CS6135_HW1_110062619_report

1. Your name and student ID

Name: 楊淨富 Student ID:110062619

2. A comparison table like the following one, and an explanation of the result (The table should be built under a fixed core utilization and clock period and you should specify them in the report.)

- core utilization: 0.95

```
floorPlan -coreMarginsBy die -site FreePDK45_38x28_10R_NP_162NW_34O -r 1.0 0.95 4.0 4.0 4.0 4.0
```

- clock period: 1.7

```
create_clock [get_ports {clk}] -name VCLK -period 1.7 -waveform {0.0 0.85}
```

(congestion, timing) denotes (congestion-driven, timing-driven)

	(congestion, timing)	(congestion, timing)	(congestion, timing)	(congestion, timing)	(congestion, timing)	(congestion, timing)
	(Low, off)	(Low, on)	(Medium, off)	(Medium, on)	(High, off)	(High, on)
slack	0.068	0.020	0.076	0.051	0.060	0.059
total wirelength	275572.927 um	275598.723 um	275979.648 um	276884.533 um	280453.818 um	280695.430 um

3. The difference(s) between the congestion-driven placement and timing-driven placement

- congestion-driven placement

通常用於die size較小的情況，最主要是將std cells間的間隔拉大，用來減輕congestion，但也會讓wirelength也隨之上升，此外，運行時間也會上升，可能也會對clock造成一定影響。

- timing-driven placement

主要就是把timing-critical path上的cells放的緊密一點，以減少延遲，但可能造成congestion增多，此外，仍必須符合setup timing。

Ref:

https://www.edaboard.com/threads/timing-driven-and-congestion-driven.108810/

https://link.springer.com/content/pdf/10.1007/1-4020-8063-8_5.pdf

4. An explanation of why we insert filler cells

💡 In standard cells APR flow, the cells in the design are placed on the row. To make sure that each cells gets power and ground connection, the cells are abutted together so that the VDD and VSS terminal of neighboring cells short together. This makes it possible to tap power only at one point anywhere in the row. But it is virtually impossible to fill 100% of the die area with regular cells. So, we use filler cells to fill these spaces between regular library cells to route power rails. Filler cells are used for connecting the gaps between the cells after placement.

💡 利用Filler cells(擺放於std cells之間的empty space)，可以使每個cell維持穩定連接power和ground(即維持continuity)，此外也能避免nwell spacing DRC error。

Ref:

https://www.quora.com/Why-do-we-use-filler-cells-for-N-well-continuity

https://vlsi.pro/physical-only-cells-filler-cells/

https://blog.csdn.net/Tao_ZT/article/details/102456754

5. Show your best result (including clock period, total area, total wirelength, slack, congestion-driven effort and timing-driven on/off settings, and their snapshots) to maintain a non-negative slack and no DRC violation.

- clock period

```
[g110062619@ic51 Low,on]$ cat design.sdc
#############################################################
#  Generated by:      Cadence Innovus 18.13-s088_1
#  OS:                Linux x86_64(Host ID hansolo.poly.edu)
#  Generated on:      Thu Oct 21 00:04:46 2021
#  Design:            aes
#  Command:           write_db out/post_impl_aes_70_orig.final
#############################################################
current_design aes
set_clock_gating_check -rise -setup 0
set_clock_gating_check -fall -setup 0
create_clock [get_ports {clk}] -name VCLK -period 1.7 -waveform {0.0 0.85}
```

- total area of the chip: 40813.604um^2

```
==============================
Floorplan/Placement Information
==============================
Total area of Standard cells: 37552.284 um^2
Total area of Standard cells(Subtracting Physical Cells): 35054.810 um^2
Total area of Macros: 0.000 um^2
Total area of Blockages: 0.000 um^2
Total area of Pad cells: 0.000 um^2
Total area of Core: 37552.284 um^2
Total area of Chip: 40813.604 um^2
Effective Utilization: 1.0000e+00
Number of Cell Rows: 138
% Pure Gate Density #1 (Subtracting BLOCKAGES): 100.000%
% Pure Gate Density #2 (Subtracting BLOCKAGES and Physical Cells): 93.349%
% Pure Gate Density #3 (Subtracting MACROS): 100.000%
% Pure Gate Density #4 (Subtracting MACROS and Physical Cells): 93.349%
% Pure Gate Density #5 (Subtracting MACROS and BLOCKAGES): 100.000%
% Pure Gate Density #6 (Subtracting MACROS and BLOCKAGES for insts are not placed): 93.349%
% Core Density (Counting Std Cells and MACROs): 100.000%
% Core Density #2(Subtracting Physical Cells): 93.349%
% Chip Density (Counting Std Cells and MACROs and IOs): 92.009%
% Chip Density #2(Subtracting Physical Cells): 85.890%
# Macros within 5 sites of IO pad: No
Macro halo defined?: No
```

