

2011 University/College IC Design Contest

Cell-Based IC Design Category for Graduate Level

Local Median Filter Engine

1.問題描述

請完成一 Local Median Filter Engine(後文以 **LMFE 表示**)的電路設計。此電路可以將任意輸入的二維(2D)灰階(Gray Level)影像訊號，做中間值濾波器(Median Filter)的運算，並將運算後影像的每一像素(Pixel)之結果，循序輸出。有關 LMFE 詳細規格將描述於後。

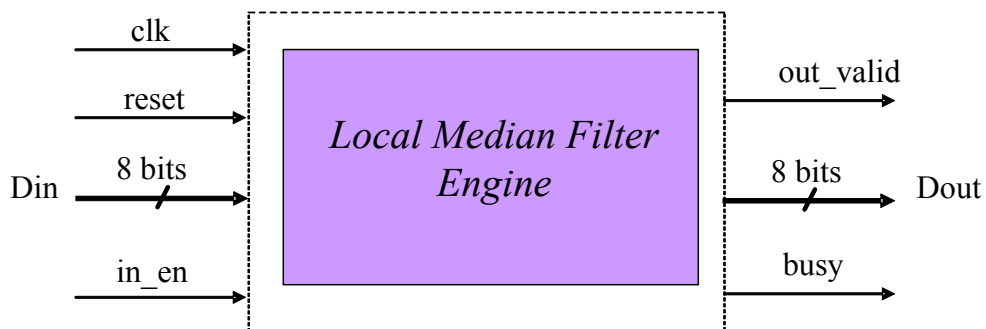
本電路各輸入輸出信號的功能說明，請參考表一。每個參賽隊伍必須根據下一節所給的設計規格及附錄 A 中的測試樣本完成設計驗證。

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/icc2011/gcb/icc2011cb.tar***，請執行以下指令取得測試樣本：

```
tar xvf /usr/cad/icc2011/gcb/icc2011cb.tar
```

