iEDA & AiEDA

Open-source BoF Meeting

Xingquan Li

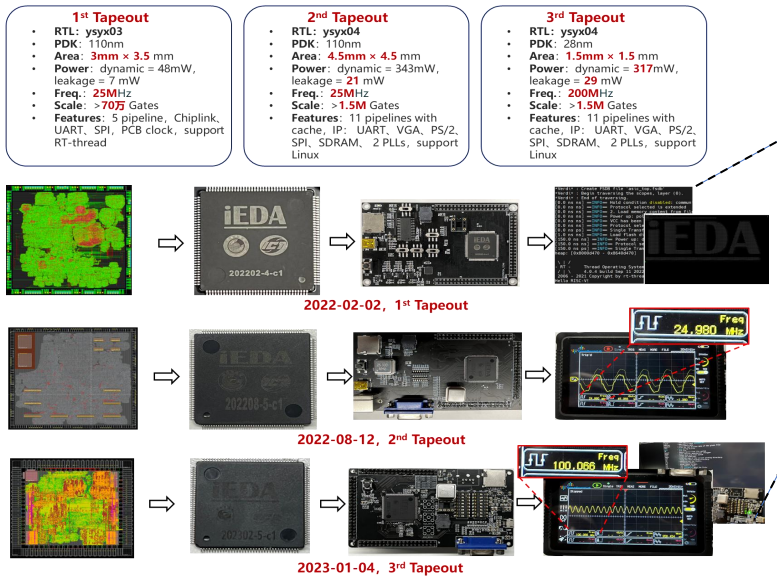
iEDA

01 iEDA

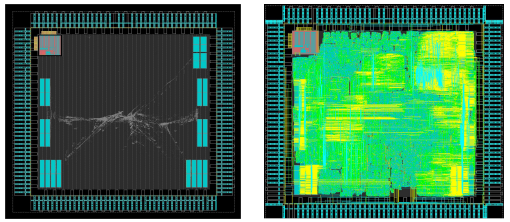
02 AiEDA

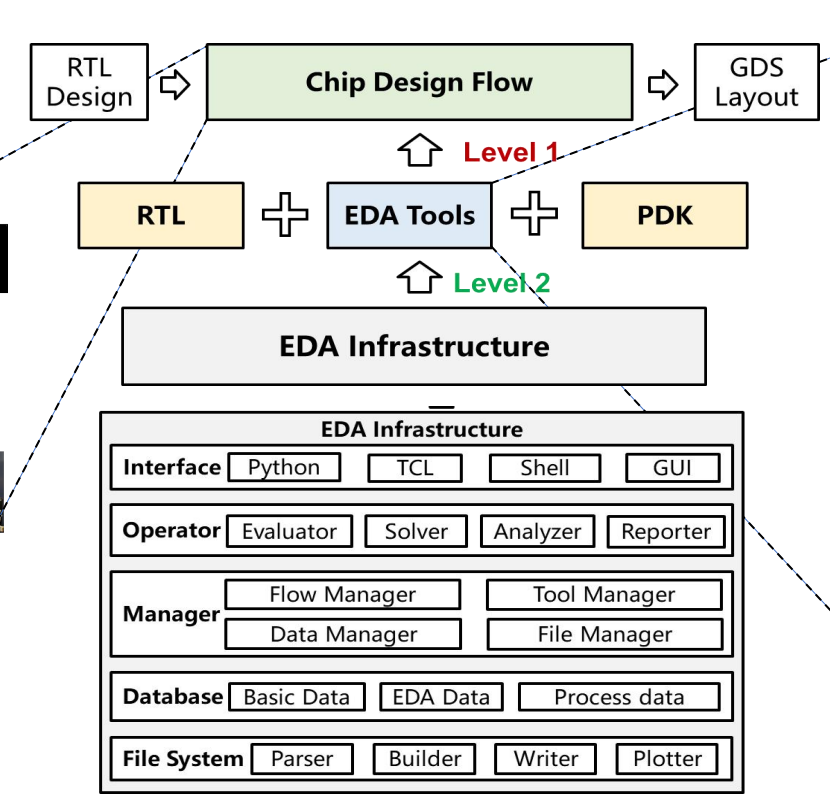
iEDA Overview

- 1 EDA Infrastructure、10 EDA Tools、3 times tape-out design by iEDA
  - Level 1: Open-source EDA, RTL, PDK, supporting chip design;
  - Level 2: Open-source Infrastructure supports EDA development and research
- Open-source: (search "iEDA" on Gitee/Github/Gitlink): https://github.com/OSCC-Project/iEDA

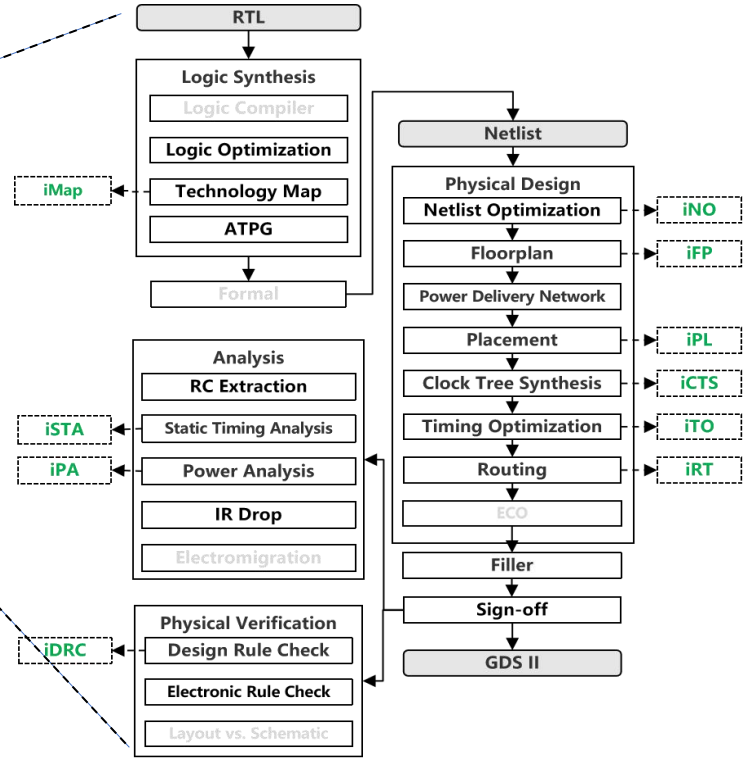


3 times tape-out





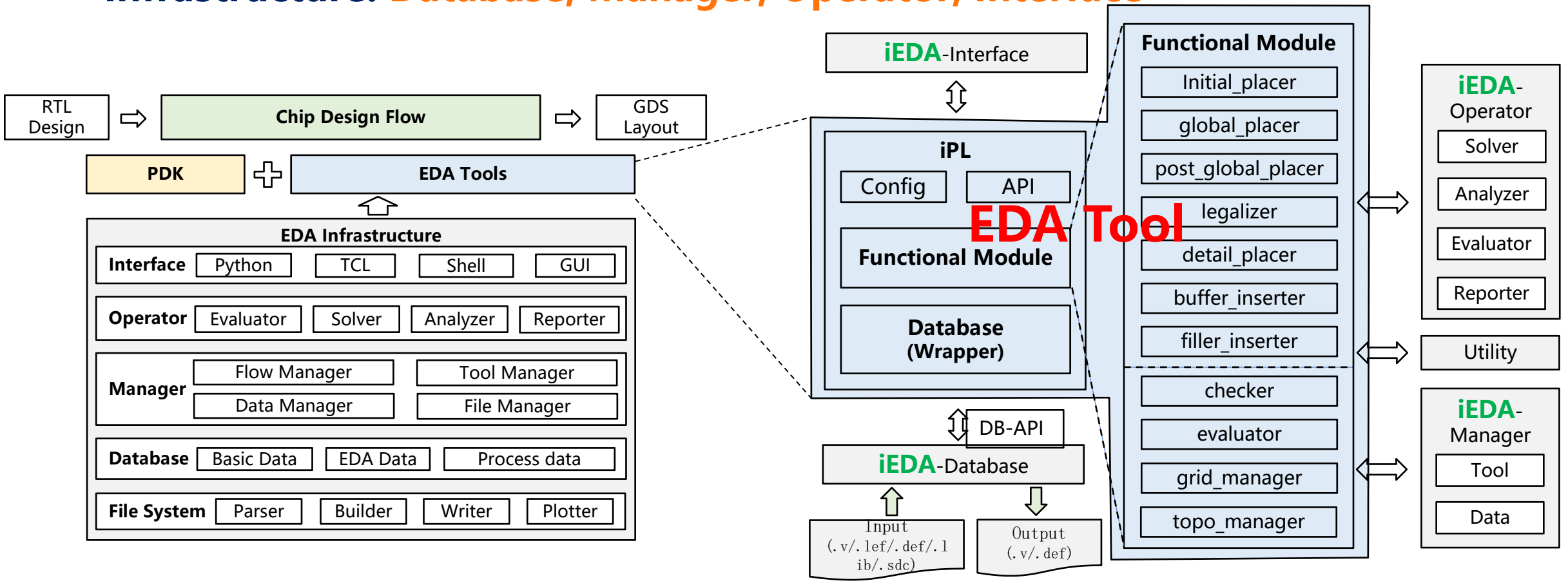
1 EDA infrastructure



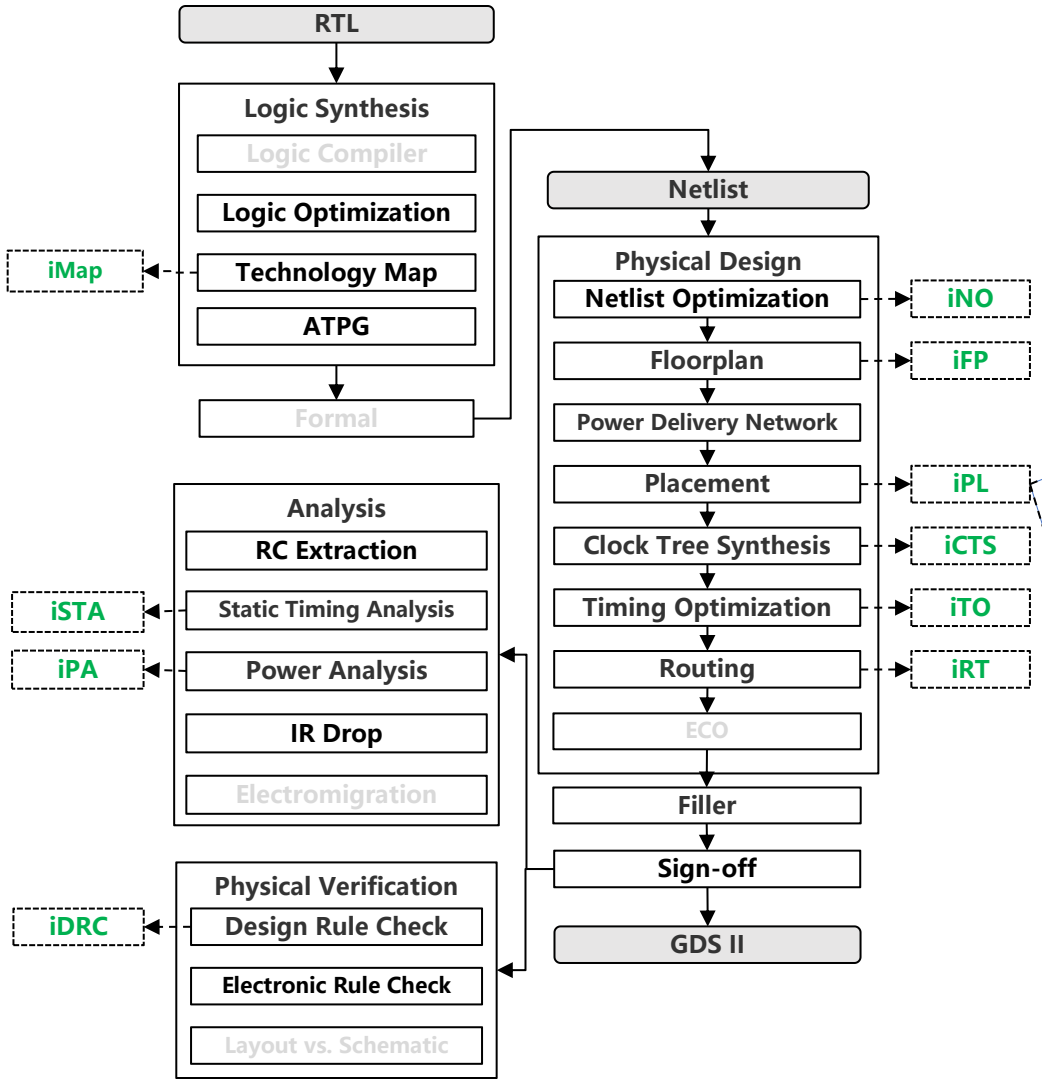
10 EDA tools, 200 technologies, 300 K lines codes

