

應用於 JPEG2000 之上提式離散小波轉換電路

陳慶鴻

國立台灣海洋大學
電機工程學系

E-mail:cloud@ds.ee.ntou.edu.tw

呂紹偉

國立台灣海洋大學
電機工程學系

E-mail:swleu@ntou.net

李政宏

國立台灣科技大學
電子工程學系

E-mail:D9102102@mail.ntust.edu.tw

摘要

本論文針對應用於 JPEG2000 之上提式離散小波轉換提出一管線化的電路架構。在此一架構中，我們將 JPEG2000 使用的兩種濾波器加以整合。這兩種濾波器分別為應用於無失真壓縮的 5/3 濾波器和應用於有失真壓縮的 9/7 濾波器。我們所提出的一維架構具有精簡且彈性的特點，但是需要對外部的記憶體作大量的存取。針對這個缺點我們提出了一個以一維架構為基礎而延伸出來的二維架構，經由將輸入訊號的順序做最佳化的安排，可以有效減少記憶體的使用量。相對於一維架構，二維架構可以減少一半的外部記憶體存取，完成轉換所需的時間也減少將近一半。在記憶體的使用量方面，當影像的解析度為 $N \times N$ 時，我們所提出的二維架構只需 $9N$ 的儲存單元，且由於整合 5/3 了和 9/7 濾波器，記憶體使用量也較個別實現二種濾波器時所需的 $14N$ 為少。

關鍵字：上提式離散小波轉換、JPEG2000、影像處理。

一、簡介

近年來電腦科技的發展日新月異，各種聲音、影像、圖片等多媒體資訊的應用與技術也不斷的推陳出新，隨著網路傳輸以及資料備份儲存的需求日益殷切，資料壓縮的技術越顯得重要且不可避免。經過壓縮的資料不但可以節省儲存的空間及網路傳輸的時間，還可以增加資料的安全性。

自 90 年代起，小波轉換受到廣泛的討論及應用，如數值分析、信號分析、統計學、生物醫學、影像辨識及影像編碼等各種領域。由於小波轉換具有多重解析 (multiresolution) 的特性，使其特別適合應用在影像的分析和壓縮 [1]。在硬體實現方面，雖然曾有許多架構被提出 [2]-[7]，但大都過於複雜。Sweldens [8]-[10] 在 1996 年提出了一種新的小波轉換架構，即

上提式小波轉換 (lifting-based discrete wavelet transform)。此種轉換方式所需的硬體資源較先前的方式少，且完成計算所需的時間也較短，因此已被納入國際標準組織的 JPEG2000 標準 [11]。

在本論文中，我們針對應用於 JPEG2000 中之上提式離散小波轉換，先提出一管線化之一維轉換架構，然後利用此架構為基礎發展二維架構。我們所提出的二維架構能有效的減少記憶體的使用量；以一個解析度為 $N \times N$ 的影像為例，執行 JPEG2000 之有失真 9/7 濾波器運算只需要 $9N$ 的記憶體。

二、上提式小波轉換

上提式小波轉換的組成架構，主要分為四個部分：splitting、predicting、updating 和 scaling。圖 1 為上提式小波轉換之示意圖，以下為各部分的簡要說明。

- (1) Splitting：將輸入訊號 $x[n]$ 分離成兩個部分，即 $x[n]$ 的奇數部分 $x_o[n]$ 和偶數部分 $x_e[n]$ ，其中 $x_o = x[2n+1]$ ， $x_e = x[2n]$ 。
- (2) Predicting(P)： $x_e[n]$ 訊號經過 predicting 模組產生一預測值，然後再求此一預測值和 $x_o[n]$ 之誤差而得到高頻小波係數 $d^1[n]$ ，以數學式表示為 $d^1[n] = x_o[n] - P(x_e[n])$ 。
- (3) Updating(U)： $d^1[n]$ 經由 updating 模組然後和 $x_e[n]$ 相加產生一個低頻小波訊號 $s^1[n]$ ，以數學式表示為 $s^1[n] = x_e + U(d^1[n])$ 。

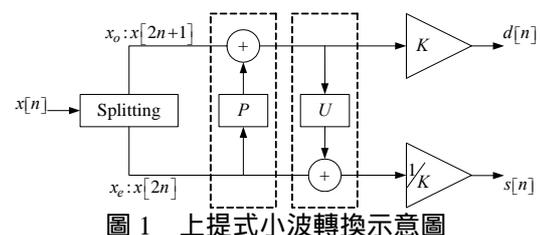


圖 1 上提式小波轉換示意圖

- (4) $\text{Scaling}(K, 1/K)$: 將所得的高頻小波係數和低頻小波係數分別乘以 K 和 $1/K$, 也就是將這兩個係數正規化, 即 $s[n] = (1/K) \times s^1[n]$ 和 $d[n] = K \times d^1[n]$ 。

JPEG2000 標準使用兩種轉換, 即 $5/3$ 濾波器和 $9/7$ 濾波器。 $5/3$ 濾波器稱為整數小波, 它將影像的整數灰階值轉換成整數小波係數, 主要是應用在無失真資料壓縮 (lossless compression), 又稱為可逆 (reversible) 轉換。 $9/7$ 濾波器稱為浮點小波, 它將影像的整數灰階值轉換成實數形式的小波係數, 適合應用在有失真的資料壓縮 (lossy compression), 又稱為不可逆 (irreversible) 轉換。 圖 2 為 $5/3$ 濾波器之上提式小波轉換示意圖, 其所對應之數學式如下:

$$P(z) = \alpha \times (1 + z)$$

$$U(z) = \beta \times (z^{-1} + 1)$$

其中 $\alpha = -1/2$ $\beta = 1/4$, 使得其硬體較為簡單, 只需移位運算即可實現。

圖 3 為 $9/7$ 濾波器之上提式小波轉換示意圖, 其所對應之數學式如下:

$$P1(z) = \alpha \times (1 + z)$$

$$U1(z) = \beta \times (z^{-1} + 1)$$

$$P2(z) = \gamma \times (1 + z)$$

$$U2(z) = \delta \times (z^{-1} + 1)$$

其中 $\alpha = -1.586134342$, $\beta = -0.052980118$ $\gamma = 0.882911075$, $\delta = 0.443506852$ 。

