iEDA Tutorial Agenda

- **Part 0:** iEDA Overview **(Xingquan Li)**
- **Part 1:** iEDA Infrastructure **(Zengrong Huang)**
- **Part 2:** iPL: Placement Tool and Its Technologies **(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 **(He Liu)**
- **Part 6:** iPA: Power Analysis Tool and Its Technologies **(Simin Tao)**

iPL-1.0
An Open-source Placement Tool in iEDA

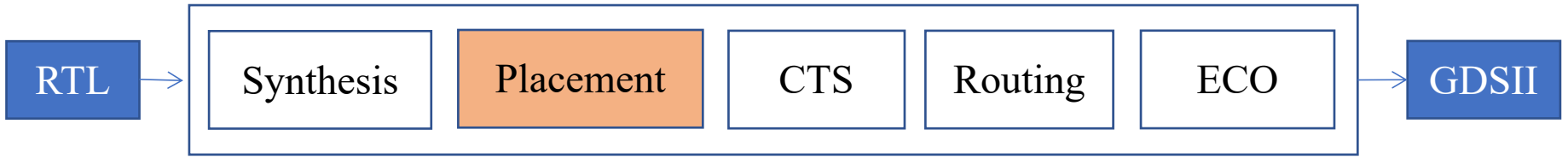
Shijian Chen[1, 3], Yihang Qiu[2], Xueyan Zhao[3], Wenrui Wu[4], Xingquan Li[1]

[1]Peng Cheng Laboratory
[2]University of Chinese Academy of Sciences
[3]Institute of Computing Technology, Chinese Academy of Sciences
[4] Shenzhen University
Email: chenshj@pcl.ac.cn, lixq01@pcl.ac.cn

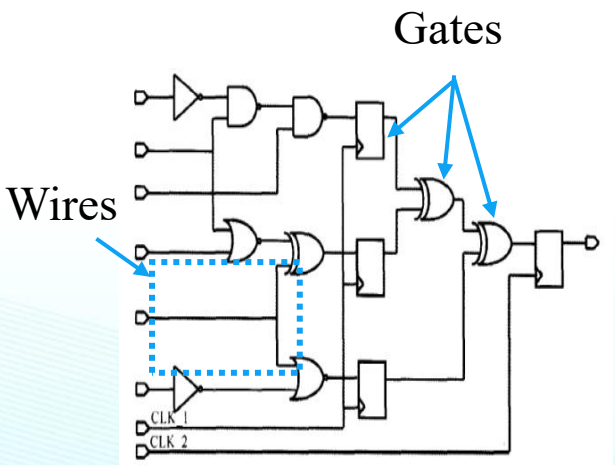
Outline

Background

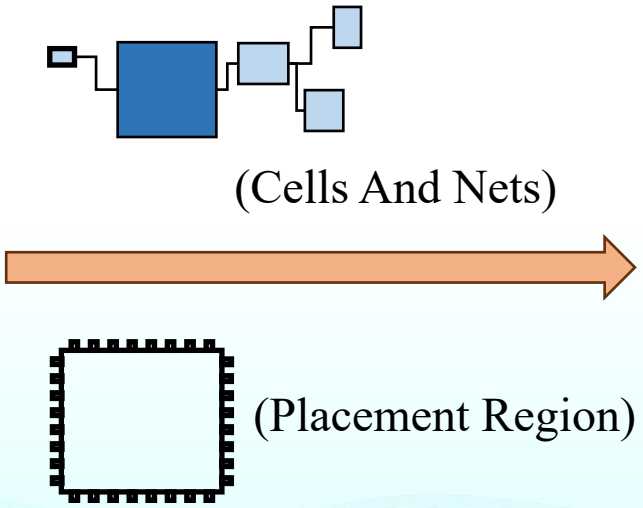
- Placement in physical design

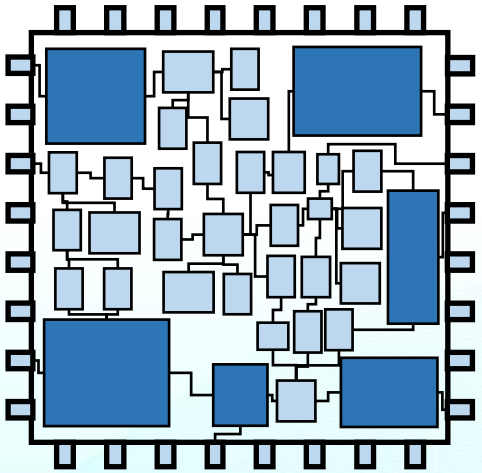


- What placement means?



(Netlist)

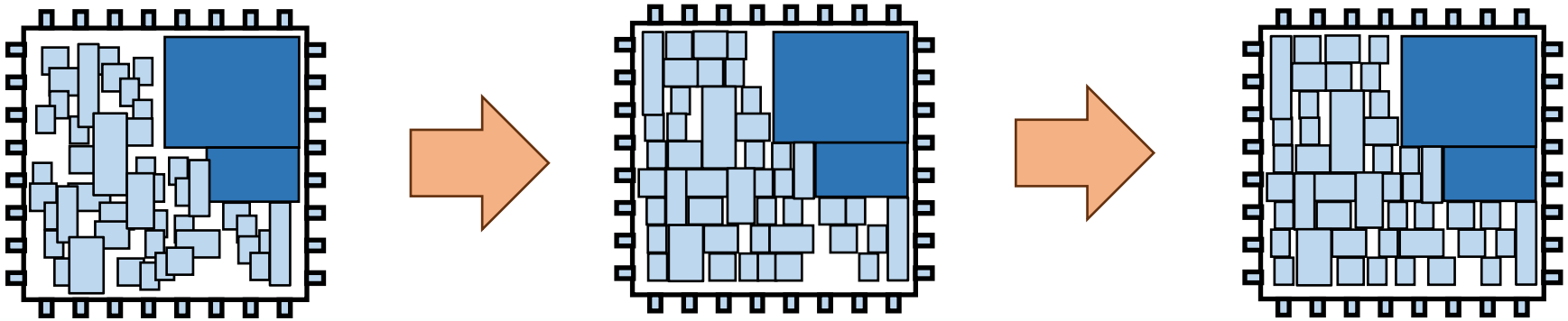




(Placement Result)

Background

- Three main steps in placement
  - Due to the complexity (NP-hard) of placement, we usually divide the placement problem into three subproblems.



Global Placement: Cells spread out to the appropriate location, ignoring cell overlaps

Legalization: Put the cells on the row/site and eliminate the overlap between the cells

Detail Placement: Adjust the elements locally to optimize the design goals

Background

- Many types of placement problems arise from contests

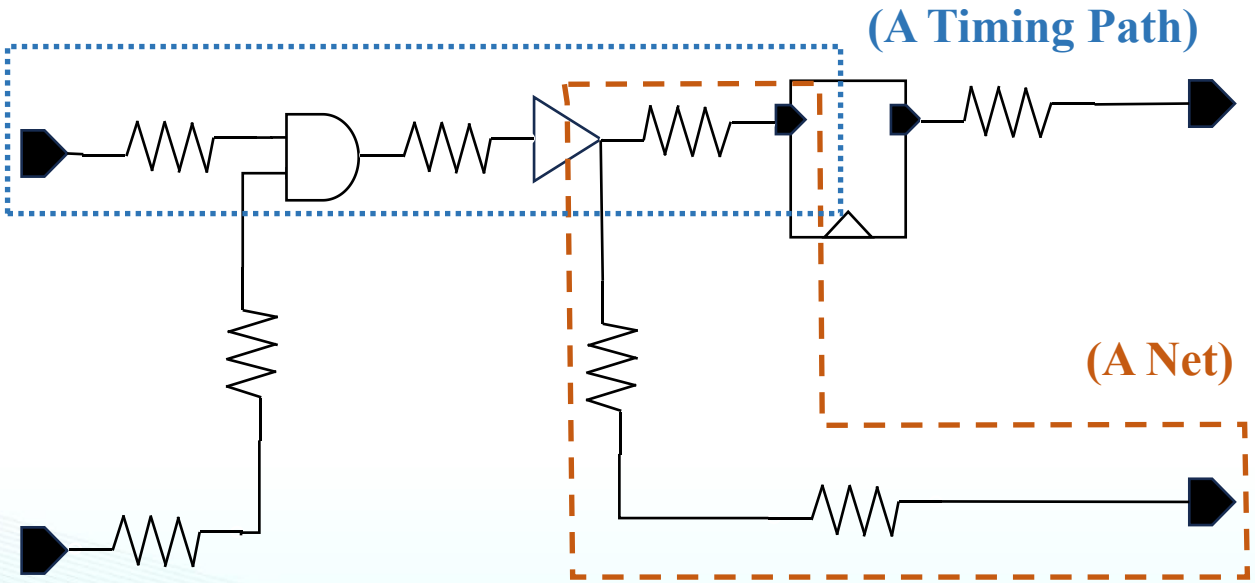
Contests	Topics
ISPD'2005	Placement
ISPD'2006	Placement
ISPD'2011	Routability-Driven Placement
DAC'2012	Routability-Driven Placement
ICCAD'2012	Design Hierarchy Aware Routability-Driven Placement
ICCAD'2013	Detailed Placement
ISPD'2014	Detailed Routing-Driven Placement
ICCAD'2014	Incremental Timing-Driven Placement
ISPD'2015	Blockage-Aware Detailed Routing-Driven Placement
ICCAD'2015	Incremental Timing-Driven Placement
ICCAD'2017	Multi-Deck Standard Cell Legalization
ICCAD'2022	3D Placement with D2D Vertical Connections
ICCAD'2023	3D Placement with Macros

- **Wirelength-Driven Placement**
  - ISPD'2005, ISPD'2006
- **Routablity-Driven Placement**
  - ISPD'2011, DAC'2012, ICCAD'2012, ISPD'2014, ISPD'2015
- **Detail Placement**
  - ICCAD'2013
- **Timing-Driven Placement**
  - ICCAD'2014, ICCAD'2015
- **Legalization**
  - ICCAD'2017
- **3D Placement**
  - ICCAD'2022, ICCAD'2023

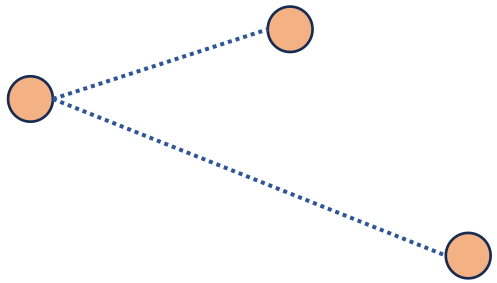
Outline

Preliminaries

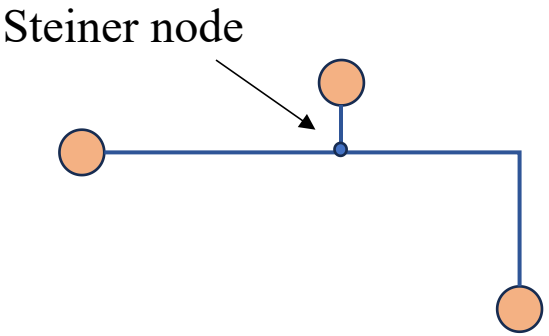
- Wire connection between cells greatly affects wire length, timing, power…



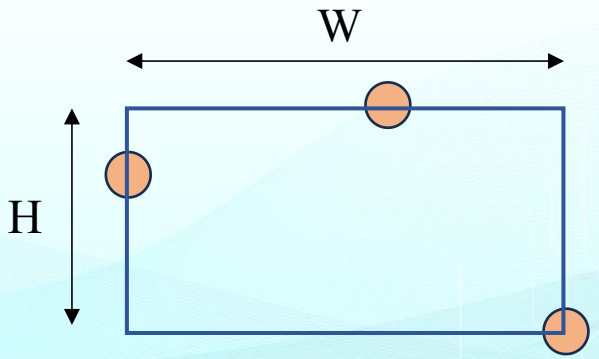
(Figure to illustrate timing path and net)



(A Net)



(RSMT)

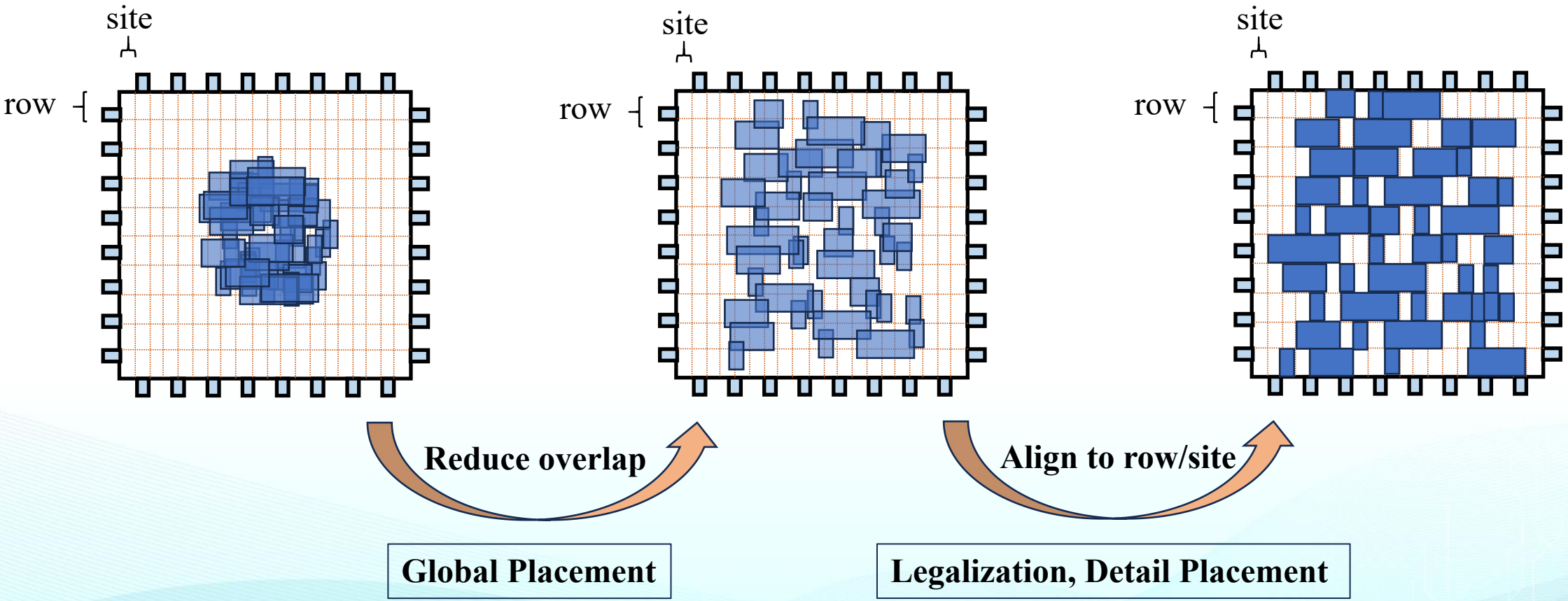


(HPWL)

- *RSMT: Rectilinear Steiner Minimal Tree*
- *HPWL: Half Perimeter Wirelength*

Preliminaries

- Placement needs to promise that cells do not overlap and are assigned to row/site

