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

<table>
<tr><td>Name</td><td colspan="2">Student ID</td></tr>
<tr><td>簡笠恩</td><td colspan="2">E14102305</td></tr>
<tr><td>Practical</td><td>Points</td><td>Marks</td></tr>
<tr><td>Lab 7_1</td><td>30</td><td></td></tr>
<tr><td>Lab 7_2</td><td>65</td><td></td></tr>
<tr><td>Report</td><td>5</td><td></td></tr>
<tr><td>Demo</td><td>10</td><td></td></tr>
</table>

Notes		

Due: 15:00 May 1, 2024@ moodle

Summary

Hardware				
TOP		RTL(∨/X)		Synthesis(∨/X)
Lab7_1		V		V
Lab7_2		V		V
Synthesis result				
Clock period(ns)	Area		Simulation time (ns)	
2.0	38293.12		197933000	
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(%)
1709	141	0	91.75

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.

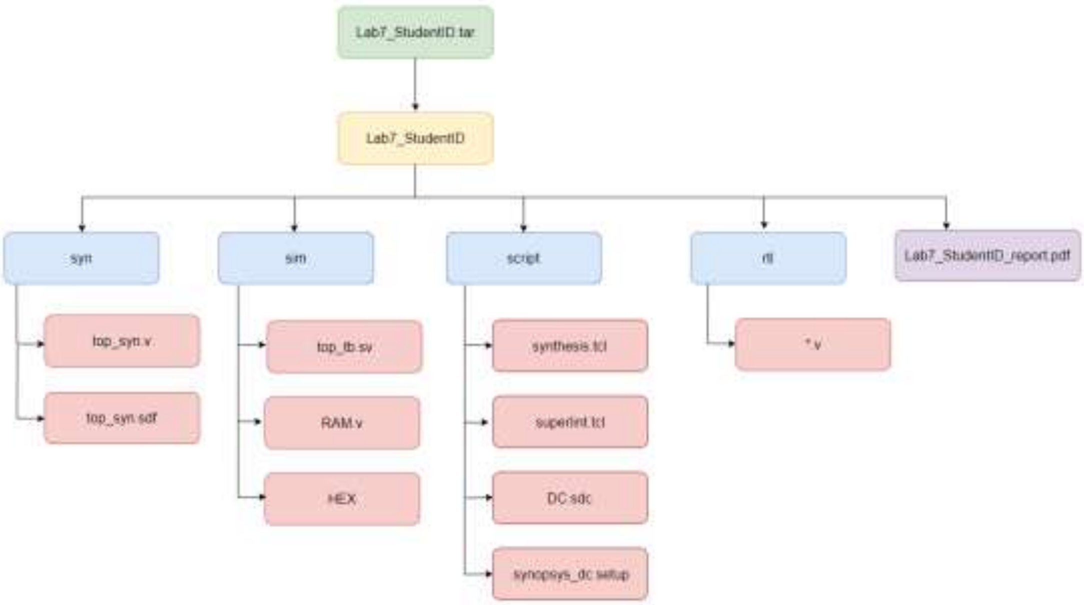
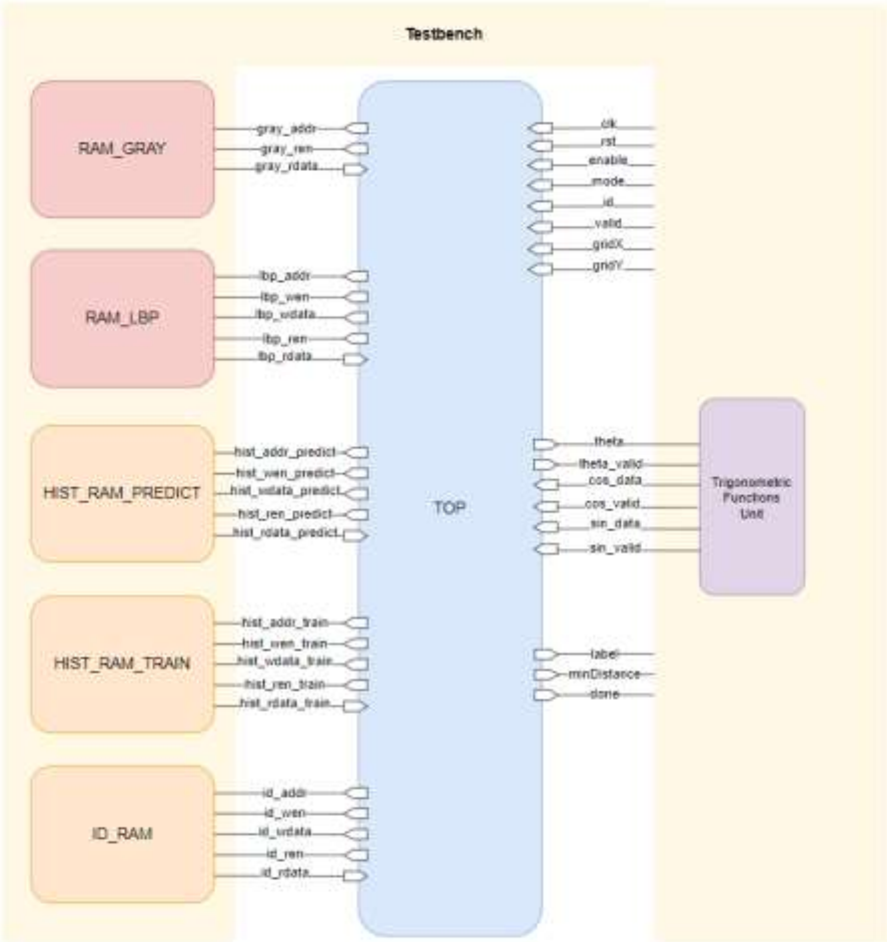


