

# Final project

## Triangle Rendering Engine

### 1. 問題描述

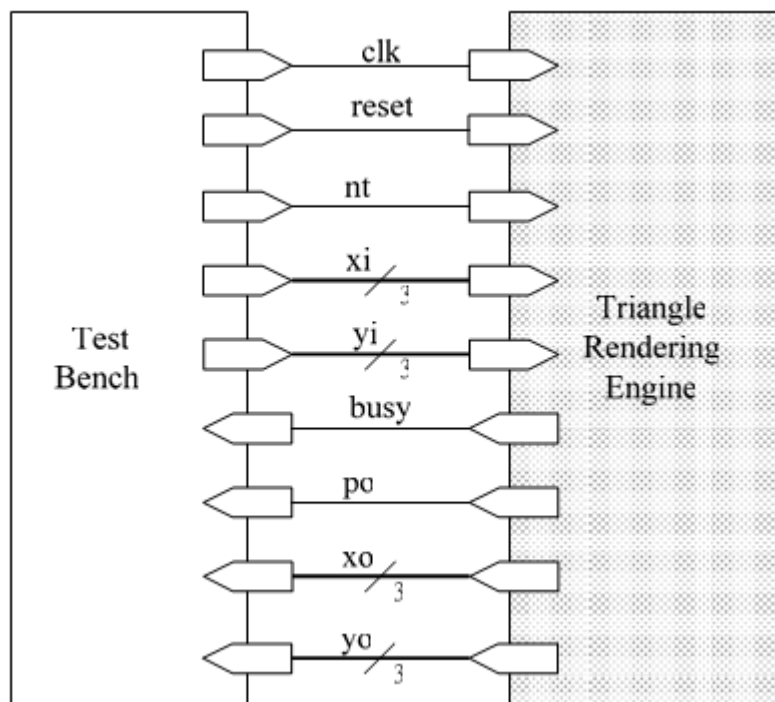
請完成一三角形座標轉譯系統(*Triangle Rendering Engine*)。此三角形之座標轉譯系統，將可於 testbench 所提供的三角形的三個頂點座標如 $(x1,y1)$ 、 $(x2,y2)$ 與 $(x3,y3)$ 後，轉譯系統將依序完成涵蓋於三角形平面內的所有座標點之輸出。

本次專案的作答時間為 12 月 27 日上午 09:00 到 1 月 3 日下午 9:00。當作答時間結束後，課程助教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，各組同學應參考附錄 E 中所列的要求，附上評分所需要的檔案。

本題目完成後請上傳至 “[140.118.121.247/Computer/File system/Project/Group\\_X](http://140.118.121.247/Computer/File system/Project/Group_X)” (The group list has already been announced in Moodle.)，詳情請參照附錄 E、F。

### 2. 設計規格

#### 2.1 系統方塊圖



圖一、系統方塊圖

## 2.2 輸入/輸出介面

表一、輸入/輸出訊號

Signal Name	Direction	Width(bit)	Description
reset	input	1	高位準非同步之系統重置信號。
clk	input	1	時脈信號。此系統為同步於時脈正緣之同步設計。
nt	input	1	三角形座標的輸入標記。當 nt 信號為高位準，表示有 3 組三角形的頂點座標輸入。(請注意!! 只有當 busy 信號為低位準時，nt 信號才能為高位準。)
xi	input	3	三角形頂點座標之 X 軸輸入值。
yi	input	3	三角形頂點座標之 Y 軸輸入值。
busy	output	1	當 busy 信號為高位準時，表示三角形座標轉譯系統正在運作，並防止新的(下一個)三角形頂點座標輸入於系統中。
po	output	1	po 為有效的資料輸出指示信號。當信號 po 為高位準時，於 xo 及 yo 輸出埠所輸出之資料均為有效之座標軸資料。
xo	output	3	三角平面內所涵蓋的座標點 X 軸之輸出結果
yo	output	3	三角平面內所涵蓋的座標點 Y 軸之輸出結果

