

行政院國家科學委員會專題研究計畫 成果報告

晶片系統架構技術及開發平台研究之推動(3/3)

計畫類別：個別型計畫

計畫編號：NSC93-2215-E-002-010-

執行期間：93年08月01日至94年07月31日

執行單位：國立臺灣大學電子工程學研究所

計畫主持人：陳少傑

共同主持人：蔡加春

報告類型：完整報告

處理方式：本計畫可公開查詢

中華民國 95 年 1 月 31 日

目錄

壹、計畫緣由	2
貳、舉辦徵求研究計畫及執行情形	2
參、十個附屬計畫與執行成果	3
1. 晶片系統架構技術及開發平台研究之推動(3/3)-附屬計畫一 適用於 H.264/MPEG-4 AVC 的嵌入式即時多媒體系統之整合設計	
2. 晶片系統架構技術及開發平台研究之推動(3/3)-附屬計畫二 多個影像追蹤代理器之 FPGA 加速硬體模組實作	
3. 晶片系統架構技術及開發平台研究之推動(3/3)-附屬計畫三 使用同步讀—寫演算法以提昇快閃記憶體的大量存取效能	
4. 晶片系統架構技術及開發平台研究之推動(3/3)-附屬計畫四 應用於多媒體電路供電系統核心電路之研究	
5. 晶片系統架構技術及開發平台研究之推動(3/3)-附屬計畫五 使用 FPGA 設計二維封包分類演算法	
6. 晶片系統架構技術及開發平台研究之推動(3/3)-附屬計畫六 SOC 嵌入式系統之電源規劃	
7. 晶片系統架構技術及開發平台研究之推動(3/3)-附屬計畫七 無線感測器網路基礎架構設計與實作 — 智慧型辦公室/家庭應用	
8. 晶片系統架構技術及開發平台研究之推動(3/3)-附屬計畫八 CCD/CMOS 彩色影像插補點晶片設計	
9. 晶片系統架構技術及開發平台研究的推動(3/3)-附屬計畫九 數位相機之影像辨認系統研究	
10. 晶片系統架構技術及開發平台研究之推動(3/3)-附屬計畫十 ARM-based SOC 平台以整合並加速影像擷取處理	
肆、舉辦十五場 SoC 訓練課程	52
伍、舉辦五場 SoC 人才培訓海外講座	53

壹、計畫緣由

我們的工作是結合北部地區（包括中壢桃園、台北縣市、宜蘭、花蓮等地區）之各大學院校（包括綜合大學、科技大學、技職院校），及有興趣在這些領域從事研究的教師學生，共同努力達成下列目標：

1. 協調在 SoC/IP 領域的研究團隊，從事規劃性的主題研究，以期更有效率地研發上述各領域之新的技術。

2. 鼓勵教師學生投入 SoC/IP 設計技術之研究，擴大研究能量之基礎規模，預期投入之教師有 50 位以上，每年畢業的學生達 700 人以上（其中包括取得學士學位投入工作者 300 人，取得碩士投入工作者 300 人，取得博士投入工作者 100 人）。

3. 加強產、研、學溝通與結合，期使學校之研發成果能直接地有助於 SoC/IP 設計產業之技術。並進一步獲得產業界之回饋，以充實學校之研發資源。

在過去一年來，我們舉辦了一場徵求研究計畫說明會、十五場 SOC 訓練課程、及五場 SOC 人才培訓海外講座。

貳、舉辦徵求研究計畫及執行情形

1. 93.06.29 國科會"北區 SOC 系統整合設計技術研究計畫"公開徵求研究計畫說明 (Call for Proposal)
2. 93.08.04 北區 SOC 附屬計畫(經費補助清單)
3. 94.03.11 於台灣大學電機二館 124 會議室討論附屬計畫之執行現況並準備成果展示

參、十個附屬計畫與執行成果

1. 晶片系統架構技術及開發平台研究之推動(3/3)- 附屬計畫一
適用於 H.264/MPEG-4 AVC 的嵌入式即時多媒體系統之整合設計
2. 晶片系統架構技術及開發平台研究之推動(3/3)- 附屬計畫二
多個影像追蹤代理器之 FPGA 加速硬體模組實作
3. 晶片系統架構技術及開發平台研究之推動(3/3)- 附屬計畫三
使用同步讀—寫演算法以提昇快閃記憶體的大量存取效能
4. 晶片系統架構技術及開發平台研究之推動(3/3)- 附屬計畫四
應用於多媒體電路供電系統核心電路之研究
5. 晶片系統架構技術及開發平台研究之推動(3/3)- 附屬計畫五
使用 FPGA 設計二維封包分類演算法
6. 晶片系統架構技術及開發平台研究之推動(3/3)- 附屬計畫六
SOC 嵌入式系統之電源規劃
7. 晶片系統架構技術及開發平台研究之推動(3/3)- 附屬計畫七
無線感測器網路基礎架構設計與實作 — 智慧型辦公室/家庭應用
8. 晶片系統架構技術及開發平台研究之推動(3/3)- 附屬計畫八
CCD/CMOS 彩色影像插補點晶片設計
9. 晶片系統架構技術及開發平台研究的推動(3/3)- 附屬計畫九
數位相機之影像辨認系統研究
10. 晶片系統架構技術及開發平台研究之推動(3/3)- 附屬計畫十
ARM-based SOC 平台以整合並加速影像擷取處理

晶片系統架構技術及開發平台研究之推動(3/3)-附屬計畫一
適用於 H.264/MPEG-4 AVC 的嵌入式即時多媒體系統之整合設計

**Integrated Design of an Embedded Real-Time Multimedia System
for H.264/MPEG-4 AVC**

計畫編號：NSC93-2215-E002-010/001

執行期限：93/8/1~94/7/31

計畫主持人：李宗演 教授 台北科技大學電子系
計畫參與人員：戴吉雄、陳鉉盈、徐昱淳、吳宜穆

一、中文摘要

本研究計畫之目的，在整合應用以 ARM 為基礎的嵌入式多媒體即時作業系統於 H.264/MPEG-4 多媒體上。由於 H.264 是一種因應即時系統(RTOS)所設計的可擴展性串流傳輸(scalability stream media communication)的編碼技術。H.264 主要架構於細粒度可擴展(Fine Granular Scalability, FGS)的壓縮編碼機制。細粒度可擴展壓縮編碼技術是最新 MPEG-4 串流式傳輸標準，能依頻寬的差異來調整傳輸的方式。細粒度擴展縮編碼技術以編入可選擇性的增強層(enhanced layers)於碼中，來提高影像傳輸的品質。本計畫主要在於設計一種簡單有效的即時階層可擴展的影像傳輸系統。在增強層編碼及 H.264 的基本層(base layer)編碼上使用漸進的細粒度可擴展編碼(Progressive Fine Granularity Scalable, PFGS)能直接使用 H.264 的格式特色來實現 FGS。同時加入了 LB-LLF (Layer-Based Least-Laxity-First scheduling algorithm) 的排程演算法，來增進網路傳輸影像的品質。由實驗結果顯示本系統在串流影像品質 PSNR 值上確有較佳的效能。

關鍵詞

即時作業系統、排程、嵌入式系統、多媒體應用

Abstract

With the recent development in stream media, scalable H.264 video encoding is designed to be used in real-time video communications systems. The method is based on the Fine Granular Scalability (FGS) video coding method. It provides a framework to adapt to variations in the channel bandwidth and was recently standardized in the Streaming Video Profile of MPEG-4. In this project we proposed and implemented a simple and effective real-time layer-based scalable video communication system. In this system we use Progressive Fine Granularity Scalable (PFGS) scheme to encode the enhancement layer and H.264 to encode the base layer. This scheme is a direct implementation of FGS onto H.264 and does not employ any H.264-specific features for FGS. We also use the adaptive Layer-Based Least-Laxity-First (LB-LLF) scheduling algorithm to improve the output quality of video on network and to achieve synchronized playback effect. The experimental results show that subjective quality of the video stream is significantly improved using our methods.

Keywords

Real-Time Operating System (RTOS), Scheduling, Embedded System, Multimedia Application.

二、動機與目的

由於大部分的多媒體應用需要處理大量的資料運算。故適應網路傳輸頻寬及儲存容量限制的視訊編碼標準變的較為重要，2001年，ISO/IEC MPEG 及 ITU-T VCEG 決定共同發展能較有效的解決頻寬及儲存空間的視訊編碼標準。於是成立了 JVT (Joint Video Team) 來制定新的標準。在參考 H.26L 等視訊編碼後制定了名為 H.264/MPEG-4 part 10 的新視訊編碼標準，以使其在編解碼及網路傳輸上有更佳效率。H.264/MPEG-4 part 10 這個新興的編碼標準能將 MPEG-2 的串流碼壓縮為原本的一半而保留相當的品質。

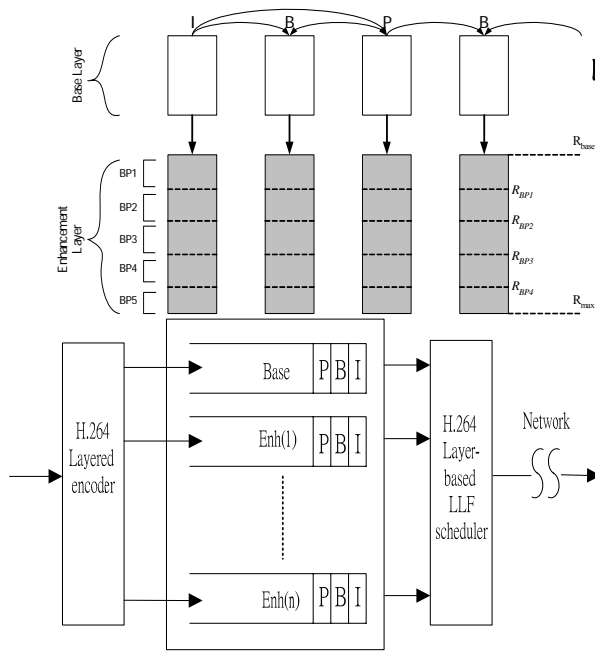
H.264 不同於先前的視訊編碼標準，在傳輸 MPEG-4 的影片上，它能提升約 50% 效能。這使得 H.264 更適用於網路的傳輸應用上。雖然 H.264 有依不同比率切換不同位元串流的特色，但它仍不能適應不同的網路頻寬狀況來做即時的處理。因此若能結合 FGS 及 H.264 等相關技術，則可解決其封包傳輸的問題。在編碼方面 PFGS 主要被使用在增強層，而 H.264 則被使用在基本層。

但由於 H.264 可擴展性的研究介紹並不多，本計畫將直接把 FGS 實踐於 FGS 上而不使用 H.264 內含的 FGS 特性，並使用 LB-LLF (Layer-Based Least-Laxity-First scheduling algorithm) 的排程演算法，以階，來提高此系統於網路上傳輸的品質與效能。

三、研究方法與成果

本計畫的前半段，我們先研究 H.264 的擴展性編碼方式。FGS 是仰賴 MPEG-4 標準的一種技術，藉由參考的幀(frame)。只有基礎層是由最原始的資料串所拆解出的參考 frame 來預測重建，而其他的增強層則由其他增強層參考的 frame 來交互參照建立的。這種利用增強層來細粒度擴展的編碼方式，可以使得影像能適應網路傳輸上通道及頻寬的變動。即當原始資料以細粒度擴展的方式產生內嵌式的位元串流資料後，增強層可以當時的網路狀況來決定接收的資料量。然而，使用移動預測的方式編碼，所以當基礎層的影像品質低落時。即使使用 FGS 也未必比傳統的雜訊比可變編碼(SNR scalable coding)來的有效率。

由於 H.264 的壓縮演算法主要在處理相似度較高的原始資料，也就是說原始資料在時間及空間域的改變上較小。每個輸入的影像資料被處理成 I-frame (intra-frame), P-frame(previous image) 或是 B-frame (bi-directional image) 輸出。而這樣的切割方式能適應網路上的通道解碼及丟包解碼的要求。FGS 的架構在 MPEG-4 的標準裡，以 R_{base} 的位元速率(bit-rate)處理非擴展性基礎層的編碼及最快的位元速率 (R_{max}) 處理增強層的編碼，在傳送時，則依頻寬來決定增強層的位元速率 ($R_{available}$)，而如圖一則是表示 FGS 在 MPEG-4 串流視訊編碼的架構。



圖一 FGS 之結構圖，BP1, BP2,...BP5 是指增強層的 bit-planes。

圖二 PFGS 的串流傳輸系統架構

圖二為本計畫所設計之 PFGS 的串流傳輸系統架構圖。為了使影像在網路上的傳輸有較佳的品質，所以我們在編解碼的傳送端和接收端間增加了基本層(Base layer)和增強層(Enhance layer)緩衝器來處理各層級間的串流資料。當原始影像串流資料被擷取後，經由 H.264 的可層級編碼器編碼分成一個基本層及 n 個增強層資料流入緩衝器。

我們將 PFGS 所編碼後，各層(包含 Base Layer 和 Enhancement Layer)的輸出做可調式 LB-LLF(Layer-Based Least-Laxity-First)之排程，來達到在解碼時，各層品質與效率之間的最佳化。

下列為每個 Task (T)的特性:

$Tx(y)$: 在第 x 個畫面的第 y 階層。

$Ex(y)$: 在 transmission buffer 裡 $Tx(y)$ 準備要到排程之 Ready Buffer 的最後時限。

$Px(y)$: $Tx(y)$ 的整個過程的時限。

$D(x)$: 在第 x 個畫面之所有 T 要傳送出的

去的最晚傳送時間。

$Lx(y)$: $Tx(y)$ 的 laxity 值。

$$Lx(y) = D(x) - Px(y)$$

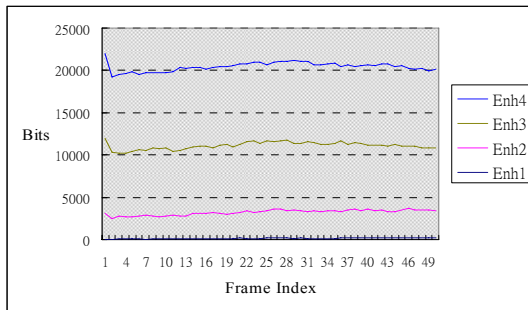
$LTx(y)$: 每個單位時間 $Tx(y)$ 的 laxity 值。

可調式 LB-LLF 演算法

- Step 1: let R = frame set of ready packets with the deadline in the server transmission buffers
- Step 2: compare the current time t_{cur} with the deadline $D(x)$ of all the packets in R
If $t_{cur} > D(x)$, remove the packet from R
- Step 3: calculate the least-laxity value $Lx(y)$ for each set from set = 1 to N (where N is the buffer size in server)
- Step 4: select the smallest least-laxity value $LTx(y)$ packets from R
- Step 5: send out the set of packets, and let $t_{cur} = t_{cur} + Px(y)$
- Step 6: Go to Step 1

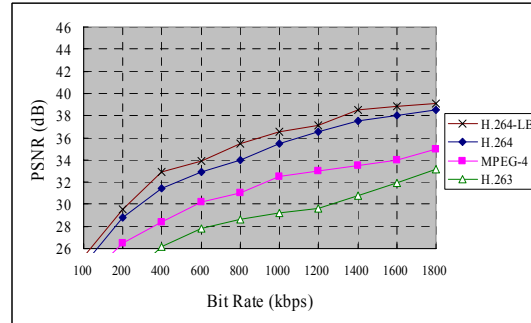
在接收端，我們使用了分派器程式 (Dispatcher) 將網路接收到的串流資料導入接收端各對應之基本層和增強層緩衝器，再由 H.264 層級解碼器解碼輸出影像信號。

圖三為我們編碼後的 4 階增強層的位元數。所有層次的綜合恢復出原來的視頻序列，因此能夠提供的視頻品質從基本層 (最差) 到所有層的綜合 (最好)。增強層是有級別的，不同級別對應不同的重要性。一般最低要求是至少保證基本層，依次按層次級別遞增，因為如果缺少下面的任何一層，在上面的增強層就沒有任何作用。



圖三 4 階增強層的位元數

為了驗證本研究計畫所提出的系統架構及演算法，我們使用 JVT (Joint Video Team) JM 9.0 encoder/decoder 來模擬實驗。圖四的實驗數據是以每秒 25 個 frame 的 Foreman QCIF 格式以不同的變動位元速率與 H.263、MPEG-4、H.264 做 PSNR 值影像效能比較。由數據顯示，本系統及演算法比 H.263、MPEG-4 均高出約 2~4dB。比無階層的 H.264 影像傳輸亦較略高 1dB。



圖四 變動位元速率的 PSNR 測量值

四、結論

在這計畫裡我們提出並且實現一個完整和有效的 H.264 即時層級式可擴展的視訊通信系統。我們使用 PFGS 編碼增強層和 H.264 編碼基礎層。本研究同時加入了 LB-LLF (Layer-Based Least-Laxity- First scheduling algorithm) 的排程演算法，來增進網路傳輸影像的品質。

五、參考文獻

- [1] Y. He, F. Wu, S. Li, Y. Zhong, and S. Yang, "H.26L-based Fine Granularity Scalable Video Coding," in Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS), Scottsdale, Arizona, USA, 26-29 May 2002, vol. 4, pp. 548-551.
- [2] Wang, Q., Wu, F., Li, S., Zhong, Y., and Zhang, Y., (2001) 'Fine-granularity spatially scalable video coding', Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, 2001, (ICASSP '01), vol.3, pp.1801-1804.
- [3] Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264/ISO/IEC 14 496-10 AVC), Mar. 2003.
- [4] A. Tamhankar and K. R. Rao, "An overview of H.264/MPEG-4 Part 10," The 4th EURASIP Conference focused on Video/Image Processing and Multimedia Communications, Vol. 1, pp.1-51 July 2003.
- [5] Gharsalli, F.; Baghdadi, A.; Bonaciu, M.; Majauskas, G.; Cesario, W.; Jerraya, A.A., "An efficient architecture for the implementation of message passing programming model on massive multiprocessor," Proceedings of the 15th IEEE International Workshop on Rapid System Prototyping, 2004. 28-30 June 2004,

- pp. 80 – 87.
- [6] Chia-Ying Tseng, Liang-Teh Lee, Yu-Lan Shih, and Kang-Yuan Liu, "Adaptive Layer-Based Scheduling for Real-Time Transmission on Scalable Multimedia Stream," Proceedings of the Tenth International Conference on Distributed Multimedia Systems (DMS'2004), Sep. 2004, pp. 389-392.
 - [7] K. Ugur and P. Nasiopoulos, "Design Issues and a Proposal for H.264-based FGS," contribution MPEG03/M9505, ISO/IEC JTC/SC29/WG11, Pattaya, Thailand, March 2003.
 - [8] Gao, K., Gao, W., He, S., Gao, P., and Zhang, Y., (2003) 'Real-Time Scheduling on scalable media stream delivery', Proceedings of the 2003 International Symposium on Circuits and Systems, 2003(ISCAS '03), vol.2, pp. II-824 - II-827.
 - [9] Gao, K., Zhang, Y., Gao, W., and He, S., (2003) 'Real-Time Scheduling Supporting VCR Functionality For Scalable Video Streaming', 14th IEEE Proceedings on Personal, Indoor and Mobile Radio Communications, 2003(PIMRC 2003), vol.3, pp.2711–2715.
 - [10] Hildebrandt, J., Golatowski, F., and Timmermann, D., (1999) 'Scheduling Coprocessor for Enhanced Least-Laxity-First Scheduling in Hard Real-Time Systems', Proceedings of the 11th Euromicro Conference on Real-Time Systems, June 1999, pp.208-215.
 - [11] Oh, S.H. and Yang, S.M., (1998) 'A Modified Least-Laxity-First Scheduling Algorithm for Real-Time Tasks', Proceedings of the 5th International Conference on Real-Time Computing Systems and Applications, Hiroshima, Japan, 1998, pp. 31-36.

