

CS6135 VLSI Physical Design Automation

Homework 5 Report: Global Routing

108062605 呂宸漢

1. 如何編譯及執行程式：

進入 HW5/src/後，輸入 make 可編譯程式並輸出執行檔 hw5 至 HW5/bin/。

如：\$ make

之後進入 HW5/bin/，輸入 ./hw5 <modified.txt file> <result file> 執行程式。

如：\$./hw5 ../testcase/ibm01.modified.txt ../output/ibm01.result

注意：modified.txt file 的副檔名須為 .modified.txt、result file 的副檔名須為 .result。

2. overflow, wirelength and runtime for each case：

	ibm01	ibm02	ibm04
total overflow	0	0	195
total wirelength	60693	158498	162622
I/O time	0.06	0.13	0.15
routing time	2.42	6.85	297.36
runtime	2.48	6.98	297.51

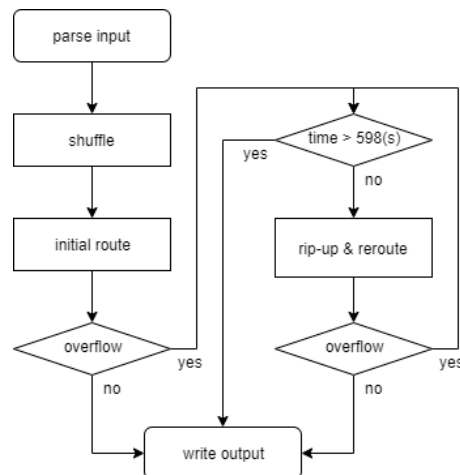
(時間以 clock_gettime 的 CLOCK_MONOTONIC 測量)

由上圖可以看出，最花時間的 stage 為 routing，前面兩個 case 都可以在短時間內做出沒有 overflow 的結果，可是最後一個 case 雖然仍有 overflow，可是我並沒有執行到 time out，因為在這個 overflow 之後，執行越多 rip-up reroute overflow 並沒有降低，因此我選擇執行到小於 200 就結束。

3. the detail of my implementation：

程式流程：

- 讀取檔案(包含讀取 modified.txt file)
- 建立 routing 需要的 array
- 執行 initial route(monotonic A* routing)
- 如果有 overflow 則執行 rip-up and reroute(A* routing with congestion cost)
- 如果沒有 overflow 或是執行時間已達 598 秒則寫入檔案(寫入 result file)



以下列出我的程式與課程的不同處

a. k1 不同：

雖然 k1 變大很有可能導致 overflow，可是在我測試下來其實第一個 case 和第二個 case 在 k1 為 100 的時候可以很快的 route 出不錯的結果，可是在第 3 個 case 時，反而會變很差，因此我只調整 case 1 和 case 2 的 k1，其餘的 case 仍然留在 5。

以下呈現不同 k1 在 ibm01 case 的執行結果

	in class (k1 = 5)	my
total overflow	0	0
total wirelength	60681	60693
I/O time	0.06	0.06
routing time	16.47	2.42
runtime	16.53	2.48