## 2.3 系統功能描述

此設計之重置(reset)信號為非同步信號，當重置信號釋放(release)後，系統即開始依著時脈信號的正緣進行同步運作。系統之 nt 信號，當 nt 為高位準時(此時的 busy 信號應為低位準)，表示三角形的三個頂點 $(x1,y1)$ 、 $(x2,y2)$ 與 $(x3,y3)$ 座標，開始依序輸入於系統中。待 $(x3,y3)$ 頂點座標輸入前，busy 信號必須由低位準提昇至高位準，以防止下一個(新的)三角形的頂點座標輸入於系統。待產生輸出結果 $(xo, yo)$ 的同時，po 信號將由低位準提昇至高位準，以表示系統之輸出為有效之資料。課程助教將根據各組之 po 高位準信號，來判讀設計結果之正確性。此外，所有的測試輸出結果，將依下列範例之輸出順序(先由左而右，再由下而上)，依序進行輸出。

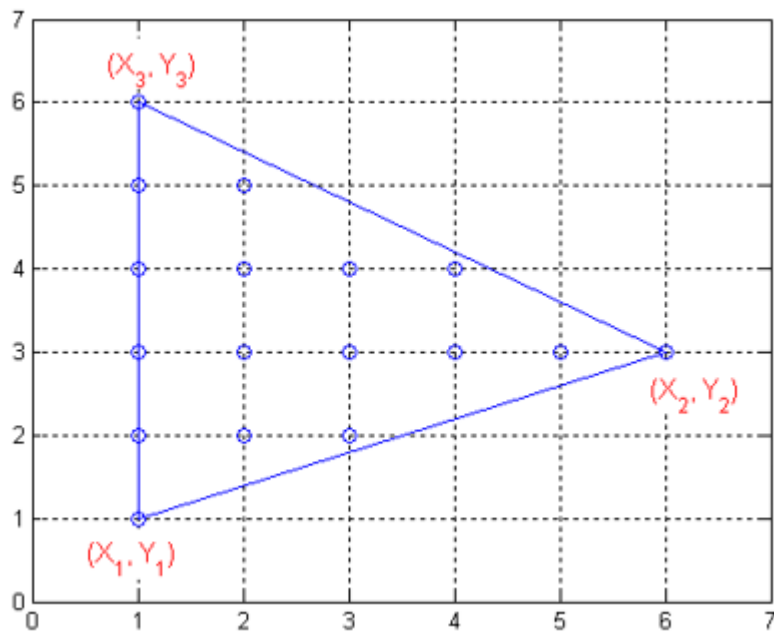
此三角形的三個頂點輸入為 $(x1,y1)$ 、 $(x2,y2)$ 與 $(x3,y3)$ ，並有下式之關係。隨所給定的三個頂點值，使三角形平面大小有不同之變化。

$$\begin{aligned}x1 &= x3 \\ y1 &< y2 < y3\end{aligned}$$

( $x1$ 、 $x2$ 、 $x3$ 、 $y1$ 、 $y2$ 、 $y3$  均為正整數) 亦即此三角形的其中一邊必定是垂直線。

圖二所示為三角形之座標範例。以下圖二為例，三角形的三個頂座標 $(x1,y1)$ 、 $(x2,y2)$  與 $(x3,y3)$  分別為 $(1,1)$ 、 $(6,3)$ 與 $(1,6)$ ，則此三角形之座標轉譯系統將先由左而右，再由下而上的依序輸出如下：

$(1,1), (1,2), (2,2), (3,2), (1,3), (2,3), (3,3), (4,3), (5,3), (6,3), (1,4), (2,4), (3,4), (4,4), (1,5), (2,5), (1,6)$ .



