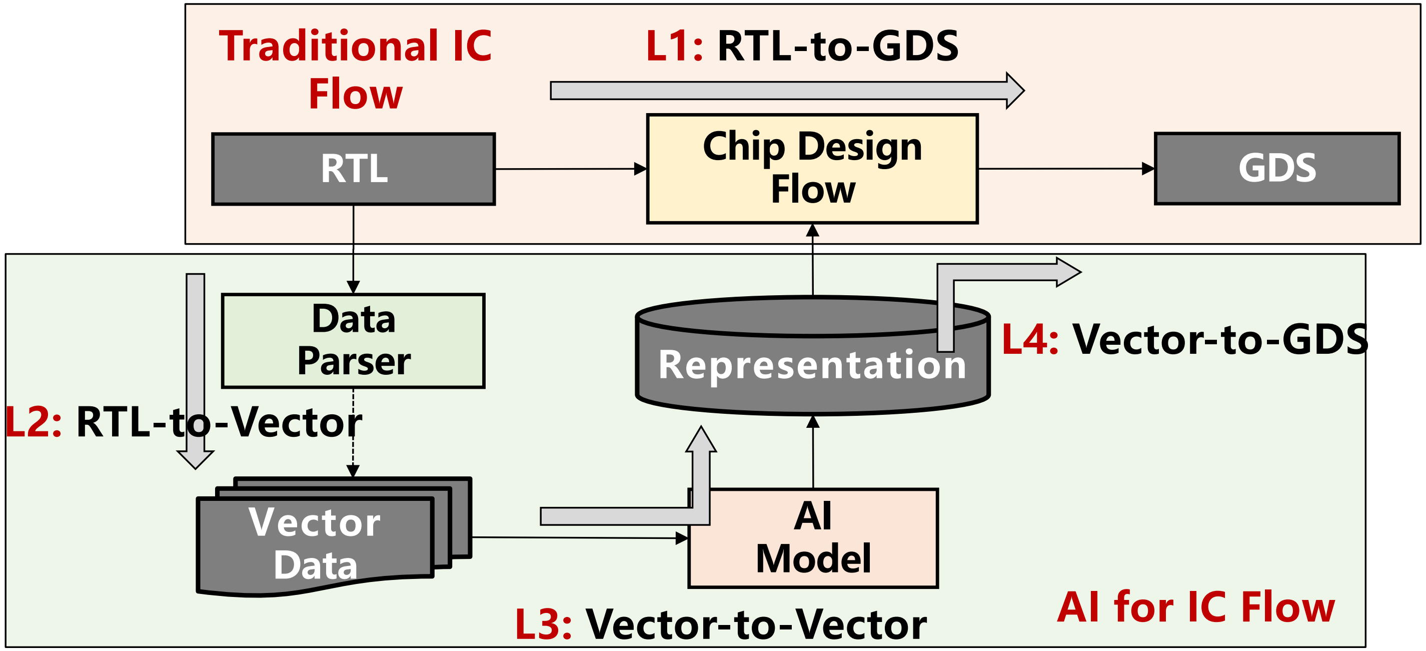
RTL-to-Vector-to-GDS

陶思敏（Simin Tao）

AI-aided Design (AAD)

- **AI 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).



01 RTL-to-GDS (iEDA)

02 RTL-to-Vector (AiEDA)

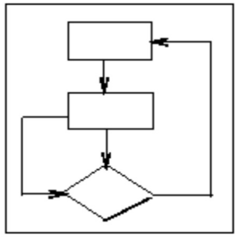
03 Vector-to-Vector

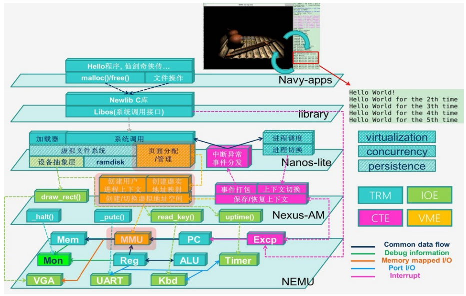
04 Vector-to-GDS

RTL-to-GDS Design Flow

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

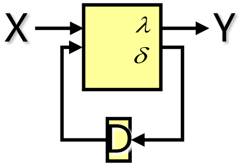
System Specifications → Architectural Design → Functional Design → Logic Synthesis → Tech Mapping → Physical Design → Physical Verification → Layout Processing → Fabrication → Package and Test

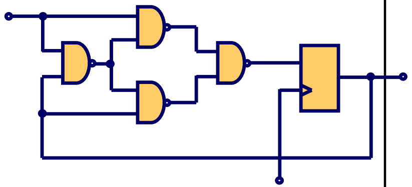


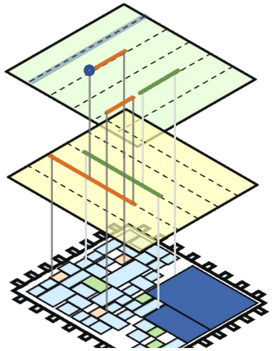


```
Hello World!
Hello World for the 2th time
Hello World for the 3th time
Hello World for the 4th time
Hello World for the 5th time
```

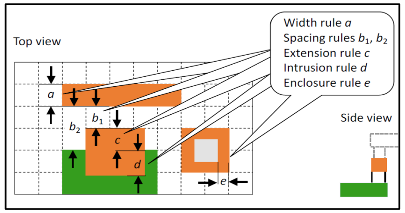
```
module conv;
reg [31:0] m[0:8192];
reg [12:0] pc;
reg [31:0] acc;
reg[15:0] ir;

always
  begin
    ir = m[pc];
    if(ir[15:13] == 3b’ 000)
      pc = m[ir[12:0]];
    else if (ir[15:13] == 3’ b010)
      acc = -m[ir[12:0]];
    ...
```

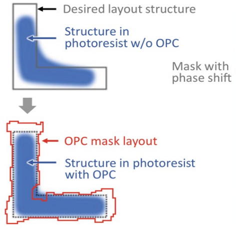




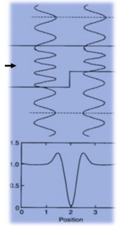


Timing, Power, DRC, ERC, LVS

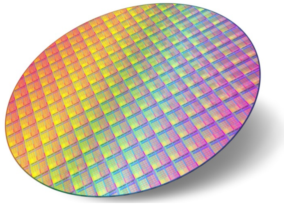


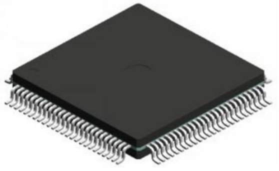

Desired layout structure
Structure in photoresist w/o OPC
Mask with phase shift
OPC mask layout
Structure in photoresist with OPC

Optical Proximity Correction (OPC)



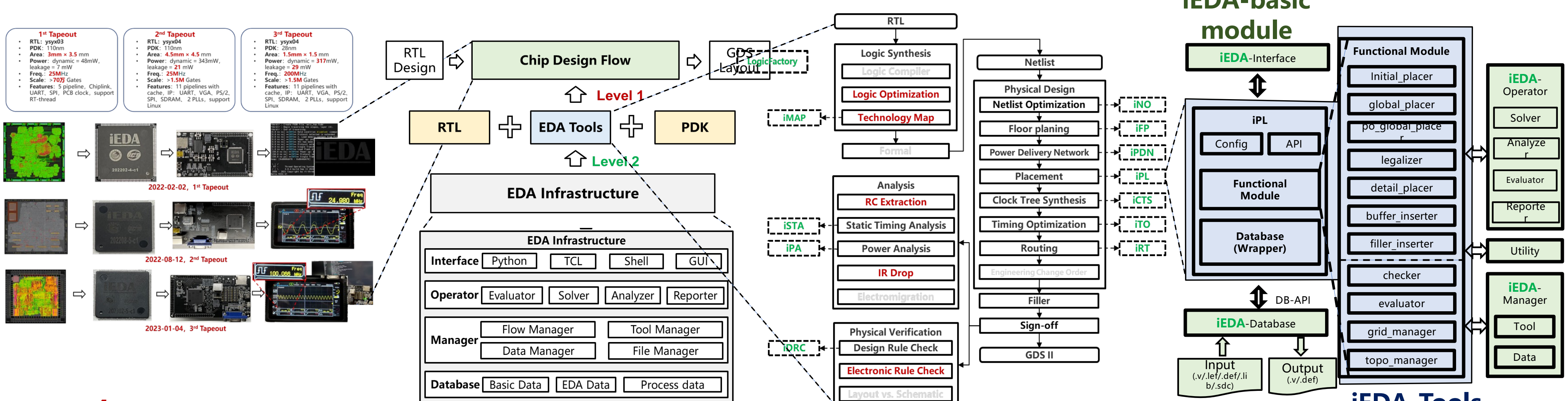
Phase Shift Mask





iEDA：RTL-to-GDS

- 1 EDA platform, 5 basic modules, 11 EDA tools, over 200 algorithm technologies, 300,000 lines of code, completed 4 tape-outs, 1 problem white paper, 6 technical manuals.
- 500 stars, 110 forks, 75 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 49,000 views.



1st Tapeout
- RTL：ysyx03
- PDK：110nm
- Area：3mm × 3.5 mm
- Power：dynamic = 48mW, leakage = 7 mW
- Freq.：25MHz
- Scale：>70万 Gates
- Features：5 pipeline，Chiplink、UART、SPI，PCB clock，support RT-thread

2nd Tapeout
- RTL：ysyx04
- PDK：110nm
- Area：4.5mm × 4.5 mm
- Power：dynamic = 343mW, leakage = 21 mW
- Freq.：25MHz
- Scale：>1.5M Gates
- Features：11 pipelines with cache，IP： UART、VGA、PS/2、SPI、SDRAM、 2 PLLs，support Linux

3rd Tapeout
- RTL：ysyx04
- PDK：28nm
- Area：1.5mm × 1.5 mm
- Power：dynamic = 317mW, leakage = 29 mW
- Freq.：200MHz
- Scale：>1.5M Gates
- Features：11 pipelines with cache，IP： UART、VGA、PS/2、SPI、SDRAM、 2 PLLs，support Linux

2022-02-02，1st Tapeout

2022-08-12，2nd Tapeout

2023-01-04，3rd Tapeout

iEDA-Tools

4 tape-outs

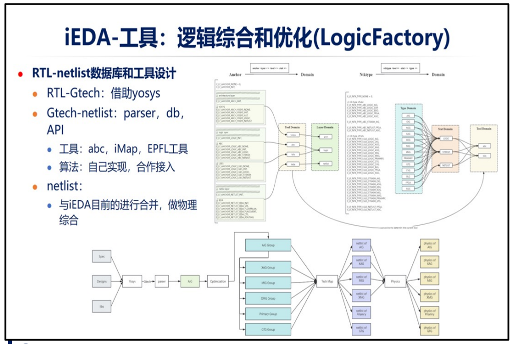
1 EDA basic module

11 tools, 200 technologies, 300,000 lines of code.

iEDA：RTL-to-GDS

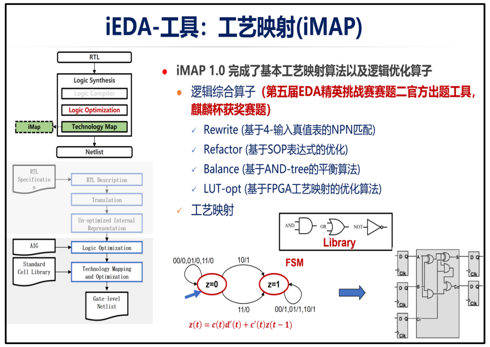
iEDA：RTL-to-GDS

- Logic synthesis (logic optimization, technology mapping)
- Physical Design (Floorplanning, Power Network Planning, Layout, Clock Tree Synthesis, Routing)
- Sign-off Verification (Static Timing Analysis, Timing Optimization, Power Analysis, IR Drop Analysis, Simulation, Parasitic Parameter Extraction, Design Rule Checking)



iEDA-工具：逻辑综合和优化(LogicFactory)

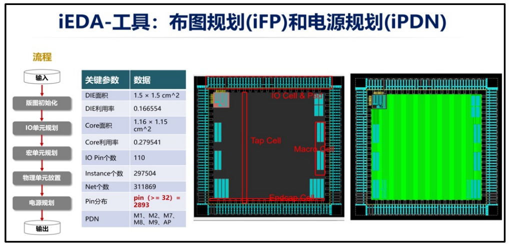
- RTL-netlist数据库和工具设计
  - RTL-Gtech：借助yosys
  - Gtech-netlist：parser，db，API
    - 工具：abc，iMap，EPFL工具
    - 算法：自己实现，合作接入
  - netlist：
    - 与iEDA目前的进行合并，做物理综合



iEDA-工具：工艺映射(iMAP)

- iMAP 1.0 完成了基本工艺映射算法以及逻辑优化算子
  - 逻辑综合算子（第五届EDA精英挑战赛赛题二官方出题工具，麒麟杯获奖赛题）
    - ✓ Rewrite (基于4-输入真值表的NPN匹配)
    - ✓ Refactor (基于SOP表达式的优化)
    - ✓ Balance (基于AND-tree的平衡算法)
    - ✓ LUT-opt (基于FPGA工艺映射的优化算法)
  - ✓ 工艺映射

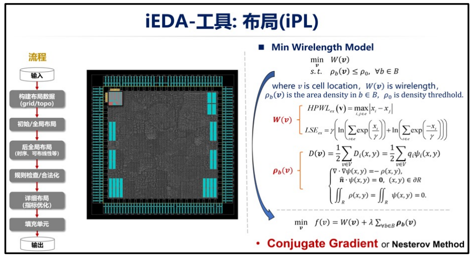
$z(t) = c(t)d'(t) + c'(t)z(t-1)$



iEDA-工具：布图规划(iFP)和电源规划(iPDN)

流程：输入 → 版图初始化 → IO单元规划 → 宏单元规划 → 物理单元放置 → 电源规划 → 输出

关键参数	数据
DIE面积	1.5 × 1.5 cm^2
DIE利用率	0.166554
Core面积	1.16 × 1.15 cm^2
Core利用率	0.279541
IO Pin个数	110
Instance个数	297504
Net个数	311869
Pin分布	pin（>= 32）= 2893
PDN	M1、M2、M7、M8、M9、AP



iEDA-工具: 布局(iPL)

流程：输入 → 构建布局数据(grid/topo) → 初始/全局布局 → 后全局布局（时序、可布线性等） → 规则检查/合法化 → 详细布局（指标优化） → 填充单元 → 输出

- Min Wirelength Model

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

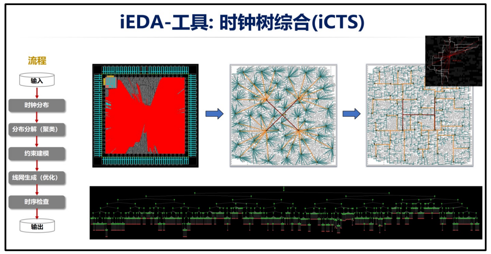
where v is cell location, $W(v)$ is wirelength, $\rho_b(v)$ is the area density in $b \in B$, ρ_0 is density thredhold.