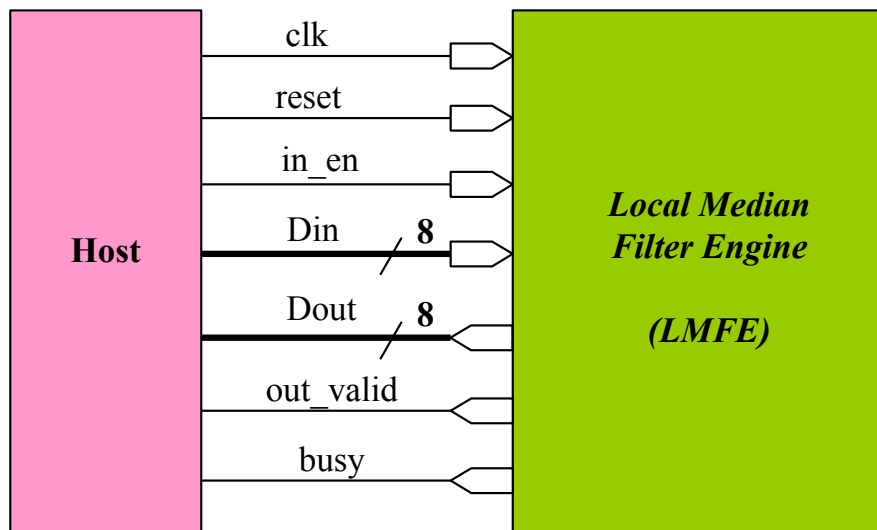
軟體環境及設計資料庫說明請參考附錄 F 與附錄 G。



圖一、Local Median Filter Engine 之方塊圖

2.設計規格

2.1 系統方塊圖



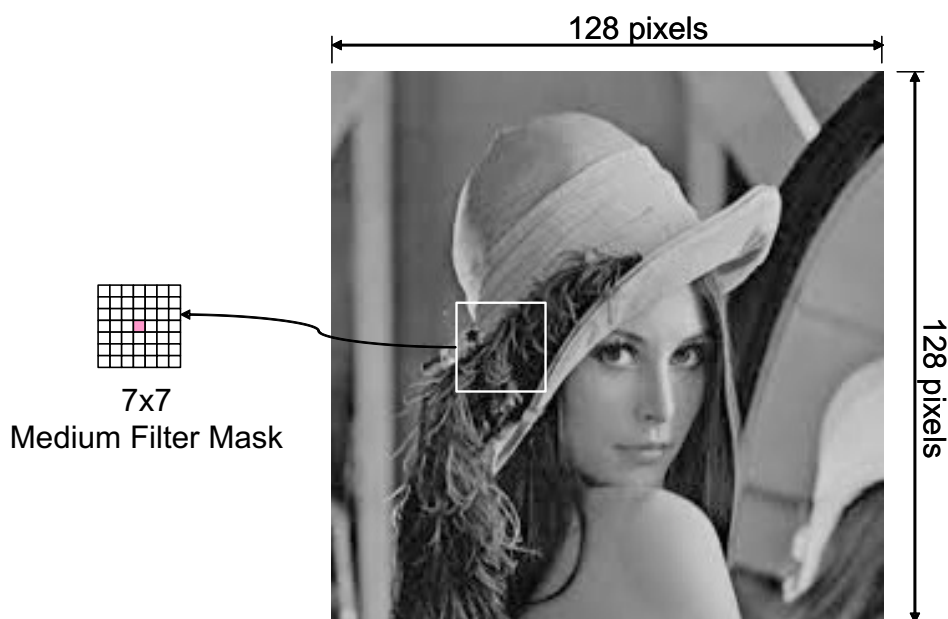
圖二、系統方塊圖

2.2 輸入/輸出介面

表 1 -輸入/輸出訊號

Signal Name	I/O	Width	Simple Description
clk	I	1	本系統為同步於時脈正緣之同步設計。 (註: Host 端採 clk ”負”緣時送資料。)
reset	I	1	高位準”非”同步(active high asynchronous)之系統重置信號。
in_en	I	1	資料輸入致能控制訊號。當 Host 端有資料要輸入時，此時 busy 訊號為 Low，in_en 就會為 High，反之，busy 訊號為 High 時，in_en 就會為 Low。當 Host 端所有資料送完後，該訊號到模擬結束前將永遠維持為 Low。
Din	I	8	LMFE 資料輸入的匯流排。Host 端會透過此匯流排將整張完整影像的訊號進行輸入。每一個週期僅能輸入一個 Pixel 值，且輸入過的 Pixel 值無法再重複輸入一次。註：輸入順序請參照 2.3.1。
busy	O	1	LMFE 忙碌之控制訊號。當為 High 時，表示系統正處於忙碌階段，告知 Host 端，暫時停止 Din 資料的輸入；反之，當為 Low 時，表示告知 Host 端可繼續由 Din 輸入資料。

out_valid	O	1	輸出資料有效之控制訊號。當為 High 時，表示目前輸出的資料為有效的；反之，當為 Low 時，表示目前輸出資料為無效的，即不被採用。
Dout	O	8	LMFE 資料輸出的匯流排。當 LMFE 計算完畢後，可透過此匯流排將运算完畢的 Pixel 值，輸出至 Host 端。 注意：每一個週期僅能輸出一筆 Pixel 值。



圖三、灰階影像資料

2.3 系統描述

2.3.1 輸入端灰階影像訊號

LMFE 電路只針對灰階影像訊號做處理，如圖三所示，主辦單位所提供之每張灰階影像尺寸固定為 128x128，共計 16384 個 Pixels，每個 Pixel 皆以 8 位元表示，故其訊號範圍為 0 ~ 255。影像輸入順序，固定為由左而右，由上而下的順序(即如圖四所示，影像訊號輸入順序按照方格中之編號 0,1,2,3,4,……,16382,16383)，循序輸入至參賽者的 LMFE 電路中。

注意：

1. 任何一個 Pixel 之影像訊號只能讀取一次，並無反覆讀取之功能
2. 讀取過程中，參賽者若想暫停影像訊號輸入，可以透過 Busy 控制訊號拉為 High 即可，待處理適宜後，再將 Busy 控制訊號拉為 Low 即可繼續進行影像資料之讀取。

