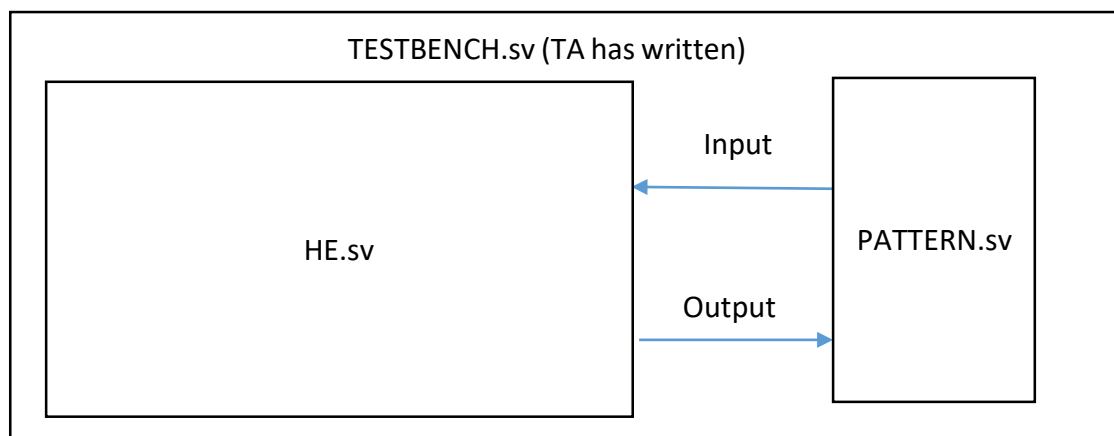


資料準備

1. 從 TA 目錄資料夾解壓縮
% tar -xvf ~dcsta01/HW04.tar
2. 解壓縮資料夾 HW04 包含以下：
 - A. 00_TESTBED/
 - B. 01_RTL/
 - C. 02_SYN/
 - D. 03_GATE/
 - E. 09_UPLOAD/

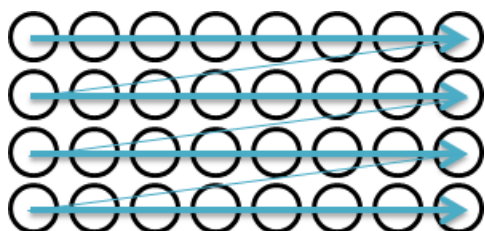
Block Diagram



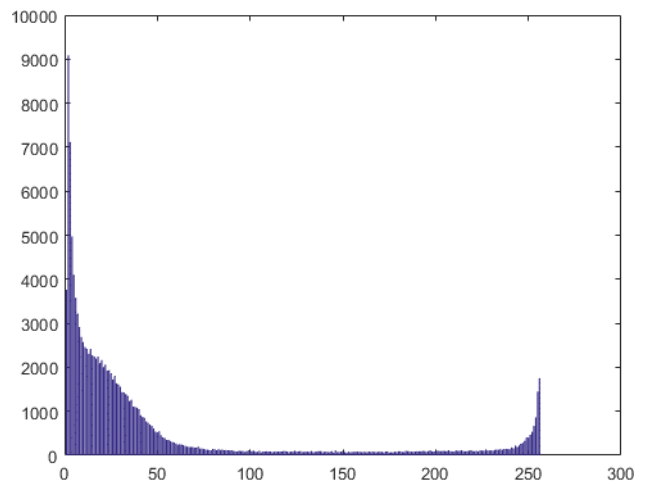
設計描述

這次作業要求利用 **pipeline** 去設計 Histogram Equalizer 之硬體。
Histogram Equalizer 是傳統的影像處理演算法，可以加強灰階圖片的對比度。
演算法如下：

1. 先將圖片做histogram(分布統計):

