2009 University/College IC Design Contest

Cell-Based IC Design Category for Graduate Level

Color Transform Engine

1.問題描述

請完成一 Color Transform Engine(**後文以 CTE 表示**)的電路設計。如圖一，本 CTE 電路功能有二，(1) 將彩色訊號的每個像素(Pixel)之 YUV 訊號轉換成 RGB 訊號，(2)將彩色訊號的每個 Pixel 之 RGB 訊號轉成 YUV 訊號，其詳細規格將描述於後。

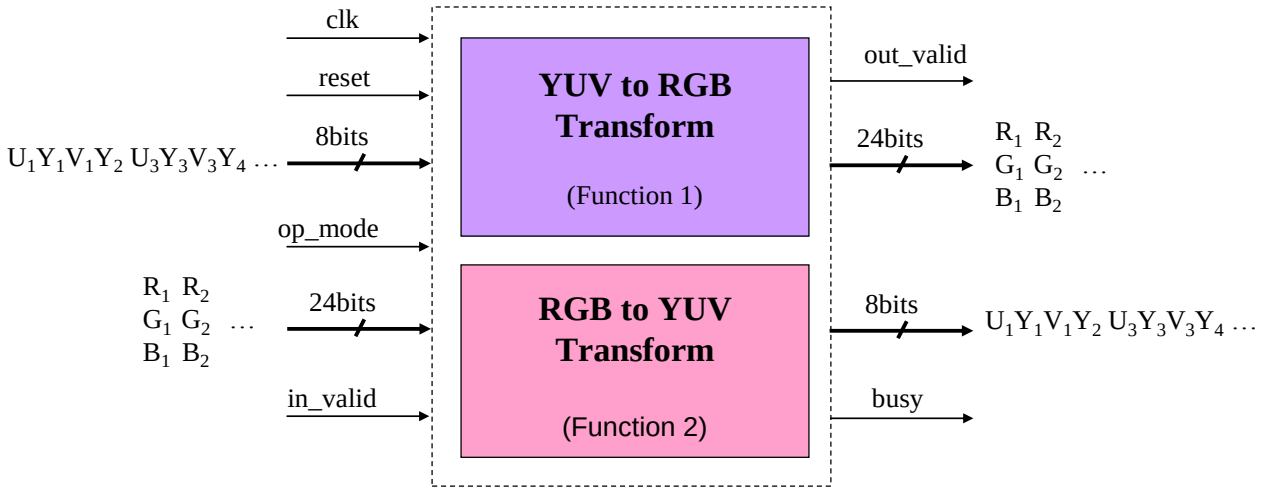
本電路各輸入輸出信號的功能說明，請參考表一。每個參賽隊伍必須根據下一節所給的設計規格及附錄 A 中的測試樣本完成設計驗證。

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，CIC 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，各參賽隊伍應參考附錄 E 中所列的要求，附上評分所需要的檔案。

本題目之測試樣本置於 ***/usr/cad/icc2009/cb/icc2009cb.tar*** ，請執行以下指令取得測試樣本：

tar xvf /usr/cad/icc2009/cb/icc2009cb.tar

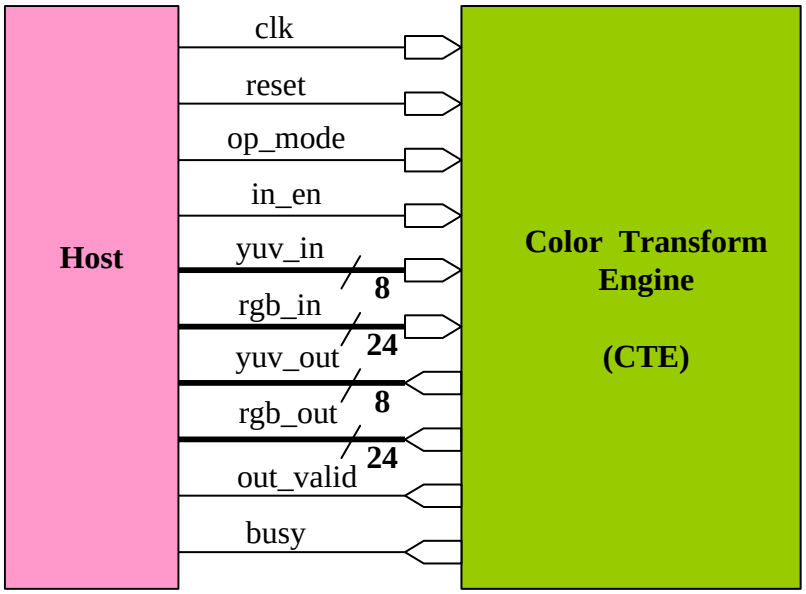
軟體環境及設計資料庫說明請參考附錄 F 與附錄 G。



圖一、彩色訊號轉換功能之方塊圖

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。 (註: Host 輸入訊號為 clk 負緣時送入資料。)
reset	I	1	高位準非同步(active high asynchronous)之系統重置信號。
op_mode	I	1	功能切換控制訊號。當為 Low 時，表示進行 YUV 訊號轉換成 RGB 訊號之功能。當為 High 時，表示進行 RGB 訊號轉換成 YUV 訊號之功能。
in_en	I	1	資料輸入致能控制訊號。當 Host 端有 YUV 或 RGB 訊號要輸入時，該訊號就會一直維持在 High，直到輸入訊號全部輸入完畢，該訊號才會為 Low。
busy	O	1	CTE 忙碌之控制訊號。當為 High 時，表示系統正處於忙碌階段，告知 Host 端，暫時停止 YUV 或 RGB 訊號的輸入；反之，當為 Low 時，表示告知 Host 端可繼續輸入 YUV 或 RGB 訊號。
yuv_in	I	8	YUV 三種訊號個別輸入的資料匯流排。YUV 輸入訊號都是 8bits，三訊號採個別輸入。只有當 **in_en** 為 High, **busy** 為 Low 時，輸入的資料才是有效的。

rgb_in	I	24	RGB 三種訊號合併輸入的資料匯流排。RGB 輸入訊號都是 8bits，三訊號共計 24bits 採合併輸入。只有當 **in_en** 為 High, **busy** 為 Low 時，輸入的資料才是有效的。
yuv_out	O	8	YUV 三種訊號個別輸出的資料匯流排。YUV 輸出訊號都是 8bits，三訊號採個別輸出。
rgb_out	O	24	RGB 三種訊號合併輸出的資料匯流排。RGB 輸出訊號都是 8bits，三訊號共計 24bits 採合併輸出。
out_valid	O	1	輸出資料有效之控制訊號。當為 High 時，表示目前 YUV 或 RGB 訊號為有效的輸出訊號;反之,當為 Low 時，表示目前輸出訊號為無效的，即不被採用。

