

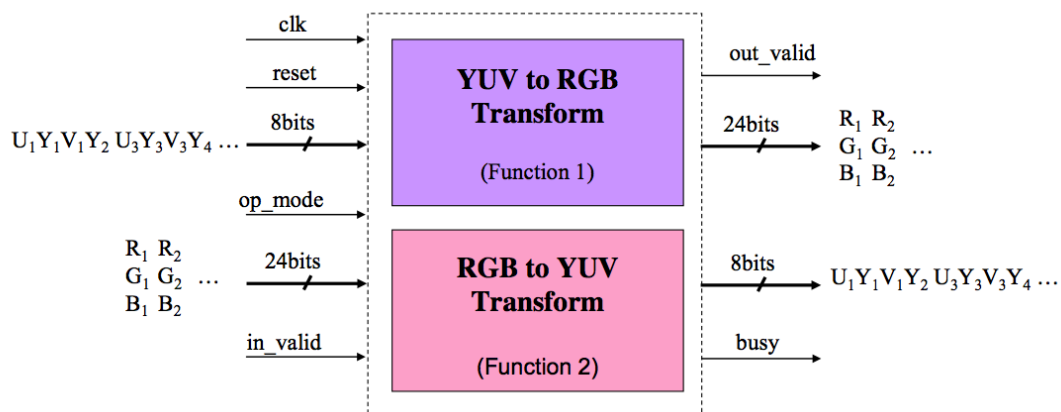
Computer-Aided VLSI System Design, Fall 2011

Verilog HW 2

1. Problem Description:

請完成Color Transform Engine(CTE)的電路設計。如圖一,本CTE電路功能有二:

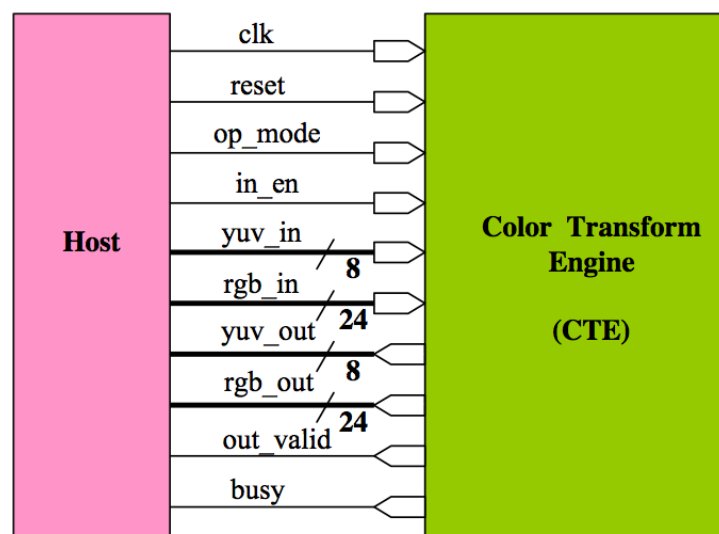
- (1)將彩色訊號的每個Pixel之YUV訊號轉換成RGB訊號,
- (2)將彩色訊號的每個Pixel之RGB訊號轉成YUV訊號,其詳細規格將描述於後。



圖一、彩色訊號轉換功能之方塊圖

2. Specification

2.1 Block Diagram of System



圖二、系統方塊圖

2.2 Input Signals and Output Signals

Signal Name	I/O	Width	Simple Description
clk	I	1	The clock of CTE, all signal timing are related to the rising edge of clk.
reset	I	1	The reset is an active high asynchronous signal.
op_mode	I	1	Operation mode: op_mode=low(YUV->RGB) op_mode=high(RGB->YUV)
in_en	I	1	Input data enable signal: in_en=0(disable) in_en=1(enable)
busy	O	1	The busy signal of CTE: busy=0 (Host can input YUV or RGB data) busy=1 (Host can not input YUV or RGB data)
yuv_in	I	8	Y,U, and V are 8-bit input data. When in_en=1 and busy=0, input data is valid.
rgb_in	I	24	RGB is a 24-bit input data bus. (R,G,B) When in_en=1 and busy=0, input data is valid.
yuv_out	O	8	Y,U, and V are 8-bit output data.
rgb_out	O	24	RGB is a 24-bit output data bus. (R,G,B)
out_valid	O	1	Output control signal: out_valid=0 (YUV or RGB is a invalid output data) out_valid=1 (YUV or RGB is a valid output data)