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

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

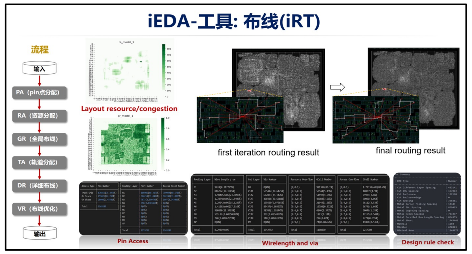
$$\min_{v} \; f(v) = W(v) + \lambda \sum_{\forall b \in B} \rho_b(v)$$

- Conjugate Gradient or Nesterov Method



iEDA-工具: 时钟树综合(iCTS)

流程：输入 → 时钟分布 → 分布分解（聚类） → 约束建模 → 线网生成（优化） → 时序检查 → 输出



iEDA-工具: 布线(iRT)

流程：输入 → PA（pin点分配） → RA（资源分配） → GR（全局布线） → TA（轨道分配） → DR（详细布线） → VR（布线优化） → 输出

Layout resource/congestion

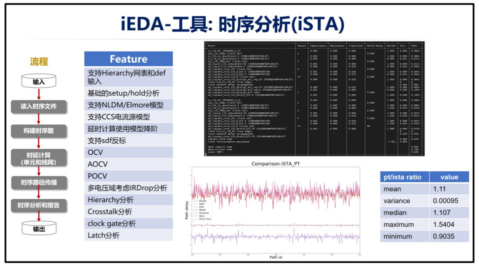
first iteration routing result

final routing result

Pin Access

Wirelength and via

Design rule check



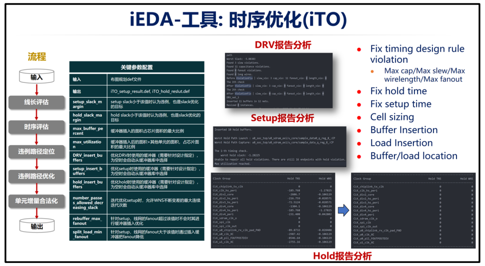
iEDA-工具: 时序分析(iSTA)

流程：输入 → 读入时序文件 → 构建时序图 → 时延计算（单元和线网） → 时序路径传播 → 时序分析和报告 → 输出

Feature
支持Hierarchy网表和def输入
基础的setup/hold分析
支持NLDM/Elmore模型
支持CCS电流源模型
延时计算使用模型降阶
支持sdf反标
OCV
AOCV
POCV
多电压域考虑IRDrop分析
Hierarchy分析
Crosstalk分析
clock gate分析
Latch分析

Comparison-iSTA_PT

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



iEDA-工具: 时序优化(iTO)

流程：输入 → 线长评估 → 时序评估 → 违例路径定位 → 违例路径优化 → 单元增量合法化 → 输出

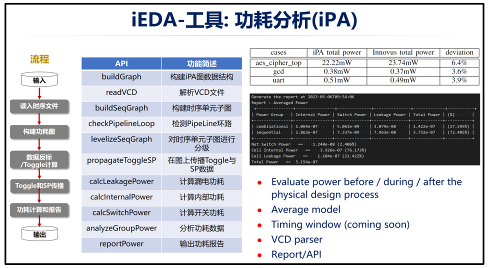
关键参数配置	
输入	布图规划def文件
输出	iTO_setup_result.def, iTO_hold_reslut.def
setup_slack_margin	setup slack小于该值时认为违例，也是slack优化的目标
hold_slack_margin	hold slack小于该值时认为违例，也是slack优化的目标
max_buffer_percent	缓冲器插入的面积占芯片面积的最大比例
max_utilization	缓冲器插入后的面积+其他单元的面积，占芯片面积的最大比例
DRV_insert_buffers	优化DRV时使用的缓冲器（需要针对设计指定），为空时会自动从缓冲器库中选择
setup_insert_buffers	优化setup时使用的缓冲器（需要针对设计指定），为空时会自动从缓冲器库中选择
hold_insert_buffers	优化hold时使用的缓冲器（需要针对设计指定），为空时会自动从缓冲器库中选择
number_passes_allowed_decreasing_slack	迭代优化setup时，允许WNS不断变差的最大连续迭代次数
rebuffer_max_fanout	针对setup，线网的fanout超过该值时不会对其进行缓冲器插入优化
split_load_min_fanout	针对setup，线网的fanout大于该值时通过插入缓冲器把fanout降低

DRV报告分析

Setup报告分析

Hold报告分析

- 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



iEDA-工具: 功耗分析(iPA)

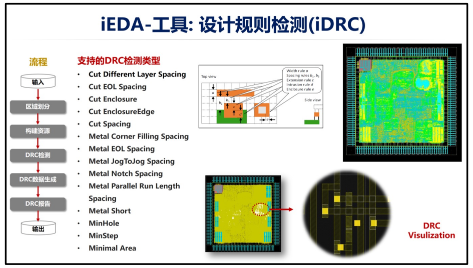
流程：输入 → 读入时序文件 → 构建功耗图 → 数据反标/Toggle计算 → Toggle和SP传播 → 功耗计算和报告 → 输出

API	功能简述
buildGraph	构建iPA图数据结构
readVCD	解析VCD文件
buildSeqGraph	构建时序单元子图
checkPipelineLoop	检测PipeLine环路
levelizeSeqGraph	对时序单元子图进行分级
propagateToggleSP	在图上传播Toggle与SP数据
calcLeakagePower	计算漏电功耗
calcInternalPower	计算内部功耗
calcSwitchPower	计算开关功耗
analyzeGroupPower	分析功耗数据
reportPower	输出功耗报告

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

```
Generate the report at 2023-05-06T09:54:06
Report : Averaged Power
+---------------+----------------+--------------+---------------+-------------+-----------+
| Power Group   | Internal Power | Switch Power | Leakage Power | Total Power | (%)       |
+---------------+----------------+--------------+---------------+-------------+-----------+
| combinational | 1.064e-07      | 5.063e-09    | 3.079e-08     | 1.422e-07   | (27.595%) |
| sequential    | 2.862e-07      | 7.337e-09    | 7.963e-08     | 3.732e-07   | (72.405%) |
+---------------+----------------+--------------+---------------+-------------+-----------+
Net Switch Power   ==   1.240e-08 (2.406%)
Cell Internal Power   ==   3.926e-07 (76.173%)
Cell Leakage Power   ==   1.104e-07 (21.422%)
Total Power  ==  5.154e-07
```

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



iEDA-工具: 设计规则检测(iDRC)

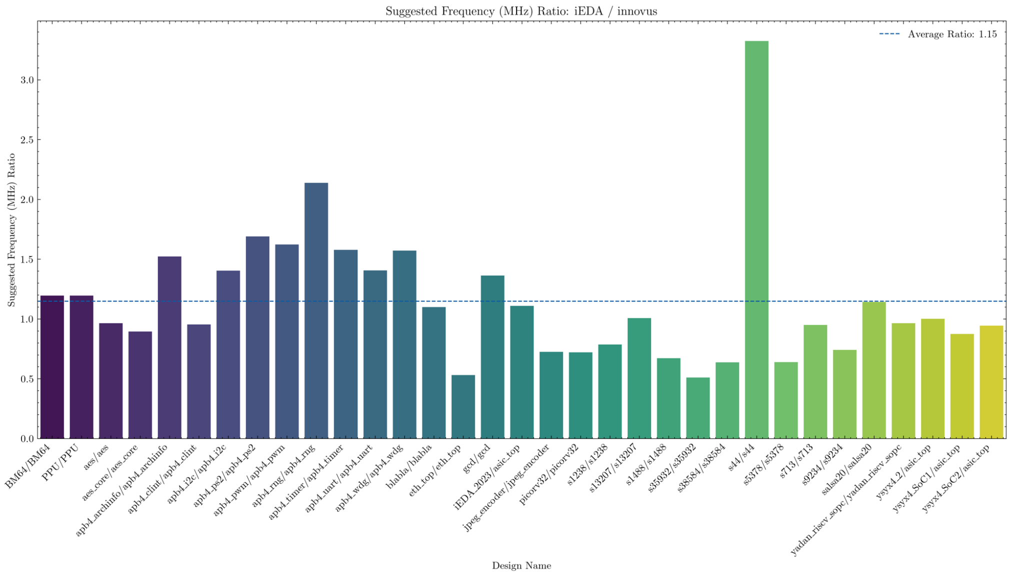
流程：输入 → 区域划分 → 构建资源 → DRC检测 → DRC数据生成 → DRC报告 → 输出

支持的DRC检测类型

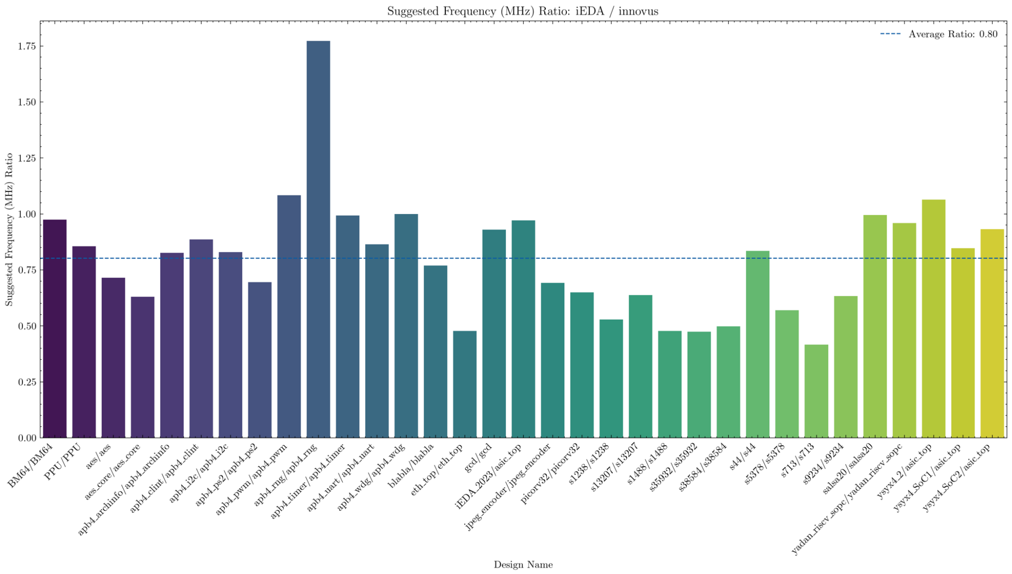
- Cut Different Layer Spacing
- Cut EOL Spacing
- Cut Enclosure
- Cut EnclosureEdge
- Cut Spacing
- Metal Corner Filling Spacing
- Metal EOL Spacing
- Metal JogToJog Spacing
- Metal Notch Spacing
- Metal Parallel Run Length Spacing
- Metal Short
- MinHole
- MinStep
- Minimal Area

DRC Visulization

iEDA & Innovus



Placement Suggested Frequency: iEDA/Innovus = 1.15

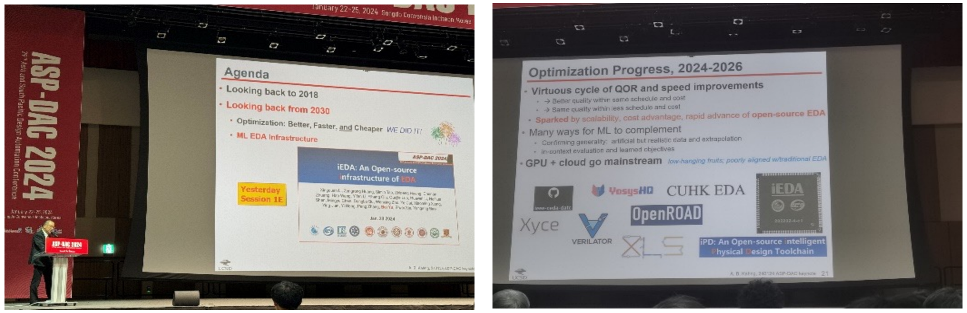


Routing Suggested Frequency: iEDA/Innovus = 0.8

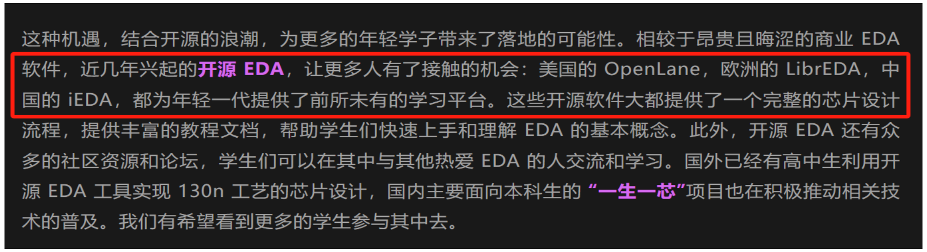
iEDA Code and Influence

- **Code**
  - GitHub: https://github.com/OSCC-Project/iEDA
  - Gitee: https://gitee.com/oscc-project/iEDA
  - AtomGit： https://atomgit.com/ieda/iEDA
  - OpenI： https://openi.pcl.ac.cn/OSCC/iEDA

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



Andrew B. Kahng (ACM/IEEE Fellow, a leading promoter of international open-source EDA) introduced iEDA in the keynote presentation at the EDA International Conference ASPDAC.



这种机遇，结合开源的浪潮，为更多的年轻学子带来了落地的可能性。相较于昂贵且晦涩的商业 EDA 软件，近几年兴起的**开源 EDA**，让更多人有了接触的机会：美国的 OpenLane，欧洲的 LibrEDA，中国的 iEDA，都为年轻一代提供了前所未有的学习平台。这些开源软件大都提供了一个完整的芯片设计流程，提供丰富的教程文档，帮助学生们快速上手和理解 EDA 的基本概念。此外，开源 EDA 还有众多的社区资源和论坛，学生们可以在其中与其他热爱 EDA 的人交流和学习。国外已经有高中生利用开源 EDA 工具实现 130n 工艺的芯片设计，国内主要面向本科生的**“一生一芯”**项目也在积极推动相关技术的普及。我们有希望看到更多的学生参与其中去。

- **Paper**
  - **iEDA:** An Open-Source Intelligent Physical Implementation Toolkit and Library", In Proc. ISEDA, 2023. **(Best Paper Award)**
  - **iEDA:** An Open-source infrastructure of EDA (invited), In Proc. ASPDAC, 2024.
  - **iPD:** An Open-source intelligent Physical Design Tool Chain (invited), In Proc. ASPDAC, 2024.
  - **iPL-3D:** A Novel Bilevel Programming Model for Die-to-Die Placement, ICCAD, 2023.
  - **iRT:** Net Resource Allocation: A Desirable Initial Routing Step, DAC, 2024
  - **iCTS:** Toward Controllable Hierarchical Clock Tree Synthesis with Skew-Latency-Load Tree, DAC, 2024

- **Open source iEDA platform and tools (including 40W lines of code)**
- **Obtained 500 stars, 110 folks, 75 people contributed code.**
- **Support the construction of EDA courses in 5 universities, and set problems for 4 EDA competitions.**
- **Supports research for over 10 domestic and 2 international EDA teams.**
- **The iEDA self-media video has received 44,000 views.**

