OASIS 3D Fault Tolerant Router Hardware Physical Design with TSV OASIS-3DFTRV1

Technical Report

Division of Computer Engineering
School of Computer Science and Engineering
University of Aizu

Contact: [d8141104, m5181121, benab] @ u-aizu.ac.jp

Edition: May 28, 2015

IMPORTANT NOTES

- This tutorial focuses mainly on the integration of TSV with 3D-OASIS-NoC router.
- It covers the TSV creation, synthesis, place and route, and LVS and DRC check.
- Post layout simulation and pad insertion steps are not included.
- Complete details about the execution of these two steps can be found in the previously made technical report [**Ref.1**].

TSV Physical Design Steps

1. Environment
2. TSV layout
3. Modify lef file
4. Place & Route
5. ACKNOWLEDGEMENT

Objectives

- After completing this tutorial you will be able to:
  1. Design TSV layout using Virtuoso
  2. Synthesize 3D-OASIS-NoC (3D-ONoC) router with TSV using Design-Compiler CAD tool
  3. Place & Route (P&R) 3D-ONoC router with TSV using Cadence SoC-Encounter
  4. Learn how to make TSV layout, the synthesis and P&R via:
     - The CAD Graphic User Interface
     - Tcl script

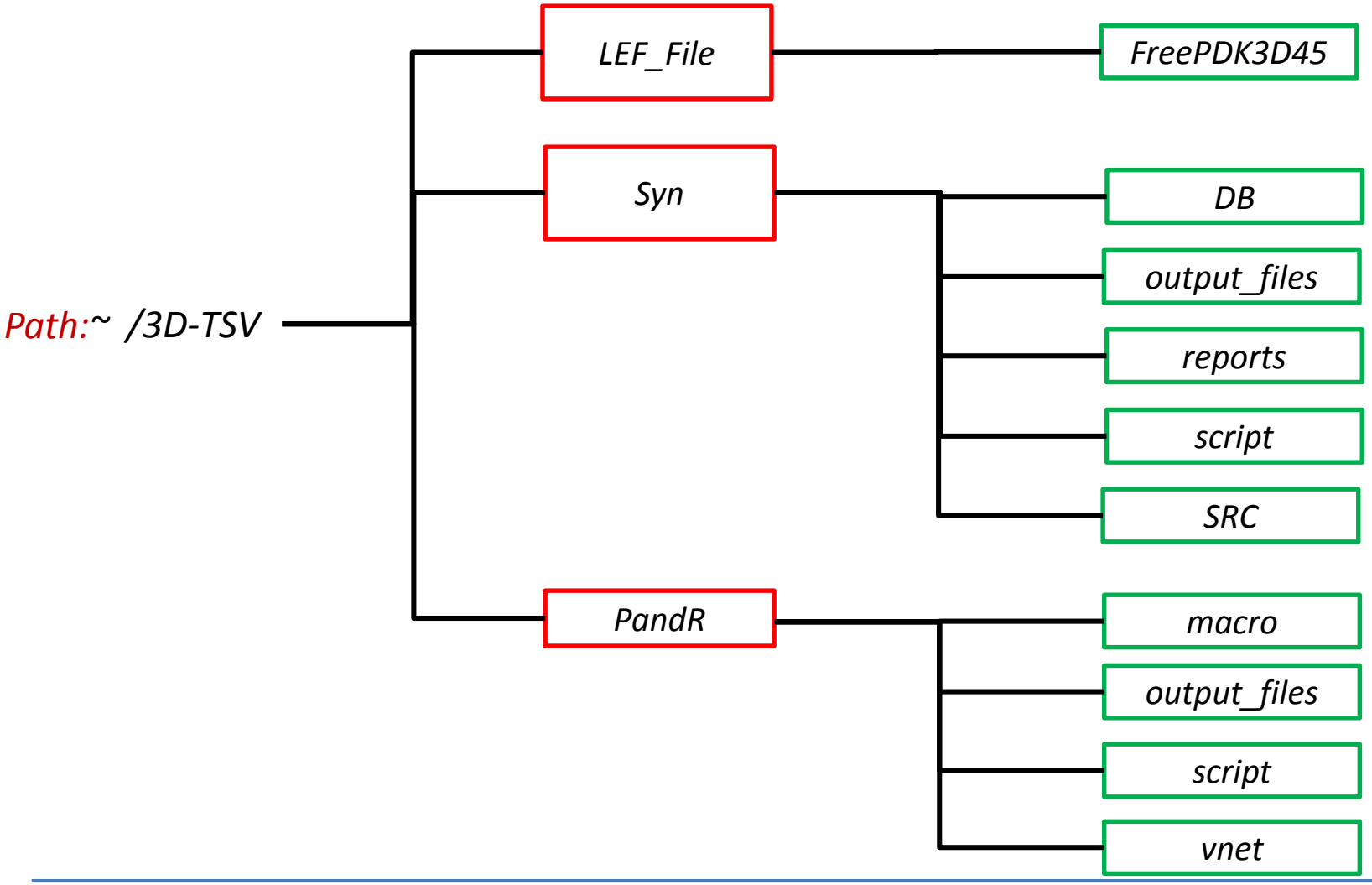
Prerequisite

- Understand the previously made “OASIS 3D-Router Hardware Physical Design” technical report [Ref.1].
- Study 3D-OASIS-NoC architecture and its main components [Ref.2].
- Read the Manual and Release Notes of the FreePDK3D45 design tool kit [Ref.3] and understand the aspects of the used TSV component.

[<== Back to Contents]

1. Environment

Tutorial directory structure

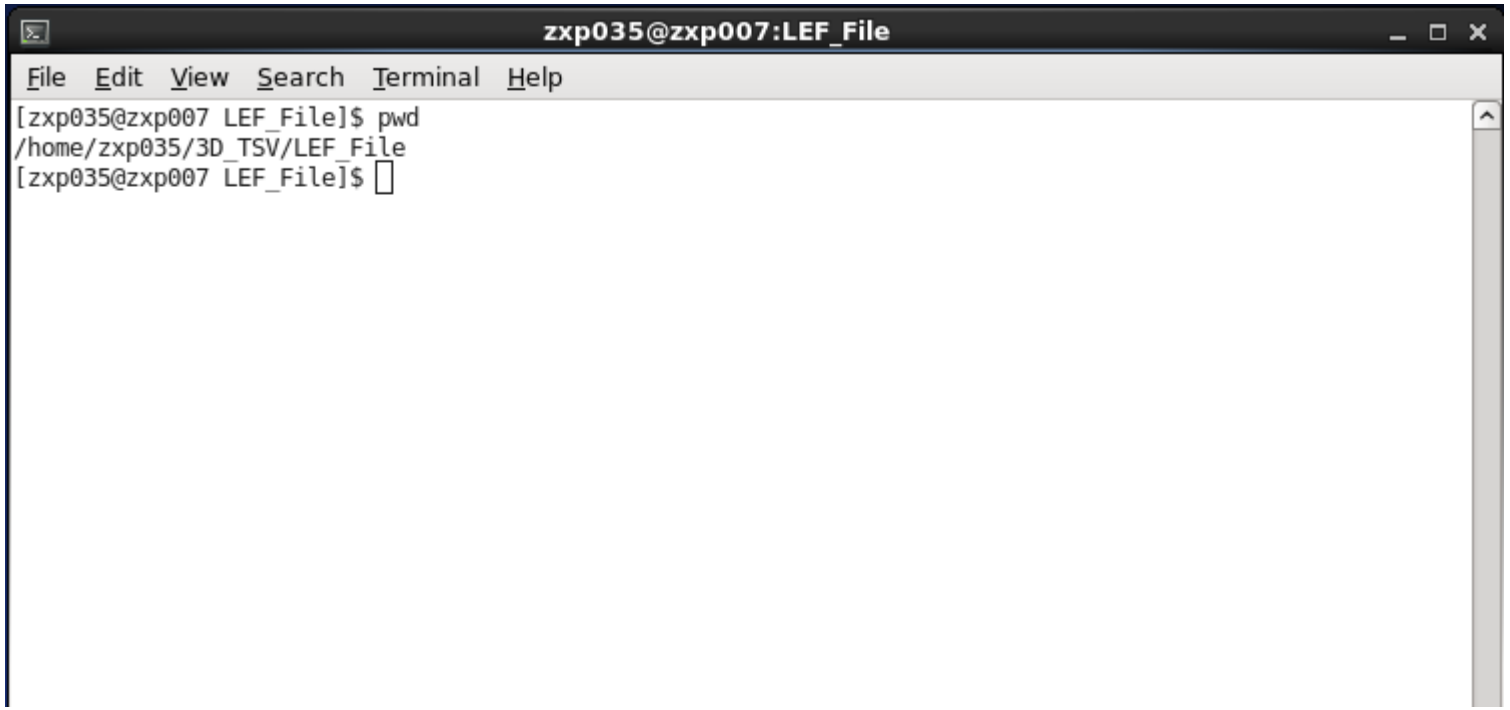


Environment: FreePDK3D45

- The Process Design Kit used in this tutorials are obtained from NCSU FreePDK3D45.
- Go to: http://www.eda.ncsu.edu/eda_registration.php
- Verification is required so enter your email address.
- You will receive an email with a link to download FreePDK3D45.
- Extract the downloaded archive and copy the *FreePDK3D45* folder in your working directory:

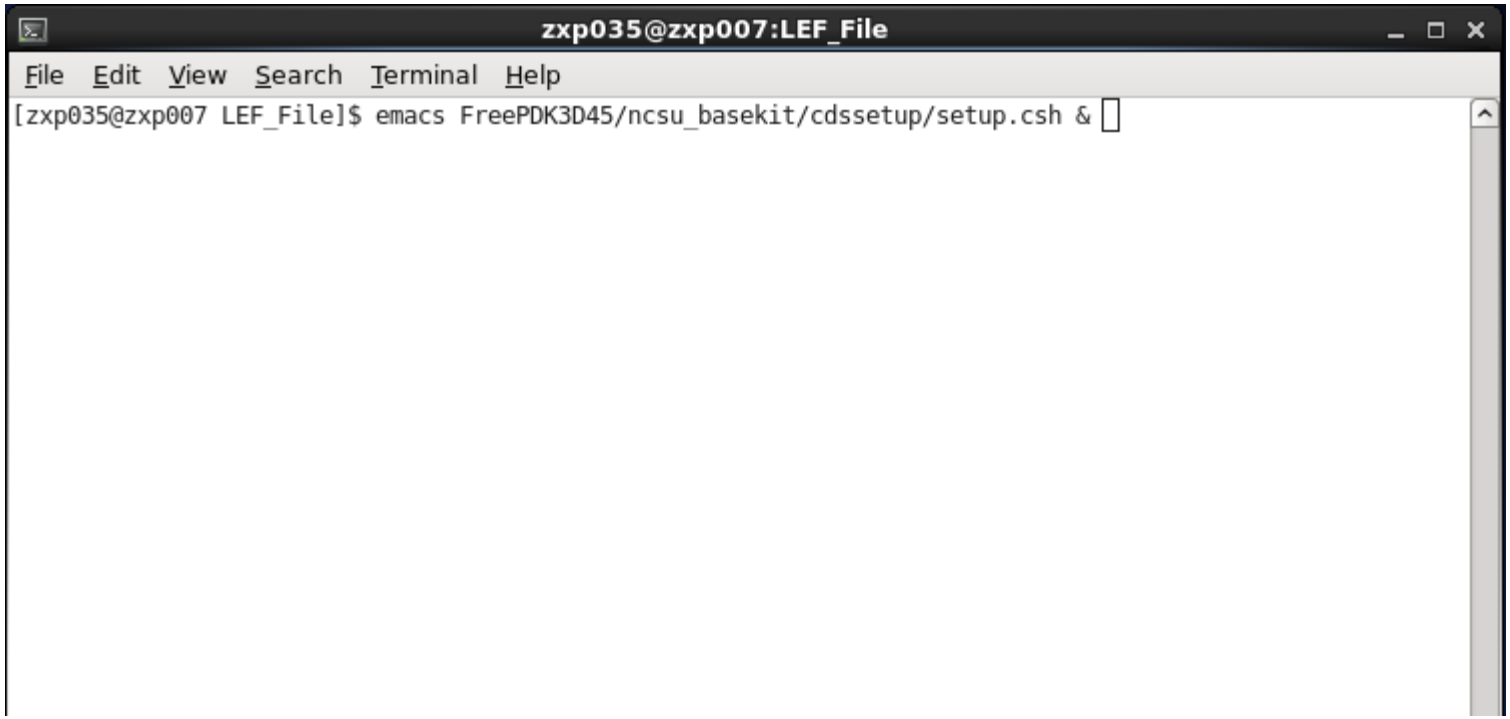
/home/zxp035/3D-TSV/LEF_File

Environment: Setup



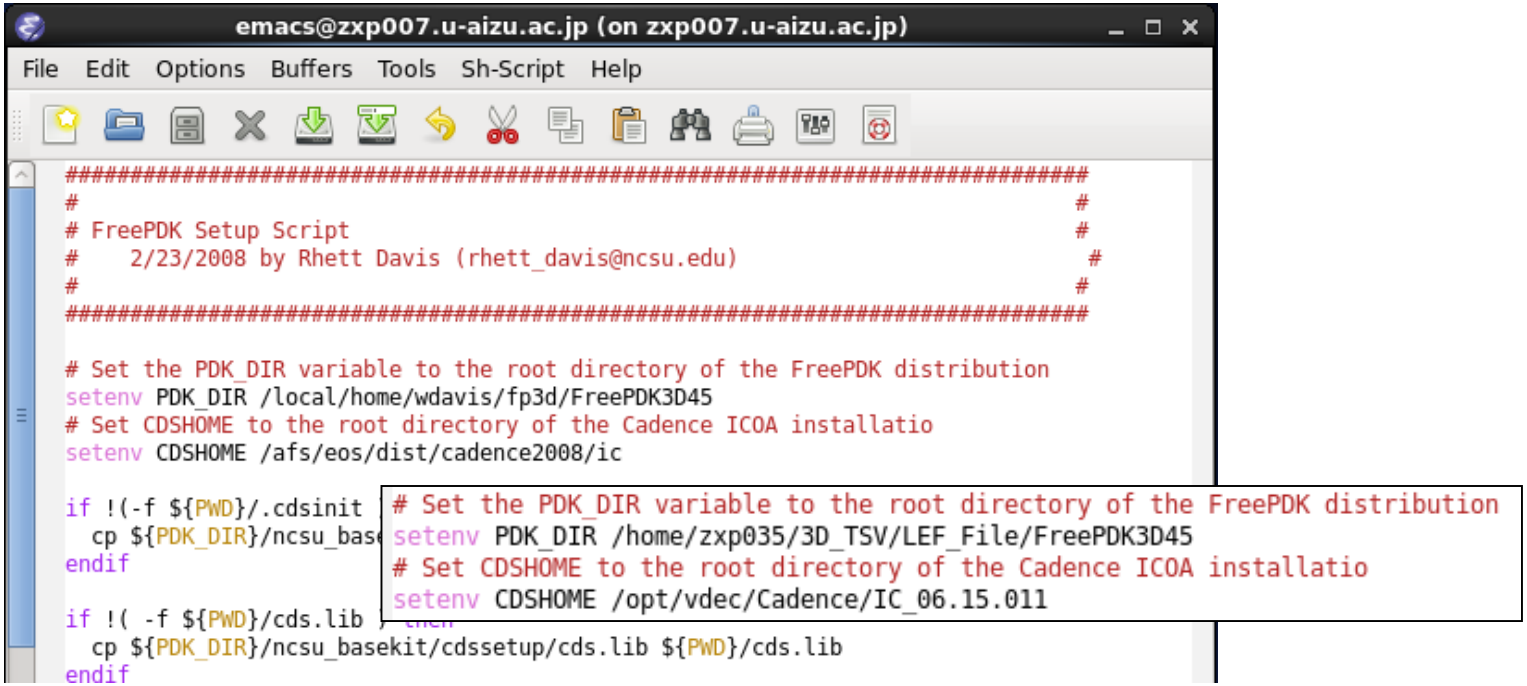
Go to ***/home/zxp035/3D-TSV/LEF_File*** where the TSV folder is located

Environment: Setup



Type **emacs FreePDK3D45/nscu_basekit/cdssetup/setup.csh &**

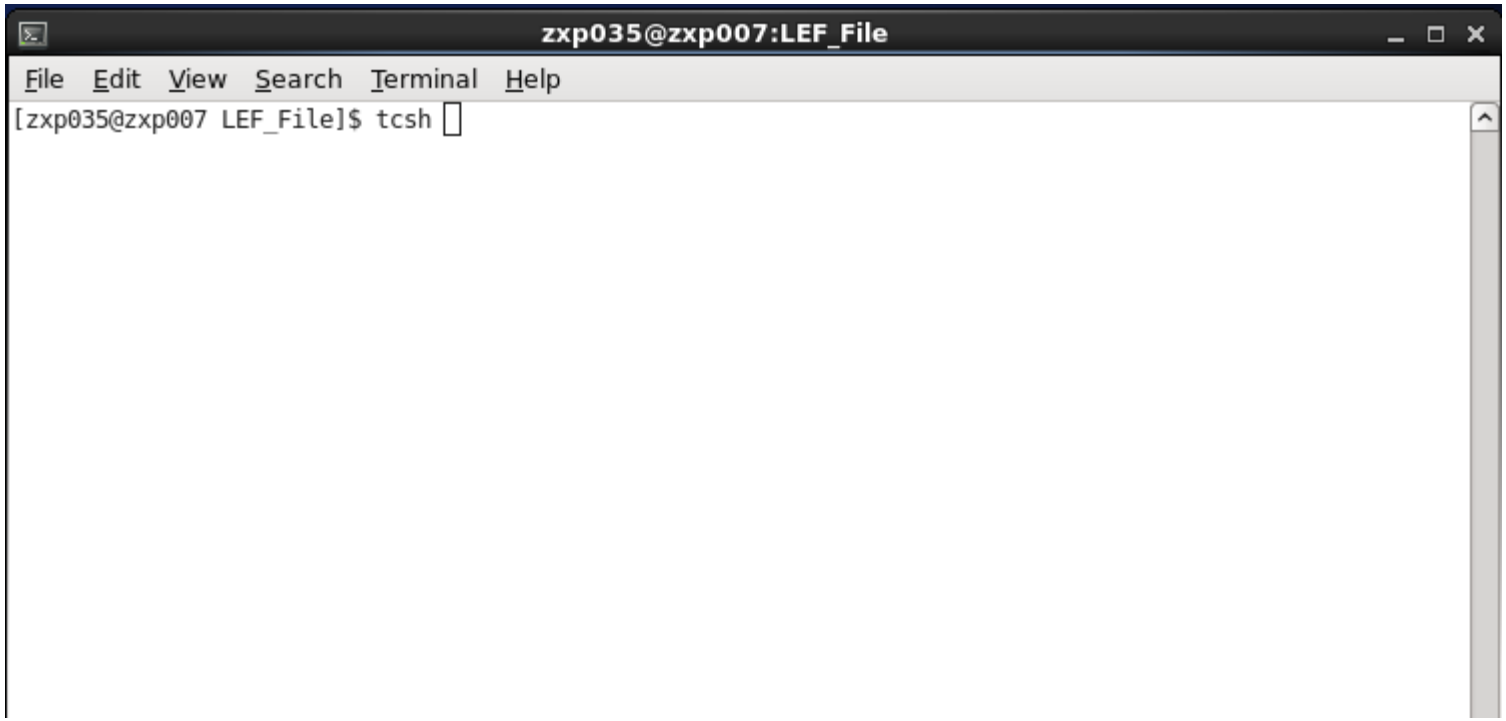
Environment: Setup



Change the following paths to link to the FreePDK3D45 libraries:

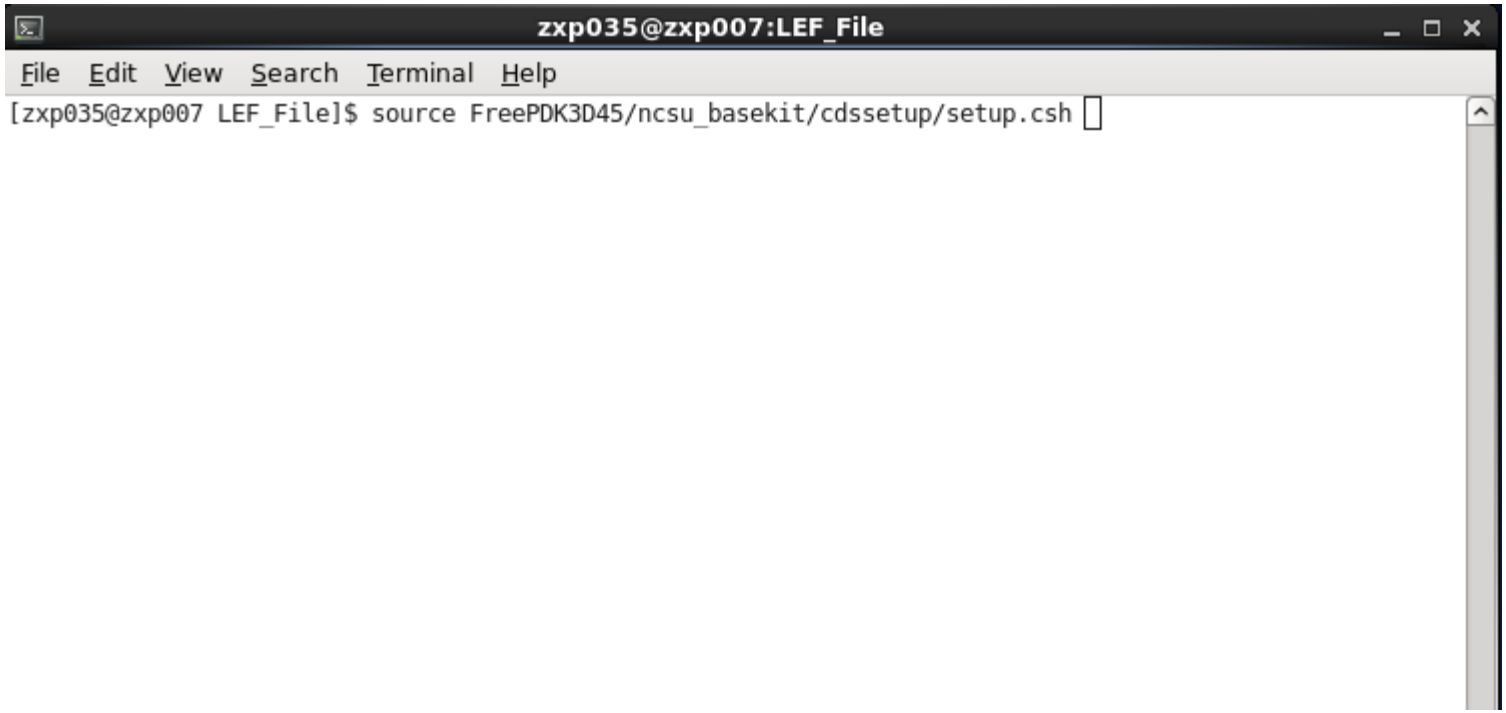
- /local/home/wdavis/fp3d/FreePDK3D45→ /home/zxp035/3D_TSV/LEF_File/FreePDK3D45
- /afs/eos/dist/cadence2008/ic → /opt/vdec/Cadence/IC_06.15.011

Environment: Setup



Initially *bash* will start, so type **“tcsh”** to start cshr

Environment: Setup

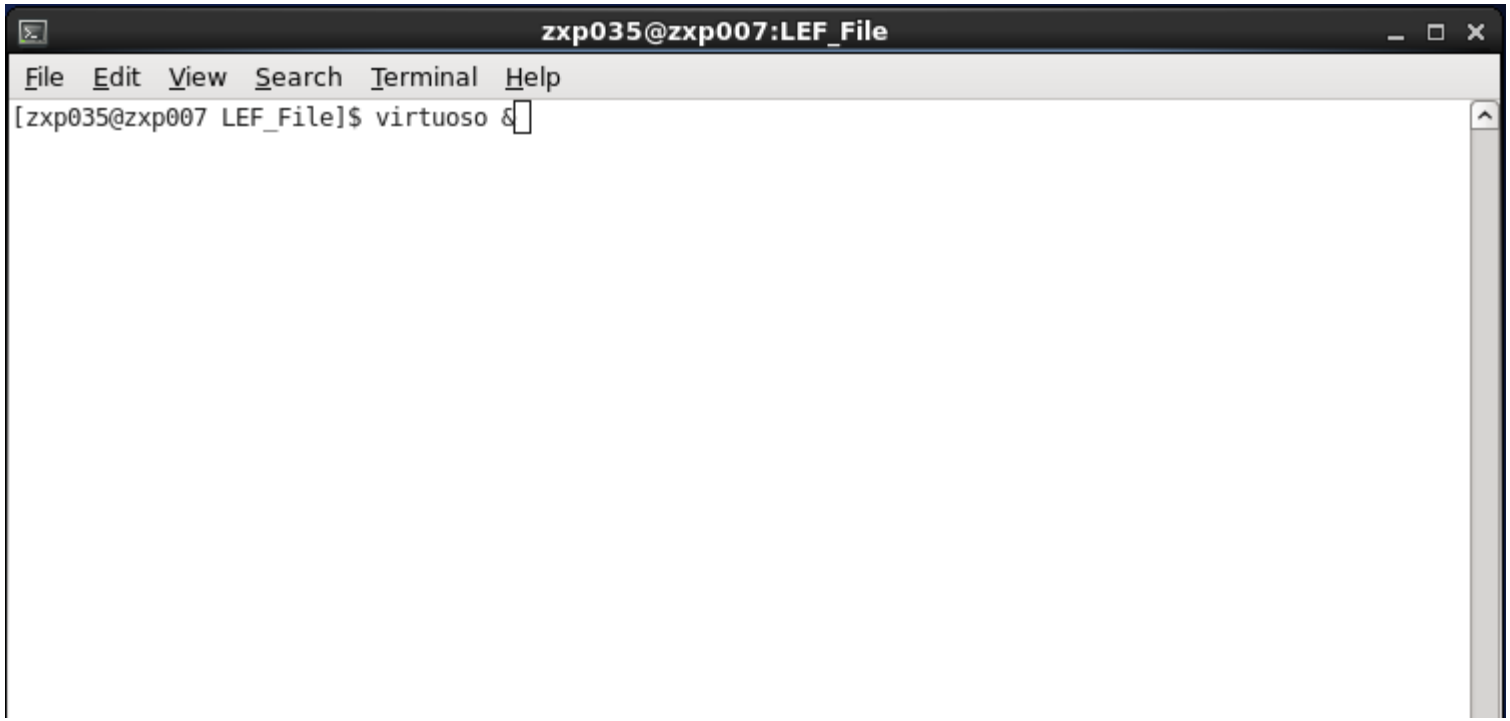


Type **source FreePDK3D45/ncsu_basekit/cdssetup/setup.csh**

<== Back to Contents

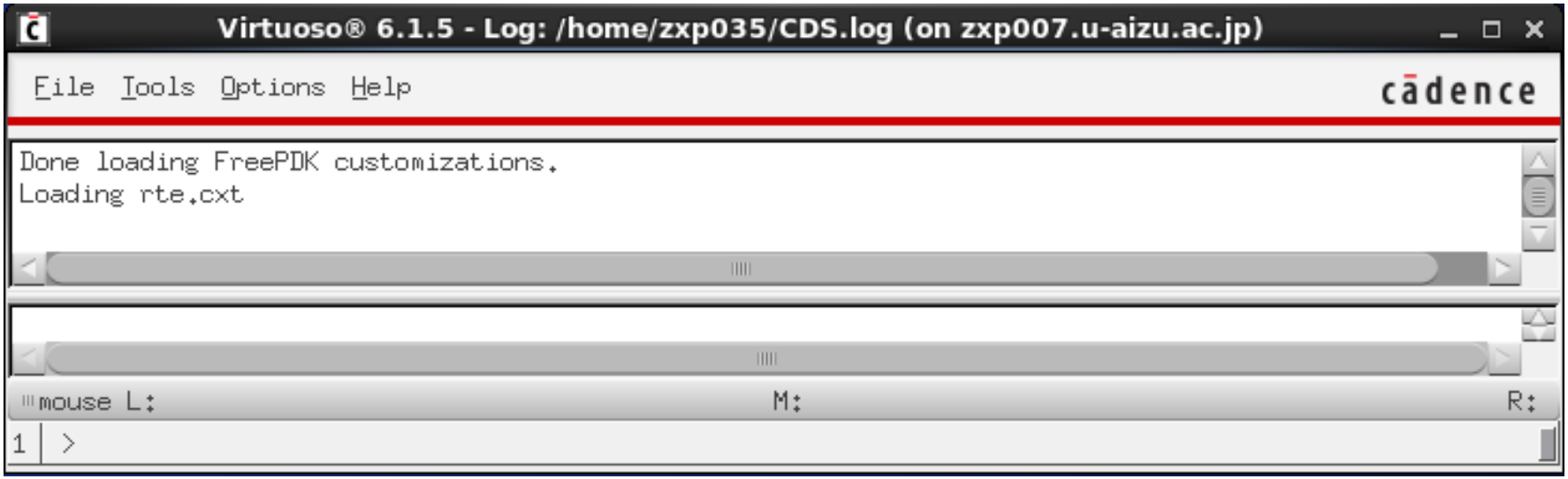
2. TSV Layout

Virtuoso: TSV layout steps



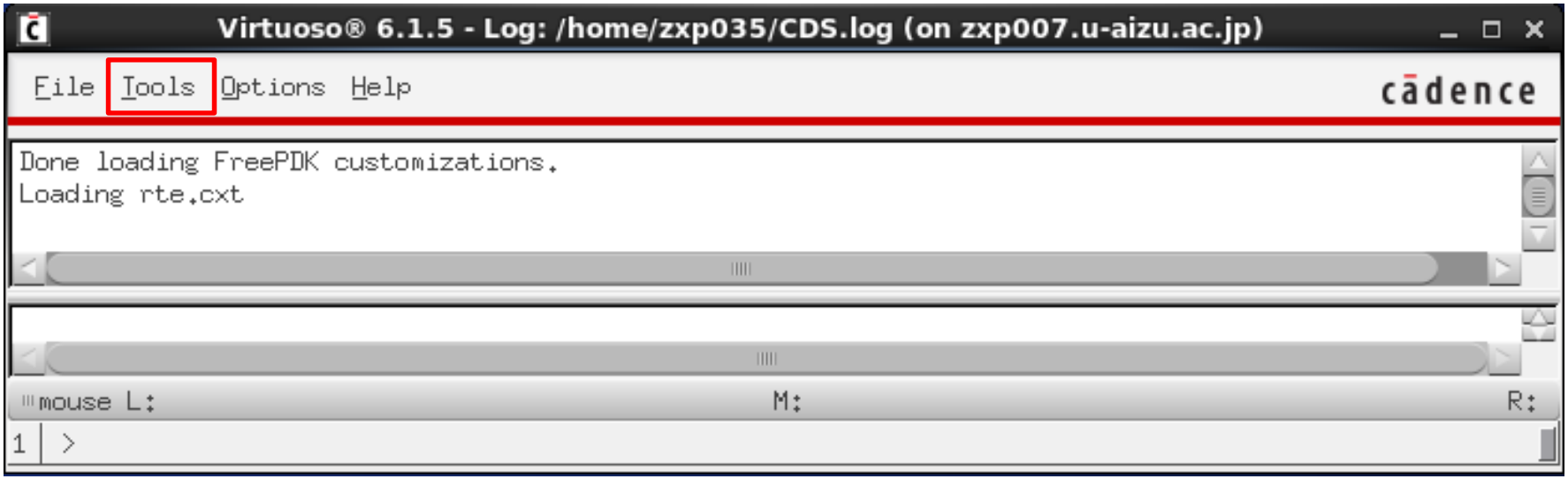
Type **virtuoso&** to start Virtuoso

Virtuoso: TSV layout steps



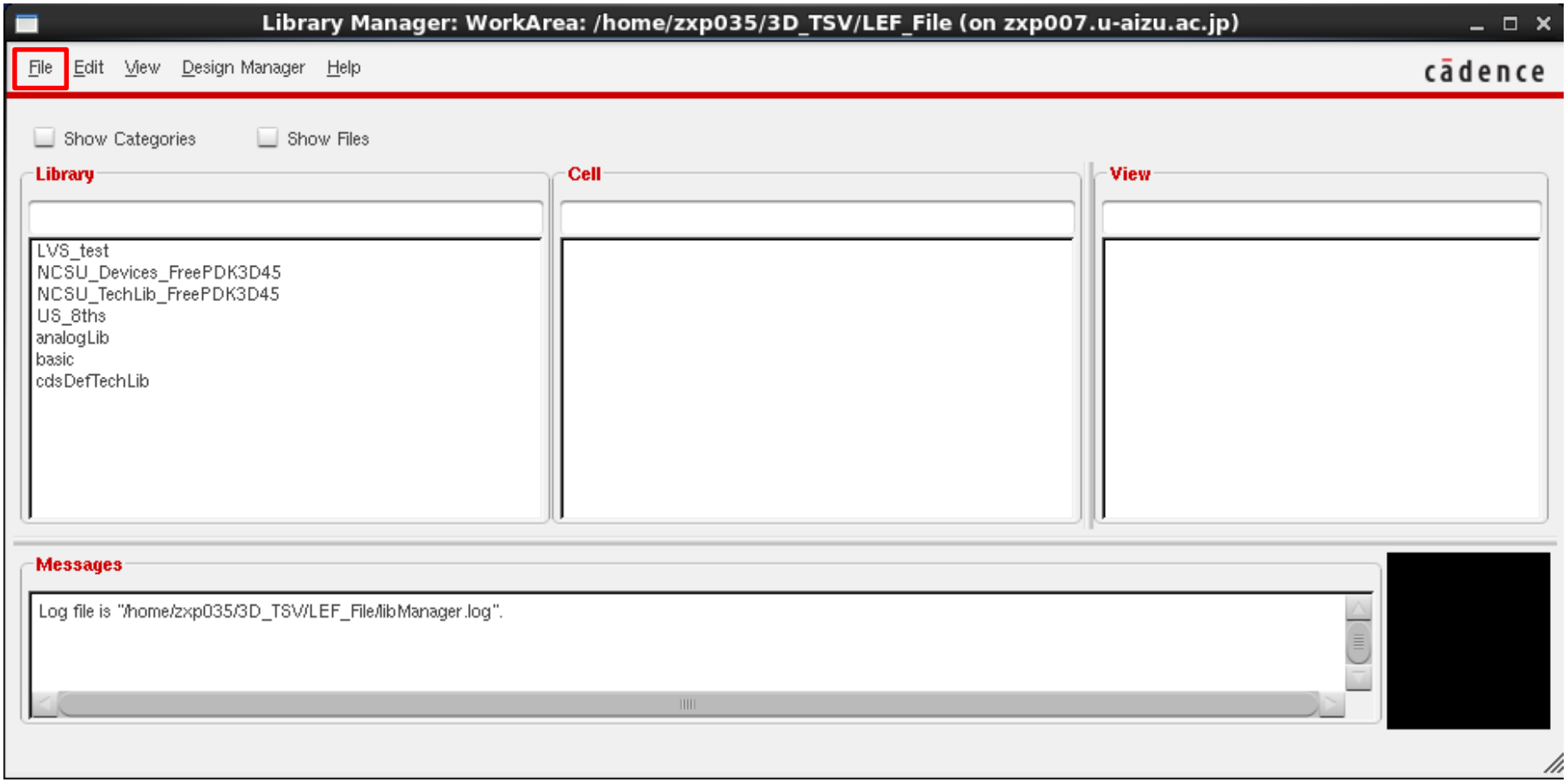
Welcome ***Command Interpreter Window(CIW)***

Virtuoso: TSV layout steps



First, we should create a library.
Click **Tools->Library Manager**

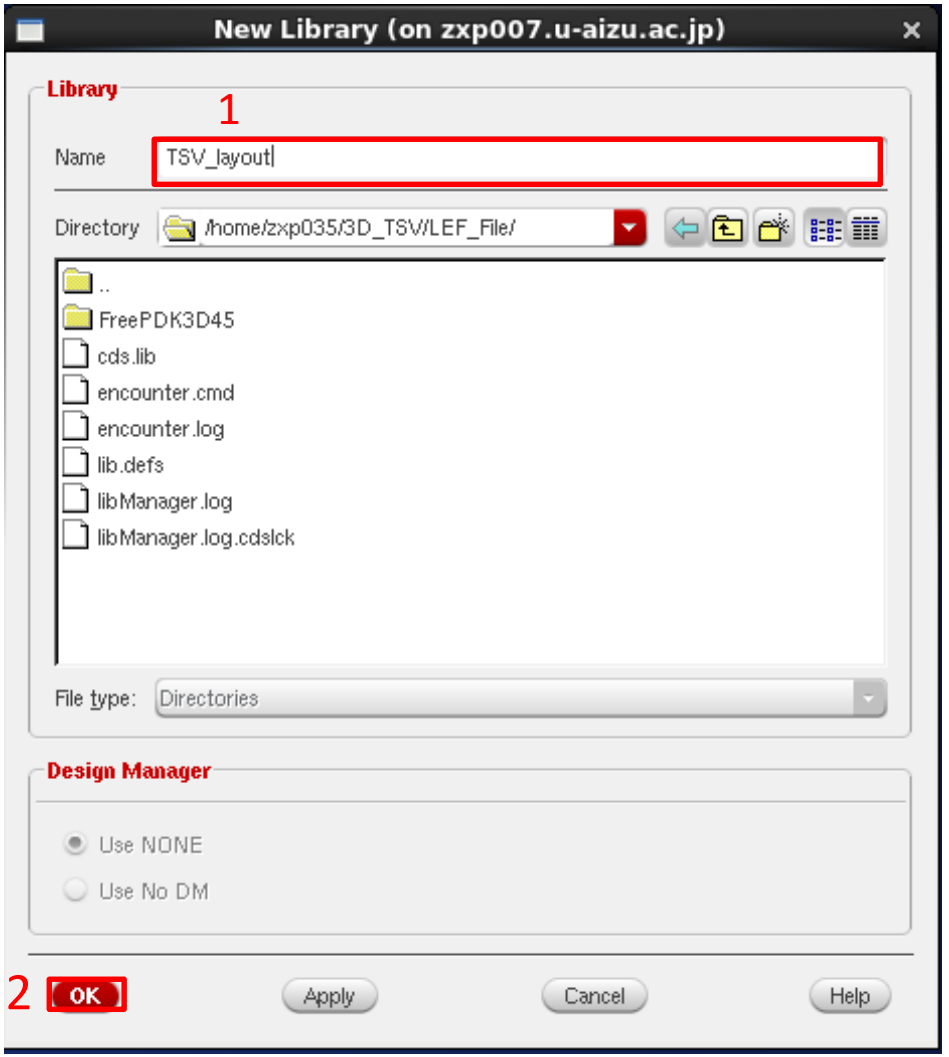
Step1: Create library



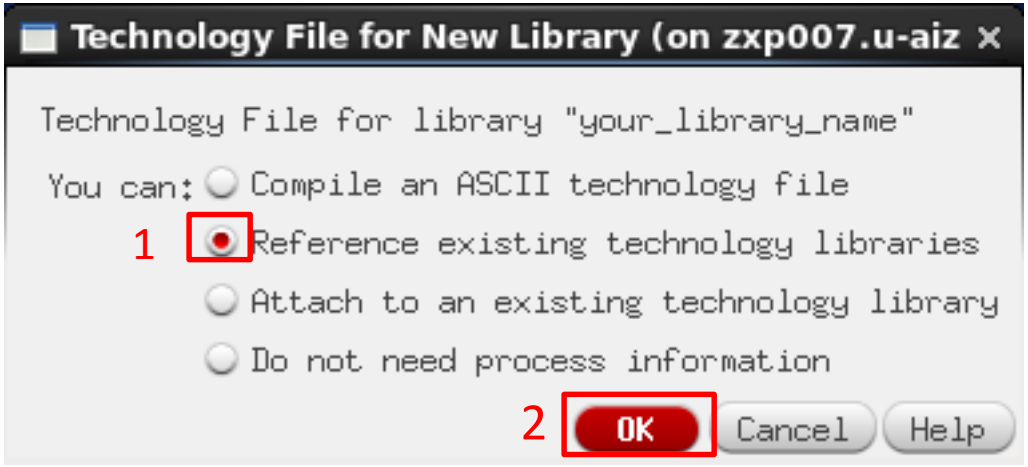
Click **File -> New -> Library**

Step1: Create library
a- Library name

1. In *Library Name*, type **TSV_layout**
2. Click **OK**

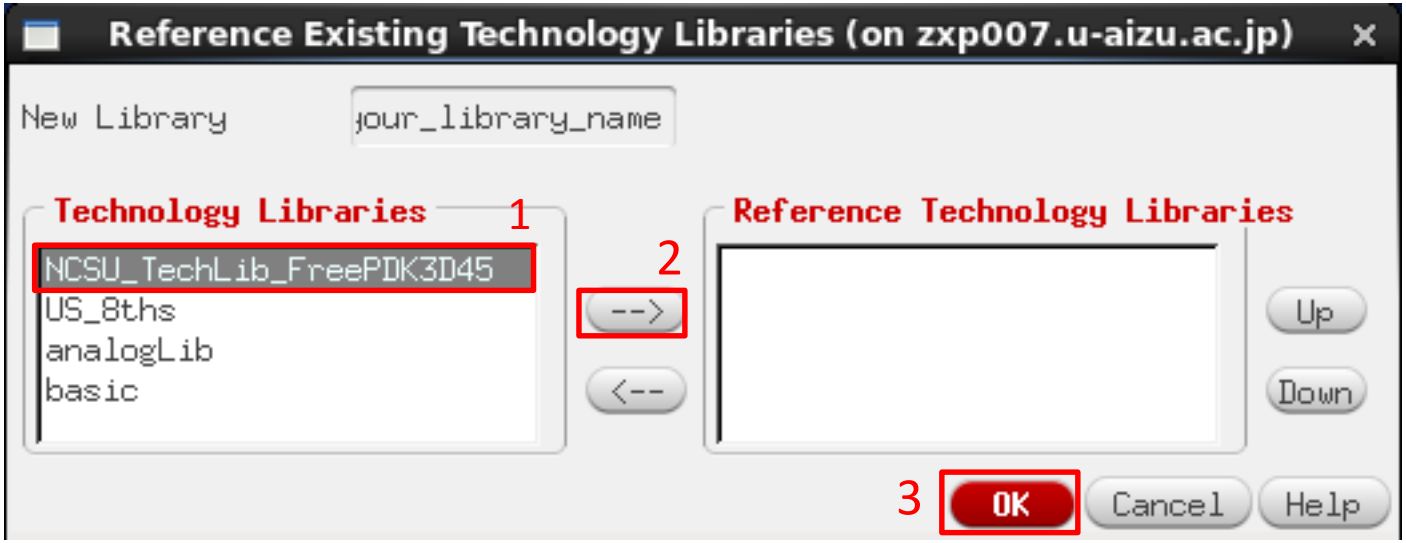


Step1: Create library
b- Technology file



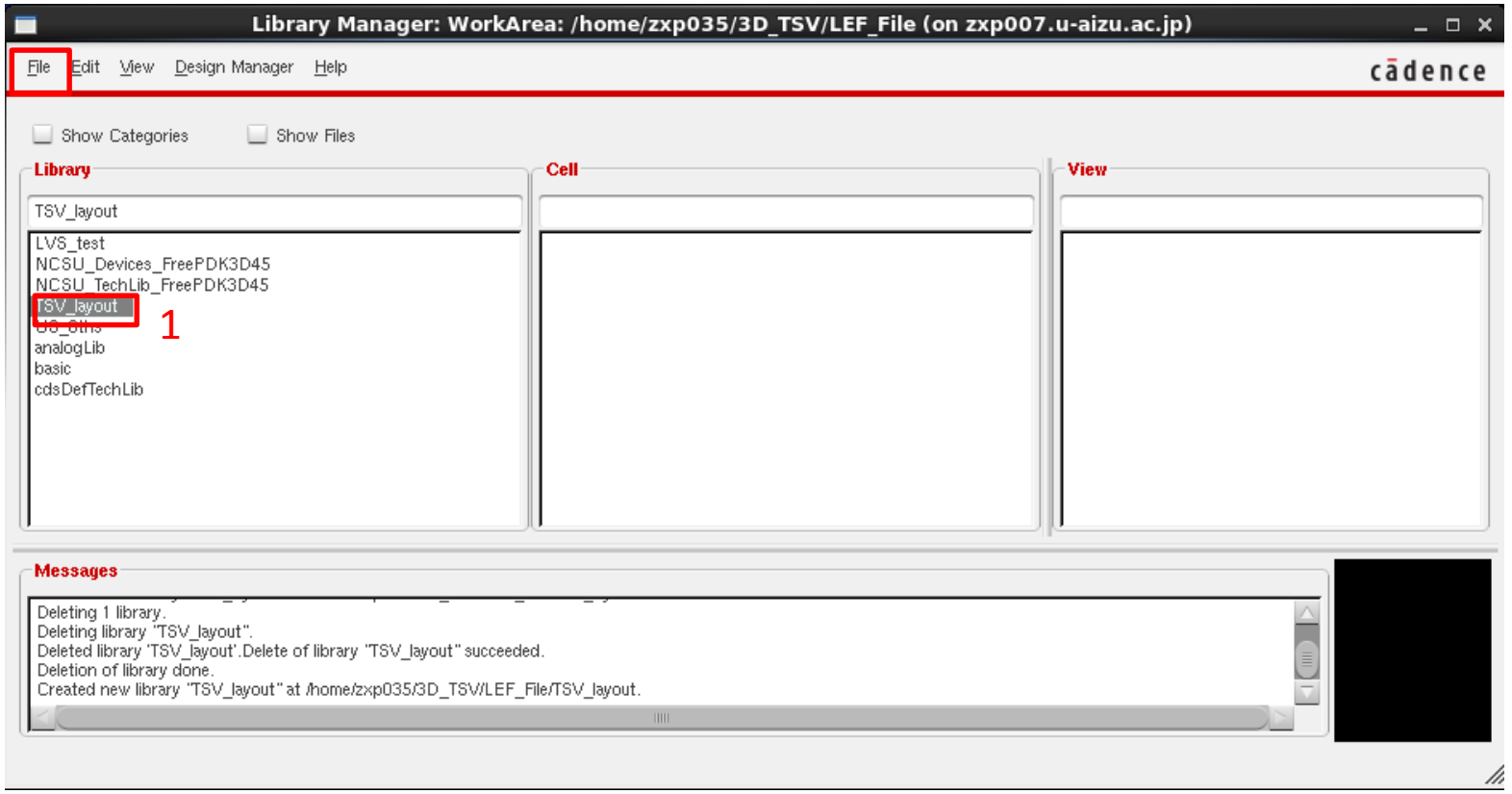
1. Check **Reference existing technology libraries**
2. Click **OK**