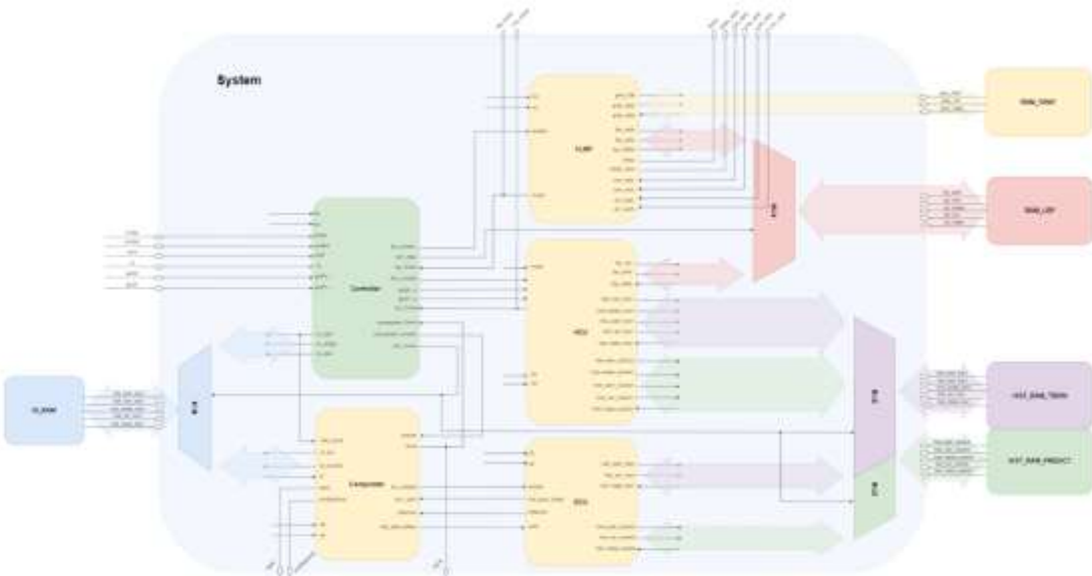
Fig.1 File hierarchy for Homework submission

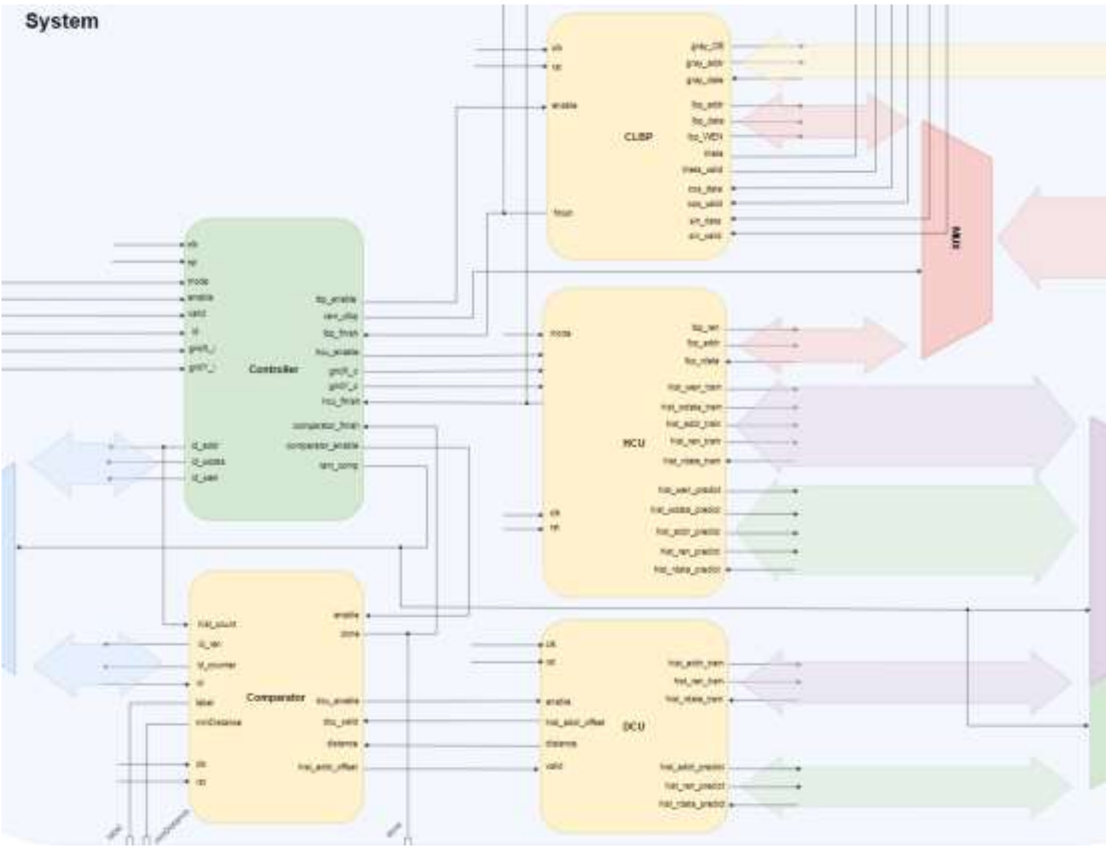
Lab 7

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.

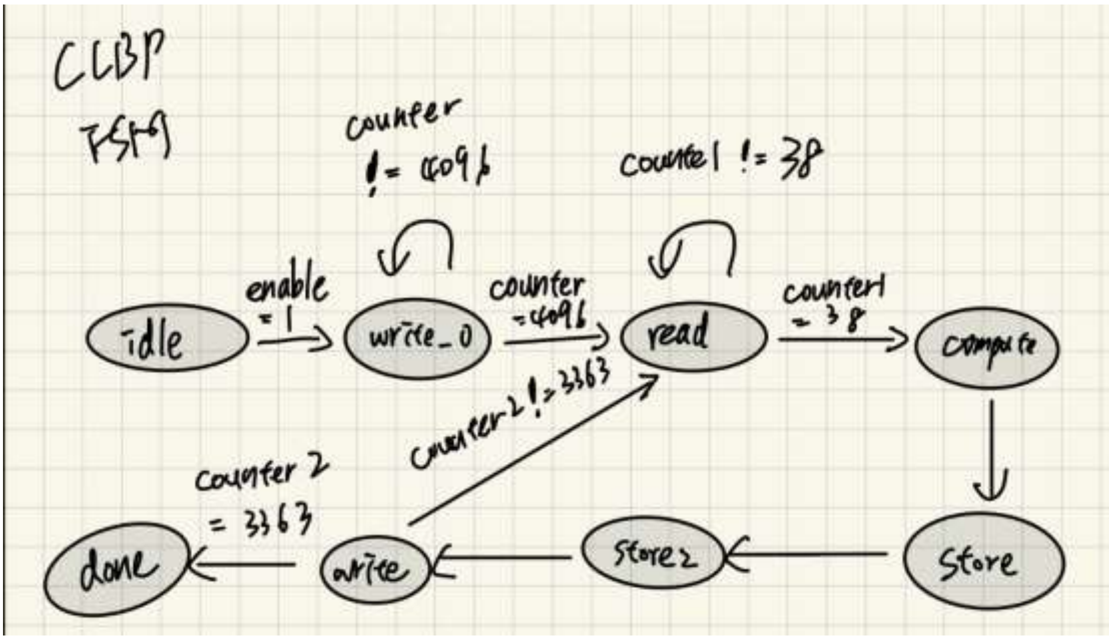
i) Output *label* & *minDistance* & *done* signal.

