

# 2017 IC Design Contest Preliminary

## 研究所類標準元件數位電路設計

### Distance Transform

#### 1. 問題描述

距離變換(Distance Transform , DT)是一種應用在二值圖像處理的演算法，其運算結果為一灰階圖像，此灰階圖像與一般灰階圖像不同，其灰階圖像強度並非表示亮度值，而是表示物件內部每一點與物件邊緣的距離。

本題請完成一 Distance Transform 電路(後文以 DT 電路表示)，其輸入為一張二值化圖像(如圖 1.所示)，此二值化圖像存放於 Host 端的**輸入圖像 ROM 模組(sti\_ROM)**中，DT 電路須從 Host 端的 sti\_ROM 記憶體模組讀取二值化圖像資料，再對此圖像資料進行 distance transform 運算，運算後的結果需寫入 Host 端的**輸出結果圖像 RAM 模組(res\_RAM)**內，並在整張圖像處理完成後，將 *done* 訊號拉為 High，接著系統會比對整張圖像資料的正確性。有關本題電路訊號的定義與 DT 運算方式，詳述於後。

本電路各輸入輸出信號的功能說明，請參考表一。本電路系統方塊圖請參考圖 2。

每個參賽隊伍必須根據下一節所給的設計規格及附錄 B 中的測試樣本完成設計驗證。

本次 IC 設計競賽比賽時間為上午 08:30 到下午 08:30。當 IC 設計競賽結束後，CIC 會根據第 3 節中的評分標準進行評分。為了評分作業的方便，各參賽隊伍應參考附錄 C 中所列的要求，附上評分所需要的檔案。

軟體環境及設計資料庫說明請參考附錄 A 與附錄 B。



圖 1. 二值化影像範例

## 2. 設計規格

請注意：

1. 各模組及檔案名稱、大小寫須完全符合**附錄B規範**，若有引入其它模組或檔案請自行寫在設計檔內，測試檔不允許任何修改否則不予計分。
2. 最後評分方式為使用最後上傳檔案版本評分，並以最後上傳檔案版本時間為依據，請參考3.評分標準。

### 2.1 系統方塊圖

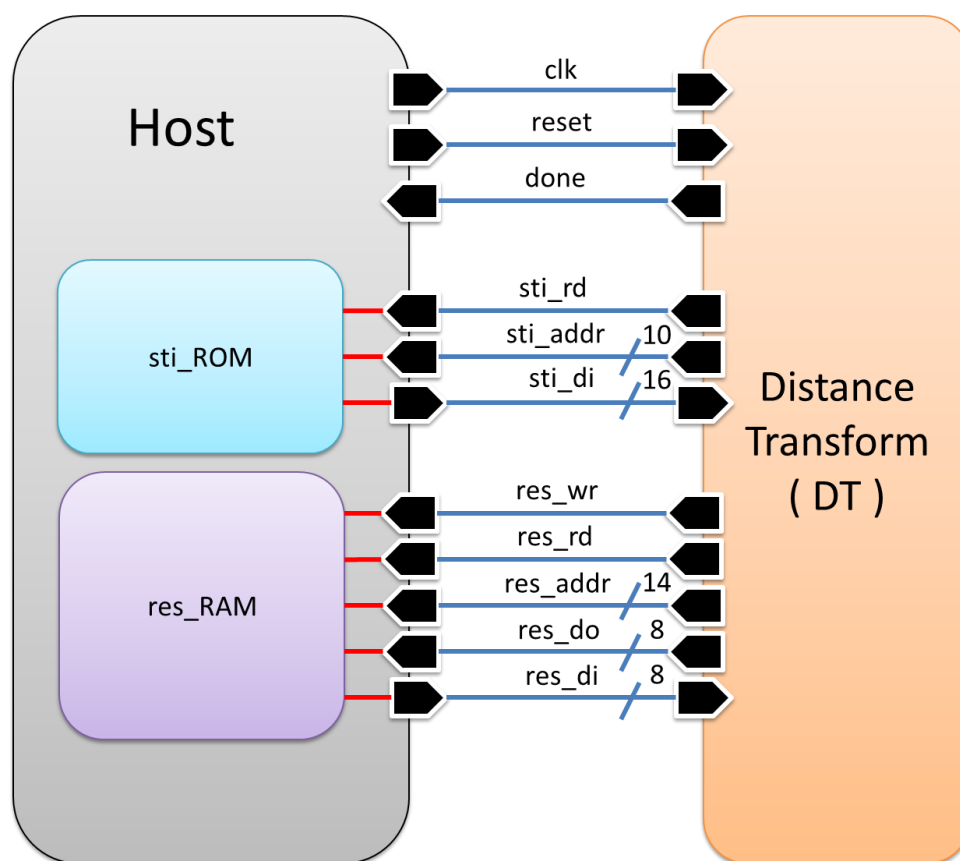


圖 1. 系統方塊圖

### 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	系統時脈訊號。
reset	I	1	低位準”非”同步(active high asynchronous)之系統重置信號。
sti_rd	O	1	sti_ROM 記憶體讀取致能訊號。當為 <b>High</b> 時，表示 DT 端要向 Host 端索取資料。

