CS6135 VLSI Physical Design Automation

Homework 1: P&R Tool

112062590 劉俊圻

1. Comparison Table

	(congestion-driven, timing-driven)					
	(L, off)	(L, on)	(M, off)	(M, on)	(H, off)	(H, on)
slack	0.304	0.321	0.402	0.553	0.402	0.553
Total wirelength	110881.355	103962.625	110082.985	103126.470	110082.985	103126.470

Core utilization: 0.7, Clock period: 10

在固定的 Core utilization 和 Clock period 下發現以下關係:

(1) 固定的 congestion-driven 程度和不同的 time-driven 選項

從結果中可觀察到 timing-congestion on 的結果會比 timing-congestion off 的結果有更長的 slack time 和更短的 total wire length。我認為由於 time-driven 會基於 timing 進行優化，越短的 wire length 就會有越短的 slack time，因此得到這樣的結果

(2) 固定的 time-driven 和不同的 congestion-driven 選項

從結果中可觀察到更高等級的 congestion-driven 會得到更長的 slack time 和更短的 total wire length，但達到一定程度之後兩者便不會再有更優化的結果。這部分和我預期的結果不同，我認為 congestion-driven 會基於發生 congestion 處(standard cell 擁擠處)進行調整，減輕 congestion 的現象，因此 standard cell 應該會分散，使得 total wire length 增加，同時減少 slack time。因此我調整了幾次不同參數再次實驗發現以下結果

	(congestion-driven, timing-driven)					
	(L, off)	(L, on)	(M, off)	(M, on)	(H, off)	(H, on)
slack	1.057	0.997	1.189	1.153	1.189	1.153
Total wirelength	108837.82	109580.905	109304.51	109331.91	109304.51	109331.91

Core utilization: 0.6

Clock period: 10

同樣是固定的 timing-driven，不同的 congestion-driven 程度，越高的 congestion-driven 程度如預期中得到越高的 total wire length，但 slack time 卻會增加，而且在這組實驗結果中，在同樣 congestion-driven 程度的情況下，開啟 timing-driven 所得到的 slack time 卻會比沒有開啟 timing-driven 來的少，我認為會有這樣的結果是因為 congestion-driven 和 timing-driven 所進行的調整過程與 core utilization 相關，不同的 core utilization 可能會得到不一樣的調整變化。

2. Differences between congestion-driven and timing-driven

Timing-driven placement 是以 timing 為優化對象進行 placement，針對在 timing critical path 上的 wire 進行設計，因此開啟 timing-driven placement 後的 slack time 會增加。

Congestion-driven placement 則是針對空間不足，standard cells 密度過高，發生 congestion 區域進行優化以解決 cells 擁擠問題。

兩者針對的問題不同，舉例來說，若是在 timing critical path 上發生 congestion，congestion-driven placement 會對 congestion 的部分進行優化，但可能會造成 timing violation；timing-driven 則會以 timing 為優先，不一定會針對 timing critical path 的 congestion 部分進行優化。

3. Why do we insert filler cells?

Filler cell 是沒有邏輯功能的 standard cell，用於填充 standard cells 之間的空白區域。我認為 Filler cell 是用於連續 VDD/VSS，讓電源/接地相關的 wire 得以在 standard cells 之間延續，同時也為了 die 的表面平整性而對 standard cell 之間的空白區域進行填充，得到一致的高度，以利後續的製造或封裝程序。

4. Best result

Parameters	
Clock period	7
Congestion-driven	medium
Timing-driven	off
Result	
Total area of chip	13132.101 um^2
Total wire length	100731.1450 um
Slack	0.090

Clock period: 7

```
et_units -time ns -resistance MOhm -capacitance fF -voltage V -current mA
create_clock [get_ports clk] -name CLK -period 7.0 -waveform {0 3.5}
roup_path -name input_group  -from [list [get_ports clk] [get_ports nreset]
```

Congestion-driven effort: medium, Timing-driven: off

```
etPlaceMode -reset
setPlaceMode -congEffort medium -timingDriven 0 -clkGateAware 1 -powerDriven 0 -ignoreScan 1 -reorderScan 1 -ignoreSpare 0 -placeIOPins 1 -moduleAwareSpare 0 -p
reserveRouting 1 -rmAffectedRouting 0 -checkRoute 0 -swapEEQ 0
etPlaceMode -fp false
```

Total area of chip: 13132.101um^2

```
1618 Total area of Core: 11311.650 um^2
1619 Total area of Chip: 13132.101 um^2
1620 ffective Utilization: 1.0000e+00
```

Total wire length: 100731.1450um

```
3 otal metal6 wire length: 8577.1000 um
4 Total wire length: 100731.1450 um
5 verage wire length/net: 13.8234 um
```

Slack time: 0.090

```
  Arrival Time                  6.824
= Slack Time                    0.090
    Clock Rise Edge
```

5. Snapshot of final best layout result