

適用於嵌入式硬體平台之色彩空間轉換器設計

夏至賢^{1*} 孔 燕² 張智貞¹

¹中國文化大學電機工程學系

^{1*}Email: chhsia625@gmail.com

²南京信息工程大学计算机与软件学院

²Email: kongyan4282@163.com

摘 要

一般在設計色彩空間轉換器(Color Space Conversion, CSC)時，因為 RGB 三原色的關係在超大型積體電路(Very Large Scale Integration, VLSI)設計時會使用較多的暫存器(Register)進而產生其硬體成本的增加。有鑑於此，本研究提出了一個管線架構(Pipelining architecture)並配合摺疊架構(Folding architecture)的設計，應用於 RGB 轉換至 $YCbCr$ 色彩空間；在改良架構中以管線處理的步驟，將其處理優化，並利用三原色非同步處理的特性配合摺疊架構成一組管線架構，使其具有即時性、高硬體使用率、以及低成本的特性。最後，本文使用 Xilinx FPGA 實現，將實際的 RGB 資料經由本系統做轉換並處理後的 $YCbCr$ 值與其理論值比較，得到較低的誤差平均值為 $Y=0.231\%$ 、 $C_b=0.221\%$ 、 $C_r=0.149\%$ 、減少暫存器使用量為 36.59% 以及減少邏輯元件的使用量為 67.57%。

關鍵詞： 影像色彩處理器、色彩空間轉換、管線架構、摺疊架構。

A Cost-Efficient VLSI Architecture for Color Space Conversion Processors

Chih-Hsien Hsia^{1*}, Yan Kong², and Chih-Chen Chang¹

¹Department of Electrical Engineering, Chinese Culture University, Taiwan

^{1*}Email: chhsia625@gmail.com

²School of Computer and Software, Nanjing University of Information Science & Technology, China

²Email: kongyan4282@163.com

ABSTRACT

During general Color Space Conversion (CSC), the Red, Green, and Blue (RGB) color model in VLSI requires many registers and thus is associated with high hardware-related costs. This study proposed a folded and pipeline architectures in which RGB was converted into a $YCbCr$ color space. The new design included optimized pipeline processing and independent processing of RGB channels and had higher instantaneity, hardware usage, and cost-efficiency. Finally, Xilinx FPGA was used to compare the actual

RGB data and the converted and processed $YCbCr$ values and theoretical values. The errors obtained were minimized to $Y=0.231\%$, $C_b=0.221\%$, and $C_r=0.149\%$ and register usage and logical element usage were decreased to 36.59% and 67.57% , respectively.

Keywords: Image Color Processor, Color space conversion; Pipeline architecture; Folded architecture.

文稿收件日期 104.12.17; 文稿修正後接受日期 105.9.10; *通訊作者
Manuscript received December 17, 2015; revised September 10, 2016; * Corresponding author

一、前言

色彩空間轉換(Color Space Conversion, CSC)廣泛應用於影像/視訊處理、壓縮、以及傳輸等技術；一般常使用的色彩空間轉換，包含 RGB、 $YCbCr$ 、YUV、HSV、以及 CMYK 等[1]。其中以 RGB 為影像擷取設備中最常用到的色彩空間，其資料量較多且不適合傳輸，故在考慮資訊傳輸與人眼對亮度較為敏感的問題下常將 RGB 轉換為 $YCbCr$ 使用[2-3]。

近年來，許多文獻針對 CSC 的架構提出改良與探討，其中 Lin *et al.* [4]提出利用移位加法(Shift-and-Add, SAA)的硬體加速器(Hardware Accelerator, HA)來實現影像 CSC，使用管線處理(Pipeline processing)來提升其硬體之效率，並且透過基因演算法(Genetic Algorithm, GA)提升轉換後的資料準確度；但其架構需使用較大量的暫存器(Register)與邏輯元件(Logic Element, LE)，在使用上成本(Cost)較高以致於不適合應用於目前較低階的嵌入式(Embedded)平台。Yang *et al.* [5]對於不同的影像 CSC 演算法做比較，分別使用 1)直接法(Direct Method, DM)、2)查表法(Look-up Table, LUT)、以及 3)硬體法(Hardware Method, HM)，比較其運算時間(Computing time)、成本、記憶體(Memory)使用量、以及誤差等因素。Sapkhal *et al.* [6]基於周邊組件連接(Peripheral Component Interconnect, PCI)系統的 CSC，其系統可支援多方面微處理器，但在傳輸資料格式轉換過程中效率較低且硬體面積較大。Jiang

et al. [7]提出快速的 RGB 轉換 $YCbCr$ 器，利用 LUT 結合管線架構，提升轉換效率；但在建立大量資料表格時，其使用較多記憶體空間。綜合以上問題，本文提出一個使用管線架構配合摺疊架構的 CSC，並改善過往文獻在硬體下常產生的暫存器、邏輯元件使用量、以及資料誤差問題作探討。最後，以 Xilinx FPGA 平台實現本文硬體電路架構。