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



CLBP 分成 8 個狀態，分別為

1.idle: 電路致能前，當 enable 升起前，會一直維持在這。

2.write_0: 將 lbp_ram 設定初始值 0，直到寫滿 4096 個。

3.read: 讀取 gray_ram 的值，一次讀取 33 個(center，8*4 (for neighbor using))

4.compute: 計算出各個 neighbor 值。

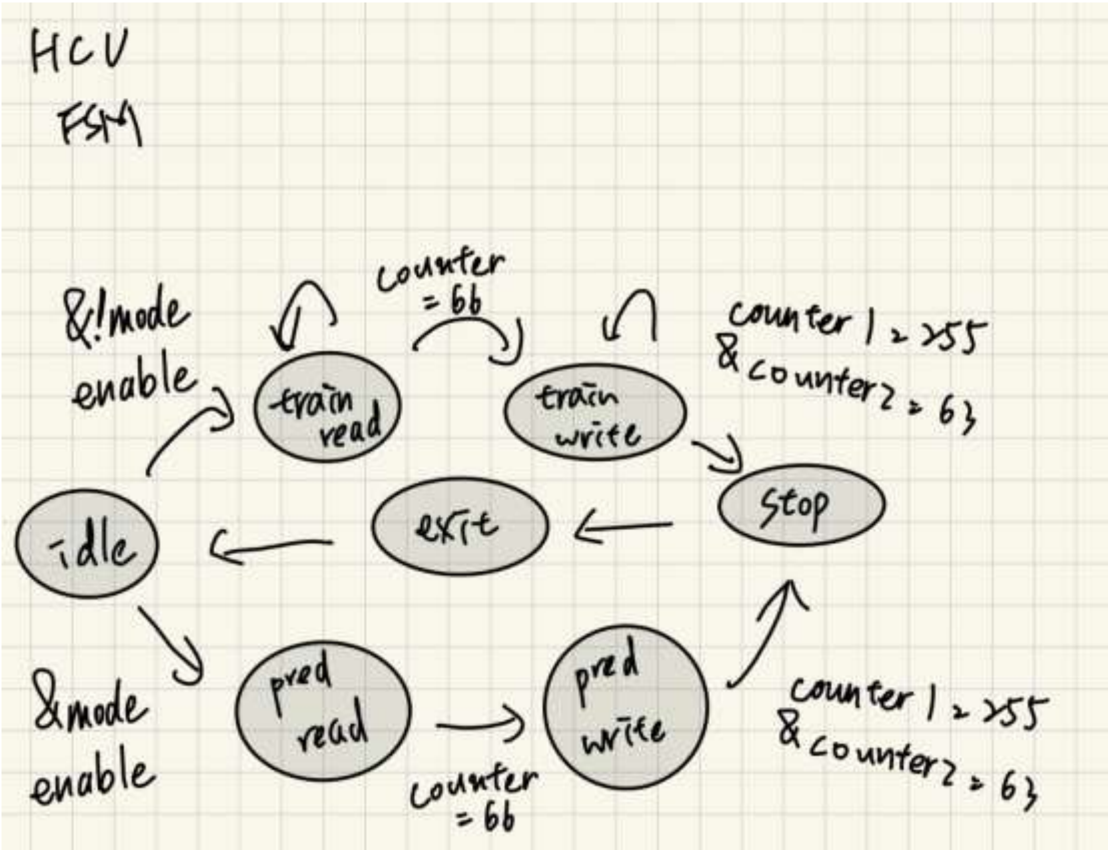
5.store: 比對 neighbor 與 center 值，並將 threshold function 設定的值存入。

6.store2: 將前一步驟的值相加 存入 lbp_data。

7.write: 將 lbp_data 寫入 lbp_ram。寫完跳入 done，沒寫完則跳回 read。

8.done: 寫入完成舉起 finish。並回去 idle 等下次 enable。

■ HCU



Hcu 有 7 個狀態，分別為

1. idle: 電路致能前。當 enable 升起後進 train_read。

2. train_read: 讀取 lbp_ram，並儲存每個 gird 的 histrogram 在讀取完 64 個後跳 train_write

3. train_write: 將存到的 data 寫入 hist_train_ram 中，寫完後跳回 train_read，直到 64grid 寫完跳 stop。

