analysis and synthesis of electronic circuits

5HH10 course 2006
www.es.ele.tue.nl/education/
Ralph Otten
otten@ics.ele.tue.nl
lecture schedules

tuesdays, 13:30:
• 28 maart*: electronics (5HH00)
•  4 april*: automation (5HH10)
• 11 april: automation (5HH10)
• 18 april: automation (5HH10)
• 25 april: automation (5HH10)
•   2 mei: electronics (5HH00)
•  16 mei*: automation (5HH10)
•  23 mei: automation (5HH10)
•  30 mei: automation (5HH10)
•  6 juni: automation (5HH10)
• 13 juni: electronics (5HH00)
• 20 juni: to be determined

thursdays, 13:30:
•  30 maart: automation (5HH10)
•  6 april:  electronics (5HH00)
• 13 april: electronics (5HH00)
• 20 april: PSpice (2 hours)
• 27 april: dies natalis
•   4 mei:  no lecture
• 18 mei:  electronics (5HH00)
• 25 mei:  ascension
•  1 juni:  electronics (5HH00)
•  8 juni:  electronics (5HH00)
• 15 juni: electronics (5HH00) (2hours)

* : college begint om 14:30

tentamens:
•  23 juni: electronics (5HH00)
•  30 juni: automation (5HH10)
floorplan design for a module

sub module data
- shape constraints
  (at least area estimates)
- pin positions
- module delay
  . . . . . .

assessed environment
- shape estimate
- pin position estimates
- signal timing

proximities
- connectedness
  (e.g. net list)
- timing constraints
  . . . . . .

libraries
- technology
- function
- performance

relative positions
(= floorplan)

dissection
global wiring
timing analysis
(re-) synthesis

assessed submodule environments
floorplan design for a hierarchy

chip footprint → functional hierarchy → net list

libraries

libraries

libraries

placement and routing
floorplanning calls

- accept the functional hierarchy and keep it recoverable
- determine preliminary environments for lower level modules
- start geometrical layout design of a cell after all its ancestors have a floorplan (including preliminary shapes)

```plaintext
module queue := < root > ;
while module queue is not empty do
    dequeue M from module queue;
    if M is a compound
        then FLOORPLAN ( M )
        else ASSEMBLE ( M ) end;
    enqueue submodules of M to module queue
end
```
what is a floorplan?

- a floorplan is a data structure that contains the relative positions of modules in a supermodule

possible representations:
1. rectangle dissection
2. point configurations
3. graphs
4. trees

graphs for:
1. abutments
2. adjacencies
3. numerical relations
4. packing rules
a zoo of graphs

rectangle dissections

topology: adjacencies

channel incidences

size implications
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 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
grason graph

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 it is the dual of the floorplan graph and its cube extension

- rectangles
- T-junctions
- extensions
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
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
- horizontal
- vertical
- extensions
grason h-digraph with top and bottom vertex
extended grason digraph
grason v-digraph with left and right vertex
a zoo of graphs

rectangle dissections

- topology: adjacencies
- channel incidences
- size implications

(co)loured
floorplan graphs

grason
(di)graphs
size implications

dissections constrain the dimension of its rectangles:

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

\[ w_i + w_j = w_p + w_q \]
size implications

dissections constrain the dimension of its rectangles:

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

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

\[ h_i = h_j + h_k + h_g \]

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

\[ w_i + w_j = w_p + w_q \]

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

\[ w_i + w_j = w_p + w_q \]

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

\[ w_i + w_j = w_p + w_q \]

kirchhoff "current" equation

\[ h_i = h_j + h_g + h_k \]

kirchhoff "voltage" equation
a zoo of graphs

rectangle dissections

topology: adjacencies

(colored) floorplan graphs

gerson (di)graphs

channel incidences

channel digraphs

size implications

polar digraphs

kirchhoff equations
channel ordering

channel routing problem

- given the longitudinal pin position
- given the nets exiting at the channel ends
- construct the detailed wire geometry for that channel

channels come together at T-junctions

- to route a channel the longitudinal positions must be known
- therefore the "base" of the "T" has to be routed before the "bar"
channel digraph formation

- put an arrowhead at every T-junction towards the crossbar
- put a vertex on every line segment
- contract every (undirected) edge
channel digraph formation

- put an arrowhead at every T-junction towards the crossbar
- put a vertex on every line segment
- contract every (undirected) edge

U. Flemming, 1978
channel digraph

- $\gamma^+ = 2$, except for two outer channels ($\gamma^+ = 0$)
- bipartite (horizontal and vertical partition can be derived)
- T-junction sequence on one side preserved in representation

U. Flemming, 1978
channel digraph properties

\( \gamma^+ = 2 \), except for two outer channels \( \gamma^+ = 0 \)