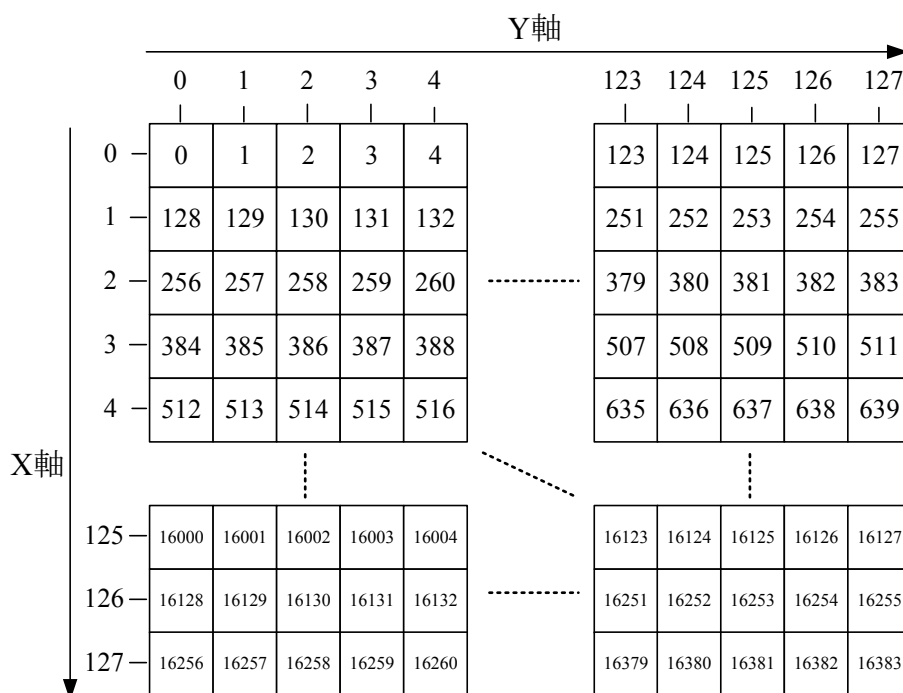
2.3.2 LMFE 電路運算方法

LMFE 電路運算方法係使用 $N \times N$ 遮罩(Mask)來框住影像資料之局部區域(主辦單位在此規定本題 Mask 為固定 7×7 之大小)，將被框住的範圍內 $N \times N$ 個 Pixels 中，找出其中間值來取代原影像點。如圖五所示，以 $N=3$ 為例， 3×3 Mask 座標軸中心(X 軸, Y 軸) = (2, 2)，被框住的九個 Pixels 影像訊號值經排列後可找出中間值為”147”，所以新的影像座標(2, 2)這點的值由”147”取代之。倘若被 $N \times N$ Mask 框住的影像訊號超出影像邊界範圍外，請參賽者自行補 0，再行計算。如圖六所示，以 $N=3$ 為例， 3×3 Mask 座標軸中心(X 軸, Y 軸) = (0, 0)，共有五個 Pixels 的訊號超出範圍外，將這些 Pixel 補上 0 後再行排列後，可找出中間值為”0”。

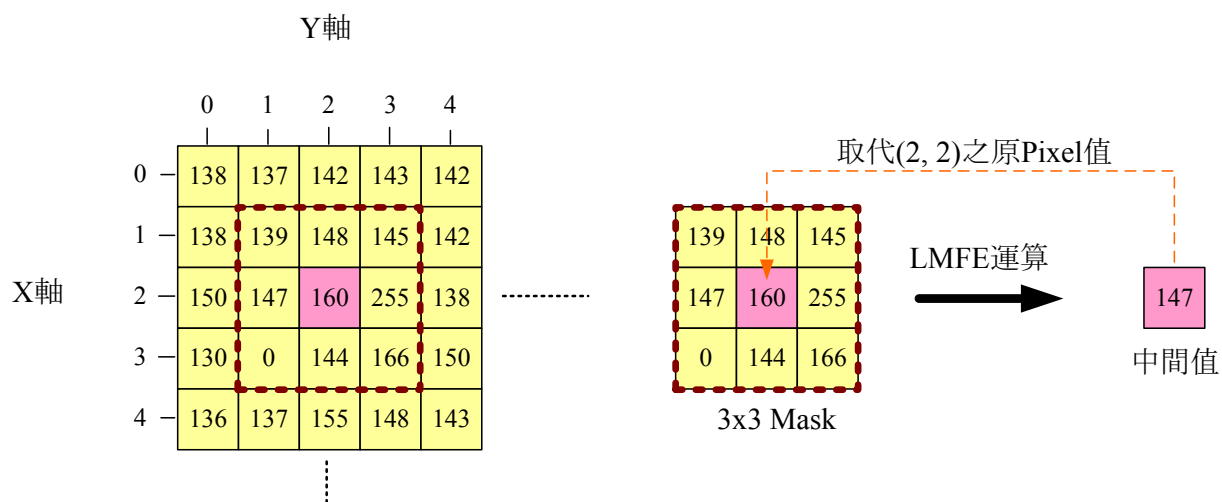
運算過程中， $N \times N$ Mask 中心之座標點可以由參賽者自行決定最恰當之順序，直到整張影像計算完畢。

注意：

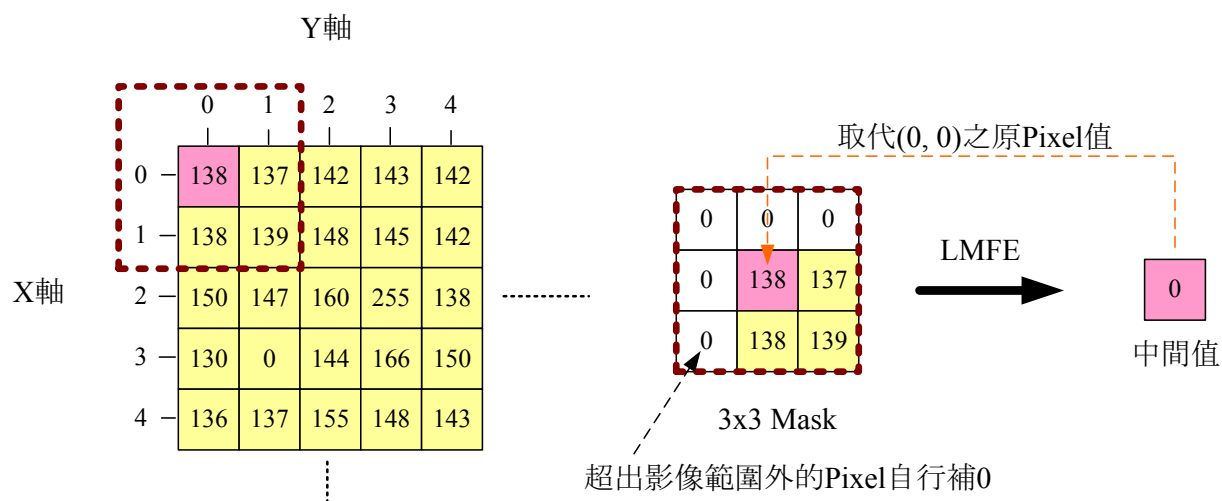
1. 主辦單位在此規定本題 Mask 為固定 7×7 之大小，即固定 $N=7$ 。



