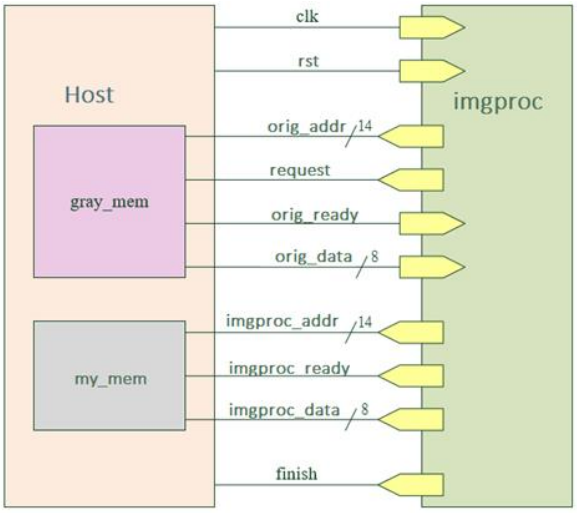
HW4 Verilog – Testbench Design

1. 問題描述

本次練習為撰寫 testbench.v，待測試之電路其輸入為一灰階影像，此灰階影像存放於 Host 端(testbench.v)的灰階影像記憶體模組(gray_mem)中，imgproc 端須發送訊號至 Host 端以索取灰階影像資料，再對灰階影像中每個 pixel 各自進行獨立運算，運算後的結果請寫入 Host 端的記憶體模組(my_mem)內，並在整張影像處理完成後，將 finish 訊號拉為 High，等候系統比對整張影像資料的正確性，輸出模擬結果。有關 imgproc 的過程描述於後。

2. 設計規格



圖一、系統方塊圖

信號名稱	輸出/入	位元寬度	說明
clk	*input*	1	時脈信號，本系統為同步於時脈正緣設計
rst	*input*	1	高位準非同步(active high asynchronous)系統重置訊號
orig_ready	*input*	1	輸入影像資料的 ready 訊號
orig_data	*input*	8	輸入影像資料
orig_addr	*output*	14	要求輸入影像資料之位址
request	*output*	1	輸入影像資料的 request 訊號
imgproc_ready	*output*	14	輸出處理後影像資料的 ready 訊號
imgproc_data	*output*	1	輸出處理後影像資料
imgproc_addr	*output*	8	輸出處理後影像資料的位址
finish	*output*	1	imgproc 完成整張影像處理的 finish 訊號

表一、輸入/輸出訊號

※ 簡單處理步驟:

1. orig_addr 和 request 為一組，一起發送出去，跟 Host 要影像資料。
2. orig_ready 和 orig_data 為一組，一起傳回來，imgproc 端看到 orig_ready 代表 orig_data 可以接收了。
3. imgproc_addr, imgproc_ready, imgproc_data 為一組，一起發送出去，將結果送到 HOST 端。
4. finish 訊號拉為 High 後，等候 testbench 檢查處理後影像資料(my_mem 中)的正確性。

