

# Homework 5

## *Trapezoid Rendering Engine*

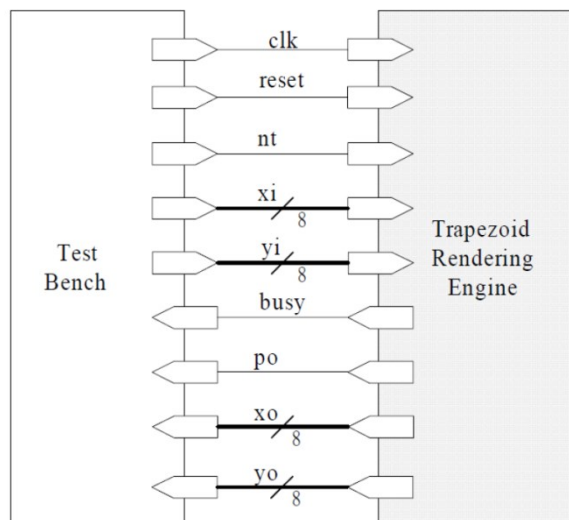
### 一、 作業要求

請設計一四邊形描繪產生器。此產生器在 testbench 依序輸入 4 組座標( $x_1, y_1$ ), ( $x_2, y_2$ ), ( $x_3, y_3$ ) and ( $x_4, y_4$ )作為四邊形的頂點後執行描繪的計算，接著把座標序列依序從  $x_o$  及  $y_o$  輸出。

設計的規格(如：系統方塊圖、輸入/輸出介面、功能描述、與時序規格)將於其後的章節說明。

### 二、 硬體規格描述

#### 1. 系統方塊圖



圖一、系統方塊圖

#### 2. 輸入/輸出介面

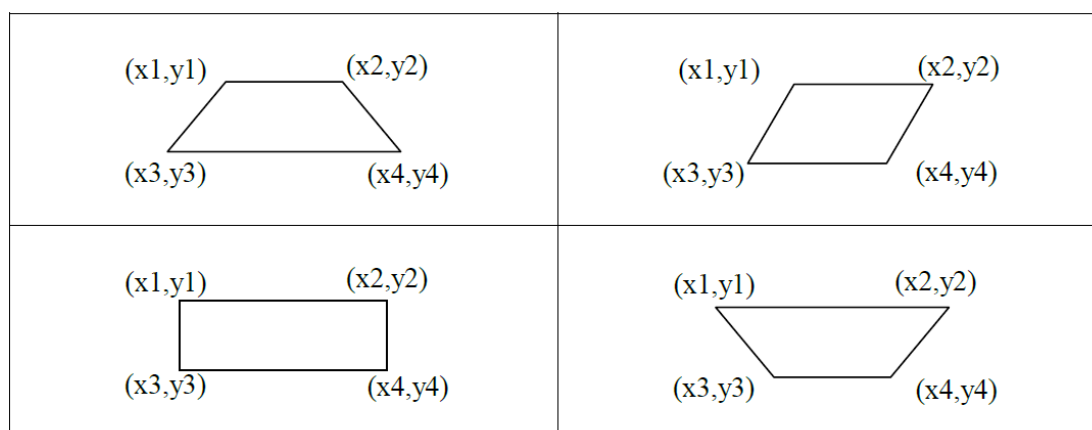
表一、輸入/輸出信號

訊號名稱	方向	位元數	描述
<i>reset</i>	input	1	Active-High 的 <b>非同步重置</b> 訊號。
<i>clk</i>	input	1	時脈訊號，此電路為 <i>clk</i> 正緣觸發的同步電路設計。
<i>nt</i>	input	1	控制此電路開始描繪一四邊形的訊號，當 <i>busy</i> 訊號為 low 時，若此訊號為 high，testbench 將 <b>依序</b> 輸入四個頂點座標作為四邊形的四個頂點，接著便開始描繪此四邊形。
<i>xi</i>	input	8	梯形頂點的 x 座標輸入埠，輸入值為 <b>二補數型態</b> 。
<i>yi</i>	input	8	梯形頂點的 y 座標輸入埠，輸入值為 <b>二補數型態</b> 。
<i>busy</i>	output	1	當此訊號為 high，代表電路正在執行繪圖的動作，此時不能再輸入新的頂點座標。
<i>po</i>	output	1	標記所輸出的(x,y)座標為有效輸出的信號，當此訊號為 high，表示 <i>xo</i> 和 <i>yo</i> 所輸出的(x,y)座標值是屬於此四邊形的有效作標。
<i>xo</i>	output	8	將此梯形的所有 x 座標輸出的埠，輸出值為 <b>二補數型態</b> 。
<i>yo</i>	output	8	將此梯形的所有 y 座標輸出的埠，輸出值為 <b>二補數型態</b> 。

