VLSI HW1

112065504 徐祈

1. Comparing table

	(congestion-driven, timing-driven)					
	(L, off)	(L, on)	(M, off)	(M, on)	(H, off)	(H, on)
Slack time	0.304	0.321	0.402	0.554	0.402	0.554
Total wire length	110881.3550	103962.6250	110082.9850	103126.4700	110082.9850	103126.4700

clock period: 10 ns, core utilization: 0.7

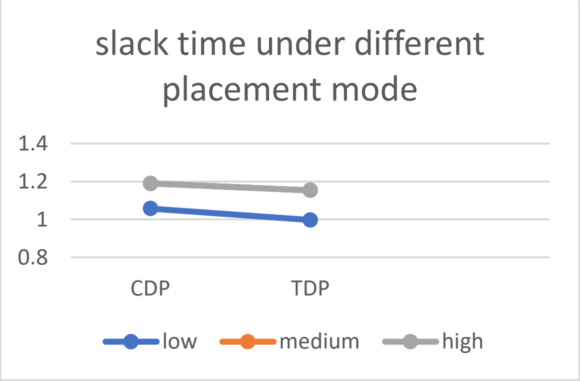
	(congestion-driven, timing-driven)					
	(L, off)	(L, on)	(M, off)	(M, on)	(H, off)	(H, on)
Slack time	1.057	0.997	1.190	1.153	1.190	1.153
Total wire length	108837.8200	109580.9050	109307.5700	109331.9100	109304.5100	109331.9100

clock period: 10 ns, core utilization: 0.6

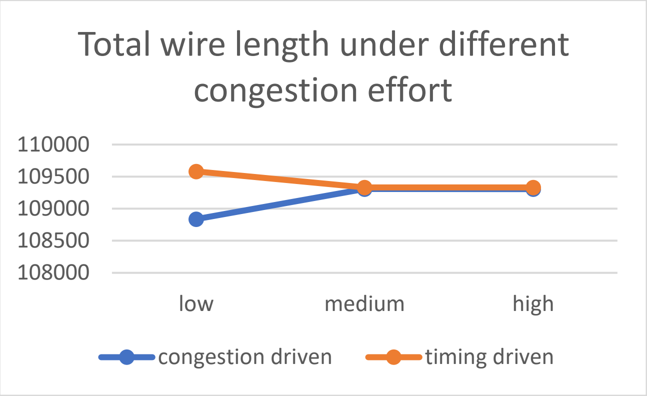
※(M, off), (H, off) and (M, on), (H, on) have the exactly same experimental statistics when core utilization=0.7

To make sure that the same statistics are mistakes or not, I redo all the experiments again with different core utilization. This time, (M, off), (H, off) and (M, on), (H, on) are still highly correlated. The following graph are made with core utilization=0.6.

Slack time is usually smaller if timing-driven is on. The reason is the critical path becomes shorter, so margin of delay becomes shorter, too.



Under congestion-driven placement, the total wire length becomes longer when the congestion effort increases. It's straightforward since when the spaces between cells increase, the longer wire length are needed. However, under timing-driven placement, the total wire length dropping as the congestion effort increase. The timing-driven placement aim to minimize the critical path on the chip, which the congestion effort doesn't affect a lot.



2. Congestion-driven placement vs Timing-driven placement

Timing-driven placement's main purpose is to reduce the critical path length, so that the chip can run faster. Timing-driven placement is usually more tolerant to congestion since if we try to put cells closer to reduce the critical path, the congestion level will increase.

The goal of congest-driven placement is to alleviate congestion on the chip. The logic elements may spread wider to prevent congestion, which means some timing optimization may be sacrificed.

3. Inserting filler cells

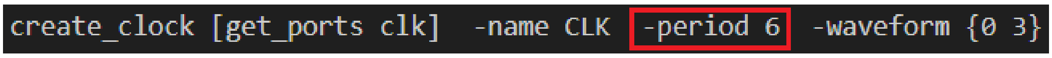
After automatic place and route, it's impossible to fill the chip 100%, since spaces are needed during routing. We insert non-functional cells that make VDD & VSS short together, which called filler cells. They are abutted the standard cells, which can also prevent some spacing DRC violations.

References:

- Why do we use filler cells in VLSI? - Siliconvlsi
- Physical Only Cells: Filler Cells – VLSI Pro

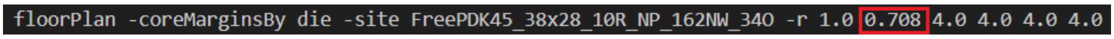
4. Best result

- parameters
    - ✧ clock period=6 ns



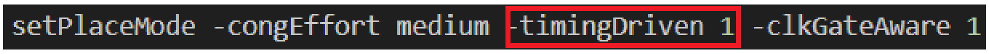
```
create_clock [get_ports clk]  -name CLK -period 6  -waveform {0 3}
```

    - ✧ core utilization=0.708



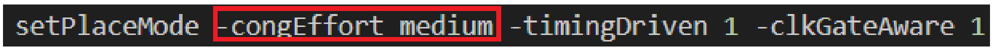
```
floorPlan -coreMarginsBy die -site FreePDK45_38x28_10R_NP_162NW_34O -r 1.0 0.708 4.0 4.0 4.0 4.0
```

    - ✧ time-driven=true



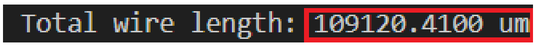
```
setPlaceMode -congEffort medium -timingDriven 1 -clkGateAware 1
```

    - ✧ congestion effort=medium



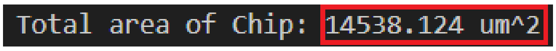
```
setPlaceMode -congEffort medium -timingDriven 1 -clkGateAware 1
```

- statistics
    - ✧ total wire length=109120.4100 um



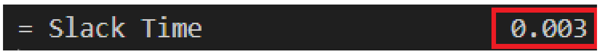
```
Total wire length: 109120.4100 um
```

    - ✧ total area of chip=14538.124 um^2



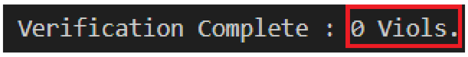
```
Total area of Chip: 14538.124 um^2
```

    - ✧ slack time=0.003



```
= Slack Time                    0.003
```

- checking
    - ✓ DRC=0 Viols



```
Verification Complete : 0 Viols.
```

    - ✓ Non-negative slack=true (0.003)
- Snapshot
    - ✓ final chip layout (with / without net signal)