

AI디지털집적회로

Assignment #4

전기전자공학부

202110410

조민우

## 서론

과제 2에서는 과제 1에서 합성한 RCA 기반 8-bit multiplier를 하위 모듈(submodule)로 포함하는 MAC을 대상으로 hierarchical synthesis를 진행하고 Static Timing Analysis(STA)를 수행하였다. 또한, 과제 3에서는 과제 2에서 만든 mac8 unit을 하위 모듈로 사용해 systolic array(sa4x4) 모듈을 hierarchical sysnthesis하고, STA(Static Timing Analysis)를 수행하여 SDF 파일을 생성하고 QuestaSim 시뮬레이터에 back-annotation하여, 합성된 회로의 delay가 반영된 동적 타이밍 시뮬레이션을 진행하였다.

과제 4에서는 과제 3에서 합성한 sa4x4 모듈을 가져와 ICC2 툴을 사용하여 PNR(Placement & Routing) 과정을 진행하려 한다. 이 과정에서 ICC2 툴의 commands와 options들에 익숙해지고, 물리적 layout을 생성하여 Backend 설계 흐름을 익히고자 한다. 또한, 이 PNR 결과를 바탕으로 PPA를 평가 및 파악하고 최적의 설계점을 찾고자 한다.

본격적인 분석에 앞서 마지막 과제인 만큼, 지금까지 진행된 과제들의 흐름을 정리하고 이번 과제가 갖는 의미를 고찰하고자 한다.

구분	Assn1	Assn2	Assn3	Assn4
타겟 모듈	mult8	mac8	sa4x4	sa4x4(from Assn3)
사용 Tool	Design Compiler	Design Compiler, PrimeTime, QuestaSim	Design Compiler, PrimeTime, QuestaSim	ICC2
분석 방법	Timing Report Analysis	STA	Dynamic Timing Verification	Floorplanning, Placement, CTS, Routing
검증 방법	Behavior Simulation	STA	Dynamic Gate-level Simulation	DRC, LVS, Post- layout STA, IR drop Analysis, Connectivity Check
Corner 고려	Typical	Slow/Fast/OCV	Slow/Fast/OCV + SDF max delay	Single Corner Only, MCMM (이번 과제는 제외)
설계 방식	Frontend	Frontend	Frontend	Backend

<Fig 1.1 Comparison between Assignments>

과제 1부터 3까지는 Frontend 설계 단계로, RTL 코드의 logic으로 올바른 gate-level netlist로 변환하는 과정에 집중했다. 이 과정에서는 Design Compiler를 이용한 logic synthesis를 통해 standard cell들의 조합으로 회로를 구성하지만, 각 cell들이 실제 chip에서 어디에 배치될지, 어떻게 연결될지는 결정되지 않는다. Timing 분석 역시 wire load model이나 RC estimation 같은 추정치에 의존하며, PrimeTime의 STA도 이론적인 worst-case 분석에 그친다.

Partitioning	Top-level에서 기능별 모듈들을 논리적으로 분할
Chip planning	<b>Floorplanning</b> 이라고도 함, user가 직접 모듈 위치/크기를 결정
Placement	Standard cell들을 chip 영역에 물리적으로 배치
Clock Tree Synthesis (CTS):	Clock distribution network 생성, Clock skew 최소화가 목표
Signal Routing	Standard cell 간 metal wire로 연결
Timing Closure	Setup/Hold time violation 해결

<Fig 1.2 Physical Design Flow>

반면 과제 4는 Backend 설계 단계로, ICC2를 사용하여 netlist를 실제 제조 가능한 물리적 layout으로 구현하는 Place-and-Route 과정을 다룬다. 이 단계에서는 floorplanning을 통해 die size와 I/O 위치를 결정하고, power planning으로 power delivery network를 생성하며, placement로 standard cell들의 물리적 좌표를 할당하고, routing으로 실제 metal layer에 배선을 완성한다. 가장 중요한 차이는 parasitic extraction을 통해 실제 배선의 RC 값을 추출하여 정확한 timing을 분석할 수 있다는 점이며, DRC/LVS 같은 물리적 검증을 통과해야만 실제 fabrication이 가능한 GDSII 파일을 생성할 수 있다.

또한, Frontend에서는 timing이 만족되더라도 Backend에서는 실제 배선 delay와 IR drop, crosstalk 등의 물리적 효과로 인해 추가적인 timing violation이 발생할 수 있어, 이를 해결하기 위한 반복적인 optimization이 필수적이다.

## 본론

\* 이번 과제에서 새롭게 추가된 icc2 directory File들을 우선 분석해보려한다.

### 2.1.1 icc2/config/

-icc2\_common\_setup.tcl

해당 파일은 PNR 공정 시작 전에 환경, 라이브러리, input file등의 설정을 담당하는 초기 세팅용 파일이다. Design 이름, include할 SUBMODULE, netlist 파일 경로 등을 설정하고, floorplanning, UPF, MCMM 등의 환경을 구축한다. 아래는 script중 일부 중요 부분을 발췌한 figure들이다.

```

50
37 #set REFERENCE_LIBRARY      [list ] ;# A list of reference libraries for the design library.
38 set REFERENCE_LIBRARY      [list \
39           $STD_CELL_RVT_NDM_DIR/cmos28hpc_1p9m_6x1z1u_13b_rvt_frame.ndm \
40           $STD_CELL_RVT_NDM_DIR/cmos28hpc_1p9m_6x1z1u_13b_rvt_physical_only.ndm \
41           $STD_CELL_RVT_NDM_DIR/tcbn28hpcplusbwp30p140tt0p9v25c_ccs.ndm \
42 ]                                ;# A list of reference libraries for the design library.
43                                     ;#   for hierarchical designs using bottom-up flows: include sub- \
44                                     ;#       for hierarchical designs using ETM+; include the ET

```

사용할 NDM(standard cell physical library)들을 선언하여 PNR에서 참조할 수 있도록 설정한다. (이번 과제에서는 RVT 공정의 라이브러리를 사용한다.

```

54 set VERILOG_NELIST_FILES [list]
55 foreach m $DESIGN_INCLUDED {
56   lappend VERILOG_NELIST_FILES "${IMPORT}/$m/results/${m}.mapped.v"
57 }
58 lappend VERILOG_NELIST_FILES    "${IMPORT}/${DESIGN_NAME}/results/${DESIGN_NAME}.mapped.v"      ;# Verilog netlist files;
59 #set VERILOG_NELIST_FILES      "${IMPORT}/${DESIGN_NAME}/results/${DESIGN_NAME}.mapped.v"      ;# Verilog netlist files;
60                                ;#          for DP: required
61                                ;#          for PNR: required if INIT_DESIGN_INPUT is ASCII in icc2_pnr_setup.tcl; not re
62 or DP_RM_NDM

```

앞선 과제에서 Design Compiler 툴을 이용해 생성한 netlist(sa4x4.mapped.v, mac8.mapped.v, mult8.mapped.v)를 가져와 추가한다.

```
74 #set TCL_MCMC_SETUP_FILE      ""      ;# Specify a Tcl script to create your corners, i  
75 set TCL_MCMC_SETUP_FILE      "./config/${DESIGN_NAME}/${DESIGN_NAME}.mcmc.top.tcl" ;# Speci  
enarios and load respective constraints;
```

MCMM 설정 파일을 지정한다. 이 mcmm.top.tcl file내부에는 전압, 온도 등의 변화에 따른 시나리오와 칩의 기능이 정의되어 있다. 이를 통해, setup과 hold time 위반 없이 설계를 최적화할 수 있다.

```
set ROUTING_LAYER_DIRECTION_OFFSET_LIST {"M1 horizontal" {M2 vertical} {M3 horizontal} {M4 vertical} {M5 horizontal} {M6 vertical} {M7 horizontal} {M8 vertical} {M9 horizontal} {AP vertical}"# Specify the routing layers as well as their direction and offset in a list of space delimited pairs."
```

Metal Layer별 배선 방향을 정의한다. M1, M3 같은 홀수 층은 horizontal, M2, M4 같은 짝수 층은 vertical로 교차 배선하였다.