01 **RTL-to-GDS (iEDA)**

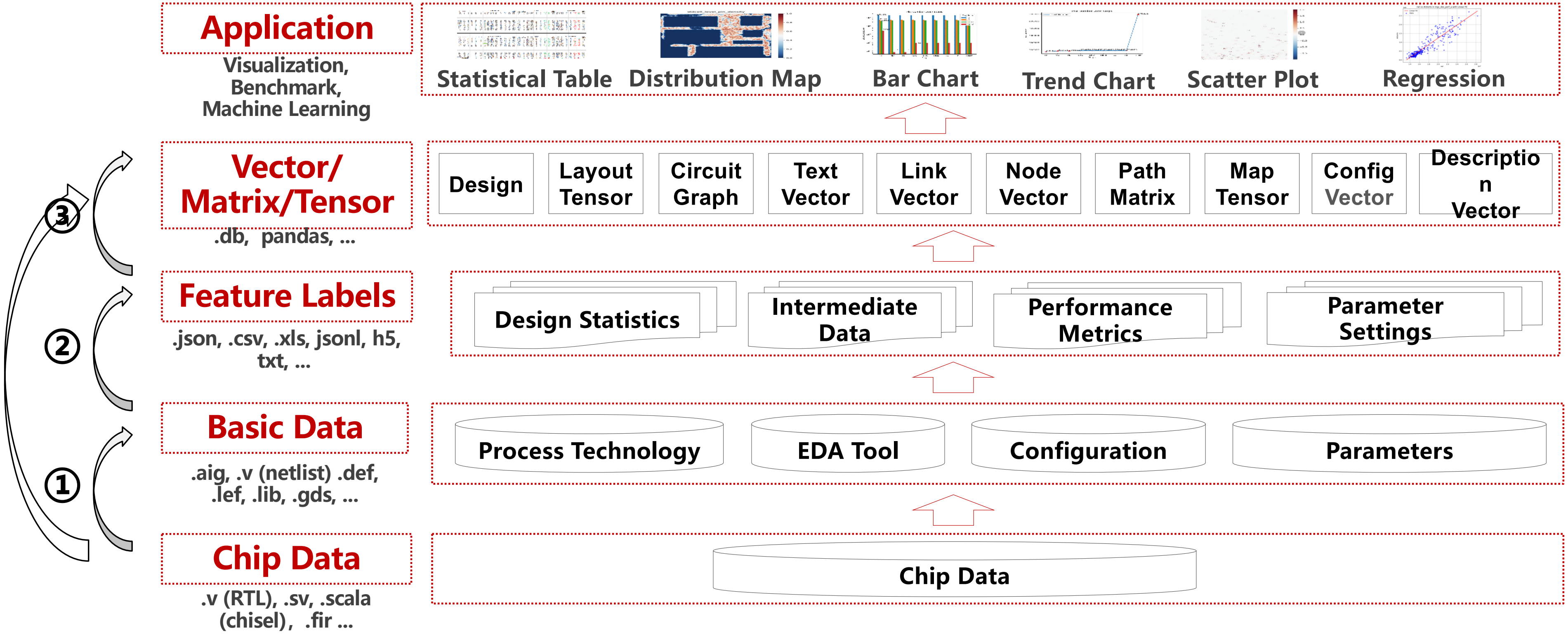
02 **RTL-to-Vector (AiEDA)**

03 **Vector-to-Vector**

04 **Vector-to-GDS**

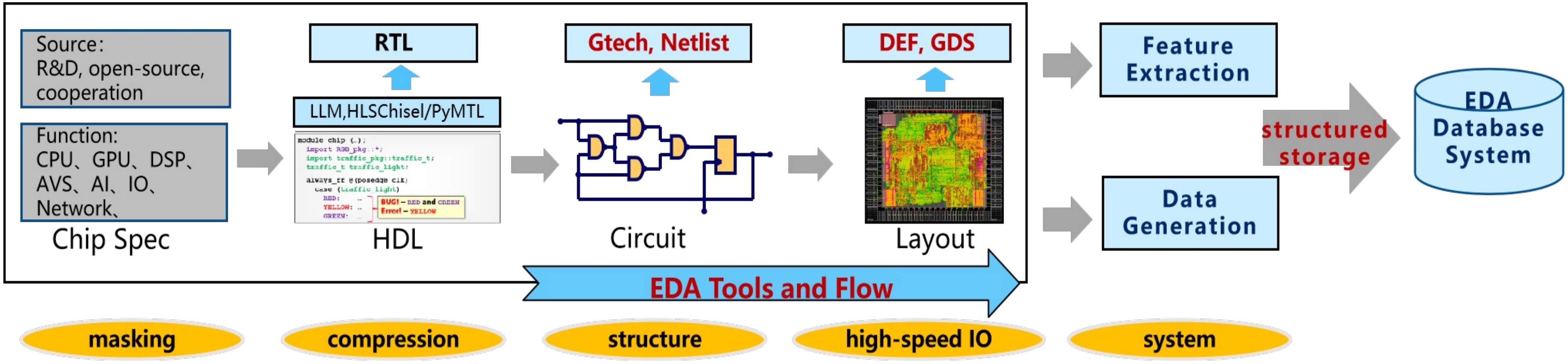
AiEDA：RTL-to-Vector

- **AiEDA：The first AI framework and toolbox in the EDA/IC field**
  - Supports RTL-to-Vector (RTL2Vec), converting chip data into vectorized data that can be input into AI models.



①: RTL-to-Gtech/Netlist/DEF/GDS

- **Using EDA tools to transform RTL data into standard files at various stages. (Gtech/Netlist/DEF/GDS)**
  - RTL-Gtech：Commerical tools：**DC**；Open Source Tools：**yosys**
  - Gtech-Netlist：Commerical tools：**DC**；Open Source Tools：**abc**，iMap，**EPFL**
  - Netlist-DEF/GDS：Commerical tools：**Innovus**；Open Source Tools：**iEDA**

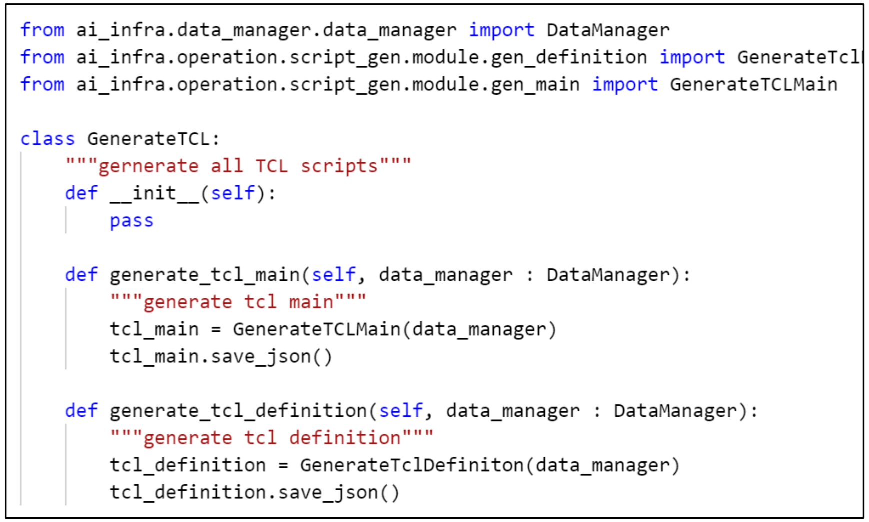


```
module chip (…);
  import RGB_pkg::*;
  import traffic_pkg::traffic_t;
  traffic_t traffic_light;

  always_ff @(posedge clk)
    case (traffic_light)
      RED:    …
      YELLOW: …
      GREEN:  …
```

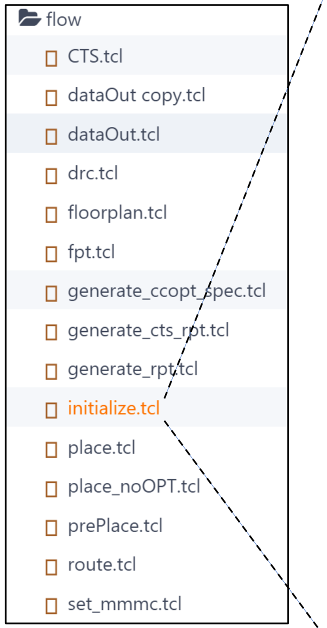
BUG! – RED and GREEN
Error! – YELLOW

② Report/Log/DB-to-jsonl/csv/image

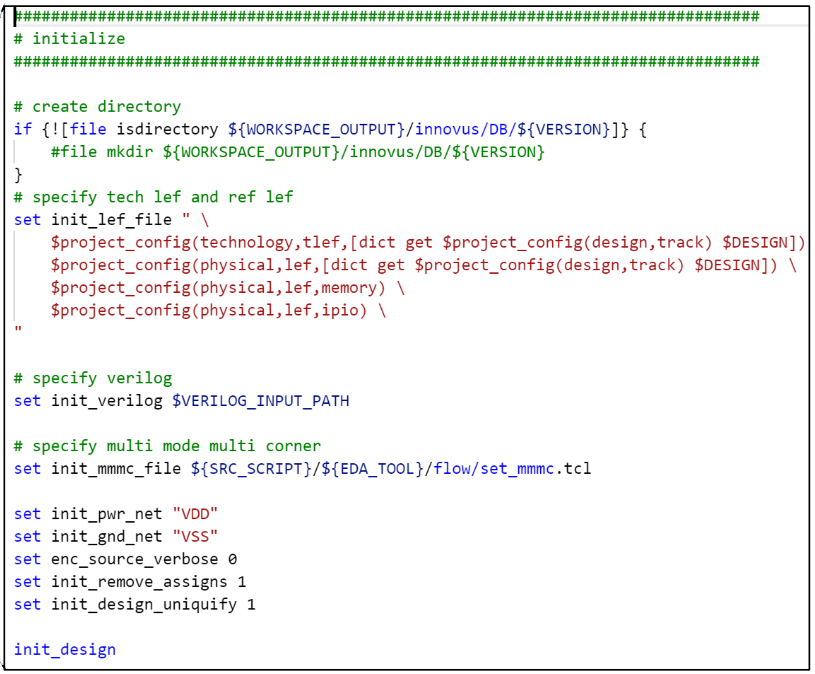
- **Run flow using commerical tool:** run by python->TCL
  - Intelligent generation of JSON configuration parameters
  - Generate TCL scripts for calling EDA tools based on configuration using Python.



```python
from ai_infra.data_manager.data_manager import DataManager
from ai_infra.operation.script_gen.module.gen_definition import GenerateTcl
from ai_infra.operation.script_gen.module.gen_main import GenerateTCLMain

class GenerateTCL:
    """gernerate all TCL scripts"""
    def __init__(self):
        pass

    def generate_tcl_main(self, data_manager : DataManager):
        """generate tcl main"""
        tcl_main = GenerateTCLMain(data_manager)
        tcl_main.save_json()

    def generate_tcl_definition(self, data_manager : DataManager):
        """generate tcl definition"""
        tcl_definition = GenerateTclDefiniton(data_manager)
        tcl_definition.save_json()
```



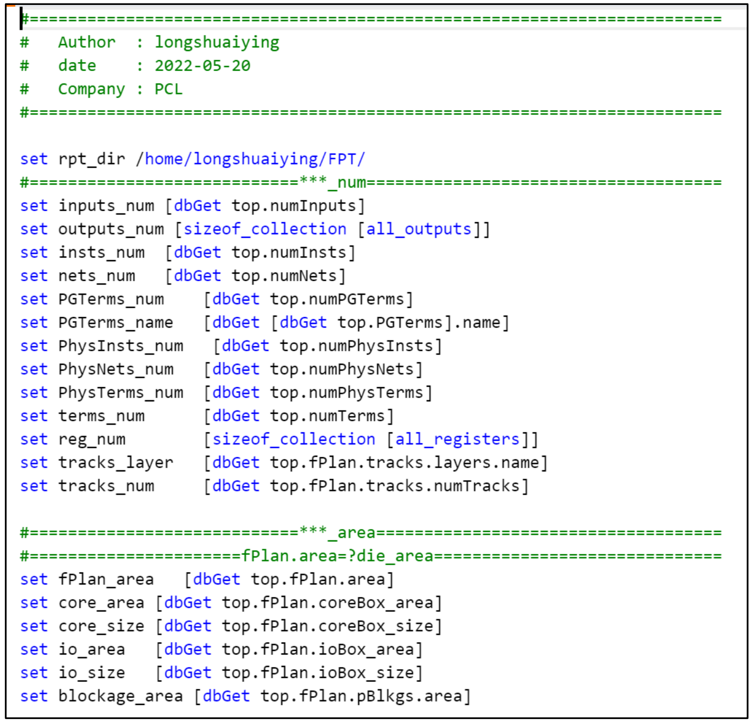
flow
- CTS.tcl
- dataOut copy.tcl
- dataOut.tcl
- drc.tcl
- floorplan.tcl
- fpt.tcl
- generate_ccopt_spec.tcl
- generate_cts_rpt.tcl
- generate_rpt.tcl
- initialize.tcl
- place.tcl
- place_noOPT.tcl
- prePlace.tcl
- route.tcl
- set_mmmc.tcl



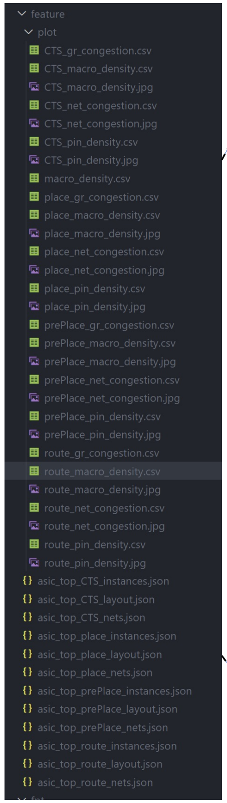
```tcl
################################################################################
# initialize
################################################################################

# create directory
if {![file isdirectory ${WORKSPACE_OUTPUT}/innovus/DB/${VERSION}]} {
    #file mkdir ${WORKSPACE_OUTPUT}/innovus/DB/${VERSION}
}
# specify tech lef and ref lef
set init_lef_file " \
    $project_config(technology,tlef,[dict get $project_config(design,track) $DESIGN])
    $project_config(physical,lef,[dict get $project_config(design,track) $DESIGN]) \
    $project_config(physical,lef,memory) \
    $project_config(physical,lef,ipio) \
"


# specify verilog
set init_verilog $VERILOG_INPUT_PATH

# specify multi mode multi corner
set init_mmmc_file ${SRC_SCRIPT}/${EDA_TOOL}/flow/set_mmmc.tcl

set init_pwr_net "VDD"
set init_gnd_net "VSS"
set enc_source_verbose 0
set init_remove_assigns 1
set init_design_uniquify 1

init_design
```

② Report/Log/DB-to-jsonl/csv/image

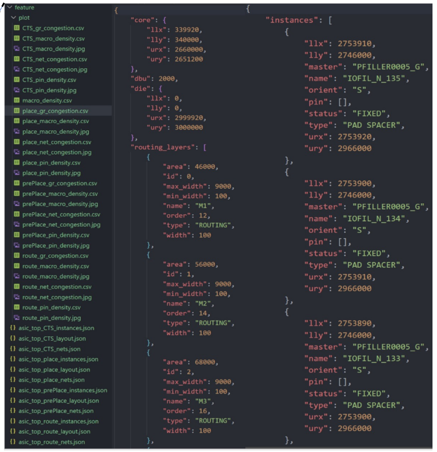
- **Extract Commerical Tool Design Data for Label:** Design an automated feature extraction tool to extract features and evaluation metrics at various stages from commercial tool reports, logs, and memory data, and save them as formatted data files to create a labeled dataset.



```
#=========================================================================
#   Author  : longshuaiying
#   date    : 2022-05-20
#   Company : PCL
#=========================================================================

set rpt_dir /home/longshuaiying/FPT/
#============================***_num=====================================
set inputs_num [dbGet top.numInputs]
set outputs_num [sizeof_collection [all_outputs]]
set insts_num  [dbGet top.numInsts]
set nets_num   [dbGet top.numNets]
set PGTerms_num    [dbGet top.numPGTerms]
set PGTerms_name   [dbGet [dbGet top.PGTerms].name]
set PhysInsts_num   [dbGet top.numPhysInsts]
set PhysNets_num   [dbGet top.numPhysNets]
set PhysTerms_num  [dbGet top.numPhysTerms]
set terms_num      [dbGet top.numTerms]
set reg_num        [sizeof_collection [all_registers]]
set tracks_layer   [dbGet top.fPlan.tracks.layers.name]
set tracks_num     [dbGet top.fPlan.tracks.numTracks]

#===========================***_area=====================================
#=====================fPlan.area=?die_area===============================
set fPlan_area    [dbGet top.fPlan.area]
set core_area [dbGet top.fPlan.coreBox_area]
set core_size [dbGet top.fPlan.coreBox_size]
set io_area   [dbGet top.fPlan.ioBox_area]
set io_size   [dbGet top.fPlan.ioBox_size]
set blockage_area [dbGet top.fPlan.pBlkgs.area]
```

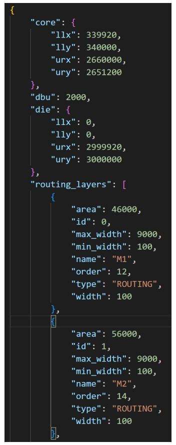


```
feature
  plot
    CTS_gr_congestion.csv
    CTS_macro_density.csv
    CTS_macro_density.jpg
    CTS_net_congestion.csv
    CTS_net_congestion.jpg
    CTS_pin_density.csv
    CTS_pin_density.jpg
    macro_density.csv
    place_gr_congestion.csv
    place_macro_density.csv
    place_macro_density.jpg
    place_net_congestion.csv
    place_net_congestion.jpg
    place_pin_density.csv
    place_pin_density.jpg
    prePlace_gr_congestion.csv
    prePlace_macro_density.csv
    prePlace_macro_density.jpg
    prePlace_net_congestion.csv
    prePlace_net_congestion.jpg
    prePlace_pin_density.csv
    prePlace_pin_density.jpg
    route_gr_congestion.csv
    route_macro_density.csv
    route_macro_density.jpg
    route_net_congestion.csv
    route_net_congestion.jpg
    route_pin_density.csv
    route_pin_density.jpg
  asic_top_CTS_instances.json
  asic_top_CTS_layout.json
  asic_top_CTS_nets.json
  asic_top_place_instances.json
  asic_top_place_layout.json
  asic_top_place_nets.json
  asic_top_prePlace_instances.json
  asic_top_prePlace_layout.json
  asic_top_prePlace_nets.json
  asic_top_route_instances.json
  asic_top_route_layout.json
  asic_top_route_nets.json
```

