

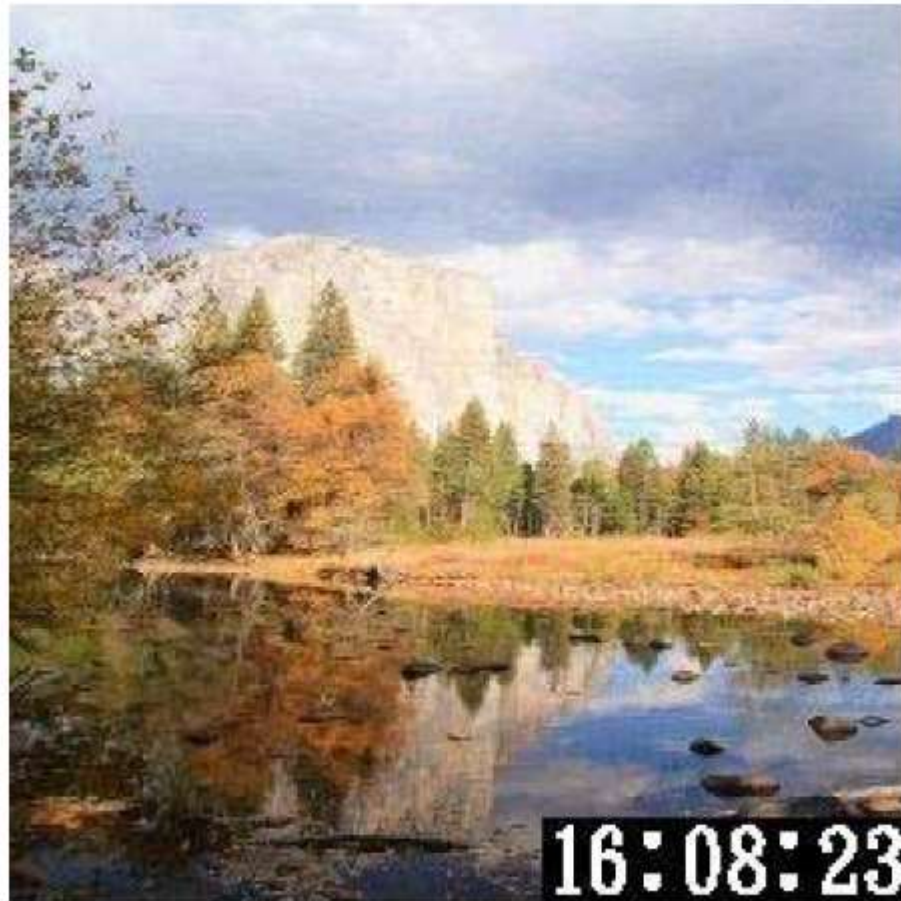
# Computer-Aided VLSI System Design, Spring 2018

## Digital Photo Album

### 1. 問題描述

請完成一數位相框電路設計。如圖一，本數位相框具有時鐘功能，每固定時間會自動切換照片，並可顯示當時時間，其詳細規格將描述於後。

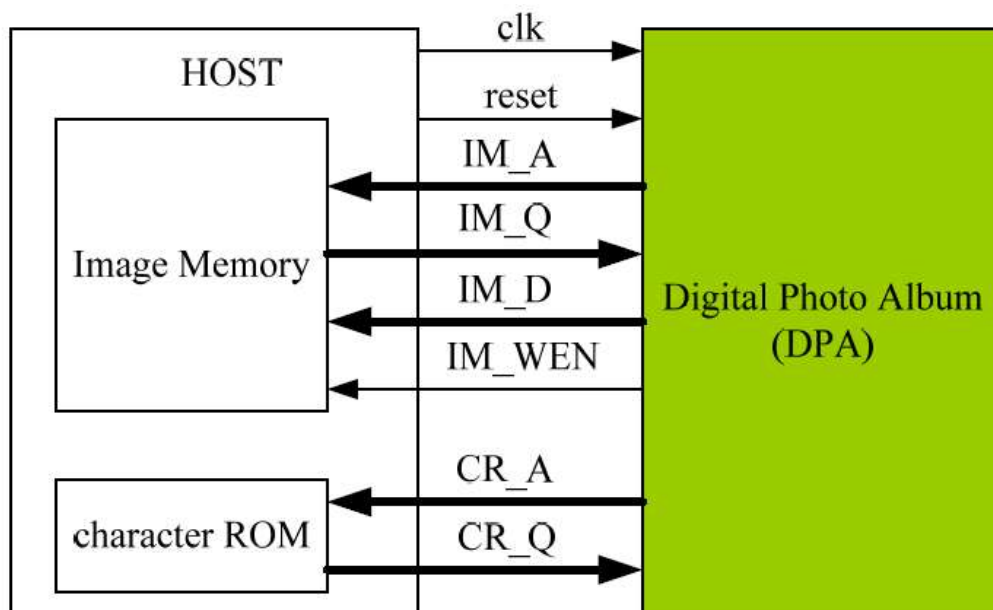
本控制電路各輸入輸出信號的功能說明，請參考表一。每個同學必須根據下一節所給的設計規格及附錄C 中的測試樣本完成設計驗證。



