



PROJECT REPORT

On

RTL TO GDSII FLOW OF PicoRV 32-BIT PROCESSOR

Submitted in partial fulfillment of the requirements for successful completion of the

Integrated VLSI Design Course Training

Under the G3 Batch

By

PREETHI GRACE MANTHENA - B. Tech graduate, 2024

ELECTRONICS AND COMMUNICATION ENGINEERING

Under the guidance and supervision of

SRI. VEERAMANI SIR

STAFF ENGINEER, Synopsys



SURE ProEd

Sreeguru Towers, Second Floor, Gopuram Road, Opp. Union Bank of India, Puttaparthi,

Andhra Pradesh – 515134

Website: www.suretrustforruralyouth.com



Declaration

The project titled "**RTL TO GDSII FLOW OF PicoRV 32-BIT PROCESSOR**" has been mentored by Sri. Veeramani, organized by SURE ProEd, from March 2024 to April 2025, for the benefit of educated, unemployed rural youth to gain hands-on experience in working on industry-relevant projects that would take them closer to prospective employers.

I declare that, to the best of my knowledge, the student mentioned below has worked on this project successfully and has enhanced practical knowledge in the VLSI design domain.

Student Name:

Ms. Preethi Grace Manthena

Mentor's Details:

Sri. Veeramani
Staff Engineer – Synopsys

Authorized Signatory:

Prof. Radhakumari
Executive Director & Founder
SURE ProEd



ACKNOWLEDGEMENT

I would like to express my deepest gratitude to SURE ProEd for organizing the G3 Integrated VLSI Design Course Training and for providing us with the opportunity to gain hands-on exposure to industry-standard tools and flows.

It is with immense respect and appreciation that I acknowledge the efforts of all the trainers and mentors who guided me throughout the course:

- Mr. Nikhil Sir, for RTL Design, and for helping us strengthen our fundamentals in digital design.
- Mr. Devi Prasad Sir, for RTL Verification, whose insights into test benches and simulation helped us gain confidence in verifying digital logic.
- Mr. Aditya Sir, for Synthesis and DFT, where we learned to bridge the gap between RTL and gate-level design with precision.
- Ms. Reshma Ma'am, for Physical Design, whose sessions built our understanding of real-world implementation techniques.
- Mr. Prudvi Sir, for Analog Mixed Signal (AMS), and for making complex analog concepts accessible and exciting.

I want to express my heartfelt thanks to my beloved mentor, ***Sri Veeramani Sir***, a Staff Engineer at Synopsys, for his invaluable mentorship, patience, and continuous encouragement throughout every step of our project, especially through the complexities of the RTL to GDSII flow.

I also sincerely thank ***Mr. Sathish Sir*** and ***Mr. Anurag Sir*** for their constant support, timely clarifications, and encouraging words that kept us motivated during the training.

Last but not least, my heartfelt gratitude to ***Prof. Radhakumari Ma'am***, Executive Director & Founder of SURE ProEd, for her visionary leadership and continuous dedication to empowering rural youth like me with high-quality, industry-relevant training.

Thank you all for being on this fruitful journey!

By
Preethi grace



Table of Contents

Executive Summary.....	5
Introduction.....	6
Methodology & Results.....	8
OpenLane.....	8
Constraints Followed.....	8
Configuration File.....	8
Running the Flow.....	10
Synthesis.....	12
Floorplan.....	17
Power Planning.....	17
Placement.....	25
Global Placement (Coarse Placement).....	26
Detailed Placement.....	26
Clock Tree Synthesis (CTS).....	44
Routing.....	54
Signoff.....	63
Social / Industry Relevance of the Project.....	71
Learning & Reflection.....	72
Future Scope & Conclusion.....	72
Future Scope:.....	73
Conclusion:.....	73



Executive Summary

This project, “RTL to GDSII Flow of PicoRV 32-Bit Processor”, was done as part of the G3 Integrated VLSI Design Course Training under SURE ProEd. The main goal was to learn and practice the complete process of designing a chip, from writing the RTL code to generating the final GDSII layout, which is ready for fabrication.

We used an open-source tool called OpenLane for this project. The processor we worked on is called PicoRV32, a small and efficient RISC-V core. Step by step, we completed all the important stages of an ASIC design flow: synthesis, floorplanning, placement, clock tree synthesis, routing, and finally, signoff. At each step, we applied proper design rules, fixed any errors, and made sure our design met the required performance, area, and power targets.

This hands-on project helped us understand how a real chip design happens in the industry. It taught us how to deal with timing issues, manage layout space, and make sure the chip works properly before it's manufactured. Overall, it was a valuable learning experience that brought together both theory and practical skills in Physical design.



Introduction

In today's digital world, VLSI (Very Large Scale Integration) design plays a key role in building the tiny yet powerful chips inside our electronic devices. From smartphones to laptops, most gadgets we use every day rely on these integrated circuits. Learning how to design them from scratch gives us a deep understanding of how real hardware is made.

This project focuses on taking a RISC-V 32-bit processor called PicoRV32 and implementing it using the RTL to GDSII flow. This means we start from a high-level hardware description (RTL - Register Transfer Level) and go all the way to the final layout (GDSII), which is what a foundry uses to fabricate the chip.

I used OpenLane, an open-source tool, to carry out this flow. The process included different stages like synthesis (converting RTL to logic), partitioning, floorplanning (setting up the layout), power planning(PDN generation), placement, clock tree synthesis (CTS), routing (connecting everything), and signoff (final checks). Along the way, I applied specific design constraints like timing uncertainty, derating, floorplan utilization, and placing the input and output ports on a specific metal layer, in control, to make the design more realistic and production-ready.

This project helped me not only learn the Physical Design flow practically but also understand how industry-level design decisions are made at every stage.



PROJECT OBJECTIVES

The primary objective of this project is to implement a complete RTL to GDSII flow for a 32-bit RISC-V processor (PicoRV32) using the OpenLane tool. This involves taking the RTL description through various stages of the ASIC physical design flow, including synthesis, floorplan, placement, clock tree synthesis (CTS), routing, and signoff checks.

The specific goals of the project are:

1. Design Constraints:

- Need to set synthesis uncertainty to 15% of the clock period.
- Setting post-CTS uncertainty to 5% of the clock period.
- Applying a 3% derate throughout the design to account for process variation.
- Defining input and output delays as 50% of the clock period.

2. Floorplan Targets:

- Maintaining the utilization target of 60%.
- Floorplan aspect ratio set to 0.7.
- Ensuring port placement with inputs on one side and outputs on the opposite side using Metal4.
- Enabling core ring structure for power delivery.
- Defining a placement blockage in any area to control congestion.

3. Clock Tree Design:

- Setting CTS Target skew of 5% of the clock period.
- Inserting clock network delay after CTS to meet timing goals.



Methodology & Results

This section details the end-to-end flow that I followed for implementing the RTL to GDSII design of the PicoRV32 32-bit processor using OpenLane.

OpenLane

OpenLane is an open-source RTL to GDSII design flow. It automates every stage of the ASIC physical design process using tools like Yosys, OpenROAD, Magic, Netgen, OpenSTA, CVC, SPEF-Extractor, KLayout. It supports both sky130 and gf180mcu PDKs.

Constraints Followed

- Uncertainty: 15% of clock period pre-CTS, 5% post-CTS
- Derate: 3% throughout the design
- Input/Output Delay: 40% of the clock period
- Utilization: 60% set during floorplanning
- Aspect Ratio: 0.7
- Port Placement: Inputs on top, outputs at bottom on Metal4, and a few on sides on chip on Metal 3
- Core Ring: Enabled
- Placement Blockage: (Not supported by OpenLane 1.1.1)
- CTS Skew Target: (Removed in latest OpenLane)
- Post CTS Delay Insertion: Enabled



Configuration File

All above constraints were specified in `config.tcl`, which serves as the central file to control the entire flow.
Default config.tcl file

```
set ::env(DESIGN_NAME) {picorv}
set ::env(VERILOG_FILES) [glob $::env(DESIGN_DIR)/src/*.v]
set ::env(CLOCK_PORT) "clk"
set ::env(CLOCK_PERIOD) "10.0"

set ::env(FP_PDN_MULTILAYER) {1}

set tech_specific_config "$::env(DESIGN_DIR)/$::env(PDK)_$::env(STD_CELL_LIBRARY)_config.tcl"
if { [file exists $tech_specific_config] == 1 } {
    source $tech_specific_config
}
~
```

Updated according to our constraints: -

```
set ::env(DESIGN_NAME) {picorv32}
set ::env(VERILOG_FILES) [glob $::env(DESIGN_DIR)/src/*.v]
set ::env(CLOCK_PORT) "clk"
set ::env(CLOCK_PERIOD) "10"
set ::env(SYNTH_CLOCK_UNCERTAINTY) [expr 0.15 * $::env(CLOCK_PERIOD)]
set ::env(IO_PCT) "0.5"
set ::env(SYNTH_TIMING_DERATE) "0.03"
set ::env(FP_PDN_CORE_RING) "1"
set ::env(FP_CORE_UTIL) "45"
#set ::env(FP_PDN_CORE_RING_VOFFSET) 2
#set ::env(FP_PDN_CORE_RING_HOFFSET) 2
#set ::env(FP_IO_MIN_DISTANCE) "10"
#set ::env(SYNTH_STRATEGY) "DELAY 3"
#set ::env(PNR_SDC_FILE) "/home/preethi-grace/OpenLane/designs/picorv32/src/picorv32.sdc"
#set ::env(SIGNOFF_SDC_FILE) "/home/preethi-grace/OpenLane/designs/picorv32/src/picorv32.sdc"
#set ::env(PDK_ROOT) "/home/preethi-grace/openlane/pdks/volare/sky130/versions/bdc9412b3e468c102d01b7cf6337be06ec6e9c9a"
set ::env(PDK) "sky130A"
set ::env(STD_CELL_LIBRARY) "sky130_fd_sc_hd"
set ::env(LVS_EXTRA_STD_CELL_LIBRARY) "$::env(PDK_ROOT)/sky130A/libs.ref/sky130_fd_sc_hd/spice/sky130_fd_sc_hd.spice"
set ::env(LIB_SYNTH) "$::env(PDK_ROOT)/sky130A/libs.ref/sky130_fd_sc_hd/lib/sky130_fd_sc_hd_tt_025C_1v80.lib"
set ::env(LIB_FASTEST) "$::env(PDK_ROOT)/sky130A/libs.ref/sky130_fd_sc_hd/lib/sky130_fd_sc_hd_ff_100C_1v95.lib"
set ::env(LIB_SLOWEST) "$::env(PDK_ROOT)/sky130A/libs.ref/sky130_fd_sc_hd/lib/sky130_fd_sc_hd_ss_100C_1v60.lib"
#set ::env(SYNTH_DRIVING_CELL) "sky130_fd_sc_hd_inv_8"
#set ::env(CTS_ROOT_BUFFER) "sky130_fd_sc_hd_clkbuf_16"
#set ::env(RE_BUFFER_CELL) "sky130_fd_sc_hd_buf_16"
#set ::env(ROOT_CLK_BUFFER) "sky130_fd_sc_hd_clkbuf_16"
set ::env(LVS_CONNECT_BY_LABEL) 1
#set ::env(GPL_CELL_PADDING) "2"
#set ::env(DPL_CELL_PADDING) "2"
#set ::env(FP_SIZING) "absolute"
```



```
#set ::env(CORE_AREA) "10 10 730 538"
#set ::env(RT_MAX_LAYER) "met6"
#set ::env(RT_CLOCK_MAX_LAYER) "met6"
set ::env(PL_TARGET_DENSITY) "0.48"
#set ::env(OUTPUT_CAP_LOAD) "36"
#set ::env(CURRENT_INDEX) "10"
set ::env(GRT_ALLOW_CONGESTION) "1"
set ::env(FP_PIN_ORDER_CFG) "$::env(DESIGN_DIR)/pin_order.cfg"
#set ::env(PIN_CONSTRAINTS) "$::env(DESIGN_DIR)/pin_constraints.cfg"
#set ::env(PNR_SDC_FILE) "$::env(DESIGN_DIR)/src/picorv32.sdc"
#set ::env(SIGNOFF_SDC_FILE) "$::env(DESIGN_DIR)/src/picorv32.sdc"
set ::env(FP_IO_MODE) "1"
set ::env(FP_ASPECT_RATIO) "0.7"
set ::env(FP_IO_HLAYER) "met3"
set ::env(FP_IO_VLAYER) "met4"
set ::env(PL_RESIZER_SETUP_SLACK_MARGIN) "0.1"
#set ::env(PL_RESIZER_HOLD_SLACK_MARGIN) "0.1"
#set ::env(PL_RESIZER_SETUP_MAX_BUFFER_PERCENT) "100"
#set ::env(PL_RESIZER_HOLD_MAX_BUFFER_PERCENT) "100"
set ::env(GRT_REPAIR_ANTENNAS) "1"
#set ::env(DIODE_ON_PORTS) "both"
set ::env(RUN_HEURISTIC_DIODE_INSERTION) "1"
set ::env(MAX_FANOUT_CONSTRAINT) "10" ;# Reduce fanout to lower net congestion
set ::env(FP_PDN_MULTILAYER) {1}

set tech_specific_config "$::env(DESIGN_DIR)/$::env(PDK)_$::env(STD_CELL_LIBRARY)_config.tcl"
if { [file exists $tech_specific_config] == 1 } {
    source $tech_specific_config
}
```

Running the Flow

- Step 1: Enter OpenLane directory

```
preethi-grace@HP-245-G7-Notebook:~$ cd OpenLane
```

- Step 2: Mount OpenLane using `make mount`

```
preethi-grace@HP-245-G7-Notebook:~/OpenLane$ make mount
cd /home/preethi-grace/OpenLane && \
    docker run --rm -v /home/preethi-grace:/home/preethi-grace -v /home/preethi-grace/OpenLane:/openlane -v /home/preethi-grace/OpenLane/empty:/openlane/install -v /home/preethi-grace/.volare:/home/preethi-grace/.volare -e PDK_ROOT=/home/preethi-grace/.volare -e PDK=sky130A --user 1000:1000 -e DISPLAY=:1 -v /tmp/.X11-unix:/tmp/.X11-unix -v /home/preethi-grace/.Xauthority:/Xauthority --network host --security-opt seccomp=unconfined -ti efabless/openlane:e73fb3c57e687a0023fcfd4dcfd1566ecd478362a-amd64
OpenLane Container (1.1.1):/openlane% 
```

This command launches the OpenLane Docker container. (*Here, I used OpenLane version 1.1.1*)

Now we need to run the design flow using `flow.tcl`

Firstly, execute the `flow.tcl` file and name the design folder

- Step 3: Launch design flow with: `./flow.tcl -design picorv32 -tag run`



```
OpenLane Container (1.1.1):/openlane% ./flow.tcl -design picorv32 -tag run
```

This executes all the steps from synthesis to GDSII (non-interactive mode).

```
OpenLane Container (1.1.1):/openlane% ./flow.tcl -design picorv32 -tag finalfix -overwrite
OpenLane v1.1.1 (e73fb3c57e687a0023fcfd4dcfd1566ecd478362a)
All rights reserved. (c) 2020-2024 Efabless Corporation and contributors.
Available under the Apache License, version 2.0. See the LICENSE file for more details.

[INFO]: Using configuration in 'designs/picorv32/config.tcl'...
[INFO]: Process Design Kit: sky130A
[INFO]: PDK Root: /home/preethi-grace/.volare
[INFO]: Standard Cell Library: sky130_fd_sc_hd
[INFO]: Optimization Standard Cell Library: sky130_fd_sc_hd
[INFO]: Run Directory: /openlane/designs/picorv32/runs/finalfix
[INFO]: Removing existing /openlane/designs/picorv32/runs/finalfix...
[INFO]: Saving runtime environment...
[INFO]: Preparing LEF files for the nom corner...
[INFO]: Preparing LEF files for the min corner...
[INFO]: Preparing LEF files for the max corner...
[WARNING]: PNR_SDC_FILE is not set. It is recommended to write a custom SDC file for the design. Defaulting to BASE_SDC_FILE
[WARNING]: SIGNOFF_SDC_FILE is not set. It is recommended to write a custom SDC file for the design. Defaulting to BASE_SDC_FILE
[INFO]: Running linter (Verilator) (log: designs/picorv32/runs/finalfix/logs/synthesis/linter.log)...
[INFO]: 0 errors found by linter
```

Physical Design is the process of transforming a digital circuit's RTL description into a physical layout that can be fabricated on a silicon chip. It involves placing standard cells, routing interconnects, ensuring timing closure, and checking for DRC/LVS violations. The primary goal is to optimize area, power, and performance while adhering to all foundry constraints.

The flow includes:

- Synthesis: Converting RTL to gate-level netlist.
- Floorplan: Defining die/core area and placing ports/macros.
- Placement: Placing standard cells.
- Clock Tree Synthesis (CTS): Distributing the clock signal uniformly.
- Routing: Connecting all cells with metal wires.
- Signoff: Final checks including timing, power, DRC/LVS, and GDSII generation.



Synthesis

Converts RTL to a gate-level netlist using the standard cell libraries, which are from the foundry.

For example, One commonly used standard cell library is `sky130_fd_sc_hd_tt_025C_1v80.lib`. This indicates:

- sky130_fd_sc_hd: Process node (130nm), foundry (SkyWater), standard cells (SC), high density (HD) variant
- tt: Typical-Typical process corner
- 025C: Temperature at 25°C
- 1v80: Operating voltage is 1.8V

Inputs:

- RTL Verilog files(.v)
- config.tcl
- Liberty file (.lib)
- Timing constraints (.sdc)
- Technology LEF file (.tlef)

Steps:

Converting a high-level Hardware Description Language (HDL) design into a gate-level netlist, optimizing it, and then mapping it to a target technology library

Quality Checks:

- Timing: $\text{WNS} \geq 0$, $\text{TNS} = 0$, $\text{NVP} = 0$

WNS (Worst Negative Slack) ≥ 0

TNS (Total Negative Slack) = 0

NVP (Number of Violating Paths) =0



```
=====
report_tns
=====
tns -3.97
=====

report_wns
=====
wns -2.08
=====

report_worst_slack -max (Setup)
=====
worst slack -2.08
=====

report_worst_slack -min (Hold)
=====
worst slack -1.07
2-syn_sta.summary.rpt (END)
```

```
=====
report_clock_skew
=====
Clock clk
Latency      CRPR      Skew
_13856_/CLK ^   9.42
_13856_/CLK ^   8.87      0.00      0.55
2-syn_sta.skew.rpt (END)
```

Uncertainty and derate

```
Setting output delay to: 5.0
Setting input delay to: 5.0
Setting load to: 0.033442
Setting clock uncertainty to: 1.5
Setting clock transition to: 0.15
Setting timing derate to: 3.0 %
```

- Area and Power within the limit

```
Chip area for module '\picorv32': 96597.644800
```



```
=====
report_power
=====
===== Typical Corner =====

Group           Internal Power   Switching Power   Leakage Power   Total Power (Watts)
-----
Sequential      6.92e-03    7.36e-04    1.37e-08    7.65e-03    56.9%
Combinational   3.30e-03    2.49e-03    2.49e-08    5.79e-03    43.1%
Clock           0.00e+00    0.00e+00    0.00e+00    0.00e+00    0.0%
Macro           0.00e+00    0.00e+00    0.00e+00    0.00e+00    0.0%
Pad             0.00e+00    0.00e+00    0.00e+00    0.00e+00    0.0%

Total          1.02e-02    3.22e-03    3.86e-08    1.34e-02   100.0%
                76.0%        24.0%        0.0%
```

2-syn_sta.power.rpt (END)

- Cell report

```
== picorv32 ==
Number of wires:          8400
Number of wire bits:       8782
Number of public wires:    1513
Number of public wire bits: 1895
Number of memories:        0
Number of memory bits:     0
Number of processes:       0
Number of cells:
  sky130_fd_sc_hd_a2111o_2      5
  sky130_fd_sc_hd_a2111oi_2     1
  sky130_fd_sc_hd_a211o_2      91
  sky130_fd_sc_hd_a211oi_2      8
  sky130_fd_sc_hd_a21bo_2      24
  sky130_fd_sc_hd_a21boi_2      3
  sky130_fd_sc_hd_a21o_2      251
  sky130_fd_sc_hd_a21oi_2     118
  sky130_fd_sc_hd_a221o_2      105
  sky130_fd_sc_hd_a22o_2      183
  sky130_fd_sc_hd_a22oi_2      1
  sky130_fd_sc_hd_a2bb2o_2      20
  sky130_fd_sc_hd_a311o_2      12
  sky130_fd_sc_hd_a31o_2      123
  sky130_fd_sc_hd_a31oi_2      4
  sky130_fd_sc_hd_a32o_2      29
  sky130_fd_sc_hd_a32oi_2      1
:
```



Outputs:

- **Synthesized Netlist**

```
/* Generated by Yosys 0.38 (git sha1 543faed9c8c, clang++ 17.0.6 -fPIC -Os) */

module picorv32(clk, resetn, trap, mem_valid, mem_instr, mem_ready, mem_addr, mem_wdata, mem_wstrb, mem_rdata, mem_la_read, mem_l
a_write, mem_la_addr, mem_la_wdata, mem_la_wstrb, pcpi_valid, pcpi_insn, pcpi_rs1, pcpi_rs2, pcpi_wr, pcpi_rd
, pcpi_wait, pcpi_ready, irq, eoi, trace_valid, trace_data);
    wire _00000;
    wire _00001;
    wire _00002;
    wire _00003;
    wire _00004;
    wire _00005;
    wire _00006;
    wire _00007;
```

```
.A(mem_la_wdata[1]),
.X(pcpi_rs2[1])
);
sky130_fd_sc_hd_buf_2 _19555_ (
.A(mem_la_wdata[2]),
.X(pcpi_rs2[2])
);
sky130_fd_sc_hd_buf_2 _19556_ (
.A(mem_la_wdata[3]),
.X(pcpi_rs2[3])
);
sky130_fd_sc_hd_buf_2 _19557_ (
.A(mem_la_wdata[4]),
.X(pcpi_rs2[4])
);
sky130_fd_sc_hd_buf_2 _19558_ (
.A(mem_la_wdata[5]),
.X(pcpi_rs2[5])
);
sky130_fd_sc_hd_buf_2 _19559_ (
.A(mem_la_wdata[6]),
.X(pcpi_rs2[6])
);
sky130_fd_sc_hd_buf_2 _19560_ (
.A(mem_la_wdata[7]),
.X(pcpi_rs2[7])
);
endmodule
"picorv32.v" 67606L, 1282539B
```



- **Simulation Description Format(.sdf)**

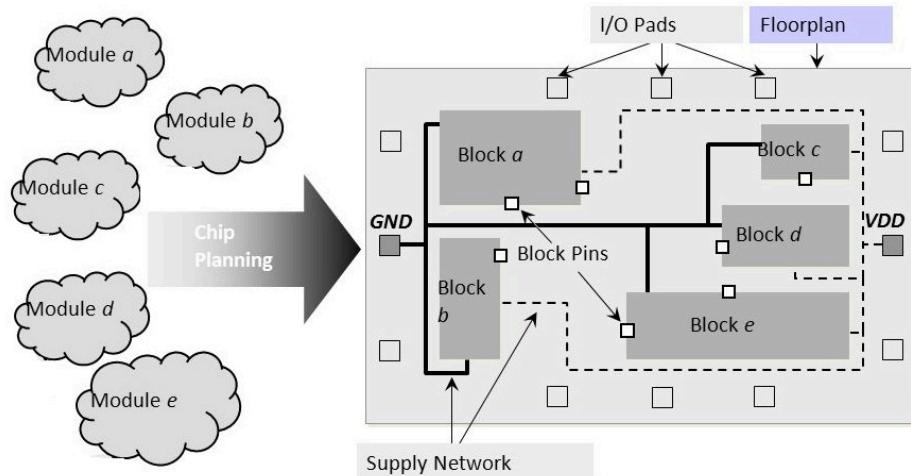
```
/home/preethi-grace/OpenLane/designs/picorv32/runs/finalone/results/synthesis/picorv32.sdf
```

```
(DELAYFILE
  (SDFVERSION "3.0")
  (DESIGN "picorv32")
  (DATE "Wed Apr 23 11:31:33 2025")
  (VENDOR "Parallax")
  (PROGRAM "STA")
  (VERSION "2.5.0")
  (DIVIDER .)
  (VOLTAGE 1.800::1.800)
  (PROCESS "1.000::1.000")
  (TEMPERATURE 25.000::25.000)
  (TIMESCALE 1ns)
  (CELL
    (CELLTYPE "picorv32")
    (INSTANCE)
    (DELAY
      (ABSOLUTE
        (INTERCONNECT clk _13856_.CLK (9
```



