CS6135 VLSI Physical Design Automation
Homework 1: P&R Tool

111062625 蔡哲平

1. Comparison table

Core utilization = 0.7, Clock period= 4ns

	(Congestion-driven, Timing-driven)					
	(L, off)	(L, on)	(M, off)	(M, on)	(H, off)	(H, on)
Slack	0.379	0.511	0.346	0.496	0.346	0.496
Total wirelength (um)	311835.948	309827.965	312612.563	310789.010	312612.563	310789.010

a. Timing-driven placement increases slack and reduces total wirelength.

Slack value equals to required time minus arrival time. The higher the slack value, the more time you have left. When timing-driven placement is turned on, it will surely increase slack value as a result of minimizing critical path delay. As for the reason why it reduces wirelength, in my opinion, it's because during timing-driven placement, it will try to shorten the wire to reduce the delay. Therefore, total wirelength is reduced.

b. Total wirelength is shorter when congestion-driven is low.

When congestion-driven placement is turned on, the separation between standard cells will be increased to solve the congestion problem. As a result of increased distance, total wirelength will be longer to connect within cells. According to the comparison table above, we can see that total wirelength is same during low and medium congestion-driven placement. In my opinion, it's because the maximum level of congestion problem that can be solved is bounded by core utilization. Hence, total wirelength isn't increasing when high congestion-driven placement is taken. I also made an experiment as below to prove my opinion.

*Timing-driven = off, Period = 4ns	Core utilization = 0.6	Core utilization = 0.7	Core utilization = 0.8
Congestion-Low wirelength	335122.7250 um	311835.9475 um	292854.3075 um
Congestion-Medium wirelength	337576.8950 um	312612.5625 um	294501.7875 um
Congestion-High wirelength	337576.8950 um	312612.5625 um	294501.7875 um

2. **What are the difference(s) between the congestion-driven placement and timing-driven placement?**

   If the number of routing tracks available for routing in one area is less than the required routing tracks, then the area said to be congested. Hence, Congestion driven placement is performed to reduce the congestion. Besides congestion, timing is also very important. Incorrect timing can lead to incorrect output of an IC. Therefore, timing-driven placement is designed specifically targeting wires on timing critical paths.

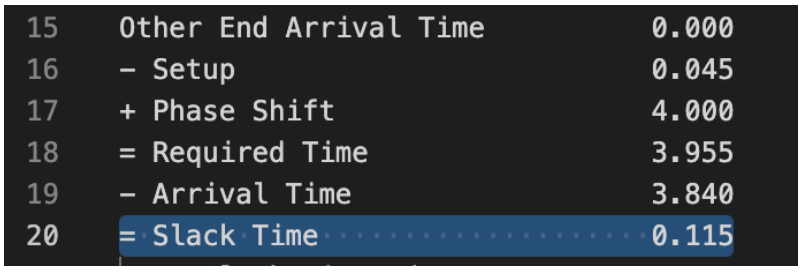
3. **Why do we insert filler cells?**

   Filler cells are used to fill any spaces between regular library cells to avoid planarity problems. They are needed when the density of the required metal or layer has not met the foundry or fab requirement. In order to avoid the problem of yield during mass production due to the uneven distribution of standard cell density when the layout is made into a chip, filler cells are added to make the chip density more uniform.

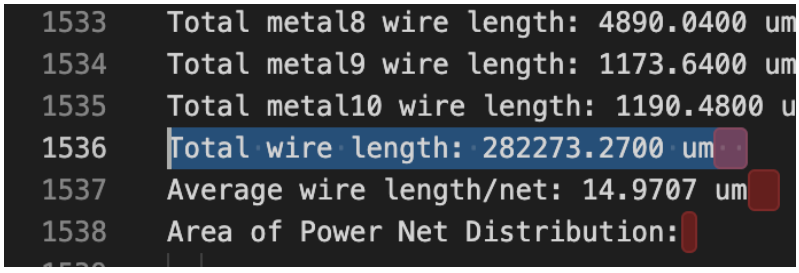
4. **Best result**

Parameters		Result	
Clock period (ns)	4	Slack	0.115
Core utilization	0.999999	Total area of chip (um^2)	38823.019
Congestion-driven	low	Total wirelength (um)	282273.2700
Timing-driven	off	DRC violations	0

Snapshot: slack time = 0.115



Snapshot: Total wire length = 2882273.2700 um



Snapshot: Total area of chip = 38823.019 um^2

```
1506    Total area of Core: 35644.000 um^2
1507    Total area of Chip: 38823.019 um^2
1508    Effective Utilization: 1.0000e+00
```

Snapshot: 0 drc violations

```
19    VERIFY DRC ..... Thread : 4 finished.
20
21    Verification Complete : 0 Viols.
22
```

Snapshot: clock period = 4ns

```
9   set_clock_gating_check -rise -setup 0
10  set_clock_gating_check -fall -setup 0
11  create_clock [get_ports {clk}] -name VCLK -period 4.0 -waveform {0.0 2.0}
12  set_max_fanout 10  [get_ports {clk}]
13  set_max_fanout 10  [get_ports {reset_n}]
14  set_max_fanout 10  [get_ports {cs}]
```

Snapshot: Core utilization = 0.999999

```
31  saveDesign setup
32  getIoFlowFlag
33  setIoFlowFlag 0
34  floorPlan -coreMarginsBy die -site FreePDK45_38x28_10R_NP_162NW_340 -r 1.0 0.999999 4.0 4.0 4.0 4.0
35  uiSetTool select
36  getIoFlowFlag
```

Snapshot: Congestion-driven = low, Timing driven = off

```
    BUF_X1 INV_X32 INV_X16 INV_X8 INV_X4 INV_X2 INV_X1} -maxAllowedDelay 1
44  setPlaceMode -reset
45  setPlaceMode -congEffort low -timingDriven 0 -clkGateAware 1 -powerDriven 0 -ignoreScan 1 -reorderScan 1 -ignoreSpare 0
    -placeIOPins 1 -moduleAwareSpare 0 -preserveRouting 1 -rmAffectedRouting 0 -checkRoute 0 -swapEEQ 0
46  setPlaceMode -fp false
47  place_design
```