iEDA: Infrastructure

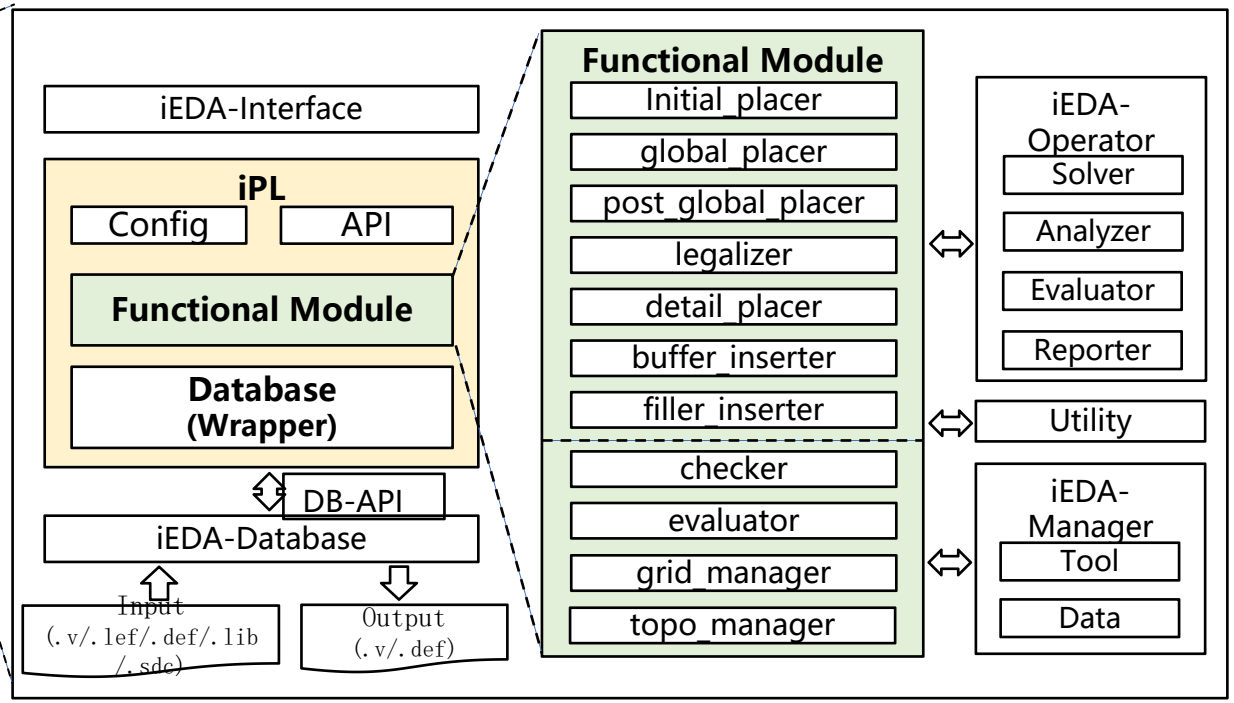
- To fast develop high-quality EDA tool, we need a **S**oftware **D**evelopment **K**it (**SDK**)
- iEDA can be used to support developing EDA tool or algorithm
- **Infrastructure: Database, Manager, Operator, Interface**



iEDA: Toolchain and Tool



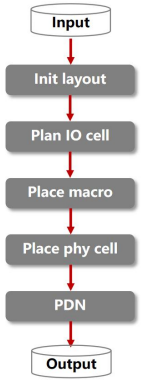
- **Netlist-to-GDS II**
  - **10 tools, and other 5 tools are R&Ding.**
  - **Design, Analysis, Verification**
- **Design Concept:**
  - **Unified framework, deconstructed and merged**
  - **multi-lingual interface, hot-pluggable modules.**



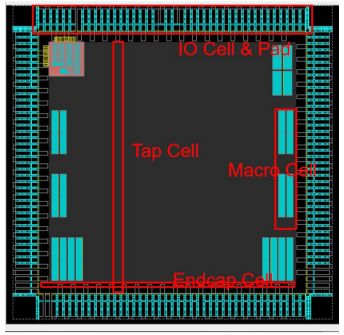
Module 1: Physical Design

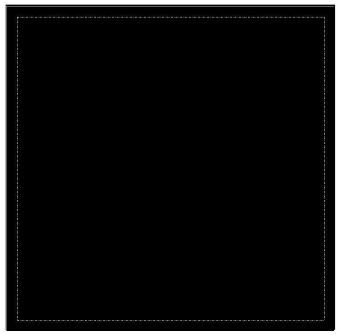
Floorplan (iFP)

Flow



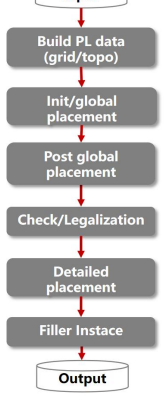
Key Metrics	Data
DIE Area	1.5 × 1.5 mm^2
DIE Utili	0.166554
Core Area	1.16 × 1.15 cm^2
Core Utili	0.279541
#IO Pin	110
#Instance	297504
#Net	311869
Pin	pin (>= 32) = 2893
PDN	M1、M2、M7、M8、M9、AP

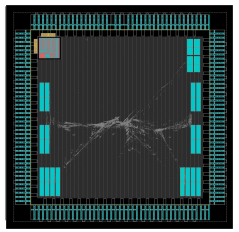




Placement (iPL)

Flow





■ **Min Wirelength Model**

$$\min_{v} \quad W(v)$$
$$s.t. \quad \rho_b(v) \le \rho_0, \ \forall b \in B$$

where v is cell location, $W(v)$ is wirelength, $\rho_b(v)$ is the area density in $b \in B$, ρ_0 is density thredhold.

$$W(v) \begin{cases} HPWL_{ex}(\mathbf{v}) = \max_{i,j \in e} |x_i - x_j| \\ LSE_{ex} = \gamma\left(\ln\left(\sum_{i \in e}\exp\left(\frac{x_i}{\gamma}\right)\right) + \ln\left(\sum_{i \in e}\exp\left(\frac{-x_i}{\gamma}\right)\right)\right) \end{cases}$$

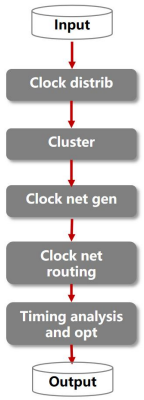
$$\rho_b(v) \begin{cases} D(v) = \frac{1}{2}\sum_{v \in V} D_i(x,y) = \frac{1}{2}\sum_{v \in V} q_i \psi_i(x,y) \\ \begin{cases} \nabla \cdot \nabla\psi(x,y) = -\rho(x,y), \\ \hat{\boldsymbol{n}} \cdot \psi(x,y) = \mathbf{0}, \ (x,y) \in \partial R \end{cases} \\ \iint_R \rho(x,y) = \iint_R \psi(x,y) = 0. \end{cases}$$

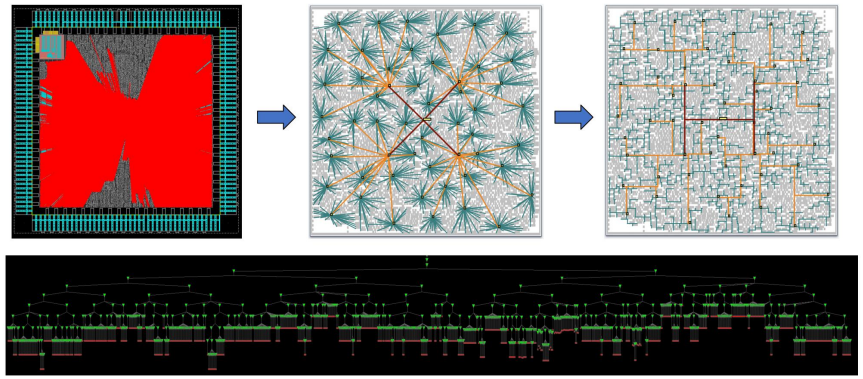
$$\min_{v} \quad f(v) = W(v) + \lambda \sum_{\forall b \in B} \rho_b(v)$$

- **Nesterov Method or** Conjugate Gradient

Clock Tree Synthesis (iCTS)

Flow





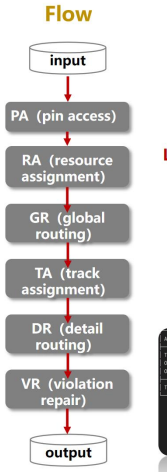
Timing
- Latency (max delay)
- Skew

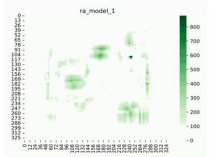
Power
- Buffering
- Wirelength

Violation
- Fanout
- Capacitance
- Slew (transition)

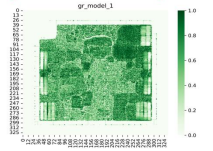
Routing (iRT)

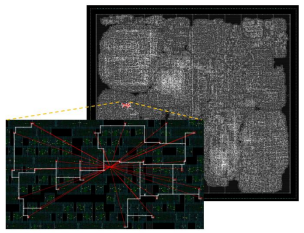
Flow



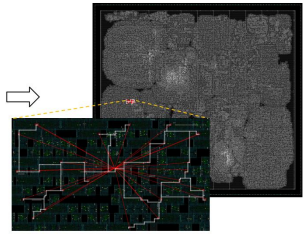


Layout resource/congestion

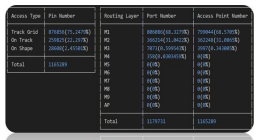




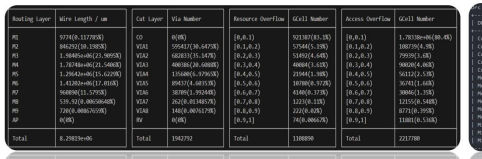
first iteration routing result



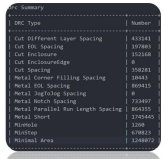
final routing result



Pin Access



Wirelength and via

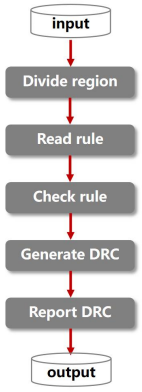


Design rule check

Module 2: Analysis and Verification

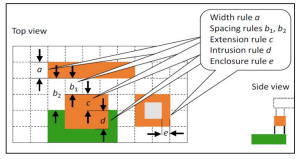
Design Rule Check (iDRC)

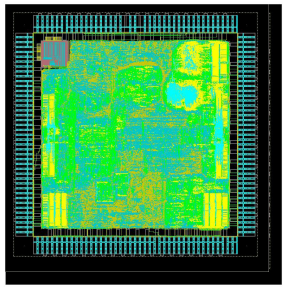
Flow

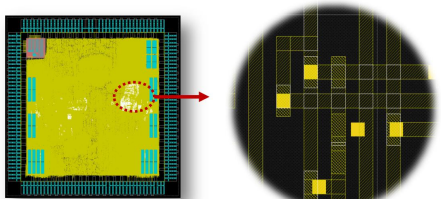


Support DRC Rules:

- Cut Different Layer Spacing
- Cut EOL Spacing
- Cut Enclosure
- Cut Enclosure Edge
- Cut Spacing
- Metal Corner Filling Spacing
- Metal EOL Spacing
- Metal JogToJog Spacing
- Metal Notch Spacing
- Metal Parallel Run Length Spacing
- Metal Short
- MinHole
- MinStep
- Minimal Area



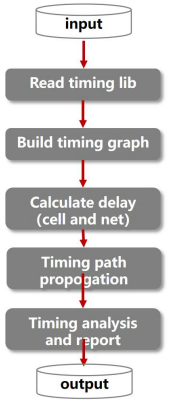




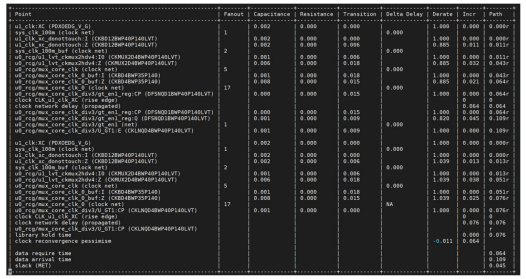
DRC Visulization

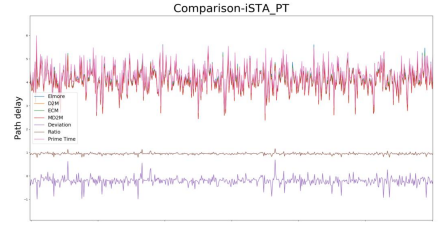
Static Timing Analysis (iSTA)

FLow



Feature
Support hierarchy netlist and def
Basic setup/hold analysis
Support NLDM/Elmore
Support CCS model
Support high-level net delay model
Support sdf mark
OCV
AOCV
POCV
Consider IRDrop analysis on multi-voltage domain
Hierarchy analysis
Crosstalk analysis
clock gate analysis
Latch analysis

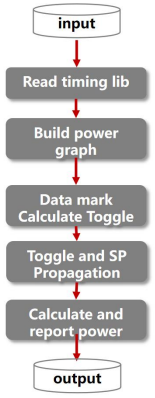




pt/ista ratio	value
mean	1.11
variance	0.00095
median	1.107
maximum	1.5404
minimum	0.9035

Power Analysis (iPA)

Flow



API	Description
buildGraph	Build iPW graph data structure
readVCD	Parse the VCD file
buildSeqGraph	Build timing subgraph
checkPipelineLoop	Detect PipeLine loop
levelizeSeqGraph	Grade timing subgraph
propagateToggleSP	Propagate Toggle and SP data on the graph
calcLeakagePower	Calculate the leakage power
calcInternalPower	Calculate internal power
calcSwitchPower	Calculate switching power
analyzeGroupPower	Analyze power data
reportPower	Output power report

