National Cheng Kung University

Department of Electrical Engineering

Introduction to VLSI CAD (Spring 2024)

Lab Session 7

Design of Local Binary Pattern Facial Recognition System

Name	Student ID	
鄭喆嚴	E14096724	
Practical	Points	Marks
Lab 7_1	30	
Lab 7_2	65	
Report	5	
Demo	10	
Notes		

Due: 15:00 May 1, 2024@ moodle

Summary

Hardware		
TOP	RTL(∨/X)	Synthesis(∨/X)
Lab7_1	∨	∨
Lab7_2	∨	∨

Synthesis result		
Clock period(ns)	Area	Simulation time (ns)
2.0	18580.545117	25454060.035

Superlint(number of inline messages, just write down the final design result, i.e. if you only finish lab7_1, write your Superlint result of lab7_1, otherwise, write down lab7_2 only)			
Total lines	Warning	Error	coverage(%)
2061	29	0	98.59%

Note: You must complete and fill out this form with your design information!!!

Deliverables

1) All Verilog codes including testbenches for each problem should be uploaded.
   NOTE: Please **DO NOT** include source code in the paper report!
2) All homework requirements should be uploaded in this file hierarchy.
3) NOTE: 1. Please **DO NOT** upload waveforms (.fsdb or .vcd)!
4) If you upload a dead body which we can't even compile, you will get NO credit!
5) All Verilog file should get at least 90% SuperLint Coverage.
6) All homework requirements should be uploaded in this file hierarchy or you will not get full credit, if you want to use some sub modules in your design but you do not include them in your tar file, you will get 0 point.

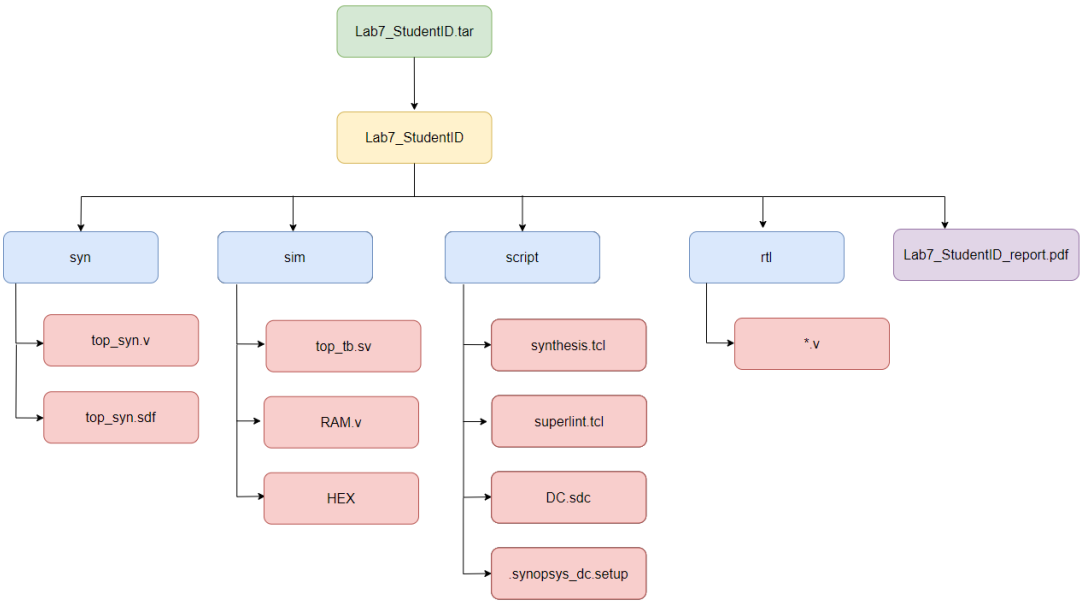
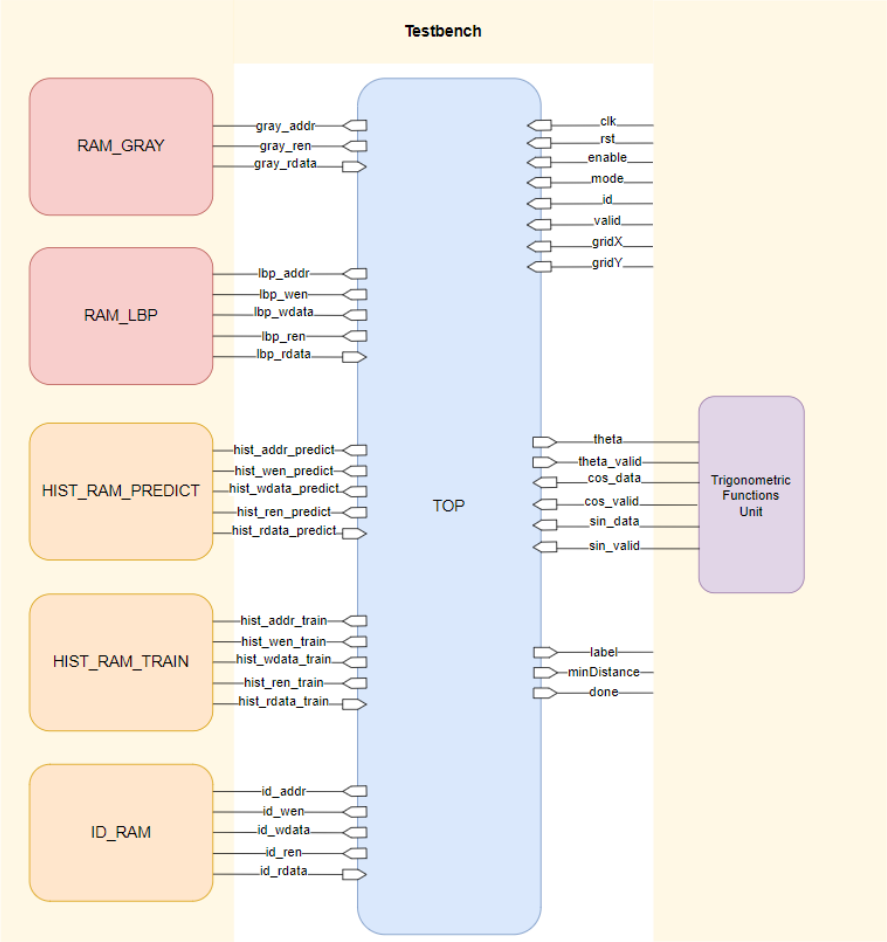
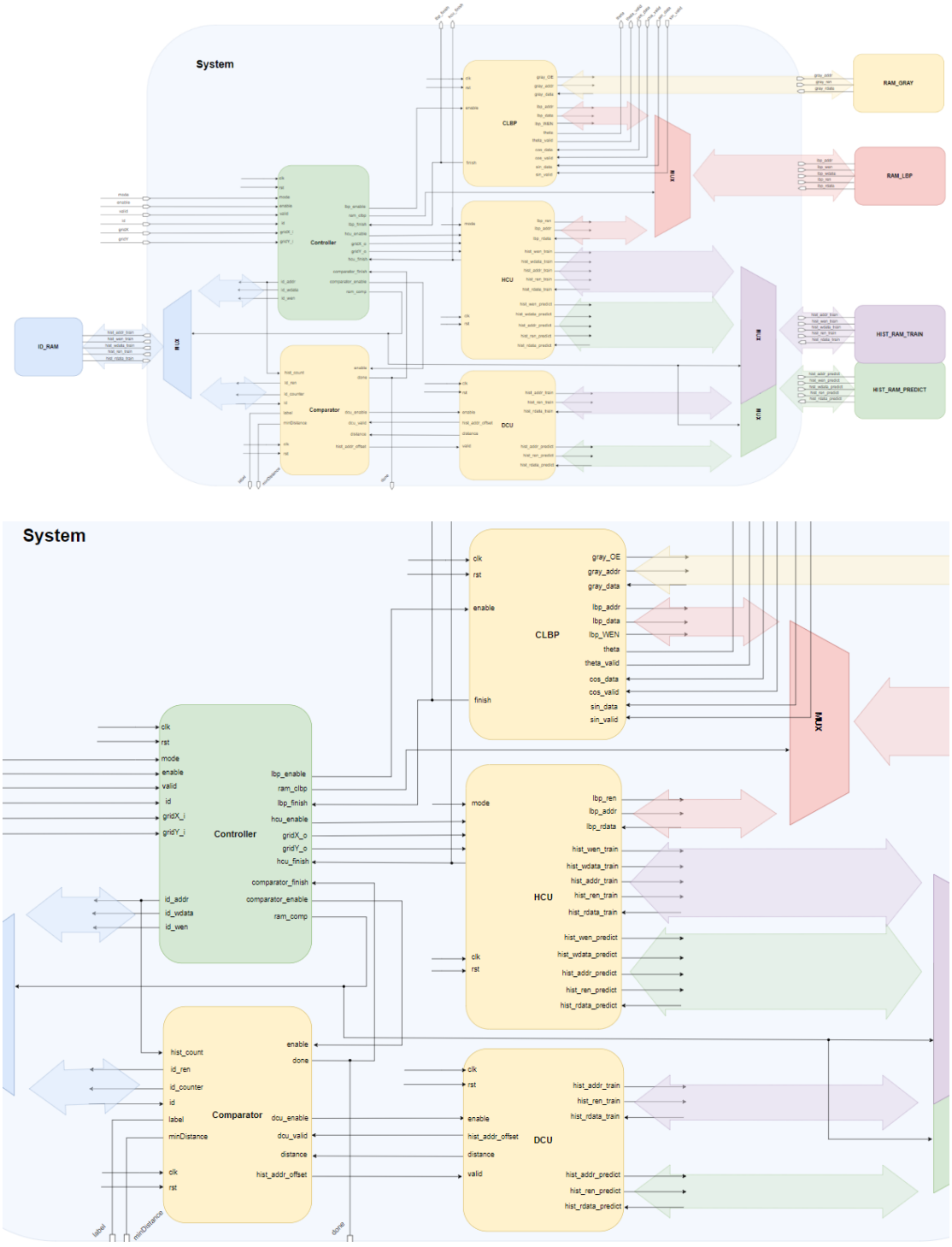


