GeniusRoute: A New Analog Routing Paradigm Using Generative Neural Network Guidance

Keren Zhu, Mingjie Liu, Yibo Lin, Biying Xu, Shaolan Li, Xiyuan Tang, Nan Sun and David Z. Pan
ECE Department
The University of Texas at Austin
This work is supported in part by the NSF under Grant No. 1704758, and the DARPA ERI IDEA program
Outlines

• Introduction and Problem Formulation
• GeniusRoute Framework
• Experimental Results
• Conclusion
High Demand of Analog/Mixed-Signal IC

- Anything related to sensors needs analog!
- Internet of Things (IoT), autonomous and electric vehicles, communication and 5G networks…
A Bottleneck in IC Design: Analog/Mixed-Signal

Analog parts of IC take large design efforts

A major reason: analog circuit layout is usually done manually

[IBS and Dr. Handel Jones, 2012]
Typical Automatic Analog Circuit Design Flow

- Automated analog design often consists of front-end and back-end flows
- Physical design (back-end) is separated in placement and routing
Analog Routing Problem

Placement

Routed Layout
Challenges in Formulating Analog Routing Problem

Symmetry constraints are widely accepted

No standard rule for additional constraints. Design-dependent.

Automatically learn from human layouts?

Shielding, Avoid active region, ...

[Ou et al., 2014]
Emerging Machine Learning Applications

Lithography: GAN-OPC

[Yang et al., 2018]

Physical Design: WellGAN

[Xu et al., 2019]
Automatically Learn Guidance from Human Layouts

- Learn routing guidance
  - Where the human would likely to route the nets
- Extract training data from labeled layouts
- Apply learned model to automatic routing as guidance
A ML-Guided Routing Problem

**Heuristic constraints:** use a set of detailed heuristics as routing constraints

**Conventional Approach**

Placement → Explicated Constraints → Routing

**Routing guide:** routing strategies learned from human

**GeniusRoute Approach**

Placement → Symmetric Constraints + ML-based Routing Guide → Routing
The GeniusRoute Flow

- Learn from GDS layouts
- Pre-process layouts into images
- Predict routing probability using autoencoder
- Use prediction as detailed routing guidance
Generating Images with Generative Neural Network
Data-Preprocessing: Extracting Routing from Layouts

Extract “pins” and routing of nets

Three categories of models:

• Symmetric nets
• Clocks
• Power and Ground
GeniusRoute: Learning Routing Patterns from Human

**Training Phase**

- Pins of Entire Design → Neural Network → Minimize Loss → Ground Truth: Manual Routing

**Inference Phase**

- Do we have enough data?

- Pins of Entire Design → Neural Network → Generated Routing Region → Trained

- pins of Interested Nets → Downstream AMS Router → Routed Layout
3-Stage Semi-supervised Training Algorithm

- Labeled layouts are hard to get
- Could rely on unlabeled data to help train the model
Stage 1: Unsupervised Feature Extraction using VAE

Use cheap unlabeled data to learn a general feature extraction
Network Architecture: Unsupervised for Stage 1
Stage 2: Supervised Decoder Training

Fix the feature extraction to learn the generative model
Stage 3: Supervised Decoder Fine-Tune

Fine-tune the network for better accuracy with lower learning rate
Network Architecture: Supervised for Stage 2&3
Framework Implementation and Environment Setup

- Data preprocessing: C++
- ML model: Python with Tensorflow
- Router: Modified maze routing in C++
- Simulation: Cadence ADE simulator with TSMC 40nm PDK
Experimental Result Examples

Model Output

Ground Truth

Prediction

Routed Layout
Experimental Results: Simulation Results

- Test on comparators and OTAs
- Evaluate with post layout simulation
- Compare with manual layout and previous methods

<table>
<thead>
<tr>
<th>COMP1</th>
<th>Schematic</th>
<th>Manual</th>
<th>w/o guide</th>
<th>GeniusRoute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Offset (uV)</td>
<td>/</td>
<td>480</td>
<td>2530</td>
<td>830</td>
</tr>
<tr>
<td>Delay (ps)</td>
<td>102</td>
<td>170</td>
<td>164</td>
<td>163</td>
</tr>
<tr>
<td>Noise (uVrms)</td>
<td>439.8</td>
<td>406.6</td>
<td>439.7</td>
<td>420.7</td>
</tr>
<tr>
<td>Power (uW)</td>
<td>13.45</td>
<td>16.98</td>
<td>16.82</td>
<td>16.8</td>
</tr>
</tbody>
</table>

Closer results to the manual layout
### Experimental Results: More Simulation Results

<table>
<thead>
<tr>
<th>COMP1</th>
<th>Schematic</th>
<th>Manual</th>
<th>w/o guide</th>
<th>GeniusRoute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Offset (uV)</td>
<td>/</td>
<td>480</td>
<td>2530</td>
<td>830</td>
</tr>
<tr>
<td>Delay (ps)</td>
<td>102</td>
<td>170</td>
<td>164</td>
<td>163</td>
</tr>
<tr>
<td>Noise (uVrms)</td>
<td>439.8</td>
<td>406.6</td>
<td>439.7</td>
<td>420.7</td>
</tr>
<tr>
<td>Power (uW)</td>
<td>13.45</td>
<td>16.98</td>
<td>16.82</td>
<td>16.8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>COMP2</th>
<th>Schematic</th>
<th>Manual</th>
<th>w/o guide</th>
<th>GeniusRoute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Offset (uV)</td>
<td>/</td>
<td>550</td>
<td>1180</td>
<td>280</td>
</tr>
<tr>
<td>Delay (ps)</td>
<td>102</td>
<td>196</td>
<td>235</td>
<td>241</td>
</tr>
<tr>
<td>Noise (uVrms)</td>
<td>439.8</td>
<td>380.0</td>
<td>369.6</td>
<td>367.8</td>
</tr>
<tr>
<td>Power (uW)</td>
<td>13.45</td>
<td>20.28</td>
<td>20.23</td>
<td>20.15</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>OTA</th>
<th>Schematic</th>
<th>Manual</th>
<th>w/o/ guide</th>
<th>GeniusRoute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gain (dB)</td>
<td>38.20</td>
<td>37.47</td>
<td>36.61</td>
<td>37.36</td>
</tr>
<tr>
<td>PM (degree)</td>
<td>64.66</td>
<td>72.46</td>
<td>94.68</td>
<td>76.40</td>
</tr>
<tr>
<td>Noise (uVrms)</td>
<td>222.0</td>
<td>223.7</td>
<td>292.7</td>
<td>224.8</td>
</tr>
<tr>
<td>Offset (mV)</td>
<td>/</td>
<td>0.88</td>
<td>3.21</td>
<td>0.39</td>
</tr>
<tr>
<td>CMRR (dB)</td>
<td>/</td>
<td>59.61</td>
<td>58.52</td>
<td>59.15</td>
</tr>
<tr>
<td>BW (MHz)</td>
<td>110.5</td>
<td>102.5</td>
<td>232.1</td>
<td>107.3</td>
</tr>
<tr>
<td>Power (uW)</td>
<td>776.93</td>
<td>757.35</td>
<td>715.11</td>
<td>787.82</td>
</tr>
</tbody>
</table>
Conclusion

GeniusRoute

• A new methodology to automatic learn from human layout and apply in automatic flow

• Semi-supervised learning algorithm for data-efficiency

• Experimental results show closed-to-human post layout simulation

Future directions

• How to overcome the challenge of obtaining human layouts for labeled data
Thank you!