本論文分為五個章節，在第二章將簡單的介紹色彩空間的轉換(CSC)、傳統的影像色彩處理、以及基因演算法(GA)；接著，在第三章中，說明本文所提出的管線與摺疊架構並作其分析；在第四章，則是探討硬體設計的模擬結果與比較；最後，則是在第五章做為本文之結論。

二、相關基礎知識

色彩空間種類繁多，包含 RGB、 $YCbCr$ 、YUV、CMYK 等，其中一般常用的攝影機均為 RGB 輸入，故較為廣泛應用於影像顯示。然而人類的視覺系統，對於亮度(Luminance)較為敏感，而對於色度(Chrominance)較不敏感。且 RGB 所構成的向量空間無法對影像強度(亮度)做處理，例如柔和化、銳利化等；同時由三原色所構成的影像檔案也在傳輸時佔用較大頻寬與儲存時佔用較多的記憶體。因此有必要將原來有 RGB 的格式由線性矩陣轉換成 $YCbCr$ 的組合以方便分別處理。有鑑於此，本

研究將探討 RGB 轉換至 YC_bC_r 方法上所產生的問題，其中透過色彩影像處理的 HM 結合 GA，以設計出減少暫存器與邏輯元件使用量的 CSC。

2.1 RGB 影像轉換至 YC_bC_r 色彩空間處理

RGB 分別由三原色：紅色(Red, R)、綠色(Green, G)、以及藍色(Blue, B)所組成，將三原色以不同比例結合以產生不同的顏色並廣泛應用於影像顯示，但其具有高頻寬(較多的資訊量)的缺點，在傳輸時會有較大的缺點[9]。然而， YC_bC_r 是由亮度(Y)、藍色色度(C_b)、以及紅色色度(C_r)三個成分所組成，因人眼對與亮度的敏感度高於色度且符合人眼可視資訊。幾乎目前多數的影像與視訊壓縮系統(Compression systems)等皆採此一格式所轉換[10]，由上述方程式(1)至(3)可繪製處如圖一之 RGB 轉 YC_bC_r 色彩空間轉換模型[1]，利用乘法與加法將 RGB 的值以固定的比率轉換為

