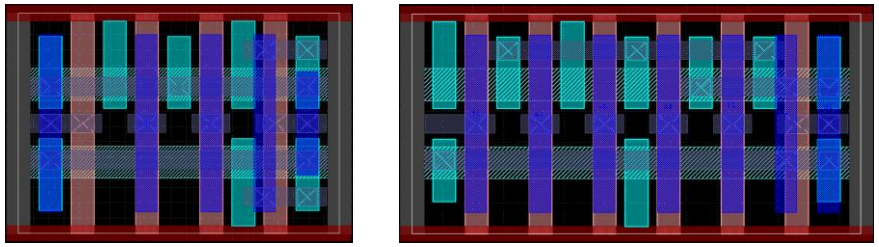
TransOpt: A Scalable Transistor-Level Placement and Routing Optimization Framework Beyond Standard Cells

Chen-Hao Hsu and David Z. Pan

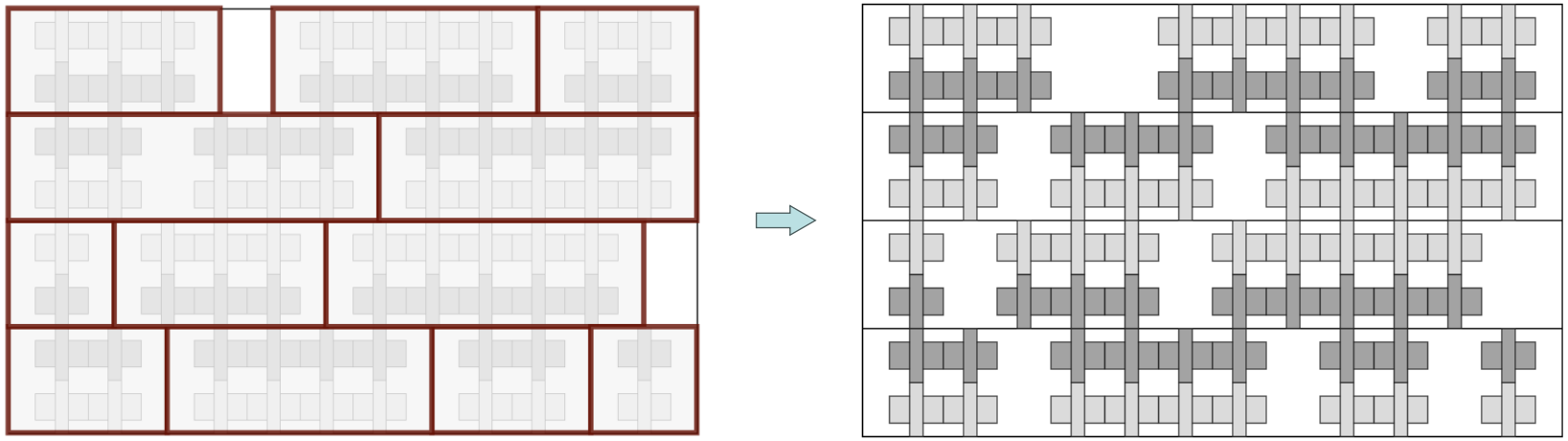
Department of Electrical and Computer Engineering
The University of Texas at Austin

Why Transistor-Level Design?

- Revisit standard-cell methodology
  - Pros: scalability, reusability, compatibility
  - Cons: limit finer-grained PPA optimization



- Break the **"abstraction"** of standard cells (SDCs)
  - Offer greater optimization flexibility
  - Squeeze out more PPA improvements



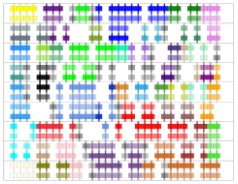
"Walls" of standard cells

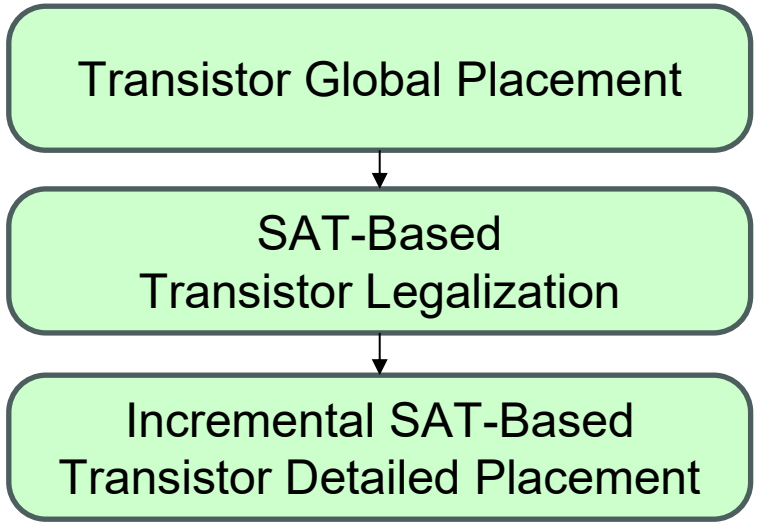
Transistor-level design

Motivation | Placement Scalability

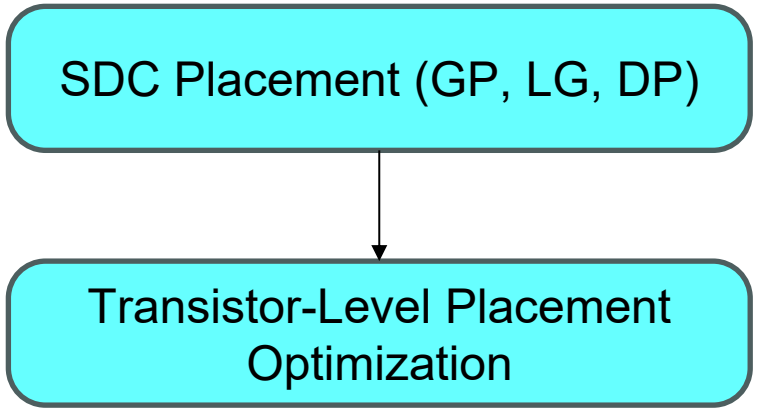
- TransPlace [ASPDAC'24] struggles to find a feasible solution for >1000 transistors
  - Formulate legalization problem into one single SAT instance
- **TransOpt** uses SDC placement as initial placement

Bottleneck:
1 SAT instance for entire design

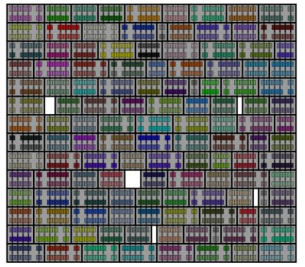




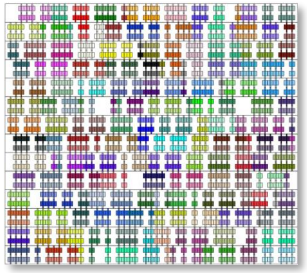
TransPlace Flow



Proposed Flow

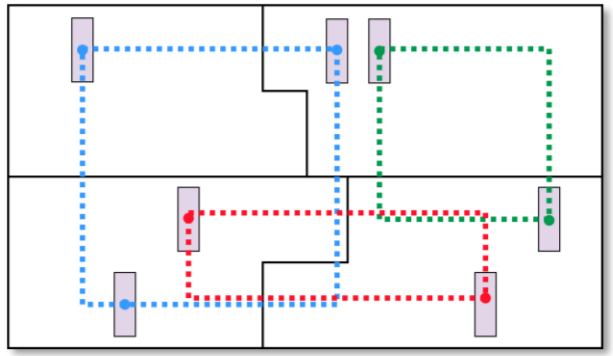


Legal transistor placement
Initial solution (hints)



