

# Tutorial 7 - Part 1

## iEDA Infrastructure

Zengrong Huang, Xingquan Li

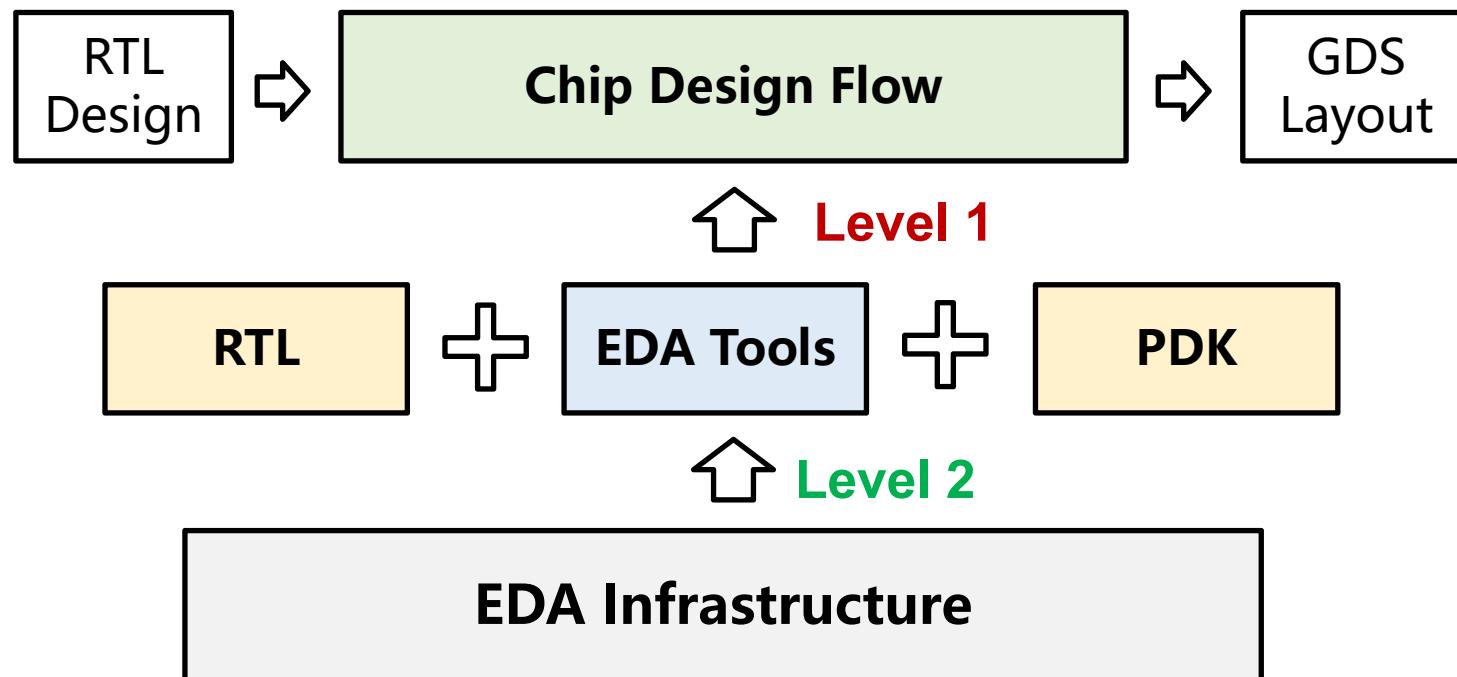
Peng Cheng Laboratory



- 01 **iEDA Infrastructure**
- 02 **Feature**
- 03 **Flow**

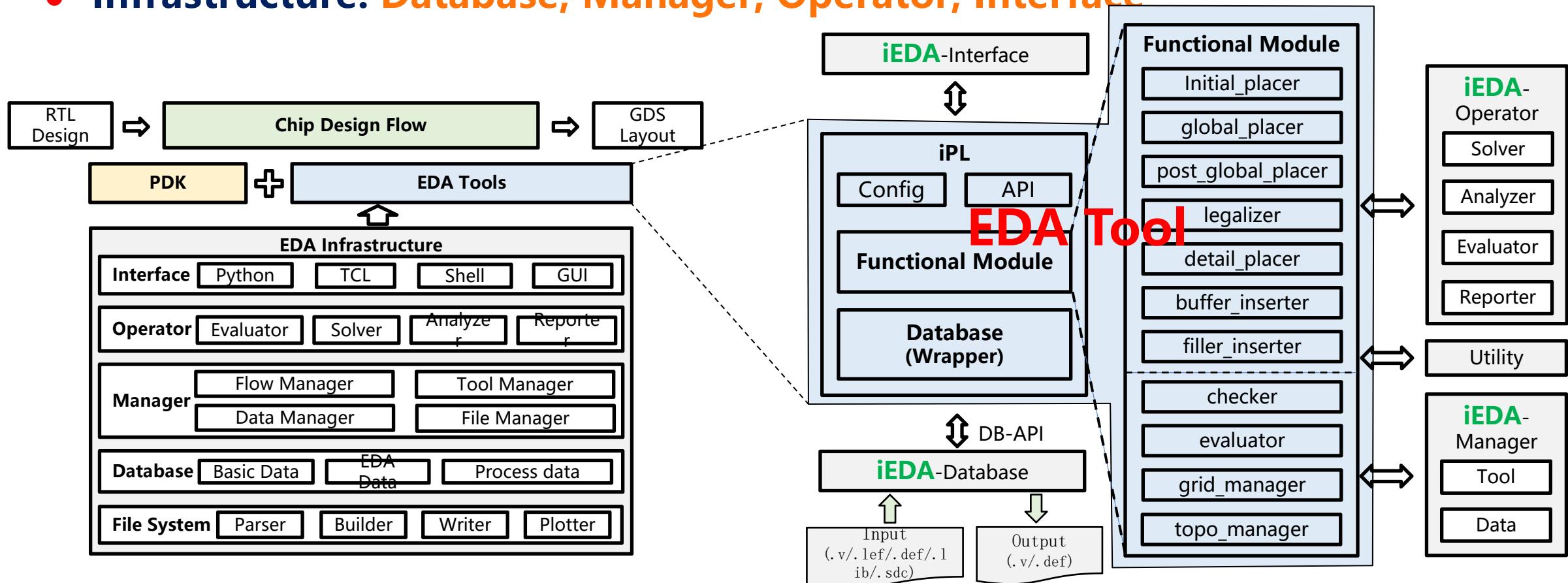
# We Need Infrastructure

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



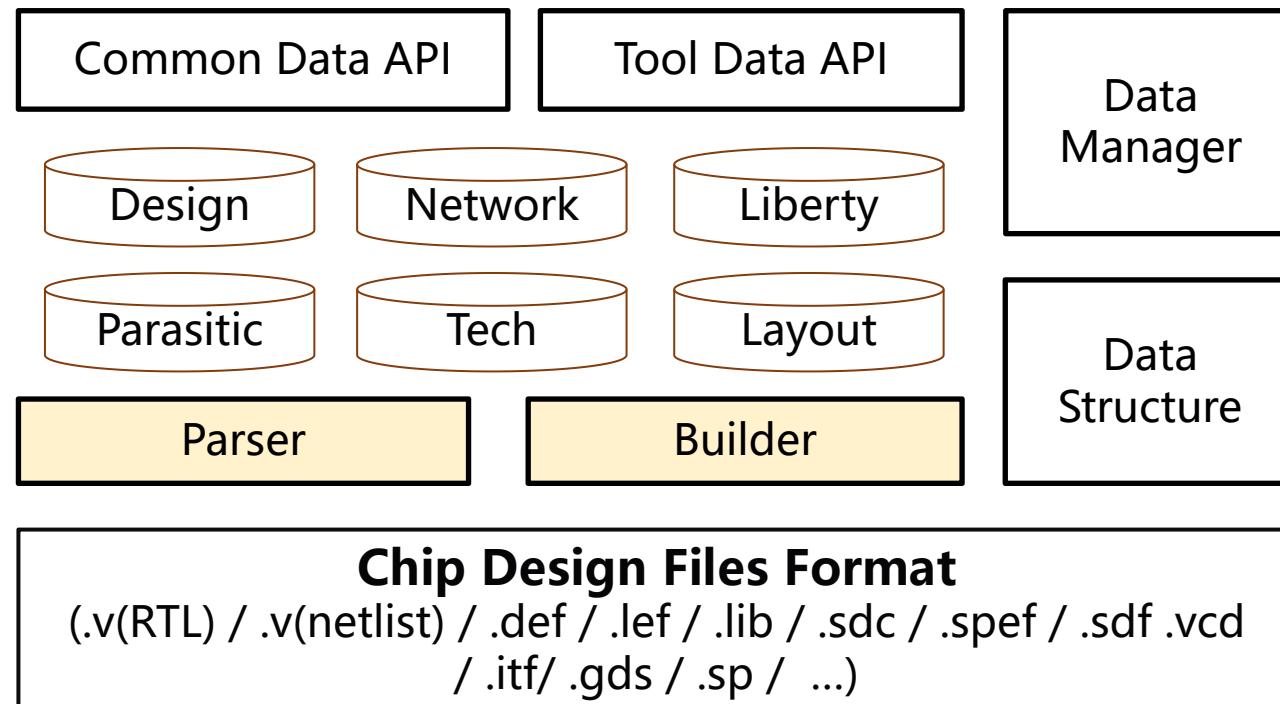
# iEDA Infra. Support EDA Tool

- 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**



# Parsers and Database

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



# Managers

## Platform Manager

### Data

- Config
- ChipData
- Interactive
- Proc Data
- ...

### Flows

- Initialize
- Input
- Process
- Output

### Tools

- Floorplan
- NetOpt
- Placement
- CTS
- TimingOpt
- Legalization
- Routing
- Filler
- DRC
- ...

### Features

- Summary
- Density
- Wire Length
- Congestion
- Profiles
- ...

### Reports

- Statistic
- Evaluation
- Flow Results
- Timing
- DRC
- ...

### Files

- Config
- Design
- Procedure
- Serialize
- ...

# Interfaces

## TCL

- Flows
- DB
- Tools
- Evaluation
- Features
- Reports
- GUI

```
tcl_config  
tcl_contest  
tcl_eval  
tcl_feature  
tcl_flow  
tcl_gui  
tcl_icts  
tcl_idb  
tcl_idrc  
tcl_ifp  
tcl_ino  
tcl_instance  
tcl_ipdn  
tcl_ipl  
tcl_ipw  
tcl_irt  
tcl_ista  
tcl_ito  
tcl_report  
tcl_util
```

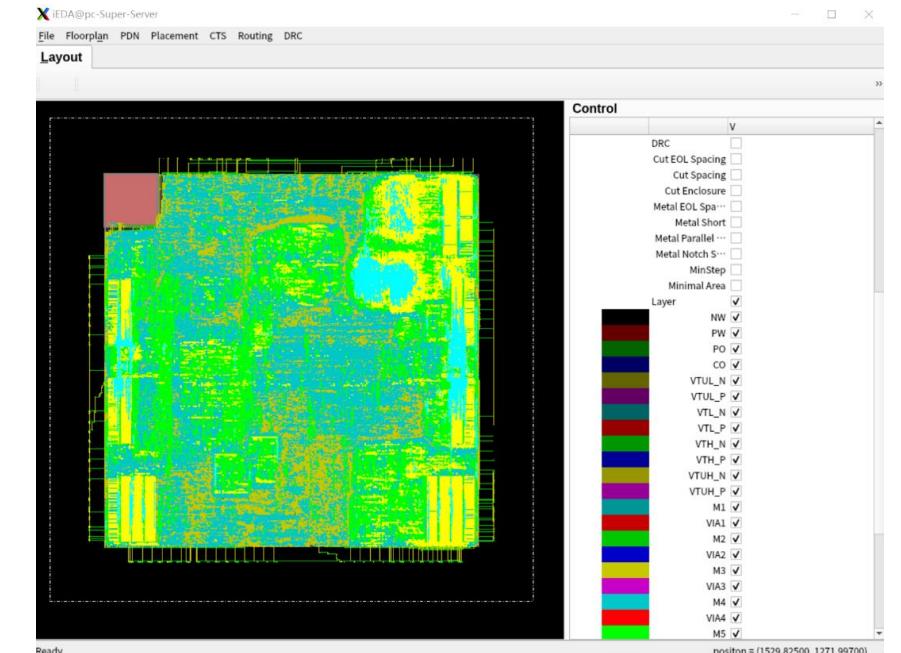
## Python

- Flows
- DB
- Tools
- Evaluation
- Features
- Reports

```
py_config  
py_eval  
py_feature  
py_flow  
py_icts  
py_idb  
py_idrc  
py_ifp  
py_imp  
py_ino  
py_instance  
py_ipdn  
py_ipl  
py_ipw  
py_irt  
py_ista  
py_ito  
py_report
```

## GUI

- File Operation
- Layout View
- Layers Control
- Shape Setting
- Instance Options
- Net Options
- PDN Options
- Track Grid
- DRC View
- Clock Tree View