Fig.1 File hierarchy for Homework submission

You are about to integrate all components (CLBP, HCU, Controller…) to form a LBP facial recognition system. The block diagram of system is as shown in **Fig2** and **Fig3**.



▲Fig2. The block diagram of system (external)



▲Fig3. The block diagram of system (internal)

- **Port list of top module:**
- TOP

Signal	I/O	Bit-width	Description
clk	I	1	Clock signal
rst	I	1	Reset signal
enable	I	1	Circuit enabling signal
mode	I	1	Indication signal of whether the system is in prediction or training phase, 0 is training, 1 is prediction
gridX	I	4	Image sliced portion in X direction, value is 8
gridY	I	4	Image sliced portion in Y direction, value is 8
valid	I	1	Indication that current subject ID is valid
id	I	5	Subject ID
hcu_finish	O	1	Indication that HCU circuit is done(should be asserted every time a subject picture is finished computing histogram)
label	O	5	Prediction result, output ID value
minDistance	O	18	Minimum distance between the prediction histogram and the closest histogram in HIST_TRAIN_RAM
done	O	1	Indication that prediction of one picture is finished

Signal	I/O	Bit-width	Description
gray_addr	O	12	Address signal connected to RAM_GRAY
gray_ren	O	1	Read enable signal to RAM_GRAY
gray_rdata	I	8	Read data signal from RAM_GRAY
lbp_addr	O	12	Address signal connected to RAM_LBP
lbp_wen	O	1	Write enable signal to RAM_LBP
lbp_wdata	O	8	Write data signal to RAM_LBP
lbp_ren	O	1	Read enable signal to RAM_LBP
lbp_rdata	I	8	Read data signal from RAM_LBP
theta	O	25(fixed-point)	Current neighbor's theta signal(unit is in radian)
theta_valid	O	1	Indication signal of current neighbor's theta is valid
cos_data	I	25(fixed-point)	Cosine value of the theta(from testbench)
cos_valid	I	1	Indication signal of cosine value is valid
sin_data	I	25(fixed-point)	Sine value of the theta(from testbench)
sin_valid	I	1	Indication signal of sine value is valid
lbp_finish	O	1	Indication signal of the CLBP circuit is finished

