EE582
Physical Design Automation of VLSI Circuits and Systems

Prof. Dae Hyun Kim
School of Electrical Engineering and Computer Science
Washington State University

Placement
Metrics for Placement

- Wirelength
- Timing
- Power
- Routing congestion
Wirelength Estimation

- Half-perimeter wirelength (HPWL)

\[ HPWL = W + H \]
Wirelength Estimation
Placement Algorithms

• Constructive
  – Min-cut based placement
  – Force-directed

• Analytical
  – Gordian
  – Kraftwerk

• Iterative improvement
  – Simulated annealing (Timberwolf)
  – Pairwise exchange
Min-Cut-Based Placement

• Idea
  – Cutsize minimization ≈ Reduction of global wires
Min-Cut-Based Placement

- Partitioning
Min-Cut-Based Placement

- Algorithm
  - Min_Cut_Placement (N, n, C)
    /* N: layout
       n: # cells to be placed
       n_0: # cells in a slot
       C: connectivity matrix (netlist) */
    begin
      if ( n ≤ n_0 ) then
        place_cells (N, n, C);
      else
        (N_1, N_2) = cut_surface (N);
        (n_1, C_1), (n_2, C_2) = partition (n, C);
        Min_Cut_Placement (N_1, n_1, C_1);
        Min_Cut_Placement (N_2, n_2, C_2);
      end
Min-Cut-Based Placement

- Example (Quadrature placement)
  - KL partitioning + Quadrature placement
Min-Cut-Based Placement

• Terminal propagation
  – Dunlop and Kernighan, TCAD’85

• Original min-cut placement algorithm
  – Does not consider the locations of terminal pins.
Min-Cut-Based Placement

• What if we swap \{1,3,6,9\} and \{2,4,5,7\}?
Min-Cut-Based Placement

- Terminal propagation
Min-Cut-Based Placement

- Terminal propagation

$P$ will stay in $R1$ for the rest of partitioning!
Min-Cut-Based Placement

- Terminal propagation

---

Don’t use $p$ to bias the solution in either direction!

Use $p$!
Min-Cut-Based Placement

- Terminal propagation

![Diagram showing terminal propagation](attachment:image.png)
Min-Cut-Based Placement

• Example

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

undirected graph model w/ k-clique weighting
thin edges = weight 0.5, thick edges = weight 1
Min-Cut-Based Placement

• Example

(a) cut 1  
(b) cut 2
Min-Cut-Based Placement

• Example

[Diagram showing a graph with nodes labeled m, n, i, j, k, l, o, p, g, h, c, d, e, f, a, b and connections between them. Two windows are marked with p1 and p2.]
Min-Cut-Based Placement

• Example
Min-Cut-Based Placement

• Example
Min-Cut-Based Placement

• Example

```
  m  n  o  p  
  i  j  k  h  
  e  f  c  g  
  a  b  d  l  

<p>| | | |
|   |   |   |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
</table>
  \  \  \  
  |  |  |  |
  \  \  \  
  p1

  n  o  p
  j  k  h
```

window
Min-Cut-Based Placement

• Example
Min-Cut-Based Placement

• Example
This page is intentionally left blank.
Analytical Placement

• Kraftwerk2
• Net model
  – Two-pin nets

• Connectivity matrix
  \[
  \begin{pmatrix}
  0 & 1 & 0 & 1 \\
  1 & 0 & 1 & 0 \\
  0 & 1 & 0 & 1 \\
  1 & 0 & 1 & 0 \\
  \end{pmatrix}
  \]

• Ideal wirelength cost function
  – \( \Gamma = |x_1 - x_2| + |y_1 - y_2| \)

• Quadratic placement
  – Cost function \( \Gamma \) is quadratic.
• Quadratic cost function
  \[ \Gamma = (x_1 - x_2)^2 + (y_1 - y_2)^2 = \Gamma_x + \Gamma_y \]

• \( \Gamma_x \): x-component
• \( \Gamma_y \): y-component

• How can we optimize the cost function?
  \[ \frac{\partial \Gamma_x}{\partial x_1} = 0, \frac{\partial \Gamma_x}{\partial x_2} = 0 \]
  \[ \frac{\partial \Gamma_y}{\partial y_1} = 0, \frac{\partial \Gamma_y}{\partial y_2} = 0 \]
• Example

\[ \Gamma_x = (x - p_1)^2 + (x - p_2)^2 \]
\[ \frac{\partial \Gamma_x}{\partial x} = 0 = 2(x - p_1) + 2(x - p_2) \]
\[ x = \frac{p_1 + p_2}{2} \]
Kraftwerk2