圖二、 三角形範例圖

所有的輸入及輸出資料均為 3 個位元，且均為 10 進制表示。

[設計提示]

座標系統中，倘若一條線段經過兩點 $(x_1, y_1)$ 與 $(x_2, y_2)$ ，我們可以得知，凡位於此線段上的座標點均滿足下式[1]。

$$\frac{x-x_1}{y-y_1} - \frac{x_2-x_1}{y_2-y_1} = 0 \quad [1]$$

同上所述，位於此線段的右邊座標點(非座落於線段上之點)，其關係滿足下式[2]。

$$\frac{x-x_1}{y-y_1} - \frac{x_2-x_1}{y_2-y_1} > 0 \quad [2]$$

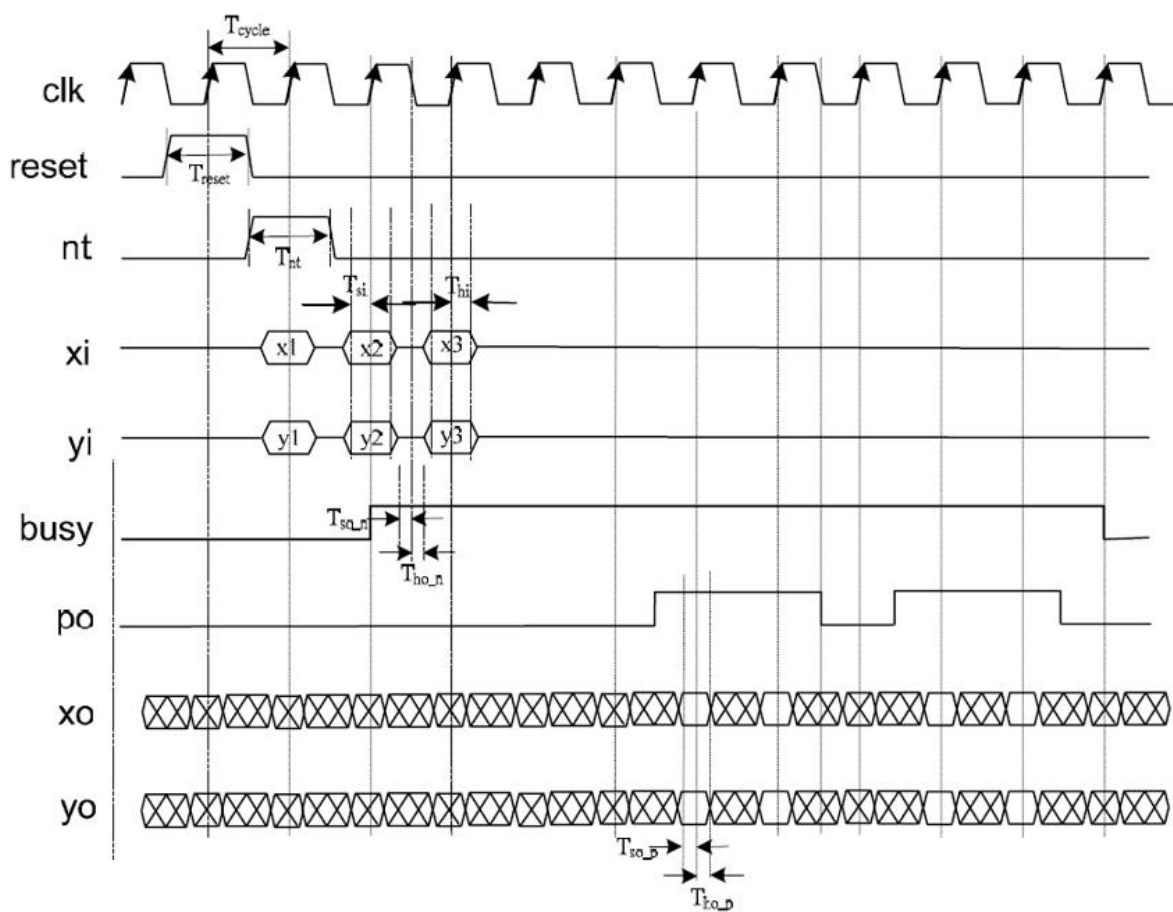
同理，位於此線段的左邊座標點(非座落於線段上之點)，其關係滿足下式[3]。

$$\frac{x-x_1}{y-y_1} - \frac{x_2-x_1}{y_2-y_1} < 0 \quad [3]$$

各組同學可以藉由上述式[1]~[3]，來鑑別位於三角形平面內所涵蓋的座標點。

## 2.4 時序規格圖

系統輸入/輸出時序規格圖，如圖三所示。



圖三、時序規格圖

表二、時序規格參數

Symbol	Description	Value
$T_{cycle}$	Clock (clk) period with duty cycle 50%	User defined
$T_{reset}$	Reset pulse width, active between negative edges of clk.	$= T_{cycle}$
