

CS 258F Class Project

Spring 2003

Detailed Placement for Standard Cell Based Design

Prof. Jason Cong

I. Project Description

The objective of this project is to let you gain better understanding of standard circuit placement, especially standard cell based detailed placement. Through this project, you will also get concrete experience about developing VLSI layout programs in general, representing and manipulating the netlists, and applying combinatorial optimization techniques to VLSI layout.

You are asked to develop a C++ program and solve the following problem: Given a netlist (specified in BookShelf format [5]) and a global placement result with overlaps, generate a detailed placement solution by removing the overlaps. The cells should be placed on rows according to the row specification the .scl file. Furthermore, the half perimeter wirelength of the solutions need to be optimized.

II. Input Circuits and Netlist Parser

A set of 20 circuits can be found at `/u/class/u/class/cs258f/2003_spring/ckt`. For each circuit, we have the following files:

<code>.aux</code>	A list of the component files for a problem instance
<code>.nodes</code>	Library file describing the cell library for the problem
<code>.pl</code>	Describe the location and orientation of each node
<code>.nets</code>	Describe each net and the pins it connects
<code>.scl</code>	Describe the standard cell row, i.e., its location, height, starting position of each row
<code>.wts</code>	Describe the weight associated with each net

The circuits named Pekox come from Peko (Placement Example with Known Optimal wirelength) suite. The cells in these circuits are uniform sized. A detailed description of Peko circuits and their optimal wirelength can be found at [6]. The global placement for Pekox circuit is obtained by snapping the optimal solution into a pre-defined bin structure. The circuits named ibmxx come from IBM-place benchmark suite. The cells in these circuits are non-uniform sized. The optimal placement solutions to these circuits are unknown. A detailed description of ibmxx circuits can be found at [7]. The global placements for these circuits are derived from Dragon's [7] results.

A netlist parser is available at `/u/class/cs258f/2003_spring/parser`. It takes an .aux filename as input, and allocates the datastructure of netlist in memory. The comments given in bookshelf.h describe the function of each item defined in the C++ classes.

III. Requirements

1. You need to find a partner and form a 2-person team to complete the project.
2. You need to turn in a one-page proposal by May. 13th to indicate your partner's name, division of responsibilities between the two persons, and an outline of your approach.
3. You need to turn in a class project report by Wednesday. Jun. 11th, 5pm PST. A typical report has 8-10 pages, including:
 - A brief summary of related work on this topic,
 - Detailed description of your approach,
 - Justification of your approach,
 - A table of about test circuits, including the numbers of cells, nets, and pins for each test circuit,
 - A table summarizing your solutions, including the area, total wirelength, and runtime for each solution and machine configuration.
 - Some implementation details, such as the computer and compiler used for implementation and experiment.Please use figures and tables effectively to improve the quality of your presentation. Each team needs to turn in only one report.
4. You need to inform me and Min Xie <xie@cs.ucla.edu> (via e.mail) where your program (executable and source codes) and output files are located on the departmental servers, so that Min and I can verify your results if necessary. Please make sure that I have the proper access right to these data for 3 weeks from the date you turn in the report.
5. Each team will give a 20-min presentation of their work on Friday, Jun. 13th, 8-11am PST.

IV. Grading

The grading includes two parts:

Basic: You'll receive 75% credit if your program works correctly for all test cases, and produces reasonably optimized solutions.

Competitive: The remaining 25% credit will be assigned based on how good our solution compared to others.

V. References

- [1]. K. Doll, F. Johannes, K. Antreich, "Iterative placement improvement by network flow methods," *IEEE Transactions on Computer-Aided Design of Integrated Circuits & Systems*, vol.13, no.10, Oct. 1994, pp. 1189-1200.
- [2]. J. Vygen, "Algorithms for detailed placement of standard cells," *Proc. Design, Automation and Test in Europe*, pp.321-324.
- [3]. A. Kahng, P. Tucker, A. Zelikovsky, "Optimization of linear placements for wirelength minimization with free sites," *Proc. of the ASP-DAC '99 Asia and South Pacific Design Automation Conference*, pp. 241-244.

- [4]. H. Sung-Woo, J. Lillis, "Mongrel: hybrid techniques for standard cell placement," *Proc. of IEEE/ACM International Conference on Computer Aided Design*. pp. 165-170.
- [5]. Bookshelf file format description:
<http://vlsicad.ucsd.edu/GSRC/bookshelf/Slots/Placement/plFormats.html>
- [6]. Peko placement benchmark description:
<http://ballade.cs.ucla.edu/~pubbench/placement/>
- [7]. Dragon placement benchmark description:
<http://www.cs.ucla.edu/~xjyang/Dragon/ibm-place.html>