Spiking Neuron Processing Core (SNPC) Physical Design

Technical Report

© Adaptive Systems Laboratory
Division of Computer Engineering
School of Computer Science and Engineering
University of Aizu

Contact: [d8211104, benab] @ u-aizu.ac.jp

Edition: April 22, 2021

SNPC Design Phases

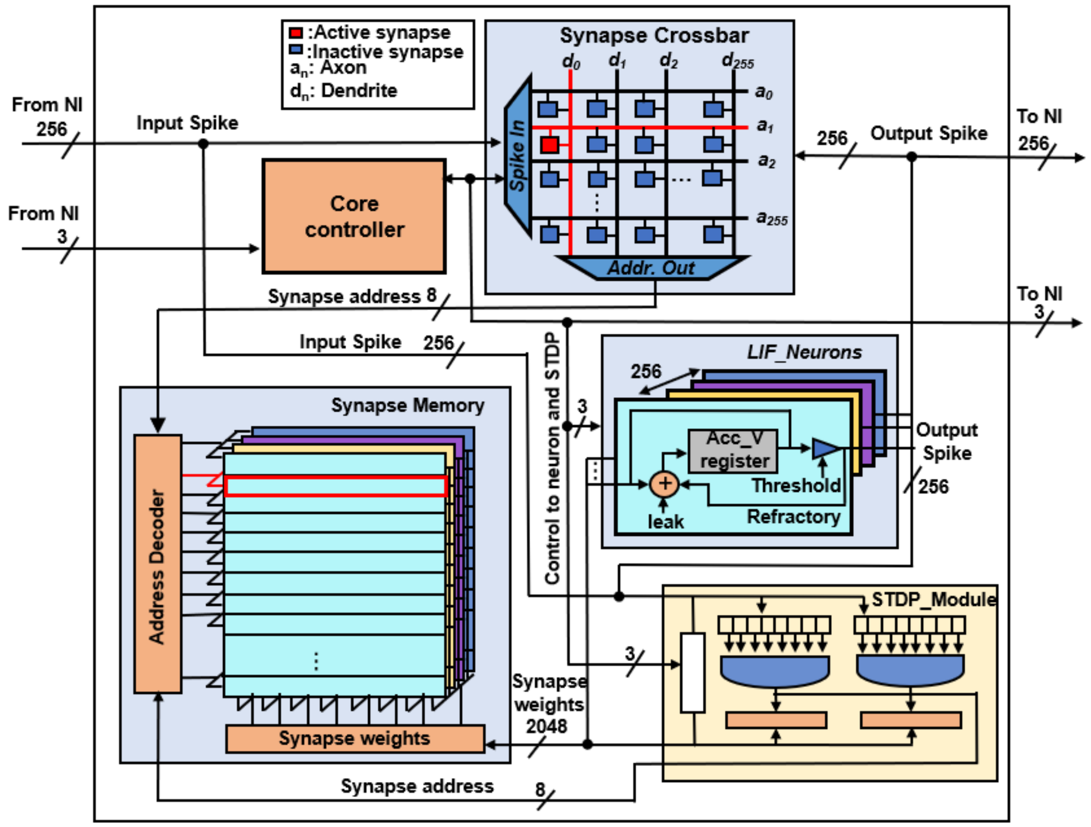
1. Specification and RTL design.
2. RTL Functional Simulation with Modelsim.
3. Synthesis with Cadence Genus.
4. Gate Level Simulation.
5. Place and Route with Cadence Innovus.

Phase 1:
Specification and RTL design

Contents

- Architecture
- Specification
- RTL Design

High Level View of SNPC Architecture



Specification and RTL Design

- First the logical behavior of the SNPC design is specified in Verilog HDL language. The logical behavior is described in the RTL level of abstraction, because it's most suitable for clocked designs. The SNPC design consist of sub-designs which are described in separate HDL codes. These sub designs are used in designs higher in hierarchy.
- Specification:
  - 256 leaky integrate and fire (LIF) neurons
  - 65K synapses: 256 SRAMs (256-bank, 8-bit)
  - Spike timing-dependent plasticity (STDP) on-chip learning
- The design hierarchy of SNPC and the sub-designs is described below:

SNPC

```
├SNPC_cntrl (core controller)
├LIF_neuron
└Xbar (synapse crossbar, synapse memory)
```

Phase 2:
RTL Functional Simulation with Modelsim

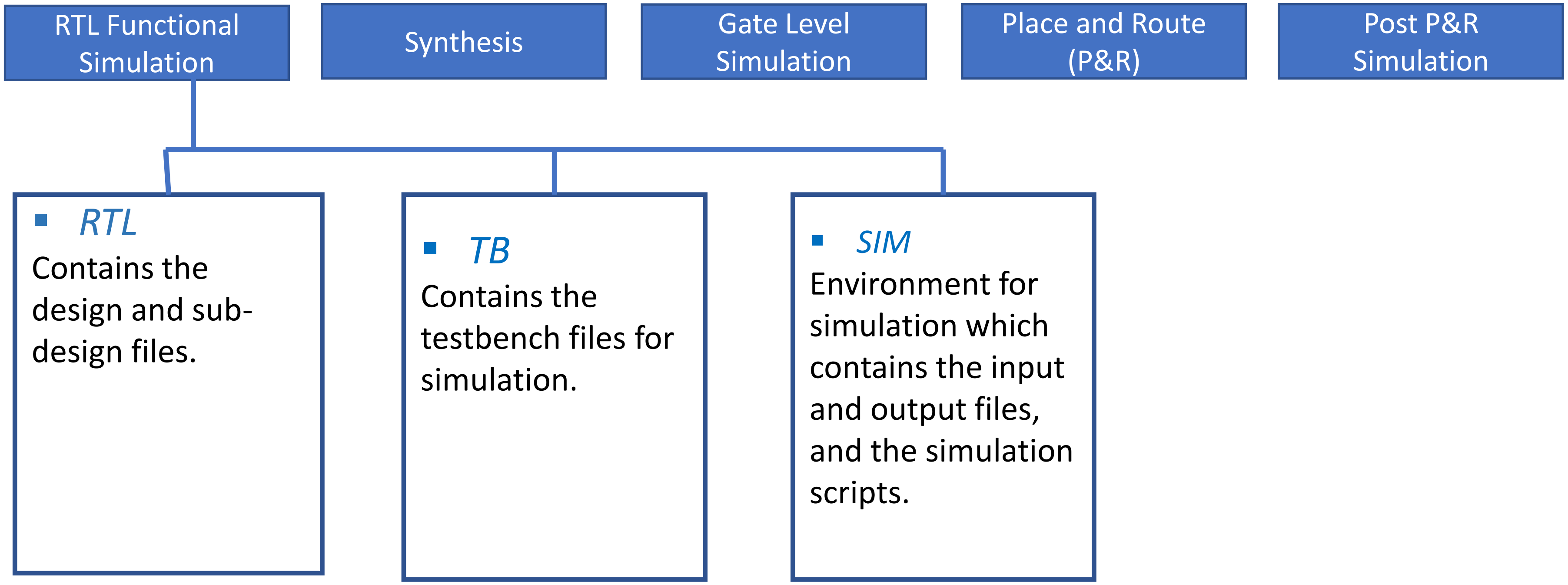
Contents

- RTL Functional simulation directory structure
- Environment setup
- Modelsim simulation and Result Verification

RTL Simulation with Modelsim

- To test the logical behavior of the SNPC and the sub-designs, testbenches are specified. The testbenches are basically also a description in Verilog HDL which supplies the designs and sub-design with stimulus signals. The functional simulation is performed with interactive batch scripts. The SNPC design sources are first loaded in modelsim, and compiled starting from the top design, to the sub-designs, and then the testbench source. After compilation, the test bench is run, to perform the simulation (Both the compilation and simulation is performed using the interactive batch scripts). For this functional simulation, the MNIST dataset of 10,000 16×16 images are classified according to their labels from 1-9. The weights for this classification has been trained off-chip as ANN, and then converted to SNN. The MNIST images are converted to spikes using Poisson distribution.
- Specification:
  - Network: 256:10 (single layer), 256:225:10 (2 layer)
  - Synapses: 25,600, 59,850 (57,600 + 2250) 8-bit
- This simulation is going to be performed using batch scripts are in ***~/SNPC/RTL_SIM/script/*** directory.
- The trained synapse weights are loaded into the synapse memory before simulation begins.

Directory Structure



RTL Functional Simulation

Synthesis

Gate Level Simulation

Place and Route (P&R)

Post P&R Simulation

- *RTL*

  Contains the design and sub-design files.

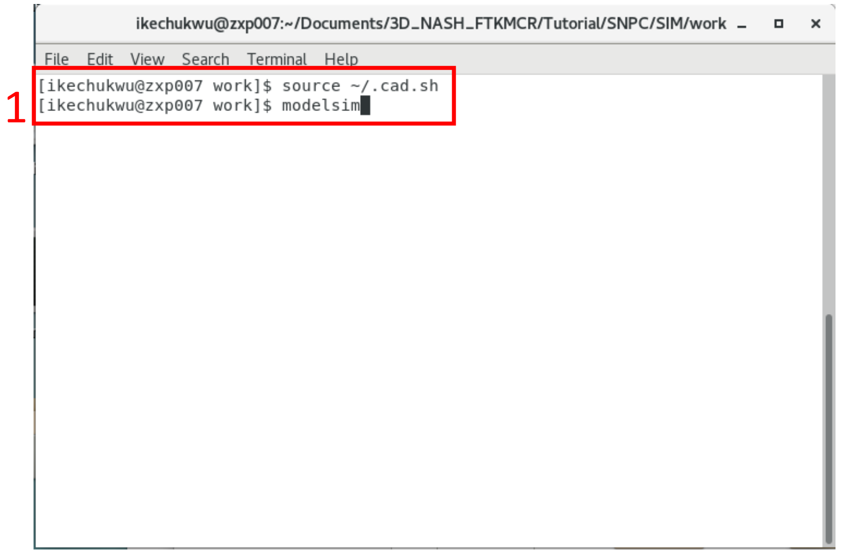
- *TB*

  Contains the testbench files for simulation.

- *SIM*

  Environment for simulation which contains the input and output files, and the simulation scripts.

Environment Setup


ikechukwu@zxp007:~/Documents/3D_NASH_FTKMCR/Tutorial/SNPC/SIM/work
File Edit View Search Terminal Help
1

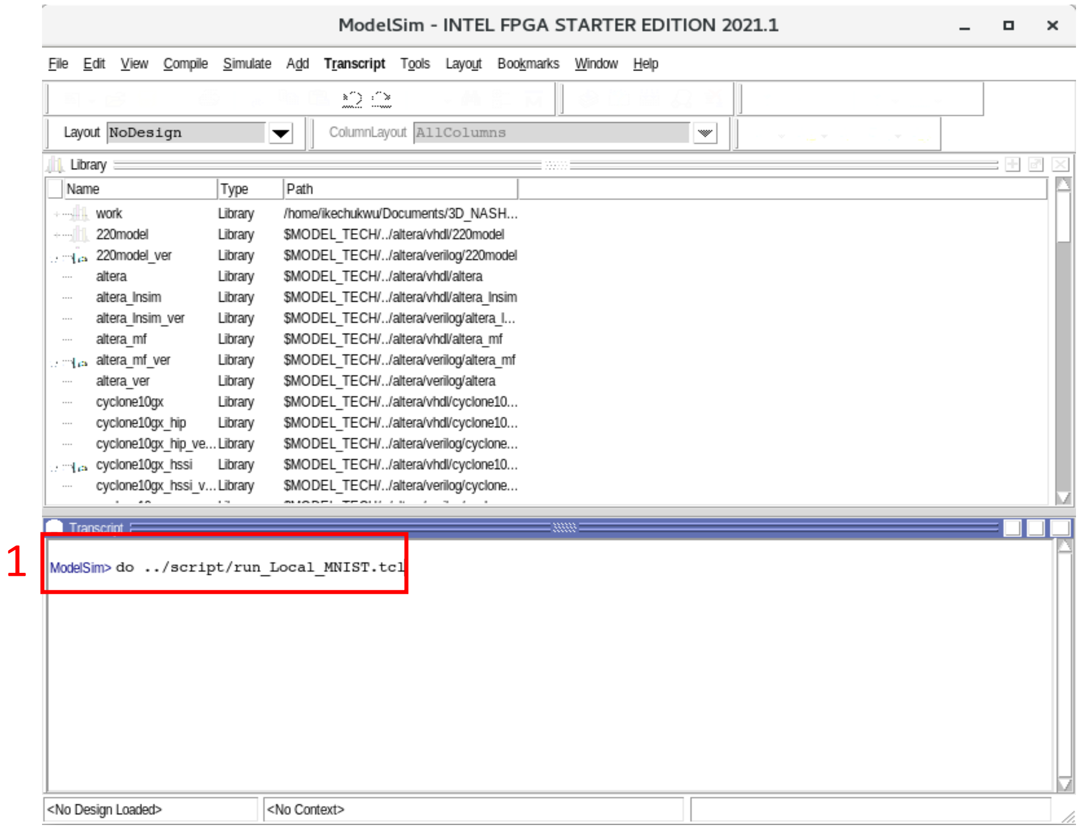
```
[ikechukwu@zxp007 work]$ source ~/.cad.sh
[ikechukwu@zxp007 work]$ modelsim
```

(1) To set the functional simulator working environment, from the terminal, navigate to **~/SNPC/RTL_SIM/work/ .** If you are using file explorer, go to the same directory, right click and select **Open in Terminal**. Use the following commands to set the environment then launch modelsim.

source ~/.cad.sh

modelsim

Begin RTL Simulation



ModelSim - INTEL FPGA STARTER EDITION 2021.1

File Edit View Compile Simulate Add Transcript Tools Layout Bookmarks Window Help
Layout NoDesign
ColumnLayout AllColumns
Library
1
Transcript

Name	Type	Path
work	Library	/home/ikechukwu/Documents/3D_NASH...
220model	Library	$MODEL_TECH/../altera/vhdl/220model
220model_ver	Library	$MODEL_TECH/../altera/verilog/220model
altera	Library	$MODEL_TECH/../altera/vhdl/altera
altera_lnsim	Library	$MODEL_TECH/../altera/vhdl/altera_lnsim
altera_lnsim_ver	Library	$MODEL_TECH/../altera/verilog/altera_l...
altera_mf	Library	$MODEL_TECH/../altera/vhdl/altera_mf
altera_mf_ver	Library	$MODEL_TECH/../altera/verilog/altera_mf
altera_ver	Library	$MODEL_TECH/../altera/verilog/altera
cyclone10gx	Library	$MODEL_TECH/../altera/vhdl/cyclone10...
cyclone10gx_hip	Library	$MODEL_TECH/../altera/vhdl/cyclone10...
cyclone10gx_hip_ve...	Library	$MODEL_TECH/../altera/verilog/cyclone...
cyclone10gx_hssi	Library	$MODEL_TECH/../altera/vhdl/cyclone10...
cyclone10gx_hssi_v...	Library	$MODEL_TECH/../altera/verilog/cyclone...

```
ModelSim> do ../script/run_Local_MNIST.tcl
```

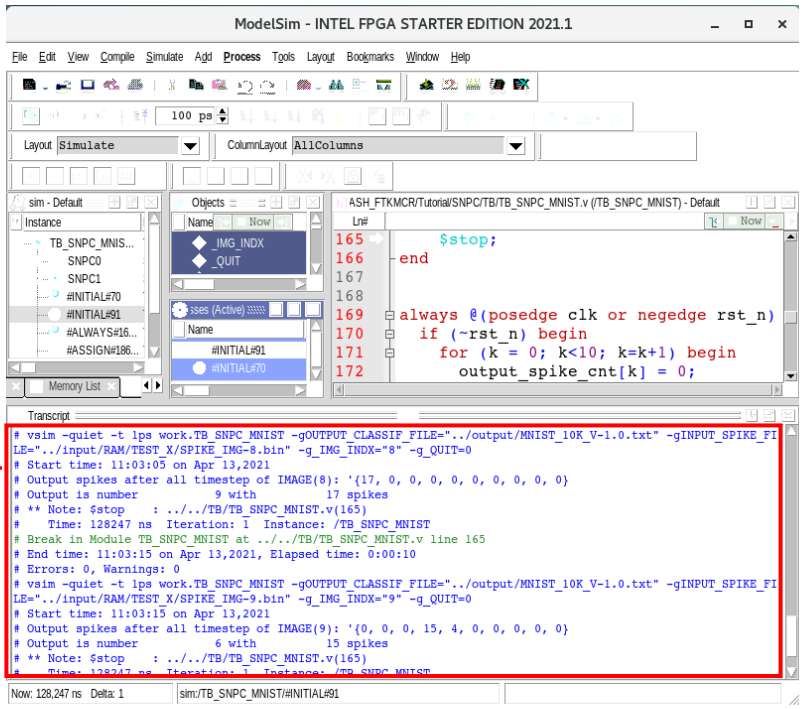
<No Design Loaded> <No Context>

(1) After launching modelsim, use the following command to run the simulation.

do ../script/run_Local_MNIST.tcl

Verification of RTL Simulation Results

1

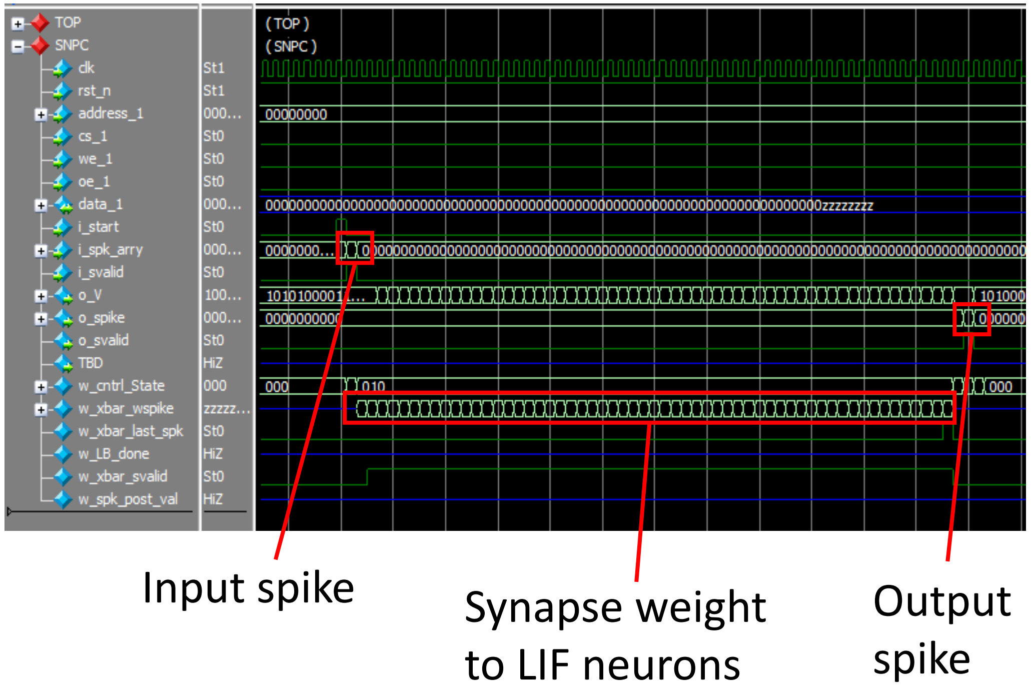


```
    $stop;
end

always @(posedge clk or negedge rst_n)
  if (~rst_n) begin
    for (k = 0; k<10; k=k+1) begin
      output_spike_cnt[k] = 0;
```

```
# vsim -quiet -t 1ps work.TB_SNPC_MNIST -gOUTPUT_CLASSIF_FILE="../output/MNIST_10K_V-1.0.txt" -gINPUT_SPIKE_FI
LE="../input/RAM/TEST_X/SPIKE_IMG-8.bin" -g_IMG_INDX="8" -g_QUIT=0
# Start time: 11:03:05 on Apr 13,2021
# Output spikes after all timestep of IMAGE(8): '{17, 0, 0, 0, 0, 0, 0, 0, 0, 0}
# Output is number           9 with          17 spikes
# ** Note: $stop    : ../../TB/TB_SNPC_MNIST.v(165)
#    Time: 128247 ns  Iteration: 1  Instance: /TB_SNPC_MNIST
# Break in Module TB_SNPC_MNIST at ../../TB/TB_SNPC_MNIST.v line 165
# End time: 11:03:15 on Apr 13,2021, Elapsed time: 0:00:10
# Errors: 0, Warnings: 0
# vsim -quiet -t 1ps work.TB_SNPC_MNIST -gOUTPUT_CLASSIF_FILE="../output/MNIST_10K_V-1.0.txt" -gINPUT_SPIKE_FI
LE="../input/RAM/TEST_X/SPIKE_IMG-9.bin" -g_IMG_INDX="9" -g_QUIT=0
# Start time: 11:03:15 on Apr 13,2021
# Output spikes after all timestep of IMAGE(9): '{0, 0, 0, 15, 4, 0, 0, 0, 0, 0}
# Output is number           6 with          15 spikes
# ** Note: $stop    : ../../TB/TB_SNPC_MNIST.v(165)
#    Time: 128247 ns  Iteration: 1  Instance: /TB_SNPC_MNIST
```

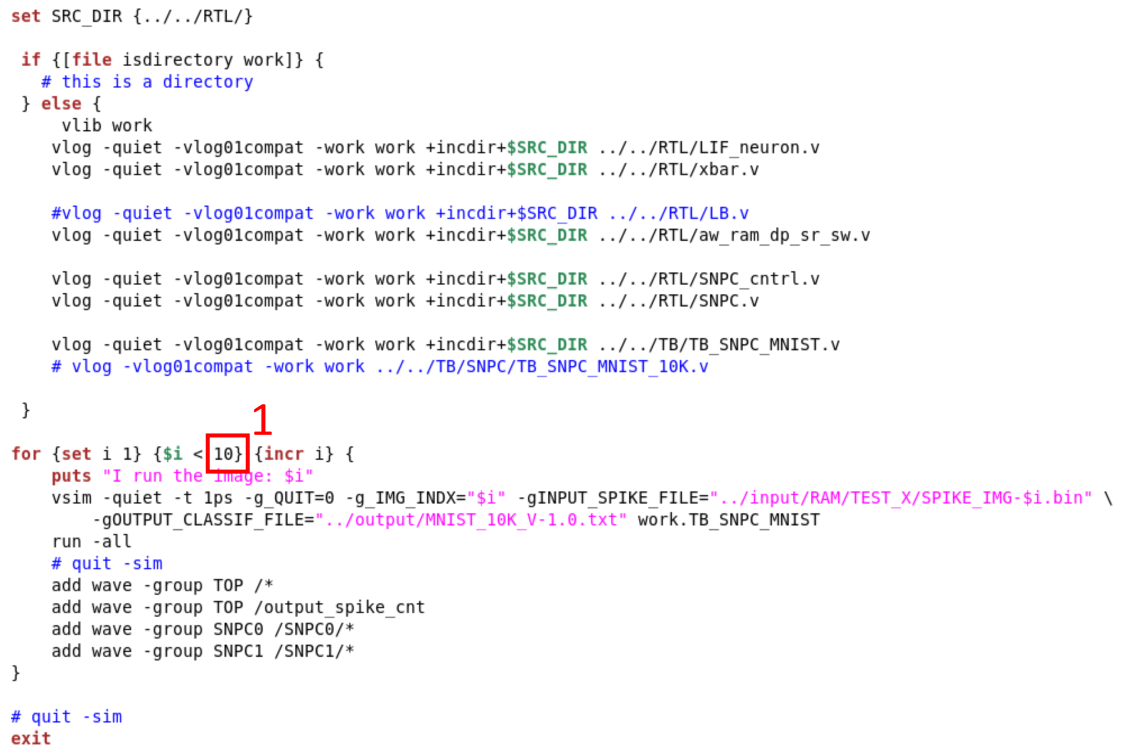
```
Now: 128,247 ns   Delta: 1     sim:/TB_SNPC_MNIST/#INITIAL#91
```

2



(1) The simulation result is displayed in the modelsim transcript.
(2) Simulation waveform of the SNPC showing the input spikes, synapse weights sent to the LIF neurons and output spike.

Modify RTL Simulation Script



```
set SRC_DIR {../../RTL/}

 if {[file isdirectory work]} {
   # this is a directory
 } else {
     vlib work
    vlog -quiet -vlog01compat -work work +incdir+$SRC_DIR ../../RTL/LIF_neuron.v
    vlog -quiet -vlog01compat -work work +incdir+$SRC_DIR ../../RTL/xbar.v

    #vlog -quiet -vlog01compat -work work +incdir+$SRC_DIR ../../RTL/LB.v
    vlog -quiet -vlog01compat -work work +incdir+$SRC_DIR ../../RTL/aw_ram_dp_sr_sw.v

    vlog -quiet -vlog01compat -work work +incdir+$SRC_DIR ../../RTL/SNPC_cntrl.v
    vlog -quiet -vlog01compat -work work +incdir+$SRC_DIR ../../RTL/SNPC.v

    vlog -quiet -vlog01compat -work work +incdir+$SRC_DIR ../../TB/TB_SNPC_MNIST.v
    # vlog -vlog01compat -work work ../../TB/SNPC/TB_SNPC_MNIST_10K.v

 }

for {set i 1} {$i < 10} {incr i} {
    puts "I run the image: $i"
    vsim -quiet -t 1ps -g_QUIT=0 -g_IMG_INDX="$i" -gINPUT_SPIKE_FILE="../input/RAM/TEST_X/SPIKE_IMG-$i.bin" \
        -gOUTPUT_CLASSIF_FILE="../output/MNIST_10K_V-1.0.txt" work.TB_SNPC_MNIST
    run -all
    # quit -sim
    add wave -group TOP /*
    add wave -group TOP /output_spike_cnt
    add wave -group SNPC0 /SNPC0/*
    add wave -group SNPC1 /SNPC1/*
}

# quit -sim
exit
```

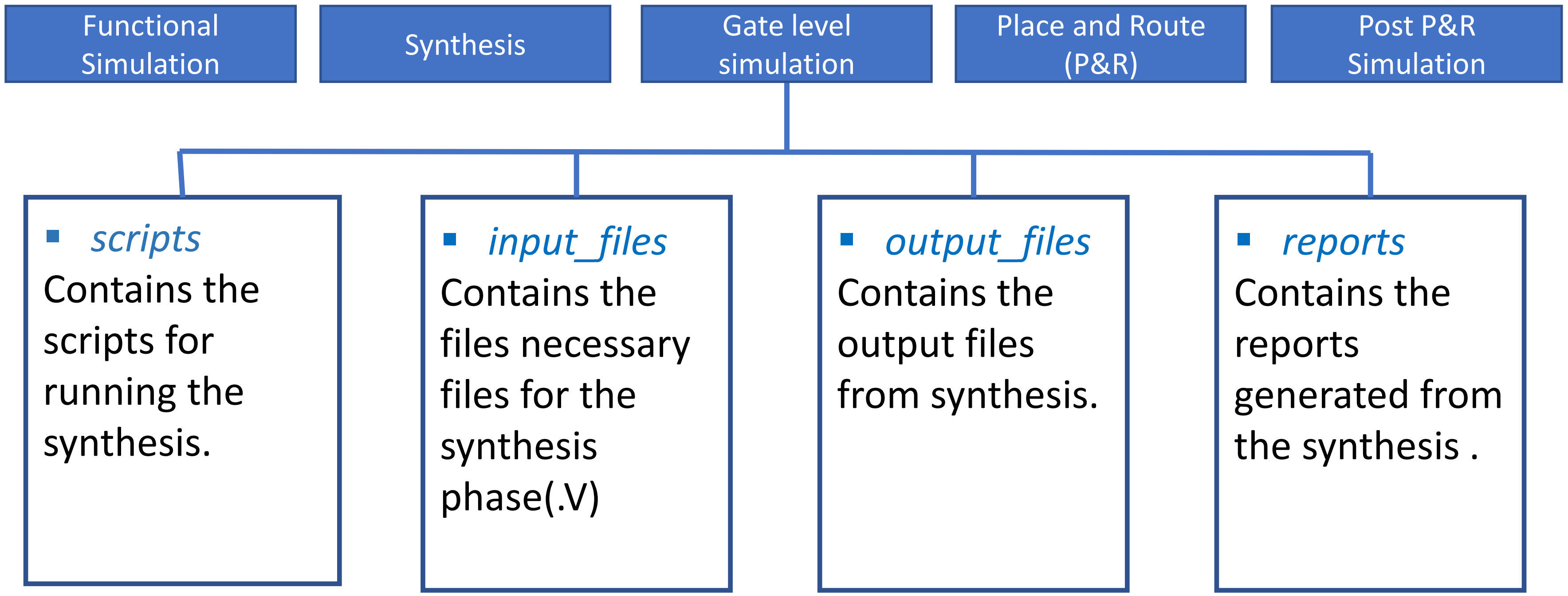
(1) Note: The MNIST simulation runs for 10 images by default. To change this number, open ***~/SNPC/RTL_SIM/script/ run_Local_MNIST.tcl*** in text editor and the number of images can be changed to a maximum of 10,000. The simulation output is stored in ***~/SNPC/RTL_SIM/output***

Phase 3:
Synthesis with
Cadence Genus

Contents

- Synthesis directory structure
- Environment and Genus synthesis steps
- Synthesis generated files and reports

Synthesis directory structure



- *scripts*

Contains the scripts for running the synthesis.

- *input_files*

Contains the files necessary files for the synthesis phase(.V)

- *output_files*

Contains the output files from synthesis.

- *reports*

Contains the reports generated from the synthesis .

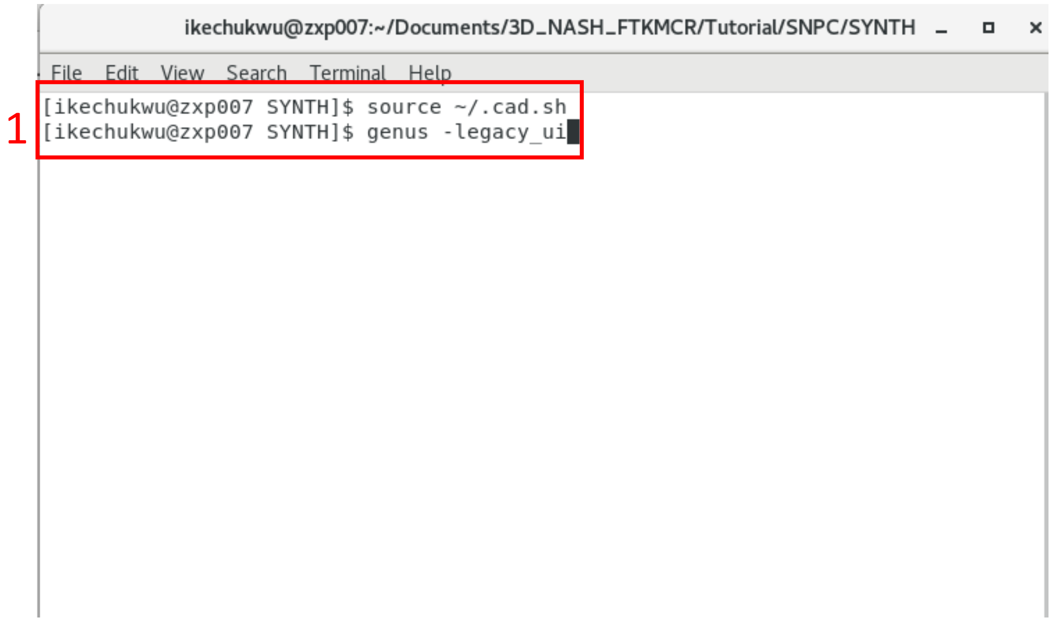
Synthesis with Cadence Genus

- Now that the SNPC RTL design is logically valid, it should be synthesized into design which consists of only the standard cells and macros from the different libraries. For this, the single layer network and Cadence genus tool will be used. The synthesis is going to be done with a script. The script is written in TCL language, and specify the synthesis configuration, files, and constraints like clock period, input/output delay, library etc. The Nangate open cell library is used for this synthesis.

