

2017 Digital IC Design

Homework 5: Color Transform Engine

1. Introduction

請完成一Color Transform Engine(後文以CTE 表示)的電路設計。本CTE 電路功能有二，將彩色訊號的每個像素(Pixel)之YUV 訊號轉換成RGB 訊號，以及將彩色訊號的每個Pixel 之RGB 訊號轉成YUV 訊號。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。

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 Function 1 之功能是将 YUV 訊號轉換成 RGB 訊號。

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

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。Testbench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fig. 1.所示。

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

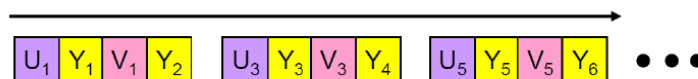


Fig. 1. YUV 訊號個別輸入之順序(UYVY 格式)

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 fig. 2.所示。

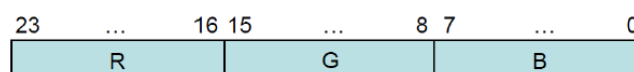


Fig. 2. RGB 訊號輸出格式定義

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，當輸出值小於0，輸出為0，當輸出值大於255，輸出為255，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fig. 3.所示。(注意: 四捨五入機制，只有在輸出才做,計算過程中的小數部分請勿任意作四捨五入!)

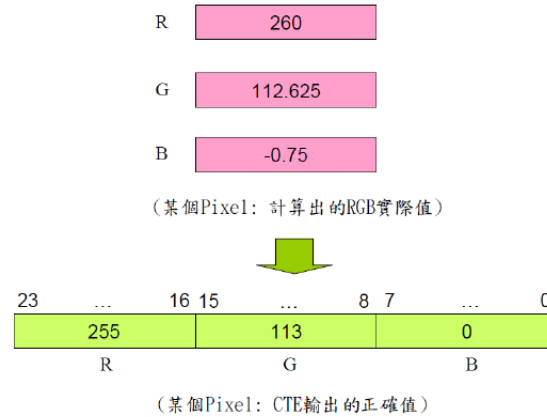


Fig. 3. CTE Function 1 正確輸出值之範例

CTE 電路 Function 1 計算的規則如 fig.4.所訂定，其涵義為，CTE 電路的第一個輸出 Pixel 1，其 R1G1B1 訊號值是用 Y1U1V1 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 Pixel 2，其 R2G2B2 訊號值是用 Y2U1V1 的輸入訊號經由(1)式矩陣運算轉換而來的，其餘以此類推。

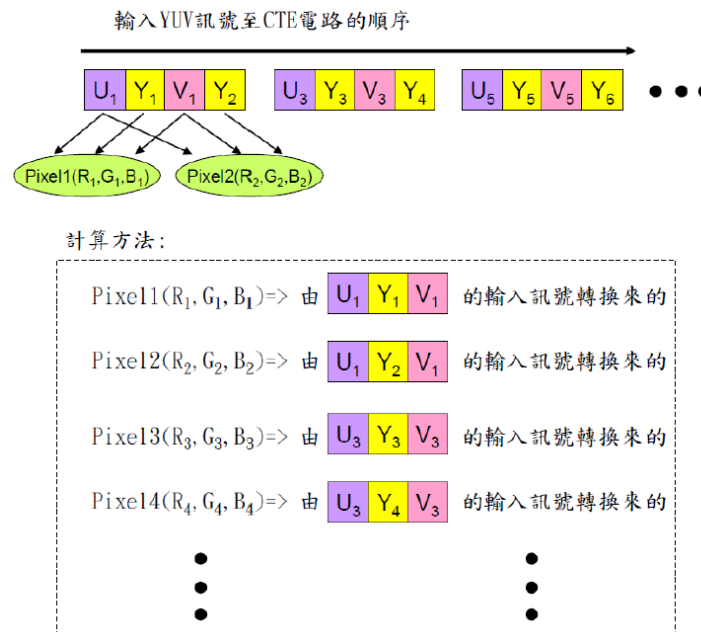


Fig. 4. CTE 電路 Function1 的計算規則

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。

RGB 彩色模型轉換成 YUV 彩色模型，其矩陣表示式如(2)式。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，R、G、B 訊號皆為 8bits，因此題目所提供的 RGB

個別的輸入訊號範圍值為 0-255 的整數值。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 fig. 5.所示。

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

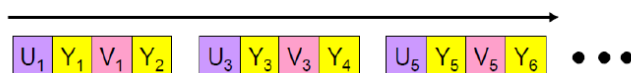


Fig. 5. YUV 訊號個別輸出之順序(UYVY 格式)

