

# 2025 University/College IC Design Contest

## Cell-Based IC Design Category for Graduate Level

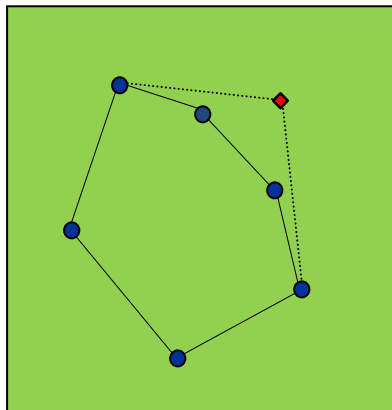
### *Convex hull*

#### 1.問題描述

Convex hull，中文翻譯凸包，指的是在二維平面上給定一組散落的點，找到一個最小的凸多邊形，該多邊形能夠包含所有的點。

請實現一個 Convex hull 系統（以下簡稱 **CONVEX**）。每當系統輸入一個新點後，會將該點加入並計算出涵蓋所有點的最小凸多邊形。

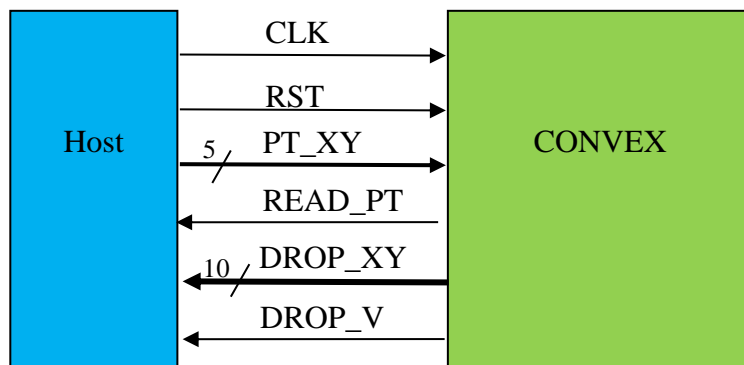
本次 IC 設計競賽比賽時間為上午 08:30 到下午 20:30。當 IC 設計競賽結束後，會根據第三章節中的評分標準進行評分。為了評分作業的方便，各參賽隊伍應參考附錄三中所列的要求，附上評分所需要的檔案。



圖一、CONVEX 系統示意圖

#### 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

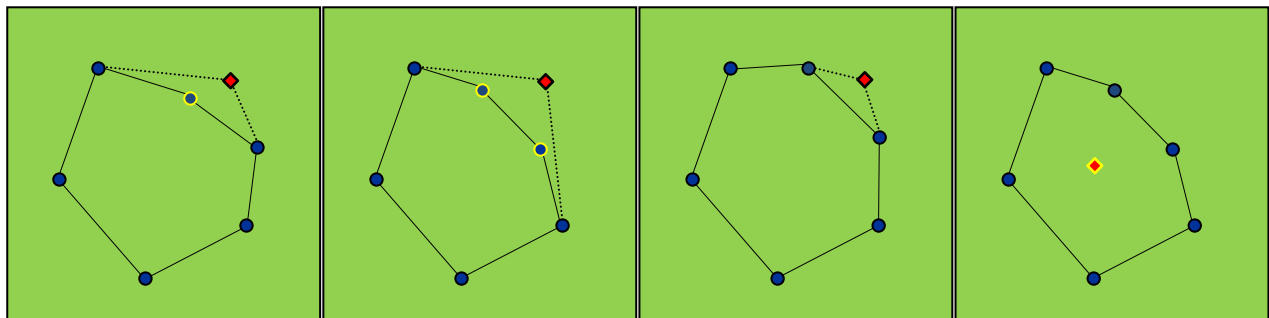
表 1 -輸入/輸出訊號

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。
RST	I	1	高位準非同步(active high asynchronous)系統重置信號。
PT_XY	I	5	新點之輸入埠。分 4 個 cycle 輸入新點的 X 及 Y 座標
READ_PT	O	1	CONVEX 要求輸入新點
DROP_XY	O	10	拋棄點座標，X 座標及 Y 座標皆由此埠輸入
DROP_V	O	1	有效拋棄點。當 DROP_V 為 High，表示目前輸出的 DROP_XY 為有效的輸出。

## 2.3 系統描述

### 2.3.1 系統功能

每當系統 Host 輸入一個新點，CONVEX 會根據此點更新資料，並計算出新的最小凸多邊形。隨後，CONVEX 僅保留最小凸多邊形的頂點，將其餘的點拋棄並輸出這些被拋棄的點的座標。根據不同情況，被拋棄的點數量會有所不同，可能是單個點、多個點，或甚至沒有任點被拋棄。若新點剛好是原多邊形的頂點，則直接拋棄並且輸出。