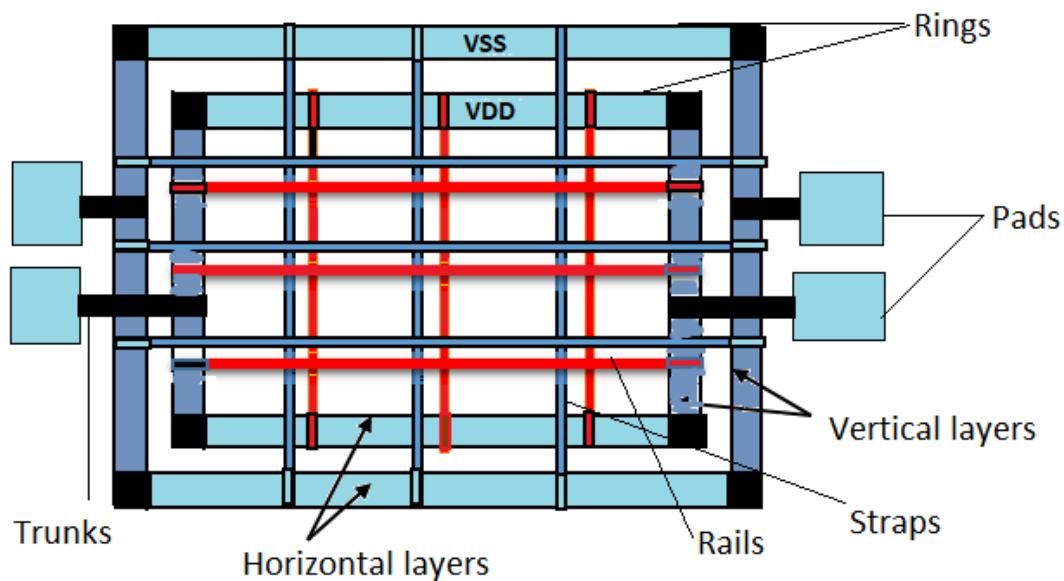
Floorplan

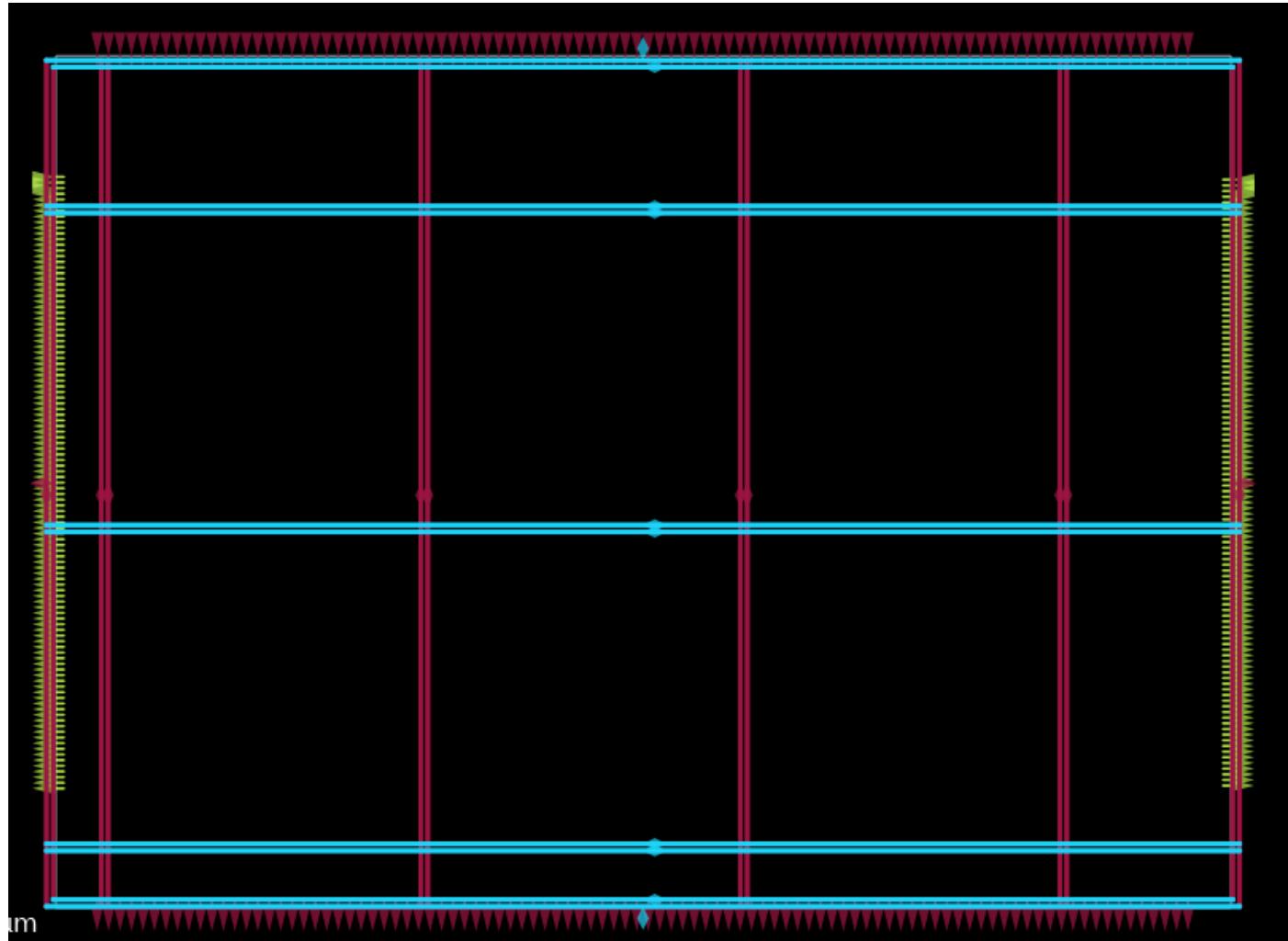
Defines the die and core area, port locations, and placement of macro blocks.



Power Planning

Power planning is the process of creating a robust and efficient power delivery network (PDN) to distribute power (VDD) and ground (VSS) across the chip. This ensures all cells receive a stable voltage with minimal IR drop and noise.





Inputs:

- Synthesized Netlist(output file from synthesis stage)
- Technology LEF(.lef)
- Constraints from config.tcl

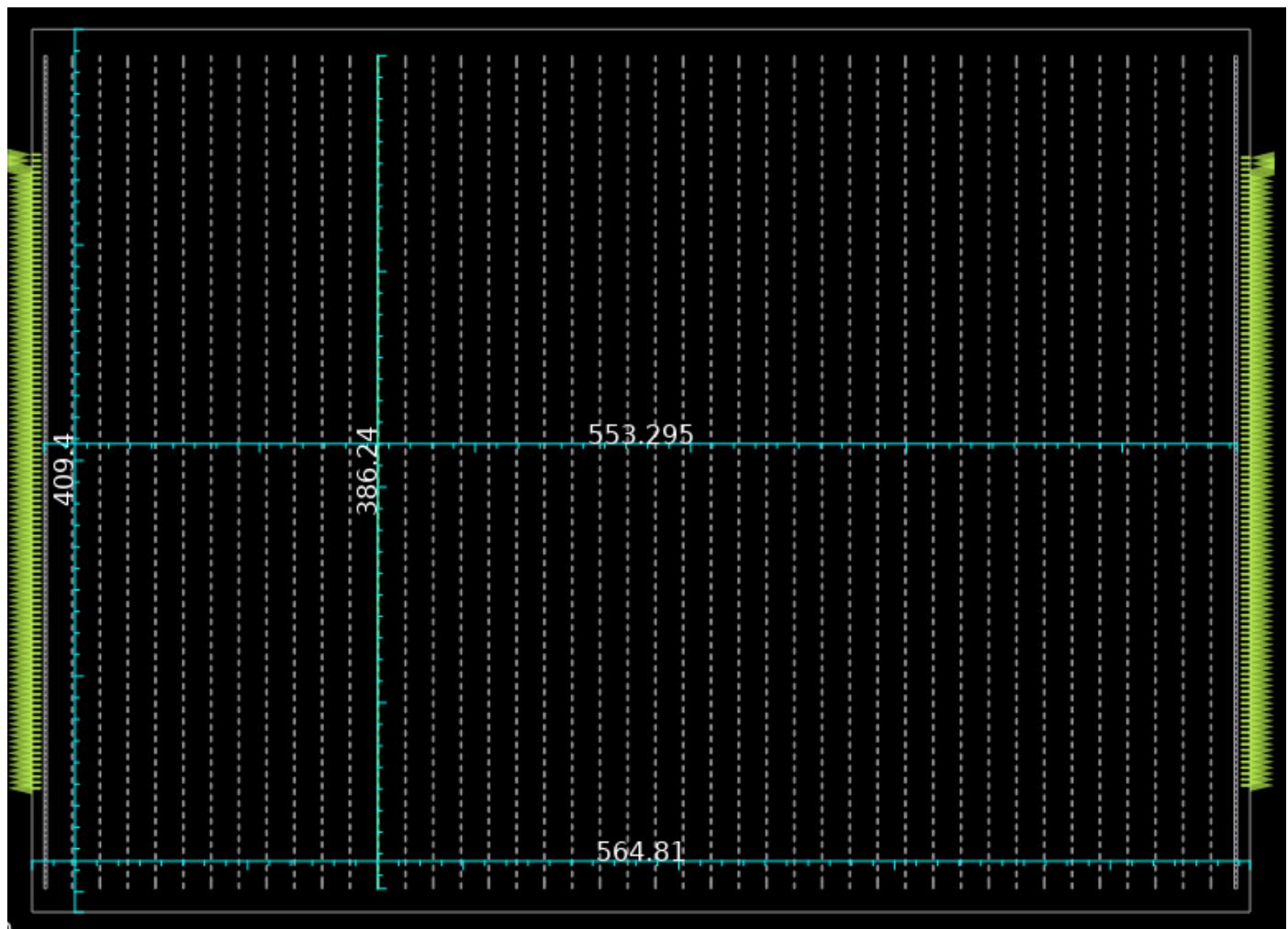
Steps:

1. Die and Core Area (0.7 aspect ratio)

```
5.52 10.88 558.9 397.12  
3-initial_fp_core_area.rpt
```



```
VERSION 5.8 ;
DIVIDERCHAR "/";
BUSBITCHARS "[]";
DESIGN picorv32 ;
UNITS DISTANCE MICRONS 1000 ;
DIEAREA ( 0 0 ) ( 564810 409400 ) ;
ROW ROW_0 unithd 5520 10880 N DO 120
```



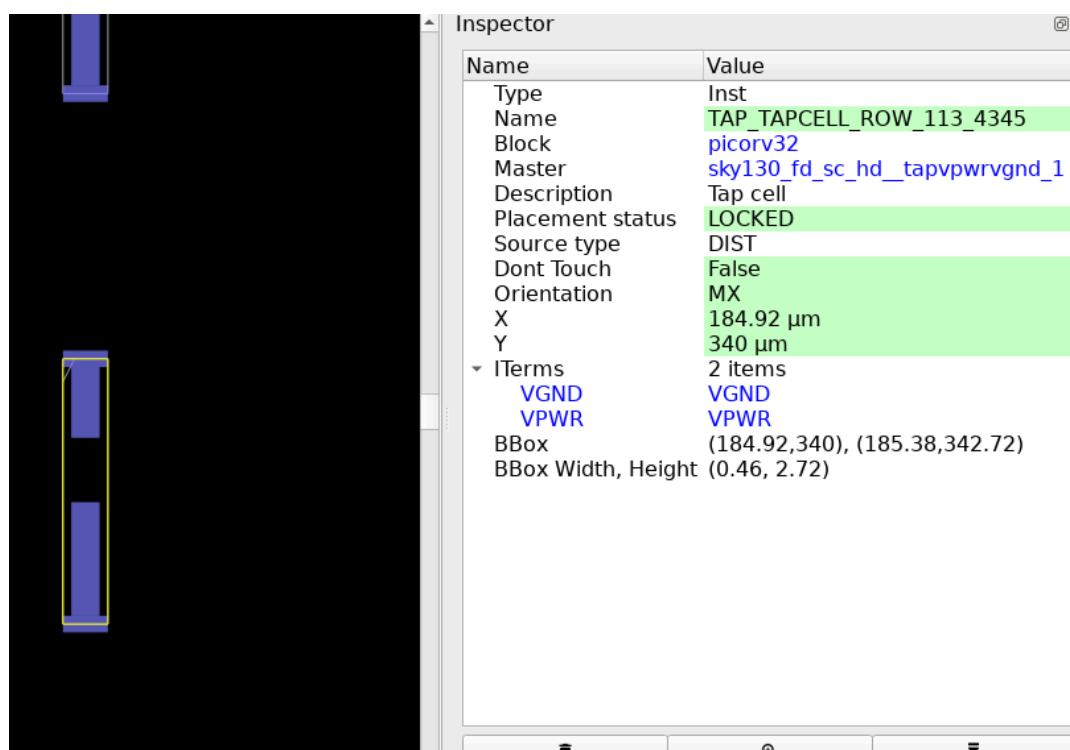
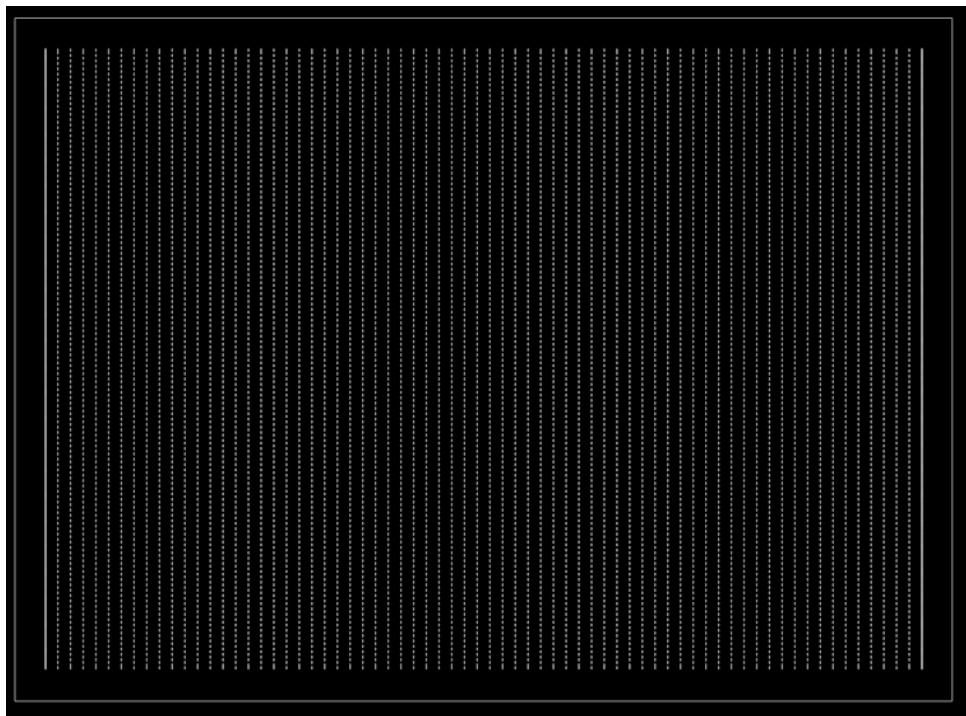


2. Port Placement (Inputs top, Outputs : bottom, sides)

<pre>#N @min_distance=0.9 \$3 irq.*</pre>	<pre>#E @min_distance=1.0 mem_wstrb.*</pre>
<pre>\$2 mem_rdata.*</pre>	<pre>mem_addr.*</pre>
<pre>\$2 pcpi_rd.*</pre>	<pre>trap</pre>
<pre>\$3</pre>	<pre>pcpi_rs1.*</pre>
<pre>#S @min_distance=0.9 \$2</pre>	<pre>mem_la_read</pre>
<pre>pcpi_insn.*</pre>	<pre>mem_la_write</pre>
<pre>\$2 pcpi_rs2.*</pre>	<pre>mem_la_wdata.*</pre>
<pre>\$2 eoи.*</pre>	<pre>pcpi_valid</pre>
	<pre>trace_valid</pre>
	<pre>pcpi_wait</pre>
	<pre>pcpi_ready</pre>
	<pre>pcpi_wr</pre>
	<pre>#W @min_distance=1.0 mem_valid</pre>
	<pre>mem_instr</pre>
	<pre>mem_la_addr.*</pre>
	<pre>mem_la_wstrb.*</pre>
	<pre>trace_data.*</pre>
	<pre>mem_wdata.*</pre>
	<pre>resetn</pre>
	<pre>clk</pre>
	<pre>mem_ready</pre>



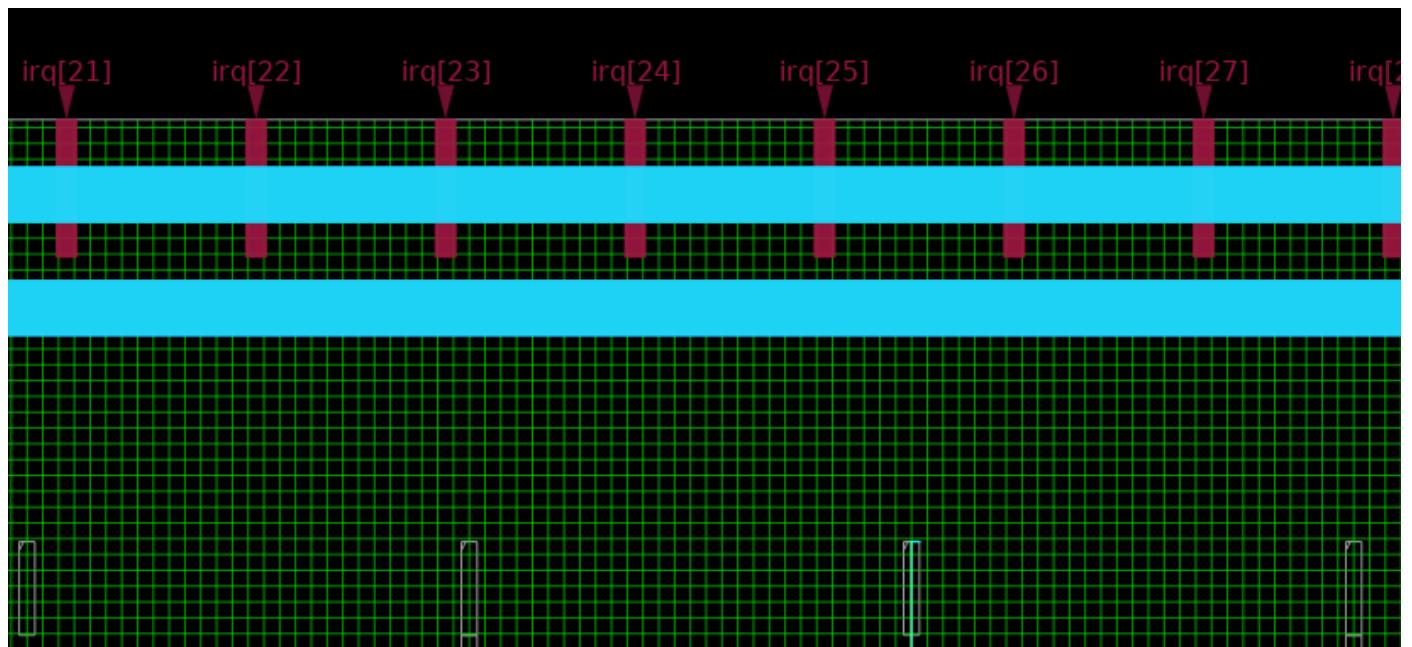
3. Physical only cell



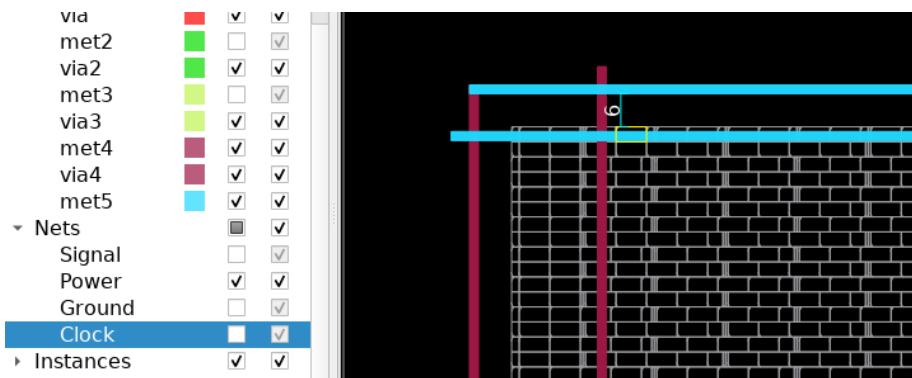


Inspector	
Name	Value
Type	Inst
Name	PHY_EDGE_ROW_130_Left_362
Block	picorv32
Master	sky130_fd_sc_hd_decap_3
Description	Fill cell
Placement status	LOCKED
Source type	DIST
Dont Touch	False
Orientation	R0
X	30.36 μ m
Y	386.24 μ m
ITerms	
VGND	VGND
VNB	VGND
VPB	VPWR
VPWR	VPWR
BBox	(30.36, 386.24), (31.74, 388.96)
BBox Width, Height	(1.38, 2.72)

4. Power planning

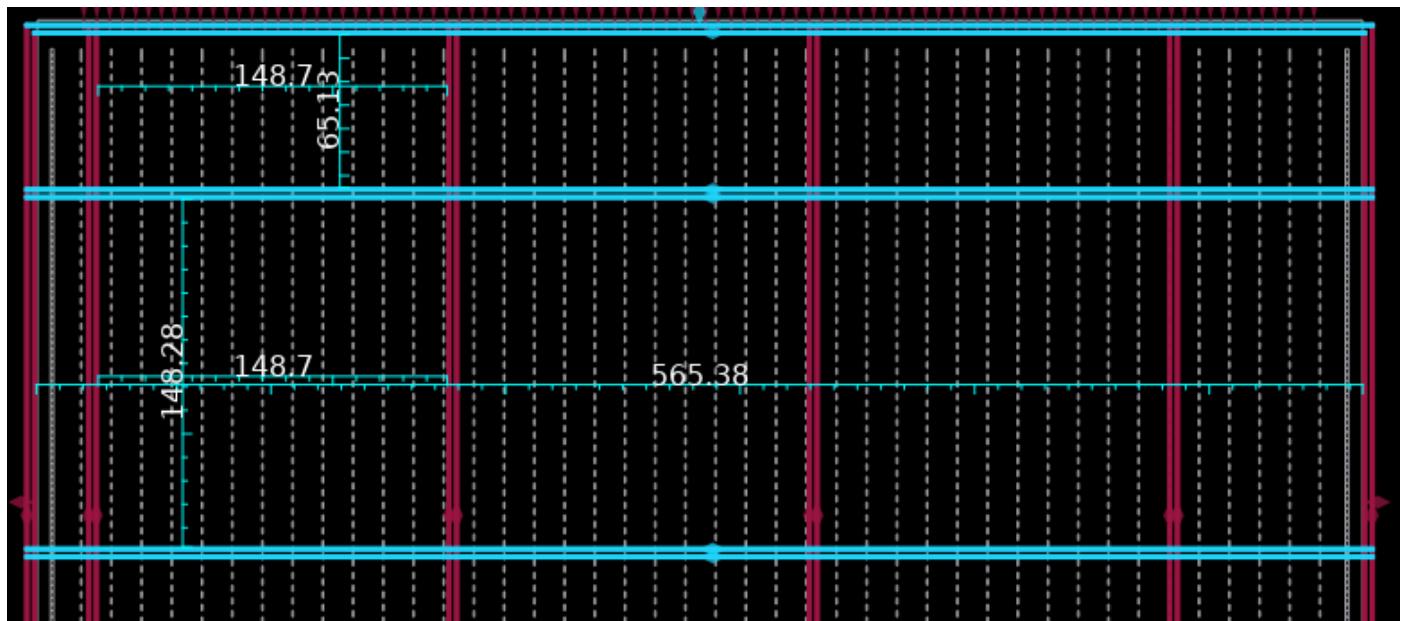


We can see PDN core ring is enabled out of the core region.