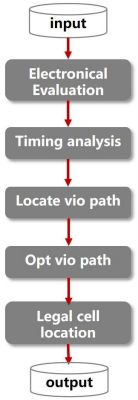
cases	iPA total power	Innovus total power	deviation
aes_cipher_top	22.22mW	23.74mW	6.4%
gcd	0.38mW	0.37mW	3.6%
uart	0.51mW	0.49mW	3.9%

```
Generate the report at 2023-05-06T09:54:06
Report : Averaged Power
+--------------+----------------+--------------+---------------+--------------+-----------+
| Power Group  | Internal Power | Switch Power | Leakage Power | Total Power  |    (%)    |
+--------------+----------------+--------------+---------------+--------------+-----------+
| combinational | 1.064e-07     | 5.063e-09    | 3.079e-08     | 1.422e-07    | (27.595%) |
| sequential   | 2.862e-07      | 7.337e-09    | 7.963e-08     | 3.732e-07    | (72.405%) |
+--------------+----------------+--------------+---------------+--------------+-----------+
Net Switch Power   ==   1.248e-08 (2.406%)
Cell Internal Power   ==   3.926e-07 (76.173%)
Cell Leakage Power   ==   1.104e-07 (21.422%)
Total Power   ==   5.154e-07
```

- Evaluate power before / during / after the physical design process
- Average model
- Timing window (coming soon)
- VCD parser
- Report/API

Timing Optimization (iTO)

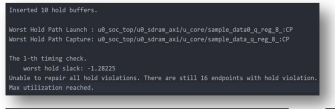
Flow

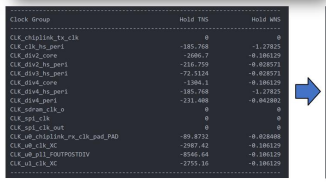


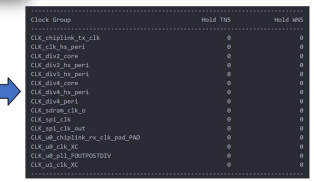
Key parameter config	
Input	iPL.def, iCTS.def
output	iTO_setup_result.def, iTO_hold_reslut.def
setup_slack_margin	setup slack value
hold_slack_margin	hold slack value
max_buffer_percent	Area ratio of inserted buffer
max_utilization	Core utilization
DRV insert_buffers	Available buffer for optimizing DRV
setup_insert_buffers	Available buffer for optimizing setup
hold_insert_buffers	Available buffer for optimizing hold
number_passes_allowed_decreasing_slack	The number of times that WNS is allowed continuously decrease when opt setup
rebuffer_max_fanout	For setup, a wire network is not optimized for buffer insertion when its fanout exceeds this value
split_load_min_fanout	For setup, fanout is reduced by inserting a buffer when fanout is greater than this value

DRV report

Setup report





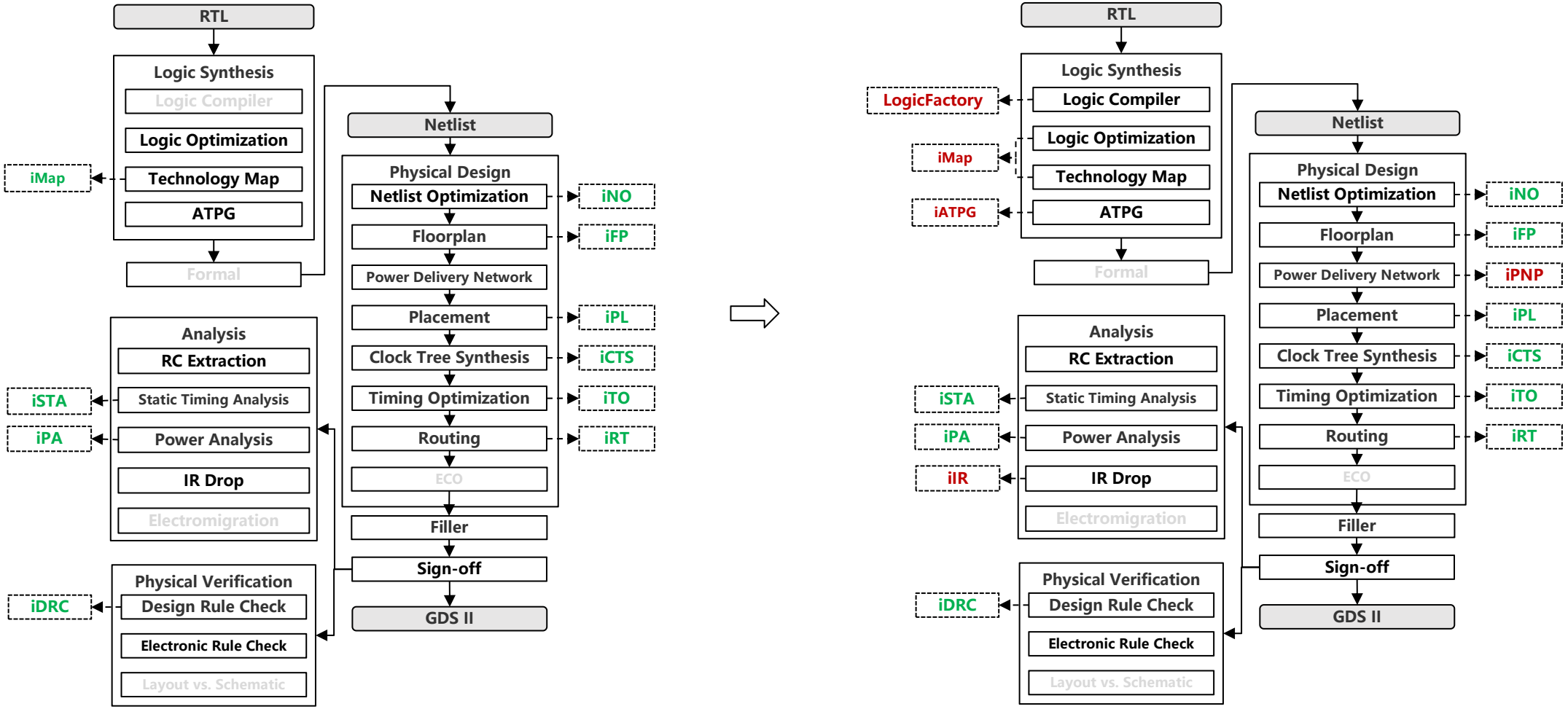


Hold report

- Fix timing design rule violation
  - Max cap/Max slew/Max wirelength/Max fanout
- Fix hold time
- Fix setup time
- Cell sizing
- Buffer Insertion
- Load Insertion
- Buffer/load location

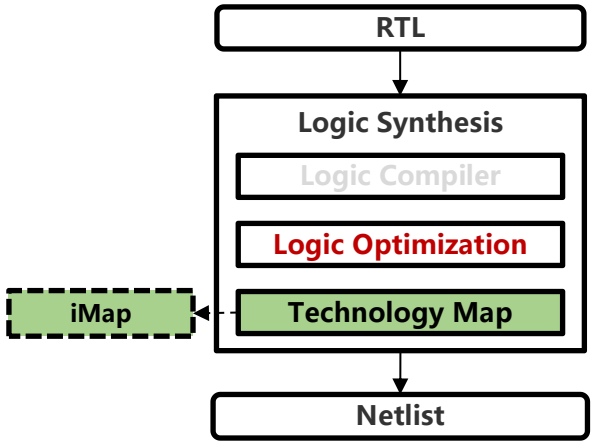
Update ①: New EDA Tools

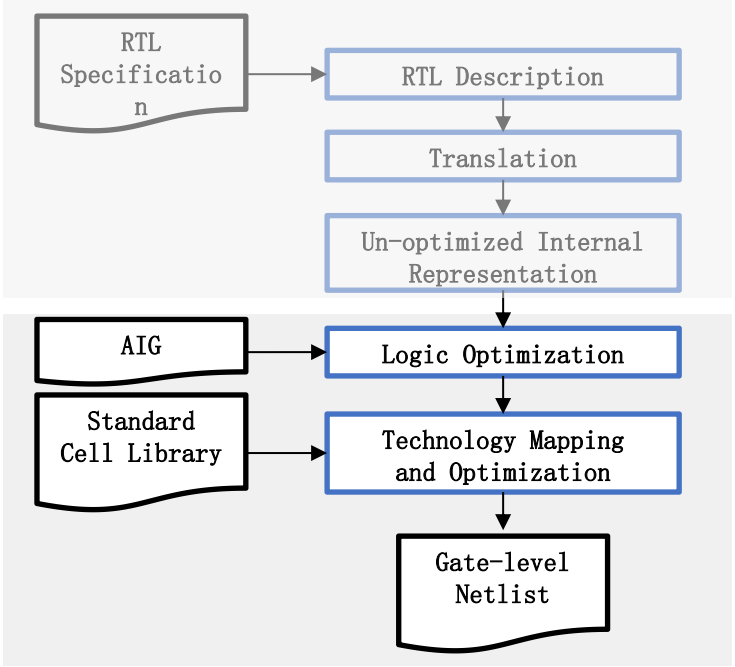
- **We develop/extend 5 tools:**
  - **LogicFactory** for Logic Synthesis, **iMap** for logic optimization and tech mapping, **iATPG** for automation test pattern generation, **iIR** for IR drop calculation, **iPNP** for power network planning.



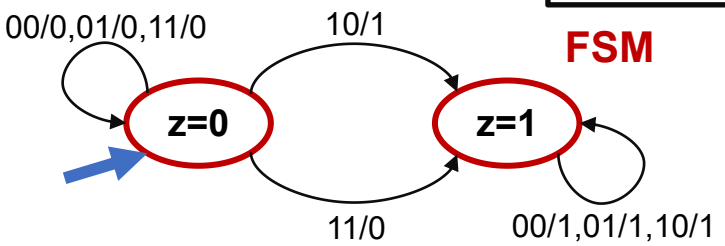
Logic Opt & Tech Map (LogicFoctory & iMap)

iEDA

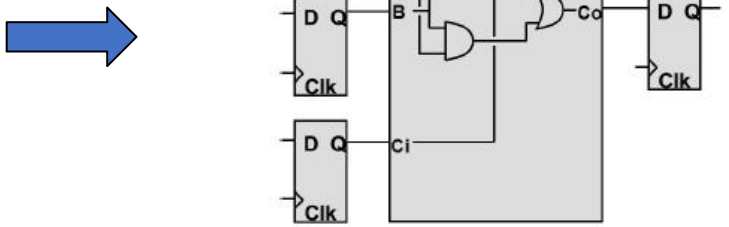




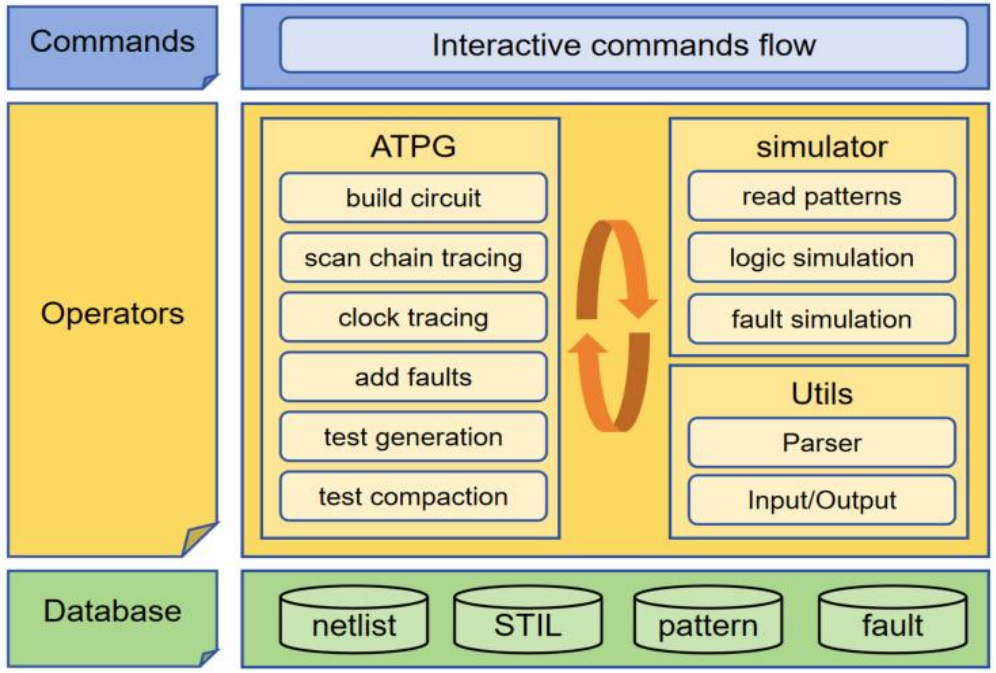
- **iMap 1.0 includes Logic Optimization and Tech Mapping**
  - ✓ **Logic Optimization operators**
    - ✓ Rewrite (NPN matching based on 4-input truth table)
    - ✓ Refactor (optimization based on SOP expression)
    - ✓ Balance (balance algorithm based on AND-tree)
    - ✓ LUT-opt (optimization based on FPGA tech mapping)
  - ✓ **Technology Mapping for ASIC and FPGA**

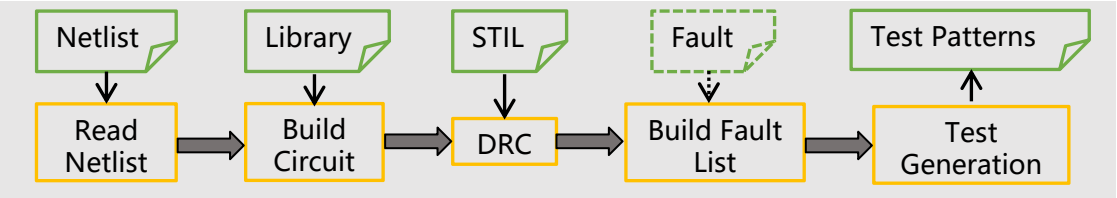


$$z(t) = c(t)d'(t) + c'(t)z(t-1)$$



Automation Test Pattern Generation (iATPG)