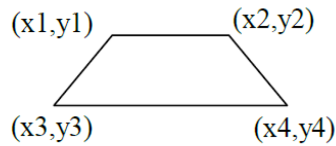
### 3. 功能描述

此電路在 *reset* 訊號為 high 時重置，且其為非同步重置(asynchronously reset)。當 *reset* 訊號變回 low 時，便開始等待 *nt* 訊號為 high。

*nt* 只有在 *busy* 訊號為 low 時才可能為 high，當 *nt* 為 high 時 testbench 將給予四邊形的四個頂點，依序為(x1,y1), (x2,y2), (x3,y3), (x4,y4)，輸入的四邊形可能如下。



接著，在最後一個頂點的座標(x4,y4)輸入前，trapezoid rendering engine 就得將 *busy* 訊號設為 high，直到處理完所有座標為止，以免還沒將此梯形的所有座標輸出，testbench 就試著要輸入下一個四邊形的四個頂點座標。在座標輸出的過程中，*po* 是用來指出 *xo* 和 *yo* 目前的輸出是否為有效值，也就是說當 *po* 為 high 時，testbench 才會將(*xo*,*yo*)的輸出和 EXPECT.DAT 裡的預期資料比較，若為 low，



testbench 將忽略此時 $(x0,y0)$ 的輸出。

$(x0, y0)$ 的有效輸出順序應與以下所列之參考用演算法一致。

```

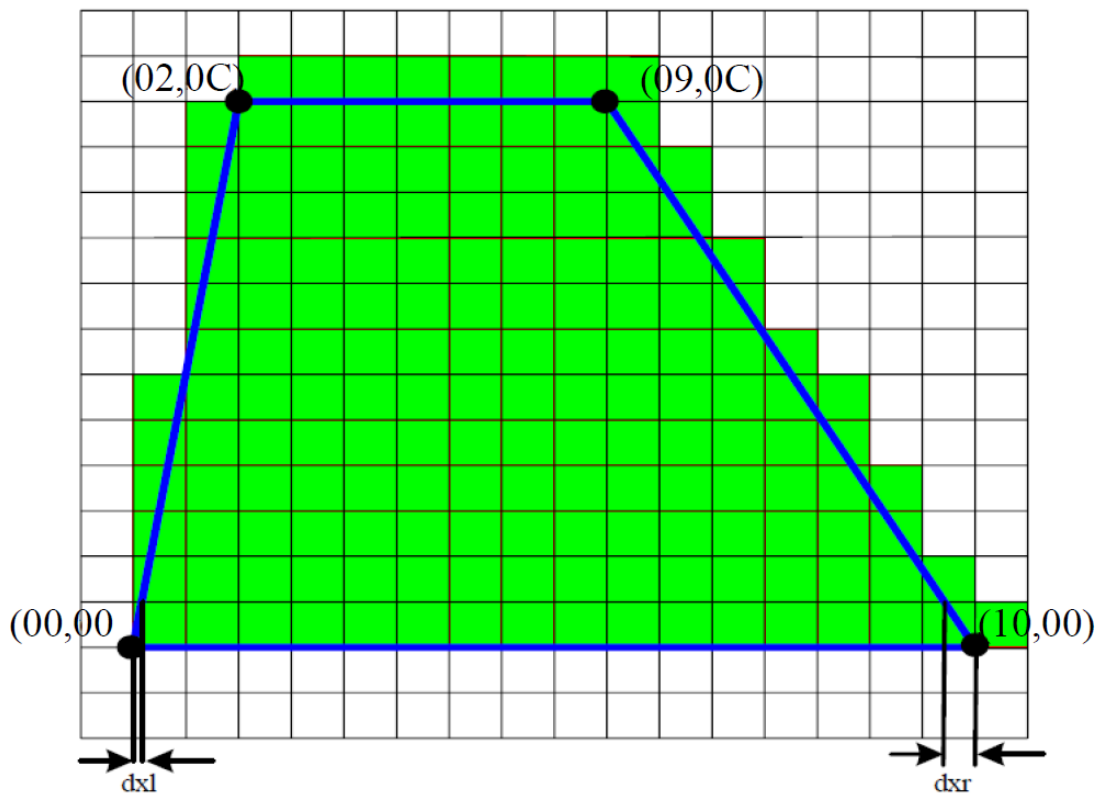
dxl = (x1 - x3) / (y1 - y3); //NOTE: y1 > y3, x4 > x3, x2 > x1
dxr = (x2 - x4) / (y1 - y3);
xl = x3; xr = x4;
for(i = y3; i <= y1; i++)
{
    if(xl ∈ Integer)    end_x = xr;
    else if (xr ∈ Integer) end_x = xr + 1;
    else                end_x = ⌈xl⌉; //NOTE: ⌈ ⌉ 為ceiling函數

    for(j = xl; j <= end_x; j++)
    {
        xo = ⌊j⌉; //NOTE: ⌊ ⌉ 為floor函數
        yo = i;
        output(xo, yo);
    }
    xl = xl + dxl;
    xr = xr + dxr;
}

