

VSD Sky130 OpenLANE Workshop – Advanced Physical Design Using OpenLANE / Sky 130

This repository contains all the information included in the Advanced Physics Design Using OpenLANE / Sky 130 workshop. This workshop helped me in learning ASIC implementation steps from RTL to GDSII flow using open source EDA tools and Google – Skywater technology PDK. Google – Skywater130 PDK is the first manufacturable 130nm opensource PKD.

Table of Content

1. Introduction to OpenLANE
2. Day 1: Inception of Open-Source EDA, OpenLANE & Sky130 PDK
3. Day 2: Good Floor planning Vs Bad Floor planning and Introduction to library cells
4. Day 3: Design library cell using Magic layout and ngSpice characterization
5. Day 4: Pre-Layout timing analysis and importance of good clock tree
6. Day 5: Final steps for RTL2GDS using TritonRoute and OpenSTA
7. Acknowledgements

1. Introduction to OpenLANE

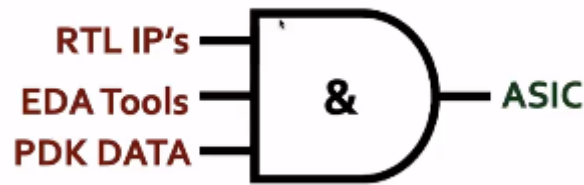
OpenLANE is an automated RTL to GDSII flow which includes different opensource tools like OpenROAD, Yosys, Magic, Netgen, Fault etc., apart from many custom methodology scripts for design exploration and optimization. It is an open-source flow for a true open source tape-out experiment. Its main goal is to produce a clean GDSII with no human interaction. It is tuned for Skywater 130nm Open PDK. It can be used to harden macros & chips.

It operates in 2 modes: Autonomous & Interactive.

2. Day 1: Inception of Open-Source EDA, OpenLANE & Sky130 PDK

Day1 starts with introduction to System on Chip (SoC) basics like chip, PADs, core, die etc., and RISC-V Instruction Set Architecture (ISA). ISA is a language of the computer and the way we talk to the computer. In RISC-V to layout flow, the code in the RISC-V Architecture is implemented using RTL and converts code into Hardware Description Language (HDL) then code is converted to layout using the general RTL to GSD flow.

For open-source implementation of Digital ASIC Design, we need 3 components:

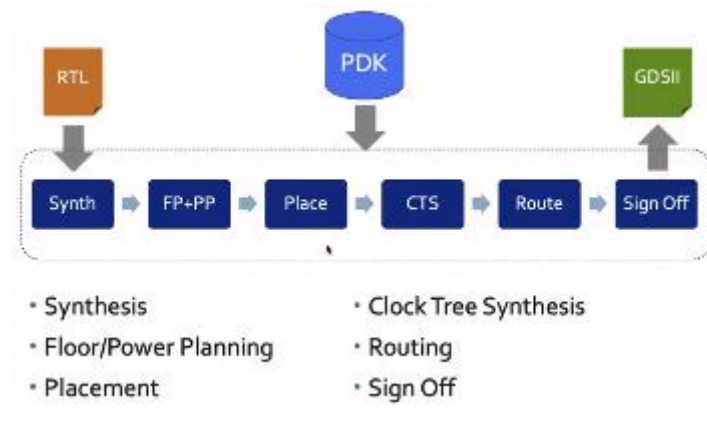


RTL IP's: these are easily available on open-source platforms like librecores.org, opencores.org, github.com etc.,

EAD Tools: QFlow, OpenROAD, OpenLANE are open-source EDA tools available.

PDK DATA: Open-source PDK released by Google + Skywater technology.

Simplified RTL to GDS flow:



- **Synthesis**

Design RTL is translated into circuits made out of components from Standard Cell Library (SCL). The resultant circuit is described in HDL and usually refers to as the gate level netlist. The gate level netlist is functionally equal to the RTL.

- **Floor Planning and Power Planning (FP & PP)**

Floor planning and power planning varied depending upon is we are implementing on one component (macro floor planning) or a whole chip. The main objective here is to plan a silicon area & create robust power distribution network to power the circuits.

In Macro floor planning, macro dimension, pin location, rows & routing tracks are defined.

In power planning, power network is constructed using power pads, power straps and power rings.

- **Placement**

In Placement, gate level netlist cells are replaced by macros. Placement of the cells on the floorplan, aligned with the sites is called placement, it helps to interconnect the delay and also enable successful routing. Usually placement is done in 2 steps: Global placement and detailed placement.

Global Placement: It tries to find the optimal placements for all cells, such placement is legal which means there can be some cell overlaps.

Detailed Placement: In detailed Placement, positions obtained from the global placement are minimally altered to be legal.

- Clock Tree Synthesis (CTS)

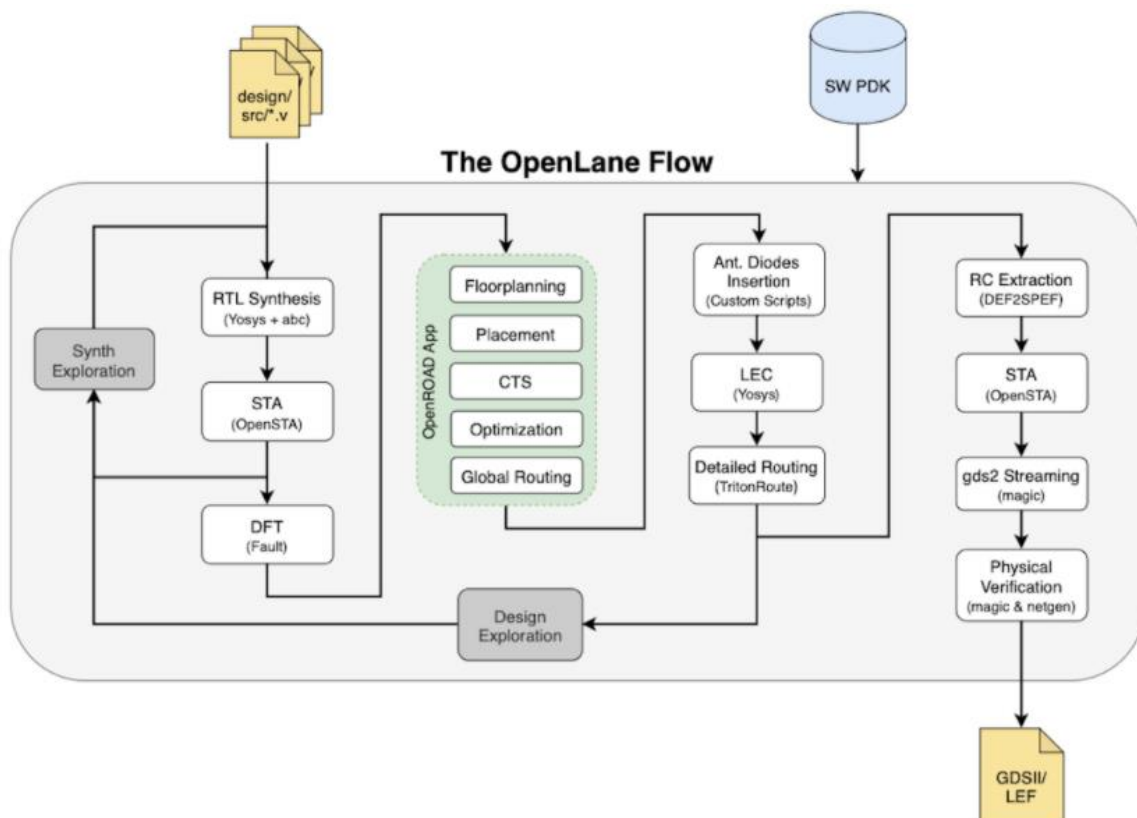
This step is used to route the clock before the signal are routed. By creating a clock distribution network to deliver the clock to all sequential elements, with minimal skew and in a good shape. The clock network is usually in a shape of tree.

- Routing

After clock routing now it comes to signal routing. In Routing, valid horizontal and vertical pattern of wires are found to implement net connecting cells together using available metal layers. Routing is performed in 2 steps: Global routing – generates routing guides and detailed routing – uses the routing guides to implement the actual wiring.

- Sign-Off

In sign-off stage *Physical verification* is done using Design Rule Checking (DRC) & Layout Vs Schematic (LVS) and *Timing verification* is done using Static Timing Analysis (STA).

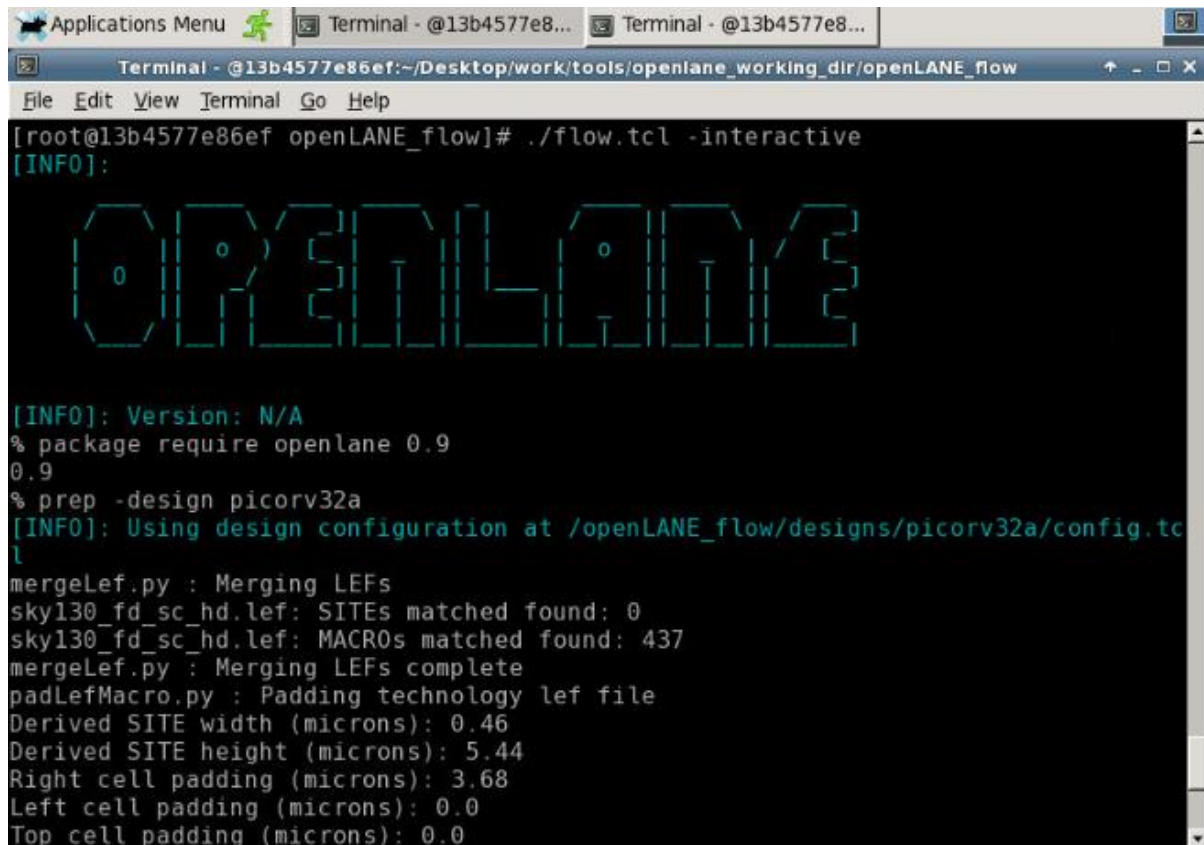


Above picture shows the detailed ASIC design flow. Here new concepts like design exploration, Design for test (DFT), logic equivalence check (LEC), antenna diodes insertion was explained.

Design exploration helps in finding the best configuration for the design. DFT is useful to make sure our design is ready for testing after fabrication. As optimization is performed in physical design step, which can form change in the gate level netlist (netlist generated by the synthesis step), we need to perform LEC. Every time netlist changes, verification must be performed. LEC formally confirms the function did not change after modifying the netlist. Antenna Diode problem arises for long nets which is avoided by using fake antenna diodes next to every cell after placement. Later Antenna Checker (Magic) is run. If it reports antenna violations then fake diode replaced by actual one.

LAB:

To run OpenLANE

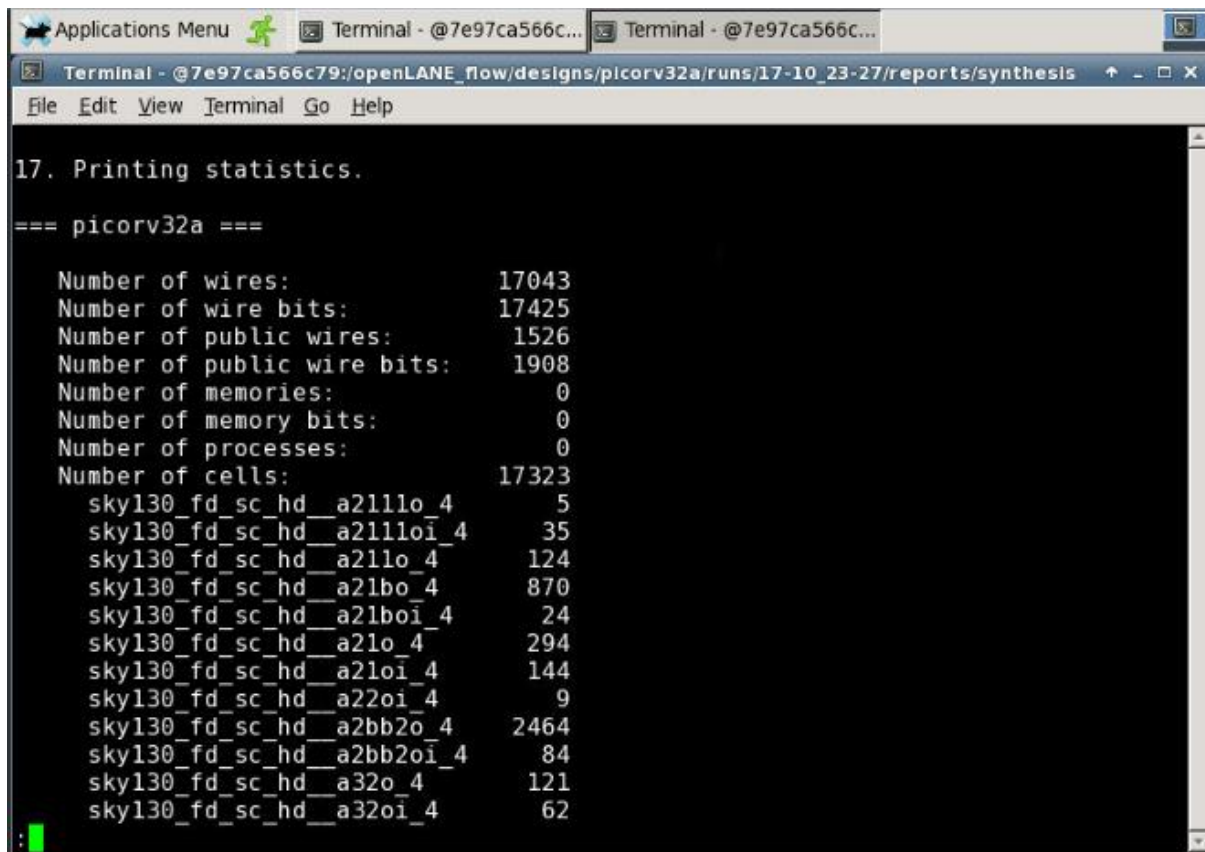


```
Terminal - @13b4577e86ef:~/Desktop/work/tools/openlane_working_dir/openLANE_flow
File Edit View Terminal Go Help
[root@13b4577e86ef openLANE_flow]# ./flow.tcl -interactive
[INFO]:

  O P E N L A N E

[INFO]: Version: N/A
% package require openlane 0.9
0.9
% prep -design picorv32a
[INFO]: Using design configuration at /openLANE_flow/designs/picorv32a/config.tcl
mergeLef.py : Merging LEFs
skyl30_fd_sc_hd.lef: SITES matched found: 0
skyl30_fd_sc_hd.lef: MACROs matched found: 437
mergeLef.py : Merging LEFs complete
padLefMacro.py : Padding technology lef file
Derived SITE width (microns): 0.46
Derived SITE height (microns): 5.44
Right cell padding (microns): 3.68
Left cell padding (microns): 0.0
Top cell padding (microns): 0.0
```

