Project Overview

• What is the project about?
  ○ Using Normalized polish expression based Simulated annealing to determine the location of block in a floorplan.

• Goal of the project?
  ○ Generate floorplan without overlaps and reduce the total floorplan area.

• Scope?
  ○ Used rectangular shaped hard blocks with no rotation allowed.

• Language used?
  ○ C++ and Java
Normalized Polish Expression

- Polish expression can be obtained from the post-order traversal on binary tree.
- A polish expression which does not have any consecutive operators of same type (H or V).
- Using LIFO structure to convert polish expression to binary tree.

PE = 25V1H374VH6V8VH
Algorithm
Start
E-> Initial Exp

MT->0
Uphill ->0

Randomly Select Move

M1
M2
M3

Calculate New Area

Gain or Random < e^{-Δcost/T}

True
Accept

False
Reject

uphill>N or MT > 2N

Reject/MT > .95 or T < ε or out of time

Reduce Temp

End

Yes

No

Yes
Data Structures

• Arrays
  o For storing width and height of each block.
  o For storing the polish expression.

• Stacks
  o For calculating area and generating trees.

• Binary Trees
  o For calculating the coordinates and generating graphs.
Implementation

• Parse the initial normalized polish expression and the size of the blocks from a text file.
• Store the polish expression as integer array where H and V are replaced by -1 and -2 respectively.
• Cost function was defined as the total Area of floorplan which is calculated as following :-

\[
\begin{array}{c|c}
\text{Block No} & \text{Size} \\
\hline
0 & 1,3 \\
1 & 5,1 \\
2 & 3,3 \\
\end{array}
\]

\[
\text{Height of } H = 1 + 3 = 4
\]

\[
\text{Width of } H = \max(1,5) = 5
\]

\[
\text{Height of } V = \max(3, 4) = 4
\]

\[
\text{Width of } V = 5 + 3 = 8
\]

\[
\text{Area} = 8 \times 4 = 32
\]
Calculating the floorplan coordinates

• Block Orientation :-

\[
\begin{align*}
\text{Width} \, H &= \max(\text{Width} \, 1, \text{Width} \, 2) \\
\text{Height} \, H &= \text{Height} \, 1 + \text{Height} \, 2 \\
\text{Width} \, V &= \text{Width} \, 1 + \text{Width} \, 2 \\
\text{Height} \, V &= \max(\text{Height} \, 1, \text{Height} \, 2)
\end{align*}
\]

• First we create a binary tree from the polish expression by using a stack.

• While creating tree we also calculate the size of each room which is as following :-
Initial

Push V in stack
H’s Coordinates of 3 = V’s coordinates of V
X Coordinate of 3 = X coordinate of H + width of H.

Pop V

((x_v, y_v), (x_v + w_v, y_v))
X Coordinate of 2 = X coordinate of H
Y Coordanate of 2 = Y coordinate of H + height of H.
Demo
## Parameters Used

<table>
<thead>
<tr>
<th>No. of Block</th>
<th>Initial Temp</th>
<th>k</th>
<th>Cooling Ratio</th>
<th>Freezing Temp</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>200</td>
<td>16</td>
<td>0.96</td>
<td>144.277</td>
</tr>
<tr>
<td>10</td>
<td>320</td>
<td>16</td>
<td>0.96</td>
<td>2.38613</td>
</tr>
<tr>
<td>30</td>
<td>380</td>
<td>15</td>
<td>0.96</td>
<td>2.7201</td>
</tr>
<tr>
<td>100</td>
<td>3422</td>
<td>16</td>
<td>0.96</td>
<td>0.00393676</td>
</tr>
<tr>
<td>150</td>
<td>10040</td>
<td>20</td>
<td>0.99</td>
<td>0.000032</td>
</tr>
</tbody>
</table>
Floorplans
5 Blocks

Initial

Area = 65

Final

Area = 40
10 Blocks

Initial
Area = 147

Final
Area = 70
100 Blocks

Initial
Area = 7119

Final
Area = 1056
150 Blocks

Final
Area = 1554

Initial
Area = 14104
## Experimental Results

<table>
<thead>
<tr>
<th>No. of Block</th>
<th>Initial Area</th>
<th>Final Area</th>
<th>Percentage Improvement</th>
<th>Time in Min</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>68</td>
<td>40</td>
<td>41.17647059</td>
<td>0.00006625</td>
</tr>
<tr>
<td>10</td>
<td>147</td>
<td>70</td>
<td>52.38095238</td>
<td>0.002007383</td>
</tr>
<tr>
<td>30</td>
<td>1075</td>
<td>308</td>
<td>71.34883721</td>
<td>0.017282667</td>
</tr>
<tr>
<td>100</td>
<td>7119</td>
<td>1056</td>
<td>85.16645596</td>
<td>0.527993333</td>
</tr>
<tr>
<td>150</td>
<td>14104</td>
<td>1554</td>
<td>88.98184912</td>
<td>8.006683333</td>
</tr>
</tbody>
</table>
Percentage White Space

No of Blocks

Percentage White space

5
10
30
100
150
Challenges

• Finding the initial parameters for simulated annealing which give minimum floorplan area.
• Reducing the run time and the memory requirement for simulated annealing.

Future Scope

• Use wire length as additional parameter in the cost function for simulated annealing.
• Add support for soft and L shaped blocks.
Conclusion

- Simulated annealing can reduce the floorplan area significantly but for the large circuits with higher initial temperature it has a long runtime.
- Runtime depends on the no of blocks, initial temperature, cooling ratio, no of moves per temperature and freezing temperature.
- No of rejected moves are more at higher temperatures.
- For our implementation there was good reduction in the floorplan area with low runtime.
Questions ?
Thank You
Reference