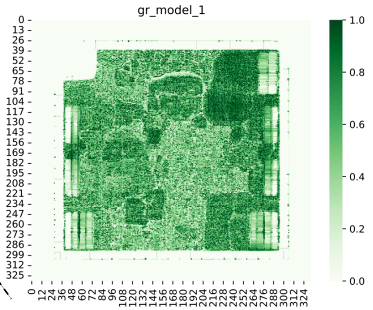


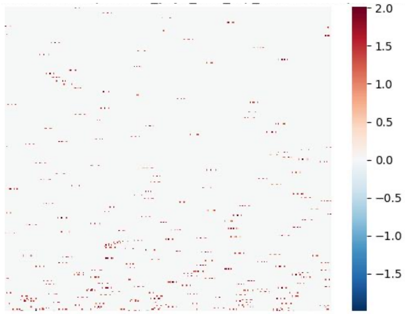
```
{
    "core": {
        "llx": 339920,
        "lly": 340000,
        "urx": 2660000,
        "ury": 2651200
    },
    "dbu": 2000,
    "die": {
        "llx": 0,
        "lly": 0,
        "urx": 2999920,
        "ury": 3000000
    },
    "routing_layers": [
        {
            "area": 46000,
            "id": 0,
            "max_width": 9000,
            "min_width": 100,
            "name": "M1",
            "order": 12,
            "type": "ROUTING",
            "width": 100
        },
        {
            "area": 56000,
            "id": 1,
            "max_width": 9000,
            "min_width": 100,
            "name": "M2",
            "order": 14,
            "type": "ROUTING",
            "width": 100
        },
        {
            "area": 68000,
            "id": 2,
            "max_width": 9000,
            "min_width": 100,
            "name": "M3",
            "order": 16,
            "type": "ROUTING",
            "width": 100
        },
```

```
{
    "instances": [
        {
            "llx": 2753910,
            "lly": 2746000,
            "master": "PFILLER0005_G",
            "name": "IOFIL_N_135",
            "orient": "S",
            "pin": [],
            "status": "FIXED",
            "type": "PAD SPACER",
            "urx": 2753920,
            "ury": 2966000
        },
        {
            "llx": 2753900,
            "lly": 2746000,
            "master": "PFILLER0005_G",
            "name": "IOFIL_N_134",
            "orient": "S",
            "pin": [],
            "status": "FIXED",
            "type": "PAD SPACER",
            "urx": 2753910,
            "ury": 2966000
        },
        {
            "llx": 2753890,
            "lly": 2746000,
            "master": "PFILLER0005_G",
            "name": "IOFIL_N_133",
            "orient": "S",
            "pin": [],
            "status": "FIXED",
            "type": "PAD SPACER",
            "urx": 2753900,
            "ury": 2966000
        },
```



```
{
    "core": {
        "llx": 339920,
        "lly": 340000,
        "urx": 2660000,
        "ury": 2651200
    },
    "dbu": 2000,
    "die": {
        "llx": 0,
        "lly": 0,
        "urx": 2999920,
        "ury": 3000000
    },
    "routing_layers": [
        {
            "area": 46000,
            "id": 0,
            "max_width": 9000,
            "min_width": 100,
            "name": "M1",
            "order": 12,
            "type": "ROUTING",
            "width": 100
        },
        {
            "area": 56000,
            "id": 1,
            "max_width": 9000,
            "min_width": 100,
            "name": "M2",
            "order": 14,
            "type": "ROUTING",
            "width": 100
        },
```



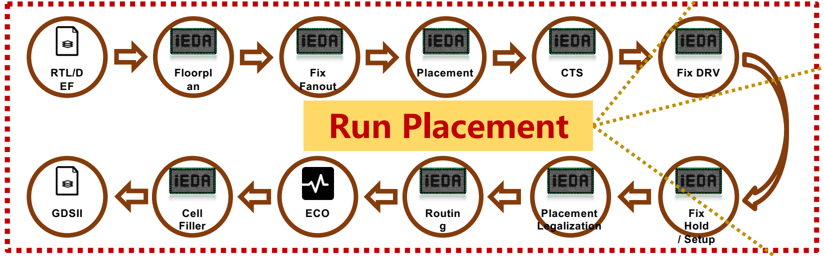


② API-to-jsonl/csv/image

- **Open source flow running and extraction：**

LogicFactory，iEDA

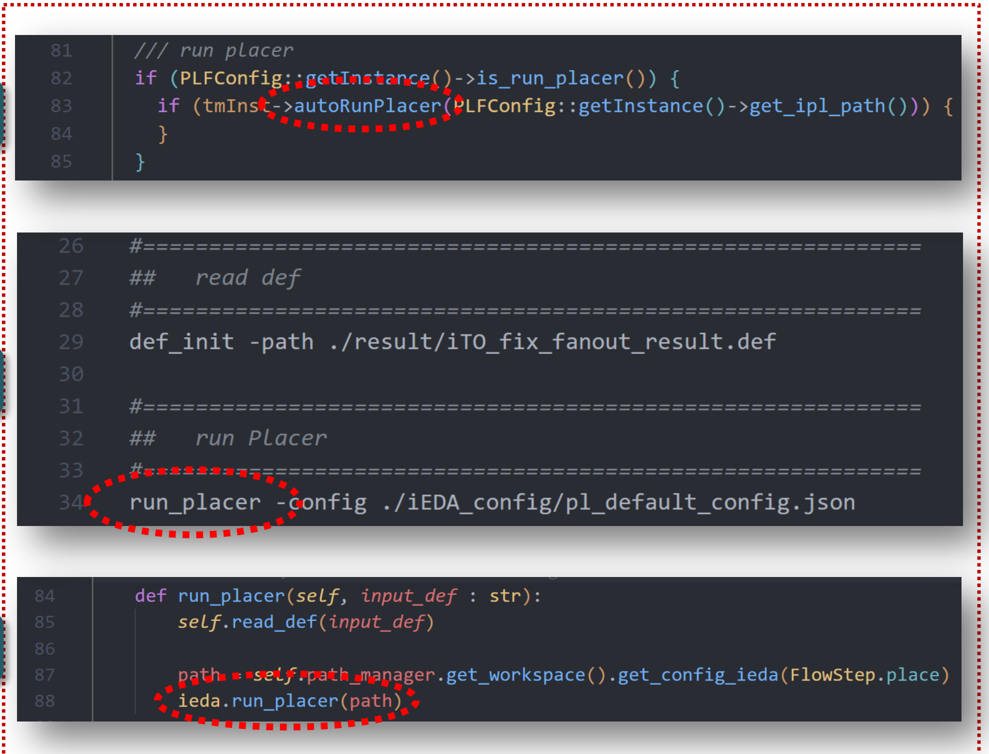
- Support **C++**、**RUST**、**TCL**、**Python**
- API Functionally complete, rich in interfaces.



C++ API

TCL API

Python API



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

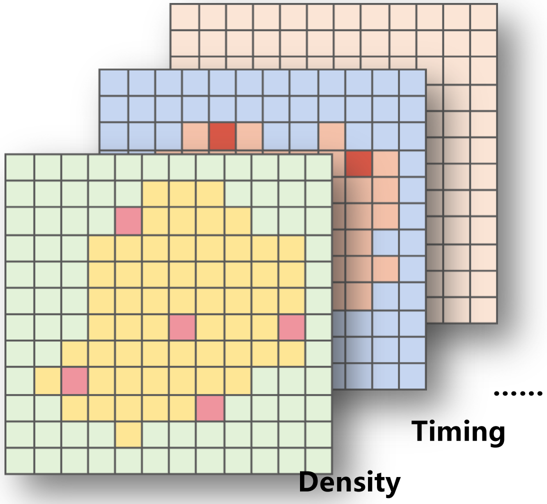
```tcl
#============================================================
##   read def
#============================================================
def_init -path ./result/iTO_fix_fanout_result.def

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

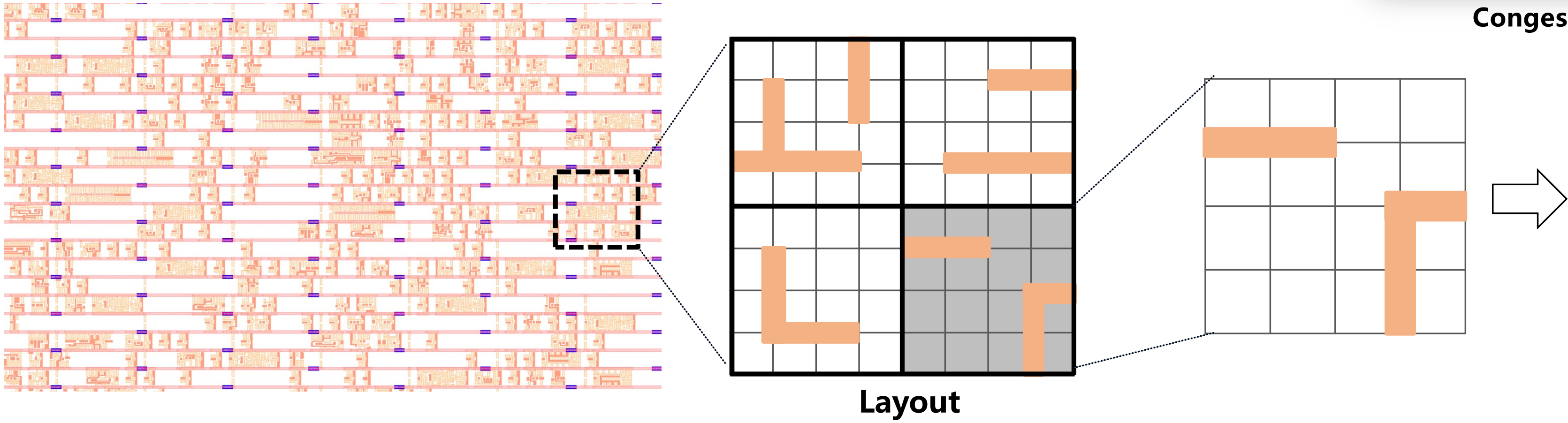
```python
def run_placer(self, input_def : str):
    self.read_def(input_def)

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

iEDA、AiEDA 实际项目工程代码

③ DEF-to-Vector

- Convert the map and indicator Map into matrix or tensor format.
  - Multi-channel representation: Multi-layer layout (Inst, Pin, OBS, Cut, Metal), cell density, pin density, congestion, DRC, timing distribution graph, power distribution graph.
  - Netlist, cell, metal wire, represented individually.
- Sparse storage



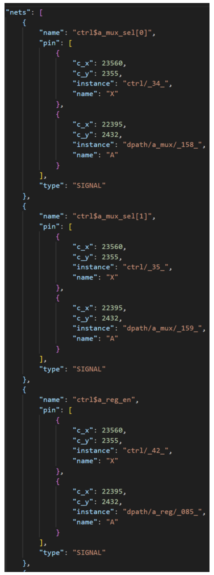


$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

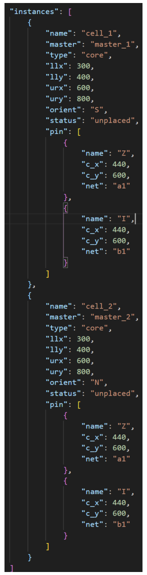
iBM：Vector Dataset

EDA Label Dataset and its Generation Framework

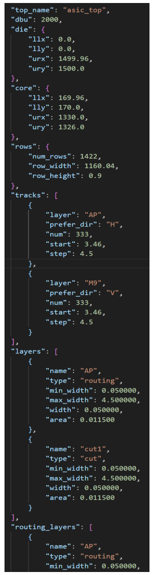
- **iBM：** Tagged chip dataset, completed 60 RTLs, 120 Netlists (thousands to tens of millions of gates), 500 DEF/GDS files;



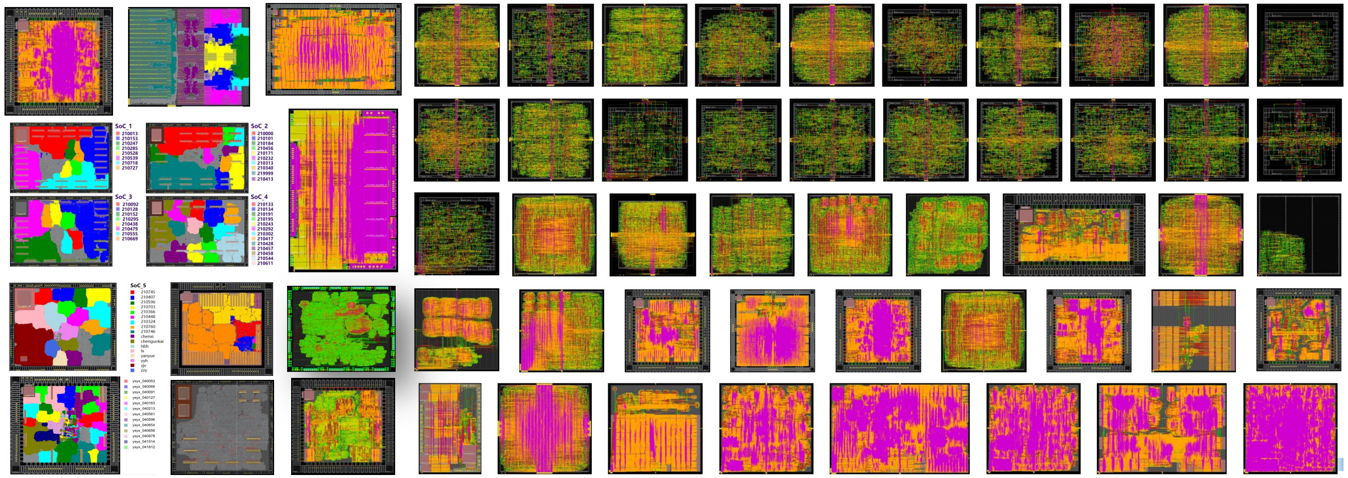
```
"nets": [
    {
        "name": "ctrl$a_mux_sel[0]",
        "pin": [
            {
                "c_x": 23560,
                "c_y": 2355,
                "instance": "ctrl/_34_",
                "name": "X"
            },
            {
                "c_x": 22395,
                "c_y": 2432,
                "instance": "dpath/a_mux/_158_",
                "name": "A"
            }
        ],
        "type": "SIGNAL"
    },
    {
        "name": "ctrl$a_mux_sel[1]",
        "pin": [
            {
                "c_x": 23560,
                "c_y": 2355,
                "instance": "ctrl/_35_",
                "name": "X"
            },
            {
                "c_x": 22395,
                "c_y": 2432,
                "instance": "dpath/a_mux/_159_",
                "name": "A"
            }
        ],
        "type": "SIGNAL"
    },
    {
        "name": "ctrl$a_reg_en",
        "pin": [
            {
                "c_x": 23560,
                "c_y": 2355,
                "instance": "ctrl/_42_",
                "name": "X"
            },
            {
                "c_x": 22395,
                "c_y": 2432,
                "instance": "dpath/a_reg/_085_",
                "name": "A"
            }
        ],
        "type": "SIGNAL"
    },
```



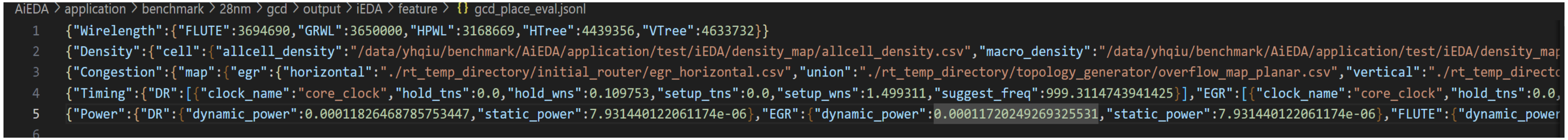
```
"instances": [
    {
        "name": "cell_1",
        "master": "master_1",
        "type": "core",
        "llx": 300,
        "lly": 400,
        "urx": 600,
        "ury": 800,
        "orient": "S",
        "status": "unplaced",
        "pin": [
            {
                "name": "Z",
                "c_x": 440,
                "c_y": 600,
                "net": "a1"
            },
            {
                "name": "I",
                "c_x": 440,
                "c_y": 600,
                "net": "b1"
            }
        ]
    },
    {
        "name": "cell_2",
        "master": "master_2",
        "type": "core",
        "llx": 300,
        "lly": 400,
        "urx": 600,
        "ury": 800,
        "orient": "N",
        "status": "unplaced",
        "pin": [
            {
                "name": "Z",
                "c_x": 440,
                "c_y": 600,
                "net": "a1"
            },
            {
                "name": "I",
                "c_x": 440,
                "c_y": 600,
                "net": "b1"
            }
        ]
    }
]
```



```
"top_name": "asic_top",
"dbu": 2000,
"die": {
    "llx": 0.0,
    "lly": 0.0,
    "urx": 1499.96,
    "ury": 1500.0
},
"core": {
    "llx": 169.96,
    "lly": 170.0,
    "urx": 1330.0,
    "ury": 1326.0
},
"rows": {
    "num_rows": 1422,
    "row_width": 1160.04,
    "row_height": 0.9
},
"tracks": [
    {
        "layer": "AP",
        "prefer_dir": "H",
        "num": 333,
        "start": 3.46,
        "step": 4.5
    },
    {
        "layer": "M9",
        "prefer_dir": "V",
        "num": 333,
        "start": 3.46,
        "step": 4.5
    }
],
"layers": [
    {
        "name": "AP",
        "type": "routing",
        "min_width": 0.050000,
        "max_width": 4.500000,
        "width": 0.050000,
        "area": 0.011500
    },
    {
        "name": "cut1",
        "type": "cut",
        "min_width": 0.050000,
        "max_width": 4.500000,
        "width": 0.050000,
        "area": 0.011500
    }
],
"routing_layers": [
    {
        "name": "AP",
        "type": "routing",
        "min_width": 0.050000,
```



