

# Integrated Circuit Design

## Final Project - Image Processing

Due: 2017/06/26 17:00

### 1. 問題描述

本題輸入為一灰階影像(如圖 1.所示)，此灰階影像存放於 Host 端的灰階圖像記憶體模組 (gray\_mem)中，**imgproc** 端須發送訊號至 **Host** 端以索取灰階影像資料，再對灰階影像中每個 pixel 各自進行獨立運算，**運算後的結果請寫入 Host 端的記憶體模組 (my\_mem)**內，並在整張影像訊號處理完成後，將 finish 訊號拉為 High，接著系統會自動進行比對整張影像資料的正確性。

有關 imgproc 的過程描述於後。



圖 1. 灰階影像範例

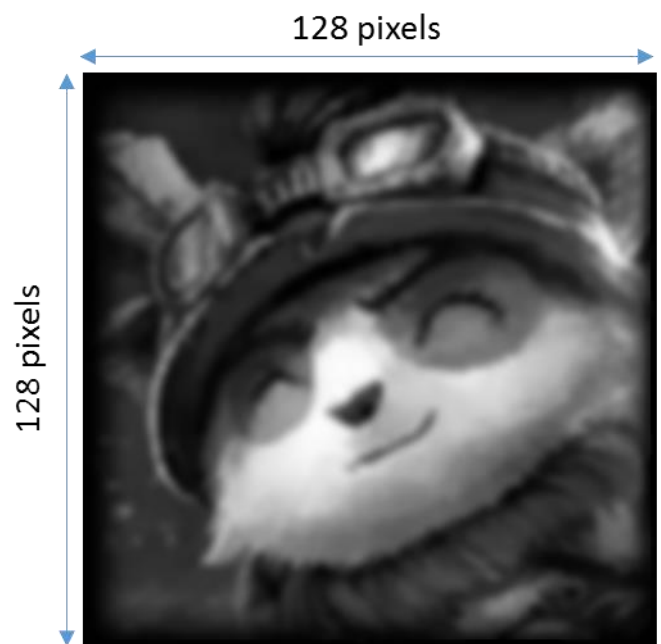


圖 1. 輸出灰階影像範例

## 2. 設計規格

### 2.1 系統方塊圖

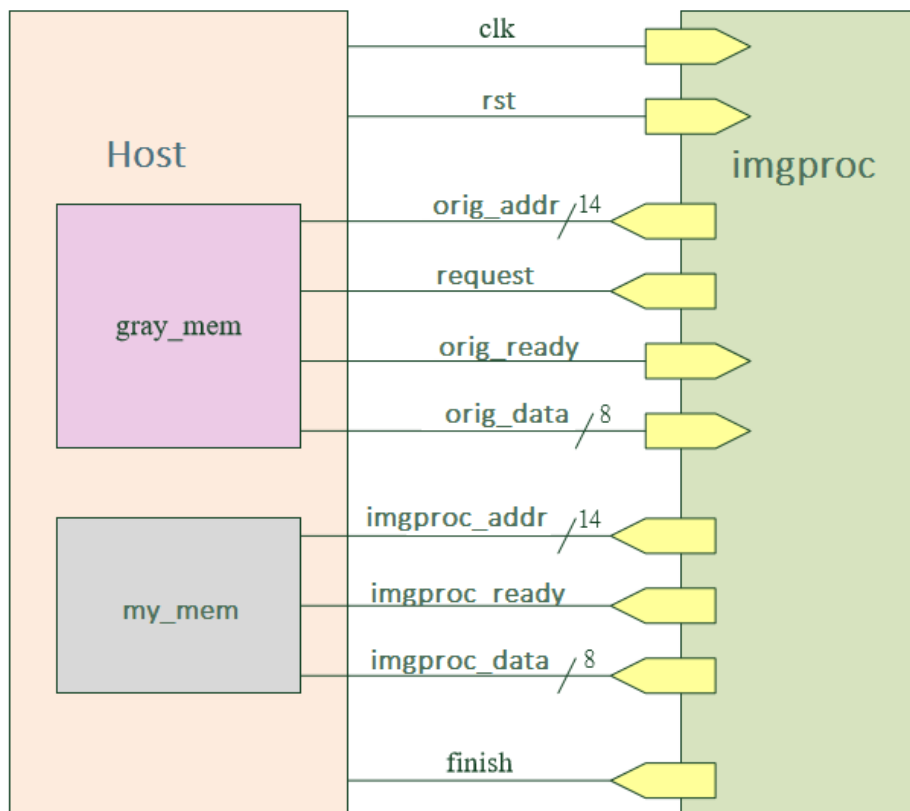


圖 2. 系統方塊圖

### 2.2 輸出入訊號和記憶體描述

Signal Name	I/O	Width	Simple Description
clk	input	1	positive edge clock system
rst	input	1	active high asynchronous system reset signal
orig_addr	output	14	address for input image data
request	output	1	ask for input image data
orig_ready	input	1	input image data is ready for your system
orig_data	input	8	Input image data
