

OpenLane

A Digital ASIC Flow for SkyWater 130nm Open PDK

efabless.com



THE AMERICAN
UNIVERSITY IN CAIRO

Who am I?

- Engineer
 - EDA Developer: Mentor Graphics
 - Digital ASIC Designer: Motorola SPS
 - Embedded Developer: Mentor Graphics and Mindspeed
 - Consultant: eFabless
- Educator
- Open-Source EDA Believer

Who am I?

- Engineer
- Educator
 - Associate Professor at AUC
- Open-Source EDA Believer

Who am I?

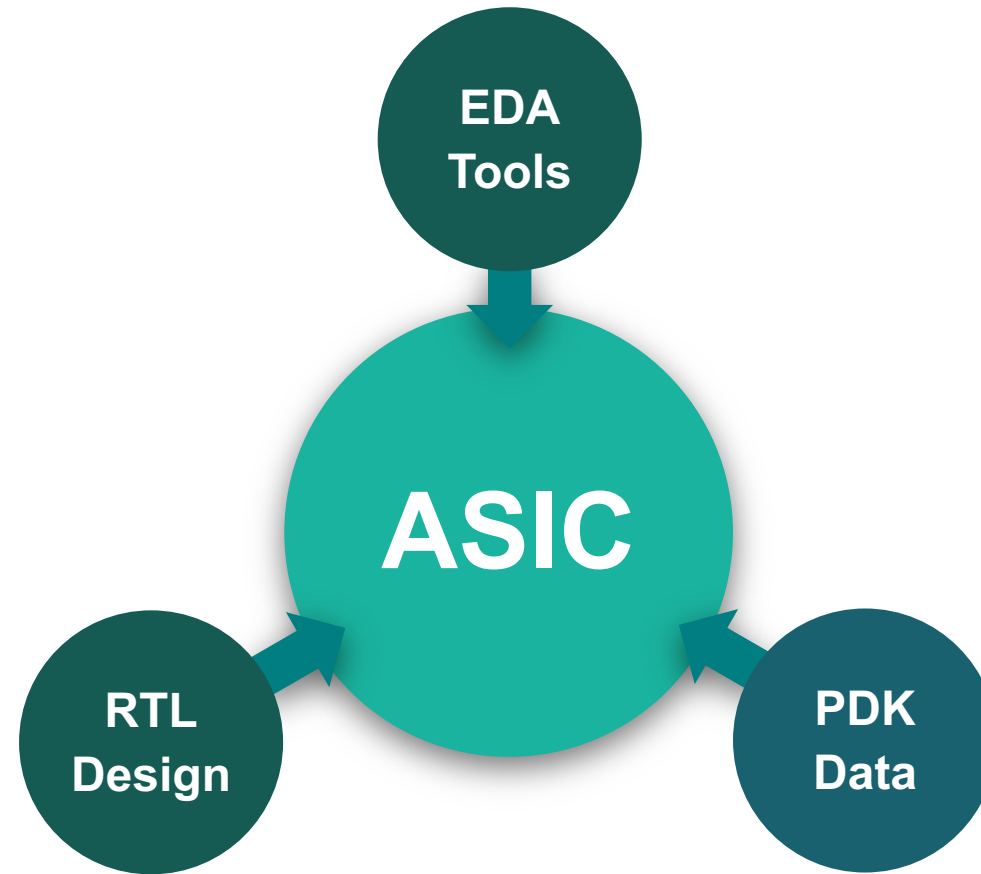
- Engineer
- Educator
- Open-Source EDA Believer
 - CloudV
 - Fault
 - ...



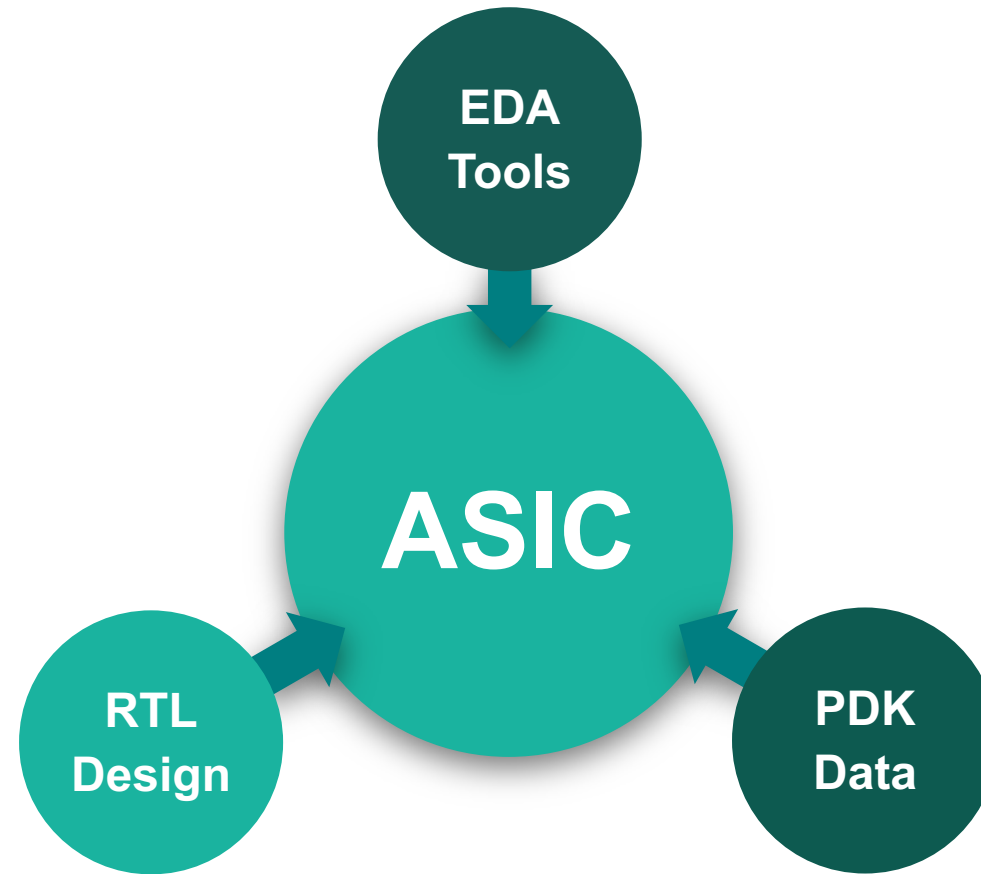
Why am I here
today?

Present a piece of the puzzle of enabling silicon
access to all innovators

Digital ASIC Design



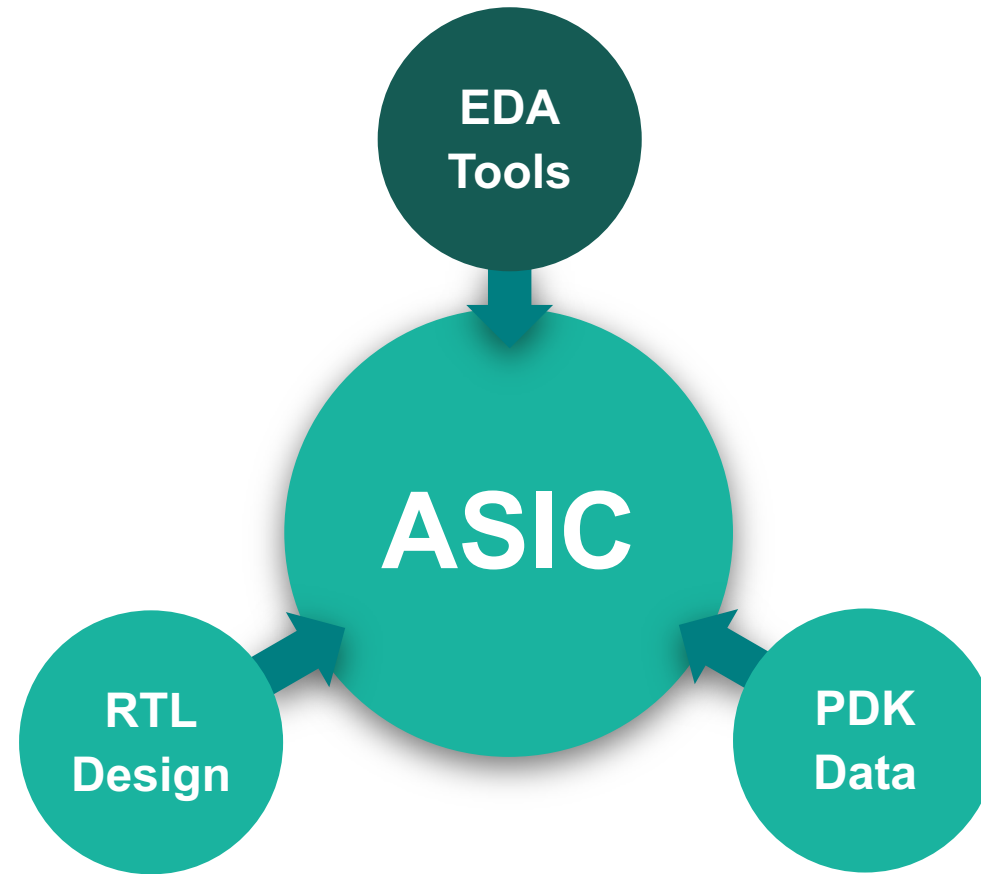
Open Source Digital ASIC Design



librecores.org
opencores.org
github.com

...

Open Source Digital ASIC Design



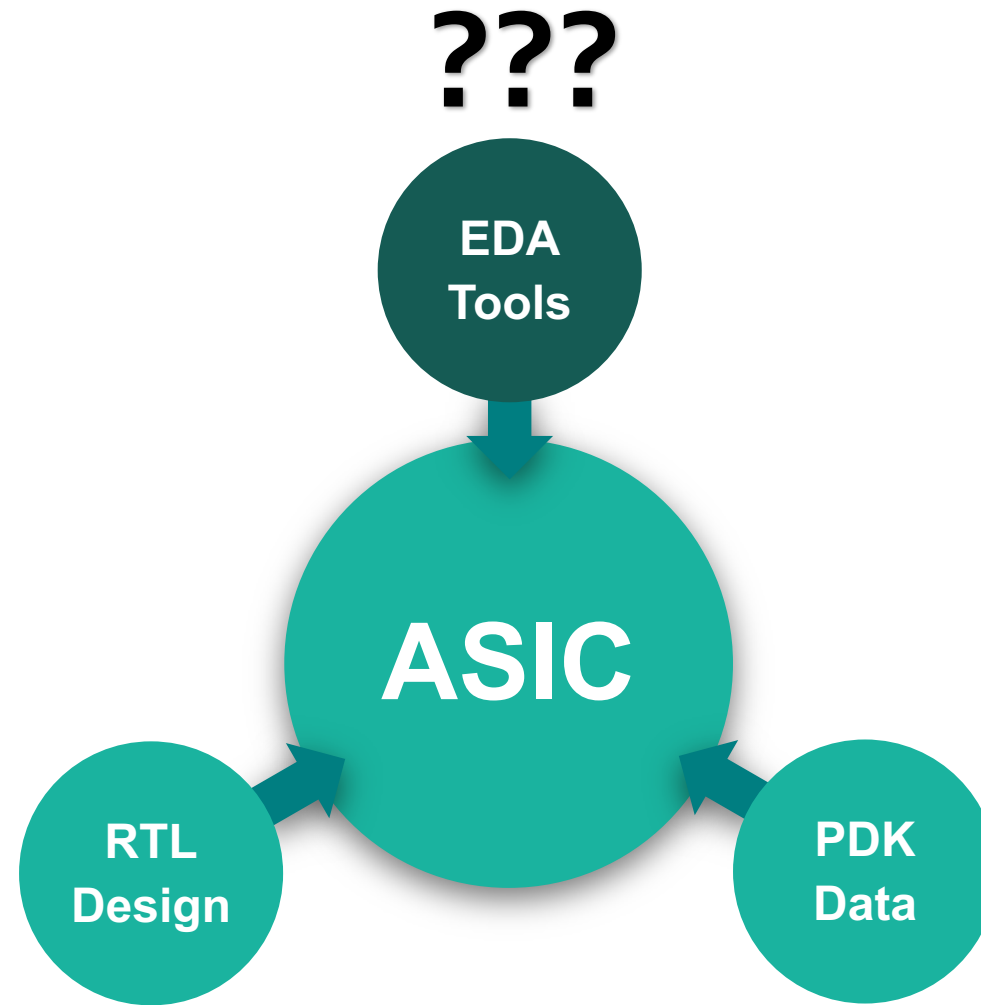
librecores.org
opencores.org
github.com

...



FOSS 130nm Production PDK
github.com/google/skywater-pdk

Open Source Digital ASIC Design



librecores.org
opencores.org
github.com

...



FOSS 130nm Production PDK
github.com/google/skywater-pdk

EDA Tools

Floor Planning

HDL Design Entry

IR Drop Analysis

Logic Equivalency Checker

DFM

DFT

Power Planning

Static Timing Analysis

Logic Synthesis

Global Routing

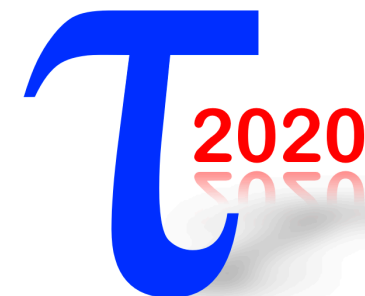
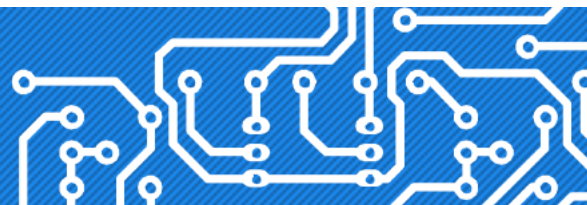
Detailed Routing

Static Code Analysis

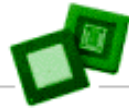
Clock Tree Synthesis

Open Source EDA?

International Symposium
on Physical Design



Open Source EDA?



Alli@nce

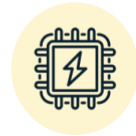
► A free VLSI cad system



CORIOLIS VLSI CAD TOOLS

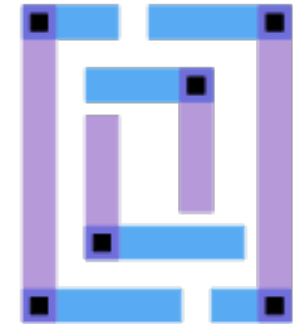
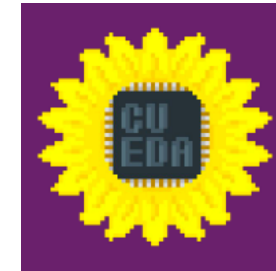


Electric



Fault

Yosys



QFlow



Cloud V



ABC

OpenROAD



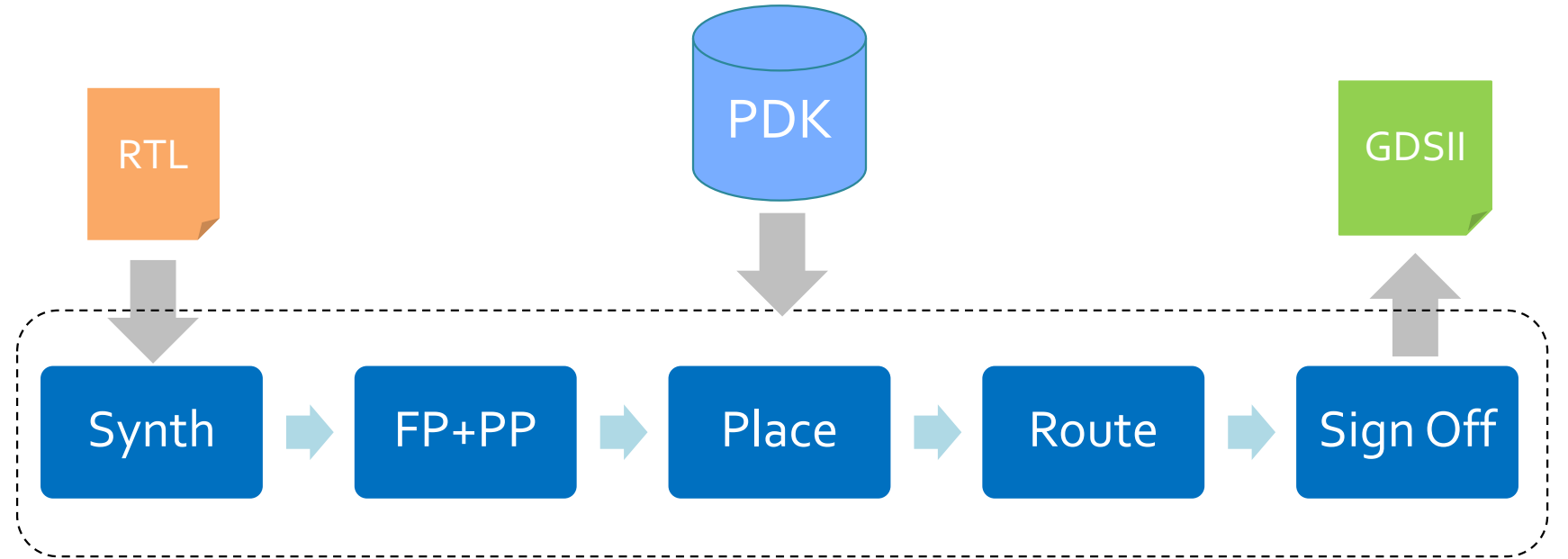
Open Source EDA

So, we have Open Source EDA tools. Are we ready for Open Source ASIC?