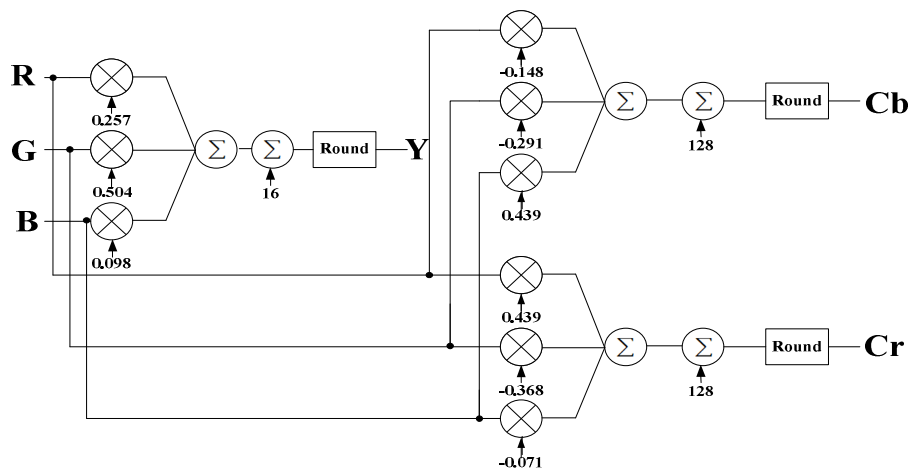
YC_bC_r ，其方程式如(1)-(3)所示：

$$Y = 0.257 \times R + 0.504 \times G + 0.098 \times B + 16 \quad (1)$$

$$C_b = -0.148 \times R - 0.291 \times G + 0.439 \times B + 128 \quad (2)$$

$$C_r = 0.439 \times R - 0.368 \times G - 0.071 \times B + 128 \quad (3)$$

傳統上，常用的 CSC 有三種方法，分別為直接法(DM)、查表法(LUT)、以及硬體法(HM)[5]。1)DM：直接做浮點乘法運算，優點是結果非常精確；缺點則是需要使用大量的記憶體或暫存器元件。2)LUT：建立一個 RGB 轉 YC_bC_r 運算結果的對照表，優點為省去運算所要花費的時間；缺點亦為需要使用較大量的記憶體。3)HM：此方法使用運算放大器和電阻網路，亦利用多個特殊應用積體電路(Application Specific Integrated Circuit, ASIC)來實現 RGB 轉 YC_bC_r 。其優點為大幅的提升處理速度，因為僅需要幾個時脈(Clock)便可完成，並且記憶體和暫存器使用量較少；缺點則是相較於軟體，因為硬體增加故成本亦大幅提升。



圖一、RGB 轉 YC_bC_r 色彩空間轉換模型[1]

2.2 基因演算法

基因演算法(GA)是依據人類在生物學，中適者生存、不適者淘汰的觀念所發展出的一種啟發式演算法，利用選擇(Selection)、交配(Crossover)、以及突變(Mutation)等步驟去尋找最佳化(Optimization)的解。對於任何一個問題，

通常會有一些相對應的條件用來評估其優劣；然而，這些條件在 GA 中稱之為適應函數值(Fitness value, F)[11]。使用者可以依據應用自訂其交配率、突變率、疊代數、以及資料筆數，並且透過交配率與突變率來決定有多少筆資料要做交配或突變，所產生出新的資料會經過

式(4)算出適應函數值，當適應函數值越高，此資料就有越高的機率會被選擇保留下來。經過多次的交配、突變、以及選擇後，系統會得到一最佳解。Lin *et al.* [4]提出一組最佳的SAA公式，如式(5)-(7)；其中資料量為80筆，疊代數為200次，交配率為70%，突變率為10%。

$$F(R, G, B) = (\sum_{i=1}^N |f_{i,desired} - f_{i,obtained}|)^{-1} \quad (4)$$

其中 $f_{i,desired}$ 為資料之理論值； $f_{i,obtained}$ 為資料之實際值

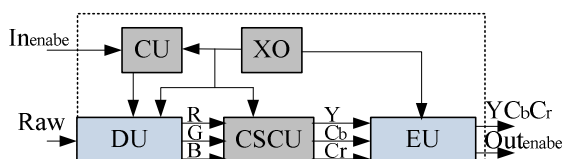
$$Y = \{ [R + (R \ll 1) + (R \ll 4) + (R \ll 9) + G + (G \ll 1) + (G \ll 2) + (G \ll 3) + (G \ll 4) + (G \ll 5) + (G \ll 6) + (G \ll 7) + (G \ll 8) + (G \ll 9) + B + (B \ll 1) + (B \ll 2) + (B \ll 3) + (B \ll 6) + (B \ll 7)] \gg 11 \} + 16 \quad (5)$$

$$C_b = \{ [-R - (R \ll 3) - (R \ll 5) - (R \ll 8) - G - (G \ll 5) - (G \ll 6) - (G \ll 9) + B + (B \ll 1) + (B \ll 2) + (B \ll 3) + (B \ll 7) + (B \ll 8) + (B \ll 9)] \gg 11 \} + 128 \quad (6)$$

$$C_r = \{ [(R \ll 1) + (R \ll 2) + (R \ll 7) + (R \ll 8) + (R \ll 9) - (G \ll 4) - (G \ll 5) - (G \ll 6) - (G \ll 7) - (G \ll 9) - (B \ll 4) - (B \ll 7)] \gg 11 \} + 128 \quad (7)$$

三、系統架構與流程

本研究使用電場可程式邏輯閘陣列(Field Programmable Gate Array, FPGA)作為硬體平台的開發與驗證，硬體架構與輸入輸出，如圖二所示。其中包含控制模組(Controller Unit, CU)、色彩空間轉換模組(Color Space Conversion Unit, CSCU)、以及40MHz的石英震盪器(Crystal Oscillator, XO)提供本系統之工作時脈。



