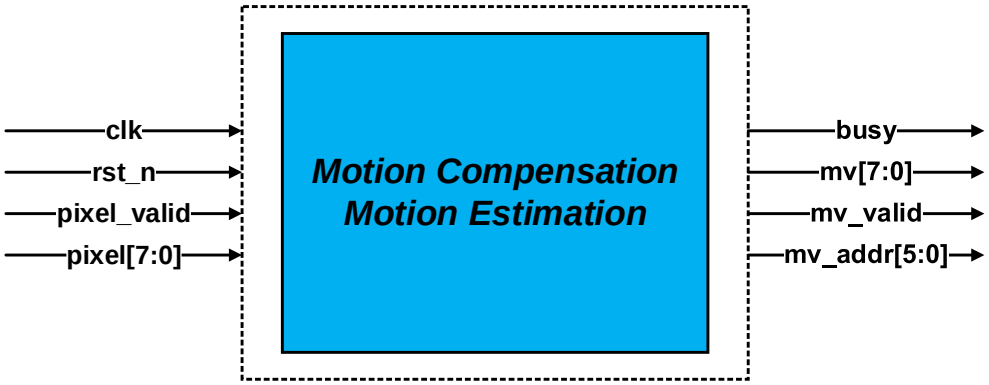
Computer-Aided VLSI System Design, Fall 2019

Motion Estimation/Compensation

1. 問題描述

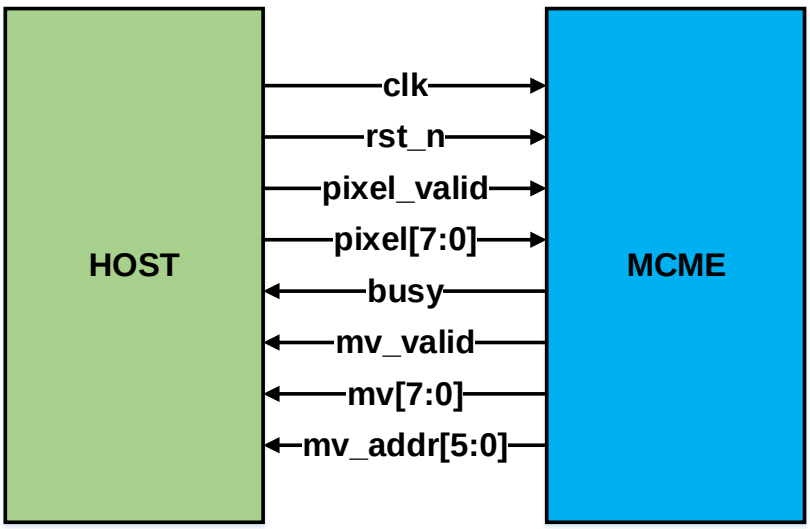
請完成一 Motion Estimation/ Compensation (後文以 MEMC 表示)的電路設計。此電路可接收多張 48x64 像素(Pixels)的 8bits 灰階影像(Gray Level Image)訊號，再根據 Block-Matching Motion Estimation 演算法，輸出該張影像的 Motion Vector (後文以 mv 表示)訊號，且達到指定之 PSNR。關於 Motion Estimation 詳細規格將描述於後。本電路各個輸入輸出信號的功能說明，請參考表一。每組隊伍必須根據下一節所給的設計規格及附錄 A 中的測試樣本完成設計驗證。