imgproc_addr	output	14	send your output data to this address
imgproc_ready	output	1	tell host that output data is ready
imgproc_data	output	8	output image data
finish	output	1	whole processing is finished

※ PS:

1. **orig\_addr** 和 **request** 為一組，一起發送出去，跟 HOST 要圖片。
2. **orig\_ready** 和 **orig\_data** 為一組，一起傳回來，看到 **orig\_ready** 代表 **orig\_data** 可以接收了。
3. **imgproc\_addr**, **imgproc\_ready**, **imgproc\_data** 為一組，一起發送出去，將結果送到 HOST 端。
4. **finish** 訊號拉起來之後，開始檢查正確性。

## 2.3 系統功能描述

本電路功能為當 reset 結束後，imgproc 端才可開始對 Host 端進行動作。當 Host 端在每個時脈訊號負緣觸發時若偵測到 finish 訊號為 Low 且 request 訊號為 High 時表示 imgproc 端對 Host 端要求索取灰階圖像資料。當 orig\_ready 訊號為 High，此時 Host 端準備好資料，會依 orig\_addr 匯流排所指示的位址將灰階圖像記憶體內的位址資料由 orig\_data 匯流排輸入 imgproc 端。

**本電路主要功能是將影像作高斯模糊，將影像通過 5\*5 的 Gaussian blur filter，小數點部分做四捨五入。**

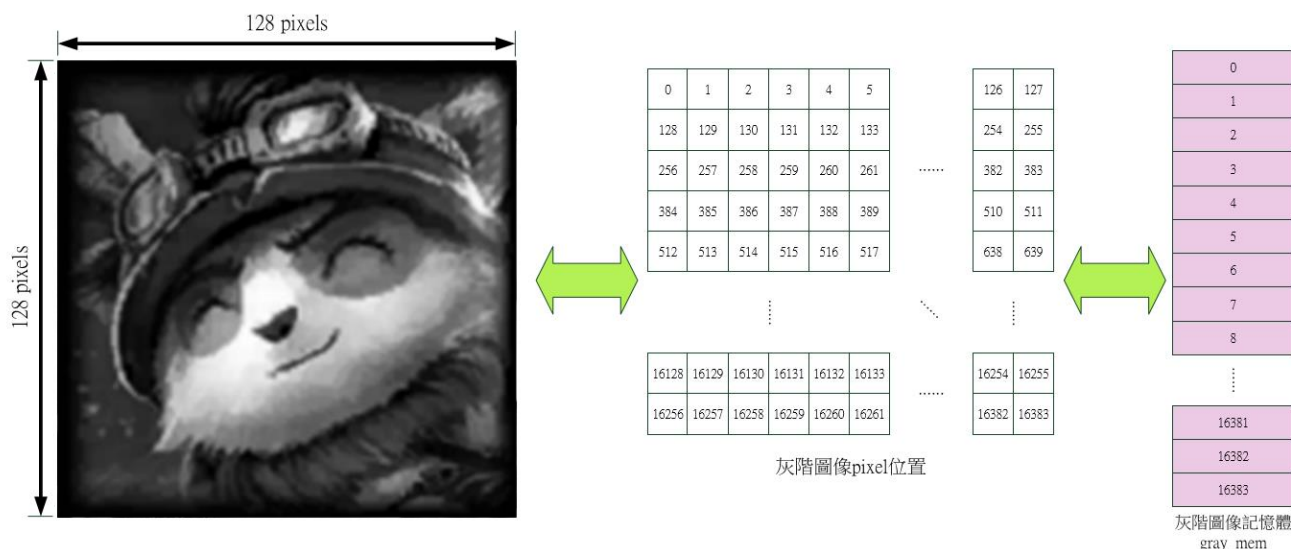
**請以 23bit 進行運算 8bit 整數+15bit 小數，Gaussian blur 之各點參數數值可以參考附錄。**