Results of synthesis showing design components



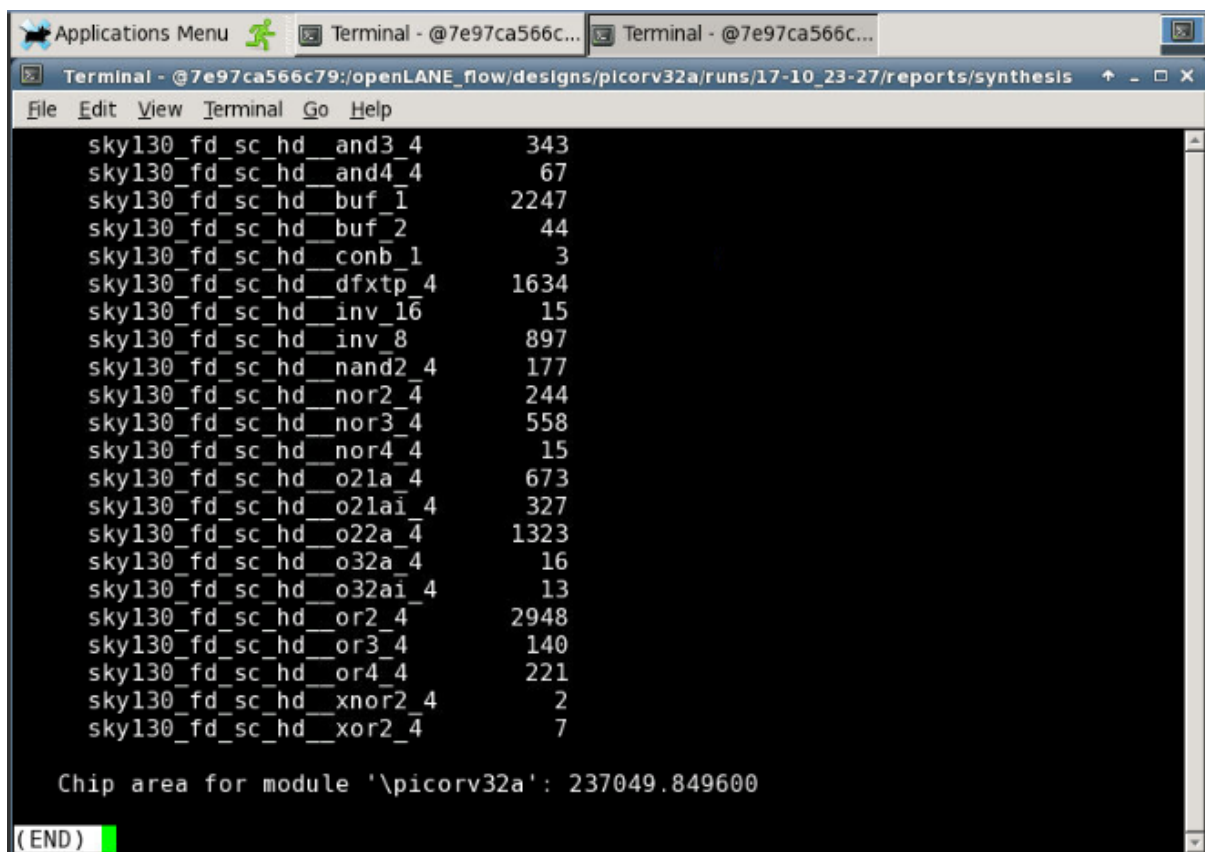
```
Applications Menu Terminal - @7e97ca566c... Terminal - @7e97ca566c...
Terminal - @7e97ca566c79:/openLANE_flow/designs/picorv32a/runs/17-10_23-27/reports/synthesis
File Edit View Terminal Go Help

17. Printing statistics.

=== picorv32a ===

Number of wires:          17043
Number of wire bits:      17425
Number of public wires:   1526
Number of public wire bits: 1908
Number of memories:       0
Number of memory bits:    0
Number of processes:      0
Number of cells:          17323
  skyl130_fd_sc_hd__a211lo_4      5
  skyl130_fd_sc_hd__a211loi_4     35
  skyl130_fd_sc_hd__a21lo_4      124
  skyl130_fd_sc_hd__a21bo_4      870
  skyl130_fd_sc_hd__a21boi_4     24
  skyl130_fd_sc_hd__a21lo_4      294
  skyl130_fd_sc_hd__a21oi_4      144
  skyl130_fd_sc_hd__a22oi_4       9
  skyl130_fd_sc_hd__a2bb2o_4    2464
  skyl130_fd_sc_hd__a2bb2oi_4    84
  skyl130_fd_sc_hd__a32o_4      121
  skyl130_fd_sc_hd__a32oi_4      62
:
```

Synthesis showing chip area and number of D F/F's



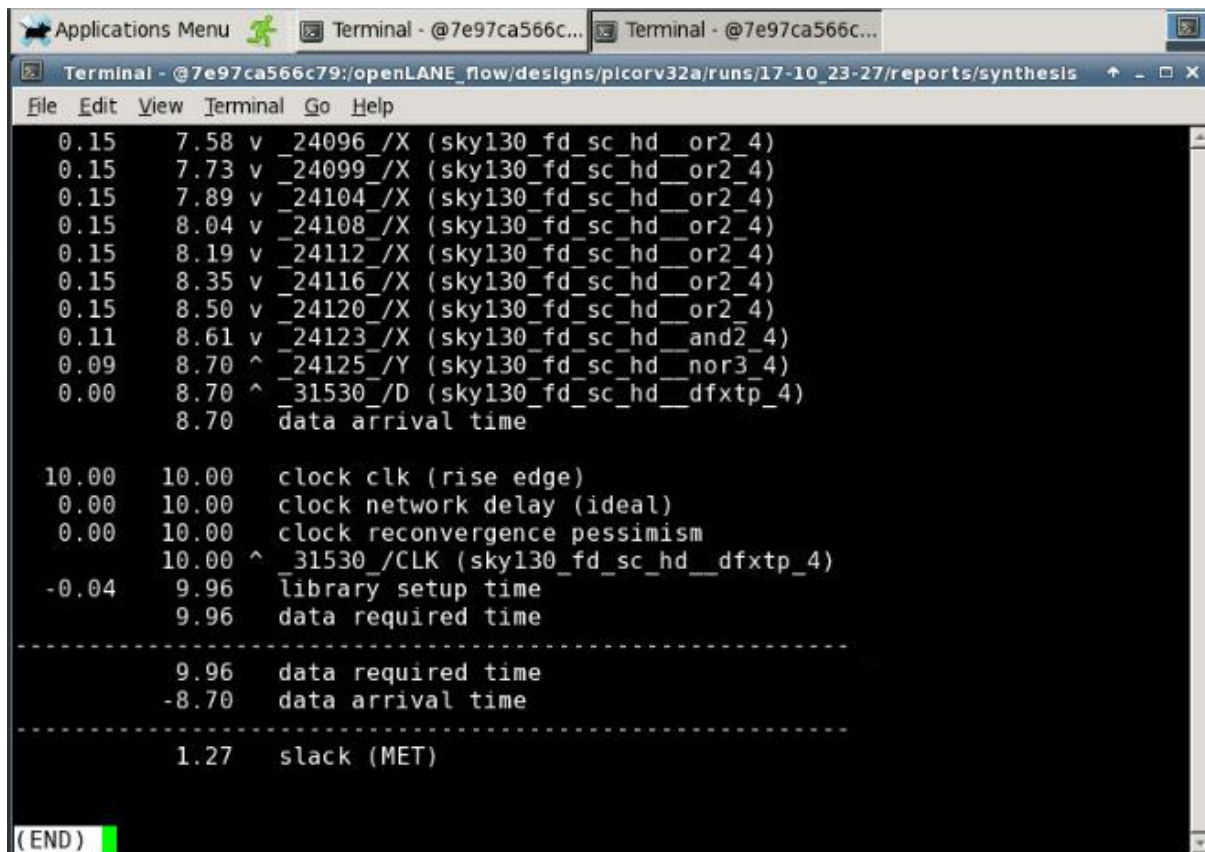
```
Applications Menu Terminal - @7e97ca566c... Terminal - @7e97ca566c...
Terminal - @7e97ca566c79:/openLANE_flow/designs/picorv32a/runs/17-10_23-27/reports/synthesis
File Edit View Terminal Go Help

  skyl130_fd_sc_hd__and3_4      343
  skyl130_fd_sc_hd__and4_4       67
  skyl130_fd_sc_hd__buf_1      2247
  skyl130_fd_sc_hd__buf_2       44
  skyl130_fd_sc_hd__conb_1       3
  skyl130_fd_sc_hd__dfxtp_4    1634
  skyl130_fd_sc_hd__inv_16       15
  skyl130_fd_sc_hd__inv_8      897
  skyl130_fd_sc_hd__nand2_4     177
  skyl130_fd_sc_hd__nor2_4     244
  skyl130_fd_sc_hd__nor3_4     558
  skyl130_fd_sc_hd__nor4_4       15
  skyl130_fd_sc_hd__o21a_4      673
  skyl130_fd_sc_hd__o21ai_4     327
  skyl130_fd_sc_hd__o22a_4    1323
  skyl130_fd_sc_hd__o32a_4       16
  skyl130_fd_sc_hd__o32ai_4      13
  skyl130_fd_sc_hd__or2_4     2948
  skyl130_fd_sc_hd__or3_4      140
  skyl130_fd_sc_hd__or4_4      221
  skyl130_fd_sc_hd__xnor2_4       2
  skyl130_fd_sc_hd__xor2_4       7

Chip area for module 'picorv32a': 237049.849600

(END)
```


Slack calculation of design



```
0.15 7.58 v _24096_/X (sky130_fd_sc_hd__or2_4)
0.15 7.73 v _24099_/X (sky130_fd_sc_hd__or2_4)
0.15 7.89 v _24104_/X (sky130_fd_sc_hd__or2_4)
0.15 8.04 v _24108_/X (sky130_fd_sc_hd__or2_4)
0.15 8.19 v _24112_/X (sky130_fd_sc_hd__or2_4)
0.15 8.35 v _24116_/X (sky130_fd_sc_hd__or2_4)
0.15 8.50 v _24120_/X (sky130_fd_sc_hd__or2_4)
0.11 8.61 v _24123_/X (sky130_fd_sc_hd__and2_4)
0.09 8.70 ^ _24125_/Y (sky130_fd_sc_hd__nor3_4)
0.00 8.70 ^ _31530_/D (sky130_fd_sc_hd__dfxtp_4)
      8.70 data arrival time

10.00 10.00 clock clk (rise edge)
0.00 10.00 clock network delay (ideal)
0.00 10.00 clock reconvergence pessimism
      10.00 ^ _31530_/CLK (sky130_fd_sc_hd__dfxtp_4)
-0.04 9.96 library setup time
      9.96 data required time
-----
      9.96 data required time
     -8.70 data arrival time
-----
      1.27 slack (MET)

(END)
```

3. Day 2: Good Floor planning Vs Bad Floor planning and Introduction to library cells

On Day 2, definition of width and height of core and die were explained. Later, the concepts like *Utilization factor* and *Aspect ratio* were explained in detail and their importance to understand the design was also mentioned.

$$\text{Utilization factor} = (\text{Area occupied by netlist}) / (\text{Total area of the core})$$

Ideally, utilization is considered to be 50% - 60% and utilization factor to be 0.5 / 0.6.

$$\text{Aspect ratio} = (\text{Height (of the die)}) / (\text{Width (of the die)})$$

Steps involved to define *Pre-placed* cell & its advantage of enhancing reusability and *de-coupling capacitors* and how they help during switching to avoid failure was explained. Once the pre-placed cells are located on the core, they cannot be moved further so these needs to be place carefully. The pre-placed cells are surrounded by De-coupling capacitors. A fully charged *de-coupling capacitor* is placed parallel to circuits to ensure proper supply of peak current I_{peak} by decoupling them from main supply voltage. Hence *de-coupling capacitor* ensures *proper local communication* while multiple V_{dd} & V_{ss} lines lead to *proper global communication* (which is taken care in power planning) avoiding voltage droop and ground bounce conditions. Step of *pin*

placement & logical cell placement blockage is explained. *Logical cell placement* is done to avoid PnR tool to place anything cells and to reserve the area for inputs and output pins.

Cell Design:

Cell design is don't in 3 parts.

- Inputs
Inputs for cell design flow are PDKs (Process design kits), DRC & LVS rules, SPICE models, library & user-defined specs.
- Design Steps
Design steps of cell design involve Circuit Design, Layout Design, Characterization. The software GUNA used for characterization. The characterization can be classified as Timing characterization, Power characterization and Noise characterization.
- Outputs
Outputs of the Design are CDL (Circuit Description Language), GDSII, LEF, extracted Spice netlist (.cir), timing, noise, power.libs, function.