圖一、MCME之方塊圖

2. 設計規格

2.1. 系統方塊圖



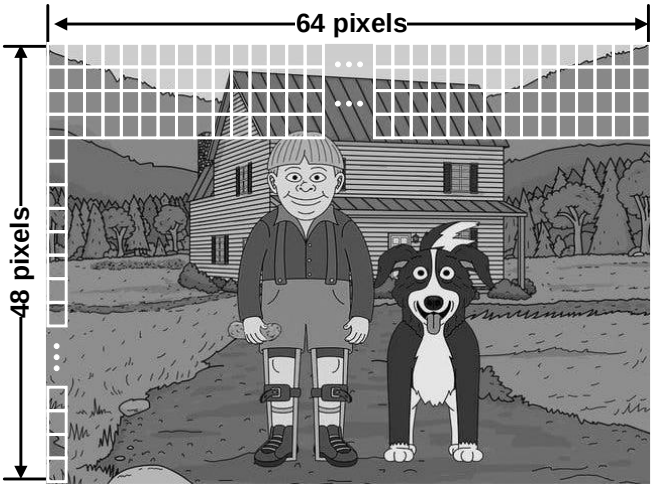
圖二、系統方塊圖

2.2. 輸入/輸出介面

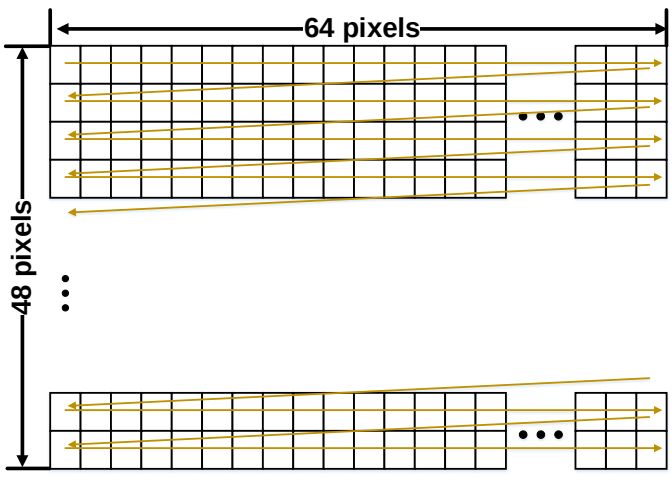
表一-輸入/輸出訊號

Signal Name	I/O	Width	Simple Description
clk	I	1	Design Clock，其週期默認值為 10ns，可以在符合 timing rule 的狀況下進行調整。
rst_n	I	1	Reset Signal。(Active Low)
pixel_valid	I	1	Input pixel signal valid。當該訊號為 high 時 pixel 為有效值。
pixel	I	8	輸入的灰階像素訊號(8 bits)，做為整張 frame 之 pixel 訊號輸入用，輸入順序由左上到又下 P(0, 0)、P(0, 1)、P(0, 2) … P(0, 63)、P(1,0)、P(1,1) … P(47, 63)，其詳細輸入方式請參考 **2.3 系統描述**。
busy	O	1	當該訊號為 high 時，host 端會暫停更新輸入的 pixel 訊號。
mv_valid	O	1	輸出之 motion vector 有效訊號。當該訊號為 high 時，mv、mv_addr、mv_frame 為有效值。
mv	O	8	Motion Vector Signal。該訊號為 8 bits，前 4 bits 為該 vector 之 row 座標。後 4bits 為該 vector 之 column 座標。 **mv = {4'bRow, 4'bColumn}。**
mv_addr	O	6	此訊號為 Motion Vector 在該張 frame 的座標。共 6 bits，前 3 bits 為 row 座標，後 3 bits 為 column 座標。

