Andy, Yu-Guang Chen

EE6094 CAD for VLSI Design
Programming Assignment 4: Channel Routing
(Due: 23:59:59, 2023/06/11)

Introduction

Channel routing is a challenging problem in the detailed routing during physical design in VLSI. The objective is to minimize the channel area as well as the area of the chip. In standard cell design, the area of channel can be reduced by minimizing the usage of horizontal tracks. Channel is formed by opposite sides of two blocks with the other two sides are open. The top and bottom of channel have terminals which need to be connected. These terminals are the input for routing problem. The routing is done by using horizontal layer for placing horizontal segment and vertical layer for placing vertical segment to connect the given terminals.

There are many objectives of channel routing but it is difficult to satisfy all terminals. There are different algorithms which aim at achieving one or more objectives. The attrition of channel area by utilizing fewer tracks for routing channel is considered here.

In this programming assignment, you are asked to implement a 2-layer channel router to minimize the channel area. You can use Left-Edge algorithms, greedy channel routing algorithm, constraint-graph based algorithm, or any algorithm you find or develop.

Channel Routing Problem in this assignment is defined as follows

Input: Given two rows of terminals

Output: A legal wire segmentation distribution

Objective: Minimize channel area by utilizing fewer tracks for routing channel

Here are some definitions we apply to this programming assignment:

The channel is defined by a rectangular region with two rows of terminals along its top and bottom sides.

- Each terminal is assigned a number between 0 and N. The range of N is [0, 65535].
- Terminals marked by the same label i belong to the same net i.
- Terminals marked by ‘0’ indicates no connection.

The netlist is usually represented by two vectors TOP and BOT.

- TOP(k) and BOT(k) represents the labels on the grid points on the top and bottom sides of the channel in column k, respectively.

The task of the channel router is to:

- Assign horizontal/vertical segments of nets to tracks.

Input file format

The first line gives the terminals of top sides. The second line gives the terminals of bottom sides.

Example:

```
0  1  3  2  11  5  3  1  0
1  5  11 5  1   1  4  2  4
```

Output file format

Format: wire segments of a net

```
.begin    net_n          //wire segments of net n
.H   x    y    x'        //a horizontal wire segment [x,x'] at track(row) y
.V   x    y    y'        //a vertical wire segment [y,y'] at track(col) x
.end
```

Example:

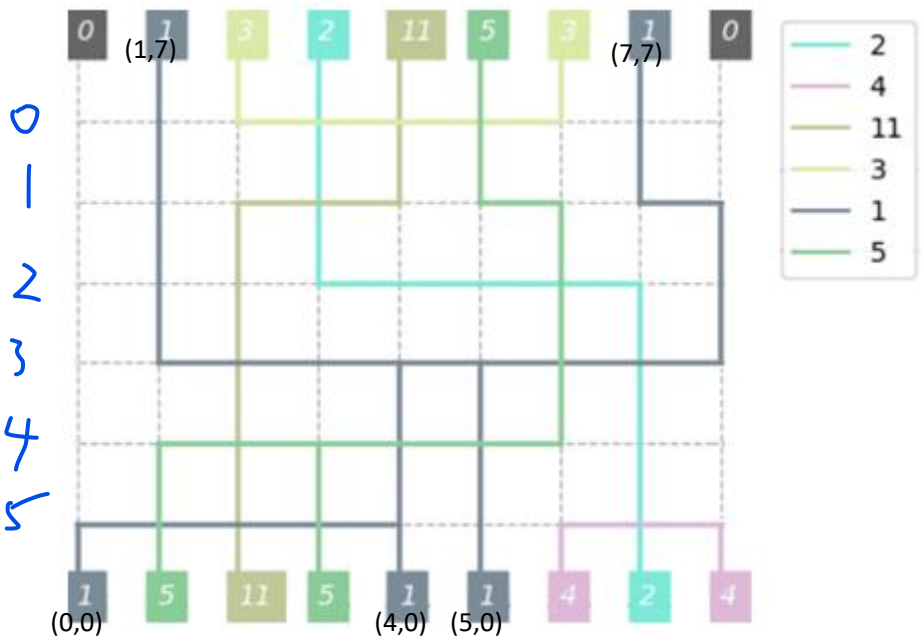
```
.begin 1
.H 1 3 8
.V 1 3 7
.V 8 3 5
.H 7 5 8
.V 7 5 7
.V 4 0 3
.V 5 0 3
.V 0 0 1
.H 0 1 4
.end

   ⋮

.begin 11

   ⋮

.end
```



You can draw the routing results on the screen to make debugging easier. This is not necessary.