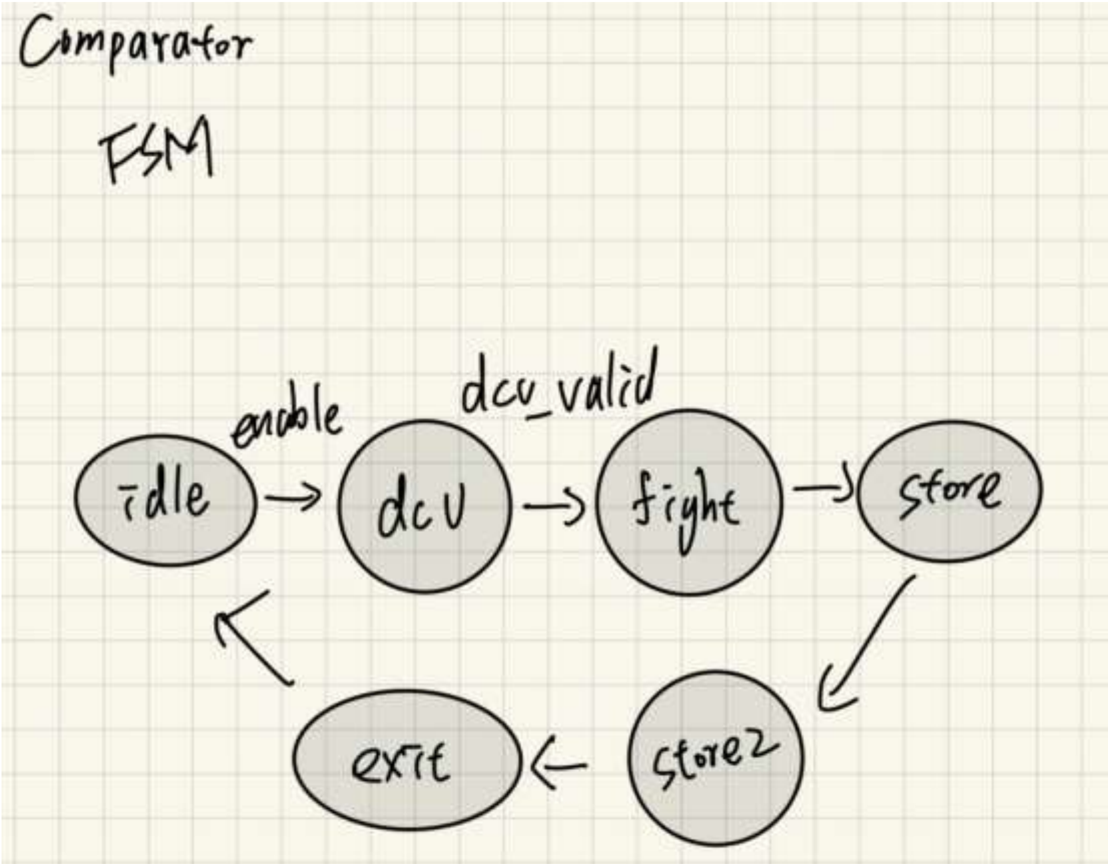
4. pred_read:同 train_read

5. pred_write:同 train_pred

6. stop : 為了寫入 hist_train_ram 最後一位的 delay。

7. exit : 升起 hcu_finish。並回去 idle 等下次 enable。

■ Comparator



Comparator 有 5 個狀態，分別為

1. idle: 電路致能前。等待 enable 升起。

2. dcu: 將 dcu_enable 拉起，並等待 Dcu 回傳。

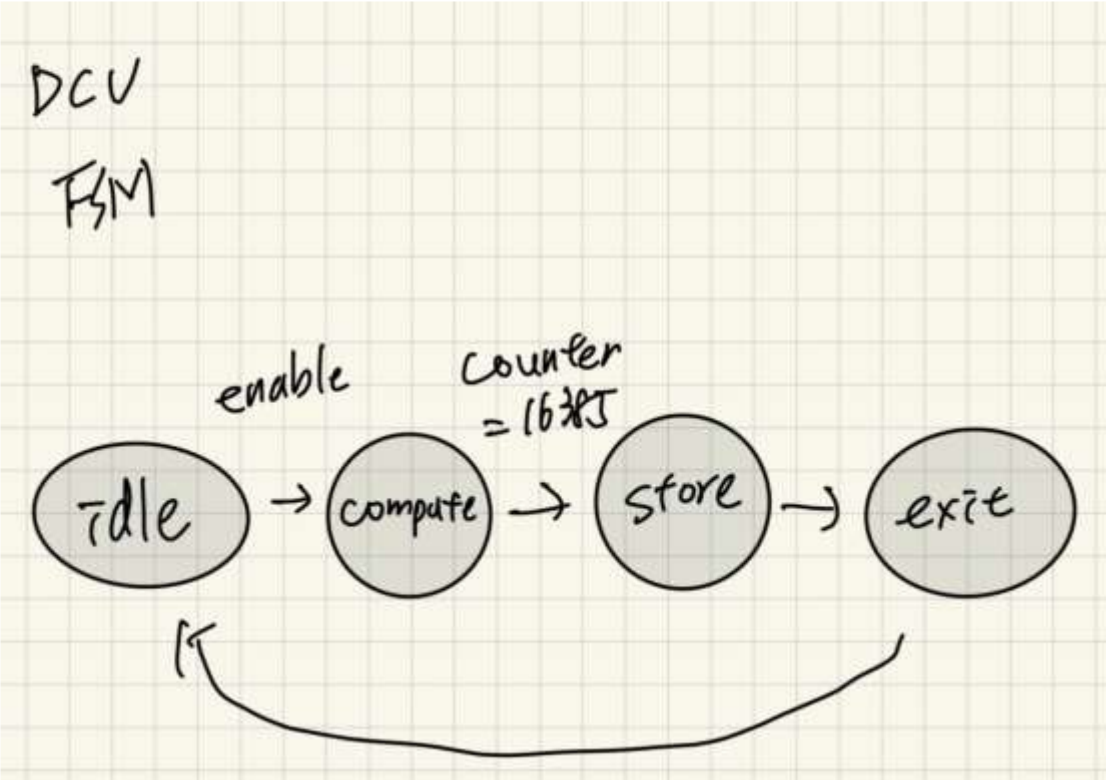
3. fight: 將 pred 與 train 資料庫所算出來的 distance 做對比，對比完後進下一段

4. store: 將 fight 結束後的最短路徑和 id 存入該到的地方。

5. store2: 為了合成後的 delay，讓 data 來得及傳入。

5. exit 將 done 升起，讓 tb 對答案，:並回去 idle 等下次 enable。

■ DCU



DCU: 主要有四狀態，分別為:

1. idle : 電路致能前。等待 enable 升起。

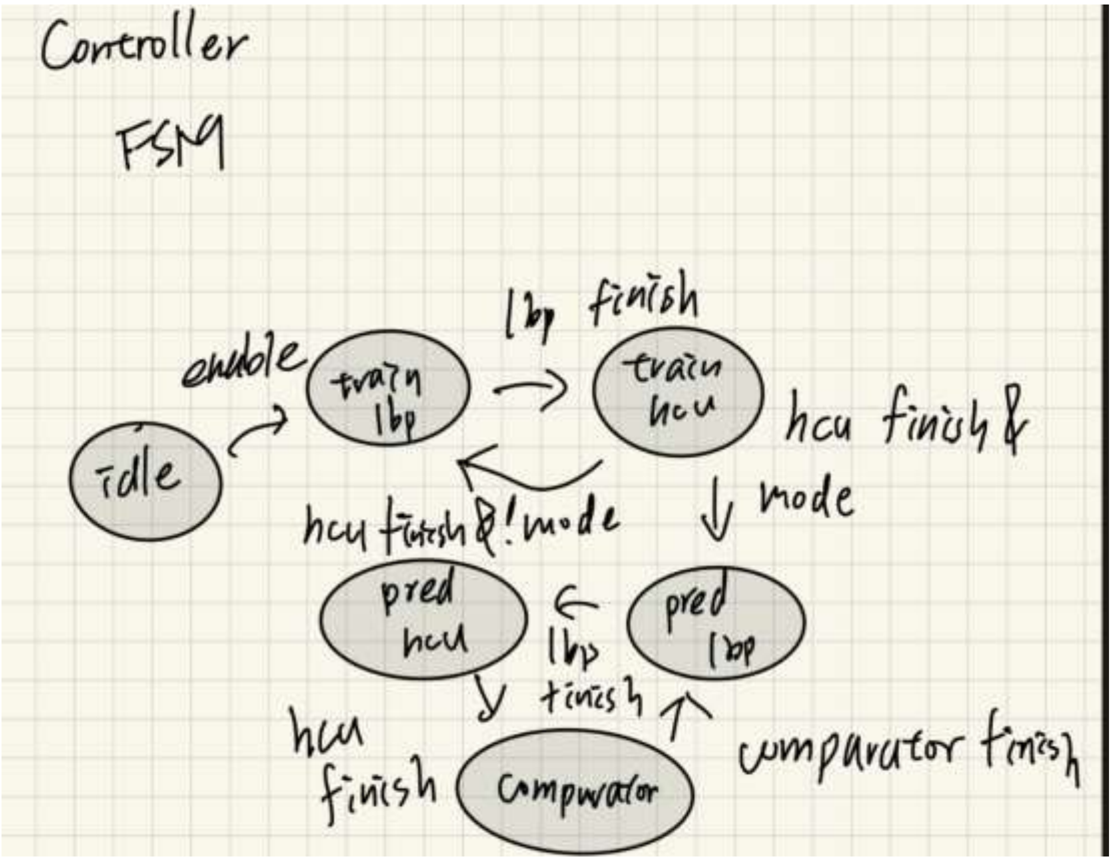
2. compute : 將 hist_predict_ram 的值與當前 id 的 hist_train_ram 算距離，當算完後，會跳入 store。

3. store : 將距離傳入 distance，並進入 exit。

4. exit: 升起 dcu_enable，並回到 idle。

■ Controller

◆ Draw your state diagram in controller and explain it



我的 controller 分為 6 個狀態，分別為

1. idle:電路致能前，enable 升起進入 train_lbp。

2. train_lbp:升起 lbp_enable，使 CLBP 電路運作

3. train_hcu:將 hcu_enable 升起，使 HCU 電路運作，再跳回 train_lbp。

4. pred_lbp:同 train_lbp。

5. pred_hcu:同 train_hcu。

6. comparator:升起 comparator_enable，進行 hist_train_ram 與

hist_predict_ram 的比較。

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

No!!!

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.

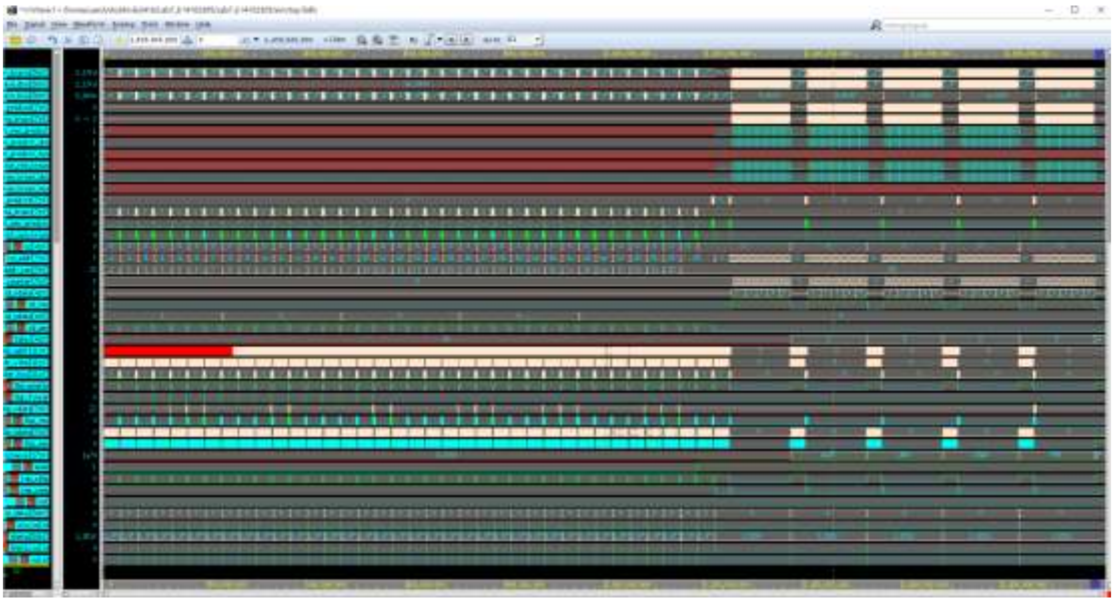
3) Synthesize your *top.v* with following the constraints:

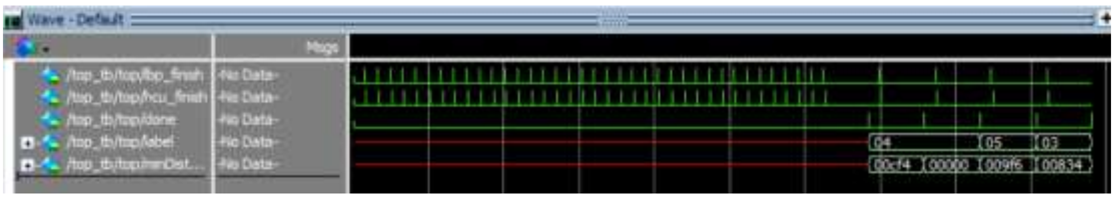
- Clock period: no more than 2.0 ns.
- Don’t touch network: clk.
- Wire load model: Wire load model: N16ADFP_StdCellss0p72vm40c.

- 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.

