High-Definition Routing Congestion Prediction for Large-Scale FPGAs

Mohamed Baker Alawieh¹, Wuxi Li¹, Yibo Lin², Love Singhal³, Mahesh Iyer³ and David Z. Pan¹

¹ECE Department, University of Texas at Austin
²CS Department, Peking University
³Intel Corporation, USA
FPGA Routing Congestion Prediction

Field Programmable Gate Arrays
High Energy Efficiency
Good Reprogrammability
Rapidly Growing Capacity

FPGA Placement
Has a significant impact on FPGA routing quality

Routability Aware
Incorporates congestion prediction into the placement process

Congestion
Primitive congestion prediction techniques have demonstrated significant impact on routing quality
Conventional Approaches

**RouteNet**
Predicts congestion hotspot
Design rule violation detection
[Xie+, ICCAD’18]

**RUDY**
Bounding box-based routing estimation
Overestimates the routing demand
[Spindler+, DATE’07]

**GAN-Based**
Predicts congestion based on placement
Cannot handle industrial-size designs
[Yu+, DAC’19]

**Regression-based Prediction**
Congestion prediction based on global routing info
[Pui+, ICCAD’17]
Conditional GANs for Image Translations

**GANS**
Generative Adversarial Networks
Generate Images from a distribution

**CGANS**
Conditional GANs
Generate an image based on input

**Image Translation**
CGANs can be used for the task
Apply domain transfer
Take image from one domain and generate output in another
During training, pairs of matched images are used

[cartoon credit: Gall, 18, dzone.com]
**GAN-based Congestion Estimation**

[Yu+, DAC’19]

### Features

Uses VTR academic tool  
Works for small designs only

Netlist information is encoded using flying lines  
*For a large design with over 700K nets

This representation becomes obsolete for large designs

---

**GAN Model**

*pix2pix* model [Isola+, CVPR 2017]  
Limited resolution 256x256  
Cannot handle large-scale FPGAs
GAN Model

*pix2pix* model [Isola+, CVPR 2017]
Limited resolution 256x256
Cannot handle large-scale FPGAs

![FPGA chip](image)

Virtex UltraScale+ VU19 has
~663K CLB slices

Use a high definition image translation model
Handle resolution up to 4000x1000

Features

Uses VTR academic tool
Works for small designs only

Novel feature encoding for placement and netlist
Use different channels of input image
Input Features Encoding

Pin Density
Reflects placement information
Encoded on the blue channel

Vertical Demand
Estimates vertical routing demand
Computed analogous to RUDY
Encoded on green channel

Horizontal Demand
Estimates vertical routing demand
Computed analogous to RUDY
Encoded on red channel

Resulting RGB image
Output Features Encoding

**Vertical Routing**
Routing congestion along the vertical direction

**Horizontal Routing**
Routing congestion along the horizontal direction

Resulting RGB image

*Blue channel left empty*
Generator Design

Dual generator architecture
For high resolution generation

Global Generator ($G_1$):
Performs the core translation
Works at half desired resolution

Local Enhancer ($G_2$):
Generates high resolution images
Fine-tunes details in the image
**High Definition Image Translation**

**Generator Design**

Dual generator architecture
For high resolution generation

Global Generator ($G_1$):
Performs the core translation
Works at half desired resolution

Local Enhancer ($G_2$):
Generates high resolution images
Fine-tunes details in the image

**Discriminator Design**

Three level discrimination

- $D_1$
- $D_2$
- $D_3$

Real Image  
Synthesized Image

Scale 1/4  
Scale 1/2  
Scale 1
High Definition Image Translation

Generator Design
Dual generator architecture
For high resolution generation

Global Generator ($G_1$):
Performs the core translation
Works at half desired resolution

Local Enhancer ($G_2$):
Generates high resolution images
Fine-tunes details in the image

Discriminator Design
Three level discrimination

Loss Function
GAN Loss
Feature Loss Mapping loss

$$\min_G \max_{D_1, D_2, D_3} \sum_{k=1,2,3} \mathcal{L}_{GAN}(G, D_k) + \lambda \mathcal{L}_{FM}(G, D_k)$$

$$\mathcal{L}_{GAN}(G, D_k) = \mathbb{E}_{x, y} [\log D_k(x, y)] + \mathbb{E}_x [\log (1 - D_k(x, G(x)))]$$

$$\mathcal{L}_{FM}(G, D_k) = \mathbb{E}_{x, y} \sum_{i=1}^T ||D_k(x, y) - D_k^i(x, G(x))||_1$$
Experimental Setup

**Benchmark**
ISPD 2016
Placement: elfPlace [Li+, ICCAD’19]
Routing: NCTU-GR [Liu+, TCAD’13]

For each design:
200 placements are generated
Placements are routed
Congestion maps obtained

**Training Setup**
Train 12 different models
11 for train, 1 for test

**Evaluation Metrics**
NRMS:
Normalized root mean square

**Comparisons**
1. GAN-Based [Yu+, DAC’19]
   - Updated features
   - Proper scaling

2. RUDY [Spindler+, DATE’07]

**Resources**

<table>
<thead>
<tr>
<th>Feature</th>
<th>Resources</th>
</tr>
</thead>
<tbody>
<tr>
<td>50K</td>
<td>1075K</td>
</tr>
<tr>
<td>55K</td>
<td>1728</td>
</tr>
<tr>
<td>100</td>
<td>768</td>
</tr>
<tr>
<td>12</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Difference in pixel distributions