2.3 System Description

2.3.1 Introduction to YUV and RGB

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) \Rightarrow (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) \Rightarrow (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 Function of YUV to RGB

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。

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

2.3.2.1 Input of Function 1

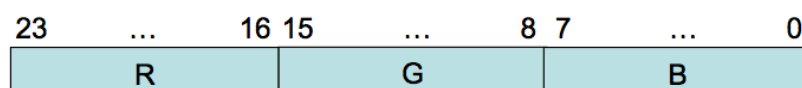
YUV 都是 8 bits 的一維(1D)輸入訊號,Y、U、V 輸出訊號皆為 8bits,其中 Y 訊號輸入範圍為 0~255 的整數值,U 訊號輸入範圍為-117~+117 的整數值,V 訊號輸入範圍為-111~+111 的整數值。所提供的 YUV 輸入訊號已事先針對 U、V 訊號作 Down Sample 2 之處理,因此 Y 訊號假設提供 N 筆資料量,則 U、V 訊號提供為各 N/2 筆資料量,Y、U、V 訊號是個別輸入的,其輸入順序採用 UYVY 格式,該格式輸入順序如圖三所示。(註:所有負數值,都採用 2 的補數(2's Complement)來表示。)



圖三、YUV 訊號個別輸入之順序(UYVY 格式)

2.3.2.2 Output of Function 1

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。

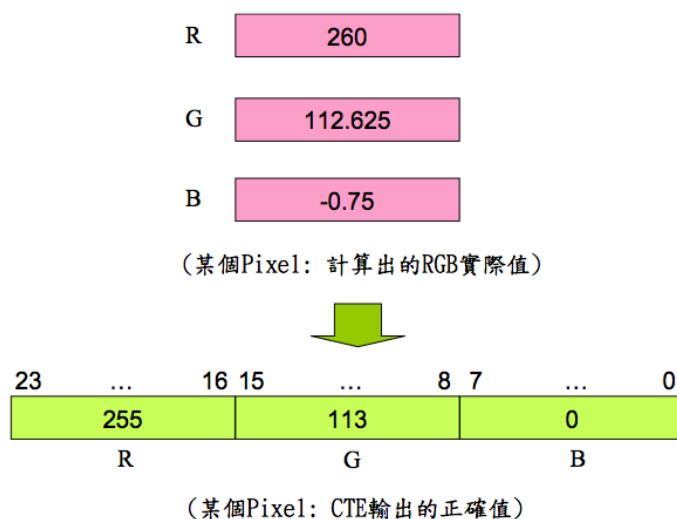


(一個Pixel由RGB三個基色分量所構成)

