AiEDA: Al-Aided Design Library

李兴权 (Xingquan Li)







































Past OS @ 西门子 Events







Nov. 2024



Mar. 2025

iEDAOpen souce EDA

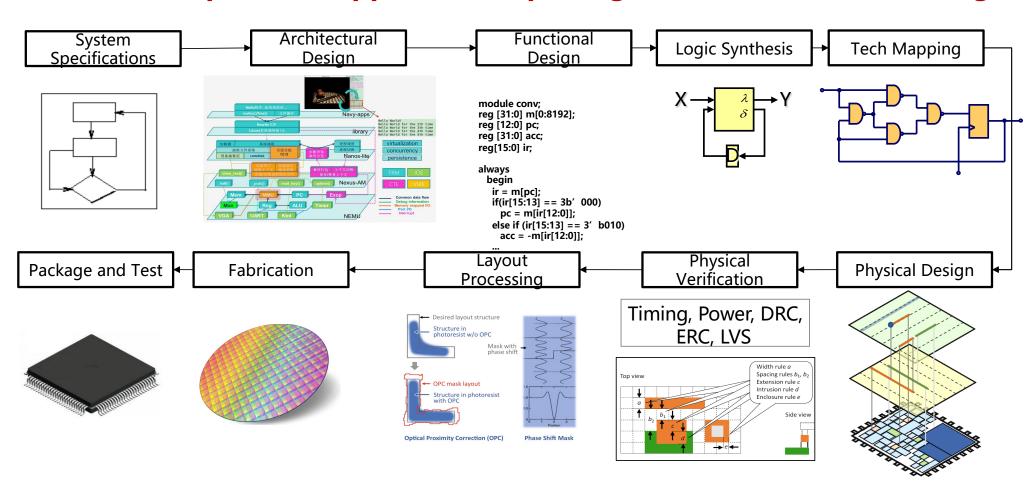
AiEDAAl EDA Library

• AI: iDATA, iPCL

Cloud: ECOS Studio

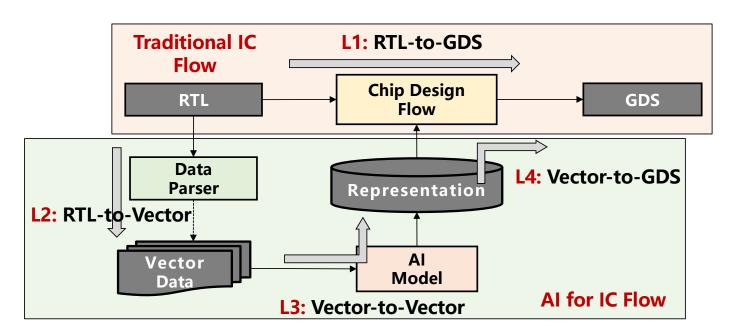
RTL-to-GDS Design Flow

• EDA tools provide support for chip design, verification, and testing.



Al-Aided Design (AAD)

- Al for IC: Exploring new methodologies for IC design to enhance chip design quality and efficiency.
 - Traditional IC design flow (RTL-to-GDS)
 - Represent chip data using AI models (RTL-to-Vector), then use the representative information to guide chip design (Vector-to-GDS).





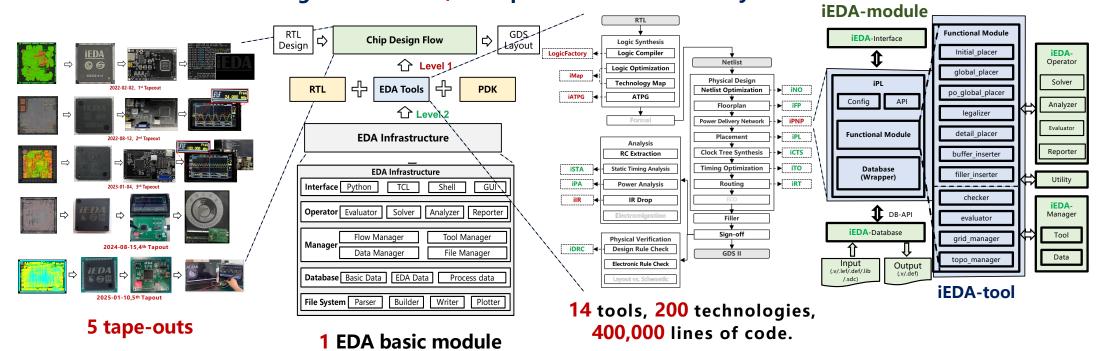
RTL-to-GDS (iEDA)

- 02
- **RTL-to-Vector (AiEDA)**
- 03
- **Vector-to-Vector (AiEDA-Models)**
- 04

Vector-to-GDS (iPCL)

iEDA: RTL-to-GDS

- 1 EDA platform, 5 basic modules, 14 EDA tools, over 200 algorithm technologies, 400,000 lines of code, completed 4 tape-outs, 1 problem white paper, 6 technical manuals.
- 650+ stars, 130+ forks, 100 code contributions, supported 4 EDA competitions, supported the curriculum construction of 5 universities, and assisted over 10 EDA teams with research, with iEDA videos achieving 66K views. 2,000 open-source community

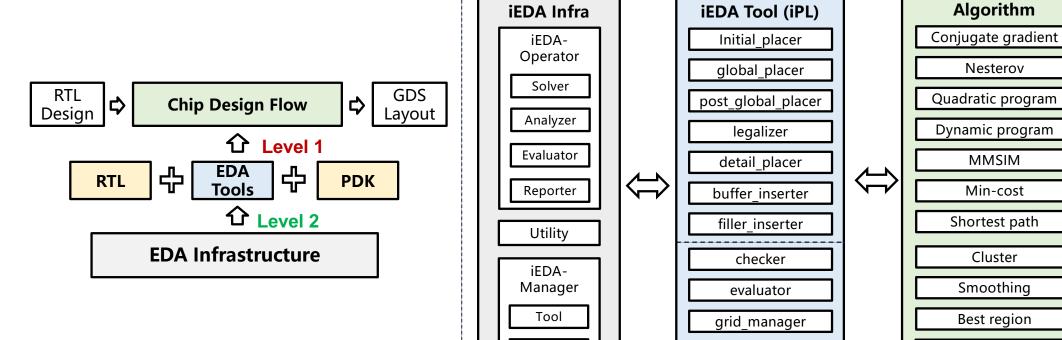


https://github.com/OSCC-Project/iEDA

iEDA Structure

- Level 1: Open-source EDA tools, RTL, PDK support chip design
- Level 2: Open-source EDA infrastructure supports EDA and algorithm design

Swapping



Data

topo manager

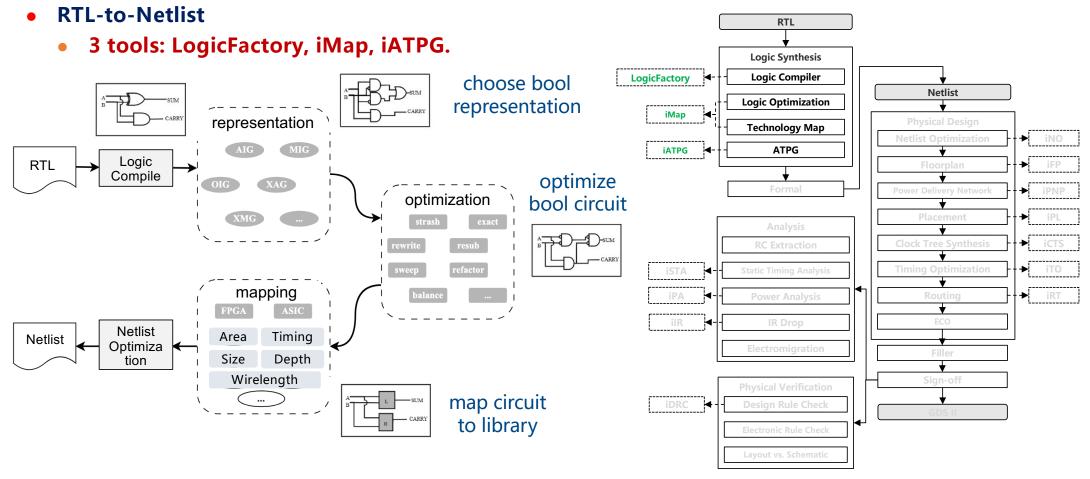
iEDA: Infrastructure

- 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

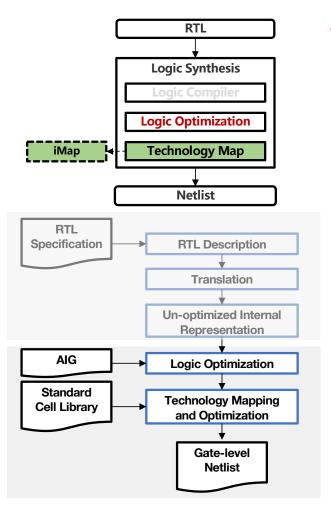
Infrastructure: Database, Manager, Operator, Interface **Functional Module iEDA**-Interface Initial placer iEDA-RTL **GDS** Operator **Chip Design Flow** \Rightarrow global placer Design Lavout Solver iPL post global placer 叴 **PDK EDA Tools** Config API Analyzer legalizer **EDA Infrastructure** Evaluator detail placer **Functional Module Interface** Python TCL Shell GUI Reporter buffer inserter **Operator** Evaluator Solver **Database** filler inserter Utility (Wrapper) Flow Manager Tool Manager checker Manager ! iEDA-Data Manager File Manager 1 DB-API evaluator Manager **Database** Basic Data **iEDA**-Database EDA Data Process data Tool grid manager File System Parser Builder Writer Plotter Data topo manager Input Output (.v/.lef/.def/.lib/.sd (.v/.def)

iEDA-Module 1: Logic Synthesis

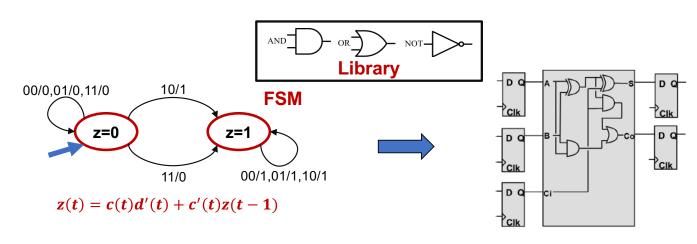
• Logic Synthesis: converts a RTL into a circuit description, optimize performance power area (PPA)



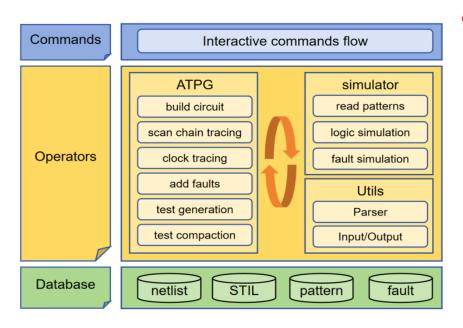
iEDA-Tool: Technology Map (iMAP)

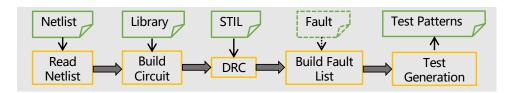


- iMAP 1.0 includes Logic Optimization and Tech Mapping
 - Data format support: AIG->netlist
 - Logic Optimization operators
 - Rewrite (NPN matching based on 4-input truth table)
 - Refactor (optimization based on SOP expression)
 - ✓ Balance (balance algorithm based on AND-tree)
 - ✓ LUT-opt (optimization based on FPGA tech mapping)
 - Technology Mapping for ASIC and FPGA



iEDA-Tool: Auto 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	n #Gates iATPG		Commercial tool			Fault Cover. Ratio		
Design	#Gates	fault cover.	time(sec.)	# of patterns	fault cover.	time(sec.)	# of patterns	(iATPG/Com. tool)
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%

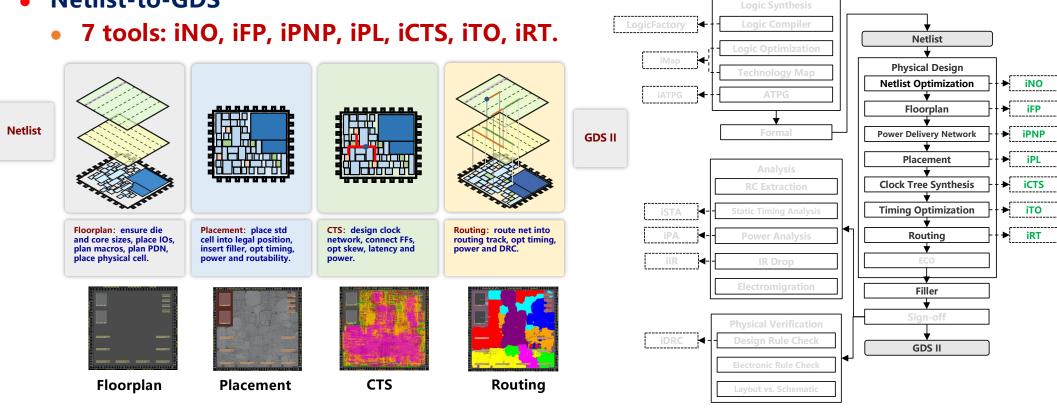
Fault Cover Ratio: 98%

Module 2: Physical Design

Physical Design: converts a circuit description into a geometric description, optimize

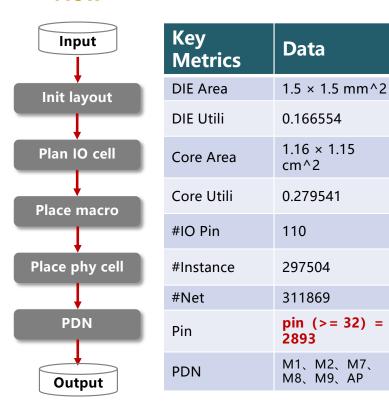
PPA

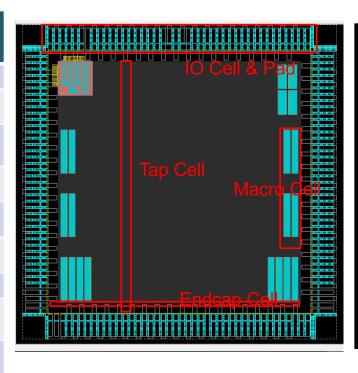
Netlist-to-GDS

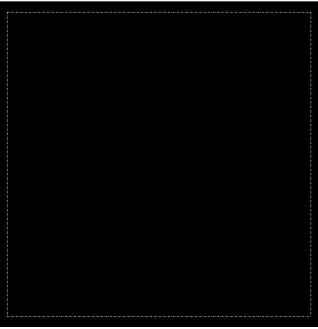


iEDA-Tool: iFP (Floorplan)

Flow

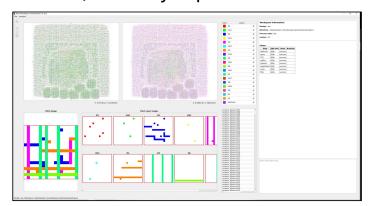


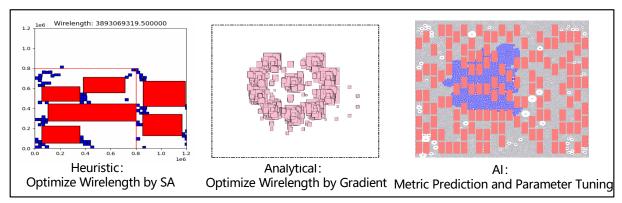


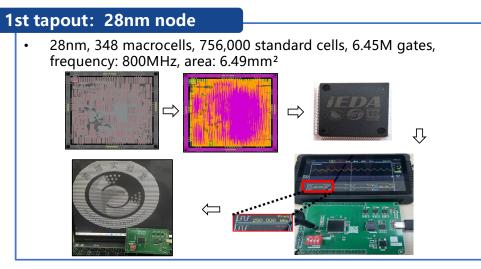


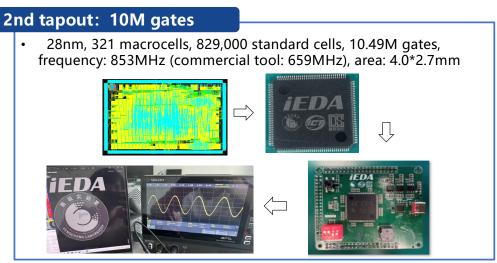
iEDA-Tool: Macro Placement (iMP)

- Al Macro Placement (iMP): Combine traditional methods, prior knowledge, and Al, reduce iterations.
- > Al: 1) Quantify experience and characterize design; 2) Fast evaluate layout quality; 3) Explore key parameters

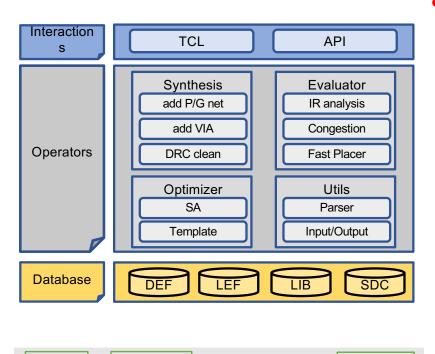








iEDA-Tool: Power Network Plan (iPNP)