記憶體模組的寫入方式如下，當 Host 端在每個時脈訊號負緣觸發時若偵測到 imgproc\_ready 訊號為 High 時，就會將目前 imgproc\_data 匯流排上的內容，寫入到 imgproc\_mem 記憶體模組的 imgproc\_addr 匯流排所指示的位址內，當所有 pixel 都處理完畢後，請將 finish 訊號拉為 High，接著 Host 端就會開始進行結果驗證。

**由於影像邊界在 filter 的處理上較複雜，非本次作業重點，因此 Host 端僅針對輸出影像中央 124x124 pixels 進行結果驗證。**

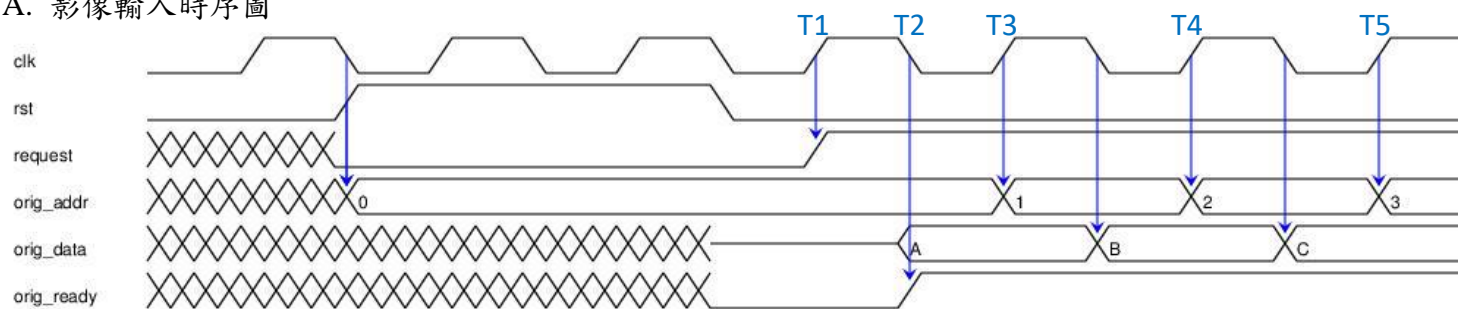
## 2.4 灰階圖像記憶體對應方式

灰階圖像大小固定為 128x128 pixels，每個 pixel 為 unsigned 8bit 灰階(每個 8bit 灰階圖像 pixel 的值介於 0 到 255 之間)，因此 Host 端的灰階圖像記憶體模組(gray\_mem)共有 16384 個位址用以存放各 pixel 的灰階圖像資料，圖像與記憶體模組的對應方式如下圖所示。