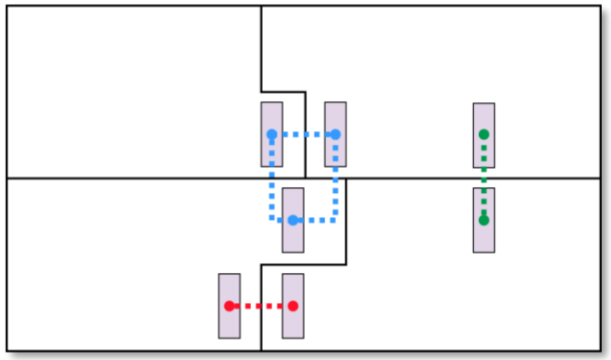
Motivation | Escape-Pin Location

- TransRoute [DAC'25] overlooks escape-pin location optimization
- Minimize escape-net HPWL
  - Reduce upper-layer routing

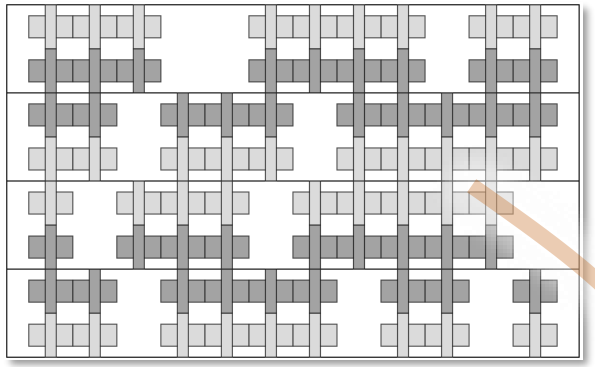


Without escape-net HPWL

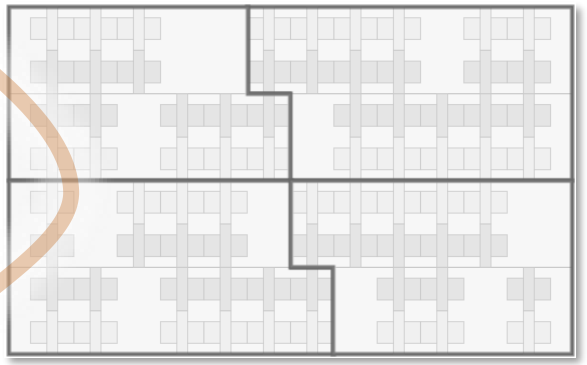
With escape-net HPWL

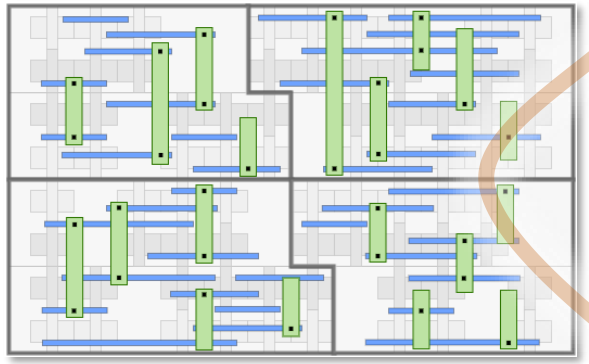


Transistor placement

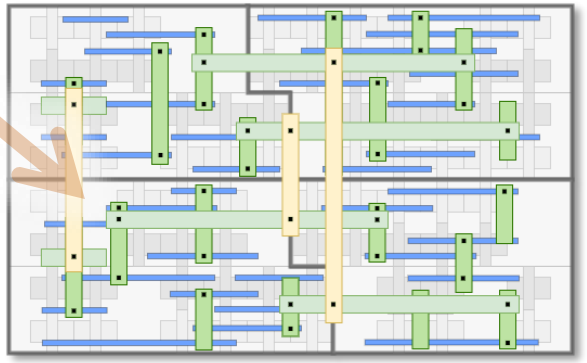


Layout partitioning



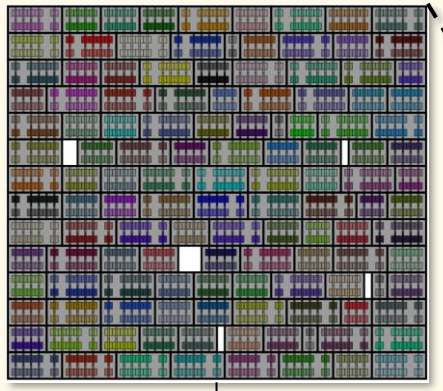


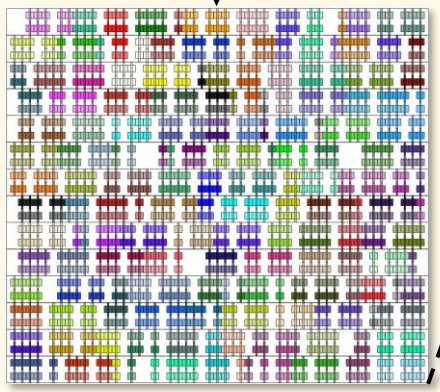
Lower-layer routing



Upper-layer routing

TransOpt: Transistor-Level P&R Optimization Flow





Netlist

Design Canvas

Design Rules

Placement

Standard Cell Placement

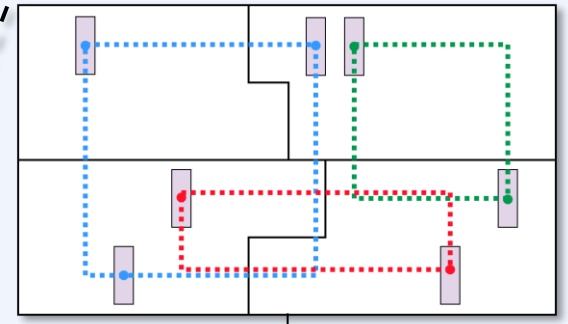
Transistor-Level Placement Optimization

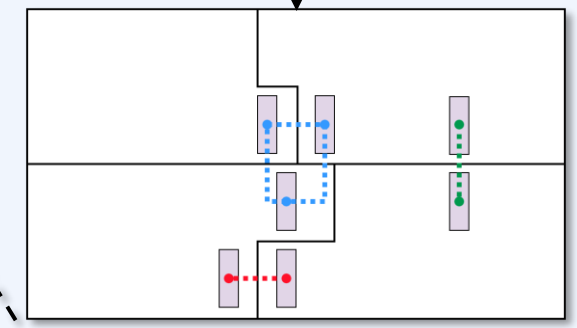
Routing

Transistor-Level Routing

Transistor-Level Routing Optimization

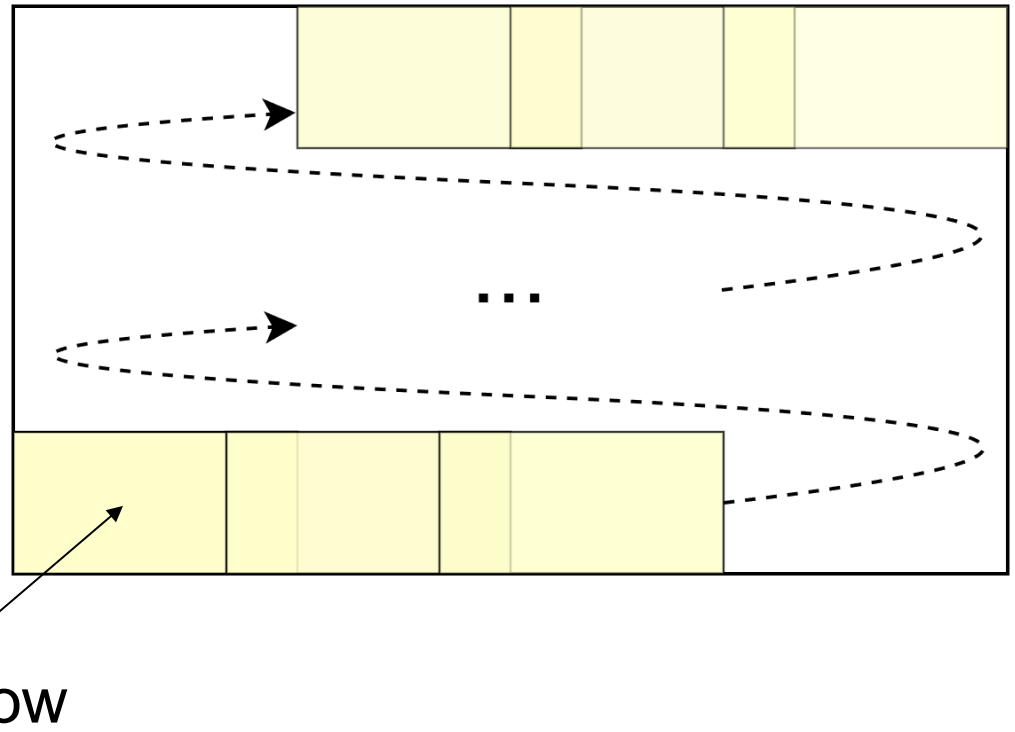
GDSII Layout





PlaceOpt Strategy

- Use sliding windows
  - Unlike TransPlace [ASPDAC'24]: random windows
- Window size
  - 4 SDC rows by 20 sites
- Shifting direction
  - 10 sites rightward
  - 2 rows upward
- Use existing transistor placement as hints to CP-SAT solver

