

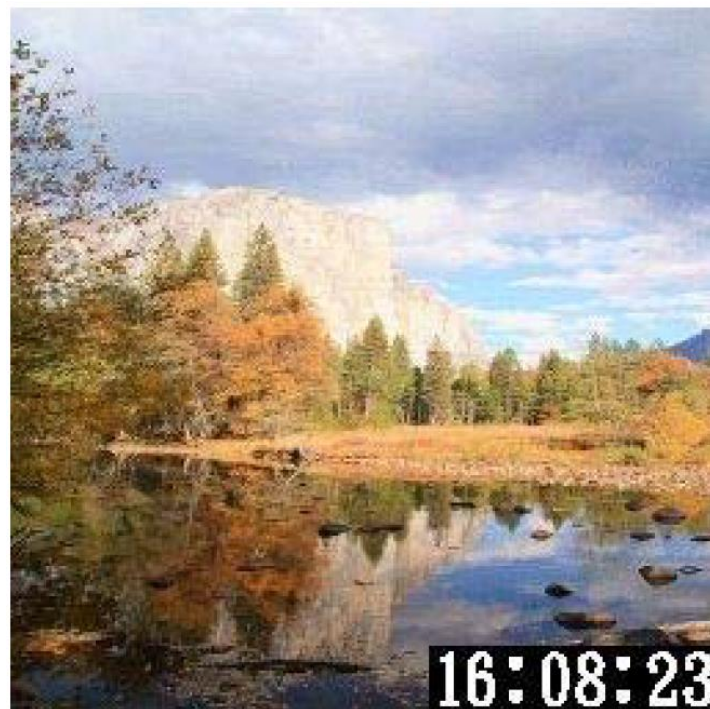
Computer-Aided VLSI System Design, Fall 2015

HW3 - 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，並可顯示當時的時間，其詳細規格將描述於後。

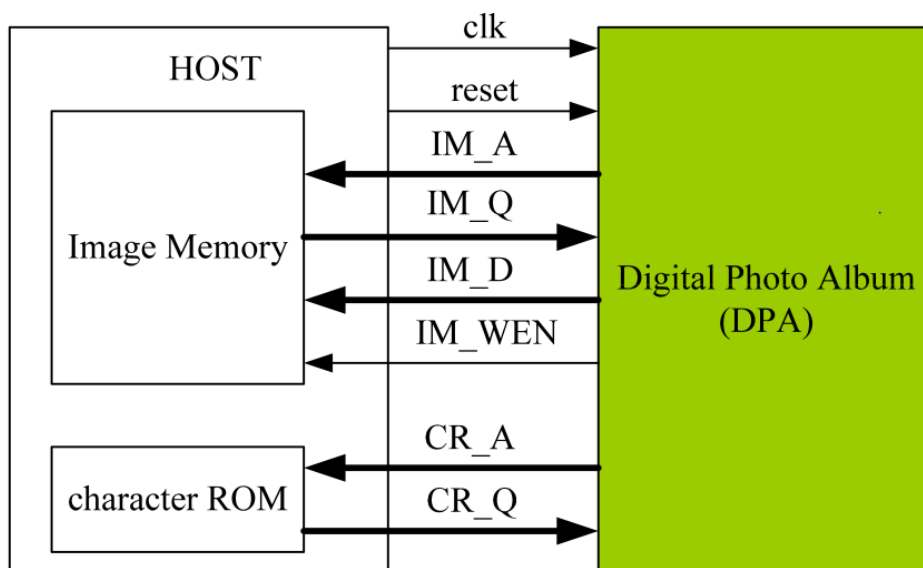
本控制電路各輸入輸出信號的功能說明，請參考表一。完成電路設計後必須根據下一節所給的設計規格及附錄 B 中的測試樣本完成設計驗證。並且應該參考附錄 C 中所列的要求，附上所需要的檔案。



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 protocol 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 控制信號，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