Optimization

DEF/LEF

Read

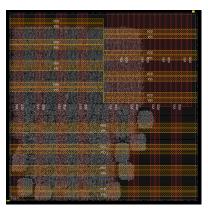
DEF/LEF

Template

Synthesize

p/g net

- 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



Result DEF_

Output

Fig1. Power Stripes

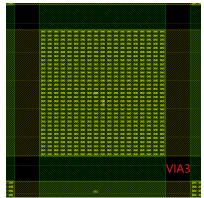


Fig2. Power VIA

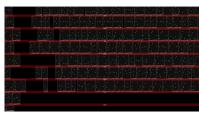


Fig3. Power Row

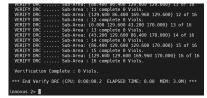
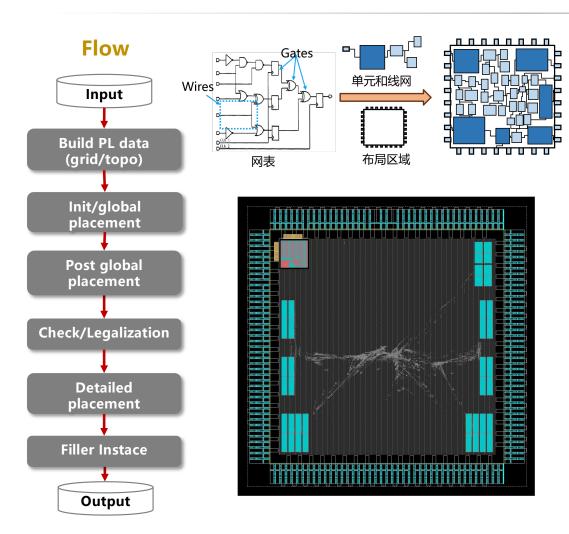


Fig4. DRC Clean

iEDA-Tool: **iPL** (Placement)



Min Wirelength Model

$$\min_{\boldsymbol{v}} W(\boldsymbol{v})$$

$$s. t. \ \rho_b(\boldsymbol{v}) \leq \rho_0, \quad \forall b \in B$$
where \boldsymbol{v} is cell location, $W(\boldsymbol{v})$ is wirelength,
$$\rho_b(\boldsymbol{v}) \text{ is the area density in } b \in B, \ \rho_0 \text{ is density thredhold.}$$

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

$$D(\boldsymbol{v}) = \frac{1}{2} \sum_{\boldsymbol{v} \in V} D_i(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{2} \sum_{\boldsymbol{v} \in V} q_i \psi_i(\boldsymbol{x}, \boldsymbol{y})$$

$$\begin{cases} \nabla \cdot \nabla \psi(\boldsymbol{x}, \boldsymbol{y}) = -\rho(\boldsymbol{x}, \boldsymbol{y}), \\ \widehat{\boldsymbol{n}} \cdot \psi(\boldsymbol{x}, \boldsymbol{y}) = \mathbf{0}, \quad (\boldsymbol{x}, \boldsymbol{y}) \in \partial R \end{cases}$$

$$\begin{cases} \int_R \rho(\boldsymbol{x}, \boldsymbol{y}) = \int_R \psi(\boldsymbol{x}, \boldsymbol{y}) = 0. \end{cases}$$

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

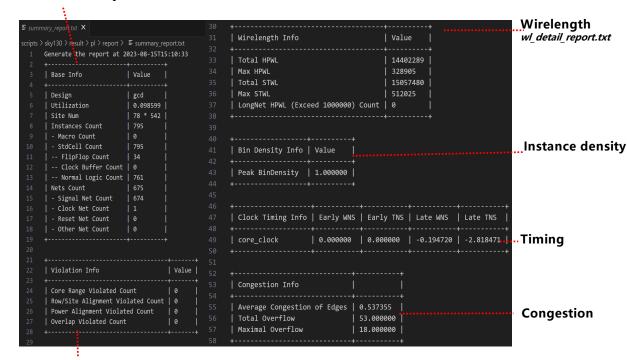
Nesterov Method or Conjugate Gradient

iEDA-Tool: **iPL** (Placement)



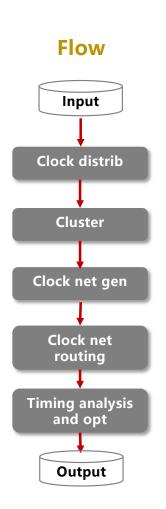
Key parameter config				
Input	iFP.def, iFP.v			
output	iPL_result.def, iPL.v			
is_max_length_opt	Whether to enable max wirelength optimization			
max_length_constraint	set max wirelength constraint			
is_timing_aware_mode	Whether to enable timing opt			
ignore_net_degree	ignore net whose pin number > k			
num_threads	set number of CPU thread			
[BUFFER] max_buffer_num	Set the number of using max buffer			
[BUFFER] buffer_type	Set available buffer name			
[GP-Wirelength] min_wirelength_force_ba r	Control wirelength range			
[GP-Density] target_density	Set target density			
[GP-Density] bin_cnt_x	Set the number of horizontal Bin			
[GP-Density] bin_cnt_y	Set the number of vertical Bin			
[LG] global_right_padding	Set instance spacing (/site)			
[DP] global_right_padding	Set instance spacing (/site)			
[Filler] min_filler_width	Set min width of filler (/site)			

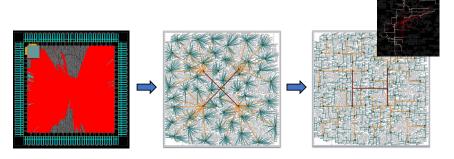
Basic Summary

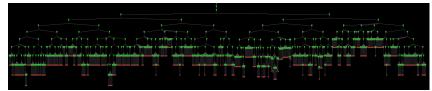


Design rule violation violation_detail_report.txt

iEDA-Tool: iCTS (Clock Tree Synthesis)







Timing

- Latency (max delay)
- Skew

Power

- Buffering
- Wirelength

l Level	+ Inst Num	+ Min Skew	+ Max Skew	+ Avg Skew	-++ Violation
+			+	+	-+
1	1210	6.62388e-05	0.0145301	0.00105275	0
2	204	0.000406356	0.0278176	0.00972537	0
3	80	0.0042316	0.0411625	0.0190886	0
4	41	0.00690457	0.0566811	0.0323446	0
5	22	0.0261657	0.0760005	0.0524973	0
6		0.0261657	0.0799602	0.0623044	0
7		0.0261657	0.08	0.0669329	0
8	4	0.0603809	0.08	0.0743834	0
9	2	0.0799208	0.08	0.0799604	0
10	1	0.08	0.08	0.08	0
+	+				

+	+ +		+	+	+	
	Level	Inst Num	Min Delay	Max Delay	Avg Delay	Violation
4	++		+		+	
	1	1210	0.000237424	0.0432362	0.00143337	None
	2	204	0.0518461	0.0983796	0.0688073	None
	3	80	0.119261	0.174637	0.145464	None
	4		0.187633	0.255905	0.219663	None
	5		0.250207	0.319777	0.294133	None
	6		0.324988	0.400232	0.365337	None
	7		0.405067	0.452041	0.435055	None
	8		0.500763	0.549897	0.522962	None
	9		0.573994	0.578742	0.576368	None
	10		0.608399	0.608399	0.608399	None
-	++		+	++-		

Type	Wire Length
Top	161.021
Trunk	2255.200
Leaf	9267.600
Total	l 11683.821

Max net length | 232.360

+		+-		+
Ĺ	Туре		HP Wire Length	١
+		+-		٠
Т	Тор		161.021	١
Т	Trunk		1347.840	١
Т	Leaf		3871.380	١
Т	Total		5380.241	١
Т	Max net length		161.021	١

Name +	Туре +	Inst Count	Inst Area (um^2)
CKBD12BWP35P140	Buffer Buffer	45	96.39 22.176
CKBD16BWP35P140	Buffer	8 12	40.824
CKBD24BWP35P140 CKBD4BWP35P140	Buffer Buffer	63 1082	254.016 954.324
CKBD6BWP35P140	Buffer	1129	1280.29
CKBD8BWP35P140	Buffer	81	122.472

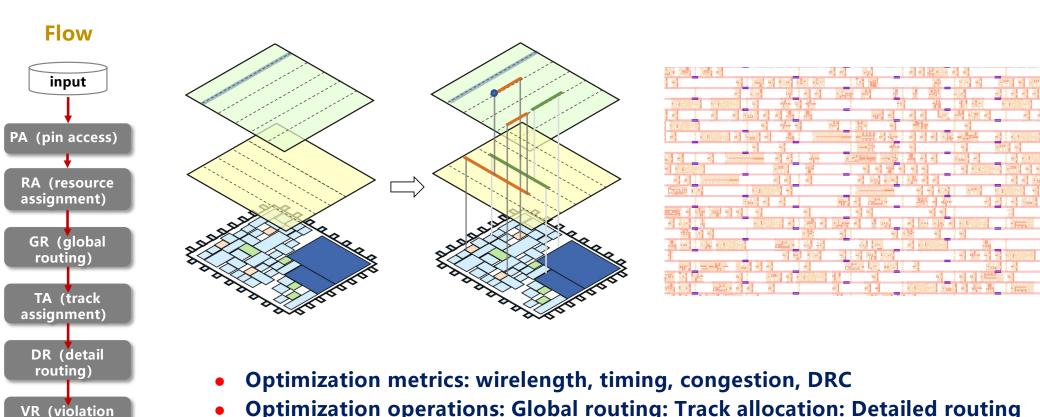
Violation

- Fanout
- Capacitance
- Slew (transition)

++			÷	+	 +
Level	Inst Num	Min Slew	Max Slew	Avg Slew	Violation
++				+	+
1	1210	0.0355823	0.0881141	0.0564704	970
2	204	0.0342527	0.0938317	0.0559007	134
3	80	0.0148774	0.103097	0.0581444	45
4	41	0.00736284	0.102664	0.0516092	20
5	22	0.0175251	0.103516	0.0535373	9
6	13	0.0116574	0.0884235	0.0402797	4
7	7	0.00706415	0.101609	0.0506611	3
8	4	0.015983	0.0220045	0.0188449	0
9	2	0.0259563	0.0286678	0.027312	0
10	1	0	2.22507e-308	0	0
+				+	+

Net / InstPin	MaxTranTime	TranTime	TranSlack	CellPort	Remark
sdram_clk_o	1				
sdram_clk_o_15397_buf:I	0.800r/0.800f	1.583r/1.494f	-0.783r/-0.694f	CKBD248WP35P140/I	R
sdram_clk_o	1				
	0.800r/0.800f	1.537r/1.446f		INVD18WP48P148LVT/ZN	R
osc 25m out pad					
υθ_clk:XOUT	5.000r/5.000f	5.784r/9.550f	-0.704r/-4.550f	PDX0EDG_V_G/X0UT	R
osc_100m_out_pad					
u1_clk:X0UT	5.000r/5.000f	5.784r/9.558f	-0.784r/-4.558f	PDX0EDG_V_G/X0UT	R
clk_hs_peri					
clk_hs_peri_11489_buf:I	0.800r/0.800f	0.850r/0.880f	-0.050r/-0.080f	CKBD248WP35P140/I	R
u0_rcg/u0_pll_clk	1				
u8 rcg/u1 lvt_cknux2hdv4:I1	0.267r/0.267f	0.189r/0.188f	0.078r/0.078f	CKMUX2D4BWP4GP14GLVT/I1	
u0_rcg/u0_pll_clk	1				
u8_rcg/u8_pll:FOUTPOSTDIV	8.267r/8.267f	0.000r/0.000f	0.267r/0.267f	PLLTS28HPMLAINT/FOUTPOSTDIV	
clk_core					
clk_core_1724_buf:I	0.800r/0.800f	0.522r/0.533f	0.278r/0.267f	CKBD48WP35P140/I	
clk_hs_peri					
u0_soc_top/u0_sdram_axi/u_core/U300:I	0.800r/0.800f	0.519r/0.568f	0.281r/0.232f	INVD18WP48P149LVT/I	
u8_soc_top/n495					
fanout buf 40:I	1 8.437c/8.437f	1 0.898c/0.056f	p.339r/p.381f	BUFFDRRWP38P148LVT/T	

iEDA-Tool: iRT (Routing)

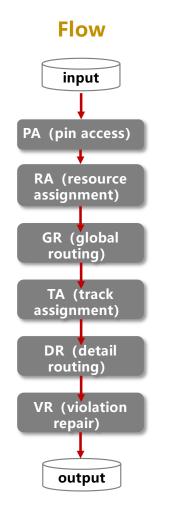


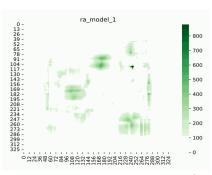
repair)

output

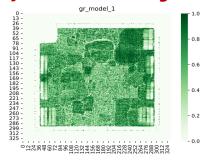
- **Optimization operations: Global routing: Track allocation: Detailed routing**
- Routing algorithms: Pattern routing, A* routing, Steiner tree, Non-linear programming, Integer programming

iEDA-Tool: iRT (Routing)



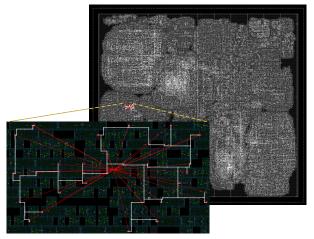


Layout resource/congestion

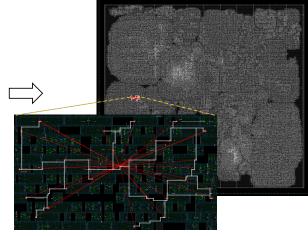


Access Type	Pin Number	Routing Layer	Port Number	Access Point Number
Track Grid	876856(75.2479%)	M1	806086(68.3279%)	799044(68.5705%)
On Track	259825(22.297%)	M2	366214(31.0422%)	362248(31.0865%)
On Shape	28608(2.45501%)	M3	7073(0.599543%)	3997(0.343005%)
	i — — — — — — — — — — — — — — — — — — —	M4	358(0.0303459%)	0(0%)
Total	1165289	M5	0(0%)	0(0%)
	ii	M6	0(0%)	0(0%)
		M7	0(0%)	0(0%)
		M8	0(0%)	0(0%)
		M9	0(0%)	0(0%)
		AP	0(0%)	0(0%)
		Total	1179731	1165289





first iteration routing result



final routing result

Routing Layer	Wire Length / um	Cut Layer	Via Number	Resource Overflow	GCell Number	Access Overflow	GCell Number
M1	9774(0.117785%)	со	0(0%)	[0,0.1)	921387(83.1%)	[0,0.1)	1.78338e+06(80.4%)
M2	846292(10.1985%)	VIA1	595417(30.6475%)	[0.1,0.2)	57544(5.19%)	[0.1,0.2)	108739(4.9%)
	1.98405e+06(23.9095%)	VIA2	682833(35.147%)	[0.2,0.3)	51492(4.64%)	[0.2,0.3)	79939(3.6%)
M4	1.78748e+06(21.5406%)	VIA3	400386(20.6088%)	[0.3,0.4)	40084(3.61%)	[0.3,0.4)	90020(4.06%)
M5	1.29642e+06(15.6229%)	VIA4	135600(6.97965%)	[0.4,0.5)	21944(1.98%)	[0.4,0.5)	56112(2.53%)
M6	1.41202e+06(17.016%)	VIA5	89437(4.60353%)	[0.5,0.6)	10780(0.972%)	[0.5,0.6)	36741(1.66%)
M7	960890(11.5795%)	VIA6	38709(1.99244%)	[0.6,0.7)	4140(0.373%)	[0.6,0.7)	30046(1.35%)
M8	539.92(0.00650648%)	VIA7	262(0.0134857%)	[0.7,0.8)	1223(0.11%)	[0.7,0.8)	12155(0.548%)
M9	720(0.00867659%)	VIA8	148(0.0076179%)	[0.8,0.9)	222(0.02%)	[0.8,0.9)	8771(0.395%)
AP	0(0%)	RV	0(0%)	[0.9,1]	74(0.00667%)	[0.9,1]	11881(0.536%)
Total	8.29819e+06	Total	1942792	Total	1108890	Total	2217780

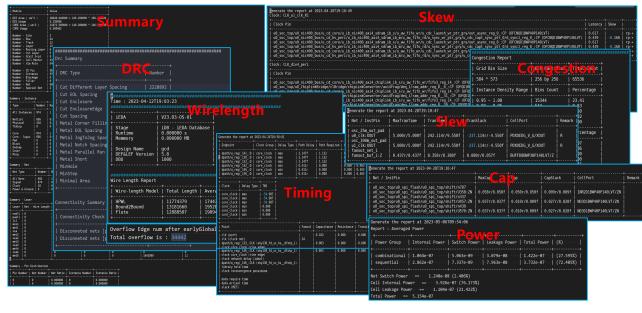
Wirelength and via		
--------------------	--	--