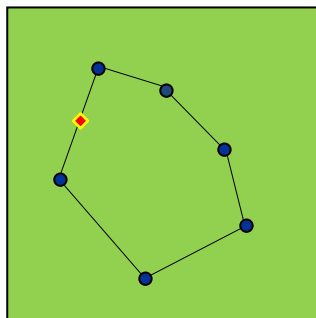


輸出(拋棄)一點

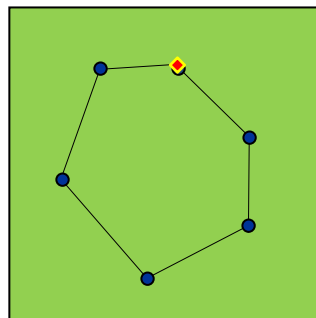
輸出(拋棄)多點

無輸出點

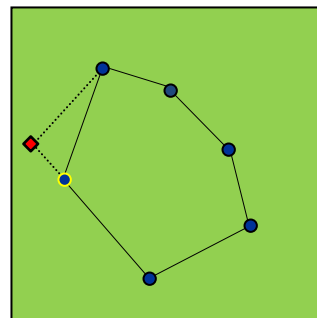
新加入點被拋棄



新加入點共線被拋棄



新加入點共點被拋棄

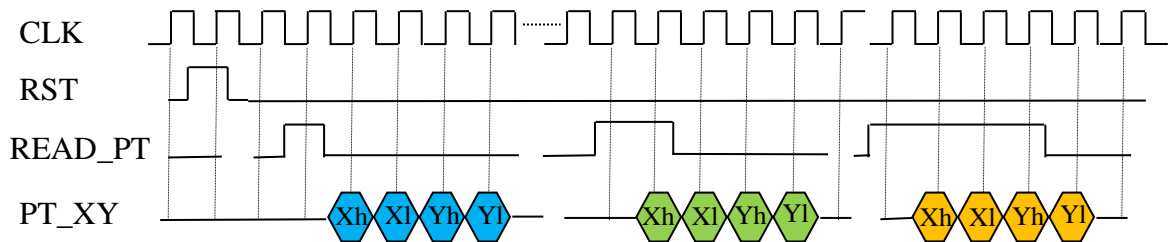


原多邊形點共線被拋棄

圖三、操作範例。藍點:原多邊形頂點，菱形紅點:新加點，黃框點:拋棄點

### 2.3.2 資料輸入

1. 系統 reset 後 CONVEX 即可拉高 READ\_PT 向 Host 要求輸入新點。
2. Host 接收到 READ\_PT 要求後，接著 4 個 cycle 會從 PT\_XY 依序送出 Xh、Xl、Yh、Yl；Xh 指的是新點 X 座標的前 5bit (X[9:5])，Xl 指的是新點 X 座標的後 5bit (X[4:0])，Yh 指的是新點 Y 座標的前 5bit (Y[9:5])，Yl 指的是新點 Y 座標的後 5bit (Y[4:0])
3. READ\_PT 拉高後的 3 個 cycle，Host 會暫停接受 CONVEX 讀取新點的要求，如圖四的第二組輸入，雖然 READ\_PT 連續拉高 2 cycle，但只會得到一組資料；第三組輸入，雖然 READ\_PT 連續拉高 4 cycle，也只會得到一組資料。