2.3. 系統描述



圖三、48x64 灰階影像範例

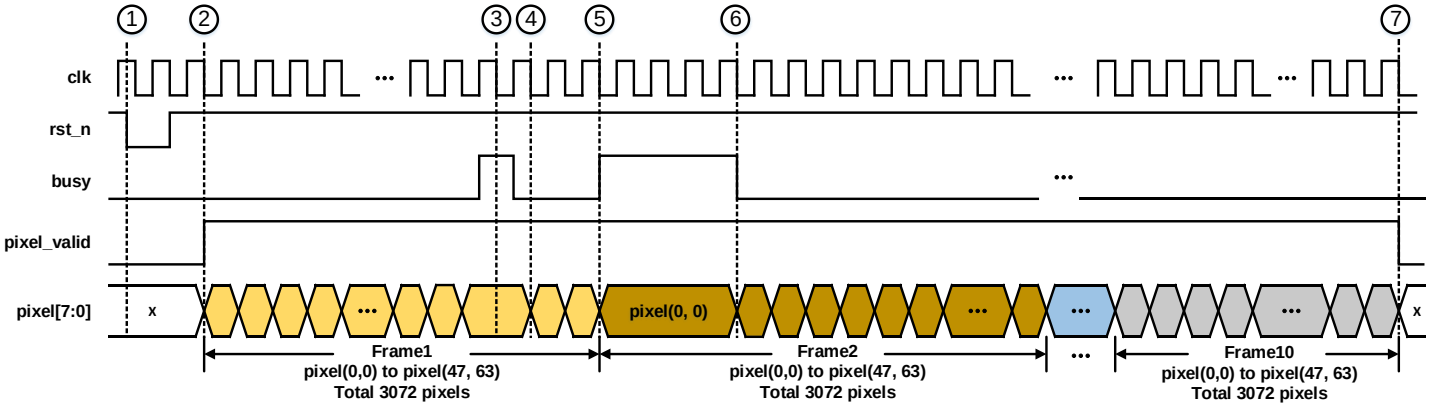


圖四、pixel輸入順序

圖三是一張 48x64 大小的灰階影像圖，這樣一張圖定義為一個 frame，圖片中一格白色方框稱為一個 pixel，每個 pixel 用 8 bits 表示。當 MEMC 的 busy 為 low，HOST 端會由第一個 frame 的第一個 pixel(0,0)，連續輸入 pixels 直到最後一個 frame 的最後一個 pixel(47, 63)。當 MEMC 的 busy 拉起為 high，HOST 端會停止更新輸入的 pixel，此時 pixel_valid 會維持再 high 且 pixel 會維持不變，直到 busy 再次回 low，pixel 的值才會繼續更新。注意，MEMC 不能向 HOST 端讀取先前輸入過的 pixels。

在 MEMC 輸出端必須依照輸入 frame 的順序輸出對應之 mv，也就是在第一張 frame 的 mv 輸出完之前不能輸出第二張 frame 之 mv。但在同一張 frame 中的 mv 順序沒有規定，各組可透過 mv_addr 控制。HSOT 端在 mv_valid 為 high 時才會判斷該 mv、mv_addr 為有效值，並在每張 frame 之 mv 輸入完畢後輸出該張 frame 之 frame per second (fps)及其 PSNR。同時考慮到模擬時間，僅需處理 10 張 frame。

