MAGICAL: Toward Fully Automated Analog IC Layout Leveraging Human and Machine Intelligence

Biying Xu, Keren Zhu, Mingjie Liu, Yibo Lin, Shaolan Li,
Xiyuan Tang, Nan Sun, and **David Z. Pan**

Dept. of Electrical and Computer Engineering

The University of Texas at Austin

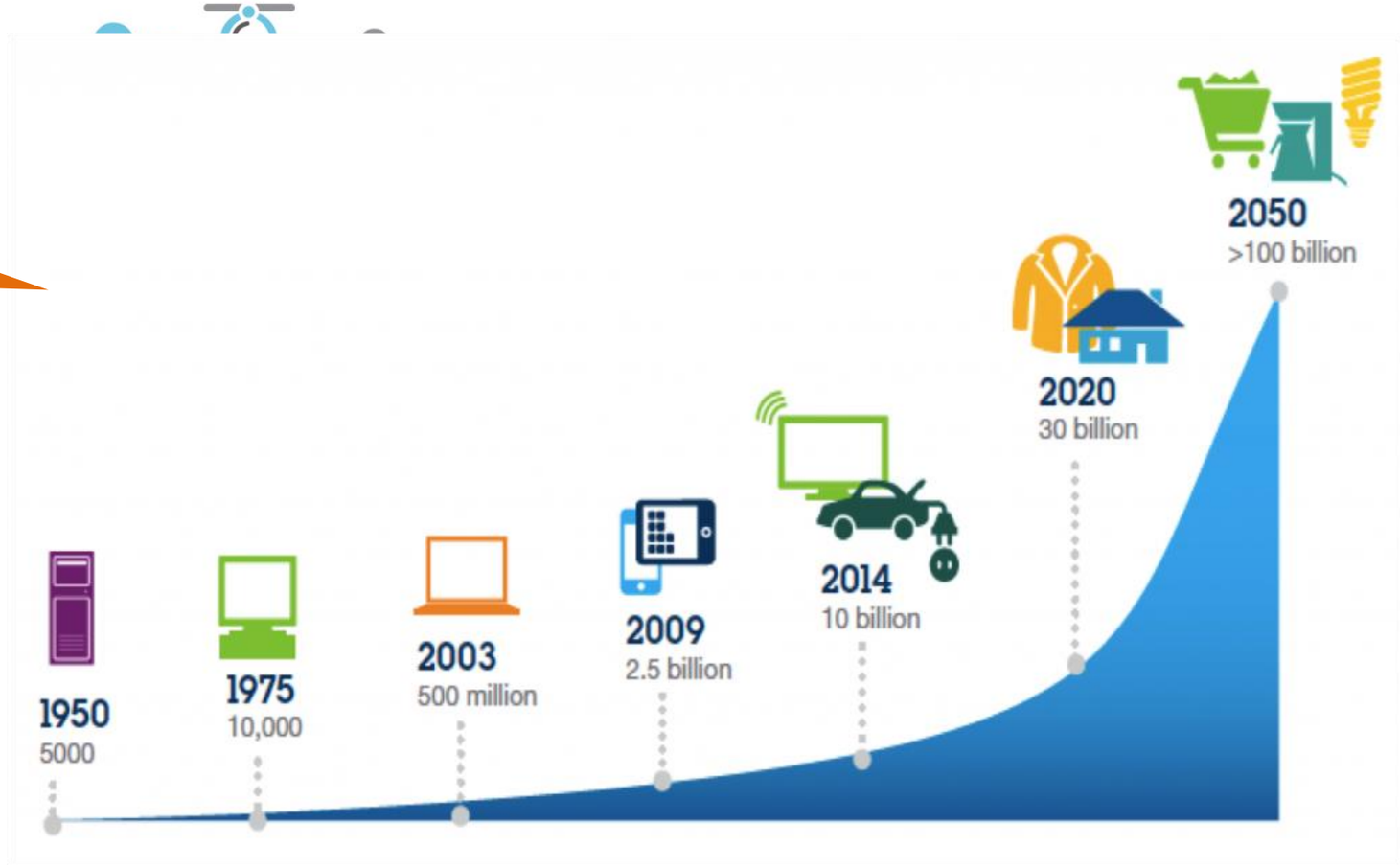
https://www.cerc.utexas.edu/utda/

Outline

- Introduction and MAGICAL Overview
- MAGICAL Components
  - ✔ Parametric Device Generation
  - ✔ Analog Layout Constraint Extraction
  - ✔ Analog Placement
  - ✔ Analog Routing
- MAGICAL Experimental Results
- Conclusion

Introduction

- High demand of analog/mixed-signal IC in emerging applications
  - ✔ Internet of Things (IOT), autonomous and electric vehicles, communication and 5G networks…



Introduction

- Analog IC layout design still heavily manual
  - ✔ Cf. digital IC layout automation
  - ✔ Very tedious and error-prone

- Modern SoCs: 20% or less analog, but maybe over 80% design time
  - ✔ Follow complex design rules in advanced tech nodes
  - ✔ Mitigate parasitic, noise, and layout-dependent effects
  - ✔ Handle manufacturability, reliability and yield issues

Prior Art of Analog Layout

- Placement: [Lampaert+, JSSC'95], [Strasser+, ICCAD'08], [Lin+, TCAD'09], [Ma+, TCAD'11], [Wu+, ICCAD'12], [Lin+, TCAD'16]
- Routing: [Ozdal+, ICCAD'12], [Ou+, DAC'13], [Ou+, TCAD'16]
- Constraint extraction: [Malavasi+, TCAD'99], [Massier+, TCAD'08]
- Flows/Tools:

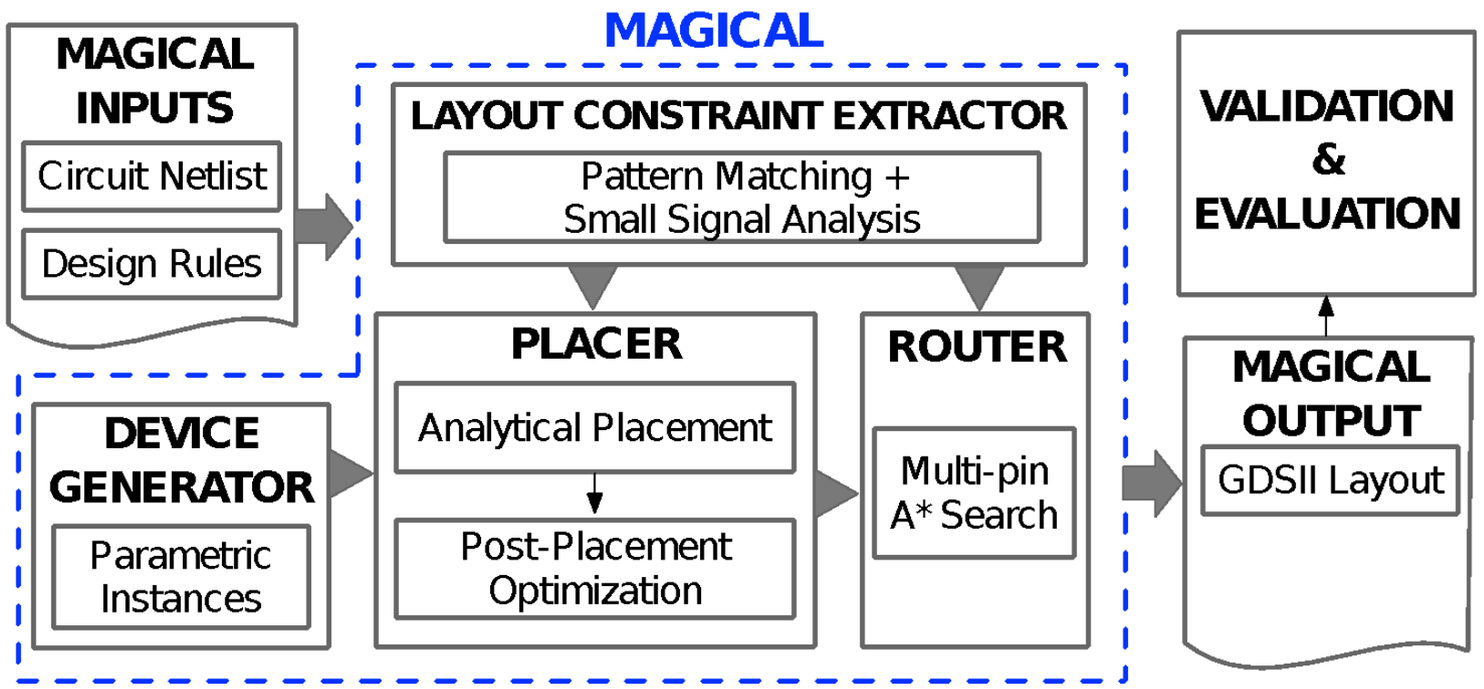
Approaches	Optimization-based	Template-based	Standard cell-based
Prior Art	ILAC, LAYLA, KOAN/ANAGRAM II, Malavasi+, GELSA, Strasser+	IPRAIL, ALG, Zhang+, Liu+, LAYGEN, LAYGEN II	Waters+, Weaver+, Deng+, Liu+
Pros	(+) Flexible (+) Optimize towards different objectives	(+) Runs fast (+) Suitable to layout retargeting	(+) Leverages digital APR tools (+) Runs fast
Cons	(-) Usually runs slow	(-) Lack of flexibility	(-) Requires new circuit design (-) Performance still not in line with state-of-the-art

Limitations of Prior Art and Our Goal

- Why previously works are not widely used in industry
  - ✔ Only capture limited design intents
  - ✔ Efficiency and scalability issues
  - ✔ Limited capability to transfer to different technology nodes
  - ✔ Not easily accessible for trial and use
- Our Goal to address the DARPA IDEA mission
  - ✔ End-to-end analog layout automation, leveraging recent AI and machine learning advancement
  - ✔ □ Machine-generated analog IC layout (MAGICAL) system

MAGICAL Layout System Framework

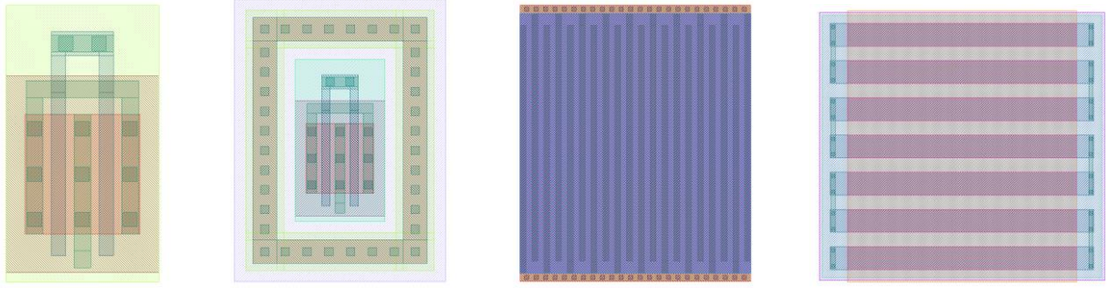
- Input: unannotated netlist
- Output: GDSII Layout
- Key Components:
  - ✔ Device Generation
  - ✔ Constraint Extraction
  - ✔ Analog Placement
  - ✔ Analog Routing



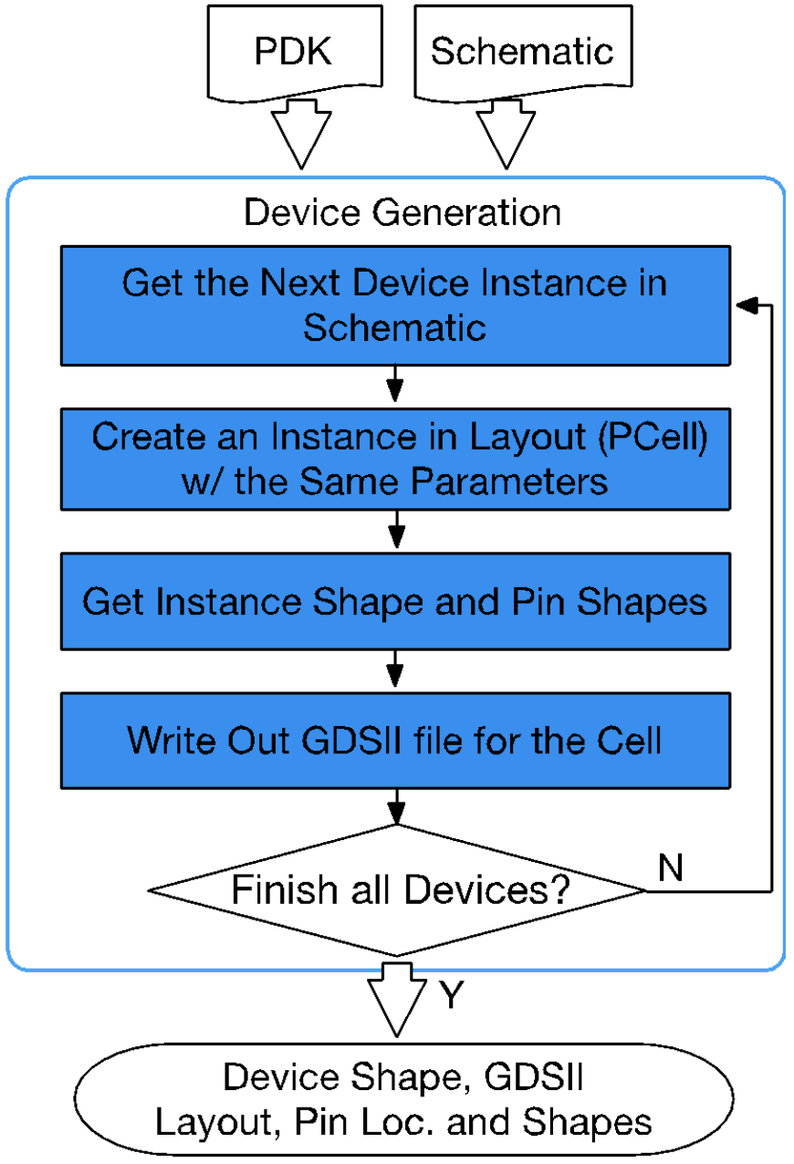
- **Fully-automated (no-human-in-the-loop)**
- Guided by analytical, heuristic, and machine learning algorithms
- Obtained promising initial results
- **Open-sourced** on GitHub: https://github.com/magical-eda/MAGICAL