# Evaluator: Horizontal Comparison

- Compare and analyze the Q&R
  - Designs and Flows
  - Tools and Algorithms

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

Flow Comparison

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

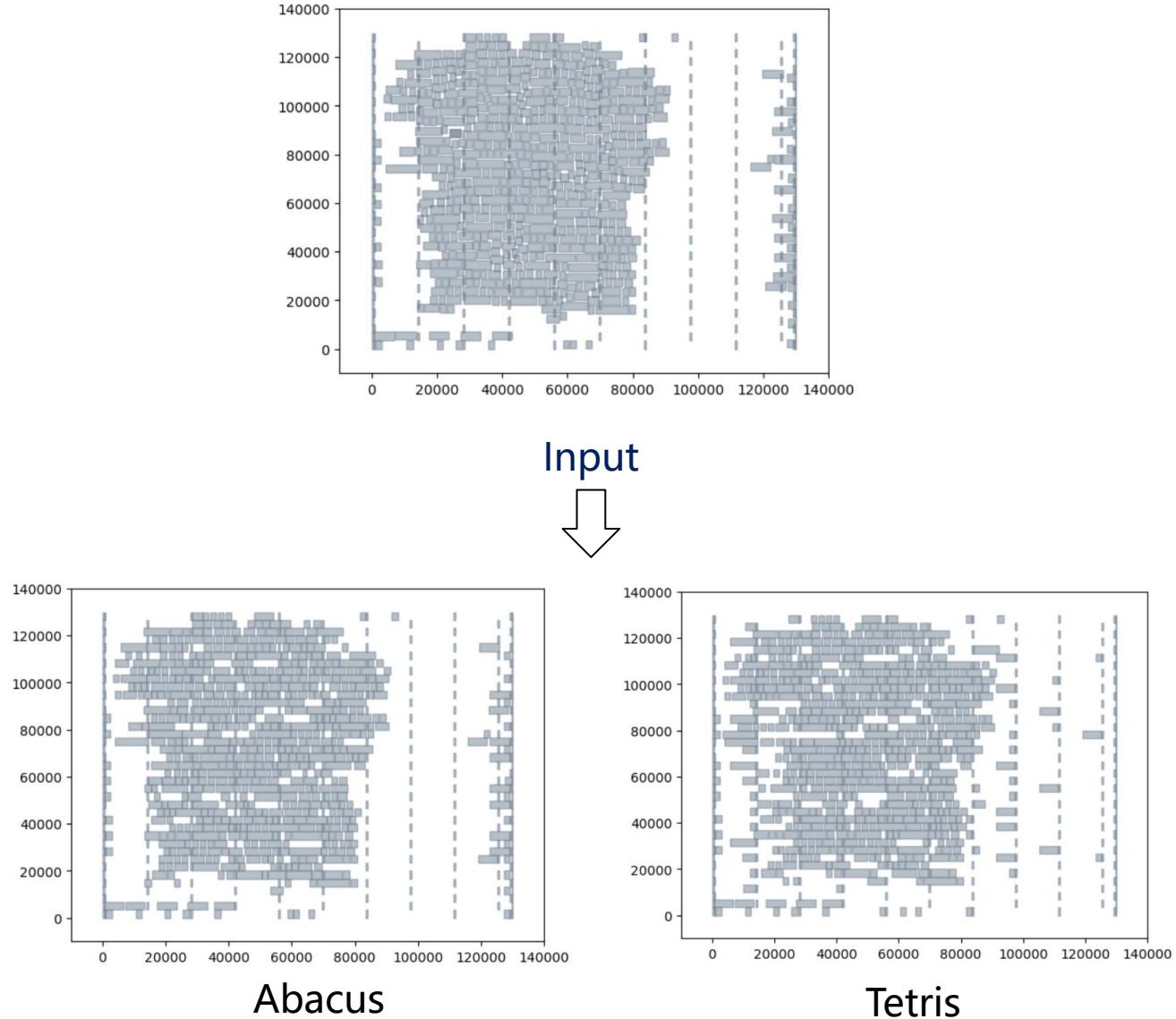
Design Comparison

# Evaluator: Horizontal Comparison

- Compare and analyze the Q&R
  - Designs and Flows
  - **Tools and Algorithms**

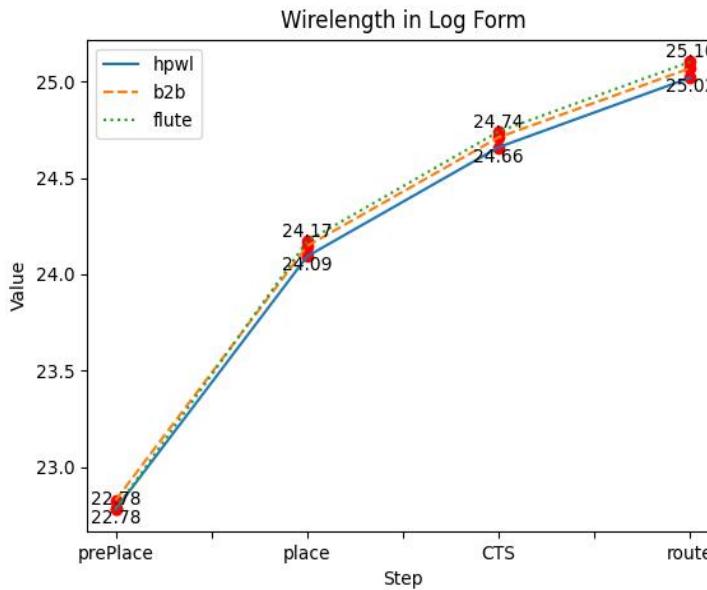
part metrics	abacus	tetris
global placement HPWL	10127910	10127910
legalization HPWL	10426323	13168231
detail placement HPWL	9901517	10928985
detail placement STWL	10637190	11674987
maximum STWL	431085	415325
total movement	795829	8705103
maximum movement	5684	218214
average congestion	0.8215	0.8134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

## Algorithm Comparison

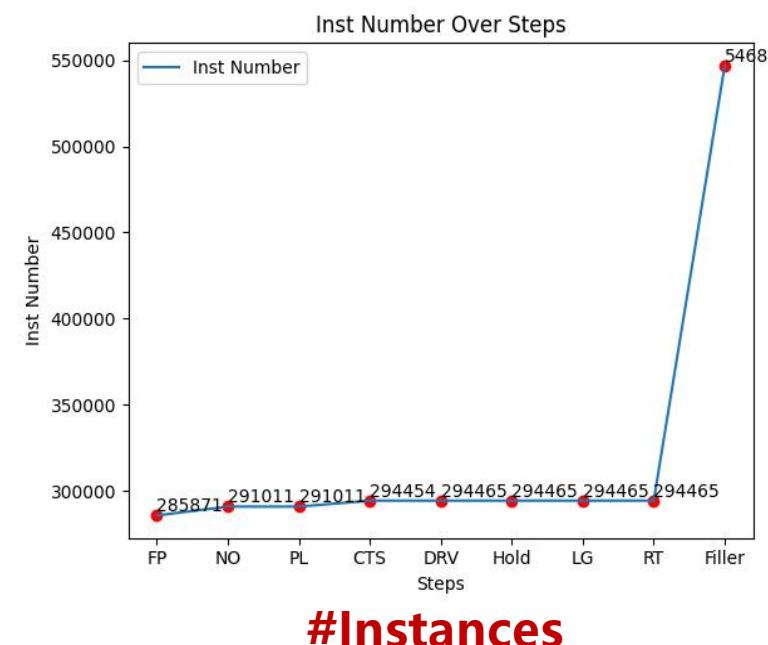


# Analyzer : Vertical Comparison

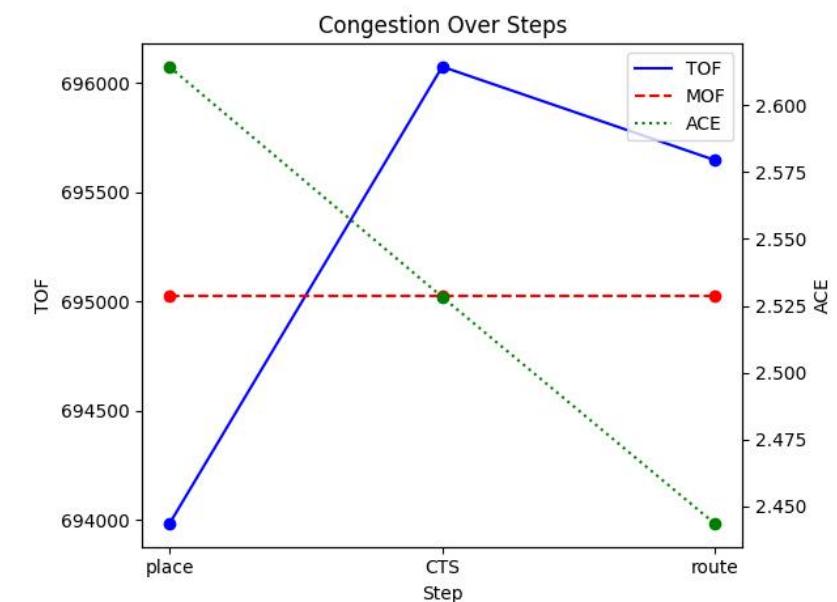
- Analyzing the numerical changes of an indicator **across different stages**
- Differences from: 1) data change, and 2) differences evaluation models
- Usage: evaluating the design quality, analyzing the margin, and optimizing collaboratively



Wirelength



#Instances



Congestion

- 01 iEDA Infrastructure
- 02 Feature
- 03 Flow

# iEDA Features

## For IC Designer

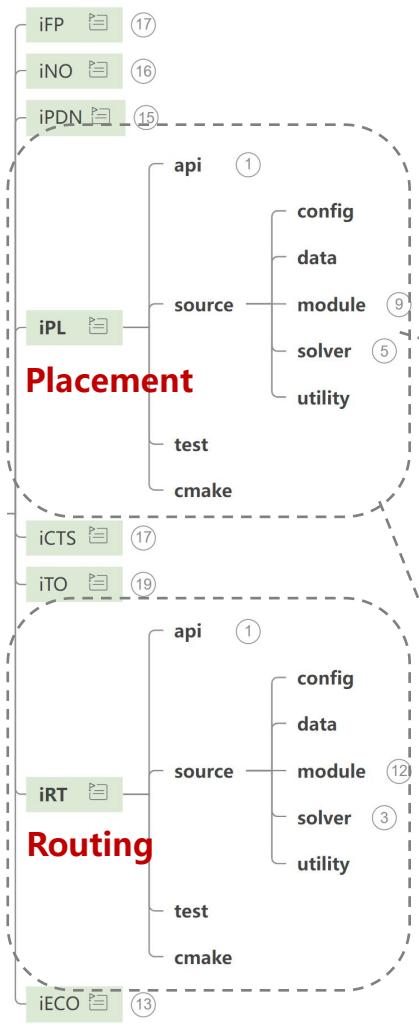
- Establish the backend physical design flow
- Support TCL language
- Provide rich performance evaluations
- Support layout visualization
- Support data snapshots, and toolchain data recovery
- User-friendly, with user guides and community support

- Provide Netlist -> GDSII interface
- Offer multi-language version interfaces (C++, Python, TCL, RUST)
- Offer rich performance evaluation interfaces
- Provide kinds of analysis and debugging tools
- Unified development framework, basic infrastructure
- Development manuals, community support

## For EDA Coder

# Uniform Software Framework and API

## Software Structure



## API

The API structure is organized into Python and TCL modules. Both modules contain several sub-modules corresponding to the software structure.

**Python:** py\_config, py\_eval, py\_feature, py\_flow, py\_icts, py\_idb, py\_idrc, py\_ifp, py\_ino, py\_instance, py\_ipdn, py\_ipl, py\_int, py\_ista, py\_ito, py\_report.

**TCL:** tcl\_config, tcl\_eval, tcl\_feature, tcl\_flow, tcl\_gui, tcl\_icts, tcl\_idb, tcl\_idrc, tcl\_ifp, tcl\_ino, tcl\_instance, tcl\_ipdn, tcl\_ipl, tcl\_int, tcl\_ista, tcl\_ito, tcl\_qt, tcl\_report, tcl\_util.

## Application

Python application code demonstrating the use of the API:

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

## Python

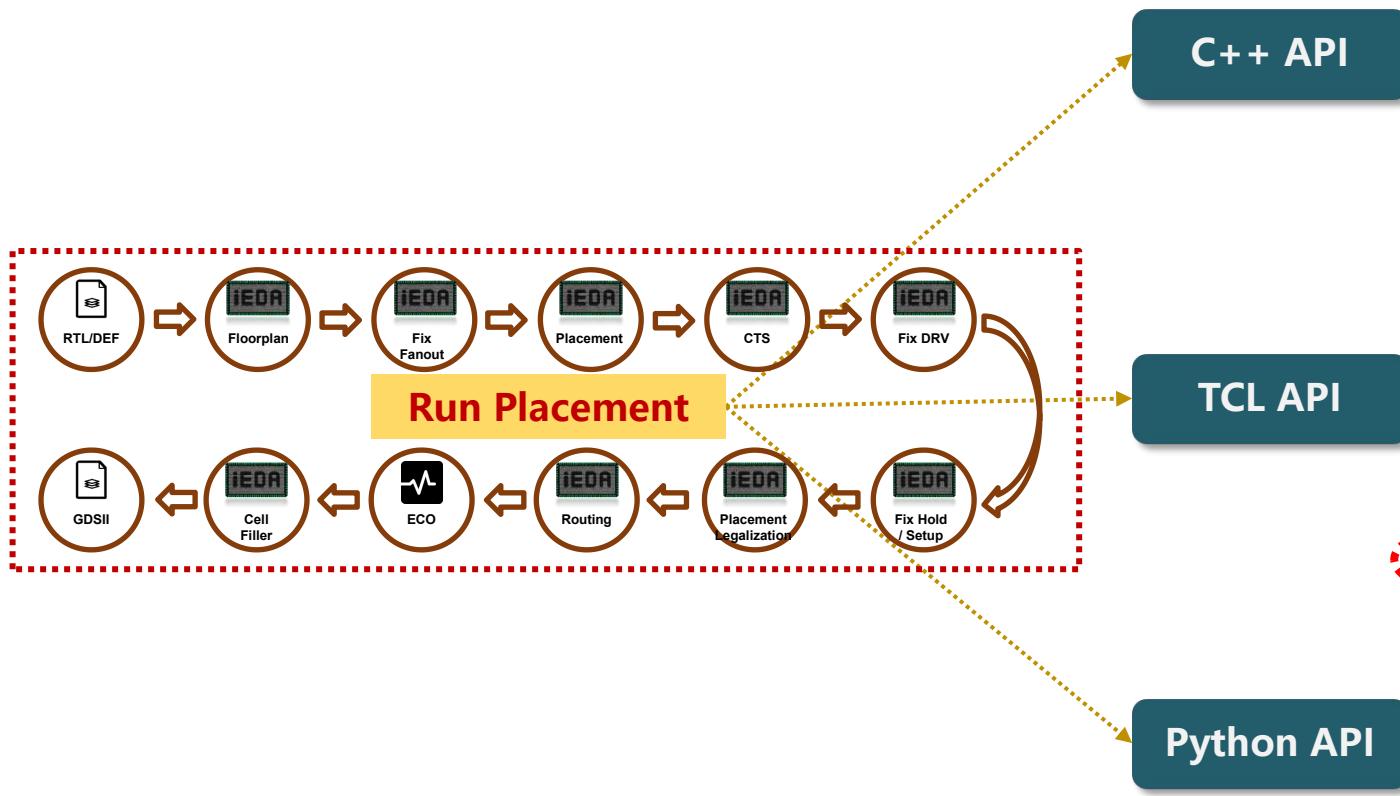
TCL application code demonstrating the use of the API:

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

## TCL

# Multiple Programming Language

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



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

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

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

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

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

iEDA Code

# Evaluation

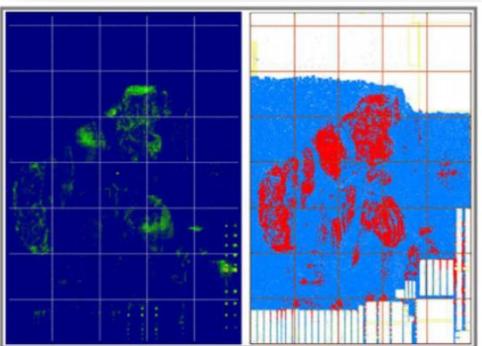
## ✓ iSTA/iPA: Timing and Power

- Setup/Hold
- Recovery/Removal
- Mlticycle
- Clock Gate
- Power
- IR Drop

```
Point
+---+---+
| net_top_pad (port)
| clock CLK_spk_clk_out (rise edge)
| logic
| spl_also_pad (port)
| u0_spl_also_pad (port)
| u0_spl_also_pad (PAD (PDWB4DGZ_H_G))
| u0_spl_also_pad (PAD (PDWB4DGZ_H_G))
| spl_flash_miso (net)
| edTFF_PDC3380_spl_flash_miso[0] (BUFFD16IMP3SP140LVT)
| edTFF_PDC3380_spl_flash_miso[1] (BUFFD16IMP3SP140LVT)
| edTFF_PDC12025_spl_flash_miso[0] (BUFFD4IMP3SP140LVT)
| edTFF_PDC12025_spl_flash_miso[1] (BUFFD4IMP3SP140LVT)
| FE_PDN12025_spl_flash_miso (net)
| u0_soc_top/u0_spl_flash/u0_spl_flash_shft_z77[0] (INR2D18MP4SP140LVT)
| u0_soc_top/u0_spl_flash/u0_spl_flash_shft_z77[1] (INR2D18MP4SP140LVT)
| edTFF_PDC6801_u0_soc_top/u0_spl_flash_u0_spl_flash_shft_z87[0] (BUFFD38IMP3SP140LVT)
| edTFF_PDC6801_u0_soc_top/u0_spl_flash_u0_spl_flash_shft_z87[1] (BUFFD38IMP3SP140LVT)
| FE_PDN6804_u0_soc_top/u0_spl_flash_u0_spl_flash_shft_z77 (clock net)
| edTFF_PDC6803_u0_soc_top/u0_spl_flash_u0_spl_flash_shft_z77[0] (BUFFD38IMP3SP140LVT)
| edTFF_PDC6802_u0_soc_top/u0_spl_flash_u0_spl_flash_shft_z77[1] (BUFFD28IMP3SP140LVT)
| edTFF_PDC6802_u0_soc_top/u0_spl_flash_u0_spl_flash_shft_z77[2] (BUFFD28IMP3SP140LVT)
| FE_PDN6802_u0_soc_top/u0_spl_flash_u0_spl_flash_shft_z77[0] (clock net)
| edTFF_PDC6801_u0_soc_top/u0_spl_flash_u0_spl_flash_shft_z77[2] (BUFFD38IMP3SP140LVT)
```

## ✓ iEVAL: Wirelength, Density, Congestion

- Density: Macro, Std Cell, Pin
- Congestion: GlobalNet, RUDYNet, LUTRYDY,
- Wirelength: WLD, HPWL, P2P, Steiner, eGR, GR, DR



## ✓ iDRC: Design Rule Vios

- Wire Connection
- Spacing Check, Enclosure Check, PRL Check, Metal Short, Minimal Area, MinStep
- DRC report, DRC visualization

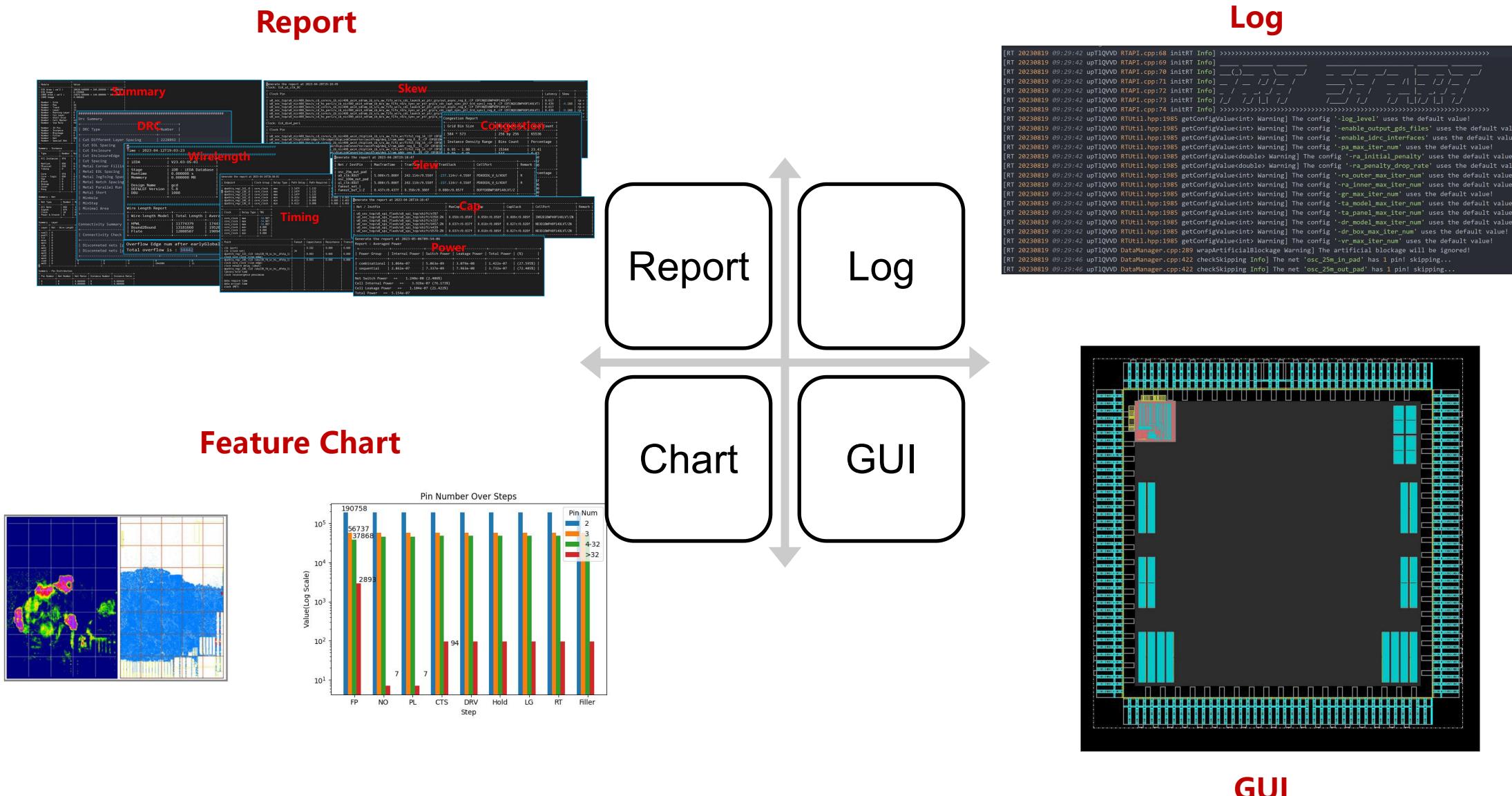
Drc Summary	
DRC Type	Number
Cut Different Layer Spacing	0
Cut EOL Spacing	0
Cut Enclosure	610489
Cut EnclosureEdge	0
Cut Spacing	264504
Metal Corner Filling Spacing	0
Metal EOL Spacing	8404599
Metal JogToJog Spacing	0
Metal Notch Spacing	1621088
Metal Parallel Run Length Spacing	3253512
Metal Short	6429817
MinHole	94
MinStep	7794204
Minimal Area	744067

## ✓ DB get: Count Basic Data in DB

- Layout: Area, Utilization, Row
- Cell: Nums, Pins, Area,
- Net: Nums, Fanout, Length
- Path: Nums, Depth
- Layer: Wire Density, Overflow

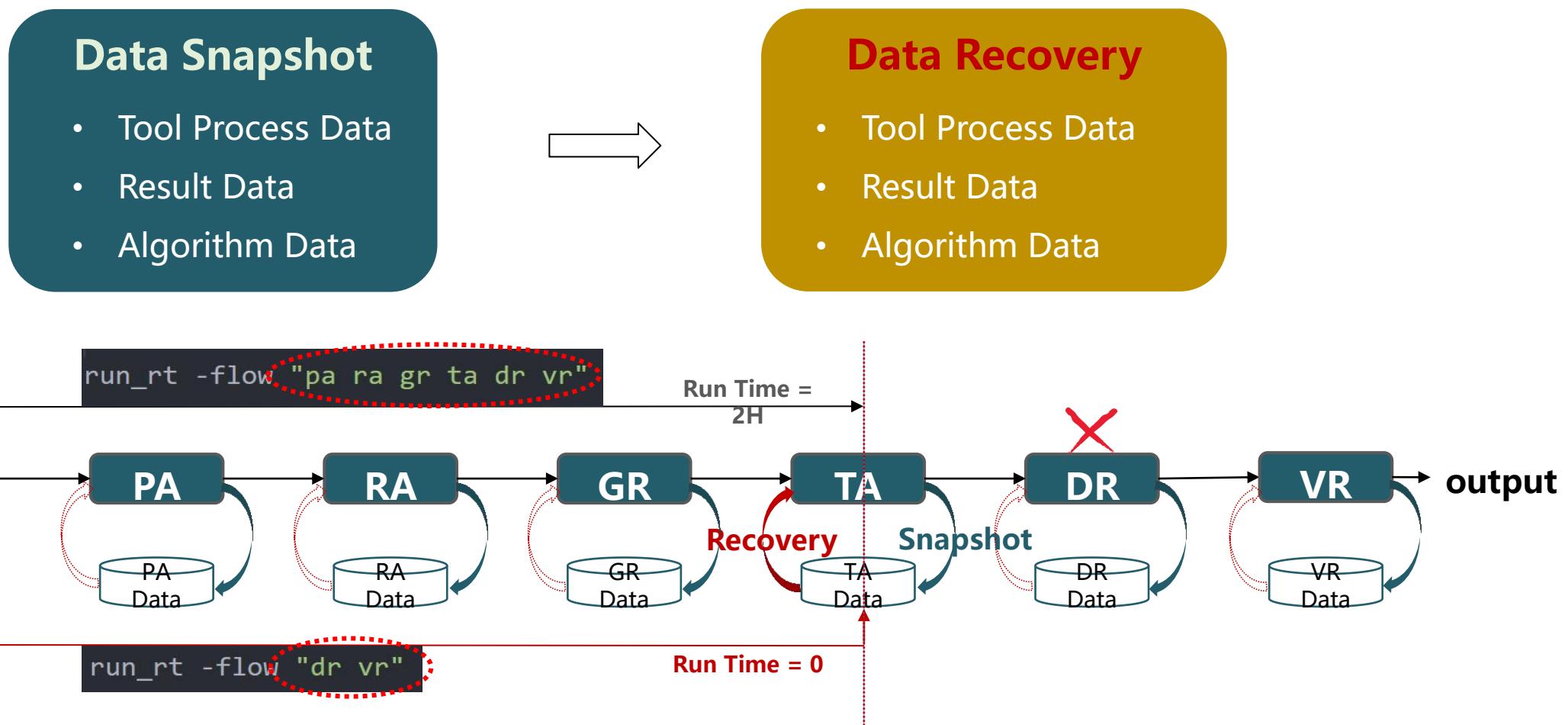
Summary	
Module	Value
DIE Area ( um^2 )	2249940.00000 = 1499.960000 * 1500.00000
DIE Usage	0.249103
CORE Area ( um^2 )	1340542.22400 = 1160.040000 * 1155.60000
CORE Usage	0.402985
Number - Site	9
Number - Row	1284
Number - Track	20
Number - Layer	32
Number - Routing Layer	10
Number - Cut Layer	10
Number - GCell Grid	5
Number - Cell Master	16314
Number - Via Rule	501
Number - IO Pin	110
Number - Instance	362106
Number - Blockage	21
Number - Filler	0
Number - Net	376471
Number - Special Net	5

# Multiple Data Analysis and Debug Methods



# Data Snapshot & Recovery

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



# Feature Extraction

## Basic Data

```

{
  "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": {
    "instances": [
      {
        "llx": 339920,
        "lly": 340000,
        "urx": 2660000,
        "ury": 2651200
      },
      {
        "llx": 2753910,
        "lly": 2746000,
        "master": "PFILLER0005_G",
        "name": "IOFIL_N_135",
        "orient": "S",
        "pin": [],
        "status": "FIXED",
        "type": "PAD SPACER",
        "urx": 2753920,
        "ury": 2966000
      },
      {
        "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
      }
    ],
    "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
      }
    ]
  },
  "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
    }
  ]
}