*-icc2\_pnr\_setup.tcl*

해당 파일 역시 PNR 설정을 위한 다양한 변수들을 정의해둔 초기 설정 파일이다. 입력 파일 경로, 라이브러리 규칙, 공정 제약 조건 등 핵심 변수들이 정의되어 있는데 이 중 가장 주요한 몇 가지만 추려서 정리해보려 한다.

```
518 #set WRITE_GDS_LAYER_MAPFILE      "" ;# A layer map file which provides a mapping between the tool and GDS layers  
519 set WRITE_GDS_LAYER_MAPFILE      "$REFLIB_PATH/RF/APR_Tech/Synopsys/tm28clpr002s1_1_9_1a/PRTF_ICC_28nm_Syn_V19_1a/PR_tech/Synopsys/Gds0u  
ap/gdsout_6X1Z1U.map" ;# A layer map file which provides a mapping between the tool and GDS layers  
520 set WRITE_OASIS_LAYER_MAPFILE    "" ;# A layer map file which provides a mapping between the tool and OASIS layers
```

Design tool의 layer와 공정 mask layer를 매칭한다.

```
set TCL_FLOORPLAN_FILE "" ;# Tcl floorplan file written by the write_floorplan command; for example, floorplan/floorplan.tcl  
 ;# TCL_FLOORPLAN_FILE and DEF_FLOORPLAN_FILES are mutually exclusive; please specify only one of them  
 ;# Not effective if INIT_DESIGN_INPUT = DC_ASCII or DP_RM_NDM.  
 ;# The write_floorplan command writes a floorplan.tcl Tcl script and a floorplan.def DEF file;  
 ;# reading floorplan.tcl alone can restore the entire floorplan - refer to write_floorplan man for mo
```

Chip size, I/O pin의 위치, macro의 위치 등을 정의하는 floorplan file이다. (우리는 우리가 정의한 floorplanning 파일을 사용할 예정이므로 공란이다.)

```
set TCL_PG_CREATION_FILE          "./scripts/pns.tcl" ;# a Tcl scr  
;# such as create_pg*, set_pg_st
```

추후 다룰 pns.tcl 파일을 이용해 Vdd, Vss를 정의하고 연결한다.

**set CTS\_LIB\_CELL\_PATTERN\_LIST "\*/CKBD\* \*/CKND\* "**

CTS에 사용할 균형잡힌 library cell pattern들의 list를 선언한다.

Clk 신호에 사용할 선을 지정하는 NDR을 선언한다.

```
_NAME  
set CTS_NDR_MIN_ROUTING_LAYER      "M4" ;# M  
ied to.  
set CTS_NDR_MAX_ROUTING_LAYER      "M5" ;# M  
ied to.
```

Clk 신호를 배선할 layer를 설정한다. M1~M3는 저항이 높아 속도가 느리므로, 저항이 낮은 M4, M5의 중간 layer에 clk을 배선한다.

-sa4x4(sa4x4.mcmm.top.tcl

```
15 ## We are working on typical-only corner (not MCMM)  
16 ## So, unify late and early are all tt  
17 set_parasitic_parameters \  
18 -corners {tt0p9v25c} \  
19 -late_spec tt0p9v25c -late_temperature 25 \  
20 -early_spec tt0p9v25c -early_temperature 25 \  
21  
22  
23 set_operating_conditions tt0p9v25c \  
24 -analysis_type on_chip_variation \  
25 -library /tech/TSMC_HPC28/Standard_Cell/core_cell_library/gate_length_30nm/9-tr  
_power_noise/CCS/tcbn28hpcpluswp30p140_180a/tcbn28hpcpluswp30p140tt0p9v25c_ccs  
26  
27 source ../dc/${DESIGN_NAME}/results/${DESIGN_NAME}.sdc  
28  
29 set_scenario_status -none -setup true -hold false tt0p9v25c  
30 if [string length [get_defined_attributes -class scenario {dynamic_power}]] {  
31     set_scenario_status -dynamic_power true -leakage_power true tt0p9v25c;  
32 } else {  
33     set_scenario_status -power [expr true || true] tt0p9v25c;
```

해당 파일은 MCMM(multi-corner, multi-mode)를 가정하여 다양한 scenario에서의 분석을 가능하게 한다. 그러나, 우리 과제에서는 (tt/0.9V/25°C)의 single scenario를 가정하므로 크게 수정할 것은 없다. 우리가 설계한 칩은 0.9V, 25도, typical 공정 조건에서 동작하며 이 조건에 맞추어 timing, power를 최적화한다.

-sa4x4(sa4x4.tdf

```
17  
18 set_individual_pin_constraints -ports {A[0]} -allowed_layers {M4} -side 4 -width 0.05 -length 0.05 -offset 14  
19 set_individual_pin_constraints -ports {A[1]} -allowed_layers {M4} -side 4 -width 0.05 -length 0.05 -offset 16  
20 set_individual_pin_constraints -ports {A[2]} -allowed_layers {M4} -side 4 -width 0.05 -length 0.05 -offset 18  
21 set_individual_pin_constraints -ports {A[3]} -allowed_layers {M4} -side 4 -width 0.05 -length 0.05 -offset 20  
22 set_individual_pin_constraints -ports {A[4]} -allowed_layers {M4} -side 4 -width 0.05 -length 0.05 -offset 22  
23 set_individual_pin_constraints -ports {A[5]} -allowed_layers {M4} -side 4 -width 0.05 -length 0.05 -offset 24  
24 set_individual_pin_constraints -ports {A[6]} -allowed_layers {M4} -side 4 -width 0.05 -length 0.05 -offset 26  
25 set_individual_pin_constraints -ports {A[7]} -allowed_layers {M4} -side 4 -width 0.05 -length 0.05 -offset 28  
26
```

해당 파일은 sa4x4의 I/O pin의 위치를 수동으로 지정한다. 흘수 층의 metal은 horizontal, 짹수 층의 metal은 vertical로 설계된다. 또한, tdf I/O pin의 위치 뿐만 아니라 metal layer, width, length, offset 등을 설정한다. 또 다른 module을 만든다면, design에 맞는 configuration을 위해 해당 파일을 수정할 필요가 있다.

## 2.1.2 icc2/scripts/

## -init\_design.tcl

해당 파일은 Physical Design의 초기 setup을 담당하는 핵심 script이다. 현재 설정되어 있는 다양한 변수들을 sweep하면서 Place & Route를 다양하게 시도할 수 있다.

