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

ECE 1387 - CAD for Digital Circuit Synthesis and Layout

Exercise #2 - Multi-Level Partitioning with hMetis

Fall 2013 J. Anderson

Assignment Date:	November 22, 2013
Due Date:	November 29, 2013, at beginning of class
Late Penalty:	**-1 mark per day late, with total marks available = 10**

The purpose of this exercise is to gain familiarity with the multi-level partitioning paradigm and specifically, with algorithm discussed in class: hMetis. You will apply hMetis to partition real industrial circuits and will evaluate the results. The coarsening and refinement phases of hMetis can be done in a number of ways. You will investigate this and will experiment with some of the parameter settings of hMetis to study their effect on partitioning quality and run-time.

hMetis was developed at the University of Minnesota. In this exercise, you will be running a stand-alone executable version of hMetis. Note that hMetis is also available as a set of libraries that you can link into your own code. This may be useful if you need a partitioner for whatever problem you need to solve.

Software Location

A Linux executable is available on the course web page.

Manual

The instruction manual for hMetis is available on the course web page.

Location of Test Circuits

A set of three circuits is available on the course web page. Each step of the exercise should be done for each of these circuits. These are industrial circuits, courtesy of IBM/UCLA, that are part of the ISPD'98 benchmark suite, an important benchmark suite in placement, routing and partitioning research. Note that one of these circuits *ibm18* has more than 200K cells! Thus, hMetis is indeed able to handle some pretty big circuits (graphs) in relatively low run-time. You may want to use this suite in your own CAD research and more info can be found at:

http://vlsicad.ucsd.edu/UCLAWeb/cheese/ispd98.html

Exercise

For all steps in this exercise, the desired number of partitions is 2 (bi-partitioning) and the balance factor should be set to **10** (no partition will have less than **40%** of the vertices).

1. The first task is to explore the run-time/quality trade-offs of hMetis. In this step, set the coarsening scheme to HFC (1), the refinement scheme to FM (1), the V-cycle refinement scheme to not run any V-cycles (0), and the reconstruct parameter to 0 (remove cut hyperedges from the graph).

- Run hMetis 5 times on each circuit with the "Nruns" parameter set to 1. For each execution of hMetis, note the cut size and run-time. Since hMetis is non-deterministic, you may observe that run-time and cut size varies with each execution of hMetis. For each circuit, average the run-time and cut size over all the hMetis executions. Record, for each circuit, the average and standard deviation of run-time and cut size when hMetis is executed with Nruns=1.

- Repeat the previous step two times with different settings for "Nruns", specifically, {3,5}. This will provide additional data points for each circuit: you will have the average and deviation of run-time and cut size when hMetis does 3 partitionings, and 5 partitionings.

- Report the data from the two tasks in a table. Comment on the results. How much is the partitioner's quality and run-time impacted by non-determinism and how much by the # of bipartitions computed?

2. hMetis can work with various coarsening algorithms to build the successively smaller graphs. In this step, fix the Nruns parameter at 3 and for the remaining parameters, use the same settings as above. Experiment with different settings for the coarsening style (described in the hMetis manual), running hMetis 5 times on each circuit for each coarsening style considered. Examine the average run-time and cut size results. Summarize your observations. Is there a single coarsening approach that works best for all of the circuits?

3. hMetis can apply different refinement schemes during its de-coarsening phase. In this step, fix the coarsening style at HFC (1), Nruns at 3, and the remaining parameter settings as above. Experiment with how refinement style affects run-time and partitioning quality, again running hMetis 5 times on each circuit for each refinement style considered. Summarize your observations. Comment on the run-time reductions and quality degradations when the "early-exit" FM refinement is used vs. traditional FM refinement.

4. Fix Nruns at 3, HFC at 1, and use the early-exit FM refinement scheme, experiment with the V-cycles notion in hMetis. What are V-cycles and why do they help improve partition quality? What is the quality benefit and run-time cost of using V-cycles?