DEC 5 - 9, 2021 ◆ San Francisco, California

# Interactive Analog Layout Editing with Instant Placement Legalization

Xiaohan Gao[1], Mingjie Liu[2], David Z. Pan[2], Yibo Lin[1]

[1]Department of Computer Science, Peking University

[2]ECE Department, The University of Texas at Austin

# Analog/Mixed-Signal IC Design

- Analog design still relies heavily on manual efforts!
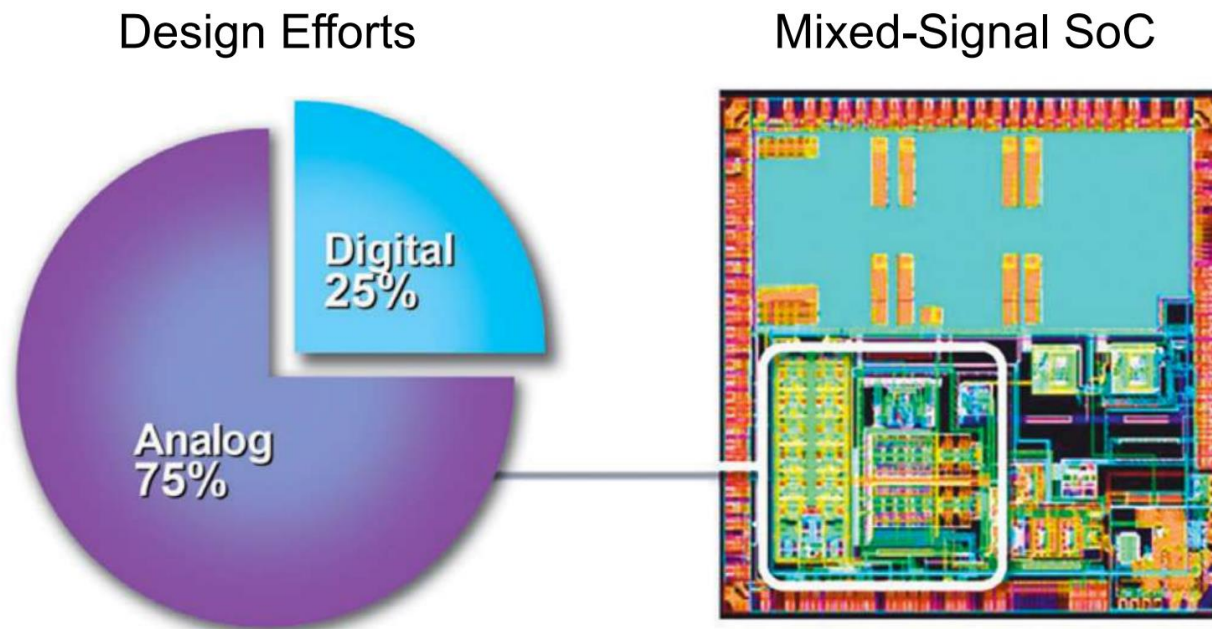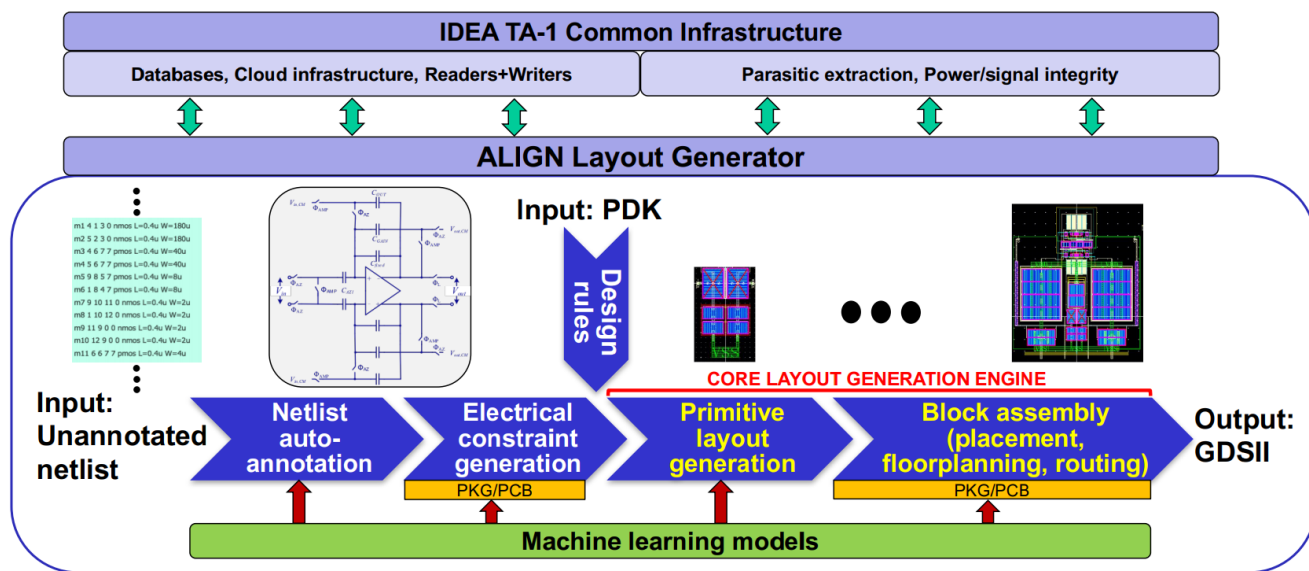
Design Efforts
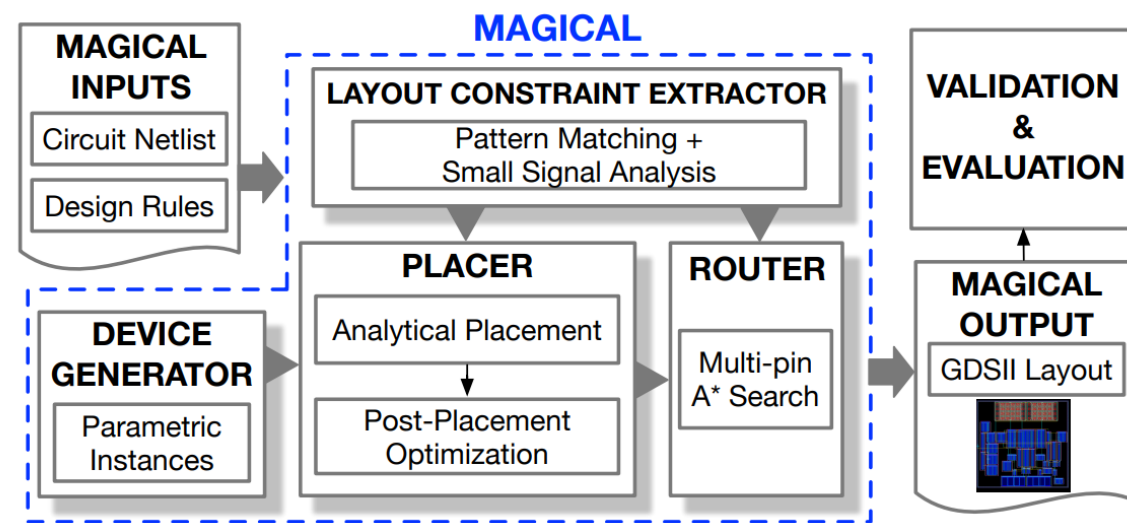
Mixed-Signal SoC