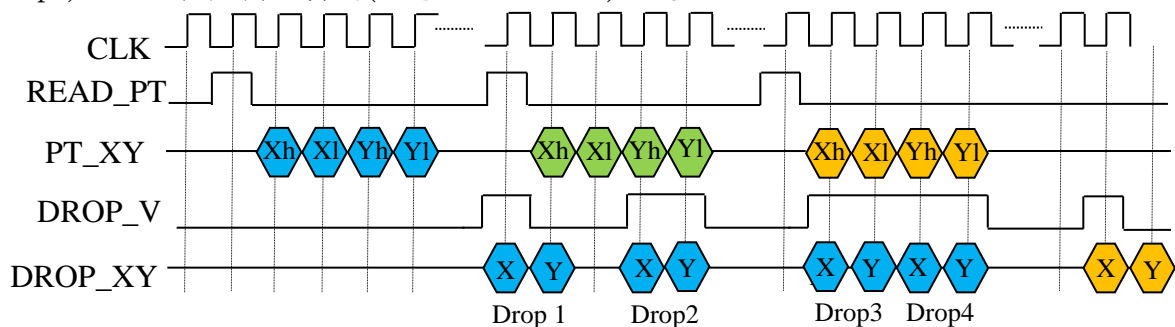


圖四、新點資料輸入

4. 本題座標範圍：X 範圍 0 到 1023，Y 範圍 0 到 1023

### 2.3.3 結果輸出

1. CONVEX 計算完成後，若有拋棄點產生，須將其座標從 DROP\_XY 輸出。輸出的順序為連續兩個 cycle，先輸出座標 X 再輸出座標 Y。在第一個 cycle(輸出座標 X)，必須將 DROP\_V 設為高電位，以表示這兩個 cycle 為有效輸出。而第二個 cycle(輸出座標 Y)，不論 DROP\_V 是高電位或是低電位，都是座標 Y 輸出週期。請見圖五，在 DROP\_XY 上的第一個輸出(Drop1)，雖然在輸出 Y 座標時 DROP\_V 是低電位，仍視為有效座標 Y 輸出。所以圖五的這幾個輸出點，DROP\_V 波形都是正確波形。
2. 一個新點可能產生超過一個拋棄點，這幾個拋棄點沒有輸出順序要求。如下圖藍色點產生 4 個拋棄點(Drop1、Drop2、Drop3、Drop4)，這四點可以亂序輸出，可以連續或分散輸出，但每個點只能輸出一次。
3. 不同點造成的拋棄點必須依序輸出，如下圖橘色拋棄點必須在藍色拋棄點之後。
4. 沒有產生拋棄點的新點，如綠色點，不會有輸出動作。
5. 第 n 點的拋棄輸出必定在該點的輸入之後，如下圖藍色點輸出(Drop1、Drop2、Drop3、Drop4)，必須在其輸入資料(藍色 Xh Xl Yh Yl)之後。

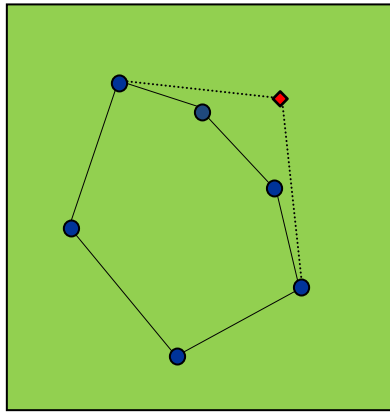


圖五、結果輸入

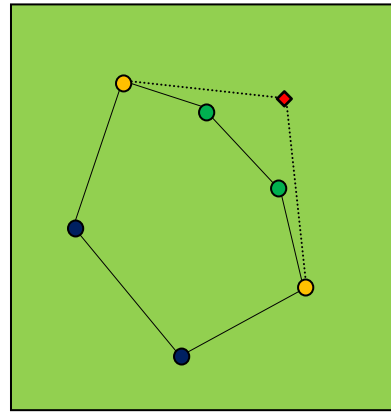
6. 輸出點的時間不受限輸入週期，如圖五藍色點輸出，雖然在綠色及橘色輸入之後，仍然是有效輸出。
7. 每組測試 pattern 有 50 輸入點，且最後一點必定有拋棄點輸出，Host 收到最後一點的輸出後會自動結束模擬。若在收到最後一點的輸出前有收到讀新點的要求，則會把最後一點重覆送出，但這不會影響到模擬結束時間。

