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

Contents

- Introduction
- Review of prior work
- Symmetric-Feasible TCG
- Packing for Symmetric Placement
- Perturbation of Symmetric-Feasible TCG
- Experimental results
- Conclusions

Introduction

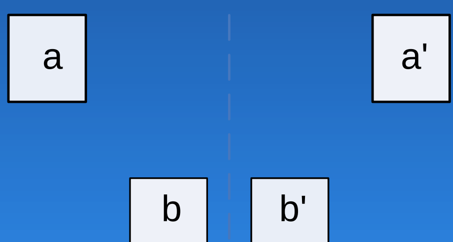
- Present design process of analog integrated circuits
  - Time-consuming hand-crafted layout
  - Rough estimation and complex constraints
- Analog device/cell placement is one of the most significant stages in the layout synthesis
- Placement strategies for analog integrated circuits

Review of Prior Work

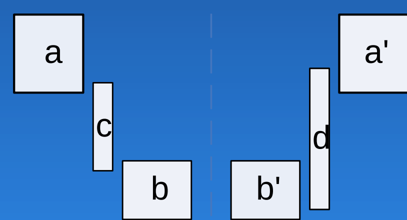
- Topological representations for digital layout placement design
- Absolute-coordinate representation used in a few well-known analog layout automation systems
- O-tree and B*-tree for satisfying the analog symmetry constraints in the placement problem
- SP, TCG-S and TCG representations for analog placement problem

SP Symmetric-Feasibility Conditions

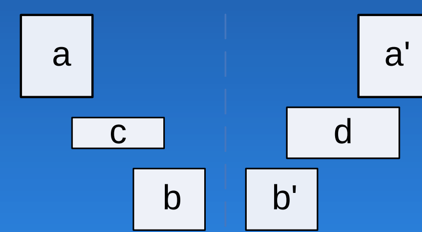
$$\alpha_a^{-1} < \alpha_b^{-1} \text{ and } \beta_{b'}^{-1} < \beta_{a'}^{-1}, \forall (a,a') \in \Gamma, (b,b') \in \Gamma, a \neq b$$



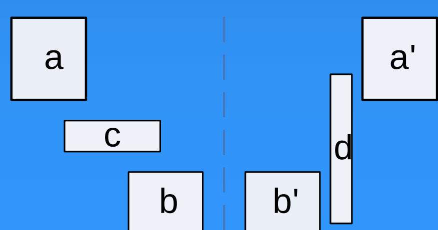
a $\vdash$ *b, b'* $\vdash$ *a'*



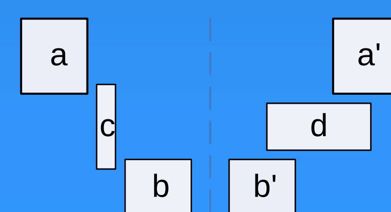
a $\vdash$ *b, b'* $\vdash$ *a'*



b $\perp$ *a, b'* $\perp$ *a'*



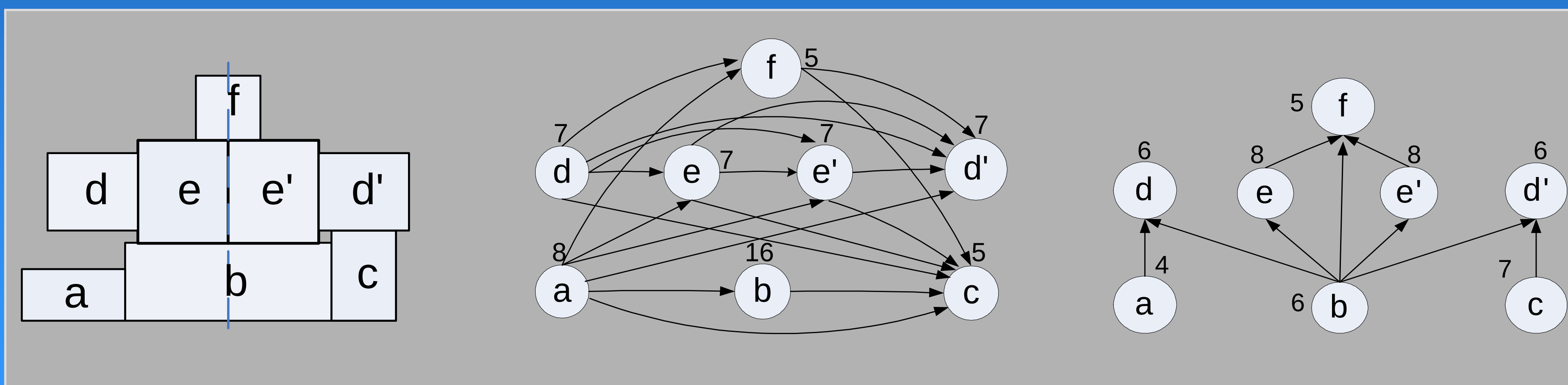
b $\vdash$ *a, b'* $\vdash$ *a'*



a $\vdash$ *b, b'* $\perp$ *a'*

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 α-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 β-sequence are ordered incrementally according to the sum of in-degrees in both G_h and G_v of each vertex

