

初賽時間為 4 月 1 日(二) **8:30~20:30**，參賽者請於**10 點半前**完成下列步驟進行初賽登錄，主辦單位將依完成此步驟之隊伍數決定各組最後得獎名額，請務必完成登錄動作，以免影響你的權益。

- **請將你的隊伍參賽資料E-mail至ic2008@vlsilab.ee.ncku.edu.tw**，

信件內容格式如下:

**參賽組別：類比電路設計**

**參賽編號：(例：9740001)**

**參賽姓名：張三、李四**

- 信件標題請標示為「**IC 設計競賽初賽資料登錄**」

# 2008 University/College IC Design Contest

## Cell-Based IC Design Category

### *Image Display Controller*

#### 1. 問題描述

請完成一影像顯示控制(Image Display Control)電路設計。此控制電路，可依指定之操控指令，使顯示端的影像進行影像放大(Zoom In)、縮小(Zoom Fit)、左旋(Rotate Left)、右旋(Rotate Right)與水平及垂直方向的平移(Shift)功能。

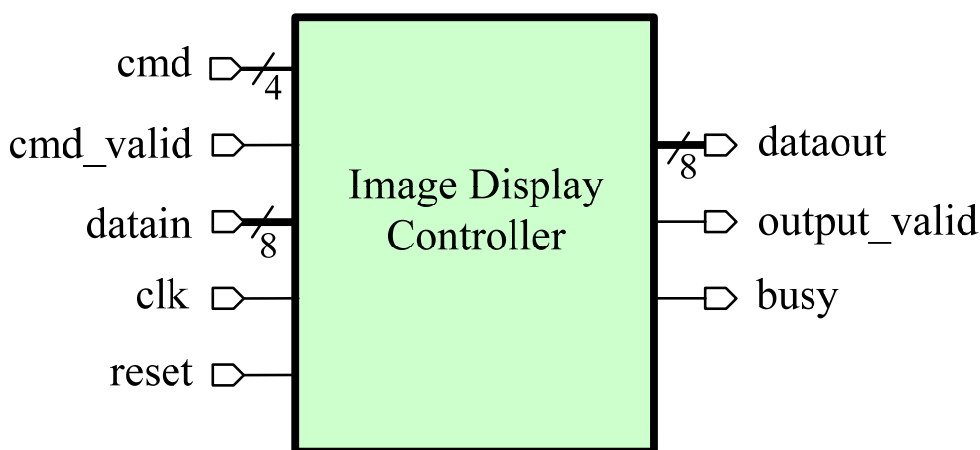
本控制電路有 5 只信號輸入(cmd, cmd\_valid, datain, clk, reset)及 3 只信號輸出(dataout, output\_valid, busy)，關於各輸入輸出信號的功能說明，請參考表一。

每個參賽隊伍必須根據下一節所給的設計規格完成設計。參賽隊伍可藉由 CIC 所提供的輸入指令及正確結果檔來檢查設計是否有達到要求，詳情請參考附錄 B。

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 D 中所列的要求，附上評分所需要的檔案。

#### 2. 設計規格

##### 2.1 系統方塊圖



圖一、系統方塊圖

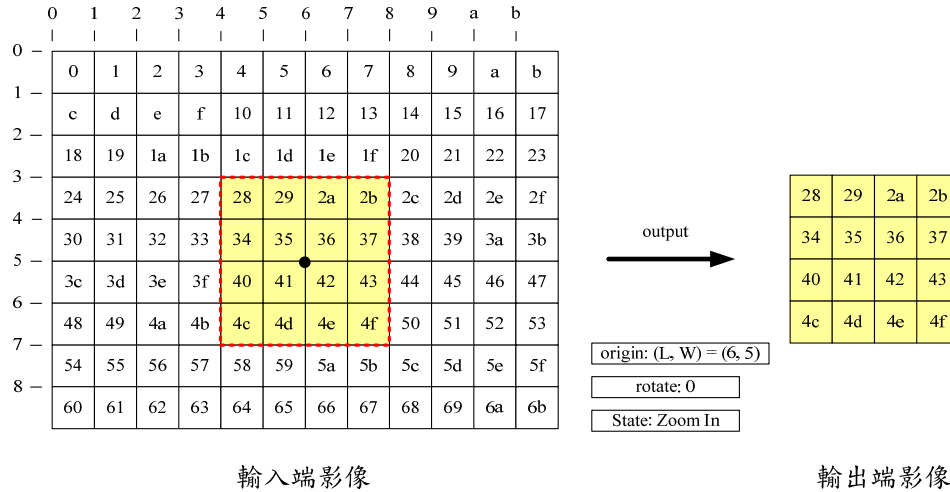
## 2.2 輸入/輸出介面

表一、輸入/輸出訊號

信號名稱	輸/出入	位元寬度	說明
<i>reset</i>	input	1	高位準非同步(active high asynchronous)之系統重置信號。 說明:本信號應於系統啟動時送出。
<i>clk</i>	input	1	時脈信號。 說明:此系統為同步於時脈正緣(posedge)之同步設計。
<i>cmd</i>	input	4	指令輸入信號。 說明:本控制器共有九種指令輸入，相關指令說明請參考表二。指令輸入只有在 <i>cmd_valid</i> 為 high 及 <i>busy</i> 為 low 時，為有效指令。
<i>cmd_valid</i>	input	1	有效指令輸入信號。 說明:當本信號為 high 時表示 <i>cmd</i> 指令為有效指令輸入。
<i>datain</i>	input	8	八位元影像資料輸入埠。
<i>dataout</i>	output	8	八位元影像資料輸出埠。
<i>output_valid</i>	output	1	有效資料輸出信號。 說明:當本信號為 high 時表示 <i>dataout</i> 為有效資料輸出。
<i>busy</i>	output	1	系統忙碌信號。 說明:當本信號為 high 時，表示此控制器正在執行現行(current)指令，而無法接收其他新的指令輸入。

## 2.3 系統功能描述

影像顯示控制器之輸入端，為一張 12x9 大小的影像資料。而輸出端則為 4x4 大小的顯示影像，如圖二所示。影像顯示控制器必須處理使用者輸入之指令，取得顯示相關之座標 (origin)、旋轉 (rotate) 方向及顯示狀態 (state) 等參數，使顯示端能輸出左旋轉、右旋轉、放大、縮小及平移後之畫面。