- The synthesis is going to be done in the ***~/SNPC/SYNTH*** directory. From terminal, navigate to the ***~/SNPC/SYNTH/*** directory, or if you are using file explorer, go to the directory, right click and select **Open in Terminal**.

- The SNPC has 256 neurons. But to be able to perform gate level simulation and the post P&R simulation, we will synthesize for the single layer network with 10 neurons in the SNPC (Note: you can change the number of neurons in the SNPC to 256 before synthesis).

Step 1: Environment and tool launch

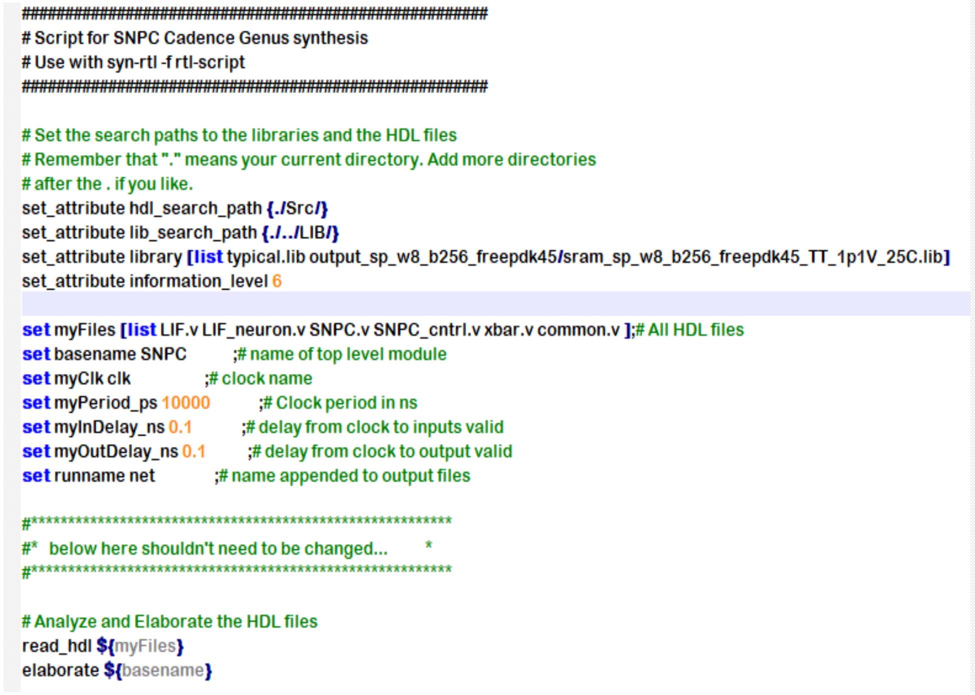

ikechukwu@zxp007:~/Documents/3D_NASH_FTKMCR/Tutorial/SNPC/SYNTH
File Edit View Search Terminal Help
1

```
[ikechukwu@zxp007 SYNTH]$ source ~/.cad.sh
[ikechukwu@zxp007 SYNTH]$ genus -legacy_ui
```

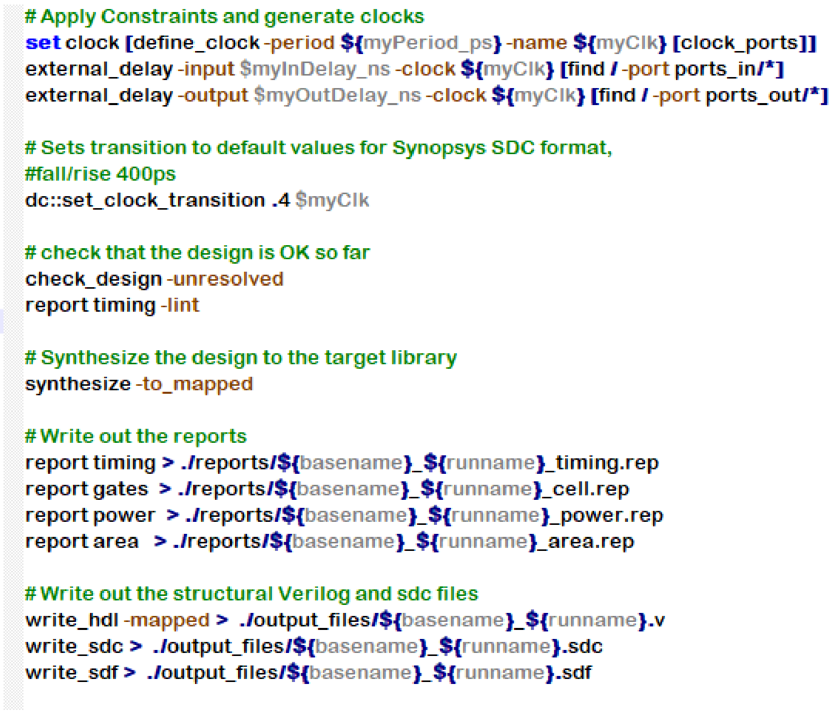
(1) In terminal, type the following command to set the environment and launch Cadence genus:

source ~/.cad.sh
genus -legacy_ui

Synthesis Script



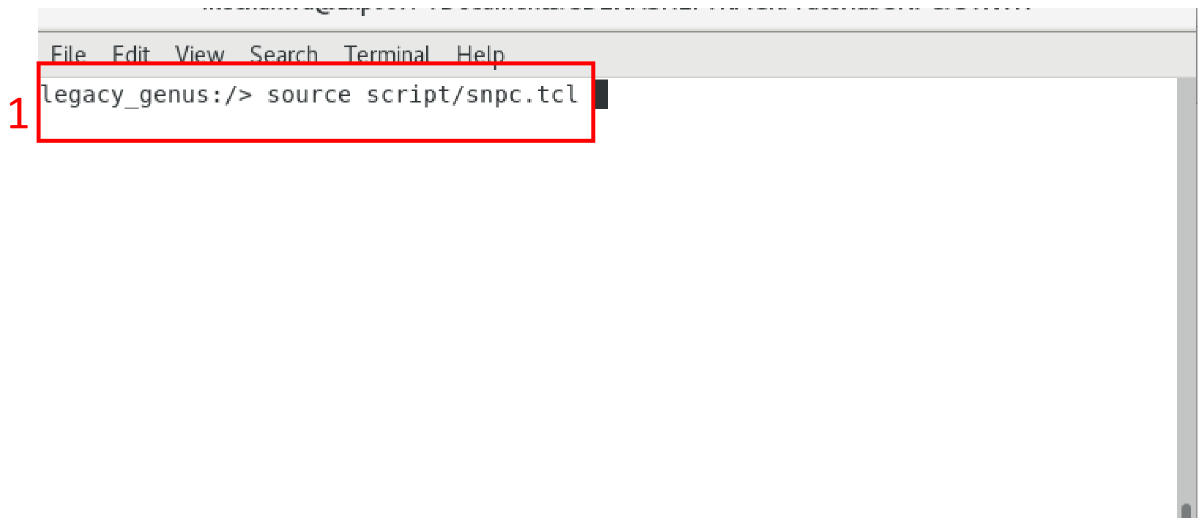
```
#######################################################################
# Script for SNPC Cadence Genus synthesis
# Use with syn-rtl -f rtl-script
#######################################################################

# Set the search paths to the libraries and the HDL files
# Remember that "." means your current directory. Add more directories
# after the . if you like.
set_attribute hdl_search_path {./Src/}
set_attribute lib_search_path {./../LIB/}
set_attribute library [list typical.lib output_sp_w8_b256_freepdk45/sram_sp_w8_b256_freepdk45_TT_1p1V_25C.lib]
set_attribute information_level 6

set myFiles [list LIF.v LIF_neuron.v SNPC.v SNPC_cntrl.v xbar.v common.v ];# All HDL files
set basename SNPC        ;# name of top level module
set myClk clk            ;# clock name
set myPeriod_ps 10000       ;# Clock period in ns
set myInDelay_ns 0.1       ;# delay from clock to inputs valid
set myOutDelay_ns 0.1       ;# delay from clock to output valid
set runname net          ;# name appended to output files

#*************************************************************
#*   below here shouldn't need to be changed...       *
#*************************************************************

# Analyze and Elaborate the HDL files
read_hdl ${myFiles}
elaborate ${basename}
```



```
# Apply Constraints and generate clocks
set clock [define_clock -period ${myPeriod_ps} -name ${myClk} [clock_ports]]
external_delay -input $myInDelay_ns -clock ${myClk} [find / -port ports_in/*]
external_delay -output $myOutDelay_ns -clock ${myClk} [find / -port ports_out/*]

# Sets transition to default values for Synopsys SDC format,
#fall/rise 400ps
dc::set_clock_transition .4 $myClk

# check that the design is OK so far
check_design -unresolved
report timing -lint

# Synthesize the design to the target library
synthesize -to_mapped

# Write out the reports
report timing > ./reports/${basename}_${runname}_timing.rep
report gates  > ./reports/${basename}_${runname}_cell.rep
report power  > ./reports/${basename}_${runname}_power.rep
report area   > ./reports/${basename}_${runname}_area.rep

# Write out the structural Verilog and sdc files
write_hdl -mapped >  ./output_files/${basename}_${runname}.v
write_sdc > ./output_files/${basename}_${runname}.sdc
write_sdf >  ./output_files/${basename}_${runname}.sdf
```

- The synthesis script can be found in ***~/SNPC/SYNTH/script/*** directory. As shown in the image above, the design source and libraries are specified then the timing parameters are set before synthesis begins.

Step 2: Begin Synthesis



```
legacy_genus:/> source script/snpc.tcl
```

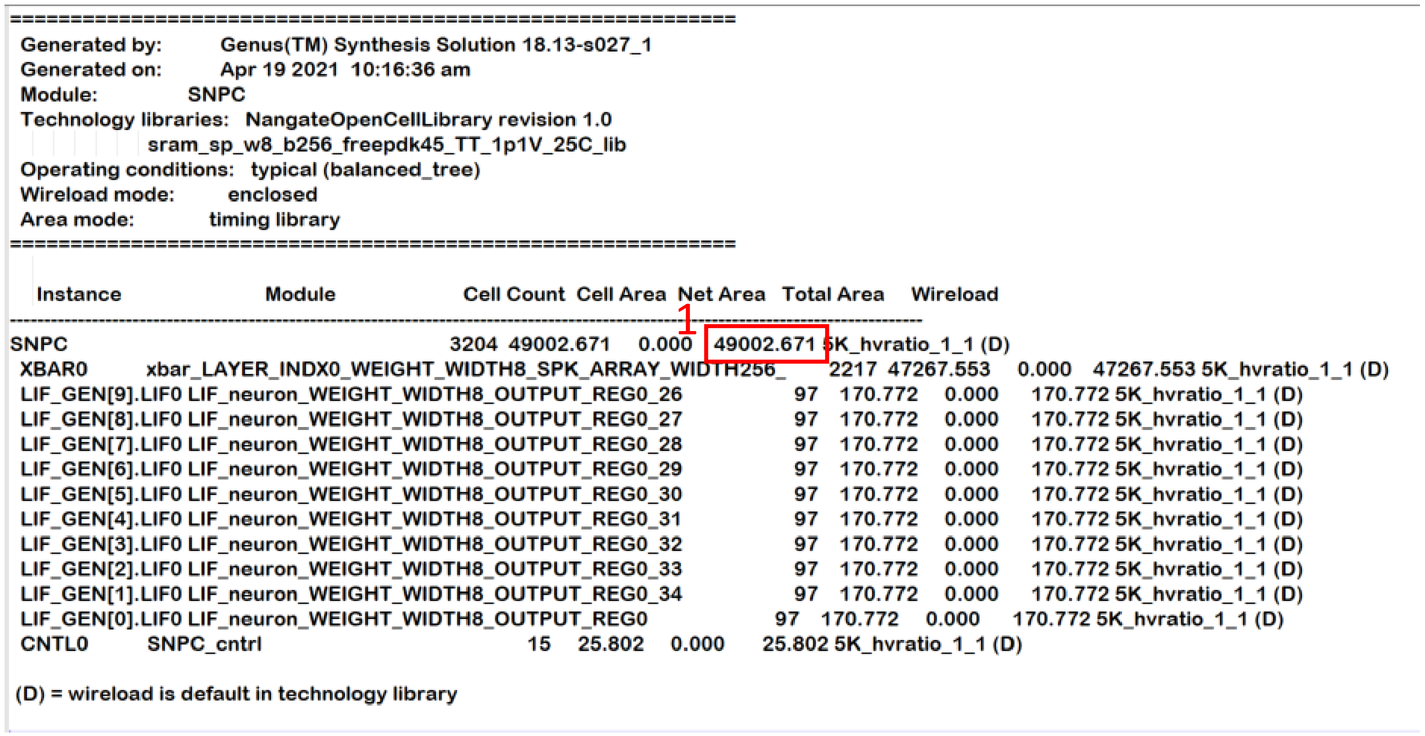
(1) In terminal, to run the synthesis script, type the following command:

source script/snpc.tcl

Generated Files and Reports

- When the synthesis is done, the generated netlist and constraint files are saved in the ***~SNPC/SYNTH/output_files/*** directory. The hardware complexity reports (area, cell, power, and timing) of both instances of SNPC are saved in the ***~SNPC/SYNTH/reports/*** *directory.*

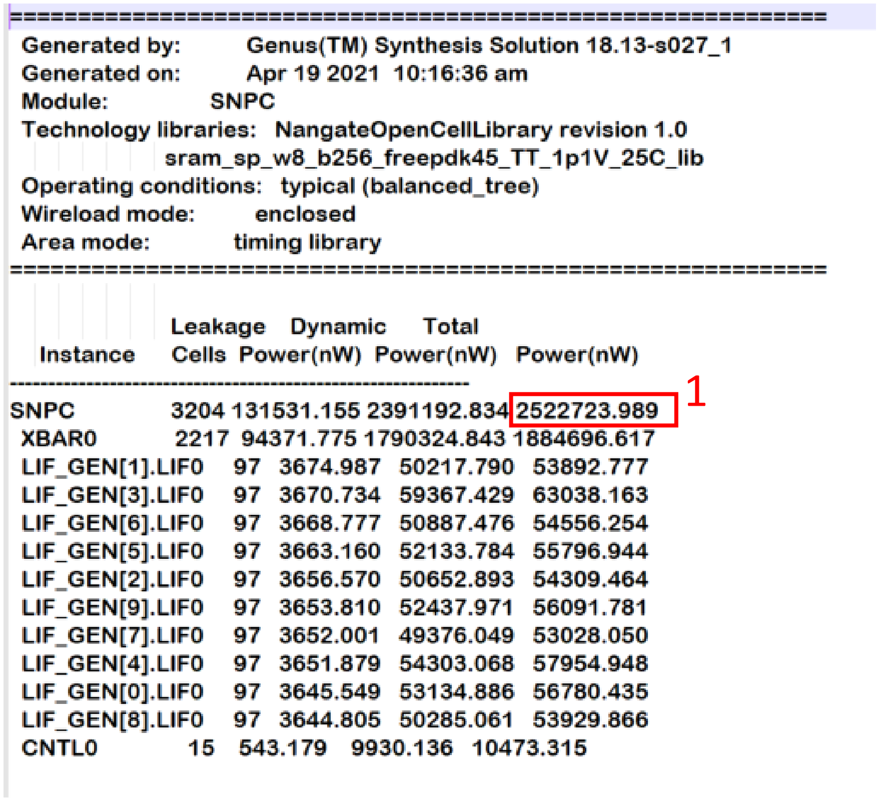
Area Report



```
============================================================
Generated by:       Genus(TM) Synthesis Solution 18.13-s027_1
Generated on:       Apr 19 2021  10:16:36 am
Module:             SNPC
Technology libraries:   NangateOpenCellLibrary revision 1.0
                        sram_sp_w8_b256_freepdk45_TT_1p1V_25C_lib
Operating conditions:   typical (balanced_tree)
Wireload mode:          enclosed
Area mode:              timing library
============================================================

  Instance              Module                  Cell Count  Cell Area  Net Area  Total Area   Wireload
---------------------------------------------------------------------------------------------------------
SNPC                                               3204  49002.671     0.000   49002.671 5K_hvratio_1_1 (D)
 XBAR0       xbar_LAYER_INDX0_WEIGHT_WIDTH8_SPK_ARRAY_WIDTH256_   2217  47267.553     0.000   47267.553 5K_hvratio_1_1 (D)
 LIF_GEN[9].LIF0 LIF_neuron_WEIGHT_WIDTH8_OUTPUT_REG0_26      97   170.772    0.000    170.772 5K_hvratio_1_1 (D)
 LIF_GEN[8].LIF0 LIF_neuron_WEIGHT_WIDTH8_OUTPUT_REG0_27      97   170.772    0.000    170.772 5K_hvratio_1_1 (D)
 LIF_GEN[7].LIF0 LIF_neuron_WEIGHT_WIDTH8_OUTPUT_REG0_28      97   170.772    0.000    170.772 5K_hvratio_1_1 (D)
 LIF_GEN[6].LIF0 LIF_neuron_WEIGHT_WIDTH8_OUTPUT_REG0_29      97   170.772    0.000    170.772 5K_hvratio_1_1 (D)
 LIF_GEN[5].LIF0 LIF_neuron_WEIGHT_WIDTH8_OUTPUT_REG0_30      97   170.772    0.000    170.772 5K_hvratio_1_1 (D)
 LIF_GEN[4].LIF0 LIF_neuron_WEIGHT_WIDTH8_OUTPUT_REG0_31      97   170.772    0.000    170.772 5K_hvratio_1_1 (D)
 LIF_GEN[3].LIF0 LIF_neuron_WEIGHT_WIDTH8_OUTPUT_REG0_32      97   170.772    0.000    170.772 5K_hvratio_1_1 (D)
 LIF_GEN[2].LIF0 LIF_neuron_WEIGHT_WIDTH8_OUTPUT_REG0_33      97   170.772    0.000    170.772 5K_hvratio_1_1 (D)
 LIF_GEN[1].LIF0 LIF_neuron_WEIGHT_WIDTH8_OUTPUT_REG0_34      97   170.772    0.000    170.772 5K_hvratio_1_1 (D)
 LIF_GEN[0].LIF0 LIF_neuron_WEIGHT_WIDTH8_OUTPUT_REG0      97   170.772    0.000    170.772 5K_hvratio_1_1 (D)
 CNTL0       SNPC_cntrl                        15    25.802    0.000     25.802 5K_hvratio_1_1 (D)

(D) = wireload is default in technology library
```

- The area report from synthesizing an SNPC with the single layer network is shown in the figure above describing the ten neurons, their synapse memory contained in the crossbar (xbar), and the control unit. The total area is depicted in 1

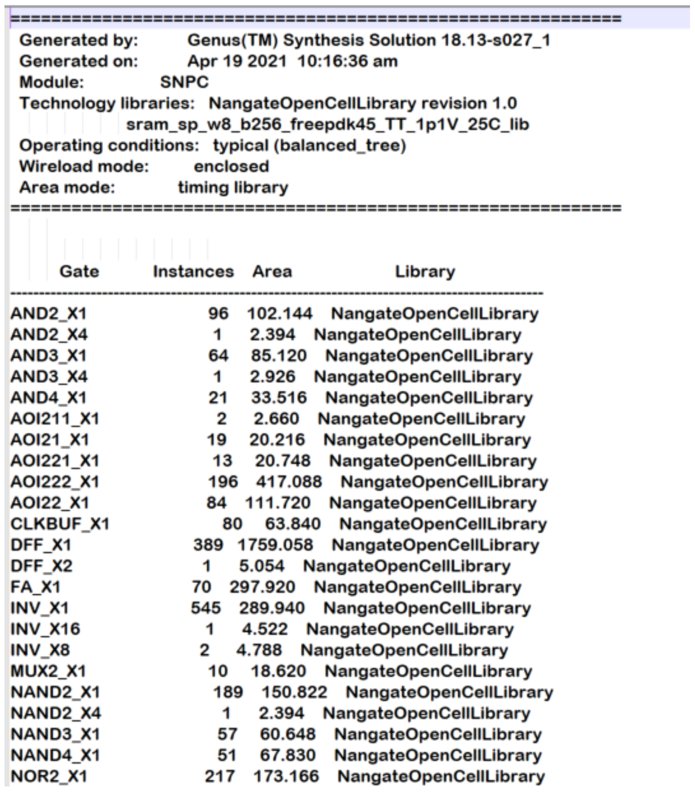
Power Report



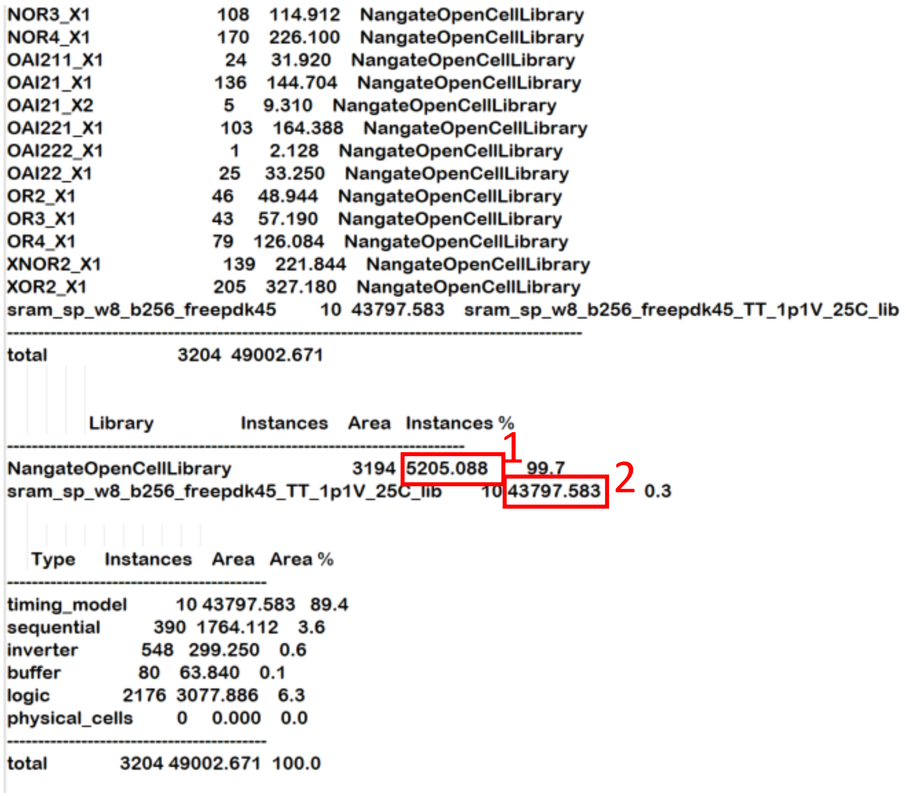
```
=================================================================
 Generated by:         Genus(TM) Synthesis Solution 18.13-s027_1
 Generated on:         Apr 19 2021  10:16:36 am
 Module:               SNPC
 Technology libraries:  NangateOpenCellLibrary revision 1.0
                        sram_sp_w8_b256_freepdk45_TT_1p1V_25C_lib
 Operating conditions:  typical (balanced_tree)
 Wireload mode:         enclosed
 Area mode:             timing library
=================================================================

                      Leakage     Dynamic      Total
   Instance    Cells  Power(nW)  Power(nW)   Power(nW)
------------------------------------------------------
SNPC            3204 131531.155 2391192.834 2522723.989
 XBAR0          2217  94371.775 1790324.843 1884696.617
 LIF_GEN[1].LIF0  97   3674.987   50217.790   53892.777
 LIF_GEN[3].LIF0  97   3670.734   59367.429   63038.163
 LIF_GEN[6].LIF0  97   3668.777   50887.476   54556.254
 LIF_GEN[5].LIF0  97   3663.160   52133.784   55796.944
 LIF_GEN[2].LIF0  97   3656.570   50652.893   54309.464
 LIF_GEN[9].LIF0  97   3653.810   52437.971   56091.781
 LIF_GEN[7].LIF0  97   3652.001   49376.049   53028.050
 LIF_GEN[4].LIF0  97   3651.879   54303.068   57954.948
 LIF_GEN[0].LIF0  97   3645.549   53134.886   56780.435
 LIF_GEN[8].LIF0  97   3644.805   50285.061   53929.866
 CNTL0            15    543.179    9930.136   10473.315
```

- The power report from synthesizing an SNPC with the single layer network is shown in the figure above describing the ten neurons, their synapse memory contained in the crossbar (xbar), and the control unit. The total power is depicted in 1

Cell Report



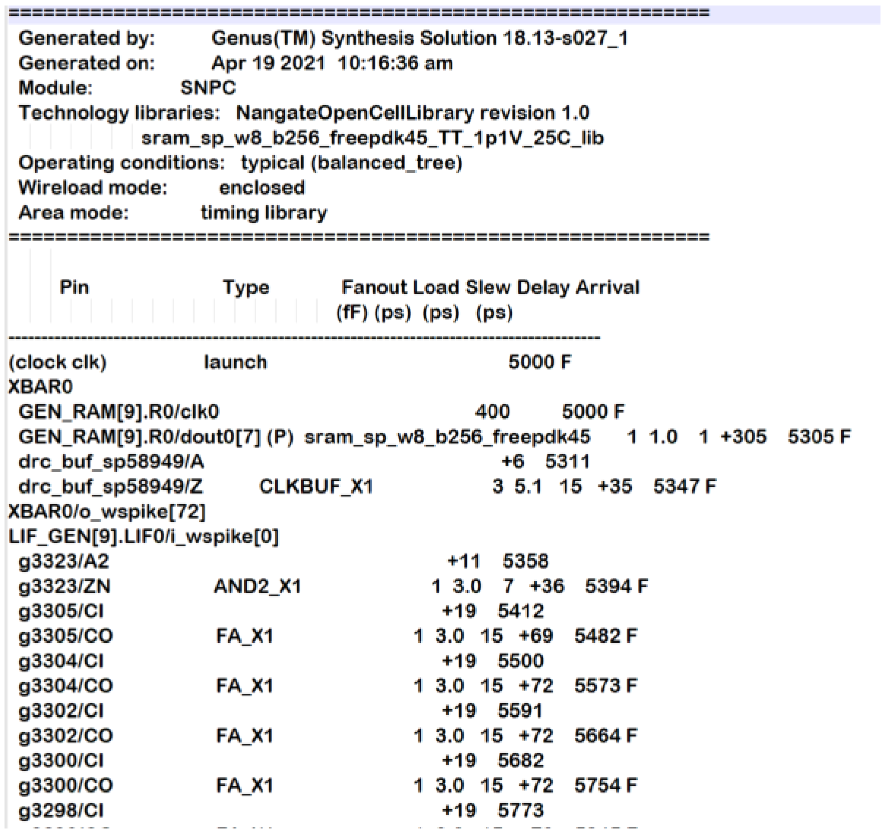
```
============================================================
 Generated by:        Genus(TM) Synthesis Solution 18.13-s027_1
 Generated on:        Apr 19 2021  10:16:36 am
 Module:              SNPC
 Technology libraries:  NangateOpenCellLibrary revision 1.0
                        sram_sp_w8_b256_freepdk45_TT_1p1V_25C_lib
 Operating conditions:  typical (balanced_tree)
 Wireload mode:         enclosed
 Area mode:             timing library
============================================================

     Gate       Instances   Area           Library
---------------------------------------------------------
AND2_X1              96   102.144   NangateOpenCellLibrary
AND2_X4               1     2.394   NangateOpenCellLibrary
AND3_X1              64    85.120   NangateOpenCellLibrary
AND3_X4               1     2.926   NangateOpenCellLibrary
AND4_X1              21    33.516   NangateOpenCellLibrary
AOI211_X1             2     2.660   NangateOpenCellLibrary
AOI21_X1             19    20.216   NangateOpenCellLibrary
AOI221_X1            13    20.748   NangateOpenCellLibrary
AOI222_X1           196   417.088   NangateOpenCellLibrary
AOI22_X1             84   111.720   NangateOpenCellLibrary
CLKBUF_X1            80    63.840   NangateOpenCellLibrary
DFF_X1              389  1759.058   NangateOpenCellLibrary
DFF_X2                1     5.054   NangateOpenCellLibrary
FA_X1                70   297.920   NangateOpenCellLibrary
INV_X1              545   289.940   NangateOpenCellLibrary
INV_X16               1     4.522   NangateOpenCellLibrary
INV_X8                2     4.788   NangateOpenCellLibrary
MUX2_X1              10    18.620   NangateOpenCellLibrary
NAND2_X1            189   150.822   NangateOpenCellLibrary
NAND2_X4              1     2.394   NangateOpenCellLibrary
NAND3_X1             57    60.648   NangateOpenCellLibrary
NAND4_X1             51    67.830   NangateOpenCellLibrary
NOR2_X1             217   173.166   NangateOpenCellLibrary
```



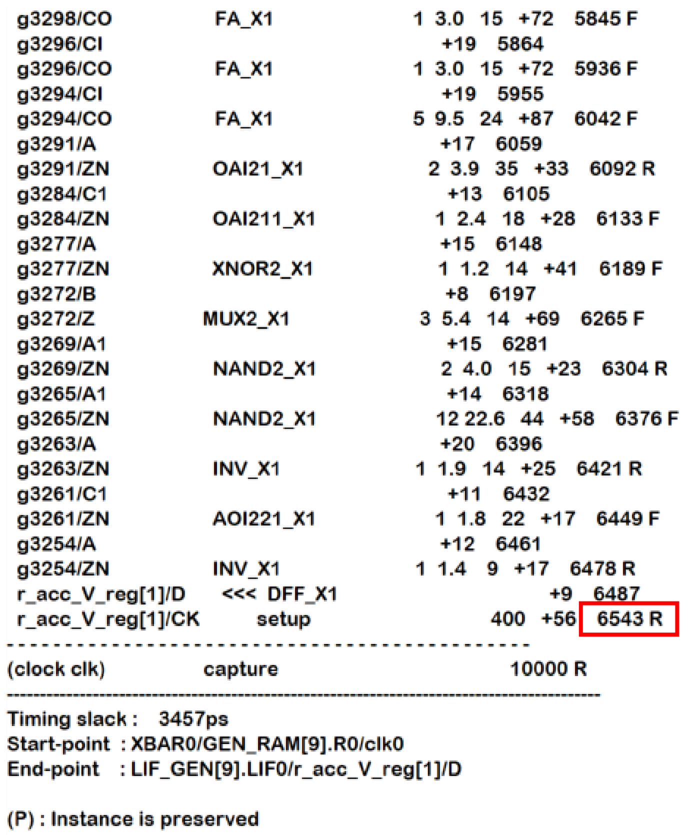
```
NOR3_X1             108   114.912   NangateOpenCellLibrary
NOR4_X1             170   226.100   NangateOpenCellLibrary
OAI211_X1            24    31.920   NangateOpenCellLibrary
OAI21_X1            136   144.704   NangateOpenCellLibrary
OAI21_X2              5     9.310   NangateOpenCellLibrary
OAI221_X1           103   164.388   NangateOpenCellLibrary
OAI222_X1             1     2.128   NangateOpenCellLibrary
OAI22_X1             25    33.250   NangateOpenCellLibrary
OR2_X1               46    48.944   NangateOpenCellLibrary
OR3_X1               43    57.190   NangateOpenCellLibrary
OR4_X1               79   126.084   NangateOpenCellLibrary
XNOR2_X1            139   221.844   NangateOpenCellLibrary
XOR2_X1             205   327.180   NangateOpenCellLibrary
sram_sp_w8_b256_freepdk45    10  43797.583  sram_sp_w8_b256_freepdk45_TT_1p1V_25C_lib
---------------------------------------------------------------------------
total                      3204  49002.671

           Library                Instances   Area   Instances %
                                                    1
---------------------------------------------------------
NangateOpenCellLibrary                 3194  5205.088     99.7
sram_sp_w8_b256_freepdk45_TT_1p1V_25C_lib  10  43797.583  2  0.3

    Type      Instances    Area    Area %
----------------------------------------
timing_model        10  43797.583   89.4
sequential         390   1764.112    3.6
inverter           548    299.250    0.6
buffer              80     63.840    0.1
logic             2176   3077.886    6.3
physical_cells       0      0.000    0.0
----------------------------------------
total             3204  49002.671  100.0
```

- The cell report provides a report of the number of instances and area of each of the library cell used for the design. 1 for the Nangate library and 2 for the SRAM library

Timing Report



```
============================================================
 Generated by:       Genus(TM) Synthesis Solution 18.13-s027_1
 Generated on:       Apr 19 2021  10:16:36 am
 Module:          SNPC
 Technology libraries:  NangateOpenCellLibrary revision 1.0
                  sram_sp_w8_b256_freepdk45_TT_1p1V_25C_lib
 Operating conditions:  typical (balanced_tree)
 Wireload mode:       enclosed
 Area mode:         timing library
============================================================

      Pin              Type         Fanout Load Slew Delay Arrival
                                           (fF) (ps)  (ps)   (ps)
--------------------------------------------------------------------
(clock clk)           launch                           5000 F
XBAR0
 GEN_RAM[9].R0/clk0                                400       5000 F
 GEN_RAM[9].R0/dout0[7] (P)  sram_sp_w8_b256_freepdk45    1  1.0    1  +305    5305 F
 drc_buf_sp58949/A                                   +6   5311
 drc_buf_sp58949/Z        CLKBUF_X1              3  5.1   15  +35    5347 F
XBAR0/o_wspike[72]
LIF_GEN[9].LIF0/i_wspike[0]
 g3323/A2                                        +11   5358
 g3323/ZN          AND2_X1                1  3.0    7  +36    5394 F
 g3305/CI                                        +19   5412
 g3305/CO          FA_X1                1  3.0   15  +69    5482 F
 g3304/CI                                        +19   5500
 g3304/CO          FA_X1                1  3.0   15  +72    5573 F
 g3302/CI                                        +19   5591
 g3302/CO          FA_X1                1  3.0   15  +72    5664 F
 g3300/CI                                        +19   5682
 g3300/CO          FA_X1                1  3.0   15  +72    5754 F
 g3298/CI                                        +19   5773
```



```
 g3298/CO           FA_X1              1  3.0   15  +72    5845 F
 g3296/CI                                        +19    5864
 g3296/CO           FA_X1              1  3.0   15  +72    5936 F
 g3294/CI                                        +19    5955
 g3294/CO           FA_X1              5  9.5   24  +87    6042 F
 g3291/A                                         +17    6059
 g3291/ZN           OAI21_X1             2  3.9   35  +33    6092 R
 g3284/C1                                        +13    6105
 g3284/ZN           OAI211_X1            1  2.4   18  +28    6133 F
 g3277/A                                         +15    6148
 g3277/ZN           XNOR2_X1             1  1.2   14  +41    6189 F
 g3272/B                                          +8    6197
 g3272/Z           MUX2_X1             3  5.4   14  +69    6265 F
 g3269/A1                                        +15    6281
 g3269/ZN           NAND2_X1              2  4.0   15  +23    6304 R
 g3265/A1                                        +14    6318
 g3265/ZN           NAND2_X1             12 22.6   44  +58    6376 F
 g3263/A                                         +20    6396
 g3263/ZN           INV_X1             1  1.9   14  +25    6421 R
 g3261/C1                                        +11    6432
 g3261/ZN           AOI221_X1            1  1.8   22  +17    6449 F
 g3254/A                                         +12    6461
 g3254/ZN           INV_X1             1  1.4    9  +17    6478 R
 r_acc_V_reg[1]/D    <<<  DFF_X1                          +9    6487
 r_acc_V_reg[1]/CK        setup                   400  +56    6543 R   1
-------------------------------------------------------------
(clock clk)           capture                          10000 R
---------------------------------------------------------------------
Timing slack :   3457ps
Start-point  : XBAR0/GEN_RAM[9].R0/clk0
End-point    : LIF_GEN[9].LIF0/r_acc_V_reg[1]/D

(P) : Instance is preserved
```

- The timing report provides a report of the validation of timing performance of the SNPC. This validation is done by checking all possible paths for timing violations. 1 shows the period of the SNPC.