TCG Symmetric-Feasibility Conditions

For $(a, a') \in \Gamma$ and $(b, b') \in \Gamma$, a TCG (G_h and G_v) representation is symmetric-feasible if both of the following conditions are satisfied:

in G_h: $a \vdash b \;\langle\neq\rangle\; a' \vdash b$

in G_v: $a \perp b \;\langle\neq\rangle\; b' \perp a'$

where $\langle\neq\rangle$ denotes that the two cases before and after this symbol cannot simultaneously appear in the same TCG

Symmetric-Feasible 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

Y-Dimensional Symmetric Packing

Begin

construct the topological order of the TCG;

calculate the vertical longest path of the TCG based on the topological order;

for (each cell c_i in the topological order) {

    if (c_i has a symmetric counterpart c_j) {

        if ($\Delta Y = c_i.y - c_j.y < 0$) {

          shift c_i and its G_v fan-out cells for $-\Delta Y$; }

        else if ($\Delta Y = c_i.y - c_j.y > 0$) {

          shift c_j and its G_v fan-out cells for ΔY; }}}

End

X-dimensional Symmetric Packing

X-Dimensional Symmetric Packing

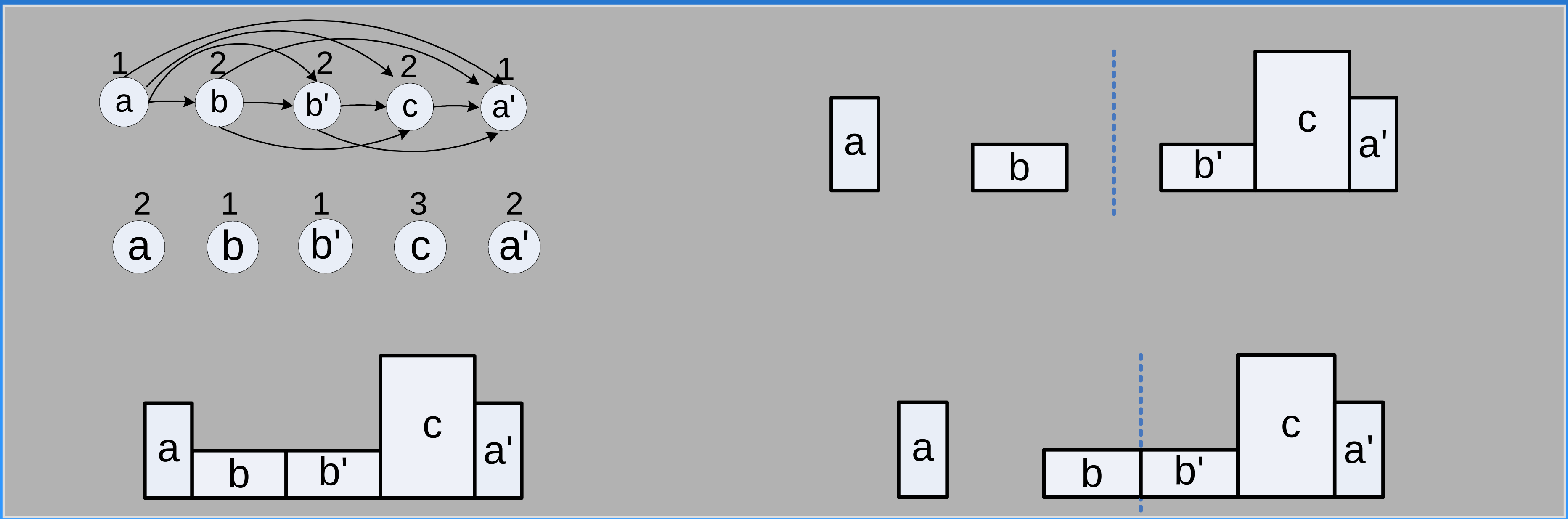
Begin

construct the topological order of the TCG;
calculate the horizontal longest path of the TCG based on the topological order;
determine the symmetry axis, and label the symmetric cells, which are chosen to calculate the symmetry axis, as c_i and c_j, ($i \leq j$);
for (any unprocessed symmetric cell c_s, $i<s<j$) {
    if (c_s has a symmetric counterpart c_t) {
        shift symmetric pair (c_s, c_t);
        mark c_s and c_t as processed; }}
for (any unprocessed symmetric cell c_s, $j<s \leq n$) {
    if (c_s has a symmetric counterpart c_t) {
        shift symmetric pair (c_s, c_t); }
    else if (c_s is a self-symmetric cell) {
        symmetrically shift c_s; }
    mark c_s(c_t) as processed; }
for (any unprocessed symmetric cell c_s, $0 \leq s<i$) {
    if (c_s has a symmetric counterpart c_t) {
        shift symmetric pair (c_s, c_t); }
    else if (c_s is a self-symmetric cell) {
        symmetrically shift c_s; }
    mark c_s (c_t) as processed; }