圖一、相框畫面示意圖

### 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。
reset	I	1	高位準非同步(active high asynchronous)之系統重置信號。
IM_A	O	20	Image Memory 的位址匯流排。
IM_Q	I	24	Image Memory 的資料輸出匯流排。
IM_D	O	24	Image Memory 的資料輸入匯流排。
IM_WEN	O	1	Image Memory 的 Write Enable 控制訊號，當 LOW 時可以寫資料進入 Image Memory。
CR_A	O	9	character ROM 的位址匯流排。
CR_Q	I	13	character ROM 的資料輸出匯流排。

## 2.3 系統描述

### 2.3.1 系統功能描述

本數位相框從 Image Memory 中讀取照片資訊，將照片及時鐘內容顯示於螢幕(寫入Frame Buffer)，並每隔2 秒鐘自動切換下一張照片，若所有的照片都顯示完，則從頭循環顯示。Frame Buffer 位於Image Memory 位址後段，Image Memory 位址分佈詳述於2.3.2。

時鐘固定顯示於螢幕最右下方（如圖一所示），**每秒鐘變換一次**，其顯示格式

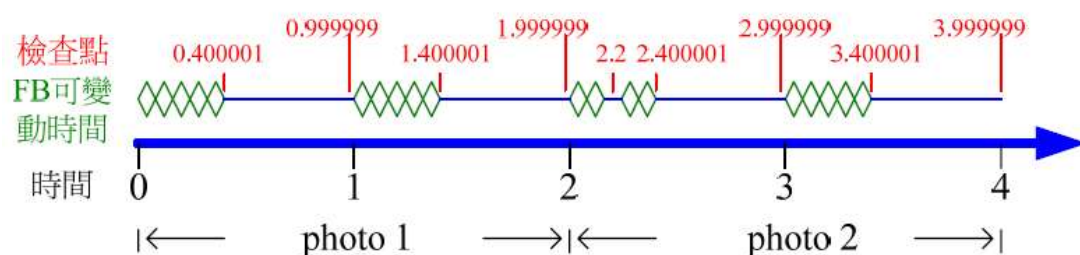
為：**小時：分鐘：秒**，其中**小時、分鐘、秒各使用2個數值**，如6點32分2

秒的顯示為：**06：32：02**，時鐘顯示為24小時制，也就是說，小時的範圍從00到23。數字及冒號的圖形記錄於character ROM中，character ROM的格式詳述於2.3.3。

本數位相框**可接受的照片大小共有3種**，分別是512x512、256x256、128x128，但**顯示螢幕(Frame Buffer)僅有一種大小(256x256)**，對於不同大小的照片顯示時的縮放方式詳述於2.3.4

照片切換顯示當中**須帶有轉場效果**，轉場的方法詳述於2.3.5

在系統一開始，所有的照片資料及 Image Memory 的header 設定都已事先儲存在Image Memory 中，從系統重設(reset)後，數位相框開始在每個整秒(1秒、2秒、3秒…)更新顯示畫面，**每次更新必須在0.4秒內完成**，**轉場畫面必須在0.2秒時完成且穩定不變**，testbench 會在每個**整秒前0.000001秒及整秒後0.400001秒**以及**照片切換後0.2秒**檢查Frame Buffer 內容正確性，而第一個檢查畫面會在0.400001秒時檢查，如圖三。

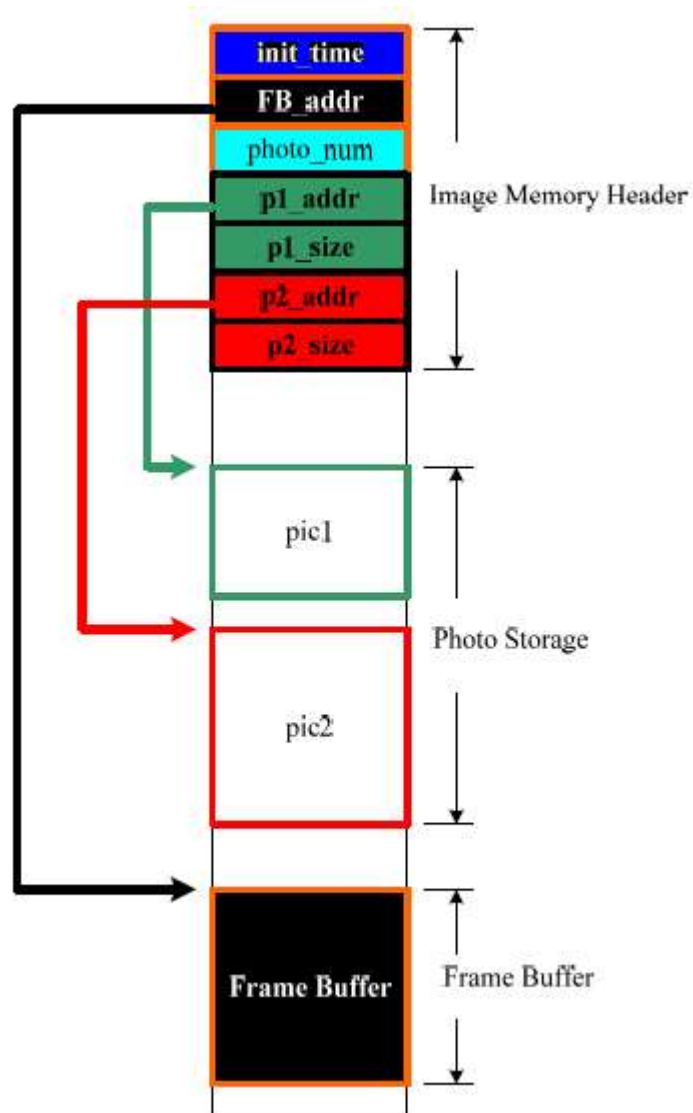


圖三、系統流程

系統時脈(clk)固定為1M，**不可修改時脈速度**，也就是說，必須在有限的cycle內(200000 cycle for 轉場畫面，400000 cycle for 完整畫面)完成畫面新。

