

# Openlane-Sky130-workshop

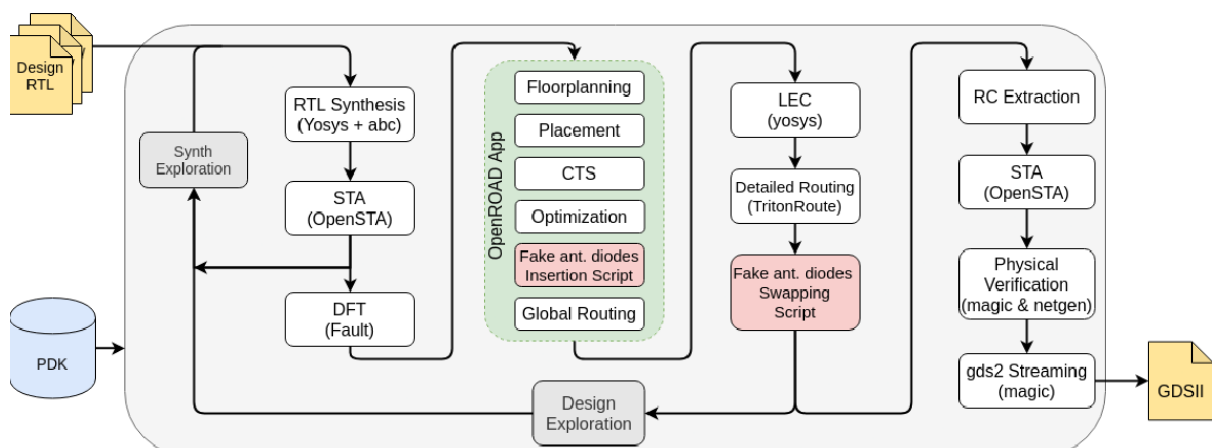
An informative workshop on advanced Physical Design using OpenLANE/Sky130 organized by VSD Corp. The workshop focused on building basics by imparting exposure to conceptual as well as practical approach. Learning's: Open source tools, Characterization of Cell, SPICE simulations, Static timing Analysis concepts, Commands of various OpenLANE tools, 16-mask CMOS Fabrication, Antenna Diode Concept, Decoupling Cap, Routing algorithm & Steps, Timing Characterization and familiarity with files like libs/db, LEF, DEF, SDC and SPEF. Tools used: Magic, Ngspice, OpenSTA, TritonRoute, yosys, SPEF extractor and OpenROAD.

## Contents

1. Introduction to OpenLane.
2. Day 1: Introduction of chip architecture and insights into Physical Design.
3. Day 2: Chip flooplan and Introduction to Library Cells.
4. Day 3: Design and Characterization of cells using Magic Layout tool and Ngspice.
5. Day 4: Timing Analysis using Triton Clock Tree Synthesis (CTS).
6. Day 5: Creation of Power Distribution Network, Routing and SPEF Extraction.

## Introduction to OpenLane

OpenLANE is an automated RTL to GDSII flow based on several components including OpenROAD, Yosys, Magic, Netgen, Fault, OpenPhySyn, SPEF-Extractor and custom methodology scripts for design exploration and optimization. It is a tool started for true open source tape-out experience and comes with APACHE version 2.0. The goal of OpenLANE is to produce clean GDSII without any human intervention. OpenLANE is tuned for Skywater 130nm open PDK and can be used to produce hard macros and chips.



The Flow of OpenLane EDA

## **OpenLane Design Stages**

OpenLane flow consists of several stages. By default all flow steps are run in sequence. Each stage may consist of multiple sub-stages.

### **1. Synthesis**

- a) Yosys/abc - Perform RTL synthesis and technology mapping.
- b) OpenSTA - Performs static timing analysis on the resulting Netlist to generate timing reports

### **2. Floorplaning**

- a) init\_fp - Defines the core area for the macro as well as the rows (used for placement) and the tracks (used for routing)
- b) ioplacer - Places the macro input and output ports
- c) pdngen - Generates the power distribution network
- d) tapcell - Inserts welltap and decap cells in the floorplan

### **3. Placement**

- a) RePLace - Performs global placement
- b) Resizer - Performs optional optimizations on the design
- c) OpenDP - Performs detailed placement to legalize the globally placed components

### **4. CTS**

- a) TritonCTS - Synthesizes the clock distribution network (the clock tree)

### **5. Routing**

- a) FastRoute - Performs global routing to generate a guide file for the detailed router
- b) TritonRoute - Performs detailed routing
- c) OpenRCX - Performs SPEF extraction

### **6. Tapeout**

- a) Magic - Streams out the final GDSII layout file from the routed def
- b) KLayout - Streams out the final GDSII layout file from the routed def as a back-up

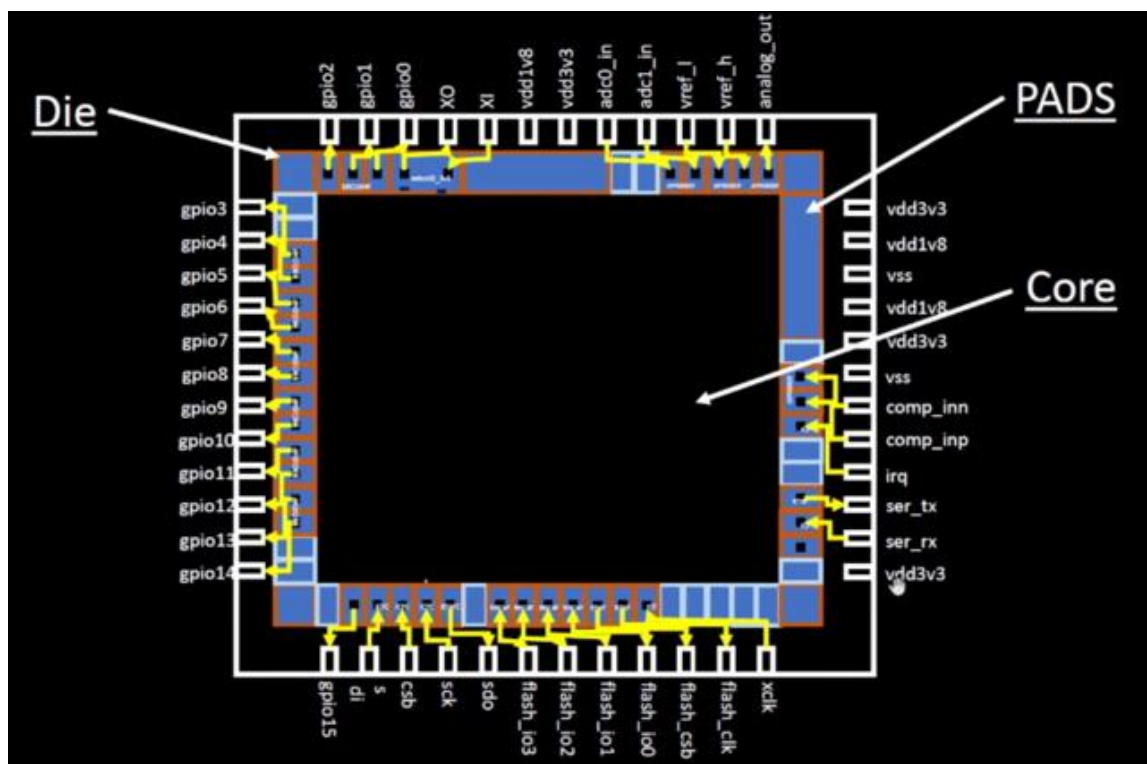
## 7. Signoff

- a) Magic - Performs DRC Checks & Antenna Checks
- b) KLayout - Performs DRC Checks
- c) Netgen - Performs LVS Checks
- d) CVC - Performs Circuit Validity Checks