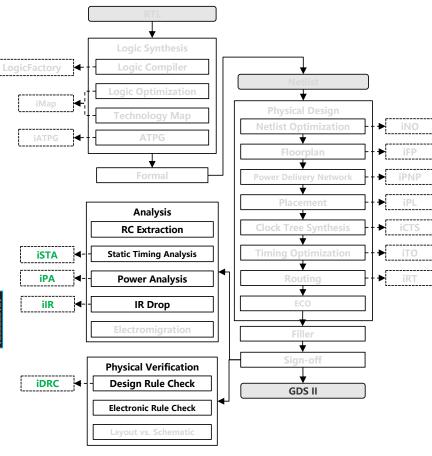
orc Summary	
DRC Type	Number
Cut Different Layer Spacing Cut EOL Spacing Cut Enclosure Cut Enclosuredge Cut Spacing Metal Corner Filling Spacing Metal DOL Spacing Metal DOL Spacing Metal JogToJog Spacing Metal JogToJog Spacing Metal Notch Spacing	433141 197803 152168 0 358281 10443 869415 0
Metal Parallel Run Length Spacing Metal Short MinHole MinStep	864355 1745445 1260 670823
Minimal Area	1248072

Design rule check

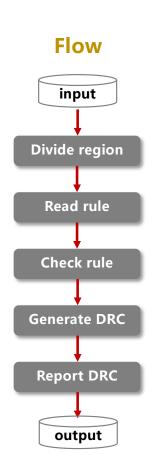
Module 3: Sign-off Analysis

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





iEDA-Tool: iDRC (Design Rule Check)



- 覆盖28nm工艺的主要规则
- 和商业工具Calibre对比

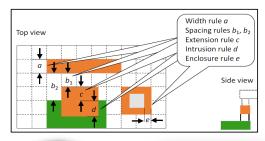
在检测准确度达到: 92.8%

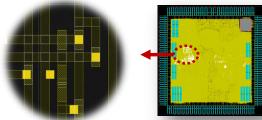
• 主要规则检测准确度达到: 97%

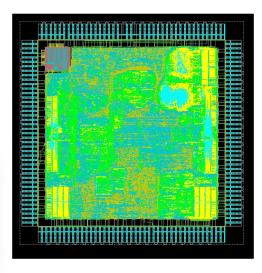
● 支持的DRC检测类型

- · Cut Different Layer Spacing
- Cut EOL Spacing
- Cut Enclosure
- Cut EnclosureEdge
- 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

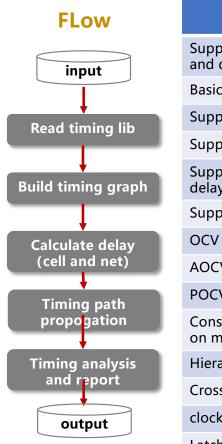
+	+	·	+	+	+	+
violation_type	correct_num	prop	incorrect_num	prop	missed_num	prop
corner_fill_spacing	39	100.00%	+ 0	+ 0.00%	+ 0	0.00%
cut_eol_spacing	12713	50.04%	5798	22.82%	6895	27.14%
cut_short	163	100.00%	0	0.00%	0	0.00%
different_layer_cut_spacing	6271	99.54%	29	0.46%	0	0.00%
enclosure_edge	825	61.94%	30	2.25%	477	35.81%
enclosure_parallel	0	0.00%	4399	100.00%	0	0.00%
end_of_line_spacing	15995	65.72%	4107	16.88%	4235	17.40%
max_via_stack	0	0.00%	89	50.00%	89	50.00%
metal_short	9422	99.38%	49	0.52%	10	0.11%
min_hole	1048	100.00%	0	0.00%	0	0.00%
min_step	55510	98.63%	384	0.68%	389	0.69%
minimum_area	258642	98.36%	2322	0.88%	1988	0.76%
nonsufficient_metal_overlap	275	100.00%	0	0.00%	0	0.00%
notch_spacing	15230	95.49%	360	2.26%	359	2.25%
parallel_run_length_spacing	23067	97.34%	339	1.43%	291	1.23%
same_layer_cut_spacing	31265	99.99%] 3	0.01%	0	0.00%
+		·	+	+	++	+
Total	424466	92.86%	17909	3.92%	14734	3.22%
+		+	+	+	+	+







iEDA-Tool: iSTA (Static Timing Analysis)



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

AOCV

POCV

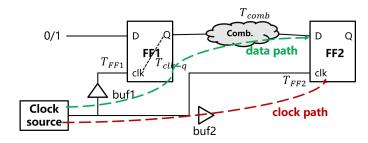
Consider IRDrop analysis on multi-voltage domain

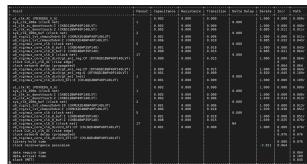
Hierarchy analysis

Crosstalk analysis

clock gate analysis

Latch analysis



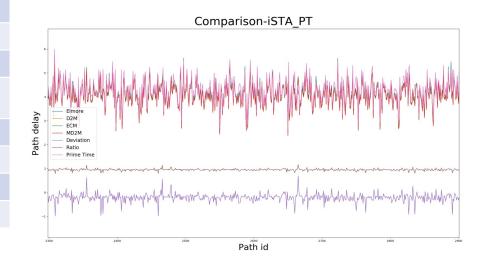


$$T_{FF1} + T_{clk-q} + T_{comb} + T_{setup} - T_{FF2} - T = T_{slack}^{late} \ge 0$$

$$T_{FF1} + T_{clk-q} + T_{comb} - T_{hold} - T_{FF2} = T_{slack}^{early} \ge 0$$

Setup Constraint

Hold Constraint



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

iEDA-Tool: iTO (Timing Optimization)

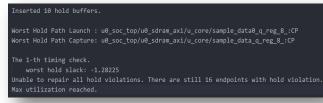


K	ey parameter config
Input	iPL.def, iCTS.def
output	iTO_setup_result.def, iTO_hold_reslut.def
setup_slack _margin	setup slack value
hold_slack_ margin	hold slack value
max_buffer _percent	Area ratio of inserted buffer
max_utiliza tion	Core utilization
DRV_insert _buffers	Available buffer for optimizing DRV
setup_inser t_buffers	Available buffer for optimizing setup
hold_insert _buffers	Available buffer for optimizing hold
number_pa sses_allowe d_decreasin g_slack	The number of times that WNS is allowed continuously decrease when opt setup
rebuffer_m ax_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



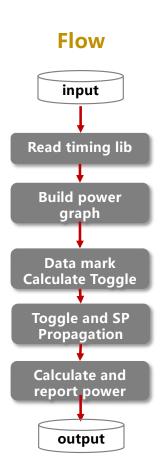
Clock Group	Hold TNS	Hold WNS
CLK_chiplink_tx_clk		e
CLK_clk_hs_peri		
CLK_div2_core	-2606.7	-0.106129
CLK_div2_hs_peri		-0.028571
CLK_div3_hs_peri	-72.5124	-0.028571
CLK_div4_core	-1304.1	-0.106129
CLK_div4_hs_peri		
CLK_div4_peri	-231.408	-0.042802
CLK_sdram_clk_o		e
CLK_spi_clk		e
CLK_spi_clk_out		e
CLK_u0_chiplink_rx_clk_pad_PAD	-89.8732	-0.028408
CLK_u0_clk_XC	-2987.42	-0.106129
CLK_u0_pll_FOUTPOSTDIV	-8546.64	-0.106129
CLK_u1_clk_XC	-2755.16	-0.106129

- 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

Clock Group	Hold WNS
CLK_chiplink_tx_clk	0
CLK_clk_hs_peri	0
CLK_div2_core	0
CLK_div2_hs_peri	0
CLK_div3_hs_peri	
CLK_div4_core	0
CLK_div4_hs_peri	0
CLK_div4_peri	0
CLK_sdram_clk_o	0
CLK_spi_clk	0
CLK_spi_clk_out	0
CLK_u0_chiplink_rx_clk_pad_PAD	
CLK_u0_clk_XC	0
CLK_u0_pl1_FOUTPOSTDIV	
CLK_u1_clk_XC	0

Hold report

iEDA-Tool: iPA (Power Analysis)

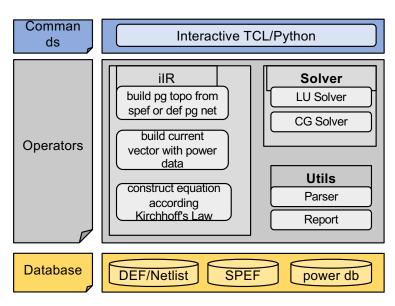


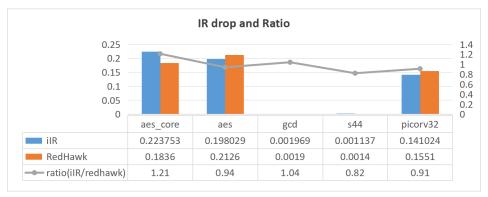
API	Description
buildGraph	Build iPW graph data structure
readVCD	Parse the VCD file
buildSeqGraph	Build timing subgraph
checkPipelineLoop	Detect PipeLine loop
levelizeSeqGraph	Grade timing subgraph
propagateToggleSP	Propagate Toggle and SP data on the graph
calcLeakagePower	Calculate the leakage power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

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

iEDA-Tool: IR Drop Analysis (iIR)





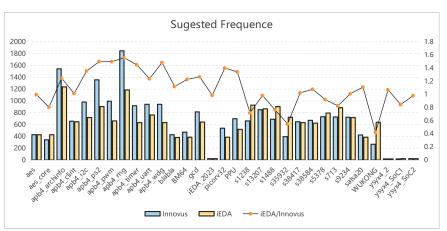
iIR: analyze the IR drop of the chip.

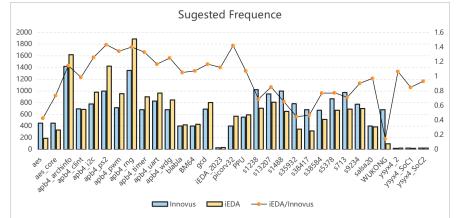
Feature

- 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.

PPA Comparison of iEDA (OS) /Innovus (Com)