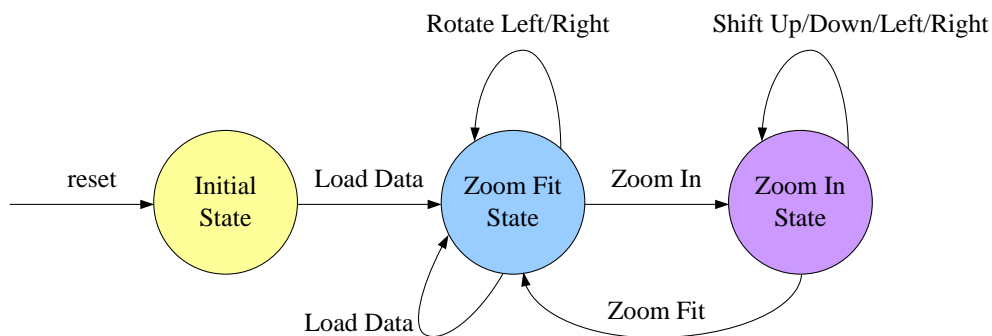


圖二、系統功能圖

影像顯示控制器其指令操作流程圖如圖三所示。一開始由外界輸入 reset 訊號，進入 Initial State 後，**接收到第一道指令必定為 Load Data**，此時顯示狀態進入 Zoom Fit State 並將該影像作取樣輸出至顯示端。

**旋轉指令 (Rotate Left/Right) 只有在 Zoom Fit State 下才有作用**，透過該指令可以決定影像旋轉方向。在 Zoom Fit State 下，若接收到 Zoom In 指令，此時影像會以**目前旋轉方向**進入 Zoom In State。

**平移指令 (Shift Up/Down/Left/Right) 只有在 Zoom In State 才會有作用**，透過該指令可以進行影像放大後的移位。在 Zoom In State 下，若收到 Zoom Fit 指令，此時影像會以**目前旋轉方向**回到 Zoom Fit State。



圖三、影像控制指令之流程圖

### 2.3.1 輸入與輸出端之影像及參數規範

#### [影像輸入]

主辦單位將提供輸入端影像資料，此影像資料為 12x9 共 108 筆測試樣本，每筆樣本為 8 位元資料。其順序為由左而右；由上而下，且以序列(Serial)的方式循序輸入至參賽者的影像控制電路中。(如圖三所示資料，輸入順序為 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f, 10, ……，69, 6a, 6b) 註:以下僅為圖例示範，詳細的輸入影像內容值未必如下圖四所示。

	0	1	2	3	4	5	6	7	8	9	a	b
0 -	0	1	2	3	4	5	6	7	8	9	a	b
1 -	c	d	e	f	10	11	12	13	14	15	16	17
2 -	18	19	1a	1b	1c	1d	1e	1f	20	21	22	23
3 -	24	25	26	27	28	29	2a	2b	2c	2d	2e	2f
4 -	30	31	32	33	34	35	36	37	38	39	3a	3b
5 -	3c	3d	3e	3f	40	41	42	43	44	45	46	47
6 -	48	49	4a	4b	4c	4d	4e	4f	50	51	52	53
7 -	54	55	56	57	58	59	5a	5b	5c	5d	5e	5f
8 -	60	61	62	63	64	65	66	67	68	69	6a	6b

圖四、輸入端影像資料

#### [影像輸出]

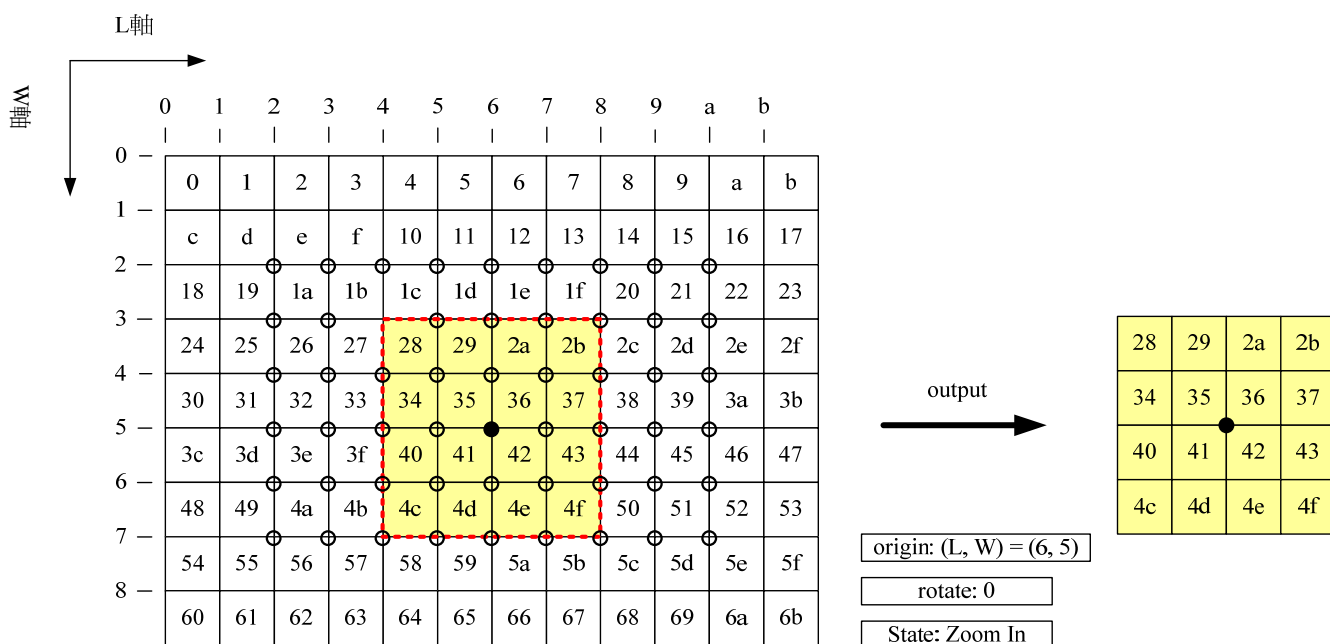
輸出端影像為 4x4 共 16 筆樣本輸出，每筆樣本為 8 位元資料。其順序是由左而右；由上而下，且以序列(Serial)的方式循序輸出結果。(如圖五所示資料，輸出順序為 28,29,2a,2b,34,35,36,37,40, 41,42,43,4c,4d,4e,4f) 。註:以下僅為圖例示範，詳細的輸出影像內容值未必如下圖五所示。

