MACHINE LEARNING FOR EDA OPTIMIZATION

Chia-Tung (Mark) Ho, NVResearch ASIC/VLSI group, 10/5/2023
AGENDA

Background & Motivation

Machine Learning for EDA Optimization

Conclusions
BACKGROUND & MOTIVATION
BACKGROUND & MOTIVATION

• Optimization is one of the fundamental problems in EDA
• Goal: Improve power, performance, area, and cost (PPAC)

\[
\begin{align*}
\text{Minimize } & f(x_1, x_2, \ldots, x_n) \\
\text{Subject to } & g(x_1, x_2, \ldots, x_n)
\end{align*}
\]

• The functions might be non-linear, non-convex, and discrete
• Design challenges of modern circuit design in advanced nodes
  • billions of transistors
  • Increasing number and complexity of design rules
  • Routability
  • Strict pattern rules
• ML opportunities: Improve the productivity, efficiency, and quality
NVCELL: REINFORCEMENT LEARNING BASED STANDARD CELL GENERATION WITH NOVEL TRANSFORMER MODEL-BASED CLUSTERING

Publication:
STANDARD CELL LAYOUT AUTOMATION

- Std cells are building blocks of digital design layout: AND, NOR, Flip-Flop, Adder, etc
- Layout mostly by hand today, long design turn around time for the library (a few months)
- Standard cell automatic layouts - Fast design turn around time, More custom cell design, Design Technology Co-Optimization

GA100 - 1.7B standard cells
Schematic
Simplified Grid-based layout diagram (Sticks)
Standard ell

Gate
MD
M0
M1
ROUTABILITY AND PPA-DRIVEN STANDARD CELL DESIGN AUTOMATION

- Standard cell layout design automation challenges as advancing beyond 5nm
  - **Limited in-cell routing resource**: less routing tracks (i.e., < 5 RTs)
  - **Design rule complexity**: Increasing number and complexity of design rules + strict patterning rules
  - **Scalability**: > hundreds of transistors cell designs

- **Multi-Objective Optimization**: Scalability, Routability, and high quality on PPA metrics

---

**Standard Cell Scaling Roadmap from IMEC**

**Routability challenges of a Latch Design in advanced node**
NVCELL: STANDARD CELL DESIGN AUTOMATION FRAMEWORK

- Layout-Aware Transformer Model Based Device Clustering
  - High-Quality clustering to reduce complexity, narrow down searching space, and assist finding routable solutions
- Lattice graph routability model in SA placer
  - Capture the local pin density and global net connections
- Reinforcement learning agent for DRC fixing
  - Model DRC fixing as a game to improve productivity and efficiency
LAYOUT-AWARE DEVICE CLUSTERING

- High quality clustering should consider transistor layout: Diffusion break/sharing, Transistor pin access, and Routing metal DRCs
- Reduce complexity, Narrow down searching space, and Assist finding routable layouts
  - Transformer model-based clustering approach

Global receptive field, netlist information, and device placement relations
LAYOUT-AWARE DEVICE CLUSTERING

Schematic of cell Logics (.sp) (Netlist Information)

Transformer Model Based Clustering

Transformer Encoder

Post Process

Device Clusters

Generated LVS/DRC Clean Latch Design (~ 100 devices)

Manual Cell Width = 58 / Generated Cell Width = 56 TWL = 671

Attention heat map of the Generated LVS/DRC Clean Latch Design (~ 100 devices)
TRANSFORMER ENCODER ARCHITECTURE

Goal: Given netlist logic graph, learn the relationship between device pairs in the Layout graph

(a) Transformer Encoder Architecture

(b) Training Flow: Similarity loss ($L_{sim}$) from layout graph

$$L_{sim} = \sum_v \left( - \sum_{i \in N(v)} \log (\sigma(y_v^T y_u)) - \sum_{k \sim \text{rand}} \log (\sigma(-y_v^T y_k)) \right)$$

$\sigma(y_v^T y_u)$: Preferred clustering probability of two devices.