晶片系統架構技術及開發平台研究之推動(3/3)-附屬計畫二： 多個影像追蹤代理器之 FPGA 加速硬體模組實作

計畫編號：NSC 93-2215-E-002 -010/002

執行期限：93 年 8 月 1 日至 94 年 7 月 31 日

主持人：簡忠漢 聖約翰技術學院電機工程系

計畫參與人員：侯承宏、黃耀璋、賴建宏、吳添寶

一、摘要

本計畫中我們針對多個影像追蹤代理器共同合作完成整體目標任務之應用，探討以 FPGA 硬體模組實作其影像處理演算法加速其運算速度。我們特別針對影像追蹤演算法中處理最為耗時的部分，例如處理影像邊緣輪廓的模組，採用 VHDL 硬體描述語言來實現此類模組。我們使用 Quartus II 與 Modelsim 軟體做為模組編譯與模擬之開發環境，至於 FPGA 晶片則選用 Altera Nios 的 Cyclone EP1C20F4007C 晶片。並自行設計資料傳輸模組，先以電腦傳輸影像資訊至 FPGA 晶片，而將 FPGA 處理後之影像傳回電腦顯示做驗證比對。

關鍵詞：影像追蹤、FPGA、代理器

二、緣由與目的

近年來結合網際網路、通訊技術、嵌入式系統與控制理論之網路化整合控制已是時勢所趨。例如資訊網路、交通運輸系統、工業生產與製造系統、家庭與辦公室自動化以及微小動態型機電系統等皆為現今之重要的大規模、分散式工程系統。在這些系統中的共同特色是它們皆具備有數量龐大的個別的個體元件，例如：移動式載具、工作站等，稱之為代理器。本計畫著重於發展代理器的影像感測能力，與整合多個代理器間的影像感測資訊做為互動之依據。藉由加入影像資訊的感測，可使代理器處理具智慧型之工作。例如：

- (1) 由影像資訊辨識代理器與目標物，並計算其位置資訊以供路徑規劃之用。
- (2) 透過影像伺服驅動代理器追蹤移動的目標物或追隨同伴。

(3) 結合影像與其他感測資訊完成多個代理器集合編組隊形。

(4) 藉由影像資訊可智慧型地達成群組整體規避繞過障礙物之功能。

近十年來國內外已有相當多的文獻，研究利用影像感測資訊為基礎，以進行目標物感測追蹤[1]~[21]，並提出許多影像追蹤演算法。本計畫中我們特別考慮多個影像追蹤代理器共同合作完成整體目標任務之應用。應用的範例有 RobotCup 聯盟推動之機器人足球賽，期望在 2050 年組成一機器人足球隊，能打敗世界盃足球賽之冠軍隊伍。又如 Carnegie Mello University 與 David Sarnoff Research Center[22]聯合嘗試結合多個影像偵測器與機動性地面與空中載具，監控戰場上敵軍的移動情形。

上述應用中需整合多個代理器間的影像感測資訊做為互動之依據。然而隨著影像處理的尺寸、解析度、與資料量不斷地增加，純粹以軟體程式執行影像處理與傳輸工作，已明顯效能不足，往往無法達到即時系統之要求。因此必須藉助專屬之處理硬體，來加速影像處理。

近幾年開始有文獻採用可重組化運算架構 (Reconfigurable Computing Architecture) 設計影像處理的硬體模組 [23][24]，因此本計畫採用具可重組計算功能之可程式系統晶片(SOPC)，發展加速影像追蹤處理與傳輸之硬體 IP 模組，並於 FPGA 晶片實做與驗證。利用其具備應用快速離型的能力讓系統的驗證與模擬除錯容易達成。

三、研究方法與成果

3.1 影像處理方法

在影像處理中，常藉助空間濾波器過濾雜訊或是強化訊號。例如高通濾波器 (high-pass filter) 可用於強調影像的邊緣及細微複雜的部份，因此適合在邊緣化影像時使用。低通濾波器 (low-pass filter) 能把影像高頻雜訊部份去除，留下較柔和之影像。以下我們介紹本計畫實現影像追蹤演算法常用之影像處理方法。

(1)、中值濾波器：

假設一原始影像以下圖 1 之矩陣表示，假設其中包含二個雜訊(像素值為 100 及 110)。

21	23	22	24	25
32	33	11	100	12
110	31	22	24	22
10	35	25	26	27
13	32	24	25	26

圖 1. 原始影像灰階值

以(2, 2)點(像素值 33)為例進行中值濾波處理，如圖 2，以(2, 2)為中心點，形成 3x3 矩陣。其值由小到大排列分別為：

11, 21, 22, 22, 31, 32, 33, 33, 110

21	33	22	24	25
32	33	11	100	12
110	31	22	24	22
10	35	25	26	27
13	32	24	25	26

圖 2. 中值濾波之 3x3 像素區域範例

其「中間值」為「31」，因此以 31 為(2, 2)點的新像素值，取代原先像素值 33。依此類推，原始影像的每一點均進行中值濾波處理，可得圖 5 之新影像，其中可發現雜訊已被消除。

0	21	12	22	0
31	31	24	24	22
31	31	26	24	22
13	25	25	25	24
0	13	25	25	0

圖3. 執行中值濾波後之影像灰階值

(2)、迴旋運算(convolution)：

迴旋運算是 DSP 系統最常應用的方法，以一特定之係數遮罩(mask)，在每一

像素上與其鄰近區域之像素，進行乘加運算。像素內的灰階值較受鄰近點的影響，故而遮罩大小以 3x3 為主，如圖 4 所示。一個 3x3 遮罩共有 9 個係數，即 m1 至 m9。為了計算原影像的一個像素點灰度值，此遮罩的中心點(m5)。經一連串的乘法和加法運算所得的結果即為該像素點的取代值。

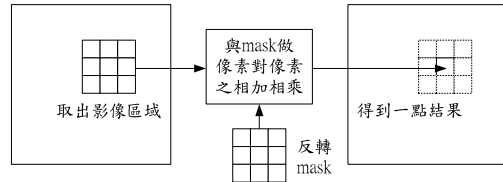


圖 4. 迴旋運算計算方式

(3)、影像邊緣化：

在數位影像處理中，邊緣偵測是相當重要的，其目的在能有效且準確地偵測到目標物邊緣。凡欲得知物體在影像中之形狀、面積、分布範圍、個數等資訊。必須先對影像邊緣化處理，處理後不但上述物體的特性不會消失，反而能更清楚的了解影像的資訊。常用之邊緣偵測方法包括 Laplacian 濾波器、Sobel 濾波器等。

Laplacian 濾波器亦採用標準的迴旋運算。其對高頻訊號的高度敏感，所以應用 Laplacian 濾波器前，應先進行平滑化濾波(如高斯平滑濾波)。

Sobel 濾波器使用空間一階導數來增強高頻的空間訊號，在影像中，這些高頻訊號通常代表較銳利的物體邊緣或線條特徵。假如我們需要的影像資訊不多，以邊緣化後的影像來代替原始影像是一個很好的選擇。

3.2 影像處理模組實做

以下介紹本計畫實做上述之影像處理 IP 模組之方法。我們主要基於參考文獻[23]之架構，其採用 VHDL 硬體描述語言來實現此類模組。我們使用 Quartus II 與 Modelsim 軟體做為模組編譯與模擬之開發環境，至於驗證之 FPGA 晶片則選用 Altera Nios 的 Cyclone EP1C20F4007C 晶片。

(1) window_3x3 模組：影像遮罩的大小通常是 3*3，因此我們採用 3*3 遮罩來做影像處理。首先我們採用 Altera Mega Function 實現 fifo_128x8u 影像資料緩衝模組，並提

供 window_3x3 模組建立出一個 3*3 矩陣 w11, w12, w13, w21, w22, w23, w31, w32, w33 的資料，其架構如下圖：

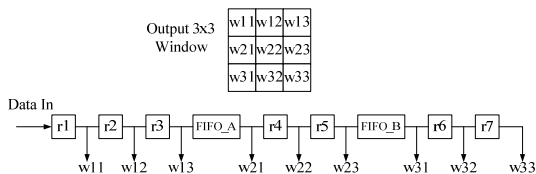


圖 5. window_3x3 模組之架構

(2) rc_counter 模組：本計畫中我們預設處理的影像大小為 128*128 像素。所以必須在每一行與每一列做計數，目標是用來告知系統已經掃描完某一行或某一列，必須做換行或換列的動作，其掃描順序如下圖：

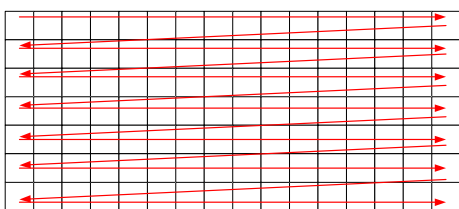


圖 6. rc_counter 模組之掃描示意圖

(3) sort_3x3 排序模組：中值濾波器是取 3*3 矩陣之中間值，因此需採用排序模組將 w11、w12、w13、w21、w22、w23、w31、w32、w33 等數值由大到小的依序排列，其架構如下圖：

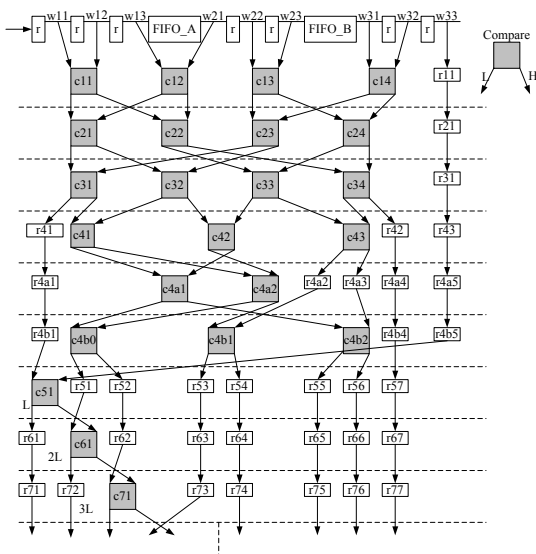


圖 7. sort_3x3 排序模組之架構圖

(4) ro_filt_3x3 模組：此模組實現中值濾波器，其採用 window_3x3、sort_3x3、rc_counter 這三個模組所演算出來的資料，並取出 sort 排序模組所建立的 order=4

的數值做輸出，此即為中值濾波處理，其架構如下圖：

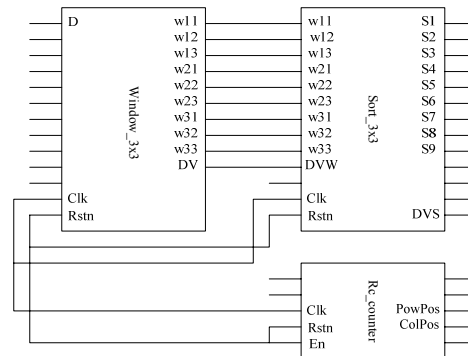


圖 8. ro_filt_3x3 模組實現中值濾波器

(5) conv_3x3 模組：以下模組實現迴旋運算，其中 k0, k1, k2, k3, k4, k5, k6, k7, k8 即是遮罩係數。利用不同的遮罩係數可實現 Laplacian 濾波器、Sobel 濾波器。

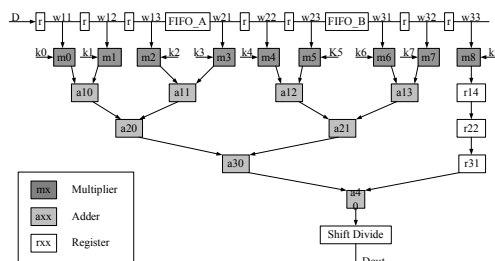


圖 9. conv_3x3 模組實現迴旋運算

3.3 模擬與實驗結果

以下我們將實作模擬上述模組。首先如附圖 1 所示，我們以假想的影像資料序列 0, 3, 6, 9, ... (3*k mod 128, k=0, 1, 2, 3...)，輸入至 window_3x3 模組，經模擬後成功建立出一個 3*3 矩陣 w11, w12, w13, w21, w22, w23, w31, w32, w33 的資料。附圖 2 為 sort_3x3 排序模組之模擬圖，其中根據 window_3x3 模組建立的 3*3 矩陣資料，成功依其大小排序(如 s1~s9)。附圖 3 為 ro_filt_3x3 模組之模擬圖，其以 sort_3x3 排序模組排序之資料取出指定順位的資料(dout)，例如中值之順位為 4。附圖 4 為 conv_3x3 模組實現迴旋運算之模擬結果。

實作上述模組之後，我們進一步以實際影像測試，並採用以下之測試驗證流程。

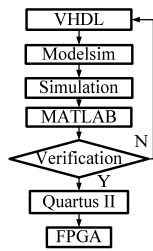


圖 10. 影像處理模組驗證測試流程

首先將設計好的 VHDL 程式，經由 Modelsim 進行編譯。正確無誤後，接著進行模擬測試驗證，我們利用 Testbench 來做驗證。Testbench 主要是用來協助程式撰寫者驗證程式的正確與可靠性。Testbench 可使輸出與輸入訊號的向量資料文件化，使其更易保存與維護。且為具系統性之方法，並非波形檢查，因此可降低人為錯誤。並可重複使用，縮短設計驗證時間，具可流通性，攜帶性極高。

在 Testbench 中我們利用 VHDL 的 TEXT/IO 函式庫讀取輸入影像資料，並將影像處理的結果寫入外部檔案。然而 TEXT/IO 的函式庫只可以讀寫文字格式的檔案，無法讀取二進位檔，但一般的圖像檔案除包含標頭外，其圖像的資料皆以二進位格式儲存，故需額外撰寫程式將標頭濾除並將二進位格式圖像資料轉換成文字格式。

透過 TEXT/IO 函式庫可讀寫純文字格式將模擬驗證之結果資料輸出至一個檔案儲存，以方便進一步以 Matlab 做驗證比對，若影像處理結果不正確時，便重複修改與測試，直至結果正確後，再進行燒錄至 FPGA 晶片上進行實際的驗證。圖 11 為實際影像處理模組模擬流程結果範例。



圖 11. 影像處理模擬流程範例

根據上述驗證流程，我們針對中值濾波器、Laplacian 濾波器、Sobel 濾波器進行驗證，結果如圖 12 所示。

經過上述一連串的模擬、驗證後，可證實我們所使用的 VHDL 影像處理模組正確無誤，能將影像資訊依照我們所設定的規則、條件，處理成我們所需要的特徵影像。

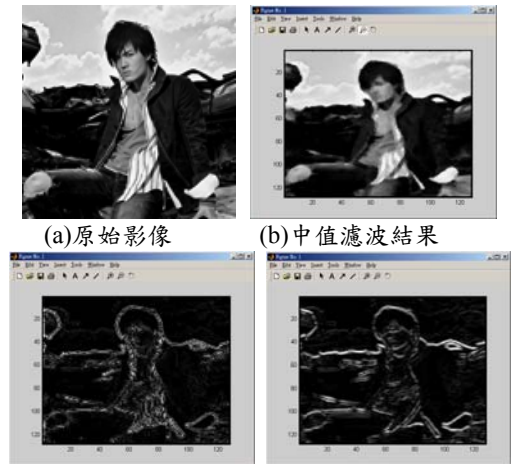


圖 12. 影像處理驗證結果

接下來就是要將此模組下載至 FPGA 上，用來實現以硬體加速影像處理。因此接下來我們使用 Quartus II 將電路載入至 FPGA。在硬體實際測試，我們使用電腦傳遞影像資料至 FPGA，再讀取由 FPGA 處理過的影像資訊傳回電腦比對，以驗證硬體下載的正確性，其設計方法如下：

(1) 資料傳遞的規劃：

我們選擇使用 RS232 串列輸入影像資訊，而經由 FPGA 處理過的影像資訊則由 Print Port 並列讀取回電腦。由於影像處理模組輸入為 8 Bit 灰階影像資料，輸出則為經乘加運算後之 17 Bit 資料。因此考慮資料傳遞的方便性，我們捨棄了最高位元的資料(通常為 0)，只以 Print Port 以每次 8 Bit 分兩次讀回的方式讀取 16 Bit 資料。其架構如下圖：

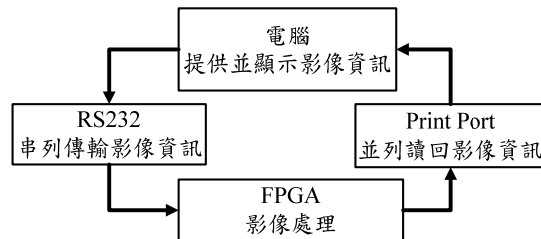


圖 12. 影像傳輸與處理流程

(2) 設計資料傳輸模組：

由於需要使用到 RS232 串列傳輸接收資料，因此，我們設計以 Maxplus II 設計 UART 的接收模組；而傳回電腦的資料需分為 2 次、一次 8 Bit，分別為高 8 Bit 與低 8 Bit 的方式傳回，因此我們採用多工器，由電腦控制選擇線，使其能正確的收到完整的 16 Bit 的資料。其架構如下圖：

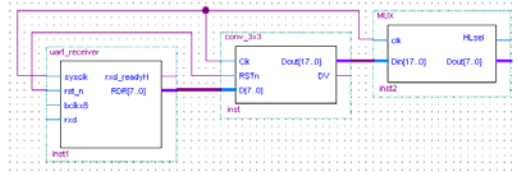


圖 13. 以 Maxplus II 設計資料傳輸模組

(3) FPGA 下載規劃：

我們使用的 FPGA 為 Altera Nios 的 Cyclone EP1C20F4007C 晶片，其外觀如下：

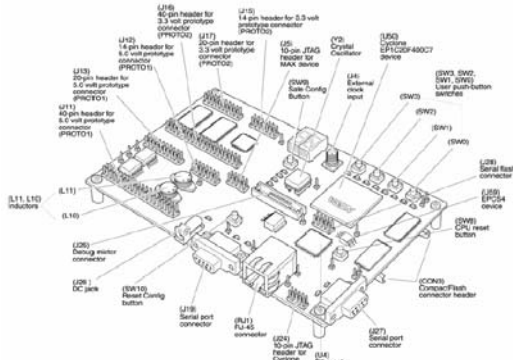


圖 14. Altera 之 Cyclone 發展板

我們直接使用它所提供的 Serial port connector (J19) 來接收電腦端的影像資訊，而在 J12 的 14-pin header 接上 DB25 的連接頭用以和電腦的 Print Port 連結。

(4) 電腦端資料傳輸軟體撰寫：

我們使用了 Visual C++ 撰寫了一段資料傳輸用的程式，用以控制 RS232 與 Print Port 與 FPGA 做資料的傳遞。

四、結論

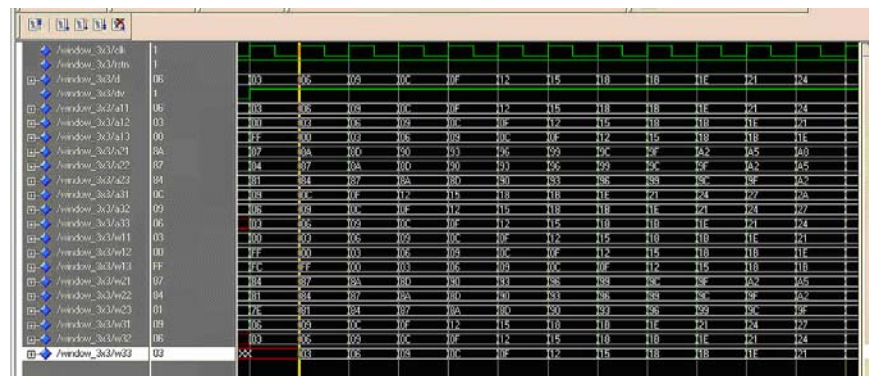
本計畫中我們針對多個影像追蹤代理器共同合作完成整體目標任務之應用，探討以 FPGA 硬體模組實作其影像處理演算法加速其運算速度。我們特別針對影像追蹤演算法中處理最為耗時的部分，例如處理影像邊緣輪廓的模組，採用 VHDL 硬體描述語言來實現此類模組。我們使用 Quartus II 與 Modelsim 軟體做為模組編譯

