

2017 University/College IC Design Contest

Cell-Based IC Design Category for Graduate Level

LED Display Controller

1.問題描述

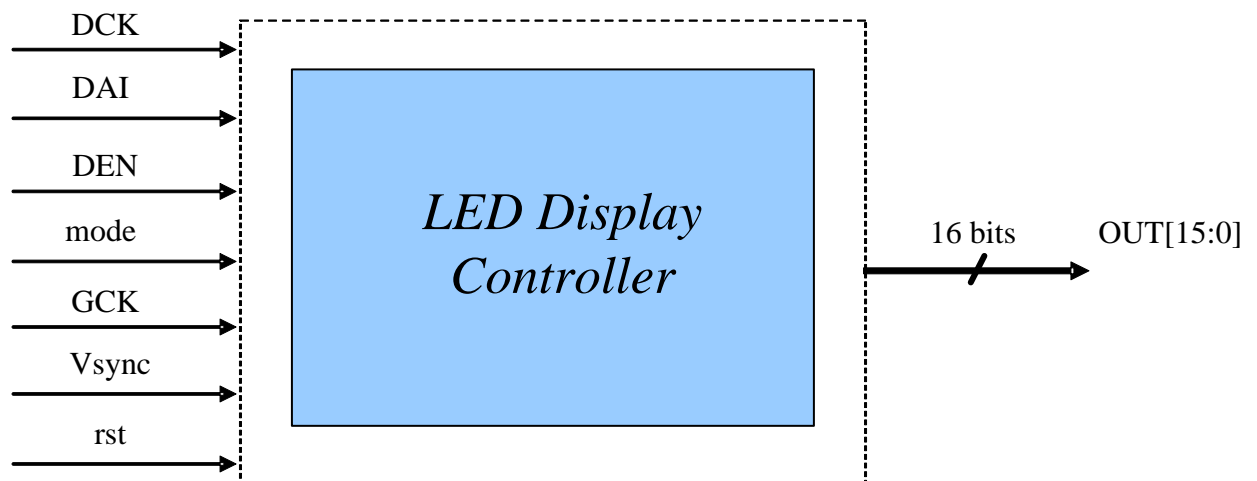
請完成一 LED Display Controller(後文以 **LEDDC 表示**)的電路設計。此電路可接收多張 32x16 像素(Pixels)的 **16bits** 灰階影像(Gray Level Image)訊號，依據該 Pixel 數值輸出 PWM 訊號，用以控制 **32x16 LED Array** 大小之每顆 LED 亮滅時間，即可形成每個畫素之色階。有關 LEDDC 詳細規格將描述於後。 本電路各輸入輸出信號的功能說明，請參考表一。每個參賽隊伍必須根據下一節所給的設計規格及附錄 A 中的測試樣本完成設計驗證。

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/icc2017/bgc/icc2017cb.tar***，請執行以下指令取得測試樣本：

```
tar xvf /usr/cad/icc2017/bgc/icc2017cb.tar
```

