1. 問題描述

局部二值模式(Local Binary Patterns, LBP)可用於描述局部紋理特徵的計算。本題請完成一 Local Binary Patterns (後文以 LBP 表示)，輸入為一灰階影像(如圖 1.所示)，此灰階影像存放於 Host 端的灰階圖像記憶體模組(gray_mem)中，LBP 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 端的局部二值模式記憶體模組 (lbp_mem)內，並在整張影像訊號處理完成後，將 finish 訊號拉為 High，接著系統會自動進行比對 整張影像資料的正確性。有關 LBP 的定義與運算方式，描述於後。

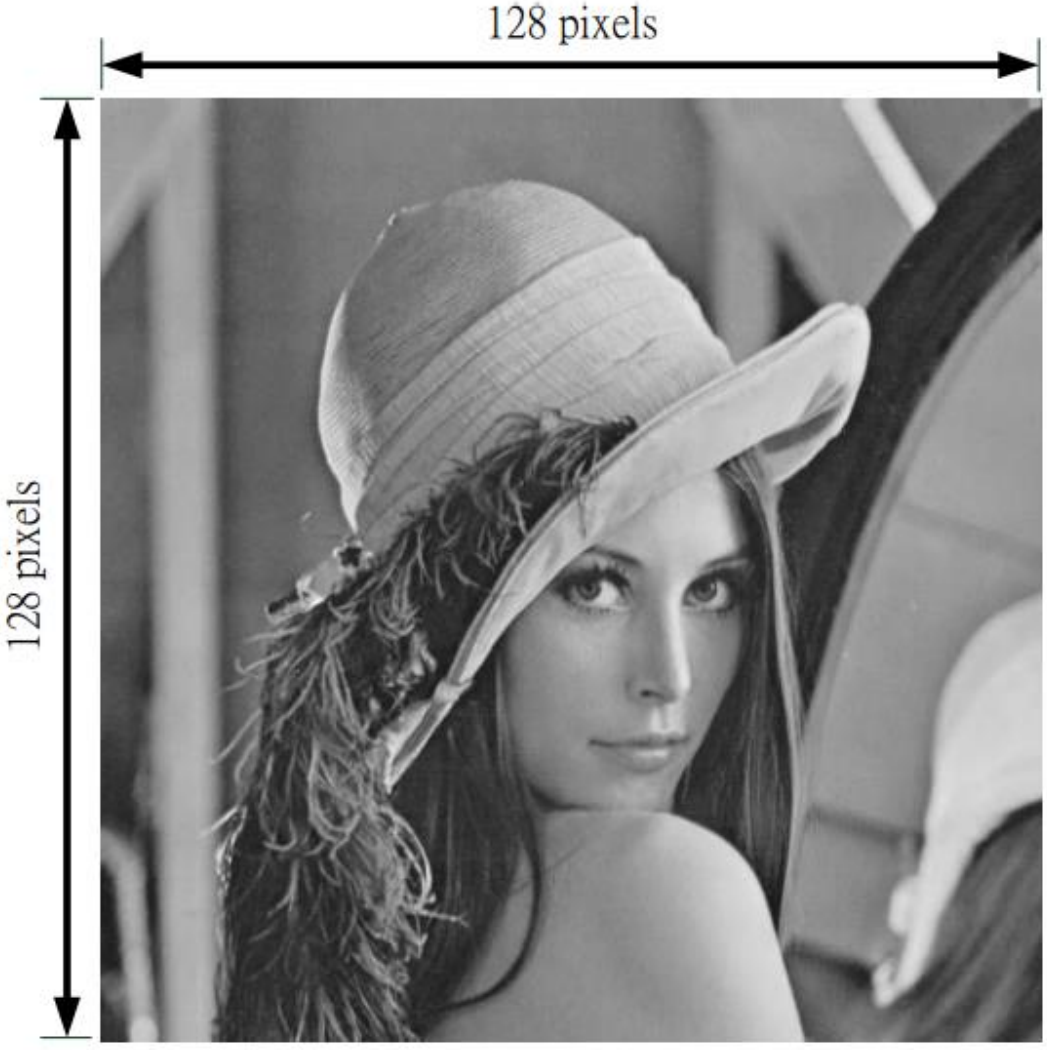


圖 1. 灰階影像範例

2. 設計規格

2.1 系統方塊圖

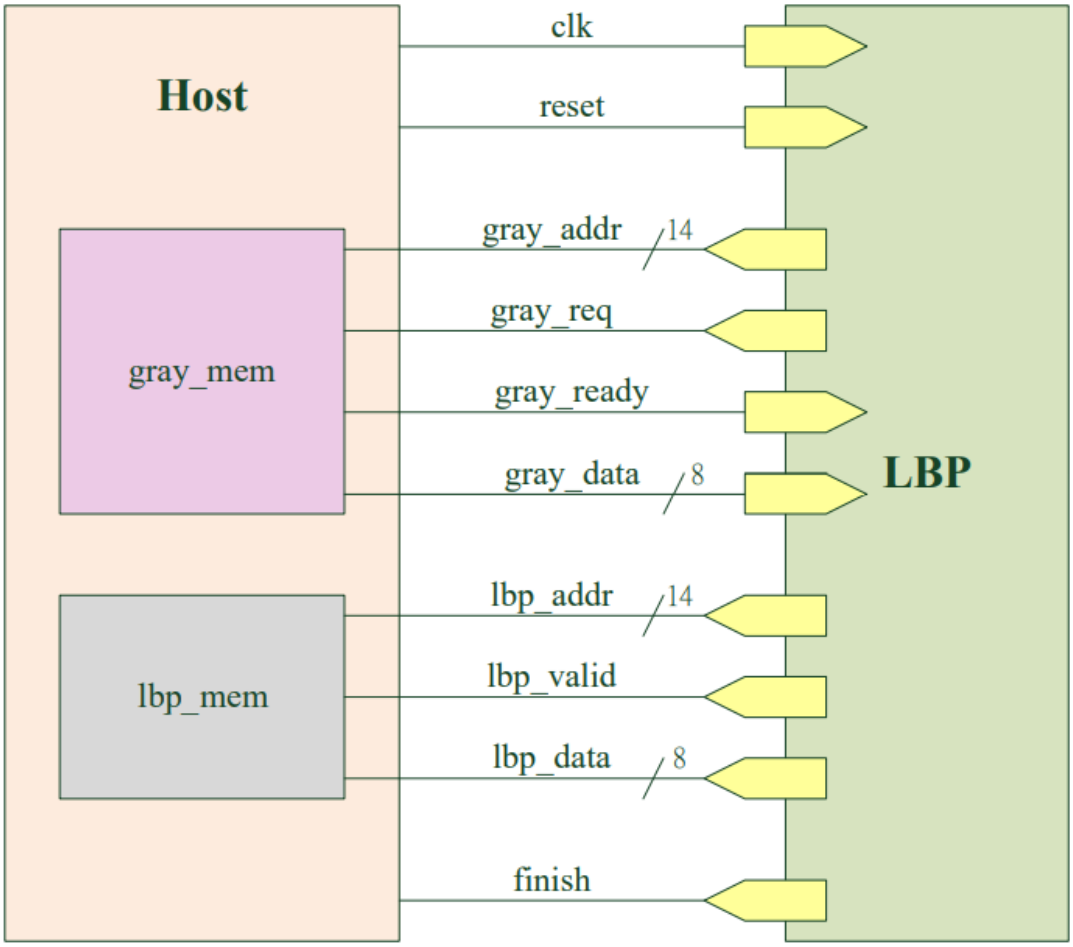


圖 2. 系統方塊圖

2.2 輸出入訊號和記憶體描述

表一、輸入/輸出信號

Signal Name	I/O	Width	Simple Description
clk	I	1	本系統為同步於時脈正緣之同步設計。
reset	I	1	高位準”非”同步(active high asynchronous)之系統重置信號。
gray_addr	O	14	灰階圖像位址匯流排。LBP 端需透過此匯流排向 Host 端的灰階圖像記憶體索取該位址的灰階影像資料。 每一個週期僅能索取一個位址的資料。 題目不限制位址及資料的索取次數。