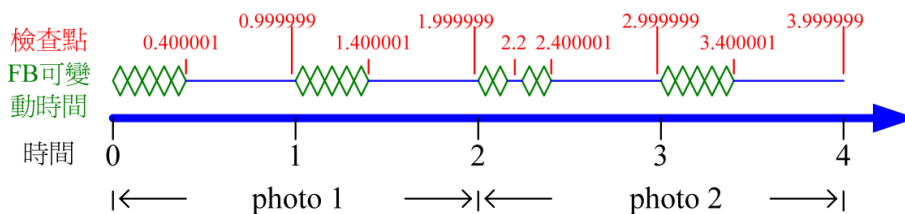
本數位相框從 Image Memory 中讀取照片資訊，將照片及時鐘內容顯示於螢幕(寫入 Frame Buffer)，並每隔 2 秒鐘自重切換下一張照片，若所有的照片都顯示完，則從頭循環顯示。Frame Buffer 位於 Image Memory 位址後段，Image Memory 位址分布詳述於 2.3.2。

時鐘固定顯示於螢幕最右下方(如圖一所示)，每秒鐘便換一次，期顯示格式為: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實的縮放方式詳述於 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 秒檢察 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。



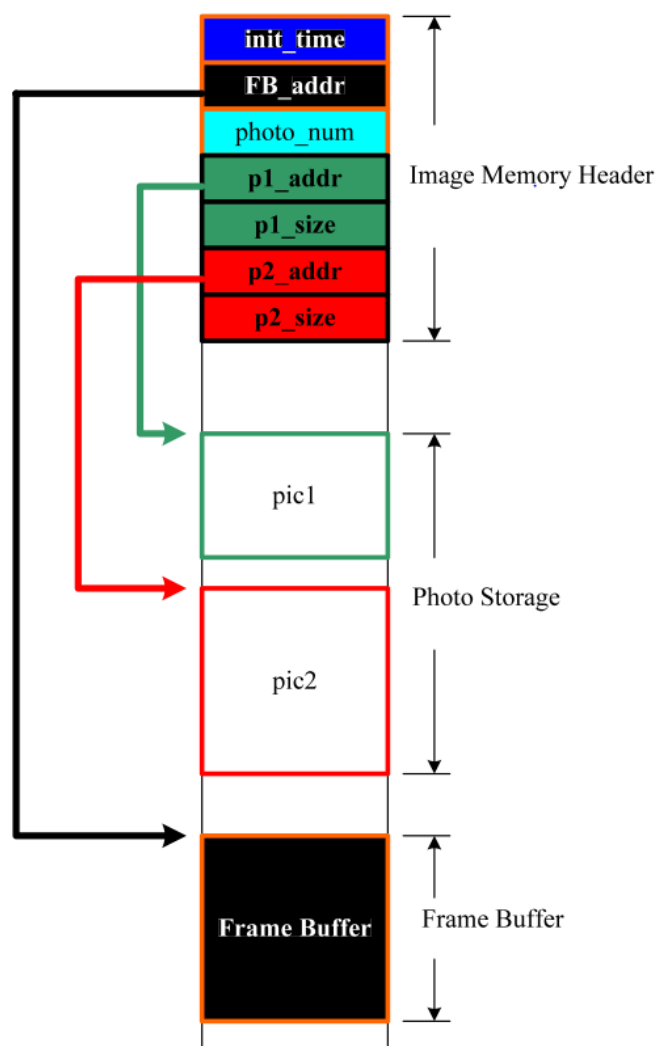
圖三、系統流程

系統時脈(clk)固定為 1M，不可修改時脈速度，也就是說電路設計者必須在有限的 cycle 內(200000 cycle for 轉場畫面，400000 cycle for 完整畫面)完成畫面更新。

本題評分方向為面積和功率，詳細評分方式見第 4 章。

2.3.2 Image Memory 位址分佈

本數位相框所使用之記憶體 Image Memory，期 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，其內容依序詳列如下：