```

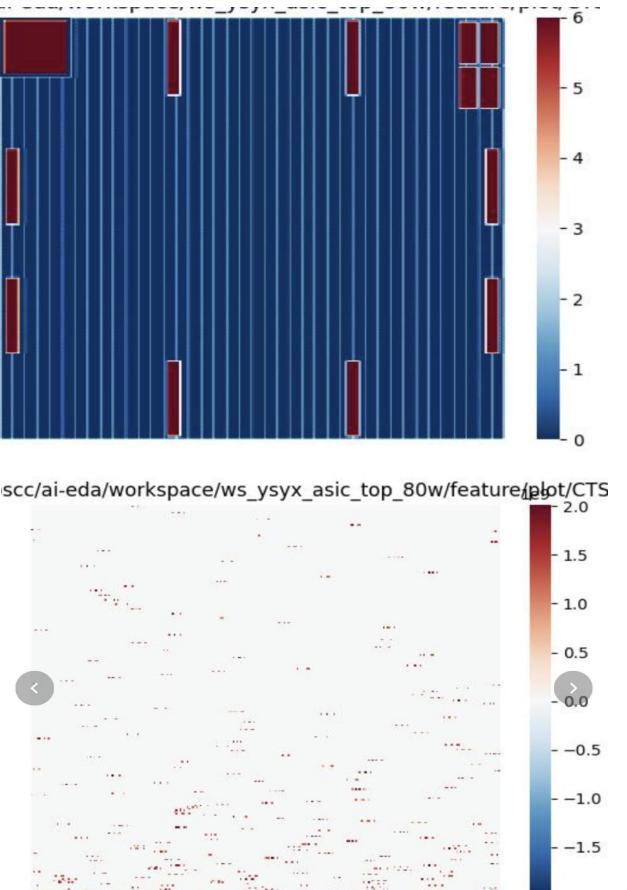
## Evaluation Data

```

ws_ysyx ASIC_top_80w> feature> plot> macro_density.csv
1,0,col_1,col_2,col_3,col_4,col_5,col_6,col_7,col_8,col_9,col_10,col_11,col_12,col_13,col_14,col_15,col_16,col_17,col_18,col_19,col_20,col_21,col_22,col_23,col_24,col_25,col_26,col_27,col_28,col_29,col_30,col_31,col_32,col_33,col_34,col_35,col_36,col_37,col_38,col_39,col_40,col_41,col_42,col_43,col_44,col_45,col_46,col_47,col_48,col_49,col_50,col_51,col_52,col_53,col_54,col_55,col_56,col_57,col_58,col_59,col_60,col_61,col_62,col_63,col_64,col_65,col_66,col_67,col_68,col_69,col_70,col_71,col_72,col_73,col_74,col_75,col_76,col_77,col_78,col_79,col_80,col_81,col_82,col_83,col_84,col_85,col_86,col_87,col_88,col_89,col_90,col_91,col_92,col_93,col_94,col_95,col_96,col_97,col_98,col_99,col_100,col_101,col_102,col_103,col_104,col_105,col_106,col_107,col_108,col_109,col_110,col_111,col_112,col_113,col_114,col_115,col_116,col_117,col_118,col_119,col_120,col_121,col_122,col_123,col_124,col_125,col_126,col_127,col_128,col_129,col_130,col_131,col_132,col_133,col_134,col_135,col_136,col_137,col_138,col_139,col_140,col_141,col_142,col_143,col_144,col_145,col_146,col_147,col_148,col_149,col_150,col_151,col_152,col_153,col_154,col_155,col_156,col_157,col_158,col_159,col_160,col_161,col_162,col_163,col_164,col_165,col_166,col_167,col_168,col_169,col_170,col_171,col_172,col_173,col_174,col_175,col_176,col_177,col_178,col_179,col_180,col_181,col_182,col_183,col_184,col_185,col_186,col_187,col_188,col_189,col_190,col_191,col_192,col_193,col_194,col_195,col_196,col_197,col_198,col_199,col_199,col_200,col_201,col_202,col_203,col_204,col_205,col_206,col_207,col_208,col_209,col_210,col_211,col_212,col_213,col_214,col_215,col_216,col_217,col_218,col_219,col_220,col_221,col_222,col_223,col_224,col_225,col_226,col_227,col_228,col_229,col_229,col_230,col_231,col_232,col_233,col_234,col_235,col_236,col_237,col_238,col_239,col_239,col_240,col_241,col_242,col_243,col_244,col_245,col_246,col_247,col_248,col_249,col_249,col_250,col_251,col_252,col_253,col_254,col_255,col_256,col_257,col_258,col_259,col_259,col_260,col_261,col_262,col_263,col_264,col_265,col_266,col_267,col_268,col_269,col_269,col_270,col_271,col_272,col_273,col_274,col_275,col_276,col_277,col_278,col_279,col_279,col_280,col_281,col_282,col_283,col_284,col_285,col_286,col_287,col_288,col_289,col_289,col_290,col_291,col_292,col_293,col_294,col_295,col_296,col_297,col_298,col_299,col_299,col_300,col_301,col_302,col_303,col_304,col_305,col_306,col_307,col_308,col_309,col_309,col_310,col_311,col_312,col_313,col_314,col_315,col_316,col_317,col_318,col_319,col_320,col_321,col_322,col_323,col_324,col_325,col_326,col_327,col_328,col_329,col_329,col_330,col_331,col_332,col_333,col_334,col_335,col_336,col_337,col_338,col_339,col_339,col_340,col_341,col_342,col_343,col_344,col_345,col_346,col_347,col_348,col_349,col_349,col_350,col_351,col_352,col_353,col_354,col_355,col_356,col_357,col_358,col_359,col_359,col_360,col_361,col_362,col_363,col_364,col_365,col_366,col_367,col_368,col_369,col_369,col_370,col_371,col_372,col_373,col_374,col_375,col_376,col_377,col_378,col_379,col_379,col_380,col_381,col_382,col_383,col_384,col_385,col_386,col_387,col_388,col_389,col_389,col_390,col_391,col_392,col_393,col_394,col_395,col_396,col_397,col_398,col_399,col_399,col_400,col_401,col_402,col_403,col_404,col_405,col_406,col_407,col_408,col_409,col_409,col_410,col_411,col_412,col_413,col_414,col_415,col_416,col_417,col_418,col_419,col_419,col_420,col_421,col_422,col_423,col_424,col_425,col_426,col_427,col_428,col_429,col_429,col_430,col_431,col_432,col_433,col_434,col_435,col_436,col_437,col_438,col_439,col_439,col_440,col_441,col_442,col_443,col_444,col_445,col_446,col_447,col_448,col_449,col_449,col_450,col_451,col_452,col_453,col_454,col_455,col_456,col_457,col_458,col_459,col_459,col_460,col_461,col_462,col_463,col_464,col_465,col_466,col_467,col_468,col_469,col_469,col_470,col_471,col_472,col_473,col_474,col_475,col_476,col_477,col_478,col_479,col_479,col_480,col_481,col_482,col_483,col_484,col_485,col_486,col_487,col_488,col_489,col_489,col_490,col_491,col_492,col_493,col_494,col_495,col_496,col_497,col_498,col_499,col_499,col_500,col_501,col_502,col_503,col_504,col_505,col_506,col_507,col_508,col_509,col_509,col_510,col_511,col_512,col_513,col_514,col_515,col_516,col_517,col_518,col_519,col_519,col_520,col_521,col_522,col_523,col_524,col_525,col_526,col_527,col_528,col_529,col_529,col_530,col_531,col_532,col_533,col_534,col_535,col_536,col_537,col_538,col_539,col_539,col_540,col_541,col_542,col_543,col_544,col_545,col_546,col_547,col_548,col_549,col_549,col_550,col_551,col_552,col_553,col_554,col_555,col_556,col_557,col_558,col_559,col_559,col_560,col_561,col_562,col_563,col_564,col_565,col_566,col_567,col_568,col_569,col_569,col_570,col_571,col_572,col_573,col_574,col_575,col_576,col_577,col_578,col_579,col_579,col_580,col_581,col_582,col_583,col_584,col_585,col_586,col_587,col_588,col_589,col_589,col_590,col_591,col_592,col_593,col_594,col_595,col_596,col_597,col_598,col_599,col_599,col_600,col_601,col_602,col_603,col_604,col_605,col_606,col_607,col_608,col_609,col_609,col_610,col_611,col_612,col_613,col_614,col_615,col_616,col_617,col_618,col_619,col_619,col_620,col_621,col_622,col_623,col_624,col_625,col_626,col_627,col_628,col_629,col_629,col_630,col_631,col_632,col_633,col_634,col_635,col_636,col_637,col_638,col_639,col_639,col_640,col_641,col_642,col_643,col_644,col_645,col_646,col_647,col_648,col_649,col_649,col_650,col_651,col_652,col_653,col_654,col_655,col_656,col_657,col_658,col_659,col_659,col_660,col_661,col_662,col_663,col_664,col_665,col_666,col_667,col_668,col_669,col_669,col_670,col_671,col_672,col_673,col_674,col_675,col_676,col_677,col_678,col_679,col_679,col_680,col_681,col_682,col_683,col_684,col_685,col_686,col_687,col_688,col_689,col_689,col_690,col_691,col_692,col_693,col_694,col_695,col_696,col_697,col_698,col_699,col_699,col_700,col_701,col_702,col_703,col_704,col_705,col_706,col_707,col_708,col_709,col_709,col_710,col_711,col_712,col_713,col_714,col_715,col_716,col_717,col_718,col_719,col_719,col_720,col_721,col_722,col_723,col_724,col_725,col_726,col_727,col_728,col_729,col_729,col_730,col_731,col_732,col_733,col_734,col_735,col_736,col_737,col_738,col_739,col_739,col_740,col_741,col_742,col_743,col_744,col_745,col_746,col_747,col_748,col_749,col_749,col_750,col_751,col_752,col_753,col_754,col_755,col_756,col_757,col_758,col_759,col_759,col_760,col_761,col_762,col_763,col_764,col_765,col_766,col_767,col_768,col_769,col_769,col_770,col_771,col_772,col_773,col_774,col_775,col_776,col_777,col_778,col_779,col_779,col_780,col_781,col_782,col_783,col_784,col_785,col_786,col_787,col_788,col_789,col_789,col_790,col_791,col_792,col_793,col_794,col_795,col_796,col_797,col_798,col_799,col_799,col_800,col_801,col_802,col_803,col_804,col_805,col_806,col_807,col_808,col_809,col_809,col_810,col_811,col_812,col_813,col_814,col_815,col_816,col_817,col_818,col_819,col_819,col_820,col_821,col_822,col_823,col_824,col_825,col_826,col_827,col_828,col_829,col_829,col_830,col_831,col_832,col_833,col_834,col_835,col_836,col_837,col_838,col_839,col_839,col_840,col_841,col_842,col_843,col_844,col_845,col_846,col_847,col_848,col_849,col_850,col_851,col_852,col_853,col_854,col_855,col_856,col_857,col_858,col_859,col_860,col_861,col_862,col_863,col_864,col_865,col_866,col_867,col_868,col_869,col_869,col_870,col_871,col_872,col_873,col_874,col_875,col_876,col_877,col_878,col_879,col_879,col_880,col_881,col_882,col_883,col_884,col_885,col_886,col_887,col_888,col_889,col_889,col_890,col_891,col_892,col_893,col_894,col_895,col_896,col_897,col_898,col_899,col_899,col_900,col_901,col_902,col_903,col_904,col_905,col_906,col_907,col_908,col_909,col_909,col_910,col_911,col_912,col_913,col_914,col_915,col_916,col_917,col_918,col_919,col_919,col_920,col_921,col_922,col_923,col_924,col_925,col_926,col_927,col_928,col_929,col_929,col_930,col_931,col_932,col_933,col_934,col_935,col_936,col_937,col_938,col_939,col_939,col_940,col_941,col_942,col_943,col_944,col_945,col_946,col_947,col_948,col_949,col_949,col_950,col_951,col_952,col_953,col_954,col_955,col_956,col_957,col_958,col_959,col_959,col_960,col_961,col_962,col_963,col_964,col_965,col_966,col_967,col_968,col_969,col_969,col_970,col_971,col_972,col_973,col_974,col_975,col_976,col_977,col_978,col_979,col_979,col_980,col_981,col_982,col_983,col_984,col_985,col_986,col_987,col_988,col_989,col_989,col_990,col_991,col_992,col_993,col_994,col_995,col_995,col_996,col_997,col_997,col_998,col_999,col_999,col_1000

```

## Feature Map



# Rich API and Documentation

# C++ API Doc

	API Command	Type	Description
buildRCTree	set_num_threads	builder	set the numbers of threads
initRcTree	set_design_work_space	builder	set the directory to output the timing reports
resetRcTree	readLiberty	builder	read the liberty files
buildGraph	readDesign	builder	read the design verilog file
isBuildGraph	readSpcf	builder	read the spcf file
resetGraph	readSdc	builder	read the sdc file
resetGraphData	readAocv	builder	read the aocv files
insertBuffer	makeOrFindRCTreeNode	builder	make RC tree internal node
removeBuffer	makeOrFindRCTreeNode	builder	make RC tree pin node
repowerInstance	incrCap	builder	set the node's cap
moveInstance	makeResistor	builder	make resistor edge of RC tree
writeVerilog	updateRCTreeInfo	builder	update the RC info after making the RC tree
setSignificantDigits	builder	builder	set the significant digits of the timing report
incrUpdateTiming	action	incremental propagation to update the timing data	
updateTiming	action	update the timing data	