$T_{nt}$	New triangle pulse width, active between negative edges of clk.	$= T_{cycle}$
$T_{si}$	Time period from valid signal to positive edge of clk.	$= 0.2 T_{cycle}$
$T_{hi}$	Time period from positive edge of clk to invalid signal.	$= 0.2 T_{cycle}$
$T_{so_p}$	Setup time from valid output to positive edge of clk.	$> 0.5ns$
$T_{ho_p}$	Hold time from positive edge of clk to invalid output.	$> 0.5ns$
$T_{so_n}$	Setup time from valid output to negative edge of clk.	$> 0.5ns$
$T_{ho_n}$	Hold time from negative edge of clk to invalid output.	$> 0.5ns$

### 3. 評分標準

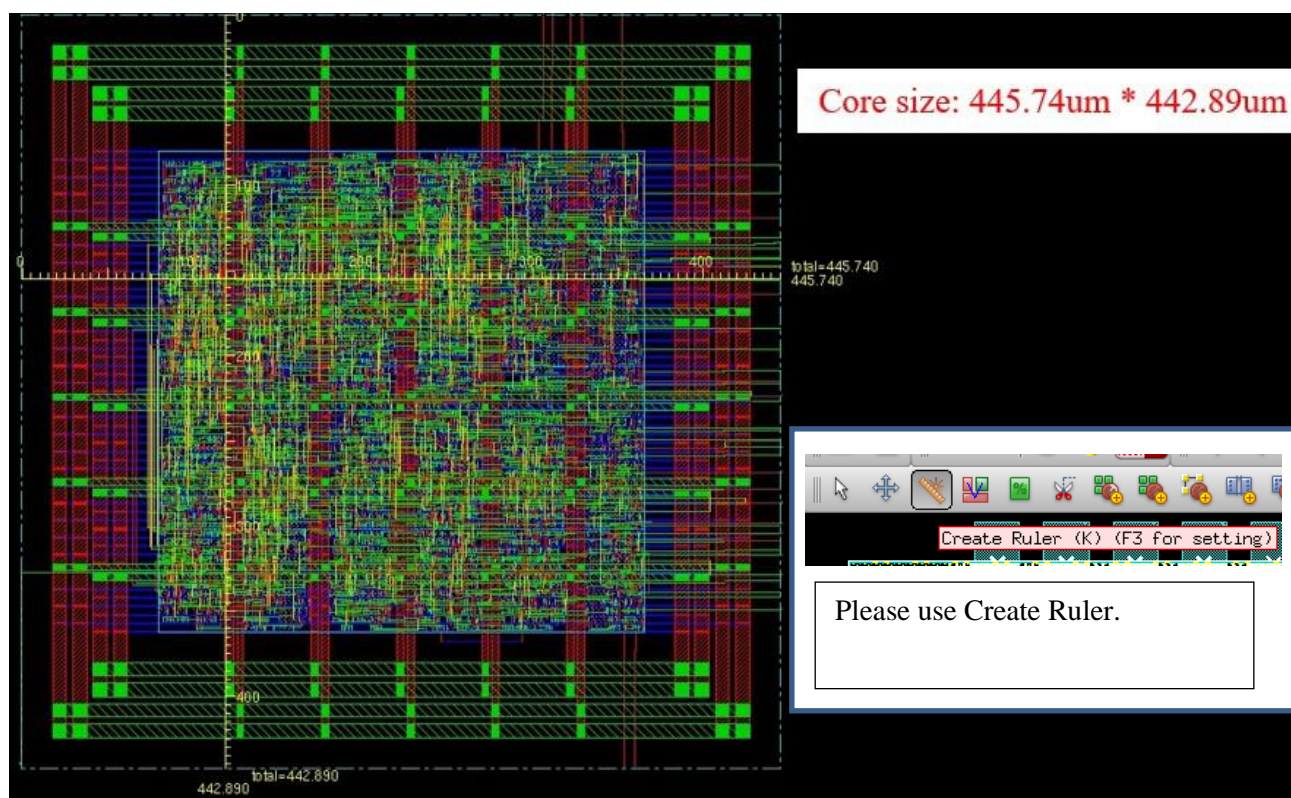
課程助教將依照各組同學提供之系統時脈進行 RTL simulation 與 gate-level simulation，以驗證設計正確性。各組同學應於本專案定義的系統時脈下，確保輸出結果無設置與保持時間(setup/hold time)的問題，並完全符合課程助教所提供的標準設計結果為準。

