

利用部分重組態架構實作多媒體數位影像處理模 組

Implementation of Multimedia Digital Signal Processing Module Using Partial Reconfiguration Architecture

黃朝章

元智大學資訊工程學系

cschwang@saturn.yzu.edu.tw

鄭樂天

元智大學資訊工程學系

s966002@mail.yzu.edu.tw

楊友仁

元智大學資訊工程學系

s966026@mail.yzu.edu.tw

黃金俊

元智大學資訊工程學系

s969102@mail.yzu.edu.tw

林之棟

台灣科技大學資工所

d9515004@yahoo.com.tw

摘要 本文提出利用部分重組態方法來實作多媒體影像處理模組，我們選擇了離散餘弦轉換與反離散餘弦轉換等二個影像處理模組來當作範例，利用矩陣轉置的特性，建立一個架構，方便利用部分重組態實作離散餘弦轉換與反離散餘弦轉換。在實作方面採用 *Xilinx ISE 9.2i* 以 *Verilog* 硬體描述語言來完成此架構，之後以 *FPGA* 進行功能的模擬和計算數值驗證。驗證正確後，將電路位元檔下載至 *FPGA* 實作，結果在檔案的大小(記憶體空間)上，完全組態電路約為 **2444KB**、部分重組態電路約為 **860KB**，百分比為 **35.18 %**；至於在時間(處理速度)上完全組態需時 **6sec**、部分重組態需時 **2sec**，百分比為 **33.33 %**。

關鍵字：離散餘弦轉換、反離散餘弦轉換、部分重組態

ABSTRACT *In this thesis, we propose the implement of multimedia digital image processing module using partial reconfiguration method, we choose discrete cosine transform (DCT) and inverse discrete cosine transform (IDCT) as the example, the use characteristic of the matrix transpose to build a architecture easier using partial reconfiguration implement discrete cosine transform and inverse discrete cosine transform. we use the Verilog HDL within Xilinx ISE 9.2i design tool to complete this architecture. After that, we use FPGA for function simulation and verify computation data. Confirm correct, then download the bitstream to the FPGA. The results we found, the file size on full configuration are 2444KB, on partial reconfiguration are 860KB, the*

percent are 35.18 %; and the times need on full configuration are 6sec, on partial reconfiguration are 2sec, the percent are 33.33 %.

Keyword: discrete cosine transform, inverse discrete cosine transform, Partial Reconfiguration

一、簡介

重組態架構(Reconfigurable architecture)或稱作可重組運算(Reconfigurable Computing, R.C.)是近年來 IC 設計領域中非常受矚目的作法，其主要原理就是在一個持續工作的 FPGA 系統中，已被規劃的電路單元因為功能性或是效能上的需要，可以將部份的硬體電路改變，而不需要因為此一變動而將所有 Bitstream 重新下載的一種架構。鑑於傳統的設計流程，往往會事先把整個設計的硬體電路以硬體描述語言(HDL)來架構，並且在編譯生成 Bitstream 後直接下載到電路板上，但是如果變更功能單元就必須將整個設計的原始碼加以修改，並再次編譯，再次下載到 FPGA 上，不但費時耗工，而且維護不易，彈性受限。

而在重組態的架構中又分成兩種不同的方式，即靜態(static)與動態(dynamic)的重組態。靜態可重組態是指系統必須是終止的時候才能夠去更改硬體的組態；而動態可重組態是指系統可以一邊運作，一邊組態，也就是系統不需停止目前的運作即可在系統運作途中更換功能模組內容。

重組態計算的想法雖然在 1962 年就已被提出，但由於當時硬體環境尚不成熟，大多研究著重在理論層面上，而沒有實際成品出來，到 1996 年由 Xilinx 發表了可重組態的 FPGA (Field Programmable Gate Array)之後，就開始有以 FPGA 為基礎實作的重組態計算。

綜觀現今雖然硬體技術進步良多，但硬體資源仍然是有限，在面對消費者所需功能越來越多，但我們卻是無法將所有的功能均實作為硬體，因為其大小與成本都是可觀的，而且是否合乎成本效益？所以為能根據系統所需之功能，彈性實作其硬體，當硬體資源大小不足實作時，我們利用部分重組態架構來解決，在固定的硬體大小中，有效的做動態置換的方式，以達更多的功能單元。

動態重組態架構與一般 FPGA 架構不同的地方在於系統可以在工作中(Power on)依照使用者需求來置換功能單元(Function Unit)，而不需整個系統關掉。利用這種特性，我們可以在一個晶片上建構許多功能單元去做動態置換，來提升系統效能與彈性。

現今由於許多可攜式的多媒體平台與嵌入式系統結合，熱門的領域包括影像、語音、網路...等。因此本文在驗證部分重組態功能單元時選擇以影像處理為例，並選擇使用 DCT 與 IDCT 這兩個在影像處理領域中十分重要的解/編碼器當作我們的功能單元。

二、DCT 與 IDCT

離散餘弦轉換(Discrete cosine transform, DCT)和離散餘弦反轉換(Inverse discrete cosine transform, IDCT)，分別被利用在資料壓縮的演算法上，如儲存媒體的 MPEG (Moving Picture Experts Group)、彩色靜態影像傳輸 JPEG (Joint Photographic Experts Group)等，因為轉換編碼可以獲得較好的壓縮可能性，進而達到降低位元速率的目的。

DCT 的轉換輸入是 8×8 個點，輸出則是 8×8 個係數，是由時域(space domain)轉換到頻域(frequency domain)，反之，由頻域轉換到時域

