

2016 University/College IC Design Contest

Cell-Based IC Design Category for Graduate Level

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。

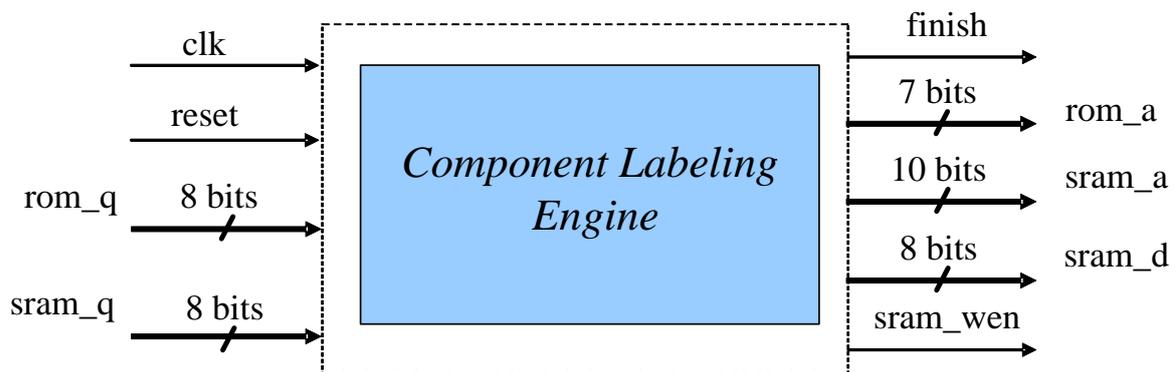
本電路各輸入輸出信號的功能說明，請參考表一。每個參賽隊伍必須根據下一節所給的設計規格及附錄 A 中的測試樣本完成設計驗證。

本次 IC 設計競賽比賽時間為上午 08:30 到下午 20:30。當 IC 設計競賽結束後，CIC 會根據第三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 E 中所列的要求，附上評分所需要的檔案。

本題目之測試樣本置於 `/usr/cad/icc2016/bgc/icc2016cb.tar`，請執行以下指令取得測試樣本：

```
tar xvf /usr/cad/icc2016/bgc/icc2016cb.tar
```