表三為本次專案的評分標準，一旦設計經課程助教驗證後，且同學的設計結果正確(post-layout)將可得到基本分80分，而其次課程助教將依據 Performance Metric 的表現，來給予而外的分數，而 Performance Metric 的計算方式如下所示，其數值越低表示設計的電路效能越好。

$$\bullet \text{ Performance Metric} = \text{Time Complexity} \times \text{Area Complexity}$$

**Time Complexity:** 當你完成 Post layout 的 gate level netlist 的 ncverilog 的驗證時，你可以從“ncverilog.log”的檔案中得到你的 Time Complexity 結果。這個時間代表整個電路完成工作所需的時間，與電路運作頻率(operating frequency)及電路效能(operations per cycle)有關。

**Area Complexity:** Area Complexity 可以從 SOC encounter 的 Layout圖透過尺規量得，如圖四所示。



圖四、Layout 尺規量測圖

## 附錄

附錄 A 提供了 2 組設計之測試樣本；附錄 B 為課程助教所提供各組同學的設計檔說明；附錄 C 為設計驗證說明，附錄 D 為評分用檔案，亦即各組同學必須繳交的檔案資料；附錄 E 為檔案上傳相關說明。

### 附錄 A、三角形座標轉譯系統之測試樣本(Testbench)

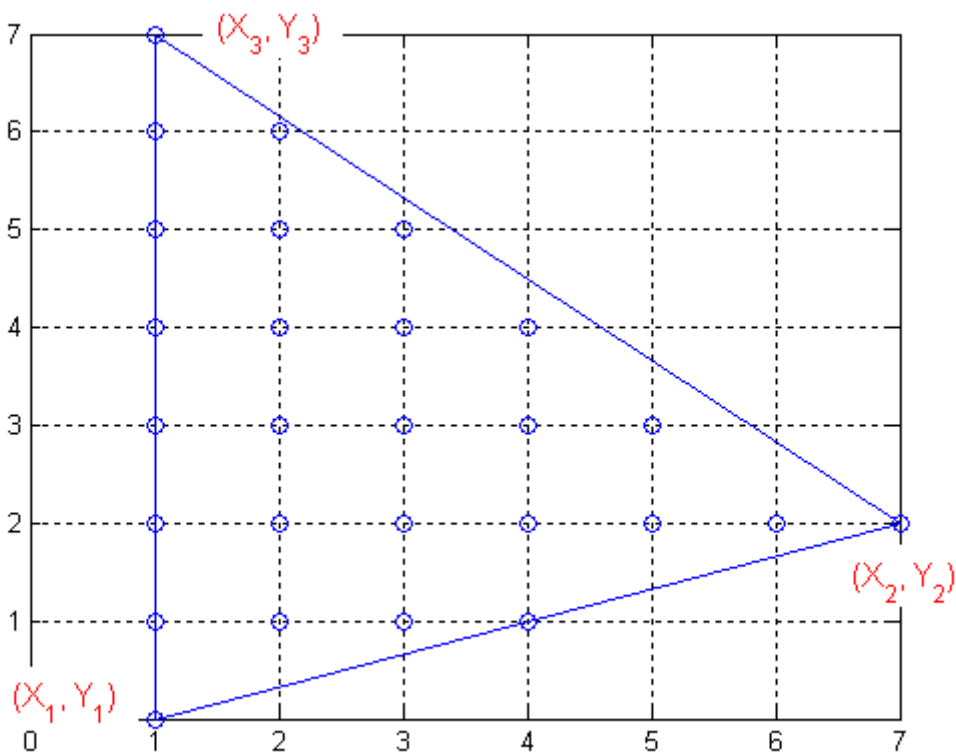
三角形樣本一：

三角形樣本一之頂點座標輸入：

$(1,0), (7,2), (1,7)$

三角形樣本一平面內所涵蓋的座標點依序輸出如下：

$(1,0), (1,1), (2,1), (3,1), (4,1), (1,2), (2,2), (3,2), (4,2), (5,2), (6,2), (7,2), (1,3), (2,3), (3,3), (4,3), (5,3), (1,4), (2,4), (3,4), (4,4), (1,5), (2,5), (3,5), (1,6), (2,6), (1,7)$



圖五、測試樣本一之三角形

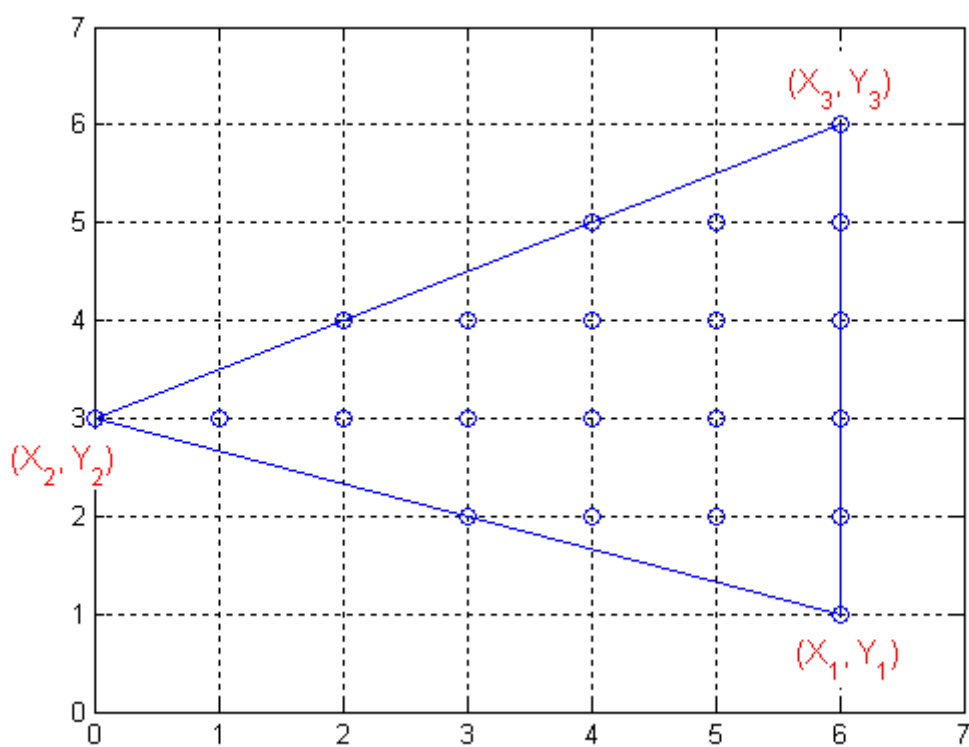
### 三角形樣本二：

三角形樣本二之頂點座標輸入：

$(6,1), (0,3), (6,6)$

三角形樣本二平面內所涵蓋的座標點依序輸出如下：

$(6,1), (3,2), (4,2), (5,2), (6,2), (0,3), (1,3), (2,3), (3,3), (4,3), (5,3), (6,3), (2,4),$   
 $(3,4), (4,4), (5,4), (6,4), (4,5), (5,5), (6,5), (6,6)$