則為 IDCT。由於 DCT 需要很長的時間來作運算，所以它是整個影像壓縮最花時間的部份，為了計算上的效率，我們把 DCT 的運算改以 2 次的 1D-DCT，來取代 1 次的 2D-DCT 作運算。

一維(One-Dimensional)的 DCT 數學式如式(1)，IDCT 如式(2)；二維(Two-Dimensional)的 DCT 數學式如式(3)，IDCT 如式(4)。

■ 1D-DCT :

$$Y(k) = \alpha(k) \sum_{n=0}^7 x(n) \cos \left[\frac{\pi(2n+1)k}{16} \right] \quad 0 \leq k \leq 7 \quad (1)$$

$$\text{where } \alpha(0) = \frac{1}{2\sqrt{2}}, \alpha(k) = \frac{1}{2} \text{ for } 1 \leq k \leq 7$$

■ 1D-IDCT :

$$x(n) = \sum_{k=0}^7 \alpha(k) Y(k) \cos \left[\frac{\pi(2n+1)k}{16} \right] \quad 0 \leq k \leq 7 \quad (2)$$

$$\text{where } \alpha(0) = \frac{1}{2\sqrt{2}}, \alpha(k) = \frac{1}{2} \text{ for } 1 \leq k \leq 7$$

■ 2D-DCT :

$$D(u,v) = \frac{C(v)}{2} \frac{C(u)}{2} \sum_{y=0}^7 \sum_{x=0}^7 f(y,x) \cos \left[\frac{\pi(2x+1)u}{16} \right] \cos \left[\frac{\pi(2y+1)v}{16} \right] \quad 0 \leq u, v \leq 7 \quad (3)$$

$$\text{where } C(0) = \sqrt{\frac{1}{2}}, C(k) = 1 \text{ for } 1 \leq k \leq 7$$

■ 2D-IDCT :

$$f(y,x) = \sum_{v=0}^7 \frac{C(v)}{2} \sum_{u=0}^7 \frac{C(u)}{2} D(v,u) \cos \left[\frac{\pi(2x+1)u}{16} \right] \cos \left[\frac{\pi(2y+1)v}{16} \right] \quad 0 \leq x, y \leq 7 \quad (4)$$

$$\text{where } C(0) = \sqrt{\frac{1}{2}}, C(k) = 1 \text{ for } 1 \leq k \leq 7$$

由上述的數學式中可以看出每一個輸出 Y(k)皆是由輸入 x(n)的 Cosine 分量所組成，我們將 Cosine 分量以 A(k)表示之，以 1-D DCT 為例，可以改寫成式(5)。

$$Y(k) = \frac{C(k)}{2} \sum_{n=0}^7 x(n) \cos \left[\frac{\pi(2n+1)k}{16} \right] \quad (5)$$

$$= \sum_{n=0}^7 [A_k(n)] \times x(n)$$

$$\text{where } A_k(n) = \frac{C(k)}{2} \cos \left[\frac{\pi(2n+1)k}{16} \right]$$

再把 A(k)的計算結果獨立出來，改寫成矩陣的方式，如式(6)。而參數 a-g，如式(7)。

$$A = \begin{bmatrix} a & a & a & a & a & a & a & a \\ b & d & e & g & -g & -e & -d & -b \\ c & f & -f & -c & -c & -f & f & c \\ d & -g & -b & -e & e & b & g & -d \\ a & -a & -a & a & a & -a & -a & a \\ e & -b & g & d & -d & -g & b & -e \\ f & -c & c & -f & -f & c & -c & f \\ g & -e & d & -b & b & -d & e & -g \end{bmatrix} \quad (6)$$

$$\begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \end{bmatrix} = \sqrt{\frac{2}{N}} \begin{bmatrix} \cos \frac{\pi}{4} \\ \cos \frac{\pi}{16} \\ \cos \frac{\pi}{8} \\ \cos \frac{3\pi}{16} \\ \cos \frac{5\pi}{16} \\ \cos \frac{3\pi}{8} \\ \cos \frac{7\pi}{16} \end{bmatrix} = \begin{bmatrix} 0.353553 \\ 0.490393 \\ 0.461940 \\ 0.415735 \\ 0.277785 \\ 0.191342 \\ 0.097545 \end{bmatrix} \quad (7)$$

此時將 A(k)代入，則我們把 1D DCT 數學式(5)變成以矩陣方式表示，如式(8)。

$$\begin{bmatrix} Y(0) \\ Y(1) \\ Y(2) \\ Y(3) \\ Y(4) \\ Y(5) \\ Y(6) \\ Y(7) \end{bmatrix} = \begin{bmatrix} a & a & a & a & a & a & a & a \\ b & d & e & g & -g & -e & -d & -b \\ c & f & -f & -c & -c & -f & f & c \\ d & -g & -b & -e & e & b & g & -d \\ a & -a & -a & a & a & -a & -a & a \\ e & -b & g & d & -d & -g & b & -e \\ f & -c & c & -f & -f & c & -c & f \\ g & -e & d & -b & b & -d & e & -g \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \\ x(4) \\ x(5) \\ x(6) \\ x(7) \end{bmatrix} \quad (8)$$

式(8)的缺點是我們必須花費很多的乘法器與加法器。因此為了減少乘法器與加法器的數量，我們觀察式(8)，可以發現常數矩陣的奇數行有左右對稱(Symmetry)的關係；而偶數行則是左右對稱且相差一個負號，故我們可以將式(8)常數矩陣分解成兩個 4×4 的奇、偶數項矩