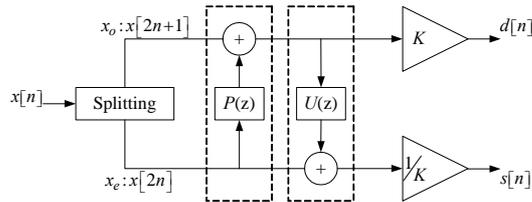


圖 2 $5/3$ 濾波器之上提式小波轉換示意圖

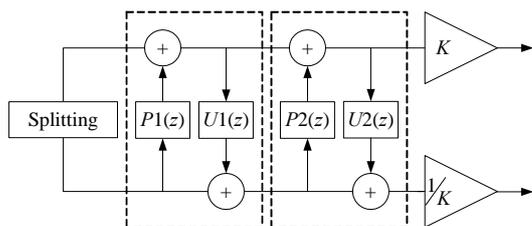


圖 3 $9/7$ 濾波器之上提式小波轉換示意圖

由於 $9/7$ 濾波器之上提式小波轉換係數皆為實數, 因此硬體實作會較為複雜。 以下各節將介紹 $5/3$ 和 $9/7$ 濾波器上提式轉換之硬體實現。

三、一維轉換

我們在本節提出一個管線化的一維上提式離散小波轉換硬體架構, 此架構將 $5/3$ 濾波器和 $9/7$ 濾波器的前段加以整合, 以節省硬體資源。 當管線填滿之後每個時脈週期可以送出一個高頻訊號和一個低頻訊號。

3-1 硬體架構

一維小波轉換的硬體架構如圖 4。 電路主要由以下的運算單元構成: Predictor、Updator、Predictor97、Updator97, 以及負責傳遞資料的 Reg 單元。 當進行無失真 $5/3$ 濾波器運算時只需上半部的電路, 而進行有失真 $9/7$ 濾波器運算則使用全部的電路。 Odd 和 Even 為電路的輸出埠, 分別為高頻訊號 (odd coefficients) 和低頻訊號 (even coefficients)。

3-2 運算單元設計

我們使用管線運算 (Pipelining) 的方式來實作每一個運算單元。 $5/3$ 濾波器需要兩個加法器和一個移位器; $9/7$ 濾波器需要兩個加法器和一個乘法器。 我們讓二種濾波器共用加法器並且將移位器和乘法器整合, 使電路簡化。 如圖 5, 當進行 $5/3$ 運算時選用移位器, $9/7$ 運算時選用乘法器。 因為 Predictor97 和 Updator97 模組是進行 $9/7$ 運算時才會使用到, 所以沒有移位器。 暫存器的資料寬度定為十六位元。 $9/7$ 濾波器的係數皆為實數, 以二補數表示如表 1。

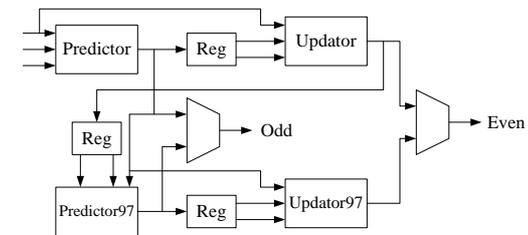


圖 4 一維小波轉換架構

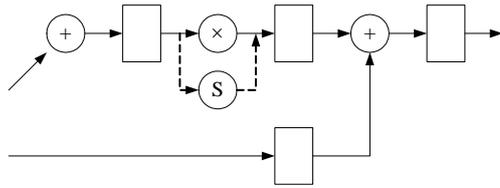


圖 5 運算單元設計

表 1 9/7 濾波器係數之二補數表示

	value	16-bit 2's complement
α	1.586134342	$2^0 + 2^{-1} + 2^{-4} + 2^{-6} + 2^{-7} + 2^{-13} + 2^{-14}$
β	0.052980118	$2^{-5} + 2^{-6} + 2^{-8} + 2^{-9} + 2^{-12}$
γ	0.882911075	$2^{-1} + 2^{-2} + 2^{-3} + 2^{-7} + 2^{-14} + 2^{-15}$
δ	0.443506852	$2^{-2} + 2^{-3} + 2^{-4} + 2^{-8} + 2^{-9} + 2^{-13}$

3-3 運算方式

我們以一個 9×9 的影像為例進行上提式離散小波轉換，圖 6 為 $5/3$ 濾波器的運算方式。0~9 的方格代表輸入的影像之一列或一行像素 (pixel)，-1、-2、9、10 則為 periodic symmetric extended signals，這些擴展訊號在圖中以線段和原先的輸入訊號相連。圖中的圓圈為基本運算單元。

圖 7 為 $9/7$ 濾波器的運算方式。因為 $9/7$ 濾波器的濾波器係數較多，所以以上提式演算法實現需要四階，也就是兩次 predict 和兩次 update。在 signal extension 方面也需要較多的係數擴展。從圖 6 和圖 7 可以很明顯看出，Predictor 模組是以兩個偶數位置的訊號和一個奇數位置的訊號為輸入，經運算後可以得到一奇數位置的輸出。Updater 模組的輸出是經由兩個 Predictor 模組的奇數輸出和一個偶數位置的輸入得到。接著我們以時序的觀點來瞭解運算的過程。