2.3.1. MEMC 輸入方式

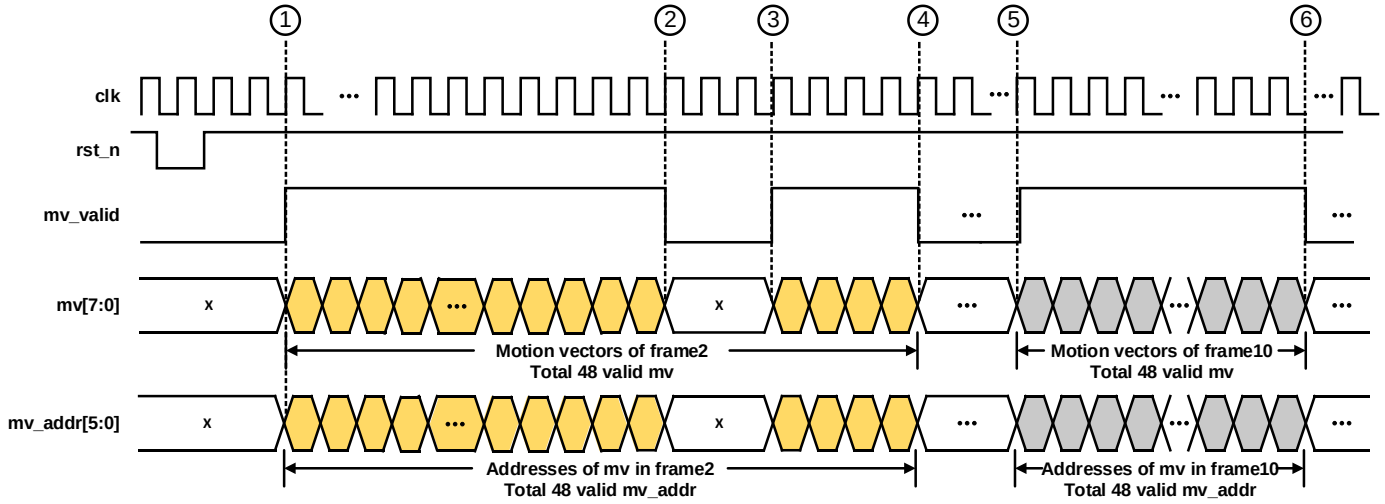


圖五、系統輸入時序圖

HOST 對 MEMC 輸入方式說明如下：

1. 圖五，系統電路模擬時間軸 10ns 時，HOST 端會先給 rst_n 為 low 約 1~3 cycle 之長。若有使用到 rst_n 訊號，請使用非同步 reset 設計電路。
2. 圖五，reset 訊號回到 high 後再隔一個週期後的負緣 (negative edge)系統開始判斷 MEMC 之 busy 訊號。若 busy 訊號為 low，則 HOST 端開始輸入有效 pixel 值。注意，HOST 端輸入以負緣觸發。
3. 圖五，當 HOST 收到之 busy 訊號為 high，則 pixel_valid 與 pixel 訊號回維持不變。
4. 圖五，當 HOST 收到之 busy 訊號為 low，則 pixel_valid 維持 high，pixel 訊號開始更新。
5. 圖五，當 HOST 已完成 3072 pixels 之輸出，便會開始輸出下一張 frame 之第一個 pixel 訊號。而 busy 訊號為 high，因此 pixel 訊號維持不變。
6. 圖五，當 busy 訊號為 low，HOST 開始更新輸出之 pixel 訊號。
7. 圖五，當最後一張 frame 之最後一個 pixel 輸出成功，HOST 停止輸出有效 pixel 訊號。注意，所謂輸出成功，必須要在 HOST 輸出有效 pixel 值時 busy 訊號為 low 才算一次輸出成功。

2.3.2. MEMC 輸出方式



圖六、MCME 輸出時序圖

MEMC 對 HOST 輸出方式如下:

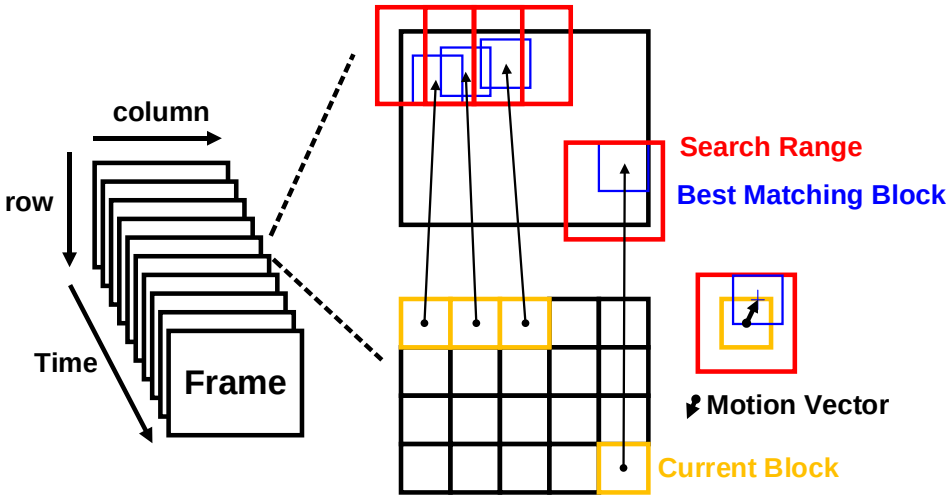
1. 圖六，在 reset 後當 mv_valid 為 high 時判定為有效 mv 以及有效 mv_addr。
2. 圖六，當 MEMC 輸出之 mv_valid 為 low 時，mv 以及 mv_addr 為無 效值，HOST 端不會採取任何動作。
3. 圖六，當 MEMC 輸出之 mv_valid 再次回到 high，HOST 才繼續讀取有效 mv 以及 mv_addr。
4. 圖六，MEMC 完成一個 frame 之 mv 以及 mv_addr 之輸出。HOST 內部會有 counter 判斷以輸出之 mv 數量，每張 frame 有 48 個 mv，當 48 個 mv 均輸出完畢後便會計算該 frame 之 PSNR 值。
5. 圖六，MEMC 開始輸出下一個 frame 之 mv 以及 mv_addr。
6. 圖六，MEMC 輸出完全不 10 張 frame 的所有 mv 以及 mv_addr 後模擬結束。