Phase 4:
Gate Level Simulation with Modelsim

Contents

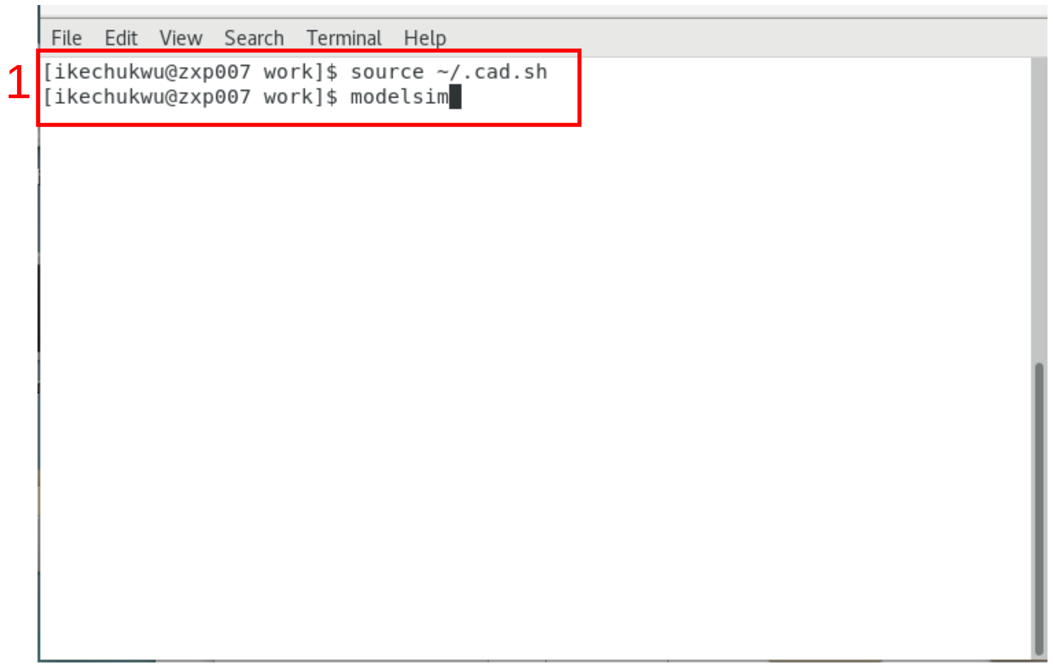
- Gate level simulation directory structure
- Environment and Modelsim simulation

Gate Level Simulation with Modelsim

- This is basically the same step as RTL simulation, only here the synthesized SNPC source is simulated together with a SDF (standard delay format) file. The synthesized SNPC source (netlist) is a structural description of the design, with all the standard cells and macro's (memory blocks) connected to each other in the best possible way, for a functional and fastest design. The SDF file contains of delay values for the used standard cells, and the interconnect delays are guessed, because Design Compiler doesn't contain any interconnect information. These delay values are annotated into the synthesized design and simulated.

- For this simulation, the same setup and MNIST dataset for RTL functional simulation is used.

Environment setup

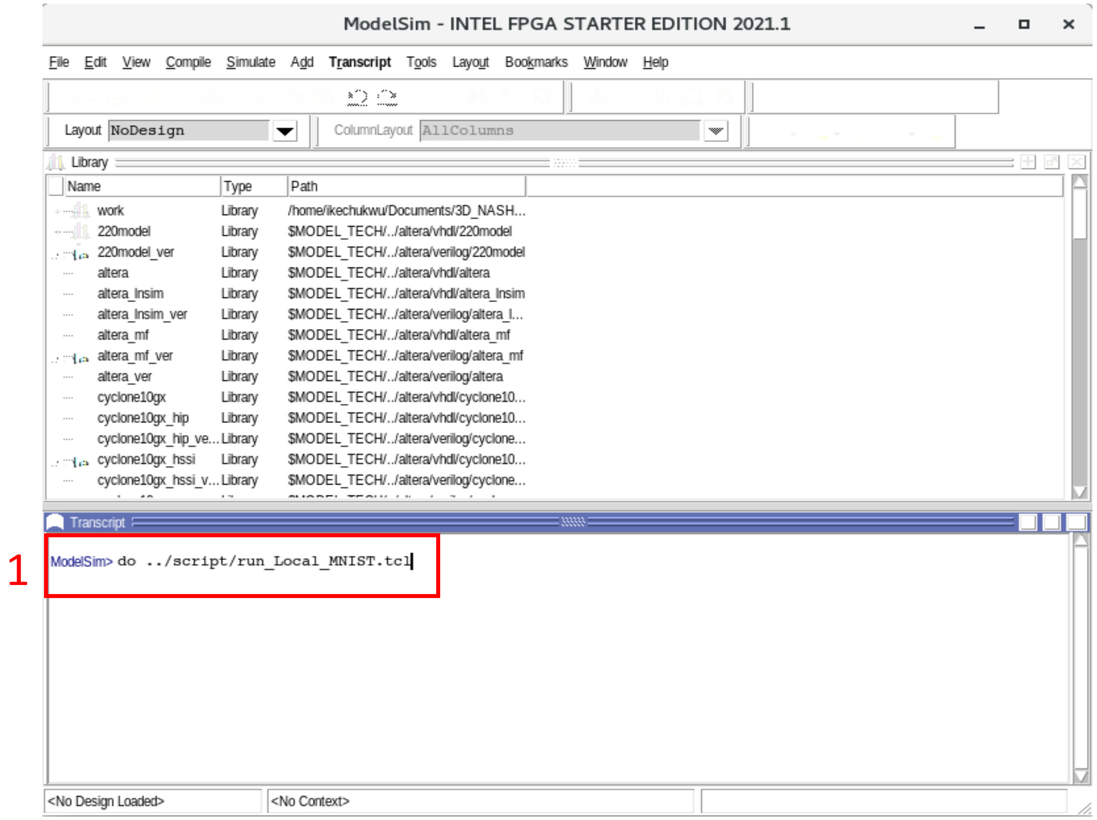


```
[ikechukwu@zxp007 work]$ source ~/.cad.sh
[ikechukwu@zxp007 work]$ modelsim
```

(1) To set the functional simulator working environment, from the terminal, navigate to **~/SNPC/GL_SIM/work/ .** If you are using file explorer, go to the same directory, right click and select **Open in Terminal**. Use the following command to set the environment and launch modelsim.

source ~/.cad.sh
modelsim"

Begin Gate Level Simulation


ModelSim - INTEL FPGA STARTER EDITION 2021.1
File Edit View Compile Simulate Add Transcript Tools Layout Bookmarks Window Help
Layout NoDesign
ColumnLayout AllColumns
Library
Transcript
1
<No Design Loaded>
<No Context>

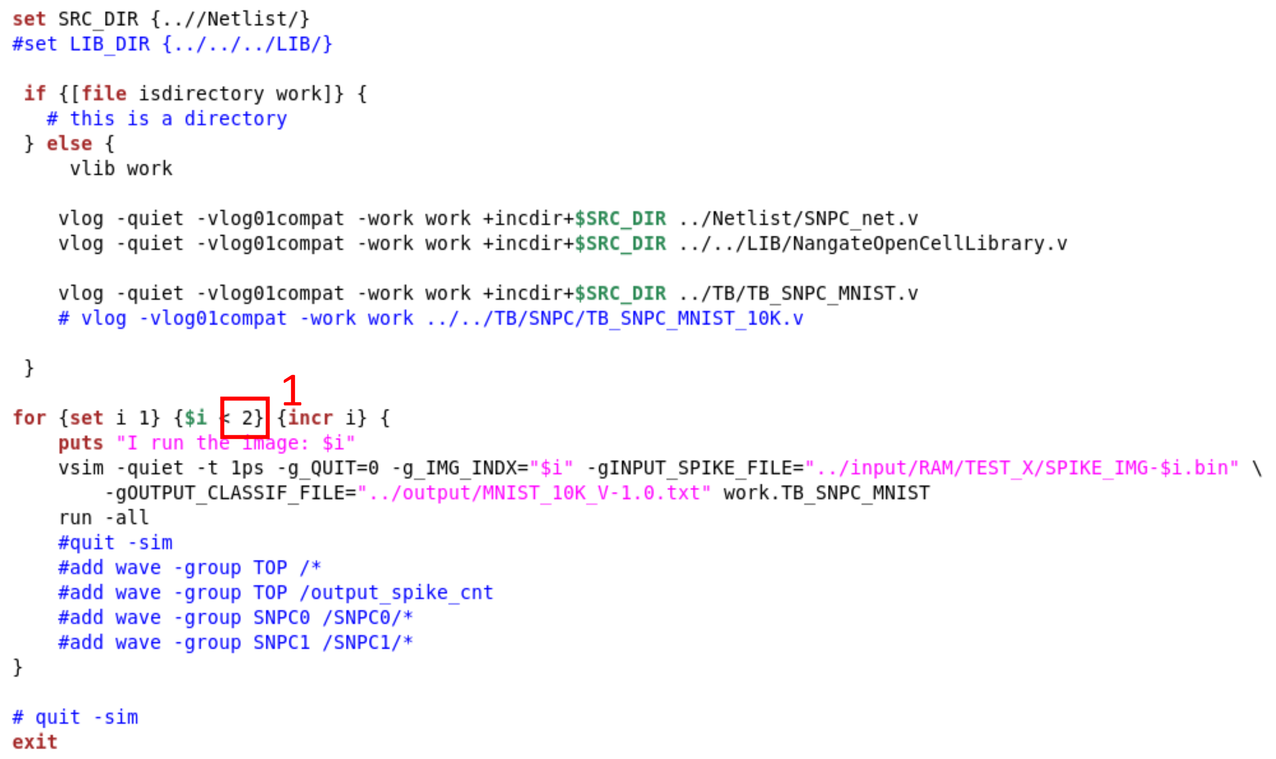
Name	Type	Path
work	Library	/home/ikechukwu/Documents/3D_NASH...
220model	Library	$MODEL_TECH/../altera/vhdl/220model
220model_ver	Library	$MODEL_TECH/../altera/verilog/220model
altera	Library	$MODEL_TECH/../altera/vhdl/altera
altera_lnsim	Library	$MODEL_TECH/../altera/vhdl/altera_lnsim
altera_lnsim_ver	Library	$MODEL_TECH/../altera/verilog/altera_l...
altera_mf	Library	$MODEL_TECH/../altera/vhdl/altera_mf
altera_mf_ver	Library	$MODEL_TECH/../altera/verilog/altera_mf
altera_ver	Library	$MODEL_TECH/../altera/verilog/altera
cyclone10gx	Library	$MODEL_TECH/../altera/vhdl/cyclone10...
cyclone10gx_hip	Library	$MODEL_TECH/../altera/vhdl/cyclone10...
cyclone10gx_hip_ve...	Library	$MODEL_TECH/../altera/verilog/cyclone...
cyclone10gx_hssi	Library	$MODEL_TECH/../altera/vhdl/cyclone10...
cyclone10gx_hssi_v...	Library	$MODEL_TECH/../altera/verilog/cyclone...

```
ModelSim> do ../script/run_Local_MNIST.tcl
```

(1) After launching modelsim, use the following command to run the simulation.

do ../script/run_Local_MNIST.tcl

Modify Gate Level Simulation Script



```
set SRC_DIR {..//Netlist/}
#set LIB_DIR {../../../LIB/}

 if {[file isdirectory work]} {
   # this is a directory
 } else {
     vlib work

    vlog -quiet -vlog01compat -work work +incdir+$SRC_DIR ../Netlist/SNPC_net.v
    vlog -quiet -vlog01compat -work work +incdir+$SRC_DIR ../../LIB/NangateOpenCellLibrary.v

    vlog -quiet -vlog01compat -work work +incdir+$SRC_DIR ../TB/TB_SNPC_MNIST.v
    # vlog -vlog01compat -work work ../../TB/SNPC/TB_SNPC_MNIST_10K.v

 }

for {set i 1} {$i < 2} {incr i} {
    puts "I run the image: $i"
    vsim -quiet -t 1ps -g_QUIT=0 -g_IMG_INDX="$i" -gINPUT_SPIKE_FILE="../input/RAM/TEST_X/SPIKE_IMG-$i.bin" \
        -gOUTPUT_CLASSIF_FILE="../output/MNIST_10K_V-1.0.txt" work.TB_SNPC_MNIST
    run -all
    #quit -sim
    #add wave -group TOP /*
    #add wave -group TOP /output_spike_cnt
    #add wave -group SNPC0 /SNPC0/*
    #add wave -group SNPC1 /SNPC1/*
}

# quit -sim
exit
```

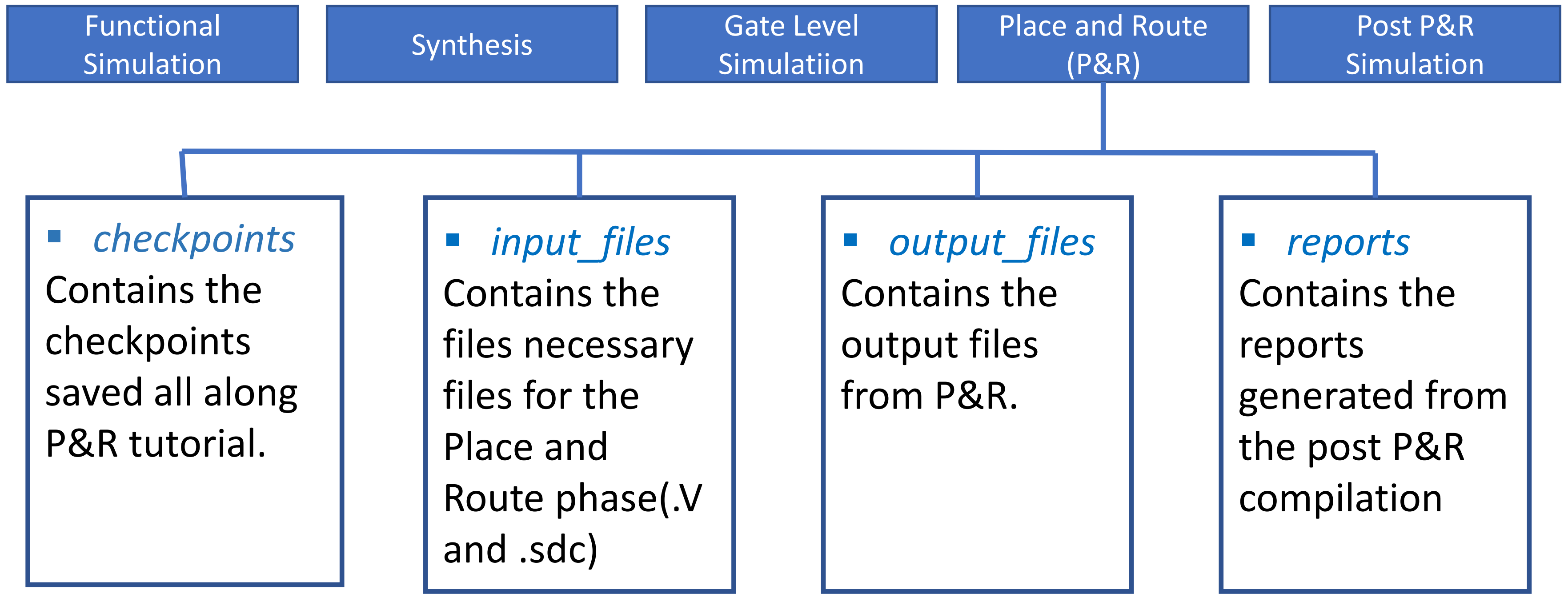
(1) Note: The MNIST simulation runs for 2 images by default. To change this number, open ***~/SNPC/GL_SIM/script/ run_Local_MNIST.tcl*** in text editor and the number of images can be changed to a maximum of 10,000. The simulation output is stored in ***~/SNPC/GL_SIM/output***

Phase 5:
Place & Route with
Cadence Innovus

Contents

- Place & Route (P&R) directory structure
- Environment
- Innovus P&R steps (Batch mode)
- Innovus P&R steps (GUI mode)

Place and Route directory structure



Functional Simulation | Synthesis | Gate Level Simulatiion | Place and Route (P&R) | Post P&R Simulation

- *checkpoints*
  Contains the checkpoints saved all along P&R tutorial.

- *input_files*
  Contains the files necessary files for the Place and Route phase(.V and .sdc)

- *output_files*
  Contains the output files from P&R.

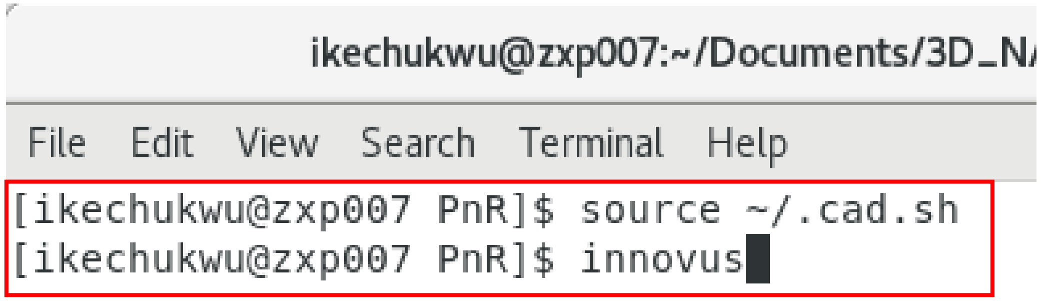
- *reports*
  Contains the reports generated from the post P&R compilation

Place and Route

- The place and route process places each macro from the synthesis netlist into an available location on the target silicon and connects the macros using routing resources available on the target silicon and technology. The place and route tools read the netlist, extract the components and nets from the netlist, place the components on the target device, and interconnect the components using the specified interconnections.

- To perform the place and route, copy the files generated from the netlist from ***~/SNPC/SYNTH/RTL_SYNTH/output_files/*** to ***~/SNPC/PnR/input_files/.*** The scripts for the place and place and route scripts are located in the ***/SNPC/PnR/scripts/*** directory. Two of the scripts *".global"* and *".view"* are used to set the parameters (libraries, constraints, netlist) and conditions for the place and route. The next five scripts which their names begin with indexes from 0 to 5 are all scripts for various stage of the place and route stages: initialize, floor plan, place macro, place standard cell, route, and finish. The final script without index is used to call the other scripts during the place and route.

Place & Route (Batch Mode) (1/3)

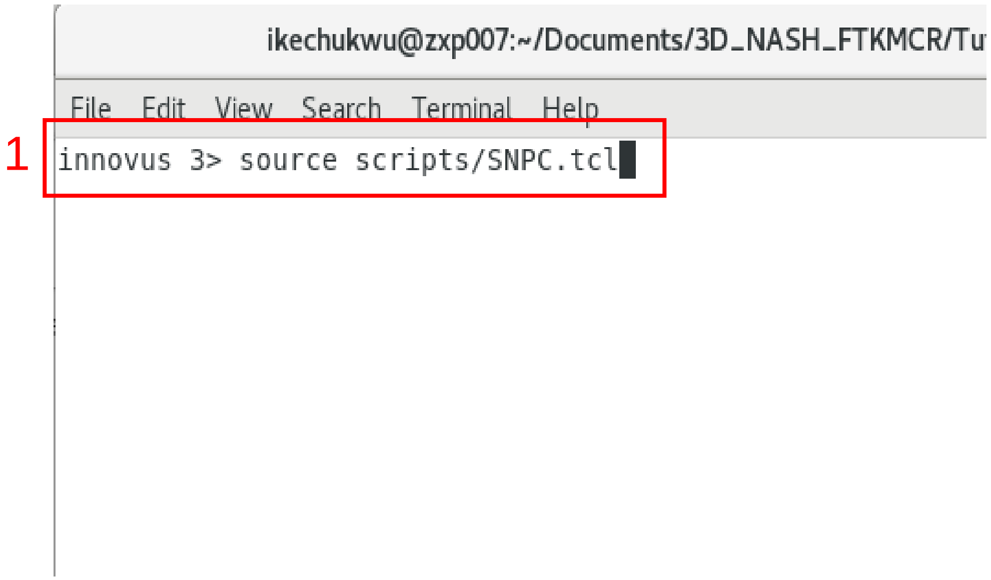

ikechukwu@zxp007:~/Documents/3D_N
File Edit View Search Terminal Help
1

```
[ikechukwu@zxp007 PnR]$ source ~/.cad.sh
[ikechukwu@zxp007 PnR]$ innovus
```

(1) To begin the place and route, navigate to ***~/SNPC/PnR/*** on terminal and type the following commands to setup the environment and launch the innovus tool:

source ~/.cad.sh
innovus

Place & Route (Batch Mode) (2/3)

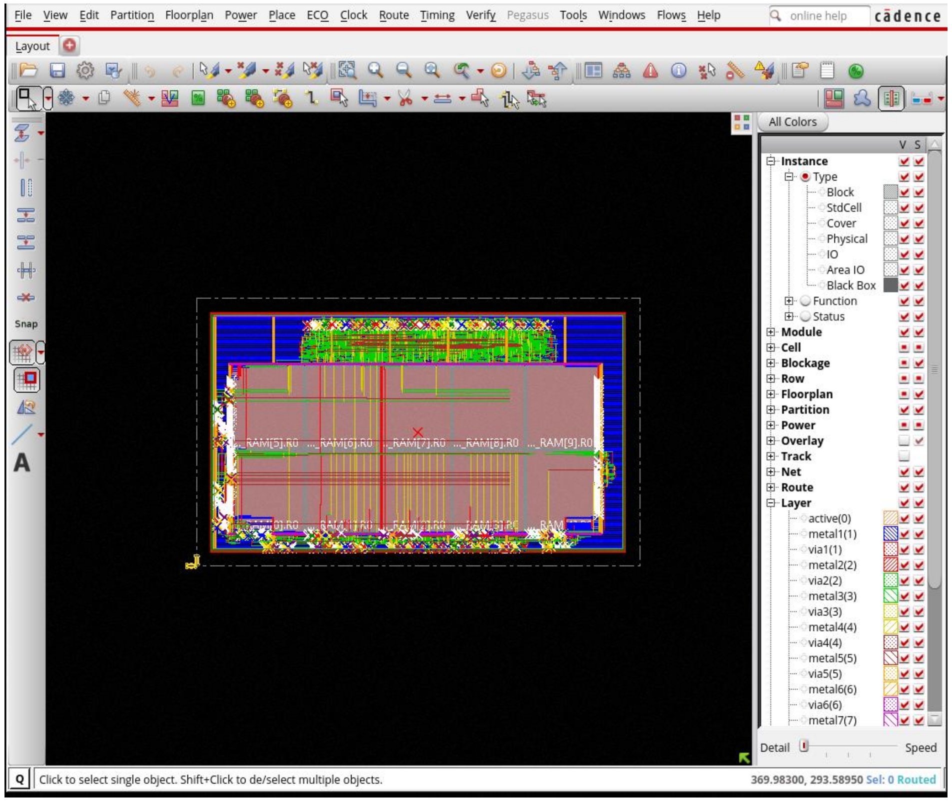

ikechukwu@zxp007:~/Documents/3D_NASH_FTKMCR/Tu
File Edit View Search Terminal Help
1

```
innovus 3> source scripts/SNPC.tcl
```

(1) After the innovus tool has been launched the graphical user interface will also be launched. However, type the following command on the terminal to launch the place and route script.   *source scripts/SNPC.tcl*

Note: The place and route will take some time to complete because the design is somewhat large.

Place & Route (Batch Mode) (3/3)

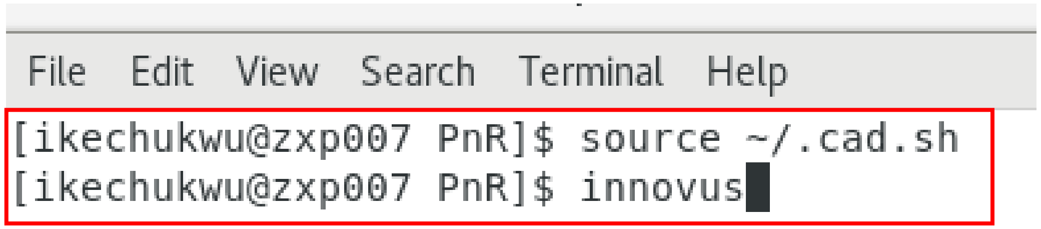


Click to select single object. Shift+Click to de/select multiple objects.

369.98300, 293.58950 Sel: 0 Routed

(1) The image above is the output of the place and route, along with other outputs and reports in the ***~/SNPC/P&R/output_files/*** and ***~/SNPC/P&R/reports/*** directories respectively.

