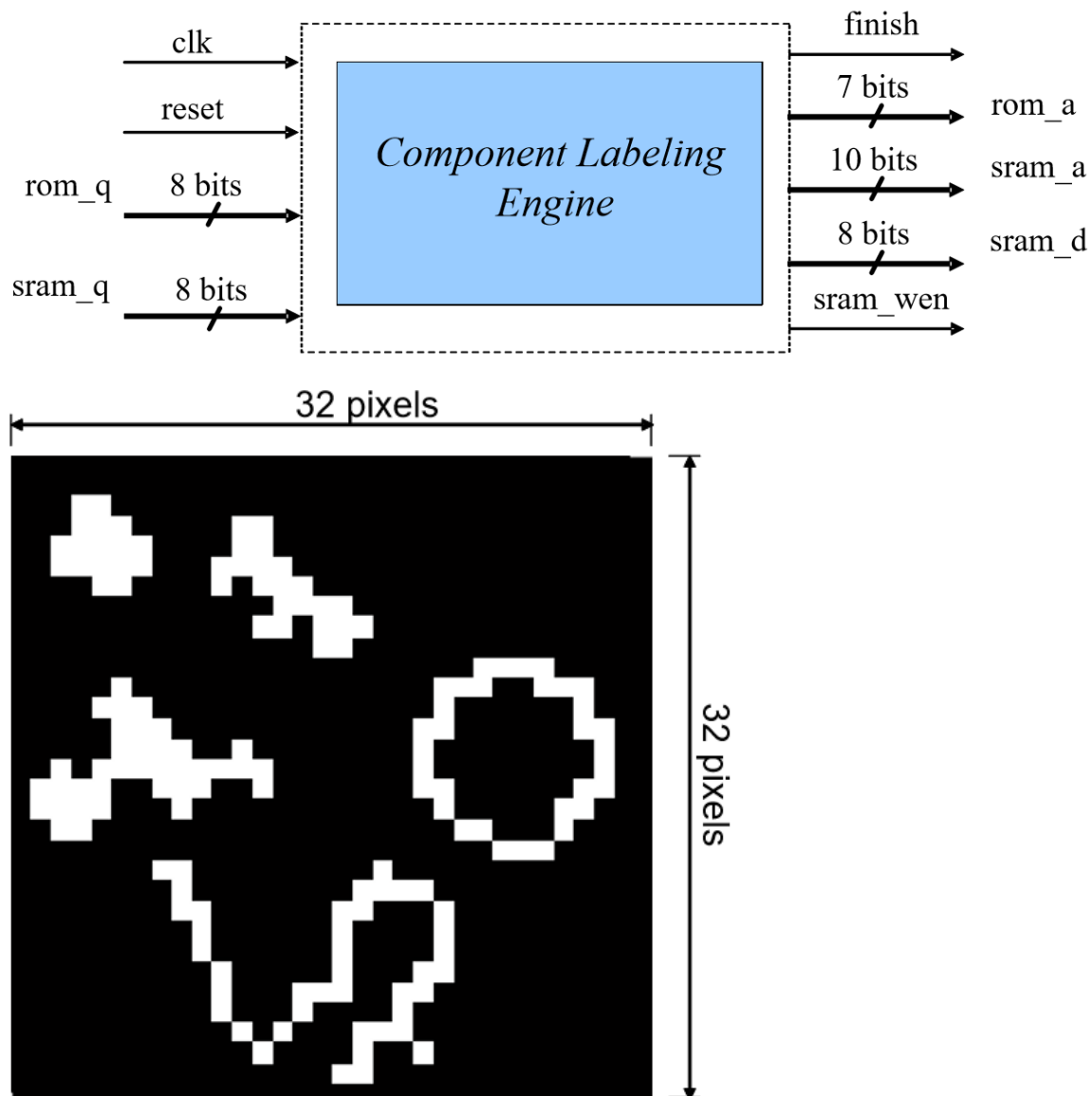


Integrated Circuit Design Final Project

Component Labeling Engine

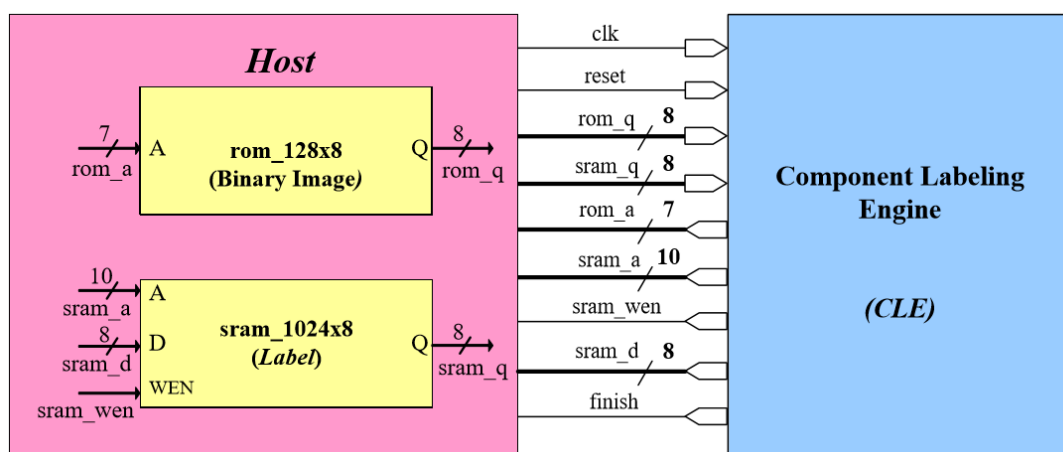
1. 問題描述

請完成一 Component Labeling Engine(後文以 **CLE** 表示)的電路設計。此電路可將任意 32x32 大小之二元影像(Binary Image)訊號，尋找出該圖片所有前景物件，然後在同一物件上給予相同的編號，完成所有物件的編號後將其儲存於 SRAM 記憶體，即完成 CLE 電路功能。有關 CLE 詳細規格將描述於後。



2. 設計規格

2.1 系統方塊圖



2.2.1 輸出入訊號和記憶體描述

Name	I/O	Width	Simple Description
clk	I	1	本系統同步於時脈正緣。
reset	I	1	高位準非同步(active high asynchronous)之系統重置信號。
rom_a	O	7	CLE 至 ROM 讀取資料時用到的 Address Bus。CLE 透過 rom_a 傳遞傳送位址至 ROM，直接與 ROM 的腳位 A 相連。ROM 的詳細規格說明，可參考 2.2.2。
