Integrated Circuit Design Homework 2

Image Processing

1. 問題描述

本題輸入為一彩色影像(如圖 1.所示)，此彩色影像存放於 Host 端的彩色圖像記憶體模組(color_mem)中，imgproc 端須發送訊號至 Host 端以索取彩色影像資料，再對彩色影像中每個 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。

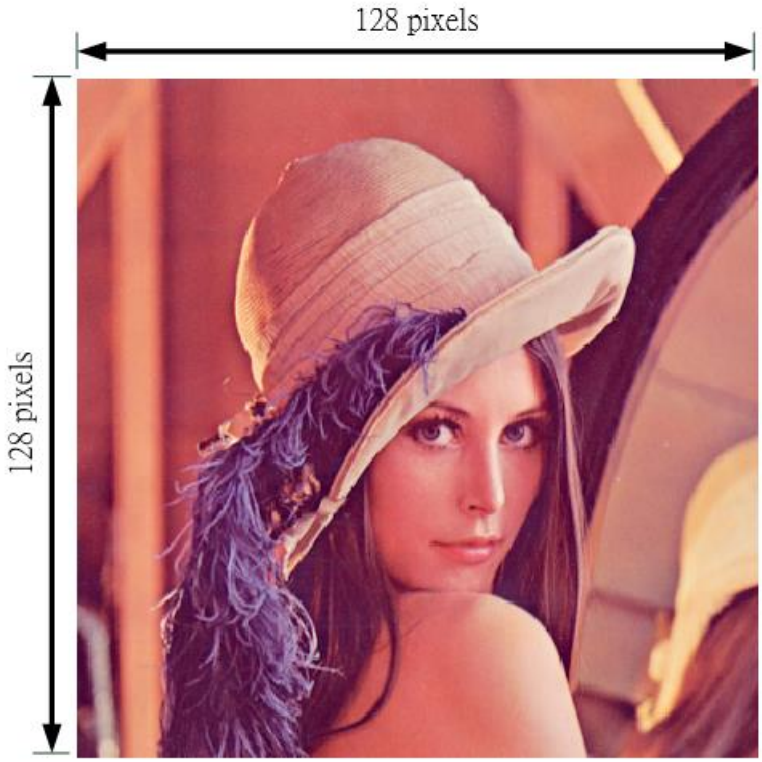


圖 1. 彩色影像範例

2. 設計規格

2.1 系統方塊圖

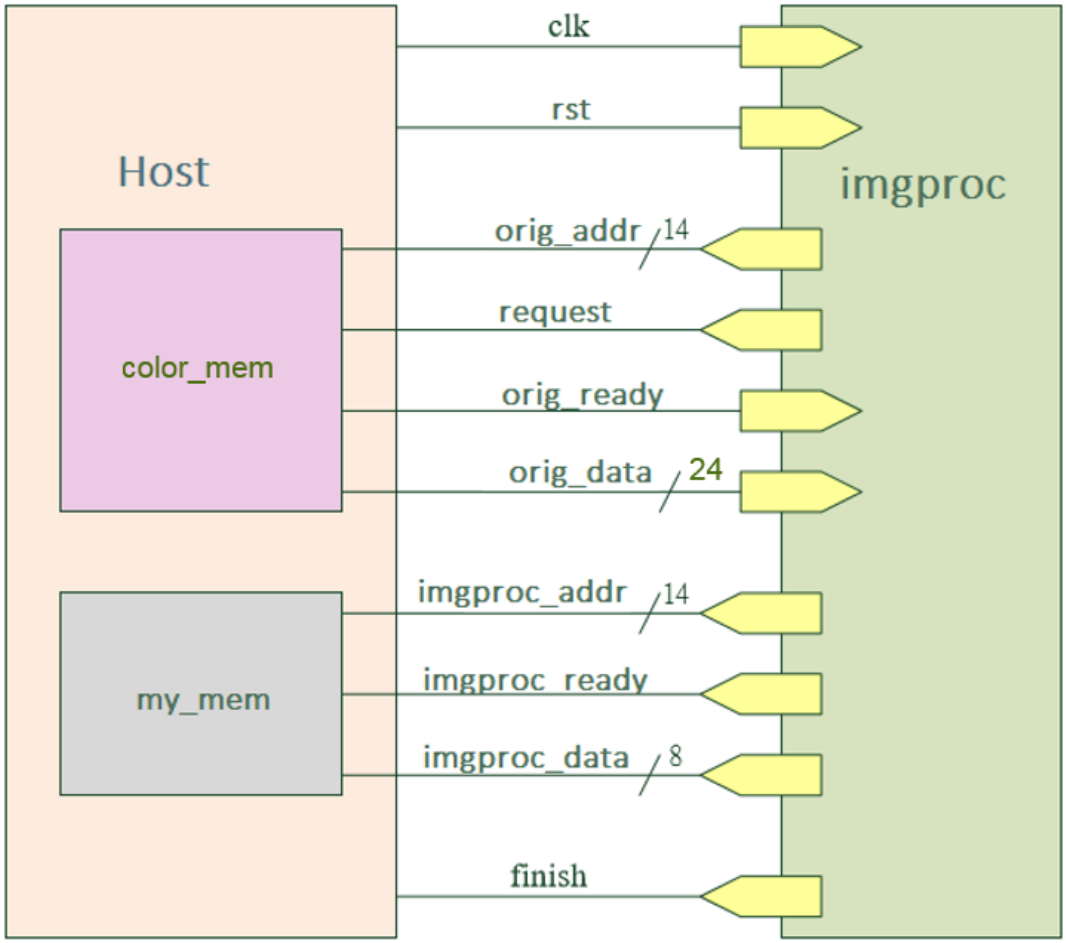


圖 2. 系統方塊圖

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	24	input color 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 dark channel 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 端要求索取彩色圖像資料。