Homework 3: Detailed Placement

314553017 蕭希敏

Result

```
grading on 314553017:
 checking item           | status
-------------------------|--------
 correct tar.gz          | yes
 correct file structure  | yes
 have README             | yes
 have Makefile           | yes
 correct make clean      | yes
 correct make            | yes

  testcase |             hpwl |    runtime | status
-----------|------------------|------------|--------
   public1 |         82468400 |       2.14 | success
   public2 |       3608961234 |      47.19 | success
   public3 |      30680521320 |     194.55 | success
   public4 |      41574214425 |     256.16 | success
+-----------------------------------------------------+
|                                                     |
|   Successfully write grades to HW3_grade.csv        |
|                                                     |
+-----------------------------------------------------+
```

Three additional elements

LEF

- **LAYER**: Defines the physical properties of each routing layer on the chip.
- **VIA**: Describes the vertical connections between different metal layers.
- **PROPERTYDEFINITIONS**: A schema section for defining custom attributes not found in the standard format.

DEF

- **TRACKS**: Defines the routing track grid (direction, start, count, pitch, layer) used to align routes on each metal layer.
- **GCELLGRID**: Defines a large, coarse grid used specifically for Global Routing.
- **SPECIALNETS**: Lists critical nets that are not regular logic signals.

Methods

- `runSlidingWindow`:

1. First find all Permutations for cells within the window size.
2. Place them Greedily into the row, skipping any blockages.
3. Calculate HPWL for each permutation.
4. If a permutation yields the best result, replace the current order with that permutation.

- `runGlobalInsertOrSwap`:

1. Calculate the Optimal Region (median X and median Y of connected pins) to get (mx, my).2. Look for rows near my.
2. Try to insert into a spot near mx within those rows.
3. If insertion failed, find a cell of the same size to Swap.

- `runRowNeighborhoodSwap`:(Vertical Swap)

1. Calculate the Optimal Region (median X and median Y of connected pins).
2. Prioritize moving cells with the highest degree.
3. If the Optimal Region is above (or below), search n nearby rows to see if you can Swap or Insert.
4. Once find the valid move and stop the process.

Pseudocode of the flow

```
myPlacer.runGlobalInsertOrSwap();
myPlacer.runSlidingWindow(3); // sliding window = 3
myPlacer.runRowNeighborhoodSwap(min(max_cells, 500000), 3); // swap +-3
rows
myPlacer.runSlidingWindow(2); // sliding window = 2
```

max_cells是cell的數量，若直接每個cell都去做swap，public4會跑不完，故設了最大跑500000 cells

What have you learned from this homework? What problem(s) have you encountered in this homework?

寫完有更了解detail routing怎麼運作，以及體會到legalization上的難點 這次的實作規模又比上次更大許多，要處理的元素更多，尤其Fixed cell處理起來真麻煩，最後寫完後又遇到許多overlap的困擾，這部分就有請到AI去幫忙debug，然後一個de完又有其他bug要de，就是個debug輪迴