EDPS, 10/04/2019

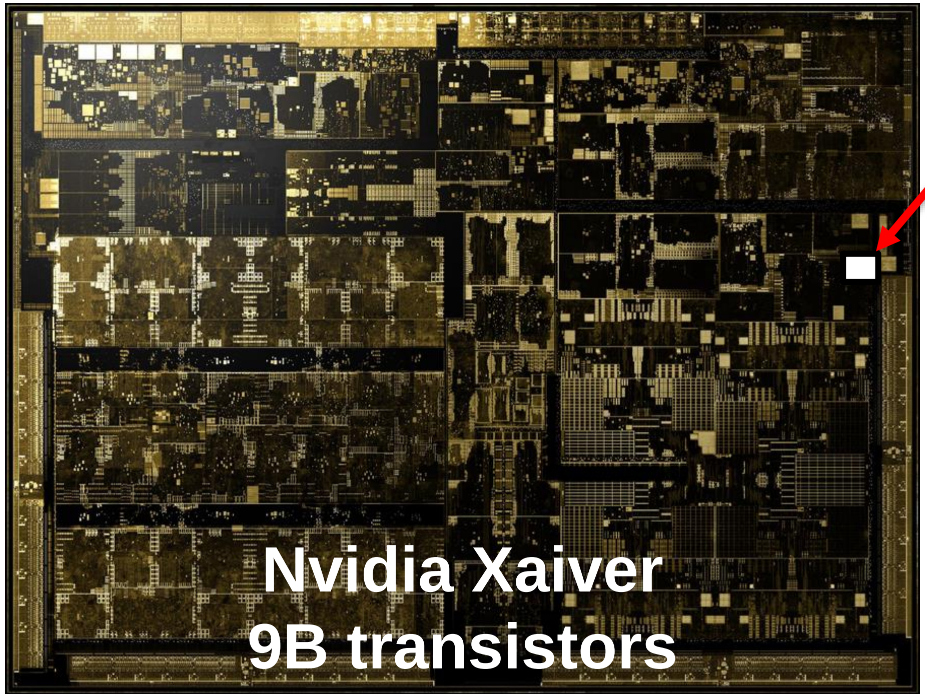
AI-Enabled Agile IC Design and Manufacturing

David Z. Pan

The University of Texas at Austin

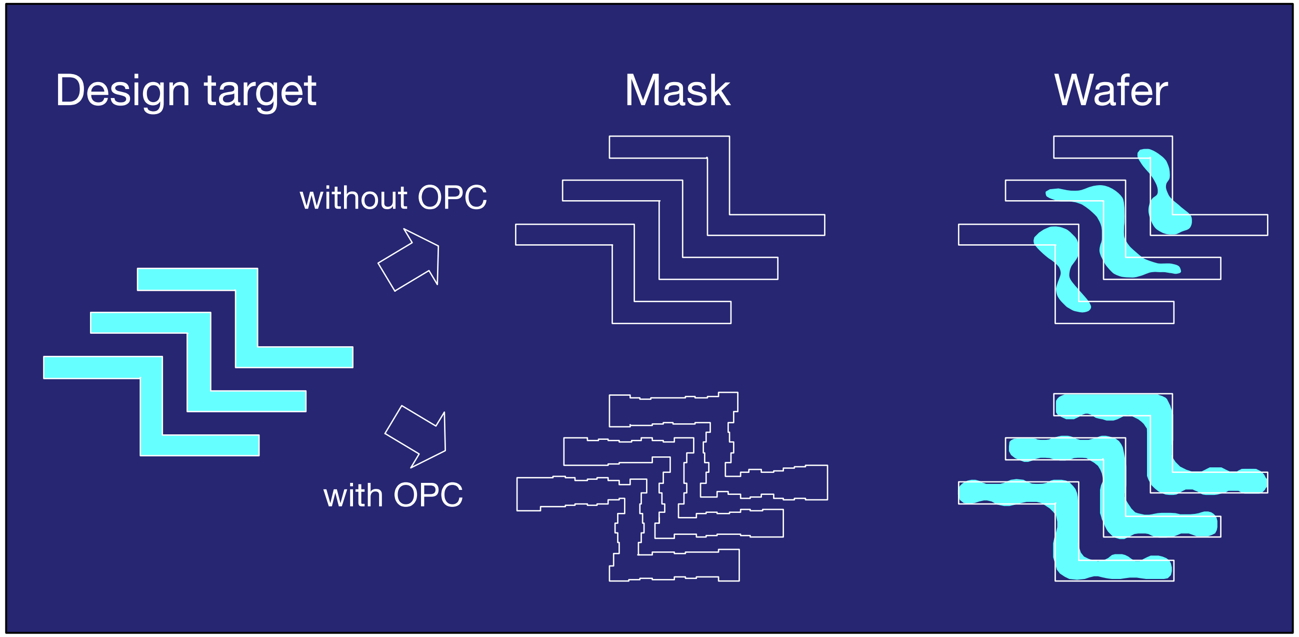
http://www.ece.utexas.edu/~dpan

IC Design/Manufacturing Complexity



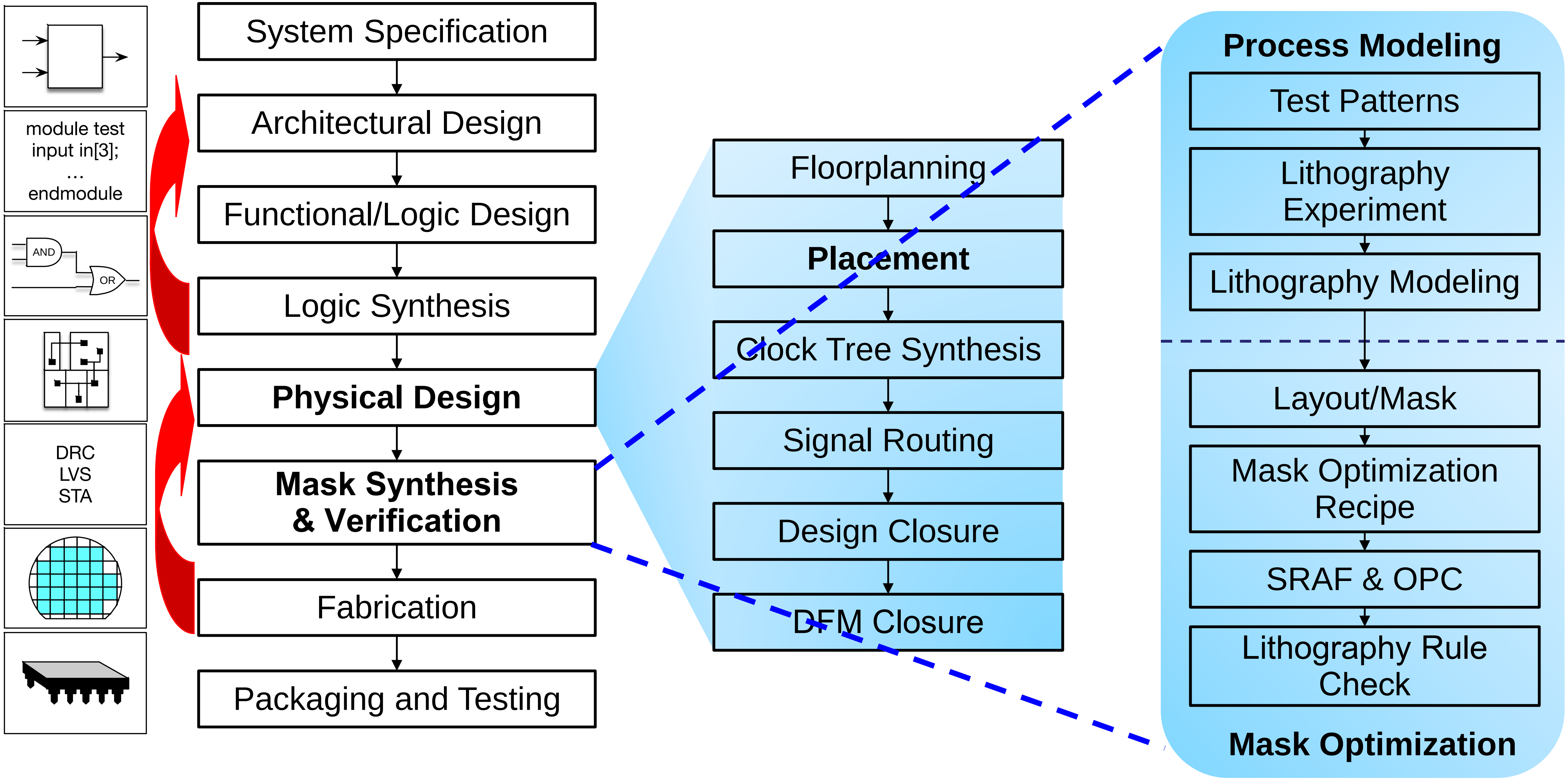
Divide a chip into small partitions e.g., **1~2M** cells per partition

Turn-around time for 1 iteration of backend flow: 3~6 days for one partition



What you see (at design) is not (necessarily) what you get (at fab)

IC Design/Manufacturing Flow



- System Specification
- Architectural Design
- Functional/Logic Design
- Logic Synthesis
- **Physical Design**
- **Mask Synthesis & Verification**
- Fabrication
- Packaging and Testing

- Floorplanning
- **Placement**
- Clock Tree Synthesis
- Signal Routing
- Design Closure
- DFM Closure