28	29	2a	2b
34	35	36	37
40	41	42	43
4c	4d	4e	4f

圖五、輸出端影像資料

#### [輸出影像原點座標]

輸出影像原點座標(origin)指的是輸出影像相對於輸入影像的參考座標，本試題已定義輸入端影像之座標軸。輸入端影像為 4:3 之比例，因此不論影像如何旋轉，影像之長邊固定稱作 L 軸，短邊固定稱作 W 軸，而 origin 表示輸出影像之正中間對應輸入影像的位置。此外，**origin 在 L 軸座標可移動範圍為 2~a，在 W 軸可移動座標範圍為 2~7**。(如圖六所示，為確保輸出影像不超出輸入影像邊界，因此限制原點之 L 軸移動範圍最小是 2、最大至 a，而 W 軸移動範圍最小是 2、最大至 7)。參賽者需根據此參考座標，進行顯示端的畫面平移(Shift)功能設計。



圖六、輸入端影像原點座標

#### [旋轉方向]

載入後的影像，隨著後續輸入的指令讓影像有三種方向：

- ◆ Rotate: 0，表示影像沒有做任何旋轉，如圖七所示為 Zoom Fit State 時不作任何旋轉之輸出。
- ◆ Rotate: Left，表示影像向左旋轉 90 度之畫面，如圖八所示為 Zoom Fit State 時 Rotate Left 的輸出。
- ◆ Rotate: Right，表示影像向右旋轉 90 度之畫面，如圖九所示為 Zoom Fit State 時 Rotate Right 的輸出。

註：只有在 Zoom Fit State 才可以進行影像旋轉。

#### [顯示狀態]

顯示狀態分為 Zoom Fit State 及 Zoom In State 兩種。當收到放大(Zoom In)或縮小(Zoom Fit)指令時會影響顯示狀態。

- ◆ Zoom Fit State 輸出取樣方式：如圖七所示，當影像在 Zoom Fit State 時輸出點固定為 d,10,13,16,25,28,2b,2e,3d,40,43,46,55,58,5b,5e。
- ◆ Zoom In State 輸出方式是以 1:1 的顯示比例來顯示原輸入影像的部分區塊。

註：4x4 顯示端之輸出順序會隨著影像旋轉方向而有所改變。

## 2.3.2 影像顯示控制器功能規範

[指令定義]

影像控制器電路有 9 項控制指令功能。輸入指令(cmd)所對應之功能如表二所示。

表二、控制指令定義

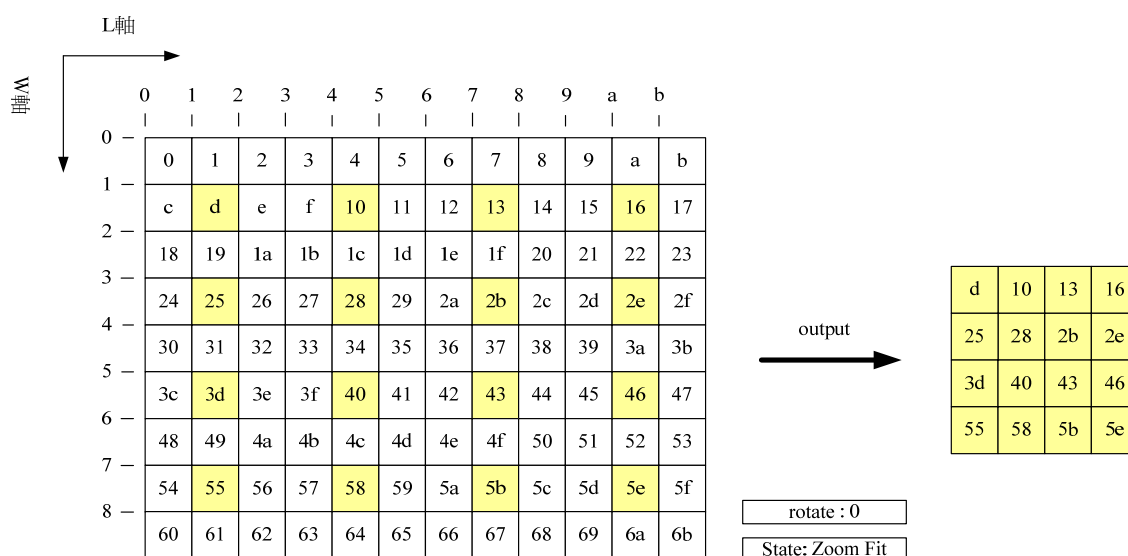
cmd 編號	控制指令說明
0	Load Data
1	Rotate Left
2	Rotate Right
3	Zoom In
4	Zoom Fit
5	Shift Right
6	Shift Left
7	Shift Up
8	Shift Down

對於所有有效之控制指令，都必須在指令處理結束後輸出 16 筆資料至顯示端。以下分別就各項指令進行說明。

### ◆ 資料載入(Load Data)

⇒ Load Data 指令，進入 Zoom Fit State，將 108 筆影像資料依序載入於 12x9 影像緩衝器中，並將其畫面作取樣後輸出至 4x4 顯示端，其輸出順序固定為 d,10,13,16,25,28,2b,2e,3d,40,43,46,55,58,5b,5e。

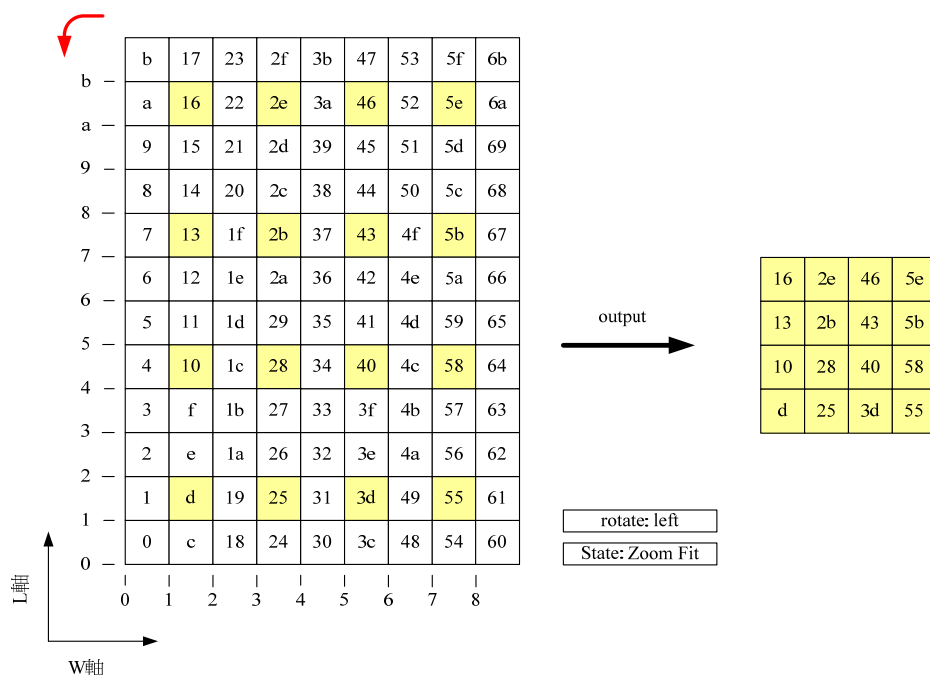
⇒ Load Data 必定為系統的第一個指令輸入。



圖七、畫面載入(Load Data)之取樣位置

◆ 畫面左旋轉(Rotate Left)

