Symmetry-Aware Placement with Transitive Closure Graphs for Analog Layout Design

L. Zhang (1), Y. Jiang (2), C.-J. R. Shi (3)

(1) ECE, Memorial University, St. John’s, Canada
(2) ECE, University of Nevada, Las Vegas, USA
(3) EE, University of Washington, Seattle, USA
Introduction

- Present design process of analog integrated circuits
  - Time-consuming hand-crafted layout
  - Rough estimation and complex constraints
- Analog device/cell placement is one of the most significant stages in the layout synthesis
- Placement strategies for analog integrated circuits
Review of Prior Work

- Topological representations for digital layout placement design
- Absolute-coordinate representation used in a few well-known analog layout automation systems
- O-tree and B*-tree for satisfying the analog symmetry constraints in the placement problem
- SP, TCG-S and TCG representations for analog placement problem
\[ \alpha_a^{-1} < \alpha_b^{-1} \text{ and } \beta_{b'}^{-1} < \beta_{a'}^{-1}, \forall (a, a') \in \Gamma, (b, b') \in \Gamma, a \neq b \]
TCG Representation

- Horizontal transitive closure graph $G_h$ and vertical transitive closure graph $G_v$
- A valid TCG has three properties
Conversion Between SP and TCG

- **Given a SP**, its corresponding TCG can be constructed in $O(n^2)$ time
- **Given a TCG**, its corresponding SP can be constructed in $O(n)$ time
  - vertices in $\alpha$-sequence are ordered incrementally according to the sum of in-degree in $G_h$ and out-degree in $G_v$ of each vertex
  - vertices in $\beta$-sequence are ordered incrementally according to the sum of in-degrees in both $G_h$ and $G_v$ of each vertex
For \((a, a') \in \Gamma\) and \((b, b') \in \Gamma\), a TCG \((G_h, G_v)\) representation is symmetric-feasible if both of the following conditions are satisfied:

- **in** \(G_h\): \(a \vdash b \Leftrightarrow a' \vdash b\)
- **in** \(G_v\): \(a \dashv b \Leftrightarrow b' \dashv a'\)

where \(\Leftrightarrow\) denotes that the two cases before and after this symbol cannot simultaneously appear in the same TCG.
Lemma: Any placement containing a symmetry group can be represented with a symmetric-feasible TCG

Four cases for two symmetric pairs/cells
- two symmetric pairs
- one symmetric-pair and one self-symmetric cell
- two self-symmetric cells
- two cells in one symmetric pair
Y-Dimensional Symmetric Packing

Begin
1. construct the topological order of the TCG;
2. calculate the vertical longest path of the TCG based on the topological order;
3. for (each cell $c_i$ in the topological order) {
   4. if ($c_i$ has a symmetric counterpart $c_j$) {
      5. if ($\Delta Y = c_i.y - c_j.y < 0$) {
         6. shift $c_i$ and its $G_v$ fan-out cells for $-\Delta Y$;
       } 
      7. else if ($\Delta Y = c_i.y - c_j.y > 0$) {
         8. shift $c_j$ and its $G_v$ fan-out cells for $\Delta Y$;
      } 
   } 
}
End
X-dimensional Symmetric Packing

Begin
1 construct the topological order of the TCG;
2 calculate the horizontal longest path of the TCG based on the topological order;
3 determine the symmetry axis, and label the symmetric cells, which are chosen to calculate the symmetry axis, as $c_i$ and $c_j$, $(i \leq j)$;
4 for (any unprocessed symmetric cell $c_s$, $i < s < j$) {
   5   if ($c_s$ has a symmetric counterpart $c_t$) {
      6      shift symmetric pair ($c_s$, $c_t$);
      7      mark $c_s$ and $c_t$ as processed; }
   8 for (any unprocessed symmetric cell $c_s$, $j < s \leq n$) {
      9      if ($c_s$ has a symmetric counterpart $c_t$) {
         10         shift symmetric pair ($c_s$, $c_t$); }
      11 else if ($c_s$ is a self-symmetric cell) {
         12         symmetrically shift $c_s$; }
      13      mark $c_s$ as processed; }
   14 for (any unprocessed symmetric cell $c_s$, $0 \leq s < i$) {
      15      if ($c_s$ has a symmetric counterpart $c_t$) {
         16         shift symmetric pair ($c_s$, $c_t$); }
      17 else if ($c_s$ is a self-symmetric cell) {
         18         symmetrically shift $c_s$; }
      19      mark $c_s$ as processed; }
End
Lemma: Given a symmetric-feasible TCG containing a symmetry group, one can build a placement satisfying the positioning and the symmetry constraints in $O(n^2)$ time.
Perturbation of Symmetric-Feasible TCG

Three operations for TCG perturbation
- vertex rotation
- symmetric-swap
- edge change

Lemma: TCG is still symmetric-feasible and valid under the vertex-rotation operation, and this operation takes $O(1)$ time

Lemma: Given a symmetric-feasible TCG, the resulting TCG after a symmetric-swap operation is still symmetric-feasible and valid, and it takes $O(1)$ time
randomChangeTcgEdges
(Input: a symmetric-feasible TCG, Output: TRUE for a successful random edge-change operation and FALSE for a failure random edge-change operation)

