National Taiwan University
Graduate Institute of Electronics Engineering
Department of Electrical Engineering
Physical Design for Nanometer ICs, Spring 2017

May 5, 2017
Handout #11
Yu-Sheng Lu & Yao-Wen Chang

Programming Assignment #3: Cell Legalization
(due 6pm, May 27, 2017 on-line)

Submission URL:

http://eda.ee.ntu.edu.tw/~yslu/pd17/pa3_submission/

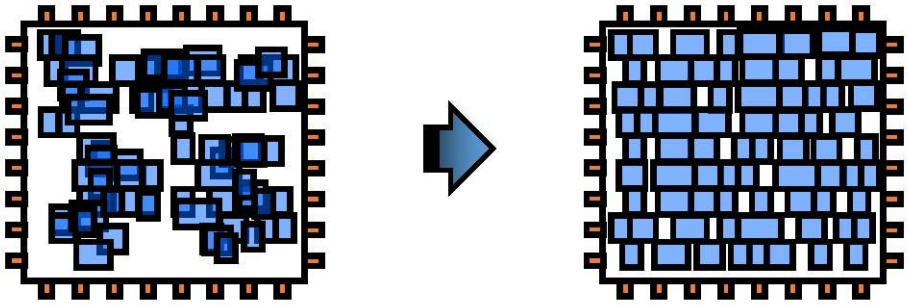
Online Resources:

http://eda.ee.ntu.edu.tw/~yslu/pd17/legalizatoin.tar.gz

1. Problem Statement

This programming assignment asks you to write a legalizer that can remove cell overlaps induced by global placement. Given a global placement result where overlaps still exist (see Figure 1(a)), legalization is a process to eliminate all overlaps by perturbing the modules as little as possible (see Figure 1(b)). The legalization aims at minimizing the total cell displacement while meeting the following constraints:

i. Cells must be placed inside the chip region;
ii. Cells must be located at placement sites on rows;
iii. Cells must be non-overlapping.



(a) (b)

Figure 1. Cell legalization

The total cell displacement D of a set of cell C can be computed by

$$D = \sum_{c_i \in C} (| x'_{c_i} - x_{c_i} | + | y'_{c_i} - y_{c_i} |),$$

where c_i denotes a cell in C, (x_{c_i}, y_{c_i}) denotes the bottom-left cell coordinate of c_i in the

initial placement result, and (x'_{c_i}, y'_{c_i}) denotes the cell coordinate of c_i after legalization.

2. Required Data Structure

```
class Module
{
  public:
  /*get functions*/
    string name();
    double x(); // coordinate of the bottom-left corner
    double y();
    double centerX();
    double centerY();
    double width();
    double height();
    double area();
    unsigned numPins();
    Pin& pin(unsigned index); // index: 0 ~ numPins()-1
  /*set functions*/
    // use this function to set the module position
    Void setPosition(double x, double y);
};
class Net
{
  public:
    unsigned numPins();
    Pin& pin(unsigned index); // index: 0 ~ numPins()-1
};
class Pin
{
  public:
    double x();
    double y();
    unsigned moduleId();
    unsigned netId();
};
class Placement
{
  public:
    double boundaryTop();
    double boundaryLeft();
    double boundaryBottom();
    double boundaryRight();
    Module& module(unsigned moduleId);
```

```
    Net& net(unsigned netId);
    Pin& pin(unsigned pinId);
    unsigned numModules();
    unsigned numNets();
    unsigned numPins();
    double computeHpwl();// compute total wirelength
    // output global placement result file for later stages
    outputBookshelfFormat(string fileName);
};
```

The functions above are used to access the data required at the legalization stage. You should use the function `setPosition(double x, double y)` to update the coordinates of modules (to move them). At the same time, the coordinates of pins on modules will be updated accordingly. Note that modules cannot be placed out of the chip boundaries.

3. Compilation & Execution

To compile the program, simply type:

```
make
```

Please use the following command line to execute the program:

```
./legalizer -aux inputFile.aux
```

For example:

```
./legalizer -aux ibm01-cu85.aux
```

Any output format error will incur severe penalty.

4. Language/Platform

- Language: C or C++. **Please specify your programming language.**
- Platform: Linux. **Please develop your programs on servers in the EDA Union Lab**

5. Submission

You need to submit the following materials in a .tar or a .zip file (e.g., f04943094-p3.zip) at the course submission website by the deadline: (1) all source codes in the src/ directory, (2) executable binary named as legalizer, (3) a text readme file named as readme.txt, and (4) a report named as report.doc on the algorithm used in your program.

6. Evaluation

This programming assignment will be graded based on the (1) correctness of the program, (2) solution quality, (3) running time (restricted to 30 minutes for each case), (4) report.doc and (5) readme.txt. Please check these items before your submission.

7. Online Resources

Sample codes, benchmarks, readme.txt, and report.doc can be found at the submission website.

8. References

You may refer to these previous works to find some algorithms for legalization.

[1] S. Chou and T.-Y. Ho, "OAL: An obstacle-aware legalization in standard cell placement with displacement minimization," in *Proceedings of the 2009 IEEE International SOC Conference (SOCC)*, pp. 329–332, 2009.

[2] N. K. Darav, I. S. Bustany, A. Kennings, L. Behjat, "A fast, robust network flow-based standard-cell legalization method for minimizing maximum movement," in *Proceedings of the ACM International Symposium on Physical Design (ISPD)*, pp. 141–148, 2017.

[3] D. Hill, "Method and system for high speed detailed placement of cells within integrated circuit designs," *U.S. Patent 6370673*, 2002.

[4] P. Spindler, U. Schlichtmann, and F. M. Johannes, "Abacus: Fast legalization of standard cell circuits with minimal movement," in *Proceedings of the ACM International Symposium on Physical Design (ISPD)*, pp. 47–53, 2008.