2.3.3. Single-Port 記憶體

本題提供 256x8、512x8、4096x8、16384x8 四種大小的 Single-Port SRAM 供各組使用。各組請自行斟酌要使用哪種記憶體，可以任意選用抑或不使用也可以。有關以上記憶體讀、寫之時序圖，請自行參考 sram_256x8.pdf、sram_512x8.pdf、sram_4096x8.pdf、sram_16384x8.pdf 檔案。

3. Motion Estimation/ Compensation 演算法

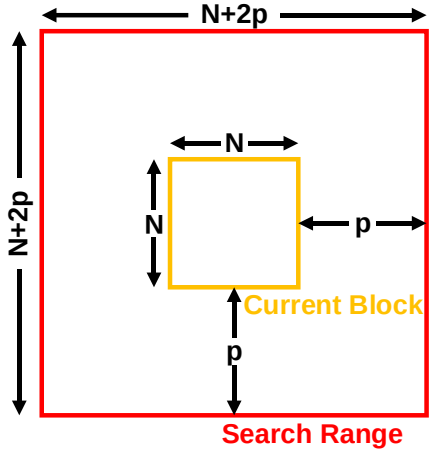
Motion Compensation (運動補償)為一描述相鄰幀差別的方法，具體來說是描述前一幀影像中的某個區塊移動到當前幀影像中的某個位置去，是一種常見的影像壓縮演算法。而 Motion Estimation (運動預測)則為透過運動向量描述兩幀影像之間的轉換之演算法，其主要分為 Pixel-based 與 Block-based 兩大宗。本題目要求各隊伍以 Block-based 演算法進行 MEMC 之實現。



圖七、MEMC 演算法

圖七解釋 MEMC 演算法如何運作，

1. 將當前幀影像分成數個 Current Block (CB)
2. 每個 CB 須在指定的 Search Range 中，對上一幀影像搜尋 Best Matching Block (BMB)
3. 以 BMB 到 CB 的差做為 Motion Vector 輸出。也就是$mv(k,l) = BMB(i,j) - CB(i,j)$



圖八、影響Best Matching Block的參數

判斷 Best Matching Block 的指標可以使用 Sum of Square Pixel Difference (SSD)或是 Sum of Absolute Pixel Difference (SAD)。**在本題中 Current Block 大小限定為 8x8，Search Range 為[+7, -7]，且<u>總 PSNR 必須達到 23</u> (本題使用灰階影像，因此 MAX_I 為 255)。**

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2$$

$$PSNR = 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right) = 20 \cdot \log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right)$$

此外各組同學可以參考資料夾中 MEMC.pdf 文件，其中有 MEMC 更詳盡的說明。

4. 評分標準

評分方式會依設計完成 程度，A、B、C、D 四種等級，排名順序為 A>B>C>D，評分項目為 Time-to-Area 值，助教會依此 Time-to-Area 值大小作為同等級之評分高低 。

✧ **評分項目：依"Time-to-Area"大小評分**

■ **範例**

innovus > 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 um^2) / alloc_area 79981 sites (135760 um^2).
Pin Density = 0.122.
            = total # of pins 23994 / total Instance area 196550.
************************* Analyze Floorplan **************************
    Die Area(um^2)             : 333039.25
    Core Area(um^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 um^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在存檔

