

# iEDA: An Open-source infrastructure of EDA

Xingquan Li, Zengrong Huang, Simin Tao, Zhipeng Huang, Chunan Zhuang, Hao Wang, Yifan Li, Yihang Qiu, Guojie Luo, Huawei Li, Haihua Shen, Mingyu Chen, Dongbo Bu, Wenxing Zhu, Ye Cai, Xiaoming Xiong, Ying Jian, Yi Heng, Peng Zhang, Bei Yu, Biwei Xie, Yungang Bao

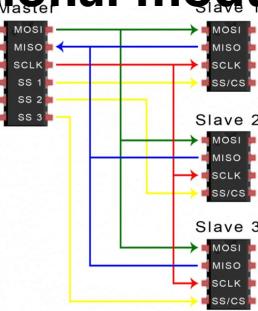
**Jan. 23 2024**



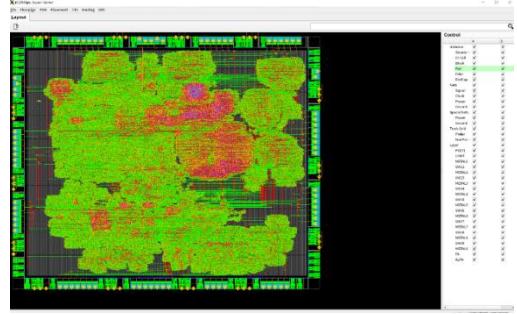
-  **01** **Introduction**
-  **02** **iEDA**
-  **03** **iEDA Application**

# Chip Design Elements

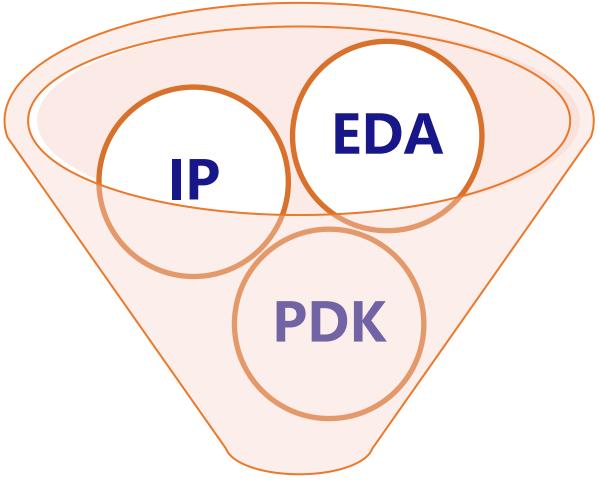
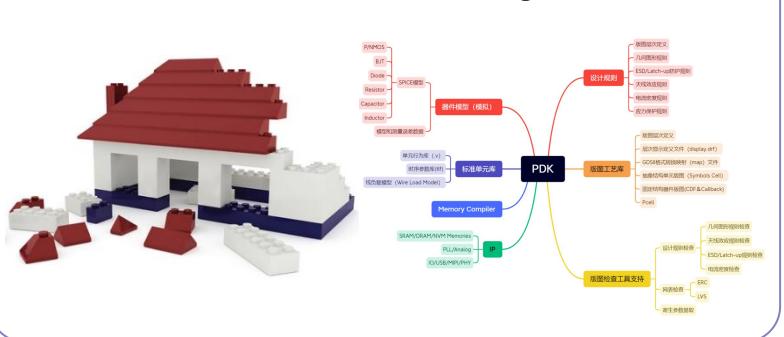
## IP: Intellectual Property Functional module



## EDA: Electronic design automation software (tool)



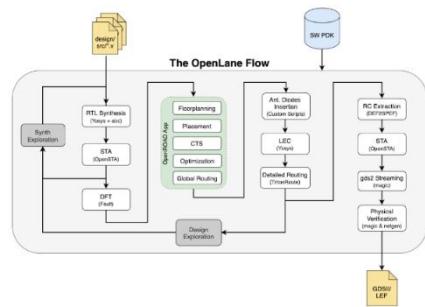
## PDK: Process design kits from foundry



Flow



## Flow: Chip design flow config and script



# Open-source RTL, PDK and Flow

## Open-source RTL

Accelerator	Analog	Connectivity	CPU	FPGA	Memory	System
aes/aes_core	<a href="#">AMS_KGD</a>	aib	<a href="#">OpenXiangShan</a>	<a href="#">FABulous</a>	<a href="#">core_axi_cac_he</a>	<a href="#">Beagle_SDR_GPS</a>
ara	<a href="#">open-pmic</a>	aib-protocols	<a href="#">a2i</a>	<a href="#">fabric_team</a>	<a href="#">HuanCun</a>	<a href="#">bsg_manycore</a>
FFTGenerator	<a href="#">Analog Basic Blocks/LDO</a>	core_ddr3_controller	<a href="#">black-parrot</a>	<a href="#">OpenFPGA</a>	<a href="#">openram</a>	<a href="#">cep</a>
fpu		HDMI	Cores-SweRV	<a href="#">prga</a>	<a href="#">lake</a>	<a href="#">esp</a>
garnet		i2c		<a href="#">core-v-verif</a>		<a href="#">hero</a>
gplgpu		litedram		<a href="#">cva6</a>		<a href="#">litex</a>
core_jpeg		liteeth		<a href="#">cv32e40p</a>		<a href="#">openFASOC</a>
vortex		litepice		<a href="#">ibex</a>		<a href="#">openpiton</a>
VeriGPU		litescope		<a href="#">microwatt</a>		<a href="#">opentitan</a>
tvm-vta		pymtl3-net		<a href="#">neorv32</a>		<a href="#">openwifi-hw</a>
...		verilog-ethernet		<a href="#">picorv32</a>		<a href="#">pulp</a>
		ravenoc		<a href="#">rocket-chip</a>		<a href="#">pulpissimo</a>
		verilog-uart		<a href="#">serv</a>		
				<a href="#">snitch</a>		
			boom			
			Low-RISC			
			OpenXuantie - OpenC910 Core			
			ysyx			

## Open-source PDK

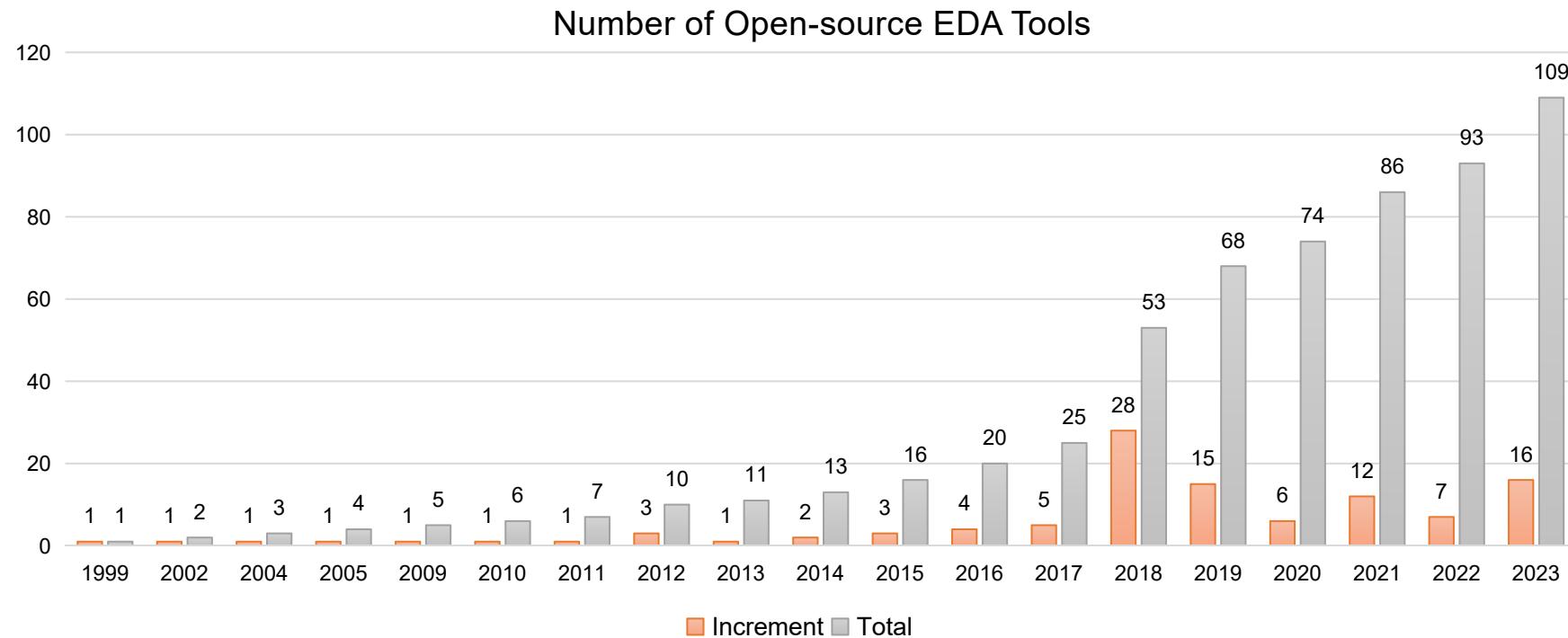
PDK name	Process node	Foundary	Institution
Sky130	130nm	Skywater	Google/Efabless
Sky90	90nm	Skywater	Google/Efabless
gf180	180nm	GlobalFoundries	Google/Efabless
NanGate45	45nm	Fake	Si2
Asap7	7nm	Fake	ARM Ltd