$N(v)$: The neighbor of device $v$ in the layout

Layout graph (Neighbor columns all connected)

Representative embedding of each device, $y_v$. 
**ROUTABILITY-DRIVEN PLACEMENT**

- Simulated Annealing based algorithm for placement: Swap, Move, Flip

  - Routability Model: Lattice graph routability model predicts congestion and routability probability
    - Capture the interactions of local pin access and global nets given the placement

  ![Lattice graph routability model](image)

**Swap, move, and flip of placement sequence**

- Swap PMOS/NMOS pair segment \((p_3,p_4)\) with \((p_1,p_2)\)
- Move PMOS segment \((3,4)\) from \(p_1\) to \(p_3\)
- Flip NMOS segment \((2,5,4)\)
LATTICE GRAPH ROUTABILITY MODEL OVERVIEW

• Given: Circuit, transistor placement, and M1 Pin Placement Information
• Predict: Demanded routing resource and routability probability of each column
  • $\hat{y}_{\text{reg}}$ : demanded routing resource (hori/vertical) at each column. dim = 1 x cell columns
  • $\hat{y}_{\text{rout}}$ : routability probability at each column. dim = 1 x cell columns

Model Input: Lattice Graph from given transistor placement
Model Output: Routability Probability and Horizontal Congestion Prediction

Graph Neural Net
Shared MLP
MLP
MLP

Circuit Net
Net1
Net3
Net2

Optional: External Pin Layer lattice grid (i.e., M1)
FEOL lattice grid

D: Source/Drain Diffusion; G: Gate

$D$ $G$ $D$ $G$ $D$ $G$ $D$ $G$ $D$
TRAINING LATTICE GRAPH ROUTABILITY MODEL

• Regression Loss Function:

\[ L_{reg} = -\frac{1}{N} \sum(y_{reg} - \hat{y}_{reg})^2 \]

• Routability Probability Loss Function:

\[ L_{rout} = D_{KL}(Y_{rout}||\hat{Y}_{rout}) = Y_{rout} \log \frac{Y_{rout}}{\hat{Y}_{rout}}, \quad Y_{rout} = \text{Softmax}(y_{rout}), \quad \hat{Y}_{rout} = \text{Softmax}(\hat{y}_{rout}) \]
ROUTING

- Leverage maze routing to generate routing candidates
  → solve the connectivity problem
- Leverage RL to fix DRC of the routing candidates
  → solve the DRC problem
- Leverage genetic algorithm to minimize unroutable nets and DRC numbers
  → solve the optimization problem
GAME OF FIXING DRC

Adding additional M0 grid to reduce DRCs

Step 0: DRC = 6

Step 1: DRC = 6
reward=0

Step 2: DRC = 3
reward=3

Step 3: DRC = 0
reward=3
FIX DRC WITH RL

Initial Layout  
Stick DRC  
DRC Game Env  
Final Layout  

DRCs

Observation:
Routing: \([H^{M1}, W^{M1}]\)
DRC: \([H^{M1}, W^{M1}]\)
Mask: \([H^{M1}, W^{M1}]\)

DRC RL Agent

select an action based on action probability: \([H^{M1} \times W^{M1}]\)
DRC RL MODEL

Agnostic to design size

Observation: $[H^{M_1}, W^{M_1}]$
Routing: $[H^{M_1}, W^{M_1}]$
DRC: $[H^{M_1}, W^{M_1}]$
Mask: $[H^{M_1}, W^{M_1}]$

State Embedding $[H^{M_1}, W^{M_1}]$

Action Probability $[H^{M_1} \times W^{M_1}]$

softmax

State Value

Pool
DRC AGENT TRAINING

RL algorithm: PPO2 in stable-baselines

Training set: 10000 random maze routes for a flip-flop cell

Generalizes to all the cells

Random route 1

Random route 2

Reward history of 9 training runs
DRC FIXING EXAMPLE
LAYOUT AND PERFORMANCE EVALUATION

Create 100% cells in industrial standard cell library

<table>
<thead>
<tr>
<th></th>
<th>Success Rate (%)</th>
<th>Cell Width Comparison</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Smaller</td>
</tr>
<tr>
<td>NVCell (DAC 2021)</td>
<td>91.2%</td>
<td>11.8%</td>
</tr>
<tr>
<td>NVCell2 (ISPD 2023)</td>
<td>98.8%</td>
<td>13.7%</td>
</tr>
<tr>
<td>Layout-Aware Clustering</td>
<td>100%</td>
<td>14.5%</td>
</tr>
</tbody>
</table>

On a difficult routing benchmark (94 cells)

Improved PPA metrics up to performance 7%, power 8%, and area 4%.
CONCLUSIONS

• We can leverage ML to improve chip design automation productivity and QoR.

• Transformer model based and generative model can be leveraged to improve the efficiency and solution quality for EDA optimizations.

• Algorithms + GPU acceleration + ML: A new EDA computing paradigm!

Source: SIGGRAPH 2023