Parametric Device Generation

- Inputs: circuit netlist, design rules
- Outputs: device GDSII layout, bounding box, pins, etc.
- Developed parametric device generation kernel in Python
- Correct by construction:
  - ✔ DRC and LVS cleaned



NMOS | PMOS w/ guard ring | MOM capacitor | Poly resistor

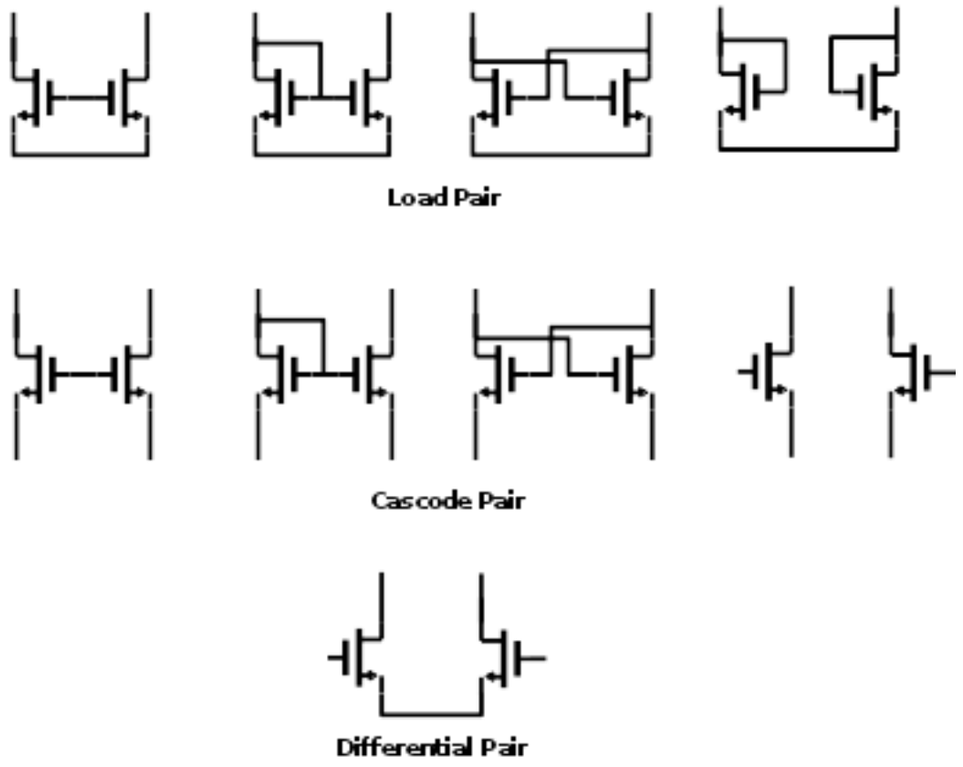


Analog Layout Constraint Extraction

- Input:
  - ✔ Unannotated circuit netlist (spice/spectre)
- Outputs:
  - ✔ Symmetric constraints for placement and routing, including symmetric device groups and symmetric nets
- Methods:
  - ✔ Graph abstraction
  - ✔ Seed pattern detection based on pattern matching
  - ✔ Graph traversal based on small signal flow analysis
  - ✔ Constraints post-processing

Analog Layout Constraint Extraction

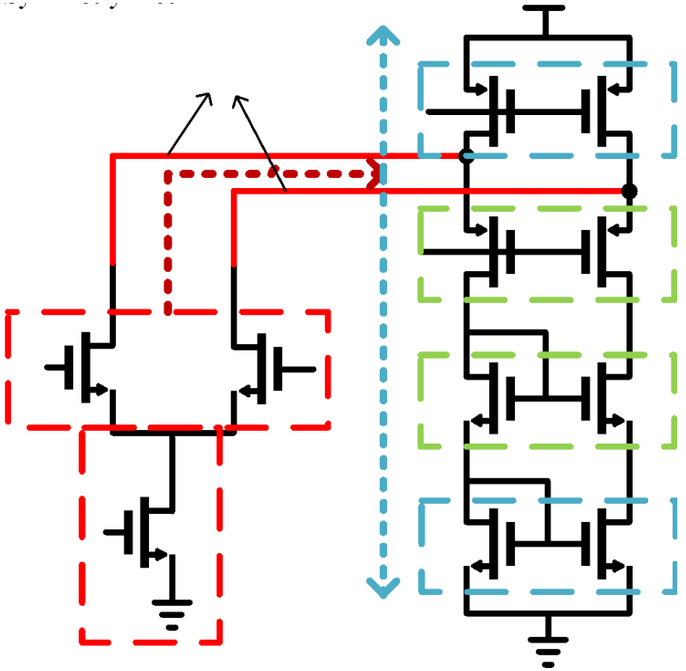
- Graph abstraction of the circuit netlist
  - ✔ Abstract circuit netlist into graph representation
  - ✔ Represent devices, pins, and nets as nodes
  - ✔ Constraint extraction is performed based on the graph representation
- Seed pattern detection
  - ✔ Transistor pairs from pattern library as seed patterns (start/end points for graph traversal)
  - ✔ Pair-wise pattern matching instead of expensive graph isomorphism algorithms
  - ✔ Ambiguity resolving for mixed-signal circuits



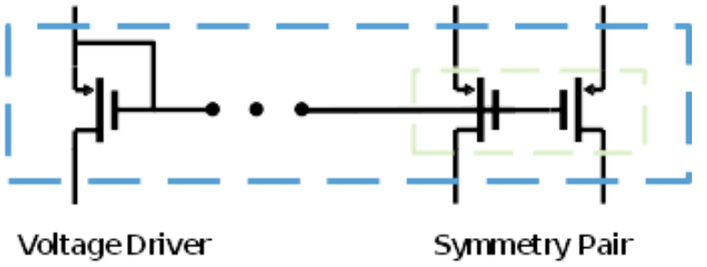
Sample Pattern Library

Analog Layout Constraint Extraction

- Signal flow based graph traversal
  - ✔ Analogous to following the differential current in small-signal analysis
  - ✔ Start from seed patterns, and ends when the flow paths meet at seed patterns
  - ✔ Visited subgraph from the same seed pattern form a symmetric group and symmetric nets
  - ✔ Also consider matching between passive devices
- Constraints post-processing
  - ✔ Identify additional self-symmetric nets and bias circuit symmetry
  - ✔ Constraints pruning to allow a device to be in at most one symmetry constraint for placement feasibility