gray_req	O	1	灰階圖像索取致能訊號。當為 High 時，表示 LBP 端要向 Host 端索取灰階圖像資料。
gray_ready	I	1	灰階圖像資料指示訊號。當為 High 時，表示 Host 端已經將灰階圖像記憶體及相關訊號準備完成了；LBP 端需在偵測到此訊號為 High 後才可以開始對 Host 端進行資料索取動作。
gray_data	I	8	灰階圖像資料匯流排。Host 端利用此匯流排將灰階圖像記憶體內的灰階圖像資料送到 LBP 端。
lbp_addr	O	14	局部二值模式位址匯流排。LBP 端利用此位址將經 LBP 運算完成後之資料儲存至局部二值模式記憶體中。
lbp_valid	O	1	局部二值模式資料致能訊號。當為 High 時，表示 LBP 端所傳輸之局部二值模式資料及位址匯流排為有效的。
lbp_data	O	8	局部二值模式資料匯流排。LBP 端需透過此匯流排指定局部二值模式資料要儲存到局部二值模式記憶體中的哪個位址。
finish	O	1	LBP 運算完畢之通知訊號。當所有的灰階圖像資料經過個別運算完畢且儲存後，需將 *finish* 訊號拉為 High，以通知 Host 端，開始進行所有資料之比對。

2.3 系統功能描述