rom_q	I	8	CLE 至 ROM 讀取資料時用到的 Data Bus。CLE 透過 rom_q 從 ROM 讀取資料，直接與 ROM 的腳位 Q 相連。
sram_a	I	10	CLE 至 SRAM 存取資料時用到的 Address Bus。CLE 透過 sram_a 傳遞傳送位址至 SRAM，直接與 SRAM 的腳位 A 相連。有關 SRAM 的詳細規格說明，可參考 2.2.3。
sram_d	O	8	CLE 至 SRAM 寫入資料時用到的 Data Bus。CLE 透過 sram_d 寫入資料到 SRAM，直接與 SRAM 的腳位 D 相連。
sram_q	O	8	CLE 至 SRAM 讀取資料時用到的 Data Bus。CLE 透過 sram_q 讀取資料，直接與 SRAM 的腳位 Q 相連。
sram_wen	O	1	CLE 對 SRAM 作 Read/Write 的控制訊號。當該訊號為 Low，表示 CLE 要對 SRAM 作寫入，反之，當訊號為 High，表示 CLE 要對 SRAM 作讀取。該訊號直接與 SRAM 的腳位 WEN 相連。
finish	O	1	當為 Low 時，表示還在運算，Host 端不作檢查；反之，當為 High，表示 CLE 電路已運算完畢，Host 端可以開始檢驗 CLE 電路運算結果是否正確。