Process Modeling

- Test Patterns
- Lithography Experiment
- Lithography Modeling

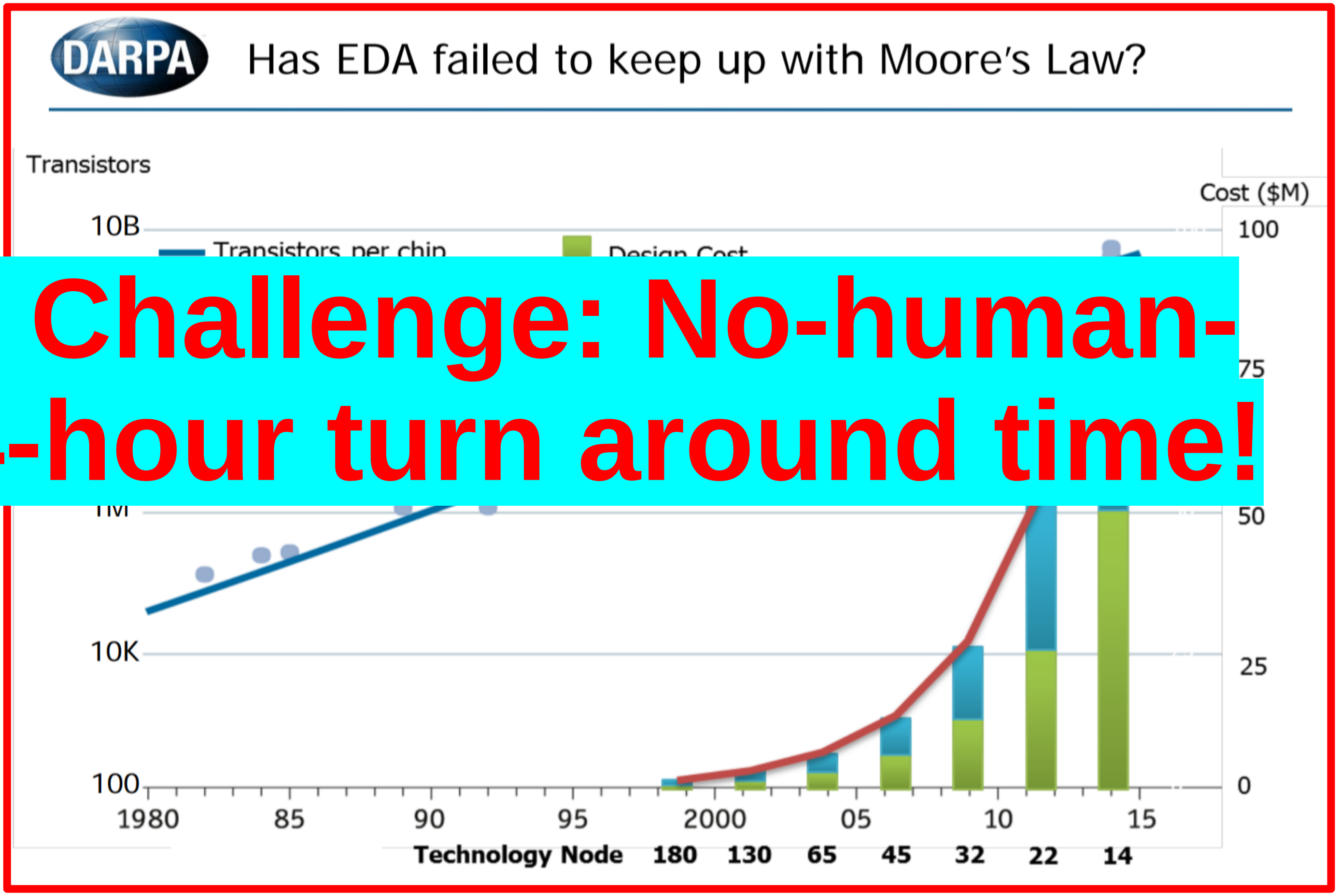
- Layout/Mask
- Mask Optimization Recipe
- SRAF & OPC
- Lithography Rule Check

Mask Optimization

Nanometer Design/Manufacturing Challenges

- Performance/Power/Area PPA
- Manufacturability/Yield
- Reliability
- Secu
- Turn-
- ……
- 2018 DARPA ERI ($1.5B) IDEA/POSH “Hardware Compiler” ($100M)

A. Olofsson, DARPA, ISPD-2018 Keynote

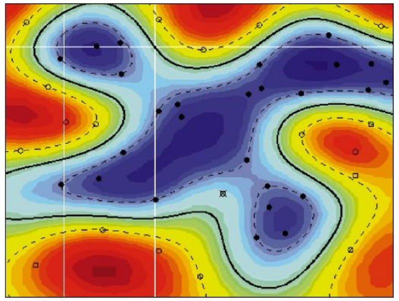


DARPA Has EDA failed to keep up with Moore’s Law?

DARPA Grand Challenge: No-human-in-the-loop, 24-hour turn around time!

How AI (ML/DL) Can Help?

- Lots of work for various stages of physical design and DFM
- For example on lithography hotspot detection
  - Our work [Ding+, ICICDT 2009 BPA] among the first to use ML (SVM) for litho-hotspot detection
  - Very active research in last 10 years, ICCAD 2012 CAD Contest
  - Meta-classification combining ML and PM [Ding+, ASPDAC'12 BPA]
  - Deep neural network [Yang+, DAC'17]
  - Big data vs. small data; transfer/active/semi-supervised learning [Lin+, ISPD'18], [Chen+, ASPDAC'19], Litho-GPA [Ye+, DATE 2019]…



- ML and PD Tutorial in July 2019 (ACM/IEEE Seasonal School)
  - http://yibolin.com/publications/tutorials/PDSeasonableSchool_ML4PD.pdf
- My talk today will cover some recent ideas/results for discussion

DREAMPlace: Deep Learning Toolkit-Enabled GPU Acceleration for Modern VLSI Placement [Lin+, DAC'19, Best Paper Award]

Source code release: https://github.com/limbo018/DREAMPlace

Typical Nonlinear Placement Algorithm

$$\min_{\mathbf{x},\mathbf{y}} \quad \sum_{e\in E} \mathrm{WL}(e;\mathbf{x},\mathbf{y}),$$

$$s.t. \quad D(\mathbf{x},\mathbf{y}) \leq t_d$$

⬇

Objective of nonlinear placement

$$\min \quad \underbrace{(\textstyle\sum_{e\in E} \mathrm{WL}(e;\mathbf{x},\mathbf{y}))}_{\text{Wirelength}} + \lambda \underbrace{D(\mathbf{x},\mathbf{y})}_{\text{Density}}$$

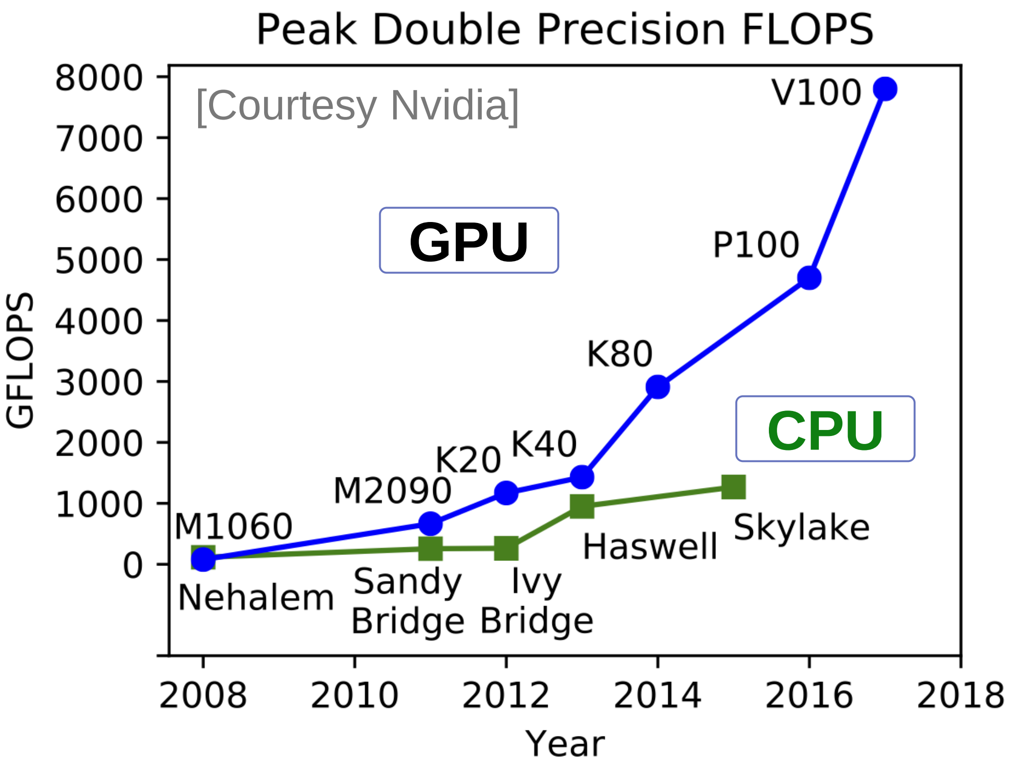
- Many papers on how to model WL, density, parameter tuning, etc.
- Huge development effort on a high-quality placement engine (e.g., > 1 year for RePlAce)
- CPU>3h to get good quality placement of 10M-cell design
- Clustering/acceleration limited → quality degradation

What is your Dream Placement Engine?

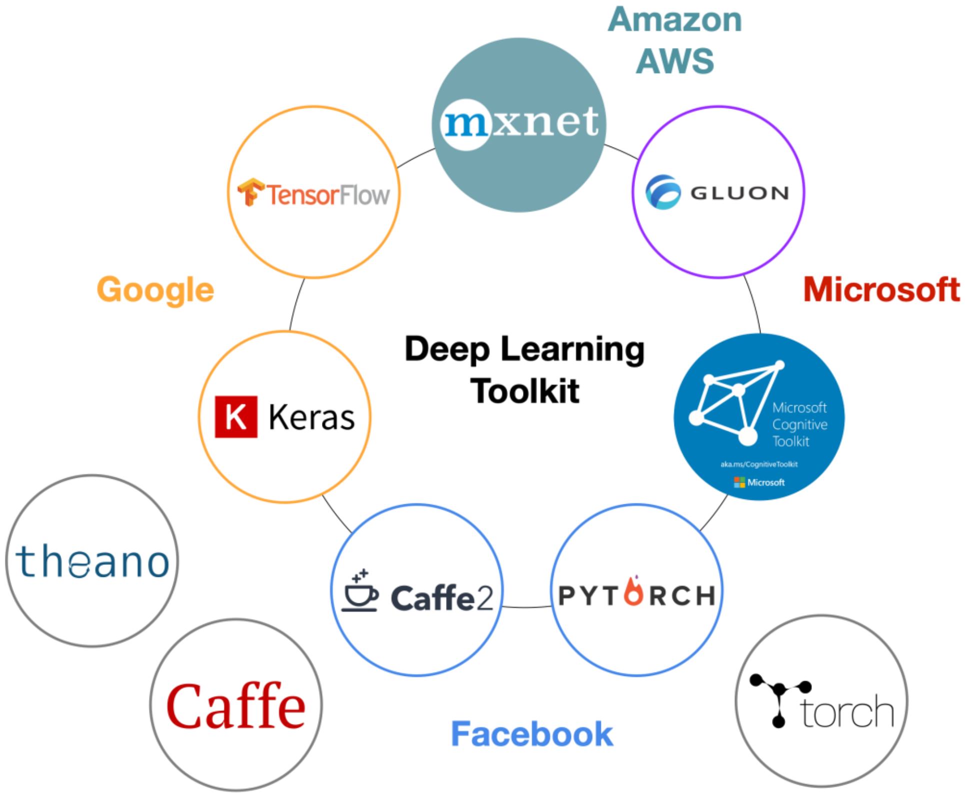
- ✓ Best quality: wirelength ➔ congestion, timing, power, …
- ✓ Ultrafast: placement is at the center of entire design flow ➔ faster design turn-around-time
- ✓ Low development overhead: ➔ from 1 year to a month or two?
- ✓ Extensible: easy to try new algorithms and acceleration techniques

