VLSI HW3

(1) 吳律穎 111062697

(2)

--HOW to Compile

In HW3/src/ , enter the following command:

$ make

It will generate the executable file "hw3" in "../bin/".

If you want to remove it, please enter the following command:

$ make clean

--How to execute

In this directory, enter the following command:

Usage: ../bin[exe] [hardblocks] [nets] [pl] [floorplan] [dead_space_ratio]

e.g.

```
/bin/hw3 ../testcases/n100.hardblocks ../testcases/n100.nets ../testcases/n100.
pl ../output/n100.floorplan 0.15
```

In "HW3/bin/", enter the following command:

Usage: ./[exe] [hardblocks] [nets] [pl] [floorplan] [dead_space_ratio]

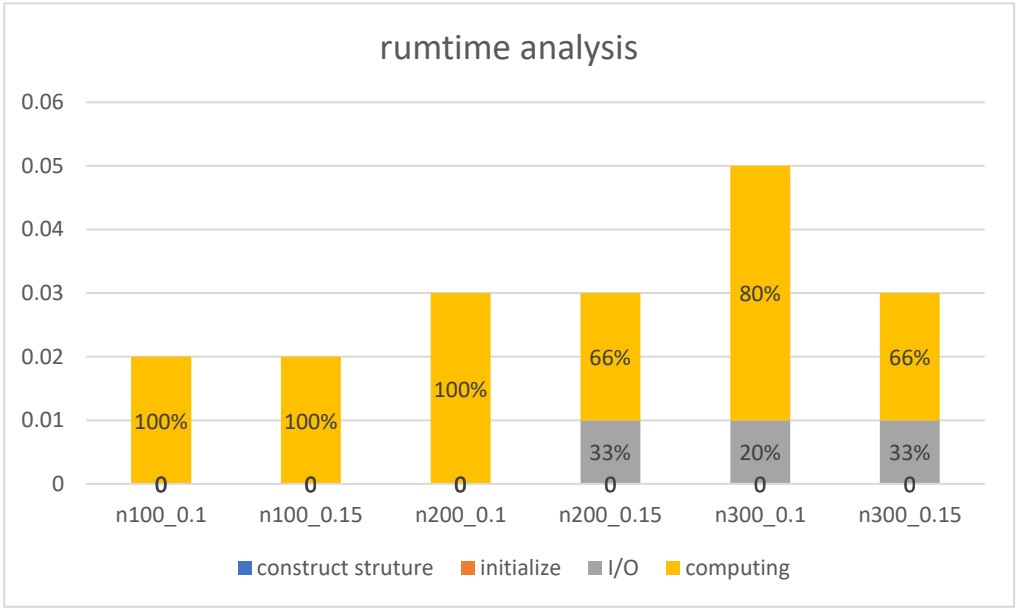
e.g.

```
./hw3 ../testcases/n100.hardblocks ../testcases/n100.nets ../testcases/n100.pl ..
/output/n100.floorplan 0.15
```

(3)

如下表所示，不論是建立結構或是初始化 polish 所花的時間都極小趨近 0%，主要的時間花費都在透過 polish 計算 floorplan 面積長度和 cost 的時間，I/O 時間也很少幾乎沒有，要讀到較大的檔如 n300 才有較多一些的時間

```
[g111061697@ic51 src]$ time ../bin/hw3 ../testcases/n100.hardblocks ../testcases/n100.net
s ../testcases/n100.pl ../output/n100.floorplan 0.1
real 0.02
user 0.02
sys 0.00
[g111061697@ic51 src]$ time ../bin/hw3 ../testcases/n100.hardblocks ../testcases/n100.net
s ../testcases/n100.pl ../output/n100.floorplan 0.15
real 0.02
user 0.02
sys 0.00
[g111061697@ic51 src]$ time ../bin/hw3 ../testcases/n200.hardblocks ../testcases/n200.net
s ../testcases/n200.pl ../output/n200.floorplan 0.1
real 0.03
user 0.03
sys 0.00
[g111061697@ic51 src]$ time ../bin/hw3 ../testcases/n200.hardblocks ../testcases/n200.net
s ../testcases/n200.pl ../output/n200.floorplan 0.15
real 0.03
user 0.02
sys 0.00
[g111061697@ic51 src]$ time ../bin/hw3 ../testcases/n300.hardblocks ../testcases/n300.net
s ../testcases/n300.pl ../output/n300.floorplan 0.1
real 0.05
user 0.04
sys 0.00
[g111061697@ic51 src]$ time ../bin/hw3 ../testcases/n100.hardblocks ../testcases/n100.net
s ../testcases/n100.pl ../output/n100.floorplan 0.15
real 0.03
user 0.02
sys 0.00
```



(4)

n100_0.1	block_area : 179501	Minarea: 193158	ratio : 0.076
n100_0.15	block_area : 179501	Minarea: 198414	ratio : 0.105
n200_0.1	block_area : 175696	Minarea: 191406	ratio : 0.089
n200_0.15	block_area : 175696	Minarea: 193984	ratio : 0.104
n300_0.1	block_area : 273170	Minarea: 294833	ratio : 0.079
n300_0.15	block_area : 273170	Minarea: 302978	ratio : 0.109

