

利用部分重組態現場可程式化邏輯陣列平台實作 有限脈衝響應數位濾波器

Implementation of Finite Impulse Response Digital Filter Using Partial Reconfigurable FPGA Platform

黃朝章

元智大學資訊工程學系

cschwang@saturn.yzu.edu.tw

楊友仁

元智大學資訊工程學系

s966026@mail.yzu.edu.tw

鄭樂天

元智大學資訊工程學系

s966002@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 本文提出一個有限脈衝響應(*Finite Impulse Response, FIR*)濾波器，可以根據不同系統提供不同階層的 *FIR* 運算，利用部分重組態架構來變換系統需要的序列長度濾波器，並且利用展開方法(*Unfolded method*)，做出一個平行處理的 *FIR* 架構，來達到高效率、高彈性的 *FIR* 濾波器系統。實作上採用 *Xilinx ISE 9.2i Verilog* 硬體描述語言來完成此架構，之後以 *FPGA* 進行功能模擬和計算數值驗證。模擬驗證正確後，再以 *PlanAhead 9.2* 將整體架構合併成完全組態電路檔以及部分重組態電路檔，最後再以 *iMPACT* 將完全組態電路檔及部分重組態電路檔下載至 *FPGA* 上執行，結果在檔案的大小上(記憶體空間需求)，完全組態電路為 **2444KB**、部分重組態電路為 **166KB**，百分比為 **6.7 %**；至於在速度(時間)上完全組態需時 **6sec**、部分重組態需時 **1sec**，百分比為 **16.6 %**。

關鍵字：部分重組態

ABSTRACT—*This paper presents a parallel FIR (Finite Impulse Response) filter system design, using PR(Partial Reconfiguration) to change tap of FIR, which can achieve high flexibility, high performance, and shorten the time of configuration.*

In the present thesis, 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. Until simulation result is correct, using Xilinx PlanAhead 9.2 to merge all of the architecture, it generates full configuration circuit file and partial reconfiguration circuit file. Then, to verify by using iMPACT to download full configuration and partial reconfiguration circuit file to FPGA. The results we found, the file size(memory space) on full configuration are 2444KB, on partial reconfiguration are 166KB, the

percent are 6.7 %; and the times need (speed) on full configuration are 6sec, on partial reconfiguration are 1sec, the percent are 16.6 %.

Keyword: Partial Reconfiguration

一、簡介

鑑於傳統的設計流程裡，往往會事先把整個設計的硬體電路以硬體描述語言(HDL)先行架構，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。1996 年 Xilinx 公司提出了一個新的架構，即在硬體 FPGA 的支援下將功能單元模組化，並且在整個系統維持不斷電(power on)的情況下仍能夠更新功能單元，這也就是今日可重組態的初步架構。

重組態架構(Reconfigurable architecture)或是稱作可重組運算(Reconfigurable Computing, R.C.)是近年來 IC 設計上的一個主流，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之 Bitstream 重新下載的一種架構。

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。靜態重組態，當系統正在執行某一項功能時，要重組態另一功能模組，則整個系統必須停止目前的運作，直到系統將新的功能重組態完成後，才會恢復原本的運作，也就是說只有在編譯時才能改變整體架構。動態重組態，當系統需要重組態功能模組時，系統不需停止目前的運作，且在系統運作的途中可隨時更換功能模

組的內容來處理不同的運算，使系統能不間斷運作並使用新功能模組，令系統更加有效率、有彈性。

在一個部分重組態架構中，會劃分成兩種不同的區域，一是固定區域(Fixed Region)，為固定不變的區域，主要是用來置放一定會使用到的功能原件，比如記憶體存取、指令存取、I/O控制等。另一個是部分重組態區域(Partial Reconfiguration Region)，簡稱為PR Region，在FPGA中可以動態部分重組態，並不影響固定區域以及其它部分重組態區域，通常拿來放置一些不同的功能模組，可以在功能模組不需要時，置換成其它目前需要的功能模組，來提高系統效率。區域與區域之間溝通的橋樑則是透過 Bus macro 來連結，如圖1所示。