本電路功能為當 reset 結束後，Host 端會將 gray_ready 訊號拉為 High 表示資料準備完成，之後 LBP 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 gray_req 訊號為 High 時表示 LBP 端對 Host 端要求索取灰階圖像資料，此時 Host 端會依 gray_addr 匯流排所指示的位址將灰階圖像記憶體內的位址資料由 gray_data 匯流排輸入 LBP 端。

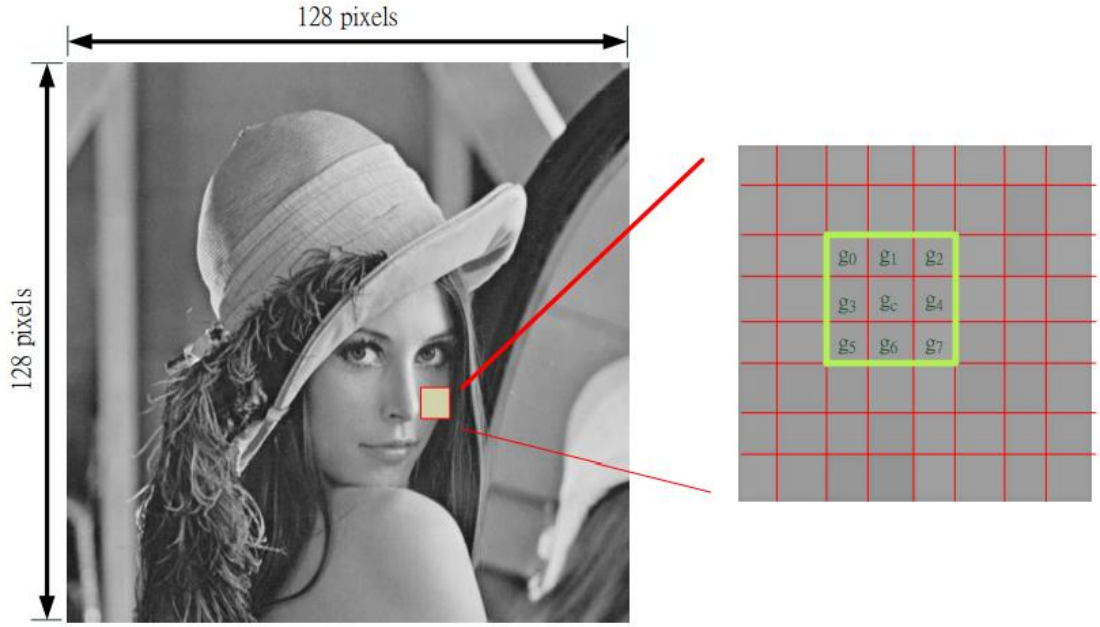


圖 3. LBP 處理區域

由 gray_data 輸入的有效灰階圖像資料須經過 LBP 編碼才可得到區域二值模式資料，LBP 編碼方式為利用每個 pixel 及其相鄰的數個 pixel 的相對應關係來計算，以如上圖 3.所示的灰階圖像架構來說明，若待處理 pixel 為 g_c ，由 g_c 向外擴張數個 pixel 為一區域，故每個區域中心點 pixel 的灰階值為 g_c ，而區域中心點相鄰 pixel 的灰階值為 g_p(P=0,1,....,P-1)，本題限定區域範圍為中心點向外擴張一個 pixel 距離的正方形區域(如上圖 3.綠色框框所示)，因此每個區域為一 3x3 pixels，所以每個 g_c都有 8 個相鄰 pixels(P=8)。如果g_c的座標位置為(x, y)，則

$$LBP(x,y)=\sum_{p=0}^{P-1}s(g_p-g_c)2^p \quad \text{，而 } s(z)=\begin{cases}1, & z\geq 0\\ 0, & otherwise\end{cases}$$