## Floorplanning

```
300      "#####
301 # modified by smkcov
302     ## Perform initialize_floorplan if neither DEF_FLOORPLAN_FILES nor TCL_FLOORPLAN_FILE is specified
303     } else {
304         puts "RM-info: creating floorplan using initialize_floorplan"
305         #initialize_floorplan -side_length {40 80} -core_offset {5} -use_site_row
306         initialize_floorplan -side_length {320 100} -core_offset {5}
307
308 #   move_objects [get_cell Ucmos10lpsvrv_ra1_hd_256x17m8] -to {10 36.5150}
309
310 #   create_keepout_margin -type hard -outer { 8.0000 2.0000 2.0000 2.0000 } {Ucmos10lpsvrv_ra1_hd_256x17m8}
311 # If you want to remove, command ::::::: remove_keepout_margins [get_keepout_margins]
312
313 #   create_placement_blockage -type hard -boundary {{10 114} {213.5 114} {213.5 35} {10 35}}
314 # If you want to remove, command ::::::: remove_placement_blockage
315
316     source "./config/${DESIGN_NAME}/${DESIGN_NAME}.tdf"
317     place_pins -self ;# to place unplaced pins if needed
318 }
```

306: core size를 설정한다. (now 320 μm(X) x 100 μm(Y))

316: 앞서 I/O pin configuration을 선언한 tdf file을 source하여 pin들을 place한다.

\* cores\_offset 5: chip boundary로부터 5μm의 margin을 유지하여 power ring 배치 or standard cell 배치(margin 안쪽)에 활용된다.

\* icc2\_shell console에서 "man create\_floorplan"을 typing하면 command description과 arguments 등의 다양한 option을 확인할 수 있고, 이를 이용해 design을 최적화할 수 있다.

## Power planning hook

```
546 # below lines are executed :: smkcov
547 #####
548 ## Power and ground network creation
549 #####
550 ## A Tcl script placeholder for your power ground network creation commands, such as create_pg*,
551 # set_pg_strategy, and compile_pg, etc.
552
553 #set TCL_PG_CREATION_FILE "/home/smkcov/QnA/digital/example_smkcov_DC_ICC2/fifo/PNR/SS/pnr/rm_icc2
554
555 ## Create standard cell PG rail
556 # Example : rm_icc2_pnr_scripts/init_design_std_cell_rail_example.tcl
557
558 # smkcov : pns.tcl includes STD Cell PG rail
559
560 if {[file exists [which $TCL_PG_CREATION_FILE]]} {
561     puts "RM-info: Sourcing [which $TCL_PG_CREATION_FILE]"
562     source -echo $TCL_PG_CREATION_FILE
563 } elseif {$TCL_PG_CREATION_FILE != ""} {
564     puts "RM-error: TCL_PG_CREATION_FILE($TCL_PG_CREATION_FILE) is invalid. Pls correct it."
565 }
566
```

564: pwr/gnd network script를 call한다.

\* 실제 PG creation commands는 pns.tcl에 적혀있기에, 단순히 source 해준다.

## End-cap cells & Tap cells

```
582 #####  
583 ## Boundary cells  
584 #####  
585  
586 ## Boundary cells: to be added around the boundaries of objects, such as voltage areas, macros, blockages, and the core area  
587 #      set_boundary_cell_rules ...  
588 #      report_boundary_cell_rules  
589 #      compile_boundary_cells  
590 #      check_boundary_cells  
591  
592 # add End cap cell  
593 set_boundary_cell_rules -left_boundary_cell tcbn28hpcplusbwp30p140tt0p9v25c_ccs/TAPCELLBWP30P140 -right_boundary_cell tcbn28hpcp  
v25c_ccs/TAPCELLBWP30P140  
594  
595  
596 #####  
597 # Fix all shaped blocks and macros  
598 #####  
599 #set_fixed_objects [get_flat_cells -filter "is_hard_macro"]  
600  
601 if [sizeof_collection [get_cells -hier -filter is_hard_macro==true -quiet]] {  
602   set_attribute -quiet [get_cells -hierarchical -filter is_hard_macro==true] status fixed  
603 }  
604 #####  
605 ## Tap cells  
606 #####  
607 # Example : create_tap_cells -lib_cell myLib/Cell1 -distance 30 -pattern every_row  
608 # Add tap cell array  
609 create_tap_cells -lib_cell tcbn28hpcplusbwp30p140tt0p9v25c_ccs/TAPCELLBWP30P140 -distance 42 -pattern every_row -skip_fixed_cell  
610 compile_boundary_cells
```

593(End-cap cells): standard cell들이 n-well을 공유하며 배치되는데, row가 끝난 시점에서도 n-well이 끊기지 않고 마감될 수 있도록 end-cap을 달아준다. (end cap이 없다면, DRC 관련 issue가 생긴다)

609(Tap cells): latch-up 현상을 방지하기 위해 Vdd, Gnd를 substrate와 well에 공급하는 tap cell을 42um 간격으로 모든 row에 배치한다. (macro는 제외)

610: 앞선 설정들을 실제로 배치한다.

-pns.tcl

해당 파일은 PNS(Power Network Synthesis)를 위한 파일로 chip의 Vdd, Gnd network를 구축한다. 해당 script를 실행하면 PG Ring, standard cell rail 삽입, PG Strap 생성 등의 task를 수행한다.

## PG Ring Creation

```

33 #####
34 #
35 # P G   R I N G   C R E A T I O N
36 #
37 #####
38 #create_pg_ring_pattern ring_pattern -parameters {h_w h_s v_w v_s} \
39   -horizontal_layer M5 -horizontal_width {@h_w} -horizontal_spacing {@h_s} \
40   -vertical_layer M4 -vertical_width {@v_w} -vertical_spacing {@v_s} \
41   -via_rule { {intersection: all} {via_master: M4_M5_via_rule} } \
42   -corner_bridge false
43
44 create_pg_ring_pattern ring_pattern -parameters {h_w h_s v_w v_s} \
45   -horizontal_layer M5 -horizontal_width {@h_w} -horizontal_spacing {@h_s} \
46   -vertical_layer M4 -vertical_width {@v_w} -vertical_spacing {@v_s} \
47   -corner_bridge false
48
49 set_pg_strategy core_ring -core \
50   -pattern {{pattern: ring_pattern}{nets: {VDD VSS}}{offset: {2 2}} {parameters:1 0.5 1 0.5}} \
51   -extension {{stop: design_boundary_and_generate_pin}}
52
53 #set_pg_strategy macro_ring -macros Ucmos10lpsvry_ra1_hd_256x17m8 \
54 #-pattern {{name : ring_pattern}{nets : {VDD VSS}}{offset : {1 1}}{skip_sides : 1 2}} \
55 #-extension {stop:innermost_ring}
56

```

Chip boundary 주변에 power ring을 생성한다. (M5 layer – horizontal, M4 layer – vertical, width = 1um, spacing = 0.5)

\* -extension stop: design\_boundary\_and\_generate\_pin option ensures that PG pins are generated at the boundary

## Standard Cell Rail Insertion

```

157 #-----
158 # S T A N D A R D   C E L L   R A I L   I N S E R T I O N
159 #
160 #####
161 create_pg_std_cell_conn_pattern \
162   std_cell_rail \
163   -layers {M1}
164 #   -rail_width {0.14 0.14}
165 #   -rail_mask {follow_pin}
166
167 set_pg_strategy rail_strat -core \
168   -pattern { {name : std_cell_rail} \
169   {nets: VDD VSS} } \
170   -extension {stop: innermost_ring}
171

```

standard cell row마다 M1 레이어에 Vdd, Vss rail을 삽입한다. 이 rail pattern은 각 cell row에 전원을 공급하는 수평 rail을 자동으로 생성하며, 가장 안쪽의 power ring까지 연결하여 standard cell들에 일관되게 전원을 공급한다.