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彩色圖像記憶體內的位址資料由 orig _data 匯流排輸入 imgproc 端。

本電路主要功能是計算簡易版本的暗通道（dark channel），輸入影像的每個像素分別有 RGB 三個通道，imgproc 端找出三個通道中，數值最小的通道，並輸出此通道的數值。以圖 3.為例，應輸出之結果為 00010100。

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。

2.4 彩色圖像記憶體對應方式

彩色圖像大小固定為 128x128 pixels，每個像素均為 24 bits 彩色（每個 24 bits 彩色圖像像素的值，都分別由 R/G/B 三個 8 bits 通道組成，每個通道分別介於 0 到 255 之間），因此 Host 端的彩色圖像記憶體模組（color_mem）共有 16384 個位址用以存放各像素的彩色圖像資料，圖像位址與記憶體模組資料的對應及排列方式如下圖所示。

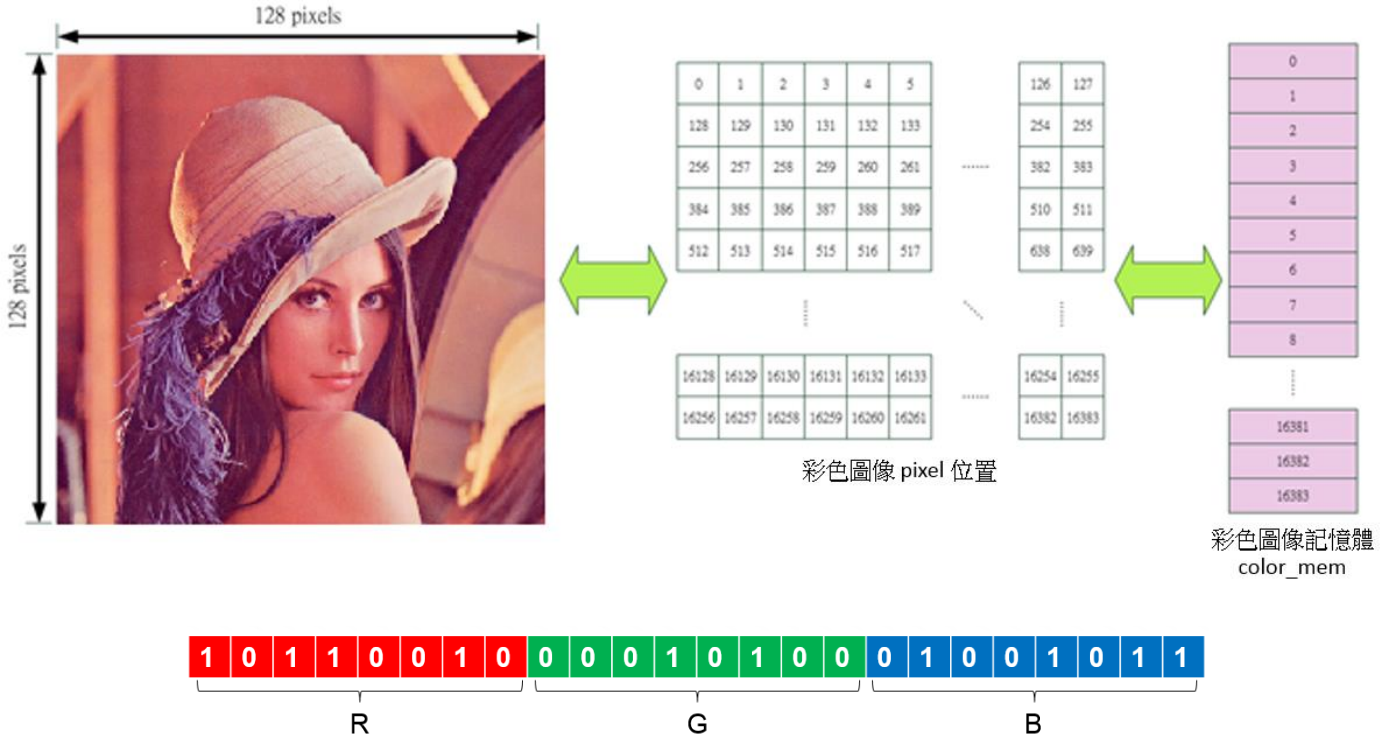
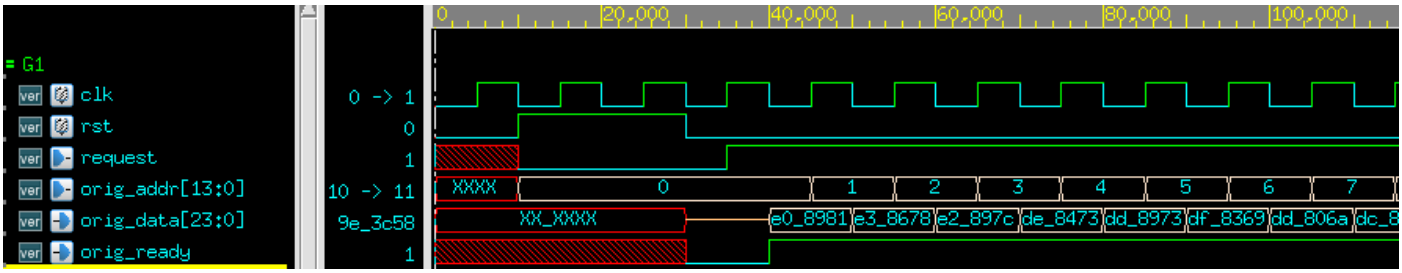


