S$^3$DET: Detecting System Symmetry Constraints for Analog Circuit with Graph Similarity

Mingjie Liu$^1$, Wuxi Li$^1$, Keren Zhu$^1$, Biying Xu$^1$, Yibo Lin$^2$, Linxiao Shen$^1$, Xiyuan Tang$^1$, Nan Sun$^1$, and David Z. Pan$^1$

$^1$ECE Department, The University of Texas at Austin  
$^2$CS Department, Peking University
Analog/Mixed-Signal IC Design

典型现代SoCs:

• 少于25%的总芯片面积用于模拟；然而，75%或更多设计工作。

模拟/混合信号IC设计仍然在各个阶段高度手动化

• 非常耗时且容易出错

Image Sources: IBS and Dr. Handel Jones, 2012
Challenges in Analog Layout Automation

- Heavily rely on geometric constraints
  - Need to guarantee precise properties
  - Symmetry and ratio matching between devices

Comparator Schematic

Comparator Layout
System Symmetry Constraints

- System designs require matching between building block cells

Time-Interleaved SAR ADC

Die Photo
System Symmetry Constraints

- Mismatch could cause significant system performance degradation
  - 0.1% mismatch in clock timing would result in 15dB SNDR degradation
  - Require calibration (design techniques) + careful implementation (layout)

Mismatch in clock skew between SAR channels
Prior Works: Symmetry Constraint Detection

- Prior works focus on level symmetry constraints for building blocks
  - Symmetry between transistors (Mosfets and BJTs)
- Sensitivity analysis [Charbon, ICCAD’93]
  - Identify geometry constraints through electrical simulations
- Graph matching algorithms
  - Graph automorphism + signal flows [Hao, ICCAS’04] [Zhou, ASICON’05]
  - Template circuit + subgraph isomorphism [Wu, ECCTD’15]
  - Pattern library + structural signal flow graphs [Eich, TCAD’11]
Prior Works: Symmetry Constraint Detection

♦ Prior works face significant challenges when migrating to systems
  • Sensitivity analysis is unaffordable for system level designs: Transistor level spice simulations of ADCs take hours
  • Graph matching algorithms are computationally expensive: System designs normally consist over hundreds of devices
  • Difficult to generate templates/patterns for systems designs: Highly flexible and custom-designed architectures and circuits
  • Passive devices are critical in matching constraints: Capacitors and resistors
System Symmetry Constraints

- System design netlists contain hierarchy
  - Normally already well-partitioned based on functionality
  - Yield important design considerations
  - An over-simplified example:
System Symmetry Constraints

- System symmetry constraints:
  - Each node in the hierarchy tree should consist constraints between its children
**System Symmetry Constraints**

- **Netlist preprocessing**
  - Label cells as digital or analog, propagate label through hierarchy tree
  - Generate symmetry candidates: cells with same labels

- **Graph abstraction**
  - Vertices: device and pins, Edges: connections
  - Easily extendable to passive devices

---

**Label Propagation**

**Graph Abstraction**
Overall Flow of $S^3$DET

For any $v$ in the hierarchy graph:
   For any pair of children $(g_1, g_2)$ of $v$:
      Compare $(g_1, g_2)$ to identify symmetry constraint;
Symmetry ambiguity

- Only detecting subcircuits similarities does not work well in practice
- Designers tend to reuse building blocks if possible
- Widely used digital standard cells create lots of issues

\[ A, B, C, \text{ and } D \text{ are the similar filters} \]
\[ \text{Only } (A,B) \text{ and } (C,D) \text{ need matching} \]
\[ \text{Over-constraints, such as } (A,C) \text{ and } (A,D) \text{ create overhead in layout parasitic or infeasible floorplans} \]
Resolving symmetry ambiguity

- Extract neighboring circuit topology for each cell
- Determine symmetry based on extracted subgraph similarity

- A, B, C, and D are the same filters
- The neighboring circuits of A is more similar compared with B, than C
- Detect symmetry based on the “context” of the circuit system
Main Idea: Determine symmetry based on extracted subgraph similarity

• Q: Why extract subgraphs?
  • A: Include neighboring circuit and system “context” to resolve ambiguity