Signal	I/O	Bit-width	Description
id_addr	O	8	Address signal connected to ID_RAM
id_ren	O	1	Read enable signal to ID_RAM
id_rdata	I	5	Read data signal from ID_RAM
id_wen	O	1	Write enable signal to ID_RAM
id_wdata	O	5	Write data signal to ID_RAM
hist_addr_train	O	21	Address signal connected to HIST_RAM_TRAIN
hist_wen_train	O	1	Write enable signal to HIST_RAM_TRAIN
hist_wdata_train	O	8	Write data signal to HIST_RAM_TRAIN
hist_ren_train	O	8	Read enable signal to HIST_RAM_TRAIN
hist_rdata_train	I	8	Read data signal from HIST_RAM_TRAIN
hist_addr_predict	O	21	Address signal connected to HIST_RAM_PREDICT
hist_wen_predict	O	1	Write enable signal to HIST_RAM_PREDICT
hist_wdata_predict	O	8	Write data signal to HIST_RAM_PREDICT
hist_ren_predict	O	8	Read enable signal to HIST_RAM_PREDICT
hist_rdata_predict	I	8	Read data signal from HIST_RAM_PREDICT

- **Port list of each module:**
- CLBP

Signal	I/O	Bit-width	Description
clk	I	1	Clock signal
rst	I	1	Reset signal
enable	I	1	CLBP circuit enabling signal
gray_addr	O	12	Address signal connected to RAM_GRAY
gray_OE	O	1	Read enable signal to RAM_GRAY
gray_data	I	8	Read data signal from RAM_GRAY
lbp_addr	O	12	Address signal connected to RAM_LBP MUX to memory
lbp_WEN	O	1	Write enable signal to RAM_LBP
lbp_data	O	8	Write data signal to RAM_LBP
theta	O	25(fixed-point)	Current neighbor's theta signal(unit is in radian)
theta_valid	O	1	Indication signal of current neighbor's thetas is valid
cos_data	I	25(fixed-point)	Cosine value of the theta (from testbench)
cos_valid	I	1	Indication signal of cosine value is valid
sin_data	I	25(fixed-point)	Sine value of the theta(from testbench)
sin_valid	I	1	Indication signal of sine value is valid
finish	O	1	Indication signal of the LBP circuit is finished

➢ HCU

Signal	I/O	Bit-width	Description
clk	I	1	Clock signal
rst	I	1	Reset signal
mode	I	1	Indication signal of whether the system is in prediction or training phase, 0 is training, 1 is prediction
enable	I	1	HCU circuit enabling signal
gridX	I	4	Image sliced portion in X direction, value is 8
gridY	I	4	Image sliced portion in Y direction, value is 8
lbp_addr	O	12	Address signal connected to RAM_LBP MUX to memory
lbp_ren	O	1	Read enable signal to RAM_LBP
lbp_rdata	I	8	Read data signal from RAM_LBP

Signal	I/O	Bit-width	Description
hist_addr_train	O	21	Address signal connected to HIST_RAM_TRAIN MUX to memory
hist_wen_train	O	1	Write enable signal to HIST_RAM_TRAIN
hist_wdata_train	O	8	Write data signal to HIST_RAM_TRAIN
hist_ren_train	O	8	Read enable signal to HIST_RAM_TRAIN MUX to memory
hist_rdata_train	I	8	Read data signal from HIST_RAM_TRAIN
hist_addr_predict	O	21	Address signal connected to HIST_RAM_PREDICT MUX to memory
hist_wen_predict	O	1	Write enable signal to HIST_RAM_PREDICT
hist_wdata_predict	O	8	Write data signal to HIST_RAM_PREDICT
hist_ren_predict	O	8	Read enable signal to HIST_RAM_PREDICT MUX to memory
hist_rdata_predict	I	8	Read data signal from HIST_RAM_PREDICT
done	O	1	Indication that HCU circuit is done(should be asserted every time a subject picture is finished computing histogram)

➢ Comparator

Signal	I/O	Bit-width	Description
clk	I	1	Clock signal
rst	I	1	Reset signal
enable	I	1	Comparator circuit enabling signal
histcount	I	8	# IDs encountered during training mode
distance	I	1	DCU computed distance value
dcu_valid	I	1	Indication that the current distance value is valid
id	I	5	Id read data from ID_RAM
id_ren	O	1	Read enable signal to ID_RAM
id_counter	O	8	The current ID address it is processing MUX to memory
dcu_enable	O	1	DCU circuit enabling signal
label	O	5	Prediction result, output ID value
minDistance	O	18	Minimum distance between the prediction histogram and the closest histogram in HIST_TRAIN_RAM
hist_addr_offset	O	21	The address offset in HIST_RAM_TRAIN of the id it is processing currently
done	O	1	Indication signal of the Comparator circuit is finished

➢ Controller

Signal	I/O	Bit-width	Description
clk	I	1	Clock signal
rst	I	1	Reset signal
mode	I	1	Indication signal of whether the system is in prediction or training phase, 0 is training, 1 is prediction
enable	I	1	Comparator circuit enabling signal
valid	I	1	Indication that current subject ID is valid
id	I	5	Subject ID
id_addr	O	8	Address signal connected to ID_RAM MUX to memory
id_wen	O	1	Write enable signal to ID_RAM
id_wdata	O	5	Write data signal to ID_RAM
lbp_enable	O	1	CLBP circuit enabling signal
lbp_finish	I	1	Indication of the CLBP circuit is finished
ram_clbp	O	1	Indication that the CLBP circuit has the access to RAM_LBP

Signal	I/O	Bit-width	Description
gridX_i	I	4	Image sliced portion in X direction, value is 8, from testbench
gridY_i	I	4	Image sliced portion in Y direction, value is 8, from testbench
hcu_enable	O	1	HCU circuit enabling signal
gridX_o	O	4	Image sliced portion in X direction, value is 8, to HCU
gridY_o	O	4	Image sliced portion in Y direction, value is 8, to HCU
hcu_finish	I	1	Indication of the HCU circuit is finished
comparator_finish	I	1	Indication of the Comparator circuit is finished
comparator_enable	O	1	Comparator circuit enabling signal
ram_comp	O	1	Indication that the Comparator circuit & DCU circuit has the access to ID_RAM, HIST_RAM_TRAIN, HIST_RAM_PREDICT

- Understanding the function:

  Once system is initialized, it

  a) Receives *gridX* and *gridY* signal.
  b) Receive *valid* and *id* signal, then compute local binary pattern value and store the result into RAM_LBP.
  c) In training mode, computes histogram information from RAM_LBP and store it to HIST_RAM_TRAIN.
  d) Repeat step(b)~(c) until encounter prediction mode.
  e) Prediction mode is detected, goes to step(b).
  f) In prediction mode, computes histogram information from RAM_LBP and store it to HIST_RAM_PREDICT.
  g) Comparator starts to work, control DCU to compute D, where D is defined as:

  $$D = \sum_{p=1}^{n} (hist\_predict_p - hist\_train_p)^2$$

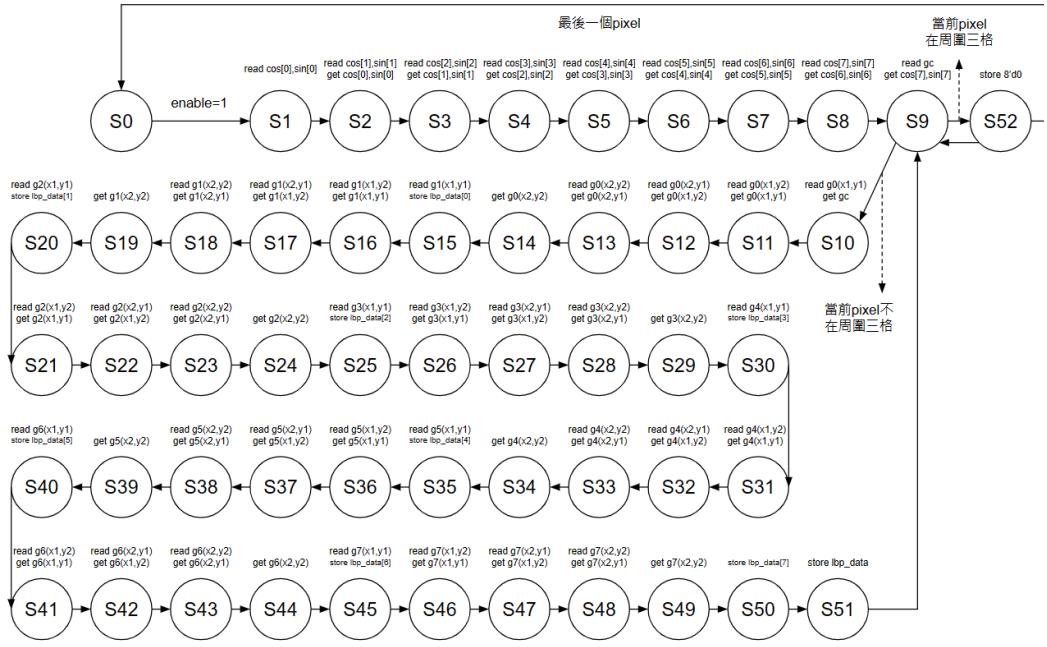
  , where n is 16384(8x8x256).
  h) Loop step(g) for 7xN times, where N is the different subject count, and find the closest histogram in HIST_RAM_PREDICT w.r.t the prediction histogram computed in step(f), see p.18 in handout.
  i) Output *label* & *minDistance* & *done* signal.
  j) Repeat step(e)~(i) until testbench stops the simulation.

- Describe your design in detail. You can draw internal architecture or block diagram to describe your dsign. If your submodule contains any FSM, you should also depict it and elaborate as well.

  Note: **if you design your own internal architecture** other than using the provided one, please feel free to **alter the block** below, and add your own design as well as decribe them in detail.

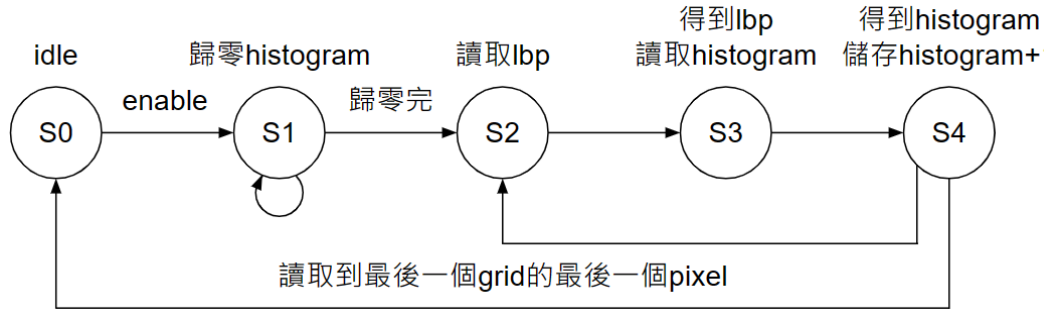
  - CLBP

我設計的 CLBP 一開始 reset 後會先在 idle 的 state，當 enable 為 1 時就會開始進行運算，先從 testbench 中一個一個 state 拿取每個角度的 sin 值和 cos 值，並且接著計算出每個角度的 rx、ry、x1、x2、y1、y2、tx、ty、w1、w2、w3、w4，我使用一個 counter 來記錄現在所在的 pixel，如果 counter 現在所在的位置是在整張圖片的周圍三格，就直接跳到將 8'd0 存入 RAM_LBP 的 state；如果 counter 現在所在的位置在整張圖片的中心(不是在周圍三格)，就先利用 FSM 從 RAM_GRAY 中一個一個 state 拿取所在的 pixel 每個角度上的四個資料，接著在每五個 state 後用內差法計算出 LBP 的數值，並且用 LBP>gc 來判斷 lbp_data 的每個 bit 為 1 或 0 後跳至下個 state，最後一個 state 再將 lbp_data 存入 RAM_LBP 中，當跑完最後一個 pixel 後就回到 idle 的 state。



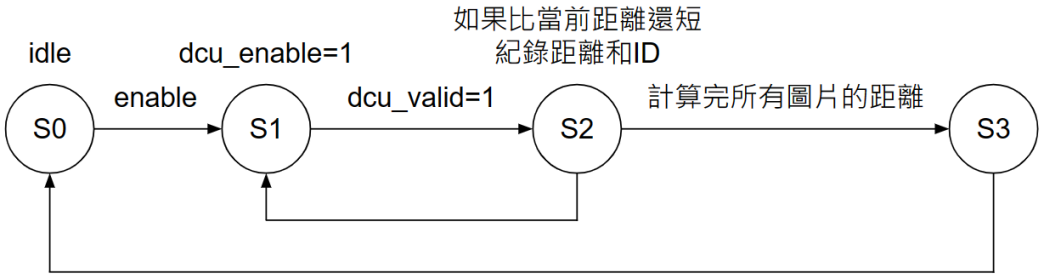
■ HCU

我設計的 HCU 一開始 reset 後會先在 idle 的 state，當 enable 為 1 時就會開始進行運算，一開始先進行將當前照片的 histogram 進行歸零，當歸零完一個 histogram 後就開始進行 histogram 的計算，先進行讀取 lbp 的資料後，再讀取相對應 histogram 的值，將該值加 1 後再存回去相同地址，當跑完最後一個 grid 的最後一個 pixel 後就回到 idle 的 state。