陣，如式(9)、式(10)為 DCT 的偶、奇數項，同理，1D-IDCT 數學運算式以矩陣形式簡化，如式(11)、式(12)為 IDCT 的偶、奇數項。將式(9)式、(10)合併；式(11)、式(12)合併，得到 1D-DCT / IDCT 的矩陣運算式。

■ 1D-DCT 矩陣運算式：

$$\begin{bmatrix} Y(0) \\ Y(2) \\ Y(4) \\ Y(6) \end{bmatrix} = \begin{bmatrix} a & a & a & a \\ c & f & -f & -c \\ a & -a & -a & a \\ f & -c & c & -f \end{bmatrix} \begin{bmatrix} x(0)+x(7) \\ x(1)+x(6) \\ x(2)+x(5) \\ x(3)+x(4) \end{bmatrix} \quad (9)$$

$$\begin{bmatrix} Y(1) \\ Y(3) \\ Y(5) \\ Y(7) \end{bmatrix} = \begin{bmatrix} b & d & e & g \\ d & -g & -b & -e \\ e & -b & g & d \\ g & -e & d & -b \end{bmatrix} \begin{bmatrix} x(0)-x(7) \\ x(1)-x(6) \\ x(2)-x(5) \\ x(3)-x(4) \end{bmatrix} \quad (10)$$

■ 1D-IDCT 矩陣運算式：

$$\begin{bmatrix} x(0) \\ x(2) \\ x(4) \\ x(6) \end{bmatrix} = \begin{bmatrix} a & a & a & a \\ c & f & -f & -f \\ a & -a & -a & a \\ f & -c & c & -f \end{bmatrix} \begin{bmatrix} Y(0) \\ Y(2) \\ Y(4) \\ Y(6) \end{bmatrix} + \begin{bmatrix} b & d & e & g \\ d & -g & -b & -e \\ e & -b & g & d \\ g & -e & d & -b \end{bmatrix} \begin{bmatrix} Y(1) \\ Y(3) \\ Y(5) \\ Y(7) \end{bmatrix} \quad (11)$$

$$\begin{bmatrix} x(1) \\ x(3) \\ x(5) \\ x(7) \end{bmatrix} = \begin{bmatrix} a & a & a & a \\ c & f & -f & -f \\ a & -a & -a & a \\ f & -c & c & -f \end{bmatrix} \begin{bmatrix} Y(0) \\ Y(2) \\ Y(4) \\ Y(6) \end{bmatrix} + \begin{bmatrix} b & d & e & g \\ d & -g & -b & -e \\ e & -b & g & d \\ g & -e & d & -b \end{bmatrix} \begin{bmatrix} Y(1) \\ Y(3) \\ Y(5) \\ Y(7) \end{bmatrix} \quad (12)$$

遵此，將 2D-DCT 的數學式(3)轉換成矩陣形式，以 $A_{u,v}$ 表示二維離散餘弦轉換的常數矩陣，而 $A_{u,v}^T$ 為常數矩陣 $A_{u,v}$ 的轉置矩陣。得到 2D-DCT 的矩陣數學運算式如式(12)，2D-IDCT 的矩陣數學運算式如式(13)。

$$D(u,v) = A_{u,v} \times f(y,x) \times A_{u,v}^T \quad (12)$$

$$f(y,x) = A_{u,v}^T \times D(u,v) \times A_{u,v} \quad (13)$$

由 2D-DCT 的式(12)與 2D-IDCT 的式(13)兩個數學運算式來加以分析，將其分解成兩個 1D-DCT 矩陣數學運算式，以 2D-DCT 為例分解後如式(14)所示。

$$\begin{aligned} Y &= AX \\ Z &= YA^T \end{aligned} \quad \text{where } Z = D(u,v), A = A_{u,v}, X = f(y,x) \quad (14)$$

第一次 1D-DCT 運算與第二次 1D-DCT 運算上有些許的不同，需設計一個以列(Row)為主的運算模組以及一個以行(Column)為主的運算模組，故在做重組態時，其 bit-stream 檔案較大。因此我們運用矩陣轉置的特性“(AB)^T = B^TA^T”，將式(14)替換成式(15)，所以我們可以第一次行方向 1D-DCT 運算與第二次列方向 1D-DCT 運算以相同的模組設計，不須特別設計行方向與列方向兩種不同的 1D-DCT 模組，減少花費 FPGA 的儲存空間。

$$\begin{aligned} Y &= AX \Rightarrow Y^T = (AX)^T \\ Z &= YA^T \Rightarrow Z = (Y^T A)^T \end{aligned} \quad (15)$$

三、設計與實驗

(一)、相關環境

本文實作 FPGA 為 Xilinx Virtex-5，搭配其軟體為 ISE 9.2i；合成與邏輯配置軟體為 PlanAhead 9.2。

1. FPGA

FPGA，現場可程式化邏輯閘陣列(Field Programmable Gate Array)提供了邏輯閘陣列的特性與可程式化陣列邏輯(Programmable Array Logic, PAL)元件或邏輯閘陣列邏輯(Gate Array Logic, GAL)的規劃彈性，所以可以縮短電子產品的雛型系統的製作時間，達到了快速雛型化