## 2.3.2 Image Memory 位址分佈

本數位相框所使用之記憶體 Image Memory，其data width 為24bit，address 定址最大為20bits，Image Memory 共分為3區段，見圖四。



圖四、Image Memory 位址配置圖

➤ Image Memory Header 區段

Image Memory Header 位於Image Memory 最前段，其內容依續詳列如下：

■ 系統初始時間(init\_time)

Image Memory 第一筆資料(位址0)為系統初始時間，其格式如下圖表示，第23~16 bits表示小時，第15~8 bits 表示分鐘，第7~0 bits 表示秒。



■ Frame Buffer 初始位址 (FB\_addr)

Image Memory 第二筆資料(位址1)用於指向Frame Buffer 初始位址。

■ 照片張數 (photo\_num)

Image Memory 第三筆資料(位址2)記錄照片張數，照片最少為1 張，最多為4 張。

■ 照片初始位址(px\_addr)及大小(px\_size)

從位址 3 開始依續分別成對記錄各照片之初始位址(px\_addr)及大小(px\_size)，照片之初始位址表示該照片在 Image Memory 的起始位址，照片之大小共有 3 種，分別是 512x512、256x256、128x128，每一張照片的大小資訊記錄在 Image Memory 的 Header 中，其記錄的方式如下：

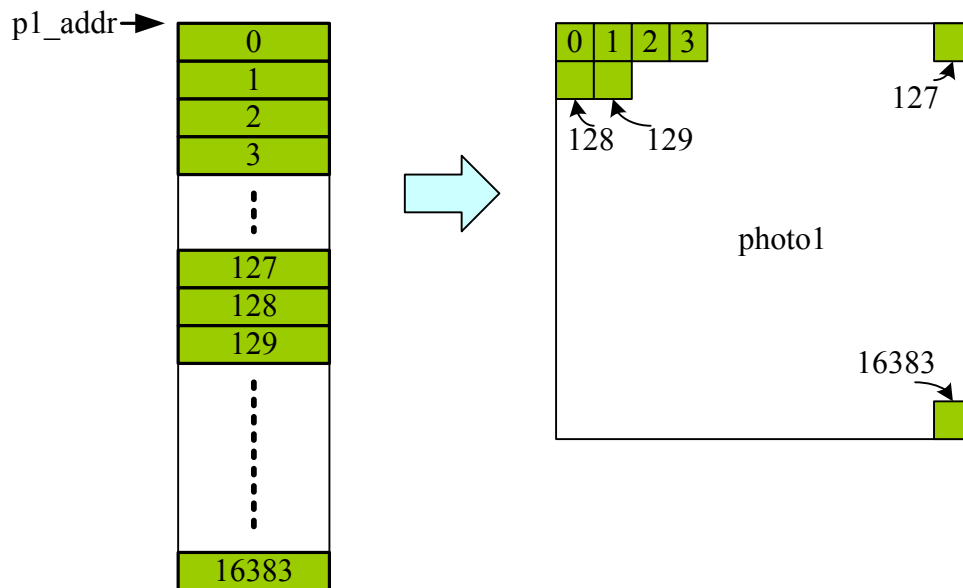
照片大小	px_size
128x128	128
256x256	256
512x512	512

➤ Photo Storage 區段

此區段儲存所有照片內容，每一張照片的初始位址記錄於 Header 中，其所佔用的記憶體大小依照片大小而不同。每一筆資料代表一個 pixel 的 RGB 值，如下圖第 23~16 bits 為 R 值，第 15~8 bits 為 G 值，第 7~0 bits 為 B 值。