- **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)紀錄照片張數，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 開始依序分別成對記錄各照片的初始位址(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紀錄在 Image Memory 的 Header 中，其紀錄的方式如下：

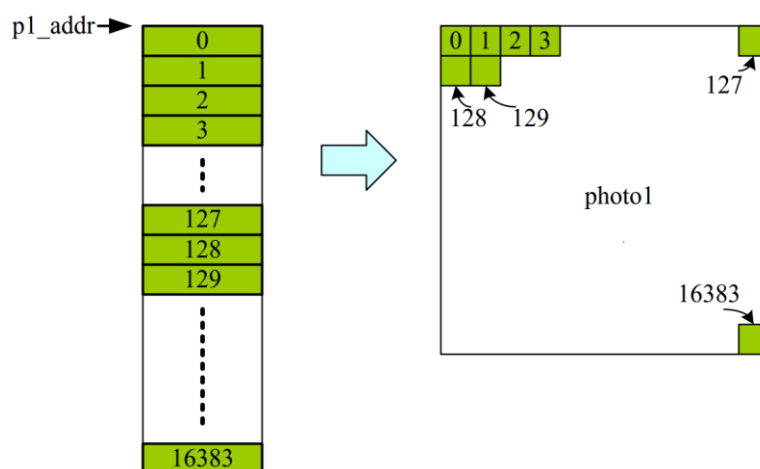
照片大小	px_size
128x128	128
256x256	256
512x512	512

● 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。



記憶體中依序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。

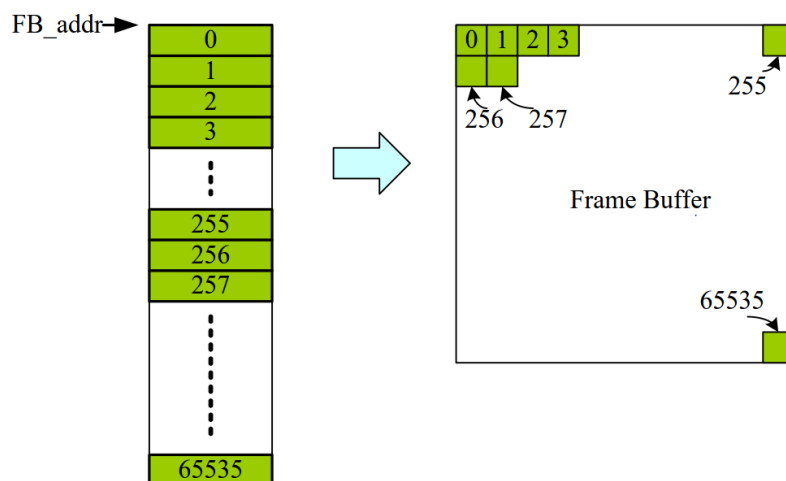


圖五、照片資料儲存順序

● Frame Buffer 區段

Frame Buffer 是 Image Memory 中**唯一可以寫入的區段**，功能是最為數位相框顯示區，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 內依序儲存的內容代表螢幕由左至右，由上至下的 pixel，如圖六。