- bipartite (horizontal and vertical partition can be derived)
- T-junction sequence on one side preserved
- \#rectangles = \(| V | - 3 = \#\text{channels}-3 = \# | A | 2 = | F |\)
- acyclic fif there is no cycle of length 4
relation table

<table>
<thead>
<tr>
<th></th>
<th>grason h-digraph ((R, HLS, \ldots))</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>grason v-digraph ((R, VLS, \ldots))</td>
<td></td>
</tr>
<tr>
<td></td>
<td>grason digraph ((R, LS, TJ))</td>
<td></td>
</tr>
<tr>
<td></td>
<td>colored floorplan graph ((TJ, LS, R))</td>
<td></td>
</tr>
<tr>
<td></td>
<td>floorplan graph</td>
<td></td>
</tr>
<tr>
<td></td>
<td>colored grason graph ((R, LS, TJ))</td>
<td></td>
</tr>
<tr>
<td></td>
<td>grason graph</td>
<td></td>
</tr>
<tr>
<td>Channel digraph ( (C, TJ, R) )</td>
<td>( \uparrow )</td>
<td>( \downarrow )</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>----------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>Colored floorplan graph ( (TJ, LS, R) )</td>
<td>( \uparrow )</td>
<td>( \downarrow )</td>
</tr>
<tr>
<td>Floorplan graph ( (TJ, LS, R) )</td>
<td>( \uparrow )</td>
<td>( \downarrow )</td>
</tr>
<tr>
<td>Grason digraph ( (R, LS, TJ) )</td>
<td>( \uparrow )</td>
<td>( \downarrow )</td>
</tr>
<tr>
<td>Grason graph ( (R, LS, TJ) )</td>
<td>( \uparrow )</td>
<td>( \downarrow )</td>
</tr>
<tr>
<td>Grason graph ( (R, LS, TJ) )</td>
<td>( \uparrow )</td>
<td>( \downarrow )</td>
</tr>
</tbody>
</table>
relation table

<table>
<thead>
<tr>
<th>polar v-digraph ( (\text{VC}, R, HC) )</th>
<th>( \uparrow )</th>
<th>( \cup )</th>
<th>grason h-digraph ( (R, HLS, ..) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>channel digraph ( (C, TJ, R) )</td>
<td>( \downarrow )</td>
<td></td>
<td>grason v-digraph ( (R, VLS, ..) )</td>
</tr>
<tr>
<td>colored floorplan graph ( (TJ, LS, R) )</td>
<td>( \leftrightarrow )</td>
<td></td>
<td>grason digraph ( (R, LS, TJ) )</td>
</tr>
<tr>
<td>floorplan graph</td>
<td>D</td>
<td></td>
<td>colored grason graph ( (R, LS, TJ) )</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>grason graph</td>
</tr>
</tbody>
</table>
## Relation Table

| polar h-digraph $(HC, R, VC)$ |  | grason h-digraph $(R, HLS, ..)$ |
|------------------------------|  |------------------------------|
| $D$                           |  | $U$                          |
| polar v-digraph $(VC, R, HC)$ |  | grason v-digraph $(R, VLS, ..)$ |
| channel digraph $(C, TJ, R)$  |  | grason digraph $(R, LS, TJ)$  |
| colored floorplan graph $(TJ, LS, R)$ |  | colored grason graph $(R, LS, TJ)$ |
| floorplan graph               |  | grason graph                 |

The table illustrates the relationships between various digraphs and graphs in a network structure.
polar digraphs and grason digraphs

1. add a top and bottom arc
2. form the line graph
<table>
<thead>
<tr>
<th>polar h-digraph (HC, R, VC)</th>
<th>channel h-incidences (R, HCI)</th>
<th>grason h-digraph (R, HLS, ..)</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>L</td>
<td></td>
</tr>
<tr>
<td>polar v-digraph (VC, R, HC)</td>
<td>channel v-incidences (R, VCI)</td>
<td>grason v-digraph (R, VLS, ..)</td>
</tr>
<tr>
<td>U</td>
<td></td>
<td></td>
</tr>
<tr>
<td>channel digraph (C, TJ, R)</td>
<td>channel incidences (R, CI)</td>
<td>grason digraph (R, LS, TJ)</td>
</tr>
<tr>
<td></td>
<td>U</td>
<td></td>
</tr>
<tr>
<td>colored floorplan graph (TJ, LS, R)</td>
<td></td>
<td>colored grason graph (R, LS, TJ)</td>
</tr>
<tr>
<td>floorplan graph</td>
<td></td>
<td>grason graph</td>
</tr>
</tbody>
</table>
more observations

• grason (di-)graphs represent adjacencies directly
• graphs representing preferred adjacencies are not always grason graphs
• if a graph is grason graph then it is the dual of a floorplan graph
• are grason graphs easily characterizable?
forbidden sub-digraph theorem

no directed path can be parallel to an arc
grason rules

1. sub-digraph rule

2. degree rule

J. Grason, 1966
R. Schmid, 1987
more observations

- Grason (di-)graphs represent adjacencies directly
- Graphs representing preferred adjacencies are not always grason graphs
- If a graph is grason graph, then it is the dual of a floorplan graph
- Are grason graphs easily characterizable?