# User Manual

```

script
script|_
scripts/design/sky130_gcd/script
|__ DB_script
|   |-- db_init_lef.tcl          # Data process flow scripts
|   |-- db_init_lib_drv.tcl      # initialize lef
|   |-- db_init_lib_fixfanout.tcl# initialize lib only for flow of fix fanout
|   |-- db_init_lib_hold.tcl     # initialize lib only for flow of optimize hold
|   |-- db_init_lib_setup.tcl    # initialize lib only for flow of optimize setup
|   |-- db_init_lib.tcl          # initialize lib for common flow
|   |-- db_init_sdc.tcl          # initialize sdc
|   |-- db_init_spf.tcl          # initialize spf
|   |-- db_path_setting.tcl      # set paths for all processing technology files, including TechLEF, LEF, Lib, sdc and spf
|   |-- run_db_checknet.tcl      # check net connectivity based on data built by DEF (.def) and LEF (.lef & .tlef)
|   |-- run_db_report_eval.tcl   # report wire length and congestion based on data built by DEF (.def) and LEF (.lef & .tlef)
|   |-- run_db.tcl               # test building data by DEF (.def) and LEF (.lef & .tlef)
|   |-- run_def_to_gds_text.tcl  # transform data from DEF (.def) to GDSII (.gdsii)
|   |-- run_def_to_verilog.tcl   # transform data from DEF (.def) to netlist (.v)
|   |-- run_netlist_to_def.tcl   # transform data from netlist (.v) to DEF (.def)
|   |-- run_read_verilog.tcl    # test read verilog file (.v)
|__ iCTS_script
|   |-- run_iCTS_eval.tcl        # CTS Flow scripts
|   |-- run_iCTS_STA.tcl         # report wire length for CTS result
|   |-- run_iCTS.tcl             # report CTS STA
|__ iDRC_script
|   |-- run_idRC_gui.tcl         # DRC(Design Rule Check) flow scripts
|   |-- run_idRC.tcl             # show GUI for DRC result
|   |-- run_DRC                 # run DRC
|__ ifP_script
|   |-- module
|   |   |-- create_tracks.tcl    # Floorplan flow scripts
|   |   |-- pdn.tcl              # submodule for Floorplan scripts
|   |   |-- set_clocknet.tcl     # create tracks for routing layers
|   |   |-- pdn_networks          # create pdn networks
|   |-- run_ipF.tcl              # set clock net
|   |-- run_ipF.tcl              # run Floorplan
|__ iGUI_script
|   |-- run_iGUI.tcl             # GUI flow scripts
|   |-- run_GUI                  # run GUI
|__ iNO_script
|   |-- run_iNO_fix_fanout.tcl   # NO(Netlist Optimization) flow scripts
|   |-- run_iNO_fix_fanout.tcl   # run Fix Fanout
|__ iPL_script
|   |-- run_iPL_eval.tcl          # Placement flow scripts
|   |-- run_iPL_filler.tcl        # report congestion statistics and wire length for Placement result
|   |-- run_iPL_gui.tcl           # run standard cell filler
|   |-- run_iPL_gui.tcl           # run gui flow that shows Global Placement Processing result

scripts
|__ design                         #iEDA flows for different designs
|   |-- ispd18                      #tbd
|   |-- sky130_gcd                  #flow of gcd in sky130
|   |__ iEDA
|   |   |-- iEDA_config              # iEDA parameters configuration files
|   |   |-- README.md
|   |   |-- result                   # iEDA result output files
|   |   |-- run_iEDA_gui.py          # Python3 script for running all iEDA flow with GUI layout
|   |   |-- run_iEDA.py              # Python3 script for running all iEDA flow
|   |   |-- run_iEDA.sh              # POSIX shell script for running all iEDA flow

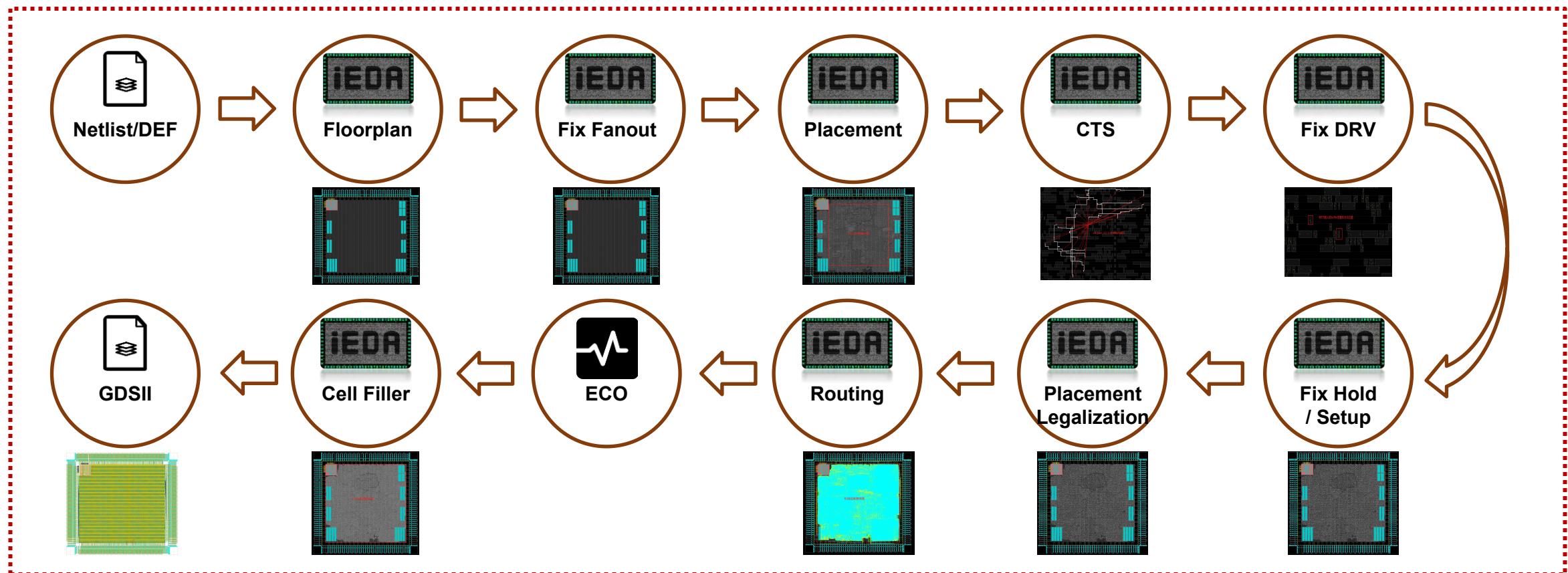
```

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

- 01 iEDA Infrastructure
- 02 Feature
- 03 Flow

# Flow: Netlist -> GDSII

- ✓ Enabling the Physical Design Flow,
- ✓ Supporting Technologies: 110nm, 28nm, Open-source Technologies (Sky130, Nangate45)



iEDA Physical Design Flow

# Environment Config

- **Userguide:** [https://gitee.com/oscc-project/iEDA/blob/master/docs/user\\_guide/iEDA\\_user\\_guide.md](https://gitee.com/oscc-project/iEDA/blob/master/docs/user_guide/iEDA_user_guide.md)

## Download and Compile iEDA

```
# 下载iEDA仓库  
git clone https://gitee.com/oscc-project/iEDA.git iEDA && cd iEDA  
# 通过apt安装编译依赖，需要root权限  
sudo bash build.sh -i apt  
# 编译 iEDA  
bash build.sh -j 16  
# 若能够正常输出 "Hello iEDA!" 则编译成功  
.bin/iEDA -script scripts/hello.tcl
```

拷贝 ./bin/iEDA 到目录 ./scripts/design/sky130\_gcd

```
# 拷贝 iEDA 到sky130 目录  
cp ./bin/iEDA scripts/design/sky130_gcd/.
```

## Design

### ✓ Netlist

## (28nm) Library

- ✓ TechLEF
- ✓ Std Cell LEF
- ✓ liberty
- ✓ sdc
- ✓ (spif)

## Environment



Server

Ubuntu 20.04.5 LTS



PDK



Design



3rd Party

# TCL Script

```
design          # iEDA flows for different designs
  isp18          # tbd
  sky130_gcd    # flow of gcd in sky130
    iEDA
    iEDA_config  # iEDA parameters configuration files
    README.md
    result        # iEDA result output files
    run_iEDA_gui.py # Python3 script for running all iEDA flow with GUI layout
    run_iEDA.py   # Python3 script for running all iEDA flow
    run_iEDA.sh   # POSTX shell script for running all iEDA flow
    script        # TCL script files
foundry
  README.md
  sky130         # SkyWater Open Source PDK
    lef           # lef files
    lib           # lib files
    sdc           # sdc files
    spef          # folder for spef files if needed
  hello.tcl      # Test running iEDA
```

```
DB_script
  db_init_lef.tcl
  db_init_lib_drv.tcl
  db_init_lib_fixfanout.tcl
  db_init_lib_hold.tcl
  db_init_lib_setup.tcl
  db_init_lib.tcl
  db_init_sdc.tcl
  db_init_spef.tcl
  db_path_setting.tcl
  run_db_checknet.tcl
  run_db_report_eval.tcl
  run_db.tcl
  run_def_to_gds_text.tcl
  run_def_to_verilog.tcl
  run_netlist_to_def.tcl
  run_read_verilog.tcl
iCTS_script
  run_iCTS_eval.tcl
  run_iCTS_STA.tcl
  run_iCTS.tcl
iDRC_script
  run_iDRC_gui.tcl
  run_iDRC.tcl
iFP_script
  module
    create_tracks.tcl
    pdn.tcl
    set_clocknet.tcl
  run_iFP.tcl
iGUI_script
  run_iGUI.tcl
iNO_script
  run_iNO_fix_fanout.tcl
iPL_script
  run_iPL_eval.tcl
  run_iPL_filler.tcl
# Data process flow scripts
# initialize lef
# initialize lib only for flow of drv
# initialize lib only for flow of fix fanout
# initialize lib only for flow of optimize hold
# initialize lib only for flow of optimize setup
# initialize lib for common flow
# initialize sdc
# initialize spef
# set paths for all processing technology files, including TechLEF, LEF, Lib, sdc and spef
# check net connectivity based on data built by DEF (.def) and LEF (.lef & .tlef)
# report wire length and congestion based on data built by DEF (.def) and LEF (.lef & .tlef)
# test building data by DEF (.def) and LEF (.lef & .tlef)
# transform data from DEF (.def) to GDSII (.gdsii)
# transform data from DEF (.def) to netlist (.v)
# transform data from netlist (.v) to DEF (.def)
# test read verilog file (.v)
# CTS flow scripts
# report wire length for CTS result
# report CTS STA
# run CTS
# DRC(Design Rule Check) flow scripts
# show GUI for DRC result
# run DRC
# Floorplan flow scripts
# submodule for Floorplan scripts
# create tracks for routing layers
# create pdn networks
# set clock net
# run Floorplan
# GUI flow scripts
# run GUI
# NO(Netlist Optimization) flow scripts
# run Fix Fanout
# Placement flow scripts
# report congestion statistics and wire length for Placement result
# run standard cell filler
```

# TCL Script

## Flow

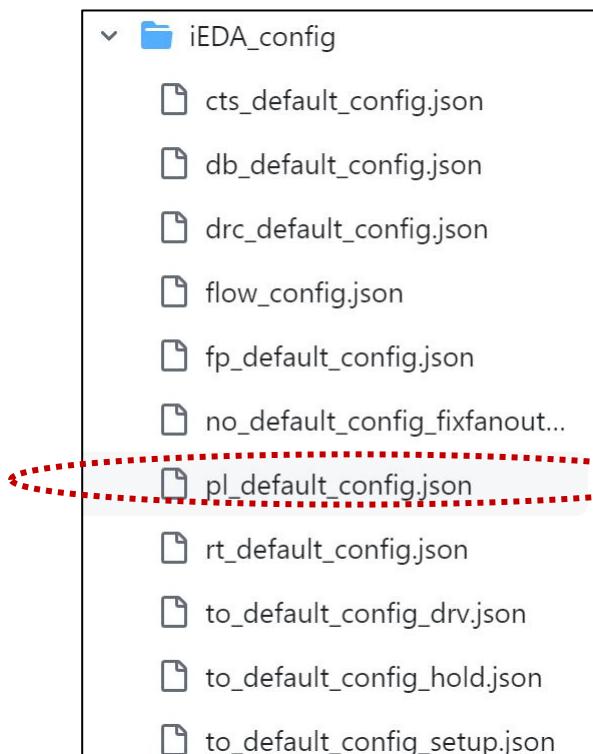
```
#####
## run floorplan
#####
os.system('./iEDA -script ./script/iFP_script/run_iFP.tcl')
#####
## run NO -- fix fanout
#####
os.system('./iEDA -script ./script/iNO_script/run_iNO_fix_fanout.tcl')
#####
## run Placer
#####
os.system('./iEDA -script ./script/iPL_script/run_iPL.tcl')
os.system('./iEDA -script ./script/iPL_script/run_iPL_eval.tcl')
# ...
# run CTS
# =====
os.system('./iEDA -script ./script/iCTS_script/run_iCTS.tcl')
os.system('./iEDA -script ./script/iCTS_script/run_iCTS_eval.tcl')
os.system('./iEDA -script ./script/iCTS_script/run_iCTS_STA.tcl')
#####
## run TO -- fix_drv
#####
os.system('./iEDA -script ./script/iTO_script/run_iTO_drv.tcl')
os.system('./iEDA -script ./script/iTO_script/run_iTO_drv_STA.tcl')
#####
# run TO -- opt_hold
#####
os.system('./iEDA -script ./script/iTO_script/run_iTO_setup.tcl')
os.system('./iEDA -script ./script/iTO_script/run_iTO_hold.tcl')
os.system('./iEDA -script ./script/iTO_script/run_iTO_hold_STA.tcl')
#####
# run PL Incremental Flow
#####
os.system('./iEDA -script ./script/iPL_script/run_iPL_legalization.tcl')
os.system('./iEDA -script ./script/iPL_script/run_iPL_legalization_eval.tcl')
# ...
# # run Router
# ...
#####
os.system('./iEDA -script ./script/iRT_script/run_iRT.tcl')
os.system('./iEDA -script ./script/iRT_script/run_iRT_eval.tcl')
os.system('./iEDA -script ./script/iRT_script/run_iRT_STA.tcl')
os.system('./iEDA -script ./script/iRT_script/run_iRT_DRC.tcl')
#####
## run Filler
#####
os.system('./iEDA -script ./script/iPL_script/run_iPL_filler.tcl')
#####
## run def to gdsii
#####
os.system('./iEDA -script ./script/DB_script/run_def_to_gds_text.tcl')
```