Step1: Create library
b- Technology file



1. Click **NCSU_TechLib_FreeFDK3D45**
2. Click **-->**
3. Click **OK**

Step2: Create cell view

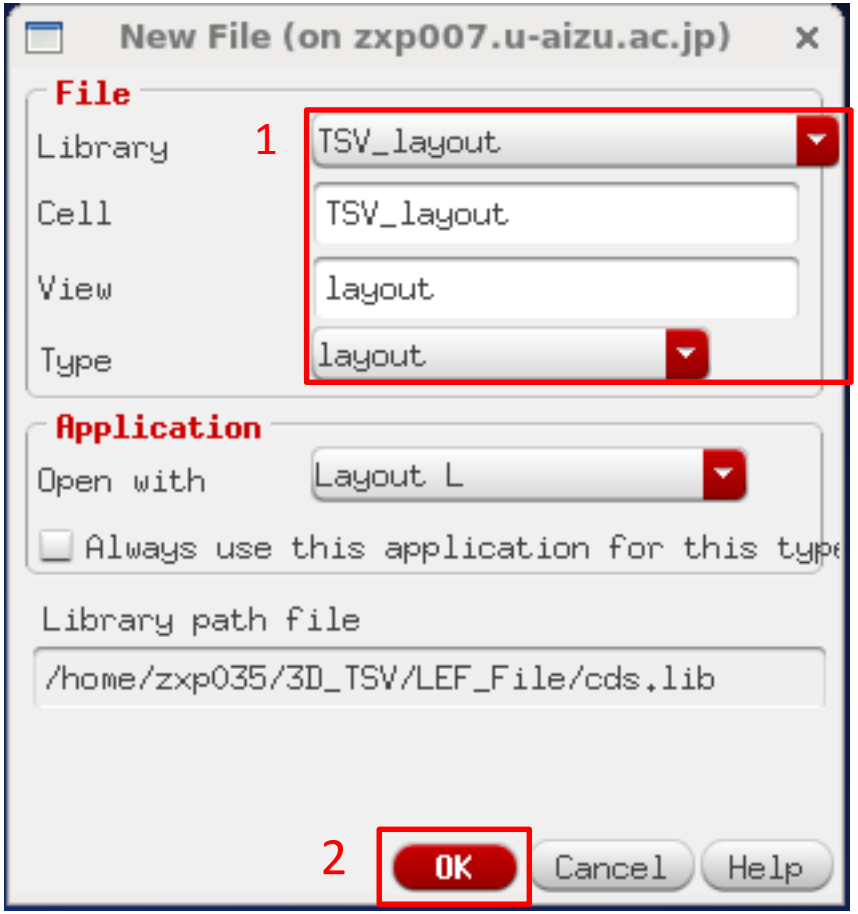


1. Click **TSV_layout**
2. Click **File -> New -> Cell View** to create cell view

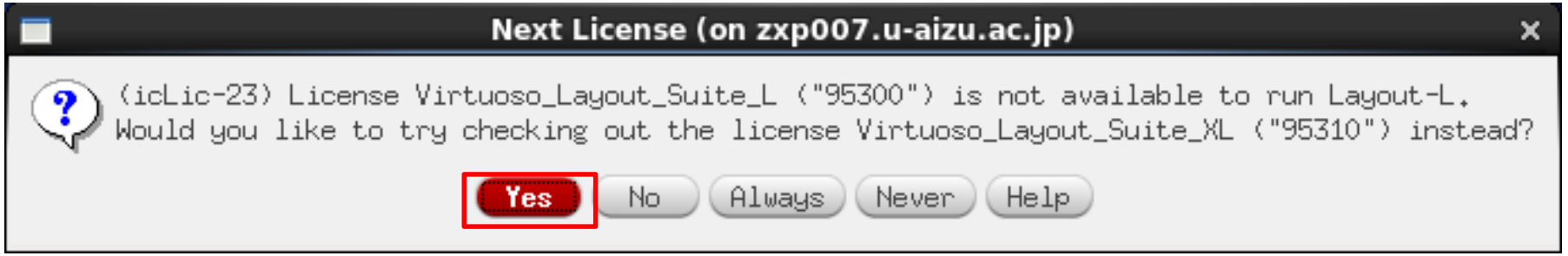
Step2: Create cell view

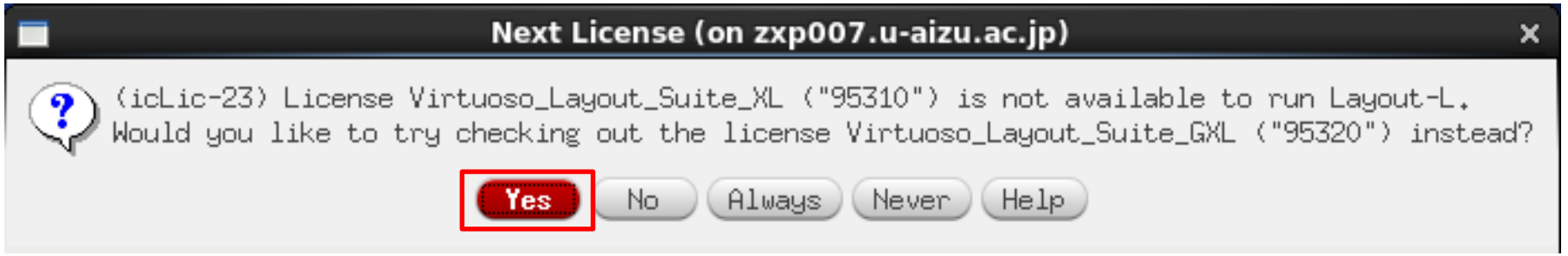
1. In File*:*
   - *Library: select* **TSV_layout**
   - *Cell:* type **TSV_layout**
   - *View:* type **layout**
   - *Type:* check **layout**

2. Click **OK**



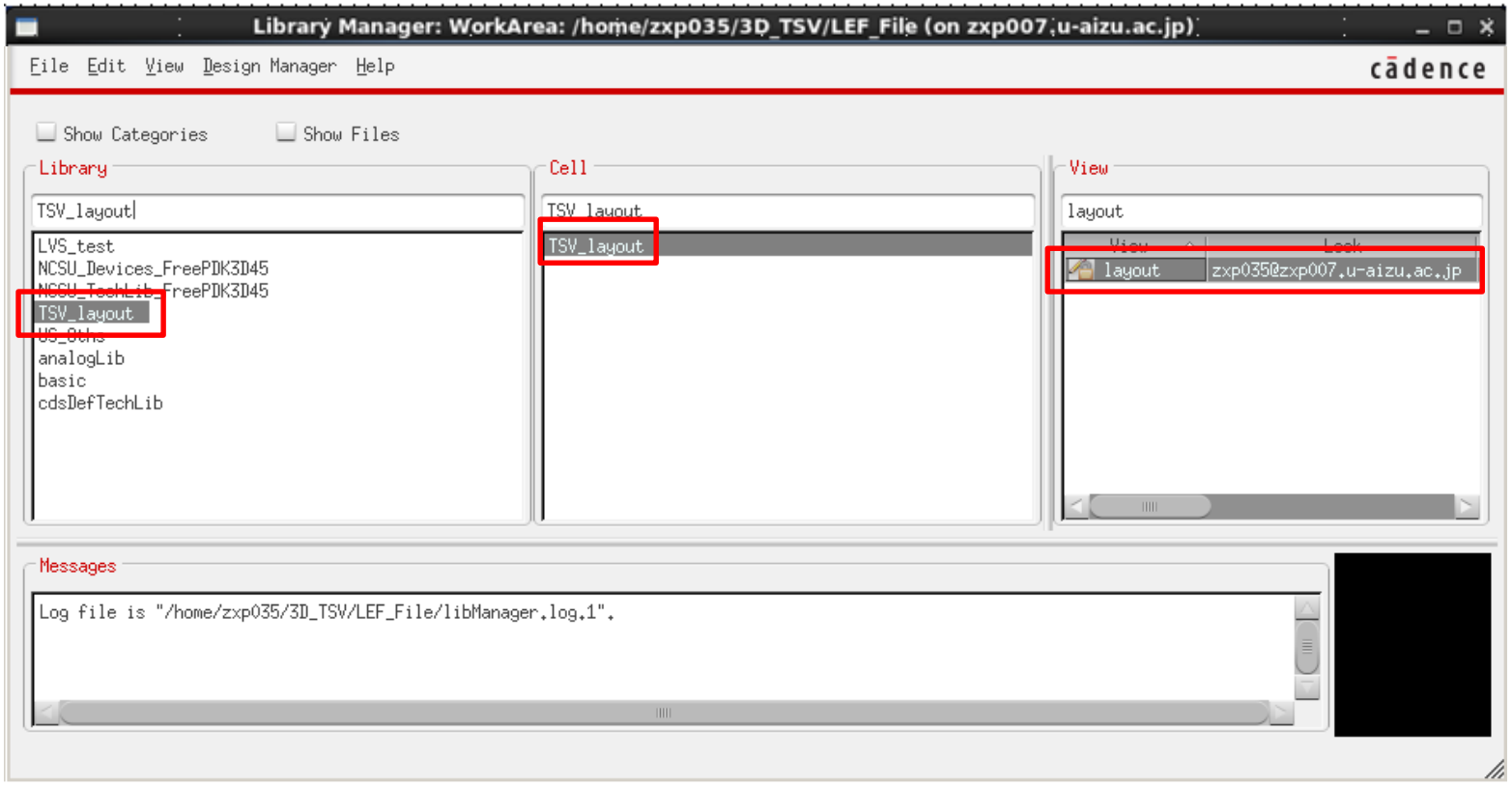
Step2: Create cell view





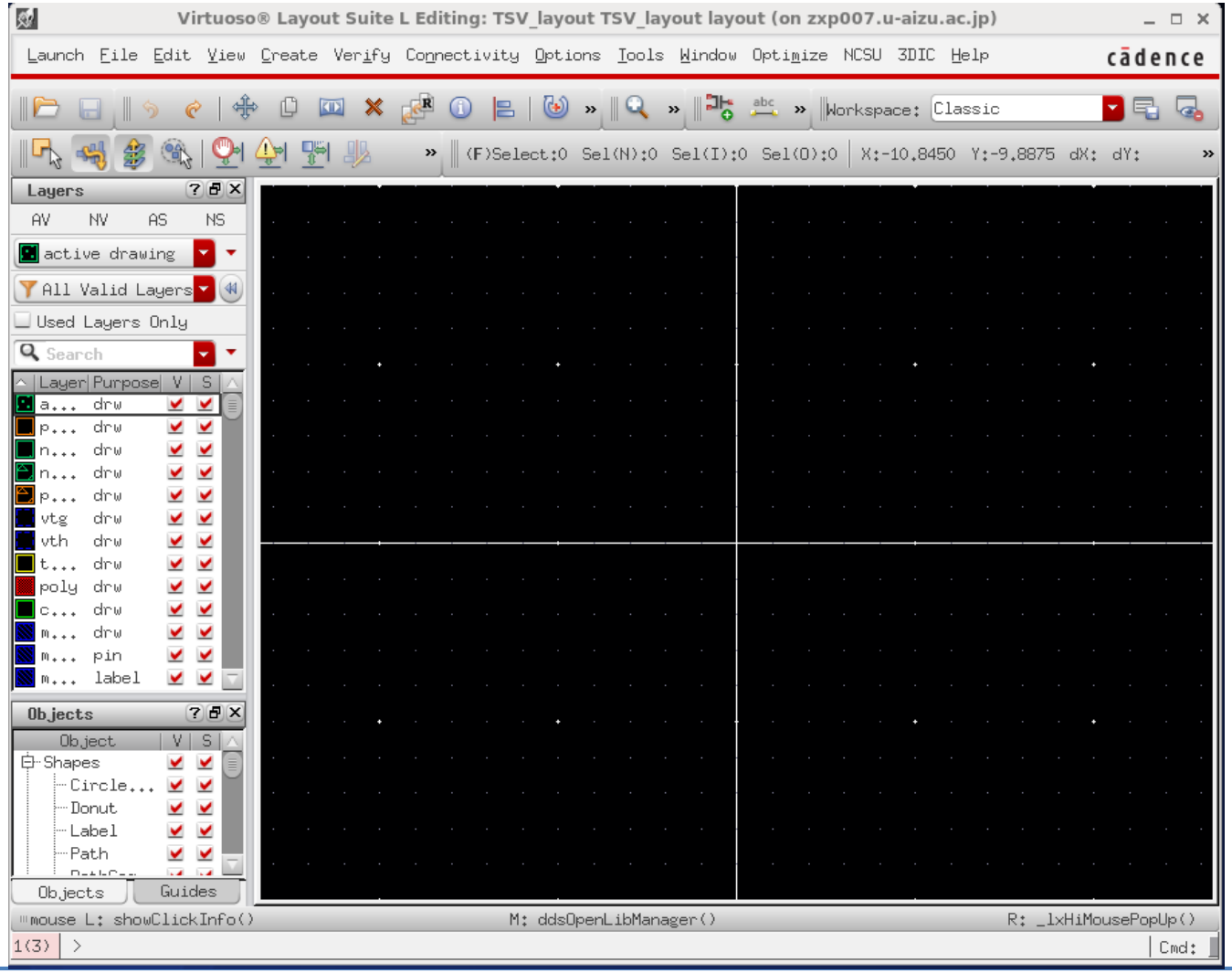
In both appeared pop-up messages, click **Yes**

Step2: Create cell view



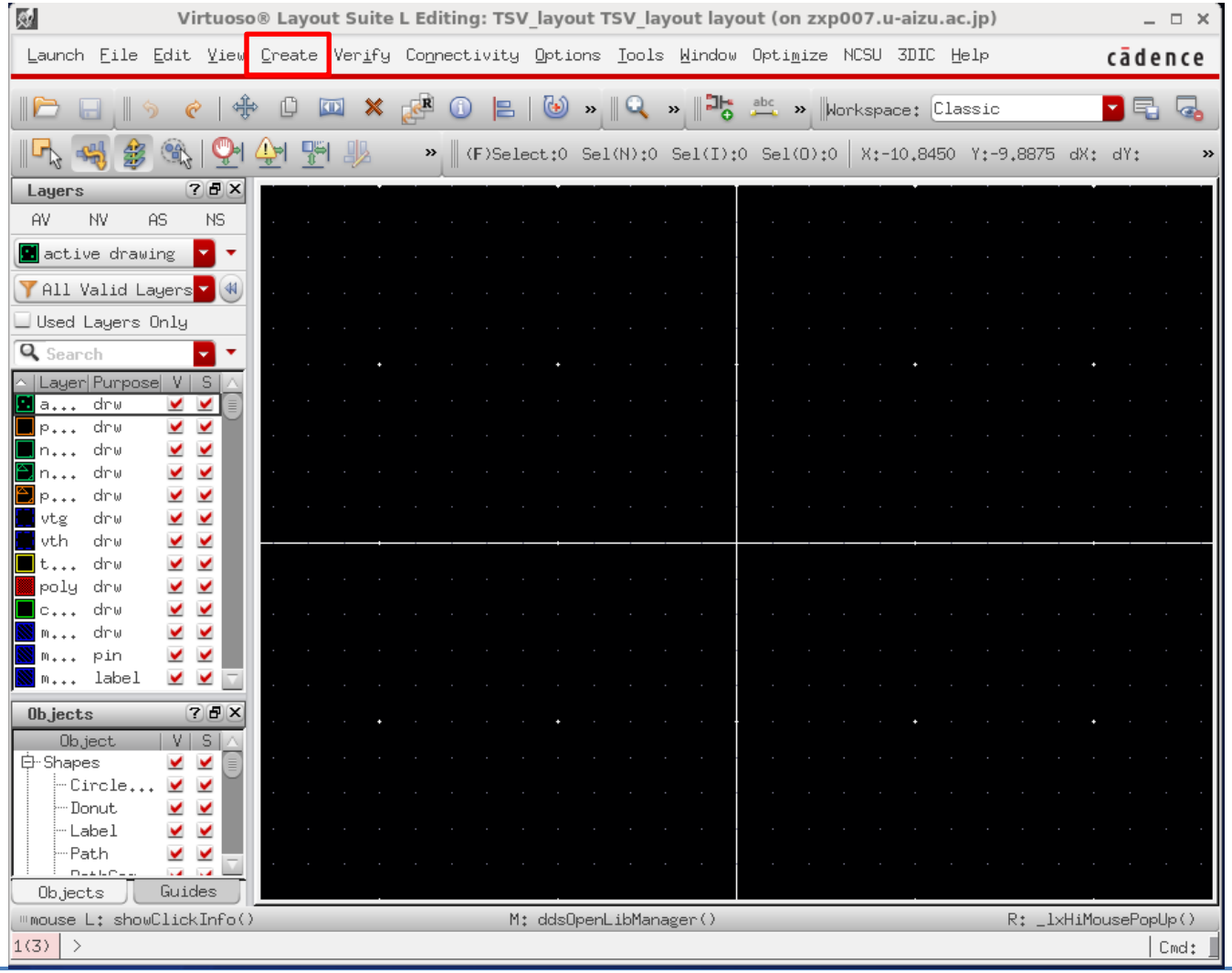
Click **your library -> Cell -> layout**

Step3: Create TSV layout



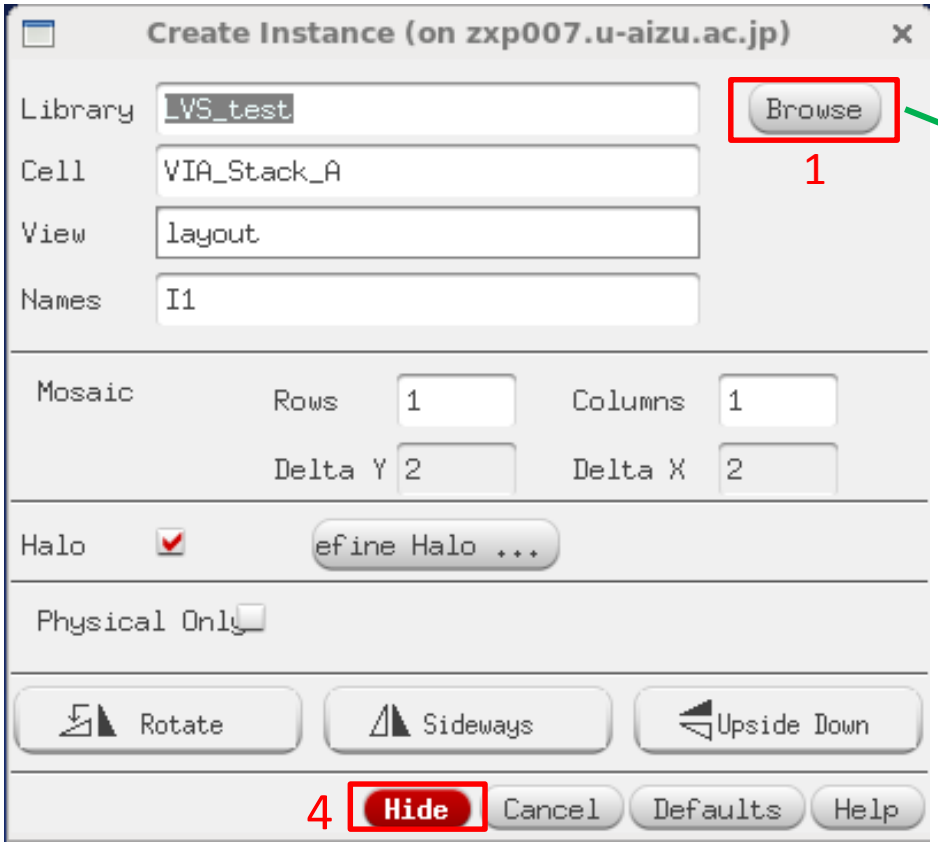
Welcome Virtuoso Layout screen

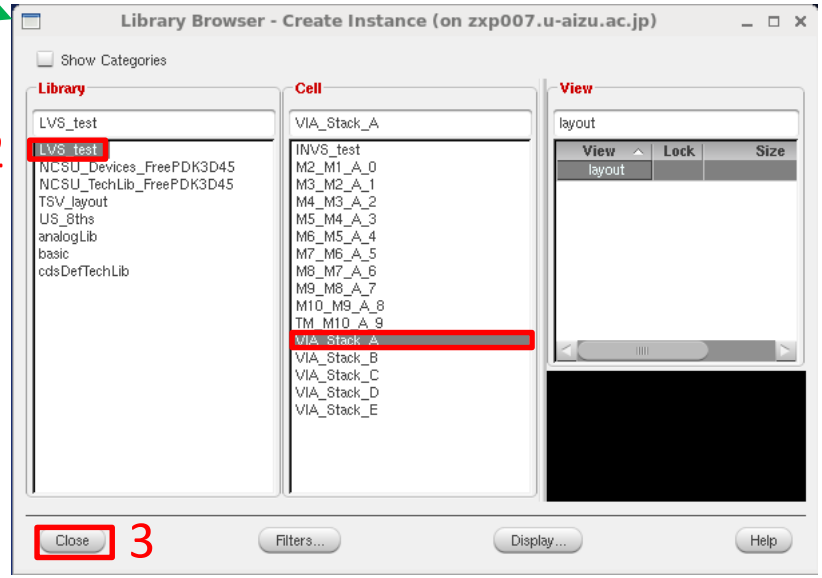
Step3: Create TSV layout
a- Create TSV instance



Click **Create -> Instance**

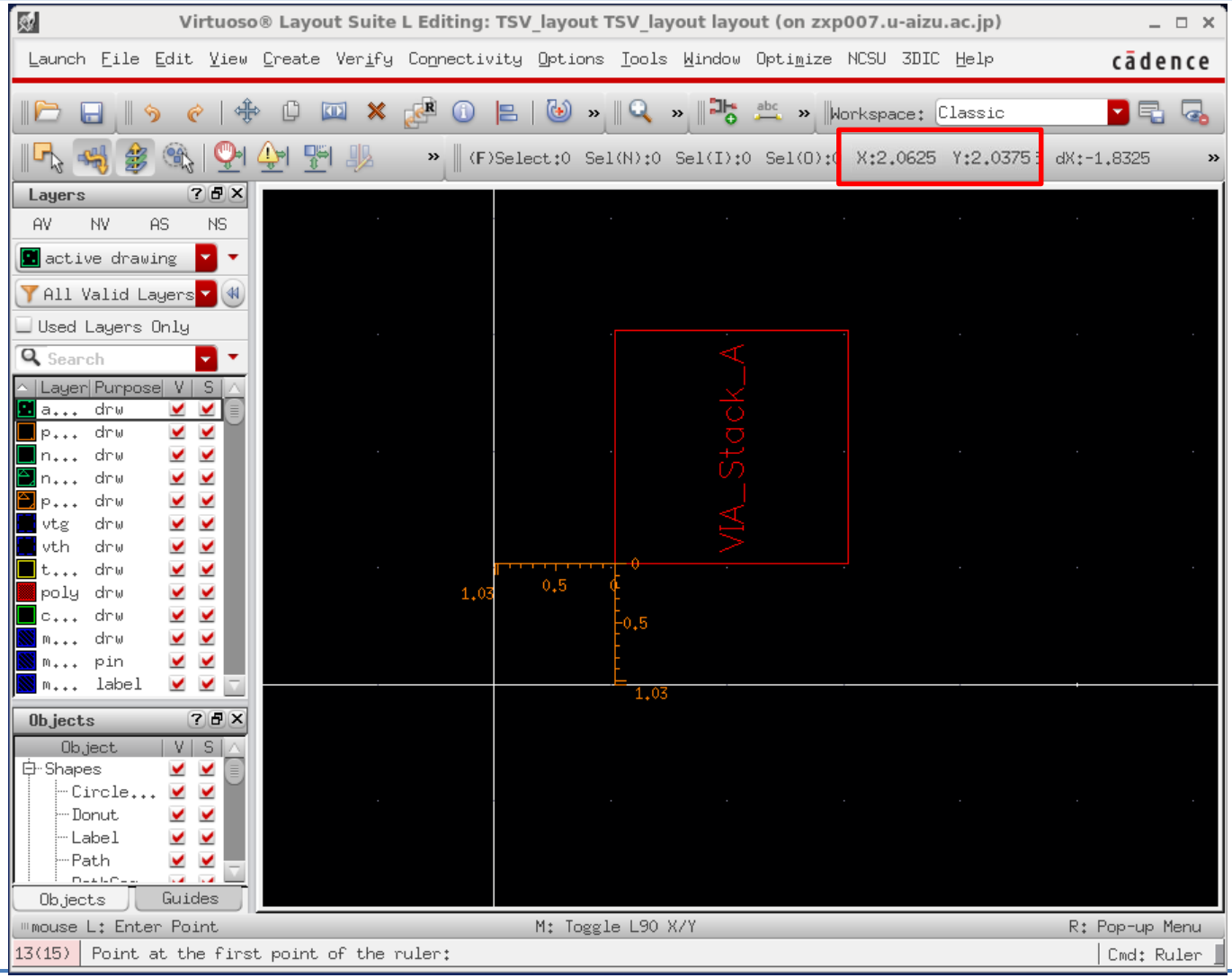
Step3: Create TSV layout
a- Create TSV instance





1. Click **Browse**
2. Click **LVS_test -> VIA_Stack_A**
3. Click **Close**
4. Click **Hide**

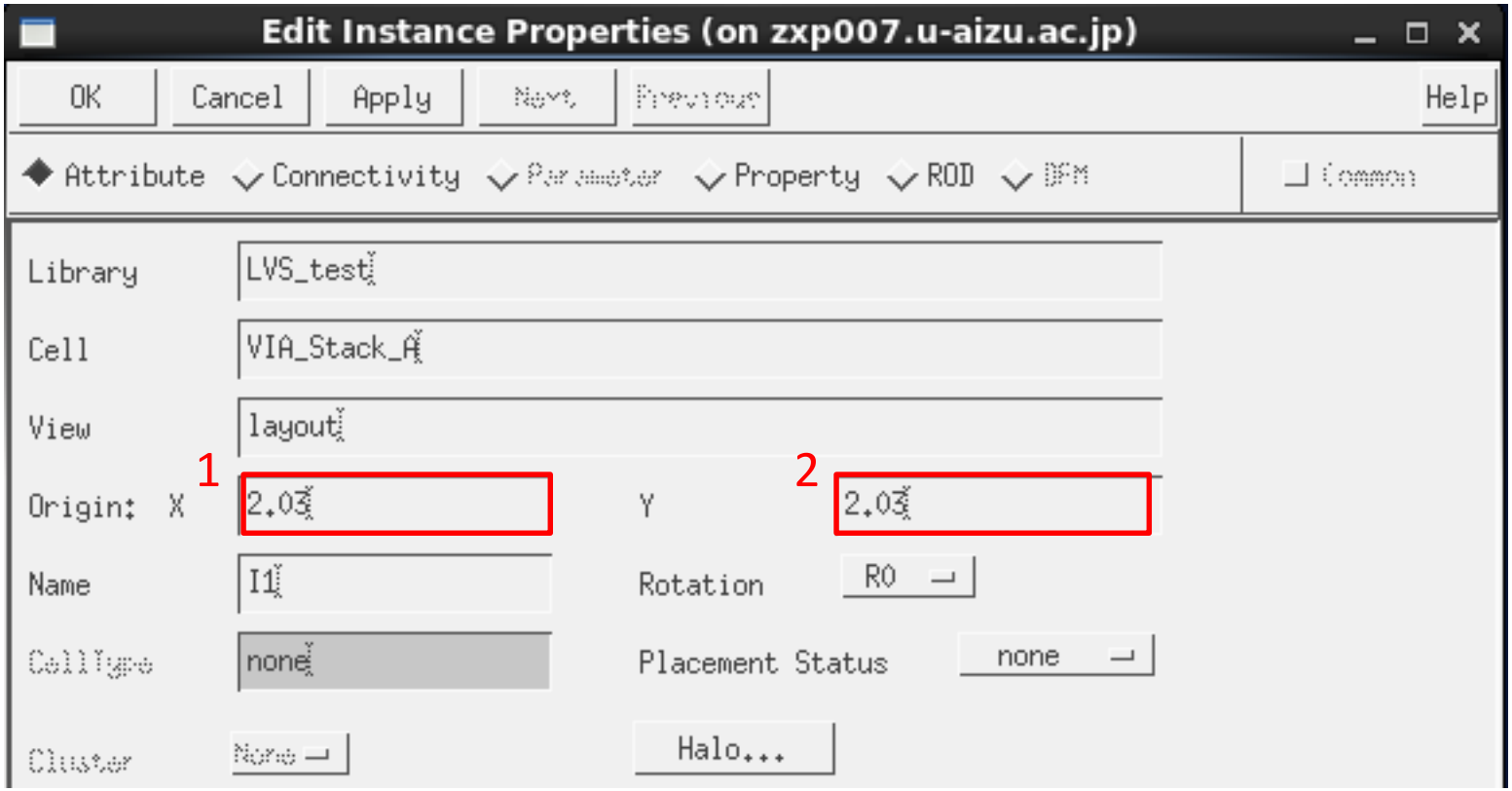
Step3: Create TSV layout
a- Create TSV instance



Click **point(X: 2.0625 Y: 2.0375).**

You may use the ruler in the toolbar for precision

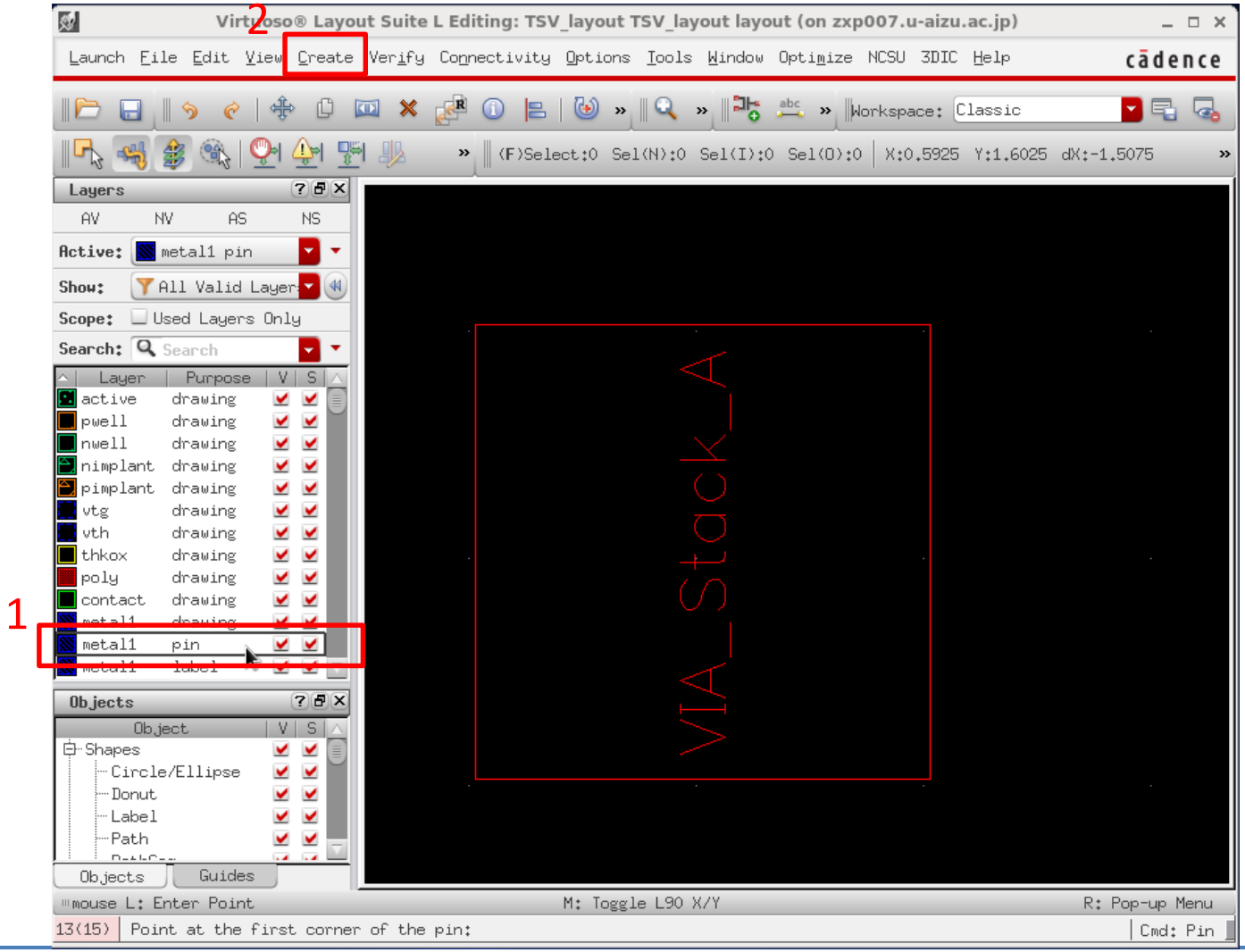
Step3: Create TSV layout
a- Create TSV instance



1. In *Origin: X* type **2.03**
2. In *Origin: Y* type **2.03**

Right click on the created **VIA_Stack_A,** and click **Properties**

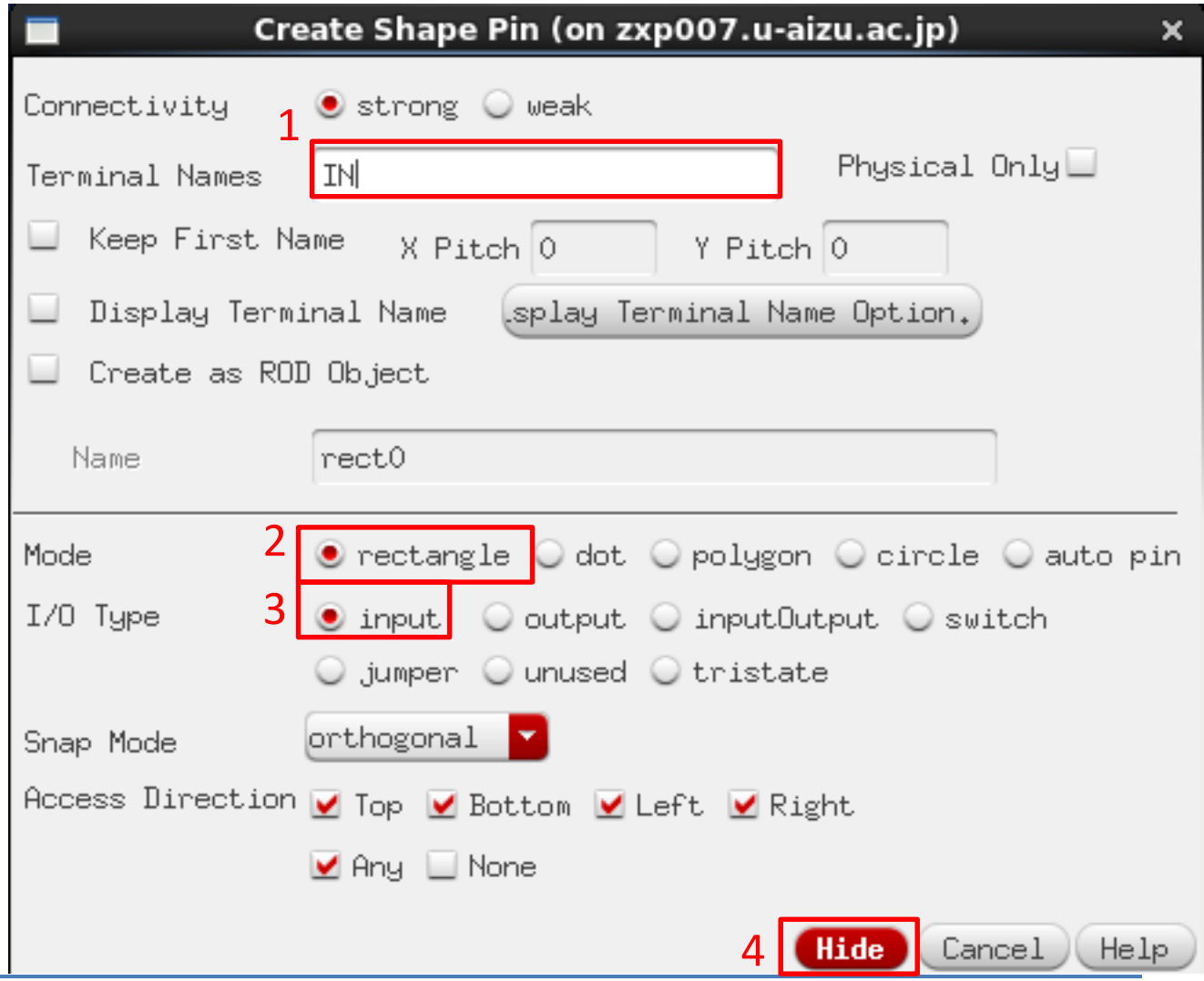
Step3: Create TSV layout
b- Create input pin



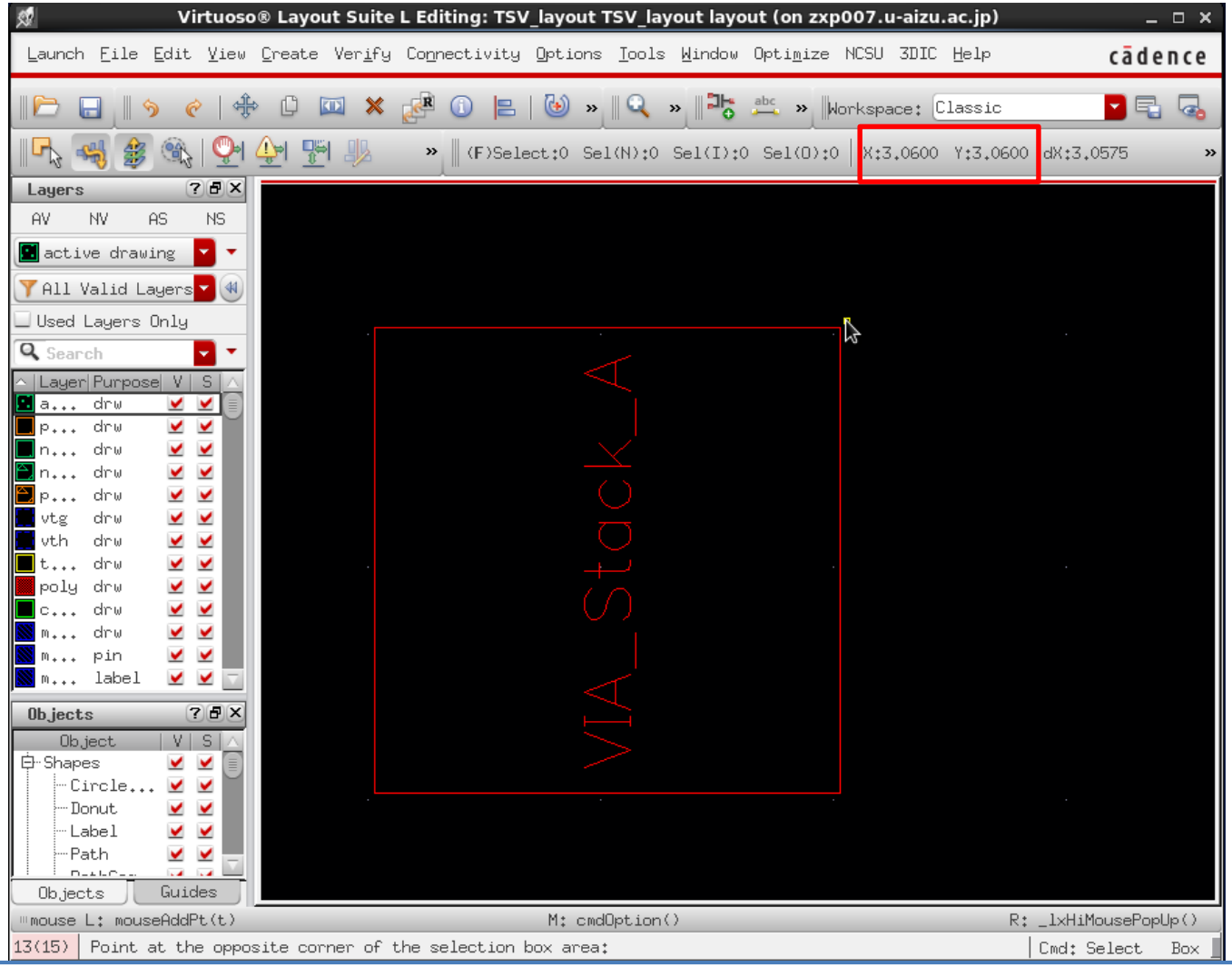
1. Click **metal1 pin**
2. Click **Create -> Pin**