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、 兩組測資都達到題目規定之 PSNR，即 RTL 模擬結果正確。

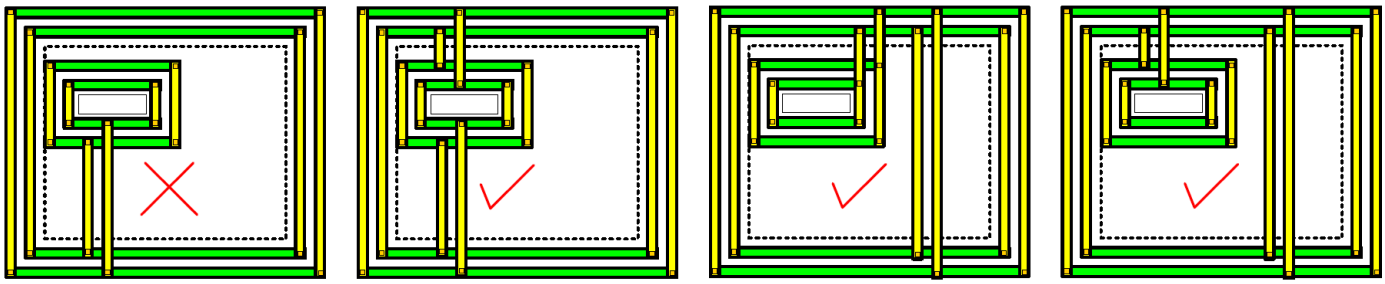
b、 兩組測資都達到題目規定之 PSNR 且完成 Synthesis，且 Gate-Level Pre-layout Simulation 結果正確。

c、 兩組測資都達到題目規定之 PSNR 且完成 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. 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

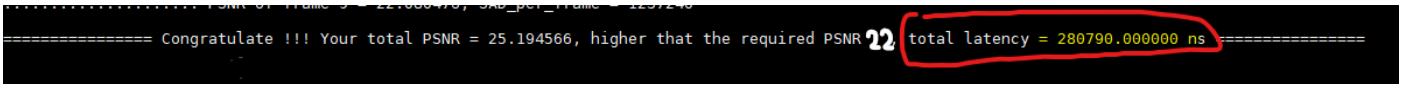


等級 A 之評分方法：

Score = Time*Area

例如：

本範例(Innovus)為，



Score = Time *Area = 280790.0*333039.25

註: Score 越小者，同級名次越好!

✧ **等級 B： 務必達成下述三項要求**

a、 兩組測資都達到題目規定之 PSNR，即 RTL 模擬結果正確。

b、 兩組測資都達到題目規定之 PSNR 且完成 Synthesis，且 Gate-Level Pre-layout Simulation 結果正確。

c、 完成 APR，但 DRC/LVS 有部分錯誤。

等級 B 之評分方法：

DRC/LVS 總錯誤數量越小越好，相同者比 Score

Score = Time*Area

註: Area 係以 Design Compiler 之 Cell Area 為主，Time 係以 Gate-Level Simulation 之 Latency 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、 兩組測資都達到題目規定之 PSNR，即 RTL 模擬結果正確。

b、 兩組測資都達到題目規定之 PSNR 且完成 Synthesis，且 Gate-Level Pre-layout Simulation 結果正確。

等級 C 之評分方法：

Score = Time*Area

註: Time 係以 Gate-Level Simulation 之 Latency 為主；Area 係以 Synthesis 後的 cell area 為主。Score 越小者，同級名次越好!