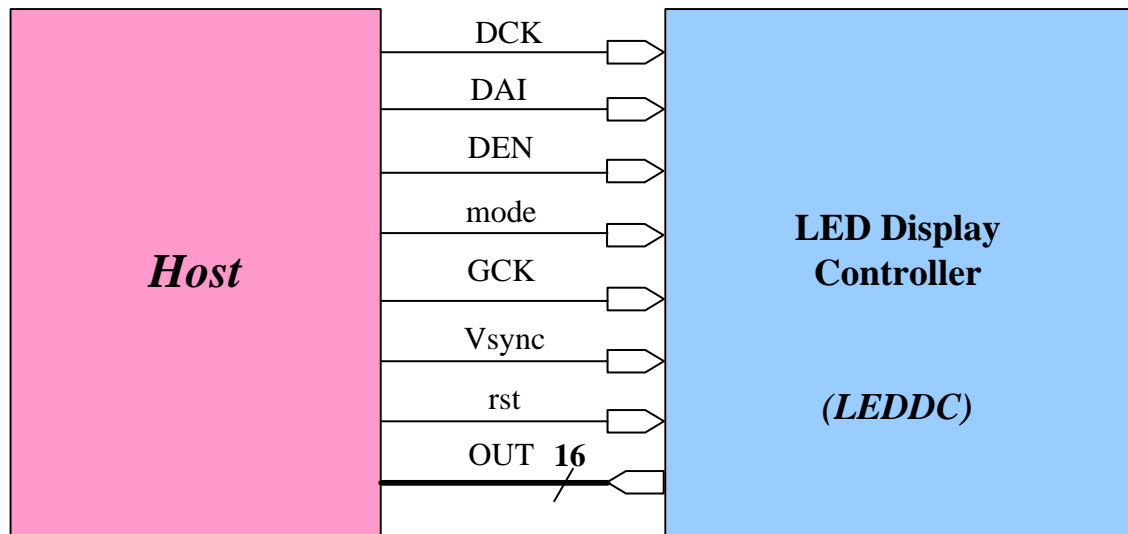
軟體環境及設計資料庫說明請參考附錄 F 與附錄 G。



圖一、LED Display Controller 之方塊圖

2.設計規格

2.1 系統方塊圖



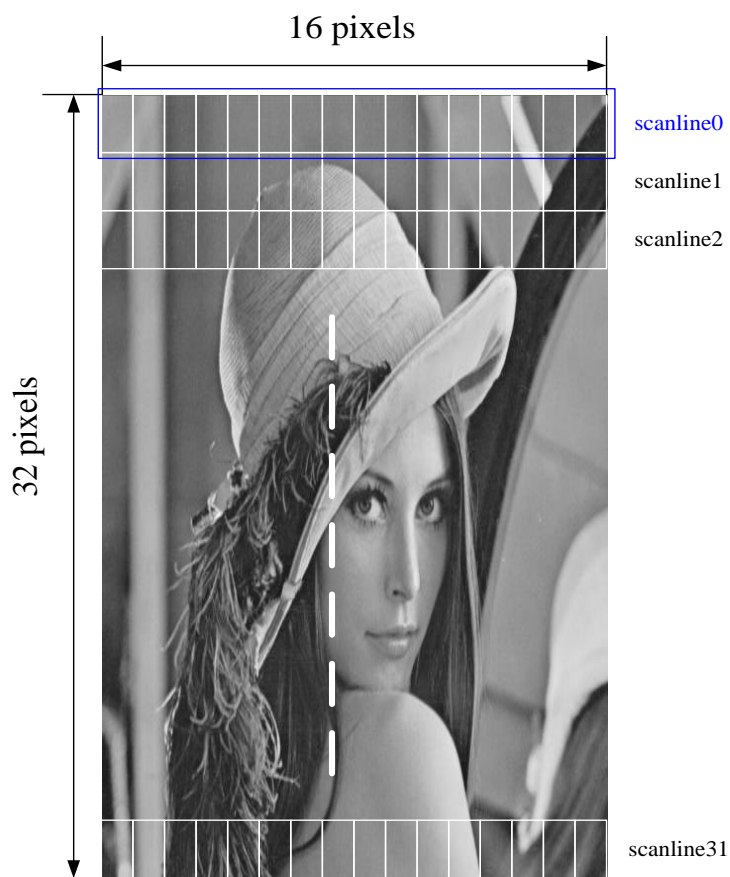
圖二、系統方塊圖

2.2 輸入/輸出介面

表 1 -輸入/輸出訊號

Signal Name	I/O	Width	Simple Description
DCK	I	1	Data Clock，其週期固定為 1300ns，因系統規格需求請勿變動，平時沒資料傳輸時 DCK 會為 0，即 Clock 訊號會停。DCK 與 GCK 兩者頻率與出現時間並無相關。 (註: Host 端採 DCK ”正”緣時送資料。)
GCK	I	1	Grayscale Clock，其週期固定為 4.50ns，因系統規格需求請勿變動，平時沒資料傳輸時 GCK 會為 0，即 Clock 訊號會停。DCK 與 GCK 兩者頻率與出現時間並無相關。 (註: Host 端採 GCK ”正”緣時送資料。)
rst	I	1	Reset Signal。(Active 為 1)
DAI	I	1	Serial Data Input，作為整張 frame 之 Pixels 訊號輸入用，其詳細輸入方式請參考 2.3.1。
DEN	I	1	Data Enable，用以告知從 DAI 輸入的 Pixel 訊號，若為 1 表示目前輸入訊號為有效的輸入，反之則為 0。

mode	I	1	本系統需製作 30fps、60fps 兩種規格。 mode 為 0 ，表示系統會以 30fps 的規格運作、 mode=1 表示系統會以 60fps 規格運作，兩者不會同時進行。
OUT	O	16	PWM Control Signal，本題 OUT 有 16 隻腳位，因此每次可輸出 16 個 Pixels 訊號值，但要以 PWM 訊號輸出。詳細輸出方式請參考 2.3.2。
Vsync	I	1	<p>Output Valid Signal & Check，當 GCK 正緣一來，若看到 Vsync 訊號為 1，Host 端就會開始比對 OUT[0] ~ OUT[15] 共計 16 個 PWM 的訊號，是否正確。詳細比對方式請參考 2.3.2.1 及 2.3.2.2。</p> <p>注意：該訊號不是由 LEDDC 電路自由控制，而是由 Host 端，透過 Vsync 訊號直接指定 LEDDC 電路必須在 Vsync 為 1 期間，輸出 PWM 訊號。沒在指定時間輸出，都算是錯誤的。</p>

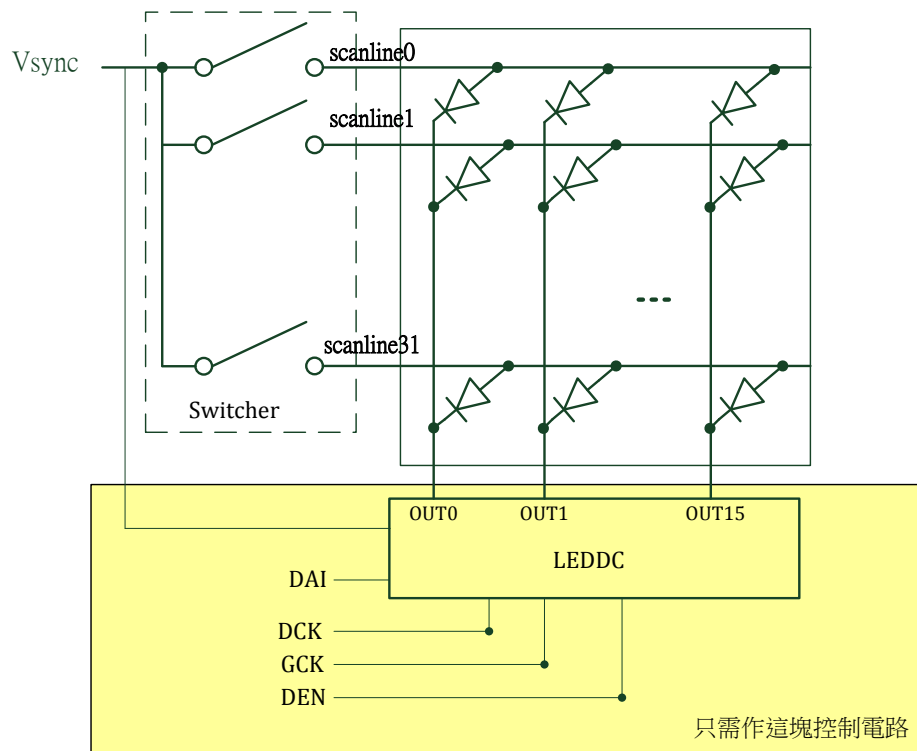


一個frame的範例

[illegible]

對應左圖各Pixel數值(16進制)