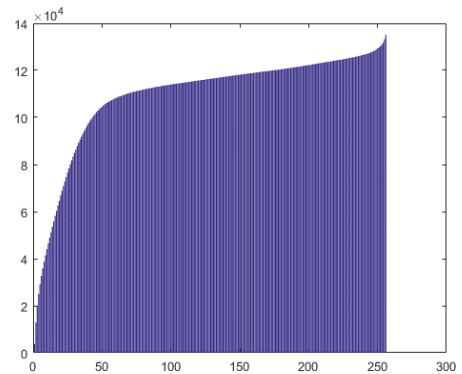
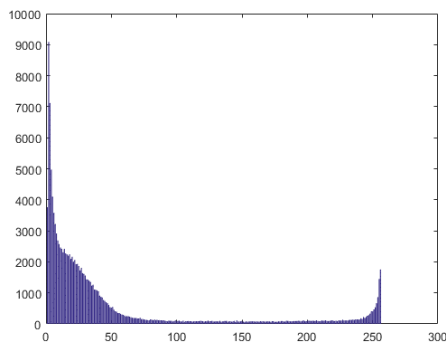


一張圖片依raster scan order(如上)的方式，將pixel 的值(0-255)數量做統計，如下：



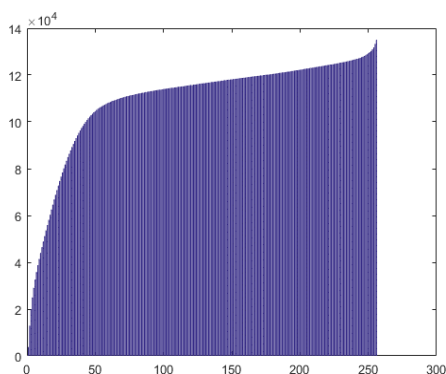
2. 將圖片做Cumulative Histogram(累計分佈統計)

將第一步分佈統計資料做累積分佈，1 資料量為 0 資料量+1 資料量，2 資料量為 0 資料量+1 資料量+2 資料量，255 資料量應為圖片大小資料量。



3. 算出transform function

使用Cumulative Histogram 算出transform function。



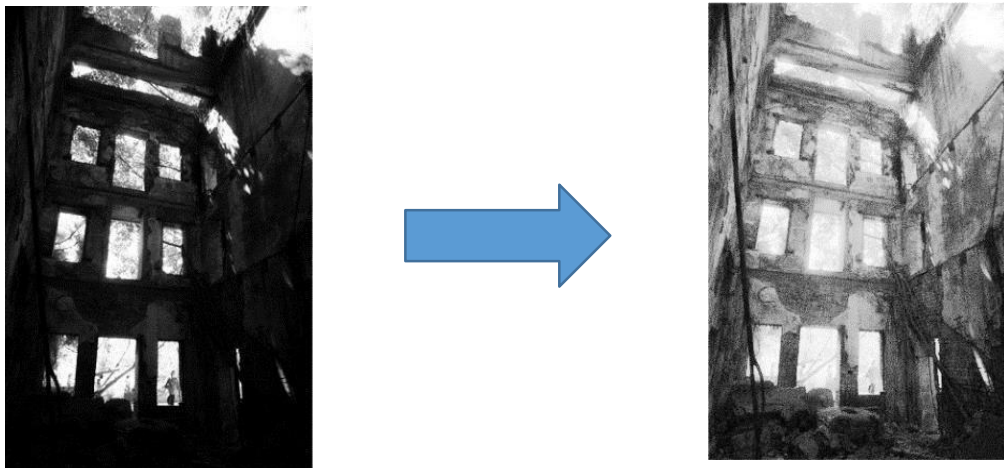
$$\times \frac{937}{4093} - 1 = \text{transform function}$$

注意: pixel 值介於 0-255 之間，這個transform function 可能會讓pixel的值超出範圍，記得必須處理邊界問題。

註: 將937/4093轉為小數也無法通過本次作業，design compiler無法處理小數。

註: 此公式為設計過的，為了是讓你們練習pipeline，實際上的公式並不長這樣。

4. 將原圖每個點使用transform function 轉化成新圖片。



5. 這次你們**必須自己寫pattern 測試，助教只會給予10筆測資**，助教的pattern 會在in_valid = 1 時連續給予1032筆值，前8筆 in_image[7:0]為要修正的pixel 值，後1024 筆 in_image[7:0]依 raster scan order 連續給予 32x32 的圖，利用這些值去統計histogram並修正助教給予的前8筆資料。