2.3 系統描述

2.3.1 YUV 訊號與 RGB 訊號基本概念

彩色影像的每個 Pixel 是由 R(Red)、G(Green)、B(Blue) 三基色分量的強弱組合來決定一個 Pixel 的顏色，例如：RGB 三基色分量(R，G，B) => (0，0，0)（即都最弱）時，該 Pixel 會呈現黑色，當 RGB 三基色分量(R，G，B) => (255，255，255)（即都最強）時，該 Pixel 會呈現白色，因此調整 RGB 三基色分量的值，可以調出各式各樣的顏色。

基於不同的應用，彩色影像的另一種表示方法是由 YUV 模型表示，其中 Y 為明亮度訊號(Luminance)，U 為色調(Hue)，V 為飽和度(Saturation)。RGB 彩色模型與 YUV 彩色模型之間關係可以用矩陣(Matrix)型態描述，彼此之間可以互作轉換。YUV 模型特色為，一張影像各 Pixel 只需單獨的 Y 訊號分量即可決定出一張灰階影像，至於與顏色有關的 U、V 訊號，會依其分量的強弱來決定該影像之各 Pixel 的色彩。人眼對於彩色訊號之敏銳度較差，因此對於每個 Pixel 的彩色訊號常會使用次取樣(Down Sample)的機制，以節省記憶空間或減少資料的傳送量。

2.3.2 YUV 訊號轉換成 RGB 訊號功能描述

YUV 彩色模型轉換成 RGB 彩色模型，其矩陣表示式如(1)式。本 CTE 電路 Function1，如圖一所示，請完成將 YUV 訊號轉換成 RGB 訊號之功能。

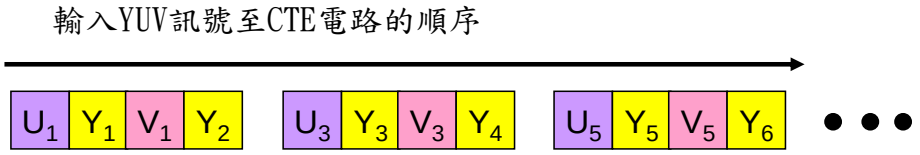
Function1 ⇨

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1.625 \\ 1 & -0.25 & -0.75 \\ 1 & 2 & 0 \end{bmatrix} \begin{bmatrix} Y \\ U \\ V \end{bmatrix} \qquad (1)$$

2.3.2.1 Function 1 之輸入端

YUV 都是 8 bits 的一維(1D)輸入訊號，Y、U、V 輸出訊號皆為 8bits，其中 Y 訊號輸入範圍為 0~255 的整數值，U 訊號輸入範圍為-117~+117 的整數值，V 訊號輸入範圍為-111~+111 的整數值。主辦單位所提供的 YUV 輸入訊號已事先針對 U、V 訊號作 Down Sample 2 之處理，因此 Y 訊號假設提供 N 筆資料量，則 U、V 訊號提供為各 N/2 筆資料量，Y、U、V 訊號是個別輸入的，其輸入順序採用 UYVY 格式，該格式輸入順序如圖三所示。

(註：所有負數值，都採用 2 的補數(**2's Complement**)來表示。)



圖三、YUV 訊號個別輸入之順序(UYVY 格式)

2.3.2.2 Function 1 之輸出端

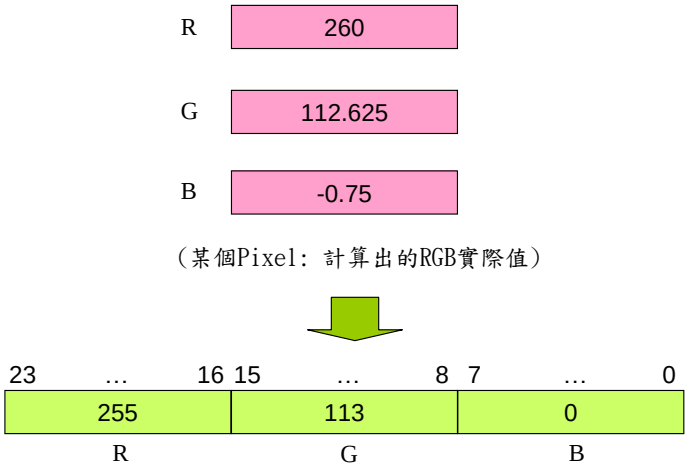
R、G、B 訊號皆為 8bits，Function 1 每次可輸出一個 Pixel，每個 Pixel 是由三個 RGB 訊號所構成，因此合計 24bits，RGB 訊號輸出格式定義如圖四所示。

(一個Pixel由RGB三個基色分量所構成)

圖四、 RGB 訊號輸出格式定義

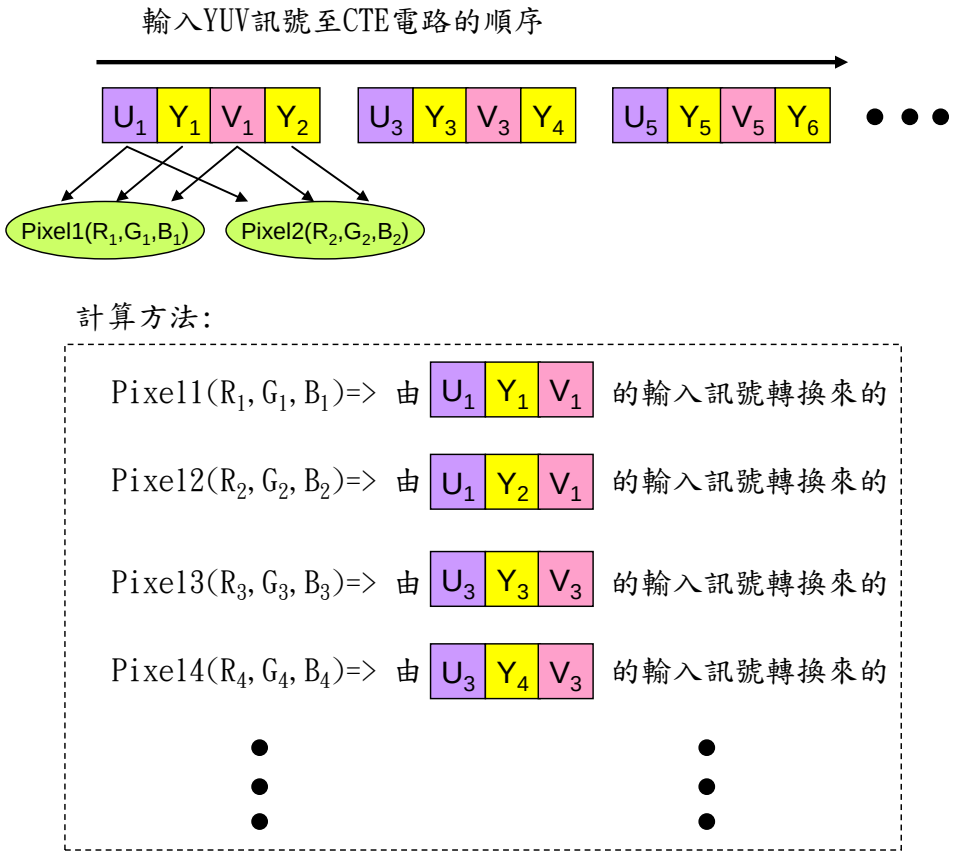
R、G、B 訊號皆為 8bits，其 R、G、B 個別訊號的輸出值範圍皆為 0-255 的整數值，亦即(1)當輸出值小於 0，輸出為 0，(2)當輸出值大於 255，輸出為 255，(3)當輸出值為 0 到 255 之間，若有小數部分將採取四捨五入法取到整數，其範例如圖五所示。