## PG Straps

```

208 # -----
209 # S T R A P
210 #
211 #####-----#
212 #create_pg_strap -layer IB -direction vertical -width 5.0000 -net VDD -start 30 -stop 60 -pitch 30 -extend_low innermost_ring -extend_high innermost_ring \
213 -via_rule { \
214 {{existing: ring}{via_master: IA_IB_via_rule}} \
215 {{intersection: undefined} {via_master: default}} } \
216 }
217 #create_pg_strap -layer IB -direction vertical -width 5.0000 -net VSS -start 38 -stop 68 -pitch 30 -extend_low innermost_ring -extend_high innermost_ring \
218 -via_rule { \
219 {{existing: ring}{via_master: IA_IB_via_rule}} \
220 {{intersection: undefined} {via_master: default}} } \
221 }
222
223 create_pg_strap -layer M4 -direction vertical -width 0.5 -net VDD -start 15 -stop 300 -pitch 10 -extend_low innermost_ring -extend_high innermost_ring
224 create_pg_strap -layer M4 -direction vertical -width 0.5 -net VSS -start 16 -stop 301 -pitch 10 -extend_low innermost_ring -extend_high innermost_ring
225 create_pg_strap -layer M5 -direction horizontal -width 0.5 -net VDD -start 15 -stop 90 -pitch 10 -extend_low innermost_ring -extend_high innermost_ring
226 create_pg_strap -layer M5 -direction horizontal -width 0.5 -net VSS -start 16 -stop 91 -pitch 10 -extend_low innermost_ring -extend_high innermost_ring
227

```

M4 vertical strap과 M5 horizontal strap을 chip에 주기적으로 삽입한다. 이 strap들은 standard cell rail과 power ring 사이를 연결하여 낮은 저항의 power delivery network를 형성한다. 현재 설정에서는  $0.5\mu\text{m}$  너비로  $10\mu\text{m}$  pitch마다 strap을 배치하는데, 이 값들은 IR drop 감소와 routing congestion 사이의 균형을 맞추기 위해 조정 가능하다.

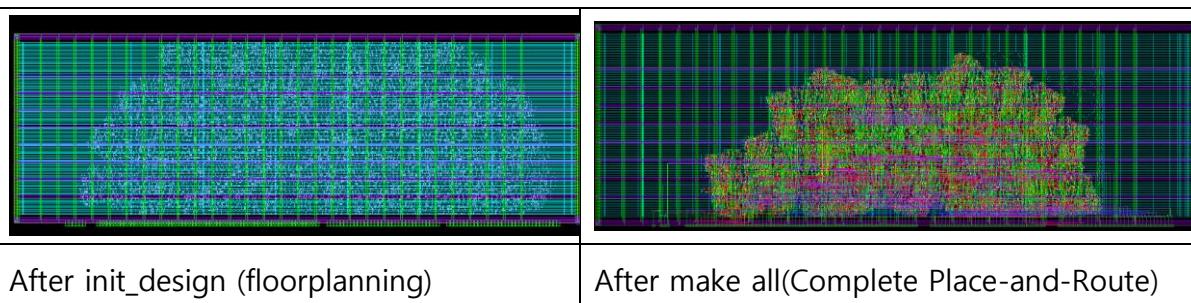
\* Strap의 width가 너무 작으면 IR drop이 증가하고, 너무 크면 routing 공간이 부족하다. 마찬가지로 pitch가 너무 넓으면 IR drop이 증가하고, 너무 좁으면 congestion이 증가한다.

\* check\_pg\_connectivity , report\_pg\_connectivity , report\_power\_plan commands를 이용해 다양한 task를 진행해 확인해 보자.

## 2.1.3 icc2/

### Floorplan / Place & Route

이제 icc2 home directory에서 make clean clean\_all init\_design 명령어를 입력하면 이전 결과를 지우고 configuration files에 근거해 다시 floorplanning하게 된다. 그리고, start\_gui command를 icc2\_shell에 입력하면 아래 그림과 같은 floorplan의 gui가 나오게 된다. GUI를 확인하면서, Floorplanned 된 chip의 core boundary, power-ring offset, pin placement, end-cap/tap cells등이 문제 없이 적용되었는지 검증한다. 그리고, icc2 shell에서 exit하고 make all을 입력하여 floorplan, PNR 모두를 진행 후 GUI를 확인할 수 있다.



이후, sa4x4 directory가 생성되고 result directory에서 다양한 report를 확인할 수 있다.

### 2.2.1 Explore to find PPA

우선, 과제3에서 optimal frequency는 2.2GHz, Max frequency는 2.4GHz(mac8\_2.6GHz, mult8\_2.4GHz)였다. 두 모듈을 이용해 default 설정으로 PNR을 진행해보았다. 2.2GHz는 1.2GHz까지, 2.4GHz는 1.3GHz까지 timing 조건을 만족했다.

Syn Freq(GHz)	Power	Area(um^2)
2.2	1.31E-02	1.12E+04
2.4	1.59E-02	1.39E+04

#### A\_0. 초기 PNR 결과 및 문제 식별

Core Size	Synthesis result	Utilization
320μm × 100μm	$1.12 \times 10^4 \mu\text{m}^2$	28.85%

초기 설계에서는 tool의 default floorplan 설정을 그대로 사용하였으며, 이로 인해 core size가 너무 크게 설정되어 있었다. 불필요하게 큰 core size로 인해 cell 간 평균 wire length 증가하면서 RC delay가 커졌고, 이는 performance 저하로 이어진다. 또한 routing resource가 필요 이상으로 분산되어 효율이 저하된다.

이에 따라, 최종 PNR 결과 70~80%대의 utilization을 만족할 수 있도록 수정하려한다.

#### A\_1. 이론적 Utilization 개선 (이론적 Utilization 계산)

적절한 core size를 결정하기 위해 이론적으로 floorplan core size를 대략 유추하였다. Synthesis된 sa4x4의 cell area는  $1.12 \times 10^4 \mu\text{m}^2$ 이다. 이에 따라 필요한 core area는 cell area에 utilization을 고려하여 약  $18,480 \mu\text{m}^2$ 로 계산되었고, core size를  $200 \mu\text{m} \times 100 \mu\text{m}$ 로 설정하고 PNR을 진행해 보았다.

#### A\_2. Error Message 발견 및 Debugging

```
tend_high innermost_ring
Error: Strap with center 215 and width 0.5 is outside design boundary. (PGR-029)
Error: Strap with center 225 and width 0.5 is outside design boundary. (PGR-029)
Error: Strap with center 235 and width 0.5 is outside design boundary. (PGR-029)
Error: Strap with center 245 and width 0.5 is outside design boundary. (PGR-029)
Error: Strap with center 255 and width 0.5 is outside design boundary. (PGR-029)
Error: Strap with center 265 and width 0.5 is outside design boundary. (PGR-029)
Error: Strap with center 275 and width 0.5 is outside design boundary. (PGR-029)
Error: Strap with center 285 and width 0.5 is outside design boundary. (PGR-029)
Error: Strap with center 295 and width 0.5 is outside design boundary. (PGR-029)
Number of threads: 8
```

Error message를 통해 PG (Power/Ground) strap의 위치가 core boundary를 벗어나는 것 이 문제임을 확인하였다.