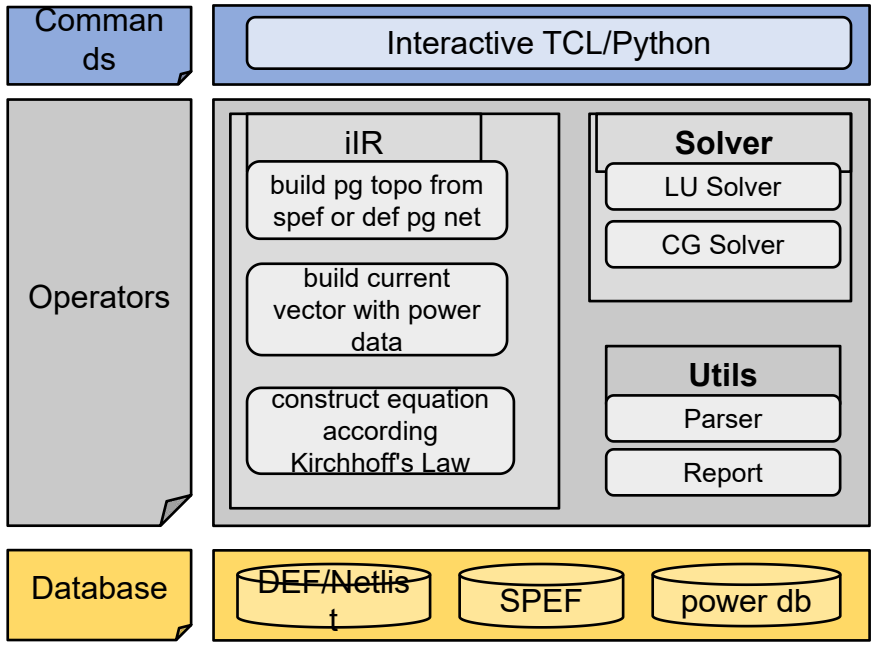


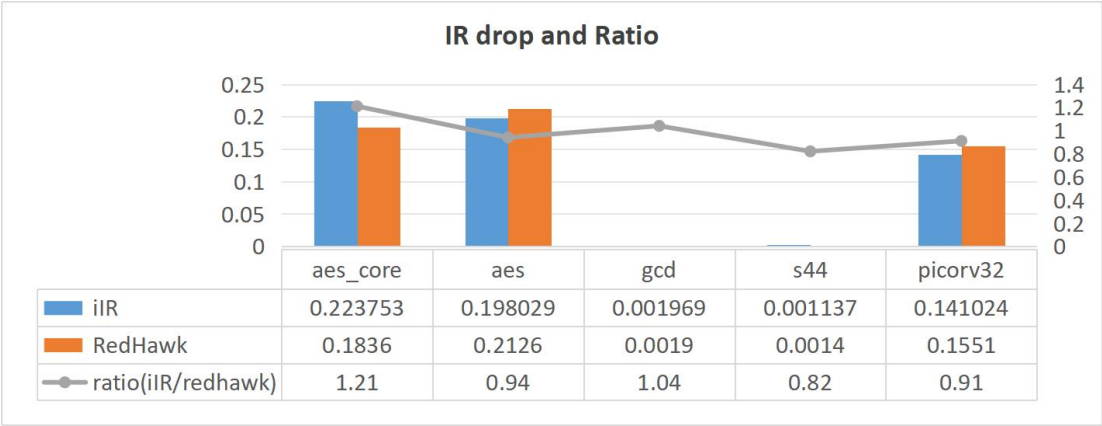
- **iATPG: Implement the basic flow of test generation**
  - **Fault Simulation**
    - ✓ Support scan pattern and functional pattern
    - ✓ Self-developed heuristic method
    - ✓ CPU-based parallelism under NUMA architecture
  - **Test Pattern Generation**
    - ✓ Combinational netlist → combinational pattern
    - ✓ Full-scan netlist → basic scan pattern
    - ✓ Partial-scan netlist → clock sequential pattern

Design	#Gates	iATPG			Commercial tool			Fault Cover. Ratio (iATPG/Com. tool)
		fault cover.	time(sec.)	# of patterns	fault cover.	time(sec.)	# of patterns	
s15850	2,334	89.86%	1.08	91	92.41%	1.30	117	**97.24%**
s5378	3,061	93.70%	2.33	323	93.96%	2.20	197	**99.73%**
s13207	4,388	90.00%	2.19	189	89.73%	1.90	96	**100.30%**
s38584	19,095	95.72%	24.60	727	95.81%	5.90	267	**99.90%**
s35932	24,080	95.28%	5.85	128	95.13%	2.50	117	**100.16%**
s38417	23,795	91.97%	70.42	936	95.43%	17.70	264	**96.37%**
b21	30,042	90.96%	139.24	902	92.82%	255.80	854	**97.99%**
b22	45,152	86.41%	337.75	1,103	95.04%	836.90	1,146	**90.92%**
b18	345,004	90.45%	12,958.18	5,886	97.21%	1,425.90	1,572	**93.05%**
wb_conmax	67,600	94.37%	239.49	1,454	96.02%	69.20	830	**98.28%**
des_perf	170,963	95.38%	532.32	409	92.33%	93.20	279	**103.31%**
vga_lcd	176,025	98.47%	10,338.98	8,245	99.33%	147.40	3,456	**99.13%**

Fault Cover Ratio: 98%

IR Drop Calculation (iIR)

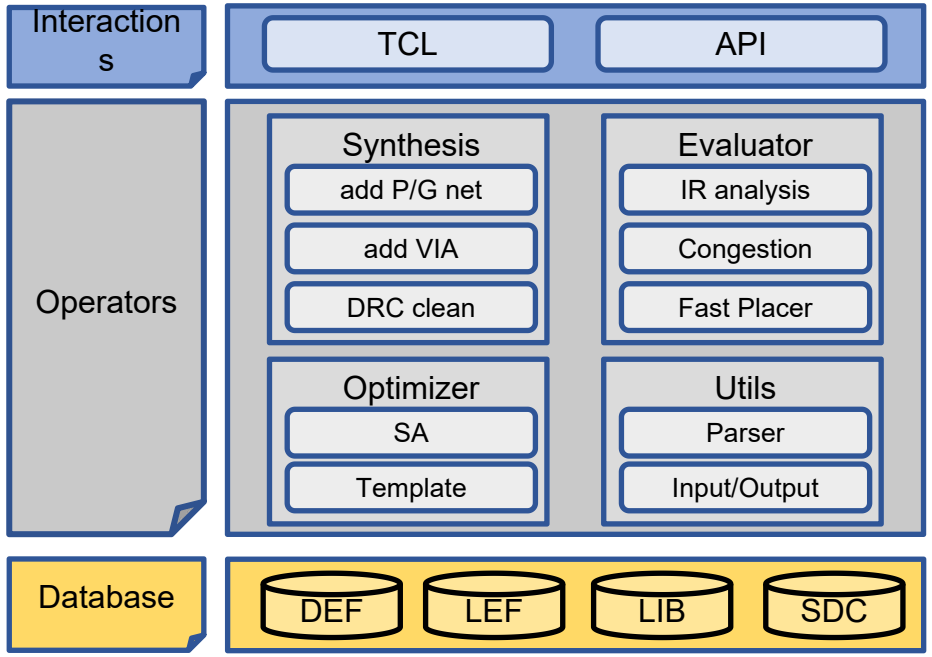


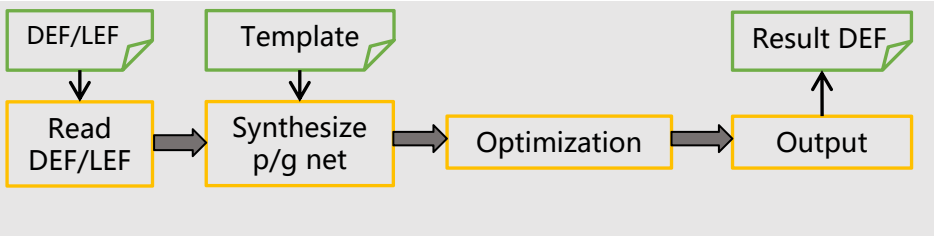


	aes_core	aes	gcd	s44	picorv32
iIR	0.223753	0.198029	0.001969	0.001137	0.141024
RedHawk	0.1836	0.2126	0.0019	0.0014	0.1551
ratio(iIR/redhawk)	1.21	0.94	1.04	0.82	0.91

- **iIR:** analyze the IR drop of the chip.
  - **Feature**
    - ✓ **Read SPEF File by Rust Parser or Estimate RC Data**: Extracts the RC data of the power network from the SPEF file or estimate the RC When iIR as a IR engine.
    - ✓ **Read Instance Power Database or Run Power Analysis Adhoc**: Reads the instance power data from the iPA DB file, or run iPA power analysis adhoc.
    - ✓ **Solving IR Drop:** Builds the conductance matrix and the current vector, and then use GV=J solver (the LU solver or the CG solver) to solve the static IR Drop.
    - ✓ **GPU Mode**: Accelerate the IR Drop solving process using CUDA.
    - ✓ **IR Engine:** Used as a standalone tool or as an IR analysis engine, such as for estimating IR Drop within iPNP.

Power Network Plan (iPNP)





- **iPNP 1.0 Implement the basic flow of P/G net generation**
  - **Power Network Generation**
    - ✓ Set a series of P/G templates including width and pitch
    - ✓ Synthesize the P/G net automatically according to template
    - ✓ Add VIA automatically
  - **Power Network Optimization**
    - ✓ Balancing the IR drop and the Congestion
    - ✓ Using Simulated Annealing algorithm

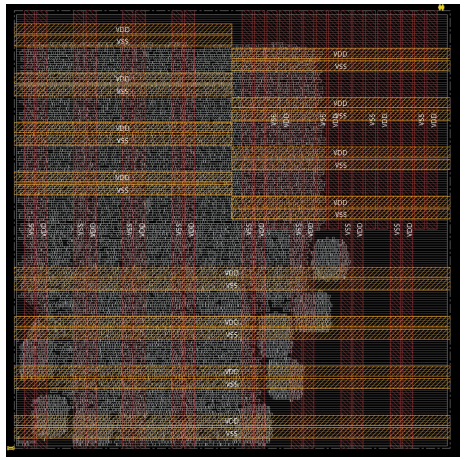


Fig1. Power Stripes

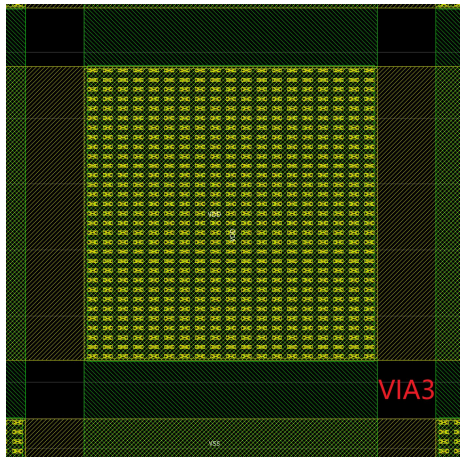


Fig2. Power VIA

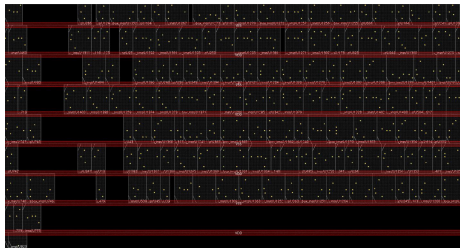


Fig3. Power Row

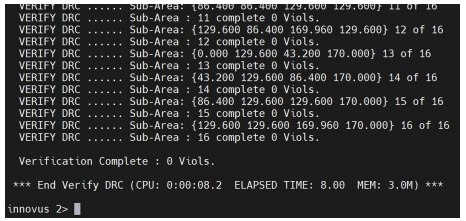
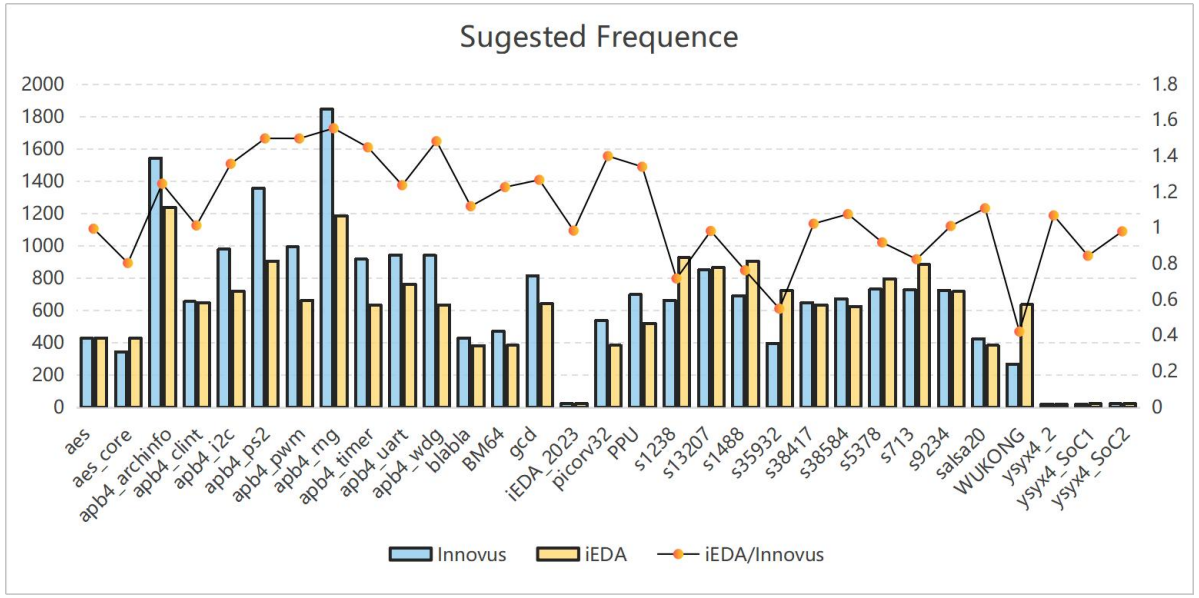