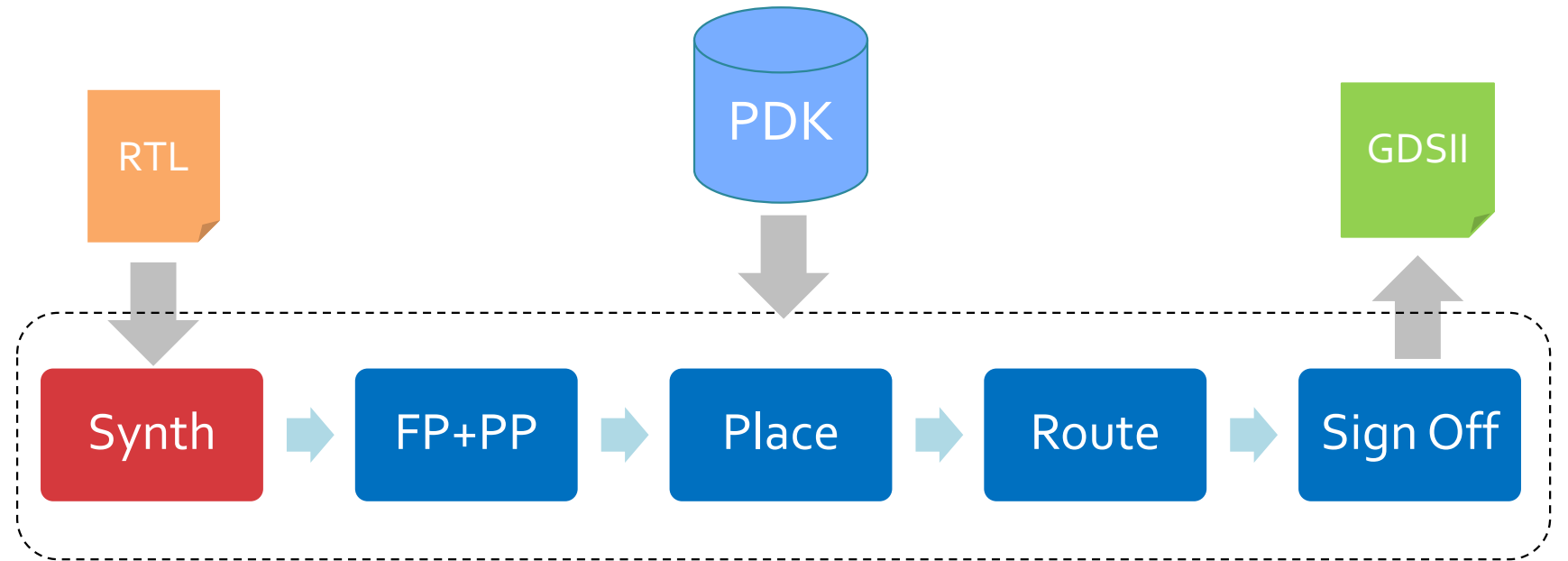
ASIC Design Flow

- ASIC implementation requires numerous steps and involves tons of details
- A **methodology** is needed for a successful ASIC implementation
- The methodology is implemented through a **flow** that glues different tools together to carry out the design process
- ASIC Flow Objective: **RTL to GDSII**

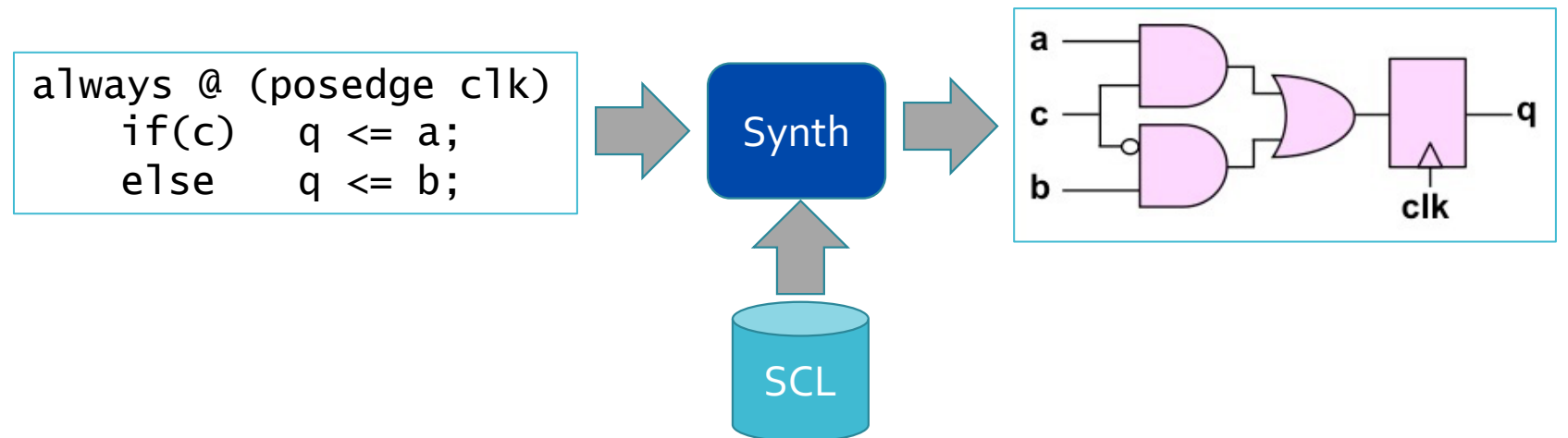
Simplified RTL to GDSII Flow



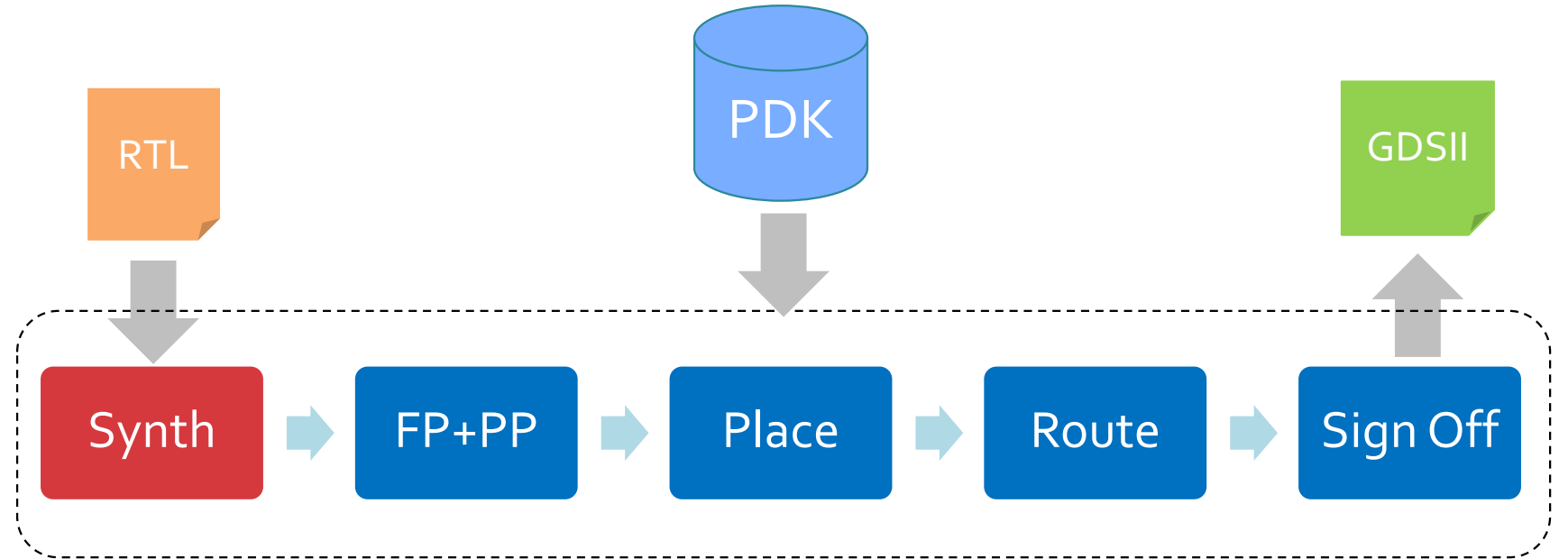
Synthesis



- Converts RTL to a circuit out of components from the standard cell library (SCL)

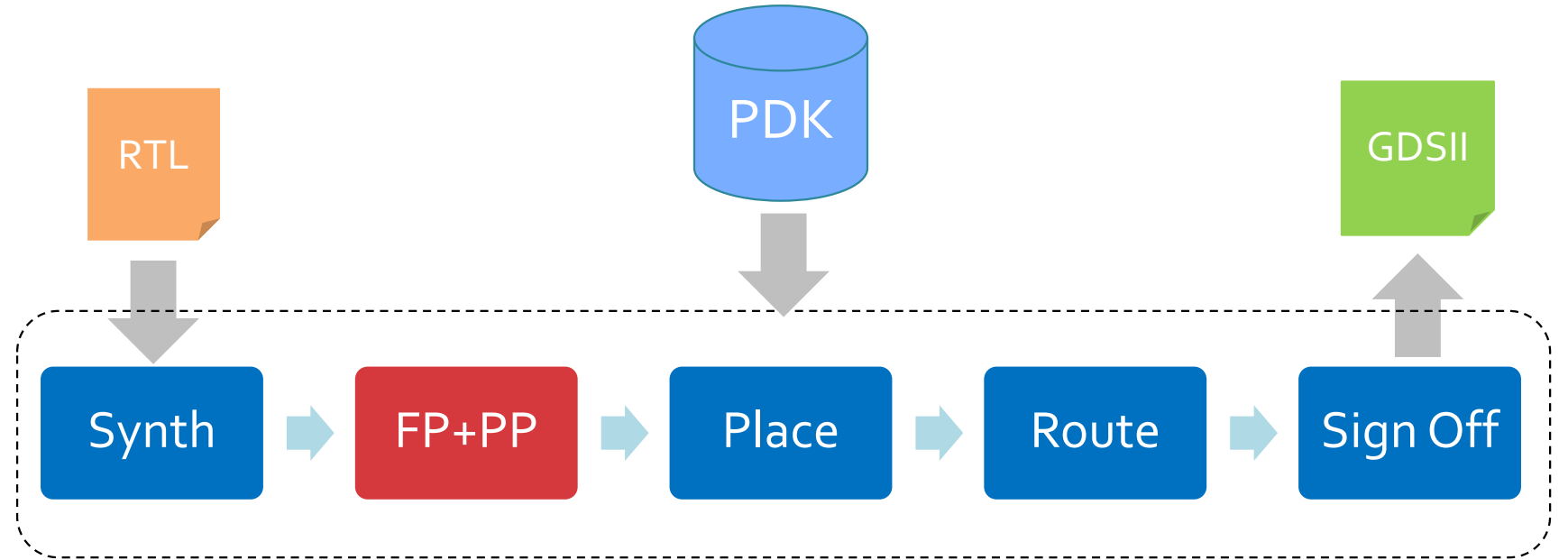


Synthesis

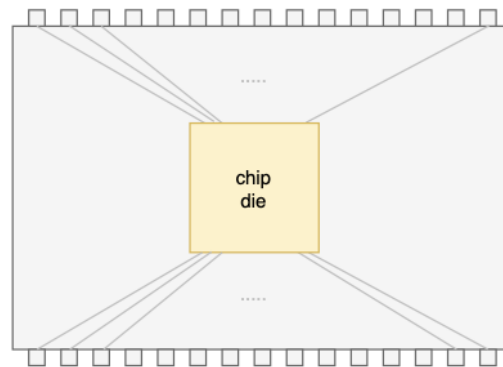


- Converts RTL to a circuit out of components from the standard cell library (SCL)
- “Standard Cells” have **regular** layout
- Each has different views/models

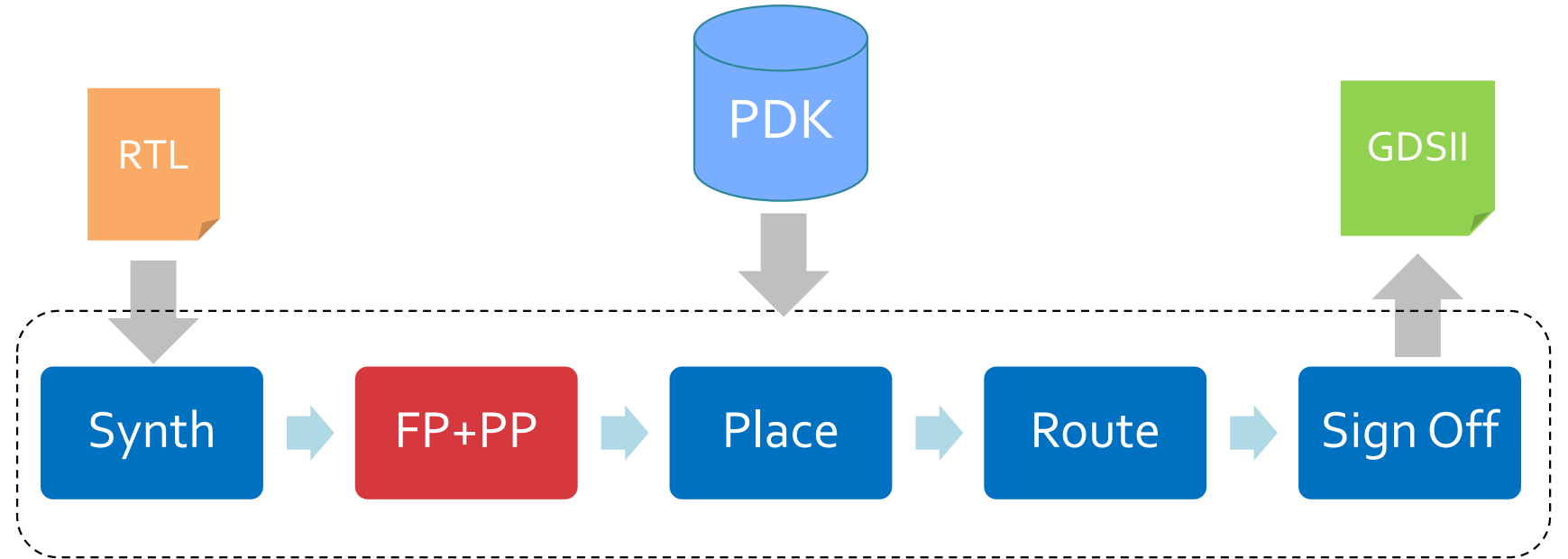
Floor and Power Planning



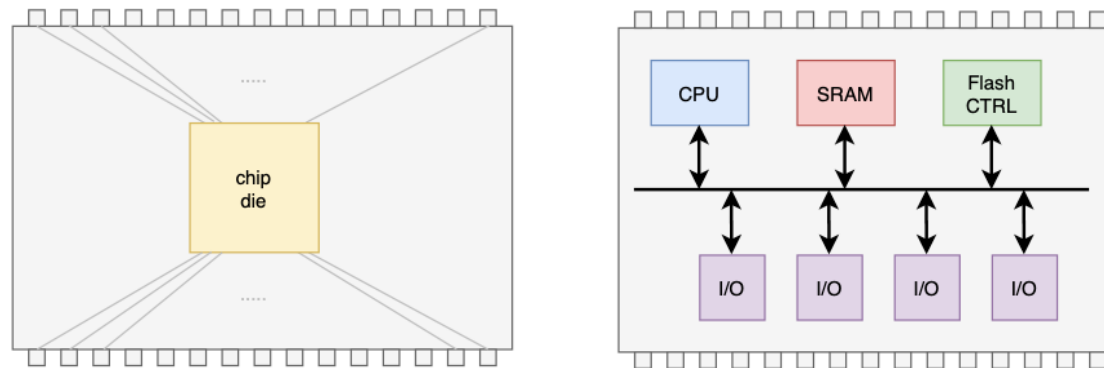
- Chip Floor-Planning: Partition the chip die between different system building blocks and place the I/O Pads



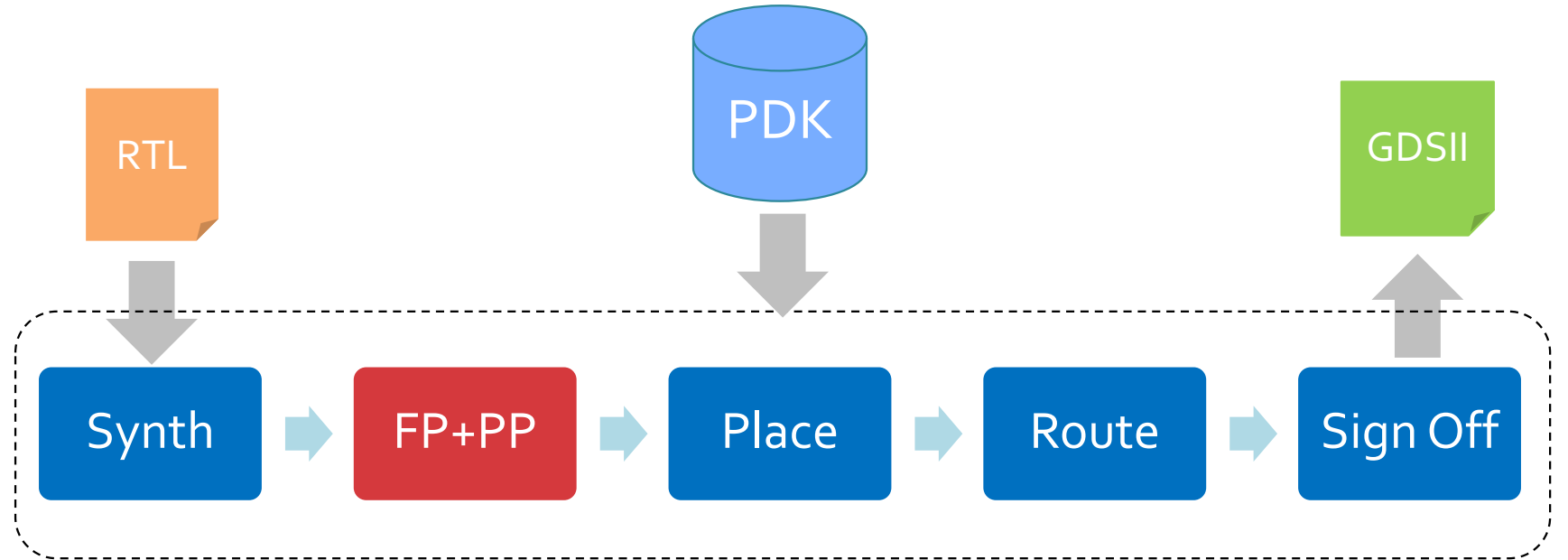
Floor and Power Planning



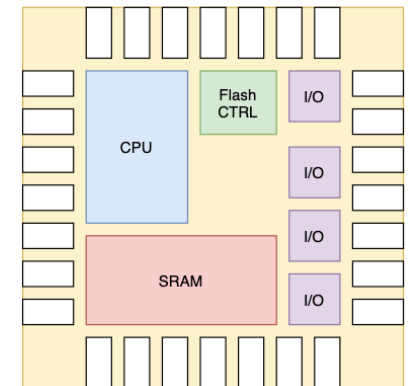
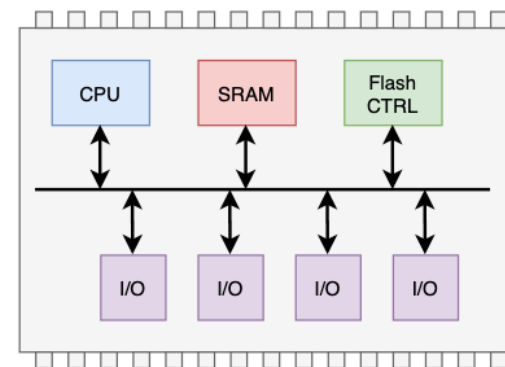
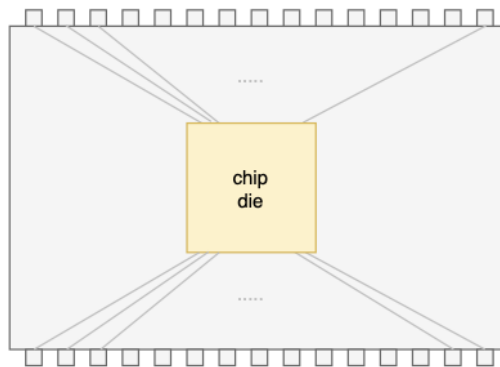
- Chip Floor-Planning: Partition the chip die between different system building blocks and place the I/O Pads



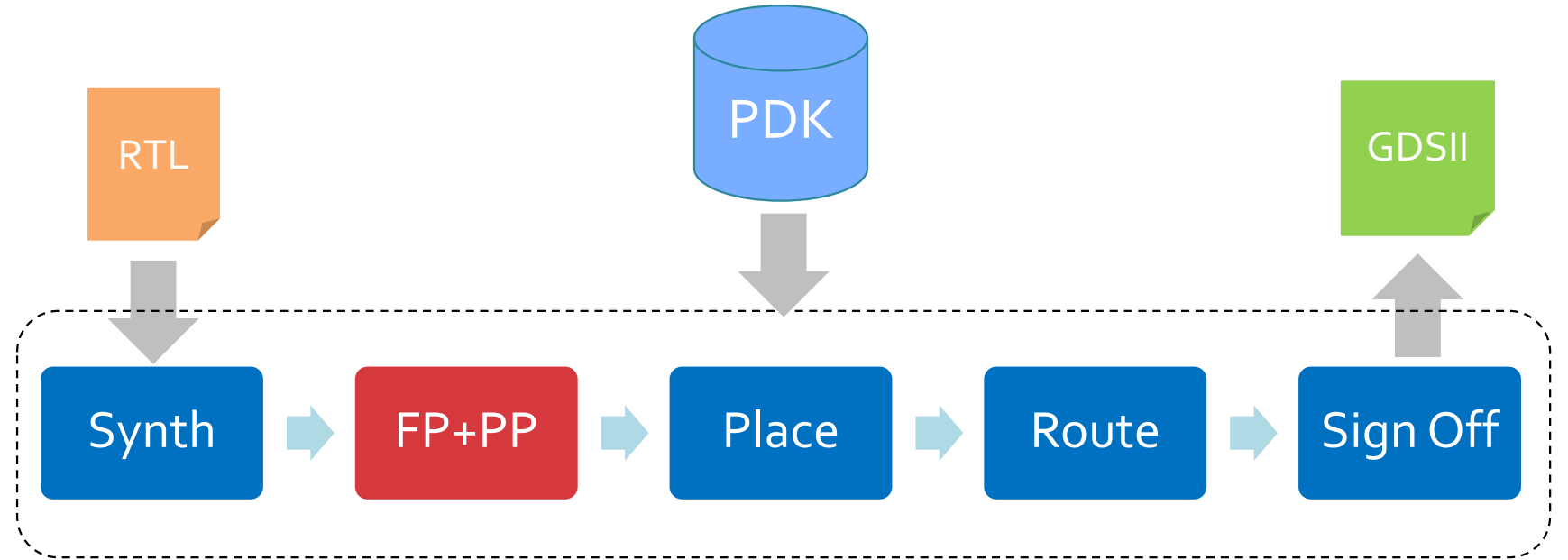
Floor and Power Planning



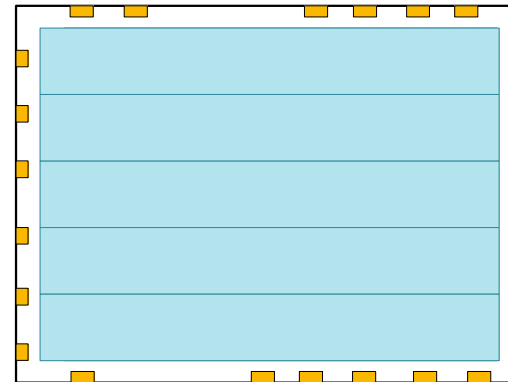
- Chip Floor-Planning: Partition the chip die between different system building blocks and place the I/O Pads