## Open-source Flow

Flow	Function	Contributor Institution
Qflow	RTL-GDS	Efabless
VSDFLOW	RTL-GDS	VLSI System Design
OpenRoad	RTL-GDS	UCSD
OpenLane	RTL-GDS	Efabless/UCSD
<a href="#">Ophidian</a>	Netlist-GDS	UFSC
Rsyn	Physical Synthesis	FURG
SiliconCompiler	RTL-GDS	Zero ASIC
SymbiFlow	FPGA design flow	F4PGA
<b>iFlow</b>	<b>RTL-GDS</b>	<b>PCL/ICT/BOSC/...</b>

# Increasing Open-source EDA Tools

- Open-source in EDA may be a tendency

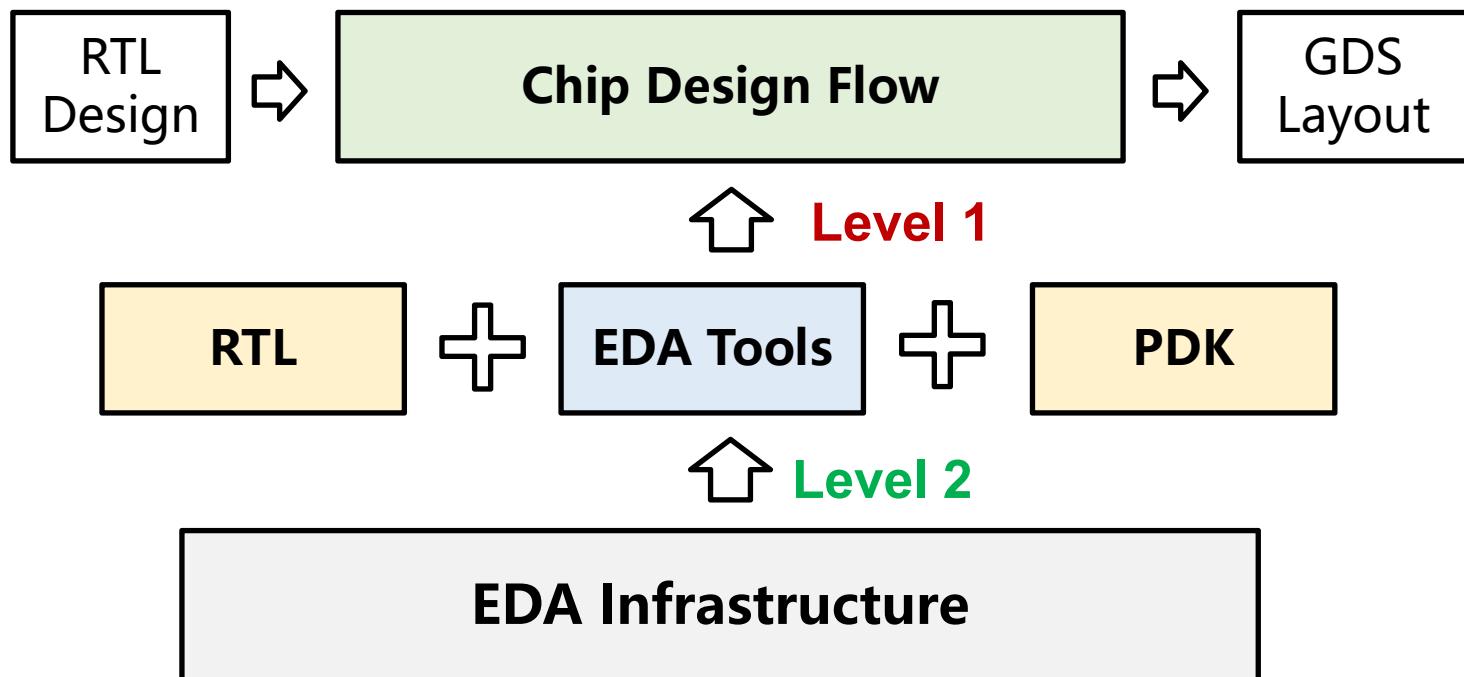


# Open-source EDA Tools

Design Module	Design Step	Some Open-source Tools						iEDA
HLS	HLS	LegUp	GAUT	PandA	FCUDA	XLS		
	Logic Simulation	Verilator/iVerilog	GHDL	FreeHDL	TkGate			
	Circuit Simulation	NGSpice	mixedsim	GnuCap	Qucs	XICE		
	Debug							
Logic Synthesis	Logic Synthesis	Yosys	ABC	EPFL-LS-Lib	LLDHL	UNIVR	iLS	
	Tech Map	ABC					iMap	
DFT	DFT	Fault						
	Formal							
Physical Design	Partition	PartitionMgr	METIS	KaHyPar	MPPart		iNO	
	Floorplan	OpenRoad TritonMacroPlacer	OpenRoad	Parquest			iFP	
	PDN	OpenPDN					iPDN	
	Placement	RePlace OpenDP	DreamPlace	Graywolf	Capo		iPL	
	CTS	TritonCTS					iCTS	
	Timing OPT	TritonSizer	Gate-Sizing				iTO	
	Routing	FastRoute TritonRoute	CUGR Dr.CU	Qrouter	NTHU-Route	BoxRouter/FGR/ORGE	iRT	
	ECO	OpenRoad-eco					iECO	
	STA	OpenSTA	OpenTimer				iSTA	
	RCX	OpenRCX	SPEF-Extractor				iRCX	
Signoff	Power	OpenRoad-pp					iPA	
	IR Drop	PDNSim	IREDGe				iIR	
	DRC	Klayout	Magic				iDRC	
	Antenna	OpenRoad-ant						
Physical Verification	LVS	Netgen						
	MPL							
	RET/ILT							
	Mask Generation							

# We Need Infrastructure

- **Level 1:** Open-source tools, RTLs, PDKs support chip design
- **Level 2:** Open-source Infrastructure supports EDA development



-  **01** **Introduction**
-  **02** **iEDA**
-  **03** **iEDA Application**

# iEDA Introduction

- **iEDA Objective**
  - EDA Infrastructure
  - Explore new and efficient EDA R&D method
  - High quality and performance EDA tool
- **Open-source: (Gitee/Github)**
  - Gitee: <https://gitee.com/oscc-project/iEDA>
  - GitHub: <https://github.com/OSCC-Project/iEDA>