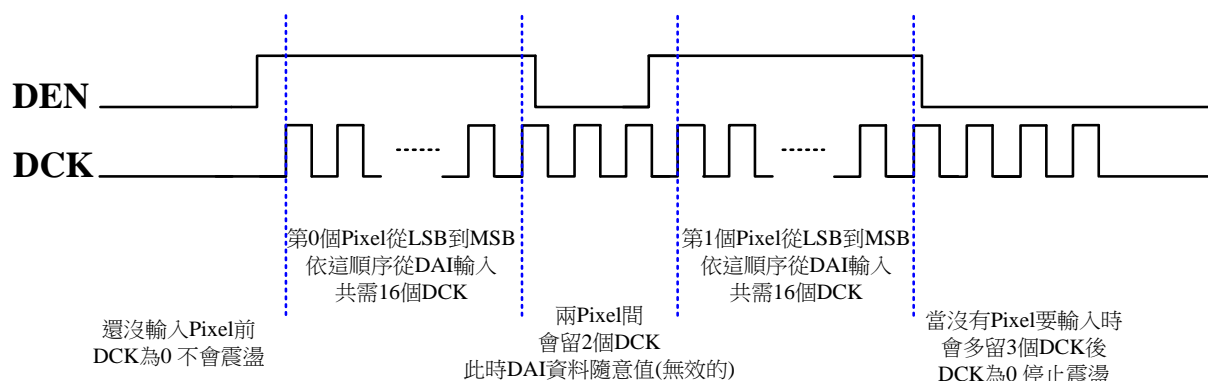
圖三、32x16 灰階影像訊號範例



圖四、將 LEDDC 電路之 OUT 輸出至 32x16 LED Array 示意圖

2.3 系統描述

圖三為一張 32x16 大小的灰階影像圖，這樣一張圖定義為一個 frame，圖中一格白色方框稱為一個 Pixel，每個 Pixel 用 16bits 表示，其訊號高低代表著該點影像的訊號的強弱，圖三右圖之數值(在此以十六進制表示)只是範例。圖三左圖之一整排白色方框，共計 16 個 Pixels，在此以藍色方框圈起，代表一條 scanline，因此本題一個 frame 共有 32 個 scanline 如圖三左圖所示，LEDDC 電路功能就是要將 scanline0 ~ scanline31 的 Pixels 訊號透過圖四之 DAI 輸入腳位依序輸入至 LEDDC 電路，每當一條 scanline 的 Pixel 訊號全部輸入完成後，Vsync 會在適當時機將其訊號拉為 High，告知 LEDDC 電路將該條 scanline 的 16 個 Pixel 訊號要在 Vsync 為 High 的期間內把 PWM 的數值藉由 LEDDC 電路之 OUT[0:15]，共計 16bits，因此一次可以輸出 16 個 Pixels 其 PWM 的訊號輸出至 32x16 LED Array，本題我們只需要做圖四黃色方框的電路即可，完成後會接續下一條 scanline1，...，直到 scanline31 都輸出後，即完成第 0 個 frame 的處理，後續還會以相同動作繼續處理第 1 個 frame、第 2 個 frame、....，本題考量到模擬時間問題，僅需處理四個 frames 即可。詳細電路運作方式，將描述於後。



圖五、LEDDC 電路輸入方式

2.3.1 LEDDC 輸入方式

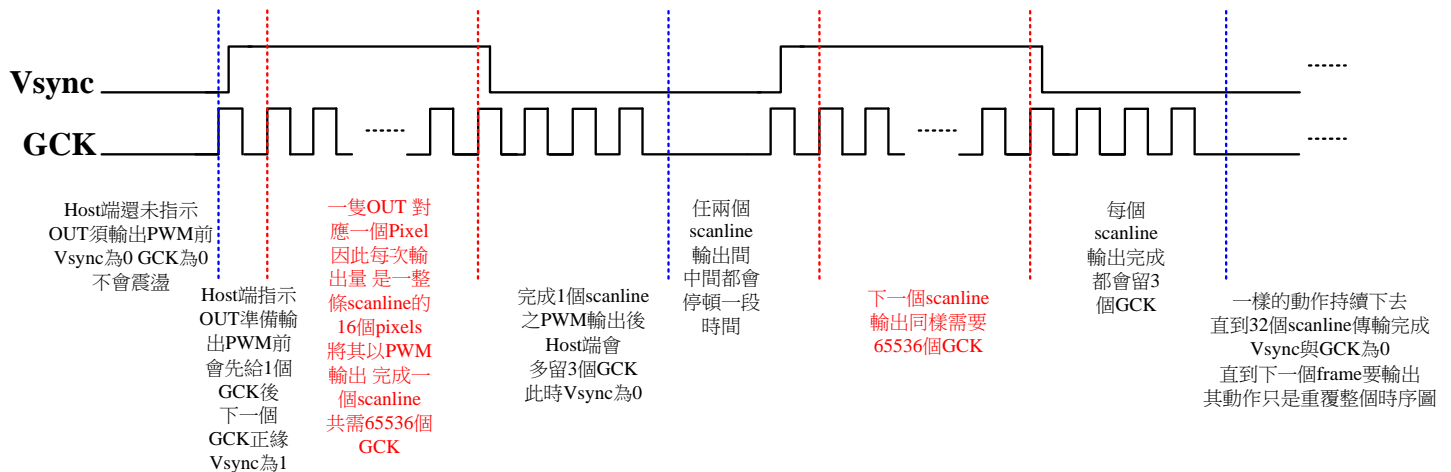
LEDDC 輸入方式說明如下：

- A. LEDDC 電路模擬時間軸 10ns 時，Host 端會先給 rst 為 High 約 1300ns 之長(即 1 個 DCK 週期長度)，但此時 DCK 為 0，並未有任何振盪。

提示：參賽者若有使用到 rst 訊號，請以非同步 Reset 設計電路。

- B. 圖五，還沒輸入 Pixel 前 DCK 為 0，都不會有震盪。
- C. 當輸入某個 scanline 之 Pixel 訊號時，如圖三，輸入順序為該 scanline 從左到右的 Pixel 依序輸入，然而 DAI 為 Serial Input，因此每個 Pixel 輸入方式為該訊號值的 LSB 開始輸入，一直到 MSB 為止，共計 16 個 DCK 週期，這期間 DEN 會一直為 High，如圖五。若以圖三為例，scanline0 第 0 個 Pixel 為 $(5454)_{16}$ ，DAI 輸入順序則為 0、0、1、0、1、0、1、0、0、0、1、0、1、0、1、0 共計 16 個 DCK 週期傳送完閉。
- D. 圖五，DAI 準備輸入下一個 Pixel 前，兩 Pixel 間會留 2 個 DCK，此時 DAI 資料為隨意值，即無效的輸入資料。

- E. 圖五，為連續兩個 Pixel 輸入之時序圖，無論是同一個 scanline 的連續兩個 Pixel 訊號或跨不同 scanline 的連續兩個 Pixel 訊號，其中間間隔方式皆相同。
- F. 圖五，當輸入到最後一個 Pixel 後（即 scanline31 的最後一個 Pixel），整個 frame 都輸入完畢，系統會再多留三個 DCK，隨後 DCK 就一直為 0，直到下一個 frame 要輸入時，圖五的時序圖將再重演。