Step3: Create TSV layout
b- Create input pin

1. Terminal Names → **IN**
2. Mode → **rectangle**
3. I/O Type → **input**
4. Click **Hide**

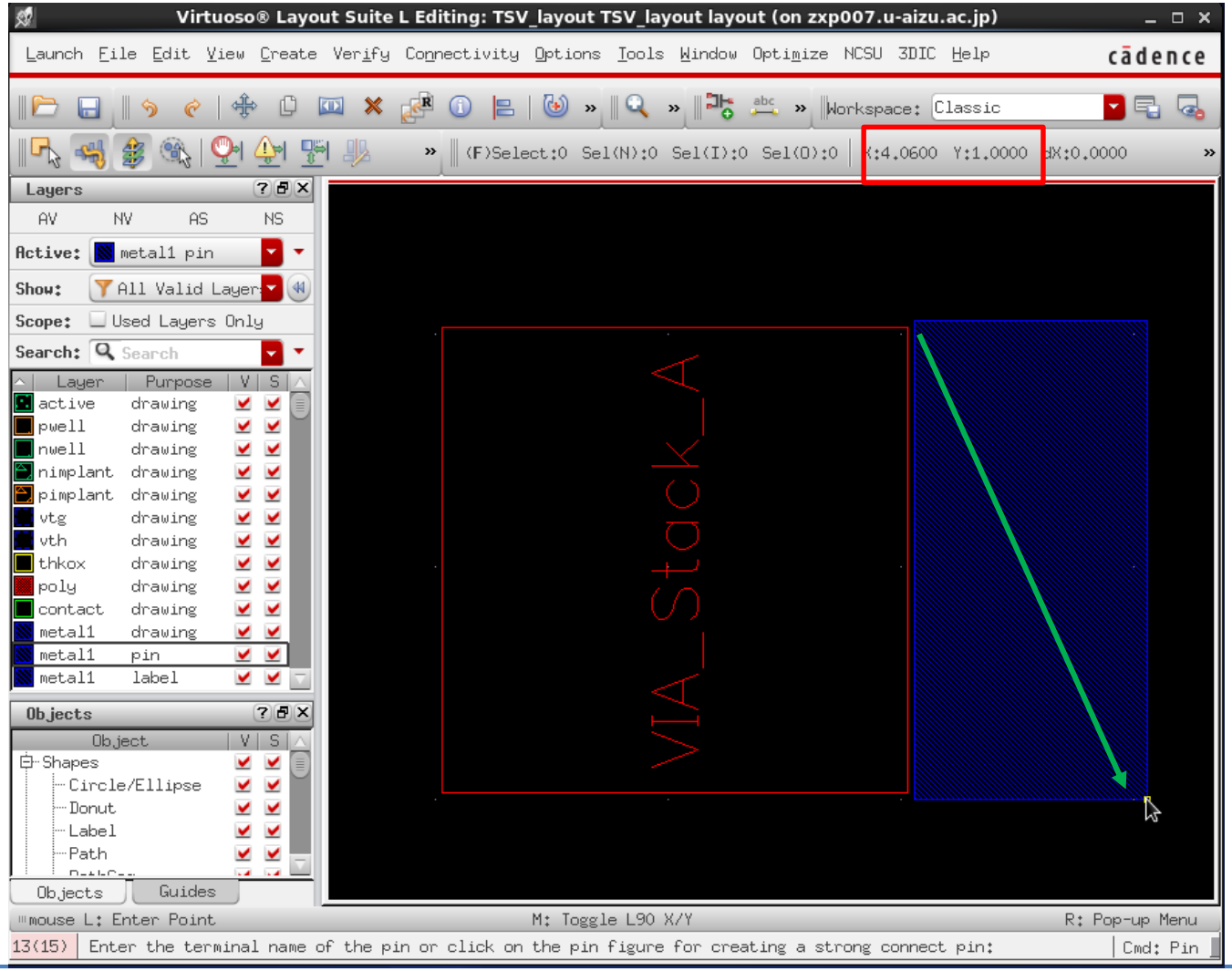


Step3: Create TSV layout
b- Create input pin



Click **point(X:3.06 Y: 3.06)**

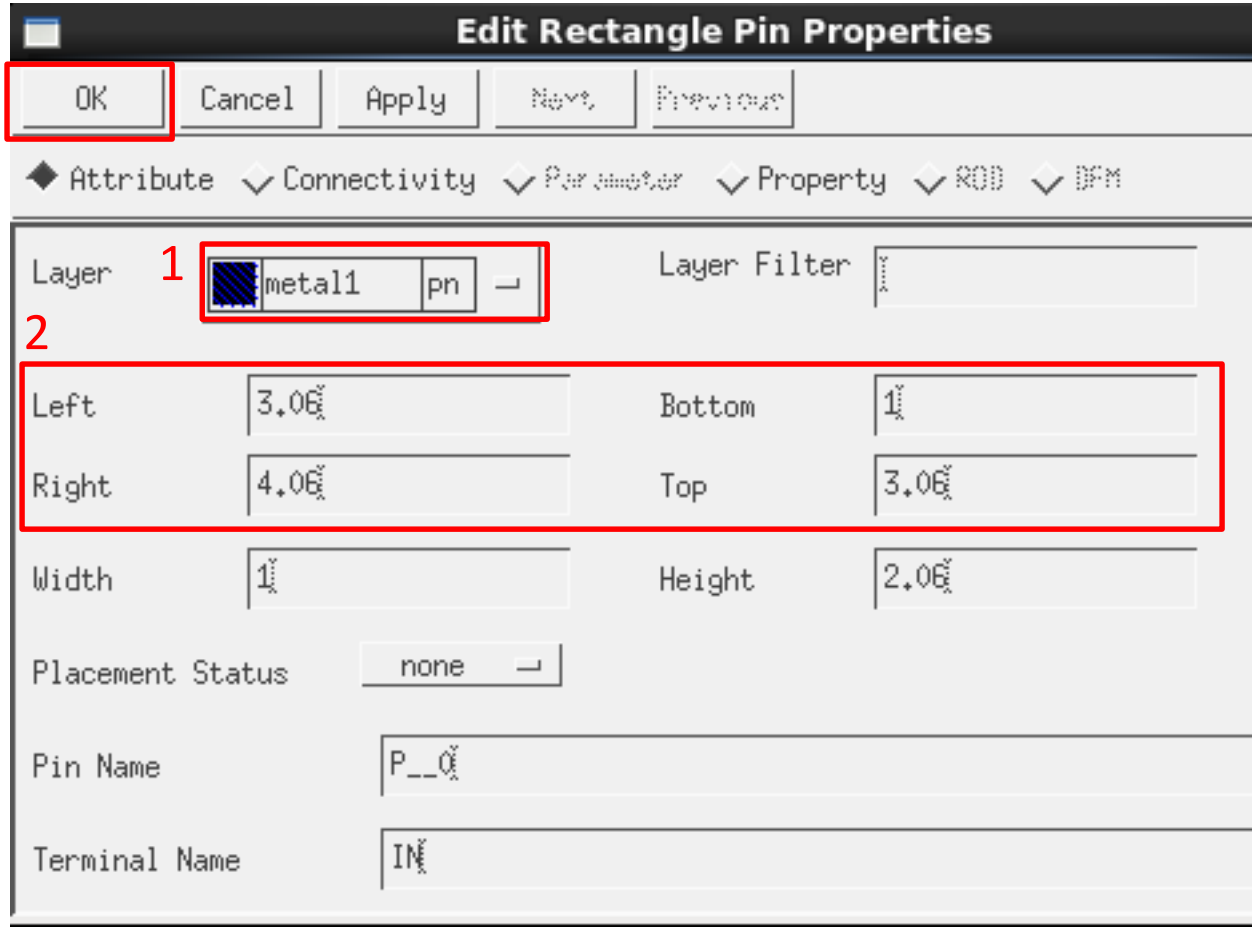
Step3: Create TSV layout
b- Create input pin



Drag the pointer to **point(X:4.06 Y: 1.00)** and click

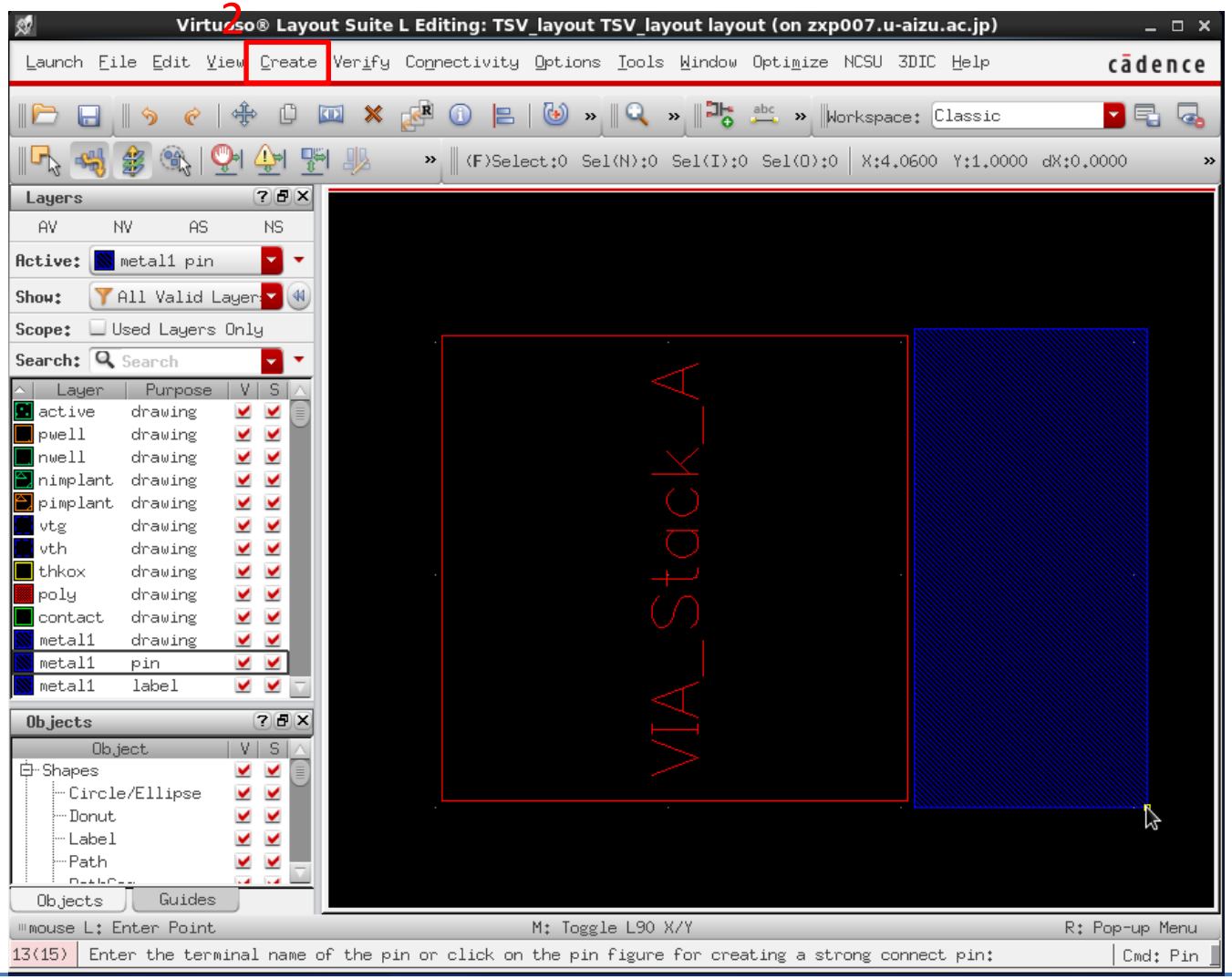
Step3: Create TSV layout
b- Create input pin

1. Select **metal1 pn**
2. Type
   - **Left: 3.06**
   - **Right: 4.06**
   - **Bottom: 1**
   - **Top: 3.06**
3. Click **OK**



Right click on the created **Input pin** and click **Properties**

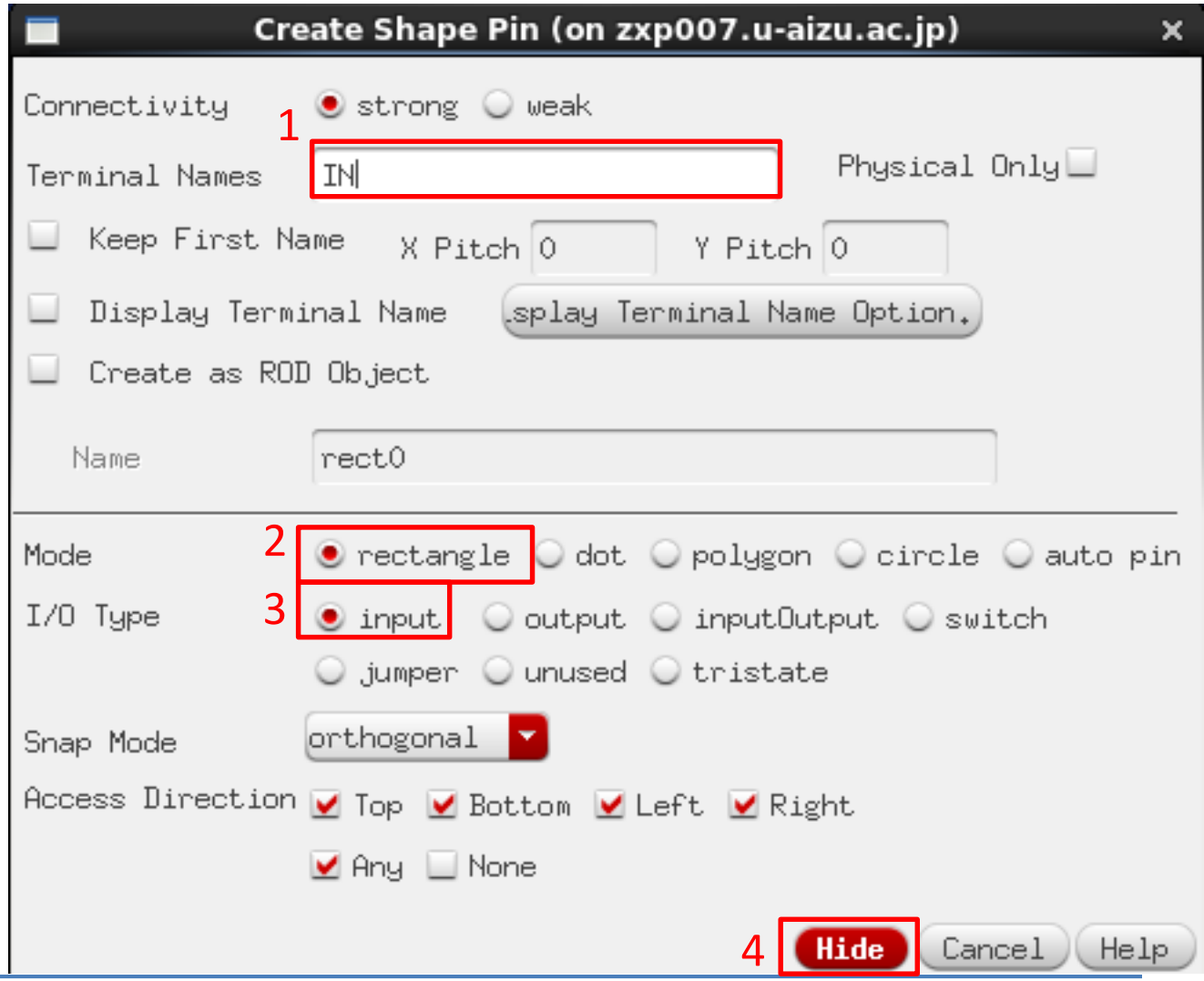
Step3: Create TSV layout
b- Create input pin



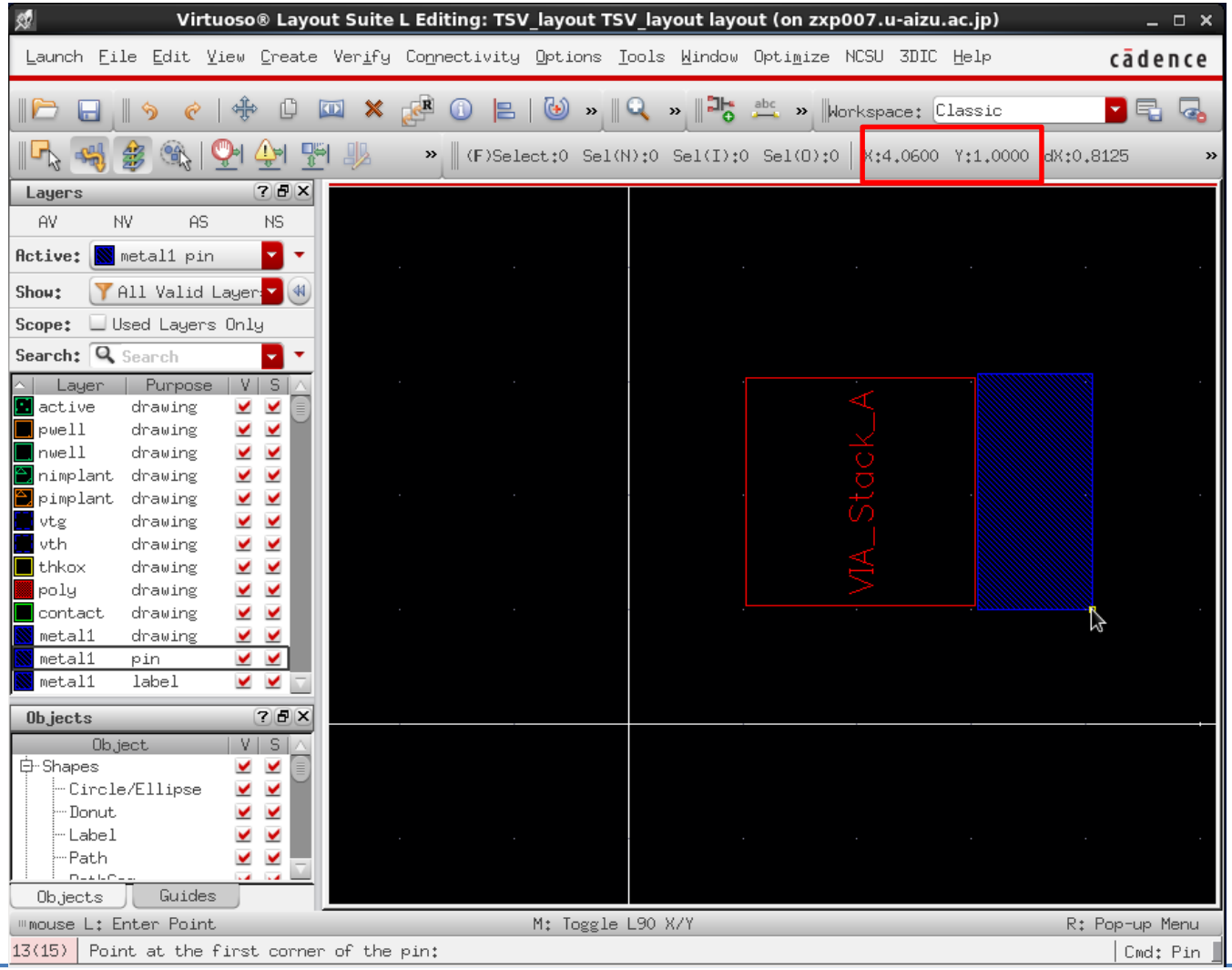
Click **Create -> Pin**

Step3: Create TSV layout
b- Create input pin

1. Type **IN**
2. Check **rectangle**
3. Check **input**
4. Click **Hide**

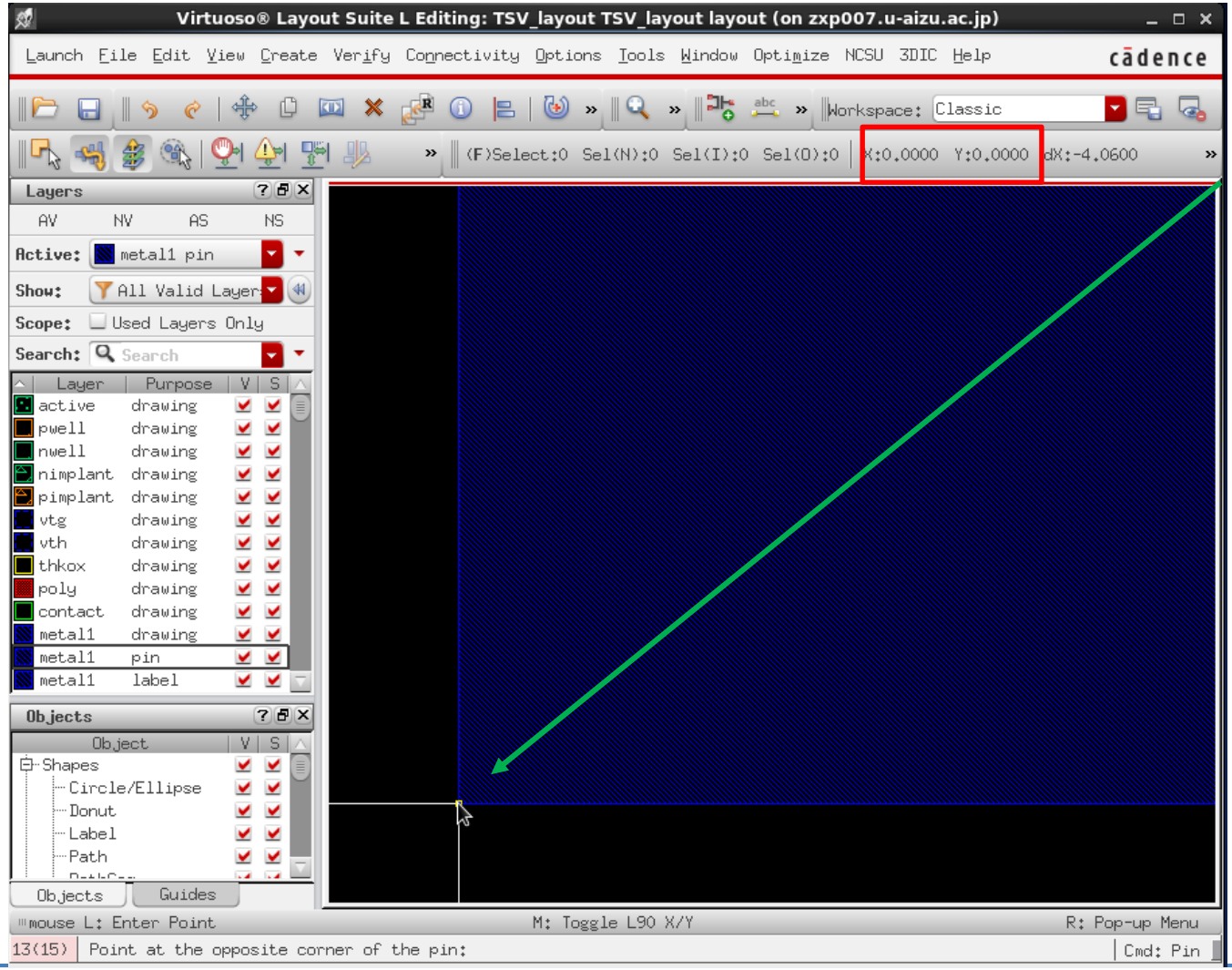


Step3: Create TSV layout
b- Create input pin



Click **point(X:4.06 Y: 1.00)**

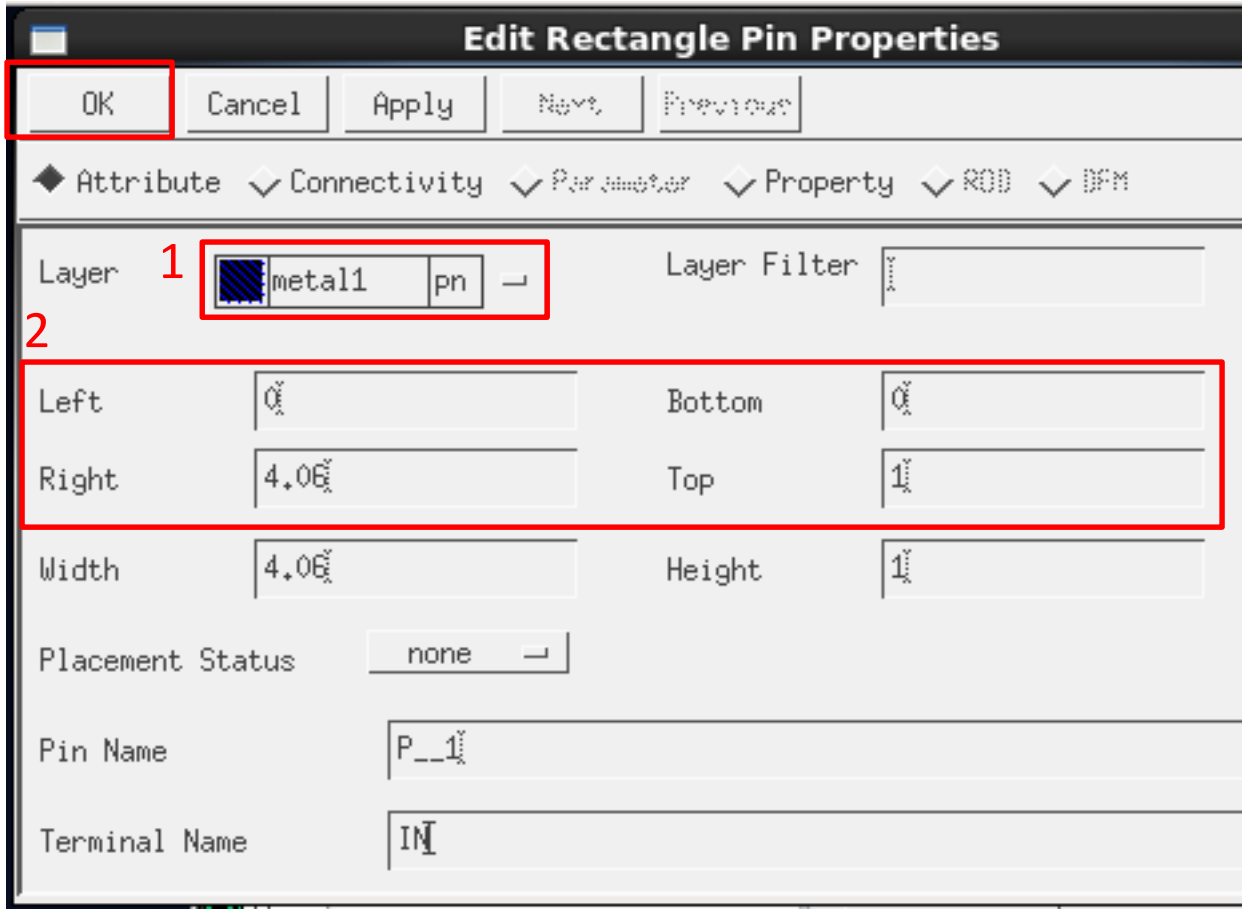
Step3: Create TSV layout
b- Create input pin



Drag the pointer to **point(X:0.00 Y: 0.00)** and click

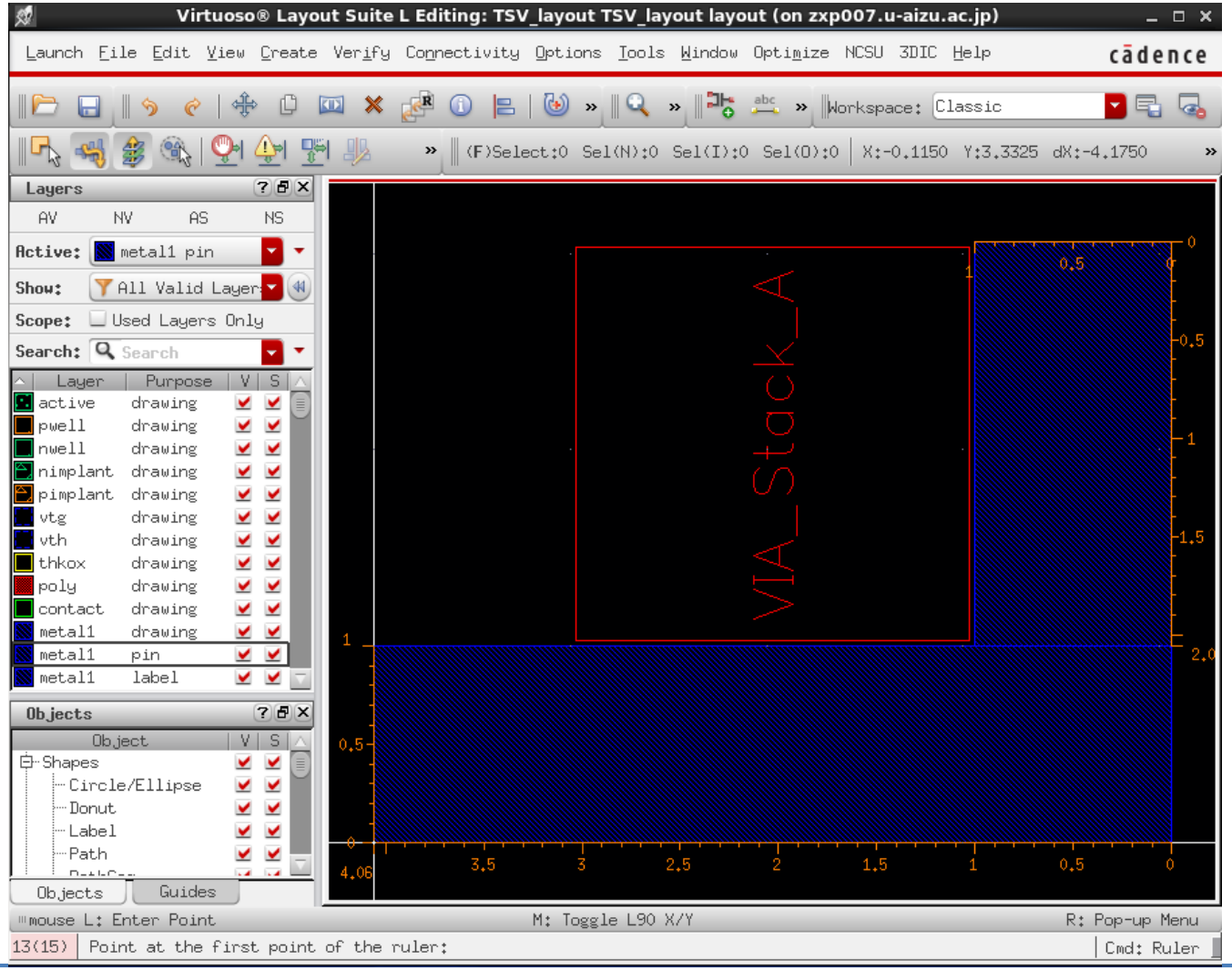
Step3: Create TSV layout
b- Create input pin

1. Select **metal1 pn**
2. Type
   - **Left: 0**
   - **Right: 4.06**
   - **Bottom: 0**
   - **Top: 1**
3. Click **OK**

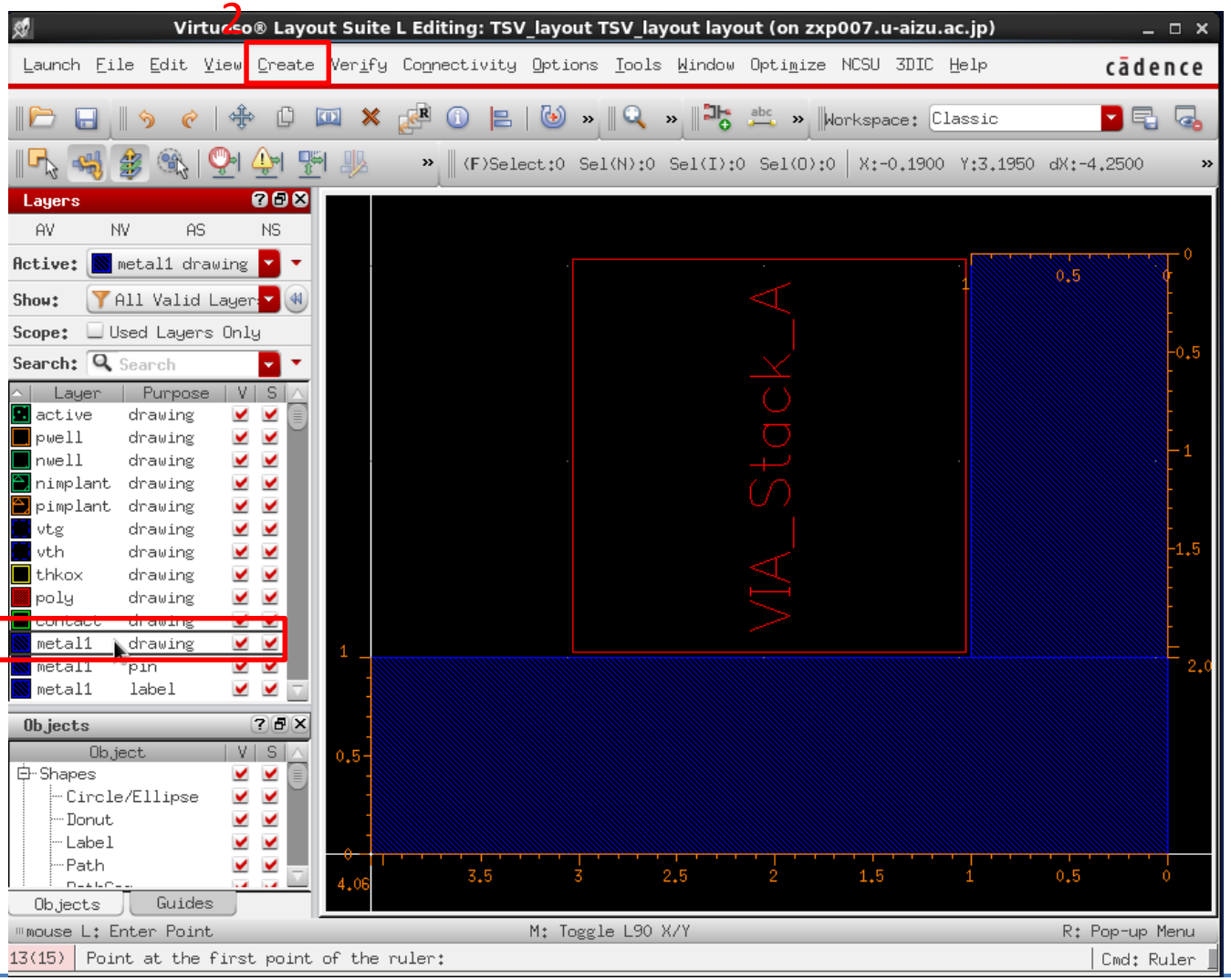


Right click created **Input pin** and click **Properties**

Step3: Create TSV layout
b- Create input pin

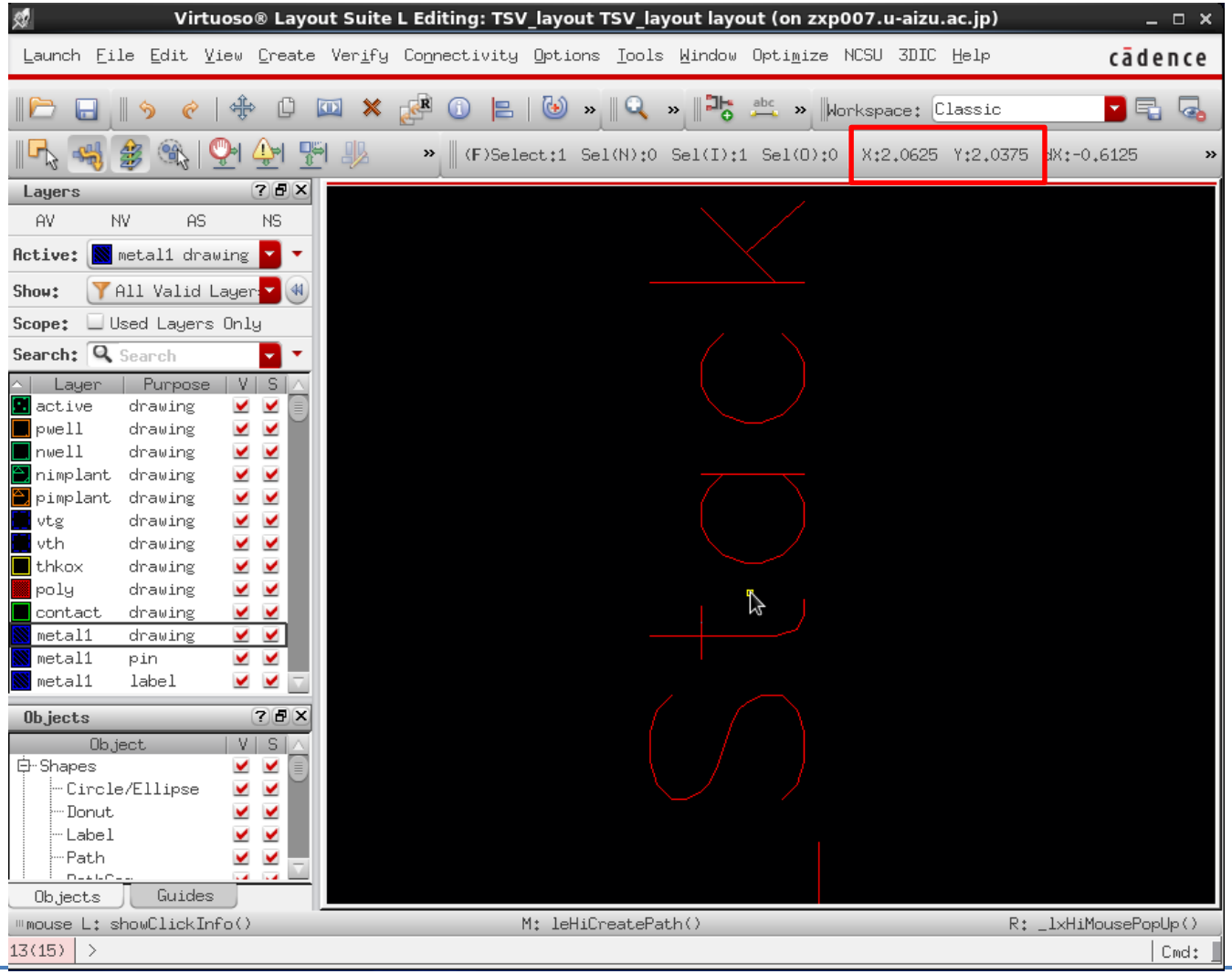


Step3: Create TSV layout
c- Create path



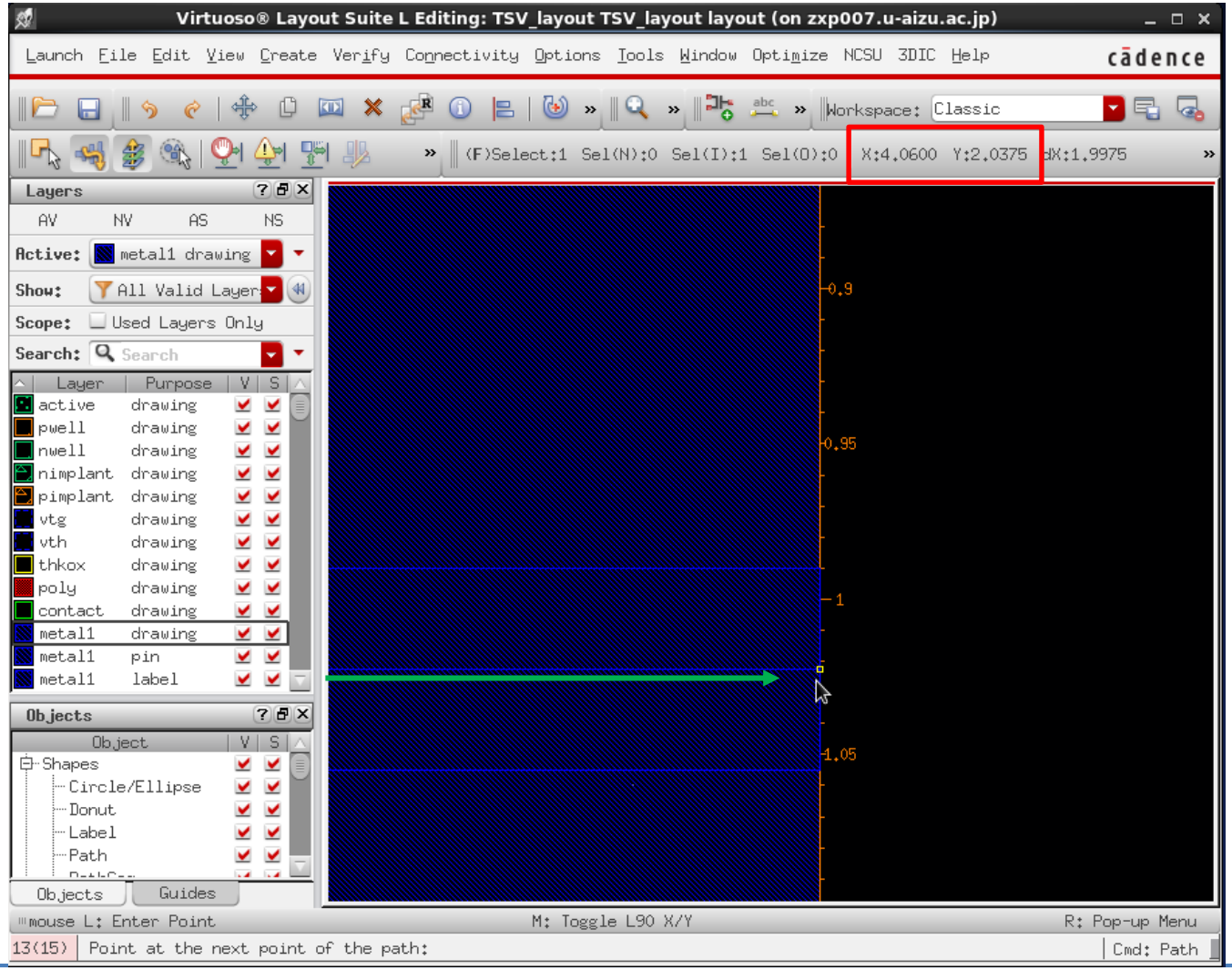
1. Click **metal1 drawing**
2. Click **Create -> Shape -> path**