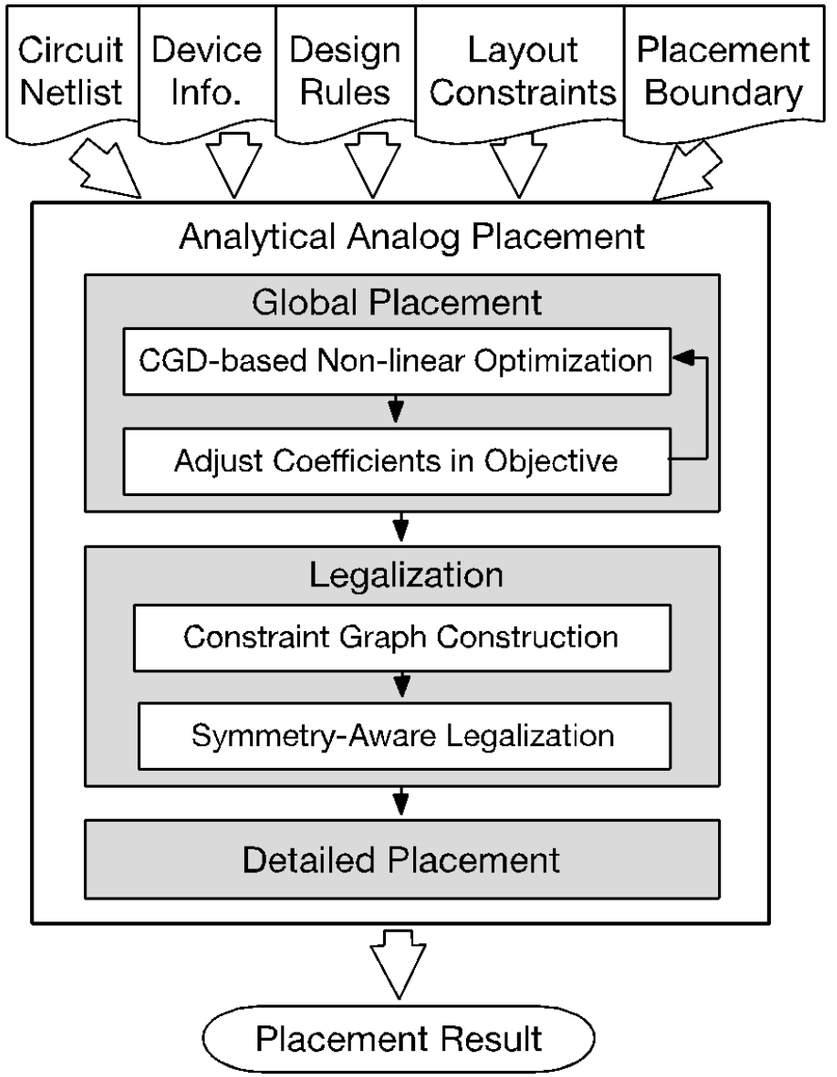
Small Signal Analysis



Bias Circuit Symmetry Detection

Analog Placement

- Input:
  - ✔ Circuit netlist
  - ✔ Device information
  - ✔ Design rules
  - ✔ Layout constraints…
- Output: a legal placement solution
- Constraints:
  - ✔ Symmetric device groups share a common symmetric axis in the placement
- Objectives:
  - ✔ Area, wirelength, …



Analytical Global Placement

- We relax the constraints into penalties in the objective, and transform the problem into an unconstrained nonlinear optimization problem
- Objective:

$$Objective = f_{WL} + a \cdot f_{OL} + b \cdot f_{BND} + c \cdot \left(f_{SYM}^{x} + f_{SYM}^{y}\right)$$

- Wirelength term (half-perimeter wirelength):

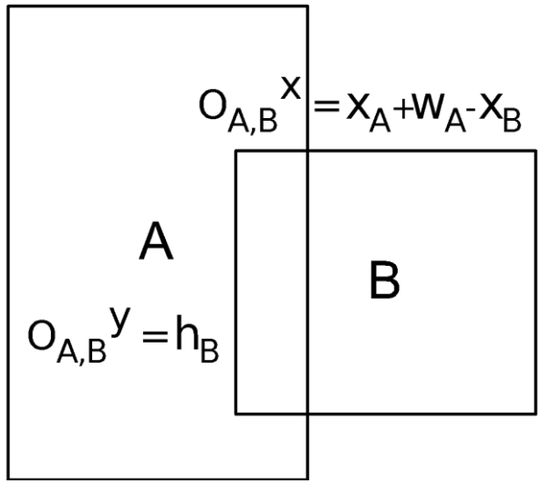
$$f_{WL} = \Sigma_{n_k}\left(\max_{i \in n_k} x_i - \min_{i \in n_k} x_i + \max_{i \in n_k} y_i - \min_{i \in n_k} y_i\right)$$

- Overlap penalty:

$$f_{OL} = \sum_{(i,j) \in L} O_{i,j}^{x} \cdot O_{i,j}^{y}$$

$$O_{i,j}^{x} = \max\left(\min\left(x_i + w_i - x_j, x_j + w_j - x_i, w_i, w_j\right), 0\right)$$

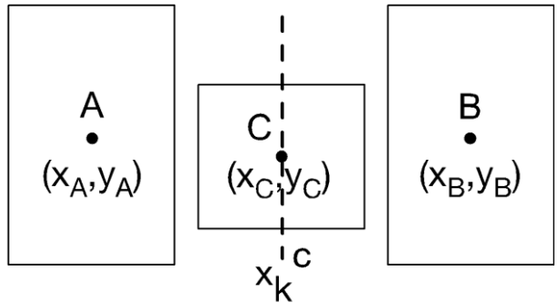
$$O_{i,j}^{y} = \max\left(\min\left(y_i + h_i - y_j, y_j + h_j - y_i, h_i, h_j\right), 0\right)$$



Analytical Global Placement

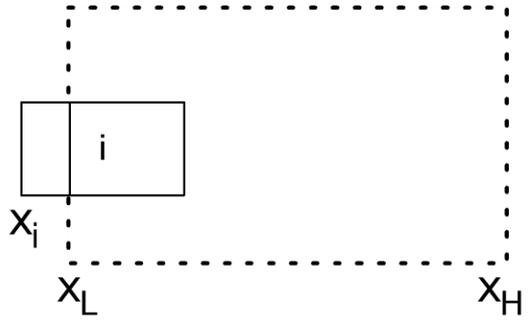
- Asymmetry penalty:

$$f_{SYM}^{x} = \sum_{g_k \in G} \left(\sum_{(i,j) \in g_k^p} \left((x_i + x_j - 2 \cdot x_k^c)^2 + (y_i - y_j)^2 \right) + \sum_{i \in g_k^s} (x_i - x_k^c)^2 \right)$$