註:本題的 ROM CEN 訊號，在 Host 端已設為 1'b0。也就是 ROM 永遠處於啟動狀態，在正緣給予 address，則在下一個負緣得到值。

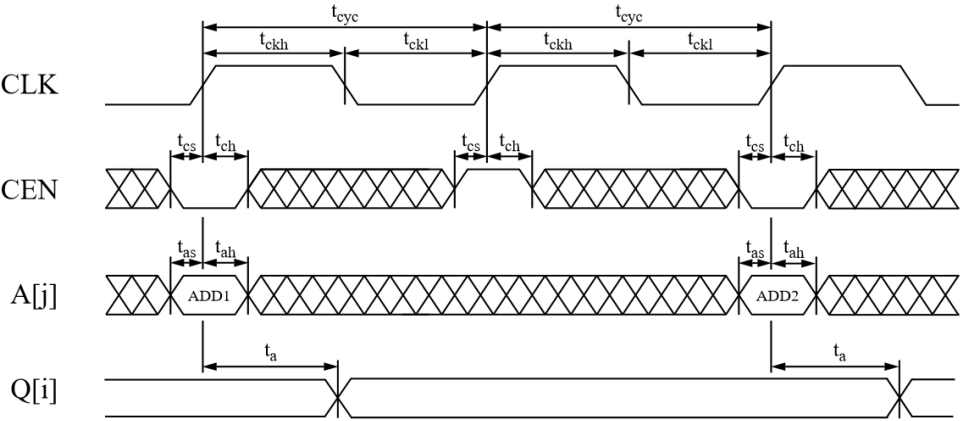
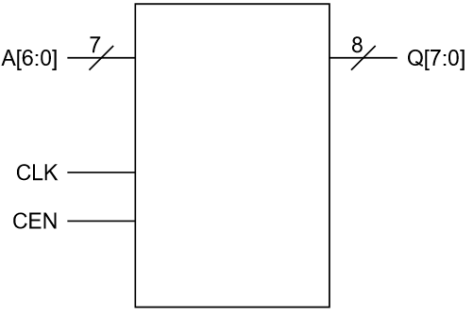
註:本題的 SRAM CEN 訊號，在 Host 端已設為 1'b0。也就是 SRAM 永遠處於啟動狀態。

2.2.2 ROM 詳細規格

Symbol

Pin Description

Pin	Description
A[6:0]	Addresses (A[0] = LSB)
CLK	Clock Input
CEN	Chip Enable
Q[7:0]	Data Outputs (Q[0] = LSB)

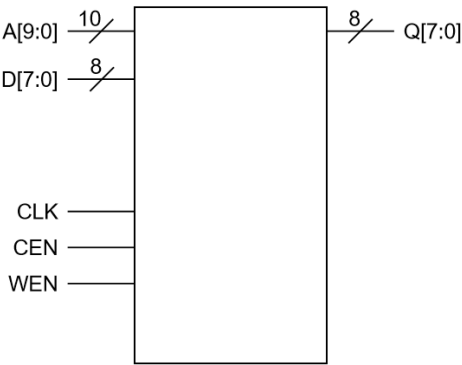


2.2.3 SRAM 詳細規格

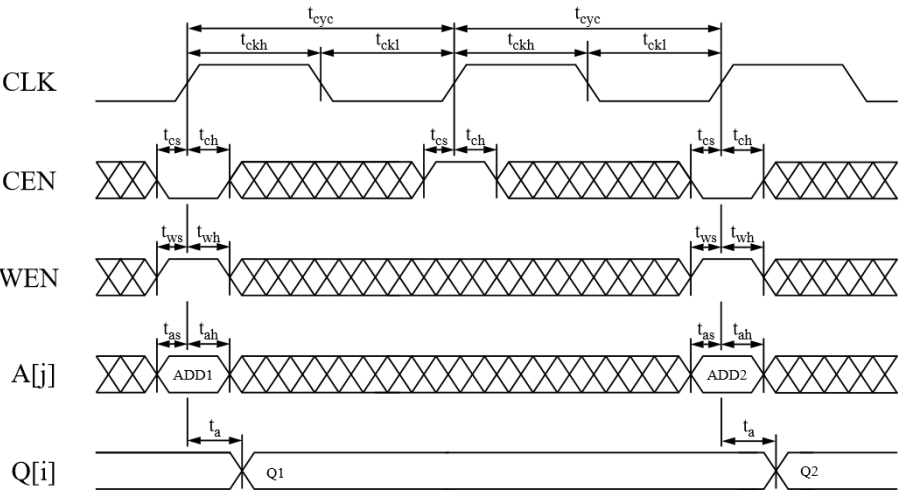
Pin Description

Pin	Description
A[9:0]	Addresses (A[0] = LSB)
D[7:0]	Data Inputs (D[0] = LSB)
CLK	Clock Input
CEN	Chip Enable
WEN	Write Enable
Q[7:0]	Data Outputs (Q[0] = LSB)

