Lab3 – Fixed Outline Floorplanning

Deadline: 23:59 Nov. 29, 2020

Lab3 Introduction

This programming assignment asks you to write a **fixed-outline floorplanner** that can handle hard macros. Given a set of rectangular macros and a set of nets, the floorplanner places all macros within a rectangular chip without any overlaps. We assume that the lower-left corner of this chip is the origin (0, 0), and no space (channel) is needed between two different macros. The objective is to minimize the area of chip bounding box and the total net wirelength. The total wirelength W of a set N can be computed by

$$W = \sum_{n_i \in N} HPWL(n_i)$$

where n_i denotes a net in N, and $HPWL(n_i)$ denotes the half-perimeter wirelength of n_i. The objective for this problem is to minimize

$$Cost = \alpha A + (1 - \alpha)W$$

where A denotes the bounding-box area of the floorplan, and α, $0 \leqq \alpha \leqq 1$, is a user defined ratio to balance the final area and wirelength. Note that a floorplan which cannot fit into the given outline is not accepted.

Input

Each test case has two input files, *input.block* and *input.nets*. The first file (*input.block*) gives the **outline size**, the **number of blocks**, and the **number of terminals** defined in this file. Then the **block dimensions** are listed, followed by the terminal locations. The file format is as follows:

```
Outline: <outline width, outline height>
NumBlocks: <# of blocks>
NumTerminals: <# of terminals>

<macro name> <macro width> <macro height>
… More macros

<terminal name> terminal <terminal x coordinate> <terminal y coordinate>
… More terminals
```

Figure 1 Input file format (*input.block*).

The second file (input.net) gives the **number of nets** in the floorplan, followed by the **terminal information for each net**. The file format is as follows:

```
NumNets: <# of nets>
NetDegree: <# of terminals in this net>
<terminal name>
… More terminal names

… More "NetDegree" and "terminal name"
```

Figure 2 Input file format (*input.nets*)

The user-defined ratio α is given through the command-line argument. It ranges between 0 and 1.

Output

The output file (*output.rpt*) records the problem output. This report consists of six parts: (1) the final cost, (2) the total wirelength, (3) the chip area, (4) the chip width and height, (5) the runtime in seconds, and (6) the bounding-box coordinate for each macro (specified by the lower-left corner and upper-right corner). The report file format is shown below.)

<final cost>
// $Cost = \alpha A + (1-\alpha)W$
<total wirelength>
// $W = \sum_{n_i \in N} HPWL(n_i)$
<chip_area>
// area = (chip_width) * (chip_height)
<chip_width> <chip_height>
//resulting chip width and height
<program_runtime>
//report the runtime in seconds
<macro_name> <x1> <y1> <x2> <y2>
<macro_name> <x1> <y1> <x2> <y2>
// (x1, y1): lower-left corner, (x2, y2): upper-right corner

Figure 3 Output file format (*output.rpt*)

Note: All the x&y coordinates should be integer, only the cost is double/float.

Command-Line Parameters

In order to test your program, you are asked to add the following command-line parameters to your program (Please name your execution file name as "lab3"):

./lab3 [α value] [input.block name] [input.net name] [output name]

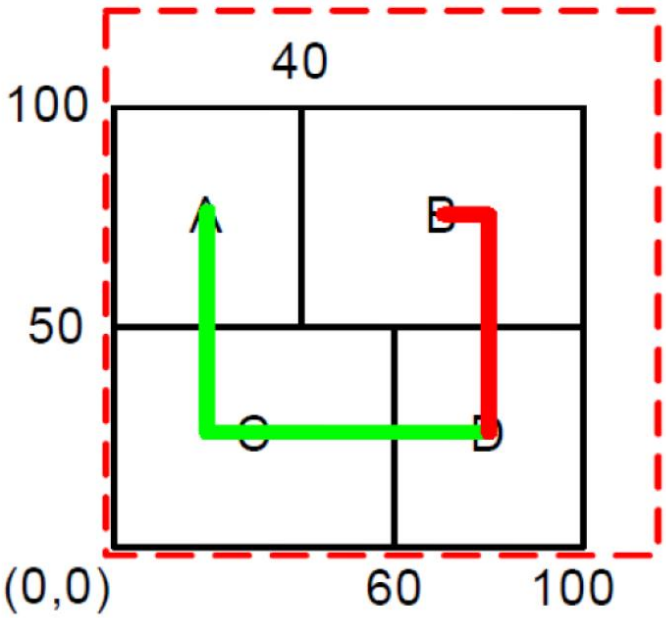
For example, "**./lab3 0.5 input.block input.nets output.rpt**".

Example

Figure 4 illustrates an example of the IO files (assume α = 0.5):

Input files (input.block):

```
Outline: 120 120
NumBlocks: 4
NumTerminals: 0
A    40   50
B    60   50
C    60   50
D    40   50
```



(input .nets)

```
NumNets: 2
NetDegree: 3
A
C
D
NetDegree: 2
B
D
```

Output files (output.rpt)

```
5085
170
10000
100  100
0.24
A    0    50   40   100
B    40   50   100  100
C    0    0    60   50
D    60   0    100  50
```

Output files (output.rpt)

```
5085                    Cost
170                     HPWL
10000                   Area
100  100                Width & Height
0.24                    Runtime
A    0    50   40   100
B    40   50   100  100
C    0    0    60   50
D    60   0    100  50
```

Figure 4 A floorplanning problem and its solution

HPWL (Half-Perimeter Wire Length): The smallest square such that all net pins are inside. Use the **center point** of the Macro as its pin (round down to integer) as shown in Figure 5.

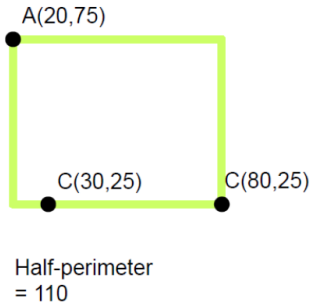


Figure 5 Example of HPWL

Language/Platform

Language: C or C++

Platform: Linux. (Please make sure your code is available on our linux server. If it cannot be executed or .zip file cannot be unzipped, you will get zero point!!)

Grading

This programming assignment will be graded based on:

(1) Correctness of the program (no error, all blocks are in the fixed outline)

(2) Solution quality (minimize the cost)

(3) Running time (at most 300 secs)

(4) readme.txt. Please check these items before your submission.

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

The primary objective is to **place all blocks into the fixed outline successfully** and to minimize the **cost**.

Submission

Please submit the following materials in a .tar or a .zip file to E3 by the deadline, specifying your student ID in the subject field (e.g., studentID_lab3_v1.zip):

(1) Source codes (.cpp, .h …)

(2) Executable binaries (lab3)

(3) A text readme file (readme.txt), stating how to build and use your programs. If you have updated version, please name your newer version by v2, v3, …, etc.

Resources

Sample input files (ami33.block/ami33.nets) and a verifier are provided for your convenience.