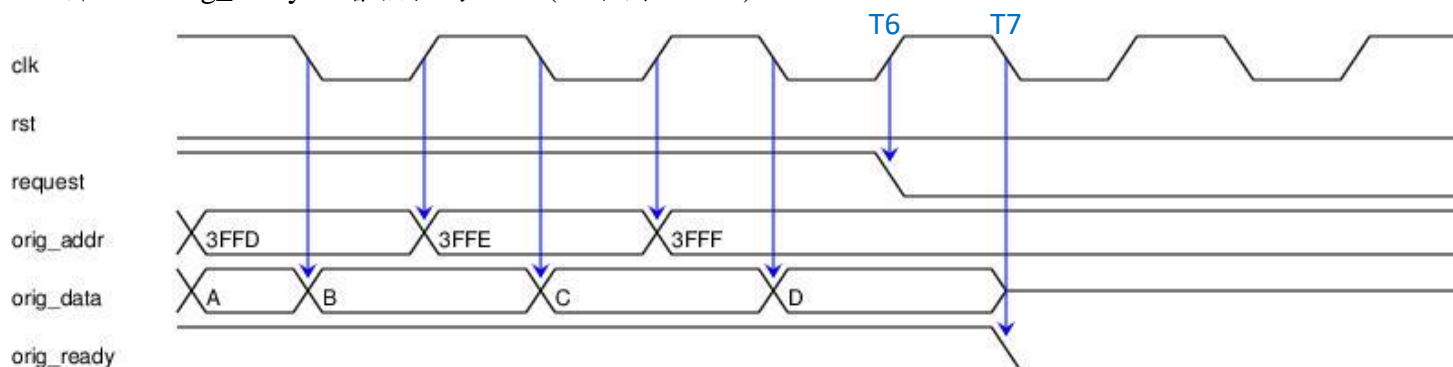


## 2.5 時序規格圖

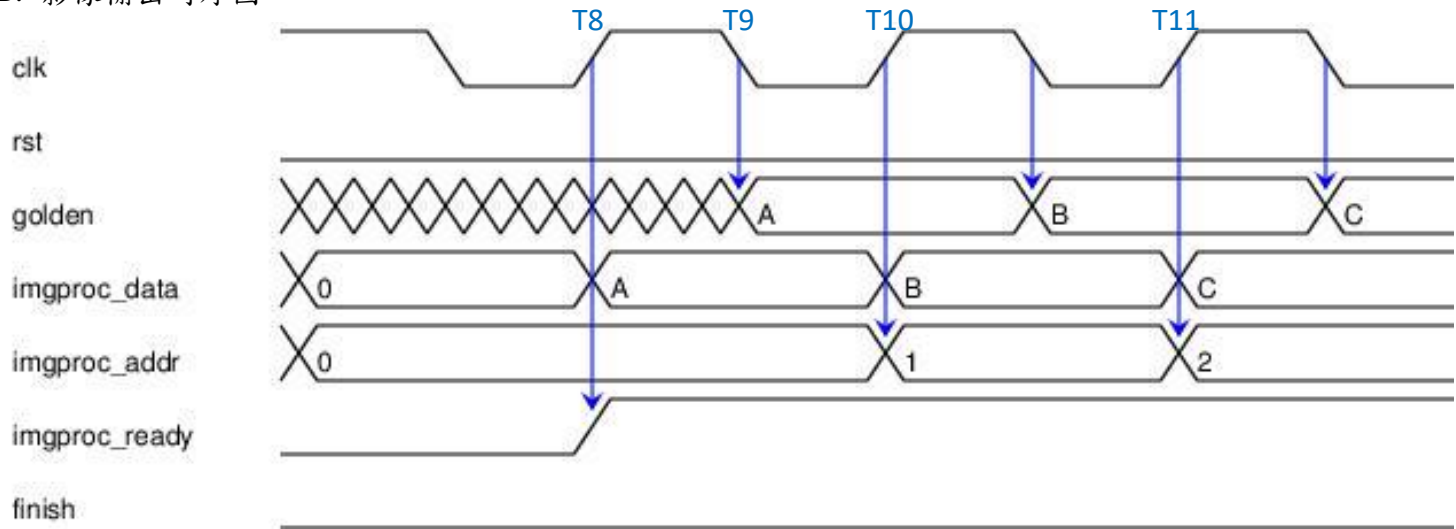
### A. 影像輸入時序圖



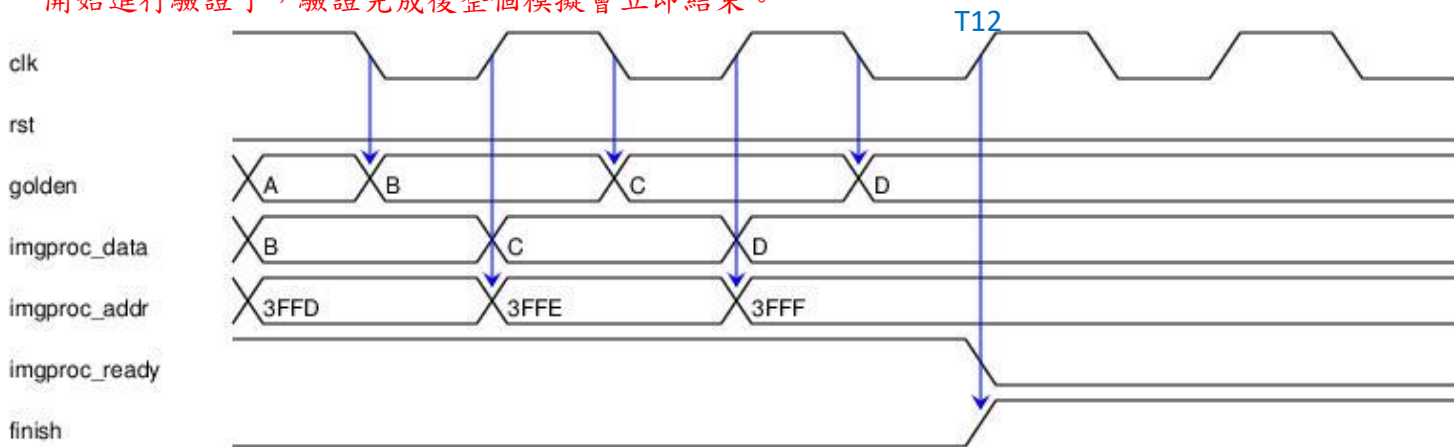
- a、reset 訊號持續兩個 Cycle 時間後，電路初始化結束。
- b、imgproc 端將 **request** 訊號拉為 High，並且同時將欲索取的灰階圖像 pixel 之位址由 **orig\_addr** 匯流排送出 (如上圖 T1)。
- c、Host 端在時脈訊號負緣觸發若偵測到 request 為 High，將 **orig\_ready** 拉為 High，則會將灰階圖像記憶體內的 orig\_addr 匯流排所指示位址的資料由 **orig\_data** 匯流排送到 imgproc 端 (如上圖 T2)。
- 若要進行連續索取，只需要將 request 維持在 High，並連續改變 orig\_addr 匯流排位址，就可在 orig\_data 匯流排連續得到該位址資料 (如上圖 T3-T5)。
- d、接著 imgproc 端就可以針對各 pixel 進行訊號處理流程。
- e、若 imgproc 端不想要對 Host 端索取任何位址資料，則只須在時脈訊號正緣觸發將 request 拉為 Low，則 Host 端在下個時脈訊號負緣觸發時就不會送出任何位址資料到 gray\_data 匯流排，且 orig\_ready 也會被拉為 Low (如下圖 T6-T7)。



