

# ICLab Final Project Checklist:

G18 謝霖泳、劉亦傑、林暄傑

## Checklist

| Design Stage                                  | Description   | Fin | Comment   |
|---|---|-----|---|
| Software (/SW)                                | image_preprocess.py<br>gaussian.py<br>med.py  | y   | image_preprocess.py 產生 tb 和 golden<br>gaussian.py, med.py 驗證 gaussian filter 和<br>median filter 的算法及產生小 module 的 tb   |
| Software (/SW)<br>RTL implementation (/RTL)   | Test pattern  | y   | flower_638_482.jpg 是原圖<br>noise_638_482.txt 是加入雜訊的電路 input<br>med_of_med_636_480.txt 是測 median<br>filter 的小 tb<br>gaussian_denoise_636_480.txt 是測<br>Gaussian filter 的小 tb<br>denoise_hog_with_sqrt_636_480.txt 是我們<br>在 median filter + HOG + sqrt 的 input<br>gaussian_hog_with_sqrt_636_480.txt 是我<br>們在 Gaussian filter + HOG + sqrt 的 input |
|   | Source file<br><Verilog file (*.v) >  | y   |   |
| RTL implementation (/RTL)<br>Synthesis (/SYN) | Simulation file<br><testbench (*.v)<br>ncverilog.log, mist.txt>   | y   | 請詳見 misc.txt  |
|   | Spyglass report<br><spyglass.rep>   | y   |   |
|   | DC<br>< netlist (*.v),<br>synthesis scripts(*.tcl),<br>synthesis log(*.log),<br>reports(*.rep),misc.txt>    | y   | report_area_top.out => area report<br>report_timing_top.out => timing report<br>report_power_top.out => power report<br>syn.log => synthesis log file<br>其餘細節請詳見 misc.txt   |
| Place and Route (/APR)                        | ICC<br><netlist (*.v), delay<br>(*.sdf)<br>P&R log (*.log), reports<br>(*.rep), DRC&LVS<br>results (*.rep)> | y   | icc_output.log => P&R log<br>CHIP_layout.sdf => delay file(放在/APR)<br>*.rep => 包含 area, power, timing,<br>DRC&LVS 其餘細節請見 misc.txt   |
| P&R (/APR)                                    | Post-Simulation<br>< testbench (*.v)<br>ncverilog.log>  | y   | 請見 misc.txt   |
| P&R (/APR)<br>pdf document.<br>(/)            | Power Analysis<br><pt_power.rep,<br>scripts(*.tcl)>   | y   | 請見 misc.txt   |
|   | 1.放入這份<br>G18_Checklist.pdf,<br>G18_finalproject.pdf  | y   |   |

## Software Usage

在 SW 底下，共有三個.py 檔，主要依靠 image\_preprocess.py 來對輸入圖片進行加入雜訊、產生 golden pattern、testbench 的功能。而在設計過程會需要進行另外兩個 filter 的驗證，因此額外有 gaussian.py 和 med.py 分別去驗證我們的算法。

Test pattern 的部分，noise\_638\_482.txt 會於 testbench 作為我們的 top module input，而 golden 對比上則共分為四個.txt 檔如下：

1. med\_of\_med\_636\_480.txt 測試至 median filter 的階段結果
2. gaussian\_denoise\_636\_480.txt 測試至 Gaussian filter 的階段結果
3. denoise\_hog\_with\_sqrt\_636\_480.txt 測試 median filter 含 HOG 的結果
4. gaussian\_hog\_with\_sqrt\_636\_480.txt 測試 Gaussian filter 含 HOG 的結果

在 Simulation 的部分，我們皆於 sim 資料夾下完成，目錄下共有兩個 testbench，test\_top.v 會包含 pre-sim 和 gate-sim，test\_CHIP.v 則包含 posim。

詳細指令整理如下

### **median filter:**

pre-sim: ncverilog -f sim.f +define+MED --> presim\_MED\_ncverilog.log

gate-sim: ncverilog -f gatesim.f +define+MED -> gate\_sim\_MED\_ncverilog.log

postsim: ncverilog -f postsim.f +define+MED -> postsim\_MED\_ncverilog.log

### **Gaussian filter:**

pre-sim: ncverilog -f sim.f +define+GAUSS -> gate\_sim\_MED\_ncverilog.log

gate-sim: ncverilog -f gatesim.f +define+GAUSS -> gate\_sim\_GAUSS\_ncverilog.log

postsim: ncverilog -f postsim.f +define+GAUSS -> postsim\_guassian\_ncverilog.log

## Project path

**Our Project Path:** /home/u107/u107061246/ICLAB/final

## Filetree

我們的 project folder 名稱為 final

hdl: 放置 RTL 檔案

top.v: 最後完成電路的 top module

denoise.v: median filter

median.v: median 運算電路

Gaussian.v: Gaussian filter

MUX.v: 選擇 mode (median or Gaussian)

HOG.v: HOG 電路

hog\_counter.v: 輔助 HOG 的 counter 電路

sqrt.v: square root 電路

spyglass\_final\_project.rpt: spyglass report

sim: 放置模擬的檔案 (Note: 模擬所需的 test pattern 放在 sim 的 golden, pix)

test\_top.v 主要模擬 pre-sim & gate-sim 的 testbench

test\_CHIP.v 主要模擬 posim 的 testbench

sim.f: 由 test\_top.v 使用 (可選 mode)

gatesim.f: 由 test\_top.v 使用 (可選 mode)

postsim.f 由 test\_CHIP.v 使用 (可選 mode)

golden: 放置 golden pattern

denoise\_hog\_with\_sqrt\_636\_480.txt

med\_of\_med\_636\_480.txt

gaussian\_denoise\_636\_480.txt

gaussian\_hog\_with\_sqrt\_636\_480.txt

pix: 放置含 noise 的 pattern

noise\_638\_482.txt

\*.fsdb 為產生的波型檔

syn: 放置 DC 產生的檔案

\*.tcl: 跑 synthesis 的檔案

report: 包含 timing, area, power report

netlist: 包含 DC 產生的 top\_syn.v, top\_syn.sdc, top\_syn.sdf (for gate-sim)

apr\_0.7: (我們嘗試了很多版本的 APR，最後請助教於 [apr\\_0.7](#) 資料夾查看最終結果)

io\_pin.tdf: I/O pin assignment 檔案

run.tcl: 跑所有 APR 的 script (包含 0~6)

CHIP\_syn.v: 拿去跑 APR 的 synthesis netlist

CHIP\_syn.sdc: 拿去跑 APR 的 synthesis constraint

post\_layout:

CHIP.gds: layout file

CHIP\_layout.v: APR 產生的 netlist

CHIP\_layout.sdf: delay file 用於 post-sim

CHIP\_layout.sdc: constraint file, 用於 power analysis

CHIP\_layout.max.gz: SPEF file，用於 power analysis

Primetime:

presim\_wv: 針對 pre-sim fsdb 進行 power analysis

postsim\_wv: 針對 post-sim fsdb 進行 power analysis