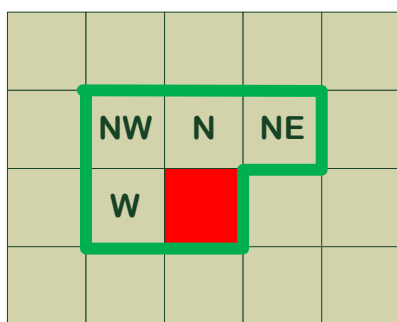
sti_addr	O	10	sti_ROM 記憶體位址匯流排。DT 端需透過此匯流排向 Host 端的 sti_ROM 記憶體指示讀取該位址的二值化圖像資料。 每一個週期僅能讀取一個位址的資料。 題目不限制位址及資料的讀取次數。
sti_di	I	16	sti_ROM 記憶體讀取資料匯流排。Host 端利用此匯流排將 sti_ROM 記憶體內的二值化圖像資料讀取送到 DT 端。
res_wr	O	1	res_RAM 記憶體寫入致能訊號。當為 High 時，表示 DT 端要向 Host 端的 res_RAM 記憶體寫入資料。
res_rd	O	1	res_RAM 記憶體讀取致能訊號。當為 High 時，表示 DT 端要向 Host 端的 res_RAM 記憶體讀取資料。
res_addr	O	14	res_RAM 記憶體位址匯流排。DT 端需透過此匯流排向 Host 端的 res_RAM 記憶體指示讀取或寫入該位址的圖像資料。 每一個週期僅能讀取及寫入一個位址的資料。 題目不限制位址及資料的讀取及寫入次數。
res_do	O	8	res_RAM 記憶體寫入資料匯流排。DT 端利用此匯流排將圖像資料寫入到 Host 端的 res_RAM 記憶體模組。
res_di	I	8	res_RAM 記憶體讀取資料匯流排。Host 端利用此匯流排將 res_RAM 記憶體內的圖像資料讀取送到 DT 端。
done	O	1	DT 電路之 distance transform 運算完畢通知訊號。當 Host 端偵測到此訊號為 High 時，就會立刻使用 res_RAM 記憶體模組內的資料進行運算結果資料比對。

## 2.3 系統功能描述

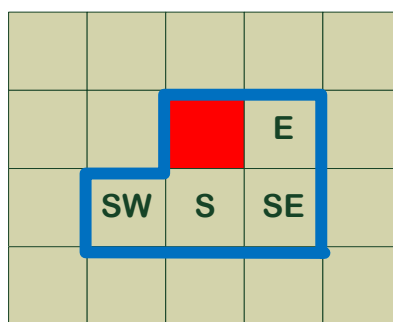
在 reset 訊號啟動時，Host 端會將二值圖像資料存放於 sti\_ROM 中，DT 電路需在 reset 訊號結束後，自行發送控制 sti\_ROM 記憶體的 sti\_rd 讀取致能訊號、sti\_addr 位址訊號給 Host 端，Host 端的 sti\_ROM 記憶體會在收到以上訊號後，利用 sti\_di 資料訊號匯流排將二值圖像資料送到 DT 電路(記憶體時序及資料儲存規格請參考 2.3.1)。DT 端收到二值圖像資料後，須利用以下 distance transform 演算法運算出灰階圖像資料，而 distance transform 演算法說明如下：

假設一圖像物件內部的兩點  $p_1 = (x_1, y_1)$  及  $p_2 = (x_2, y_2)$ ，其距離若以 Chessboard distance 距離測度表示則為  $D_s(p_1, p_2) = \max(|x_1 - x_2|, |y_1 - y_2|)$ 。如果以 1 表示物件像素，而 0 為背景像素，則

distance transform 定義為 對於每一物件區域的像素，計算其與最近的背景像素的距離，並以此距離值來取代原物件像素值。因此演算法可以下述方式進行(本題採用 8-distance transform 方式)，首先定義每個像素可取如下所示兩種 window，左邊的 window 用於前向計算(Forward pass)，右邊的 window 則用於後向計算(Backward pass)。



Forward pass window



Backward pass window

限定本題目標為一張二值化圖像，其大小固定為 128x128 pixels，若像素值為 1 表示為物件，若像素值為 0 表示為背景，**本題測試樣本給定物件區域不會延伸到二值化圖像最外圍一圈的像素。**演算法分為三步驟，步驟如下：

1. Initialize the image pixels

Label the pixels  $p_{x,y} = \begin{cases} 1 & , \text{ if this pixel is 物件 (black) } \\ 0 & , \text{ if this pixel is 背景 (white) } \end{cases}$

2. Forward pass

整張圖依序**由左到右，由上而下**讀入每個二值化圖像像素，針對每一物件像素  $p_{x,y}$  (上圖紅色位置像素) 計算 Forward pass window 中除了物件像素外的所有像素(上圖 NW、N、NE 及 W 位置像素)的最小值為  $x$ ，最後把此紅色位置的物件像素值  $p_{x,y}$  更新為  $x+1$ 。

if  $p_{x,y}$  為物件，則  $p_{x,y} = \min(p_W, p_{NW}, p_N, p_{NE}) + 1$

3. Backward pass

整張圖依序**由右到左，由下到上**讀入上述 forward pass 步驟運算完成的每一灰階圖像像素，針對每一物件像素  $p_{x,y}$  (上圖紅色位置像素) 計算 Backward pass window 中除了物件像素外的所有像素(上圖 E、SE、S 及 SW 位置的像素)的最小值為  $z$ ，接著把此紅色位置的物件像素值  $p_{x,y}$  與  $(z+1)$  比較後取最小值為  $y$ ，再將物件像素值  $p_{x,y}$  更新為  $y$ 。

if  $p_{x,y}$  為物件，則  $p_{x,y} = \min(p_{x,y}, p_E + 1, p_{SE} + 1, p_S + 1, p_{SW} + 1)$