圖六、測試樣本二之三角形



## 附錄 B、設計檔(For Verilog or VHDL)

1. 以下為課程助教所提供各參賽者的設計檔

表三、設計檔

檔名	說明
testfixture.v (.vhd)	Test bench includes the definition of clock period and the file names of test patterns.
triangle.v (.vhd)	Your design of the Triangle Rendering Engine should be here. The input and output ports are declared in this file.
input.dat	The input coordinates (xi, yi) of the triangle are listed in this file.
expect.dat	The correct results for test patterns.
Encounter_example	The example for SOC Encounter(Include mmmc, LEF, instruction for making GDS.....)
synopsys_dc.setup	Setup file for design compiler
TRE.tcl	The script for Design Compiler constraint

2. 請務必使用 ***triangle.v(.vhd)***，進行三角形座標轉譯系統之設計。其模組名稱、輸出/入埠宣告如下所示：

```
module triangle (clk, reset, nt, xi, yi, busy, po, xo, yo);  
  
input          clk, reset, nt;  
input  [2:0]   xi, yi;  
output         busy, po;  
output  [2:0]  xo, yo;  
endmodule
```

3. 請參賽隊伍使用課程助教所提供的測試樣本(testfixture.v)，來進行設計之模擬驗證。
- A. 測試樣本(testfixture.v) 包含 clk、reset、nt、xi 與 yi 測試信號。而於測試樣本內所定義的時脈週期參數(CYCLE)，可由參賽者依需求自行修改。
- B. 所有測試樣本均須通過驗證，如果結果正確測試樣本將會顯示 PASS 的字樣，否則則將顯示錯誤的時間和數值。

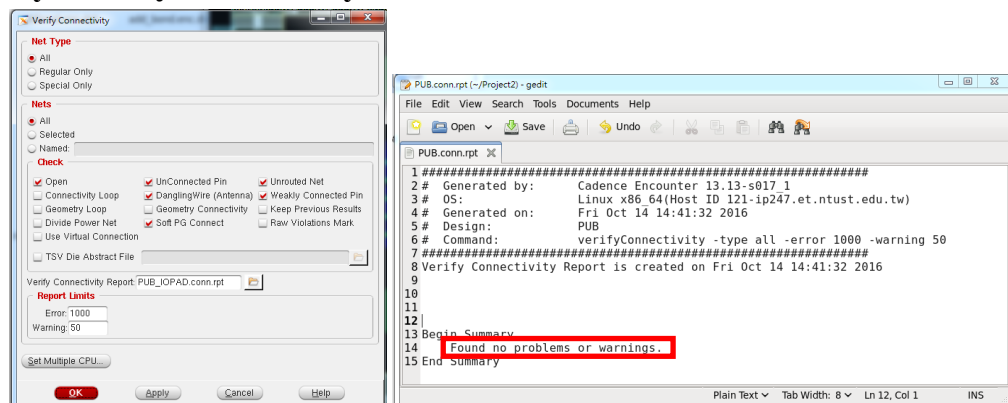
## 附錄 C 設計驗證說明

各組同學繳交資料前應完成 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：同學們必須進行 RTL simulation 及 Gate-Level simulation，模擬結果必須於參賽者自行定義的系統時脈下，**輸出結果正確且無 setup/hold time 的問題。**
- Physical 階段，包含三項驗證重點：
  1. 完成最後 layout
    - i. Marco layout，不含 IO Pad。
    - ii. VCC 與 GND power ring 請各設定為 2um。
  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**
  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**