圖二、硬體模組架構

3.1 硬體架構

本文之硬體架構分別為：解碼器單元(Decoder Unit, DU)、色彩空間轉換器單元(Color Space Converter Unit, CSCU)、控制單元(Control Unit, CU)、石英震盪器(Crystal Oscillator, XO)，本系統有兩個輸入分別為：Inenable、Raw，有兩個輸出分別為：Outenable、YCbCr，當 CU 接收到 Inenable 的輸入旗標為 1 時，即告知 DU 做解碼的工作，DU 將會把輸入的 R、G、B 訊號解碼送至 CSCU，當 CSCU 接收到 DU 的解碼資訊時，即開始做 RGB 轉換為 YCbCr 色彩空間運算，並且將 Outenable 旗標舉為 '1'，直到輸入的資訊皆計算完畢後，CSCU 會將 Outenable 旗標降為 '0'，告知外部系統已轉換完成，並將轉換完成的 Y、Cb、Cr 的資料送至 EU 做編碼的動作，EU 將會編碼為外部系統能解碼的資訊，作為本系統之輸出訊號。綜合上述流程，當外部資料輸入時，Inenable 旗標舉為 1 告知本系統開始做 RGB 轉換為 YCbCr 色彩空間，並且轉換完成後將 Outenable 旗標舉為 1，方便與外部系統溝通。