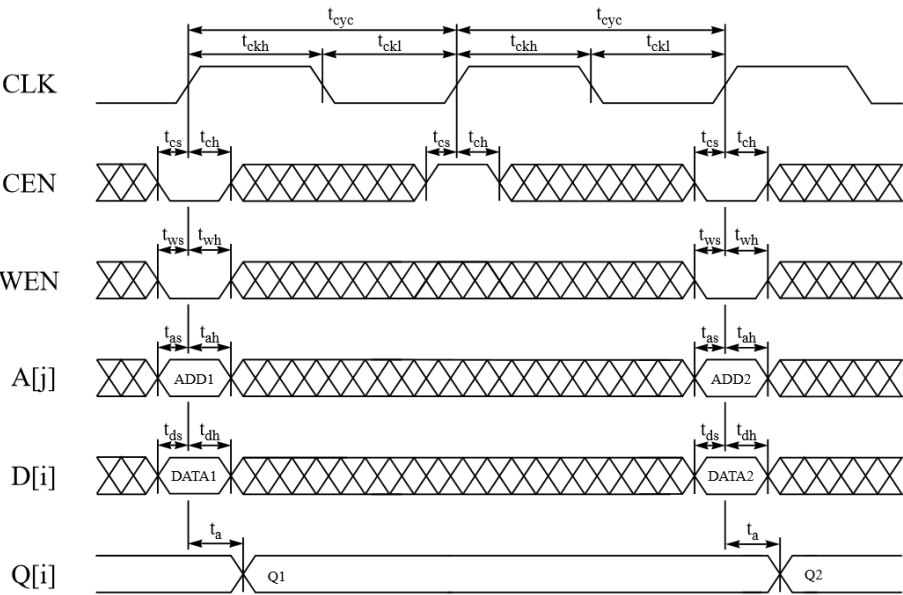
Symbol



Synchronous Single-Port SRAM Read-Cycle Timing



Synchronous Single-Port SRAM Write-Cycle Timing



2.3 系統功能描述

一張 32x32 大小的二元影像訊號，每個 Pixel 只有 0 或 1 兩種訊號，0 表示背景(黑色區域)，1 表示前景物件(白色區域)。

CLE 電路可從中尋找出所有前景物件(訊號 1 所在處)，然後判斷這些訊號 1 的 Pixel 是否有連接在一起，有連接在一起的 Pixels，將被視為同一物件，同一物件的所有 Pixels 都要被編上一組相同的編號(編號可自訂)，該編號一旦使用後，便不可再重覆使用於其他物件上，當完成所有物件的編號後，將其結果儲存於 SRAM 記憶體，即完成 CLE 電路功能。

2.3.1 CLE 電路的輸入

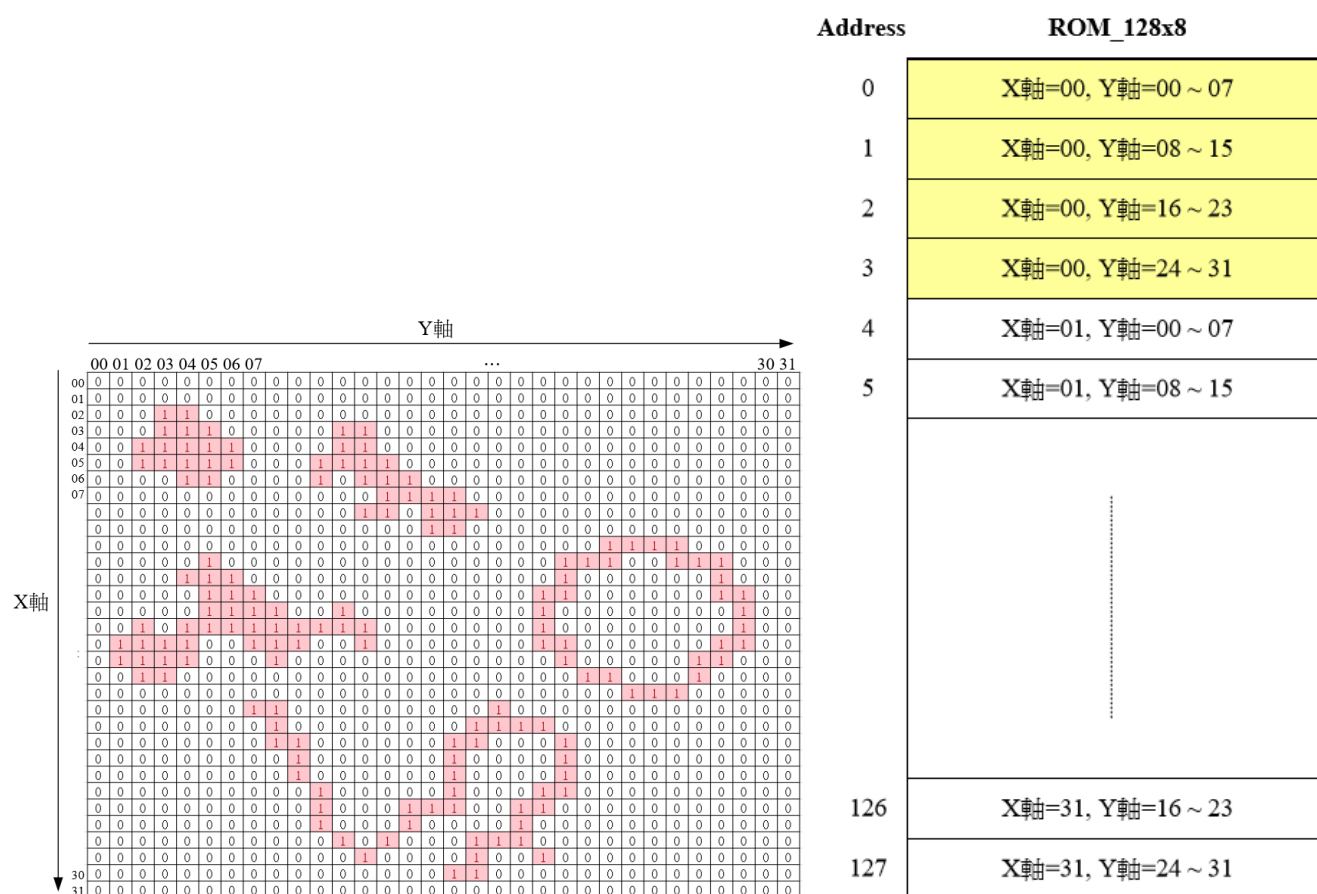
如下左圖為二元影像之實際訊號值，該訊號值已存於 128x8 的 ROM 裡，儲存方式如下右圖所示。