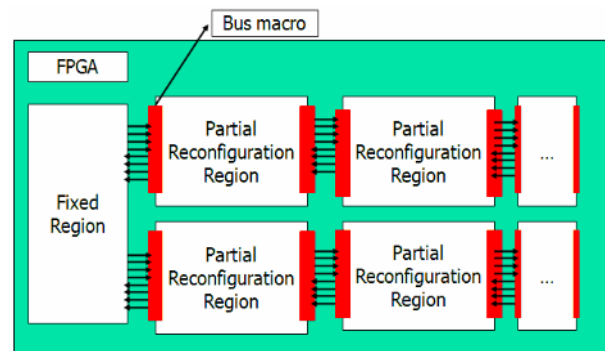


圖1：部分重組態架構圖

本文是利用部分重組態的功能，將FIR實作在FPGA上，系統可以即時改變FIR的階層，大大提高了FIR濾波器的彈性，而這種方法已在近年被提出。

有限脈衝響應(Finite Impulse Response, FIR)是一個 N 階層迴旋積(convolution)的運算式，根據使用者需求設計出不同階層數，不同係數的 FIR 變成不同功能的濾波器，例如高通濾波器(High Pass Filter)、帶通濾波器(Band

Pass Filter) 以及低通濾波器 (Low Pass Filter) 等，因此 FIR 濾波器在現代數位通訊系統及其它訊號處理應用上扮演著不可或缺的一部分。近年來，高效率的數位濾波器需求增加，產生很多大型積體電路系統用來做即時的數位訊號處理 (Digital Signal Processing, DSP)，而 FPGA 就是其中之一，近年來正被廣泛的應用在電信、網路、多媒體及數位信號處理。

二、有限脈衝響應濾波器 (FIR)

有限脈衝響應濾波器是一線性系統，將每一個輸入 $X(n)$ 與係數 coef 相乘後累加起來，而此 FIR 濾波器的架構，可由乘法累加器組成，其中係數會暫存在一個 register 裡，當需要變換係數時就由 coef 輸入來變換 register 裡的係數，而 X_{in} 為 FIR 的輸入，而 Nk_{addin} 為之前乘法累加後的結果，再與這個乘法累加器中 X_{in} 與暫存在 register 中的係數相乘的結果做累加，然後由 Nk_{addout} 輸出，乘法累加器示意圖如圖 2 所示。

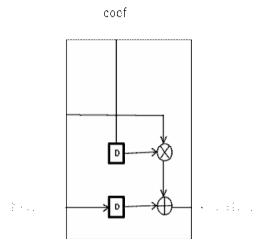


圖 2：乘法累加器

輸入信號為 $x(0)$ 、 $x(1)$... $x(n)$ ，經過該系統後的輸出信號為 $y(n)$ ， $y(n)$ 可表示為式(1)，簡化後得到式(2)，其中 $h_0, h_1, \dots, h_{(N-1)}$ 是濾波器的脈衝響應， N 是濾波器的階層數。

$$y(n) = h_0 x(n) + h_1 x(n-1) + \dots + h_{(N-1)} x(n-(N-1)) \quad (1)$$

$$y(n) = \sum_{k=0}^{N-1} h_k x(n-k) \quad (2)$$

根據式(2)導出 N 階層的 Direct Form FIR 濾波器，如圖 3 所示，圖中 Z^{-1} 為暫存器， $M_0 \sim M_{(N-1)}$ 為乘法器， M_0 表示輸入與係數 $h(0)$ 相乘， $M_1 \sim M_{(N-1)}$ 以此類推，而 $A_0 \sim A_{(N-2)}$ 為加法器。

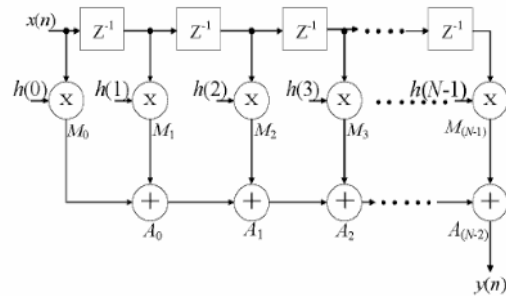


圖 3：Direct Form FIR 濾波器

根據圖 3，可以利用展開方法 (Unfolded method)，將它做 L -Unfolded 平行化如圖 4 所示，圖例是 3-Unfolded, 3-tap FIR 濾波器。圖中 D 代表 register， M_{ij} 是乘法器，代表 X_i 與 h_j 相乘， A 是加法器，將乘法結果相加，最後得到輸出。

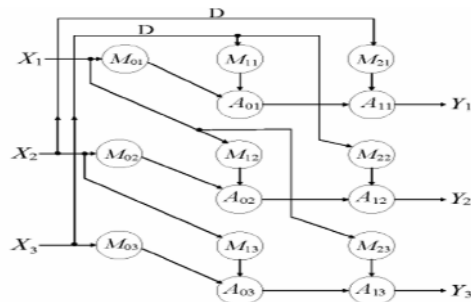


圖 4：3-Unfolded, 3-tap FIR 濾波器

三、設計與實驗

在 Direct Form 中，假設乘法器所花的時間為 T_M ，而加法器所花的時間為 T_A ，得到輸出 $Y(n)$ 所需的時間為 $T_M + (N - 1) \times T_A$ 。如果改使用 Data Broadcast 架構，如圖 5 所示，則會因為累加結果被放在暫存器中，而乘法的輸入是透過 Data Broadcast 方式傳輸，因此輸出 $Y(n)$ 所需的時間會變為 $T_M + T_A$ ，較 Direct Form 省下 $(N - 2) \times T_A$ 的時間。基此，本文將以此架構來作為設計架構。