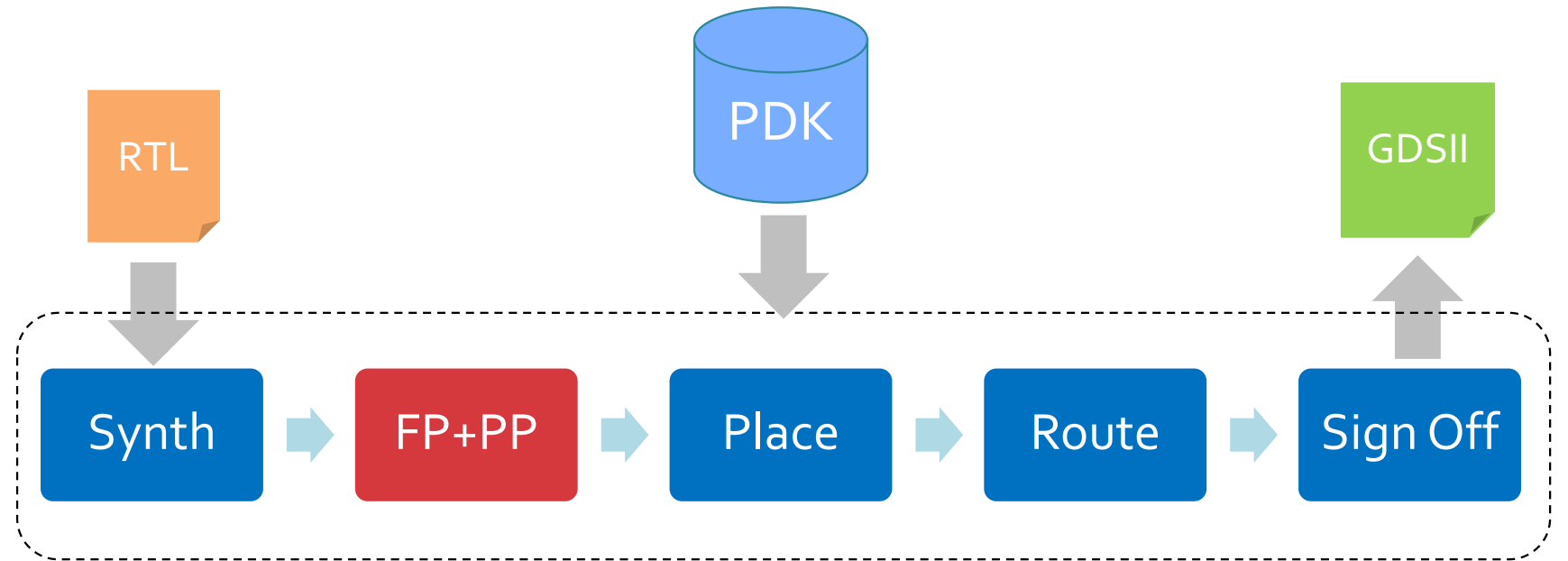
Floor and Power Planning



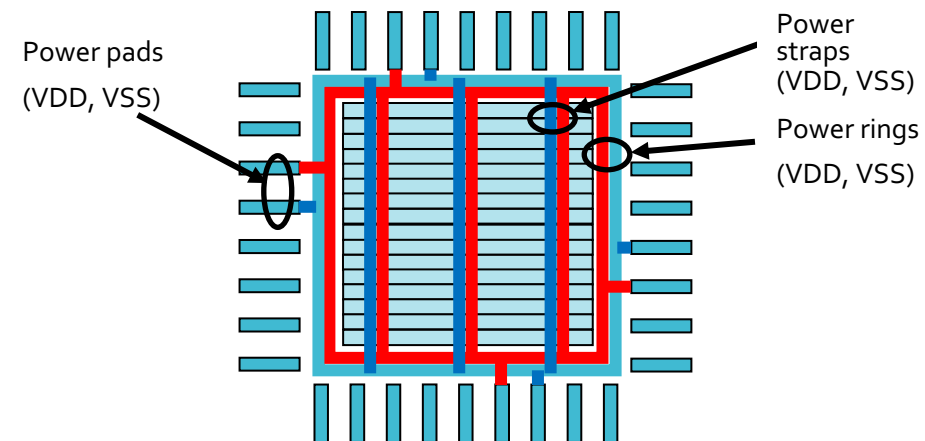
- Macro Floor-Planning: Dimensions, pin locations, rows definition



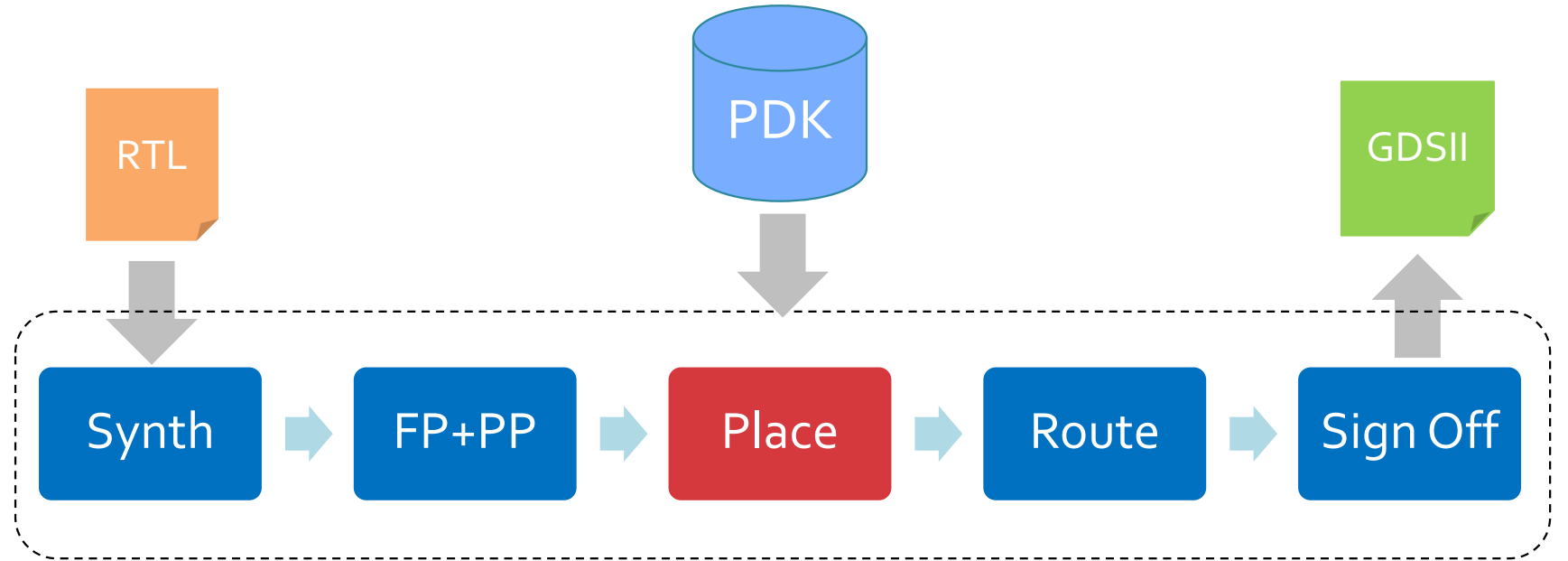
Floor and Power Planning



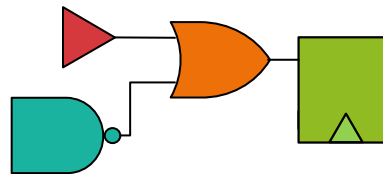
- Power Planning



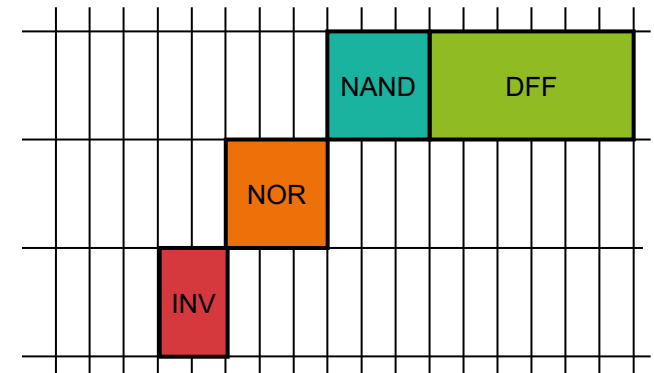
Placement



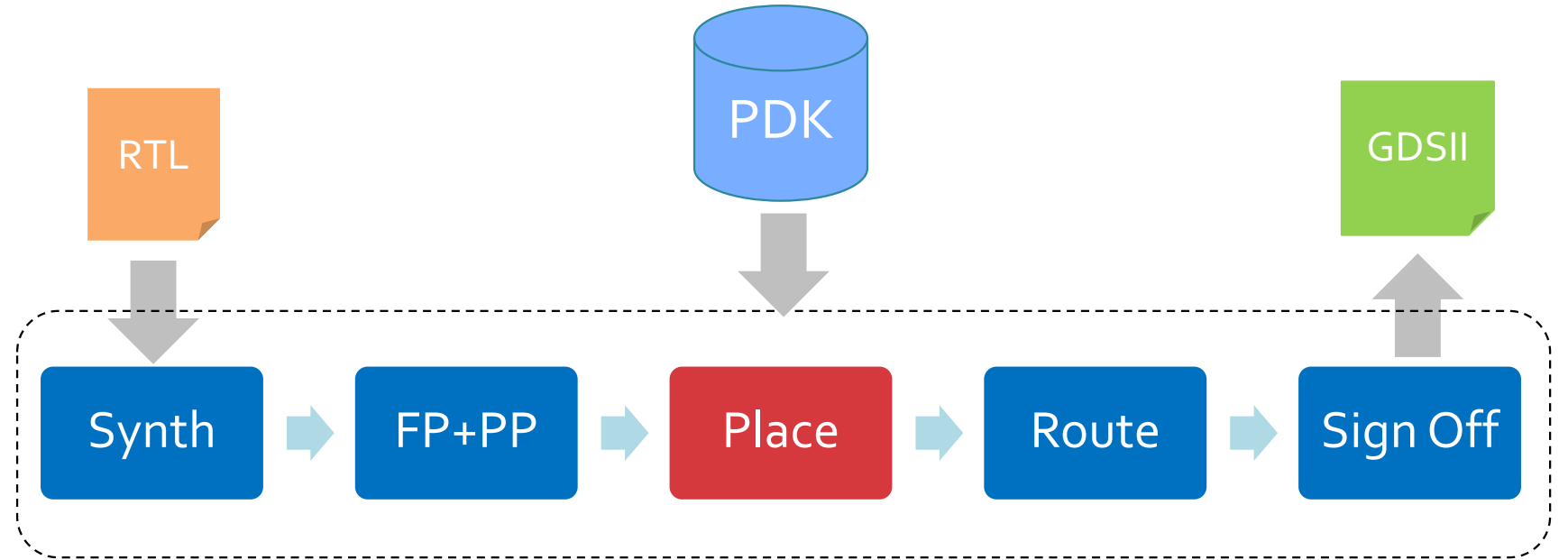
- Place the cells on the floorplan rows, aligned with the sites



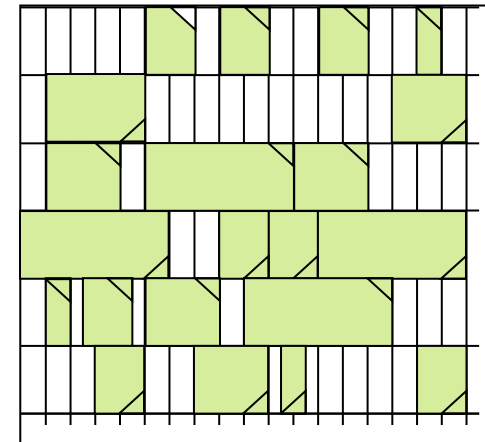
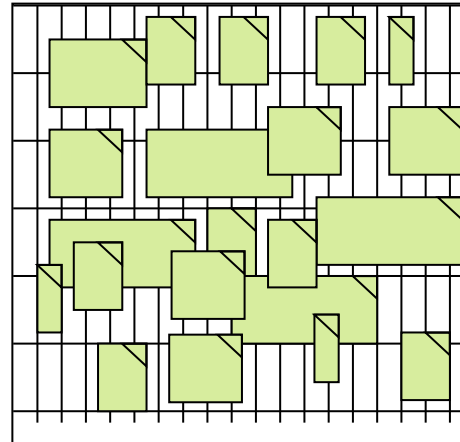
Gate-Level Netlist



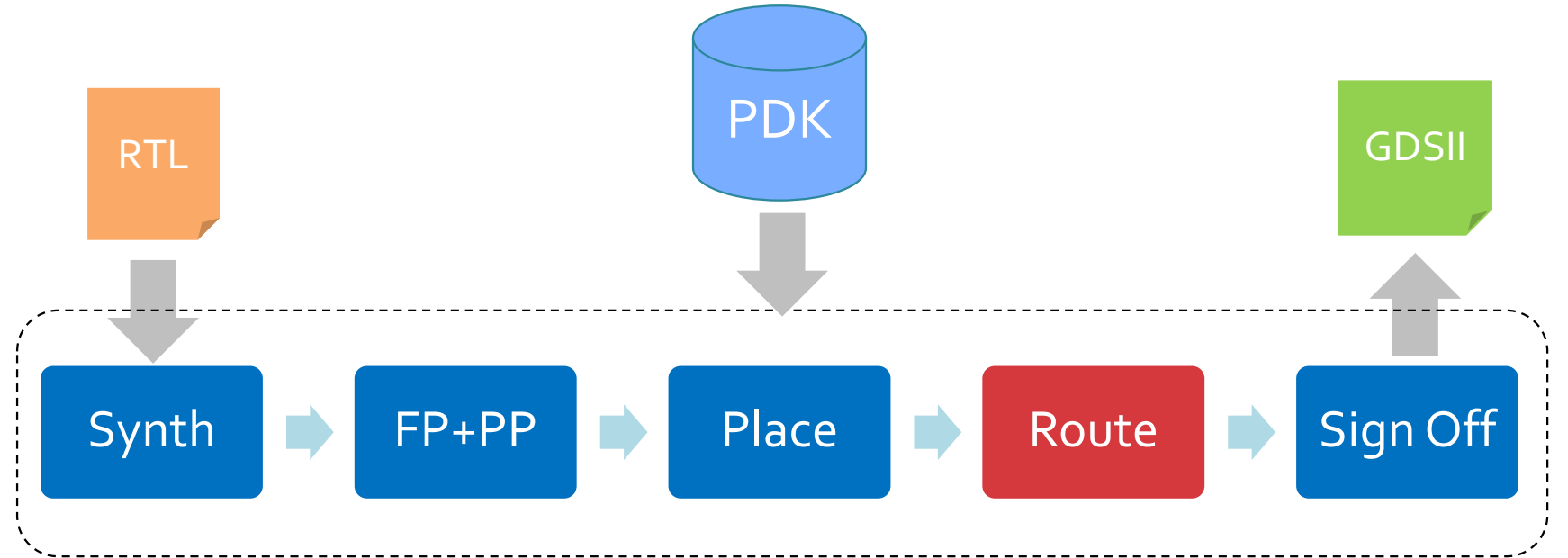
Placement



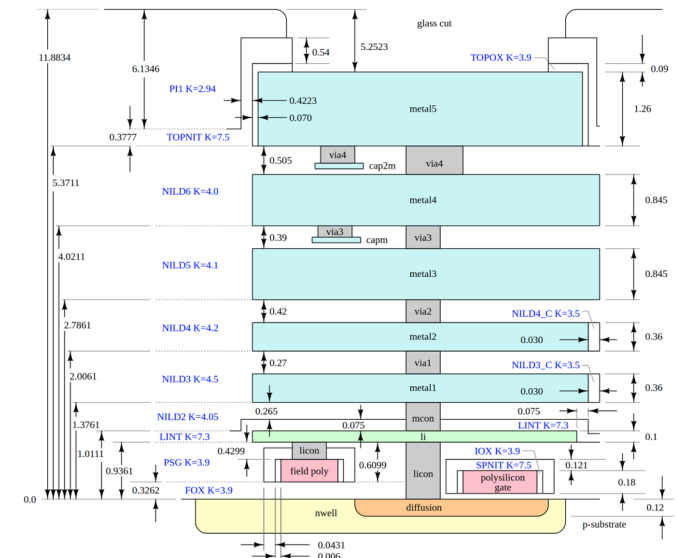
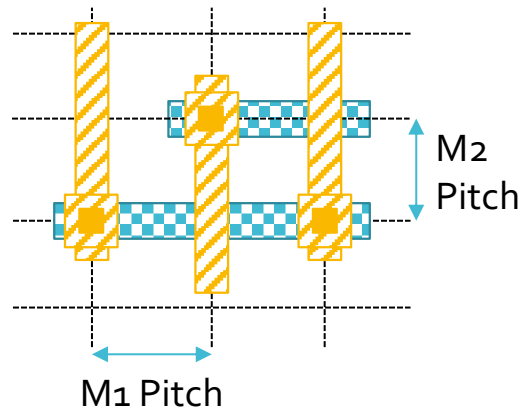
- Usually done in 2 steps: Global and Detailed