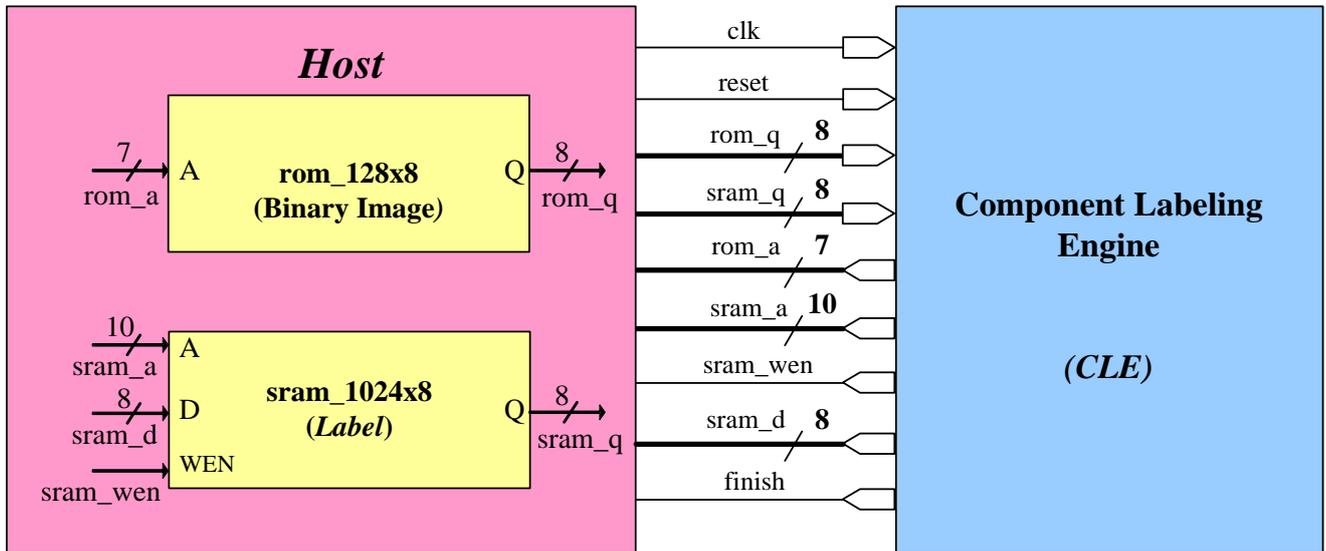
軟體環境及設計資料庫說明請參考附錄 F 與附錄 G。



圖一、Component Labeling Engine 之方塊圖

2.設計規格

2.1 系統方塊圖



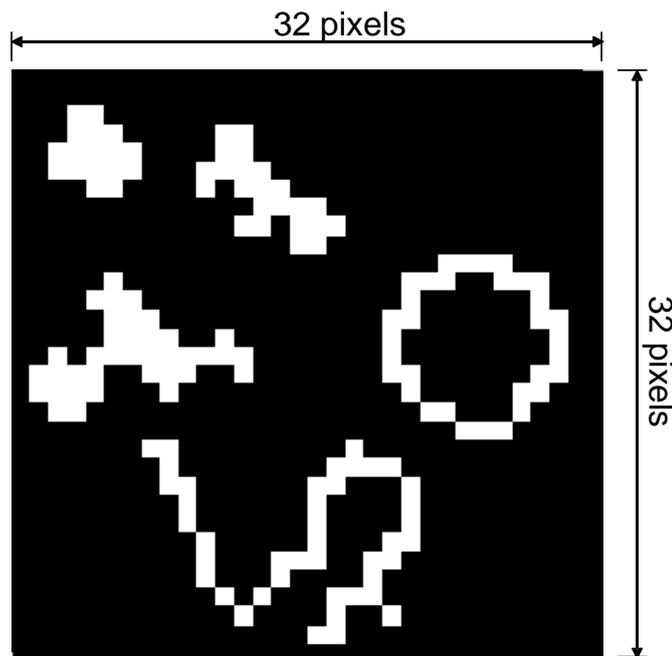
圖二、系統方塊圖

2.2 輸入/輸出介面

表 1 -輸入/輸出訊號

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。 (註: Host 端採 clk ”正”緣時送資料。)
reset	I	1	高位準非同步(active high asynchronous)之系統重置信號。
rom_a	O	7	CLE 電路至 ROM 讀取資料時會用到的 Address Bus。 CLE 傳送位址訊號至 ROM，就是透過 rom_a 傳遞，該訊號直接與 ROM 的 Address 腳位 A 相連。有關 ROM 的詳細規格說明，可參考 ROM 記憶體附件。 註：本題的 ROM CEN 訊號，在 Host 端已設為 1'b0。
rom_q	I	8	CLE 電路至 ROM 讀取資料時會用到的 Data Bus。 CLE 從 ROM 讀取資料，就是透過 rom_q 傳遞，該訊號直接與 ROM 的輸出訊號腳位 Q 相連。
sram_a	O	10	CLE 電路至 SRAM 存取資料時會用到的 Address Bus。 CLE 傳送位址訊號至 SRAM，就是透過 sram_a 傳遞，該訊號直接與 SRAM 的 Address 腳位 A 相連。有關 SRAM 的詳細規格說明，可參考 SRAM 記憶體附件。 註：本題的 SRAM CEN 訊號，在 Host 端已設為 1'b0

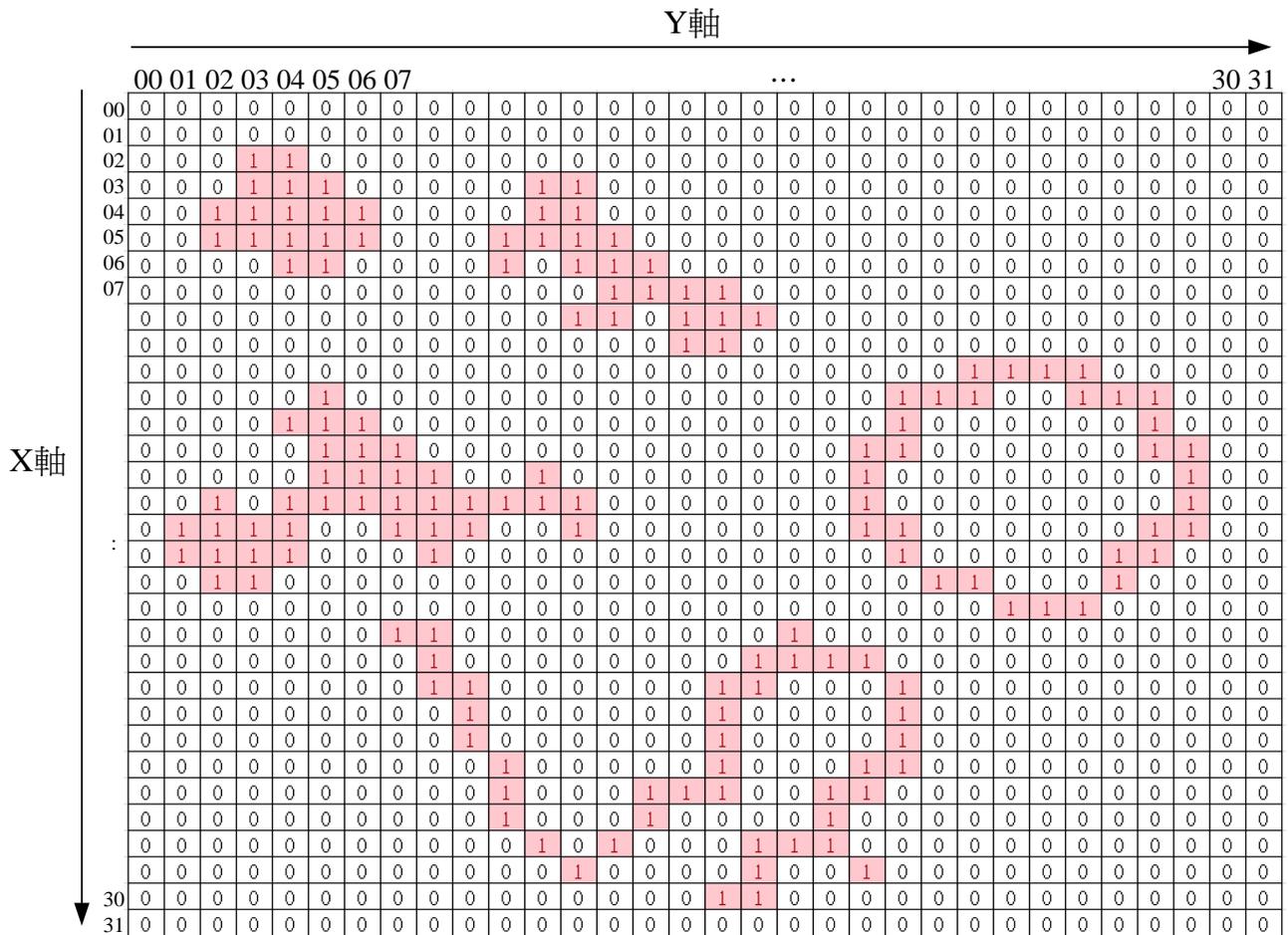
sram_d	O	8	CLE 電路至 SRAM 寫入資料時會用到的 Data Bus。CLE 若要寫入資料到 SRAM，就是透過 sram_d 傳遞，該訊號直接與 SRAM 的輸入訊號腳位 D 相連。
sram_q	I	8	CLE 電路至 SRAM 讀取資料時會用到的 Data Bus。CLE 若要從 SRAM 讀取資料，就是透過 sram_q 傳遞，該訊號直接與 SRAM 的輸出訊號腳位 Q 相連。
sram_wen	O	1	CLE 電路對 SRAM 作 Read/Write 的控制訊號。當該訊號為 Low，表示 CLE 要對 SRAM 作寫入，反之，當該訊號為 High，表示 CLE 要對 SRAM 作讀取。該訊號直接與 SRAM 的控制訊號腳位 WEN 相連。
finish	O	1	告知 Host 端，CLE 電路運算完畢，請 Host 端開始檢查 SRAM 的內容值是否運算正確。當為 Low 時，表示 CLE 電路還在運算，Host 端不作檢查；反之，當為 High 時，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。