Figure from: "IBS and Dr. Handel Jones, 2012"

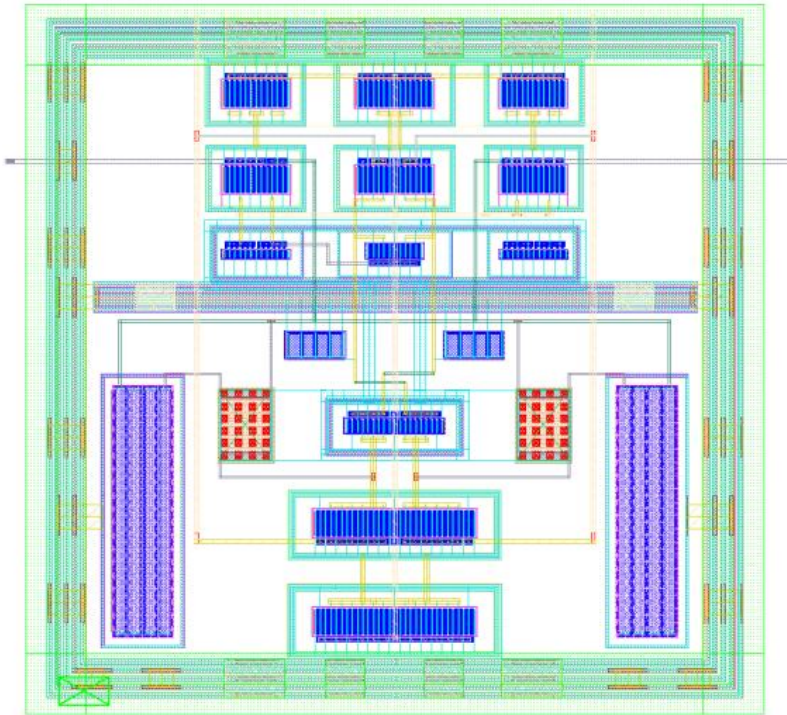# Fully-automated Analog Layout Generators
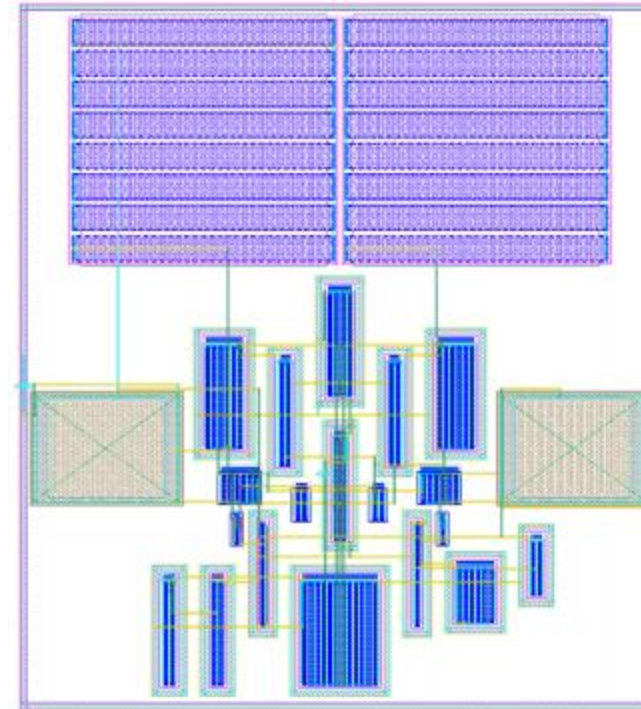


ALIGN framework



MAGICAL framework

Figure from: "Magical: Toward fully automated analog IC layout leveraging human and human intelligence";
"Align: Open-source analog layout automation from the ground up".