Begin

1. randomly choose one vertex \( a \);
2. if \((a \text{ is symmetric}) \) {
   3. obtain the slack range of vertex \( a \) in the \( \alpha- (\beta-) \) sequence and save them into Set-A (Set-B);
   4. in Set-A (Set-B), randomly pick up one vertex \( b \) (\( a \neq b \)) and only keep the vertices between \( a \) and \( b \) (including \( b \)) in the corresponding set;
   5. randomly choose to operate on Set-A or Set-B;
   6. for (each vertex \( c \) in Set-A (Set-B)) {
      7. move (move-reverse) the edge between \( a \) and \( c \);
      8. if (\( c \) is a symmetric vertex and the updated \( (a, c) \) or \( (a', c') \) violates symmetric-feasibility Eqs. (2) and (3)) {
         9. the random edge change fails and return FALSE;
      } }
} 
10. else {
   
   11. randomly pick up one vertex \( b \) (\( a \neq b \));
   12. obtain the vertices lying between \( a \) and \( b \) in the \( \alpha- (\beta-) \) sequence and save them into Set-A (Set-B);
   13. for (each vertex \( c \) in Set-A (Set-B)) {
      14. move (move-reverse) the edge between \( a \) and \( c \);
   } 
   15. the random edge change is successful and return TRUE;

End
Lemma: Given a symmetric-feasible TCG, the perturbed TCG is still symmetric-feasible and valid under the certain edge change operation, and such an operation takes $O(n)$ time.

Theorem: The solution space of symmetric-feasible TCG can be fully explored using random vertex rotation, symmetric-swap, edge change operations. The transformation of two neighboring solutions represented in TCG takes at most $O(n)$ time.
Coded in C++ and compared with other two approaches

- *AbsPlace*, one absolute placement scheme using absolute coordinates
- *SymmSP*, an implementation using SP symmetry-feasibility conditions

Simulated-annealing based symmetry-aware TCG placement algorithm

Dedicated cost function

\[ C = \alpha_{\text{area}} C_{\text{area}} + \alpha_{\text{nets}} C_{\text{nets}} + \alpha_{\text{size}} C_{\text{size}} \]
## Experimental Results (II)

<table>
<thead>
<tr>
<th>Analog Circuits</th>
<th>AbsPlace</th>
<th>SymmSP</th>
<th>SymmTCG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rail-to-rail Opamp</td>
<td>Cost</td>
<td>119.6%</td>
<td>106.2%</td>
</tr>
<tr>
<td></td>
<td>T (sec)</td>
<td>134.2%</td>
<td>94.5%</td>
</tr>
<tr>
<td>Comm.-mode-feed. Opamp</td>
<td>Cost</td>
<td>126.6%</td>
<td>109.2%</td>
</tr>
<tr>
<td></td>
<td>T (sec)</td>
<td>136.3%</td>
<td>102.1%</td>
</tr>
<tr>
<td>Low-noise opamp</td>
<td>Cost</td>
<td>126.3%</td>
<td>105.7%</td>
</tr>
<tr>
<td></td>
<td>T (sec)</td>
<td>139.5%</td>
<td>104.2%</td>
</tr>
<tr>
<td>Comparator</td>
<td>Cost</td>
<td>140.1%</td>
<td>121.6%</td>
</tr>
<tr>
<td></td>
<td>T (sec)</td>
<td>152.5%</td>
<td>98.2%</td>
</tr>
</tbody>
</table>
## Comparison of Approaches

<table>
<thead>
<tr>
<th>Approaches</th>
<th>Packing</th>
<th>Perturbation</th>
<th>Completeness</th>
</tr>
</thead>
<tbody>
<tr>
<td>SP</td>
<td>$O(n\log \log n)$</td>
<td>$O(1)$</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>$O(n^2)$</td>
<td>$O(1)$</td>
<td>No</td>
</tr>
<tr>
<td>TCG-S</td>
<td>$O(n\log n)$</td>
<td>$O(n)$</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>$O(n^2)$</td>
<td>$O(n^2)$</td>
<td>No</td>
</tr>
<tr>
<td>TCG</td>
<td>$O(n^2)$</td>
<td>$O(n^2)$</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>$O(n^2)$</td>
<td>$O(n)$</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Symm. (this work)
CMOS Analog Comparator
Conclusions

- A set of TCG symmetric-feasible conditions is proposed
- Solution space of symmetric placements can be efficiently explored by evaluating symmetric-feasible TCGs
- Efficient strategy for generating random symmetric-feasible TCG representations while keeping TCG valid in $O(n)$ time
- A new packing scheme for symmetric-feasible TCG in a simulated-annealing based placement algorithm
Symmetry-Aware Placement with Transitive Closure Graphs for Analog Layout Design

L. Zhang (1), Y. Jiang (2), C.-J. R. Shi (3)

(1) ECE, Memorial University, St. John’s, Canada
(2) ECE, University of Nevada, Las Vegas, USA
(3) EE, University of Washington, Seattle, USA