NCTU-EE IC Design LAB-Spring 2018

Innovus – From RTL to GDSII

You are required to run the APR flow with the school entrance system. Please follow the constraints and steps to generate the required data for demonstration.

1. Data Preparation

1. Change the directory to **05_APR**

2. Prepare chip design netlist:

a. Open **CHIP_SHELL.v**

- The top module name is **CHIP**
- This **CHIP** contains **I/O**, **I/O power**, and **core power** pad.
- Please calculate how many I/O and core power pads you need, and complete the netlist of pad cells.
- After defining the pad cells, please run %**./01_combine** to combine the CORE_SYN.v with CHIP_SHELL.v to be CHIP_SYN.v

b. Uniquify **CHIP_SYN.v**

- % **./02_run_uniquify,** then an uniquified netlist **CHIP_unique.v** will be generated.
- Before you go to SOC innovus, please make sure no error message is shown.

3. Prepare I/O pad location file **CHIP.io**

a. Please see **Appendix A** 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.**

4. Prepare timing constraint file **CHIP.sdc**

After synthesis, modify your design's sdc file "**CORE.sdc**" for APR simulation (a reference sdc file **REF.sdc** is provided).

5. Link library files

a. Timing libraries (LIB)

- **slow.lib** (~iclabta01/umc018/Lib/)
- **fast.lib** (~iclabta01/umc018/Lib/)
- **Umc18io3v5v_slow.lib** (~iclabta01/umc018/Lib/)
- **Umc18io3v5v_fast.lib** (~iclabta01/umc018/Lib/)

- *HardMacro*.lib (../04_MEM)

b. Physical libraries (LEF)

- **umc18_6lm.lef** (~iclabta01/umc018/Lef/)
- **umc18_6lm_antenna.lef** (~iclabta01/umc018/Lef/)
- **umc18io3v5v_6lm.lef** (~iclabta01/umc018/Lef/)
- *HardMacro*.vclef (../04_MEM)

c. RC extraction

- **umc18_1p6m.captbl** (~iclabta01/umc018/Captbl/)
- **icecaps.tch** (~iclabta01/umc018/FireIce/)
- **RCGen.tch** (~iclabta01/umc018/FireIce/)

d. CeltIC libraries

- **slow.cdb** (~iclabta01/umc018/CeltIC/)
- **fast.cdb** (~iclabta01/umc018/CeltIC/)

e. GDSII layout

- **umc18.gds2** (~iclabta01/umc018/Gds/)
- **umc18io3v5v_61m.gds2** (~iclabta01/umc018/Gds/)
- **lefdef.layermap** (~iclabta01/umc018/Layermap/)

2. Reading Cell Library information and Netlist for APR

1. Start SoC Innovus in the directory **05_APR** (don't use background execution)

   % innovus

2. In innovus menu, open ***File -> Import Design***

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

a. Verilog

- Files: **CHIP_unique.v**
- Top Cell: ◆By User : CHIP

b. Technology/Physical Libraries

- LEF Files: **umc18_6lm.lef umc18_6lm_antenna.lef**
  umc18io3v5v_6lm.lef (RA1SH.vclef)

c. Floorplan

- Files: CHIP.io

d. Power

- Power nets: VDD
- Ground nets: GND

e. Press "**Create Analysis Configuration**" tab

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

Max delay:

Name: lib_max

Timing Library: slow.lib,umc18io3v5v_slow.lib, RA1SH_slow_syn.lib

SI Library: slow.cdb

Min delay:

Name: lib_min

Timing Library: fast.lib,umc18io3v5v_fast.lib, RA1SH_fast_syn.lib

SI Library: fast.cdb

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

  Name: RC_Corner

  Cap Table: umc18_1p6m.captbl

  QRC Tech File: icecaps.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

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

  Name: func_mode

  SDC Constraint Files: CHIP.sdc

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

  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

- Double-click **Setup Analysis View** and specify the max analysis mode

Choose: av_func_mode_max

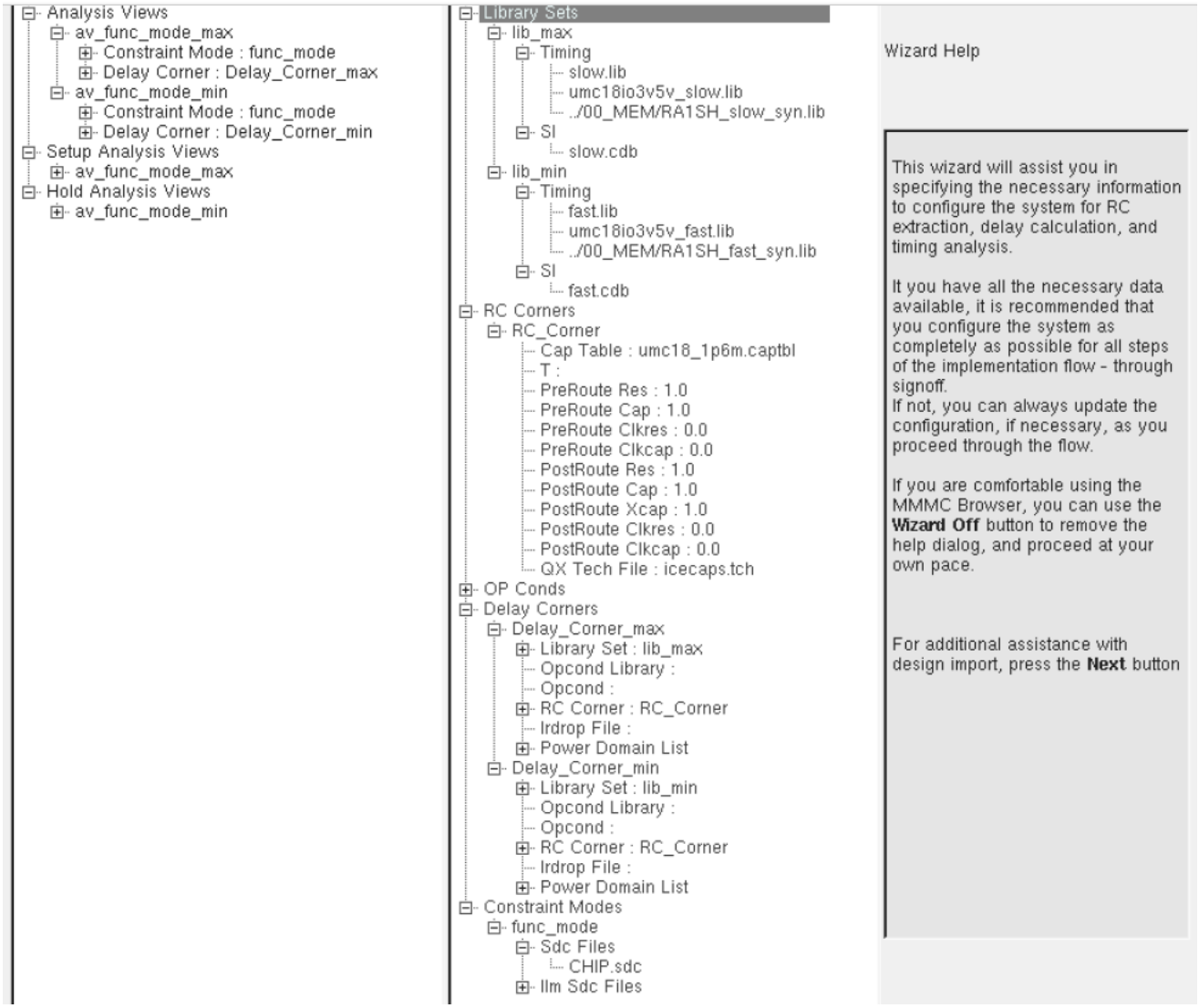
- Double-click **Hold Analysis View** and specify the min analysis mode