圖四、影像座標與影像輸入/輸出掃描順序圖



圖五、3x3 Mask 座標軸中心(2, 2)之範例



圖六、3x3 Mask 座標軸中心(0, 0)之範例

2.3.3 LMFE 電路輸出順序

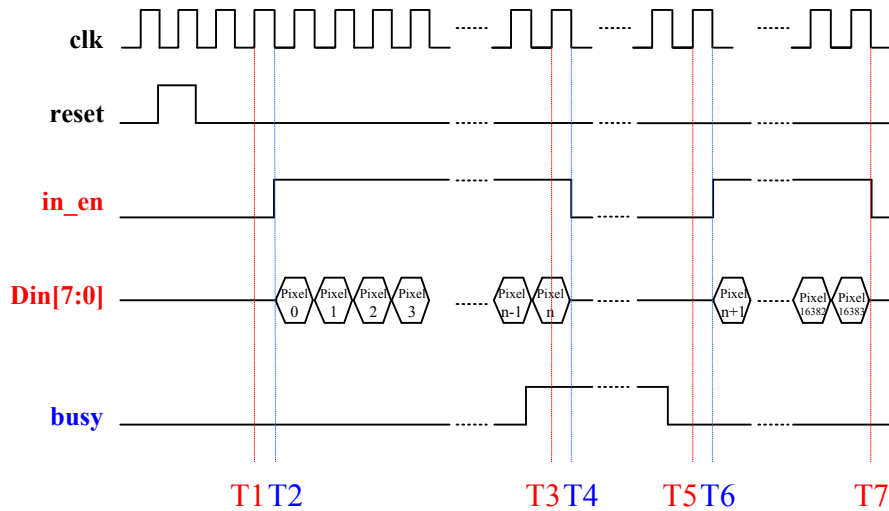
LMFE 輸出順序：主辦單位在此規定請參賽者按照如圖四方格中之編號 0, 1, 2, 3, ..., 16383 依序送出。當該 Pixel 值輸出是有效的，可將"out_valid"訊號拉為 High，反之則為 Low。

注意：

1. 參賽者可自行決定，每運算出一個 Pixel 結果值立即輸出或整張影像都完成後再一併輸出。
2. 每一個週期只能輸出一個 Pixel 值。
3. 同一個座標點之 Pixel 值不可重覆送第二次以上。

2.4 電路時序規格

2.4.1 LMFE 電路輸入時序規格



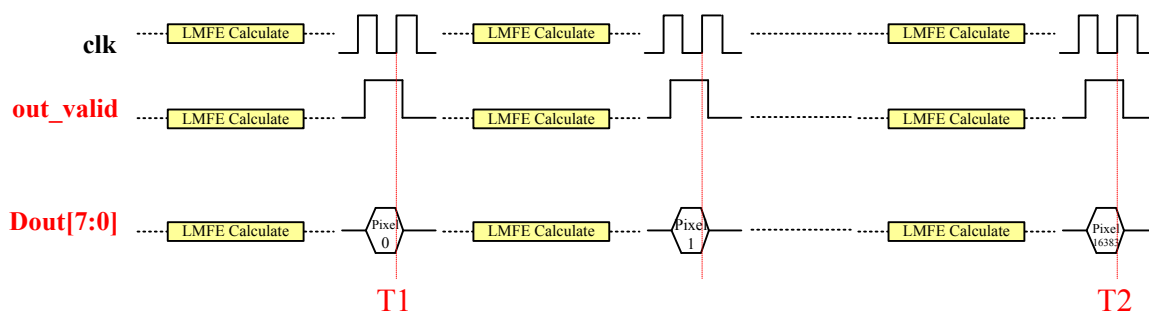
圖七(a)、LMFE 電路輸入之時序圖

1. LMFE 電路初始化，Reset 一個 Cycle 的時間。
2. T1 時間點，Reset 後等待兩個 Cycle 的時間，Host 於 T1 時間點判斷 busy 訊號為 Low，因此 T2 時間點，開始送出第一筆 Pixel 訊號。
3. T3 時間點，是假設參賽者欲暫停影像訊號輸入，可將 busy 拉為 High，Host 於 T3 時間點判斷 busy 為 High，Host 於 T4 時間點開始暫停影像訊號輸入。Host 沒有送訊號期間，Din 會一直維持在高阻抗 (High Impedance)，in_en 維持在 Low。
4. T5 時間點，Host 端判斷 busy 為 Low，於 T6 時間點繼續輸入剩餘的影像 Pixel 訊號。

註: T3~T6 時間點，僅提供給需要暫停影像訊號輸入之參賽者參考範例，並非強制要中斷讀取。

5. T7 時間點，Host 已完成輸入 16384 筆影像訊號，T7 時間點以後，Din 會一直維持在高阻抗 (High Impedance)，in_en 維持在 Low。

2.4.2 LMFE 電路輸出之時序規格



圖七(b)、LMFE 電路輸出之時序圖

參賽者可讀取局部或全部影像訊號後，開始進行 LMFE 運算。在輸出部分，參賽者可自行決定，每運算出一個 Pixel 結果值立即輸出或整張影像都完成後再一併輸出。

1. T1 時間點，參賽者已完成 Pixel0(即如圖四，輸出順序編號 0)之計算，輸出至 Host，並控制 out_valid 訊號為 High，剩餘之影像訊號依此類推，請參照 2.3.3 之規定順序輸出。
2. T2 點，參賽者已經將該影像最後一筆 Pixel16383 計算完畢後輸出，本模擬立即結束。