**Open-source is not a goal but a way**

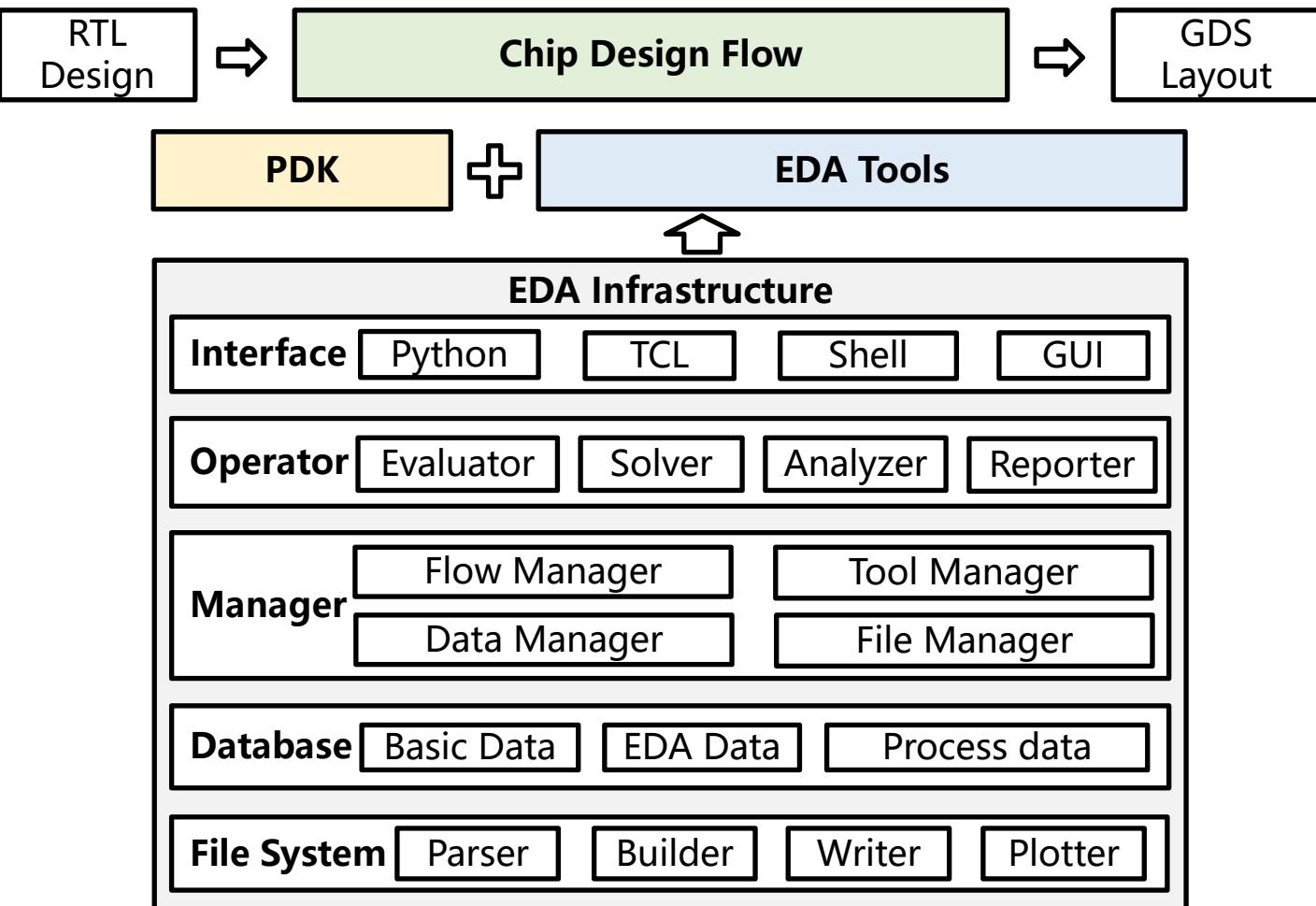
EDA Tools

↑ Level 2

EDA Infrastructure

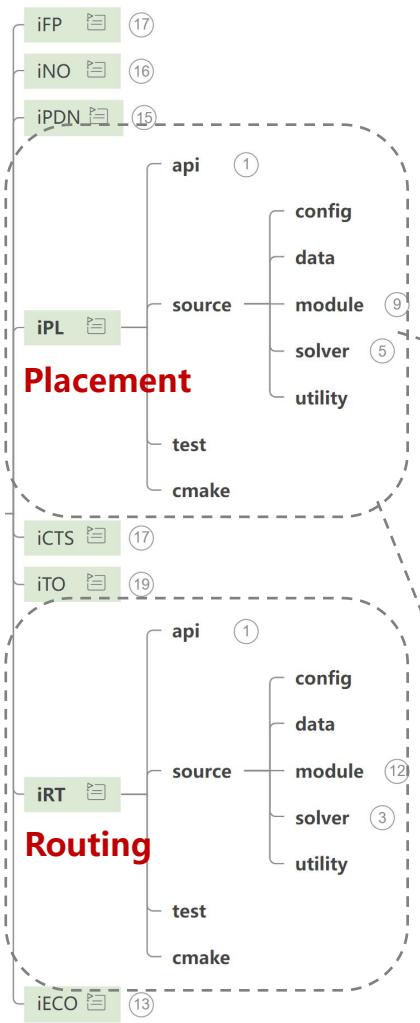
# iEDA-Infrastructure

- **File System**
- **Database**
- **Manager**
- **Operator**
- **Interface**
- **Utility**
- **Some perf tools**

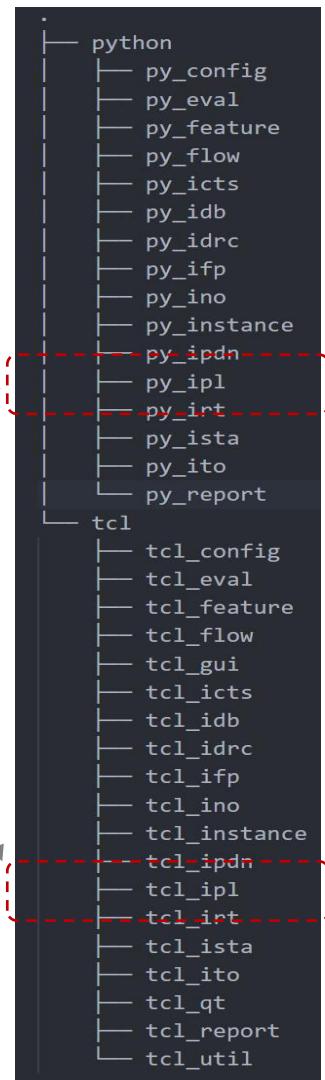


# Uniform Software Framework and API

## Software Structure



## API



## Application

```
def run_iPL(self):
    ieda.flow_init(config=".iEDA_config/flow_config.json")
    ieda.db_init(config=".iEDA_config/db_default_config.json")
    ieda.db_init(sdc_path = "./sdc/asic_top_SYN_MAX_1.sdc")
    ieda.def_init(path=".result/iTO_fix_fanout_result.def")
    ieda.run_placer(config=".iEDA_config/pl_default_config.json")
    ieda.def_save(path=".result/iPL_result.def")
    ieda.netlist_save(path=".result/iPL_result.v")
    ieda.report_db(path=".result/report/pl_db.rpt")
    ieda.flow_exit()
```

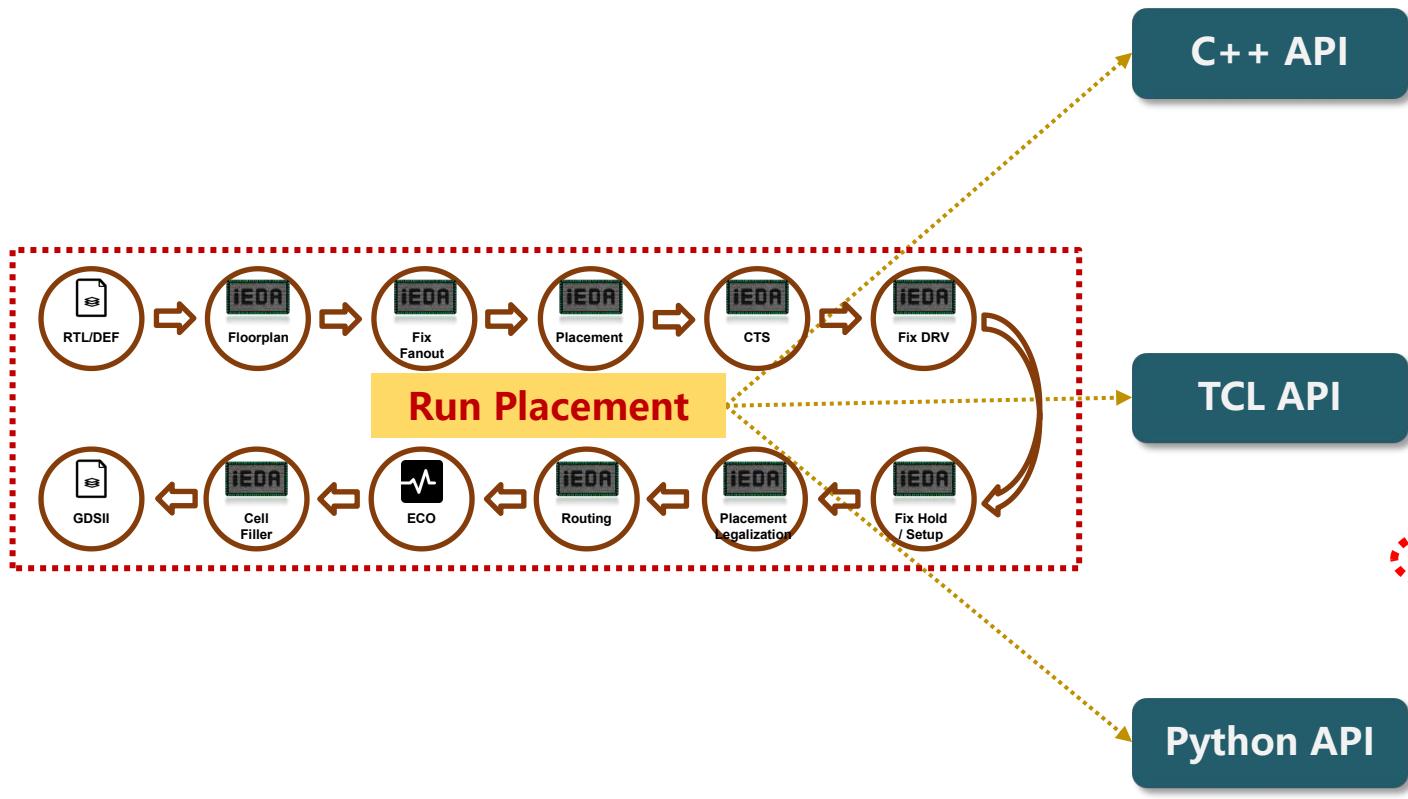
### Python

```
1 flow_init -config ./iEDA_config/flow_config.json
2 db_init -config ./iEDA_config/db_default_config.json
3 source ./script/DB_script/db_path_setting.tcl
4 source ./script/DB_script/db_init_sdc.tcl
5 source ./script/DB_script/db_init_lef.tcl
6 def_init -path ./result/iTO_fix_fanout_result.def
7 run_placer -config ./iEDA_config/pl_default_config.json
8 def_save -path ./result/iPL_result.def
9 netlist_save -path ./result/iPL_result.v -exclude_cell_names {}
10 report_db -path "./result/report/pl_db.rpt"
11 flow_exit
```

### TCL

# Multiple Programming Language

- ✓ Support **C++**、**RUST**、**TCL**、**Python**



```
/// run placer
if (PLFConfig::getInstance()->is_run_placer()) {
    if (tmInst->autoRunPlacer(PLFConfig::getInstance()->get_ipl_path()))
}
}
```