• Q: Why graph similarity?
  • A1: Graph isomorphism including neighboring circuits rare
  • A2: Graph similarity provides numeric values for comparisons

• Problem1: We need a scalable graph similarity measurement.
• Problem2: How large subgraphs to extract?
Graph similarity with spectral graph analysis

- Graph Laplacian matrix includes both degree and adjacency information.
- Its eigenvalues measure node cluster cohesiveness and have been used to approximate sparsest cuts and VLSI circuit partitions.
- We use Kolmogorov-Smirnov (K-S) statistics.

\[
D_n = \sup_x |F_{1,n}(x) - F_{2,m}(x)|
\]

- The p-value from the K-S test measures the eigenvalue distribution similarity, which we use as the quantitative measurement for graph similarity.
- The higher the p-value, the more similar the graphs.

Gera et al., “Identifying network structure similarity using spectral graph theory”, Applied Network Science, 2018
How large subgraphs to extract?

- Both too large and small subgraphs would result in over-constraints
- Too large: both subgraphs are the entire system graph and always be isomorphic
- Too small: does not include enough system context

The subgraph size need to consider

- The size of the subcircuits A, B
- The proximity of the subcircuits $dist(A, B)$
- Calculate $dist(A, B)$ with graph centers
Commonly used graph centrality measures

- Jordan Center:

\[
\min_{v} \max_{u \in V} d(v, u)
\]

- Eigenvector Centrality:

\[
\max_{v} \left( x_{v} = \sum_{u \in N(v)} x_{u} \right)
\]

- PageRank Center:

\[
\max_{v} \left( PR(v) = \vartheta \sum_{u \in N(v)} \frac{PR(u)}{\deg(u)} + \frac{1 - \vartheta}{|V|} \right)
\]

- We use the average of the three measures
Determining subgraph sizing:

- Radius of subgraph $= \frac{1}{2} \text{dist}(\text{graph}A, \text{graph}C)$
- Similarity of (A,C) is low for the proposed subgraph radius and successfully filtered this over-constraint, while a small and large subgraphs lead to over-constraint
Experimental Results

- Tested S$^3$DET on 3 ADC designs and compare with labels given by designers
  - ✓ 1000+ nodes
  - ✓ 4000+ edges

<table>
<thead>
<tr>
<th>Design</th>
<th>ADC Architecture</th>
<th>#Valid Pairs</th>
<th>#Nodes</th>
<th>#Edges</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>CT ΔΣ</td>
<td>190</td>
<td>1300</td>
<td>4158</td>
</tr>
<tr>
<td>II</td>
<td>SAR</td>
<td>776</td>
<td>2924</td>
<td>7427</td>
</tr>
<tr>
<td>III</td>
<td>CT ΔΣ SAR Hybrid</td>
<td>1229</td>
<td>4618</td>
<td>11674</td>
</tr>
</tbody>
</table>
Experimental Results

- Different graph centrality have different results
- Baseline is only matching cell topology
- Overall lower false alarms (less over-constraints) with comparable accuracy and precision

More than 10x reduction in over-constraints
Different graph centrality have different results

Baseline is only matching cell topology

Overall lower false alarms (less over-constraints) with comparable accuracy and precision

<table>
<thead>
<tr>
<th>Design</th>
<th>Jordan</th>
<th>Eigenvector</th>
<th>PageRank</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>20.0s</td>
<td>9.5s</td>
<td>10.7s</td>
</tr>
<tr>
<td>II</td>
<td>8m37.7s</td>
<td>4m13.6s</td>
<td>10m29.0s</td>
</tr>
<tr>
<td>III</td>
<td>13m52.8s</td>
<td>8m34.2s</td>
<td>13m13.9s</td>
</tr>
</tbody>
</table>
Conclusions and Future Work

Conclusions:

• S³DET: Method of detection system symmetry constraints
• Subgraph extraction with graph centrality
• Graph similarity with spectral graph analysis
• Effectively resolve constraint ambiguity and reduce false alarms

Future Work:

• Extend to array-like regularity constraints
• Fully automated layout generation for system level AMS designs
Thank You
Comparisons with Graph Edit Distance (GED)

- Continuously remove edges randomly from a graph
- Results of 50 simulations indicate strong correlations between GED and K-S p-value