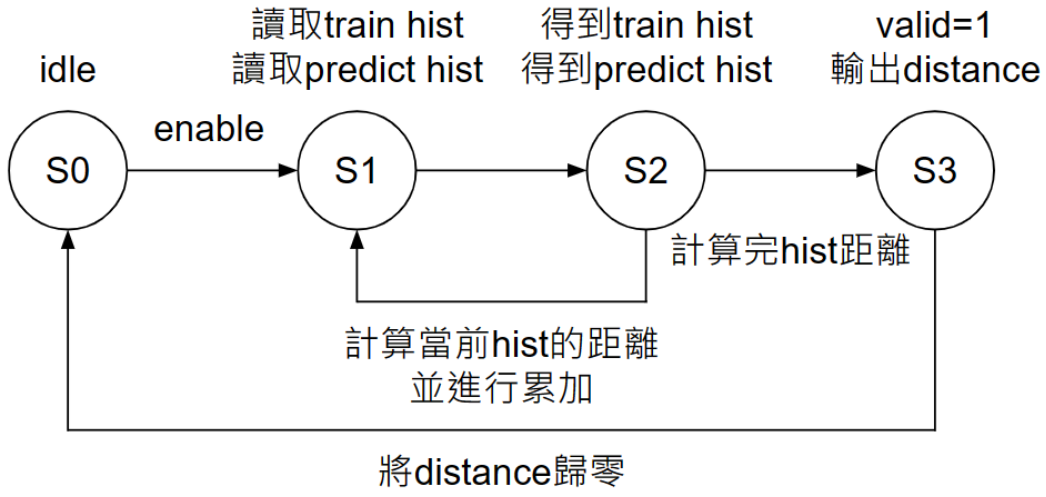
■ Comparator

我設計的 Comparator 一開始 reset 後會先在 idle 的 state，當 enable 為 1 時就會開始進行運算，一開始先讓 DCU 計算圖片與第一張圖片的距離，第一張圖片先儲存進 minDistance 和 label，接著讓 DCU 計算圖片與第二張圖片的距離，如果距離比儲存的 minDistance 還短就儲存新的 minDistance 和 label，接著繼續跑下一張圖，當跑完最後一個圖片後就回到 idle 的 state。



■ DCU

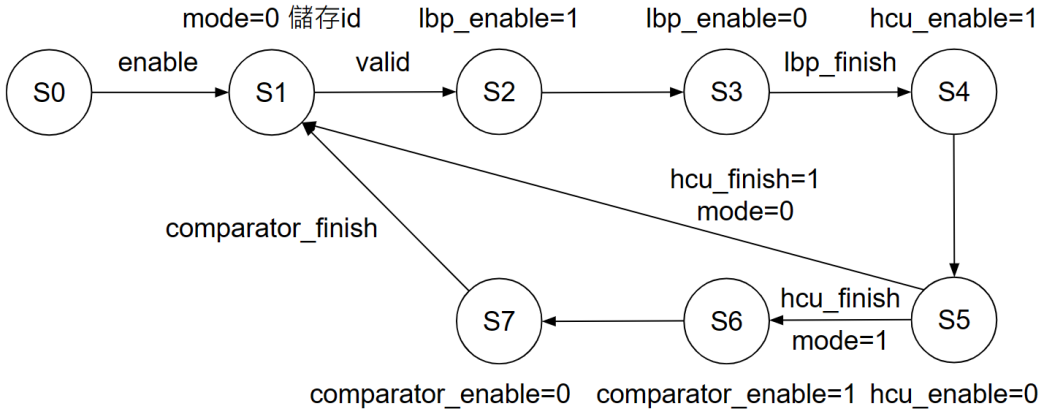
我設計的 DCU 一開始 reset 後會先在 idle 的 state，當 enable 為 1 時就會開始進行運算，一開始先讀取當前的 train histogram 和 predict histogram，接著再得到 train histogram 和 predict histogram 後計算之間的距離並進行累加，當跑完最後一個 histogram 後就把 valid 變 1 輸出 distance，並且回到 idle 的 state 把 distance 歸零。



■ Controller

◆ Draw your state diagram in controller and explain it

我設計的 Controller 一開始 reset 後會先在 idle 的 state，當 enable 為 1 時就會開始進行運算，如果是 train 模式一開始先進行 CLBP 的運算，等 CLBP 運算完成後進行 HCU 的運算，等 HCU 運算完成後就回到 S1；如果是 predict 模式一開始先進行 CLBP 的運算，等 CLBP 運算完成後進行 HCU 的運算，等 HCU 運算完成後進行 Comparator 的運算，等 Comparator 運算完成後就回到 S1。



- Your own internal architecture
  - Draw and explain if you design your own architecture, if don't, you can skip this section.

我設計的 MUX_ID、MUX_LBP、MUX_PREDICT、MUX_TRAIN 都是利用 select 的訊號來控制 RAM 的輸出和輸入，MUX_LBP 是用 ram_clbp 這個訊號來控制輸出和輸入，ram_clbp 為 0 時代表在進行 CLBP 的運算，ram_clbp 為 1 時代表在進行 HCU 的運算；MUX_ID、MUX_PREDICT、MUX_TRAIN 是用 ram_comp 這個訊號來控制輸出和輸入，ram_comp 為 0 時代表在進行 train 的運算，ram_comp 為 1 時代表在進行 predict 的運算。

1) Complete the Controller, HCU, CLBP, DCU, Comparator, and TOP module, in the system. **If you design your own architecture, please add the submodule list here!**

Submodule list:

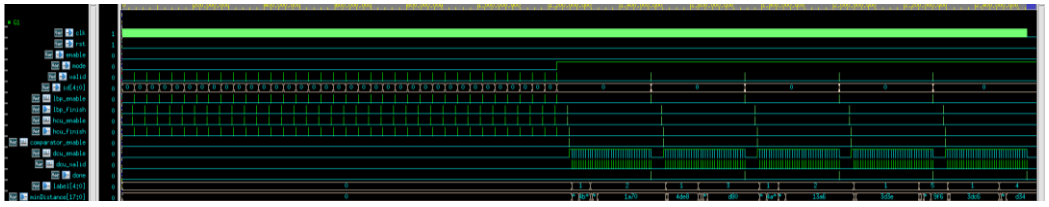
1. …

2) Compile the verilog code to verify the operations of this module works properly.