圖三、二元影像訊號

2.3 系統描述

圖三為一張 32x32 大小的二元影像訊號，影像中的每一點訊號稱為 Pixel，每個 Pixel 只有 0 或 1 兩種訊號，0 表示背景(圖中的黑色區域)，1 表示前景物件(圖中的白色區域)。CLE 電路可處理固定影像 32x32 pixels 之圖片，從中尋找出該圖片所有前景物件(即訊號 1 所在處)，然後由參賽者自行判斷，這些訊號 1 的 Pixel 是否有連接在一起(要看到整張完整圖片為主)，有連接在一起的 Pixels，將被視為同一物件，需針對這些物件作處理，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。



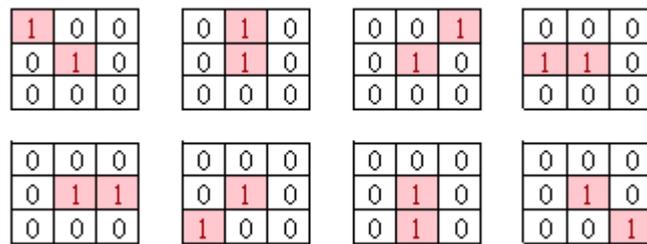
圖四、二元影像各 Pixel 之實際訊號值

Address	ROM_128x8
0	X軸=00, Y軸=00 ~ 07
1	X軸=00, Y軸=08 ~ 15
2	X軸=00, Y軸=16 ~ 23
3	X軸=00, Y軸=24 ~ 31
4	X軸=01, Y軸=00 ~ 07
5	X軸=01, Y軸=08 ~ 15
	⋮
126	X軸=31, Y軸=16 ~ 23
127	X軸=31, Y軸=24 ~ 31

圖五、二元影像訊號儲存於 ROM 的方式

2.3.1 CLE 電路的輸入：二元影像訊號已儲存於 ROM

圖四為圖三中二元影像之實際訊號值，該訊號值已存於 128x8 的 ROM 裡，儲存方式如圖五所示，ROM 位址 0，儲存圖五之【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，儲存圖五之【X 軸座標 00、Y 軸座標 08~15】，...，ROM 位址 127，儲存圖五之【X 軸座標 31、Y 軸座標 24~31】。參賽者可依電路需求自行至 ROM 讀取資料一次或多次。ROM 的 CEN 訊號，已在 Host 端設定為永遠開啟(即 CEN=1'b0)。



圖六、九宮格範圍內 Pixel 相連的八種情形

2.3.2 CLE 電路的運算方法

CLE 電路運算方式如下：

- A. 在圖四中尋找 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，像圖六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。