2.4.3 SRAM 記憶體規格與時序規格

製作 LMFE 電路，參賽者應該會用到 Single Port SRAM 記憶體。主辦單位在此提供兩種 SRAM 資源(Resource)，分別是 1KB (1024 x 8 bits)、8KB (8192 x 8 bits)記憶體，參賽者可依自己需求挑選最適當數量。例如：某參賽者需用到 16KB SRAM，可以在硬體描述語言描述中，將 8KB SRAM 呼叫兩顆或 1KB SRAM 呼叫 16 顆使用。由於本試題最後要做晶片佈局(Layout)，SRAM 也需加入一同 Layout，主辦單位建議 SRAM 使用數量要事先算清楚。

有關 1KB 及 8KB 記憶體細節規格與記憶體讀寫時序圖，詳如記憶體附件中。

注意：

1. 假如參賽者有需要用到 1Kbyte 以上記憶體，建議”不要使用 Register (Flip-Flop)來製作記憶體”，以免造成合成 or Layout 時，恐怕會跑很久甚至有如當機的現象發生，請注意!

3. 評分標準

評分方式會依設計完成程度，分成 A、B、C 三種等級，排名順序為 A>B>C，每個等級之評分項目有兩個項目，分別為**模擬時間**、**面積**，主辦單位會依此兩項目做為同等級之評分。另外，**請參賽者提供一組正確的週期時間(CYCLE TIME)**給評分人員驗證本電路之正確性。

✧ 評分項目一：依”模擬時間”(Time)長短評分

各參賽隊伍將 APR 完成後，執行 Gate-level Post-layout Simulation 模擬完後，會出現模擬時間，評分人員會以此模擬時間如下面範例，紀錄成 **Time = 16711721NS** 做評分。

```
-----
Congratulations! All data have been generated successfully!
-----PASS-----
Simulation complete via $finish(1) at time 16711721 NS + 0
```

✧ 評分項目二：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} {1138.820 683.870}
=> Area = 1138.82 x 683.87 = 778804.83 um2
```

2. SOC Encounter Report Area 範例:

```
encounter > analyzeFloorplan
***** Analyze Floorplan *****
Die Area(um^2)           : 778804.83
Core Area(um^2)          : 769066.50
Number of instance(s)    : 10132
Number of Macro(s)       : 2
Number of IO Pin(s)      : 21
Number of Power Domain(s) : 0
***** Estimation Results *****
=> Area = 778804.83 um2
```

設計完成程度三種等級，如下：

✧ 等級 A：完成設計

達成等級 A 之三項要求

- a、功能正確，RTL 模擬與 Golden Pattern 比對完全正確。
- b、完成 Synthesis，且 Gate-Level Pre-layout Simulation 結果正確。
- c、**完成 APR，並達成 APR 必要項目(註一)**，Gate-Level Post-layout Simulation 結果正確。

註一：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，不必做 interleaving。
- iii. 不要加 Dummy Metal。
- iv. Power Stripe 務必至少加一組，方向橫向或垂直皆可，其 VDD、VSS 寬度各設定為 2um。
- v. Power Rail(Standard cell pin connect)務必要加。
- vi. Core Filler 務必要加。
- vii. 若有使用 SRAM，其 VDD、VSS Pin 務必要連接至 Core Power Ring。
- viii. APR 後之 GDS 檔案務必要產生。
- ix. 完成 APR，DRC/LVS 完全無誤。

等級 A 之評分方法：

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

例如：

在前一頁範例中， $\text{Score} = \text{Time} \times \text{Area} = 16711721 \times 778804.83 = 1301516903241243$

註：Score 越小者，同級名次越好！

✧ 等級 B：DRC/LVS 有 10 個(含)以下錯誤

列入等級 B 之要求：與等級 A 相同，唯獨容許 DRC/LVS 錯誤總數量 10 個(含)以下
此等級之成績計算方式如下：

$$\text{Score} = \text{Time} \times \text{Area} \times (\text{DRC} + \text{LVS 的錯誤總數量})$$

註：Score 越小者，同級名次越好！

✧ 等級 C：僅正確做到合成階段

列入等級 C 之要求：未達到等級 A、B 之要求，但已正確做到合成階段

此等級之成績計算方式如下：

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

註：

1. Score 越小者，同級名次越好！
2. 等級 C，視 APR 為 Fail，Area 以 Design Compiler 所 Report 的 Cell Area 為主。
3. 等級 C，視 APR 為 Fail，Time 以 Gate-level Pre-layout Simulation 為主。

附錄

附錄 A 為主辦單位所提供各參賽者的設計檔說明；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 2、設計檔案說明

檔名	說明
LMFE.v	參賽者所使用的設計檔，已包含系統輸/出入埠之宣告
pattern1.dat pattern2.dat	兩組測試樣本，每組提供 16384 個 Pixels 的影像訊號，即一張完整影像訊號。欲模擬 pattern1 的結果，請務必搭配 testfixture1.v 檔案；模擬 pattern2 的結果，請務必搭配 testfixture2.v 檔案。註：兩組測試樣本皆以十六進制表示。
golden1.dat golden2.dat	兩組 Golden Pattern。每組提供一張影像經過 LMFE 運算後的影像訊號值，共計 16384 筆。註：兩組 Golden Pattern 皆以十六進制表示。
testfixture1.v testfixture2.v	兩個 Test Bench 檔案。每個 Test Bench 已自動加入對應的 pattern 檔、golden 檔。
.synopsys_dc.setup	使用 Design Compiler (DC)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，自行填上 Search Path 的設定。注意：無論合成或 APR，只需使用 worst case library；例如：slow.db、sram_1024x8_t13_slow_syn.db、sram_8192x8_t13_slow_syn.db 等 library。
LMFE_DC.sdc	使用 DC 作合成之 sdc 檔。參賽者可自行調整 cycle 值。
LMFE_SOCE.sdc	使用 SOC Encounter (SOCE)作 Layout 之 sdc 檔。參賽者可自行調整 cycle 值。
LMFE_ICC.sdc	使用 IC Compiler (ICC)作 Layout 之 sdc 檔。參賽者可自行調整 cycle 值。

sram_1024x8_t13.v	1KB SRAM Verilog 模擬用之檔案。
sram_8192x8_t13.v	8KB SRAM Verilog 模擬用之檔案。
sram_1024x8_t13_slow_syn.db	1KB SRAM DC 合成與 ICC Layout 用之檔案。
sram_8192x8_t13_slow_syn.db	8KB SRAM DC 合成與 ICC Layout 用之檔案。 注意：無論 DC 合成或用 ICC 做 APR，只需使用 worst case library，例如：slow.db、sram_1024x8_t13_slow_syn.db。
sram_1024x8_t13/	1KB SRAM FRAM ICC Layout 用之檔案。
sram_8192x8_t13/	8KB SRAM FRAM ICC Layout 用之檔案。
sram_1024x8_t13.vclef	1KB SRAM SOCE Layout 用之檔案。
sram_8192x8_t13.vclef	8KB SRAM SOCE Layout 用之檔案。
sram_1024x8_t13.gds	1KB SRAM gds 檔案。(產生 LMFE gds 檔案時可用)
sram_8192x8_t13.gds	8KB SRAM gds 檔案。(產生 LMFE gds 檔案時可用)
sram_1024x8_t13_slow_syn.lib	1KB SRAM SOCE Layout 用之檔案。
sram_8192x8_t13_slow_syn.lib	8KB SRAM SOCE Layout 用之檔案。 注意：用 SOCE 做 APR，只需使用 worst case library 例如：slow.lib、sram_1024x8_t13_slow_syn.lib。

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