(注意: 四捨五入機制，只有在輸出前才做,計算過程中的小數部分請勿任意作四捨五入!)



圖五、 CTE Function 1 正確輸出值之範例

製作 CTE 電路 Function1 時，其計算的規則如圖六所訂定，其涵義為，CTE 電路的第一個輸出 Pixel1，其 $R_1G_1B_1$ 訊號值是用 $Y_1U_1V_1$ 的輸入訊號經由(1)式矩陣運算轉換而來的，同理，第二個輸出為 Pixel2，其 $R_2G_2B_2$ 訊號值是用 $Y_2U_1V_1$ 的輸入訊號經由(1)式矩陣運算轉換而來的，其餘以此類推。



圖六、CTE 電路 Function1 的計算規則

2.3.2.3 Function 1 之矩陣

YUV 轉換成 RGB 訊號時，輸出數值在四捨五入後必須完全符合題目要求，不容許有任何的誤差值發生，Function1 才算正確完成。

2.3.3 RGB 訊號轉換成 YUV 訊號功能描述

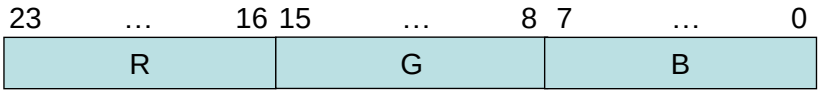
RGB 彩色模型轉換成 YUV 彩色模型，其矩陣表示式如(2)式。本 CTE 電路 Function2，如圖一所示，請完成將 RGB 訊號轉換成 YUV 訊號之功能。

Function2 →

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.29\overline{09} & 0.63\overline{03} & 0.07\overline{8} \\ -0.1\overline{45} & -0.31\overline{51} & 0.46\overline{06} \\ 0.4\overline{36} & -0.3\overline{87} & -0.04\overline{8} \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \qquad (2)$$

2.3.3.1 Function 2 之輸入端

R、G、B 訊號皆為 8bits，Function 2 每次可輸入一個 Pixel，每個 Pixel 是由三個 RGB 訊號所構成，因此共計 24bits，RGB 訊號輸入格式定義如圖七所示。R、G、B 訊號皆為 8bits，因此主辦單位所提供的 RGB 個別的輸入訊號範圍值為 0-255 的整數值。

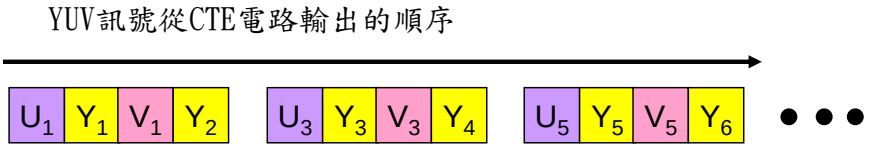


（一個Pixel由RGB三個基色分量所構成）

圖七、 RGB 訊號輸入格式定義

2.3.3.2 Function 2 之輸出端

YUV 都是 8 bits 的一維(1D)輸出訊號，U、V 訊號也採用 Down Sample 為 2 的機制，因此 CTE 電路輸出 U、V 訊號前，需自行作 Down Sample 為 2 的動作(亦即 Y 訊號假設輸出 N 筆資料量，U、V 訊號則會輸出各 N/2 筆的資料量)，Y、U、V 訊號是個別輸出的，其輸出順序採用 UYVY 格式，該格式輸出順序如圖八所示。



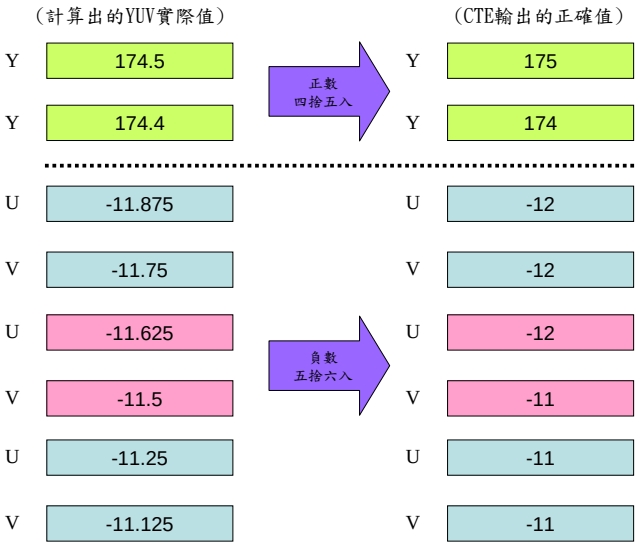
圖八、YUV 訊號個別輸出之順序(UYVY 格式)

Y、U、V 輸出訊號皆為 8bits，Y 訊號輸出範圍為 0~255 的整數值，U 訊號輸出範圍為-117~+117 的整數值，V 訊號輸出範圍為-111~+111 的整數值，當計算數值超出其輸出範圍時，必須自動修正為範圍邊界值。(註：所有負數值，都採用 **2's Complement** 來表示。)

當 Y、U、V 訊號的輸出有小數點，處理方法為:

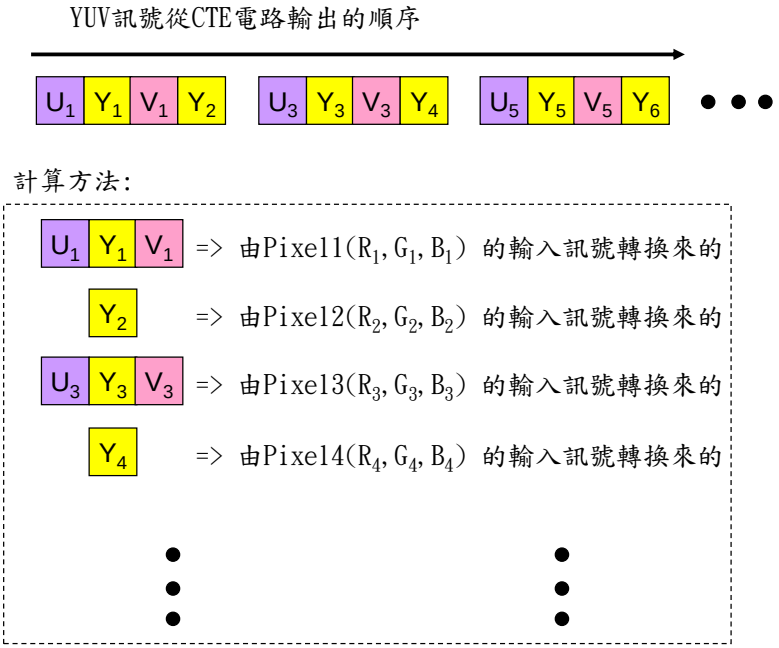
1. 若為正數，採用四捨五入法取到整數。
2. 若為負數，採用五捨六入法取到整數。

範例如圖九所示。



圖九、 CTE Function 2 正確輸出值之範例

製作 CTE 電路 Function2 時，其計算的規則如圖十所訂定，其涵義為，CTE 電路的輸出訊號 $Y_1U_1V_1$ 訊號值，可由 $R_1G_1B_1$ 的輸入訊號經由(2)式矩陣運算轉換而來的，而 Y_2 訊號值可由 $R_2G_2B_2$ 的輸入訊號經由(2)式矩陣運算轉換而來，其餘的 YUV 訊號依此類推。



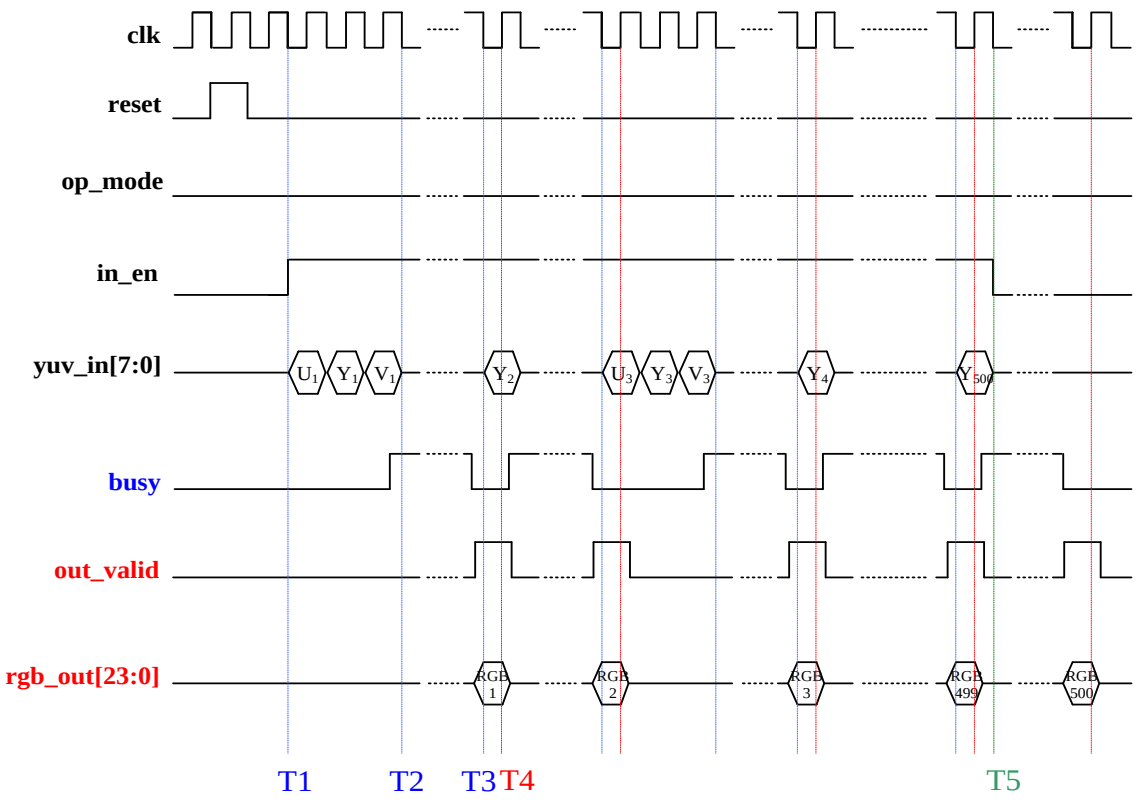
圖十、CTE 電路 Function2 的計算規則

2.3.3.3 Function 2 之矩陣

在(2)式中，矩陣的係數用符號"¯"來表示循環小數，例如：$-0.3\overline{87}$ 讀作-0.387，87 的循環，亦即 $-0.3\overline{87} \Rightarrow -0.387878787\cdots$。由於(2)式矩陣中的係數皆為循環小數，因此轉換成 YUV 訊號時，可容許有誤差值的發生，但其誤差值與 Golden Pattern 比對差異越大者，分數將會越低分。

2.4 時序規格

2.4.1 CTE Function1： YUV 訊號轉換成 RGB 訊號時序圖



圖十一、YUV 訊號轉換成 RGB 訊號時序圖

1. T1 時間點，in_en 為 High，op_mode 為 Low，系統開始進行 YUV 轉 RGB 之運算，此時 busy 為 Low，因此 Host 端便從 yuv_in 送出第一筆 U_1 訊號。

2. T2 時間點，假設參賽者可能需要在此點開始的幾個 CYCLE 內作一些處理或計算，而希望 Host 暫時停止運送新的資料進來，可以在 clk 負緣來之前(即新一筆資料要輸入前)先將 busy 訊號拉為 High。在 T2 時間點，Host 發現 busy 為 High，便停止下一筆的訊號輸入，但由於 Host 端的資料尚未送完，因此 in_ en 仍維持在 High。 (註：在圖中看到藍色的線及字，都是主要觀察 busy 的訊號。)