- Out of boundary penalty:

$$f_{BND} = \sum_{i \in D} \left(\max(x_L - x_i, 0) + \max(x_i + w_i - x_H, 0) + \max(y_L - y_i, 0) + \max(y_i + h_i - y_H, 0) \right)$$



Legalization by Linear Programming

- Linear programming (LP) based legalization according to the constraint graphs to minimize area and satisfy constraints
- Decomposed into x- and y- direction sub-problems and solved independently

$$\begin{aligned}
&\text{Minimize} && W \\
&\text{Subject to} && 0 \le x_i \le W - w_i, \forall i \in D, \\
& && x_i + w_i \le x_j, \forall e_{i,j} \in G_h, \\
& && x_i + x_j + w_j = 2 \cdot x_k^c, \forall (i,j) \in g_k^p, \\
& && 2 \cdot x_i + w_i = 2 \cdot x_k^c, \forall i \in g_k^s,
\end{aligned} \quad \forall g_k \in G,$$

Boundary constraints

Topology order constraints

Symmetry constraints

$$\begin{aligned}
&\text{Minimize} && H \\
&\text{Subject to} && 0 \le y_i \le H - h_i, \forall i \in D, \\
& && y_i + h_i \le y_j, \forall e_{i,j} \in G_v, \\
& && y_i = y_j, \forall (i,j) \in g_k^p, \forall g_k \in G,
\end{aligned}$$

Boundary constraints

Topology order constraints

Symmetry constraints

Detailed Placement

- LP-based wirelength refinement and design rules handling

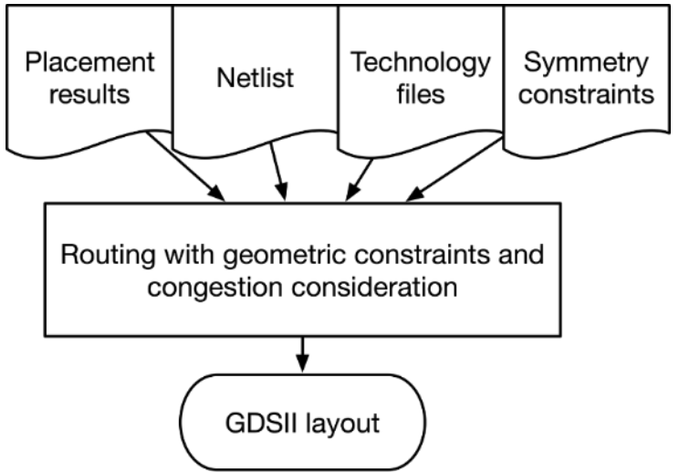
$$
\begin{aligned}
&\text{Minimize} && Wirelength \\
&\text{Subject to} && 0 \le x_i \le W^* - w_i, \forall i \in D, \\
& && x_i + w_i \le x_j, \forall e_{i,j} \in G_h, \\
& && 0 \le y_i \le H^* - h_i, \forall i \in D, \\
& && y_i + h_i \le y_j, \forall e_{i,j} \in G_v, \\
& && x_i + x_j + w_j = 2 \cdot x_k^c, \forall (i,j) \in g_k^p, \\
& && 2 \cdot x_i + w_i = 2 \cdot x_k^c, \forall i \in g_k^s, \\
& && y_i = y_j, \forall (i,j) \in g_k^p,
\end{aligned}
\qquad \forall g_k \in G,
$$

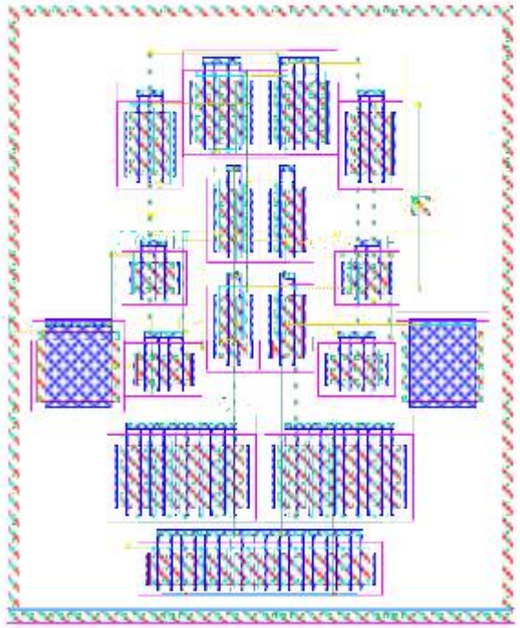
Fixed boundary and topology order constraints

Symmetry constraints

Analog Routing

- Inputs:
  - ✔ Routing constraints
  - ✔ Placement results
  - ✔ Design rules (from PDK)
- Output:
  - ✔ LVS cleaned GDSII layout
- Methods:
  - ✔ A* search based global and detailed routing with geometric constraints





Global Routing

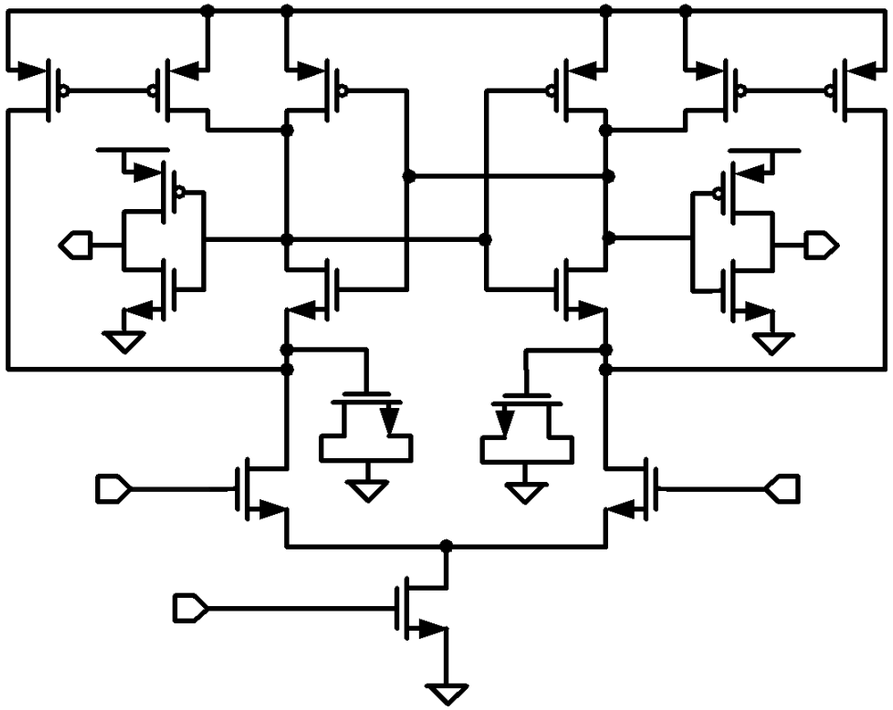
- Symmetry-aware grid-based A* search routing engine
- First divide the placement into uniform grids
- Calculate routing capacity on 2D grid edges
- Analog layout pins could be polygons varied in size and shape, which are decomposed into searching points in the path search scheme
- Split multi-pin net into two-pin nets based on minimum spanning tree with HPWL as the edge costs, considering symmetric net pairs
- Symmetric net pairs and self-symmetric nets are routed on the 3D global routing grid with exact symmetry
- A rip-up and reroute scheme is applied when failing to achieve a feasible solution in the early iterations