```

module LMFE ( clk, reset, Din, in_en, busy, out_valid, Dout);
input    clk;
input    reset;
input    in_en;
output   busy;
output   out_valid;
input    [7:0]  Din;
output   [7:0]  Dout;
endmodule

```

- 主辦單位提供兩個 Test Bench 檔案 testfixture1.v, testfixture2.v 分別對應到 pattern1.dat、pattern2.dat 與 golden1.dat、golden2.dat，這些都已加入了，參賽者只需要注意 pattern 檔案的路徑即可。

例如：

第一個 Test Bench 模擬，使用 testfixture1.v：

```

`define PAT          "./pattern1.dat"
`define EXP          "./golden1.dat"

```

註：參賽者無須作修改，只需注意 pattern1.dat 的檔案位置即可，預設為”目前目錄”。

3. 主辦單位所提供的兩個 Test Bench 檔案，多加敘述如下：

```
`define End_CYCLE 10000000
`define SDFFILE    "/LMFE.sdf"
`ifdef SDF
    initial $sdf_annotate(`SDFFILE, LMFE);
`endif
```

註：

1. End_CYCLE 預設 1000 萬個 Cycles，其目的可以防止參賽者因電路有錯，模擬陷入無窮回圈之境，倘若參賽者確定模擬需超過 1000 萬個 Cycles 以上，可自行再加大此 Cycle 數。
2. SDF 檔案，請自行修改 SDF 實際檔名後模擬。
3. 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、Layout 後，使用 NC-Verilog 模擬第一組樣本，在 UNIX 下執行下面指令

```
> ncverilog testfixture1.v LMFE_syn.v -v tsmc13_neg.v +define+SDF +access+r
```

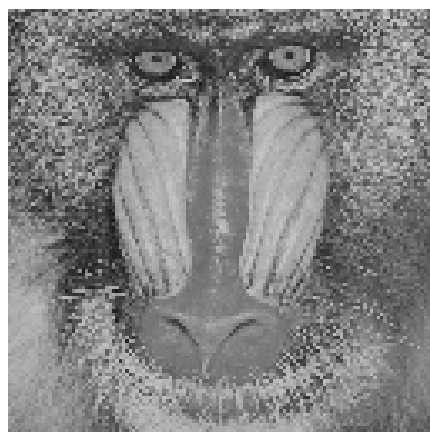
註：使用 NC-Verilog 執行 Gate-level 模擬時間甚久，為方便參賽者辨識目前模擬狀況，主辦單位提供於模擬時，比對每 1000 個 Pixels 資料正確，即秀出一次正確的訊息，其訊息如下，最後模擬到出現”Congratulations! All data have been generated successfully!”訊息表示該影像檔經 LMFE 運算比對結果完全正確。

