DESIGN AUTOMATION

5MD20

placement formulations

Ralph Otten
otten@ics.ele.tue.nl
February-March 2010
the placement problem

• given:
  - an area model
  - a set of modules with shapes
  - a net list

• constraint:
  - produce a non-overlapping placement

• objectives:
  - minimize dead space (minimize total area)
  - minimize total wire length
  - minimize congestion problems
  - fulfill timing constraints
placement: types

- standard cell:
  - uniform block height
  - (almost) uniform block size.
  - fairly good results

- block-level placement:
  - non-uniform block sizes
  - automatic algorithms perform poorly
how bad can a placement be?

- worst case: all wires cross a bisection

**BAD:**
- $n$ by $n$ cells:
  - with $n^2$ crossing boundary

**GOOD:**
- $O(n)$ wires crossing boundary
other bad things: delay

- best case: depends on gate delays
- worst case delay counts!
- delay is a path issue!
  - often between registers
  - worst when critical path crosses the entire chip
    (sometimes, it is even longer):
    worst case:
    proportional to \((2*\text{l}_{\text{side}})^2\)
- this is can be huge!, yet difficult to capture.
- cannot place ‘everything close’

\[
delay_{\text{wire}} = 0.5 * r_{\text{sq}} * c_0 * l^2
\]
bad: ‘dead space’

• the aim is to pack the blocks as tightly as possible together.
what is placement?

- manipulation of fixed (rectangular) objects in the plane
- objects should be non-overlapping in the final arrangement
- each object has fixed pin positions (maybe center)
- some objects may have (partly) pre-assigned position
- net list is available
- each (placement) configuration has a cost
- goal is to find a configuration with small cost
wire length estimates

semi-perimeter length = 10
wire length estimates

complete graph length \* \frac{2}{n} = 16
wire length estimates

chain length = 13
wire length estimates

source to sink length = 16
wire length estimates

steiner tree length = 11
wire length estimates

spanning tree length = 12
wire length cost functions

- direct
  - individual net usually estimated as half the perimeter of the minimal box enclosing all pins of that net
  - exact for nets with two or three pins (the majority)
  - reasonable for steiner trees for a few pins
  - nets with a large number of pins can be treated specially
  - traditionally total wire length (sum of the estimates) is used as a cost function
    - beneficial for area and somewhat for power
    - doubtful whether good for highest performance (weights do not reflect this properly)

- indirect
  - a structured way of controlling wires: mincut
  - often alternating "horizontal" and "vertical" cut lines
  - also $2 \times 2$ or $3 \times 3$ blocks
placement methods

- constructive placement
  - force directed
  - min cut
  - network analogons, eigenvalues
- iterative changes
  - pairwise interchange
  - force directed
  - stochastic optimization
- slot assignment
  - probing
  - matching
- linear placements
components of rectangle dissections

- (rectangular) interior
  - rectangles
  - T-junctions
  - horizontal line segments
  - vertical line segments

- boundary
  - top, bottom, left and right sides

- exterior

remark: line segments can be interpreted as maximal line segments or as parts between T-junctions
rectangle dissections immediately induce graphs represented by its line segments (the edges) and the \textit{T-junctions} (the vertices)
first observations

- A graph is a set of vertices and a binary relation over that set.
  - It does not include a representation, even if planar.
- Floorplan graphs and dissections are not in bijective relation.
  - Different dissections may have the same floorplan graph.
colored floorplan graph and its cube extension
first observations

- a graph is a set of vertices and a binary relation over that set
  - it does not include a representation, even if planar
- floorplan graphs and dissections are not in bijective relation
  - different floorplans may have the same floorplan graph

- with feasible edge-set bipartitioning (colored floorplan graph)
  the relation is (almost) 1-1
  - some symmetries are still there

- from rectangle dissection to floorplan graph is easy
  - in design the dissection is initially not available

- the question is how to generate a floorplan graph and how to obtain a feasible bipartition

- what do we have to start with?

  size estimates and proximity (adjacency) preferences
we want a graph that captures the adjacencies between the rectangles of a dissection
The Grason graph of a rectangle dissection is the dual of the floorplan graph of that dissection.
Extended Grason graph

The Grason graph of a rectangle dissection is the dual of the floorplan graph of that dissection if fully extended.