記憶體中依續儲存的內容代表照片由左至右，由上至下的 pixel，如圖五為 size=128x128 的照片儲存順序。

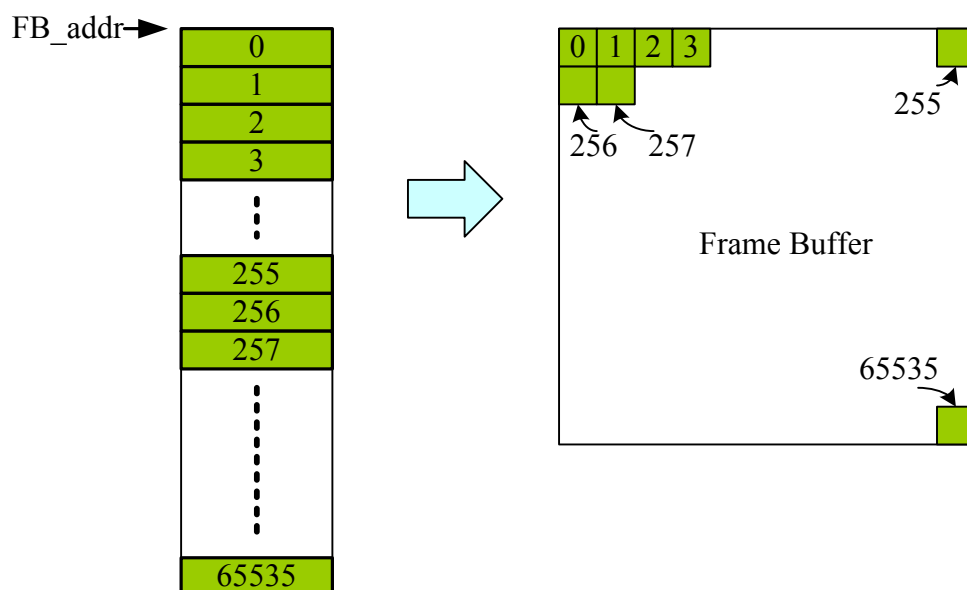


圖五、 照片資料儲存順序

➤ Frame Buffer 區段

Frame Buffer 是 Image Memory 中**唯一可以寫入的區段**，功能是做為數位相框顯示區，將資料寫入 Frame Buffer 即代表顯示於螢幕。

Frame Buffer 的初始位址記錄於 Header 中，佔用的記憶體固定為  $256 \times 256 = 65536$  筆資料，和 Photo Storage 的資料格式一樣，Frame Buffer 每一筆資料代表一個 pixel 的 RGB 值，RGB 由左至右(bit23~bit0)分別各佔 8bits，Frame Buffer 內依續儲存的內容代表螢幕由左至右，由上至下的 pixel，如圖六。

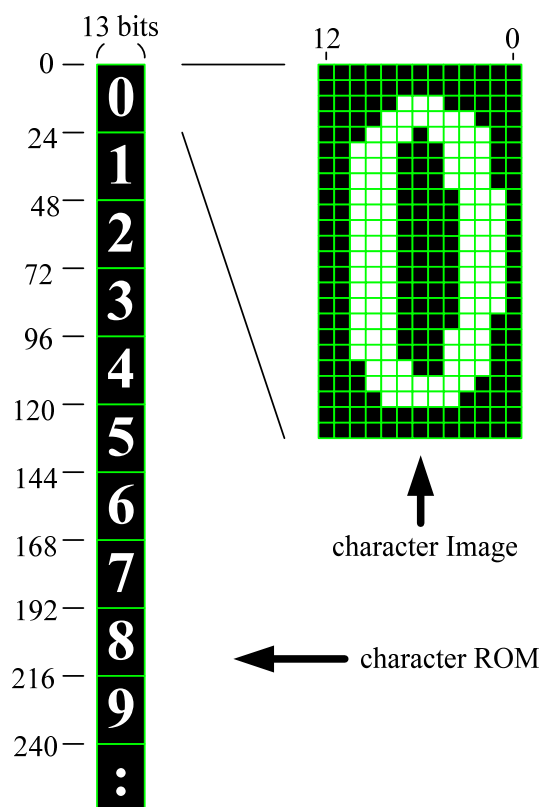


圖六、 Frame buffer 及螢幕對應順序

### 2.3.3 character ROM 格式

0 至 9 及冒號 (:) 等 11 個字元的圖形，依序儲存在 character ROM 當中，每個字元皆為 13x24 pixel，每個 pixel 以 bit map 形式儲存，1 代表白色(RGB=FFFFFF)，0 代表黑色(RGB=000000)。

如圖七，character ROM 為一 13x264 bits 的記憶體，264 個位址，每筆 data 13bits，故每個字元占用 24 筆 data，字元圖形由左至右 13pixel 分別對應到一筆 data 的第 12bit 至第 0bit。



圖七、 character ROM 格式

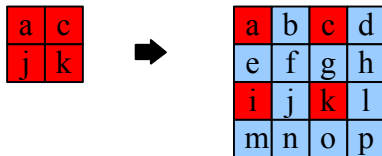


### 2.3.4 顯示縮放(Scaling)

本數位相框可接受的照片大小共有 3 種，分別是 512x512、256x256、128x128，但顯示螢幕(Frame Buffer)僅有一種大小(256x256)，對於和螢幕大小不同的照片必須先經過縮放方能顯示，縮放的規則如下：

#### ■ 128x128 照片補插點

128x128 照片以補插點方式放大到 256x256，插點計算方式為相鄰的點的平均值(小數無條件捨去)，以下圖 2x2 放大到 4x4 為例，分以 4 種狀況說明：

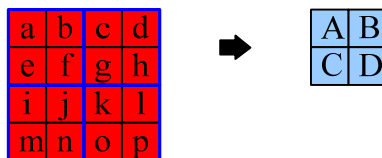


- ◆ 水平插點 (b,j 兩點)  
水平插點為左右兩點的平均值， $b=(a+c)/2$ ， $j=(i+k)/2$
- ◆ 垂直插點 (e,g 兩點)  
垂直插點為上下兩點的平均值， $e=(a+i)/2$ ， $g=(c+k)/2$
- ◆ 中心插點 (f 點)  
中心插點為周邊 4 點的平均值， $f=(a+c+i+k)/4$
- ◆ 邊緣插點(m,n,o,p,d,h,l)  
邊緣插點直接延續邊緣點， $m=i$ ， $n=j$ ， $o=k$ ， $p=k$ ， $l=k$ ， $h=g$ ， $d=c$