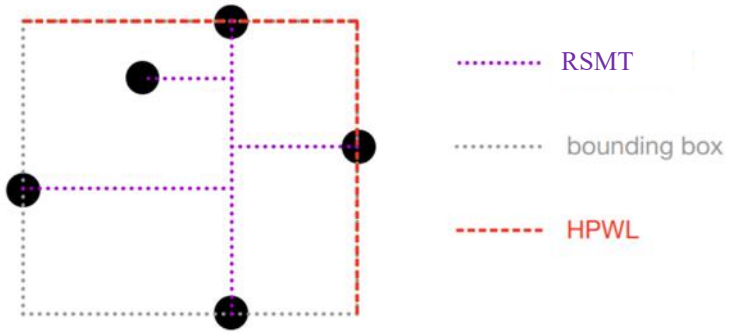


Outline

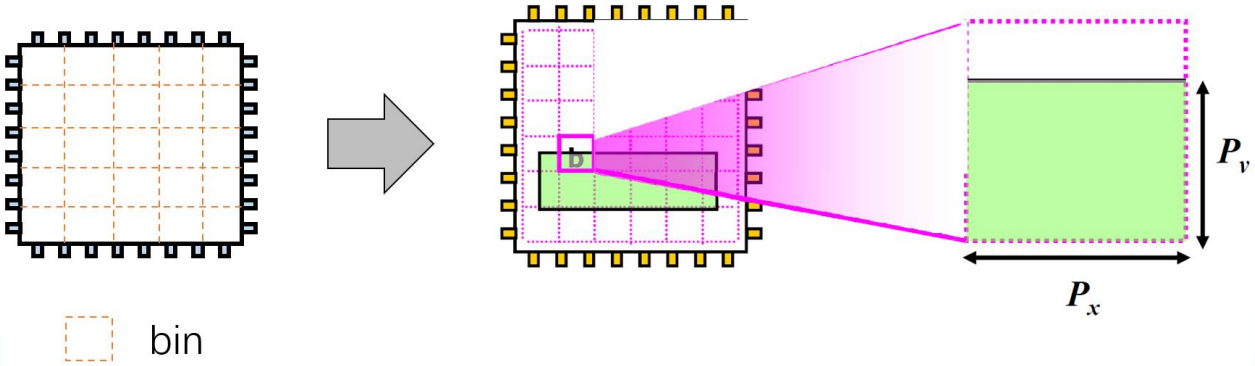
Global Placement

– The global placement target is HPWL, Written as $HPWL_e(v)$, The constraint is the density of each mesh after the bin has been divided, Written as $\rho_b(v) \leq \rho_t, \forall b \in Bin$ 。

$$\min_{v} \sum_{e \in E} HPWL_e(v) \quad s.t. \quad \rho_b(v) \leq \rho_t, \forall b \in Bin$$



(HPWL)



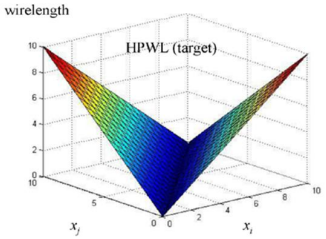
(Density Model under bins)

– The global placement objectives and constraints can be connected using the penalty method or the (augmented) Lagrangian method to transform into an unconstrained optimization problem.

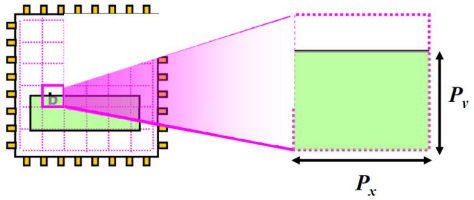
Global Placement

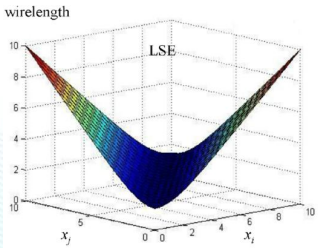
– The wire length and density models mentioned above are non-smooth, and if the gradient method is used, a **smoothing approximation** of the two is required. After the gradient vector is obtained, the solver (Nesterov, CG, ADMM, SGD, etc.) is used to solve the global placement problem iteratively.

Smoothing is evaluated for wire length

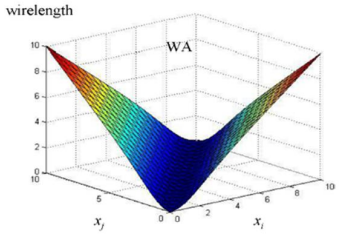


Smoothing is evaluated for density

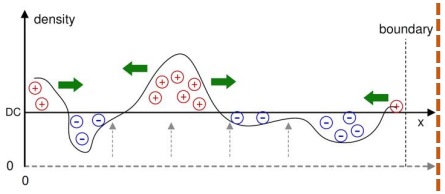




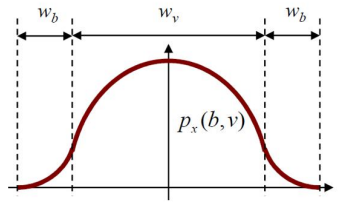
(LSE WL model)



(WA WL model)

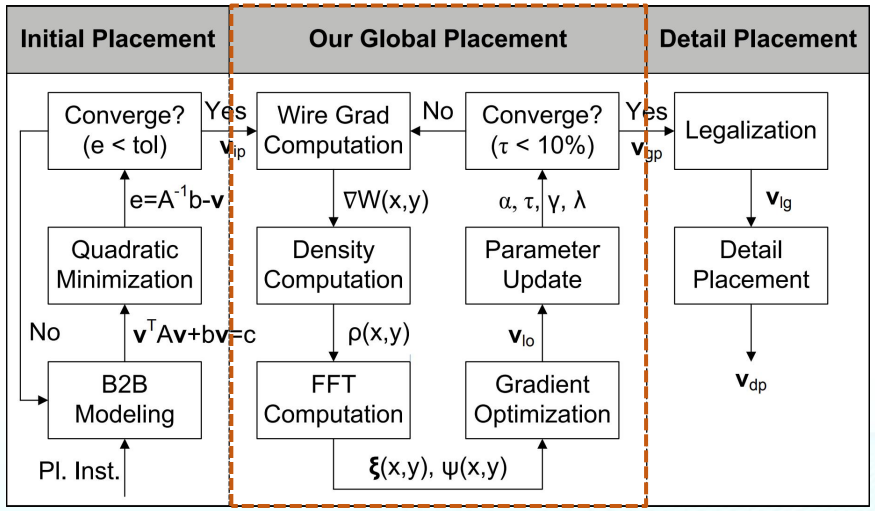


(Electric Density model)



(Bell-shape Density model)

Global placement flow



The WAWL model and the electric density model have a good smoothing approximation

- *Lu J, Chen P, Chang C C, et al. ePlace: Electrostatics-based placement using fast fourier transform and Nesterov's method[J]. ACM Transactions on Design Automation of Electronic Systems (TODAES), 2015, 20(2): 1-34*

Legalization, Detail Placement

– After the global placement phase, the cells should be aligned with the rows and sites, oriented correctly towards the power rail, and arranged without overlapping. possible methods:

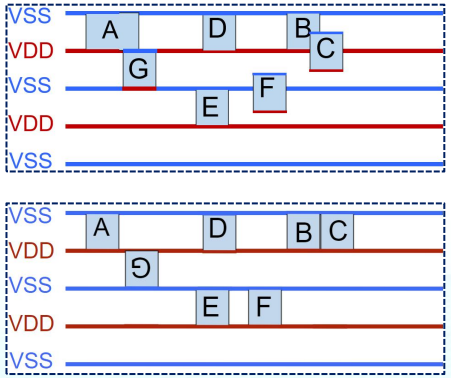
- Divide the sub-elements (profile, height, and width) to solve the problem of **minimum cost and maximum flow**
- In the two-stage method, the cells are assigned to the rows first, and the **shortest path** is found on the created map
- **Tetris, Abacus, QP, LP**

– Detailed Placement needs to **maintain the legalization results.**

Local optimization of indicators (wire length, timing, congestion) is performed.

Possible methods:

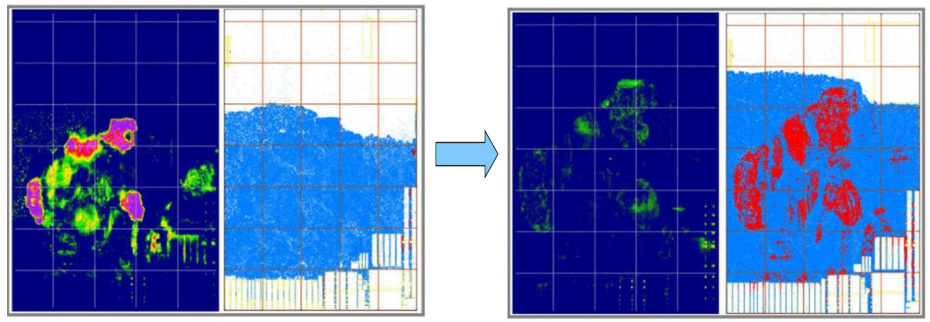
- In-line **cell shift**
- **Cell swapping**, selecting independent sets for matching
- Solve the **optimal branch delimitation** results for a small number of local cells



(Legalization)

Routablity-driven Placement

- Challenge
  - Cells **change greatly** in the **early stage** of placement
  - Difficult to **balance** performance and accuracy of evaluation
  - Increasing wire length and timing **damage** issues
- Problem description
  - Common evaluation metric: **Total Overflow (TOF)** , **Maximum Overflow (MOF)**
  - Routablity-driven placement adjusts the cell position **according to the congestion map**
- Methodology
  - Evaluation: Static, probabilistic, constructed, and network methods. At present, the **commonly used methods** are the **probability** method and the **construction** method
  - Optimization: Adjust the density to **indirectly optimize** congestion, and directly integrate congestion items to **participate in optimization**



(a) (b)

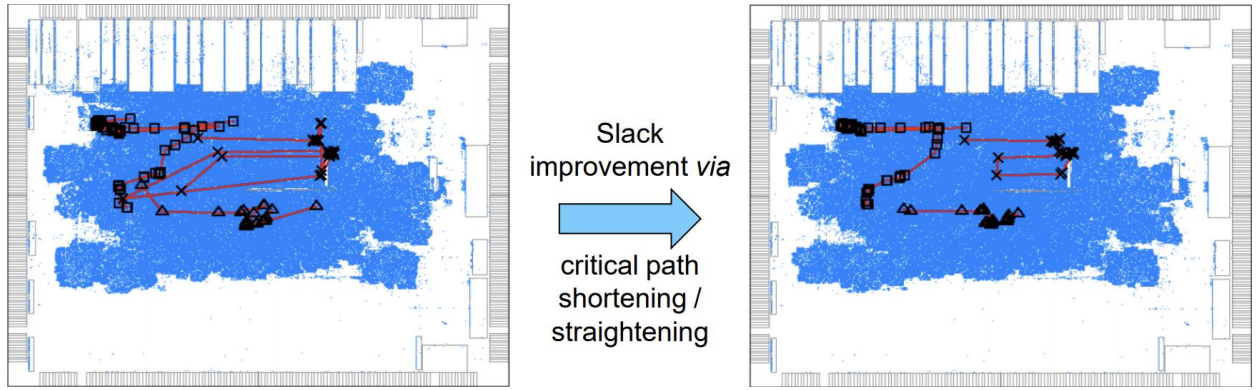
(a) Wirelength-driven Placement

(b) Routablity-driven Placement

- *N. Viswanathan, et al. The ISPD-2011 routability-driven placement contest and benchmark suite[C]. ISPD, 2011.*

Timing-driven Placement

- Challenge
  - Cells **change greatly** in the **early stage**
  - May cause **new timing violations** to appear
  - Cells aggregate, **affecting the congestion**
- Problem description
  - **Total Negative Slack (TNS), Worst Negative Slack (WNS)**
  - Get timing metrics from the timer to adjust cell coordinates
- Methodology
  - **Net-based approach**: Annotate the timing critical path information to the net level to **empower the critical net**, which can be easily adapted to all wire-length driving placements.
  - **Path-based approach**: According to the timing, a series constraint is added, and the position of the cells is updated by solving the **mathematical programming.**
  - **Differentiable methods**: Smoothing process of timing propagation process, and the timing metrics are added to the target of global placement.

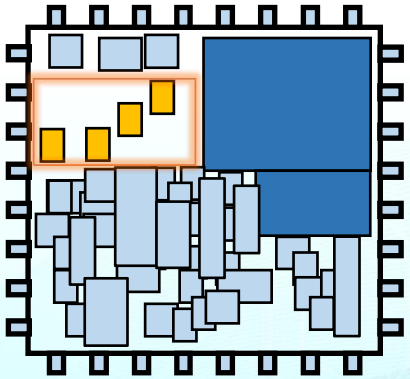


(Timing-driven Placement)

- *Kim M C, Hu J, Li J, et al. ICCAD-2015 CAD contest in incremental timing-driven placement and benchmark suite[C] IEEE, 2015: 921-926.*

Region-constraint Placement

- Motivation
  - Modern placement requires specific cells to be placed in isolated voltage areas to **improve performance**.
- Problem description
  - A **Region area** (usually consisting of one or more rectangular sub-regions) excludes the placement of **specified cells**, and the **rest of the cells cannot be placed within the region**. Region constraints need to be considered in the global placement stage, otherwise, subsequent legalization and detailed placement will lead to serious quality loss.
- Methodology
  - **NTUplace4dr**: Cell clustering, new wire length model, and new virtual net
  - **Eh?Placer**、**RippleDR**: Look-ahead legalization and optimization of upper and lower bounds
  - **DREAMPlace3.0**： Multi-electric field system placement



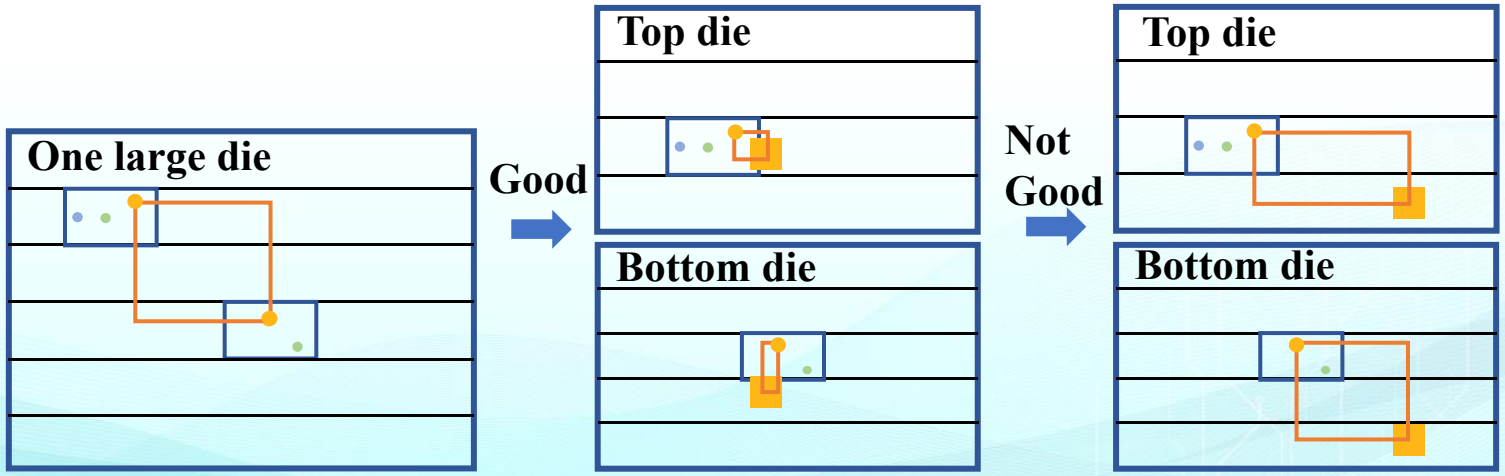
(Region-constraint Placement)

2.5D,3D Placement

– Motivation
  – The 3D integration approach is considered to be a promising way to **further address connectivity bottlenecks**, and the placement in 3D is critical to the quality of the 3D integration, as it needs to determine not only the **horizontal position but also the layer** in which it is located.