Step3: Create TSV layout
c- Create path



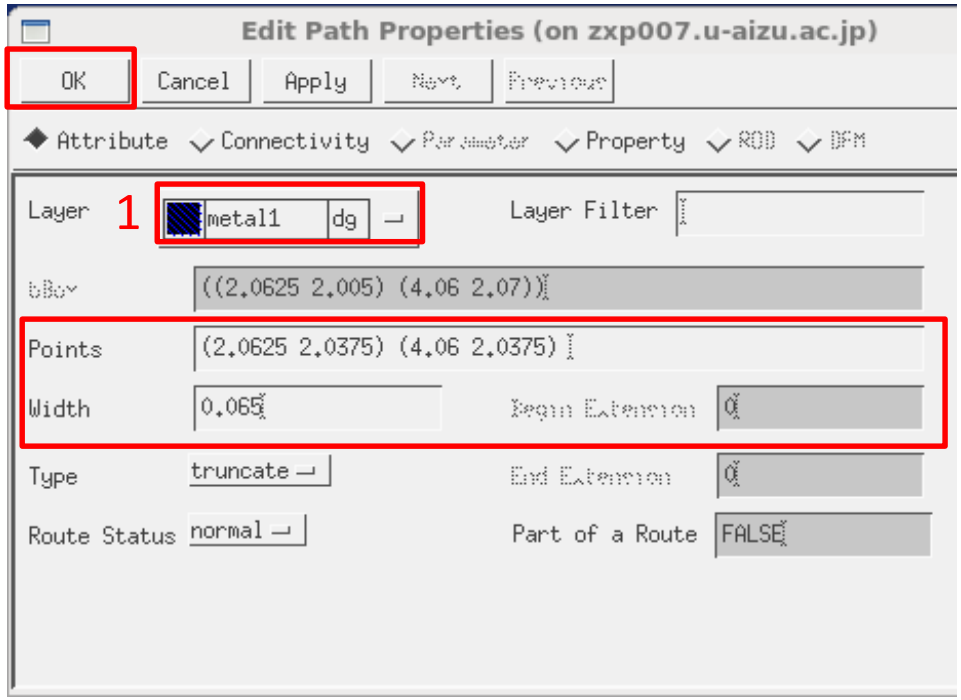
Click **point(X: 2.0625 Y: 2.0375)**

Step3: Create TSV layout
c- Create path



Drag the pointer to **point(X:4.06 Y: 2.0375)** and double click

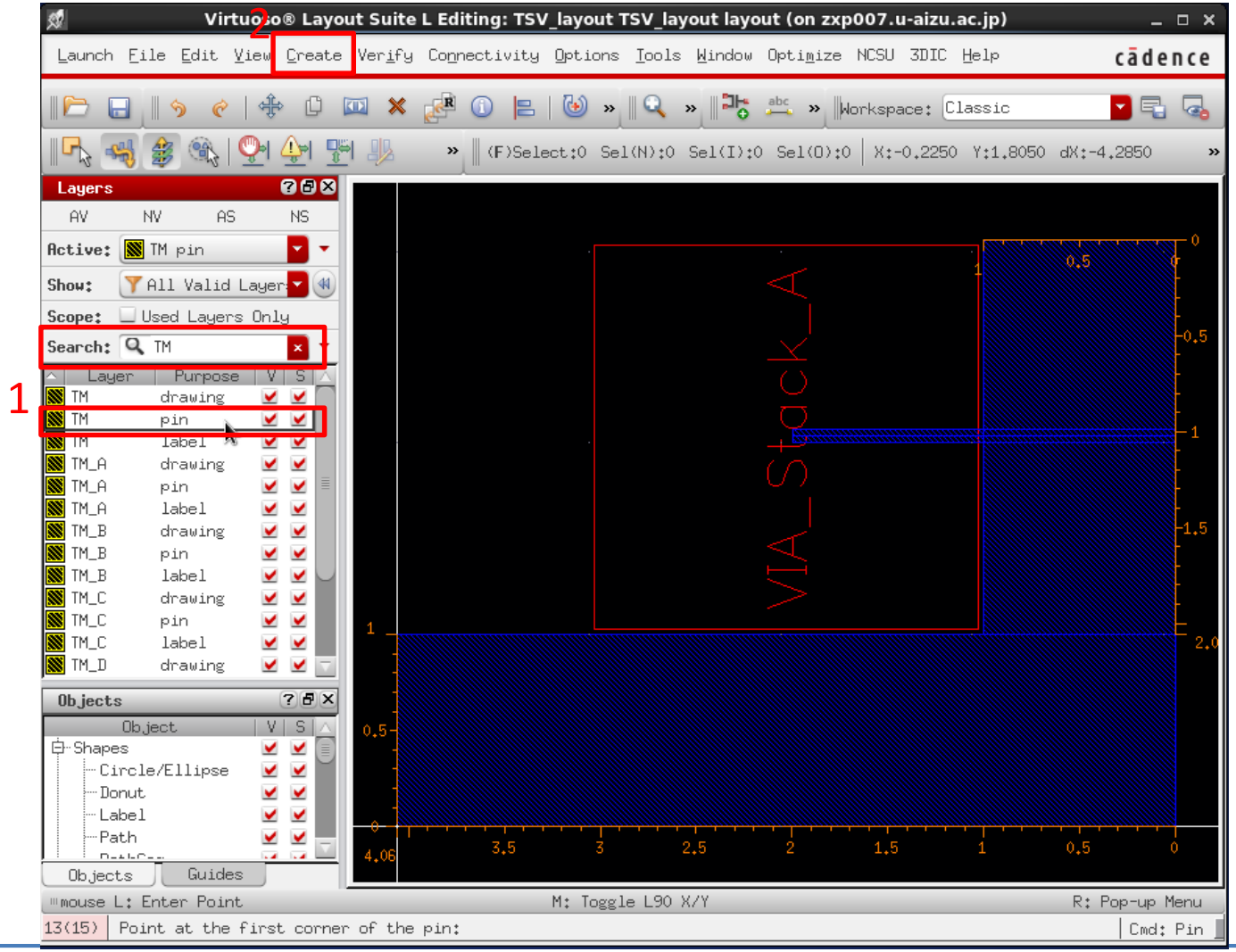
Step3: Create TSV layout
c- Create path



1. Select **metal1 dg**
2. Type
   Points: **(2.0625 2.0375) (4.06 2.0375)**
   Width: **0.065**
3. Click **OK**

Right click on the created **path** and click **Properties**

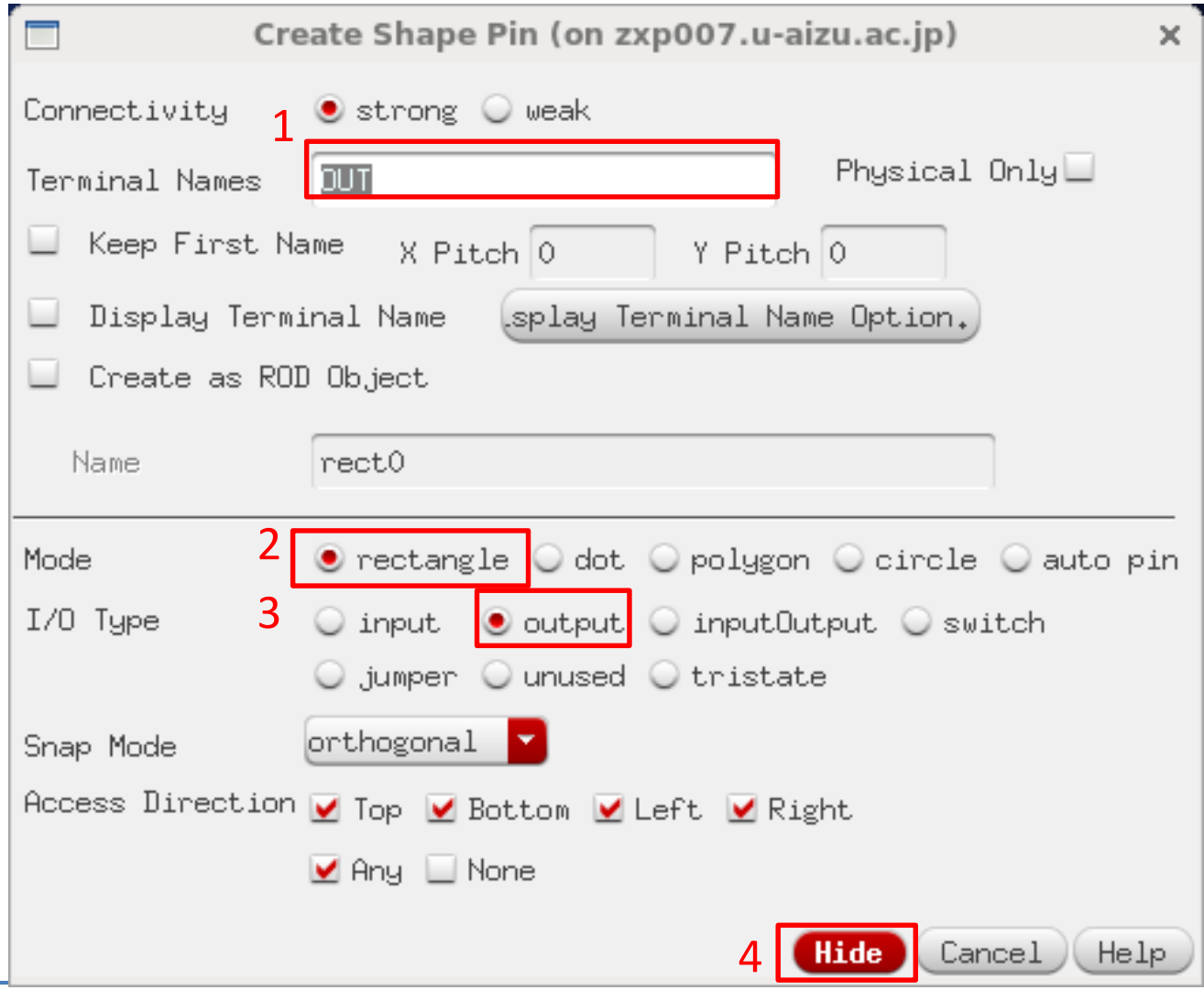
Step3: Create TSV layout
d- Create output pin



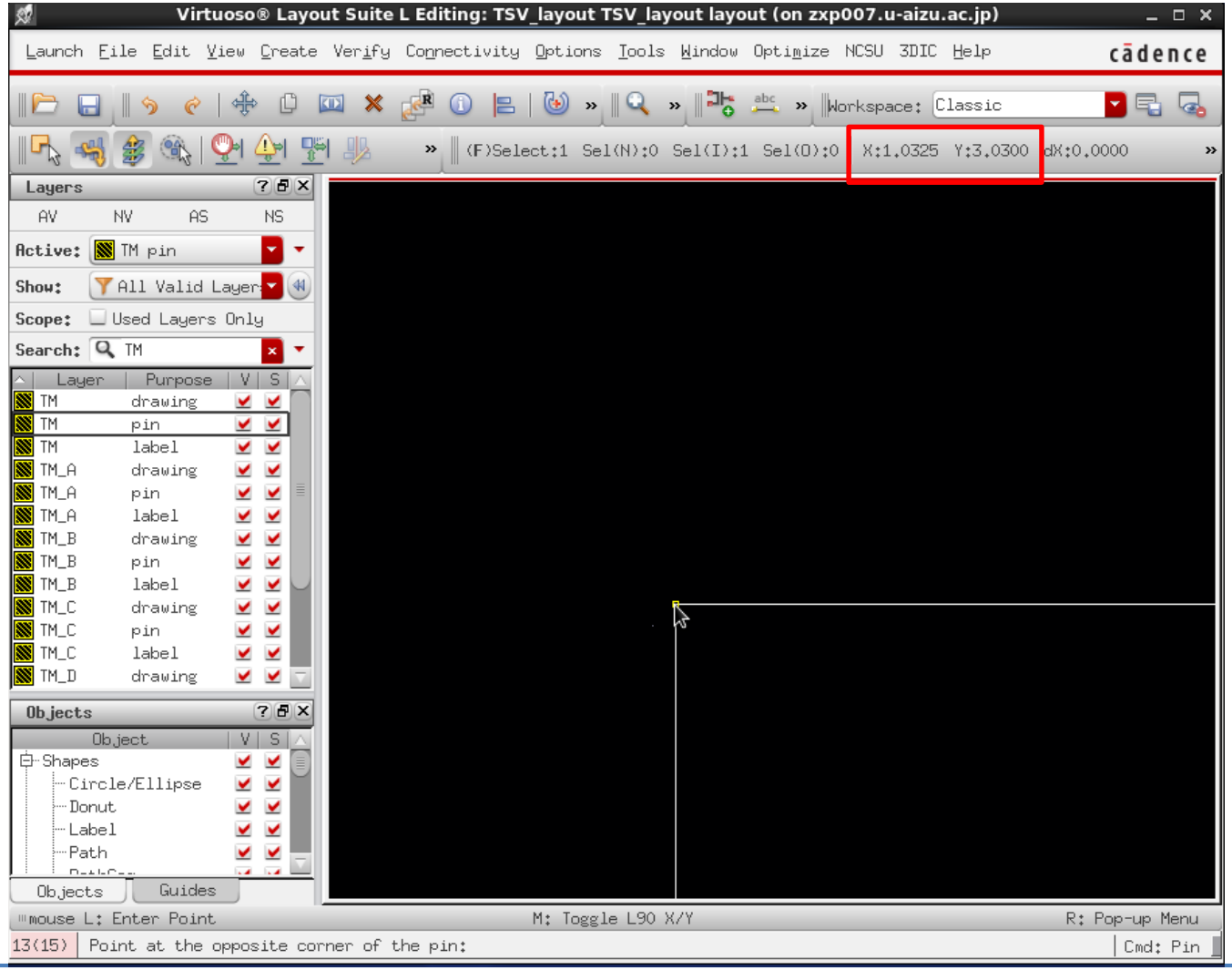
1. Click **TM pin**
2. Click **Create -> pin**

Step3: Create TSV layout
d- Create output pin

1. Type **OUT**
2. Check **rectangle**
3. Check **output**
4. Click **Hide**

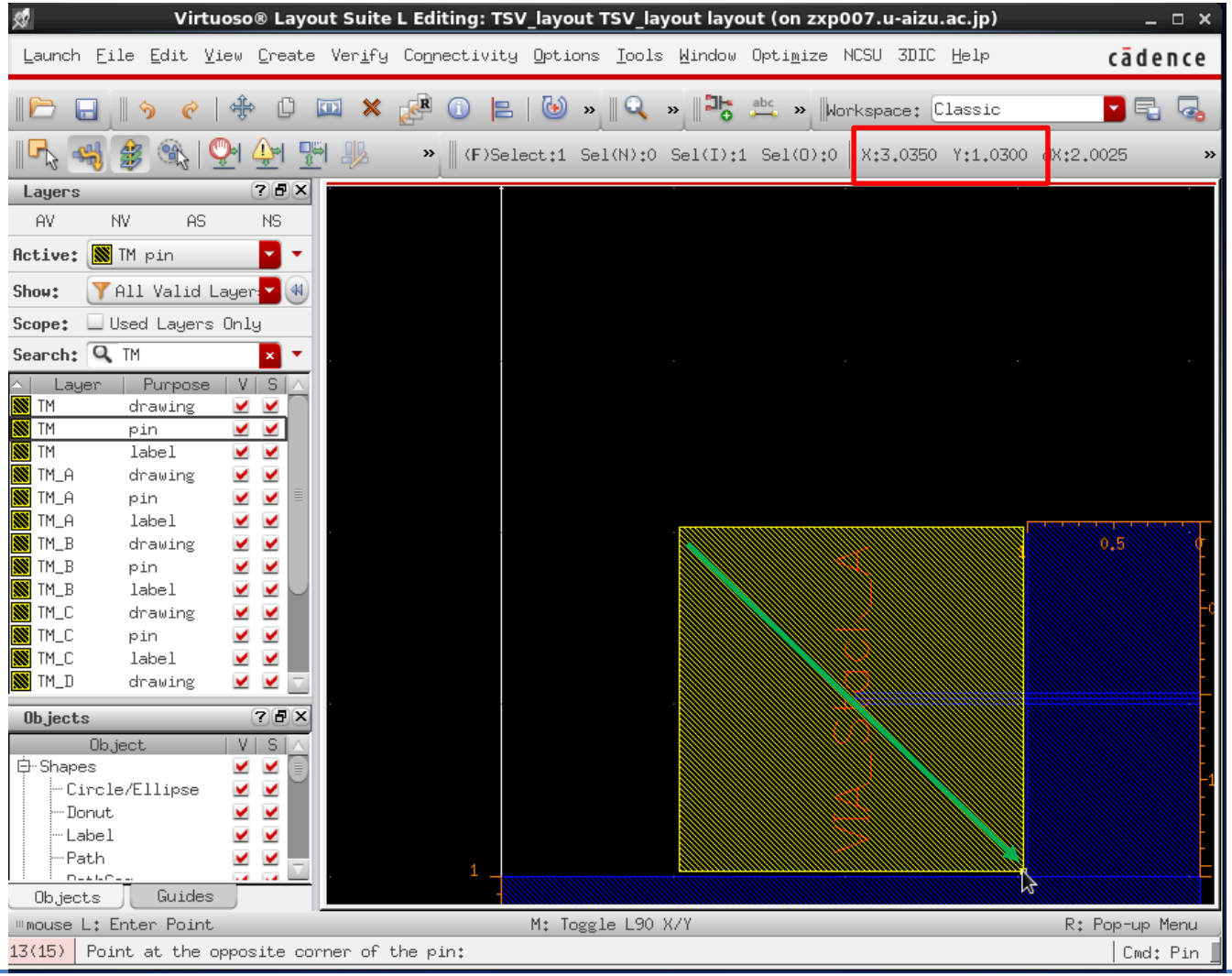


Step3: Create TSV layout
d- Create output pin



Click **point(X: 1.03 Y: 3.03)**

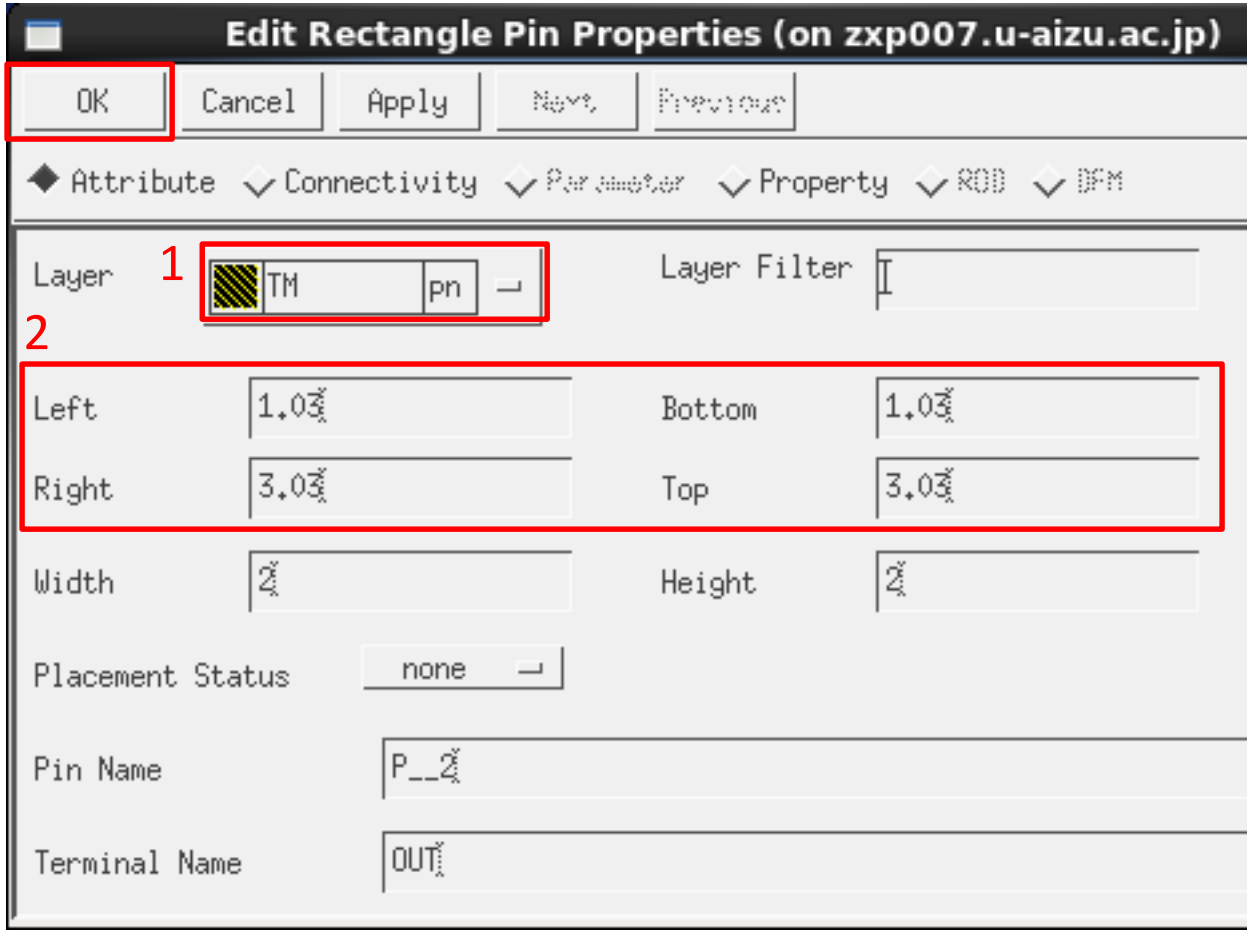
Step3: Create TSV layout
d- Create output pin



Drag the pointer to **point(X:3.03 Y: 1.03)** and click

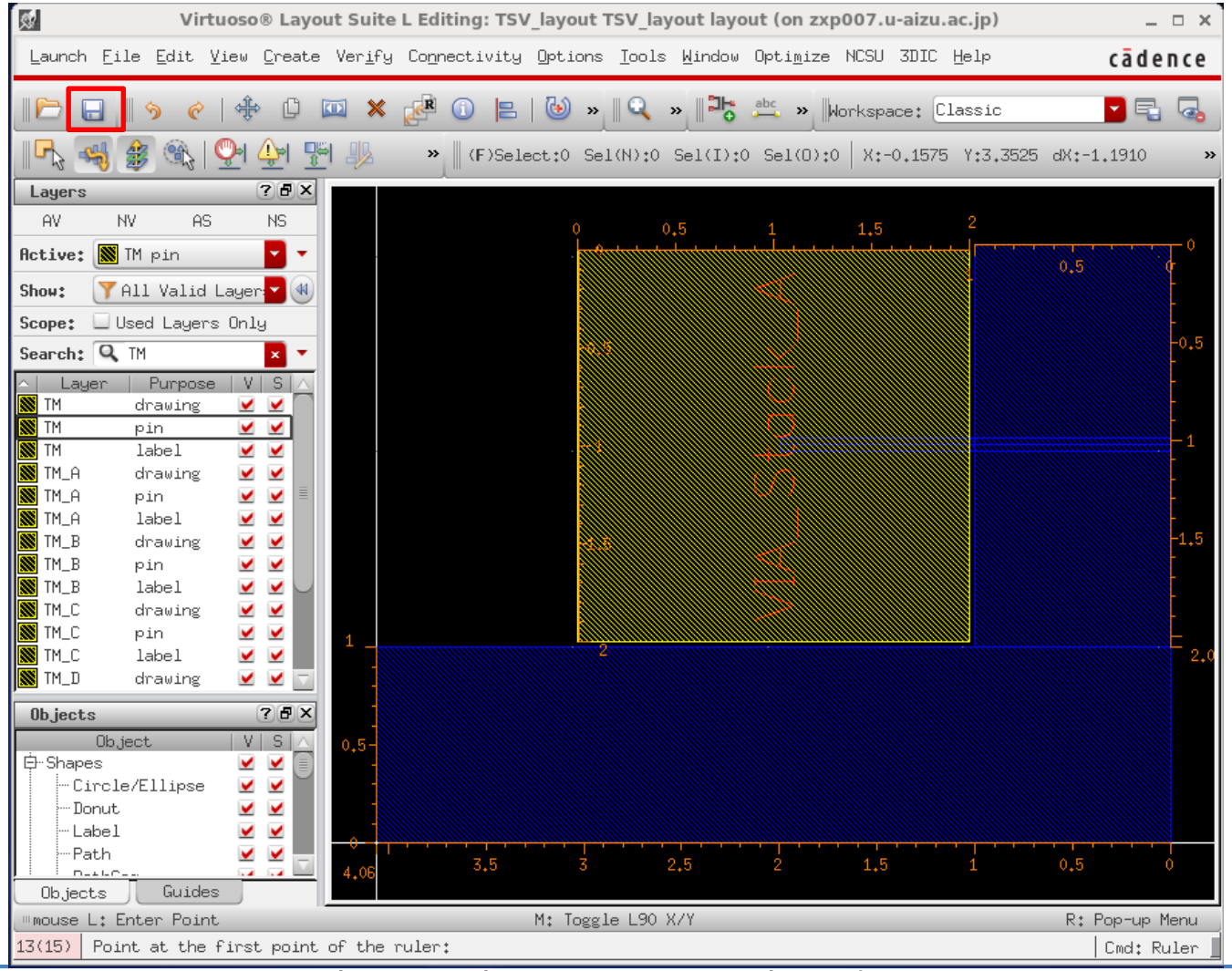
Step3: Create TSV layout
d- Create output pin

1. Select **TM pn**
2. Type
   - **Left: 1.03**
   - **Right: 3.03**
   - **Bottom: 1.03**
   - **Top: 3.03**
3. Click **OK**



Right click **created Output pin** and click **Properties**

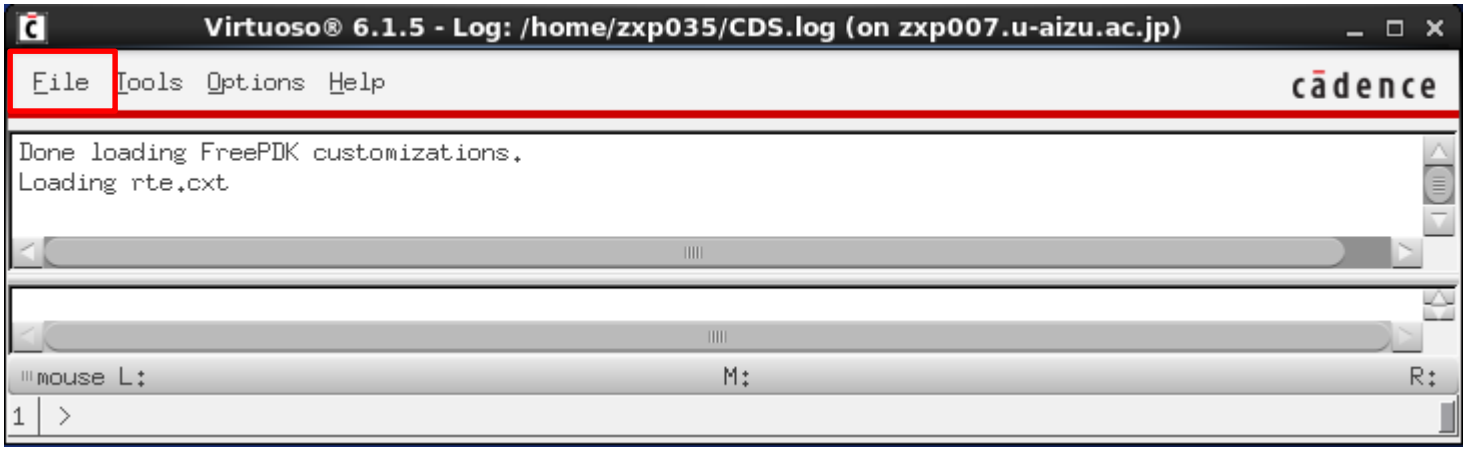
Step3: Create TSV layout



The TSV layout is completed.

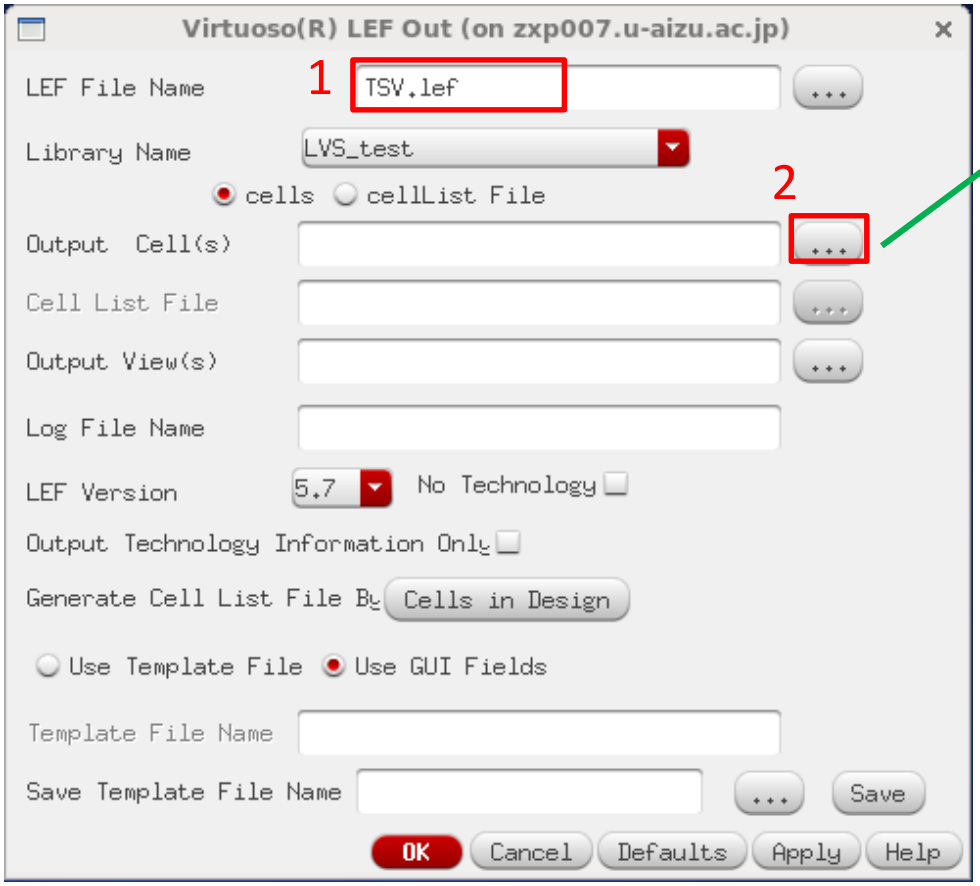
Click the **Save Icon**

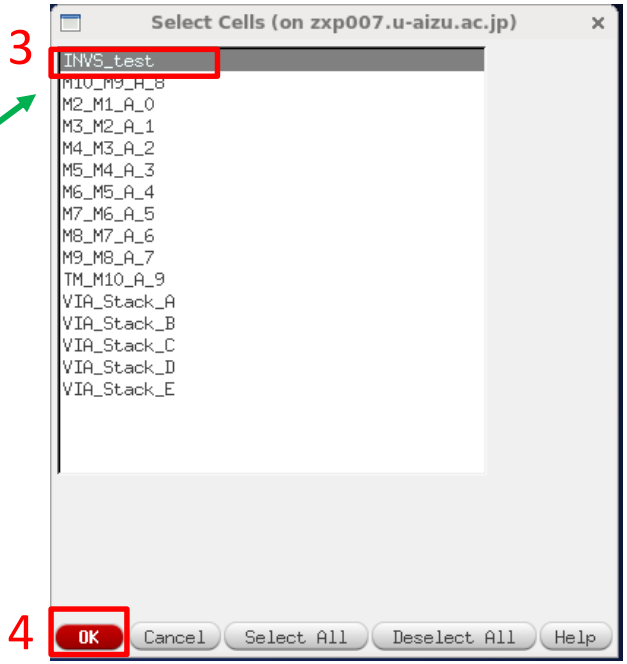
Step4: Export LEF



Click **File -> Export -> Lef** to create lef file

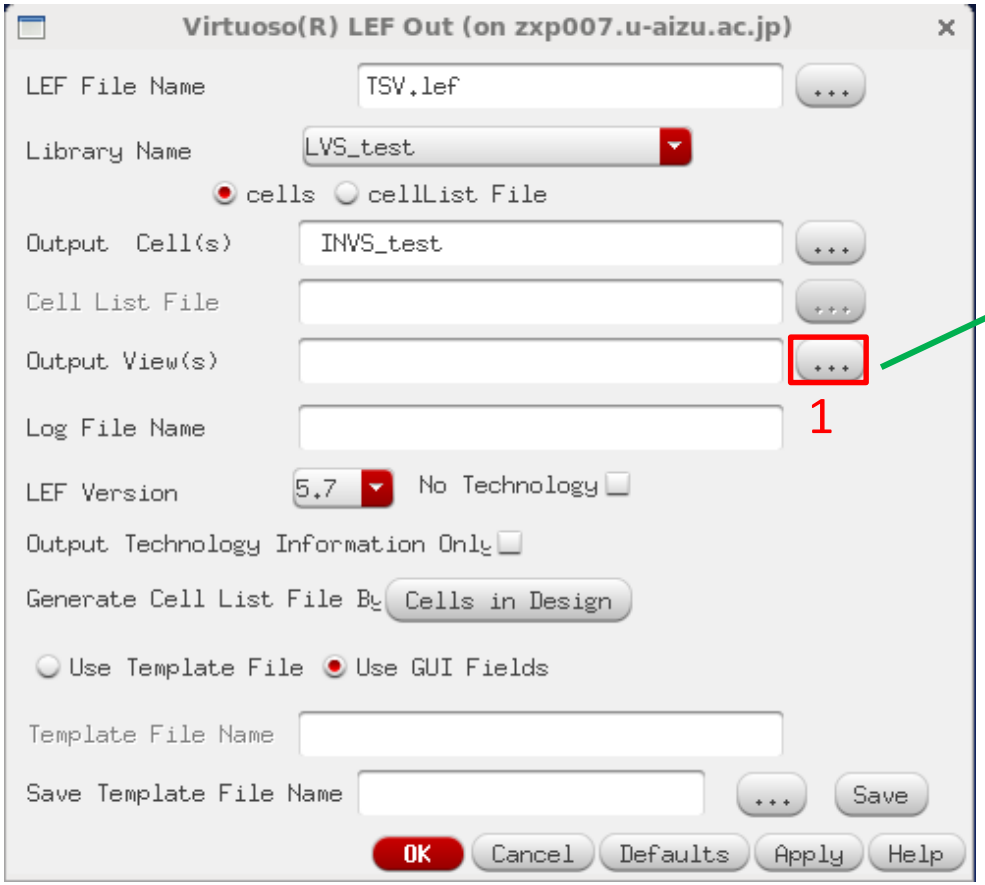
Step4: Export LEF

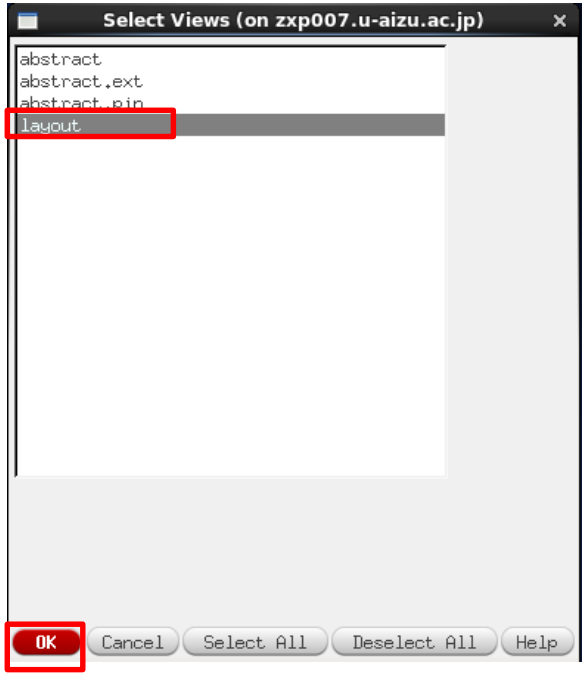




1. In *LEF File Name*, Type **TSV.lef**
2. Click on **Output Cell**
3. Click **INVS_test**
4. Click **OK**

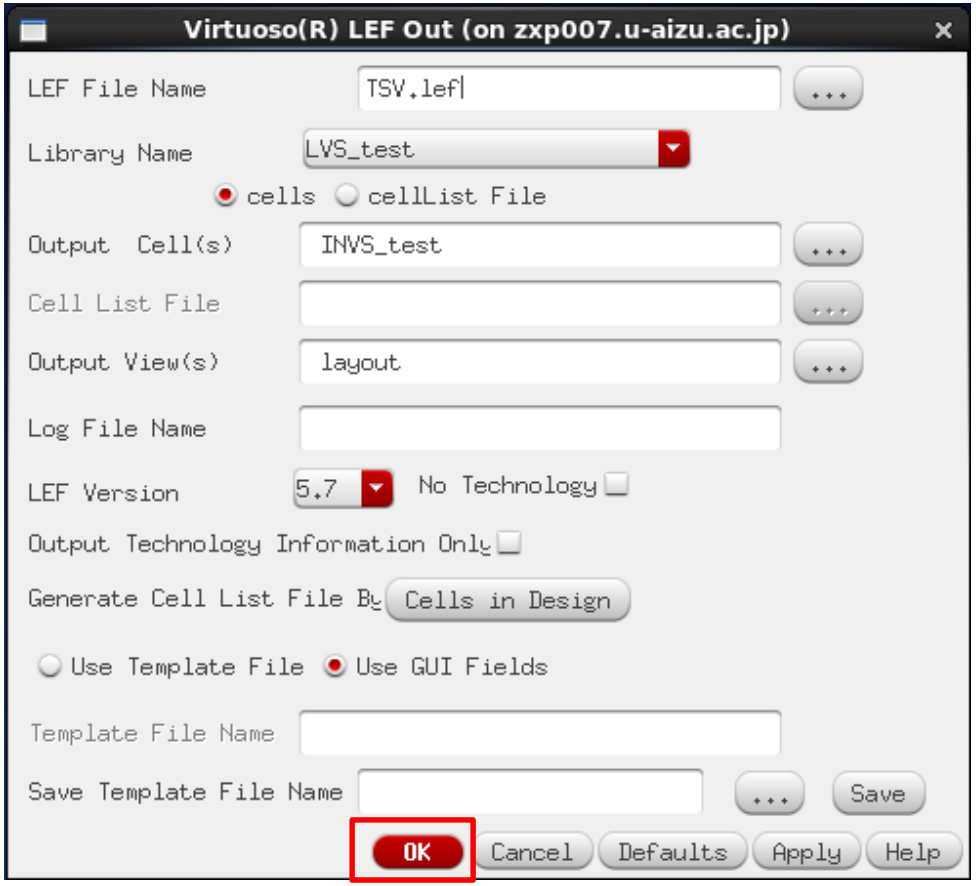
Step4: Export LEF





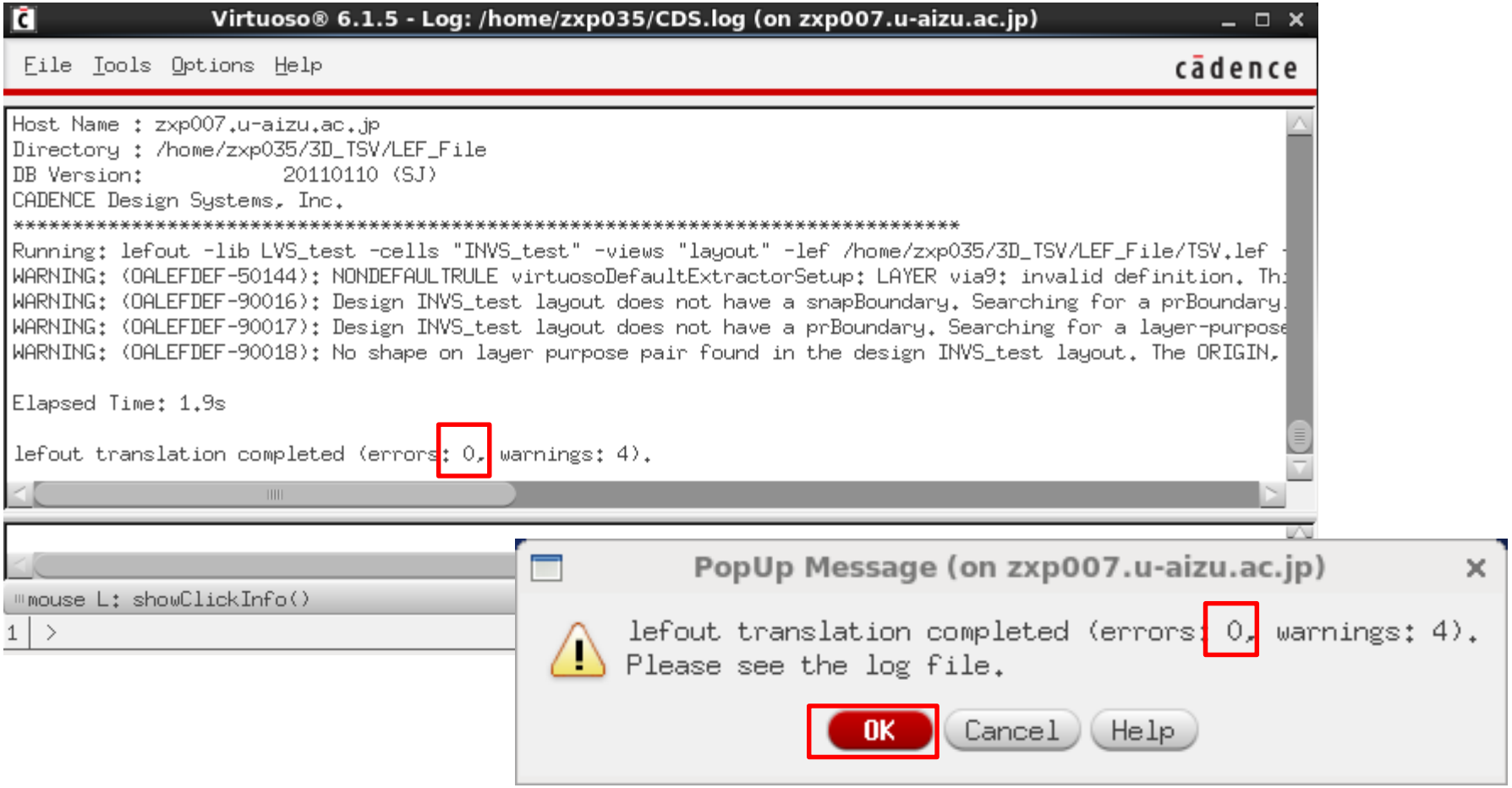
1. Click on **Output View**
2. Click **layout**
3. Click **OK**

Step4: Export LEF



Click **OK**

Step4: Export LEF

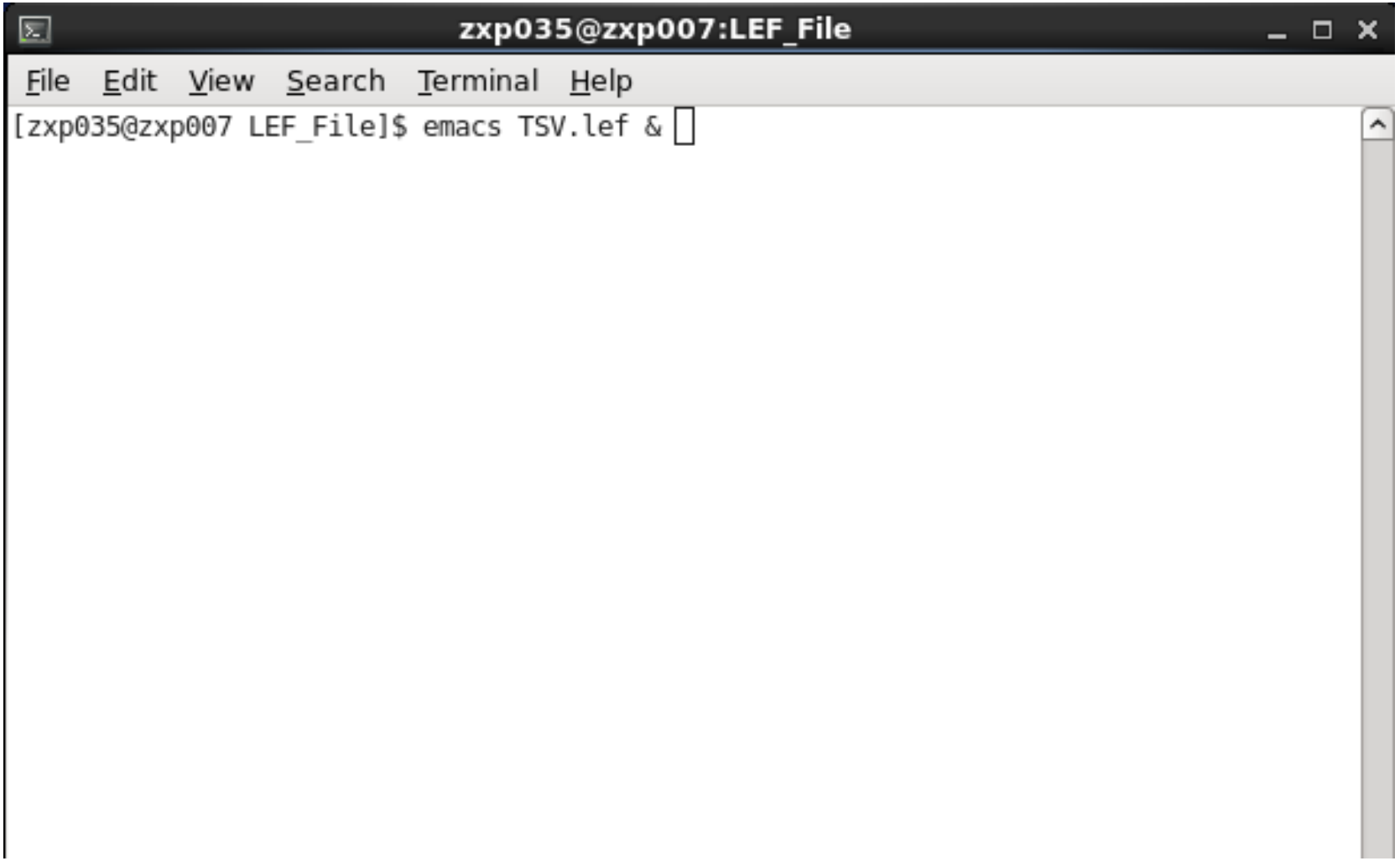


TSV.lef is exported without any errors.