10M-cell design finishes within 5min

Advances in Deep Learning Hardware/Software



Over **60x** speedup in neural network training since 2013



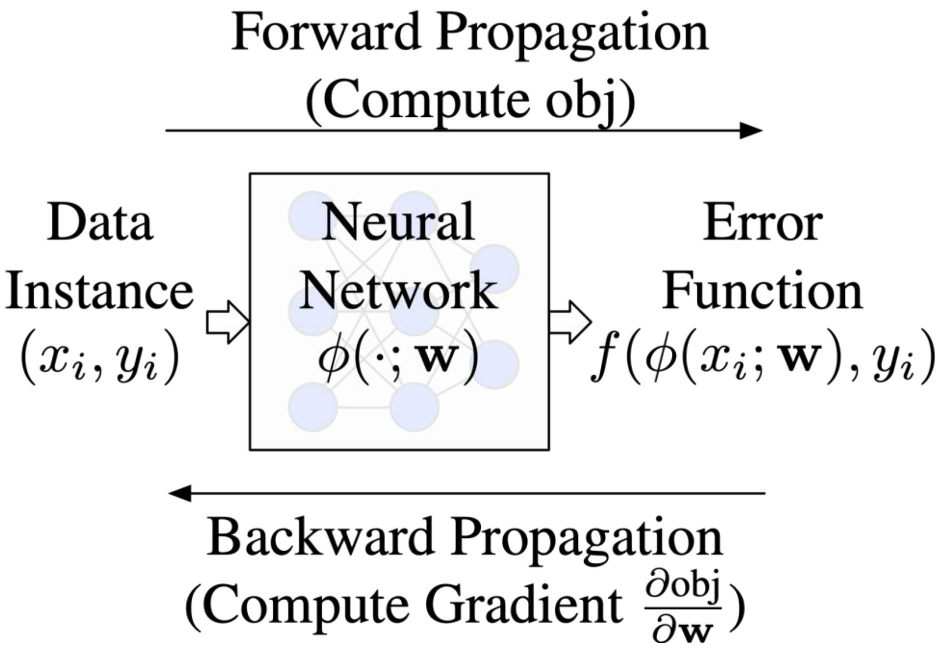
Deep learning toolkits

DREAMPlace Strategies

- We propose a novel **analogy** by casting the nonlinear placement optimization into a neural network training problem
- Greatly leverage deep learning hardware (GPU) and software toolkit (e.g., PyTorch)
- Enable ultra-high parallelism and acceleration while getting state-of-the-art results

Analogy Between NN Training and Placement

$$\min_{\mathbf{w}} \sum_{i}^{n} f(\phi(x_i; \mathbf{w}), y_i) + \lambda R(\mathbf{w})$$



Forward Propagation
(Compute obj)

Data
Instance
(x_i, y_i)

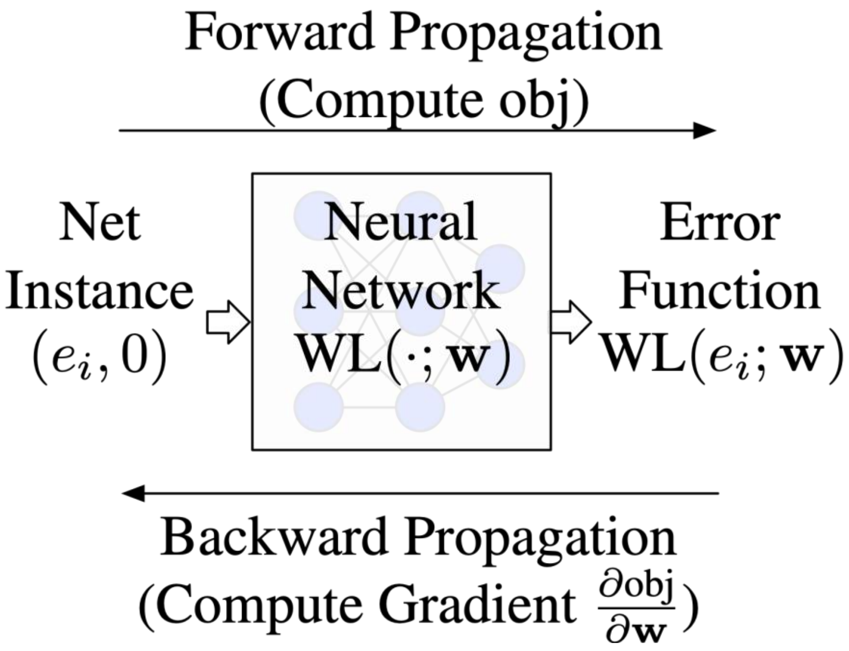
Neural
Network
$\phi(\cdot; \mathbf{w})$

Error
Function
$f(\phi(x_i; \mathbf{w}), y_i)$

Backward Propagation
(Compute Gradient $\frac{\partial \text{obj}}{\partial \mathbf{w}}$)

Train a neural network

$$\min_{\mathbf{w}} \sum_{i}^{n} \text{WL}(e_i; \mathbf{w}) + \lambda D(\mathbf{w})$$



Forward Propagation
(Compute obj)

Net
Instance
$(e_i, 0)$

Neural
Network
$\text{WL}(\cdot; \mathbf{w})$

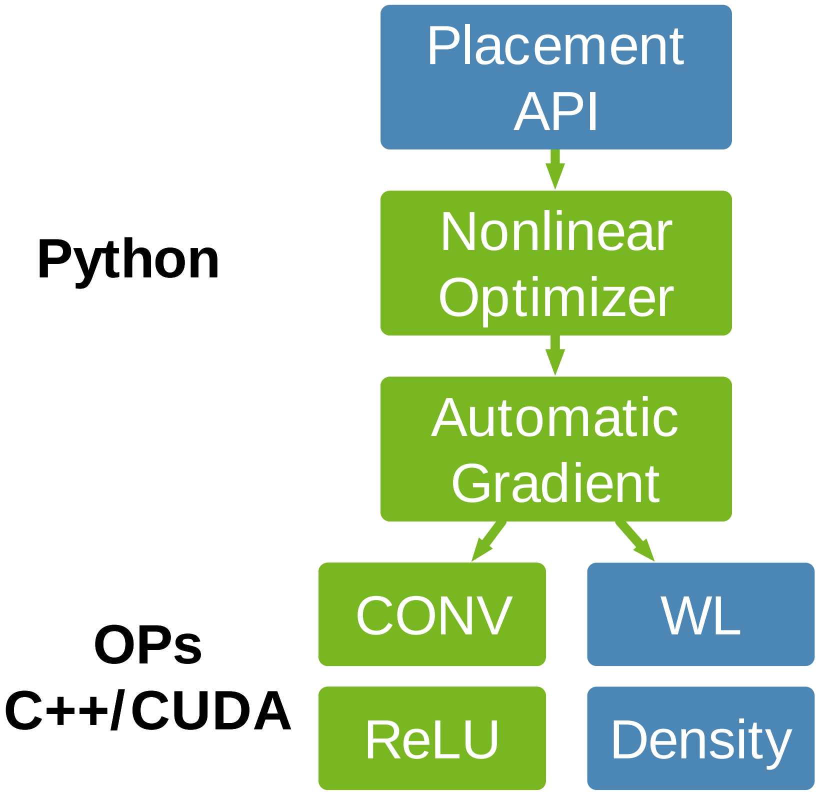
Error
Function
$\text{WL}(e_i; \mathbf{w})$

Backward Propagation
(Compute Gradient $\frac{\partial \text{obj}}{\partial \mathbf{w}}$)

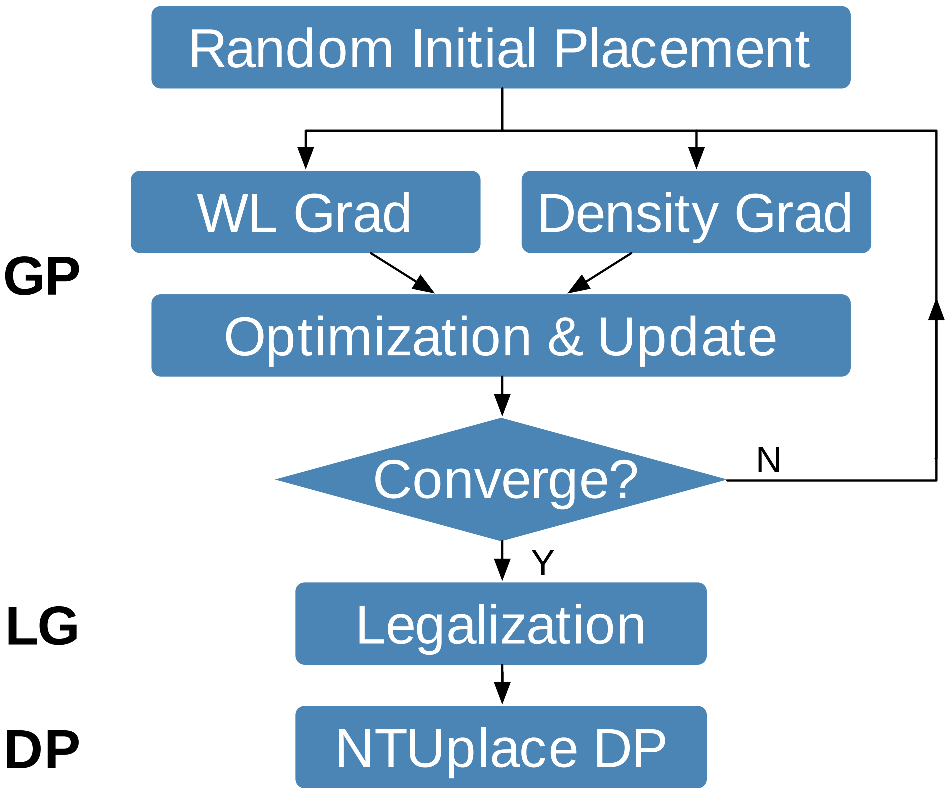
Solve a placement

DREAMPlace Architecture & Overall Flow

Leverage mature/highly optimized deep learning toolkit



DREAMPlace architecture



DREAMPlace flow

Global Placement Result Comparison

RePlAce [Cheng+, TCAD’18]