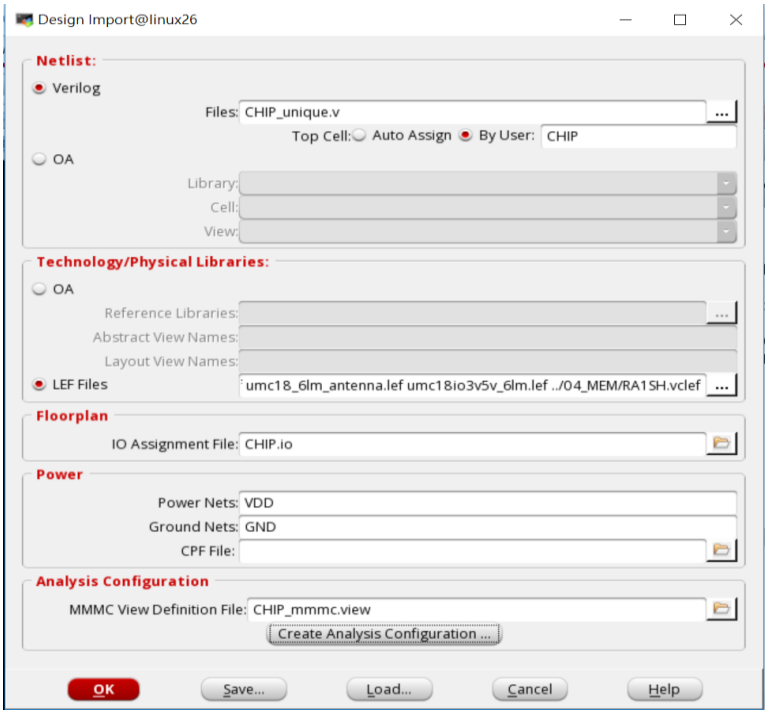
  Choose: av_func_modeV_min

- Save as “CHIP_mmmc.view”

f. IO Assignment

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



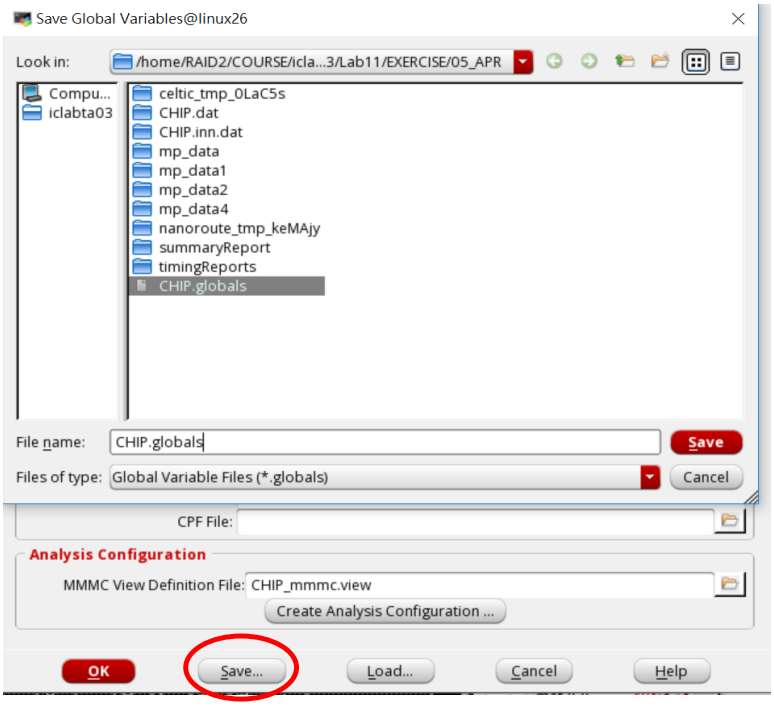


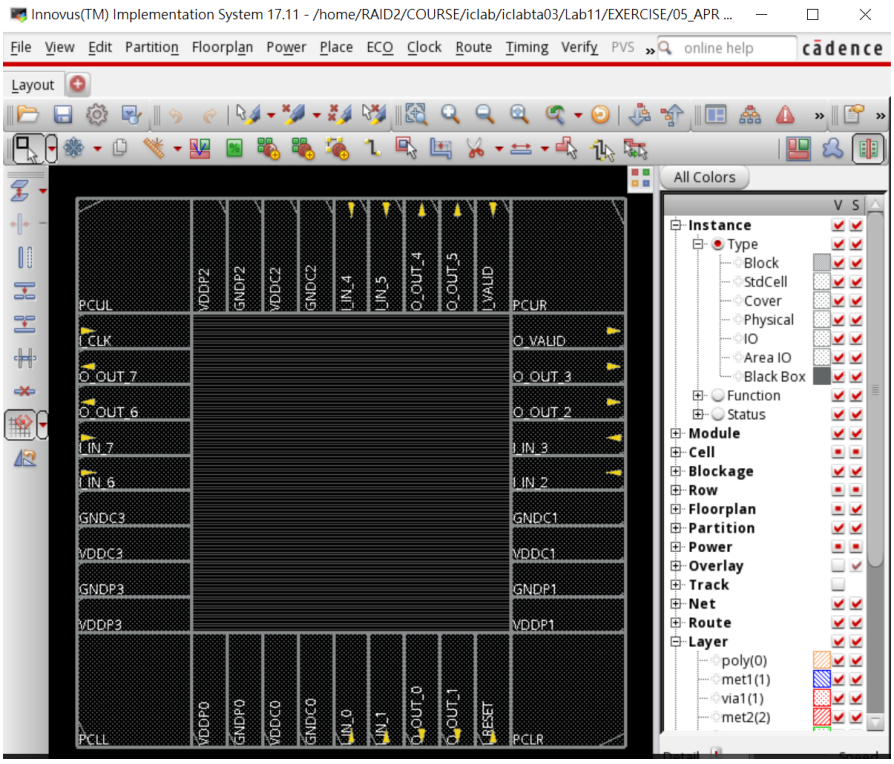
4. Save current settings:

f. Click **Save…** button

- File name: **CHIP.global**

g. Click **OK** button



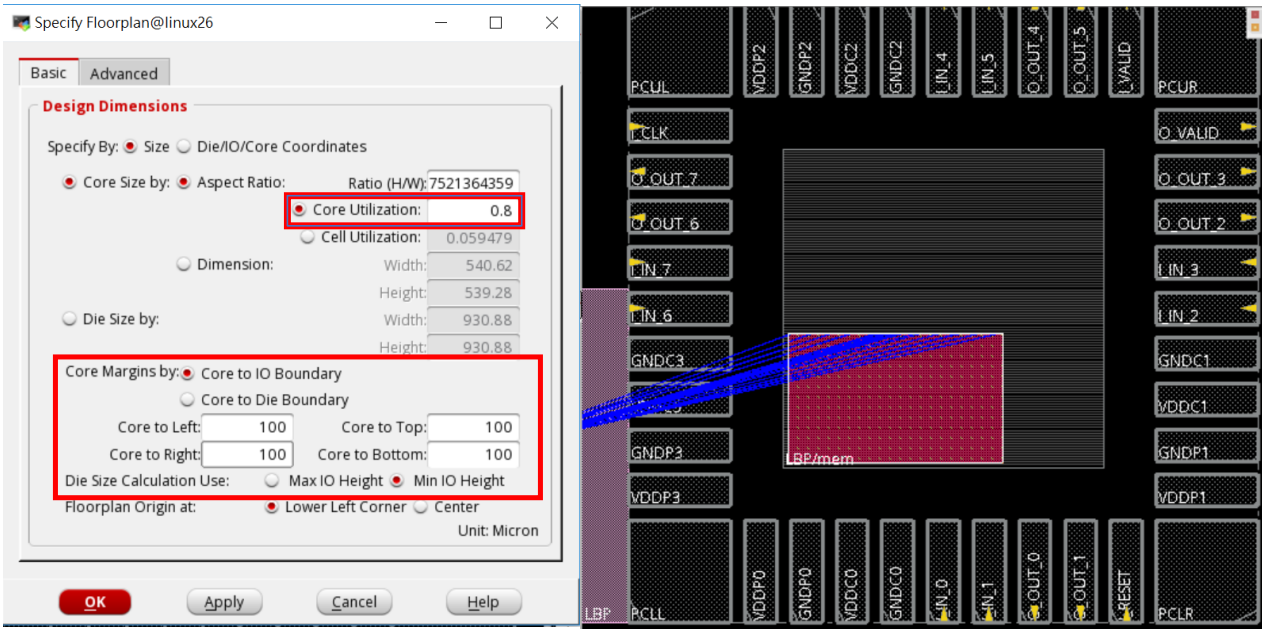


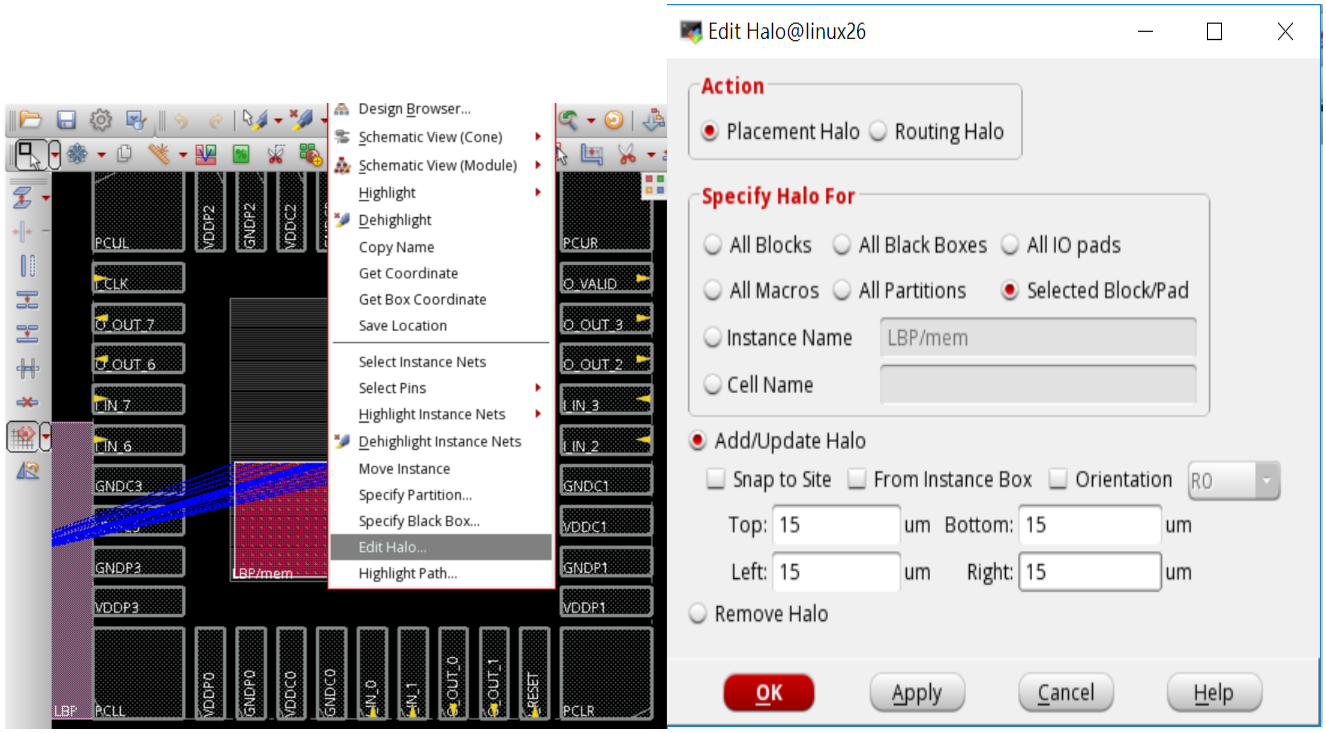
3. Specify Chip Floorplan

1. (Move macro blocks to proper position)
2. In innovus menu, open ***Floorplan -> Specify Floorplan***
3. Specify core size:

a. Core Utilization: Set any as your wish
(or you can give a specify dimension based on the size of macro)

4. Specify core margin:
   a. ◆Core to IO Boundary Core to Left: **100** Core to Right: **100** Core to Top: **100** Core to Bottom: **100**
5. Apply the specification:
   a. Click **OK** button
6. (Move macro blocks to proper position and move the mouse index to the Hard macro then click the right button of the mouse-> Choose "**Edit Halo**")
   a. Action: Placement Halo
   b. Specify Halo for: Selected Block/Pad
   c. Add/Update Halo
   - Top, Bottom, Left, Right: 15um





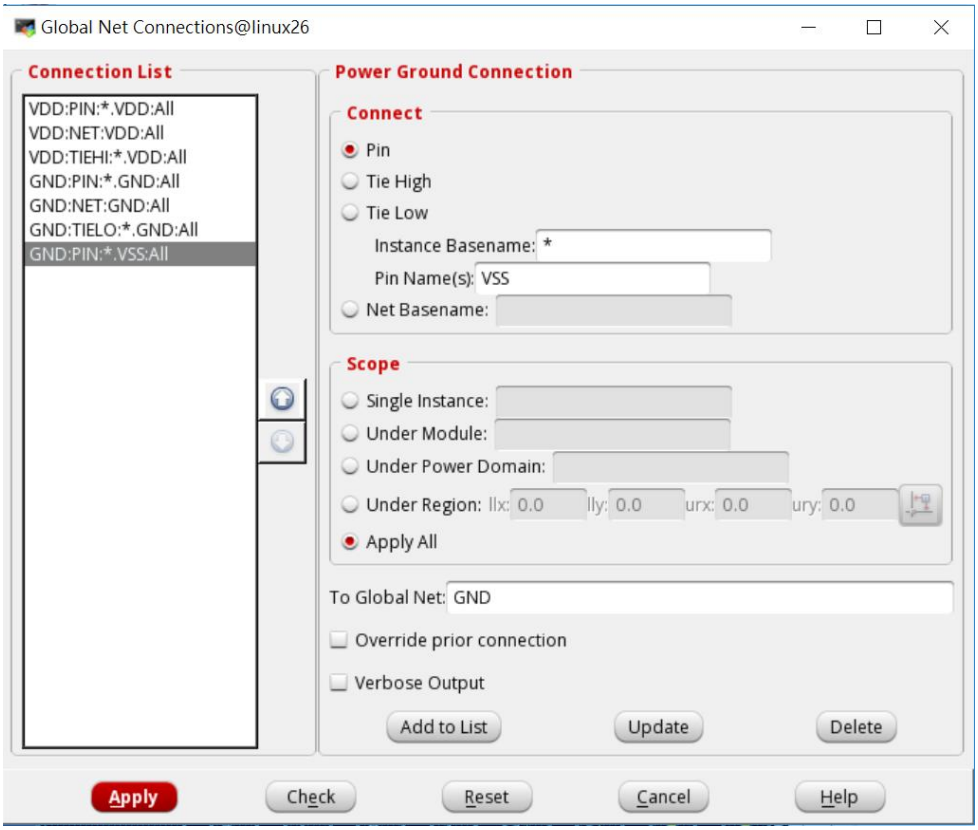
4. Power Planning

4.1. Connect/Define Global Net

1. In innovus menu, open ***Power -> Connect Global Nets…***
2. Add all VDD pins to Connection List:
   - **a. Connect** ◆Pins: VDD
   - **b. Scope** ◆Apply All
   - **c.** To Global Nets: VDD
   - **d.** Click Add to List button

3. Add all VDD nets to Connection List:
   a. Connect ◆Net Basename: VDD
   b. Scope ◆Apply All
   c. To Global Nets: VDD
   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: VDD
   d. Click Add to List button
5. Add all GND pins to Connection List:
   a. Connect ◆Pins: **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

8. Add all VSS pins to Connection List: **(This step is added for hard macro design)**
   a. Connect ◆Pins: VSS
   b. Scope ◆Apply All
   c. To Global Nets: GND
   d. Click Add to List button
9. Apply the connection list and check:
   e. Click **Apply** button
   f. Click **Check** button
   g. Click **Cancel** button



4.2. Add Power Rings

10. In innovus menu, open ***Power -> Power Planning -> Add Ring…***
11. In the **Basic** tab, fill the following field:

a. Fill in Net(s) names: GND VDD

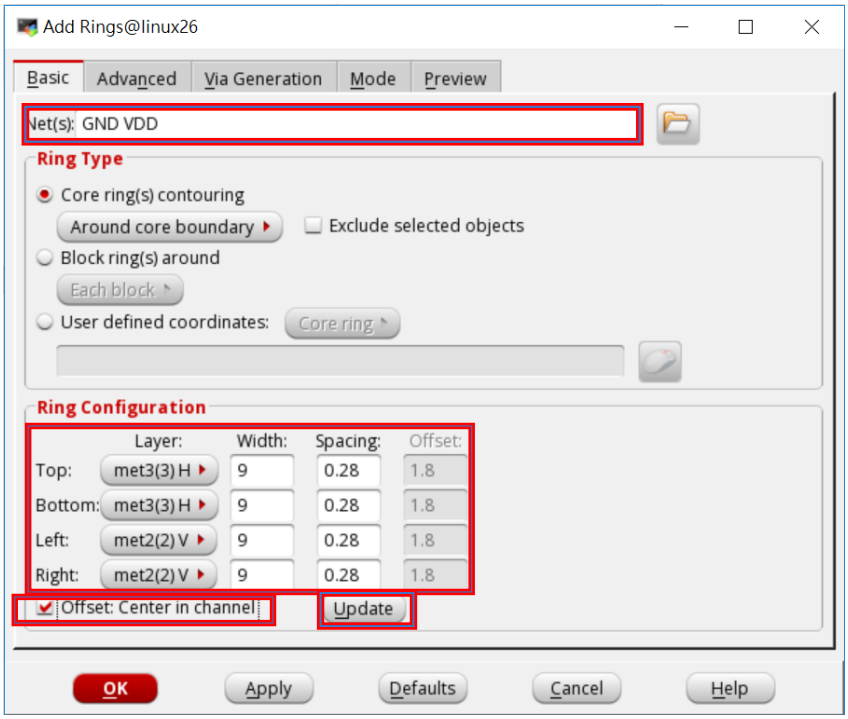
b. Specify metal layers and width

- Top Layer: **metal3 H** Width: 9
- Bottom Layer: **metal3 H** Width: 9
- Left Layer: **metal2 V** Width: 9
- Right Layer: **metal2 V** Width: 9