<== Back to Contents

3. Modify lef file

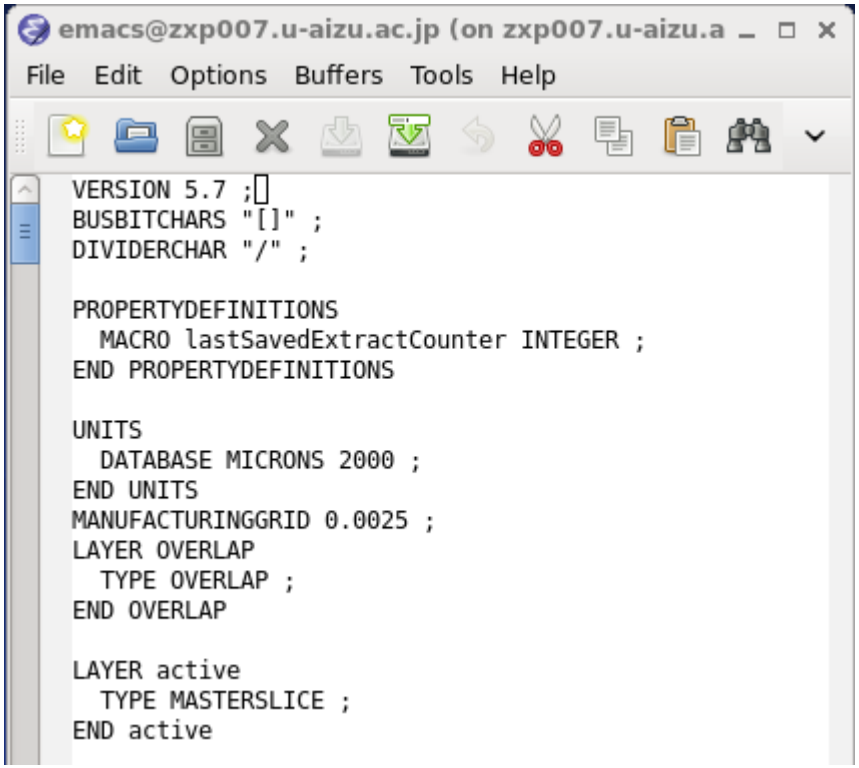
Modify lef file



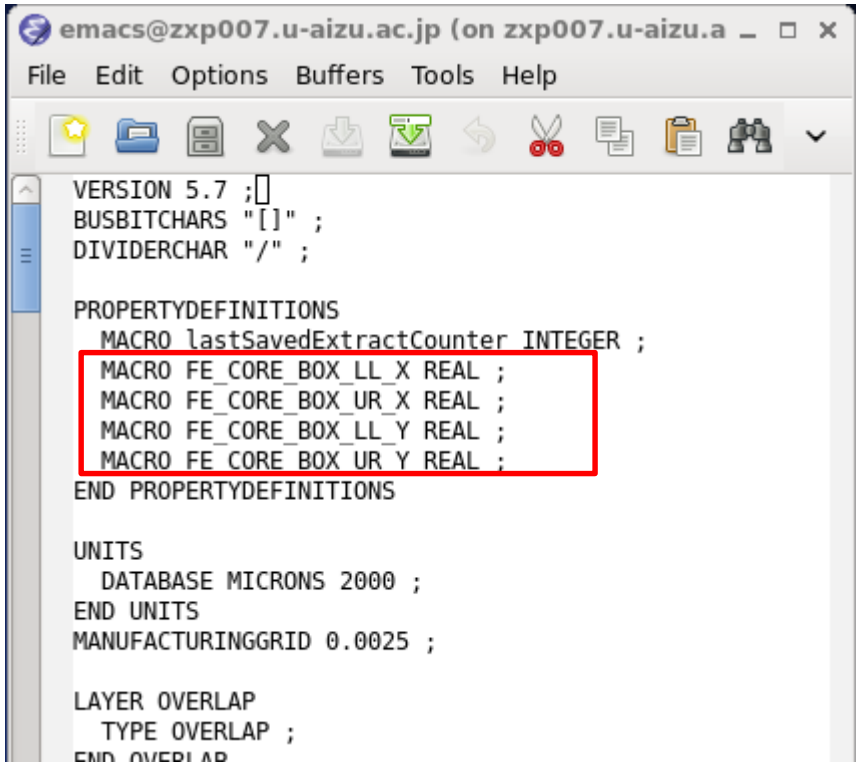
Type **emacs TSV.lef &**

Modify lef file

Before



After



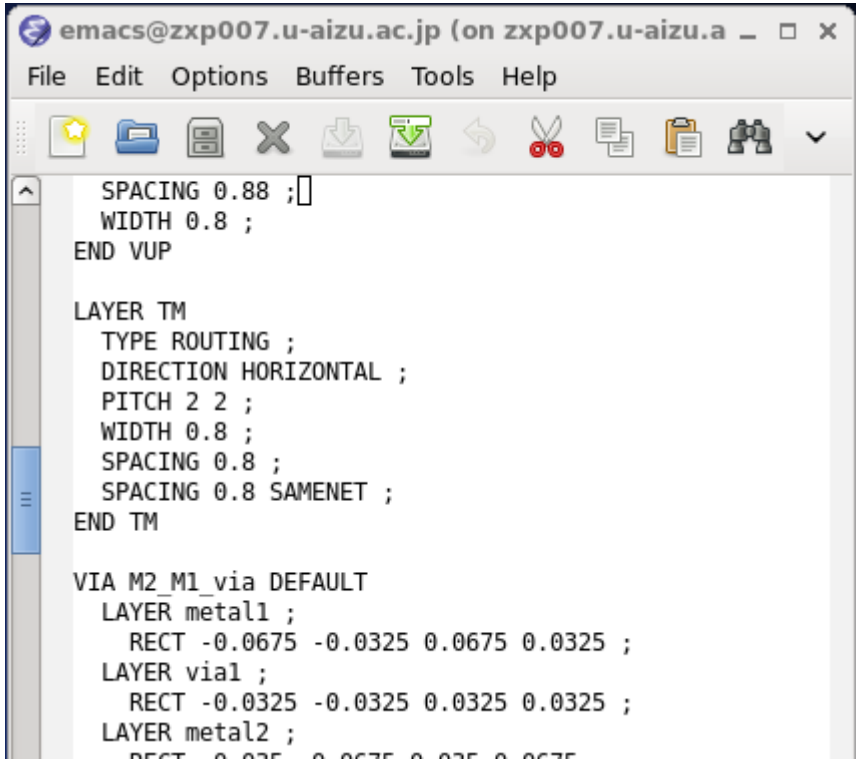
Add codes in red rectangle

Modify lef file

Before

```
LAYER TM
  TYPE ROUTING ;
  DIRECTION HORIZONTAL ;
  PITCH 2 2 ;
  WIDTH 0.8 ;
  SPACING 0.8 ;
  SPACING 0.8 SAMENET ;
END TM

LAYER VDN
  TYPE CUT ;
  SPACING 6 ;
  WIDTH 6 ;
END VDN

      ⋮

LAYER VTT_E
  TYPE CUT ;
  SPACING 6 ;
  WIDTH 6 ;
END VTT_E

VIA M2_M1_via DEFAULT
  LAYER metal1 ;
    RECT -0.0675 -0.0325 0.0675 0.0325 ;
  LAYER via1 ;
    RECT -0.0325 -0.0325 0.0325 0.0325 ;
  LAYER metal2 ;
    RECT -0.035 -0.0675 0.035 0.0675 ;
END M2_M1_via
```

After



Delete codes in a red part

Modify lef file

Before

```
VIA M10_M9_via DEFAULT
  LAYER metal9 ;
    RECT -0.4 -0.4 0.4 0.4 ;
  LAYER via9 ;
    RECT -0.4 -0.4 0.4 0.4 ;
  LAYER metal10 ;
    RECT -0.4 -0.4 0.4 0.4 ;
END M10_M9_via

VIA TM_M10_via DEFAULT
  LAYER metal10 ;
    RECT -0.4 -0.4 0.4 0.4 ;
  LAYER VUP ;
    RECT -0.4 -0.4 0.4 0.4 ;
  LAYER TM ;
    RECT -0.9975 -1.0 1.0025 1.0 ;
END TM_M10_via

VIA M2_M1_viaB DEFAULT
  LAYER metal1 ;
    RECT -0.0675 -0.0325 0.0675 0.0325 ;
  LAYER via1 ;
    RECT -0.0325 -0.0325 0.0325 0.0325 ;
  LAYER metal2 ;
    RECT -0.0675 -0.035 0.0675 0.035 ;
END M2_M1_viaB
```

After

```
VIA M10_M9_via DEFAULT
  LAYER metal9 ;
    RECT -0.4 -0.4 0.4 0.4 ;
  LAYER via9 ;
    RECT -0.4 -0.4 0.4 0.4 ;
  LAYER metal10 ;
    RECT -0.4 -0.4 0.4 0.4 ;
END M10_M9_via

VIA M2_M1_viaB DEFAULT
  LAYER metal1 ;
    RECT -0.0675 -0.0325 0.0675 0.0325 ;
  LAYER via1 ;
    RECT -0.0325 -0.0325 0.0325 0.0325 ;
  LAYER metal2 ;
    RECT -0.0675 -0.035 0.0675 0.035 ;
END M2_M1_viaB

VIA M2_M1_viaC DEFAULT
  LAYER metal1 ;
    RECT -0.0325 -0.0675 0.0325 0.0675 ;
  LAYER via1 ;
    RECT -0.0325 -0.0325 0.0325 0.0325 ;
  LAYER metal2 ;
    RECT -0.035 -0.0675 0.035 0.0675 ;
END M2_M1_viaC
```

Delete codes in red rectangle

Modify lef file

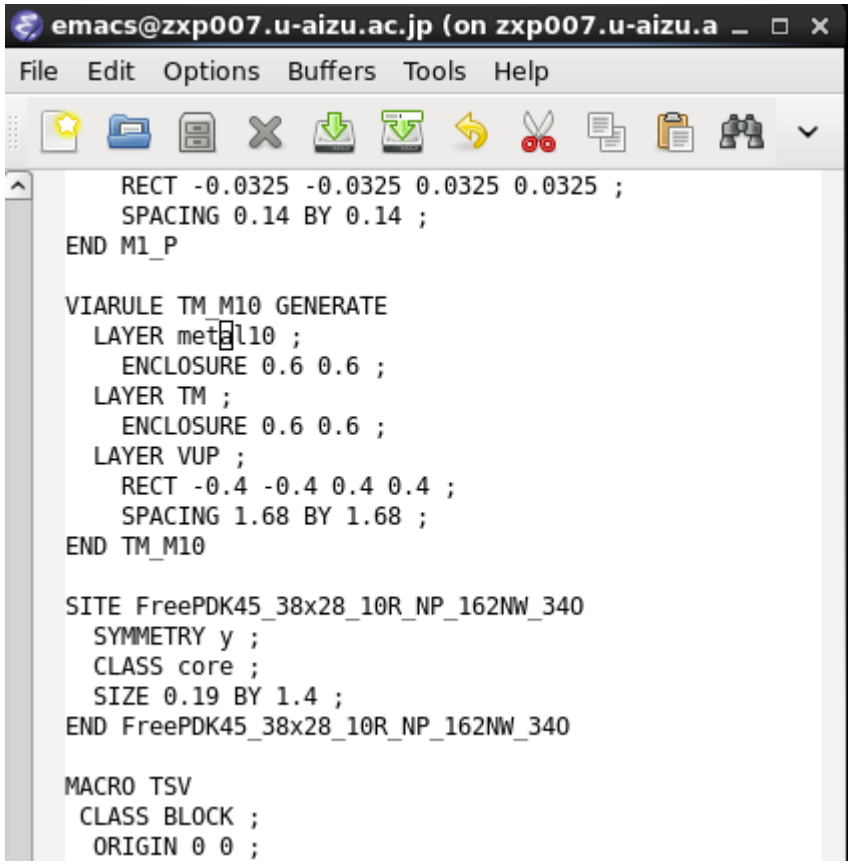
Before

```
VIARULE TM_M10 GENERATE
  LAYER metal10 ;
    ENCLOSURE 0.6 0.6 ;
  LAYER TM ;
    ENCLOSURE 0.6 0.6 ;
  LAYER VUP ;
    RECT -0.4 -0.4 0.4 0.4 ;
    SPACING 1.68 BY 1.68 ;
END TM_M10

VIARULE M1_BM GENERATE
  LAYER BM ;
    ENCLOSURE 0.6 0.6 ;
  LAYER metal1 ;
    ENCLOSURE 0.1 0.1 ;
  LAYER VDN ;
    RECT -3 -3 3 3 ;
    SPACING 12 BY 12 ;
END M1_BM
```

⋮

```
VIA BM_D_BM_E_45
  VIARULE BM_D_BM_E ;
  CUTSIZE 6 6 ;
  LAYERS BM_D VTT_E BM_E ;
  CUTSPACING 6 6 ;
  ENCLOSURE 0.6 0.6 0.6 0.6 ;
  ROWCOL 1 1 ;
END BM_D_BM_E_45

MACRO TSV
  PROPERTY lastSavedExtractCounter 1527 ;
END TSV
```

After



Delete codes in red rectangle

Modify lef file

Before

```
VIARULE TM_M10 GENERATE
  LAYER metal10 ;
    ENCLOSURE 0.6 0.6 ;
  LAYER TM ;
    ENCLOSURE 0.6 0.6 ;
  LAYER VUP ;
    RECT -0.4 -0.4 0.4 0.4 ;
    SPACING 1.68 BY 1.68 ;
END TM_M10

MACRO TSV
  PROPERTY lastSavedExtractCounter 1527 ;
END TSV
```

After

```
VIARULE TM_M10 GENERATE
  LAYER metal10 ;
    ENCLOSURE 0.6 0.6 ;
  LAYER TM ;
    ENCLOSURE 0.6 0.6 ;
  LAYER VUP ;
    RECT -0.4 -0.4 0.4 0.4 ;
    SPACING 1.68 BY 1.68 ;
END TM_M10

SITE FreePDK45_38x28_10R_NP_162NW_340
  SYMMETRY y ;
  CLASS core ;
  SIZE 0.19 BY 1.4 ;
END FreePDK45_38x28_10R_NP_162NW_340

MACRO TSV
 CLASS BLOCK ;
  ORIGIN 0 0 ;
  FOREIGN TSV 0 0 ;
  SIZE 4.06 BY 4.06 ;
  SYMMETRY X Y ;
  SITE FreePDK45_38x28_10R_NP_162NW_340 ;
```

Add codes in red rectangle

Modify lef file

Before

```
  CUTSPACING 6 6 ;
  ENCLOSURE 0.6 0.6 0.6 0.6 ;
  ROWCOL 1 1 ;
END BM_D_BM_E_45

MACRO TSV
  PROPERTY lastSavedExtractCounter 1527 ;
END TSV

END LIBRARY
```

After

```
  CLASS core ;
  SIZE 0.19 BY 1.4 ;
END FreePDK45_38x28_10R_NP_162NW_34O

MACRO TSV
 CLASS BLOCK ;
  ORIGIN 0 0 ;
  FOREIGN TSV 0 0 ;
  SIZE 4.06 BY 4.06 ;
  SYMMETRY X Y ;
  SITE FreePDK45_38x28_10R_NP_162NW_34O ;

  PIN data_out
    DIRECTION OUTPUT ;
    USE SIGNAL ;
    PORT
      LAYER TM ;
        RECT 1.03 1.03 3.03 3.03 ;
    END
  END data_out
  PIN data_in
    DIRECTION INPUT ;
    USE SIGNAL ;
    PORT
      LAYER metal1 ;
        RECT 3.06 1 4.06 3.06 ;
    END
    PORT
      LAYER metal1 ;
        RECT 0 0 4.06 1 ;
    END
  END data_in
```

Add codes in red rectangle

Modify lef file

Before

```
  PIN data_out
    DIRECTION OUTPUT ;
    USE SIGNAL ;
    PORT
      LAYER TM ;
        RECT 1.03 1.03 3.03 3.03 ;
    END
  END data_out
  PIN data_in
    DIRECTION INPUT ;
    USE SIGNAL ;
    PORT
      LAYER metal1 ;
        RECT 3.06 1 4.06 3.06 ;
    END
    PORT
      LAYER metal1 ;
        RECT 0 0 4.06 1 ;
    END
  END data_in
  PROPERTY lastSavedExtractCounter 1527 ;
END TSV

END LIBRARY
```

After

```
  END data_in

  OBS
    LAYER metal1 ;
      RECT 0 0 4.06 1.0 ;
      RECT 0 1.0 1.0 3.06 ;
      RECT 3.06 1.0 4.06 4.06 ;
      RECT 0 3.06 4.06 4.06 ;
    LAYER metal2 ;
      RECT 0 0 4.06 1.0 ;
      RECT 0 1.0 1.0 3.06 ;
      RECT 3.06 1.0 4.06 4.06 ;
      RECT 0 3.06 4.06 4.06 ;
    LAYER metal3 ;
      RECT 0 0 4.06 1.0 ;
      RECT 0 1.0 1.0 3.06 ;
      RECT 3.06 1.0 4.06 4.06 ;
      RECT 0 3.06 4.06 4.06 ;
    LAYER metal4 ;
      RECT 0 0 4.06 1.0 ;
      RECT 0 1.0 1.0 3.06 ;
      RECT 3.06 1.0 4.06 4.06 ;
      RECT 0 3.06 4.06 4.06 ;
    LAYER metal5 ;
      RECT 0 0 4.06 1.0 ;
      RECT 0 1.0 1.0 3.06 ;
      RECT 3.06 1.0 4.06 4.06 ;
      RECT 0 3.06 4.06 4.06 ;
    LAYER metal6 ;
      RECT 0 0 4.06 1.0 ;
      RECT 0 1.0 1.0 3.06 ;
      RECT 3.06 1.0 4.06 4.06 ;
      RECT 0 3.06 4.06 4.06 ;
```

Add codes in red rectangle

Modify lef file

Before

```
    LAYER metal3 ;
      RECT 0 0 4.06 1.0 ;
      RECT 0 1.0 1.0 3.06 ;
      RECT 3.06 1.0 4.06 4.06 ;
      RECT 0 3.06 4.06 4.06 ;
    LAYER metal4 ;
      RECT 0 0 4.06 1.0 ;
      RECT 0 1.0 1.0 3.06 ;
      RECT 3.06 1.0 4.06 4.06 ;
      RECT 0 3.06 4.06 4.06 ;
    LAYER metal5 ;
      RECT 0 0 4.06 1.0 ;
      RECT 0 1.0 1.0 3.06 ;
      RECT 3.06 1.0 4.06 4.06 ;
      RECT 0 3.06 4.06 4.06 ;
    LAYER metal6 ;
      RECT 0 0 4.06 1.0 ;
      RECT 0 1.0 1.0 3.06 ;
      RECT 3.06 1.0 4.06 4.06 ;
      RECT 0 3.06 4.06 4.06 ;
  PROPERTY lastSavedExtractCounter 1527 ;
END TSV

END LIBRARY
```

After

```
    LAYER metal6 ;
      RECT 0 0 4.06 1.0 ;
      RECT 0 1.0 1.0 3.06 ;
      RECT 3.06 1.0 4.06 4.06 ;
      RECT 0 3.06 4.06 4.06 ;
    LAYER metal7 ;
      RECT 0 0 4.06 1.0 ;
      RECT 0 1.0 1.0 3.06 ;
      RECT 3.06 1.0 4.06 4.06 ;
      RECT 0 3.06 4.06 4.06 ;
    LAYER metal8 ;
      RECT 0 0 4.06 1.0 ;
      RECT 0 1.0 1.0 3.06 ;
      RECT 3.06 1.0 4.06 4.06 ;
      RECT 0 3.06 4.06 4.06 ;
    LAYER metal9 ;
      RECT 0 0 4.06 1.0 ;
      RECT 0 1.0 1.0 3.06 ;
      RECT 3.06 1.0 4.06 4.06 ;
      RECT 0 3.06 4.06 4.06 ;
    LAYER metal10 ;
      RECT 0 0 4.06 1.0 ;
      RECT 0 1.0 1.0 3.06 ;
      RECT 3.06 1.0 4.06 4.06 ;
      RECT 0 3.06 4.06 4.06 ;
   LAYER VUP ;
      RECT 0 0 4.06 1.0 ;
      RECT 0 1.0 1.0 3.06 ;
      RECT 3.06 1.0 4.06 4.06 ;
      RECT 0 3.06 4.06 4.06 ;
  END

  PROPERTY lastSavedExtractCounter 1527 ;
```

Add codes in red rectangle

Modify lef file

Before

```
      RECT 0 1.0 1.0 3.06 ;
      RECT 3.06 1.0 4.06 4.06 ;
      RECT 0 3.06 4.06 4.06 ;
    LAYER metal9 ;
      RECT 0 0 4.06 1.0 ;
      RECT 0 1.0 1.0 3.06 ;
      RECT 3.06 1.0 4.06 4.06 ;
      RECT 0 3.06 4.06 4.06 ;
    LAYER metal10 ;
      RECT 0 0 4.06 1.0 ;
      RECT 0 1.0 1.0 3.06 ;
      RECT 3.06 1.0 4.06 4.06 ;
      RECT 0 3.06 4.06 4.06 ;
   LAYER VUP ;
      RECT 0 0 4.06 1.0 ;
      RECT 0 1.0 1.0 3.06 ;
      RECT 3.06 1.0 4.06 4.06 ;
      RECT 0 3.06 4.06 4.06 ;
  END

  PROPERTY lastSavedExtractCounter 1527 ;
END TSV

END LIBRARY
```

After

```
   LAYER VUP ;
      RECT 0 0 4.06 1.0 ;
      RECT 0 1.0 1.0 3.06 ;
      RECT 3.06 1.0 4.06 4.06 ;
      RECT 0 3.06 4.06 4.06 ;
  END

  PROPERTY lastSavedExtractCounter 1527 ;
  PROPERTY FE_CORE_BOX_LL_X 0.0 ;
  PROPERTY FE_CORE_BOX_UR_X 4.06 ;
  PROPERTY FE_CORE_BOX_LL_Y 0.0 ;
  PROPERTY FE_CORE_BOX_UR_Y 4.06 ;
END TSV

END LIBRARY
```

Add codes in red rectangle

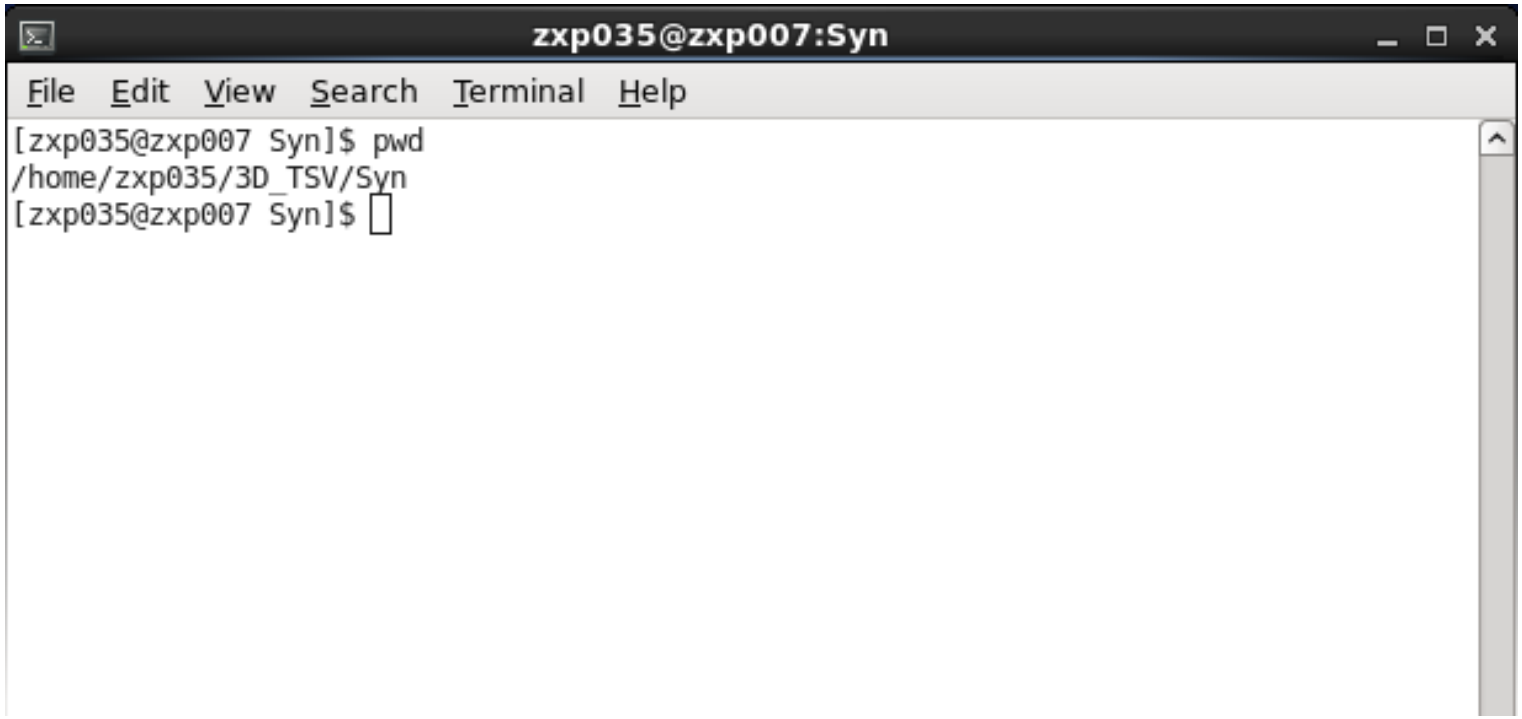
[<== Back to Contents]

4. Synthesis

Step0: Notes

- In this tutorial, the synthesis step is performed using a .tcl script with a minimal use of Synopsys Design Compiler Graphic User Interface (GUI).
- To understand in details the different synthesis operations using GUI, please refer to the previously made technical report [Ref.1].

Step1: Environment



Go to ***/home/zxp035/3D-TSV/Syn***

Step2: *syn_LAFT.tcl*

Type **emacs scripts/syn_LAFT.tcl &** to see the script

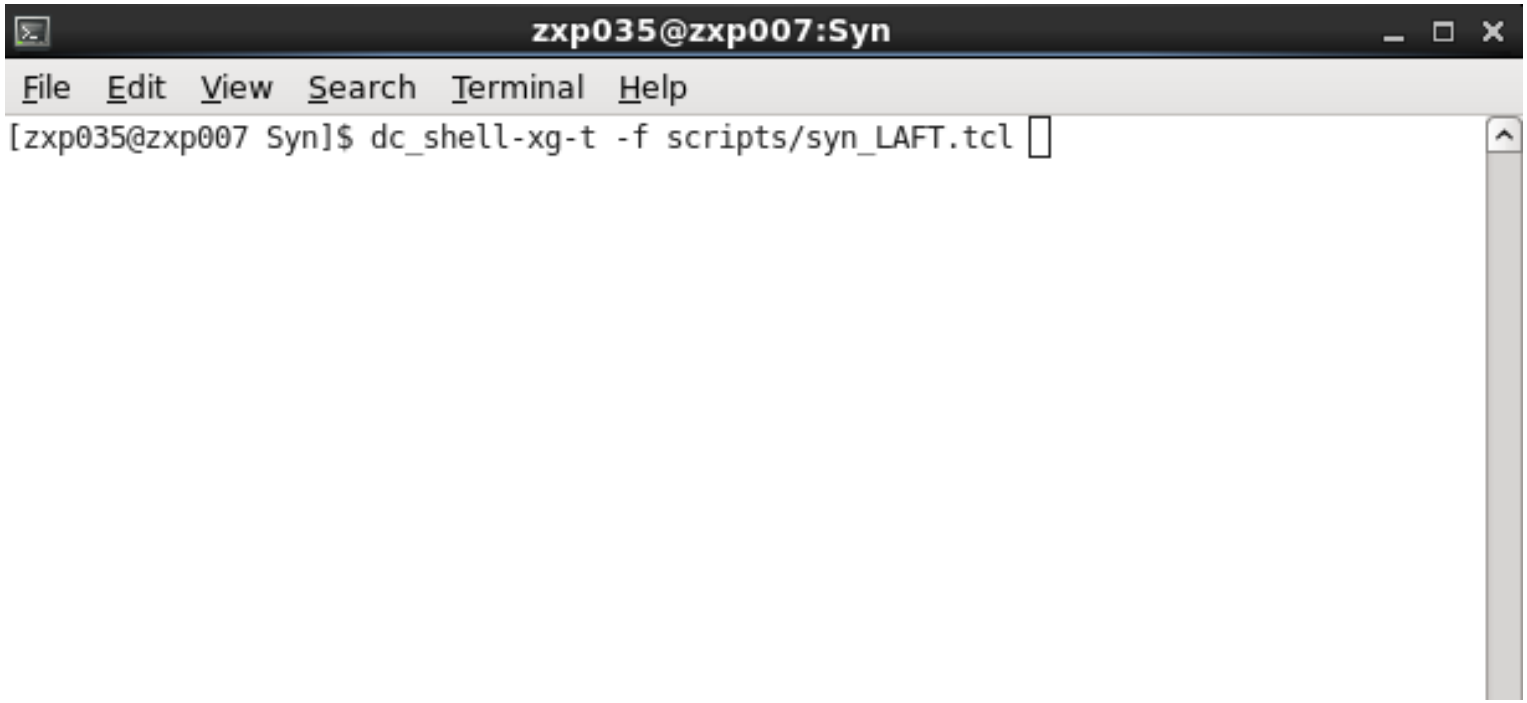
Step2: *syn_LAFT.tcl*

```
#
# Your design
#
set base_name "router_LAFT_TSV"
set clock_name "clk"
set clock_period 10.0

#
# Libraries
#
set target_library "/home/zxp035/lib/typical.db"
set synthetic_library "dw_foundation.sldb"
set link_library [concat "*" $target_library $synthetic_library]
set symbol_library "generic.sdb"
define_design_lib WORK -path ./WORK

#
# Read RTL file(s)
#

analyze -format verilog {./SRC/router_LAFT_TSV.v ./SRC/crossbar.v ./SRC/defines.v ./SRC/f
ifo.v ./SRC/input_port.v ./SRC/matrix_arb_formultistage.v ./SRC/mux_out.v ./SRC/route.v .
/SRC/non_minimal.v ./SRC/request_cntrl.v  ./SRC/stop_go.v ./SRC/sw_alloc.v ./SRC/TSV.v}
elaborate $base_name
current_design $base_name
link
uniquify

#
# Timing
#
create_clock -name $clock_name -period $clock_period [find port $clock_name]
set_clock_uncertainty 0.02 [get_clocks $clock_name]
set_input_delay 0.1 -clock clk [remove_from_collection [all_inputs] {clk reset}]
set_output_delay 0.1 -clock clk [all_outputs]
```

Step2: *syn_LAFT.tcl*

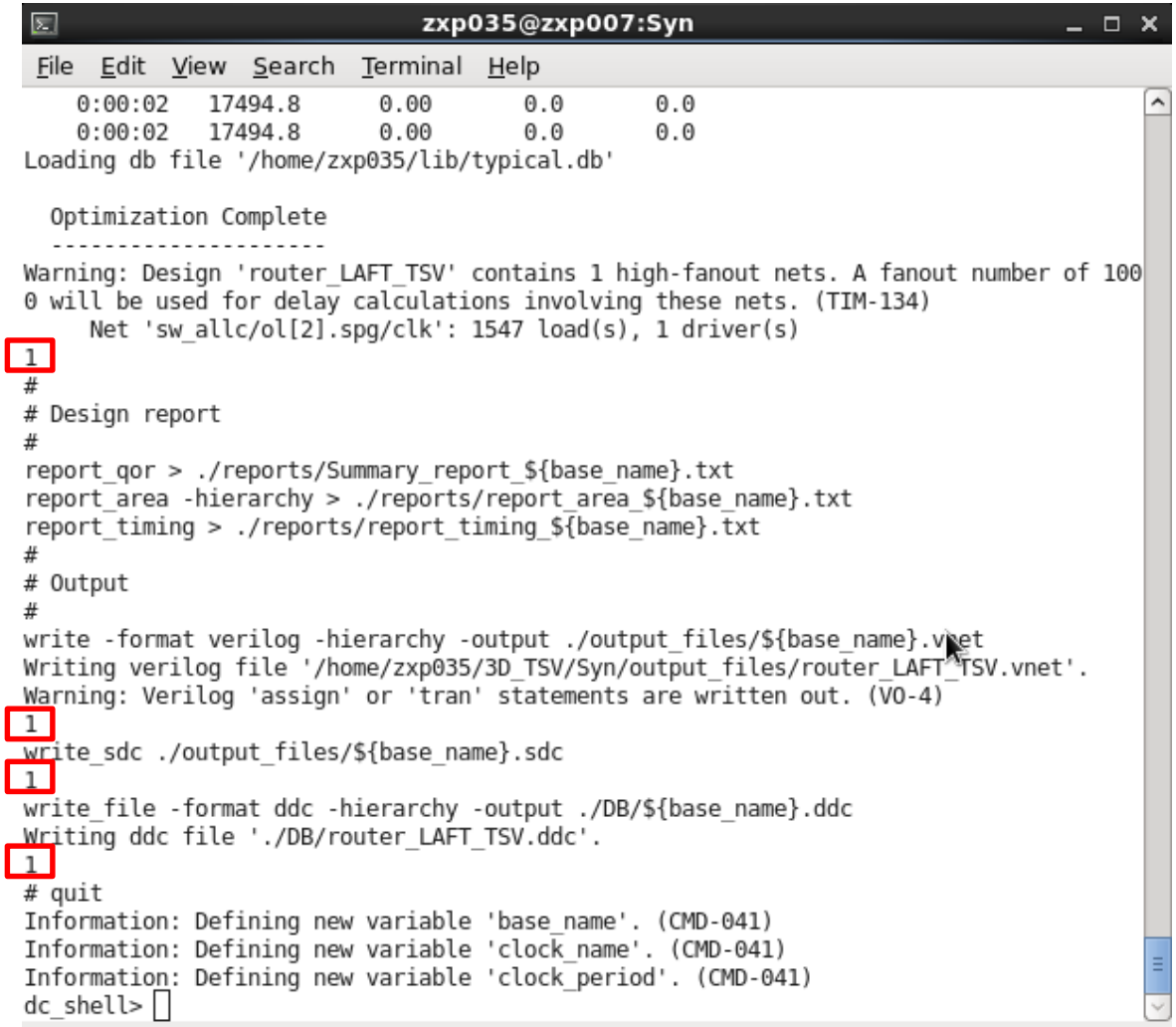
```
#
# Set wire load model
#
set_wire_load_model -name 5K_hvratio_1_1 -library NangateOpenCellLibrary


#
# Design synthesis
#
compile -map_effort high
compile -incremental_mapping -map_effort high

#
# Design report
#
report_qor > ./reports/Summary_report_${base_name}.txt
report_area -hierarchy > ./reports/report_area_${base_name}.txt
report_timing > ./reports/report_timing_${base_name}.txt
#
# Output
#
write -format verilog -hierarchy -output ./output_files/${base_name}.vnet
write_sdc ./output_files/${base_name}.sdc
write_file -format ddc -hierarchy -output ./DB/${base_name}.ddc

# quit
```

Step2: Run *syn_LAFT.tcl*



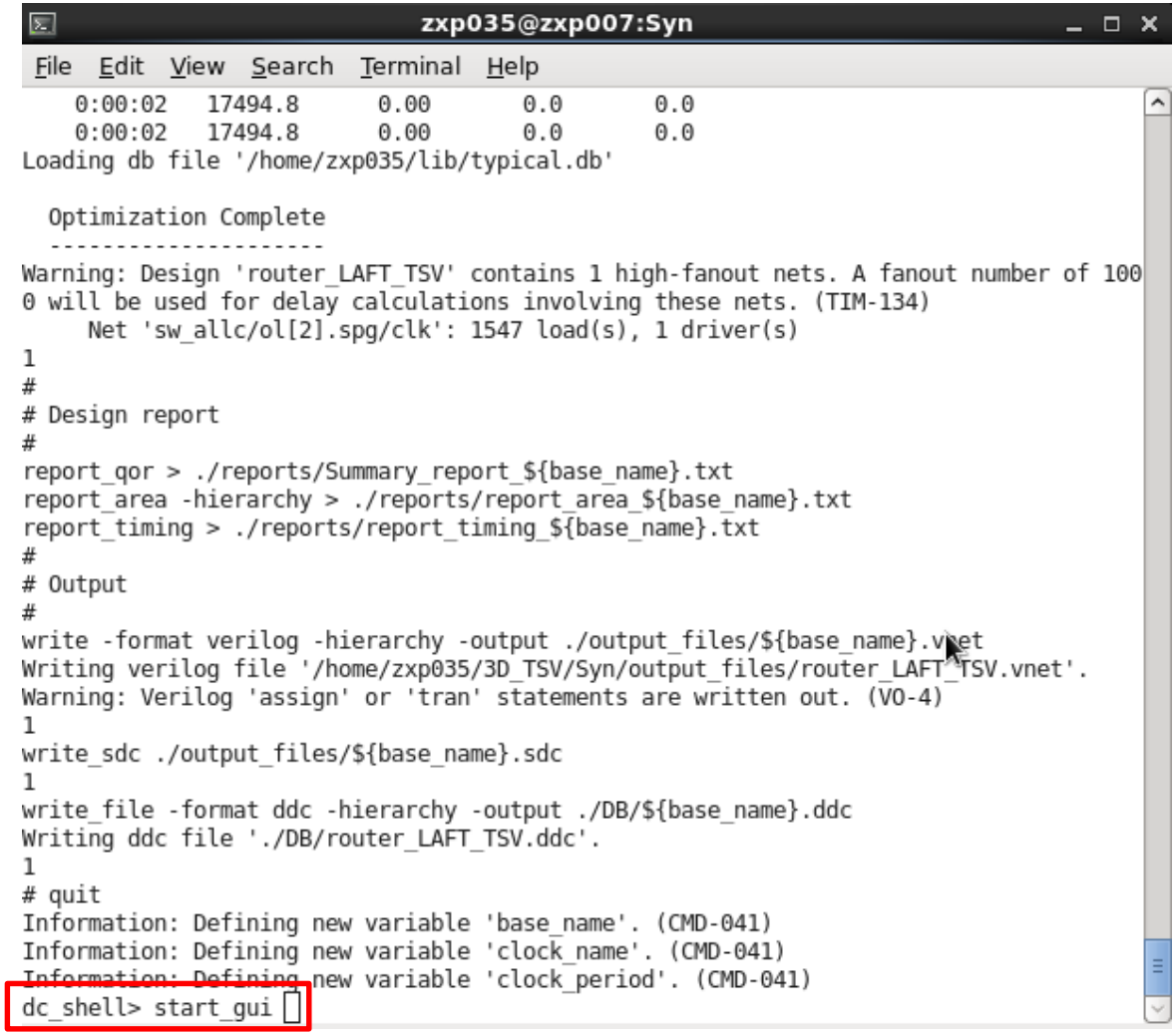
Type **dc_shell-xg –f scripts/syn_LAFT.tcl** to run the script by design compiler

Step2: Run script



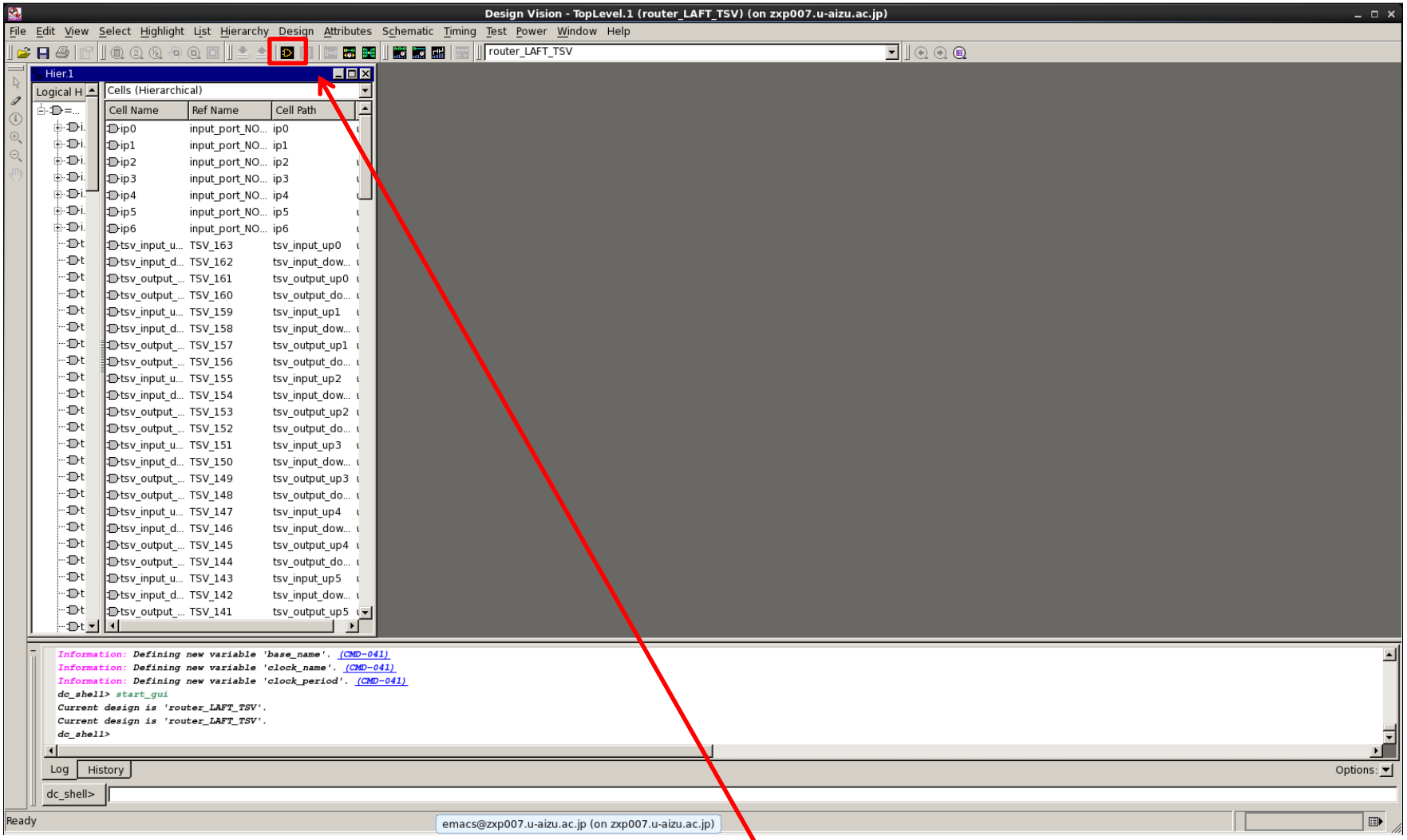
If red rectangles are all **“1”**, the script execution is succeeded