– Problem description
  – In the 3D integration, different dies may use different processes, and the size of the same cell is different when it is located on different layers, which **introduces new constraints for partitioning**. At the same time, the cells in a net may belong to different dies, and the middle layer is connected through the terminal, this **net model also poses new challenges** to the placement.

– Methodology
  – **Double-level planning**
  – **Netlist division**
  – **Legalization of MIV**



(2.5D,3D Placement)

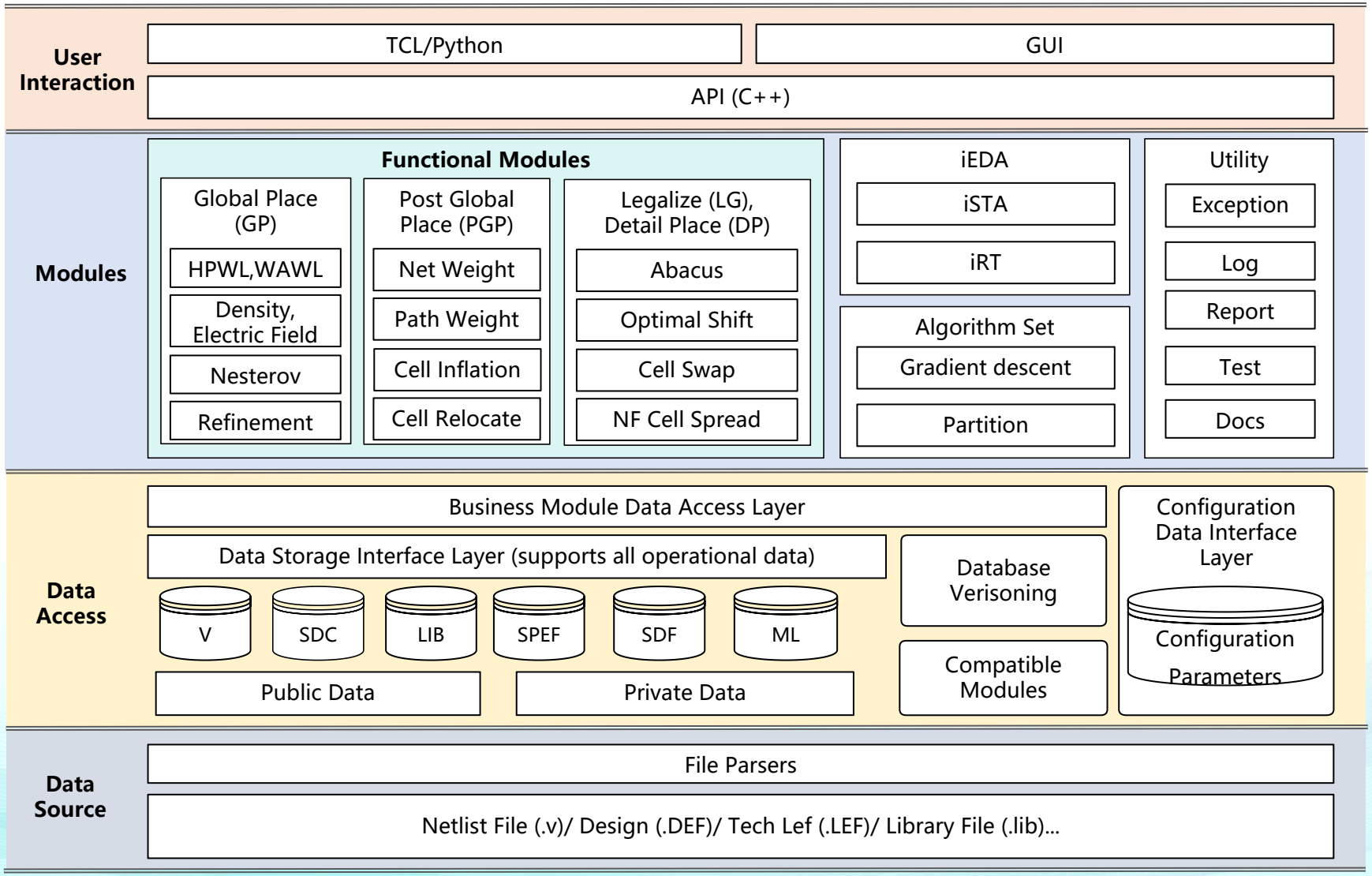
Other Placement

- New density expressions and calculations (functions, equations, or nets)
- Jointly optimize cell location/size and buffer
  Precise control of physical variables
  Differentiable Optimization Metrics (WL, Timing, Congestion)
  Second-order optimization method
  Optimize orientation learning
  Congestion estimates
  Timing estimation
  DRC estimation
  Power consumption estimation
  Hardware acceleration technology

Outline

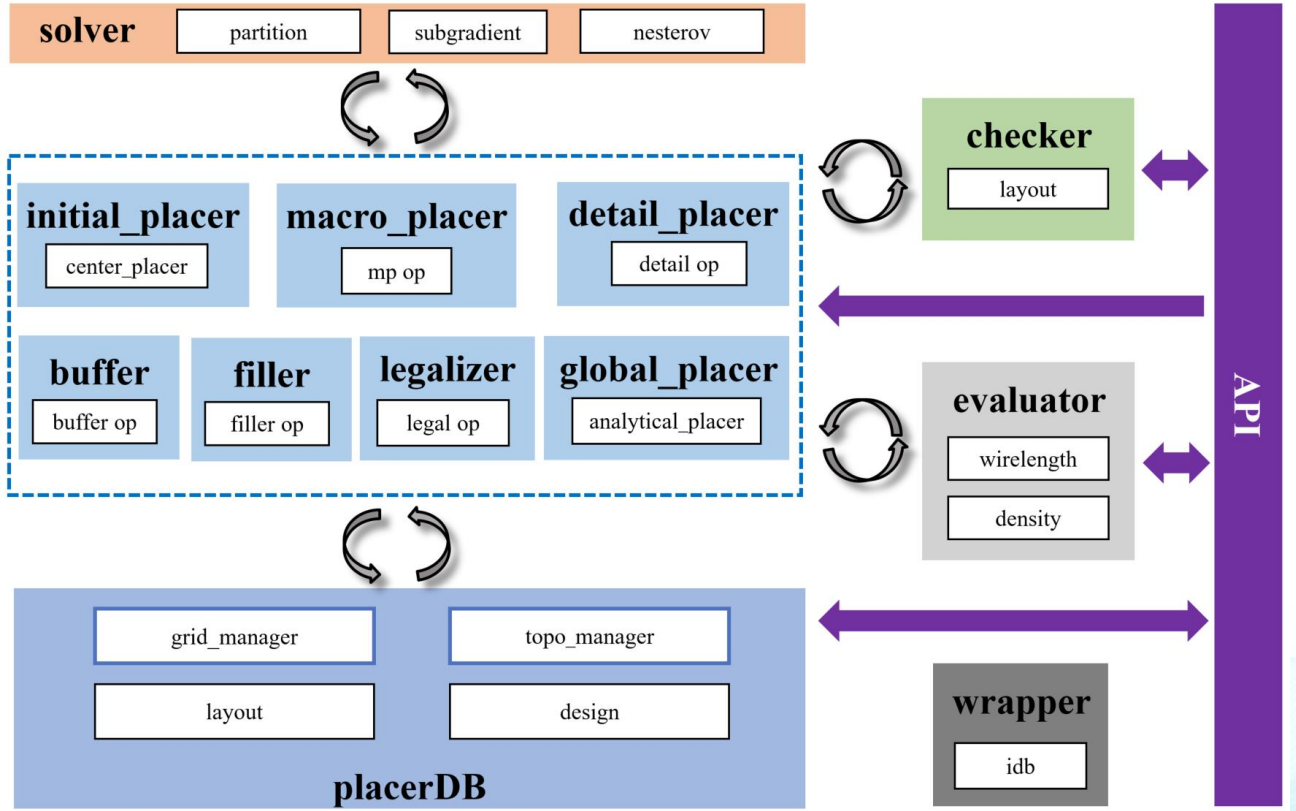
iPL Tool Framework

- **Global Placement**
  - WAWL, Electric Field, Nesterov, Refinement
- **Legalization**
  - Abacus, Incr Legalize
- **Detail Placement**
  - Optimal Shift, Cell Swap,
- **Timing-driven Placement**
  - Net Weight, Path Weight, Cell Balancing, Load Reduction
- **Routablity-driven Placement**
  - Cell Inflation, Network Flow Cell Spreading



iPL Tool Structure

- **PlacerDB：** Maintain layout and design data required for placement
- **Operator：** Extract the layout data for operation
- **Solver：** A collection of mature solver tools to aid in placement
- **Checker：** Perform violation checks, function detection, and report output on the current placement
- **Evaluator：** Evaluate the placement metrics
- **Wrapper：** Placement data is read from the data source, and the Placement result is written back to the data source
- **API：** The interface between the iPL and the external interface



(iPL Tool Structure)

iPL Basic Feature

- **iPL 1.0 completes the basic functions**
  - **Data reading** supports extracting and processing placement-related data from iDB, and configuration files support **json reading**
  - Meet the needs of the placement process:
    - **Global Placement:** Support HPWL, STWL, WAWL wire length + electric field density (gradient) evaluation, and use Nesterov method
    - **Legalization:** Using the Abacus method, full and incremental legalization modes are supported
    - **Detailed Placement**: Single-row shift of cells (optimal layout within rows), global/vertical swapping of cells, local reordering of cells, and movement within rows to eliminate density areas
    - **Filler:** You can specify the filler type to fill the blank space in the layout area
    - **Placement check:** cells placed within the layout area, aligned Row/Site, aligned Power rail, and whether the cells overlap

iPL Interconnecting Feature

- **iPL 1.0 completes the basic functions**
  - iPL 1.0 expands the **interaction** between the placer and the **iEDA toolchain**
  - Finish:
    - Invoke **eGR（Early Global Routing）** on assessment of the state of the placement
    - An **RC tree** is built based on the pre-routing information, and the timing information of the current placement state is evaluated using **iSTA**
    - Allows to return **a collection of legal spaces** that are not occupied in a specified area
    - Supports the read-in and **incremental legalization** of new cells
    - **Buffers insertion** for long wire optimization

iPL Tool API

Method	SubMethod	Type	Argument	Return	Description
initAPI	*	action	pl_json_path, idb_builder	self	Initialize the iPL
runFlow	runGP	action	void	self	Run the global placement
	runBufferInsertion	action	void	self	Run Buffer Insertion
	runLG	action	void	self	Run legalization
	runDP	action	void	self	Run the detailed placement
	writeBackSourceDataBase	action	void	self	Written back to the data source
runIncrLG	*	action	inst_list	self	Run incremental legalization
updatePlacerDB	*	action	void/inst_list	self	Update data from the data source
obtainAvailable WhiteSpaceList	*	action	row_range, site_range	Rectangle list	Gets the available placement space in the specified area (for insert cells)
checkLegality	*	accessor	void	bool	Check the legitimacy
isSTAStarted	*	accessor	void	bool	Check if STA is started
isPlacerDBStarted	*	accessor	void	bool	Check if PlacerDB is initialized
isAbucasLGStarted	*	accessor	void	bool	Check if the Abucas has been launched

iPL Tool API

Method	SubMethod	Type	Argument	Return	Description
reportPLInfo	reportHPWLInfo	accessor	feed	self	Report HPWL information
	reportSTWLInfo	accessor	feed	self	Report STWL information
	reportLongNetInfo	accessor	feed	self	Report long net information
	reportLayoutInfo	accessor	feed	self	Report Layout Violation Information
	reportPeakBinDensity	accessor	feed	self	Report the peak density of bin area
	reportTimingInfo	accessor	feed	self	Report timing information
	reportCongestionInfo	accessor	feed	self	Report congestion information
obtainTimingInfo	obtainPinEarly(Late)Slack	action	pin_name	value	Get slack information on the pin
	obtainPinEarly(Late)ArrivalTime	action	pin_name	value	Get the Arrival Time on the pin
	obtainPinEarly(Late)RequiredTime	action	pin_name	value	Get the Required Time on the pin
	obtainWNS/TNS	action	clk_name	value	Obtain WNS/TNS information
	updateTiming	action	void	self	Update the timing evaluation
obtainCongesion Info	obtainPinDens	action	void	value	Get Pin Density information
	obtainNetCong	action	rudy_type	value	Obtain net congestion information
	evalGRCong	action	void	value	Update the congestion assessment

iPL Tool Parameters

JSON参数	功能说明	参数范围	默认值
is_max_length_opt	是否开启最大线长优化	[0,1]	0
max_length_constraint	指定最大线长	[0-1000000]	1000000
is_timing_aware_mode	是否开启时序模式	[0,1]	0
ignore_net_degree	忽略超过指定pin个数的线网	[10-10000]	100
num_threads	指定的CPU线程数	[1-64]	8
[GP-Wirelength] init_wirelength_coef	设置初始线长系数	[0.0-1.0]	0.25
[GP-Wirelength] reference_hpwl	调整密度惩罚的参考线长	[100-1000000]	446000000
[GP-Wirelength] min_wirelength_force_bar	控制线长边界	[-1000-0]	-300
[GP-Density] target_density	指定的目标密度	[0.0-1.0]	0.8
[GP-Density] bin_cnt_x	指定水平方向上Bin的个数	[16,32,64,128,256,512,1024]	512
[GP-Density] bin_cnt_y	指定垂直方向上Bin的个数	[16,32,64,128,256,512,1024]	512
[GP-Nesterov] max_iter	指定最大的迭代次数	[50-2000]	2000
[GP-Nesterov] max_backtrack	指定最大的回溯次数	[0-100]	10
[GP-Nesterov] init_density_penalty	指定初始状态的密度惩罚	[0.0-1.0]	0.00008
[GP-Nesterov] target_overflow	指定目标的溢出值	[0.0-1.0]	0.1
[GP-Nesterov] initial_prev_coordi_update_coef	初始扰动坐标时的系数	[10-10000]	100
[GP-Nesterov] min_precondition	设置precondition的最小值	[1-100]	1
[GP-Nesterov] min_phi_coef	设置最小的phi参数	[0.0-1.0]	0.95
[GP-Nesterov] max_phi_coef	设置最大的phi参数	[0.0-1.0]	1.05
[BUFFER] max_buffer_num	指定限制最大buffer插入个数	[0-1000000]	35000
[BUFFER] buffer_type	指定可插入的buffer类型名字	工艺相关	列表[...,...]

iEDA User Guide:

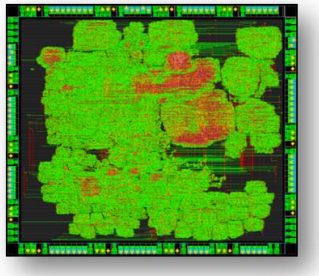
https://e.gitee.com/i-eda/repos/ieda-ipd/iEDA/blob/master/docs/user_guide/iEDA_user_guide.md