Quality Checks:

- Checking spacing between stripes and rings



Space between Stripes : 148.7 um

Space between the core ring : 565.38 um

- Timing within limits
- Utilization

Scripting

```
QSurfaceFormat::NoProfile  
>>> report_design_area  
Design area 0 u^2 47% utilization.
```



- No base DRC or PG DRC

Outputs:

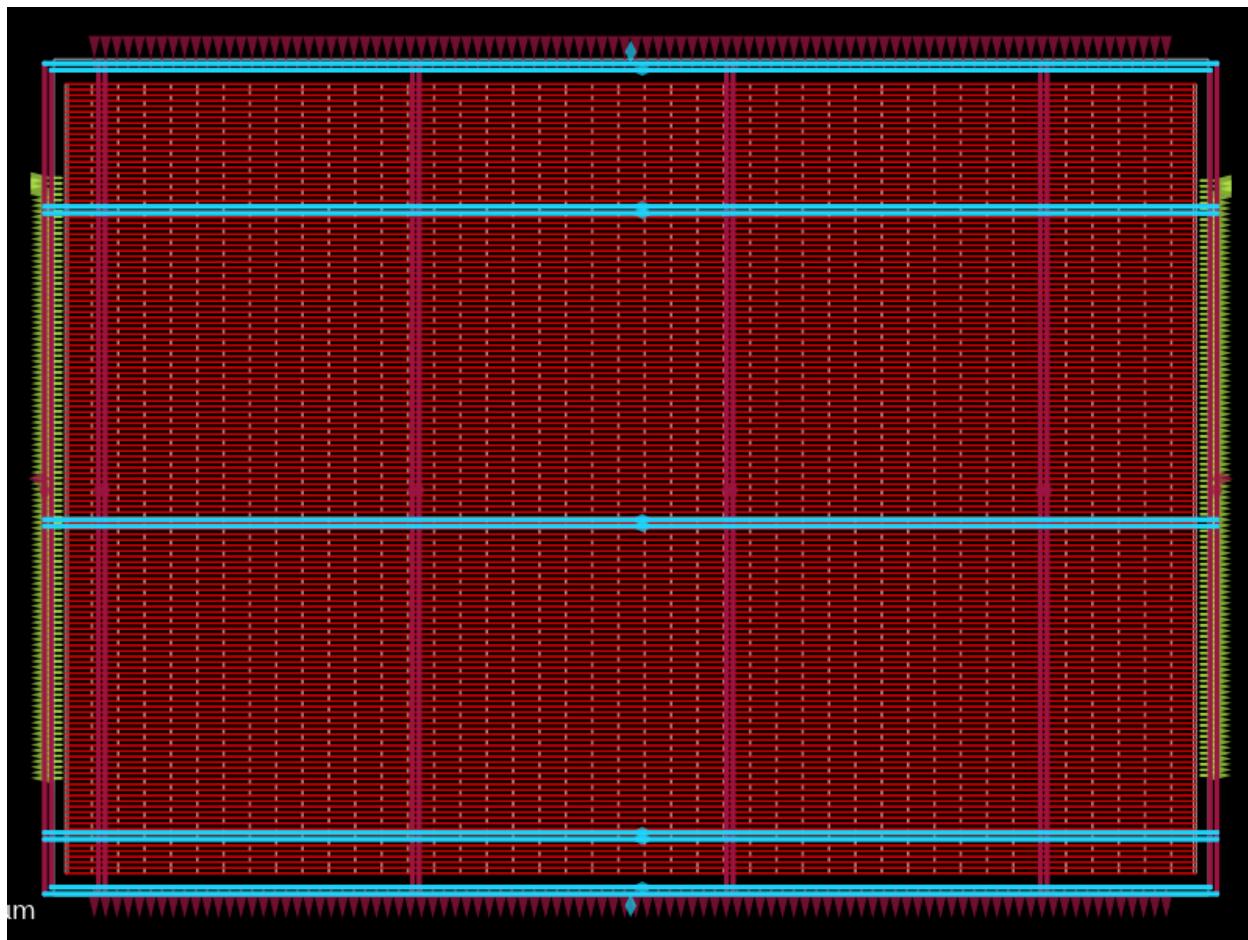
- DEF file

```
- trace_data[4] ( PIN trace_data[4] ) ( _19520_ LO ) + USE SIGNAL ;
- trace_data[5] ( PIN trace_data[5] ) ( _19521_ LO ) + USE SIGNAL ;
- trace_data[6] ( PIN trace_data[6] ) ( _19522_ LO ) + USE SIGNAL ;
- trace_data[7] ( PIN trace_data[7] ) ( _19523_ LO ) + USE SIGNAL ;
- trace_data[8] ( PIN trace_data[8] ) ( _19524_ LO ) + USE SIGNAL ;
- trace_data[9] ( PIN trace_data[9] ) ( _19525_ LO ) + USE SIGNAL ;
- trace_valid ( PIN trace_valid ) ( _19552_ LO ) + USE SIGNAL ;
- trap ( PIN trap ) ( _18366_ Q ) ( _16387_ A_N ) ( _16377_ A ) ( _15267_ A_N ) ( _15266_ A ) + USE SIGNAL ;
END NETS
END DESIGN
"picorv32.def" 39369L, 2806477B
```

3

- **Updated netlist**

The empty space between the ports are virtual ports that we defined in pin_order.cfg file.
Arrows of ports facing inside are input ports.
Arrows of ports facing outward are output ports.





Placement

Placement is the stage in physical design where synthesized standard cells are assigned physical locations within the defined core area. The objective is to ensure legal, congestion-free placement while preserving timing and optimizing area and power.

Inputs:

- Floorplanned DEF file
- Synthesized netlist
- Liberty and LEF files
- Placement constraints (blockages) are not supported by OpenLane.

Global Placement (Coarse Placement)

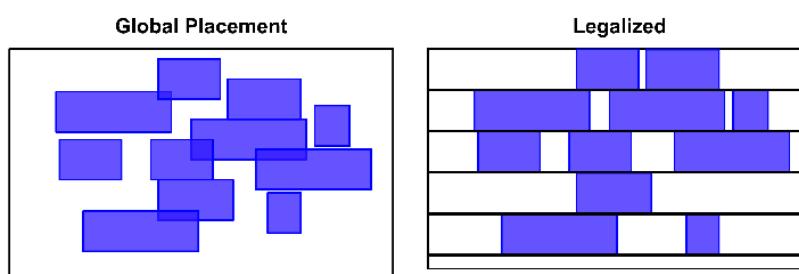
Global placement provides an initial cell distribution over the core area. It uses estimations of net delays and congestion to place cells approximately

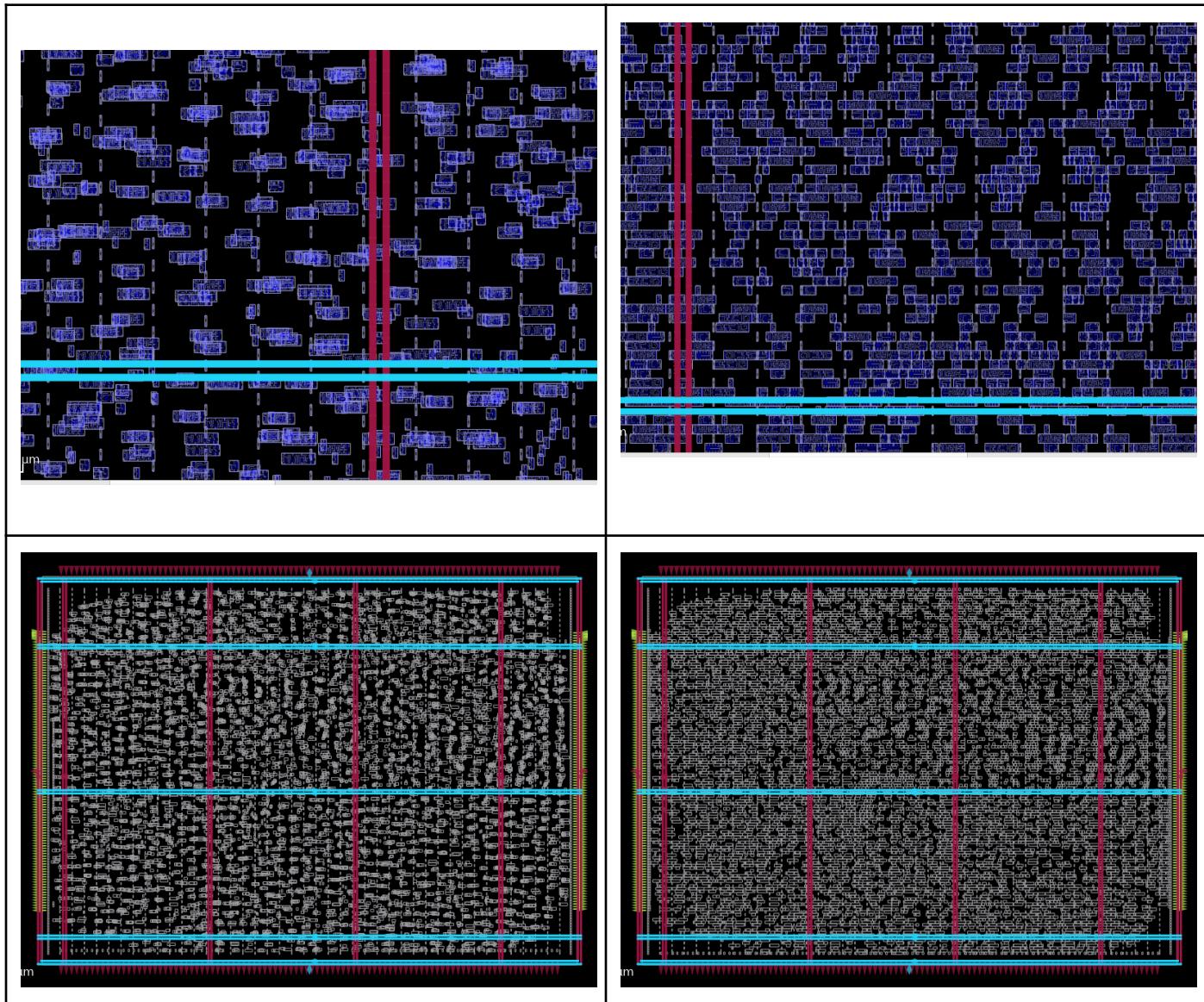
- Minimize wirelength
- Spread out the cells to reduce congestion
- Prepare for timing optimization

Detailed Placement

This step legalizes the placement done in the global stage. It ensures each standard cell is placed on legal rows without overlaps and aligns to site grids.

- Legal and rule-compliant placement
- Remove cell overlaps
- Snap cells to rows and tracks





Quality Checks

1. Timing

- Run setup and hold checks

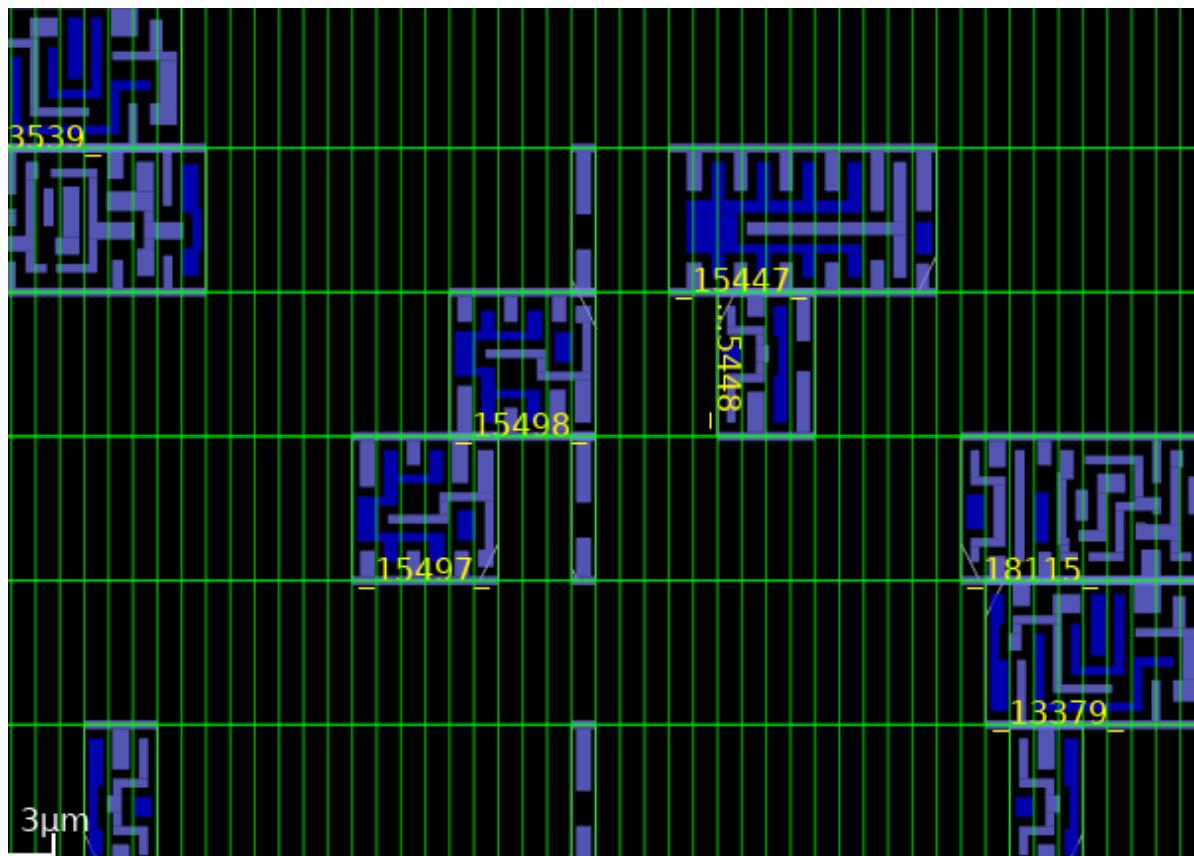


<pre>===== report_tns ===== tns -11.81</pre>	<pre>===== report_tns ===== tns -4.57</pre>
<pre>===== report_wns ===== wns -2.39</pre>	<pre>===== report_wns ===== wns -2.31</pre>
<pre>===== report_worst_slack -max (Setup) ===== worst slack -2.39</pre>	<pre>===== report_worst_slack -max (Setup) ===== worst slack -2.31</pre>
<pre>===== report_worst_slack -min (Hold) ===== worst slack -1.06</pre>	<pre>===== report_worst_slack -min (Hold) ===== worst slack -1.05</pre>
<pre>8-gpl_sta.summary.rpt (END)</pre>	<pre>11-dpl_sta.summary.rpt (END)</pre>

- Ensure no negative slack is introduced post-placement

2. Legality Check

- No overlaps between cells
- Cells must sit on site rows (grid alignment)



3. Utilization

- Capture utilization from reports

```
QSurfaceFormat::NoProfile)
>>> report_design_area
Design area 0 u^2 47% utilization.
```

Placement Blockage is not supported in Openlane, the team is working on the pavement obstruction issue. As OpenLane is currently shut down, it is sad that we may not receive updates from OpenLane.



efabless / Projects / OpenLane

Add placement obstructions #1812

Open The-OpenROAD-Project/OpenLane Public

kareefardi opened on May 22, 2023

Description

Title

Proposal

Add placement obstruction separate from routing obstruction

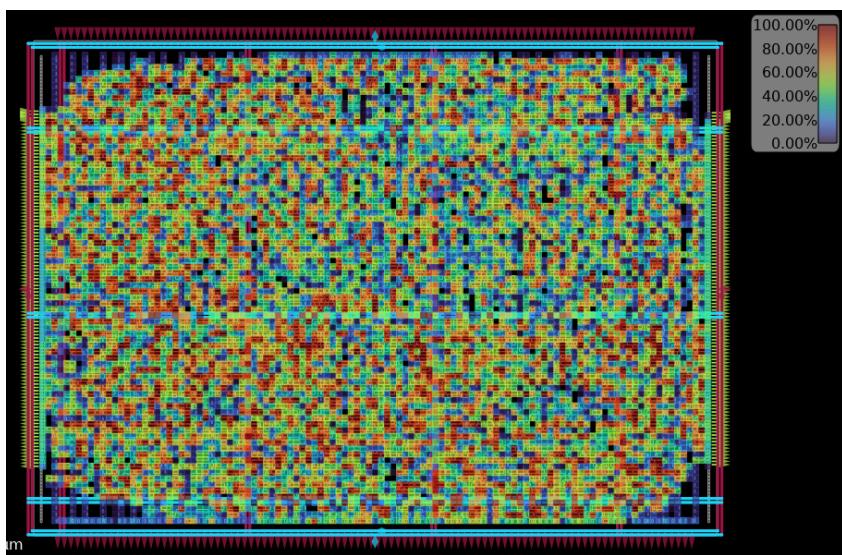
kareefardi added enhancement on May 22, 2023

kareefardi added this to OpenLane on May 23, 2023

kareefardi added OpenLane 2 on Jul 17, 2024

27 Impo
28 Revie
29 Routi
30 Add p
31 Revie
32 Upgr
33 check
34 Open
35 UPF s
36 Hier

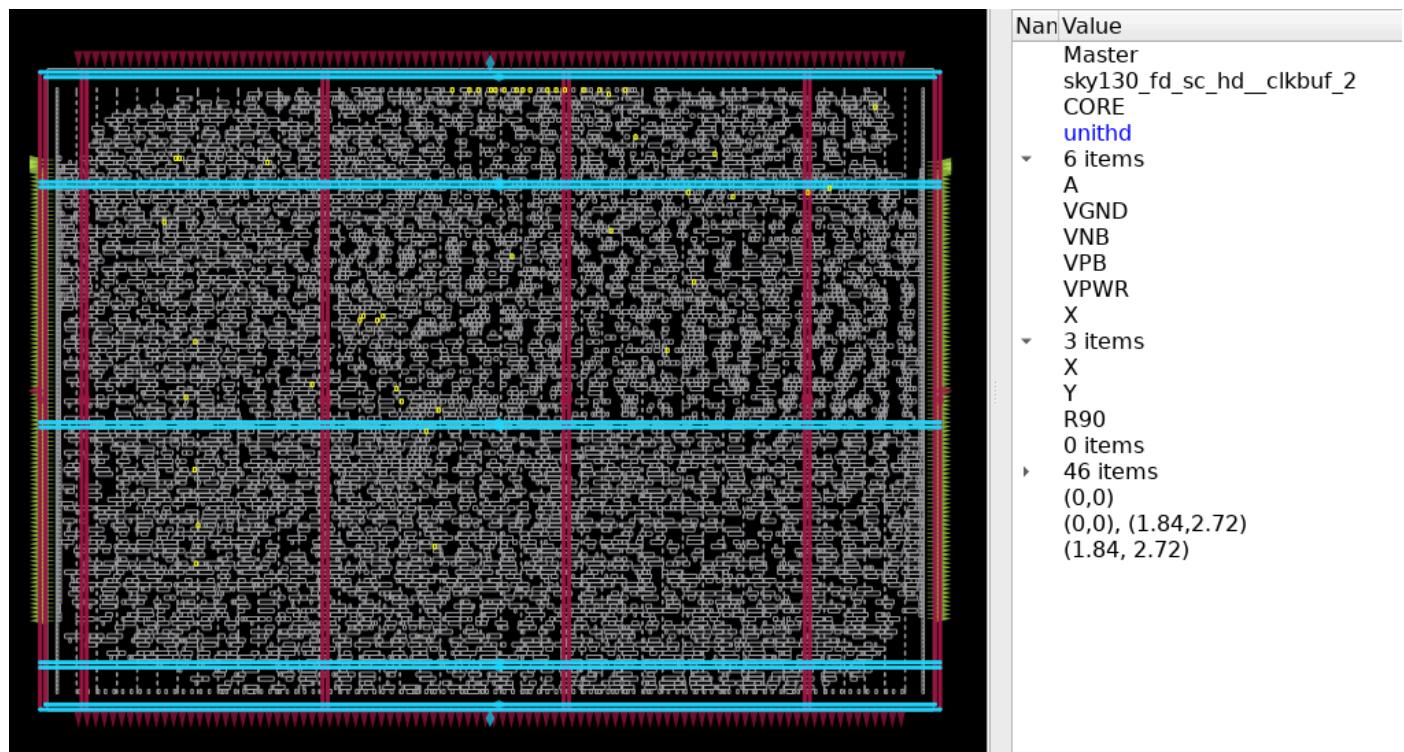
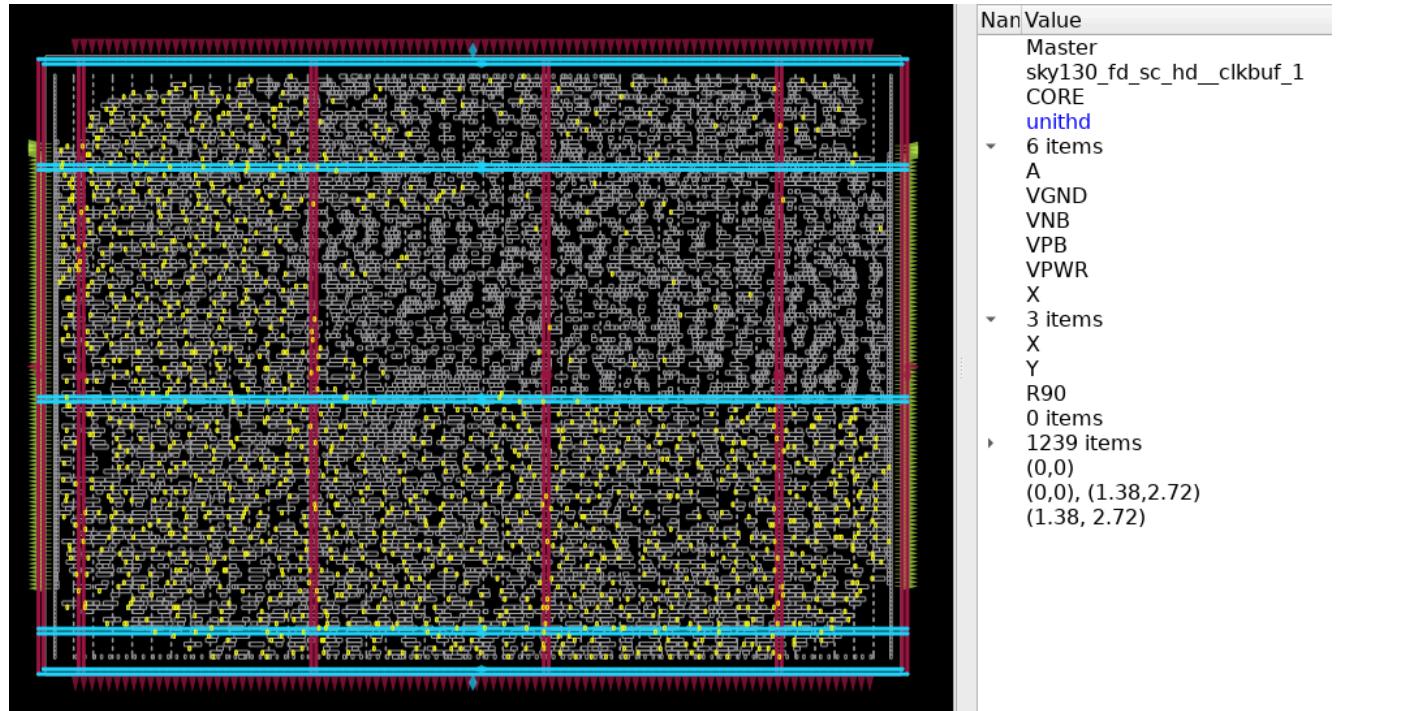
Placement density heat map

