# CS6135 VLSI Physical Design Automation

# Homework 3: Fixed-outline Slicing Floorplan Design


# 112062682 張宇越


## 1. How to compile and execute my program :

- **Compile:**

  Enter HW3/src/ and make, it'll generate the executable file to HW3/bin/

  $ cd HW3/src/

  $ make

- **Execute:**

  $ cd HW3/bin/

  $ ./hw3 <testcase file> <output file> <dead space ratio>

  E.g.

  $ ./hw3 ../testcase/public1.txt ../output/public1.out 0.1

## 2. The screenshot of the result of running the HW3_grading.sh:

```
[g112062682@ic22 ~/HW3_grading]$ ./HW3_grading.sh
+-----------------------------------------------+
|                                               |
|    This script is used for PDA HW3 grading.   |
|                                               |
+-----------------------------------------------+
host name: ic22
compiler version: g++ (GCC) 7.3.0

grading on 112062682:
 checking item          | status
------------------------|--------
 correct tar.gz         | yes
 correct file structure | yes
 have README            | yes
 have Makefile          | yes
 correct make clean     | yes
 correct make           | yes

  testcase |      ratio | wirelength |    runtime | status
-----------|------------|------------|------------|--------
   public1 |       0.15 |     191291 |     158.27 | success
   public2 |       0.15 |     408116 |     580.04 | success
   public3 |       0.15 |     601857 |     580.04 | success
   public1 |        0.1 |     205268 |     164.15 | success
   public2 |        0.1 |     427251 |     580.04 | success
   public3 |        0.1 |     629403 |     580.04 | success
   public1 |       0.09 |     214174 |     185.50 | success
   public2 |       0.09 |     439040 |     580.04 | success
   public3 |       0.09 |     623780 |     580.05 | success
+-----------------------------------------------+
|                                               |
|   Successfully write grades to HW3_grade.csv  |
|                                               |
+-----------------------------------------------+
```

## 3. Please show that how small the dead space ratio could be for your program to produce a legal result:

第二點的圖片中,可以達到的最小的 dead space ratio 是 0.09

## 4. The details of your implementation:

### (1) 讀入參數與初始化

- 由 main() 讀取三個參數：輸入檔路徑、輸出檔路徑、以及 dead space ratio (ds_ratio)。
- 建立 Floorplanner fp;，並設定 fp.dead_space_ratio = ds_ratio。
- 呼叫 fp.readInput(inFile) 讀入 Blocks、Pads、Nets。
- 呼叫 fp.set_seed()，根據測資大小與 dead space ratio 選擇固定種子，否則以 time(NULL) 取得隨機種子，再用 srand(seed) 設定全域亂數。

### (2) 計算外框 (Fixed-outline)

- fp.calcOutline()：

1. 計算 total_block_area * (1 + dead_space_ratio)。
2. 取其平方根向下取整，並同時設為 outlineW、outlineH。

### (3) 產生初始解

- vector<string> expression = fp.initSolution();
- 將所有 block 按面積(w*h)由大到小排序，且預先將寬小於高的 block 做一次 90° 旋轉。
- 若當前 row 寬度加上下一個 block 超過 outlineW，就新開一個 row。
- 最後以 slicing floorplan 的 Polish expression (在同一 row 以 V 連接，不同行以 H 連接) 回傳。

### (4) 第一階段：Simulated Annealing For Area 求可行解

- 以 fp.simulatedAnnealing(expression, false, …) 進行 SA：

1. 隨機產生鄰域解 ( swap_adjacent_operand 、 invert_chain 、 swap_random_operand)。
2. 計算新解成本：fp.getCost(neighbor, false)，只考量超出 outline 的懲罰。
3. Metropolis 準則接受或拒絕，並根據冷卻係數更新溫度。

- 一直重複, 直到找到零懲罰 (cost==0) 或超過時間限制。

(5) **第二階段：Simulated Annealing For Wirelength 求可行解**

- 將 fp.best_blocks = fp.blocks; 保留第一階段最優排列，計算 fp.getWirelength()。
- 以剩餘時間呼叫 fp.simulatedAnnealing(expression, true, …)，在滿足 outline 內的前提下最小化 penaltyFactor * area + wirelength，並最終輸出最佳線長。