## B. 影像輸出時序圖



- a、當 imgproc 端完成處理後，請將各 pixel 的處理結果寫入各相對應的記憶體位址中，其方式為將 **imgproc\_ready** 訊號拉為 High，同時把欲寫入的位址及資料分別放在 **imgproc\_addr** 及 **imgproc\_data** 匯流排 (如上圖 T8)；Host 在時脈訊號負緣觸發時，就會進行寫入的動作，golden[7:0]表示 Host 端在此時刻擷取到的資料 (如上圖 T9)。
- 若想要連續寫入的話，則只需要持續將 imgproc\_ready 維持在 High 後改變 imgproc\_data 及 imgproc\_addr 即可 (如上圖 T10-T11)。
- b、如果不想繼續寫入資料的話，將 imgproc\_ready 拉為 Low。
- c、所有的 pixel 都處理完成了，此時 imgproc 端須將 finish 拉為 High (如上圖 T12)。Host 端就會開始進行驗證了，驗證完成後整個模擬會立即結束。



### 3. 檔案說明

檔名	說明
testfixture.v	Testbench for RTL and post synthesis simulation
testfixture_apr.v	Testbench for post layout simulation
imgproc.v	<b>YOUR</b> design
*.f	Scripts for ncverilog
data/	Test and golden data
syn/	Directory for synthesis
layout/	Directory for layout
library/	Library for Encounter

#### syn 資料夾

檔名	說明
.synopsys_dc.setup	Settings for design compiler
imgproc.sdc	Standard design constraints
tsmc13_neg.v	Standard cell
run.tcl	Scripts for design compiler

#### Layout 資料夾

檔名	說明
imgproc_apr.sdc	Standard design constraints
imgproc.conf	Configuration for importing design
imgproc.view	Configuration for importing design
*.tcl	Scripts for encounter

#### A. RTL

- imgproc.v: 已經定義了 imgproc module 的輸入輸出，**將描述 imgproc 電路的 verilog code 寫在此檔案裡。**
- run\_RTL.f: 模擬時，直接執行 ncverilog -f run\_RTL.f 即可。

#### B. synthesis

- syn/imgproc.sdc: 描述了合成時對電路的 constraints，**只有 cycle time 可以自行修改，預設為 set cycle 20，代表 cycle time 為 20ns。**
- syn/run.tcl: 在 syn 資料夾內執行 dc\_shell -f run.tcl 即可完成 synthesis

(這份是基本版的 script，若有需要可以添加或修改。Ex: 如 compile\_ultra, compile - map\_effort high, optimize\_register...等)。



c. run\_DC.f: 模擬時，直接執行 `ncverilog -f run_DC.f` 即可。

### C. layout(此部分不計分，同學自行練習)

a. layout/imgproc\_apr.sdc: 描述了佈線時對電路的 constraints，只有 cycle time 可以自行修改，預設為 set cycle 20，代表 cycle time 為 20ns。

b. encounter 操作請大家參考 HW5 的說明。\*.tcl 為 encounter 操作之 scripts 檔

01\_pre\_floor.tcl 對應步驟 4

02\_floorplaned.tcl, 03\_placed.tcl 對應步驟 5 (timing analysis 請注意 WNS 並判斷需不需要 optimize)