ROM 位址 0，儲存【X 軸座標 00、Y 軸座標 00~07】，共 8bits 資料，該筆資料的 MSB 為【X 軸座標 00、Y 軸座標 00】，LSB 為【X 軸座標 00、Y 軸座標 07】。

ROM 位址 1，儲存【X 軸座標 00、Y 軸座標 08~15】，

...

ROM 位址 127，儲存【X 軸座標 31、Y 軸座標 24~31】。



2.3.2 CLE 電路的運算

Figure 1 shows the eight possible configurations of the 3x3 grid. Each configuration is a 3x3 matrix of cells, each containing a number from 0 to 9. The configurations are arranged in two rows of four. The first row shows configurations where the top row is [1, 0, 0], [0, 1, 0], [0, 0, 1], [0, 0, 0], and the second row shows configurations where the top row is [0, 0, 0], [0, 0, 0], [0, 0, 0], [0, 0, 0]. The configurations are labeled 1 through 8.

CLE 電路運算方式如下：

- A. 尋找 Pixel 訊號值為 1 的點，即找到前景物件所在處，接著自行判斷該 Pixel 於九宮格的範圍內是否有跟其他 Pixel 相連在一起，上圖這八種可能的情形都稱之為有相連，相連在一起的 Pixel 將被視為同一物件，反之，不相連的 Pixels，將被視為不同物件。

- B. 同一物件中的每個 Pixel 都要編上相同的編號，該編號可自定。

註 1：可用編號範圍：8'h01 ~ 8'hFB。

註 2：禁用編號範圍：8'hFC ~ 8'hFF。(這些編號已作為特殊用途，請勿使用!)

註 3：背景編號數值：8'h00。(前景物件請勿使用此編號)

- C. 已使用過的編號不可重複再使用，因此不同物件的編號一定不相同。

2.3.3 CLE 電路的輸出

本範例最後編號完後結果如下左圖所示，編號數值僅供參考。因僅用了五個編號，可判斷出該圖片總共有五個物件。

接著將包含 A. 背景訊號：編號固定為 8'h00，B. 前景各個物件編號，皆需儲存於 1024x8 的 SRAM 中，儲存方式如下右圖所示。

SRAM 位址 0，儲存【X 軸座標 00、Y 軸座標 00】的 Pixel 資料，由於可用編號範圍含背景為 8'h00 ~ 8'hFB，因此 SRAM 一個 Address 剛好存放一個 Pixel 編號後結果，剛好是 8bits。

SRAM 位址 1，儲存【X 軸座標 00、Y 軸座標 01】的 Pixel 資料，…，依此類推。

確定所有 Pixels 編號後的數值都已填在 SRAM 後，此時將“finish”訊號拉為 High，以告知 Host 端開始檢驗 SRAM 編號完結果，即完成 CLE 電路功能。

[illegible]

Address	SRAM_1024x8
0	X軸=00, Y軸=00
1	X軸=00, Y軸=01
2	X軸=00, Y軸=02
	⋮
31	X軸=00, Y軸=31
32	X軸=01, Y軸=00
33	X軸=01, Y軸=01
	⋮
992	X軸=31, Y軸=00
	⋮
1022	X軸=31, Y軸=30
1023	X軸=31, Y軸=31