Fig4. DRC Clean

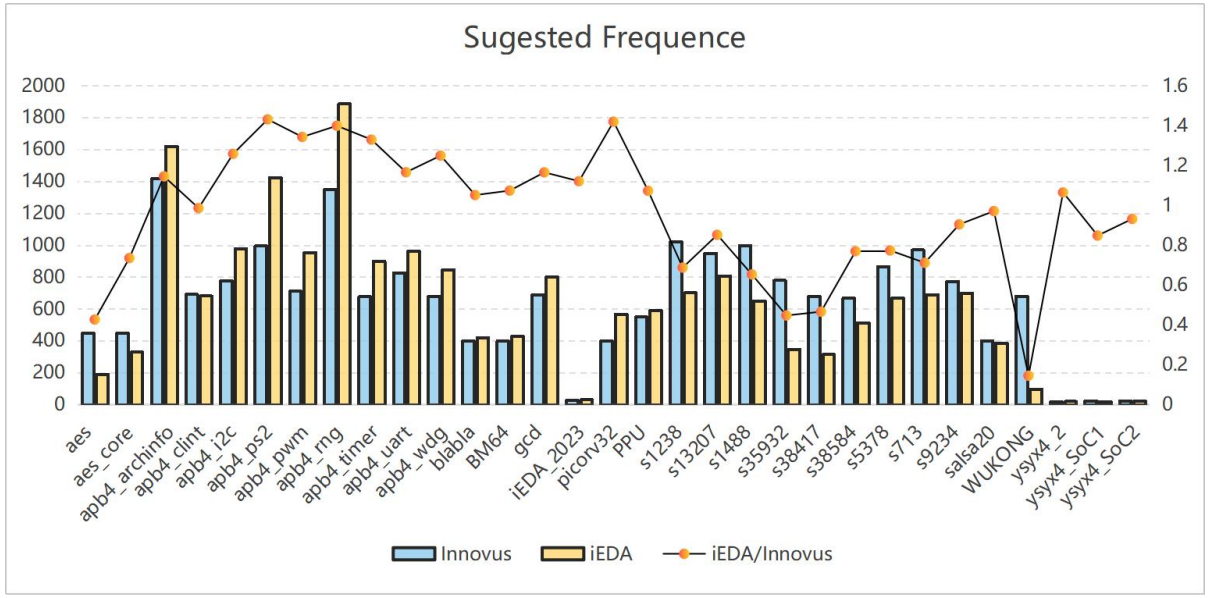
Update ②: Improve the QoR of iEDA

- **Timing Comparison of iEDA/Innovus in 28nm**
  - We test 30 cases with instance num 135 ~ 1,173,610
  - Compare with innovus, we achieve following frequnece result



Post CTS

108.89 %

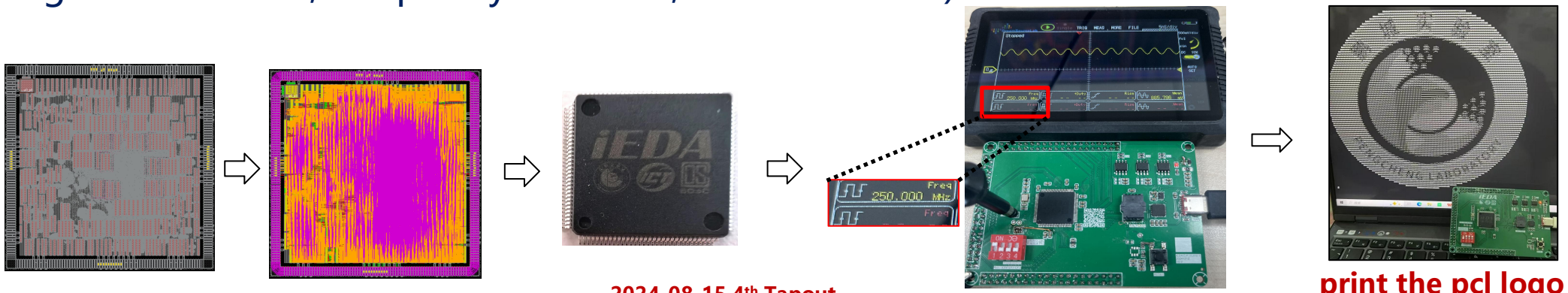


Post Route

95.40 %

Update ③: Two Times Tape-out

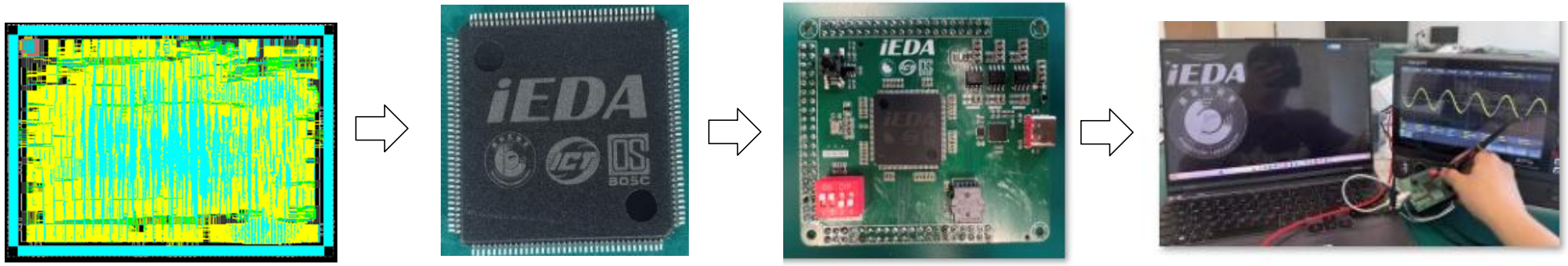
- **We acheive two times tape-out in 28nm (ten-million-gates)**
  - The fourth one: CPU "Nutshell" (**348** macro modules, 756,000 standard cells, 6.449 million gate transistors, Frequency: 800MHz, Area: 6.49mm²)



2024-08-15,4th Tapout

print the pcl logo

  - The fifth one: Ten-million-gate CPU "Nutshell-2.0" (321 macro modules, 829,000 standard cells, 10.49 million gate transistors, Frequency: 900MHz, Area: 4.0mm*2.7mm)



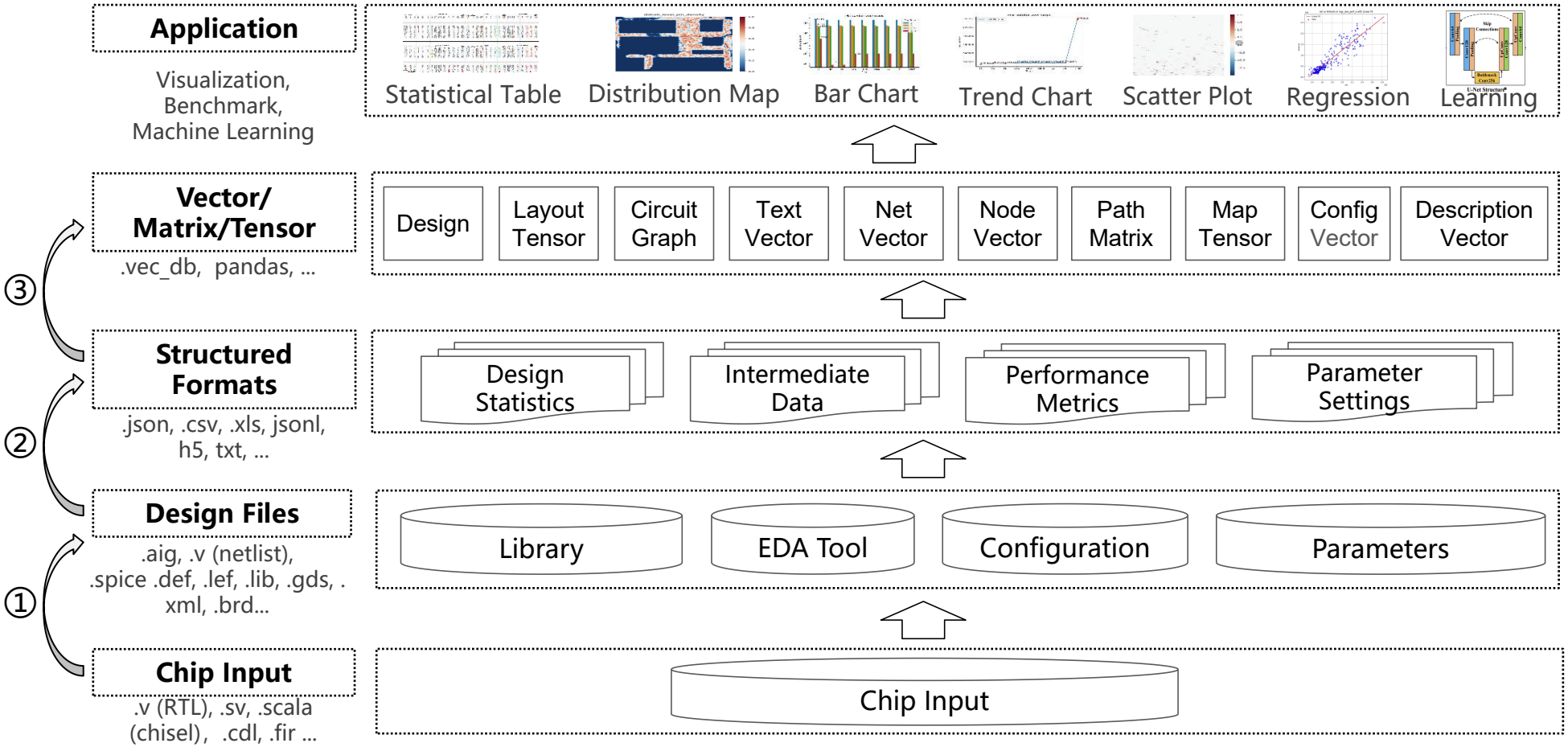
2025-01-10,5th Tapout

01 iEDA

02 AiEDA

Work ①: AI-EDA Python Library

- **AiEDA: File-to-Vector**
  - Convert standard file formats into vectorized storage
  - Transform chip data into vectorized data that can be input into AI models.



Design-to-Vector

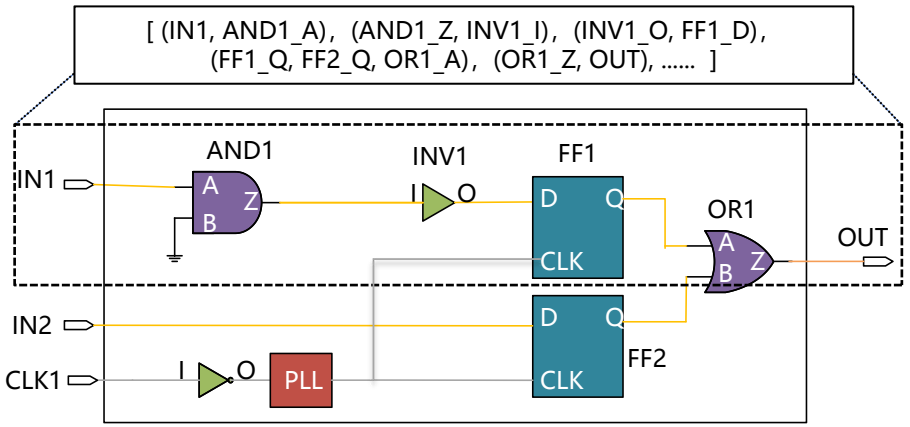
- Design-to-Vector
  - Netlist-to-Vector, Layout-to-Vector, Map-to-Vector, Net-to-Vector, Shape-to-Vector

```
module data_register (
    input clk,
    input [3:0] data_in,
    output reg [3:0] data_out
);
    always @(posedge clk) begin
        data_out <= data_in;
    end
endmodule
```

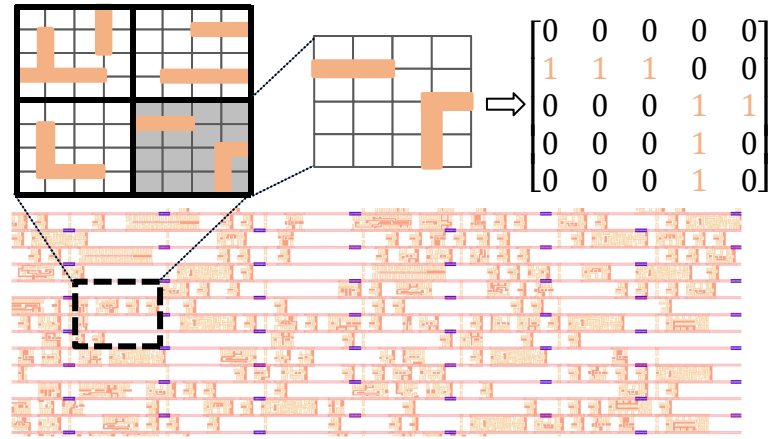
⇩

[“module”,“data_register”, “(”, “input”,“reg”, “clk”, “3”, “data_in”, “data_out”; “endmodule”, “reg”, “case”, “endcase”, ...]

