CS6135 VLSI Physical Design Automation

Homework 4: Placement Legalization

Due: 23:59, December 16 2019

---

1. Introduction

In this homework, you are asked to implement an existing algorithm, published in the ISPD-08 paper "**Abacus: fast legalization of standard cell circuits with minimal movement**" by Spindler, Schlichtmann and Johannes to legalize a given global placement result with minimal displacement.

2. Problem Description

(1) Input:

- A set of standard cells (and blockages), where each standard cell (or blockage) has a rectangular shape specified by its width and height. The design is composed of single-row height movable cells, and multiple-row height fixed blockages.
- The original coordinates of each cell (or blockage).
- Chip specification, such as the coordinates of each row, the uniform row height, the width of sites, etc.

(2) Output:

- The coordinates of each cell (or blockage) after legalization is done.

※ The coordinates of each cell or blockage is specified by its lower-left corner.

(3) Objective:

Cells are not allowed to be rotated, but they are only allowed to be moved instead. The total displacement of the legalization result and the runtime of you program should be as small as possible subject to the following constraints.

1. Aligning constraint: Each cell is not allowed to cross multiple rows, and must align its left boundary with a site of the corresponding row where the cell locates.
2. Non-overlapping constraint: No two cells overlap with each other.

3. Input File

(1) The *.aux* file:

The .aux file specifies the filenames which contain the design information you should know. Here is an example:

```
RowBasedPlacement : adaptec1.nodes adaptec1.ntup.pl adaptec1.scl
// RowBasedPlacement : .nodes file .pl file .scl file
MaxDisplacement : 120
// MaxDisplacement : times of the site width
```

- *MaxDisplacement* is a maximum displacement threshold for each cell. If your program can satisfy this constraint, you will get bonus points; if not, you will not get any penalty.

(2) The *nodes* file:

The .nodes file specifies the name, width, height and the other information about each cell/blockage node in the global placement result. Each line specifies a single cell/blockage node.

Here is an example:

```
UCLA nodes 1.0
# File header with version information, etc.
# Anything following '#' is a comment, and should be ignored.

NumNodes :    211447
//NumNodes : number of cells and blockages
NumTerminals :    543
//NumTerminals : number of blockages

    o0    8     12
    o1    500   2136      terminal
    //nodeName    width      height      moveType
     ⋮
```

- *nodeName* is an arbitrary-length alpha-numeric string, and is case-sensitive.
- *moveType* is an optional string. If not given, it means a movable cell; *terminal* means a fixed blockage.
- *width* is a multiple of the site width.
- *height* is a multiple of the row height.

(3) The *.pl* file:

The .pl file specifies the original coordinates of each cell/blockage in the global placement result. Here is an example:

```
UCLA pl 1.0

o0    6295    4107 : N
o1    3064    8619 : N /FIXED
//nodeName x-coordinate y-coordinate : orientation (always N) moveType
      ⋮
```

(4) The *.scl* file:

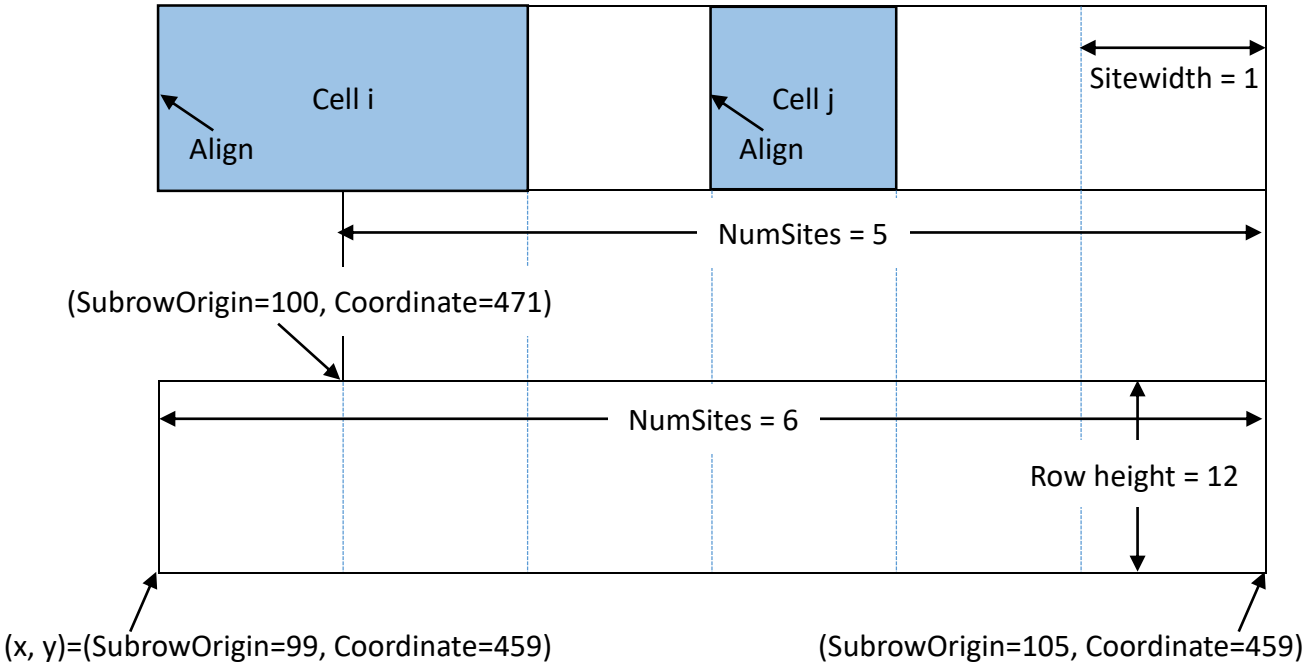
The .scl file specifies the information of rows in the placement region. Here is an example:

```
UCLA pl 1.0
# File header with version information, etc.
# Anything following '#' is a comment, and should be ignore.

NumRows :    890
//NumRows :    number of rows

CoreRow Horizontal //1st row
   Coordinate      :    459
   //Coordinate    :    y-coordinate of row
   Height          :    12
   //Height        :    row height
   Sitewidth       :    1
   //Sitewidth     :    width of a site
   Sitespacing     :    1
   //Sitespacing   :    spacing between two sites
   Siteorient      :    1
   Sitesymmetry    :    1
   SubrowOrigin    :    99    NumSites :    6
   //SubrowOrigin  :    x-coordinate of row NumSites :    number of sites
End
      ⋮
```