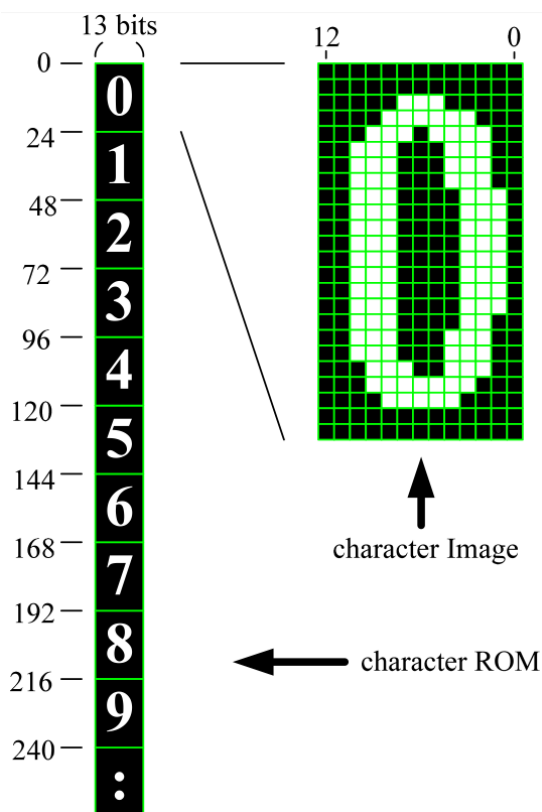


圖六、Frame Buffer 及螢幕對應順序

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，故每個字元佔用 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。



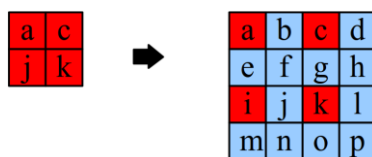
圖七、character ROM 格式

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才能顯示，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)**

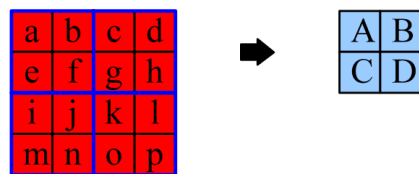
中心插點為周邊 4 點的平均值， $f=(a+c+i+k)/4$

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



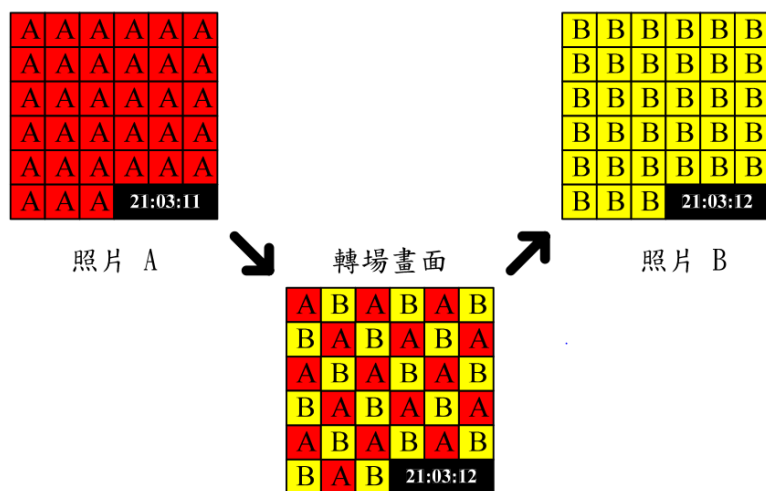
2.3.5 轉場效果(translation)

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 轉換到照片 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

需特別注意的是，**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 :

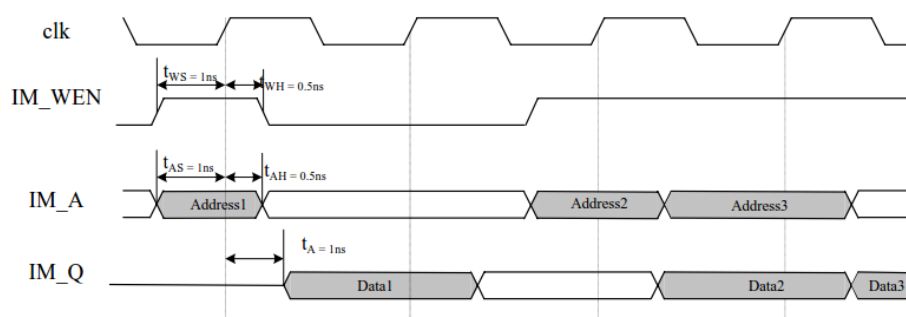
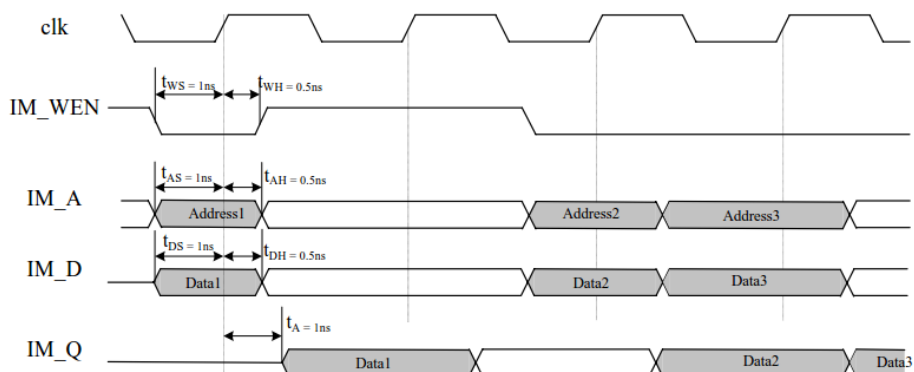


