EE382V: VLSI Physical Design Automation

Spring 2015 (Prof. David Pan)

Homework #1: due Feb. 11 (in class), 2015

1. KL algorithm: consider the following netlist NL_1. Assume that the first gate is the source, and gate area is all 1.

NL_1: $n_1 = a–b–c$, $n_2 = b–d–e–f$, $n_3 = c–f–g$, $n_4 = a\text{-}g$, $n_5 = d\text{-}e–h$, $n_6 = f–h$

a) Model the netlist with an edge-weighted undirected graph G_1. The edge weight is *1/(k-1)* where k is the number of gates, e.g., 1/2 for n_1 and 1/3 for n_2

b) Given an initial partition $P_1 = \{a,b,c,d \mid e,f,g,h\}$ of G_1, perform a single pass of KL algorithm. Give the cutsize after each swap. What are the initial, final, and best cutsizes?

2. FM algorithm

a) Model NL_1 with a non-weighted hypergraph H_1.

b) Given an initial partition P_1 (from 1b) of H_1, calculate the initial cell gain and setup buckets for block 0 and 1.

c) Perform a single pass of FM algorithm based on the area constraint [3, 5] (i.e., min is 3 and max is 5 for either side). Ties should be broken in alphabetical order. Give the cutsize after each move (you don't need to draw the buckets for each move). What are the initial, final, and best cutsizes?

3. hMetis algorithm

a) Perform Edge Coarsening on NL_1 and derive the corresponding coarsened netlist. Name clusters C_i. Visit gates in alphabetical order, and coarsen edges based on its weight. Break ties (with the same weight) based on alphabetical order, i.e., *a-c* before *a-d*.

b) Perform Hyperedge Coarsening on NL_1 and derive the corresponding coarsened netlist. Visit nets based on their weights, and break ties based on their indices, i.e., visit n_4 before n_6

c) Perform Modified Hyperedge Coarsening on NL_1 and derive the corresponding coarsened netlist. Visit nets based on their weights, and break ties based on their indices, i.e., visit n_4 before n_6.

4. FBB algorithm

a) Model NL_1 with a flow network, where $(s,t) = (a,h)$. The edge label should follow flow/capacity format.

b) Compute an augmenting path and the maximum amount of flow that can be pushed on it. Update the flow network after the additional flow.

5. Compare the runtime and space (memory) complexity of the KL, FM, hMetis, and FBB algorithms.