注意：參賽者判斷各 Pixel 是否是同一物件，最後要看到整張完整圖片為主。

- 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。

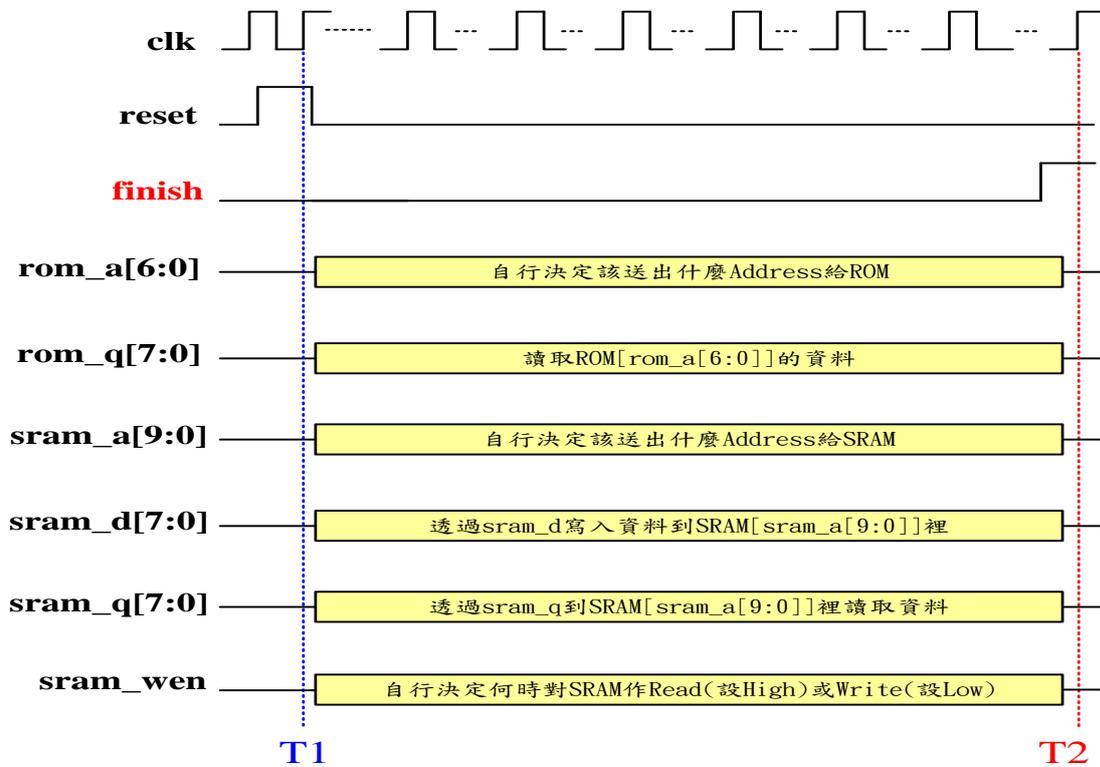
註 2：禁用編號範圍：8'hFC ~ 8'hFF。(這些編號已作為特殊用途，請勿使用!)

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。

2.4 CLE 電路時序規格

2.4.1 CLE 電路之時序圖



圖十一、CLE 電路時序圖

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。

註：SRAM 資料正確性之檢驗期間，不會花費到任何 simulation time。

2.4.2 ROM_128x8、SRAM_1024x8 之時序圖

有關 ROM_128x8、SRAM_1024x8 記憶體細項規格與記憶體讀寫時序圖，詳如記憶體附件中。

◇ 評分項目二：依”面積”(Area)大小評分

各參賽隊伍將 APR 完成後，面積分析方法如下範例，請任選其一 APR 軟體做分析。

1. IC Compiler Report Area 範例:

```
icc_shell> get_attribute [get_die_area] bbox  
{0.000 0.000} {379.540 377.310}
```

=> Area =379.540 x 377.310 =143204.2374 um²

2. Encounter Report Area 範例:

encounter > analyzeFloorplan

```
***** Analyze Floorplan *****  
Die Area(um^2)           : 142775.13  
Core Area(um^2)          : 135173.83  
Chip Density (Counting Std Cells and MACROs and IOs): 94.629%  
Core Density (Counting Std Cells and MACROs): 99.950%  
Average utilization      : 100.000%  
Number of instance(s)   : 13332  
Number of Macro(s)      : 0  
Number of IO Pin(s)     : 45  
Number of Power Domain(s) : 0  
***** Estimation Results *****  
*****
```

=> Area = 142775.13 um²

註：指令 analyzeFloorplan 會破壞已完成 routing 的結果，執行該指令後絕對不可再存檔。

設計完成程度四種等級，如下：

◇ **等級 A：**達成”完成設計”之三項要求

- a、功能正確，RTL 模擬與正確解答比對完全正確。
- b、完成 Synthesis，且 Gate-Level Pre-layout Simulation 結果正確。
- c、完成 APR，並達成 APR 必要項目，Gate-Level Post-layout Simulation 結果正確。

註：完成 APR 之必要項目

- i. 只需做 Marco layout (即不用包含 IO Pad、Bonding Pad)。
- ii. VDD 與 VSS Power Ring 寬度請各設定為 **2um**，只須做一組。
- iii. 不需加 Dummy Metal。
- iv. **Power Stripe 務必至少加一組**，其 VDD、VSS 寬度各設定為 **2um**。
(Power Stripe 垂直方向至少一組，水平方向可不加)
- v. 務必要加 Power Rail (follow pin)。
- vi. **Core Filler 務必要加**。
- vii. APR 後之 GDSII 檔案務必產生。
- viii. 完成 APR，DRC/LVS 完全無誤(見附錄 C 說明)。

