VLSI Design Automation (EECE 6086C)
Ranga Vemuri
Home Work - 2

In this homework, you will implement a design automation algorithm for placement and routing. You will implement one of the three algorithms described below.

This is a group homework. At most two students per group. You may choose your partner. The student with the smaller M number is primarily responsible for placement and the other for routing. You may use any combination of placement and routing algorithms.

The layout area must be a square. It has no I/O pads; that is, all nets are among the cells. The objective of your layout synthesis tool is to minimize the layout area while keeping it a square.

The input to your tool has the following format:

```
First Line: Number of Cells (C)
Second Line: Number of Nets (N)

Next N lines specify the nets
Each net is specified in a line in the following format:

NetNum CellNum TerminalNum CellNum TerminalNum
```

Cells are numbered from 1 to C. Nets are numbered from 1 to N.
Number of terminals on any net is is exactly 2.

Example

```
3
4
1 1 1 1 3
2 1 4 3 1
3 2 3 3 2
4 1 2 2 1
```

It is possible that some terminals are not connected to any net. It is also possible that some cells are not connected to any other cells. Such cells must be retained in the layout.

All logic cells have the same width and height and the same number of pins. Each logic cell has a width of 6 units, height of 6 units has exactly 4 terminals. All terminals are assumed to be available both in Metal 1 and Metal 2 (ie. there is an implicit via at each terminal location). Terminals are assumed to occupy one grid location. The terminals are numbered 1, 2, 3 and 4 and are positioned at grid locations at coordinates (2,6), (5,6), (2,1) and (5,1) relative to the lower left corner of the cell which is assumed to be at (1,1).

The layout surface is a grid. Two metal layers are available for routing. Metal wires must be of 1 unit width and inter-wire spacing must be 1 unit. Assume that the vias are of 1x1 size with no overhang requirement. Minimum separation between the edge of a cell to any wire is 1 unit.

Minimum spacing between any two cells is 1 unit. Over-the-cell routing is not allowed in either layer.

Your layout synthesis tool is allowed to flip or rotate the cells during synthesis. Cells (both logic and feed-through cells) cannot overlap in the final layout.

You are allowed to add any number of "feed-through" cells as you may need. A feed through cell has a width of 3 units, a height of 6 units and has two terminals positioned at (2,6) and (2,1) relative to the lower left corner of the cell. The two terminals are assumed to be internally connected in both layers.

You will receive a set of benchmark net-lists to be placed and routed using your program. Number of cells and nets will be limited to 1000 and 1000 respectively in any benchmark. Benchmark files will be named 1, 2, 3 etc.

Your programs must be in C or C++, compiled and executed and all data gathered on one of the VLSI lab machines.

Your output for each benchmark should be printed in .mag format suitable for viewing with the magic layout editor. (.mag format document is available on blackboard.)

Your submission must be a single .zip file and contain the items described below.

1. Your source code, properly commented and indented for easy readability and submitted as a single file. Include a make file and/or compilation instructions as appropriate.

2. .mag files generated by your tool for each of the benchmarks. Simply name them 1.mag, 2.mag etc. You should label the cells with the cell numbers, the terminals with the terminal numbers and the nets with the net numbers.

3. A description of the placement and/or routing algorithms used, how they interact (if applicable), how they are tuned, choice of data structures and anything else that would explain your algorithms and their implementation.

4. A table showing, for each benchmark, the bounding box dimensions, bounding box area, number of feed-through cells inserted, total wire length, number of vias in the wiring, execution time, memory used and the seed for the random number generator used, if any. Include any other information and experimental data you feel necessary to judge the quality of your algorithm, your software and your effort.

5. Include a snapshot of the magic window for each benchmark result as part of your report.

6. Include a retrospective describing what you think is the cause of the good/poor performance of your tool and how might you be able to improve the performance (both execution speed and quality of result) of your system if you had more time?

7. Finally, include all your source files including a make file.