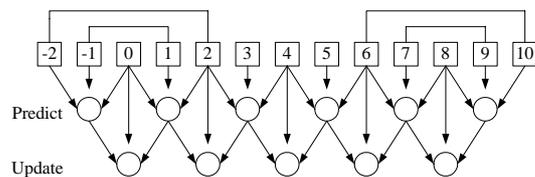


圖 6 5/3 濾波器運算

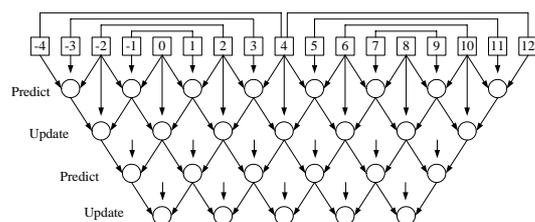


圖 7 9/7 濾波器運算

表 2 5/3 濾波器部分時序

Time	Predictor			Updater		
	Adder1	S/M	Adder2	Adder1	S/M	Adder2
1						
2	$X_{0,0} X_{0,2}$					
3	$X_{0,2} X_{0,4}$	RA1				
4	$X_{0,4} X_{0,6}$	RA1	RS $X_{0,1}$			
5	$X_{0,6} X_{0,8}$	RA1	RS $X_{0,3}$			
6		RA1	RS $X_{0,5}$	$Y_{0,1} Y_{0,3}$		
7			RS $X_{0,7}$	$Y_{0,3} Y_{0,5}$	RA1	$Y_{0,0}$
8				$Y_{0,5} Y_{0,7}$	RA1	RS $X_{0,2}$
9					RA1	RS $X_{0,4}$
10						RS $X_{0,6}$
11						$Y_{0,8}$

3-4 運算時序

表 2 為一列像素進行 $5/3$ 濾波器運算時資料的時序，其中 X 代表輸入的信號，Y 代表輸出，X、Y 的下標代表該像素在影像中的位置。以一個 9×9 的影像為例，若輸入的訊號是該影像第零列的像素，則在第 4 到第 7 個週期可以得到高頻訊號 $Y_{0,1}$ 、 $Y_{0,3}$ 、 $Y_{0,5}$ 及 $Y_{0,7}$ ；第 7 到 11 週期可以得到低頻訊號 $Y_{0,0}$ 、 $Y_{0,2}$ 、 $Y_{0,4}$ 、 $Y_{0,6}$ 及 $Y_{0,8}$ 。如此週而復始完成所有的列運算之後，再進行行運算，即可完成一次二維離散小波轉換。

同樣由表 3 可以瞭解一列像素進行 $9/7$ 濾波器運算時資料的時序。因為 $9/7$ 濾波器的運算電路比 $5/3$ 濾波器大了一倍，所以 $9/7$ 濾波器的管線階數也較多。表 3 中 X 代表輸入的信號，Z 代表輸出，X、Z 的下標代表該像素在影像中的位置。在第 11 到第 14 週期可以得到高頻訊號 $Z_{0,1}$ 、 $Z_{0,3}$ 、 $Z_{0,5}$ 及 $Z_{0,7}$ ；第 14 到第 18 個週期可以得到低頻訊號 $Z_{0,0}$ 、 $Z_{0,2}$ 、 $Z_{0,4}$ 、 $Z_{0,6}$ 及 $Z_{0,8}$ 。

3-5 效能分析

當管線填滿之後每個時脈週期可以送出一個高頻訊號和一個低頻訊號。對一列長度為 N 的訊號，執行 $5/3$ 濾波器運算需要 $(N/2) + 5$ 個週期，執行 $9/7$ 濾波器運算需要 $(N/2) + 12$ 個週期。一個 $N \times N$ 的影像作一次轉換， $5/3$ 濾波器需要 $N^2 + 5$ 個週期， $9/7$ 濾波器則需要 $N^2 + 12$ 個週期。

表 3 9/7 濾波器部分時序

Time	Predictor			Updator			Predictor97			Updator97		
	Adder1	α	Adder2	Adder1	β	Adder2	Adder1	γ	Adder2	Adder1	δ	Adder2
1	-	-	-	-	-	-	-	-	-	-	-	-
2	X ₀₀ X ₀₂	-	-	-	-	-	-	-	-	-	-	-
3	X ₀₂ X ₀₄	RA1	-	-	-	-	-	-	-	-	-	-
4	X ₀₄ X ₀₆	RA1	RM X ₀₁	-	-	-	-	-	-	-	-	-
5	X ₀₆ X ₀₈	RA1	RM X ₀₃	Y ₀₁ Y ₀₁	-	-	-	-	-	-	-	-
6	-	RA1	RM X ₀₅	Y ₀₁ Y ₀₃	RA1	-	-	-	-	-	-	-
7	-	-	RM X ₀₇	Y ₀₃ Y ₀₅	RA1	Y ₀₀	-	-	-	-	-	-
8	-	-	-	Y ₀₅ Y ₀₇	RA1	RM X ₀₂	-	-	-	-	-	-
9	-	-	-	Y ₀₇ Y ₀₇	RA1	RM X ₀₄	Y ₀₀ Y ₀₂	-	-	-	-	-
10	-	-	-	-	RA1	RM X ₀₆	Y ₀₂ Y ₀₄	RA1	-	-	-	-
11	-	-	-	-	-	Y ₀₈	Y ₀₄ Y ₀₆	RA1	RM Y ₀₁	-	-	-
12	-	-	-	-	-	-	Y ₀₆ Y ₀₈	RA1	RM Y ₀₃	Z ₀₁ Z ₀₁	-	-
13	-	-	-	-	-	-	-	RA1	RM Y ₀₅	Z ₀₁ Z ₀₃	RA1	-
14	-	-	-	-	-	-	-	-	RM Y ₀₇	Z ₀₃ Z ₀₅	RA1	Z ₀₀
15	-	-	-	-	-	-	-	-	-	Z ₀₅ Z ₀₇	RA1	RM Y ₀₂
16	-	-	-	-	-	-	-	-	-	Z ₀₇ Z ₀₇	RA1	RM Y ₀₄
17	-	-	-	-	-	-	-	-	-	-	RA1	RM Y ₀₆
18	-	-	-	-	-	-	-	-	-	-	-	Z ₀₈