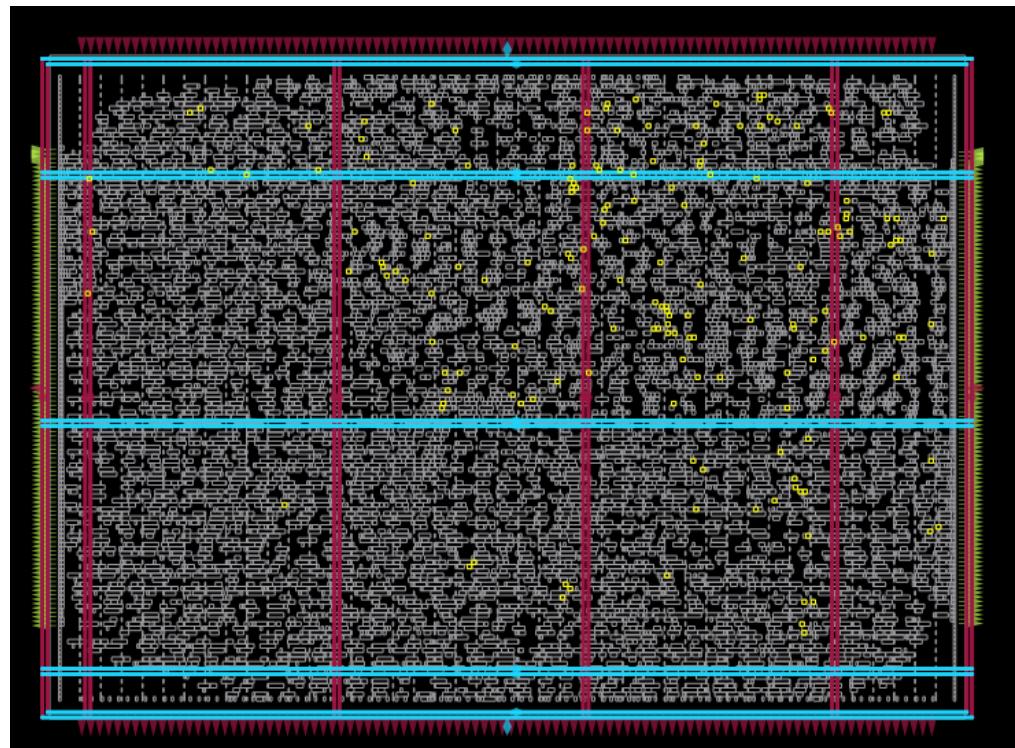




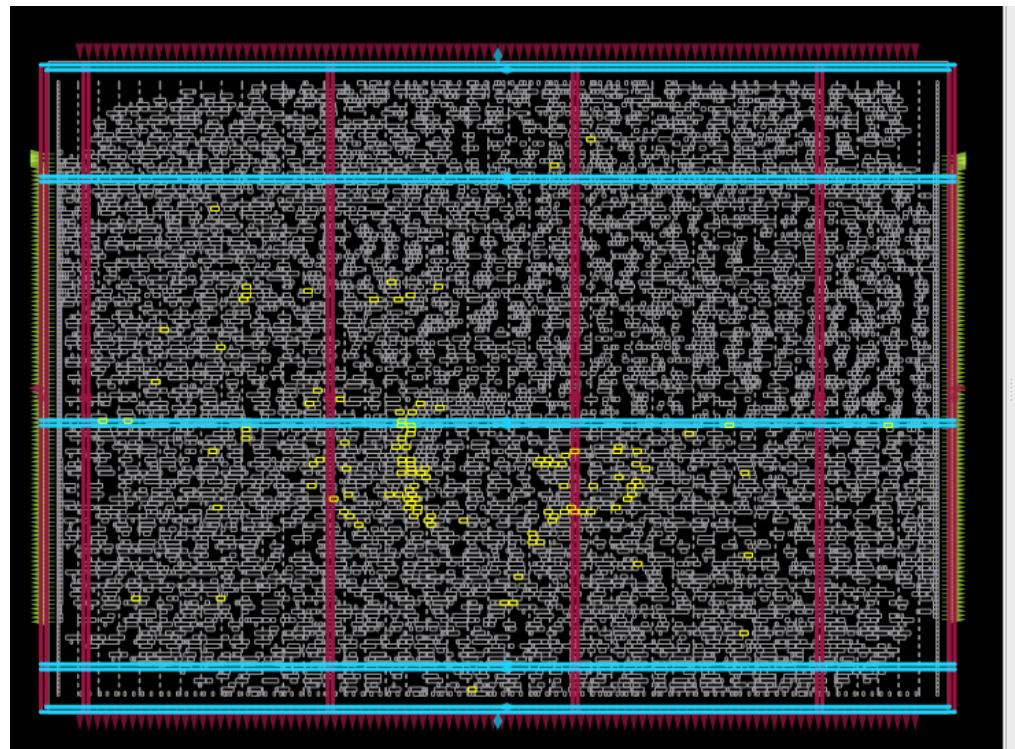
- Check if the placement added more cells due to optimization

- i. Highlight the buffers and inverters in the design

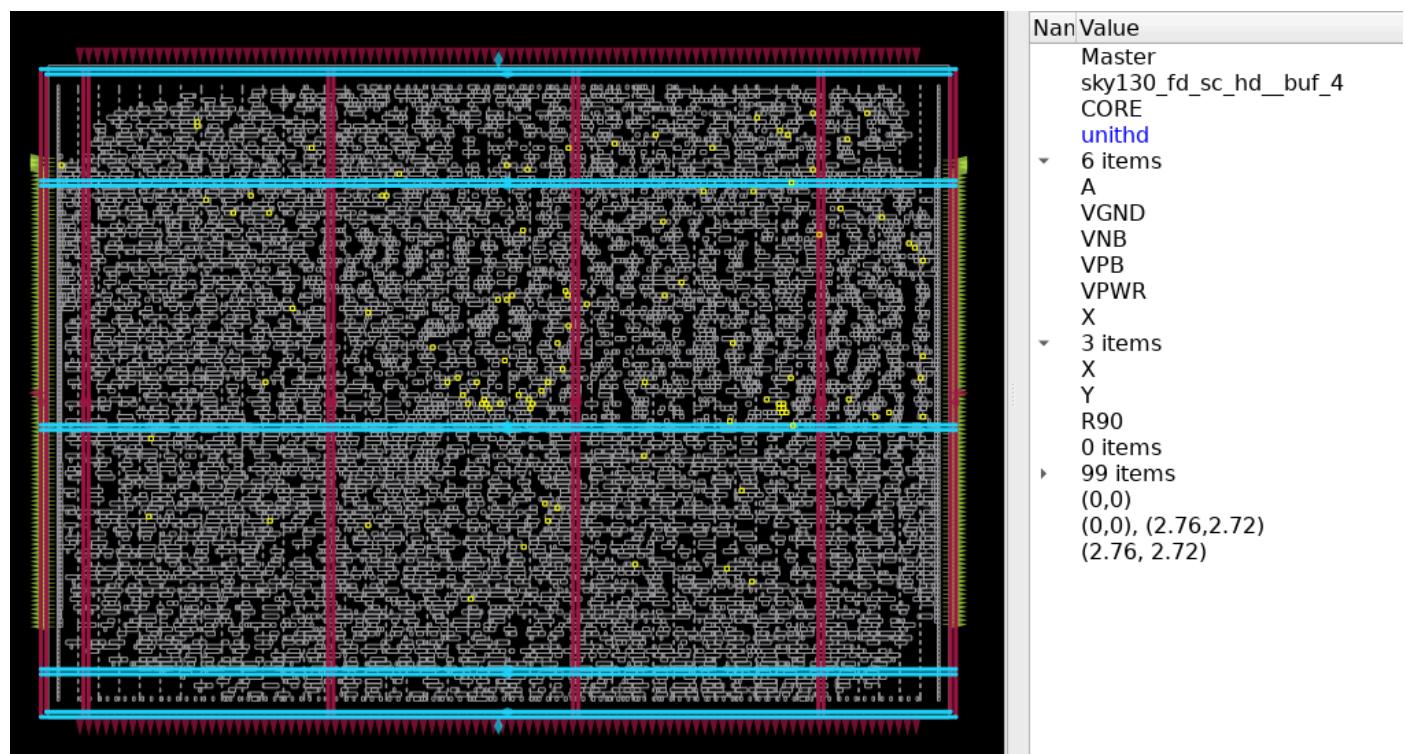
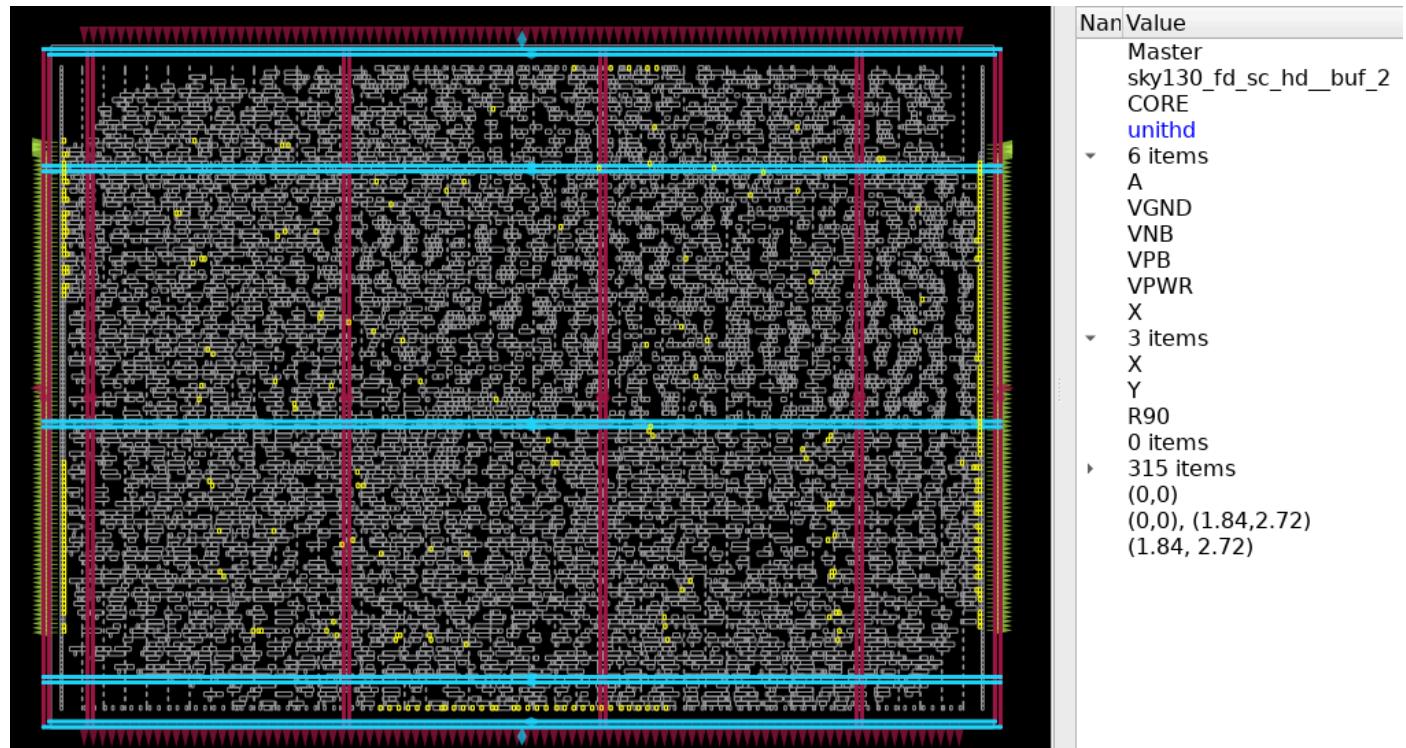


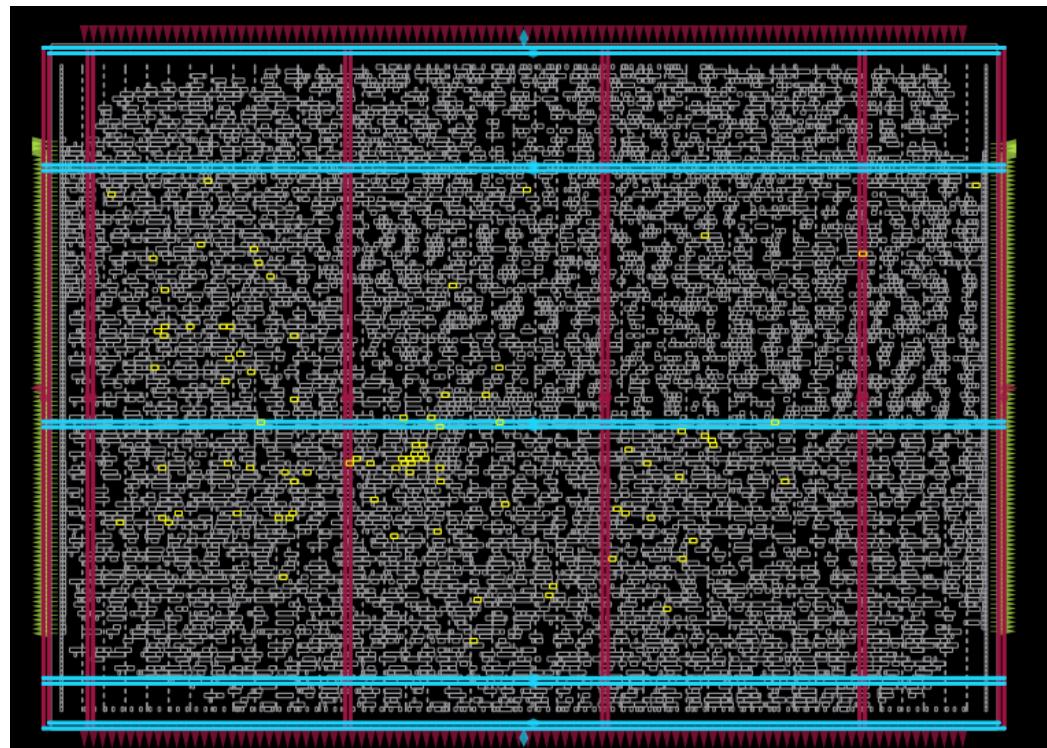


Nan	Value
Master	
sky130_fd_sc_hd_clkbuf_4	
CORE	
unithd	
6 items	
A	
VGND	
VNB	
VPB	
VPWR	
X	
3 items	
X	
Y	
R90	
0 items	
171 items	
(0,0)	
(0,0), (2.76,2.72)	
(2.76, 2.72)	

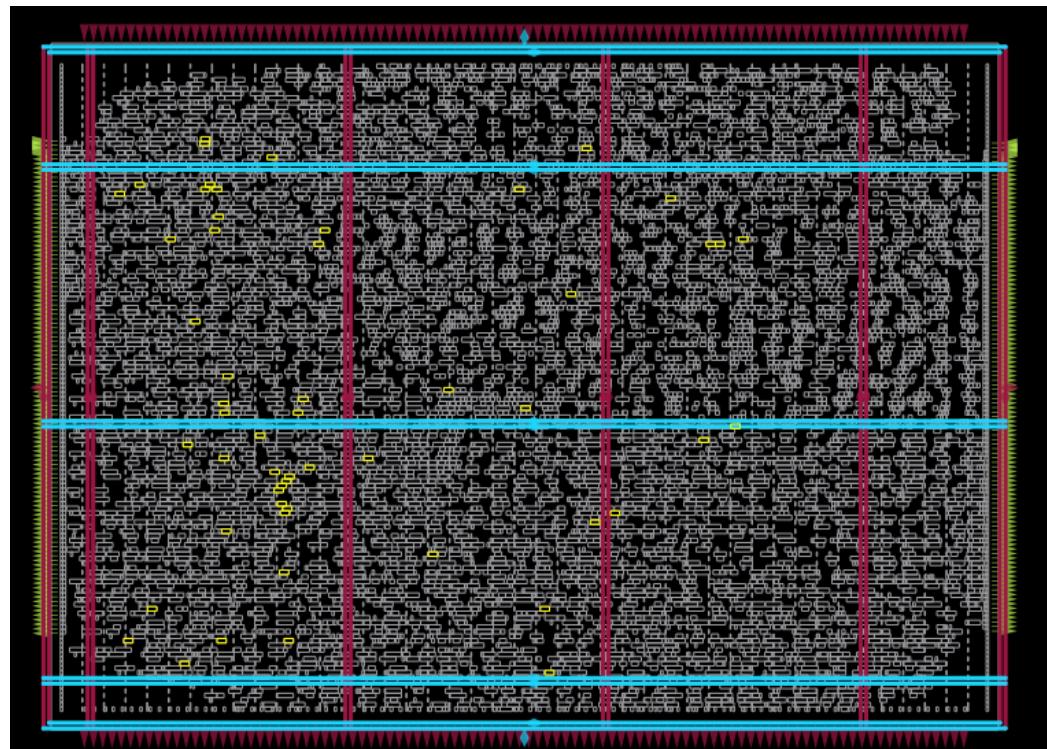


Nan	Value
Master	
sky130_fd_sc_hd_clkbuf_8	
CORE	
unithd	
6 items	
A	
VGND	
VNB	
VPB	
VPWR	
X	
3 items	
X	
Y	
R90	
0 items	
119 items	
(0,0)	
(0,0), (5.06,2.72)	
(5.06, 2.72)	





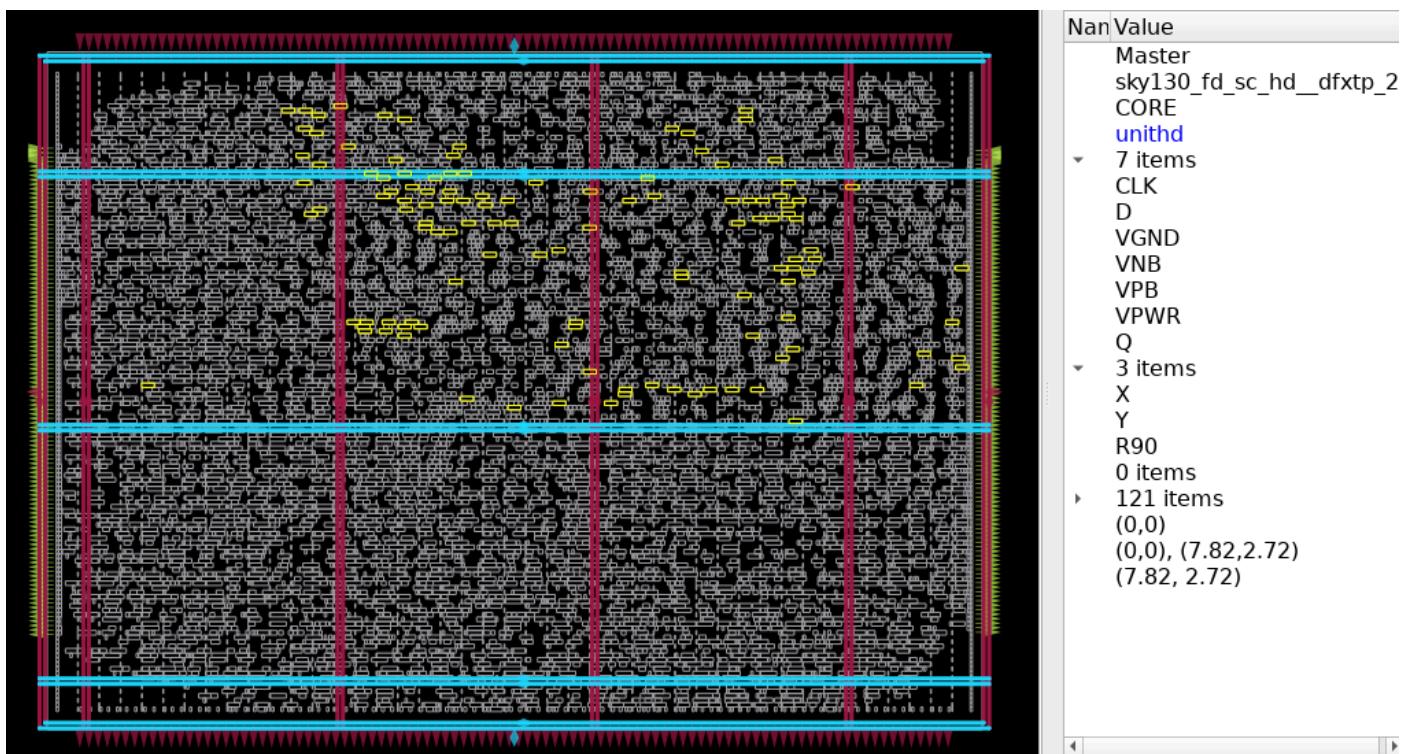
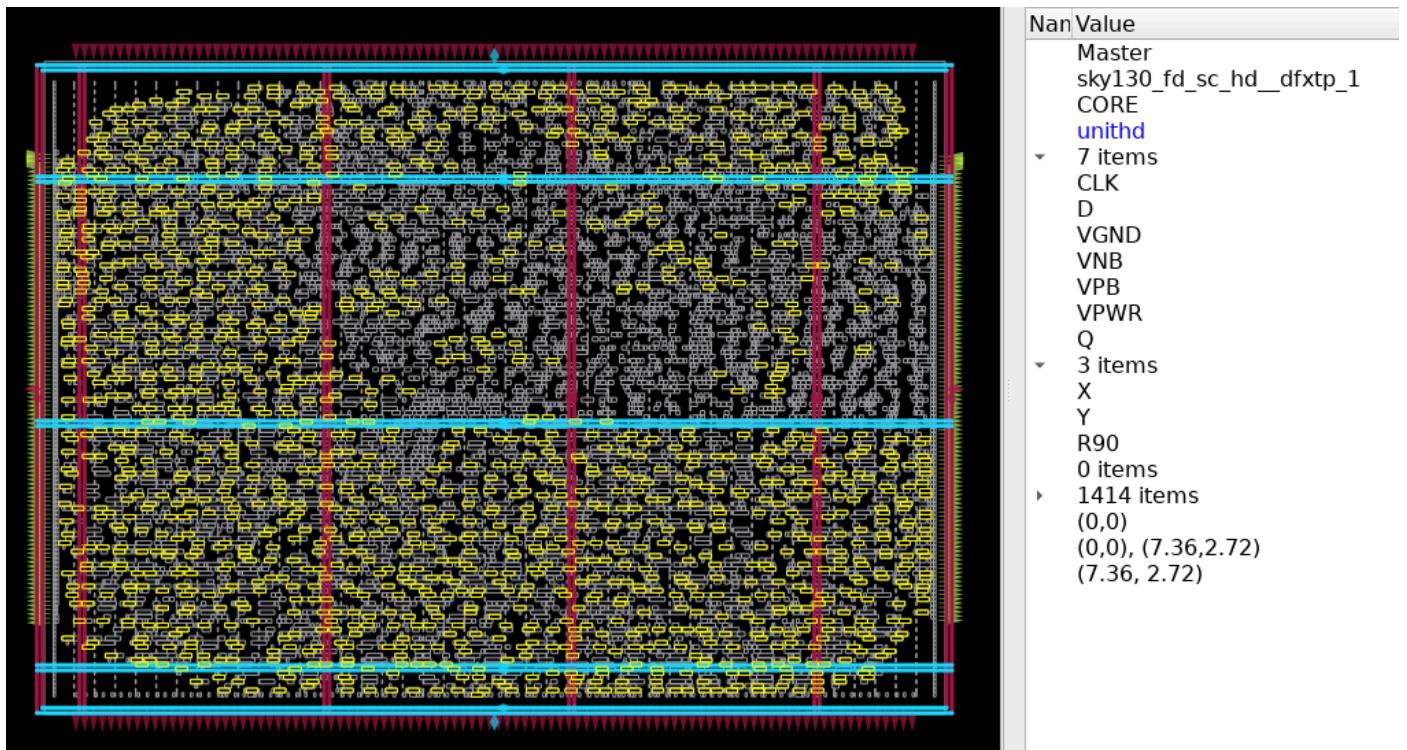
Nan	Value
Master	
sky130_fd_sc_hd_buf_6	
CORE	
unithd	
6 items	
A	
VGND	
VNB	
VPB	
VPWR	
X	
3 items	
X	
Y	
R90	
0 items	
90 items	
(0,0)	
(0,0), (4.14,2.72)	
(4.14, 2.72)	



Nan	Value
Master	
sky130_fd_sc_hd_buf_8	
CORE	
unithd	
6 items	
A	
VGND	
VNB	
VPB	
VPWR	
X	
3 items	
X	
Y	
R90	
0 items	
55 items	
(0,0)	
(0,0), (5.52,2.72)	
(5.52, 2.72)	



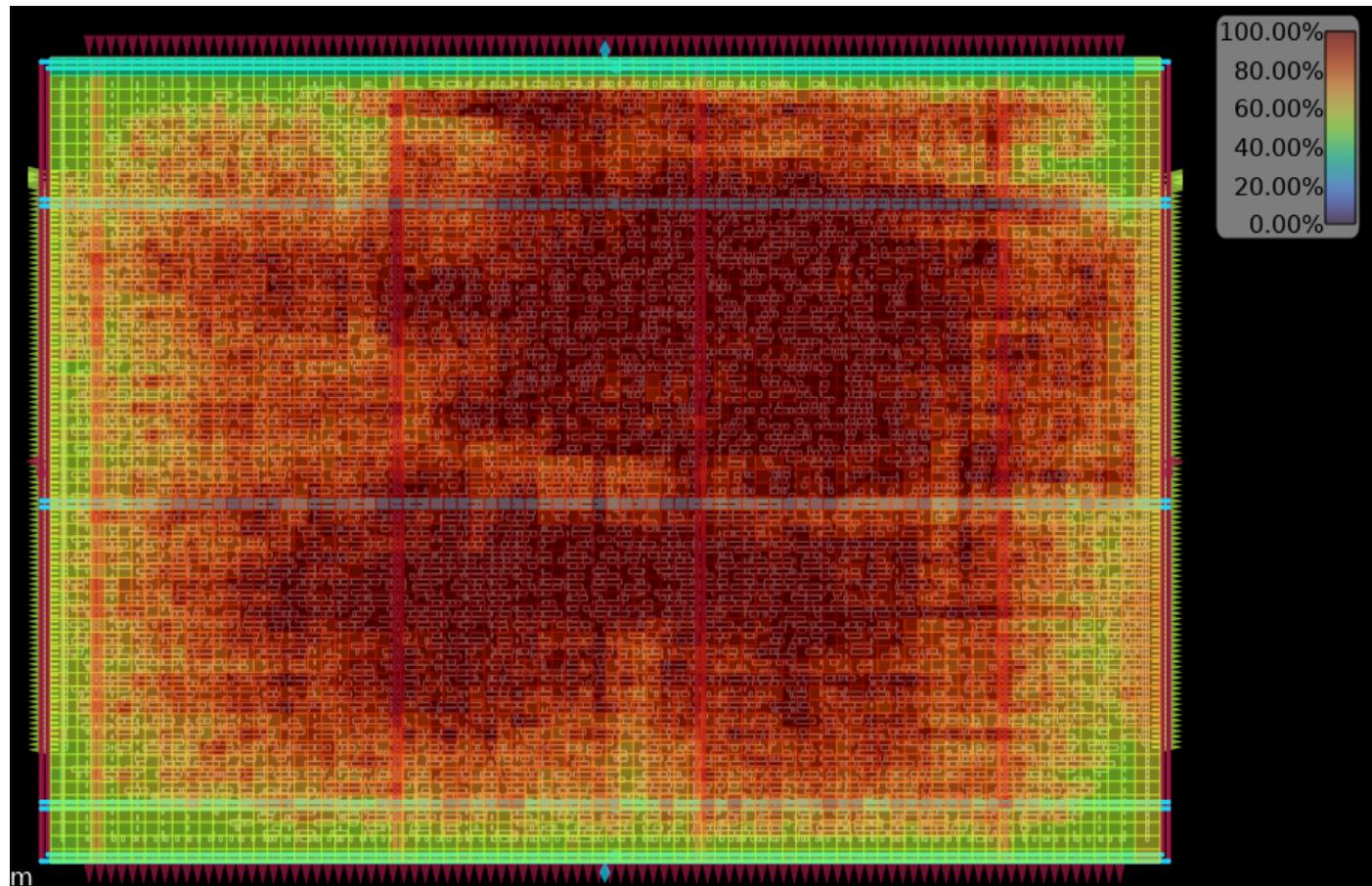
ii. Highlight the flops in the design





4. Congestion

- Review the congestion map
Estimated routing congestion at the placement stage



- Ensure no red zones (hotspots) in congestion plots
- Spread placement where necessary