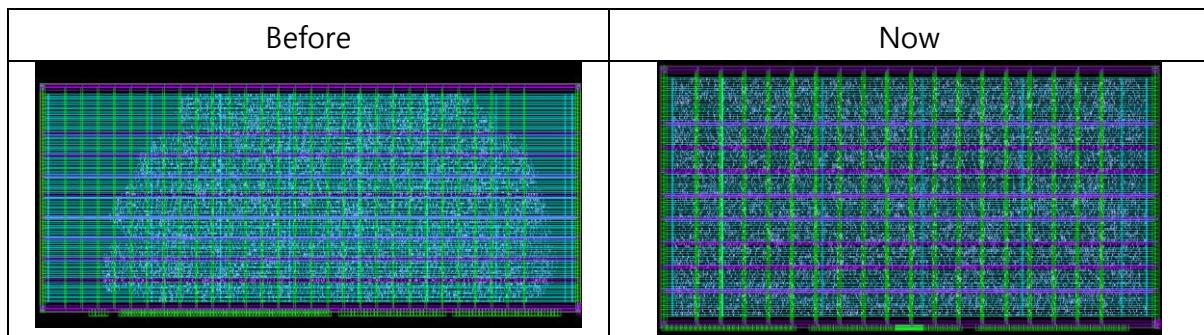
```
#It was Default strap for 320x100 design
#create_pg_strap -layer M4 -direction vertical -width 0.5 -net VDD -start 15 -stop 300 -pitch 10 -extend_low in
#create_pg_strap -layer M4 -direction vertical -width 0.5 -net VSS -start 16 -stop 301 -pitch 10 -extend_low in
#create_pg_strap -layer M5 -direction horizontal -width 0.5 -net VDD -start 15 -stop 90 -pitch 10 -extend_low in
#create_pg_strap -layer M5 -direction horizontal -width 0.5 -net VSS -start 16 -stop 91 -pitch 10 -extend_low in

#for 200x100 design
create_pg_strap -layer M4 -direction vertical -width 0.5 -net VDD -start 15 -stop 190 -pitch 10 -extend_low in
create_pg_strap -layer M4 -direction vertical -width 0.5 -net VSS -start 16 -stop 191 -pitch 10 -extend_low in
create_pg_strap -layer M5 -direction horizontal -width 0.5 -net VDD -start 15 -stop 90 -pitch 10 -extend_low in
create_pg_strap -layer M5 -direction horizontal -width 0.5 -net VSS -start 16 -stop 91 -pitch 10 -extend_low in
```

Pg\_strap의 배치를 200x100에 맞게 다시 고쳐주고 다시 시뮬레이션을 진행하였다.

```
Utilization Ratio: 0.7009
Utilization options:
- Area calculation based on: site_row of block sa4>
- Categories of objects excluded: hard_macros macro_keep
Total Area: 19972.0080
Total Capacity Area: 19972.0080
Total Area of cells: 13998.7260
Area of excluded objects:
- hard_macros : 0.0000
- macro_keepouts : 0.0000
```

icc2\_shell에서 report\_utilization 명령어를 이용해 확인해보니 floorplan 단계에서 70%의 utilization을 달성하고 timing, power 모두 향상된 모습을 확인할 수 있었다. 그러나 실제 PNR 완료 후 최종 utilization은 이보다 낮아질 수 있다.



GUI도 확인해보니 코어 utilization이 한눈에 봐도 좋아진 모습이다. 우선 이 결과를 이용해 PNR 하였다.

### A\_3 Timing Constraints

PNR을 진행하고 report를 확인해 보았을때, Hold time violation이 발생하였다. 우선 cell 간 거리가 가까워지면서 clock skew를 조정할 수 있는 여유가 부족해졌고, 이로 인해 hold margin 확보가 어려워졌다고 생각하였다. 또한 buffer insertion을 수행하기 위한 물

리적 공간이 충분하지 않아 timing 제약을 맞추기 힘들다고 생각하였다.

이러한 문제를 해결하기 위해 Opt1~5까지 다양한 방법을 시도하였는데, 그 중 효과를 본 내용을 정리하였다.

Opt3에서는 placement 단계에서의 cell 간 거리가 좁아진 것을 원인으로 판단하고, `icc2_pnr_setup.tcl`에서 `PREROUTE_PLACEMENT_MAX_DENSITY`를 0.70으로 제한하였다. 이를 통해 placement 단계에서 cell density가 강제로 낮아지면서 routing channel과 buffer insertion을 위한 공간을 확보하였다. 그 결과, 이후 optimization 단계에서 buffer 삽입이 더욱 자유롭게 가능해졌고 hold violation을 성공적으로 해결하였다.

다만 이 과정에서 최종 utilization이 43%까지 하락하는 문제가 발생하였다. 이는 목표 utilization인 60~70%에 크게 못 미쳤다.

Opt4에서는 core size를  $150\mu\text{m} \times 80\mu\text{m}$ 로 축소하였으나, floorplan 단계에서부터 utilization이 100%를 초과하는 over-utilized error가 발생하였다.

이후 core size를  $180\mu\text{m} \times 90\mu\text{m}$ 로 설정하였다. GUI 분석 결과, cell들이 매우 조밀하게 배치된 상태임을 확인할 수 있었으며, 높은 utilization은 달성되었으나 이러한 과도한 density는 때문인지 hold violation이 다시 발생하였다.

Opt5에서는 density를 추가로 완화하여 `PREROUTE_PLACEMENT_MAX_DENSITY`를 0.60으로 낮추었다. 이를 통해 buffer insertion을 위한 공간이 충분히 확보되었고, 결과적으로 hold violation은 다시 한 번 해결되었다. 그러나 이 경우에도 utilization은 약 50% 수준에 머물러 area efficiency가 낮아지는 trade-off가 발생하였다.

마지막으로 시도 6에서는 core size를  $170\mu\text{m} \times 85\mu\text{m}$ 로 설정하였다.  $160\mu\text{m} \times 80\mu\text{m}$  설정에서는 utilization error가 발생하였기 때문에 이를 완화한 설정이다. 그러나 해당 조건에서도 hold violation이 재발하였으며, placement max density를 추가로 낮추는 방식은 utilization을 과도하게 희생하게 되므로 현실적인 해결책이 아님을 확인하였다. 이에 따라, 단순한 density 조절이나 core size 변경이 아닌 새로운 접근 방식이 필요하다는 결론에 도달하였다.

## B\_0 I/O 배치 최적화

기존 설계에서는 모든 I/O port가 die의 bottom에만 집중적으로 배치되어 있었다. 이로 인해 congestion이 몰리고 clock latency가 증가하여, 이는 timing 제약으로 이어졌을 것이다. 더불어 die의 top, left, right 측면에 존재하는 routing resource가 거의 활용되지 않아, 전체 routing resource 효율이 크게 저하된 상태였다.

이러한 문제를 해결하기 위해 I/O port를 die의 네 면(top, bottom, left, right)에 분산하여 배치하였다. 특히, input port는 clock port와의 거리를 고려하여 배치하고 output port는 right, bottom에 배치하여 congestion을 최소화하였다.

이와 동시에 코어의 모양도 정사각형 모양의(110 μm x 110μm)로 수정하였다. 그 결과, utilization이 76%까지 상승하여 기존 목표 범위였던 70~80%를 달성할 수 있었다.

## B\_1 Metal Layer 최적화

Layer Name	overflow total	max	overflow (%)	# GRCs has max overflow
M1	0	0	0 ( 0.0000%)	0
M2	383	3	366 ( 2.0383%)	2
M3	0	0	0 ( 0.0000%)	0
M4	0	0	0 ( 0.0000%)	0
M5	0	0	0 ( 0.0000%)	0
M6	0	0	0 ( 0.0000%)	0
M7	0	0	0 ( 0.0000%)	0
M8	0	0	0 ( 0.0000%)	0
M9	0	0	0 ( 0.0000%)	0
AP	0	0	0 ( 0.0000%)	0
Both Dirs	383	3	366 ( 1.0192%)	2
H routing	0	0	0 ( 0.0000%)	0
V routing	383	3	366 ( 2.0383%)	2

I/O port 재배치 이후에도 일부 routing layer에서 congestion이 있어, congestion report를 기반으로 metal layer별 utilization을 확인하였다. 확인 결과, M2 layer의 사용률이 과도하게 높았으며, overflow가 발생하고 있었다.