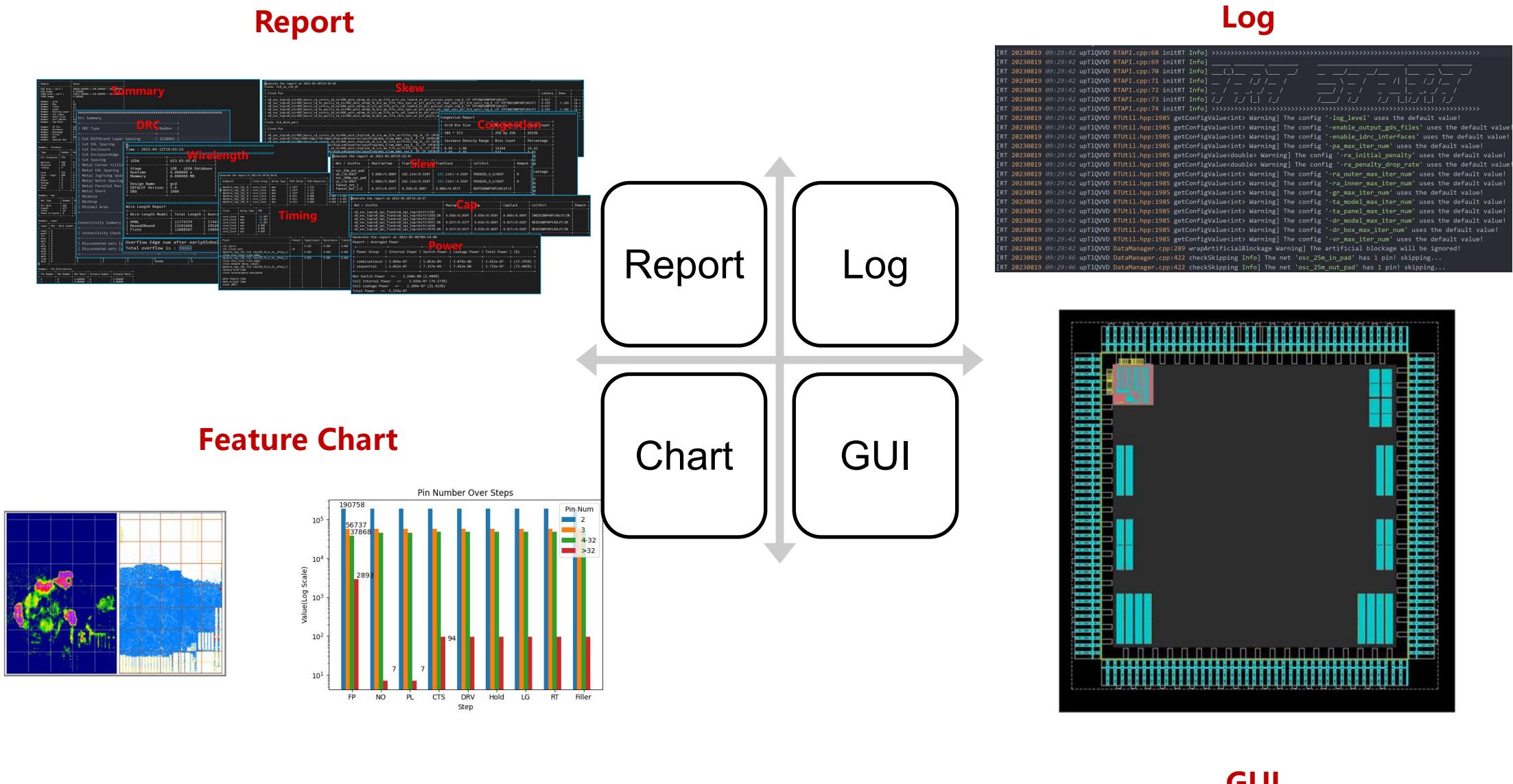
```
#####
##  read def
#####
def_init -path ./result/iTO_fix_fanout_result.def

#####
##  run Placer
#####
run_placer -config ./iEDA_config/pl_default_config.json
```

```
def run_placer(self, input_def : str):
    self.read_def(input_def)

    path = self.path_manager.get_workspace().get_config_ieda(FlowStep.place)
    ieda.run_placer(path)
```

# Multiple Data Analysis and Debug Methods



# Data Snapshot & Recovery

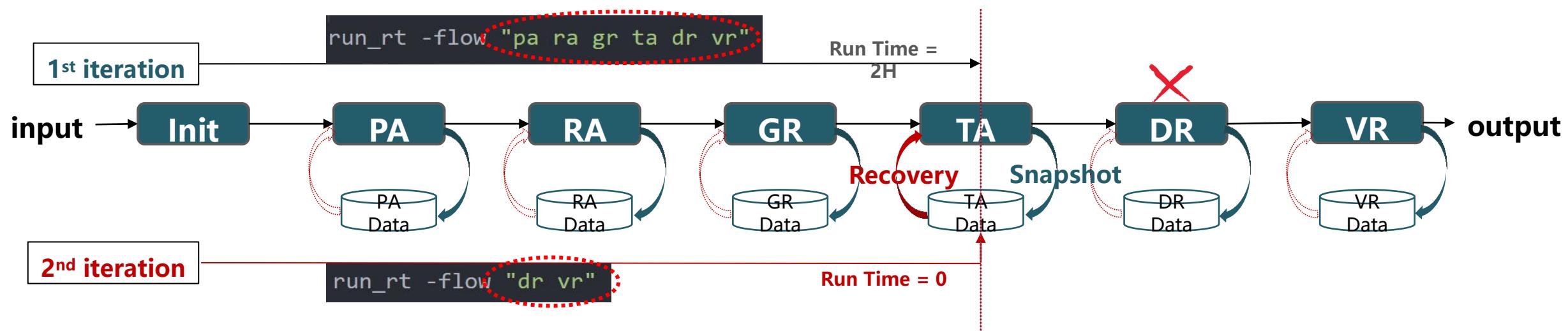
- iEDA adopts serialization and deserialization to achieve data snapshot and recovery:

## Data Snapshot

- Tool Process Data
- Result Data
- Algorithm Data

## Data Recovery

- Tool Process Data
- Result Data
- Algorithm Data



# Rich API and Documentation

## C++ API Doc

API list		
API Command	Type	Description
buildRCTree		
initRcTree		
initRcTree		
resetRcTree		
buildGraph		
isBuildGraph		
resetGraph		
resetGraphData		
insertBuffer		
removeBuffer		
repowerInstance		
moveInstance		
writeVerilog		
setSignificantDigits	builder	set the significant digits of the timing report
incrUpdateTiming	action	incremental propagation to update the timing data
updateTiming	action	update the timing data

Doc Link: <https://gitee.com/ieda-ipd/iEDA/tree/master/docs>

## User Manual

```

script
script|



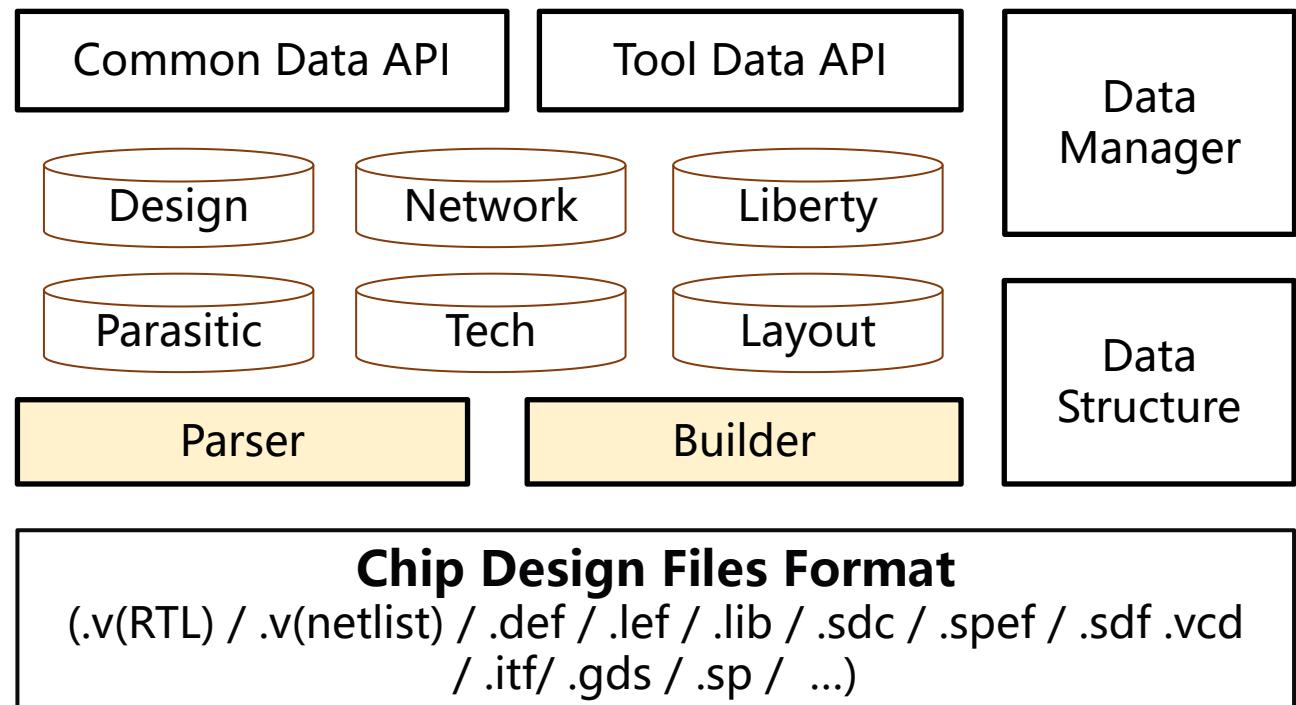
scripts/design/sky130_gcd/script
└── DB_script
    ├── db_init_lef.tcl
    ├── db_init_lib_drv.tcl
    ├── db_init_lib_fixfanout.tcl
    ├── db_init_lib_hold.tcl
    ├── db_init_lib_setup.tcl
    ├── db_init_lib.tcl
    ├── db_init_sdc.tcl
    ├── db_init_spref.tcl
    ├── db_path_setting.tcl
    ├── run_db_checknet.tcl
    ├── run_db_report_eval.tcl
    ├── run_db.tcl
    ├── run_def_to_gds_text.tcl
    ├── run_def_to_verilog.tcl
    ├── run_netlist_to_def.tcl
    └── run_read_verilog.tcl
    # Data process flow scripts
    # initialize lef
    # initialize lib only for flow of drv
    # initialize lib only for flow of fix fanout
    # initialize lib only for flow of optimize hold
    # initialize lib only for flow of optimize setup
    # initialize lib for common flow
    # initialize sdc
    # initialize spref
    # set paths for all processing technology files, including TechLEF, LEF, Lib, sdc and spref
    # check net connectivity based on data built by DEF (.def) and LEF (.lef & .tlef)
    # report wire length and congestion based on data built by DEF (.def) and LEF (.lef & .tlef)
    # test building data by DEF (.def) and LEF (.lef & .tlef)
    # transform data from DEF (.def) to GDSII (.gdsii)
    # transform data from DEF (.def) to netlist (.v)
    # transform data from netlist (.v) to DEF (.def)
    # test read verilog file (.v)
    # CTS flow scripts
    # report wire length for CTS result
    # report CTS STA
    # run CTS
    # DRC(Design Rule Check) flow scripts
    # show GUI for DRC result
    # run DRC
    # Floorplan flow scripts
    # submodule for Floorplan scripts
    # create tracks for routing layers
    # set clock net
    # run Floorplan
    # GUI flow scripts
    # run GUI
    # NO(Netlist Optimization) flow scripts
    # run Fix Fanout
    # Placement flow scripts
    # report congestion statistics and wire length for Placement result
    # run standard cell filler
    # run gui flow that shows Global Placement Processing result
    # iEDA flows for different designs
    # TBD
    # flow of gcd in sky130
    # iEDA parameters configuration files
    # IEDA result output files
    # Python3 script for running all iEDA flow with GUI layout
    # Python3 script for running all iEDA flow
    # POSIX shell script for running all iEDA flow

iEDA
文件 (4734)
├── .gitee
├── cmake
├── docs
│   ├── api
│   ├── paper
│   ├── ppt
│   ├── resources
│   ├── tbd
│   ├── tcl
│   └── user_guide
│       ├── pic
│       └── iEDA_user_guide.md
├── scripts
├── src
├── .clang-format
├── .clang-tidy
├── .gitignore
├── CMakeLists.txt
├── LICENSE
├── README-En.md
├── README.md
└── build.sh