Typical Characterization Flow:

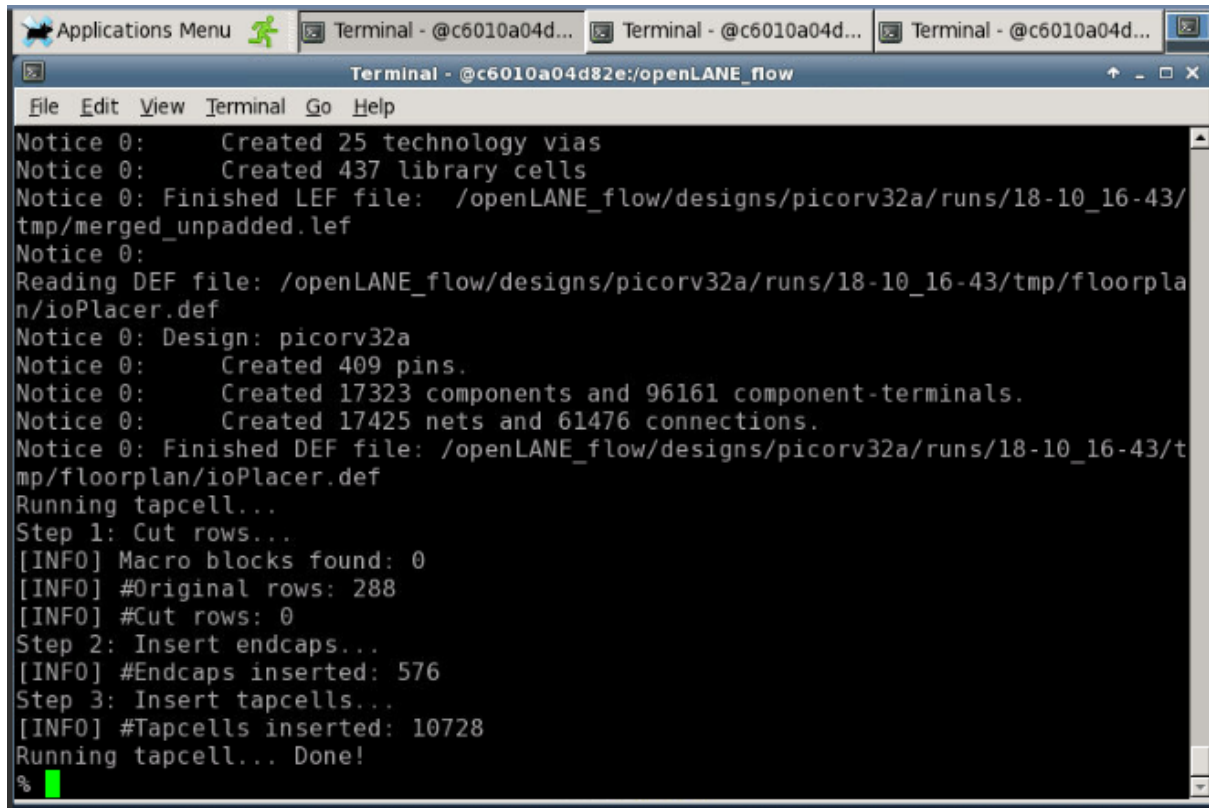
- Reading the Model file of CMOS containing basic property definitions.
- Reading the extracted Spice Netlist.
- Recognize / Define the behaviour of the cell -- buffer cell.
- Read the subcircuits -- of the inverter.
- Attach the necessary power sources.
- Apply an input or the stimulus.
- Provide the necessary output capacitance.
- Provide the necessary simulation commands.

Important parameters of Timing Characterization:

- Rise Delay: Time taken for waveform to rise from 20% to 80% of VDD.
- Fall Delay: Time taken for waveform to fall from 80% to 20% of VDD.
- Propagation Delay: Measured between 50% of Input transition to 50% of Output transition.

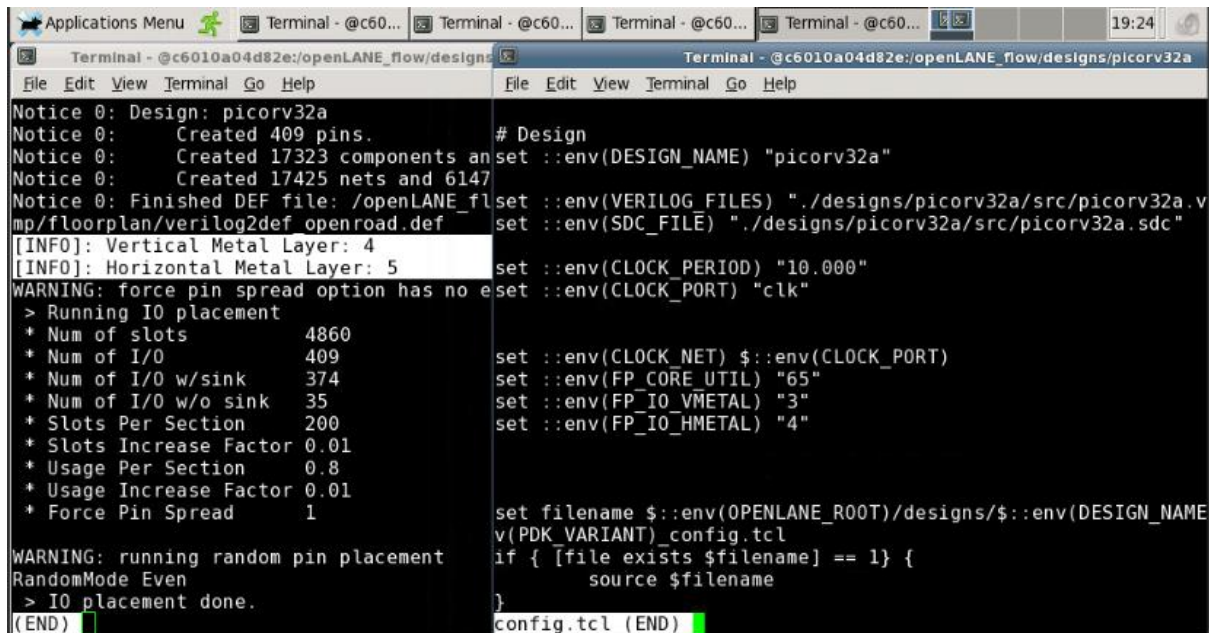
LAB:

Run Floorplan is completed successfully



```
Applications Menu Terminal - @c6010a04d... Terminal - @c6010a04d... Terminal - @c6010a04d...
Terminal - @c6010a04d82e:/openLANE_flow
File Edit View Terminal Go Help
Notice 0: Created 25 technology vias
Notice 0: Created 437 library cells
Notice 0: Finished LEF file: /openLANE_flow/designs/picorv32a/runs/18-10_16-43/tmp/merged_unpadded.lef
Notice 0:
Reading DEF file: /openLANE_flow/designs/picorv32a/runs/18-10_16-43/tmp/floorplan/ioPlacer.def
Notice 0: Design: picorv32a
Notice 0: Created 409 pins.
Notice 0: Created 17323 components and 96161 component-terminals.
Notice 0: Created 17425 nets and 61476 connections.
Notice 0: Finished DEF file: /openLANE_flow/designs/picorv32a/runs/18-10_16-43/tmp/floorplan/ioPlacer.def
Running tapcell...
Step 1: Cut rows...
[INFO] Macro blocks found: 0
[INFO] #Original rows: 288
[INFO] #Cut rows: 0
Step 2: Insert endcaps...
[INFO] #Endcaps inserted: 576
Step 3: Insert tapcells...
[INFO] #Tapcells inserted: 10728
Running tapcell... Done!
%
```

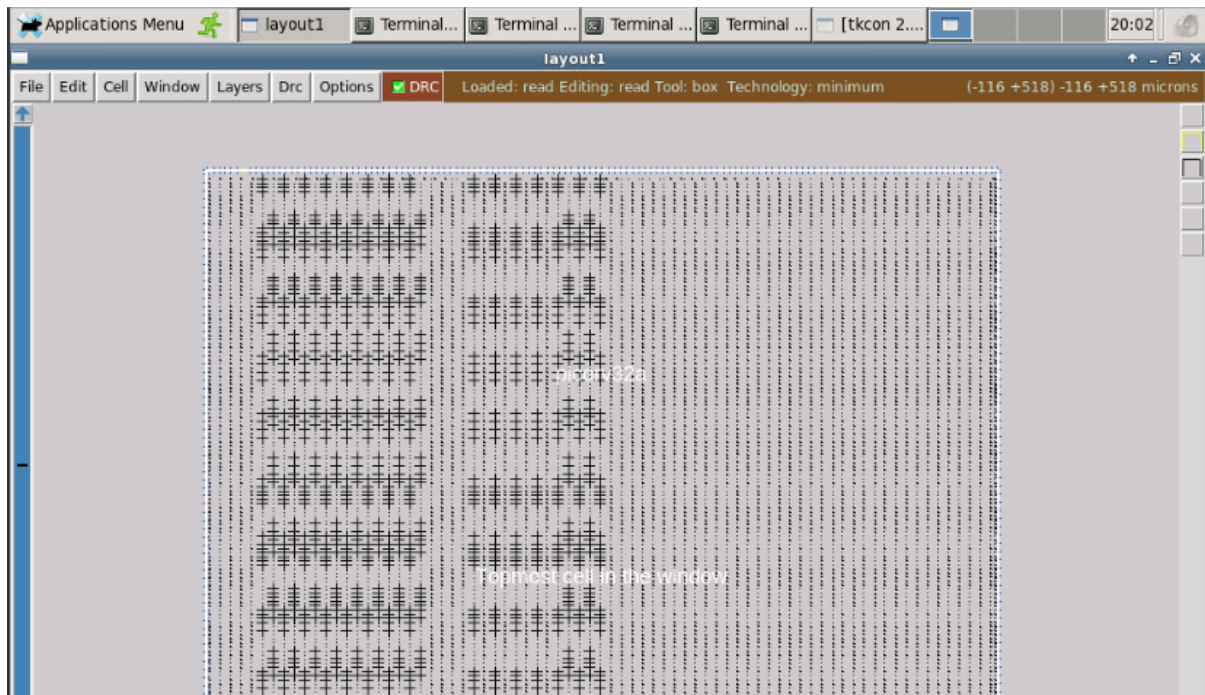
Setting up VMETAL and HMETAL



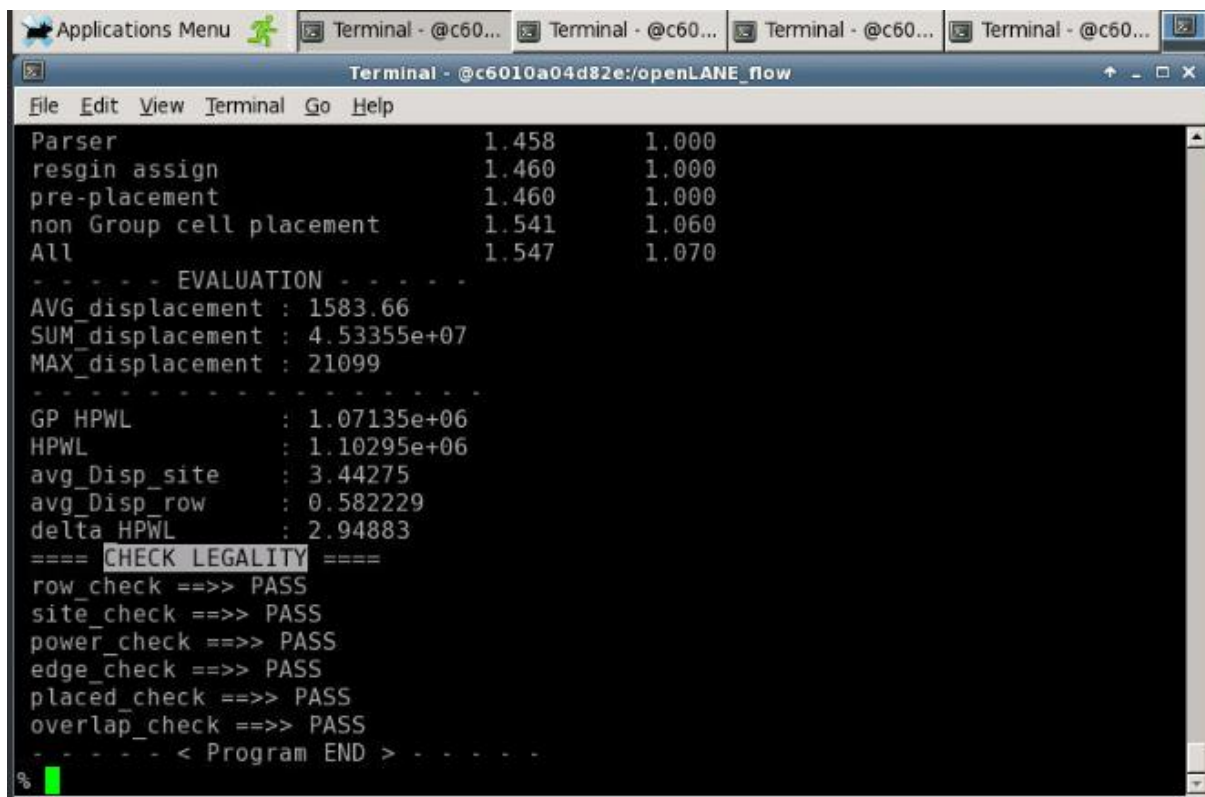
```
Applications Menu Terminal - @c60... Terminal - @c60... Terminal - @c60... Terminal - @c60... 19:24
Terminal - @c6010a04d82e:/openLANE_flow/designs Terminal - @c6010a04d82e:/openLANE_flow/designs/picorv32a
File Edit View Terminal Go Help File Edit View Terminal Go Help
Notice 0: Design: picorv32a
Notice 0: Created 409 pins.
Notice 0: Created 17323 components and 96161 component-terminals.
Notice 0: Created 17425 nets and 61476 connections.
Notice 0: Finished DEF file: /openLANE_flow/designs/picorv32a/runs/18-10_16-43/tmp/floorplan/verilog2def_openroad.def
[INFO]: Vertical Metal Layer: 4
[INFO]: Horizontal Metal Layer: 5
WARNING: force pin spread option has no effect
> Running IO placement
* Num of slots 4860
* Num of I/O 409
* Num of I/O w/sink 374
* Num of I/O w/o sink 35
* Slots Per Section 200
* Slots Increase Factor 0.01
* Usage Per Section 0.8
* Usage Increase Factor 0.01
* Force Pin Spread 1
WARNING: running random pin placement
RandomMode Even
> IO placement done.
(END)

# Design
set ::env(DSIGN_NAME) "picorv32a"
set ::env(VERILOG_FILES) "/designs/picorv32a/src/picorv32a.v"
set ::env(SDC_FILE) "/designs/picorv32a/src/picorv32a.sdc"
set ::env(CLOCK_PERIOD) "10.000"
set ::env(CLOCK_PORT) "clk"
set ::env(CLOCK_NET) $::env(CLOCK_PORT)
set ::env(FP_CORE_UTIL) "65"
set ::env(FP_IO_VMETAL) "3"
set ::env(FP_IO_HMETAL) "4"
set filename $::env(OPENLANE_ROOT)/designs/$::env(DSIGN_NAME)
v(PDK_VARIANT)_config.tcl
if { [file exists $filename] == 1 } {
    source $filename
}
config.tcl (END)
```