c. Click **Update** button

d. Specify offset

- ◆Center in channel



12. In the **Advanced** tab, fill the following field:

e. Configure wire group

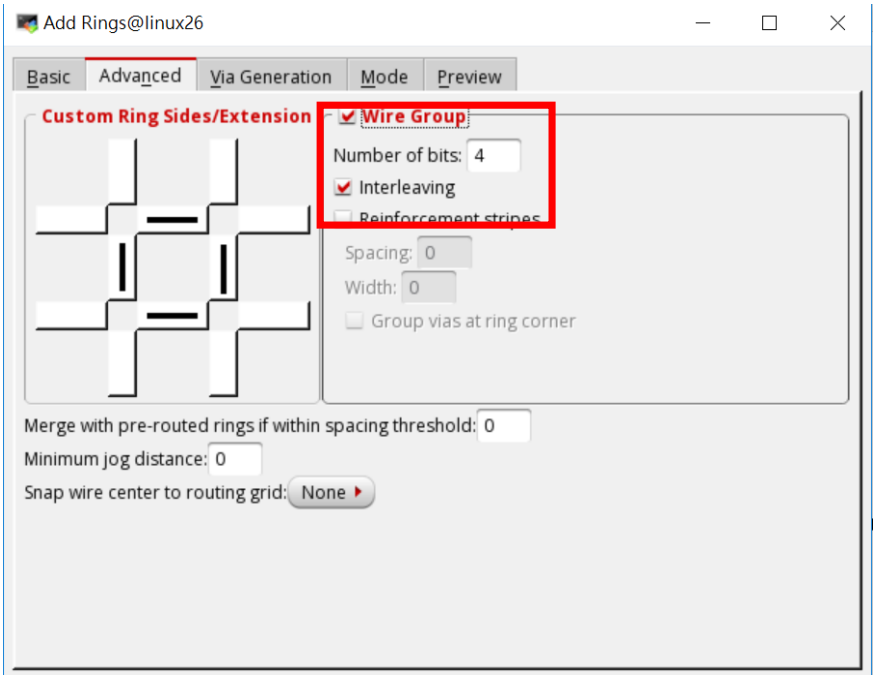
- ◆wire group
- ◆Interleaving
- Number of bits: 4

13. Apply the specification:

f. Click **Apply** button

- Check if the ring is correctly created. If not, click **undo** button (in innovus toolbar) and repeat step 2~4 again.

g. Click **Cancel** button



4.3. Connect Core Power Pin

1. In innovus menu, open ***Route -> Special Route…***

2. Connect core power:

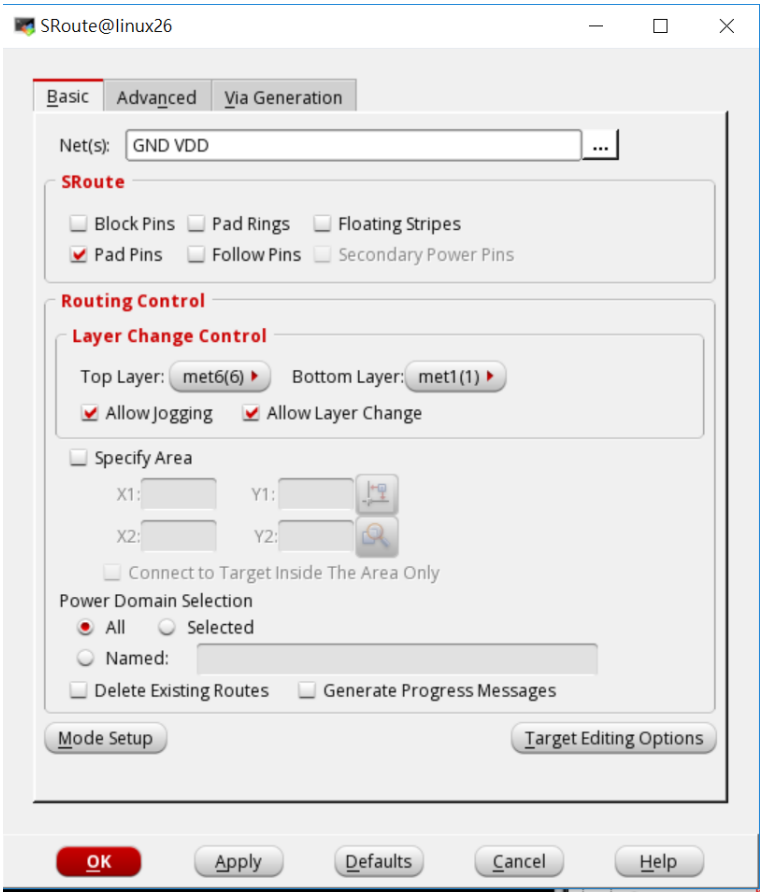
a. Fill in Net(s) names: GND VDD

b. Set the following configuration

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

c. If you want to specify core power layer, adjust the top and bottom layer in the **layer change control** section.