四、二維轉換

4-1 硬體架構

為了減少對外部記憶體存取資料的次數，我們提出一種二維轉換的方式：首先進行列運算，然後將結果儲存到記憶體中，當儲存足夠的列運算值後（三列），即可開始做列之行運算（column processing along row）。此二維小波轉換的架構如圖 8，其中 Memory1、Memory2 及 Temp 皆為記憶體模組。此架構可以說是一維架構的延伸，基本運算單元都是相同的。藉由將一維模組運算過的結果儲存到內部的記憶體，我們可以減少外部記憶體的存取次數。

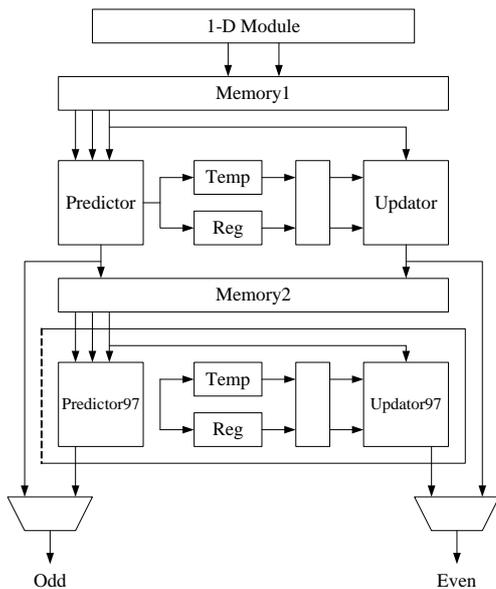


圖 8 二維小波轉換架構

表 4 5/3 濾波器之二維轉換部分時序(1)

Time	Predictor			Updator		
	Adder1	S/M	Adder2	Adder1	S/M	Adder2
1						
2	X _{0,0} X _{0,2}					
3	X _{0,2} X _{0,4}	RA1				
4	X _{0,4} X _{0,6}	RA1	RS X _{0,1}			
5	X _{0,6} X _{0,8}	RA1	RS X _{0,3}			
6	-	RA1	RS X _{0,5}	Y _{0,1} Y _{0,3}		
7	X _{2,0} X _{2,2}	-	RS X _{0,7}	Y _{0,3} Y _{0,5}	RA1	Y _{0,0}
8	X _{2,0} X _{2,2}	RA1	-	Y _{0,5} Y _{0,7}	RA1	RS X _{0,2}
9	X _{2,0} X _{2,2}	RA1	RS X _{2,1}	-	RA1	RS X _{0,4}
10	X _{2,0} X _{2,2}	RA1	RS X _{2,3}	-	-	RS X _{0,6}
11	-	RA1	RS X _{2,5}	Y _{2,1} Y _{2,3}	-	Y _{0,8}
12	X _{1,0} X _{1,2}	-	RS X _{2,7}	Y _{2,3} Y _{2,5}	RA1	Y _{2,0}
13	X _{1,2} X _{1,4}	RA1	-	Y _{2,5} Y _{2,7}	RA1	RS X _{2,2}
14	X _{1,4} X _{1,6}	RA1	RS X _{1,1}	-	RA1	RS X _{2,4}
15	X _{1,6} X _{2,8}	RA1	RS X _{1,3}	-	-	RS X _{2,6}
16	-	RA1	RS X _{1,5}	Y _{1,1} Y _{1,3}	-	Y _{2,8}
17	X _{4,0} X _{4,2}	-	RS X _{1,7}	Y _{1,3} Y _{1,5}	RA1	Y _{1,0}
18	X _{4,0} X _{4,2}	RA1	-	Y _{1,5} Y _{1,7}	RA1	RS X _{1,2}

表 5 5/3 濾波器之二維轉換部分時序(2)

Time	Predictor			Time	Predictor		
	Adder1	S/M	Adder2		Adder1	S/M	Adder2
	-	-	-	18	Z _{1,1} Z _{1,1}	-	-
15	Y _{0,1} Y _{2,1}	-	-	19	Z _{1,3} Z _{1,3}	RA1	-
16	Y _{0,3} Y _{2,3}	RA1	-	20	Z _{1,5} Z _{1,5}	RA1	RS Y _{0,1}
17	Y _{0,5} Y _{2,5}	RA1	RS Y _{1,1}	21	Z _{1,7} Z _{1,7}	RA1	RS Y _{0,3}
18	Y _{0,7} Y _{2,7}	RA1	RS Y _{1,3}	22	Z _{1,0} Z _{1,0}	RA1	RS Y _{0,5}
19	Y _{0,0} Y _{2,0}	RA1	RS Y _{1,5}	23	Z _{1,2} Z _{1,2}	RA1	RS Y _{0,7}
20	Y _{0,2} Y _{2,2}	RA1	RS Y _{1,7}	24	Z _{1,4} Z _{1,4}	RA1	RS Y _{0,0}
21	Y _{0,4} Y _{2,4}	RA1	RS Y _{1,0}	25	Z _{1,6} Z _{1,6}	RA1	RS Y _{0,2}
22	Y _{0,6} Y _{2,6}	RA1	RS Y _{1,2}	26	Z _{1,8} Z _{1,8}	RA1	RS Y _{0,4}
23	Y _{0,8} Y _{2,8}	RA1	RS Y _{1,4}	27	Z _{1,1} Z _{3,1}	RA1	RS Y _{0,6}
24	Y _{2,1} Y _{4,1}	RA1	RS Y _{1,6}	28	Z _{1,3} Z _{3,3}	RA1	RS Y _{0,8}
25	Y _{2,3} Y _{4,3}	RA1	RS Y _{1,8}	29	Z _{1,5} Z _{3,5}	RA1	RS Y _{2,1}
26	Y _{2,5} Y _{4,5}	RA1	RS Y _{3,1}	30	Z _{1,7} Z _{3,7}	RA1	RS Y _{2,3}
27	Y _{2,7} Y _{4,7}	RA1	RS Y _{3,3}	31	Z _{1,0} Z _{3,0}	RA1	RS Y _{2,5}
28	Y _{2,0} Y _{4,0}	RA1	RS Y _{3,5}	32	Z _{1,2} Z _{3,2}	RA1	RS Y _{2,7}
29	Y _{2,2} Y _{4,2}	RA1	RS Y _{3,7}	33	Z _{1,4} Z _{3,4}	RA1	RS Y _{2,0}
30	Y _{2,4} Y _{4,4}	RA1	RS Y _{3,0}	34	Z _{1,6} Z _{3,6}	RA1	RS Y _{2,2}
31	Y _{2,6} Y _{4,6}	RA1	RS Y _{3,2}	35	Z _{1,8} Z _{3,8}	RA1	RS Y _{2,4}
				36	Z _{3,1} Z _{5,3}	RA1	RS Y _{2,6}
				37	Z _{3,3} Z _{5,3}	RA1	RS Y _{2,8}