- CPU: 24-core 3GHz Intel Xeon
- 64GB memory allocated
- Current state-of-

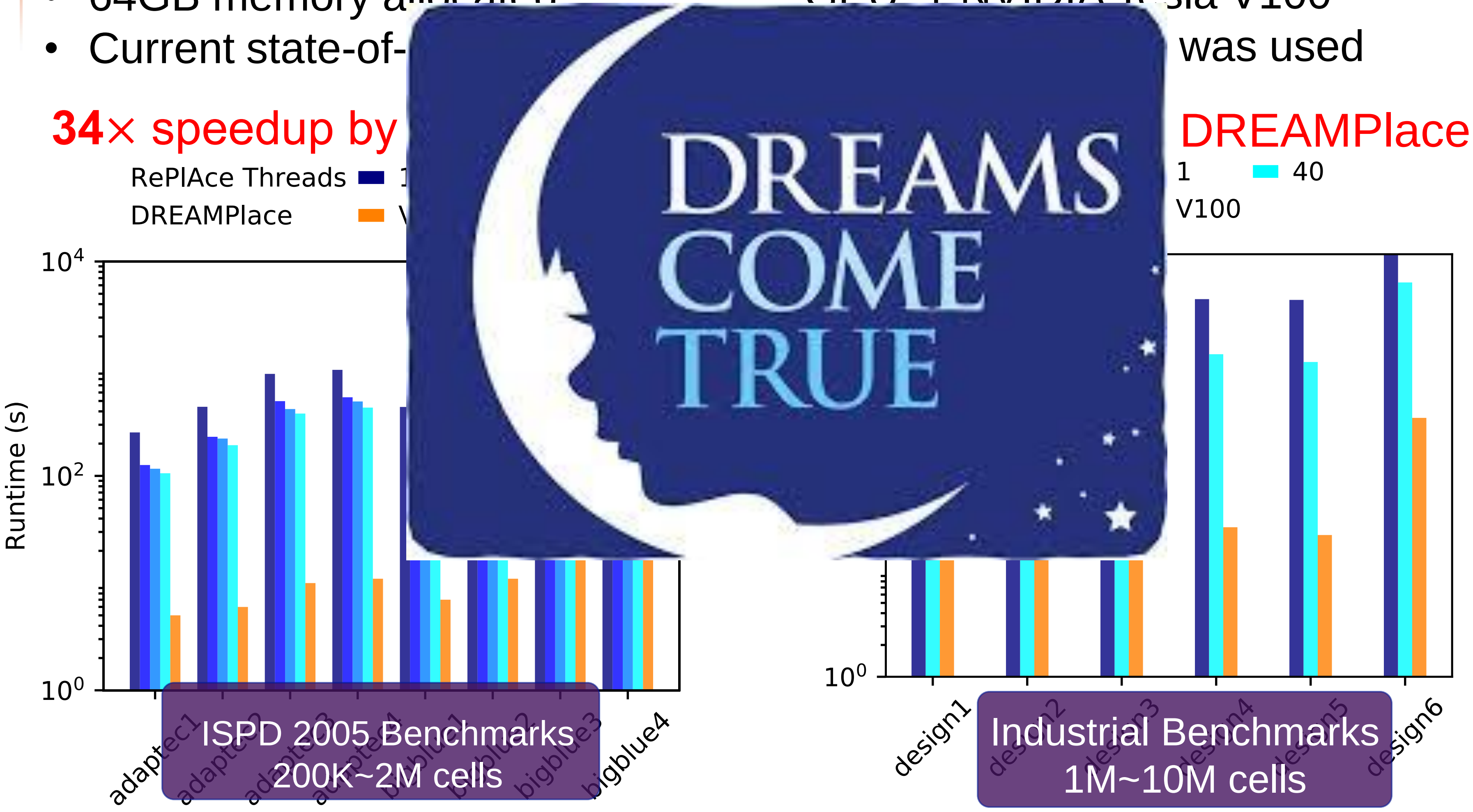
DREAMPlace [Lin+, DAC’19]

- CPU: Intel E5-2698 v4 @2.20GHz
- GPU: 1 NVIDIA Tesla V100
- was used

Same placement quality of results!

34× speedup by

DREAMPlace



ISPD 2005 Benchmarks
200K~2M cells

Industrial Benchmarks
1M~10M cells

10M-cell design finishes within 5min, instead of 3+ hrs

DREAMPlace Open-Sourced

- Leverage AI hardware and software development recently
- Decouple core algorithm innovations with implementation
  - Algorithm innovation written in high-level language, e.g. Python
  - Highly extensible: new solver options, new design objectives, …
  - Implementations just focus on certain low-level kernel OPs as needed
- Development effort: 1 year ➔ 2 months
- The paradigm can be extended to **other DA areas**
  - Significantly enhance IC design productivity and quality

MAGICAL: Machine Generated Analog IC Layout

Open source MAGICAL 0.2
https://github.com/magical-eda/MAGICAL

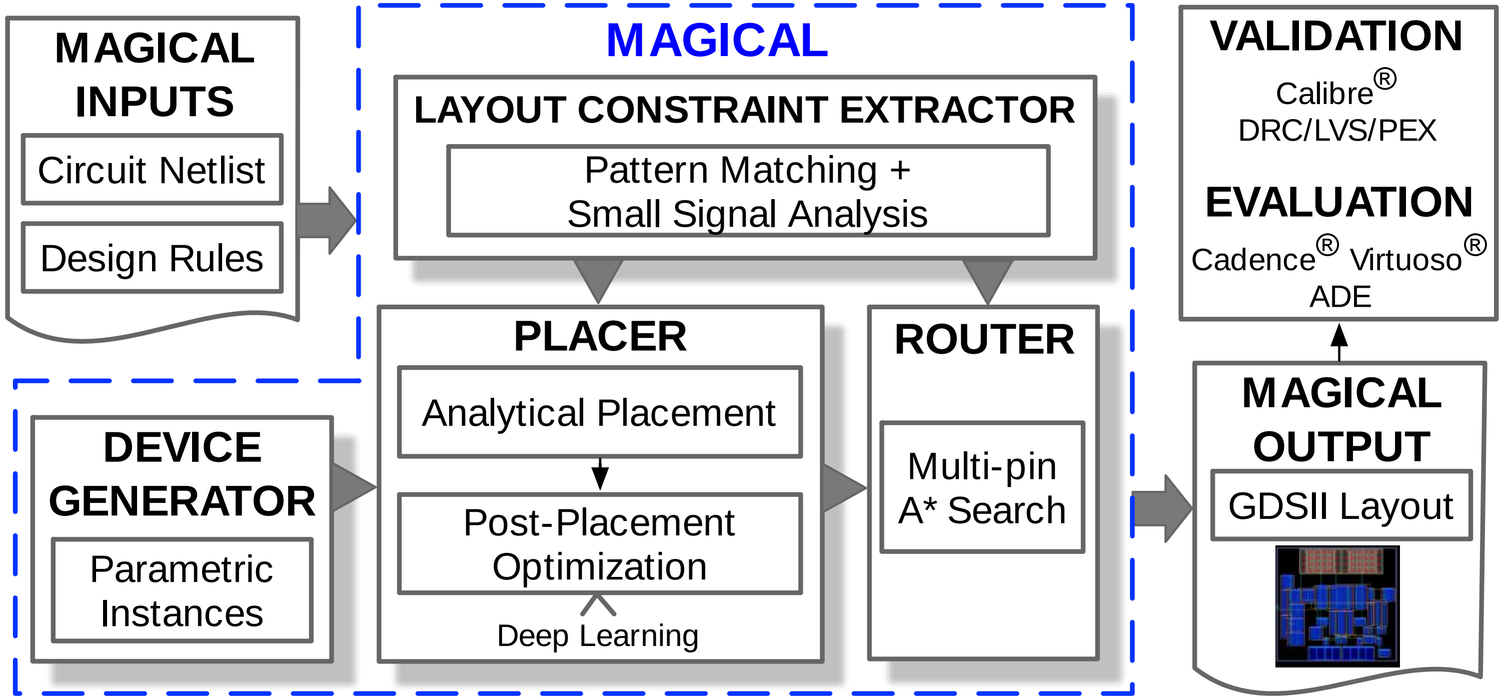
Analog IC Layout

- DREAMPlace focus on digital IC
- Analog IC to interface with outside world
- Analog IC layout design still **mostly** manual
  - Very tedious and error-prone
  - Prior DA not successful as that in digital IC

- Our mission is to develop a full-automated analog layout system, leveraging recent AI advancement
- Project started in 08/2018
- [ISPD’19; DAC’19; ICCAD’19; ASPDAC’20]

MAGICAL Layout System Framework

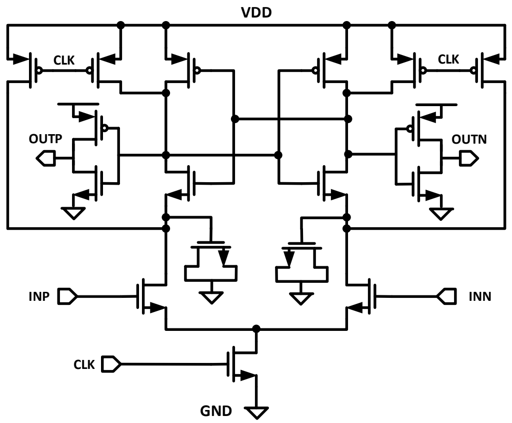
- Input: unannotated netlist
- Output: GDSII Layout
- Key Components:
  - Constraint Extraction
  - Device Generation
  - Placement
  - Routing

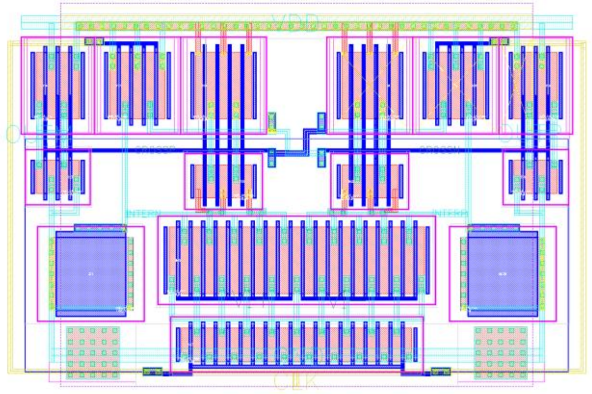


- **Fully-automated (no-human-in-the-loop)**
- Guided by analytical, heuristic, and machine learning algorithms (not everything is machine learning or deep learning!)