[LG] max_displacement	指定单元的最大移动量	[10000-1000000]	50000
[LG] global_right_padding	指定单元间的间距（以Site为单位）	[0,1,2,3,4...]	1
[DP] max_displacement	指定单元的最大移动量	[10000-1000000]	50000
[DP] global_right_padding	指定单元间的间距（以Site为单位）	[0,1,2,3,4...]	1
[Filler] first_iter	指定第一轮迭代使用的Filler	工艺相关	列表[...,...]
[Filler] second_iter	指定第二轮迭代使用的Filler	工艺相关	列表[...,...]
[Filler] min_filler_width	指定Filler的最小宽度（以Site为单位）	工艺相关	1
[MP] fixed_macro	指定固定的宏单元 (string macro_name)	设计相关	列表[...,...]
[MP] fixed_macro_coordinate	指定固定宏单元的位置坐标（int location_x, int location_y）	设计相关	列表[...,...]
[MP] blockage	指定宏单元阻塞矩形区域，宏单元应该避免摆放在该区域（i	设计相关	列表[...,...]
[MP] guidance_macro	指定指导摆放宏单元，每个宏单元可以设置期望摆放的区域	设计相关	列表[...,...]
[MP] guidance	指定对应宏单元的指导摆放区域（int left_bottom_x, int left_	设计相关	列表[...,...]
[MP] solution_type	指定解的表示方式	["BStarTree","SequencePair"]	"BStarTree"
[MP] perturb_per_step	指定模拟退火每步扰动次数	[10-1000]	100
[MP] cool_rate	指定模拟退火温度冷却率	[0.0-1.0]	0.92
[MP] parts	指定标准单元划分数（int）	[10-100]	66
[MP] ufactor	指定标准单元划分不平衡值 (int)	[10-1000]	100
[MP] new_macro_density	指定虚拟宏单元密度	[0.0-1.0]	0.6
[MP] halo_x	指定宏单元的halo（横向）	[0-1000000]	0
[MP] halo_y	指定宏单元的halo（纵向）	[0-1000000]	0
[MP] output_path	指定输出文件路径		"./result/pl"

iPL Tool Output

– iPL supports opensource chip “ysyx” of tape-out

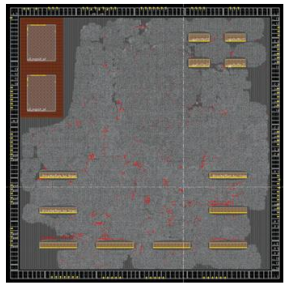
2022-02-02, 1st Tapeout



110nm node, 0.7M gates, 25MHz
(5-level pipeline, IP:Chiplink、UART、SPI)

Macro, Multi-clock, Scale increasing, Auto-design

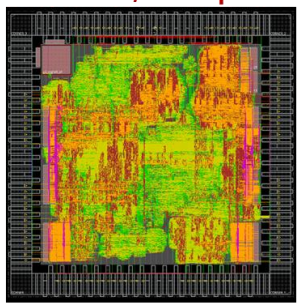
2022-08-12, 2nd Tapeout



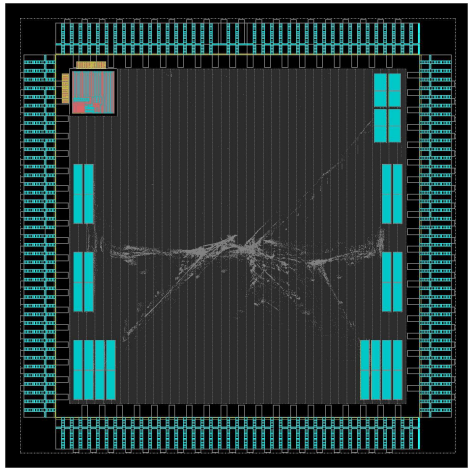
110nm node, 1.5M gates
(11-level pipeline with cache、IP: UART、VGA、PS/2、SPI、SDRAM, Two PLL on SoC, Support Linux)

110nm -> 28nm

2023-01-04, 3rd Tapeout



28nm node, 1.5M gates
(11-level pipeline with cache、IP: UART、VGA、PS/2、SPI、SDRAM, Two PLL on SoC, Support Linux)



– iPL tool outputs

```
I0730 17:36:15.176146 1203780 PlacerDB.cc:303] Design name : gcd
I0730 17:36:15.176285 1203780 PlacerDB.cc:304] Database unit : 1000
I0730 17:36:15.176290 1203780 PlacerDB.cc:305] Die rectangle : 0,0 279960,280128
I0730 17:36:15.176292 1203780 PlacerDB.cc:307] Core rectangle : 9600,9990 269760,269730
I0730 17:36:15.176295 1203780 PlacerDB.cc:309] Row height : 3330
I0730 17:36:15.176296 1203780 PlacerDB.cc:310] Site width : 480
I0730 17:36:15.176301 1203780 PlacerDB.cc:348] Core area : 67573958400
I0730 17:36:15.176304 1203780 PlacerDB.cc:349] Non place instance area : 249350400
I0730 17:36:15.176307 1203780 PlacerDB.cc:350] Place instance area : 6638155200
I0730 17:36:15.176309 1203780 PlacerDB.cc:353] Utilization(%) : 9.85992
I0730 17:36:15.176327 1203780 PlacerDB.cc:468] Instances Num : 795
I0730 17:36:15.176330 1203780 PlacerDB.cc:469] 1. Macro Num : 0
I0730 17:36:15.176332 1203780 PlacerDB.cc:470] 2. Stdcell Num : 795
I0730 17:36:15.176337 1203780 PlacerDB.cc:471] 2.1 Logic Instances : 274
I0730 17:36:15.176340 1203780 PlacerDB.cc:472] 2.2 Flipflops : 0
I0730 17:36:15.176344 1203780 PlacerDB.cc:473] 2.3 Clock Buffers : 0
I0730 17:36:15.176347 1203780 PlacerDB.cc:474] 2.4 Logic Buffers : 0
I0730 17:36:15.176350 1203780 PlacerDB.cc:475] 2.5 IO Cells : 365
I0730 17:36:15.176353 1203780 PlacerDB.cc:476] 2.6 Physical Instances : 156
I0730 17:36:15.176357 1203780 PlacerDB.cc:477] Core Outside Instances : 0
I0730 17:36:15.176359 1203780 PlacerDB.cc:478] Fake Instances : 0
I0730 17:36:15.176363 1203780 PlacerDB.cc:479] Unplaced Instances Num : 639
I0730 17:36:15.176366 1203780 PlacerDB.cc:480] Placed Instances Num : 0
I0730 17:36:15.176369 1203780 PlacerDB.cc:481] Fixed Instances Num : 156
I0730 17:36:15.176373 1203780 PlacerDB.cc:482] Optional CellMaster Num : 418
I0730 17:36:15.176383 1203780 PlacerDB.cc:523] Nets Num : 675
I0730 17:36:15.176385 1203780 PlacerDB.cc:524] 1. ClockNets Num : 1
I0730 17:36:15.176388 1203780 PlacerDB.cc:525] 2. ResetNets Num : 0
I0730 17:36:15.176393 1203780 PlacerDB.cc:526] 3. SignalNets Num : 674
I0730 17:36:15.176394 1203780 PlacerDB.cc:527] 4. FakeNets Num : 0
I0730 17:36:15.176398 1203780 PlacerDB.cc:528] Don't Care Net Num : 0
```

(Design Information)

```
I0730 17:36:15.264536 1203780 NesterovPlace.cc:447] ----------------Start Global Placement----------------
I0730 17:36:15.276017 1203780 NesterovPlace.cc:1217] [NesterovSolve] Iter: 1 overflow: 0.926664 HPWL: 12609320
I0730 17:36:15.305334 1203780 NesterovPlace.cc:1217] [NesterovSolve] Iter: 10 overflow: 0.668487 HPWL: 14180134
I0730 17:36:15.337035 1203780 NesterovPlace.cc:1217] [NesterovSolve] Iter: 20 overflow: 0.657688 HPWL: 13699254
I0730 17:36:15.368213 1203780 NesterovPlace.cc:1217] [NesterovSolve] Iter: 30 overflow: 0.654918 HPWL: 13563537
I0730 17:36:15.397320 1203780 NesterovPlace.cc:1217] [NesterovSolve] Iter: 40 overflow: 0.655028 HPWL: 13493288
I0730 17:36:15.426545 1203780 NesterovPlace.cc:1217] [NesterovSolve] Iter: 50 overflow: 0.649733 HPWL: 13460770
I0730 17:36:16.613148 1203780 NesterovPlace.cc:1217] [NesterovSolve] Iter: 450 overflow: 0.150503 HPWL: 14113866
I0730 17:36:16.641381 1203780 NesterovPlace.cc:1217] [NesterovSolve] Iter: 460 overflow: 0.130303 HPWL: 14172192
I0730 17:36:16.669582 1203780 NesterovPlace.cc:1217] [NesterovSolve] Iter: 470 overflow: 0.11647 HPWL: 14252435
I0730 17:36:16.698113 1203780 NesterovPlace.cc:1217] [NesterovSolve] Iter: 480 overflow: 0.10064 HPWL: 14335239
I0730 17:36:16.701102 1203780 NesterovPlace.cc:1305] [NesterovSolve] Finished with Overflow:0.0993077 HPWL : 14343090
I0730 17:36:16.703059 1203780 NesterovPlace.cc:459] Global Placement Total Time Elapsed: 1.43847s
I0730 17:36:16.703070 1203780 NesterovPlace.cc:460] ----------------Finish Global Placement----------------
I0730 17:36:16.722851 1203780 Report.cc:686] Current Stage Total HPWL: 14343090
```

```
I0730 17:36:16.754271 1203780 Report.cc:35] ----------------Start iPL Report Generation----------------
I0730 17:36:16.754352 1203780 Report.cc:114] Detect Core outside Instances...
I0730 17:36:16.754364 1203780 Report.cc:126] Detect Instances' Alignment...
I0730 17:36:16.754375 1203780 Report.cc:138] Detect Power Alignment...
I0730 17:36:16.754379 1203780 Report.cc:150] Detect Overlap Between Instances...
I0730 17:36:16.754525 1203780 Report.cc:49] Violation Info Write to './result/pl/report/violation_record.txt'
I0730 17:36:22.382812 1203780 Report.cc:63] Wirelength Info Write to './result/pl/report/wirelength_record.txt'
I0730 17:36:22.383855 1203780 Report.cc:75] Density Info Write to './result/pl/report/density_record.txt'
```

(Global Placement Output, Layout Check)

```
I0730 17:36:16.723914 1203780 AbacusLegalizer.cc:415] ----------------Start Legalization----------------
I0730 17:36:16.733163 1203780 AbacusLegalizer.cc:422] Total Movement: 779548
I0730 17:36:16.733968 1203780 AbacusLegalizer.cc:431] Legalization Total Time Elapsed: 0.01s
I0730 17:36:16.733983 1203780 AbacusLegalizer.cc:432] ----------------Finish Legalization----------------
I0730 17:36:16.733985 1203780 Report.cc:686] Current Stage Total HPWL: 14631748

I0730 17:36:16.737555 1203780 DetailPlacer.cc:471] ----------------Start Detail Placement----------------
I0730 17:36:16.737598 1203780 DetailPlacer.cc:474] Execution Origin Instance Shift:
I0730 17:36:16.738569 1203780 DetailPlacer.cc:478] After RowOpt HPWL: 14546764
I0730 17:36:16.738711 1203780 DetailPlacer.cc:489] Execution Swap Iteration: 0
I0730 17:36:16.741622 1203780 DetailPlacer.cc:494] ---After Global Swap HPWL: 14494797
I0730 17:36:16.743422 1203780 DetailPlacer.cc:500] ---After Vertical Swap HPWL: 14473173
I0730 17:36:16.743824 1203780 DetailPlacer.cc:507] ---After Local Reorder HPWL: 14462013
I0730 17:36:16.745075 1203780 DetailPlacer.cc:524] ---After Row Opt HPWL: 14429151
I0730 17:36:16.745206 1203780 DetailPlacer.cc:489] Execution Swap Iteration: 1
I0730 17:36:16.747771 1203780 DetailPlacer.cc:494] ---After Global Swap HPWL: 14423772
I0730 17:36:16.749518 1203780 DetailPlacer.cc:500] ---After Vertical Swap HPWL: 14414569
I0730 17:36:16.749894 1203780 DetailPlacer.cc:507] ---After Local Reorder HPWL: 14413603
I0730 17:36:16.751096 1203780 DetailPlacer.cc:524] ---After Row Opt HPWL: 14416928
I0730 17:36:16.751260 1203780 DetailPlacer.cc:535] Execution Final Instance Shift Iteration: 0
I0730 17:36:16.752424 1203780 DetailPlacer.cc:545] ---After RowOpt HPWL: 14417052
I0730 17:36:16.753118 1203780 DetailPlacer.cc:556] Detail Plaement Total Time Elapsed: 0.015518s
I0730 17:36:16.753130 1203780 DetailPlacer.cc:557] ----------------Finish Detail Placement----------------
I0730 17:36:16.754197 1203780 Report.cc:686] Current Stage Total HPWL: 14417052
```

(Legalization, Detail Placement)

iPL Tool Report

Basic information about the layout/design

Check violations, and the details of violations are here siblings are listed: *violation_detail_report.txt*

summary_report.txt

scripts > sky130 > result > pl > report > summary_report.txt

```
Generate the report at 2023-08-15T15:10:33
+----------------------+----------+
| Base Info            | Value    |
+----------------------+----------+
| Design               | gcd      |
| Utilization          | 0.098599 |
| Site Num             | 78 * 542 |
| Instances Count      | 795      |
| - Macro Count        | 0        |
| - StdCell Count      | 795      |
| -- FlipFlop Count    | 34       |
| -- Clock Buffer Count| 0        |
| -- Normal Logic Count| 761      |
| Nets Count           | 675      |
| - Signal Net Count   | 674      |
| - Clock Net Count    | 1        |
| - Reset Net Count    | 0        |
| - Other Net Count    | 0        |
+----------------------+----------+

+---------------------------------+-------+
| Violation Info                  | Value |
+---------------------------------+-------+
| Core Range Violated Count       | 0     |
| Row/Site Alignment Violated Count | 0   |
| Power Alignment Violated Count  | 0     |
| Overlap Violated Count          | 0     |
+---------------------------------+-------+
```

```
+-----------------------------------+----------+
| Wirelength Info                   | Value    |
+-----------------------------------+----------+
| Total HPWL                        | 14402289 |
| Max HPWL                          | 328905   |
| Total STWL                        | 15057480 |
| Max STWL                          | 512025   |
| LongNet HPWL (Exceed 1000000) Count | 0      |
+-----------------------------------+----------+

+------------------+----------+
| Bin Density Info | Value    |
+------------------+----------+
| Peak BinDensity  | 1.000000 |
+------------------+----------+

+------------------+----------+----------+----------+----------+
| Clock Timing Info | Early WNS | Early TNS | Late WNS | Late TNS |
+------------------+----------+----------+----------+----------+
| core_clock       | 0.000000 | 0.000000 | -0.194720 | -2.818471 |
+------------------+----------+----------+----------+----------+

+---------------------------+----------+
| Congestion Info           |          |
+---------------------------+----------+
| Average Congestion of Edges | 0.537355 |
| Total Overflow            | 53.000000 |
| Maximal Overflow          | 18.000000 |
+---------------------------+----------+
```

The wire length report, and the detailed report of the long net are in the same directory: *wl_detail_report.txt*