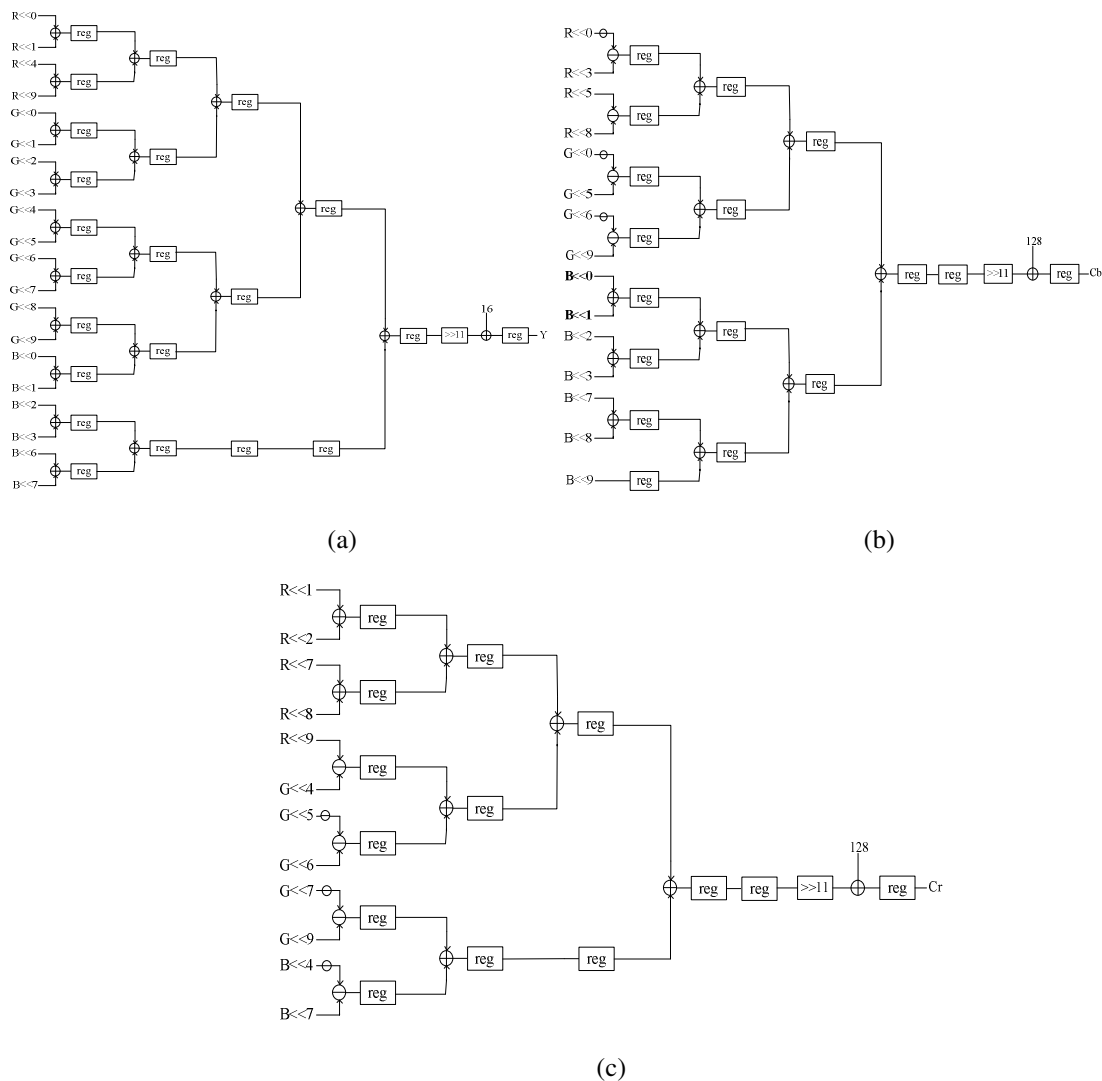
3.2 管線與摺疊架構

經由上述(5)-(7)式可繪製出 CSC 架構如圖三所示；此架構共使用了六個階層(Stage)，輸入 R、G、B 三資訊，並同時進行 Y、Cb、Cr 的移位狀態，其優點為同時進行能縮短轉換所需的時間；但缺點為使用暫存器的數量較為龐大。

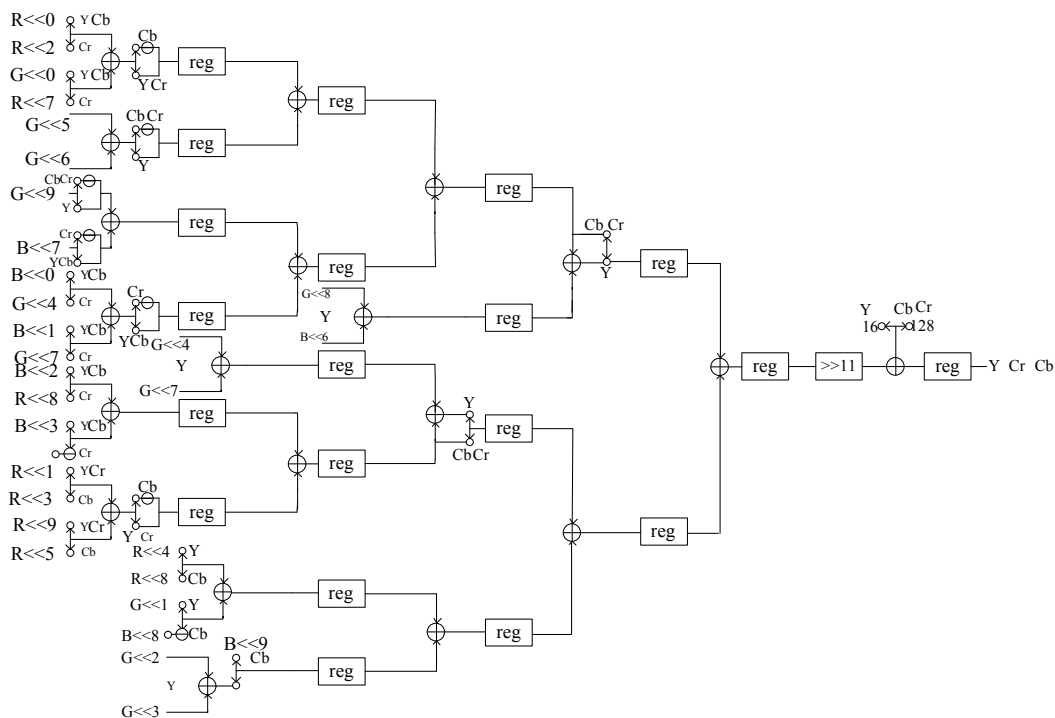
有鑑於此，本文利用摺疊架構(Folding architecture)的方式，找出 Y、Cb、Cr 在轉換時會共同使用到的移位結果做調整與共用，並利用切換的方式依次進行 Y、Cb、Cr 的色彩空間轉換，經由多一個階層的設計，使得轉換所需的暫存器使用量得以減少。另外，亦提出管線架構(Pipeline architecture)設計出 CSC，利用上

述(5)-(7) 式結合 SSA，可以設計出管線處理架構，如圖四所示。本文的架構雖使用了七個階層，輸入 R、G、B 資訊，第一個階層做亮度(Y)的移位狀態，接下來在第二個階層時切換為 C_b 的移位狀態，當來到第三個階層時切