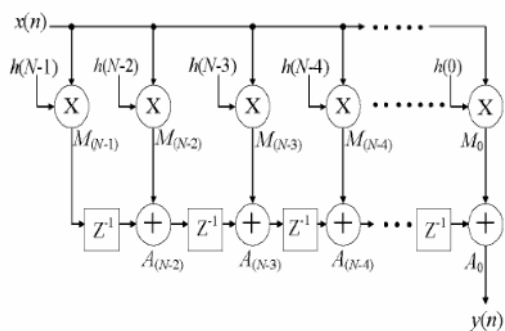


圖 5：Broadcast Form FIR 濾波器

再將圖 5 利用展開方法做 L-Unfolded 平行化，如此一來會較利於我們未來以 FPGA 作重組態處理時，不同階層數選擇的彈性，得到如圖 6 平行化的架構，圖例為一個 3-Unfolded, 3-tap broadcast Form FIR 濾波器。

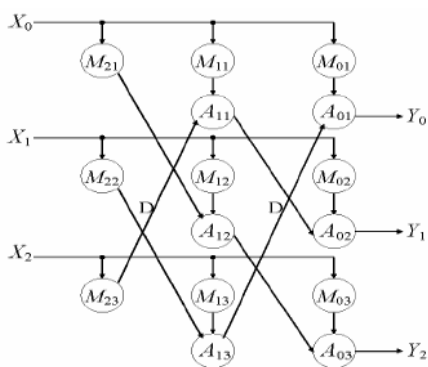


圖 6：3-Unfolded, 3-tap broadcast Form FIR filter

(一)、相關環境

本文實作的 FPGA 為 Xilinx Virtex-5，搭配的軟體為 ISE 9.2i；另外合成與邏輯配置軟體為 PlanAhead 9.2。

1. FPGA

(1). 基本架構圖

FPGA 是一個可程式化的邏輯設備，提供了邏輯開陣列的特性與可程式化陣列邏輯 (Programmable Array Logic, PAL) 元件或邏輯開陣列邏輯 (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，基本架構圖如圖 7 所示，其中分為三個區塊，即可組態邏輯 (Configurable Logic Block, CLB) 區塊、輸入輸出區塊 (I/O Block)、內部接線跟傳輸區塊。

透過軟體編譯的 bit-stream 重新組態成為一個新系統。在早期它是依據 bit-stream 將所有邏輯電路重新組態，造成 FPGA 必須要中斷等待而來重新組態，然現今 FPGA 提供了部分重組態 (Partial Reconfiguration) 的功能，縮短了極大的時間，而且只影響正要被組態的區域。