圖六、30fps 規格：LEDDC 電路輸出方式

2.3.2 LEDDC 輸出方式

本題 LEDDC 有兩種規格：30fps、60fps，mode 若為 0 表示以 30fps 方式運作，mode 若為 1 表示以 60fps 方式運作，兩者完全獨立運作，不會同時進行或混和進行。圖八為 LEDDC-30fps 規格，從圖可看出一個 frame 的輸入與輸出比對，會壓在約 1/30s 完成，圖九為 LEDDC-60fps 規格，從圖可看出一個 frame 的輸入與輸出比對會壓在約 1/60s 完成。關於這兩種規格詳細輸出方式，將描述於下。

2.3.2.1 LEDDC - 30fps 電路輸出方式

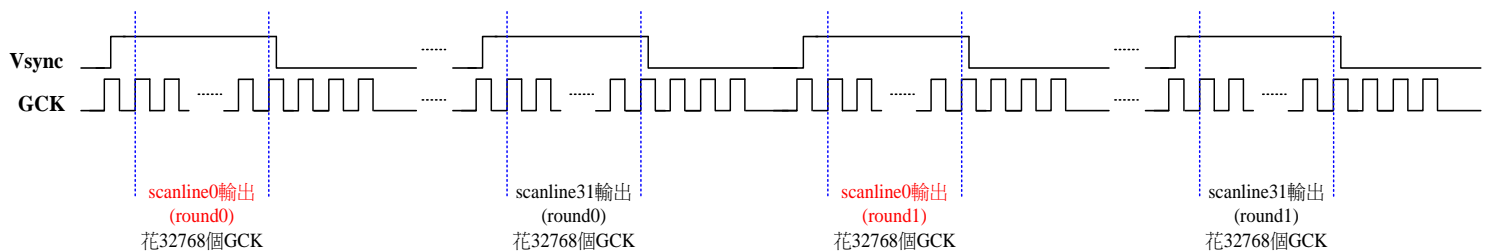
LEDDC-30 fps 輸出方式說明如下：

- A. 圖六，Host 端還未指示 OUT 需輸出 PWM 前 Vsync 為 0、GCK 為 0，不會震盪。
- B. 圖六，Host 端指示 OUT 準備輸出 PWM 訊號前，會先給 1 個 GCK，而下一個 GCK 正緣來時 Vsync 便開始為 1，此時送出第 0 個 GCK 週期的 PWM 訊號。
- C. 圖三，一條 scanline 有 16 個 Pixels 值，因此 scanline 最左 ~ 最右的 pixel，分別會以 OUT[0]~OUT[15] 負責各別輸出 PWM 訊號。PWM 訊號是指，例如：圖三 scanline0 最左邊第 0 個 Pixel 訊號值假設為 $(5454)_{16}$ ，相當於 $(21588)_{10}$ ，因此該 Pixel 的輸出固定由 OUT[0] 單一腳位，負責輸出總共 21588 個 GCK 週期皆為 1，剩餘 $(65536-21588)$ 個 GCK 週期輸出為 0。同理，圖三 scanline0 最左邊數來第 1 個 Pixel 訊號值假設為 $(5656)_{16}$ ，相當於 $(22102)_{10}$ ，因此該 Pixel 的輸出固定由 OUT[1] 單一腳位，負責輸出總共 22102 個 GCK 週期皆為 1，剩餘 $(65536-22102)$ 個 GCK 週期輸出為 0，即為所求。
- D. 圖六，每個 scanline 有 16 個 Pixels，分別交由 OUT[0] ~ OUT[15] 腳位個別輸出，因此在一個

Vsync 為 1 期間(共有 65536 個 GCK，該期間 Vsync 都會一直維持 1)，OUT[0] ~ OUT[15]腳位就要將各 Pixel 的值以輸出 PWM 訊號方式輸出。

- E. 圖六，每當完成一個 scanline 的 PWM 輸出後，會多給 3 個 GCK。
- F. 圖六，任兩個連續 scanline 輸出間，GCK、Vsync 都會為 0 一段時間。
- G. 圖六，下一個 scanline 輸出，其輸出方式與時序圖，皆與前面敘述相同。
- H. 圖六，不管哪一個 scanline 輸出，輸出結束後，都會留 3 個 GCK 提供給需要的參賽者善加利用。
- I. 圖六，當 32 個 scanline 都輸出後，該 1/30s 區間的輸出任務結束，GCK、Vsync 都會一直維持為 0，直到下一個 1/30s 時，Host 端有所指示時，會繼續以圖六之時序圖再來一次。