如此即可得到 distance transform 後的灰階圖像資料。

為方便參賽者了解以上流程，本題展示一說明範例於附錄 E，請自行參考。

接著要將灰階圖像資料寫入 Host 端的 res\_RAM 記憶體內，DT 端須利用 res\_wr 記憶體寫入致能訊號、res\_addr 位址訊號及 res\_do 輸出資料訊號給 Host 端以進行 res\_RAM 記憶體寫入動作(記憶體時序及資料儲存規格請參考 2.3.2)，當 DT 端完成灰階圖像的所有像素的資料寫入動作後，須將 done 訊號設定為 High，表示所有動作完成。此時 Host 端就會自動進行結果比對動作。

Host 端的 res\_RAM 為可重複讀寫的 RAM 記憶體模組，因此參賽者在進行 distance transform 運算過程中可重複利用此記憶體進行資料寫入與讀取的工作。若要讀取記憶體內的位址資料，只需

將 res\_rd 記憶體讀取致能訊號設定為 High，並利用 res\_addr 位置訊號送出位址，接著 res\_RAM 就會利用 res\_di 輸入資料匯流排將資料送到 DT 端(記憶體時序及資料儲存規格請參考 2.3.2)。

### 2.3.1 二值化圖像記憶體(sti\_ROM)對應方式及時序規格

二值化圖像大小固定為 128x128 pixels，每個 pixel 為 1 bit 資料，因此 Host 端的二值化圖像共有 16384 個 pixel；而 sti\_ROM 記憶體大小為 16bits 資料寬度及 1024 個位址。每個位址的 16bits 資料寬度剛好可存放 16 個 pixels 的二值圖像資料，二值圖像資料放置於 sti\_ROM 的方式為依序由左到右，由上往下存放，因此對應方式為...

二值圖像的 0 ~ 15 pixels 放置於 sti\_ROM 的第 0 個位址

二值圖像的 16 ~ 31 pixels 放置於 sti\_ROM 的第 1 個位址

二值圖像的 32 ~ 47 pixels 放置於 sti\_ROM 的第 2 個位址，依此類推.....