이에 따라, I/O port의 metal layer를 M7~M9에 배치하여 tool이 하위 layer에서부터 via stack을 통해 자연스럽게 상위 layer를 활용하도록 유도하였다. 이를 통해 전체 metal layer를 보다 균형 있게 활용하고자 하였다. 그러나 icc2\_common\_setup.tcl에서 routing layer maximum을 M7에서 M9로 확장한 결과, 예상과 달리 사용되지 않았다.

Metal layer	Hor. length	% of hor.	Ver. length	% of ver.
P0	0.0000	0.0000%	0.0000	0.0000%
M1	0.0000	0.0000%	0.0000	0.0000%
M2	1322.4900	4.9901%	10716.4850	35.6756%
M3	21393.1500	80.7221%	398.5100	1.3267%
M4	38.2600	0.1444%	17241.6050	57.3980%
M5	3286.9300	12.4025%	1.4100	0.0047%
M6	0.0000	0.0000%	1680.6800	5.5951%
M7	461.4050	1.7410%	0.0000	0.0000%
M8	0.0000	0.0000%	0.0000	0.0000%
M9	0.0000	0.0000%	0.0000	0.0000%
AP	0.0000	0.0000%	0.0000	0.0000%

생각해보니, M8과 M9는 두꺼운 layer로 RC delay가 커 setup violation을 유발할 가능성이 높아 사용이 제한되는 것으로 보였다. 이에 따라 상대적으로 M6 및 M7 layer를 중심으로 routing을 해보았으나 큰 의미가 있지는 않았다.

최종적으로는 tdf 파일에서 I/O port에 대해 하나의 metal layer를 강제하는 대신 set\_individual\_pin\_constraints -ports {Cout} -allowed\_layers {M5 M6}, 이처럼 설정하여 복수 layer 중 하나를 선택하도록 설정하였다. 이를 통해 하위 layer인 M2에서 발생하던 congestion이 조금 줄어들기는 하였다.

## B\_2 Performance 및 Power 최적화

110 μm x 110μm core size에서 최종적으로 76%의 utilization이 나오니 area는 여기서 만족하고, timing met을 맞추면서 performance 향상과 동시에 타이밍 위반을 막기 위해 clk을 Core의 중심부(55μm)에 위치 시켰고 I/O Port도 standard cell과 가까운 metal layer 인 M2, M3를 사용하였다. setup violation이 나오면 앞서 설정한 PREROUTE\_PLACEMENT\_MAX\_DENSITY 키워 cell들 간의 거리를 벌렸고, hold violation이 나오면 cell 들간의 거리를 좁히는 등의 방법을 사용하였다.

```

78 #####
79 # SIDE 2 (TOP/NORTH) : Control Signals
80 # Centered for 110x110 core
81 #####
82 set_individual_pin_constraints -ports {clk}      -side 2 -offset 55.0 -allowed_layers {M3}
83 set_individual_pin_constraints -ports {rstn}     -side 2 -offset 57.0 -allowed_layers {M3}
84 set_individual_pin_constraints -ports {en}       -side 2 -offset 59.0 -allowed_layers {M3}
85 set_individual_pin_constraints -ports {req_out}  -side 2 -offset 61.0 -allowed_layers {M3}
86

```

또한, power 개선을 위해 pg strap의 width와 pitch를 조절하고 다양한 metal layer 조합을 시도하는 등의 최적화를 진행함과 동시에 timing 조건에 맞는 조합을 수차례 시도하였다.

```

36 #for 110x110 design
37 create_pg_strap -layer M4 -direction vertical -width 0.5 -net VDD -start 10 -stop 100 -pitch 25 -extend_low innermost_ring -extend_high innermost_ring
38 create_pg_strap -layer M5 -direction vertical -width 0.5 -net VSS -start 11 -stop 101 -pitch 25 -extend_low innermost_ring -extend_high innermost_ring
39 create_pg_strap -layer M4 -direction horizontal -width 0.5 -net VDD -start 10 -stop 100 -pitch 25 -extend_low innermost_ring -extend_high innermost_ring
40 create_pg_strap -layer M5 -direction horizontal -width 0.5 -net VSS -start 11 -stop 101 -pitch 25 -extend_low innermost_ring -extend_high innermost_ring
41
42

```

## 결론

<b>init_design.tcl</b>	<b>s0</b>	<b>s1(Optimal)</b>	<b>s1_</b>	<b>s2</b>	<b>s3</b>
X,Y dim	110x110	110x110	110x110	110x110	110x110
core offset	5	5	5	5	5
<b>pns_tcl(M4,M5)</b>					
Ring Width	1	1	1	1	1
Ring Spacing	0.5	0.5	0.5	0.5	0.5
Strap Width	0.5	0.5	0.5	0.5	0.5
Strap Pitch	25	25	25	25	25
<b>sa4x4.sdc</b>					
clk Freq	1.2GHz	1.25GHz	1.2GHz	1.3GHz	1.3GHz
<b>icc2_pnr_setup.tcl</b>					

MAX_DENSITY	0.7	0.7	0.7		0.7
<b>sa4x4.tdf(M2,M3)</b>					
Routing	NSWE	NSWE	NSWE	NSWE	NSWE
Clk Signal	85um	85um	55um	85um	75um/95um
<b>Timing</b>	Met	Met	Not Met	Not Met	Not Met

<b>power</b>	8.57E+06	8.87E+06			
<b>area</b>	12067.06	12067.06			
<b>performance</b>	1.2GHz	1.25GHz			

\* Design choice variations are marked in red

### **Summary of your team's PNR process & how design choices influenced the results**

최종적으로 정리하면 이번 과제에서는 PPA 최적화를 위해 단계적으로 접근하였다. 먼저 area 측면에서 core size를 점진적으로 축소하며( $200 \times 100 \rightarrow 180 \times 90 \rightarrow 110 \times 110$ ) 목표 utilization 70-80% 범위를 만족하는 floorplan을 확보하였다. Area가 고정된 후에는 performance 최적화 단계로 진입하여, I/O port와 clock port 위치 조정, placement density 제어 등을 통해 setup/hold timing violation을 해결하며 최대 동작 주파수를 찾았다. 마지막으로 timing이 만족된 상태에서 power 최적화를 수행하였으며, PG strap의 metal layer, width, pitch의 여러 조합을 시도하여 최적화를 수행하였다.

표를 보면 clk signal을 단순히 정중앙에 위치시키는 것이 아니라, hold/setup violation 유무에 따라 적절한 위치에 배치되어야 timing 조건을 만족하고 있음을 보여준다. 그리고, s2의 경우 setup violation이 발생하여 s3에서 clk signal의 위치를 바꾸며 timing constraint를 해결하려한 모습이다.

## final PPA values (/w 2.2GHz synthesized sa4x4 netlist)

최적점 합성 결과: team\_342106/assn4\_mw/icc2/sa4x4\_s1(Optimal)

(각자 Assn3 최적 frequency도 달랐고 많은 것을 시도하다 보니 최종 PPA value가 상이합니다.)