Step3: Run GUI



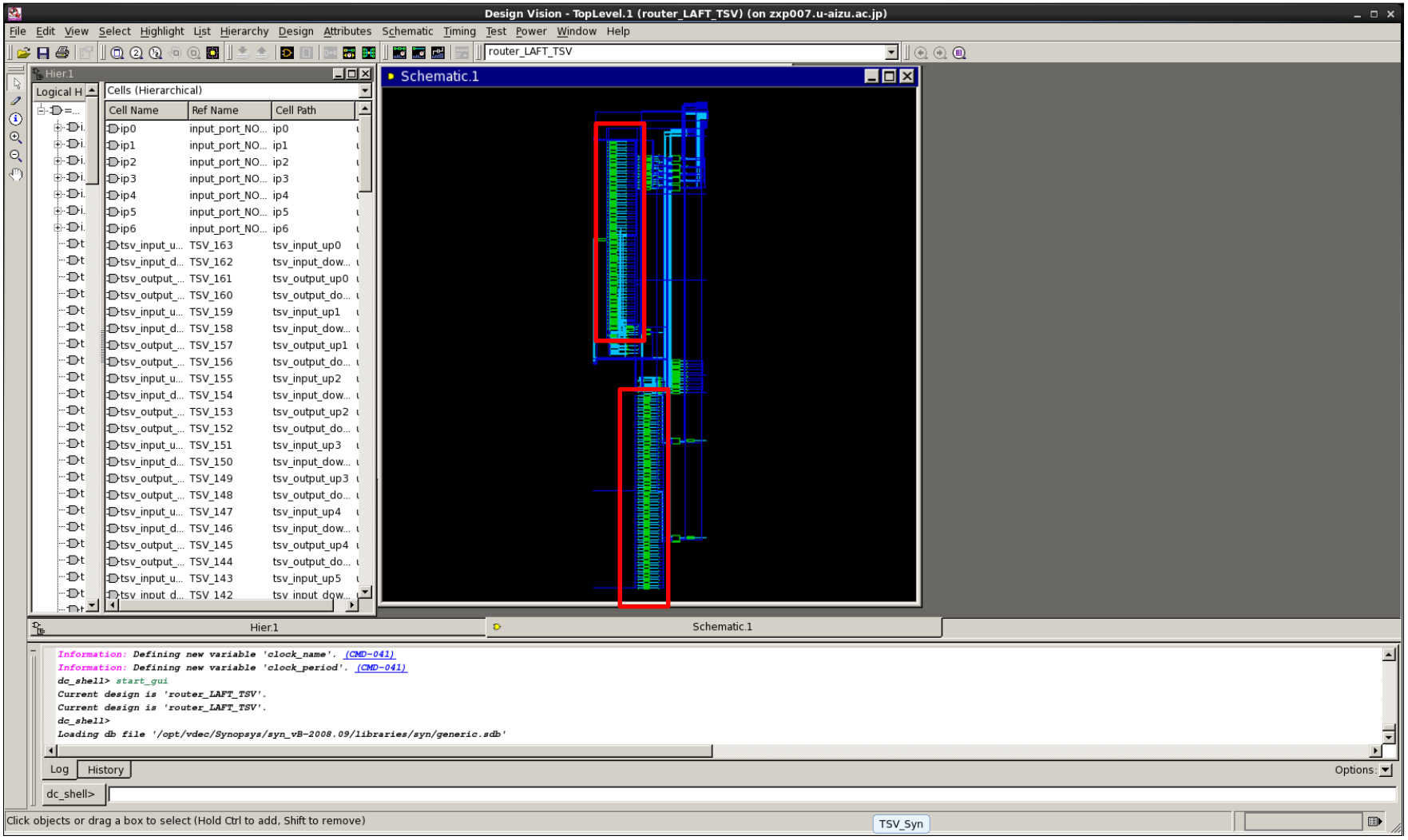
Type **start_gui**

Step3: Run GUI



Click 

Step3: Run GUI



Red rectangles are TSVs

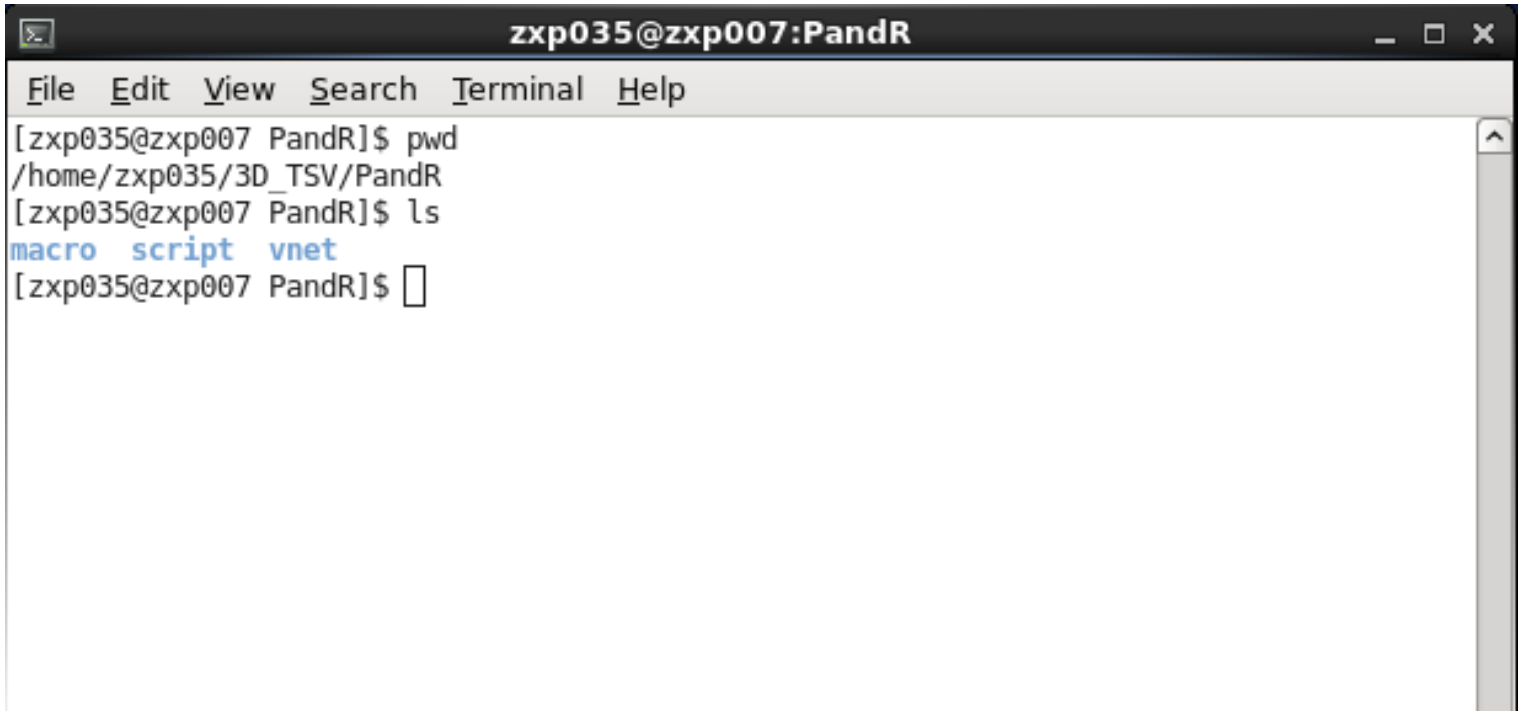
[<== Back to Contents]

4. Place & Route

Step0: Notes

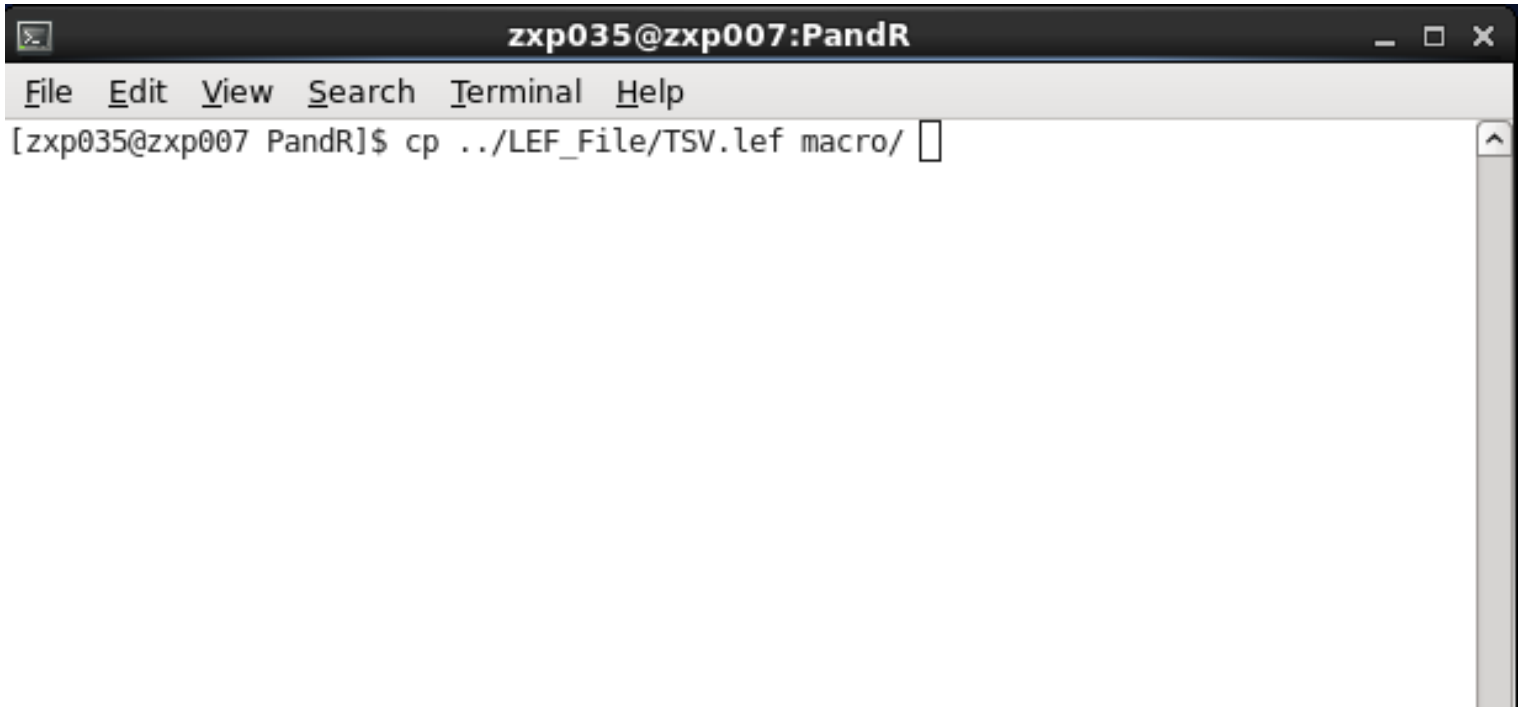
- In this tutorial, the Place and Route step is performed using both commands and GUI.
- How to execute the commands using GUI, is explained in details in the previously made technical report [Ref.1].

Step1: Environment



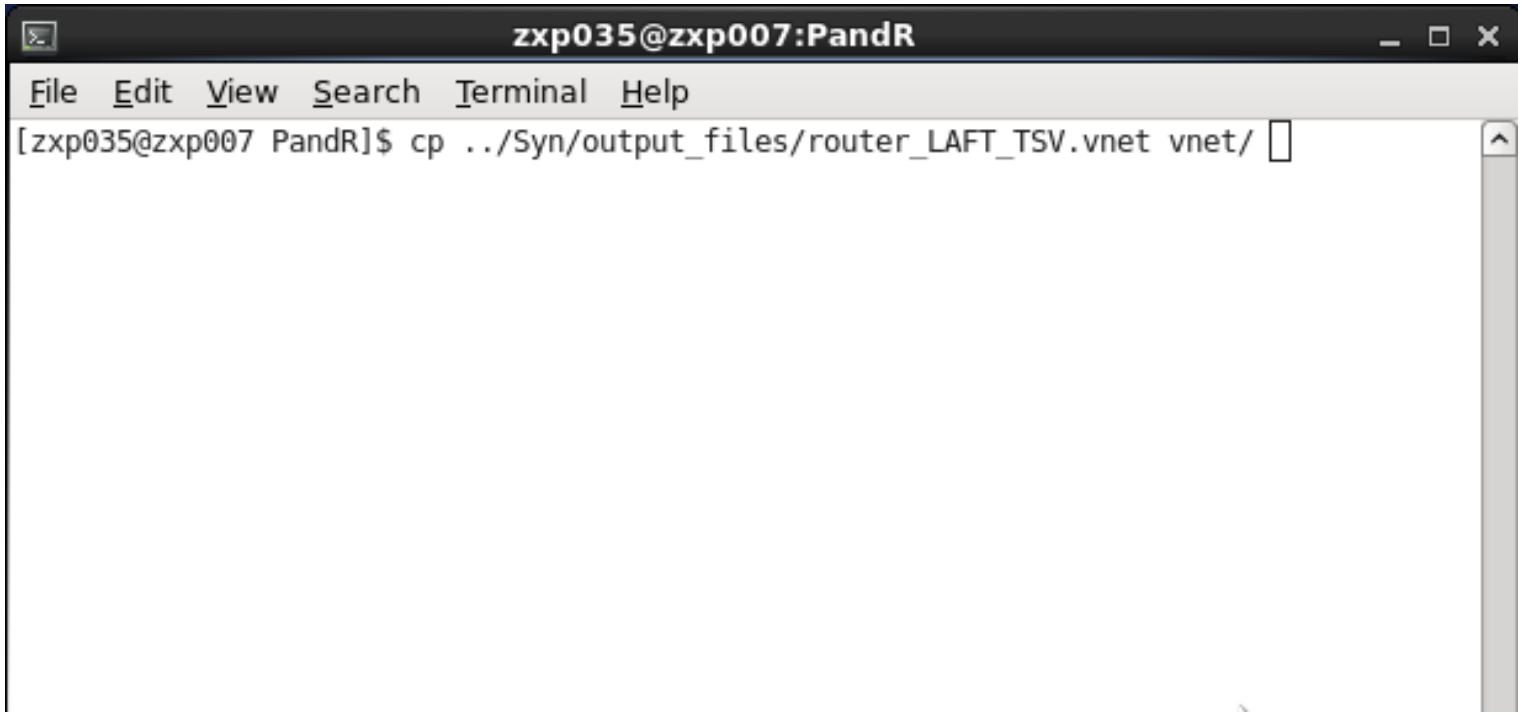
Go to ***/home/zxp035/3D-TSV/PandR***

Step1: Environment
a- Copy the files



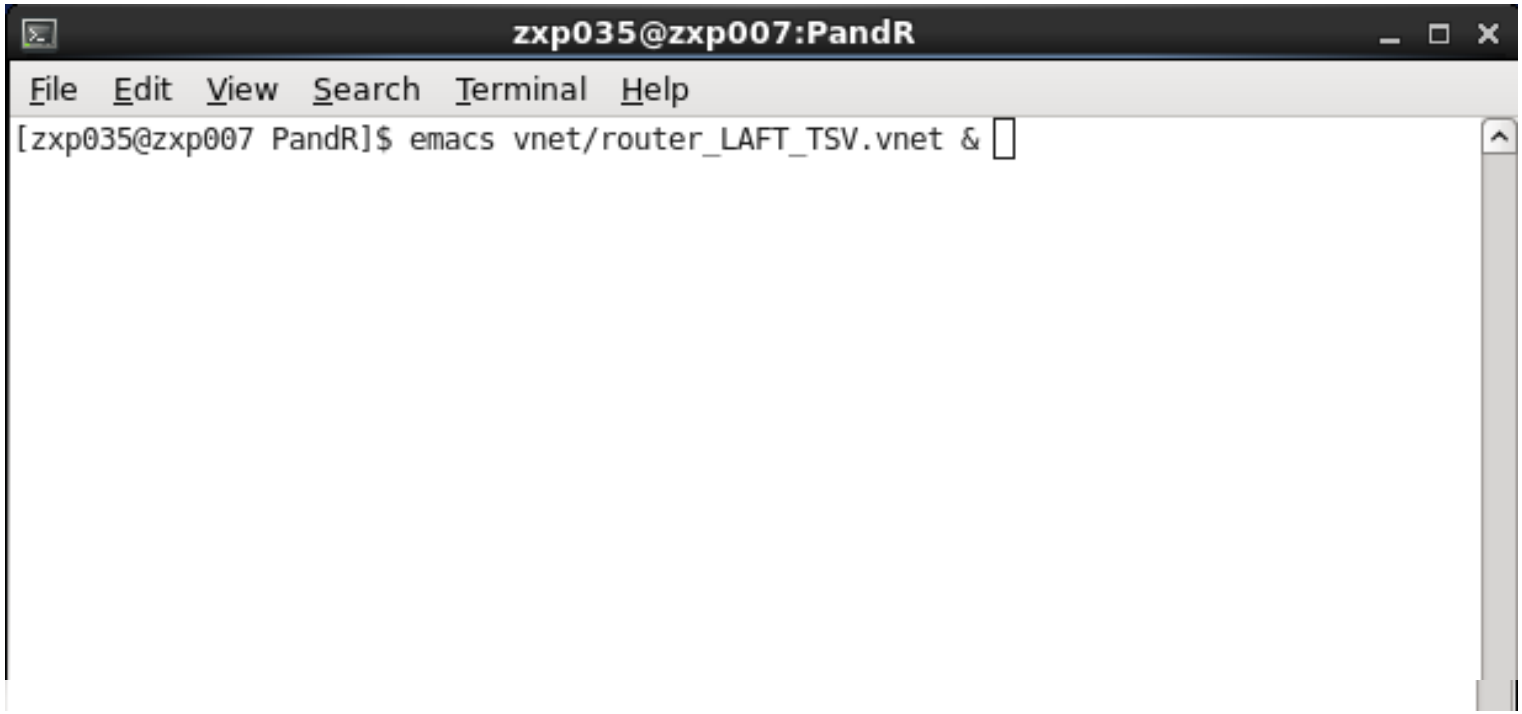
Type **cp ../LEF_File/TSV.lef macro/**

Step1: Environment
a- Copy the files



Type **cp ../Syn/output_files/router_LAFT.vnet vnet/**

Step1: Environment
b- Modify the .vnet file



Type **emacs vnet/router_LAFT_TSV.vnet &**

Step1: Environment b- Modify the vnet file

```
  MUX2_X1 U398 ( .A(\fifo[0][3] ), .B(\fifo[1][3] ), .S(n711), .Z(n458) );
  MUX2_X1 U399 ( .A(n458), .B(n457), .S(N15), .Z(second_item_nextport[2]) );
endmodule

module TSV_162 ( data_in, data_out );
  input data_in;
  output data_out;
  wire   data_in;
  assign data_out = data_in;

endmodule


module TSV_161 ( data_in, data_out );
  input data_in;
```

⋮

```
module TSV_1 ( data_in, data_out );
  input data_in;
  output data_out;
  wire   data_in;
  assign data_out = data_in;

endmodule


module TSV_0 ( data_in, data_out );
  input data_in;
  output data_out;
  wire   data_in;
  assign data_out = data_in;

endmodule


module non_minimal_NOUT7_FAULTY6_5 ( xdest, ydest, zdest, xaddr, yaddr, zaddr,
```

Delete codes in a red part

Step1: Environment b- Modify the vnet file

Before

```
  stop_go_0 \ol[6].spg  ( .clk(clk), .reset(n2), .data_sent(data_sent[6]),
        .stop_in(stop_in[6]), .blocked(blocked[6]) );
endmodule

module TSV_163 ( data_in, data_out );
  input data_in;
  output data_out;
  wire   data_in;
  assign data_out = data_in;

endmodule


module input_port_NOUT7_FAULTY6_FIFO_DEPTH4_FIFO_LOG2D2_FIFO_FULL_LVL2_0 ( clk,
```

After

```
  stop_go_0 \ol[6].spg  ( .clk(clk), .reset(n2), .data_sent(data_sent[6]),
        .stop_in(stop_in[6]), .blocked(blocked[6]) );
endmodule

module TSV ( data_in, data_out );
  input data_in;
  output data_out;
  wire   data_in;
  assign data_out = data_in;

endmodule


module input_port_NOUT7_FAULTY6_FIFO_DEPTH4_FIFO_LOG2D2_FIFO_FULL_LVL2_0 ( clk,
```

Change from **TSV_163** to **TSV**

Step1: Environment
b- Modify the vnet file

Before

```
input_port_NOUT7_FAULTY6_FIFO_DEPTH4_FIFO_LOG2D2_FIFO_FULL_LVL2_0 ip6 (
      .clk(clk), .reset(reset), .data_in(D_data_in), .data_out(
      cbar_data_in[237:204]), .faulty_in({c_faulty_in, faulty_in}), .sw_req(
      sw_req[6]), .port_req(port_req[48:42]), .sw_grant(sw_grant[6]),
      .stop_out(D_stop_out), .xaddr(xaddr), .yaddr(yaddr), .zaddr(zaddr),
      .stop_in({alloc_stop_in, stop_in}) );
  TSV_163 tsv_input_up0 ( .data_in(tsv_up_in[0]), .data_out(U_data_in[0]) );
  TSV_162 tsv_input_down0 ( .data_in(tsv_down_in[0]), .data_out(D_data_in[0])
       );
                    ⋮
  TSV_1 tsv_faulty_output_up5 ( .data_in(\faulty_out[5] ), .data_out(
      tsv_faulty_up_out[5]) );
  TSV_0 tsv_faulty_output_down5 ( .data_in(\faulty_out[5] ), .data_out(
      tsv_faulty_down_out[5]) );
  sw_alloc_NOUT7 sw_allc ( .clk(clk), .reset(reset), .sw_req(sw_req),
      .stop_in({alloc_stop_in, stop_in}), .data_sent(data_sent), .tail_sent(
      {tsv_down_out[0], tsv_up_out[0], data_out[136], data_out[102],
      data_out[68], data_out[34], data_out[0]}), .port_req(port_req),
      .grant_out(sw_grant), .sw_cntrl(sw_cntrl) );
  crossbar_NOUT7_NIN7_WIDTH34 cbar ( .clk(clk), .reset(reset), .cntrl(sw_cntrl),
 .data_in(cbar_data_in), .data_out({cbar_data_out, data_out}) );
endmodule
```

After

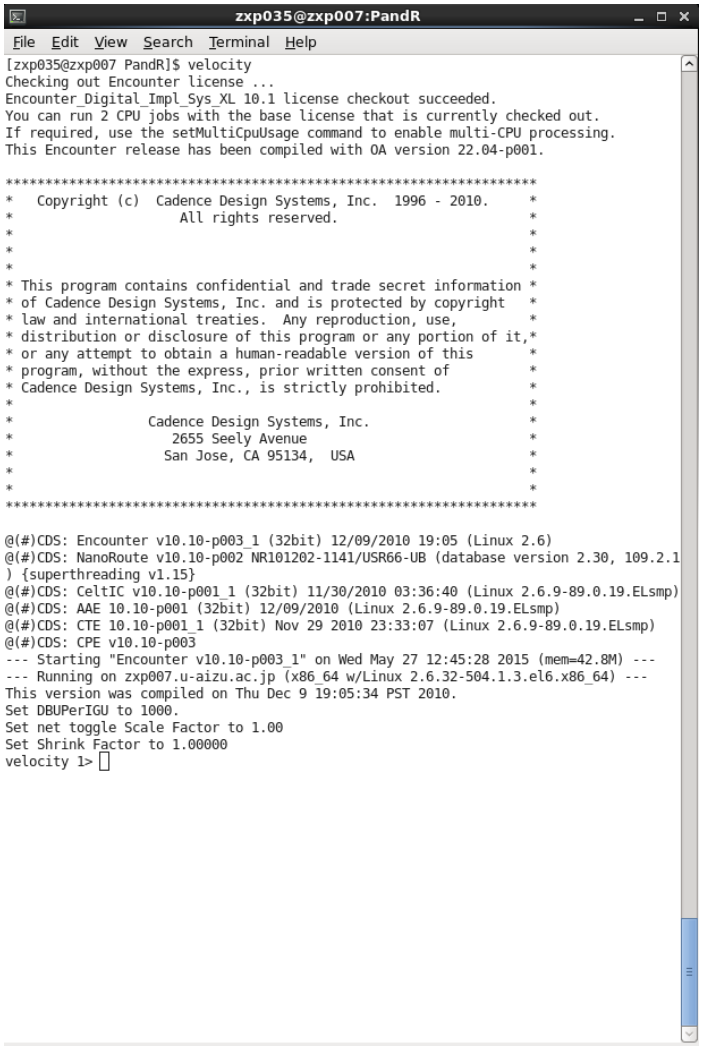
```
input_port_NOUT7_FAULTY6_FIFO_DEPTH4_FIFO_LOG2D2_FIFO_FULL_LVL2_0 ip6 (
      .clk(clk), .reset(reset), .data_in(D_data_in), .data_out(
      cbar_data_in[237:204]), .faulty_in({c_faulty_in, faulty_in}), .sw_req(
      sw_req[6]), .port_req(port_req[48:42]), .sw_grant(sw_grant[6]),
      .stop_out(D_stop_out), .xaddr(xaddr), .yaddr(yaddr), .zaddr(zaddr),
      .stop_in({alloc_stop_in, stop_in}) );
  TSV tsv_input_up0 ( .data_in(tsv_up_in[0]), .data_out(U_data_in[0]) );
  TSV tsv_input_down0 ( .data_in(tsv_down_in[0]), .data_out(D_data_in[0])
       );
                    ⋮
  TSV tsv_faulty_output_up5 ( .data_in(\faulty_out[5] ), .data_out(
      tsv_faulty_up_out[5]) );
  TSV tsv_faulty_output_down5 ( .data_in(\faulty_out[5] ), .data_out(
      tsv_faulty_down_out[5]) );
  sw_alloc_NOUT7 sw_allc ( .clk(clk), .reset(reset), .sw_req(sw_req),
      .stop_in({alloc_stop_in, stop_in}), .data_sent(data_sent), .tail_sent(
      {tsv_down_out[0], tsv_up_out[0], data_out[136], data_out[102],
      data_out[68], data_out[34], data_out[0]}), .port_req(port_req),
      .grant_out(sw_grant), .sw_cntrl(sw_cntrl) );
  crossbar_NOUT7_NIN7_WIDTH34 cbar ( .clk(clk), .reset(reset), .cntrl(sw_cntrl),
 .data_in(cbar_data_in), .data_out({cbar_data_out, data_out}) );
endmodule
```

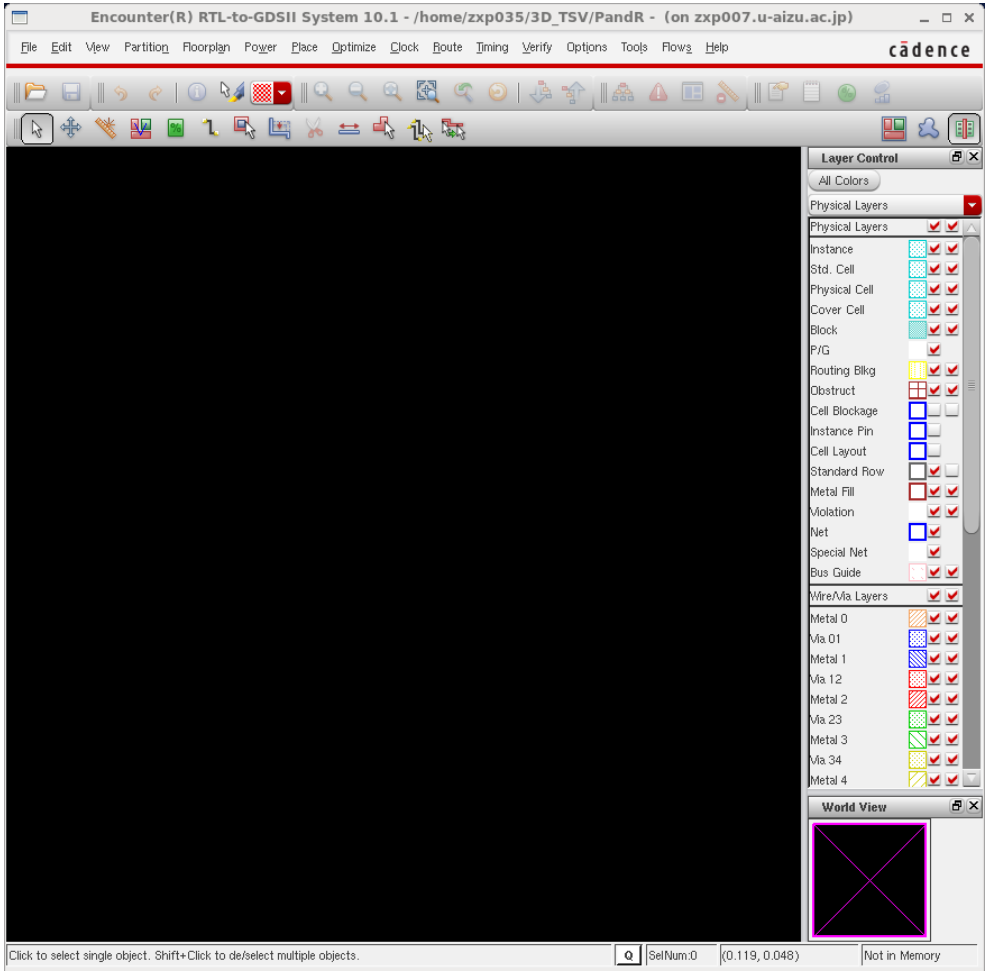
Change from **TSV_*** to **TSV**

Step2: Run SoC Encounter

Type **velocity** to start SoC Encounter

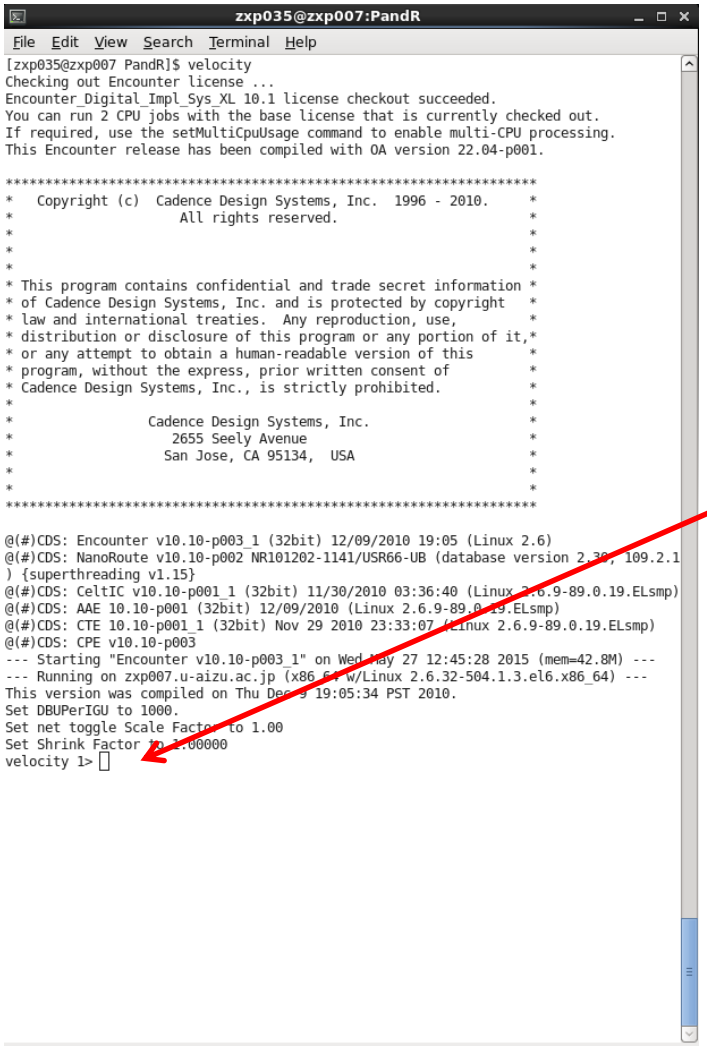
Step2: Run SoC Encounter





Welcome screen of SoC Encounter.
The CUI is on the left side and the GUI is in the right side.

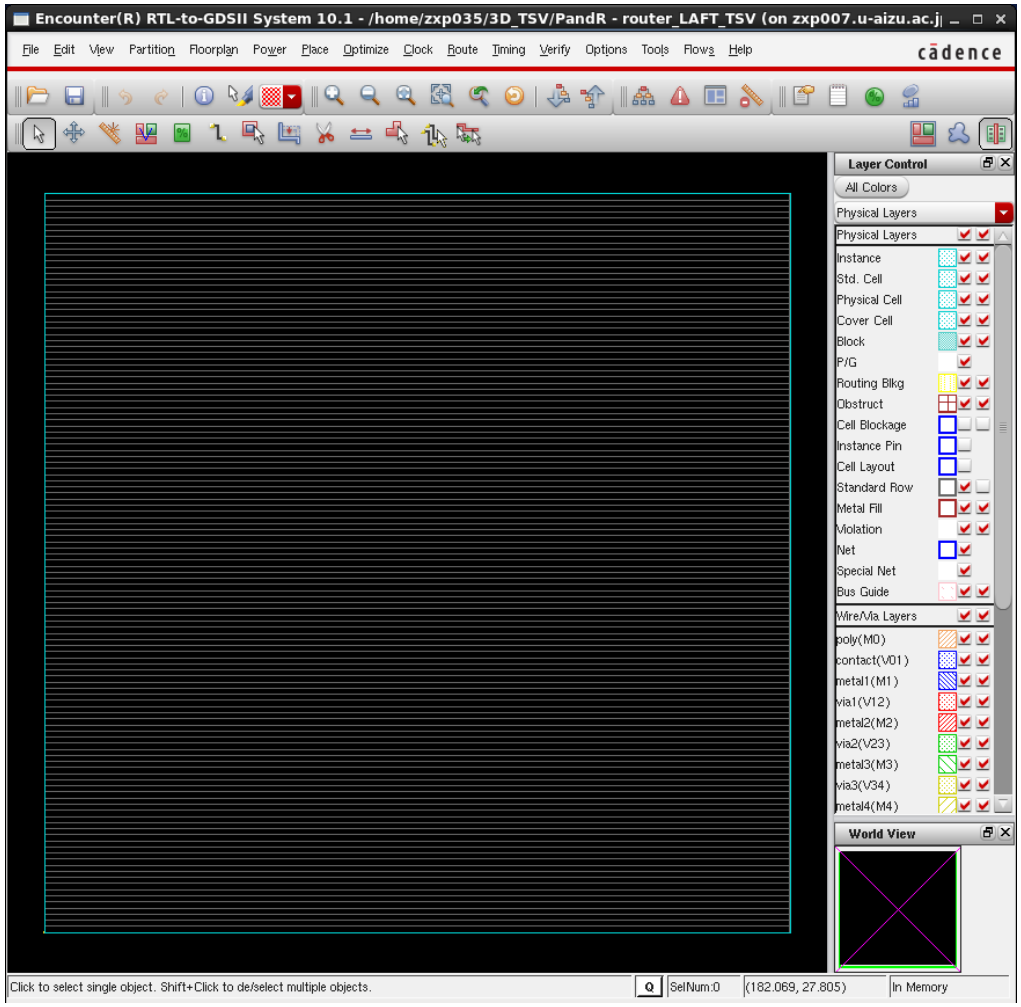
Step3: Import Design




```
#
# Step 1: Setup (File --> Import Design)
#
setUIVar rda_Input ui_netlist vnet/router_LAFT_TSV.vnet
setUIVar rda_Input ui_timingcon_file ../Syn/output_files/router_LAFT_TSV.sdc
setUIVar rda_Input ui_topcell router_LAFT_TSV
setUIVar rda_Input ui_leffile {macro/TSV.lef /home/zxp035/lib/NangateOpenCellLibrary.le
f}
setUIVar rda_Input ui_timelib {/home/zxp035/lib/typical.lib macro/TSV.lib}

setUIVar rda_Input ui_pwrnet VDD
setUIVar rda_Input ui_gndnet VSS
setUIVar rda_Input ui_cts_cell_list {CLKBUF_X1 CLKBUF_X2 CLKBUF_X3}
commitConfig
```

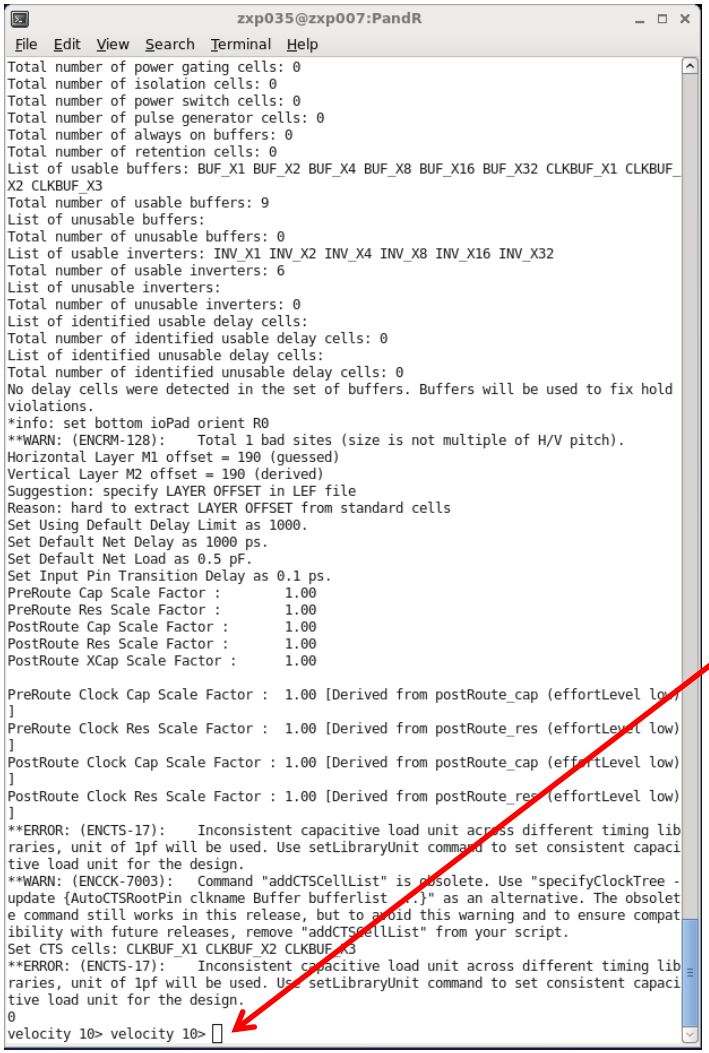
Execute the commands on the right hand in the CUI line by line to **import** the previously synthesized design

Step3: Import Design



The initial chip layout **after importing** the design

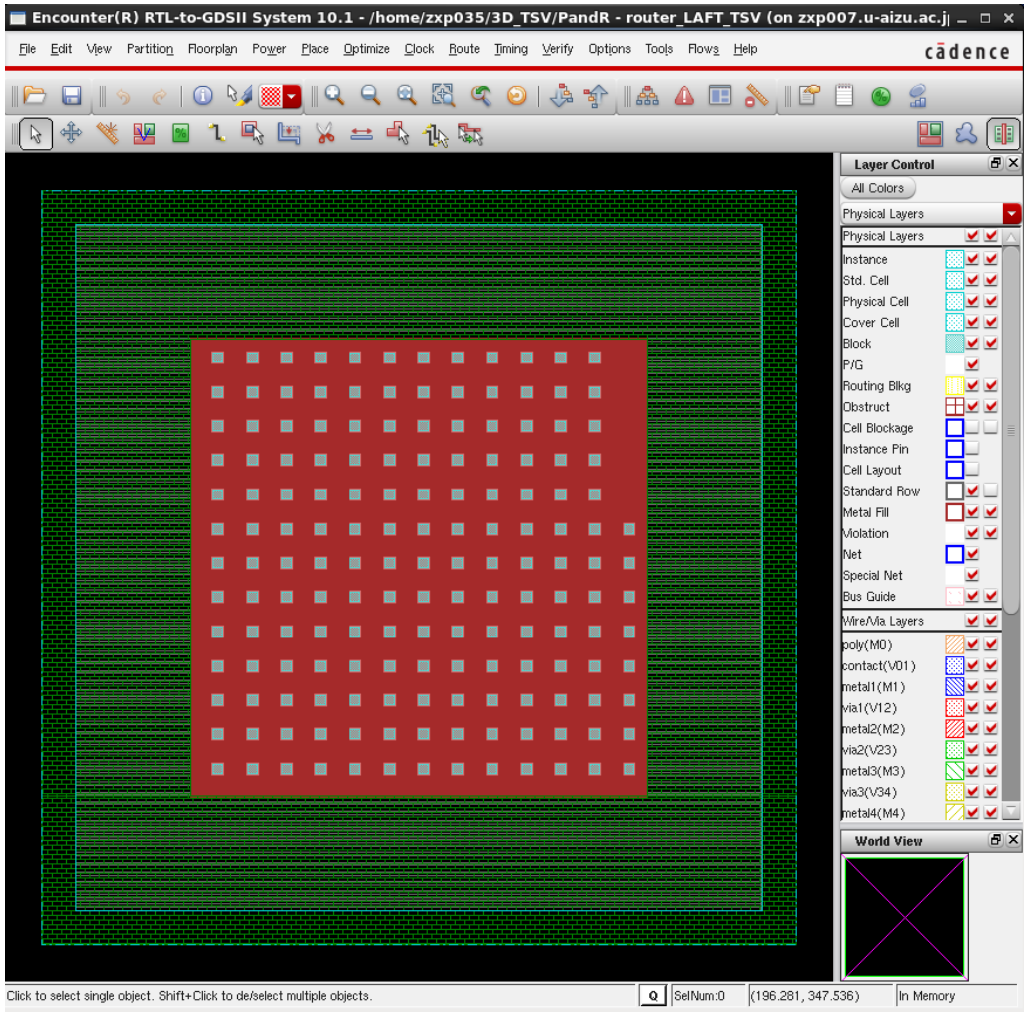
Step4: SoC Floorplan and macro




```
floorPlan -s 300 300 15 15 15 15
#set halo#
addHaloToBlock 0.5 0.5 0.5 0.5 -allBlock

createRouteBlk -box 0 0 380 65 -layer 11
createRouteBlk -box 0 65 65 265 -layer 11
createRouteBlk -box 265 65 380 265 -layer 11
createRouteBlk -box 0 265 380 380 -layer 11

#place macro
placeInstance tsv_input_up0 75 75 R0
placeInstance tsv_input_up1 75 90 R0

                    ⋮

placeInstance tsv_faulty_output_down4 255 120 R0
placeInstance tsv_faulty_input_up5 255 135 R0
placeInstance tsv_faulty_input_down5 255 150 R0
placeInstance tsv_faulty_output_up5 255 165 R0
placeInstance tsv_faulty_output_down5 255 180 R0

createObstruct 65 65 265 265
```