MAGICAL Preliminary Results

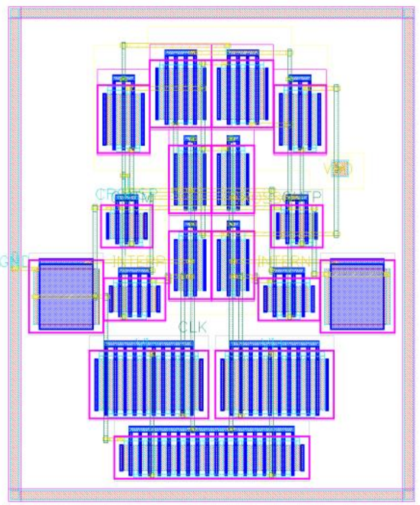
- A comparator design in 40nm TSMC





Manual Layout (taped out)

Days



MAGICAL Layout

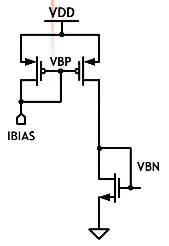
Seconds

Post extraction simulation results

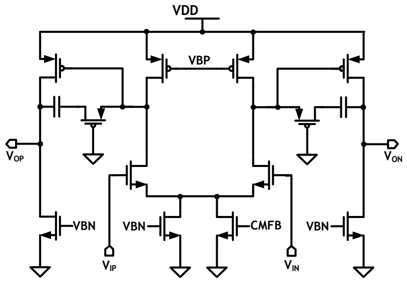
	Manual	MAGICAL
Power (uW)	16.8	18.7
Output Delay (ps)	150	152
Input-referred Noise (uVrms)	380	334
Input-referred Offset (mV)	0.15	0.50

MAGICAL Preliminary Results

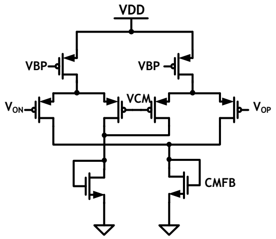
- A 2-stage miller-compensated OTA design in 40nm TSMC



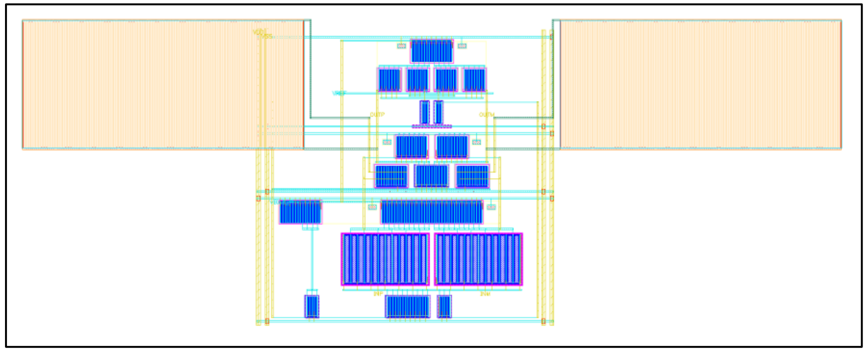
Bias



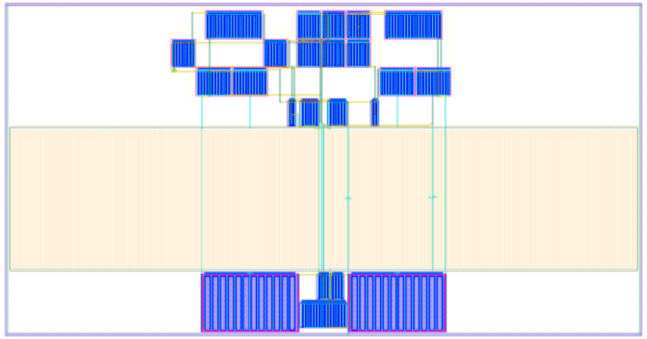
Gain Path



Common-mode Feedback



Manual Layout



MAGICAL Layout

Post extraction simulation results

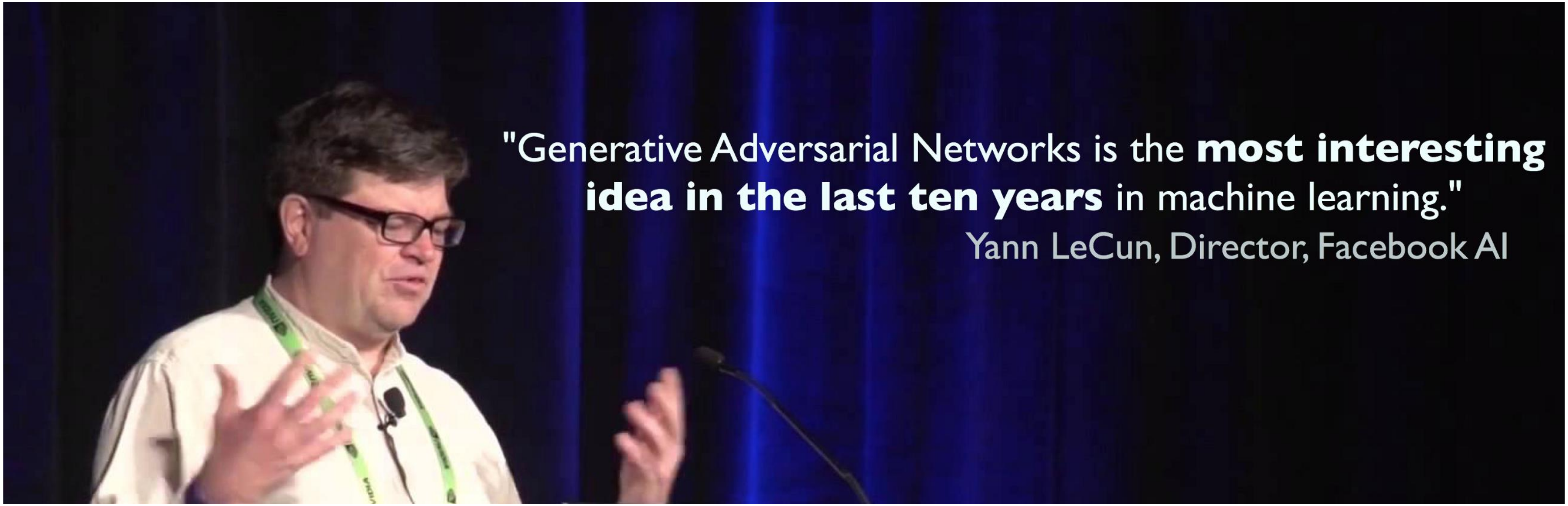
	Manual	MAGICAL
DC Gain (dB)	37.7	38.0
Unity-gain Bandwidth (MHz)	110	107.5
Phase Margin (degree)	67.8	62.3
Input-referred Noise (uVrms)	219	221.5
CMRR (dB)	103	92.5
Input-referred Offset (mV)	0.2	0.48

LithoGAN: End-to-End Lithography Modeling with Generative Adversarial Networks [Ye+, DAC’19 BPA Candidate]

Harder question cf. lithography hotspot detection: Without going through litho-simulations, can we directly get printed images?

GAN and CGAN

- Generative Adversarial Network (GAN) [Goodfellow et al, 2014]
  - Two neural networks contest (Generator and Discriminator)
  - Produce images similar to those in the training data set
- Conditional GAN (CGAN) can take a picture in one domain and translate it to another one [Isola et al, CVPR'17]

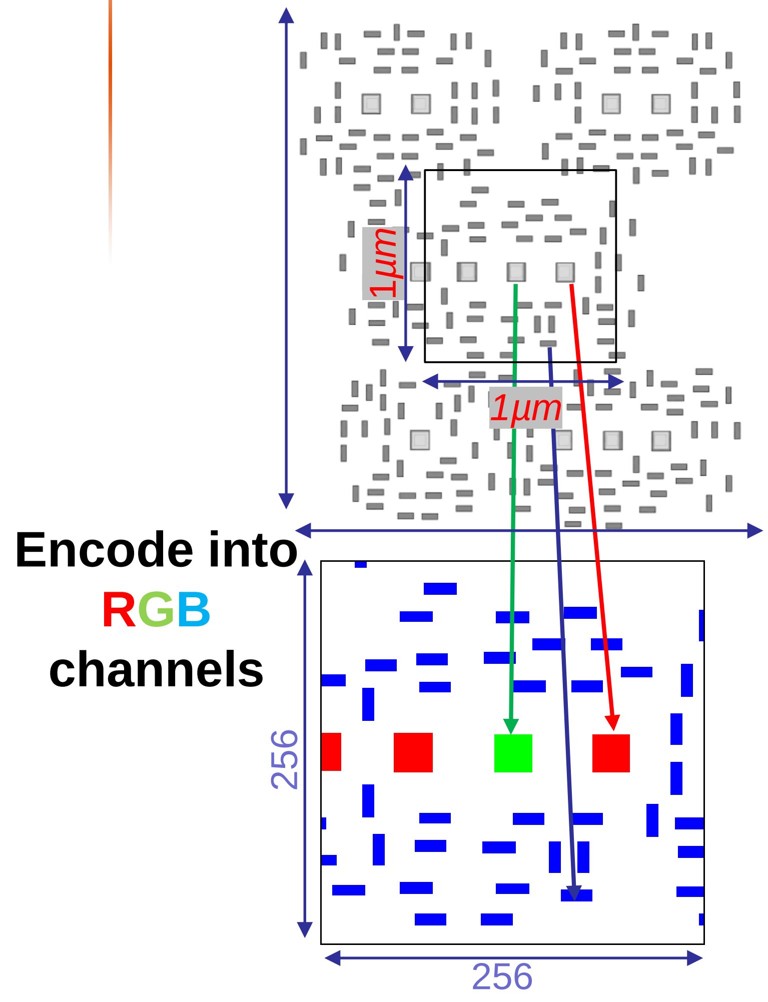


"Generative Adversarial Networks is the **most interesting idea in the last ten years** in machine learning."

