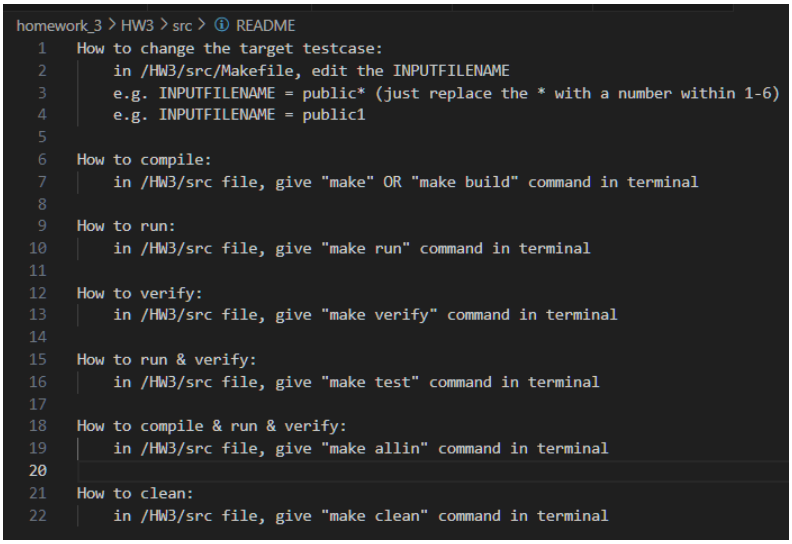
CS6135 VLSI Physical Design Automation

Homework 3: Fixed-outline Floorplanning with Fixed and Soft Modules

1. 112062525 蔡品棠

2. **How to compile and execute my program:**



```
homework_3 > HW3 > src > ⓘ README
1   How to change the target testcase:
2       in /HW3/src/Makefile, edit the INPUTFILENAME
3       e.g. INPUTFILENAME = public* (just replace the * with a number within 1-6)
4       e.g. INPUTFILENAME = public1
5
6   How to compile:
7       in /HW3/src file, give "make" OR "make build" command in terminal
8
9   How to run:
10      in /HW3/src file, give "make run" command in terminal
11
12  How to verify:
13      in /HW3/src file, give "make verify" command in terminal
14
15  How to run & verify:
16      in /HW3/src file, give "make test" command in terminal
17
18  How to compile & run & verify:
19      in /HW3/src file, give "make allin" command in terminal
20
21  How to clean:
22      in /HW3/src file, give "make clean" command in terminal
```

如 README 檔所示，進入/src 資料夾後，只要在終端機下「make allin」指令，就能一鍵跑完編譯與執行流程。

e.g. $ cd ./HW3/src

$ make allin

若要改變 testcase，於/src 的 Makefile 檔裡編輯「INPUTFILENAME」即可。

e.g. $ vim ./HW3/src/Makefile

in Makefile:

INPUTFILENAME = public1

若要分別進入/src 編譯和/bin 執行也沒問題，在/src 下 make 或 make build 指令後進到/bin 輸入以下指令即可:

$ ./hw3 ../testcase/public1.txt ../output/public1.floorplan

3. **The wirelength and the runtime of each testcase**

public1: wirelength = 186677514, runtime = 595.01 secs

public2: wirelength = 28679936, runtime = 595.01 secs

public3: wirelength = 3593805, runtime = 595.01 secs

public4: wirelength = 102510475, runtime = 595.01 secs

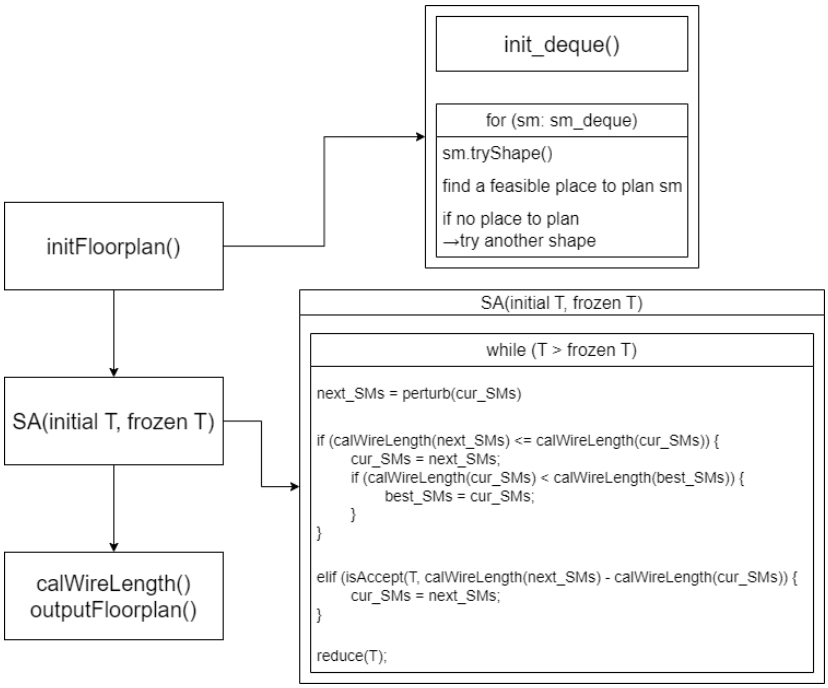
4. **How did you determine the shapes of the soft modules? What are the benefits of your approach?**

   我是在建構 SoftModule object 時，便呼叫 SoftModule::findAllShape()這個 method 來尋找可能的 height & width pair。將 min_area 開平方根後，從 sqrt(min_area)往下尋找可以整除 min_area 的 h，並記錄該次的 h 和 w，直到下限 min_h = initial_h / sqrt(2)。由於 aspect ratio 下限為 0.5，假設 min_h 為 x 則 max_w 為 2x，$2x^2$ = min_area = $initial\_h^2$，x = initial_h / sqrt(2) = min_h。若一組解都找不到，則把 min_area 的最小非 0 位數增加 1，再重新尋找一次可能解，如原先的 min_area = 64300，就會變成 64400 接著尋找可能解。

   而找完後會再把可能解做鏡像互換增加，比如找到一組 hw_pair (h1, w1) = (3, 2)，就會增加一組(h2, w2) = (2, 3)。

   這樣的好處在於建立 SoftModule object 時就先找好可能的形狀，並且不用考慮是否需要 rotate 的情況，因為在 retryShape 時就會一併嘗試。

5. **The details of your floorplanning algorithm.**



init_deque()僅用於初始化必要的 vector 變數。而會使用 deque 主要是因為其提供了與 vector 相同的隨機存取功能，以及常數時間且首尾皆能的的 push/pop。

6. **What tricks did you use to enhance your solution quality?**

   我將 SoftModules 以 min_area 由大到小排序，讓大面積的 module 會先被擺入，小面積的留到後面再找剩餘空間塞入即可。這樣可以把 initial floorplan 的失敗可能性降到最低。

   FixedModules 則以 x 座標由小到大排序，用於 debug 時方便找出 overlap() 的錯誤。

   由於實作的 Simulated Annealing 較簡易，我對初始溫度設置的較高，且降溫速度也非常低，藉以運行時能嘗試較多次 perturb。

7. **Parallelization?**

   我沒有做任何平行化。

8. **What have you learned from this homework? What problem(s) have you encountered in this homework?**

   起初我想利用 B* tree 來實作這次作業，但因為找到的論文都需要一個符合 constraint 的 initial floorplan 才能進行，因此我便先著手於在 initial floorplan 建立 admissible B* tree。不過經過多番嘗試，皆僅能通過 public2 以外的 testcases，光這裡就花了一週半的時間。

   最後是和朋友討論後，嘗試以我原本建立的資料結構來實作我們討論出來的方法才得以通過所有 testcases。