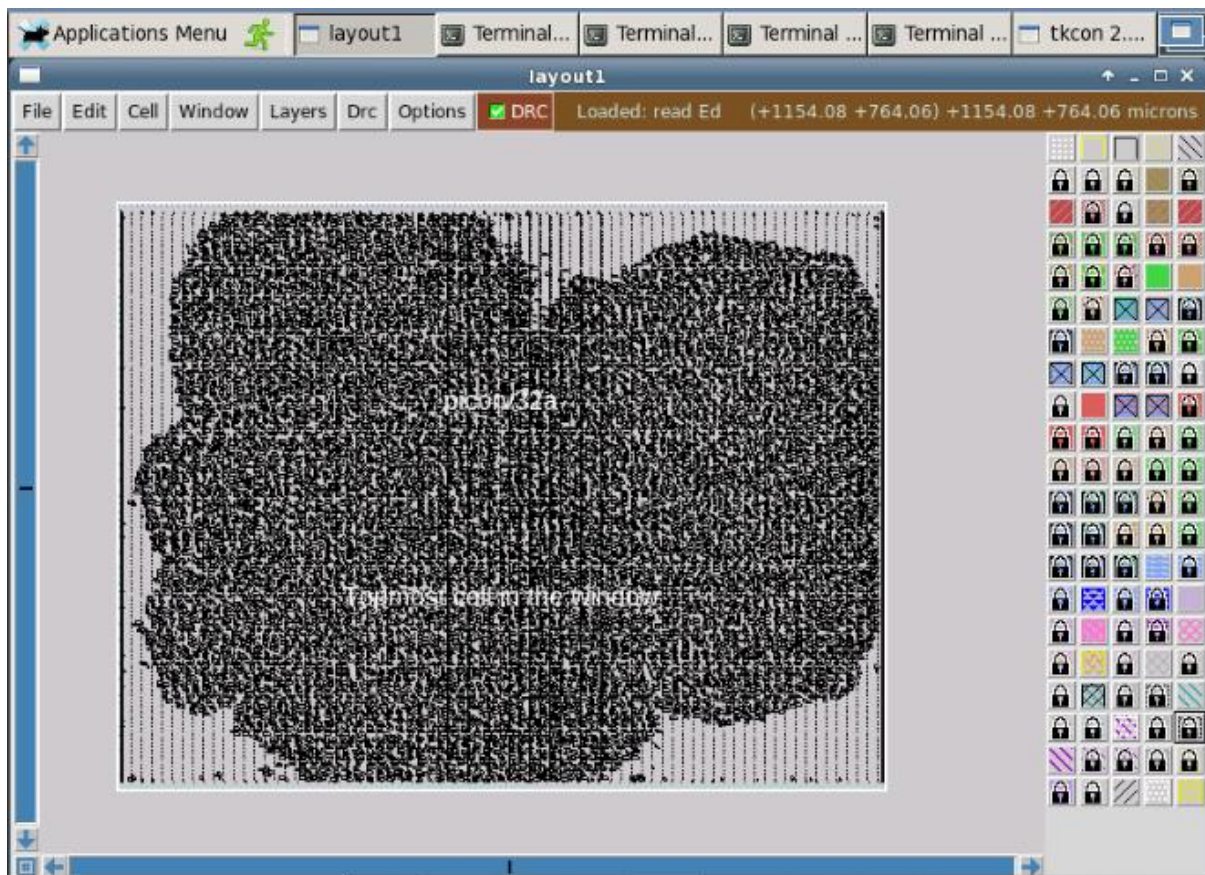

Review floor plan layout in Magic



Run placement is completed successfully



Review placement layout in Magic



4. Day 3: Design library cell using Magic layout and ngSpice characterization

On Day3, SPICE deck creation for an inverter was explained and later, 16-Mask CMOS process was explained step-by-step in detail. 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's like components connectivity, component values, Identify 'nodes' and name '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 parameters that are *Switching Threshold* (V_m). Switching threshold defined by condition $V_{in} = V_{out}$.

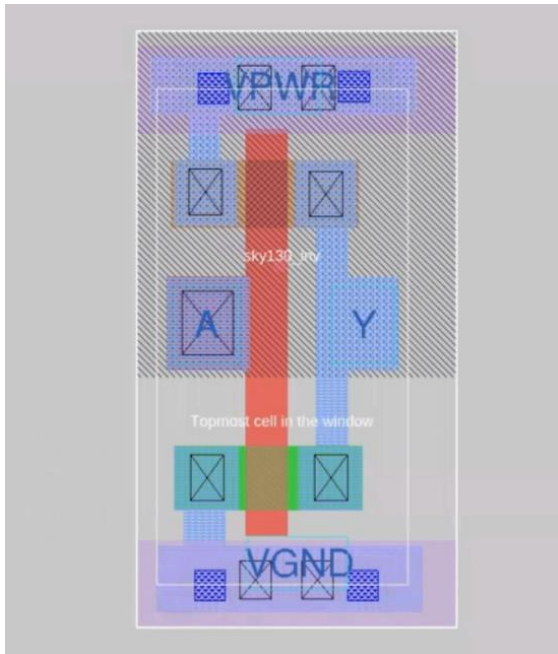
16-Mask CMOS Process steps:

1. Selecting a substrate.
2. Create Active region for transistors.
3. Formation of N-Well and P-Well.
4. Formation of Gate.
5. Lightly Doped Drain (LDD) formation.
6. Source and drain Formation.

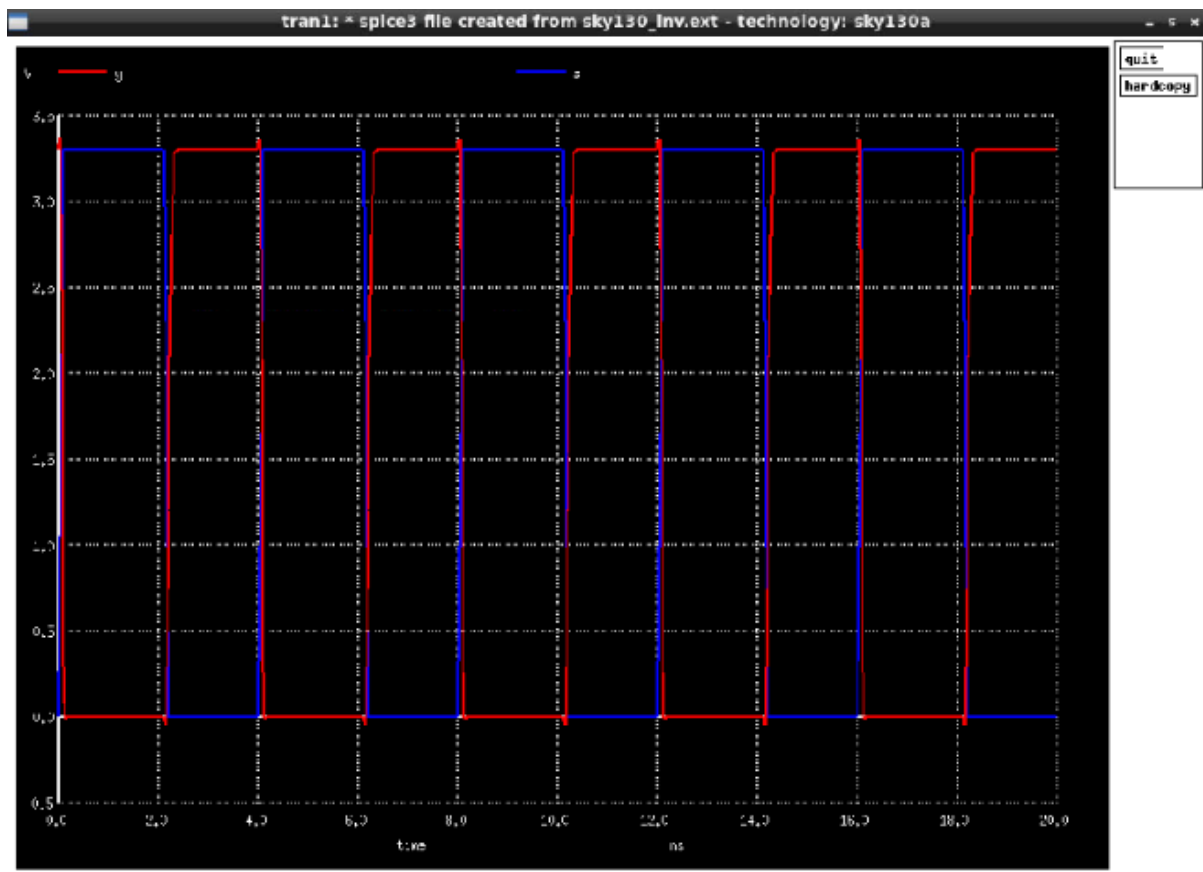
7. Steps to form Contacts and interconnects (local).
8. Higher level formation.

LAB:

Inverter Layout in Magic



Transient analysis of an inverter in ngspice



5. Day 4: Pre-Layout timing analysis and importance of good clock tree

On Day4, Delay tables, setup & hold time concepts were explained. 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.

Setup & Hold Slack Analysis:

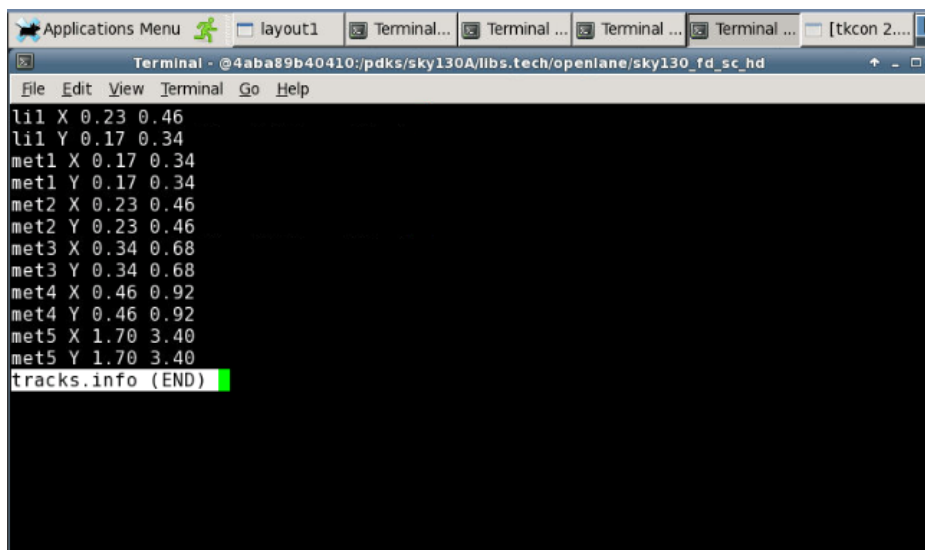
Setup and hold time define a window of time in which our data should remain unchanged for desired data transfer to take place. Factors like uncertainty and skew also play an important role in this. Clock skew is the difference between Source Clock path and Destination Clock path. Slack defined as difference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

OpenROAD Commands:

```
read_lef /openLANE_flow/designs/picorv32a/runs/trial/tmp/merged.lef
read_def /openLANE_flow/designs/picorv32a/runs/trial/results/cts/picorv32a.cts.def
write_db pico1.db
read_verilog
/openLANE_flow/designs/picorv32a/runs/trial/results/synthesis/picorv32a.synthesis_cts.v
read_liberty $::env(LIB_SYNTH_COMPLETE)
lik_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
```

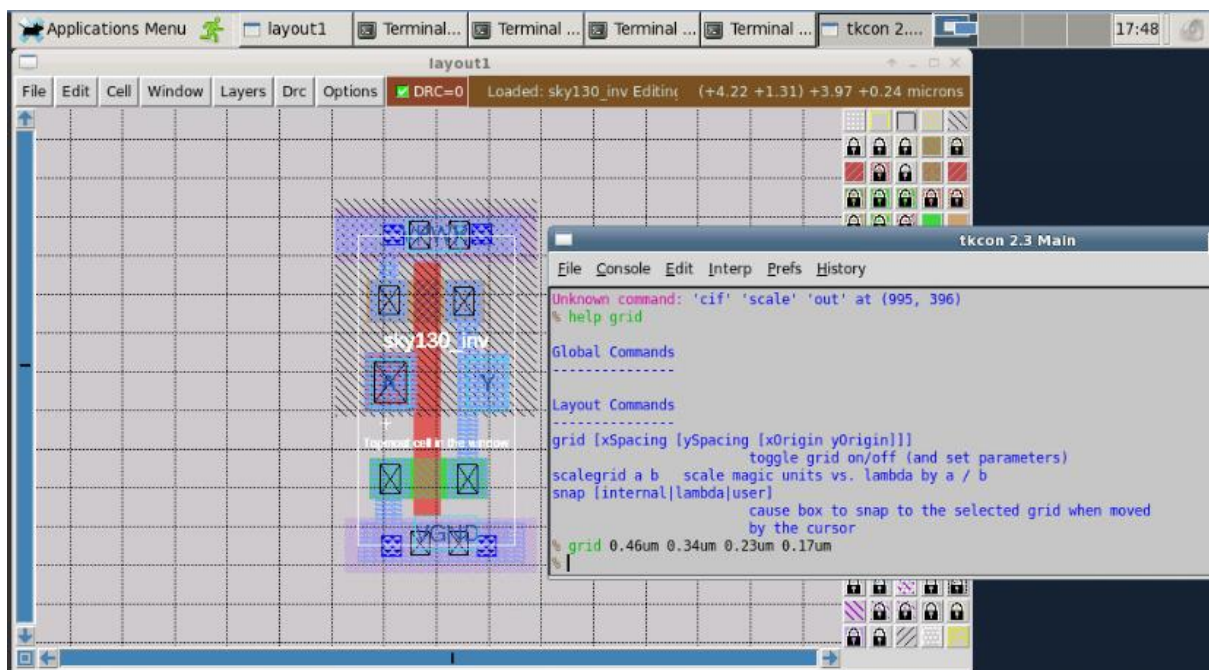
LAB:

