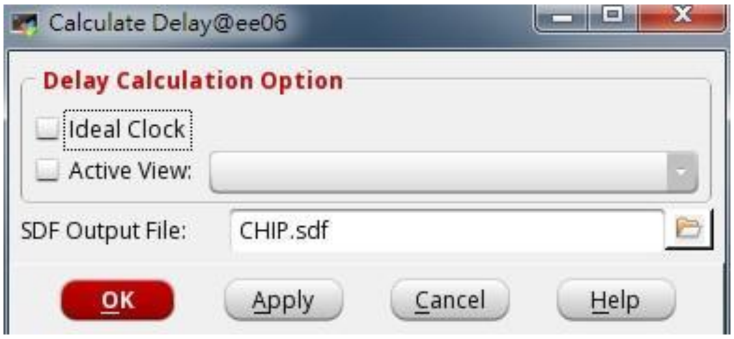
   a. In innovus menu, open ***Timing → Write SDF***

   b. Delay Calculation Option

   ➢ ◇ Ideal Clock

   Ideal Clock should be disabled

   c. Click **OK** button


Calculate Delay@ee06

Delay Calculation Option

□ Ideal Clock

□ Active View:

SDF Output File: CHIP.sdf

(or you can use the command "write_sdf CHIP.sdf")

3. Save design netlist CHIP.v for post-layout simulation:

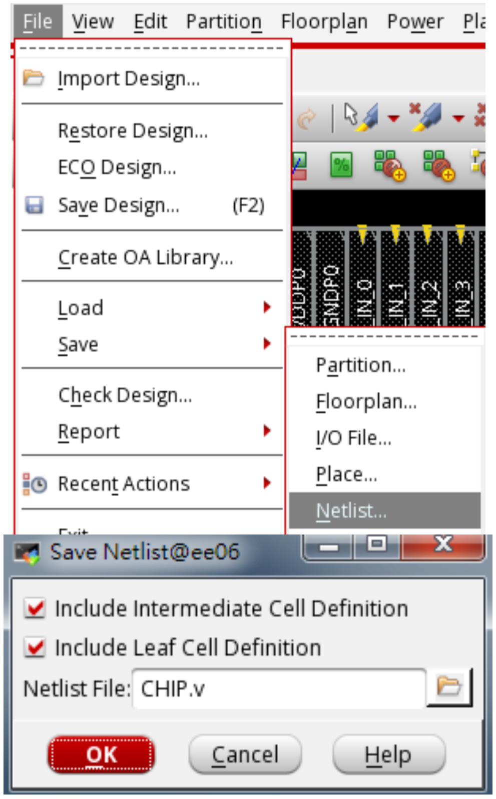
   a. In innovus menu, open ***File→ Save → Netlist…***

   ➢ ◆ Include Intermediate Cell Definition

   ➢ ◆ Include Leaf Cell Definition

   ➢ Netlist File: **CHIP.v**

- Click **OK** button



Save Netlist@ee06

☑ Include Intermediate Cell Definition

☑ Include Leaf Cell Definition

Netlist File: CHIP.v

21. Post-Layout Gate-Level Simulation

1. Change to directory **06_POST**

2. Perform Post-Layout Gate-level simulation of CHIP.v

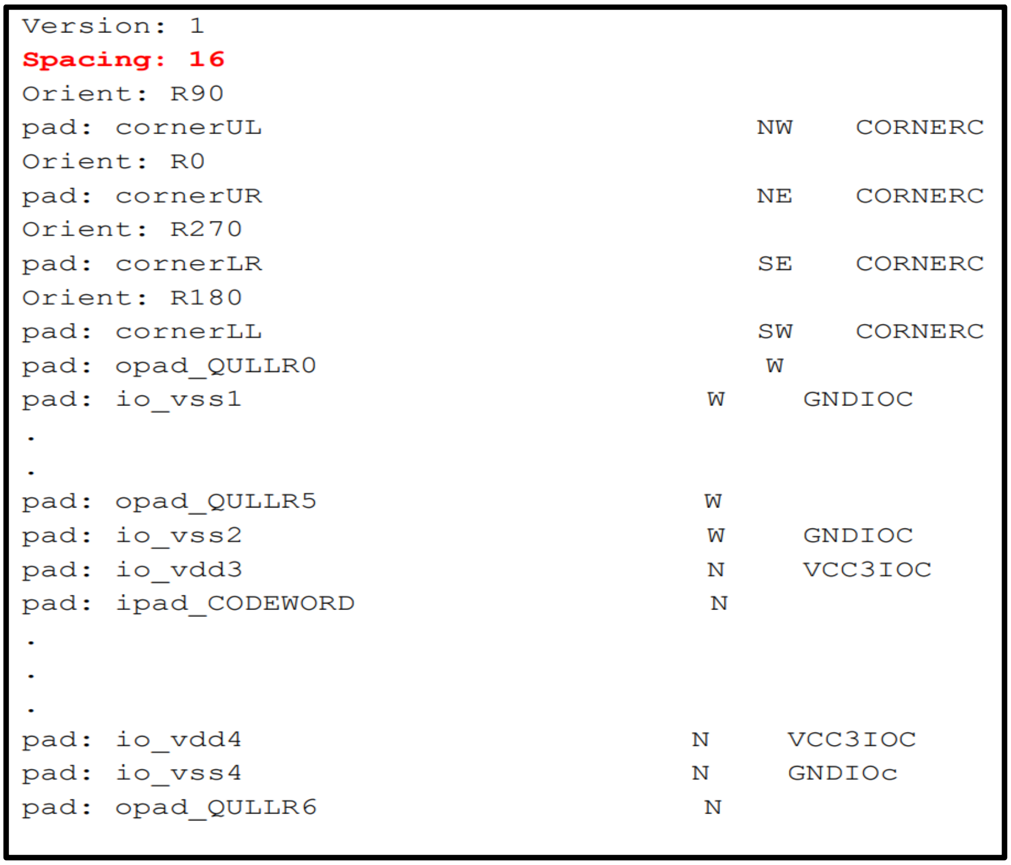
   % ./01_run_vcs_post The PATTERN latency should be the same as gate level simulation