## Day 1: Introduction of chip architecture and insights into Physical Design.

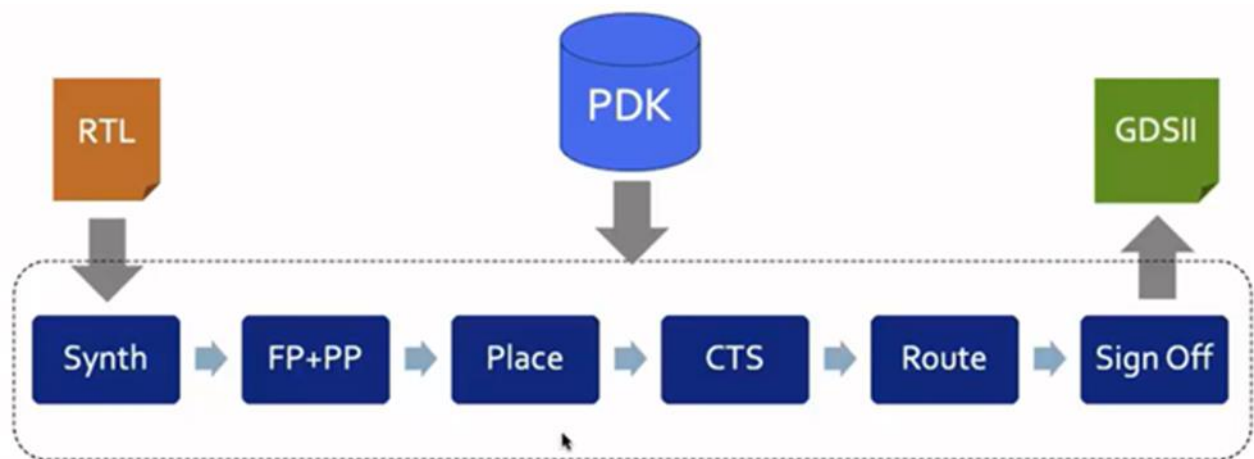
The chip will be sitting at the center of the package. Chip is connected to the package, as shown in the above figure.

- PADS – It is an I/O port which can be used send and receive the signals via PADS.
- CORE – The Foundry IP's as well as many other digital and analog blocks are placed in the chip.
- DIE – The size of the entire chip. Die will be manufactured on the silicon wafer.



## Introduction into Physical Design:

In integrated circuit design, **physical design** is a step in the standard design cycle which follows after the circuit design. At this step, circuit representations of the components (devices and interconnects) of the design are converted into geometric representations of shapes which, when manufactured in the corresponding layers of materials, will ensure the required functioning of the components. This geometric representation is called integrated circuit layout. This step is usually split into several sub-steps, which include both design and verification and validation of the layout.



There are mainly six steps in Physical design:

### 1. Synthesis :

Physical design is based on a netlist which is the end result of the synthesis process. Synthesis converts the RTL design usually coded in VHDL or Verilog HDL to gate-level descriptions which the next set of tools can read/understand. This netlist contains information on the cells used, their interconnections, area used, and other details.

### 2. Floorplanning :

The second step in the physical design flow is floorplanning. Floorplanning is the process of identifying structures that should be placed close together, and allocating space for them in such a manner as to meet the sometimes conflicting goals of available space (cost of the chip), required performance, and the desire to have everything close to everything else. Based on the area of the design and the hierarchy, a suitable floorplan is decided upon. Floorplanning takes into account the macros used in the design, memory, other IP cores and their placement needs, the routing possibilities, and also the area of the entire design. Floorplanning also determines the IO structure and aspect ratio of the design. A bad floorplan will lead to wastage of die area and routing congestion.

### 3. Placement

Placement uses RC values from Virtual Route (VR) to calculate timing. VR is the shortest Manhattan distance between two pins. VR RCs are more accurate than WLM RCs.

Placement is performed in four optimization phases:

- Pre-placement optimization
- In placement optimization
- Post Placement Optimization (PPO) before clock tree synthesis (CTS)
- PPO after CTS.

Pre-placement Optimization optimizes the netlist before placement, HFNs (High Fanout Nets) are collapsed. It can also downsize the cells. In-placement optimization re-optimizes the logic based on VR. This can perform cell sizing, cell moving, cell bypassing, net splitting, gate duplication, buffer insertion, area recovery. Optimization performs iteration of setup fixing, incremental timing and congestion driven placement. Post placement optimization before CTS performs netlist optimization with ideal clocks. It can fix setup, hold, max trans/cap violations. It can do placement optimization based on global routing. It re does HFN synthesis. Post placement optimization after CTS optimizes timing with propagated clock. It tries to preserve clock skew.

#### 4. Clock tree synthesis

The goal of **clock tree synthesis (CTS)** is to minimize skew and insertion delay. Clock is not propagated before CTS as shown in the picture. After CTS hold slack should improve. Clock tree begins at .sdc defined clock source and ends at stop pins of flop. There are two types of stop pins known as ignore pins and sync pins. 'Don't touch' circuits and pins in front end (logic synthesis) are treated as 'ignore' circuits or pins at back end (physical synthesis). 'Ignore' pins are ignored for timing analysis. If clock is divided then separate skew analysis is necessary.

#### 5. Routing


There are two types of routing in the physical design process, global routing and detailed routing. Global routing allocates routing resources that are used for connections. It also does track assignment for a particular net. Detailed routing does the actual connections. Different constraints that are to be taken care during the routing are DRC, wire length, timing etc.

## OpenLANE LAB:

The working Directory will be,

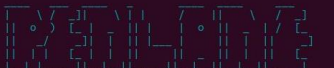
```
vsduser@vsdsquadron:~/Desktop/work/tools/openlane_working_dir/openlane$
```

The commands used enter into OpenLANE directory is,

```
vdsuser@vdsquadron:~/Desktop/work/tools/openlane_working_dir/openlane$ docker
bash-4.25 pwd
/openlane_flow
bash-4.25 ./flow.tcl -interactive
[INFO]:

[INFO]: Version: v0.21
[INFO]: Running interactively
% package require openlane 0.9
0.9
% prep -design picorv32a
[INFO]: Using design configuration at /openlane_flow/designs/picorv32a/config.tcl
[INFO]: Sourcing Configurations from /openlane_flow/designs/picorv32a/config.tcl
[INFO]: PDKs root directory: /home/vdsuser/Desktop/work/tools/openlane_working_dir/pdks
[INFO]: PDK: sky130A
[INFO]: Setting PDKPATH to /home/vdsuser/Desktop/work/tools/openlane_working_dir/pdks/sky130A
[INFO]: Standard Cell Library: sky130_fd_sc_hd
[INFO]: Sourcing Configurations from /openlane_flow/designs/picorv32a/config.tcl
[INFO]: Current run directory is /openlane_flow/designs/picorv32a/runs/02-06-14-46
[INFO]: Preparing LEF Files
[INFO]: Extracting the number of available metal layers from /home/vdsuser/Desktop/work/tools/openlane_working_dir/pdks/sky130A/libraries.ref/sky130_fd_sc_hd/techlef/sky130_fd_sc_hd.tlef
[INFO]: The number of available metal layers is 6
[INFO]: The available metal layers are l1 l1 met1 met2 met3 met4 met5
[INFO]: Merging LEF Files...
mergeLef.py : Merging LEFs
sky130_fd_sc_hd.lef: SITES matched found: 0
sky130_fd_sc_hd.lef: MACROS matched found: 437
sky130_ef_sc_hd_fill_12.lef: SITES matched found: 0
sky130_ef_sc_hd_fill_12.lef: MACROS matched found: 1
sky130_ef_sc_hd_decap_12.lef: SITES matched found: 0
sky130_ef_sc_hd_decap_12.lef: MACROS matched found: 1
sky130_ef_sc_hd_fakediode_2.lef: SITES matched found: 0
sky130_ef_sc_hd_fakediode_2.lef: MACROS matched found: 1
mergeLef.py : Merging LEFs complete
[INFO]: Trimming Liberty...
[INFO]: Generating Exclude List...
[INFO]: Storing configs into config.tcl ...
[INFO]: Preparation complete
%
```