3. 系統功能描述

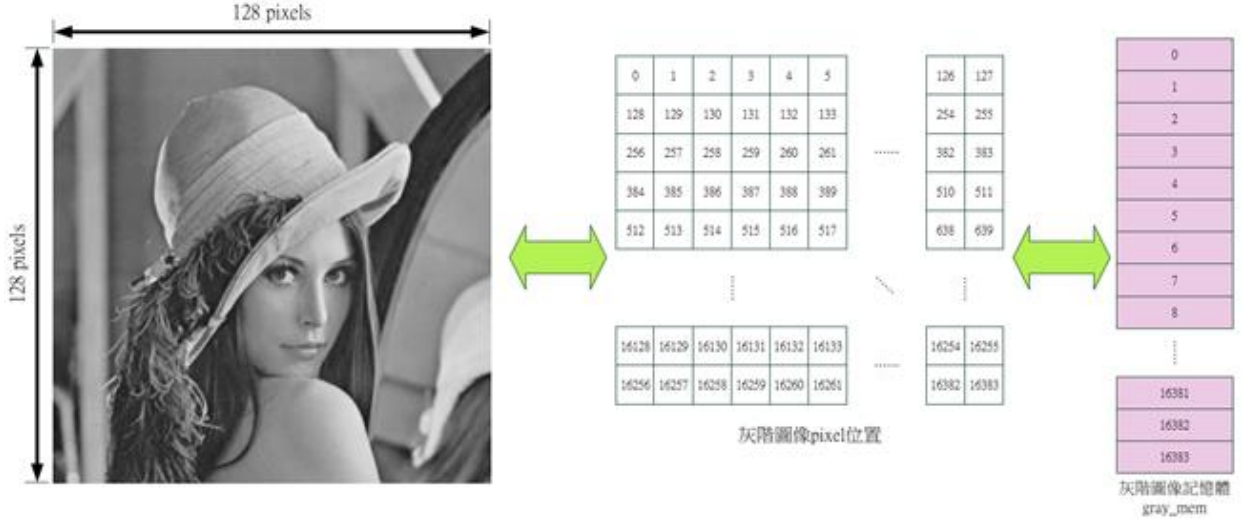
本電路功能為當 reset 結束後，imgproc 端才可開始對 Host 端進行動作。當 Host 端在每個時脈訊號負緣觸發時若偵測到 finish 訊號為 Low 且 request 訊號為 High 時表示 imgproc 端對 Host 端要求索取灰階圖像資料。

當 orig_ready 訊號為 High，此時 Host 端準備好資料，會依 orig _addr 匯流排所指示的位址將灰階影像記憶體內的位址資料由 orig _data 匯流排輸入 imgproc 端。本電路主要功能是將影像亮度線性降低，整張圖片亮度變為原本的一半，小數點部分做四捨五入。

記憶體模組的寫入方式如下，當 Host 端在每個時脈訊號負緣觸發時偵測到 imgproc_ready 訊號為 High 時，就會將目前 imgproc_data 匯流排上的內容，寫入到 imgproc_mem 記憶體模組的 imgproc_addr 匯流排所指示的位址內，當所有 pixel 都處理完畢後，請將 finish 訊號拉為 High，等候 Host 端進行結果驗證。

4. 灰階影像記憶體對應方式

灰階影像大小固定為 128x128 pixels，每個 pixel 為 8bit 灰階，值介於 0 到 255 之間，因此 Host 端的灰階圖像記憶體模組(gray_mem)共有 16384 個位址用以存放各 pixel 的灰階影像資料，記憶體模組用 1D register array 儲存即可，圖像與記憶體模組的對應方式如下圖所示。