### 2.3.4 凸多邊形計算

假設現有一凸多邊形，頂點已依順時鐘或逆時鐘排序好，如圖六藍色圓點。



圖六、凸多邊形計算

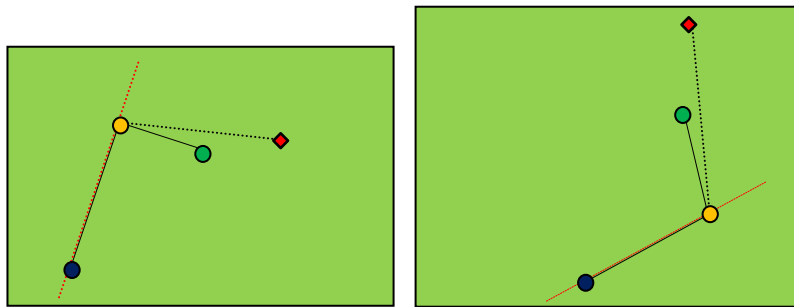


圖七、切點及內凹點

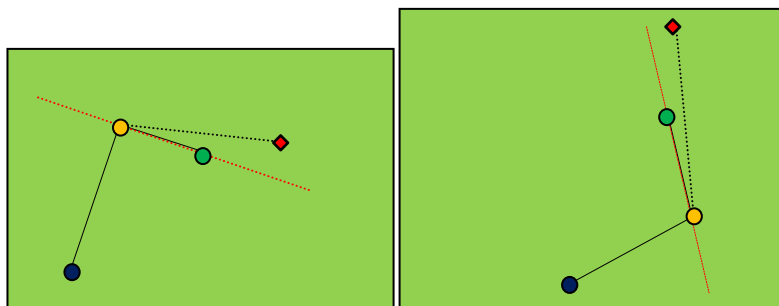
今新加入一點(菱形紅點)，要計算新的凸多邊形，可以用以下方法：

1. 找到新點和原多邊形的切點(圖七橘色圓點)，內凹點(圖七綠色圓點)。
2. 移除內凹點，並以新點替代。
3. 重新整理成排序狀態。

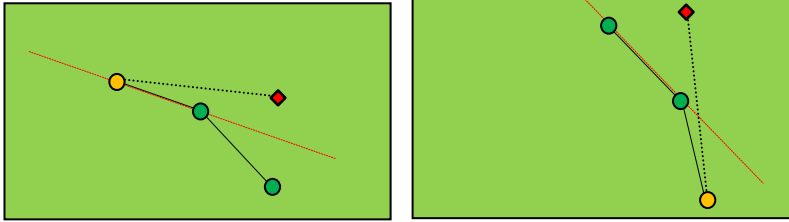
- 切點特徵：以切點和兩鄰點作直線，其中一直線會將新點和另一鄰點分在同側



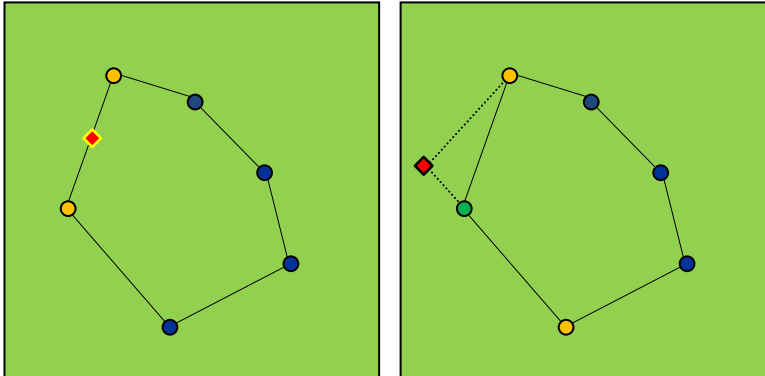
另一直線會將新點和另一鄰點分隔兩側



- 內凹點特徵：以內凹點和任一鄰點作直線，則新點和另一鄰點會在該直線的兩側



- 依以上特徵，可將要判斷的點代入直線方程式，然後可找到切點和內凹點。
- 當新點和另兩點共線(圖三兩共線狀況)，則代入直線方程式會等於 0，這兩狀況同樣可納入上面切點和內凹點判斷方法，請自行推敲如何擴充判斷。



- 若新點直接重疊原頂點，可直接判斷為拋棄點。
- 若完全找不到內凹點或切點，表示新點在原多邊形內。

#### 特別注意:

1. 本題 test pattern 前三點不會共線，必定會成為一三角形。
2. 本題 test pattern 形成的凸多邊形，最多不會超過 12 邊。
3. 本題 test pattern 每點產生的拋棄點，最多不會超過 6 個。
4. 題目內所提供建立凸多邊形的方法，非唯一方法，只要能完成功能，不限定一定要使用題目的方法。
5. 設計必須確實完成題目功能，**嚴禁針對題目 test pattern 特定內容做設計**，比如設計中判斷 pattern 為某固定數值，或是判斷第 n 個 pattern 直接設定輸出結果等。如經發現一律不予錄取。

### 3.評分標準

本題評分方法為：(三組 pattern 總模擬 cycle 數) x 合成面積

- 模擬結束會顯示模擬 cycle：

```
-----
-- Congratulation! Simulation completed
-- Total points:50 , failed points: 0
-- Total Simulation CYCLE #####
-----
```

- 合成面積：

Design compile report area 範例：

```
dc_shell> report_area
Combinational area:          41975.005127
Buf/Inv area:                2082.709759
Noncombinational area:      8700.872564
Macro/Black Box area:       0.000000
Net Interconnect area:      384301.939301

Total cell area:             50675.877691
Total area:                  434977.816992
```

本題評分使用 CBDK\_IC\_Contest\_v2.5 版本，**slow case** library，請勿使用錯誤的 cell library 或是 corner，以免造成無法評分。

本題設定 clock 週期時間為 8ns，參賽者不可調整 clock 週期時間。

評分方式依設計完成程度，分成 A、B、C、D、E 五種等級，排名順序為 A>B>C>D>E。各等級中，皆以 **Score 越小者** 名次越佳。

- ◇ 等級 A：

- 在 clock 週期為 8ns 環境下，可完成 Gate-Level 與 RTL 模擬，且結果正確。

等級 A 之評分方法：**Score = cycle x Area**

- ◇ 等級 B：

- 在 clock 週期為 8ns 環境下，模擬有錯。
- 調整 clock 週期超過 8ns，可完成 Gate-Level 與 RTL 模擬，且結果正確。

等級 B 之評分方法：**Score = (模擬正確之 clock period) x cycle x Area**

- ◇ 等級 C：

- RTL 模擬結果正確。
- 已完成合成，Gate-Level 可模擬但有部分錯誤。

等級 C 之評分方法：**Score = Gate-Level 模擬三組 pattern 總錯誤點數**

模擬結束會顯示錯誤點數：

```
-----
-- Simulation Failed
-- Failed at POINT 43
-- Total Failed points: 8
-----
```