To run synthesis,

Command: run\_synthesis

```
vdsuser@vdsquadron:~/Desktop/work/tools/openlane_working_dir/openlane$ docker
bash-4.25 pwd
/openlane_flow
bash-4.25 ./flow.tcl -interactive
[INFO]:

[INFO]: Version: v0.21
[INFO]: Running interactively
% package require openlane 0.9
0.9
% prep -design picorv32a
[INFO]: Using design configuration at /openlane_flow/designs/picorv32a/config.tcl
[INFO]: Sourcing Configurations from /openlane_flow/designs/picorv32a/config.tcl
[INFO]: PDKs root directory: /home/vdsuser/Desktop/work/tools/openlane_working_dir/pdks
[INFO]: PDK: sky130A
[INFO]: Setting PDKPATH to /home/vdsuser/Desktop/work/tools/openlane_working_dir/pdks/sky130A
[INFO]: Standard Cell Library: sky130_fd_sc_hd
[INFO]: Sourcing Configurations from /openlane_flow/designs/picorv32a/config.tcl
[INFO]: Current run directory is /openlane_flow/designs/picorv32a/runs/02-06-14-46
[INFO]: Preparing LEF Files
[INFO]: Extracting the number of available metal layers from /home/vdsuser/Desktop/work/tools/openlane_working_dir/pdks/sky130A/libraries.ref/sky130_fd_sc_hd/techlef/sky130_fd_sc_hd.tlef
[INFO]: The number of available metal layers is 6
[INFO]: The available metal layers are l1 l1 met1 met2 met3 met4 met5
[INFO]: Merging LEF Files...
mergeLef.py : Merging LEFs
sky130_fd_sc_hd.lef: SITES matched found: 0
sky130_fd_sc_hd.lef: MACROS matched found: 437
sky130_ef_sc_hd_fill_12.lef: SITES matched found: 0
sky130_ef_sc_hd_fill_12.lef: MACROS matched found: 1
sky130_ef_sc_hd_decap_12.lef: SITES matched found: 0
sky130_ef_sc_hd_decap_12.lef: MACROS matched found: 1
sky130_ef_sc_hd_fakediode_2.lef: SITES matched found: 0
sky130_ef_sc_hd_fakediode_2.lef: MACROS matched found: 1
mergeLef.py : Merging LEFs complete
[INFO]: Trimming Liberty...
[INFO]: Generating Exclude List...
[INFO]: Storing configs into config.tcl ...
[INFO]: Preparation complete
% run_synthesis
```

After synthesis step,



```

License of LV9: GNU GPL version 3 <http://gnu.org/licenses/gpl.html>

This is free software, and you are free to change and redistribute it
under certain conditions; type 'show_copyright' for details.
This program comes with ABSOLUTELY NO WARRANTY; for details type 'show_warranty'.
create_clock [get_ports $::env(CLOCK_PORT)] -name $::env(CLOCK_PORT) -period $::env(CLOCK_PERIOD)
set input_delay_value [expr $::env(CLOCK_PERIOD) * $::env(IO_PCT)]
set output_delay_value [expr $::env(CLOCK_PERIOD) * $::env(IO_PCT)]
puts "[INFO]: Setting output delay to: $output_delay_value"
[INFO]: Setting output delay to: 4.946000000000001
puts "[INFO]: Setting input delay to: $input_delay_value"
[INFO]: Setting input delay to: 4.946000000000001
set_max_fanout $::env(SYNTH_MAX_FANOUT) [current_design]
set clk_idx [lsearch [all_inputs] [get_port $::env(CLOCK_PORT)]]
#set rst_idx [lsearch [all_inputs] [get_port resetn]]
set all_inputs_wo_clk [lreplace [all_inputs] $clk_idx $clk_idx]
#set all_inputs_wo_clk_rst [lreplace $all_inputs_wo_clk $rst_idx $rst_idx]
set all_inputs_wo_clk_rst $all_inputs_wo_clk
# correct resetn
set input_delay $input_delay_value -clock [get_clocks $::env(CLOCK_PORT)] $all_inputs_wo_clk_rst
#set input_delay 0.0 -clock [get_clocks $::env(CLOCK_PORT)] {resetn}
set_output_delay $output_delay_value -clock [get_clocks $::env(CLOCK_PORT)] [all_outputs]
# TODO set this as parameter
set_driving_cell -lib_cell $::env(SYNTH_DRIVING_CELL) -pin $::env(SYNTH_DRIVING_CELL_PIN) [all_inputs]
set_cap_load [expr $::env(SYNTH_CAP_LOAD) / 1000.0]
puts "[INFO]: Setting load to: $cap_load"
[INFO]: Setting load to: 0.01765
set_load $cap_load [all_outputs]
tns -711.59
wns -23.89
[INFO]: Synthesis was successful
%

```

## Day 2: Chip floorplan and Introduction to Library Cells

On Day 2, definition of width and height of core and die. Factors like Utilization factor and Aspect ratio important to understand a design were introduced and their effects were discussed. Steps involved to define location of pre-placed cell & its advantage of enhancing reusability and de-coupling capacitor and how they help during switching to avoid failure explained. A fully charged De-coupling cap placed parallel to circuits to ensure proper supply of peak current  $I_{peak}$  by decoupling them from main supply voltage. Hence de-coupling cap ensures proper local communication while multiple Vdd & Vss lines lead to proper global communication avoiding voltage droop and ground bounce conditions. Step of placing logical cell placement blockage to avoid PnR tool to place anything.

### Characterization

Input information required by Characterization software's are PDKs, DRC & LVS rules and spice models. The design steps of it involve *Circuit Design* and *Layout design* characterization. The software GUNA used for characterization. The characterization can be classified as Timing characterization, Power characterization and Noise characterization.

### Characterization flow steps

- Model file of CMOS containing basic property definitions
- Read extracted Spice Netlist
- Recognize the behaviour of cell
- Read the subcircuits
- Attract the power sources
- Apply input or stimulus