```

# Perser and Database

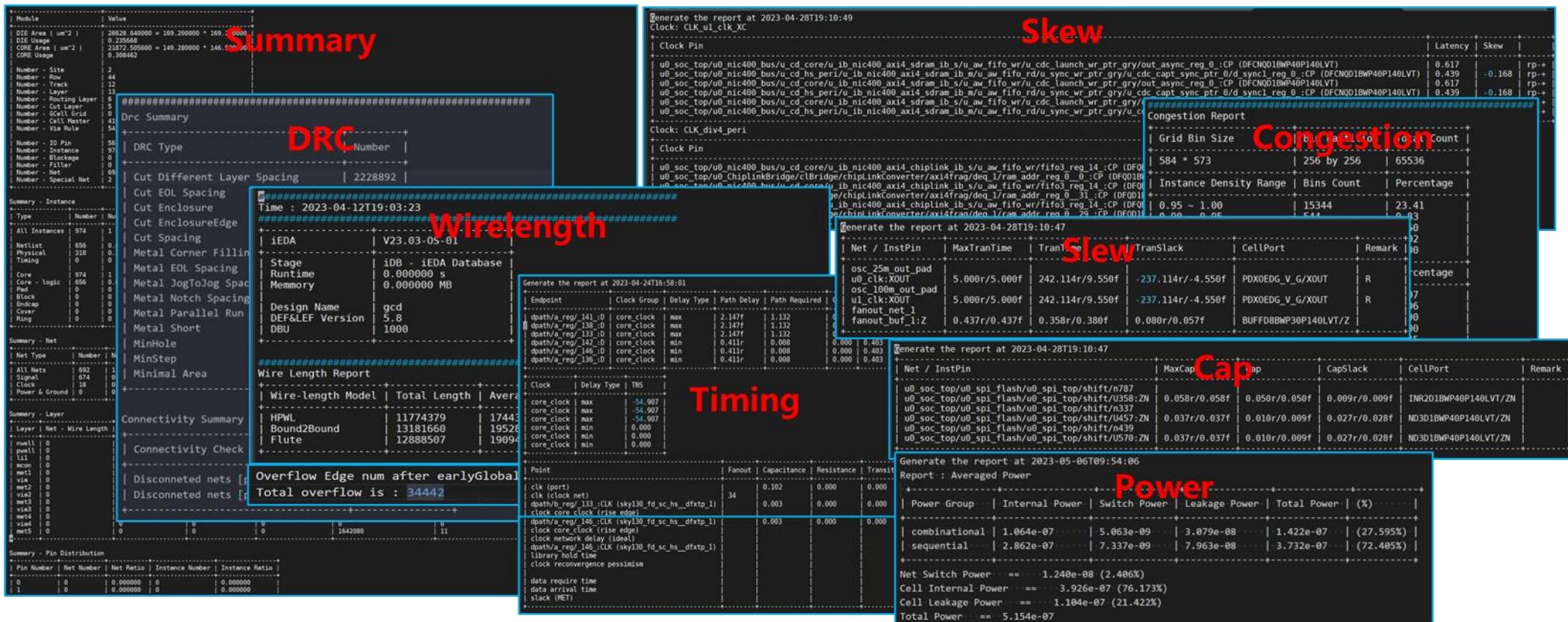
- **Parser: Verilog, SPEF, Liberty, SDF, VCD, SDC, LEF/DEF, ITF, and GDSII**
- **Database: Design, Layout, Tech, Timing, Parasitic, Network**



# Reporter

- Metrics

- Timing, power, area, wirelength, congestion, skew, res, cap, slew, fanout, utilization, density, cells, nets, drc (metal, cut, connection), ...



# Evaluator: Horizontal Comparison

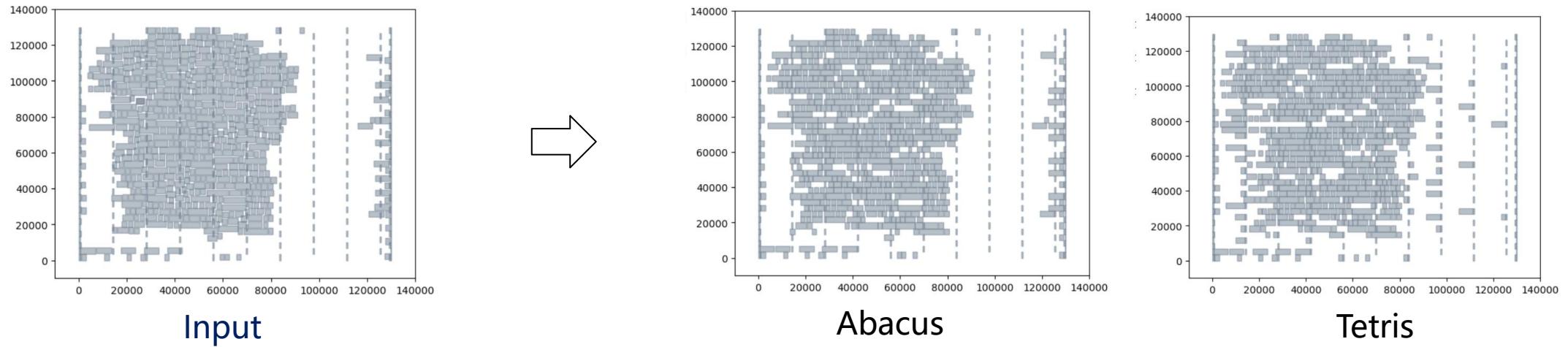
- Compare and analyze the Q&R of designs, tools, algorithms and flows

<b>design</b>	<b>aes</b>	<b>aes_core</b>
PDK	sky130	sky130
instance area	408034.7568	371050.9776
IO pin	76	520
instances	45854	42044
nets	30634	28536
core_area	1352765.88	1230601.766
total wire length	2695657	2809505
total vias	280870	271884
setup_slack (max)	14.7	14.73
hold_slack (min)	0.22	0.4
suggest freq (MHz)	188.6792453	189.7533207

**Design Comparison**

# Evaluator: Horizontal Comparison

- Compare and analyze the Q&R of designs, tools, algorithms and flows



## Algorithm Comparison

part metrics	abacus	tetris
global placement HPWL	10127910	10127910
legalization HPWL	10426323	13168231
detail placement HPWL	9901517	10928985
detail placement STWL	10637190	11674987
maximum STWL	431085	415325
total movement	795829	8705103
maximum movement	5684	218214
average congestion	0.8215	0.8134
total overflow	49	49
peak bin density	1	1
legalization runtime (s)	0.0667	0.0064

# Evaluator: Horizontal Comparison

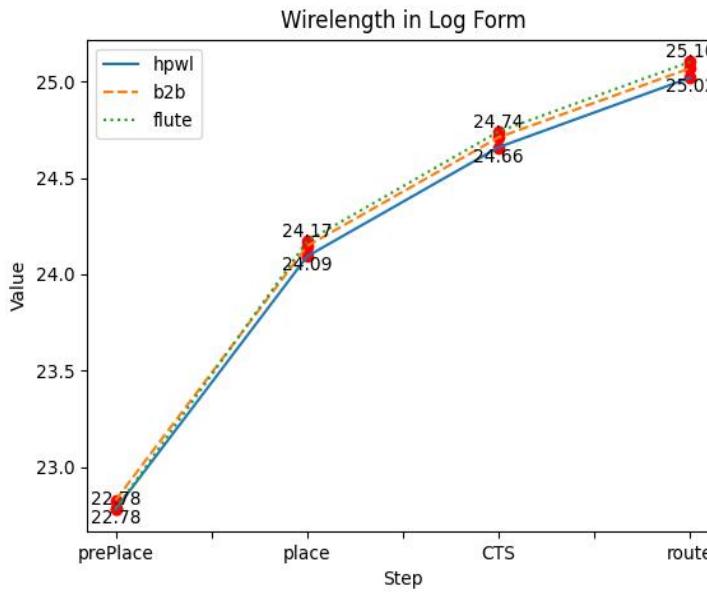
- Compare and analyze the Q&R of designs, tools, algorithms and flows

part metrics	flow1	flow2
detail routing HPWL (um)	10879081	11025675
final wirelength (um)	11471595	12071042
setup slack (ns)	-0.492	-0.484
hold slacke (ns)	0.426	0.427
suggest frequency (MHz)	345.804	346.784
power (mW)	0.956	0.966
#DRC	755	643