- ◇ 等級 D：

- RTL 模擬結果正確。
- 無法合成或 Gate-Level 無法模擬

等級 D 之評分方法：**Score = RTL simulation cycle**

- ◇ 等級 E：

- RTL 模擬有部分錯誤。

等級 E 之評分方法：**Score = RTL 模擬三組 pattern 總錯誤點數**

## 附錄

附錄一為設計檔案說明；附錄二為測試樣本圖形；附錄三為評分用檔案，也就是完賽時須繳交的檔案資料；附錄四則為設計檔案壓縮整理步驟說明；

### 附錄一 設計檔案說明

#### 1 下表本題之設計檔案

表 2、設計檔案說明

檔名	說明
CONVEX.v	請使用本檔案進行設計，檔案內已包含 CONVEX 模組及輸出入埠宣告。
tb.sv	Test Bench 檔案。
convex_p*.dat	Test Pattern 資料
dc_syn.tcl	dc_shell 合成參考指令
.synopsys_dc.setup synopsys_dc.setup	使用 Design Compiler 做合成之初始化設定檔。參賽者請依 Library 實際擺放位置，自行修改 Search Path 的設定。注意：合成時請使用 worst case library。
CONVEX.sdc	合成之 constraint 檔案。
report.txt	report 檔格式，見附錄三。
dc_syn.tcl	dc 合成參考指令
xrun.cmd	xrun 模擬參考指令
vcs.cmd	vcs 模擬參考指令
vsim.cmd	vsim 模擬參考指令
pattern1.pdf，pattern2.pdf， pattern3.pdf	testpattern 各點輸入後的狀態圖，供除錯使用

#### 2 請使用 **CONVEX.v**，進行本題電路設計。其 Verilog 模組名稱、輸出/入埠宣告如下： 若有需要，參賽者可自行移除 output 的 reg 宣告。

```
module CONVEX (  
    input CLK,  
    input RST,  
    input [4:0] PT_XY,  
    output reg READ_PT,  
    output reg [9:0] DROP_XY,  
    output reg DROP_V);  
endmodule
```

3 本題所提供的 testbench 檔案，有幾行定義如下：

```
`define SDFFILE      "/CONVEX_syn.sdf"
`ifdef SDF
    initial $sdf_annotate (`SDFFILE, u_CONVEX);
`endif
```

3.1 SDF 檔之檔名，請自行依 SDF 實際檔名及路徑進行修改後再模擬。

3.2 在進行 gate-level 模擬時，需於模擬命令上多加 +define+SDF 參數才能順利模擬，範例如下

```
xrun -sv tb.sv CONVEX_syn.v -v tsmc13_neg.v +define+SDF
```

4 本題並有三組測試樣本，模擬時 default 使用測試樣本 1，若要模擬其它測試樣本，請使用 +define+P2 或 +define+P3 參數。範例如下

```
xrun -sv tb.sv CONVEX_syn.v -v tsmc13_neg.v +define+SDF+P2
```

```
xrun -sv tb.sv CONVEX_syn.v -v tsmc13_neg.v +define+SDF+P3
```

5 testbench 使用顏色讓模擬訊息較易閱讀，如果您的模擬環境無法顯示顏色造成排版錯亂，請將 tb.sv 內 USECOLOR 定義移除。如下

```
`timescale 1ns/10ps
`define CYCLE      8.0
`define SDFFILE    "/CONVEX_syn.sdf"
`define MAX_CYCLE  10000
//`define USECOLOR
```

6 MAX\_CYCLE 預設 1 萬個 cycles，目的是防止因設計電路有錯，模擬陷入無窮迴圈，參賽者可視需要請自行加大 MAX\_CYCLE 定義。

7 testbench(tb.sv)以 system verilog 格式撰寫，使用 xrun 模擬時請加入 -sv 參數；使用 vcs 模擬時請加入 -sverilog 參數。

8 RTL 及 Gate-level 模擬的指令說明如下

使用 P1 測試樣本之 RTL simulation 之指令：

➤ 使用 xrun 模擬指令(xrun.cmd)

```
xrun -sv tb.sv CONVEX.v +define+P1 +access+r -clean
```

➤ 使用 vcs 模擬指令(vcs.cmd)

```
vcs -R -sverilog tb.sv CONVEX.v +define+P1 +access+r +vcs+fsdbon +fsdb+mda
+fsdbfile+CONVEX.fsdb
```

➤ 使用 modelsim 模擬，請參考 vsim.cmd 內指令：

```
vsim -c -do vsim.cmd
```

vsim.cmd 內容：

```
vlib work
vlog tb.sv CONVEX.v +define+P1
vsim work.tb
run -All
```



使用 P1 測試樣本之 Gate-level simulation 之指令：

- 使用 xrun 模擬指令(**xrun.cmd**)

```
xrun -sv tb.sv CONVEX_syn.v -v tsmc13_neg.v +define+P1+SDF +access+r  
-clean -maxdelays
```

- 使用 vcs 模擬指令(**vcs.cmd**)

```
vcs -R -sverilog tb.sv CONVEX_syn.v -v tsmc13_neg.v +define+P1+SDF +access+r  
+vcs+fsdbon +fsdb+mda +fsdbfile+CONVEX.fsdb +maxdelays +neg_tchk
```