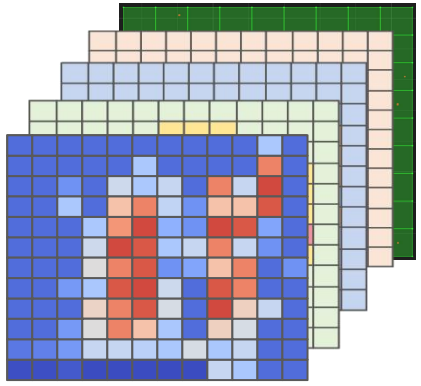
RTL



Netlist

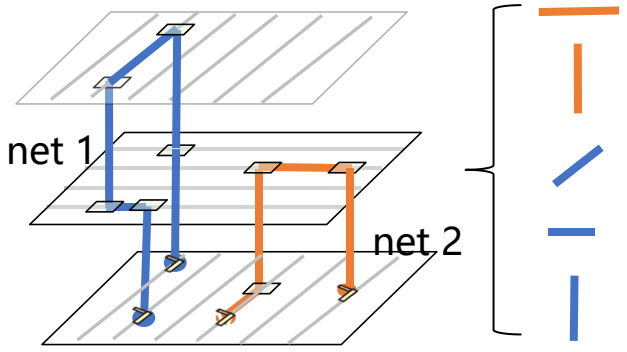


Layout



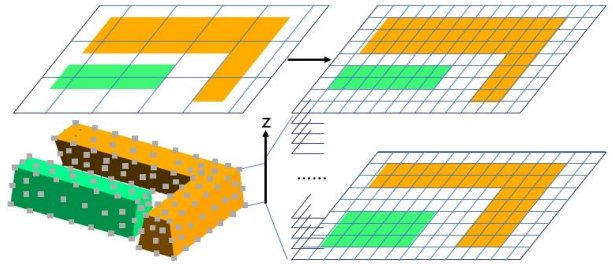
Map

$$\begin{bmatrix} 0 & 0 & 1 & 2 & 0 \\ 1 & 4 & 3 & 3 & 1 \end{bmatrix} \cdots \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 5 & 7 & 6 & 4 & 0 \\ 4 & 3 & 2 & 5 & 0 \\ 1 & 5 & 8 & 7 & 0 \\ 0 & 3 & 4 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{matrix} 2 \\ 1 \\ 0 \\ 0 \end{matrix} \cdots$$



Net

Wire & Via



Polygon & Polyhedron

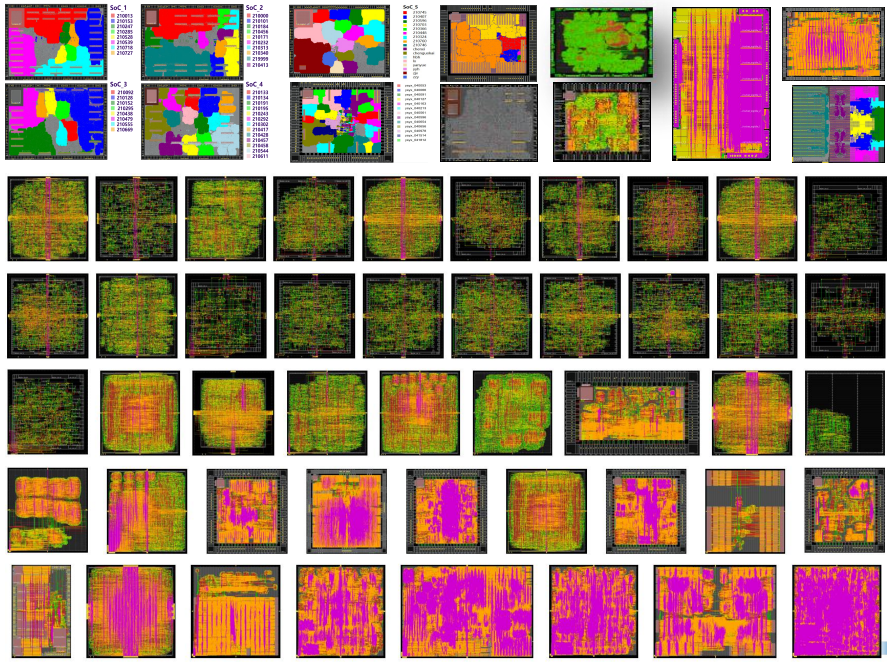
Work ②: A Dataset for AI-EDA

- **iData:** A chip dataset with **source data**, **vectorized data**, **feature data** for **AI4EDA** and **large model** task
  - 50 Circuits range from small (**1000 gates**) to large (**30M+ gates**); memories, CPUs, GPUs, and SoCs
  - Complete vectorization at **design, netlist, layout, net, path, patch and map levels**

Feature levels

Circuits	Design				Net		Path		Patch	
	#Cells	#Nets	#Pins	#Wires	#Files	Size	#Files	Size	#Files	Size
s713	135	125	345	1426	121	890K	56	676K	324	1.59M
s44	178	128	494	2095	128	1.31M	128	558K	441	2.31M
rng	195	204	581	2230	169	1.39M	132	1.29M	441	2.45M
gcd	297	270	894	3733	270	2.32M	136	3.81M	625	3.85M
s1238	349	290	1050	4998	290	3.00M	24	284K	576	4.27M
s1488	380	325	1202	6422	325	3.82M	24	584K	506	4.99M
arch	392	381	1232	5122	346	3.19M	304	1.45M	841	5.14M
ps2	515	497	1669	6542	432	4.04M	372	5.72M	1024	6.35M
s9234	657	585	1981	8592	577	5.24M	456	5.36M	1225	8.05M
timer	721	689	2239	8960	653	5.65M	332	4.51M	1521	9.32M
s13207	727	647	2107	8182	636	5.07M	744	5.03M	1600	8.83M
i2c	790	727	2449	10248	676	6.37M	568	9.10M	1521	9.85M
s5378	881	774	2628	12422	774	7.62M	564	6.60M	1600	11.61M
pwm	974	889	3162	13596	838	8.36M	480	4.66M	1936	13.02M
wdg	1029	945	3245	13889	910	8.58M	456	7.69M	1936	13.17M
clint	1069	1004	3241	13634	969	8.43M	524	7.63M	1936	13.07M
ASIC	1228	796	2748	10737	796	6.82M	76	660K	121	5.36M
s15850	2088	1926	6639	27941	1925	17.33M	1724	21.00M	4225	27.66M
uart	5981	5606	20021	83268	5555	53.45M	4652	99.96M	11449	83.34M
s38417	6028	5573	20185	85054	5573	53.30M	5764	113.04M	12544	84.65M
s35932	6375	5837	19396	81158	5837	52.02M	6912	33.66M	14161	93.73M
s38584	7023	6586	22839	97771	6585	61.51M	4916	62.76M	12100	94.02M
BM64	9358	9510	30285	132076	9510	85.63M	5164	183.90M	16641	142.47M
picorv32	9430	9077	30600	136455	9010	87.75M	6560	177.09M	21316	150.30M
PPU	9547	8895	33510	140136	8895	90.05M	9552	164.43M	20164	143.55M
blabla	15154	15672	47516	216427	15671	144.32M	4396	230.33M	24649	252.01M
aes_core	17940	17371	66207	310215	17371	202.32M	9876	623.39M	32041	311.46M
aes	19181	18117	70455	325550	18117	212.03M	11940	623.84M	36100	323.91M
salsa20	21270	20432	66869	291172	20432	190.66M	14916	439.55M	40401	320.98M
jpeg	27671	29160	92424	366397	29160	245.32M	18128	786.71M	66049	424.45M
eth_top	42279	38552	142784	646875	38552	434.20M	20000	562.18M	169744	967.16M
yadan	63514	31280	107120	483369	31280	331.30M	19832	816.37M	15376	313.25M
beihai	211236	133086	452119	2161829	132424	1.53G	52628	4.87G	92256	1.60G
SHMS	268721	251772	850432	3610024	251686	2.51G	70672	6.45G	19153	1.98G
mdlla	289344	226974	739140	3708427	226904	2.69G	20000	1.21G	28224	2.40G
ZJUC	348985	323195	1093302	4880442	323175	3.35G	184416	12.61G	23968	2.66G
iEDA23	368147	335112	1138727	5132004	335026	3.50G	219852	21.55G	28224	2.84G
yyyx4S2	494962	449847	1558094	6967779	448950	4.79G	11864	1.58G	28224	3.84G
beihai2	582645	393308	1086497	5491501	391062	4.21G	70264	11.04G	126630	4.19G
AIMP	742210	535618	1407879	9980714	534081	8.07G	26296	1.48G	77841	8.08G
AIMP2	816677	560525	1494198	9133333	558606	7.03G	22540	3.81G	134246	7.24G
yyyx6	1090820	1029515	3486956	15486068	1029496	10.98G	184700	26.41G	60501	8.95G
wukong	1102663	1032718	3390152	15653639	1031971	10.98G	20120	1.88G	39480	9.05G
yyyx41	1123368	1105564	3833714	18248292	1105478	12.24G	219848	21.67G	28224	9.73G
yyyx42	1173610	1147953	3906678	17416249	1147867	11.98G	199704	14.18G	36100	9.56G
T1	1262053	1227098	4208135	18769036	1227019	12.63G	40000	2.62G	24649	10.03G
T1_M	2222669	2162147	7419623	33115508	2162068	22.82G	40000	3.08G	43681	18.27G
nanhu	2793215	2646672	8838039	42524007	2643701	27.84G	20136	1.70G	75040	25.04G
C910	3282828	2948743	9618301	52259408	2942510	36.25G	40588	2.85G	152100	33.82G
T1_S	4816399	4728816	16688213	79050737	4728737	50.22G	40000	5.99G	77841	44.13G
Total	23.26M	21.47M	72.02M	347.15M	21.45M	235.91G	1.63M	149.87G	1.61M	207.18G

Chip statistics



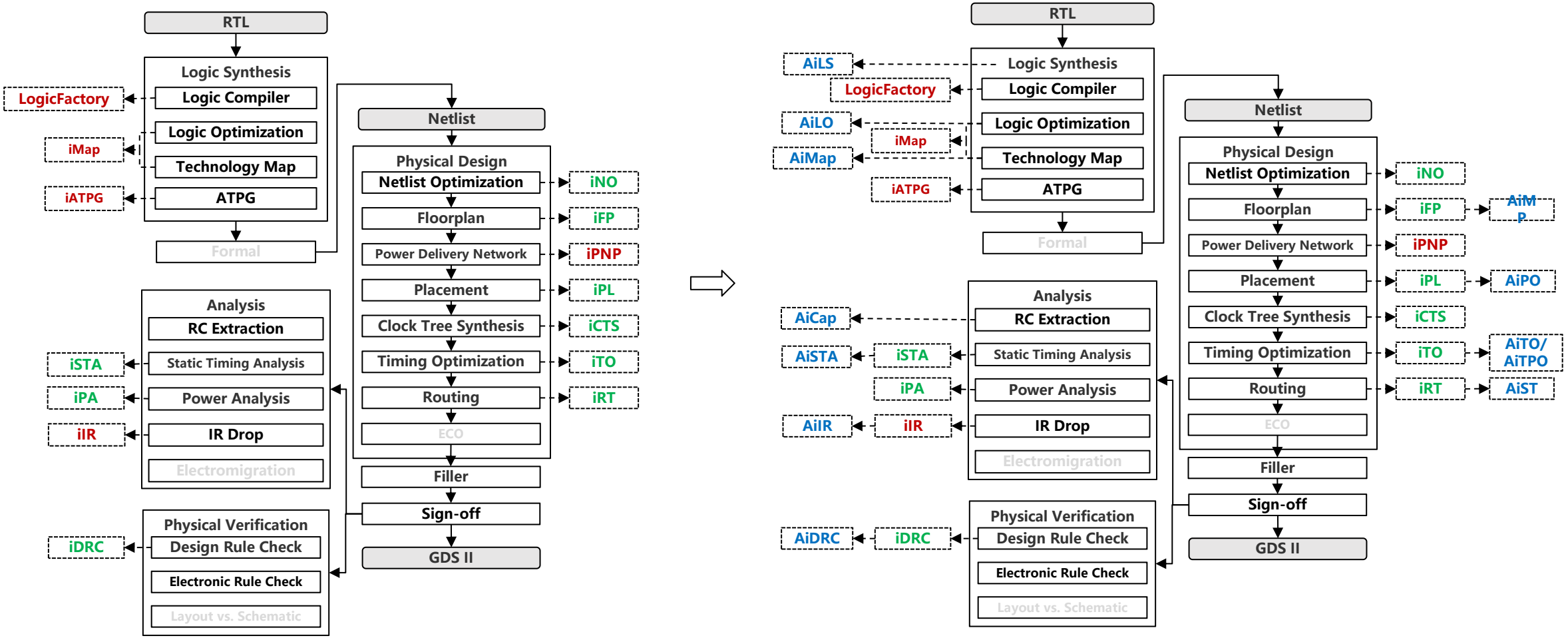
Chip Layouts

AiEDA > application > benchmark > 28nm > gcd > output > iEDA > feature > {} gcd_place_eval.jsonl


```
{"Wirelength":{"FLUTE":3694690,"GRWL":3650000,"HPWL":3168669,"HTree":4439356,"VTree":4633732}}
{"Density":{"cell":{"allcell_density":"/data/yhqiu/benchmark/AiEDA/application/test/iEDA/density_map/allcell_density.csv","macro_density":"/data/yhqiu/benchmark/AiEDA/application/test/iEDA/density_mar
{"Congestion":{"map":{"egr":{"horizontal":"./rt_temp_directory/initial_router/egr_horizontal.csv","union":"./rt_temp_directory/topology_generator/overflow_map_planar.csv","vertical":"./rt_temp_directc
{"Timing":{"DR":[{"clock_name":"core_clock","hold_tns":0.0,"hold_wns":0.109753,"setup_tns":0.0,"setup_wns":1.499311,"suggest_freq":999.3114743941425}],"EGR":[{"clock_name":"core_clock","hold_tns":0.0,
{"Power":{"DR":{"dynamic_power":0.00011826468785753447,"static_power":7.931440122061174e-06},"EGR":{"dynamic_power":0.00011720249269325531,"static_power":7.931440122061174e-06},"FLUTE":{"dynamic_power
```

