Physical design PA1 report

EE3 B05901172 黃禹靖

* Algorithm Flow

The codes completely follow the Fiduccia-Mattheyses algorithm referred in lecture notes:

1. Initialize the partition
2. Find the cell to move that have the greatest gain, and lock it.
3. Repeatedly do step 2 until all the cells are locked or no legal movement that meet the balance constrain is available.
4. Apply the partition that minimize the cutsize in step 2, 3 if the partition possesses positive gain.

The way to initialize gain and update it also follow which in lecture notes.

In addition, I randomly initialize the partition and repeatedly do the algorithm until there aren’t any improvement 3 times in a row. Then it will write the best solution.

* Data Structure:

I implement a class called “ BucketList ” to attain that operations like finding max gain cell, insertion, remove, and update could be done in constant time.

The “ BucketList “ contains pointers of nodes. And the node structure contains next node pointer, previous node pointer, gain, and cell id. Each nodes represent a cell.

* Discussion

1. There are some pitfalls in input files. Some net would contain two or more same nodes. In this problem, only one node should exist, or it would cause some issues.
2. I have done some check to verify that my code follows the algorithm. But it still performs not so good in input 0 comparing to other’s result. Even though I do randomly partition many times, it cannot perform better than simply divide the cells into first half and second half in the order in parsing. But I can’t find out the reason.