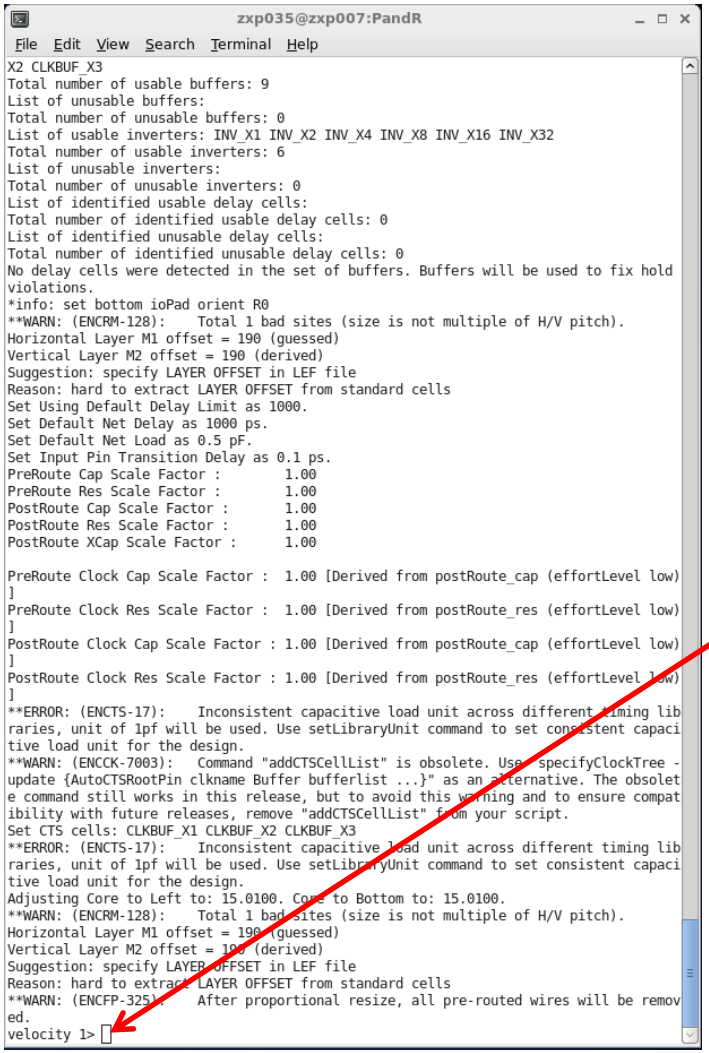
Execute the commands on the right hand in the CUI line by line to **set floorplan and macro locations**

Step4: SoC Floorplan and macro



Chip layout after **Floorplan and macro placement**

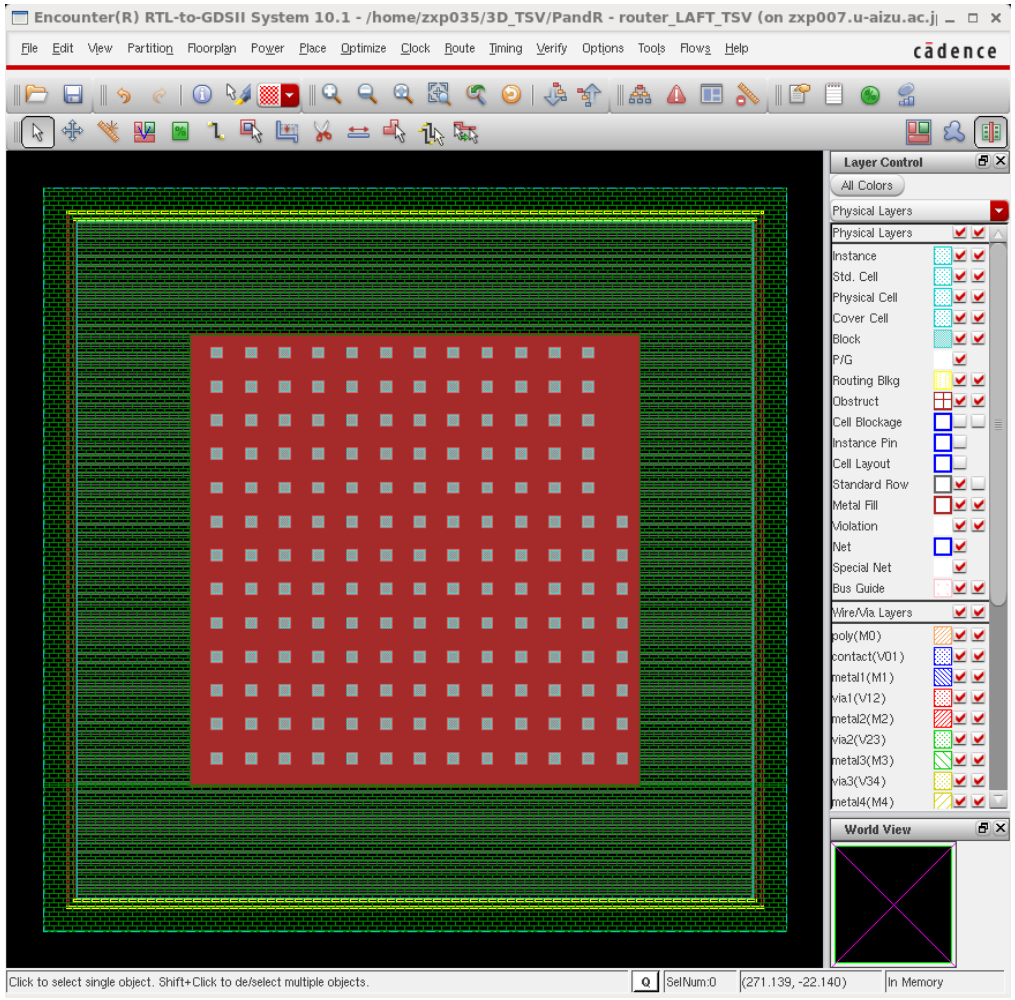
Step5: Power ring




```
#
# Step 3: Power ring (Power --> Power Planning --> Add Ring)
#
createPGPin VDD -net VDD
createPGPin VSS -net VSS

addRing -nets {VSS VDD} -type core_rings \
  -spacing_top 2 -spacing_bottom 2 -spacing_right 2 -spacing_left 2 \
  -width_top 1 -width_bottom 1 -width_right 1 -width_left 1 \
  -around core -jog_distance 0.095 -threshold 0.095 \
  -layer_top metal10 -layer_bottom metal10 -layer_right metal9 \
  -layer_left metal9 \
  -stacked_via_top_layer metal10 -stacked_via_bottom_layer metal1
```

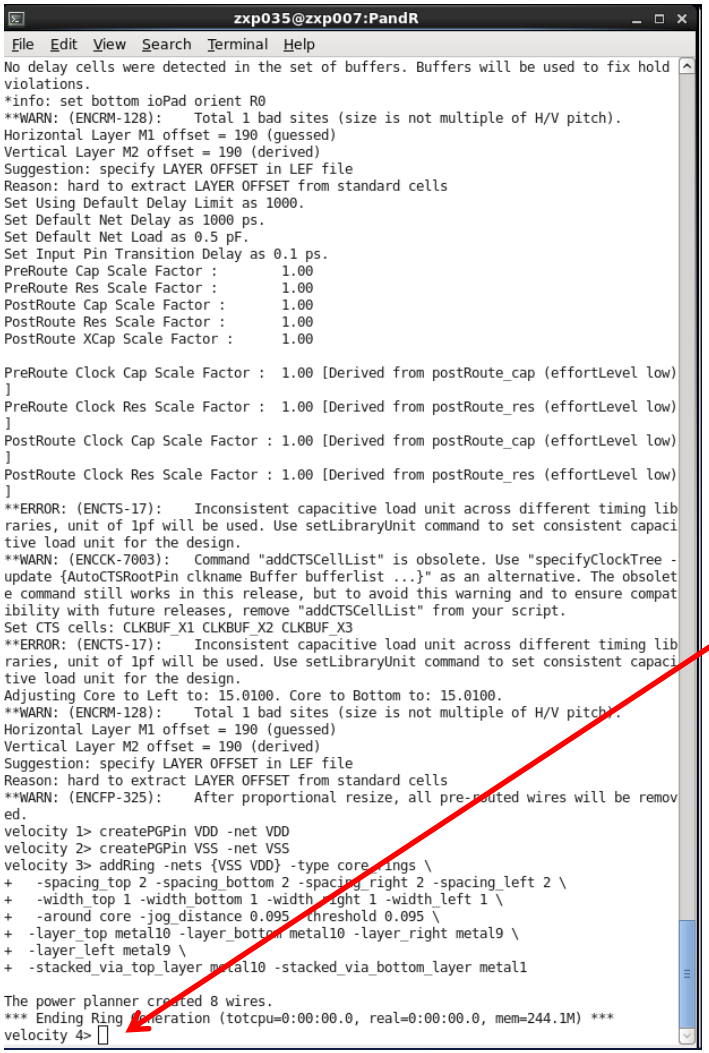
Execute the commands on the right hand in the CUI line by line to **add the power ring**

Step5: Power ring



Chip layout after adding **the power ring**

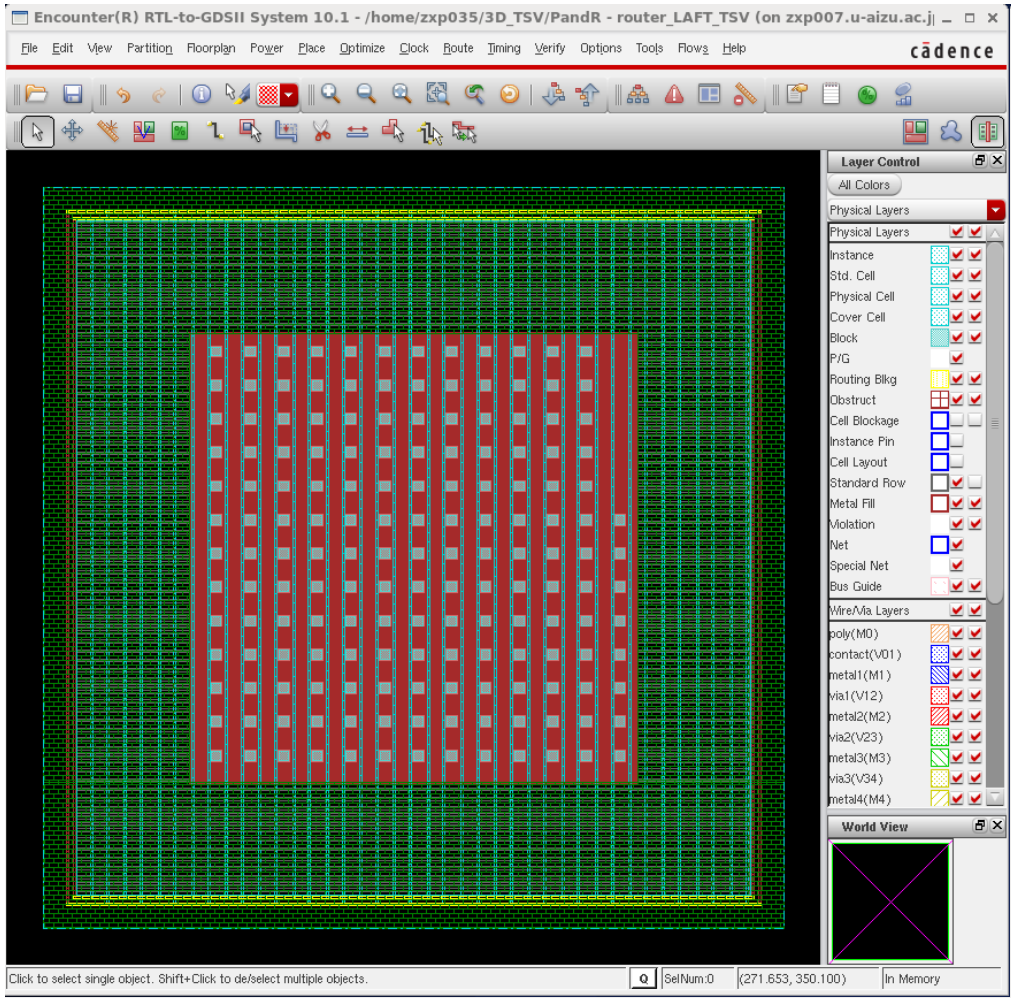
Step6: Power Stripe




```
#
# Step 4: Power stripe (Power --> Power Planning --> Add Striple)
#
addStripe -nets {VSS VDD} -layer metal8 -width 1 -spacing 6 \
  -block_ring_top_layer_limit metal9 -block_ring_bottom_layer_limit metal7 \
  -padcore_ring_top_layer_limit metal9 -padcore_ring_bottom_layer_limit metal7 \
  -stacked_via_top_layer metal10 -stacked_via_bottom_layer metal1 \
  -set_to_set_distance 15 -xleft_offset 6 -merge_stripes_value 0.095 \
  -max_same_layer_jog_length 1.6
```

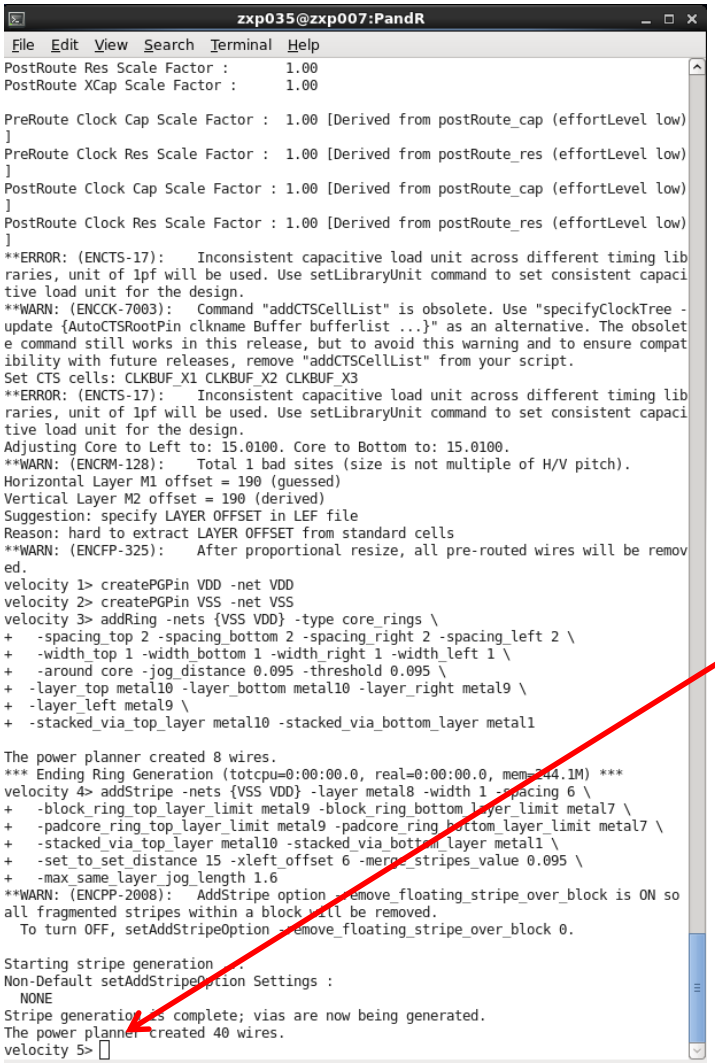
Execute the commands on the right hand in the CUI line by line to **add power stripe**

Step6: Power Stripe



Chip layout after adding **the power stripe**

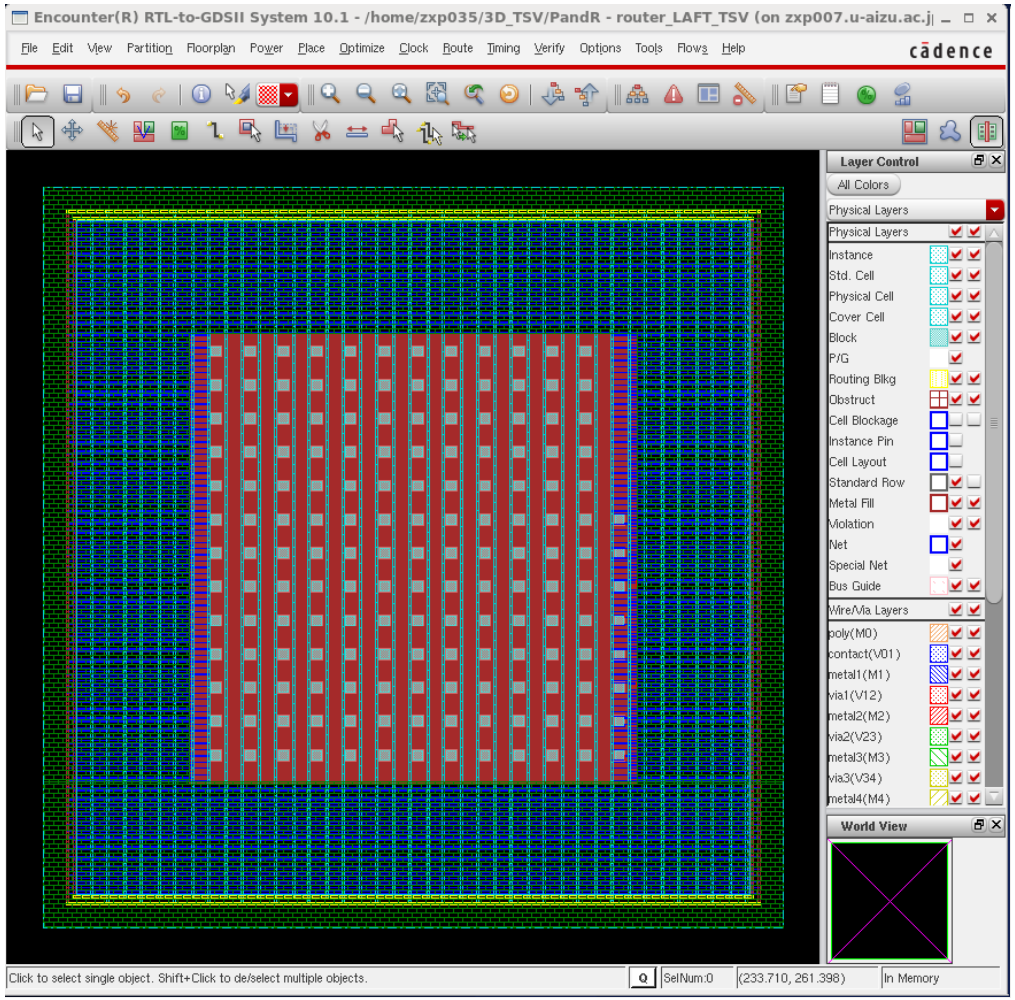
Step7: Power Routing




```
#
# Step 5: Power route (Route --> Special Router)
#

sroute -nets {VSS VDD} -layerChangeRange {1 10} \
  -connect { blockPin padPin padRing corePin floatingStripe } \
  -blockPinTarget { nearestRingStripe nearestTarget } \
  -padPinPortConnect { allPort oneGeom } \
  -checkAlignedSecondaryPin 1 -blockPin useLef -allowJogging 1 \
  -crossoverViaBottomLayer 1 -allowLayerChange 1 -targetViaTopLayer 10 \
  -crossoverViaTopLayer 10 -targetViaBottomLayer 1
```

Execute the commands on the right hand in the CUI line by line to **make special route**

Step7: Power Routing



Chip layout after making **the special route**

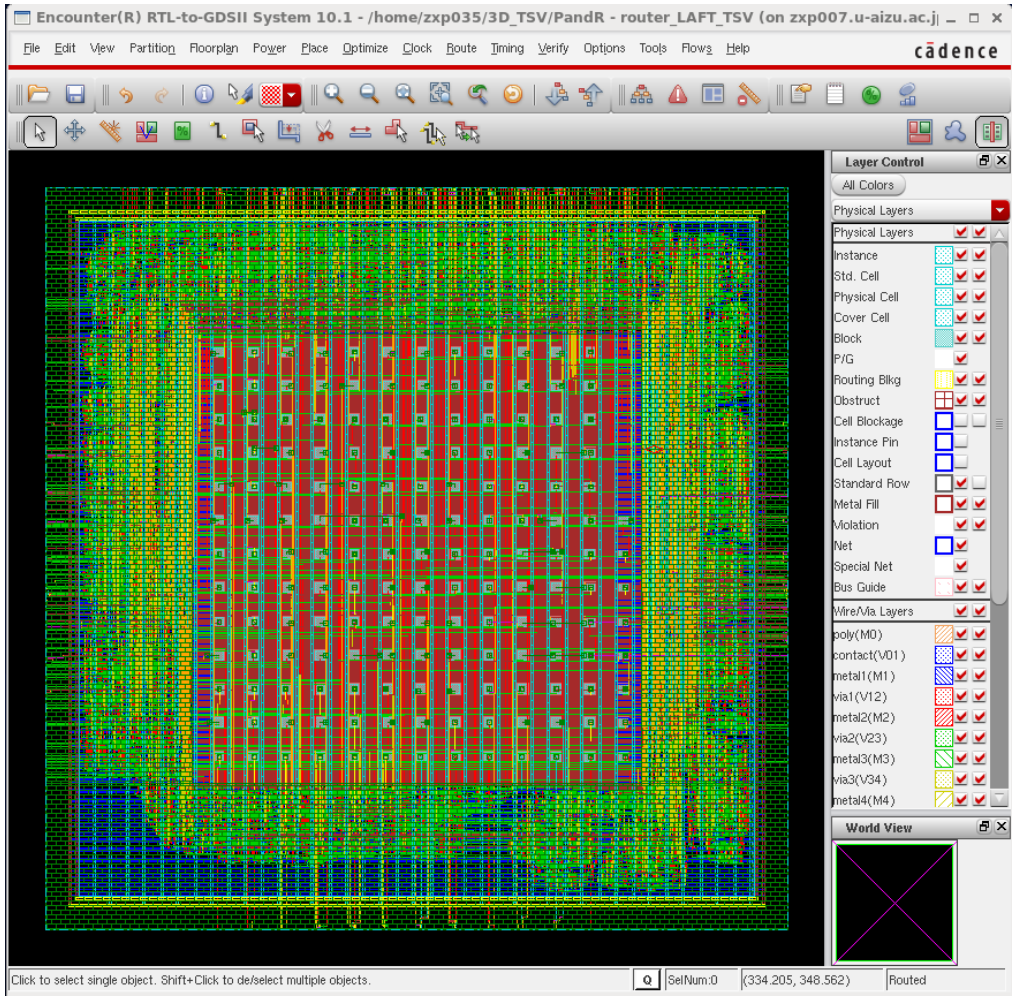
Step8: Placement

```
zxp035@zxp007:PandR
File  Edit  View  Search  Terminal  Help
**WARN: (ENCSR-468):    No core cells defined in COMPONENTS section
 and/or   No core cells defined in SPECIALNETS VSS
**WARN: (ENCSR-468):    No core cells defined in COMPONENTS section
 and/or   No core cells defined in SPECIALNETS VSS
CPU time for FollowPin 0 seconds
**WARN: LAYER via1, same net spacing is larger than different net spacing!
**WARN: LAYER via2, same net spacing is larger than different net spacing!
**WARN: LAYER via3, same net spacing is larger than different net spacing!
**WARN: LAYER via1, same net spacing is larger than different net spacing!
**WARN: LAYER via2, same net spacing is larger than different net spacing!
**WARN: LAYER via3, same net spacing is larger than different net spacing!
**WARN: LAYER via1, same net spacing is larger than different net spacing!
**WARN: LAYER via2, same net spacing is larger than different net spacing!
**WARN: LAYER via3, same net spacing is larger than different net spacing!
**WARN: LAYER via1, same net spacing is larger than different net spacing!
**WARN: LAYER via2, same net spacing is larger than different net spacing!
**WARN: LAYER via3, same net spacing is larger than different net spacing!
**WARN: LAYER via1, same net spacing is larger than different net spacing!
**WARN: LAYER via2, same net spacing is larger than different net spacing!
**WARN: LAYER via3, same net spacing is larger than different net spacing!
**WARN: LAYER via1, same net spacing is larger than different net spacing!
**WARN: LAYER via2, same net spacing is larger than different net spacing!
**WARN: LAYER via3, same net spacing is larger than different net spacing!
**WARN: LAYER via1, same net spacing is larger than different net spacing!
**WARN: LAYER via2, same net spacing is larger than different net spacing!
**WARN: LAYER via3, same net spacing is larger than different net spacing!
**WARN: LAYER via1, same net spacing is larger than different net spacing!
**WARN: LAYER via2, same net spacing is larger than different net spacing!
**WARN: LAYER via3, same net spacing is larger than different net spacing!
  Number of IO ports routed: 0
  Number of Block ports routed: 0
  Number of Stripe ports routed: 0
  Number of Core ports routed: 716
  Number of Pad ports routed: 0
  Number of Power Bump ports routed: 0
  Number of Followpin connections: 358
End power routing: cpu: 0:00:00, real: 0:00:00, peak: 493.00 megs.



 Begin updating DB with routing results ...
 Updating DB with 65 via definition ...Extracting standard cell pins and blockage ...
...
Pin and blockage extraction finished


sroute post-processing starts at Wed May 27 13:06:53 2015
The viaGen is rebuilding shadow vias for net VSS.
sroute post-processing ends at Wed May 27 13:06:53 2015

sroute post-processing starts at Wed May 27 13:06:53 2015
The viaGen is rebuilding shadow vias for net VDD.
sroute post-processing ends at Wed May 27 13:06:53 2015
sroute: Total CPU time used = 0:0:0
sroute: Total Real time used = 0:0:0
sroute: Total Memory used = 6.36 megs
sroute: Total Peak Memory used = 250.96 megs
velocity 6>
```

```
#
# Step 6: Placement (Place --> Standard Cell)
#
placeDesign -prePlaceOpt
```

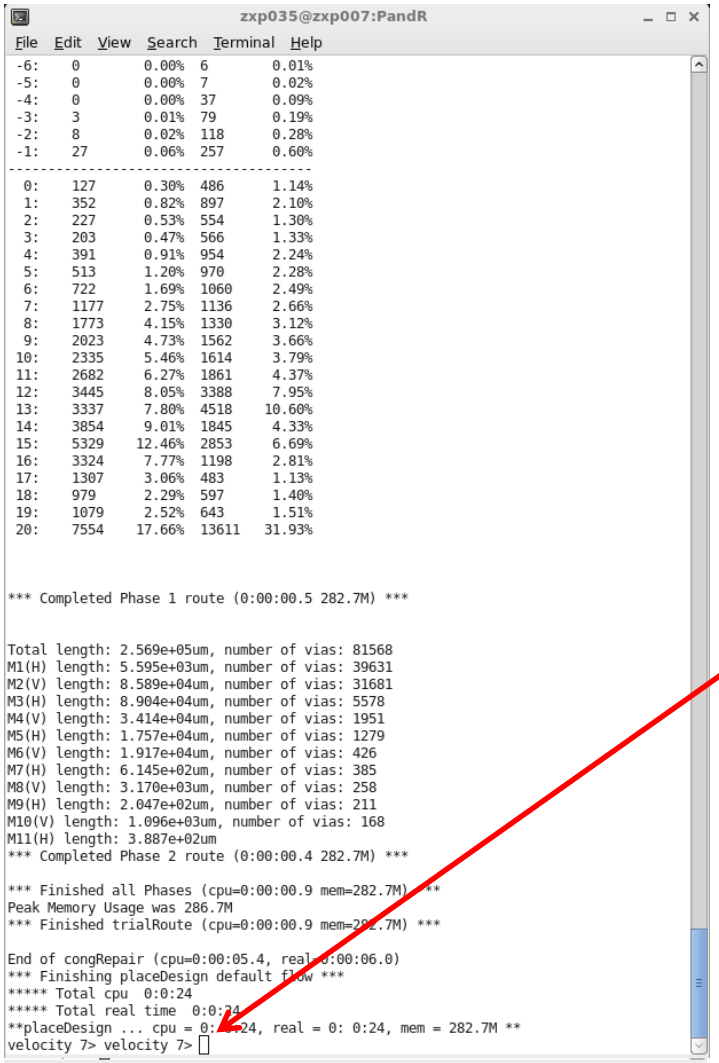
Execute the commands on the right hand in the CUI line by line to **place the standard cells**

Step8: Placement



Chip layout after **placing the standard cells**

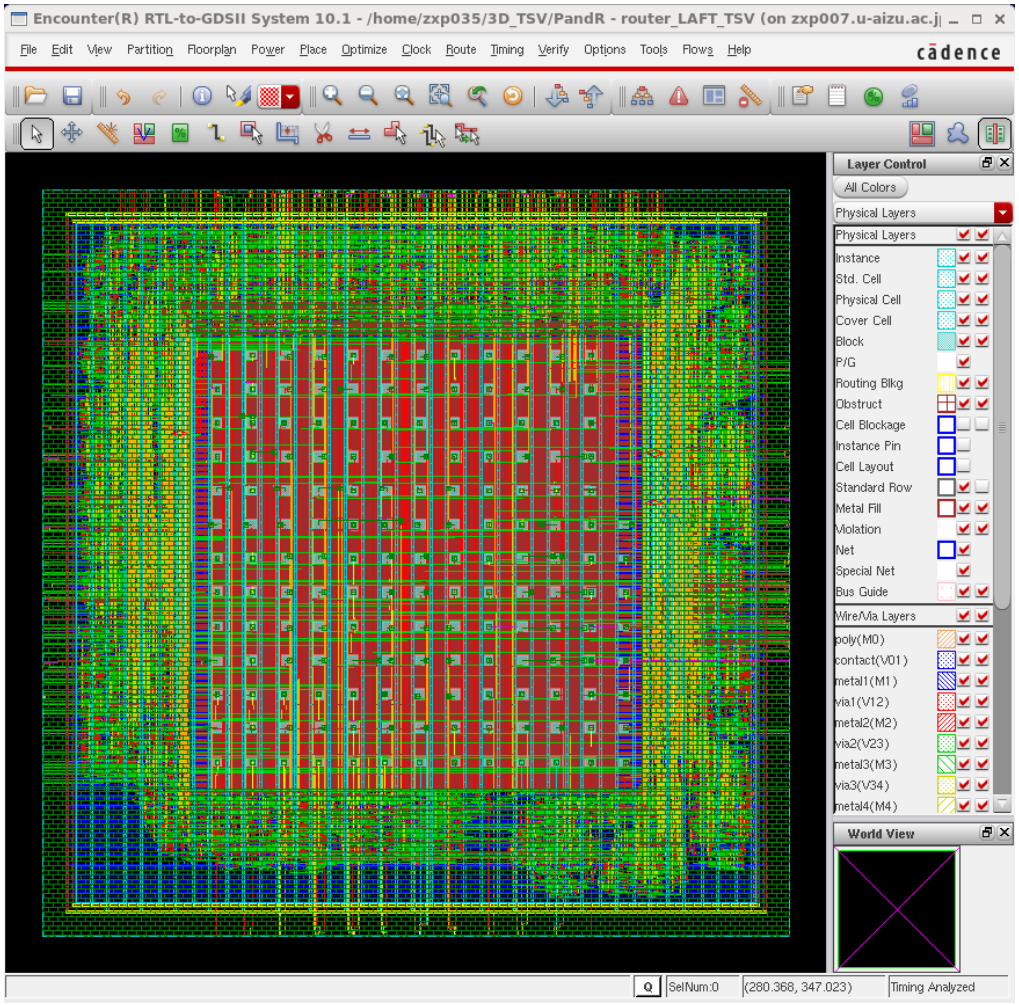
Step9: Optimization




```
#
# Step 7: Optimization (preCTS) (Optimize --> Optimize Design)
#
optDesign -preCTS
```

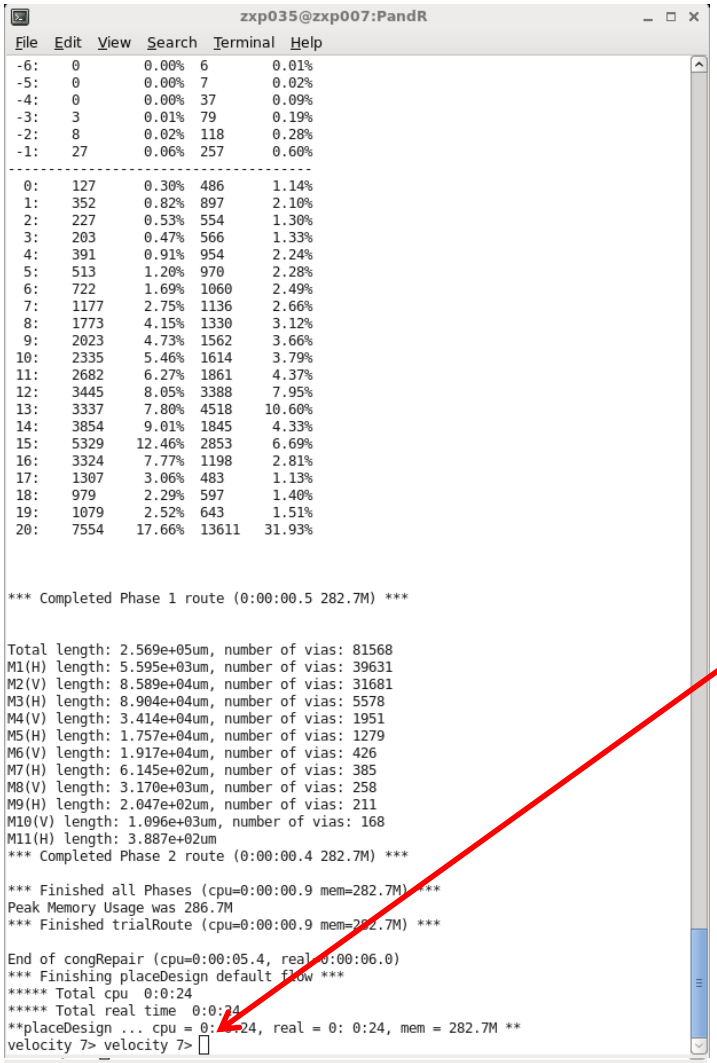
Execute the commands on the right hand in the CUI line by line to perform the **Pre-clock synthesis optimization**

Step9: Optimization



Chip layout after the **Pre-clock synthesis optimization**

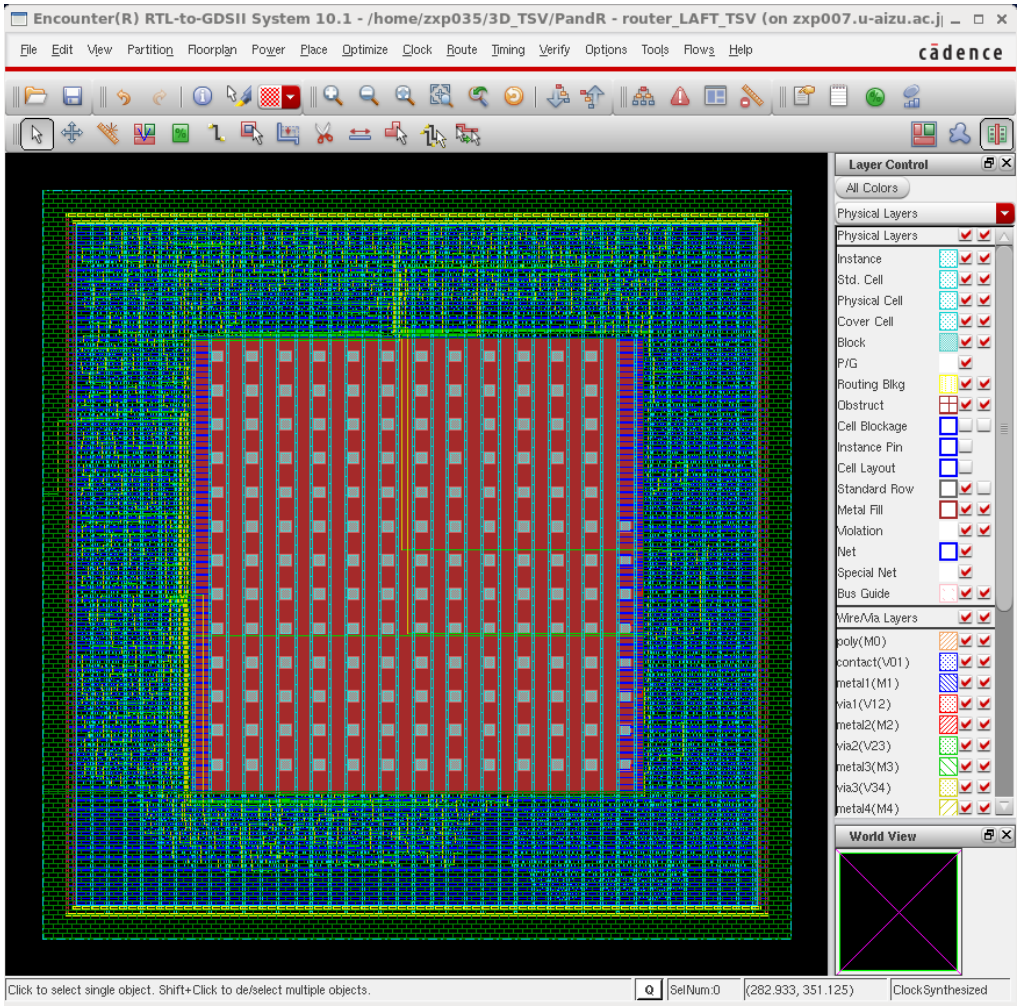
Step10: Clock Tree




```
#
# Step 8: Clock tree synthesis (CTS) (Clock --> Cynthesize Clock Tree)
#
addCTSCellList {CLKBUF_X1 CLKBUF_X2 CLKBUF_X3}
clockDesign -genSpecOnly Clock.ctstch
clockDesign -specFile Clock.ctstch -outDir clock_report -fixedInstBeforeCTS
```

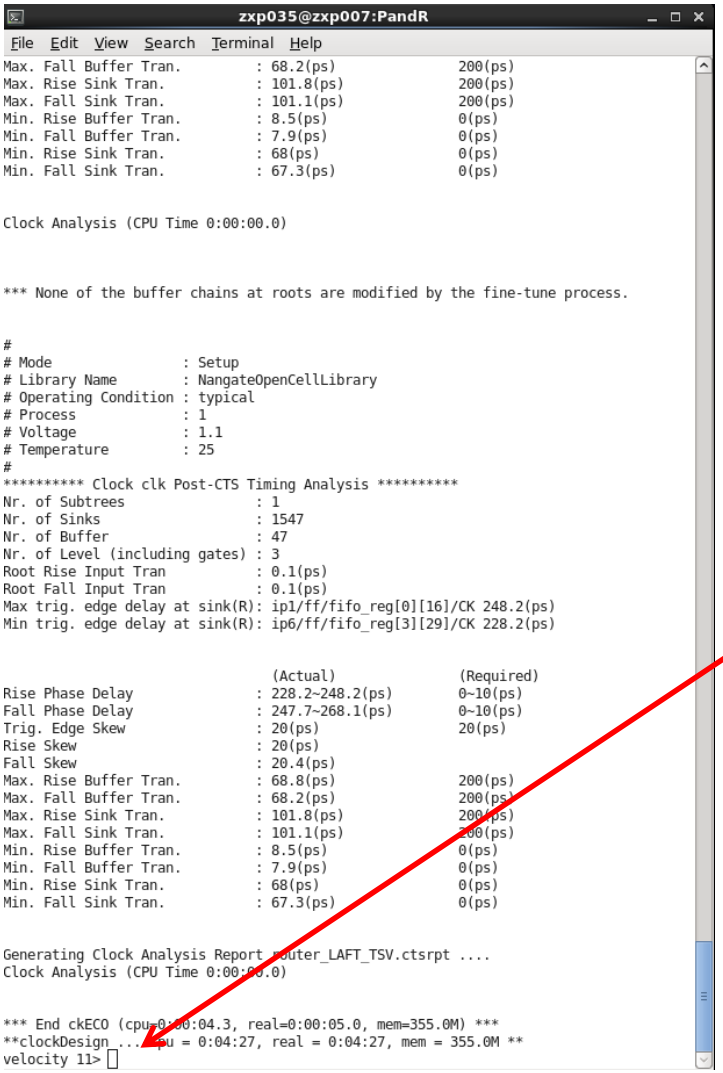
Execute the commands on the right hand in the CUI line by line to **synthesize clock tree**

Step10: Clock Tree



Chip layout after **synthesizing the clock tree**

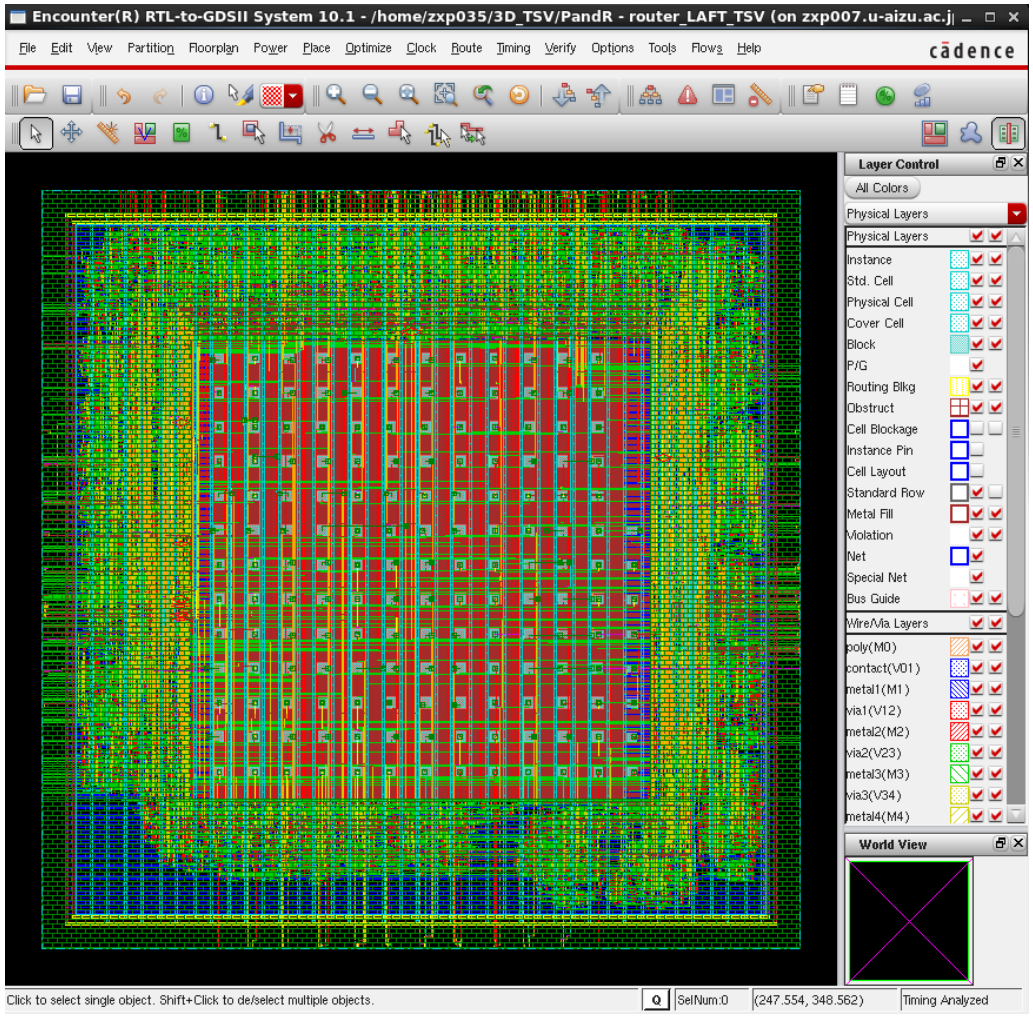
Step11: Optimization




```
#
# Step 9: Optimization (postCTS) (Optimize --> Optimize Design)
#
optDesign -postCTS
optDesign -postCTS -hold
```