✧ **等級 D：務必達成下述兩項要求**

a、 兩組測資都達到題目規定之 PSNR，即 RTL 模擬結果正確。

Score = Time

註: 等級 D，只以 RTL Simulation Latency 為主，Score 越小者，同級名次越好。

附錄

附錄 A 為所提供各同學的設計檔案說明；附錄 B 為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，亦即同學必須繳交的資料評分用檔案；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. 下表為所提供各位同學的設計檔

表 3、設計檔案說明

檔名	說明
01_RTL/MEMC.v	本題之設計檔，已包含系統 Input/Output 之宣告。
01_RTL/tb1.v 01_RTL/tb2.v	本題共計 兩個 TestBench，分別呼叫兩組 pattern。
Patterns/**pattern_MrPickles_gray_4s_sr5.hex** Patterns/**pattern_RickandMorty_gray_2s_sr5.hex**	作為 MEMC 電路模擬時之 frames 的輸入訊號。註：該檔案已被加入到 TestBench。
SRAM/sram_OOOx8 相關檔案	Memory 相關檔案，每個 size 的資料夾都有以下檔案 sram_OOOx8.pdf sram_OOOx8.vclef sram_OOOx8.vp (此為加密過的 behavior model) sram_OOOx8.ant.lef sram_OOOx8_slow_syn.db sram_OOOx8_slow_syn.lib
02_SYN/.synopsys_dc.setup	使用 Design Compile 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。
02_SYN/MEMC_DC.sdc	Design Compiler 作合成之 Constraint 檔案，該檔案僅供參考，同學可依電路實際情形，調整 Constraints。 注意：環境相關參數請勿更改。
03_GATE/ tsmc13.v	For Gate-level simulation
~~04_APR/MEMC_APR.sdc~~	**APR 用.sdc 請由 synthesis 後產生之.sdc file 修改。 (之後 APR 課程會教)**

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

```
module MEMC(
      clk,
      rst_n,
      pixel_valid,
      pixel,
      busy,
      mv_valid,
      mv,
      mv_addr
);

input clk;
input rst_n;
input pixel_valid;
input [7:0] pixel;
output busy;
output mv_valid;
output [7:0] mv;
output [5:0] mv_addr;

endmodule
```

2. 本題共兩個 Patterns，批改時助教不會使用隱藏 Pattern。

3. 本題所提共的 TestBench 檔案，有多增加幾行特別用途的敘述如下：

```
`define SDFFILE     "MEMC_syn.sdf"
`ifdef SDF
      initial $sdf_annotate(`SDFFILE, u_MEMC);
 `endif
```

註：

1. SDF 檔案，請自行修改 SDF 時計檔名及路徑後在模擬
2. 在 Test Bench 中，助教提供`ifdef SDF 的描述，是為了讓本 Test Bench 可以做為 RTL 模擬、合成後模擬與 Layout 後模擬皆可使用。注意：當同學在合成或 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** tb1.v MEMC_syn.v sram_512x8.vp tsmc13.v **+define+SDF** +access+r

附錄 B 測試樣本

兩組十個 frames 之 Pixel 訊號，已存於 patterns資料夾中，每個pattern中有超過 10x3072 個 Pixels，但各組只需使用前十張frames即可。

註：資料左側為十六進制。註：資料左側為十六進制。

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。

- 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. 依題目各項要求，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 Innovus 軟體說明 netlist 與 sdf 寫出步驟。

i. 使用 Cadence Innovus 者，執行步驟如下：

在 Innovus 視窗下點選：

“ File → Save → Netlist … ”

Netlist File	MEMC_pr.v
All other options	Default value

按 OK。

“ Timing → Extract RC… ” ，按 OK。

“ Timing → Write SDF… ”

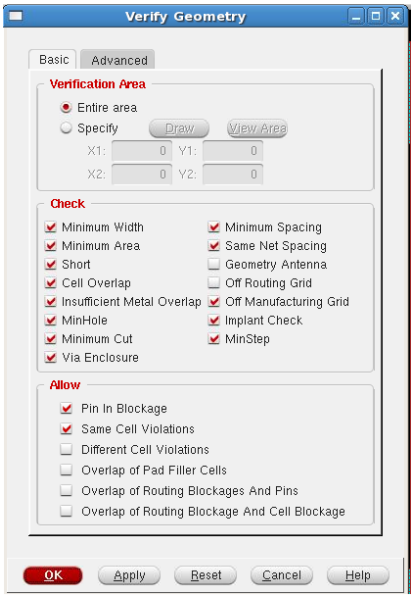
Ideal Clock	Disable
SDF Output File:	MEMC_pr.sdf