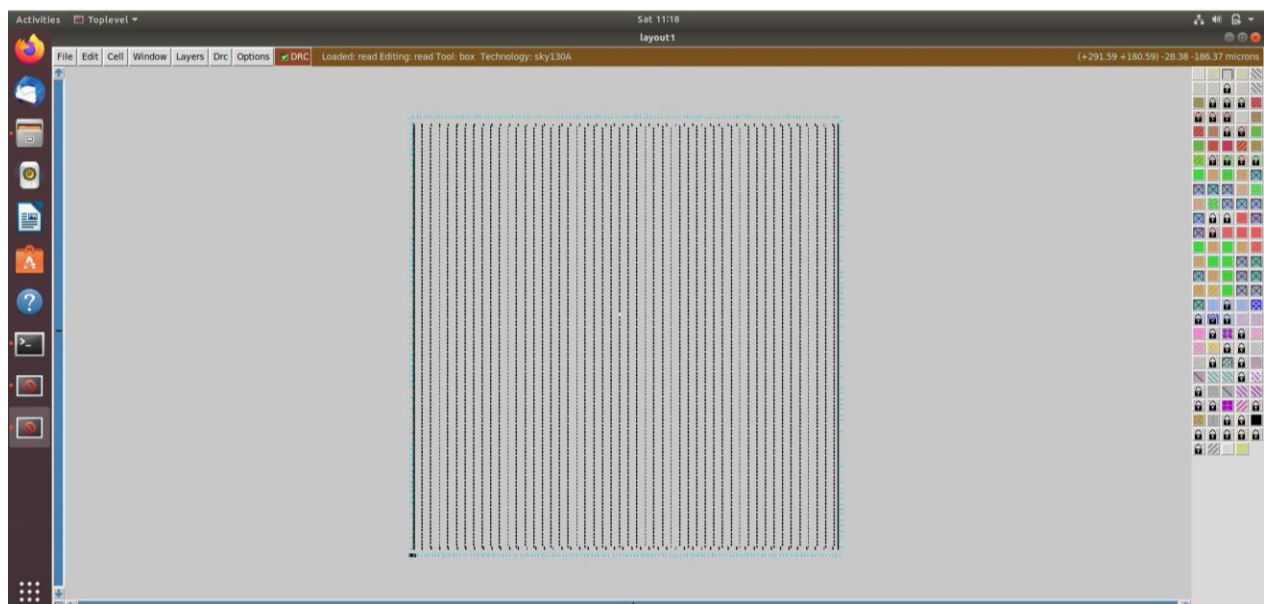
- Provide necessary output capacitance
- Provide necessary simulation commands

To run Floorplan stage

Command : run\_floorplan

## Floorplan Result

```
Running tapcell...
Step 1: Cut rows...
[INFO] Macro blocks found: 0
[INFO] #Original rows: 285
[INFO] #Cut rows: 0
Step 2: Insert endcaps...
[INFO] #Endcaps inserted: 570
Step 3: Insert tapcells...
[INFO] #Tapcells inserted: 10473
Running tapcell... Done!
```



Floorplan of picorv32a

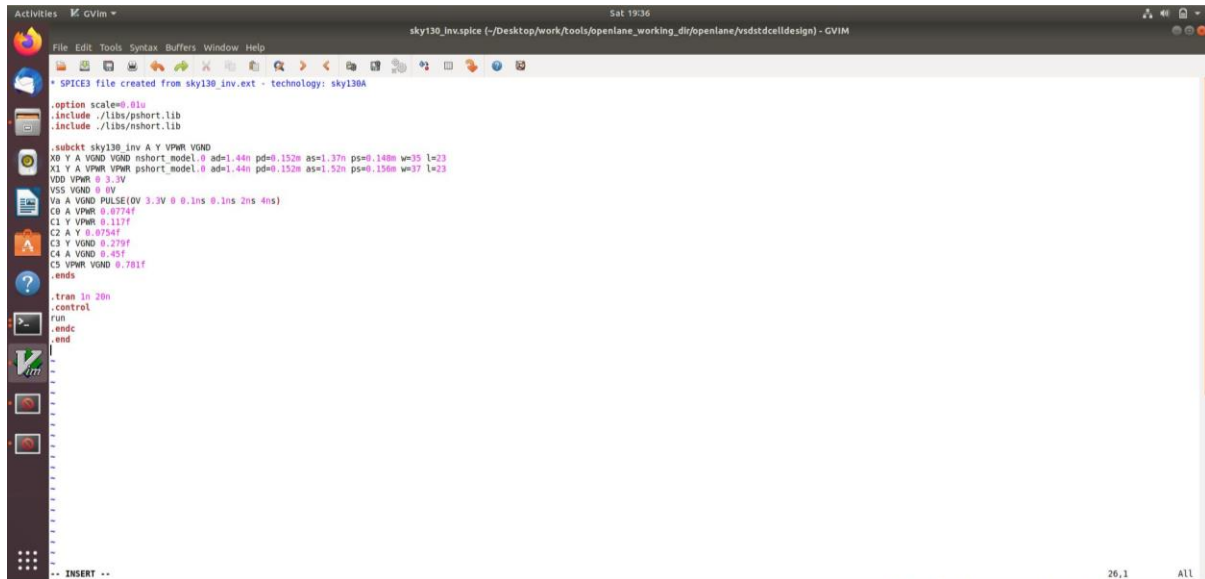
## Day 3: Design and Characterization of cells using Magic Layout tool and Ngspice

OpenLANE offers an interesting feature of making changes into parameters on the go. This helps to deal with issues like congestion. SPICE deck formation contains information like components connectivity and values and information about nodes. It was showed how W/L ratio of MOS impacts its conductivity and hence reason of carefully defining W/L ratio of



MOS to ensure same *rise* and *fall* delay for clock signals. CMOS robustness defined with the help of parameter that is **Switching Threshold (Vm)**. Here name of inverter is vsdshinv.

Spice deck Description of vsdshinv:



```

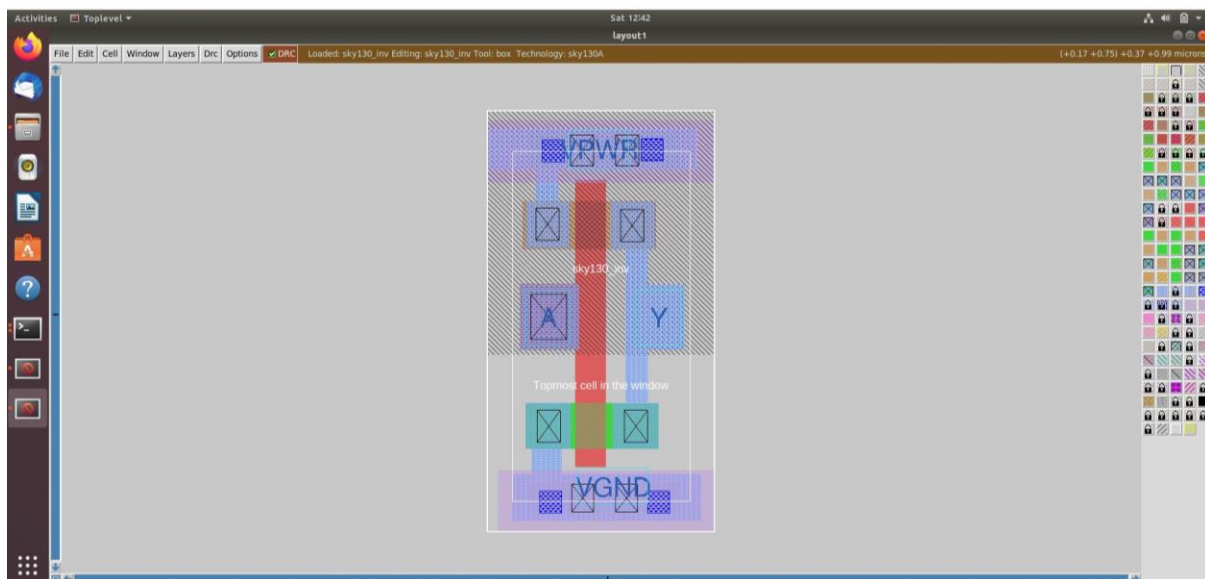
SPICE3 file created from sky130_inv.ext - technology: sky130A

.option scale=0.01u
.include ./libs/pshort.lib
.include ./libs/nshort.lib

subckt sky130_inv A Y VPWR VGND
X0 Y A VGND VSDINV nshort_model.0 ad=1.44n pd=0.152n as=1.37n ps=0.140n w=35 l=23
X1 Y A VPWR VPWR pshort_model.0 ad=1.44n pd=0.152n as=1.52n ps=0.150n w=37 l=23
VDD VPWR 0 3.3V
VSS VGND 0 0V
Va A VGND PULSE(0V 3.3V 0 0.1ns 0.1ns 2ns 4ns)
C0 A VPWR 0.0774f
C1 Y VPWR 0.1177f
C2 A Y 0.0754f
C3 Y VGND 0.279f
C4 A VGND 0.45f
C5 VPWR VGND 0.781f
.ends

.tran in 20n
.control
run
endc
.end
  
```

Layout view of vsdshinv:



## 16-Mask CMOS Process Steps

- **Substrate Selection:** Selection of base layer on which other regions will be formed.
- **Create active region for transistors:** SiO<sub>2</sub> and si<sub>3</sub>N<sub>2</sub> deposited. Pockets created using photoresist and lithography.
- **Nwell & Pwell formation:** Pwell uses boron and nwell uses phosphorous. Drive in diffusion by placing in high temp furnace.