```

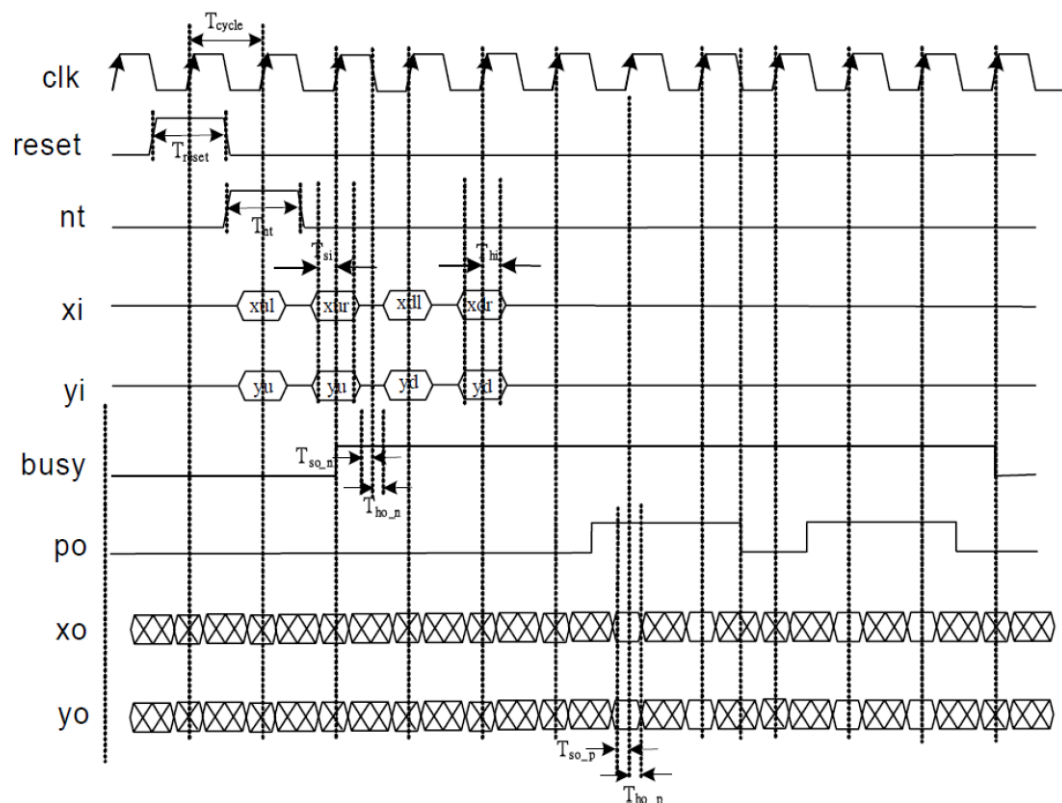
※  $xl, xr, end\_x, j, dxl$  and  $dxr$  are real numbers

下圖是一個四邊形描繪的例子，依序給予十六進制的二補數格式座標(02,0C), (09,0C), (00,00), (10,00) 作為四邊形的四個頂點，則此電路描繪出來的梯形就如下圖綠色部分所示。



