Physical Design of Nanometer ICs PA1 report

B09901089 黃柏穎

0. Collaborator: R12921037 陳韋旭
1. Data structures:

   I didn't use any special data structure in this PA. The implementation of the bucket list follows the given example, which uses std::map and doubly linked list to implement. The only difference is that I use only one bucket list to accommodate all cells instead of two.
2. Findings:

   The first finding is that the initial partition matters a lot. At first, I used a skewed partition by assigning (1-d)/2 (1+d)/2 cells to each part, which tends to have better performance than equal or random partition. After discussed with collaborator, I modified the strategy to first assign all cells to a part, then run a iteration of FM to make it comply with the constraint with least accumulated increase in cut size. Compared to original partition, this heuristic finds the greedy optimum initial partition, the improvement is especially significant in case 0, which has cutsize around 60000 originally and 3000 by the heuristic.

   Second, since original FM-partition is a greedy algorithm, it stuck in local minimum after a few iterations. To further reduce the cut size, a process to add some random perturbation to the solution is inevitable. I stop the FM-partition when the improvement is less than $\alpha * initial\_cutsize$, where $\alpha$ is constant between 1 and 0, and randomly moves some cells to try different possibilities. Each cell records its position when the cutsize reaches the minimum, and revert to it in the end of the process. The number of perturbed cells decreases as the times of perturbation goes up, to make it gradually converge to the minimum. This way we can make FM partition escape local minimum and further reduced the cutsize.