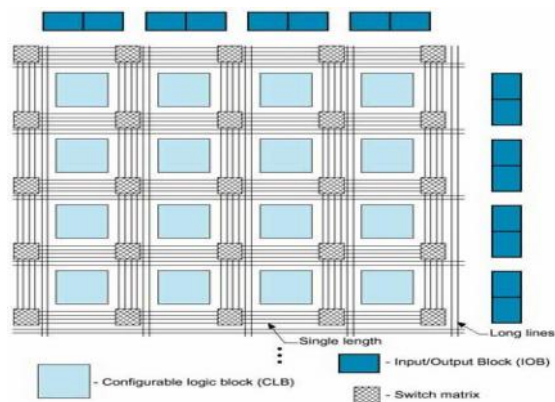


圖 7：FPGA 基本架構圖

(2). Xilinx Virtex-5 FPGA

圖 8 是 Xilinx Virtex-5 FPGA 的方塊圖。

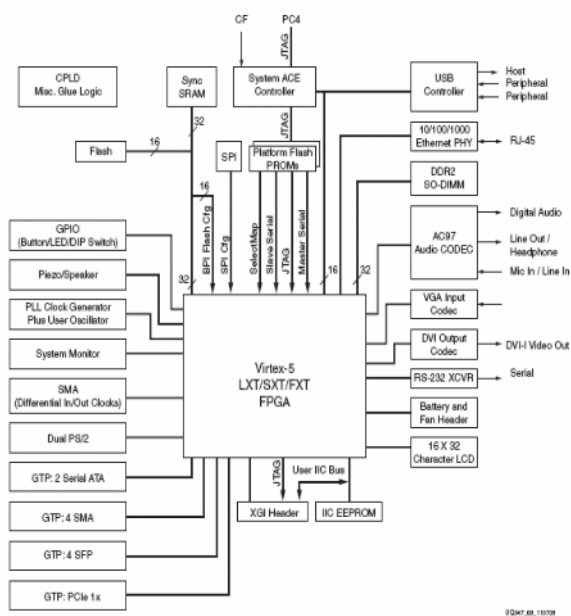


圖 8：Xilinx Virtex-5 架構圖

A. Block Memory

在 Xilinx Virtex-5 FPGAs 的 block RAM 中可供儲存 36Kb 的資料，而且可以被組態成兩個獨立的 18Kb RAMs 或者是單一個 36Kb RAM。每一個 36 Kb block RAM 可以組態 64K x 1、16K x 2、8K x 4、4K x 9、2K x 18 或者 1K x 36 的記憶體。每一個 18 Kb block RAM 可以組態 16K x 1、8K x 2、4K x 4、2K x 9、or 1K x 18 的記憶體。

B. 可組態邏輯區塊 (Configurable Logic Blocks, CLB)

CLB 是用來實現組合電路的主要邏輯資源，每一個 CLB 元件透過一個 Switch Matrix 來連結到 general routing matrix，每一個 CLB 元件包含兩個 slices，兩個 slices 是互相獨立沒

有直接相連，且各自有獨立的 carry chain，如圖 9 所示、標示法如圖 10 所示。

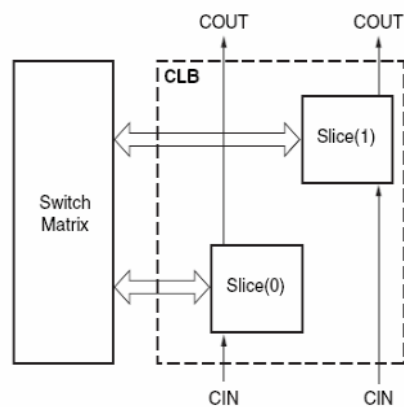


圖 9：CLB 與 Switch Matrix 連結圖

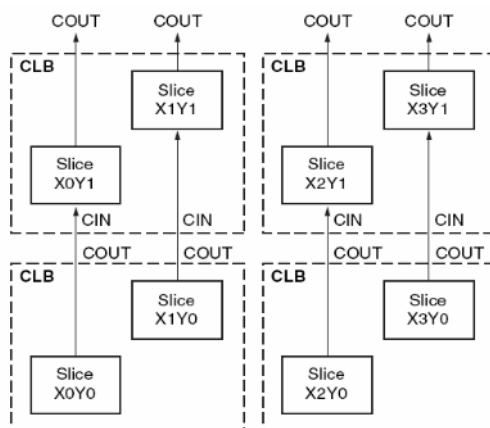


圖 10：CLB Slice 標示圖

2. ISE 9.2i

搭配 Xilinx Virtex-5 軟體的 ISE，當開啟專案後的主畫面，畫面上共可以分為四個區塊，分別為 Source Window、Processes Window、Transcript Window 和 Workspace，如圖 11 所示，接下來將分別介紹這四個區塊的功能。