- 使用 modelsim 模擬，請參考 **vsim.cmd** 內指令：

```
vsim -c -do vsim.cmd
```

vsim.cmd 內容：

```
vlib work  
vlog tb.sv CONVEX_syn.v -v tsmc13_neg.v +define+P1+SDF  
vsim work.tb  
run -All
```

- 9 以 dc\_shell 進行 synthesis 參考指令下法說明如下

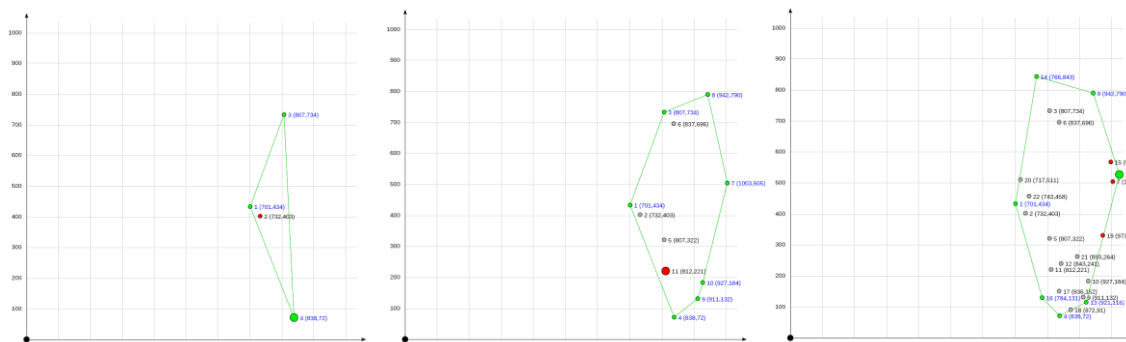
```
dc_shell -f dc_syn.tcl
```

合成結束會連同面積報告(area.log)寫出。

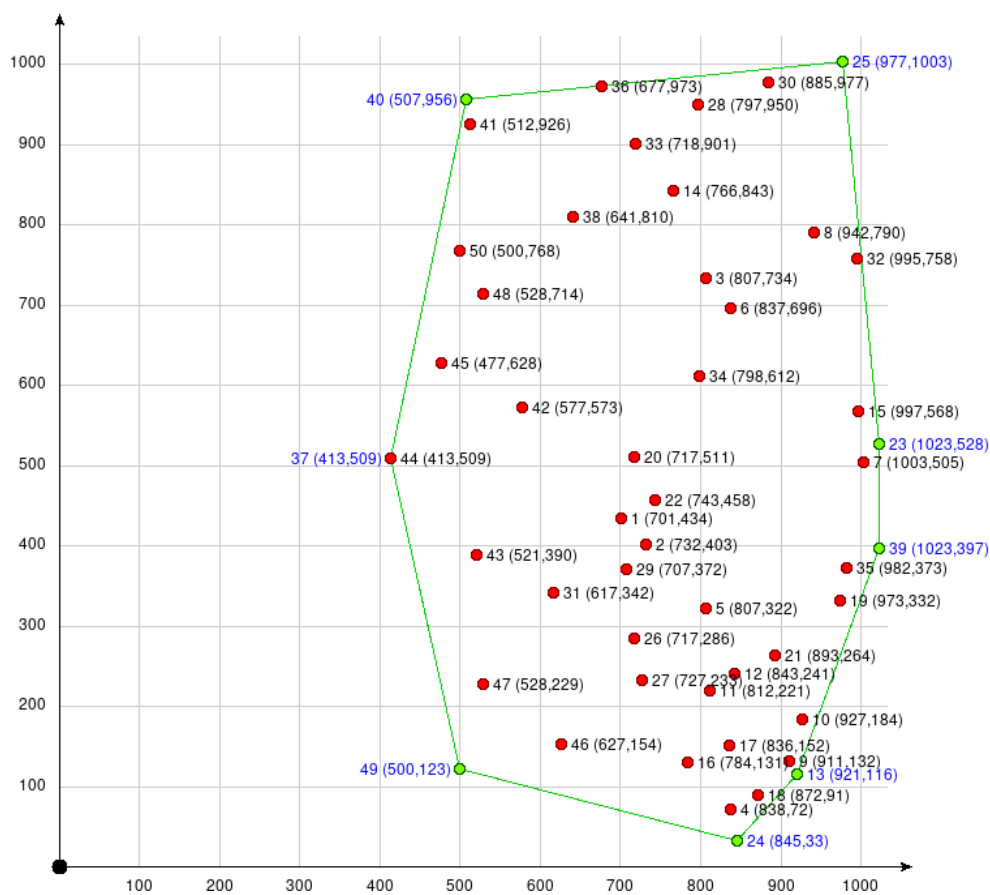
- 10 請儘可能直接在 **linux** 環境將設計檔解開，避免在 window 環境解壓縮後才 ftp 傳至 linux 環境，以免因 ftp 在兩系統間置換換行符號不正確造成 testbench 無法模擬的問題。

