Lab4 – Legalization with Minimal Displacement

Deadline: 23:59 Dec 27th, 2021

Lab4 Introduction

This programming assignment asks you to write a **legalizer with Minimum Displacement**. Given a chip design and cell information with a global placement result. Align all standard cells to feasible rows. Legalize all standard cells without overlap. Minimize the movement of cells between global placement and legalized placement

Input Format (ISPD Placement Benchmark Format)

1. input.aux (Contains all other file)
2. input.pl (Description of Placement)
3. input.scl (Description of Chip)
4. input.node (Description of Node Dimension)

Input.pl

1. Describes the **original position** of node
2. For each node

[Node_name] [lower_left_X_coordinate] [lower_left_Y_coordinate] : [orientation] [moveType]

A. Orientation will always be N

input.pl

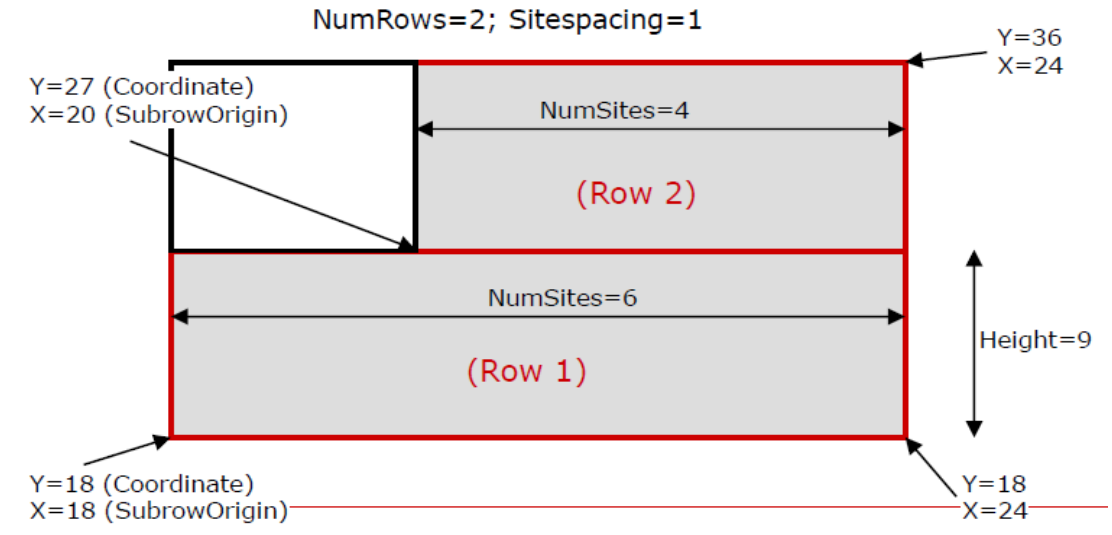
```
UCLA pl 1.0
# File header with version information, etc.
# Anything following "#" is a comment, and should be ignored

o0 0 0 : N
o1 0 0 : N
o2 0 0 : N
o3 7831 7452 : N /FIXED
p0 1215 7047 : N /FIXED_NI
```

Input.scl

1. NumRows : Number of circuit rows for placement
2. CoreRow – Horizontal circuit row followed by the row specification
3. Coordinate : **Y**-coordinate of the bottom edge of the circuit row
4. Height : Circuit row height (= standard-cell height)
5. Sitespacing : Absolute distance between neighboring placement sites in a row
6. SubrowOrigin : **X**-coordinate of the left edge of the subrow
7. NumSites : Number of placement sites in this subrow
8. Hence, **X**-coordinate of the right edge of the subrow = **SubrowOrigin + NumSites*Sitespacing**

input.scl

```
UCLA scl 1.0
# File header with version information, etc.
NumRows : 1
CoreRow Horizontal
Coordinate : 18
Height : 9
Sitewidth : 1 # optional: equal to Sitespacing
Sitespacing : 1
Siteorient : N # optional: can be ignored
Sitesymmetry : Y # optional: can be ignored
SubrowOrigin : 18 NumSites : 11605
End
```



Input.node

1. Number of terminal = terminal + #terminal_NI
2. For each node
   [Node_name] [width] [height] [moveType]

   A. MoveType
      i. If a line does not specify a movetype, the associated node is a movable node
      ii. **Terminal** – this node can not be moved or overlap with other node
      iii. **Terminal_NI –** this node can not be moved but overlap is allowed

input.node

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

NumNodes : 5
NumTerminals : 2

o0 4 9 # movable node
o1 6 9
o2 24 9
o3 414 2007 terminal # terminal node (fixed node)
p0 1 1 terminal_NI # terminal_NI node (fixed node, but overlap
is allowed with this node
```

Explanation of the Input

1. Three input benchmark is needed during placement
2. A parser is written for you; you may choose to use them. Please note that the provided parser is provided at your ease, **it does not guarantee to be bug free. USE IT AT YOUR OWN RISK**

Required Output

1. Same with input.pl except you need to decide new coordinate for cells
2. The required output must pass the verifier

output.pl

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

o0 0 0 : N
o1 0 0 : N
o2 0 0 : N
o3 7831 7452 : N /FIXED
p0 1215 7047 : N /FIXED_NI
```

Algorithm

Processes in SimPL, Simplified Tetris algorithm, Abacus, use bipartite matching to alleviate congestion …etc

Naming Rule

1. Name of binary: Lab4
2. Name of output: output.pl

Executing Procedure

1. Compile (Please describe how to compile your file in readme)
2. **./Lab4 [input.aux] (e.g. ./Lab4 adaptec1.aux**
3. Search for output.pl, if not found→ break →enter 0 point
4. **./ Lab4_verifier adaptec1.aux output.pl**
5. If pass→run Lab4_evaluate, else →break→enter 0 point
6. **./ Lab4_evaluate adaptec1.pl output.pl**

Submission

Please submit the following materials in a .zip file to E3 by the deadline, specifying your student ID in the subject field (e.g., StudentID.zip):

(1) Source codes (.cpp, .h …)

(2) Makefile

(3) Executable binaries (Lab4)

A text readme file (readme.txt), stating how to build and use your programs.

Grading Criteria – (Use Highest Score as Reference)

1. **Displacement**
2. **Run-Time**

   We will determine your score according to the displacement result and the run time. (Priority: displacement result > run time)
3. For each case, the run time limit is up to 300 seconds. It will be regarded as "failed" if you use more than 300 seconds.
4. Please make sure your code is available on our linux server. If it cannot be executed, you will get zero point.
5. Accept four days late submission, 10% deduction per day. That is, if you hand in on 12/28, the score will be *0.9; if you hand in on 12/31, the score will be *0.6, and submission will not be accepted after 12/31.

Also, a script on checking similarity for codes will be performed. If two codes exist high similarity. Both codes receive zero mark after confirmation.