- *Coordinate* represents the y-coordinate of the bottom boundary of the row.
- *Sitewidth* and *Sitespacing* are two identical integers. They both represent the width of a site.
- *SubrowOrigin* represents the x-coordinate of the left boundary of the row.



4. Output File

(1) The *.result* file:

The .result file specifies the legalization result containing the coordinates of each cell (or blockage), whose format is the same as the input .pl file. And the filename is the same as the .aux file. Here is an example:

```
UCLA pl 1.0

o0    6295    4119 : N
o1    3064    8619 : N /FIXED
      ⋮
```

5. Language/Platform

(1) Language: C/C++

(2) Platform: Unix/Linux

6. Report

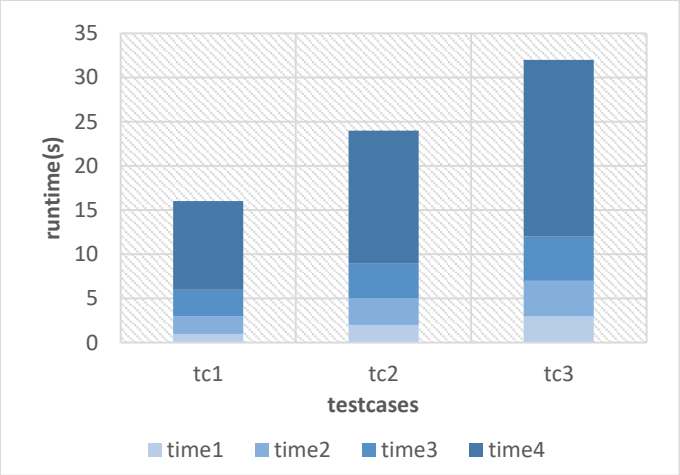
Your report must contain the following contents, and you can add more as you wish.

(1) Your name and student ID

(2) How to compile and execute your program and give an execution example.

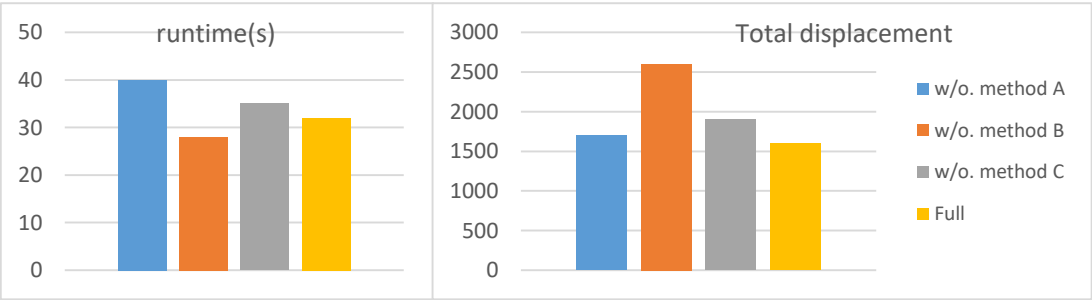
(3) The total displacement, the maximum displacement and the runtime of each testcase.

Notice that the runtime contains I/O, constructing data structures, computing parts, etc. The more details your experiments have, the more clearly you will know where the runtime bottlenecks are. You can plot your results like the one shown below.



(4) 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.

(5) What tricks did you do to speed up your program or to enhance your solution quality? Also plot the effects of those different settings like the ones shown below.



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

7. Required Items

Please compress `HW4/` (using tar) into one with the name `CS6135_HW4_${StudentID}.tar.gz` before uploading it to iLMS.

(1) `src/` contains all your source code, `Makefile` and `README`.
- `README` must contain how to compile and execute your program. An example is like the one shown in HW2.

(2) `output/` contains all your outputs of testcases for the TA to verify.

(3) `bin/` contains your executable file.

(4) `CS6135_HW4_${STUDENT_ID}_report.pdf` contains your report.

You can use the following command to compress your directory on a workstation:

```
$ tar -zcvf CS6135_HW4_{StudentID}.tar.gz <directory>
```

For example:

```
$ tar -zcvf CS6135_HW4_108062500.tar.gz HW4/
```

8. Grading

- ✓ 70 ~ 80%: The solution quality (total displacement) and the runtime of each testcase; hidden testcases included.
- ✓ 20 ~ 30%: The completeness of your report.
- ✓ **5% Bonus**: Parallelization. Please specify your system specification.
- ✓ **10% Bonus**: Maximum displacement thresholds.

※ If the aligning constraint is violated, your grade will be set to 0; if the non-overlapping constraint is violated, your grade will not be set to 0, but will get a strong penalty.

※ The executable filename must be named as *hw4*.

※ Please make sure the following command format can run your program.

```
$ hw4 *.aux
```

E.g.: `$ hw4 ../testcase/adaptec1/adaptec1.aux`

※ Program must be terminated within 10 minutes for each testcase.

Notes:

- Grading of the result of each testcase is based on the total displacement (80%) and runtime (20%).