# Problem of Fully-automated Generation

- Fully automated flow is not flexible enough to satisfy customization demands



manual design

MAGICAL solution

Figure from: "Magical: Toward fully automated analog IC layout leveraging human and human intelligence"
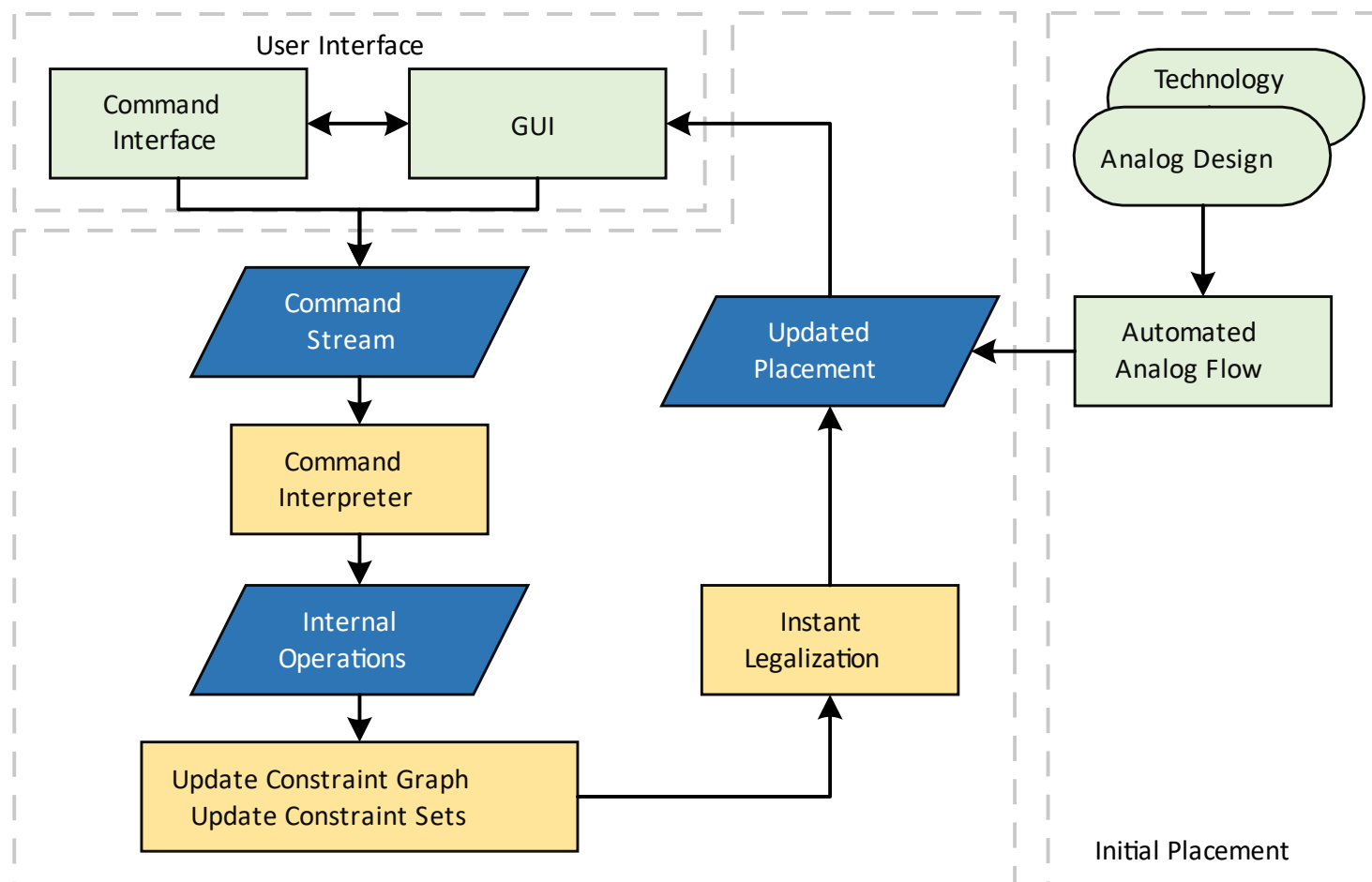
# Basic Idea of Interactive Layout Editing

- The fully-automated analog layout generators provide good <span style="color:red">initial solutions</span> for analog designs

- Designers have the freedom to <span style="color:red">interactively edit</span> the layout

- The framework tackles the detailed design rules: we can show real-time <span style="color:red">legalized layout</span> result

# Interactive Layout Editing Workflow

# Layout Editing Command

Requirements of layout editing command:

- Complete: a sequence of those commands on the initial solution can represent any layout

- Simple: the commands are supposed to be easy to use

- Expressive: The sequence of commands that designers convert a layout to another expected layout is mostly short

# Defined Command Set

- User command set
  - Fine-grained: topology-related commands
  - Coarse-grained: constraint-related commands

| Command | Parameters | Description |
|---------|-----------|-------------|
| move | device $v_i$, location $(x, y)$ | move a device to a location |
| spacing | devices $v_i, v_j$, spacing width $W$ | add spacing between devices |
| resize | shape $w, h$ | change the shape of a device |
| swap | devices $v_i, v_j$ | swap two devices |
| arrayAdd | devices $\{v_i\}$ | add array constraint |
| symAdd | devices $v_i, v_j$ sym axis $A_k$ | add symmetry constraint |

# Command Interpretation

- Internal data representation
  - constraint graph
  - constraint set

- Internal operation
  - `insert`
    insert a cell into constraint graph
  - `remove`
    remove a cell from constraint graph

- Command interpretation
  - user command: swap $v_i$ $v_j$
    internal operations: remove($v_i$) remove($v_j$) insert($v_i, p_i$) insert($v_j, p_j$)