圖四、RGB 訊號輸出格式定義

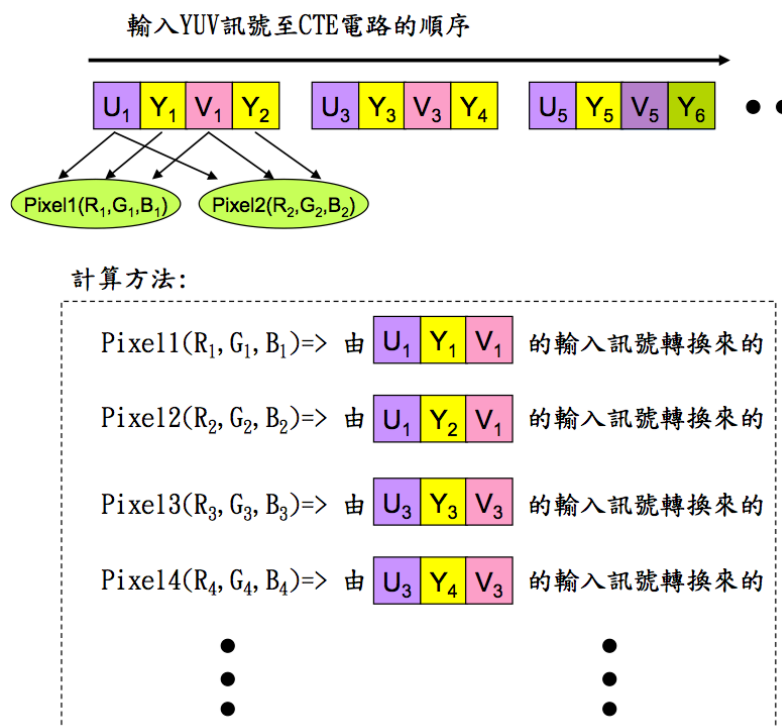
R、G、B 訊號皆為 8bits,其 R、G、B 個別訊號的輸出值範圍皆為 0-255 的整數值,亦即 (1)當輸出值小於 0,輸出為 0,(2)當輸出值大於 255,輸出為 255,(3)當輸出值為 0 到 255 之間,若有小數部分將採取四捨五入法取到整數,其範例如圖五所示。

(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 Output Value of Function 1

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。

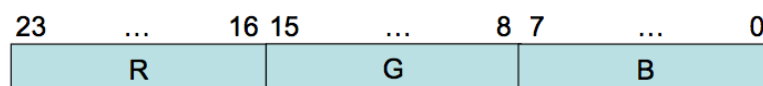
2.3.3 Function of RGB to YUV

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。

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.2909 & 0.6303 & 0.078 \\ 0.145 & -0.3151 & 0.4606 \\ 0.436 & 0.387 & 0.048 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2)$$

2.3.3.1 Input of Function 2

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 Output of Function 2

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。



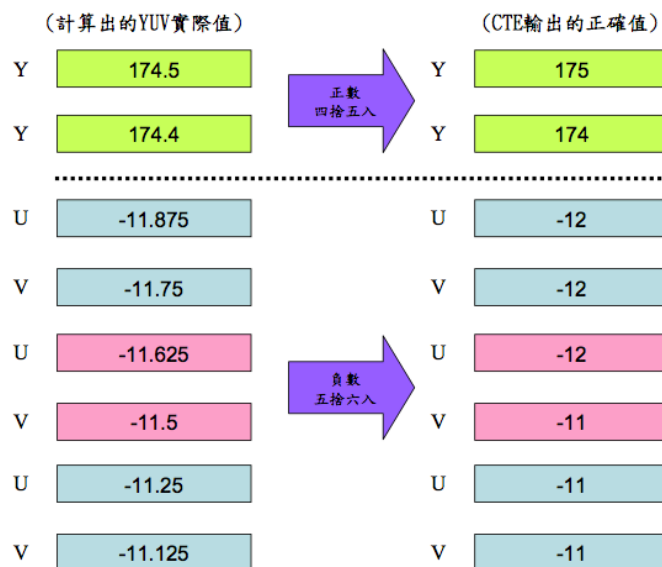
圖八、YUV 訊號個別輸出之順序(UYVY 格式)

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:

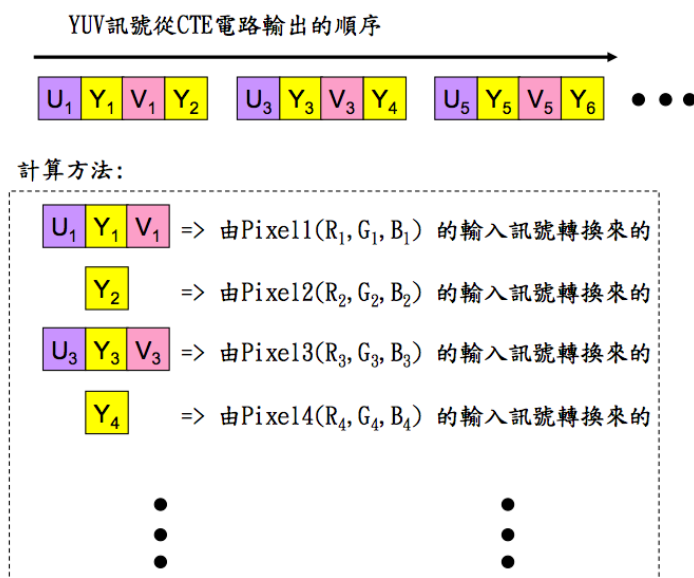
1. 若為正數,採用四捨五入法取到整數。
2. 若為負數,採用五捨六入法取到整數。