等級 A 之評分方法：

$$\text{Score} = \text{Area} \times \text{Time}^2$$

例如：

本範例(Encounter)為，

$$\text{Score} = \text{Area} \times \text{Time}^2 = 142775.13 \times 56827^2 = 461064859373005$$

註：Score 越小者，同級名次越好！

- ◇ **等級 B**：已做到 APR，但等級 A 之”APR 必要項目”有部分不符合，DRC/LVS 錯誤總數量容許 5 個(含)以下

此等級之成績計算方式如下：

$$\text{Score} = \text{Area} \times \text{Time}^2 \times (\text{DRC} + \text{LVS 的錯誤總數量})$$

註：Score 越小者，同級名次越好！

- ◇ **等級 C**：僅完成合成，或做到 APR，但 DRC/LVS 錯誤總數量超過 5 個以上

此等級之成績計算方式如下：

$$\text{Score} = \text{Area} \times \text{Time}^2$$

註：

1. Score 越小者，同級名次越好！
2. 等級 C，視 APR 為 Fail，Area 以 Design Compiler 所 Report 的 Cell Area 為主。
3. 等級 C，視 APR 為 Fail，Time 以 Gate-level Pre-layout Simulation 為主。

- ◇ **等級 D**：未達成前三等級者，成績計算方式為 All RTL Simulation，比對結果之 error 總數量越少者，分數越高。

$$\text{Score} = \text{Total error of All RTL Simulations}$$

註：

1. 等級 D，Score 評分方式為所有模擬的 error 總數作相加。
2. 等級 D，視合成及 APR 皆為 Fail，Area、Timing 將不予考慮之。
3. 等級 D，只以 RTL Simulation 正確率為主，Score 越小者(即 error 越少)，同級名次越好。
4. 等級 D，若本題寫入 SRAM 的資料每一筆假若都是相同的值，例如：全部以 8'h00 寫入至 SRAM，則評分結果將被視為 Fail(即每張圖片的 error 數量=1024)，請注意！

附錄

附錄 A 為主辦單位所提供各參賽者的設計檔案說明；附錄 B 為主辦單位提供的測試樣本說明；附錄 C 為設計驗證說明；附錄 D 為評分用檔案，亦即參賽者必須繳交的檔案資料；附錄 E 則為設計檔案壓縮整理步驟說明；附錄 F 中說明本次競賽之軟體環境；附錄 G 中說明本次競賽使用之設計資料庫。

附錄 A 設計檔(For Verilog)

1. 下表為主辦單位所提供各參賽者的設計檔

表 3、設計檔案說明

檔名	說明
CLE.v	參賽者所使用的設計檔，已包含系統輸/出入埠之宣告。
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。
rom_128x8_a.v rom_128x8_b.v rom_128x8_c.v	CLE 電路會用到的三組 32x32 pixels 之影像，已擺在 ROM 檔案裡。當模擬軟體看到 rom_128x8_a.v 檔案時，會自動讀取影像資料 rom_128x8_verilog_a.rcf ；同理，當模擬軟體看到 rom_128x8_b.v 檔案時，會自動讀取影像資料 rom_128x8_verilog_b.rcf ，依此類推！ 註：此三檔案已被 include 到 TestBench。
sram_a.dat sram_b.dat sram_c.dat	sram_a.dat 為圖片 a 的 Golden 結果檔案。 sram_b.dat 為圖片 b 的 Golden 結果檔案。 sram_c.dat 為圖片 c 的 Golden 結果檔案。 註：此三檔案已被加入到 TestBench。
.synopsys_dc.setup	使用 Design Compiler 作合成或 IC Compiler Layout 之初始化設定檔。參賽者請依 Library 實際擺放位置，自行修改 Search Path 的設定。 注意：無論合成或 APR，只需使用 worst case library。
CLE_DC.sdc	Design Compiler 作合成之 Constraint 檔案。參賽者可依需求自行修改 cycle 的設定。 注意：環境相關參數請勿更改。
CLE_APR.sdc	Encounter 或 IC Compiler 作 Layout 之 Constraint 檔案。參賽者可依需求自行修改 cycle 的設定。 注意：環境參數請勿更改。

請使用 **CLE.v**，進行 CLE 電路之設計。其模組名稱、輸出/入埠宣告如下所示：

```
`timescale 1ns/10ps
module CLE ( clk, reset, rom_q, rom_a, sram_q, sram_a, sram_d, sram_wen, finish);
input      clk;
input      reset;
input [7:0] rom_q;
output [6:0] rom_a;
input [7:0] sram_q;
output [9:0] sram_a;
output [7:0] sram_d;
output      sram_wen;
output      finish;
endmodule
```

2. 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，**參賽者只要注意 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！**
3. 本題所提供的 Test Bench 檔案，有多增加幾行特別用途的敘述如下：

```
`define SDFFILE      "./CLE_syn.sdf"
`ifdef SDF
    initial $sdf_annotate(`SDFFILE, u_CLE);
`endif
```

註：

1. SDF 檔案，請自行修改 SDF 實際檔名及路徑後再模擬。
2. 在 Test Bench 中，主辦單位提供`ifdef SDF 的描述，其目的是讓本 Test Bench 可以作為 RTL 模擬、合成後模擬與 Layout 後模擬皆可使用。注意：當參賽者在合成或 Layout 後模擬，請務必多加一個參數”**+define+SDF**”，方可順利模擬。