當四個頂點如上圖所示，則 trapezoid rendering engine 的有效輸出依序為以下 162 個座標: (00,00), (01,00), (02,00), (03,00), (04,00), (05,00), (06,00), (07,00), (08,00), (09,00), (0A,00), (0B,00), (0C,00), (0D,00), (0E,00), (0F,00), (10,00), (00,01), (01,01), (02,01), (03,01), (04,01), (05,01), (06,01), (07,01), (08,01), (09,01), (0A,01), (0B,01), (0C,01), (0D,01), (0E,01), (0F,01), (00,02), (01,02), (02,02), (03,02), (04,02), (05,02), (06,02), (07,02), (08,02), (09,02), (0A,02), (0B,02), (0C,02), (0D,02), (0E,02), (00,03), (01,03), (02,03), (03,03), (04,03), (05,03), (06,03), (07,03), (08,03), (09,03), (0A,03), (0B,03), (0C,03), (0D,03), (0E,03), (00,04), (01,04), (02,04), (03,04), (04,04), (05,04), (06,04), (07,04), (08,04), (09,04), (0A,04), (0B,04), (0C,04), (0D,04), (00,05), (01,05), (02,05), (03,05), (04,05), (05,05), (06,05), (07,05), (08,05), (09,05), (0A,05), (0B,05), (0C,05), (0D,05), (01,06), (02,06), (03,06), (04,06), (05,06), (06,06), (07,06), (08,06), (09,06), (0A,06), (0B,06), (0C,06), (01,07), (02,07), (03,07), (04,07), (05,07), (06,07), (07,07), (08,07), (09,07), (0A,07), (0B,07), (01,08), (02,08), (03,08), (04,08), (05,08), (06,08), (07,08), (08,08), (09,08), (0A,08), (0B,08), (01,09), (02,09), (03,09), (04,09), (05,09), (06,09), (07,09), (08,09), (09,09), (0A,09), (01,0A), (02,0A), (03,0A), (04,0A), (05,0A), (06,0A), (07,0A), (08,0A), (09,0A), (0A,0A), (01,0B), (02,0B), (03,0B), (04,0B), (05,0B), (06,0B), (07,0B), (08,0B), (09,0B), (02,0C), (03,0C), (04,0C), (05,0C), (06,0C), (07,0C), (08,0C), (09,0C).

#### 4. 時序規格



Symbol	Description	Value
$T_{\text{cycle}}$	Clock (clk) period with duty cycle 50%	User define
$T_{\text{reset}}$	Reset pulse width, active between negative edges of clk.	$= T_{\text{cycle}}$
$T_{\text{nt}}$	New trapezoid pulse width, active between negative edges of clk.	$= T_{\text{cycle}}$
$T_{\text{si}}$	Time period from valid signal to positive edge of clk.	$= 0.2 T_{\text{cycle}}$
$T_{\text{hi}}$	Time period from positive edge of clk to invalid signal.	$= 0.2 T_{\text{cycle}}$
$T_{\text{so\_p}}$	Setup time from valid output to positive edge of clk.	$> 0.5\text{ns}$
$T_{\text{ho\_p}}$	Hold time from positive edge of clk to invalid output.	$> 0.5\text{ns}$
$T_{\text{so\_n}}$	Setup time from valid output to negative edge of clk.	$> 0.5\text{ns}$
$T_{\text{ho\_n}}$	Hold time from negative edge of clk to invalid output.	$> 0.5\text{ns}$

### 三、 注意事項

(1)完成設計的兩項要求：

- 完成上述演算法(Trapezoid Rendering Engine)的模組
- 必須通過作業所附的 **testbench**，完成 **Pre-sim**、**Post-sim**。

(2)此題為循序電路，請特別注意是否有 **latch**

(3)請附上Flow Summary (Processing -> Compilation Report)於報告中

(4)於 Post-sim 時，可調整 “testfixture.v”內之`define CYCLE 100，請將可測試通過的最小值記錄於報告

### 四、 作業繳交方式

(1)請將作業上傳到 **Moodle 作業區 HW5**，並且壓縮成以下格式：

HW5\_學號\_姓名.zip or HW5\_學號\_姓名.rar

附註：

第一版為HW5\_學號\_名字\_001.zip

若有更改則為HW5\_學號\_名字\_002.zip，以此類推

(2)請將繳交檔案分成三個資料夾，如下表分別為說明文件、Pre\_Sim 與Post\_Sim

目錄名稱	檔案名稱
Pre_sim	trapezoid.v
Post_sim	trapezoid.v, trapezoid_v.sdo

說明文件	HW5_學號_名字_版本.doc 附註： 請於報告內說明設計概念、技巧、使用面積、電路操作速度...
------	--

(3)為避免網路擁塞影響作業繳交，請盡早上傳作業

## 五、 評分方式

- (1)作業DEMO 的時間公佈在網頁。
- (2)評分比例：Pre-Sim (70%)、Post-Sim (30%)。
- (3)遲交或上傳檔案有病毒者一律以0 分計算。
- (4)抄襲他人作業者一律以0 分計算。
- (5) 工作時脈越高、面積越小者，分數越高。

## 六、 Q&A

有任何問題請 mail 給助教([p78001324@mail.ncku.edu.tw](mailto:p78001324@mail.ncku.edu.tw))