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 $Y_i U_i V_i$ 訊號值,可由 $R_i G_i B_i$ 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 $R_2 G_2 B_2$ 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。



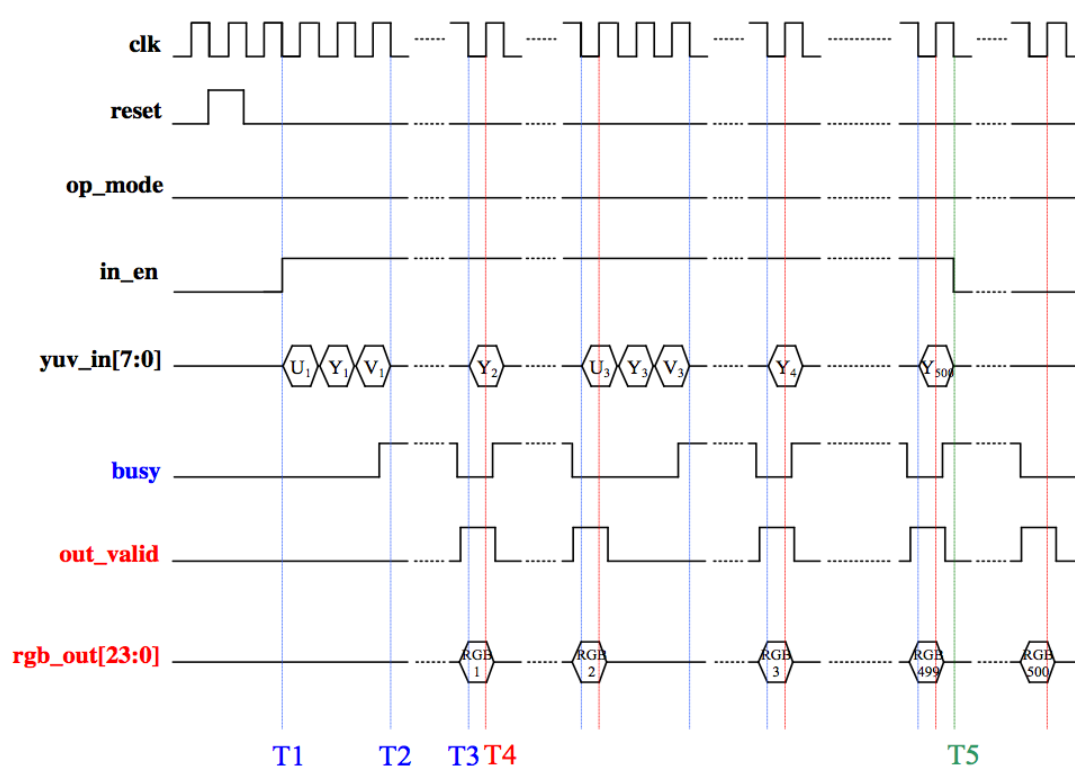
圖十、CTE 電路 Function2 的計算規則

2.3.3.3 Output Value of Function 2

在(2)式中,矩陣的係數用符號"—"來表示循環小數,例如: $-0.38\overline{7}$ 亦即 $-0.387 \Rightarrow -0.387878787\cdots$ 。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比對差異越小者,分數將會越高分。