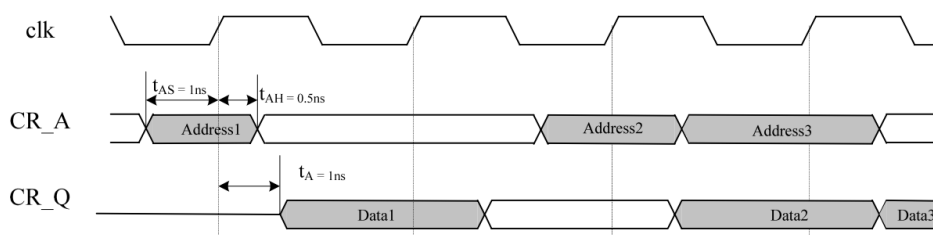
Image Memory Write :



圖九、Image Memory 時序圖

2.4.2 character ROM 時序圖

character ROM Read



圖十、character ROM 時序圖

3.模擬及除錯

1. 模擬中

- ncverilog 指令範例如下:

```
ncverilog testfixture.v DPA.v +define+tb1 +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**。
- 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。

2. 因波形檔很大，如果可以 fsdb dump 取代 vcd dump 的話最好，如果工作站沒有支援的話，盡可能以**文字 display 方式除錯**。(助教已提供此功能)

vcd、fsdb 相關參數以紅色表示

```
ncverilog testfixture.v DPA.v +define+tb1+FSDB +access+r
```

```
ncverilog testfixture.v DPA.v +define+tb1+VCD +access+r
```

3. 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 目錄內對應的 .golden 檔案比較來幫助除錯。
4. 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，幫助同學視覺上除錯，如果工作站支援的話，請以下面指令觀看此檔案。