# Mixed constraint graph

- Mixed constraint graph (mcg)
  - A combination of hcg and vcg
  - Special property with its topological sort
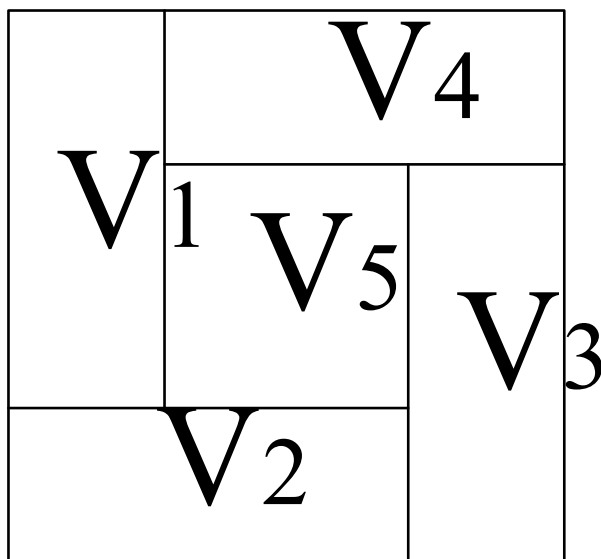


an example layout
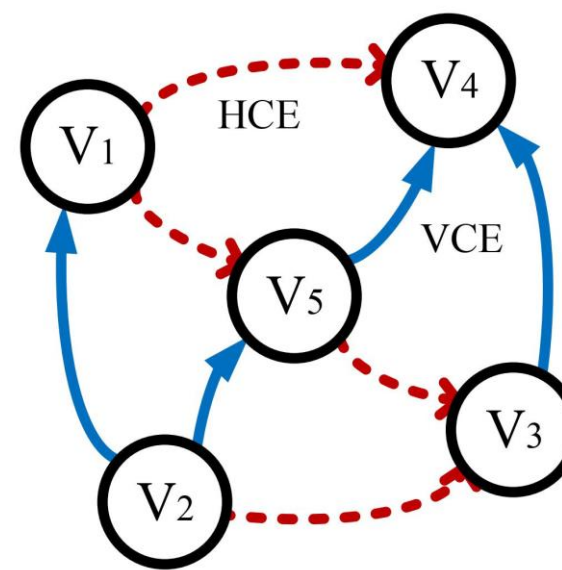


corresponding mcg

# Instant Legalization Problem Formulation

- Given an input analog placement solution with its mixed constraint graph and layout constraints, legalize the placement subject to the layout constraints with minimum perturbation to the layout and minimum runtime