Power reports

```
=====
report_power
=====
===== Typical Corner =====

Group          Internal Power   Switching Power   Leakage Power   Total Power (Watts)
-----
```

Sequential	6.92e-03	1.27e-03	1.37e-08	8.20e-03	50.3%
Combinational	3.37e-03	4.73e-03	2.49e-08	8.10e-03	49.7%
Clock	0.00e+00	0.00e+00	9.20e-10	9.20e-10	0.0%
Macro	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.0%
Pad	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.0%
Total	1.03e-02	6.00e-03	3.95e-08	1.63e-02	100.0%
	63.2%	36.8%	0.0%		

```
8-gpl_sta.power.rpt (END)
```

```
=====
report_power
=====
===== Typical Corner =====

Group          Internal Power   Switching Power   Leakage Power   Total Power (Watts)
-----
```

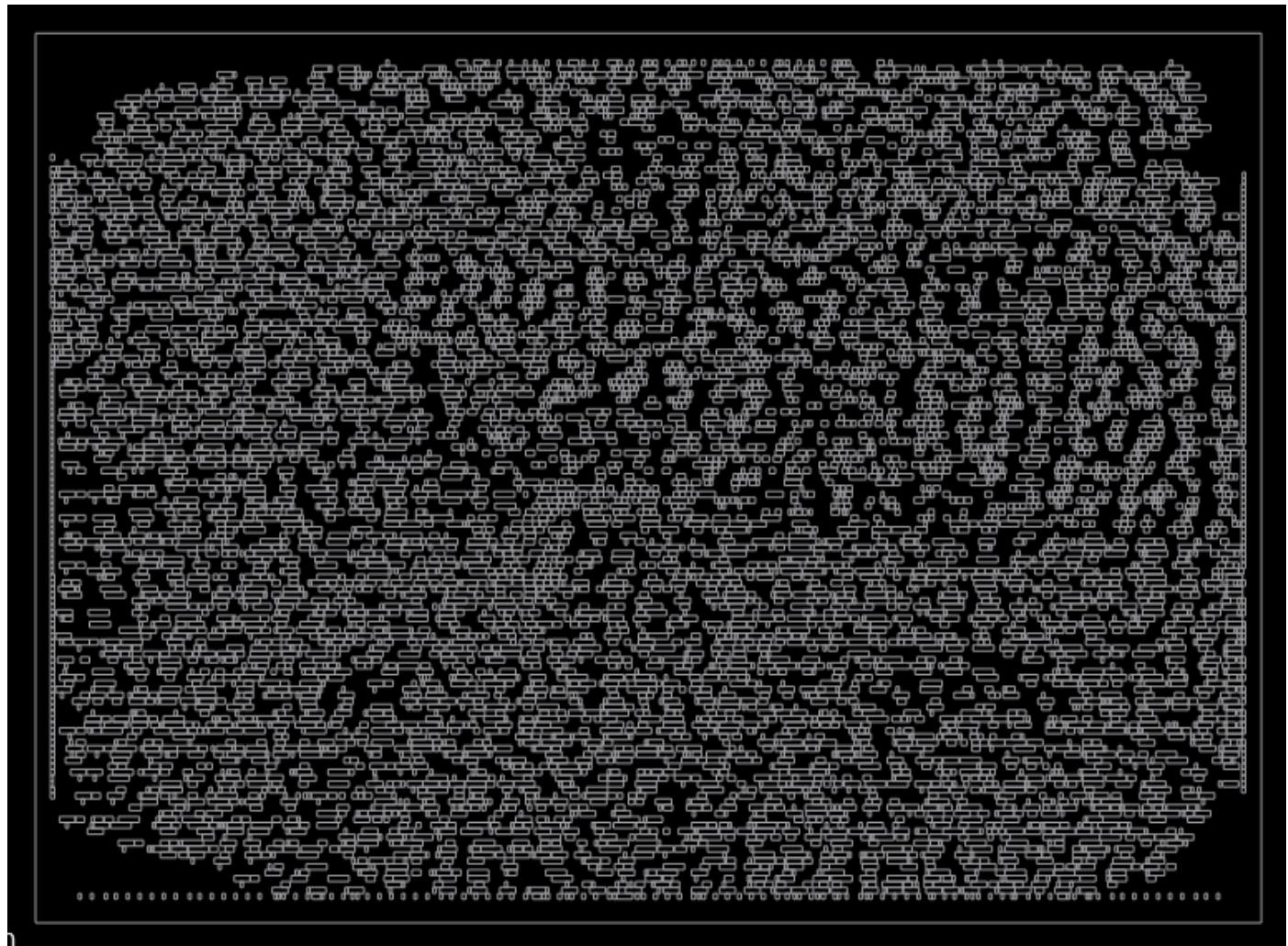
Sequential	6.93e-03	8.80e-04	1.30e-08	7.81e-03	47.9%
Combinational	3.62e-03	4.88e-03	3.47e-08	8.50e-03	52.1%
Clock	0.00e+00	0.00e+00	9.20e-10	9.20e-10	0.0%
Macro	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.0%
Pad	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.0%
Total	1.05e-02	5.76e-03	4.86e-08	1.63e-02	100.0%
	64.7%	35.3%	0.0%		

```
11-dpl_sta.power.rpt (END)
```



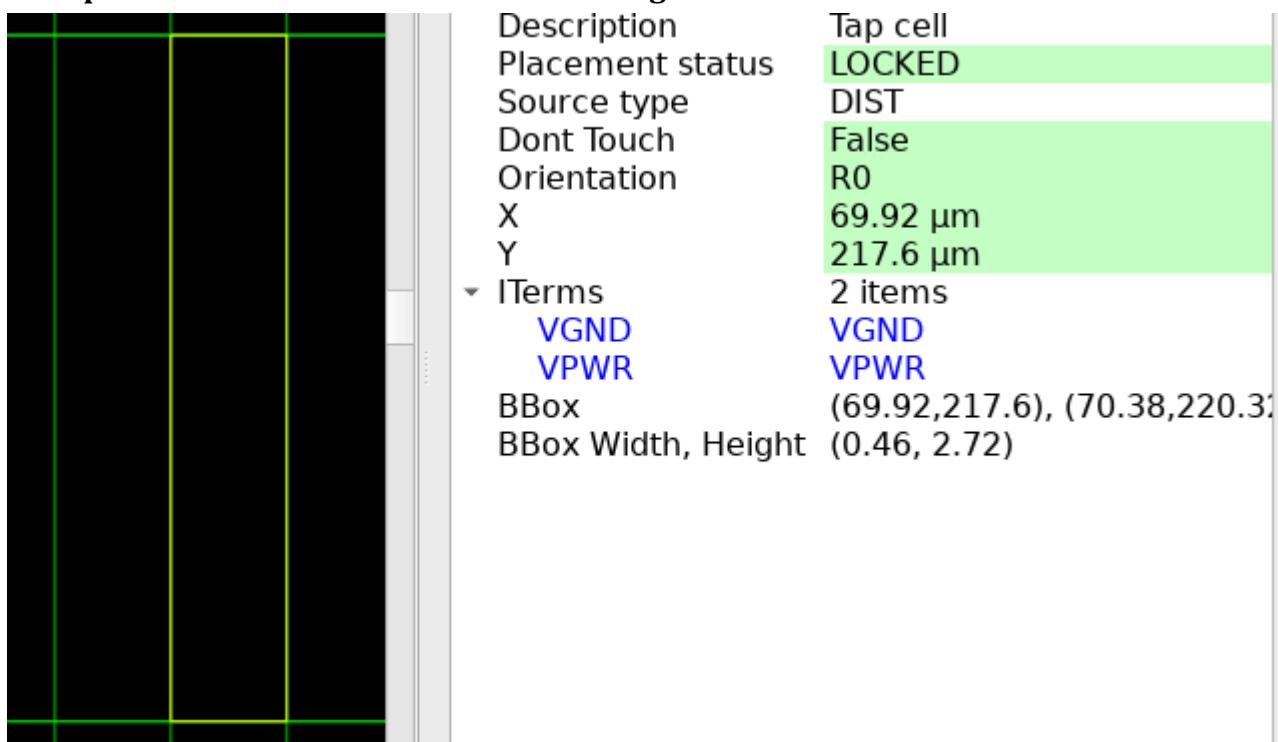
Output Files

- DEF file with placed standard cells



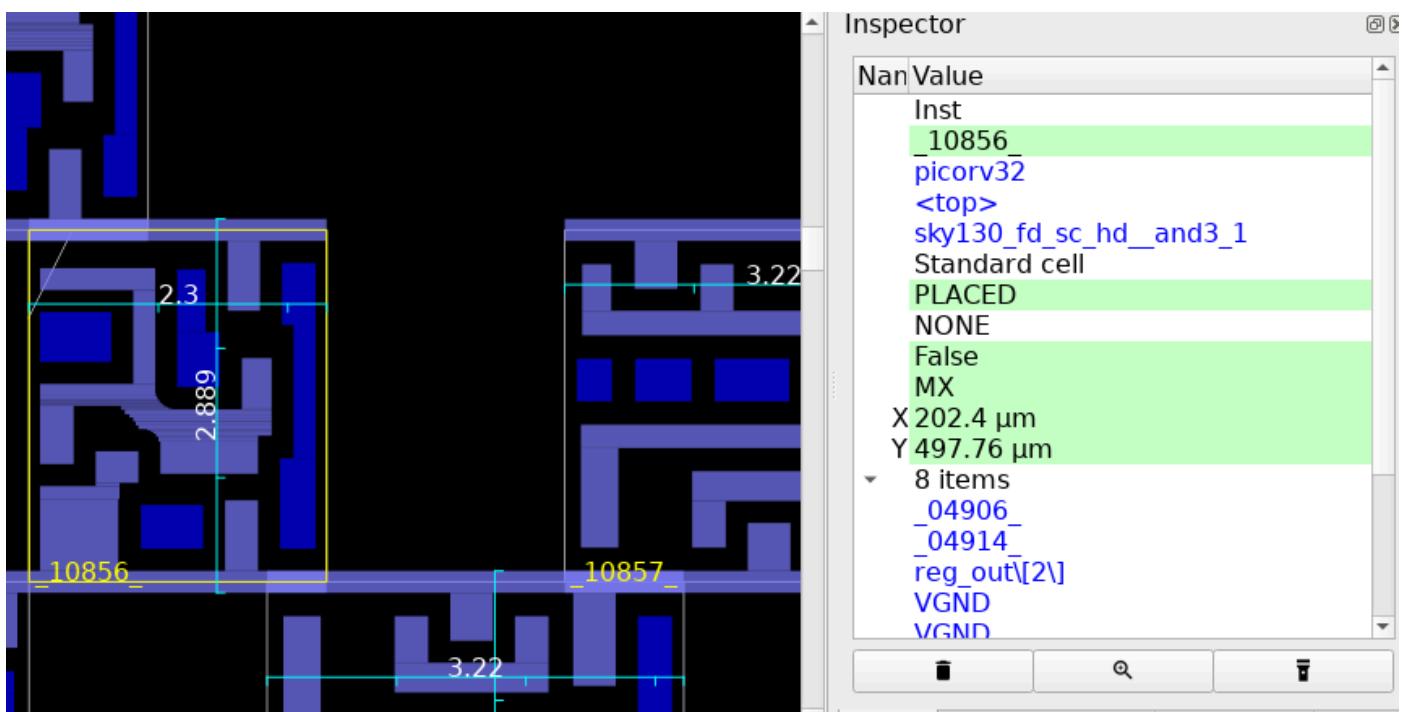
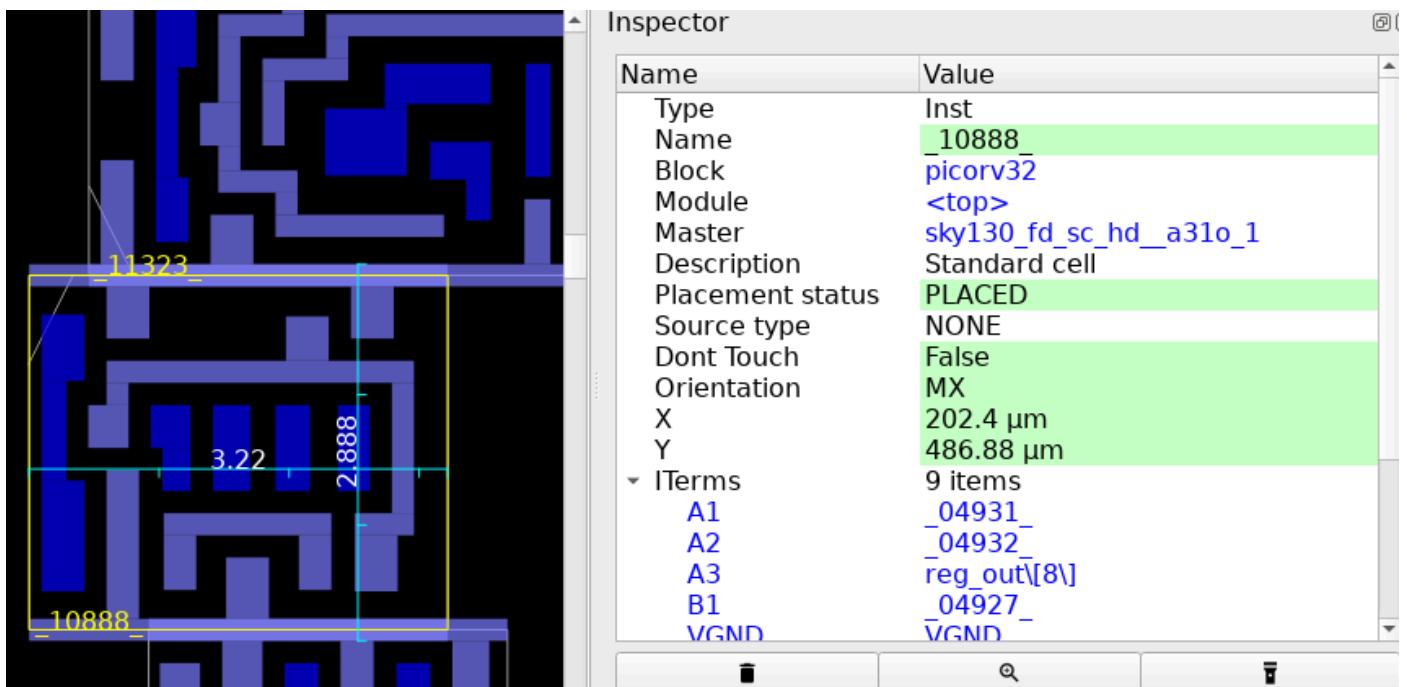


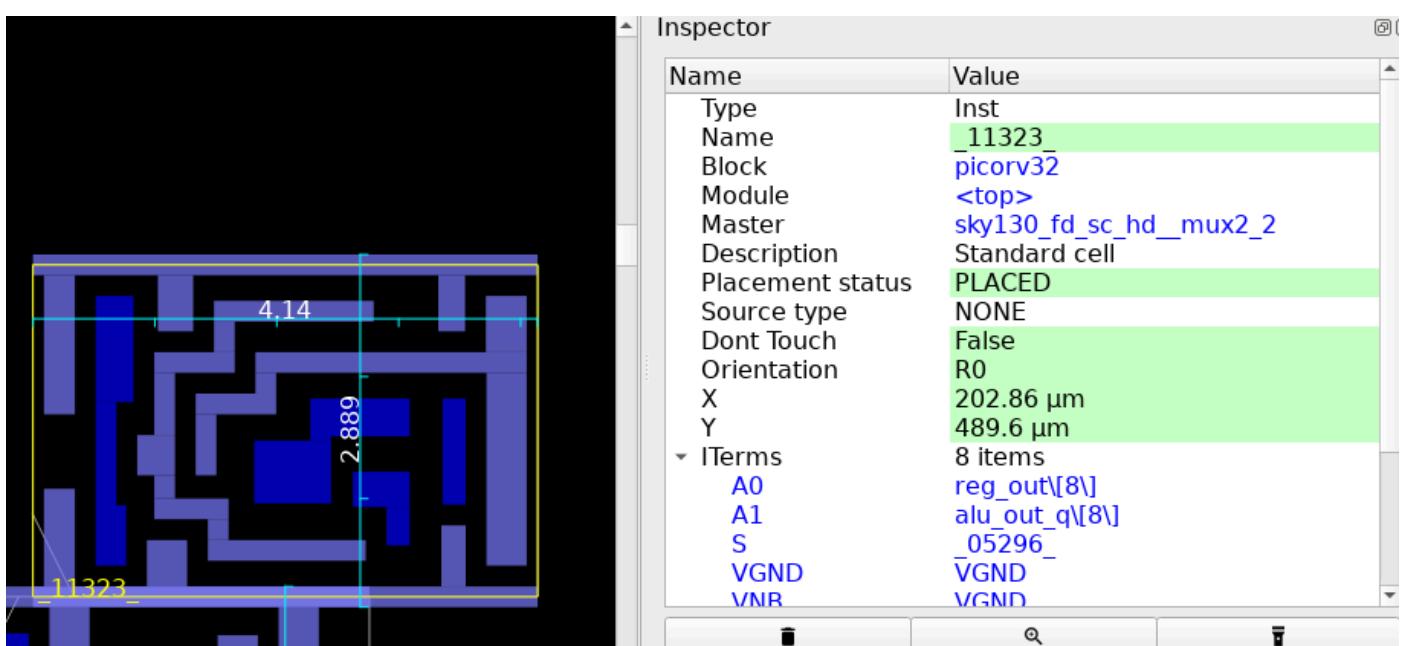
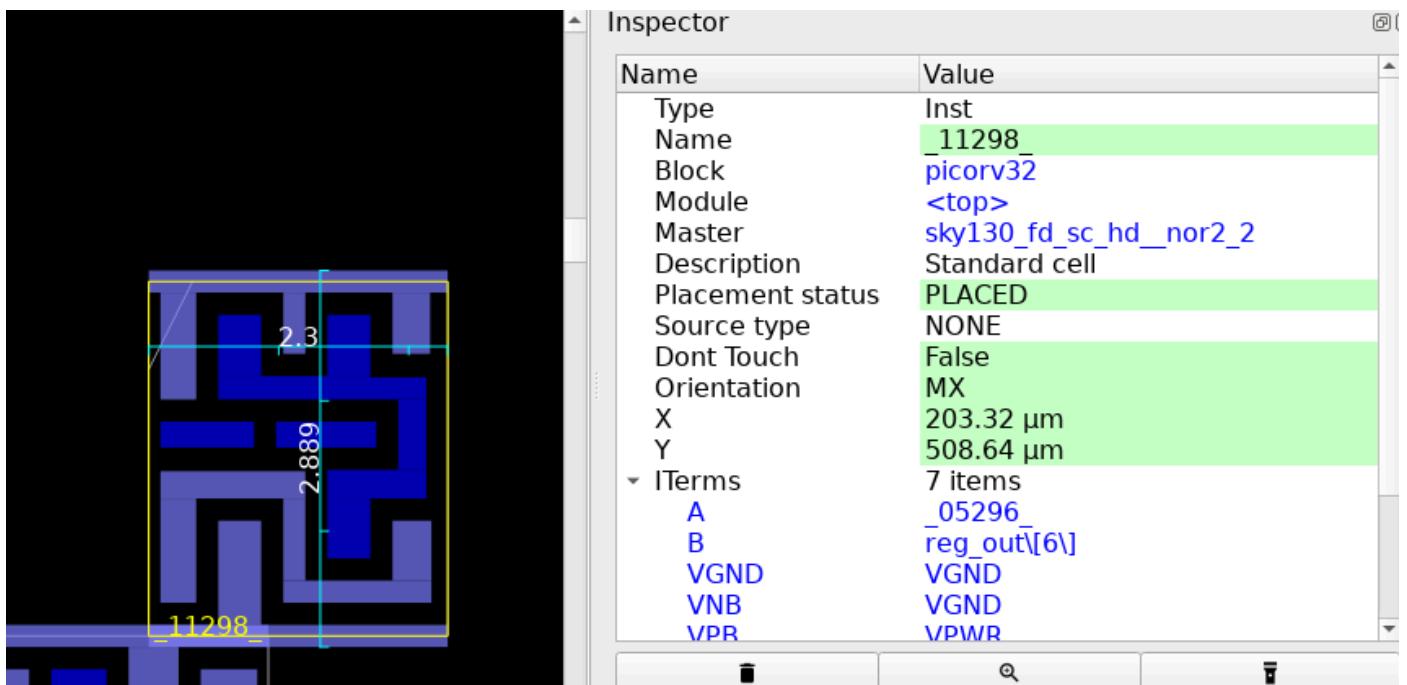
iii. Capture the site row and measure the height and width