3. If the 06_POST folder doesn’t have CHIP.v and CHIP.sdf, please link them by commands:

- ln -s ../05_APR/CHIP.v ./CHIP.v
- ln -s ../05_APR/CHIP.sdf ./CHIP.sdf

Appendix: IO Pad assignment

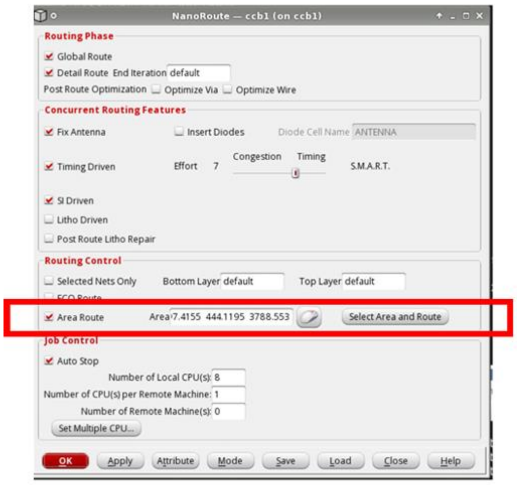
CHIP.io:



```
Version: 1
Spacing: 16
Orient: R90
pad: cornerUL                          NW   CORNERC
Orient: R0
pad: cornerUR                          NE   CORNERC
Orient: R270
pad: cornerLR                          SE   CORNERC
Orient: R180
pad: cornerLL                          SW   CORNERC
pad: opad_QULLR0                        W
pad: io_vss1                         W    GNDIOC
.
.
pad: opad_QULLR5                    W
pad: io_vss2                         W    GNDIOC
pad: io_vdd3                         N    VCC3IOC
pad: ipad_CODEWORD                   N
.
.
.
pad: io_vdd4                        N    VCC3IOC
pad: io_vss4                        N    GNDIOc
pad: opad_QULLR6                    N
```

- ## Short issue

Delete the short metal line, then use area nano route (Area Route--> Select Area and Route) to re-route that path. (sometimes may need to delete a longer metal line so the new routed result won't have short again)



NanoRoute — ccb1 (on ccb1)

Routing Phase

☑ Global Route
☑ Detail Route End Iteration default
Post Route Optimization □ Optimize Via □ Optimize Wire

Concurrent Routing Features

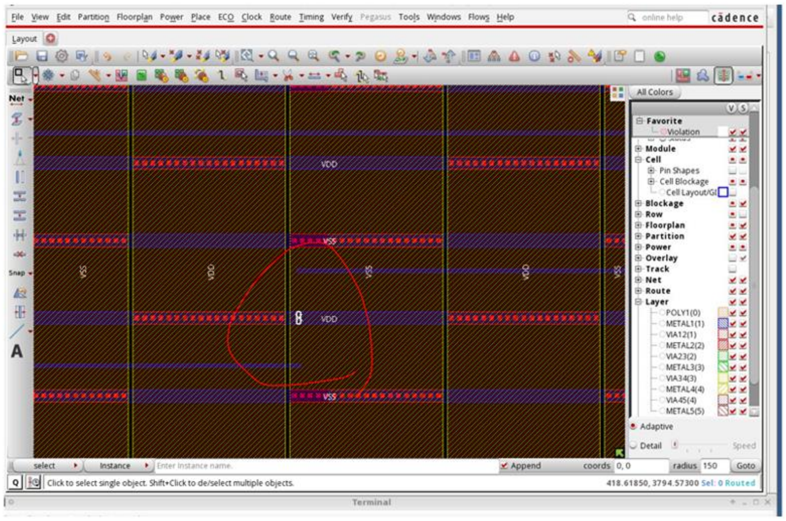
☑ Fix Antenna □ Insert Diodes Diode Cell Name ANTENNA
☑ Timing Driven Effort 7 Congestion Timing S.M.A.R.T.
☑ SI Driven
□ Litho Driven
□ Post Route Litho Repair

Routing Control

□ Selected Nets Only Bottom Layer default Top Layer default
□ ECO Route
☑ Area Route Area 7.4155 444.1195 3788.553 Select Area and Route

Job Control

☑ Auto Stop
Number of Local CPU(s): 8
Number of CPU(s) per Remote Machine: 1
Number of Remote Machine(s): 0



Click to select single object. Shift+Click to de/select multiple objects.
418.61850, 3794.57300 Sel: 0 Routed

- ## Check Area

type > summaryReport -noHtml -outfile summaryReport.rpt

then you will see the CORE area in summaryReport.rpt

- You can execute >> source ./cmd/run_apr.cmd in innovus interface to do Step 2. and then you can directly go to Floorplan Step!