d. Click **OK** button



4.4. Power Planning (Add Stripes)

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

2. Create vertical power stripes:

a. Specify metal layer, width, direction and spacing

- Net(s): GND VDD
- Layer: **metal4**

- Direction: ◆Vertical
- Width: 2
- Click **Update** Button

b. Specify set-to-set distance

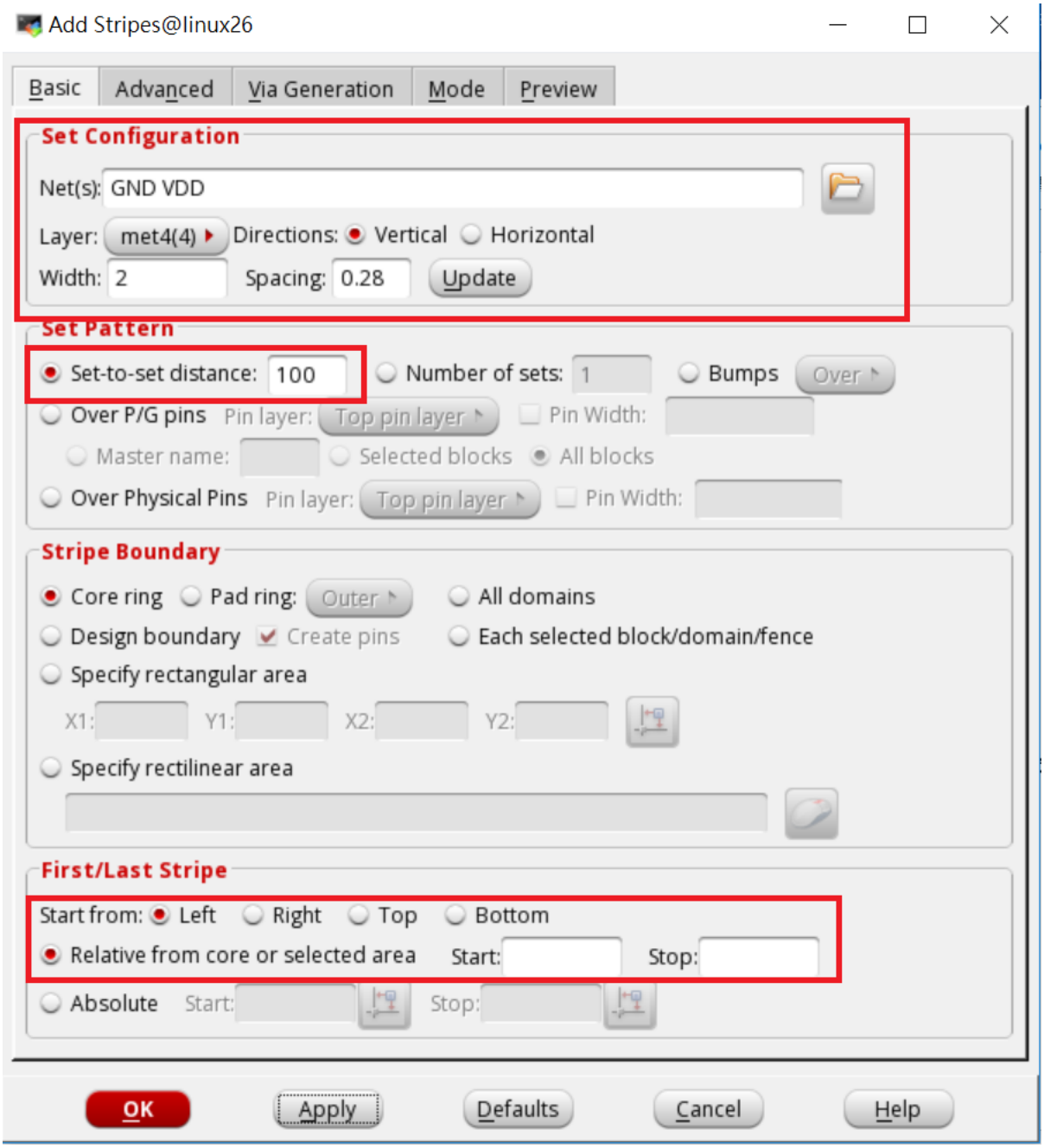
- ◆Set-to-set distance: **100**

c. Specify locations by ◆Relative from core or selected area

NOTE: You can choose another way to add stripes; just remember to meet the constraint given by TA.

d. Click **Apply** button

- Check if the stripes are correctly created. If not, click **undo** button (in innovus toolbar) and repeat step a~d again.



3. Create horizontal stripes:

a. Specify metal layer, width, direction and spacing

- Layer: **metal5**

- Direction: ◆Horizontal
- Width: 4
- Click **Update** Button

b. Specify set-to-set distance

- ◆Set-to-set distance: **80**

c. Specify locations by ◆Relative from core or selected area

- start from: bottom

NOTE: You can choose another way to add stripes; just remember to meet the design constraint.

4. Create one more vertical power stripes: **(optional)(add if needed)**

a. Specify metal layer, width, direction and spacing

- Net(s): GND VDD
- Layer: **metal6**
- Direction: ◆Vertical
- Width: 4
- Click **Update** Button

b. Specify set-to-set distance

- ◆Set-to-set distance: **100**

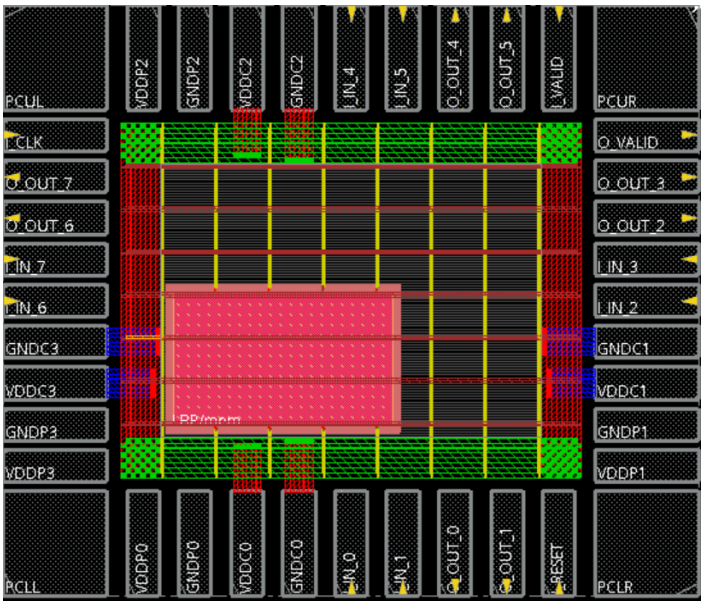
c. Specify locations by ◆Relative from core or selected area

- X star from left: 50

d. Click **Apply** button

- Check if the stripes are correctly created. If not, click **undo** button (in innovus toolbar) and repeat step a~d again.

e. Click **Cancel** button



4.5. Connect Standard Cell Power Line

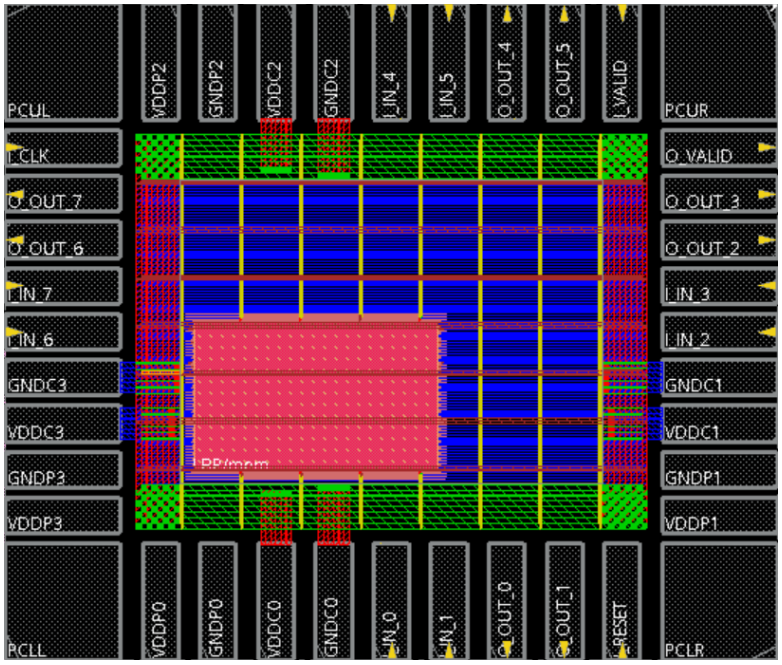
1. (Make sure hard macro blockage had been added.)