Input

Signal name	Number of bit	Description
clk	1 bit	clock
rst_n	1 bit	Asynchronous active-low reset
in_valid	1-bit	Inputs are now valid, 0 for invalid, 1 for valid.
in_image	8 bit	First 8 cycle give the pixels which need to do the histogram equalize. Following 1024 cycle give the original image

Output

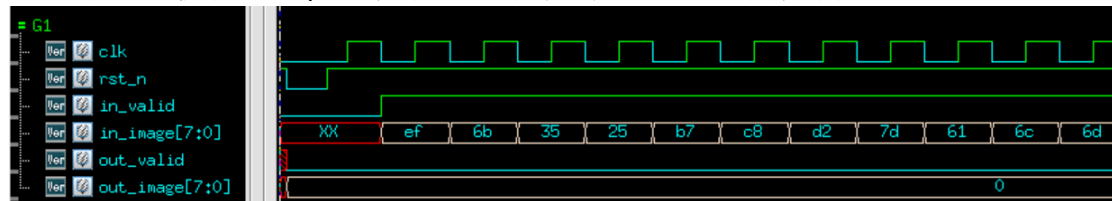
Signal name	Number of	Description
out_valid	1 bit	Should set to high when your image_out[7:0] is ready

out_image	8 bits	TA's pattern will check your result.
-----------	--------	--------------------------------------

Example Waveform

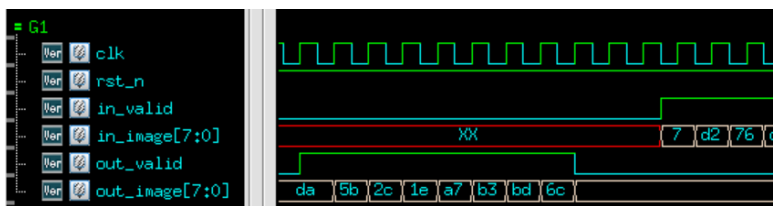
Input

前 8 筆為要被轉換的 pixel 資料，接下來就是 32x32 圖片資料。



Output

連續拉起 8 cycle out_valid，分別輸出已被轉換的 pixel 資料，Out_valid 結束後下 3 個 negedge 內給予下一組 in_valid。



Specification

1. Top module name : HE (File name: HE.sv) (-5% if naming error)
2. 所有 output 必須為 0，在非同步負準位 reset。
3. In_valid 落下後 1000 cycle 之內 out_valid 必須拉起。
4. 02_SYN result 不行有 error 且不能有 latches、slack 必須為 MET。
5. 01_RTL & 03_GATE 不能有 timing violation 且通過助教測資。
6. Clock period 5ns。
7. Input delay = 0.5 * clock period; output delay = 0.5 * clock period。
8. 不要使用 for loop

上傳檔案

1. 程式碼請使用 09_upload 上傳。
2. report_dcsxx.pdf，xx is your server account. 上傳至 new e3。(naming error -5%)
3. 1de 請在 2023/4/27 23:59:59 之前上傳
4. 2de 請在 2023/5/4 23:59:59 之前上傳

Grading Policy

1. Pass the 01_RTL & 02_Synthesis & 03_GATE : 60%
2. Performance: 30%
Ranking formula: total latency * area
3. Report: 10%

Note

Template folders and reference commands:

1. 01_RTL/ (RTL simulation) ./01_run
2. 02_SYN/ (Synthesis) ./01_run_dc
3. 03_GATE/(GATE simulation) ./01_run

報告請簡單且重點撰寫，不超過兩頁 A4，並包括以下內容

1. 描述你的設計方法，包含但不限於如何加速(減少 critical path)或降低面積。
2. 基於以上，畫出你的架構圖(Block diagram)與 FSM diagram，以及你如何運用 pipeline去設計你的design。
3. 心得報告，不侷限於此次作業，對於作業或上課內容都可以寫下。
4. 遇到的困難與如何解決。