- **Creating Gate terminal:** For desired *threshold value* NA (doping Concentration) and Cox to be set.
- **Lightly Doped Drain (LDD) formation:** LDD done to avoid *hot electron effect* and *short channel effect*.
- **Source and Drain formation:** Forming the source and drain.
- **Contacts & local interconnect Creation:** SiO2 removed using HF etch. *Titanium* deposited using sputtering.
- **Higher Level metal layer formation:** Upper layers of metals deposited.

Ngspice environment:

To enter into Ngspice : command `ngspice file_name.spice`

```

vdsuser@vdsquadron:~/Desktop/work/tools/openlane_working_dir/openlane/vdstddcelldesign$ gvin sky130_inv.spice
vdsuser@vdsquadron:~/Desktop/work/tools/openlane_working_dir/openlane/vdstddcelldesign$ ngspice sky130_inv.spice
*****
** ngspice-27 : Circuit level simulation program
** The U. C. Berkeley CAD Group
** Copyright 1985-1994, Regents of the University of California.
** Please get your ngspice manual from http://ngspice.sourceforge.net/docs.html
** Please file your bug-reports at http://ngspice.sourceforge.net/bugrep.html
** Creation Date: Tue Dec 26 17:10:20 UTC 2017
*****
Circuit: * spice3 file created from sky130_inv.ext - technology: sky130a

Scale set
Error on line 12 :
va a vgnd pulse(ov 3.3v 0 0.1ns 0.1ns 2ns 4ns)
no such parameter on this device
Doing analysis at TEMP = 27.000000 and TNOM = 27.000000

Warning: va: has no value, DC 0 assumed

Initial Transient Solution
-----
Node          Voltage
----          -
y              3.3
a              0
vgnd           0
vpwr           3.3
va#branch      0
vss#branch     3.32336e-12
vdd#branch    -3.32337e-12

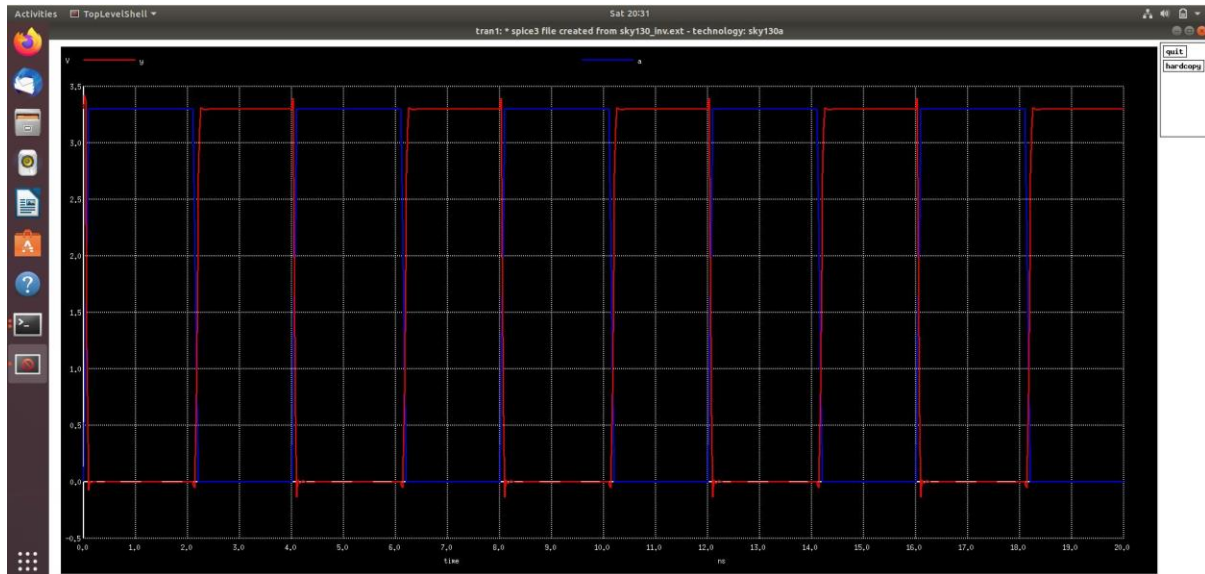
Reference value : 0.00000e+00

*** No. of Data Rows : 59
*** ngspice 1 ->

```

:

Transient analysis of vdsdshinv:



#### Day 4: Timing Analysis using Triton Clock Tree Synthesis (CTS).

Topic of Delay tables explored on Day 5. I learned that delay and output transition values of any particular cell are calculated with the help of values of *input transition* and *output load* values. Delay table and output transition table of a cell contain different value for each combination of input trans and output load, represented in form of lookup tables in liberty file. With the help of interpolation and extrapolation the values of point in range can also be calculated for precise result.

**Setup & Hold Slack Analysis** Setup and hold time define a window of time in which our data should remain unchanged for desired data transfer to take place. Factors like uncertainty and skew also play an important role in this. **Clock skew is the difference between Source Clock path and Destination Clock path.** Slack defined as difference between actual time and required time is monitored. Positive or zero Slack indicates no violation whereas negative slack value indicates violation of timing.

After adding vsdshinv to merged.lef file. The synthesis and floorplan is done again to get better timing results.

```
Activities Terminal v Sun 15:54
vdsuser@vdsquadron: ~/Desktop/work/tools/openlane_working_dir/openlane

File Edit View Search Terminal Help

sky130_fd_sc_hd_nor2_2 524
sky130_fd_sc_hd_nor2b_2 1
sky130_fd_sc_hd_nor3_2 42
sky130_fd_sc_hd_nor4_2 1
sky130_fd_sc_hd_o211a_2 2
sky130_fd_sc_hd_o211a_2 69
sky130_fd_sc_hd_o211a1_2 6
sky130_fd_sc_hd_o21a_2 54
sky130_fd_sc_hd_o21a1_2 141
sky130_fd_sc_hd_o21ba_2 209
sky130_fd_sc_hd_o21ba1_2 1
sky130_fd_sc_hd_o221a_2 204
sky130_fd_sc_hd_o221a1_2 7
sky130_fd_sc_hd_o22a_2 1312
sky130_fd_sc_hd_o22a1_2 59
sky130_fd_sc_hd_o2bb1a_2 119
sky130_fd_sc_hd_o2bb2a1_2 92
sky130_fd_sc_hd_o311a_2 8
sky130_fd_sc_hd_o31a_2 19
sky130_fd_sc_hd_o31a1_2 1
sky130_fd_sc_hd_o32a_2 109
sky130_fd_sc_hd_o41a_2 2
sky130_fd_sc_hd_or2_2 1088
sky130_fd_sc_hd_or2b_2 25
sky130_fd_sc_hd_or3_2 68
sky130_fd_sc_hd_or3b_2 5
sky130_fd_sc_hd_or4_2 93
sky130_fd_sc_hd_or4b_2 6
sky130_fd_sc_hd_or4bb_2 2
sky130_vsdinv 1554

Chip area for module 'picorv32a': 147712.918400

29. Executing Verilog backend.
Dumping module 'picorv32a'.

Warnings: 307 unique messages, 307 total
End of script. Logfile hash: 52fd8d7cf0, CPU: user 48.42s system 0.36s, MEM: 96.04 MB peak
 Yosys 0.9+3621 (git sha1 84e9fa7, gcc 8.3.1 -fPIC -Os)
Time spent: 48% 2x abc (41 sec), 16% 33x opt_expr (14 sec), ...
[INFO]: Changing netlist from 0 to /openLANE_flow/designs/picorv32a/runs/04-06-10-21/results/synthesis/picorv32a.synthesis.v
[INFO]: Running Static Timing Analysis...
[INFO]: current step Index: 2
OpenSTA 2.2.0 38b4d39ad Copyright (c) 2019, Parallax Software, Inc.
License GPLv3: GNU GPL version 3 <http://gnu.org/licenses/gpl.html>
```