換為 C_r 的移位狀態，第四個階層時會再切換為 Y 的移位狀態做下一組三原色的輸入轉換，以此類推；在完成後分別得到輸出 Y、 C_b 、 C_r 三個色度的平面資訊。



圖三、RGB 轉換 $YCbCr$ 之 VLSI 架構[4]：(a) Y 硬體架構、(b) C_b 硬體架構、(c) C_r 硬體架構



圖四、提出管線與摺疊之 RGB 轉換 $YCbCr$ 硬體架構

四、系統實測與結果

本研究提出一個硬體成本為基礎的 VLSI 架構，利用 Verilog 硬體描述語言開發並配合 Xilinx ISE 編譯器做功能模擬。模擬時輸入與輸出皆為 8 位元，且為了提升訊號的精確度，在設計上使用 24 位元的暫存器供 SSA 所使用。最後，在 Xilinx FPGA 晶片為 Spartan3A 系列的 XC3S200AN 作為測試，如圖八所示，其平台系統時脈為 40MHz。並比較管線架構所產生的暫存器使用量，將彩色輸入影像的 RGB 資料轉換為 $YCbCr$ 做系統數值的誤差統計，如圖五所示。其中以方程式(1)至(3)帶入測試影像之 RGB 數值計算之結果與本研究架構計算出結果做誤差比較，以其中 128 筆資料做誤差比較統計，得出較低的誤差值為 $Y=0.231\%$ 、 $C_b=0.221\%$ 、 $C_r=0.149\%$ 。

表一中，比較圖三中硬體架構與文獻[4]，其架構暫存器使用量以及邏輯元件使用量，在非管線架構(Non-pipelined) CSC HA 的暫存器

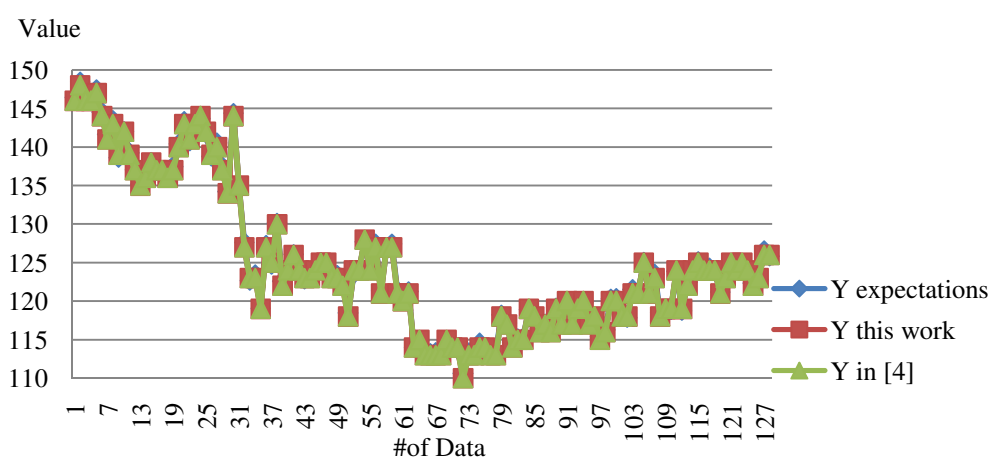
使用量為 464 個、邏輯元件總數為 1317 個；然而，在管線架構(Pipelined) CSC HA 的暫存器使用量為 940 個、邏輯元件總數為 1243 個。圖四中，本文所提出的混合式架構(管線配合摺疊)的暫存器使用量可以減少為 344 個、且邏輯元件總數為 841 個，皆少於前述兩者架構。雖然與文獻[4]作比較，本文之管線架構雖然多增加一個階層，但利用 Y、 C_b 、 C_r 三個平面時序上可以非同步的特性，以摺疊架構結合為一以節省計憶元件使得暫存器使用量減少為 344 個，並改善文獻[4]的硬體使用率至 100%，以及邏輯元件使用量減少為 841 個。比較圖四中硬體架構與文獻[4]架構的臨界路徑(Critical path)以及管線的層數，其[4]中 Pipelined CSC HA 的最長路徑為 6 個加法器，管線層數數量為 6，與本文所提出的處理架構的最長路徑為 6Ta 均相同。