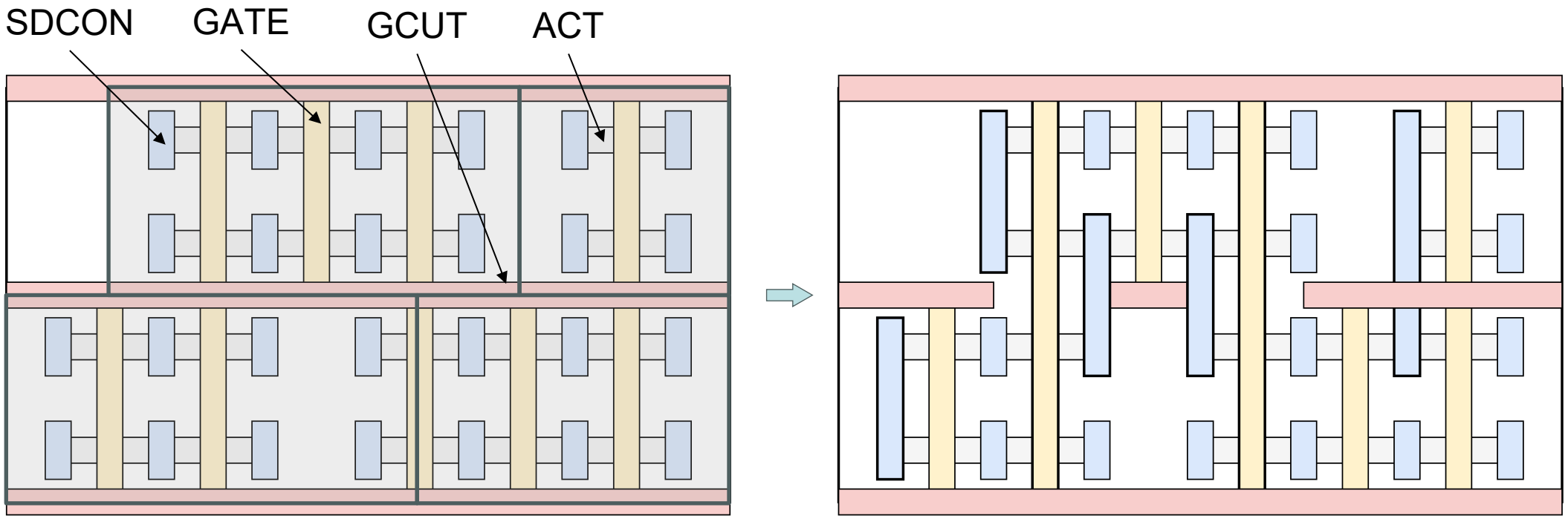


CP-SAT Formulation for PlaceOpt

- Unlike TransPlace [ASPDAC'24]
  - Pure SAT formulation + incremental SAT solving for optimization
- Constraints (adopt TransPlace [ASPDAC'24] formulation)
  - Transistor placement constraint
  - Non-overlapping constraint
  - Diffusion sharing constraint
  - Transistor pairing constraint (Optional)
- Objectives $\phi_{place} = \alpha \cdot \boldsymbol{\phi_{HPWL}} - \beta \cdot \boldsymbol{\phi_{sharing}} - \gamma \cdot \boldsymbol{\phi_{direct}}$
  - Total HWPL $\boldsymbol{\phi_{HPWL}}$
  - Diffusion sharing count $\boldsymbol{\phi_{sharing}}$
  - Direct connection count $\boldsymbol{\phi_{direct}}$

Direct Connection

- Poly layer: GATE in GT3
- Metal-to-diffusion (MD) layer: SDCON in GT3
- Reduce routing demand on metal layers



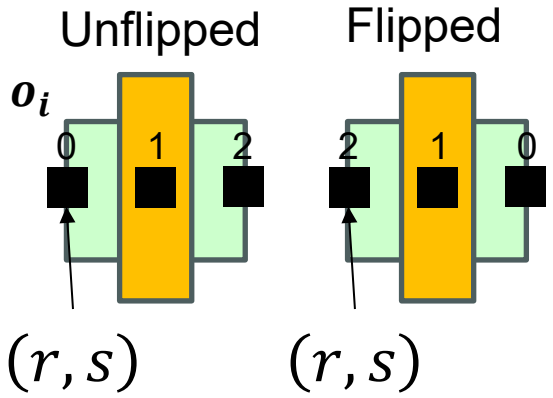
No direct connection

Apply direct connection

Linear Expression of Pin Location

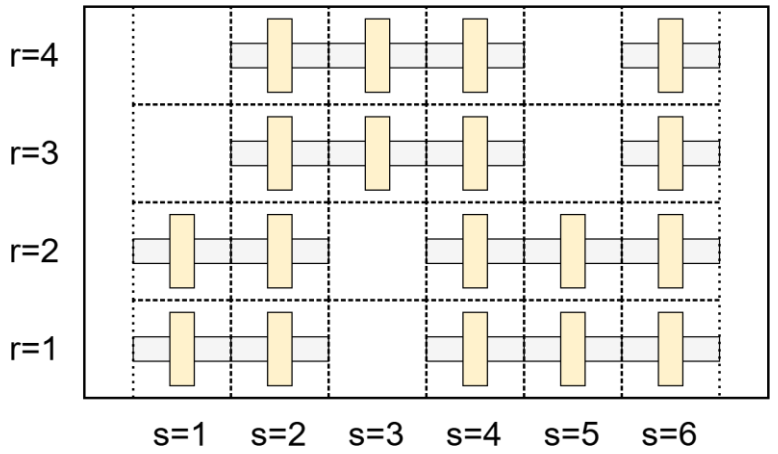
- Pin i's x location (integer)

$$x_i = \underbrace{\left(\sum_{(r,s) \in L_t} 2s \cdot l^t_{r,s} \right)}_{\text{Transistor } x \text{ location}} + \underbrace{(o_i \cdot \neg f_t \cdot + (2 - o_i) \cdot f_t)}_{\text{Pin } x \text{ location offset}}$$



- Pin i's y location (integer)

$$y_p = \underbrace{\left(\sum_{(r,s) \in L_t} r \cdot l^t_{r,s} \right)}_{\text{Transistor } y \text{ location}}$$



Placement Objective | HPWL

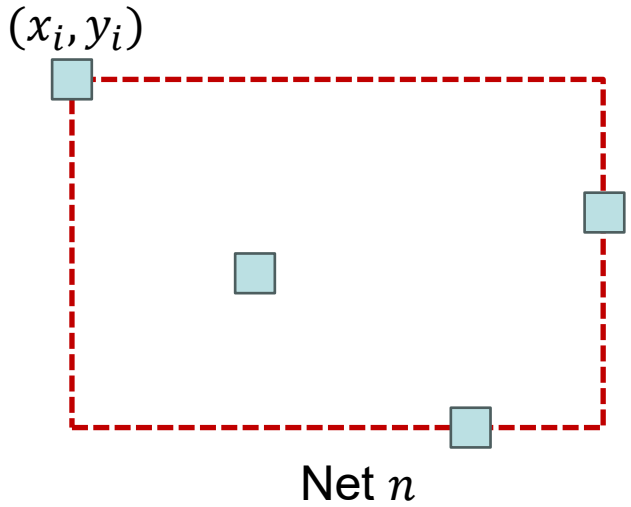
- Net n's HPWL

$$\boldsymbol{HPWL_n} = \left(\max_{i \in n} x_i - \min_{i \in n} x_i\right) + \left(\max_{i \in n} y_i - \min_{i \in n} y_i\right)$$