If fully extended, it is the dual of the floorplan graph and its cube extension.
extended grason digraph

of course, also grason graphs do not have unique dissections

we have to know the partition in horizontal and vertical adjacencies

- rectangles
- T-junctions
- extensions
of course, also grason graphs do not have unique dissections

we have to know the partition in horizontal and vertical adjacencies
grason h-digraph with top and bottom vertex
extended grason digraph
grason v-digraph with left and right vertex
size implications

dissections constrain the dimension of its rectangles:

1. \textbf{widths} of rectangles above a horizontal line segment should \textbf{sum up} to the same total as the rectangle widths below that line segment

2. \textbf{heights} of rectangles between the same pair of horizontal line segments must \textbf{sum up} to the same total

\[ h_i = h_j + h_k + h_g \]

\textbf{how do we capture these constraints systematically?}
polar digraph formation

- put a vertex on every corner
- put a diagonal arrow in every rectangle
- delete the vertical line segments
polar digraph formation

- put a vertex on every corner
- put a diagonal arrow in every rectangle
- delete the vertical line segments
- contract the horizontal of line segments
polar h-digraph

- put a vertex on every corner
- put a diagonal arrow in every rectangle

- delete the vertical line segments
- contract the horizontal of line segments
Kirchhoff equations

\[ w_i + w_j = w_p + w_q \]

\[ h_i = h_j + h_g + h_k \]
size implications

\[ w_i + w_j = w_p + w_q \]

\[ h_i = h_j + h_g + h_k \]
Kirchhoff equations

\[ w_i + w_j = w_p + w_q \]

Kirchhoff "current" equation

\[ h_i = h_j + h_g + h_k \]

Kirchhoff "voltage" equation
introducing sizes and shapes

- a way of representing the feasible shapes is required
  - only positive dimensions
  - any width or height above a given minimum is allowed
  - for every allowed width there is a minimum value above which any height is feasible
  - increasing one dimension can never force the other dimension to increase as well

**ANSWER SHAPE CONSTRAINTS**
shape constraint
shape constraint
shape constraint
shape constraints

fixed shape

flexible block

channel

choice of blocks
bounding functions

1. support $F \subseteq \mathbb{R}_0^+$, connected, and $\infty \in \text{support } F$
2. right continuous
3. monotonously not increasing
(4. Piecewise Linear)

shape constraints:
$A \in \text{support } F$
$B \geq F(A)$

shaded area is "bounded region"
introducing sizes and shapes

- a way of representing the feasible shapes is required
  - only positive dimensions
  - any width or height above a given minimum is allowed
  - for every allowed width there is a minimum value above which any height is feasible
  - increasing one dimension can never force the other dimension to increase as well

- a way of guaranteeing the topology as fixed in the floorplan
  - relative positions should be realized
  - the dissection must fit

- an objective function for obtaining the best contour

represent them by shape constraints (bounding functions) and kirchhoff relations
mathematical program

- vertical constraints (from polar v-digraph)
- horizontal constraints (from polar h-digraph)
- shape constraints

1. support $F \subseteq \mathbb{R}_0^+$, connected, and $\infty \in \text{support } F$
2. right continuous
3. monotonously not increasing

(4. Piecewise Linear)

shape constraints:
- $A \in \text{support } F$
- $B \geq F(A)$

minimize: $f(h_0, w_0)$
Floorplan design

- Assessed environment
- Adjacencies (from proximities, net list, ...)
  - Planarization
  - Plane triangulation (no complex triangles)
  - Corner assignment
  - Dualization (polar graphs)
- Shape constraints
- Optimization
  - Rectangle dissection
- Floorplan
mathematical program

- vertical constraints (from polar v-digraph)
- horizontal constraints (from polar h-digraph)
- shape constraints

1. support $F \subseteq \mathbb{R}_0^+$, connected, and $\infty \in \text{support } F$
2. right continuous
3. monotonously not increasing
(4. Piecewise Linear)

shape constraints:
$A \in \text{support } F$
$B \geq F(A)$

minimize: $f(h_0, w_0)$
quadratic programming

- vertical constraints (from polar v-digraph)
- horizontal constraints (from polar h-digraph)
- shape constraints defining convex bounded regions

\[
\begin{bmatrix}
A \\
\end{bmatrix}
\begin{bmatrix}
w \\
h \\
\end{bmatrix} \geq c
\]

\[
A \geq 0
\]

\[
c \geq 0
\]