End

Symmetric-Feasible TCG and Symmetric Placement

- **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

Edge Change Operation

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

randomly choose one vertex *a*;
if (*a* is symmetric) {
  obtain the slack range of vertex *a* in the α- (β-) sequence and save them into Set-A (Set-B);
  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;
  randomly choose to operate on Set-A or Set-B;
  for (each vertex *c* in Set-A (Set-B)) {
    move (move-reverse) the edge between *a* and *c*;
    if (*c* is a symmetric vertex and the updated (*a*, *c*) or (a', c') violates symmetric-feasibility Eqs. (2) and (3)) {
      the random edge change fails and return FALSE;}}}
else {
  randomly pick up one vertex *b* ($a \neq b$);
  obtain the vertices lying between *a* and *b* in the α- (β-) sequence and save them into Set-A (Set-B);
  for (each vertex *c* in Set-A (Set-B)) {
      move (move-reverse) the edge between *a* and *c*;}}
the random edge change is successful and return TRUE;

End

Perturbation of Symmetric-Feasible TCG

- **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

Experimental Results (I)

- 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

$$C = \alpha_{area} C_{area} + \alpha_{nets} C_{nets} + \alpha_{size} C_{size}$$

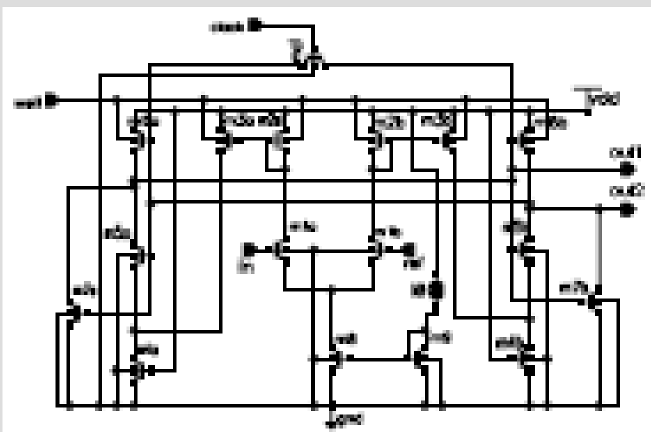
Experimental Results (II)

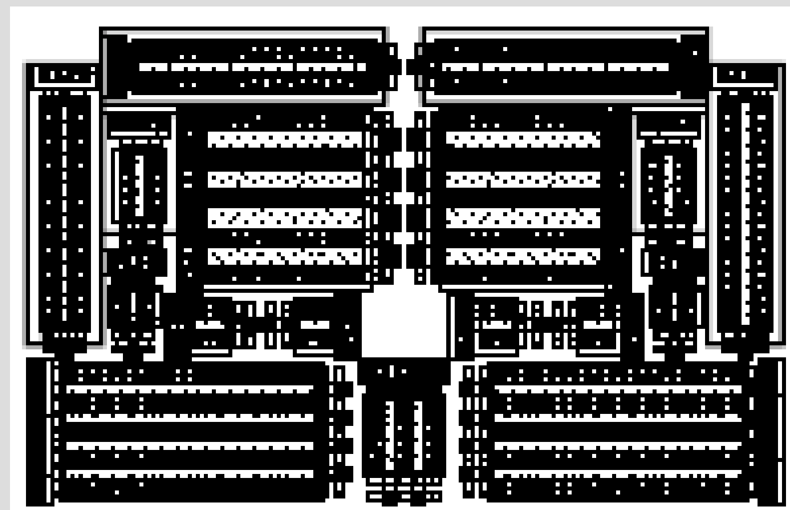
Analog Circuits		*AbsPlace*	*SymmSP*	*SymmTCG*
Rail-to-rail Opamp	Cost	119.6%	106.2%	142643
	T (sec)	134.2%	94.5%	48
Comm.-mode-feed. Opamp	Cost	126.6%	109.2%	306512
	T (sec)	136.3%	102.1%	72
Low-noise opamp	Cost	126.3%	105.7%	286559
	T (sec)	139.5%	104.2%	84
Comparator	Cost	140.1%	121.6%	19720
	T (sec)	152.5%	98.2%	118

Comparison of Approaches

Approaches		*Packing*	*Perturbation*	*Completeness*
SP	General	$O(nlglgn)$	$O(1)$	*Yes*
	Symm.	$O(n^2)$	$O(1)$	*No*
TCG-S	General	$O(nlgn)$	$O(n)$	*Yes*
	Symm.	$O(n^2)$	$O(n^2)$	*No*
TCG	General	$O(n^2)$	$O(n^2)$	*Yes*
	Symm. (this work)	$O(n^2)$	$O(n)$	*Yes*

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