Lab1: Layer information in sky130_fd_sc_hd file

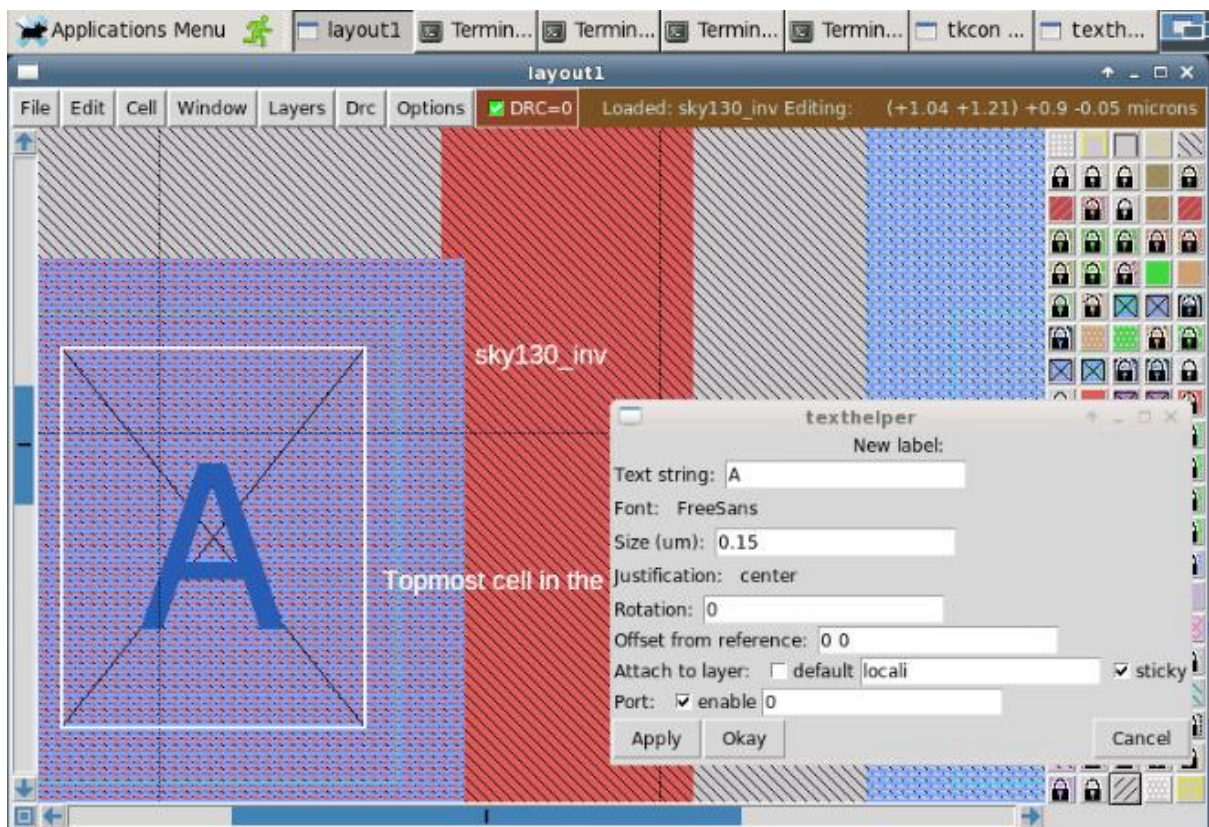


```
lil X 0.23 0.46
lil Y 0.17 0.34
met1 X 0.17 0.34
met1 Y 0.17 0.34
met2 X 0.23 0.46
met2 Y 0.23 0.46
met3 X 0.34 0.68
met3 Y 0.34 0.68
met4 X 0.46 0.92
met4 Y 0.46 0.92
met5 X 1.70 3.40
met5 Y 1.70 3.40
tracks.info (END)
```

