可重組式數位相機設計

黃駿賢、張世學、熊博安

國立中正大學資訊工程學系 嵌入式系統實驗室

一、簡介

由於傳統設計上通常是軟硬體個別設計，常常造成最後整合時會出現所不能預期的問題。若提早在設計時就同時考慮軟硬體兩者，利用共同設計與驗證(Codesign and Coverification)的方法，加上FPGA可動態部分重組(Dynamic Partial Reconfiguration)的技術，充分利用有限的資源，來達到產品設計時間和空間上的最佳化，以及達到低功率(low power)之效能。如今我們欲用此一出發點，將其實作於數位相機的設計上，採用Xilinx ML310做為我們設計平台，利用鏡頭擷取影像後，經過處理器(PowerPC)，決定由軟體、硬體或軟硬體共同實作後，將其影像儲存在CF卡上，或利用網路將其影像傳送出去。將來也可以結合手機或PDA，達到即時監視某定點的功能。

二、設計動機

一般傳統數位相機的設計方法，即透過感光耦合元件(CCD)或互補性氧化金屬半導體(CMOS)將外界影像擷取之後，利用SOC(System-on-a-chip)的技術，如在一個積體電路(ASIC)放上數個硬體的矽智產(IP)，透過一個控制器(controller)來控制整個運作的流程。但隨著使用者的需求不斷增加，加上影像處理的技術不斷提升，更複雜處理所需耗費的電量與電路部分提高，因此我們提出一套新的方法，將ASIC替換為現場可規劃閘陣列(Field Programmable Gate Array, FPGA)來因應這些改變。以下為我們所提出的改善方法:

1. 軟硬體切換(HW/SW switch): 在不同情況下軟硬體交錯使用及共同工作(co-work)，提升速度，提高效率，省電。例如: (1)因為使用FPGA較耗電，若在電源不足的情況下，全使用軟體執行，可以節省電。(2)若只要求時間不考慮耗電方面，則可全部使用硬體執行，以達到最快速，最省時。(3)耗電及時間雙方面皆需考量下，則可使用部分軟體，部分硬體共同運作，以達到不同的需求。

2. 動態部分硬體重新組態(Dynamic partial HW reconfiguration): 使用動態部份的切換FPGA上面的電路，已達到**節省空間**，**降低成本**，**省電**等功用。 (1)因為FPGA上面空間有限，當功能較多時，無法一次將所有電路放進FPGA，傳統的方法只能使用邏輯閘(gate count)較大的FPGA，但成本就相對的提高許多。若使用動態部份重新組態，即可在有限邏輯閘的FPGA上設計任何使用者想要的電路。(2)另一方面，若將全部功能燒入到FPGA，因為有些電路可能少用，如此會增加耗電量。但若是將這些少用電路在需使用到時，才動態燒入到FPGA，則會達到省電效果。

三、實作說明

甲. 軟硬體平台簡述

我們就軟硬體分別介紹，其系統實圖(如圖1)與架構圖(如圖 2)所示。

硬體:

1. Xilinx ML 310:做為我們的開發板，使用到的有 FPGA(XC2VP30-FF896)，兩顆處理器(PowerPC)，快閃記憶體卡(CF card)，隨機存取記憶體(RAM)，網路(RJ45)，USB 等週邊。
2. 網路照相機(web camera):當作鏡頭，擷

取影像後，利用 USB 接頭，連接到 ML 310。

軟體:

1. ISE & EDK & ChipScope:為Xilinx公司開發設計FPGA之開發工具。
2. 作業系統: MontaVista Linux先存於CF卡,開啟後再裝載(Load)到RAM上執行。
3. 網路相機驅動程式：修改SourceForge中的一個專案計劃-QuickCam Express driver提供的開放式來源程式碼。