Work ③: AI-EDA Models

- **To achieve AI-Aided Design, we trained some AI4EDA models**
  - We use **AiEDA** (python library) to train AI model and run chip design flow with AI4EDA model assistence
  - We feed vectorized data, feature data in **iData** (dataset) to AI model

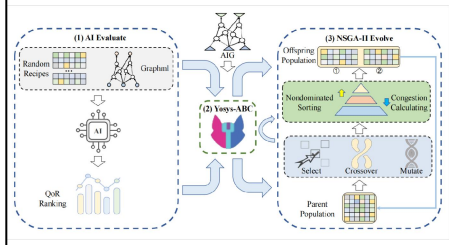


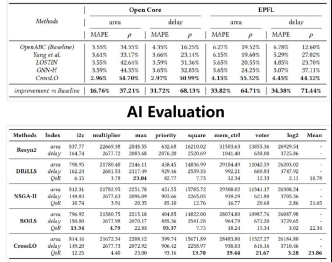
AI-EDA Models

- **1)** Logic Optimization, **2)** Steiner Tree Generation, **3)** Capacitance Extraction,
- **4)** Tech Mapping, **5)** Timing Optimization, **6)** Timing Analysis

AiLO: AI for Logic Optimization

- Optimize AIG（Delay, Area）
  - Evaluate delay and area by AI, guide delay and area optimization





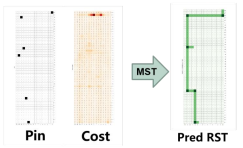
AI Evaluation

Delay and Area Optimization

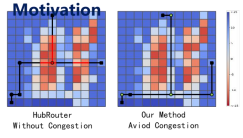
[TODAES-25]

AiST: AI for Steiner Tree

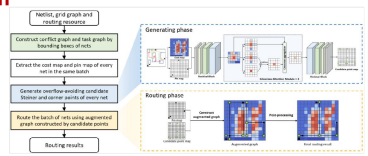
- **AI for Steiner Tree Generation**



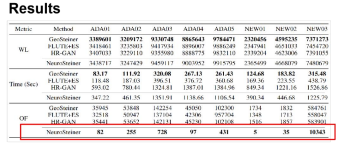
- **Motivation**



- **Method**



- **Results**



- Liu et. al., NeuralSteiner: Learning Steiner Tree for Overflow-avoiding Global Routing in Chip Design, In Proc. of NIPS, 2024

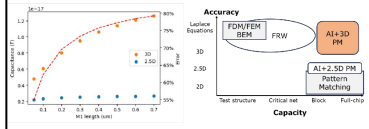
[NIPS-24]

AiCap: AI for 3D Capcitance Extration

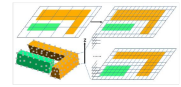
- **Formulation**
  - **Laplace's equation**

$$\begin{cases} \varepsilon_i \nabla^2 \phi = 0, & \text{in } \Omega_i \\ \phi = \phi_0, & \text{on } \Gamma_u \\ q = \partial\phi/\partial \boldsymbol{n} = 0, & \text{on } \Gamma_q, \end{cases} \quad \nabla^2 \phi = \frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} + \frac{\partial^2 \phi}{\partial z^2} = 0.$$

- **Numerical Methods:**
  - FDM, FEM, BEM, RWM
- **2.5D & 3D**



- **Point Cloud**

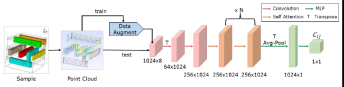


High resolution ensures unique geometry

Sparse data leads to redundant storage

Discretization trouble in the third dimension

- **Transformer**



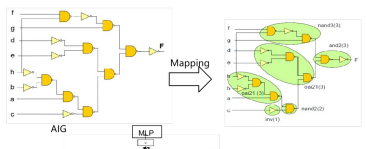
- **Experiment**
  - Relative errors within ±5% reaches 96.53%;
  - Average relative error is 1.4%, meeting industrial standards;

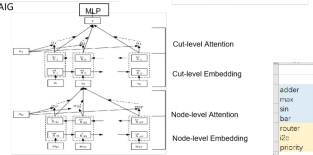
- S. Wu, et. al., An Adaptive Partition Strategy of Galerkin Boundary Element Method for Capacitance Extraction，ASPDAC'23.
- Y. Liang, et. al., PCT-Cap: Point Cloud Transformer Network for 3D Capacitance Extraction, ISEDA'24

[ASPDAC-23, ISEDA-24]

AiMap：AI for Tech Mapping

- **In tech mapping, learn cut delay and apply it to cut sorting and selection.**
  - **Minimize depth —> delay**





name	input	output	Area	Depth	Area(STA)	Delay(STA)	Delay/Depth
log2	32	32	26556.98	3891.66	26561.26	6797.77	1.75
square	64	128	15738.82	2541.48	15744.07	3680.87	1.45
adder	256	129	898.31	2613.78	898.13	3770.65	1.44
sin	24	25	5206.79	1842.34	5207.04	3955.57	2.15
div	128	128	60539.96	44488.53	60509.1	66126.44	1.49
hyp	256	128	210418.6	176427	210437.5	743139.2	4.21
max	512	130	2312.56	2510.89	2312.27	3809.03	1.52
sqrt	128	64	20274.38	48291.56	20252.2	180518.1	3.74
Multiplier	128	128	25454.65	2682.57	25458.31	4649.1	1.73
bar	135	128	2681.62	152.96	2680.39	1114.9	7.29

The gap between Depth and Delay is very large.

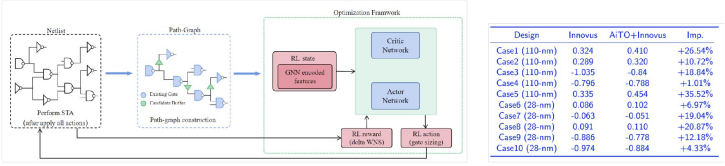
ABC			Strategy 1			Strategy 2			Strategy3		
	Area	Delay	Area	Delay	Gain	Area	Delay	Gain	Area	Delay	Gain
adder	898.31	3770.65	955.69	3404.78	3.3%	1095.1	2776.91	4.4%	1098.32	2642.03	7.7%
max	2312.56	3809.03	2328.33	3049.55	19.3%	2770.22	2806.23	26.7%	2270	2796.24	28.4%
sin	5206.79	3955.57	4670.22	3432.19	23.5%	4570.22	3432.19	23.5%	4670.22	3432.19	23.5%
bar	2681.62	1114.9	2571.65	616.94	48.6%	2571.95	616.94	48.9%	2571.95	616.94	48.6%
router	241.96	471.04	236.69	450.19	6.5%	230.81	456.51	7.7%	204.58	507.32	7.7%
i2c	640.93	280.22	557.38	230.85	11.7%	557.38	230.96	11.7%	557.38	230.85	11.7%
priority	772.84	2546.25	748.04	2548.25	3.2%	741.63	2548.25	4.0%	741.63	2548.25	4.0%

- Junfeng Liu, et. al., AiMap: Learning to Improve Technology Mapping for ASICs via Delay Prediction, In Proc. ICCD, 2023.

[ICCD-23, TCAD-25]

AiTO: AI for Timing Optimization

- **AI for Timing Optimization (Buffer Insertion and Cell Sizing)**
  - We constructed a model that combines GNN with RL, achieving better WNS optimization.
  - Compared to OpenROAD, an average performance improvement of 10.006%.
  - Further refined using Innovus, demonstrated an average performance improvement of 4.336% compared to directly using Innovus for optimization.



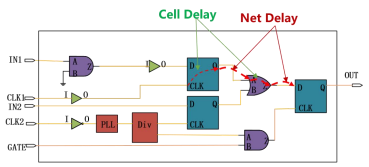
Design	Innovus	AiTO+Innovus	Imp.
Case1 (110-nm)	0.324	0.410	+26.54%
Case2 (110-nm)	0.289	0.320	+10.72%
Case3 (110-nm)	-1.035	-0.84	+18.84%
Case4 (110-nm)	-0.796	-0.788	+1.01%
Case5 (110-nm)	0.335	0.454	+35.52%
Case6 (28-nm)	0.086	0.102	+6.97%
Case7 (28-nm)	-0.063	-0.051	+19.04%
Case8 (28-nm)	0.091	0.110	+20.87%
Case9 (28-nm)	-0.886	-0.778	+12.18%
Case10 (28-nm)	-0.974	-0.884	+4.33%

- Wu et. al., AiTO: Simultaneous gate sizing and buffer insertion for timing optimization with GNNs and RL, Integration, 2024

[Integration-24]

AiSTA: AI for Timing Analysis

- **Fit delay calculation, fast and precise computation of path delays.**



$$Path_{Delay} = Cell_{Delay} + Net_{Delay}$$

$$\begin{cases} C\dot{x}(t) + Gx(t) = Bu(t), \\ y(t) = L^T x(t), \end{cases}$$

98% paths, the deviation is within 3%.

Category	Features
Cell delay	Input transition
	Load capacitance
	Signal polarity
	Cell type
	Cell port name
Wire delay	Output transition
	Resistance
	Capacitance

Basic Feature

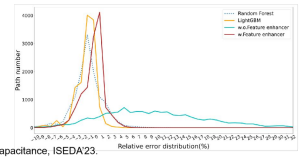
$$D_{Elmore} = \sum_{i \in N} (R_i * \sum C_D)$$

$$D_{D2M} = \frac{m_1^2}{\sqrt{m_2}} \ln 2$$

$$D_{ECM} = \sum_{i \in N} (R_i * \sum C_{left})$$

$$D_{MD2M} = \frac{m_1^2}{\sqrt{m_2}} \ln 2$$

Enhance Feature



- H. Liu, et. al, Accurate Timing Path Delay Learning using Feature Enhancer with Effective Capacitance, ISEDA'23.

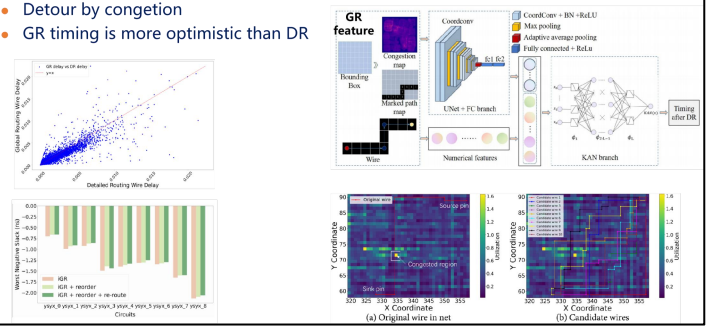
[ISEDA-23]

AI-EDA Models

- **7)** Timing Prediction and Optimization, **8)** IR Drop Calculation, **9)** Design Rule Checking and Prediction, **10)** Parameter Optimization, **11)** Logic Synthesis, **12)** Macro Placement

AiTPO: AI for Timing Prediction and Optimization

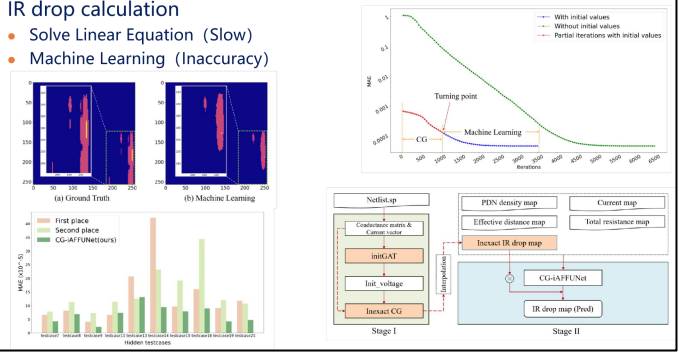
- Timing Gap between（GR） and （DR）
  - Detour by congetion
  - GR timing is more optimistic than DR



[TODAES-25]

AiIR: AI for IR Drop

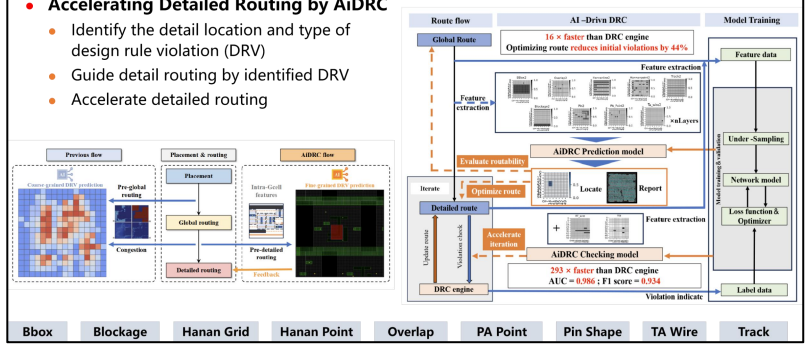
- IR drop calculation
  - Solve Linear Equation（Slow）
  - Machine Learning（Inaccuracy）



[ICCD-24]