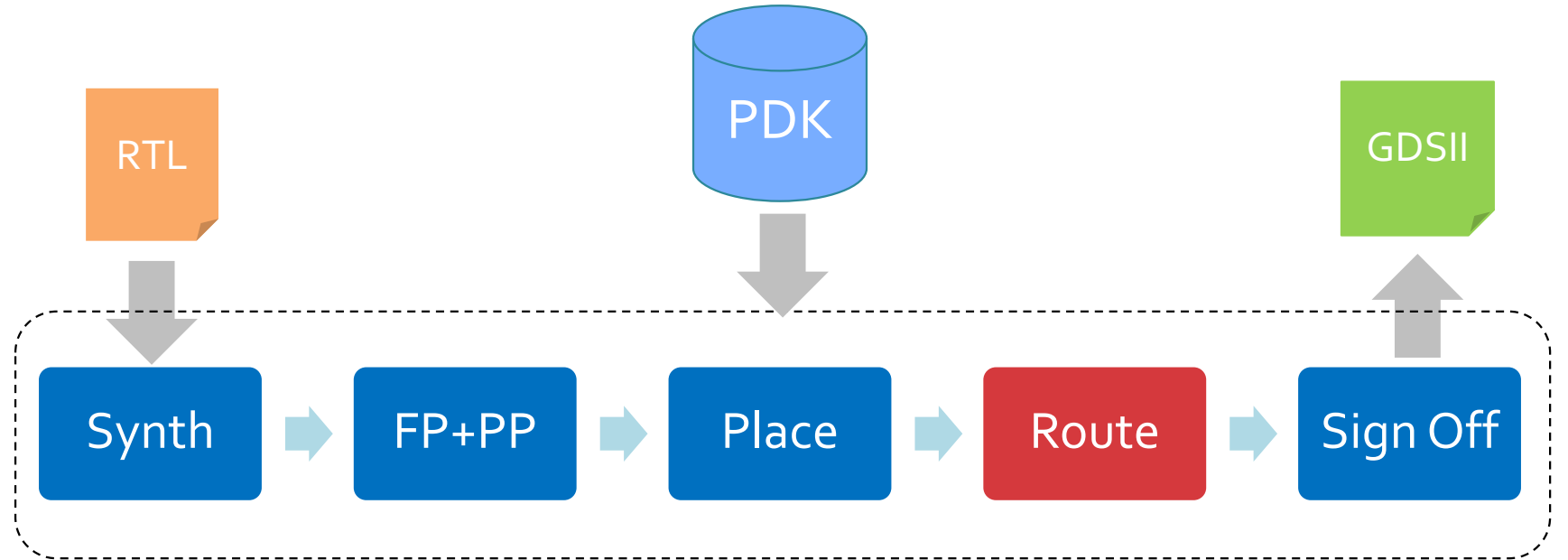
Routing



- Implement the interconnect using the available metal layers

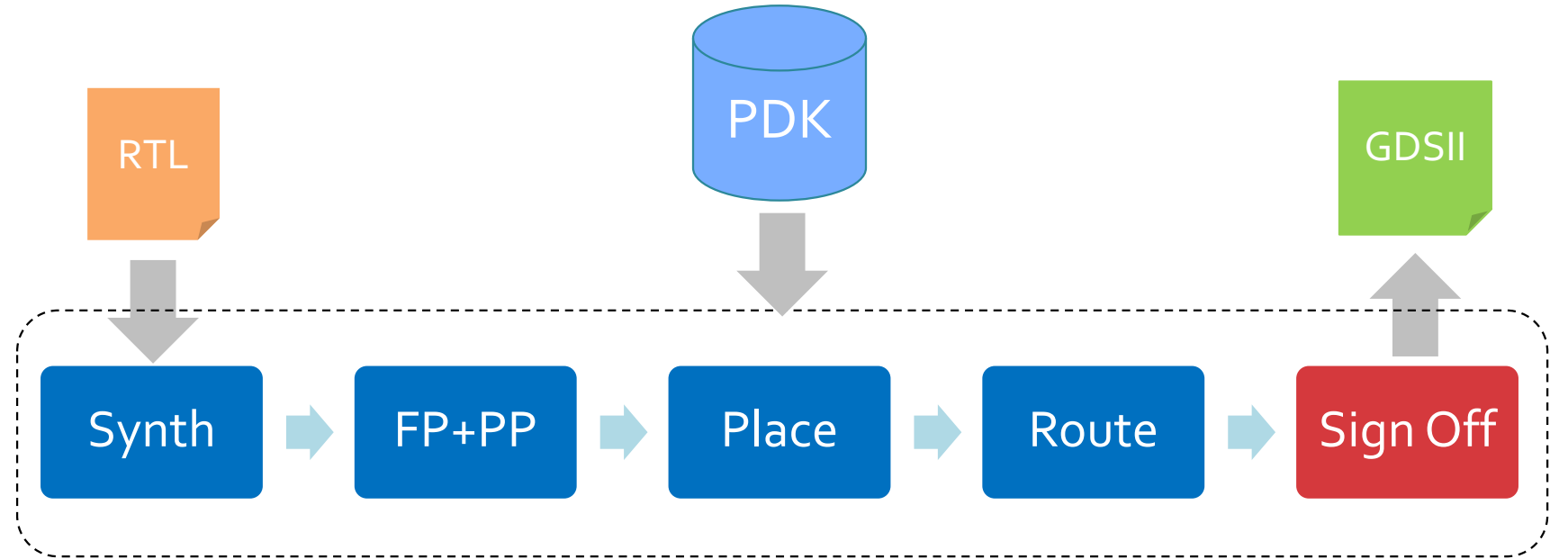


Routing



- Metal tracks form a routing grid
- Routing grid is huge
- Divide and Conquer
 - Global Routing: Generates routing guides
 - Detailed Routing: Uses the routing guides to implement the actual wiring

Sign Off



- Design Rules Checking (DRC)
- Layout vs. Schematic (LVS)
- Static Timing Analysis (STA)

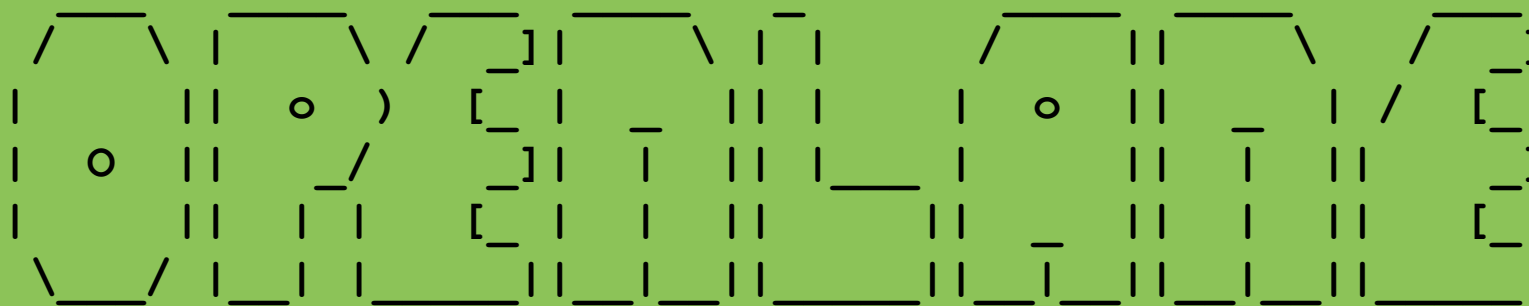
EDA File Formats

- Liberty (LIB):
 - Standard Cells Electrical Models
- Library Exchange Format (LEF):
 - Abstract Layout of cells and finished designs
 - Technology information
- Design Exchange Format (DEF):
 - Design Abstract Layout
- Standard Parasitic Exchange Format (SPEF):
 - RC values of the design wires
- Graphic Design System (GDS)
 - Final Layout (all details)
 - Used to print photomasks used in fabrication

Open Source ASIC Flow

- The Problem is tougher when using OS EDA
 - Tools qualification
 - Tools Calibration
 - Missing Tools

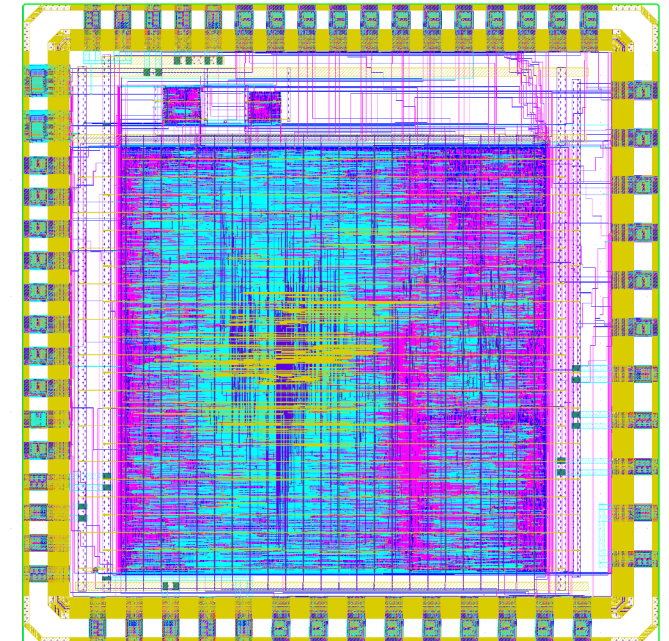
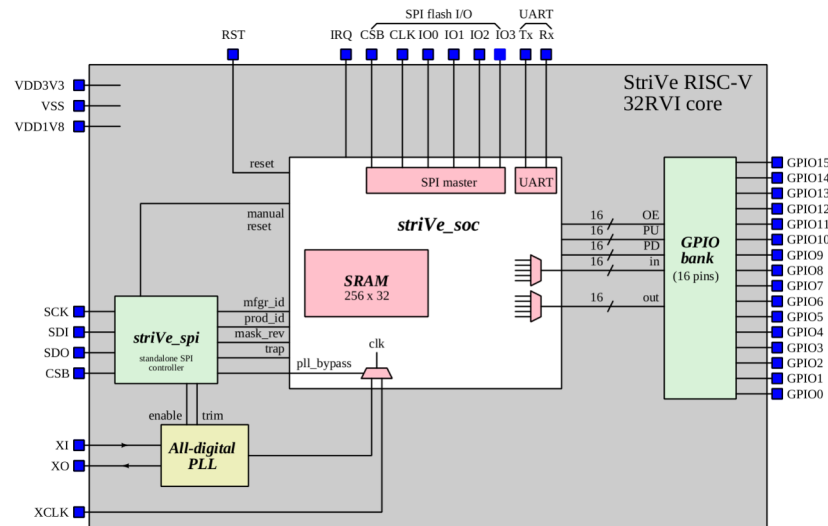
Introducing



openlane.io

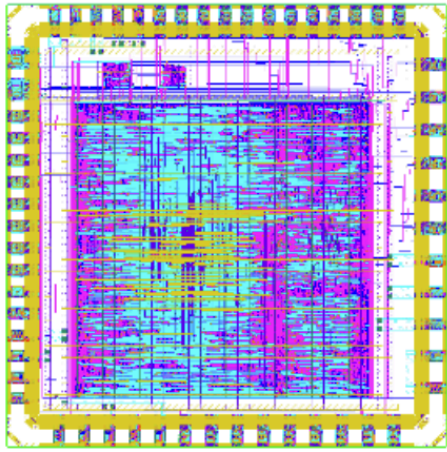
OpenLane

- Started as an Open-Source Flow for a **True** Open Source Tape-out Experiment
- stiVe is a family of **open everything** SoCs
 - Open PDK, Open EDA, Open RTL



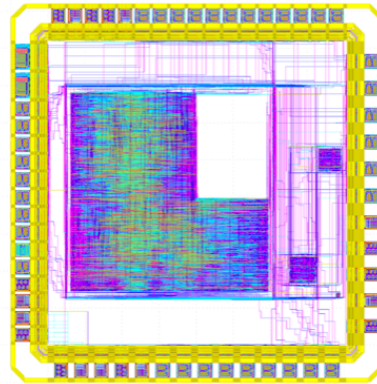
SkyWater sky130 - 130nm

striVe



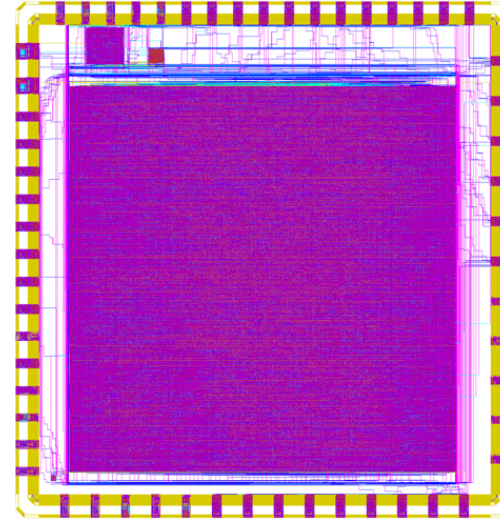
2.743 x 2.751 mm²

striVe2



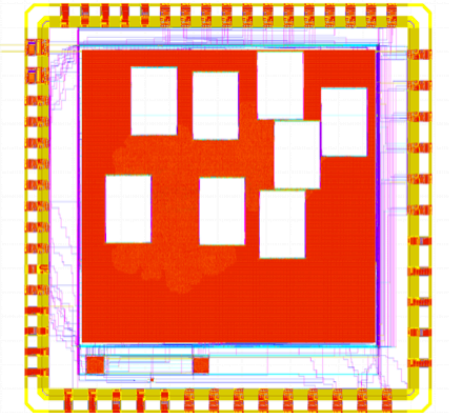
2.013 x 2.042 mm²

striVe3



4.301 x 4.451 mm²

striVe5



3.373 x 3.4 mm²



striVe SoC Family

Mohamed Kassem

Tuesday, August 25, 2020

16:00 GMT



OpenLane ASIC Flow

- Main Goal:
Produce a clean GDSII with no human intervention (no-human-in-the-loop)
- Clean means:
 - No LVS Errors
 - No DRC Errors

OpenLane ASIC Flow

- Main Goal:
Produce a clean GDSII with no human intervention (no-human-in-the-loop)
- Tuned for SkyWater 130nm Open PDK
 - Also, supports XFAB180 and GF130G

OpenLane ASIC Flow

- Main Goal:
Produce a clean GDSII with no human intervention (no-human-in-the-loop)
- Tuned for SkyWater 130nm Open PDK
- Containerized
 - Functional out of the box
 - Instructions to build and run natively will follow

OpenLane ASIC Flow

- Can be used to harden Macros and Chips
 - Full Chip (SoC or ASIC) hardening support is not released pending the availability of a public IO Library

OpenLane ASIC Flow

- Can be used to harden Macros and Chips
- Two modes of operation:
 - Autonomous or Interactive

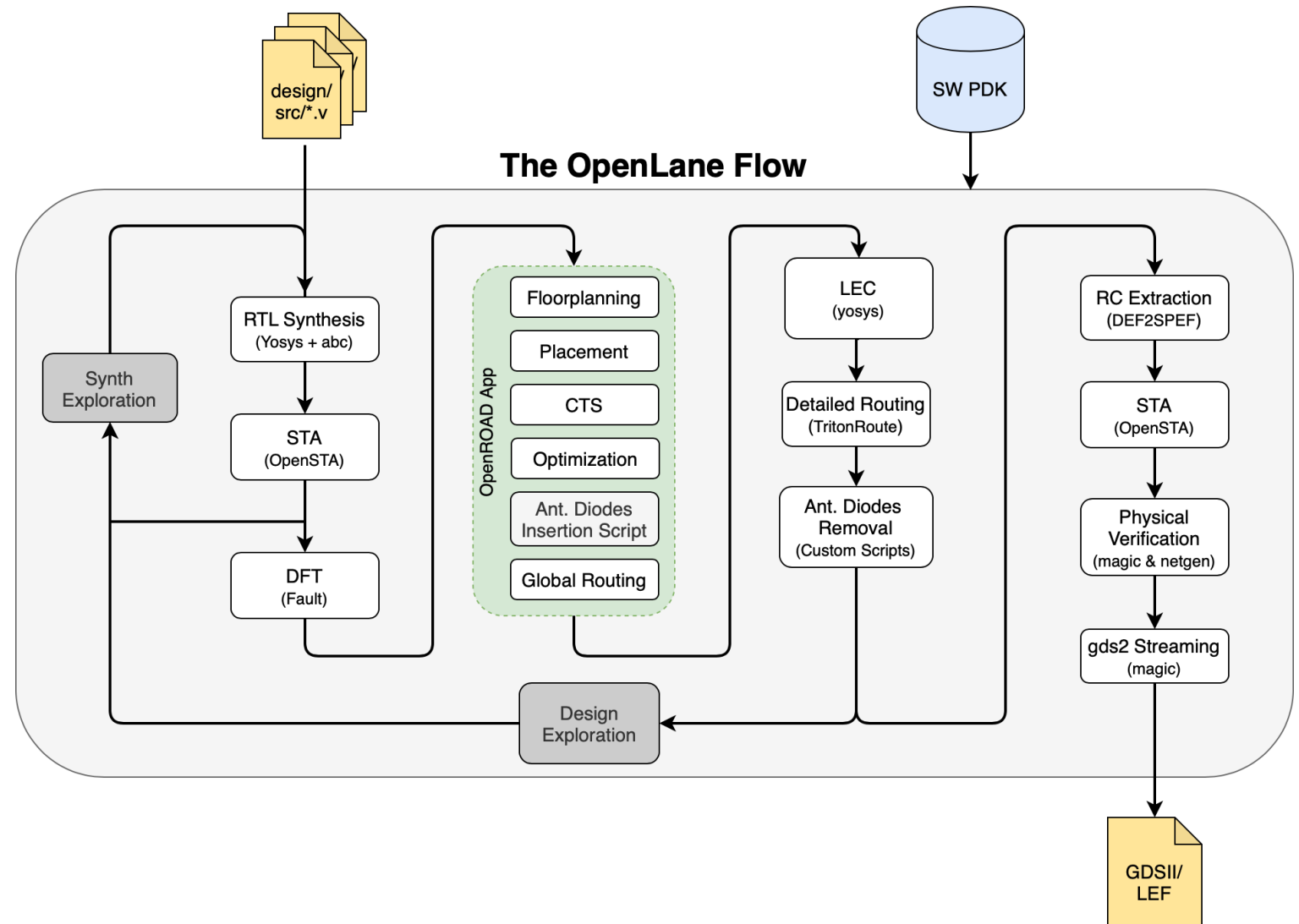
OpenLane ASIC Flow

- Can be used to harden Macros and Chips
- Two modes of operation:
- Design Space Exploration:
 - Find the best set of flow configurations

OpenLane ASIC Flow

- Can be used to harden Macros and Chips
- Two modes of operation:
- Design Space Exploration:
- Large number of design examples
 - 43 designs with their best configurations
 - More will be added soon