## Placement

```
#####
## init flow config
#####
flow_init -config ./iEDA_config/flow_config.json
#####
## read db config
#####
db_init -config ./iEDA_config/db_default_config.json
#####
## reset data path
#####
source ./script/DB_script/db_path_setting.tcl
#####
## reset sdc
#####
source ./script/DB_script/db_init_sdc.tcl
#####
## read Lef
#####
source ./script/DB_script/db_init_lef.tcl
#####
## read def
#####
def_init -path ./result/iTO_fix_fanout_result.def
#####
## run Placer
#####
run_placer -config ./iEDA_config/pl_default_config.json
#####
## Save def
#####
def_save -path ./result/iPL_result.def
#####
## Save netlist
#####
netlist_save -path ./result/iPL_result.v -exclude_cell_names {}
#####
## report
#####
report_db -path "./result/report/pl_db.rpt"
#####
## Exit
#####
flow_exit
```

# Parameter Config

## Config



iEDA / scripts / design / sky130\_gcd / iEDA\_config / pl\_default\_config.json

Code	Blame	82 lines (82 loc) · 2.38 KB
3		"is_max_length_opt": 0,
4		"max_length_constraint": 1000000,
5		"is_timing_effort": 0,
6		"is_congestion_effort": 0,
7		"ignore_net_degree": 100,
8		"num_threads": 1,
9		"GP": {
10		"Wirelength": {
11		"init_wirelength_coeff": 0.25,
12		"reference_hpwl": 446000000,
13		"min_wirelength_force_bar": -300
14		},
15		"Density": {
16		"target_density": 0.8,
17		"bin_cnt_x": 128,
18		"bin_cnt_y": 128
19		},
20		"Nesterov": {
21		"max_iter": 2000,
22		"max_backtrack": 10,
23		"init_density_penalty": 0.00008,
24		"target_overflow": 0.1,
25		"initial_prev_coordi_update_coeff": 100,
26		"min_precondition": 1.0,
27		"min_phi_coeff": 0.95,
28		"max_phi_coeff": 1.05
29		},
30		"BUFFER": {
31		"max_buffer_num": 10000,
32		"buffer_type": [
33		"sky130_fd_sc_hs_buf_1"
34		]
35		},
36		"LG": {
37		"max_displacement": 1000000,
38		"global_right_padding": 0
39		

iEDA / scripts / design / sky130\_gcd / iEDA\_config / pl\_default\_config.json

Code	Blame	82 lines (82 loc) · 2.38 KB
42		"max_displacement": 1000000,
43		"global_right_padding": 0,
44		"enable_networkflow": 0
45		},
46		"Filler": {
47		"first_iter": [
48		"sky130_fd_sc_hs_fill_8",
49		"sky130_fd_sc_hs_fill_4",
50		"sky130_fd_sc_hs_fill_2",
51		"sky130_fd_sc_hs_fill_1"
52		],
53		"second_iter": [
54		"sky130_fd_sc_hs_fill_8",
55		"sky130_fd_sc_hs_fill_4",
56		"sky130_fd_sc_hs_fill_2",
57		"sky130_fd_sc_hs_fill_1"
58		],
59		"min_filler_width": 1
60		},
61		"MP": {
62		"fixed_macro": [],
63		"fixed_macro_coordinate": [],
64		"blockage": [],
65		"guidance_macro": [],
66		"guidance": [],
67		"solution_type": "BStarTree",
68		"SimulateAnneal": {
69		"perturb_per_step": 100,
70		"cool_rate": 0.92
71		},
72		"Partition": {
73		"parts": 66,
74		"ufactor": 100,
75		"new_macro_density": 0.6
76		},
77		"halo_x": 0,
78		"halo_y": 0,
79		"output_path": "\${RESULT_DIR}/pl/"

# How to Run Netlist -> GDSII Flow by iEDA



Flow	Script	Config	Design Input
布图规划 (Floorpan)	./iEDA -script ./script/iFP_script/run_iFP.tcl		./result/verilog/gcd.v
网表优化 (Fix Fanout)	./iEDA -script ./script/iNO_script/run_iNO_fix_fanout.tcl	./iEDA_config/cts_default_config.json	./result/iFP_result.def
布局 (Placement)	./iEDA -script ./script/iPL_script/run_iPL.tcl	./iEDA_config/pl_default_config.json	./result/iTO_fix_fanout_resu
布局结果评估 (评估线长和拥塞)	./iEDA -script ./script/iPL_script/run_iPL_eval.tcl		./result/iPL_result.def
时钟树综合 (CTS)	./iEDA -script ./script/iCTS_script/run_iCTS.tcl	./iEDA_config/cts_default_config.json	./result/iPL_result.def
时钟树综合结果评估 (评估线长)	./iEDA -script ./script/iCTS_script/run_iCTS_eval.tcl		./result/iCTS_result.def
时钟树综合时序评估 (评估时序)	./iEDA -script ./script/iCTS_script/run_iCTS_STA.tcl		./result/iCTS_result.def
修复DRV违例 (Fix DRV Violation)	./iEDA -script ./script/iTO_script/run_iTO_drv.tcl	./iEDA_config/to_default_config_drv.json	./result/iCTS_result.def
Fix DRV结果评估 (评估时序)	./iEDA -script ./script/iTO_script/run_iTO_drv_STA.tcl		./result/iTO_drv_result.def
修复Hold违例 (Fix Hold Violation)	./iEDA -script ./script/iTO_script/run_iTO_hold.tcl	./iEDA_config/to_default_config_hold.json	./result/iTO_drv_result.def
Fix Hold结果评估 (评估时序)	./iEDA -script ./script/iTO_script/run_iTO_hold_STA.tcl		./result/iTO_hold_result.def
单元合法化 (Legalization)	./iEDA -script ./script/iPL_script/run_iPL_legalization.tcl	./iEDA_config/pl_default_config.json	./result/iTO_hold_result.def
合法化结果评估 (评估线长和拥塞)	./iEDA -script ./script/iPL_script/run_iPL_legalization_eval.tcl		./result/iPL_lg_result.def
布线 (Routing)	./iEDA -script ./script/iRT_script/run_iRT.tcl		./result/iPL_lg_result.def

# Report

## ✓ Statistics

- iFP
  - iNO
  - iPL
  - iCTS
  - iTO
  - iRT

```

Summary
+-----+
| Module | Value
+-----+
| DIE Area ( um^2 ) | 2249940.00000 = 1499.960000 * 1500.000000
| DIE Usage | 0.237865
| CORE Area ( um^2 ) | 1340542.224000 = 1160.040000 * 1155.600000
| CORE Usage | 0.399228

| Number - Site | 9
| Number - Row | 1284
| Number - Track | 20
| Number - Layer | 32
| Number - Routing Layer | 10
| Number - Cut Layer | 10
| Number - GCell Grid | 5
| Number - Cell Master | 16314
| Number - Via Rule | 492

| Number - IO Pin | 110
| Number - Instance | 362883
| Number - Blockage | 21
| Number - Filler | 0
| Number - Net | 377248
| Number - Special Net | 5
+-----+

```

# ✓ Evaluation

- Placement
  - CTS
  - Physical Incremental
  - Routing

Congestion Report		
Grid Bin Size	Bin Partition	Total Count
9063 * 9029	256 by 256	65536
Instance Density Range		
Instance Density Range	Bins Count	Percentage
0.95 ~ 1.00	5372	8.20
0.90 ~ 0.95	34	0.05
0.85 ~ 0.90	29	0.04
0.80 ~ 0.85	329	0.50
0.75 ~ 0.80	0	0.00
Pin Count Range		
Pin Count Range	Bins Count	Percentage
205 ~ 228	22	0.03
182 ~ 205	129	0.20
160 ~ 182	514	0.78
137 ~ 160	1555	2.37
114 ~ 137	252	0.38

## ✓ Design rule violations

- iRT

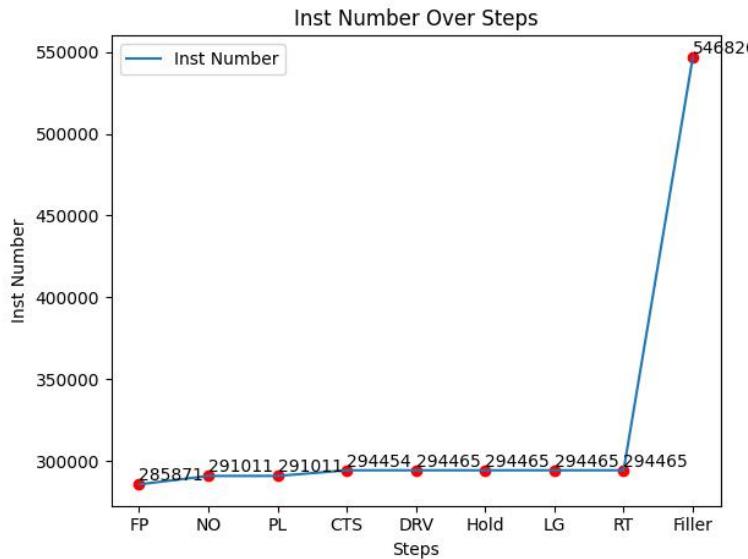
Drc Summary	
DRC Type	Number
Cut Different Layer Spacing	0
Cut EOL Spacing	0
Cut Enclosure	610489
Cut EnclosureEdge	0
Cut Spacing	264504
Metal Corner Filling Spacing	0
Metal EOL Spacing	8404599
Metal JogToJog Spacing	0
Metal Notch Spacing	1621088
Metal Parallel Run Length Spacing	3253512
Metal Short	6429817
MinHole	94
MinStep	7794204
Minimal Area	744067

## ✓ Timing & Power

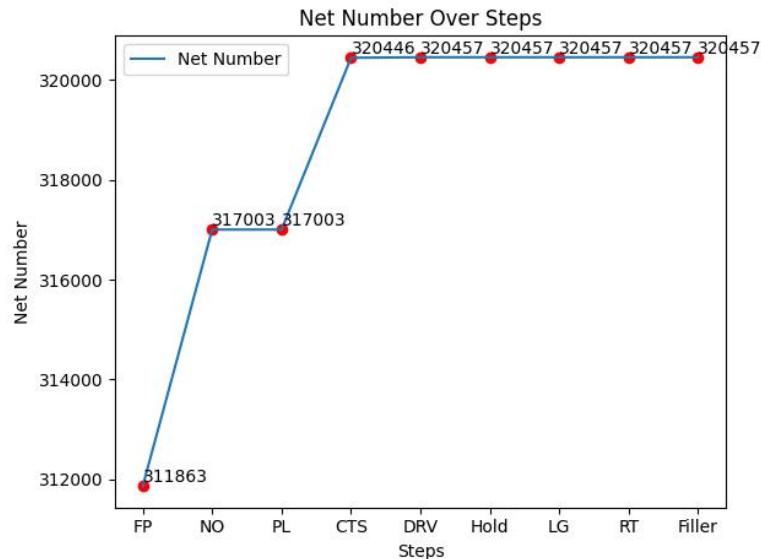
- Setup, Hold, Violations, Power
  - Placement, CTS, Fix DRV, Fix Setup/Hold, Routing