- ⇒ Rotate Left 指令，狀態維持不變，逆時針旋轉影像 90 度如圖八，影像水平方向為 W 軸，垂直方向為 L 軸，並將左旋轉後畫面作取樣輸出至 4x4 顯示端，其輸出順序固定為 16,2e,46,5e,13,2b,43,5b,10,28,40,58,d,25,3d,55。
- ⇒ Rotate Left 指令，只有在 Zoom Fit State 時，旋轉指令才有作用。反之，在 Zoom In State 時，輸出至顯示端的值再重複輸出一遍。
- ⇒ 主辦單位所提供之左旋轉指令，不會讓畫面旋轉 180 度，最多只有向左旋轉 90 度。

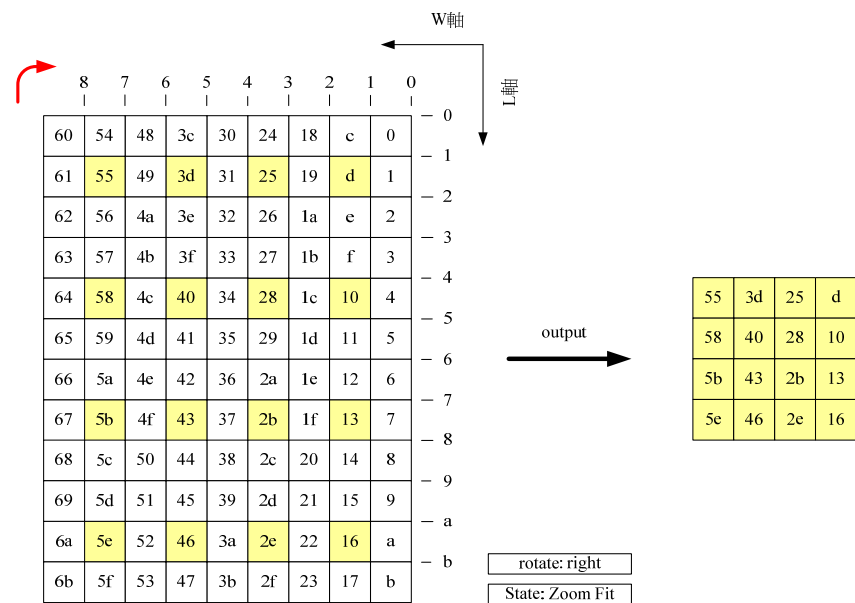


圖八、畫面左旋轉(Rotate Left)之取樣位置

◆ 畫面右旋轉(Rotate Right)

- ⇒ Rotate Right 指令，狀態維持不變，順時針旋轉影像 90 度如圖九，影像水平方向為 W 軸，垂直方向為 L 軸，並將右旋轉後畫面作取樣輸出至 4x4 顯示端，其輸出順序固定為 55,3d,25,d,58,40,28,10,5b,43,2b,13,5e,46,2e,16。
- ⇒ Rotate Right 指令，只有在 Zoom Fit State 時，旋轉指令才有作用。反之，在 Zoom In State 時，輸出至顯示端的值再重複輸出一遍。
- ⇒ 主辦單位所提供之右旋轉指令，不會讓畫面旋轉 180 度，最多只有向右旋轉 90 度。