Density of cell distribution

Timing information for placement results

Congestion information of the Placement results

iPL Tool Result

design	instances	fix instances	preset period	origin global placement										legaliztion								origin detail placement						
				target density	bin number	overflow	HPWL	STWL	pin density	congestion	tns	wns	suggest freq(MHz)	total movement	max movement	HPWL	STWL	pin density	congestion	tns	wns	suggest freq(MHz)	HPWL	STWL	pin density	congestion	tns	wns	suggest freq(MHz)
s9234	654	109	0.5	0.8	1024	0.099	2368.32	2740.506	9.727	310	-11.874	-0.234	1362.671	265.398	2.144	2731.559	3119.993	4.33	315	-12.973	-0.239	1353.526	2630.007	2998.719	4.328	289	-12.782	-0.236	1358.982
s5378	875	143	0.5	0.8	1024	0.1	4431.45	4992.777	8.104	446	-1.016	-0.122	1606.939	401.099	2	4894.768	5466.622	4.52	442	-1.341	-0.141	1560.009	4712.884	5271.503	5.346	409	-1.289	-0.146	1547.161
s13207	721	143	0.5	0.8	1024	0.098	2423.16	2799.438	8.724	328	-0.236	-0.043	1840.882	323.34	4.513	2877.73	3254.422	5.556	359	-0.327	-0.053	1806.766	2776.128	3158.136	7.231	325	-0.316	-0.055	1801.468
s15850	2075	240	0.5	0.8	4096	0.099	9525	11050.76	10.928	1105	-28.075	-0.327	1209.259	947.607	1.964	10746.75	12330.61	5.162	1102	-30.092	-0.337	1194.937	10317.48	11828.02	5.808	1059	-29.407	-0.337	1194.265
s38584	6978	476	0.5	0.8	4096	0.099	35087.2	40740.13	8.414	3581	-31.279	-0.384	1130.755	4378.767	3.926	40276.91	46202.29	4.39	4016	-38.247	-0.411	1097.439	38851.73	44767.31	4.39	3903	-36.938	-0.411	1098.288
s1238	349	74	0.5	0.8	1024	0.099	1351.3	1606.042	8.055	143	0	0.003	2013.336	94.274	1.222	1443.468	1691.484	5.089	144	0	0.001	2004.169	1376.559	1601.629	5.059	136	0	0	1999.94
s38417	5977	484	0.5	0.8	4096	0.1	24939.8	30751.57	13.152	3057	-146.608	-0.368	1152.14	3916.849	3.829	30331.62	36583.7	4.522	3595	-161.737	-0.398	1114.072	29036.9	35204.86	4.728	3463	-160.378	-0.386	1129.02
s35932	6315	574	0.5	0.8	4096	0.1	40176.8	47320.76	6.065	3469	0	0.058	2264.831	4078.345	3.337	46724.19	54204	4.55	3958	0	0.051	2229.605	45434.78	52925.59	4.983	3912	0	0.056	2250.124
s713	135	46	0.5	0.8	256	0.098	442.758	465.992	8.113	41	-0.15	-0.057	1794.121	37.333	1.049	478.772	503.425	6.493	43	-0.172	-0.061	1783.151	466.413	490.47	6.493	43	-0.162	-0.056	1797.834
s1488	380	64	0.5	0.8	1024	0.099	1899.63	2343.549	7.008	167	-0.044	-0.021	1918.506	99.35	0.959	1980.038	2426.567	4.376	160	-0.058	-0.028	1895.275	1847.887	2245.791	4.372	150	-0.038	-0.021	1918.484
picorv32	9374	608	1	0.8	4096	0.099	64399.6	74120.87	7.358	4777	-295.297	-0.796	556.849	6594.731	3.371	72779.73	83437.18	3.543	5417	-309.095	-0.825	547.877	70063.46	80828.77	3.407	5296	-297.092	-0.814	551.141
apb4_wdg	1026	161	0.5	0.8	1024	0.1	4288.43	5077.686	6.881	522	-15.571	-0.46	1041.937	537.36	3.556	4913.804	5735.896	4.94	541	-16.626	-0.466	1034.936	4724.135	5544.553	4.944	525	-16.436	-0.463	1038.347
apb4_pwm	971	161	0.5	0.8	1024	0.099	4063.21	4800.453	8.098	515	-10.665	-0.314	1228.068	468.461	1.753	4640.426	5456.172	5.076	525	-11.249	-0.329	1206.415	4455.016	5262.859	5.083	496	-11.089	-0.323	1214.837
apb4_clint	1065	161	0.5	0.8	1024	0.099	4305.61	5198.097	7.887	508	-6.481	-0.406	1103.928	539.157	2.014	5019.946	5912.505	5.287	552	-6.845	-0.417	1091.038	4819.7	5730.576	5.289	513	-6.864	-0.417	1090.163
apb4_timer	719	141	0.5	0.8	1024	0.099	3100.74	3681.118	7.728	355	-13.85	-0.357	1167.41	351.374	5.237	3668.604	4268.195	5.804	380	-14.774	-0.379	1138.037	3524.812	4119.097	5.323	364	-14.758	-0.379	1137.114
apb4_rng	194	67	0.5	0.8	256	0.1	1001.31	1143.27	5.482	99	0	0.191	3239.8	69.528	1.466	1096.764	1240.3	3.845	86	0	0.188	3206.032	1052.713	1186.965	4.385	81	0	0.191	3238.164
apb4_ps2	513	96	0.5	0.8	1024	0.1	1720.48	2238.422	6.615	301	-0.274	-0.037	1863.613	210.223	1.509	1986.076	2518.197	3.969	273	-0.504	-0.049	1821.563	1886.506	2399.051	4.633	273	-0.317	-0.045	1835.054
apb4_i2c	786	141	0.5	0.8	1024	0.1	2885.42	3525.79	7.948	447	-7.949	-0.248	1336.795	369.067	5.659	3378.784	4031.739	5.733	440	-8.382	-0.258	1319.352	3233.585	3887.708	5.292	408	-8.434	-0.26	1316.248
apb4_archinfo	390	87	0.5	0.8	1024	0.1	1608.47	2026.14	7.38	234	0	0.343	6356.35	134.495	1.961	1807.326	2229.803	5.515	238	0	0.344	6399.755	1748.358	2155.912	5.51	198	0	0.344	6410.297
apb4_uart	5940	462	0.5	0.8	4096	0.1	26865.2	34999.61	8.268	3394	-2.572	-0.165	1503.319	3928.755	2.839	31612.65	40162.75	4.55	3811	-2.966	-0.178	1475.967	30352.11	38911.66	4.553	3715	-2.965	-0.177	1476.725
aes	19072	1108	1	0.8	65536	0.1	145099	180334.4	8.418	12847	-679.322	-0.413	707.669	7356.815	3.16	153539.8	188919.3	6.551	12584	-685.911	-0.421	703.963	148005.8	182175.6	6.551	11773	-615.648	-0.419	704.521
aes_core	17850	959	1	0.8	65536	0.1	176974	208824.5	9.021	11827	-846.076	-0.489	671.462	6490.486	2.785	183386.4	215320.6	6.028	11385	-866.814	-0.486	672.983	178103	208977.8	6.029	10669	-845.563	-0.515	660.093
blabla	15115	837	1	0.8	16384	0.1	189061	198132.5	9.267	7054	-337.94	-0.54	649.529	8043.581	3.818	198992.8	208427.8	5.707	7459	-362.613	-0.554	643.311	196236	205643.7	6.063	7224	-356.968	-0.556	642.502
BM64	9311	621	1	0.8	16384	0.099	84565.2	92994.31	10.689	4788	-133.036	-0.593	627.651	4602.728	2.371	90907.8	99302.64	5.63	5055	-136.413	-0.597	626.28	89267.09	97534.57	5.63	4775	-132.308	-0.593	627.848
gcd	296	77	1	0.8	1024	0.115	1518.62	1792.996	8.595	144	0	0.133	1152.756	89.064	2.503	1623.833	1895.38	6.146	138	0	0.125	1142.365	1535.443	1796.569	6.146	137	0	0.135	1155.413
PPU	9464	690	1	0.8	16384	0.099	49478.1	63058.57	9.376	6665	-6.37	-0.289	775.511	4809.541	3.338	56265.45	70129.82	5.427	6680	-7.602	-0.32	757.359	54552.46	67918.85	6.414	6466	-7.423	-0.313	761.548
s44	178	67	1	0.8	256	0.099	633.729	754.641	9.888	78	0	0.852	6739.952	59.748	1.638	713.067	839.563	6.592	78	0	0.851	6693.529	681.689	798.133	6.592	66	0	0.851	6716.819
salsa20	21137	1176	1	0.8	65536	0.1	155819	175708.4	12.829	13022	-252.935	-0.644	608.383	8824.83	2.904	167160.2	187357.6	6.916	12831	-266.444	-0.644	608.264	163219.8	183647.1	6.915	12388	-264.336	-0.654	604.717
jpeg_encoder	27508	1765	5	0.8	65536	0.099	146370	175167.3	14.256	16044	0	1.558	290.505	13648.801	4.587	174267.1	203456.4	7.13	16644	0	1.555	290.282	168506.7	197425.5	7.84	15958	0	1.572	291.7
ratio based on GP				*	*	*	1.00	1.00	1.00	1.00	1.00	1.00	1.00	*	*	1.12	1.11	0.62	1.03	1.12	1.03	0.99	1.08	1.07	0.64	0.98	1.06	1.01	0.99
ratio based on iPL				1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

- **Leglaization damages** wire length and timing, while **detail placement refines** it
- **Pin density and congestion decreased by 36% and 2%,** respectively relative to global Placement results
- Compared with the global placement, the final **HPWL and STWL increased by 8% and 7%; TNS and WNS grew by 6% and 1%**, respectively