- 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

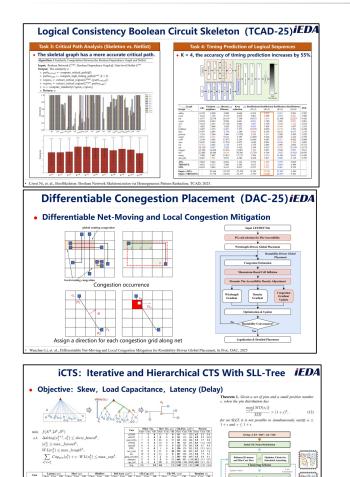
95.40 %

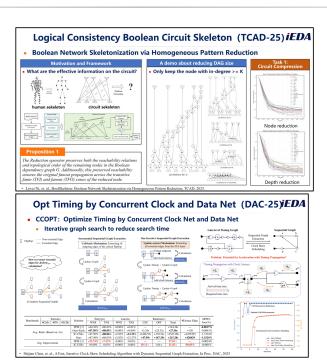
#Cells #Nets #Wires s713 135 125 1426 s44 178 128 2095 apb4_rng 195 204 2230 gcd 297 270 3733 s1238 349 290 4998 s1488 380 325 6422 apb4_ps2 515 497 6542 apb4_pp2 515 497 6542 apb4_ps2 515 497 6542 apb4_ps2 515 497 6542 apb4_ps2 721 689 8960 s13207 727 647 8182 apb4_pb4_timer 721 689 8960 s13207 727 647 8182 apb4_pb4 1029 945 13889 apb4_pb 1029 945 13889 apb4_pb 1029 945 13889 apb4_duart 5981 5606 83268	Circuits		Design	
s44 178 128 2095 apb4_rng 195 204 2230 gcd 297 270 3733 s1238 349 290 4998 s1488 380 325 6422 apb4_pb2 515 497 6542 apb4_pp 515 497 6542 apb4_timer 721 689 8960 s13207 727 647 8182 apb4_jcc 790 727 10248 s5378 881 774 10242 apb4_ppm 974 889 13596 apb4_wdg 1029 945 13889 apb4_wdg 1029 945 13889 apb4_uart 5981 5606 83268 s38417 6028 5573 85054 apb4_uart 5981 5606 83268 s38417 6028 5573 85054 s38417 35895 4506	Circuits	#Cells	#Nets	#Wires
s44 178 128 2095 apb4_rng 195 204 2230 gcd 297 270 3733 s1238 349 290 4998 s1488 380 325 6422 apb4_pb2 515 497 6542 apb4_pp 515 497 6542 apb4_timer 721 689 8960 s13207 727 647 8182 apb4_jcc 790 727 10248 s5378 881 774 10242 apb4_ppm 974 889 13596 apb4_wdg 1029 945 13889 apb4_wdg 1029 945 13889 apb4_uart 5981 5606 83268 s38417 6028 5573 85054 apb4_uart 5981 5606 83268 s38417 6028 5573 85054 s38417 35895 4506	s713	135	125	1426
gcd 297 270 373 s1238 349 290 4998 s1488 380 325 6422 apb4_arch 392 381 5122 apb4_ps2 515 497 6542 sp2334 657 585 8592 apb4_timer 721 689 8960 s13207 727 647 8182 apb4_pb4 790 727 10248 s5378 881 774 1242 apb4_pwm 974 889 13596 apb4_pwm 974 889 13596 apb4_clint 1069 1004 13634 ASIC 1228 796 10737 s15850 2088 1926 27941 apb4_uart 5981 5606 83268 s38417 6028 5573 85054 s35893 6375 5837 81158 s38584 7023 6586	s44	178	128	2095
gcd 297 270 373 s1238 349 290 4998 s1488 380 325 6422 apb4_arch 392 381 5122 apb4_ps2 515 497 6542 sp2334 657 585 8592 apb4_timer 721 689 8960 s13207 727 647 8182 apb4_pb4 790 727 10248 s5378 881 774 1242 apb4_pwm 974 889 13596 apb4_pwm 974 889 13596 apb4_clint 1069 1004 13634 ASIC 1228 796 10737 s15850 2088 1926 27941 apb4_uart 5981 5606 83268 s38417 6028 5573 85054 s35893 6375 5837 81158 s38584 7023 6586	apb4 rng	195	204	2230
\$1238 349 290 4998 \$1488 380 325 6422 apb4_arch 392 381 5122 apb4_bres 515 497 6542 sp234 657 585 8592 apb4_timer 721 689 8960 \$13207 727 647 8182 apb4_jcc 790 727 10248 \$5378 881 774 12422 apb4_pwm 974 889 13596 apb4_dwdg 1029 945 13889 apb4_dwdg 1029 945 13889 apb4_dwdg 1029 945 13889 apb4_uart 5981 5606 83268 \$38417 6028 5573 85054 \$38417 6028 5573 85054 \$38417 6028 5573 85054 \$38417 8359 1132076 686 97771 BM64 9358<				
s1488 380 325 6422 apb4_arch 392 381 5122 apb4_ps2 515 497 6542 s9234 657 585 8592 apb4_timer 721 689 8960 s13207 727 647 8182 apb4_i2c 790 727 10248 sp3378 881 774 12422 apb4_wdg 1029 945 13889 apb4_wdg 1029 945 13889 apb4_wdg 1029 945 13889 apb4_wdg 1029 945 13889 apb4_udt 1069 1004 13634 ASIC 1228 796 10737 s18850 2088 1926 27941 apb4_uart 5981 5606 83268 s38417 6028 5573 85054 s38584 7023 6586 97771 BM64 9358 9510 </td <td></td> <td></td> <td></td> <td></td>				
apb4_arch 392 381 5122 apb4_ps2 515 497 6542 sp234 657 585 8592 apb4_timer 721 689 8960 s13207 727 647 8182 s5378 881 774 12422 apb4_pwm 974 889 13596 apb4_pwm 974 889 13596 apb4_pwd 1029 945 13889 apb4_clint 1069 1004 13634 ASIC 1228 796 10737 s15850 2088 1926 27941 apb4_uart 5981 5606 83268 s38417 6028 5573 85054 s358932 6375 5837 81158 s35894 7023 6586 97711 BM64 9358 9510 132076 picorv32 9430 9077 136455 PPU 9547 88				
apb4_ps2 515 497 6542 s9234 657 585 8592 apb4_timer 721 689 8960 s13207 727 647 8182 apb4_i2c 790 727 10248 s5378 881 774 12422 apb4_pwm 974 889 13596 apb4_wdg 1029 945 13889 apb4_clint 1069 1004 13634 ASIC 1228 796 10737 s15850 2088 1926 27941 apb4_uart 5981 5606 83268 s38417 6028 5573 85054 s38417 6028 5573 85054 s38941 7023 6586 97771 BM64 9358 9510 132076 picorv32 9430 9077 36455 PPU 9547 8895 140136 blabla 15154 15672 216427 aes_core 17940 17371 310215 aes 19181 18117 325550 salsa20 21270 20432 291172 aes_core 17940 17371 310215 aes 19181 38117 325550 salsa20 21270 20432 291172 aes_core 27674 29160 366397 eth_top 42279 38552 646875 yadan 63514 31280 4833689 beihai 211236 133086 2161829 beihai 211236 133086 2161827 JUC 348985 32195 4880427 JUC 348985 32195 4880427 JUC 348985 32195 4880427 JUC 348985 32195 4880427 JUB 348985 32195 4880427 JUG 348985 32195				
s9234 657 585 8592 apb4_timer 721 689 8960 s13207 727 647 8182 apb4_i2c 790 727 10248 s5378 881 774 12422 apb4_wdg 1029 945 13899 apb4_wdg 1029 945 13899 apb4_udg 1029 945 13899 apb4_udg 1029 945 13899 apb4_udg 1029 1004 13634 ASIC 1228 796 10737 s18850 2088 1926 27941 apb4_uart 5981 5606 83268 s38417 6028 5573 85054 s35932 6375 5837 81158 s3644 7023 6586 97771 BM64 9358 9510 132076 per 17947 18012 140136 blabla 15154 1				
apb4_timer 721 689 8960 s13207 727 647 8182 apb4_i2c 790 727 10248 s5378 881 774 12422 apb4_pwm 974 889 13598 apb4_clint 1069 1004 13634 ASIC 1228 796 10737 s15850 2088 1926 27941 apb4_uart 5981 5606 83268 s38417 6028 5573 85054 s335932 6375 5837 81158 s38584 7023 6586 97771 BM64 9358 9510 132076 picorv32 9430 9077 136455 PPU 9547 8895 140136 blabla 15154 15672 216427 aes_core 17940 17371 310215 aes_ 19181 18117 325550 aes_ 19181 18117 325550 salsa20 21270 20432 291172 jpeq 27671 29160 366397 eth_top 42279 38552 646875 yadan 63514 31280 483369 beihai 211236 133086 2161829 SHMS 26872 1251772 3610024 nvd1a 289344 226974 3708427 syxy452 494962 449847 6967779 beihai2 582645 393308 5491501 AIMP 742210 535618 9980714 AIMP2 816677 560525 9133333 sysyx6 1090820 1029515 15486068 wukong 1102663 1032718 15653639 ysyx41 1123368 1105564 18248292 ysyx42 T1 12601 1147953 17416249 T1 1262053 122709 18769036 T1_sand 4816399 4728816 79050737				
s13207 727 647 8182 apb4_i2c 790 727 10248 s5378 881 774 10248 s5378 881 774 12422 apb4_pwm 974 889 13596 apb4_wdg 1029 945 13889 apb4_clint 1069 1004 13634 ASIC 1228 796 10737 s15850 2088 1926 27941 apb4_uart 5981 5606 83268 s38417 6028 5573 85054 s38584 7023 6586 97771 BM64 9358 9510 132076 picorv32 9430 9077 36455 PPU 9547 8895 140136 blabla 15154 15672 216427 aes_core 17940 17371 310215 aes 21270 20432 291172 geth_top 42279				
apb4_i2c 790 727 10248 s5378 881 774 12422 apb4_pwm 974 889 13596 apb4_clint 1069 1004 13634 ASIC 1228 796 10737 s15850 2088 1926 27941 apb4_uart 5981 5606 83268 s38417 6028 5573 85054 s35932 6375 5837 81158 s38584 7023 6586 97771 BM64 9358 9510 132076 picorv32 9430 9077 136455 PPU 9547 8895 140136 blabla 15154 15672 216427 aes_core 17940 17371 310215 aes 19181 18117 325550 salsa20 21270 20432 291172 jpeg 27671 29160 366397 eth_top 42279 38552 646875 yadan 63514 31280 483369 beihai 211236 13368 2161829 symy452 494962 449847 6967779 beihai2 368147 335112 5132004 ysyx452 494962 449847 6967779 beihai2 368147 335112 5132004 ysyx452 494962 449847 6967779 beihai2 582645 393308 5491501 s1MP2 816677 560525 9133333 ysyx6 1090820 1029515 15486068 wukong 1102663 1032718 15653639 ysyx41 1123368 1105564 18248292 ysyx42 1173610 1147953 17416249 T1 123608 112790 3115508 penc910 3282828 2948743 52259408 T1_sand 4816399 4728816 79050737				
s5378 88I 774 12422 apb4_pwm 974 889 13596 apb4_pwd 1029 945 13889 apb4_clint 1069 1004 13634 ASIC 1228 796 10737 s15850 2088 1926 27941 apb4_uart 598I 5606 83268 s38417 6028 5573 85054 s35932 6375 5837 81158 s35894 7023 6586 9771 BM64 9358 9510 132076 picorv32 9430 9077 136455 PPU 9547 8895 140136 blabla 15154 15672 216427 aes_core 11940 17371 310215 aes 9181 18117 325550 salsa20 21270 20432 291172 jpeq 27671 29160 366397 eth_top 42279				
apb4_pwm 974 889 13596 apb4_wdg 1029 945 13889 apb4_clint 1069 1004 13634 ASIC 1228 796 10737 s15850 2088 1926 27941 apb4_uart 5981 5606 83268 s35932 6375 5837 81158 s35932 6375 5836 97771 BM64 9358 9510 132076 picorv32 9430 9077 136455 PPU 9547 8895 140136 blabla 15154 15672 216427 aes_core 17940 17371 310215 aes 19181 18117 32550 salsa20 21270 20432 291172 yadan 63514 31280 483369 beihai 211236 133086 2161829 StMS 268721 251772 3610024 nvdla				
apb4_wdg 1029 945 13889 apb4_clint 1069 1004 13634 ASIC 1228 796 10737 s15850 2088 1926 27941 apb4_uart 5981 55066 83268 338417 6028 5573 85054 835932 6375 5837 81158 838584 7023 6586 97771 186455 PU 9547 8895 140136 140136 15154 15672 216427 aes_core 17940 17371 310215 aes 19181 18117 325550 321820 20432 291172 31969 27671 29160 366397 eth_top 42279 38552 646875 34686 251482 251772 3610024 24184 24294 34369 24182 2418				
apb4_clint 1069 1004 13634 ASIC 1228 796 10737 s15850 2088 1926 27941 apb4_uart 5981 5606 83268 s38417 6028 5573 85054 s35932 6375 5837 81158 s35884 7023 6586 97711 BM64 9358 9510 132076 picorv32 9430 9077 136455 PPU 9547 8895 140136 blabla 15154 15672 216427 aes_core 17940 17371 310215 aes 19181 18117 325550 salsa20 21270 20432 291172 jpeq 27671 29160 366397 eth_top 42279 38552 646875 yadan 63514 31280 483369 beihai 211236 133086 2161829 SHMS				
ASIC 1228 796 10737 s15875 2088 1926 27941 apb4_uart 5981 5606 83268 s38417 6028 5573 85054 s35932 6375 5837 81158 s38584 7023 6586 97771 BM64 9358 9510 132076 picorv32 9430 9077 136455 PPU 9547 8895 140136 blabla 15154 15672 216427 aes_core 17940 17371 310215 aes 19181 18117 325550 salsa20 21270 20432 291172 jpeg 27671 29160 366397 eth_top 42279 38552 646875 yadan 63514 31280 483369 beihai 211236 13368 2161829 s1MS 268721 251772 3610024 nvdla 289344 226974 3708427 ZJUC 348985 323195 4880442 zJUC 348985 323195 4880442 zJUC 348985 323195 4880442 xJYX4S2 494962 449847 6967779 beihai2 368147 335112 5132004 ysyx4S2 494962 449847 6967779 beihai2 582645 393308 5491501 AIMP 742210 355618 9980714 AIMP2 816677 560525 9133333 ysyx6 1090820 1029515 15486308 ysyx41 1123368 1105564 18248292 ysyx42 1173368 1105564 18248292 ysyx41 1123368 1105564 18248292 ysyx42 173610 1147953 17416249 T1 1262033 1227098 18769036 T1_mach 2793215 26466672 45224007 openC910 3282828 2948743 52259408 T1_sand 4816399 4728816 79050737				
s15850 2088 1926 27941 apb4_uart 5981 5606 83268 s38417 6028 5573 85054 s35932 6375 5837 81158 s38584 7023 6586 97771 BM64 9358 9510 132076 picorv32 9430 9077 136455 PPU 9547 8895 140136 blabla 15154 15672 216427 aes_core 17940 17371 310215 aes 19181 18117 325550 salsa20 21270 20432 291172 jpeg 27671 29160 366397 eth_top 42279 38552 646875 yadan 63514 31280 2161829 SHMS 268721 251772 3610024 nvdla 289344 226974 3708427 zJUC 348985 323195 488044 iEDA23 </td <td></td> <td></td> <td></td> <td></td>				
apb4_uart 5981 5606 83268 s38417 6028 5573 85054 s35932 6375 5837 85054 s38584 7023 6586 97771 BM64 9358 9510 132076 picorv32 9430 9077 136455 PPU 9547 8895 140136 blabla 15154 15672 216427 aes_core 17940 17371 310215 aes 20181 18117 325550 salsa20 21270 20432 291172 jpeq 27671 29160 366397 eth_top 42279 38552 646875 yadan 63514 31280 483369 beihai 211236 133086 2161829 SHMS 268721 251772 3610024 nvdla 289344 226974 3708427 ZUC 348985 323195 4880422 iED				
s38417 6028 5573 81054 s35932 6375 5837 81154 s38584 7023 6586 97771 BM64 9358 9510 132076 picorv32 9430 9077 136455 PPU 9547 8895 140136 blabla 15154 15672 216427 aes_core 17940 17371 310215 aes 21270 20432 291172 jpeg 27671 29160 366397 eth_top 42279 38552 646875 yadan 63514 31280 483369 beihai 211236 133086 2161829 SHMS 268721 251772 3610024 nvdla 289344 226974 3708427 ZJUC 348985 323195 488042 ysyx452 494962 449847 6967779 beihai2 35814 335112 5132004 <				
s35932 6375 5837 81158 s38584 7023 6586 97771 BM64 9358 9510 132076 picorv32 9430 9077 136455 PPU 9547 8895 140136 blabla 15154 15672 216427 aes_core 17940 17371 310215 aes 19181 18117 325550 salsa20 21270 20432 291172 jpeg 27671 29160 366397 eth_top 42279 38552 646875 yadan 63514 31280 483369 beihai 211236 133086 2161829 SHMS 268721 251772 3610024 nvdla 289344 226974 3708427 ZJUC 348985 323195 488044 jehihai2 368147 335112 5132004 ysyx452 494962 449487 6967779				
s38584 7023 6586 97771 BM64 9358 93510 132076 ppicorv32 9430 9077 136455 PPU 9547 8895 140136 blabla 15154 15672 216427 aes_core 17940 17371 310215 aes 19181 18117 325550 salsa20 21270 20432 291172 jpeq 27671 29160 366397 eth_top 42279 38552 646875 yadan 63514 31280 483369 beihai 211236 133086 2161829 SHMS 268721 251772 3610024 nvdla 289344 226974 3708427 ZUC 348985 323195 488042 iEDDA23 368147 335112 5132004 ysyx452 494962 449847 6967779 beihai2 582645 393308 5491501 <				
BM64 9358 9510 132076 picorv32 9430 9077 136455 PPU 9547 8895 140136 blabla 15154 15672 216427 aes_core 17940 17371 310215 aes 19181 18117 325550 salsa20 21270 20432 291172 ydan 63514 31280 483369 beihai 211236 13386 2161829 SHMS 268721 251772 3610024 nvdla 289344 226974 3708427 ZJUC 348985 323195 4880442 ysyx452 494962 449847 6967779 beihai 582645 393308 5491501 AIMP 742210 35618 980714 AIMP 742210 35618 980714 AIMP 742210 35618 980714 AIMP 742210 35618 980714				
picorv32 9430 9077 136455 PPU 9547 8895 140136 blabla 15154 15672 216427 aes_core 17940 17371 310215 aes 19181 18117 325550 salsa20 21270 20432 291172 jpeg 27671 29160 366397 eth_top 42279 38552 646875 yadan 63514 31280 483369 beihai 211236 133086 2161829 SHMS 268721 251772 3610024 nvdla 289344 226974 3708427 ZJUC 348985 323195 4880442 iEDA23 368147 335112 5132004 ysyx452 494962 449487 6967779 beihai2 582645 393308 \$491501 AIMP 742210 535618 9980714 AIMP 742210 535618 9980714 <td></td> <td></td> <td></td> <td></td>				
PPU 9547 8895 140136 blabla 15154 15672 216427 aes_core 17940 17371 310215 aes 19181 18117 325550 salsa20 21270 20432 291172 jpeq 27671 29160 366397 eth_top 42279 38552 646875 yadan 63514 31280 483369 beihai 211236 133086 2161829 SHMS 268721 251772 3610024 nvdla 289344 226974 3708427 ZUC 348985 323195 488042 iEDDA23 368147 335112 5132004 ysyx452 494962 449847 6967779 beihai2 582645 393308 591501 AIMP 742210 535618 9980714 AIMP 742210 535618 9980714 AIMP 742210 535618 9980714 <td></td> <td></td> <td></td> <td></td>				
blabla 15154 15672 216427 aes_core 17940 17371 310215 aes 19181 18117 325550 salsa20 21270 20432 291172 geth_top 42279 38552 646875 yadan 63514 31280 483369 beihai 211236 133086 2161829 SHMS 268721 251772 3610024 nvdla 289344 226974 3708427 2JUC 348985 323195 4880442 2JUC 348985 323195 4880442 349422 449847 6967779 beihai2 356818 393308 5491501 AIMP 742210 335618 9980714 AIMP2 816677 560525 9133333 ysyx6 1090820 1029515 15486068 wukong 1102663 1032718 15653639 ysyx41 1123368 1105564 8248292 ysyx42 1173610 1147953 17416249 T1 1262053 1227098 18769036 T1_mach 2222669 2162147 33115508 anhu-G 2793215 2646672 42524007 openC910 3282828 2948743 52259408 T1_sand 4816399 4728816 79050737				
aes_core 17940 17371 310215 aes 19181 18117 325550 salsa20 21270 20432 291172 jpeg 27671 29160 366397 eth_top 42279 38552 646875 yadan 63514 31280 483369 beihai 211236 133086 2161829 SHMS 268721 251772 3610024 nvdla 289344 226974 3708427 ZJUC 348985 323195 4880442 iEDA23 368147 335112 5132004 ysyx452 494962 449847 6967779 beihai2 582645 393308 5491501 AIMP 742210 535618 9980714 AIMP2 816677 565525 913333 ysyx41 1123368 1105564 18248292 ysyx42 1173610 1147953 17416249 T1 1262053 1227098 18				
aes salsa20 21270 20432 291172 29160 salsa20 21270 20432 291172 29160 366397 eth_top 42279 38552 646875 yadan 63514 31280 483369 beihai 211236 133086 2161829 SHMS 268721 251772 3610024 nvdla 289344 226974 3708427 ZJUC 348985 323195 4880442 15EDA23 368147 335112 5132004 ysyx452 494962 449847 6967779 beihai2 582645 393308 5491501 AIMP 742210 535618 9980714 AIMP2 816677 56525 9133333 ysyx6 1090820 1029515 15486068 wukong 1102663 1032718 15653639 ysyx41 1123368 1105564 18248292 ysyx42 1173610 1147953 17416249 T1 1262053 1227098 18769036 T1_mach 2793215 2646672 42524007 openC910 3282828 2948743 52259408 T1_sand 4816399 4728816 79050737				
salsa20 21270 20432 291172 jpeg 27671 29160 366397 eth_top 42279 38552 464875 yadan 63514 31280 483369 beihai 211236 133086 2161829 SHMS 268721 251772 3610024 nvdla 289344 226974 3708427 ZJUC 348985 323195 4880442 iEDA23 368147 335112 513204 ysyx452 494962 449847 697779 beihai2 582645 393308 5491501 AIMP 742210 535618 9980714 AIMP2 816677 560525 9133333 ysyx41 1123368 1105564 1824829 ysyx42 1173361 1147953 17416249 T1 1262053 1227098 18769036 T1_mach 2793215 25466672 24524007 openC910 3282828 2948743				
19eq				
eth_top yadan 63514 31280 483369 beihai 211236 133086 2161829 SHMS 268721 251772 3610024 nvdla 289344 226974 3708427 ZJUC 348985 323195 4880445 jebhai2 368147 335112 5132004 ysyx4S2 494962 449847 6967779 beihai2 582645 393308 5491501 AIMP 742210 535618 9980714 AIMP2 816677 560525 9133333 ysyx6 1090820 1029515 15486068 wukong 1102663 1032718 15653639 ysyx41 1123368 1105564 18248292 ysyx42 1173610 1147953 17416249 T1 1262053 1227098 18769036 T1_mach 2793215 2646672 42524007 openC910 3282828 2948743 52259408 T1_sand 4816399 4728816 79050737	salsa20			
yadan 63514 31280 483369 beihai 211236 133086 2161829 SHMS 268721 251772 3610024 nvdla 289344 226974 3708427 ZJUC 348985 323195 4880442 iEDA23 368147 335112 5132004 ysyx452 494962 449847 697779 beihai2 582645 393308 5491501 AIMP 742210 535618 9980714 AIMP2 816677 560525 9133333 ysyx46 1090820 1029515 15486068 wukong 1102663 1032718 15655639 ysyx41 1123368 1105564 18248292 ysyx42 1173610 1147953 17416249 T1 1262053 1227098 18769036 T1_mach 2793215 25466672 42524007 openC910 3282828 2948743 52259408 T1_sand 4816399				
Seihai 211236 133086 2161829 SHMS 268721 251772 3610024 nvdla 289344 226974 3708427 ZJUC 348985 323195 4880442 iEDA23 368147 335112 5132004 ysyx4S2 494962 449847 6967779 beihai2 582645 393308 5491501 AIMP 742210 535618 9980714 AIMP2 816677 560525 913333 ysyx46 1009820 1029515 15486068 wukong 1102663 1032718 15653639 ysyx41 1123368 1105564 18248292 ysyx42 1173610 1147953 17416249 T1 1262053 1227098 18769036 T1_mach 2793215 2646672 24524007 openC910 3282828 2948743 52259408 T1_sand 4816399 4728816 79050737				
SHMS 268721 251772 3610024 nvdla 289344 226974 3708427 ZJUC 348985 323195 4880442 iEDA23 368147 335112 5132004 ysyx4S2 494962 449847 6967779 beihai2 582645 393308 5491501 AIMP 742210 535618 9980714 AIMP2 816677 560525 9133333 ysyx46 1090820 1029515 15486068 wukong 1102663 1032718 15653639 ysyx41 1123368 1105564 18248292 ysyx42 1173610 1147953 17416249 T1 1262053 1227098 18769036 T1_mach 2793215 5246672 42524007 openC910 3282828 2948743 52259408 T1_sand 4816399 4728816 79050737	yadan	63514	31280	483369
nvd1a 289344 226974 3708427 ZJUC 348985 323195 4880442 iEDA23 368147 335112 5132004 ysyx452 494962 449847 696779 beihai2 582645 393308 5491501 AIMP 742210 535618 9980714 AIMP2 816677 56525 913333 ysyx46 1102663 1032718 15653639 ysyx41 1123368 1105564 18248292 ysyx42 1173610 1147953 17416249 T1 1262053 1227098 18769036 T1_mach 2793215 2646672 42524007 openC910 3282828 2948743 52259408 T1_sand 4816399 4728816 79050737				
ZJUC 348985 323195 4880442 iEDA23 368147 335112 5132004 51952449847 6967779 5961542 449847 6967779 5961542 449847 6967779 5961542 449847 6967779 5961542 449847 6967779 696779 69				
iEDA23 368147 335112 5132004 ysyx452 449462 449847 6967779 beihai2 582645 393308 5491501 AIMP 742210 535618 9980714 AIMP2 816677 560525 9133333 ysyx6 1090820 1029515 15486068 wukong 1102663 1032718 15653639 ysyx41 1123368 1105564 18248292 ysyx422 1173610 1147953 17416249 T1 1262053 1227098 18769036 T1_mach 2222669 2162147 33115508 c9penC910 3282828 2948743 52259408 T1_sand 4816399 4728816 79050737				
ysyx4S2 494962 449847 6967779 beihai2 582645 393308 5491501 AIMP 742210 535618 9980714 AIMP2 816677 560525 9133333 ysyx61 1090820 1029515 15486068 wukong 1102663 1032718 15653639 ysyx41 1123368 1105564 18248292 ysyx42 1173610 1147953 17416249 T1 1262053 1227098 18769036 T1_mach 2222669 2162147 33115508 anhu-G 2793215 2646672 42524007 openC910 3282828 2948743 52259408 T1_sand 4816399 4728816 79050737				
beihai2 582645 393308 5491501 AIMP 742210 535618 9980719 S16407 560525 9133333 ysyx6 1090820 1029515 15486068 wukong 1102663 1032718 15653639 ysyx41 123368 1105564 18248292 ysyx42 1173610 1147953 17416249 T1 1262053 1227098 18769036 T1_mach 2793215 2646672 24524007 openC910 3282828 2948743 52259408 T1_sand 4816399 4728816 79050737				
AIMP 742210 535618 9980714 AIMP2 816677 560525 9133333 ysyx6 1090820 1029515 15486068 wukong 1102663 1032718 15653639 ysyx41 1123368 1105564 18248292 ysyx42 1173610 1147953 17416249 T1 1262053 1227098 18769036 T1_mach 2222669 2162147 33115508 copenC910 3282828 2948743 52259408 T1_sand 4816399 4728816 79050737				
AIMP2 816677 560525 9133333 ysyx6 1090820 1029515 15486068 wukong 1102663 1032718 15653639 ysyx41 1123368 1105564 18248292 ysyx42 1173610 1147953 17416249 T1 1262053 1227098 18769036 T1_mach 2222669 2162147 33115508 openC910 3282828 2948743 52259408 T1_sand 4816399 4728816 79050737	beihai2			
ysyx6 1090820 1029515 15486068 wukong 1102663 1032718 15653639 ysyx41 1123368 1105564 18248292 ysyx42 1173610 1147953 17416249 T1 1262053 1227098 18769036 T1_mach 2222669 2162147 33115508 nanhu-G 2793215 2646672 42524007 openC910 3282828 2948743 52259408 T1_sand 4816399 4728816 79050737	AIMP			
wukong ysyx41 1123368 1105564 18248292 ysyx42 1173610 1147953 17416249 T1 1262053 1227098 18769036 T1_mach 2292669 2162147 33115508 copenC910 3282828 2948743 52259408 T1_sand 4816399 4728816 79050737	AIMP2			
ysyx41 1123368 1105564 18248292 ysyx42 1173610 1147953 17416249 T1 1262053 1227098 18769036 T1_mach 2222669 2162147 33115508 nanhu-G 2793215 2646672 42524007 openC910 3282828 2948743 52259408 T1_sand 4816399 4728816 79050737	ysyx6	1090820	1029515	15486068
ysyx42 1173610 1147953 17416249 T1 1262053 1227098 18769036 T1_mach 2222669 2162147 33115508 nanhu-G 2793215 2646672 42524007 openC910 3282828 2948743 52259408 T1_sand 4816399 4728816 79050737	wukong	1102663	1032718	15653639
T1 1262053 1227098 18769036 T1_mach 2222669 2162147 33115508 2793215 2646672 42524007 openC910 3282828 2948743 52259408 T1_sand 4816399 4728816 79050737	ysyx41	1123368	1105564	18248292
T1_mach 2222669 2162147 33115508 nanhu-G 2793215 2646672 42524007 openC910 3282828 2948743 52259408 T1_sand 4816399 4728816 79050737	ysyx42	1173610	1147953	17416249
nanhu-G 2793215 2646672 42524007 openC910 3282828 2948743 52259408 T1_sand 4816399 4728816 79050737	T1	1262053	1227098	18769036
nanhu-G 2793215 2646672 42524007 openC910 3282828 2948743 52259408 T1_sand 4816399 4728816 79050737	T1_mach	2222669	2162147	33115508
openC910 3282828 2948743 52259408 T1_sand 4816399 4728816 79050737		2793215	2646672	42524007
T1_sand 4816399 4728816 79050737				
Total 23.26M 21.47M 347.15M				
	Total	23.26M	21.47M	347.15M