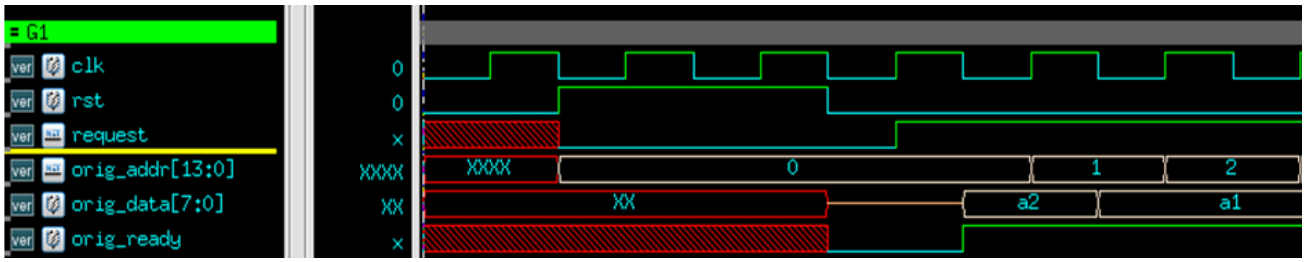


圖二、影像與記憶體對應方式

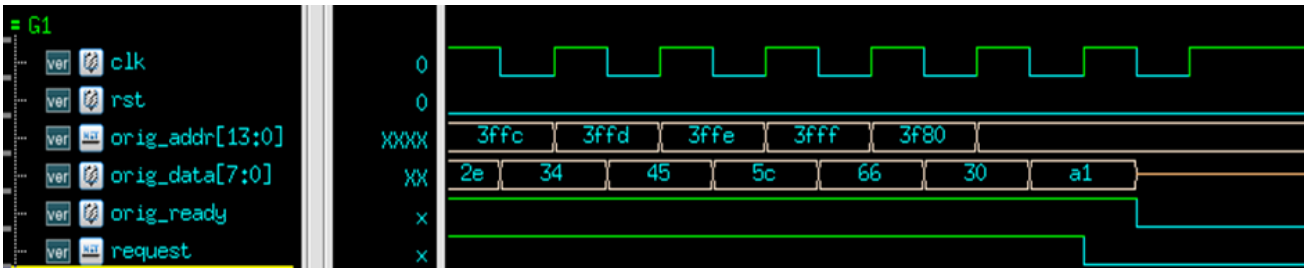
5. 時序規格圖

A. 影像輸入時序圖

1. reset 訊號持續兩個 Cycle 時間後，電路初始化結束。
2. imgproc 端將 **request** 訊號拉為 High，並且同時將欲索取的灰階圖像 pixel 之位址由 **orig_addr** 匯流排送出。
3. Host 端在時脈訊號負緣觸發若偵測到 request 為 High，將 **orig_ready** 拉為 High，則會將灰階圖像記憶體內的 orig_addr 匯流排所指示位址的資料由 **orig_data** 匯流排送到 imgproc 端。若 imgproc 端欲進行連續索取，會將 request 維持在 High，並連續改變 orig_addr 匯流排位址，Host 端需在 orig_data 匯流排連續發送該位址資料。
4. imgproc 端就可以針對各 pixel 進行訊號處理流程。
5. 若 imgproc 端不想要對 Host 端索取任何位址資料，就會在時脈訊號正緣觸發將 request 拉為 Low，則 Host 端在下個時脈訊號負緣觸發時就不會送出任何位址資料到 orig_data 匯流排，且 orig_ready 也會被拉為 Low。