Outline

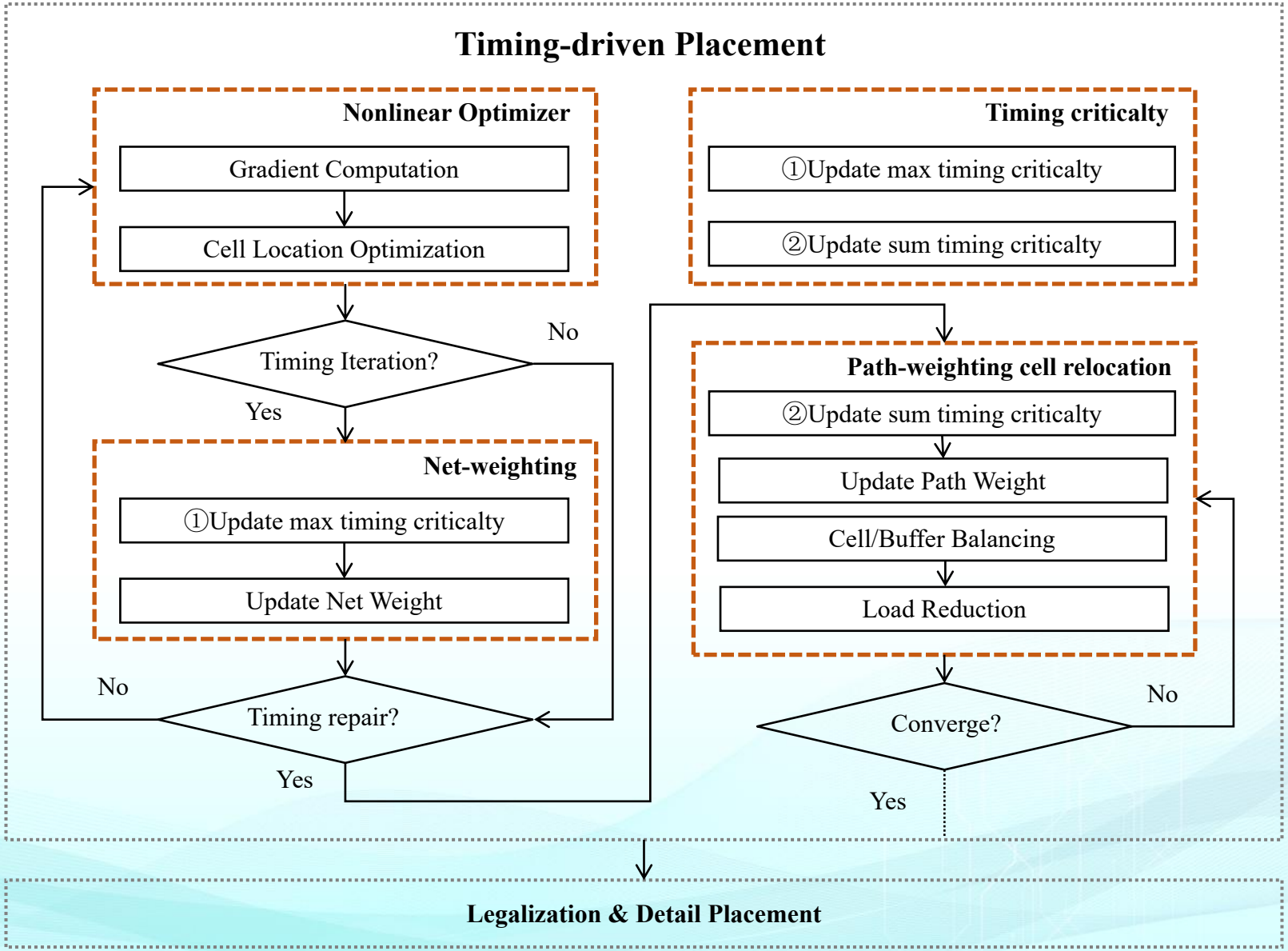
iPL Timing-driven Result

design	instances	fix instances	preset period	timing-driven global placement									legaliztion								timing-driven detail placement						
				target density	bin number	overflow	HPWL	STWL	pin density	congestion	tns	wns	suggest freq(MHz)	total movement	max movement	HPWL	STWL	pin density	congestion	tns	wns	suggest freq(MHz)	HPWL	STWL	pin density	congestion	tns	wns	suggest freq(MHz)
s9234	654	109	0.5	0.8	1024	0.099	2407.88	2776.801	8.106	341	-11.543	-0.227	1375.828	272.401	2	2773.705	3153.179	4.339	324	-12.43	-0.231	1367.2	3127.63	3482.056	4.881	306	-11.59	-0.233	1363.339
s5378	875	143	0.5	0.8	1024	0.099	4457.2	5008.695	8.918	443	-0.752	-0.108	1644.169	407.104	2.899	4949.277	5515.554	4.931	460	-0.97	-0.116	1624.284	5575.283	6137.768	4.931	462	-0.967	-0.119	1615.817
s13207	721	143	0.5	0.8	1024	0.099	2436.51	2806.642	10.371	313	-0.178	-0.046	1831.982	330.982	3.655	2890.775	3275.096	6.646	344	-0.306	-0.064	1772.296	3030.189	3405.571	6.646	354	-0.265	-0.06	1784.771
s15850	2075	240	0.5	0.8	4096	0.098	9555.68	11090.43	9.635	1118	-27.445	-0.326	1210.901	933.339	2.022	10759.84	12316.67	5.161	1160	-29.839	-0.337	1195.337	11838.2	13365.27	5.803	1133	-28.011	-0.328	1207.623
s38584	6978	476	0.5	0.8	4096	0.1	35186.2	40815.71	8.968	3637	-27.899	-0.386	1128.503	4453.886	4.052	40512.13	46438.18	4.207	4014	-35.423	-0.416	1091.869	47128.74	52918.02	4.207	3986	-30.684	-0.397	1114.658
s1238	349	74	0.5	0.8	1024	0.099	1351.3	1606.042	8.055	143	0	0.003	2013.336	94.274	1.222	1443.468	1691.484	5.089	144	0	0.001	2004.169	1793.035	2047.52	5.105	142	-0.034	-0.034	1874.246
s38417	5977	484	0.5	0.8	4096	0.099	25096.8	30813.01	11.095	3120	-141.273	-0.367	1153.918	4053.517	3.465	30583.46	36743.92	4.726	3589	-156.314	-0.394	1118.082	38166.51	43911.2	4.726	3563	-148.462	-0.385	1129.328
s35932	6315	574	0.5	0.8	4096	0.1	40176.8	47320.76	6.065	3469	0	0.058	2264.831	4078.345	3.337	46724.19	54204	4.55	3958	0	0.051	2229.605	45434.78	52925.59	4.983	3912	0	0.056	2250.124
s713	135	46	0.5	0.8	256	0.284	436.453	457.289	12.62	38	-0.068	-0.04	1851.464	64.159	1.635	502.79	529.342	6.4	44	-0.128	-0.05	1817.471	537.382	564.156	6.377	51	-0.077	-0.036	1864.924
s1488	380	64	0.5	0.8	1024	0.099	1915.58	2354.072	5.265	159	-0.012	-0.012	1954.541	115.552	0.97	1990.962	2420.521	4.376	163	-0.022	-0.015	1941.552	2569.168	2959.808	4.376	159	-0.011	-0.009	1966.104
picorv32	9374	608	1	0.8	4096	0.099	64380.9	74069.7	6.814	4708	-291.286	-0.78	561.678	6605.858	3.455	72698.89	83339.04	3.406	5300	-307.307	-0.81	552.419	80222.69	90825.84	3.407	5339	-280.602	-0.785	560.306
apb4_wdg	1026	161	0.5	0.8	1024	0.099	4305.74	5061.221	7.214	515	-15.382	-0.454	1048.003	524.809	3.42	4947.19	5734.022	4.616	539	-16.361	-0.467	1033.885	5524.726	6302.903	4.614	531	-16.017	-0.453	1049.806
apb4_pwm	971	161	0.5	0.8	1024	0.1	4086.02	4839.995	8.424	506	-10.092	-0.307	1239.839	470.603	2.208	4669.512	5488.174	5.076	519	-10.906	-0.326	1210.224	5696.122	6515.263	5.093	526	-10.528	-0.32	1218.979
apb4_clint	1065	161	0.5	0.8	1024	0.1	4319.99	5195.241	7.887	536	-6.196	-0.395	1116.922	552.119	2.83	5003.16	5900.002	4.964	555	-6.621	-0.408	1101.445	5696.243	6601.515	4.956	568	-6.349	-0.4	1111.088
apb4_timer	719	141	0.5	0.8	1024	0.099	3105.23	3683.347	8.223	359	-13.597	-0.351	1174.768	339.368	3.795	3657.153	4255.826	6.291	370	-14.425	-0.371	1148.273	3723.447	4319.168	6.294	371	-14.213	-0.365	1156.083
apb4_rng	194	67	0.5	0.8	256	0.1	1001.31	1143.27	5.482	99	0	0.191	3239.8	69.528	1.466	1096.764	1240.3	3.845	86	0	0.188	3206.032	1052.713	1186.965	4.385	81	0	0.191	3238.164
apb4_ps2	513	96	0.5	0.8	1024	0.099	1753.16	2277.679	6.606	265	-0.018	-0.015	1941.898	201.404	1.595	2001.184	2534.246	4.649	275	-0.08	-0.028	1893.183	2521.375	3030.886	4.652	272	-0.12	-0.026	1902.327
apb4_i2c	786	141	0.5	0.8	1024	0.099	2901.34	3517.388	7.5	439	-6.62	-0.212	1404.897	377.639	4.235	3395.061	4028.909	5.297	448	-7.308	-0.228	1374.17	3611.244	4233.445	5.297	443	-7.174	-0.226	1376.766
apb4_archinfo	390	87	0.5	0.8	1024	0.1	1608.47	2026.14	7.38	234	0	0.343	6356.35	134.495	1.961	1807.326	2229.803	5.515	238	0	0.344	6399.755	1748.358	2155.912	5.51	198	0	0.344	6410.297
apb4_uart	5940	462	0.5	0.8	4096	0.1	26910.6	35076.76	8.89	3410	-2.379	-0.157	1521.403	3895.961	2.897	31585.41	40118.98	4.345	3818	-2.909	-0.183	1463.066	43239.57	51597.61	4.551	3752	-4.021	-0.186	1458.402
aes	19072	1108	1	0.8	65536	0.1	145159	180346.3	8.422	12757	-611.084	-0.415	706.504	7360.623	2.633	153517.7	188826.5	5.615	12492	-636.698	-0.419	704.846	212348.8	246549.6	5.615	12439	-369.979	-0.325	754.465
aes_core	17850	959	1	0.8	65536	0.099	177089	208872.9	9.019	11762	-830.87	-0.477	677.265	6449.054	3.242	183461.8	215266.6	6.032	11362	-851.248	-0.48	675.764	281159.8	311415.5	6.029	11589	-602.798	-0.438	695.226
blabla	15115	837	1	0.8	16384	0.099	189178	198157.5	9.981	7052	-328.961	-0.535	651.451	8109.145	3.723	199114.6	208439.1	6.064	7523	-351.097	-0.548	645.806	213932.9	222760.8	5.707	7437	-317.973	-0.523	656.554
BM64	9311	621	1	0.8	16384	0.099	84608.5	92968.46	11.255	4808	-121.251	-0.57	636.966	4672.479	3.555	90975.91	99298.45	5.068	5167	-127.603	-0.586	630.397	101306.1	109608.7	5.068	4935	-97.806	-0.536	650.954
gcd	296	77	1	0.8	1024	0.115	1518.62	1792.996	8.595	144	0	0.133	1152.756	89.064	2.503	1623.833	1895.38	6.146	138	0	0.125	1142.365	1535.443	1796.569	6.146	137	0	0.135	1155.413
PPU	9464	690	1	0.8	16384	0.099	49547.3	63049.41	9.868	6672	-4.876	-0.289	775.653	4855.375	3.211	56361.12	70184.78	5.921	6660	-6.221	-0.311	762.597	56982.69	70774.39	5.921	6660	-5.372	-0.313	761.629
s44	178	67	1	0.8	256	0.099	633.729	754.641	9.888	78	0	0.852	6739.952	59.748	1.638	713.067	839.563	6.592	78	0	0.851	6693.529	681.689	798.133	6.592	66	0	0.851	6716.819
salsa20	21137	1176	1	0.8	65536	0.1	155790	175711.4	10.858	12910	-250.689	-0.643	608.753	8923.344	3.113	167261.2	187447.6	6.914	12757	-262.933	-0.644	608.327	203926.6	224781.8	6.914	12544	-235.561	-0.617	618.54
jpeg_encoder	27508	1765	5	0.8	65536	0.099	146370	175167.3	14.256	16044	0	1.558	290.505	13648.801	4.587	174267.1	203456.4	7.13	16644	0	1.555	290.282	168506.7	197425.5	7.84	15958	0	1.572	291.7
ratio based on GP				*	*	*	1.00	1.00	1.00	1.00	1.00	1.00	1.00	*	*	1.13	1.11	0.62	1.04	1.27	1.06	0.99	1.27	1.23	0.63	1.02	1.23	0.61	1.00
ratio based on iPL				1.00	1.00	1.06	1.00	1.00	1.01	0.99	0.87	0.94	1.01	1.03	1.05	1.00	1.00	1.01	1.00	0.90	0.96	1.01	1.18	1.15	0.99	1.04	0.86	0.92	1.01

- Leglaization damages wire length and timing, while detail placement refines it
- Compared to the initial version of iPL, **TNS and WNS improved 14% and 8%** , respectively
- Some **damages occurred in wirelength and congestion**

iPL Timing-driven Flow

- **Nonlinear Optimizer**
  - Use the gradient method to move the cell
- **Timing Criticalty**
  - Criticalty: “max”, “sum” calculation
- **Net-weight Cell Movement**
  - The net weight is updated to the gradient according to the “max” of the timing criticalty
- **Path-wight Cell Relocation**
  - Update “sum” of the timing criticalty
  - Cell/Buffer Balancing
  - Load Reduction

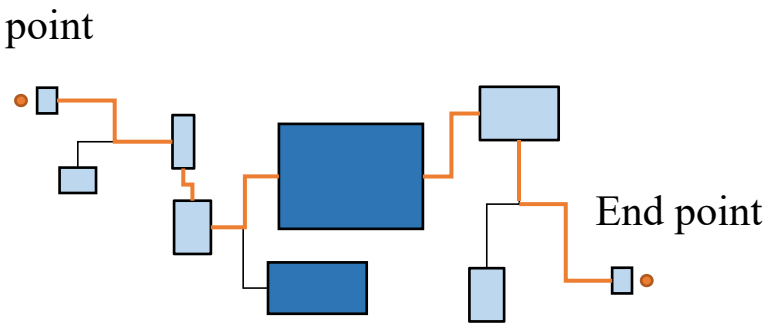


Timing Criticalty

– Initially defined at the end of the timing path

$$SLACK^{Late} = RequiredTime^{Late} - ArrivalTime^{Late} \quad (1)$$

$SLACK^{Late} < 0$, Indicates that there is a timing violation at the end of the path



– The criticality of Primary Output (PO) is modeled as：

$$\lambda_{PO}^{0} = 1 \quad (2)$$

$$\lambda_{PO}^{k+1} = \lambda_{PO}^{k} * \left(\frac{ArrivalTime^{Late}}{RequiredTime^{Late}}\right) \quad (3)$$

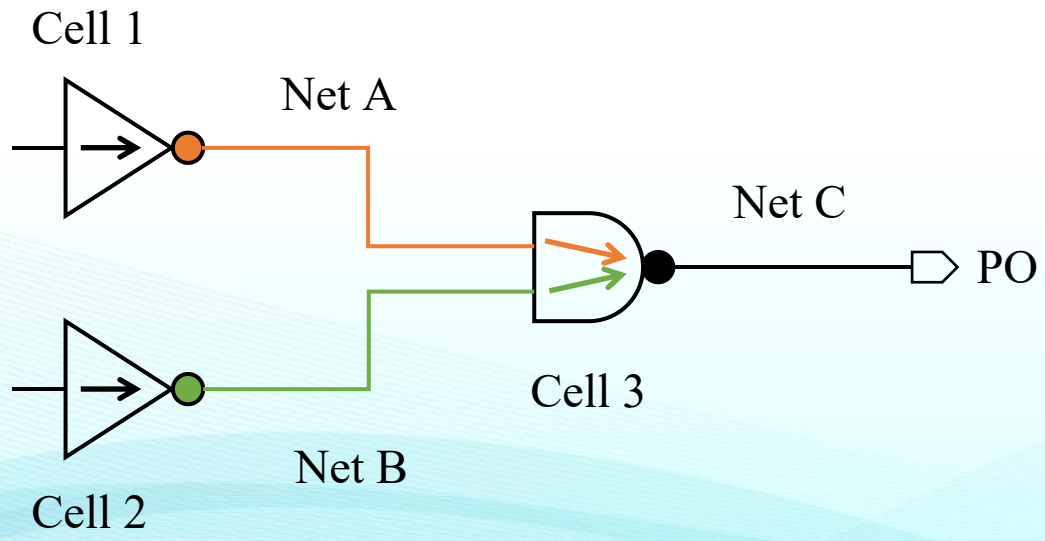
- If there is a timing violation, the value on the right side of the equation will be greater than 1. The more severe the violation, the higher the value and the higher the criticality. The multiplication maintains the historical information of the previous step.

Timing Criticalty

– The net criticality is propagated through the inverse topological order assignment of cells

$$\lambda_{Net_{i,j}} = \lambda_{Net_{i,j}} * \left(\frac{ArrivalTime_i^{Late} + Delay_{i,j}^{Late}}{ArrivalTime_j^{Late}}\right)$$

– For example



$$\lambda_{A_1,3} = \lambda_{A_1,3} * \left(\frac{ArrivalTime_1^{Late} + Delay_{1,3}^{Late}}{ArrivalTime_3^{Late}}\right)$$

$$\lambda_{B_2,3} = \lambda_{B_2,3} * \left(\frac{ArrivalTime_2^{Late} + Delay_{2,3}^{Late}}{ArrivalTime_3^{Late}}\right)$$

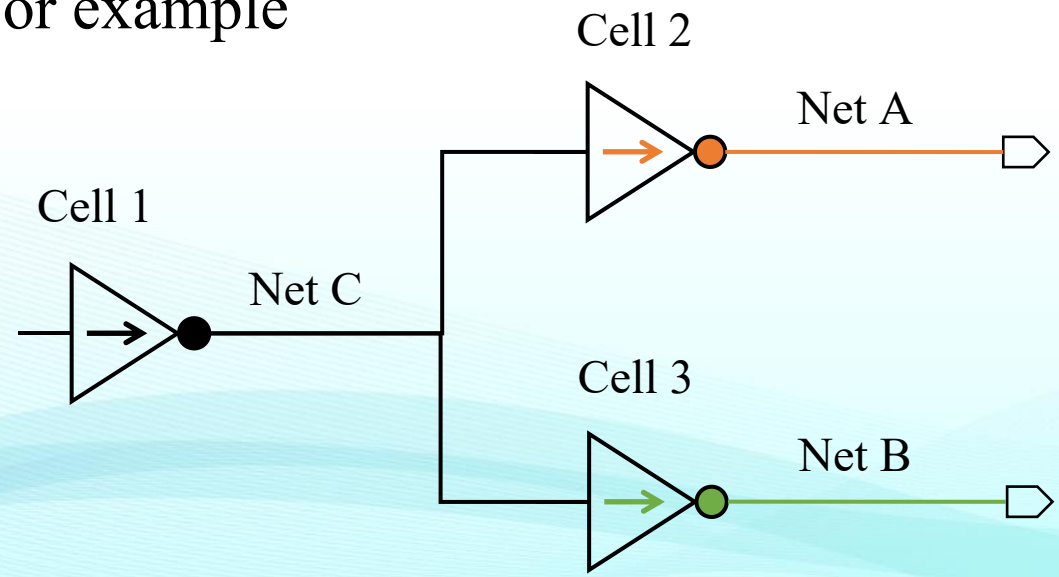
Timing Criticalty

– Net criticality is propagated through wire net inverse topological order allocation

① The **max timing criticality** method: $\lambda_{Net_i} = \max_{j \in Net_i} \lambda_{Net_{j,j}}$ → For net weighting

② The **sum timing criticality** method: $\lambda_{Net_i} = \sum_{j \in Net_i} \lambda_{Net_{j,j}}$ → For path weighting

– For example



The "max" method : $\lambda_{C_1} = max(\lambda_{A_2}, \lambda_{B_3})$

The "sum" method : $\lambda_{C_1} = \lambda_{A_2} + \lambda_{B_3}$

Net Weighting

– Net weighting based on net criticality assessment using momentum method

1. Known the criticality of the (m-1)-th round and the mth round iterative net λ_{net}^{m-1}、 λ_{net}^{m}
2. Variation of the weights of the line network in the m-th iteration: $\Delta w_{net}^{m} = \boldsymbol{\beta} * \lambda_{net}^{m-1} + (\mathbf{1} - \boldsymbol{\beta}) * \lambda_{net}^{m}$
3. Net weight $w_{net}^{m} = w_{net}^{(m-1)} + \Delta w_{net}^{m}$

