UNIVERSITY OF BRITISH COLUMBIA
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

EECE 583: CAD Algorithms for Integrated Circuits
2017/2018 Term 2

Assignment 3: Partitioning

Due: Monday March 5th 2018, 11:59pm

In this assignment, you are to implement an enhanced version of the Kernighan-Lin based bi-partitioning algorithm. Your program must work for circuits with multi-fanout nets (nets that connect more than two nodes).

Your program is to minimize *the number of nets* that connect blocks in the two partitions, while keeping the sizes of the two partitions exactly equal (or differing by at most 1 if there are an odd number of blocks). *This means that each net (even a multi-fanout net) contributes either 0 or 1 to the total crossing count, never more than this*. A net contributes 1 to the crossing count if it connects cells that fall in both partitions.

A simple implementation of the K&L algorithm for circuits with multi-terminal nets may *not* work well. Consider a net that connects 10 nodes. Suppose that, in the initial partition, eight of those nodes are in Partition A. According to the simple Gain equation in class, moving any of the nodes would have no impact on the Gain (considering only this net), and thus, moving any of the nodes would be considered "equal". Yet, clearly, it would be better to move one of the two nodes from Partition B to Partition A (in the hopes that the final remaining node can be moved later). In this assignment, you will enhance the K&L algorithm to work well with multi-terminal nets.

There are many ways to enhance the K&L algorithm to work well with multi-terminal nets. I want you to be creative, and come up with an effective way to do this. Your algorithm must be primarily K&L with only small enhancements; no annealing or major algorithmic changes are allowed. A good solution would be algorithmically "simple", yet give good cut-size minimization results. Part of your mark will be based on how well you manage to minimize the size of the cut-set, averaged over all benchmark circuits.

Remember, regardless of how you compute the gain, or the representation of your circuit, your goal is to minimize *the number of nets* that connect blocks in the two partitions (if this is more than the number of nets in your circuit, you know you have a problem). In your write-up, you must report this quantity for each benchmark circuit, and also show the arithmetic average across all benchmark circuits (note that "paira" and "pairb" are actually the same circuit, so only include that circuit once in your average).

The input format is exactly the same as in Assignment 2. In fact, you will use the same benchmark circuits as Assignment 2. This means you can re-use your input parser and possibly your internal data structures. This will save you a lot of time.

As in previous assignments, you should include some graphics. For this assignment, it is ok to draw your final partitioning solution only (you don't need to show the progress of the algorithm as it proceeds). However, I can imagine that it would be possible to somehow graphically show the cost function dropping as the algorithm proceeds; if you want to do this, go ahead.

Run-time is not a major issue here, but your algorithm must perform no more than 6 full iterations of K&L.

(continued on next page…)

You must handin your files using the Connect web site. You should hand in a zip file containing the following:

- A text or PDF file (report.pdf or report.txt) containing:
    - A description of how your program works (assuming I already have a basic idea of how the base K&M algorithm works). Concentrate on how you handled multi-sink nets, and anything you did to make your algorithm fast.
    - A table of results for the example files
    - An indication of how the program can be run. If you are using a departmental Linux/Unix machine, include the full path-name to the executable, and make sure the permissions are set properly (global readable/executable).
- If you are using a MAC or PC, an executable for your program. I have to be able to run this without installing any non-standard libraries.
- The source code. If you are using an departmental linux/unix machine, you can just tell me the path-name to your source code (just make sure it is globally readable/writable). If you are using a PC or Mac, you can include the source code in your zip file.

Marking Scheme for Assignment 3: Maximum Score = 17

Do you perform K&L and calculate the final cut-size correctly?

Score	0	2	4
Description	No.	Partially correct.	Always.

How good are the results? I'm going to compare your "average cut-set" with the average cut set found by everyone else in the class. Remember, you can perform at most 6 full iterations of K&L for each benchmark circuit.

Score	0	1	2	3
Description	Really Bad	Bottom ¼ of the class.	Top ½ of the class.	Top ¼ of the class

Efficiency:

Score	0	1	2
Description	The implementation is not efficient; run-time is much longer than it needs to be.	The implementation is somewhat efficient, but small changes could make it faster.	The implementation is efficient; there is clear evidence of care creating an algorithm that runs fast.

Graphics:

Score	0	1	2
Description	No or very little graphics.	The graphics are adequate in showing the final solution, or parts of the final solution are not graphically displayed.	The graphics are very good and clearly show the final solution.

Code Quality:

Score	0	1	2
Description	Code is lacking in structure and comments.	Code is of good "academic code" quality including extensive comments. Code is well structured.	Code is of industrial quality, including evidence of unit tests and/ or extensive system tests.

Report (maximum 3 pages):

Score	0	1	2
Description	Report is unclear or difficult to read and/or understand..	Report describes most aspects of marking scheme clearly. The English has grammar/clarity errors that would not be acceptable in a major IEEE or ACM journal or conference.	Report describes all aspects of marking scheme clearly. The English is of a professional standard that would be acceptable in a major IEEE or ACM journal or conference.

Initiative: (be sure to identify any extensions in your report):

Score	0	1	2
Description	Assignment implemented as in handout.	The implementation contains one or more straightforward extensions.	The implementation goes beyond what is described in the handout in a non-trivial way.