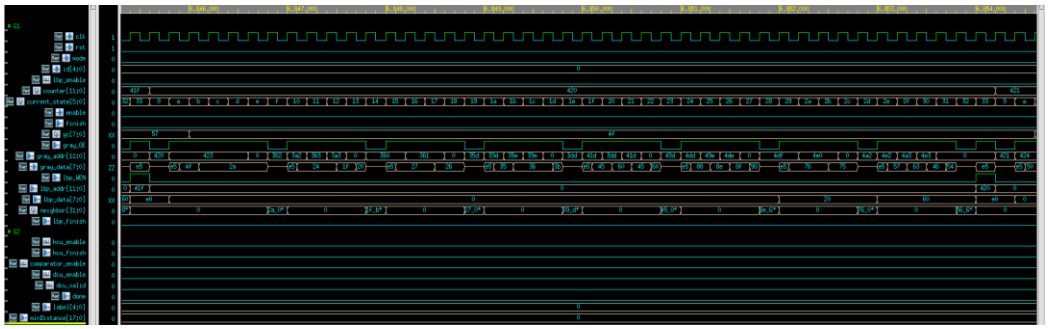
3) Synthesize your *top.v* with following the constraints:

- Clock period: no more than 2.0 ns.
- Don't touch network: clk.
- Wire load model: Wire load model: N16ADFP_StdCellss0p72vm40c.
- Synthesized verilog file: *top_syn.v*.
- Timing constraint file: *top_syn.sdf*.

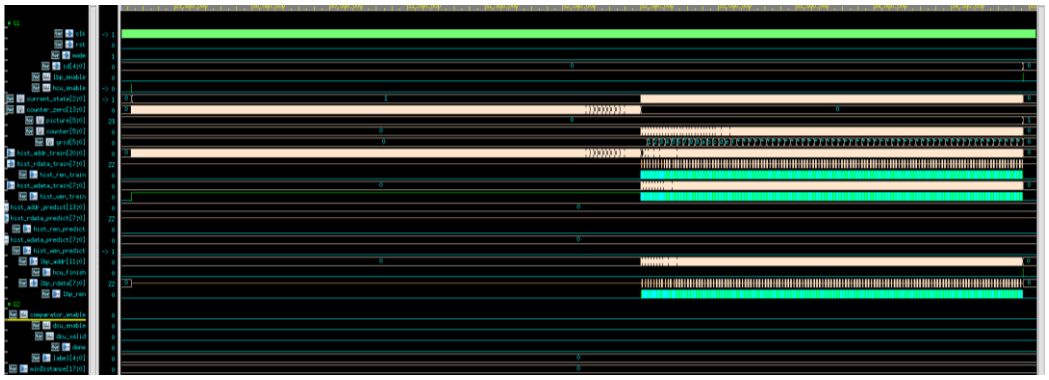
4) Please **attach your waveforms** and **specify your operations** on the waveforms. The more you elaborate, the higher the score is.



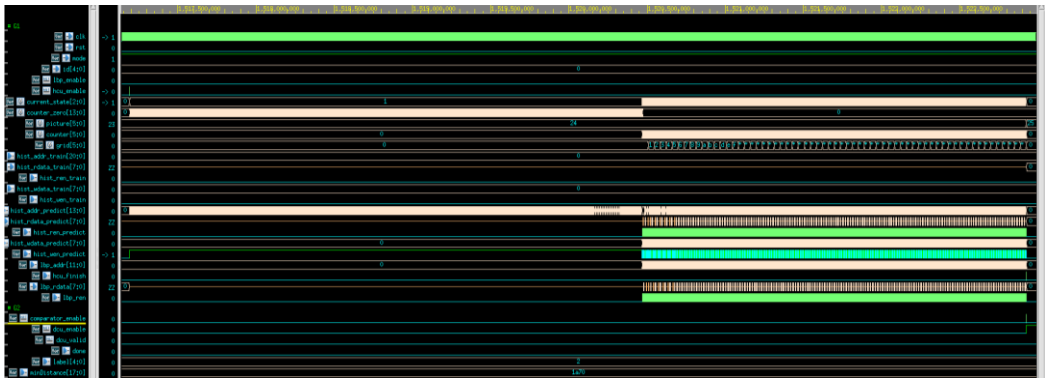
從上方波形中可以看到，這是整體運行的波形圖，可以看到前半段是進行 train 後半段是進行 predict。



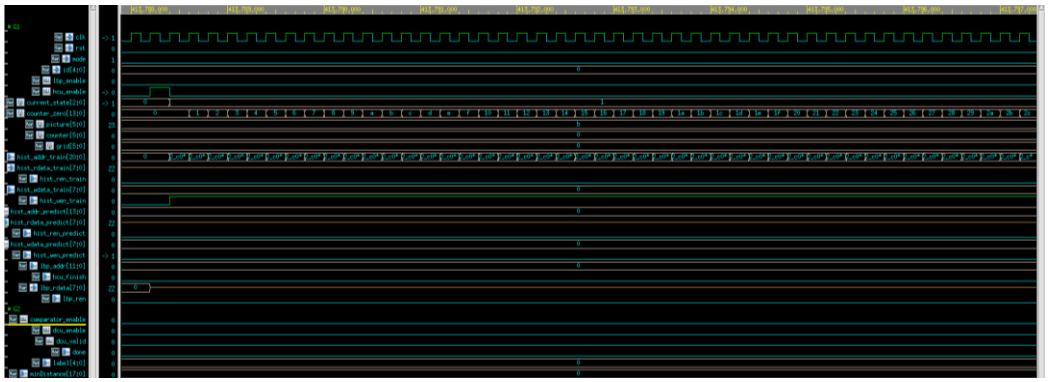
從上方波形中可以看到，這是 CLBP 在每個 state 從 RAM_GRAY 中拿取 x1y1, x1y2, x2y1, x2y2 的資料後就馬上進行運算 lbp_data 後跳到下一個 k，並且在最後一個 state 將運算完的資料儲存回 RAM_LBP 中，接著再回到開始運算的 state。