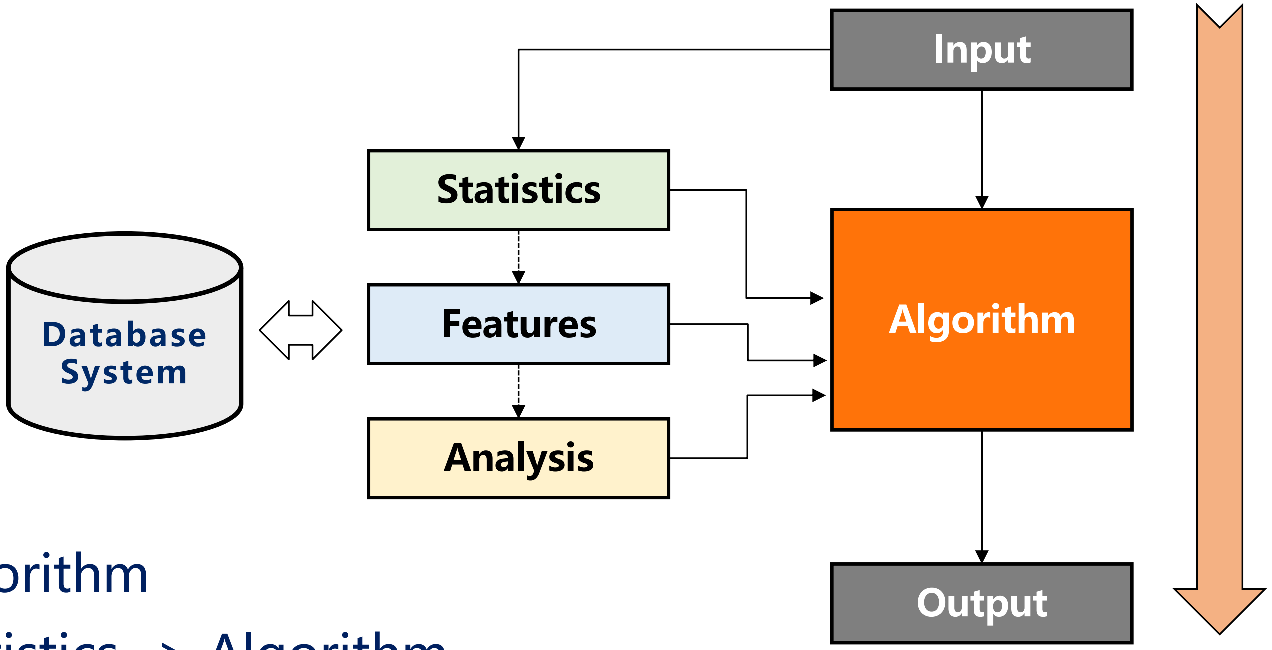
60 chip layout designs



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

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

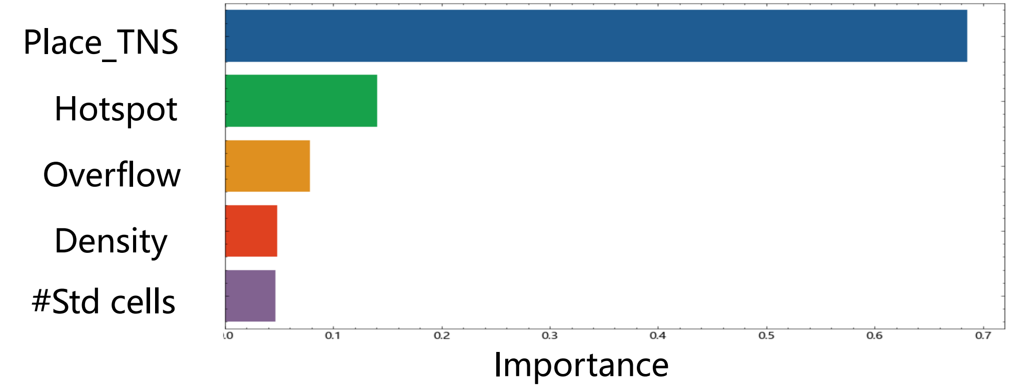
Machine Learning (ML)-aided Design



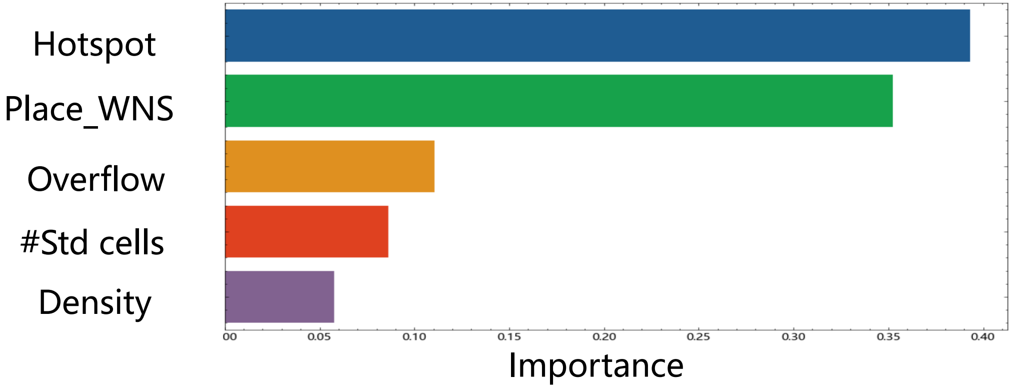
- **L0:** Algorithm
- **L1:** Statistics -> Algorithm
- **L2:** Statistics -> Feature -> Algorithm
- **L3:** **Statistics -> Feature -> Analysis** -> Algorithm

Machine Learning: Correlation Analysis

- TNS vs. Place_TNS, Overflow, Density, Congestion hospot, #std Cells



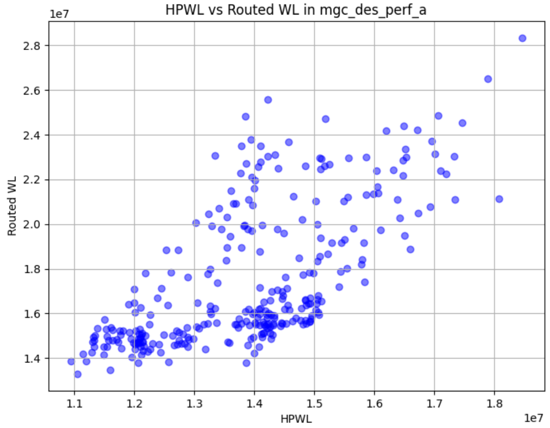
- WNS vs. Place_TNS, Overflow, Density, Congestion hospot, #std Cells

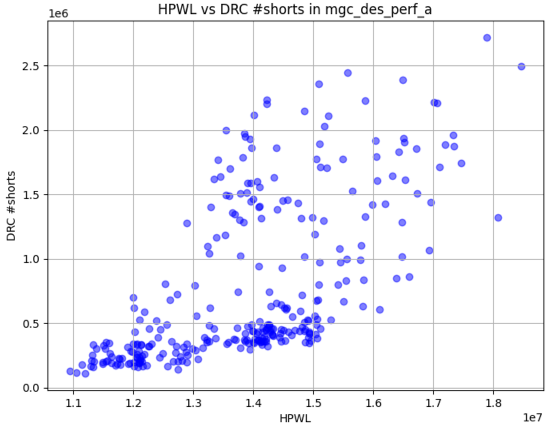


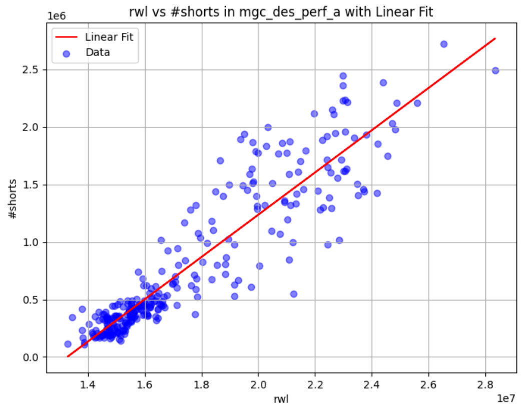
Machine Learning: Regression

- Linear Regression (**HPWL, routed_WL (rWL) vs. #shorts**)
  - It is observed that the scatter plot of HPWL to routed WL is similar to that of HPWL to #shorts, speculating that routed WL and #shorts have a strong correlation
  - by inferring routed WL from HPWL, combined with the linear model of routed WL and #shorts that has been fitted, we can understand the influence of HPWL on #shorts.

Features	Pearson	Spearman
HWPL vs Routed WL	0.6621	0.6828
HPWL vs #shorts	0.6250	0.6983
Routed_WL vs #shorts	**0.9195**	**0.9215**



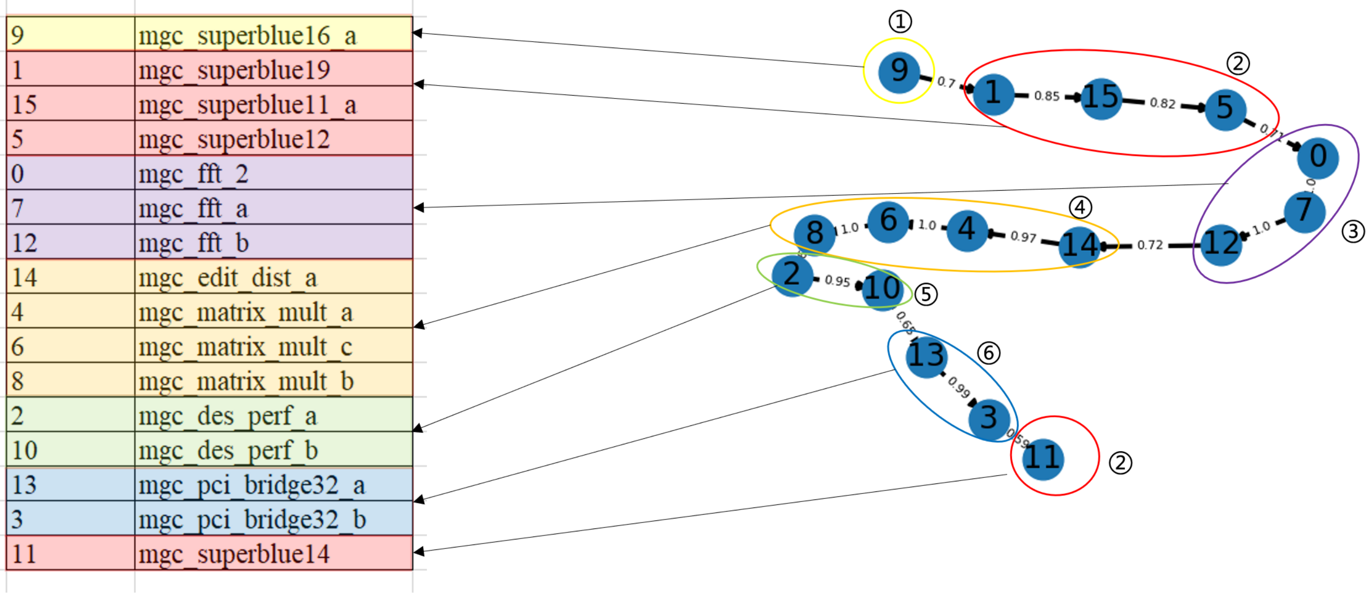




$$\#shorts = 0.184 * rWL - 2443844$$

Machine Learning: Chip Design Classification

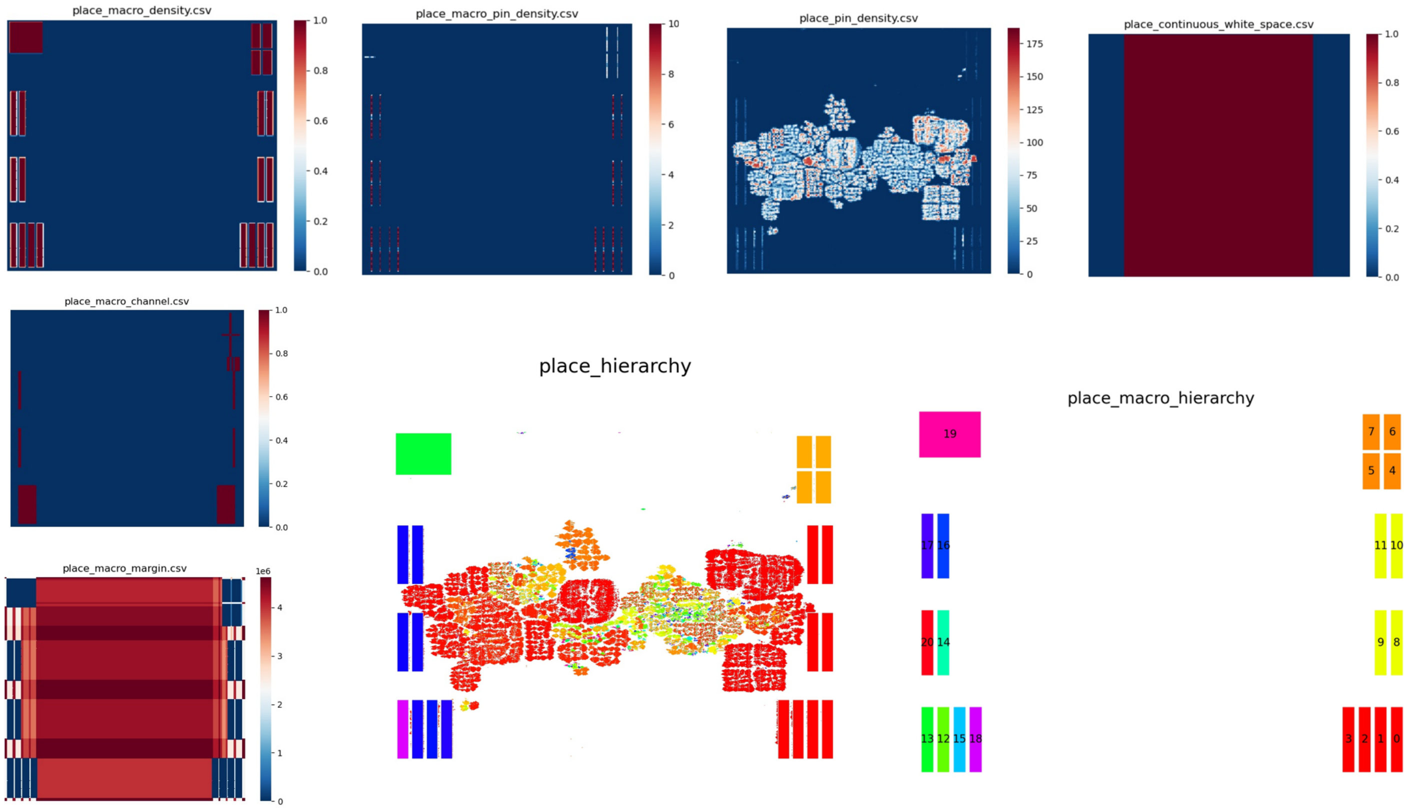
- Given a serials of chip designs, ML can be used to divide them as several groups