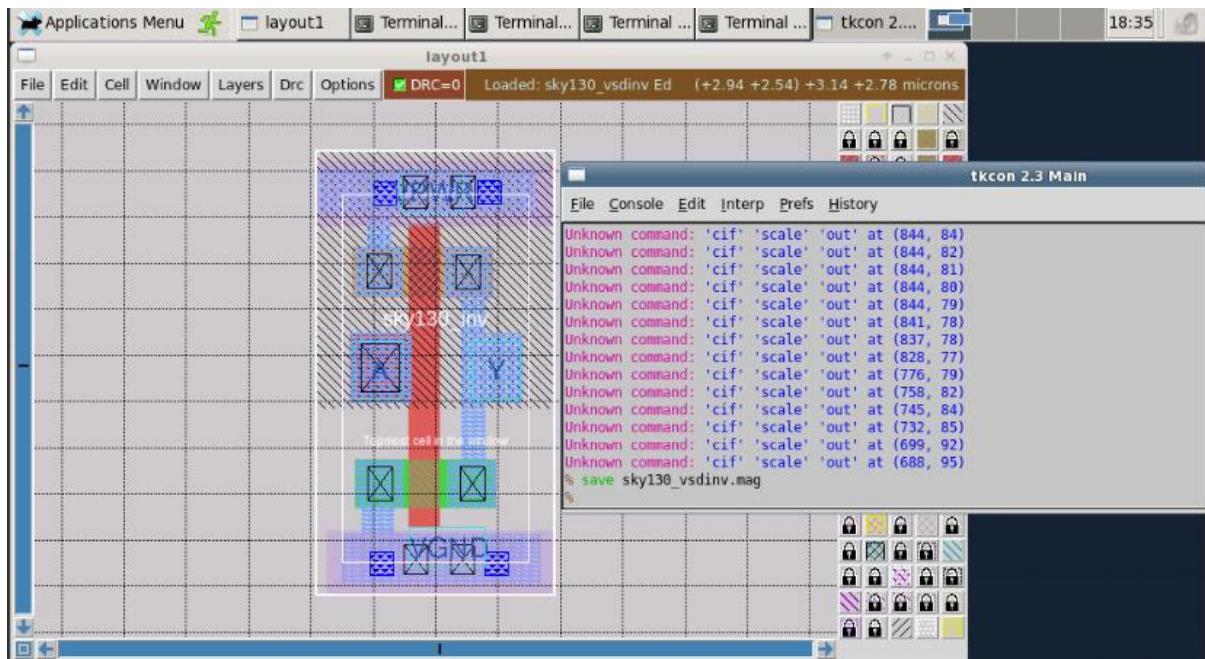

Gird size modified to track size



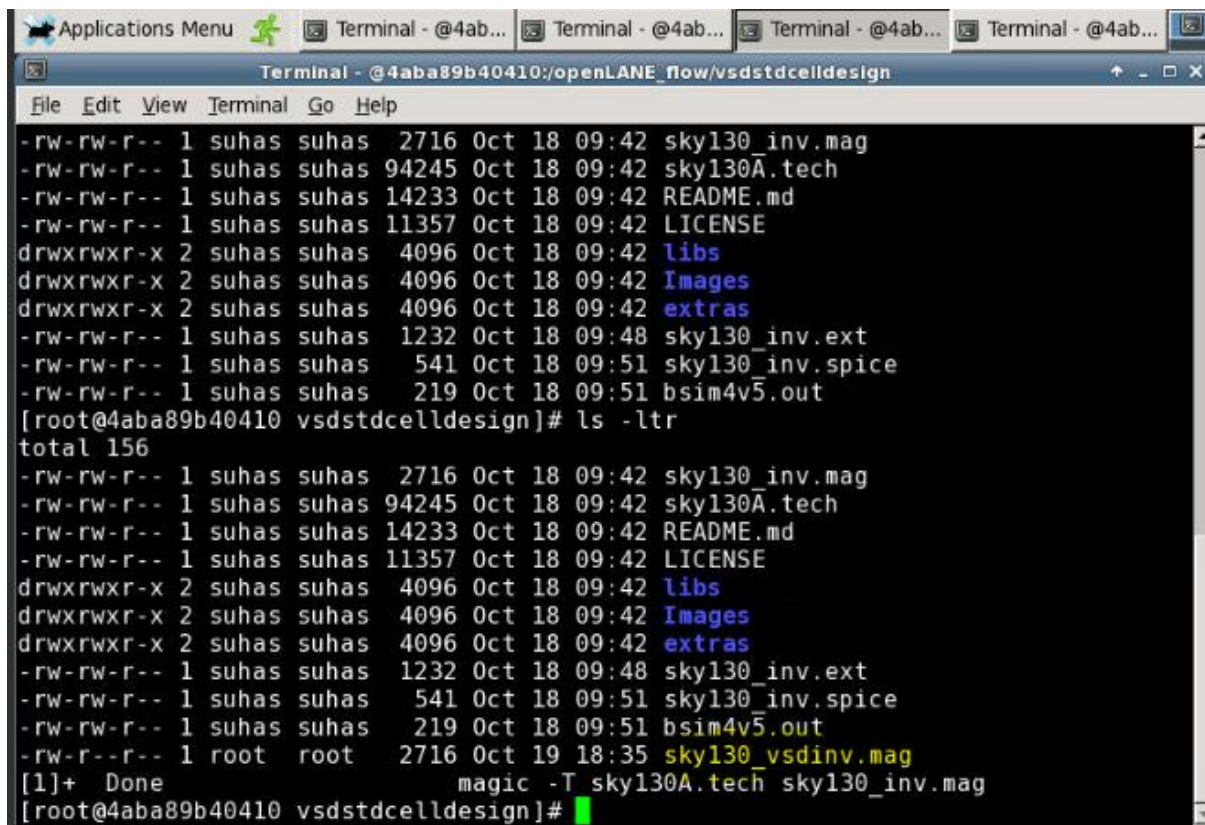
Process to modify labels



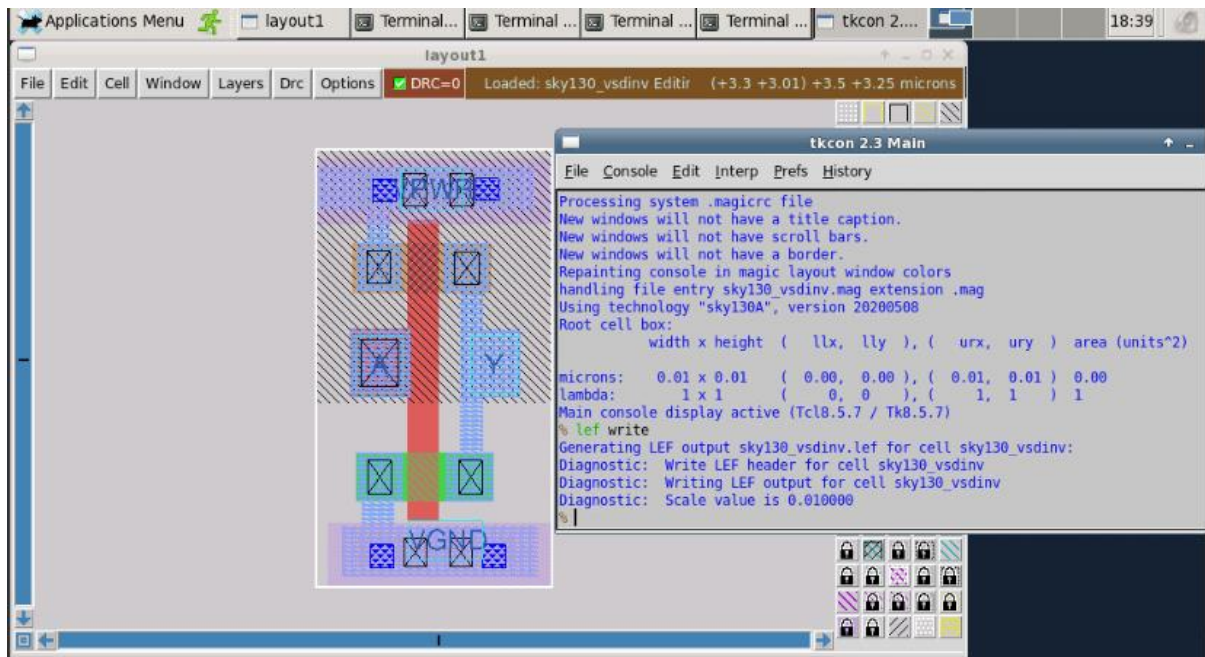
Command to convert Magic layout to .mag



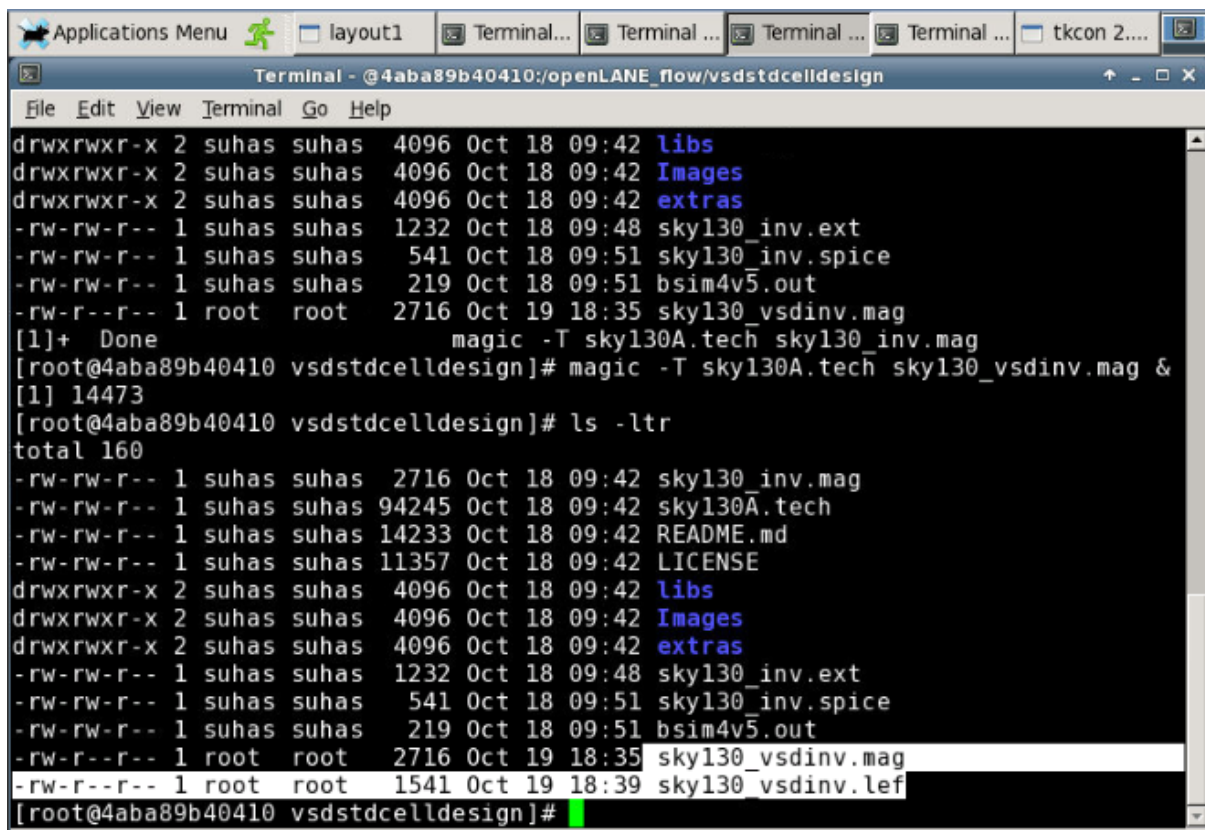
Converting Magic Layout to .mag



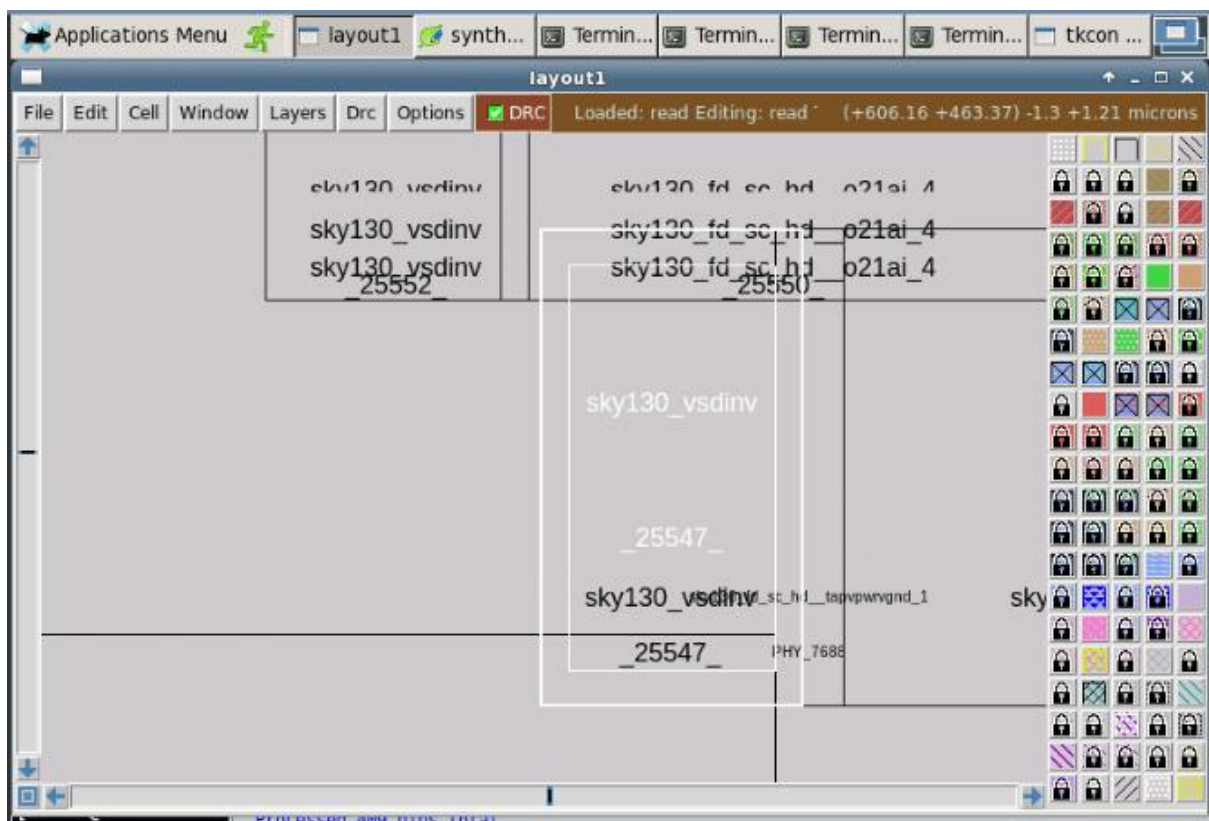
Creating LEF file from Magic layout



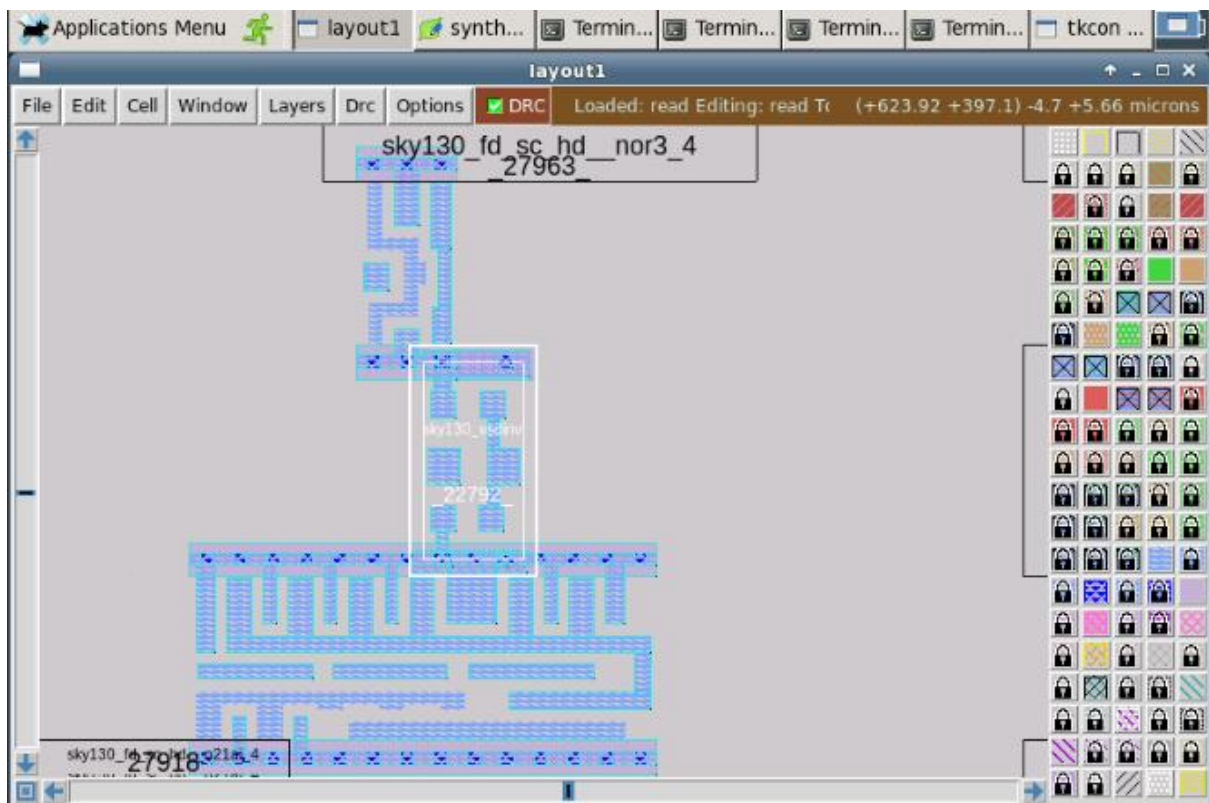
LEF file created



Layout of the sky130_vdsinv Cell



Layout in expanded view



Lab2: Reduction in the Slack Violation and increase in area

```
synthesis data
File Edit Search Options Help
before:
Chip area for module '\picorv32a': 225955.459200
-17.96 slack (VIOLATED)
tns -2593.43
wns -17.96

after:
Chip area for module '\picorv32a': 317725.974400
-5.36 slack (VIOLATED)
tns -356.06
wns -5.36

after fanout as 4:
Chip area for module '\picorv32a': 325893.808000
-3.76 slack (VIOLATED)
tns -96.76
wns -3.76

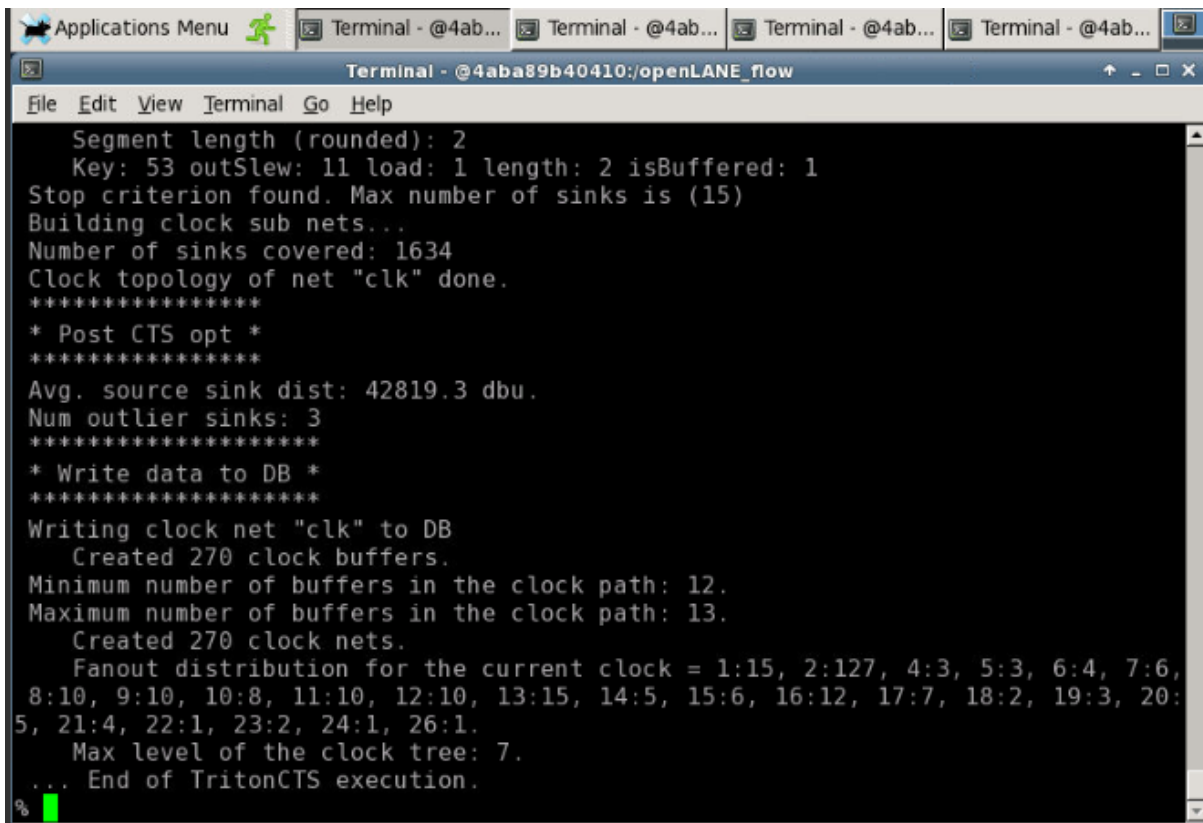
after 2 buffer upsizes:
-0.91 slew
tns -7.71
wns -1.16
```

```
Terminal - @4aba89b40410:/openLANE_flow
File Edit View Terminal Go Help
0.00 12.62 v _50075_/D (skyl30_fd_sc_hd__dfxtp_4)
12.62 data arrival time

12.00 12.00 clock clk (rise edge)
0.00 12.00 clock network delay (ideal)
0.00 12.00 clock reconvergence pessimism
12.00 ^ _50075_/CLK (skyl30_fd_sc_hd__dfxtp_4)
-0.28 11.72 library setup time
11.72 data required time
-----
11.72 data required time
-12.62 data arrival time
-----
-0.91 slack (VIOLATED)

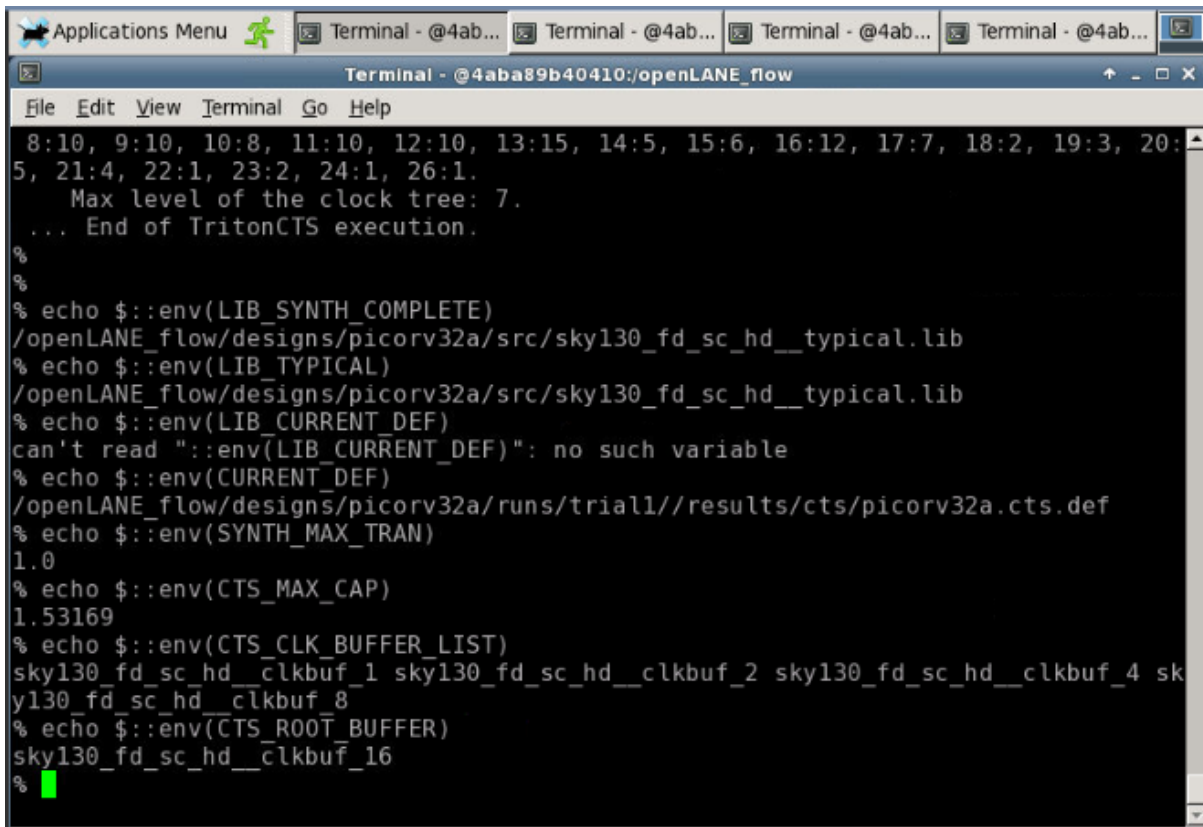
% report_tns
tns -7.71
% report_wns
wns -1.16
%
```