- minimize: $h_0 \ w_0$

Ohtsuki: 1970
quadratic programming

- vertical constraints (from polar v-digraph)
- horizontal constraints (from polar h-digraph)
- shape constraints defining convex bounded regions

\[
\begin{align*}
A \begin{pmatrix} w \\ h \end{pmatrix} & \geq c \\
A & \geq 0 \\
c & \geq 0
\end{align*}
\]

- minimize: 
  \[ h_0 \ w_0 \]

Ohtsuki: 1970
linear programming

- vertical constraints (from polar v-digraph)
- horizontal constraints (from polar h-digraph)
- shape constraints defining convex bounded regions

minimize: $a h_0 + b w_0$

$A \begin{pmatrix} w \\ h \end{pmatrix} \geq c$

$A \geq 0$

$c \geq 0$

Otten: 1975
mixed 0-1 linear programming

- vertical constraints (from polar v-digraph)
- horizontal constraints (from polar h-digraph)
- shape constraints  (piecewise linear)

\[ A \begin{pmatrix} w \\ h \end{pmatrix} \geq c_1 v + c_2 (1 - v) \]
\[ v \in \{0,1\} \]
\[ A \geq 0 \]
\[ c \geq 0 \]

\[ \text{minimize: } a h_0 + b w_0 \]

Zibert, Saal: 1974
floorplan optimization

- vertical constraints (from polar v-digraph)
- horizontal constraints (from polar h-digraph)
- shape constraints

floorplan optimization is strictly np-hard

L. Stockmeyer, 1983

bounding function \( F \)

minimize: \( f(h_0, w_0) \)
placement methods

- constructive placement
  - force directed
  - min cut
    - network analogons, eigenvalues
- iterative changes
  - pairwise interchange
  - force directed
  - stochastic optimization
- slot assignment
  - probing
  - matching
- linear placements
oriented min-cut

recursively find a suitable bipartitioning
mostly using a move-based partitioning method and area balancing

rectangle dissection polar h-digraph (and polar v-digraph)
oriented min-cut

recursively find a suitable bipartitioning
(mostly using a move-based partitioning method and area balancing)

rectangle dissection

polar h-digraph (and
polar v-digraph)
oriented min-cut

recursively find a suitable bipartitioning
(mostly using a move-based partitioning method and area balancing)

rectangle dissection

polar h-digraph (and polar v-digraph)
oriented min-cut

recursively find a suitable bipartitioning
(mostly using a move-based partitioning method and area balancing)

rectangle dissection

polar h-digraph (and polar v-digraph)

Lauther: 1979
oriented mincut

after the partitioning stage improvement operations are applied (rotation, reflection, squeezing)

squeezing example

Lauther: 1979
oriented mincut

after the partitioning stage improvement operations are applied (rotation, reflection, squeezing)

squeezing example

Lauther: 1979
problem of recursive partitioning

cost of these 2 partitionings are not the same.
terminal propagation

- need to consider nets connecting to external terminals or other modules as well.
- do partitioning in a breath-first manner (i.e., finish all higher-level partitioning first).

the dummy terminal will try to pull B to the top partition.
mincut placement

- perform quadrature mincut onto $4 \times 4$ grid
  - start with vertical cut first

\[
\begin{align*}
  n_1 &= \{e, f\} \\
  n_2 &= \{a, e, i\} \\
  n_3 &= \{b, f, g\} \\
  n_4 &= \{c, g, l\} \\
  n_5 &= \{d, l, h\} \\
  n_6 &= \{e, i, j\} \\
  n_7 &= \{f, j\} \\
  n_8 &= \{g, j, k\} \\
  n_9 &= \{l, o, p\} \\
  n_{10} &= \{h, p\} \\
  n_{11} &= \{i, m\} \\
  n_{12} &= \{j, m, n\} \\
  n_{13} &= \{k, n, o\}
\end{align*}
\]

undirected graph model with k-clique weighting
thin edges = weight 0.5, thick edges = weight 1
bisection

- first cut has min-cutsize of 3 (not unique)
  - both cuts 1 and 2 divide the entire chip
bisection

- first cut has min-cut size of 3 (not unique)
  - both cuts 1 and 2 divide the entire chip
- each cut minimizes cut size
  - helps reduce overall wirelength
bisection