04\_powerring.tcl 對應步驟 6

05\_powerstripe.tcl 對應步驟 7,8 (DRC check 請注意 layout 上是否有 DRC error)

06\_tiehilow.tcl 對應步驟 9(同樣注意一下 timing analysis 資訊)

07\_cts.tcl 對應步驟 10,11,12(注意 timing analysis 資訊)

08\_powerroute.tcl 對應步驟 13,14(確認是否有 violation 資訊)

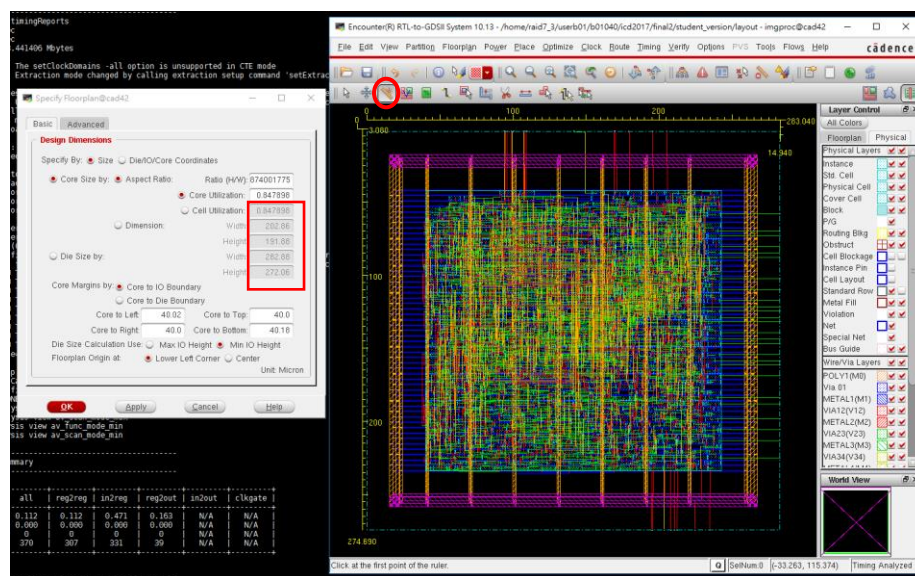
09\_route.tcl 對應步驟 15,16(同樣注意一下 timing analysis 資訊)

10\_output.tcl 對應步驟 17,18

進入 encounter 後，指令輸入 `source *.tcl` 即可使用該 scripts (Ex: `source 01_pre_floor.tcl`)

c. run\_apr.f: 模擬時，直接執行 `ncverilog -f run_apr.f` 即可。

d.



所有步驟完成後，點選上圖紅色圓圈處 create ruler，即可測量晶片 layout 後大小。  
點選 Floorplan -> Specify Floorplan，紅色方框處也可觀察 layout 大小。

#### 4. 評分標準

本次作業為兩人一組，繳交之報告分為兩部分，須包含以下內容。

##### A. RTL 部分

請將 ncverilog 模擬畫面截圖放入 Report 中。

僅完成此部分可得 60 分。

##### B. Gate level 部分

(1) 文字紀錄合成後的 cell area、cycle time、total time。

Ex:

Cell area = 31139  $\mu\text{m}^2$

Cycle time = 20 ns

Total time = 1895300 ns

Cell area\*Total time = 59017746700  $\mu\text{m}^2 \cdot \text{ns}$

本次作業評分依 Cell area\*Total time 大小排名，Cell area\*Total time 越小越好，分數分布在 70~100

(2) 截圖需要有 (A) desing compiler 的 area report 與 timing report。

(B) gate level netlist 再跑一次 ncverilog 模擬並截圖。(記得 test bench 中的 cycle time 要改成與合成時的 cycle time 一樣再跑模擬。)

```
*****
Report : area
Design : imgproc
Version: G-2012.06
Date   : Thu May 25 17:35:14 2017
*****

Library(s) Used:

    slow (File: /home/raid7_2/course/cvsd/CBDK_IC_Contest/CIC/SynopsysDC/db/slow.db)

Number of ports:          50
Number of nets:          1838
Number of cells:         1187
Number of combinational cells: 846
Number of sequential cells: 307
Number of macros:         0
Number of buf/inv:       127
Number of references:     94

Combinational area:      22636.526924
Noncombinational area:   8502.276485
Net Interconnect area:   228341.164062

Total cell area:         31138.803409
Total area:              259479.967472
,
```