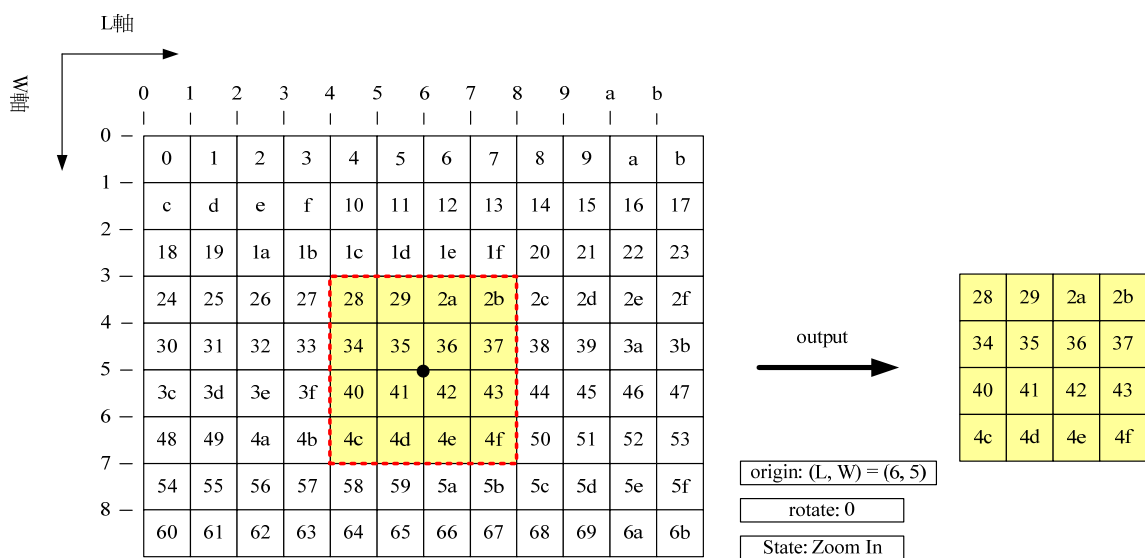




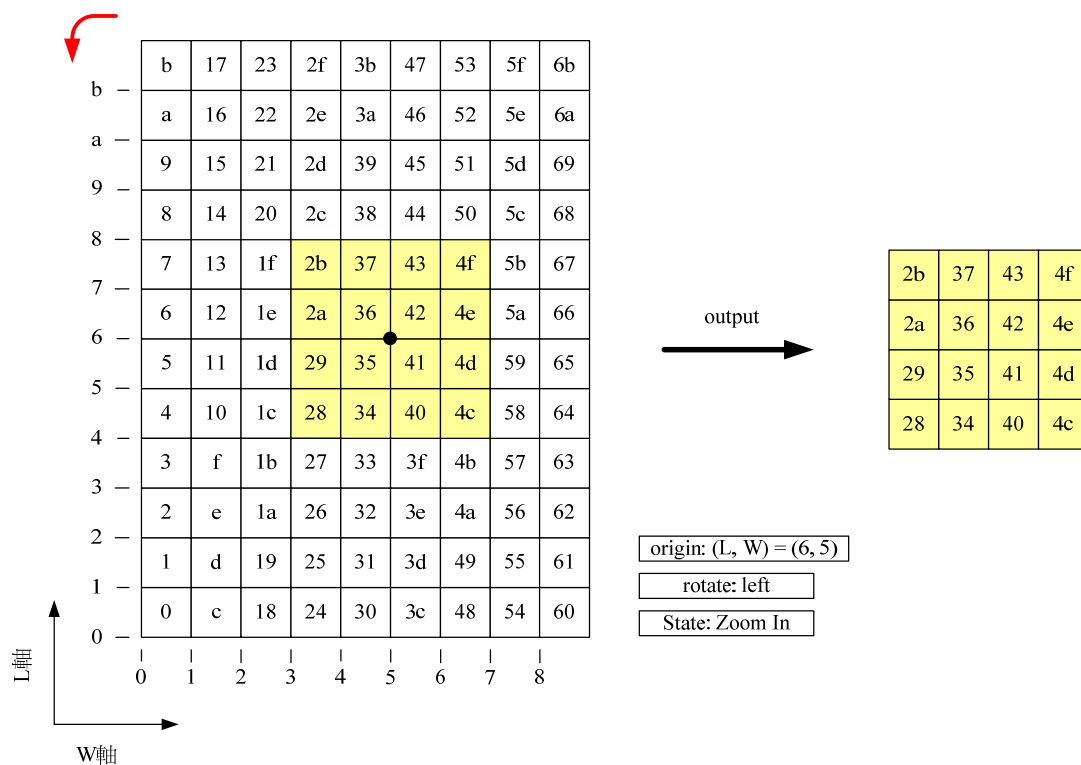
圖九、畫面右旋轉(Rotate Right)之取樣位置

#### ◆ 畫面放大(Zoom In)

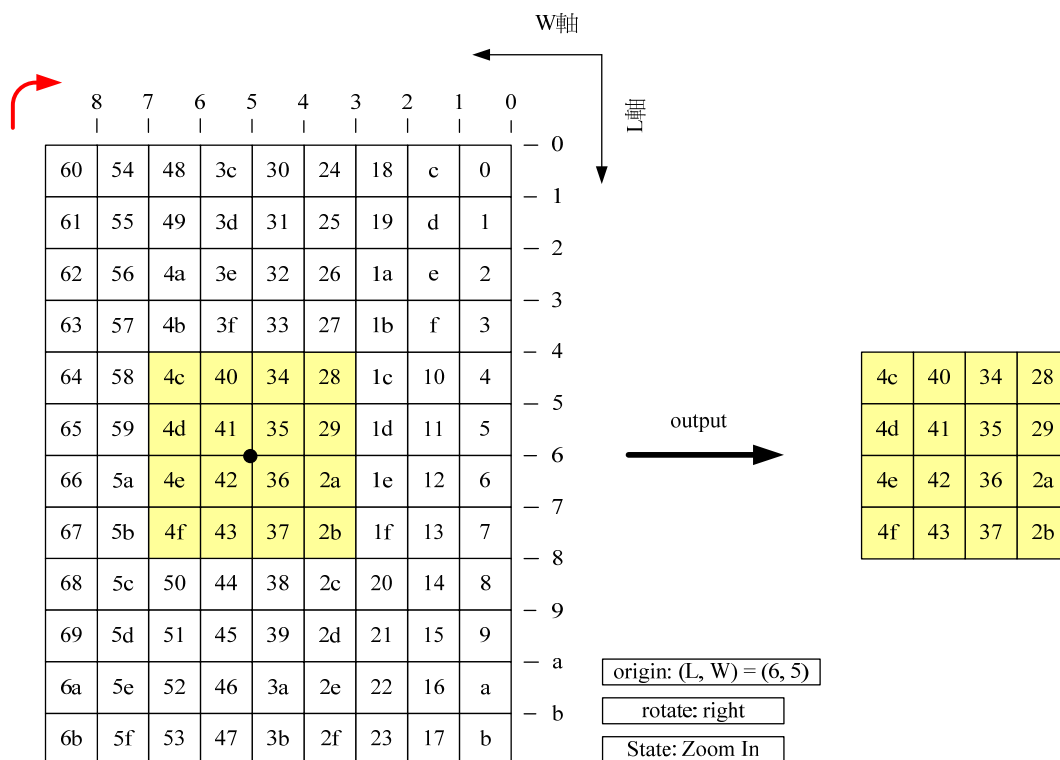
- ⇒ Zoom In 指令，進入 Zoom In State，此時影像是以 1:1 的顯示比例來顯示原輸入影像的部分區塊，並將該區塊內容輸出至 4x4 顯示端。
- ⇒ Zoom In 指令會自動將原點 origin 指向(6,5)。
- ⇒ 移動(Shift)指令，只有在該 Zoom In State 才有效用。
- ⇒ Zoom In State 中，主辦單位不會在此狀態輸入 Load Data 指令。
- ⇒ Zoom In State，畫面旋轉方向不變，因此放大模式有三種初始之輸出畫面：
  - 當放大前畫面 Rotate: 0 如圖七，放大後初始畫面如圖十。
  - 當放大前畫面 Rotate: Left 如圖八，放大後初始畫面如圖十一。
  - 當放大前畫面 Rotate: Right 如圖九，放大後初始畫面如圖十二。



圖十、放大(Zoom In)前畫面 Rotate: 0，放大後之結果



圖十一、放大(Zoom In)前畫面 Rotate: Left，放大後之結果



圖十二、放大(Zoom In)前畫面 Rotate: Right，放大後之結果

◆ 畫面縮小(Zoom Fit)

⇒ Zoom Fit 指令，進入 Zoom Fit State，並將縮小後之畫面作取樣輸出至 4x4 顯示端。

⇒ Zoom Fit State 中，畫面旋轉方向不變，因此縮小模式有三種輸出可能：

- 當畫面 Rotate: 0 如圖七，其顯示端輸出順序固定為 d,10,13,16,25,28,2b,2e,3d,40,43,46,55,58,5b,5e。
- 當畫面 Rotate: Left 如圖八，其顯示端輸出順序固定為 16,2e,46,5e,13,2b,43,5b,10,28,40,58,d,25,3d,55。
- 當畫面 Rotate: Right 如圖九，其顯示端輸出順序固定為 55,3d,25,d,58,40,28,10,5b,43,2b,13,5e,46,2e,16。