9	mgc_superblue16_a
1	mgc_superblue19
15	mgc_superblue11_a
5	mgc_superblue12
0	mgc_fft_2
7	mgc_fft_a
12	mgc_fft_b
14	mgc_edit_dist_a
4	mgc_matrix_mult_a
6	mgc_matrix_mult_c
8	mgc_matrix_mult_b
2	mgc_des_perf_a
10	mgc_des_perf_b
13	mgc_pci_bridge32_a
3	mgc_pci_bridge32_b
11	mgc_superblue14

Machine Learning: Cell Classification

- For many back-end tasks, we need cluster cell instances as different parts. User can set cluster number, AiEDA resturn cluster result



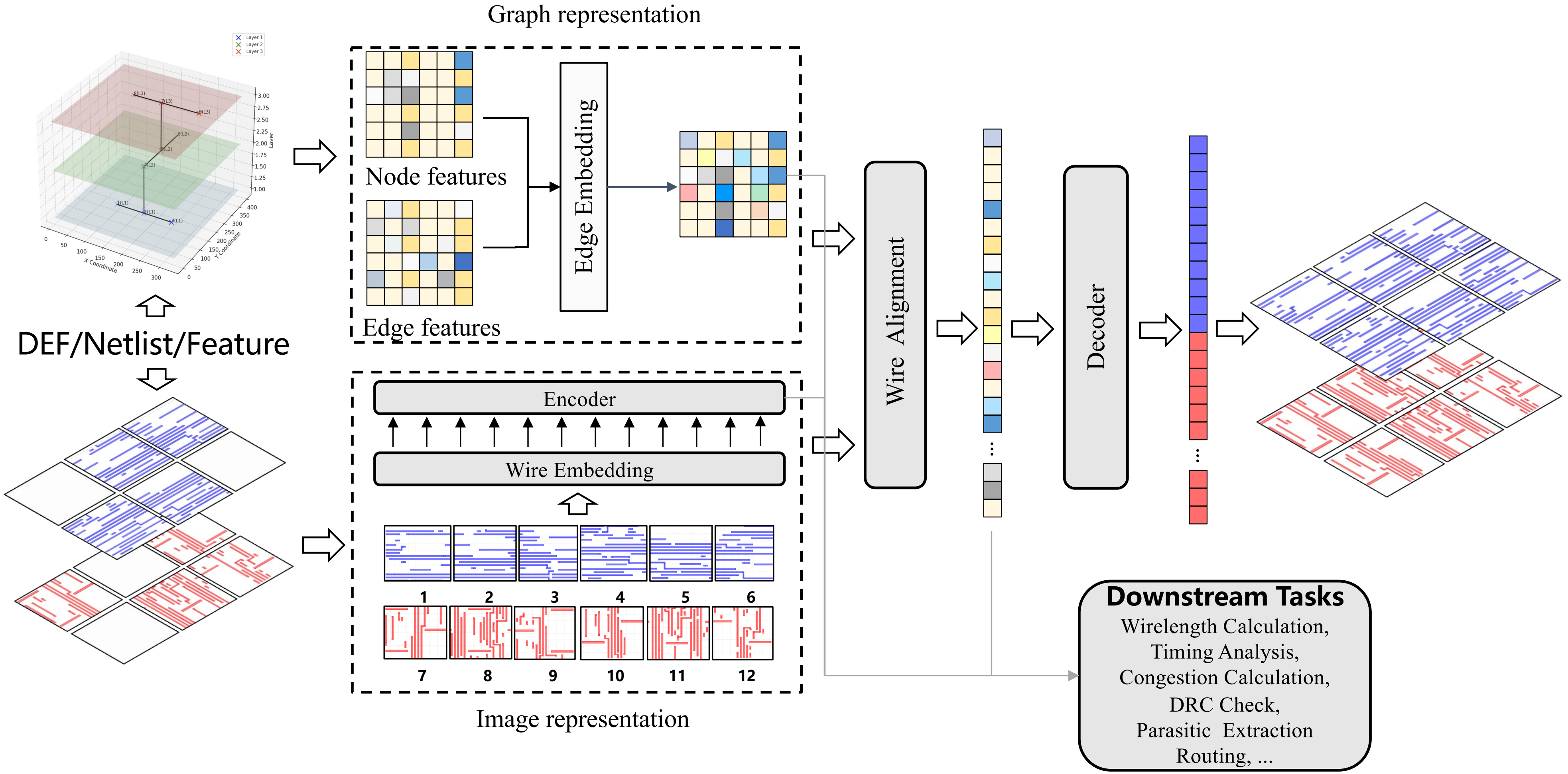
01 RTL-to-GDS (iEDA)

02 RTL-to-Vector (AiEDA)

03 Vector-to-Vector

04 Vector-to-GDS

Representation



Graph representation

Node features

Edge features

Edge Embedding

DEF/Netlist/Feature

Encoder

Wire Embedding

Image representation

Wire Alignment

Decoder

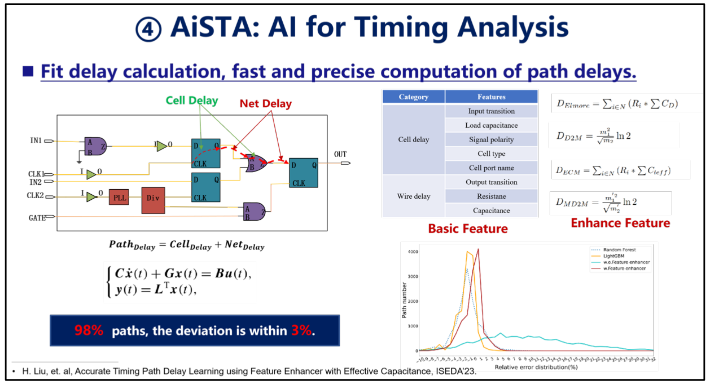
Downstream Tasks

Wirelength Calculation,
Timing Analysis,
Congestion Calculation,
DRC Check,
Parasitic Extraction
Routing, ...

Downstream AI4EDA Tasks

Key AI models in the EDA field

- Breakthroughs have been achieved in timing and power analysis and prediction, extraction of 3D parasitic capacitance and resistance parameters, process library mapping algorithms, timing optimization techniques, and Steiner tree generation. Relevant achievements have been published at top EDA conferences such as DAC, ICCAD, and ICCD.



④ AiSTA: AI for Timing Analysis

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

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

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

98% paths, the deviation is within 3%.

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

Basic Feature

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

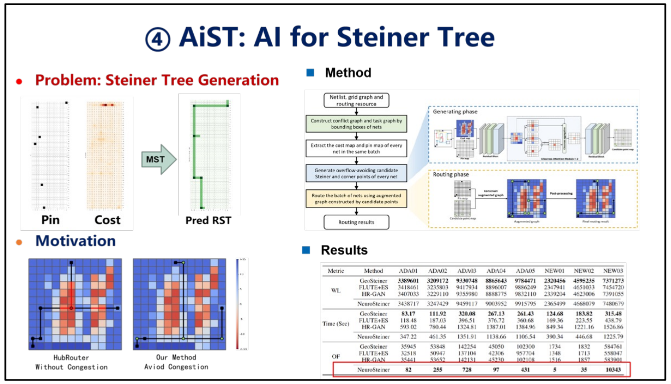
$$D_{D2M} = \frac{m_1^2}{\sqrt{m_2}}\ln 2$$

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

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

Enhance Feature

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



④ AiST: AI for Steiner Tree

- Problem: Steiner Tree Generation

- Motivation

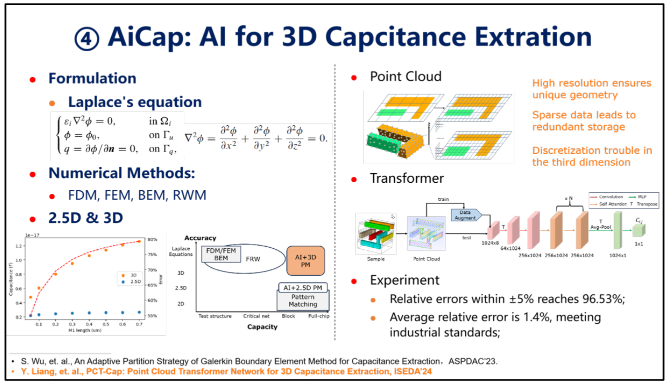
HubRouter
Without Congestion
Our Method
Aviod Congestion

■ Method

Netlist, grid graph and routing resource
Construct conflict graph and task graph by bounding boxes of nets
Extract the cost map and pin map of every net in the same batch
Generate overflow-avoiding candidate Steiner and corner points of every net
Route the batch of nets using augmented graph constructed by candidate points
Routing results
Generating phase
Routing phase
Post-processing

■ Results

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



④ AiCap: AI for 3D Capcitance Extration

- Formulation
  - Laplace's equation

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

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

- Point Cloud

High resolution ensures unique geometry

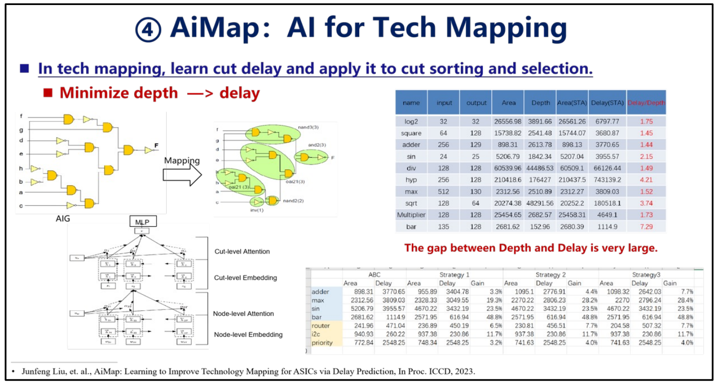
Sparse data leads to redundant storage

Discretization trouble in the third dimension

- Transformer

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

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



④ AiMap：AI for Tech Mapping

■ In tech mapping, learn cut delay and apply it to cut sorting and selection.

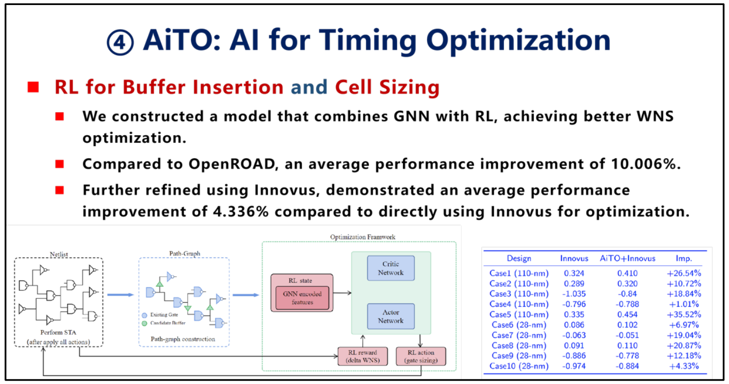
■ Minimize depth —> delay

name	input	output	Area	Depth	Area(STA)	Delay(STA)	Delay/Depth
log2	32	32	26556.98	3891.66	26561.26	6797.77	1.75
square	64	128	15738.82	2541.48	15744.07	3680.87	1.45
adder	256	129	898.31	2613.78	898.13	3770.65	1.44
sin	24	25	5206.79	1842.34	5207.04	3955.57	2.15
div	128	128	60539.96	44486.53	60509.1	66126.44	1.49
hyp	256	128	210418.6	176427	210437.5	743139.2	4.21
max	512	130	2312.56	2510.89	2312.27	3809.03	1.52
sqrt	128	64	20274.38	48291.56	20252.2	180518.1	3.74
Multiplier	128	128	25454.65	2682.57	25458.31	4649.1	1.73
bar	135	128	2681.62	152.96	2680.39	1114.9	7.29

The gap between Depth and Delay is very large.

	ABC		Strategy 1			Strategy 2			Strategy3		
	Area	Delay	Area	Delay	Gain	Area	Delay	Gain	Area	Delay	Gain
adder	898.31	3770.65	955.89	3404.78	3.3%	1095.1	2776.91	4.4%	1098.32	2642.03	7.7%
max	2312.56	3809.03	2328.33	3049.55	19.3%	2270.22	2806.23	28.2%	2270	2796.24	28.4%
sin	5206.79	3955.57	4670.22	3432.19	23.5%	4670.22	3432.19	23.5%	4670.22	3432.19	23.5%
bar	2681.62	1114.9	2571.95	616.94	48.8%	2571.95	616.94	48.8%	2571.95	616.94	48.8%
router	241.96	471.04	236.89	450.19	6.5%	230.81	456.51	7.7%	204.58	507.32	7.7%
i2c	940.93	260.22	937.38	230.86	11.7%	937.38	230.86	11.7%	937.38	230.86	11.7%
priority	772.84	2548.25	748.34	2548.25	3.2%	741.63	2548.25	4.0%	741.63	2548.25	4.0%

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



④ AiTO: AI for Timing Optimization

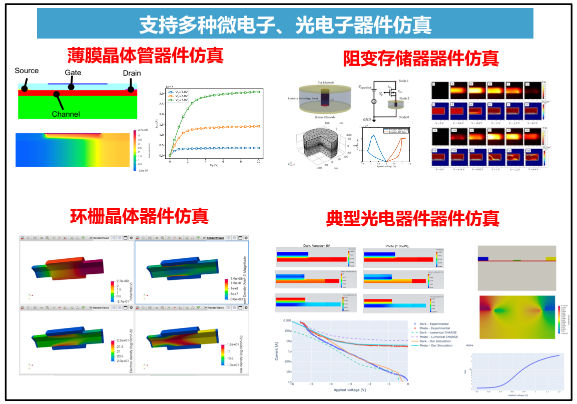
■ RL for Buffer Insertion and Cell Sizing

■ We constructed a model that combines GNN with RL, achieving better WNS optimization.

■ Compared to OpenROAD, an average performance improvement of 10.006%.

■ Further refined using Innovus, demonstrated an average performance improvement of 4.336% compared to directly using Innovus for optimization.

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



支持多种微电子、光电子器件仿真

薄膜晶体管器件仿真

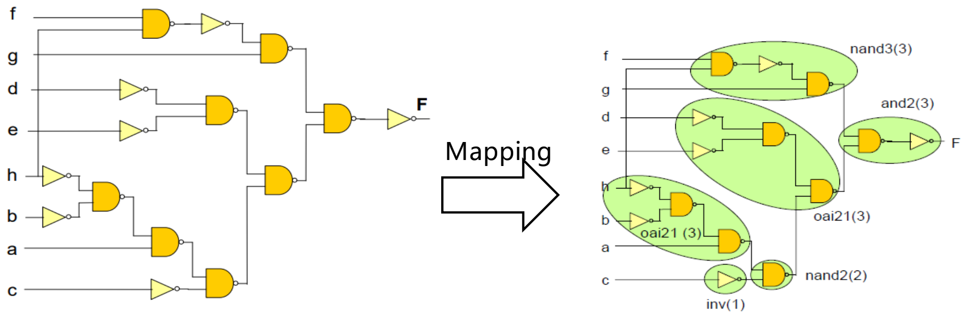
阻变存储器器件仿真

环栅晶体器件仿真

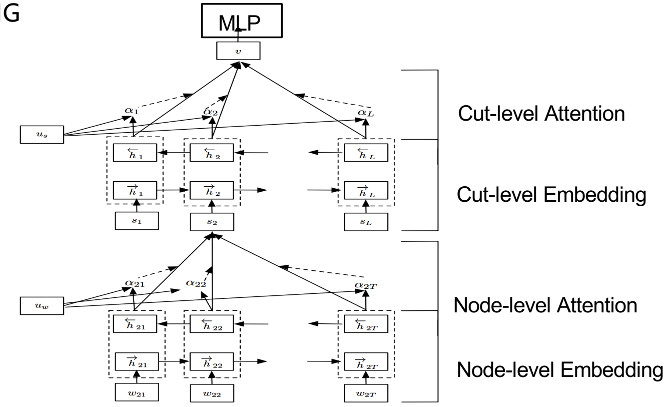
典型光电器件器件仿真

AI EDA：AiMap

- **AI for Tech Mapping, learn cut delay and apply it to cut sorting and selection.**
  - **Minimize depth —> delay**

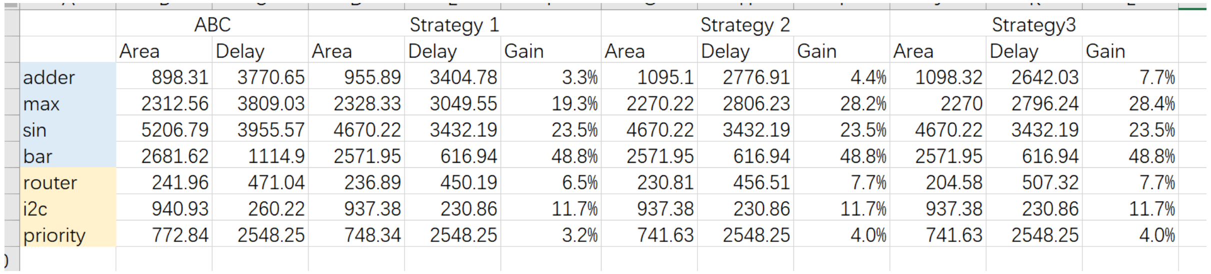


AIG



name	input	output	Area	Depth	Area(STA)	Delay(STA)	Delay/Depth
log2	32	32	26556.98	3891.66	26561.26	6797.77	1.75
square	64	128	15738.82	2541.48	15744.07	3680.87	1.45
adder	256	129	898.31	2613.78	898.13	3770.65	1.44
sin	24	25	5206.79	1842.34	5207.04	3955.57	2.15
div	128	128	60539.96	44486.53	60509.1	66126.44	1.49
hyp	256	128	210418.6	176427	210437.5	743139.2	4.21
max	512	130	2312.56	2510.89	2312.27	3809.03	1.52
sqrt	128	64	20274.38	48291.56	20252.2	180518.1	3.74
Multiplier	128	128	25454.65	2682.57	25458.31	4649.1	1.73
bar	135	128	2681.62	152.96	2680.39	1114.9	7.29

The gap between Depth and Delay is very large.