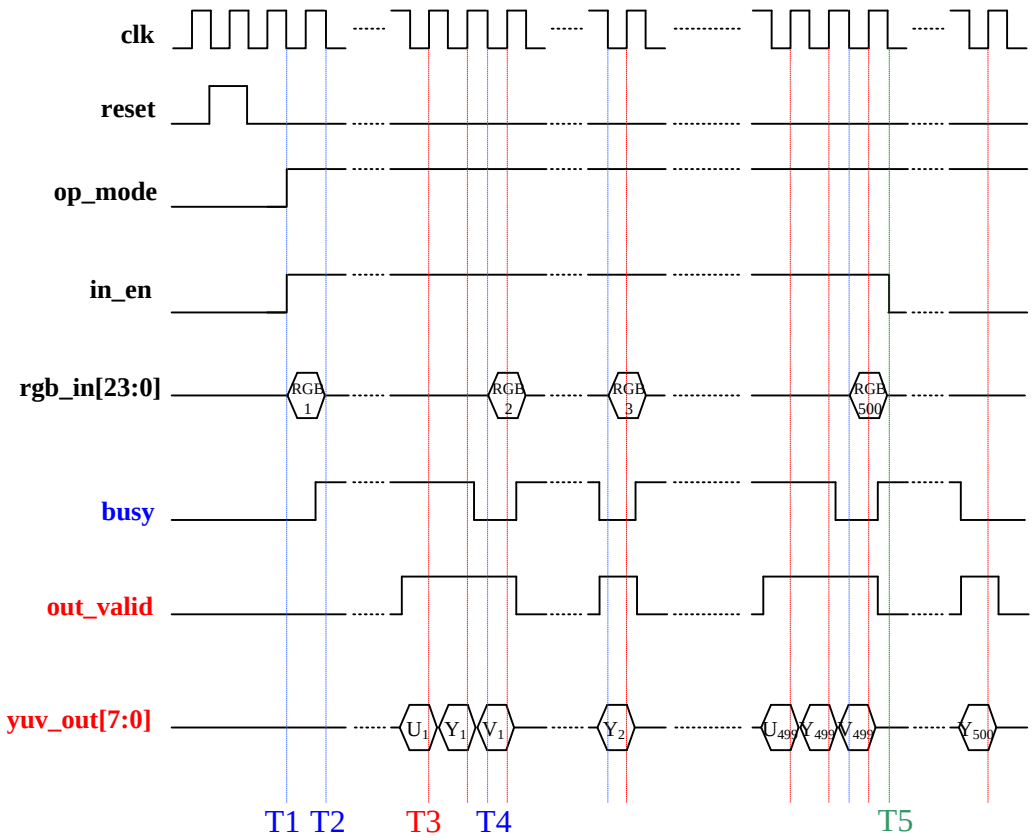
3. T3 時間點，busy 訊號為 Low，表示告知 Host 可以再送下一筆資料 Y2 進來。

4. T4 時間點，第一筆 RGB 訊號算完並將其輸出至 rgb_out，out_valid 要 High 一個 CYCLE 的時間。(註：在圖中看到紅色的線及字，都是在觀察 out_valid 及 rgb_out 的訊號。)

5. 如此反覆地輸入及輸出，直到所有資料處理完畢為止。

6. T5 時間點，in_en 為 Low 表示 500 個 Piexls 的資料量全數輸入完成。

2.4.2 CTE Function2：RGB 訊號轉換成 YUV 訊號時序圖



圖十二、RGB 訊號轉換成 YUV 訊號時序圖

1. T1 時間點，in_en 為 High，op_mode 為 High，系統開始進行 RGB 轉 YUV 之運算，此時 busy 為 Low，因此 Host 端便從 rgb_in 送出第一筆 $R_1G_1B_1$ 訊號。

2. T2 時間點，假設參賽者可能需要在此點開始的幾個 CYCLE 內作一些處理或計算，而希望 Host 暫時停止運送新的資料進來，可以在 clk 負緣來之前(即新一筆資料要輸入前)先將 busy 訊號拉為 High。在 T2 時間點，Host 發現 busy 為 High，便停止下一筆的訊號輸入，但由於 Host 端的資料尚未送完，因此 in_ en 仍維持在 High。 (註：在圖中看到藍色的線及字，都是主要觀察 busy 的訊號。)

3. T3 時間點，第一筆訊號 U1 輸出至 yuv_out，out_valid 要 High 一個 CYCLE 的時間。當然，如果緊接著下一個 CYCLE 要連續輸出 YUV 之訊號，每輸出一筆到 yuv_out，out_valid 就要維持 High 一個 CYCLE 時間。 (註：在圖中看到紅色的線及字，都是在觀察 out_valid 及 yuv_out 的訊號。)

4. T4 時間點，busy 訊號為 Low，表示告知 Host 可以再送下一筆 RGB 訊號進來。

5. 如此反覆地輸入及輸出，直到所有資料處理完畢為止。

6. T5 時間點，in_en 為 Low 表示 500 個 Piexls 的資料量全數輸入完成。

3.評分標準

各參賽隊伍必須做到"完成設計"，即完成下列四項要求。倘若達成"完成設計"的組數太少，主辦單位才會考慮依這四項要求，來進行同級完成度的評分排名，完成越多項目之隊伍成績越佳。注意，完成本設計請按照 a、b、c、d 的要求順序完成。另外，主辦單位的評分人員將依照參賽者提供之週期時間(CYCLE TIME)來進行模擬及驗證設計之正確性，各參賽隊伍請先確認所給定的 CYCLE TIME 完全沒有設置與保持時間(setup/hold time)的問題。

✧ **"完成設計"的四項要求：**

a、 YUV 訊號轉換成 RGB 訊號，RTL 模擬必須與正確結果(golden pattern)完全一樣。

b、 RGB 訊號轉換成 YUV 訊號，RTL 模擬與 golden pattern 比對，依誤差值進行評分。

c、 完成 Synthesis，且 Gate-Level Pre-layout Simulation 有達到 a、b 項之要求。

d、 完成 APR，DRC、LVS 驗證無誤，且 Gate-Level Post-layout Simulation 有達到 a、b 項之要求。

✧ **依"誤差值"進行第一階段評分：**

主辦單位在 Test Bench 檔案裡，已加入(3)式來計算"RGB 訊號轉換成 YUV 訊號之誤差值"，依據此誤差值進行評分排名，共區分六個等級如下：

$$\text{error} = \frac{\sum(Y - Y')^2 + \sum(U - U')^2 + \sum(V - V')^2}{\sum Y^2 + \sum U^2 + \sum V^2} \qquad (3)$$

註： Y、U、V ：主辦單位所提供 golden pattern 之訊號值。

Y'、U'、V'：參賽隊伍所設計的 CTE 電路，其 YUV 訊號實際輸出值。

A 級：error < 0.0000002

B 級：0.0000002 ≤ error < 0.0000005

C 級：0.0000005 ≤ error < 0.0000010

D 級：0.0000010 ≤ error < 0.0000050

E 級：0.0000050 ≤ error < 0.0000300

F 級： error ≥ 0.0000300 => 主辦單位視 F 級為 Function2 是錯誤的(Fail)，請注意!