#### ■ 512x512 照片取樣

512x512 以四點平均的方式縮小到 256x256，以下圖 4x4 縮小到 2x2 為例

$A=(a+b+e+f)/4$ ， $B=(c+d+g+h)/4$ ， $C=(i+j+m+n)/4$ ， $D=(k+l+o+p)/4$



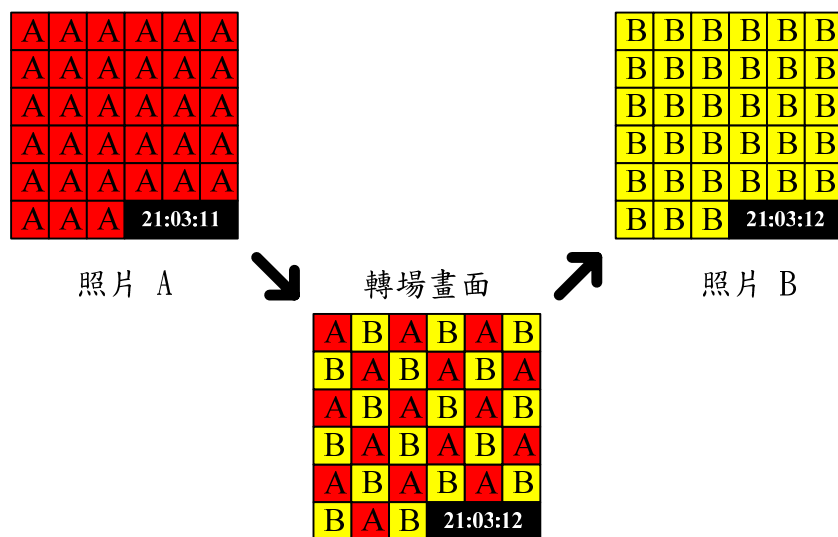
### 2.3.5 轉場效果(transition)

轉場效果出現於當顯示螢幕（即 Frame Buffer 內容）由一張照片轉換到另一張照片當中的過渡期，本數位相框採一階段轉場，即轉場畫面只有一張，轉場方式如圖八所示。

圖八以 6x6 的顯示螢幕為範例，當螢幕顯示由照片 A 轉換照片 B 時，必須先完成轉場畫面後才可完全轉換到照片 B，且此轉場畫面必須在一小段時間內穩定不變。轉場的原則為

1. 對於偶數行，將奇數的點置換為新照片
2. 對於奇數行，將偶數的點置換為新照片

須特別注意的是，**時鐘的區域不作轉場效果**，且轉場畫面的時間屬於後一張畫面的時間。



圖八、轉場效果

## 2.4 時序規格

### 2.4.1 Image Memory 時序圖

Image Memory Read :

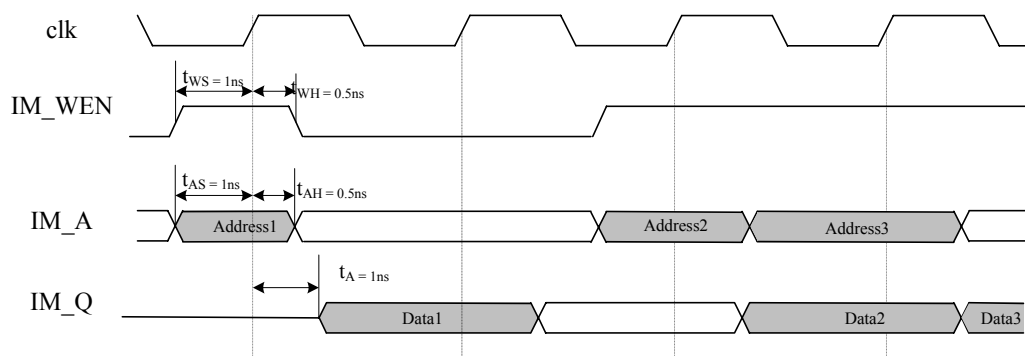
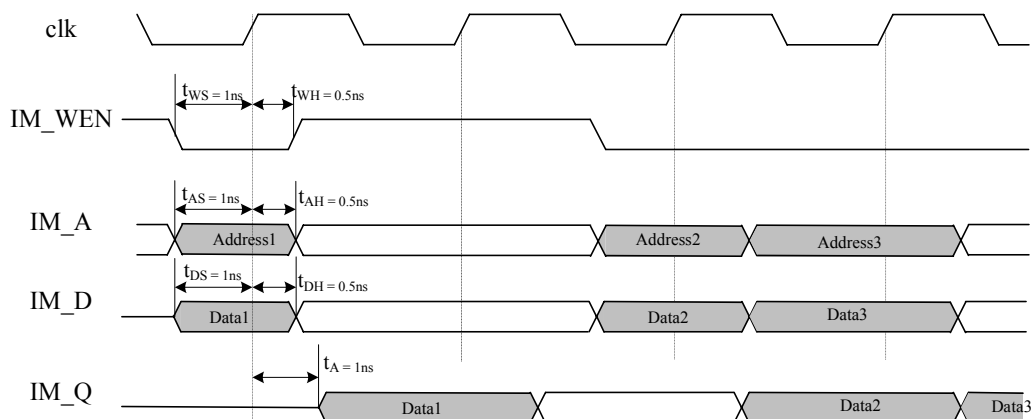