(Fast Prototyping)的目標，圖 1 說明了基本的 FPGA 架構圖，分為三個基本區塊，即可組態邏輯(Configurable Logic Block, CLB)區塊、輸入輸出區塊(Input/Output Block, IOB)、及內部接線跟傳輸區塊(Switch Matrix)。

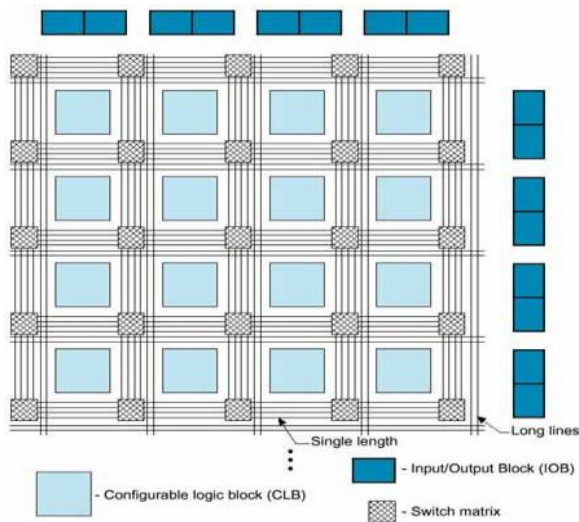


圖 1：FPGA 基本架構圖

2. ISE 9.2i

ISE 環境主要有 4 個區塊(如圖 2 所示),透過其輔助，我們可以從一開始的設計 HDL 開始，直到產生可燒入至 FPGA 的檔案，這中間流程皆可在 Xilinx ISE 上進行操作，下面分別介紹。

(1)、用以顯示 FPGA 的型號與設計文件。其中設計文件包含了模組設計(Module)、模擬波形設計(Testbench)與限制設計(Constrain File)等三者。

(2)、選取模組設計文件可以進行合成(Synthesize)、電路實現(Implement Design)或產生可程式檔案(Generate Programming File)；選取模擬波形設計文件，則可驗證模組設計是否正確。