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。

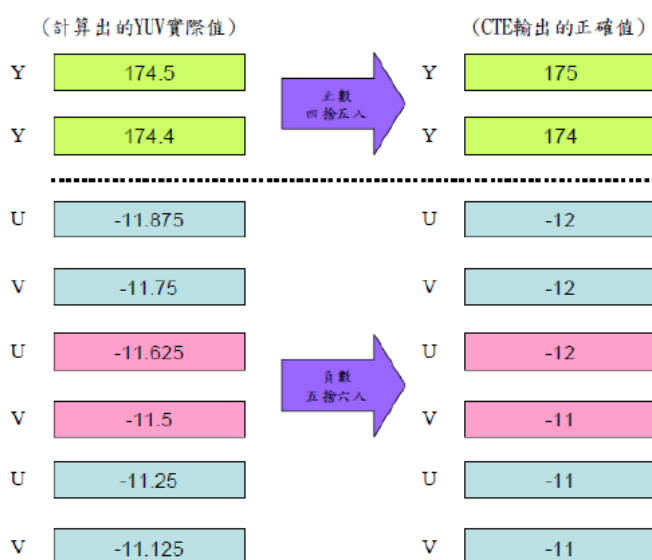


Fig. 6. YUV 輸出值之範例

CTE 電路 Function2 計算的規則如 fig. 7.所訂定，其涵義為，CTE 電路的輸出訊號 Y1U1V1 訊號值，可由 R1G1B1 的輸入訊號經由(2)式矩陣運算轉換而來的，而 Y2 訊號值可由 R2G2B2 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。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。

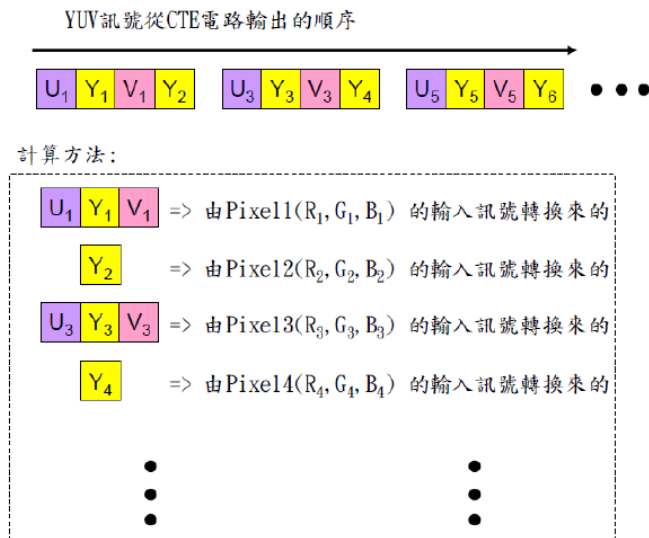
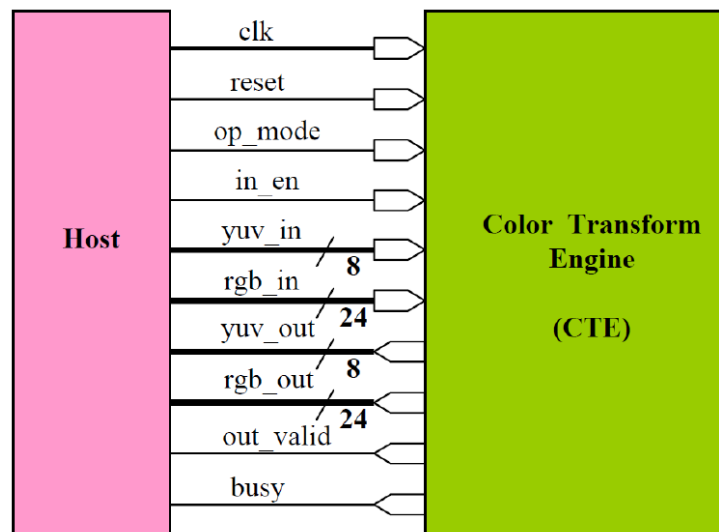


Fig. 7. CTE 電路 Function2 的計算規則

2. Design Specifications

2.1 block diagram



2.2 I/O Interface

訊號名稱	輸出/入	位元	說明
clk	input	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	input	1	高位準非同步(active high asynchronous)之系統重置信號。
op_mode	input	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。