註：Function2 有三組測試樣本，三個都要模擬，分數是三個模擬結果中，取最差的一個。

✧ **依"Cost"作第二階段評分：**

根據上述等級，主辦單位會將同等級之隊伍依(4)式，作二度的評分，Cost 越低成績越佳。

$$\text{Cost} = (T1 + T2) * \text{Area} \qquad (4)$$

T1：YUV 訊號轉換成 RGB 訊號，完成所有轉換之實際模擬時間(Real Simulation Time)

T2：RGB 訊號轉換成 YUV 訊號，完成所有轉換之實際模擬時間(Real Simulation Time)

Area：APR 之後，layout 實際面積大小(單位：um^2)

✧ **實際模擬時間(Real Simulation Time)之補充：**

範例 1: CTE Function1 的 T1 值
假設參賽者作 CTE Function1 之模擬結果如下:

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

Congratulations! All data have been generated successfully!

-----------------------PASS-----------------------

Simulation complete via $finish(1) at time 8635 NS + 0
./testfixture1.v:126            #(`CYCLE/2); $finish;
```

在本例中，T1 = 8635ns。

範例 2: CTE Function2 的 T2 值
假設參賽者作 CTE Function2 之模擬結果如下:

```
ncsim> run
-----------------------------------------------
Square Distance of All YUV = 0.000000
Square of All YUV Signal     = 23192033.000000
-----------------------------------------------
So Your Error Ratio:
(Square Distance of YUV)/(Square of All YUV Signal) = 0.000000
-----------------------------------------------
Your Score Level: A
Congratulations! CTE's Function2 Successfully!
-----------------------PASS-----------------------
Simulation complete via $finish(1) at time 13244 NS + 0
./testfixture2.v:188            #(`CYCLE/2); $finish;
```

在本例中，T2 = 13244ns。

附錄

附錄 A 為主辦單位所提供各參賽者的設計檔說明；附錄 B 為主辦單位提供的測試樣本說明；附錄 C 為設計驗證說明；附錄 D 為評分用檔案，亦即參賽者必須繳交的檔案資料；附錄 E 則為設計檔案壓縮整理步驟說明；附錄 F 中說明本次競賽之軟體環境；附錄 G 中說明本次競賽使用之設計資料庫。

附錄 A 設計檔(For Verilog or VHDL)

1. 下表為主辦單位所提供各參賽者的設計檔

表 2、設計檔案說明

檔名	說明
CTE.v	參賽者所使用的設計檔，已包含系統輸/出入埠之宣告
testfixture1.v	測試樣本檔 1。用來驗證 CTE Function1:YUV 轉換成 RGB 訊號功能是否正確。參賽者請自行調整本檔案定義之 CYCLE 值，確保在沒有 Setup/Hold Time 的問題下，花最少實際模擬時間，完成所有驗證測試。
testfixture2.v	測試樣本檔 2。用來驗證 CTE Function2: RGB 轉換成 YUV 訊號功能是否正確。參賽者請自行調整本檔案定義之 CYCLE 值，確保在沒有 Setup/Hold Time 的問題下，花最少實際模擬時間，完成所有驗證測試。 註: Function2 的測試樣本有三組，參賽者請自行修改檔名，共計三次模擬。注意三次模擬之對應關係。 模擬 1: pattern_rgb1.dat => golden_yuv1.dat 模擬 2: pattern_rgb2.dat => golden_yuv2.dat 模擬 3: pattern_rgb3.dat => golden_yuv3.dat
pattern_yuv.dat	提供一組測試樣本，提供 500 個 Pixels 資料量作為 CTE Function1 的 Input Pattern。採 UYVY 順序輸入，其中 U、V 訊號已作 DownSample2。(YUV 訊號以 16 進位表示)
golden_rgb.dat	CTE Function1 : YUV 訊號轉換成 RGB 訊號之正確結果。(以 16 進位表示)
pattern_rgb1.dat pattern_rgb2.dat pattern_rgb3.dat	提供三組測試樣本，皆為 500 個 Pixels 資料量作為 CTE Function2 的 Input Pattern。每次輸入為一個 Pixel 的 R_G_B 訊號，故每筆合計 24bits。 (RGB 訊號以 16 進位表示)

golden_yuv1.dat golden_yuv2.dat golden_yuv3.dat	提供三組測試樣本之正確結果檔案，作為 CTE Function2 : RGB 訊號轉換成 YUV 訊號之正確結果。(以 16 進位表示) 註：因轉換矩陣系數為循環小數，在此正確結果意指使用真實數值計算後四捨五入(負數為五捨六入)成 8 位元的結果。
.synopsys_dc.setup	使用 Design Compiler 作合成之初始化設定檔。參賽者請依 Library 實際擺放位置，自行填上 Search Path 的設定。
CTE_DC.sdc	使用 Design Compiler 作合成之 sdc 檔。參賽者可自行調整最佳之 cycle 值。
CTE_SOCE.sdc	使用 SOC Encounter 作 Layout 之 sdc 檔。參賽者可自行調整最佳之 cycle 值。
CTE_Astro.sdc	使用 Astro 作 Layout 之 sdc 檔。參賽者可自行調整最佳之 cycle 值。

請使用 ***CTE.v***，進行 Color Transform Engine 之設計。其模組名稱、輸出/入埠宣告如下所示：

```
module CTE ( clk, reset, op_mode, in_en, yuv_in, rgb_in, busy, out_valid, rgb_out, yuv_out);
    input     clk ;
    input     reset ;
    input     op_mode;
    input     in_en;
    output    busy;
    output    out_valid;
    input     [7:0]     yuv_in;
    output    [23:0]    rgb_out;
    input     [23:0]    rgb_in;
    output    [7:0]     yuv_out;

endmodule
```

2. 主辦單位提供 testfixture1.v 及 testfixture2.v，分別作為測試 CTE Function1 及 Function2 之用。Function1 有一組 Pattern 要測試，Function2 有三組要測試，請注意。Function2 有三組樣本要測試，請先開啟 testfixture2.v 檔再修改，修該方法如下：

   第一組 Function2 模擬：