Power[nW]	Performance[GHz]	Area[μm^2]
<pre>Total Power   ( * ) 0.0000e+00   ( 0.0%) 0.0000e+00   ( 0.0%) 1.1121e+02   ( 0.0%) 7.5267e+06   ( 84.8%) 5.7245e+05   ( 6.4%) 0.0000e+00   ( 0.0%) 7.7869e+05   ( 8.8%)  8.8779e+06 nW</pre> <p>8.88E+06</p>	<pre>Attributes:   G - Propagated clock   G - Generated clock   U - Unexpanded generated clock  Clock          Period    Waveform      Attrs      Sources clk            0.8000 {0 0.4}        p          {clk} #1</pre> <p>1.25</p>	<pre>***** Utilization Ratio: 0.7630 Utilization options: - Area calculation based on: site_ - Categories of objects excluded: ha Total Area: 12067.0200 Total Capacity Area: 12067.0200 Total Area of cells: 9206.8200 Area of excluded objects:</pre> <p>12067.06</p>

\*sa4x4\_s1(Optimal)/results/chip\_finish.report\_clocks, sa4x4\_s1(Optimal)//results/chip\_finish.report\_power,  
sa4x4\_s1(Optimal)/results/chip\_finish.report\_utilization

## Justification for Our PPA Point

### 1. Area

```
*****
7 Utilization Ratio: 0.7630
8 Utilization options:
9 - Area calculation based on: site_row of b
0 - Categories of objects excluded: hard_macros
1 Total Area: 12067.0200
2 Total Capacity Area: 12067.0200
3 Total Area of cells: 9206.8200
4 Area of excluded objects:
5 - hard_macros : 0.0000
6 - macro_keepouts : 0.0000
7 - soft_macros : 0.0000
8 - io_cells : 0.0000
9 - hard_blockages : 0.0000
0 Total Area of excluded objects: 0.0000
1 Ratio of excluded objects: 0.0000
2
3 Utilization of site-rows with:
4 - Site 'unit': 0.7630
5
6 0.7630
7
```

Utilization 76.3%는 초기 목표로 삼았던 70-80% 범위 내에 위치하며, Area efficiency가 효율적인 design을 달성했다. 또한, timing met을 위한 buffer 삽입 공간이 확보되어 있어 setup timing 관리에서 met이 가능했다. 또한, routing congestion이 적절히 관리되어 DRC violation 없이 routing이 완료되었다. 만약 utilization을 80% 이상으로 높였다면 routing congestion으로 인해 timing이 더욱 악화되어 1.25GHz 달성을 불가능했을 것이다. 반대로 70% 이하로 낮췄다면 area를 15,000μm<sup>2</sup> 이상으로 낭비하게 되었을 것이다.

### 2. Performance

1.25GHz는 timing met을 달성할 수 있는 실질적인 최대 주파수이다. setup slack과 hold slack이 각각 +11.2ps와 +0.7ps로 빠빠하며, 특히 critical path는 +0.7ps로 거의 한 계에 도달했다. 더 높은 frequency를 시도했을 때 timing violation이 발생하여 PNR이 실패하였으므로, 1.25GHz가 현재 design과 floorplan 조건에서 달성 가능한 maximum

performance이다. 참고로, 1.3GHz로 timing met에 성공하긴 하였지만(sa4x4\_2.2\_110x110\_1.3G\_M2\_clk\_mid), power가 9.60E+06이 나와 실질적인 최대 주파수로 1.25GHz를 선정했다.

### 3. Power

```

inf, value = 0 048409 (POW-046)
Note - message 'POW-046' limit (10) exceeded. Remainder will be suppressed.

Cell Internal Power      = 7.0009e+06 nW ( 79.3%)
Net Switching Power     = 1.8306e+06 nW ( 20.7%)
Total Dynamic Power      = 8.8315e+06 nW (100.0%)
Cell Leakage Power       = 4.6396e+04 nW

Attributes
  u - User defined power group
  i - Includes clock pin internal power

Power Group      Internal Power      Switching Power      Leakage Power      Total Power      ( % )      Attrs
io_pad           0.0000e+00          0.0000e+00          0.0000e+00          0.0000e+00      ( 0.0%)
memory          0.0000e+00          0.0000e+00          0.0000e+00          0.0000e+00      ( 0.0%)
black_box        0.0000e+00          0.0000e+00          1.1121e+02          1.1121e+02      ( 0.0%)
clock_network   6.1939e+06          1.3319e+06          9.3558e+02          7.5267e+06      ( 84.8%)      i
register         4.2813e+05          1.2927e+05          1.5042e+04          5.7245e+05      ( 6.4%)
sequential       0.0000e+00          0.0000e+00          0.0000e+00          0.0000e+00      ( 0.0%)
combinational   3.7887e+05          3.6951e+05          3.0308e+04          7.7869e+05      ( 8.8%)
Total            7.0009e+06 nW      1.8306e+06 nW      4.6396e+04 nW      8.8779e+06 nW
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
279
280
281
282
283
284
285
286
287
288
289
289
290
291
292
293
294
295
296
297
298
299
299
300
301
302
303
304
305
306
307
308
309
309
310
311
312
313
314
315
316
317
318
319
319
320
321
322
323
324
325
326
327
328
329
329
330
331
332
333
334
335
336
337
338
339
339
340
341
342
343
344
345
346
347
348
349
349
350
351
352
353
354
355
356
357
358
359
359
360
361
362
363
364
365
366
367
368
369
369
370
371
372
373
374
375
376
377
378
379
379
380
381
382
383
384
385
386
387
388
389
389
390
391
392
393
394
395
396
397
398
399
399
400
401
402
403
404
405
406
407
408
409
409
410
411
412
413
414
415
416
417
418
419
419
420
421
422
423
424
425
426
427
428
429
429
430
431
432
433
434
435
436
437
438
439
439
440
441
442
443
444
445
446
447
448
449
449
450
451
452
453
454
455
456
457
458
459
459
460
461
462
463
464
465
466
467
468
469
469
470
471
472
473
474
475
476
477
478
479
479
480
481
482
483
484
485
486
487
488
489
489
490
491
492
493
494
495
496
497
498
499
499
500
501
502
503
504
505
506
507
508
509
509
510
511
512
513
514
515
516
517
518
519
519
520
521
522
523
524
525
526
527
528
529
529
530
531
532
533
534
535
536
537
538
539
539
540
541
542
543
544
545
546
547
548
549
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
698
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
787
788
789
789
790
791
792
793
794
795
796
796
797
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
878
879
879
880
881
882
883
884
885
886
887
887
888
888
889
889
890
891
892
893
894
895
895
896
896
897
897
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
918
918
919
919
920
921
922
923
924
925
926
927
928
928
929
929
930
931
932
933
934
935
936
937
938
938
939
939
940
941
942
943
944
945
946
947
948
948
949
949
950
951
952
953
954
955
956
957
958
958
959
959
960
961
962
963
964
965
966
967
968
968
969
969
970
971
972
973
974
975
976
977
977
978
978
979
979
980
981
982
983
984
985
986
986
987
987
988
988
989
989
990
991
992
993
994
995
996
997
998
999
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1038
1039
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1048
1049
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1068
1069
1069
1070
1071
1072
1073
1074
1075
1076
1077
1077
1078
1078
1079
1079
1080
1081
1082
1083
1084
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1091
1092
1093
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1101
1102
1103
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1111
1112
1113
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1320
1321
1321
1322
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1330
1331
1331
1332
1332
1333
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1340
1341
1341
1342
1342
1343
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1350
1351
1351
1352
1352
1353
1353
1354
1354
1355
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1360
1361
1361
1362
1362
1363
1363
1364
1364
1365
1365
1366
1366
1367
1367
1368
1368
1369
1369
1370
1370
1371
1371
1372
1372
1373
1373
1374
1374
1375
1375
1376
1376
1377
1377
1378
1378
1379
1379
1380
1380
1381
1381
1382
1382
1383
1383
1384
1384
1385
1385
1386
1386
1387
1387
1388
1388
1389
1389
1390
1390
1391
1391
1392
1392
1393
1393
1394
1394
1395
1395
1396
1396
1397
1397
1398
1398
1399
1399
1400
1400
1401
1401
1402
1402
1403
1403
1404
1404
1405
1405
1406
1406
1407
1407
1408
1408
1409
1409
1410
1410
1411
1411
1412
1412
1413
1413
1414
1414
1415
1415
1416
1416
1417
1417
1418
1418
1419
1419
1420
1420
1421
1421
1422
1422
1423
1423
1424
1424
1425
1425
1426
1426
1427
1427
1428
1428
1429
1429
1430
1430
1431
1431
1432
1432
1433
1433
1434
1434
1435
1435
1436
1436
1437
1437
1438
1438
1439
1439
1440
1440
1441
1441
1442
1442
1443
1443
1444
1444
1445
1445
1446
1446
1447
1447
1448
1448
1449
1449
1450
1450
1451
1451
1452
1452
1453
1453
1454
1454
1455
1455
1456
1456
1457
1457
1458
1458
1459
1459
1460
1460
1461
1461
1462
1462
1463
1463
1464
1464
1465
1465
1466
1466
1467
1467
1468
1468
1469
1469
1470
1470
1471
1471
1472
1472
1473
1473
1474
1474
1475
1475
1476
1476
1477
1477
1478
1478
1479
1479
1480
1480
1481
1481
1482
1482
1483
1483
1484
1484
1485
1485
1486
1486
1487
1487
1488
1488
1489
1489
1490
1490
1491
1491
1492
1492
1493
1493
1494
1494
1495
1495
1496
1496
1497
1497
1498
1498
1499
1499
1500
1500
1501
1501
1502
1502
1503
1503
1504
1504
1505
1505
1506
1506
1507
1507
1508
1508
1509
1509
1510
1510
1511
1511
1512
1512
1513
1513
1514
1514
1515
1515
1516
1516
1517
1517
1518
1518
1519
1519
1520
1520
1521
1521
1522
1522
1523
1523
1524
1524
1525
1525
1526
1526
1527
1527
1528
1528
1529
1529
1530
1530
1531
1531
1532
1532
1533
1533
1534
1534
1535
1535
1536
1536
1537
1537
1538
1538
1539
1539
1540
1540
1541
1541
1542
1542
1543
1543
1544
1544
1545
1545
1546
1546
1547
1547
1548
1548
1549
1549
1550
1550
1551
1551
1552
1552
1553
1553
1554
1554
1555
1555
1556
1556
1557
1557
1558
1558
1559
1559
1560
1560
1561
1561
1562
1562
1563
1563
1564
1564
1565
1565
1566
1566
1567
1567
1568
1568
1569
1569
1570
1570
1571
1571
1572
1572
1573
1573
1574
1574
1575
1575
1576
1576
1577
1577
1578
1578
1579
1579
1580
1580
1581
1581
1582
1582
1583
1583
1584
1584
1585
1585
1586
1586
1587
1587
1588
1588
1589
1589
1590
1590
1591
1591
1592
1592
1593
1593
1594
1594
1595
1595
1596
1596
1597
1597
1598
1598
1599
1599
1600
1600
1601
1601
1602
1602
1603
1603
1604
1604
1605
1605
1606
1606
1607
1607
1608
1608
1609
1609
1610
1610
1611
1611
1612
1612
1613
1613
1614
1614
1615
1615
1616
1616
1617
1617
1618
1618
1619
1619
1620
1620
1621
1621
1622
1622
1623
1623
1624
1624
1625
1625
1626
1626
1627
1627
1628
1628
1629
1629
1630
1630
1631
1631
1632
1632
1633
1633
1634
1634
1635
1635
1636
1636
1637
1637
1638
1638
1639
1639
1640
1640
1641
1641
1642
1642
1643
1643
1644
1644
1645
1645
1646
1646
1647
1647
1648
1648
1649
1649
1650
1650
1651
```