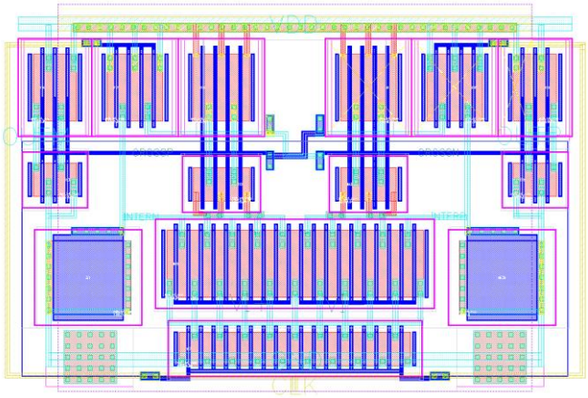
Detailed Routing

- Similar to the global routing, an A* search based scheme is applied
- Only need to align to the manufacturing grid specified in the process technology design rules
- Design rules are checked during the A* search
- When routing the symmetric net pairs or self-symmetric nets, both sides along the symmetry axis are routed at the same time
- The resulting routing solution is symmetric-feasible by construction
- Other analog circuit-specific routing considerations will also be incorporated to improve the post-layout circuit performance

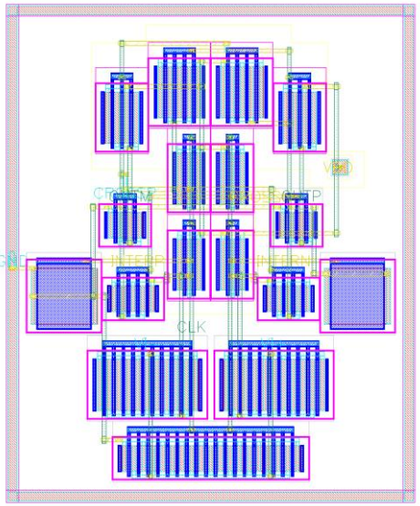
MAGICAL Preliminary Results

- A comparator design





Manual Layout



MAGICAL Layout

MAGICAL Preliminary Results

- A comparator design

Post-Layout Performance Simulations

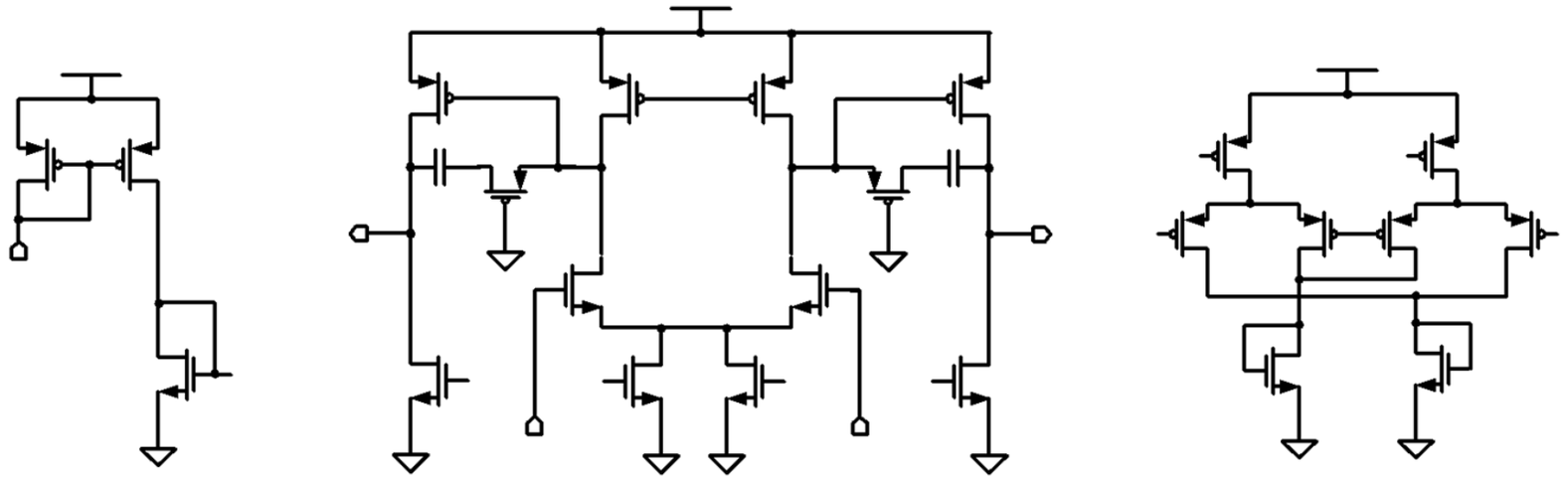
	Manual	MAGICAL
Power (uW)	16.8	18.7
Output Delay (ps)	150	152
Input-referred Noise (uVrms)	380	334
Input-referred Offset (mV)	0.15	0.50