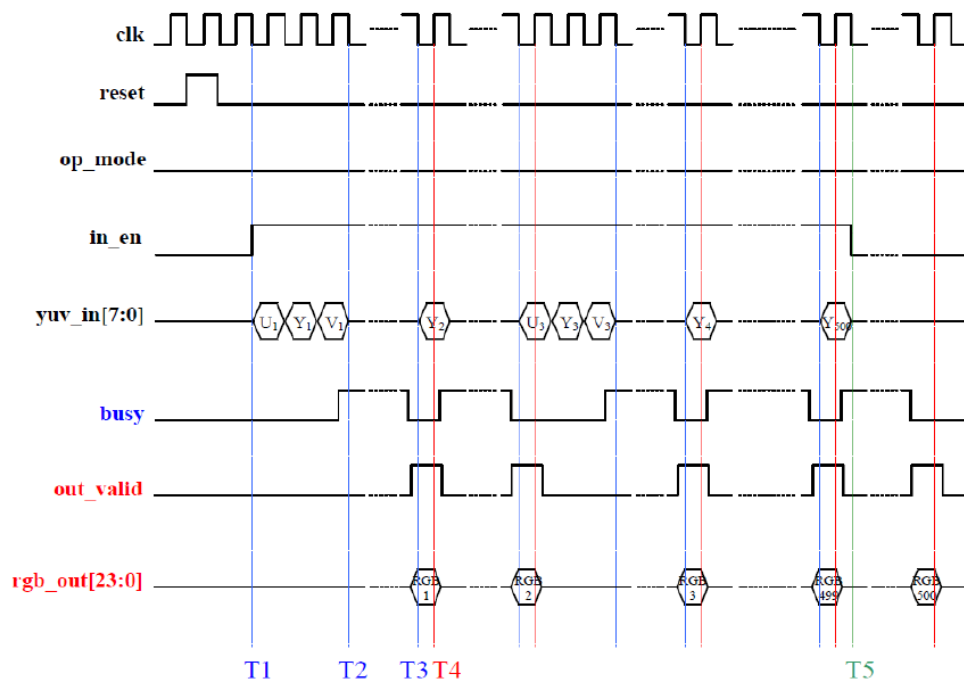
in_en	input	1	資料輸入致能控制訊號。當 Host 端有 YUV 或 RGB 訊號要輸入時，該訊號就會一直維持在 High，直到輸入訊號全部輸入完畢，該訊號才會為 Low。
Busy	Output	1	CTE 忙碌之控制訊號。當為 High 時，表示系統正處於忙碌階段，告知 Host 端，暫時停止 YUV 或 RGB 訊號的輸入；反之，當為 Low 時，表示告知 Host 端可繼續輸入 YUV 或 RGB 訊號。
yuv_in	input	8	YUV 三種訊號個別輸入的資料匯流排。YUV 輸入訊號都是 8bits，三訊號採個別輸入。只有當 in_en 為 High, busy 為 Low 時，輸入的資料才是有效的。
rgb_in	input	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	output	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	output	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	output	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 File Description

File Name	Description
CTE.v	RTL code for using Verilog
testfixture1.v	Test bench for verifying function 1
pattern_yuv.dat	Input pattern for function 1
golden_rgb.data	Golden output pattern for function 1
testfixture2.v	Test bench for verifying function 2
pattern_rgb1.dat	Input pattern 1 for function 2
pattern_rgb2.dat	Input pattern 2 for function 2
pattern_rgb3.dat	Input pattern 3 for function 2
golden_yuv1.data	Golden output pattern 1 for function 2
golden_yuv2.data	Golden output pattern 2 for function 2
golden_yuv3.data	Golden output pattern 3 for function 2
cycloneii_atoms.v	Simulation library for gate-level simulation