因此二值圖像的 112 ~ 127 pixels 放置於 sti\_ROM 的第 7 個位址

而 二值圖像的 16368 ~ 16383 pixels 放置於 sti\_ROM 的第 1023 個位址，其他依此類推如下圖 3.所示。

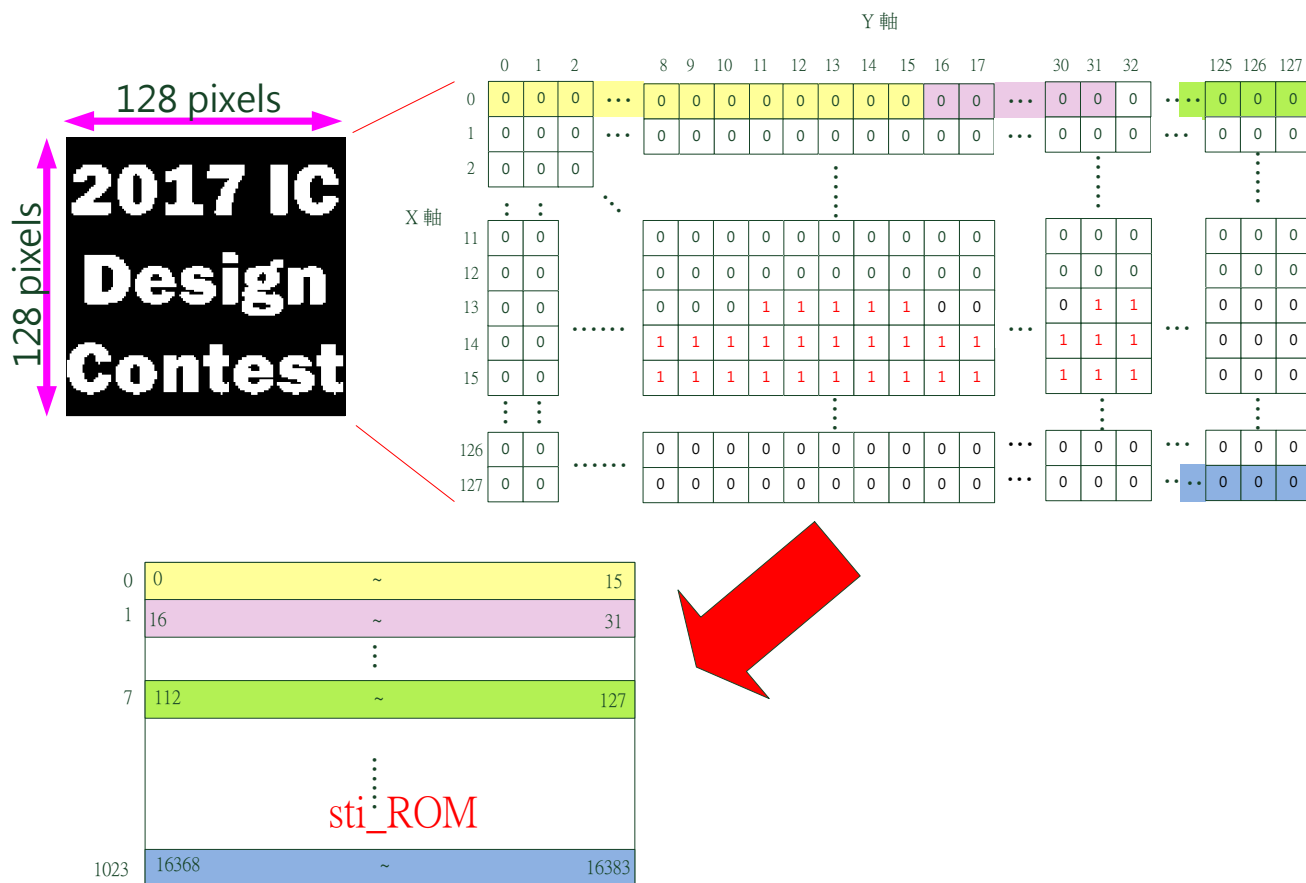


圖 3. 二值化圖像記憶體對應方式

sti\_ROM 記憶體的時序如下圖所示。sti\_ROM 會在每個負緣時脈訊號時，若 sti\_rd 訊號為 High(如下圖 4.的 T1 及 T2 時間)，則會將 sti\_addr 訊號所指定之位址的資料立即由 sti\_di 匯流排送到 DT 端；如果 sti\_rd 為 Low (如下圖 4.的 T3 時間)，則 sti\_ROM 將不會進行任何動作。本記憶體不需考慮讀取延遲的問題

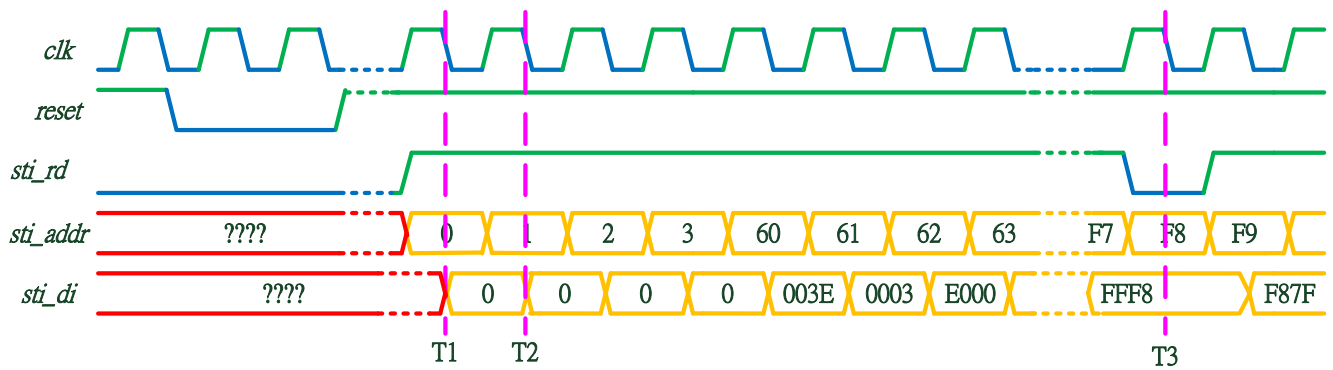


圖 4. 二值化圖像(sti\_ROM)記憶體時序

### 2.3.2 輸出結果圖像記憶體(res\_RAM)對應方式及時序規格

經 distance transform 運算完成的輸出灰階圖像為 128x128 pixels，每個 pixel 為 8bit，因此 Host 端的輸出結果圖像記憶體模組(res\_RAM)共有 16384 個位址用以存放各 pixel 的處理結果。DT 電路需將運算完成之灰階圖像資料，分別將各 pixel 依序由左到右、由上到下寫到輸出結果圖像記憶體內，應對方式如下圖 5.所示。