Yann LeCun, Director, Facebook AI

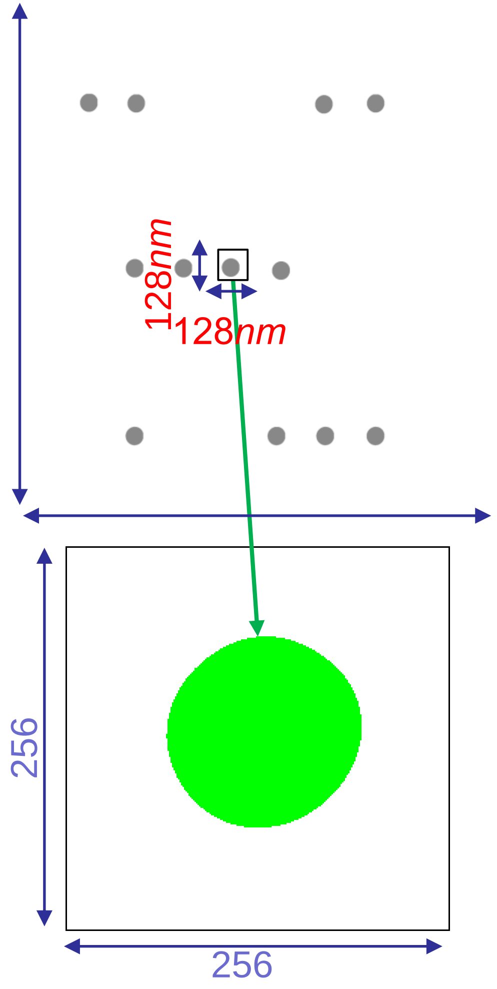
Image Translation for Litho Modeling

[Ye+, DAC'19]



Expensive Litho Simulation

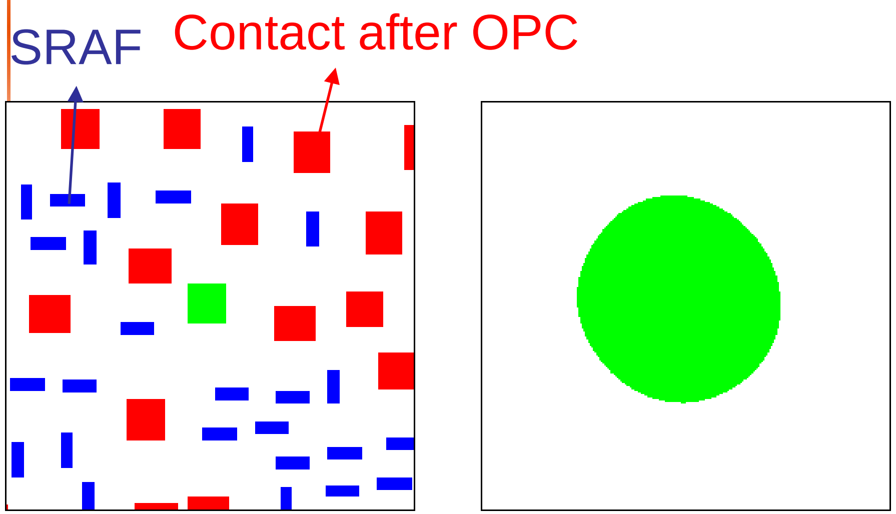
This is now a (modified) **image translation** task



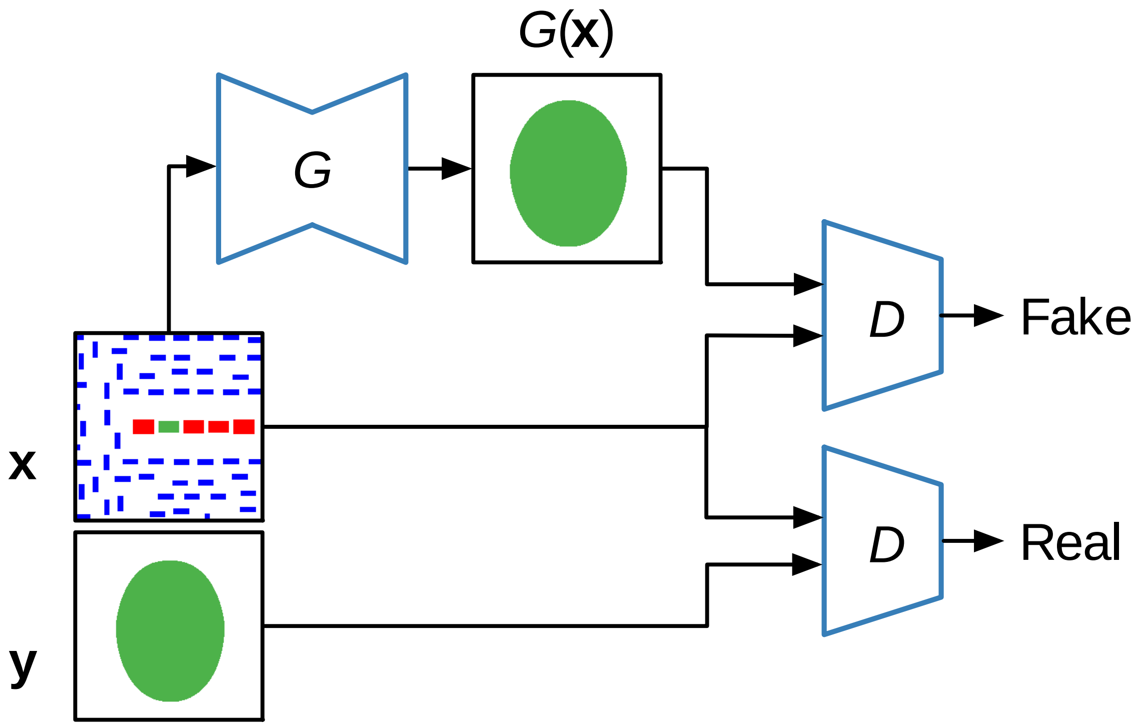
- Different elements encoded on different image channels
- Resist pattern zoomed in for high-resolution/accuracy

CGAN for Lithography Modeling

[Ye+, DAC’19]



Sample pairs for training CGAN

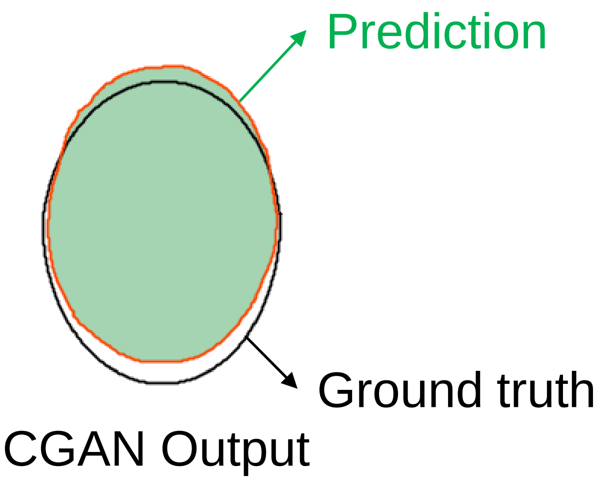


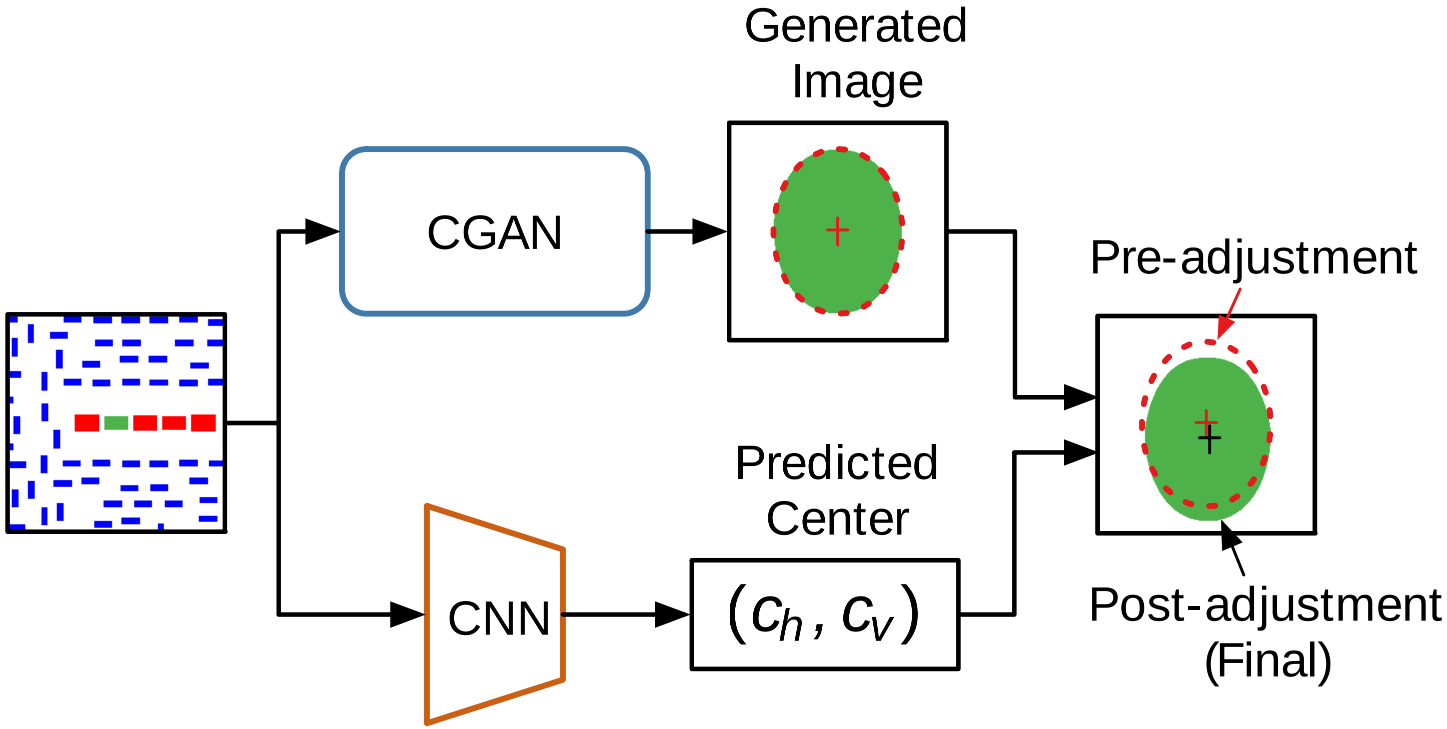
- CGAN architecture
  - Generator G generates a fake resist pattern G(x) with input mask pattern x
  - Discriminator D needs to classify the image pair (x, G(x)) as fake and to predict the image pair (x, y) as real