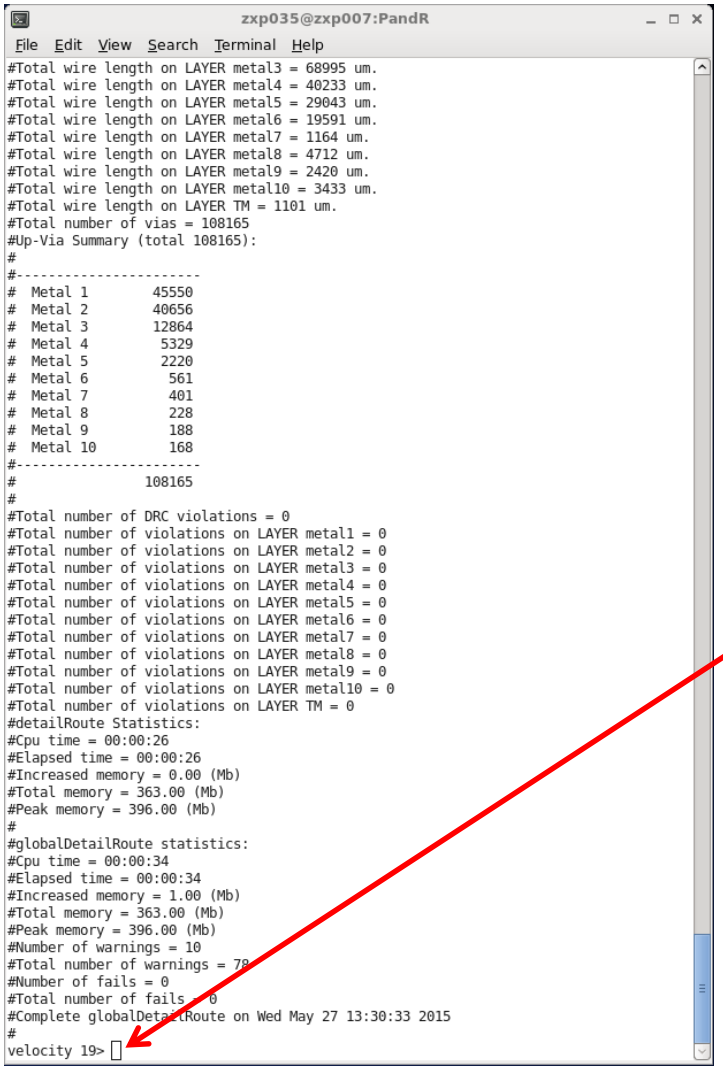
Execute the commands on the right hand in the CUI line by line to perform the **Post-clock synthesis optimization**

Step11: Optimization



Chip layout after **Post-clock synthesis optimization**

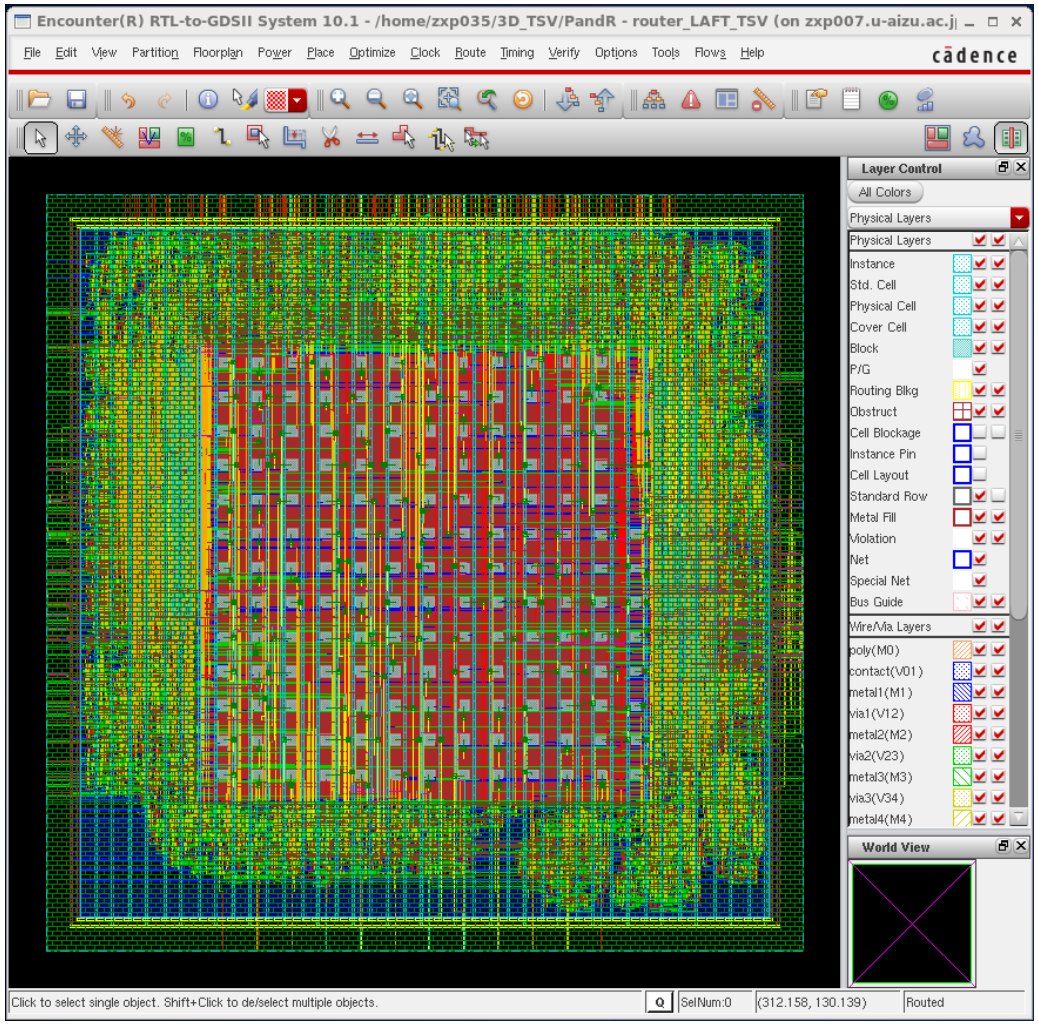
Step12: Nano Route




```
#
# Step 10: Detailed route (Route --> Nano Route --> Route)
#
setNanoRouteMode -quiet -routeWithTimingDriven true
setNanoRouteMode -quiet -routeTopRoutingLayer default
setNanoRouteMode -quiet -routeBottomRoutingLayer default
setNanoRouteMode -quiet -drouteEndIteration default
setNanoRouteMode -quiet -routeWithTimingDriven true
routeDesign -globalDetail
```

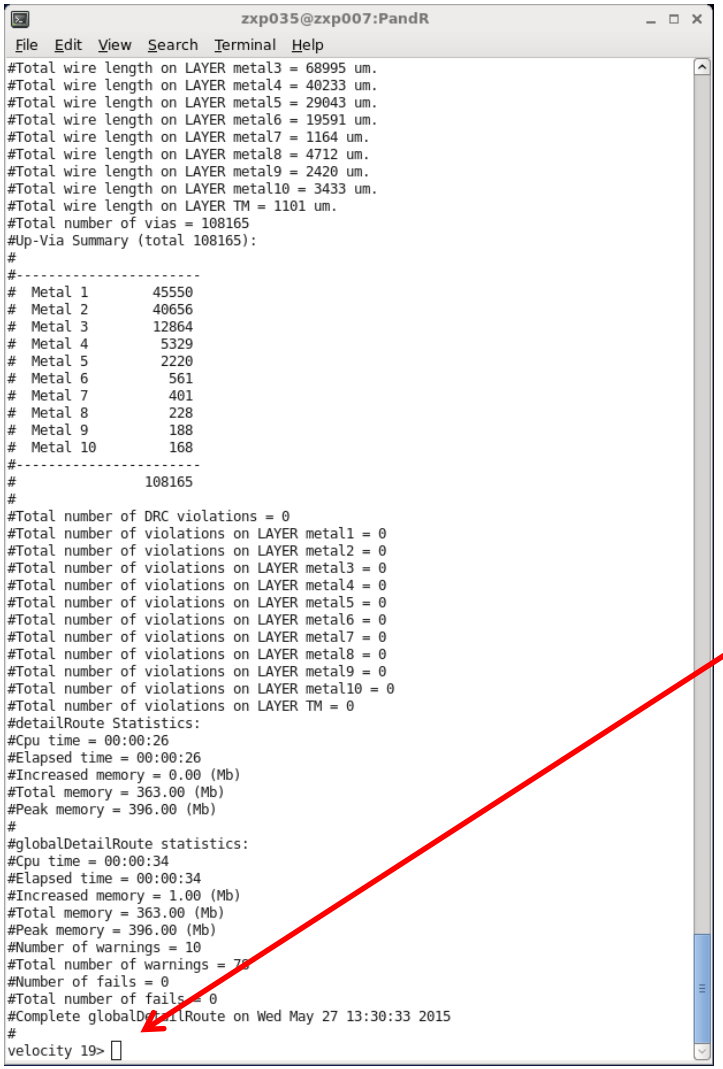
Execute the commands on the right hand in the CUI line by line to **add Nano route**

Step12: Nano Route



Chip layout after **Nano-route**

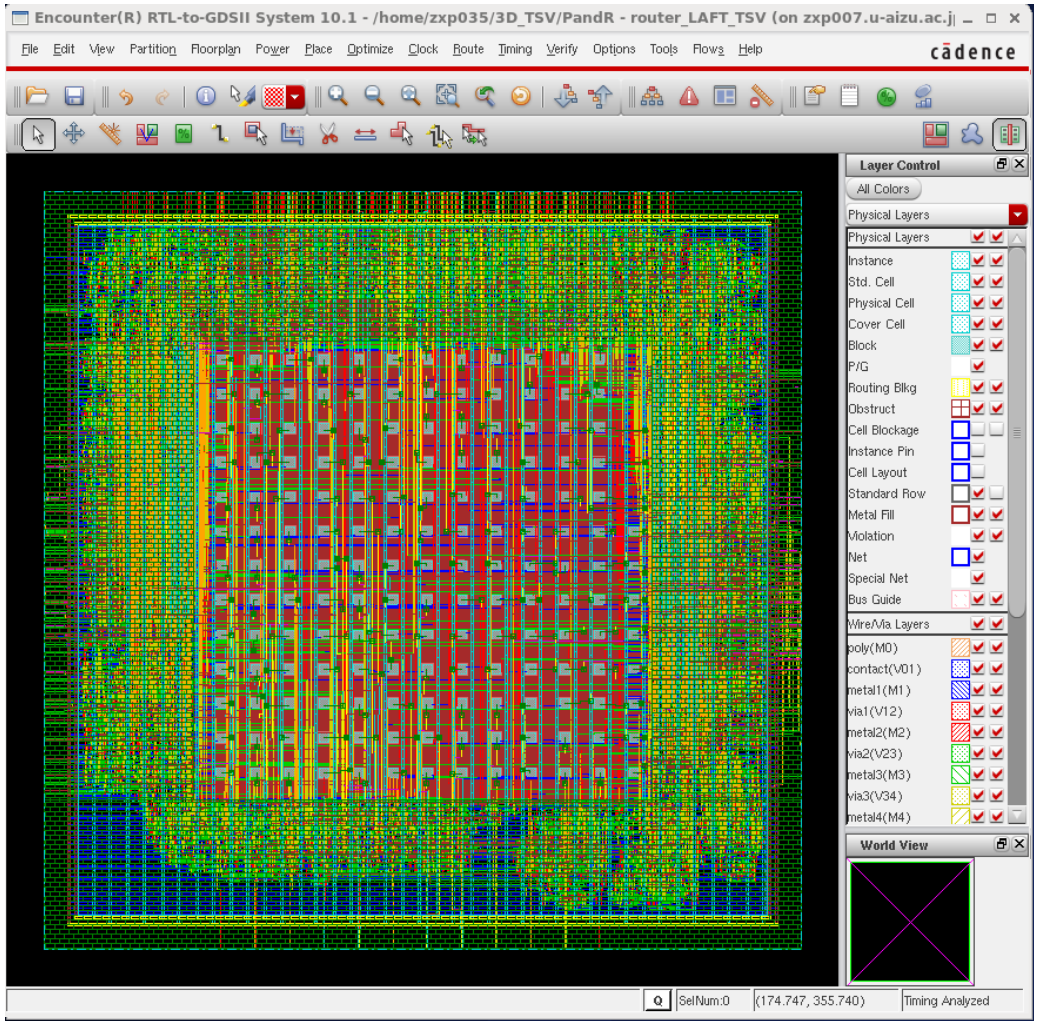
Step13: Optimization




```
#
# Step 11: Optimization (postRoute) (Optimize --> Optimize Design)
#
 optDesign -postRoute
 optDesign -postRoute -hold
```

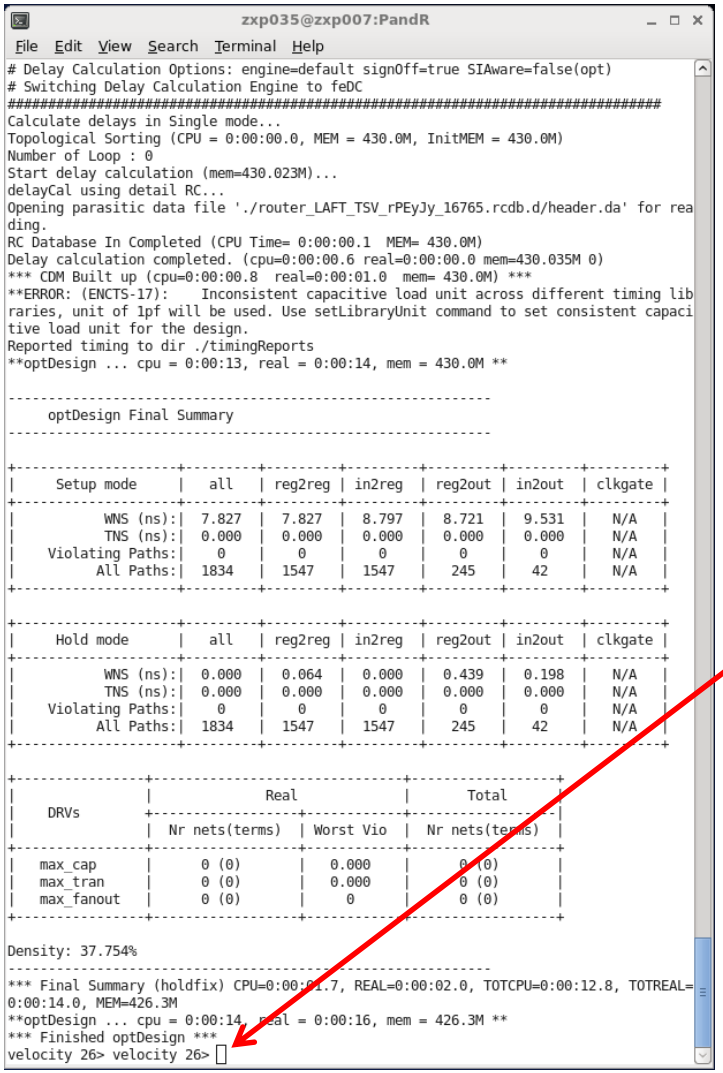
Execute the commands on the right hand in the CUI line by line to perform the **Post-route optimization**

Step13: Optimization



Chip layout after **Post-route optimization**

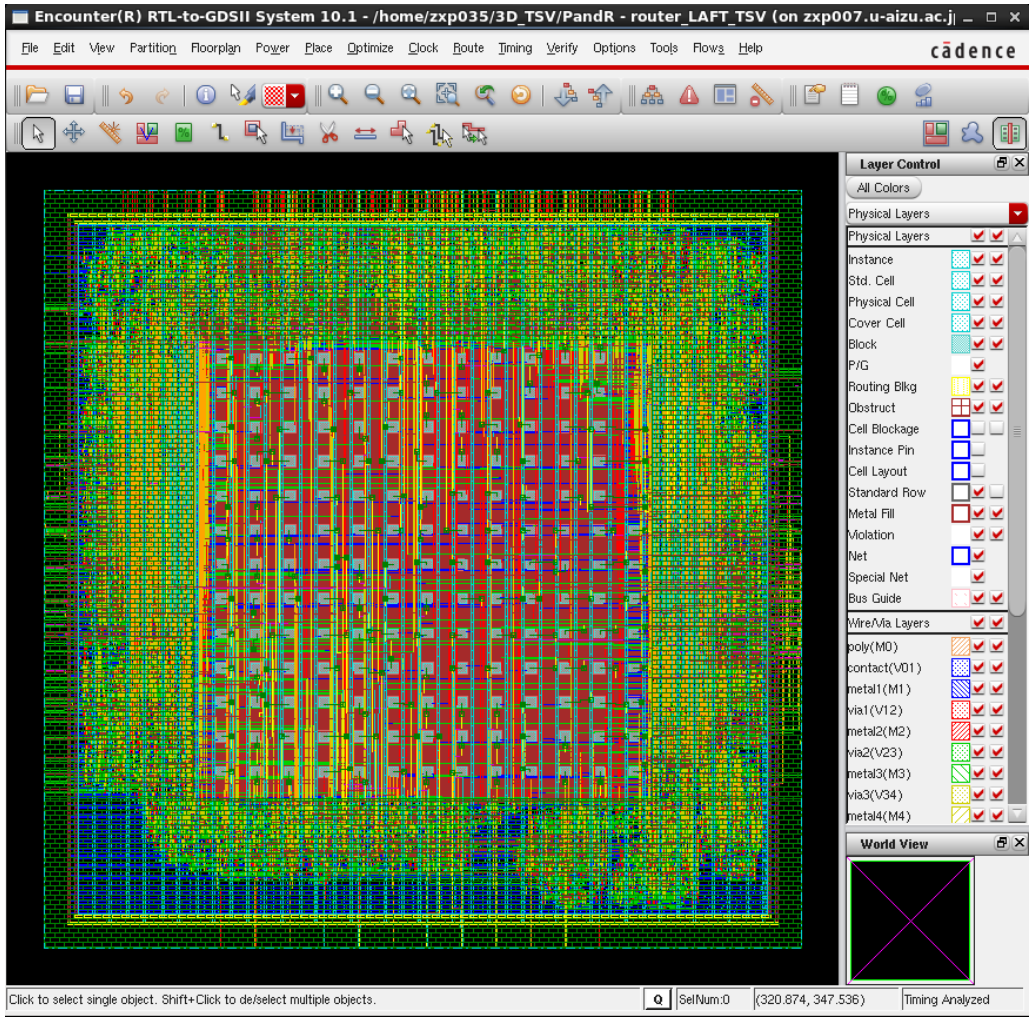
Step14: Adding Fillers




```
#
# Step 12: Add fillers (Place --> Physical Cells --> Add Filler)
#
 addFiller -prefix FILLER -cell FILLCELL_X1 FILLCELL_X2 FILLCELL_X4 \
   FILLCELL_X8 FILLCELL_X16 FILLCELL_X32
```

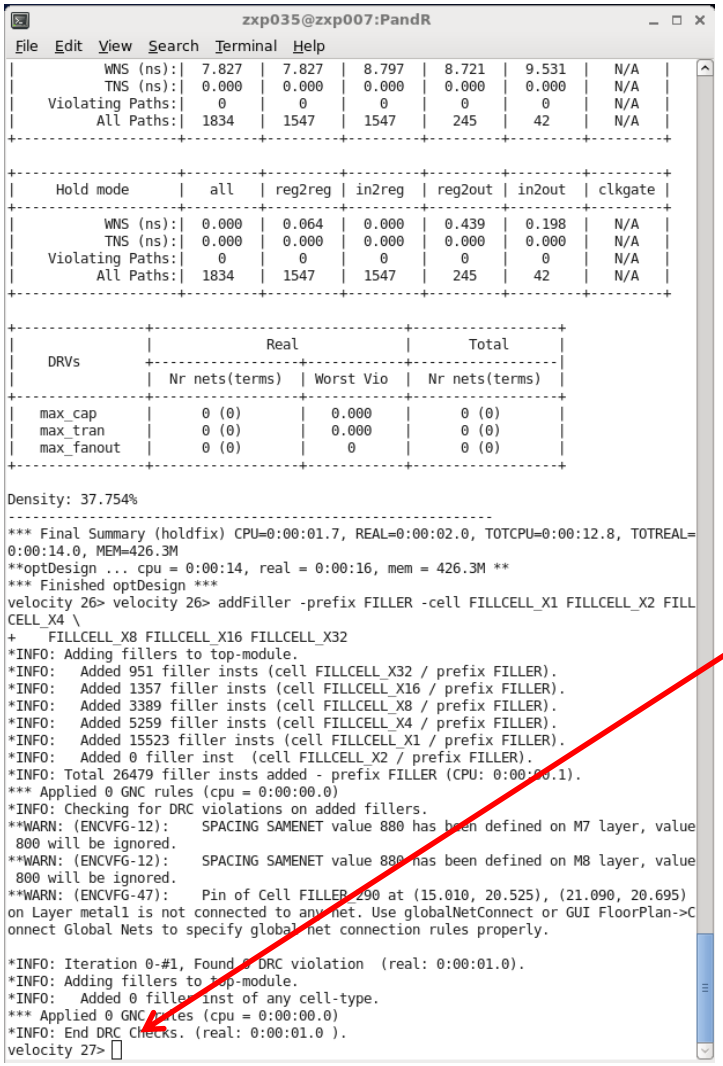
Execute the commands on the right hand in the CUI line by line to **add fillers**

Step14: Adding Fillers



Chip layout after **adding fillers**

Step15: Layout Vs. Schematic(LVS)




```
#
# Step 13: Verification (LVS) (Verify --> Verify Connectivity)
#
 verifyConnectivity -type all -error 1000 -warning 50
```

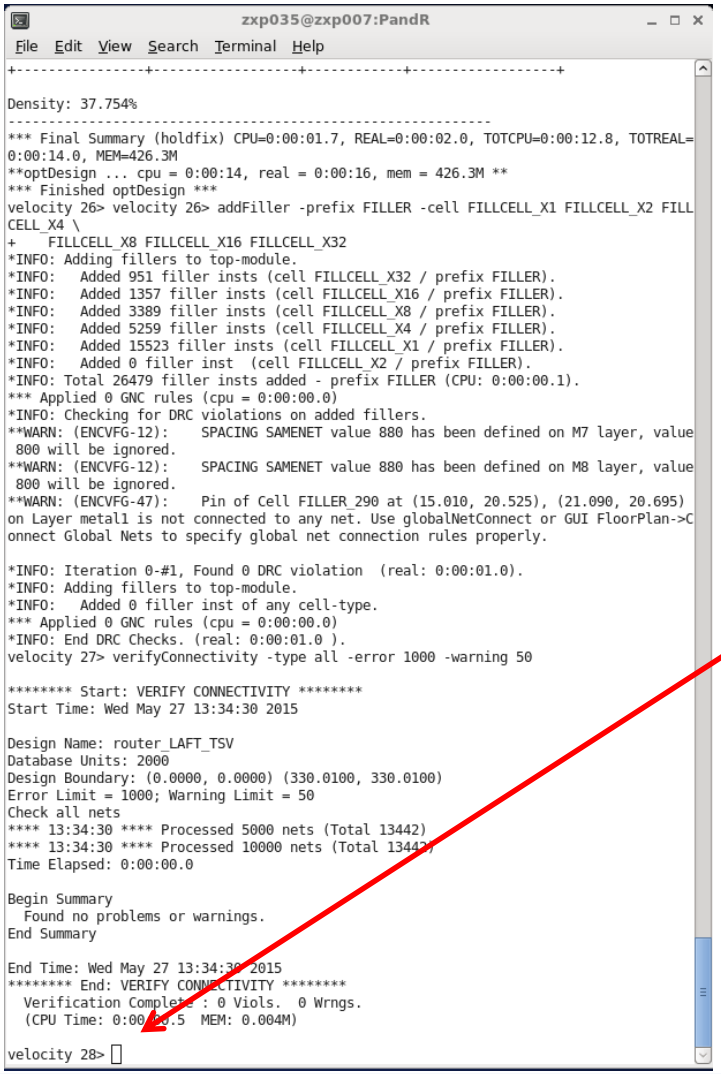
Execute the commands on the right hand in the CUI line by line to perform the **LVS check**

Step15: Layout Vs. Schematic(LVS)

```
******** Start: VERIFY CONNECTIVITY ********
Start Time: Wed May 27 13:34:30 2015

Design Name: router_LAFT_TSV
Database Units: 2000
Design Boundary: (0.0000, 0.0000) (330.0100, 330.0100)
Error Limit = 1000; Warning Limit = 50
Check all nets
**** 13:34:30 **** Processed 5000 nets (Total 13442)
**** 13:34:30 **** Processed 10000 nets (Total 13442)
Time Elapsed: 0:00:00.0

Begin Summary
  Found no problems or warnings.
End Summary

End Time: Wed May 27 13:34:30 2015
******** End: VERIFY CONNECTIVITY ********
  Verification Complete : 0 Viols.  0 Wrngs.
  (CPU Time: 0:00:00.5  MEM: 0.004M)
```

LVS check report on the CUI
Successful check (No errors and no warning)

Step16: Design Rule Check(DRC)




```
#
# Step 14: Verification (DRC) (Verify --> Verify Geometry)
#
 verifyGeometry
```

Execute the commands on the right hand in the CUI line by line to perform the **DRC check**

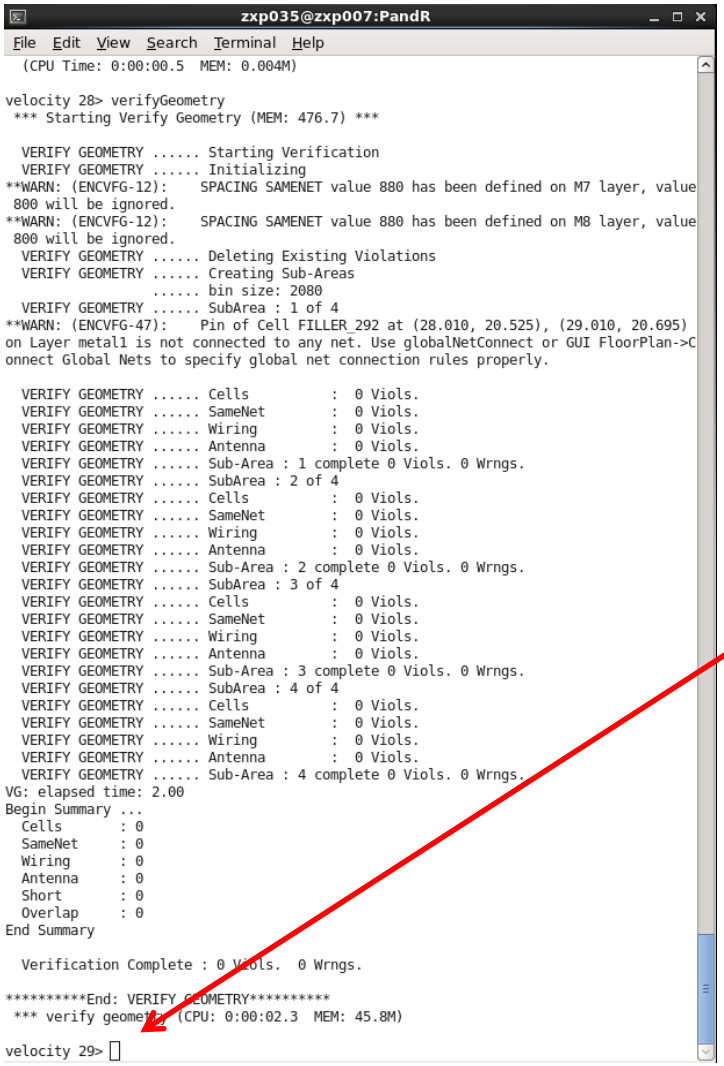
Step16: Design Rule Check(DRC)

```
  VERIFY GEOMETRY ..... Cells          :  0 Viols.
  VERIFY GEOMETRY ..... SameNet        :  0 Viols.
  VERIFY GEOMETRY ..... Wiring         :  0 Viols.
  VERIFY GEOMETRY ..... Antenna        :  0 Viols.
  VERIFY GEOMETRY ..... Sub-Area : 1 complete 0 Viols. 0 Wrngs.
  VERIFY GEOMETRY ..... SubArea : 2 of 4
  VERIFY GEOMETRY ..... Cells          :  0 Viols.
  VERIFY GEOMETRY ..... SameNet        :  0 Viols.
  VERIFY GEOMETRY ..... Wiring         :  0 Viols.
  VERIFY GEOMETRY ..... Antenna        :  0 Viols.
  VERIFY GEOMETRY ..... Sub-Area : 2 complete 0 Viols. 0 Wrngs.
  VERIFY GEOMETRY ..... SubArea : 3 of 4
  VERIFY GEOMETRY ..... Cells          :  0 Viols.
  VERIFY GEOMETRY ..... SameNet        :  0 Viols.
  VERIFY GEOMETRY ..... Wiring         :  0 Viols.
  VERIFY GEOMETRY ..... Antenna        :  0 Viols.
  VERIFY GEOMETRY ..... Sub-Area : 3 complete 0 Viols. 0 Wrngs.
  VERIFY GEOMETRY ..... SubArea : 4 of 4
  VERIFY GEOMETRY ..... Cells          :  0 Viols.
  VERIFY GEOMETRY ..... SameNet        :  0 Viols.
  VERIFY GEOMETRY ..... Wiring         :  0 Viols.
  VERIFY GEOMETRY ..... Antenna        :  0 Viols.
  VERIFY GEOMETRY ..... Sub-Area : 4 complete 0 Viols. 0 Wrngs.
VG: elapsed time: 2.00
Begin Summary ...
  Cells       : 0
  SameNet     : 0
  Wiring      : 0
  Antenna     : 0
  Short       : 0
  Overlap     : 0
End Summary

  Verification Complete : 0 Viols.  0 Wrngs.
```

DRC check report on the CUI

Successful check (No violations and no warning)

Step17: Output files




```
#
# Step 15: Data out (Timing --> Extract RC, Timing --> Write SDF,
#                    File --> Save --> Netlist)
saveNetlist output_files/router_LAFT_macroTSV_final.vnet
isExtractRCModeSignoff
rcOut -spef output_files/router_LAFT_macroTSV.spef
delayCal -sdf output_files/router_LAFT_macroTSV.sdf -idealclock

saveDesign router_LAFT_macroTSV_final.enc
```

Execute the commands on the right hand in the CUI line by line to generate the **output files** and **finish** P&R

PandR.tcl script (1/5)

```
#
# Step 1: Setup (File --> Import Design)
#
setUIVar rda_Input ui_netlist vnet/router_LAFT_TSV.vnet
setUIVar rda_Input ui_timingcon_file ../Syn/output_files/router_LAFT_TSV.sdc
setUIVar rda_Input ui_topcell router_LAFT_TSV
setUIVar rda_Input ui_leffile {macro/TSV.lef /home/zxp035/lib/NangateOpenCellLibrary.le
f}
setUIVar rda_Input ui_timelib {/home/zxp035/lib/typical.lib macro/TSV.lib}

setUIVar rda_Input ui_pwrnet VDD
setUIVar rda_Input ui_gndnet VSS
setUIVar rda_Input ui_cts_cell_list {CLKBUF_X1 CLKBUF_X2 CLKBUF_X3}
commitConfig
win

floorPlan -s 300 300 15 15 15 15
#set halo#
addHaloToBlock 0.5 0.5 0.5 0.5 -allBlock

createRouteBlk -box 0 0 380 65 -layer 11
createRouteBlk -box 0 65 65 265 -layer 11
createRouteBlk -box 265 65 380 265 -layer 11
createRouteBlk -box 0 265 380 380 -layer 11

#place macro
placeInstance tsv_input_up0 75 75 R0
placeInstance tsv_input_up1 75 90 R0
placeInstance tsv_input_up2 75 105 R0
placeInstance tsv_input_up3 75 120 R0
placeInstance tsv_input_up4 75 135 R0
placeInstance tsv_input_up5 75 150 R0
placeInstance tsv_input_up6 75 165 R0
placeInstance tsv_input_up7 75 180 R0
placeInstance tsv_input_up8 75 195 R0
placeInstance tsv_input_up9 75 210 R0
placeInstance tsv_input_up10 75 225 R0
placeInstance tsv_input_up11 75 240 R0
placeInstance tsv_input_up12 75 255 R0
```

PandR.tcl script (2/5)

```
placeInstance tsv_faulty_output_up0 225 255 R0
placeInstance tsv_faulty_output_down0 240 75 R0
placeInstance tsv_faulty_input_up1 240 90 R0
placeInstance tsv_faulty_input_down1 240 105 R0
placeInstance tsv_faulty_output_up1 240 120 R0
placeInstance tsv_faulty_output_down1 240 135 R0
placeInstance tsv_faulty_input_up2 240 150 R0
placeInstance tsv_faulty_input_down2 240 165 R0
placeInstance tsv_faulty_output_up2 240 180 R0
placeInstance tsv_faulty_output_down2 240 195 R0
placeInstance tsv_faulty_input_up3 240 210 R0
placeInstance tsv_faulty_input_down3 240 225 R0
placeInstance tsv_faulty_output_up3 240 240 R0
placeInstance tsv_faulty_output_down3 240 255 R0

placeInstance tsv_faulty_input_up4 255 75 R0
placeInstance tsv_faulty_input_down4 255 90 R0
placeInstance tsv_faulty_output_up4 255 105 R0
placeInstance tsv_faulty_output_down4 255 120 R0
placeInstance tsv_faulty_input_up5 255 135 R0
placeInstance tsv_faulty_input_down5 255 150 R0
placeInstance tsv_faulty_output_up5 255 165 R0
placeInstance tsv_faulty_output_down5 255 180 R0

createObstruct 65 65 265 265

# Place your hard-macro manually

#saveDesign floor.enc

#
# Step 3: Power ring (Power --> Power Planning --> Add Ring)
#
createPGPin VDD -net VDD
createPGPin VSS -net VSS
```

PandR.tcl script (3/5)

```
addRing -nets {VSS VDD} -type core_rings \
  -spacing_top 2 -spacing_bottom 2 -spacing_right 2 -spacing_left 2 \
  -width_top 1 -width_bottom 1 -width_right 1 -width_left 1 \
  -around core -jog_distance 0.095 -threshold 0.095 \
  -layer_top metal10 -layer_bottom metal10 -layer_right metal9 \
  -layer_left metal9 \
  -stacked_via_top_layer metal10 -stacked_via_bottom_layer metal1

#
# Step 4: Power stripe (Power --> Power Planning --> Add Striple)
#
addStripe -nets {VSS VDD} -layer metal8 -width 1 -spacing 6 \
  -block_ring_top_layer_limit metal9 -block_ring_bottom_layer_limit metal7 \
  -padcore_ring_top_layer_limit metal9 -padcore_ring_bottom_layer_limit metal7 \
  -stacked_via_top_layer metal10 -stacked_via_bottom_layer metal1 \
  -set_to_set_distance 15 -xleft_offset 6 -merge_stripes_value 0.095 \
  -max_same_layer_jog_length 1.6

#
# Step 5: Power route (Route --> Special Router)
#

sroute -nets {VSS VDD} -layerChangeRange {1 10} \
  -connect { blockPin padPin padRing corePin floatingStripe } \
  -blockPinTarget { nearestRingStripe nearestTarget } \
  -padPinPortConnect { allPort oneGeom } \
  -checkAlignedSecondaryPin 1 -blockPin useLef -allowJogging 1 \
  -crossoverViaBottomLayer 1 -allowLayerChange 1 -targetViaTopLayer 10 \
  -crossoverViaTopLayer 10 -targetViaBottomLayer 1

#
# Step 6: Placement (Place --> Standard Cell)
#
placeDesign -prePlaceOpt

#
# Step 7: Optimization (preCTS) (Optimize --> Optimize Design)
#
optDesign -preCTS
```

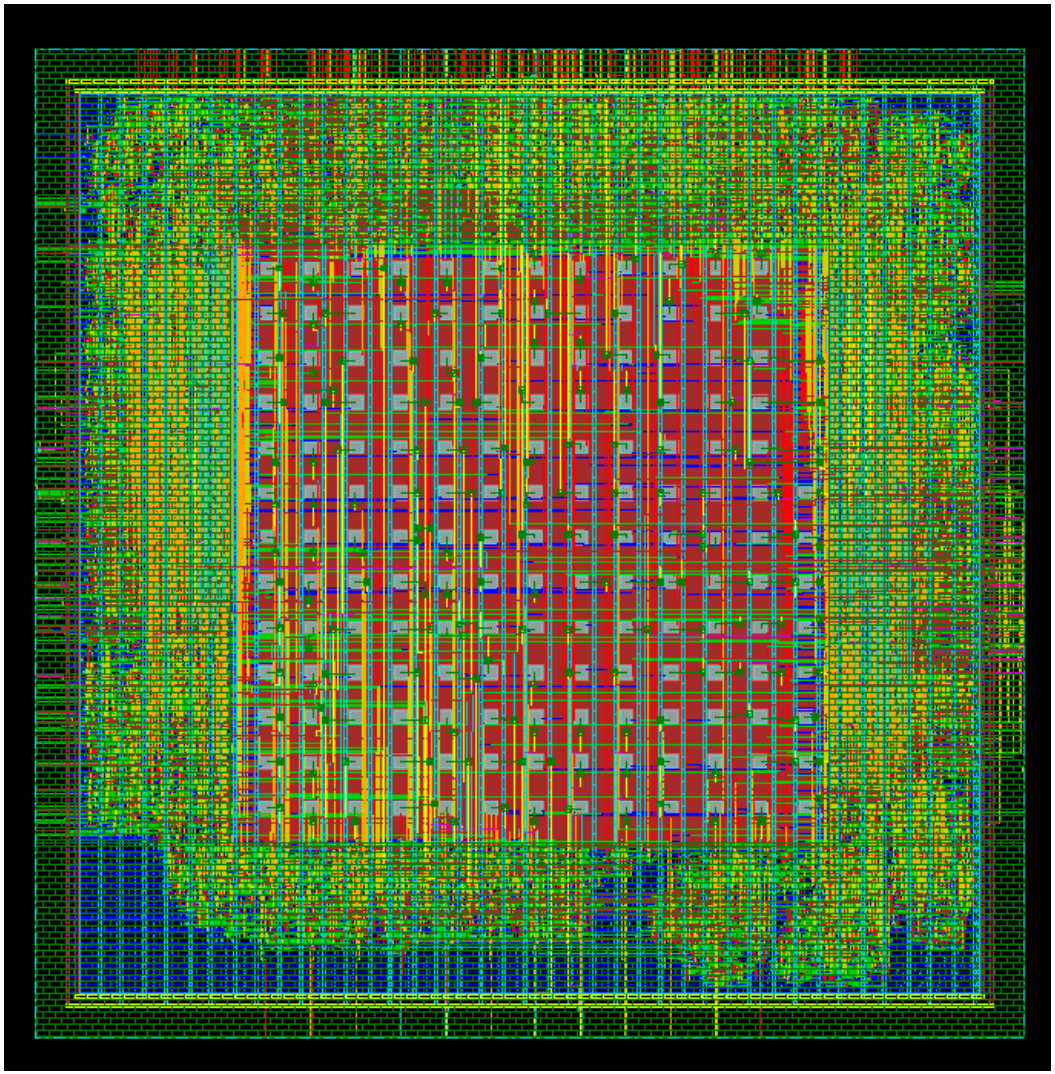
PandR.tcl script (4/5)

```
#
# Step 8: Clock tree synthesis (CTS) (Clock --> Cynthesize Clock Tree)
#
addCTSCellList {CLKBUF_X1 CLKBUF_X2 CLKBUF_X3}
clockDesign -genSpecOnly Clock.ctstch
clockDesign -specFile Clock.ctstch -outDir clock_report -fixedInstBeforeCTS

#
# Step 9: Optimization (postCTS) (Optimize --> Optimize Design)
#
optDesign -postCTS
optDesign -postCTS -hold

#
# Step 10: Detailed route (Route --> Nano Route --> Route)
#
setNanoRouteMode -quiet -routeWithTimingDriven true
setNanoRouteMode -quiet -routeTopRoutingLayer default
setNanoRouteMode -quiet -routeBottomRoutingLayer default
setNanoRouteMode -quiet -drouteEndIteration default
setNanoRouteMode -quiet -routeWithTimingDriven true
routeDesign -globalDetail

#
# Step 11: Optimization (postRoute) (Optimize --> Optimize Design)
#
optDesign -postRoute
optDesign -postRoute -hold
```

PandR.tcl script (5/5)

```
#
# Step 12: Add fillers (Place --> Physical Cells --> Add Filler)
#
 addFiller -prefix FILLER -cell FILLCELL_X1 FILLCELL_X2 FILLCELL_X4 \
   FILLCELL_X8 FILLCELL_X16 FILLCELL_X32

#
# Step 13: Verification (LVS) (Verify --> Verify Connectivity)
#
 verifyConnectivity -type all -error 1000 -warning 50

#
# Step 14: Verification (DRC) (Verify --> Verify Geometry)
#
 verifyGeometry

#
# Step 15: Data out (Timing --> Extract RC, Timing --> Write SDF,
#                    File --> Save --> Netlist)
saveNetlist output_files/router_LAFT_macroTSV_final.vnet
isExtractRCModeSignoff
rcOut -spef output_files/router_LAFT_macroTSV.spef
delayCal -sdf output_files/router_LAFT_macroTSV.sdf -idealclock

saveDesign router_LAFT_macroTSV_final.enc
```

Final chip layout



References

- [Ref.1] A. Ben Ahmed, A. Ben Abdallah, OASIS 3D-Router Hardware Physical Design, Technical Report, Adaptive Systems Laboratory, Division of Computer Engineering, School of Computer Science and Engineering, University of Aizu, July 8, 2014.
- [Ref.2] A. Ben Ahmed, A. Ben Abdallah, Graceful Deadlock-Free Fault-Tolerant Routing Algorithm for 3D Network-on-Chip Architectures, Journal of Parallel and Distributed Computing 74/4 (2014), pp. 2229-2240.
- [Ref.3] http://www.eda.ncsu.edu/wiki/FreePDK3D45:Manual

Acknowledgement

<== Back to Contents

- THIS WORK IS SUPPORTED BY VLSI DESIGN AND EDUCATION CENTER (VDEC), THE UNIVERSITY OF TOKYO, JAPAN, IN COLLABORATION WITH SYNOPSYS, Inc. and CADENCE DESIGN SYSTEMS, Inc.
- THIS WORK IS SUPPORTED BY COMPETITIVE RESEARCH FUNDING, UNIVERSITY OF AIZU, JAPAN, Ref. P-12-2014- 2015.