• Example

\[ \Gamma_x = (x_1 - p_1)^2 + (x_1 - x_2)^2 + (x_2 - p_2)^2 \]
\[ \frac{\partial \Gamma_x}{\partial x_1} = 0 = 2(x_1 - p_1) + 2(x_1 - x_2) \]
\[ \frac{\partial \Gamma_x}{\partial x_2} = 0 = -2(x_1 - x_2) + 2(x_2 - p_2) \]
\[ x_1 = \frac{2p_1 + p_2}{3} \]
\[ x_2 = \frac{p_1 + 2p_2}{3} \]
Kraftwerk2

• The location of each cell is represented by
  – \((x, y)\)

• The x-locations of M movable cells
  – \(x = (x_1, x_2, \ldots, x_M)^T\)
  – \(C_x\): cell-to-cell connectivity matrix
  – \(d_x\): cell-to-pin connectivity matrix (constant)

\[
\Gamma_x = 0.5 \ x^T C_x x + x^T d_x + \text{const.}
\]
Kraftwerk2

• Wirelength minimization

\[ \frac{\partial \Gamma_x}{\partial x_1} = 0, \frac{\partial \Gamma_x}{\partial x_2} = 0, \ldots, \frac{\partial \Gamma_x}{\partial x_M} = 0 \]

– i.e., \( \nabla_x \Gamma_x = C_x x + d_x = 0 \)

• where \( \nabla_x = (\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \ldots, \frac{\partial}{\partial x_M})^T \)

– Net force: \( F_{x}^{\text{net}} = \nabla_x \Gamma_x = C_x x + d_x \)

\[ \Gamma_x = 0.5 x^T C_x x + x^T d_x + \text{const.} \]
• Applying only the net force makes a lot of overlaps.
Kraftwerk2

• Move force
  – Removes overlaps = spread cells out.

• \( x \): current location (to be computed)
• \( x' \): last location
• Change in the cell location
  – \( \Delta x = x - x' \)
Kraftwerk2

• Density function
  – $D_{\text{cell}}(x, y)$
    • Cell density at each location

• Move force
  – $F_{x,i}^{\text{move}} = w_i \cdot (x_i - x_i^o)$
    • $x_i$: current location (to be computed)
    • $x_i^o$: target location

\[
\dot{x}_i = x_i' - \frac{\partial}{\partial x} \Phi(x, y) \bigg|_{(x_i', y_i')} \quad \text{Density}
\]
Kraftwerk2

- Net force

\[ F_{x}^{\text{net}} = C_{x}x + d_{x} \]

- Move force

\[ F_{x}^{\text{move}} = C_{x}^{o}(x - x^{o}) = C_{x}^{o}(x - x' + \Phi_{x}) \]
Kraftwerk2

- Net force is used for wirelength minimization
- Do not collapse the cells back to their initial locations.

- Hold force

\[ F_x^{\text{hold}} = -(C_x x' + d_x) \]
• Final equation

\[ F_x^{\text{net}} + F_x^{\text{move}} + F_x^{\text{hold}} = 0 \]

\[ [C_x x + d_x] + [C_x^o (x - x' + \Phi_x)] + [-(C_x x' + d_x)] = 0 \]

\[ (C_x + C_x^o) \cdot (x - x') = -C_x^o \cdot \Phi_x \]

\[ (C_x + C_x^o) \cdot \Delta x = -C_x^o \cdot \Phi_x \]