## 附錄二 測試樣本

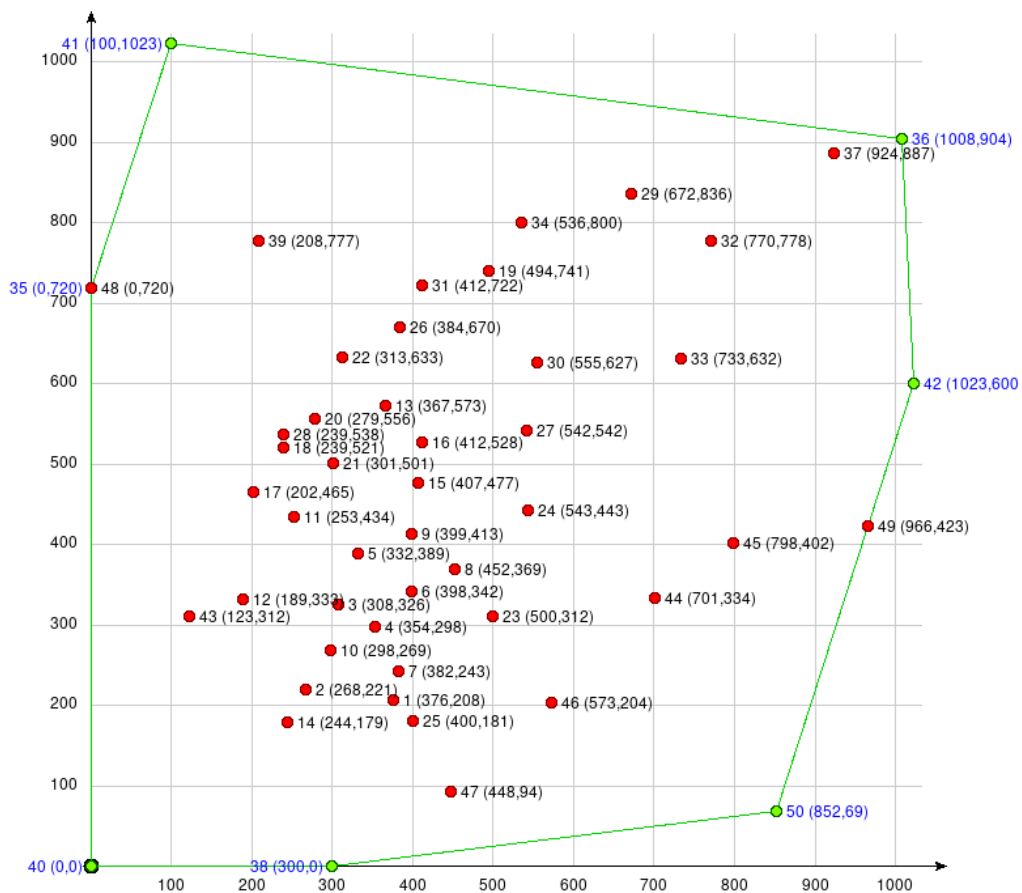
1. 測試檔案中提供 [pattern1.pdf](#) , [pattern2.pdf](#) 及 [pattern3.pdf](#) , 記錄所有點輸入後的狀態圖, 請參考檔案內圖像除錯。底下為部分截圖。



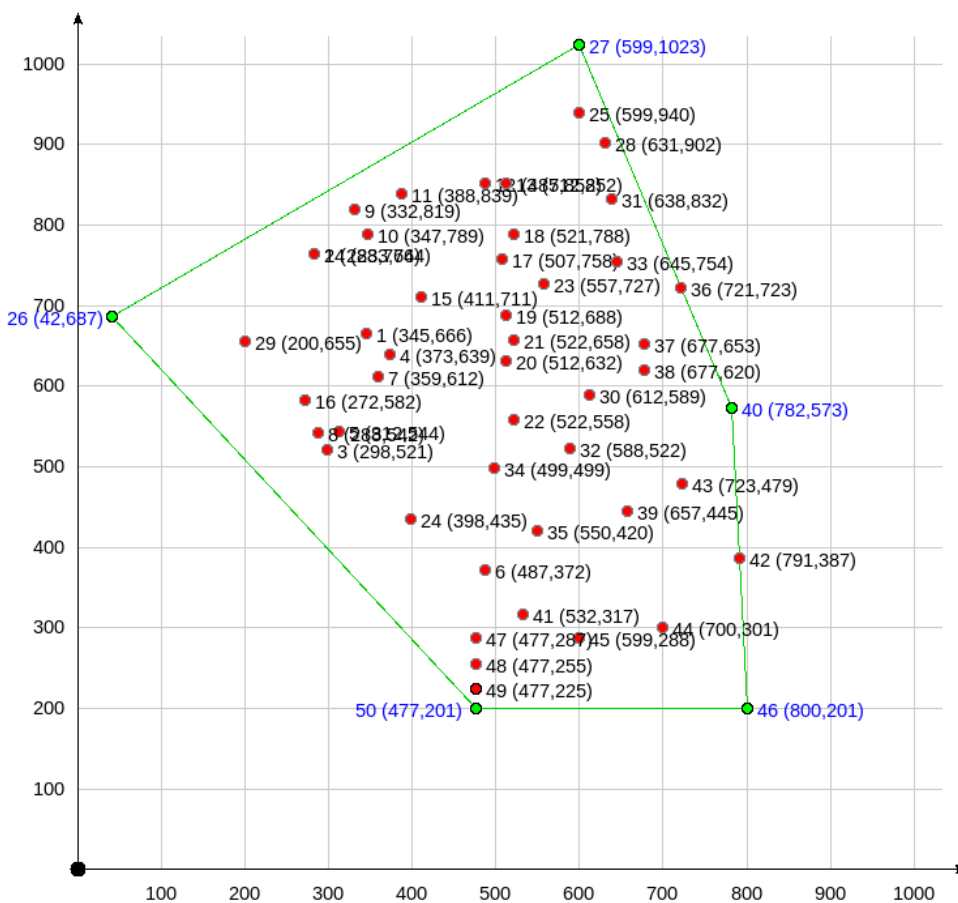
2. 以下提供三測試樣本的結束圖形, 依模擬順序, 輸入點編號由 1 到 50 號,