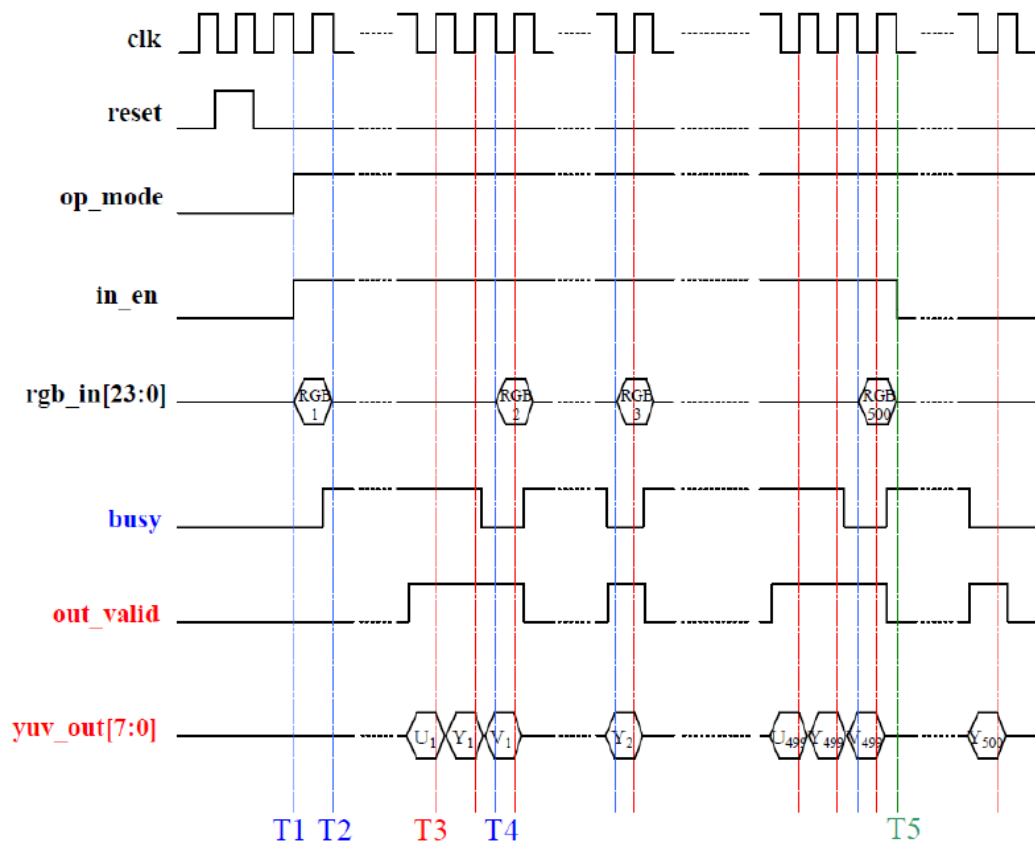
2.4 timing specification

2.4.1 YUV to RGB



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U1 訊號。
2. T2 時間點，假設同學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 RGB to YUV



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，假設同學可能需要在這點開始的幾個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
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. Scoring

1. [15%] Function1 YUV 訊號轉換成RGB 訊號，RTL模擬與正確結果 golden pattern完全一樣。
2. [15%] 完成Quartus合成，且Gate-Level Simulation 能通過function 1驗證。
3. [25%] Function2 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。Testbench 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，共區分六個等級如下：

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

A 級：error < 0.0000225pt

B 級：0.00002 ≤ error < 0.0000520pt

C 級：0.00005 ≤ error < 0.0001015pt

D 級：0.00010 ≤ error < 0.0005010pt

E 級：0.00050 ≤ error < 0.003005pt

F 級：error ≥ 0.00300 =>視 F 級為 Function2 是錯誤的，請注意!

此處共有 3 組測資，取最低者進行評分。

4. [25%]完成 Quartus 合成(Device: **Cyclone II EP2C70F896C8**)，且 Gate-Level Simulation 通過 function 2 模擬。評分標準依照第 3 項說明。

5. [20%] 完成上述要求並記錄合成後經過，以及 function2 gate-level 模擬時能通過的最小 cycle 與模擬總時間。評分標準為

*(Total logic elements + total memory bit + 9*embedded multiplier 9-bit element) × (longest gate-level simulation time in ns)*

4. Submission

4.1 Submitted files

You should classified your files into three directories and compressed to **.zip** format. The naming rule is **HW5_studentID_name_version.zip**. The vision is v1 for the first submission, and v2, v3... for the revisions.

RTL category	
*.v	All of your verilog RTL code
Gate-Level category	
*.vo	Gate-Level netlist generated by Quartus
*.sdo	SDF timing information generated byQuartus
Documentary category	
*.pdf	The report file of your design (in pdf).

4.2 Report file

Please follow the spec of report. You are asked to describe how the circuit is designed as detailed as possible.

4.3 Please submit your .zip file to folder HW5 in the ftp site.

Deadline: 2017-12-29 23:59

ftp : 140.116.245.92

Username : ic_design

Password : icdesign

5. If you have any problem, please contact the TA by email :
p78031175@mail.ncku.edu.tw