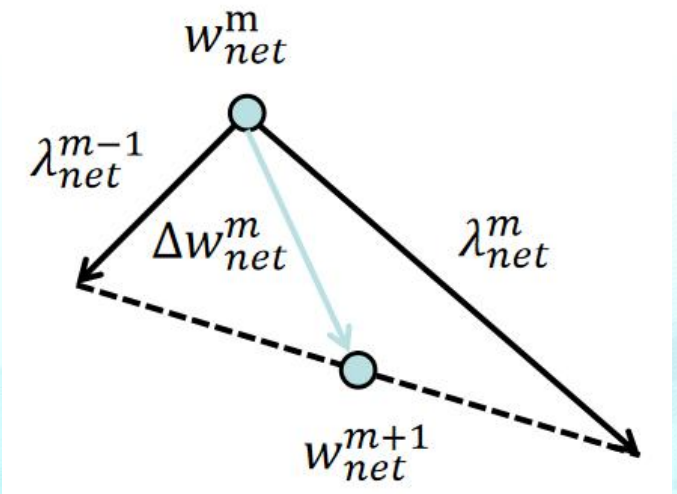
Update ω_e

$$\underset{(x,y)}{\text{Minimize}} \sum_{e \in E} \omega_e * WL_e(x,y)$$

$$Subject\ to$$

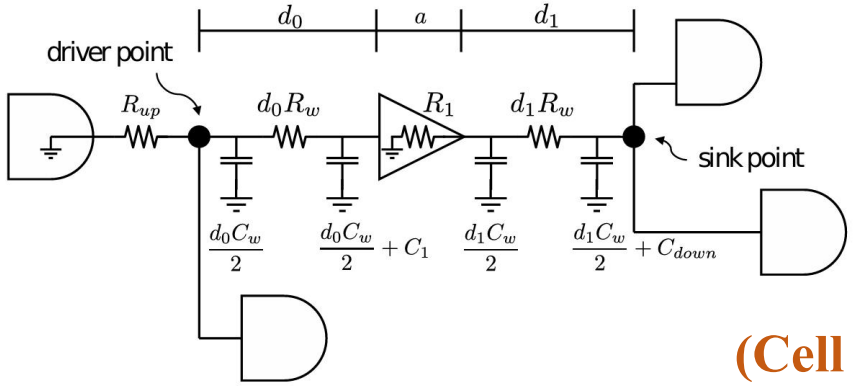
$$\rho_b(x,y) \le \rho_0,\ \forall b \in B\ ;$$

The momentum method is used to maintain the weight change of the previous step

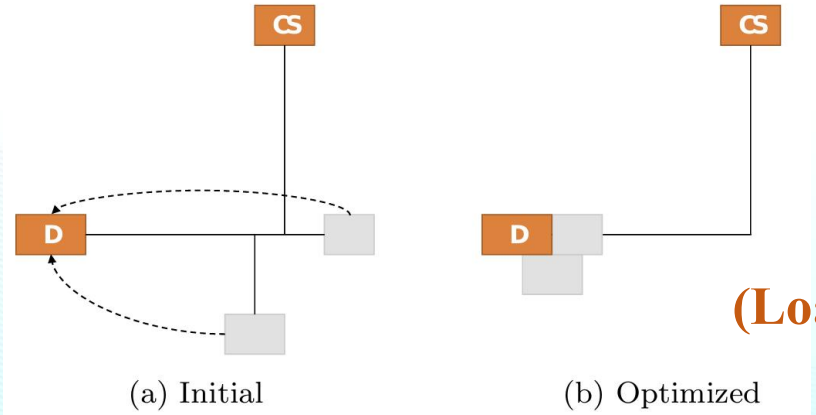


Cell Relocation

– Mix multiple cell relocation methods

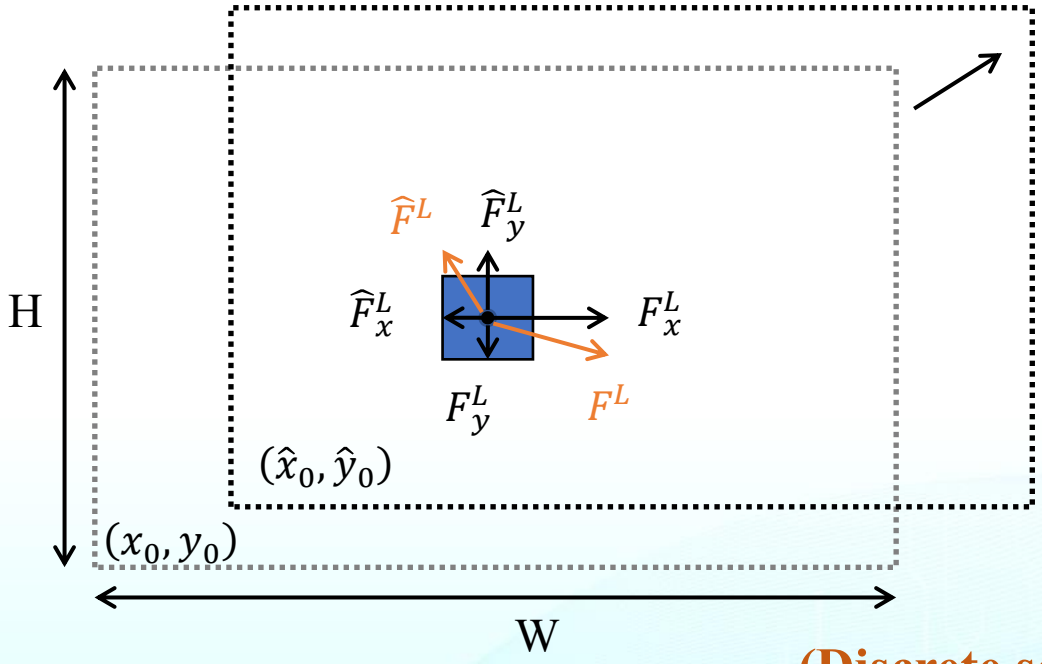


(Cell Balancing)



(Load Reduction)

$$\hat{x}_0 = x_i - \frac{|L_i|}{|L_i| + |R_i|} \qquad \hat{y}_0 = y_i - \frac{|D_i|}{|D_i| + |U_i|}$$



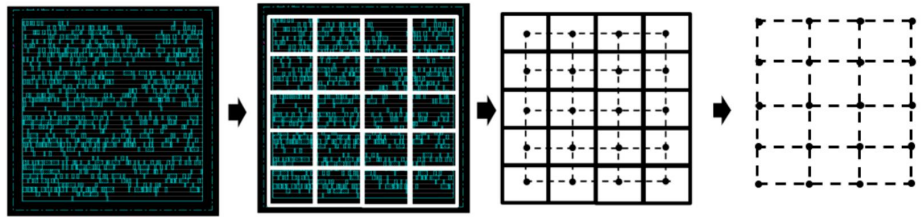
(Discrete search)

- Flach G, Fogaça M, Monteiro J, et al. Drive strength aware cell movement techniques for timing driven placement[C]//Proceedings of the 2016 on International Symposium on Physical Design. 2016: 73-80.

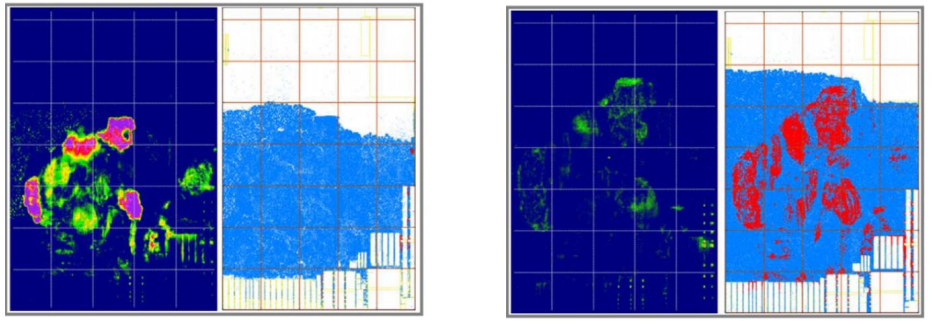
Outline

Significance

- **Relationship between Routability and Placement**
  - Placement determines the cell positions, affecting the routing of interconnects, leading to mismatched demand and capacity values for edges in the routing graph.
  - Routability-driven placement adjusts cell positions based on the congestion map to avoid hotspots, which facilitates subsequent routing. Without routability consideration, routing may encounter detours or even inability to connect between cells, necessitating iterative revisions and reducing VLSI design efficiency.
  - Content of Routability-driven Placement (Focus on Evaluation and Optimization)
    - Wirelength-driven placement until cells are adequately dispersed
    - Building routing graph
    - Evaluation: Congestion, DRC violations, Pin accessibility
    - Optimization: Cell inflation, Congestion gradients



(The Process of Building a Routing Graph)



(Comparison: Wirelength-driven vs. Routability-driven Placement)

Evaluation

- **Methods**
  - In placement, rapid and accurate congestion evaluations are essential. Evaluating pin distribution helps identify congestion-prone areas and generate a congestion map.
  - Congestion for Each Grid:

$$capacity_{i,j} = \left\lfloor \frac{grid\_h}{d_{pitch}(layer)} \right\rfloor$$

$$overflow_{i,j} = \max[0, (\text{demand}_{i,j} - \text{capacity}_{i,j})]$$

$$util_{i,j} = \text{demand}_{i,j} / \text{capacity}_{i,j}$$

  - Input:
    - Net/ Pin Coordinates
    - Blockage
    - Track/ layer
  - Output:
    - Congestion map
  - Metrics:
    - Total overflow / Maximum overflow
    - Peak Weighted Congestion(PWC)

$$PWC = \frac{\sum(K_x \times ACE(x))}{\sum K_x}$$

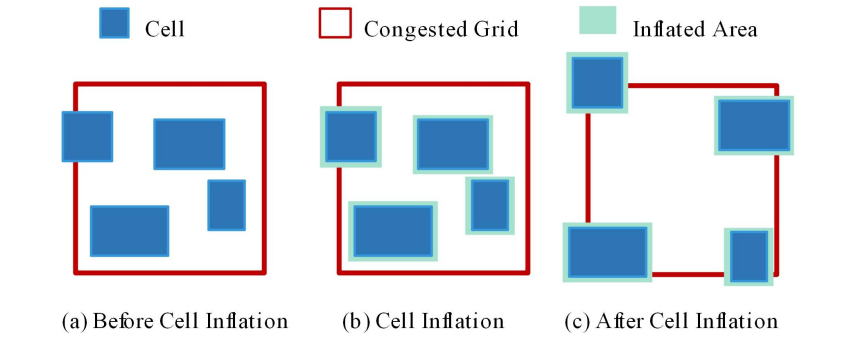
$ACE(x)$, $x \in \{0.5, 1, 2, 5\}$

(Comparison of Four Congestion Evaluation Methods)

Method	Description	Accuracy	Speed
Static	Based on simple design metrics	4th	1st
Probabilistic	Relies on routing patterns and probabilities	3rd	3rd
Constructive	Utilizes global routing tools	2nd	4th
Network	Trains machine learning models	1st	2nd

Optimization

- **Methods**
  - Indirect Congestion Optimization via Density Tuning
    - Cell Inflation:
      - Temporarily enlarge congested grid cells.
      - Key considerations: which cells to inflate, direction, and amount.
    - Threshold Density Adjustment:
      - Decrease the density threshold of congested grid cells, such as based on pin density corrections.



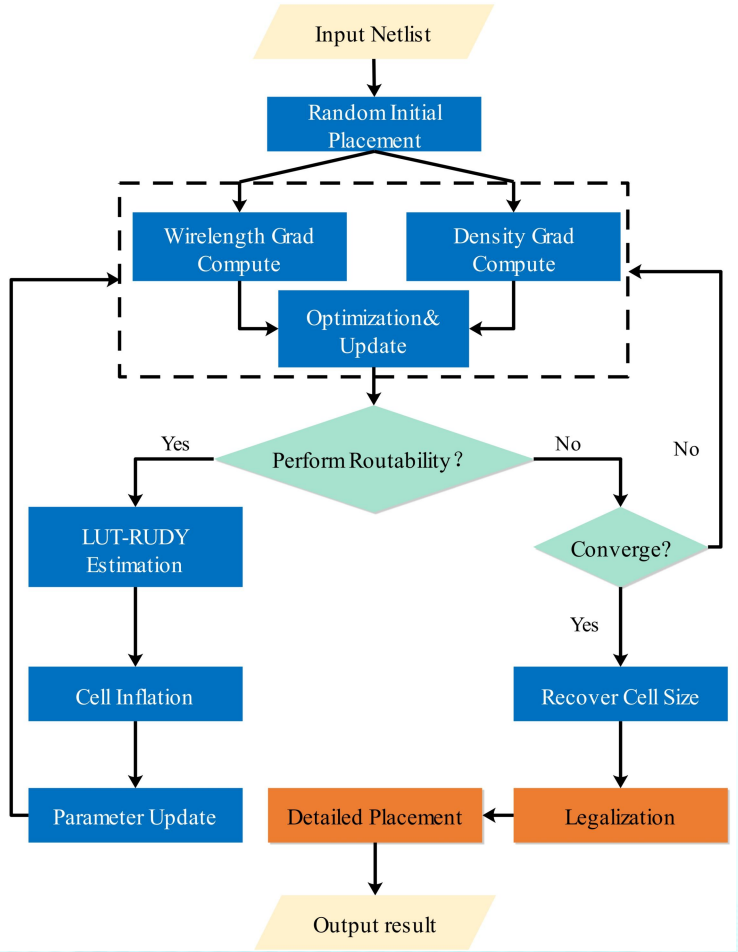
(The process of cell inflation)

- Direct Congestion Representation in Objective Function
  - Smooth Net Congestion:
    - min $\lambda_1 WL + \lambda_2 \sum (D - M_b)^2 + \lambda_3 \sum (C - S_b)^2$
  - Smooth Pin Density:
    - min $\lambda_1 WL + \lambda_2 \sum (D - M_b)^2 + \lambda_3 P$
  - Congestion Weighting on Wirelength:
    - $WL = \sum (\alpha_e \max |x_i - x_j| + \beta_e \max |y_i - y_j|)$
    - $\alpha_e = 1 + (f_y - f_x),\ \beta_e = 1 - (f_y - f_x$

(Comparison of Strategies for Cell Inflation Methods)

Tool	Time	Cell Selection	Inflation Direction	Inflation Rate
SimPLR	2011	Top 5% Most Congested Grids	Horizontal	Simple Weighting Function
Ripple	2013	All Congested Grids	Horizontal/ Vertical	Simple Piecewise Function
POLAR	2014	Top 10% Most Congested Grids	Unspecified	Fixed at 10%
RePlAce	2019	All Congested Grids	Unspecified	Superlinear Function

Our Routability-driven Solution

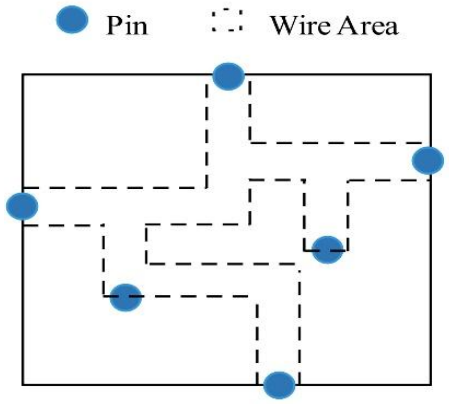


- Wirelength Gradient: WA model
- Density Gradient: e-Density electrostatic field model
- Algorithm: Nesterov gradient descent
- Congestion Evaluation Methods
  - LUT-RUDY (Look Up Table-based RUDY)
  - Early-GR
- Fine-Grained Cell Inflation:
  - Selection of cells from peak congested grids
  - Independent inflation in horizontal (H) and vertical (V) directions
  - Dynamic inflation rate adjustment
  - Superlinear inflation index correction

Routability evaluation and optimization begin once all cells in global placement are sufficiently dispersed (density overflow < 0.2).