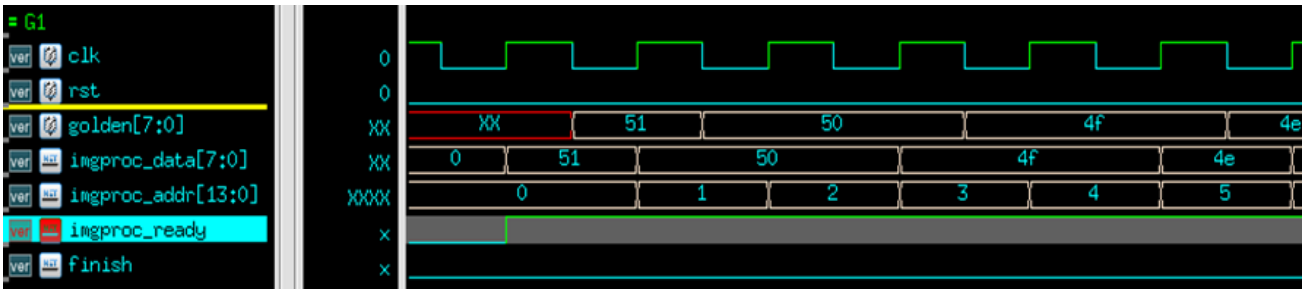
圖三、request 訊號拉為 High，testbench 開始送出資料



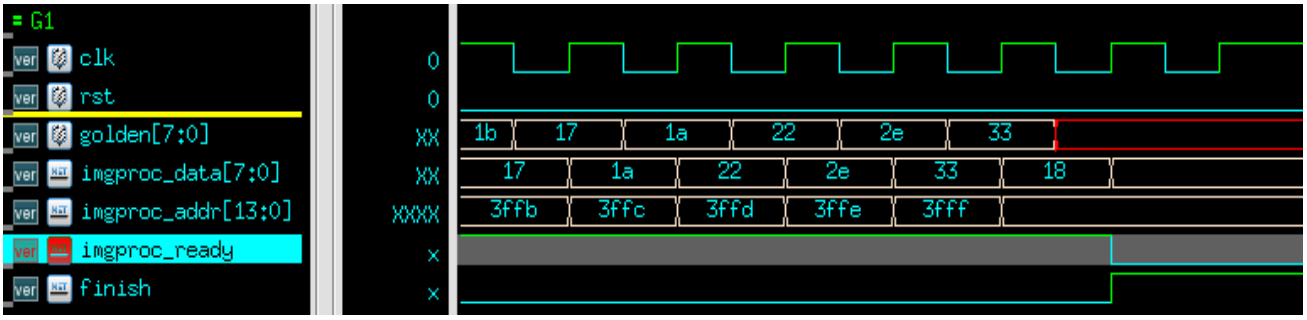
圖四、request 訊號拉為 Low，testbench 結束送出資料

B. 影像輸出時序圖

1. 當 imgproc 端完成處理後，會將各 pixel 的處理結果寫入各相對應的記憶體位址中，其方式為將 **imgproc_ready** 訊號拉為 High，同時把欲寫入的位址及資料分別放在 **imgproc_addr** 及 **imgproc_data** 匯流排；Host 在時脈訊號負緣觸發時，就會進行寫入的動作。若想要連續寫入的話，則會持續將 imgproc_ready 維持在 High 後改變 imgproc_data 及 imgproc_addr。
2. 如果 imgproc 端不想繼續寫入資料的話，則 imgproc_ready 拉為 Low。
3. 所有的 pixel 都處理完成了，此時 imgproc 端會將 finish 拉為 High。等候 Host 端驗證，驗證完成後整個模擬會立即結束。



圖五、imgproc_ready 訊號拉為 High，imgproc 開始送出資料及位址



圖六、finish 訊號拉為 High，等候 testbench 比對資料

6. 檔案說明

檔名	說明
testbench.v	測試樣本檔，此次作業須完成的部分。
imgproc.v	設計檔，已經完成，不須更改。
pattern.dat golden.dat	testbench 所使用的測資。 pattern:依序分別表示輸入灰階影像的各 pixel 數值。 golden:依序分別表示運算後之正確影像的各 pixel 數值。

7. 模擬指令與結果

設計完成的 testbench，使用模擬相關指令如下。

ncverilog testbench.v imgproc.v

設計輸出波形，可以使用 *+define+FSDB* 或者是 *+define+VCD* 並且加上 *+access+r*

ncverilog testbench.v imgproc.v +define+FSDB +access+r

ncverilog testbench.v imgproc.v +define+VCD +access+r

模擬結果的正確與否的顯示方式可以自行設計。testbench 的寫法可參考之前 HW 中提供的檔案。

8. 作業要求

1. 設計的 testbench 能正確地比對 imgproc.v 輸出結果的正確性。可以故意讓 imgprov.v 輸出錯誤的結果，測試 testbench 能否偵錯。
2. 使用 ENDCYCLE 來處理模擬時間逾時。
3. 繳交檔案如下：b0*901***_HW4.zip

分類	檔案名稱	描述
RTL	testbench.v	Testbench Verilog Code
Report	b0*901***_report.pdf	內含自行模擬產生後類似 5.時序規格圖中的**四張時序圖**、影像開始輸出 RTL-Level **模擬結果截圖**、簡單講解自己的 testbench 包含哪些內容。

9. 繳交期限

10/28 (一)中午 12:00 以前上傳至 Ceiba

同學如果有任何問題，請透過 email 詢問助教或約定時間。剛開始學習大家遇到的問題都會蠻像的，如果要寄 email 給助教，記得在信件前加 [專題研究] 避免漏信。

助教 林奕憲 d06943006@ntu.edu.tw
助教 葉陽明 d05943006@ntu.edu.tw