測試樣本一



測試樣本二



測試樣本三

### 附錄三 評分用檔案

評分所須檔案可以下幾個部份：

- (1) RTL design，各參賽隊伍對該次競賽設計的 RTL code，  
為方便評分，若設計採模組化而有多個檔案，繳交 RTL 時請將所有檔案合併至 CONVEX.v 中，只繳交一個 CONVEX.v 檔案。
- (2) Gate-Level design，由合成軟體所產生的 gate-level netlist，以及對應的 SDF 檔；
- (3) report file，參賽隊伍須依照自己的設計內容，撰寫 report.txt 檔，減少評分出錯機會。

表 3、繳交檔案

<b>RTL category</b>		
<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
<b>Gate-Level category</b>		
<i>Design Stage</i>	<i>File</i>	<i>Description</i>
Pre-layout	*_syn.v	Verilog gate-level netlist
Gate-level Simulation	*_syn.sdf	Pre-layout gate-level sdf

#### report.txt 檔

FTP account:	B25xxx, FTP 帳號
Level:	A/B/C/D/E 設計完成等級
Synthesis area :	120000，合成 report 的 cell area
Total cycle :	5000，模擬三個測試樣本的 cycle 總合
Total error point :	0，模擬三個測試樣本的錯誤點總合
--- RTL category---	
HDL simulator :	xrun/vcs/vsim，使用之 HDL 模擬器名稱
RTL filename :	CONVEX.v，RTL 檔案名稱
--- Pre-layout gate-level ---	
gate_level filename:	CONVEX_syn.v，gate-level 檔案名稱
gate-level sdf filename:	CONVEX_syn.sdf, sdf 檔案名稱
------(annotation)-----	
(其餘注意事項依各參賽隊伍的需求填寫)	

## 附錄四 檔案壓縮整理步驟

當表 3 所列文件準備齊全。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壓縮，步驟如下：

1. 建立一個 result\_xxx 資料夾。其中“xxx”表示繳交版本。例如 “000” 表示為第一次上傳；“001”表示為第二度上傳；002 表示為第三度上傳，以此類推…。

> **mkdir result\_000**

2. 將附錄三要求的檔案複製到 result\_xxx 這個目錄。例如：

> **cp CONVEX.v result\_000**

> **cp CONVEX\_syn.v result\_000**

> **cp CONVEX\_syn.sdf result\_000**

> **cp report.txt result\_000**

.....

為方便評分，請將 RTL 與 Gate-level 繳交檔案都放在 result\_000 目錄中，**不要在此目錄中另建其它層目錄。**

3. 執行 tar 指令將 result\_xxx 資料夾包裝起來，tar 的指令範例如下：

> **tar cvf result\_000.tar result\_000**

執行完後應該會得到 result\_000.tar 的檔案

若使用 windows 系統可用 zip 格式壓縮，**除 tar 及 zip 格式外，請勿使用其它壓縮格式。**

4. 使用 ftp 將 result\_xxx.tar 上傳至 TSRI 提供的 ftp server，評審將以最後上傳的設計檔編號進行評分作業。

**上傳之 FTP 需切換為二進制模式(binary mode)，且傳輸埠均設為 21(port:21)。**

ftp 的帳號和密碼在賽前已用 email 寄給各參賽者。若有任何問題，請聯絡主辦單位

FTP site1 (新竹半導體中心)：iccftp.tsri.org.tw (140.126.24.18)

EDA Cloud內： ICCFTP(192.168.67.101)

5. **若需要繳交更新版本，請重覆以上步驟，並記得修改 tar 檔的版本編號，因為您無法修改或刪除或覆蓋之前上傳的資料。**