\[ x = x' + \Delta x \]
Kraftwerk2

(a) 1  
(b) 5a  
(c)
### TABLE II
RESULTS IN THE ISPD 2005 CONTEST BENCHMARK SUITE

| Circuit   | Kraftwerk2 HPWL | Kraftwerk2 CPU | FastPlace3 HPWL | FastPlace3 CPU | RQL HPWL | RQL CPU | NTUPlace3 HPWL | NTUPlace3 CPU | APlace2 HPWL | APlace2 CPU | mFAR HPWL | mFAR CPU | Dragon HPWL | Dragon CPU | mPL5 HPWL | mPL5 CPU | Capo HPWL | Capo CPU | FengShui HPWL | FengShui CPU |
|-----------|-----------------|----------------|-----------------|----------------|----------|---------|----------------|---------------|-------------|------------|----------|----------|----------|----------|-----------|----------|----------|--------|----------|-----------|----------|
| adaptec   | 82.43           | 262            | 79.38           | 353            | 77.82    | 751     | 80.93          | 883            | n.a.        | n.a.      | n.a.     | n.a.     | n.a.     | n.a.     | n.a.     | n.a.     | n.a.    |
| adaptec2  | 92.85           | 349            | 93.08           | 559            | 88.51    | 1247    | 89.85          | 906            | 87.31       | 91.53     | 94.72    | 97.11    | 99.71    | 122.99   |         |
| adaptec3  | 227.22          | 713            | 217.80          | 2275           | 210.96   | 2405    | 214.20         | 1944           | n.a.        | n.a.      | n.a.     | n.a.     | n.a.     | n.a.     | n.a.     | n.a.    |
| adaptec4  | 199.43          | 709            | 201.36          | 1411           | 188.86   | 2096    | 193.74         | 2325           | 187.65      | 190.84    | 200.88   | 200.94   | 211.25   | 337.22   |         |
| bigblue1  | 97.67           | 407            | 95.68           | 604            | 94.98    | 1160    | 97.28          | 1675           | 94.64       | 97.70     | 102.39   | 98.31    | 108.21   | 114.57   |         |
| bigblue2  | 154.74          | 559            | 155.10          | 1380           | 150.03   | 2261    | 152.20         | 3352           | 143.82      | 168.70    | 159.71   | 173.22   | 172.30   | 285.43   |         |
| bigblue3  | 343.32          | 2070           | 379.88          | 4642           | 323.09   | 4864    | 348.48         | 6256           | 357.89      | 379.95    | 380.45   | 369.66   | 382.63   | 471.15   |         |
| bigblue4  | 852.40          | 4147           | 832.88          | 6862           | 797.66   | 12410   | 829.16         | 11308          | 833.21      | 876.28    | 903.96   | 904.19   | 1098.76  | 1040.05  |         |
| Average   | 1.000           | 1.000          | 1.000           | 2.00           | 0.959    | 3.12    | 0.979          | 3.48           | 0.967       | 1.028     | 1.046    | 1.053    | 1.126    | 1.438    |         |
### TABLE III

**RESULTS IN THE ISPD 2006 CONTEST BENCHMARK SUITE.** (a) **Kraftwerk’s Results.** As required in this benchmark suite, the CPU factor is limited to ±10%. The “raw” CPU factors are −13.50% and −10.98%, respectively. (b) Results of Other Placers