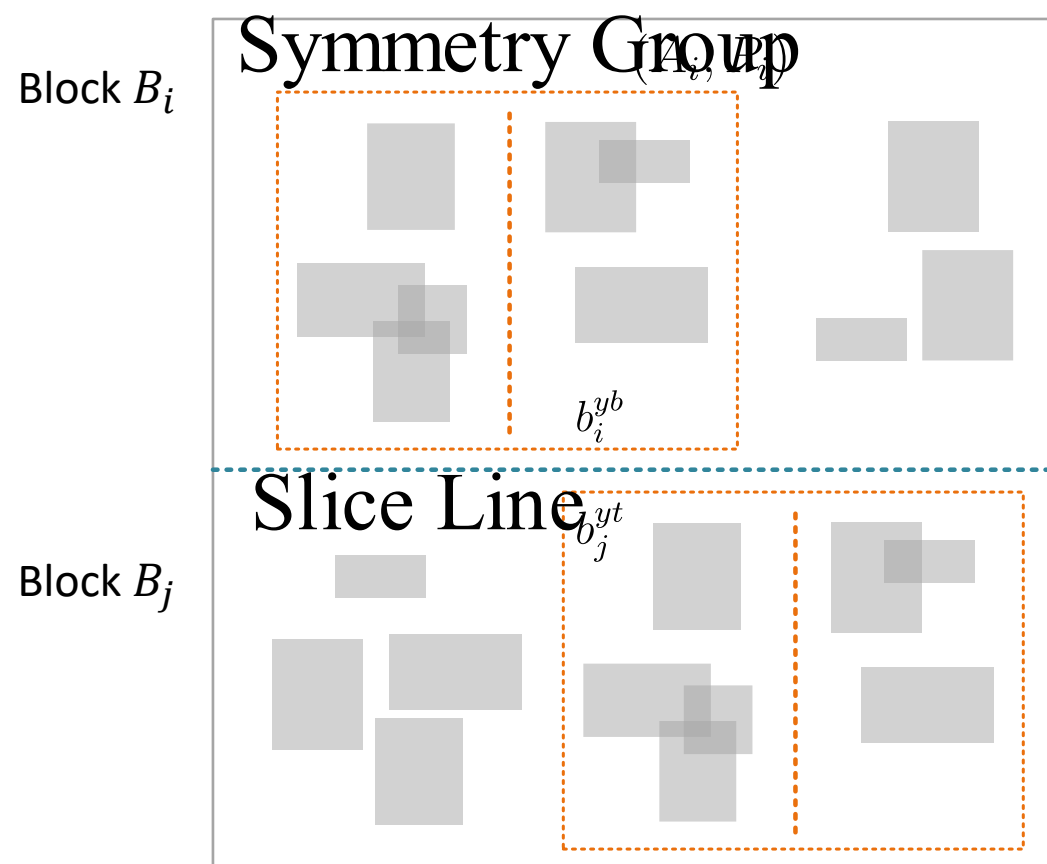# Instant Legalization Algorithm

- Two-stage legalization algorithm
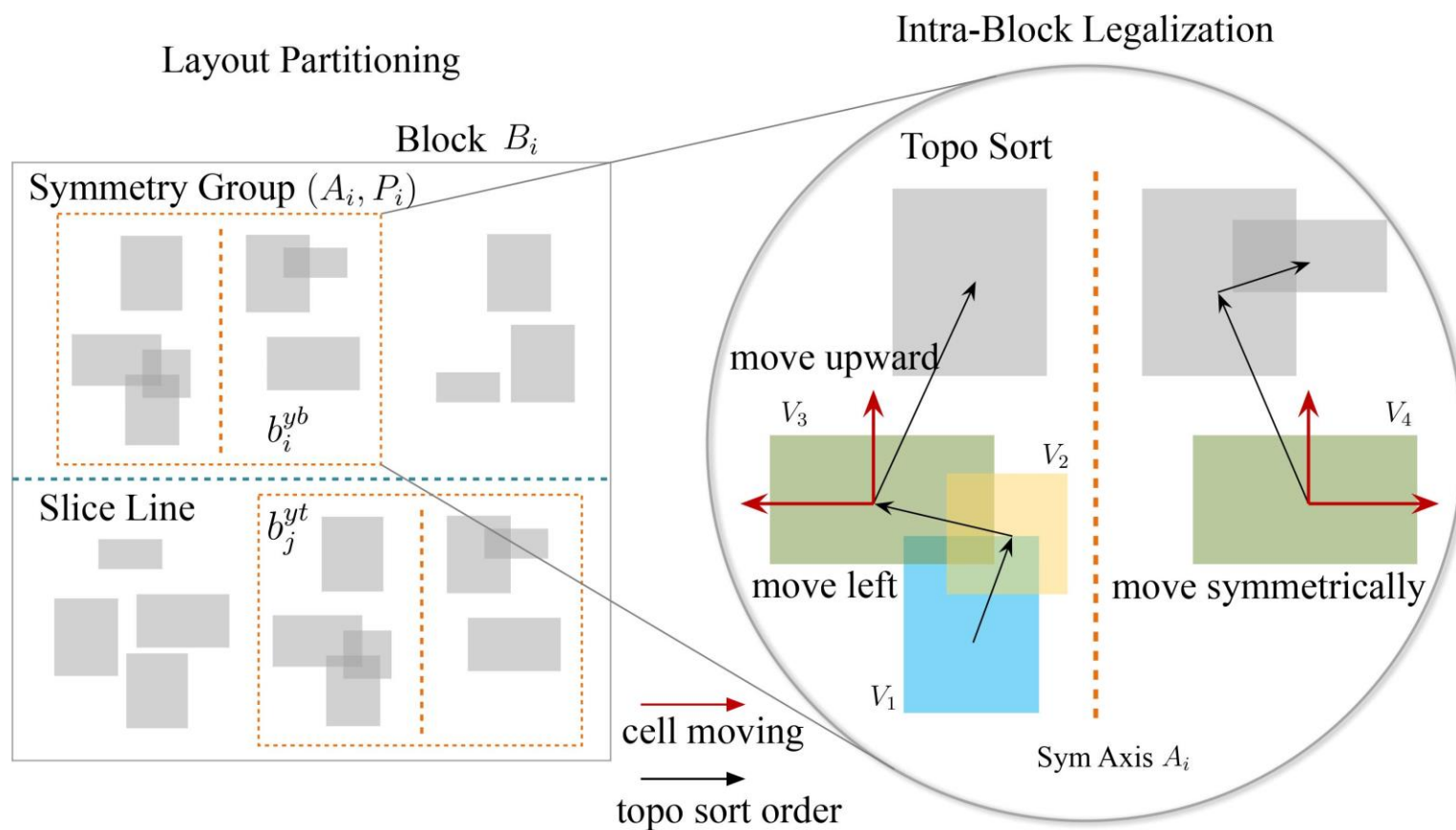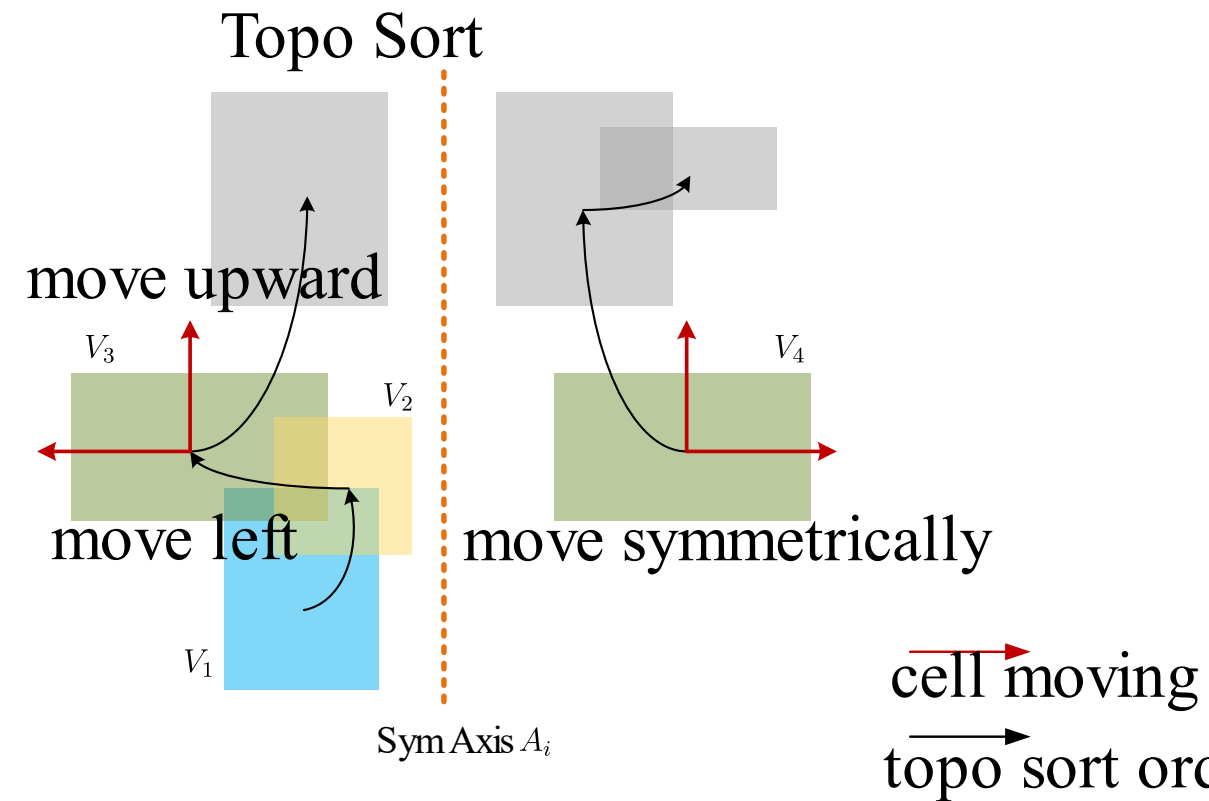
# Layout partitioning