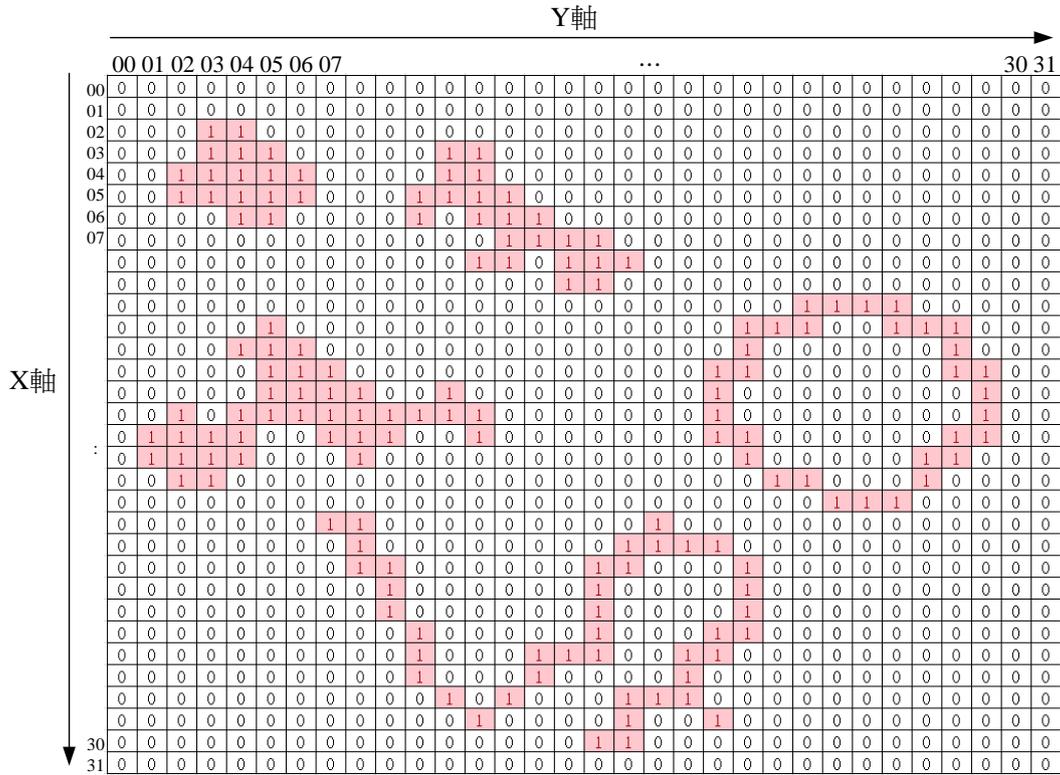
例如：當合成後，使用 NC-Verilog 模擬第一組樣本，在 UNIX 下執行下面指令

```
> ncverilog +ncmaxdelays testfixture_a.v CLE_syn.v -v tsmc13_neg.v +define+SDF
+access+rw
```

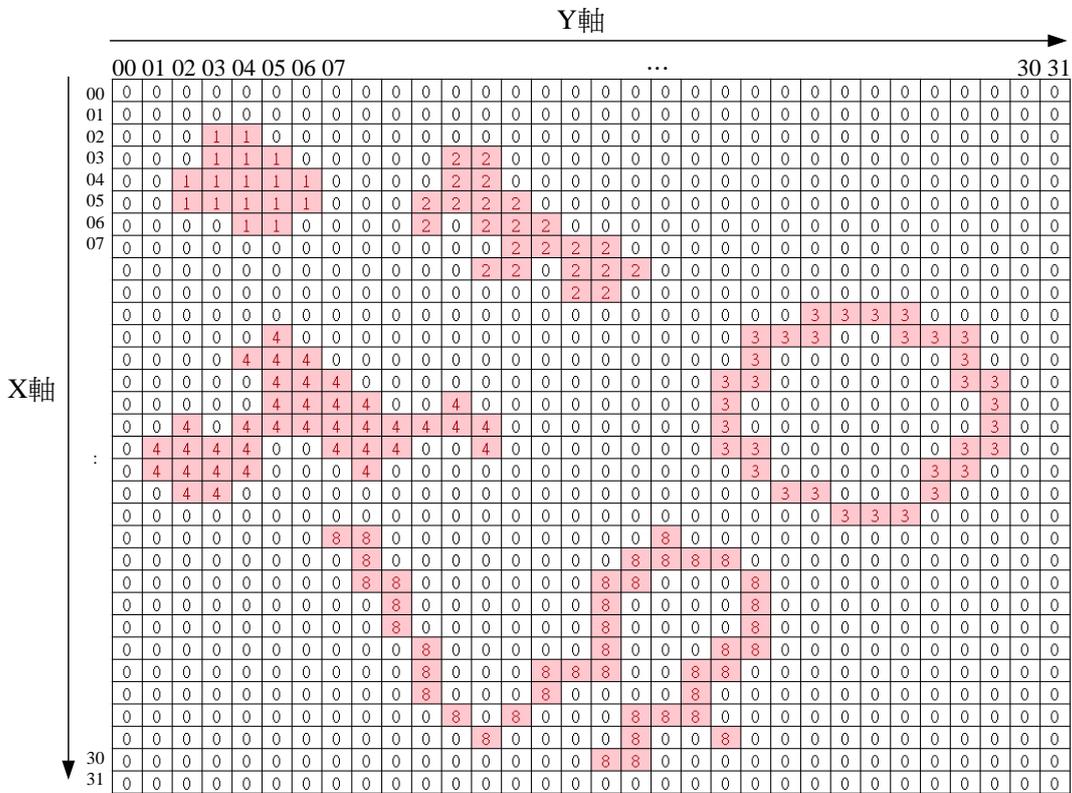
註：使用 Encounter 作 APR 之參賽者，APR 後的模擬務必加上+ncmaxdelays 參數，請注意！