舉例說明，若區域內容如下圖 4.綠色框框所示，g_c則為黑色圓圈位置，則利用上式所計算出各g_p的 Threshold 值 s(z)就如圖 4.紅色框框所示，將各g_p的 Threshold 值乘上各位置的權重值 2^p (如圖 4.藍色框框)就可以得到如圖 4.紫色框框所示結果，因此該區域 g_c的 LBP 運算結果就是將紫色框框內所有 pixels 的值相加即可得到。

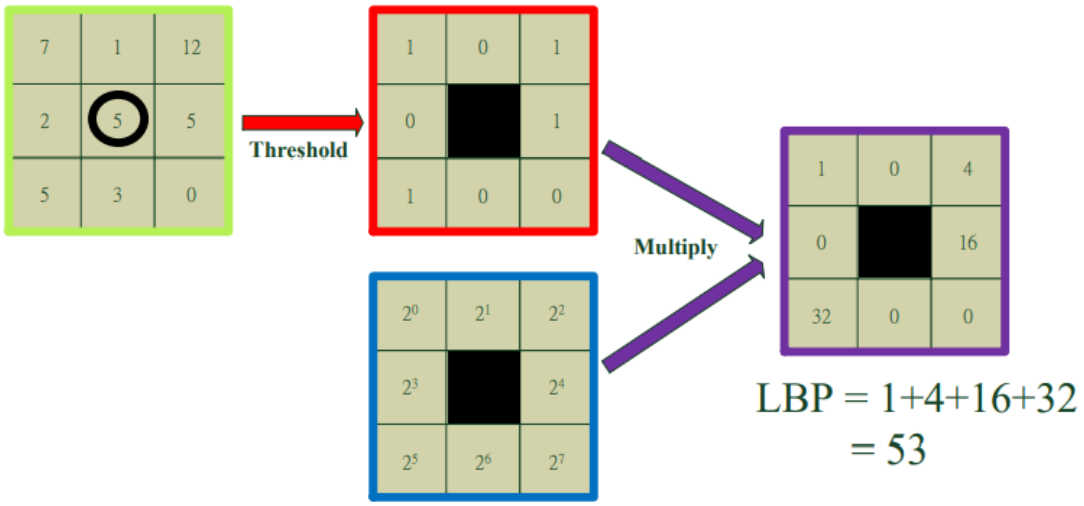


圖 4. LBP 運算範例

計算完成後，接著要將各位址 pixel 的 LBP 運算結果存到局部二值模式記憶體模組內，本題規定由第 k(k=0,1,....,16383)個灰階圖像記憶體(gray_mem)位址所讀取的灰階圖像資料經 LBP 運算後的結果須存到局部二值模式記憶體模組(lbp_mem)的第 k 位址；另本題要求灰階圖像最外圈一圈的 pixel 不須做 LBP 運算，並且這一圈的 pixel 在局部二值模式記憶體模組的數值須為 0，如下圖 6. 所示。為簡化題目難度，Host 端會初始化整個局部二值模式記憶體模組的所有數值為 0。

局部二值模式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若偵測到 lbp_valid 訊號為 High 時，就會將目前 lbp_data 匯流排上的內容，寫入到 lbp_mem 記憶體模組的 lbp_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，接著 Host 端就會開始進行結果驗證。

2.3.1 灰階圖像記憶體對應方式

灰階圖像大小固定為 128x128 pixels，每個 pixel 為 8bit 灰階(每個 8bit 灰階圖像 pixel 的值介於 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 的灰階圖像資料，圖像與記憶體模組的對應方式如下圖 5.所示。