	<b>1.4GHz</b>	<b>1.8GHz</b>	<b>2.2GHz</b>
Performance	0.9GHz	1GHz	1.25GHz
Power	5.15E+06	6.24E+06	8.88E+06
Area	8168	10979.01	12067.06

위의 표에서 synthesis frequency에 따른 아래와 같은 netlist 특성이 명확하게 최종 결과에 반영되었음을 확인할 수 있다.

1. synthesis frequency가 높을수록 최종 PNR frequency도 높아진다 ( $0.9\text{GHz} \rightarrow 1.0\text{GHz} \rightarrow 1.25\text{GHz}$ ). 이는 높은 synthesis frequency로 합성된 netlist가 큰 cell과 많은 buffer를 포함하여 critical path delay가 짧기 때문이다.
2. synthesis frequency가 높을수록 power consumption도 비례하여 증가한다 ( $5.15\text{mW} \rightarrow 6.24\text{mW} \rightarrow 8.88\text{mW}$ ). Large cell은 내부 capacitance가 크고 leakage current가 높으며, 많은 buffer는 switching power를 증가시키기 때문이다.
3. synthesis frequency가 높을수록 area도 증가한다 ( $8,168\mu\text{m}^2 \rightarrow 10,979\mu\text{m}^2 \rightarrow 12,067\mu\text{m}^2$ ). Large cell과 buffer의 물리적 크기가 크고, timing을 맞추기 위한 additional logic도 증가하기 때문이다.

조원 3명의 서로 다른 PPA point는 각각이 특정 domain에서의 optimization objective에 따라 최적의 PPA가 될 수 있다.

### **Additional discussion on potential methods for further PPA improvement**

이번 과제에서는 utilization 목표를 70%대로 aggressive하게 잡고 최적화를 시도하니 power/performance 측면에서 다양한 시도를 해보았음에도 timing 잡기가 쉽지 않았다. 이는 placement 밀도가 높아 routing congestion과 buffer 삽입 공간이 부족하여 생긴 문제였던 것 같다.

다음에는 area를 처음부터 강하게 제한하기 보다는 상대적으로 여유 있는 core size에서 performance를 우선적으로 확보한 후, timing margin을 기반으로 area를 축소하는 방식의 접근이 더욱 효과적일 것으로 보인다. 즉, 초기 단계에서는 timing margin을 확보하고, 이후 simulation을 반복하는 과정을 통해 area를 점진적으로 조정함으로써 PPA 간의 균형을 맞추는 전략이 필요할 것으로 생각된다.

과제3에서 optimal frequency로 2.2GHz가 선정되어 해당 netlist를 기반으로 PNR을 진

행하였다. 만약 더 낮은 주파수에서 합성된 netlist를 사용하였다면, synthesis 단계에서의 timing constraint이 완화되어 gate upsizing이나 buffer 삽입이 줄어들고, 이에 따라 cell count와 routing complexity가 감소하여 hold timing/Power/Area 측면에서 이점을 가져갈 수 있었을 것으로 생각된다.

과제4에서는 sa4x4 모듈 단독 합성을 기준으로 floorplan 및 I/O 배치를 설계하였으나, 실제 설계에서는 해당 모듈이 단독으로 사용되는 경우는 거의 없으며, 최종적으로는 상위 Top module에서 통합되기 마련이다. 따라서 향후 설계에서는 top level에서의 연결 구조, 데이터 흐름, clock distribution, 그리고 주변 블록과의 관계를 사전에 고려한 설계가 필수적이다.

## Assesment

한민준- 디버깅 시 안되는 것이 있으면 잘 대답해주었으며, 꼭 확인해보아야할 명령어들을 알려주었음.

주형훈- 과제4 기본 환경 세팅 및 다양한 작업들을 먼저 진행해주었으며, 그로 인해 얻은 다양한 팁들을 공유해줌.