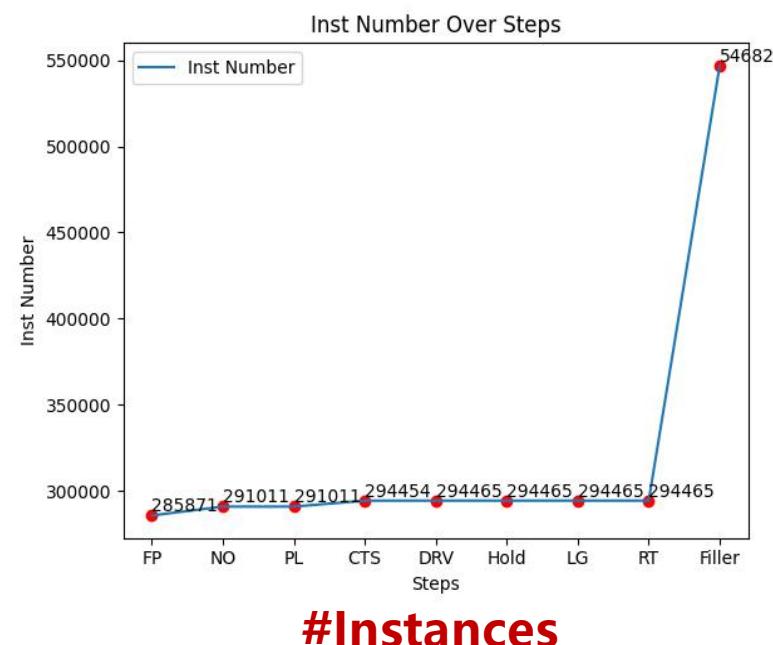
## Flow Comparison

# Analyzer : Vertical Comparison

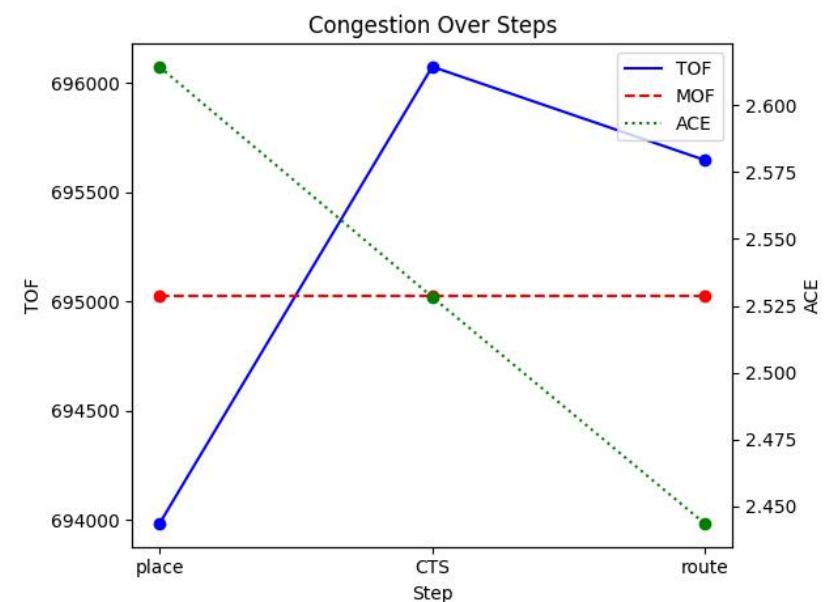
- Analyzing the numerical changes of an indicator **across different stages**
- Differences from: 1) data change, and 2) differences evaluation models
- Usage: evaluating the design quality, analyzing the margin, and optimizing collaboratively



Wirelength



#Instances

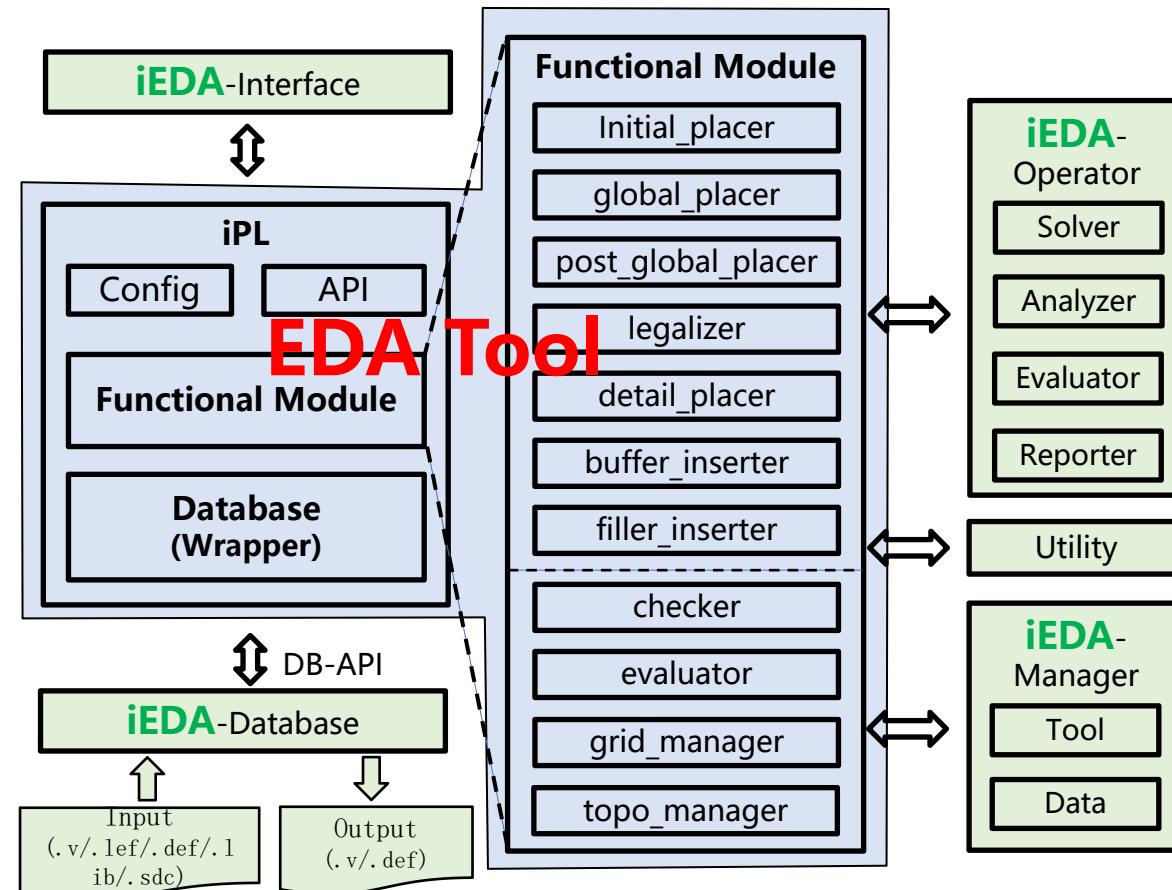


Congestion

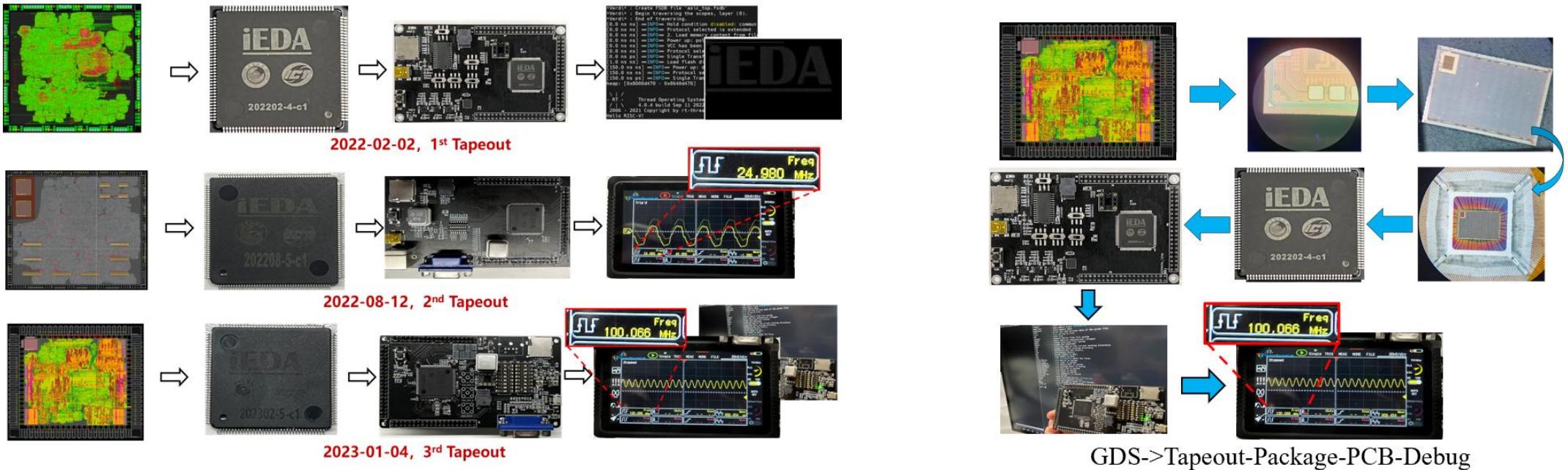
-  **01** **Introduction**
-  **02** **iEDA**
-  **03** **iEDA Application**