  - Create new integer variables for min/max values
  - CP-SAT solver provides *MinEquality* and *MaxEquality* functions

- Total HPWL
  - Sum of all nets' HPWL

$$\sum_{n \in N} \boldsymbol{HPWL_n}$$

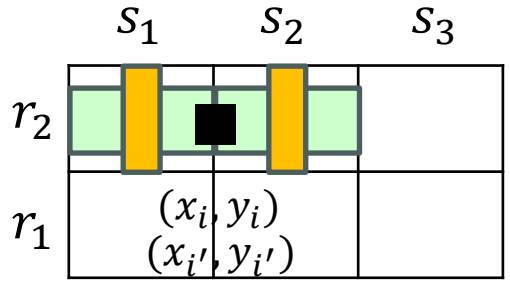


Placement Objective | Diffusion Sharing & Direct Connection

- Diffusion sharing count $\phi_{sharing}$
  - For each net, check **every pair of pins** (i, i') if their locations are identical

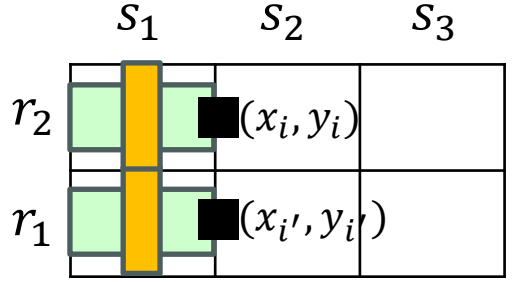
$$\boldsymbol{\phi_{sharing}} = \frac{1}{2}\sum_{n\in N}\sum_{\substack{i,i'\in n \\ i\neq i'}}((x_i = x_{i'}) \wedge (y_i = y_{i'}))$$

symmetry

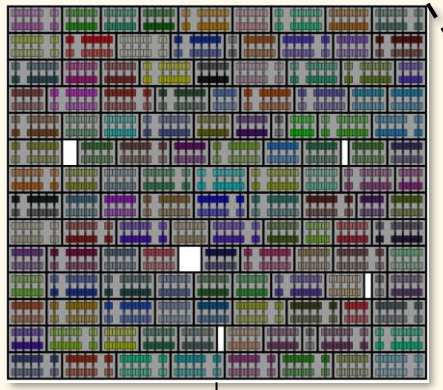


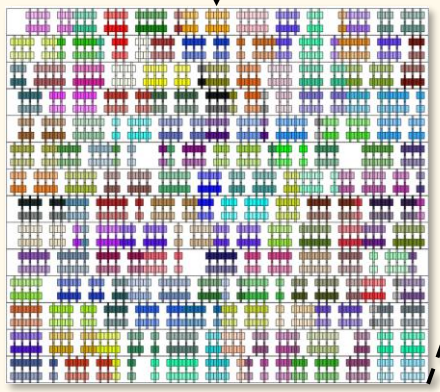
- Direct connection count ϕ_{direct}
  - For each net, check **every pair of pins** (i, i') if their x-coordinates are identical and y-coordinates differ by one

$$\boldsymbol{\phi_{direct}} = \sum_{n\in N}\sum_{\substack{i,i'\in n \\ i\neq i'}}((x_i = x_{i'}) \wedge (y_i = y_{i'} + 1))$$