<table>
<thead>
<tr>
<th>Circuit</th>
<th>HPWL</th>
<th>Overflow factor</th>
<th>CPU</th>
<th>CPU factor</th>
<th>HPWL</th>
<th>HPWL+ Overflow</th>
<th>HPWL+ Overflow+ CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>adaptec5</td>
<td>433.84</td>
<td>3.606%</td>
<td>1618</td>
<td>−9.35%</td>
<td>1.071</td>
<td>1.032</td>
<td>0.939</td>
</tr>
<tr>
<td>newblue1</td>
<td>65.92</td>
<td>0.415%</td>
<td>603</td>
<td>−8.38%</td>
<td>1.057</td>
<td>1.043</td>
<td>0.956</td>
</tr>
<tr>
<td>newblue2</td>
<td>203.91</td>
<td>1.286%</td>
<td>508</td>
<td>−10.00%*</td>
<td>1.033</td>
<td>1.082</td>
<td>0.975</td>
</tr>
<tr>
<td>newblue3</td>
<td>278.51</td>
<td>0.382%</td>
<td>526</td>
<td>−10.00%+</td>
<td>1.018</td>
<td>1.067</td>
<td>0.961</td>
</tr>
<tr>
<td>newblue4</td>
<td>304.24</td>
<td>1.709%</td>
<td>1553</td>
<td>−8.63%</td>
<td>1.068</td>
<td>1.033</td>
<td>0.945</td>
</tr>
<tr>
<td>newblue5</td>
<td>548.38</td>
<td>2.694%</td>
<td>2622</td>
<td>−9.50%</td>
<td>1.109</td>
<td>1.054</td>
<td>0.957</td>
</tr>
<tr>
<td>newblue6</td>
<td>528.59</td>
<td>1.702%</td>
<td>2579</td>
<td>−9.89%</td>
<td>1.048</td>
<td>1.036</td>
<td>0.936</td>
</tr>
<tr>
<td>newblue7</td>
<td>1126.58</td>
<td>3.155%</td>
<td>4828</td>
<td>−9.06%</td>
<td>1.053</td>
<td>1.051</td>
<td>0.958</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td><strong>1.869%</strong></td>
<td><strong>−9.35%</strong></td>
<td><strong>1.057</strong></td>
<td><strong>1.050</strong></td>
<td><strong>0.953</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Placer</th>
<th>Overflow factor</th>
<th>CPU factor</th>
<th>HPWL</th>
<th>HPWL+ Overflow</th>
<th>HPWL+ Overflow+ CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kraftwerk2</td>
<td>1.87 %</td>
<td>−9.35 %</td>
<td>1.057</td>
<td>1.050</td>
<td>0.953</td>
</tr>
<tr>
<td>NTUplace3</td>
<td>6.26 %</td>
<td>−2.61 %</td>
<td>0.976</td>
<td>1.007</td>
<td>0.990</td>
</tr>
<tr>
<td>RQL</td>
<td>6.80 %</td>
<td>n.a. %</td>
<td>0.981</td>
<td>1.018</td>
<td>n.a.</td>
</tr>
<tr>
<td>Fastplace3</td>
<td>n.a.</td>
<td>−8.17 %</td>
<td>n.a.</td>
<td>n.a.</td>
<td>1.040</td>
</tr>
<tr>
<td>mPL6</td>
<td>1.36 %</td>
<td>1.58 %</td>
<td>1.035</td>
<td>1.020</td>
<td>1.040</td>
</tr>
<tr>
<td>mFAR</td>
<td>2.71 %</td>
<td>−0.12 %</td>
<td>1.108</td>
<td>1.107</td>
<td>1.108</td>
</tr>
<tr>
<td>APlace3</td>
<td>3.83 %</td>
<td>5.31 %</td>
<td>1.097</td>
<td>1.107</td>
<td>1.165</td>
</tr>
<tr>
<td>Dragon</td>
<td>0.12 %</td>
<td>−5.90 %</td>
<td>1.331</td>
<td>1.300</td>
<td>1.232</td>
</tr>
<tr>
<td>DPlace</td>
<td>9.32 %</td>
<td>−4.54 %</td>
<td>1.343</td>
<td>1.414</td>
<td>1.364</td>
</tr>
<tr>
<td>Capo</td>
<td>0.32 %</td>
<td>2.69 %</td>
<td>1.375</td>
<td>1.344</td>
<td>1.385</td>
</tr>
</tbody>
</table>

(a)
### TABLE IV

**Results in Mixed-Size and Floorplacement Benchmark Suites.**

(a) ICCAD 2004 Mixed-Sized Benchmark Suite. (b) IBM-HB+ Floorplacement Benchmark Suite.