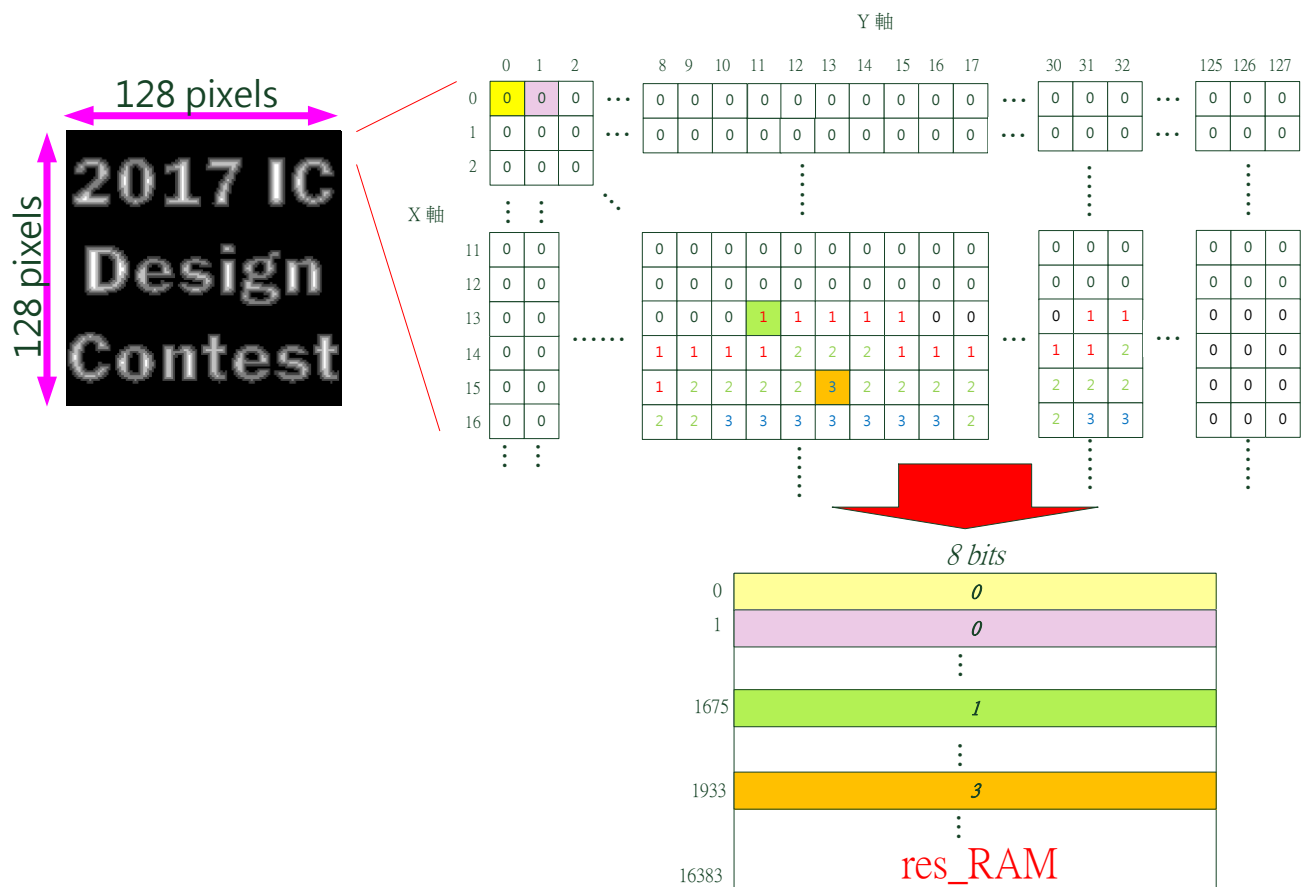


圖 5.輸出結果圖像記憶體位址方式對應及運算結果

輸出結果圖像記憶體的時序如下圖所示。res\_RAM 會在每個負緣時脈訊號時，若 res\_rd 訊號為 High(如下圖 6.的 T4 及 T5 時間)，則會將 res\_addr 訊號所指定之位址的資料立即由 res\_di 匯流排送到 DT 端；如果 res\_rd 為 Low 則 res\_RAM 將不會進行任何動作。而當正緣時脈訊號時，若 res\_wr 訊號為 High，則此時在 res\_do 匯流排上的資料就會被寫入到 res\_addr 所指定的位址內(如下圖 6 的