2. In innovus menu, open ***Route -> Special Route…***
3. Connect core power:

a. Set the following configuration

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

b. Click **OK** button



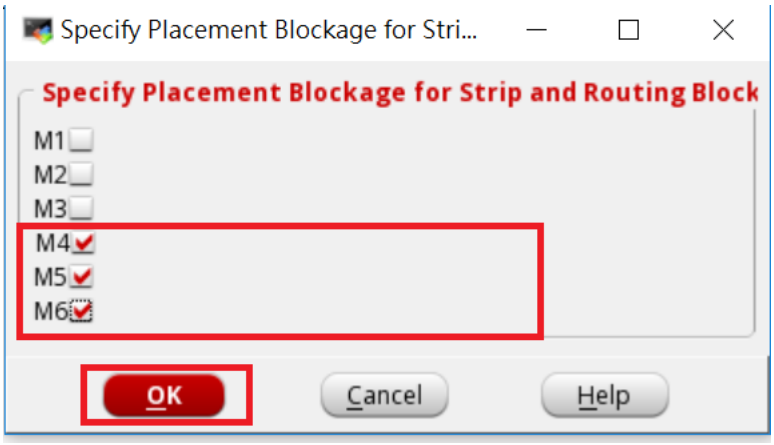
5. Place Standard Cells

1. In innovus menu, open ***Place -> Specify -> Placement Blockage***
2. Specify placement blockage for stripes

a. Specify placement blockage under metal4, metal5 and metal6

- ◇M1
- ◇M2
- ◇M3
- ◆M4
- ◆M5
- ◆M6

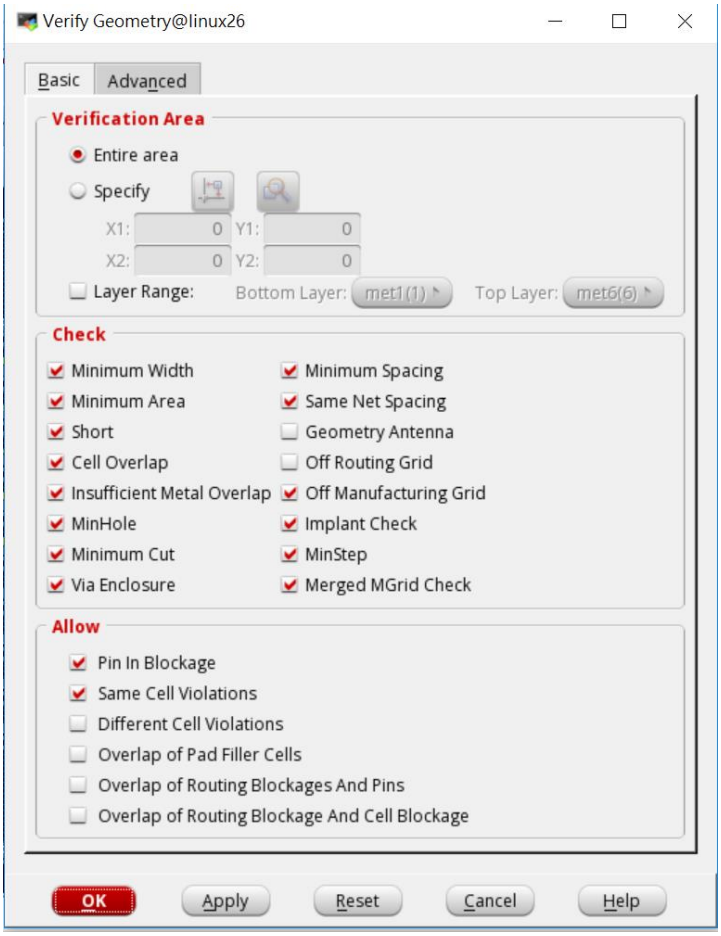
b. Click **OK** button



Verify

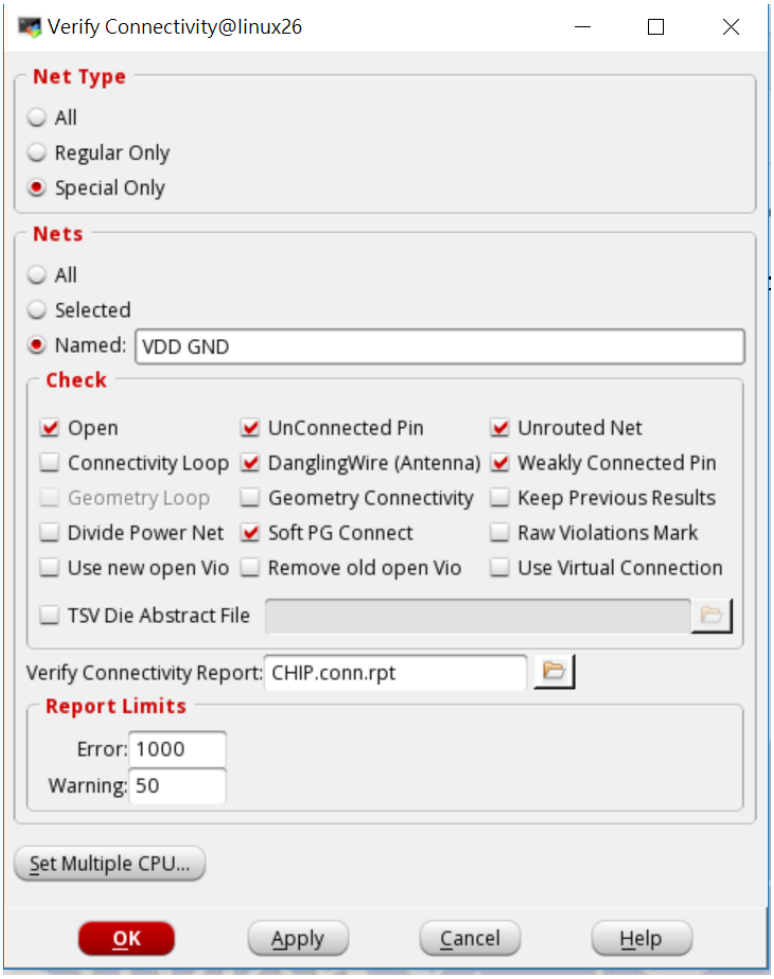
(I) In innovus menu, open ***Verify -> Geometry***

Check routing for DRC error



(II) In innovus menu, open ***Verify -> Verify Connectivity***

Check routing for LVS error



Tips: If there are **dangling wire** at GND/VDD wire, you can just delete the dangling wire. Then the LVS error will be fixed.

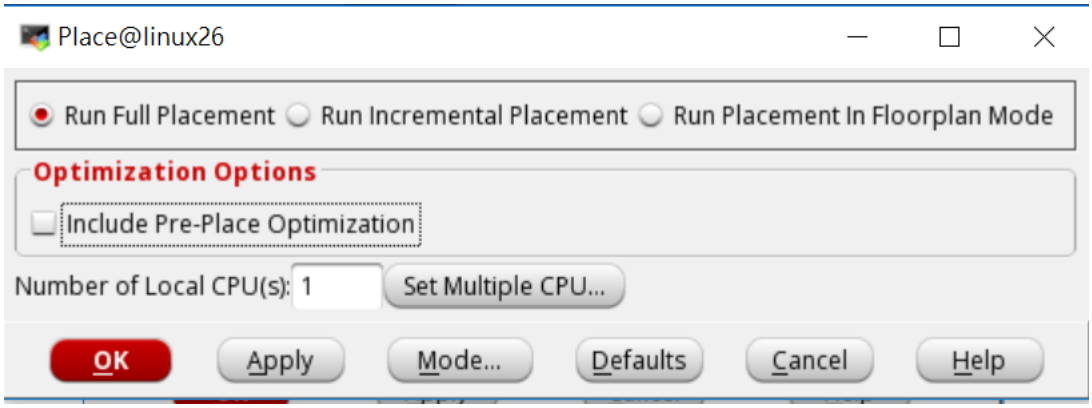
3. In innovus menu, open ***Place -> Place Standard Cell…***
4. Configure place options:

a. Run Full Placement

b. Optimization Options

- ◇Include Pre-Place Optimization

c. Click **OK** button



6. In-Place Optimization (IPO)

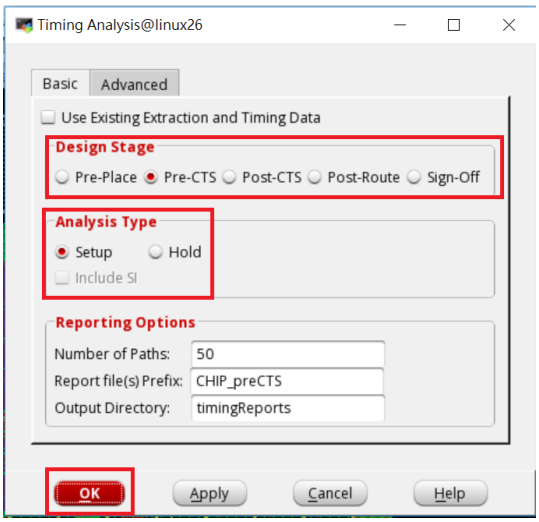
- Before Clock Tree Synthesis

1. In innovus menu, open ***Timing -> Report Timing…***
2. Perform First Innovus trial route to model the interconnection RC effects

   a. Design Stage ◆pre-CTS

   b. Analysis Type ◆Setup

   c. Click **OK** button



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_reg2reg.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

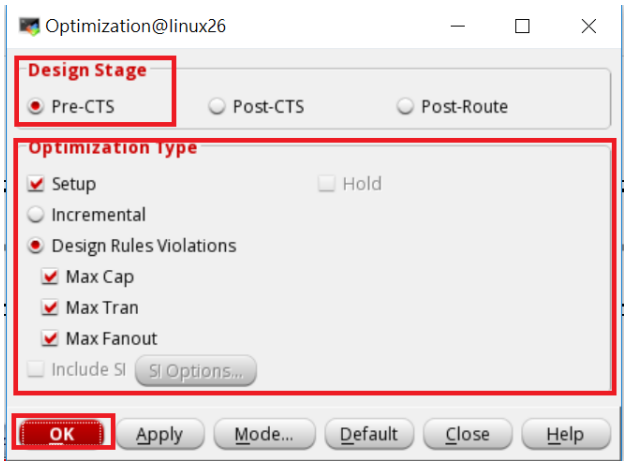
5. Perform pre-CTS IPO

   a. Design Stage ◆pre-CTS

   b. Optimization Type

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

   c. Click **OK** button



7. Clock Tree Synthesis (CTS)

1. In innovus command prompt, execute the following commands:
   - Source A20180518015-OT-DUTY.tcl

8. In-Place Optimization (IPO)

- After Clock Tree Synthesis

1. In innovus menu, open ***Timing -> Report Timing***

2. Perform First Innovus trial route to model the interconnection RC effects

   a. Design Stage ◆post-CTS

   b. Analysis Type ◆Setup

   c. Click **OK** button

3. After CTS, further timing optimization is performed to meet timing constraints if there is negative timing slack or DRVs. Open ***ECO -> Optimize Design..*** in innovus menu

4. 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

5. Verify if the hold time constraint is satisfied or not. Open ***Timing -> Report Timing*** in innovus menu

   a. Design Stage ◆post-CTS

   b. Analysis Type ◆Hold

   c. Click **OK** button

6. If hold time slack is negative, open ***ECO -> Optimize Design..*** in innovus menu

7. Perform post-CTS IPO for hold time fixing

   a. Design Stage ◆post-CTS

   b. Optimization Type
   - ◇Setup ◆Hold

◇Design Rule Violation

   c. Click **OK** button
   - See timing reports in **timingReports/** directory, For detail path report, see **CHIP_postCTS_reg2reg.tarpt** (setup time check) and **CHIP_postCTS_reg2reg_hold.tarpt** (hold time check). DRV violations report files: ***.cap**, ***.fanout**, and ***.tran**.