Top module

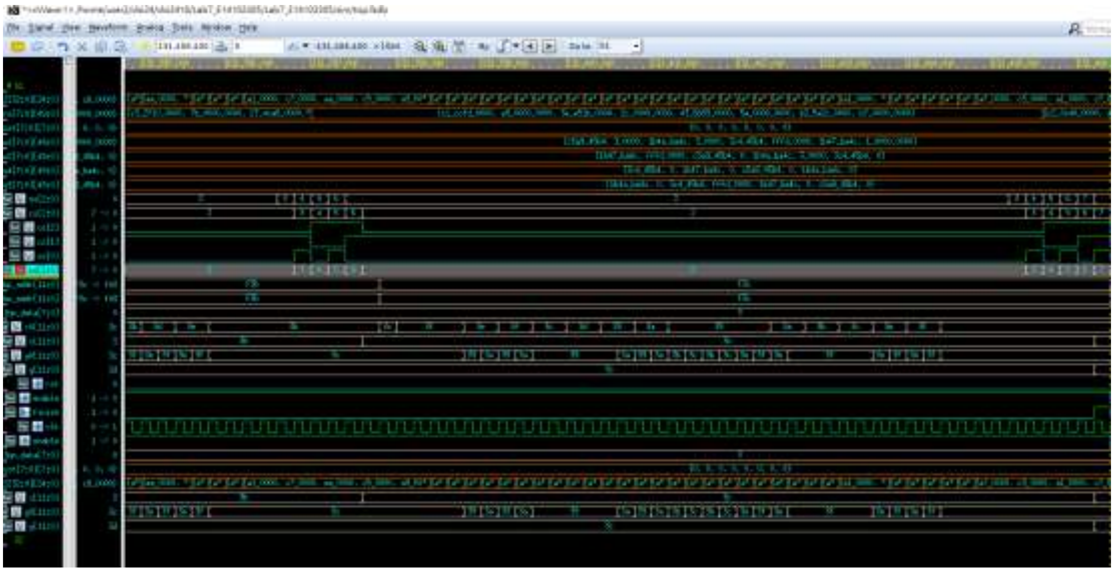




Top module 中可以看到 ram_clbp、ram_comp 是用來決定誰要在什麼時候對 RAM 進行操作的，而 distance、label 是我們最重要的東西，當 done 舉起時，會和 tb 對答案(label、minDistance)，以下面那張圖較清楚。

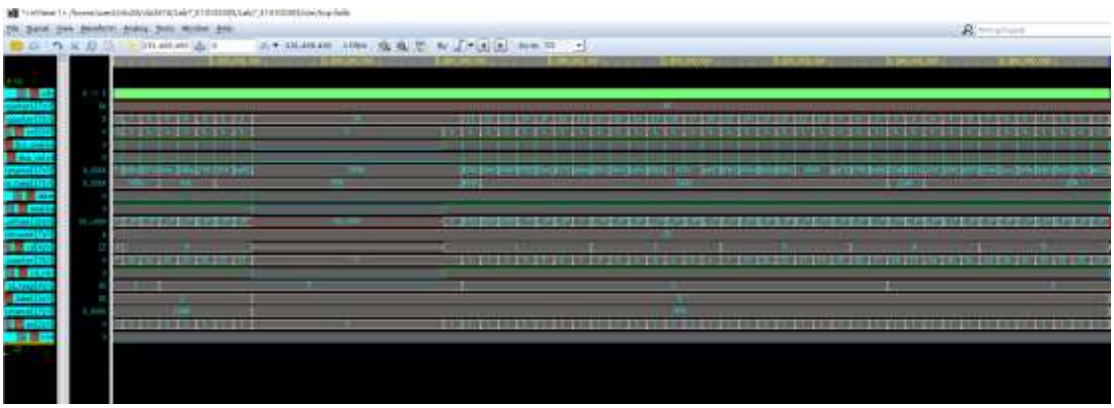
Sub module:

CLBP:



我們 cs =2 送入要讀的地址，然後再過 2 個 clk 後才讀取值，是為了迎合 RAM 的特性，因 為它會在 1.5 個 clk 吐出值，在讀完值，在 cs 為 s3 時進行運算 lbp_data，然後在 s4 進行寫入，所以下個狀態會進入 s5，並舉 起 finish 完成終止程式的動作