```
`define   PAT      "./pattern_rgb1.dat"
`define   EXP      "./golden_yuv1.dat"
```

第二組 Function2 模擬：

`define PAT "./pattern_rgb2.dat"

`define EXP "./golden_yuv2.dat"

第三組 Function2 模擬：

`define PAT "./pattern_rgb3.dat"

`define EXP "./golden_yuv3.dat"

因此 Function2 有三組樣本要模擬，分數是三個模擬結果中，取最差的一組，請注意。設計過程中若欲透過 Waveform 作 Debug，可以從 test module 底下的 i 訊號得知目前是第幾筆 Pattern 輸入，**exp_num** 訊號得知目前是第幾筆 Pattern 輸出。

附錄 B 測試樣本

為了讓參賽者看完題目後，更能確定題意，在此分別針對 CTE Function1 及 Function2 各提供前 16 組的 Input Pattern 及 Golden Pattern 讓參賽者試算用。

測試樣本一

CTE Function1			
YUV Input Signal		RGB Output Signal	
Order	Signal	Order	Signal
U1 Y1 V1 Y2	1C 72 54 C1	R1_G1_B1	FB_2C_AA
		R2_G2_B2	FF_7B_F9
U3 Y3 V3 Y4	0 99 2A A8	R3_G3_B3	DD_7A_99
		R4_G4_B4	EC_89_A8
U5 Y5 V5 Y6	D1 CE D9 52	R5_G5_B5	8F_F7_70
		R6_G6_B6	13_7B_00
U7 Y7 V7 Y8	10 1B 4 B5	R7_G7_B7	22_14_3B
		R8_G8_B8	BC_AE_D5

U7 Y7 V7 Y8	FF 86 36 78	R9_G9_B9	DE_5E_84
		R10_G10_B10	D0_50_76
U7 Y7 V7 Y8	FB 63 47 21	R11_G11_B11	D6_2F_59
		R12_G12_B12	94_00_17
U7 Y7 V7 Y8	16 3E 16 9A	R13_G13_B13	62_28_6A
		R14_G14_B14	BE_84_C6
U7 Y7 V7 Y8	10 44 EE 76	R15_G15_B15	27_4E_64
		R16_G16_B16	59_80_96

測試樣本二

CTE Function2			
RGB Input Signal		YUV Output Signal	
Order	Signal	Order	Signal
R1_G1_B1	3D_A5_C5	U1 Y1 V1 Y2	1E 89 D1 73
R2_G2_B2	27_8F_AF		
R3_G3_B3	E3_53_DE	U3 Y3 V3 Y4	2B 88 38 65
R4_G4_B4	C0_30_BB		
R5_G5_B5	9A_63_A0	U5 Y5 V5 Y6	14 78 15 C3
R6_G6_B6	E5_AE_EB		
R7_G7_B7	E9_E3_37	U7 Y7 V7 Y8	B0 D7 0B 66
R8_G8_B8	73_6D_00		

R9_G9_B9	4B_AD_72	U7 Y7 V7 Y8	F3 8C D8 5F
R10_G10_B10	1E_80_45		
R11_G11_B11	4B_F3_49	U7 Y7 V7 Y8	CA B5 BF 9D
R12_G12_B12	33_DB_31		
R13_G13_B13	0D_35_2F	U7 Y7 V7 Y8	3 29 EF 6E
R14_G14_B14	52_7A_74		
R15_G15_B15	35_92_F0	U7 Y7 V7 Y8	39 7E D3 65
R16_G16_B16	1C_79_D7		

附錄 C 設計驗證說明

參賽者繳交資料前應完成 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：參賽者必須進行 RTL simulation 及 Gate-Level simulation，模擬結果必須於題目所定義的系統時脈下，**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. **VDD 與 VSS power ring 寬度請各設定為 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**，以下分為 Astro 及 SOC Encounter 兩軟體說明 netlist 與 sdf 寫出步驟。
   i. 使用 Synopsys Astro 者，執行步驟如下：
   在 Astro 視窗底下點選
   “ Timing > SDF Out ”

Specify Version	Version 2.1
Operation Mode	Normal SDF
File Name	CTE_pr.sdf

按 OK。

“ Cell > Hierarchical Verilog Out ”

Flattened Cell Name (.EXP .CEL)	CTE.CEL
Enter File Name	CTE_pr.v
No power/ground ports	Enable
No power/ground nets	Disable
Output bus as individual bits	Disable
No empty Cell Module Definitions	Enable
No Corner Pad Instances	Enable
No Pad Filler Cell Instances	Enable
No Core Filler Cell Instances	Enable
No Unconnected Cell Instances	Enable
No Unconnected Ports	Enable
Strip BackSlash Before Hierarchy Separator	Enable
No Diode Ports	Enable
Output Wire Declaration	Enable
Output 1’b1 for Power(VDD, vdd, …) and 1’b0 for Ground(VSS, gnd, …)	Enable
Generate macro definitions	Disable

按 OK。

ii. 使用 Cadence SOC Encounter 者，執行步驟如下：

在 SOC Encounter 視窗下點選：

“ Design → Save → Netlist… ”

Netlist File	CTE_pr.v
All other options	Default value

按 OK。

“ Timing → Calculate Delay… ”

存成 CTE_pr.sdf，按 OK。

註：如果發現 Calculate Delay 功能是灰色的(無法點選)，請先將目前結果存檔後離開 Encounter，再重新進入 Encounter 並 Restore 回原本 Design 即可。

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，以下分為 Astro 及 SOC Encounter 說明執行步驟。**

i. 使用 Synopsys Astro 者，驗證 DRC 與 LVS 步驟如下：

在 Astro 視窗底下點選

“ Verify > DRC ”

List Error Summary Immediately	Enable
All other options	Default value

按 OK。

將跳出來的 DRC report 存成 DRC.report 檔。

“ Verify > LVS ” Default 值，按 OK。

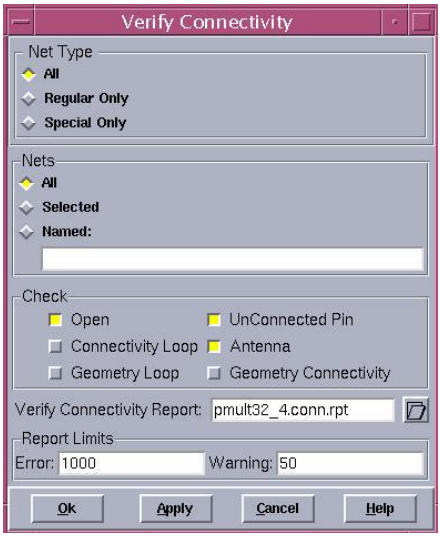
List Error Summary Immediately	Enable
All other options	Default value

將跳出來的 LVS report 存成 LVS.report 檔。

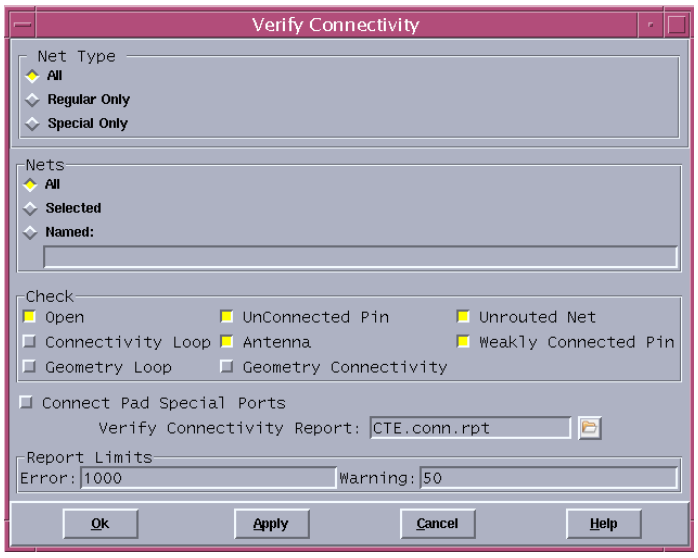
ii. 使用 Cadence SOC Encounter 者，驗證 DRC 與 LVS 步驟如下：

在 SOC Encounter 視窗下點選

“Verify → Verify Connectivity… ” Default 值，按 OK。