T6 時間)。本記憶體不需考慮寫入及讀取延遲的問題。

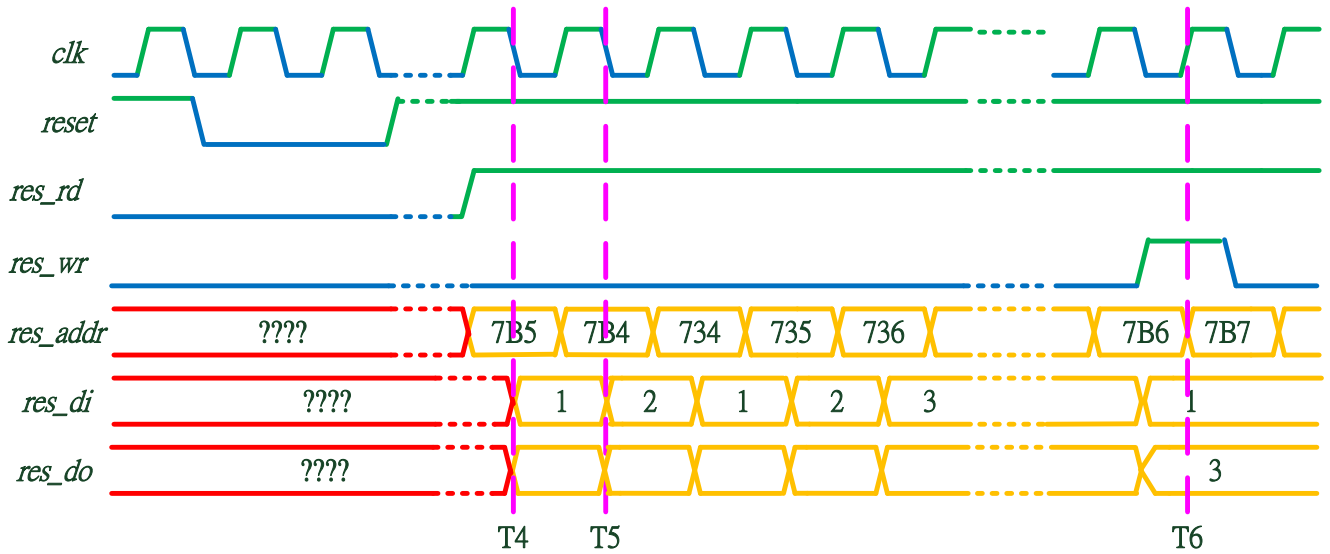


圖 6. 輸出結果圖像(res\_RAM)記憶體時序

## 2.4 系統時序規格圖

系統控制訊號時序規格圖，如下圖 7.所示。

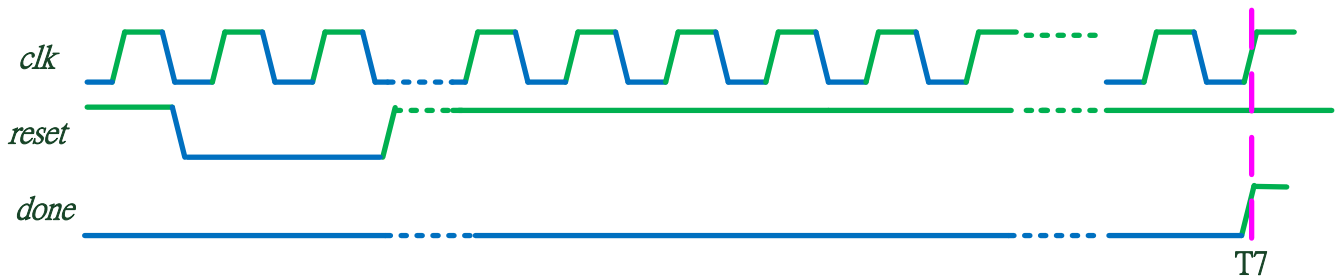


圖 2. 系統控制訊號時序

- 本電路的 reset 訊號為低位準啟動，reset 訊號持續兩個 Cycle 時間後結束。
- reset 訊號動作結束後，DT 電路需發送 sti\_ROM 記憶體相關控制訊號以讀取二值化圖像資料。請參考 2.3.1 節。
- 接著 DT 端須運算 distance transform 演算法將二值化圖像資料轉成灰階圖像資料，(請參考 2.3 節之描述)，當進行 distance transform 運算時，若有需要可自行對 res\_RAM 記憶體進行讀取及寫入動作。
- 完成後將結果資料寫到 res\_RAM。(請參考 2.3.2 節)。
- 將灰階圖像資料全數寫到 res\_RAM 後，DT 端須將 done 訊號設定為 High(如上圖 7.的 T7 時間)。Host 端就會開始進行驗證，驗證完成後整個模擬會立即結束。



### 3. 評分標準

主辦單位的評分人員將依照參賽者提供之系統時脈進行 RTL simulation 或 gate-level simulation，以驗證設計正確性，並且依據設計檔上傳至 CIC FTP 檔案伺服器(請參閱附錄 D)的時間來進行排名。各參賽隊伍應於參賽者定義的系統時脈下，確保輸出結果無設置與保持時間(setup/hold time)的問題，並完全符合主辦單位所提供的標準設計結果。

#### ☆ 評分項目一：依”模擬時間”(Time)長短評分

各參賽隊伍執行 RTL Simulation 及 Gate-level Post-synthesis Simulation 模擬完後，會出現模擬時間，評分人員會以此模擬時間如下面範例，紀錄成 **Time = 2491825 NS** 做評分。

```
-----  
Congratulations!!! All data have been generated successfully!  
----- The test result is ..... PASS -----  
  
-----  
Simulation complete via $finish(1) at time 2491825052 PS + 0  
./testfixture.v:173      #(`CYCLE/3); $finish;  
ncsim> exit
```

#### ☆ 評分項目二：依”面積”(Area)大小評分

各參賽隊伍完成電路合成後，Cell Area 可利用以下指令產生而得知

以 DC 產生 QoR report 的指令： report\_qor > DT.qor

以 RC 產生 QoR report 的指令： report qor > DT.qor

評分人員會以此 DT.qor 如下面範例，紀錄成 **Area = 12106** (以小數點後做四捨五入)做評分。

```
Area  
-----  
Combinational Area:      7475.349565  
Noncombinational Area:  4630.507271  
Buf/Inv Area:            1037.111386  
Total Buffer Area:       220.66  
Total Inverter Area:     816.45  
Macro/Black Box Area:   0.000000  
Net Area:                113820.575806  
Cell Area:               12105.856836  
Design Area:            125926.432642
```



#### ☆ 評分方法及錄取標準

評分方式會依設計完成程度，分成 A、B、C、D 三種等級，排名順序為 **A > B > C > D**。主辦單位將根據設計內容的完成度給予記分。審查成績將另擇期通知。

1. **A 等級**：RTL 與 Gate-level simulation 結果完全正確，且 Area 小於 **12500**，且 Time 小於 **1,920,000 ns**。

註: Time 以 **Gate-level simulation** 時間計算。

註: 本等級中，**最後上傳時間越早者**為優先錄取。

2. **B 等級**：RTL 與 Gate-level simulation 結果完全正確，且 Area 小於 **12500**，而 Time 在 **1,920,000** 與 **2,880,000** 之間。

註: Time 以 **Gate-level simulation** 時間計算。

註: 本等級中，**最後上傳時間越早者**為優先錄取。

3. **C 等級**：RTL 與 Gate-level simulation 結果完全正確，但 Area 大於 **12500**，而 Score 小於 **48,000,000,000**。此等級之成績計算方式如下：

$$\text{Score} = \text{Time} \times \text{Area}$$

註: Time 以 **Gate-level simulation** 時間計算。

註: 本等級中，**Score 越小者**為優先錄取!

4. **D 等級**：RTL simulation 結果完全正確。在 clock cycle 設定為 10ns 的情況下，此等級之成績計算方式如下：

$$\text{Score} = \text{Time}$$

註: 本等級中，**Score 越小者**為優先錄取!