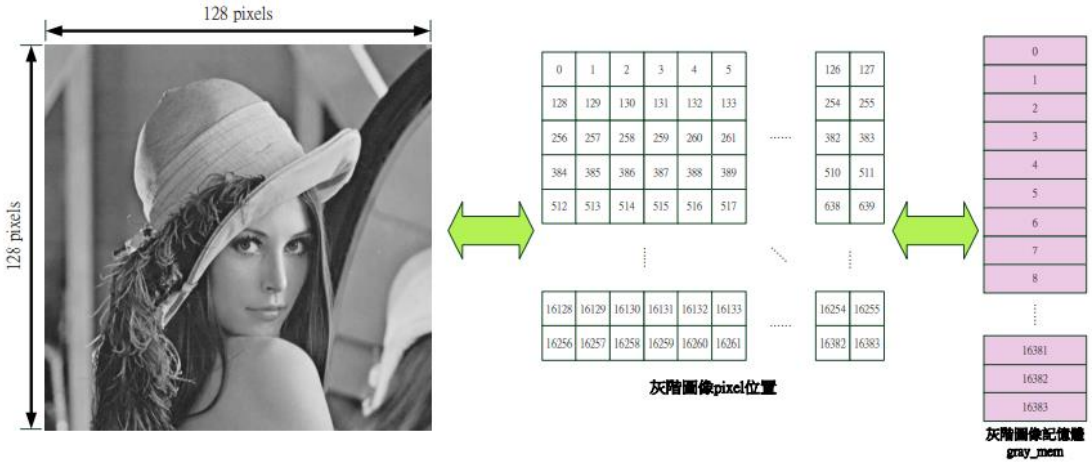


圖 5. 灰階圖像記憶體對應方式

2.3.2 局部二值模式記憶體對應方式

局部二值模式圖像為 128x128 pixels，每個 pixel 為 8bit，因此 Host 端的局部二值模式記憶體 模組(lbp_mem)共有 16384 個位址用以存放各 pixel 的處理結果，本題目規定最外圍一圈 pixel 的值須為 0，因此 LBP 處理結果及局部二值模式記憶體的對應方式及處理結果應如下圖 6.所示。