Place & Route (GUI Mode)



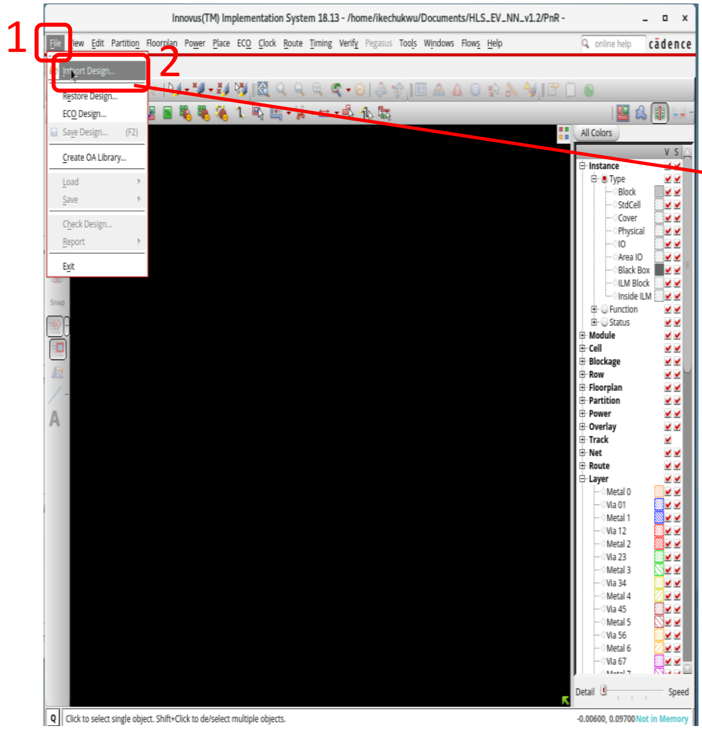
1

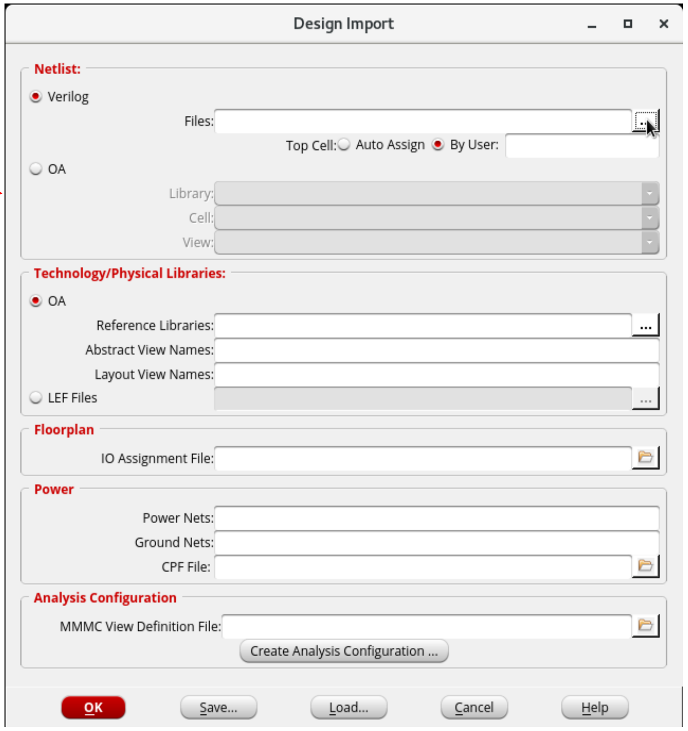
```
[ikechukwu@zxp007 PnR]$ source ~/.cad.sh
[ikechukwu@zxp007 PnR]$ innovus
```

(1) To begin the place and route, navigate to ***~/SNPC/PnR/*** on terminal and type the following commands to setup the environment and launch the innovus tool:

source ~/.cad.sh
innovus

Step 1: Import Design





Design Import

Netlist:

[● Verilog] ○ OA

Files:

Top Cell: ○ Auto Assign [● By User:]

Library:

Cell:

View:

Technology/Physical Libraries:

[● OA] ○ LEF Files

Reference Libraries:

Abstract View Names:

Layout View Names:

Floorplan

IO Assignment File:

Power

Power Nets:

Ground Nets:

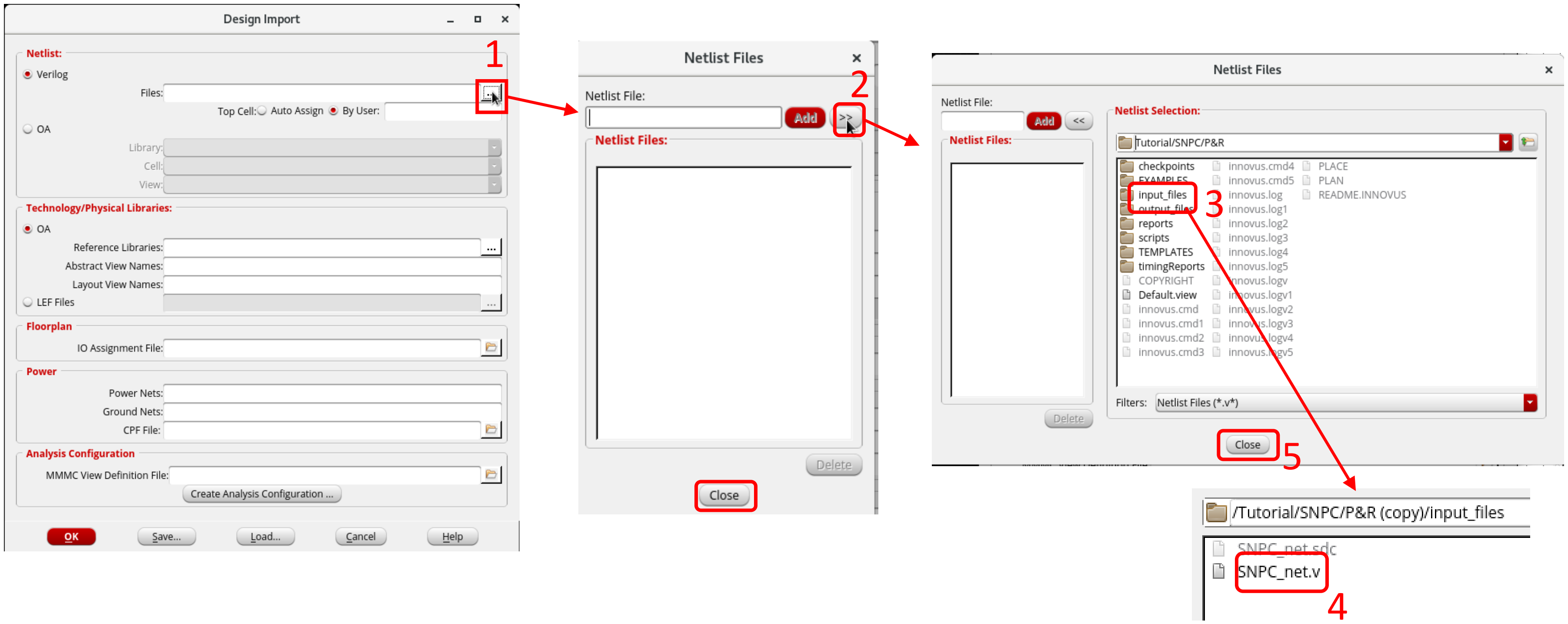
CPF File:

Analysis Configuration

MMMC View Definition File:

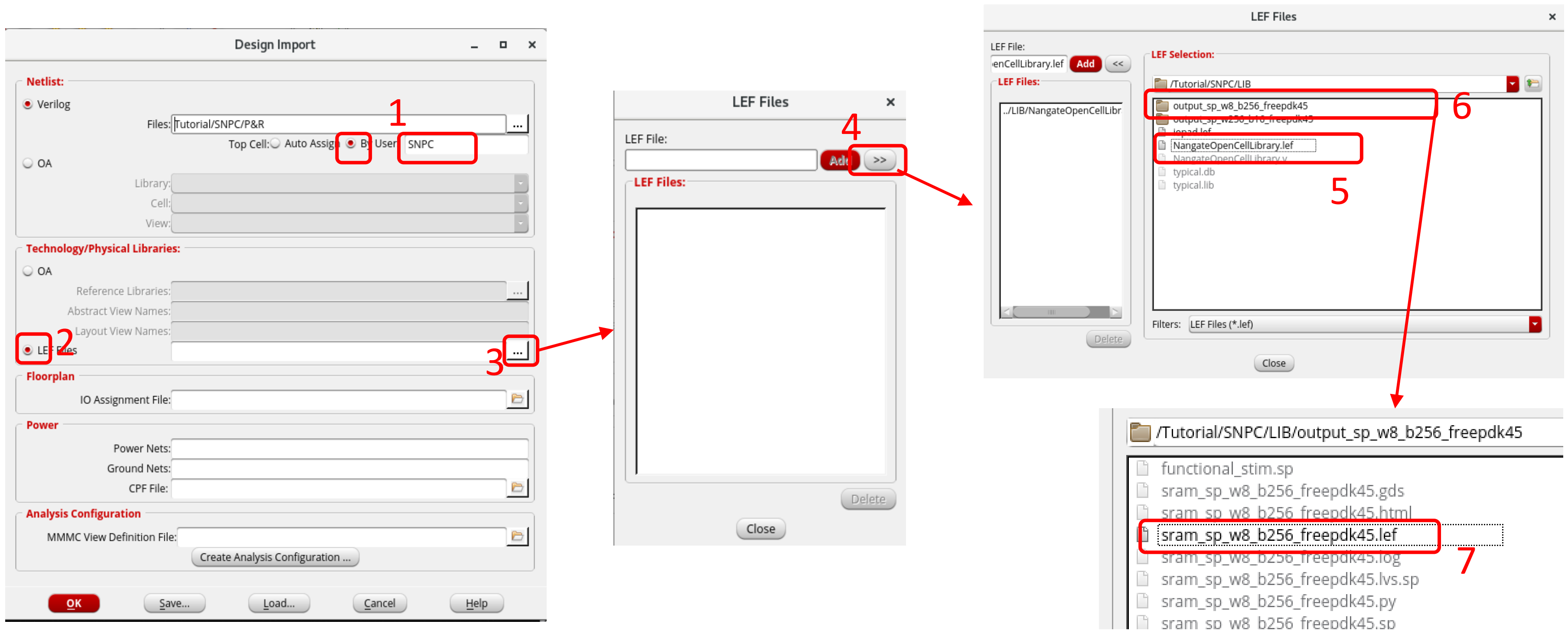
(1) Click on **File** -> (2) **Import Design**

Import design- Netlist File



(1) Click **Files** to import the netlist. (2) Click on **>>** to expand. (3) Go to **./input_files** folder. (4) Double click on **SNPC_net.v**. (5) Click **Close** until you get to **Design import** window.

Import design- LEF Files



Design Import

Netlist:

[✓ Verilog]

Files: Tutorial/SNPC/P&R

Top Cell: Auto Assign [✓ By User] SNPC

OA

Library:

Cell:

View:

Technology/Physical Libraries:

OA

Reference Libraries:

Abstract View Names:

Layout View Names:

[✓ LEF Files]

Floorplan

IO Assignment File:

Power

Power Nets:

Ground Nets:

CPF File:

Analysis Configuration

MMMC View Definition File:

LEF Files

LEF File:

LEF Files:

LEF Files

LEF File: enCellLibrary.lef

LEF Files: ../LIB/NangateOpenCellLibr

LEF Selection: /Tutorial/SNPC/LIB

output_sp_w8_b256_freepdk45
output_sp_w256_b16_freepdk45
iopad.lef
NangateOpenCellLibrary.lef
NangateOpenCellLibrary.v
typical.db
typical.lib

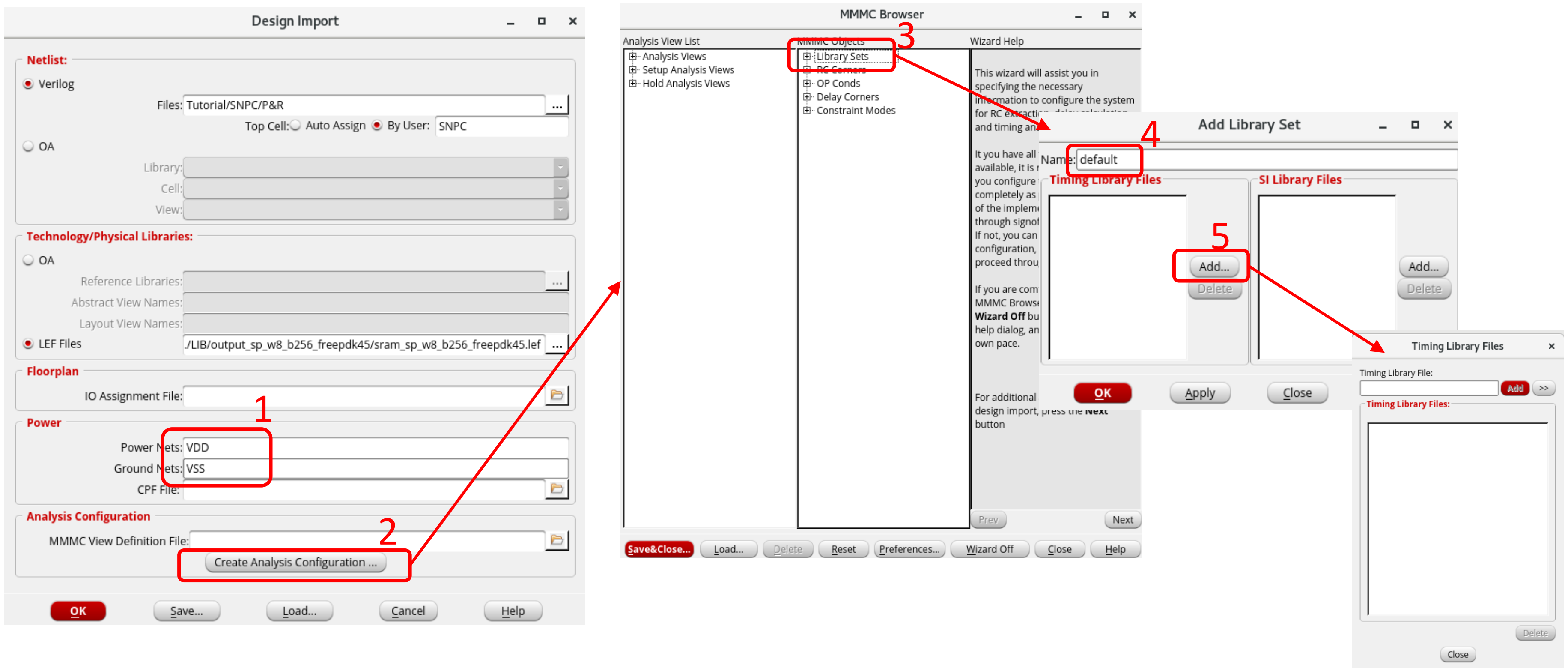
Filters: LEF Files (*.lef)

/Tutorial/SNPC/LIB/output_sp_w8_b256_freepdk45

functional_stim.sp
sram_sp_w8_b256_freepdk45.gds
sram_sp_w8_b256_freepdk45.html
sram_sp_w8_b256_freepdk45.lef
sram_sp_w8_b256_freepdk45.log
sram_sp_w8_b256_freepdk45.lvs.sp
sram_sp_w8_b256_freepdk45.py
sram_sp_w8_b256_freepdk45.sp

(1) In Top Cell: type **SNPC.** (2) Click on **LEF files.** (3) Click on **...** to add LEF files (4) Click on **>>** to expand and go to **~/SNPC/LIB** folder. (5) Double click on **NangateOpenCellLibrary.lef** and **sram_sp_w8_b256_freepdk45.lef** (6,7) Click **Close** until you get to the **Design import** window

Import design- Analysis Setup (1/9)



Design Import

Netlist:

[✓ Verilog] OA

Files: Tutorial/SNPC/P&R

Top Cell: Auto Assign [✓ By User:] SNPC

Technology/Physical Libraries:

OA [✓ LEF Files]

LEF Files: ./LIB/output_sp_w8_b256_freepdk45/sram_sp_w8_b256_freepdk45.lef

Floorplan

IO Assignment File:

Power

Power Nets: VDD

Ground Nets: VSS

CPF File:

Analysis Configuration

MMMC View Definition File:

MMMC Browser

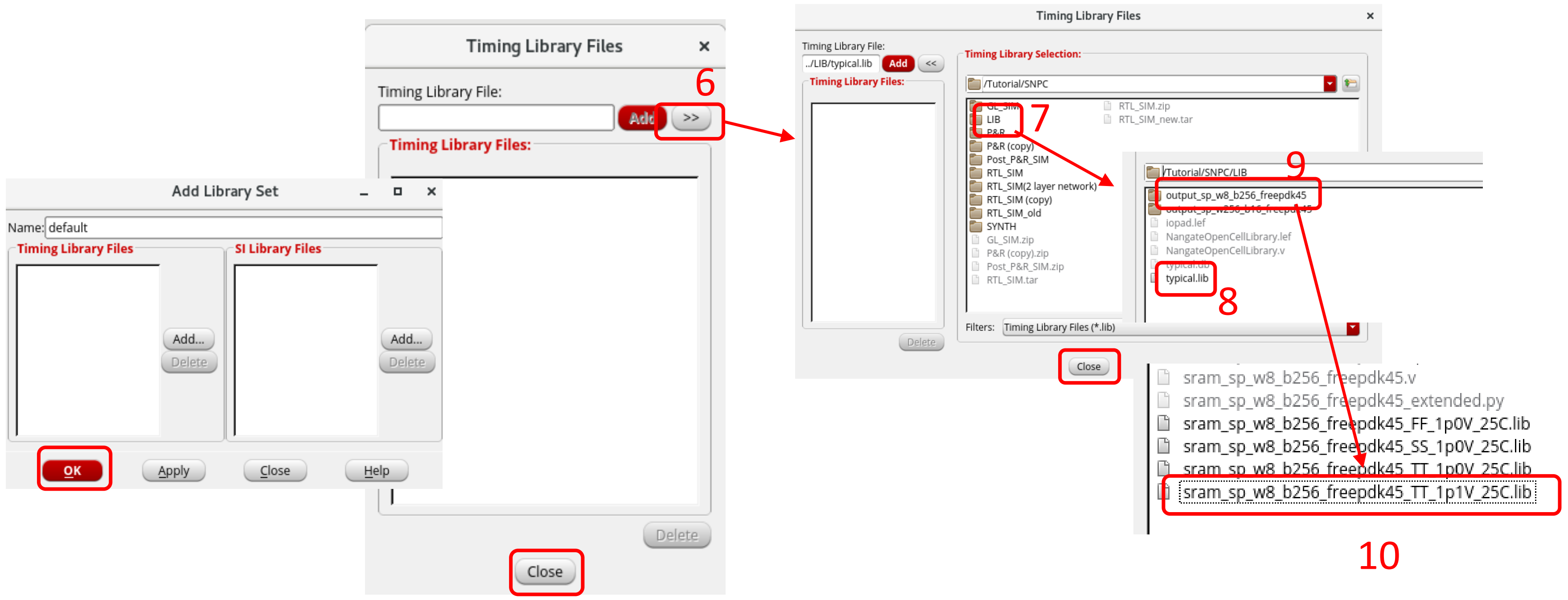
Add Library Set

Name: default

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

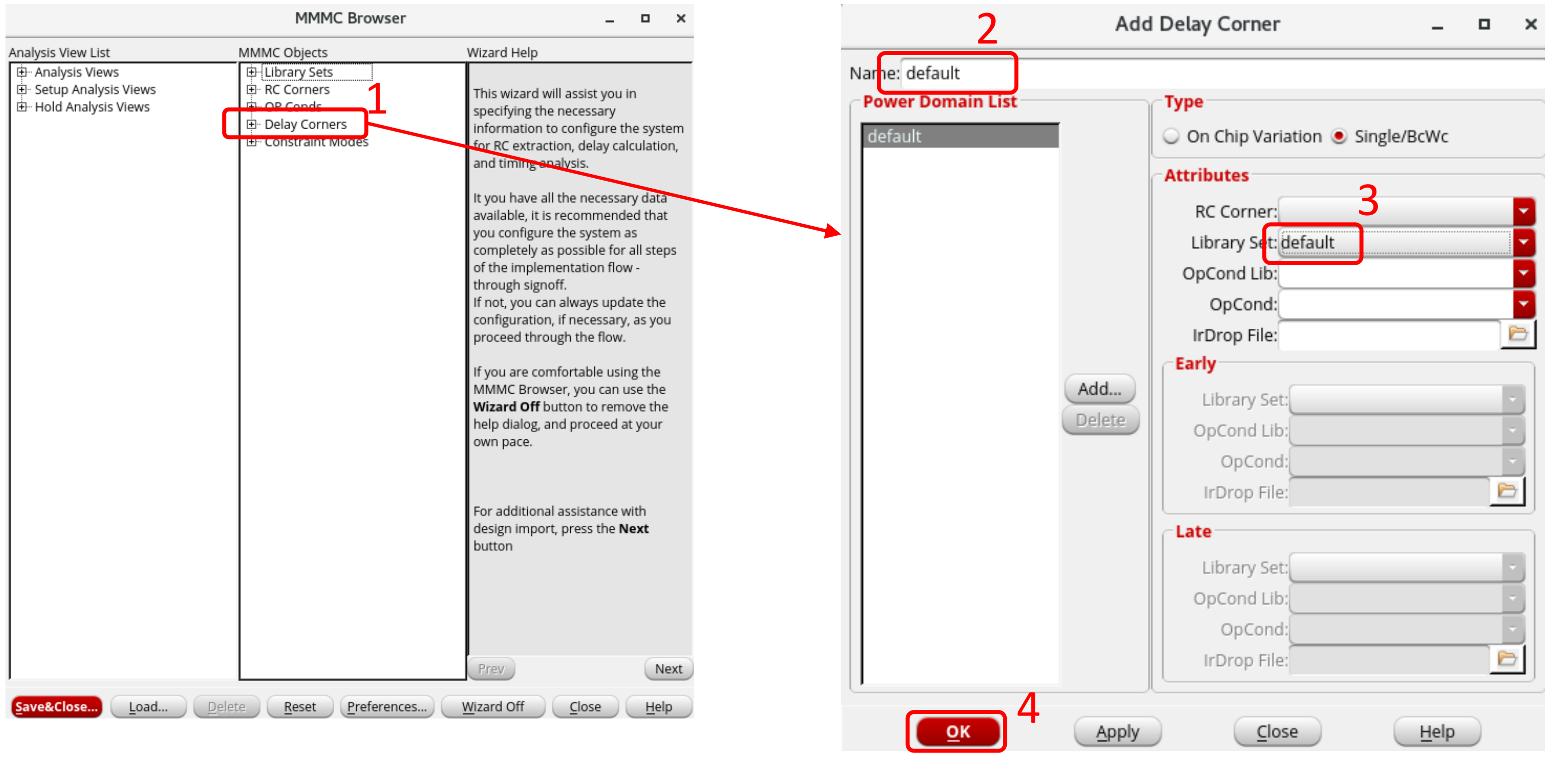
(1) Click on **Create Analysis Configuration**. (2) Double click on **Library Sets** in the **MMMC Browser window** (3) On the add *Library Set* window, type **default** in *Name* (4) Click on **Add**.

Import design- Analysis Setup (2/9)



(7) In the *Timing Library* Window, go to **~/SNPC/LIB** and (8) select **typical.lib.** (9,10) select **Sram_sp_w8_b256_freepdk45_TT_1p1V_25C.lib.** Click **close**(*Timing Library Files*) and then **OK**(Add *Library Set*)

Import design- Analysis Setup (3/9)



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

It you have all the necessary data available, it is recommended that you configure the system as completely as possible for all steps of the implementation flow - through signoff.
If not, you can always update the configuration, if necessary, as you proceed through the flow.

If you are comfortable using the MMMC Browser, you can use the **Wizard Off** button to remove the help dialog, and proceed at your own pace.

For additional assistance with design import, press the **Next** button

Name: default

Power Domain List: default

Type: On Chip Variation [✓ Single/BcWc]

Attributes

RC Corner:
Library Set: default
OpCond Lib:
OpCond:
IrDrop File:

Early

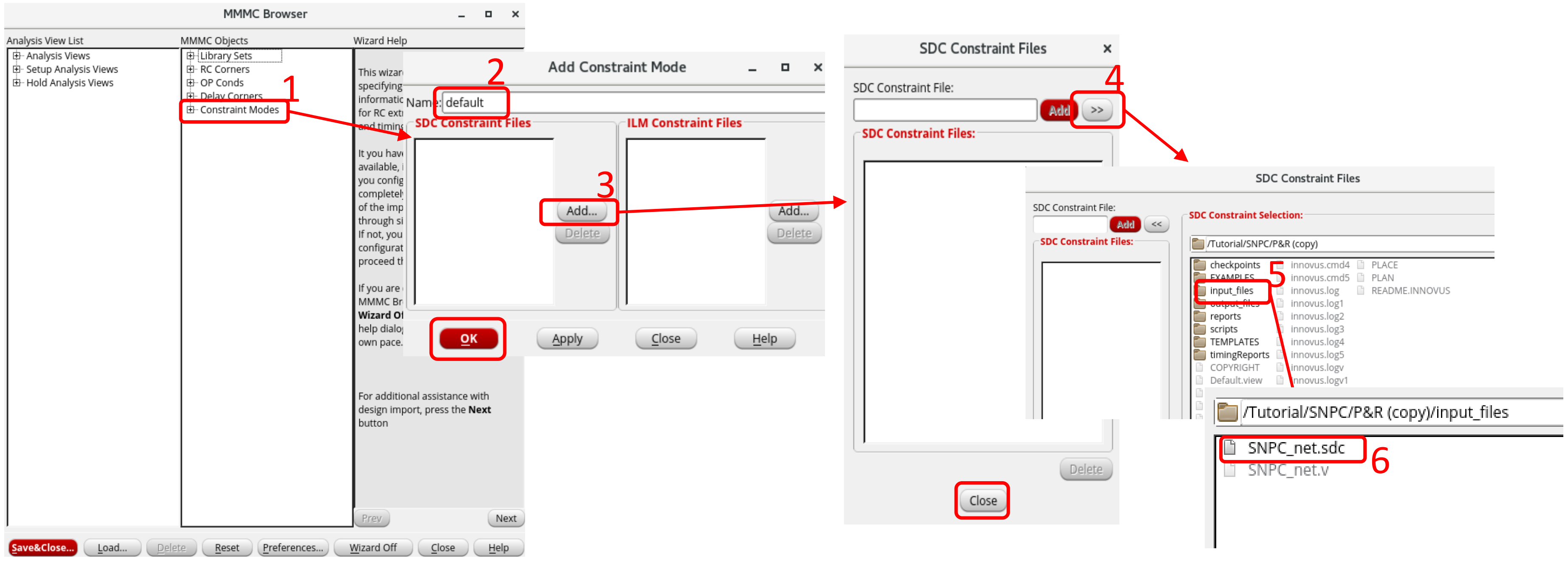
Library Set:
OpCond Lib:
OpCond:
IrDrop File:

Late

Library Set:
OpCond Lib:
OpCond:
IrDrop File:

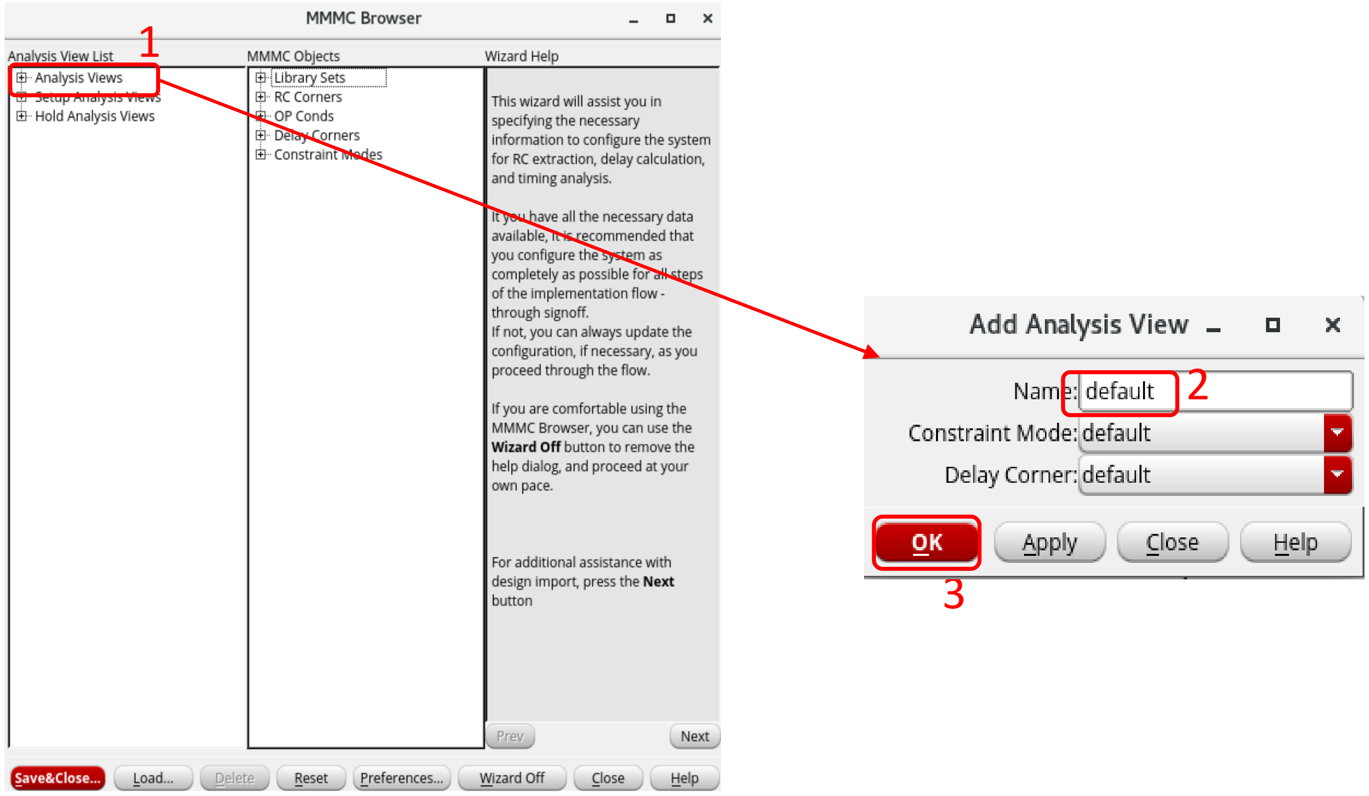
(1) Double click on **Delay Corners** in the *MMMC browser* window. (2) On the *Add Delay Corner* window, type **default** in *Name*. (3) Change the *Library Set* to **default**. (4) Click **OK**

Import design- Analysis Setup (4/9)



(1) Double click on ***Constraint Modes*** in the *MMMC browser* window. (2) On the *Add Constraint Mode* window, type **default** in *Name.* (3) Click on **Add**. (4) In the *SDC Constraint File* window, CLICK **>> .** (5,6) Go to **./input_files** and select **SNPC_net.sdc**. Click **Close**(*SDC Constraint File* window) and then **Ok** (*Add Constraint Mode* window).

Import design- Analysis Setup (5/9)



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

If you have all the necessary data available, it is recommended that you configure the system as completely as possible for all steps of the implementation flow - through signoff.
If not, you can always update the configuration, if necessary, as you proceed through the flow.

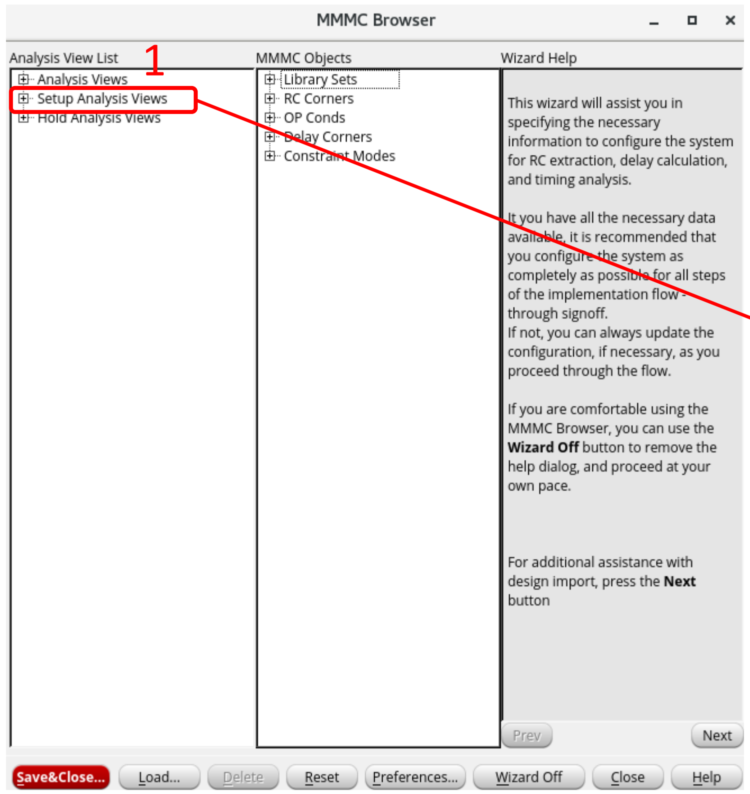
If you are comfortable using the MMMC Browser, you can use the **Wizard Off** button to remove the help dialog, and proceed at your own pace.

For additional assistance with design import, press the **Next** button

Name: default
Constraint Mode: default
Delay Corner: default

(1) Double click on **Analysis Views** in the *MMMC browser* window
(2) On the ***Add Analysis View*** window, type **default** in *Name*
(3) Click **OK**

Import design- Analysis Setup (6/9)



MMMC Browser

Analysis View List

- Analysis Views
- Setup Analysis Views
- Hold Analysis Views

MMMC Objects

- Library Sets
- RC Corners
- OP Conds
- Delay Corners
- Constraint Modes

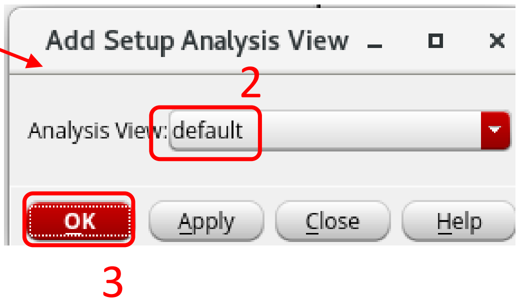
Wizard Help

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

If you have all the necessary data available, it is recommended that you configure the system as completely as possible for all steps of the implementation flow - through signoff.
If not, you can always update the configuration, if necessary, as you proceed through the flow.

If you are comfortable using the MMMC Browser, you can use the **Wizard Off** button to remove the help dialog, and proceed at your own pace.