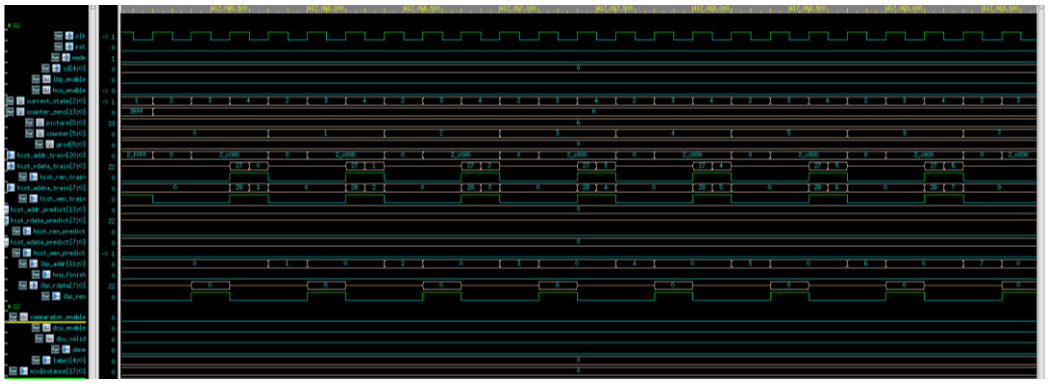
從上方波形中可以看到，這是 HCU 在 train mode 的整體運行的波形圖，前半段是將所有的 histogram 先進行歸零，後半段是進行 histogram 的計算。



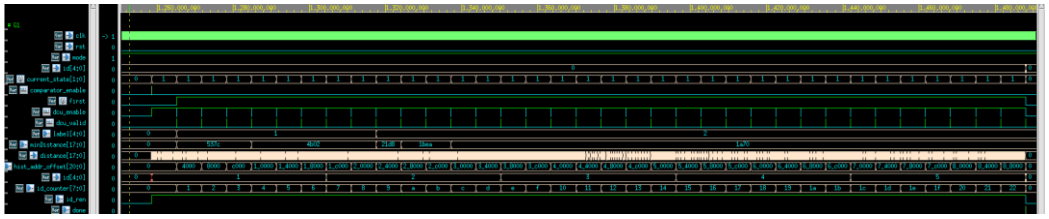
從上方波形中可以看到，這是 HCU 在 predict mode 的整體運行的波形圖，前半段是將所有的 histogram 先進行歸零，後半段是進行 histogram 的計算。



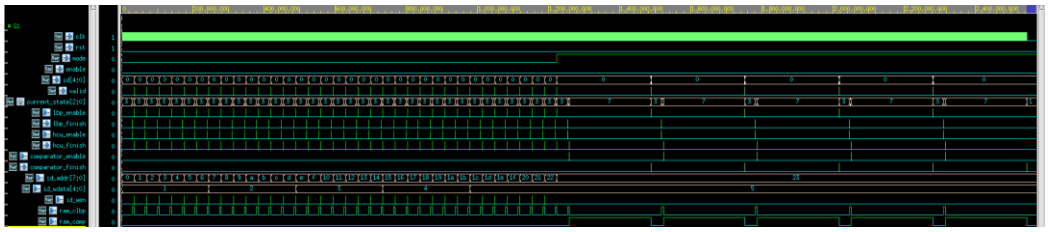
從上方波形中可以看到，這是 HCU 在 train mode 進行歸零時的波形圖，將所有的 histogram 先進行歸零，這在 predict mode 也是相同的，只是 RAM 的輸入輸出從 train 改成了 predict。



從上方波形中可以看到，這是 HCU 在 train mode 進行 histogram 的計算時的波形圖，將所有的 histogram 進行歸零後，先進行讀取 lbp 的資料後，再讀取相對應 histogram 的值，將該值加 1 後再存回去相同地址，這在 predict mode 也是相同的，只是 RAM 的輸入輸出從 train 改成了 predict。

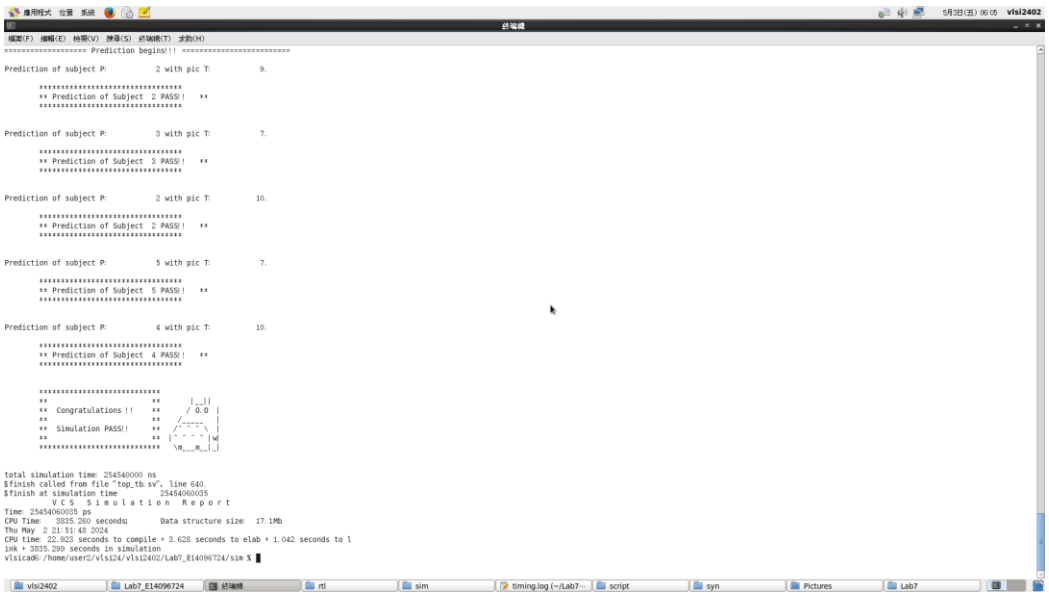


從上方波形中可以看到，這是 Comparator 和 DCU 在 predict mode 的整體運行的波形圖，一開始先讓 dcu_enable 變 1 計算圖片與第一張圖片的距離，等 dcu_valid 變 1 後第一張圖片先儲存進 minDistance 和 label，接著讓 dcu_enable 變 1 計算圖片與第二張圖片的距離，等 dcu_valid 變 1 後如果距離比儲存的 minDistance 還短就儲存新的 minDistance 和 label，接著繼續跑下一張圖，直到跑到最後一張圖片。



從上方波形中可以看到，這是 Controller 整體運行的波形圖，如果是 train 模式一開始先讓 lbp_enable 變 1，等 lbp_finish 變 1 後讓 hcu_enable 變 1，等 hcu_finish 變 1 後就回到 S1 等待新的 id 和 valid；如果是 predict 模式一開始先讓 lbp_enable 變 1，等 lbp_finish 變 1 後讓 hcu_enable 變 1，等 hcu_finish 變 1 後讓 comparator_enable 變 1，等 comparator_finish 變 1 後就回到 S1 等待新的 id 和 valid。