Lab3: Run CTS is completed successfully



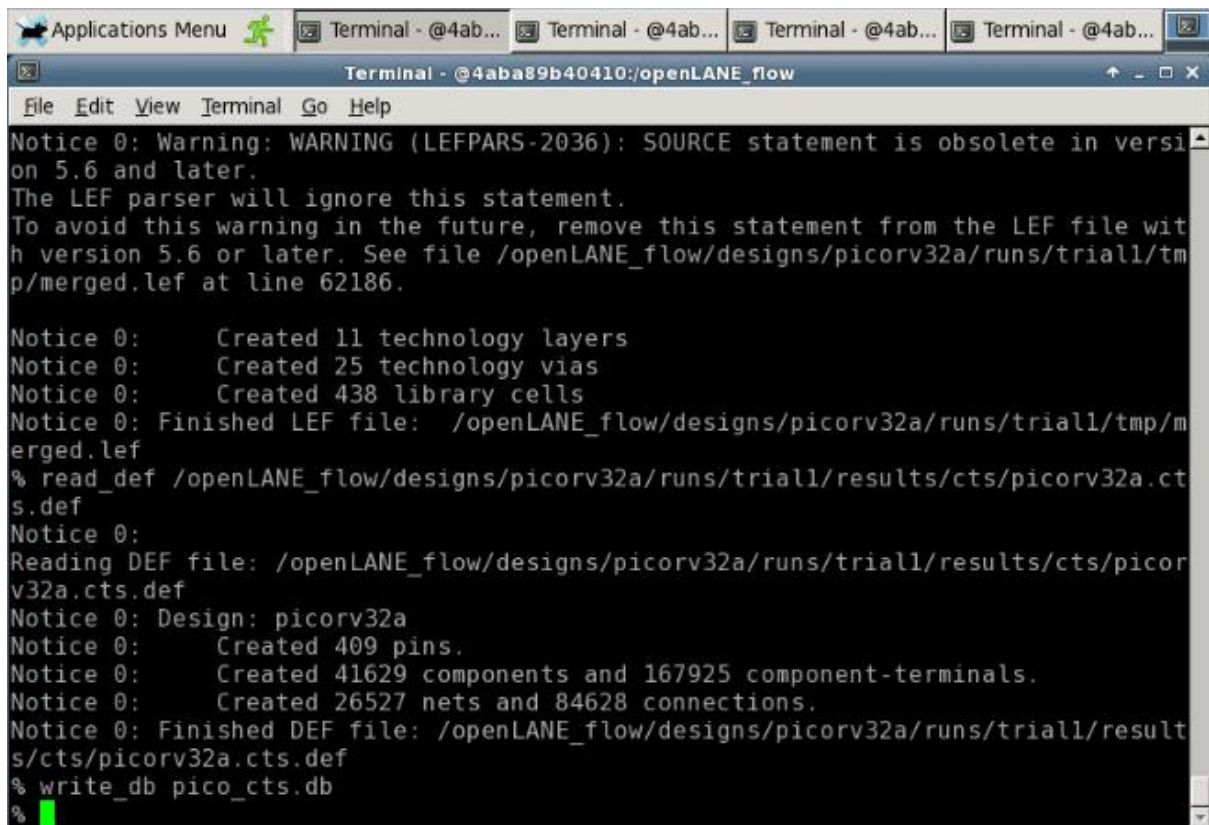
```
Segment length (rounded): 2
Key: 53 outSlew: 11 load: 1 length: 2 isBuffered: 1
Stop criterion found. Max number of sinks is (15)
Building clock sub nets...
Number of sinks covered: 1634
Clock topology of net "clk" done.
*****
* Post CTS opt *
*****
Avg. source sink dist: 42819.3 dbu.
Num outlier sinks: 3
*****
* Write data to DB *
*****
Writing clock net "clk" to DB
Created 270 clock buffers.
Minimum number of buffers in the clock path: 12.
Maximum number of buffers in the clock path: 13.
Created 270 clock nets.
Fanout distribution for the current clock = 1:15, 2:127, 4:3, 5:3, 6:4, 7:6,
8:10, 9:10, 10:8, 11:10, 12:10, 13:15, 14:5, 15:6, 16:12, 17:7, 18:2, 19:3, 20:
5, 21:4, 22:1, 23:2, 24:1, 26:1.
Max level of the clock tree: 7.
... End of TritonCTS execution.
%
```

Lab4: CTS values



```
8:10, 9:10, 10:8, 11:10, 12:10, 13:15, 14:5, 15:6, 16:12, 17:7, 18:2, 19:3, 20:
5, 21:4, 22:1, 23:2, 24:1, 26:1.
Max level of the clock tree: 7.
... End of TritonCTS execution.
%
%
% echo $::env(LIB_SYNTH_COMPLETE)
/openLANE_flow/designs/picorv32a/src/skyl30_fd_sc_hd__typical.lib
% echo $::env(LIB_TYPICAL)
/openLANE_flow/designs/picorv32a/src/skyl30_fd_sc_hd__typical.lib
% echo $::env(LIB_CURRENT_DEF)
can't read "$::env(LIB_CURRENT_DEF)": no such variable
% echo $::env(CURRENT_DEF)
/openLANE_flow/designs/picorv32a/runs/trial1//results/cts/picorv32a.cts.def
% echo $::env(SYNTH_MAX_TRAN)
1.0
% echo $::env(CTS_MAX_CAP)
1.53169
% echo $::env(CTS_CLK_BUFFER_LIST)
skyl30_fd_sc_hd__clkbuf_1 skyl30_fd_sc_hd__clkbuf_2 skyl30_fd_sc_hd__clkbuf_4 sk
yl30_fd_sc_hd__clkbuf_8
% echo $::env(CTS_ROOT_BUFFER)
skyl30_fd_sc_hd__clkbuf_16
%
```

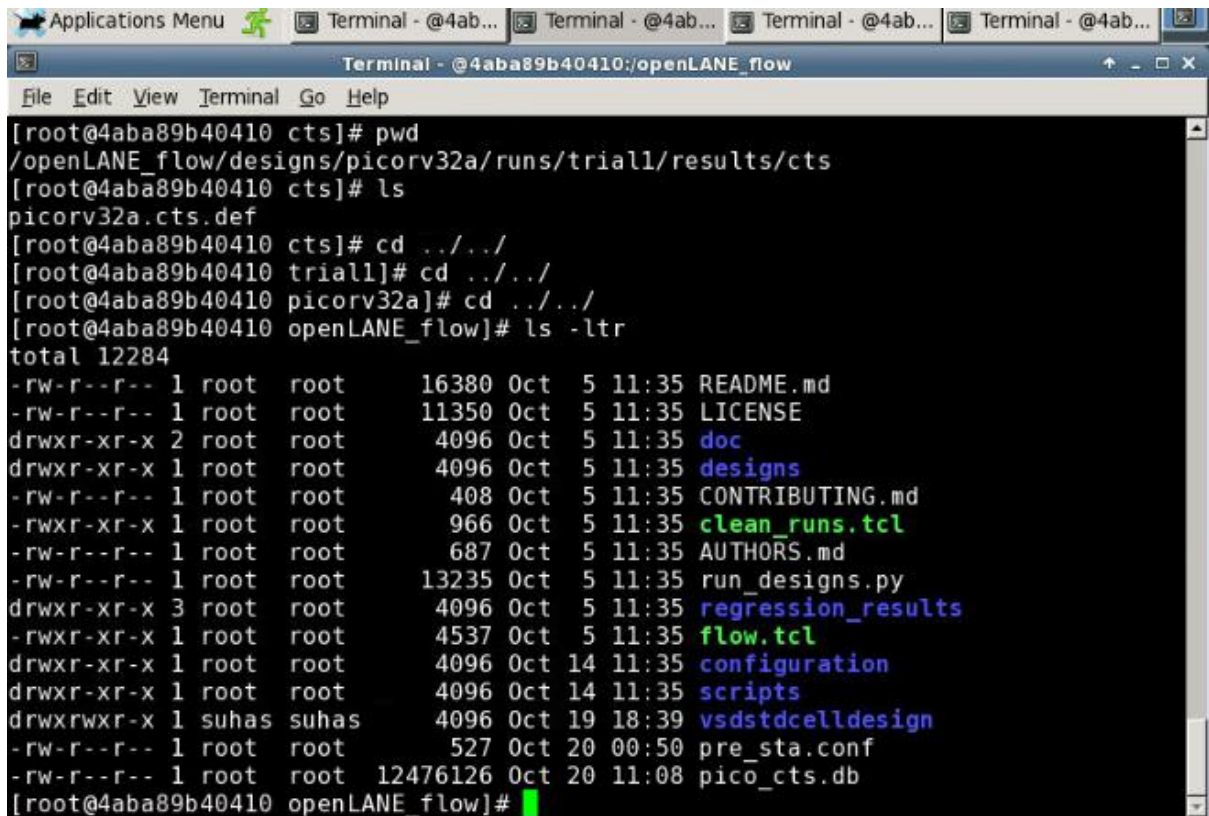

Creating db file



```
Applications Menu Terminal - @4ab... Terminal - @4ab... Terminal - @4ab... Terminal - @4ab...
Terminal - @4aba89b40410:/openLANE_flow
File Edit View Terminal Go Help
Notice 0: Warning: WARNING (LEFPARS-2036): SOURCE statement is obsolete in version 5.6 and later.
The LEF parser will ignore this statement.
To avoid this warning in the future, remove this statement from the LEF file with version 5.6 or later. See file /openLANE_flow/designs/picorv32a/runs/trial1/tmp/merged.lef at line 62186.

Notice 0: Created 11 technology layers
Notice 0: Created 25 technology vias
Notice 0: Created 438 library cells
Notice 0: Finished LEF file: /openLANE_flow/designs/picorv32a/runs/trial1/tmp/merged.lef
% read_def /openLANE_flow/designs/picorv32a/runs/trial1/results/cts/picorv32a.cts.def
Notice 0:
Reading DEF file: /openLANE_flow/designs/picorv32a/runs/trial1/results/cts/picorv32a.cts.def
Notice 0: Design: picorv32a
Notice 0: Created 409 pins.
Notice 0: Created 41629 components and 167925 component-terminals.
Notice 0: Created 26527 nets and 84628 connections.
Notice 0: Finished DEF file: /openLANE_flow/designs/picorv32a/runs/trial1/results/cts/picorv32a.cts.def
% write_db pico_cts.db
%
```

Db file created in OpenLANE_flow directory



```
Applications Menu Terminal - @4ab... Terminal - @4ab... Terminal - @4ab... Terminal - @4ab...
Terminal - @4aba89b40410:/openLANE_flow
File Edit View Terminal Go Help
[root@4aba89b40410 cts]# pwd
/openLANE_flow/designs/picorv32a/runs/trial1/results/cts
[root@4aba89b40410 cts]# ls
picorv32a.cts.def
[root@4aba89b40410 cts]# cd ../../
[root@4aba89b40410 trial1]# cd ../../
[root@4aba89b40410 picorv32a]# cd ../../
[root@4aba89b40410 openLANE_flow]# ls -ltr
total 12284
-rw-r--r-- 1 root root 16380 Oct 5 11:35 README.md
-rw-r--r-- 1 root root 11350 Oct 5 11:35 LICENSE
drwxr-xr-x 2 root root 4096 Oct 5 11:35 doc
drwxr-xr-x 1 root root 4096 Oct 5 11:35 designs
-rw-r--r-- 1 root root 408 Oct 5 11:35 CONTRIBUTING.md
-rwxr-xr-x 1 root root 966 Oct 5 11:35 clean_runs.tcl
-rw-r--r-- 1 root root 687 Oct 5 11:35 AUTHORS.md
-rw-r--r-- 1 root root 13235 Oct 5 11:35 run_designs.py
drwxr-xr-x 3 root root 4096 Oct 5 11:35 regression_results
-rwxr-xr-x 1 root root 4537 Oct 5 11:35 flow.tcl
drwxr-xr-x 1 root root 4096 Oct 14 11:35 configuration
drwxr-xr-x 1 root root 4096 Oct 14 11:35 scripts
drwxrwxr-x 1 suhas suhas 4096 Oct 19 18:39 vsdstdcclldesign
-rw-r--r-- 1 root root 527 Oct 20 00:50 pre_sta.conf
-rw-r--r-- 1 root root 12476126 Oct 20 11:08 pico_cts.db
[root@4aba89b40410 openLANE_flow]#
```

Reading SDC file

```
% read_liberty -min $::env(LIB_MIN)
1
% read_sdc /openLANE_flow/designs/picorv32a/src/my_base.sdc
[INFO]: Setting output delay to: 2.4000000000000004
[INFO]: Setting input delay to: 2.4000000000000004
[INFO]: Setting load to: 0.01765
0
%
```

Slack value MET

```
Applications Menu Terminal - @4ab... Terminal - @4ab... Terminal - @4ab... Terminal - @4ab...
Terminal - @4aba89b40410:/openLANE_flow
File Edit View Terminal Go Help
0.3445 3.7926 ^ _29067_/X (skyl30_fd_sc_hd_buf_1)
0.3445 4.1372 ^ _29128_/X (skyl30_fd_sc_hd_buf_1)
0.1776 4.3147 ^ _31098_/X (skyl30_fd_sc_hd_buf_1)
0.1560 4.4707 ^ _31099_/X (skyl30_fd_sc_hd_buf_1)
0.2349 4.7056 ^ _31100_/X (skyl30_fd_sc_hd_buf_1)
0.1725 4.8781 ^ _31101_/X (skyl30_fd_sc_hd_or2_4)
0.0524 4.9306 v _31102_/Y (skyl30_fd_sc_hd_o21ai_4)
0.1955 5.1261 ^ _31109_/Y (skyl30_fd_sc_hd_a21oi_4)
0.0000 5.1261 ^ mem_la_addr[2] (out)
5.1261 data arrival time

12.0000 12.0000 clock clk (rise edge)
0.0000 12.0000 clock network delay (propagated)
0.0000 12.0000 clock reconvergence pessimism
-2.4000 9.6000 output external delay
9.6000 data required time
-----
9.6000 data required time
-5.1261 data arrival time
-----
4.4739 slack (MET)

%
```