For additional assistance with design import, press the **Next** button

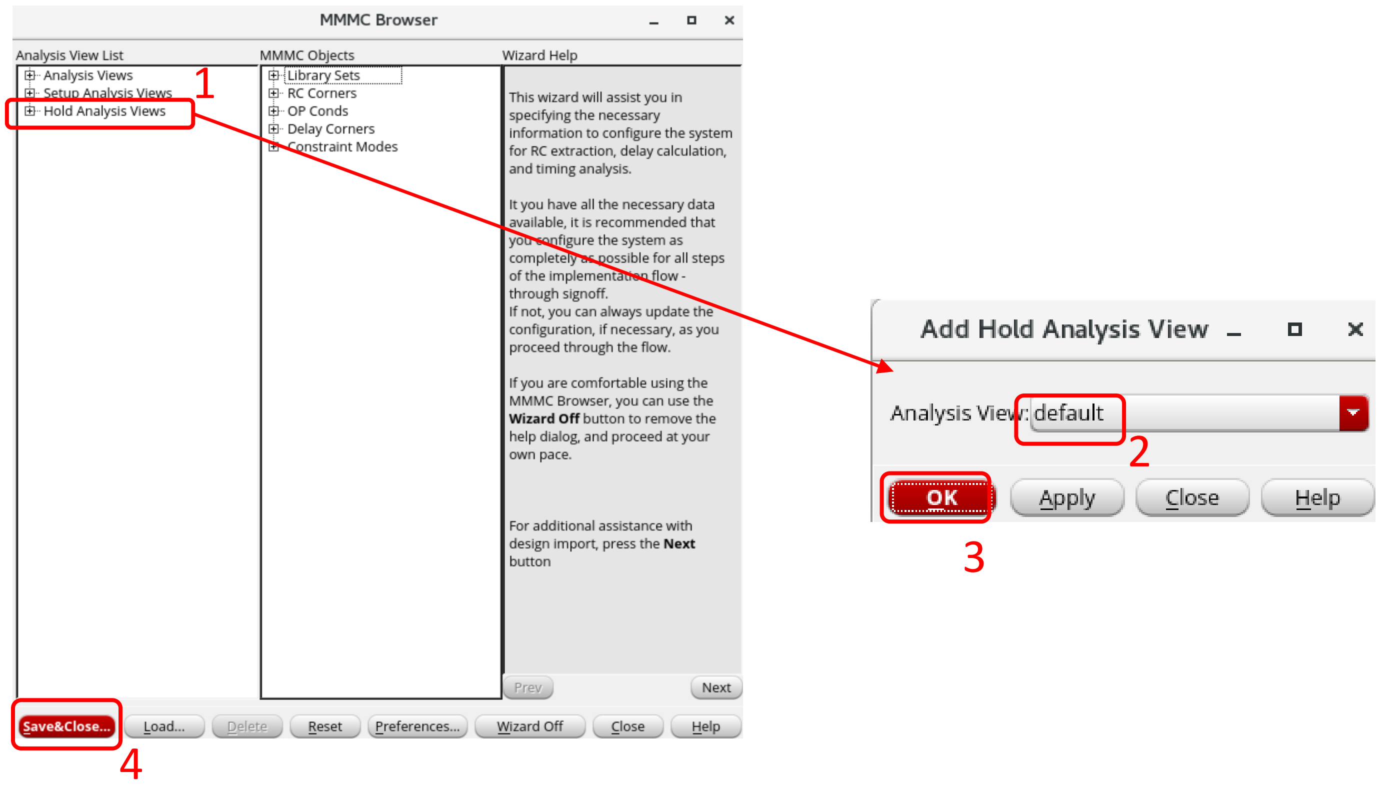


Add Setup Analysis View

Analysis View: default

(1) Double click on **Analysis Views** in the ***MMMC browser*** window
(2) On the ***Add Setup Analysis View*** window, type **default** in *Name*
(3) Click **OK**

Import design- Analysis Setup (7/9)



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

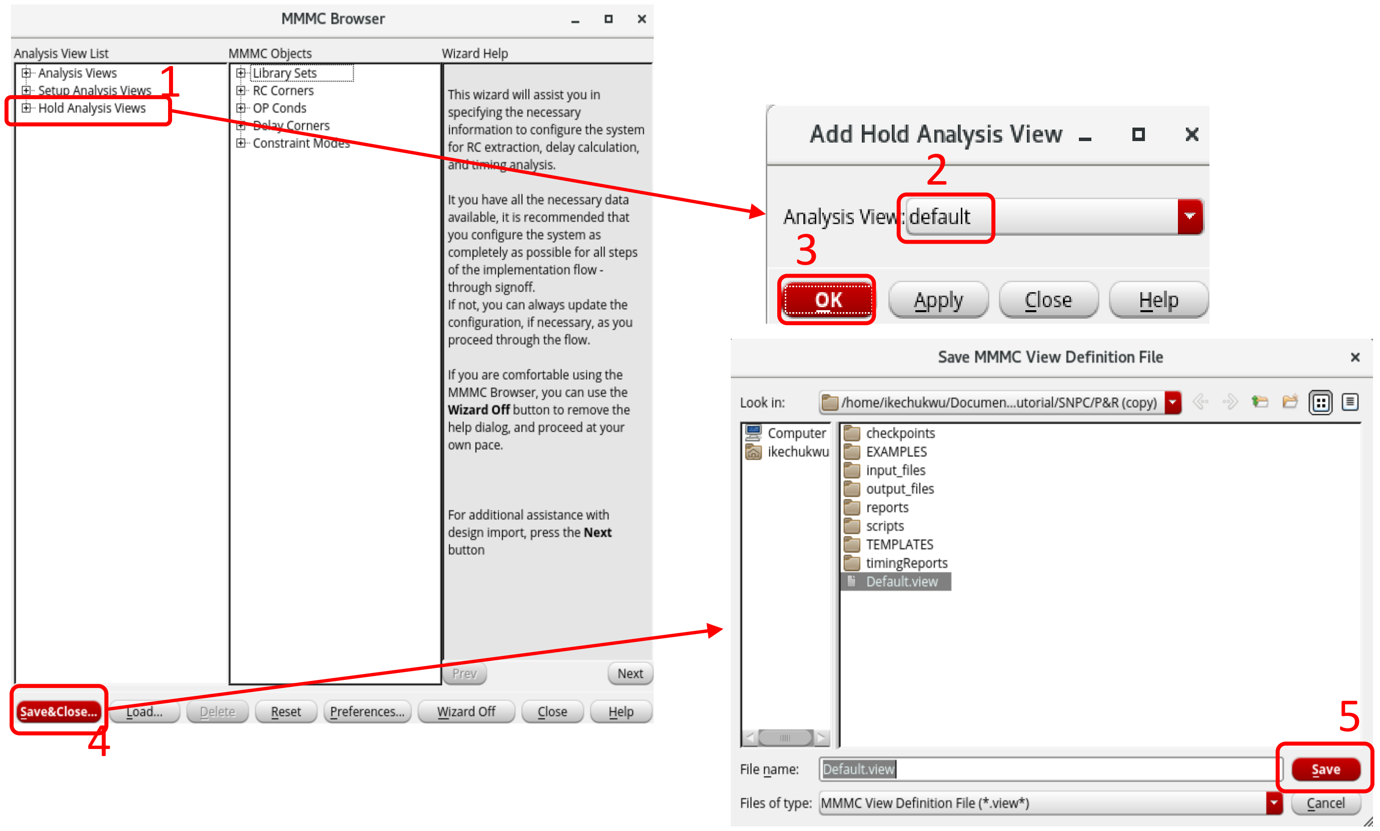
It you have all the necessary data available, it is recommended that you configure the system as completely as possible for all steps of the implementation flow - through signoff.
If not, you can always update the configuration, if necessary, as you proceed through the flow.

If you are comfortable using the MMMC Browser, you can use the **Wizard Off** button to remove the help dialog, and proceed at your own pace.

For additional assistance with design import, press the **Next** button

(1) Double click on **Setup Analysis Views** in the ***MMMC browser*** window
(2) In the ***Add Setup Analysis View*** window, make sure that **Analysis View** is set to **default** (3) Click **OK.** (4) Click **save&close**.

Import design- Analysis Setup (8/9)



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

It you have all the necessary data available, it is recommended that you configure the system as completely as possible for all steps of the implementation flow - through signoff.
If not, you can always update the configuration, if necessary, as you proceed through the flow.

If you are comfortable using the MMMC Browser, you can use the **Wizard Off** button to remove the help dialog, and proceed at your own pace.

For additional assistance with design import, press the **Next** button

Analysis View: default

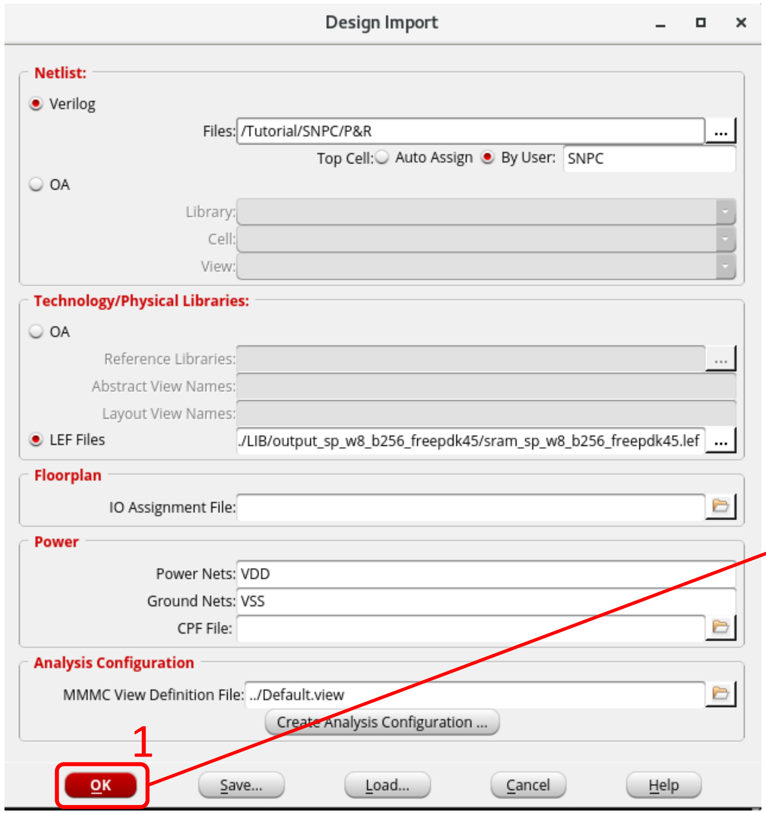
Look in: /home/ikechukwu/Documen...utorial/SNPC/P&R (copy)

File name: Default.view

Files of type: MMMC View Definition File (*.view*)

(1) Double click on **Hold Analysis Views** in the *MMMC browser* window. (2) In the *Add Hold Analysis View* Window, make sure that *Analysis View* is set to **default.** (3) Click **OK.** (4) Click on **Save&Close…** in the *MMMC browser* window. (5) Click **Save**

Import design- Analysis Setup (9/9)



Design Import

Netlist:

[● Verilog]

Files: /Tutorial/SNPC/P&R

Top Cell: ○ Auto Assign [● By User:] SNPC

○ OA

Library:

Cell:

View:

Technology/Physical Libraries:

○ OA

Reference Libraries:

Abstract View Names:

Layout View Names:

[● LEF Files] ./LIB/output_sp_w8_b256_freepdk45/sram_sp_w8_b256_freepdk45.lef

Floorplan

IO Assignment File:

Power

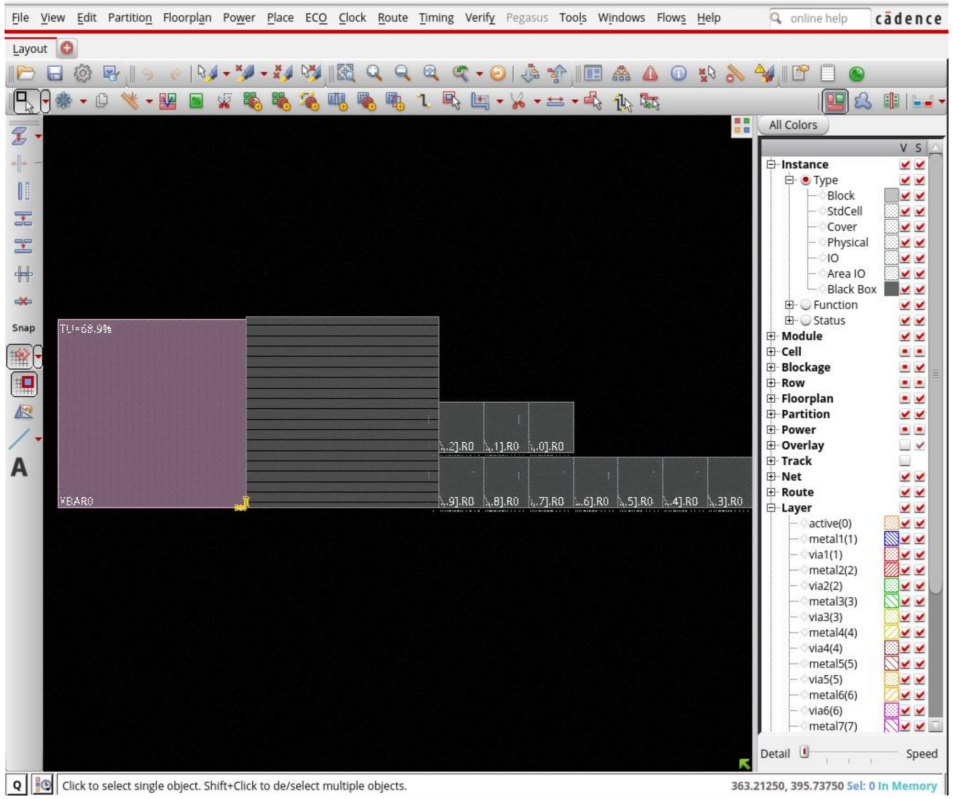
Power Nets: VDD

Ground Nets: VSS

CPF File:

Analysis Configuration

MMMC View Definition File: ../Default.view

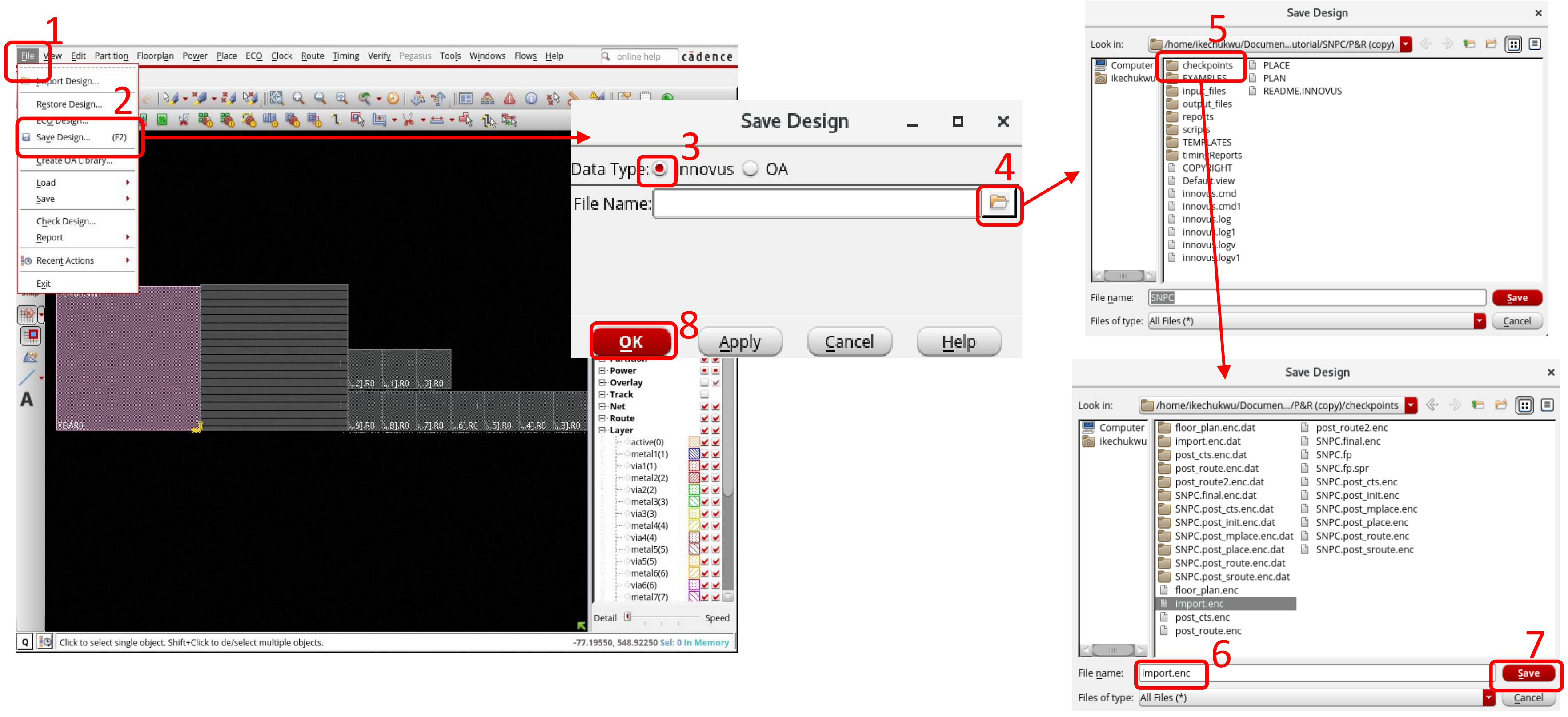


Click to select single object. Shift+Click to de/select multiple objects.

363.21250, 395.73750 Sel: 0 In Memory

(1) Click **Ok**

Import Design- Checkpoint



Save Design

Data Type: [✓ Innovus] OA

File Name:

Save Design

Look in: /home/ikechukwu/Documen...utorial/SNPC/P&R (copy)

Computer
ikechukwu

checkpoints
EXAMPLES
input_files
output_files
reports
scripts
TEMPLATES
timingReports
COPYRIGHT
Default.view
innovus.cmd
innovus.cmd1
innovus.log
innovus.log1
innovus.logv
innovus.logv1
PLACE
PLAN
README.INNOVUS

File name: SNPC

Files of type: All Files (*)

Save Design

Look in: /home/ikechukwu/Documen.../P&R (copy)/checkpoints

Computer
ikechukwu

floor_plan.enc.dat
import.enc.dat
post_cts.enc.dat
post_route.enc.dat
post_route2.enc.dat
SNPC.final.enc.dat
SNPC.post_cts.enc.dat
SNPC.post_init.enc.dat
SNPC.post_mplace.enc.dat
SNPC.post_place.enc.dat
SNPC.post_route.enc.dat
SNPC.post_sroute.enc.dat
floor_plan.enc
import.enc
post_cts.enc
post_route.enc
post_route2.enc
SNPC.final.enc
SNPC.fp
SNPC.fp.spr
SNPC.post_cts.enc
SNPC.post_init.enc
SNPC.post_mplace.enc
SNPC.post_place.enc
SNPC.post_route.enc
SNPC.post_sroute.enc

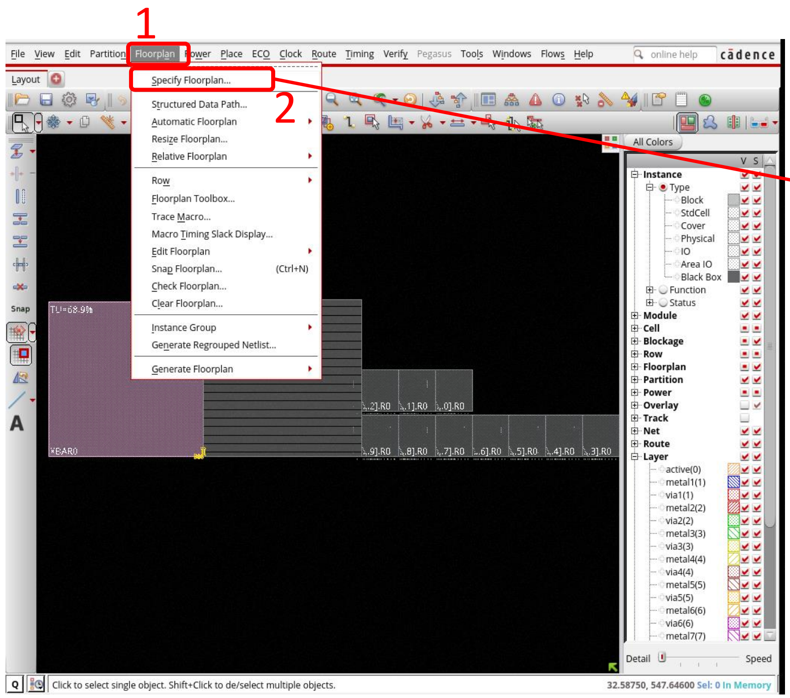
File name: import.enc

Files of type: All Files (*)

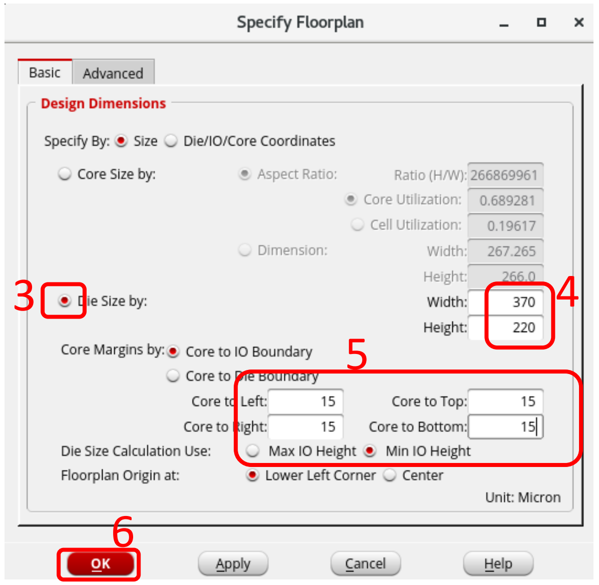
Click to select single object. Shift+Click to de/select multiple objects. -77.19550, 548.92250 Sel: 0 In Memory

In the welcome screen, we can see the modules of SNPC before placement. We should save the progress at each step. (1) Click **File**-> (2) **Save Design (**3) Check **Innovus** (4) Click on the folder icon. (5) Select the **checkpoints** folder (6) Save file as **import.enc (**7) **Save** and click **OK**

Step 2: Specify Floorplan



Click to select single object. Shift+Click to de/select multiple objects. 32.58750, 547.64600 Sel: 0 In Memory



Specify Floorplan

Basic | Advanced

Design Dimensions

Specify By: [✓ Size] Die/IO/Core Coordinates

Core Size by: Aspect Ratio: Ratio (H/W): 266869961
Core Utilization: 0.689281
Cell Utilization: 0.19617
Dimension: Width: 267.265
Height: 266.0

[✓ Die Size by:] Width: 370
Height: 220

Core Margins by: [✓ Core to IO Boundary] Core to Die Boundary

Core to Left: 15 Core to Top: 15
Core to Right: 15 Core to Bottom: 15

Die Size Calculation Use: Max IO Height [✓ Min IO Height]

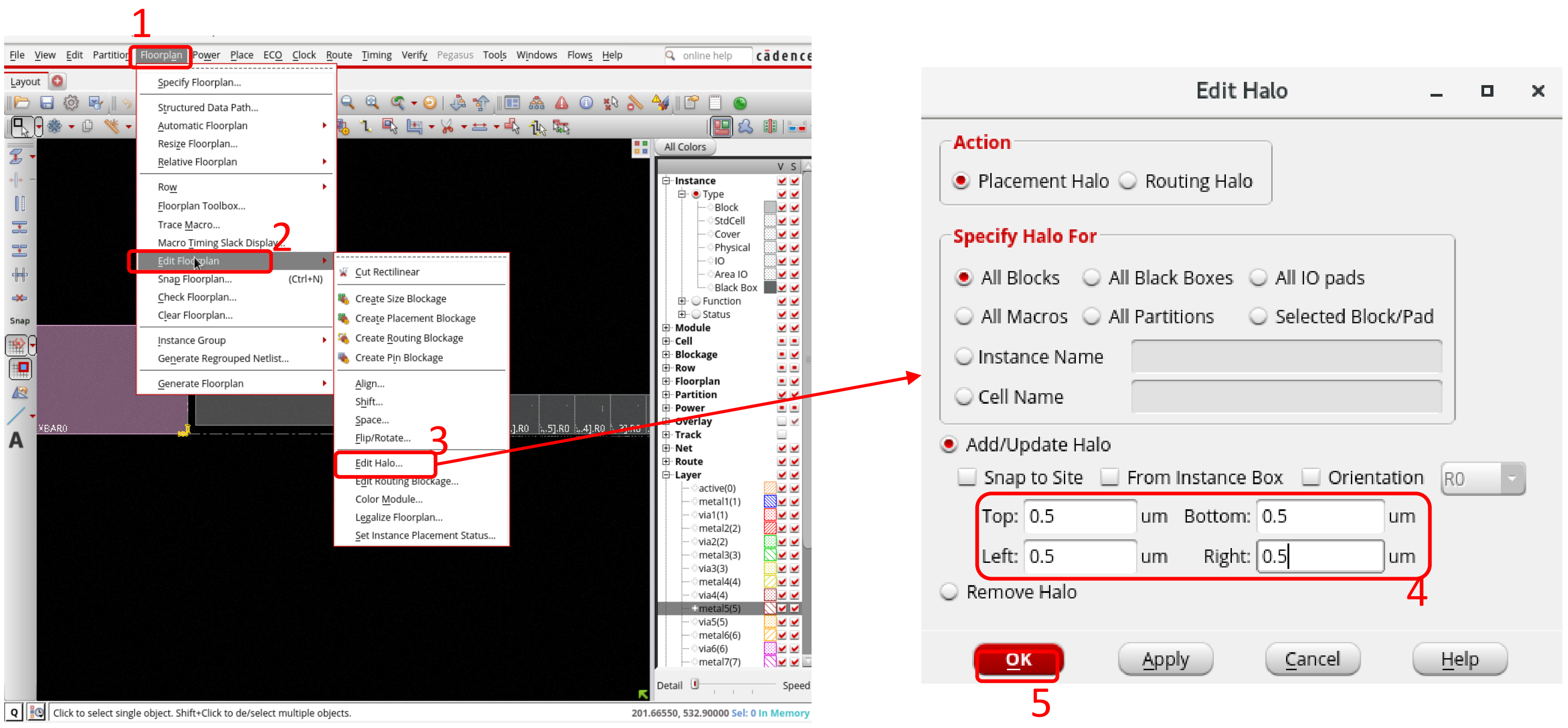
Floorplan Origin at: [✓ Lower Left Corner] Center

Unit: Micron

OK | Apply | Cancel | Help

(1) In this step we specify the floorplan Click **Floorplan**-> (2) **Specify Floorplan. (**3) Check **Die Size by.** (4) Enter 370 for Width and 220 Height. (5) Enter 15 for • **Core to Left** • **Core to Right** • **Core to Top** • **Core to Bottom. (**6) Click **OK**

Step 3: Add Halo to Block



Edit Halo

Action: [✓ Placement Halo] Routing Halo

Specify Halo For: [✓ All Blocks] All Black Boxes All IO pads All Macros All Partitions Selected Block/Pad Instance Name Cell Name

[✓ Add/Update Halo]

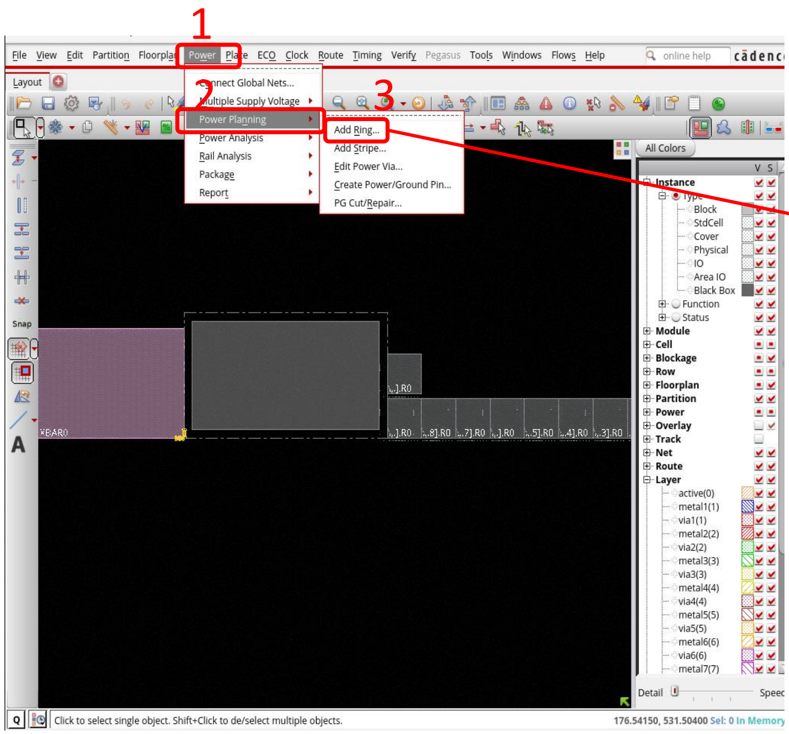
Snap to Site From Instance Box Orientation R0

Top:	0.5	um	Bottom:	0.5	um
Left:	0.5	um	Right:	0.5	um

Remove Halo

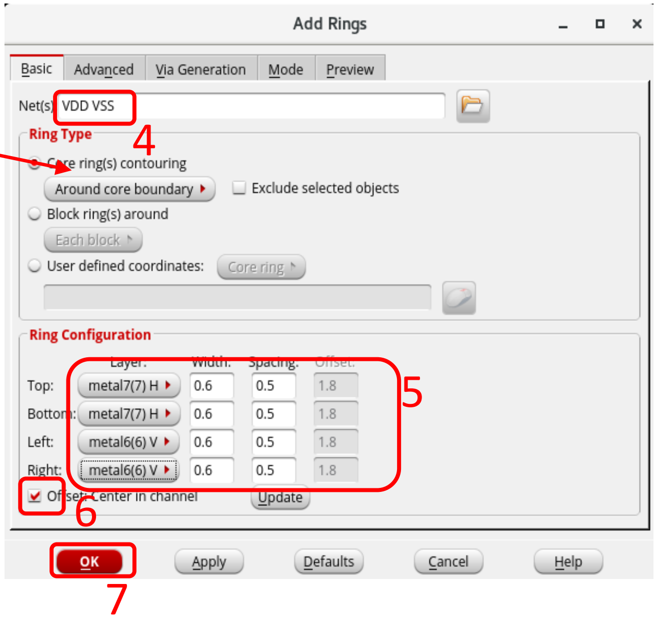
(1) In this step we specify the floorplan Click **Floorplan**->. (2) **Edit Floorplan. (**3) Click **Edit Halo** (4) Enter 0.5 for **•Top • Bottom •Left •Right. (**5) Click **OK**

Step 4: Add Core Rings



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

176.54150, 531.50400 Sel: 0 In Memory



Net(s): VDD VSS

Ring Type

- [✓ Core ring(s) contouring] Around core boundary
- Exclude selected objects
- Block ring(s) around: Each block
- User defined coordinates: Core ring