\*\*\*\*\*

Report : timing  
 -path full  
 -delay max  
 -max\_paths 1

Design : imgproc  
 Version: G-2012.06  
 Date : Thu May 25 17:35:05 2017  
 \*\*\*\*\*

Operating Conditions: slow Library: slow  
 Wire Load Model Mode: top

Startpoint: d10\_reg\_1\_ (rising edge-triggered flip-flop clocked by clk)  
 Endpoint: imgproc\_data\_reg\_7\_ (rising edge-triggered flip-flop clocked by clk)  
 Path Group: clk  
 Path Type: max

Des/Clust/Port	Wire Load Model	Library
imgproc	tsmc13_wl10	slow

Point	Incr	Path
clock clk (rise edge)	0.00	0.00
clock network delay (ideal)	0.50	0.50
d10_reg_1_/CK (DFFX1)	0.00	0.50 r
d10_reg_1_/Q (DFFX1)	0.49	0.99 r
add_2_root_add_0_root_add_312_14/A_1_ (imgproc_DW01_add_16)	0.00	
add_2_root_add_0_root_add_312_14/U1_1/CO (ADDFXL)	0.80	0.99 r
add_2_root_add_0_root_add_312_14/U1_2/CO (ADDFXL)	0.50	1.79 r
add_2_root_add_0_root_add_312_14/U1_3/S (ADDFXL)	0.65	2.29 r
add_2_root_add_0_root_add_312_14/SUM_3_ (imgproc_DW01_add_16)	0.00	2.94 r
add_0_root_add_0_root_add_312_14/B_3_ (imgproc_DW01_add_8)	0.00	
add_0_root_add_0_root_add_312_14/U1_3/S (ADDFXL)	0.81	2.94 r
add_0_root_add_0_root_add_312_14/SUM_3_ (imgproc_DW01_add_8)	0.00	3.76 r
mult_312_3/a_3_ (imgproc_DW_mult_uns_3)	0.00	3.76 r
mult_312_3/U69/S (ADDFXL)	0.84	3.76 r
mult_312_3/product[3] (imgproc_DW_mult_uns_3)	0.00	4.60 r
add_4_root_add_0_root_add_312_24_U1_3/CO (ADDFXL)	0.86	4.60 r
add_4_root_add_0_root_add_312_24_U1_4/S (ADDFXL)	0.62	5.46 r
add_3_root_add_0_root_add_312_24/B_4_ (imgproc_DW01_add_3)	0.00	6.07 r
add_3_root_add_0_root_add_312_24/U1_4/S (ADDFXL)	0.88	6.07 r
add_3_root_add_0_root_add_312_24/SUM_4_ (imgproc_DW01_add_3)	0.00	6.95 f
add_2_root_add_0_root_add_312_24/B_4_ (imgproc_DW01_add_2)	0.00	6.95 f
add_2_root_add_0_root_add_312_24/U1_4/S (ADDFXL)	0.85	6.95 f
add_2_root_add_0_root_add_312_24/SUM[4] (imgproc_DW01_add_2)	0.00	7.80 f
add_1_root_add_0_root_add_312_24/B[4] (imgproc_DW01_add_1)	0.00	7.80 f
add_1_root_add_0_root_add_312_24/U1_4/S (ADDFXL)	0.96	7.80 f
add_1_root_add_0_root_add_312_24/SUM[4] (imgproc_DW01_add_1)	0.00	8.77 f
add_0_root_add_0_root_add_312_24/A_4_ (imgproc_DW01_add_0)	0.00	8.77 f
add_0_root_add_0_root_add_312_24/U27/Y (OAI21XL)	0.50	8.77 f
add_0_root_add_0_root_add_312_24/U26/Y (OAI2BB1X1)	0.27	9.27 r
add_0_root_add_0_root_add_312_24/U25/Y (OAI21XL)	0.48	9.55 f
add_0_root_add_0_root_add_312_24/U24/Y (OAI2BB1X1)	0.27	10.02 r
		10.30 f