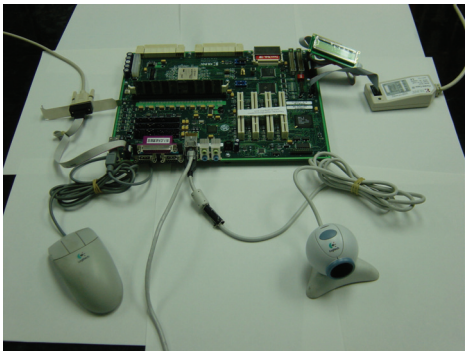

圖 1.系統實圖

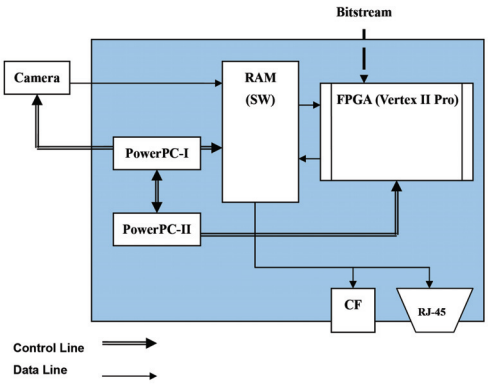

圖 2.系統架構圖

乙. 實作內容

我們從硬體、軟硬體切換與溝通和軟體三個方向個別實作，再將其整合在一起。以下就各部分加以詳細說明。

硬體:

1. FPGA實作流程

a.設計時將影像處理所需要的各種處理以模組的方式建構，並建立函式庫(module library)加以儲存。

b.當收到擷取影像的命令時,傳送命令給網路照像機進行影像擷取的動作，並將擷取到的資料傳送到ML310上。

c.依據所收到的命令要求，比對現有存在於FPGA上的模組後，由PowerPC向模組函式庫要求命令所需要但尚未存在於FPGA上的模組(bitstream)。

d.將由module library所傳回的module (需要但不存在於FPGA內)，經過PowerPC的排程(考慮到空間，效率等其他因素的影響)，依序燒入FPGA內。

e.將影像資料放入FPGA中開始進行影像處理，待結束後將資料傳到RAM中。

f.依據要求，將檔案放入CF card 中或是由網路傳送出去。

2. FPGA實作方法

程式碼:JPEG Hardware Compressor(取自www.opencore.org)

部分重新燒錄之設計(使用模組基礎設計)

a.將整個硬體分成兩個部分(module):固定邏輯(Fixed Logic)和部分重組邏輯(PR Logic).

- 固定邏輯: 通用異步接收發器(UART)控制器， CF卡讀取控制器,，網路傳輸控制器，PS/2控制器，USB控制器。
- 部分重組邏輯 : JPEG壓縮。

b.設計項目:依據部分可重組指導方針(partial reconfiguration guidelines)修改，撰寫，合成硬體描述語言程式(VHDL，Verilog)。

c.初始預訂(Initial Budgeting):設計整個系統的平面規劃(floorplan)，還有對頂層設計(top-level design)及每個模組加上時間限制。

d.實作ISE模組化設計必須流程:

(1) 每個可重組式模組。

(2) 特定可重組模組的結構。

e.合成階段之實作

(1)最低限度(Minimum):完全設計，初始化電源結構(initial power-up configuration)。

(2)建議(Recommended):固定和可重組裝置架構模組每種可能的組合，加以模擬與驗證。

f.驗證設計

- 靜態時間分析
- 功能性模擬

g. 使用FPGA_Editor(ISE)確定FPGA繞線(routing)和擺置(placement)的問題。

h.對整個設計產生bitstream (初始化電源結構)。

i. 針對每個可重組式模組產生個別的bitstream。

j. 燒入ML310(with initial power-up configuration)。

k.依照需求"動態部分燒錄FGPA"。

軟硬體切換與溝通

1. 軟硬體切換與溝通實作流程(如圖3)

a.首先會在初始狀態。

b.軟硬體控制程式會讀取設定檔。

c.依據設定檔，作一系列的軟體，硬體影像處理程序。

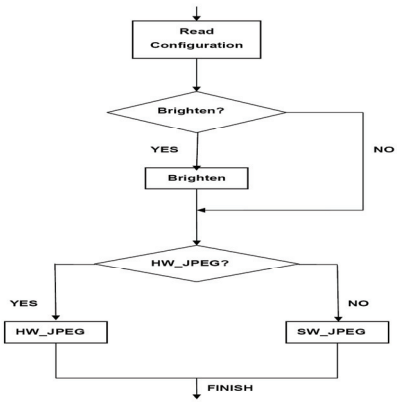

圖3.軟硬體切換流程

這邊可分別四種實作:

實作1: 影像亮化處理(SW) + JPEG壓縮(SW)

實作2: JPEG壓縮(SW)

實作3: 影像亮化處理(SW) + JPEG壓縮(HW)

實作4: JPEG壓縮(HW)

2. 軟硬體切換與溝通實作方法(如圖4)

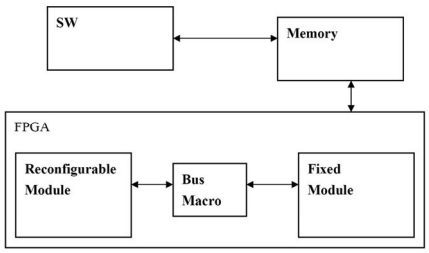

圖4.軟硬體切換與溝通

a.硬體之間溝通:在FPGA裡，硬體模組與硬體模組之間的溝通是透過Bus Macro(如圖5)來達成。

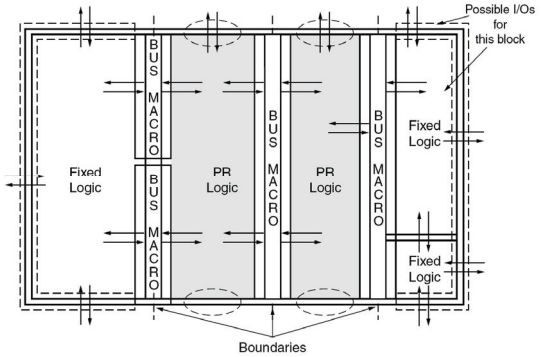

圖5. Bus Macro

b.軟硬體之間溝通:軟體與硬體溝通是透過共享記憶體(share memory)的方式來達成溝通，使用一個flag(預設為0)，當軟體影像處理做完，就去設定flag為1，接著硬體就是去執行JPEG壓縮。

軟體部分

1. 作業系統:在ML310文件中提到，有兩套作業系統－VxWorks和MontaVista Linux確定可以將其放置到目標板上，我們參考美國依利諾州大學Soft System Lab及Wolfgang klingauf所提供文件，將MontaVista Linux放

置到我們的目標板ML310上。

2. 控制程式部分:我們撰寫一個程式，利用滑鼠左右鍵做為影像處理的選擇，並將欲處理影像的方式用一個記錄檔來儲存，再依此記錄檔做相對應的影像處理動作。

實作內容：

一開始時會在初始模式，經使用者選擇下一步要進入哪個模式，“設定模式”(左鍵)或”拍照模式”(右鍵)，流程如圖6所示。

a.設定模式:

(1)首先設定照片是否需經過軟體亮化處理(左鍵表示要，右表否)。

(2)設定JPEG壓縮是由軟體JPEG壓縮或者硬體JPEG壓縮(左鍵表軟體，右鍵表硬體)。

(3)最後將其設定儲存成一個設定檔，回到初始模式。

b.拍照模式: (1)首先讀取設定檔。(2)開始拍照，截取影像。(3)最後依據設定檔，作一系列的影像處理動作。

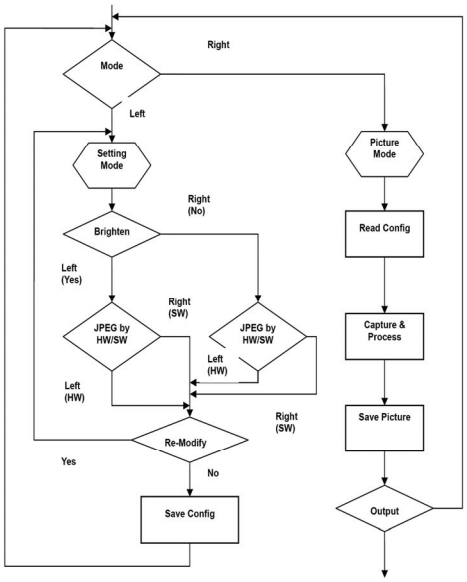

圖6.控制流程

3.亮化與 JPEG 軟體程式碼:流程如圖 7 和圖 8。

圖 7.亮化程序

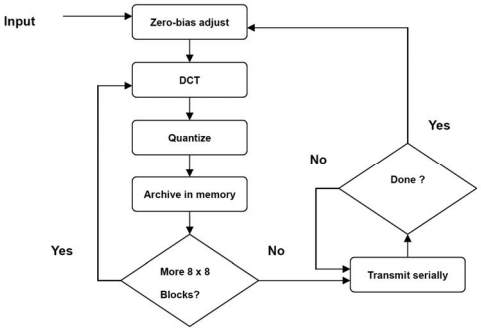

圖 8.JPEG 處理程序

實作結果

實作1	實作2	實作3	實作4
			

	實作1	實作2	實作3	實作4
效能(秒)	9.1	6.1	3.01	1.09
功率(瓦)	0.0033	0.0030	0.170	0.166
大小(開數)	5230	5230	15660	15660
能量(焦耳)	0.0300	0.0183	0.5117	0.1809

由實驗得到的數據跟結果，我們可以很清楚的發現，使用軟體做影像處理，與使用硬體做影像處理，所得到影像是相同的情況下:

(1)當使用硬體做影像處理時，時間相對於軟體做影像處理會小很多(實作3 V.S.實作1＆實作4 V.S.實作2)，所以使用硬體相對於軟體可以明顯達到加速的效果。

(2)當使用軟體做影像處理時，消耗的能量相對於硬體做影像處理會小很多(實作1 V.S.實作3 ＆實作2 V.S.實作4)，所以使用軟體可達到低功率的效果。

四、結論

在嵌入式系統設計與 SOC 設計中，軟硬共同設計是很重要的觀念。如何運用此技巧來達到最佳效能和低功率(low power)是很重要的議題。我們在本次實驗中，實作：(1)軟硬體切換；

(2)動態部份重組，來達到此兩個重要特性。而其中軟硬體需在何時做切換，切換時花費的時間，硬體做動態部份重組的負擔(overhead)所額外消耗的能源(power)，以及重新燒錄多花費時間是否合乎經濟效益，都是值得更進一步探討。

五、參考資料

1. Xilinx, Inc.FPGA Configuration Guidelines, XAPP 090, November 24, 1997 (Version 1.1) .
2. Andrey Filippov Elphel, Inc. Reconfigurable High Resolution Network Camera. Proceedings of the 11th Annual IEEE Symposium on Field-Programmable Custom Computing Machines, pages 276 – 277, April 2003.
3. Christoph Steiger, Herbert Walder, and Marco Platzner. Operating Systems for Reconfigurable Embedded Platforms: Online Scheduling of Real-Time Tasks. IEEE Transactions on Computers, Vol. 53, No. 11, Page(s): 1393- 1407, November 2004.
4. Mesquita, D. Moraes, F. Palma, J. Moller, L. Calazans, N. , Remote and Partial Reconfiguration of FPGAs: Tools and Trends, Proceedings of the Parallel and Distributed Processing Symposium, April 2003.
5. Clemson 大學助理教授 Ron Sass, Xilinx 公司高階語言工具部門資深經理 Don Davis, FPGA 自動化執行階段重設組態研究 http://www.chip123.com.tw/index.php.
6. Matthias Dyer, Christian Plessl, Marco Platzner. Partially Reconfigurable Cores for Xilinx Virtex. 12th International Conference on Field Programmable Logic and Applications (FPL), LNCS 2438, Springer, Montpellier (La Grande-Motte), France, September 2002.
7. J-Y. Mignolet, S. Vernalde, M. Engels. A Low-Power Reconfigurable Internet Appliance. Revue Hf. Electronique, Telecommunications, 2000.
8. Ian Robertson, James Irvine, A Design Flow for Partially Reconfigurable Hardware. ACM Transactions on Embedded Computing Systems, Volume 3, Issue 2, pages: 257 – 283, May 2004.
9. JPEG Hardware Compressor http://www.opencores.org/projects.cgi/web/jpeg/overview
10. Two Flows for Partial Reconfiguration: Module Based or Difference Based.
11. ISE Development System Reference Guide – Chap 4 Modular Design.
12. Xilinx ML300/ML310 Prototyping Boards http://www.klingauf.de/v2p/index.phtml#2003-10-05
13. ML310 User Guide.