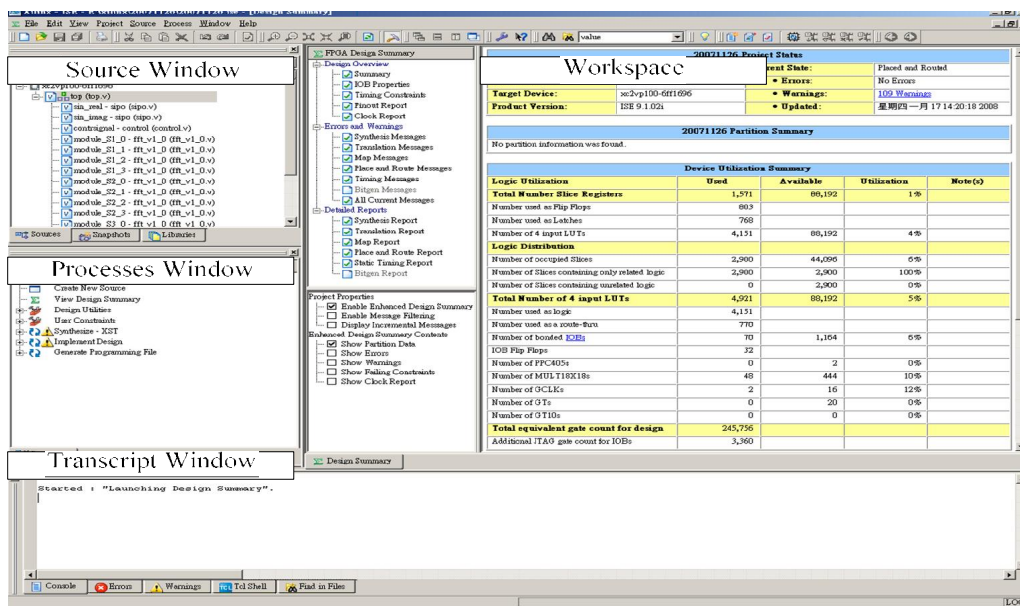


圖 11：ISE 工作視窗

(1). Source Window

顯示目前專案中所設計電路的相關資訊，檔案根據相關性以階層方式表示，包含專案名稱、選用的FPGA型號、設計的模組(Module)、模擬程式(Test Bench)以及在硬體配置限制檔案(Constrain File)。

(2). Processes Window

處理的動作與在Source Window中選取的檔案有關係，若選取為Test Bench則可進行功能模擬，可對上層模組(Top Module)進行合成(Synthesize)、電路實現(Implement Design)或是產生可程式檔案(Generate Programming File)，也就是在燒入FPGA所需要的檔案。除上述功能，還可開啟設計摘要以及對繞線擺放位置或是訊號傳遞時間的限制進行編輯，以及觀察在繞線完成後FPGA內部的繞線配置。

(3). Transcript Window

為訊息視窗，在進行硬體描述語言(Hardware Description Language, HDL)編譯時、電路合成(Synthesize)以及電路實現時(Implement Design)，會顯示目前伴隨這些程序產生的相關資訊，當程序發生錯誤時也是由此視窗告知相關警告或是錯誤訊息。

(4). Workspace

主要編輯畫面都會在此視窗內開啟，包含HDL、Test Bench和Constrain的編譯，以及瀏覽設計摘要或是RTL(register transfer level)。

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 12 所示。

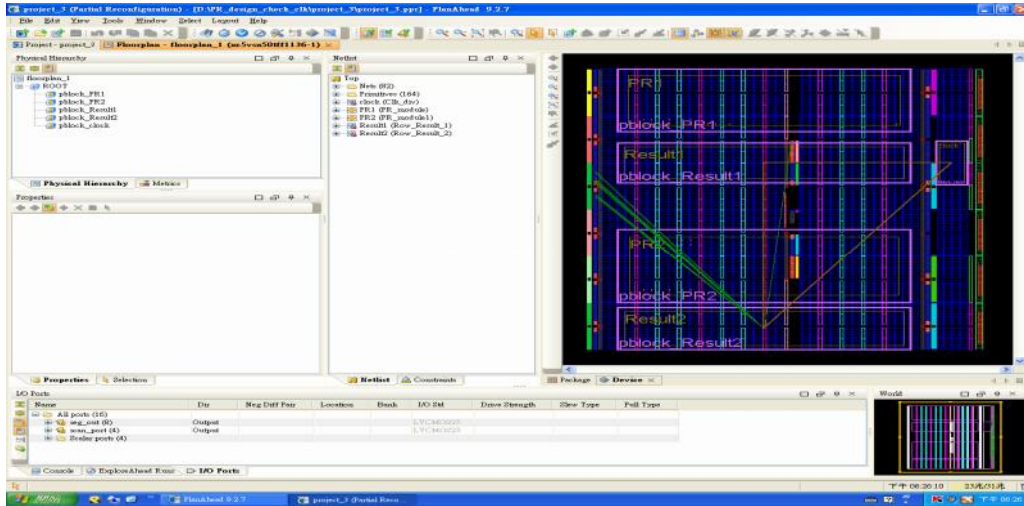


圖 12：PlanAhead 環境

(二)、設計與實驗

依圖 6 的 Unfolded Broadcast Form 推出一個 L-length 及 N-tap 的平行 FIR 架構如圖 13，從圖中可以看到每一個階層的規律，第一個階層的運算結果是 1-tap 的 $Y(n-L-1) \sim Y(n)$ ，然後接連到下一個階層，則是求出 2-tap 的

$Y(n-L-1) \sim Y(n)$ ，因此將每一個階層做成一個模組，如圖中藍色底色 MAC 處，從最下層加上這個模組，就可以因而藉以增加 tap 數，而且在任意一個階層 N 求得的輸出即為 N-tap 的 FIR 運算結果，因此可以任意選擇要哪一階層的輸出來做為設計者需求的輸出。