註：由於考量到模擬時間問題，本題僅提供 4 張 32x16 的 frames。



圖七、60fps 規格：LEDDC 電路輸出方式

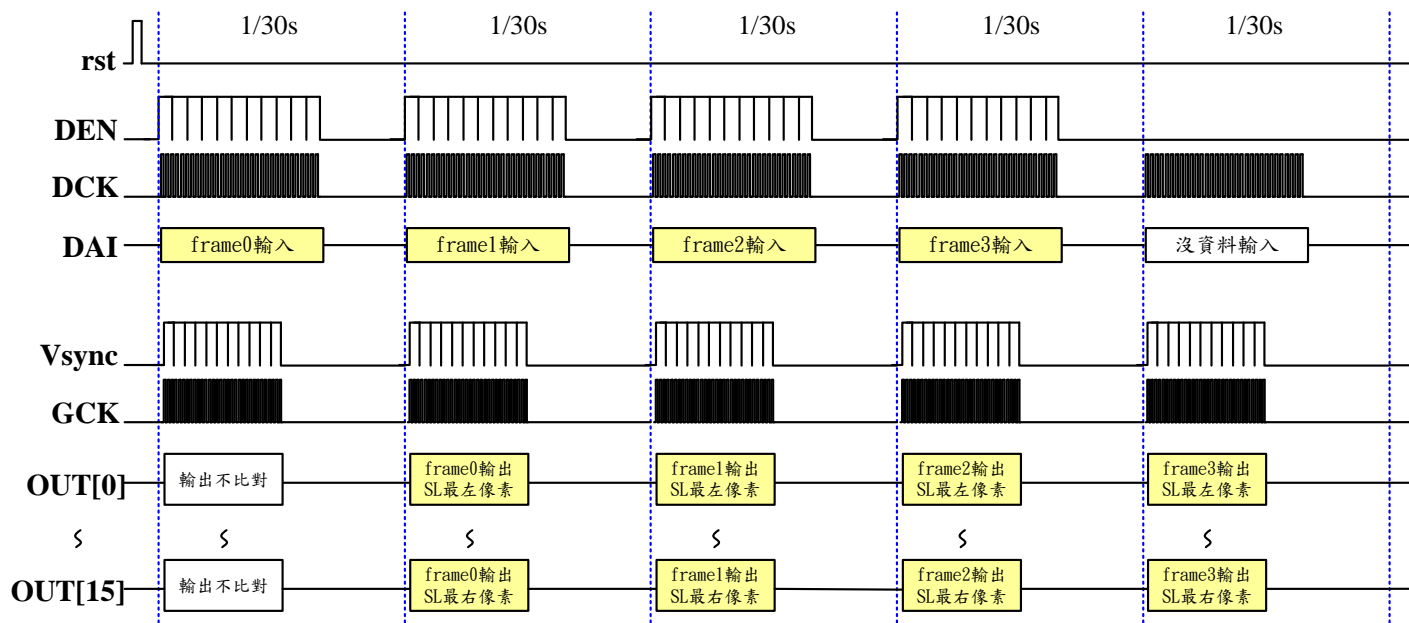
2.3.2.2 LEDDC - 60fps 電路輸出方式

30fps、60fps 兩種規格動作皆相似，因此以下將只針對兩者不同處，予以說明如下：

- A. 圖七，**60fps 規格**其顯示的變動頻率快兩倍，因此任一個 Pixel 訊號值的輸出都要分兩回合(即 round0、round1)送出，其輸出流程順序是，scanline0 的 round0、scanline1 的 round0、...、scanline31 的 round0、scanline0 的 round1、scanline1 的 round1、...、scanline31 的 round1。
- B. 圖七，由於每個 Pixels 訊號值會被分兩回合輸出，因此每個 scanline 單一回合的輸出週期只剩一半，例如：scanline0 的 round0，僅有 32768 個 GCK 要輸出，scanline0 的 round1，也僅有 32768 個 GCK 要輸出。
- C. 圖七，由於每個 Pixels 訊號值會被分兩回合輸出，其 Pixel 訊號值也將被減半作輸出，例如：假設一個 Pixel 訊號值為 $(9)_{10}$ ，因為該值為奇數，因此在 round0 期間，先輸出 5，即要在 Vsync 為 1 的 32768 個 GCK 期間輸出 1，共計 5 個 GCK，剩餘的 $(32768-5)$ 個 GCK 期間輸出 0，然後在 round1 期間，輸出剩餘的數值 4，即要在 Vsync 為 1 的 32768 個 GCK 期間輸出 1，共計 4 個 GCK，剩餘的 $(32768-4)$ 個 GCK 期間輸出 0，即為所求。若 Pixel 數值為偶數，例如：數字 8，round0 輸出 4、round1 輸出 4。
- D. 圖七，無論任兩個 scanline 或不同 round 間的 scanline，其中間的間隔處，都會是固定 3 個 GCK 後一段停頓時間。
- E. 圖七，當 32 個 scanline 分兩回合都輸出後，該 1/60s 區間的輸出任務結束，GCK、Vsync 都會一直維持為 0，直到下一個 1/60s 時(即下一個 frame)，Host 端有所指示時，會繼續重複圖七之時序圖。

2.4 LEDDC 電路時序規格

2.4.1 LEDDC – 30fps 電路之時序圖(mode=0)

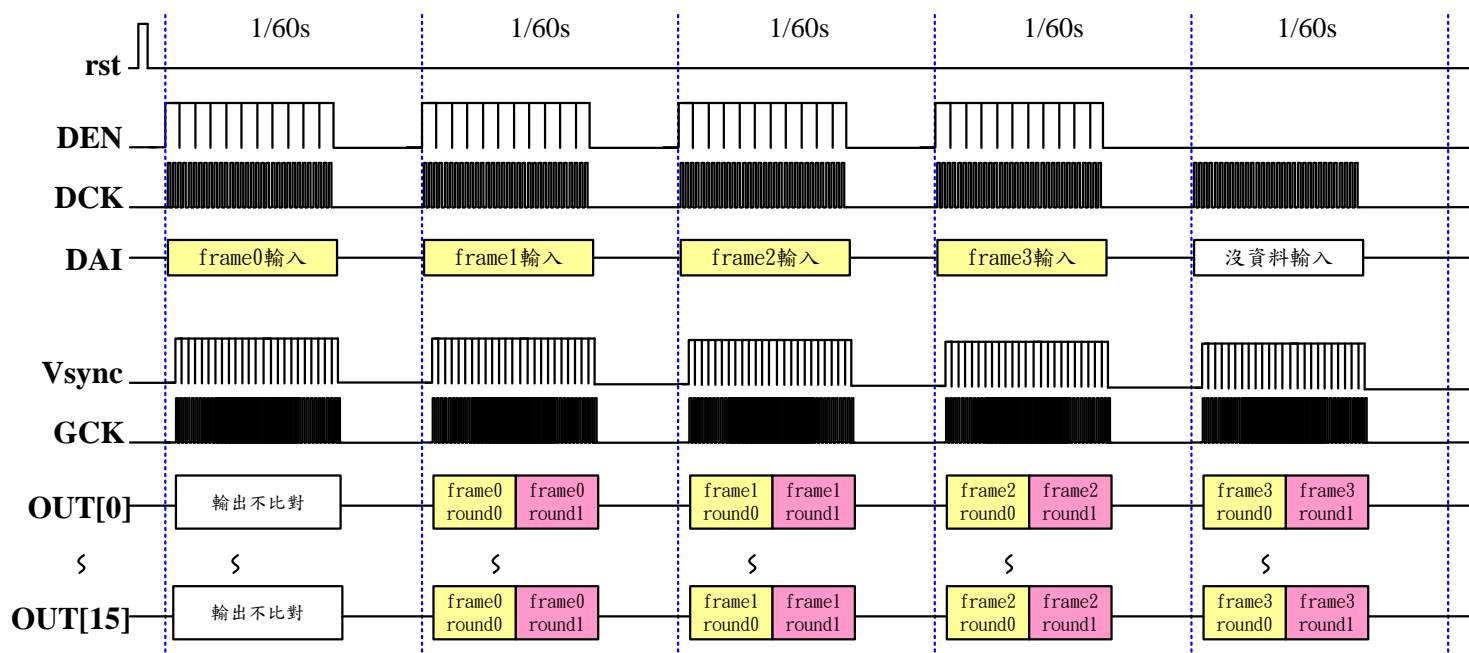


圖八、LEDDC – 30fps 電路時序圖

LEDDC-30 fps 電路時序圖說明如下：

- 圖八，LEDDC 電路模擬時間軸 10ns 時，Host 端會先給 rst 為 High 約 1300ns 之長(即 1 個 DCK 週期長度)，但此時 DCK 為 0，並未有任何振盪。
- 圖八，第 0 個 1/30s，Host 端會依序輸入第 0 個 frame 的所有 Pixels 訊號值。第 0 個 1/30s，雖然 GCK、Vsync 都會與第 1 個以後的 frame 區間訊號相同，但 Host 端並不會做任何輸出比對，請注意。
- 圖八，第 1 個 1/30s，Host 端會依序輸入第 1 個 frame 的所有 Pixels 訊號值。第 1 個 1/30s，開始會作第 0 個 frame 的輸出比對。後續的 1/30s，動作都相同，依此類推。
- 圖八，最後一個 1/30s，Host 端因已送完四個 frames 的訊號，因此 DCK 雖然會震盪，但沒有資料輸入，DEN 因此為 0，DAI 無資料輸入。最後一個 1/30s，開始比對第 3 個 frame 的輸出，比對完成後，即完成 30fps 規格之模擬，本模擬結束。