圖 3. 圖像位址與記憶體模組資料的對應及排列方式

2.5 時序規格圖

A. 影像輸入時序圖



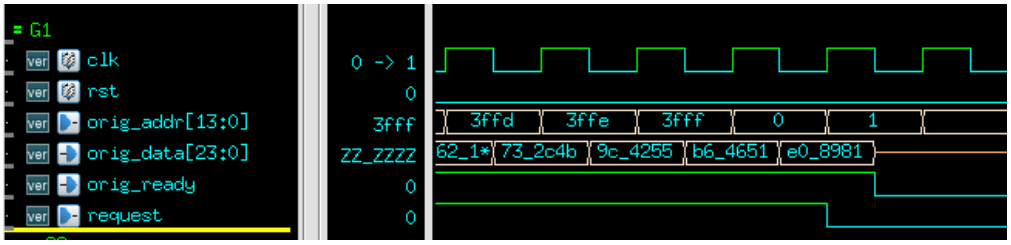
a、 reset 訊號持續兩個 Cycle 時間後，電路初始化結束。

b、 imgproc 端將 **request** 訊號拉為 High， 並且同時將欲索取的彩色圖像像素，其位址由 **orig_addr** 匯流排送出。

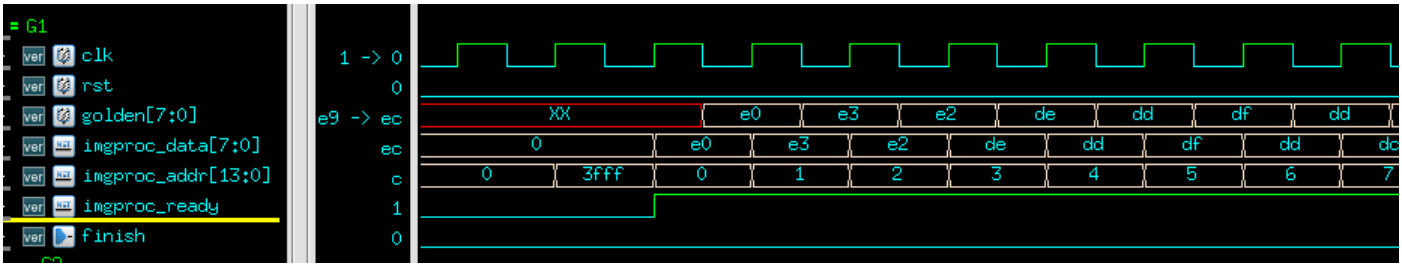
c、 Host 端在時脈訊號負緣觸發若偵測到 request 為 High，將 **orig_ready** 拉為 High，則會將彩色圖像記憶體內的 orig_addr 匯流排所指示位址的資料由 **orig_data** 匯流排送到 imgproc 端。若要進行連續索取，只需要將 request 維持在 High，並連續改變 orig_addr 匯流排位址，就可在 orig_data 匯流排連續得到不同位址資料。