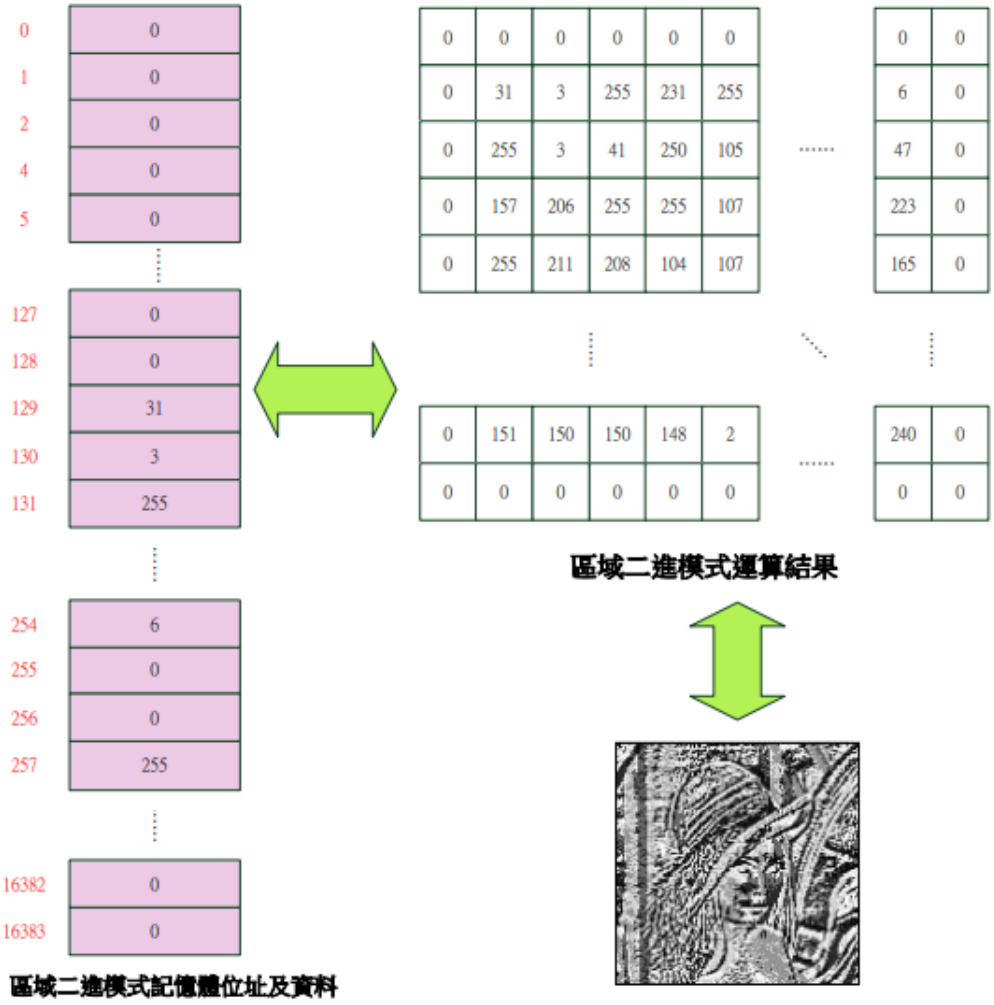


圖 6.局部二值模式記憶體位址方式對應及運算結果

2.4 時序規格圖

系統輸入/輸出時序規格圖及參數，分別如圖 7.及圖 8.所示。

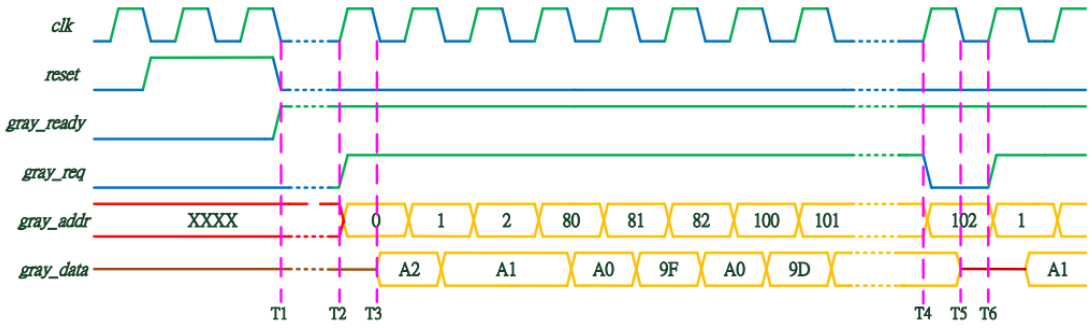


圖 7. 時序規格圖一

a、 T1 時間點，reset 訊號持續兩個 Cycle 時間後，LBP 電路初始化結束，Host 端在 T1 時間點將 gray_ready 拉為 High，表示 Host 端準備接受 LBP 端的資料索取動作。

b、 LBP 端在收到 Host 端發出了 gray_ready 為 High 之後，在 T2 時間點將 gray_req 訊號拉為 High， 並且同時將欲索取的灰階圖像 pixel 之位址由 gray_addr 匯流排送出。

c、 Host 端在時脈訊號負緣觸發若偵測到 gray_req 為 High，則會將灰階圖像記憶體內的 gray_addr 匯流排所指示位址的資料由 gray_data 匯流排送到 LBP 端，此時為 T3 時間點。若要進行連續 索取，只需要將 gray_req 維持在 High，並連續改變 gray_addr 匯流排位址，就可在 gray_data 匯流排連續得到該位址資料。

d、 接著 LBP 端就可以針對各 pixel 進行區域二值模式訊號處理流程。

e、 若 LBP 端不想要對 Host 端索取任何位址資料，則只須在 T4 時間點將 gray_req 拉為 Low，則 Host 端在 T5 時間點就不會送出任何位址資料到 gray_data 匯流排。