# Data Analysis

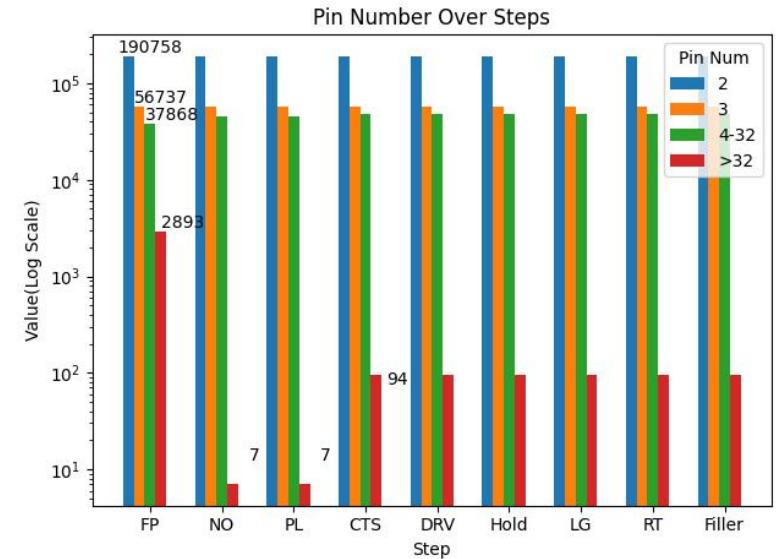
## Instance number



## Net number



## Pin number



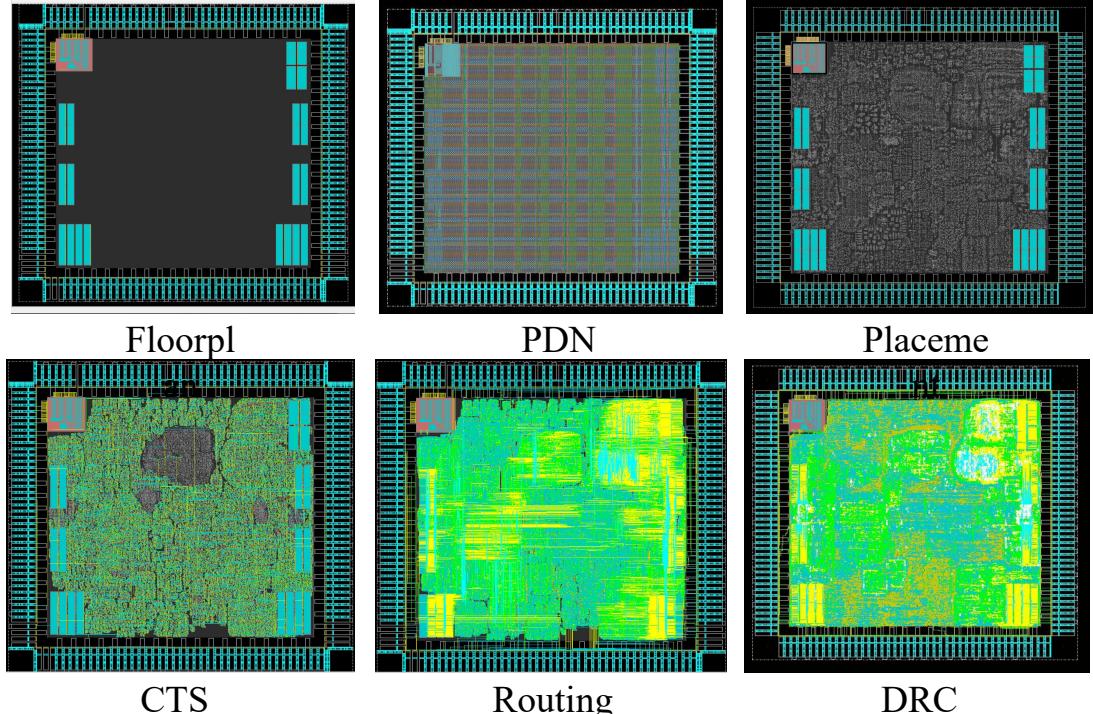
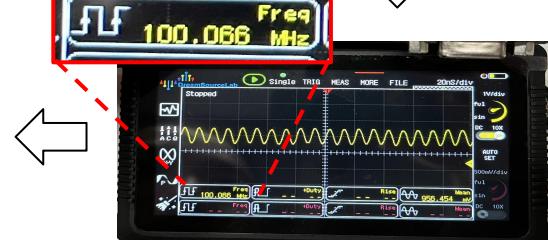
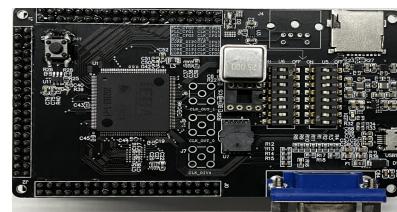
- ✓ Floorplanning -> Routing, where #Inst increased by **8594**, generated by the Fix Fanout, CTS, DRV, and Fix Setup/Hold.
- ✓ In the Filler stage, **252361** Instances is added.

- ✓ The total increase #nets in backend physical design flow is **8324**, primarily contributed by netlist optimization and clock tree synthesis stages, which are 5140 and 3284, respectively.

- ✓ Most of the nets consist of **2** pins and **3** pins.
- ✓ The number of nets with excessive fanout (Pins in Net) was optimized in the NO stage, reducing from **2893** to **7**.
- ✓ In the CTS stage, **87** new clock nets with excessive fanout were generated (Pin Number > 32).

# Example Design: ysyx-04-01

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

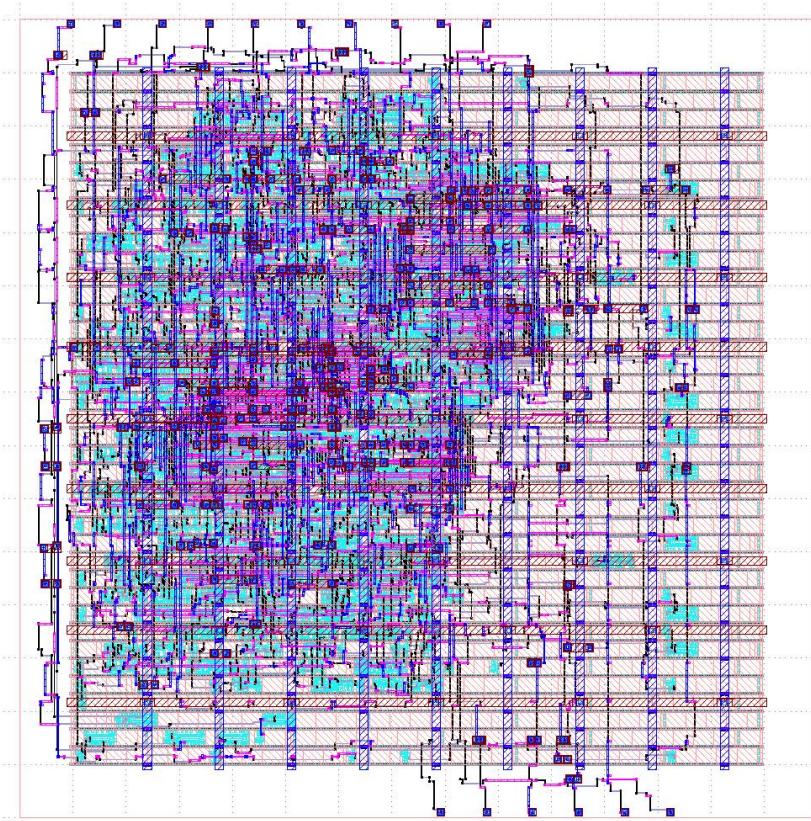


part metrics	iPL (place)	iCTS	iTO	iRT (route)
#inst	1043440	1057291	1057549	1057549
#net	1015532	1029383	1029641	1029641
utilization	0.563929	0.570644	0.570768	0.570768
HPWL	34108823398	35042653984	35044866877	50157263995*
STWL	46195026227	46580611921	46581568292	
frequency	245.245	238.226	241.386	224.254
#DRC	0	0	0	233335