表 6 9/7 濾波器之二維轉換部分時序(1)

Time	Predictor			Updator			Predictor97			Updator97		
	Adder1	α	Adder2	Adder1	β	Adder2	Adder1	γ	Adder2	Adder1	δ	Adder2
1	-	-	-	-	-	-	-	-	-	-	-	-
2	X ₀₀ X ₀₂	-	-	-	-	-	-	-	-	-	-	-
3	X ₀₂ X ₀₄	RA1	-	-	-	-	-	-	-	-	-	-
4	X ₀₄ X ₀₆	RA1	RM X ₀₁	-	-	-	-	-	-	-	-	-
5	X ₀₆ X ₀₈	RA1	RM X ₀₃	-	-	-	-	-	-	-	-	-
6	-	RA1	RM X ₀₅	Y ₀₁ Y ₀₃	-	-	-	-	-	-	-	-
7	X ₂₀ X ₂₂	-	RM X ₀₇	Y ₀₃ Y ₀₅	RA1	Y ₀₀	-	-	-	-	-	-
8	X ₂₂ X ₂₄	RA1	-	Y ₀₅ Y ₀₇	RA1	RM X ₀₂	-	-	-	-	-	-
9	X ₂₄ X ₂₆	RA1	RM X ₂₁	-	RA1	RM X ₀₄	Y ₀₀ Y ₀₂	-	-	-	-	-
10	X ₂₆ X ₂₈	RA1	RM X ₂₃	-	-	RM X ₀₆	Y ₀₂ Y ₀₄	RA1	-	-	-	-
11	-	RA1	RM X ₂₅	Y ₂₁ Y ₂₃	-	Y ₀₈	Y ₀₄ Y ₀₆	RA1	RM Y ₀₁	-	-	-
12	X ₁₀ X ₁₂	-	RM X ₂₇	Y ₂₃ Y ₂₅	RA1	Y ₂₀	Y ₀₆ Y ₀₈	RA1	RM Y ₀₃	-	-	-
13	X ₁₂ X ₁₄	RA1	-	Y ₂₅ Y ₂₇	RA1	RM X ₂₂	-	RA1	RM Y ₀₅	Z ₀₁ Z ₀₃	-	-
14	X ₁₄ X ₁₆	RA1	RM X ₁₁	-	RA1	RM X ₂₄	Y ₂₀ Y ₂₂	-	RM Y ₀₇	Z ₀₃ Z ₀₅	RA1	Z ₀₀
15	X ₁₆ X ₁₈	RA1	RM X ₁₃	-	-	RM X ₂₆	Y ₂₂ Y ₂₄	RA1	-	Z ₀₅ Z ₀₇	RA1	RM Y ₀₂
16	-	RA1	RM X ₁₅	Y ₁₁ Y ₁₃	-	Y ₂₈	Y ₂₄ Y ₂₆	RA1	RM Y ₂₁	-	RA1	RM Y ₀₄
17	X ₄₀ X ₄₂	-	RM X ₁₇	Y ₁₃ Y ₁₅	RA1	Y ₁₀	Y ₂₆ Y ₂₈	RA1	RM Y ₂₃	-	-	RM Y ₀₆
18	X ₄₂ X ₄₄	RA1	-	Y ₁₅ Y ₁₇	RA1	RM X ₁₂	-	RA1	RM Y ₂₅	Z ₂₁ Z ₂₃	-	Z ₀₈
19	X ₄₄ X ₄₆	RA1	RM X ₄₁	-	RA1	RM X ₁₄	Y ₁₀ Y ₁₂	-	RM Y ₂₇	Z ₂₃ Z ₂₅	RA1	Z ₂₀
20	X ₄₆ X ₄₈	RA1	RM X ₄₃	-	-	RM X ₁₆	Y ₁₂ Y ₁₄	RA1	-	Z ₂₅ Z ₂₇	RA1	RM Y ₂₂
21	-	RA1	RM X ₄₅	Y ₄₁ Y ₄₃	-	Y ₁₈	Y ₁₄ Y ₁₆	RA1	RM Y ₁₁	-	RA1	RM Y ₂₄
22	X ₃₀ X ₃₂	-	RM X ₄₇	Y ₄₃ Y ₄₅	RA1	Y ₄₀	Y ₁₆ Y ₁₈	RA1	RM Y ₁₃	-	-	RM Y ₂₆
23	X ₃₂ X ₃₄	RA1	-	Y ₄₅ Y ₄₇	RA1	RM X ₄₂	-	RA1	RM Y ₁₅	Z ₁₁ Z ₁₃	-	Z ₂₈