## Commands to change configuration variables settings

set ::env(SYNTH\_STRATEGY) 1

set ::env(SYNTH\_SIZING) 1

## OpenROAD Commands

read\_lef /openLANE\_flow/designs/picorv32a/runs/tmp/merged.lef

read\_def /openLANE\_flow/designs/picorv32a/runs/results/cts/picorv32a.cts.def

write\_db pico12.db

read\_verilog

/openLANE\_flow/designs/picorv32a/runs/results/synthesis/picorv32a.synthesis\_cts.v

read\_liberty \$::env(LIB\_SYNTH\_COMPLETE)

link\_design picorv32a

read\_sdc /openLANE\_flow/designs/picorv32a/src/my\_base.sdc

set\_propagated\_clock [all\_clocks]

report\_checks -path\_delay min\_max -digits 4

**sky130\_fd\_sc\_hd** File containing layer information of Cell

```

li1 X 0.23 0.46
li1 Y 0.17 0.34
met1 X 0.17 0.34
met1 Y 0.17 0.34
met2 X 0.23 0.46
met2 Y 0.23 0.46
met3 X 0.34 0.68
met3 Y 0.34 0.68
met4 X 0.46 0.92
met4 Y 0.46 0.92
met5 X 1.70 3.40
met5 Y 1.70 3.40
~
~

```

Placement results of picorv32a:

```

Activities Terminal Sat 11:33
vdsuser@vdsquadron: ~/Desktop/work/tools/openlane_working_dir/openlane

vdsuser@vdsquadron:~/Desktop/work/tools/openlane_working_dir/openlane
[WARNING PSM-0030] Vsrc location at (565.520um, 290.880um) and size =10.000um, is not located on a power stripe. Moving to closest stripe at (567.000um, 333.650um).
[WARNING PSM-0030] Vsrc location at (5.520um, 430.880um) and size =10.000um, is not located on a power stripe. Moving to closest stripe at (5.400um, 486.830um).
[WARNING PSM-0030] Vsrc location at (145.520um, 430.880um) and size =10.000um, is not located on a power stripe. Moving to closest stripe at (145.800um, 486.830um).
[WARNING PSM-0030] Vsrc location at (285.520um, 430.880um) and size =10.000um, is not located on a power stripe. Moving to closest stripe at (286.200um, 486.830um).
[WARNING PSM-0030] Vsrc location at (425.520um, 430.880um) and size =10.000um, is not located on a power stripe. Moving to closest stripe at (426.600um, 486.830um).
[WARNING PSM-0030] Vsrc location at (565.520um, 430.880um) and size =10.000um, is not located on a power stripe. Moving to closest stripe at (567.000um, 486.830um).
[WARNING PSM-0030] Vsrc location at (285.520um, 570.880um) and size =10.000um, is not located on a power stripe. Moving to closest stripe at (286.200um, 640.010um).
[WARNING PSM-0030] Vsrc location at (425.520um, 570.880um) and size =10.000um, is not located on a power stripe. Moving to closest stripe at (426.600um, 640.010um).
[WARNING PSM-0030] Vsrc location at (565.520um, 570.880um) and size =10.000um, is not located on a power stripe. Moving to closest stripe at (567.000um, 640.010um).
[INFO PSM-0031] Number of nodes on net VPWR = 20660.
[INFO PSM-0037] G matrix created successfully.
[INFO PSM-0040] Connection between all PDM nodes established in net VPWR.
[WARNING PSM-0016] Voltage pad location (vsrc) file not specified, defaulting pad location to checkerboard pattern on core area.
[WARNING PSM-0017] X direction bump pitch is not specified, defaulting to 140um.
[WARNING PSM-0018] Y direction bump pitch is not specified, defaulting to 140um.
[WARNING PSM-0019] Voltage on net VGND is not explicitly set.
[WARNING PSM-0021] Using voltage 0.000V for ground network.
[INFO PSM-0026] Creating G matrix.
[INFO PSM-0028] Extracting power stripes on net VGND.
[WARNING PSM-0030] Vsrc location at (5.520um, 10.880um) and size =10.000um, is not located on a power stripe. Moving to closest stripe at (5.400um, 103.880um).
[WARNING PSM-0030] Vsrc location at (145.520um, 10.880um) and size =10.000um, is not located on a power stripe. Moving to closest stripe at (145.800um, 103.880um).
[WARNING PSM-0030] Vsrc location at (285.520um, 10.880um) and size =10.000um, is not located on a power stripe. Moving to closest stripe at (286.200um, 103.880um).
[WARNING PSM-0030] Vsrc location at (425.520um, 10.880um) and size =10.000um, is not located on a power stripe. Moving to closest stripe at (426.600um, 103.880um).
[WARNING PSM-0030] Vsrc location at (565.520um, 10.880um) and size =10.000um, is not located on a power stripe. Moving to closest stripe at (567.000um, 103.880um).
[WARNING PSM-0030] Vsrc location at (285.520um, 150.880um) and size =10.000um, is not located on a power stripe. Moving to closest stripe at (286.200um, 103.880um).
[WARNING PSM-0030] Vsrc location at (425.520um, 150.880um) and size =10.000um, is not located on a power stripe. Moving to closest stripe at (426.600um, 103.880um).
[WARNING PSM-0030] Vsrc location at (565.520um, 150.880um) and size =10.000um, is not located on a power stripe. Moving to closest stripe at (567.000um, 103.880um).
[WARNING PSM-0030] Vsrc location at (5.520um, 290.880um) and size =10.000um, is not located on a power stripe. Moving to closest stripe at (5.400um, 257.060um).
[WARNING PSM-0030] Vsrc location at (145.520um, 430.880um) and size =10.000um, is not located on a power stripe. Moving to closest stripe at (145.800um, 410.240um).
[WARNING PSM-0030] Vsrc location at (285.520um, 430.880um) and size =10.000um, is not located on a power stripe. Moving to closest stripe at (286.200um, 410.240um).
[WARNING PSM-0030] Vsrc location at (425.520um, 430.880um) and size =10.000um, is not located on a power stripe. Moving to closest stripe at (426.600um, 410.240um).
[WARNING PSM-0030] Vsrc location at (565.520um, 430.880um) and size =10.000um, is not located on a power stripe. Moving to closest stripe at (567.000um, 410.240um).
[WARNING PSM-0030] Vsrc location at (285.520um, 570.880um) and size =10.000um, is not located on a power stripe. Moving to closest stripe at (286.200um, 563.420um).
[WARNING PSM-0030] Vsrc location at (425.520um, 570.880um) and size =10.000um, is not located on a power stripe. Moving to closest stripe at (426.600um, 563.420um).
[WARNING PSM-0030] Vsrc location at (565.520um, 570.880um) and size =10.000um, is not located on a power stripe. Moving to closest stripe at (567.000um, 563.420um).
[INFO PSM-0031] Number of nodes on net VGND = 19223.
[INFO PSM-0037] G matrix created successfully.
[INFO PSM-0040] Connection between all PDM nodes established in net VGND.
[INFO] PDM generation was successful.
[INFO] Changing layout from /openLANE_flow/designs/plcorv32a/runs/03-06_05-05/results/floorplan/plcorv32a.floorplan.def to /openLANE_flow/designs/plcorv32a/runs/03-06_05-05/tmp/floorplan7-pdn.def
1
% run_placement