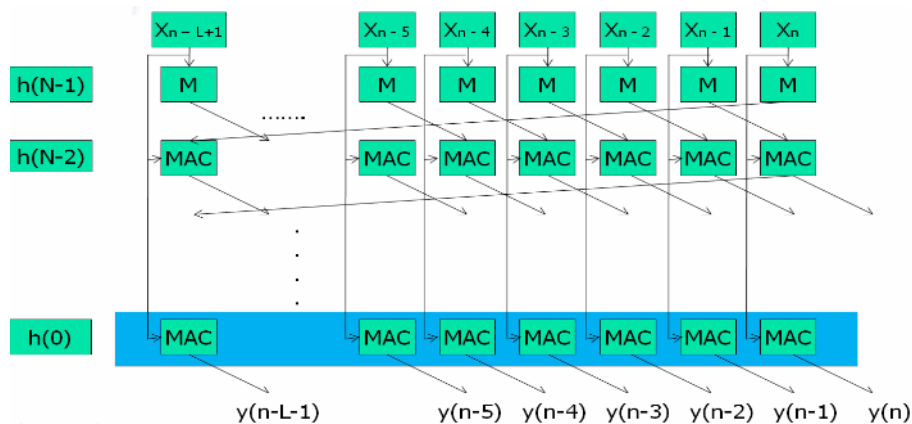


圖 13：L-length, N-tap Unfolded Broadcast Form FIR 濾波器

依循圖 13 完成後，我們將整體架構規劃分為 Fixed Region 及兩個 PR Region，如圖 14 所示，運算上 Fixed Region 會基本進行 8-tap FIR 運算，而兩個 PR Region 各可增加 8-tap FIR 運算，所以在這個架構上是可運算 1-tap ~ 24-tap FIR 運算結果，最後運算結果輸出至 Fixed Region 的 Data Out module，透過這個 module 將值輸出到 FPGA 外。

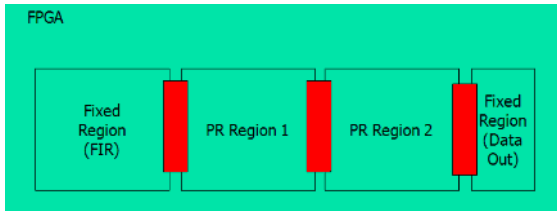


圖 14：整體架構圖

四、結果驗證

實作以一個 10-length 10-tap 的 FIR 濾波器為例，輸入為 $X_0 \sim X_9$ ，係數為 $coef_0 \sim coef_9$ ，輸出為 $Y_0 \sim Y_9$ ，模擬結果如圖 15 所示。再以 Microsoft Visual C++ 撰寫 FIR 運算來加以驗證，給予相同的輸入與係數值，得到結果如圖 16 所示，驗證兩者結果是相符的。

模擬驗證正確後，將設計的 FIR 平行架構做成兩部分，分別為靜態區域與部分重組態區域，再以 PlanAhead 將整體架構作區塊的規畫，如圖 17 所示。區塊規畫好後，產生完全組態燒錄檔及部分重組態燒錄檔，硬體繞線圖如圖 18 所示。