\* Total wirelength after routing

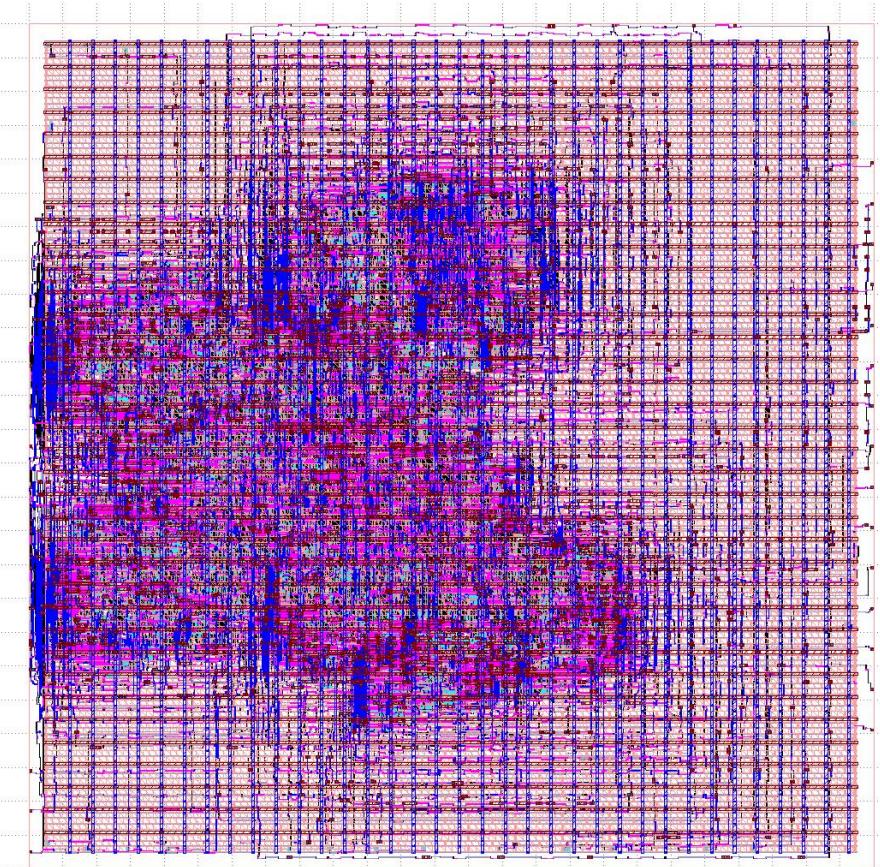
# Example Design: from other users

- gcd & APU



gcd, skywater 130nm

Area: 0.15mm × 0.15 mm



APU, skywater 130nm

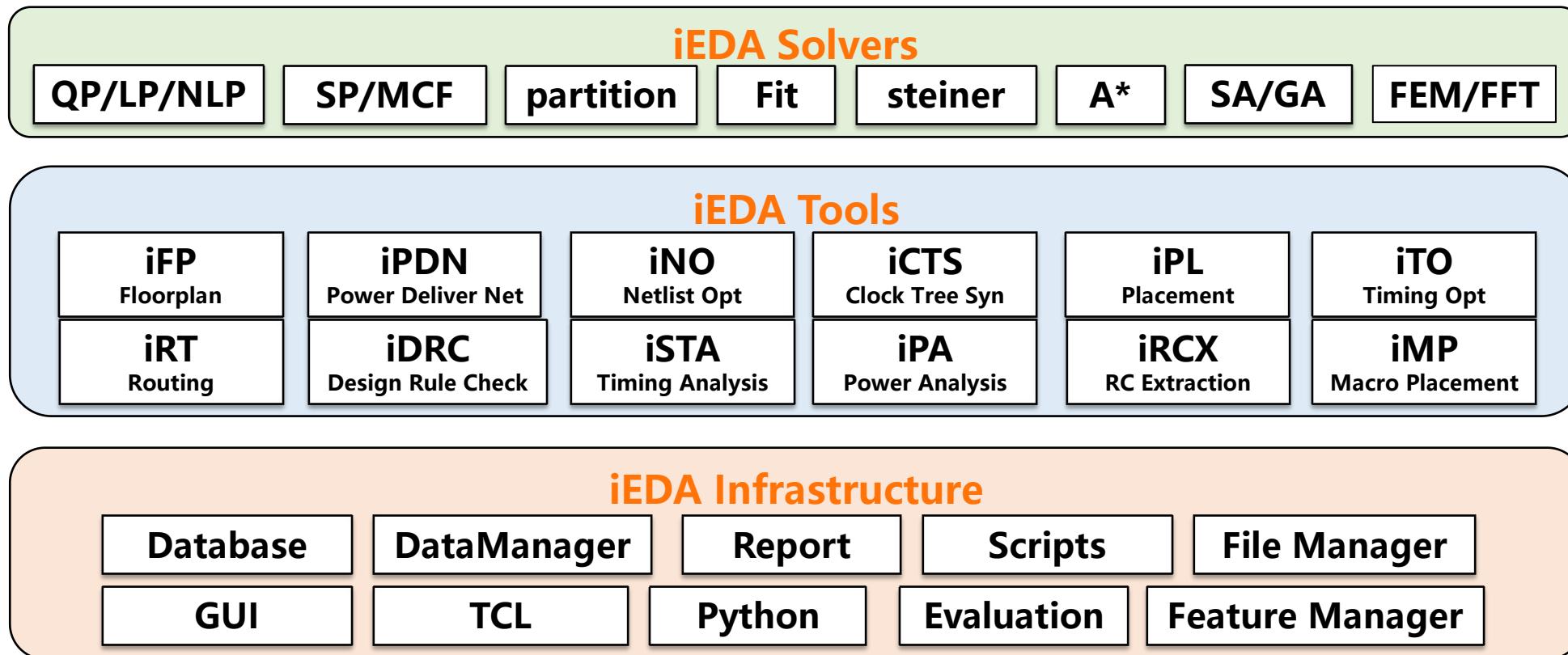
Area: 0.45mm × 0.45 mm

# iEDA & OpenROAD

Flow		PDK	时钟_MHz	placement		routing			sta		
				GP_original HPWL (um)	DP HPWL(PL_eval)	droute run time (s)	total wire length	total vias	setup_slack (max)	hold_slack (min)	suggest freq(MHz)
metrics	design	PDK	时钟_MHz	GP_original HPWL (um)	DP HPWL(PL_eval)	droute run time (s)	total wire length	total vias	setup_slack (max)	hold_slack (min)	suggest freq(MHz)
openroad	APU	sky130	50	108025.8	304807.6	900	348193	42309	14.28	0.44	174.8251748
	BM64	sky130	50	950721.2	1151209.6	660	1298431	118906	14.22	0.42	173.0103806
	PPU	sky130	50	799473.6	1325241.3	14220	1666739	173502	15.46	0.38	220.2643172
	aes	sky130	50	1736825.9	2234547.8	1920	2695657	280870	14.7	0.22	188.6792453
	aes_core	sky130	50	1915862.8	2353877.1	2040	2809505	271884	14.73	0.4	189.7533207
	blabla	sky130	50	2162081.7	2401241.7	1320	2651252	226526	9.84	0.38	98.42519685
	caravel_upw	sky130	50	35240.4	60479.9	180	66621	7595	17.82	0.4	458.7155963
	gcd	sky130	50	10958.5	23153.9	60	25345	3798	16.73	0.44	305.8103976
	picorv32a	sky130	50	955226.7	1458821.9	2040	1659581	177160	8.78	0.38	89.12655971
	s44	sky130	50	3153.7	6408.8	1020	7149	1220	19.25	0.42	1333.333333
iEDA	salsa20	sky130	50	2014421.6	2231403.5	1140	255535	262945	9.93	0.41	99.30486594
	APU	sky130	50	101311.635	108052.855	59.15	153766.992	39682	15.81	0.364	238.638
	BM64	sky130	50	724102.907	734325.379	167.71	814055.014	183714	15.73	0.386	234.199
	PPU	sky130	50	798854.543	814966.702	236.65	1133497.302	141243	16.185	0.354	262.124
	aes	sky130	50	1787923.281	1804659.776	456.84	2447231.105	284325	15.856	0.261	241.341
	aes_core	sky130	50	1869162.753	1880282.585	417.38	2360070.252	261450	15.934	0.366	245.913
	blabla	sky130	50	1915513.912	1935383.056	305.18	2004497.857	192730	12.203	0.341	128.25
	caravel_upw	sky130	50	34144.56	35161.52	1.17	50311073	15885	18.313	0.353	592.825
	gcd	sky130	50	11282.203	11774.379	0.78	16126254	9534	17.735	0.403	441.47
	picorv32a	sky130	50	928885.74	951888.682	236.62	1093014.143	163736	13.825	0.341	161.933
Ratio	s44	sky130	50	3008.324	3292.983	0.27	4641801	2764	19.421	0.388	1727.05
	salsa20	sky130	50	1895922.266	1938562.099	429.95	2331960.542	267963	12.629	0.384	135.676
	APU	sky130	50	1.066272398	2.820912043	15.21	2.264419662	1.0662013	0.903225806	1.208791209	0.732595709
	BM64	sky130	50	1.312964208	1.56771049	3.94	1.59501628	0.647234288	0.904005086	1.088082902	0.738732363
	PPU	sky130	50	1.000774931	1.626129383	60.09	1.470439318	1.22839362	0.955205437	1.073446328	0.840305799
	aes	sky130	50	0.97142082	1.23821001	4.20	1.101513051	0.987848413	0.927093845	0.842911877	0.781795241
	aes_core	sky130	50	1.024984473	1.251874117	4.89	1.190432784	1.039908204	0.924438308	1.092896175	0.771627855
	blabla	sky130	50	1.128721481	1.240706171	4.33	1.322651451	1.175354122	0.806359092	1.114369501	0.767447929
	caravel_upw	sky130	50	1.032094132	1.720059315	154.50	0.001324182	0.478124016	0.973079233	1.133144476	0.773779102
	gcd	sky130	50	0.971308529	1.966464643	77.09	0.001571661	0.398363751	0.943332394	1.091811414	0.692709352
Average	picorv32a	sky130	50	1.028357589	1.532555148	8.62	1.518352723	1.081985635	0.635081374	1.114369501	0.55039158
	s44	sky130	50	1.048324582	1.946198933	3,784.06	0.001540135	0.441389291	0.991195098	1.082474227	0.772029376
	salsa20	sky130	50	1.062502211	1.15106114	2.65	0.10957947	0.981273534	0.786285533	1.067708333	0.731926545
				1.058884123	1.641989218	374.51	0.961530974	0.866006925	0.88630011	1.082727813	0.741212805

# Functional Shelf

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



# R & D EDA Tools or Algorithms

## ● Min Wirelength Model

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

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

## ● Nesterov Method Or Conjugate Gradient

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

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

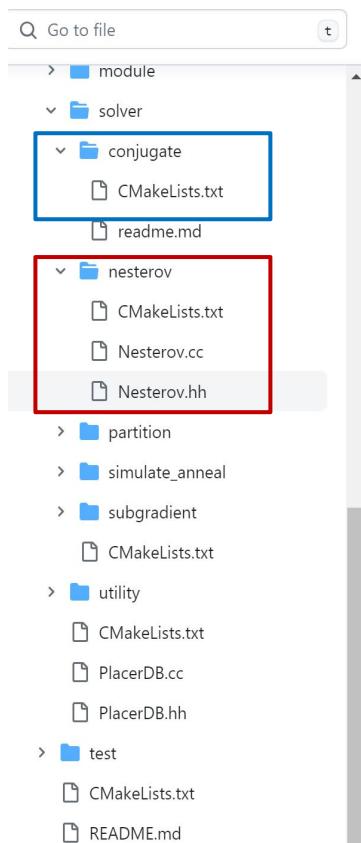
$$x_{k+1} = x_k + \alpha_k p_k$$

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

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

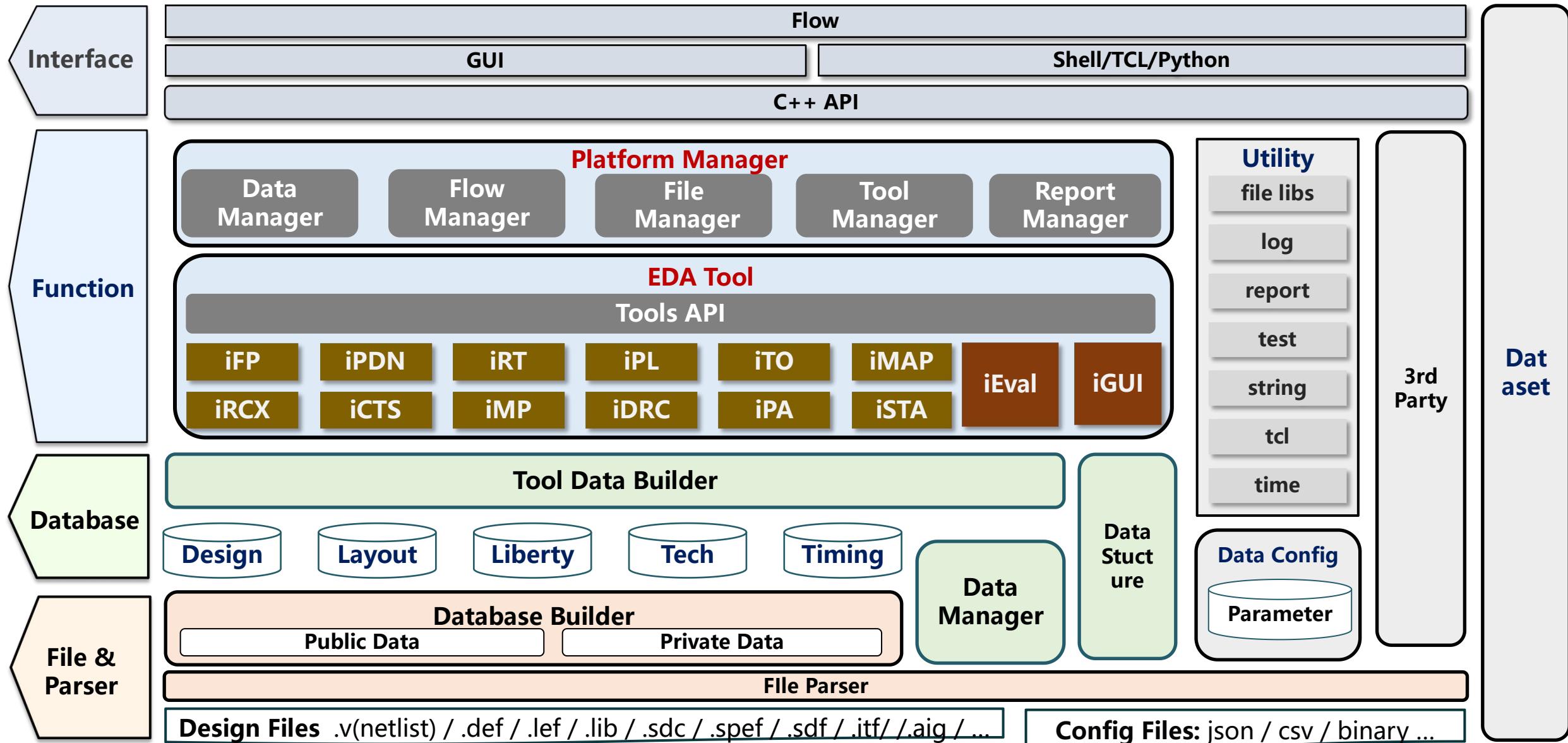
$$p_{k+1} = -r_{k+1} + \beta_{k+1} p_k$$

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

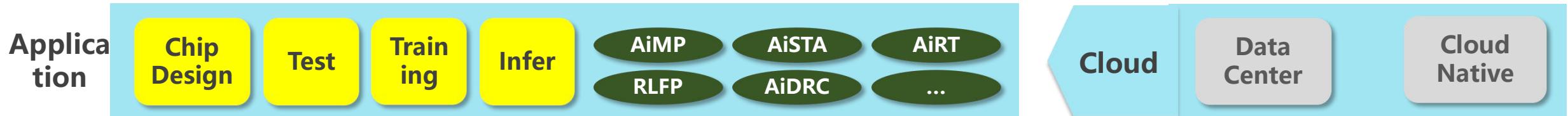


```
38 // class Nesterov
39 {
40     public:
41     Nesterov();
42     Nesterov(const Nesterov& other) = delete;
43     Nesterov(Nesterov&& other) = delete;
44     ~Nesterov() = default;
45
46     // getter.
47     int get_current_iter() const { return _current_iter; }
48     const std::vector<Point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

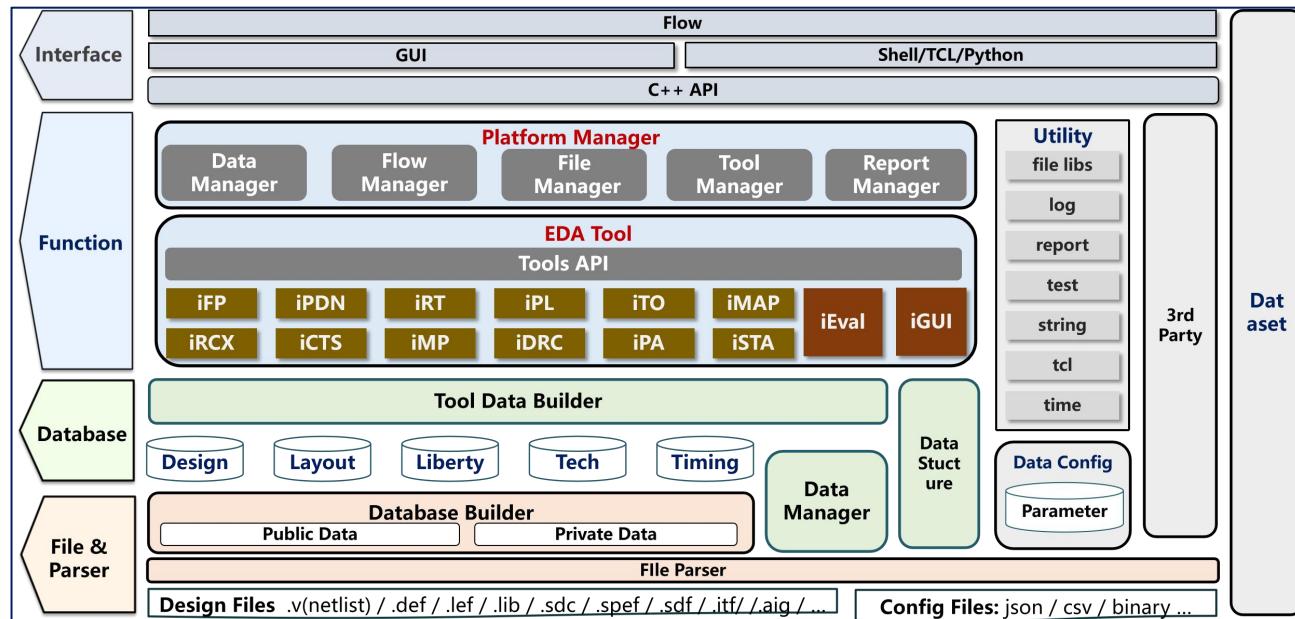
# iEDA Infra.: Evolution – System



# iEDA Infra.: Evolution – Engine



iEDA  
Engine



# Conclusions and Future works

- **Conclusions**

- The iEDA infrastructure
- Its features
- How to use to design chip from netlist to GDS

- **Future works**

- More Solvers
- High-performance
- More APIs

# iEDA Tutorial Agenda

---

- **Part 0:** iEDA Overview (**Xingquan Li**)
- **Part 1:** iEDA Infrastructure (**Zengrong Huang**)
- **Part 2:** iPL: Placement Tool and Its Technology (**Shijian Chen**)
- **Part 3:** iCTS: Clock Tree Synthesis Tool and Its Technologies (**Weiguo Li**)
- **Part 4:** iRT: Routing Tool and Its Technologies (**Zhisheng Zeng**)
- **Part 5:** iSTA: Static Timing Analysis Tool and Its Technologies (**Simin Tao/He Liu**)
- **Part 6:** iPA: Power Analysis Tool and Its Technologies (**Siming Tao**)

 OSCC-Project / iEDA

iEDA Public

master 3 Branches 0 Tags

Go file Add file Code

OXharry and gitee-org 11 Merge iPD 57a6b6a · last week 2,107 Commits

.gitee init repo of OSCC/EDA last year

cmake feature:support IR rust and C operation 2 weeks ago

docs finish iPL Timing-driven placement 2 months ago

scripts select SPEF file for tcl script last week

src Merge -project/EDA last week

.clang-format !1 up last year

.clang-tidy !1 up last year

.dockernignore update last month

.gitignore feature 6 months ago

.gitmodules update src/third\_party/mt-kahypar submodule. last month

CMakeLists.txt feature:add rust cmake 27 days ago

Dockerfile update dockerfile last month

LICENSE fix typo from LICENSE 7 months ago

README-CN.md Merge branch 'master' of gitee.com:oscc-project/iEDA into ... last month

README.md Merge branch 'master' of gitee.com:oscc-project/iEDA into ... last month

build.sh Merge branch 'master' of gitee.com:oscc-project/iEDA into ... last month

About

No description, website, or topics provided.

Readme View license Activity Custom properties 231 stars 3 watching 21 forks Report repository

Releases

No releases published Create a new release

Packages

No packages published Publish your first package

Contributors 25

+ 11 contributors

Languages



# Thanks

Xingquan Li  
lixq01@pcl.ac.cn