TransOpt: Transistor-Level P&R Optimization Flow





Netlist

Design Canvas

Design Rules

Placement

Standard Cell Placement

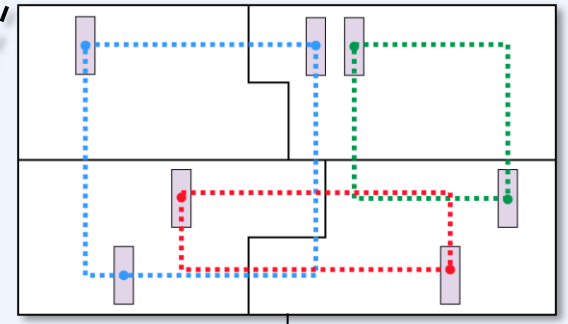
Transistor-Level Placement Optimization

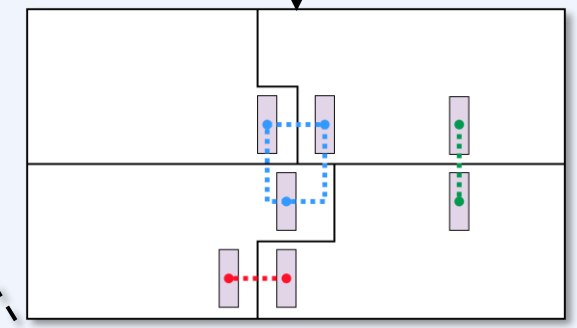
Routing

Transistor-Level Routing

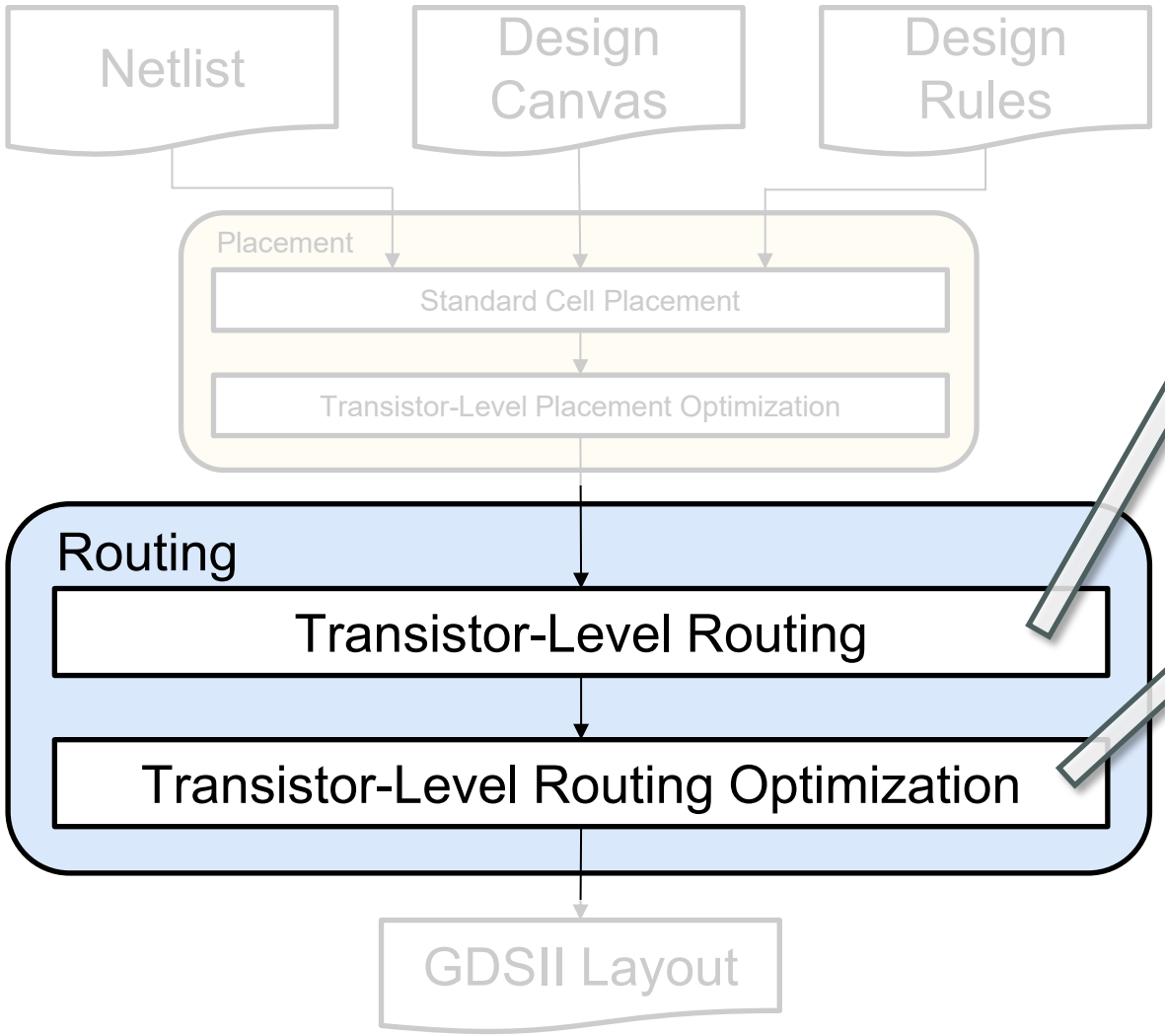
Transistor-Level Routing Optimization

GDSII Layout





RouteOpt Strategy

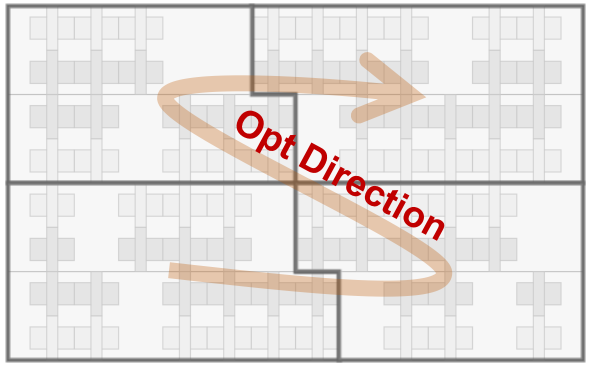


Parallel

- All partitions are **independent**
- No escape-net HPWL objective

Sequential

- Use existing routing as **hints**
- Partition-by-partition optimization

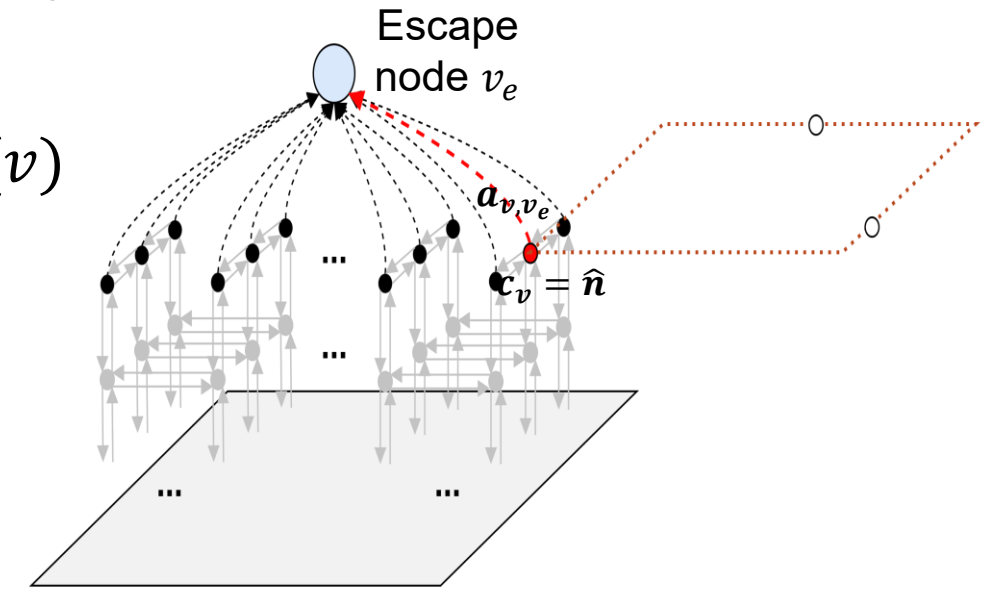


Routing Objective

- Routing objective $\phi_{route} = \underbrace{\alpha' \cdot \boldsymbol{\phi_{cost}}}_{\text{Routing cost [DAC'25]}} + \underbrace{\beta' \cdot \boldsymbol{\phi_{esc}}}_{\textbf{Escape-net HPWL}}$
- Escape variable s_v^n
  - Net n escapes from vertex v: $\boldsymbol{s_v^n} \rightarrow (a_{v,v_e} \wedge (c_v = \hat{n}))$

Arc (v, v_e) selected; Vertex v's encoding; Net n encoding

- Dynamic escape location (x_n^p, y_n^p)
  - $\boldsymbol{x_n^p} = \sum_{v \in V_{esc}^p} \boldsymbol{s_v^n} \cdot x(v)$; $\boldsymbol{y_n^p} = \sum_{v \in V_{esc}^p} \boldsymbol{s_v^n} \cdot y(v)$
- Escape-net HPWL ϕ_{esc}
  - $\phi_{esc} = \sum_{n \in E_p} ((\max_{p \in P_n} \boldsymbol{x_n^p} - \min_{p \in P_n} \boldsymbol{x_n^p}) + (\max_{p \in P_n} \boldsymbol{y_n^p} - \min_{p \in P_n} \boldsymbol{y_n^p}))$



Benchmarks

- Open-source 3nm GAAFET PDK: **GT3**
- Largest design: **5,503** SDCs, **38,838** transistors, **440** partitions
- Combinational circuits, high SDC utilization

Benchmark	#SDCs	#SDC Nets	SDC Util. (%)	#Tran.	#Tran. Nets	Tran. Util. (%)	#Partitions
adder8	**28**	44	97.13	**252**	142	72.41	3
adder16	**64**	96	98.38	**538**	301	72.70	5
mul4	**65**	73	97.04	**420**	218	69.08	5
adder32	**128**	192	98.94	**1114**	621	73.68	14
mul8	**318**	334	98.93	**2132**	1082	71.07	21
adder64	**256**	384	98.44	**2266**	1261	73.57	20
mul16	**1364**	1396	98.52	**9272**	4668	70.67	100
mul32	**5503**	5567	98.88	**38838**	19483	71.41	440

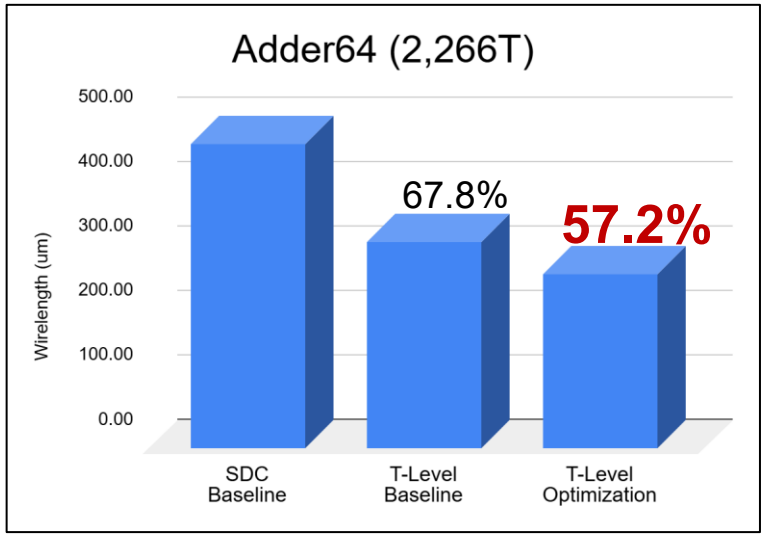
Overall Quality Comparison

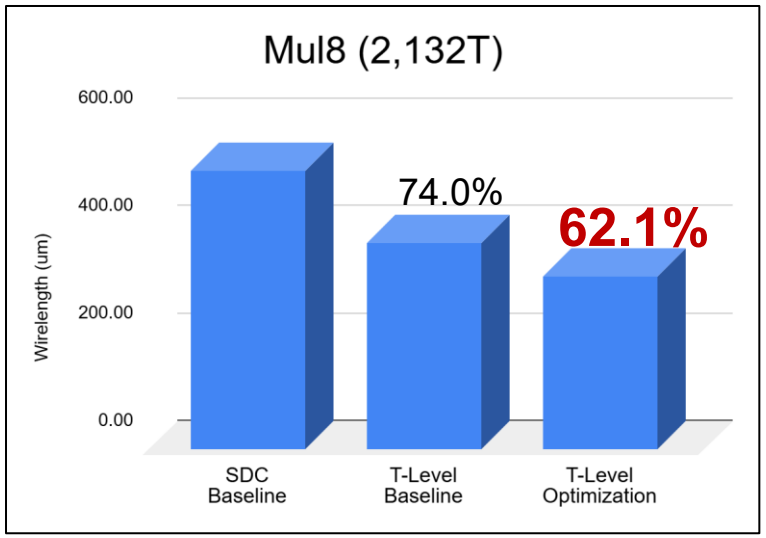
- Average **40%** wirelength reduction, **34%** via count reduction
- All LVS/DRC clean