(3)、用以編輯硬體描述語言(Hardware Description Language, HDL)。

(4)、顯示合成、電路實現或是產生可程式檔案的警告或錯誤訊息。

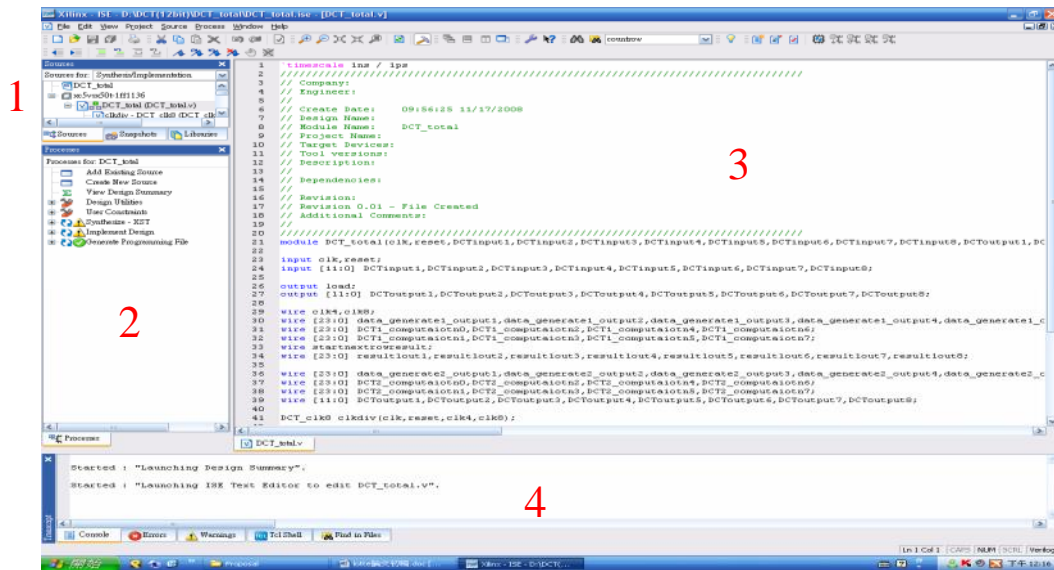


圖 2：ISE 環境

3、PlanAhead 9.2

PlanAhead 9.2 軟體能加速合成與邏輯配置 (place-and-route) 作業，協助我們掌握與了解如何完成設計建置工作，並以最少的設計步

驟，達到使用頻率最大的預期目標，而且支援部分重組態的設計流程，以圖形介面來呈現，對於我們設計部分重組態模組是相當方便，如圖 3 所示。

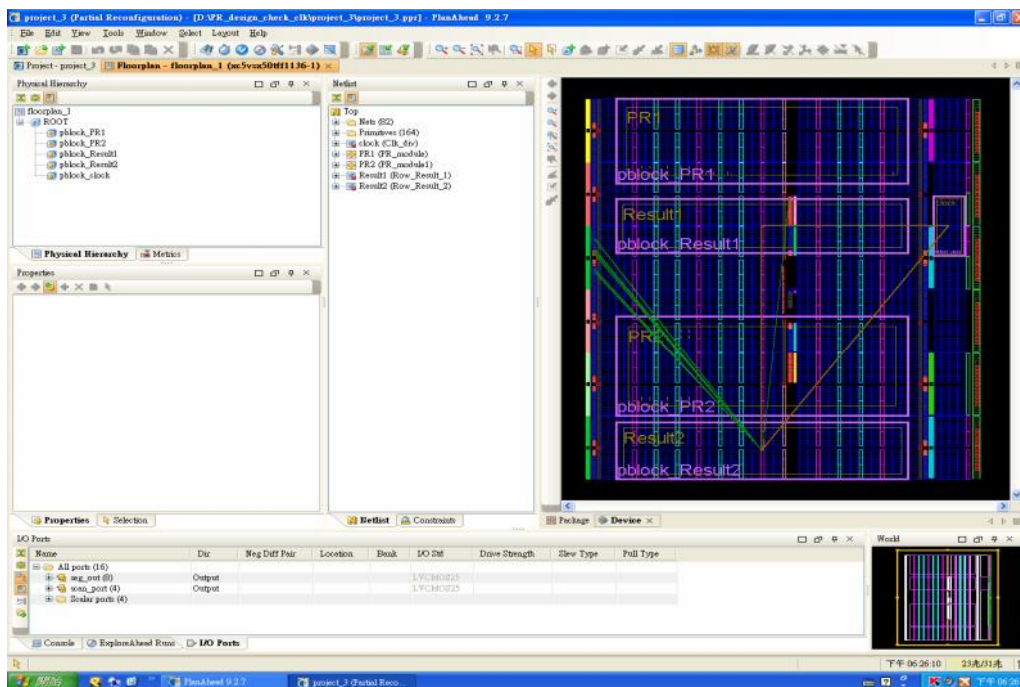


圖 3：PlanAhead 環境

(二)、設計與實驗

設計部分重組態可以分為七個步驟，如圖 4 所示。

- (1)、用硬體描述語言設計功能單元。
- (2)、對功能單元做區域、時序、I/O 腳位限制。
- (3)、實作沒有部分重組態的模組。
- (4)、驗證沒有部分重組態模組是否正確。
- (5)、實作部分重組態模組中公用部分。
- (6)、實作個別需要部分重組態模組。
- (7)、結合全部模組，產生 Bit-stream 檔。

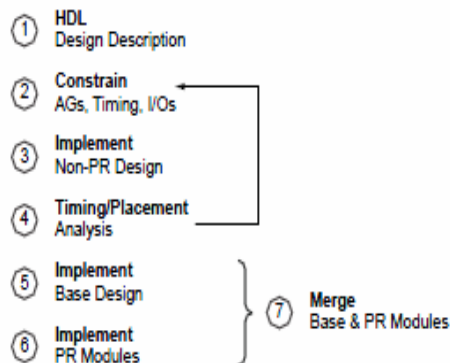


圖 4：部分重組態設計流程圖

本文以模組化設計 DCT/IDCT，利用部分重組態去加速 DCT/IDCT 快速置換的時間，其中包含固定區域(Fixed Area)和部分重組態區域(Partial Reconfiguration Area)。

首先我們先把兩個 1D DCT/IDCT 運算單元視為部分重組態區域，兩個矩陣轉置視為靜態區域，如此我們只需要置換 1D DCT/IDCT 運算單元就可以達到 2D DCT/IDCT 之功能，如圖 5 所示。

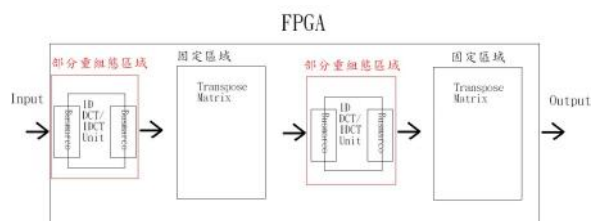


圖 5：2D DCT/IDCT 架構圖

四、結果驗證

實作上輸入 2D-DCT/IDCT 的輸入值，如圖 6、圖 9；再以 Xilinx ISE 9.2i 內建 Simulator 模擬 2D-DCT/IDCT 的運算結果，輸出如圖 7、圖 10；並與 Matlab 7.1 相互比較、驗證運算的結果，如圖 8、圖 11 所示，結果顯示兩者都是相符的。