- Layout partitioning
  - Calculate boundaries for constraint groups
  - Sort the boundaries
  - Add slicing lines
  - Divide the layout into grids
  - Cluster grids to blocks

Block $B_i$

Symmetry Group $(x_i, y_i)$

$b_i^{yb}$

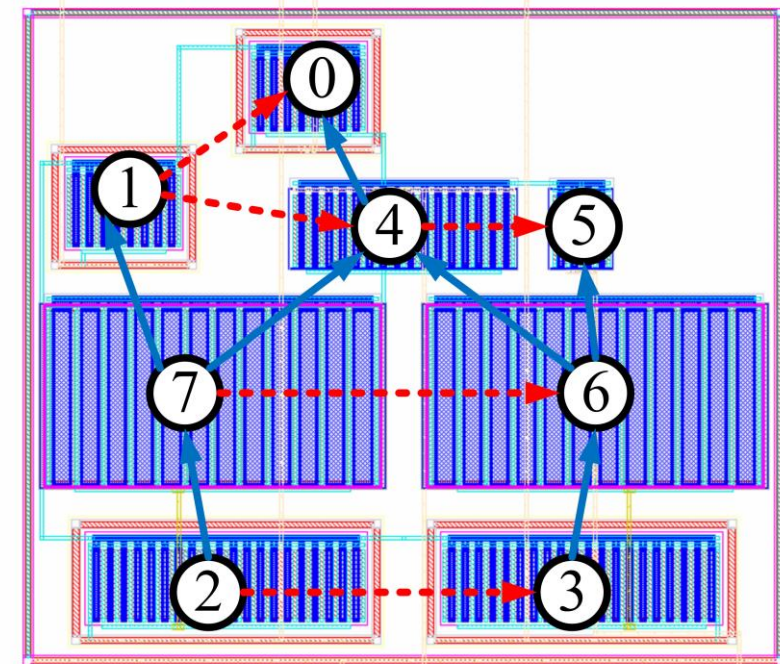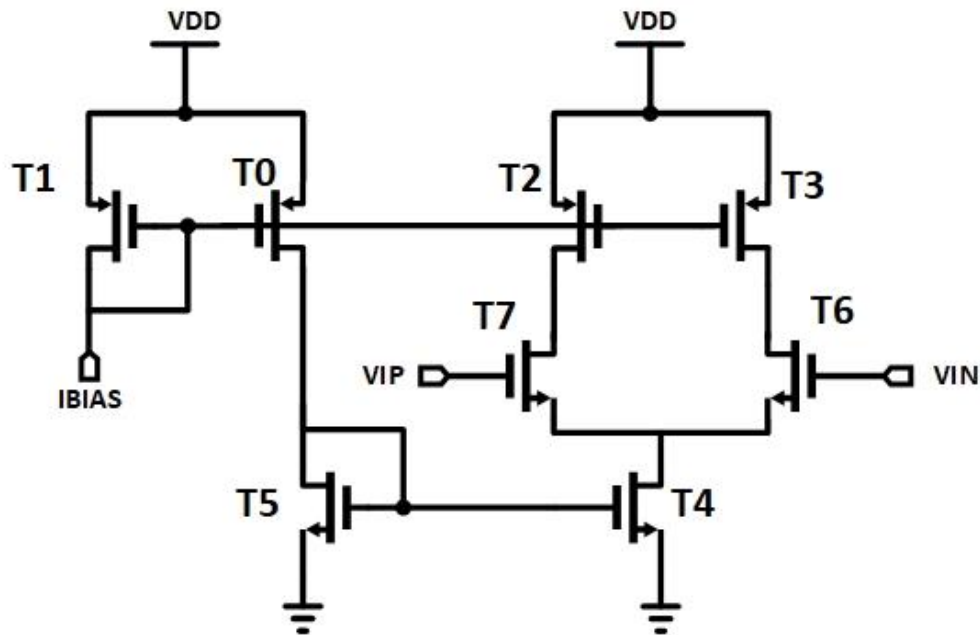Slice Line $b_j^{yt}$