AiDRC: AI for DRC

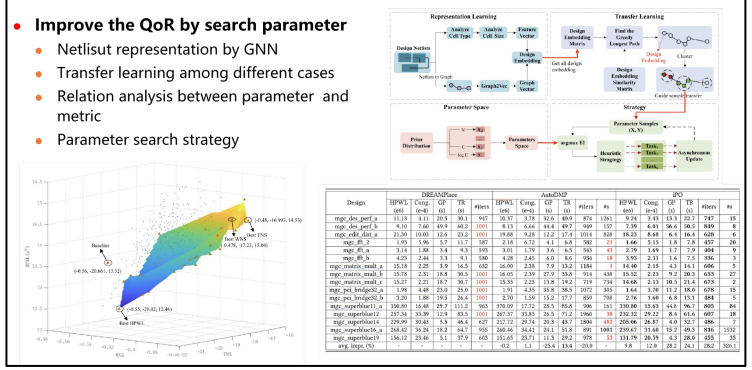
- **Accelerating Detailed Routing by AiDRC**
  - Identify the detail location and type of design rule violation (DRV)
  - Guide detail routing by identified DRV
  - Accelerate detailed routing



[TODAES-25]

AiPO: AI for Parameter Optimization

- **Improve the QoR by search parameter**
  - Netlisut representation by GNN
  - Transfer learning among different cases
  - Relation analysis between parameter and metric
  - Parameter search strategy

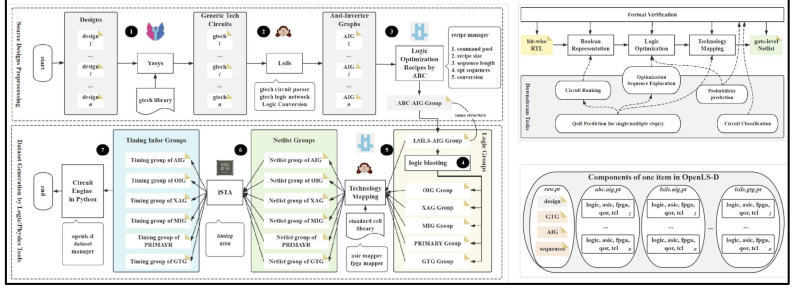


Design	DREAMPlace HPWL (m)	DREAMPlace Cong. (m-d)	DREAMPlace OF (s)	DREAMPlace TR (s)	DREAMPlace #Iters	AutoDMP HPWL (m)	AutoDMP Cong. (m-d)	AutoDMP OF (s)	AutoDMP TR (s)	AutoDMP #Iters	AutoDMP #x	IPO HPWL (m)	IPO Cong. (m-d)	IPO OF (s)	IPO TR (s)	IPO #Iters	#x
mgc_des_perf_a	11.13	6.11	35.5	36.1	947	10.37	3.78	32.6	40.9	874	1261	9.24	3.43	19.3	22.7	747	15
mgc_des_perf_b	9.10	7.60	49.9	60.2	1001	8.13	6.66	44.4	49.7	949	157	7.39	6.81	36.6	50.5	949	8
mgc_edit_dist_a	21.30	10.03	12.6	23.2	1001	19.88	9.28	12.2	17.4	1014	818	18.23	8.68	6.9	16.4	628	6
mgc_fft_2	1.93	5.96	5.7	11.7	587	2.18	6.72	4.1	6.8	582	21	1.66	5.15	1.8	7.8	457	20
mgc_fft_a	3.14	1.88	3.4	9.3	593	3.01	1.79	3.6	6.5	545	43	2.79	1.69	1.7	7.9	404	9
mgc_fft_b	4.23	2.46	3.3	9.1	580	4.28	2.45	6.0	8.6	936	18	3.93	2.31	1.6	7.5	336	5
mgc_matrix_mult_a	15.18	2.25	3.9	16.5	632	16.00	2.38	7.9	13.2	1184	1	14.40	2.15	4.3	14.1	606	5
mgc_matrix_mult_b	15.76	2.31	16.8	36.5	1001	16.05	2.39	27.9	33.8	914	638	15.52	2.23	9.2	28.3	635	27
mgc_matrix_mult_c	15.27	2.21	18.7	36.7	1001	15.35	2.25	13.8	19.2	719	734	14.68	2.13	10.3	21.6	673	2
mgc_pci_bridge32_a	1.98	4.48	23.0	25.0	1001	1.91	4.35	35.8	38.5	1072	303	1.64	3.78	11.2	18.0	678	15
mgc_pci_bridge32_b	3.20	1.88	19.5	26.4	1001	2.70	1.59	15.2	17.7	859	708	2.76	1.60	6.8	15.1	484	5
mgc_superblue11_a	356.80	16.48	29.7	111.2	965	370.09	17.72	28.5	85.8	906	161	330.00	15.63	44.8	96.7	805	84
mgc_superblue12	257.34	33.39	12.9	83.3	1001	267.37	33.85	26.5	71.2	1960	38	232.32	29.22	8.4	61.6	607	18
mgc_superblue14	229.99	30.43	5.3	46.4	627	217.72	29.74	20.3	43.7	1836	682	205.06	26.87	4.0	32.7	486	7
mgc_superblue16_a	268.42	36.24	10.2	64.7	955	268.46	36.43	24.1	51.8	891	1001	239.67	31.60	15.2	49.5	816	1332
mgc_superblue19	156.12	23.46	5.1	37.9	605	151.65	23.71	11.3	29.2	978	53	131.79	20.59	4.3	28.0	455	35
avg. impr. (%)	--	--	--	--	--	-0.2	1.1	-25.4	13.4	-10.0	--	9.8	12.0	28.2	24.1	28.2	326.1

[TODAES-25]

AiLS: AI for Logic Synthesis

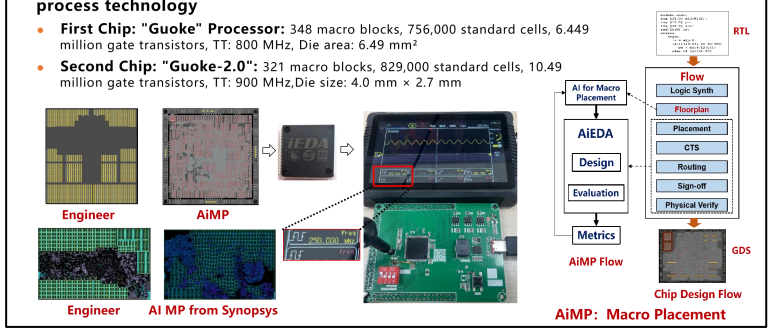
- RTL->Gtech->Netlist
  - Logic Translation, Logic Optimization, Tech Mapping



[TCAD-25]

AiMP: AI for Macro Placement

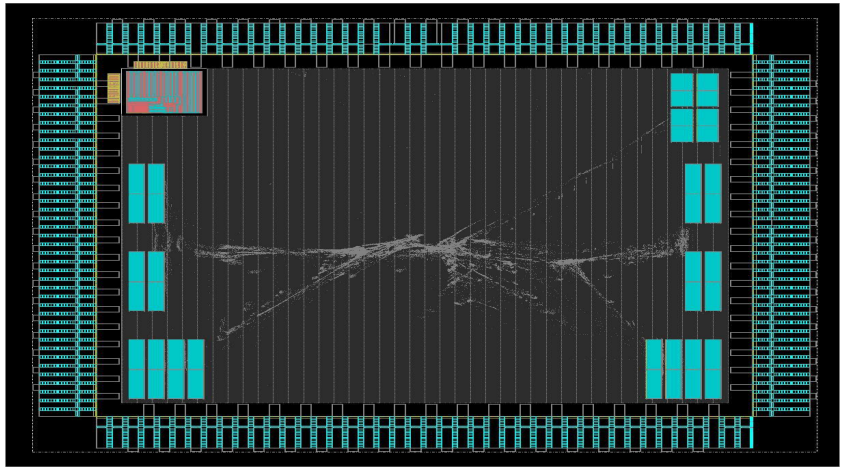
- **AiMP: complete the design of 2 processor chips using 28nm process technology**
  - **First Chip: "Guoke" Processor:** 348 macro blocks, 756,000 standard cells, 6.449 million gate transistors, TT: 800 MHz, Die area: 6.49 mm²
  - **Second Chip: "Guoke-2.0":** 321 macro blocks, 829,000 standard cells, 10.49 million gate transistors, TT: 900 MHz,Die size: 4.0 mm × 2.7 mm



Conclusion

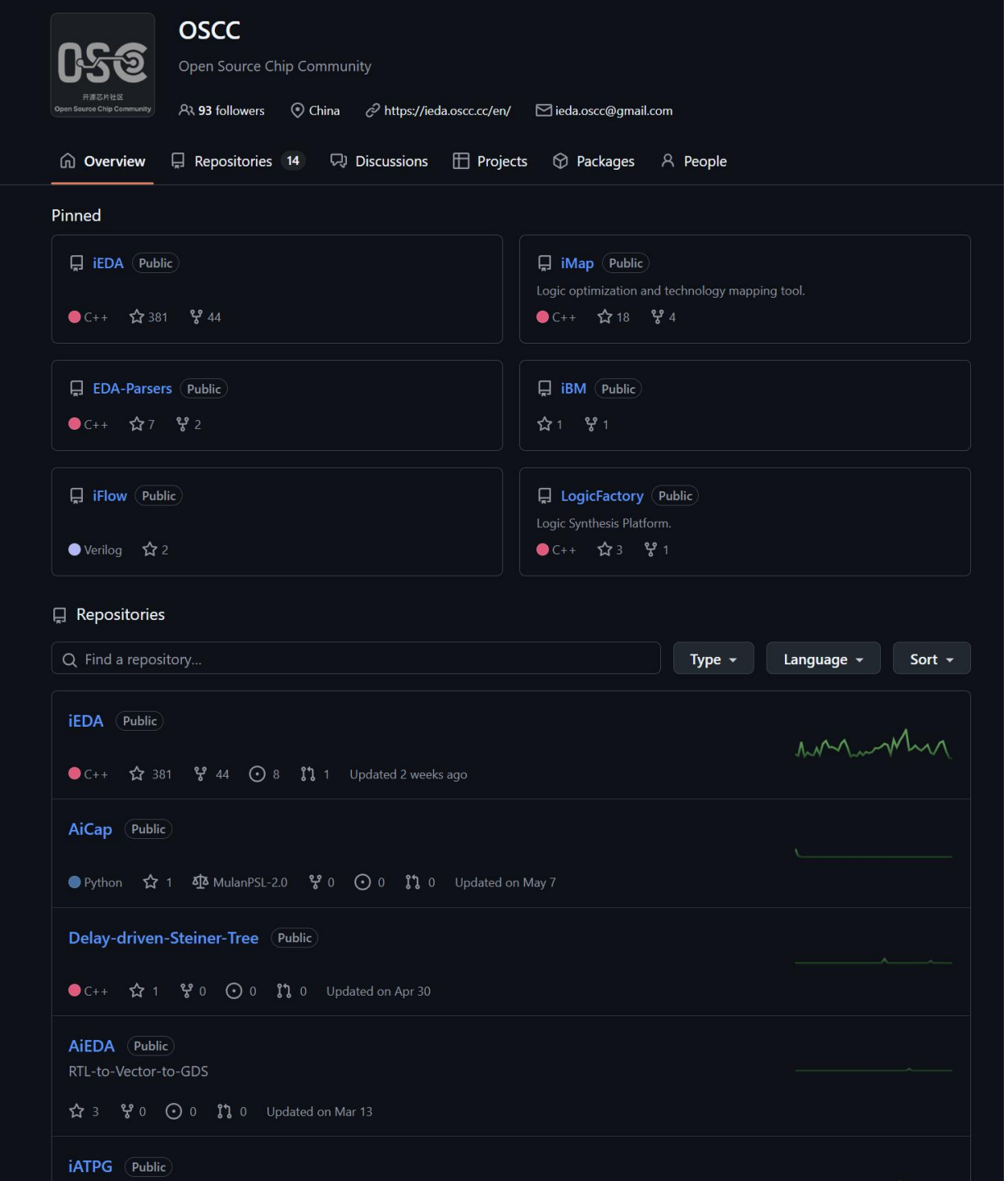
- **iEDA Objective**
  - **EDA Infrastructure**
  - **Explore new** and efficient **EDA R&D method**
  - **High quality and performance EDA tool**

- **Open-source：(Gitee/Github)**
  - **GitHub: https://github.com/OSCC-Project/iEDA**
  - **Gitee: https://gitee.com/oscc-project/iEDA**



Open-source is not a goal but a way

- **Papers**
  - **iEDA:** An Open-source Intelligent Physical Implementation Toolkit and Library, ISEDA, 2023. (BPA)
  - **OpenLS-DGF:** An Adaptive Open-Source Dataset Generation Framework for Machine Learning Tasks in Logic Synthesis, TCAD, 2025.
  - **iPL-3D:** A Novel Bilevel Programming Model for Die-to-Die Placement, ICCAD, 2023.
  - **iEDA:** An Open-source infrastructure of EDA (invited), ASPDAC, 2024.
  - **iPD:** An Open-source intelligent Physical Design Tool Chain (invited), ASPDAC, 2024.
  - **iRT:** Net Resource Allocation: A Desirable Initial Routing Step, DAC, 2024.
  - **iCTS:** Toward Controllable Hierarchical Clock Tree Synthesis with Skew-Latency-Load Tree, DAC, 2024.
  - **iPL:** A Fast, Iterative Clock Skew Scheduling Algorithm with Dynamic Sequential Graph Extraction， DAC, 2025.



Thanks

iEDA

Xingquan Li
lixq01@pcl.ac.cn