Physical design PA2 report

EE3 B05901172 黃禹靖

* Algorithm Flow

1. It sequentially routes each net in the order of total Manhattan distance of subnets of the net.
2. Subnets of nets are computed by given MST algorithm.
3. For each net, it sequentially routes each subnet.
4. Each subnet is routed by Dijkstra’s algorithm.
5. After one complete routing, it would do rerouting. The first routing can find where is crowded and adjust the weights of edges there, so that the second routing is easier.

* Data Structure:

1. A three-dimension Grid of Nodes.

Node\*\*\* \_Grid

1. A three-dimension array of vector<Edge> to implement adjacent list for each Nodes on Grid.

Vector<Edge>\*\*\* \_Edges

1. A Boolean array “Black” and a Boolean array “Grey” to record whether a node is black or is grey for usage in Dijkstra’s algorithm.

bool\* \_Black

bool\* \_Grey

* Discussion

1. For each net, I sequentially route each subnet. But I want the overlaps of the subnets to be as much as possible, so that it becomes a Steiner tree. To achieve that, after each routing of subnets, I set all the weights of edges on path to zero, and recover the weights after all subnet are routed.
2. Originally, I pack the bool black in class Node to record whether itself is black for Dijkstra’s algorithm. But I find that for each subnet routing ( each Dijksta’s algorithm process ), running initialization to reset all the variable “black” and distance of each nodes would have to iterates through 3-dimension grid, which is extremely slow. So I instead record variable “black“ outside class as a 1-dimesion array, so that it could be reset by memset function. And variable “grey” is similar to “black”, which avoid resetting distance of each node by iterating through all grid.
3. The whole routing process would repeat two times. The idea is from B05901025王鈺能同學. It would normally route the first time, and weights of edges increase during the process. Then reserve the weights as bias and rerouting with same algorithm. It can significant reduce total overflow for the case that over flow is much.
4. For each subnet routing, I choose the pin which is more far from center as source. Because it can slightly reduce searching area in Dijkstra’s algorithm.