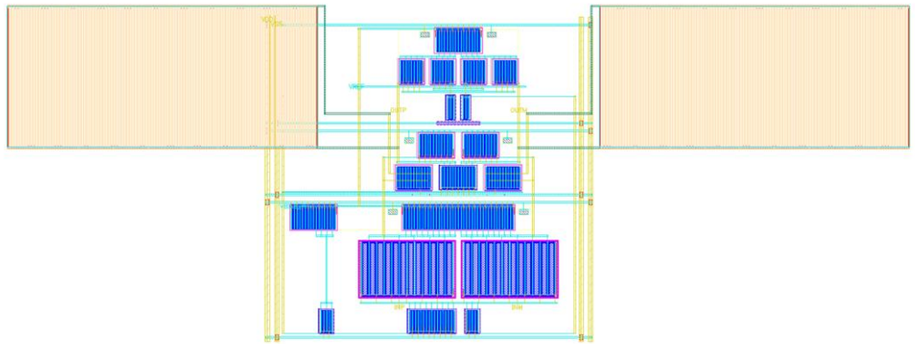
Symmetry-related

Parasitic or compactness related

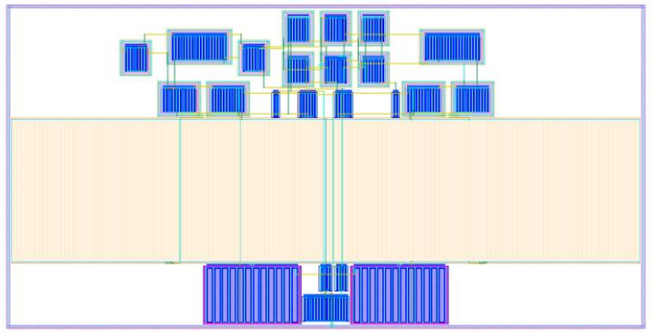
MAGICAL Preliminary Results

- A 2-stage miller-compensated OTA design





Manual Layout



MAGICAL Layout

MAGICAL Preliminary Results

- A 2-stage miller-compensated OTA design

Post-Layout Performance Simulation

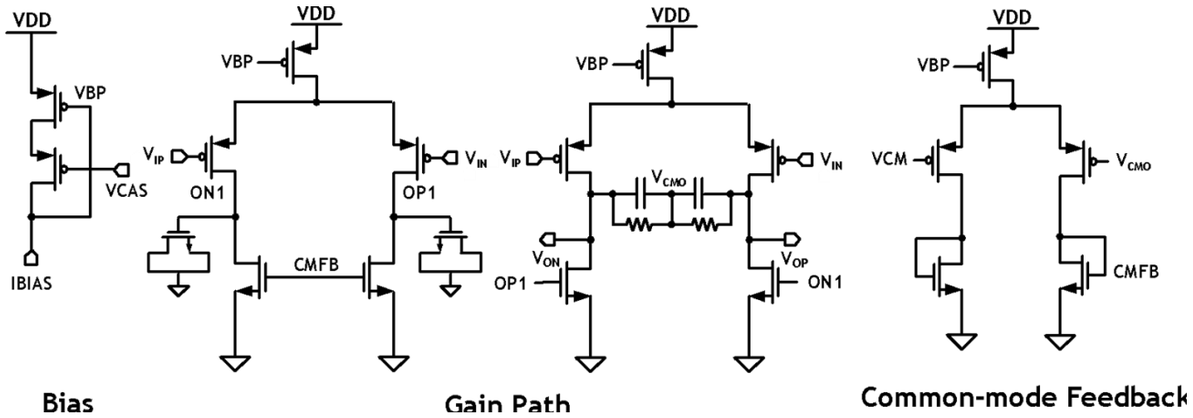
	Manual	MAGICAL
DC Gain (dB)	37.7	38.0
Unity-gain Bandwidth (MHz)	110	107.5
Phase Margin (degree)	67.8	62.3
Input-referred Noise (uVrms)	219	221.5
CMRR (dB)	103	92.5
Input-referred Offset (mV)	0.2	0.48

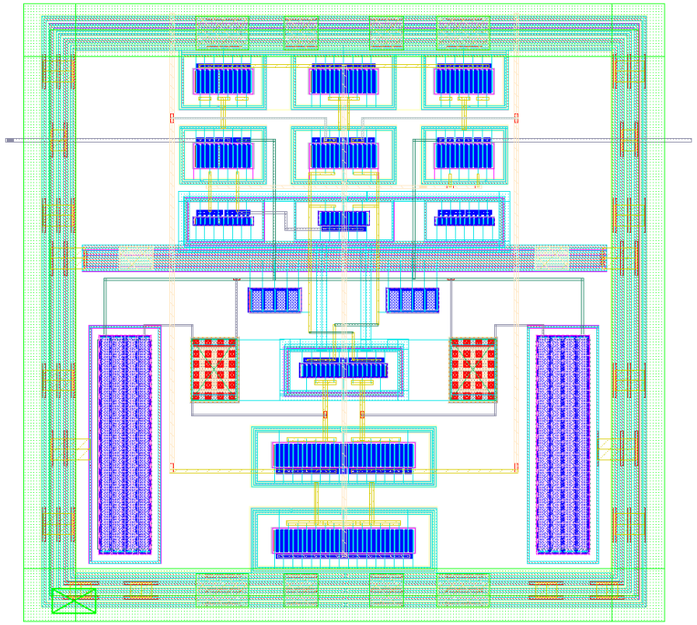
Parasitic or compactness related

Symmetry-related

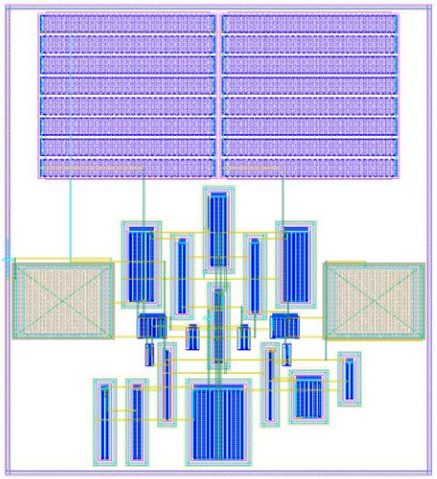
MAGICAL Preliminary Results

- A 2-stage feedforward-compensated OTA design





Manual Layout



MAGICAL Layout

MAGICAL Preliminary Results

- A 2-stage feedforward-compensated OTA design

Post-Layout Performance Simulation

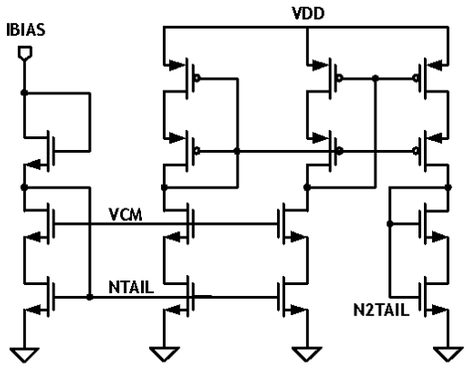
	Manual	MAGICAL
DC Gain (dB)	35.3	35.3
Unity-gain Bandwidth (MHz)	2200	2200
Phase Margin (degree)	77.6	77.9
CMRR (dB)	126.1	88.9
Input-referred Offset (mV)	<0.01	0.161