<table>
<thead>
<tr>
<th>Circuit</th>
<th><strong>Kraftwerk2</strong></th>
<th><strong>FDP</strong></th>
<th><strong>NTUPlace3</strong></th>
<th><strong>APlace2</strong></th>
<th><strong>mPL5</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>HPWL</td>
<td>CPU</td>
<td>HPWL</td>
<td>CPU</td>
<td>HPWL</td>
</tr>
<tr>
<td>ibm01</td>
<td>2.24</td>
<td>11</td>
<td>2.42</td>
<td>145</td>
<td>2.17</td>
</tr>
<tr>
<td>ibm02</td>
<td>4.90</td>
<td>27</td>
<td>5.11</td>
<td>284</td>
<td>4.63</td>
</tr>
<tr>
<td>ibm03</td>
<td>6.61</td>
<td>24</td>
<td>7.08</td>
<td>337</td>
<td>6.65</td>
</tr>
<tr>
<td>ibm04</td>
<td>7.63</td>
<td>29</td>
<td>7.69</td>
<td>317</td>
<td>7.21</td>
</tr>
<tr>
<td>ibm05</td>
<td>9.79</td>
<td>33</td>
<td>n.a.</td>
<td>n.a.</td>
<td>9.66</td>
</tr>
<tr>
<td>ibm06</td>
<td>6.11</td>
<td>40</td>
<td>6.20</td>
<td>389</td>
<td>5.94</td>
</tr>
<tr>
<td>ibm07</td>
<td>10.42</td>
<td>52</td>
<td>10.57</td>
<td>607</td>
<td>9.90</td>
</tr>
<tr>
<td>ibm08</td>
<td>12.97</td>
<td>85</td>
<td>13.30</td>
<td>719</td>
<td>12.29</td>
</tr>
<tr>
<td>ibm09</td>
<td>11.98</td>
<td>71</td>
<td>13.30</td>
<td>713</td>
<td>12.00</td>
</tr>
<tr>
<td>ibm10</td>
<td>30.15</td>
<td>232</td>
<td>30.70</td>
<td>924</td>
<td>28.49</td>
</tr>
<tr>
<td>ibm11</td>
<td>17.59</td>
<td>107</td>
<td>18.41</td>
<td>950</td>
<td>17.54</td>
</tr>
<tr>
<td>ibm12</td>
<td>31.42</td>
<td>124</td>
<td>36.46</td>
<td>1472</td>
<td>32.07</td>
</tr>
<tr>
<td>ibm13</td>
<td>22.48</td>
<td>147</td>
<td>23.60</td>
<td>1175</td>
<td>22.16</td>
</tr>
<tr>
<td>ibm14</td>
<td>35.13</td>
<td>308</td>
<td>37.84</td>
<td>2185</td>
<td>35.36</td>
</tr>
<tr>
<td>ibm15</td>
<td>47.58</td>
<td>468</td>
<td>47.69</td>
<td>2468</td>
<td>45.38</td>
</tr>
<tr>
<td>ibm16</td>
<td>54.17</td>
<td>527</td>
<td>61.27</td>
<td>2792</td>
<td>57.59</td>
</tr>
<tr>
<td>ibm17</td>
<td>66.63</td>
<td>474</td>
<td>69.45</td>
<td>3577</td>
<td>66.73</td>
</tr>
<tr>
<td>ibm18</td>
<td>42.36</td>
<td>609</td>
<td>44.88</td>
<td>4369</td>
<td>41.58</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td>1.000</td>
<td>1.00</td>
<td>1.056</td>
<td>9.02</td>
<td>0.982</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Circuit</th>
<th><strong>Kraftwerk2</strong></th>
<th><strong>SCAMPI</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>HPWL</td>
<td>CPU</td>
</tr>
<tr>
<td>ibm-HB+01</td>
<td>2.82</td>
<td>10</td>
</tr>
<tr>
<td>ibm-HB+02</td>
<td>5.87</td>
<td>26</td>
</tr>
<tr>
<td>ibm-HB+03</td>
<td>9.23</td>
<td>16</td>
</tr>
<tr>
<td>ibm-HB+04</td>
<td>9.98</td>
<td>21</td>
</tr>
<tr>
<td>ibm-HB+06</td>
<td>8.79</td>
<td>12</td>
</tr>
<tr>
<td>ibm-HB+07</td>
<td>14.80</td>
<td>16</td>
</tr>
<tr>
<td>ibm-HB+08</td>
<td>21.27</td>
<td>19</td>
</tr>
<tr>
<td>ibm-HB+09</td>
<td>17.44</td>
<td>18</td>
</tr>
<tr>
<td>ibm-HB+10</td>
<td>47.51</td>
<td>47</td>
</tr>
<tr>
<td>ibm-HB+11</td>
<td>25.92</td>
<td>23</td>
</tr>
<tr>
<td>ibm-HB+12</td>
<td>51.38</td>
<td>43</td>
</tr>
<tr>
<td>ibm-HB+13</td>
<td>34.90</td>
<td>23</td>
</tr>
<tr>
<td>ibm-HB+14</td>
<td>63.11</td>
<td>42</td>
</tr>
<tr>
<td>ibm-HB+15</td>
<td>92.88</td>
<td>46</td>
</tr>
<tr>
<td>ibm-HB+16</td>
<td>95.60</td>
<td>54</td>
</tr>
<tr>
<td>ibm-HB+17</td>
<td>148.16</td>
<td>96</td>
</tr>
<tr>
<td>ibm-HB+18</td>
<td>73.95</td>
<td>52</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td>1.000</td>
<td>1.00</td>
</tr>
</tbody>
</table>