9. Add PAD Filler

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

- **addIoFiller -cell PFILL -prefix IOFILLER**
- **addIoFiller -cell PFILL_9 -prefix IOFILLER**
- **addIoFiller -cell PFILL_1 -prefix IOFILLER**
- **addIoFiller -cell PFILL_01 -prefix IOFILLER –fillAnyGap**
- PAD filler must be added before detail route, or there may have some DRC/LVS violations after PAD filler insertion

10. SI-Prevention Detail Route (NanoRoute)

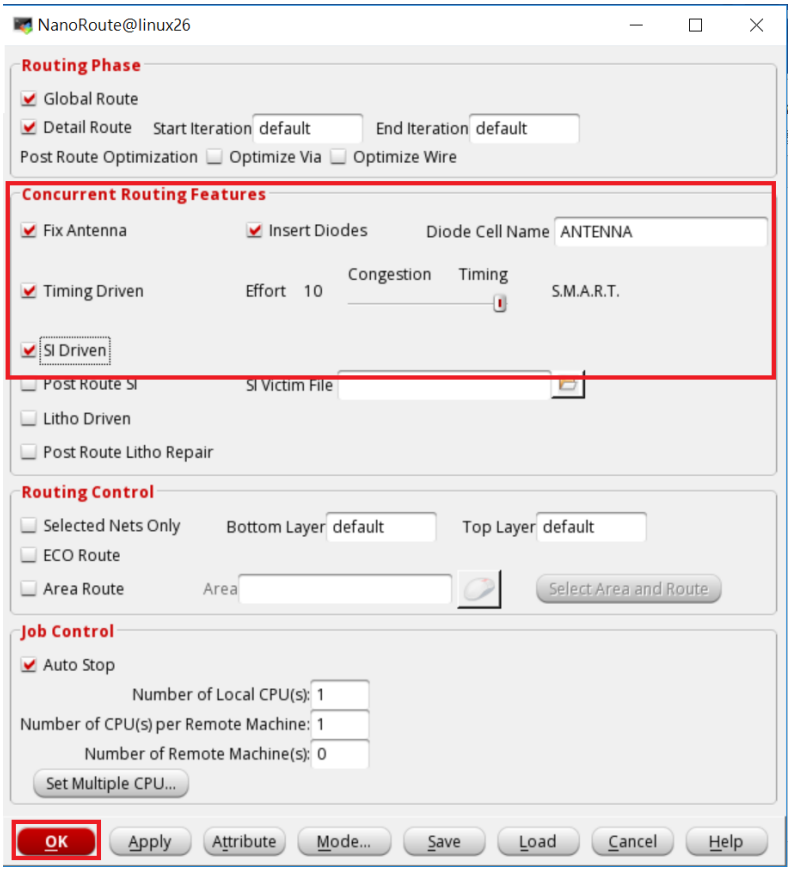
e. In innovus menu, open ***Route -> NanoRoute -> Route***

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

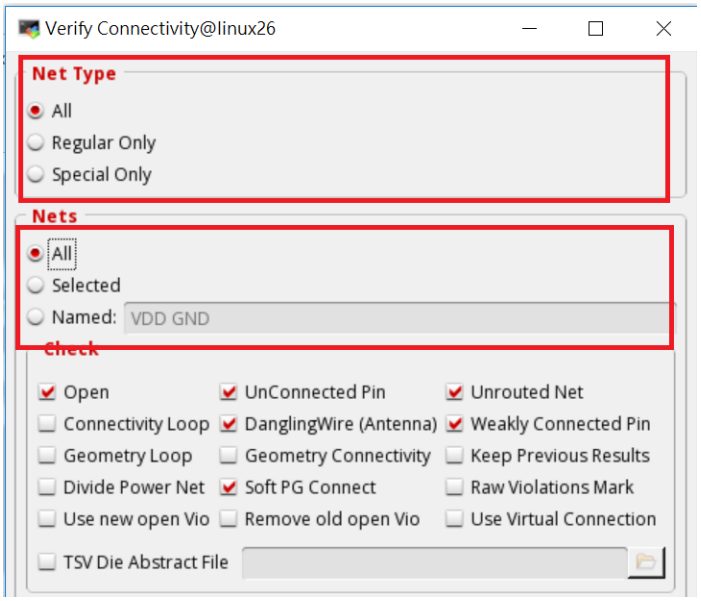
a. Concurrent routing features

- ◆Fix Antenna
- ◆Insert DiodesDiode Cell Name: ANTENNA
- ◆Timing DrivenEffort: 10
- ◆SI Driven

b. Click **OK** button



g. In innovus menu, open ***Verify -> Verify Connectivity***



h. Check routing for LVS error

a. Click **OK** button

- If you see any violations, routing result is not correct. (LVS error)
- Check CHIP.conn.rpt to fix LVS errors.

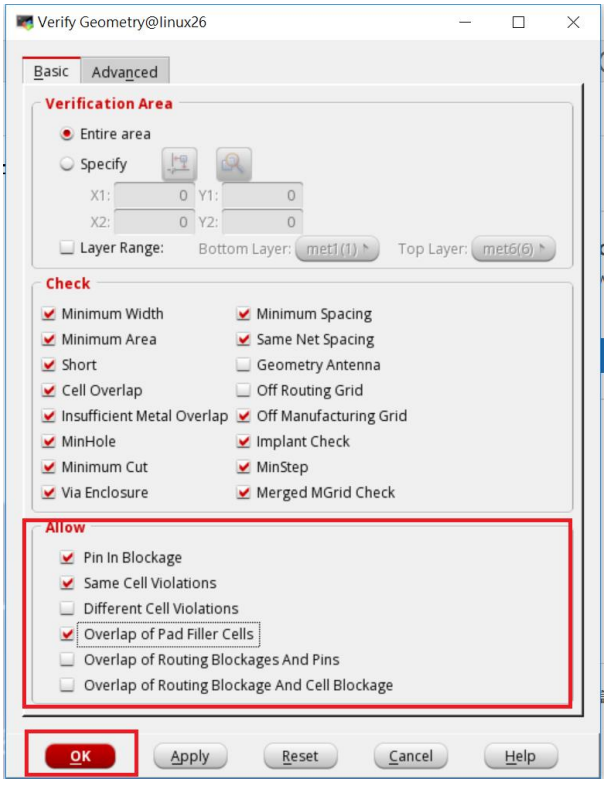
i. In innovus menu, open ***Verify -> Geometry***

j. Check routing for DRC error

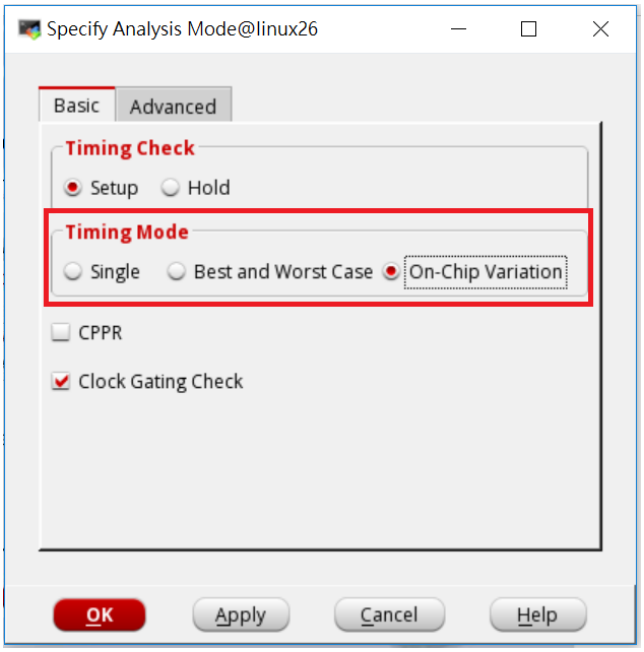
a. Allow ◆Overlap of Pad Filler Cells

b. Click **OK** button

- If you see any violations, routing result is not correct. (DRC error)



k. In innovus menu, open **Tools->Set mode->Specify Analysis Mode**



11. In-Place Optimization (IPO)

- After Detail Route

1. In innovus menu, open ***Timing -> Report Timing…***

2. Perform First innovus RC extraction for timing calculation

   a. Design Stage ◆post-Route

   b. Analysis Type ◆Setup

   c. Click **OK** button

3. Further timing optimization is performed to meet timing constraints if there is negative timing slack or DRVs. Open ***ECO -> Optimize Design..*** in innovus menu

4. 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

5. Verify if the hold time constraint is satisfied or not. Open ***Timing -> Report Timing*** in innovus menu

   a. Design Stage ◆post-Route

   b. Analysis Type ◆Hold

   c. Click **OK** button

6. If hold time slack is negative, open ***ECO -> Optimize Design…*** in innovus menu

7. Perform post-Route IPO for hold time fixing

a. Design Stage ◆post-Route

b. Optimization Type

- ◇Setup ◆Hold
- ◇Design Rule Violation

c. Click **OK** button

See timing reports in **timingReports/** directory, For detail path report, see **CHIP_postRoute_reg2reg.tarpt** (setup time check) and **CHIP_postRoute_reg2reg_hold.tarpt** (hold time check). DRV violations report files: ***.cap**, ***.fanout**, and ***.tran**.