Overall LithoGAN Algorithm

[Ye+, DAC'19]

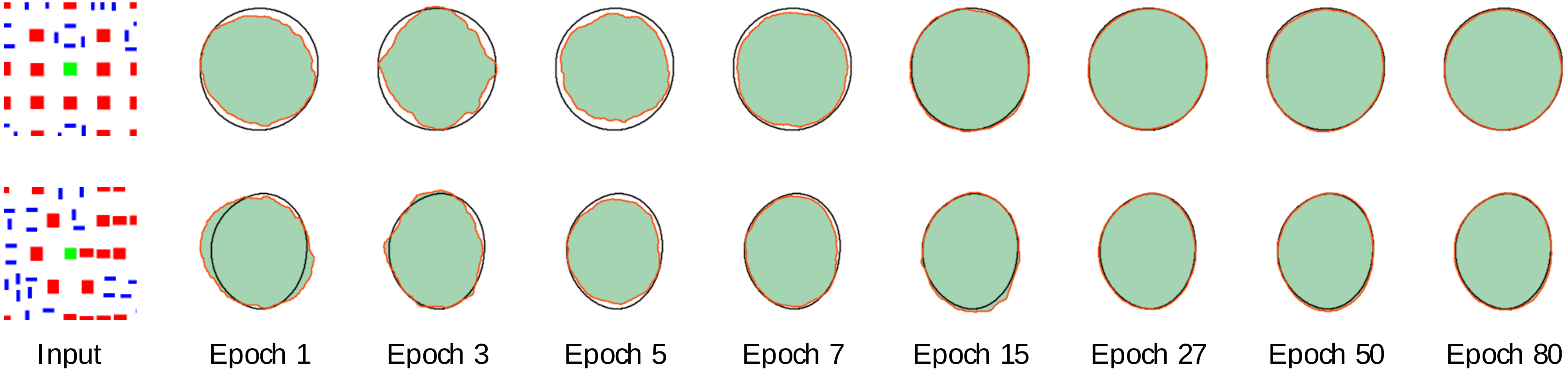
- Dual learning framework
  - CGAN to predict the shape of the resist pattern
  - CNN to predict the resist shape center



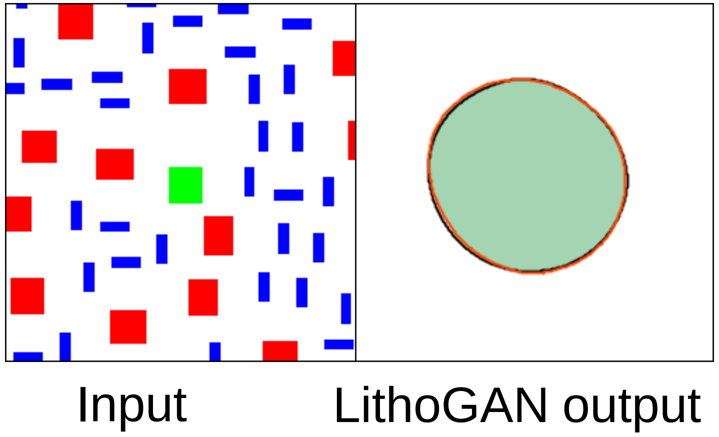


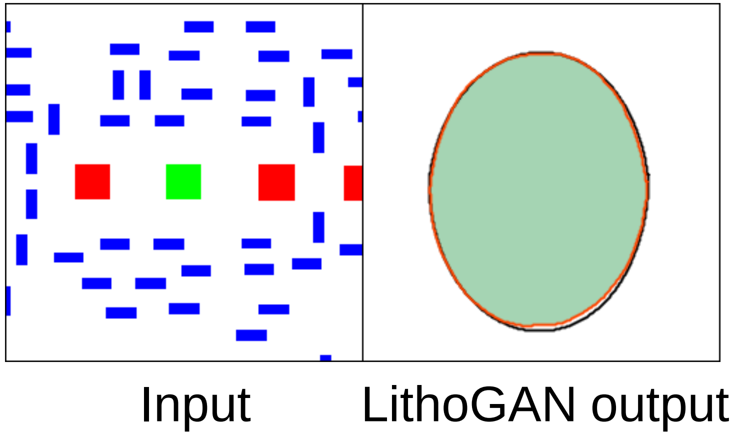
LithoGAN Results

[Ye+, DAC’19]



Model advancement progress

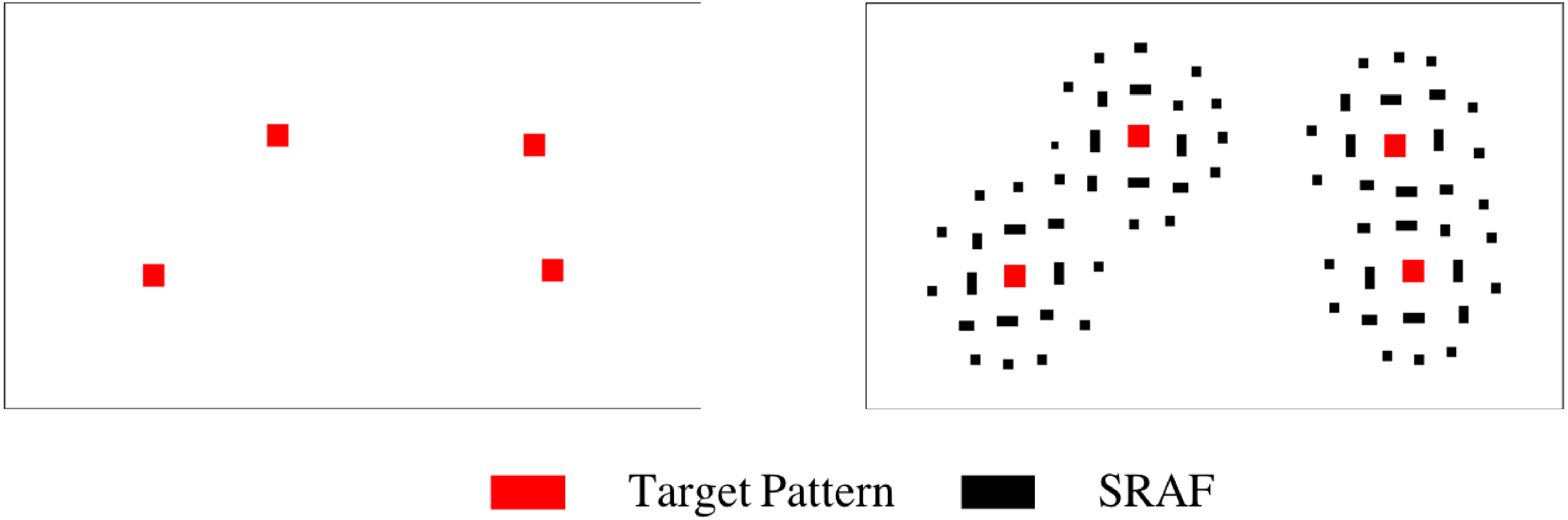




LithoGAN is **1800x** faster than rigorous simulations, with acceptable error (in consultation with industry)

GAN-SRAF [Alawieh+, DAC’19]

- Sub-Resolution Assist Feature Generation using Conditional Generative Adversarial Networks
- Directly generate sub-resolution assist feature (SRAF)
- **144x** faster than model based approach with similar QoR



LAPD

Another LAPD

- To bridge design and manufacturing, we propose Lithography Aware Physical Design (LAPD) ➔
  - Litho Hotspot Detection
  - Litho Hotspot Correction

- My group has made many key contributions in LAPD ⇧
- **LithoGAN** opens new directions with tremendous potential

Detection

Correction

Design/Manufacturing for Hardware Security

- Global IC supply chain of design, manufacture, test, package...