OpenLane ASIC Flow

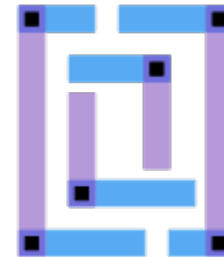


OpenLane ASIC Flow

OpenROAD



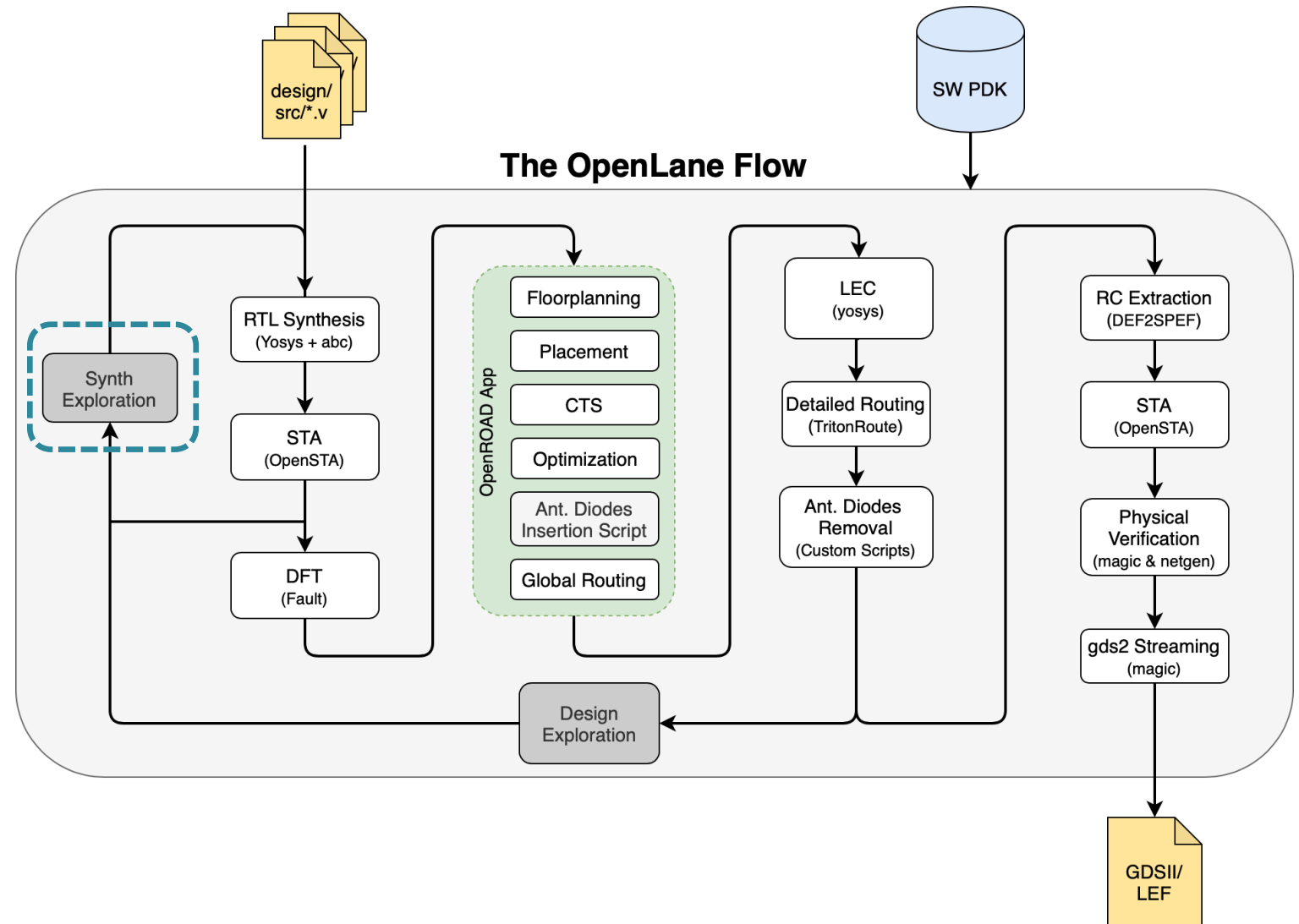
Yosys



QFlow

ABC

OpenLane ASIC Flow



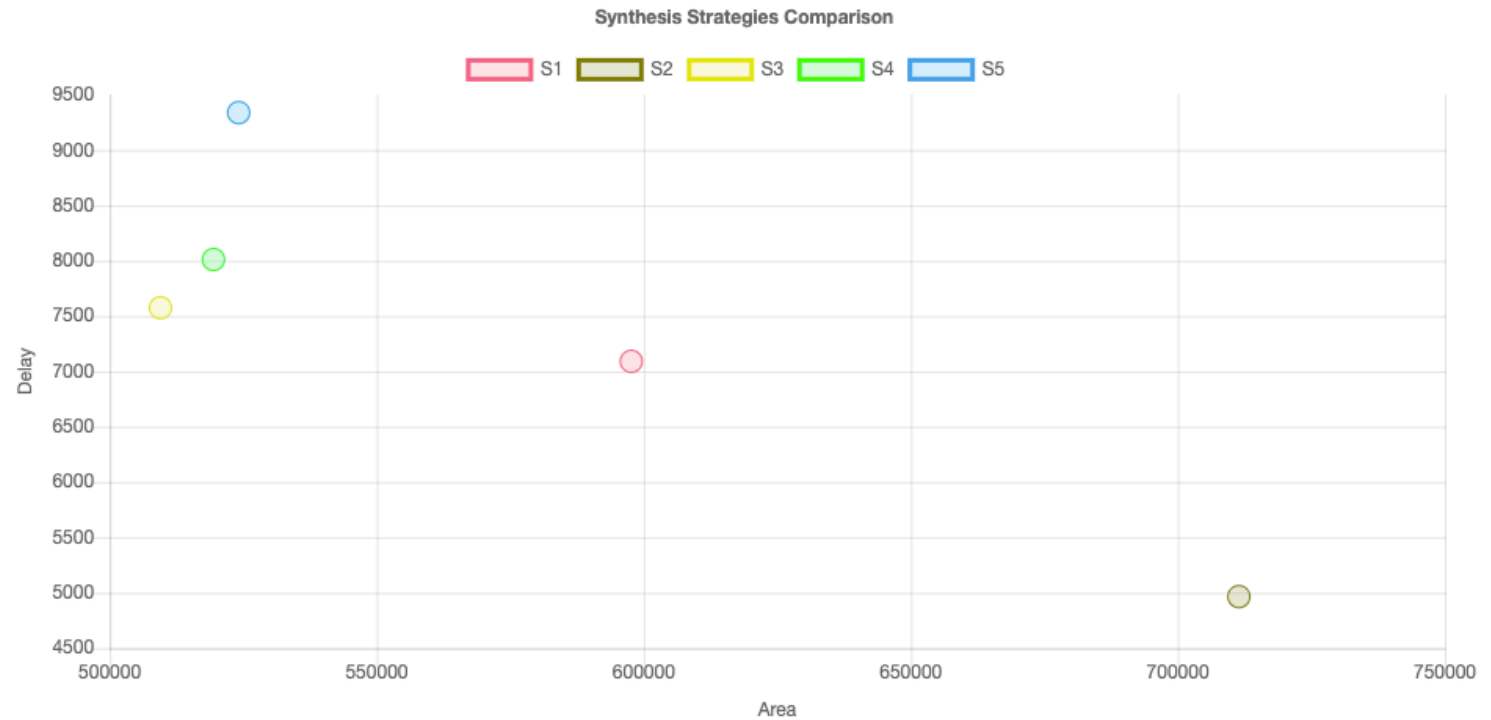
Synthesis Exploration

- Yosys is used for RTL synthesis
- ABC is used for logic synthesis and technology mapping
- ABC needs a script that defines the sequence of optimization operations
- OpenLane comes with several synthesis scripts
 - We call them “Synthesis Strategies”
 - Synthesis Exploration can help picking the best strategy for a given design

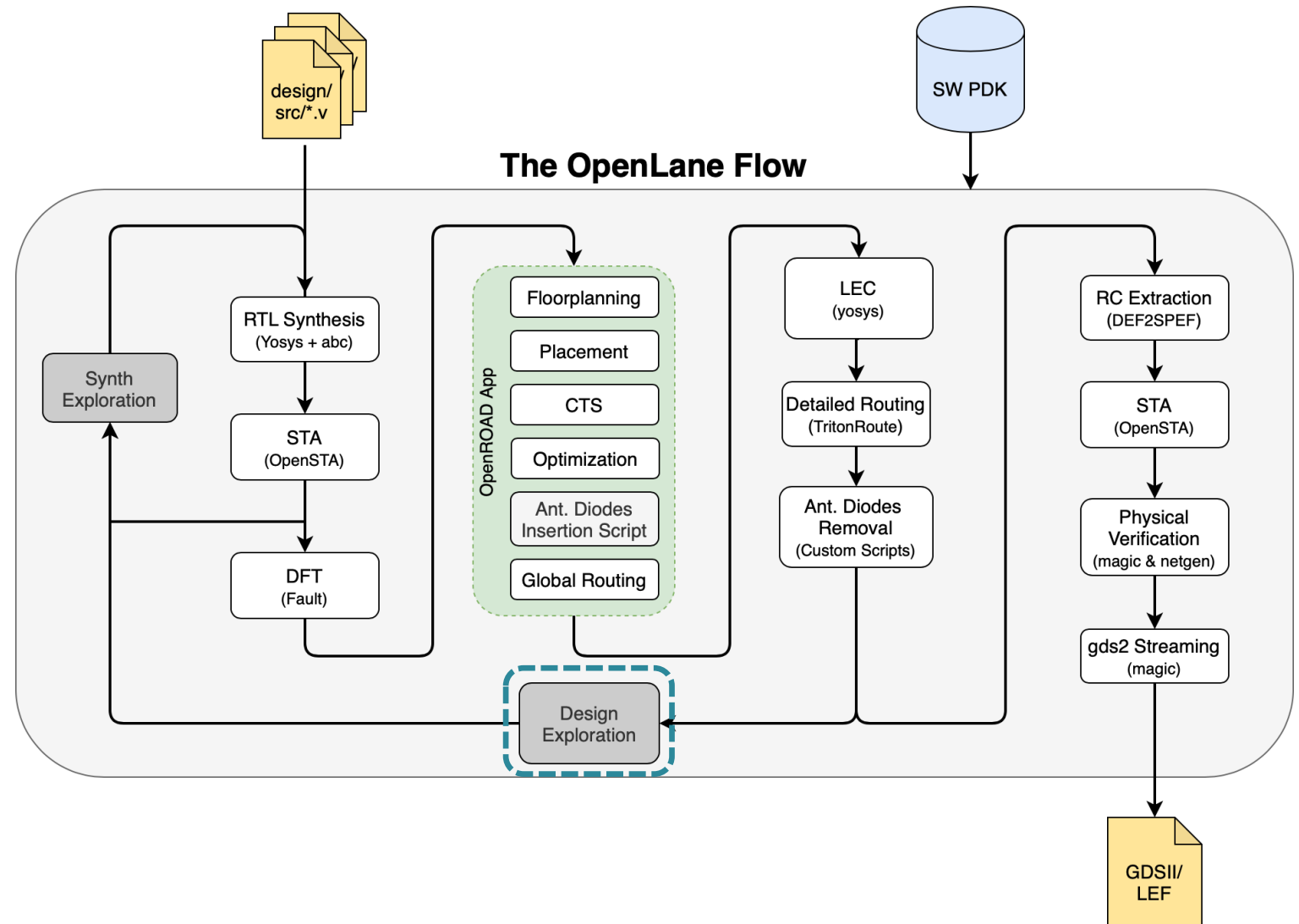
Synthesis Exploration

Best Area	Best Gate Count	Best Delay
509503.00	17257	4964.82
scpt_2	scpt_3	scpt_1

Startegy	Gate Count	Area (um^2)	Delay (ps)	Gates Ratio	Area Ratio	Delay Ratio
S1: scpt_0	18471	597665.00	7091.03	1.07	1.173	1.428
S2: scpt_1	23137	711444.00	4964.82	1.34	1.396	1
S3: scpt_2	19940	509503.00	7574.80	1.155	1	1.525
S4: scpt_3	17257	519434.00	8011.78	1	1.019	1.613
S5: scpt_4	18896	524152.00	9338.71	1.094	1.028	1.88



OpenLane ASIC Flow



Design Exploration

- OpenLane has 16 design specific configurations
- Not all combinations may result in a DRC clean layout
- OpenLane can sweep different parameters to help with that
- For each run, OpenLane collects around 35 different design metrics

Design Exploration

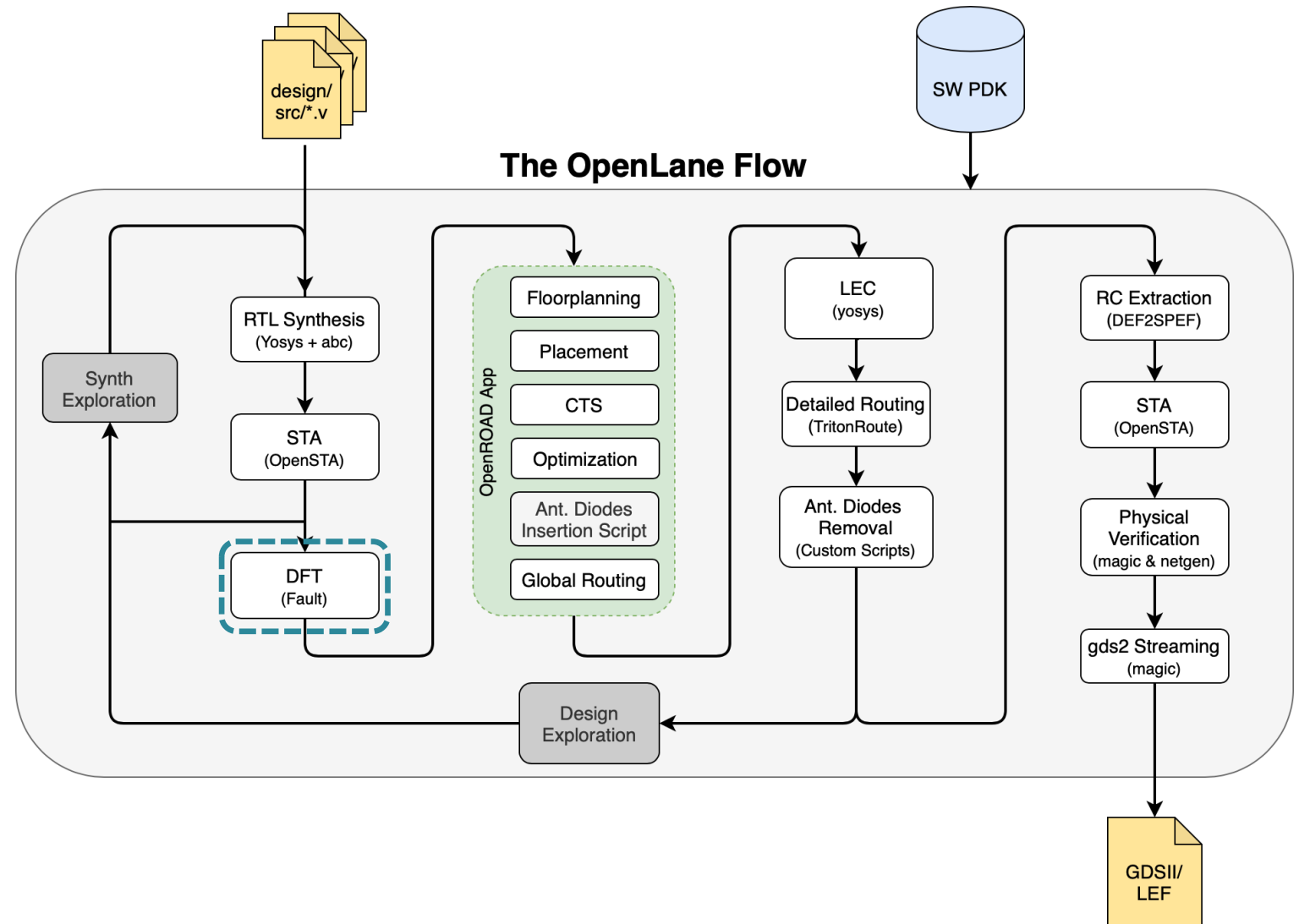
Design	Runtime	Cell Count	TR Vios	FP_CORE_UTIL	ROUTING_STRATEGY	GLB_RT_ADJUSTMENT
aes	1h29m8s	22932	1	40	1	0.05
aes	1h34m31s	22932	2	30	1	0.05
aes	1h41m14s	22932	9	40	1	0.05
aes	1h47m14s	22932	1	45	1	0.05
aes	1h44m14s	22932	1	40	1	0.05
aes	1h47m59s	22932	1	45	1	0.05
aes	1h49m7s	22932	1	45	1	0.05
aes	1h43m54s	22932	2	30	1	0.05
aes	1h42m58s	22932	8	30	1	0.05
cordic	0h10m51s	8275	0	45	0	0.15
cordic	0h10m35s	8275	0	45	0	0.15
cordic	0h9m55s	8275	2	40	0	0.15
cordic	0h11m25s	8275	0	45	0	0.15
cordic	0h10m3s	8275	0	30	0	0.15
cordic	0h11m6s	8275	4	40	0	0.15
cordic	0h11m3s	8275	4	40	0	0.15
cordic	0h10m25s	8275	0	30	0	0.15
cordic	0h10m26s	8275	3	30	0	0.15

OpenLane Regression

- The design exploration utility is, also, used for regression testing
- We run OpenLane on ~70 designs and compare the results to the best known ones

Design	Runtime	Cell COunt	TR Vios
jpeg_encoder	3h16m7s	73624	0
striVe_soc	3h14m0s	73271	0
aes256	1h35m51s	64435	0
genericfir	1h2m36s	48849	0
aes128	1h7m50s	44658	0
TEA	2h11m8s	44026	0
rc6_core	1h43m44s	35304	0
double_sqrt	1h14m18s	29252	0
iir5sfix	1h14m5s	24950	0
y_huff	0h54m48s	16826	0
sha3	0h21m18s	16372	0
ocs_blitter	0h17m48s	10997	0
sub86	0h12m35s	7655	0
CPU	0h11m7s	7342	0
cordic	0h10m13s	7210	0

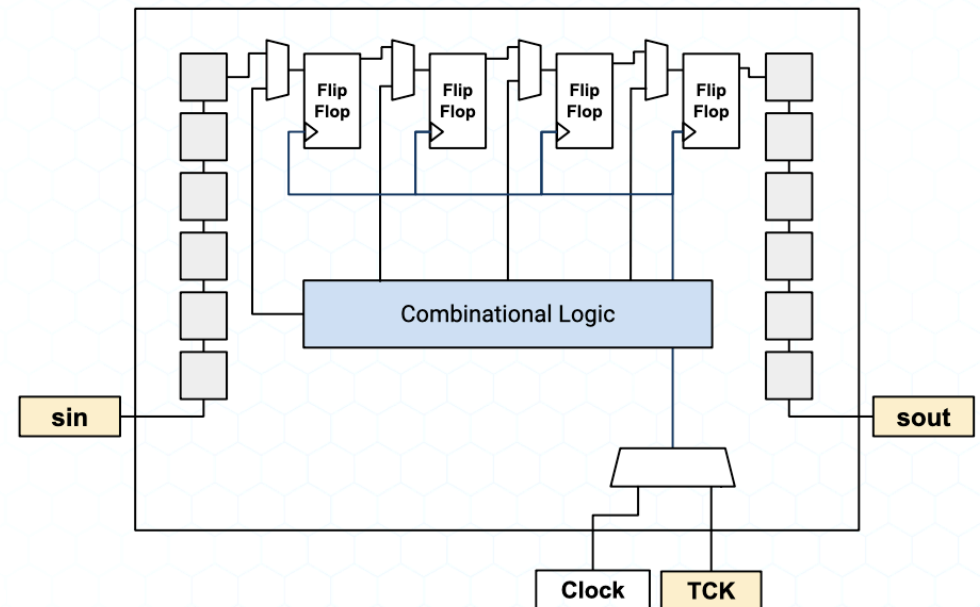
OpenLane ASIC Flow



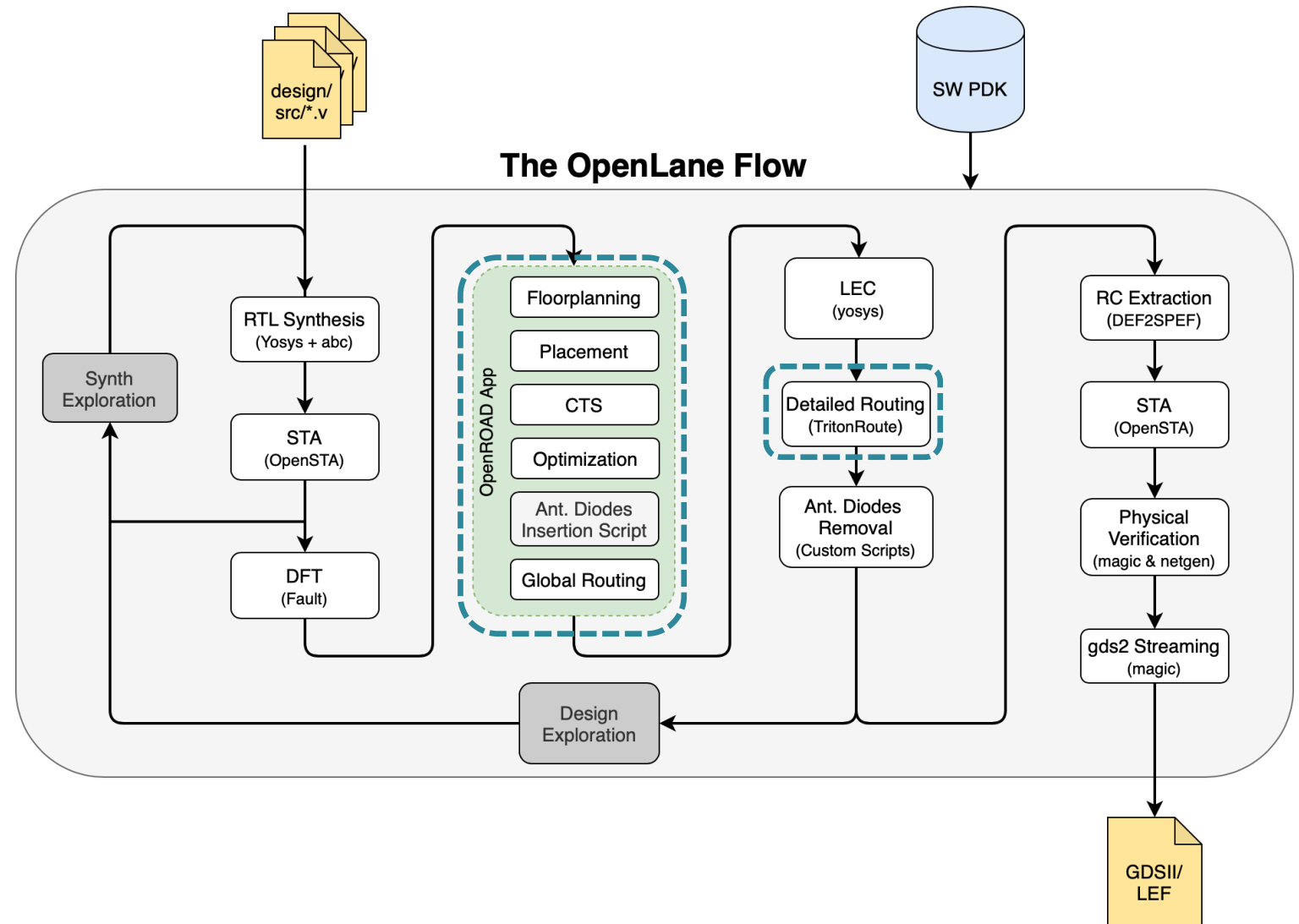
Design for Test (DFT)

Fault

- Scan Insertion
- Automatic Test Pattern Generation (ATPG)
- Test Patterns Compaction
- Fault Coverage
- Fault Simulation