5) Show simulation result

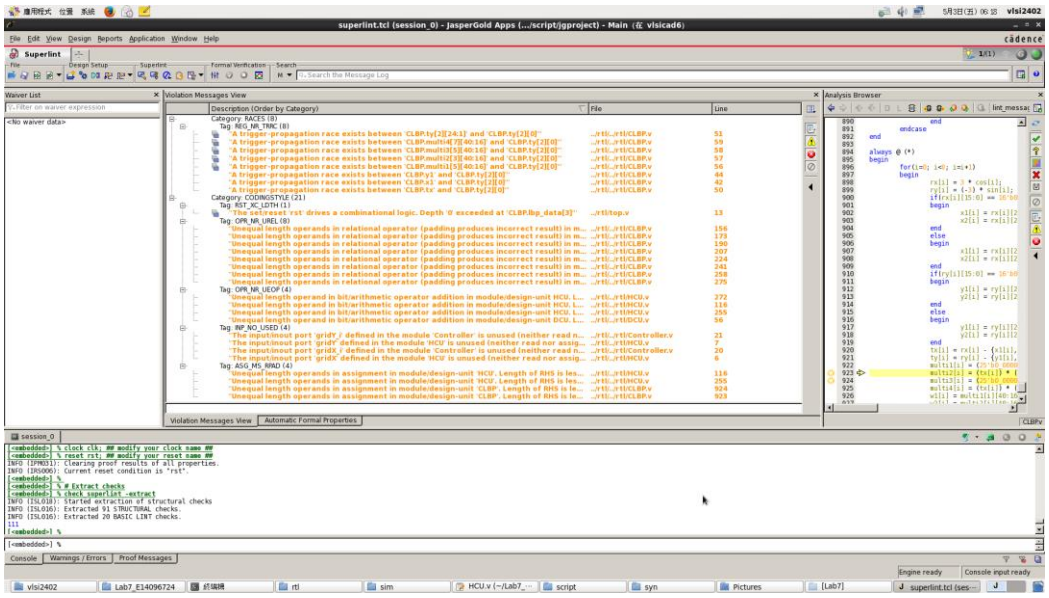


```
***   $sdf_annotate() version 1.2R
***    SDF file: "../syn/top_syn.sdf"
***    Annotation scope:  top_tb.top
***    No MTM selection argument specified
***    No SCALE FACTORS argument specified
***    No SCALE TYPE argument specified
***    MTM selection defaulted to "TOOL_CONTROL":
            (+typdelays compiled, TYPICAL delays selected)
***    SCALE FACTORS defaulted to "1.0:1.0:1.0":
***    SCALE TYPE defaulted to: "FROM_MTM"
***    Turnoff delay: "FROM_FILE"
***    Approximation (mipd) policy: "MAXIMUM"

***    SDF annotation begin:  Thu May  2 22:25:02 2024


SDF Info:  +pulse_r/100, +pulse_e/100 in effect
```

6) Show SuperLint coverage (top.v)



top.v : (1 - 1/270) * 100% = 99.63%

CLBP.v : (1 - 18/932) * 100% = 98.07%

HCU.v : (1 - 7/324) * 100% = 97.84%

DCU.v : (1 - 1/88) * 100% = 98.86%

Comparator.v : (1 - 0/103) * 100% = 100%

Controller.v : (1 - 2/180) * 100% = 98.89%

MUX_ID.v : (1 - 0/39) * 100% = 100%

MUX_LBP.v : (1 - 0/39) * 100% = 100%

MUX_PREDICT.v : (1 - 0/43) * 100% = 100%

MUX_TRAIN.v : (1 - 0/43) * 100% = 100%

total : (1 - 29/2061) * 100% = 98.59%

7) Your clock period, total cell area, post simulation time (top.v) in screenshot.

Clock period: 2.0ns

Total cell area: 18580.545117

Post simulation time: 25454060035(ps) = 25454060.035(ns)

```
set clk_period  2.0
```

```
Total cell area:                    18580.545117
```

```
Time:  25454060035 ps
```

8) Please describe how you optimize your design when you run into problems in synthesis .ex: plug in some registers between two instances to shorten your datapath, resource sharing for some registers to reduce your cell area.

這次的 Lab 在合成時我的 time slack 一開始是負的，我從 timing report 中看我的 critical path 稍微進行修改，雖然我的 time slack 還是負的，跑合成後的模擬卻可以通過，可能那個 critical path 不是那麼重要或是沒有用到，後來沒時間再進行修改只好使用這個版本。我把每個 module 中的每個 clock 所做的事盡量的減少，這樣就可以盡量讓 clock period 減少來，讓 simulation time 也跟這減少，我還有在 CLBP 中拿取資料後運算使用同一個 register 來讓 cell area 盡量減少。

- Lessons learned from this lab

這次的 Lab 讓我學到如何在一個 top module 中包含全部的小 module，利用 Controller 讓各個小 module 可以正常運行，並且要在每個 module 設置 FSM 才能讓整體的運行不會互相打擾，如何利用 enable 和 finish 來讓每個 module 來溝通，設計每個 module 的 FSM 是這次的 Lab 讓我學習到最多的東西。

- Suggestions for us (we appreciate your feedback)

我覺得做這種比較複雜的 Lab 應該可以給更多時間，這樣或許能夠完成的人數會變多，還有完成 Lab 的成果像是 PA 會更好，或是讓大家分組來進行這個 Lab 這樣會比較快速，而且多了小組討論不一定會讓學到的東西變少，這樣不會讓大家在期中考期間還要花費大量時間來做這個作業。

Please compress all the following files into one compressed file (".tar " format) and submit through Moodle website:

※ **NOTE**:

1. **If there are other files used in your design, please attach the files too and make sure they're properly included**.
2. Simulation commands

Lab7	Commands
superlint	% cd script % jg –superlint superlint.tcl
synthesis	% cd script % dv –f synthesis.tcl
Pre-sim	% cd sim % vcs -R -sverilog top_tb.sv -debug_access+all -full64
Post-sim	% cd sim % vcs -R -sverilog top_tb.sv -debug_access+all -full64 +define+SDF+SYN
Dump waveform	+define+FSDB

Don't use +define+FSDB when running post-sim, it'll occupy substantial amount of memory!