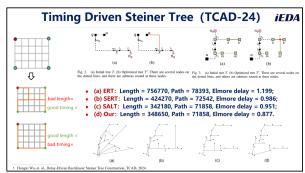
iEDA Docs and Videos

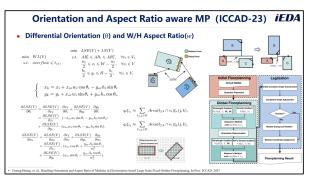
- iEDA Readme:
 - https://gitee.com/oscc-project/iEDA
- iEDA Userguide:
 - https://ieda.oscc.cc/tools/ieda-platform/guide.html
- iEDA install, compile, run videos:
 - https://www.bilibili.com/video/BV1mp4y1P7C7/?spm id from=333.999.0.0
- RTL-to-GDS videos base iEDA's python APIs
 - https://www.bilibili.com/video/BV1Brm8YVEaY/?spm_id_from=333.999.0.0&vd_source=2ac617c2
 41afd7f9774b0add4e647179
 - https://atomgit.com/chipcreative/RTL2GDS/blob/master/notebook.ipynb

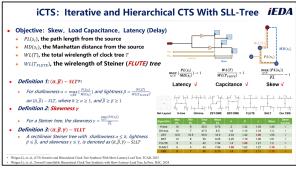
iEDA Research

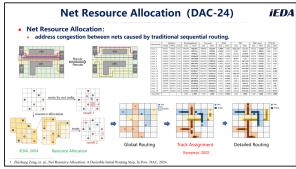














RTL-to-GDS (iEDA)



RTL-to-Vector (AiEDA)



Vector-to-Vector (AiEDA-Models)

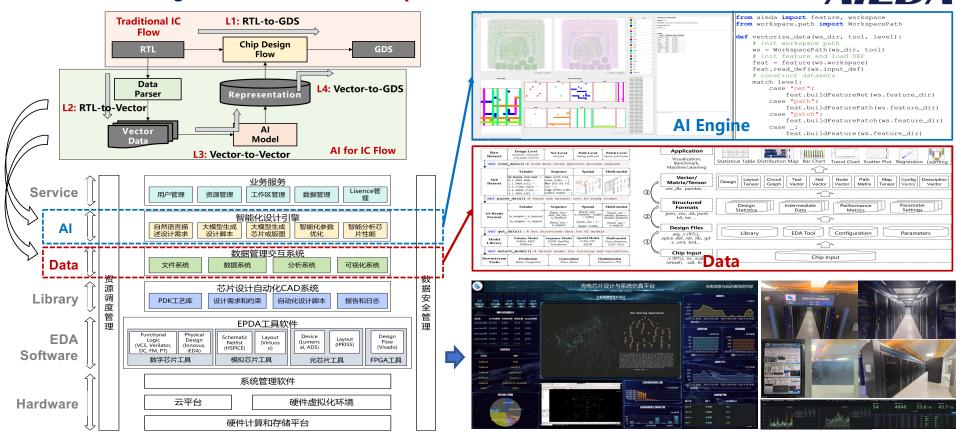


Vector-to-GDS (iPCL)

EDA + Data + AI (EDA)

- New foundation for AI-for-IC design (vectorized data and AI-EDA library)
 - Traditional IC design mainly relies on EDA tools and CPU,
 - Al-for-IC design needs vectorized chip data, Al-EDA libraries, and GPU.

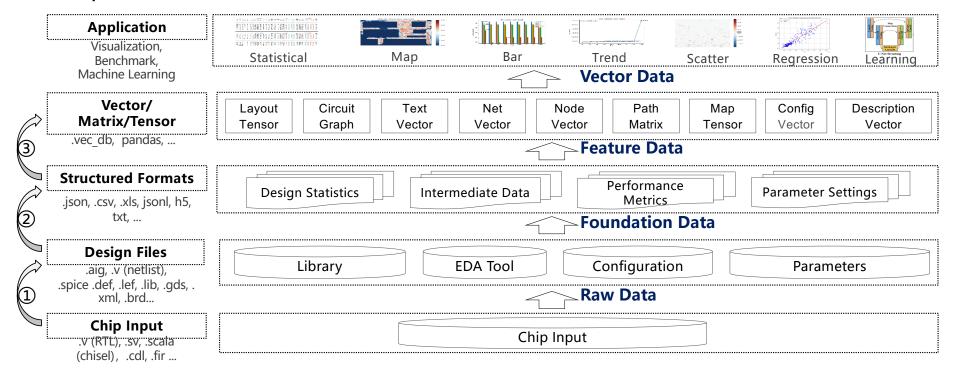




AiEDA: Data Transformer

AiEDA (Al-Aided Design Library):

 Supports RTL-to-Vector (RTL2Vec), converting chip data into vectorized data that can be input into AI models.



Chip Input ⇒ Raw Data ⇒ Foundation Data ⇒ Feature Data ⇒ Vector Data

File Formats and Formation

Involving File Formats

Input
.spm (System Planner model file), .xml, .csv
(parameter settings)
.lef (Library Exchange Format), .def (Design
Exchange Format), .lib (timing library), .verilog
(netlist)
.cdl (circuit description language), .spice
(simulation netlist), .oa (OpenAccess database)
.spef, .spf (detailed parasitic files), .vcd (value
change dump)
.lef, .def, .sdc
.xml (thermal parameters), .csv (thermal boundary
conditions)
.sdc, .lib, .spef
.gds, .lef, .rule (DRC rules)
.brd (board file), .xml (design settings)
.brd, .dra (design file), .pcb
Likely .db or .oa (under development, format not
finalized)

Main Storage Information

- Text Description
- Circuit Netlist
- Layout Geometry
- Physical Library

Text Description

Circuit **Netlist**

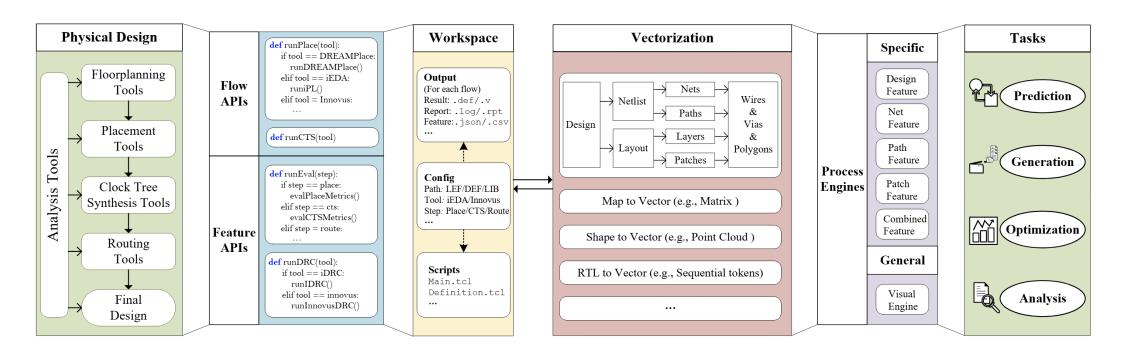


Physical Library

Geometry **Layout**

AiEDA: Design-to-Data-to-Analysis

Run Chip Design Flow ⇒ Decompose and Convert Data ⇒ Analyze and Report Result



Run Chip Design Flow

- Commercial flow: Call commercial EDA tool, python -> TCL
 - It can generate config parameters in Json
 - Using python to generate TCL scripts based on Json config.
- Open-source flow: Extend python API on Open-source EDA tools

```
flow
from ai infra.data manager.data manager import DataManager
                                                                                                        ☐ CTS.tcl
from ai infra.operation.script gen.module.gen definition import GenerateTcl
                                                                                                                                    # create directory

  □ dataOut copy.tcl

                                                                                                                                    if {![file isdirectory ${WORKSPACE_OUTPUT}/innovus/DB/${VERSION}]} {
from ai infra.operation.script gen.module.gen main import GenerateTCLMain
                                                                                                                                        #file mkdir ${WORKSPACE OUTPUT}/innovus/DB/${VERSION}
                                                                                                       □ dataOut.tcl
class GenerateTCL:

□ drc.tcl

                                                                                                                                    # specify tech lef and ref lef
                                                                                                                                    set init lef file " \
    """gernerate all TCL scripts"""
                                                                                                       ☐ floorplan.tcl
                                                                                                                                        $project_config(technology,tlef,[dict get $project_config(design,track) $DESIGN]
    def init (self):
                                                                                                                                        $project config(physical,lef,[dict get $project config(design,track) $DESIGN])
                                                                                                        fpt.tcl
                                                                                                                                        $project_config(physical,lef,memory) \
         pass
                                                                                                                                        $project config(physical,lef,ipio) \
                                                                                                        ☐ generate_ccopt_spec.tcl
    def generate tcl main(self, data manager : DataManager):
                                                                                                        generate_cts_rpt.tcl
                                                                                                                                    # specify verilog
         """generate tcl main"""
                                                                                                        ☐ generate_rpt/tcl
                                                                                                                                    set init_verilog $VERILOG_INPUT_PATH
         tcl main = GenerateTCLMain(data manager)

□ initialize.tcl

                                                                                                                                     # specify multi mode multi corner
         tcl main.save json()
                                                                                                                                    set init_mmmc_file ${SRC_SCRIPT}/${EDA_TOOL}/flow/set_mmmc.tcl
                                                                                                        □ place.tcl
                                                                                                        □ place_noOPT.to
    def generate_tcl_definition(self, data_manager : DataManager):
                                                                                                                                    set init_pwr_net "VDD"
                                                                                                                                    set init gnd net "VSS"
         """generate tcl definition"""
                                                                                                        □ prePlace.tcl
                                                                                                                                    set enc source verbose 0
         tcl definition = GenerateTclDefiniton(data manager)
                                                                                                                                    set init_remove_assigns 1

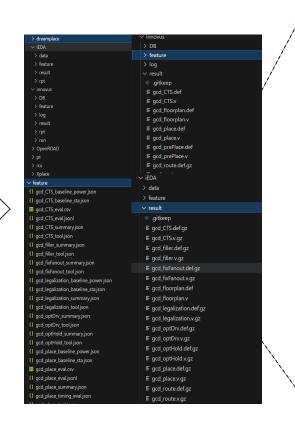
□ route.tcl

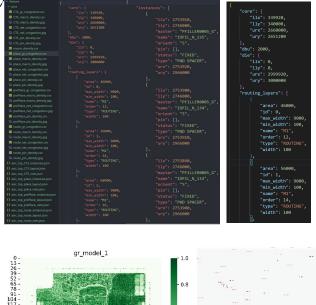
                                                                                                                                    set init_design_uniquify 1
         tcl definition.save json()
                                                                                                        □ set mmmc.tcl
                                                                                                                                    init design
```

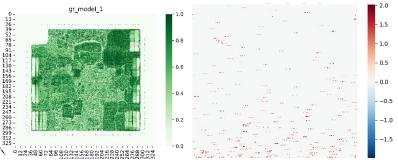
Data Genration and Feature Extraction

 AiEDA extract and various stage features and evaluation metrics from open-source and commercial tool reports, logs, and memory data.