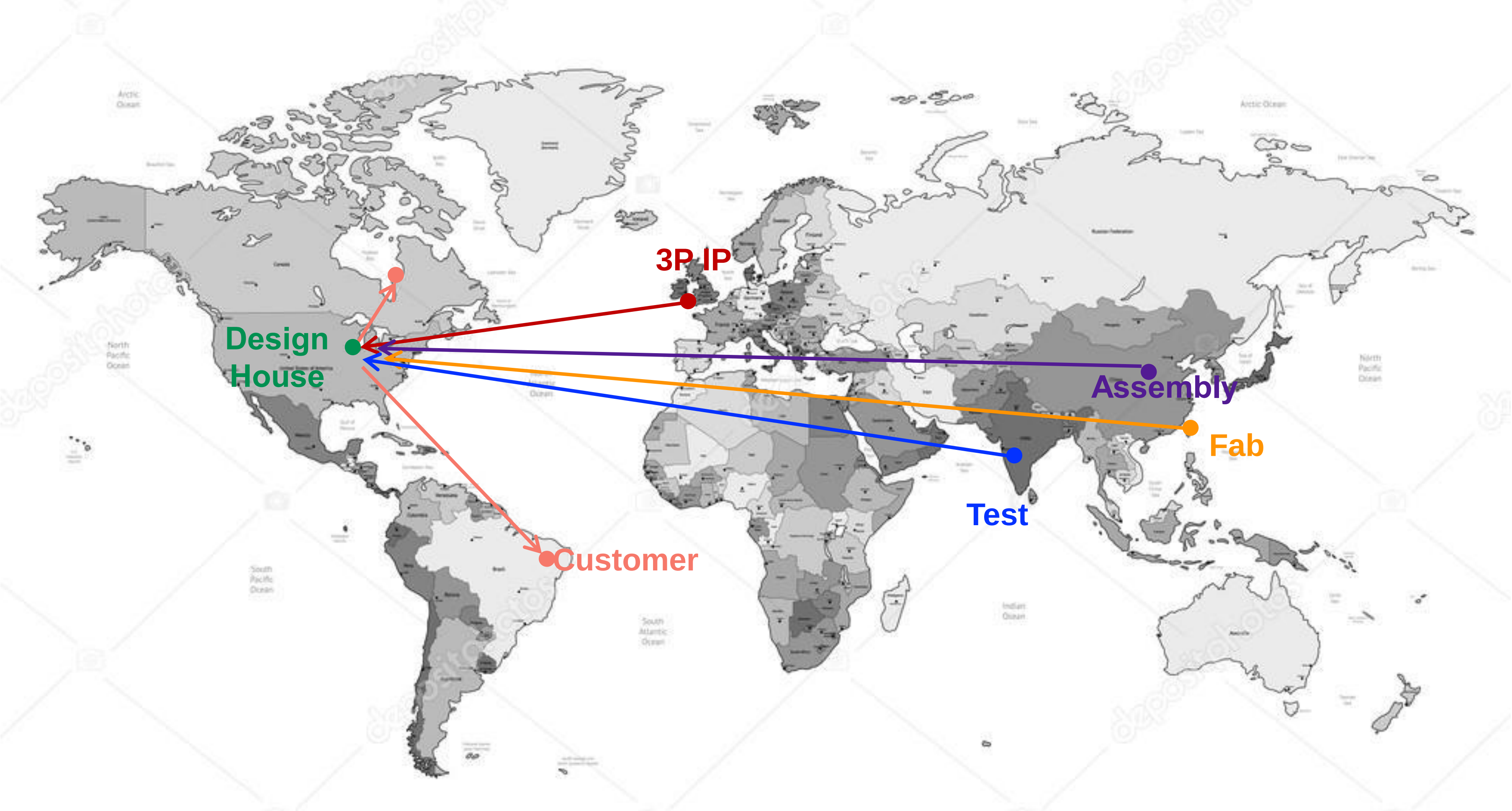
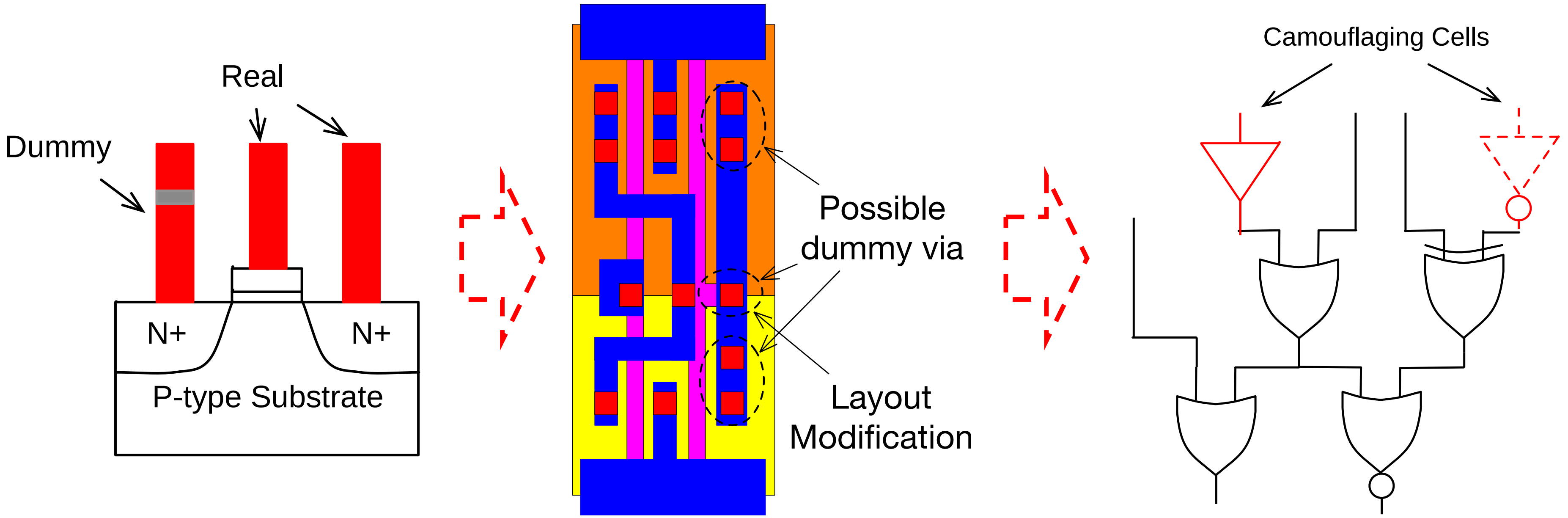


Image source: https://depositphotos.com/2801291/stock-illustration-gray-detailed-world-map.html

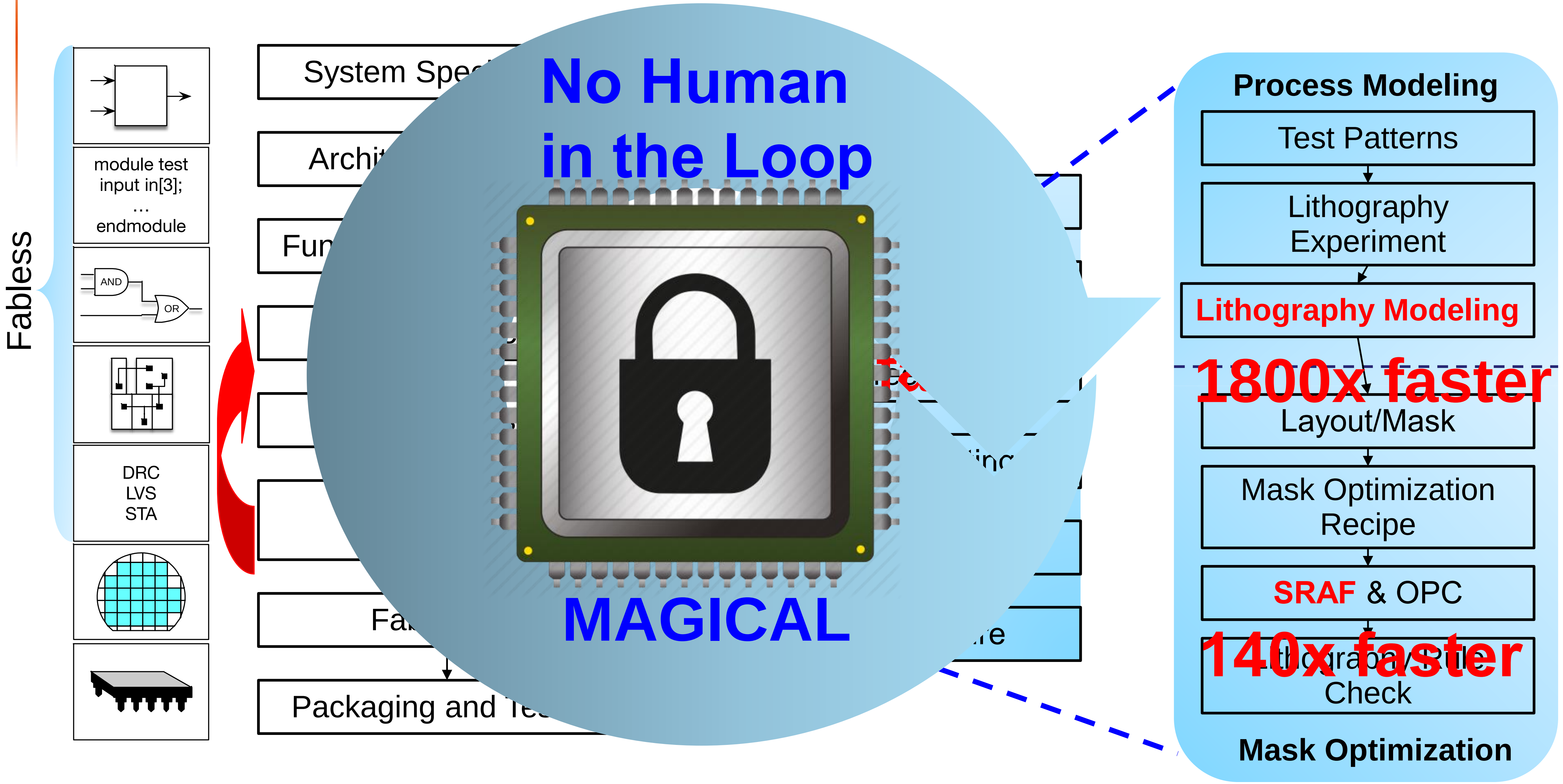
Design/Manufacturing for Hardware Security

- Arm race between attacking and protection
- Hardware IP reverse engineering using learning techniques
- Intelligent IC camouflaging [Li+, ICCAD’16, TCAD’17, HOST’17 BPA]
- Former PhD Meng Li won **ACM SRC Grand Finals First Place** in 2018



Fabrication Level Cell Level Netlist Level

To Recap



Fabless

module test
input in[3];
...
endmodule

DRC
LVS
STA

System Spec

Archit

Fun

Fa

Packaging and Te

No Human in the Loop

MAGICAL

Process Modeling

Test Patterns

Lithography Experiment

Lithography Modeling

1800x faster

Layout/Mask

Mask Optimization Recipe

SRAF & OPC

140x faster

Lithography Rule Check

Mask Optimization

Conclusion

- Some recent results in AI-enabled agile IC design & DFM
  - DREAMPlace
  - MAGICAL: GeniusRoute…
  - LithoGAN
- Tremendous potentials to leverage both AI hardware and software advancements
- BIG data, small data, or no (training) data at all (by recasting problems e.g., placement into DREAMPlace)
- Synergistic AI-IC co-design

Acknowledgment

- Funding support / collaborations from NSF, DARPA, Intel, Nvidia, Toshiba Memory, Xilinx, Synopsys
- **Many students/post-docs** who do the real work
- Many collaborators in academia and industry
  - Dr. Ren and Dr. Khailany from NVIDIA for DREAMPlace
  - Prof. Nan Sun at UT for MAGICAL
  - Dr. Nojima et al. from Toshiba Memory on DFM

Thanks!

Q & A?