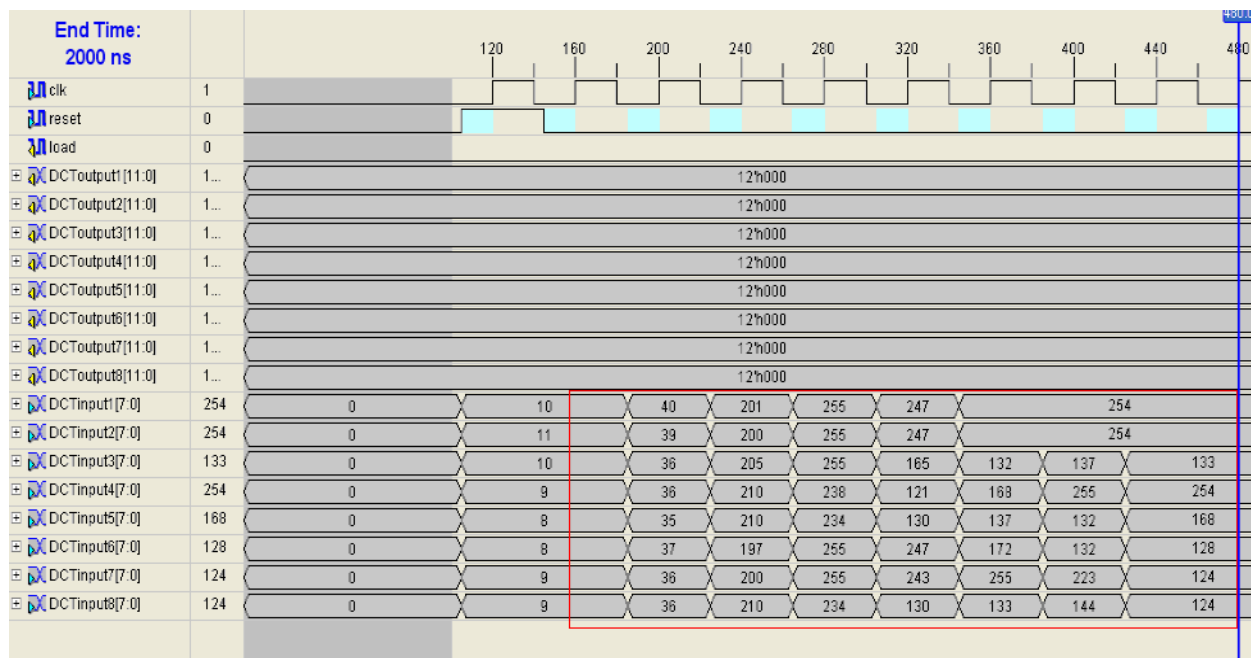


圖 6：2D DCT 輸入資料

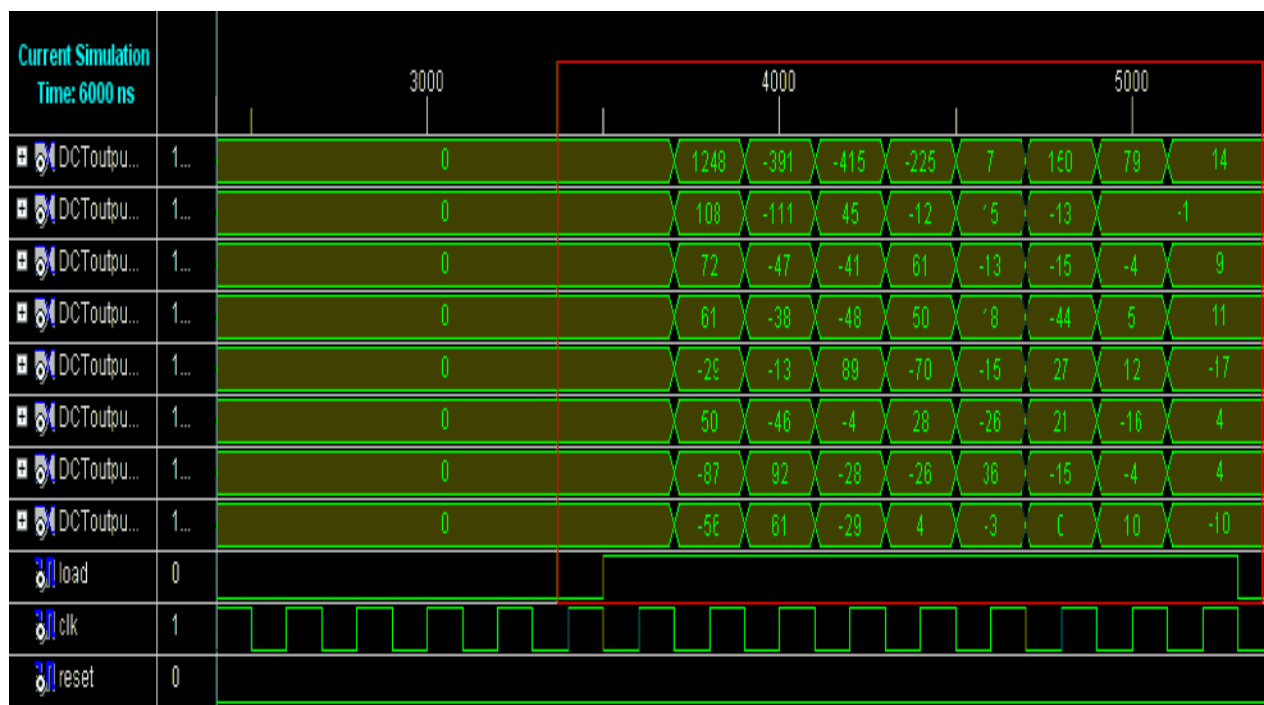


圖 7：2D DCT 運算結果

Array Editor - z								
Stack: Base								
	1	2	3	4	5	6	7	8
1	1248.5	-390.2115	-415.0703	-224.4935	7.5	150.9428	79.1126	14.9342
2	108.7212	-110.8218	45.1831	-11.8236	15.4612	-12.4751	-0.0118	-0.7443
3	72.2971	-46.9562	-40.6348	61.9718	-12.6525	-14.6226	-3.3063	9.7141
4	61.5109	-37.0655	-47.894	50.7856	18.0865	-43.639	5.095	11.8407
5	-28.5	-11.9937	89.1481	-69.7015	-15	27.7014	12.0519	-16.922
6	50.5968	-45.6752	-3.3239	28.0362	-25.9858	21.3809	-15.2135	4.6833
7	-86.772	92.657	-27.8063	-25.9948	36.089	-14.963	-3.8652	4.9981
8	-55.757	61.9309	-28.1751	4.7264	-2.8844	1.0109	10.068	-9.3447