與模擬之開發環境，至於 FPGA 晶片則選用 Altera Nios 的 Cyclone EP1C20F4007C 晶片。實驗初期先以電腦傳輸影像資訊至 FPGA 晶片，而將 FPGA 處理後之影像傳回電腦顯示以作驗證比對。未來將自行設計影像擷取模組直接使用 FPGA 擷取 CMOS 影像資訊，並經 FPGA 內影像處理模組處理後，將影像資訊送回電腦，或直接輸出至 LCD 顯示器顯示，以加速影像處理速度。

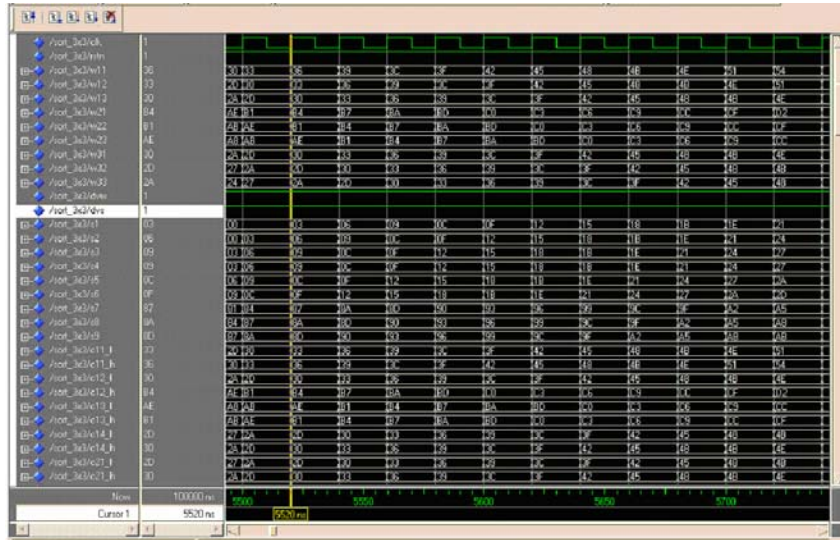
五、參考文獻

- [1] S. Hutchinson, G. Hager, and P. Croke, "A Tutorial on Visual Servo Control", IEEE Trans. on Robotics and Automation, Vol. 12, No. 5, 1996, pp. 651~670.
- [2] William J. W., Carol C. W. H., and Graham S. B., "Relative End-Effector Control Using Cartesian Position Based Visual Servoing", IEEE Trans. on Robotics and Automation, Vol. 12, No. 5, Oct. 1996, pp. 684~696.
- [3] N. Papanikolopoulos and P. Khosla, "Adaptive Robotic Visual Tracking", Proc. American Control Conf., June 1991, pp. 962~967.
- [4] N. Papanikolopoulos, P. Khosla, and T. Kanade, "Visual Tracking of a Moving Target by a Camera Mounted on a Robot", IEEE Trans. on Robotics and Automation, Vol. 9, No. 1, 1993, pp. 14~35.
- [5] N. Papanikolopoulos and P. Khosla, "Adaptive Robotic Visual Tracking Theory and Experiments", IEEE Trans. On Automatic Control, Vol. 38, No. 3, 1993, pp. 429~445.
- [6] N. Papanikolopoulos, B. Nelson, and P. Khosla, "Six Degree-of-Freedom Hand/Eye Visual Tracking with Uncertain Parameters", IEEE Trans. on Robotics and Automation, Vol.11, No. 5, 1995, pp. 725~732.
- [7] Kok-Ming Lee, "Real-time Vision-based Tracking Control of an Unmanned Vehicle", Mechatronics, Dec. 1995.
- [8] Ricardo C., Oscar H. N., and Benjamin K., "Adaptive Robot Control with Visual Feedback", Proc. Of American Control Conf., Maryland, Jun. 1994, TM8-3:10.
- [9] Oscar H. N., Ricardo C., and Benjamin K., "Tracking Adaptive Control of Robots with Visual Feedback", Proc. of the 13th IFAC World Congress, San Francisco, USA, 1996, pp. 265~270.
- [10] K. Hashimoto and T. Noritsugu, "Observer-Based Control for Visual Servoing", Proc. of the 13th IFAC World Congress, San Francisco, 1996, pp. 453~458.
- [11] K. Hashimoto, T. Ebine, and H. Kimura, "Visual Servoing with Hand-Eye Manipulator-Optimal Control Approach", IEEE Trans. on Robotics and Automation, Vol. 12, No. 5, 1996, pp. 766~774.

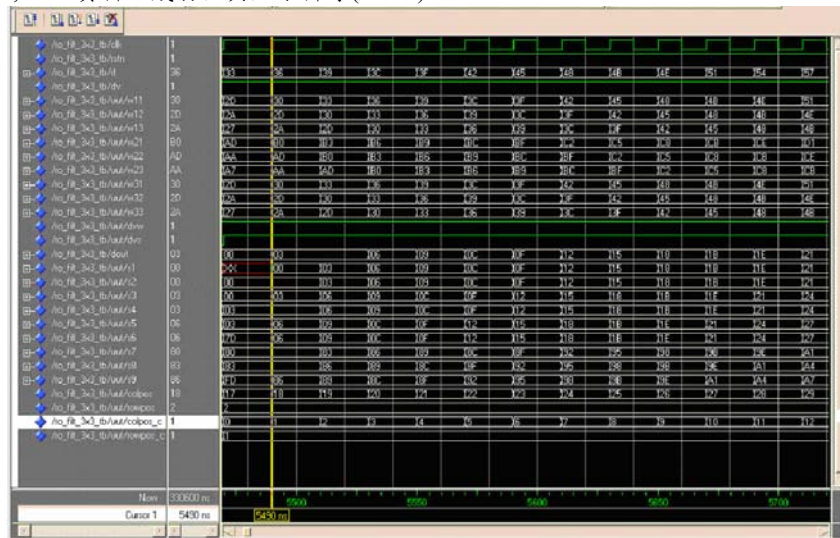
- [12] Hing Il Suh, "Visual Servoing Algorithm Using Fuzzy Logics and Fuzzy-neural Networks", Proc. IEEE International Conference on Robotics and Automation, Vol.4, 1996.
- [13] J. T. Feddema, and C. S. G. Lee, "Adaptive Image Feature Prediction and Control for Visual Tracking with a Hand-Eye Coordinated Camera", IEEE Trans. On Systems, Man, and Cybernetics, Vol. 20, No. 5, Sep. 1990, pp. 1172~1183.
- [14] Keqiang Deng, "Visual-tracking-based Robot Vision System", SPIE-The International Society for Optical Engineering Boston, MA, USA, Nov. 18~20, 1992.
- [15] H. Tu and Li-Chen Fu, "Visual Tracking by Image Moment Flow of Rigid Planar Patches", Proc. R.O.C. Automatic Control Conf., 1995.
- [16] H. Tu and Li-Chen Fu, "Application of Image Moment Flow of a RPP to 6 DOF Visual Tracking", Proc. IEEE, Conf. On Decision and Control, 1995.
- [17] Chien-Hsiang Chen, "A System for 3D Target Trajectory Detection via Stereo Visual Tracking", Department of Electrical Engineering of National Taiwan University, Director : Prof. Li-Chen Fu, 1997.
- [18] H. Tu, Chien-Hsiang Chen, Chih-Yu Chen, and Li-Chen Fu, "A System for 3D Target Trajectory Detection via Stereo Visual Tracking", IEEE Trans. on Control System Tech., 1997.(revised)
- [19] Chih-Yu Chen, "5D Target Trajectory Detection via Intelligent Monocular Visual Tracking System in Real-Time with Air-Target Orientation Recognition", Department of Electrical Engineering of National Taiwan University, Director : Prof. Li-Chen Fu, 1998. (12th Dragon Thesis Award)
- [20] D. Liu and L. C. Fu, "Target Tracking in an Environment of Nearly Stationary and Biased Clutter," IEEE International Conference on Intelligent Robots and Systems, Vol. 3, pp. 1358-1363, 2001.
- [21] T. Kuo, "A Robust Visual Servo Based Headtracker with Auto-Zooming in Cluttered Environment," Master Thesis, Dept. of Electrical Eng., National Taiwan University, 2002.
- [22] T. Kanade, R. T. Collins, A. Lipton, P. Anandan, P. Burt, and L. Wixson. "Cooperative multisensor video surveillance." In Proc. of the DARPA Image Understanding Workshop, pages 3--10, 1997
- [23] A. E. Nelson, "Implementation of Image Processing Algorithm on FPGA Hardware", Master Thesis, Vanderbilt University.
- [24] Steffen Klupsch Markus, "Real Time Image Processing based on Reconfigurable Hardware Acceleration"



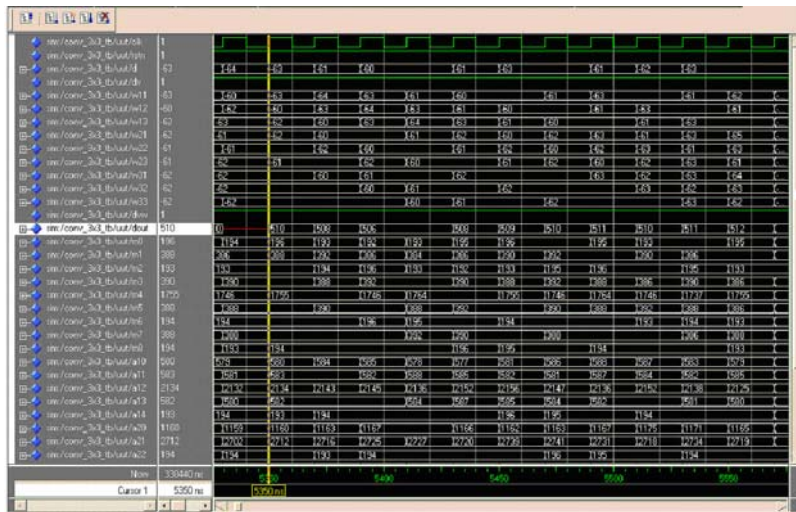
附圖 1. window_3x3 模組之模擬圖，我們輸入假想的影像資料序列 $3*k \bmod 128, k=0, 1, 2, 3, \dots$ ，經模擬後成功建立出一個 $3*3$ 矩陣 $w_{11}, w_{12}, w_{13}, w_{21}, w_{22}, w_{23}, w_{31}, w_{32}, w_{33}$ 的資料。



附圖 2. sort_3x3 排序模組之模擬圖，我們以 window_3x3 模組建立的 3*3 矩陣 w11, w12, w13, w21, w22, w23, w31, w32, w33 資料，成功依其大小排序(s1~s9)。



附圖 3. ro_filt_3x3 模組之模擬圖，其以 sort_3x3 排序模組排序之資料取出指定順位的資料(dout)，例如中值之順位為 4。



附圖 4. conv_3x3 模組實現迴旋運算之模擬圖。

晶片系統架構技術及開發平台研究之推動(3/3) - 附屬計畫三 使用同步讀—寫演算法以提昇快閃記憶體的大量存取效能

Using current Readers-Writers algorithm to improve performance of massive access on flash memory

計畫編號：NSC93-2215-E-002-010/003

執行期限：93/8/1~94/7/31

計畫主持人：吳家榮 講師 龍華科技大學電子系

計畫參與人員：陳怡君、吳端儒

一、中文摘要

由於快閃記憶體(Flash Memory)單位面積儲存大幅提昇,近年來已經廣泛使用於各種消費電子產品上[1-2]。本計畫主要著重在快閃記憶體的存取效能之評估,希望藉由此方面的探討可對於嵌入式作業系統有一個可評比的依據,使往後在開發系統時可做一個整體性的評估,進而挑選出最適切的作業系統平台與開發環境。本計畫內容分為四個部分:

- (1). 一般而言,作業系統的功能愈強大其整體的效能應該愈差,但是經過本計畫的試驗發現並非如此。
- (2). 以程式語言的觀點而言,編譯式的(Compiler)的程式語言通常其執行效能遠勝於以直譯式(Interpreter)程式語言所撰寫的程式,業經我們的實驗發現卻與使用的作業系統有關。
- (3). 依照作業系統的理論觀點而言[16][18],多執行緒處理其執行效能會比循序處理執行效能佳,經過本計畫的實驗發現卻與使用的作業系統有關。
- (4). 探討現有的同步處理程序並開發出適合於快閃記憶體的同步讀—寫(Concurrent Readers-Writers)的演算法,以提昇大量存取之效能。

關鍵詞: 快閃記憶體、同步讀—寫演算法、多執行緒。

● Abstract

In last decade, flash memory is found in a variety of common electronic devices (such as consumer electronics) due to high-capacity flash memory have made them possible [1-2]. In our project is focus on serving the performance of access of flash memory, operating systems. How to choice the platform and programming languages to develop embedded systems and obtain the better performance. The proposed project is divided into four parts:

- (1) In general cases, more functionality of operation systems are less performance for total system, after our experiment we find that are not always true.
- (2) From the view of programming language, the performance of program of compiler language is always better than interpreter language, after our testing the result is dependent on operating systems.
- (3) From the points of view of operating systems[16][18], the performance of multi-thread process is better than sequential process, but this result is dependent on operating system.

(4) We will modify current Readers-Writers algorithm to gain better performance for accessing massive video-audio data on flash memory.

Keywords: flash memory, current Readers-Writers algorithm, multi-threads.

二、動機與目的

近兩三年來製造個人電腦的微利時代已經來臨，因此生產高附加價值的電子產品勢在必行，現今市場上流行的產品，如：MP3-播放器、隨身碟、數位像機、數位攝影機、PDA 或功能完備的手機等，而這些電子產品的共同趨勢—使用快閃記憶體的量大幅增加，為使快閃記憶體存放資料時正確無誤[3][4]，並且提昇大量儲存影音資料的效能，以及增長期使用壽命為本計畫最主要的探討目的。

大部分的電子產品為了考量到整體的成本，通常所使用的快閃記憶體在出廠前已經存在局部是不良品，對於無法正常儲存的記憶體，生產廠只是將所屬的區段(Block(s))作一個標記；表示故障。從出廠之後，整個區段再也不使用。再者，快閃記憶體的可寫入動作都存在有限次數的現象(平均約在十萬次以上)，而一般的電子產品通常是忽略這樣的現象，照常其寫入的動作，因而使得部份資料是不正確的。大量儲存影音資料是目前這些電子產品的特色之一，在單一處理機(Single-processor)的環境下，在不增加成本的考量下如何以較高的效能，更是這類產品在市場上最大的競爭利機。

在軟體的範疇中，舉凡作業系統、資料結構或演算法等各學門[5-18]，所論述的演算法皆以適用於一般主記憶體為主。

自動監控或自動處理的一些系統在早

期又稱之為 SOB (System-on -Board)，在此其間作業系統所佔據的份量幾乎微乎其微，而這些系統為達成相關功能大多是設計者自行撰寫程式。現今的 SOC(System-on-Chip) 嵌入式系統 (Embedded System) 由於體積小與低功耗等特性，使得市場的使用量大大提高，而這些系統由於功能愈趨複雜，與開發環境的配合，使作業系統扮演和參與的份量也愈來愈多，目前用於開發嵌入式系統的作業系統主要有：Win CE, uC/OS-II, uCLinux, ...等作業系統。

以現今的作業系統理論上而言，為提升執行效能常使用多執行緒處理 (Processes) 的方式撰寫程式，因為多執行緒的程式碼可以共用對於整個系統的資源 (Resources) 需求較低，所以效能較佳。而嵌入式系統的開發作業系統是以 uCLinux 系列為主，此系列的作業系統並不能真正支援多執行緒，換句話說這些作業系統表面上可以接受多執行緒的執行，但實際上每一個執行緒仍然以循序的方式執行，所以並不能利用到多元程式 (Multiprogramming) 的好處，也就是說 I/O(Input/Output) 與執行 (指一般指令) 重疊，已達成分享不同的效能。

本計畫為達成驗證更多的作業系統的多執行緒與循序之程序對於大量資料存入或自快閃記憶體讀取出，以找出最適切的作業系統以滿足嵌入式系統在不同應用環境下的需求。

本計畫以微軟公司的作業系統 Windows ME, Windows XP, 與 Windows 2000 Server 版進行測試，並配合具有提供多執行緒的程式語言 Java(Borland 公司的 Jbuilder) 與 C++(微軟公司的 Visual C++ 6.0 版)，撰寫演算法並以不同長短的存取資料長度測試各種的存取時間。

三、研究方法與成果

3.1 多執行緒演算法的實作

以微軟公司的 Visual C++ 6.0 版而言，多執行緒的執行必須將翻譯的 (Compiler) 環境重新設定。首先設定翻譯 C/C++ 的 Category 為 Code Generation 與 Use run-time library 為 Multithreaded DLL 即可。

反觀 Java 與微軟公司的 C# .net 環境下執行並不用特別加以修改任何設定，本計畫所使用的多執行緒的架構如圖 1 所示。

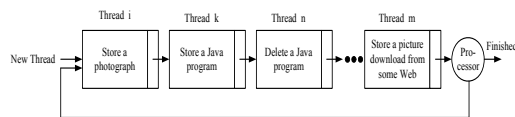
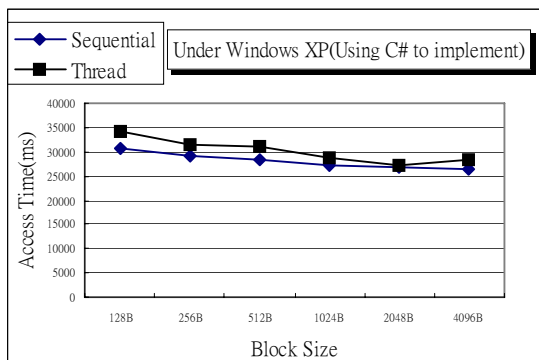


圖 1 多執行緒的同步存取架構示意圖

3.2 使用不同程式語言在三種作業系統下進行多執行緒與循序存取的試驗

在 Windows XP 下，分別使用循序的方式與多執行緒的方式執行，其模擬的結果如圖 2(使用 C# 撰寫演算法)與圖 3(使用 Java 撰寫演算法)所示，在本計畫所使用的演算法下，多執行緒分別增加了 7.4% 與



18.3% 的執行時間。

圖 2 在 Windows XP 使用 C# 撰寫演算法在不同存取區段之存取時間

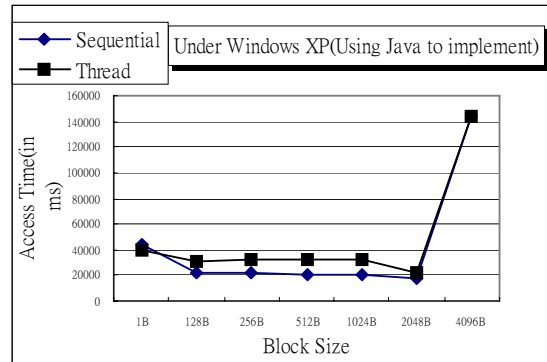
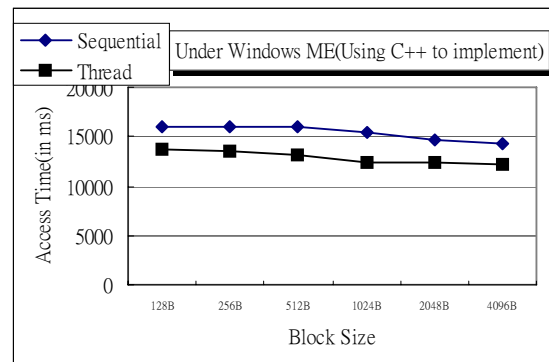