YES!!

**SO WHAT??**

- More importantly is an adjacency preference graph "grasonizable"?
- Or more commonly phrased:

  is a given adjacency graph a subgraph of a graph that has a rectangular dual (and if not ...........)

rectangular duals

problems:  
- does a given graph have a rectangular dual?  
- if so, construct a corresponding rectangle dissection

the given graph represents required adjacencies (edges) between modules (vertices) and the sides or the periphery
impossible adjacencies

- four rectangles cannot be neighbours of each other
- the plane grason graph cannot have complex triangles
  - a complex triangle is 3-circuit enclosing vertices
- adjacency graph with complex triangles cannot be a subgraph of a grason graph (i.e. a graph with rectangular dual)
- besides, any internal vertex has a degree of at least four
existence of rectangular duals (sides known)

A graph is the dual of a floorplan graph with cube extension fif it has

- at least 6 vertices
- at least 1 vertex of degree 4
- a plane triangulation without complex triangles

K. Kozminski, 1985
J. Bhasker, 1987

A plane triangulation does not contain complex triangles fif it is 4-connected
two reduction cases

1. at least one of the "side vertices" has degree 3
two reduction cases

1. at least one of the "side vertices" has degree 3
two reduction cases

1. at least one of the "side vertices" has degree 3

2. all "side vertices" have degree 4 or more
two reduction cases

1. at least one of the "side vertices" has degree 3

2. all "side vertices" have degree 4 or more
two reduction cases

1. at least one of the "side vertices" has degree 3

2. all "side vertices" have degree 4 or more
outline of proof

assume that it is possible to generate rectangle dissections
for all extended plane triangulations with less than \( n \) vertices

a plane triangulation with \( n \) vertices satisfies
one of the two reduction cases

reduction case 1: reduces the problem to a graph with \( n-1 \) vertices
the rectangular dual is to be combined with
a single rectangle on the appropriate side

reduction case 2: reduces the problem to
two graphs with less than \( n \) vertices;
the two rectangular duals have to be combined by
aligning the rectangles represented in the splitting path

to complete that scheme into a proof:

- show that a splitting path exist in reduction case 2
- show that splitting does not create complex triangles
- show that the path adjacencies can be of the same kind
existence of a rectangular dual (only periphery)

A graph $G$ with a designated vertex $e$ is the periphery graph of a dissection if

- $G$ has no articulation vertex
- $G\setminus\{e\}$ has a plane triangulation without complex triangles
- there is a corner assignment such that every complex triangle splits the periphery in two parts each containing a corner

K. Kozminski, 1985
R. Schmid, 1987
floorplanning by dualization

- existence of a rectangular dual can be established in linear time
- when a rectangular dual exists it can be constructed during the existence test
- the algorithm needs a full grason graph (all adjacencies given)
- in practice, both "missing" and "additional" adjacencies occur

problems of subgraphs and supergraphs that have rectangular duals

for example: select a minimum subset $F$ of $E$ such that every complex triangle has an edge in $F$

- weighted complex triangle elimination is NP-hard
  - however, an approximation algorithm is known to exist
- the complexity of the unweighted version is unknown
  - but that is not a very useful version
adjacency based floorplan design

1. adjacencies (from proximities, net list, ..)
2. planarization
3. plane triangulation (no complex triangles)
4. corner assignment
5. dualization
6. floorplan
7. dissection

?
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 constraints

fixed shape

flexible block

channel

choice of blocks
bounding functions

shaded area is "bounded region"

shape constraints:
A ∈ support F
B ≥ F(A)

1. support $F \subseteq \mathbb{R}_0^+$, connected, and $\infty \in$ support $F$

2. right continuous

3. monotonously not increasing

(4. Piecewise Linear)
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
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)
- optimization
- floorplan
- shape constraints
- rectangle dissection
quadratic programming

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

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

- 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

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

\[ A \geq 0 \]

\[ c \geq 0 \]

minimize: \[ h_0 \quad w_0 \]
linear programming

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

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

$$A \geq 0$$

$$c \geq 0$$

minimize: $\ a \ h_0 + b \ w_0$

Otten: 1975
linear programming

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

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

- minimize: \( a \ h_0 + b \ w_0 \)

Otten: 1975
mixed 0-1 linear programming

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

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

\[v \in \{0, 1\}\]

\[A \geq 0\]

\[c \geq 0\]

- 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

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

floorplan optimization is strictly np-hard

L. Stockmeyer, 1983