University of Toronto, Faculty of Applied Science and Engineering
Department of Electrical and Computer Engineering

ECE 1387 - CAD for Digital Circuit Synthesis and Layout

Assignment #2 – Analytical Placement, Clique/Bound2Bound Net Models, SimPL-Style Spreading

October 2012 J. Anderson

Assignment Date: October 13, 2012

Due Date: 11:59pm, October 28, 2012, by email to Jason.

Late Penalty: **-2 marks per day late, with total marks available = 20**

You are to write an implementation of an analytical placer (AP) with overlap removal (spreading). You will also compare the clique net model to the Bound2Bound net model. As described in class, you will formulate the placement problem mathematically as a system of linear equations to be solved. You will use an existing package (UMFPACK) to represent the matrices and vectors, and then to solve the linear system.

Your program should display its results using graphics, as in Assignment #1, using the same graphics package. Your graphics should show the placement results and the connectivity between blocks (rat's nest of wires). Blocks (cells) should appear as points in your placement.

The netlist file input format has three sections. The three sections are separated from one another by a –1 appearing by itself on a line. The first section has only a single line X Y, indicating the extent of the placement region in the X and Y dimensions. The second section specifies the blocks to be placed and the connectivity between them. Each line has the following form:

blkname blocknum netnum_1 netnum_2 netnum_3 ... netnum_n -1

where blkname is the name of the cell, and blocknum is a positive integer giving the number of the cell, and the netnum_i are the numbers of the nets that are attached to that block. Every block that has the same netnum_i on its description line is attached. Note that each block may have a different number of nets attached to it. Each line is terminated by a –1.

Example input file:

```
50 50
-1
blk1 1 2 3 4 -1
blk2 2 5 4 -1
blk3 3 5 6 2 -1
blk4 4 6 3 -1
 -1
blk1 1 50 0
blk4 4 0 50
-1
```

The first line shows that the placement region spans 50 units in the X and Y dimensions. Moving onto the second section, observe that block 1 (called "blk1") is connected to nets 2, 3 and 4. Note that each net may be connected to more than two blocks (that is, there are multi-fanout nets). Also, note that net numbers are not related to block numbers.

As discussed in class, the AP formulation requires there to be a set of pre-placed (fixed) cells, usually I/Os. The third section of the netlist file specifies the placement of fixed cells. It has the following form:

blkname blocknum x_position y_position

In the above example, block 1 is pre-placed at the position with x = 50, y = 0. The list of fixed cells is terminated by a –1 by itself on a line.

You should run your placer on the **four** test circuits provided on the course web page. These are real circuits from the VPR 5.0 and MCNC circuit benchmark suites.

What to do and what to hand in?

Hand in the location of the executable. Provide instructions on how to run your program. Make sure to set file and directory permissions so I can access your placer.

You should hand in a short description of the flow of your program, the main routines and what they do. You do not need to explain analytical placement in your write-up; rather, describe (briefly) any techniques/innovations you used or experimented with for each part.

1. Formulate and solve the analytical placement problem. Use the *clique* net model[1], as described in class. Do not do any overlap removal in this step. Your program should display the placement and rat's nest (wires between cells) using the graphics package. Hand in a plot of the placement results for each test circuit. Your program should also compute the half-perimeter bounding box wirelength (HPWL) of the placement. Hand in a table showing the HPWL for each placed test circuit.

NOTE: You will need to read Section 2 of the UMFPACK quick start guide (on the web page) to formulate the problem in UMFPACK's sparse matrix format and then solve the system.

2. As in #1 above, formulate and solve the AP problem, however, in this step, use the *Bound2Bound* net model[2], as described in class, and also in the SimPL paper on the course website. Continue to refine edge weights and re-solve the system until the HPWL is not changing much (changes by less than 1% between solves). For the first solve, before any placement information is known, you may wish to use the clique model. Do not implement any overlap removal for this step. Hand in a plot of the placement results for each test circuit. Report, for each test circuit, 1) the HPWL, 2) the number of solves required for each test circuit until edge-weight convergence. Compare and contrast the values for HPWL with those observed when the clique model is used (in #1).

3. Using the Bound2Bound net model, implement SimPL-style spreading. Assume a bin grid size of 1x1, and a utilization target for each bin of 100%, i.e. $\gamma = 1$. Assume that cells are "points" (i.e. so they cannot straddle bin boundaries) and that cells have unit area (i.e. each bin of the placement bin grid is intended to accommodate 1 cell). Given a contiguous cluster of over-utilized bins, you have a choice for how to "spread" them into the expanded region R.

For spreading, you may implement the recursive alternating vertical/horizontal cut-based spreading, as described in class and in the SimPL paper. Or alternately, you may implement *any* spreading method of your own choosing to distribute the cells in the expanded region R. One idea that is easy to implement and may work (it's untested!) is to use *weighted-bipartite graph matching* to spread overlapping cells into R in a manner that minimizes the distance cells are moved from their *solved* positions to their *spread* positions. Weighted bipartite matching is a well-known problem in graph theory and a weighted matching solver with an easy-to-understand interface (that we are using in our LegUp high-level synthesis project) is available on the course webpage (courtesy C. Stachniss, University of Freiburg).

Given a spread placement, introduce weighted pseudo nets into the formulation. Update the Bound2Bound net model weights and re-solve the formulation. Iterate the spreading, pseudo-nets introduction, and net model updates as you

---

[1] In the clique model, a net with p pins is represented as a complete graph (clique) with p*(p-1)/2 edges. Each edge in the complete graph has weight of 2/p. For example, a net with 2 pins has 1 edge with edge weight = 1. A net with 4 pins has 6 edges with edge weight = 2/4.

[2] In the Bound2Bound model, for a net with p pins, the extreme pins are connected to one another, and also connected to each internal pin. The weight on ea edge is 1/[(p-1)*(edge length from prior placement)]

gradually increase the weights on the pseudo nets. You can implement the weight-increase scheme in the SimPL paper, or any other scheme you find to be effective.

Let Q represent the set of all bins in the placement bin grid. Compute the overlap:

$$Overlap = \frac{\sum_{r \in Q} \max(0, m_r - 1)}{m}$$

where m is the total number of *moveable* cells, and m_r is the number of *moveable* cells placed in bin r. m_r should be computed assuming that cells are points. The intuition behind the above equation is that each 1x1 region can accommodate one cell.

Aim to spread each test circuit until Overlap is under 20%. Report the HPWL for the final placement. Remember that you should *not* include the lengths of the pseudo nets in your HPWL values! Hand in a plot of the placement results, showing the progression of spreading for each circuit and the final spread placement (showing a few intermediate placements is enough). Describe the approach you used for spreading and any innovations you implemented. In class, I'll report on the HPWL values attained and the various spreading techniques investigated.