圖 3 在 Windows XP 使用 Java 撰寫演算法在不同存取區段之存取時間

反之在 Windows ME 的作業系統下，



也分別使用循序式與多執行緒的方式執行，其模擬的結果如圖 4(使用 C# 撰寫演算法)與圖 5(使用 Java 撰寫演算法)所示，在本計畫所使用的演算法下，多執行緒分別減少了 7.4% 與增加 1.5% 的執行時間。

圖 4 在 Windows ME 使用 C++ 撰寫演算法在不同存取區段之存取時間

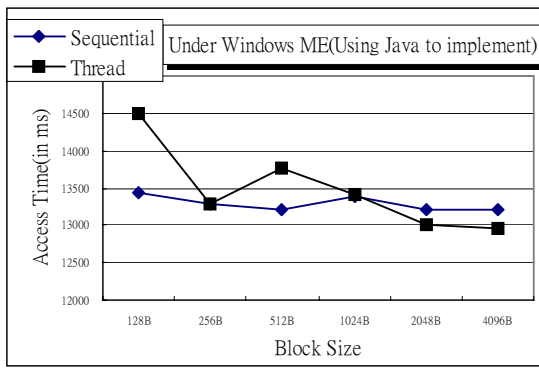


圖 5 在 Windows ME 使用 Java 撰寫演算法在不同存取區段之存取時間

作業系統功能的多寡與其效能的高低，通常是相互背離，簡而言之，作業系統為了提供更多的功能，因此整個系統也愈來愈大，致使整體的效能大大下降，尤其在一些多人使用的中大型電腦下的作業系統更是如此，本計畫初期以 Windows ME 與 XP 下的評比後，也證實此一現象，但經過 Windows 2000 Server 版上執行後，卻發現與前面及一些電腦的常態不同，在 Windows 2000 Server 版的作業系統下不論使用 C++ 語言(如圖 6 所示，多執行緒可提升 68.2%的執行效能)或 Java 語言(如圖 7 所示，多執行緒可提升 56.5%的執行效能)撰寫程式，都可以得到比較好的執行效能。在同一部電腦下模擬甚至比 Windows ME 和 Windows XP 下執行效能要好很多，所以我們可以從圖 2 至圖 7 觀察出在 Windows 2000 Server 版下不論是執行緒或者是循序程序的執行皆可得到相對較佳的效能。

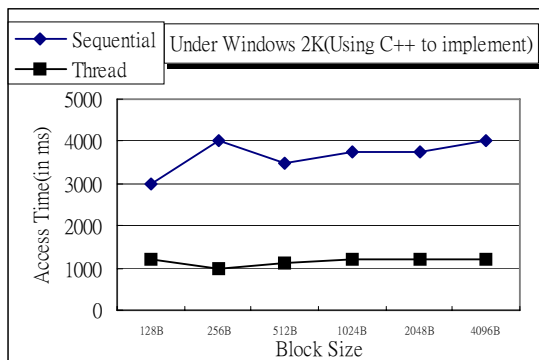


圖 6 在 Windows 2000 Server 版下使用 C++ 撰寫演算法在不同存取區段之存取時間

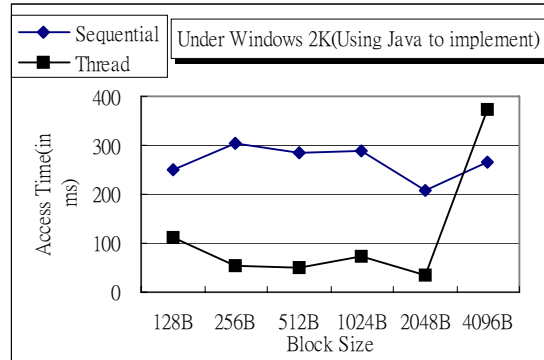


圖 7 在 Windows 2000 版下使用 Java 撰寫演算法在不同存取區段之存取時間

存取區段的大小也會影響到整體的執行效能，在 Windows ME 下，由部分實驗的數據顯示其規劃的緩衝區(Buffer)相當的大，所以除每次存取區段太小(如：少數幾個位元組)外，其存取所花的時間差異不大，Windows XP 下所規劃的緩衝區相對之下就比較小，其存取區段也是與 Windows ME 相當不能少數幾個位元組。而在 Windows 2000 Sever 版上存取區段也不能太小，否則其存取效能一樣相當差。

一般程式語言的觀點是編譯式會比直譯式的效能要好，在 Windows ME 與 Windows XP 下，再分別以 C++、C#或 Java 撰寫出的程式後，經本計畫的實驗驗證後 Java 語言反而效能稍微好一點，但整體上而言差異並不十分顯著，反觀在 Windows 2000 Server 版上，Java 所撰寫的程式其效能卻比 C++好上十幾倍，圖 8 顯示使用 Java 比 C++效能快 13.7%，圖 9 也顯示使用 Java 比 C++效能快 10.0%，這也是違反一般常理。

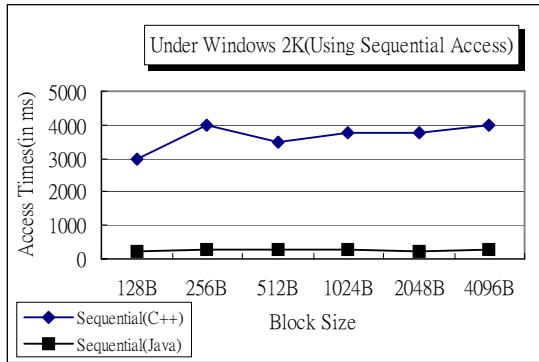


圖 8 在 Windows 2000 Server 版下使用 C++ 與 Java 撰寫演算法以循序的方式在不同存取區段之存取時間

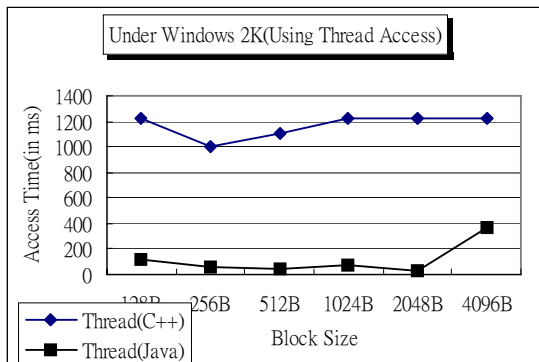


圖 9 在 Windows 2000 Server 版下使用 C++ 與 Java 撰寫演算法以循序的方式在不同存取區段之存取時間

四、結論

一般而言，多執行緒的程序之執行會比循序方式的執行效能要好，這指的是每一個程序在每一次的動作時間(Burst Time)較短的情形下，否則作業系統(如：Windows XP)因為整個系統的執行緒太多，而所有的程序為了共用 CPU 的時間，反而必須常常進行本文交換(Context Switch)；而浪費不少時間在儲存現有程序或回存下一個即將被執行之程序的暫存器等內容，這些儲存與回存的動作對整個程

序的進行只是額外負擔(Overhead)。綜合以上的論述，反而在 Windows XP 以 Java 語言實作的程式進行本計畫的試驗，可以獲得較好的效能。反觀在 Windows 2000 Server 版下多執行緒的程序，以及直譯式的 Java 所撰寫的程序皆可達到較佳的執行效能，後者卻有違反一般常態。

我們未來的計畫希望致力於作業系統的修改，初步以了解最小的嵌入式作業系統 uC/OS-II 為主，以其排程程式為著手目標，了解其排程使用的演算法，並進行修改以搭配嵌入式系統在不同環境獲得最佳的效能。進而深入了解在 Windows 2000 Server 版下與其他作業系統，其對於 JVM(Java Virtual Machine)的支援程度。

五.參考文獻

1. Frank Vahid and Tony Givargis, 2002, "Embedded System Design: A Unified Hardware/Software Introduction", John Wiley & Sons, Inc., pp 2~27.
2. Daniel W. Lewis, 2002, "Fundamentals of Embedded Software: Where C and Assembly Meet", Prentice Hall, pp. 1~13.
3. VISA International 1997, "CAD/Service Payment Terminal Specification", Version 3.0, pp.4-67~4-70.
4. Atmel Corporation 1996, "Atmel Nonvolatile Memory Data Book", Overview and Preface.
5. Peter C.B., Doron J.C.,1979, "Introduction to Data Structures and Non-Numeric Computation", Prentice-Hall, Inc. pp 209~279.
6. Robert L. Kruse,1987,"Data Structures and Program Design", Second Edition, Prentice-Hall International Edition, pp 104~153.
7. E. Horowitz, S. Sahni, 1984, "Fundamentals of Data Structures in Pascal", Second Edition, Computer Science Press, pp 94~180.
8. Thomas H. Cormen, Charles E. Lesiserson, Ronald L. Rivenst, and Clifford Stein, 2001, "Introduction to Algorithm", Second Edition, The MIT Press, pp 200~214.
9. Robert Sedgewick, 1998, "Algorithms In C Parts 1-4 : Fundamentals Data Structures, Sorting and Searching ", Third Edition, Addition Wesley, pp 69~126.
10. Ellis Horowitz, Sartaj Sahni, and Sanguthevar Rajasekaran, 1997, "Computer Algorithms C++",

- Computer Science Press, pp 339~378.
11. Donald E. Knuth, 1981, "The Art of Computer Programming : Seminumerical Algorithms", Volume II, Second Edition, Addition-Wesley, pp 1~170.
 12. Donald E. Knuth, 1981, "The Art of Computer Programming : Fundamental Algorithms", Volume I, Second Edition, Addition-Wesley, pp 228~361.
 13. Richard Neapolitan and Kumarss Naimipour, 1998, "Foundations of Algorithms using C++ Pseudocode", Second Edition, Jones and Bartlett.
 14. Michael T. GoodRich and Roberto Tamassia, 2001, "Algorithm Design : Foundations, analysis, and Internet Examples", Wiley, 55~212.
 15. Harvey M. Deitel and Paul J. Deitel, 2001, "C++ How to Program", Third Edition, Prentice Hall, pp 304~388.
 16. Abraham Silberschatz, Peter B. Galvin, and Greg Gagne, 2003, "Operating System Concept : Windows XP Update", Sixth Edition, John Wiley, pp 273~451.
 17. R. C. T. Lee, R. C. Chang, S. S. Tseng, and Y. T. Tsai, 2001, "Introduction to the Design and Analysis of Algorithms", Second Edition, 旗標出版社, pp 619~656.
 18. Andrew S. Tanenbaum, 2001, "Modern Operating Systems", Prentice Hall, pp 189~268.
 19. David A. Patterson and John L. Hennessy, 1998, "Computer Organization & Design : The Hardware / Software Interface", Second Edition, Morgan Kaufmann, pp 538~628.
 20. Kenneth A. Lambert and Martin Osborne, 2000, "Java : A Framework for Program Design and Data Structures", Brooks/Cole Thomson Learning, pp 114~144.
 21. Kenneth A. Lambert and Martin Osborne, 2000, "Java : Complete Course in Programming & Problem Solving", South-Western Educational Publishing, pp 156~181.
 22. Harvey M. Deitel and Paul J. Deitel, 2001, "Java How to Program", Third Edition, Prentice Hall, pp 1053~1098.
 23. Arthur E. Anderson and William J. Heinze, 1992, "C++ Programming and Fundamental Concepts", Prentice Hall, pp 215~240.
 24. 吳家榮, 1998, "連結串列再快閃記憶體上的應用", 八十六年專科學校重點科技專題製作論文研討會, pp1-1~1-15.
 25. 新華電腦股份有限公司編著, 2004, ARM 內核嵌入式 SOC 原理以 ARM7TDMI S3C4510B 為例, 全華出版股份有限公司.
 26. 張正源編著, 2005, ARM Linux 嵌入式系統發展技術, 全華出版股份有限公司.
 27. 黃文增編譯, 2005, MicroC/OS-II : 即時作業系統核心 (原文書 : MicroC/OS-II: The real-time kernel, Second Edition, Jean J. Labrosse), 全華出版股份有限公司.

晶片系統架構技術及開發平台研究之推動(3/3) - 附屬計畫四

【應用於多媒體電路供電系統核心電路之研究】

Research on power supply system core circuit for multimedia applications

計畫編號：NSC93-2215-E-002-010/004

執行期間：93/8/1~94/7/31

計畫主持人：林永裁 講師 龍華科技大學電子工程系

參與人員：林和正、王鴻展

一、中文摘要

行動電子產品在目前消費性市場的需求快速增加，這些行動電子產品需要使用高效率低電壓的電壓轉換器以使用單一電池的供電能力能夠使系統使用的時間達到最長之目的。為了降低這些行動電子產品的體積與重量，直接將供電系統小型化是最有效的方法，因此，電路研究與設計的目標已經鎖定在高效率與低功率電壓轉換器的設計技術上，在這些應中，選擇一個適用的技術便成為很重要的保證與問題，從過去的文獻可知交換式電源技術是各種技術當中最具有效率的一種技術，脈波寬度調變 (Pulse Width Modulation, PWM) 技術是最常用於交換式電源電路的一種控制方法，此種方式乃是改變切換電晶體的導通時間以達到控制功率之流通及調整輸出電壓至預定值為目的。

本計畫提出以 CMOS 製程技術設計降壓型的直流轉換器，轉換器的核心控制電路是以最新的環型振盪器來產生 PWM 電路，這個 PWM 電路是以由稱為擬似雙曲線(pseudo hyperbola)電路加以電流補償而達定頻操作，不需要任何外加被動元件且可以於低電壓低功率環境下操作，其應用非常的廣泛，極適用於低功率的多媒體電

路系統，電源系統與電子通訊電路上。

關鍵字：直流對直流轉換器、低電壓、低功率、脈波寬度調變電路。

Abstract

Recently, portable electronic devices are in great demand of consumer market. These devices require high-efficiency low-voltage dc-dc converters to maximize the run time of the devices from a single battery source. In order to decrease the size and weight of these devices, miniaturization of the power modules is essential. As a result, the trend is focused on implementation of converter with high-efficiency and low-power consumption. To select an appropriate technology is an important issue for these applications. Pulse width modulation (PWM) technique is the most frequently used technique in switching converters. The kernel circuit of PWM is voltage-controlled duty-cycle (VCDC) circuit. A variable in a power system can be controlled through modulating the width of a control pulse in each switching cycle. In this project, we propose a monolithic, low-voltage

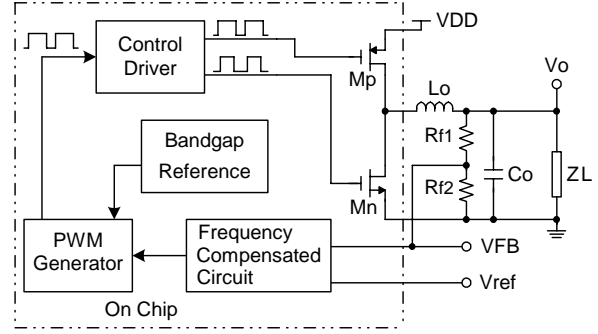
step-down dc-dc converter with on-chip PWM controlled circuit for the portable electronic systems. The performances of the proposed switching converter are verified with 2P4M 0.35- μm models through Spice simulations.

Keywords: DC-DC Converter, Low-power, low-voltage, bandgap reference, pulse width modulation.

二、計畫緣由與目的

目前大多數的交換式電源供給器都以脈波寬度調變(pulse width modulation, PWM)的控制方式[1-10]來設計，此種方式乃改變切換電晶體的導通時間以達到控制功率之流通及調整輸出電壓至預定值。

本計畫使用 CMOS 技術設計一個單晶片的供電控制電路這是一個可由使用者設定電壓值的轉換器，其功能方塊如圖一所示，其中頻率補償電路是為了補償轉換器的極點與零點，以使轉換器獲得較大的操作頻寬與穩定度，另一個功能是讓使用者可以由電路需求自行設定輸出電壓；PWM 產生電路是使用去年計畫所設計的環型振盪器再加以電流補償的電路，它提供控制所需的控制信號用以使開關電晶體導通或斷路以決定輸電壓值；控制驅動電路是不重疊的信號處理電路，以防止兩個電晶體同時導通而燒毀。



圖一 多媒體電路供電系統核心電路方塊圖

一個可控制電流的延時核元電路(delay cell circuit)，用來控制振盪訊號的作用週期，其電路結構如圖二所示，電路的轉移電壓(transfer voltage)可表示為

$$V_M \cong V_{DD} - \left[|V_{TP}| + \sqrt{\frac{I + I_{c2}}{2\mu_p C_{ox} \left(\frac{W}{L}\right)_p}} \right] \quad I_{c1} \gg I_{c2}$$

$$\cong V_{TN} + \sqrt{\frac{I + I_{c1}}{2\mu_n C_{ox} \left(\frac{W}{L}\right)_n}} \quad I_{c2} \gg I_{c1}$$

$$\cong \frac{V_{TN} + (V_{DD} + V_{TP}) \sqrt{\mu_p \left(\frac{W}{L}\right)_p / \mu_n \left(\frac{W}{L}\right)_n}}{1 + \sqrt{\mu_p \left(\frac{W}{L}\right)_p / \mu_n \left(\frac{W}{L}\right)_n}} \quad I_{c1} \approx I_{c2}$$

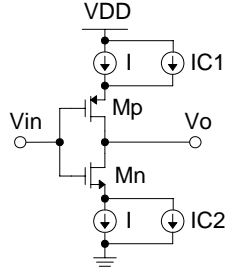
(1)

其中 μ_p , C_{ox} , $(W/L)_p$, V_{TP} , 和 V_{DD} 分別為 PMOS 電晶體的通道遷移率，閘極單位面積電容，寬長比，界限電壓及供應電壓；而 μ_n , C_{ox} , $(W/L)_n$ 和 V_{TN} ，分別為 NMOS 電晶體的通道遷移率，閘極單位面積電容，寬長比，界限電壓。延時核元電路的上升時間 t_{LH} 與下降時間 t_{HL} 的推導結果分別為

$$t_{LH} \approx \frac{C_P V_M}{I_{c2}} \quad (2)$$

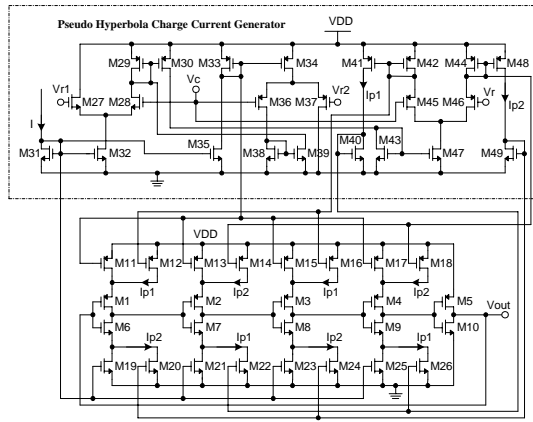
$$t_{HL} \approx \frac{C_P (V_{DD} - V_M)}{I_{c1}} \quad (3)$$

其中 C_P 是延時核元電路的輸入電容。



圖二 延時核元電路

使用延時核元電路接成的PWM電路架構如圖三所示。



圖三 延時核元電流控制作用週期 PWM 電路

圖三是一個以電流控制作用週期的振盪器，其振盪週期 T 與作用週期 D 可分別表示如下：

$$T = C_P \left(\frac{2 \frac{V_M}{I_{p1}} + 2 \frac{V_{DD} - V_M}{I_{p1}}}{2 \frac{V_M}{I_{p2}} + 2 \frac{V_{DD} - V_M}{I_{p2}}} \right) = 2C_P V_{DD} \left(\frac{1}{I_{p1}} + \frac{1}{I_{p2}} \right) \quad (4)$$

$$D = C_P \frac{2 \frac{V_{DD} - V_M}{I_{p2}} + 2 \frac{V_M}{I_{p2}}}{T} = \frac{1}{\frac{1}{I_{p1}} + \frac{1}{I_{p2}}} \quad (5)$$