2.4.2 LEDDC – 60fps 電路之時序圖(mode=1)



圖九、LEDDC – 60fps 電路時序圖

LEDDC-60 fps 電路時序圖說明如下：

- 圖九，LEDDC 電路模擬時間軸 10ns 時，Host 端會先給 rst 為 High 約 1300ns 之長(即 1 個 DCK 週期長度)，但此時 DCK 為 0，並未有任何振盪。
- 圖九，第 0 個 1/60s，Host 端會依序輸入第 0 個 frame 的所有 Pixels 訊號值。第 0 個 1/60s，雖然 GCK、Vsync 都會與第 1 個以後的 frame 區間訊號相同，但 Host 端並不會做任何輸出比對，請注意。
- 圖九，第 1 個 1/60s，Host 端會依序輸入第 1 個 frame 的所有 Pixels 訊號值。第 1 個 1/60s，開始會作第 0 個 frame 的輸出比對，其輸出比對方式，先 frame0 的 round0 所有 scanline，再 frame0 的 round1 所有 scanline。後續的 1/60s，動作都相同，依此類推。
- 圖九，最後一個 1/60s，Host 端因已送完四個 frames 的訊號，因此 DCK 雖然會震盪，但沒有資料輸入，DEN 因此為 0，DAI 無資料輸入。最後一個 1/60s，開始比對第 3 個 frame 的輸出，比對完成後，即完成 60fps 規格之模擬，本模擬結束。

2.4.2 Two Port 記憶體- SRAM_512x16、SRAM_256x16 之時序圖

Two Port SRAM 有特殊功能，即同一個 Clock 週期內，可以讀一筆資料及寫一筆資料之特性，本題提供兩種 Two Port 記憶體規格，分別是 SRAM_512x16、SRAM_256x16，請自行斟酌要使用哪一種記憶體或兩種都用，也可以兩者都不使用，由參賽者自行斟酌。有關 Two Port 記憶體讀、寫之時序圖，請自行參考 sram_512x16.pdf、sram_256x16.pdf 檔案。

3. 評分標準

評分方式會依設計完成程度，分成 A、B、C、D 四種等級，排名順序為 A>B>C>D，評分項目僅面積單一項目，主辦單位會依此面積大小作為同等級之評分高低。注意，模擬時 DCK、GCK 週期時間不可變更，因該速度是本系統要求之規格，電路效能務必達成指定的速度。

☆ 評分項目：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} {551.580 550.740}
```

=> Area = 551.580 x 550.740 = 303777.1692 μm^2

2. Encounter Report Area 範例:

```
encounter > analyzeFloorplan
```

```
encounter 2> analyzeFloorplan  
Start to collect the design information.  
Build netlist information for Cell LEDDC.  
Finished collecting the design information.  
Average module density = 1.000.  
Density for the design = 1.000.  
    = stdcell_area 79981 sites (135760  $\mu\text{m}^2$ ) / alloc_area 79981 sites (135760  $\mu\text{m}^2$ ).  
Pin Density = 0.122.  
    = total # of pins 23994 / total Instance area 196550.  
***** Analyze Floorplan *****  
Die Area( $\mu\text{m}^2$ )           : 333039.25  
Core Area( $\mu\text{m}^2$ )           : 314558.83  
Chip Density (Counting Std Cells and MACROs and IOs): 84.316%  
Core Density (Counting Std Cells and MACROs): 89.270%  
Average utilization       : 100.000%  
Number of instance(s)     : 14105  
Number of Macro(s)        : 2  
Number of IO Pin(s)       : 23  
Number of Power Domain(s) : 0  
***** Estimation Results *****  
*****
```

=> Area = 333039.25 μm^2

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，說明如下：

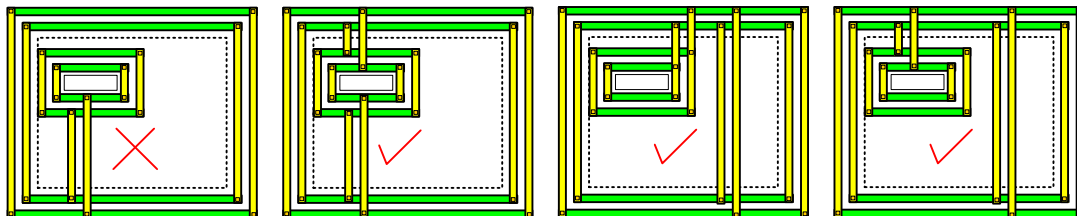
◇ **等級 A：** 務必達成下述三項要求

- a、 30fps、60fps 兩者規格皆功能正確，即 RTL 模擬結果正確。
- b、 30fps、60fps 兩者規格皆完成 Synthesis，且 Gate-Level Pre-layout Simulation 結果正確。
- c、 30fps、60fps 兩者規格皆完成 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. 內建的所有記憶體 SRAM，其 VDD、VSS Pin 務必要連接至 Core Power Ring，寬度請各設定為 **2um**。
- v. **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)
- vi. **務必要加 Power Rail (follow pin)**。
- vii. Core Filler 務必要加。
- viii. APR 後之 GDSII 檔案務必產生。
- ix. 完成 APR，DRC/LVS **完全無誤**(見附錄 C 說明)。

註: Power Stripe 指的是直接穿過 core area 的 power line，見下圖



等級 A 之評分方法：

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

例如：

本範例(Encounter)為，

$$\text{Score} = \text{Area} = 333039.25$$

註: **Score 越小者，同級名次越好!**

☆ **等級 B：** 務必達成下述兩項要求

- a、 30fps、60fps 兩者規格皆功能正確，即 RTL 模擬結果正確。
- b、 30fps、60fps 兩者規格皆完成 Synthesis，且 Gate-Level Pre-layout Simulation 結果正確。

等級 B 之評分方法：

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

註： Area 係以 Design Compiler 之 Cell Area 為主，Score 越小者，同級名次越好！

☆ **等級 C：** 務必達成下述兩項要求

- a、 30fps 單一規格功能正確即可，即 RTL 模擬結果正確。
- b、 30fps 單一規格完成 Synthesis，且 Gate-Level Pre-layout Simulation 結果正確。

等級 C 之評分方法：

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

註： Area 係以 Design Compiler 之 Cell Area 為主，Score 越小者，同級名次越好！

☆ **等級 D：** 未達成前三等級者，成績計算方式為 30fps 單一規格之 RTL Simulation 為主(60fps 規格不必製作)，比對結果之 error 總數量越少者，分數越高。

$$\text{Score} = \text{Total error of 30fps RTL Simulations}$$

註： 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。

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
LEDDC.v	本題之設計檔，已包含系統 Input/Output Port 之宣告。
testfixture_30fps.v testfixture_60fps.v	本題共計兩個 TestBench，分別用於 30fps、60fps 兩種規格之模擬。請勿修改 DCK、GCK 週期時間。
pattern.dat	作為 LEDDC 電路模擬時之四個 frames 的輸入訊號。 註：該檔案已被加入到 TestBench。
golden_30fps.dat golden_60fps.dat	30fps 規格之 Golden 結果檔案。 60fps 規格之 Golden 結果檔案。 註：此兩檔案已被加入到 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。 註：無論合成或 APR，只需使用 worst case library。
LEDDC_DC.sdc	Design Compiler 作合成之 Constraint 檔案，該檔案僅供參考，參賽者可依電路實際情形，調整 Constraints。 注意：請勿修改 cycle 的設定、環境相關參數請勿更改。
LEDDC_ICC.sdc	IC Compiler 作 Layout 之 Constraint 檔案，該檔案僅供參考，參賽者可依電路實際情形，調整 Constraints。 注意：請勿修改 cycle 的設定、環境相關參數請勿更改。
LEDDC EDI.sdc	Encounter 作 Layout 之 Constraint 檔案，該檔案僅供參考，參賽者可依電路實際情形，調整 Constraints。 注意：請勿修改 cycle 的設定、環境相關參數請勿更改。

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