```
xv tb1_image01.xpm&
```

```
gimp tb1_image01.xpm&
```

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

tick	進行秒數，reset 後 tick 為 0，之後每秒增加 1
count_time	週期數，reset 後的週期數，每週期為 1us
CHECKIMAGE	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。
CHECKTRANS	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。
hour、min、sec	模擬時間的小時、分、秒

6. romfile 目錄內 **tb_n_IMAGE.rcf** 為第 **n** 個測試樣本使用的 Image Memory 內容檔，此檔內對照片的每個 pixel 做了數值、作標、位址等註解，有助於程式除錯。

4. 評分標準

(1) 依是否完成設計第一階段評分(40%):

即完成下列三項要求。請按照 a、b、c 的要求順序完成。

✧ “完成設計”的三項要求:

A、RTL 通過作業所有提供的測試樣本模擬。(20%)

B、RTL 在助教的測試資料之下通過模擬。(5%)

C、完成 Synthesis，且 DPA_syn.v 的 GateLevel Pre-layout Simulation 有達到 a、b 項之要求。(10%)

<Note1> 請確認助教在 RTL 模擬上可以直接跑下列指令。如果有額外的 module 在其它.v 檔，請利用`include 檔案在 DPA.v 之中，並且請確認檔案的命名。

在 gate-level 模擬，助教會用以下指令:

對於 gate-level simulation，請記得更改 SDF 檔案的名稱以及確認檔案位置，與 testbench 中的變數”SDFFILE”一致。

ncverilog testfixture.v DPA_syn.v -v tsmc13.v +define+tb1+SDF (for test pattern 1)

<Note2> 合成環境的設置: 請根據 synthesis lab 所提供的”.synopsys_dc.setup”設置。

target_librabry	slow.db
link_librabry	* \$target_library dw_foundation.sldb
symbol_library	generic.sdb
synthetic_librabry	dw_foundation.sldb

<Note3> 關於合成 **sdc constraint** 檔不可更動。但若須要使用 design compiler 作 power optimization 的動作是允許的。

(2) 依設計的 Cost 做第二階評分(40%):

助教會將符合第一階段完成設計(三項要求都完成)者，依據下式作第二階段的評分，其中 Cost 越低成績越佳。

$$\text{Cost} = \text{Area}(\mu\text{m}^2) * \text{Power Consumption}(\mu\text{W})$$

1. Area = synthesis report 中的 Total Cell Area (不含 wire load model 面積)
2. power consumption = Total Dynamic Power + Cell Leakage Power
3. 請將 design compiler 關於 Area 以及 Power 截圖附在報告中以證明。

A Class: ~5% (40 points)

B Class: 6%~20% (30 points)

C Class: 21%~70% (20 points)

D Class: 71%~100% (10 points)

E Class: 在 gate-level 模擬時有錯誤 (0 points)

✧ 在此次作業中助教會嚴厲抓抄襲，實際看.v 檔以及合成時的.ddc 檔(工作檔，可知道 constraint 等)，並且檢查報告內容是否與程式相符以確認公平性。任何作弊行為將會導致此次作業 0 分。

(3) 作業報告(20%):

在報告中，請針對以下幾點做說明，若需要可配上圖片講解。

1. 針對電路的控制部分如何設計？說明 State diagram 轉換，並且在轉換旁標註電路控制訊號以示轉換條件。
2. 對於電路的架構，Data path 部份如何的做設計。如何利用固定 cycle 以及固定 cycle 數這點切入設計，Area 跟 Power 間又如何做 trade-off?
3. 此次設計中，Image memory 以及 char ROM 部份電路皆包含在 testbench 之中，如果今天除了 FB.v 在 testbench 中，其它兩電路檔(Image_ROM.v、charROM.v)皆包含於設計電路 DPA.v 中，則在合成時應該注意哪些東西才可以確保時序上的正確性？
4. 對於第二階段評分的數據，以及相關證明截圖。

附錄 A 設計檔

1. 下圖為此次作業所提供的設計檔

表 2、設計檔案說明

檔名	說明
testfixure.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	同學所使用的設計檔，已包含系統輸/出入信號之宣告
FB.v	Frame Buffer 的 verilog model
IMAGE_ROM.v	Image Memory 前段資料區的 verilog model
charROM.v	caracter 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 個正確的模擬結果圖形檔
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 的環境設定檔 (lab3 提供)
DPA_syn.sdc	合成時使用之 sdc 檔

附錄 B 測試樣本

作業提供六組測試樣本，為方便設計者除錯之用，在 IMAGE_ROM.v 的 header 皆有提供註解以方便設計者除錯之用。

附錄 C 繳交檔案

RTL Category		
Design Stage	File	Description
N/A	StudentID.pdf	Design Report Form
RTL Simulation	DPA.v	Verilog synthesizable RTL code
Gate-level Category		
Design Stage	File	Description
Pre-layout Gate-level Simulation	DPA_syn.v	Verilog gate-level netlist generated by Synopsys Design Compiler
	DPA_syn.sdf	Pre-layout gate-level sdf
	DPA_syn.ddc	Design database generated by Synopsys Design Compiler

Naming convention: **Student_HW3_vk.zip**

(k is number of version, k = 1,2,...)

◇ 繳交時間

Submission Deadline: 11/11 18:00