“Verify → Verify Geometry… ” Default 值，按 OK。



"Verify → Violation Browser… "
將 Verify 的結果存成 CTE.viols.rpt

附錄 D 評分用檔案

評分所須檔案可以下幾個部份：(1)RTL design，即各參賽隊伍對該次競賽設計的 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，以免評審進行評分時，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，**使用 Synopsys Astro 者，請記得將 Astro 整個相關的 design library，壓縮成一個檔案。使用 Cadence SOC Encounter 者，請將 SOC Encounter 相關的 design library（包含一個.enc 檔及一個.dat 目錄），壓縮成一個檔案。**壓縮的檔案格式如下：假設參賽者的 design library 目錄名稱為"your_lib"，請執行底下的 UNIX 指令，最後可以得到"your_name.tar"的檔案。

> ***tar cvf your_name.tar your_lib***

在執行以上的指令之前，請確定將你使用的 P&R Tool 儲存後關閉，再執行以上的指令，否則在壓縮的過程會出現錯誤。

表 3

RTL category		
Design Stage	*File*	*Description*
N/A	`N/A`	Design Report Form
RTL Simulation	`*.v or *.vhd`	Verilog (or VHDL) synthesizable RTL code
Gate-Level category		
Design Stage	*File*	*Description*
Pre-layout Gate-level	`*_syn.v`	Verilog gate-level netlist generated by Synopsys Design Compiler

Simulation	*_syn.sdf	Pre-layout gate-level sdf
Physical category		
Design Stage	*File*	*Description*
P&R	*.tar	**archive of the design library directory**
	*.gds	GDSII layout
	DRC/LVS report	For Astro：DRC.report；LVS.report For SOC Encounter：CTE.viols.rpt
Post-layout Gate-level Simulation	*_pr.v	Verilog gate-level netlist generated by Cadence SOC Encounter or Synopsys Astro
	*_pr.sdf	Post-layout gate-level sdf

附錄 E 檔案整理步驟

當所有的文件準備齊全如表 3 所列，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，步驟如下:

1. 在自己的 home directory 建立一個新目錄，名稱叫做"**result**" 例如：
   > ***mkdir ~/result***
2. 將附錄 D 要求的檔案複製到 result 這個目錄。例如:
   > ***cp CTE.v ~/result/***
   > ***cp CTE_syn.v ~/result/***
   .....
3. 在 Design Report Form 中，填入所需的相關資訊。

附錄 F 軟體環境

1. 軟體環境設定檔： /usr/cad/cshrc/env.cshrc
2. 設定軟體環境，請在登入後，開啟 terminal 視窗並依以下步驟執行：

   cp /usr/cad/cshrc/env.cshrc .cshrc
   source .cshrc
3. 此 cshrc 所設定好的軟體環境包括：
   NC-Verilog
   NC-VHDL
   SOC Encounter
   Verdi
   Laker
   ModelSim
   Design Vision
   Astro
   joe

textedit
nedit
vim
gvim
xv

EDA 軟體所須使用的 license 皆已設定完成，不須額外設定，且每組限定**每個軟體只能使用一套 license**。

附錄 G 設計資料庫

設計資料庫位置： **/usr/cad/icc2009/CBDK_IC_Contest_v2.0**

目錄架構

Astro/	
tsmc13gfsg_fram/	Astro core library
tsmc13_CIC.tf	Astro technology
macro.map	layer mapping file
SOCE/	
lef/	
tsmc13fsg_8lm_cic.lef	LEF for core cell
antenna_8.lef	LEF for antenna
lib/	
fast.lib	best case for core cell
slow.lib	worst case for core cell
typical.lib	typical case for core cell
streamOut.map	Layout map for GDSII out
SynopsysDC/	
db/	
fast.db	Synthesis model (fast)
slow.db	Synthesis model (slow)
typical.db	Synthesis model (typical)
lib/	
fast.lib	timing and power model
slow.lib	timing and power model
typical.lib	timing and power model
Verilog/	
tsmc13_neg.v	Verilog simulation model
VHDL/	
tsmc13.vhd	VHDL simulation model

Design Report Form

隊號(Team number):		
RTL category		
Design Stage	*Description*	*File Name*
RTL Simulation	使用之 HDL 名稱 (請填入 Verilog 或 VHDL)	
RTL Simulation	RTL 檔案名稱 (RTL Netlist file name)	
Gate-Level category		
Design Stage	*Description*	*File Name*
Pre-layout Gate-level Simulation	Gate-Level 檔案名稱 (Gate-Level Netlist file name)	
	Pre-layout sdf 檔案名稱	
	Gate-Level simulation, 所使用最小的 CYCLE Time	( ) ns
Physical category		
Design Stage	*Descritpion*	*File Name or Value*
P&R	使用之 P&R Tool (請填入 Astro 或 SOC Encounter)	
	設計資料庫檔案名稱(Library name)	
	佈局檔檔案名稱(GDSII file name)	
	佈局面積(layout area)	( ) um X ( ) um
	佈局座標點	左下角座標點(Lower-Left Coordinate) : XLB = YLB = 右上角座標點(Upper-Right Coordinate): XRT = YRT =
	DRC report file	
	LVS report file	
Post-layout Gate-level Simulation	Post-layout Gate-Level 檔案名稱	
	Post-layout sdf 檔案名稱	
	Gate Level simulation, 所使用最小的 CYCLE Time (小數點只有一位)	Function1:( ) ns Function2:( ) ns
	Gate Level simulation, 完成所有運算所需的時間	T1: ( ) ns T2: ( ) ns
其他說明事項(Any other information you want to specify:(如設計特點 ...) 如寫不下可寫於背面		