附錄 B 測試樣本

測試樣本 a: (32x32pixel 影像已存於 ROM，其儲存內容可參考 rom_a.dat，有詳細的座標說明)



圖十二、測試樣本_a(已儲存於 rom_128x8_verilog_a.rcf)



圖十三、測試樣本_a - 編號後的參考解答(已儲存於 sram_a.dat，編號值僅供參考)

附錄 C 設計驗證說明

參賽者繳交資料前應完成 RTL，Gate-Level 與 Physical 三種階段驗證，以確保設計正確性。**注意：**每組限定只能使用 1 license, 勿使用 Multi-CPU。

- RTL 與 Gate-Level 階段：參賽者必須進行 RTL simulation 及 Gate-Level simulation，模擬結果必須於參賽者提供之 CYCLE 數值下，功能完全正確。
- Physical 階段，包含三項驗證重點：
 1. 依主辦單位各項要求，實現完整且正確的 layout (詳細之各項要求，請見評分標準)。
 2. 完成 post-layout simulation：參賽者必須使用 P&R 軟體寫出之 netlist 檔與 sdf 檔完成 **post-layout gate-level simulation**，以下分為 IC Compiler、Encounter 兩種軟體說明 netlist 與 sdf 寫出步驟。

- i. 使用 Synopsys IC Compiler 者，執行步驟如下：

在 IC Compiler 主視窗底下點選

“ **File > Export > Write SDF...** ”

Specify Version	Version 2.1
Instance	空白即可
File name	CLE_pr.sdf
Significant digits	2

按 。

對應指令： `write_sdf -version 2.1 CLE_pr.sdf`

“ **File > Export > Write Verilog...** ”

先按

Output verilog file name	CLE_pr.v
Output physical only cells	disable
Wire declaration	enable
Backslash before Hierarchy Separator	Enable
All other options	Default value

按 。

- ii. 使用 Cadence Encounter 者，執行步驟如下：

在 Encounter 視窗下點選：

“ **File → Save → Netlist...** ”

Netlist File	CLE_pr.v
All other options	Default value

按 。

“ **Timing → Extract RC...** ”，按 。

“Timing → Write SDF...”

Ideal Clock	Disable
SDF Output File:	CLE_pr.sdf

按 。

3. 完成 DRC 與 LVS 驗證：參賽者必須以其所使用之 P&R 軟體內含之 DRC 與 LVS 驗證功能完成 DRC 與 LVS 驗證，以下分為 IC Compiler、Encounter 兩種軟體說明執行步驟。

- i. 使用 Synopsys IC Compiler 者，驗證 DRC 與 LVS 步驟如下：

在 IC Compiler Layout 視窗底下點選

“Route > Verification > DRC ...”

Read child cell from	Cell view
All other options	Default value

按 。

將跳出 Error Browser 視窗，請參賽者自行查看是否有錯，若有請自行修改 Layout 到 0 個 Violation 為止。

“Route > Verification > LVS ...”

Pins not connected to a wire segment(Floating port)	disable
All other options	Default value