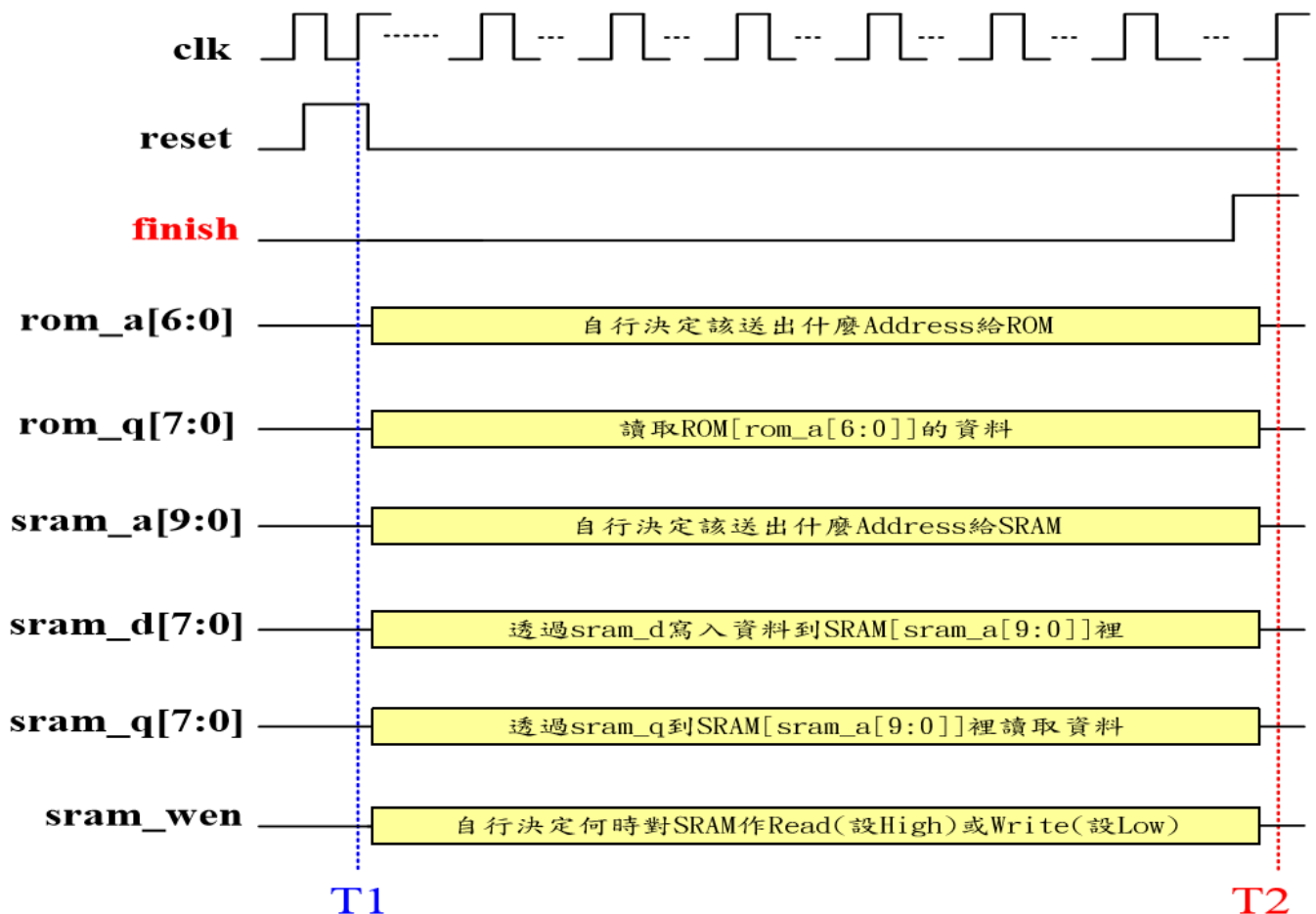
A. 先將 SRAM 編號後的資料，以 Log Message 顯示。

B. 再將 **SRAM** 資料與標準解答作比對，比對結果如果正確，**Log Message** 的畫面會如上圖，但假若比對結果有不同處，會用以下特殊符號表示，如下表所示。假設 **SRAM** 全部編號 **8'h00**，其顯示出訊息如下圖所示，結果顯示：背景所有 **Pixels** 全對，以 **00** 顯示，前景物件全錯以小寫 **xx** 顯示，最後一個 **Pixel** 是 **Unknown**，因隸屬於背景訊號區域，故 **Log Message** 以大寫 **UU** 來表示錯誤。