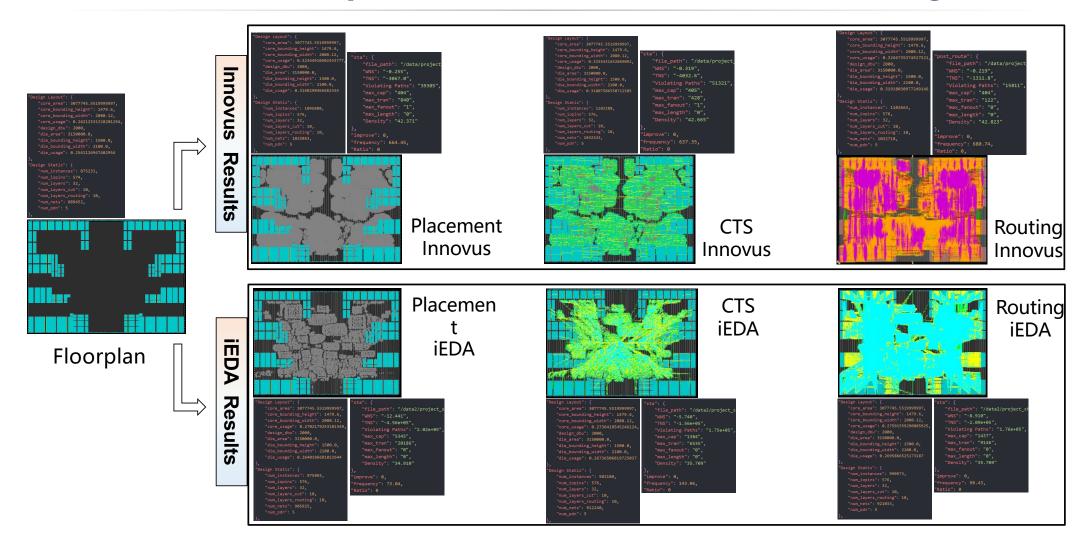
```
from aieda.workspace import workspace create
from aieda.flows import RunFlow
from aieda.data import RunFeature
def generate_data(ws_dir, tool, step):
  ws = workspace create(ws dir, tool)
  match step:
     case "floorplan":
      flow = RunFlow.runFP(ws, tool)
       feat = RunFeature.fp(ws, tool)
     case "place":
      flow = RunFlow.runPL(ws, tool)
       feat = RunFeature.pl(ws, tool)
     case "cts":
      flow = RunFlow.runCTS(ws, tool)
       feat = RunFeature.cts(ws, tool)
     case "routing":
      flow = RunFlow.runRT(ws, tool)
       feat = RunFeature.rt(ws, tool)
     case:
       flow = RunFlow.run(ws, tool)
       feat = RunFeature.flow(ws, tool)
  return result
```





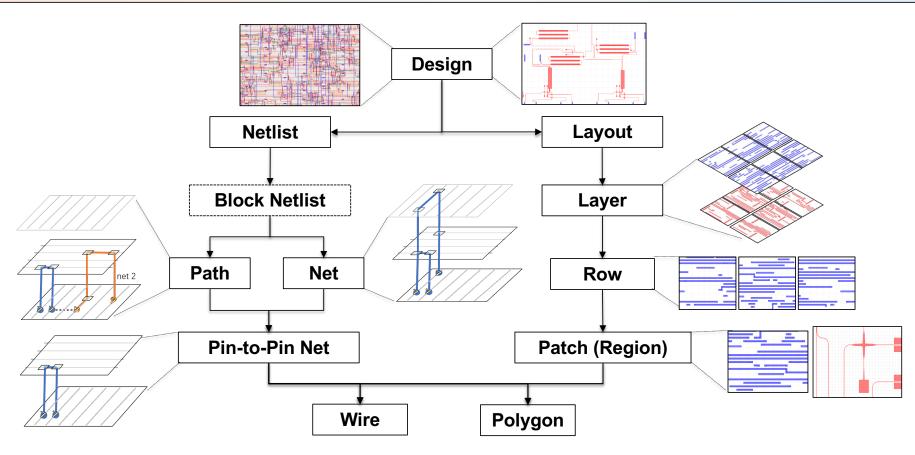


Feature Comparison (Innovus & iEDA) "wukong"



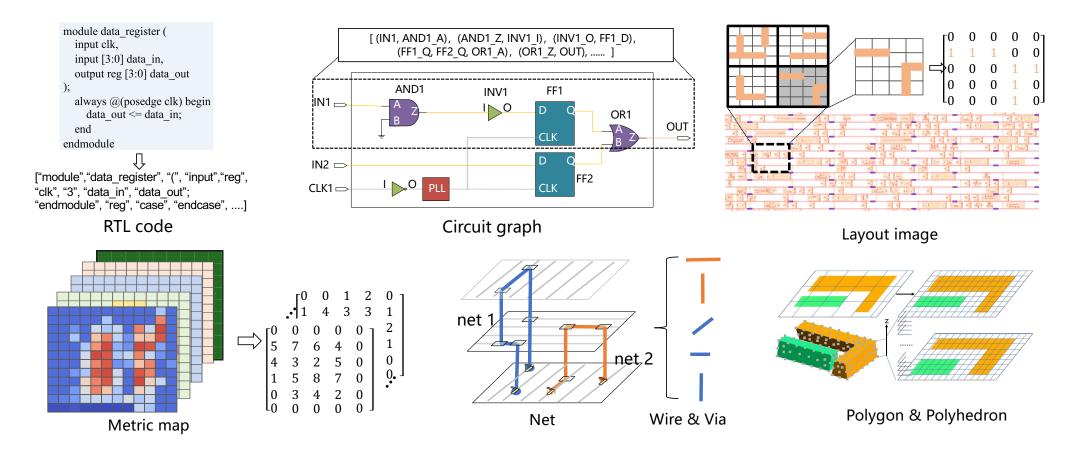
Design Decomposition

A design (netlist/layout) can be decomped as different modular (graph, image) and scale (net, path, patch)

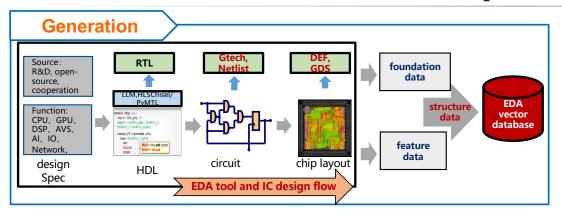


Design Transformer

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

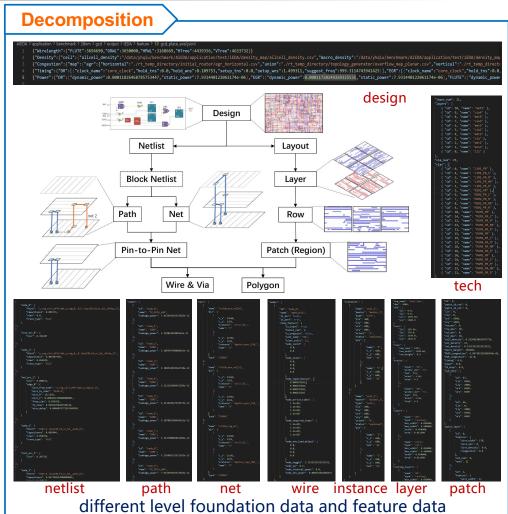


Gneration, Decompostion, Transformation



Transformation

```
from aieda.workspace import workspace_create
                                                                                                                      {} patch 2.json
from aieda data import RunVectors
                                                                                       () net 3.isor
                                                                                       {} net 4.isor
                                                                                       {} net 5.isor
def vectorize_dataset(workspace_dir, tool, level):
                                                                                       () net 6.isor
                                                                                                                      {} patch_6.json
                                                                                       {} net 7.jsor
   ws = workspace_create(ws_dir, tool)
                                                                                       () net 8.json
   vectors = RunVectors(ws)
                                                                                       1 net 9 ison
   vectors.read_def(ws.input_def)
                                                                                       f) net 10.ison
                                                                                                                      {} patch 9.ison
                                                                                       {} net 11.ison
                                                                                                                      {} patch 10.ison
    # construct vectorization datasets
                                                                                       () net 12.ison
                                                                                       () net 13.ison
   match level:
                                                                                       wire paths
                                                                                                                      {} patch 13.ison
                                                                                        wire path 2025-03-28T23 01 42 1.vm
          case "design":
                                                                                        wire path 2025-03-28T23 01 42 2.vml
                                                                                       ! wire path 2025-03-28T23 01 42 3.vml
                          vectors.generateDesign(ws.vec_dir)
                                                                                        wire path 2025-03-28T23 01 42 4.vm
                                                                                                                      {} patch_17.json
                                                                                        wire path 2025-03-28T23 01 42 5.ym
                         vectors.generateNet(ws.vec_dir)
                                                                                                                       {} patch 18.ison
                                                                                                                       {} patch_19.json
                                                                                        wire_path_2025-03-28T23_01_42_7.ym
            case "graph":
                                                                                        wire_path_2025-03-28T23_01_42_8.ym
                                                                                                                      {} patch_20.json
                         vectors.generateGraph(ws.vec_dir)
                                                                                                                      {} patch 21.ison
                                                                                        wire_path_2025-03-28T23_01_42_10.yml
            case "path":
                                                                                                                       {} patch_22.json
                         vectors.generatePath(ws.vec dir)
                                                                                                                      {} patch 24.ison
            case "patch":
                                                                                                                      {} patch 25.json
                         vectors.generatePatch(ws.vec_dir)
                         feat.generateVectors(ws.vec_dir)
                                                                                                                      {} patch 28 ison
                                                                                                                      {} patch_29.json
```



Data Vectorization

- iDATA: includes source data, foundation data, feature data, vectorized data
 - iDATA covers 50 chip scales from 1K gates to 30M gates, including memories, CPUs, GPUs, and SoCs.
 - By design space exploration in logic synthesis and placement stage, we obtain 250 netlists and 1250 defs.

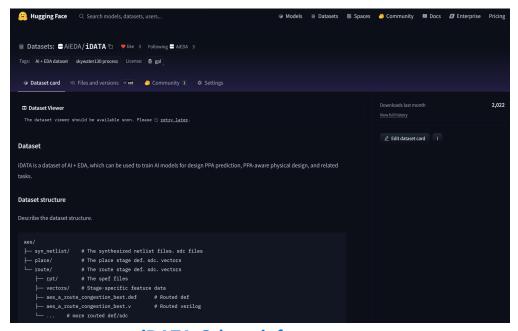


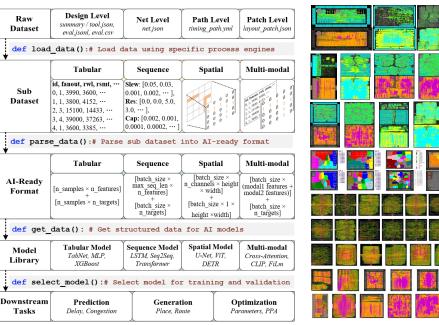
50 Chip RTLs

50*5*5 = 1250 defs

iDATA: A Dataset for AI-EDA

- We release iDATA in huggingface, includes source data, foundation data, feature data
 - iDATA encompasses multiple levels: design, RTL, layout, net, path, patch, and map levels.
 - **iDATA** can be used to **evaluate chips** and EDA tools, **train** Al4EDA **model** and large model, **embed** vector **database**.





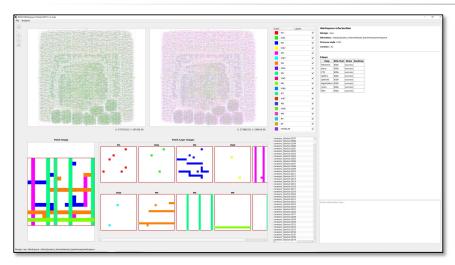
iDATA @ hugginface

Data Proccesor

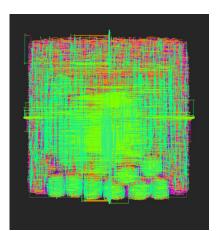
Chip Layouts

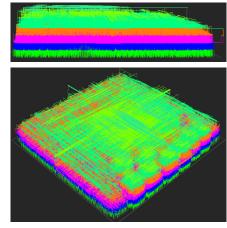
https://huggingface.co/datasets/AiEDA/iDATA

Analysis and Report



Visualization of the chip design process







GCD: 60 x 60 grid (3,600 patches)

Feature Statistics Across All Designs

 Cell Density Ang Mean: A781e-01, Ang Max: 1,000+00 Ang Hotspot Ratio (>90th percentile): 10,0%

 Pin Density Ang Mean: A781e-01, Ang Max: 1,000+10 Ang Hotspot Ratio (>90th percentile): 8,9%

 Compession Ang Mean: A1081e-101, Ang Max: 1,000+100 Ang Hotspot Ratio (>90th percentile): 8,5%

 Timing Ang Mean: Ann, Ang Max: Ann Ang Hotspot Ratio (>90th percentile): 0,0%

 Prover Ang Mean: Ann, Ang Max: Ann Ang Hotspot Ratio (>90th percentile): 0,0%

 Richard Ang Mean: Ann Ang Max: Ann Ang Hotspot Ratio (>90th percentile): 0,0%

 Richard Ang Mean: Ann Ang Max: A108e-00 Ang Hotspot Ratio (>90th percentile): 10,0%

 Richard Ang Mean: A108e-05, Ang Max: 3,829e-04 Ang Hotspot Ratio (>90th percentile): 10,0%

 Richard Ang Mean: A108e-04, Ang Max: A272e-01 Ang Hotspot Ratio (>90th percentile): 10,0%

 Richard Ang Mean: A108e-04, Ang Max: A270e-03

 Ang Hotspot Ratio (>90th percentile): 10,0%

Per Design Detailed Summary

 • GCD (60x60)

 Max: Congestion: -1,000, Max Timing: nan, Max Power nan

 • Most non-uniform feature: net density (variance: 1,600e-01)

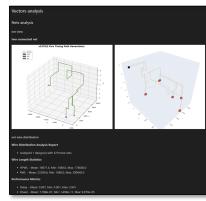
 **Coll Density Ang Mean: A108e-04

 **

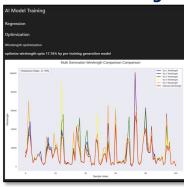
chip design report

process data analysis

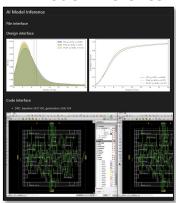
data vectorization



Al model training



Al model inference



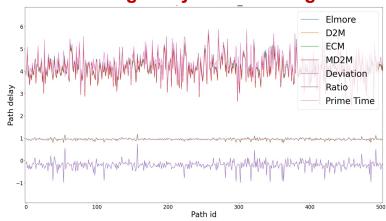
- 01
- **RTL-to-GDS (iEDA)**
- 02
- **RTL-to-Vector (AiEDA)**
- 03
- **Vector-to-Vector (AiEDA-Models)**
- 04

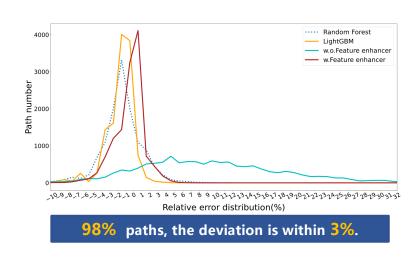
Vector-to-GDS (iPCL)

Al-Aided Design (AAD) for Chip Design

- How AI technology can assist in IC analysis and design.
 - Metric Evaluation
 - Metric Analysis: For example, by fitting delays with wirelengths, we can achieve 98% timing paths whose deviations with truth dalay are within 3%.
 - **Reduce Evaluation Time:** We only need to calculate the wirelengths and call the AI model for inference, which is almost as effective as using STA tools, then we may reduce significant evaluation time.
 - Al may provide greater time savings in Thermal, Stress, SI/PI analysis.

Fitting Delay with Wirelength



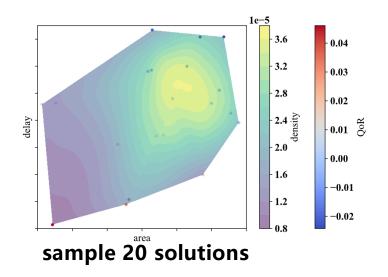


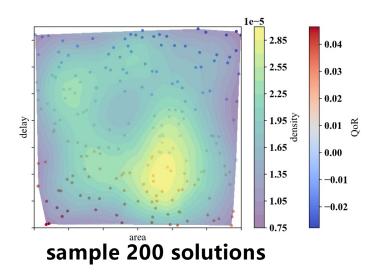
Al-Aided Design (AAD) for Chip Design

How AI technology can assist in IC analysis and design.