以一張完整 512×512 的 Lena 影像，經過本研究提出之平台運算後的結果，運用 OpenCV[12]顯示出 Y、 C_b 、 C_r 之影像結果如

圖六所示。在使用 Xilinx ISE 編譯器做功能模擬，發現當 In_{enable} 旗標舉起時觀察本系統之轉換結果如圖七所示，如同 3.1 章節描述，外部系統將 R、G、B 資訊輸入並且將 In_{enable} 旗標舉起。本系統做 RGB 轉換為 YC_bC_r 運算，當計算完成時舉起 Out_{enable} 旗標並可以將 $Y \cdot C_b$ 、 C_r 之結果輸出。

表一、暫存器與邏輯元件使用量

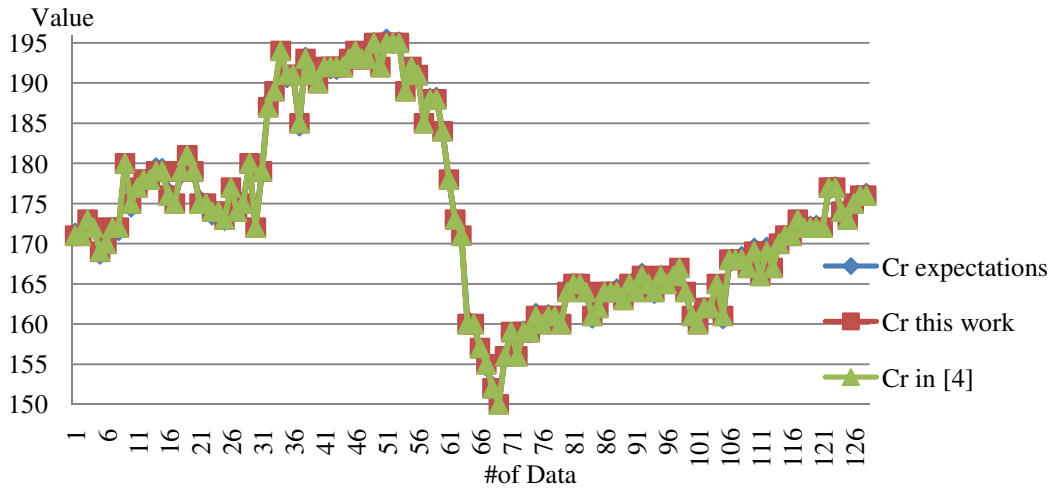
| | Non-pipelined CSC HA [4] | Pipelined CSC HA [4] | This work |
|-----------------------|-----------------------------|-------------------------|--------------|
| # of registers | 464 | 940 | 344 |
| # of logic element | 1317 | 1243 | 840 |
| Critical path | 6 Ta | 6 Ta | 6 Ta |



(a)

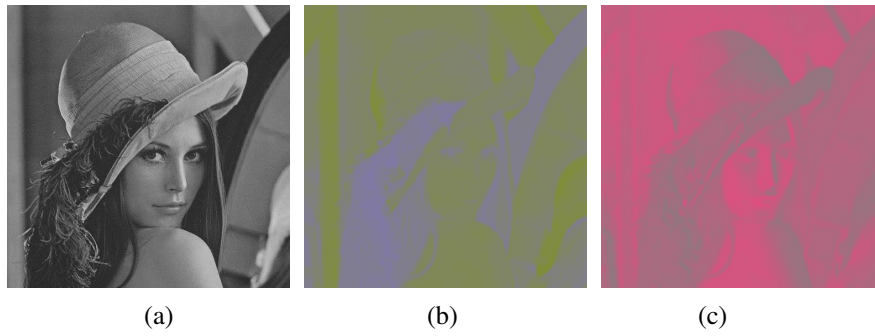


(b)



(c)