## 附錄

在附錄 A 中說明本次競賽之軟體環境；附錄 B 為主辦單位所提供各參賽者的設計檔說明；附錄 C 為評分用檔案，亦即參賽者必須回傳至 CIC 的檔案資料；附錄 D 則為設計檔上傳步驟說明。

### 附錄 A 軟體環境

競賽所提供的設計軟體與版本如下表二。驗證評分時，係以所列軟體及版本作為驗證依據。

表二、設計軟體版本

Functionality	Corresponding EDA tools
Logic Simulator	nc-verilog 2014.10.005 modelsim 10.5c vcs-mx 2016.06-sp1-1
Logic Synthesizer	design-compiler 2016.12

## 附錄 B 設計檔案說明

### 1. 下表三.為主辦單位所提供各參賽者的設計檔案

表三、設計檔

檔名	說明
testfixture.v	測試樣本檔。此測試樣本檔定義了時脈週期與測試樣本之輸入及預期輸出信號。
DT.v ( DT.vhd)	參賽者所使用的設計檔，已包含系統輸/出入埠之宣告
./dat/*_sti.dat	測試樣本檔案
./dat/*_bcexp.dat	比對樣本檔案
report.000	結果報告範本
DT.sdc	Design Compiler 電路合成規範檔。 本規範檔除了 cycle 可修改外，其餘 constraints 皆不可修改。
tsmc13_neg.v	Gate-level simulation 所需要之 cell library file
synopsys_dc.setup	Design Compiler 初始設定範例檔案

### 2. 請使用 DT.v (.vhd)，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：

```

module DT(
    input          clk,
    input          reset,
    output reg     done ,
    output reg     sti_rd ,
    output reg     [9:0] sti_addr ,
    input          [15:0] sti_di,
    output reg     res_wr ,
    output reg     res_rd ,
    output reg     [13:0] res_addr ,
    output reg     [7:0]  res_do,
    input          [7:0]  res_di
);

endmodule

```

3. 比賽共提供二組測試樣本，參賽者可依下面範例來進行模擬：

**主辦單位於評分時，將可能額外再使用其他組測試樣本進行評分工作。**

為方便參賽者偵錯，本題提供測試樣本一(TB1)的部分結果於附錄 F，請參賽者自行參考。

RTL Simulation 時使用指令如下：

- 使用 ncverilog 模擬指令範例如下：

**ncverilog testfixture.v DT.v +define+TB1**

- 使用 modelsim 模擬，則是在 compiler verilog 時，使用下面指令：

**vlog testfixture.v +define+TB1**

Gate-Level Simulation 時使用指令如下：

- 使用 ncverilog 模擬指令範例如下：

**ncverilog testfixture.v DT\_syn.v -v tsmc13\_neg.v +define+SDF +define+TB1**  
define 中加上 SDF 可讓測試程式引入 gate level netlist 的 sdf 檔案資訊。

- 上述指令中 **+define+TB1** 指的是當進行第一組測試樣本的模擬，當使用第二組測試樣本請自行修改此參數為 **+define+TB2**。

- **NC-Verilog** 使用者若想要輸出 **FSDB** 格式波形檔案(請先 source verdi 的環境設定檔)，可自行再加入 **+define+FSDB +access+r**  
**modelsim** 使用者，請直接使用內建波形來進行除錯。

## 附錄 C 評分用檔案

評分所需檔案可分為三部份：(1)RTL design，即各參賽隊伍對該次競賽設計的 RTL code，若設計採模組化而有多個設計檔，請務必將使用到的各 module 檔寫進 DT.v，以免評審進行評分時，無法進行編譯；(2)gate-level design，即由合成軟體所產生的 gate-level netlist，以及對應的 SDF 檔；(3)report file，參賽隊伍必須依照自己的設計內容，撰寫 report.000 檔，以方便主辦單位進行評分，report.000 的格式如圖八所示。(report 檔以後三碼序號表示版本，若繳交檔案更新版本，則新版的 report 檔的檔名為 report.001，依此類推)

表六、評分用檔案

RTL category		
Design Stage	File	Description
N/A	report.xxx	design report
RTL Simulation	*.v or *.vhd	Verilog (or VHDL) synthesizable RTL code
Gate-Level category		
Design Stage	File	Description
Pre-layout Gate-level Simulation	*_syn.v	Verilog gate-level netlist generated by Synopsys Design Compiler
	*_syn.sdf	SDF timing information generated by Synopsys Design Compiler
	*_syn.ddc	design database generated by Synopsys Design Compiler

FTP 帳號(FTP account): 999999

通過 gate-level simulation 之 cell area report : 9000

通過 gate-level simulation 之 clock cycle time (ns) : 10

--- RTL category---

使用之 HDL 模擬器名稱(HDL simulator ): nc-verilog

RTL 檔案名稱(RTL filename): DT.v 以及使用到的子模組檔案...

--- Pre-layout gate-level ---

gate-level 檔案名稱(gate\_level filename): DT\_syn.v

gate-level sdf filename: DT\_syn.sdf

design compiler 合成資料庫(dc library): DT\_syn.ddc

-----注意事項(annotation)-----

(其餘注意事項依各參賽隊伍的需求填寫)