若充電電流與放電電流的和等於常數且等於偏壓電流 I_{ss} ，則當不同的控制電壓

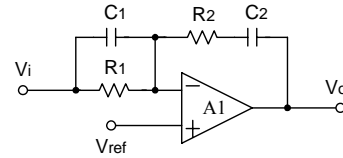
V_c 控制時，振盪器將無法得到定頻輸出，因此充電電流與放電電流的和必須以式(4)與(5)的條件加以補償，吾人以 Pseudo Hyperbola 電路[8]補償使得

$$\frac{1}{I_{p1}} + \frac{1}{I_{p2}} = \frac{T}{2C_P V_{DD}} = \text{常數} \quad (6)$$

轉換器含有兩個極點與零點[9, 10]，為了達到穩定控制的目的我們必須使用如圖四所示的電路加以補償，其控制轉移函數為

$$\frac{v_o(s)}{v_i(s)} = \frac{(1 + sR_1C_1)(1 + sR_2C_2)}{sR_1C_2} \quad (7)$$

由(7)中適當選擇的電阻與電容值即可得所需的控制要求。

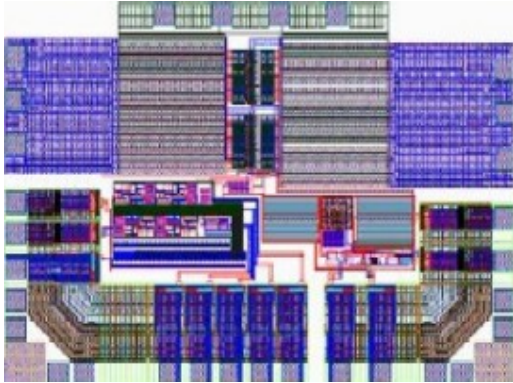


圖四 頻率補償電路

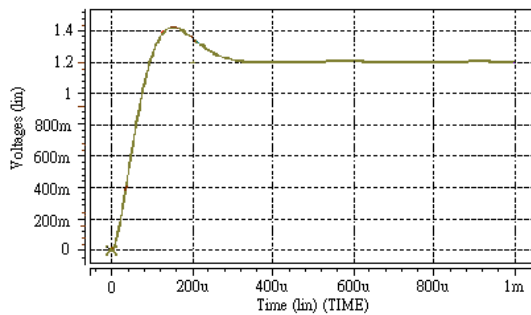
三、結果與討論

本計畫已成功的完成低功率降壓式之直流對直流轉換器設計並完成積體電路之佈局，這個電源處理電路是以 TSMC 0.35- μm 2P4M 的元件模型來設計，其功能是以積體電路模擬軟體加以驗證，轉換器的電路佈局如圖五所示，電路佈局不含接點所需的面積為 880 \times 1040 μm^2 ，電路特性經積體電路模擬軟體加以驗證後，模擬結果顯示在供電電壓為 3.3 V 時可以穩定的輸出電壓為 1.0 to 2.2 V，如圖六所示為轉換器之階層上升(step-up)為 1.2 V 之輸出特性，其穩定時間約為 0.36 ms；當輸出電壓為 1.2 V 時，其漣波電壓約為 10 mV

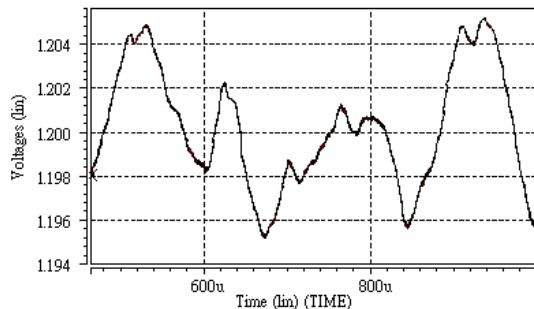
其結果如圖七所示；轉換器之最高效率為百分之九十四，而負載電流由 100 mA 變化 600 mA 時其轉換效率都還可以維持百分之八十五以上，其結果如圖八所示。



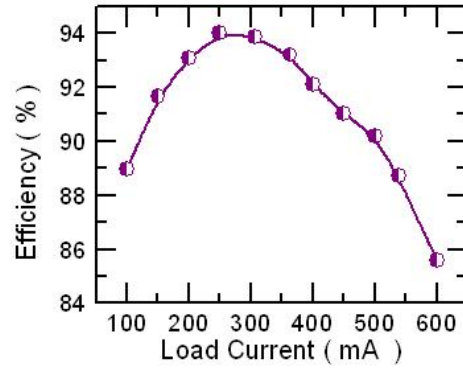
圖五 轉換器電路之佈局，其不含接點所需面積為 $880 \times 1040 \mu\text{m}^2$



圖六 轉換器在供電電壓為 $V_{DD} = 3.3 \text{ V}$ 時之輸出情況，其輸電壓為 1.2V，穩定時間約為 0.36 ms。



圖七 轉換器之輸出電壓為 1.2V 時之漣波輸出情況，其最大漣波約為 10 mV。



圖八 轉換器之轉換效率與不同負載電流之變化情況。

四、結論

小型化與長效性的電壓供電系統是行動電子產品極為重要的需求之一，利用交換式電源技術可以有效的達到這項要求，本計畫以數位 CMOS 技術作為開發類比積體電路的交換式電源核心電路晶片，已順利的完成低電壓降壓式直流對直流轉換器積體電路設計與晶片佈局，應用於多電壓系統可以大幅度的解決多媒體系統與行動電子產品供電系統的體積及效率等問題，對於增長電池的使用時間及產品小型化有極大的助益，這個計畫的成果可輕易的被嵌入多媒體與行動電子產品中。本計畫之部份研究成果已發表[11]或已被接受[12] 由 IEEE 舉辦的國際性相關研討會上。

由於時間的限制，部份目標尚在進行中，如升降壓型轉換技術、電流感測技術、磁滯電流控制電路與及最佳化控制機制等技術，都是提高電池的使用效率之極佳論點，也是未來要繼續努力的方向。

五、參考文獻

- [1] Smith K. M., Jr., Lai Z., and Smedley K. M.: "A new PWM controller with one-cycle response," *IEEE Trans. on Power Electronics*, Jan. 1999, 14, (1), pp. 142-150.
- [2] K. M. Smedley, S. Cuk, "One-cycle control of switching converters," in *proc. of Power Electronics Specialists Conference, 1991, PESC '91*, June 1991, pp. :888-896.

- [3] Dongsheng Ma, Wing-Hung Ki, Chi-Ying Tsui, "An integrated one-cycle control buck converter with adaptive output and dual loop for output error correction," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 1 Jan. 2004.
- [4] A. M. Wu, Jinwen Xiao, D. Markovic, S. R. Sanders, "Digital PWM control: application in voltage regulation modules," in *proc. of 30th Annual IEEE PESC 99*, 1999, vol. 1, pp. 77-83.
- [5] Chung-Hsien Tso and Jiin-Chuan Wu, "An integrated digital PWM DC/DC converter using proportional current feedback," in *proc. of the 2001 IEEE International Symposium on Circuits and Systems, ISCAS 2001*, vol. 3, pp. 65-68.
- [6] C. F. Lee and P. K. T. Mok, "A monolithic current-mode CMOS dc-dc converter with on-chip current-sensing technique," *IEEE Journal of Solid-State Circuits*, Vol. 39, No. 1, pp. 3-14, Jan. 2004.
- [7] Cheung Fai Lee, and Philip K. T. Mok, "On-chip current sensing technique for CMOS monolithic switch-mode power converter," in *proc. of IEEE ISCAS*, May 2002, PP. V265-V268.
- [8] Jiann-Jong Chen, Yeong-Tsair Lin, Hung-Yih Lin, and Wen-Yaw Chung, "Integrated Pulse-width-modulation Circuit Using CMOS Processes," in *proc. of 2004 35th IEEE Power Electronics Specialists Conference, PESC'04*, June 20-25 2004, Aachen, Germany, pp. 1356-1358.
- [9] Lai Z. and Smedley K. M.: "A general constant-frequency pulse-width modulator and its applications," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, April 1998, 45, (4), pp. 386-396.
- [10] Robert W. Erickson, *Fundamental of Power Electronics*. Norwell, MA, Kluwer Academic Publishers, 1999.
- [11] Yeong-Tsair Lin, Wen-Yaw Chung, Dong-Shiu Wu, Ho-Cheng Lin, and Robert Lin, "A Low Voltage CMOS Bandgap Reference," in *proc. of the 3rd IEEE International Northeast Workshop on Circuits and Systems 2005, NEWCAS 2005*, June 19-22 2005, Quebec, Canada, pp. 227-230.
- [12] Yeong-Tsair Lin, Wen-Yaw Chung, Dong-Shiu Wu, Hung-Chan Wang, Hung-Yih Lin, and Jiann-Jong Chen, "A monolithic CMOS step-down dc-dc converter," accepted by *the 48th MWSCAS*, Aug. 7-10 2005, Cincinnati, U.S.A.

晶片系統架構技術及開發平台研究之推動(3/3) - 附屬計畫五

使用 FPGA 設計二維封包分類演算法

Use FPGA to design two-dimensional packet classification algorithms

計畫編號：NSC93-2215-E-002-010/005

執行期間：93/8/1~94/7/31

計畫主持人：吳其政 講師 龍華科技大學電子工程系

參與人員：劉治銘、王柏淳

一、中文摘要

由於網際網路(Internet)的快速發展,網際網路訊務(traffic)的增加需要高速的連線速度(link speed)和路由器(routers),目前已有 10Gbps OC-192 光纖線路,因此網際網路訊務的主要瓶頸(bottleneck)已由網路線路轉移至路由器。另一方面,網際網路服務提供者(ISP)會希望路由器能提供差異性服務(differentiated service)的功能。但是傳統的路由器主要是查詢 IP 封包(packet)的目的位址(destination address)來決定封包的下一站(next-hop)位址,因此傳統的路由器對於所有相同目的位址之封包的處理並無分別,所以它無法提供差異性服務的功能。具有封包分類(packet classification)的路由器能夠根據封包的目的位址、來源位址、目的埠(destination port)、來源埠(source port)和通訊協定類別(protocol type)等欄位來分類封包。這種多欄位(multi-field)封包分類能提供各種不同的型式的差異性服務;例如:阻擋有危害安全的封包進出某網路(如防火牆),服務品質路由(QoS routing),以及資源保留服務(RSVP)等。然而,在封包的各欄位中,處理目的位址和來源位址視為主要的瓶頸所在,本計畫使用 BDD (binary decision diagram)方法設計出二維(來源 IP 位址,目

的 IP 位址)的封包分類器演算法,BDD 具有節點共享的優點,刪除重複的節點,有效率地降低組合電路的成本,而且運用管線化(pipelined)的技術提升整體封包分類的處理速度,是一個適合 FPGA 實現的演算法,我們成功的將它以 FPGA 硬體實現。

關鍵字

路由器、封包分類、BDD (binary decision diagram)、FPGA。

Abstract

The Internet traffic is rapidly growing not only because of the increase of users but also because of the multimedia applications. With the increased traffic, fast link speed and routers are required. Currently, gigabit links are available (such as OC-192). The chief bottleneck of the traffic has moved from the links to the routers. Meanwhile, the Internet Service Providers (ISPs) would like to provide the differentiated services. The traditional routers lookup routing table for next-hop address based on packet's destination IP address. That is, the routers treat all the packets with the same destination address identically. Thus, the

traditional routers cannot provide the functions of differentiated service. A packet classification router based on multi-field from the packet header fields, such as destination and source IP addresses, destination and source ports, and protocol type. The multi-field packet classification can provide various types of differentiated service include of Virtual Private Networks (VPNs), packet filtering in firewalls, QoS routing, and RSVP. Consequently, multi-field packet classification has become an essential component of next generation routers and firewalls. However, the multi-field packet classification has proved to be far more difficult and take more CPU time. Recently, several methods have been proposed for packet classification. Although some of these methods provide fast packet classify performance, require $O(n^k)$ memory. Warkkhede et al. show a two-field packet classification, a binary tuple-space search method, requires $O(\log^2 w)$ hashes and only $O(n \log^2 w)$ memory. In this project, we propose a binary decision diagrams (BDDs) based combinational circuit for an efficient implementation of fast two-dimensional packet classification scheme in reconfigurable hardware, FPGA.

Keywords

router, packet classification, binary decision diagrams (BDDs), FPGA.

二、動機與目的

多欄位封包分類根據封包的目的位址、來源位址、目的埠、來源埠和通訊協定類別等欄位來分類封包。這種多欄位封包分類可使用於各種網路服務和應用程式

中，包含阻擋有危害安全的封包進出某網路(如防火牆)、服務品質路由(QoS routing)、資源保留服務(RSVP)、以及虛擬私人網路(VPN)等。因此多欄位封包分類已經成為新一代路由器及防火牆所必備的功能。

網際網路原本是使用銅線來當傳輸媒體，但是現在光纖已漸漸取代了銅線。光纖傳輸系統提供高頻寬(high bandwidth)，他傳送資料的速度高達每秒鐘好幾個 gigabytes 的速度(OC48 = 2.4Gbps, OC192 = 10Gbps, OC768 = 40Gbps)，這個速度稱為線速度(wire speed)。由於線速度的提升，網際網路路由器必須增快封包處理的速度。然而，多欄位封包分類需要花費較多 CPU 時間，成為路由器主要的瓶頸(bottleneck)，雖然目前已有許多封包分類演算法的研究[1-8]，其中有些演算法雖然能提供快速的查詢效能，但是需要 $O(n^k)$ 的記憶體空間，其中 k 為處理欄位的個數， n 為封包分類器資料庫規則的個數。當 n 很大時，這些演算法並不實際。本計畫使用的方法是以 binary decision diagram(BDD) [9, 10]的架構來建構封包分類函數，由於 BDD 具有節點共享(node sharing)的優點，可以省去重複的節點，不僅有效率地降低了電路的成本而且提升了封包分類的處理速度。

三、研究方法與成果

封包分類依照封包的 K 個標頭欄位來分類，則稱此分類器(classifier) F (或稱 filter，過濾器)為 K -tuple($F[1], F[2], \dots, F[K]$)封包分類器，其中 $F[i]$ 可為不同長度的前置位元串(prefix bit string)或是某範圍(range)。最常見的欄位有 IP 目的位址，IP 來源位址，TCP 目的埠號(port number)，TCP 來源埠號和通訊協定型式(protocol

type)等，通常每個分類器 F 代表一個規則並附帶一個動作指示(action directive)，例如在防火牆會指明分類器 F 的動作是允許(allow)或禁止(deny)。例如，分類器 $F = (140.131.3.* , * , 21 , * , TCP , deny)$ ，表示分類器 F 的目的地為子網路 140.131.13.*、目的埠為 21、來源位址和來源埠不檢查、使用 TCP、動作是 deny，這表示禁止使用 ftp 進入子網路 140.131.3.*。

分類器資料庫 FS 包含有 N 個規則表示有 N 個分類器 F_1, F_2, \dots, F_N 。每一個分類器 F_j 有 K 個不同欄位或 K 維， $F_j[1], F_j[2], \dots, F_j[K]$ 。表(一)所示為一個簡單的封包分類器資料庫。

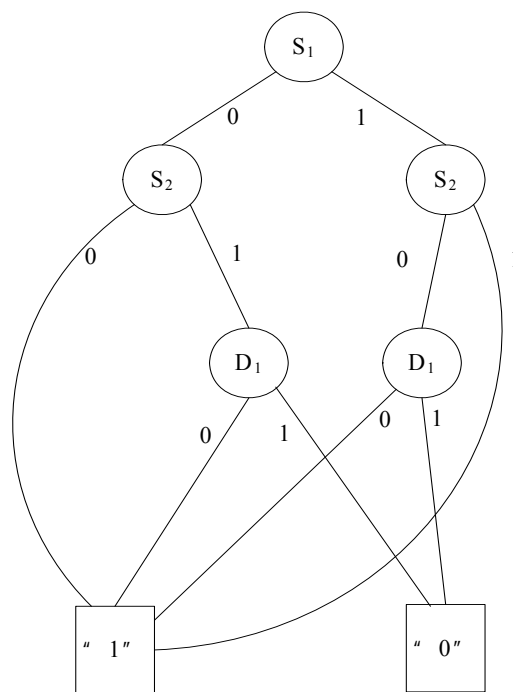
表一 簡單的封包分類器資料庫的例子

Dest Addr	Src Addr	Dest Port	Src Port	Prot	Action
140.131.1.*	140.131.4.*	*	*	*	allow
140.131.2.*	140.131.5.*	*	gt 1023	TCP	allow
140.131.3.*	140.131.**	[20,21]	*	TCP	deny
140.131.**	10.1.**	*	*	*	allow
*	*	*	*	*	deny

我們的二維封包分類器電路的產生可分為二個階段，首先，我們利用封包分類器資料庫的規則建立起 BDD 樹狀結構。例如，表二為一簡單的二維封包分類器資料庫的例子，其中來源位址欄位只有兩個位元，目的位址欄位只有一個位元，Action 欄位中 1 表示是 allow，0 表示是 deny。根據這個二維封包分類器規則，我們可建立起如圖(一)所示的 BDD 樹狀結構圖。

表二 二維封包分類器資料庫。

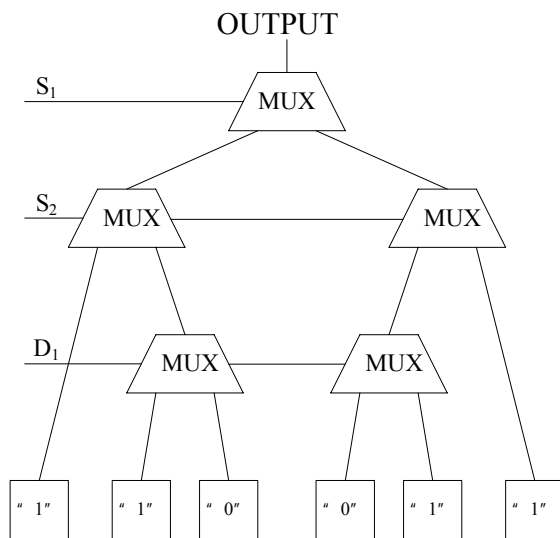
Src Addr	Dest Addr	Action
$S_1 S_2$	D_1	
0 *	0	1
1 0	1	1
0 0	1	1
1 1	*	1
*	*	0



圖一 表(二)所建立的 BDD 樹狀結構圖。

在第二階段，我們依照所建立的 BDD 樹狀結構圖來產生相對應的組合(combination)或管線化(pipelined)電路，在 BDD 樹狀結構圖中的每一個節點我們使用多工器(MUX)來替代，多工器的輸入為 ACTION 0 或 1，節點內含值當作多工器選擇線的輸入。如圖(二)所示，即為圖(一)BDD 樹狀結構圖的組合電路。之後，

我們將此組合電路以 FPGA 來實現。



圖二 圖(一)BDD 樹狀結構圖的組合電路

四、結論

本計畫使用 BDD 方法設計出二維(來源 IP 位址, 目的 IP 位址)的封包分類器演算法, BDD 具有節點共享的優點, 刪除重複的節點, 有效率地降低組合電路的成本, 而且運用管線化的技術提升整體封包分類的處理速度, 是一個適合 FPGA 實現的演算法, 我們成功的將它以 FPGA 硬體實現。實驗結果證明, 當一個封包到達時, 封包分類器能快速地找出最佳的符合規則, 採取適當的動作。我們相信所設計的封包分類器具有彈性, 能符合未來 IPv6 128 個 IP 位元的擴充空間, 也很容易的擴充來處理多維的封包分類。

五、參考文獻

[1] V. Srinivasan, S. Suri, G. Varghese, and M. Waldvogel, Fast and Scalable Layer four Switching, *Proc. of ACM SIGCOMM*,

September 1998, pp. 203-214.