```
ncsim> run
Output pixel: 0 ~ 1000 are correct!
Output pixel: 0 ~ 2000 are correct!
Output pixel: 0 ~ 3000 are correct!
Output pixel: 0 ~ 4000 are correct!
Output pixel: 0 ~ 5000 are correct!
Output pixel: 0 ~ 6000 are correct!
Output pixel: 0 ~ 7000 are correct!
Output pixel: 0 ~ 8000 are correct!
Output pixel: 0 ~ 9000 are correct!
Output pixel: 0 ~ 10000 are correct!
Output pixel: 0 ~ 11000 are correct!
Output pixel: 0 ~ 12000 are correct!
Output pixel: 0 ~ 13000 are correct!
Output pixel: 0 ~ 14000 are correct!
Output pixel: 0 ~ 15000 are correct!
Output pixel: 0 ~ 16000 are correct!

-----
Congratulations! All data have been generated successfully!
-----PASS-----
Simulation complete via $finish(1) at time 16711721 NS + 0
```



(a)



(b)

圖八、兩張真實 128x128 灰階影像之測試樣本(a) Lena 圖，(b) Baboon 圖

附錄 B 測試樣本

主辦單位提供兩組真實影像之測試樣本，如圖八(a)、(b)所示。為了讓參賽者看完題目後，更能明確題意，主辦單位在此以 Lena 影像訊號之測試樣本為例，如圖九(a)所示，為原始 Lena 影像訊號(由於篇幅有限，在此僅秀出 Lena 最左上方影像 7x7 之大小作範例)，經過 7x7 Mask 做 LMFE 運算後可以得到新 Lena 影像訊號如圖九(b)所示。

	0	1	2	3	4	5	6		0	1	2	3	4	5	6
0	138	138	140	131	134	130	132	7X7 Mask LMFE運算後	0	0	0	130	130	130	130
1	137	136	136	132	131	134	131		1	0	131	132	132	131	131
2	133	133	134	135	133	134	129		2	0	132	133	133	132	132
3	134	132	132	133	131	129	130		3	131	133	134	134	133	134
4	134	134	134	134	135	132	132		4	132	133	134	134	134	134
5	135	130	134	134	135	134	126		5	132	134	134	134	134	135
6	129	130	134	135	135	135	131		6	132	134	134	134	135	137

(a)

(b)

圖九、Lena 128x128 灰階影像訊號(a)原始影像訊號，(b) LMFE 運算後新影像訊號值

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，模擬結果必須於參賽者提供之 CYCLE 數值下，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、SOC 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	LMFE_pr.sdf
Significant digits	2

按 。

對應指令： write_sdf -version 2.1 LMFE_pr.sdf

“ File > Export > Write Verilog...”

先按

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

按 。

- ii. 使用 Cadence SOC Encounter 者，執行步驟如下：

在 SOC Encounter 視窗下點選：

“ Design → Save → Netlist...”

Netlist File	LMFE_pr.v
All other options	Default value

按 。

“ Timing → Extract RC...”，按 。

“ Timing → Calculate Delay...”

存成 LMFE_pr.sdf，按 。

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、SOC 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 ...”

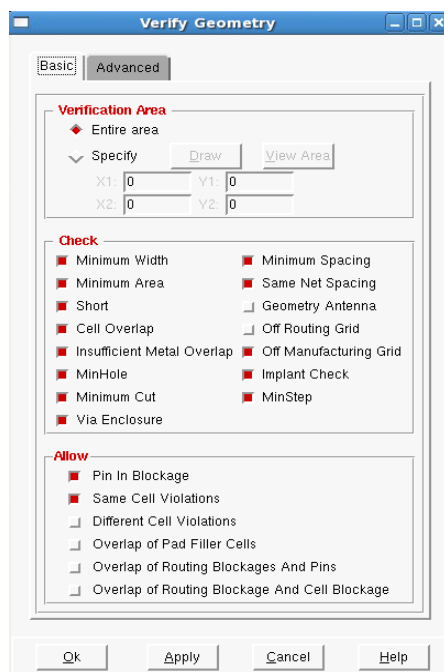
Pins not connected to a wire segment(Floating port)	disable
All other options	Default value

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。

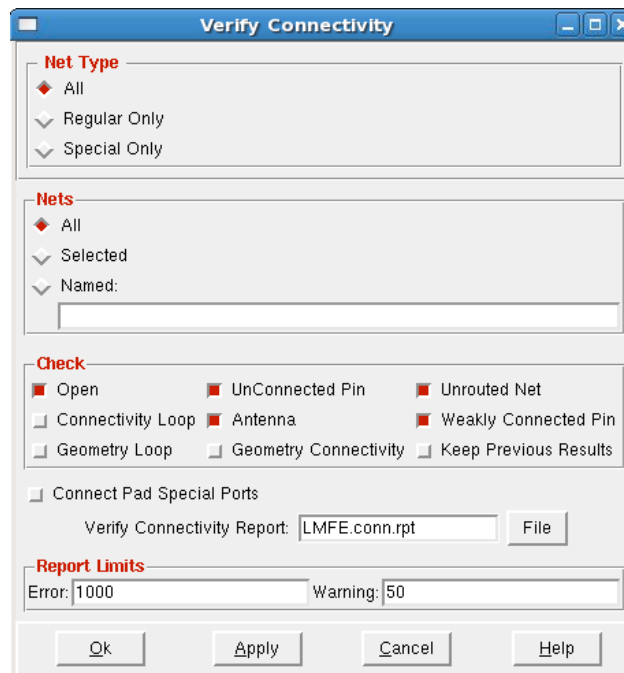
- ii. 使用 Cadence SOC Encounter 者，驗證 DRC 與 LVS 步驟如下：
在 SOC Encounter 視窗下點選

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



註：若 DRC 有發生錯誤，請選“Verify → 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“Verify → 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 database**，壓縮成一個檔案。使用 **Cadence SOC Encounter** 者，請將 SOC 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 目錄名稱為“final”，請執行底下的 UNIX 指令，最後可以得到“final.tar”的檔案。