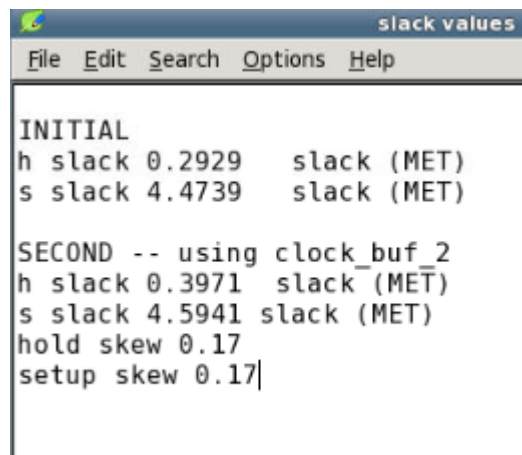
Removed clkbuf_1

```
% exit
% echo $::env(CTS_CLK_BUFFER_LIST)
skyl30_fd_sc_hd_clkbuf_1 skyl30_fd_sc_hd_clkbuf_2 skyl30_fd_sc_hd_clkbuf_4 skyl30_fd_sc_hd_clkbuf_8
%
% lreplace $::env(CTS_CLK_BUFFER_LIST) 0 0
skyl30_fd_sc_hd_clkbuf_2 skyl30_fd_sc_hd_clkbuf_4 skyl30_fd_sc_hd_clkbuf_8
% echo $::env(CTS_CLK_BUFFER_LIST)
skyl30_fd_sc_hd_clkbuf_1 skyl30_fd_sc_hd_clkbuf_2 skyl30_fd_sc_hd_clkbuf_4 skyl30_fd_sc_hd_clkbuf_8
% set $::env(CTS_CLK_BUFFER_LIST) [lreplace $::env(CTS_CLK_BUFFER_LIST) 0 0]
skyl30_fd_sc_hd_clkbuf_2 skyl30_fd_sc_hd_clkbuf_4 skyl30_fd_sc_hd_clkbuf_8
%
```

Inserted clkbuf_1 to the clock buffers

```
% exit
% echo $::env(CTS_CLK_BUFFER_LIST)
skyl30_fd_sc_hd__clkbuf_2 skyl30_fd_sc_hd__clkbuf_4 skyl30_fd_sc_hd__clkbuf_8
% set ::env(CTS_CLK_BUFFER_LIST) [linsert $::env(CTS_CLK_BUFFER_LIST) 0 skyl30_
fd_sc_hd__clkbuf_1]
skyl30_fd_sc_hd__clkbuf_1 skyl30_fd_sc_hd__clkbuf_2 skyl30_fd_sc_hd__clkbuf_4 sk
yl30_fd_sc_hd__clkbuf_8
% echo $::env(CTS_CLK_BUFFER_LIST)
skyl30_fd_sc_hd__clkbuf_1 skyl30_fd_sc_hd__clkbuf_2 skyl30_fd_sc_hd__clkbuf_4 sk
yl30_fd_sc_hd__clkbuf_8
%
```

Change in slack due to clkbuf_2



```
slack values
File Edit Search Options Help

INITIAL
h slack 0.2929    slack (MET)
s slack 4.4739    slack (MET)

SECOND -- using clock_buf_2
h slack 0.3971    slack (MET)
s slack 4.5941    slack (MET)
hold skew 0.17
setup skew 0.17|
```

6. Day 5: Final steps for RTL2GDS using TritonRoute and OpenSTA

On Day5, explanation of how routing process done using TritonRoute and explanation of SPEF file extraction in done. 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 1 spaced pattern is finalised for route.

Typical DRC rules for pair of wires:

- Wire width
- Wire pitch
- 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 2 steps:

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

TritonRoute 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
- Splitting
- Merging
- Bridging
- Pre-processed Route

LAB:

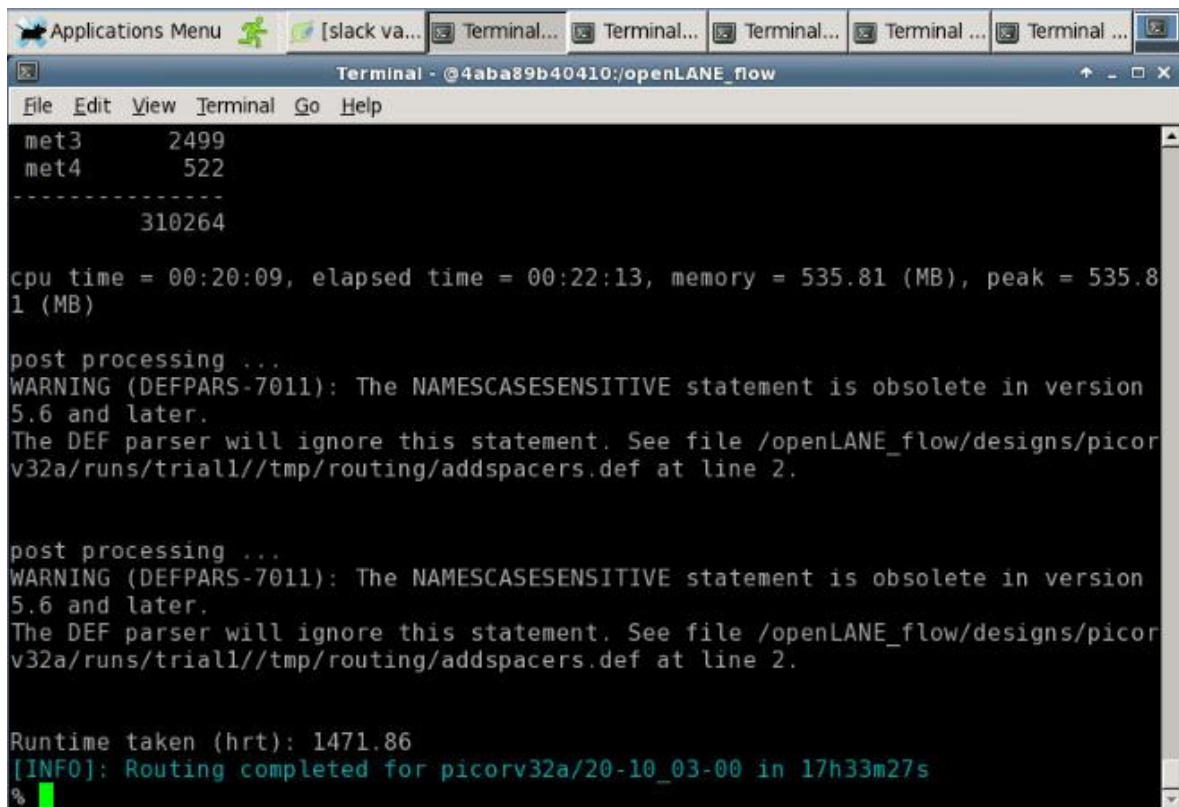
Generating PDN

```
% echo $::env(CURRENT_DEF)
/openLANE_flow/designs/picorv32a/runs/trial1//results/cts/picorv32a.cts.def
% gen_pdn
```

PDN generation completed successfully

```
Applications Menu [slack va... Terminal... Terminal ... Terminal ... Terminal ... Terminal ...
Terminal - @4aba89b40410:/openLANE_flow
File Edit View Terminal Go Help
s/cts/picorv32a.cts.def
[INFO] [PDNG-0016] Power Delivery Network Generator: Generating PDN
[INFO] [PDNG-0016] config: /pdks/skyl30A/libs.tech/openlane/common_pdn.tcl
[INFO] [PDNG-0008] Design Name is picorv32a
[INFO] [PDNG-0009] Reading technology data
[INFO] [PDNG-0011] ***** INFO *****
Type: stdcell, grid
  Stdcell Rails
    Layer: met1 - width: 0.480 pitch: 2.720 offset: 0.000
  Straps
    Layer: met4 - width: 1.600 pitch: 153.600 offset: 16.320
    Layer: met5 - width: 1.600 pitch: 153.180 offset: 16.650
    Connect: {met1 met4} {met4 met5}
Type: macro, macro_1
  Macro orientation: R0 R180 MX MY R90 R270 MXR90 MYR90
  Straps
    Connect:
[INFO] [PDNG-0012] **** END INFO ****
[INFO] [PDNG-0013] Inserting stdcell grid - grid
[INFO] [PDNG-0015] Writing to database
% echo $::env(CURRENT_DEF)
/openLANE_flow/designs/picorv32a/runs/trial1//tmp/floorplan/pdn.def
% echo $::env(ROUTING_STRATEGY)
0
% run_routing
```


Run routing is completed successfully

A terminal window titled "Terminal - @4aba89b40410:/openLANE_flow" showing the output of a routing process. The output includes a table of metrics for 'met3' and 'met4', a summary of CPU time, elapsed time, and memory usage, and two identical warning messages about the 'NAMECASESENSITIVE' statement being obsolete. The process ends with a confirmation message that routing is completed.

```
met3      2499
met4      522
-----
          310264

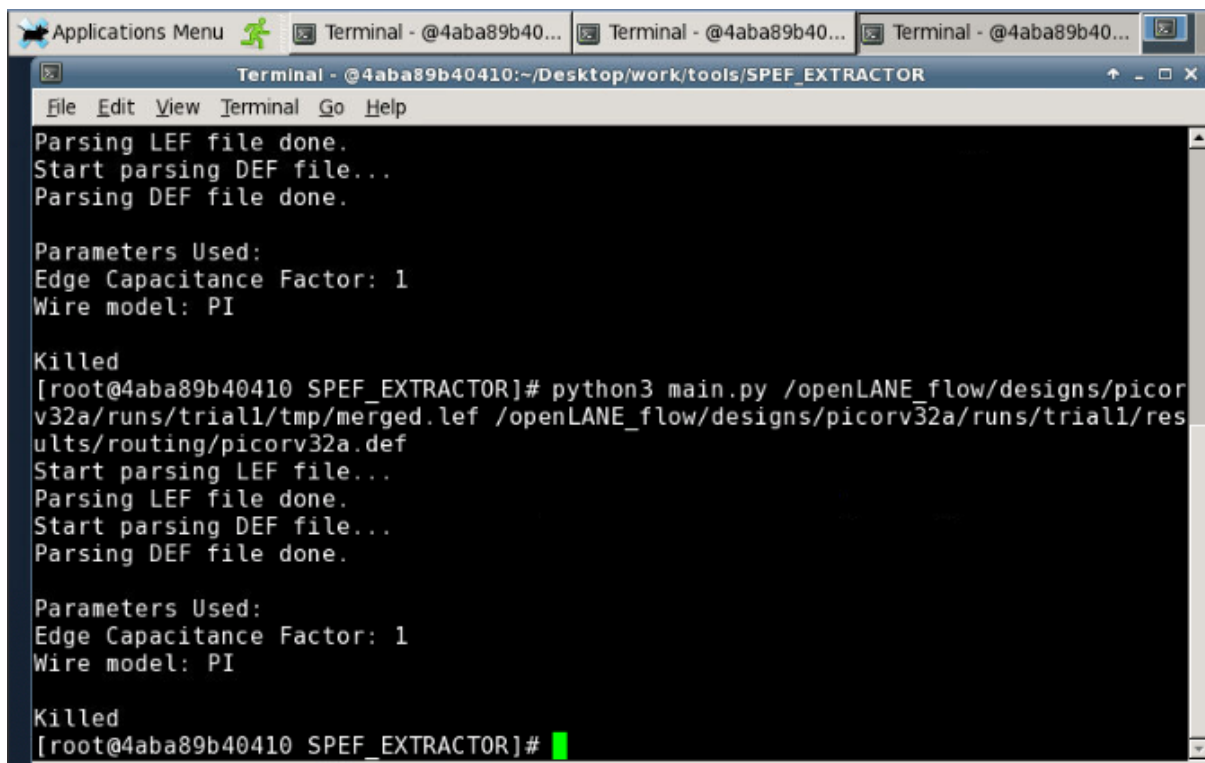
cpu time = 00:20:09, elapsed time = 00:22:13, memory = 535.81 (MB), peak = 535.81 (MB)

post processing ...
WARNING (DEFPARS-7011): The NAMECASESENSITIVE statement is obsolete in version 5.6 and later.
The DEF parser will ignore this statement. See file /openLANE_flow/designs/picorv32a/runs/trial1/tmp/routing/addspacers.def at line 2.

post processing ...
WARNING (DEFPARS-7011): The NAMECASESENSITIVE statement is obsolete in version 5.6 and later.
The DEF parser will ignore this statement. See file /openLANE_flow/designs/picorv32a/runs/trial1/tmp/routing/addspacers.def at line 2.

Runtime taken (hrt): 1471.86
[INF0]: Routing completed for picorv32a/20-10_03-00 in 17h33m27s
%
```

SPEF extraction (mine got killed)

A terminal window titled "Terminal - @4aba89b40410:~/Desktop/work/tools/SPEF_EXTRACTOR" showing the output of a SPEF extraction process. The process successfully parses LEF and DEF files and displays parameters used. However, it is then killed twice, as indicated by the "Killed" messages and the prompt returning to the user.

```
Parsing LEF file done.
Start parsing DEF file...
Parsing DEF file done.

Parameters Used:
Edge Capacitance Factor: 1
Wire model: PI

Killed
[root@4aba89b40410 SPEF_EXTRACTOR]# python3 main.py /openLANE_flow/designs/picorv32a/runs/trial1/tmp/merged.lef /openLANE_flow/designs/picorv32a/runs/trial1/results/routing/picorv32a.def
Start parsing LEF file...
Parsing LEF file done.
Start parsing DEF file...
Parsing DEF file done.

Parameters Used:
Edge Capacitance Factor: 1
Wire model: PI

Killed
[root@4aba89b40410 SPEF_EXTRACTOR]#
```


7. Acknowledgements

- Kunal Ghosh, Co-founder (VSD Corp. Pvt. Ltd)