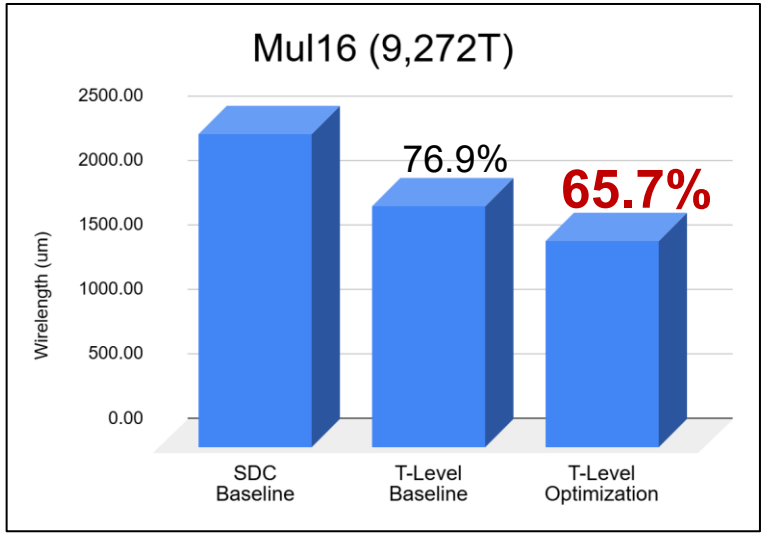
Benchmark	Total Wirelength (μm)					Total Via Count				
	SDC Baseline	T-Level Baseline	RouteOpt Only	PlaceOpt Only	**PlaceOpt + RouteOpt**	SDC Baseline	T-Level Baseline	RouteOpt Only	PlaceOpt Only	**PlaceOpt + RouteOpt**
adder8	50.09	32.55	30.59	26.38	**25.80**	658	461	444	411	**410**
mul4	98.30	65.27	62.79	60.15	**57.30**	1218	779	771	780	**767**
adder16	106.57	73.55	69.12	64.96	**60.46**	1418	1027	1010	966	**901**
adder32	235.95	164.42	153.79	136.97	**131.78**	2995	2218	2144	2054	**1990**
mul8	517.12	382.57	363.42	343.01	**320.91**	6318	4377	4370	4231	**4094**
adder64	470.79	319.12	301.42	281.81	**269.14**	6030	4359	4186	4186	**4055**
mul16	2419.53	1859.46	1778.12	1664.35	**1590.82**	27606	19773	19533	19004	**18479**
mul32	12535.69	10084.14	9711.38	9782.12	**9438.50**	119222	87754	**86662**	88392	87205
Avg. Ratio	1.00	0.71	0.68	0.63	**0.60**	1.00	0.71	0.69	0.68	**0.66**

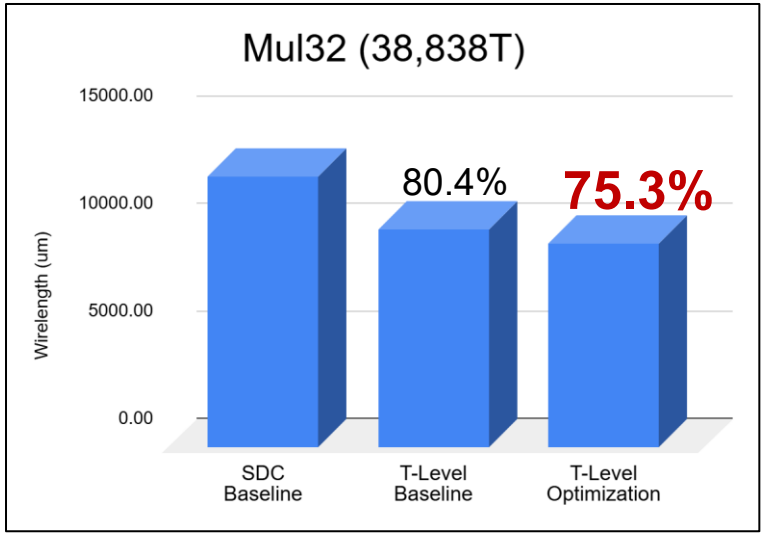
Comparison of Wirelength

- **SDC Baseline**
  SDC P&R (Traditional Flow)
- **T-Level Baseline**
  SDC Placement + T-Level Routing
- **T-Level Optimization**
  SDC Placement + **PlaceOpt**
  + T-Level Routing + **RouteOpt**