- [2] P. Gupta and N. McKeown, Packet classification on multiple fields, *Proc. of ACM SIGCOMM*, Sept. 1999.
- [3] P. Gupta and N. McKeown, Packet Classification using Hierarchical Intelligent Cutting, *IEEE Micro*, vol. 20, no. 1, January/February 2000, pp. 34-41.
- [4] T. V. Lakshman and D. Stiliadis, High-speed policy-based packet forwarding using efficient multi-dimensional range matching, *Proc. of ACM SIGCOMM*, Sept. 1998, pp. 191-202.
- [5] V. Srinivasan, G. Varghese, S. Suri, and M. Waldvogel, Scalable level 4 switching and fast firewall processing, *Proc. of ACM SIGCOMM*, Sept. 1998.
- [6] V. Srinivasan, G. Varghese, and S. Suri, Fast packet classification using tuple space search, *Proc. of ACM SIGCOMM*, Sept. 1999.
- [7] T. Woo, A modular approach to packet classification: algorithms and results, *Proc. of IEEE INFOCOM*, Israel, March 2000.
- [8] A. Feldmann and S. Muthukrishnan, Tradeoffs for packet classification, *Proc. of IEEE INFOCOM*, vol. 3, March 2000, pp 1193-1202.
- [9] R. Bryant, Graph-based Algorithms for Boolean Function Manipulation, *IEEE transactions on computers*, vol. C-35, no. 8, pp.677-691, Aug. 1986.
- [10] K. Brace, R. Rudell, and R. Bryant, Efficient implementation of a BDD package, *proc. 27th IEEE/ACM Design Automation Conf.*, 1990, pp. 40-45.

晶片系統架構技術及開發平台研究之推動(3/3)-附屬計畫六

NSC 93-2215-E-002 -010/006 精簡報告

SOC 嵌入式系統之電源規劃

方志鵬 魏國展 許毓辰
國立台北科技大學電機系

摘要

由於 SOC (System On Chip) 技術的進步，諸如監控、量測、家電等各種不同的領域都有研發人員採用 SOC 晶片製成各式各樣的嵌入式系統。這些嵌入式系統都具有輕薄短小的特點，而它們的問世所帶來的好處就是讓生活更方便、更有效率。除了性能的優劣及功能的多寡外，功率消耗也是設計嵌入式系統時的一個重要考量因素。目前對嵌入式系統率消耗的研究多是從架構設計、硬體電路、作業系統等方面著手。我們認為在設計嵌入式系統時，每一個環節都應該盡力避免無謂的功率消耗，而在設計過程中，開發應用程式的階段顯然是較被忽略的一環。所以本論文討論如何從應用程式的層次改善嵌入式系統之功率消耗。

關鍵字：嵌入式系統、功率消耗

壹、緒論

為了讓生活更方便、工作更有效率，人們都希望所用的工具、設備更輕便、功能更多、操控連絡的範圍更遠。隨著 VLSI 技術的進步，SOC 晶片變成可行[3, 4]，並以嵌入式系統的形式逐漸出現在日常生活、通訊、工業、醫療、軍事等應用上。目前在 SOC 嵌入式系統的應用方面，以 8 位元及 32 位元為主，8 位元系統已算是成熟的設計，而 32 位元的系統可說是方興未艾，已經有許多 32 位元的嵌入式系統作成，例如 PDA、機上盒、大哥大、網路電話等，其它像家庭娛樂、遠端監控也都陸續有 32 位元的嵌入式系統問世。

SOC 嵌入式系統的設計流程如圖一所示，要得出好的設計，必需考量很多因素，包括：硬體的考量(如 SOC 晶片的選擇、系統記憶體的配置、IP 或介面 IC 的選用)、軟體的考量(如作業系統的選擇、應用程式應達成的功能)[5]。而這些考量因素所要求的目標除了使設計的產品具有合適的性能及功能外，還希望產品的功率消耗

降至合理範圍內[6]。這是因為嵌入式系統所使用之環境可能沒有方便的交流電源插座，而需要以電池提供電源。如果系統耗電量過大，頻繁的電池更換變成擾人的負擔，重要的工作如常因為電池供電不足而中斷更是掃興。再者，32 位元嵌入式系統中很重要的一項功能就是無線傳輸[1]，這項功能在整體系統的功率消耗所佔的比率相當大。所以功率消耗是設計嵌入式系統時的一個重要考量因素。

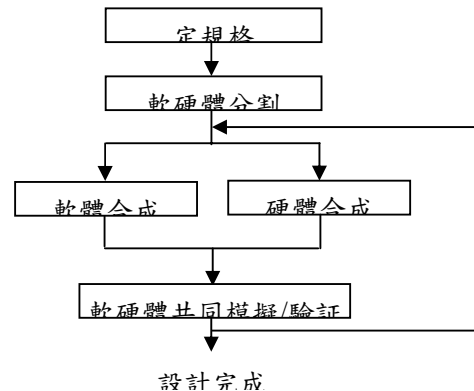


圖 1：SOC 嵌入式系統的設計流程

貳、目標

目前所看到的研究多是從架構設計、硬體電路、作業系統等方面對 SOC 的功率消耗進行改善[7, 8, 9, 10]。其中包括時脈的開控、 V_{dd} 電源的定時關閉、匯流排的架構設計、記憶體讀寫策略、及動態的電源管理等。我們認為在設計嵌入式系統時，每一個環節都應該盡力避免無謂的功率消耗，而在設計過程中，開發應用程式的階段顯然是較被忽略的一環。因此我們針對「如何從應用程式對嵌入式系統進行節能」進行較深入的研究。

我們針對以 SOC 晶片配合週邊介面的整體

嵌入式系統進行功率消耗的研究，在這樣的嵌入式系統中，硬體架構及電路已然成形，作業系統及相關驅動程式亦已決定，圖二(a)之陰影區所示者即為已定案之部份。我們專門針對圖二(a)中未加陰影的部份(即應用程式)進行功率消耗的改進。另外，雖然在實務上常透過系統組態的規劃使系統的性能及功率消耗達到最佳化，如圖二(b)粗線所框之區域即為經過組態規劃的結果，但因為這道步驟和應用程式的寫法無關，所以在這項研究中也將系統組態的規劃列入考慮。

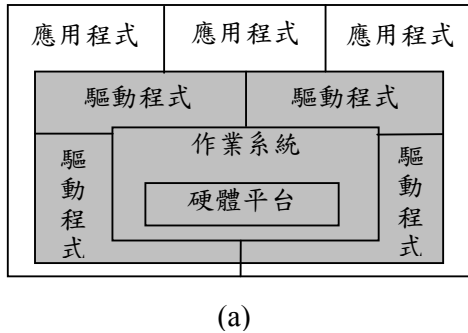


圖 2：硬體平台、作業系統、驅動程式、及應用程式

所以，假定系統電路中有 n 個功能方塊，若將各功能方塊視為不同的電路模組 b_i ($i=1$ to n)，且各電路模組的功率消耗為 $p(b_i)$ ，我們的目標就是透過應用程式將整個嵌入式系統的功率消耗 P_e 降至最低，即

$$P_e = \min \left(\sum_{i=1 \text{ to } n} p(b_i) \right) \quad (1)$$

參、實驗方法及過程

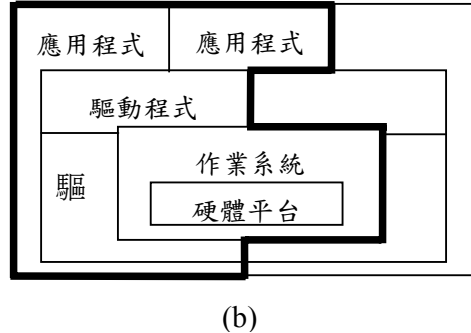
一、軟硬體平台

我們的實驗平台採用市售的 SOC 開發系統[1, 2]，硬體部份包括 S3C4510 CPU、SDRAM、Flash ROM、擴充記憶體、Ethernet、UART、USB、CODEC (語音模組)、LCD 模組、CF 卡、及影像模組，作業系統則是採用 *uCLinux*。編譯工具以 GNU Tool Chain 為主。另外，還利用電流量測模組針對 SOC 開發系統中不同的功能方塊進行個別測試。電流量測模組之等效電阻為 Z ，用來將電流值轉成電壓訊號並以數位電表(DVM)顯示，所顯示之電壓值 V 和待測模組之消耗電流 I 呈 $V = I \times Z$ 的關係，若將硬體平台之電路以模組的形式表示，而且我們準備量測某個電路的

消耗功率，則待測電路、其它電路模組、電流量測模組、及顯示用之數位電表間的關係可以用圖 3 表示，其中的 V_{dd} 及 Gnd 分別表示電源的正負端。

二、實驗過程

圖 4 為本項研究之應用程式開發流程，透過 `make` 檔的協助，編譯程式編譯連結 `library` 可得出 `image` 檔，顯然這個流程和典型的應用程式開發過程一樣。



我們所進行的實驗基本上分成兩個階段：功率消耗的分析及應用程式最佳化。在第一階段量測不同的電路模組在待機狀態和工作狀態下所耗用的功率，同時監讀待測模組在工作狀態下對其它模組的影響。就第一階段所得的結果進行分析後，我們就可以進入第二階段，也就是應用程式的最佳化。

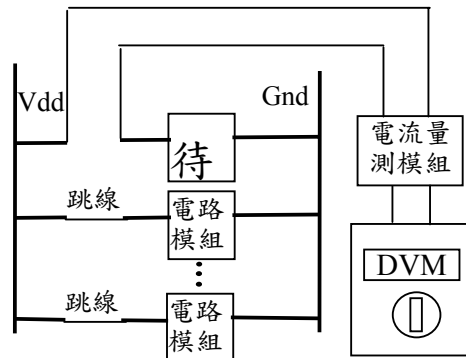


圖 3：電流量測模組與實驗平台

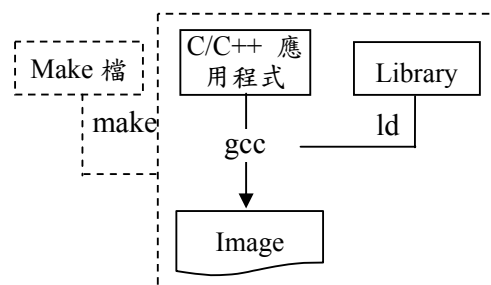


圖 4：應用程式之開發流程

應用程式最佳化的主要策略包括：(a)耗電量大的工作優先完成，(b)所有的工作盡量以較短的時間完成，(c)所有的工作盡量以較省電的方式完成。

讓耗電量大的工作優先完成的基本原則就是一旦該工作啟動後，就讓它一氣呵成，盡量不要打斷它。因為無線傳輸所耗用的功率在整個系統的功率消耗中佔了相當的比率，因此這項策略對於無線傳輸而言尤其重要。

要讓工作儘快完成的方法包括：

- (1). 以資料代碼代替資料 例如在無線傳輸前，資料接收端預建資料檢視表，進行傳輸時所傳送的是資料代碼而非資料，資料接收端收到資料代碼後再以資料代碼為索引，到資料檢視表中取得正確的資料。
- (2). 以巨集代替副程式 減少叫用副程式時在堆疊存放返回位址，及結束副程式時由堆疊取出返回位址所耗用的時間。
- (3). 以較快的指令代替較慢的指令 例如在數學運算時，乘除指令所耗用的時脈數遠比加減指令大，所以盡量用加減指令代替乘除指令。此外，有些乘除的運算可以直接用移位指令完成，而移位指令的速度不但比乘除指令快很多，也比加減指令還快。
- (4). 減少對記憶體的存取 將經常存取的變數置於暫存器中，而非記憶體中，以 C 語言為例，以下的宣告即可將變數 x 放在暫存器中。

```
reg int x;
```

當某項工作以較短的時間完成時，事實上它就是以較省電的方式完成。但我們實現第三項策略的方法還包括：

- (1). 以較省電的路徑完成任務 例如，可以進行有線傳輸時就以有線傳輸代替無線傳輸。
- (2). 減少不必要的 LCD 顯示 這是因為

LCD 電路模組在顯示圖形或字元時所耗用的功率比不顯示時高。

- (3). 以快速演算法代替慢速演算法 若無法找到同樣精確且速度較快的演算法，只要不影響系統的運作，則速度較快但精確度稍差的演算法也是很好的選擇。

以上三個讓系統省電的方法都可以經由應用程式來完成。

圖 5 為本項研究所使用之硬體平台。圖中包括 S3C4510-based 電路板、ICE、及 DVM，應用程式編譯成 IMAGE 後係載入電路板上的 SDRAM 中。



圖 5：以應用程式進行功率消耗所用之平台

肆、結果與討論

我們針對 SOC 開發系統中串列介面、CODEC、及 Ethernet 介面、LCD 顯示等功能方塊分別測試。在不同條件下對各功能方塊所量測之電流值以 mA 為單位分別表示如下。

表一 串列介面

串列介面	開機後待機狀態	連續傳送資料	等待接收資料	
			輪詢模式	中斷模式
UART	186	186	186	186
CPU	840	1060	1064	1030
記憶體	91	431	444	426

表二 CODEC 介面

CODEC 介面	開機後待機狀態	程式啟動	開始錄音	開始放音
CODEC 模	34	50	51	51

組				
CPU	841	1058	1062	1062
記憶體	90	430	428	428

表三 Ethernet 介面

Ethernet 介面	開機後待機狀態	接收封包	傳送封包
Ethernet 模組	673	674	674
CPU	840	1040	1038
記憶體	90	430	429

由表一至表三中，我們可以發現各功能模組一旦由待機狀態進入工作狀態，功能模組本身、CPU、記憶體的耗電量幾乎是固定值。

改善方法之一就是使不用的電路回到待機狀態。以表四的 LCD 顯示功能為例，由標示陰影的欄位可看出，經由程式來強迫 LCD 恢復待機可節省 LCD 模組之功率消耗達 1/3。

表四 LCD 顯示

LCD 顯示	開機後待機狀態	填滿字元	填滿空格	填滿字元數 秒後強迫恢復待機
LCD 模組	12	18	17	12
CPU	841	1045	1045	1032
記憶體	90	438	439	442

伍、結論

設計嵌入式系統時，每一個環節都應該盡力避免無謂的功率消耗，而在設計過程中，開發應用程式的階段顯然是較被忽略的一環。因此我們針對「如何從應用程式對嵌入式系統進行節能」深入研究。實驗的結果也證實我們所作的努力的確可以有效減少嵌入式系統的功率消耗。

功率消耗之研究是沒有止境的，本項研究主要是從應用程式對功率消耗作改善，初步的結果顯示適時關閉不用的電路確有效果。未來將進一

步根據所提的省電策略，從應用程式的層面改善功率的消耗。

陸、參考文獻

1. 新華電腦股份有限公司，內核嵌入式 SOC 實作，全華科技圖書股份有限公司，2004。
2. 新華電腦股份有限公司，ARM SOC 原理-以 ARM7TDMI S3C4510B 為例，全華科技圖書股份有限公司，2004。
3. B. Tabbara, A. Tabbara, A. Sangiovanni-Vincentelli, "Function/ Architecture Optimization and Co-Design of Embedded Systems," *Kluwer Academic Publisher*, 2000.
4. H. Chang, L. Cooke, M. Hunt, G. Martin, A. McNelly, L. Todd, "Surviving the SoC Revolution," *Kluwer Academic Publisher*, 1999.
5. T. K. Tan, A. Raghunathan, N. K. Jha, "Software Architectural Transformations: A New Approach to Low Energy Embedded Software," *Proceeding, Date*, 2003.
6. <http://public.itrs.net/Files/2003ITRS/Home2003.htm>, *ITRS 2003 Edition*.
7. F. Vahid, T. Givargis, "Embedded System Design," *John Wiley & Sons, Inc.*, 2002.
8. Wiberg P. A. and Bilstrup U., *Wireless Technology in Industry-Applications and User Scenarios, Emerging Technologies and Factory Automation, 8th IEEE Int'l Conference*, Vol. 1, pp 123-131, 2001.
9. K.-J. Chang, T.-F. Chen, J.-S. Shen and M.-W. Chang, "Fast Run-Time Power Monitoring Methodology for SOC Embedded System," *15th VLSI Design/Cad Symposium*, 2004.
10. T. Okuma, Y. Cao, M. Muroyama, H. Yasuura, "Reducing Access Energy of On-Chip Data Memory Considering Active Data Bitwidth," *Proceedign of ISLPED*, 2002.

(本報告已改寫成論文發表於資訊管理國際研討會，2005/6/3，台灣苗栗)

晶片系統架構技術及開發平台研究之推動(3/3)-附屬計畫七
無線感測器網路基礎架構設計與實作—智慧型辦公室/家庭應用

**Design and Implementation of Wireless Sensor Network
Infrastructure—A Smart Office/Home Application**

計畫編號：NSC 93-2215-E-002-010/007

執行期限：93/8/1~94/7/31

計畫主持人：蕭榮修 國立台北科技大學電子系

計畫參與人員：劉新盛、呂詠新

一、摘要

智慧型環境係依據情境資訊的變化讓使用者與環境產生互動，因此亦稱為情境感知環境。建構一個智慧型環境應用系統的關鍵在於情境資訊的擷取、表示、處理與儲存。目前有關此領域的研究，都隨著應用的不同，而各自研發其獨特的系統架構，因此要研發一個新的應用系統，普遍缺乏可供利用的軟/硬體架構。

本計畫提出一個適合辦公室/家庭應用的層級架構，以方便其他同類型應用之研發。此系統的架構包含：情境擷取與傳輸、情境資訊處理及應用三個層級。依據此架構，我們使用 Tmote 感測器模組及 TinyOS 嵌入式作業系統作為無線網路平台，擷取與傳輸情境資料。為了測試此架構的可用性，我們設計了一個家庭自動化的劇情，並實做一個情境感知家庭離型。

關鍵字：感測器網路 (sensor network)，情境感知 (context-aware)，智慧型辦公室/家庭

Abstract

Smart environments let user interaction with the environment according to the updated context information, which is also known as context-aware environment. The key to build a smart environment application is in acquisition,

representation, handling, and storing of context information. Most context-aware researches develop their proprietary architecture with their available technology. Therefore, the lack of a suitable infrastructure to support building a context-aware application is a major problem.

This project proposes a layered system infrastructure. To make it easier for context-aware office/home designers develop their applications. The system infrastructure consists of: context acquisition, context handling, and application layers. Based on this infrastructure, we use Tmote sensor module and TinyOS embedded operating system as the wireless network platform to acquire and transmit context data. In order to test the usability of the system infrastructure, we design a home automation scenario and implement a context-aware home.