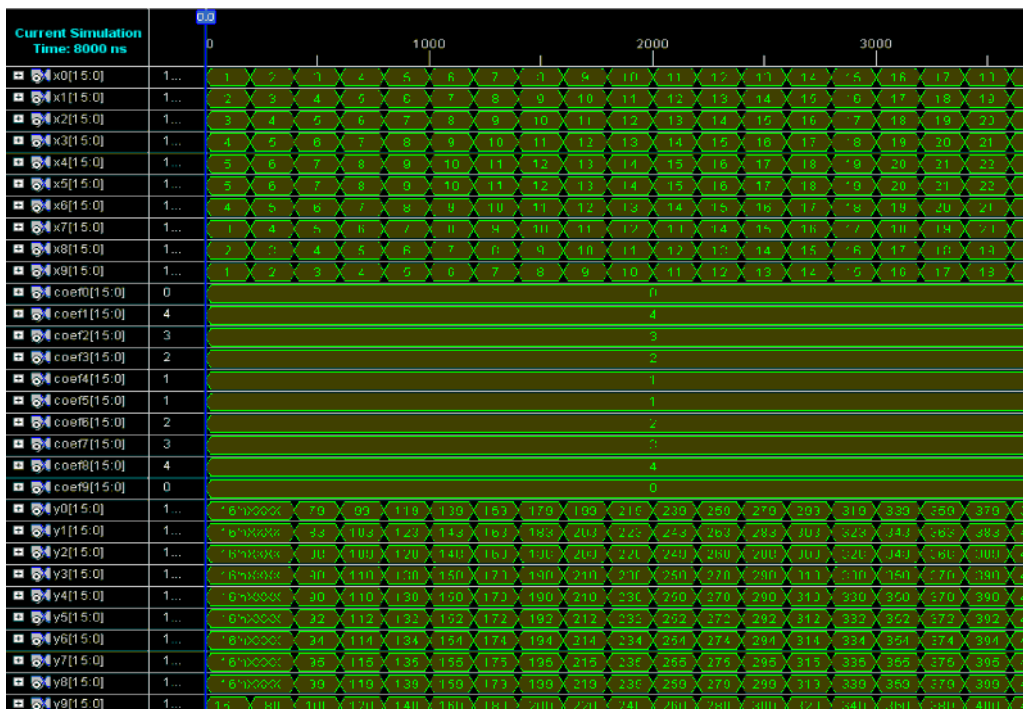


圖 15：10-length, 10-tap 平行 FIR 濾波器模擬結果


```

x = < 3 4 5 6 7 7 6 5 4 3 >
h = < 0 4 3 2 1 1 2 3 4 0 >
y = < 79 83 88 90 90 92 94 95 99 100 >

x = < 4 5 6 7 8 8 7 6 5 4 >
h = < 0 4 3 2 1 1 2 3 4 0 >
y = < 99 103 108 110 110 112 114 115 119 120 >

x = < 5 6 7 8 9 9 8 7 6 5 >
h = < 0 4 3 2 1 1 2 3 4 0 >
y = < 119 123 128 130 130 132 134 135 139 140 >

x = < 6 7 8 9 10 10 9 8 7 6 >
h = < 0 4 3 2 1 1 2 3 4 0 >
y = < 139 143 148 150 150 152 154 155 159 160 >

x = < 7 8 9 10 11 11 10 9 8 7 >
h = < 0 4 3 2 1 1 2 3 4 0 >
y = < 159 163 168 170 170 172 174 175 179 180 >

x = < 8 9 10 11 12 12 11 10 9 8 >
h = < 0 4 3 2 1 1 2 3 4 0 >
y = < 179 183 188 190 190 192 194 195 199 200 >

x = < 9 10 11 12 13 13 12 11 10 9 >
h = < 0 4 3 2 1 1 2 3 4 0 >
y = < 199 203 208 210 210 212 214 215 219 220 >

x = < 10 11 12 13 14 14 13 12 11 10 >
h = < 0 4 3 2 1 1 2 3 4 0 >
y = < 219 223 228 230 230 232 234 235 239 240 >

x = < 11 12 13 14 15 15 14 13 12 11 >
h = < 0 4 3 2 1 1 2 3 4 0 >
y = < 239 243 248 250 250 252 254 255 259 260 >

x = < 12 13 14 15 16 16 15 14 13 12 >
h = < 0 4 3 2 1 1 2 3 4 0 >
y = < 259 263 268 270 270 272 274 275 279 280 >

x = < 13 14 15 16 17 17 16 15 14 13 >
h = < 0 4 3 2 1 1 2 3 4 0 >
y = < 279 283 288 290 290 292 294 295 299 300 >

x = < 14 15 16 17 18 18 17 16 15 14 >
h = < 0 4 3 2 1 1 2 3 4 0 >
y = < 299 303 308 310 310 312 314 315 319 320 >

x = < 15 16 17 18 19 19 18 17 16 15 >
h = < 0 4 3 2 1 1 2 3 4 0 >
y = < 319 323 328 330 330 332 334 335 339 340 >