	ABC		Strategy 1			Strategy 2			Strategy3		
	Area	Delay	Area	Delay	Gain	Area	Delay	Gain	Area	Delay	Gain
adder	898.31	3770.65	955.89	3404.78	3.3%	1095.1	2776.91	4.4%	1098.32	2642.03	7.7%
max	2312.56	3809.03	2328.33	3049.55	19.3%	2270.22	2806.23	28.2%	2270	2796.24	28.4%
sin	5206.79	3955.57	4670.22	3432.19	23.5%	4670.22	3432.19	23.5%	4670.22	3432.19	23.5%
bar	2681.62	1114.9	2571.95	616.94	48.8%	2571.95	616.94	48.8%	2571.95	616.94	48.8%
router	241.96	471.04	236.89	450.19	6.5%	230.81	456.51	7.7%	204.58	507.32	7.7%
i2c	940.93	260.22	937.38	230.86	11.7%	937.38	230.86	11.7%	937.38	230.86	11.7%
priority	772.84	2548.25	748.34	2548.25	3.2%	741.63	2548.25	4.0%	741.63	2548.25	4.0%

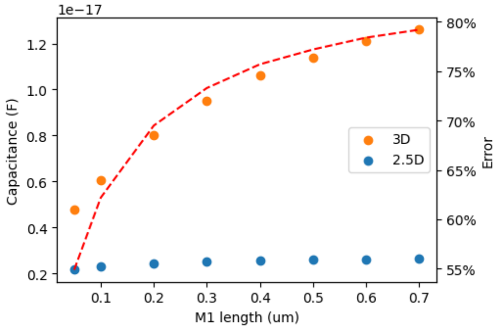
- Junfeng Liu, et. al., AiMap: Learning to Improve Technology Mapping for ASICs via Delay Prediction, In Proc. ICCD, 2023.

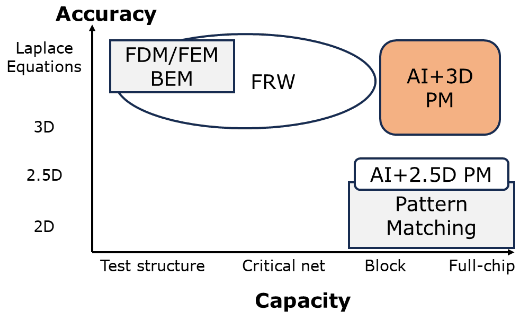
AI EDA: AiCap

- **AI for 3D Capcitance Extration**
  - **Laplace's equation**

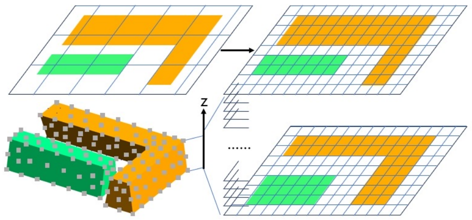
$$\begin{cases} \varepsilon_i \nabla^2 \phi = 0, & \text{in } \Omega_i \\ \phi = \phi_0, & \text{on } \Gamma_u \\ q = \partial\phi/\partial\boldsymbol{n} = 0, & \text{on } \Gamma_q, \end{cases} \qquad \nabla^2\phi = \frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} + \frac{\partial^2\phi}{\partial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

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





- Point Cloud

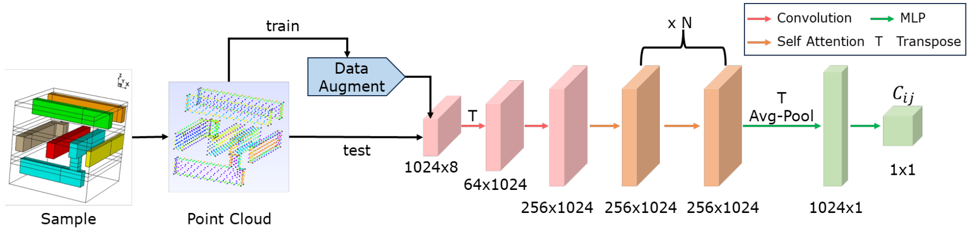


High resolution ensures unique geometry

Sparse data leads to redundant storage

Discretization trouble in the third dimension

- Transformer

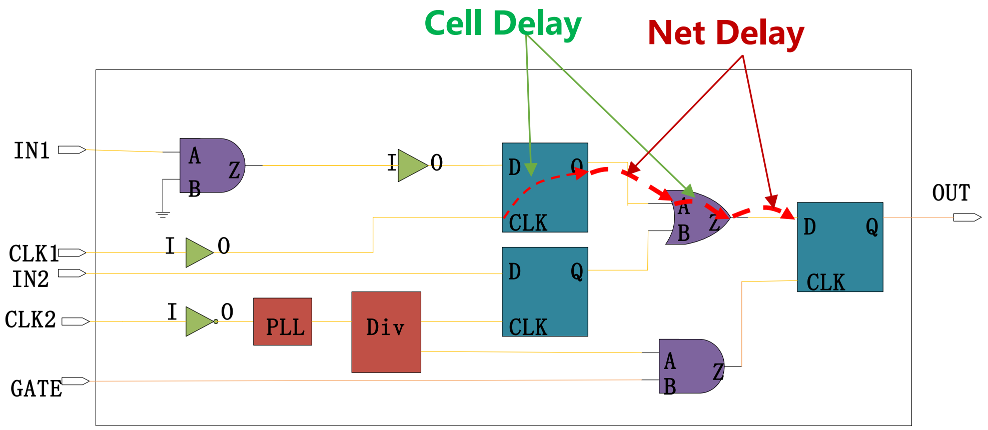


- Experiment
  - Relative errors within ±5% reaches 96.53%;
  - Average relative error is 1.4%, meeting industrial standards;

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

AI EDA: AiSTA

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



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

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

98% paths, the deviation is within 3%.

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

Basic Feature

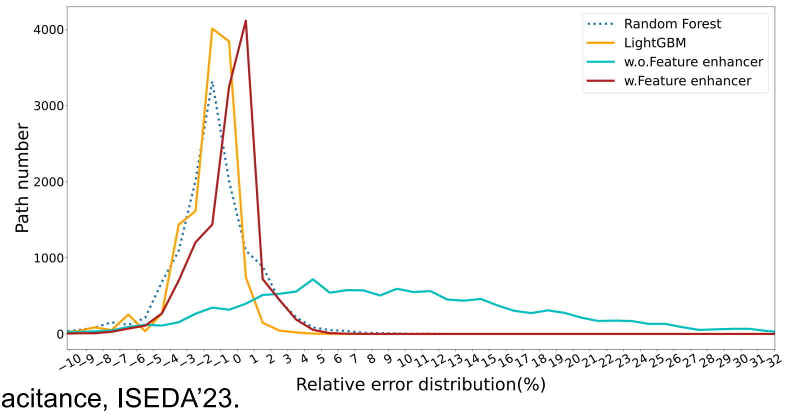
$$D_{Elmore} = \sum_{i\in N}\left(R_i * \sum C_D\right)$$

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

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

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

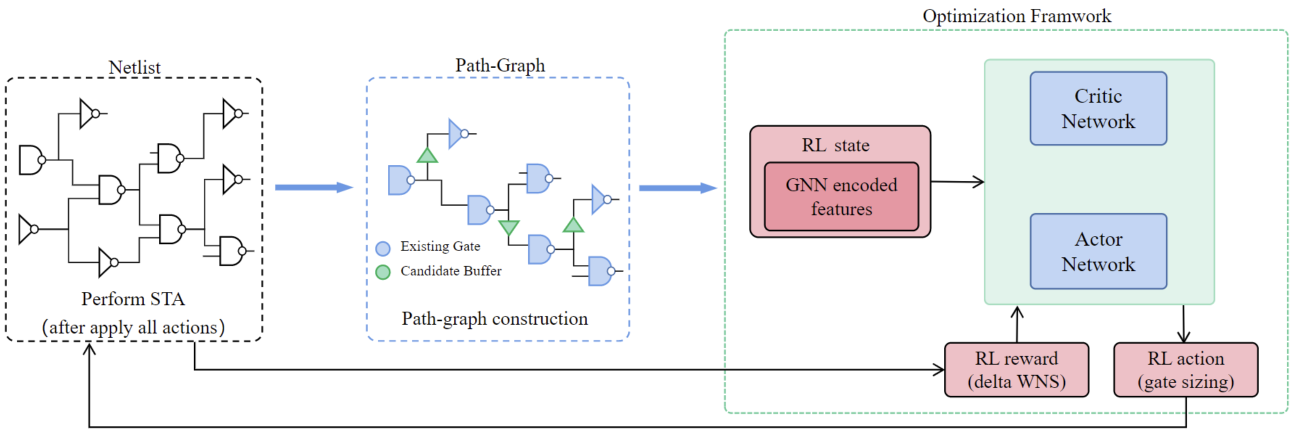
Enhance Feature

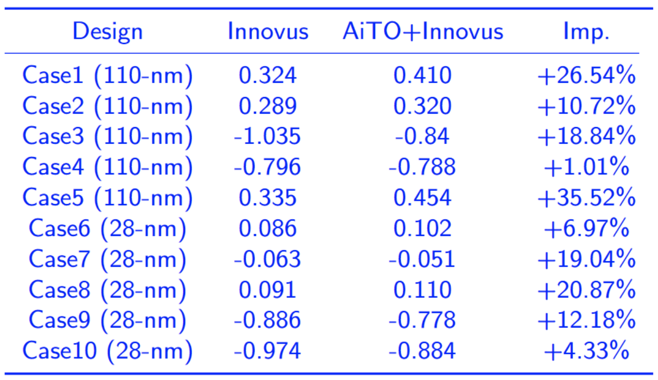


- H. Liu, et. al, Accurate Timing Path Delay Learning using Feature Enhancer with Effective Capacitance, ISEDA'23.

AI EDA: AiTO

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

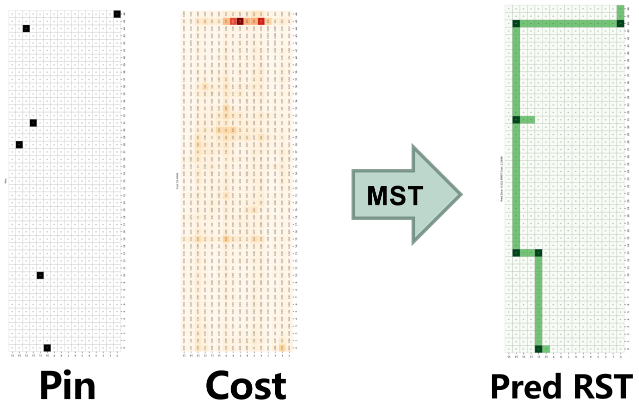




Design	Innovus	AiTO+Innovus	Imp.
Case1 (110-nm)	0.324	0.410	+26.54%
Case2 (110-nm)	0.289	0.320	+10.72%
Case3 (110-nm)	-1.035	-0.84	+18.84%
Case4 (110-nm)	-0.796	-0.788	+1.01%
Case5 (110-nm)	0.335	0.454	+35.52%
Case6 (28-nm)	0.086	0.102	+6.97%
Case7 (28-nm)	-0.063	-0.051	+19.04%
Case8 (28-nm)	0.091	0.110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

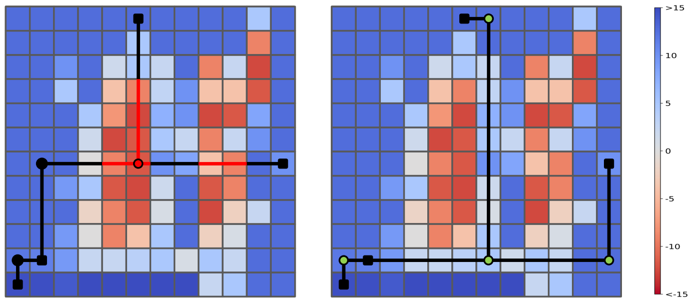
AI EDA: AiST

AI for Steiner Tree Generation



Pin Cost Pred RST

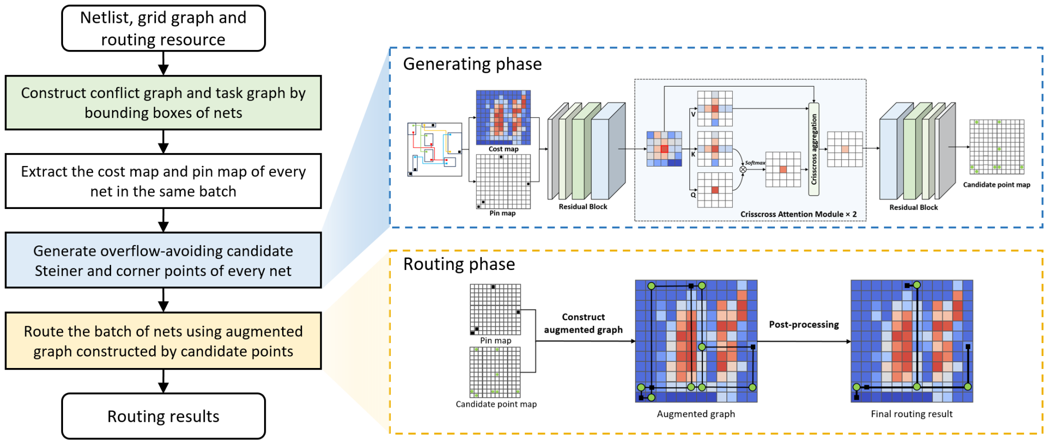
Motivation



HubRouter
Without Congestion

Our Method
Aviod Congestion

Method



Netlist, grid graph and routing resource

Construct conflict graph and task graph by bounding boxes of nets

Extract the cost map and pin map of every net in the same batch

Generate overflow-avoiding candidate Steiner and corner points of every net

Route the batch of nets using augmented graph constructed by candidate points

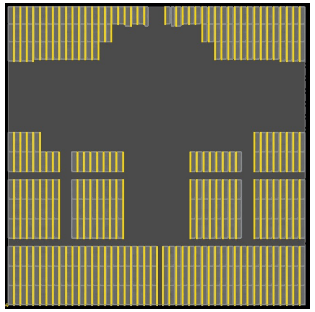
Routing results

Results

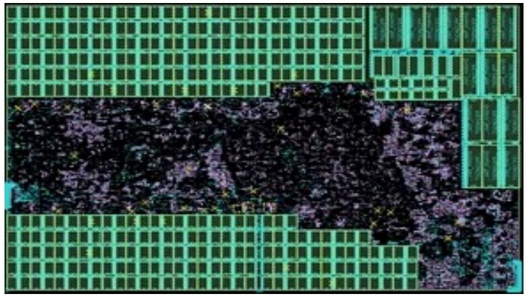
Metric	Method	ADA01	ADA02	ADA03	ADA04	ADA05	NEW01	NEW02	NEW03
WL	GeoSteiner	**3389601**	**3209172**	**9330748**	**8865643**	**9784471**	**2320456**	**4595235**	**7371273**
	FLUTE+ES	3418461	3235803	9417934	8896007	9886249	2347941	4651033	7454720
	HR-GAN	3407033	3229110	9355980	8888775	9832110	2339204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