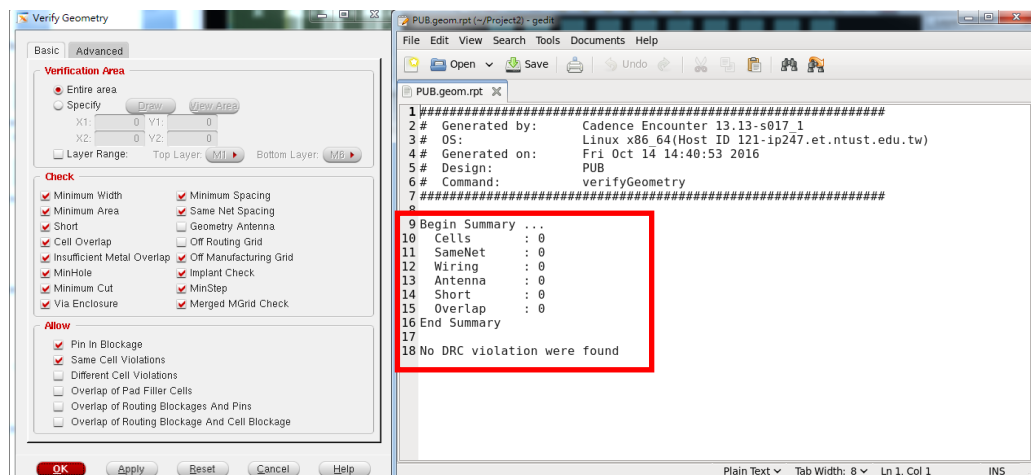
i 驗證 DRC 與 LVS 步驟如下：在 SOC Encounter 視窗下點選

“Verify → Verify Connectivity...” Default 值，按 **OK**。



在此步驟完成後會產生 \*.conn.rpt 檔，如驗證結果正確，會顯示如右圖紅框處之文字。

“Verify → Verify Geometry...” Default 值，按 **OK**。



在此步驟完成後會產生 \*.geom.rpt 檔，如驗證結果正確，會顯示如右圖紅框處之文字。

## 附錄 D 評分用檔案

評分所須檔案可以分為以下幾個部份：(1)RTL design，即各組同學對該次專案設計的 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，以免助教進行評分時，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，即由 **Cadence SOC Encounter** 所產生的 GDS Layout 檔，以及 DRC/LVS report 檔等。

表四

<i><b>RTL category</b></i>		
<i>Design Stage</i>	<i>File</i>	<i>Description</i>
N/A	*.doc	Design Report Form
RTL Simulation	*.v or *.vhd	Verilog (or VHDL) synthesizable RTL code
<i><b>Gate-Level category</b></i>		
<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
	*_syn.vcd	Wave information generated by Ncverilog
	ncverilog.log	The log file generated by Ncverilog
<i><b>Physical category</b></i>		
<i>Design Stage</i>	<i>File</i>	<i>Description</i>
P&R	*.gds	GDSII layout
	*.conn.rpt *.geom.rpt	DRC/LVS report generated by Cadence SOC Encounter
Post-layout Gate-level Simulation	*_pr.v	Verilog gate-level netlist generated by Cadence SOC Encounter
	*_pr.sdf	Post-layout gate-level sdf
	*_pr.vcd	Wave information generated by Ncverilog
	ncverilog.log	The log file generated by Ncverilog

## 附錄 E 檔案上傳

所有包含於如附錄 D 中表格所示的檔案，均需要提交至課程助教指定的資料夾內。步驟如下：

1. 參考附錄 D 評分用檔案，將除了 Design Report Form 以外的檔案繳交檔案複製到 (140.118.121.247/Computer/File System/Project/Group\_X) 資料夾中，其中 X 表示組別，請務必確保檔案放入正確的資料夾內。
2. Design Report Form 的 WORD 檔，請於填寫完後上傳到 Moodle，檔名請依各組組別命名(ex: Group\_1)。

若有任何問題，請聯絡課程助教：

陳偉哲 s2017091@gmail.com ; 辦公室: IB714-1

曾昱豪 M10402146@mail.ntust.edu.tw 辦公室: IB714-1