OpenLane ASIC Flow

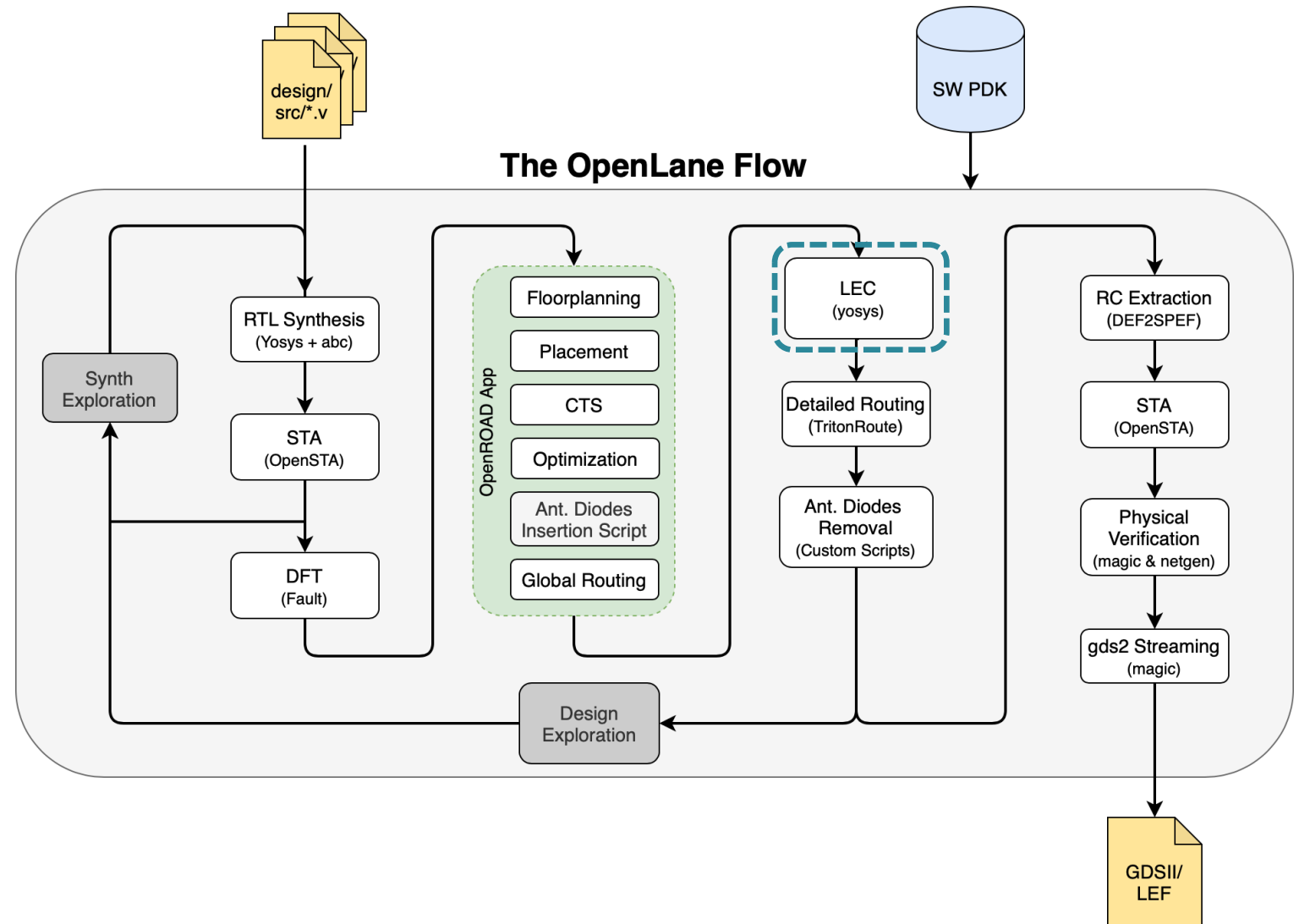


Physical Implementation

OpenROAD

- Also called automated PnR (Place and Route)
- Floor/Power Planning
- End Decoupling Capacitors and Tap cells insertion
- Placement: Global and Detailed
- Post placement optimization
- Clock Tree Synthesis (CTS)
- Routing: Global and Detailed

OpenLane ASIC Flow

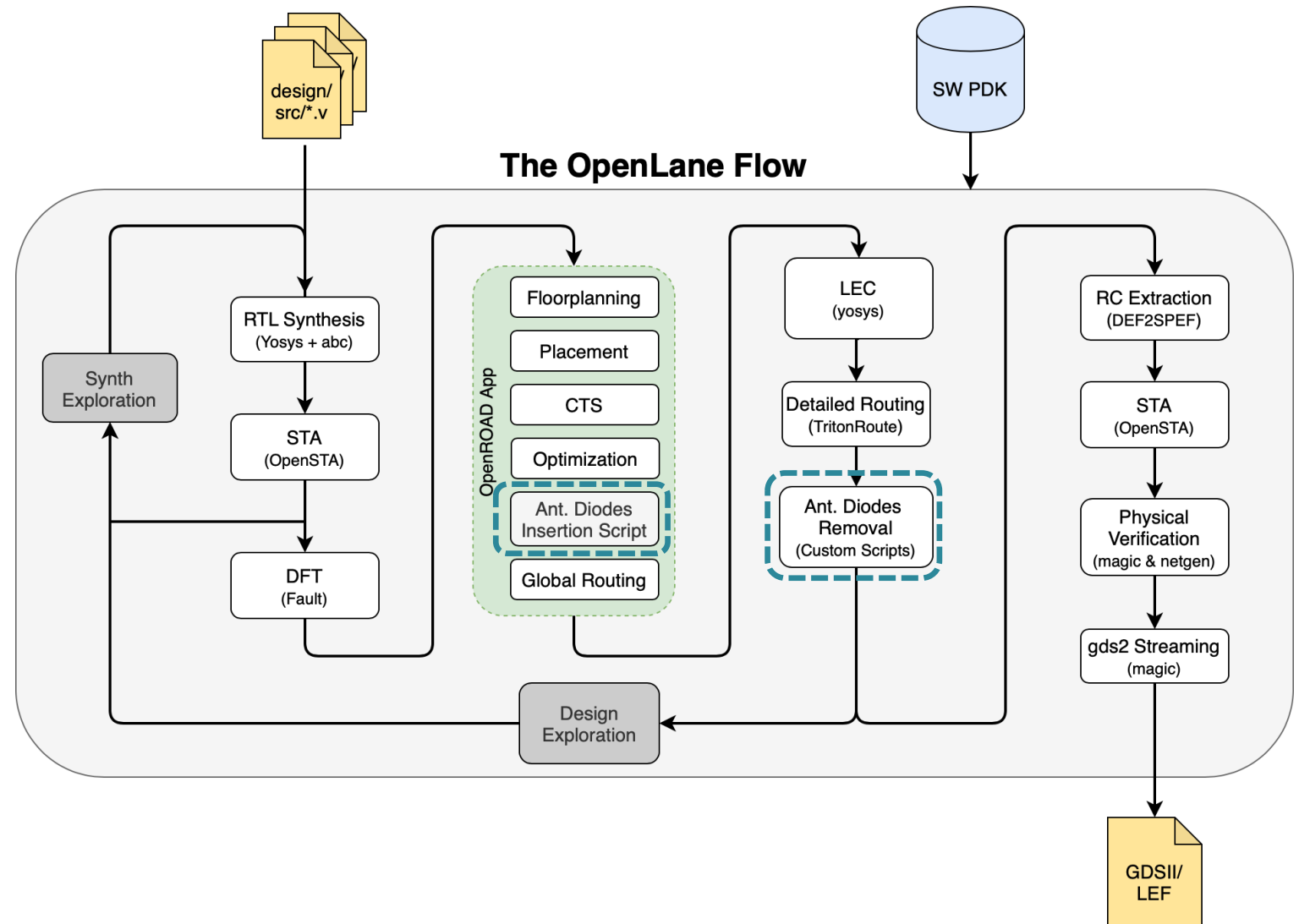


Logic Equivalence Check (LEC)

Yosys

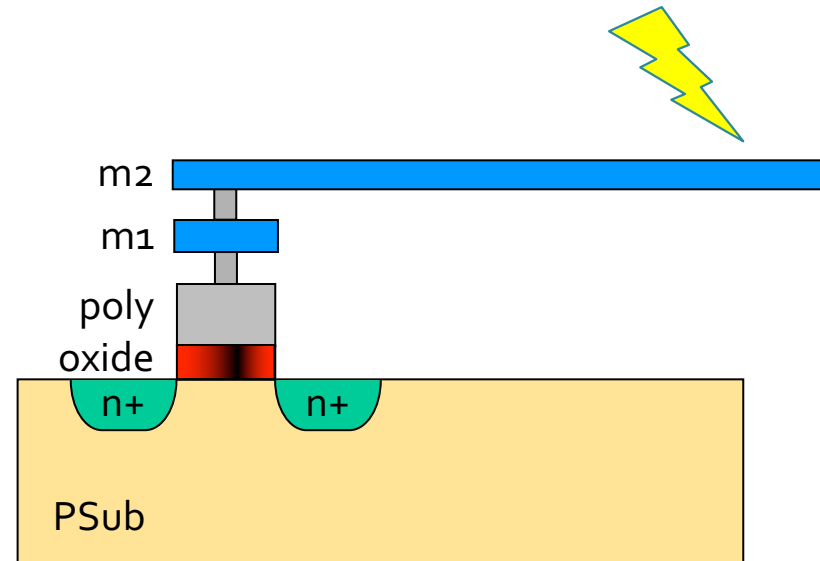
- Every time the netlist is modified (ECO), verification must be performed
 - CTS modifies the netlist
 - Post Placement optimizations modifies the netlist
- LEC is used to formally confirm that the function did not change by modifying the netlist

OpenLane ASIC Flow



Dealing with Antenna Rules Violations

- When a metal wire segment is fabricated, it can act as an antenna.
 - Reactive ion etching causes charge to accumulate on the wire.
 - Transistor gates can be damaged during fabrication



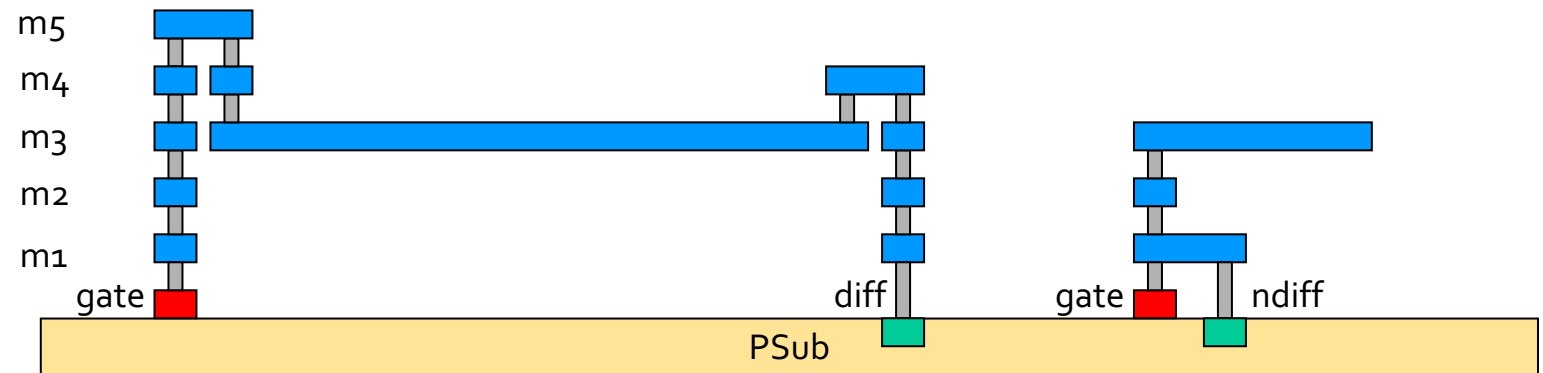
Dealing with Antenna Rules Violations

- Two solutions:
 - Bridging attaches a higher layer intermediary
 - Requires Router awareness (not there yet!)



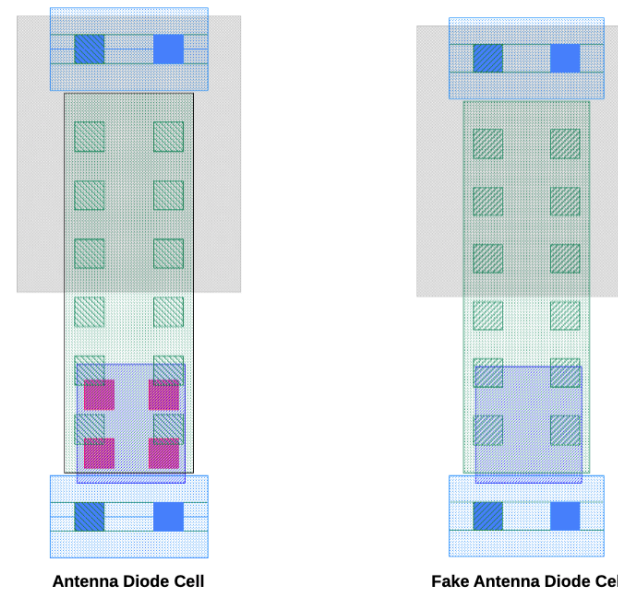
Dealing with Antenna Rules Violations

- Two solutions:
 - Bridging attaches a higher layer intermediary
 - Requires Router awareness (not there yet!)
 - Add antenna diode cell to leak away charges
 - Antenna diodes are provided by the SCL

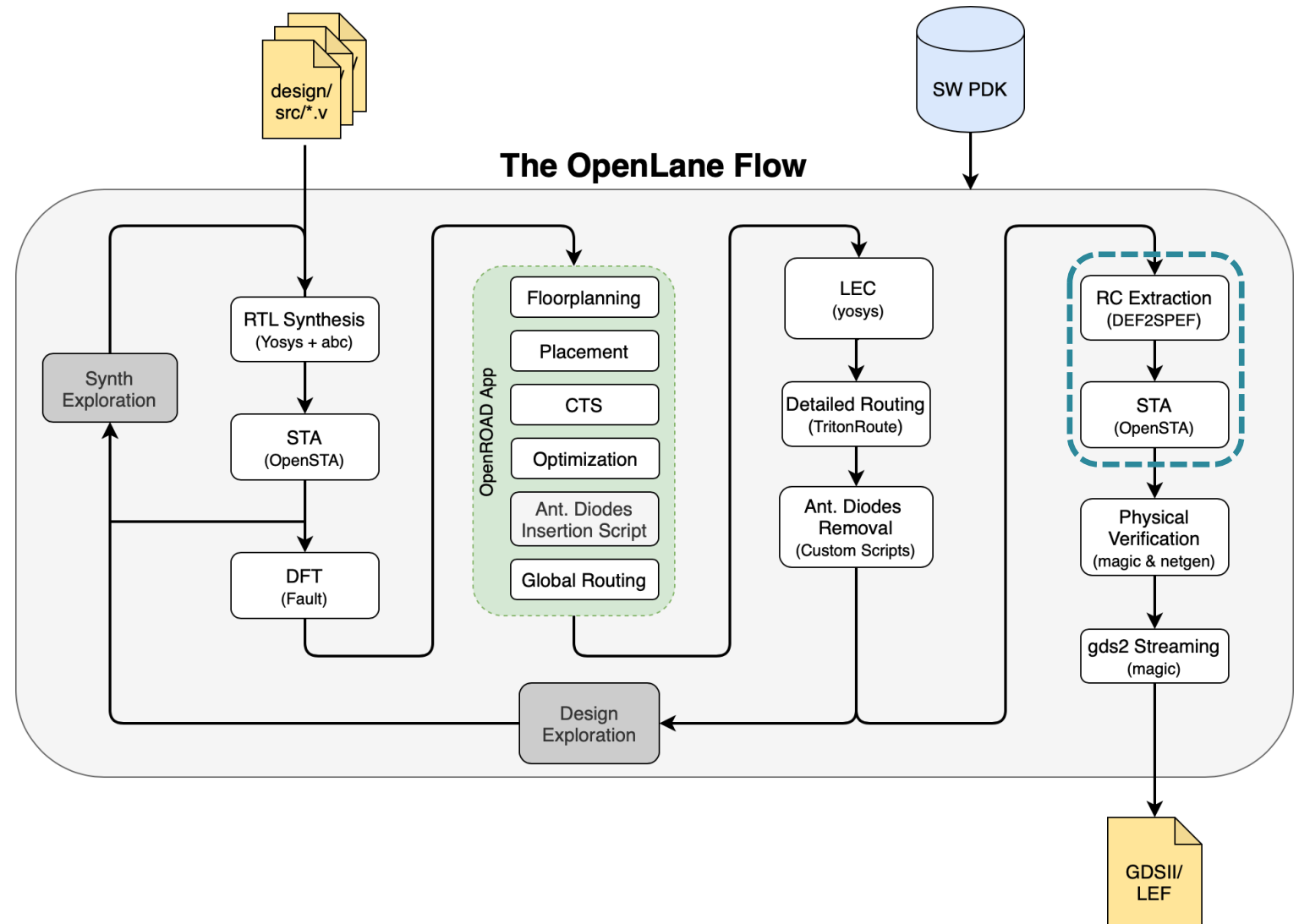


Dealing with Antenna Rules Violations

- We took a preventive approach
 - Add a Fake Antenna Diode next to every cell input after placement
 - Run the Antenna Checker (Magic) on the routed layout
 - If the checker reports a violation on the cell input pin, replace the Fake Diode cell by a real one



OpenLane ASIC Flow



Static Timing Analysis

- RC Extraction: DEF2SPEF
- STA: OpenSTA (OpenROAD)

```
Startpoint: _5435_ (rising edge-triggered flip-flop clocked by vclk)
Endpoint: _5435_ (rising edge-triggered flip-flop clocked by vclk)
Path Group: vclk
Path Type: min

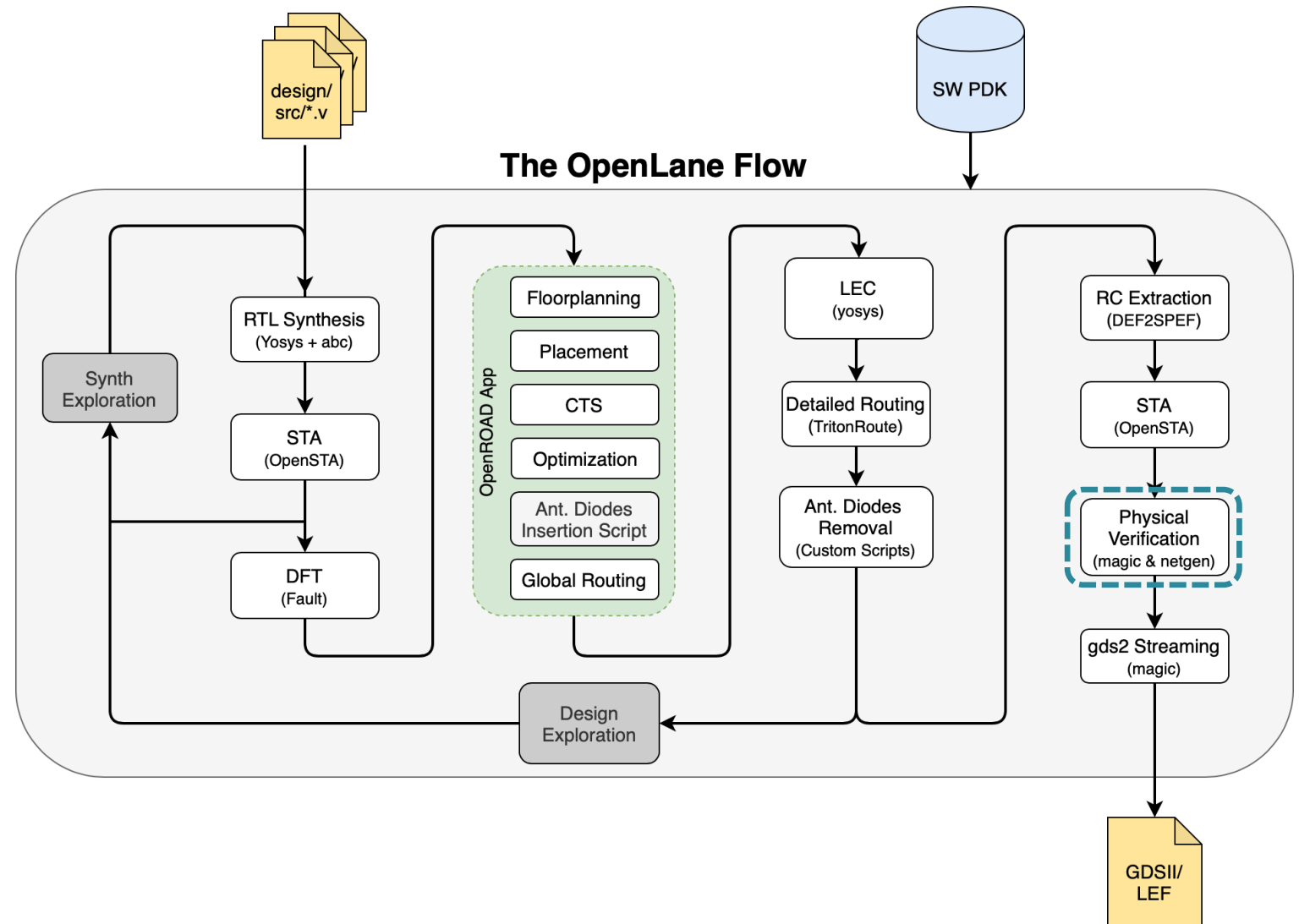
Delay    Time    Description
-----
0.00     0.00    clock vclk (rise edge)
0.00     0.00    clock network delay (ideal)
0.00     0.00    ^ _5435_/CLK (sky130_fd_sc_hd__dfrrtp_4)
0.78     0.78    ^ _5435_/Q (sky130_fd_sc_hd__dfrrtp_4)
0.22     0.99    ^ _4647_/X (sky130_fd_sc_hd__a21bo_4)
0.00     0.99    ^ _5435_/D (sky130_fd_sc_hd__dfrrtp_4)
0.99     0.99    data arrival time

0.00     0.00    clock vclk (rise edge)
0.00     0.00    clock network delay (ideal)
0.00     0.00    clock reconvergence pessimism
0.00     0.00    ^ _5435_/CLK (sky130_fd_sc_hd__dfrrtp_4)
-0.08    -0.08    library hold time
-0.08    -0.08    data required time

-----
-0.08    data required time
-0.99    data arrival time
-----

1.08     slack (MET)
```