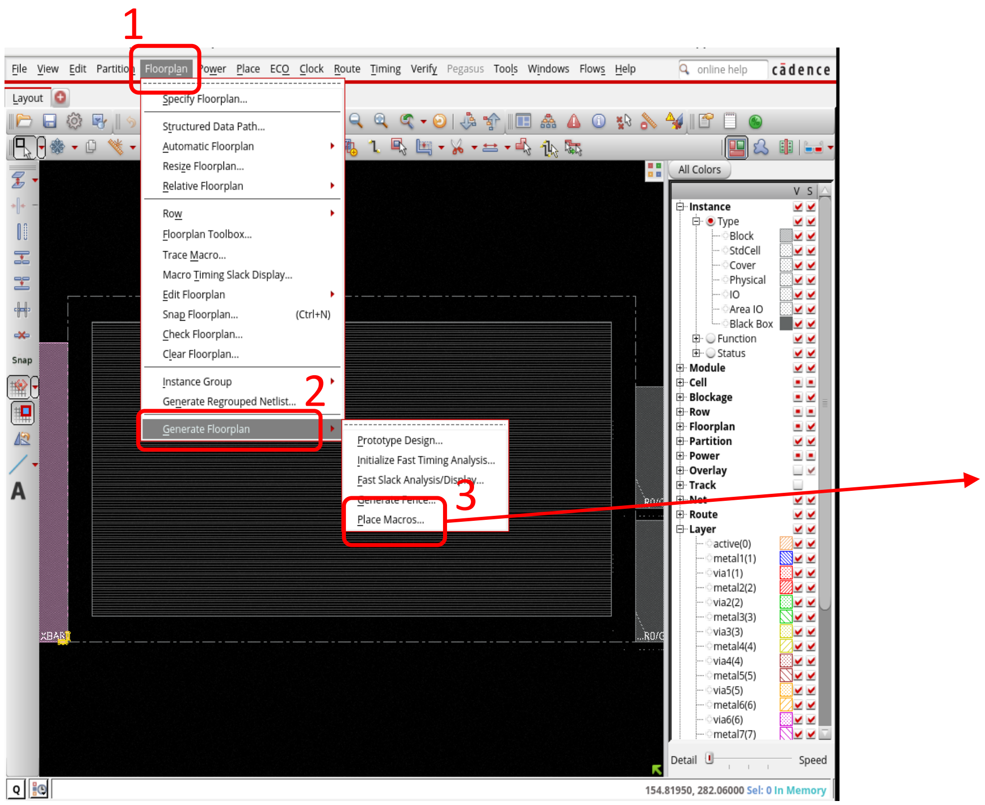
Ring Configuration

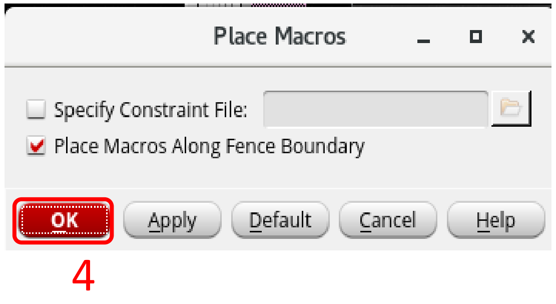
	Layer:	Width:	Spacing:	Offset:
Top:	metal7(7) H	0.6	0.5	1.8
Bottom:	metal7(7) H	0.6	0.5	1.8
Left:	metal6(6) V	0.6	0.5	1.8
Right:	metal6(6) V	0.6	0.5	1.8

[✓] Offset: Center in channel

(1) In this step we specify the floorplan Click **Power**-> (2)Select **Power Planning.** (3) Click **Add Rings** (4) Select **VDD VSS**. (5) Select **metal7H** for • **Top** and • **Bottom** (**Layer**), select **metal6V** for• **LEFT** and • **Right** (**Layer**). Enter **0.6** for **Width** and **0.5** for **Spacing.** (6) Check **Offset Center in Channel** (7) Click **OK**

Step 5: Place Macro (1/2)

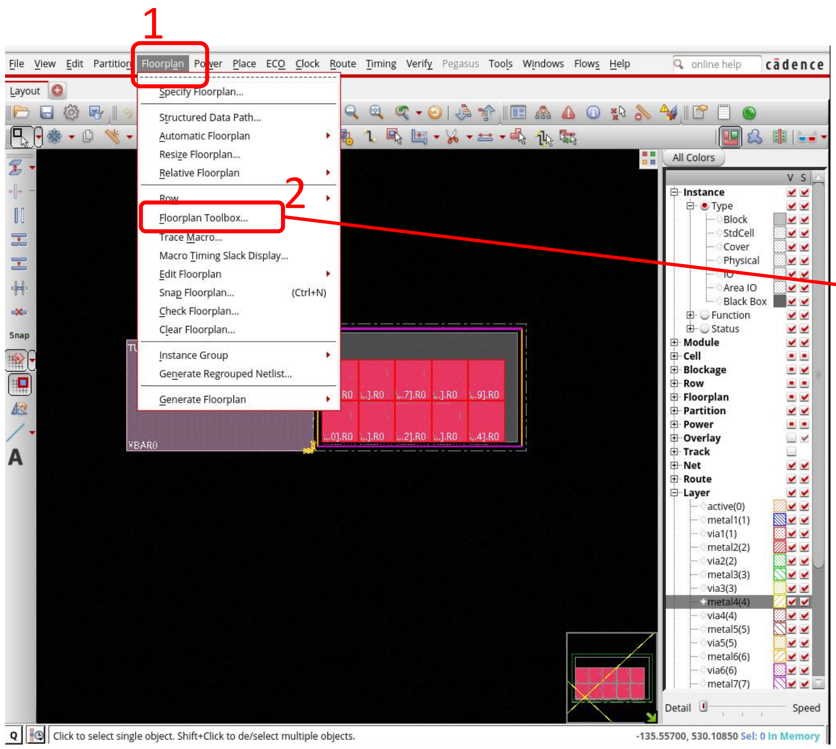




Specify Constraint File: [ ]
Place Macros Along Fence Boundary [✓]

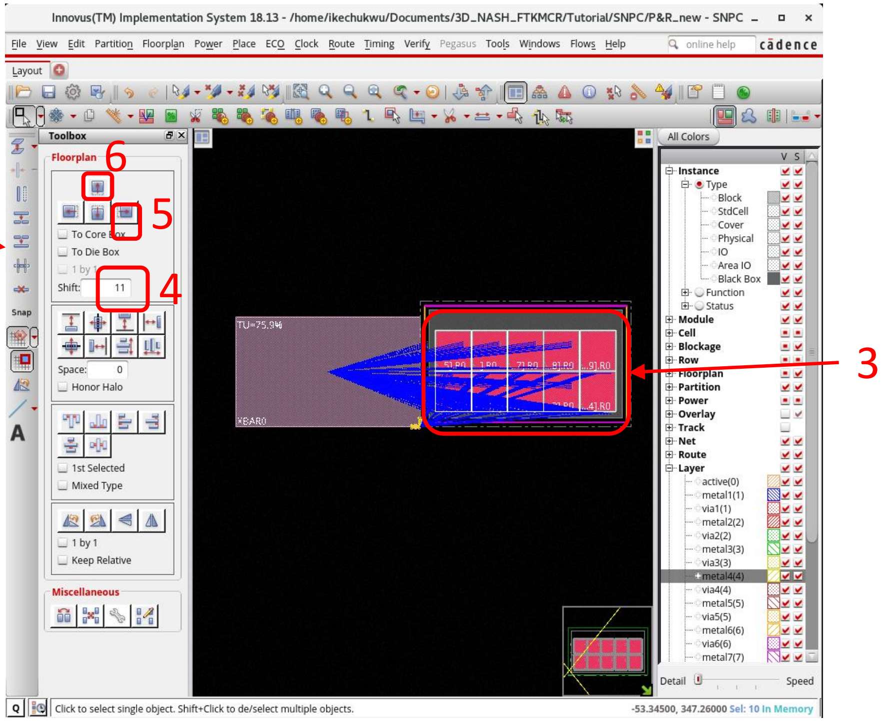
(1) Click on **Floorplan** (2) Select **Generate floorplan**. (3) Click on **Place Macro**.
(4) Click **OK**

Step 5: Place Macro (2/2)



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

-135.55700, 530.10850 Sel: 0 In Memory

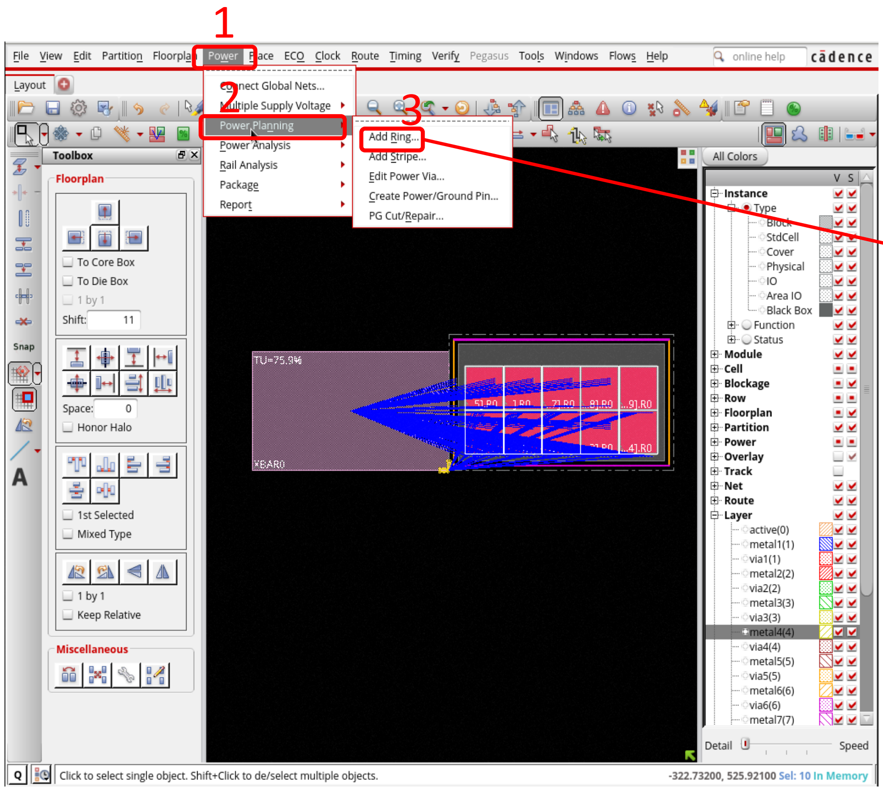


Click to select single object. Shift+Click to de/select multiple objects.

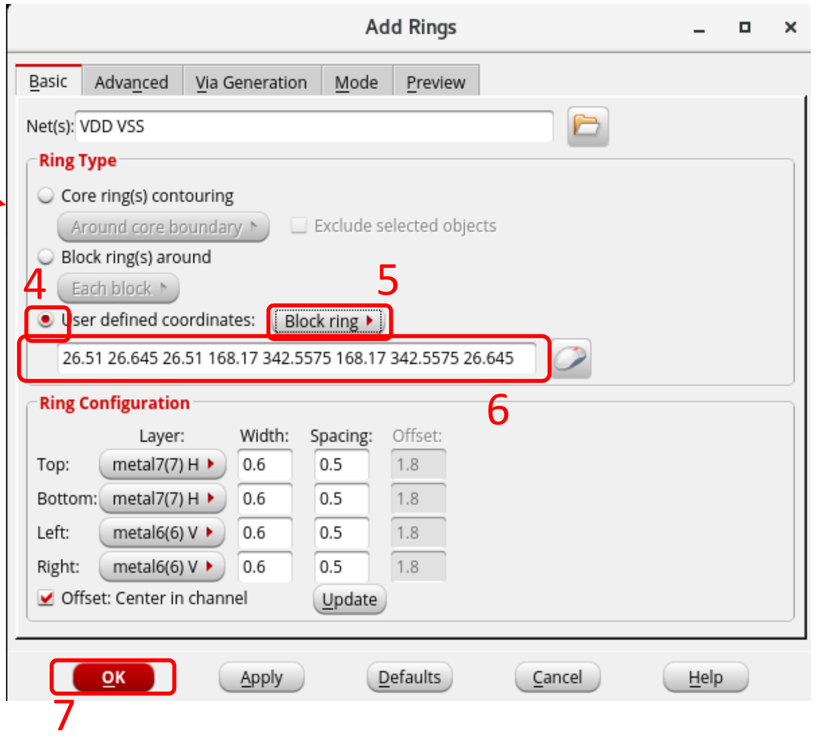
-53.34500, 347.26000 Sel: 10 In Memory

(1) Click on **Floorplan.** (2) Select **Floor Plan Toolbox**. (3) Select the SRAM macros by holding Ctrl and left clicking on them. (4) Set the **Shift** value to 11. (5) Click on the right arrow, and (6) the up arrow to shift once in each direction

Step 6: Add Block Rings



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



Add Rings

Basic | Advanced | Via Generation | Mode | Preview

Net(s): VDD VSS

Ring Type

- Core ring(s) contouring — Around core boundary; Exclude selected objects
- Block ring(s) around — Each block
- [✓ User defined coordinates:] Block ring

26.51 26.645 26.51 168.17 342.5575 168.17 342.5575 26.645

Ring Configuration

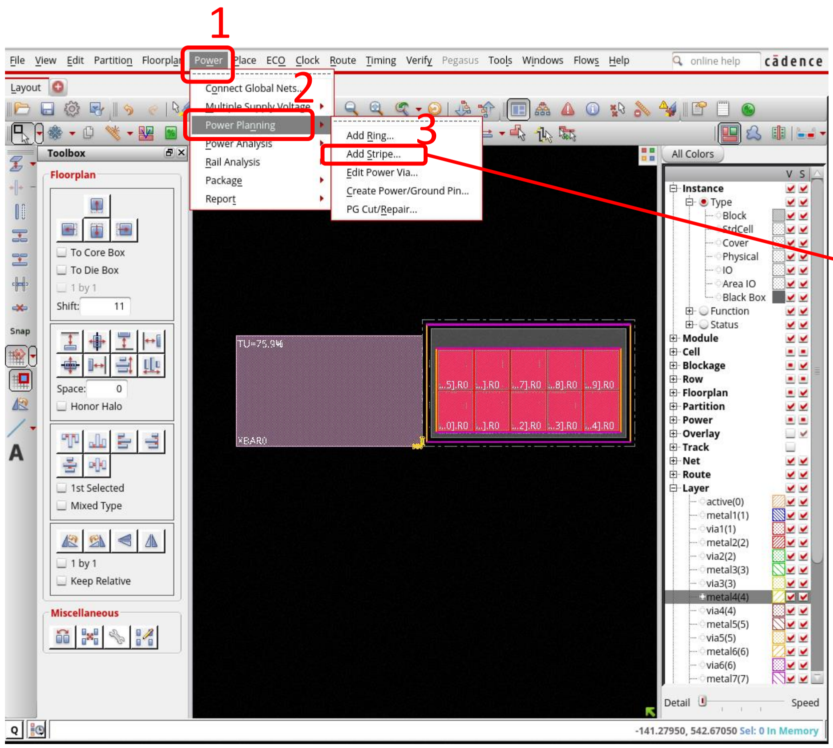
	Layer:	Width:	Spacing:	Offset:
Top:	metal7(7) H	0.6	0.5	1.8
Bottom:	metal7(7) H	0.6	0.5	1.8
Left:	metal6(6) V	0.6	0.5	1.8
Right:	metal6(6) V	0.6	0.5	1.8

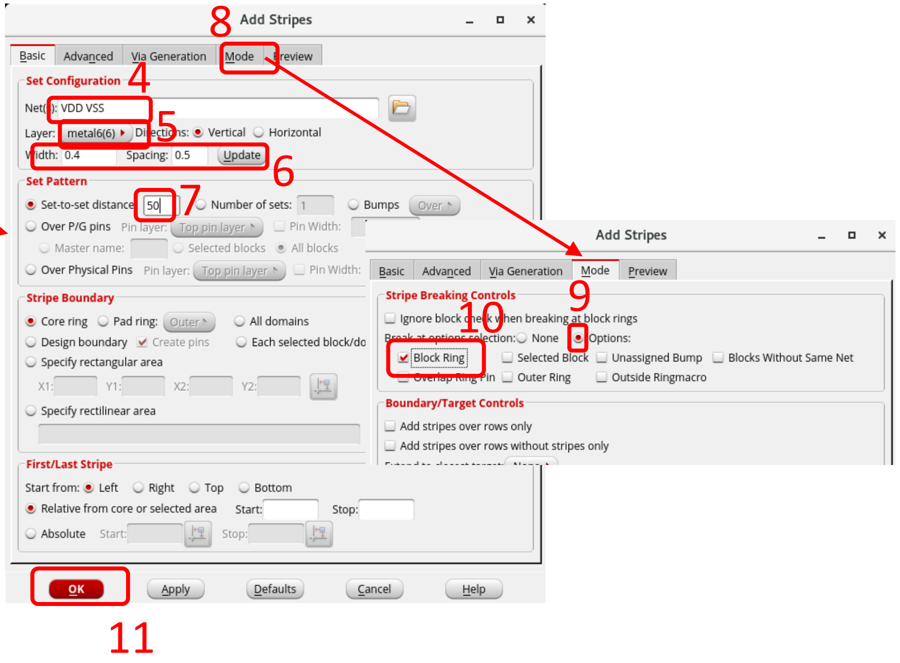
[✓] Offset: Center in channel   Update

OK   Apply   Defaults   Cancel   Help

(1) In this step we specify the floorplan Click **Power**-> (2)Select **Power Planning.** (3) Click **Add Rings** (4) Select **User defined coordinates**. (5) Select **Block ring.** (6) Enter the values **26.51 26.645 26.51 168.17 342.5575 168.17 342.5575 26.245** for the clock ring coordinates. (7) Click **OK**

Step 7: Add Power Stripe

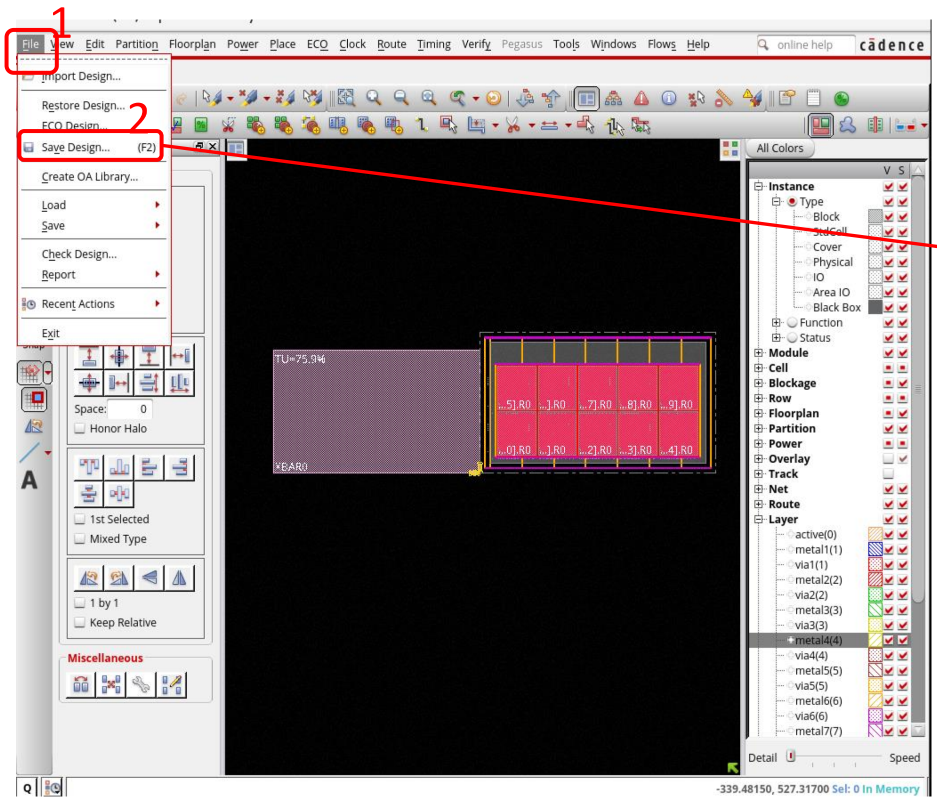


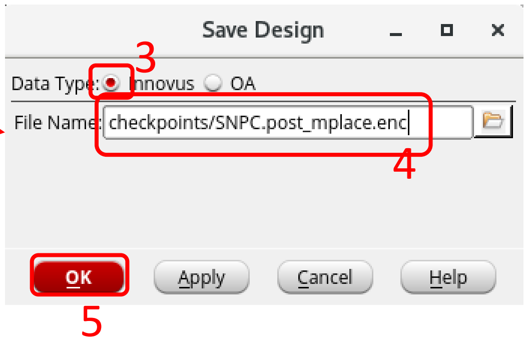


Net(s): VDD VSS
Layer: metal6(6)
Direction: [✓ Vertical] Horizontal
Width: 0.4
Spacing: 0.5
Set-to-set distance: 50
Break at options selection: None [✓ Options:]
[✓ Block Ring]

(1) Click on Power-> (2) Power Planning-> (3)Add Stripe. (4) Input VDD VSS in Nets(s): (5) Change Layer to metal **6**, (6) Set: • Width to **0.4** • Spacing to **0.5**, (7) Change Set-to-set-distance to **50.** (8) Click on **Mode.** (9) Check **Options**. (10) Check **Block Rings.** (11) Click **OK**

Save Design Progress- Checkpoint



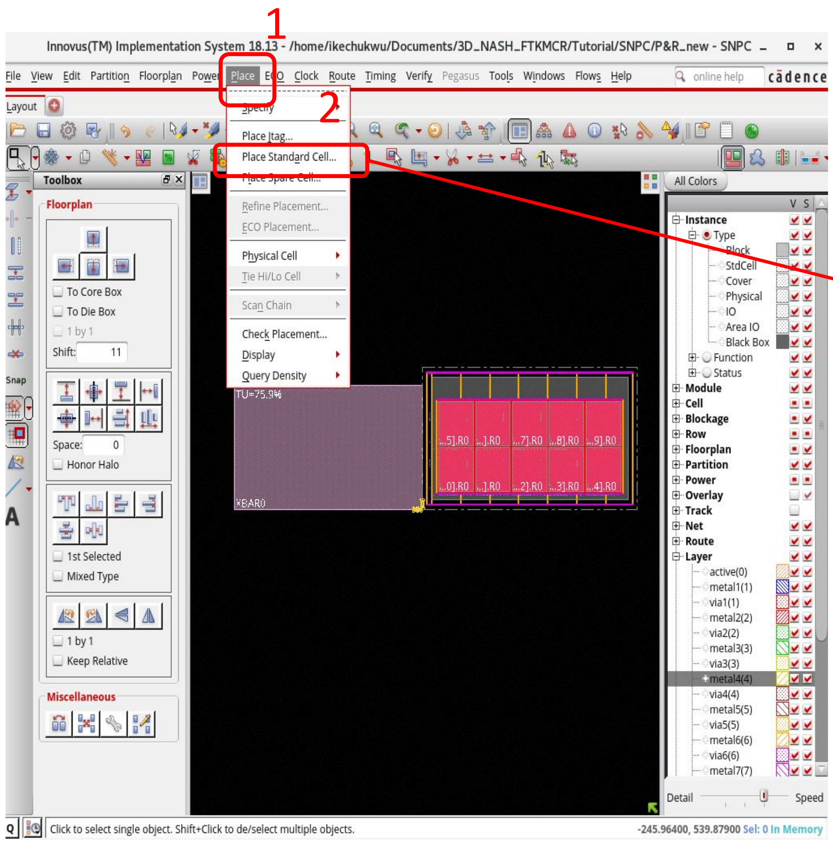

Save Design

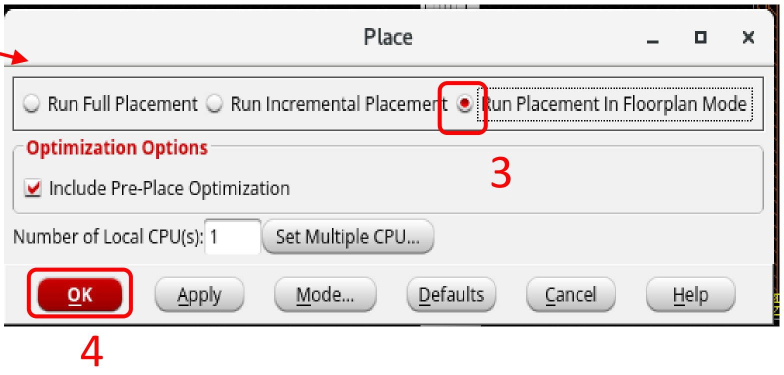
Data Type: [✓ Innovus] OA

File Name: checkpoints/SNPC.post_mplace.enc

In the welcome screen, we can see the floorplan and the placed macros. Save the progress. (1) Click **File**-> (2) Click on **Save Design** (3) select **Innovus.** (4) Enter the **checkpoints** directory, and save the design progress as **SNPC.post_mplace.enc**. (5) Click on **OK**

Step 8: Place Standard Cell


1
2
Innovus(TM) Implementation System 18.13 - /home/ikechukwu/Documents/3D_NASH_FTKMCR/Tutorial/SNPC/P&R_new - SNPC
File View Edit Partition Floorplan Power Place ECO Clock Route Timing Verify Pegasus Tools Windows Flows Help
Specify
Place Jtag...
Place Standard Cell...
Place Spare Cell...
Refine Placement...
ECO Placement...
Physical Cell
Tie Hi/Lo Cell
Scan Chain
Check Placement...
Display
Query Density
Toolbox
Floorplan
Miscellaneous
All Colors
Instance
Module
Cell
Blockage
Row
Floorplan
Partition
Power
Overlay
Track
Net
Route
Layer
Click to select single object. Shift+Click to de/select multiple objects.



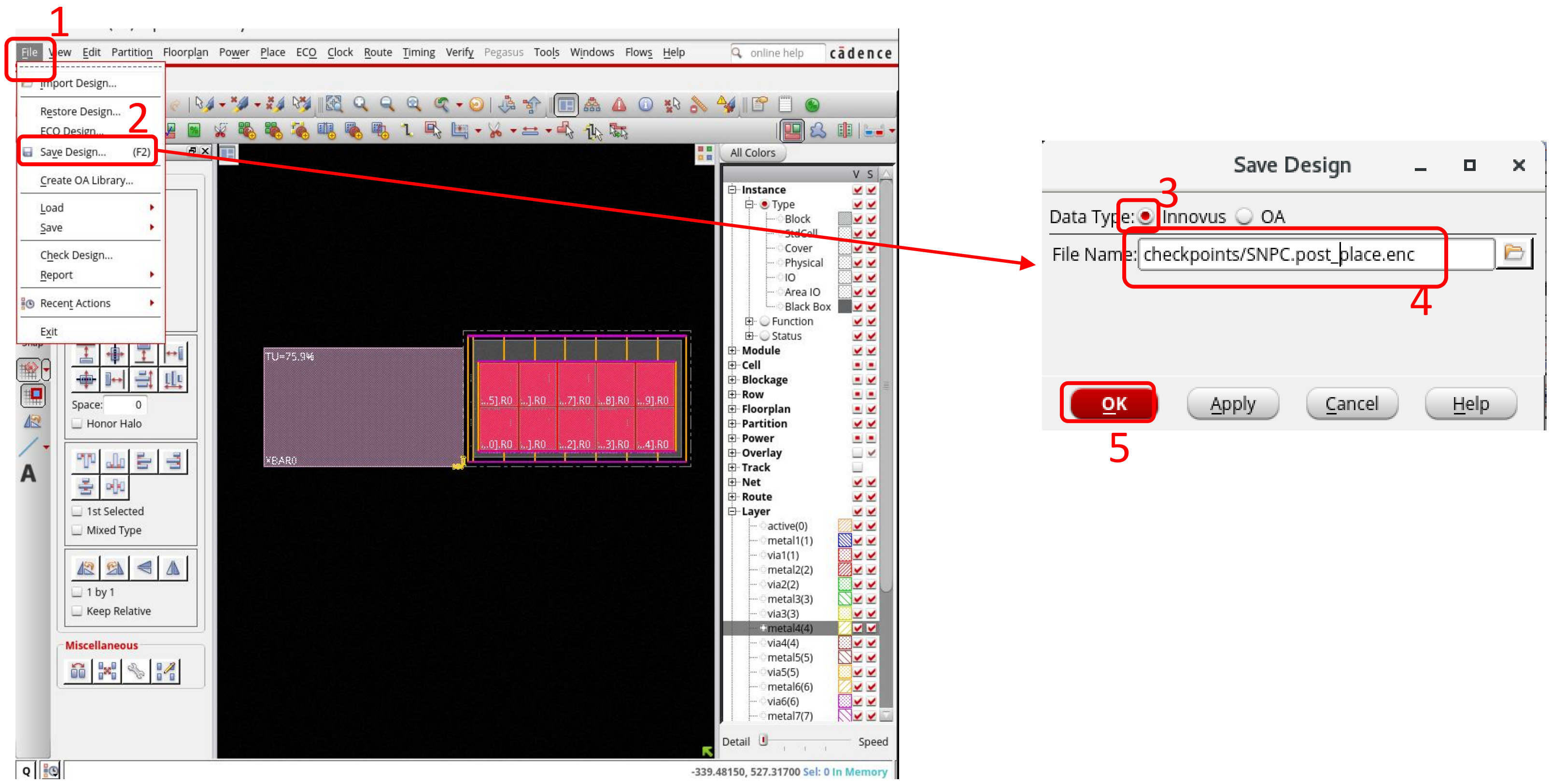
Run Full Placement Run Incremental Placement [✓ Run Placement In Floorplan Mode]

[✓] Include Pre-Place Optimization

Number of Local CPU(s): 1

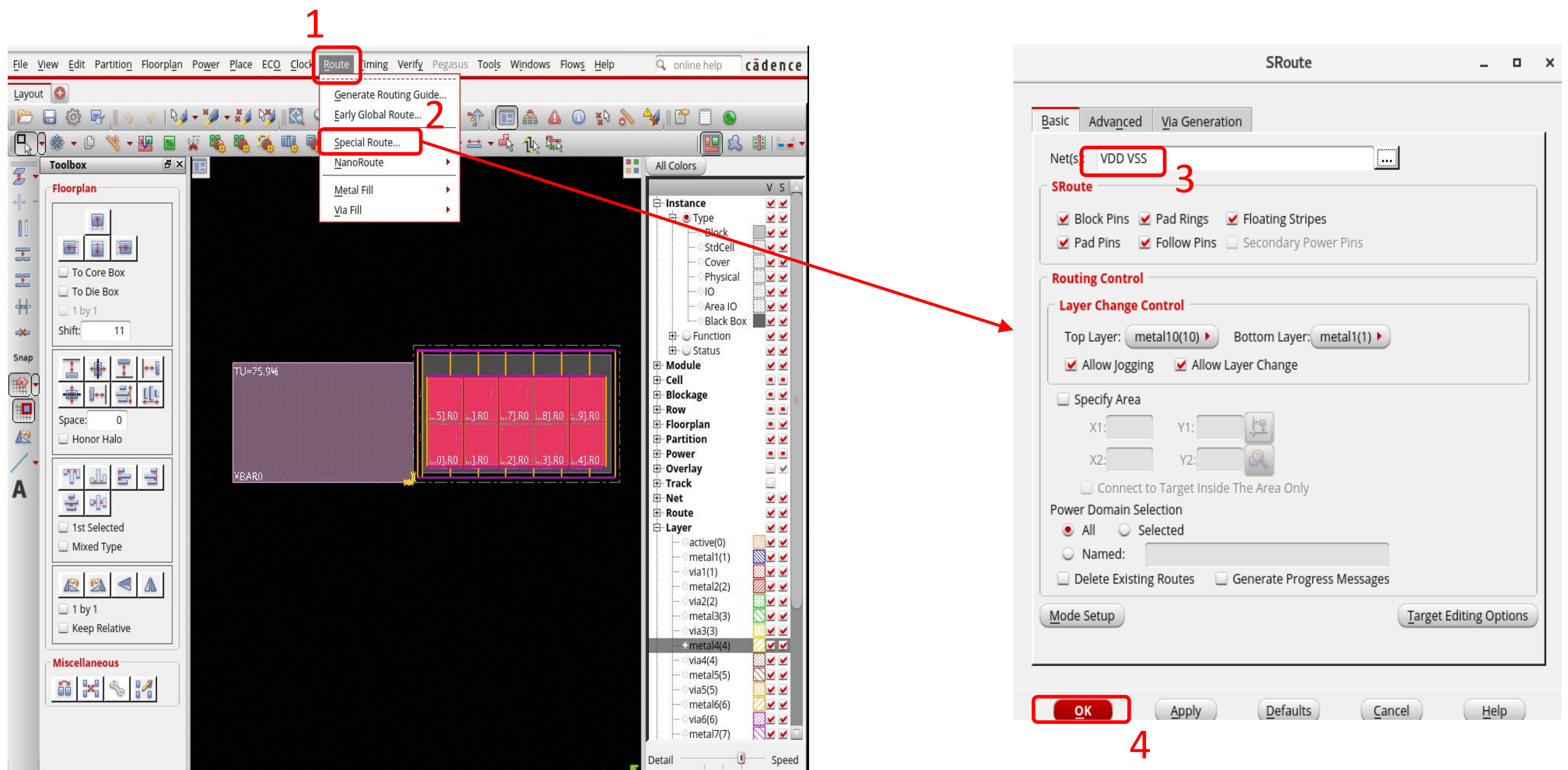
Now we place the SNPC modules on the die: (1) Click **Place**-> (2) **Place Standard Cell.** (3) select **Run Placement in Floorplan Mode**. (4) Click **OK**