表 7 9/7 濾波器之二維轉換部分時序(2)

Time	Predictor			Time	Updator		
	Adder1	S/M	Adder2		Adder1	S/M	Adder2
	-	-	-		-	-	-
20	Z _{0,1} Z _{2,1}	-	-	23	A _{1,1} A _{1,1}	-	-
21	Z _{0,3} Z _{2,3}	RA1	-	24	A _{1,3} A _{1,3}	RA1	-
22	Z _{0,5} Z _{2,5}	RA1	RS Z _{1,1}	25	A _{1,5} A _{1,5}	RA1	RS Z _{0,1}
23	Z _{0,7} Z _{2,7}	RA1	RS Z _{1,3}	26	A _{1,7} A _{1,7}	RA1	RS Z _{0,3}
24	Z _{0,9} Z _{2,9}	RA1	RS Z _{1,5}	27	A _{1,9} A _{1,9}	RA1	RS Z _{0,5}
25	Z _{0,2} Z _{2,2}	RA1	RS Z _{1,7}	28	A _{1,2} A _{1,2}	RA1	RS Z _{0,7}
26	Z _{0,4} Z _{2,4}	RA1	RS Z _{1,9}	29	A _{1,4} A _{1,4}	RA1	RS Z _{0,9}
27	Z _{0,6} Z _{2,6}	RA1	RS Z _{1,2}	30	A _{1,6} A _{1,6}	RA1	RS Z _{0,2}
28	Z _{0,8} Z _{2,8}	RA1	RS Z _{1,4}	31	A _{1,8} A _{1,8}	RA1	RS Z _{0,4}
29	Z _{2,1} Z _{4,1}	RA1	RS Z _{1,6}	32	A _{1,1} A _{3,1}	RA1	RS Z _{0,6}
30	Z _{2,3} Z _{4,3}	RA1	RS Z _{1,8}	33	A _{1,3} A _{3,3}	RA1	RS Z _{0,8}
31	Z _{2,5} Z _{4,5}	RA1	RS Z _{3,1}	34	A _{1,5} A _{3,5}	RA1	RS Z _{2,1}
32	Z _{2,7} Z _{4,7}	RA1	RS Z _{3,3}	35	A _{1,7} A _{3,7}	RA1	RS Z _{2,3}
33	Z _{2,9} Z _{4,9}	RA1	RS Z _{3,5}	36	A _{1,9} A _{3,9}	RA1	RS Z _{2,5}
34	Z _{2,2} Z _{4,2}	RA1	RS Z _{3,7}	37	A _{1,2} A _{3,2}	RA1	RS Z _{2,7}
35	Z _{2,4} Z _{4,4}	RA1	RS Z _{3,9}	38	A _{1,4} A _{3,4}	RA1	RS Z _{2,9}
36	Z _{2,6} Z _{4,6}	RA1	RS Z _{3,2}	39	A _{1,6} A _{3,6}	RA1	RS Z _{2,2}

4-2 運算時序

如果依一般按列進行的一維轉換運算方式，在二維架構中將會浪費許多記憶體，因此我們重定列運算的順序，以便將二維小波轉換所需的記憶體存取次數降至最低。從前面各節可知，上提式離散小波轉換中的高頻訊號是經由兩個偶係數和一個奇係數求得，低頻訊號是經由兩個奇係數和一個偶係數得到；所以我們維轉換列運算的部分時序，如同一維架構運算的方式，將像素列依特定順序送入 1-D Module

中。表中 X 代表輸入像素值，Y 代表列運算的輸出值，Y 被儲存到 Memory1 中。由表 4 可知輸出的列順序為 0 2 1 4 3...。當 Memory1 中依序儲存了第零列、第二列、第一列的列運算結果後，即可進行行運算。表 5 為 5/3 濾波器行運算的時序，其中 Y 為運算單元 Predictor 的輸入訊號，其輸出訊號 Z 作為運算單元 Updator 的輸入。最後我們在 Predictor 單元和 Updator 單元的輸出得到結果 Z。

表 6 為一個 9×9 的影像進行 9/7 濾波器二維轉換列運算的部分時序，其中 X 代表輸入像素值，Z 代表列運算的輸出值，Z 將被儲存到 Memory1 中。如同 5/3 濾波器的運算，Memory1 儲存了一定量的訊號之後即可開始進行行運算。表 7 及 8 為 9/7 濾波器行運算的時序。9/7 濾波器運算會用到四個運算單元；即 Predictor、Updator、Predictor97 以及 Updator97。Z 為運算單元 Predictor 的輸入訊號，其輸出訊號 A 作為運算單元 Updator 的輸入，同時 A 也必須儲存到 Memory2 中。

經由 Predictor 和 Updator 模組分別產生 1、3、5、7...列和 2、4、6、8...列的輸出。當 Memory2 中儲存了 0、2、1 列之後就可以讓 Predictor97 模組和 Updator97 模組進行運算。由表 8，A 為從 Memory2 中讀出之輸入，在 Predictor97 模組的輸出端可以得到奇數列的 B 輸出，而 Updator97 模組的輸出端可以得到偶數列的 B 輸出。

表 8 9/7 濾波器之二維轉換部分時序(3)