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 color_data 匯流排，且 orig_ready 也會被拉為 Low。



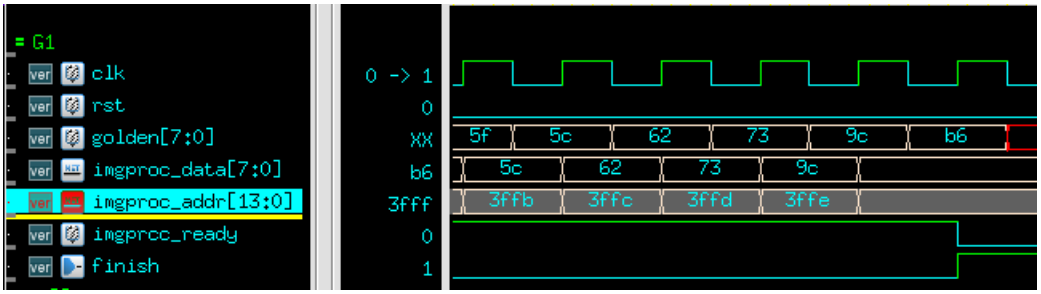
B. 影像輸出時序圖



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；Host 在時脈訊號負緣觸發時，就會進行寫入的動作。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。

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。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

image_process 資料夾中共有 3 個 .v 檔、syn 資料夾和 data 資料夾。

syn 資料夾中共有 3 個檔案，與 design compiler 相關。

data 資料夾中為測資。

A. RTL simulation 部分用到檔案

a. testfixture.v: 本次作業的 testbench，描述 HOST 端記憶體的行為。

b. 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。**

c. 模擬時，直接執行：ncverilog testfixture.v imgproc.v

若要記錄 debug 用模擬波形檔，請執行：

ncverilog testfixture.v imgproc.v +access+r +define+FSDB

d. data/ 資料夾: 本次作業的測資與 golden data。

B. synthesis 部分用到檔案

RTL 寫完後，請同學將電路合成 gate level netlist。

a. imgproc.v: 已完成的 RTL code。

b. syn/.synopsys_dc.setup: Design compiler 的基本設定檔，若 library 位置與預設不同，需修改此檔案內 set search_path 路徑。

c. 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 100，代表 cycle time 為 100ns。

(100ns 是很寬鬆的 constraint，合成時盡量讓 cycle time 小，代表合出來的電路比較快。)

d. syn/run.tcl: **在 syn 資料夾內**執行 dc_shell –f run.tcl 即可完成 synthesis，合成成功後會產生 imgproc_syn.v 即為電路 gate level netlist。(這份是基本版的 script，如果想要更進一步優化電路，可以自行添加指令。Ex: 如 compile_ultra, compile –map_effort high, optimize_register…等)。

除此之外，如果電路有錯誤，請大家試著打開 design compiler 逐行執行 run.tcl 的指令，並去看合成過程中的 error message，很可能是 code 中寫了無法合成的東西、設定檔沒有就緒，或是產生了非預期的 latch，試著看錯誤訊息去排除問題。

C. Gate level simulation 部分所需檔案

產生 gate level netlist 後仍需要做模擬確定電路正確性。

a. testfixture.v: 同 RTL 用的 testbench，**要根據合成時的 cycle time 去修改裡面的參數，預設是 define CYCLE 100。**

b. img_proc_syn.v: 合成時產生的 gate level netlist。

c. tsmc13_neg.v: 各個 gate 製程的相關資訊。

d. syn/img_proc.sdf: 合成時產生的 standard delay fomat 檔案，描述了電路中每個元件的延遲時間。

e. 模擬時，直接執行：ncverilog testfixture.v imgproc_syn.v tsmc13_neg.v +define+SDF

若要記錄 debug 用模擬波形檔，請執行：

ncverilog testfixture.v imgproc_syn.v tsmc13_neg.v +access+r +define+SDF+FSDB

4. 評分標準

A. RTL 部分

Report 中請將 ncverilog 模擬畫面截圖（至少從 START 到 exit），若成功則會出現以下畫面。

```
START!!! Simulation Start .....

--------------------------------------------------

==================================================

 Congratulations!!! Every outputs are correct!

==================================================

Simulation complete via $finish(1) at time 96706900 PS + 0
./testfixture.v:116         $finish;
ncsim> exit
[r04030@thor hw4]$
```