Image Memory Write :

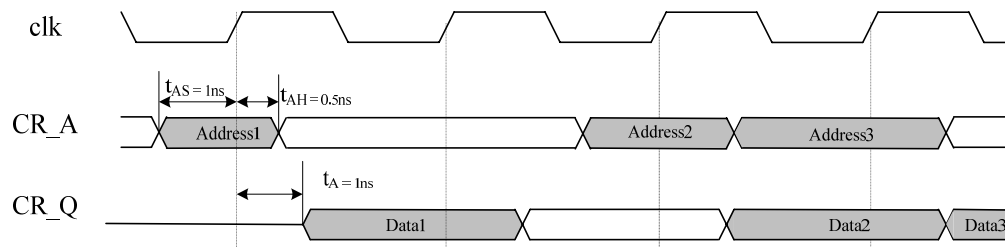


圖九、Image Memory 時序圖



## 2.4.2 character ROM 時序圖

character ROM Read



圖十、Image Memory 時序圖

## 3.模擬及除錯

1. 本試題以 toggle count 作為評分依據，toggle count 指的是 netlist 中所有 net 的開關次數總和。本數據須藉由額外 PLI 功能取得，使用 PLI 模擬之參數請見底下。

2. 因模擬時間很長，請以 ncverilog 取代 verilog 以加快模擬速度

■ ncverilog 指令範例如下：

```
ncverilog testfixture.v dpa.v +define+tb1 +loadpli=tglib:tg +access+r
```

- 上述指令中 **+define+tb1** 指的是使用第一組測試樣本模擬，若須使用其它測試樣本請自行修改此參數。以第二組測試樣本為例: **+define+tb2**。
- 上述 ncverilog 指令中 **+loadpli=tglib:tg** 意義為使用 toggle count PLI。
- toggle count PLI 僅能計算 **gate-level netlist** 的結果，在 RTL-level 時得到 toggle count 為 0 屬於正常結果。
- 關於模擬時使用的一些記憶體 model (charROM.v、IMAGE\_ROM.v、FB.v)，因已經以 include 方式加在 testfixture.v 裏，所以不需加在模擬指令裏。

3. 因波形檔很大，請以 fsdb dump 取代 vcd dump，dump fsdb 須使用指令如下：

■ 使用 ncverilog 者使用指令(fsdb 相關參數以紅色標示)

```
ncverilog testfixture.v dpa.v +define+tb1+FSDB +loadpli=tglib:tg +access+r
```

- **dump 波形檔會嚴重降低模擬速度，請盡可能以文字 display 方式除錯**

- 模擬過程中檢查畫面時會將 Frame Buffer 的每個 pixel 內容儲存下來，儲存檔名為 **tb $n$ \_image0 $m$ .out**，轉場畫面內容儲存為 **tb $n$ \_image0 $m$ \_t.out** ( $n$  代表第幾個測試樣本， $m$  代表第幾秒)，可以和 golden 目錄內的對應的 .golden 檔案比較來幫助除錯。
- 模擬過程中會在每秒及轉場時將 Frame Buffer 的畫面儲存成圖形檔，檔案格式為 xpm 檔，請以下面指令觀看此檔案(以 tb1\_image01.xpm 為例)：

```
xv tb1_image01.xpm&  
gimp tb1_image01.xpm&
```

- testfixture 內有些變數，有助於設計中除錯，這些變數的說明如下：

<b>tick</b>	進行秒數，reset 後 tick 為 0，之後每秒增加 1
<b>count_time</b>	週期數，reset 後的週期數，每週期為 1us
<b>CHECKIMAGE</b>	檢查 Frame Buffer 的驅動訊號，每整秒前及 0.4 秒後會驅動一次，當 high 時會檢查畫面是否正確。
<b>CHECKTRANS</b>	檢查轉場畫面的驅動訊號，每兩秒後 0.2 秒會驅動一次，當 high 時會檢查畫面是否正確。
<b>hour、min、sec</b>	模擬時間的小時、分、秒

- romfile 目錄內 **tb $n$ \_IMAGE.rcf** 為第  $n$  個測試樣本的使用的 Image Memory 內容檔，此檔內對照片的每個 pixel 做了數值、座標、位址等註解，應該有助於程式除錯。

## 4. 評分標準

### 1. 完成設計

完成設計者意指：

- RTL 通過所有測試樣本模擬
- 完成 Synthesis 及 APR，
- APR 結果依照附錄 C 所列之驗證完成實體驗證
- APR 後的 netlist 通過**第二、三、七組**測試樣本模擬

通過以上條件者以下列所列之公式評分。

$$\text{Cost} = \text{Area} * \text{Toggle}$$

Area：P&R 之後，layout 實際面積大小(單位：um<sup>2</sup>)