最小為 0.076

(5)

我初始化的 polish 與他不同，我是盡量將同高的 block 擺一起，碰到 outline 邊緣就往上繼續疊，是為了在開始就提供一個接近方形且面積小的 solution。

我的 cost function 與他不同，在超出 outline 的情況下 area 和 wire 的 cost 和最大只能為 1，超出 outline 的 penalty 為 max(超出的高 , 超出的寬)^2，而若在 outline 中 penalty 為 0，這種 penalty 可以讓形狀維持在接近正方形，並且會先收縮到 outline 中再優化 area 或 wire。

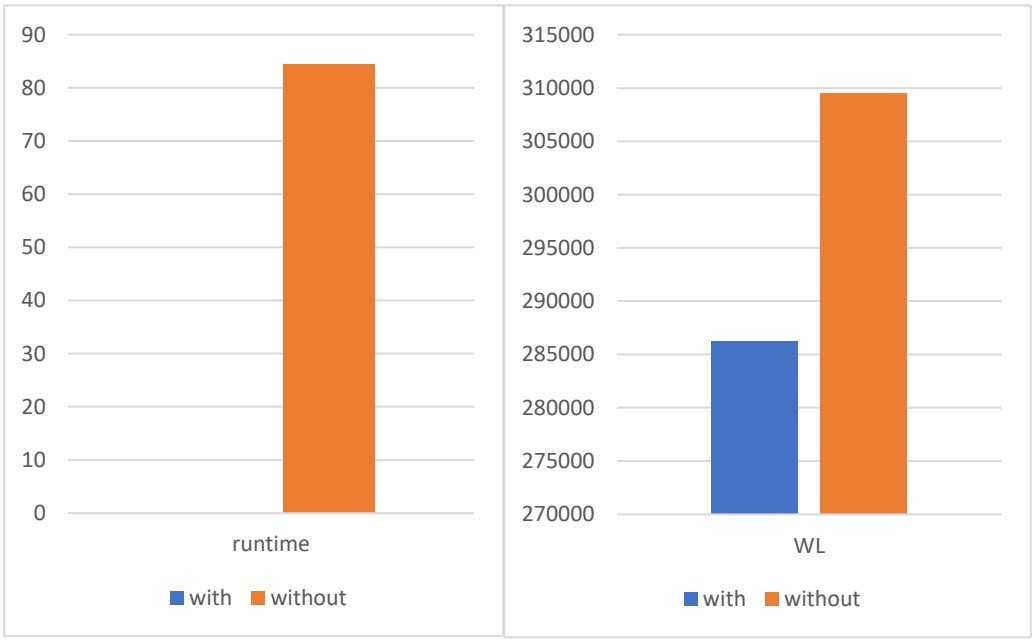
(6)

如(5)所述，照這種初始化的排法，原始 floorplan 就已經為一個在 outline 中或接近 outline 的解，所以可以大大縮短時間。

若不使用結果如下

```
[g111061697@ic51 src]$ time ../bin/hw3 ../testcases/n100.hardblocks ../testcases/n100.net
s ../testcases/n100.pl ../output/n100.floorplan 0.1
time : 84.52
seed : 1669970284
real 84.72
user 84.52
sys 0.00
```

Wirelength 為 309559



(7)

我的結果：

```
[g111061697@ic51 HW3_grading]$ bash HW3_grading.sh
|-------------------------------------------------|
|                                                 |
|    This script is used for PDA HW3 grading.     |
|                                                 |
|-------------------------------------------------|
grading on 111062697:
  testcase |       ratio | wirelength |     runtime | status
      n100 |        0.15 |     286230 |        0.02 | success
      n200 |        0.15 |     541929 |        0.03 | success
      n300 |        0.15 |     829303 |        0.04 | success
      n100 |         0.1 |     298135 |        0.02 | success
      n200 |         0.1 |     561899 |        0.03 | success
      n300 |         0.1 |     821088 |        0.04 | success
|-------------------------------------------------------|
|                                                       |
|    Successfully generate grades to HW3_grade.csv      |
|                                                       |
|-------------------------------------------------------|
```

Wirelength			Runtime		
n100	n200	n300	n100	n200	n300
286230	541929	829303	0.02	0.03	0.04

與前三名比較 (增加或減少的 %)：

	Wirelength			Runtime		
Ranks	n100	n200	n300	n100	n200	n300
1	+ 38%	+ 49.3%	+ 62.2%	- 99.85%	-99.96%	-99.98%
2	+ 36.7%	+ 42.7%	+ 58.9%	- 99.92%	-99.97%	-99.98%
3	+ 36.1%	+ 38.1%	+ 52.1%	- 99.94%	-99.98%	-99.99%

我的結果在長度方面輸給了它們，但於時間方面有優勢，可能是由於我的跳脫條件為進到 outline 就結束，但在進 outline 前我的 cost function 不會對 wire 或 area 直接給予好處或懲罰所導致

(8)

透過不斷的嘗試，學到了不同 cost function 對於結果的影響會有多大，也知道 initial 的差別會對於速度和 quality 有非常直接的關係，也學到不同 temperature 變動的策略。對於上述的各個因素，都進行了多次的測試，最後才能達到規定的 constraint。