```

After the run:



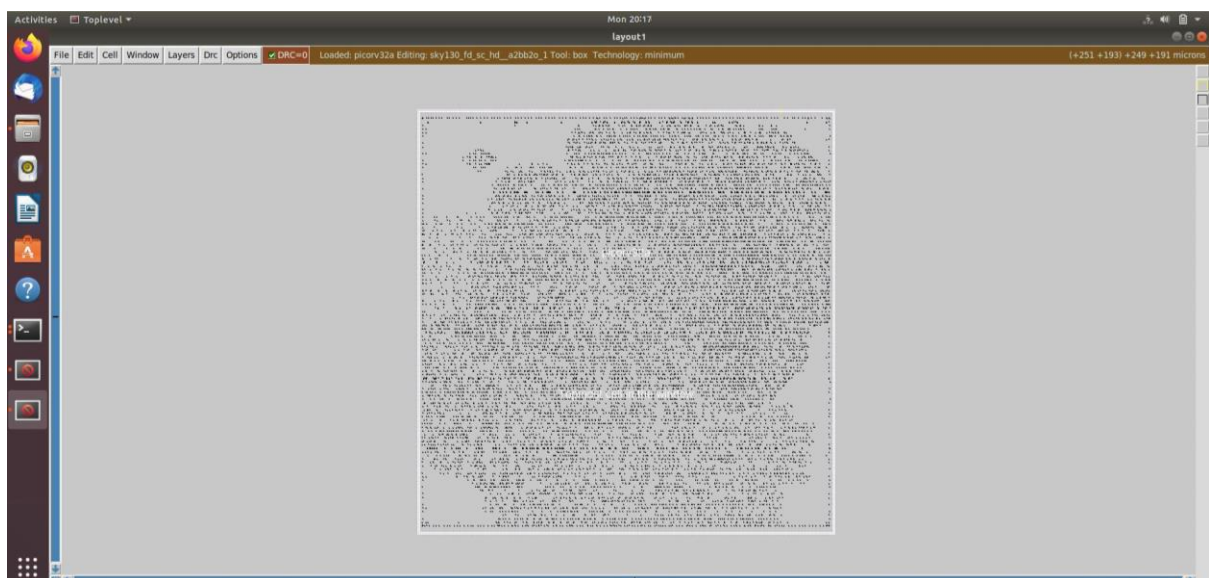
```
Terminal
Mon 19:32
vdsuser@vdsquadron: ~/Desktop/work/tools/openlane_working_dir/openlane

File Edit View Search Terminal Tabs Help
vdsuser@vdsquadron: ~/Desktop/work/tools/openlane_working_dir/openlane
vdsuser@vdsquadron: ~/Desktop/work/tools/openlane_working_dir/openlane/designs/picorv32a

multi cells      : 0
fixed cells      : 0
total nets       : 14978
design area       : 4.05209e+11
total f_area     : 0
total m_area     : 1.4188e+11
design util       : 35.614
num rows         : 234
row height       : 2720

-----
non_group_cell_region_assign done ..
non_group_cell_placement done ..
-----
Reading /openLANE_flow/designs/picorv32a/runs/04-06_07-58/tmp/placement/5-replace.def is Done
DEF file write success !!
location : /openLANE_flow/designs/picorv32a/runs/04-06_07-58/results/placement/picorv32a.placement.def
-----
Tasks
Parser          1.145    1.100
resgn assign     1.149    1.100
pre-placement    1.149    1.100
non Group cell placement 1.162    1.110
All              1.165    1.110
-----
EVALUATION
AVG_displacement : 2242.55
SUM_displacement : 2.98753e+07
MAX_displacement : 21880
-----
GP_HPWL         : 852337
HPWL             : 878577
avg_Disp_site   : 4.87512
avg_Disp_row    : 0.824468
delta_HPWL       : 4.0225
==== CHECK LEGALITY ====
row_check ==> PASS
site_check ==> PASS
power_check ==> PASS
edge_check ==> PASS
placed_check ==> PASS
overlap_check ==> PASS
-----
[INFO]: Changing layout from /openLANE_flow/designs/picorv32a/runs/04-06_07-58/tmp/placement/5-replace.def to /openLANE_flow/designs/picorv32a/runs/04-06_07-58/results/placement/picorv32a.placement.def
```

Layout view of placement.def,



CTS run of Picorv32a:





## Day 5: Creation of Power Distribution Network, Routing and SPEF Extraction.

Routing method finds out best possible pattern for connection between two end points, of which one point is *target node* while the other is *source node*. **Maze Routing-Lee's Algorithm** was introduced. In this technique firstly routing grids are created and source & target nodes are identified. Then the blocks adjacent to one under consideration are assigned same numbers and this process is repeated till we reach the target node. Once this done the pattern with minimum number of turns preferably a *L* spaced pattern is finalised for route.

### Typical DRC rules for pair of wires

1. Wire width
2. Wire pitch
3. Wire Spacing

These DRC rules exist because of limitations of the Lithography technique. Deviation takes place in lithography leading to changes which might lead to an unintended open circuit or short circuit and to avoid this the DRC rules are fixed. Another DRC Violation type is Signal short which can be dealt by changing layers.

### Routing Technique Classified into

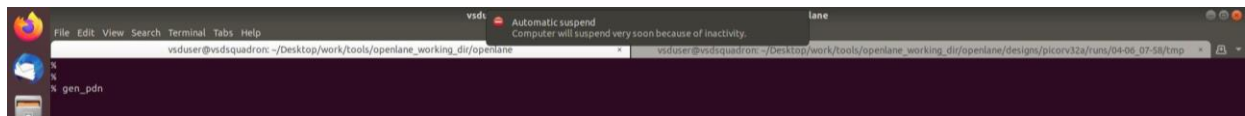
- **Global Route** : Performed using fast route. In this step route guides are formed.
- **Detailed Route** : Performed using *TritonRoute*.

**Triton Route** It performs *initial detailed route* and tries to route within the route guide provided by fast route. It works on MILF-based panel routing with intra-layer parallel route and inter-layer sequential route technique. Input files required for triton route are LEF,DEF and pre-processed route guide. Output is in form of detailed routing with optimum wire length and Via count.