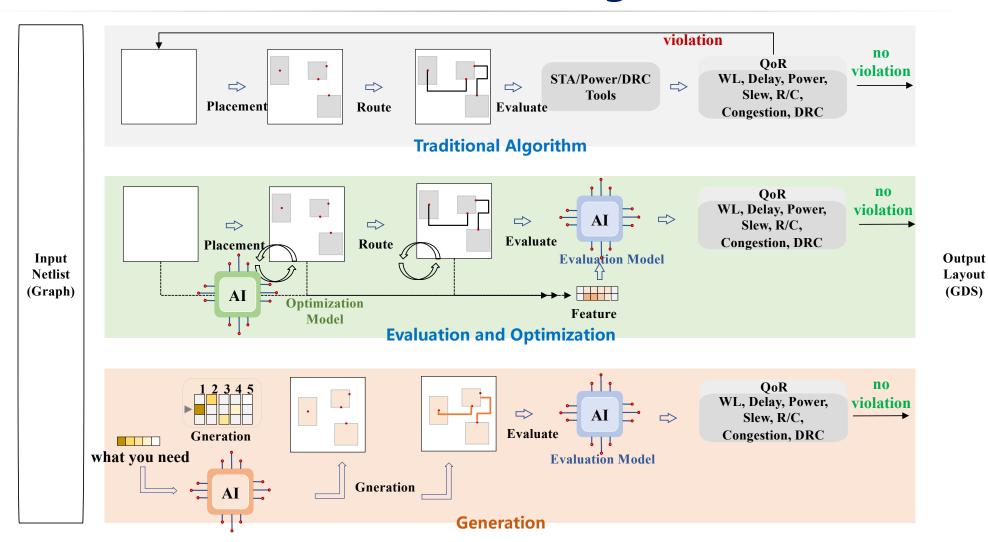
Solution Exploration

- Solution Distribution: By performing sampling on solutions and evaluating through prediction of objectives, the distribution of the solution space can be obtained.
 - When we sample 20 solutions, we observe that the solution with the highest probability (brightest region) is in the top-right corner.
 - When the sample size increases to 200 solutions, the highest probability region begins to shift toward the bottom-left corner.



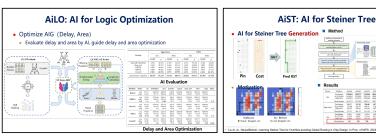


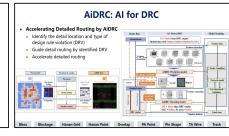
AI4EDA Paradigm

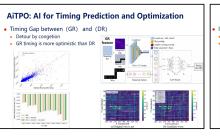


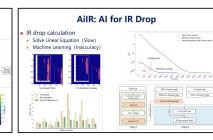
AiEDA-Models

- Based on AiEDA and iDATA, 16 AIEDA models were trained for tasks in the EDA field.
 - 1) Logic Optimization, 2) Steiner Tree Generation, 3) Design Rule Checking and Prediction, 4) Timing Prediction and Optimization, 5) IR Drop Calculation
 - 6) Tech Mapping, 7) Timing Optimization, 8) Timing Analysis, 9) Parameter Optimization, 10) Logic Synthesis









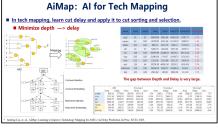
[TODAES-25]

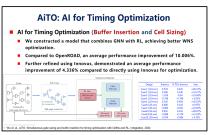
[NIPS-24]

[TODAES-25]

[TODAES-25]

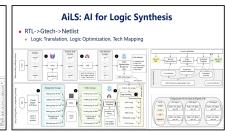
[ICCD-24]











[ICCD-23, TCAD-25] [Integration-24]

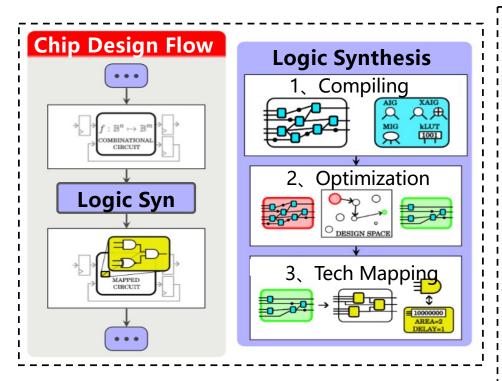
[ISEDA-23]

[TODAES-25]

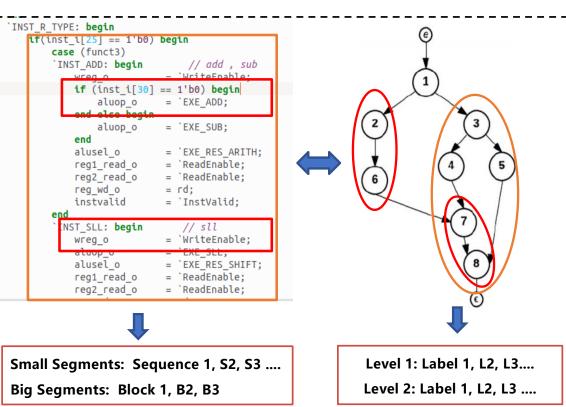
[TCAD-25]

AiRTL: Metrics Estimation of RTL by LLM

• Extract the features of the gate circuits for each RTL module, and fine-tune LLM to achieve the estimation from RTL to gate circuits, and then estimate the PPA of the gate circuits



Logic Synthesis Flow



RTL - Netlist Estimation

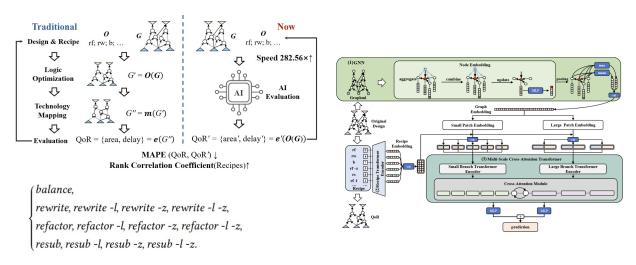
AiLO: Metric Prediction and Optimization for Logic Optimization

 Metrics of AIG (Delay and Area) are predicted by AI, and this is used to guide heuristic optimization.

Fitness

Rewards

Trust Region



Crossover

Mutate ...

Process

HQ Data

QoR₁

QoR₂

QoR,

	Open Core				EPFL			
Methods	area		delay		area		delay	
	MAPE	ρ	MAPE	ρ	MAPE	ρ	MAPE	ρ
OpenABC (Baseline)	3.55%	34.35%	4.35%	16.25%	6.27%	19.52%	6.78%	12.60%
Yang et al.	3.61%	33.17%	3.66%	23.14%	6.15%	19.60%	5.29%	27.02%
LOSTIN	3.55%	42.64%	3.59%	31.36%	5.65%	20.55%	4.85%	23.70%
GNN-H	3.59%	44.35%	3.65%	32.85%	5.65%	24.25%	5.07%	37.11%
CrossLO	2.96%	54.70%	2.97%	50.99%	4.15%	55.32%	4.45%	44.12%
nprovement vs Baseline	16.76%	37.21%	31.72%	68.13%	33.82%	64.71%	34.38%	71.449

Al Prediction Accuracy

Methods	Index	i2c	multiplier	max	priority	square	mem_ctrl	voter	log2	Mean
Resyn2	area delay	837.77 164.74	22669.38 2677.72	2848.55 2083.68	632.68 2076.20	16210.02 2520.69	31503.63 1041.40	13855.36 638.00	26929.54 3725.86	-
DRiLLS	area delay QoR	798.95 162.24 6.15	21780.40 2681.53 3.78	2146.11 2117.49 23.04	458.45 929.46 82.77	14836.99 2539.33 7.73	29104.49 992.21 12.34	11042.59 688.83 12.33	26203.02 3747.92 2.11	18.78
NSGA-II	area delay QoR	812.31 149.83 10.74	21782.95 2677.63 3.91	2251.78 2096.09 20.35	451.55 903.66 85.10	15785.72 2265.05 12.76	29308.02 939.29 16.77	11341.17 621.80 20.68	26308.24 3705.36 2.86	21.65
BOiLS	area delay QoR	796.92 150.80 13.34	21580.75 2677.98 4.79	2215.18 2070.17 22.88	404.85 885.36 93.37	14822.00 2541.28 7.75	28074.80 964.79 18.24	10987.76 672.20 15.34	26087.98 3729.65 3.02	22.34
CrossLO	area delay QoR	814.16 149.20 12.25	21672.34 2677.73 4.40	2208.12 2072.92 23.00	399.74 906.42 93.16	15671.89 2258.97 13.70	28483.80 938.83 19.44	11327.27 616.16 21.67	26184.80 3710.46 3.28	23.86

Delay and Area Improvement

Add & Guide

Recipe;

EDA tool

Logic Optimization

Technology

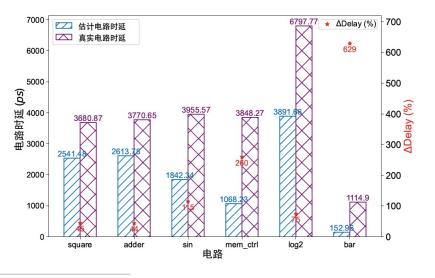
mapping

[•] Ye Cai, Rui Wang, et. al., AiLO: A Predictive Framework for Logic Optimization Using Multi-Scale Cross-Attention Transformer, TODAES, 2025.

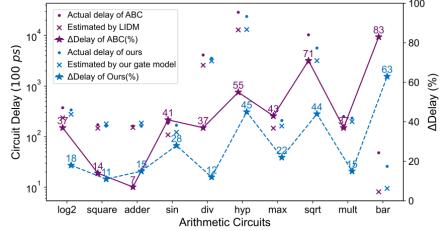
AiMap-2.0: Delay-driven Mapping by Parameter Learning

- Tech mapping lacks prior knowledge of gate delay. The alignment of the AIG before mapping and the netlist after mapping is difficult.
- resulting in a significant gap in physical information between the mapping before and after.

max.blif	#AND	# Depth	Delay(ps)
Script1	3631	37	598.38
Script2	2917	62	599.18
Script3	4255	40	650.27



best arrive time of node n $A(n) = \min_{M} \{A(n, M)\}$ $A(n, M) = \max_{i} \{A(i) + \delta(i, g)\} \longrightarrow \text{estimated delay}$ candidate match M of node n



[•] Junfeng Liu, et. al., "A Delay-Driven Iterative Technology Mapping Framework, TCAD, 2024.

AiCap: Fast Capacitance Calculation based on Point Clouds

Al for 3D Capcitance Extration

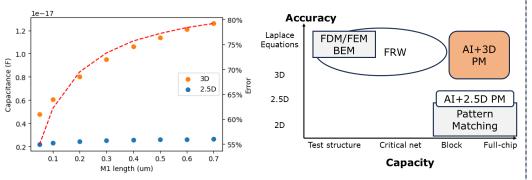
Laplace's equation

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

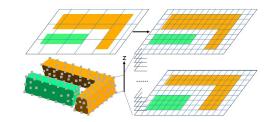
• Numerical Methods:

FDM, FEM, BEM, RWM

2.5D & 3D



Point Cloud

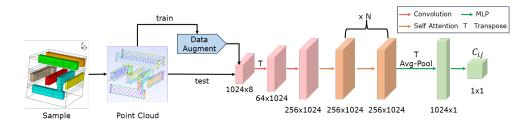


High resolution ensures unique geometry

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;

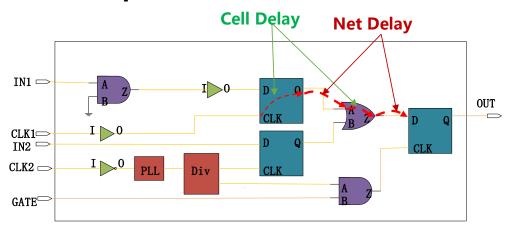
[•] S. Wu, et. al., An Adaptive Partition Strategy of Galerkin Boundary Element Method for Capacitance Extraction, ASPDAC, 2023.

[•] Y. Liang, et. al., PCT-Cap: Point Cloud Transformer Network for 3D Capacitance Extraction, ISEDA, 2024

AiSTA: Al Improves the Accuracy of iSTA

■ Al for Timing Analysis: Fit delay calculation of path delays.

■ (compared with commercial tools)



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

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

98% paths, the deviation is within **3%**.

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

Basic Feature

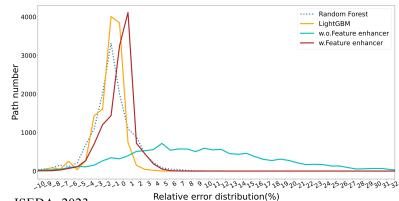
$$D_{Elmore} = \sum_{i \in N} (R_i * \sum C_D)$$

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

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

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

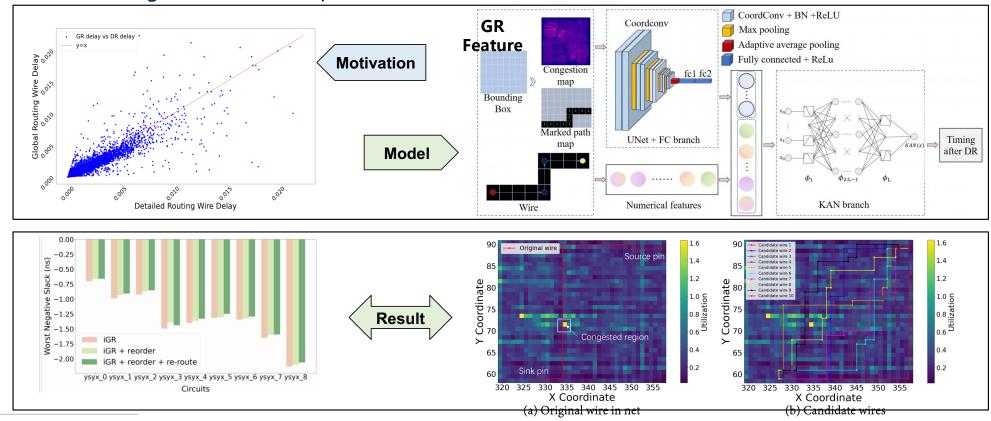
Enhance Feature



[•] H. Liu, et. al, Accurate Timing Path Delay Learning using Feature Enhancer with Effective Capacitance, ISEDA, 2023.

AiTPO: Timing Estimation and Optimization with Congestion

- Timing deviation between global routing (GR) and detailed routing (DR)
 - Local congestion can lead to wire looping.
 - The timing of GR is more optimistic than that of DR.

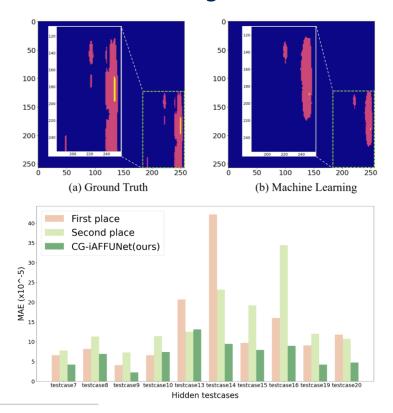


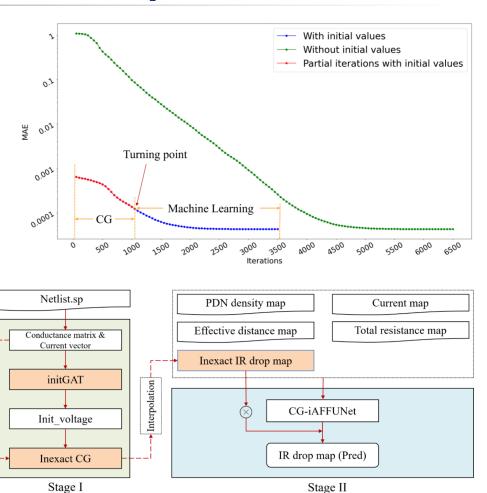
• He Liu, et. al., AiTPO: KAN-UNet Heterogeneous Network for Timing Prediction and Optimization at Global Routing, TODAES, 2025.

AilR: ML and CG to Solve IR Drop (ICCD-24)

IR drop Calculation

- Solving linear equations (slow),
- Machine learning methods (inaccurate)

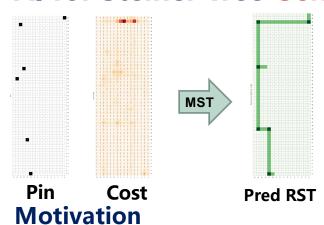


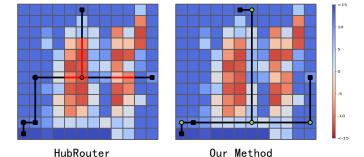


[•] He Liu, et. al., Simultaneous Conjugate Gradient and iAFF-UNet for Accurate IR Drop Calculation,, In Proc. ICCD, 2024.