```
`timescale 1ns/10ps
module  LEDDC( DCK, DAI, DEN, GCK, Vsync, mode, rst, OUT);
input      DCK;
input      DAI;
input      DEN;
input      GCK;
input      Vsync;
input      mode;
input      rst;
output [15:0]  OUT;

endmodule
```

2. 本題 30fps、60fps 兩種規格，皆使用同一組 Pattern 作測試。假若比賽過程中無任何舞弊行為，主辦單位便**不會有隱藏的 Pattern 作測試**。
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      "./LEDDC_syn.sdf"
`ifdef SDF
    initial $sdf_annotate( SDFFILE, u_LEDDC );
`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，在 UNIX 下執行下面指令

```
> ncverilog  +ncmaxdelays  testfixture_30fps.v  LEDDC_syn.v  sram_512x16.v
              tsmc13_neg.v  +define+SDF  +access+r
```

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

四個 frames 之 Pixel 訊號，已存於 pattern.dat，共計 2048 個 Pixels(說明：32 x 16 x 4 個 frames)，其內容如下：

```
5454 //frame_00 scanline_00 => pixel_00 = 21588
5656 //frame_00 scanline_00 => pixel_01 = 22102
5555 //frame_00 scanline_00 => pixel_02 = 21845
5353 //frame_00 scanline_00 => pixel_03 = 21331
5858 //frame_00 scanline_00 => pixel_04 = 22616
5252 //frame_00 scanline_00 => pixel_05 = 21074
4646 //frame_00 scanline_00 => pixel_06 = 17990
6565 //frame_00 scanline_00 => pixel_07 = 25957
7A7A //frame_00 scanline_00 => pixel_08 = 31354
8888 //frame_00 scanline_00 => pixel_09 = 34952
9393 //frame_00 scanline_00 => pixel_10 = 37779
9494 //frame_00 scanline_00 => pixel_11 = 38036
9595 //frame_00 scanline_00 => pixel_12 = 38293
9696 //frame_00 scanline_00 => pixel_13 = 38550
8A8A //frame_00 scanline_00 => pixel_14 = 35466
7777 //frame_00 scanline_00 => pixel_15 = 30583
4E4E //frame_00 scanline_01 => pixel_00 = 20046
5353 //frame_00 scanline_01 => pixel_01 = 21331

:
```

註：資料左側為十六進制，資料右側以註解方式說明該筆 Pixel 屬於第幾個 frame、第幾個 scanline、對應的十進制數值為何，以方便參賽者作 Debug。

30fps 規格之標準解答已存於 golden_30fps.dat，其局部輸出說明如下：