Toggle：final netlist 模擬第**七組**測試樣本所得的 toggle count 值

Cost 值越低，所得分數越高。

## 2. 完成部分設計

對於未完成 APR 設計者，共分為三個等級，等級由高至低排列為 Physical Level、Gate Level、RTL Level，等級高者優先錄取

### ■ Physical Level

進入這個等級意指

- a、RTL 通過所有測試樣本模擬
  - b、已完成 APR，但 layout 尚有少量 DRC 或 connectivity error
  - c、APR 後的 netlist 通過第二、三、七組測試樣本模擬
- 此等級的排名列於”完成設計者”之後，評分方式和”完成設計”相同

$$\text{Cost} = \text{Area} * \text{Toggle}$$

### ■ Gate Level

進入這個等級意指

- a、RTL 通過所有測試樣本模擬
- b、完成 synthesis
- c、synthesis 後的 netlist 通過第二、三、七組測試樣本模擬

評分方式

$$\text{Cost} = \text{Area} * \text{Toggle}$$

Area：synthesis report 之 cell area

### ■ RTL Level

評分方式

七個測試樣本模擬中，測試樣本通過數量，通過越多者，分數越高

## 附錄

### 附錄 A 設計檔

1. 下表為助教提供的設計檔

表 2、設計檔案說明

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
DPA.v	參賽者所使用的設計檔，已包含系統輸/出入埠之宣告
FB.v	Frame Buffer 的 verilog model
IMAGE_ROM.v	Image Memory 前段資料區的 verilog model
charROM.v	character ROM 的 verilog model
golden/tb $n$ _image0 $m$ .golden	第 $n$ 組測試樣本的第 $m$ 個正確的模擬結果
golden/tb $n$ _image0 $m$ .golden.xpm	第 $n$ 組測試樣本的第 $m$ 個正確的模擬結果的圖形檔
golden/tint???.xpm golden/v?_???.xpm	測試樣本中所使用的原始圖形檔
golden/tint???.hex golden/v?_???.hex	測試樣本中所使用的原始圖形的每個 pixel 的資料 (以 16 進位記錄)
romfile/charROM.rcf	character rom 的 rom file (binary)
romfile/tb $n$ _IMAGE.rcf	第 $n$ 組測試樣本使用的 Image Memory 的 rom file (binary)
task/tb $n$ _xpmwrite.task	測試樣本使用的 verilog task
synopsys_dc.setup	Design Compiler 設定檔
DPA_syn.sdc	合成時使用之 sdc 檔
DPA_pr.sdc	APR 時使用之 sdc 檔
tglib.so	toggle count PLI 檔案

2. 請使用 **DP4.v**，進行遊戲引擎控制器之設計。其模組名稱、輸出/入埠宣告如下所示：

```
module DPA ( clk, reset, IM_A, IM_Q, IM_D, IM_WEN, CR_A, CR_Q);
```

```
input  clk;
```

```
input  reset;
```

```
output [19:0] IM_A;
```

```
input  [23:0] IM_Q;
```

```
output [23:0] IM_D;
```

```
output IM_WEN;
```

```
output [8:0] CR_A;
```

```
input  [12:0] CR_Q;
```

```
endmodule
```

3. 期末專題共提供七組測試樣本，以define 參數方式選擇測試樣本，define 參數設定方法請見第3 章、模擬與除錯之第2 項

## 附錄 B 測試樣本

比賽提供七組測試樣本，為方便同學除錯之用，故提供 Image Memory 的 header 以供參考，(原始照片之圖形檔及資料檔儲存在golden 目錄裏，見附錄 A 表2)

測試樣本1-6 目的是方便除錯，進入 Gate Level 及Physical Level 後，助教僅以測試樣本二、三、七組做驗證。

1. 測試樣本 1:

模擬完成時間: 2 秒

photo1 使用照片: tintgreen256.xpm

描述	IM 位址	數值(dec)	數值(hex)
初始時間	0	12:06:30	0c:06:1e
FB 位址	1	917504	0e0000
照片數	2	1	000001
photo1 位址	3	8	000008
photo1 大小	4	256	000100

2. 測試樣本 2:

模擬完成時間: 2

photo1 使用照片: tintblue512.xpm

描述	IM 位址	數值(dec)	數值(hex)
初始時間	0	12:12:59	0c:0c:3b
FB 位址	1	839680	0cd000
照片數	2	1	000001
photo1 位址	3	12	00000c
photo1 大小	4	512	000200

## 附錄 C 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。

RTL 與Gate-Level 階段：參賽者必須進行RTL simulation 及Gate-Level simulation，模

擬結果必須於題目所定義的系統時脈下，輸出結果正確且無setup/hold time 的問題。

Physical 階段，包含三項驗證重點：

1. 完成最後 layout，
  - i. Marco layout，不含IO Pad。
  - ii. VDD 與VSS power ring 寬度請各設定為2um。
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，使用 Cadence SOC Encounter，執行步驟如下：

在 SOC Encounter 視窗下點選：

“Design → Save → Netlist...”

Netlist File	DPA_pr.v
All other options	Default value

按 。

“Timing → Calculate Delay...”