按 OK。

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

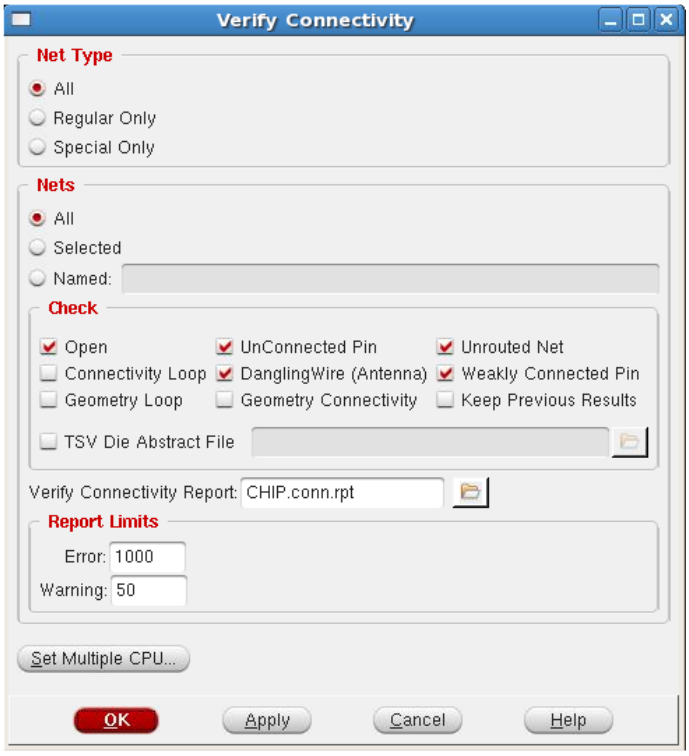
在 Innovus 視窗下點選

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



註：若 DRC 有發生錯誤，請選 ”**Tool → 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 ”**Tool → 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到的各 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 Cadence Innovus 者，請將 Innovus 相關的 design database（包含.enc

檔案與 and .enc.dat 目錄，簡單來說就是讓助教可以開啟 Innovus 介面重現各組設計的檔案），壓縮成一個檔案。壓縮的檔案格式如下：假設參賽者的 design database 目錄名稱為"teamID_lib" ，請執行底下的 UNIX 指令，最後可以得到" teamID _lib.tar"的檔案。注意: ID 數字部分請用兩碼。ex. 若第一組，則為 team01

- tar cvf teamID_lib.tar teamID_lib

範例: tar cvf team01_lib.tar team01_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		
Design Stage	*File*	*Description*
N/A	`N/A`	Design Report Form
RTL Simulation	`*.v or *.vhd`	Verilog (or VHDL) synthesizable RTL code
Gate-Level category		
Design Stage	*File*	*Description*
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		
Design Stage	*File*	*Description*
P&R	`*.tar`	**archive of the design database directory**
	`*.gds`	GDSII layout
	`DRC/LVS report`	截圖 DRC/LVS Report 檔案! 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. 建立一個新目錄，名稱叫做**< teamID_vk>** 例如：

   > ***mkdir team01_v1***

2. 將附錄 D 要求的檔案複製到 < teamID_vk>這個目錄。例如:

   > ***cp MEMC.v teamID_v1***

   > ***cp MEMC_syn.v teamID_v1***

   …….

3. 在Design Report Form 中，填入所需的相關資訊。

4. **上傳 < teamID_vk> 資料夾至以下 FTP 的 /final_project/ 資料夾中。**

FTP IP	140.112.175.174
Account	1081CVSD_student
Password	ilovecvsd

期末專題重要時限

- **01/12 (Sun)　before 23:59　Final Project Submission Deadline**
- **01/16 (Thur)　14:20-17:20　Project Presentation**