Block $B_j$

# Topological-sort-based legalization

- Intra-block legalization
  - Mirror the mixed constraint graph
  - Compute the topological sort
  - Traverse along topo sort
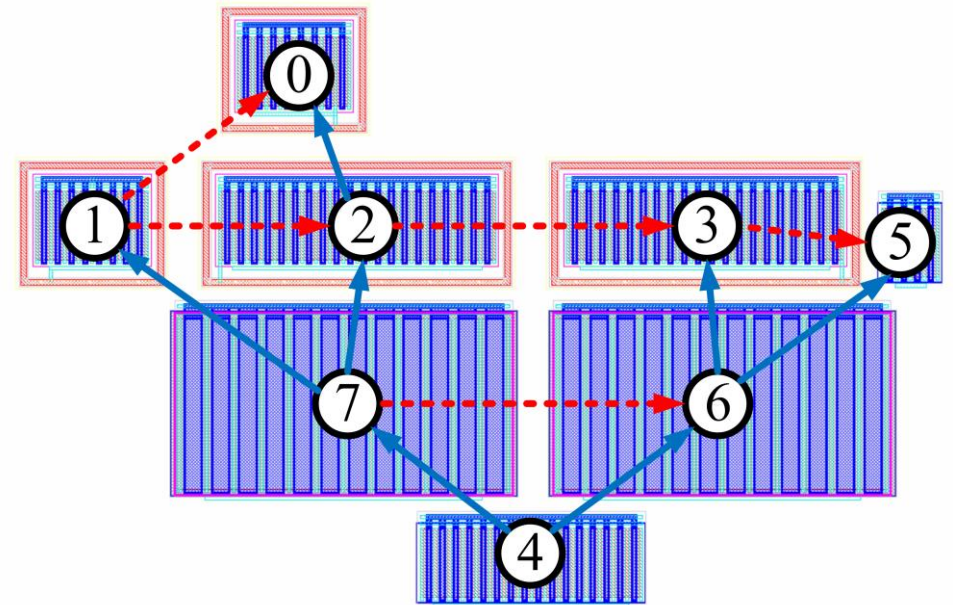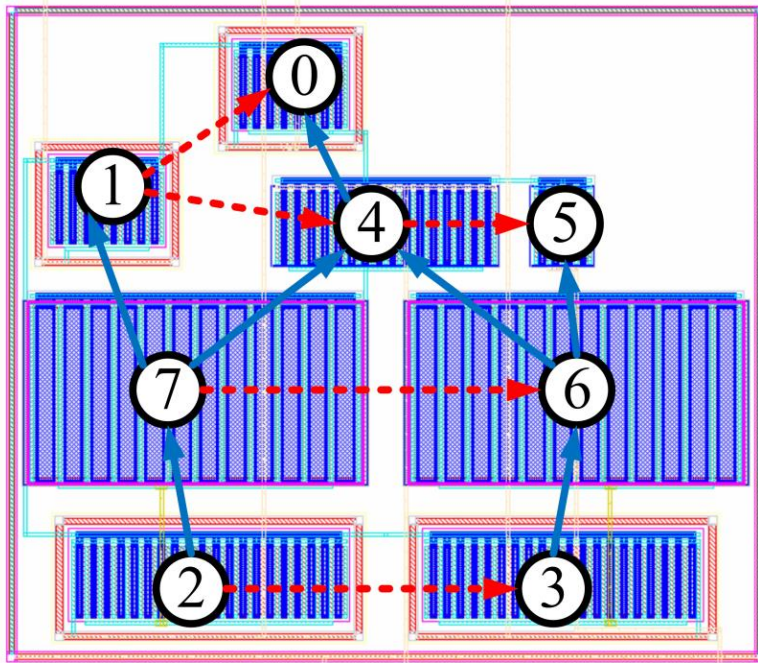  - move the devices to eliminate overlaps