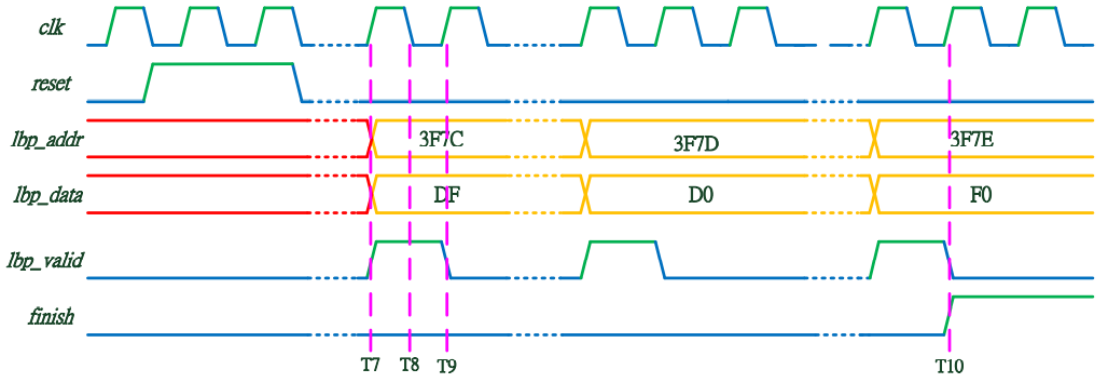


圖 8. 時序規格圖二

f、 當 LBP 端完成區域二值模式處理後，請將各 pixel 的處理結果寫入各相對應的區域二值模式記 憶體位址中，其方式為在 T7 時間點將 lbp_valid 訊號拉為 High，同時把欲寫入的位址及資料 7 分別放在 lbp_addr 及 lbp_data 匯流排；Host 在 T8 時間點的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，則只需要持續將 lbp_valid 維持在 High 後改變 lbp_data 及 lbp_addr 即可。如果不想繼續寫入資料的話，請在 T9 時間點將 lbp_valid 拉為 Low。

g、 T10 時間點，所有的 pixel 都處理完成了，此時 LBP 端須將 finish 拉為 High。Host 端就會開始進行驗證了，驗證完成後整個模擬會立即結束。

3.1 請使用 LBP.v，進行本題電路之設計。其 Verilog 模組名稱、輸出/入埠宣告如下所示：

```
`timescale 1ns/10ps
module LBP ( clk, reset, gray_addr, gray_req, gray_ready, gray_data, lbp_addr,
      lbp_valid, lbp_data, finish);
   input clk;
   input reset;
   input gray_ready;
   input [7:0] gray_data;
   output [13:0] gray_addr;
   output gray_req;
   output [13:0] lbp_addr;
   output lbp_valid;
   output [7:0] lbp_data;
   output finish;
   //================================================================
   //your code
   //================================================================
   endmodule
```

3.2 需要完成的檔案為 LBP.v，其他所提供的檔案包括:

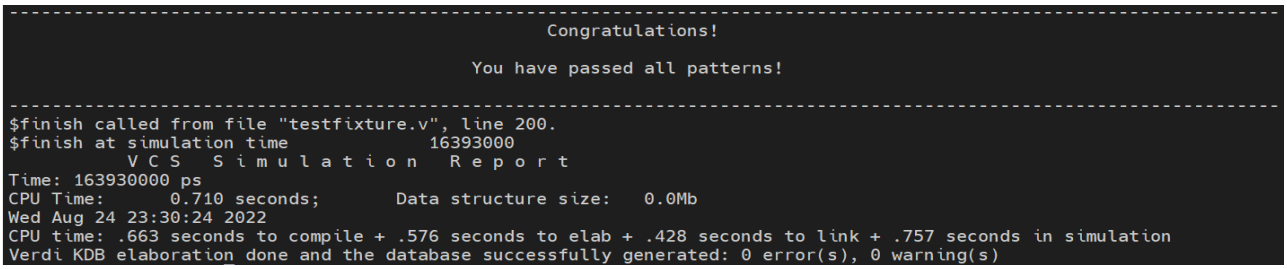
testfixture.v: 模擬用 testbench，會使用到以下兩個檔案
pattern1.dat: 模擬用 pattern 資料
golden1.dat: golden 資料，用來比對答案是否正確

3.3 本題目僅要求完成 RTL 模擬通過，模擬工具可選擇使用

Vivado/Modelsim/Icarus Verilog/NCverilog/VCS 等你所熟悉的 tool，若你沒有使用過這些 tool，建議使用前三種之一，都可以在 windows 系統安裝執行，RTL Simulation 時使用指令，以 modelsim 為例:

- vlog testfixture.v 或者使用 GUI 操作，Vivado 使用方法類似。

另 Vivado/Modelsim 都有內建波形工具可供 debug。
跑完模擬看到"Congratulations!You have passed all patterns!代表通過。



若有問題，可來信 pet112358@gmail.com