

CS6135 VLSI Physical Design Automation

Homework 5: Global Routing

108062537 魏聖修

2.

compile : make

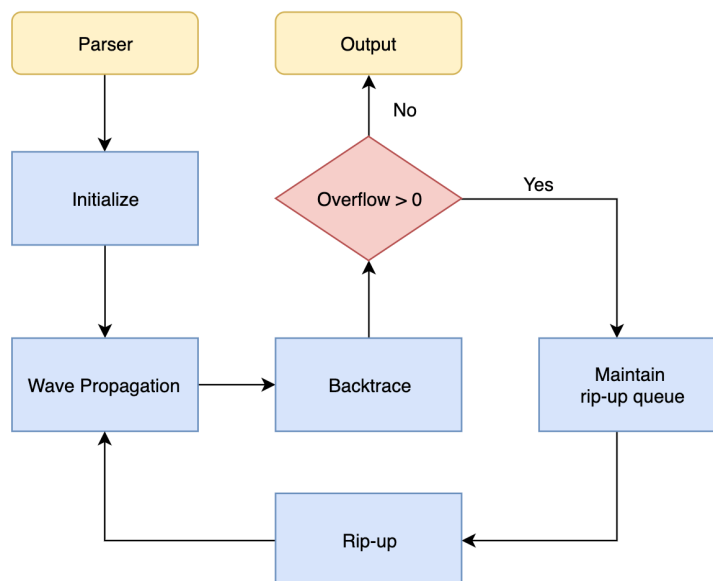
execute : ./hw5 <input_file> <output_name>

```
[[zscaxd5651@ic56 src]$ make
g++ -g -O3 main.cpp -o hw5
[[zscaxd5651@ic56 src]$ time ./hw5 ../testcase/ibm01.modified.txt ../output/ibm01.result
```

3.

	Runtime (sec)	Overflow	Wirelength
ibm01	90.09	0	60245
ibm02	255.66	0	159956
ibm04	476.94	142	161146

4.



這次作業主要對右邊那張圖的結構進行運算，所以要使用大量的 2D array，

initialize：

主要就是初始化這些 2D array。

routing：

主要的方法是類似 Lee algo，用 wave propagation 的方式在各個 grid 上 label，再 back_trace 找到要走的路徑，較不同的是這裡的 routing 不像 Lee algo 限制只能走最短路徑，而是利用在 grid 上 label cost 去選擇 cost 較少的路徑，這樣才能解決 overflow 的問題。

cal_overflow：

如剛剛所說，這次主要是要解決 overflow，所以我們做完一輪就會去計算現在的 overflow 數量，目標是 0。再來為了解決 overflow，我們進行 rip-up & reroute。

rip-up & reroute：

rip-up 的部分要去 maintain rip-up 的順序，我是去計算哪一條 edge 上的 overflow 最大，就去找有經過那個 edge 的所有 net 把他們 rip-up，然後再一條一條 reroute 回去，reroute 的部分就是 routing，做的事完全一樣。至於為甚麼重繞就能解決 overflow，是因為每條 edge cost 的計算，如同講義上的公式

History Based Cost Function

$$cost_e = b_e + h_e \times p_e + vc_e$$

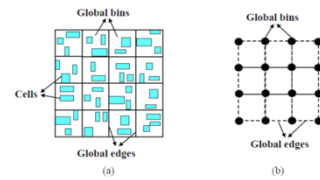
- b_e : adaptive base cost function
- $h_e \times p_e$: congestion cost function with overflow prediction
- vc_e : via cost function for multi-layer design
- Sharing edges does not need additional cost

主要是隨著時間越久，較容易 overflow 的 edge 上的 cost 就會越大，就越不會選擇他去走，因此就能解決 overflow 的問題。

5.

對於加速和優化，cost 的設定佔很大一部分。我有想過加速的方面，我發現如果不希望走較長的路徑，表示有 detour 就有 penalty 的話，速度會比較慢，不過 wirelength 會較短。反之如果把 penalty 刪掉後，速度會快很多，比較快就找出不會 overflow 的路徑，但相對 wirelength 也就較大。但最後也只有 ibm01 和 ibm02 能夠繞得出來，我最後還是放棄修改 cost function 了，但主要還是乖乖照著講義上的公式。

至於我想過第一次的 routing 結果會不會影響速度和品質，但我發現，我依

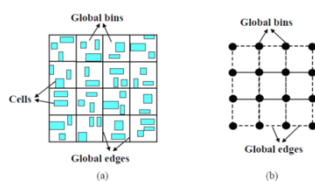


照兩個 pin 的距離，由最長的開始邊，這樣第一次的 routing 出來，結果是較好的 overflow 比較少。但是在全體看下來，和最原本的直接由 net_id 順序去繞，速度是差不多的，而且 wirelength 反而較長。所以最後也不做此修改。另外這次的作業難度真的很高，我主要的目標都是放在 overflow 而已，wirelength 真的就是隨緣。

速度方面，我覺得我的速度會慢是因為針對 overflow edge 的 net 做 rip-up reroute，這樣感覺就會對某些 net 重複做很多次無用的動作，也許能夠有更好的挑選順序方式，能每次都針對較 critical 的 net 去做 rip-up reroute，想必這樣速度應該會快上許多的。

6.

這次作業難度真的很高，一開始完全照著 Lee algo 做，做出來的 routing 再去計算 cost，卻發現怎樣都解決不了 overflow，才發現不能用最短路徑走，整個 wave propagation 和 back trace 變成都要做改變。我覺得最難的部分就是這兩塊和 cost 的計算。此外下圖也讓我費了不少時間後，才發現 grid 和 edge 應該



要分開計算，不然連有沒有 overflow 都計算不出來。而且大量的 2D array，設定起來真得頗為複雜，常常就跑出邊界造成 segmentation fault。果然 routing 不是一門簡單的學問。這次已經有一堆假設來簡化題目還是搞的焦頭爛額，真正 routing

問題勢必更難以解決。

最後這學期真的學到不少的東西，但也發現到自己還有很多的地方可以更加強。謝謝教授和助教這學期的教導，你們辛苦了！！！！