特殊符號	解說
XX	該 Pixel 是屬於背景訊號，但參賽者卻對此 Pixel 編號錯誤
UU	該 Pixel 是屬於背景訊號，但參賽者卻對此 Pixel 編號為 Unknow 訊號
xx	該 Pixel 是屬於前景物件，但參賽者卻對此 Pixel 編號錯誤
uu	該 Pixel 是屬於前景物件，但參賽者卻對此 Pixel 編號為 Unknow 訊號

[illegible]

2.4 CLE 電路時序規格



1. CLE 電路初始化，Reset 一個 Cycle 的時間。
2. T1 時間點，reset 後可隨即送出 ROM Address 開始讀取二元影像訊號。
二元影像訊號要讀取哪個位址、同一個 Pixel 要重覆讀取幾次與電路規畫有關，可自行決定。
讀取二元影像訊號後，要開始怎麼針對物件作編號處理，SRAM 位址要怎麼送、怎麼讀、怎麼寫，也與電路規劃與自行想出的演算法有關，在此皆無硬性規定。
3. 當整張圖片之前景物件編號完成及背景訊號 8'h00 皆寫入 SRAM 後，T2 時間點將 finish 訊號拉為 High，即完成 CLE 電路運作！此時 Host 端會開始自動檢驗 SRAM 裡的資料其正確性，檢驗結果會以 Log Message 秀出。

3. 檔案說明

檔名	說明
source.cshrc	source source.cshrc 即可 source 所有要用的 library
RTL 相關	
CLE.v	已經定義了輸入輸出，將描述電路的 verilog code 寫在此檔案裡。
testfixture_a.v testfixture_b.v testfixture_c.v	本題共計三個 TestBench，每組測試皆為 32x32 pixels 之影像。
run_RTL_a.v run_RTL_b.v run_RTL_c.v	三組的 script 都要執行。直接執行 ncverilog -f run_RTL_x.f 即可。
rom_128x8_x.v rom_128x8_verilog_x.rcf rom_x.dat	CLE 電路會用到的三組 32x32 pixels 之影像 註：檔案已被 include 到 TestBench。
sram_1024x8.v sram_x.dat	圖片的 Golden 結果檔案。註：檔案已被加入到 TestBench。
synthesis 相關	
syn/.synopsys_dc.setup	使用 Design Compiler 作合成或 IC Compiler Layout 之初始化設定檔。請依 Library 實際擺放位置，自行修改 Search Path 的設定。
CLE.v	已完成的 RTL code
syn/CLE_DC.sdc	Design Compiler 作合成之 Constraint 檔案。可依需求自行修改 cycle 的設定。注意：環境相關參數 請勿更改。
syn/run.tcl	在 syn 資料夾內執行 dc_shell -f run.tcl 即可完成 synthesis
Gate level 相關	
testfixture_a.v testfixture_b.v testfixture_c.v	同 RTL 用的 testbench，要根據合成時的 cycle time 去修改裡面的參數，預設是 define CYCLE 100。
CLE_syn.v	合成時產生的 gate level netlist
tsmc13_neg.v	各個 gate 製程的相關資訊
syn/CLE_syn.sdf	合成時產生的 standard delay format 檔案，描述了電路中每個元件的延遲時間。
run_DC_a.f run_DC_b.f run_DC_c.f	模擬時，直接執行 ncverilog -f run_DC_x.f 即可。

4. 模擬方法

本題有三組 Pattern 要測試：testfixture_a.v、testfixture_b.v、testfixture_c.v，這些檔案會用到之相關檔案已設定完成，只要注意 rom_128x8_a.v、rom_128x8_b.v、rom_128x8_c.v、rom_128x8_verilog_a.rcf、rom_128x8_verilog_b.rcf、rom_128x8_verilog_c.rcf、sram_1024x8.v、sram_a.dat、sram_b.dat、sram_c.dat 等十個檔案，有擺放在目前目錄即可！

A. RTL

分別輸入

```
ncverilog -f run_RTL_a.f
```

```
ncverilog -f run_RTL_b.f
```

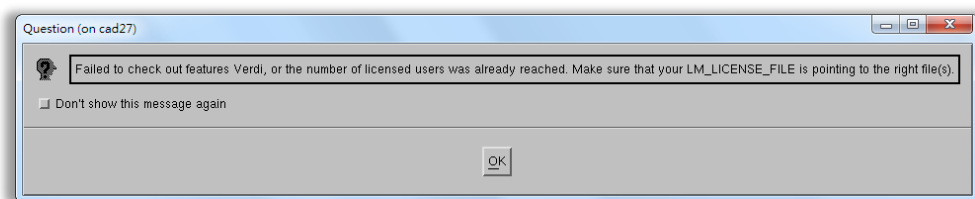
```
ncverilog -f run_RTL_c.f
```

即可跑完三筆測資

B. nWave

本次作業 ncverilog 產生的波形檔為.vcd，source source.cshrc 後即可使用 nWave 開啟。

To avoid the Verdi warning window occurs,



please type the following command:

```
setenv LM_LICENSE_FILE '26585@lsntu:26585@lsncku'
```