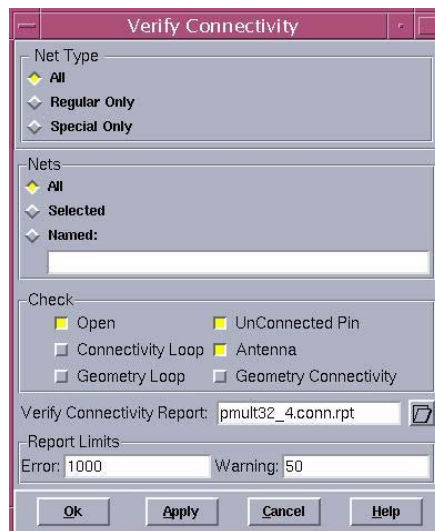
存成 DPA\_pr.sdf，按 。

註：如果發現 Calculate Delay 功能是灰色的(無法點選)，請先將目前結果存檔後離開Encounter，再重新進入Encounter 並Restore 回原本Design 即可。

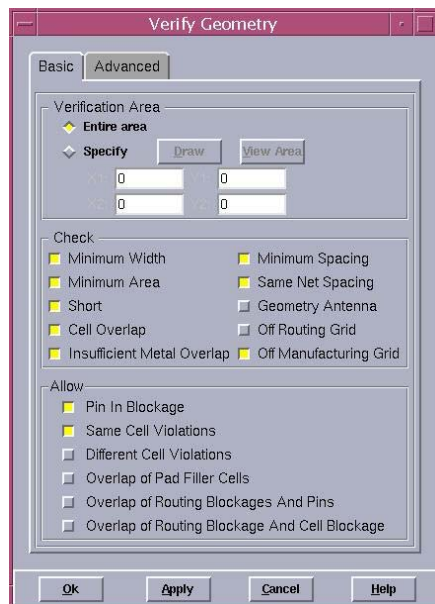
3. 完成 DRC 與LVS 驗證：參賽者必須以其所使用之P&R 軟體內含之DRC 與LVS 驗證功能完成DRC 與LVS 驗證，驗證DRC 與LVS 步驟如下：

在 SOC Encounter 視窗下點選

“Verify → Verify Connectivity...” Default 值，按 。



“Verify → Verify Geometry...” Default 值，按 **OK**。



“Verify → Violation Browser...”  
將 Verify 的結果存成 DPA.viols.rpt



3. 測試樣本 3:

模擬完成時間: 2 秒

photo1 使用照片: tintred128.xpm

描述	IM 位址	數值(dec)	數值(hex)
初始時間	0	3:59:59	03:3b:3b
FB 位址	1	33036	00810c
照片數	2	1	000001
photo1 位址	3	16396	00400c
photo1 大小	4	128	000080

4. 測試樣本 4:

模擬完成時間: 3 秒

photo1 使用照片:tintred128.xpm

photo2 使用照片:tintgreen256.xpm

描述	IM 位址	數值(dec)	數值(hex)
初始時間	0	22:12:02	16:0c:02
FB 位址	1	851968	0d0000
照片數	2	2	000002
photo1 位址	3	12	00000c
photo1 大小	4	128	000080
photo2 位址	5	16400	004010
photo2 大小	6	256	000100

5. 測試樣本 5:

模擬完成時間: 5 秒

photo1 使用照片:tintred256.xpm

photo2 使用照片:tintgreen512.xpm

描述	IM 位址	數值(dec)	數值(hex)
初始時間	0	23:59:59	17:3b:3b
FB 位址	1	917504	0e0000
照片數	2	2	000002
photo1 位址	3	4096	001000
photo1 大小	4	256	000100
photo2 位址	5	135166	020ffe
photo2 大小	6	512	000200

6. 測試樣本 6:

模擬完成時間: 3 秒

photo1 使用照片: tintgreen256.xpm

photo2 使用照片: tintred128.xpm

描述	IM 位址	數值(dec)	數值(hex)
初始時間	0	00:07:04	00:07:04
FB 位址	1	786432	0c0000
照片數	2	2	000002
photo1 位址	3	7	000007
photo1 大小	4	256	000100
photo2 位址	5	65543	010007
photo2 大小	6	128	000080

7. 測試樣本 7:

模擬完成時間: 8 秒

photo1 使用照片: v1\_128.xpm

photo2 使用照片: v2\_512.xpm

photo3 使用照片: v3\_256.xpm

photo4 使用照片: v4\_512.xpm

描述	IM 位址	數值(dec)	數值(hex)
初始時間	0	23:59:57	17:3b:39
FB 位址	1	983040	0f0000
照片數	2	4	000004
photo1 位址	3	16	000010
photo1 大小	4	128	000080
photo2 位址	5	16400	004010
photo2 大小	6	512	000200
photo3 位址	7	278544	044010
photo3 大小	8	256	000100
photo4 位址	9	344080	054010
photo4 大小	10	512	000200

## 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Cadence SOC Encounter 者，請將SOC Encounter 相關的design library（包含一個.enc 檔及一個.dat 目錄），壓縮成一個檔案

表 3

<i>RTL category</i>		
<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
<i>Gate-Level category</i>		
<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
<i>Physical category</i>		
<i>Design Stage</i>	<i>File</i>	<i>Description</i>
P&R	*.tar	archive of the design library directory
	*.gds	GDSII layout
	DRC/LVS report	For Astro : DRC.report ; LVS.report For SOC Encounter : DPA.viols.rpt
Post-layout Gate-level Simulation	*_pr.v	Verilog gate-level netlist generated by Cadence SOC Encounter or Synopsys Astro
	*_pr.sdf	Post-layout gate-level sdf