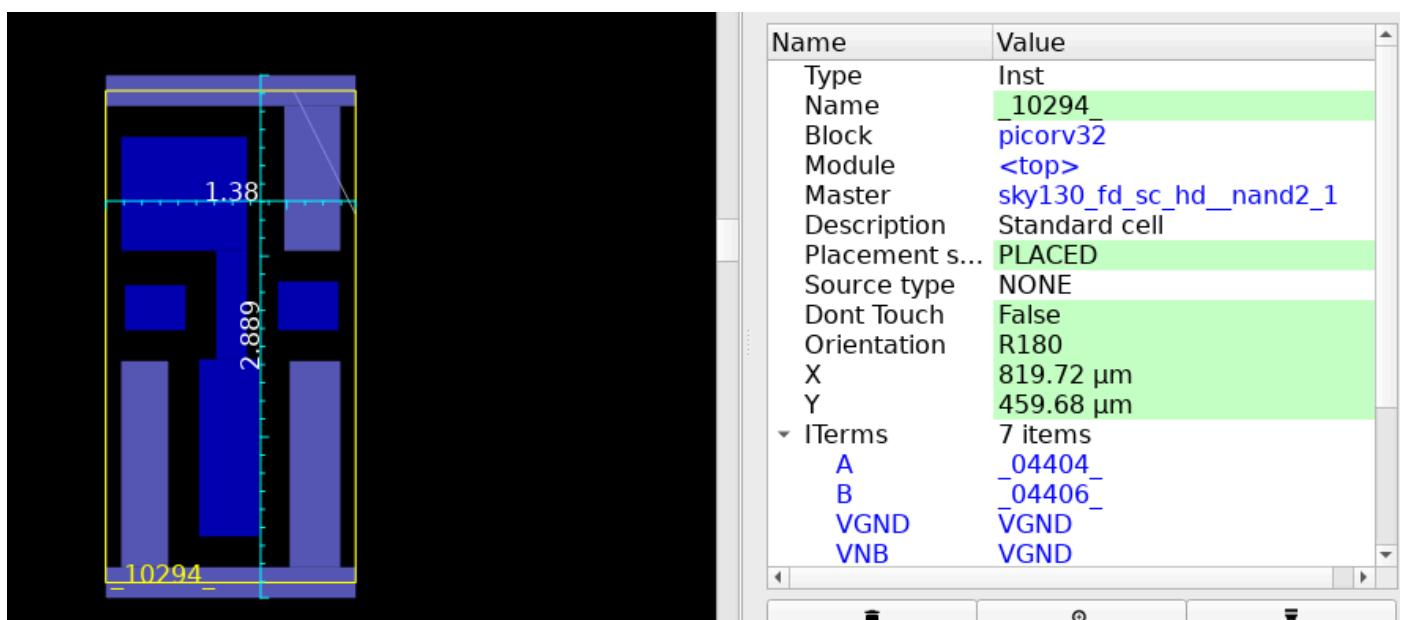
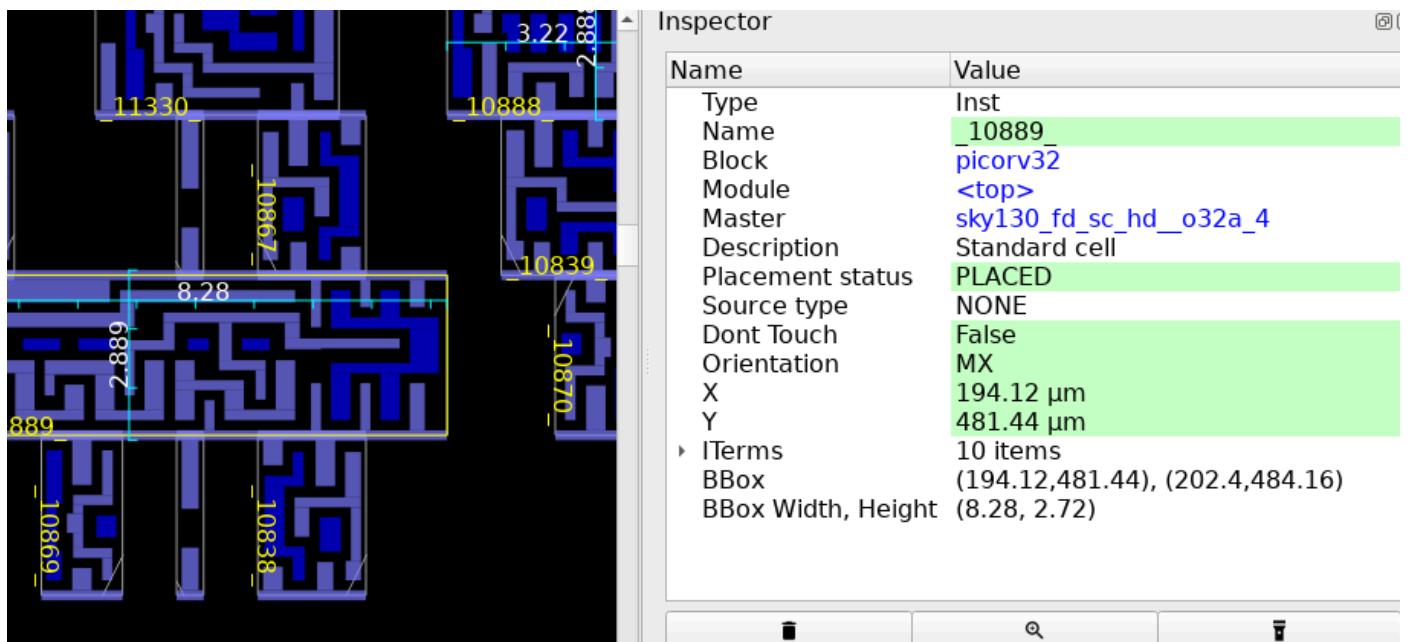


iv. Find the different library cell heights and widths



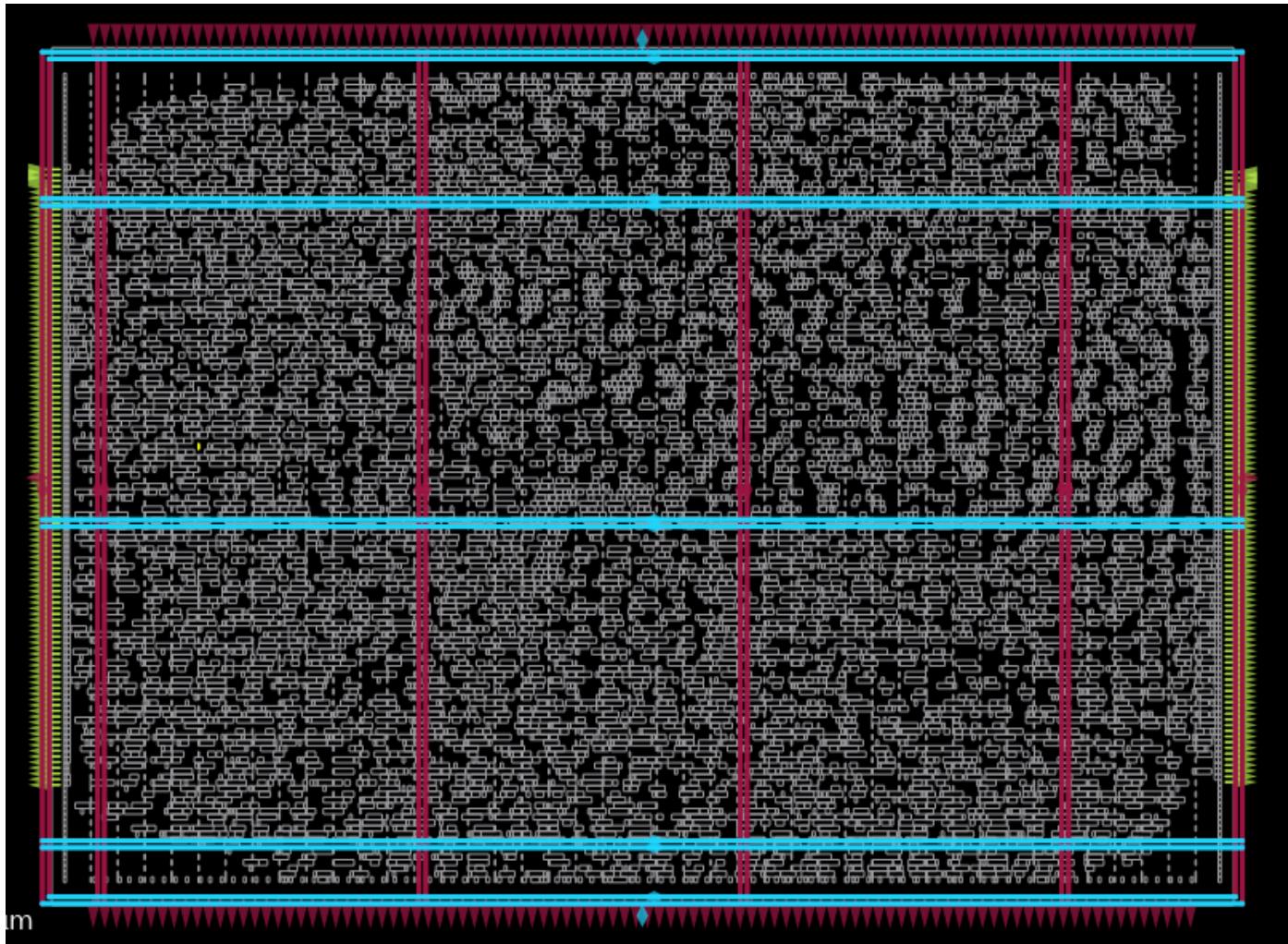








Design after placement:





Clock Tree Synthesis (CTS)

Clock Tree Synthesis is the process of distributing the clock signal from a single source (clock root/clock source) to all the sequential elements (like flip-flops) in a way that minimizes clock skew and ensures timing closure. It transforms the ideal clock path into a physically realized one, considering delays, buffers, and physical routing.

Inputs

- Placement DEF
- Logical netlist
- Clock constraints (target skew, uncertainty)
- Library and technology files
- Configurations for skew, fanout, max transition, and metal layers

Steps Involved in CTS

1. Guideline Setup
 - Skew Target: We set skew target as 5% of the clock period.(This parameter is removed from openlane)
 - Metal Layer: Typically, higher metal layers are used for better performance.
 - Max Transition/Fanout: Default values used to avoid excessive loading and delay.
 - Uncertainty: Set at 5% post-CTS.



```
proc run_cts_step {args} {
    if { ! [ info exists ::env(CTS_CURRENT_DEF) ] } {
        set ::env(CTS_CURRENT_DEF) $::env(CURRENT_DEF)
    } else {
        set ::env(CURRENT_DEF) $::env(CTS_CURRENT_DEF)
    }

    run_cts
    puts "Setting Uncertainty to 5% of clock period"
    set ::env(SYNTH_CLOCK_UNCERTAINTY) [expr 0.05 * $::env(CLOCK_PERIOD)]
    puts "Uncertainty successfully set to 5% of clock period (POST CTS)"
    run_resizer_timing
}
```

```
[STEP 12]
[INFO]: Running Clock Tree Synthesis (log: designs/picorv32/runs/finalfix/logs/cts/12-cts.log)...
[STEP 13]
[INFO]: Running Single-Corner Static Timing Analysis (log: designs/picorv32/runs/finalfix/logs/cts/13-cts_sta.log)...
Setting Uncertainty to 5% of clock period
Uncertainty successfully set to 5% of clock period (POST CTS)
[STEP 14]
[INFO]: Running Placement Resizer Timing Optimizations (log: designs/picorv32/runs/finalfix/logs/cts/14-resizer.log)...
[STEP 15]
```

At cts stage



Post CTS :

2. Clock Tree Construction

- Clock buffers and inverters are inserted to drive the clock signal.
- Tree is built hierarchically for balanced paths.

3. Clock Optimization

- Removing excessive skew.
- Optimizing for power and delay.
- Balancing latency from source clock to sink.

Quality Checks

1. Timing (Setup/Hold Analysis)

- Report timing post-CTS to check for violations.
- Ensure all clock paths meet the setup and hold requirements.

2. Clock Network Delay

- Check the total delay from clock root to clock sinks.
- Clock latency (Startpoint to Endpoint) is analyzed.

3. Skew Check

- Skew should be within the target limit (5% of clock period)(removed from Openlane).
- Pick a timing path and subtract latency between start and endpoint.

4. Power

- Compare clock tree power with pre-CTS (placement stage).
- Check the percentage increase in power consumption.



5. Design Rule Compliance

- Ensure no DRC violations are introduced.
- Legal placement of buffers and no congestion hotspots.

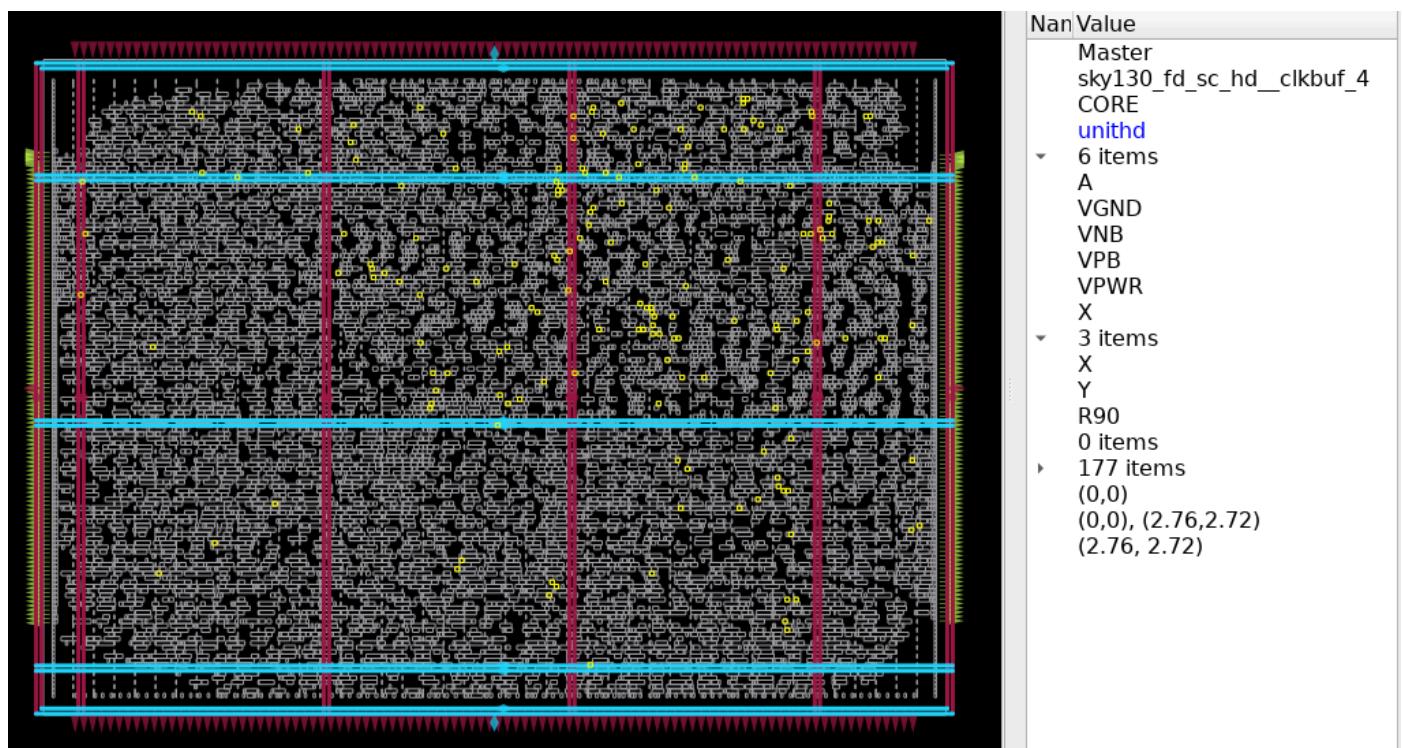
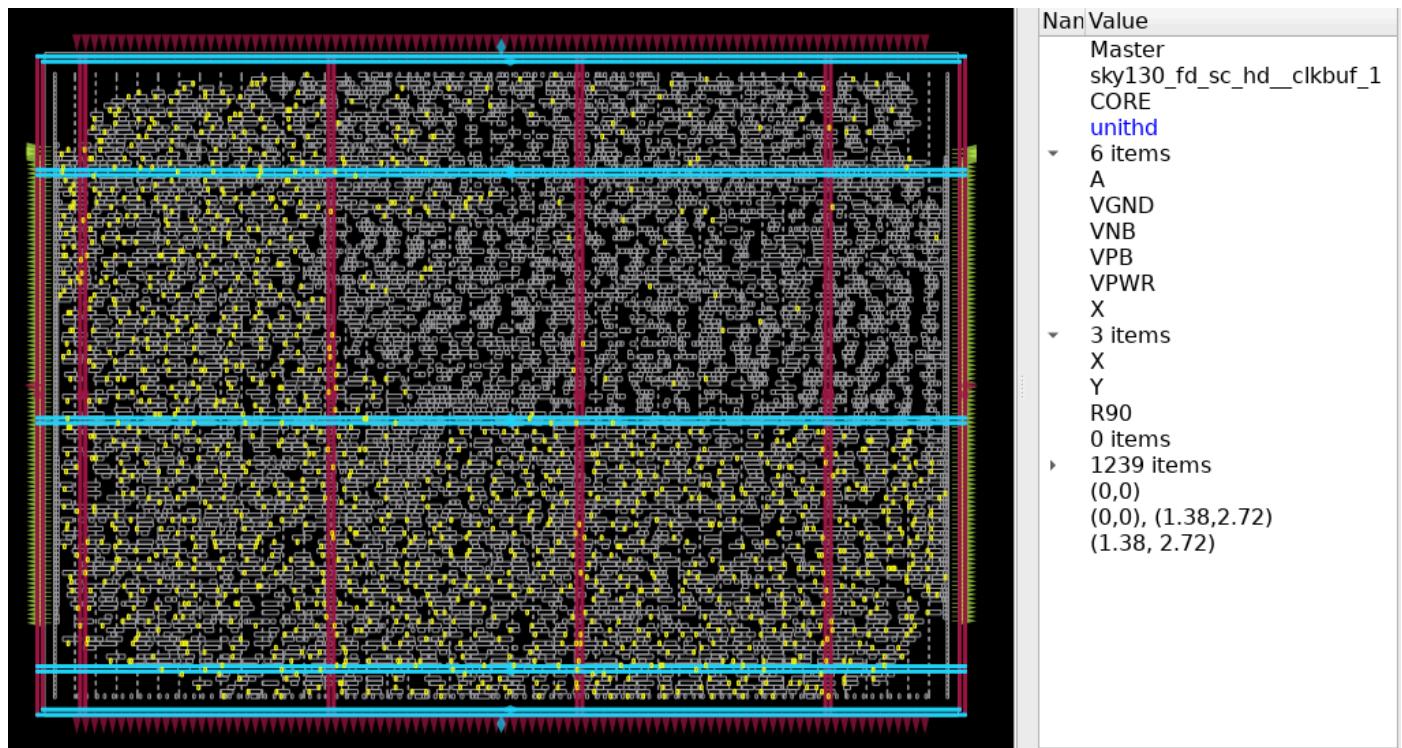
Input output delay from picorv32.sdc(50% of clock period)

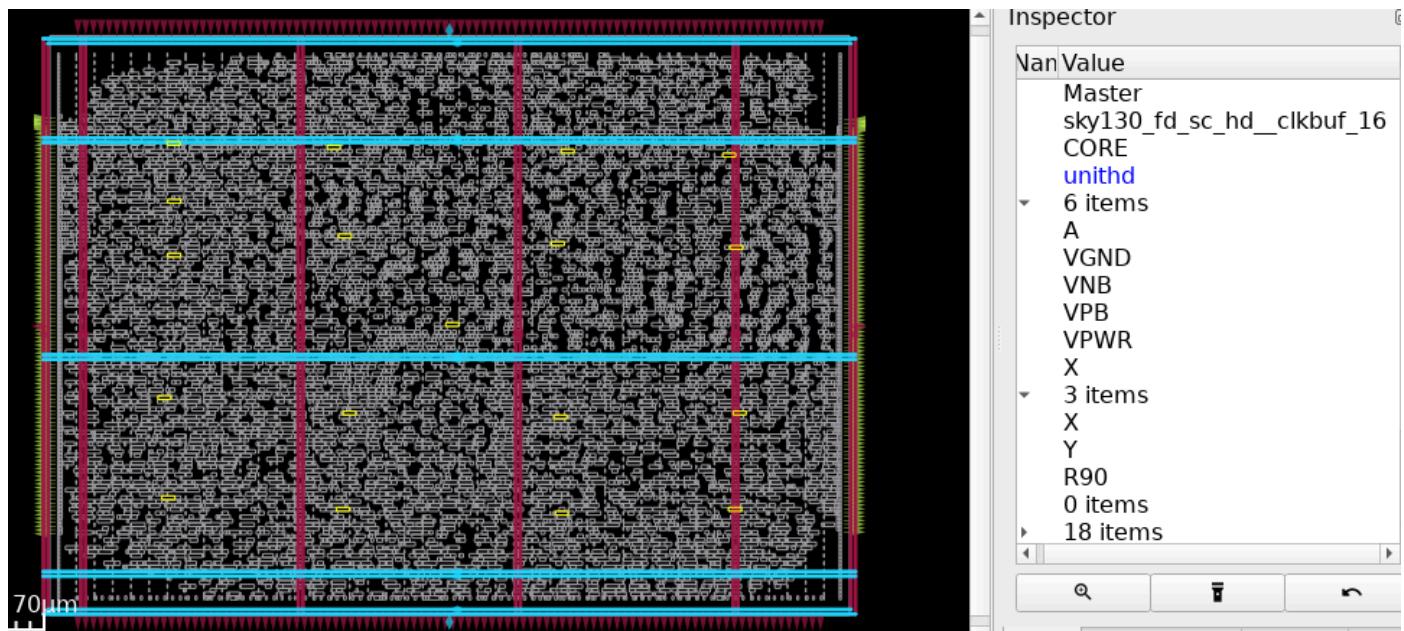
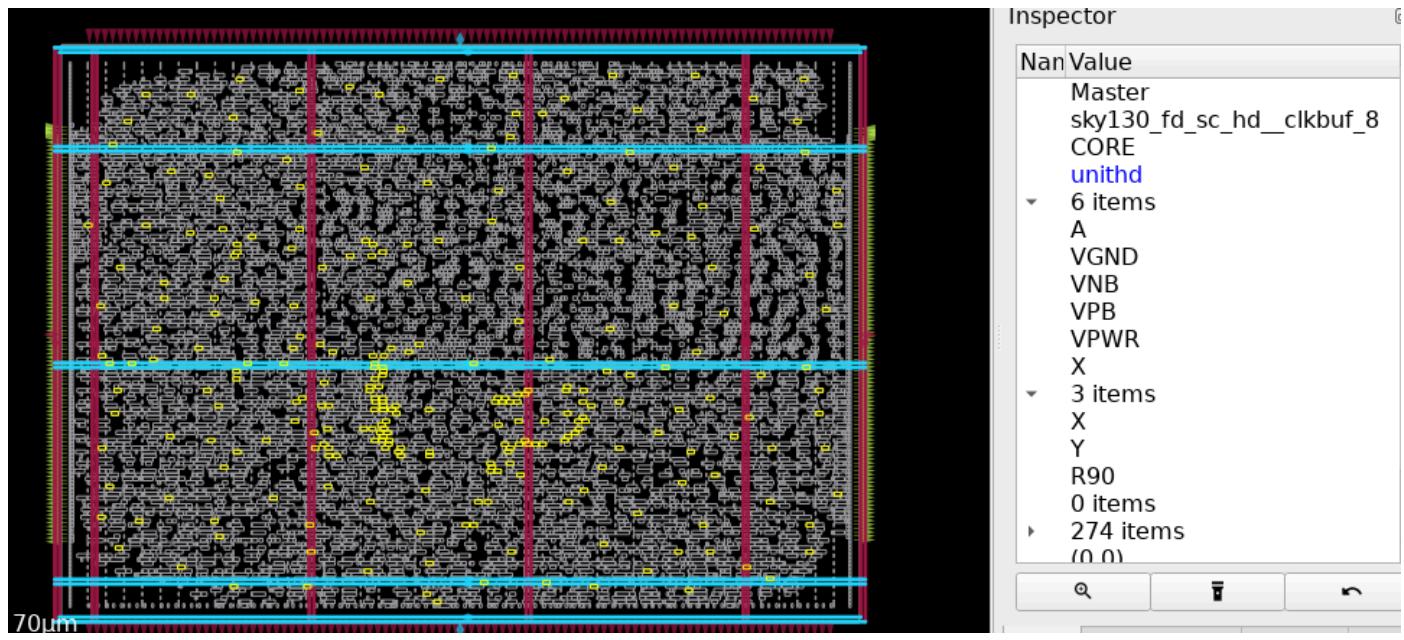
```
set_clock_uncertainty 1.5000 clk
set_propagated_clock [get_clocks {clk}]
set_input_delay 5.0000 -clock [get_clocks {clk}] -add_delay [get_ports {irq[0]}]
set_input_delay 5.0000 -clock [get_clocks {clk}] -add_delay [get_ports {irq[10]}]
set_input_delay 5.0000 -clock [get_clocks {clk}] -add_delay [get_ports {irq[11]}]
set_input_delay 5.0000 -clock [get_clocks {clk}] -add_delay [get_ports {irq[12]}]
set_input_delay 5.0000 -clock [get_clocks {clk}] -add_delay [get_ports {irq[13]}]
set_input_delay 5.0000 -clock [get_clocks {clk}] -add_delay [get_ports {irq[14]}]
set_input_delay 5.0000 -clock [get_clocks {clk}] -add_delay [get_ports {irq[15]}]
set_input_delay 5.0000 -clock [get_clocks {clk}] -add_delay [get_ports {irq[16]}]
set_input_delay 5.0000 -clock [get_clocks {clk}] -add_delay [get_ports {irq[17]}]
set_input_delay 5.0000 -clock [get_clocks {clk}] -add_delay [get_ports {irq[18]}]
set_input_delay 5.0000 -clock [get_clocks {clk}] -add_delay [get_ports {irq[19]}]
```

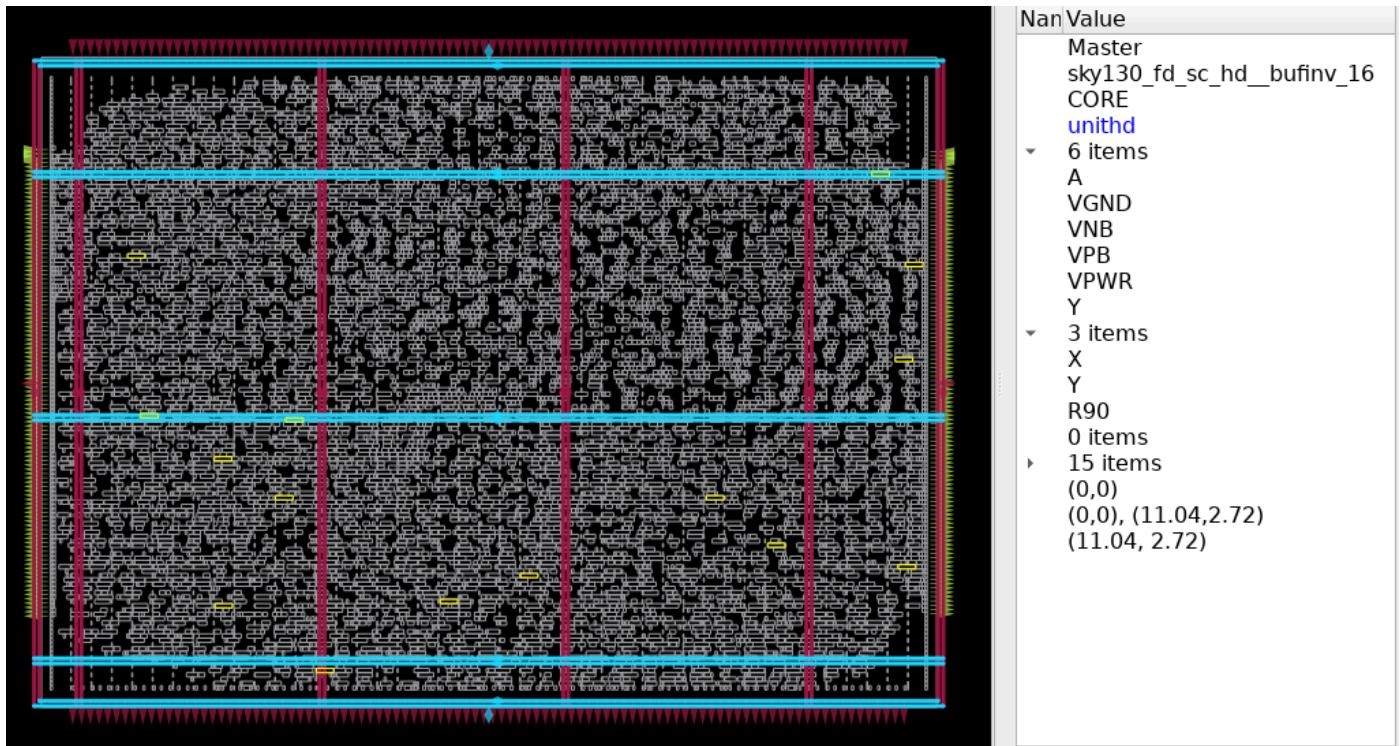
6. Utilization

- Must stay below 70% even after CTS buffer insertion.

```
QSurfaceFormat::NoProfile)
>>> report_design_area
Design area 0 u^2 49% utilization.
```







- Prevents overcrowding and ensures routability.