Keywords: sensor network, context-aware, smart office/home

二、動機與目的

Weiser [8] 提出普及運算 (ubiquitous computing) 的概念，主張電腦不只是可以無所不在，而且要存在於無形之中。此後隨著無線感測器網路的技術漸趨成熟，興起了此

一領域的研究熱潮。

情境 (context) 意指特徵化一個體(entity) 狀況的資訊，其與使用者和環境之間的互動有關；以情境來提供資訊及 (或) 服務給使用者的系統則稱為情境感知 (context-aware) 系統 [2]。於無所不在計算 (pervasive computing) 領域中，已有許多情境感知的應用之研究，然而各個應用都有其獨特的架構，缺乏一個適當的基礎架構支援，要實做一個應用系統，則頗費周章。本研究的主要目的即提出一個以感測器網路為基礎的軟／硬體基礎架構，以方便小型辦公室／家庭應用之建構。

為了驗證本架構之可用性，我們使用 Moteiv 的 Tmote 感測器模組 [6]，及 TinyOS [7] 嵌入式作業系統作為無線網路平台，實做一個情境感知家庭系統。

三、 文獻探討

有關情境感知的研究甚多，我們則專注於情境模型 (context model)、情境資訊管理基礎架構 (infrastructure) 及情境感知家庭 (context-aware home) 等議題。

Henricksen 等人 [5] 提出一套個體關聯 (ER) 的情境模型化概念，此模型化的概念是以物件的方式 (object-based approach) 提供一個形式上的基礎 (formal basis) 作為情境資訊的屬性之表示 (representation) 與推理 (reasoning) 之用，情境資訊是以個體為核心組織而成，這些個體可以是實體的 (physical) 或概念的 (conceptual) 物件，例如人、裝置、空間等；個體的特質則以屬性 (attribute) 表示之，例如人名、裝置名稱、地點等，個體與屬性相連結形成關聯 (association)，這種情境模型適合以關聯式資料庫 (relational database) 來管理。

Capra [1] 等人提出映射的中介軟體 (reflective middleware) 之情境資訊管理解決方案，此中介軟體建構於網路作業系統之上，藉著映射 (reflection) 及後設資料 (metadata) 提供變動的情境資訊給上層的應用服務請求。

Dey 等人[3]為智慧型的環境提出一個軟體的基礎架構，此架構引用層級的抽象概念，以一個情境處理程式 (widget) 來處理情境資訊，每一個情境處理程式負責處理一個感測器，它隔離應用程式與感測器，隱藏了底層感測器的細節，方便應用程式設計者建構應用系統。另外有一個伺服器 (server) 負責聚集所有情境處理程式所擷取的資訊，然後供給解譯器 (interpreter) 作為決策之用。

四、 情境感知辦公室／家庭應用之設計

(1) 情境感知家庭的劇情

研發一個情境感知家庭，首先必須擬定一套劇情 (scenario)，劇情涵蓋的範圍非常廣泛，視個人的需求和喜好而定，我們僅選定家庭自動化 (home automation) 的一小部份來實做。劇情一如下：

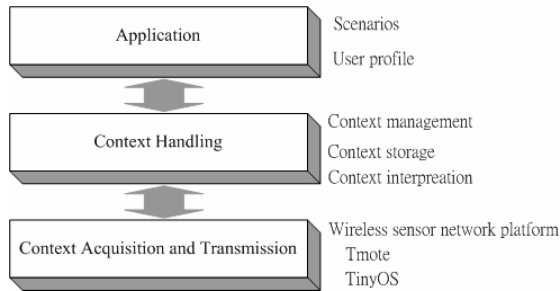
下午 7 點，大偉和珍妮回家，進入客廳，燈光自動開啟；此時室溫為攝氏 30 度，冷氣自動開啟；大偉與珍妮坐在沙發上，電視機自動打開。

(2) 系統架構

我們的設計主要著重於兩個部分：一個是整個系統的軟、硬體架構，主要目標是提出一個具有可調適的 (adaptable)、可擴充的 (extensible) 及可延展的 (scalable) 架構；另一個部分是提供一個合適的情境資訊處理架構，方便智慧型家庭研發者擷取、表示、處理、儲存及使用情境資訊。

本系統的整體架構如圖一所示的層級堆疊 (layer stack)，最底層為情境擷取和傳輸 (context acquisition and transmission)，我們選用 Moteiv 的 Tmote 智慧型感測器模組，此模組主要包含一個低功率消耗的微控器—TI MSP430，一個 Chipcon CC2420 無線通信晶片，和濕度、溫度、光等感測器。在此模組之上，我們安裝一個嵌入式作業系統—TinyOS，它扮演硬體抽象化的功能，提供一組使用者應用介面 (API)，以方便研發者設計程式。中間的情境處理 (context handling)

層包含情境資料庫和解譯程式 (interpreter)，負責管理、儲存情境資訊及情境的推理。應用層 (application) 為使用者自訂的劇情、設定及使用者介面。



圖一、系統的整體架構

(3) 情境感知模型

一個情境感知基礎架構需要一個適當的情境模型來表示，我們採用 ER 的方法建立情境模型，並以一階邏輯述詞 (first-order logic predicates) 來推理劇情。

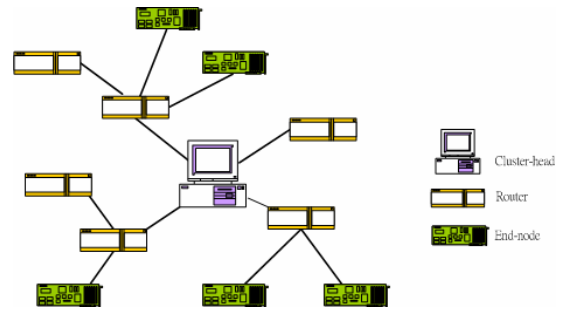
(4) 無線感測器網路

感測器網路在建立之初，必須先將每個感測節點設定成感測器 (sensor) 或致動器 (actuator)，接著感測器便搜尋服務合適的致動器，建立一張配對表，以便將來重置後，不需再搜尋，感測器直接控制對應的致動器。

感測器網路的拓撲採用以叢集為基礎 (cluster-based) 的架構，整個網路共有三種角色，分為叢集頭 (cluster-head)、路由器 (router) 與終端節點 (end-node) [4]，感測器與致動器皆為終端節點。終端節點在加入網路時，便將要求加入網路的訊息傳送給路由器，由路由器轉發訊息給叢集頭。叢集頭再依發送此訊息的節點是何角色及其所需的服務，指定另一終端節點為其配對的對象，建立起兩者的連線，將來兩者重置後，感測器就可以直接控制對應的致動器。

本系統的感測器網路之角色示意圖如圖二所示，叢集頭內部建有完整的路由表

(拓撲的配置)，並負責將路由器所發送過來之感測器的服務要求，交由路由器轉發給合適的致動器，致動器接收到要求便做出對應的動作。路由器接收到終端節點所發送過來的訊息後，便負責轉發給叢集頭或是適當的終端節點。終端節點則負責感測並將感測的數據發送給路由器，或是接收路由器所轉發的要求，進行對應的動作。由於感測器亦為致動器，所以接收到訊息時，必須做出適合角色的動作。



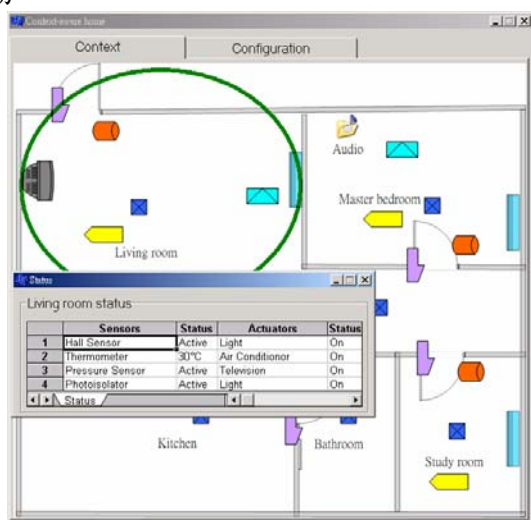
圖二、角色示意圖

終端節點負責情境資料的擷取或是致動，功能較簡單，並且使用電池作為電力來源。路由器負責轉發訊息，並且需維護網路的拓撲 (新增、刪除與更新)，所以傳輸模組功能必須較終端節點完善，並且以市電為電力來源，以即時回應終端節點與叢集頭的訊息要求。叢集頭必須維護路由表，並且依使用情境來分配工作，所以需要功能完善的傳輸模組與運算能力。在本系統中，終端節點和路由器皆使用 Tmote 感測器模組，而叢集頭則使用桌上型電腦。

五、 情境感知家庭實作

我們以擬定的劇情配置無線感測器網路。如劇情一，當大偉和珍妮打開大門時，啟動裝設在門邊的霍爾開關，並走過光遮斷器的偵測區，這些情境訊息便發送給鄰近的路由器，路由器儲存並轉發，直到訊息抵達叢集頭為止。叢集頭為書房內的個人電腦，其具備整個家庭的情境設定，並依據目前的

情境推理，找尋適當的致動器，發現客廳的燈為其對應的致動器，便發送訊息給客廳的燈。又客廳的溫度感測器偵測目前的室溫，亦以上述的方式通報叢集頭，叢集頭發現客廳的溫度超過臨限值，就再度發送訊息給客廳的冷氣機，啟動空調以降低室溫。當大偉和珍妮坐在沙發上時，啟動了沙發的壓力感測器，壓力感測器會透過上述的機制通報叢集頭，叢集頭再依相同的方式通報電視機啟動。



圖三、 情境感知家庭的監看畫面

圖三為情境感知家庭的監視畫面，只要點選任一房間，就可監看各個房間的狀態及裝置的運作情形。此外使用者亦可按下組態設定 (configuration)，依各人的習慣或喜好初始化各個裝置是否啟動 (enable) 及其工作時間。

六、 結論與未來工作

本研究提出一個建構情境感知的應用架構，並建構一個情境感知的家庭自動化雛形，測試此架構之可用性。依據此架構的抽象概念，研發者可專注於系統的應用層，而不需複雜的情境感知及無線感測器網路之理論背景，可縮短系統研發的時程。

有關情境感知的研究尚在起步的階段，因此缺乏普遍性的研發工具與架構可供利

用。本研究僅針對小型辦公室／家庭提出一個可行的架構及測試，所做的應用也僅為家庭自動化的一小部份，未來將擴大應用的規模及測試的範圍，使情境感知的辦公室／家庭可以具體實現。

七、 參考文獻

- [1] L. Capra, W. Emmerich, and C. Mascolo, "Reflective Middleware solutions for context-aware applications," *Proceedings of REFLECTION 2001. The Third International Conference on Metalevel Architectures and Separation of Crosscutting Concerns*, 2001.
- [2] A. K. Dey, *Providing Architectural Support for Building Context-Aware Applications*, PhD thesis, College of Computing, Georgia Institute of Technology, December 2000.
- [3] A. K. Dey, D. Salber and G. D. Abowd, "A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications," *Human Computer Interaction*, Vol 16, 2001.
- [4] A. Hac, *Wireless Sensor Network Designs*, John Wiley & Sons, 2003.
- [5] Henricksen, J. Indulska, and A. Rakotonirainy, Modeling context information in pervasive computing systems. *Proceedings of 1st International Conference on Pervasive Computing*, (Zurich, Switzerland): 167 – 180, 2002.
- [6] "Tmote sky," Available in the site of: <http://www.moteiv.com/>
- [7] "TinyOS," Available in the site of: <http://www.tinyos.net/>
- [8] M. Weiser, "[The Computer for the Twenty-First Century](#)," *Scientific American*, pp. 94-10, September 1991.

晶片系統架構技術及開發平台研究之推動(3/3) -附屬計畫八

CCD/CMOS 彩色影像插補點晶片設計

CCD/CMOS Color Interpolation Chip Design

計畫編號：NSC 93-2215-E-002-010/008

執行期限：93/8/1~94/7/31

計畫主持人：黃奇武 副教授 台灣師範大學工業教育系

計畫參與人員：陳偉昕、陳映儒、王奕淳

一、中文摘要

數位相機/攝錄影機多採用單一 CCD/CMOS 感測器,只取得 1/3 色彩資訊,需採用色彩插補點 (color filter array interpolation)演算法進行色彩內插,使所有圖素均有 R/G/B 三種顏色。為了改善影像模糊,色彩失真或假色的不良品質現象,現有的演算法大多存在許多分支條件控制流程,根據許多影像特徵,決定套用那一個公式或演算法進行色彩插補。

這些演算法大多由 ARM+DSP 的環境來執行,但這些演算法無法充份利用 DSP 之單一指令多重資料流的特性,進行快速運算,造成整個彩色影像後處理的瓶頸。本研究首先找尋較好的 CFA 演算法,先在 PC 環境上模擬,其次成功的移植到 ARM embedded system 環境,體會有限資源的慢速執行,然後利用 FPGA 電路的可平行運算特性,將 CFA 演算法實作到 FPGA 晶片上,對照在 ARM embedded system 環境的運算輸出完全無誤。達成快速處理”改善影像模糊,色彩失真或假色的不良品質現象”目標。

Abstract

Due to the Bayer pattern, only one color channel for each pixel is available for digital still/video cameras. The final picture relies on color filter array (CFA) interpolation to generate the full R/G/B color data for the other two missing channels. The typical problems in typical CFA interpolation algorithm/hardware include image blurring, color artifacts, and false colors. To solve these problems,

most of available algorithm analyzes the features of the image pixels and apply many heuristic rules to control the program running into several conditional branches.

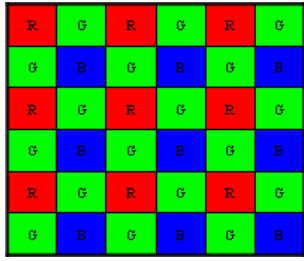
This type of conditional branch algorithm can not be applied onto a platform with digital signal processor (DSP), which is usually designed as single instruction multiple data (SIMD) architecture. The SIMD can only handle the data without any conditional branches. Hence CFA interpolation is usually the bottleneck of the color image pipeline. In this research, we design a fast CFA algorithm as well as implementing on FPGA hardware platform. The experimental results show that the proposed CFA algorithm as well as hardware implementation are fast and the color distortion is very small.

Keywords

Bayer pattern、CFA、FPGA、ARM、embedded system

二、動機與目的

以 RGB domain 來說,可見光是由 RGB 三種頻譜成份組成,不同比例的 RGB 可構成不同的顏色,目前市面上大眾化的數位相機或攝錄影機都是單一 CCD/CMOS 感測器,每個像素(pixel)只能感測一種頻譜成份,必須搭配彩色濾波器陣列(color filter array, CFA)一起使用,如圖一,目前通用的彩色濾波器陣列為 Bayer CFA pattern[2],特色在於感測器上感測 G 頻譜成份的像素數量是 R 頻譜成份或 B 頻譜成份的兩倍。因此需要有一個彩色插補點演算法來計算出其他的頻譜成份,以得到完整的影像。



圖一 Bayer CFA pattern

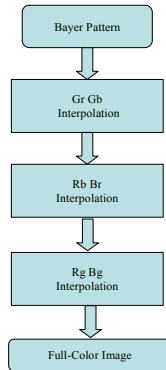
傳統的雙線性插補法運算簡單，是最常被使用的方法，但會讓影像中的邊緣模糊，並且產生色彩失真與假色的不良現象。

數位相機或數位攝錄影機上的影像處理系統包含色彩插補點、自動白平衡 (auto white balance, AWB)、色彩校正(gamma correction)與邊緣強化(edge enhancement)等影像處理方式，而傳統的色彩插補點演算法多利用 ARM+DSP 做處理，由於無法充分利用 DSP 單一指令多重資料流的特性，因此無法進行快速運算，本研究目的在尋求解決方案。

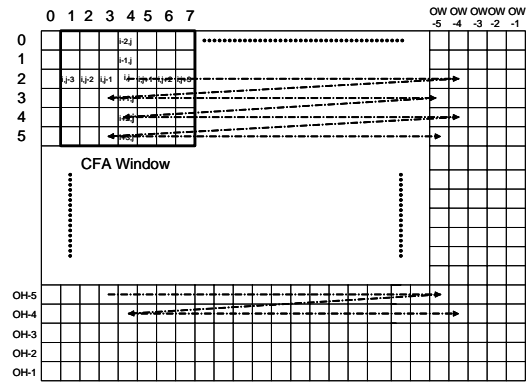
三、研究方法與成果

1. 彩色影像插補點演算法

本研究的基礎，建立於曾指導及進行參考文獻[1][5]的研究，為改善 CFA 演算法產生的色彩失真與假色，我們的構想為利用前次已插補出的資訊，做為後續插補的參考，演算法流程如圖二所示，輸入一張 Bayer pattern，先插補出 G 值，再利用已插補出的 G 值插補出 R 值與 B 值。圖二每一步驟的 CFA window 掃瞄流程如圖三所示。圖二的詳細演算法如下：

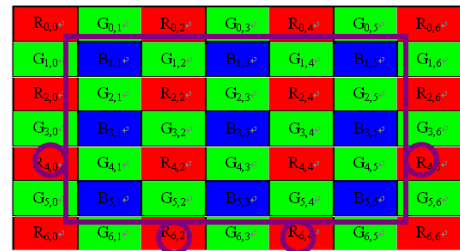


圖二 彩色影像插補點演算法流程



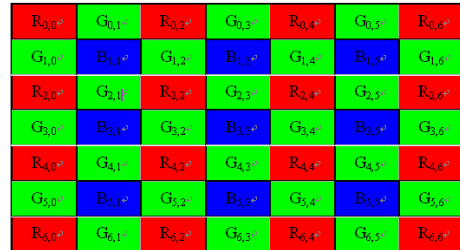
圖三 CFA window 掃瞄流程

步驟一：插補 Gr、Gb (r pattern 缺少的 G 與 b pattern 缺少的 G)，如圖四所示。



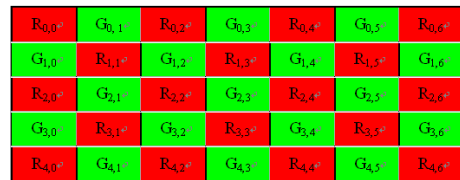
圖四 插補 Gr Gb 之 CFA Window

步驟二：插補 Rb、Br (b pattern 缺少的 R 與 r pattern 缺少的 B)，如圖五所示。



圖五 插補 Rb Br 之 CFA Window

步驟三：插補 Rg、Bg (g pattern 缺少的 R 與 B)，如圖六所示。



圖六 插補 Rg Bg 之 CFA Window

因為 Bayer pattern 上的 r 與 b 的相對位置是一樣的，因此插補 Gr 的部分與 Gb 類似，以下以插補 Gb 為例。

如圖四，先使用一個 6*7 的 window，判斷 Edge 的方向為水平或垂直， E_{hor} 、 E_{ver} 為用一個

Sobel 運算子算出的值，越大代表是 Edge 的可能性越大，兩者作比較後，得到最有可能的 Edge 方向，當 E_{hor} 、 E_{ver} 兩者接近時，我們用近似度 D_{hor} 和 D_{ver} 進一步判斷 Edge 的方向。

$$\begin{aligned} & \text{if}(E_{hor} - E_{ver} \geq \text{threshold}) \\ & \quad G_{3,3} = I_{hor}; \\ & \text{else if}(E_{hor} - E_{ver} \leq -\text{threshold}) \\ & \quad G_{3,3} = I_{ver}; \\ & \text{else} \\ & \quad D_{hor} = |B_{3,1} - B_{3,3}| + |B_{3,5} - B_{3,3}| + |G_{3,2} - G_{3,4}| \\ & \quad D_{ver} = |B_{1,3} - B_{3,3}| + |B_{5,3} - B_{3,3}| + |G_{2,3} - G_{4,3}| \\ & \quad \text{if}(D_{hor} < D_{ver}) \\ & \quad \quad G_{3,3} = I_{hor}; \\ & \quad \text{else if}(D_{hor} > D_{ver}) \\ & \quad \quad G_{3,3} = I_{ver}; \\ & \quad \text{else} \\ & \quad \quad G_{3,3} = (I_{hor} + I_{ver})/2 \end{aligned}$$

其中：

$$\alpha = 0.5$$

$$\text{threshold} = 1024$$

$$E_{hor} = \left(|G_{2,2} - \tilde{G}_{4,2}| + 2 \times |G_{2,3} - G_{4,3}| + |G_{2,4} - \tilde{G}_{4,4}| \right) \times 128 / \left[(G_{2,3} + G_{4,3}) / 2 \right]$$

$$= \left(|G_{2,2} - \tilde{G}_{4,2}| + 2 \times |G_{2,3} - G_{4,3}| + |G_{2,4} - \tilde{G}_{4,4}| \right) \times 256 / (G_{2,3} + G_{4,3})$$

$$E_{ver} = \left(|G_{2,2} - G_{2,4}| + 2 \times |G_{3,2} - G_{3,4}| + |G_{4,2} - \tilde{G}_{4,4}| \right) \times 128 / \left[(G_{3,2} + G_{3,4}) / 2 \right]$$

$$= \left(|G_{2,2} - G_{2,4}| + 2 \times |G_{3,2} - G_{3,4}| + |G_{4,2} - \tilde{G}_{4,4}| \right) \times 256 / (G_{3,2} + G_{3,4})$$