```
5454_5656_5555_5353_5858_5252_4646_6565_7A7A_8888_9393_9494_9595_9696_8A8A_7777 //frame_00 scanline_00
4E4E_5353_4E4E_5050_4F4F_4C4C_4242_5D5D_7B7B_8C8C_9494_9393_9595_9696_8888_7676 //frame_00 scanline_01
5353_5050_5353_4040_4040_4545_3D3D_5F5F_7676_8C8C_9393_9696_9696_9898_8787_7777 //frame_00 scanline_02
9595_5353_5353_5959_4E4E_4545_3B3B_5858_7B7B_8888_9898_9595_9999_9898_8888_7474 //frame_00 scanline_03
5454_5858_5252_5454_4E4E_4A4A_4040_5C5C_7777_8C8C_9494_9999_9B9B_9999_888B_7474 //frame_00 scanline_04
5757_5959_5252_4D4D_4C4C_4444_3B3B_5555_7C7C_8C8C_9393_9797_9D9D_9B9B_8F8F_7777 //frame_00 scanline_05
5555_5454_5151_4F4F_4747_4545_3535_5454_7878_8E8E_9595_9797_9B9B_9999_8D8D_7A7A //frame_00 scanline_06
5757_4E4E_4D4D_4848_4444_3D3D_3636_4B4B_7676_8C8C_9797_9999_9A9A_9B9B_8E8E_7B7B //frame_00 scanline_07
4A4A_4E4E_4747_4848_4747_3838_3131_4B4B_7272_8C8C_9595_9797_9A9A_9C9C_9090_7878 //frame_00 scanline_08
4C4C_4C4C_4545_4545_4242_3F3F_3131_4F4F_7A7A_888B_9494_9595_9898_9D9D_8F8F_7D7D //frame_00 scanline_09
4747_4343_4242_4040_3D3D_3A3A_2F2F_4C4C_7474_8888_9393_9797_9A9A_9A9A_8F8F_7B7B //frame_00 scanline_10
3F3F_4242_3E3E_4040_3A3A_3737_3030_4848_7979_8686_9292_9595_9696_9696_9292_7B7B //frame_00 scanline_11
4747_3D3D_4040_3C3C_3C3C_3232_2B2B_4F4F_7777_8686_9191_9292_9595_9A9A_8B8B_7979 //frame_00 scanline_12
4343_4040_3D3D_3F3F_3C3C_3131_2727_4848_7575_8989_8F8F_9393_9090_9797_8C8C_7878 //frame_00 scanline_13
4848_4444_4040_4343_3838_3030_2828_4A4A_7272_8989_9494_9797_9393_9B9B_8C8C_7C7C //frame_00 scanline_14
4949_4545_4343_3F3F_3737_2E2E_2525_4444_7272_8888_9292_9595_9595_9898_9090_7D7D //frame_00 scanline_15
4949_4444_4545_4242_3B3B_3232_2B2B_4F4F_7979_8989_8F8F_9292_9797_9999_8F8F_7777 //frame_00 scanline_16
4343_4545_4242_4242_3737_3B3B_3838_5D5D_7C7C_8D8D_8F8F_9191_9595_9797_9292_7979 //frame_00 scanline_17
3F3F_4646_3F3F_4848_3B3B_3E3E_4848_6868_7F7F_8B8B_8E8E_9090_9797_9999_9090_7B7B //frame_00 scanline_18
4545_4141_4444_4343_3B3B_3535_4949_6969_7C7C_8989_8F8F_9090_9696_9A9A_9393_7E7E //frame_00 scanline_19

:
```

註：資料左側為十六進制，“_”符號，表示任意兩 Pixel 數值之間格。

註：資料右側提供該筆資料是第幾個 frame、第幾個 scanline 的輸出，以方便參賽者作 Debug。

註：有關 60fps 規格之標準解答，請自行參考 golden_60fps.dat。

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，模擬結果必須滿足本題指定之 Period 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	LEDDC_pr.sdf
Significant digits	2

按 **OK**。

對應指令： write_sdf -version 2.1 LEDDC_pr.sdf

“ File > Export > Write Verilog...”

先按 **Default**

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

按 **OK**。

- ii. 使用 Cadence Encounter 者，執行步驟如下：

在 Encounter 視窗下點選：

“ File → Save → Netlist...”

Netlist File	LEDDC_pr.v
All other options	Default value

按 **OK**。

“ Timing → Extract RC...”，按 **OK**。

“Timing → Write SDF...”

Ideal Clock	Disable
SDF Output File:	LEDDC_pr.sdf

按 **OK**。

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 **OK**。

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

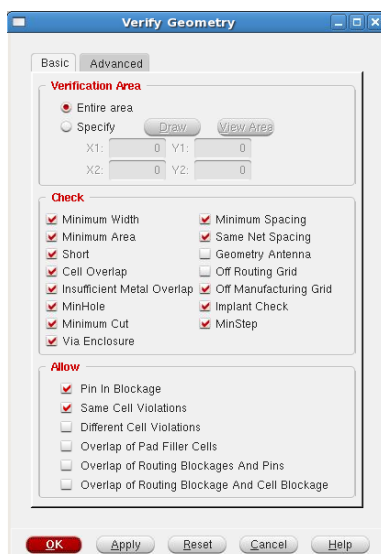
按 **OK**。

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：

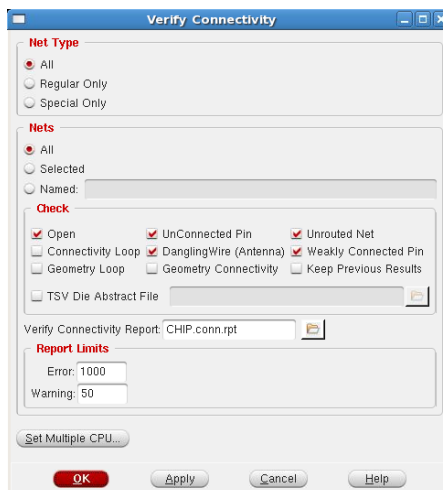
在 Encounter 視窗下點選

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



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 **and .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 LEDDC.v ~/result/
```

```
> cp LEDDC_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	ncverilog
	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/icc2017/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
	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
sram_512x16/		
	sram_512x16.pdf	Memory Data Sheet
	sram_512x16.v	Memory Verilog file
	sram_512x16_slow_syn.db	Memory DB file
	sram_512x16_slow_syn.lib	Memory LIB file
	sram_512x16.vclef	Memory LEF file
	sram_512x16_ant.lef	Memory Antenna file
	sram_512x16.gds	Memory GDSII file
	sram_512x16/	Memory FRAM View
sram_256x16/		同上說明

Design Report Form

登入帳號(login-id)			
RTL category			
Design Stage	Description		File Name
RTL Simulation	使用之 HDL 名稱 (例如：Verilog、VHDL)		
RTL Simulation	RTL 檔案名稱 (RTL Netlist file name)		
Gate-Level category			
Design Stage	Description		File Name
Pre-layout Gate-level Simulation	Gate-Level 檔案名稱 (Gate-Level Netlist file name)		
	Pre-layout sdf 檔案名稱		
Physical category			
Design Stage	Description		File Name or Value
P&R	使用之 P&R Tool (請填入 IC compiler 或 Encounter)		
	設計資料庫檔案名稱 (Library name) (ICC: Milkyway Library Name, Encounter: xxx.enc.dat)		
	DRC 錯誤總數量 (ex: 0 個)		
	LVS 錯誤總數量 (ex: 0 個)		
	佈局檔檔案名稱 (GDSII file name)		
	佈局面積 (Layout Area)	() um ²	
Post-layout Gate-level Simulation	Gate-Level 檔案名稱 (Gate-Level Netlist file name)		
	Post-layout sdf 檔案名稱		
Over All	最後完成之等級? (ex: 等級 A)		
其他說明事項 (Any other information you want to specify: (如設計特點 ...) 如寫不下可寫於背面			