12. In-Place Optimization (consider crosstalk effects)

1. In innovus menu, open ***Timing ->Report Timing***

2. Use **Celtic** to analyze the cross talk effects.

a. Design Stage ◆Sign-Off

b. Analysis Type

- ◆Setup
- ◆Include SI
- Reprot file(s) Prefix: CHIP_postRoute_SI

c. Click **OK** button

3. Verify if the hold time constraint is satisfied or not. Open ***Timing ->Report Timing*** in innovus menu

a. Design Stage ◆Sign-Off

b. Analysis Type

- ◆Hold
- ◆Include SI
- Reprot file(s) Prefix: CHIP_postRoute_SI

c. Click **OK** button

See timing reports in **timingReports/** directory, For detail path report, see **CHIP_postRoute_si_reg2reg.tarpt** (setup time check) and **CHIP_postRoute_si_reg2reg_hold.tarpt** (hold time check). DRV violations report files: ***.cap**, ***.fanout**, and ***.tran**.

13. 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:

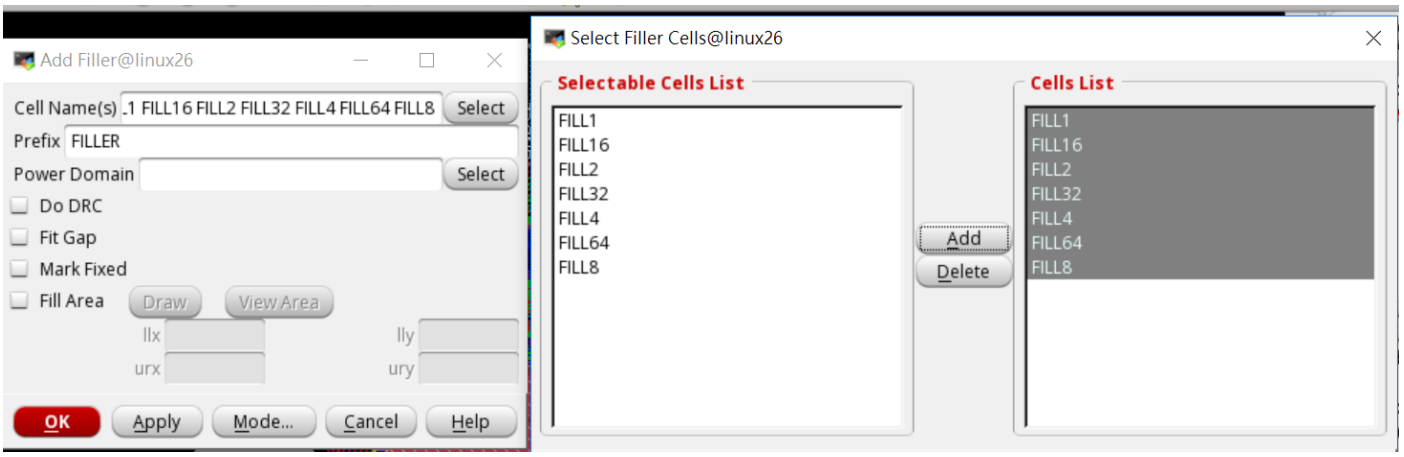
a. Click **Select** button

b. Select all core filler cells

c. Click **Add** button

d. Click **Close** button

e. Click **OK** button



14. Stream Out and Write Netlist

1. Write the worst-case design timing file CHIP.sdf :

a. At innovus command window:

- innovus 1> setAnalysisMode -checkType setup -analysisType bcwc -cppr none -clockGatingCheck 1 -timeBorrowing 1 -domain clock -useOutputPinCap 1
- innovus 1> write_sdf -edges check_edge CHIP.sdf

2. 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_LAYOUT.v**
- Click **OK** button

3. Save design chip layout for violations and performance checking at DEMO

a. In innovus menu, open ***File->Save Design***

- Data type: Innovus
- File name: **CHIP.inn**
- Click **OK** button

15. Post-Layout Gate-Level Simulation

1. Change to directory **06_POSTSIM**

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

3. After post-layout gate-level simulation of CHIP.v, CHIP.fsdb is generated

```
96.8M)
**ERROR: (IMPOAX-142):  Could not open shared library libinnovusoax22.so : /RAID2/eda/INNOVUS17.11/tools.lnx86/lib
ovusoax22.so: undefined symbol: _ZTIN12OpenAccess_418oaConflictObserverINS_8oaAppDefELj0EEE

**ERROR: (IMPOAX-142):  Could not open shared library libcdsSkillPcell.so : /RAID2/eda/INNOVUS17.11/tools.lnx86/li
sSkillPcell.so: undefined symbol: _ZTIN12OpenAccess_413oaFSComponentE

**ERROR: (IMPSYT-6245): Error , while saving MS constraint file.
Generated self-contained design CHIP.dat.tmp
```

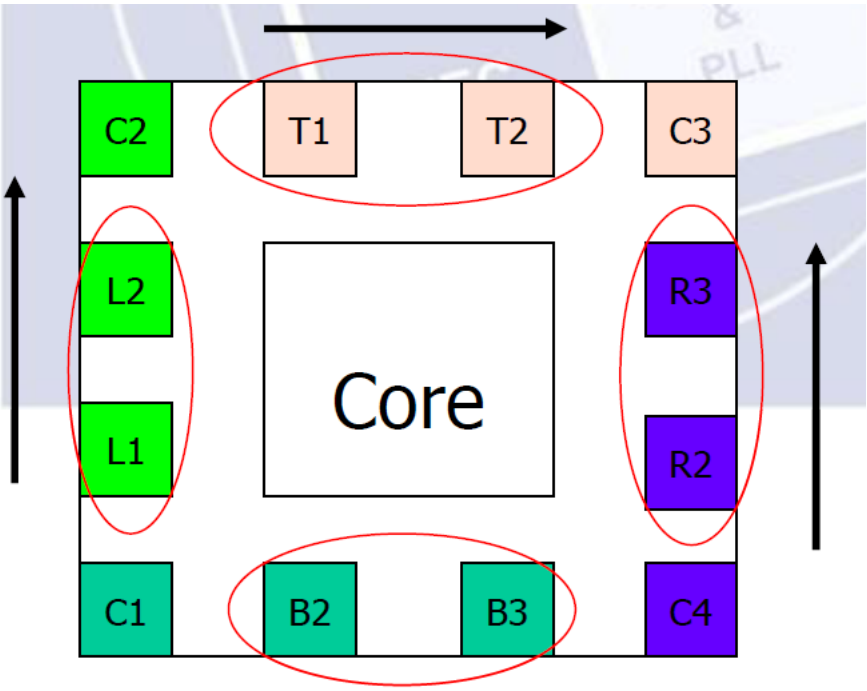
This error is OK when saving the file

Notice that some time CHIP.inn.dat may be empty then you must upload CHIP.inn.dat.tmp

APPENDIX A IO Pad assignment

CHIP.io

```
Version: 2

Pad: B2          S
Pad: B3          S

Pad: R2          E
Pad: R3          E          PFILL          # If this pad is a pad filler

Pad: T1          N
Pad: T2          N

Pad: L1          W
Pad: L2          W

Pad: C1          SW         PCORNER
Pad: C2          NW         PCORNER
Pad: C3          NE         PCORNER
Pad: C4          SE         PCORNER
```


C2
T1
T2
C3
L2
Core
R3
L1
R2
C1
B2
B3
C4