SSIM:
Structural similarity index

EMD:
Earth moving distance
Sample Results – FPGA 08

Vertical Congestion

Golden

Proposed

pix2pix

RUDY

Horizontal Congestion

RUDY ~ [Spindler+, DATE’07]

pix2pix ~ [Yu+, DAC’19]*
Quantitative Comparison

RUDY ~ [Spindler+, DATE’07]

pix2pix ~ [Yu+, DAC’19]*

<table>
<thead>
<tr>
<th>Metric</th>
<th>RUDY Horizontal</th>
<th>pix2pix Horizontal</th>
<th>Proposed Horizontal</th>
<th>RUDY Vertical</th>
<th>pix2pix Vertical</th>
<th>Proposed Vertical</th>
</tr>
</thead>
<tbody>
<tr>
<td>NRMS</td>
<td>0.241</td>
<td>0.621</td>
<td>0.189</td>
<td>0.239</td>
<td>0.778</td>
<td>0.226</td>
</tr>
<tr>
<td>SSIM (higher)</td>
<td>0.407</td>
<td>0.523</td>
<td>0.752</td>
<td>0.616</td>
<td>0.439</td>
<td>0.656</td>
</tr>
<tr>
<td>EMD</td>
<td>0.162</td>
<td>0.225</td>
<td>0.137</td>
<td>0.137</td>
<td>0.233</td>
<td>0.127</td>
</tr>
</tbody>
</table>
In Placement

Models were used for routability estimation within elfPlaceF replacing RUDY

<table>
<thead>
<tr>
<th>Design</th>
<th>Full Routing Capacity</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Rudy</td>
<td>Proposed</td>
<td>Imp</td>
</tr>
<tr>
<td>FPGA-1</td>
<td>336117</td>
<td>336117</td>
<td>0.00%</td>
</tr>
<tr>
<td>FPGA-2</td>
<td>691618</td>
<td>691618</td>
<td>0.00%</td>
</tr>
<tr>
<td>FPGA-3</td>
<td>3062734</td>
<td>3062734</td>
<td>0.00%</td>
</tr>
<tr>
<td>FPGA-4</td>
<td>5550659</td>
<td>5551473</td>
<td>-0.01%</td>
</tr>
<tr>
<td>FPGA-5</td>
<td>10538770</td>
<td>9797007</td>
<td>7.04%</td>
</tr>
<tr>
<td>FPGA-6</td>
<td>5773333</td>
<td>5773333</td>
<td>0.00%</td>
</tr>
<tr>
<td>FPGA-7</td>
<td>9182199</td>
<td>9163640</td>
<td>0.20%</td>
</tr>
<tr>
<td>FPGA-8</td>
<td>9053192</td>
<td>9053192</td>
<td>0.00%</td>
</tr>
<tr>
<td>FPGA-9</td>
<td>11641853</td>
<td>11635870</td>
<td>0.05%</td>
</tr>
<tr>
<td>FPGA-10</td>
<td>5515319</td>
<td>5515319</td>
<td>0.00%</td>
</tr>
<tr>
<td>FPGA-11</td>
<td>11777500</td>
<td>11757650</td>
<td>0.16%</td>
</tr>
<tr>
<td>FPGA-12</td>
<td>6235694</td>
<td>6235694</td>
<td>0.00%</td>
</tr>
</tbody>
</table>

FPGA-5 is the most congested design
## Model Application

### In Placement

Models were used for routability estimation within elfPlaceF replacing RUDY

<table>
<thead>
<tr>
<th>Design</th>
<th>Full Routing Capacity</th>
<th>Reduced Routing Capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Rudy</td>
<td>Proposed</td>
</tr>
<tr>
<td>FPGA-1</td>
<td>336117</td>
<td>336117</td>
</tr>
<tr>
<td>FPGA-2</td>
<td>691618</td>
<td>691618</td>
</tr>
<tr>
<td>FPGA-3</td>
<td>3062734</td>
<td>3062734</td>
</tr>
<tr>
<td>FPGA-4</td>
<td>5550659</td>
<td>5551473</td>
</tr>
<tr>
<td>FPGA-5</td>
<td>10538770</td>
<td>9797007</td>
</tr>
<tr>
<td>FPGA-6</td>
<td>5773333</td>
<td>5773333</td>
</tr>
<tr>
<td>FPGA-7</td>
<td>9182199</td>
<td>9163640</td>
</tr>
<tr>
<td>FPGA-8</td>
<td>9053192</td>
<td>9053192</td>
</tr>
<tr>
<td>FPGA-9</td>
<td>11641853</td>
<td>11635870</td>
</tr>
<tr>
<td>FPGA-10</td>
<td>5515319</td>
<td>5515319</td>
</tr>
<tr>
<td>FPGA-11</td>
<td>11777500</td>
<td>11757650</td>
</tr>
<tr>
<td>FPGA-12</td>
<td>6235694</td>
<td>6235694</td>
</tr>
</tbody>
</table>

**FPGA-5 is the most congested design**

[elfPlace [Li+, ICCAD’19]]
Conclusions

We propose an accurate FPGA routing congestion estimation framework based on high-definition image translation.

Our proposed approach demonstrate superior accuracy compared to state-of-the-art techniques.

Our proposed approach results in up to 7% reduction in routed wirelength.
Future Work

♦ Further improve feature representation
  › Preserve original connectivity information in feature encoding

♦ Develop new placement algorithm built around such accurate congestion estimation

♦ Extend the application to ASIC