add_0_root_add_0_root_add_312_24/U18/Y (OAI21XL)	0.48	10.78 r
add_0_root_add_0_root_add_312_24/U17/Y (OAI2BB1X1)	0.34	11.11 f
add_0_root_add_0_root_add_312_24/U23/Y (OAI2BB1X1)	0.39	11.51 f
add_0_root_add_0_root_add_312_24/U3/Y (OAI21XL)	0.45	11.95 r
add_0_root_add_0_root_add_312_24/U5/Y (OAI2BB1X1)	0.36	12.32 f
add_0_root_add_0_root_add_312_24/U16/Y (OAI21XL)	0.51	12.83 r
add_0_root_add_0_root_add_312_24/U9/Y (OAI2BB1X1)	0.28	13.11 f
add_0_root_add_0_root_add_312_24/U7/Y (OAI21XL)	0.29	13.39 r
add_0_root_add_0_root_add_312_24/U6/Y (OAI2BB1X1)	0.33	13.72 f
add_0_root_add_0_root_add_312_24/U1/Y (OAI21XL)	0.32	14.04 r
add_0_root_add_0_root_add_312_24/U22/Y (OAI2BB1X1)	0.25	14.29 f
add_0_root_add_0_root_add_312_24/U2/Y (OAI21X2)	0.22	14.51 r
add_0_root_add_0_root_add_312_24/U4/Y (OAI2BB1X1)	0.21	14.72 f
add_0_root_add_0_root_add_312_24/U21/Y (OAI21XL)	0.47	15.19 r
add_0_root_add_0_root_add_312_24/U20/Y (OAI2BB1X1)	0.20	15.39 f
add_0_root_add_0_root_add_312_24/U1_14/CO (ADDFXL)	0.52	15.91 f
add_0_root_add_0_root_add_312_24/U1_15/CO (ADDFXL)	0.54	16.44 f
add_0_root_add_0_root_add_312_24/U1_16/CO (ADDFXL)	0.29	16.74 f
add_0_root_add_0_root_add_312_24/U1_17/CO (ADDFXL)	0.25	16.98 f
add_0_root_add_0_root_add_312_24/U11/Y (NAND2X1)	0.18	17.16 r
add_0_root_add_0_root_add_312_24/U14/Y (NAND3X1)	0.19	17.35 f
add_0_root_add_0_root_add_312_24/U1_19/CO (ADDFXL)	0.51	17.87 f
add_0_root_add_0_root_add_312_24/U1_20/CO (ADDFXL)	0.53	18.39 f
add_0_root_add_0_root_add_312_24/U1_21/S (ADDFXL)	0.57	18.96 r
add_0_root_add_0_root_add_312_24/SUM_21_ (imgproc_DW01_add_0)	0.00	18.96 r
add_318_round/A[7] (imgproc_DW01_inc_10)	0.00	18.96 r
add_318_round/U1_1_7/CO (ADDFXL)	0.34	19.30 r
add_318_round/U5/Y (XOR2X1)	0.28	19.58 f
add_318_round/SUM_8_ (imgproc_DW01_inc_10)	0.00	19.58 f
U1137/Y (A022XL)	0.42	20.00 f
imgproc_data_reg_7_/D (DFFX1)	0.00	20.00 f
data arrival time		20.00
clock clk (rise edge)	20.00	20.00
clock network delay (ideal)	0.50	20.50
clock uncertainty	-0.10	20.40
imgproc_data_reg_7_/CK (DFFX1)	0.00	20.40 r
library setup time	-0.31	20.09
data required time		20.09
data required time		20.09
data arrival time		-20.00
slack (MET)		0.09

START!!! Simulation Start .....

=====  
 Congratulations!!! Every outputs are correct!  
 =====

Simulation complete via \$finish(1) at time 1895300 NS + 0  
 ./testfixture.v:117 \$finish;  
 ncsim> exit

## 5. 繳交檔案

本次作業為兩人一組，一組繳交一份檔案即可。

評分所需檔案可分為兩部份，請壓縮後命名為 **學號 1\_學號 2\_final**。

RTL category		
Design stage	File	Descirption
N/A	Report.pdf	Design Report
RTL Simulation	imgproc.v	Verilog synthesizable RTL code
Gate-Level category		
Pre-layout Gate-level Simulation	imgproc_syn.v	Verilog gate-level netlist generated by Synopsys Design Compiler
	imgproc_syn.sdf	Pre-layout gate-level sdf

## 6. 附錄

Gaussian blur filter 之各點參數值

23'h00001a	23'h0000d8	23'h0001b5	23'h0000d8	23'h00001a
23'h0000d8	23'h000707	23'h000e41	23'h000707	23'h0000d8
23'h0001b5	23'h000e41	23'h001ce8 center	23'h000e41	23'h0001b5
23'h0000d8	23'h000707	23'h000e41	23'h000707	23'h0000d8
23'h00001a	23'h0000d8	23'h0001b5	23'h0000d8	23'h00001a