(6) **結果輸出**

- fp.writeOutput(outFile)：輸出總線長、block number，及每個 block 的左下角座標與旋轉狀態 (0/1)。

## 5. Different between your implementation and the algorithm in the DAC-86 paper :

### a. 初始解生成策略調整

paper 的初始解是把所有 block 按編號依序接成「12V3V…nV」的 Polish Expression，但這種做法常常造成多數 block 跑出 outline 之外，讓第一階段 SA 花費過多時間才找到可行解。

為了改善，我改用「寬度累加／換行」的方式：

- 先將 block 依面積由大到小排序。
- 依序把 block 丟到當前 row，若累計寬度超過 outlineW，就自動換到下一行。
- 每列內以 V 連接，列與列之間以 H 連接，形成初始 Polish expression。

這樣產生的初始解大部分 block 都已置於 outline 內，更貼近可行解，也能大幅縮短 SA 找到可行解的時間。

### b. 雙階段 Cost Function 設計

paper 直接把 area + wirelength 同時當成 SA 的 cost function，結果在解還沒合法（出界）之前，就一直在計算線長，造成大量不必要的運算開銷。

我改成兩階段處理：

1. SA for Area：只用面積懲罰（超出 outline 的 area penalty）當 cost，專注找出合法可行的排放方式；
2. SA for Wirelength：在第一階段得到的可行解上，再將 area +

wirelength 作為 cost，進一步最小化線長。

這樣做能避免在第一階段就不斷計算線長，在 block 都還沒塞進 outline 時浪費時間，也能加快整體收斂速度。

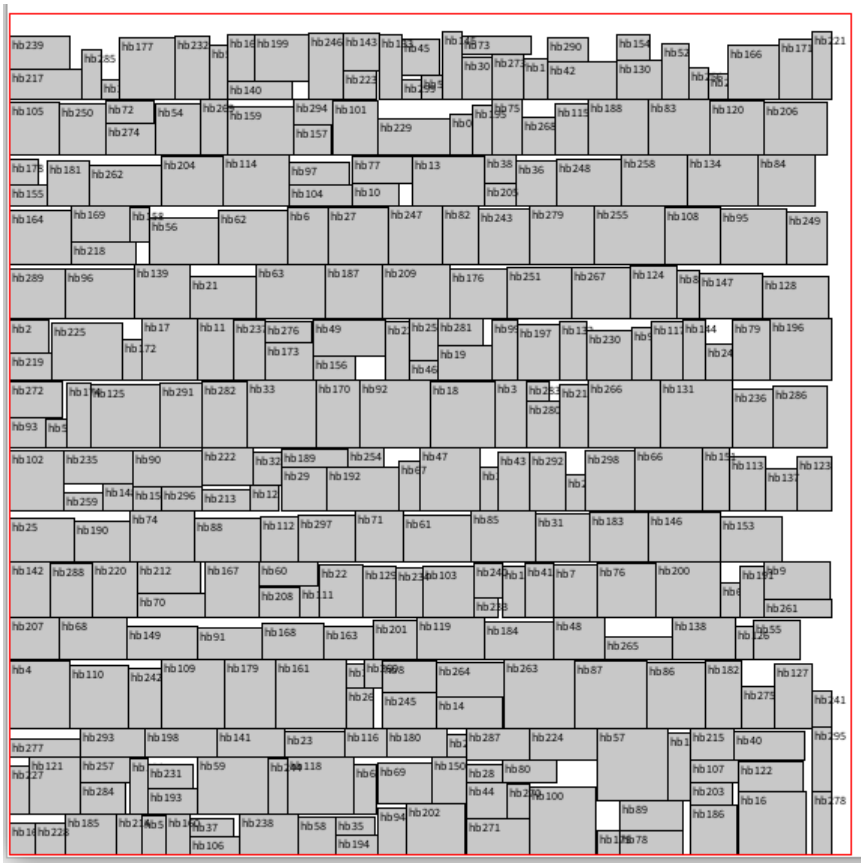
## 6. Please describe your method of your initial floorplan:

- 將所有 block 按面積（w*h）由大到小排序，且預先將寬小於高的 block 做一次 90° 旋轉。
- 若當前 row 寬度加上下一個 block 超過 outlineW，就新開一個 row。
- 最後以 slicing floorplan 的 Polish expression（在同一 row 以 V 連接，不同行以 H 連接）回傳。

## 7. What tricks did you do to speed up your program or to enhance your solution quality ?

用第六點的方式做 initial floorplan 時, 若沒有先做第一步(對 block 做前置處理), 會有很多測資超出 outline, 無法得到合法的結果。

以下是沒有優化,在 public3.txt 使用 dead space ratio = 0.15 的合法結果:



以下是沒有優化, 很多測資會超出 outline 的結果:

```
[g112062682@ic22 ~/HW3_grading]$ ./HW3_grading.sh
+-----------------------------------------------+
|                                               |
|    This script is used for PDA HW3 grading.   |
|                                               |
+-----------------------------------------------+
host name: ic22
compiler version: g++ (GCC) 7.3.0

grading on 112062682:
 checking item          | status
------------------------|--------
 correct tar.gz         | yes
 correct file structure | yes
 have README            | yes
 have Makefile          | yes
 correct make clean     | yes
 correct make           | yes

  testcase |      ratio | wirelength |    runtime | status
-----------|------------|------------|------------|--------
   public1 |       0.15 |     194309 |     183.86 | success
   public2 |       0.15 |        N/A |        N/A | There is an error in the output results of public2 ([Error] Constraint
Violated! Hard block "hb198" overlaps with hard block "hb199".).
   public3 |       0.15 |     622798 |     580.06 | success
   public1 |        0.1 |     200791 |     174.05 | success
   public2 |        0.1 |        N/A |        N/A | There is an error in the output results of public2 ([Error] Constraint
Violated! Hard block "hb198" overlaps with hard block "hb199".).
   public3 |        0.1 |        N/A |        N/A | There is an error in the output results of public3 ([Error] Constraint
Violated! Hard block "hb298" overlaps with hard block "hb299".).
   public1 |       0.09 |     205981 |     174.30 | success
   public2 |       0.09 |        N/A |        N/A | There is an error in the output results of public2 ([Error] Constraint
Violated! Hard block "hb198" overlaps with hard block "hb199".).
   public3 |       0.09 |        N/A |        N/A | There is an error in the output results of public3 ([Error] Constraint
Violated! Hard block "hb298" overlaps with hard block "hb299".).
+-----------------------------------------------+
|                                               |
|   Successfully write grades to HW3_grade.csv  |
|                                               |
+-----------------------------------------------+
```

## 8. What have you learned from this homework ?

第一次從頭撰寫 SA，深入理解了溫度（Temperature）、能量差（Δcost）、Metropolis 准則，以及如何設計冷卻排程（cooling schedule）。

實際測試多組初溫、最低溫、冷卻係數和迭代次數，瞭解到 SA 收斂速度與解品質高度依賴這些參數的平衡。

因為是第一次使用 SA，特別注意了 seed 的選擇與 srand()，才能保證每次執行結果可重現，方便撰寫與除錯。

將問題拆成「先只優化面積可行性、再優化線長」兩個階段，相比一次打包全部成本，有效降低不必要的計算開銷。

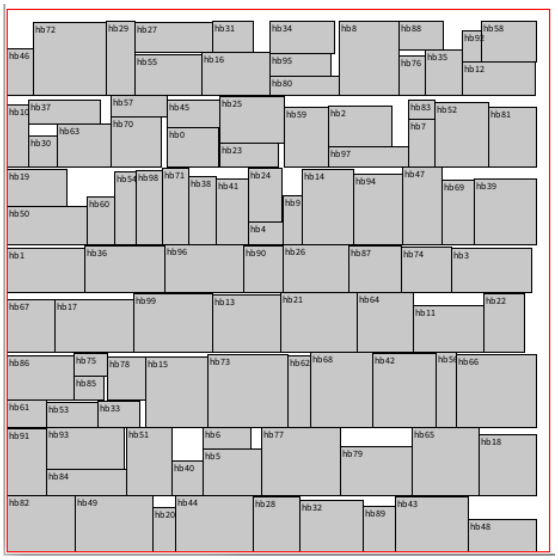
## 9. What problem(s) have you encountered in this homework ?