```
> tar cvf final.tar final
```

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 3

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	APR 存檔 database (page 16)
	*.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 SOC 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 3 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 LMFE.v ~/result/
```

```
> cp LMFE_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. 主辦單位已將所有軟體環境設定於:/usr/cad/cshrc/env.cshrc，參賽同學不需再做任何設定。

2. 在 env.cshrc 所設定好的軟體環境包括：

NC-Verilog

NC-VHDL

VCS

Verdi

Laker

ModelSim

Design Compiler (Design Vision)

IC Compiler

SOC Encounter

gedit / nedit

vi / vim

gvim

gnome-calculator (工程計算機執行檔，可開啟 View -> Scientific mode)

gcc

EDA 軟體所須使用的 license 皆已設定完成，不須額外設定，且每組限定**每個 EDA 軟體只能使用一套 license**。

附錄 G 設計資料庫

設計資料庫位置： /usr/cad/icc2011/CBDK_IC_Contest_v2.1

目錄架構

ICC/	tsmc13gfsg_fram/ tsmc13_CIC.tf macro.map tluplus/ t013s8mg_fsg_typical.tluplus t013s8mg_fsg.map	ICC core library ICC technology layer mapping file t13 tluplus file t13 tluplus mapping file
SOCE/	lef/ tsmc13fsg_8lm_cic.lef antenna_8.lef lib/ slow.lib streamOut.map	LEF for core cell LEF for antenna worst case for core cell Layout map for GDSII out
SynopsysDC/	db/ slow.db lib/ slow.lib	Synthesis model (slow) timing and power model
Verilog/	tsmc13_neg.v	Verilog simulation model
VHDL/	tsmc13.vhd	VHDL simulation model
Memory/	sram_1024x8_t13/ sram_1024x8_t13_slow_syn.db sram_1024x8_t13_slow_syn.lib sram_1024x8_t13.v sram_1024x8_t13.vclef sram_1024x8_t13_ant.lef sram_1024x8_t13/ sram_1024x8_t13.pdf sram_1024x8_t13.gds sram_8192x8_t13/ sram_8192x8_t13_slow_syn.db sram_8192x8_t13_slow_syn.lib sram_8192x8_t13.v sram_8192x8_t13.vclef sram_8192x8_t13/ sram_8192x8_t13_ant.lef sram_8192x8_t13.pdf sram_8192x8_t13.gds	Synthesis model (worst-case) SOCE APR 可使用 Verilog simulation model SOCE APR 可使用 SOCE 可使用 (LEF for antenna) ICC 可使用 (1K SRAM Fram View) SRAM Spec Document SRAM gds file Synthesis model (worst-case) SOCE APR 可使用 Verilog simulation model SOCE APR 可使用 ICC 可使用 (8K SRAM Fram View) SOCE 可使用 (LEF for antenna) SRAM Spec Document SRAM gds file

Design Report Form

隊號 (Team number)		登入帳號 (login-id)	
RTL category			
<i>Design Stage</i>	<i>Description</i>	<i>File Name</i>	
RTL Simulation	使用之 HDL 名稱 (請填入 Verilog 或 VHDL)		
	RTL 檔案名稱 (RTL Netlist file name)		
Gate-Level category			
<i>Design Stage</i>	<i>Description</i>	<i>File Name</i>	
Pre-layout Gate-level Simulation	Gate-Level 檔案名稱 (Gate-Level Netlist file name)		
	Pre-layout sdf 檔案名稱		
	Gate-Level simulation, 所使用的 CYCLE Time (請確定模擬功能正確)	() ns	
Physical category			
<i>Design Stage</i>	<i>Description</i>	<i>File Name or Value</i>	
P&R	使用之 P&R Tool (請填入 IC compiler 或 SOC Encounter)		
	APR 存檔 database (*.tar)		
	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)		
Post-layout Gate-level Simulation	Gate-Level 檔案名稱 (Gate-Level Netlist file name)		
	Post-layout sdf 檔案名稱		
	Post-layout Simulation 所使用的 CYCLE Time (請確定模擬功能正確) Ex: 10ns		
	Post-layout Simulation Time (Simulation Time, ex: 167117 ns) Time = ?		
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			