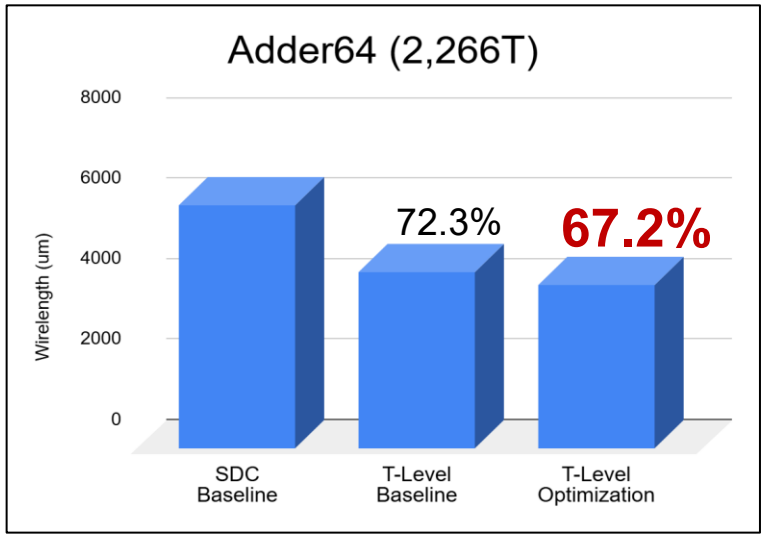


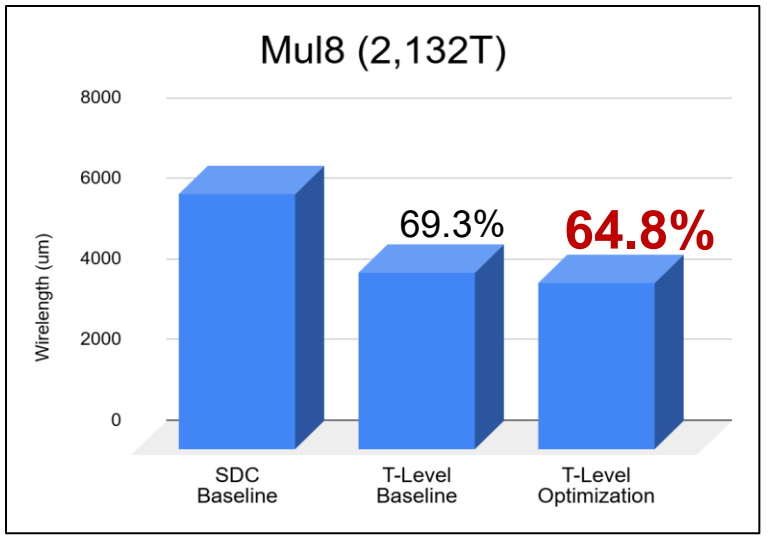


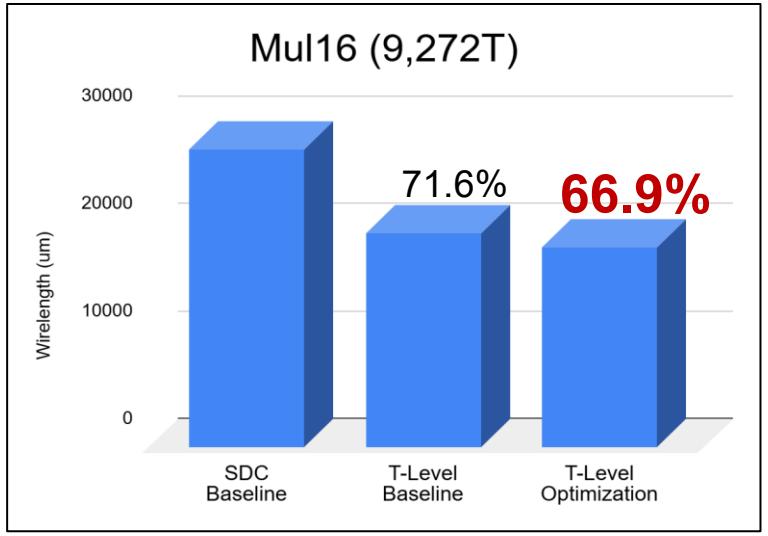


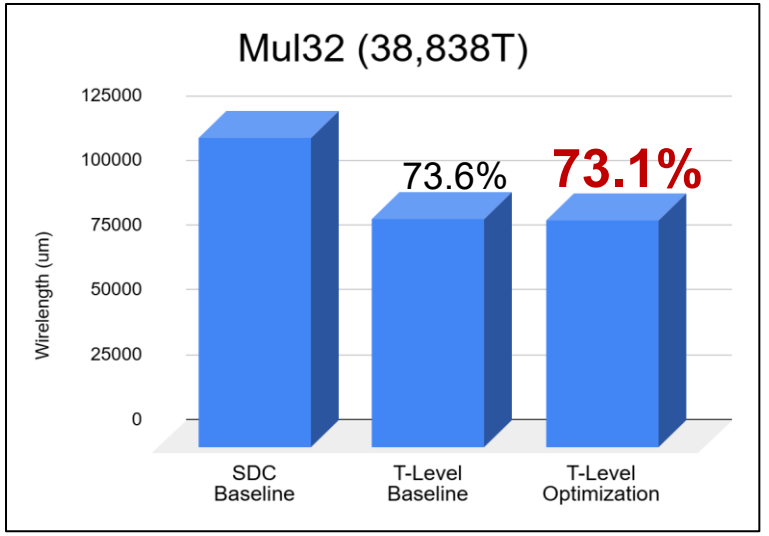
Comparison of Via Count

- **SDC Baseline**
  SDC P&R (Traditional Flow)
- **T-Level Baseline**
  SDC Placement + T-Level Routing
- **T-Level Optimization**
  SDC Placement + **PlaceOpt**
  + T-Level Routing + **RouteOpt**



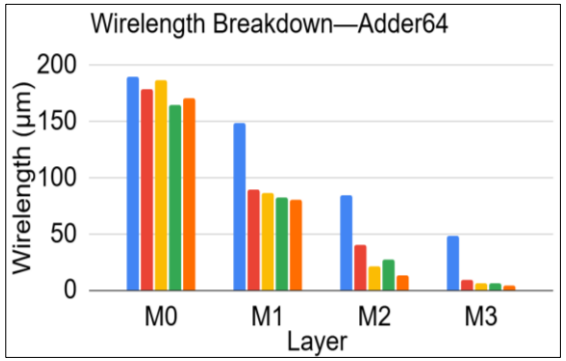


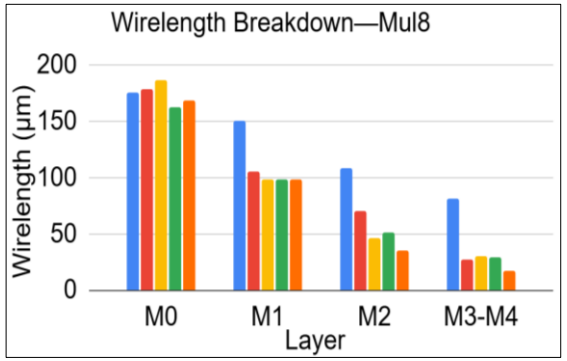


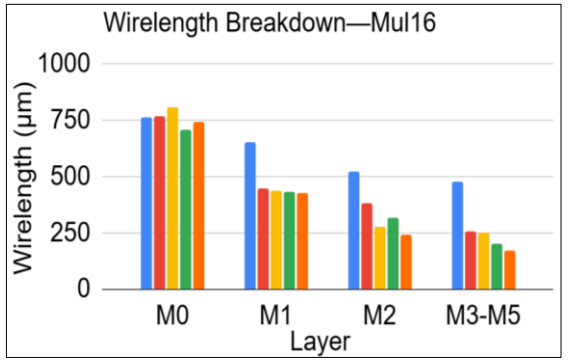


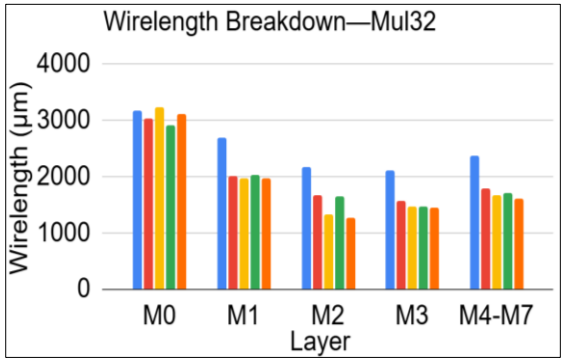
Wirelength & Via Count Breakdown by Layer

- **Wirelength**: reduced upper-layer routing (M2+)