圖 8：Matlab 2D DCT 運算結果

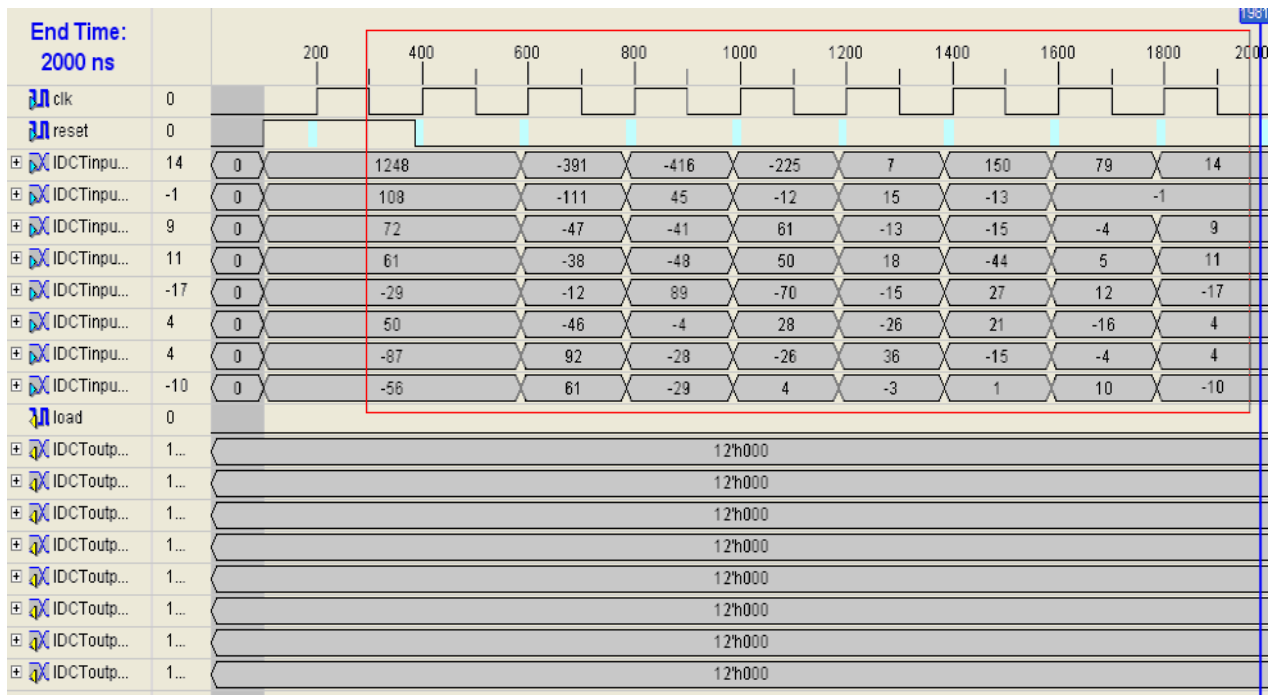


圖 9：2D IDCT 輸入資料

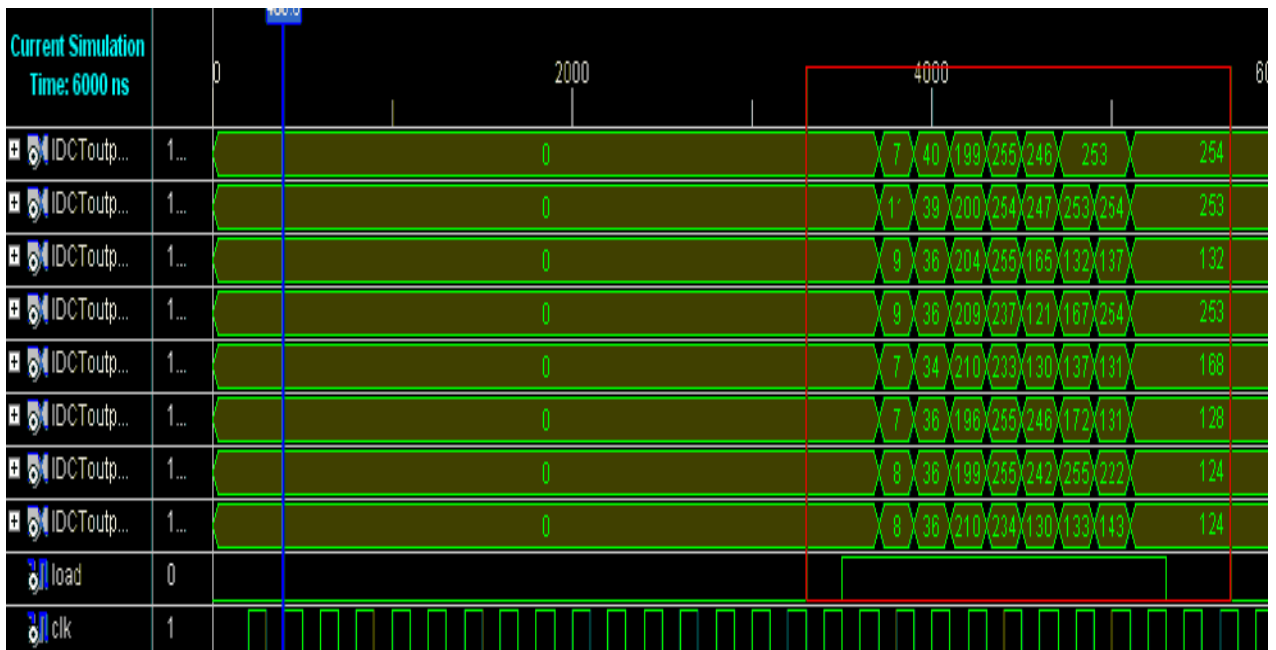


圖 10：2D IDCT 運算結果

	1	2	3	4	5	6	7	8
1	10	40	201	255	247	254	254	254
2	11	39	200	255	247	254	254	254
3	10	36	205	255	165	132	137	133
4	9	36	210	238	121	168	255	254
5	8	35	210	234	130	137	132	168
6	8	37	197	255	247	172	132	128
7	9	36	200	255	243	255	223	124
8	9	36	210	234	130	133	144	124