Constraints

1. No open error: floating wire segments
2. No short error: overlapping wire segments
3. The number of columns can be used is equal to the number of terminals in a row, and No extra columns can be used. We use the same example as shown above to explain again. In this case, both upper and lower sides have 9 terminals. Therefore, the number of columns is 9. Both upper and lower sides are always having the same number of terminals.
4. The output file should follow the order of the net label. There is no fixed wire segment order.

Requirement

1. You must write this program in C or C++. No open source codes are allowed to use. (i.e., you MUST implement the tool by yourself). You can use any data structure to realize your program. We will verify your program on workstation. Therefore, **you have to make sure your program can be executed correctly and successfully on workstation, any other version of C++ or compilers are not allowed. The run time of your program is limited to at most 1 hours per testcase.** Note that the cout on the screen will slow down the run time. Please use cout carefully.
   The workstation information is shown below:

- System: CentOS 6.10
- Compiler: gcc 4.8.2
- C++ version: C++11

2. We will verify your program on a workstation with a **Makefile**. Therefore, you need to write a **Makefile** which can compile and execute your program directly. Your **Makefile** should at least contain these 3 commands, which are (1) **make all**, (2) **make run**, and (3) **make clean**. The descriptions of each command are shown below.
   (1) **make all**: This command will automatically compile your source codes and generate the corresponding objects and executable file.
   (2) **make run input=testcase1_in.txt output=testcase1_out.txt**: This command will execute your executable file and run your program. Where **testcase1_in.txt** is the input file name, and **testcase1_out.txt** is the output file name.
   (3) **make clean**: This command will automatically remove all the objects and executable file generated by **make all**.

3. All files should be submitted through ee-class. You have to submit **a source code file** named as *StudID*_PA4.cpp (ex: 9862534_PA4.cpp) and **a report** named *StudID_Name*_PA4_report.pdf (ex: 9862534_陳聿廣_PA4_report.pdf), and a **Makefile** to compile and execute your program. If your source code contains more than one file, only the "driver" file that contains main function should follow the naming rule mentioned above. Note that the only acceptable report file format is .pdf, no .doc/.docx or other files are acceptable. **BE SURE to follow the naming rule mentioned above. Otherwise, your program will be not graded.**

4. We don't restrict the report format and length. In your report, you have to at least include:
   (1) How to compile and execute your program. (You can use screenshot to explain)
   (2) The completion of the assignment. (If you complete all requirements, just specify all)
   (3) The hardness of this assignment and how you overcome it.
   (4) Any suggestions about this programming assignment?

Grading

The grading is as follows:

(1) Correctness of your code: 30%
(2) The quality of your solution: 30%
(3) Readability of your code: 10%
(4) The report: 10%
(5) Demo session: 20%

Please submit your assignment on time. Otherwise, the penalty rule will apply:

- Within 24hrs delay: 20% off

- Within 48hrs delay: 40% off
- More than 48hrs: 0 point

Contact

For all questions about PA4, please send E-mail to TA 楊云緯 (ca081820@gmail.com)

Reference

[1] Vinuta H., and Prof. Pavan Kumar E., " Design And Optimization Of Tracks For Channel Routing In Vlsi Physical Design," IOSR Journal of VLSI and Signal Processing (IOSR-JVSP) Volume 6, Issue 3, Ver. II (May. -Jun. 2016), PP 07-11, doi: 10.9790/4200-0603020711

[2] T. . -T. Ho, S. S. Iyengar and S. . -Q. Zheng, "A general greedy channel routing algorithm," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 10, no. 2, pp. 204-211, Feb. 1991, doi: 10.1109/43.68407.