圖五、資料誤差分析圖(a)Y、(b)Cb、(c)Cr

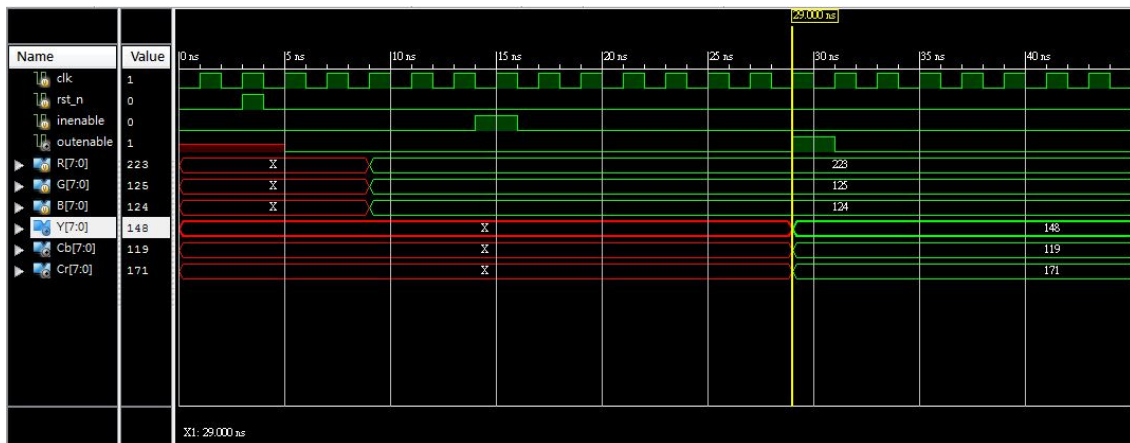


(a)

(b)

(c)

圖六、色彩空間轉換結果圖(a)Y、(b)Cb、(c)Cr



圖七、色彩空間轉換模擬



圖八、實際燒入示意圖

五、結論

本研究提出一改良混合式 VLSI 架構，應用於 RGB 轉換至 $YCbCr$ 色彩空間；在改良架構中以管線處理的步驟，將其處理優化，並利用三原色非同步處理的特性配合摺疊架構成一組管線架構，以減少一般在管線架構的暫存器與邏輯元件使用量，透過摺疊 Y 、 C_b 、 C_r 三個 VLSI 架構，使得硬體使用的元件大幅減少，並透過不同的解碼器與編碼器可適用到不同的外部系統，讓本研究的價值提升。在實驗結果得知，本文得到較低的誤差值為 $Y=0.231\%$ 、 $C_b=0.221\%$ 、 $C_r=0.149\%$ 、減少暫存器使用量為 36.59% 以及減少邏輯元件的使用量為 67.57% 。

參考文獻

- [1] Gonzalez, R. C. and Woods, R. E., *Digital Image Processing*, Firth Edition, Prentice Hall Inc, 2002.
- [2] M. Sima, S. Vassiliadis, S. Cotofana, and J. T. J. van Eijndhoven, "Color space conversion for MPEG decoding on FPGA-augmented TriMedia processor," *IEEE International Conference on Application-Specific Systems, Architectures, and Processors*, pp. 250-259, June 2003.
- [3] W. Wang, and J. S. Shi, "Color transfer processing based on $YCbCr$ color space," *Proc. SPIE- Multispectral Image Processing and Pattern Recognition*, Vol. 8917, October 2013.
- [4] S.-A. Li, C.-Y. Chen, and, C.-H. Chen, "Design of a shift-and-add based hardware accelerator for color space conversion," *Journal of Real-Time Image Processing*, vol. 10, no. 2, pp. 193-206, June 2015.
- [5] Y. Yang, Y. Peng, and Z. Liu, "A fast algorithm for $YCbCr$ to RGB conversion," *IEEE Transactions on Consumer Electronics*, vol. 53, no. 4, pp. 1490-1493, November 2007.
- [6] A. M. Sapkal, M. Munot, Joshi M. A., "R'G'B' to $YCbCr$ color space conversion using FPGA," *IET International Conference on Wireless, Mobile and Multimedia Networks*, pp. 255-258, January 2008.
- [7] H. Jiang, H. Li, T. Liu, P. Zhang, and J. Lu, "A fast method for RGB to $YCrCb$ conversion based on FPGA," *IEEE International Conference on Computer Science and Network Technology*, pp. 588-591, December 2013.
- [8] S. Yerva, S. Nair, and K. Kutty, "Lossless image compression based on data folding," *IEEE International Conference on Recent Trends in Information Technology*, pp. 999-1004, June 2011.

- [9] Primary color [Online]. Available:
https://en.wikipedia.org/wiki/Primary_color
- [10] YCbCr [Online]. Available:
<https://en.wikipedia.org/wiki/YCbCr>.
- [11] M. Melanie, *An Introduction to Genetic Algorithms*, First Edition, Prentice Hall Inc, 2002.
- [12] OpenCV. Open Source Computer Vision [Online]. Available: <http://opencv.org>.