Floorplanning

Readings: Section 6.1

Assign portions of design to regions of the chip
Blocks have fixed or adjustable sizes

Seek to reduce routing delay & area

Hierarchical Floorplans

Assume all circuit elements of fixed size (will relax later)

Define a floorplan via a tree (hierarchy)
Nodes indicate how elements are jointed

Tree operations
Pairs
Triples
quintuples
...

Partitioning

Floorplanning

Placement

Global Routing

Detailed Routing

Compaction

Control
FP
Reg
Mult
Cache
ALU

Control
FP
Reg
Mult
Cache
ALU
**Generic Hierarchical Algorithm**

Form tree with node fanout <= restriction

Foreach node, bottom up
  Select best of all possible groupings
  (Minimize area, incl. routing, or delay)

**Hierarchical Sizing Example**

<table>
<thead>
<tr>
<th>Width</th>
<th>Height</th>
<th>Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.2x1.6</td>
<td>5x1.3</td>
<td>1.1x1.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Width</th>
<th>Height</th>
<th>Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
</tbody>
</table>
Bottom-Up Tree Construction

Cluster nodes based on connectivity
No cluster may have more nodes than size restriction

Size Imbalances in Bottom-Up Tree Construction

Problem: Small, low-connectivity nodes left to top of tree
Solution: Limit the size of clusters at higher levels
**Top-Down Tree Construction**

Recursively partition logic
Partition sub-partitions until # of nodes <= restriction

**Variable Node Size in Floorplanning**

Many cells are variable size - multiple implementations possible
Express as set of (w,h) of possible implementations
Variable Node Size in Floorplanning (cont.)

Merge:

Variable Node Size Algorithm

Express as set of \((w,h)\) of possible implementations

Step 1: Order in increasing width (decreasing height), removing useless sizes

Step 2: Form sizes of sub-floorplans iteratively

   From smallest width to largest:
     Form floorplan with front of list
     Remove list front w/largest height (both if heights are equal) and repeat
     Stop when either list empty
   From largest width to smallest:
     Form floorplan with front of list
     Remove list end w/largest width (both if widths are equal) and repeat
     Stop when either list empty

Step 3: Merge together lists, from smallest width to largest:

   Remove useless from front of lists, then accept element with smallest width
   When one list empty, add all of other list
Linear Programs (our first “Oracle”)

Problems written as set of linear inequalities, plus goal to maximize/minimize

Example: A treasure pile has 20 bars of gold, 1lb each, worth $1,000/bar
400 silver coins, 0.1lb each, worth $50/coin. If you can only carry C lbs of treasure, how much should you take (C is a constant).

Mixed Integer Linear Programming for Floorplanning

Previous techniques use relatively local optimization to form tree
Develop more global solver technique

Use Integer Linear Programming (ILP)
Formulate as series of linear integer equations and an optimization goal:

1.) No cells can overlap
2.) Cells must be given the proper area
3.) Minimize the overall area of the floorplan

ILP is NP-Complete, but good heuristic algorithms are available
Cell Variables

For cell i
- min_aspect<sub>i</sub>, max_aspect<sub>i</sub>: provided minimum & maximum aspect ratio w/h<sub>i</sub>
- area<sub>i</sub>: provided required area
- w<sub>i</sub>, h<sub>i</sub>: calculated width and height
- (x<sub>i</sub>, y<sub>i</sub>): calculated coordinates of lower left corner

For floorplan
- F_width, F_height: calculated overall floorplan width and height

```
min_aspect<sub>i</sub> \leq \frac{w<sub>i</sub>}{h<sub>i</sub>} \leq max_aspect<sub>i</sub>
```

Non-Overlapping Cells

To not overlap, other cells must be to the right, left, above, or below this one.

```
x_k + w_k \leq x_i
x_k + w_k \leq x_i
y_k + h_k \leq y_i
```

Requires OR of 4 equations

Add binary values P<sub>k</sub> and Q<sub>k</sub>, where (00→right, 01→left, 10→up, 11→down)

- P<sub>k</sub> integer; Q<sub>k</sub> integer;
- P<sub>k</sub> > -1, P<sub>k</sub> < 2, Q<sub>k</sub> > -1, Q<sub>k</sub> < 2
- right(00): x<sub>i</sub> + w<sub>i</sub> ≤ x<sub>k</sub> + \infty((P<sub>k</sub> + Q<sub>k</sub>))
- left(01): x<sub>k</sub> + w<sub>k</sub> ≤ x<sub>i</sub> + \infty((P<sub>k</sub> + (1 - Q<sub>k</sub>))
- up(10): y<sub>i</sub> + h<sub>i</sub> ≤ y<sub>k</sub> + \infty((1 - P<sub>k</sub>) + Q<sub>k</sub>)
- down(11): y<sub>k</sub> + h<sub>k</sub> ≤ y<sub>i</sub> + \infty((1 - P<sub>k</sub>) + (1 - Q<sub>k</sub>))
Cell Area & Aspect Ratios

Problem:
   Area is non-linear in width and height
Solution:
   Linearize the function (inexact)

\[ h_i = \text{slope}_i \times w_i + \text{intercept}_i \]

Aspect ratio:
   Given area and aspect ratio, can determine \( \text{max\_width}_i \) and \( \text{max\_height}_i \)

\[ h_i \leq \text{max\_height}_i \]
\[ w_i \leq \text{max\_width}_i \]

Minimize Overall Area

Goal is “Minimize the overall area of the floorplan”
   Again, area is not a linear equation
Solution:
   Fix overall width, and solve for height. Binary search width for best area

\[ \text{minimize } F_{\text{height}} \]
\[ y_i + h_i \leq F_{\text{height}} \]
\[ x_i + w_i \leq F_{\text{width}} \]
\[ 0 \leq y_i \]
\[ 0 \leq x_i \]