CS6135 VLSI PDA HW4

111062684　林鼎勳

1) How to compile and execute your program, and give an execution example:

- Compile: Go into the directory “HW4/src/” and enter the command:

  ```
  $ make
  ```

  This will generate the executable file “hw4” in the directory “HW4/bin/”.

- Execute: In the same directory (“HW4/src/”) and enter the command:

  ```
  $ make test
  ```

  Then, the terminal will let you enter the testcase. We take "adaptec1" for example. This will execute "hw4" and read "adaptec1.aux" under the directory "../testcase/adaptec1/" as input and write the result into "adaptec1.result" in the directory "../output".

  After execution, it will automatically run the verification program to test the result.

  Other available testcases are: adaptec3, ibm01, ibm07, and ibm09.

2) The total displacement, the maximum displacement and the runtime of each testcase:

	Total Displacement (10^6)	Maximum Displacement	Total Runtime (sec)
ibm01	5.70	2146.22	0.10
ibm07	31.12	4835.72	0.57
ibm09	48.70	5130.90	0.88
ada1	7.27	2172.43	16.25
ada3	8.45	2036.50	32.80
	Input Timing (sec)	Abacus Timing (sec)	Output Timing (sec)
ibm01	0.03	0.05	0.02
ibm07	0.14	0.37	0.06
ibm09	0.16	0.65	0.07
ada1	0.91	15.04	0.29
ada3	2.76	29.42	0.62

Screenshot for the result of HW4_grading.sh:

```
[dhlin22@ic51 HW4_grading]$ bash HW4_grading.sh
|-----------------------------------------------|
|                                               |
|    This script is used for PDA HW4 grading.   |
|                                               |
|-----------------------------------------------|
grading on 111062684:
  testcase |  max disp. |     total disp. |    runtime | status
  adaptec1 |    2172.43 |      7266915.00 |      16.38 | Maximum displacement constraint was violated for adaptec1.
  adaptec3 |    2036.50 |      8446082.00 |      32.68 | Maximum displacement constraint was violated for adaptec3.
     ibm01 |    2146.22 |      5699286.50 |       0.21 | success
     ibm07 |    4835.72 |     31119856.00 |       0.66 | success
     ibm09 |    5130.90 |     48701784.00 |       0.97 | success
|-------------------------------------------------|
|                                                 |
|    Successfully generate grades to HW4_grade.csv|
|                                                 |
|-------------------------------------------------|
```

3) The details of your implementation. If there is anything different between your implementation and the algorithm in the ISPD-08 paper, please reveal the difference(s) and explain the reasons:

   The main difference between my implementation and the ISPD-08 paper is in the “PlaceRow” process. In the paper, during each PlaceRow, it calculates the cluster in one row from the beginning. But in my implementation, I recursively modify the previous cluster if the cluster overlaps the cell I’m now trying to place. By doing this, it can reduce the time because it don’t need to calculate the cluster from the beginning each time.

   The second difference is I just check the nearby row or site during finding a minimum-displacement place. In the paper, it goes through all the rows to check the minimum cost. In my implementation, I first find the nearest row and check on that row. Later I go up or down to find if it is possible to find a smaller cost place. By doing this, it can also reduce the time on finding at too far row.

   Another difference is the paper doesn’t need to concern about whether the plane has some fixed blockages.

4) What tricks did you do to speed up your program or to enhance your solution quality?

   Just reveals in the above answer: 1) recursively modify the previous cluster if the cluster overlaps the cell I’m now trying to place and 2) just check the nearby row or site during finding a minimum-displacement place. Besides, I apply early-stop on it: if the vertical distance is larger than the minimum cost I’ve recorded, it will finish the search.

5) Please compare your results with the previous top 5 students' results, and show your advantage either in runtime or in solution quality. Are your results better than theirs?

	Total Displacement (10^6)				
Rank	ibm01	ibm07	ibm09	ada1	ada3
1	5.5	27.86	41.96	3.07	5.08
2	6.1	30.24	47.92	3.42	5.47
3	6.24	31.63	52.21	3.32	5.21
4	6.04	31.11	50.28	3.38	5.27
5	5.57	29.9	46.5	3.69	5.48
Mine	5.70	31.12	48.70	7.27	8.45

In the total displacement part, I did pretty well with the testcases that have no blockages inside. At the same time, they all satisfy the maximum displacement constraint. However, with the testcases having blockages inside, my performance is poor. Until now I don't figure what is the exact problems, but I might change the way to implement the blockage-avoided part.

	Runtime (sec)				
Rank	ibm01	ibm07	ibm09	ada1	ada3
1	0.06	0.23	0.35	2.35	4.17
2	0.07	0.27	0.33	1.74	3.44
3	0.05	0.17	0.2	1.07	3.26
4	0.28	1.22	1.22	12.81	58.14
5	0.10	0.42	0.48	28.21	99.99
Mine	0.10	0.57	0.88	16.25	32.80

In the runtime part, I also did well. This is because I use some tricks to avoid calculating on too far rows or sites.

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

In this homework, I learned that the concept of legalization was easy, but in practice, it takes time to implement and improve performance. Besides, with the fixed blockages involved in some testcases, it also increases the difficulty of this work.