Save Design Progress- Checkpoint



In the welcome screen, we can see the floorplan and the placed macros. Save the progress. (1) Click **File**-> (2) Click on **Save Design** (3) select **Innovus.** (4) Enter the **checkpoints** directory, and save the design progress as **SNPC.post_place.enc**. (5) Click on **OK**

Step 9: Power Routing



SRoute

Basic | Advanced | Via Generation

Net(s): VDD VSS

SRoute

- [✓] Block Pins
- [✓] Pad Rings
- [✓] Floating Stripes
- [✓] Pad Pins
- [✓] Follow Pins
- [ ] Secondary Power Pins

Routing Control

Layer Change Control

Top Layer: metal10(10) Bottom Layer: metal1(1)

- [✓] Allow Jogging
- [✓] Allow Layer Change
- [ ] Specify Area

X1: Y1: X2: Y2:

- [ ] Connect to Target Inside The Area Only

Power Domain Selection

[● All] ○ Selected ○ Named:

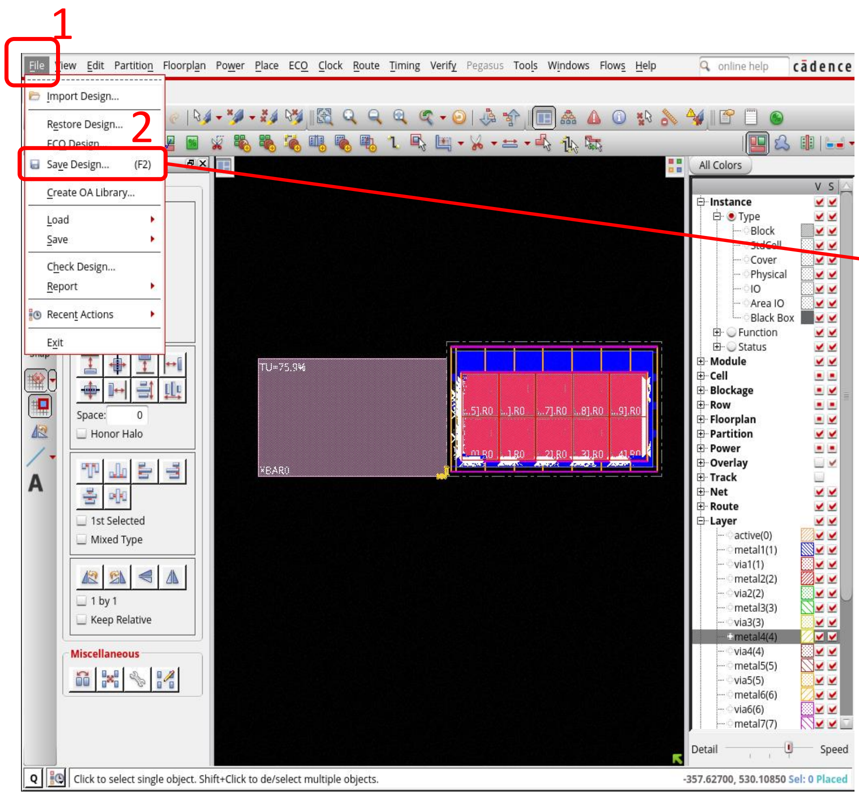
- [ ] Delete Existing Routes
- [ ] Generate Progress Messages

Mode Setup | Target Editing Options

OK | Apply | Defaults | Cancel | Help

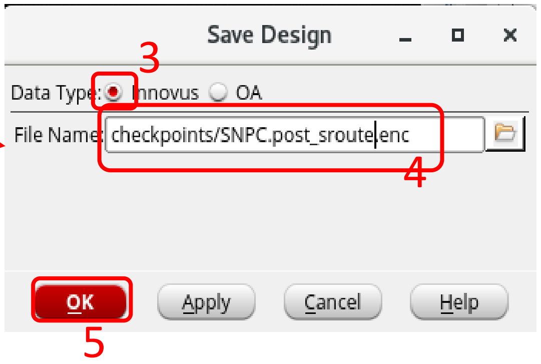
(1) Click on **Route->** (2) **Special Route.** (3) Input **VDD VSS** in Nets(s): (4) Click **OK**

Save Design Progress- Checkpoint



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

-357.62700, 530.10850 Sel: 0 Placed

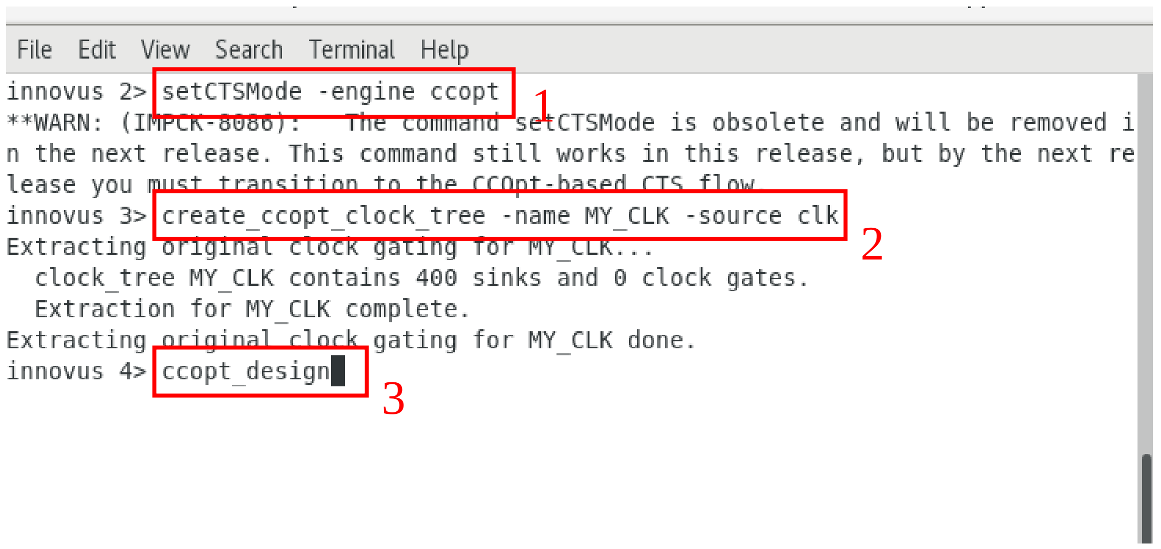


Data Type: [✓ Innovus] OA

File Name: checkpoints/SNPC.post_sroute.enc

In the welcome screen, we can see the floorplan and the placed macros. Save the progress. (1) Click **File**-> (2) Click on **Save Design** (3) select **Innovus.** (4) Enter the **checkpoints** directory, and save the design progress as **SNPC.post_sroute.enc**. (5) Click on **OK**

Step 10: Clock Tree Synthesis



```
innovus 2> setCTSMode -engine ccopt
**WARN: (IMPCK-8086):   The command setCTSMode is obsolete and will be removed i
n the next release. This command still works in this release, but by the next re
lease you must transition to the CCOpt-based CTS flow.
innovus 3> create_ccopt_clock_tree -name MY_CLK -source clk
Extracting original clock gating for MY_CLK...
  clock_tree MY_CLK contains 400 sinks and 0 clock gates.
  Extraction for MY_CLK complete.
Extracting original clock gating for MY_CLK done.
innovus 4> ccopt_design
```

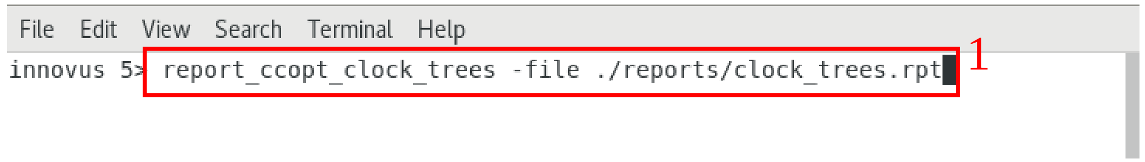
To perform the clock tree synthesis, we have to use the following commands on the terminal from which we launched innovus.

(1) *setCTSMode –engine ccopt*

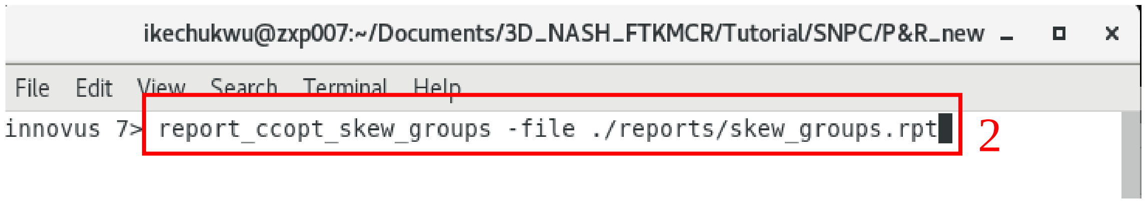
(2) *create_ccopt_clock_tree –name MY_CLK –source clk*

(3) *ccopt_design*

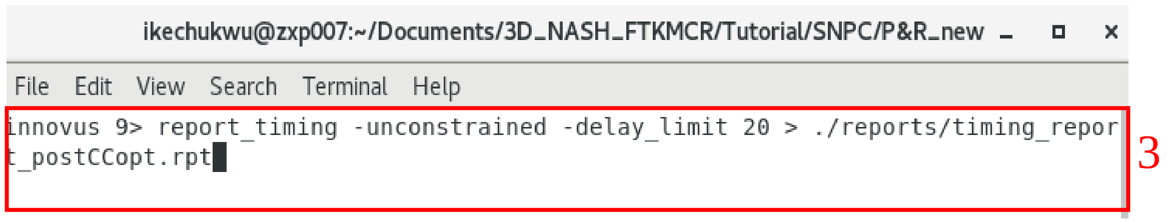
Step 10: Clock Tree Synthesis Report



```
innovus 5> report_ccopt_clock_trees -file ./reports/clock_trees.rpt
```


ikechukwu@zxp007:~/Documents/3D_NASH_FTKMCR/Tutorial/SNPC/P&R_new
File Edit View Search Terminal Help
2

```
innovus 7> report_ccopt_skew_groups -file ./reports/skew_groups.rpt
```


ikechukwu@zxp007:~/Documents/3D_NASH_FTKMCR/Tutorial/SNPC/P&R_new
File Edit View Search Terminal Help
3

```
innovus 9> report_timing -unconstrained -delay_limit 20 > ./reports/timing_repor
t_postCCopt.rpt
```

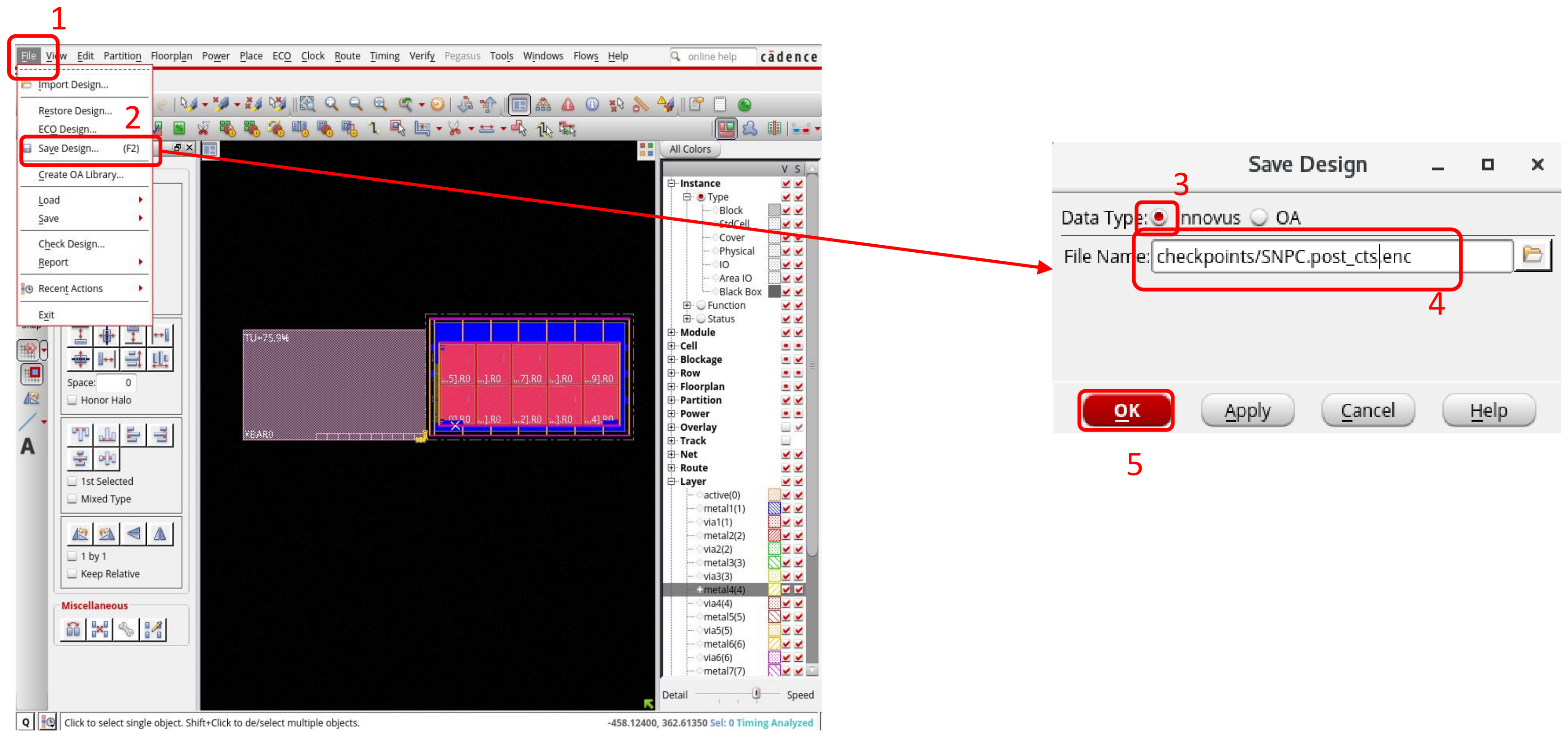
To generate clock tree synthesis report, we have to use the following commands when the clock tree synthesis is complete.

(1) *report_ccopt_clock_trees -file ./reports/clock_trees.rpt*

(2) *report_ccopt_skew_groups -file ./reports/skew_groups.rpt*

(3) *report_timing -unconstrained -delay_limit 20 > ./reports/timing_report_postCCopt.rpt*

Save Design Progress- Checkpoint

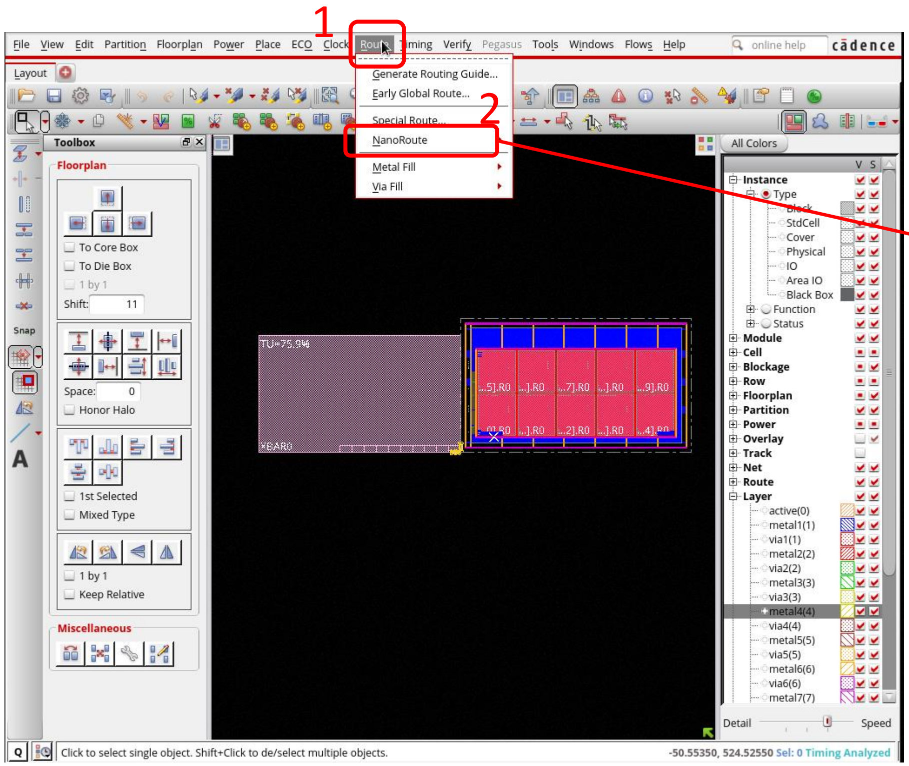


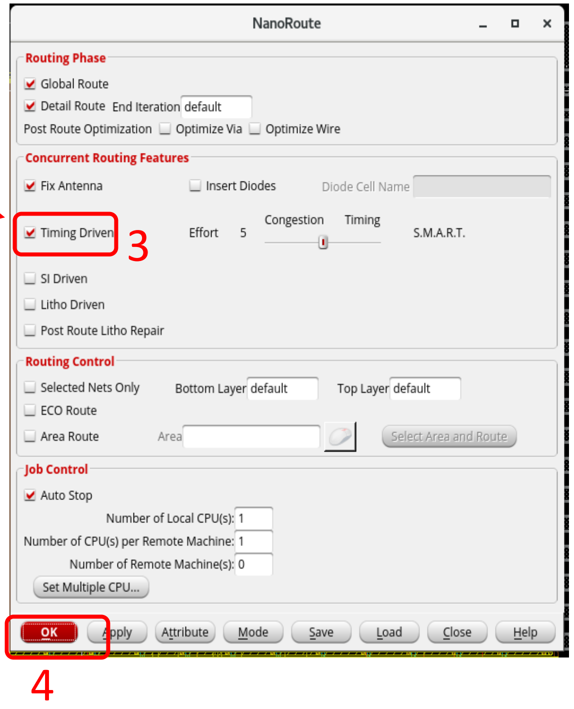
Data Type: [✓ Innovus] OA

File Name: checkpoints/SNPC.post_cts.enc

Save the progress. (1) Click **File**-> (2) Click on **Save Design** (3) select **Innovus.** (4) Enter the **checkpoints** directory, and save the design progress as **SNPC.post_cts.enc**. (5) Click on **OK**

Step 11: Global Route


1
File View Edit Partition Floorplan Power Place ECO Clock Route Timing Verify Pegasus Tools Windows Flows Help
Generate Routing Guide...
Early Global Route...
Special Route...
2
NanoRoute
Metal Fill
Via Fill



Routing Phase
- [✓] Global Route
- [✓] Detail Route   End Iteration: default
- Post Route Optimization [ ] Optimize Via [ ] Optimize Wire

Concurrent Routing Features
- [✓] Fix Antenna   [ ] Insert Diodes   Diode Cell Name:
- [✓] Timing Driven   Effort 5   Congestion / Timing   S.M.A.R.T.
- [ ] SI Driven
- [ ] Litho Driven
- [ ] Post Route Litho Repair

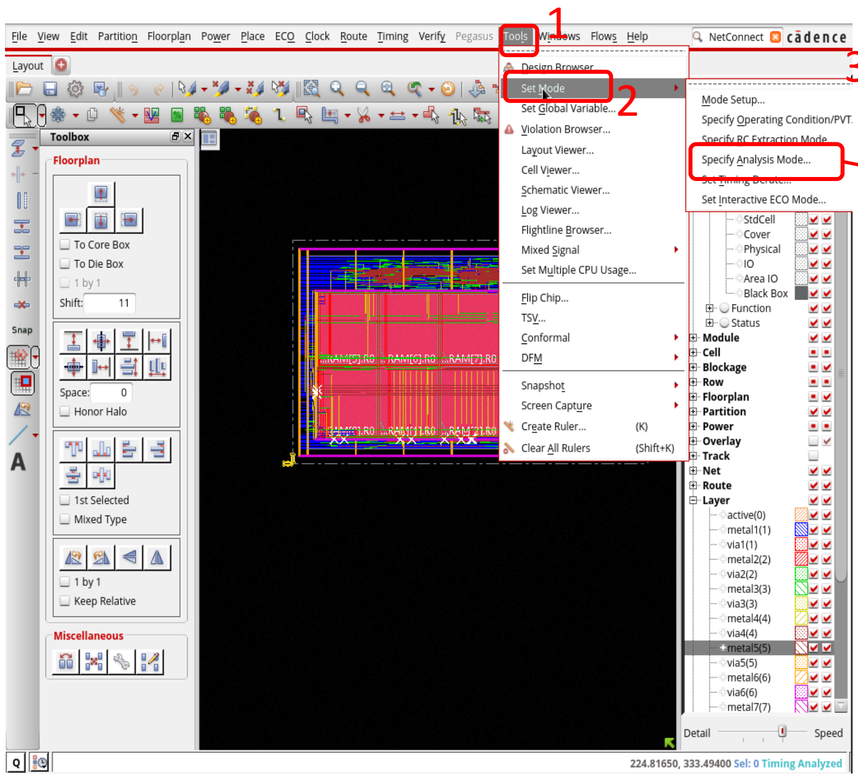
Routing Control
- [ ] Selected Nets Only   Bottom Layer: default   Top Layer: default
- [ ] ECO Route
- [ ] Area Route   Area: Select Area and Route

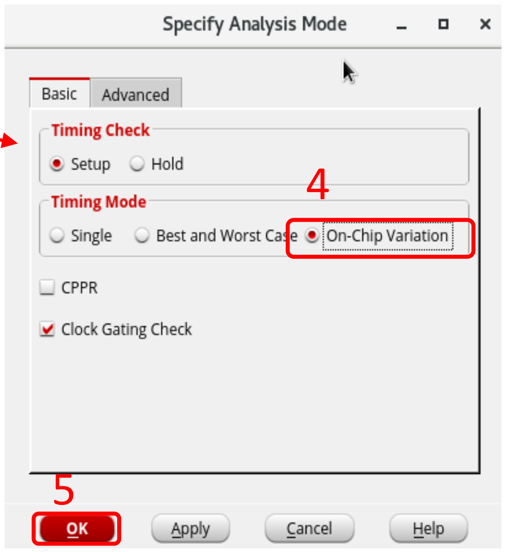
Job Control
- [✓] Auto Stop
- Number of Local CPU(s): 1
- Number of CPU(s) per Remote Machine: 1
- Number of Remote Machine(s): 0
- Set Multiple CPU...

OK   Apply   Attribute   Mode   Save   Load   Close   Help

Perform early global route. (1) Click **Route**-> (2) Click on **Nanoroute** -> **Route….** (3) Check **Timing Driven**. (4)Click on **OK**

Step 12: Optimization- Setting





Timing Check: [✓ Setup] Hold

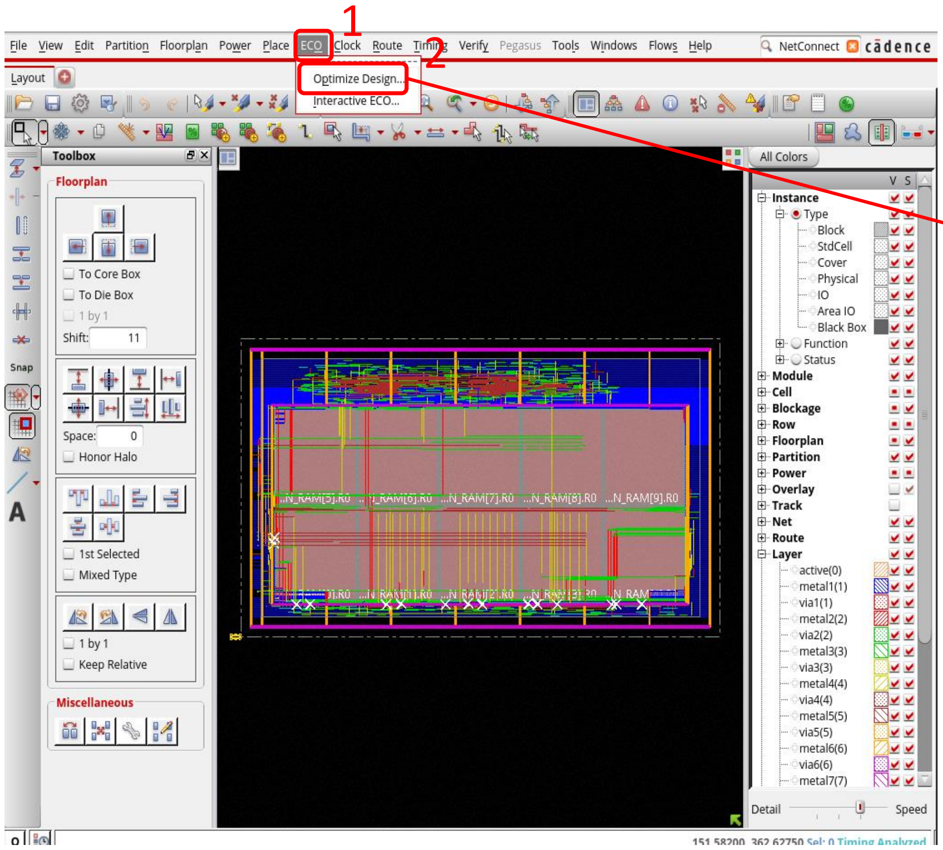
Timing Mode: Single Best and Worst Case [✓ On-Chip Variation]

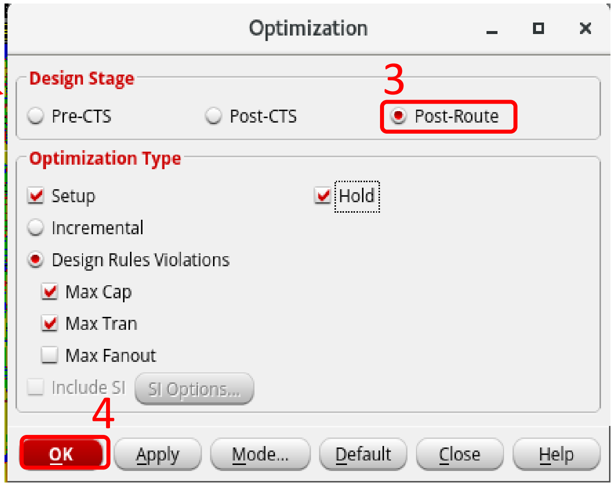
CPPR

[✓] Clock Gating Check

(1) Click on **Tools**-> (2) **Set Mode**-> (3) **Specify Analysis Mode.** (4)Check **On-Chip Variation** and then (5) Click **OK**

Step 12: Optimization


1
2
File View Edit Partition Floorplan Power Place ECO Clock Route Timing Verify Pegasus Tools Windows Flows Help
Optimize Design...
Interactive ECO...
Toolbox
Floorplan
Miscellaneous
All Colors



Optimization

Design Stage

- Pre-CTS
- Post-CTS
- [✓ Post-Route]

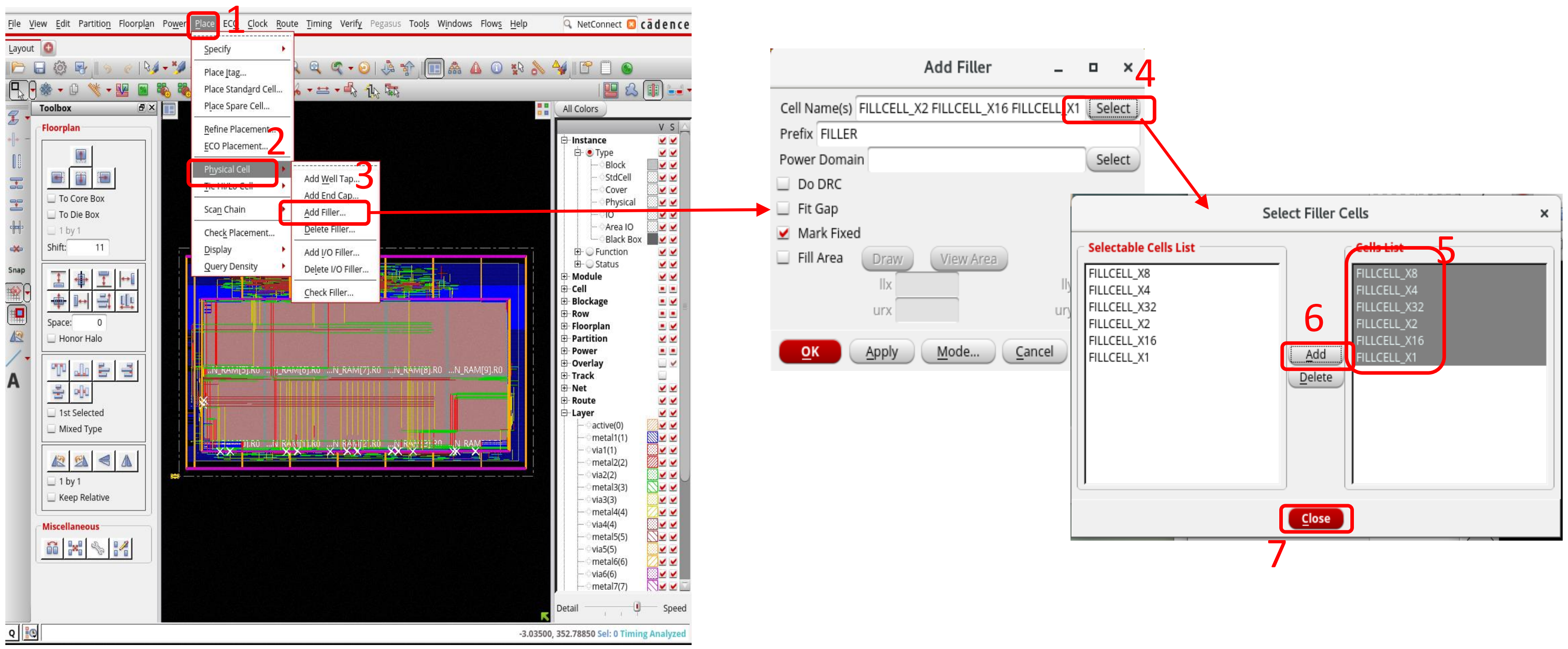
Optimization Type

- [✓] Setup
- [✓] Hold
- Incremental
- [✓ Design Rules Violations]
  - [✓] Max Cap
  - [✓] Max Tran
  - [ ] Max Fanout
- [ ] Include SI