Time	Predictor97			Time	Updater97		
	Adder1	S/M	Adder2		Adder1	S/M	Adder2
35	A _{0,1} A _{2,1}	-	-	38	B _{1,1} B _{1,1}	-	-
36	A _{0,3} A _{2,3}	RA1	-	39	B _{1,3} B _{1,3}	RA1	-
37	A _{0,5} A _{2,5}	RA1	RS A _{1,1}	40	B _{1,5} B _{1,5}	RA1	RS A _{0,1}
38	A _{0,7} A _{2,7}	RA1	RS A _{1,3}	41	B _{1,7} B _{1,7}	RA1	RS A _{0,3}
39	A _{0,0} A _{2,0}	RA1	RS A _{1,5}	42	B _{1,0} B _{1,0}	RA1	RS A _{0,5}
40	A _{0,2} A _{2,2}	RA1	RS A _{1,7}	43	B _{1,2} B _{1,2}	RA1	RS A _{0,7}
41	A _{0,4} A _{2,4}	RA1	RS A _{1,0}	44	B _{1,4} B _{1,4}	RA1	RS A _{0,0}
42	A _{0,6} A _{2,6}	RA1	RS A _{1,2}	45	B _{1,6} B _{1,6}	RA1	RS A _{0,2}
43	A _{0,8} A _{2,8}	RA1	RS A _{1,4}	46	B _{1,8} B _{1,8}	RA1	RS A _{0,4}
44	A _{2,1} A _{4,1}	RA1	RS A _{1,6}	47	B _{1,1} B _{3,1}	RA1	RS A _{0,6}
45	A _{2,3} A _{4,3}	RA1	RS A _{1,8}	48	B _{1,3} B _{3,3}	RA1	RS A _{0,8}
46	A _{2,5} A _{4,5}	RA1	RS A _{3,1}	49	B _{1,5} B _{3,5}	RA1	RS A _{2,1}
47	A _{2,7} A _{4,7}	RA1	RS A _{3,3}	50	B _{1,7} B _{3,7}	RA1	RS A _{2,3}
48	A _{2,0} A _{4,0}	RA1	RS A _{3,5}	51	B _{1,0} B _{3,0}	RA1	RS A _{2,5}
49	A _{2,2} A _{4,2}	RA1	RS A _{3,7}	52	B _{1,2} B _{3,2}	RA1	RS A _{2,7}
50	A _{2,4} A _{4,4}	RA1	RS A _{3,0}	53	B _{1,4} B _{3,4}	RA1	RS A _{2,0}
51	A _{2,6} A _{4,6}	RA1	RS A _{3,2}	54	B _{1,6} B _{3,6}	RA1	RS A _{2,2}

表 9 硬體資源需求比較

Architecture	Mult.	Adder	Storage
[2]	32	32	20N
[7]	2N ²	2N ²	3N ²
[12]	24	21	14N
[13]	32	28	2N ²
[14]	10	16	14N
This work	10	16	9N

4-3 記憶體組織架構

在我們所提出的二維轉換的架構中，有三個記憶體模組：Memory1、Memory2 及 Temp，所使用的記憶體架構及資料儲存的方式分別如圖 9、10 和 11 所示。在 Memory1 中，我們需要三組如圖 9 所示之記憶體，以便儲存經由一維列運算所產生的數值。這些值再依奇數訊號和偶數訊號分別儲存。

在 Memory2 中，需要四組如圖 10 所示之記憶體來儲存 9/7 濾波器運算所產生之值。我們將同一列的資料儲存在一頁中，記憶體前半的位址儲存奇數訊號，偶數訊號則儲存在記憶體後半的位址。

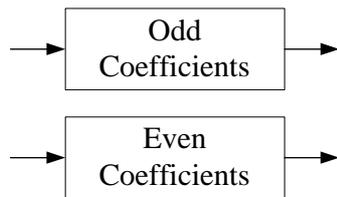


圖 9 Memory1 的記憶體架構



圖 10 Memory2 的記憶體架構

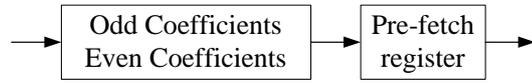


圖 11 Temp 的記憶體架構

對 Temp 作存取的時候，必須作記憶體的預先讀取以免發生資料衝突。如圖 11，我們將一列的資料儲存在一頁中，記憶體前半段儲存奇數訊號，偶數訊號儲存在記憶體後半段。在讀出端我們加上一個預讀取暫存器來儲存預先讀出的資料。

4-4 記憶體需求

假設要轉換的影像大小為 $N \times N$ ，在 Memory1 中我們需要使用六組大小為 $N/2$ 的儲存模組，Memory2 需要四組大小為 N 的儲存模組。進行 9/7 濾波器運算所需要的記憶體大小為 $9N$ 。

4-5 效能分析

如同一維架構，一個 $N \times N$ 的影像不論是以 5/3 濾波器運算或以 9/7 濾波器運算，當管線的資料填滿之後，每個週期可以送出一個高頻訊號和一個低頻訊號。當進行 5/3 轉換時，要填滿管線大約需要 $N + 3$ 個週期，進行 9/7 轉換填滿管線大約需要 $2N + 10$ 個週期，則完成一次二維轉換 5/3 濾波器需要 $(N^2/2) + N + 3$ 個週期，9/7 濾波器需要 $(N^2/2) + 2N + 10$ 個週期。

表 9 將我們的二維架構在進行 9/7 濾波器運算時，使用到的乘法器數量、加法器數量、以及記憶體需求和其它的架構作一個比較。輸入的影像解析度為 $N \times N$ 。由表中可以看出此架構具有節省記憶體的優勢。

五、效能評估

在效能評估及驗證方面，我們以 Synplicity 公司的 Synplify Pro 來進行 FPGA 的合成，目標元件為 Altera 公司所生產的 EPF10K200SRC240-1，完成佈局之一維架構操作頻率為 29.6 MHz。在此工作頻率下，以解

析度為 256×256 和 512×512 的灰階影像進行一階轉換分別需要

$$512 \times 512 : \frac{512 \times 512}{29.6 \text{ MHz}} = 8.85 \text{ ms}$$

$$1024 \times 1024 : \frac{1024 \times 1024}{29.6 \text{ MHz}} = 35.42 \text{ ms}$$