2.4 Timing Diagram

2.4.1 CTE Function1: YUV to RGB

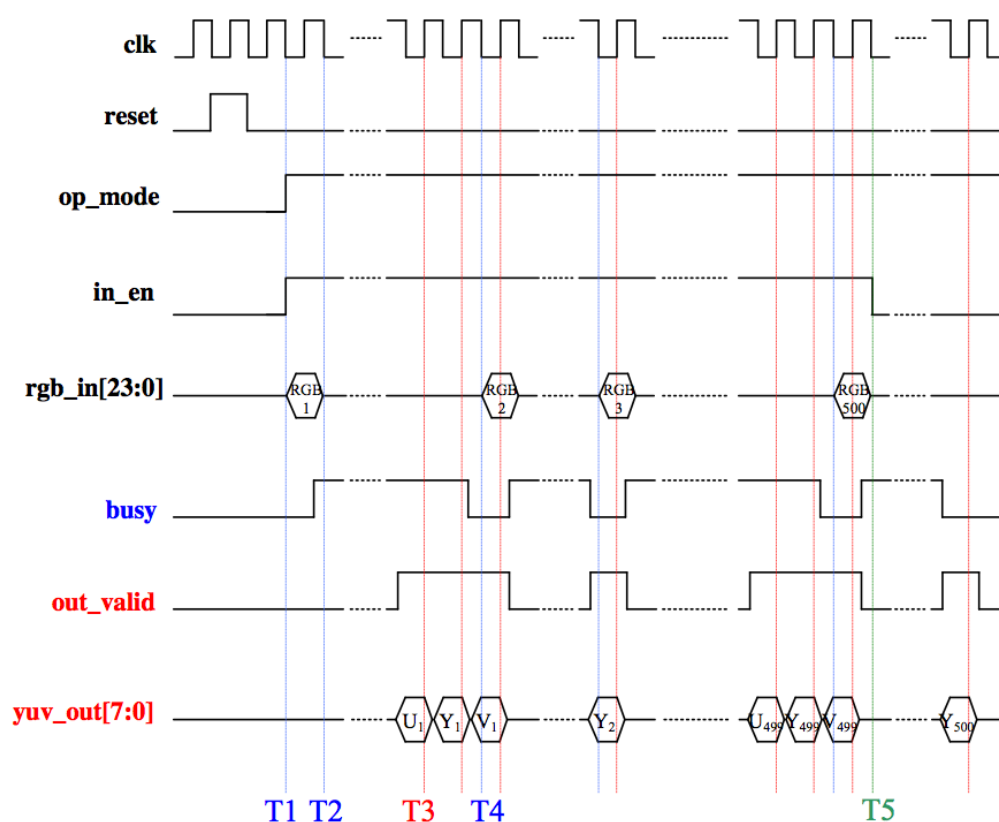


圖十一、YUV 訊號轉換成 RGB 訊號時序圖

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,假設可能需要在該點開始的幾個 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Pixls的資料量全數輸入完成。

2.4.2 CTE Function2: RGB to YUV



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 R₁G₁B₁ 訊號。
2. T2時間點,假設**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Pixels的資料量全數輸入完成。

3. Grading Policy

3.1 評分規定:

- a. (40%) YUV → RGB: RTL模擬必預與golden pattern完全一樣。
- b. (60%) RGB → YUV: RTL模擬與 golden pattern 比對,依誤差值進行評分。

3.2 依”誤差值”進行RGB→YUV評分:

在Test Bench檔案裡,已加入(3)式來計算”RGB → 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:

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

註: Y、U、V : golden pattern 之訊號值。 Y'、U'、V': 設計者所設計的CTE電路,其 YUV 訊號實際輸出值。

- A 級: error < 0.0000002 (60分)
- B 級: 0.0000002 ≤ error < 0.0000005 (50分)
- C 級: 0.0000005 ≤ error < 0.0000010 (40分)
- D 級: 0.0000010 ≤ error < 0.0000050 (30分)
- E 級: 0.0000050 ≤ error < 0.0000300 (20分)
- F 級: error ≥ 0.0000300 (10分)

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

附錄：設計檔案說明

表 2、設計檔案說明

檔名	說明
CTE.v	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→ RGB訊號功能是否正確。 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→ YUV訊號功能是否正確。 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,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。