

## Lab3 – Legalization with Minimal Displacement

**Deadline: 23:59 Dec 30<sup>th</sup>, 2018**

### Lab3 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

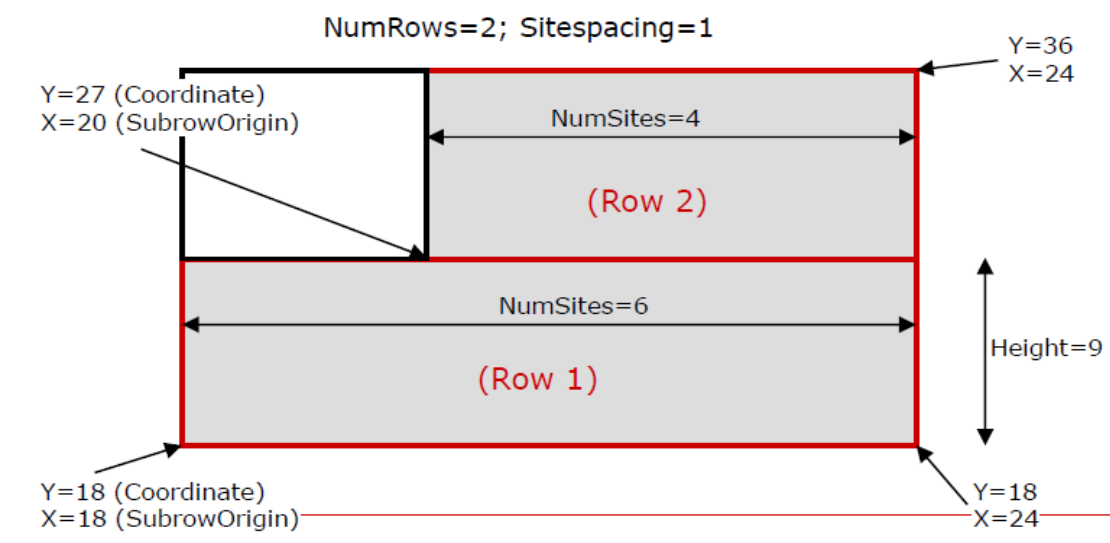
o0 0 0 : N
o1 0 0 : N
o2 0 0 : N
o3 7831 7452 : N /FIXED
p0 1215 7047 : N /FIXED_NI
```

### 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

```

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: Lab3
2. Name of output: output.pl

### Executing Procedure

1. Compile (Please describe how to compile your file in readme)
2. ./Lab3 [input.aux] (e.g. ./Lab3 adaptec1.aux
3. Search for output.pl, if not found → break → enter 0 point
4. ./ Lab3\_verifier adaptec1.aux output.pl
5. If pass → run Lab3\_evaluate, else → break → enter 0 point
6. ./ Lab3\_evaluate adaptec1.pl output.pl

### Grading Criteria – (Use Highest Score as Reference)

1. Displacement
2. Run-Time