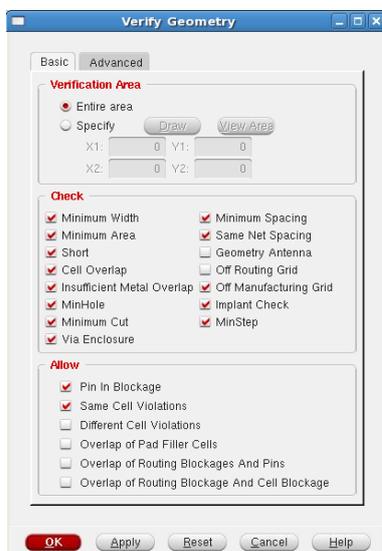
按 。

將跳出 Error Browser 視窗，檢查看看是否有錯，若有請自行修正到 0 個 Violation 為止。

- ii. 使用 Cadence Encounter 者，驗證 DRC 與 LVS 步驟如下：

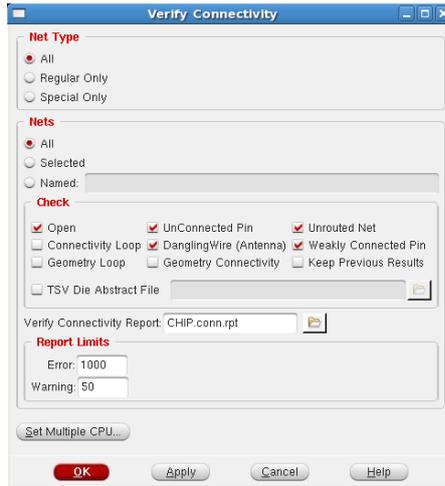
在 Encounter 視窗下點選

1. DRC 驗證：請選“Verify → Verify Geometry...” Default 值，按 。



註：若 DRC 有發生錯誤，請選“Tools → Violation Browser...”查明原因。

2. LVS 驗證：請選“Verify → Verify Connectivity...” Default 值，按 **OK**。



註：若 LVS 有發生錯誤，請選“Tools → Violation Browser...”查明原因。

附錄 D 評分用檔案

評分所須檔案可以下幾個部份：(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，請務必將合成所要用的各 module 檔放進來，以免評審進行評分時，無法進行模擬；(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)Physical design，使用 **Synopsys IC Compiler** 者，請記得將整個 **Milkyway Library** 等相關的 design database，壓縮成一個檔案。使用 **Cadence Encounter** 者，請將 Encounter 相關的 design database (包含 **.enc** 檔案與 **.enc.dat** 目錄)，壓縮成一個檔案。壓縮的檔案格式如下：假設參賽者的 design database 目錄名稱爲”your_lib”，請執行底下的 UNIX 指令，最後可以得到”your_name.tar”的檔案。

```
> tar cvf your_name.tar your_lib
```

在執行以上的指令之前，請確定將你使用的 P&R Tool 儲存後關閉，再執行上述的指令，否則在壓縮的過程會出現錯誤。

表 4

RTL category		
<i>Design Stage</i>	<i>File</i>	<i>Description</i>
N/A	N/A	Design Report Form
RTL Simulation	*.v or *.vhd	Verilog (or VHDL) synthesizable RTL code
Gate-Level category		
<i>Design Stage</i>	<i>File</i>	<i>Description</i>
Pre-layout Gate-level Simulation	*_syn.v	Verilog gate-level netlist generated by Synopsys Design Compiler
	*_syn.sdf	Pre-layout gate-level sdf
Physical category		
<i>Design Stage</i>	<i>File</i>	<i>Description</i>
P&R	*.tar	archive of the design database directory
	*.gds	GDSII layout
	DRC/LVS report	不用儲存 DRC/LVS Report 檔案!只需在 Design Report Form 上填寫 DRC/LVS 錯誤總數量 即可。(目標要做到 0 個錯誤!)
Post-layout Gate-level Simulation	*_pr.v	Verilog gate-level netlist generated by Cadence Encounter or Synopsys IC Compiler
	*_pr.sdf	Post-layout gate-level sdf

附錄 E 檔案整理步驟

當所有的文件準備齊全如表 4 所列，請按照以下的步驟指令，提交相關設計檔案，將所有檔案複製至同一個資料夾下，步驟如下：

1. 在自己的 home directory 建立一個新目錄，名稱叫做“**result**” 例如：
 - > **mkdir ~/result**
2. 將附錄 D 要求的檔案複製到 result 這個目錄。例如：
 - > **cp CLE.v ~/result/**
 - > **cp CLE_syn.v ~/result/**
 -
3. 在 Design Report Form 中，填入所需的相關資訊。

附錄 F 軟體環境

1. 使用者登入後自動會設定好以下軟體環境：

Vendor	Tool	Executable
Cadence	Virtuoso *1	icfb
	Composer	icfb
	NC-Verilog	nverilog
	SOC Encounter	encounter
Synopsys	Design Compiler	dv, dc_shell
	VCS-MX	vcs
	IC Compiler	icc_shell -gui
	Hspice	hspice
	Cosmos Scope *1	cscope
	Custom Explorer *1	wv
	Laker *1	laker
	Laker ADP*1	adp
	Verdi *1	verdi, nWave, nLint
Mentor	Calibre *3	calibre
	ModelSim	vsim
Utility	vi	vi, vim
	gedit	gedit
	nedit	nedit
	pdf reader	acroread
	calculate	gnome-calculator, bc -l
	gcc	gcc

EDA 軟體所須使用的 license 皆已設定完成，不須額外設定

*1 該軟體限定使用 1 套 license

*3 該軟體限定使用 3 套 license

附錄 G 設計資料庫

設計資料庫位置： /usr/cad/icc2016/CBDK_IC_Contest_v2.1

目錄架構

ICC/

tsmc13gfsg_fram/	ICC core library
tsmc13_CIC.tf	ICC technology
macro.map	layer mapping file
tluplus/	
t013s8mg_fsg_typical.tluplus	t13 tluplus file
t013s8mg_fsg.map	t13 tluplus mapping file

SOCE/

lef/	
tsmc13fsg_8lm_cic.lef	LEF for core cell
antenna_8.lef	LEF for antenna
lib/	
slow.lib	worst case for core cell
streamOut.map	Layout map for GDSII out

SynopsysDC/

db/	
slow.db	Synthesis model (slow)
lib/	
slow.lib	timing and power model

Verilog/

tsmc13_neg.v	Verilog simulation model
--------------	--------------------------

Phantom/

tsmc13gfsg_fram.gds	Standard Cell GDSII file
---------------------	--------------------------