Evaluation Motivation

- **Issues with Existing Methods:**
  - Static methods provide limited information, like pin density, and are fast but less accurate in representing congestion.
  - Constructive methods directly use global routers to estimate congestion, offering high accuracy but can be slow due to router limitations.
  - Probabilistic methods analyze wire density distribution, often using techniques like RUDY (Rectangular Uniform wire DensitY). RUDY balances evaluation speed and accuracy but makes unrealistic assumptions that limit precision.
    - Assumption 1: No need for accurate routing demand estimation for each net.
    - Assumption 2: Routers prioritize routing within the bounding box of each net.



(RUDY Conceptual Figure)

$$D_e = \frac{wire\ area}{net\ area} = \frac{HPWL * Width}{BBox\ Area}$$

- Previous research has emphasized routability optimization, often using default methods for evaluation such as invoking routers. However, evaluation is equally crucial as optimization!
- Key Questions: **How to design a routability evaluation method that balances performance and accuracy?**

$$\frac{D(e)}{C(e)} = \sum_{i=1}^{m} \frac{HPWL(i) \times Width(i) \times Overlap(i)}{BBoxArea(i) \times C(e)}$$

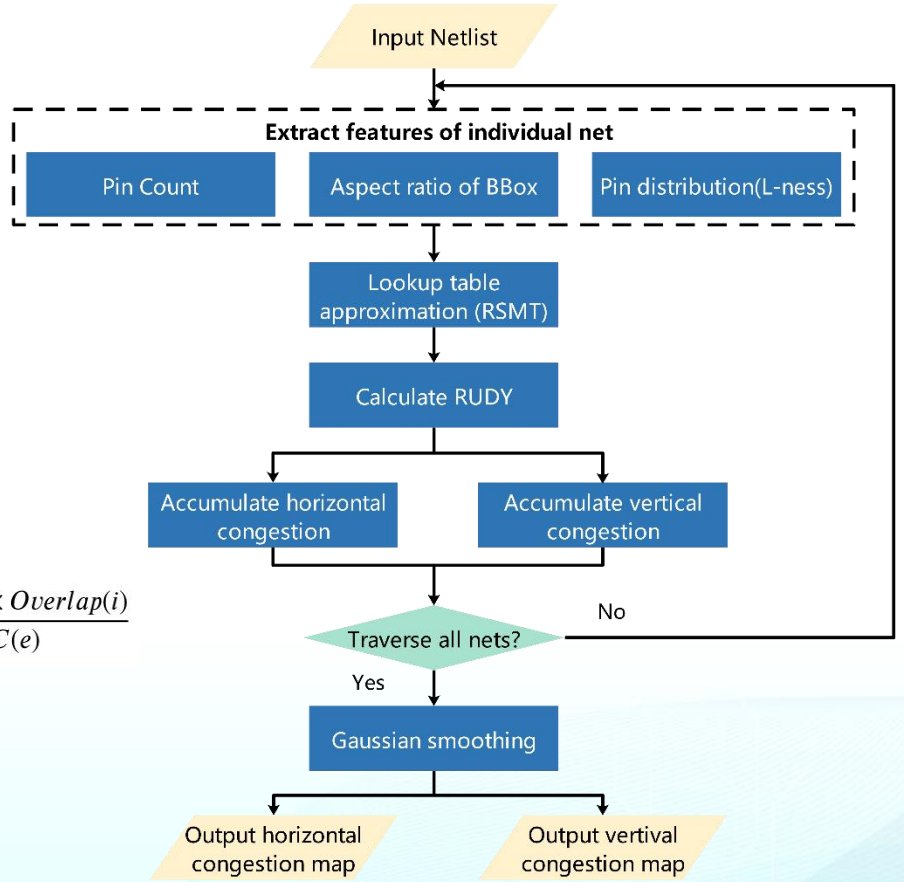
m represents all nets, $\frac{D(e)}{C(e)}$ represents the congestion value of each bin, $Overlap(i)$ indicates the overlap area ratio between net and the current bin.

LUT-RUDY

- **Proposed Method: LUT-RUDY**
  - Two Assumptions of RUDY:
    - Assumption 1: No need for accurate routing demand estimation for each net.
    - Assumption 2: Routers prioritize routing within the bounding box of each net.
  - Addressing Assumption 1:
    - Inaccurate routing demand estimation for individual nets can accumulate and be magnified in modern designs with thousands of nets, leading to imprecise overall routing demand estimation. To improve the accuracy of individual net evaluation, we propose a strategy based on lookup table (LUT)-based true wirelength approximation.
  - Addressing Assumption 2:
    - To address unpredictable router behavior and congestion-prone designs requiring detour routing, we propose simulating routing behavior using Gaussian smoothing, which considers scenarios beyond net bounding rectangles.

$$\frac{D(e)}{C(e)} = \sum_{i=1}^{m} LUT(PC, AR, Lness) \times \frac{HPWL(i) \times Width(i) \times Overlap(i)}{BBoxArea(i) \times C(e)}$$

$$w_{ij} = \frac{1}{2\pi\sigma^2} \times e^{-\frac{(i-i_0)^2+(j-j_0)^2}{2\sigma^2}}$$



(Congestion Evaluation Process based on LUT-RUDY)

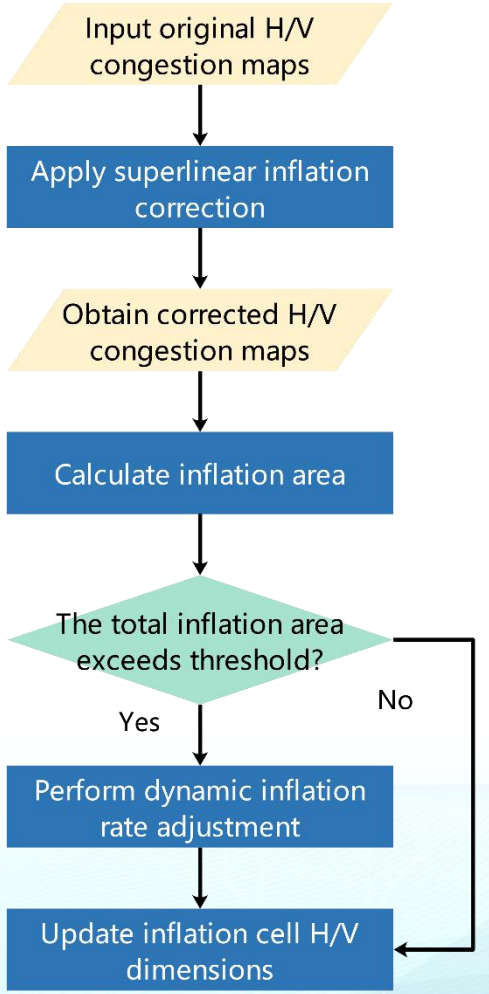
Routability Optimization-1

- **Fine-Grained Cell Inflation Strategy**
  - Congestion Map Preprocessing
    - Apply superlinear correction function to enhance pixel value differences and improve subsequent cell inflation effects.
      - Limit the unidirectional inflation rate $h_{max}(v_{max})$ to not exceed 1.6 to maintain algorithm stability.

$$\text{Ratio}_h = \max\left(\left(\frac{D(e)}{C(e)}\right)^{\gamma_{\text{super}}^{hor}}, h_{max}\right)$$

$$\gamma_{\text{super}}^{hor} = \alpha \cdot \text{PWC}_{hor}$$

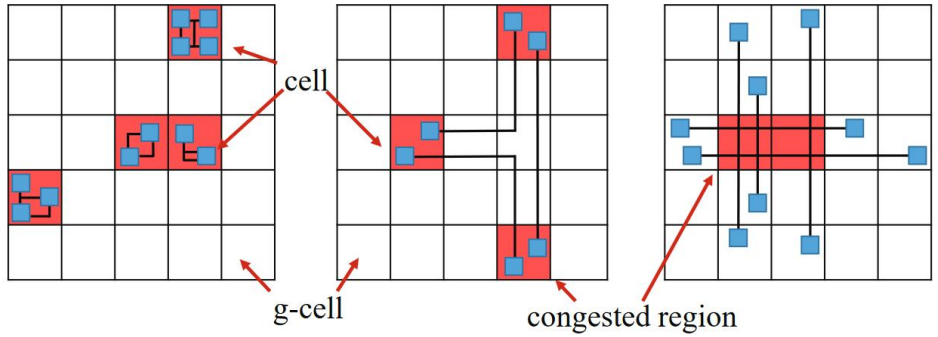
  - Selection of Cells for Inflation
    - Focus on areas with high local congestion.
      - Inflate cells within grids with congestion above the square root of the average peak weighted congestion ($\sqrt{PWC}$)
  - Direction of Inflation
    - Independently inflate cell height and width guided by the H/V congestion maps.
  - Extent of Inflation
    - Inflation rate decreases with decreasing average peak congestion.
    - Dynamic Inflation Rate Adjustment, where the total inflation area threshold is 10% of the layout's whitespace area.



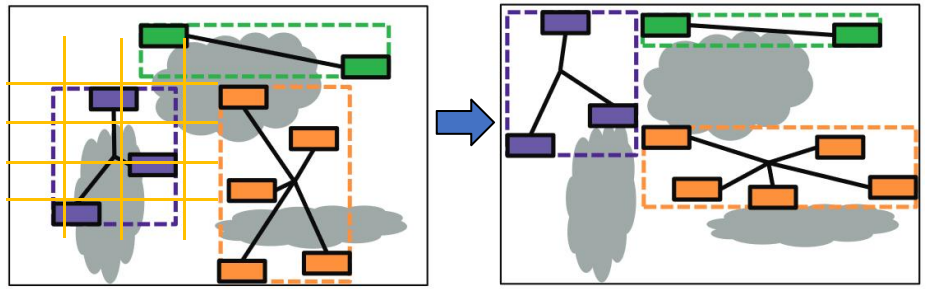
(Algorithm Flow for Cell Inflation)

Routability Optimization-2

- **Differentiable Congestion Gradient**
  - Motivation:
    - The cell inflation method cannot handle cases where there are no cells within congested grids.
    - The final result of $\min\ \lambda_1 WL + \lambda_2 \sum (D - M_b)^2 + \lambda_3 \sum (C - S_b)^2$ leads to a uniform wire density distribution, whereas congestion should focus more on locality.
  - Model Assumptions:
    - Routers will only route within the BBox of each net.
    - Equal probability of routing at all positions within each net.
  - Mathematical Model:
    - $\min\ \lambda_1 WL + \lambda_2 \sum (D - M_b)^2 + \lambda_3 \sum C$
      - $C_i = \sum \frac{P_x(N_i, g_k) \times P_y(N_i, g_k) \times Overflow_k}{\sum P_x(N_i, g_k) \times P_y(N_i, g_k)}$
      - Introduce Bell-shaped functions to smooth $P_x(N_i, g_k)$ and $P_y(N_i, g_k)$



(Local and Global Congestion)



(Net movement using net congestion penalties)

- Center point x_{N_i} of N_i is represented as $x_{N_i} = \frac{\max x_k + \min x_k}{2}$
- Width w_{N_i} of N_i is represented as $w_{N_i} = \max x_k - \min x_k$
- Apply WA function to smooth x_{N_i} and w_{N_i}

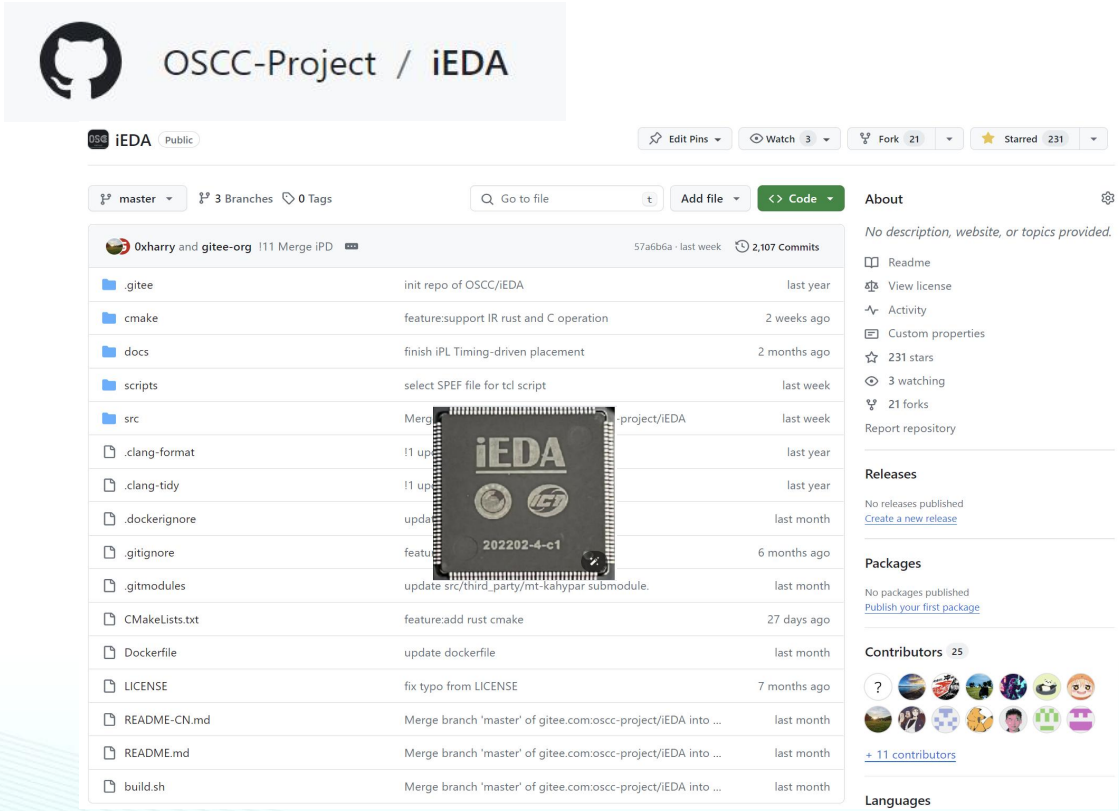
J.-M. Lin, et al. Routability-driven Global Placer Target on Removing Global and Local Congestion for VLSI Designs[C]. ICCAD,2021.

Outline

Conclusion

- **Summaries about iPL 1.0**
  - We have developed a tool iPL to complete automatic placement, including global placement, legalizations, detailed placement, and support for timing-driven and routablity-driven placements
  - iPL works with other iEDA tools and provides rich interface calls
  - The interface supports TCL, python and C++
  - The goal of the design is to have a clear hierarchy, decoupled modules, and ease of use
  - In the future, we will iteratively optimize with iCTS, iRT, and iSTA to achieve better PPA. At the same time, AI models will be connected in the future to promote the improvement of placement.

Q & A



Thanks

Shijian Chen
chenshj@pcl.ac.cn