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

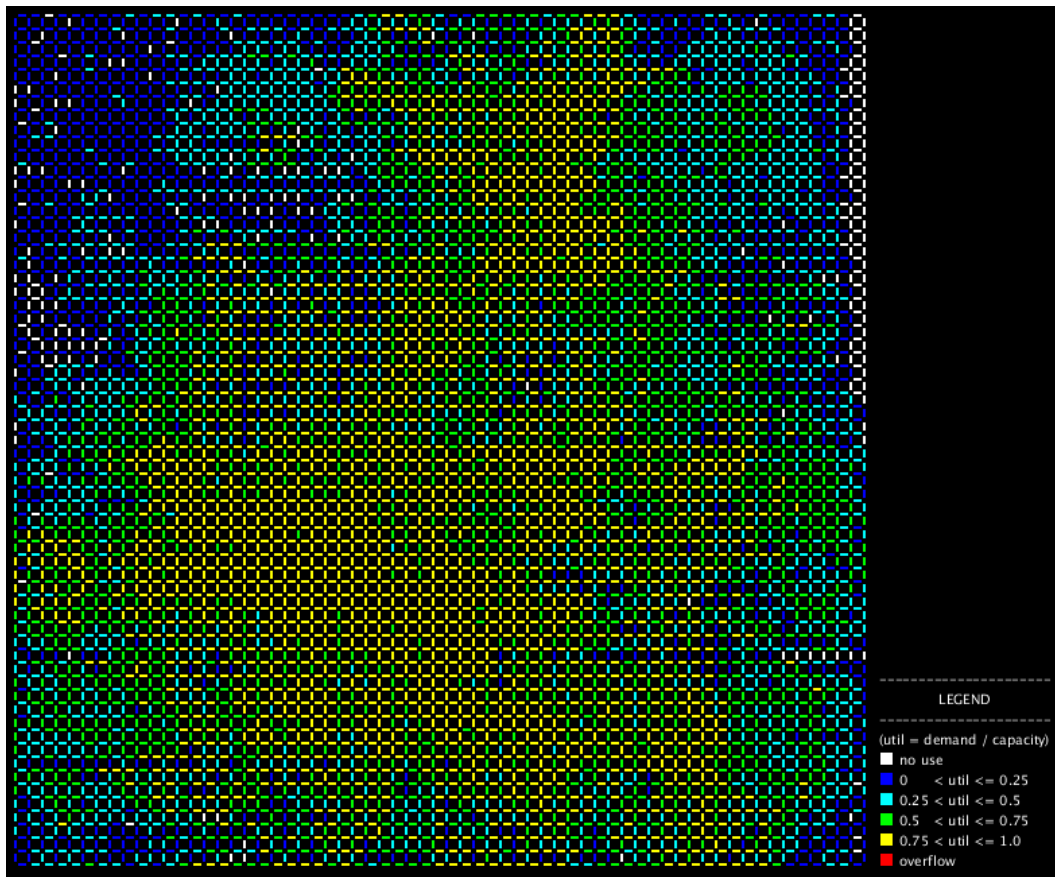
這次的作業參考了 nthu-route 2.0 的 code 與額外的一些資源，因為這次作業比較複雜，每個 edge 都要記錄有哪些 net 通過，而且 net 也要各自記錄自己的路徑，一開始在分辨 horizontal 與 vertical 很常自己被自己搞混，算是蠻複雜的 code 尤其是 rip-up and reroute 更是麻煩，有的時候反而會越 route overflow 越多，在嘗試了許久之後一直沒辦法把 ibm04 做到沒有 overflow，在多次觀察後，我才決定專為 ibm04 設一個 overflow 的限制，就是不做到 0，有較好的結果就直接輸出，才真正的完成了這次作業，不過很可惜還是沒有看懂 nthu-route 2.0 的 code，不然應該可以做出沒有 overflow 的結果。

5. If you implement parallelization, please describe the implementation details and provide some experimental results.

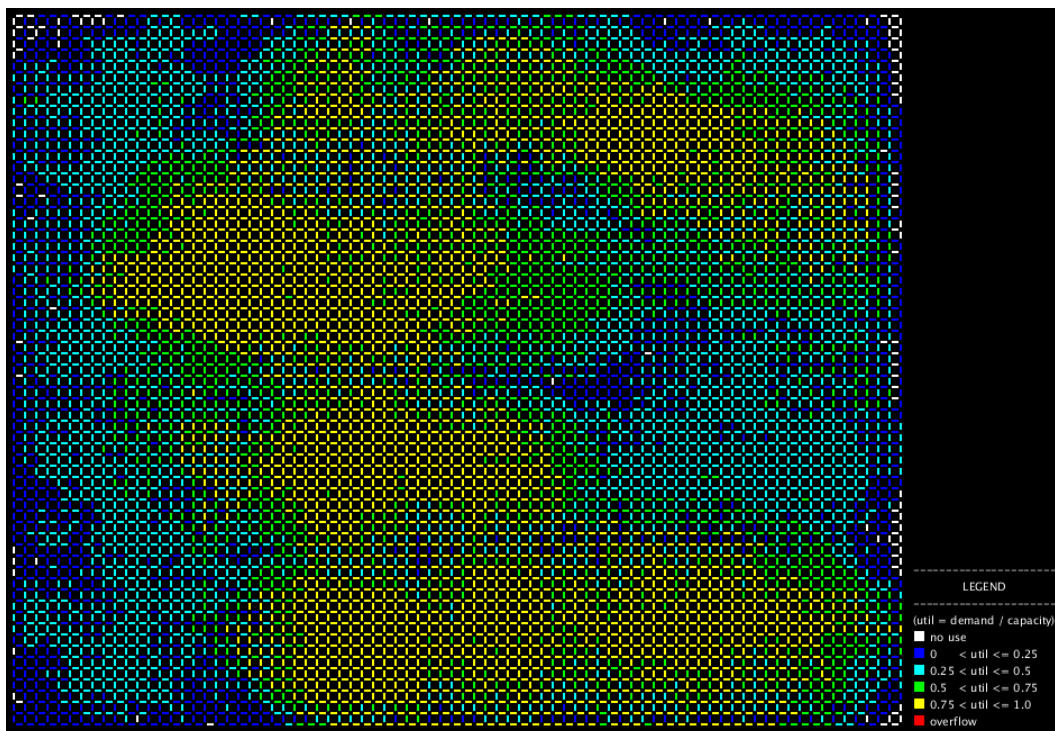
我沒有實作平行化的程式。

6. congestion map for each testcase

ibm01 :



ibm02 :



ibm04 :