OpenLane ASIC Flow



Physical Verification DRC & LVS

- Magic is used for Design Rules Checking and SPICE Extraction from Layout
- Netgen is used for LVS
 - Extracted SPICE by Magic vs. Verilog netlist

OpenLane Jump Start

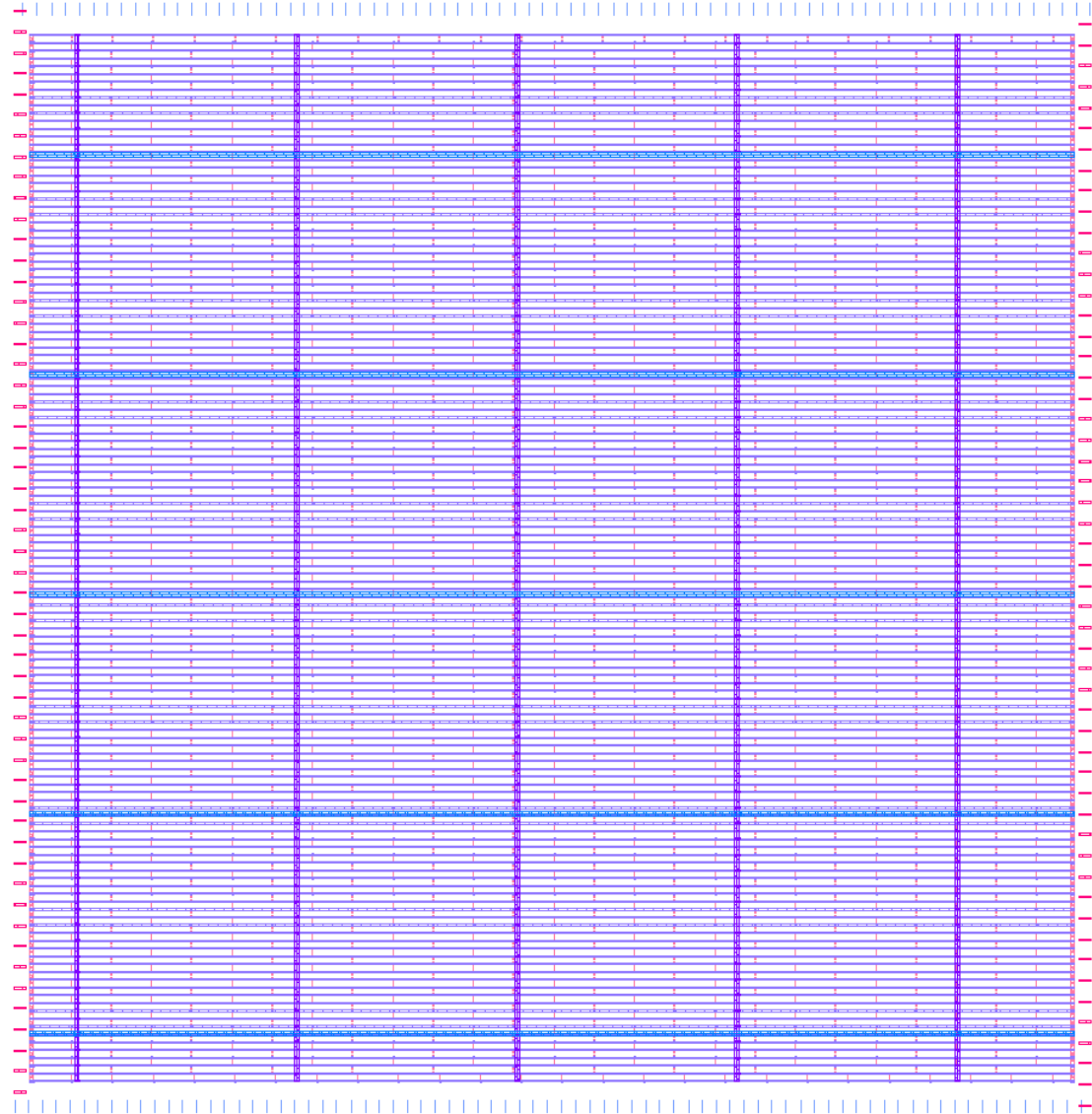
- Dependencies (Check GH for the versions):
 - Docker
 - Magic VLSI Layout tool
- Visit: www.openlane.io
- Follow the instructions to:
 - Setup SkyWater PDK
 - Go for HD SCL if you are just exploring
 - Set up OpenLane
- Start OpenLane Docker Container
 - Don't forget to mount a local folder inside OpenLane container
- To run OpenLane on one of the provided designs (e.g., xtea)
`./flow.tcl -design xtea`

OpenLane Jump Start

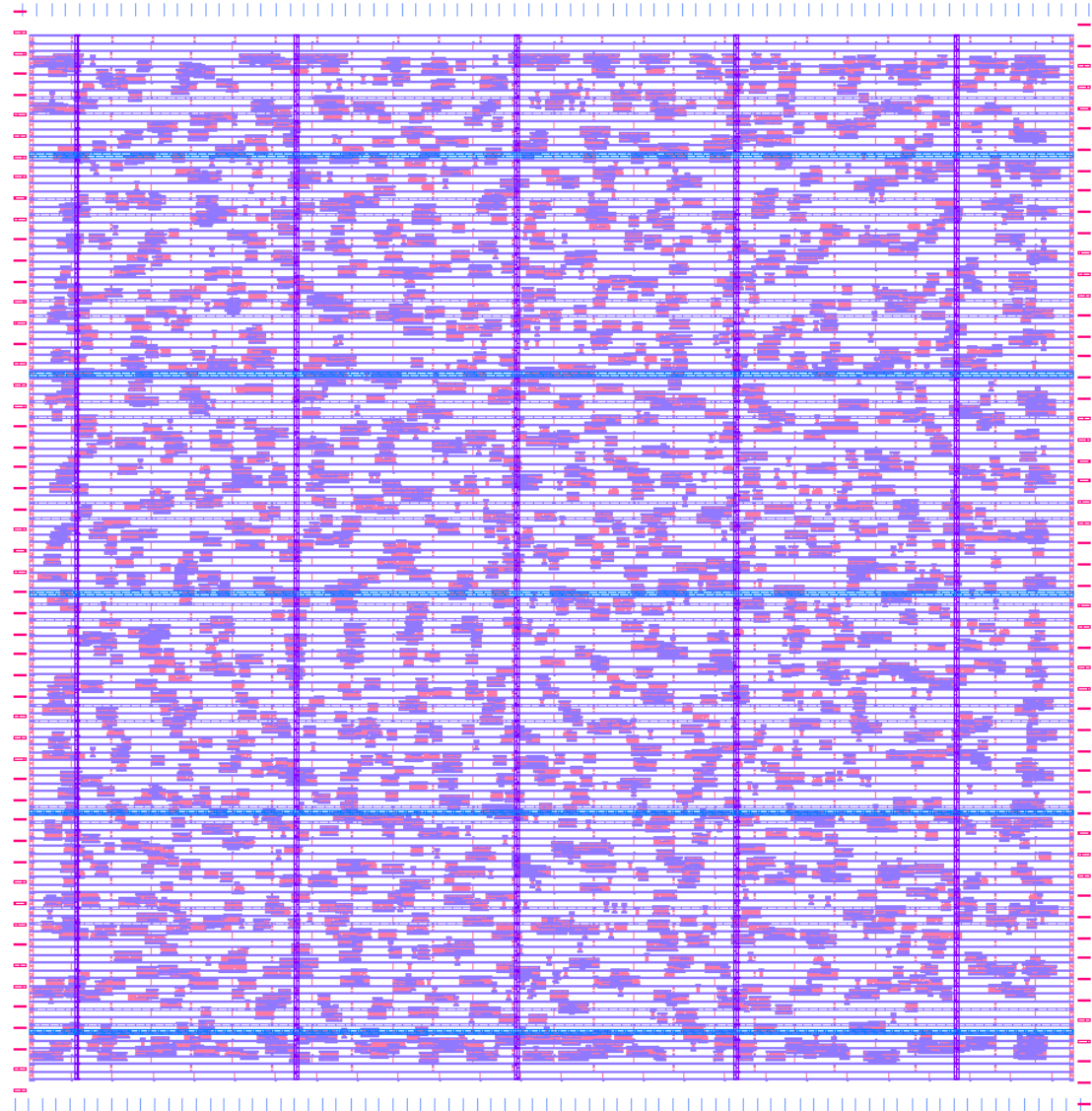
- After few minutes the flow should be done generating a clean GDSII for xtea
- Number of Cells: ~ 3k Cells
- Number of Pins: 262 Pins
- Macro Area: ~ 375x386 μm^2
- Final GDSII is under `results/magic`

```
designs/<xtea>
├─ runs
│   └─ <time_stamp>
│       ├── config.tcl
│       ├── logs
│       ├── cts
│       ├── floorplan
│       ├── magic
│       ├── placement
│       ├── routing
│       └─ synthesis
│   └─ reports
│       ├── cts
│       ├── floorplan
│       ├── magic
│       ├── placement
│       ├── routing
│       └─ synthesis
│   └─ results
│       ├── cts
│       ├── ....
│       └─ tmp
│           ├── cts
│           └─ ....
```