以一般 JPEG2000 運算需要進行五階轉換，則完成一張灰階影像需要的時間為

$$512 \times 512 : 8.85 + 2.21 + 0.55 + 0.14$$

$$+ 0.03 = 11.78 \text{ ms}$$

$$1024 \times 1024 : 35.42 + 8.85 + 2.21 + 0.55 + 0.14$$

$$+ 0.03 = 47.2 \text{ ms}$$

因而每秒可以連續編碼的灰階影像數為

$$512 \times 512 : \left\lfloor \frac{1000}{11.78} \right\rfloor = 84 \text{ (frames/s)}$$

$$1024 \times 1024 : \left\lfloor \frac{1000}{47.2} \right\rfloor = 21 \text{ (frames/s)}$$

二維架構的運算需要較長的時間延遲之後才能有結果輸出，但是往後的每個週期皆可兩個像素的輸出，當輸入的影像解析度夠大時，二維架構的效能約為一維架構的兩倍。同樣以 29.6 MHz 的工作頻率進行編碼，則完成一張灰階影像需要的時間為

$$1024 \times 1024 : 23.6 \text{ ms}$$

$$2048 \times 2048 : 70.84 + 23.6 = 94.44 \text{ ms}$$

而每秒可以編碼的灰階影像數為

$$1024 \times 1024 : \left\lfloor \frac{1000}{23.6} \right\rfloor = 42 \text{ (frames/s)}$$

$$2048 \times 2048 : \left\lfloor \frac{1000}{94.44} \right\rfloor = 10 \text{ (frames/s)}$$

可知在解析度為 1024×1024 時，其效能已能滿足動態影像編碼的要求。

五、結論

在本論文中，我們針對應用於 JPEG2000 之上提式離散小波轉換，提出一種整合 $5/3$ 和 $9/7$ 濾波器的管線化電路架構。我們所提出的一維架構具有精簡且彈性的特點，但是需要對外部的記憶體作大量的存取。以一維架構為基礎而延伸出來的二維架構，經由重新安排輸入訊號的順序，可以有效減少記憶體的使用量。相較於一維架構，二維架構可減少一半的外部記憶體存取，運算速度也快了將近一倍，但只能針對特定的影像解析度作運算。

對於未來的發展與改進，在效能方面可以進一步提升運算元件的速度，因為整個系統的運算速度主要是受限於管線暫存器之間的運算延遲。此外，我們所提出的架構，只要修改

運算的參數，就可以進行反轉換的運算，如此便可以完成一個完整的編碼解碼系統。

參考文獻

- [1] S. G. Mallat, "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.11 no. 7 pp. 674-693, July 1989.
- [2] M. Vishwanath, R. Owens, and T. Acharya, "VLSI Architecture for the Discrete Wavelet Transform," *IEEE Trans. Circuits Syst. II*, vol. 42, pp. 305-316, May 1995.
- [3] J. S. Fridman and E. S. Manolakis, "Discrete Wavelet Transform: Data Dependence Analysis and Synthesis of Distributed Memory and Control Array Architecture," *IEEE Trans. Signal Processing*, vol. 45, pp.1291-1308, May 1997.
- [4] T. Acharya, "A High Speed Systolic Architecture for Discrete Wavelet Transforms," in *Proc. IEEE Global Telecommun. Conf.*, vol. 2, 1997 pp. 669-673.
- [5] K. K. Parhi and T. Nishitani, "VLSI Architectures for Discrete Wavelet Transforms," *IEEE Trans. VLSI Syst.*, vol. 1, pp. 191-202, June 1993.
- [6] A. Grzeszczak, M. K. Mandal, S. Panchanathan, and T. Yeap, "VLSI Implementation of Discrete Wavelet Transform," *IEEE Trans. VLSI Syst.*, vol. 4, pp. 421-433, June 1996.
- [7] C. Chakrabarti and M. Vishwanath, "Efficient Realizations of the Discrete and Continuous Wavelet Transforms: From Single Chip Implementations to Mappings on SIMD Array Computers," *IEEE Trans. Signal Processing*, vol. 43, pp.759-771, Mar. 1995.
- [8] W. Sweldens, "The Lifting Scheme: A Construction of Second Generation Wavelets," Tech. Rep. 1995:6 Industrial Mathematics Initiative. Department of Mathematics. University of South Carllina 1995.
- [9] W. Sweldens, "The Lifting Scheme: A Custom-design construction of Biorthogonal Wavelets," *Appl. Comput. Harmon. Anal.* vol. 3(2). pp. 186-200. 1996.
- [10] I. Daubechies and W. Sweldens, "Factoring Wavelet Transform into Lifting Steps." *J. Foureier Anal. Appl.*, vol 4, Nr. 3, pp. 247-296.
- [11] M. Boliek, C. Christopoulos and E. Majani, "JPEG 2000 part I Final Committee Draft Version 1.0 (ISO/IEC FDIS15444-1),"

- ISO/IEC JTC/SC29/WG/N164R, March 16, 2000.
- [12] C. Yu and S. J. Chen, "Design of An Efficient VLSI Architecture for 2-D Discrete Wavelet Transforms," *IEEE Trans. Consumer Elect.*, vol. 45, no 1, pp. 135-140, 1999.
- [13] S. K. Paek, and L. S. Kim, "2D DWT VLSI Architecture for Wavelet Image Processing," *Electron. Lett.*, vol. 34, no. 6, pp. 537-538, 1998.
- [14] C. -T. Huang, P.-C. Tseng, L.-G. Chen, "Efficient VLSI Architectures of Lifting-based Discrete Wavelet Transform by Systematic Design Method," *IEEE International Symposium on Circuits and Systems (ISCAS)*, vol. 5, pp.V-565-V-568, 26-29 May 2002.