Outputs

- Updated netlist with CTS cells (buffers/inverters)
- CTS reports (skew, clock tree power, latency)

```
=====
report_clock_skew
=====

Clock clk
  1.02 source latency _19053_CLK ^
  -0.78 target latency _18651_CLK ^
  1.50 clock uncertainty
  -0.02 CRPR
-----
  1.72 setup skew

13-cts_sta.skew.rpt (END)
```



```
=====
report_power
=====
===== Typical Corner =====

Group          Internal Power   Switching Power   Leakage Power   Total Power (Watts)
-----
Sequential      7.01e-03    7.86e-04    1.30e-08    7.80e-03  26.7%
Combinational   5.00e-03    8.10e-03    4.24e-08    1.31e-02  44.8%
Clock           5.20e-03    3.11e-03    5.36e-09    8.31e-03  28.5%
Macro            0.00e+00    0.00e+00    0.00e+00    0.00e+00  0.0%
Pad              0.00e+00    0.00e+00    0.00e+00    0.00e+00  0.0%

-----
Total           1.72e-02    1.20e-02    6.07e-08    2.92e-02  100.0%
      58.9%        41.1%        0.0%
```

13-cts_sta.power.rpt (END)

```
=====
report_tns
=====
tns -1.76

=====
report_wns
=====
wns -1.27

=====
report_worst_slack -max (Setup)
=====
worst slack -1.27

=====
report_worst_slack -min (Hold)
=====
worst slack -1.05
```

13-cts_sta.summary.rpt (END)



- DEF file with clock tree placement
- Timing analysis reports post-CTS
- Clock distribution statistics

Clock Tree

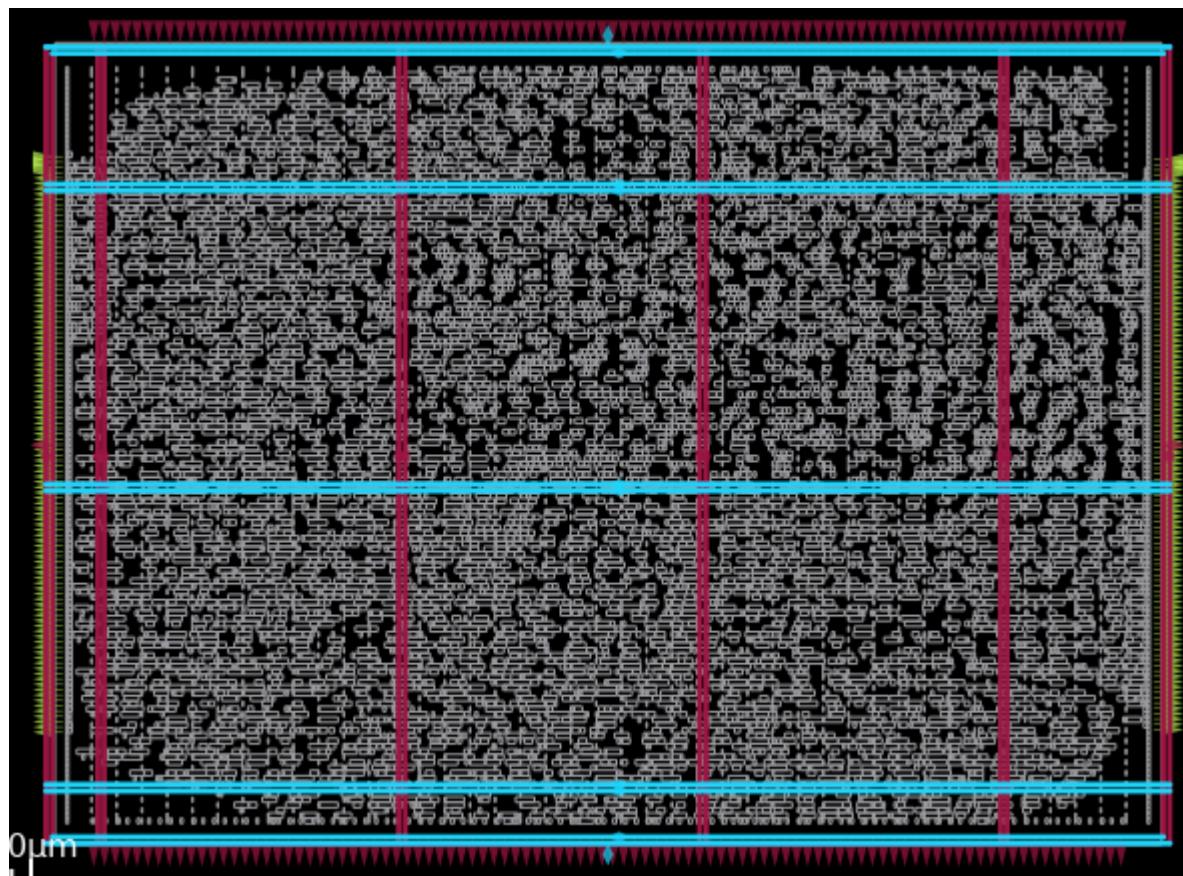
Highlight the clock routing and its metal layer usage



Here, metals 1,2 and 4 are used for clock wires.



CTS Output .odb file.





Routing

Routing is the process of connecting all logical nets in the design using metal layers, based on the placement of standard cells and macros. It ensures signal integrity, meets timing requirements, and adheres to all design rules. Routing typically follows placement and clock tree synthesis.

Inputs:

- DEF file after CTS
- Standard cell LEF
- Technology LEF (for metal rules)
- Timing constraints (.sdc)
- Placement information (coordinates of pins, macros, cells)

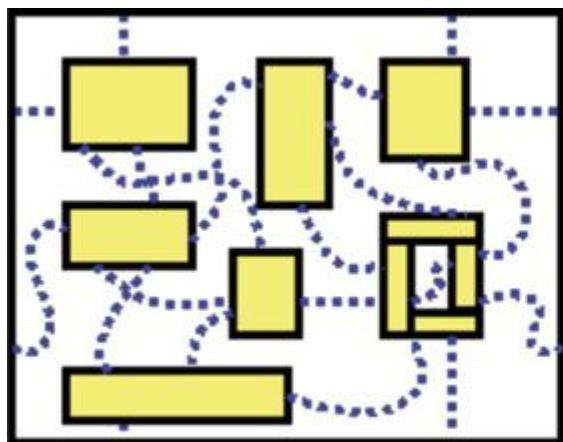
Steps Involved in Routing

1. Global Routing

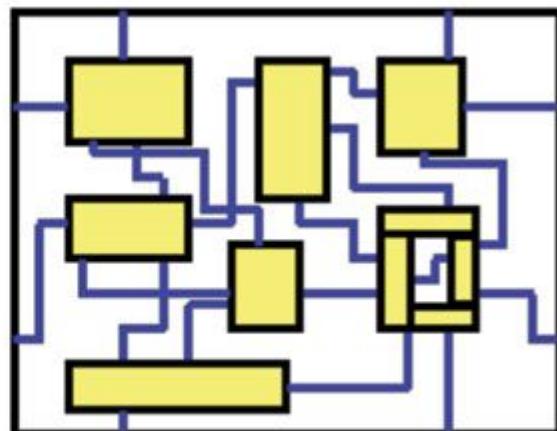
Global routing creates an approximate path for all nets by dividing the chip into a grid and estimating routing resources. It does not route nets fully but estimates wirelength and congestion based on placement and routing rules. The goal is to plan routes without causing congestion.

2. Detailed Routing

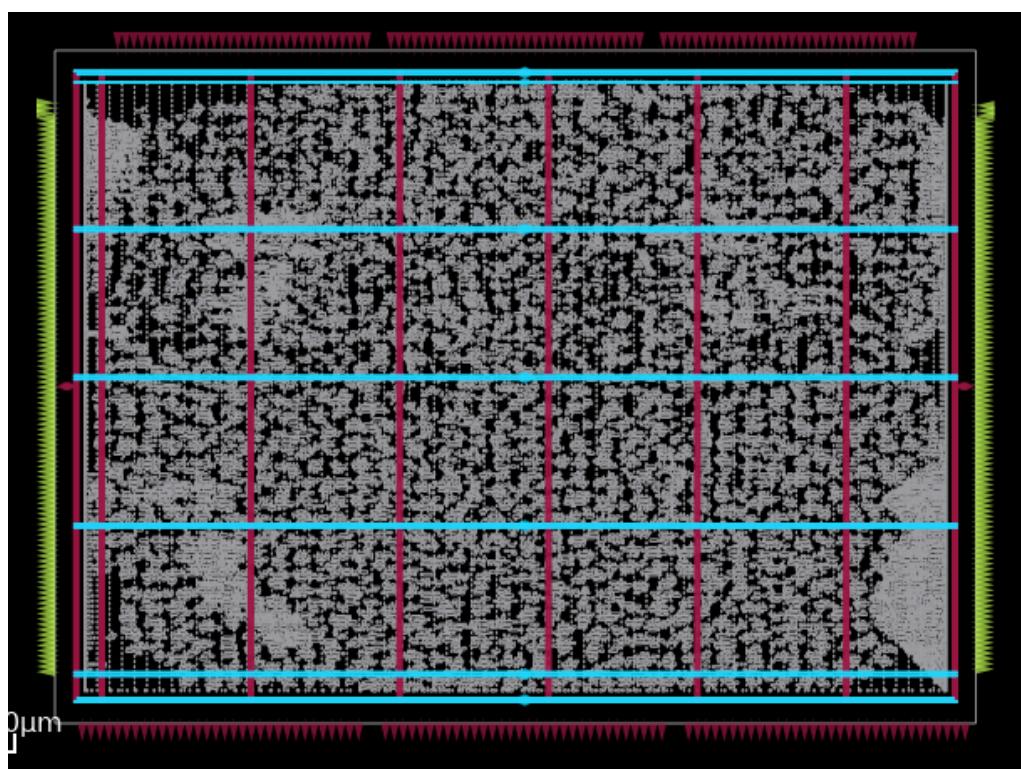
Detailed routing performs actual routing of nets with exact geometries and pin-to-pin connections. It adheres strictly to DRC rules for spacing, width, via usage, and layer assignments. It handles complex connections using multiple layers and routing tracks.



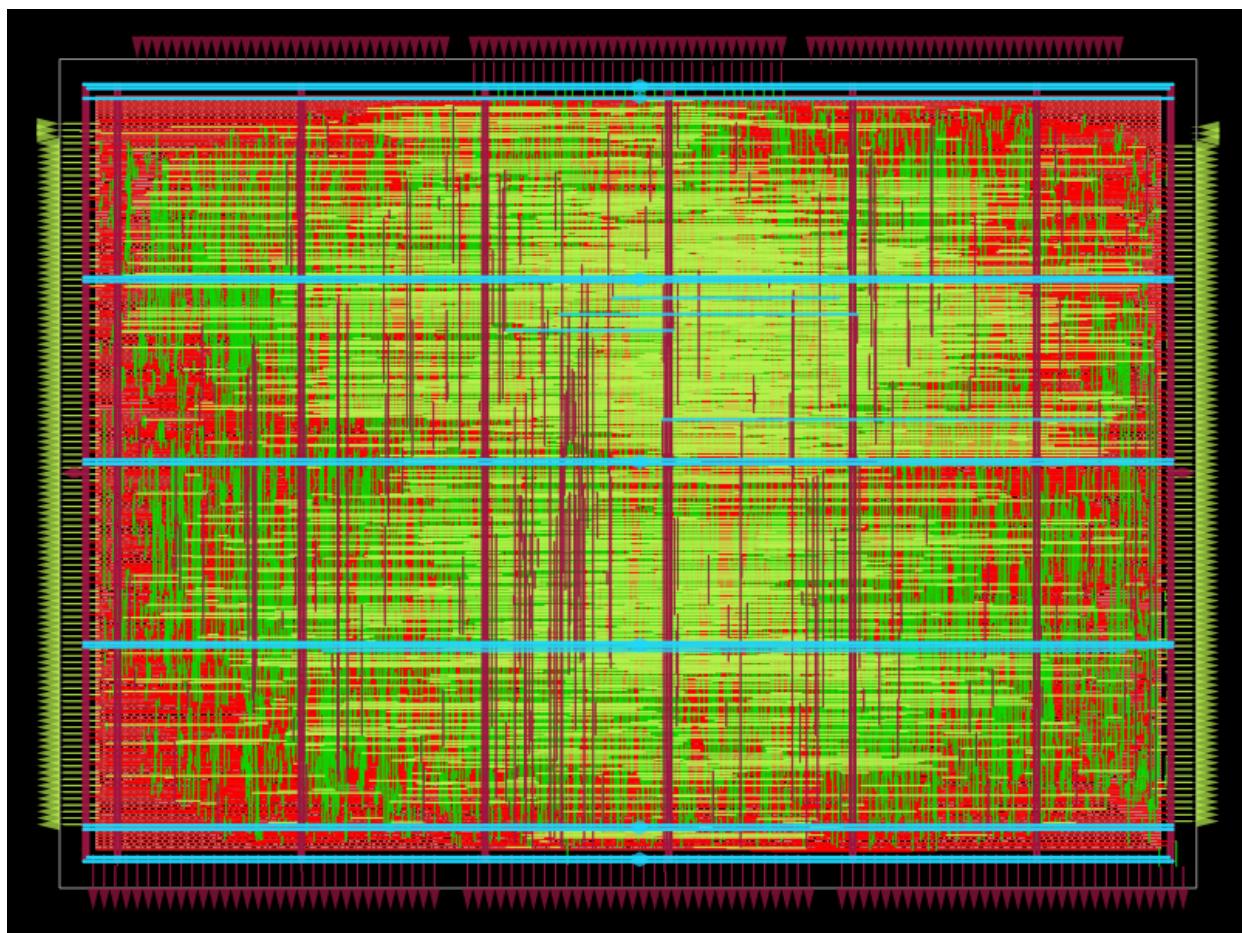
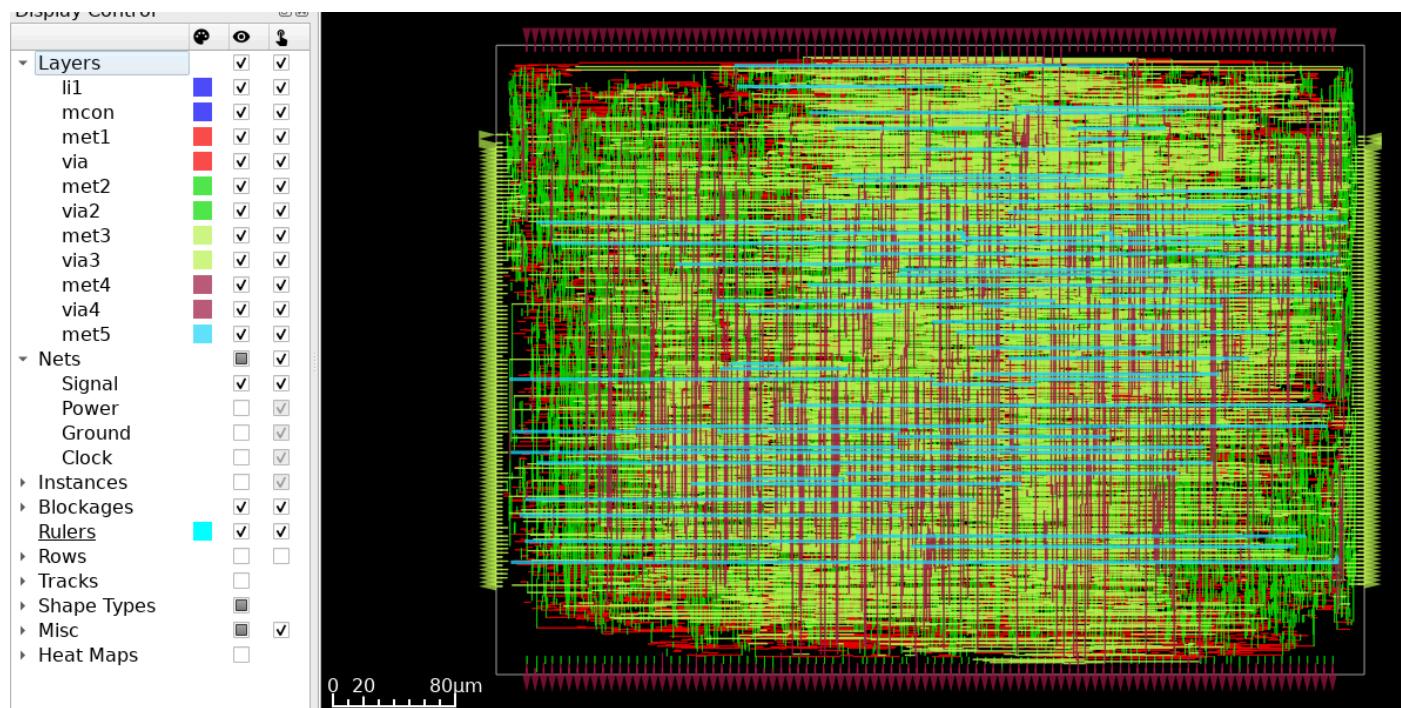
(a) Global routing



(b) Detailed routing



i. Highlight the signal routing and its metal layers





3. DRC Fixing

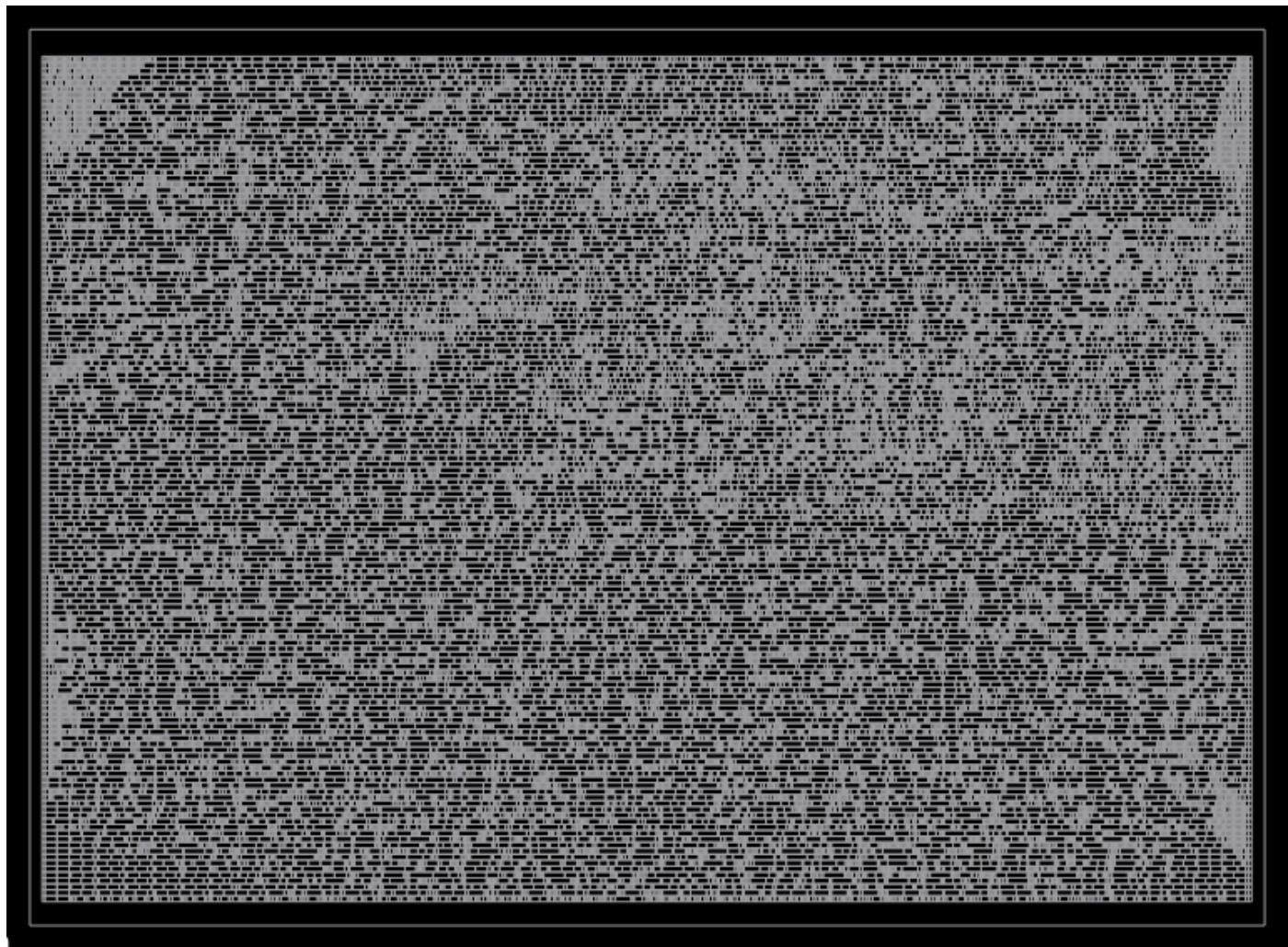
Post-routing, design rule violations such as spacing, via cuts, and wire width issues are identified and resolved either manually or using automatic repair features.

4. Timing Optimization

To meet timing requirements, the routed design is analyzed and optimized. Buffers may be inserted, and critical paths may be rerouted to fix setup or hold violations.

5. Filler Cell Insertion

After routing is complete, filler cells are inserted to fill the empty spaces between cells. This helps in maintaining well tap continuity and avoids open metal segments which may lead to DRC violations.