⇒ 在 Zoom Fit State 中，無法做上下左右等平移功能。假使在 Zoom Fit State 中收到上下左右平移(Shift Up/Down/Left/Right)的指令時，顯示結果仍維持不變，僅重覆輸出相同之取樣結果。

◆ 畫面右移(Shift Right)

⇒ 右移顯示區塊。執行此 Shift Right 指令，是將目前畫面之 origin 向右移 1 格，並將移動後之畫面輸出至 4x4 顯示端，意指為：

- 當畫面 Rotate: 0 如圖十，origin 在 L 軸座標值增加 1，但最大不可大於 a。
- 當畫面 Rotate: Left 如圖十一，origin 在 W 軸座標值增加 1，但最大不可大於 7。
- 當畫面 Rotate: Right 如圖十二，origin 在 W 軸座標值減少 1，但最小不可小於 2。

⇒ 當 origin 已到達 L 或 W 軸座標之可移動的邊界點上時，假使再收到右移指令，origin 將維持原來值，即輸出顯示端內容值不作改變。

◆ 畫面左移(Shift Left)

⇒ 左移顯示區塊。執行此 Shift Left 指令，是將目前畫面之 origin 向左移 1 格，並將移動後之畫面輸出至 4x4 顯示端，意指為：

- 當畫面 Rotate: 0 如圖十，origin 在 L 軸座標值減少 1，但最小不可小於 2。
- 當畫面 Rotate: Left 如圖十一，origin 在 W 軸座標值減少 1，但最小不可小於 2。
- 當畫面 Rotate: Right 如圖十二，origin 在 W 軸座標值增加 1，但最大不可大於 7。

⇒ 當 origin 已到達 L 或 W 軸座標之可移動的邊界點上時，假使再收到左移指令，origin 將維持原來值，即輸出顯示端內容值不作改變。

◆ 畫面上移(Shift UP)

⇒ 上移顯示區塊。執行此 Shift Up 指令，是將目前畫面之 origin 向上移 1 格，並將移動後之畫面輸出至 4x4 顯示端，意指為：

- 當畫面 Rotate: 0 如圖十，origin 在 W 軸座標值減少 1，但最小不可小於 2。
- 當畫面 Rotate: Left 如圖十一，origin 在 L 軸座標值增加 1，但最大不可大於 a。

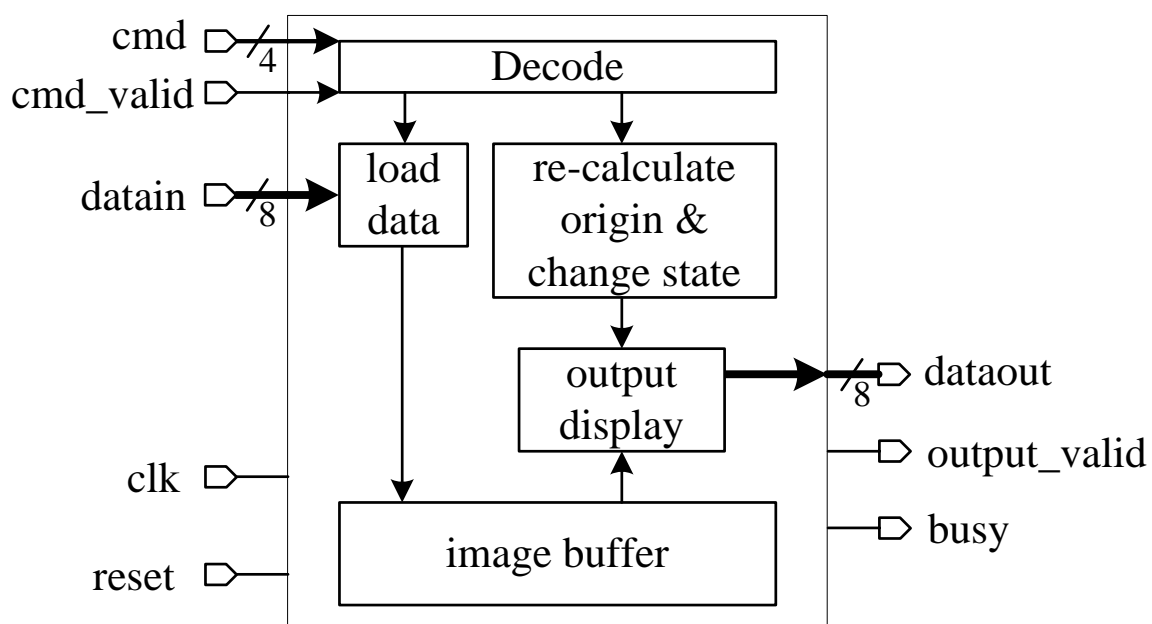
- 當畫面 Rotate: Right 如圖十二，origin 在 L 軸座標值減少 1，但最小不可小於 2。
- ⇒ 當 origin 已到達 L 或 W 軸座標之可移動的邊界點上時，假使再收到上移指令，origin 將維持原來值，即輸出顯示端內容值不作改變。

#### ◆ 畫面下移(Shift Down)

- ⇒ 下移顯示區塊。執行此 Shift Down 指令，是將目前畫面之 origin 向下移 1 格，並將移動後之畫面輸出至 4x4 顯示端，意指為：
  - 當畫面 Rotate: 0 如圖十，origin 在 W 軸座標值增加 1，但最大不可大於 7。
  - 當畫面 Rotate: Left 如圖十一，origin 在 L 軸座標值減少 1，但最小不可小於 2。
  - 當畫面 Rotate: Right 如圖十二，origin 在 L 軸座標值增加 1，但最大不可大於 a。
- ⇒ 當 origin 已到達 L 或 W 軸座標之可移動的邊界點上時，假使再收到下移指令，origin 將維持原來值，即輸出顯示端內容值不作改變。