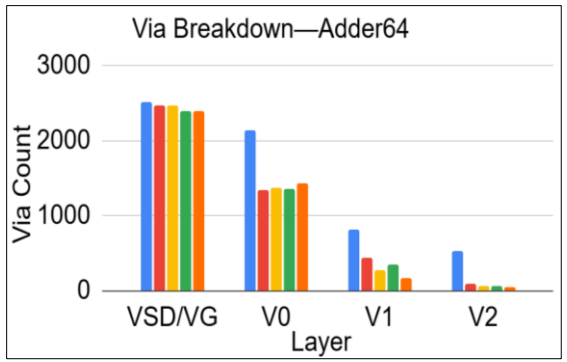


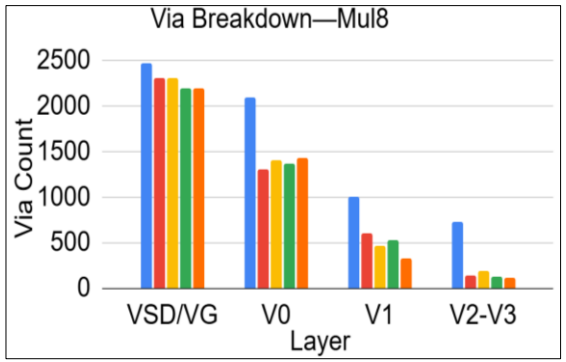


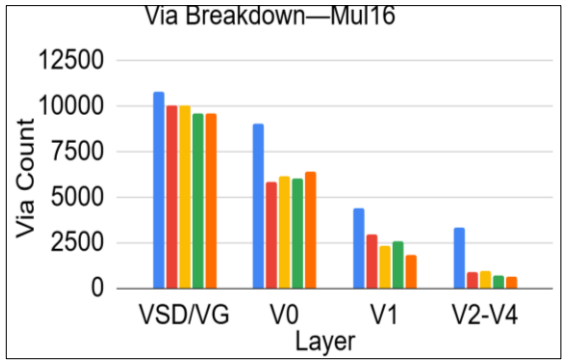


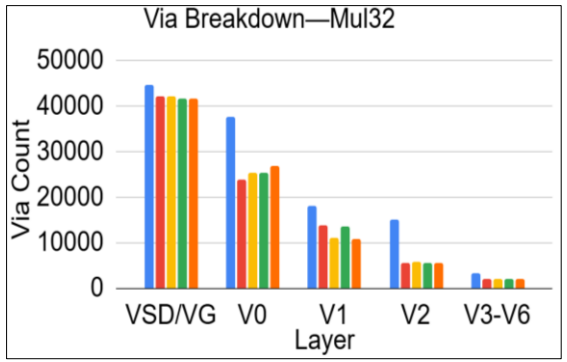


- **Via Count**: reduced VSD/VG



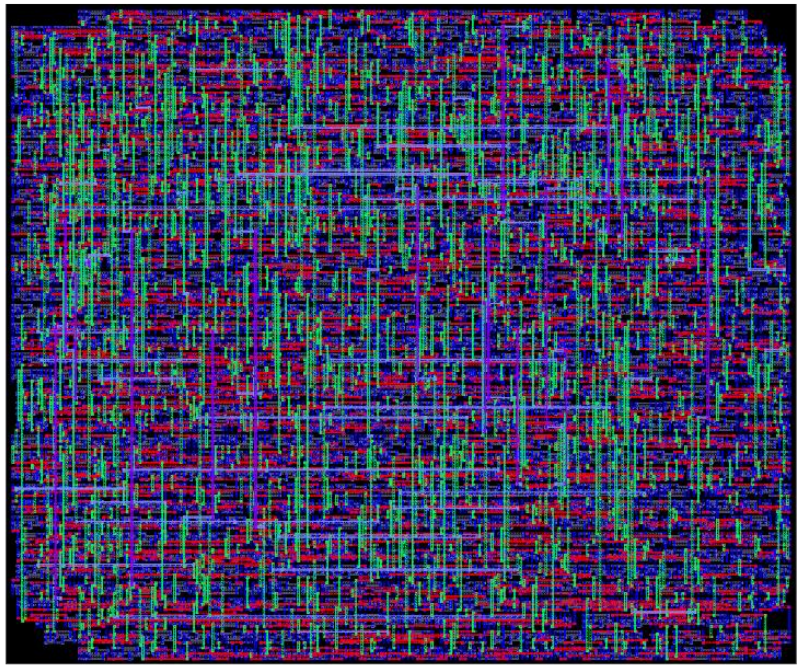




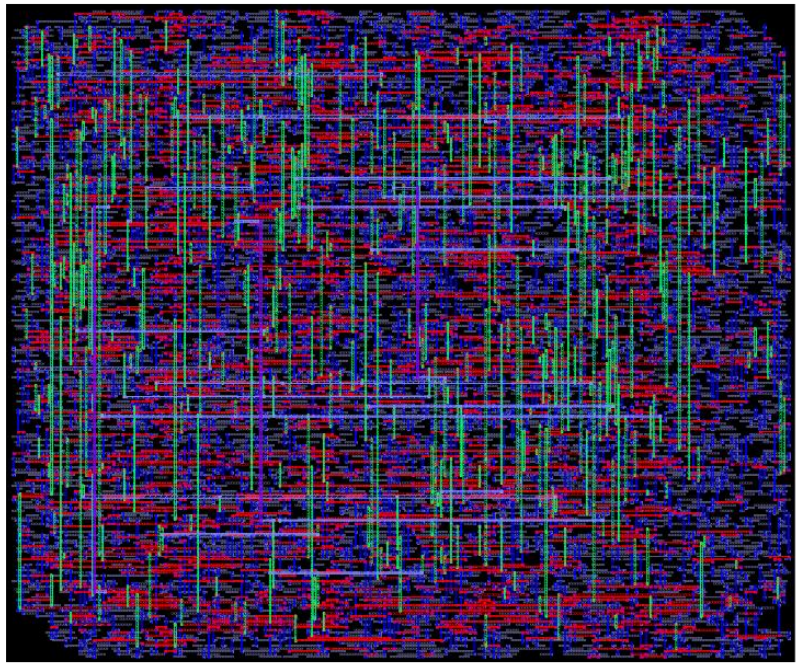


SDC Baseline T-Level Baseline RouteOpt Only PlaceOpt Only PlaceOpt & RouteOpt

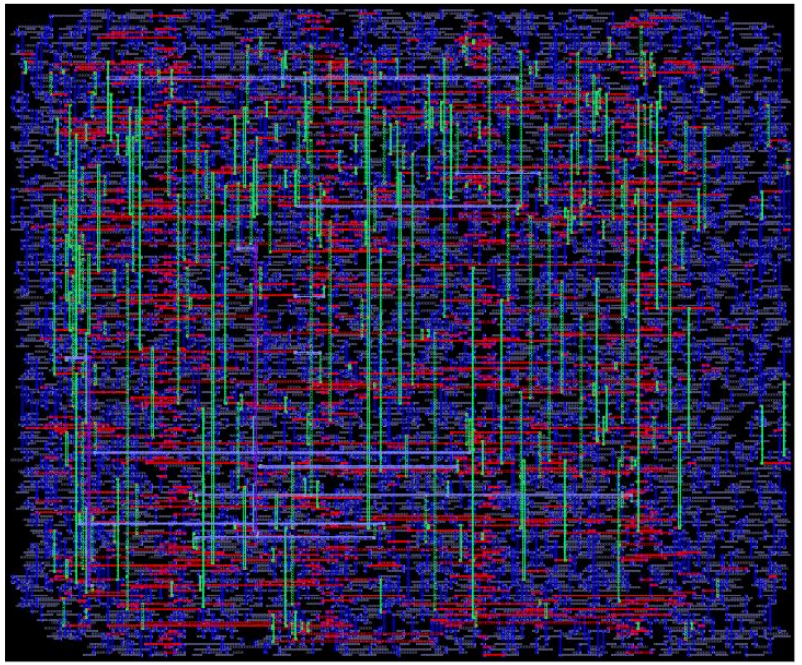
Layout Visualization | Mul16



SDC Baseline



T-Level Baseline

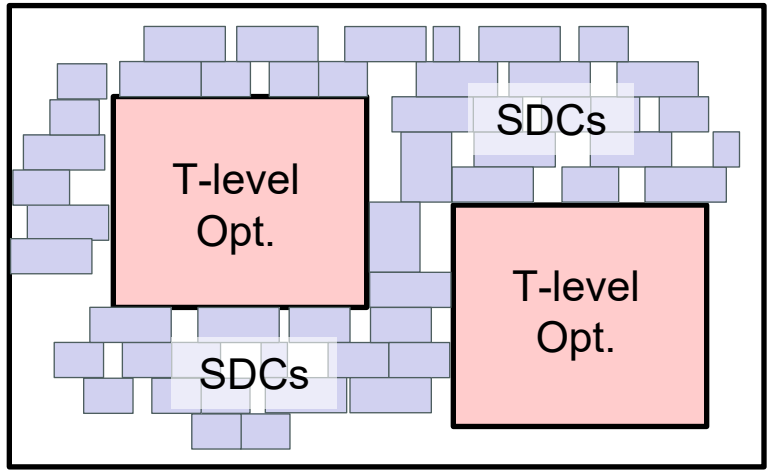


T-Level Optimization (PlaceOpt & RouteOpt)

Significant wirelength reduction!

Conclusion

- We present TransOpt, a transistor-level P&R optimization framework
  - Transistor placement optimization for a given SDC placement
    - Total HPWL, diffusion sharing, direct connection
  - Escape pin location optimization to reduce upper-layer routing demand
    - Total routing cost, escape-net HPWL
- Experimental results show significant wirelength/via reduction compared to SDC baselines
- Future direction: hybrid manner
  - Perform T-level optimization to critical regions
  - Keep other parts as standard cells



Thank You!