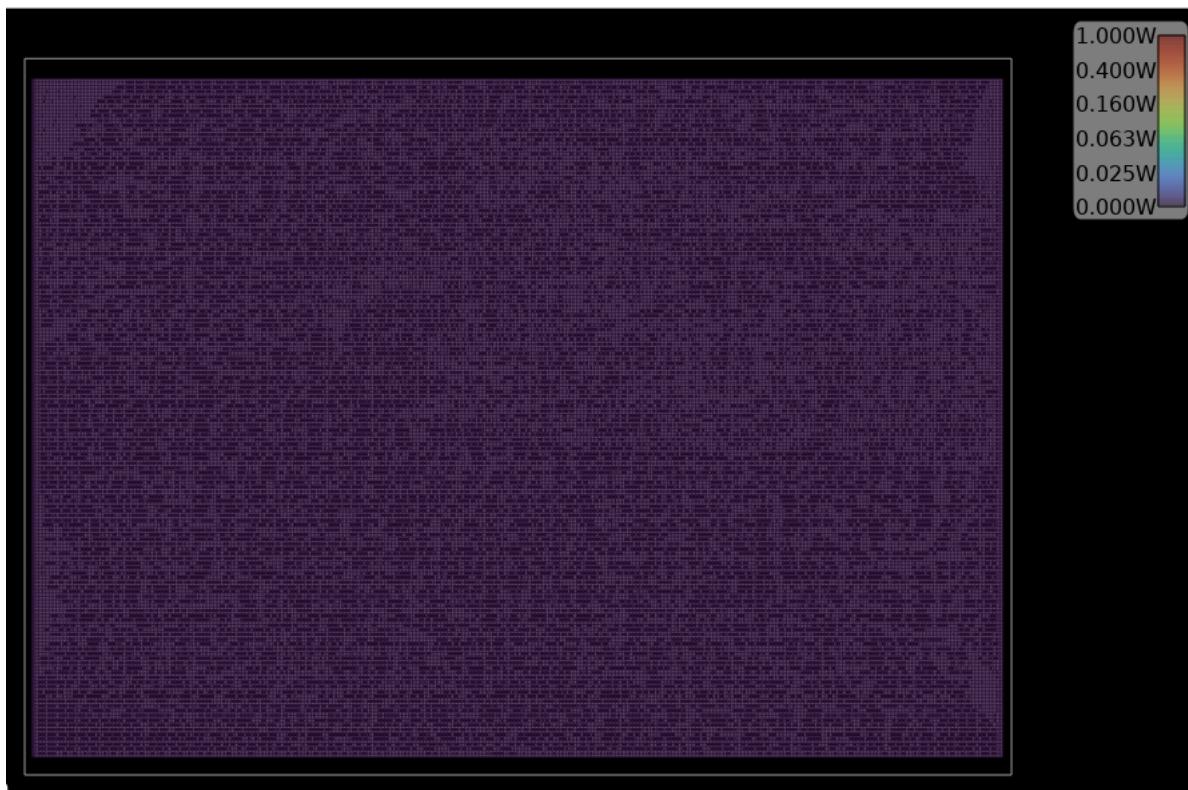




Quality Checks

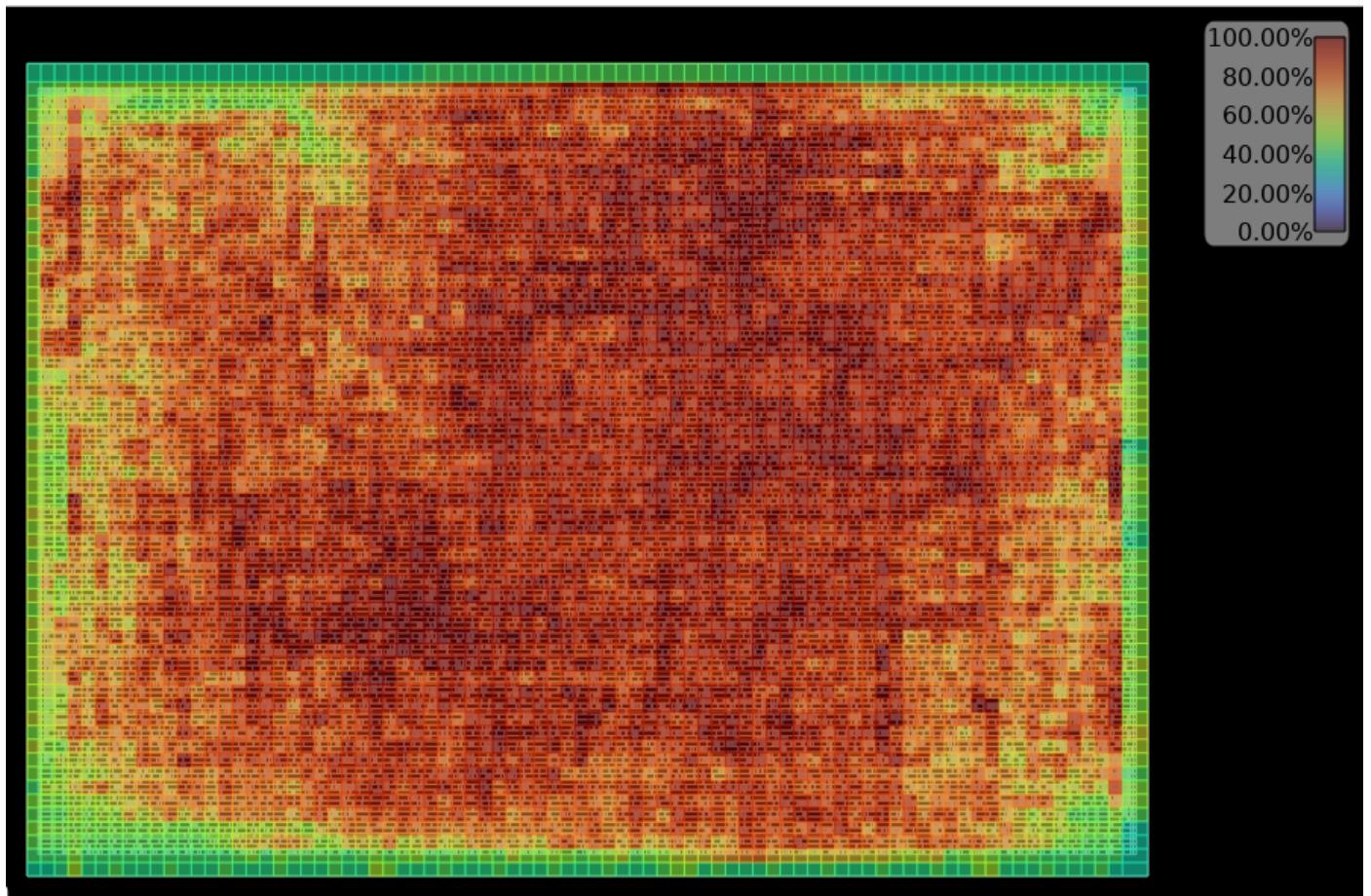
Parameter	Goal
Timing	No setup/hold violations
DRC Violation	0 violations for spacing, width, via
Congestion	No hotspots, smooth routes
Utilization	Within safe limits (below 70%)
LVS (Layout vs Schematic)	Netlist matches the schematic
DRV (Design Rule Violations)	Clean of violations
Antenna Violation	Avoid charge buildup on long wires
Power & Area	Within budget

Power Density Heat map:





Routing Congestion Heat Map:



Outputs:

- Final DEF file
- Extracted Netlist
- Final GDSII layout (after DRC and antenna fixes)
- SPEF



Timing reports

<pre>===== report_tns ===== tns 0.00 ===== report_wns ===== wns 0.00 ===== report_worst_slack -max (Setup) ===== worst slack 0.45 ===== report_worst_slack -min (Hold) ===== worst slack 0.19 25-grt_sta.summary.rpt (END)</pre>	<pre>===== report_tns ===== tns 0.00 ===== report_wns ===== wns 0.00 ===== report_worst_slack -max (Setup) ===== worst slack 0.40 ===== report_worst_slack -min (Hold) ===== worst slack 0.25 18-rsz_timing_sta.summary.rpt (E</pre>
---	---

Power report of routing :

<pre>===== report_power ===== ===== Typical Corner ===== Group Internal Power Switching Power Leakage Power Total Power (Watts) ----- Sequential 7.01e-03 2.94e-04 1.30e-08 7.31e-03 29.7% Combinational 5.46e-03 4.39e-03 5.37e-08 9.85e-03 40.0% Clock 5.17e-03 2.30e-03 5.42e-09 7.47e-03 30.3% Macro 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.0% Pad 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.0% ----- Total 1.76e-02 6.98e-03 7.21e-08 2.46e-02 100.0% 71.7% 28.3% 0.0%</pre>
<pre>25-grt_sta.power.rpt (END)</pre>



```
=====
report_power
=====
===== Typical Corner =====

Group          Internal Power   Switching Power   Leakage Power   Total Power (Watts)
-----
Sequential      7.00e-03    4.72e-04    1.30e-08    7.47e-03    27.1%
Combinational   5.49e-03    6.66e-03    5.37e-08    1.22e-02    44.1%
Clock           5.21e-03    2.75e-03    5.36e-09    7.96e-03    28.8%
Macro           0.00e+00    0.00e+00    0.00e+00    0.00e+00    0.0%
Pad             0.00e+00    0.00e+00    0.00e+00    0.00e+00    0.0%

Total          1.77e-02    9.88e-03    7.20e-08    2.76e-02    100.0%
       64.2%        35.8%        0.0%
```

18-rsz_timing_sta.power.rpt (END)



Signoff

Signoff is the final stage of the physical design flow, where the design is thoroughly verified to ensure that it meets all functional, timing, power, area, and manufacturing requirements before tape-out. This step includes checks for timing, DRC, LVS, antenna, and parasitic extraction to validate the integrity of the final layout.

Inputs

- Final routed DEF
- Final netlist
- Liberty timing libraries
- Technology LEF
- Parasitic data (extracted SPEF)
- Constraints file (.sdc)

Steps Involved in Signoff

1. Parasitic Extraction

- Extract RC values (resistance and capacitance) from the layout using tools like SPEF-Extractor.
- Required for accurate post-layout timing analysis.

2. Static Timing Analysis (STA)

- Re-run STA using extracted parasitics to ensure setup and hold times are still met.
- Ensure $WNS \geq 0$, $TNS = 0$, $NVP = 0$.



Timing reports of min, nominal/typical, max :

=====	=====	=====
report_tns	report_tns	report_tns
=====	=====	=====
tns -206.79	tns -289.19	tns -363.15
=====	=====	=====
report_wns	report_wns	report_wns
=====	=====	=====
wns -2.58	wns -2.96	wns -3.27
=====	=====	=====
report_worst_slack -max (Setup)	report_worst_slack -max (Setup)	report_worst_slack -max (Setup)
=====	=====	=====
worst slack -2.58	worst slack -2.96	worst slack -3.27
=====	=====	=====
report_worst_slack -min (Hold)	report_worst_slack -min (Hold)	report_worst_slack -min (Hold)
=====	=====	=====
worst slack 0.14	worst slack 0.15	worst slack 0.16
multi_corner_sta.summary.rpt (EN)	multi_corner_sta.summary.rpt (EN)	multi_corner_sta.summary.rpt (EN)

3. Physical Verification

- DRC (Design Rule Check): Checks the design against the foundry's manufacturing rules.
- LVS (Layout vs Schematic): Ensures the layout matches the schematic/netlist.
- Antenna Check: Verifies that no long metal segments could damage gates during fabrication.
- ERC (Electrical Rule Check): Validates power/ground connections and well tap coverage.

4. Power Analysis

- Run a power report to check dynamic and leakage power using post-layout data.



Power reports for max corner:

===== Fastest Corner =====					
Group	Internal Power	Switching Power	Leakage Power	Total Power (Watts)	
Sequential	8.34e-03	1.98e-03	4.19e-05	1.04e-02	21.3%
Combinational	7.17e-03	1.90e-02	1.25e-04	2.63e-02	54.1%
Clock	6.72e-03	5.21e-03	1.17e-05	1.19e-02	24.6%
Macro	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.0%
Pad	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.0%
Total	2.22e-02	2.62e-02	1.79e-04	4.86e-02	100.0%
	45.7%	53.9%	0.4%		

===== Slowest Corner =====					
Group	Internal Power	Switching Power	Leakage Power	Total Power (Watts)	
Sequential	5.45e-03	1.34e-03	2.20e-05	6.80e-03	21.5%
Combinational	4.43e-03	1.28e-02	6.60e-05	1.73e-02	54.5%
Clock	4.26e-03	3.35e-03	6.15e-06	7.62e-03	24.1%
Macro	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.0%
Pad	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.0%
Total	1.41e-02	1.75e-02	9.42e-05	3.17e-02	100.0%
	44.6%	55.1%	0.3%		

===== Typical Corner =====					
Group	Internal Power	Switching Power	Leakage Power	Total Power (Watts)	
Sequential	7.00e-03	1.68e-03	1.30e-08	8.69e-03	21.8%
Combinational	5.52e-03	1.61e-02	5.37e-08	2.16e-02	54.3%
Clock	5.21e-03	4.33e-03	7.02e-08	9.53e-03	23.9%
Macro	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.0%
Pad	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.0%
Total	1.77e-02	2.21e-02	1.37e-07	3.99e-02	100.0%
	44.5%	55.5%	0.0%		



Power reports for nominal corner :

===== report_power =====					
===== Fastest Corner =====					
Group	Internal Power	Switching Power	Leakage Power	Total Power (Watts)	
Sequential	8.34e-03	1.88e-03	4.19e-05	1.03e-02	21.7%
Combinational	7.12e-03	1.80e-02	1.25e-04	2.53e-02	53.5%
Clock	6.71e-03	5.03e-03	1.17e-05	1.17e-02	24.8%
Macro	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.0%
Pad	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.0%
Total	2.22e-02	2.49e-02	1.79e-04	4.73e-02	100.0%
	46.9%	52.8%	0.4%		

===== Slowest Corner =====					
===== Slowest Corner =====					
Group	Internal Power	Switching Power	Leakage Power	Total Power (Watts)	
Sequential	5.45e-03	1.27e-03	2.20e-05	6.74e-03	21.9%
Combinational	4.43e-03	1.21e-02	6.60e-05	1.66e-02	53.8%
Clock	4.26e-03	3.23e-03	6.15e-06	7.50e-03	24.3%
Macro	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.0%
Pad	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.0%
Total	1.41e-02	1.66e-02	9.42e-05	3.08e-02	100.0%
	45.9%	53.8%	0.3%		

===== Typical Corner =====					
===== Typical Corner =====					
Group	Internal Power	Switching Power	Leakage Power	Total Power (Watts)	
Sequential	7.00e-03	1.60e-03	1.30e-08	8.60e-03	22.2%
Combinational	5.52e-03	1.53e-02	5.37e-08	2.08e-02	53.6%
Clock	5.21e-03	4.17e-03	7.02e-08	9.38e-03	24.2%
Macro	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.0%
Pad	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.0%
Total	1.77e-02	2.11e-02	1.37e-07	3.88e-02	100.0%
	45.7%	54.3%	0.0%		



Power report of min corner :

===== Fastest Corner =====					
Group	Internal Power	Switching Power	Leakage Power	Total Power (Watts)	
Sequential	8.34e-03	1.72e-03	4.19e-05	1.01e-02	22.2%
Combinational	7.07e-03	1.67e-02	1.25e-04	2.39e-02	52.5%
Clock	6.69e-03	4.82e-03	1.17e-05	1.15e-02	25.3%
Macro	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.0%
Pad	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.0%
Total	2.21e-02	2.32e-02	1.79e-04	4.55e-02	100.0%
	48.6%	51.0%	0.4%		

===== Slowest Corner =====					
Group	Internal Power	Switching Power	Leakage Power	Total Power (Watts)	
Sequential	5.45e-03	1.16e-03	2.20e-05	6.63e-03	22.4%
Combinational	4.44e-03	1.12e-02	6.60e-05	1.57e-02	52.8%
Clock	4.26e-03	3.09e-03	6.15e-06	7.35e-03	24.8%
Macro	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.0%
Pad	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.0%
Total	1.41e-02	1.54e-02	9.42e-05	2.97e-02	100.0%
	47.7%	52.0%	0.3%		

===== Typical Corner =====					
Group	Internal Power	Switching Power	Leakage Power	Total Power (Watts)	
Sequential	7.00e-03	1.46e-03	1.30e-08	8.47e-03	22.7%
Combinational	5.52e-03	1.41e-02	5.37e-08	1.96e-02	52.6%
Clock	5.21e-03	3.99e-03	7.02e-08	9.20e-03	24.7%
Macro	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.0%
Pad	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.0%
Total	1.77e-02	1.96e-02	1.37e-07	3.73e-02	100.0%
	47.5%	52.5%	0.0%		



Skew values in min, nominal ,max corner reports:

report_clock_skew		
Clock clk		
Latency	CRPR	Skew
19181/_CLK ^		
2.27		
18651/_CLK ^		
1.74	-0.04	0.49
multi_corner_sta.skew.rpt (END)		

report_clock_skew		
Clock clk		
Latency	CRPR	Skew
18763/_CLK ^		
2.36		
17889/_CLK ^		
1.80	-0.04	0.52
multi_corner_sta.skew.rpt (END)		

report_clock_skew		
Clock clk		
Latency	CRPR	Skew
18763/_CLK ^		
2.43		
17889/_CLK ^		
1.86	-0.04	0.53
multi_corner_sta.skew.rpt (END)		

5. Final Quality Metrics

- Timing (no violations)
- Area (within limits)
- Power (within limits)
- Clean DRC/LVS/ANT reports
- Design is ready for GDSII generation

Quality Checks

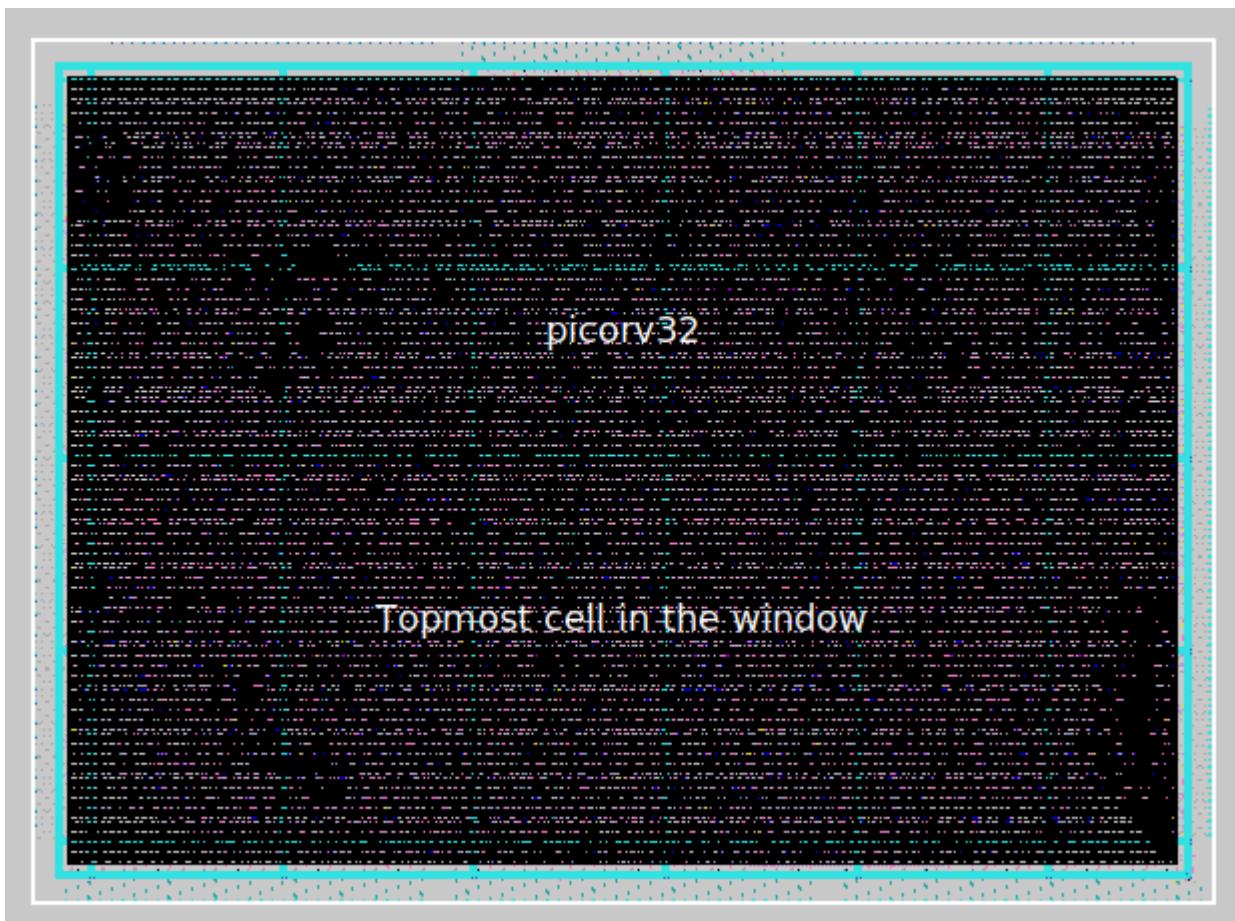
- Timing: Must meet or exceed setup/hold constraints
- DRC: No violations
- LVS: Netlist and layout must match
- Antenna: All violations must be fixed
- Power: Should be within target limits
- Utilization: Final utilization should be acceptable (<70%)
- GDSII generation: Layout must be exportable in GDSII format for fabrication



Outputs

- Verified GDSII file
- Final extracted netlist
- Signoff reports (STA, DRC, LVS, Power)
- SPEF file
- Tape-out-ready design archive

Final layout in Magic 8.3 version





Final Design Reports and Metrics

1. Area

Name of the Stage	Area(um)
Synthesis	96597.644800
Floorplan	Die: 231233.214, Core: 213737.4912
Placement	GPL: 100381, DPL: 101342
CTS	125318
Routing	Die: 231233, Core: 213737
Signoff	122615.555

2. Utilization

Name of the stage	Utilization(%)
Synthesis	-
Floorplan	47
Placement	47
CTS	59
Routing	69
Signoff	-



3. Power

Name of the stage	Power(Watts)
Synthesis	1.34×10^{-2}
Floorplan	-
Placement	GPL: 1.63×10^{-2} , DPL: 1.63×10^{-2}
CTS	2.09×10^{-2}
Routing	1.89×10^{-2}
Signoff	Min: fastest -3.38×10^{-2} , slowest -2.19×10^{-2} , typical -2.76×10^{-2}
	Nominal: fastest -3.52×10^{-2} , slowest -2.27×10^{-2} , typical -2.87×10^{-2}
	Max: fastest -3.64×10^{-2} , slowest -2.35×10^{-2} , typical -2.97×10^{-2}

4. Setup and hold slack values

Name of the stage	In-reg, reg-out setup(ns)	Reg to reg setup (ns)	In-reg, reg-out hold(ns)	Reg to reg hold(ns)
Synthesis	I-r: 1.12, r-o: 0.64	0.98	-	-1.0691
Floorplan	-	-	-	-
Placement	I-r: 0.97, r-o: 0.78	0.93	-	-1.0515
CTS	I-r: 1.81, r-o: -0.11	1.84	-	-1.06
Routing	I-r: 2.38, r-o: 1.44	-	-	0.23
Signoff (Typical/ nominal)	FF I-r: 2.93, r-o: 1.24	-	-	0.14
	SS I-r: 2.38, r-o: 1.44	-	-	1.43
	TT I-r: 2.38, r-o: -0.43	-	-	0.46



Social / Industry Relevance of the Project

This project is highly relevant to both the industry and society, especially in today's fast-growing semiconductor world. With the rise of smart devices, IoT, and AI applications, there is a strong demand for efficient and low-power processors. The RISC-V architecture, being open-source and flexible, is gaining popularity among chip designers worldwide.

By implementing the PicoRV32 RISC-V processor from RTL to GDSII using open-source tools, we are contributing to the growing ecosystem of accessible chip design. It shows how even students and fresh engineers can learn the full VLSI flow without expensive EDA tools. This bridges the gap between academic knowledge and real-world industry practices.

From a social perspective, projects like these empower rural and underrepresented students with hands-on skills in advanced technology. It helps create job-ready engineers who can contribute to India's semiconductor mission and global VLSI needs. It also promotes self-learning, innovation, and a deeper understanding of how the chips inside our everyday electronics are made.

This kind of project makes VLSI design more inclusive, cost-effective, and accessible, opening doors for more local talent to enter the semiconductor industry.



Learning & Reflection

Working on this project was a great learning experience for me. I started with just a basic understanding of RTL and chip design, but through this training, I got to see how an actual ASIC is developed from scratch using the RTL to GDSII flow. Each stage—synthesis, floorplanning, placement, CTS, routing, and signoff- had its importance and challenges, and I learned how to handle them step by step.

One of the most valuable parts was learning how to apply real constraints and analyze timing, utilization, power, and area. I also understood how each tool in the OpenLane flow works and how they connect to form a complete design pipeline. Hands-on work helped me understand things better than just reading theory, and the small issues I faced taught me even more than the expected flow.

This project also taught me the importance of debugging, patience, and keeping documentation properly. I now feel more confident about the VLSI flow and ASIC design. I realized how important it is to understand timing reports, power numbers, and physical layouts, which will help in future job roles or advanced projects in this field.

Lastly, I want to thank my mentor, Veeramani Sir, and trainers who supported me throughout. This experience gave me clarity about my interest in VLSI and motivated me to keep learning and growing in this domain.



Future Scope & Conclusion

Future Scope:

This project focused on implementing the RTL to GDSII flow of the PicoRV 32-bit processor using the OpenLane toolchain. While I was able to complete the full flow and generate the final GDSII, there is still a lot of scope for future improvement and learning:

- Power Optimization: Further reduction of power consumption using advanced clock gating techniques or multi-Vt cells.
- Area Optimization: Improving placement strategies and exploring better floorplanning to reduce chip area.
- Advanced Nodes: Trying out the same flow with more complex standard cell libraries or different PDKs to see how it behaves on newer technology nodes.
- Timing Closure Techniques: Deep diving into setup and hold violation fixing.
- Custom IP Integration: In the future, I can explore how to integrate memory IPs, analog blocks, or other IP cores into the flow for a more realistic SoC experience.

Conclusion:

This project helped me understand the complete backend design flow used in the VLSI industry—from Verilog RTL to final GDSII layout. It showed how different EDA tools come together in a well-defined sequence to shape a working silicon design.

Through this hands-on experience, I not only applied theoretical concepts but also learned how to debug and resolve practical issues during each stage.

I feel this project has built a strong foundation in Physical design and made me more confident to work on future VLSI projects and opportunities in the semiconductor industry.

Special thanks to our whole VLSI training team and Beloved Radha Kumari mam.