AI EDA： AiMP

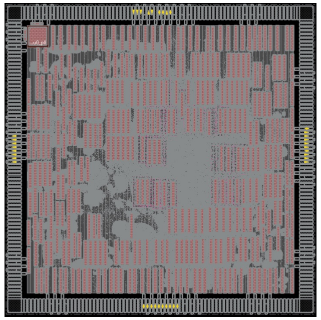
- **AI for Macro placement**
  - **OpenC910, 200W unit, tens of millions of gates, 332 macro modules, 28nm.**
    - **Engineer (based on commercial tools): Utilization 50%, Frequency: 526M**
    - **AiMP： Utilization rate 50%, frequency: 654.88M, (increased by 24%).**



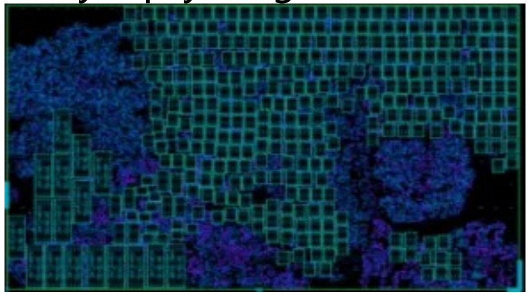
Our Engineer Result



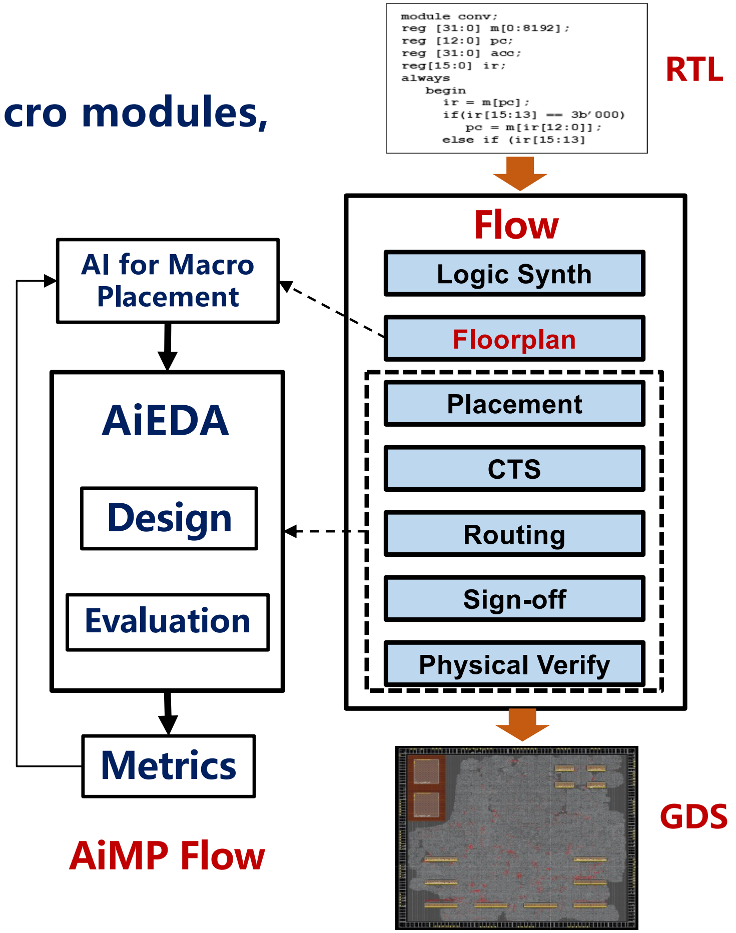
Synopsys Engineer Result



AiMP Result



AiMP
Synopsys： FreeForm Tool Result



```
module conv;
reg [31:0] m[0:8192];
reg [12:0] pc;
reg [31:0] acc;
reg[15:0] ir;
always
   begin
      ir = m[pc];
      if(ir[15:13] == 3b’000)
         pc = m[ir[12:0]];
      else if (ir[15:13]
```

AiMP Flow

Chip Design Flow

01 RTL-to-GDS (iEDA)

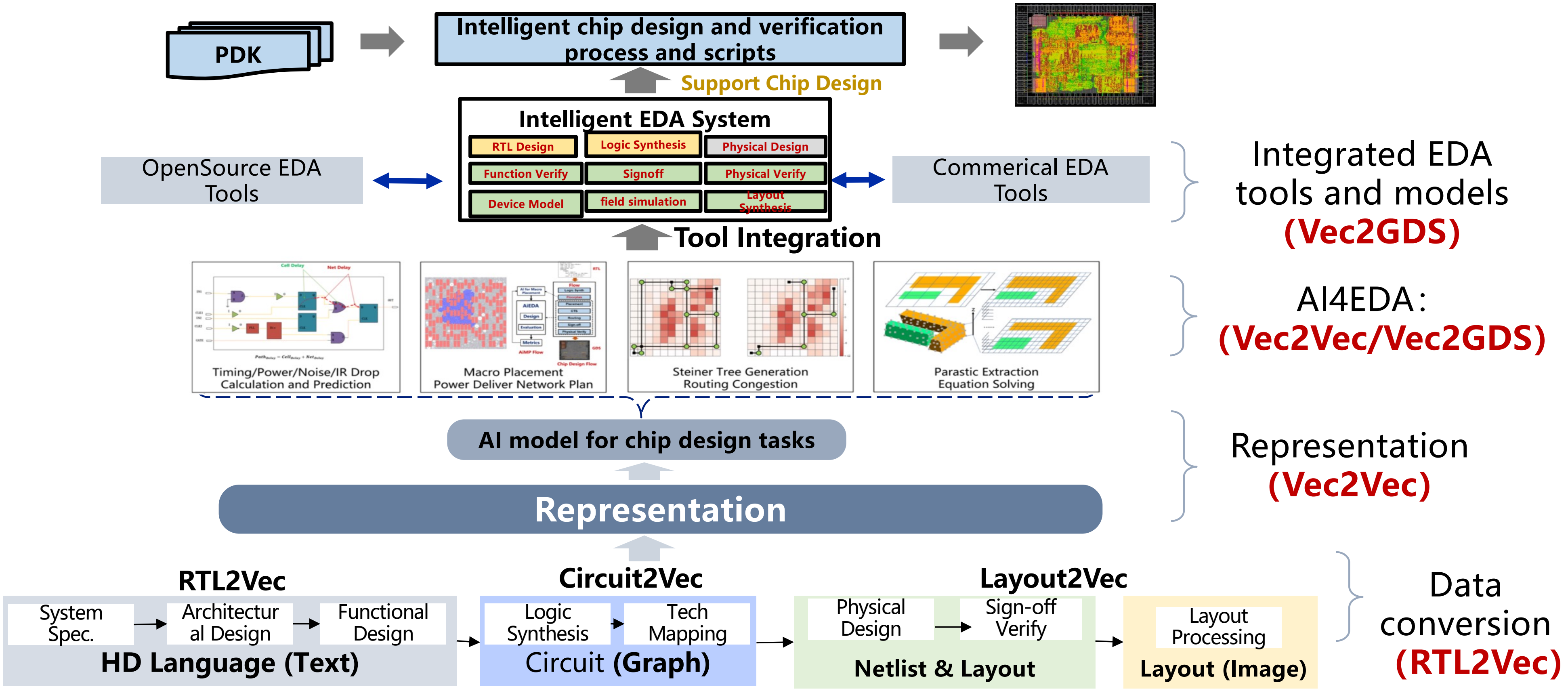
02 RTL-to-Vector (AiEDA)

03 Vector-to-Vector

04 Vector-to-GDS

Vector-to-GDS

- **Integrate existing AI-based information into EDA tools or chip design processes to achieve efficient and high-quality chip design.**



PDK

Intelligent chip design and verification process and scripts

Support Chip Design

Intelligent EDA System

RTL Design	Logic Synthesis	Physical Design
Function Verify	Signoff	Physical Verify
Device Model	field simulation	Layout Synthesis

OpenSource EDA Tools

Commerical EDA Tools

Tool Integration

Integrated EDA tools and models **(Vec2GDS)**

Timing/Power/Noise/IR Drop Calculation and Prediction

Macro Placement Power Deliver Network Plan

Steiner Tree Generation Routing Congestion

Parastic Extraction Equation Solving

AI4EDA: **(Vec2Vec/Vec2GDS)**

AI model for chip design tasks

Representation

Representation **(Vec2Vec)**

RTL2Vec

System Spec. → Architectural Design → Functional Design

HD Language (Text)

Circuit2Vec

Logic Synthesis → Tech Mapping

Circuit **(Graph)**

Layout2Vec

Physical Design → Sign-off Verify

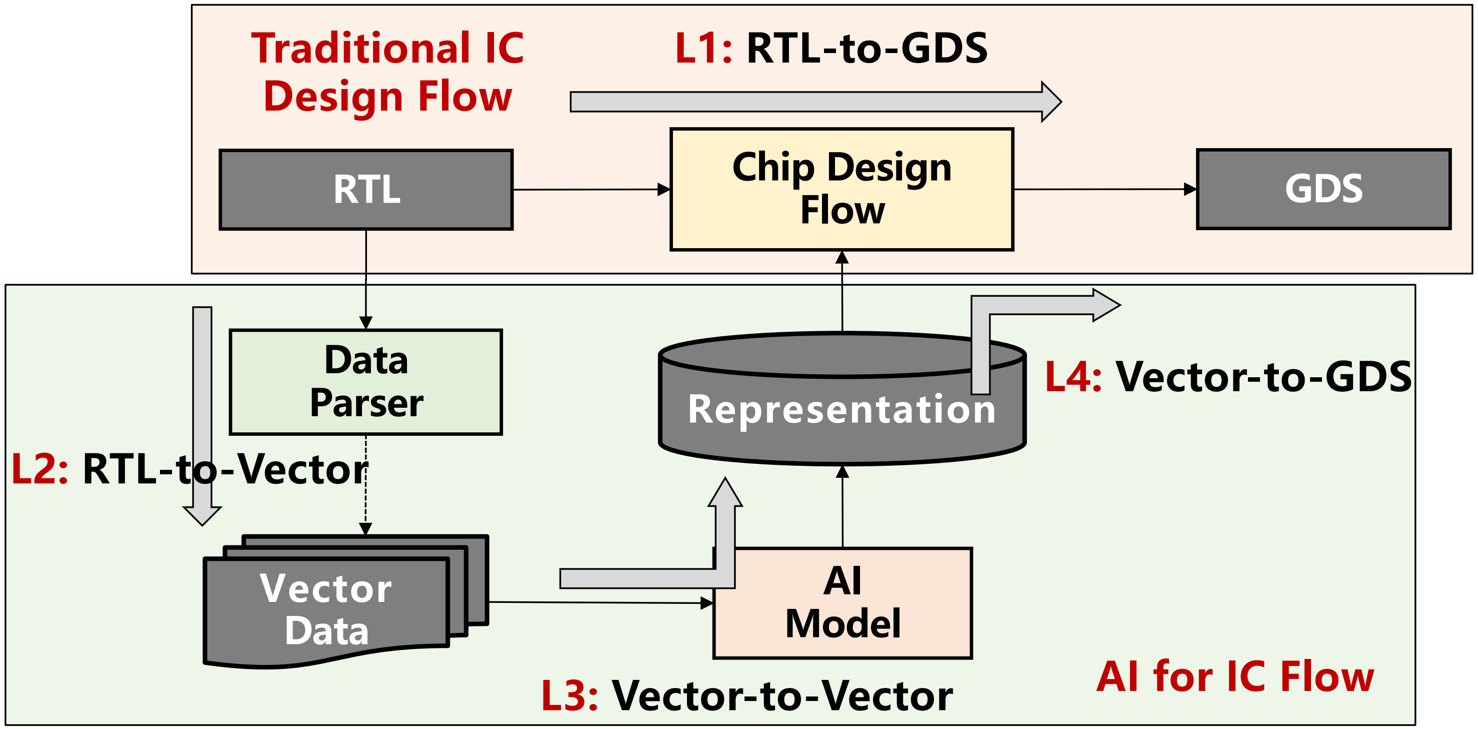
Netlist & Layout

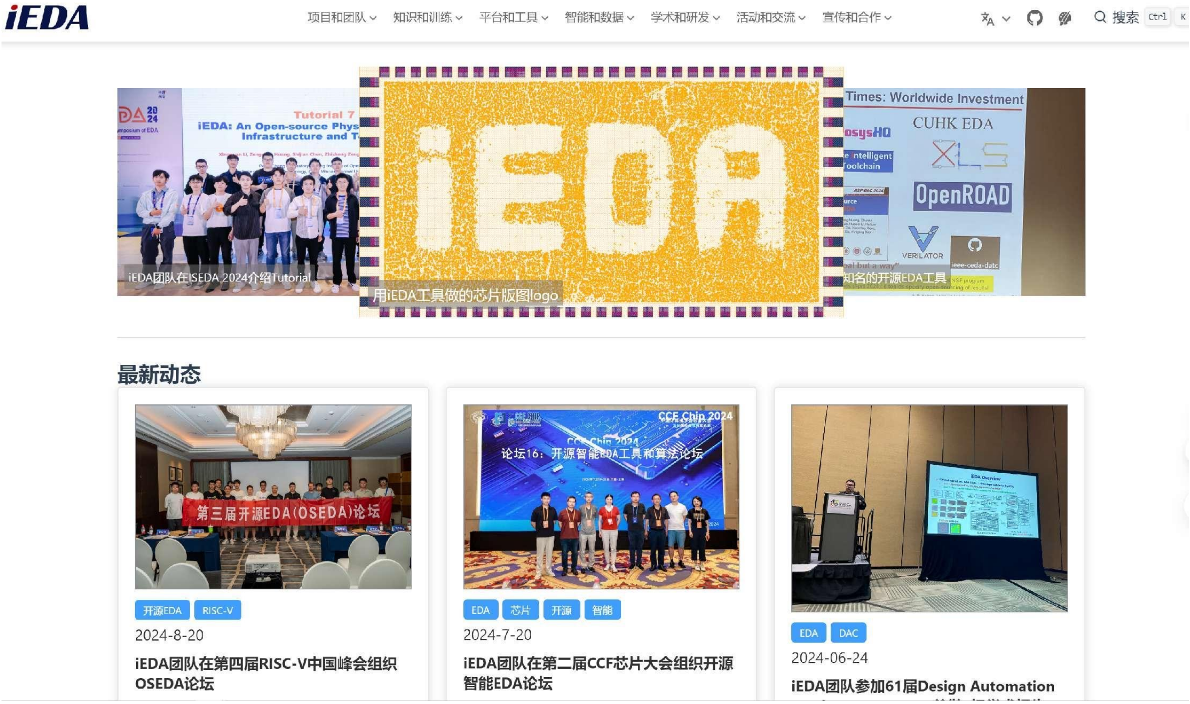
Layout Processing

Layout (Image)

Data conversion **(RTL2Vec)**

Summary


Traditional IC Design Flow
L1: RTL-to-GDS
RTL
Chip Design Flow
GDS
Data Parser
L2: RTL-to-Vector
Representation
L4: Vector-to-GDS
Vector Data
AI Model
L3: Vector-to-Vector
AI for IC Flow



iEDA团队在ISEDA 2024介绍Tutorial

用iEDA工具做的芯片版图logo

知名的开源EDA工具

最新动态

开源EDA RISC-V

2024-8-20

iEDA团队在第四届RISC-V中国峰会组织OSEDA论坛

EDA 芯片 开源 智能

2024-7-20

iEDA团队在第二届CCF芯片大会组织开源智能EDA论坛

EDA DAC

2024-06-24

iEDA团队参加61届Design Automation

Thanks

iEDA website: ieda.oscc.cc/en

陶思敏（Simin Tao)
taosm@pcl.ac.cn