### 2.3.3 影像顯示控制器範例

以下為影像控制器電路方塊圖範例，**僅供參賽者作為設計之參考，參賽者亦可自行發揮創意！**

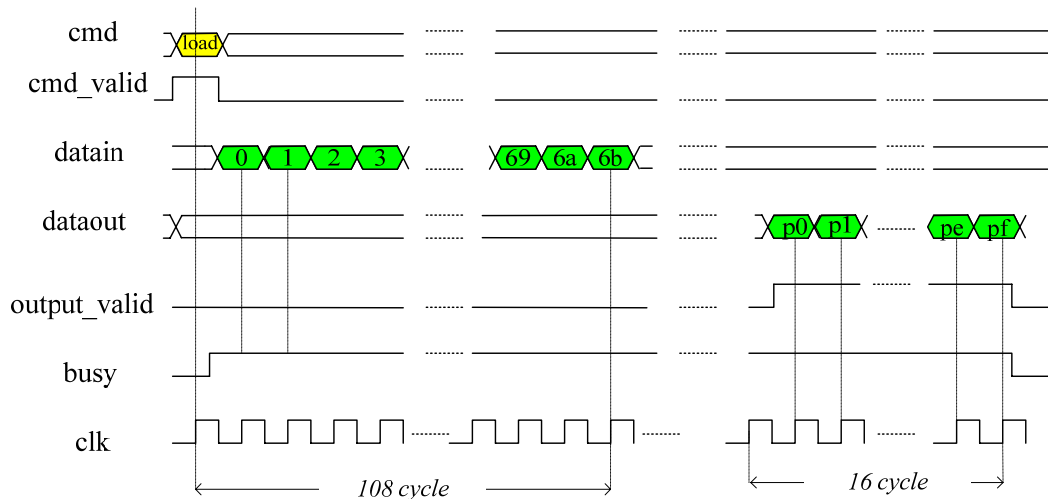


圖十三、影像控制器電路方塊圖

## 2.4 時序規格圖

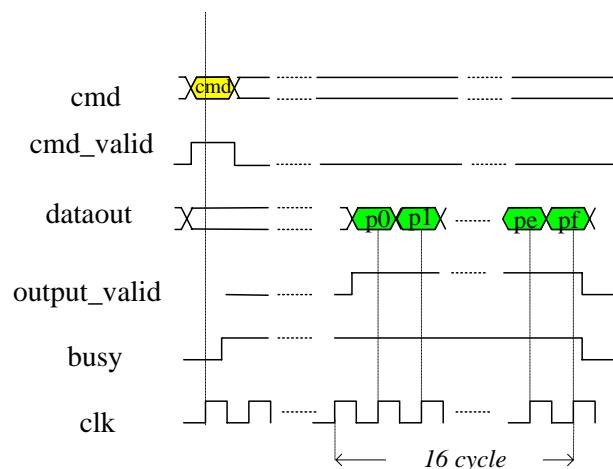
- ◆ 載入資料(Load Data)之時序規格圖，如下圖十四所示。
  - ⇒ 在有效的 Load Data 指令之後，會緊接著連續輸入 108 筆影像資料

- ⇒ 當影像顯示控制電路內部處理完成後，**連續輸出 16 筆顯示資料**，輸出同時須將 **output\_valid** 設為 high。
- ⇒ 在整個處理過程中，busy 皆維持為 high。並在輸出完成後，將 busy 設回 low 以接受新指令輸入。



圖十四、資料載入之時序規格圖

- ⇒ 其它控制指令(Rotate Left、Rotate Right、Zoom In、Zoom Fit、Shift Right、Shift Left、Shift Up、Shift Down)之時序規格圖，如下圖十五所示。
- ⇒ 當影像顯示控制電路內部處理完成，**連續輸出 16 筆顯示資料**，輸出同時須將 **output\_valid** 設為 high。
- ⇒ 在整個處理過程中，busy 皆維持為 high。並在輸出完成後，將 busy 設回 low 以接受新指令輸入。



圖十五、其它控制指令之時序規格圖

### 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 與 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 E)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 CIC 所提供的標準設計結果為準。

假若通過 Gate-level 驗證的組數不足，CIC 將本試題區分六個等級之功能完成度進行評分：

1. 影像 Load Data 功能正確。
2. 影像 Rotate Left/Right 功能正確。
3. 未 Rotate 影像之 Zoom In/Fit 功能正確。
4. 影像 Rotate Left/Right 後 Zoom In/Fit 功能正確。
5. 影像 Shift Up/Down/Left/Right 功能正確。
6. 影像所有功能完全正確。

請注意，我們將以各參賽隊伍的設計結果正確為前提，並以最後上傳檔案的時間為依據。一旦設計經評審驗證後，且參賽者的設計結果正確，上傳時間越早者，其所得到的分數就越高。但倘若參賽者於競賽時間內無法完成設計，也請務必將設計檔案內容上傳，**主辦單位亦將根據上列六個等級之功能完成度進行配分，完成越多等級者，排名相對越前面。**審查成績將另擇期通知。

## 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 為測試樣本詳細資料；附錄 D 為評分用檔案，亦即參賽者必須回傳至 CIC 的檔案資料；附錄 E 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三。驗證評分時，係以所列軟體及版本作為驗證依據。

表三、設計軟體版本

Functionality	Corresponding EDA tools
Logic Simulator	NC-Verilog v6.1 ModelSim v6.3
Logic Synthesizer	Design-Compiler v2007.03-sp3

## 附錄 B 設計檔 (For Verilog or VHDL)

1. 下表四為主辦單位所提供各參賽者的設計檔

表四、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信號
lcd_ctrl.v	參賽者所使用的設計檔，已包含系統輸/出入埠之宣告
.synopsys_dc.setup	design compiler 初始設定範例檔案
lcd_ctrl.sdc	design compiler 合成範例檔案
cmd1.dat	第一組測試樣本之指令
image1.dat	第一組測試樣本使用之影像檔
out_golden1.dat	第一組測試樣本之正確結果
cmd2.dat	第二組測試樣本之指令
image2.dat	第二組測試樣本使用之影像檔
out_golden2.dat	第二組測試樣本之正確結果
report.000	結果報告範本

2. 請使用 *lcd\_ctrl.v(.vhd)*，進行影像顯示控制器之設計。其模組名稱、輸出/入埠宣告如下所示：

```
module LCD_CTRL(clk, reset, datain, cmd, cmd_valid, dataout, output_valid, busy);
input          clk;
input          reset;
input  [7:0]   datain;
input  [3:0]   cmd;
input          cmd_valid;
output [7:0]   dataout;
output        output_valid;
output        busy;
endmodule
```

3. 比賽共提供二組測試樣本，請自行修改 *testfixture.v* 內容，以使用第二組測試樣本，修改方法如下：

```
`define IMAGE    "./image1.dat"
`define CMD      "./cmd1.dat"
`define EXPECT   "./out_golden1.dat"
```



將以上三行修改為：

```
`define IMAGE    "./image2.dat"
`define CMD      "./cmd2.dat"
`define EXPECT   "./out_golden2.dat"
```

另外，在 `testfixture.v` 檔案中有提供兩個訊號請注意：

1. 訊號 `i`，表示目前輸入是第幾筆 `cmd`。
2. 訊號 `pattern_num`，表示目前輸出是比對到第幾筆 `golden pattern`

參賽者可在 `Debug` 波形時，加入這兩個訊號，以方便自己 `Debug` 使用。

4. 請參賽隊伍使用主辦單位所提供的測試樣本(`testfixture.v`)，來進行設計之模擬驗證。主辦單位除了將利用本試題所提供的測試樣本進行各參賽隊伍之設計測試外，我們亦準備另一份測試樣本，進行設計之二次驗證。唯通過二次驗證且均能符合輸出結果，始能稱為無誤之設計。