### Steps for pre-processed route guides

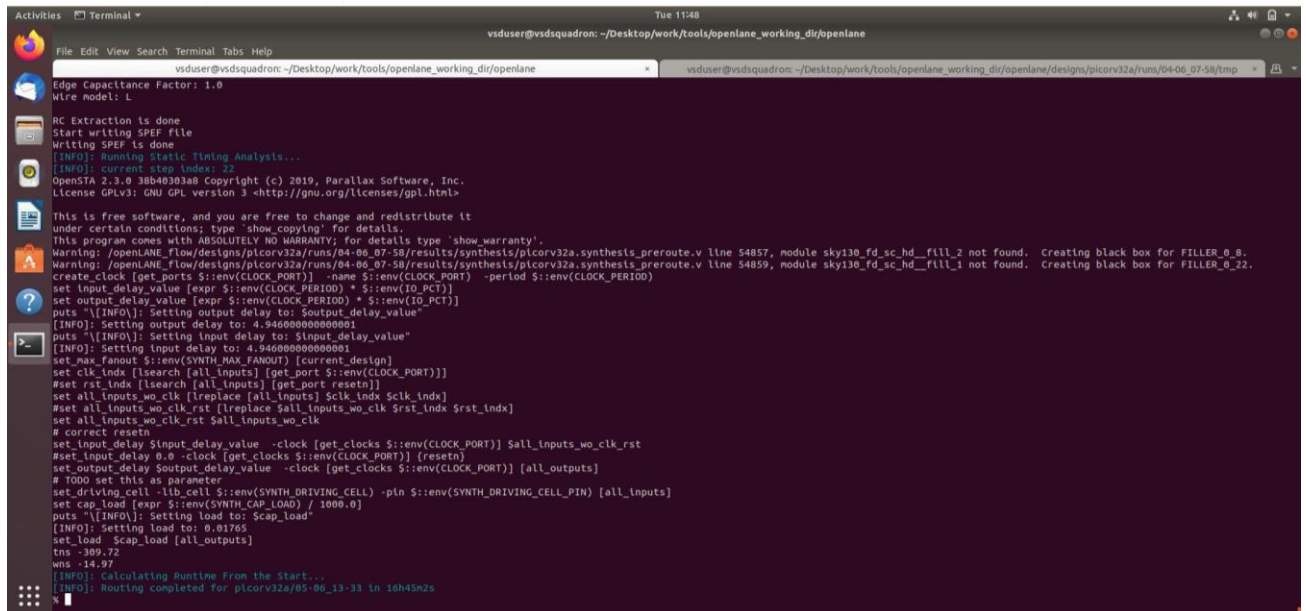
- **Initial route guide** : Basic connecting information using route guides.
- **Splitting** : Routes in non-preferred direction are split into unit width.
- **Merging** : Touching guides have edge orthogonal to the preferred route guide direction are merged.
- **Bridging** : Edges parallel to preferred one are bridged using additional layers.
- **Pre-processed Route** : Now all route guides are in preferred direction as required.

Power Planning of picorv32a,

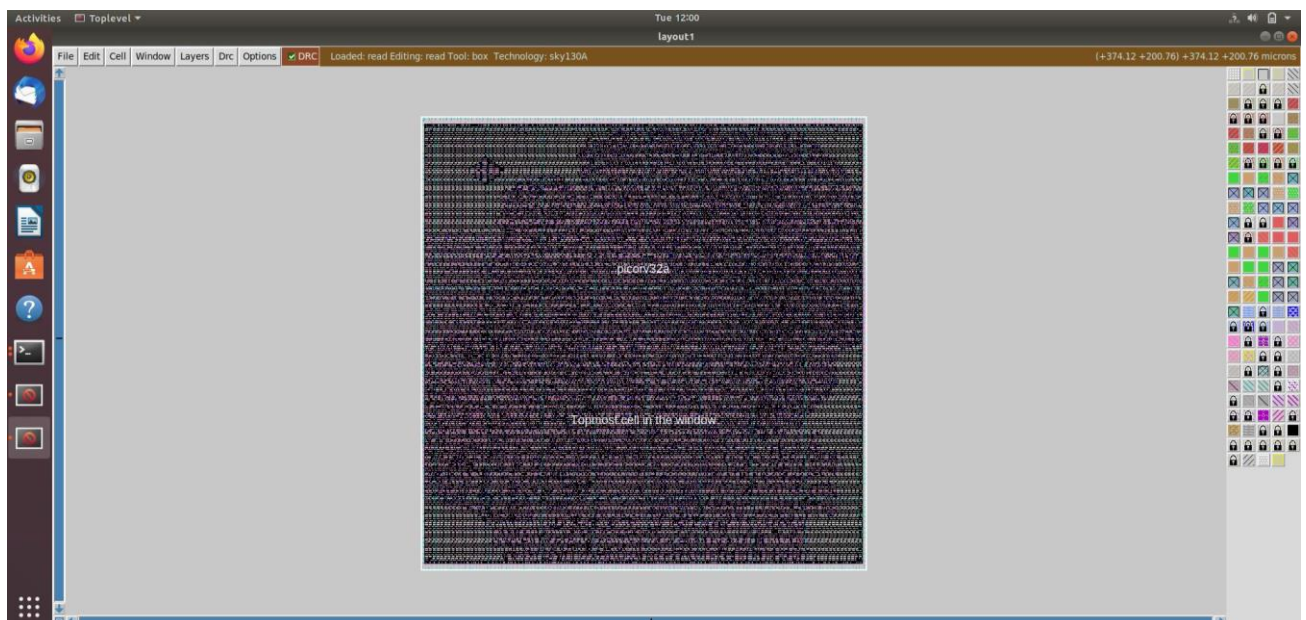


Routing result of picorv32a:

Command : run\_routing

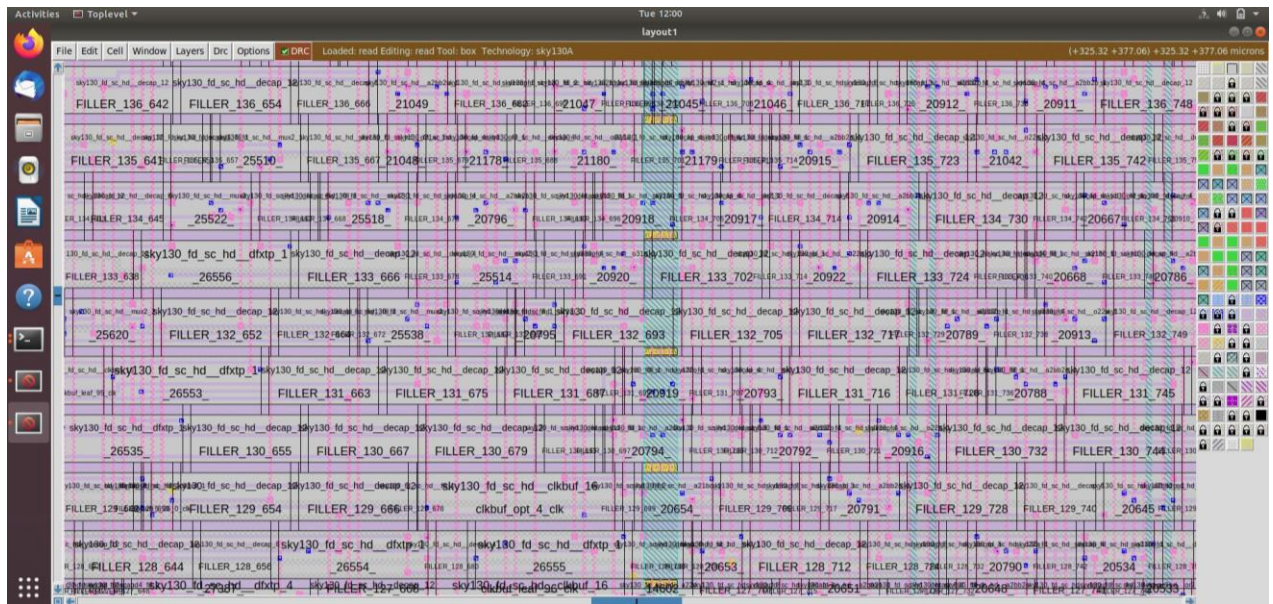


Layout view of picorv32a\_routing.def,



Detailed view of design,





## Acknowledgements

- Kunal Ghosh, Co-founder (VSD Corp. Pvt. Ltd).
- Nickson Jose - Workshop Instructor