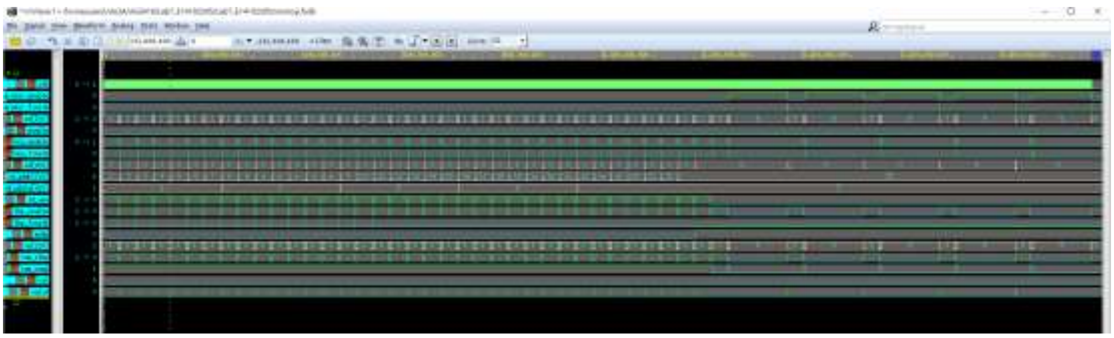
Controller



Controller 會在 1、2 的時候進行 train 的操控，3、4、5 則是 predict

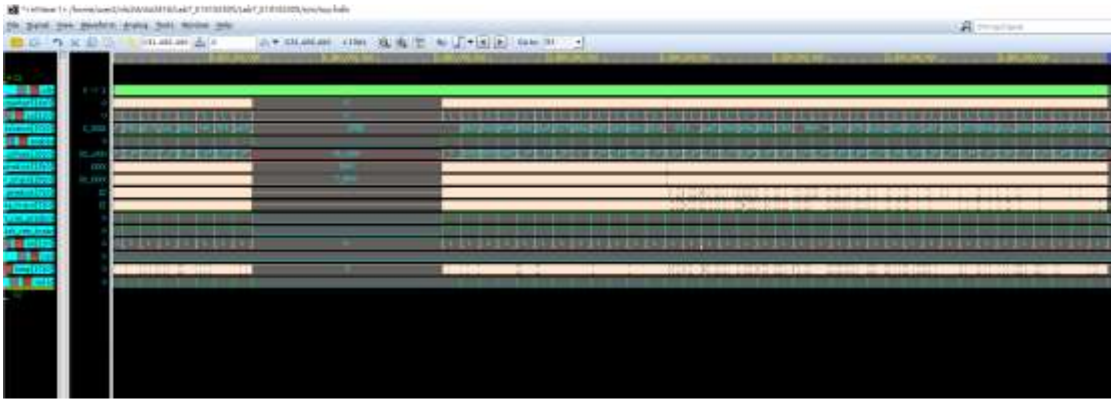
的操左，前兩個都是做 lbp 和 hcu 的致能與結束，5 則是操作 comparator。分別會以 lbp_finish，hcu_finish，comparator_finish 下去當狀態的跳轉條件。

Comparator:



Comparator 主要運作的地方在第二行那後五片，最長的為 cs = 5 時，是在進行 pred 與 train 資料庫的比對，而其他狀態則是為了，進行比對做準備。

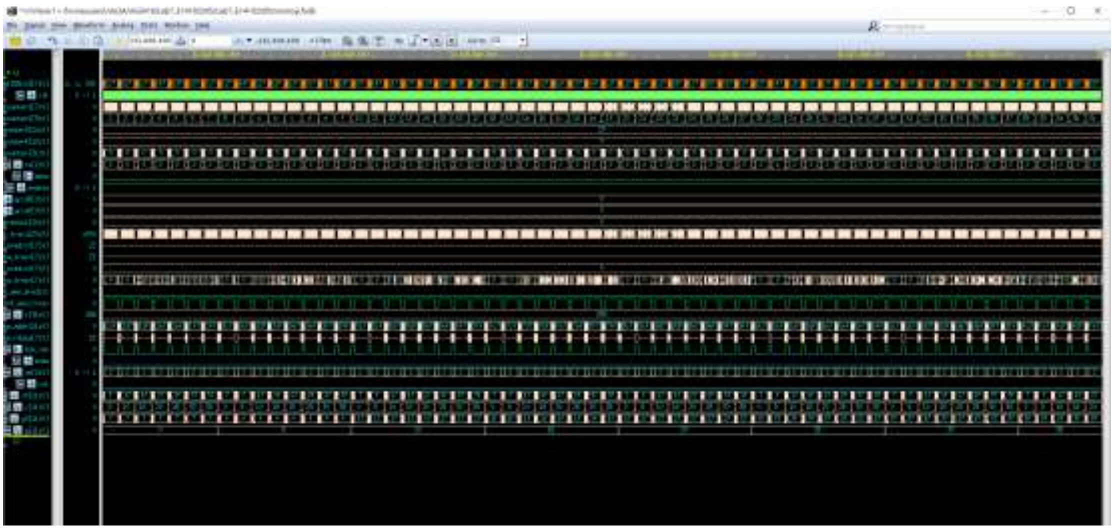
DCU:



DCU 後半片是在進行 pred_ram 與 train_ram 的每一片(16384)的每個值運算，

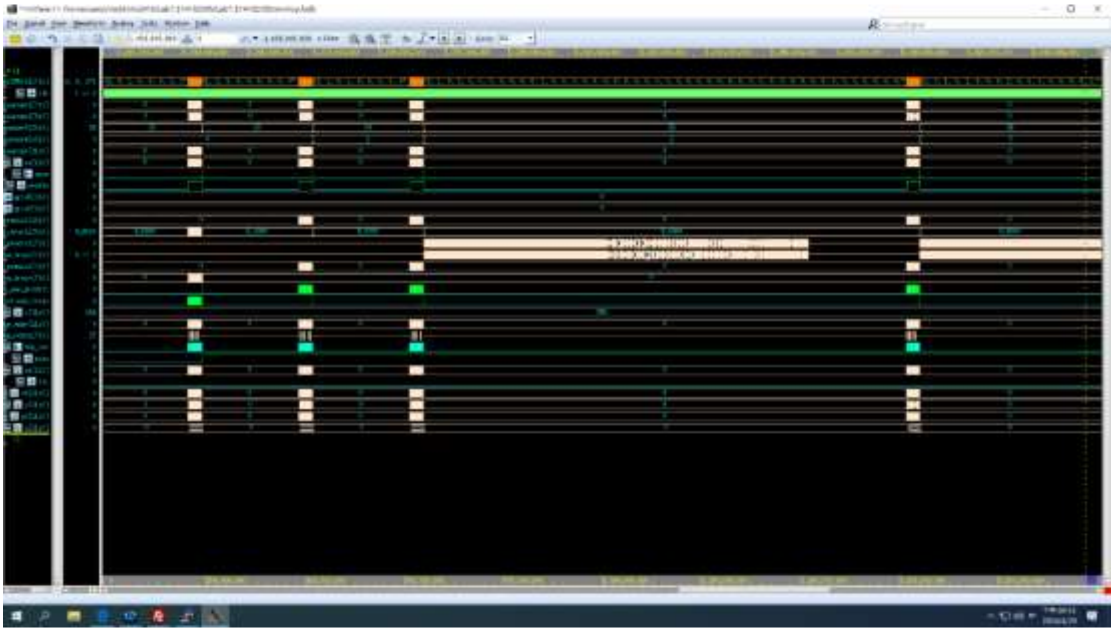
最下面的valid則是運算完成後，升起valid距離回傳到comparator，而做完一次運算在此 tb 裡是 5*7=35 個，最後會停一段時間是要再做下一張圖的 predict。

HCU_train:



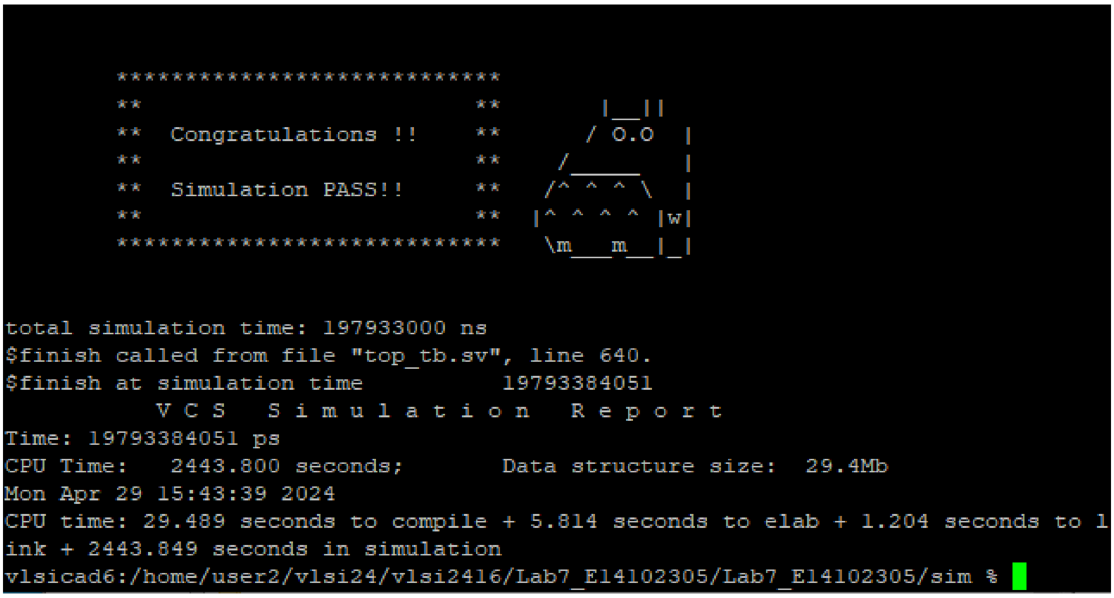
當 mode 為 train 時，先在 cs=1 時讀取 LBP_ram 裡的第一個 grid 之後我們會再 cs = 2 的狀態去進行寫入每個值的個數，並重複進行 64 次，之後便舉起 done 即為外面的 hcu_valid。

HCU_pred:

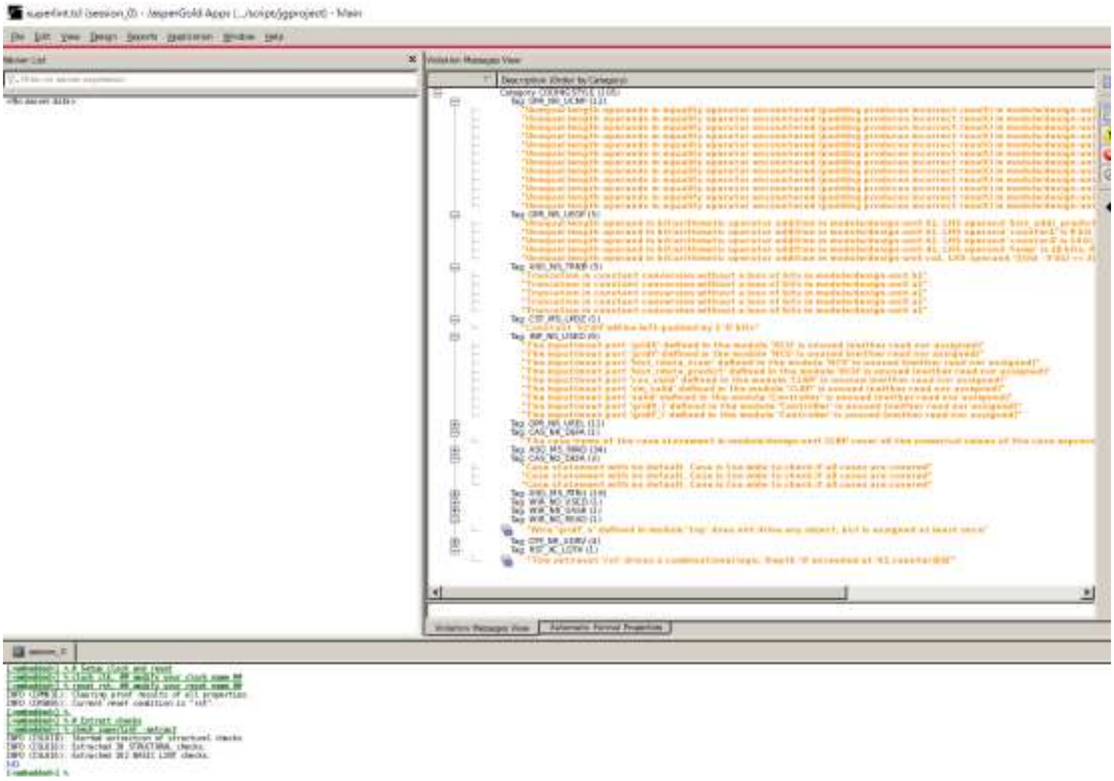


Predict 狀態的差別則是不用管 addr 超過 16383 的部分，因為每次重寫都會覆蓋之前的數值，所以較為簡易，原理也是一樣，寫完後舉起 hcu_valid。

5) Show simulation result



6) Show SuperLint coverage (top.v)



All file waring = 141

Total line number: comparator(137) + CLBP(802) +

controller(160) + DCU(72) + HCU(375) + top(163) = 1709

Coverage = 1 - 141/1709 = 91.75%

7) Your clock period, total cell area, post simulation time

(top.v) in screenshot.

Clock period:2.0

Total cell area:38293.12

Post simulation time: 197933000(ns)

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.

我在存 id 的時候本來有叫一個 counter 下去算目前存到第幾個再將 counter 值送入 id_addr 後來我發現這樣程式的彈性並不高，所以我將其去掉改用，mode 去判別 id_addr 可以加到多少，只要進 model 就不會再增加。

- Lessons learned from this lab

我學到了狀態機真的是一項非常重要的東西，有了狀態機，可以讓整個電路的流程非常清楚，在修這門課之前，我是只會把所以訊號混在 sequential 裡寫，不會運用 FSM 去分割，到最後就是訊號一團亂，但還好現在對於 FSM 有初步的認識，以至於可以順利走到這。

- Suggestions for us (we appreciate your feedback)

沒啥好建議的，助教群實在太電了!

若非要講一個，應該是點名，感覺可以在講授課點就好，主要是並非大家都會在當天做 lab 可能隔天有考試之類的，所以都會提早走，但可能就還是要來電腦教室點名。

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!