測試樣本(`testfixture.v`) 包含 `clk`, `reset`, `datain`, `cmd`, 與 `cmd_valid` 測試信號。而於測試樣本內所定義的時脈週期參數(`CYCLE`)，可由參賽者依需求自行修改。

## 附錄 C 測試樣本

比賽共提供兩組測試樣本，為方便設計者除錯之用，將測試樣本之影像資料及指令輸入詳列如下：

### ◆ 測試樣本一

⇒ 相關檔案: image1.dat , cmd1.dat , out\_golden1.dat

⇒ 影像資料:

0	1	2	3	4	5	6	7	8	9	a	b
c	d	e	f	10	11	12	13	14	15	16	17
18	19	1a	1b	1c	1d	1e	1f	20	21	22	23
24	25	26	27	28	29	2a	2b	2c	2d	2e	2f
30	31	32	33	34	35	36	37	38	39	3a	3b
3c	3d	3e	3f	40	41	42	43	44	45	46	47
48	49	4a	4b	4c	4d	4e	4f	50	51	52	53
54	55	56	57	58	59	5a	5b	5c	5d	5e	5f
60	61	62	63	64	65	66	67	68	69	6a	6b

⇒ 指令輸入順序:

load, rotate left, rotate right, rotate right, rotate left, load, zoom in, zoom in, zoom fit, zoom fit, load, rotate left, zoom in, zoom in, zoom fit, zoom fit, rotate right, rotate right, zoom in, zoom in, zoom fit, zoom fit, load, zoom fit, shift up, shift down, shift down, shift left, shift right, shift right, zoom in, shift right, shift down, shift right, shift down, shift down, shift right, shift right, shift right, zoom fit, zoom in, shift left, shift left, shift left, shift left, shift left, shift up, shift up, shift up, shift up, shift up, zoom fit, rotate left, shift up, shift down, shift left, shift right, zoom in, shift right, shift right, shift right, shift up, shift up, shift up, shift right, shift up, shift up, shift left, shift left, shift left, shift left, shift left, shift up, shift left, shift left, shift left, shift down, zoom fit, rotate right, rotate right, shift up, shift down, shift left, shift right, zoom in, shift left, shift left, shift left, shift down, shift down, shift down, shift down, shift down, shift right, shift down, shift right, shift down, shift up, shift up, shift right, shift up, shift up, shift right, shift right, shift up, shift right, shift up, shift right, shift up, shift up, shift up, shift up, shift up, shift left, shift left, shift up, shift left, shift left, shift left, shift left, shift left, shift up, zoom fit, load, rotate right, load, rotate left, load, zoom fit, load, load

## ◆ 測試樣本二

⇒ 相關檔案: image2.dat , cmd2.dat , out\_golden2.dat

⇒ 影像資料:

ff	36	e7	f0	55	32	75	42	32	03	12	75
18	20	57	30	eb	af	ec	11	af	ef	ee	df
61	49	93	22	67	a0	05	c5	a0	f2	f9	c6
28	44	62	66	cc	76	97	79	76	63	56	43
56	28	09	ff	40	18	80	33	18	83	45	9f
e6	f0	e9	ea	87	dd	ed	95	dd	ea	ec	ef
78	d4	d3	bb	f4	77	52	c3	77	63	42	57
c4	aa	b5	92	98	ee	00	a9	ee	dd	df	fa
80	83	85	88	89	75	99	af	be	c0	ca	fl

⇒ 指令輸入順序

load, rotate right, rotate left, rotate left, rotate right, load, zoom in, zoom in, zoom fit, zoom fit, load, rotate right, zoom in, zoom in, zoom fit, zoom fit, rotate left, rotate left, zoom in, zoom in, zoom fit, zoom fit, load, zoom fit, shift up, shift down, shift down, shift left, shift right, shift right, zoom in, shift right, shift down, shift right, shift down, shift down, shift right, shift right, shift right, zoom fit, zoom in, shift left, shift left, shift left, shift left, shift left, shift up, shift up, shift up, shift up, shift up, zoom fit, rotate left, shift up, shift down, shift left, shift right, zoom in, shift right, shift right, shift right, shift up, shift up, shift up, shift right, shift up, shift up, shift left, shift left, shift left, shift left, shift left, shift up, shift left, shift left, shift left, shift down, zoom fit, rotate right, rotate right, shift up, shift down, shift left, shift right, zoom in, shift left, shift left, shift left, shift down, shift down, shift down, shift down, shift right, shift down, shift right, shift down, shift up, shift up, shift right, shift up, shift up, shift right, shift right, shift up, shift right, shift up, shift right, shift up, shift up, shift up, shift up, shift left, shift left , shift up , shift left, shift left, shift left, shift left, shift left, shift left, shift up, zoom fit, load, rotate left, load, rotate right, load, zoom fit, load, load

## 附錄 D 評分用檔案

評分所須檔案可分為四部份：(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檔擺放進來，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 的格式如圖十五所示。(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檔檔名為report.001，依此類推)

表五、評分用檔案

<b>RTL category</b>		
<i>Design Stage</i>	<i>File</i>	<i>Description</i>
N/A	report.xxx	design report
RTL Simulation	*.v or *.vhd	Verilog (or VHDL) synthesizable RTL code
<b>Gate-Level category</b>		
<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	SDF timing information generated by Synopsys Design Compiler
	*_syn.ddc	design database generated by Synopsys Design Compiler

隊號(Team number): 999999

--- RTL category---

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

RTL 檔案名稱(RTL filename): lcd\_ctrl.v, sub1.v, sub2.v

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

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

gate-level sdf filename: lcd\_ctrl\_syn.sdf

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

simulation clock period: 100ns

-----注意事項(annotation)-----

(其餘注意事項依各參賽隊伍的需求填寫)

圖十五、report.000 的範本

## 附錄 E 檔案上傳

所有包含於如附錄 D 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 D 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**。

ftp 的帳號和密碼在 4 天前已用 email 寄給各參賽者。若有任何問題，請聯絡 CIC

FTP site1 (台灣大學)：iccftp.ee.ntu.edu.tw (140.112.20.92)

FTP site2 (新竹晶片中心)：iccftp.cic.org.tw (140.126.24.18)

FTP site3 (南區晶片中心)：iccftp2.cic.org.tw(140.110.117.9)

5. 若你須要繳交更新版本，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。