C. Gate level

分別輸入

```
ncverilog -f run_DC_a.f
```

```
ncverilog -f run_DC_b.f
```

```
ncverilog -f run_DC_c.f
```

即可跑完三筆測資

5. 評分標準

A. RTL 部分

Report 中請將 ncverilog 模擬畫面截圖，若成功則會出現以下畫面。

[illegible]

B. Gate level 部分

Report 中請大家擷取 design compiler 的 area report 與 timing report。

除此，要把 gate level netlist 再跑一次 ncverilog 模擬並截圖。(記得 test bench 中的 cycle time 要改成與合成時的 cycle time 一樣再跑模擬。)

範例如下：

Cell area = 504794 μm^2

Cycle time = 100ns

Total time = 307901.1ns

Cell area \times total time = 155,426,627,873.7 ($\mu\text{m}^2 \cdot \text{ns}$)

```
Total cell area:      504794.865029
Total area:           5323058.772256
```

Point	Incr	Path
-----	-----	-----
clock clk (rise edge)	0.00	0.00
clock network delay (ideal)	0.50	0.50
counter_reg[0]/CK (DFFRX1)	0.00	0.50 r
counter_reg[0]/Q (DFFRX1)	0.82	1.32 f
U32538/Y (CLKBUF3)	0.91	2.23 f
U31387/Y (CLKBUF3)	1.06	3.29 f
U28816/Y (CLKBUF3)	0.66	3.95 f
U15544/Y (CLKBUF3)	0.91	4.86 f
U32517/Y (MX4X1)	0.60	5.46 f
U32472/Y (MXI4X1)	0.49	5.96 r
U5483/Y (MXI2X1)	0.42	6.38 f
U31313/Y (NOR4X1)	0.40	6.78 r
U6907/Y (NAND2X1)	0.56	7.34 f
U5688/Y (NOR2X1)	0.82	8.15 r
U15542/Y (INVX3)	0.54	8.69 f
U10609/Y (OAI2BB1X2)	0.54	9.23 f
U31353/Y (OAI22XL)	0.59	9.82 r
U31352/Y (AOI211X1)	0.22	10.04 f
U31319/Y (AOI2BB2X1)	0.36	10.40 f
U31318/Y (OAI221XL)	0.29	10.69 r
sram_d_reg[2]/D (DFFNSRX1)	0.00	10.69 r
data arrival time		10.69
clock clk (fall edge)	50.00	50.00
clock network delay (ideal)	0.50	50.50
clock uncertainty	-0.10	50.40
sram_d_reg[2]/CKN (DFFNSRX1)	0.00	50.40 f
library setup time	-0.05	50.35
data required time		50.35
-----	-----	-----
data required time		50.35
data arrival time		-10.69
-----	-----	-----
slack (MET)		39.66

```
-----
Simulation Summary
-----
Congratulations! All data have been generated successfully!
-----PASS-----
err= 0
Simulation complete via $finish(1) at time 307901199 PS + 0
```

C. 評分方式

共分兩部分

- a. Report: 佔 30%。詳細說明自己的演算法以及硬體設計。
- b. 效能(total time x cell area): 佔 70%
 - (a) A 等級: RTL 與 Gate-level simulation 結果完全正確
 - (b) B 等級: RTL 結果完全正確。
 - (c) C 等級: RTL 未完成。

※達 A 等級，根據 score 大小排名，得分分布在 40~70。

※達 B 等級，可得 30 分。

※C 等級者，根據完成度給分。

6. 繳交檔案

評分所需檔案可分為三部份：

(1) RTL design，若設計採模組化而有多個設計檔，請務必將合成所要用到的各 module 檔放進來，以免無法進行編譯。

(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。

Design stage	File	Descirption
N/A	Report.pdf	design report
RTL Simulation	*.v	Verilog synthesizable RTL code
Pre-layout Gate-level Simulation	*_syn.v	Verilog gate-level netlist generated by Synopsys Design Compiler
	*_syn.sdf	Pre-layout gate-level sdf