圖 11： Matlab 2D IDCT 運算結果

驗證結果相符後，我們把利用部分重組態設計的 DCT 與 IDCT，跟沒有利用部分重組態(完全組態)設計的 DCT 與 IDCT 的 Bit-stream 檔以 Xilinx iMPACT 經由 JTAG(Joint Test Action Group)下載到 FPGA 實驗板上，進行檔案大小(記憶體空間)與時間(速度)上的比較與驗證。結果發現沒有利用部分重組態設計的檔案大小為 2444KB，利用部分重組態設計的檔案大小約為 860KB，百分比為 35.18 %。另外在置換時間上，沒有利用部分重組態設計的置換時間需要 6sec；利用部分重組態設計的置換時間需要 2sec，百分比為 33.33 %。如表 1、表 2、表 3，明顯減少許多。

表 1：完全重組態時間表

```

IPIRG pulsed : 0
read back crc error : 0
Indicates that efuse logic is busy : 0
INFO:iMPACT:2219 - Status register values:
INFO:iMPACT - 0011 1111 1011 1110 0000 1011 1000 0000
INFO:iMPACT:579 - '5': Completed downloading bit file to device.
INFO:iMPACT - '5': Checking done pin....done.
'5': Programmed successfully.
PROGRESS_END - End Operation.
Elapsed time = 6 sec.

```

表 2：部分重組態時間表

```

IPIRG pulsed : 0
read back crc error : 0
Indicates that efuse logic is busy : 0
INFO:iMPACT:2219 - Status register values:
INFO:iMPACT - 0011 1111 1011 1110 0000 1011 1000 0000
INFO:iMPACT:579 - '5': Completed downloading bit file to device.
INFO:iMPACT - '5': Checking done pin....done.
'5': Programmed successfully.
PROGRESS_END - End Operation.
Elapsed time = 1 sec.

```

表 3：部分重組態檔案與時間比較表

	是否部分重組態	檔案大小	置換時間
DCT	Yes	850Kb	2 秒
IDCT	Yes	878Kb	2 秒
DCT	No	2444Kb	6 秒
IDCT	No	2444Kb	6 秒

五、結論與未來研究方向

(一)、結論

由於晶片硬體資源的大小終究有一個上限，加上現今硬體產品的功能日趨增加，因此利用動態重組態架構來提昇硬體資源的重覆使用率以及執行效能的想法，並解決空間不足所造成的系統無法順利運作的問題，已成為現今硬體晶片產品設計的主流。本文提出利用現有的硬體資源以 1D DCT/IDCT 的作法來實作 2D DCT/IDCT，並且利用矩陣轉置的特性，在同一個架構之下利用部分重組態的方式來置換 2D DCT/IDCT 功能。經過下載 FPGA 確認無誤後，驗證此設計方式是可行的。在相同大小的硬體空間下，去置換不同的功能單元，其除了增加硬體系統彈性與硬體資源重覆使用率外，也為了保有原本功能模組的執行效能，甚至更好的執行效能，所以在設計離散餘弦轉換演算法上，對架構的設計加入了許多可以考量的因素。

(二)、未來研究方向

在建立一個部分重組態平台上，可以加上 MicroBlaze 軟體核心，並且利用 ICAP(Internal Configuration Access Port)來加快部分重組態的時間；或是把設計的模組掛在 MicroBlaze 軟體核心底下，變成為 MicroBlaze 的內建系統，讓部分重組態有更多的功能。

誌謝

本文可以順利的完成要感謝，我的指導教授黃朝章教授在課業和研究方面的細心指導；郭祥舟教授和曾王道教授對本文提出的寶貴意見，使得本文內容得以更加完整；另外實驗室

金澍、智勇、億樹、宗翰學長對於研究及課業上的協助；信易和冠銓的幫忙、陪伴與支持，一併致謝。

參考文獻

- [1] 李宗翰，”以彈性重組態架構實現不同功能之運算”，2008.
- [2] 唐啟豪，”在 FPGA 架構的嵌入式系統中建立可自我重組態機制”，元智大學碩士論文，2005，pp13-14。
- [3] 陳寰宇，”以自我重組態架構設計嵌入式影像處理系統平台”，元智大學碩士論文，2005，p25-26.
- [4] 鄭信源，”VHDL 數位電路設計”，2002.
- [5] Avanindra Madiseti, Alan N. Willson, Jr. , “A 100 MHz 2-D 8 x 8 DCT/IDCT Processor for HDTV Applications”，1995.
- [6] Chao-Jang Hwang, Chih-Tung Lin, Shi-Jinn Horng, “Implementation of Different Function Units Using Flexible and Reconfigurable Architecture”, IEEE, Computer Society, 2009, pp85-86.
- [7] Khurram Bukhari, Georgi Kuzmanov, Stamatis Vassiliadis, “DCT and IDCT Implementations on Different FPGA Technologies”, Computer Engineering Lab, Delft University of Technology.