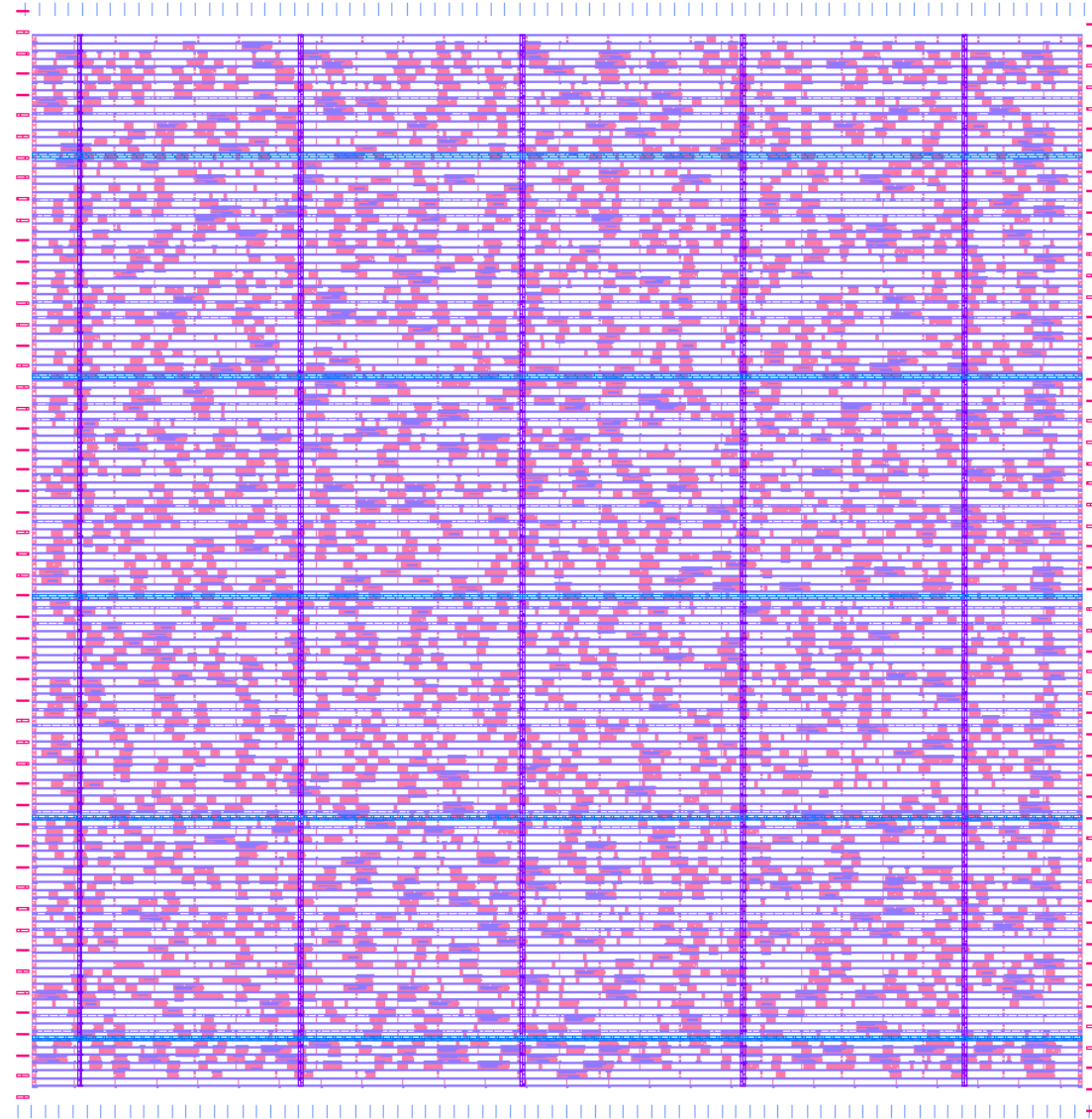
Floor Planning



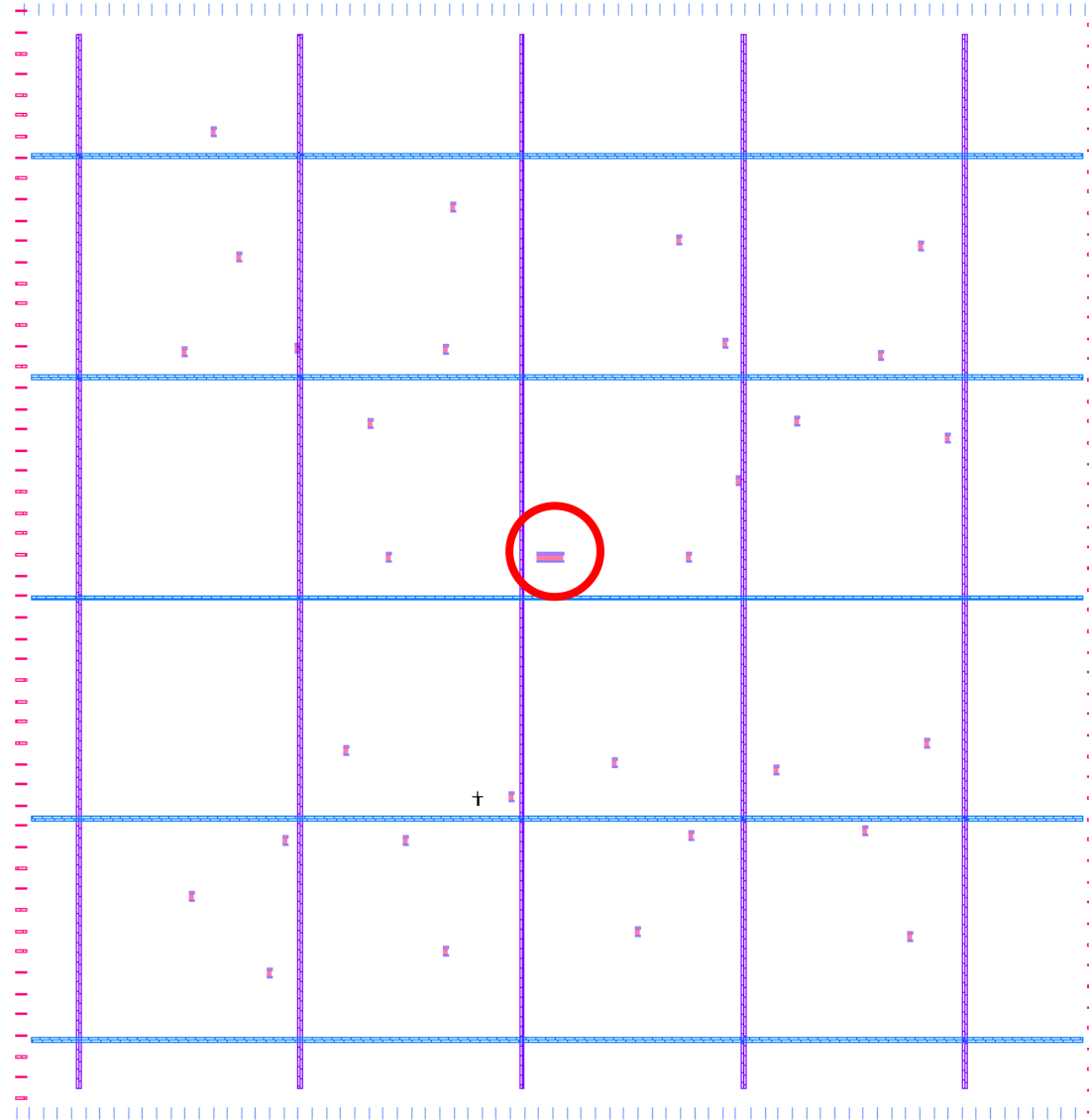
Global Placement



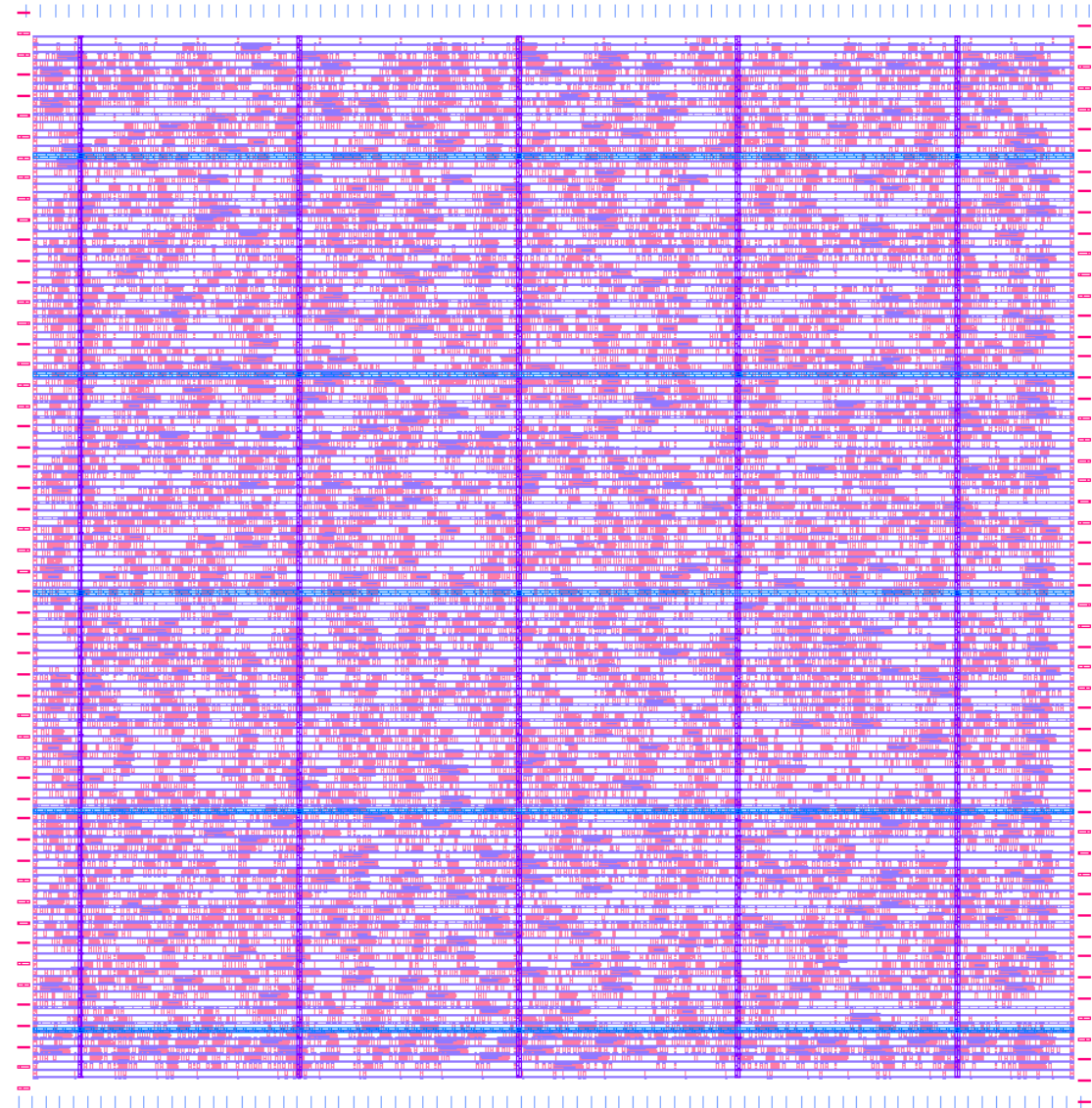
Detailed Placement



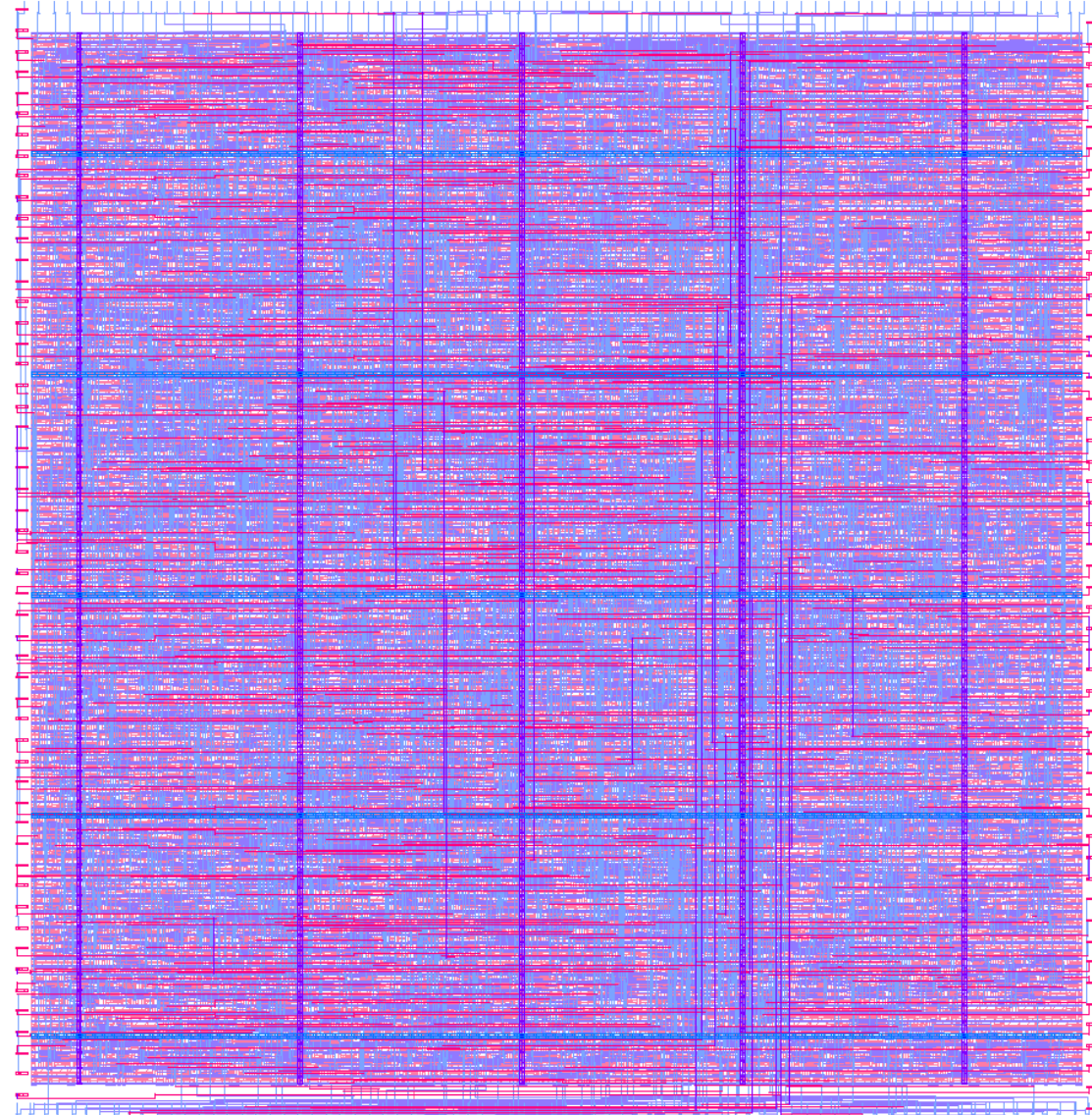
The Clock Tree



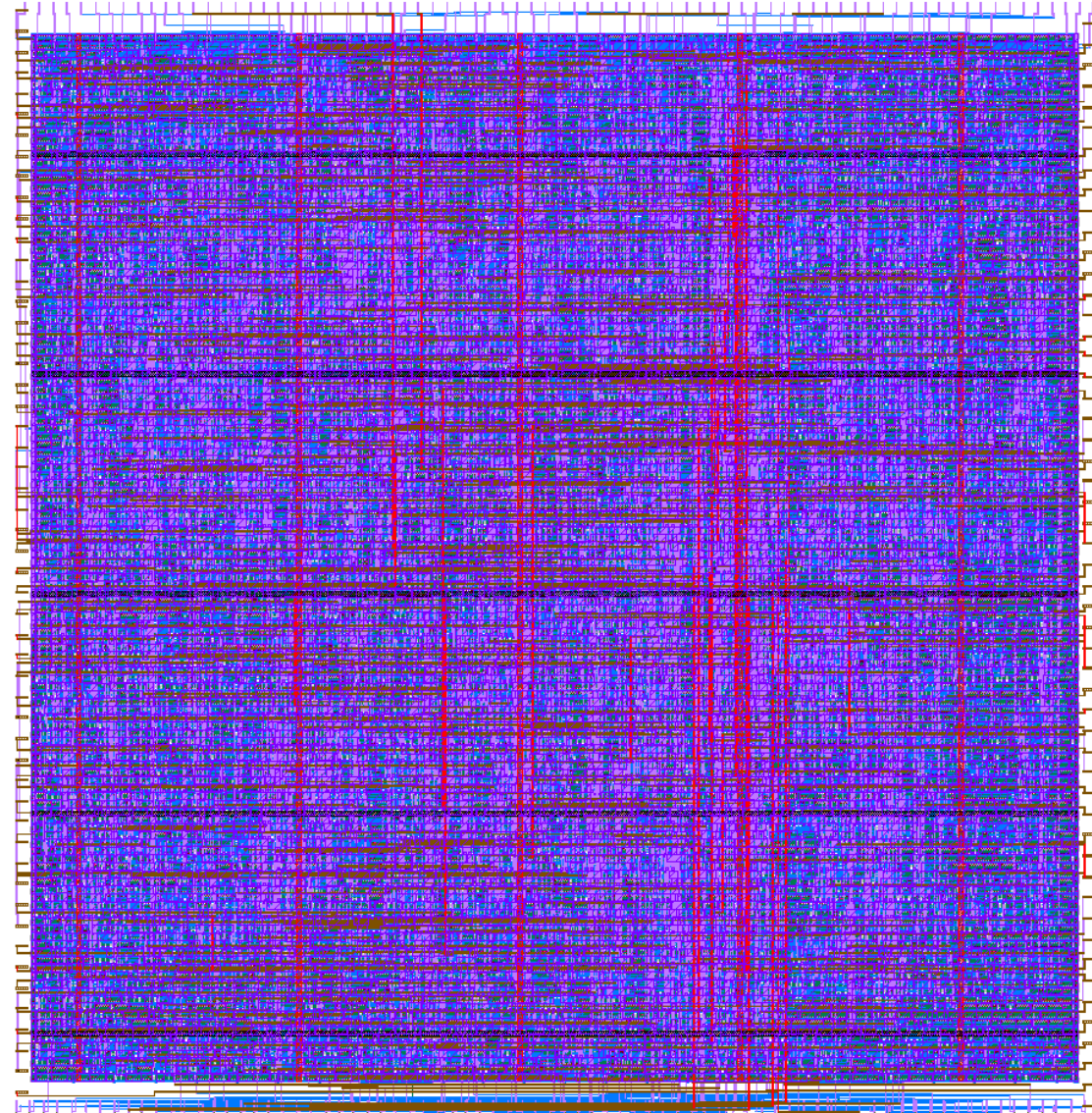
After Diode Insertion



After Detailed Routing



Final GDSII

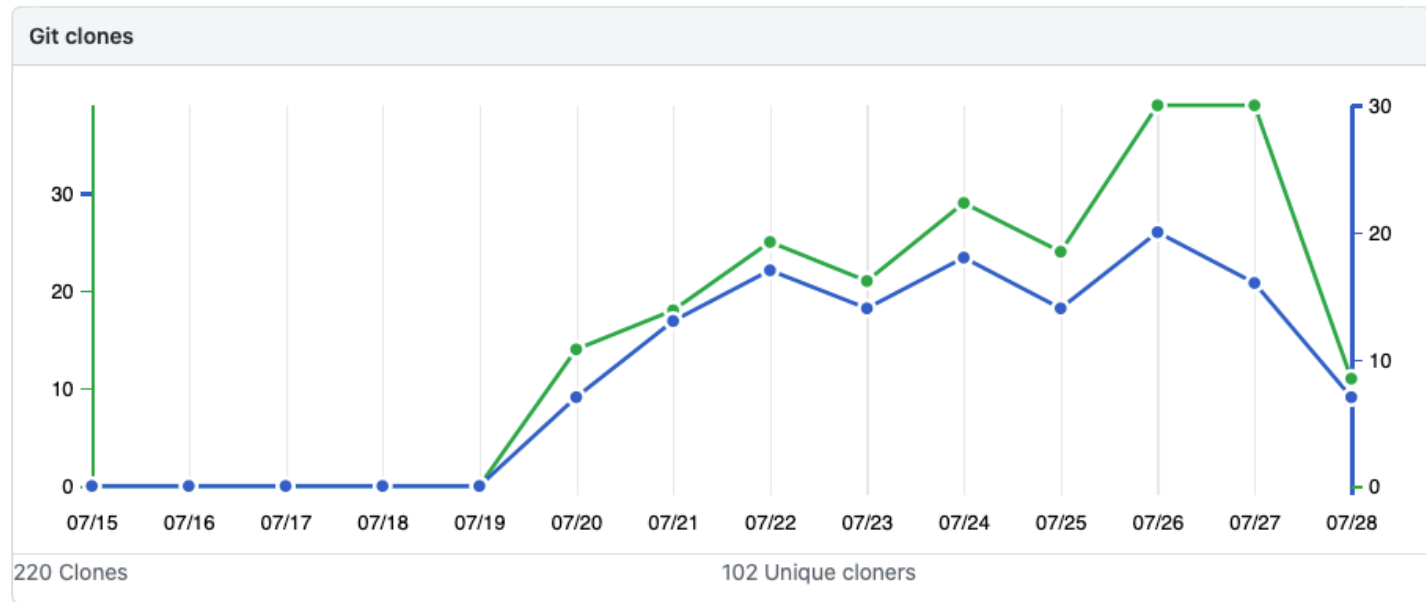


Open Lane Stats

- 16 EDA tools from different projects
- Over 70 scripts and utilities
 - Pre-processing, Optimization, Abstraction of views, etc.
- Over 40 design/flow configurations
 - ~16 are design specific

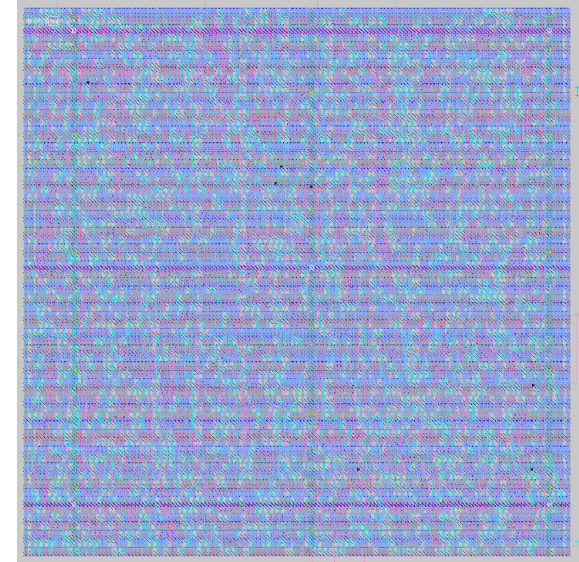
Community Response

- OpenLane repo became public on July 20, 2020
- As of 2:00 PM GMT today:
 - 674 Unique Visitors, 102 Unique Clones
 - 68 Starts
 - 10 Forks
 - 11 Issues, 2 Pull Requests



Community Output

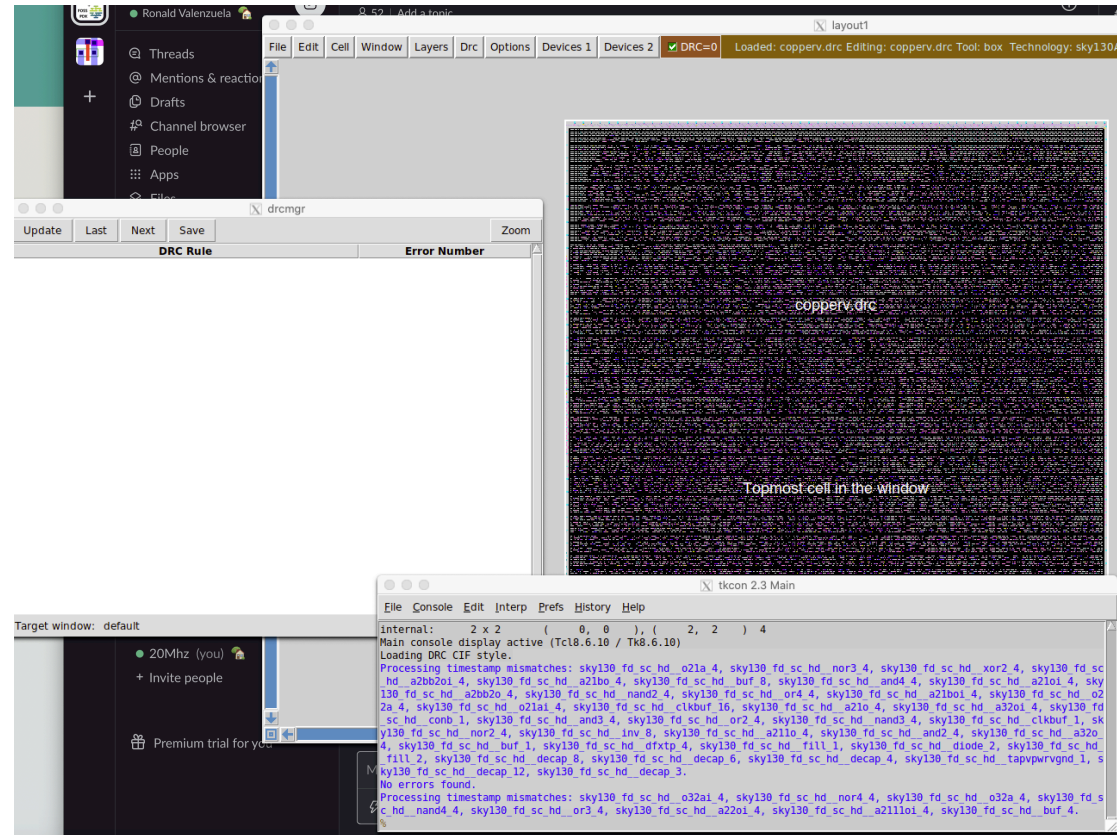
- VGA Clock by Matt Venn
 - <https://github.com/mattvenn/vga-clock>



180x180 μm^2

Community Output

- Hardening Copperv (12K cells RISC-V CPU core) by "20MHz"
- Area: $\sim 0.74 \times 0.75 \text{ mm}^2$



Join The
community

<https://join.slack.com/t/skywater-pdk/signup>

OpenLane Limitations

- Timing Constraints:
 - Limited
- Timing Closure:
 - No post routing optimizations
 - CTS does not help here

OpenLane WIP

- Full Chip Hardening
 - Used to tape out striVe family of SoCs
 - Release is pending I/O Library availability
 - Till then, working on perfecting it
- DFT Support
- Support OSU SCL for SkyWater 130nm
- OpenLane Cloud Runner
 - Submit your Open-Source Design GH repo URL for exploration and hardening

OpenLane WIP

OpenLane Cloud Runner

Automate your design flow using OpenLANE Cloud Runner. The open-source solution that will allow you to deploy, monitor, and modify your OpenLane designs.



[Sign In](#)

[Sign Up](#)

Email Address

Password

☐ Remember me

[Forgot Password?](#)

SIGN IN

 Sign in with Google

 Sign in with GitHub

OpenLane WIP

OpenLane Cloud Runner

Automate your design flow using OpenLANE Cloud Runner. The open-source solution that will allow you to deploy, monitor, and modify your OpenLane designs.

[Sign In](#)

[Sign Up](#)

Email Address

Password

☐ Remember me

[Forgot Password?](#)

SIGN IN

 Sign in with Google

 Sign in with GitHub

OpenLANE Cloud



test account

 Dashboard

 Jobs

 Settings

ADD JOB

Job ID	Design Name	Repository URL	Status	Submission Time	Completion Time	Actions
9	spm	https://github.com/KhaledSoliman/spm	submitted	2020-07-28T12:29:16.750Z		 

Rows per page: 10 1-1 of 1

Copyright © OpenLane Cloud 2020.

OpenLane To do

- Post Routing Timing Closure
- Better Support for Timing Constraints
- Automatic Design Configuration
- Evaluation of other EDA tools
 - E.g., OpenTimer, CU-GR, DR-CU, ...

Meet The Team



Ahmed Ghazy



Kareem Farid



Amr Gouhar

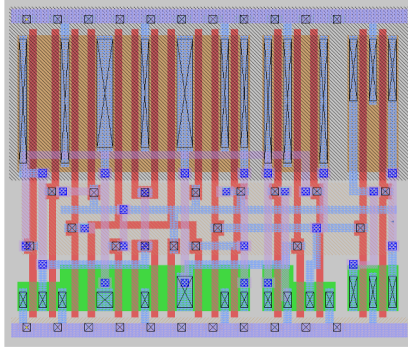
striVe SoC Family

Mohamed Kassem

Tuesday, August 25, 2020

16:00 GMT





Designing new cells for SkyWater 130nm

James Stine

Tuesday, September 22, 2020

16:00 GMT





OpenRAM on SkyWater 130nm

Matt Guthaus

Tuesday, October 20, 2020

16:00 GMT

UC SANTA CRUZ



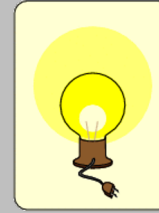


Magic for DRC on SkyWater 130nm

Tim Edwards

Tuesday, November 17, 2020

16:00 GMT



efabless.com



Questions?