AiST: Steiner Tree Generation by AI (NIPS-24)

Al for Steiner Tree Generation

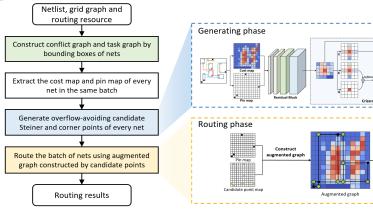




Aviod Congestion

Without Congestion

Method



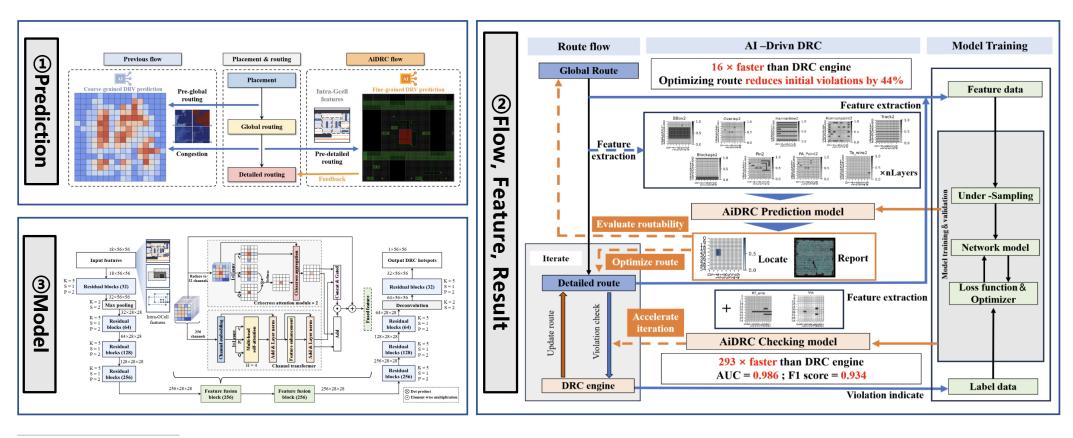
Results

Metric	Method	ADA01	ADA02	ADA03	ADA04	ADA05	NEW01	NEW02	NEW03
WL	GeoSteiner FLUTE+ES HR-GAN	3389601 3418461 3407033	3209172 3235803 3229110	9330748 9417934 9355980	8865643 8896007 8888775	9784471 9886249 9832110	2320456 2347941 2339204	4595235 4651033 4623006	7371273 7454720 7391055
Time (Sec)	NeuroSteiner	3438717	3247429	9459117	9003952	9915795	2365499	4668079	7480679
	GeoSteiner FLUTE+ES HR-GAN	83.17 118.48 593.02	111.92 187.03 780.44	320.08 396.51 1324.81	267.13 376.72 1387.01	261.43 360.68 1384.96	124.68 169.36 849.34	183.82 223.55 1221.16	315.48 438.79 1526.86
	NeuroSteiner	347.22	461.35	1351.91	1138.66	1106.54	390.34	446.68	1225.79
OF	GeoSteiner FLUTE+ES HR-GAN	35945 32518 35441	53848 50947 53652	142254 137104 142131	45050 42306 45230	102300 957704 102108	1734 1348 1516	1832 1713 1857	584761 558047 583901
	NeuroSteiner	82	255	728	97	431	5	35	10343

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

AiDRC: DRC Prediction and Optimization (TODAES-25)

- In detailed routing (DR), the specific locations of DRC are predicted, and guide the DR optimization
- Reduce evaluation time, and reduce the number of total iterations



[•] Yifan Li, et. al., AiDRC: Accelerating Detailed Routing by AI-Driven Design Rule Violation Prediction and Checking, TODAES, 2025.



RTL-to-GDS (iEDA)



RTL-to-Vector (AiEDA)



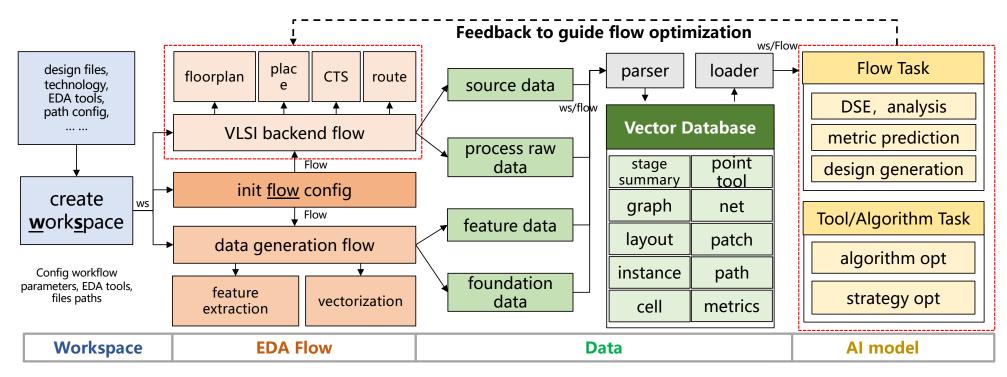
Vector-to-Vector (AiEDA-Models)



Vector-to-GDS (iPCL)

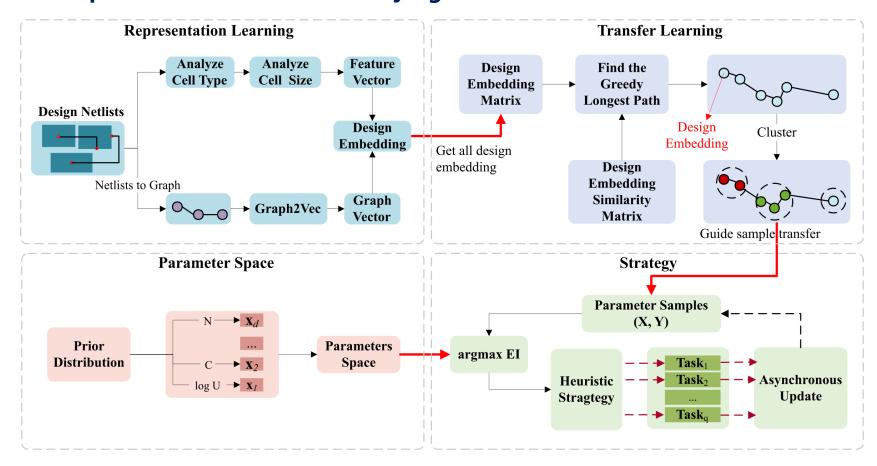
AiEDA Workflow

- We need integrate trained AI model into EDA tools or IC design flow to achieve AI aided design.
- Some AI models can be integrated into the chip design flow without any modification to existing EDA tools, whereas others require appropriate adjustments to the EDA tools or underlying algorithms before they can be effectively utilized.



Parameter Optimization for Placement (TODAES-25)

Flow-level optimization without modifying EDA tool internals

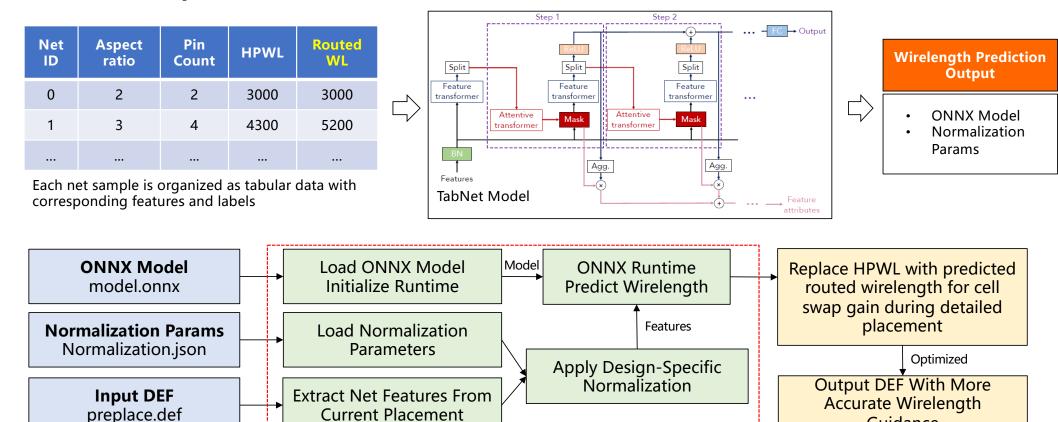


[•] Xinhua Lai, et. al., iPO: Constant Liar Parameter Optimization for Placement with Representation and Transfer Learning, TODAES, 2025.

Detailed Placement Optimization Guidance

Tool-level optimization with AI models embedded into EDA tool internals

Input Configuration



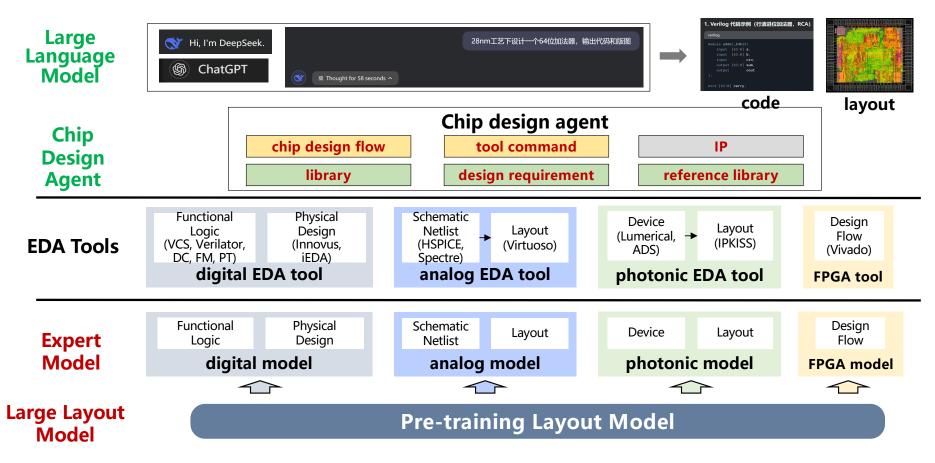
RunIEDA.run ai placement

Guidance

iEDA Placement (iPL)

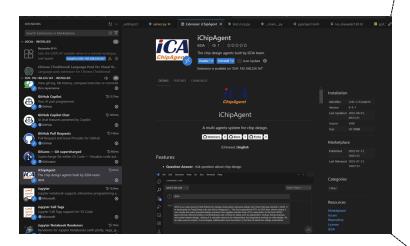
Large Model for Chip Design

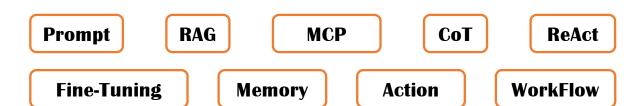
- 1) Build agents based on existing **LLM** to replace **manually designed** chips;
- 2) Design and train large layout models to replace EDA tools for chip design;

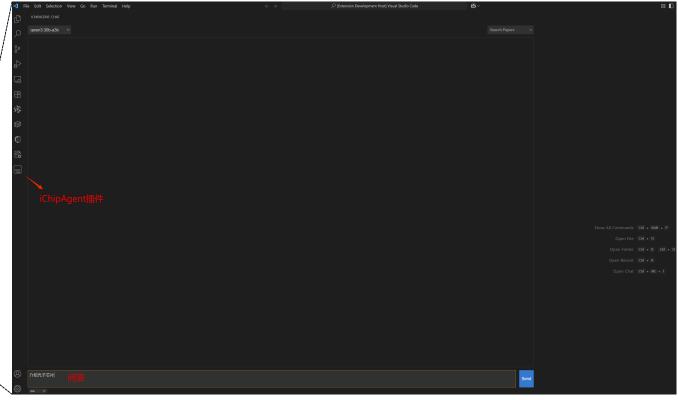


Chip Design Agent (iCA)

- Agent based on LLM
 - Knowledge Q&A
 - Data Retrieval
 - Script Generation
 - Tool Invocation
 - Analysis and Diagnosis
 - Layout Optimization

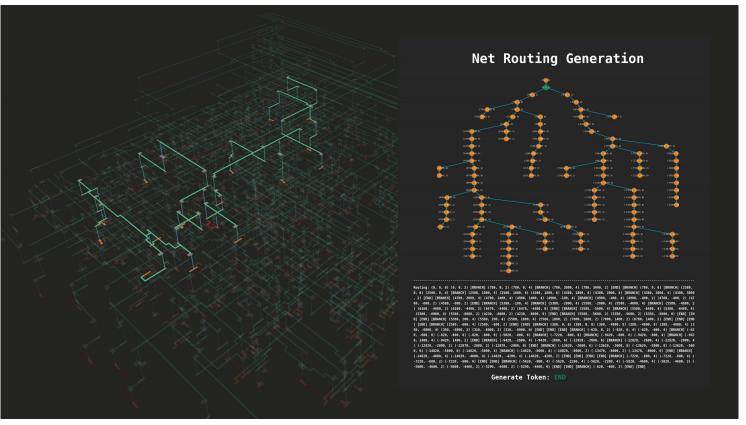






Pre-training for Chip Layout (iPCL)

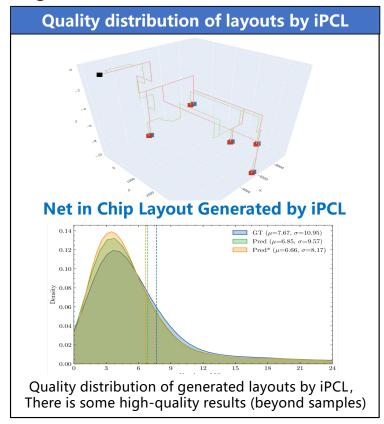
- iPCL: Design and train pre-training models, and replace EDA tools to design chips;
 - Design chip layout symbols, pre-training model with 100M parameters from 1T foudantion chip data
 - Achieve chip layout generation without EDA tools and human intervention

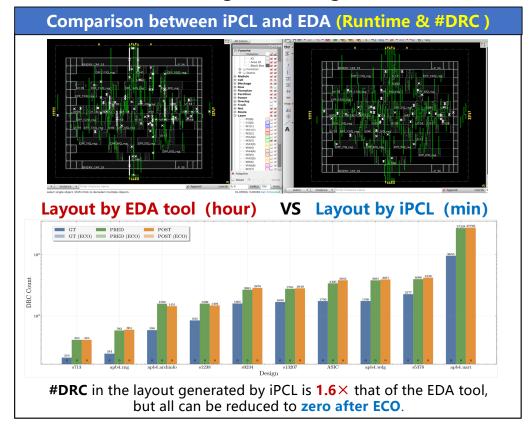




iPCL: AI (Large Model) Aided Design

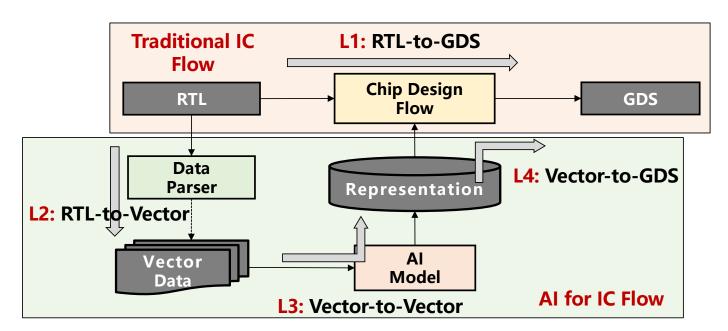
- iPCL directly generates the chip layout reduce the runtime from hour to minute
- The iPCL can simultaneously generate many layouts
 - There is a certain probability of obtaining high-quality results (beyond the learning samples)
- The generated result can be used to fabricated after conducting ECO using commercial tools





Conclusion

- Al for IC: Exploring new methodologies for IC design to enhance chip design quality and efficiency.
 - Traditional IC design flow (RTL-to-GDS)
 - Represent chip data using AI models (RTL-to-Vector), then use the representative information to guide chip design (Vector-to-GDS).



Conclusion

1. Open-Source EDA — iEDA

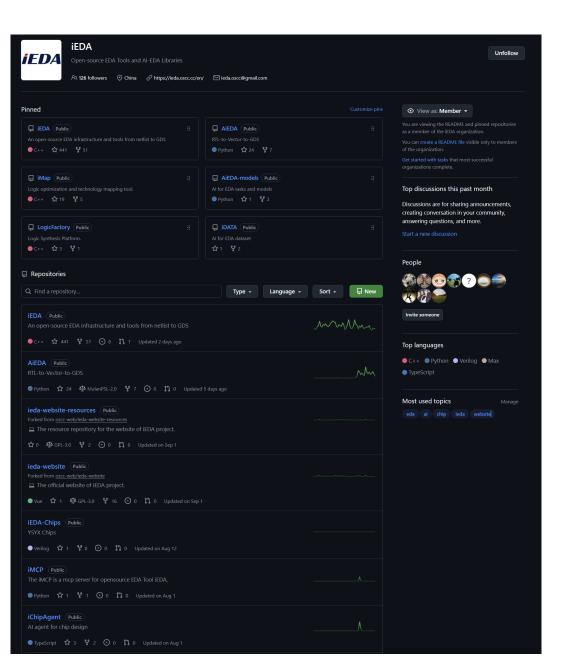
- **iEDA:** An open-source RTL-to-GDS chip design platform, includes tools such as LogicFactory, iMAP, iATPG, iNO, iFP, iPNP, iIR, iPL, iCTS, iTO, iRT, iSTA, iPA, iDRC, and iECO, as well as foundational modules like analysis, database, evaluation, interface, operation, and solver.
- iFlow: An automated backend chip design flow supporting 130/110/55/28nm technologies.
- iParsers: A collection of various EDA file parsers.

2. Intelligent EDA — AiEDA

- AiEDA: An intelligent EDA framework and toolkit supporting various AI+EDA tasks.
- AiEDA-Model: A collection of AI models designed for different EDA tasks.
- **iDATA:** A labeled EDA dataset, includes source data, process data, feature data, foundation data, vector data.

3. Foundation Models — iPCL

- iPCL: A large foundation model specialized for chip layout design.
- iCA: An agent for chip design, supporting knowledge Q&A, document retrieval, information extraction, script generation, tool invocation, report analysis, parameter optimization, result summarization.



Thanks



Web: https://ieda.oscc.cc

Code: https://github.com/OSCC-Project

Email: ieda.oscc@gmail.com



iEDA微信群



iEDA公众号



iEDA-B站视频



iEDA-QQ群