```

圖 16：Visual C++ 驗證結果

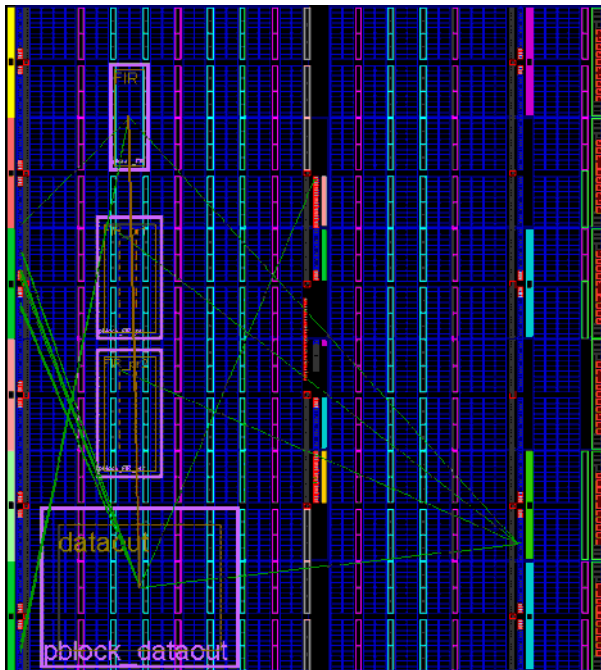


圖 17：PlanAhead 區塊規畫圖

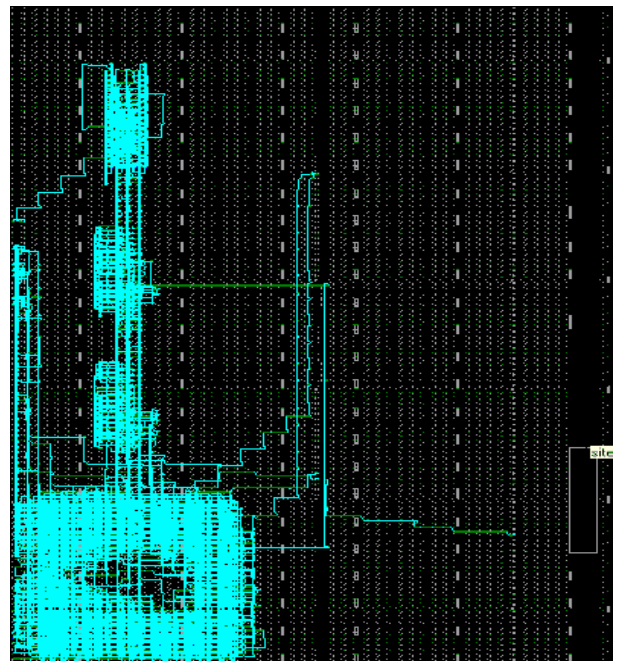


圖 18：硬體繞線圖

產生燒錄檔後，利用 iMPACT 軟體，透過 JTAG 將燒錄檔下載到 FPGA 上執行，結果在完全組態電路時間為 6sec，部分重組態時間為 1sec 如表 1、2 所示。

另外在檔案的大小上，完全組態電路為 2444KB、部分重組態電路為 166KB，百分比為 6.7 %；加上在時間(速度)上完全組態需時 6sec、部分重組態需時 1sec，百分比為 16.6 %，明顯調降許多，如表 3 所示。

表 1：完全組態時間表

```

CFG bus width auto detection result      :    0
CFG bus width auto detection result      :    0
Reserved                                 :    0
BFI address wrap around error           :    0
IFROG 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

```

CFG bus width auto detection result      :    0
Reserved                                 :    0
BFI address wrap around error           :    0
IFROG 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 =      0 sec.

```

表 3：檔案大小及組態時間表

完全組態		部分組態	
檔案大小	時間	檔案大小	時間
2444KB	6 sec	166KB	1sec

五、結論與未來研究方向

(一)、結論

利用部分重組態可彈性運算不同階層的平行架構 FIR 濾波器，除了利用平行架構來大大的加速 FIR 的運算速度外，並且利用可部分重組態來變換階層數，使用者可以自由變換由 1-tap 到 N-tap 的 FIR 濾波器，而且還可以透過改變 FIR 的係數來產生不同功能的濾波器，達到一個高彈性、高效率的 FIR 濾波器。

(二)、未來研究方向

本文實作部分重組態是利用軟體 iMPACT 來燒錄，而 Xilinx ML505 的開發板提供了嵌入式系統 Microblaze，此嵌入式系統可透過 ICAP (Internal Configuration Access Port) 從記憶卡讀取燒錄檔進行燒錄，可省去手動軟體下載的時間，未來能與 Microblaze 合併來達成部分重組態或許在檔案大小與速度(時間)上將會更加令人滿意。

誌謝

本文能夠順利的完成，感謝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

本文內容得以更臻完善。此外實驗室金澍學長在研究上的指導；智勇學長課業上的協助；福倉、信易、冠銓的互相幫忙，一併致謝。

參考文獻

- [1] A.V. Oppenheim and R.W. Schafer ,Digital signal processing. Prentice-Hall, Englewood Cliffs, NJ, 1975.
- [2] Chang-Seok Choi and Hanho Lee. “A Reconfigurable FIR Filter Design Using Dynamic Partial Reconfiguration”, ISCAS 2006.
- [3] G. Deepak, P.K. Meher and A.Sluzek. “Performace Characteristics of Parallel and Pipelined Implementation of FIR Filers in FPGA Platform”, ISSCS. 2007.
- [4] “Virtex-5 FPGA User Guide”, January 9, 2009.
- [5] Yeong-Jae Oh, Hanho Lee. “Chong-Ho Lee A Reconfigurable FIR Filter Design on a Partial Reconfiguration Platform”, CCE.2006.