# Developing EDA Tool or Algorithm

- To fast develop high-quality EDA tool, we need a Software Development Kit (SDK)
- iEDA can be used to support developing EDA tool or algorithm



# Designing Chip



## 1<sup>st</sup> Tapeout

- RTL: **ysyx(一生一芯)-03**
- PDK: 110nm
- Area: **3mm × 3.5 mm**
- Power: dynamic = 48mW, leakage = 7 mW
- Freq.: **25MHz**
- Scale: **>0.7M Gates**
- Features: 5 pipeline, Chiplink, UART, SPI, PCB clock, support RT-thread

## 2<sup>nd</sup> Tapeout

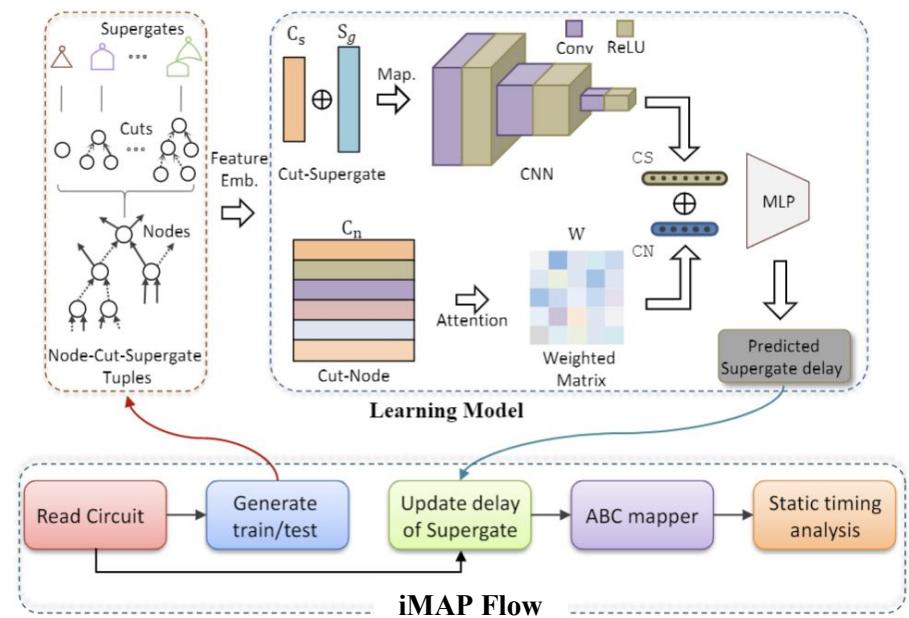
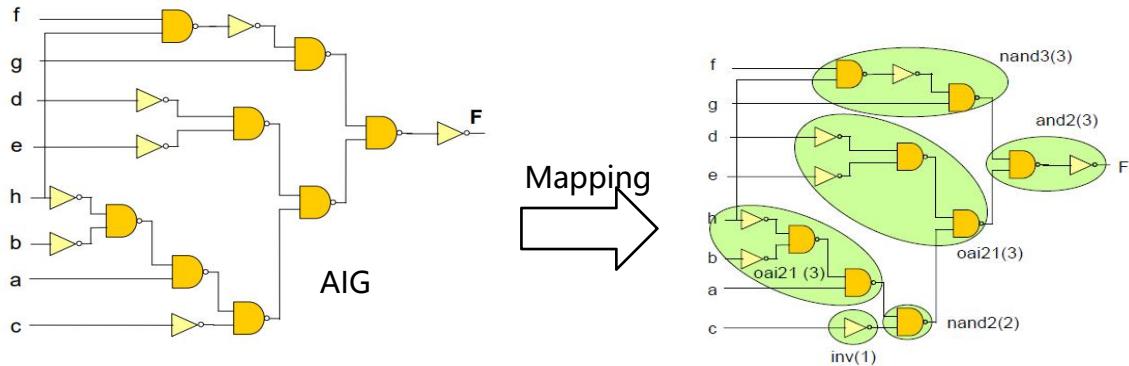
- RTL: **ysyx(一生一芯)-04**
- PDK: 110nm
- Area: **4.5mm × 4.5 mm**
- Power: dynamic = 343mW, leakage = **21 mW**
- Freq.: **25MHz**
- Scale: **>1.5M Gates**
- Features: 11 pipelines with cache, IP: UART, VGA, PS/2, SPI, SDRAM, 2 PLLs, support Linux

## 3<sup>rd</sup> Tapeout

- RTL: **ysyx(一生一芯)-04**
- PDK: 28nm
- Area: **1.5mm × 1.5 mm**
- Power: dynamic = **317mW**, leakage = **29 mW**
- Freq.: **200MHz**
- Scale: **>1.5M Gates**
- Features: 11 pipelines with cache, IP: UART, VGA, PS/2, SPI, SDRAM, 2 PLLs, support Linux

# Research: Learning to Optimize Tech Mapping

- Based on iMAP, we propose a ML method to predict cut delay



Circuits	Estimated Results		Actual Results		$\Delta$ Delay
	Area( $\mu\text{m}^2$ )	LI-Delay(ps)	Area( $\mu\text{m}^2$ )	Delay(ps)	
adder	898.31	2,613.78	898.13	3,770.65	44%
bar	2,681.62	152.96	2,680.39	1,114.9	629%
log2	26,556.98	3,891.66	26,561.26	6,797.77	75%
cavlc	463.27	185.07	463.29	93.2	50%
int2float	158.61	174.27	158.63	91.7	47%
ctrl	106.92	98.53	106.84	89.9	9%

LI-Delay refers to the load-independent delay estimation in ABC [1]. The actual delay is computed by the non-linear delay model.

# Talent Training: Curriculum Training Platform

- **Min Wirelength Model**

$$\begin{aligned} \min_{\boldsymbol{v}} \quad & W(\boldsymbol{v}) \\ \text{s.t.} \quad & \rho_b(\boldsymbol{v}) \leq \rho_0, \quad \forall b \in B \end{aligned}$$

- where  $\boldsymbol{v}$  is cell location,  $W(\boldsymbol{v})$  is wirelength,  $\rho_b(\boldsymbol{v})$  is the area density in  $b \in B$ ,  $\rho_0$  is density threshold.

- **Nesterov Method Or Conjugate Gradient**

1. Given  $x_0, r_0 = Ax_0 - b, p_0 = -r_0$
2. For  $k = 0, 1, 2, \dots$  until  $\|r_k\| = 0$

$$\alpha_k = r_k^T r_k / p_k^T A p_k$$

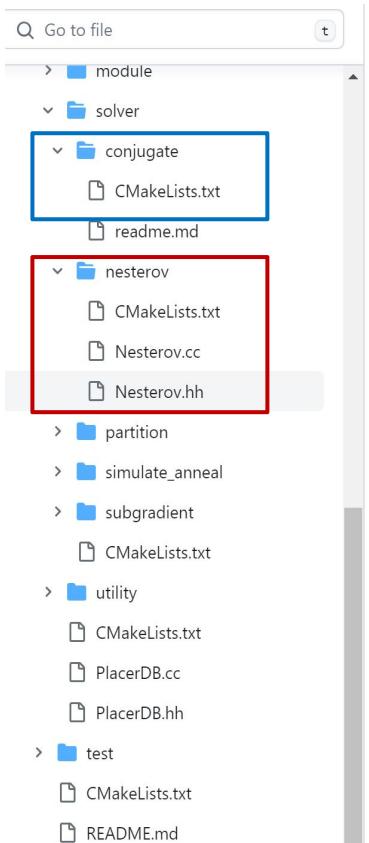
$$x_{k+1} = x_k + \alpha_k p_k$$

$$r_{k+1} = r_k + \alpha_k A p_k$$

$$\beta_{k+1} = r_{k+1}^T r_{k+1} / r_k^T r_k$$

$$p_{k+1} = -r_{k+1} + \beta_{k+1} p_k$$

- Assignment: please implement CG method by C++ or Python, and test it on “iEDA/iPL”, submit by PR to iEDA



```

38  class Nesterov
39  {
40  public:
41  Nesterov();
42  Nesterov(const Nesterov& other) = delete;
43  Nesterov(Nesterov&& other) = delete;
44  ~Nesterov() = default;
45
46 // getter.
47 int get_current_iter() const { return _current_iter; }
48 const std::vector<Point<int32_t>>& get_current_coordinis() const { return _current_coordinis; }
49 const std::vector<Point<float>>& get_current_grads() const { return _current_gradients; }
50 const std::vector<Point<float>>& get_next_grads() const { return _next_gradients; }
51 const std::vector<Point<int32_t>>& get_next_coordinis() const { return _next_coordinis; }
52 const std::vector<Point<int32_t>>& get_next_slp_coordinis() const { return _next_slp_coordinis; }
53 float get_next_stepLength() const { return _next_stepLength; }
54
55 // for RDP
56 const std::vector<Point<float>>& get_next_gradients() const { return _next_gradients; }
57 float get_next_parameter() const { return _next_parameter; }
58 void set_next_coordinis(const std::vector<Point<int32_t>>& next_coordinis) { _next_coordinis = next_coordinis; }
59 void set_next_slp_coordinis(const std::vector<Point<int32_t>>& next_slp_coordinis) { _next_slp_coordinis = next_slp_coordinis; }
60 void set_next_gradients(const std::vector<Point<float>>& next_gradients) { _next_gradients = next_gradients; }
61 void set_next_parameter(float next_parameter) { _next_parameter = next_parameter; }
62 void set_next_stepLength(float next_stepLength) { _next_stepLength = next_stepLength; }
63
64 // function.
65 void initNesterov(std::vector<Point<int32_t>> previous_coordinis, std::vector<Point<float>> previous_gradients,
66 std::vector<Point<int32_t>> current_coordinis, std::vector<Point<float>> current_gradients);
67 void calculateNextStepLength(std::vector<Point<float>> next_grads);
68
69 void runNextIter(int next_iter, int32_t thread_num);
70 void runBackTrackIter(int32_t thread_num);
71