- total wirelength: 275598.7225 um

```
==============================
Wire Length Distribution
==============================
Total metal1 wire length: 2166.5975 um
Total metal2 wire length: 71242.8575 um
Total metal3 wire length: 99073.2325 um
Total metal4 wire length: 51995.2300 um
Total metal5 wire length: 26285.0100 um
Total metal6 wire length: 17201.2400 um
Total metal7 wire length: 1428.9950 um
Total metal8 wire length: 4157.1200 um
Total metal9 wire length: 1280.4400 um
Total metal10 wire length: 768.0000 um
Total wire length: 275598.7225 um
Average wire length/net: 14.3295 um
```

- slack

```
[g110062619@ic51 Low,on]$ cat timing.rpt
############################################################
#  Generated by:      Cadence Innovus 20.10-p004_1
#  OS:                Linux x86_64(Host ID ic51)
#  Generated on:      Tue Oct  4 16:45:13 2022
#  Design:            aes
#  Command:           report_timing > timing.rpt
############################################################
Path 1: MET Setup Check with Pin core_dec_block_block_w1_reg_reg[4]/CK
Endpoint:   core_dec_block_block_w1_reg_reg[4]/D (v) checked with  leading edge
of 'VCLK'
Beginpoint: encdec_reg_reg/QN                   (v) triggered by  leading edge
of 'VCLK'
Path Groups: {VCLK}
Analysis View: generic_view
Other End Arrival Time          0.000
- Setup                         0.057
+ Phase Shift                   1.700
= Required Time                 1.643
- Arrival Time                  1.623
= Slack Time                    0.020
```

- congestion-driven effort and timing-driven on/off settings
  - congestion-driven: low
  - timing-driven: on

```
setPlaceMode -congEffort low -timingDriven 1 -clkGateAware 1 -powerDriven 0 -ignoreScan 1 -reorderScan 1 -ignoreSpare 0 -placeIOPins 1 -moduleAwareSpare 0 -preserveRouting 1 -rmAffectedRouting 0 -checkRoute 0 -swapEEQ 0
```

- DRC.rpt

```
[g110062619@ic51 Low,on]$ cat drc.rpt
 *** Starting Verify DRC (MEM: 2260.0) ***

  VERIFY DRC ..... Starting Verification
  VERIFY DRC ..... Initializing
  VERIFY DRC ..... Deleting Existing Violations
  VERIFY DRC ..... Creating Sub-Areas
  VERIFY DRC ..... Using new threading
 VERIFY DRC ..... Sub-Area: {136.080 0.000 202.730 68.040} 3 of 9  Thread : 0
 VERIFY DRC ..... Sub-Area: {0.000 0.000 68.040 68.040} 1 of 9  Thread : 3
 VERIFY DRC ..... Sub-Area: {136.080 136.080 202.730 201.320} 9 of 9  Thread : 2
 VERIFY DRC ..... Sub-Area: {0.000 136.080 68.040 201.320} 7 of 9  Thread : 4
 VERIFY DRC ..... Sub-Area: {68.040 0.000 136.080 68.040} 2 of 9  Thread : 3
 VERIFY DRC ..... Sub-Area: {68.040 136.080 136.080 201.320} 8 of 9  Thread : 4
 VERIFY DRC ..... Sub-Area: {136.080 68.040 202.730 136.080} 6 of 9  Thread : 3
 VERIFY DRC ..... Sub-Area: {0.000 68.040 68.040 136.080} 4 of 9  Thread : 4
 VERIFY DRC ..... Thread : 3 finished.
 VERIFY DRC ..... Thread : 5 finished.
 VERIFY DRC ..... Sub-Area: {68.040 68.040 136.080 136.080} 5 of 9  Thread : 4
 VERIFY DRC ..... Thread : 4 finished.

  Verification Complete : 0 Viols.

 *** End Verify DRC (CPU: 0:00:03.4  ELAPSED TIME: 2.00  MEM: 70.0M) ***
```