SI Options... | OK | Apply | Mode... | Default | Close | Help

(1) Click on **ECO**-> **Optimize Design.** Check **Post Route** and then click **OK**

Step 13: Add Fillers



Add Filler

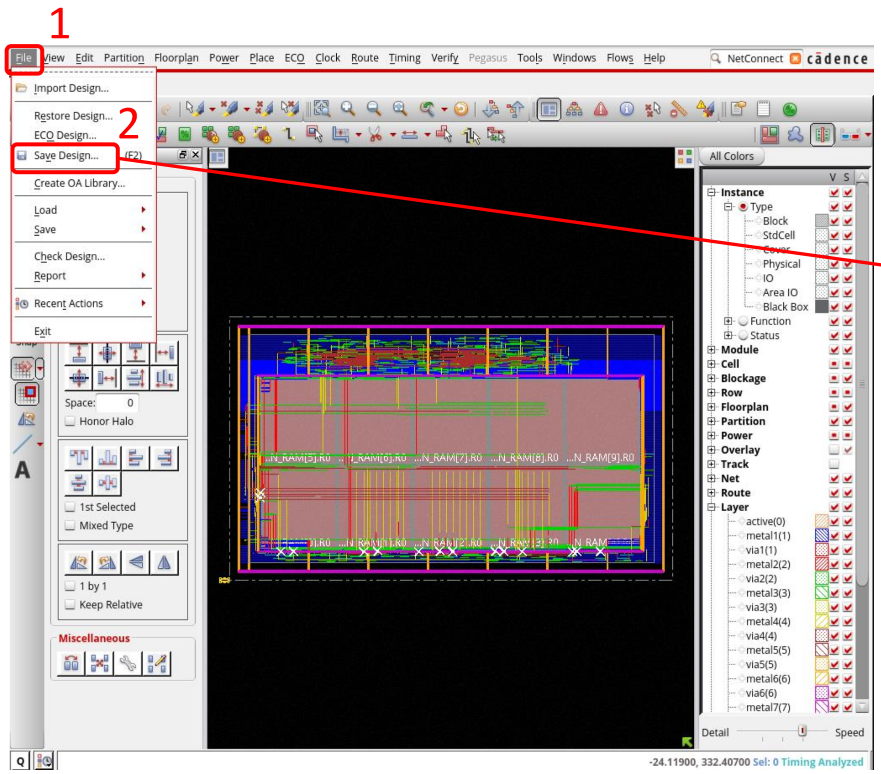
Cell Name(s)	FILLCELL_X2 FILLCELL_X16 FILLCELL_X1
Prefix	FILLER
Power Domain	
Do DRC	☐
Fit Gap	☐
Mark Fixed	✓
Fill Area	☐

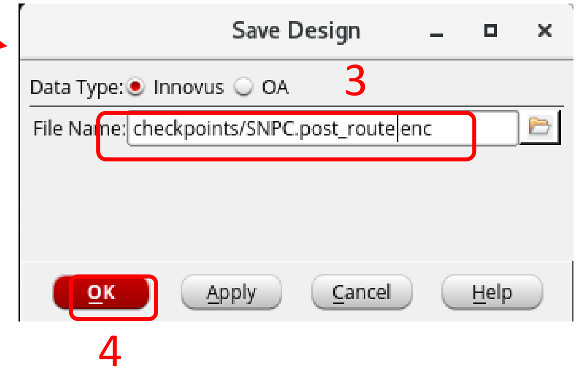
Select Filler Cells

Selectable Cells List	Cells List
FILLCELL_X8	FILLCELL_X8
FILLCELL_X4	FILLCELL_X4
FILLCELL_X32	FILLCELL_X32
FILLCELL_X2	FILLCELL_X2
FILLCELL_X16	FILLCELL_X16
FILLCELL_X1	FILLCELL_X1

(1) Click on **Place** -> (2) **Physical Cells->** (3) **Add Filler.** (4) Click **Select**. (5) Select **FILLCELL_X1,2,4,8,16,32**. (6) Click **ADD**. (7) Click **Close**. -> Click **OK**

Save Design Progress- Checkpoint



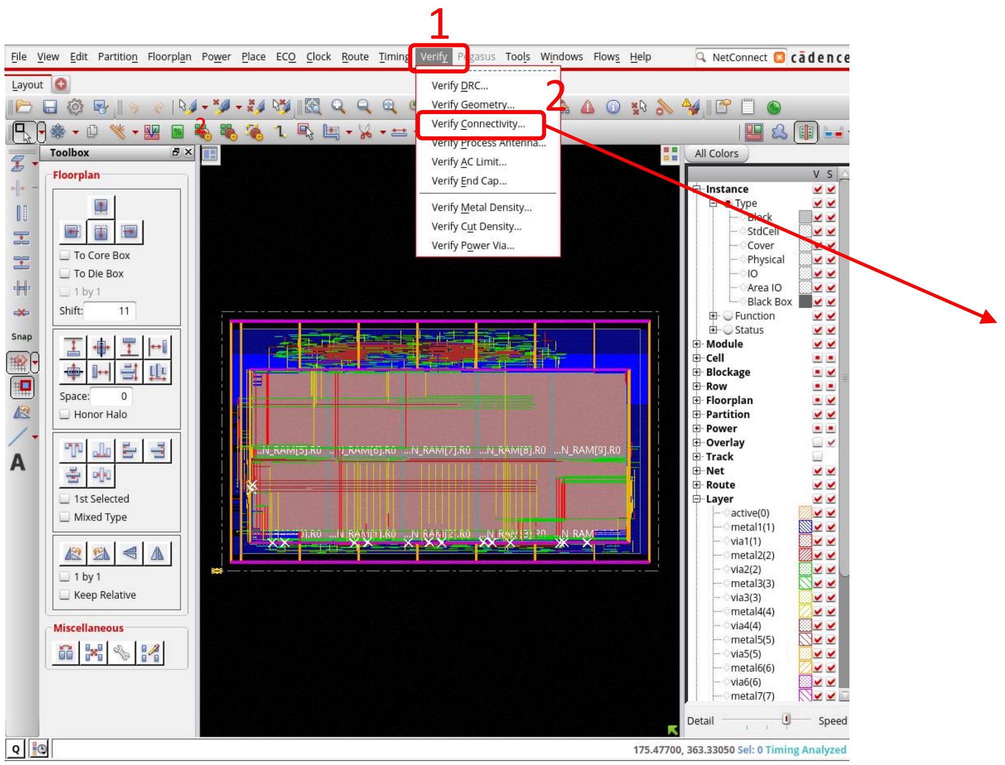


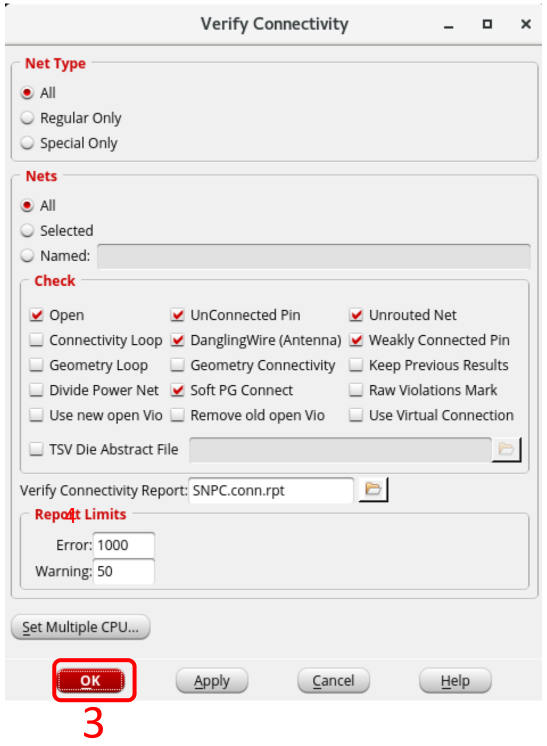
Data Type: [✓ Innovus] OA

File Name: checkpoints/SNPC.post_route.enc

Save the progress. (1) Click **File**-> (2) Click on **Save Design** (3) Enter the **checkpoints** directory, and save the design progress as **SNPC.post_route.enc**. (4) Click on **OK**

Step 14: Verification- Connectivity





Verify Connectivity

Net Type
- [✓ All]
- Regular Only
- Special Only

Nets
- [✓ All]
- Selected
- Named:

Check

✓ Open	✓ UnConnected Pin	✓ Unrouted Net
☐ Connectivity Loop	✓ DanglingWire (Antenna)	✓ Weakly Connected Pin
☐ Geometry Loop	☐ Geometry Connectivity	☐ Keep Previous Results
☐ Divide Power Net	✓ Soft PG Connect	☐ Raw Violations Mark
☐ Use new open Vio	☐ Remove old open Vio	☐ Use Virtual Connection

☐ TSV Die Abstract File

Verify Connectivity Report: SNPC.conn.rpt

Report Limits

Error: 1000

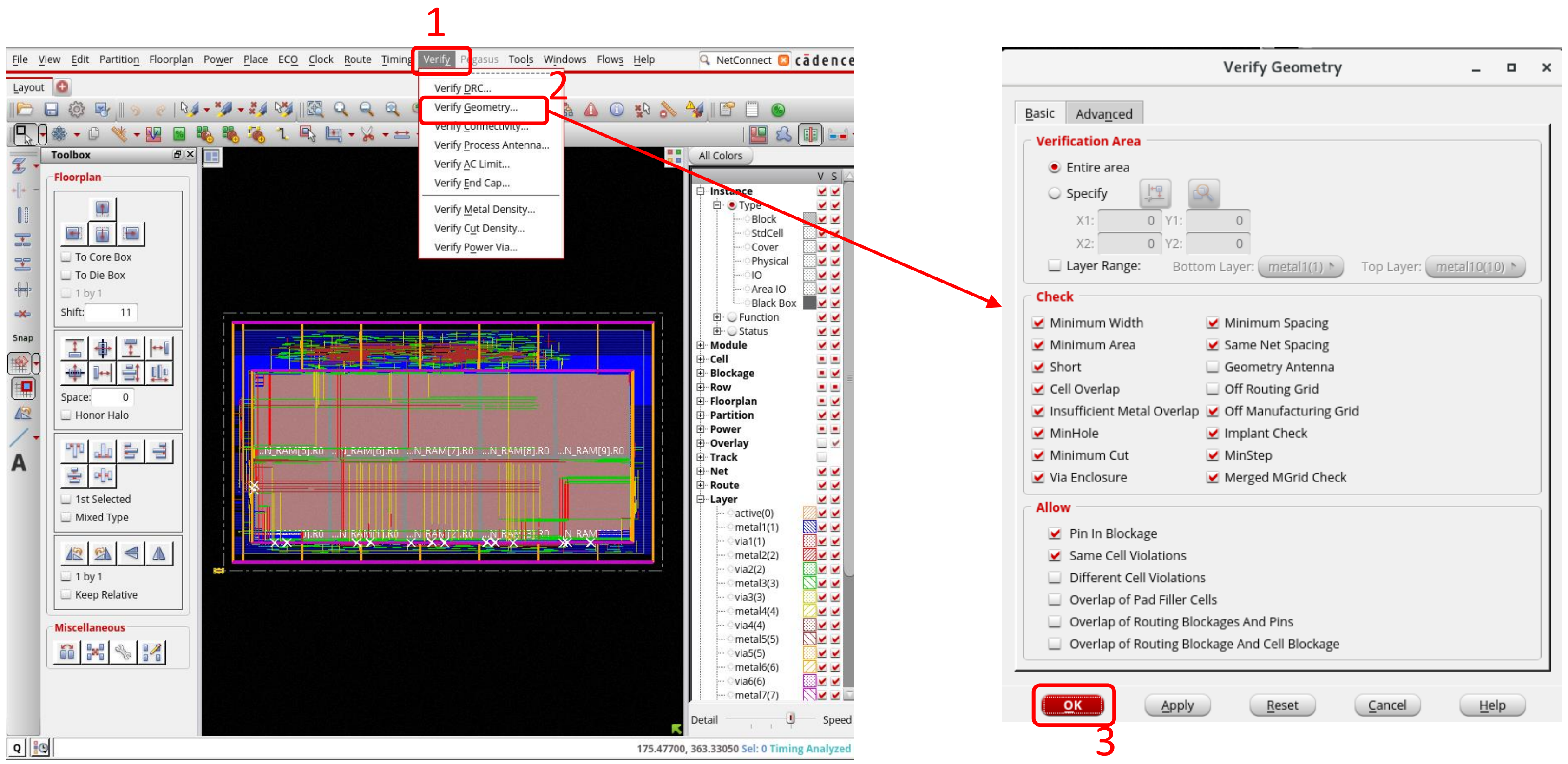
Warning: 50

Set Multiple CPU...

OK Apply Cancel Help

Verify Connectivity. (1) Click **Verify**-> (2) Click on **Verify Connectivity.** (3) Click on **OK**

Step 14:Verification- Geometry



Verify Geometry

Basic | Advanced

Verification Area

- [✓ Entire area] Specify
- X1: 0 Y1: 0
- X2: 0 Y2: 0
- ☐ Layer Range: Bottom Layer: metal1(1) Top Layer: metal10(10)

Check

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

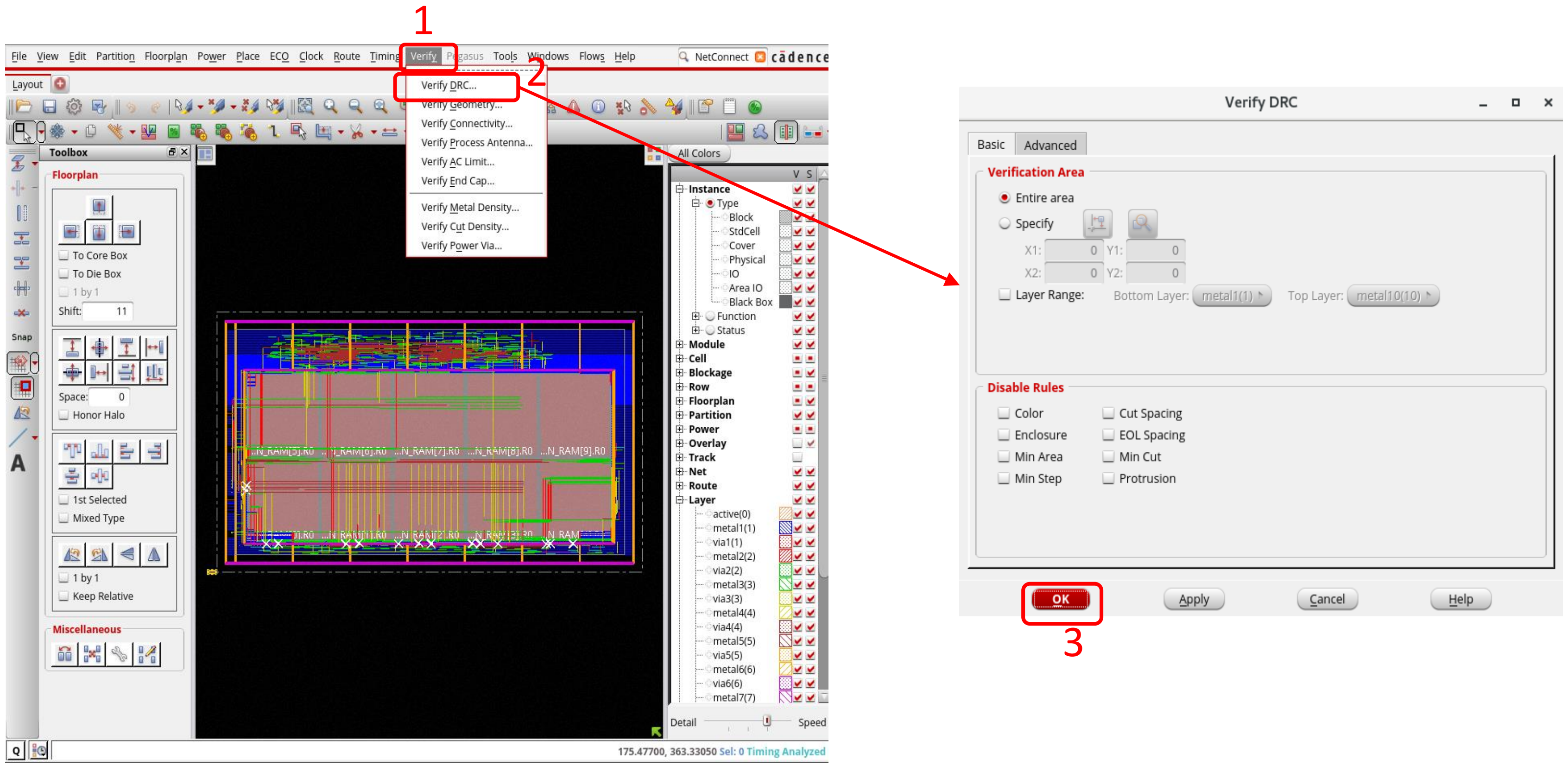
Allow

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

OK | Apply | Reset | Cancel | Help

Verify Connectivity. (1) Click **Verify**-> (2) Click on **Verify Geometry.** (3) Click on **OK**

Step 14: Verification- DRC



Verify DRC

Basic | Advanced

Verification Area

[✓ Entire area]
Specify
X1: 0 Y1: 0
X2: 0 Y2: 0
☐ Layer Range: Bottom Layer: metal1(1) Top Layer: metal10(10)

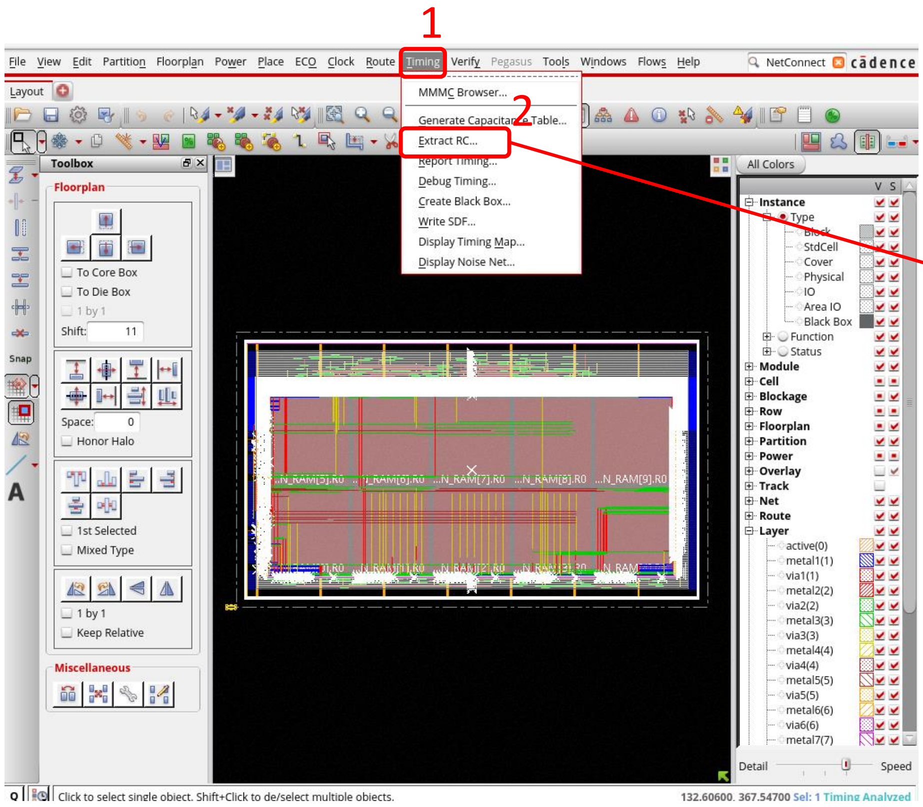
Disable Rules

☐ Color	☐ Cut Spacing
☐ Enclosure	☐ EOL Spacing
☐ Min Area	☐ Min Cut
☐ Min Step	☐ Protrusion

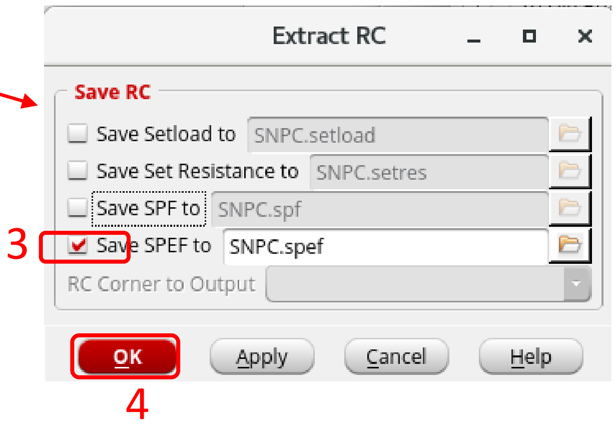
OK Apply Cancel Help

Verify Connectivity. (1) Click **Verify**-> (2) Click on **Verify DRC.** (3) Click on **OK**

Step 15: Output file- SPEF



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

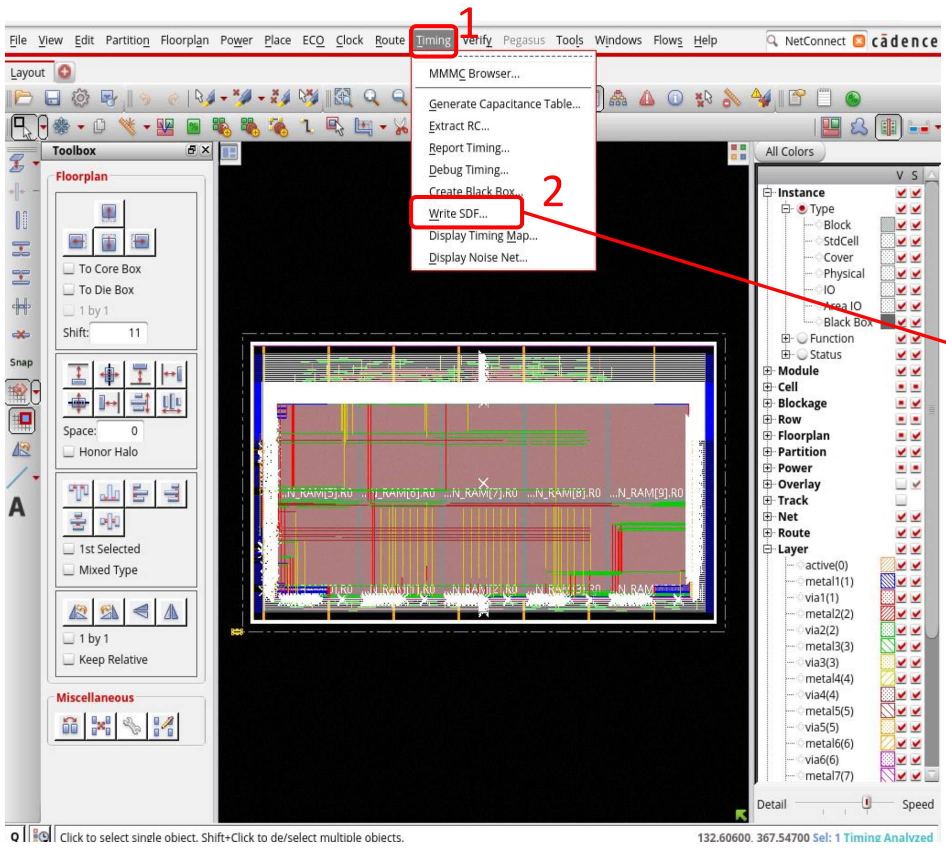


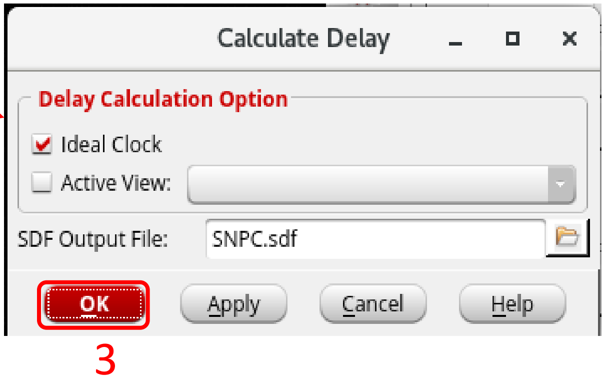
- [ ] Save Setload to SNPC.setload
- [ ] Save Set Resistance to SNPC.setres
- [ ] Save SPF to SNPC.spf
- [✓] Save SPEF to SNPC.spef

RC Corner to Output

Extract Output files: (1) Click on **Timing**-> (2) **Extract RC.** (3) Check **Save SPEF to**. (4) Click **OK**

Step 15: Output File- SDF





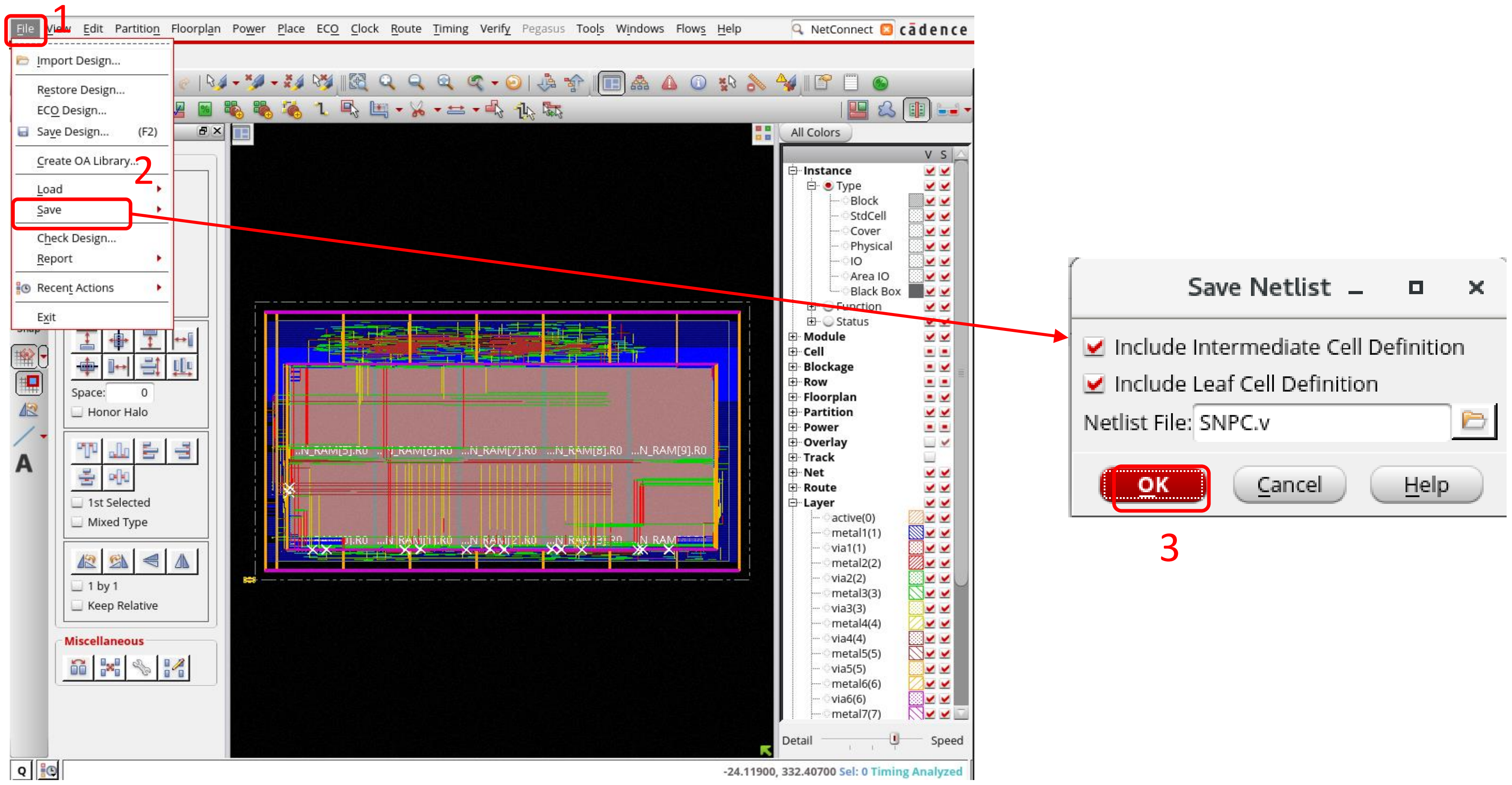
Ideal Clock [✓]

Active View:

SDF Output File: SNPC.sdf

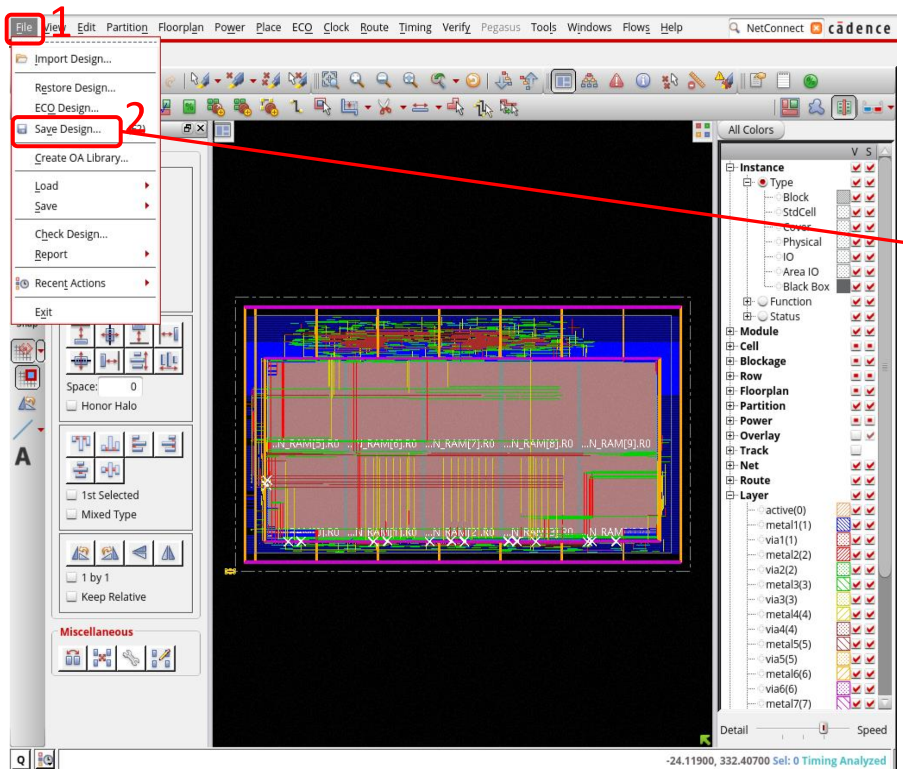
Extract Output files: (1) Click on **Timing**-> (2) **Write SDF.** (3) Click **OK**

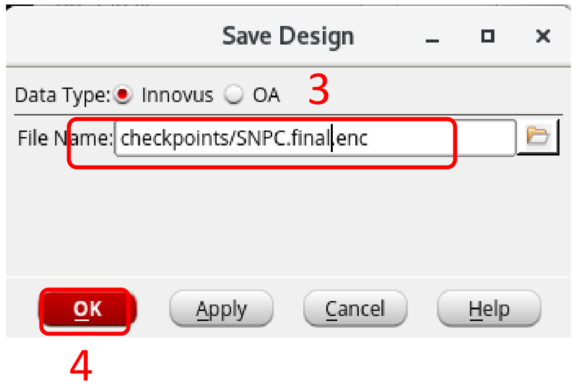
Step 15: Output File- Netlist



Save the progress. (1) Click **File**-> (2) Click on **Save**-> **Netlist** (3) Click on **OK**

Save Final Design- Checkpoint





Data Type: [✓ Innovus] OA

File Name: checkpoints/SNPC.final.enc

Save the progress. (1) Click **File**-> (2) Click on **Save Design** (3) Enter the **checkpoints** directory, and save the design progress as **SNPC.final.enc**. (4) Click on **OK**