Topo Sort

move upward

$V_3$

$V_2$

move left

$V_1$

move symmetrically

$V_4$

Sym Axis $A_i$

cell moving
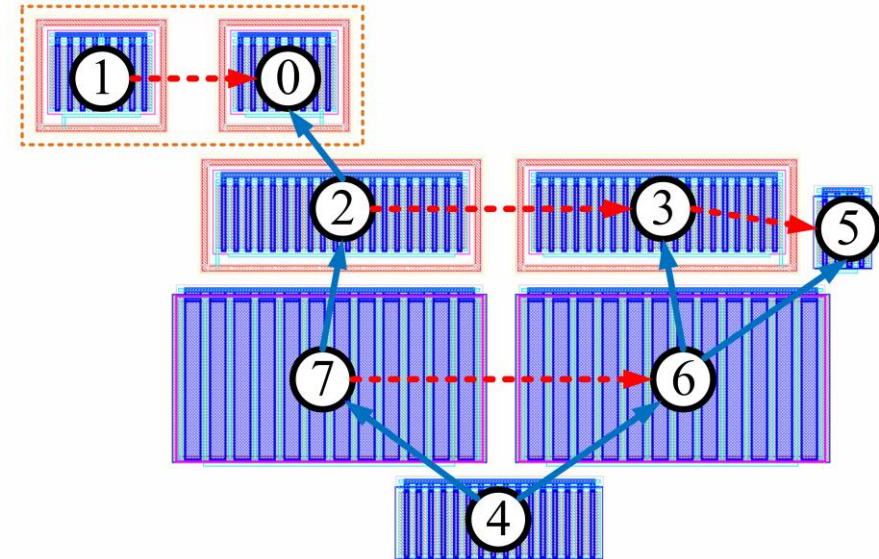
topo sort ord

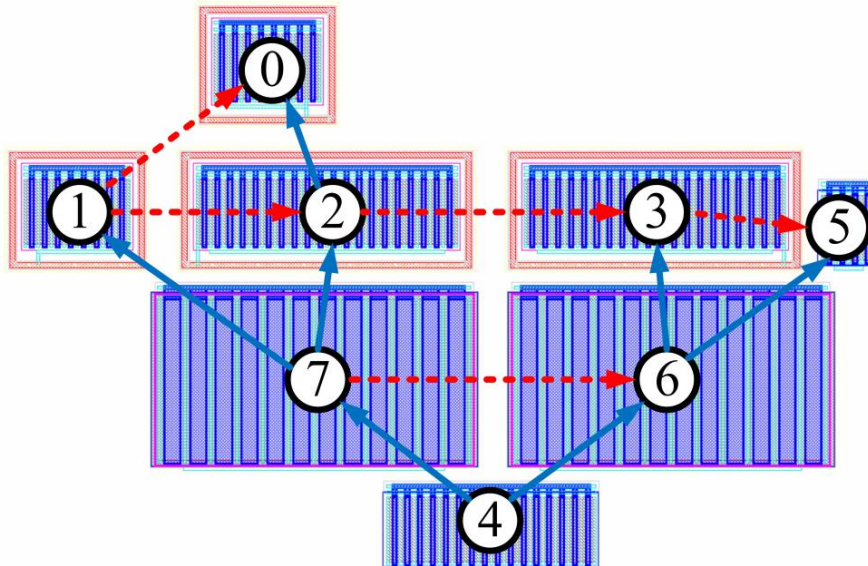# Interactive Layout Editing Process

- Editing process

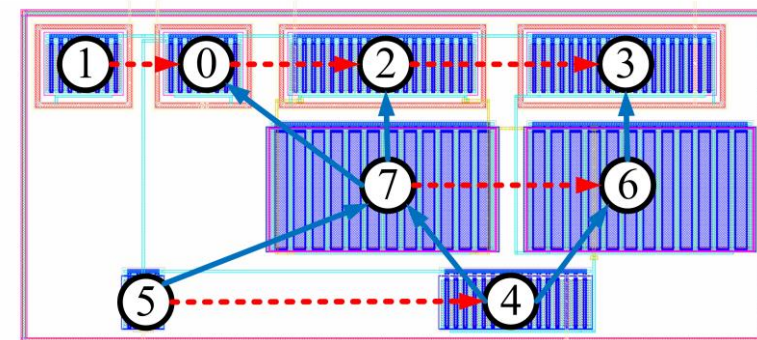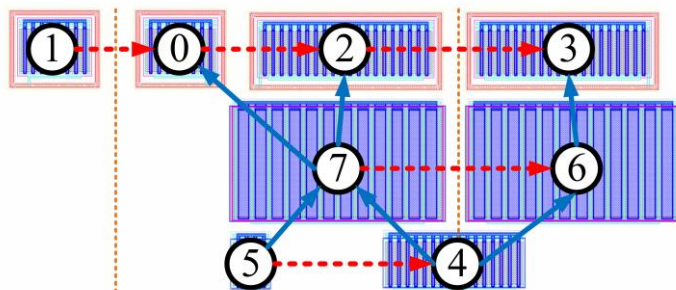# Interactive Layout Editing Process

- swap

# Interactive Layout Editing Process

- symAdd

# Interactive Layout Editing Process

- move
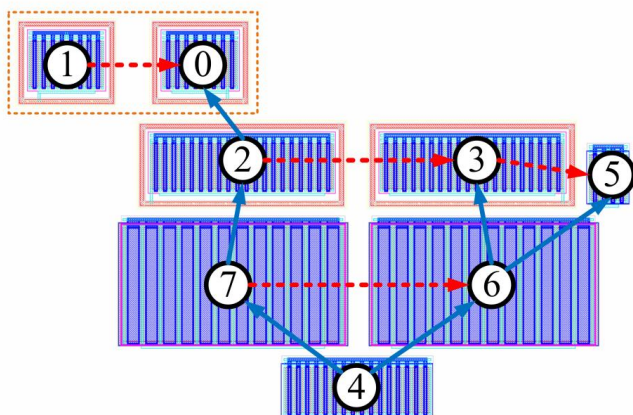
# Experimental Setup for Legalization

- Baseline
  - LP: we adapt the linear-programming-based legalizer from MAGICAL; the LP baseline directly legalize by minimizing total displacement

- Dataset statistics

| Circuit | OTA1 | OTA2 | OTA3 | COMP | ADC1 | ADC2 |
|---------|------|------|------|------|------|------|
| Devices | 25 | 49 | 42 | 17 | 114 | 211 |
| Hierarchy | - | - | - | - | ✓ | ✓ |

# Legalization Performance

| | TOPO | | | LP | | |
|---|---|---|---|---|---|---|
| | $T_{seq}$ (ms) | $D_{max}$ (%) | $D_{avg}$ (%) | $T_{seq}$ (ms) | $D_{max}$ (%) | $D_{avg}$ (%) |
| OTA1 | 0.16 | 31.1 | 17.7 | 33.3 | 40.9 | 13.0 |
| OTA2 | 0.23 | 43.4 | 5.1 | 33.4 | 43.4 | 5.1 |
| OTA3 | 0.20 | 10.9 | 3.99 | 35.4 | 16.2 | 3.3 |
| COMP | 0.12 | 11.8 | 4.20 | 31.5 | 12.6 | 3.9 |
| ADC1 | 0.84 | 14.1 | 10.1 | 101.7 | 16.4 | 8.0 |
| ADC2 | 1.06 | 21.9 | 11.1 | 253.7 | 19.6 | 6.1 |

- Better runtime
- Better maximal displacement

# Conclusions and Future Work

- Conclusions
  - A new paradigm of analog layout generation – layout editing
  - Instant legalization algorithm supporting real-time interaction
  - Novel layout topology representation – mixed constraint graph

- Future work
  - Support more design constraints
  - Explore layout editing with routing considered

# *Thanks!*
# *Questions are welcome*

Email: [xiaohangao@pku.edu.cn](mailto:xiaohangao@pku.edu.cn)