圖八、report.000 的範本

## 附錄 D 檔案上傳

所有包含於如附錄 C 中表格所示的檔案，均需要提交至 CIC。並且，提交的設計檔案，需要經過壓縮於同一個資料夾下，步驟如下：

1. 建立一個 result\_xxx 資料夾。其中“xxx”表示繳交版本。例如“000”表示為第一次上傳；“001”表示為第二度上傳；002 表示為第三度上傳，以此類推…。
2. 參考附錄 C 評分用檔案，將所有繳交檔案複製到 result\_xxx 資料夾
3. 執行 tar 指令將 result\_xxx 資料夾包裝起來，tar 的指令範例如下：  
tar cvf result\_xxx.tar result\_xxx  
其中 xxx 表示繳交版本  
執行完後應該會得到 result\_xxx.tar 的檔案
4. 使用 ftp 將 result\_xxx.tar 及 report.xxx 一併上傳至 CIC 提供的 ftp server，result\_xxx.tar 與 report.xxx 之“xxx”編號需一致，評審將以最後上傳的設計檔及報告檔編號進行評分作業。

本題限制上傳之設計檔僅可使用 tar 或 zip 壓縮格式，使用 rar 或其他格式者一律不予計分。

請注意!!上傳之 FTP 需切換為二進制模式(binary mode)，且傳輸埠均設為 21(port:21)。

ftp 的帳號和密碼在賽前已用 email 寄給各參賽者。若有任何問題，請聯絡 CIC

FTP site1 (新竹晶片中心)：iccftp.cic.org.tw (140.126.24.18)

FTP site2 (南區晶片中心)：iccftp2.cic.org.tw(140.110.117.9)

5. 若你需要繳交更新版本，請重覆以上步驟，並記得修改 report 檔及 tar 檔的版本編號，因為你無法修改或刪除或覆蓋之前上傳的資料。

附錄 E Distance Transform範例說明

✓ 範例二值圖像如下：



✓ 其內容為如下：

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1	1	0	0	0	0	0	1	1	1	1	1	0	0
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

✓ 經Forward pass運算後結果如下：

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1	0	0	0	0	0	0	1	1	1	1	1	0	0
0	1	2	2	2	2	2	2	1	1	1	1	1	1	2	2	2	2	1	0	0
0	0	0	0	1	2	2	2	2	2	2	2	2	2	2	2	3	2	1	1	0
0	0	0	0	1	2	3	3	3	3	3	3	3	3	3	3	3	2	2	2	1
0	0	0	0	1	2	3	4	4	4	4	4	4	4	4	4	3	3	3	2	1
0	1	1	1	1	2	3	4	5	5	5	5	5	5	5	4	4	4	3	0	0
0	1	2	2	2	2	3	4	5	6	6	6	6	5	5	5	4	1	0	0	0
0	0	0	0	0	0	0	0	1	2	3	4	5	6	6	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	2	3	4	5	6	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	2	3	4	5	2	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

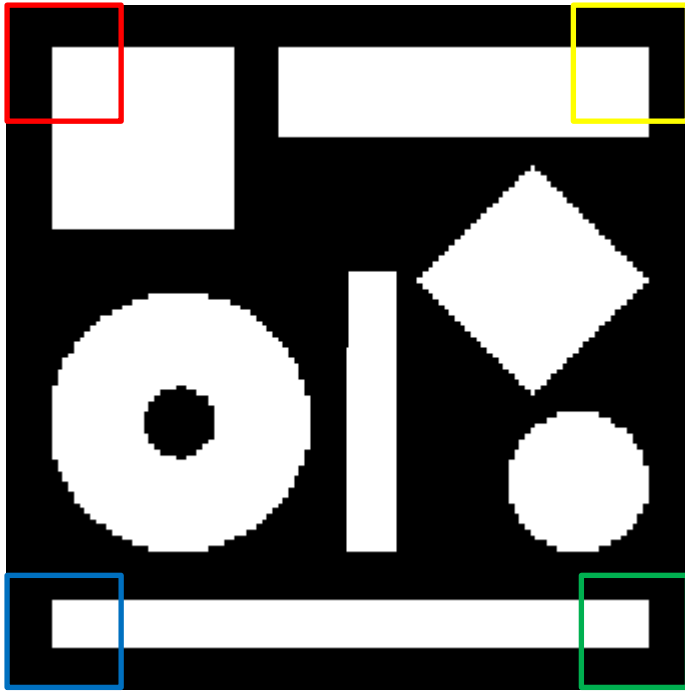
✓ 再經Backward pass運算後的最終結果如下所示：

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1	0	0	0	0	0	1	1	1	1	1	0	0	0
0	1	1	1	1	2	2	2	1	1	1	1	1	2	2	2	2	1	0	0	0
0	0	0	0	1	2	2	2	2	2	2	2	2	2	2	3	2	1	1	1	0
0	0	0	0	1	2	3	3	3	3	3	3	3	3	3	3	2	2	2	1	0
0	0	0	0	1	2	3	3	3	3	3	4	3	3	3	3	3	2	1	1	0
0	1	1	1	1	2	2	2	2	3	4	3	2	2	2	2	2	1	0	0	0
0	1	1	1	1	1	1	1	2	3	4	3	2	1	1	1	1	1	0	0	0
0	0	0	0	0	0	0	0	1	2	3	3	2	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	2	2	2	2	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



## 附錄 F 測試樣本一(TB1)部分結果

✓ 測試樣本一的二值圖像如下：



✓ 二值圖像內容如下：

左上角紅色區域如下所示

[illegible]

右上角黃色區域如下所示

[illegible]

左下角藍色區域如下所示

[illegible]

右下角綠色區域如下所示

[illegible]

✓ 經Forward pass運算後，再經Backward pass運算後的最終結果如下所示：

左上角紅色區域如下所示

[illegible]

右上角黃色區域如下所示

[illegible]

左下角藍色區域如下所示

[illegible]

右下角綠色區域如下所示

[illegible]