• First cut has min-cutsize of 3 (not unique)
  - Both cuts 1 and 2 divide the entire chip
• Each cut minimizes cutsize
  - Helps reduce overall wirelength
bisection

• first cut has min-cutsize of 3 (not unique)
  - both cuts 1 and 2 divide the entire chip
• each cut minimizes cutsize
  - helps reduce overall wirelength
bisection

- first cut has min-cutsize of 3 (not unique)
  - both cuts 1 and 2 divide the entire chip
- each cut minimizes cutsize
  - helps reduce overall wirelength
- 16 partitions generated by 6 cuts
  - HPBB wirelength = 27
min-cut placement, terminal propagation

• start with vertical cut
min-cut placement, terminal propagation

- two terminals are propagated and are “pulling” nodes
  - node $k$ and $o$ connect to $n$ and $j$: $p_1$ propagated
    (outside window)
  - node $g$ connect to $j$, $f$ and $b$: $p_2$ propagated
    (outside window)
  - terminal $p_1$ pulls $k, o$ and $g$ to top partition,
    and $p_2$ pulls $g$ to bottom
**min-cut placement, terminal propagation**

- two terminals are propagated and are “pulling” nodes
  - node $k$ and $o$ connect to $n$ and $j$: $p_1$ propagated
    (outside window)
  - node $g$ connect to $j$, $f$ and $b$: $p_2$ propagated
    (outside window)
  - terminal $p_1$ pulls $k$, $o$ and $g$ to top partition,
    and $p_2$ pulls $g$ to bottom
min-cut placement, terminal propagation

- one terminal propagated
  - node $n$ and $j$ connect to $o, k$ and $g$: $p_1$ propagated
  - node $i$ and $j$ connect to $e, f$ and $a$: no propagation
  - terminal $p_1$ pulls $n$ and $j$ to right partition
min-cut placement , terminal propagation

- three terminals propagated
  - node $i$ propagated to $p_1$, $j$ to $p_2$, and $g$ to $p_3$
  - terminal $p_1$ pulls $e$ and $a$ to left partition
  - terminal $p_2$ and $p_3$ pull $f$, $b$ and $e$ to right partition
min-cut placement, terminal propagation

- one terminal propagated
  - node $n$ and $j$ are propagated to $p_1$
  - terminal $p_1$ pulls $o$ and $k$ to left partition
min-cut placement, terminal propagation

- three terminals propagated
  - node $j/f/b$ propagated to $p_1$, $o$ and $k$ to $p_2$, and $h$ and $p$ to $p_3$
  - terminal $p_1$ and $p_2$ pull $g$ and $l$ to left partition
  - terminal $p_3$ pull $l$ and $d$ to right partition
min-cut placement, terminal propagation

- 16 partitions generated by 15 cuts
  - HPBB wirelength = 23
comparison

• quadrature vs recursive bisection + terminal propagation
  - number of cuts: 6 vs 15
  - wirelength: 27 vs 23
placement methods

- constructive placement
  - force directed
  - min cut
  - network analogons, eigenvalues
- iterative changes
  - pairwise interchange
  - force directed
  - stochastic optimization
- slot assignment
  - probing
  - matching
- linear placements
accepting the euclidean metric enables eigenvalue methods
for example

- mapping a distance space into the plane
- minimizing the energy in the springs

\[
\text{minimize } \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} \left[ (x_i - x_j)^2 + (y_i - y_j)^2 \right]
\]

which can be written as:

\[
\text{minimize } x^T B x + y^T B y \quad \text{where } B = D - C \quad \text{with } d_{ii} = \sum_{j=1}^{n} c_{ij}
\]

constraints: \( x^T y = 0; \ x^T x = 1; \ y^T y = 1 \)

solution will be the two eigenvectors associated second and third smallest eigenvalues
quadratic placement formulation

- assume that cells are points with a single location \((x, y)\)
- all nets are two-terminals (so we need to do the clique-trick)
- then, let's minimize the wire length.
- well, let's minimize the sum of the quadratic wire length:

\[
\text{minimize } \sum_{\text{nets}_{i,j}} w_{ij} \left((x_i - x_j)^2 + (y_i - y_j)^2\right)
\]

- \(w_{i,j} = 0\) if there is no net between the cells \(i\) and \(j\)
- why quadratic?
  - simply we cause we can get an analytic solution
  - write equations and solve it numerically for the best minimum