初次實作 Metropolis 判斷時，常常因為溫度設太低或冷卻太快，導致早期卡在局部最優；反之溫度太高又太慢收斂，需要反覆調參。

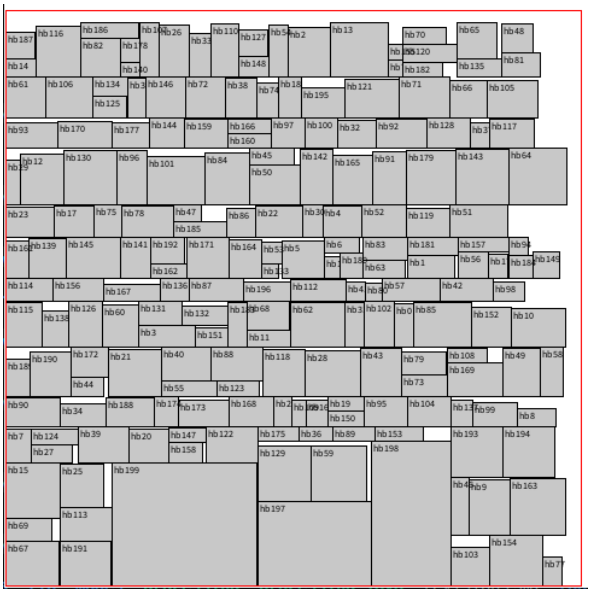
一開始直接在同一次 SA 中同時計算 area+wirelength，結果在線長還沒有意義之前就花了大量時間，後來改成雙階段才改善效能。

## 10. All testcase result:

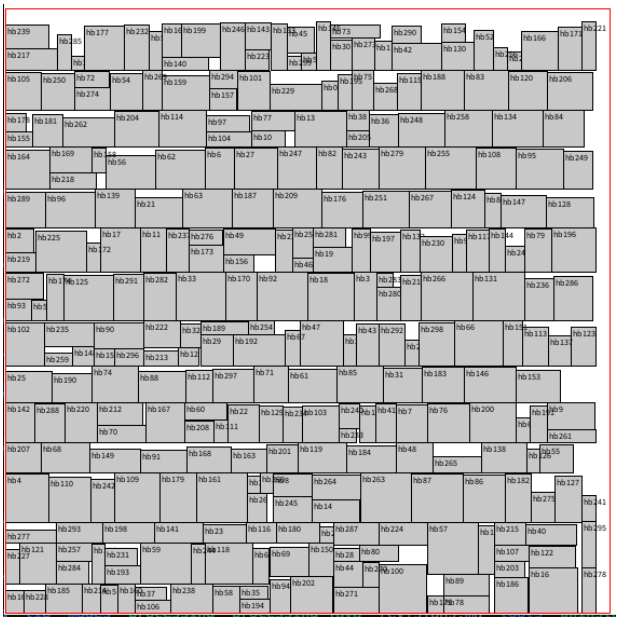
- public1.txt dead space ratio = 0.15



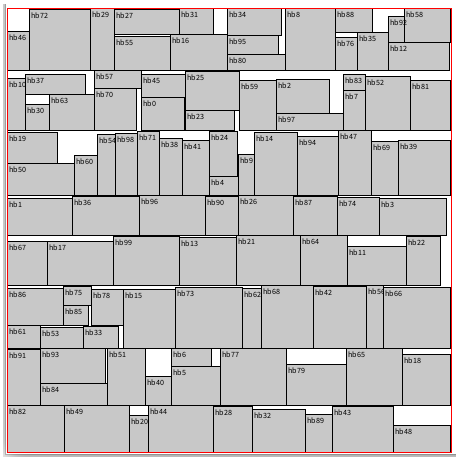
- public2.txt dead space ratio = 0.15



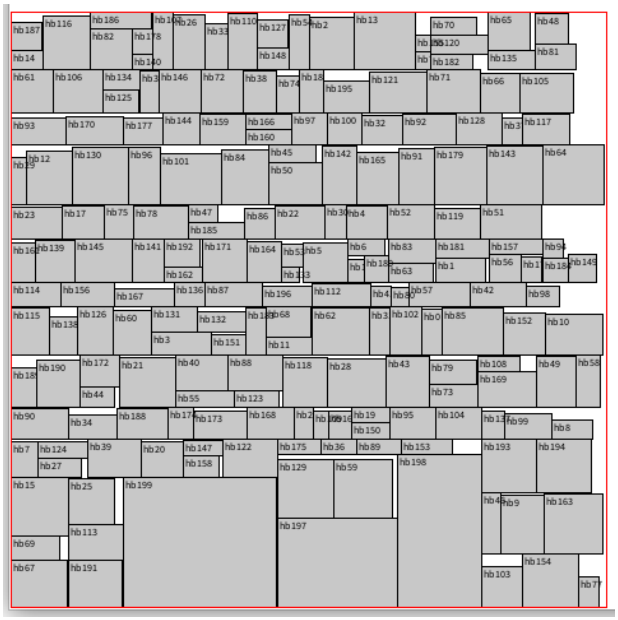
- public3.txt dead space ratio = 0.15



- public1.txt dead space ratio = 0.1



- public2.txt dead space ratio = 0.1



- public3.txt dead space ratio = 0.1