Parasitic or compactness related

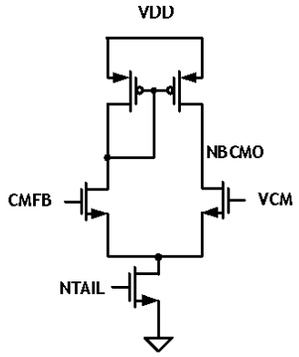
Symmetry-related

MAGICAL Preliminary Results

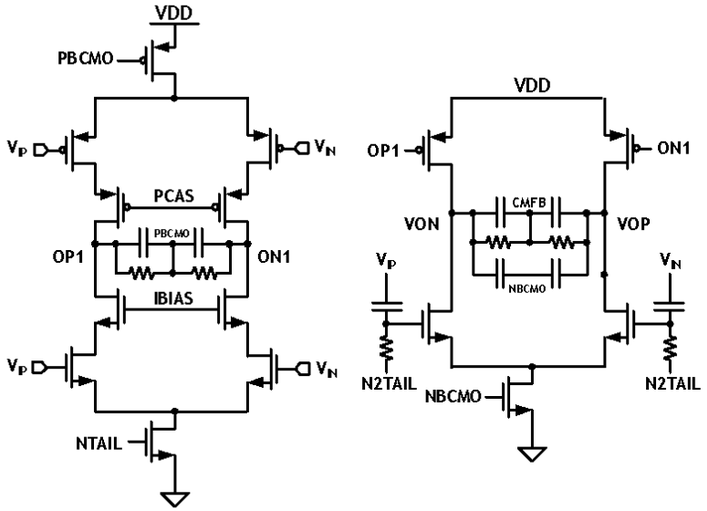
- An inverter-based OTA design



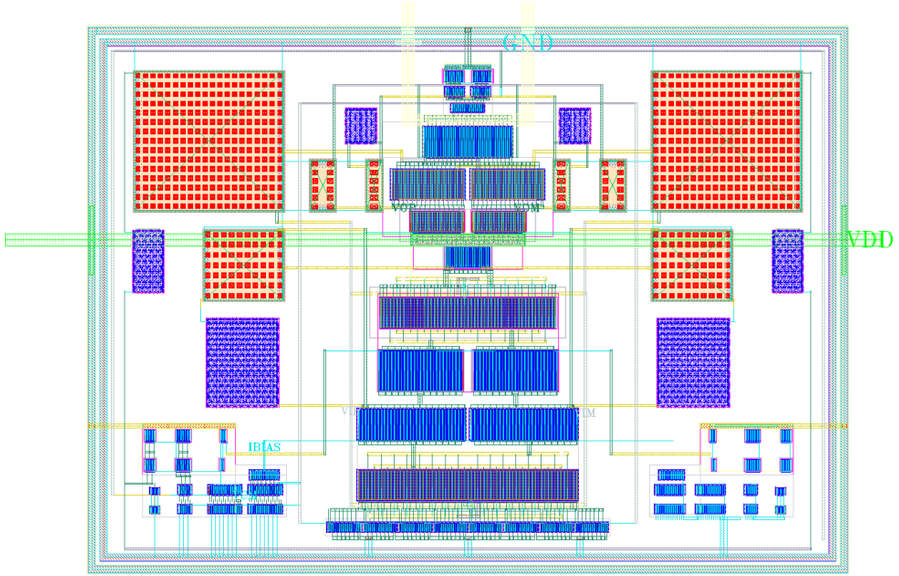
Bias



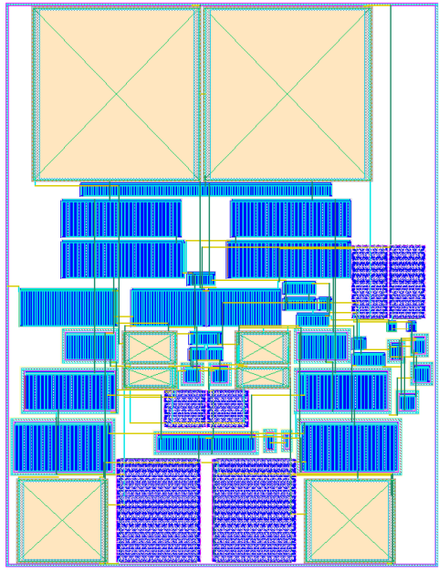
Common-mode Feedback



Gain Path



Manual Layout



MAGICAL Layout

MAGICAL Preliminary Results

- An inverter-based OTA design

Post-Layout Performance Simulation

	Manual	MAGICAL
DC Gain (dB)	69	69
Unity-gain Bandwidth (MHz)	1300	1130
Phase Margin (degree)	58	56.5
CMRR (dB)	94.5	110
Input-referred Offset (mV)	0.016	0.001

Parasitic or compactness related

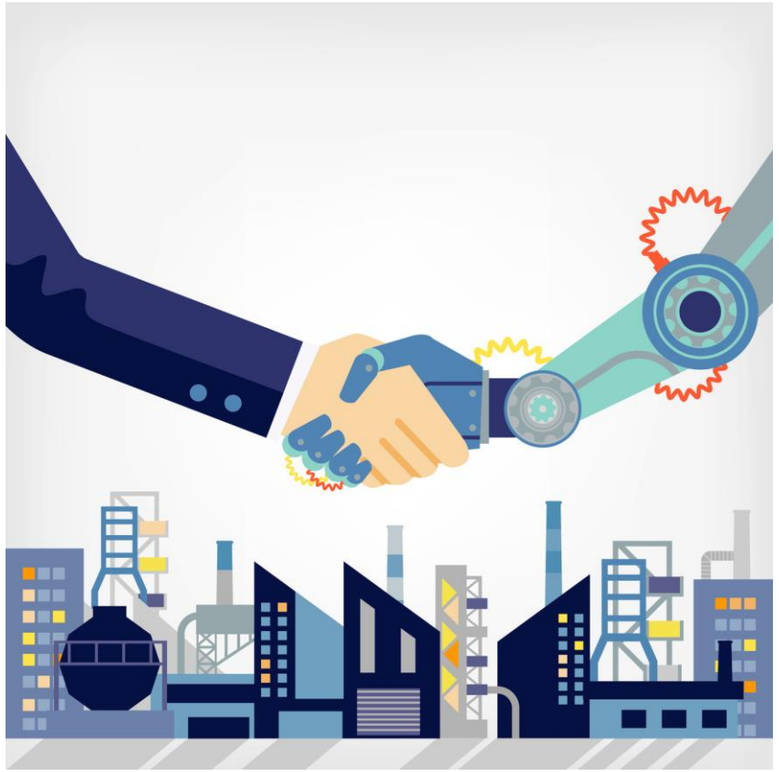
Symmetry-related

Conclusion

- Preview of MAGICAL - a fully automated (no-human-in-the-loop), machine-generated analog IC layout system
  - ✔ Key components: parametric device generation, automatic constraint generation, placement, and routing
  - ✔ Guided by various analytical, heuristic, and machine learning algorithms
  - ✔ Obtained promising initial results cf. manual design (tapeout)
  - ✔ Results are still preliminary
- **MAGICAL has been made open source** in mid-2019 (under BSD-3 license)
- GitHub: https://github.com/magical-eda/MAGICAL

Thanks!

Q & A?



Comparisons with BAG

MAGICAL

- Optimization based generator
- No template needed
- No human in the loop
  - ✔ Embedded constraint extraction
  - ✔ Fully automated P&R

BAG

- Procedural layout generator
- Require template designs
- Require additional human input
  - ✔ Relative placement location
  - ✔ Routing metal track selection