```

# Talent Training: Support Contest

iEDA

文件 (4826)

.gitee

cmake

**contest**

docs

scripts

src

.clang-format

.clang-tidy

.gitignore

CMakeLists.txt

LICENSE

README-En.md

README.md

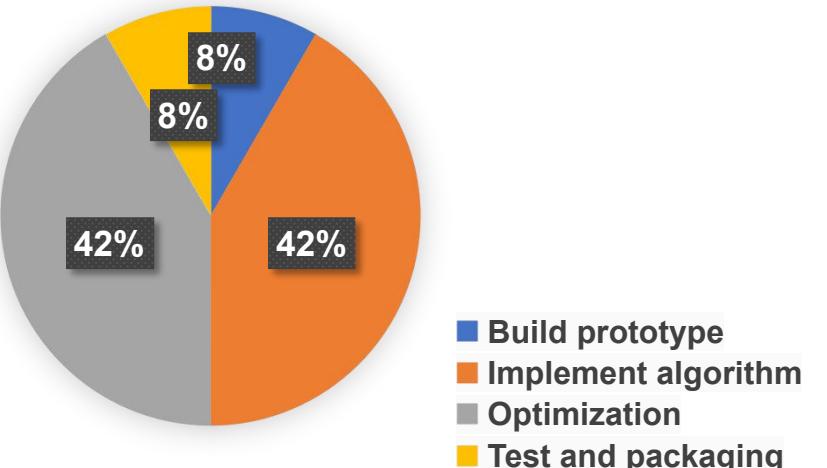
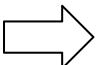
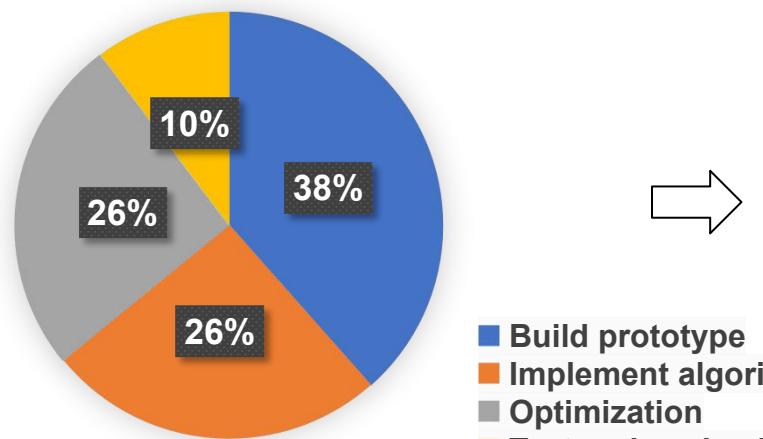
build.sh



**1<sup>st</sup> Place at  
2022  
ICCAD@  
Contest**

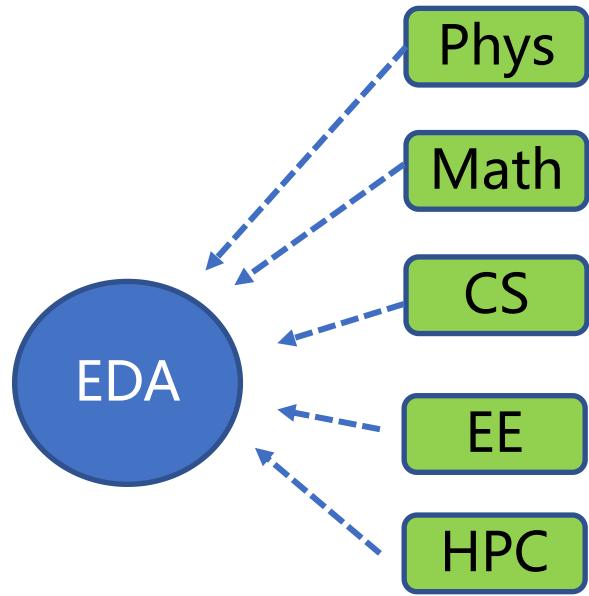


**2<sup>nd</sup> Place at  
2023  
ICCAD@  
Contest**

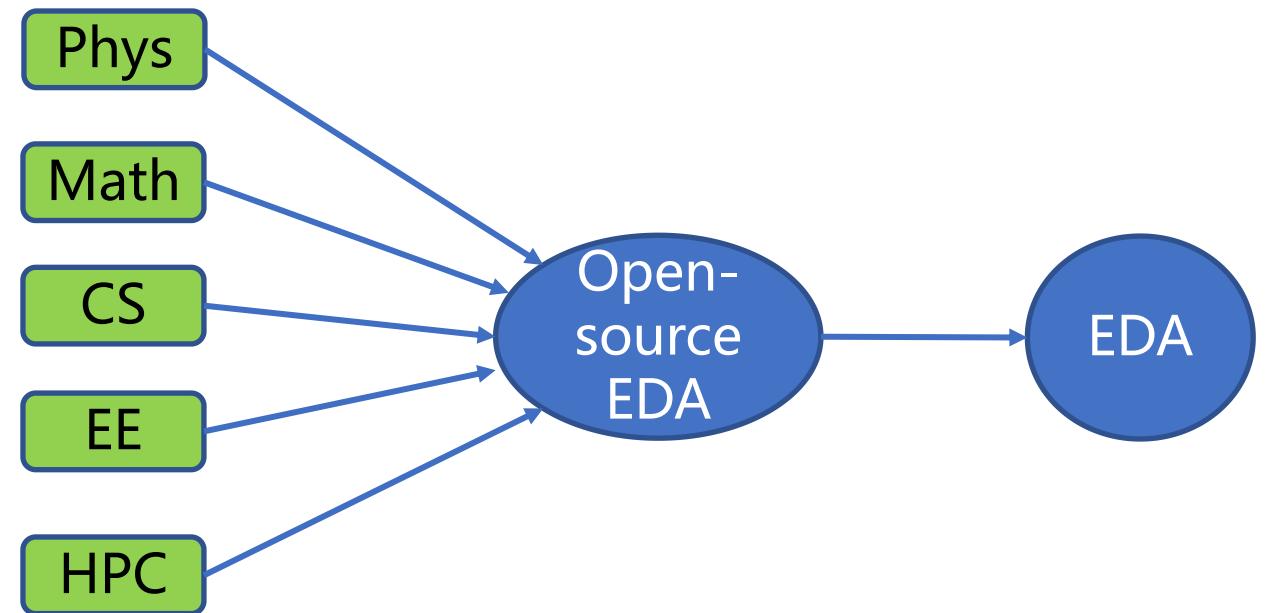


**Proportion of time spent**

# Closed vs. Open



**Closed-source EDA**



**Open-source EDA is a bridge**

# Conclusions

---

- **iEDA**

- EDA Infrastructure, explore EDA method, develop High quality EDA tool
- Uniform software, multiple language, data analysis and debug, data recovery, evaluation (vertical and horizontal)

- **iEDA Application**

- Develop EDA tool
- Design chip and tape-out
- EDA research
- Curriculum Training
- EDA Contest

# Future Works

---

- **Upcoming Solvers**

- ILP solver,
- SAT solver,
- Partition solver,
- Steiner tree solver,
- Geometric calculator,
- Non-linear program optimizer

- **Analyzer:** analyze and visualize process data

- **Performance:** improve data library efficiency

- **EDA Parsers:** implemented by RUST

OSCC-Project / iEDA

Type to search

Code Issues Pull requests Discussions Actions Projects Wiki Security Insights Settings

iEDA Public

master 2 Branches 0 Tags

Go to file Add file Code

About

No description, website, or topics provided.

Readme View license Activity 147 stars 2 watching 9 forks Report repository

Releases

No releases published Create a new release

Packages

No packages published Publish your first package

Contributors 18

File	Commit Message	Time Ago
.gitee	init repo of OSCC/EDA	8 months ago
cmake	feature:support python power in interface	4 months ago
docs	Merge branch 'master' of gitee.com:oscc-project/iEDA into ...	5 months ago
scripts	complete filter json data layer information	2 months ago
src	complete filter json data layer information	2 months ago
.clang-format	!1 update230508	8 months ago
.clang-tidy	!1 update230508	8 months ago
.gitignore	refactor:add power sort	4 months ago
CMakeLists.txt	delete contest project	4 months ago
LICENSE	fix typo from LICENSE	4 months ago
README-CN.md	update readme	last month
README.md	update readme	last month
build.sh	update build.sh and dockerfiles	7 months ago

# Thanks