若部分 pixel 出錯則會出現以下畫面，請大家利用 nWave 除錯。

```
      16364th Pixel is wrong :3c != 3b !

      16369th Pixel is wrong :2c != 2b !

      16372th Pixel is wrong :2d != 2c !

      16375th Pixel is wrong :3f != 3e !

      16378th Pixel is wrong :1c != 1b !

      16381th Pixel is wrong :23 != 22 !

The wrong pixels reached a total of  8161 or more !

Simulation complete via $finish(1) at time 96706900 PS + 0
./testfixture.v:116         $finish;
ncsim> exit
[r04030@thor hw4]$
```

若是出現以下畫面，代表你的模擬無法結束，可能是 finish 訊號線沒有處理好，同樣請大家利用 nWave 除錯。

```
--------------------------------------------------

START!!! Simulation Start .....

--------------------------------------------------

Time-out Error! Maybe there is something wrong with the 'finish' signal

Simulation complete via $finish(1) at time 590 US + 0
./testfixture.v:123         $finish;
ncsim> exit
[r04030@thor hw4]$
```

B. Gate level 部分

Report 中請大家放三張截圖：desing compiler 的 area report（包含 Total cell area）與 timing report（從 Point 到 slack (MET)），以及把 gate level netlist 再跑一次 ncverilog 模擬後的截圖（一樣從 START 到 exit）。

(記得 test bench 中的 cycle time 要改成與合成時的 cycle time 一樣再跑模擬。)

並請大家在截圖之後（report 最後面）填寫以下四項內容：（下為助教的範例）

Cell area = 5092 μm^2

Cycle time = 4 ns

Total time = 65564 ns

Cell area×total time = 333,851,888 ($\mu m^2 \cdot ns$)

```
Number of ports:                          122
Number of nets:                           465
Number of cells:                          339
Number of combinational cells:            271
Number of sequential cells:                66
Number of macros/black boxes:               0
Number of buf/inv:                         71
Number of references:                      60

Combinational area:               2866.908581
Buf/Inv area:                      762.132614
Noncombinational area:            2225.291344
Macro/Black Box area:                0.000000
Net Interconnect area:           43446.884033

Total cell area:                  5092.199925
Total area:                      48539.083958
```

```
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
data_reg[20]/CK (DFFRX1)                 0.00       0.50 r
data_reg[20]/QN (DFFRX1)                 0.99       1.49 r
U186/Y (AOI32X1)                         0.49       1.98 f
U325/Y (OAI21XL)                         0.56       2.54 r
U324/Y (AOI32X1)                         0.24       2.78 f
U180/Y (OAI22X2)                         0.31       3.09 r
U126/Y (NAND3X2)                         0.47       3.56 f
U155/Y (OAI221XL)                        0.53       4.09 r
result_reg[1]/D (DFFRX1)                 0.00       4.09 r
data arrival time                                   4.09

clock clk (rise edge)                    4.00       4.00
clock network delay (ideal)              0.50       4.50
clock uncertainty                       -0.10       4.40
result_reg[1]/CK (DFFRX1)                0.00       4.40 r
library setup time                      -0.30       4.10
data required time                                  4.10
-----------------------------------------------------------
data required time                                  4.10
data arrival time                                  -4.09
-----------------------------------------------------------
slack (MET)                                         0.00
```

```
ncsim> run
-----------------------------------------------

START!!! Simulation Start .....

-----------------------------------------------

================================================

 Congratulations!!! Every outputs are correct!

================================================

Simulation complete via $finish(1) at time 65564 NS + 0
./testfixture.v:118         $finish;
ncsim> exit
```

5. 繳交檔案

評分所需檔案可分為三部份：

(1) RTL design，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，以免無法進行編譯。

(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) Report file。

請將下表檔案先放在資料夾 hw2 內，再用 zip 進行壓縮。此 zip 檔解壓縮後必須是一個資料夾。

Design stage	File	Descirption
N/A	Report.pdf	design report
RTL Simulation	imgproc.v	Verilog synthesizable RTL code
Pre-layout Gate-level Simulation	imgproc_syn.v	Verilog gate-level netlist generated by Synopsys Design Compiler
	Imgproc_syn.sdf	SDF timing information generated by Synopsys Design Compiler
	Imgproc_syn.ddc	design database generated by Synopsys Design Compiler