(\tilde{G} 代表未知的G,以近似度的關係,預測出一個近似值)

$$\tilde{G}_{4,2} = \begin{cases} (G_{4,1} + G_{4,3}) / 2, & \text{if } D_{hor,2} \leq D_{ver,2} \\ (G_{2,2} + G_{2,4}) / 2, & \text{if } D_{hor,2} > D_{ver,2} \end{cases}$$

$$\tilde{G}_{4,4} = \begin{cases} (G_{4,3} + G_{4,5}) / 2, & \text{if } D_{hor,4} \leq D_{ver,4} \\ (G_{3,4} + G_{3,4}) / 2, & \text{if } D_{hor,4} > D_{ver,4} \end{cases}$$

$$D_{hor,2} = |G_{4,1} - G_{4,3}| + |R_{4,0} - R_{4,2}| + |R_{4,4} - R_{4,2}|$$

$$D_{ver,2} = |G_{3,2} - G_{3,4}| + |R_{2,2} - R_{4,2}| + |R_{6,2} - R_{4,2}|$$

$$D_{hor,4} = |G_{4,3} - G_{4,5}| + |R_{4,2} - R_{4,4}| + |R_{4,6} - R_{4,4}|$$

$$D_{ver,4} = |G_{3,4} - G_{3,4}| + |R_{2,4} - R_{4,4}| + |R_{6,4} - R_{4,4}|$$

$$I_{hor} = 0.5 \times [G_{3,2} + 0.5 \times \alpha \times (B_{3,1} - B_{3,1})] + 0.5 \times [G_{4,3} + 0.5 \times \alpha \times (B_{3,3} - B_{3,3})]$$

$$= (G_{3,2} + G_{4,3}) / 2 + \alpha \times (-B_{3,1} + 2B_{3,3} - B_{3,3}) / 4$$

$$= G_{3,2} / 2 + G_{4,3} / 2 + B_{3,3} / 4 - B_{3,1} / 8 - B_{3,5} / 8$$

$$I_{ver} = 0.5 \times [G_{2,3} + 0.5 \times \alpha \times (B_{3,3} - B_{3,3})] + 0.5 \times [G_{1,3} + 0.5 \times \alpha \times (B_{3,3} - B_{3,3})]$$

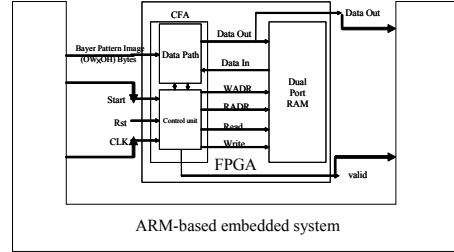
$$= (G_{2,3} + G_{1,3}) / 2 + \alpha \times (-B_{1,3} + 2B_{3,3} - B_{3,3}) / 4$$

$$= G_{2,3} / 2 + G_{1,3} / 2 + B_{3,3} / 4 - B_{1,3} / 8 - B_{5,3} / 8$$

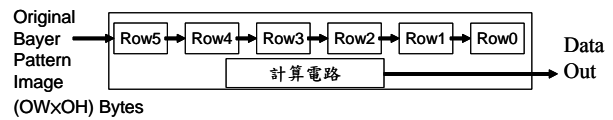
2.彩色影像插補點硬體架構

我們使用 XILINX XC2S300E FPGA 來實現 CFA 演算法，其硬體架構如圖七所示，data path 為插補電路的主要部分，每次輸入一個 pixel 的資料，一個資料為 1 byte，OW 是原始影像寬度，OH 是原始影像長度，因此總共要輸入 OW*OH 個資料。Control Unit 用以：1.控制 Bayer pattern image 以 byte 形式進入 CFA 電路作處理，當 Start 為 High 期間，以 CLK 正邊緣存入 data path 的 register files(CFA window)。2.以 valid 正邊緣將 Data Out 的 byte 資料 strobe 到外界 (ARM-based embedded system)。Dual Port RAM 則是用來暫存一系列影像(已插補的像素)待下一列使用。每當 CFA 插補出一個值出來，Write 信號就把 Data Out 存至 Dual Port RAM 中的 WADR 位址。每當 CFA

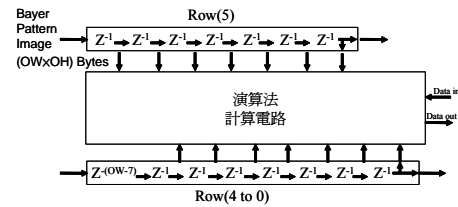
需要使用到前列已插補過的資料時，Read 信號就從 Data In 讀取 Dual Port RAM 中 RADR 位址的內容，傳送給 CFA 做運算。因為色彩插補點電路有單數列與偶數列兩種工作模式，因此需要 Control Unit 來做模式切換。



圖七 色彩插補點硬體架構圖



圖八 CFA data path



圖九 CFA window 與計算電路的介面

圖九的 Z^{-i} 代表 delay i 個週期的元件。圖八所示的 CFA data path，將 Bayer pattern image 送至 CFA window 中，經 Row5, Row4 一直到 Row0。計算電路從 CFA window 的多個 Z^{-1} 元件取得 byte 資料，插出一個 (i,j) 位置的一種顏色，以 byte 形式送到 Data Out。

3.實驗成果



圖十 左為原始影像，右為插補結果

本研究先找尋適當的 CFA 演算法，在 PC 環境進行差補，次移植到 ARM-based embedded system，充分領略了資源及速度的限制。FPGA 電路的可平行運算特性，可解決慢速問題，但其資源也有限，我們決定選取 320×240 的影像為例，

將 CFA 演算法實作到 FPGA 晶片+ARM-based embedded system 上,與在 PC 或 ARM-based embedded system 環境的運算輸出做對照,兩者完全一致。達成快速處理”改善影像模糊,色彩失真或假色的不良品質現象”目標。

如圖十左邊的原始影像,為一張 320*240 像素,每一像素為 24bits 色彩的照片。經抽取出這張影像的 Bayer pattern 後,送入插補演算法計算電路(或軟體)去處理,得到如圖十右邊的完整影像,插補過後影像的邊框會有綠色的情形是因我們略過邊框的插補計算,不影響效能。

表一所示為以圖十為例,將我們的演算法與 Pei[3]及 Tsai[4] 作 PSNR 比較的對照。因我們用 sobel 運算子來判斷是否為邊緣,作為插補選取的依據,因此在 PSNR 的比較,在邊緣區域的表現,我們的演算法比 Pei 及 Tsai 的演算法好。

表一 PSNR 比較表

	G	R	B
Pei	28.682158	25.234519	21.999913
Tsai	28.555966	15.683451	18.845231
proposed	28.394486	25.516790	22.537013

四、結論

本研究提出了一個改良型彩色影像插補點演算法,在插補時能考慮到已插補過的資訊,因此能有效改善色彩失真與假色的問題。為了使數位相機或攝錄影機的影像處理流程速度變快,本研究將演算法中可平行運算的部分實作成硬體晶片,改善了通用型 DSP 無法針對影像插補點加速的缺點。

我們目前設計的彩色影像插補點晶片,有兩種解題方案:其一使用如多輸入的 CSA 電路來解決部分長路徑問題,雖獲改善但仍有長路徑問

題,但因此法較易實現,為本研究目前採用的版本;另一使用 data pipeline 架構來解題,但其時序複雜,目前約完成插補 GrGb 70%的電路,已成功求出 I_{hor} , I_{ver} , 及兩組 D_{hor} 與 D_{ver} , 並已獲得正時,在這 70%電路中可有效改善長路徑問題。但因計畫執行時程有限,有關 data pipeline 的版本,就留待日後繼續進行研究。

五、參考文獻

- [1] 黃正誼, “內嵌 CMOS 影像感測擷取模組及影像濾波之 FPGA 平台設計,” 國立台灣師範大學工業教育研究所碩士論文, 2004。
- [2] Bryce E. Bayer, “Color imaging array,” *U.S. Patent 3,971,065*, Eastman Kodak Company, 1976.
- [3] S.-C. Pei, “Effective Color Interpolation in CCD Color Filter Array Using Single Correlation,” *IEEE Trans. Circuits and Systems for Video Technology*, vol. 13, no. 6, pp. 503-513, June 2003.
- [4] P.-S. Tsai, T. Acharya and A. K. Ray, “Adaptive Fuzzy Color Interpolation,” *Journal of Electronic Imaging*, vol. 11(3), pp. 1-24, July 2002.
- [5] 黃奇武, “晶片系統架構技術及開發平台研究的推動(2/3)-子計畫五, 網路監控之 ARM-Based 平台設計,” NSC92-2215-E-002-019 精簡報告, 2004。

晶片系統架構技術及開發平台研究的推動(3/3)-附屬計畫九

數位相機之影像辨認系統研究

Image Recognition for Digital Still Camera

計畫編號：NSC 93-2215-E002-010/009

執行期限：93 年 8 月 1 日至 94 年 7 月 31 日

主持人：高文忠 助理教授 國立台灣師範大學工業教育學系

計畫參與人員：高志忠、洪彥瑋、黃志祥

摘要

本計畫提出一個以膚色為基礎的自動人臉追蹤系統，此系統建構於一個包含有 CCD 影像擷取單元、ARM9 嵌入式微處理器、數為訊號處理器(DSP)等運算單元的數位相機平台。由於 DSP 運算資源已被即時影像壓縮(MPEG-4)所佔據，人臉追蹤模組必須充分利用 MPEG-4 壓縮過程中，所得的資訊來判斷人臉的位置也便讓此模組能在 ARM 上來執行，為達成即時追蹤的目的，我們提出多種改進技術使得整個人臉追蹤系統實現在 TI-DMS-320 DM310 系統晶片平台。其運算效能可達到每秒分析 10 畫面以上。

關鍵詞：人臉偵測、人臉辨識、DSP 影像處理

Abstract

This project proposes an automatic human face tracking system that can take MPEG-4 video and recognizes human faces in real time. The whole system is implemented on a commercialized digital camera which incorporates CCD imaging system, embedded microprocessor, and digital signal processor (DSP). The DSP computational resources are usually occupied by the MPEG-4 video compression job, so that the proposed human face tracking algorithm are assigned on the embedded microprocessor. To solve the problem of lower computation capability with the embedded microprocessor, the proposed algorithm must fully

utilize the results getting from video compression task and the tracking algorithm itself must be simple enough to achieve the real-time objective. The propose system are implemented on TI-DMS320 DM310 system-on-a-chip (SoC) platform. The experimental result shows that the tracking speed is fast enough that can be over than 10 frames/seconds.

Keywords: face detection, face recognition, video processing with DSP

1. 前言

治安與居家環境的安全是當今社會的重要課題，而一個有效的監視系統不但可以喝止宵小的犯罪，它也經常是破案的關鍵。然而傳統的監視系統並不具備人臉辨認與針對人臉的即時自動曝光和對焦控制能力，這使得所攝得的人臉影像模糊不清，尤其當嫌疑犯出現在角落時，所攝得的影像通常難以清楚辨識。因此，我們提出一個人臉影像追蹤系統，可以對人臉作即時性自動對焦以及自動曝光，使犯罪者的人臉部位影像更清晰明確，達到可以清楚辨識罪犯的目的。雖然人臉辨認已被廣泛研究，但是大多數的演算法過於複雜且都是以個人電腦(PC)來實現，因此這樣的監視系統過於龐大，不易移動，只可以針對固定位置作監控。

本研究進行一個以數位相機為平台的人臉辨認系統。我們首先建立了 MPEG-4 影音壓縮模

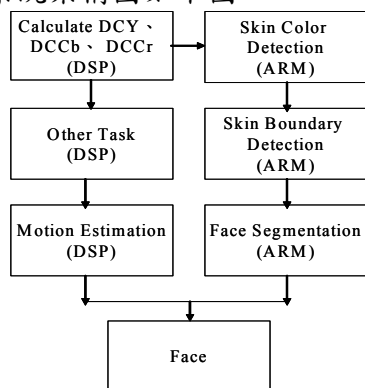
組，並研究了即時作業系統的排程，使人臉追蹤程式模組能充分利用嵌入式微處理器在多工排程的時間空檔進行人臉辨認。我們提出依全新的人臉追蹤技術，它包含了如下的技術：

- (a) 利用 Macro block 作(16x16 pixels)為基本運算單元，以降低運算複雜度。
- (b) 建立膚色之色彩模型，並利用查表法，加速膚色判斷
- (c) 提出一個新的物件輪廓搜尋演算法

整個系統採用現有實際用於數位相機之系統整合晶片(TI-TMS-320 DM310)作為開發平台的核心處理器。並且研究改進傳統循序式影像前處理之演算法，轉換成平行處理演算法，而充分利用數位相機 SoC 中之 ARM、DSP 與影像輔助運算器(iMX)等多重運算器之平行處理能力來建構即時與多工的影像處理系統。

2. 系統架構

本研究的系統架構主要是以 ARM 與 DSP 的分工運算，最後整合在 ARM 上做出最後人臉的確定。系統架構圖如下圖一：

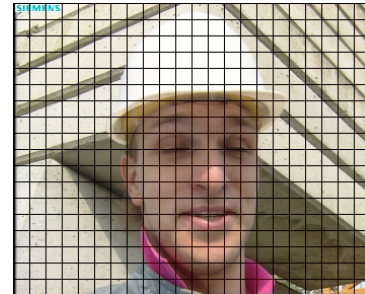


圖一、運算流程圖

如圖一所示，DSP 負責計算 MPEG-4 影像壓縮所須的所有計算。包括計算每個 frame 的每一個 Marco block 的 Y、Cb、Cr 的 DL 的 AL 成份。並計算 frame 與 frame 之間的移動向量估計(motion estimation)。由於這些計算的計算量相當大，DSP 系統已無剩餘的計算資源進行其它運算。所以本研究強調如何充分利用 DSP 所算得的資料，再進一步分析人臉的可能位置。

2.1 計算區塊 YCbCr 的平均值

針對每一個畫面 MPEG-4 壓縮過程中會先將 R/G/B 影像轉成 4:2:0 的 YCbCr 影像。目前將整個畫面如圖二所示切成許多 16x16 圖素為單元的 Macro block，以一 320x240 的影像經由此步驟處理後可得 20x15=300Macro blocks。而每個 Macro block 具有一個 DCY、DCCb、DCCr 的值，因為 ARM 微處理器根據這 300 個 DCY、DCCb、DCCr 計算膚色成分。



圖二、一張圖的所有 Macro Block

2.2 Skin Color Detection

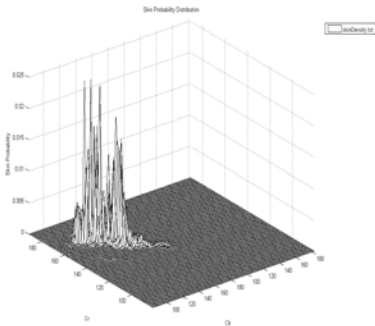
根據文獻參考以及我們實際的實驗結果，膚色的 Cb 範圍約落在(80,120)，Cr 範圍約落在(130,174)，所以以前大部份的論文以及期刊它們判斷膚色的方式，都是當該點的 Cb 以及 Cr 值同時落在設定的範圍內，就認定該點為膚色。但是根據我們以此膚色範圍來作膚色偵測發現很多近似的背景顏色都被認定為膚色，所以我們使用 2500 張無人臉的照片以及 125 張膚色樣本去統計 Cb Cr (0,255)內每一點為膚色(圖三)以及為非膚色(圖四)的機率。

膚色與非膚色之間的分布，事實上是有一些 overlap 的現象，因此單純用一個方框來定義膚色範圍無法得到較穩定的結果。定義的太窄，有些膚色會偵測不到，定義的太寬，很多背景色會被認定是膚色。所以，我們取(圖二)膚色相鄰四點機率相加的值 a，而(圖三)非膚色相鄰四點機率相加的值為 b

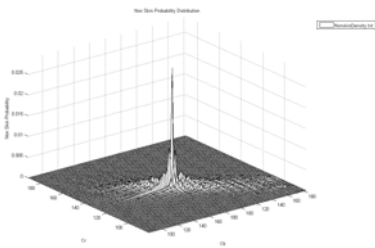
$$a = \frac{S(i,j) + S(i,j+1) + S(i+1,j) + S(i+1,j+1)}{4}$$

$$b = \frac{NS(i,j) + NS(i,j+1) + NS(i+1,j) + NS(i+1,j+1)}{4}$$

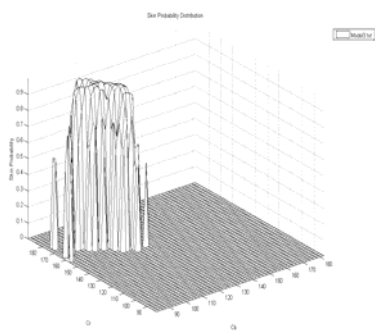
若 $a/(a+b) > b/(a+b)$ ，則我們認定該點為膚色，反之則否。所以我們會得到一個 128*128 大小的膚色機率分布圖(圖五)。該圖記錄著每一組 Cb Cr 是膚色或是非膚色。我們就根據這張膚色表來判斷該點是膚色的機率。



圖三、膚色資料統計圖



圖四、非膚色資料統計圖



圖五、正規化後的膚色機率統計圖

2.3 Skin Boundary Detection

輸入影像的每一點的 Cb Cr 經過上一個步驟的查表法後，我們將查得的結果記錄下來，膚色(1)或是非膚色(0)，這塊記憶體大小就跟影像經過 DSP 處理後，存放 DCCb DCCr 記憶體大小一樣

是 20*15。根據這張膚色圖，我們用改良型的迷宮演算法，沿著膚色邊緣收尋這塊膚色的最上、最下、最左、最右邊界，不管是走完一圈回到原點，或是最後退回原點，我們都可以找出這塊膚色的邊界。詳細演算法如下：

步驟一：定義方向優先順序，數值越小優先順序越高。

步驟二：確認下一點。當下面前三個法則都成立，我們就走到下一點，並且把這一點紀錄下來表示我們走過。若是其中一個不成立，換找下一個方向，若四個方向都不成立，表示走到死路，退回到前一點，繼續找下一個方向。

法則一：該點為膚色

法則二：該點之前沒有走過

法則三：該點周圍八點有一點為非膚色

法則四：在走的過程，我們會一直比較並紀錄最上、最下、最左、最右的邊界。

步驟三：若下一點為起點表示我們走回到原點，這時候就把搜尋到的邊界 Top、Bottom、Left、Right 記錄下來，並且把這個膚色區塊清除。

步驟四：繼續掃描剩下的膚色圖，搜尋下一個膚色區塊。當遇到膚色，回到 a，否則的話，結束膚色邊界偵測。

2.4 Face Segmentation

利用上一步驟所找出的邊界 Top、Bottom、Left、Right 畫出方框。

3. 研究結果

下列圖六與圖七是在以標準測試樣本在相機平台上所執行的結果。圖八為相機實際取像所得的輸出結果。由於我們是以 Macro block 為處理單位，所以切割位置或許會有一些邊緣部份無法被框入，但是這並不會影響我們的人臉追蹤。為了驗證系統效能我們分別定義正確檢測的數

量 N_c (correct Number)，檢測漏掉的數量 N_m (miss Number)，錯誤檢測的數量 N_w (wrong Number)。定義正確檢測率 DR (Detection Rate)和錯誤檢測率 FR (False Rate)為

$$DR = N_c / (N_c + N_m)$$

$$FR = N_w / (N_c + N_w)$$

單人情況下：正確檢測率為 90%，錯誤檢測率為 20.58%。

多人情況下：正確檢測率為 85.96%，錯誤檢測率為 40.2%



圖六、Fireman



圖七、News



圖八、實際輸出結果

4. 結論與建議

本研究提出一及具研究與應用價值的人臉追蹤系統雛形，未來將再利用 MPEG-4 的移動向

量以得到快速且正確的人臉位置，並改進自動對焦與自動曝光演算法，使之能以人臉位置作為主要標的物來進行取像。另一延伸主題併入人臉辨識功能使本系統更加完整。

5. 計畫自評

本計畫本來是要做到在相機上實作人臉辨識以及文字辨識。針對人臉搜尋的部份，由於一般在個人電腦上執行的方式並不適用於相機平台，因此我們開發適用於相機的演算法以及更精準的膚色模型，所以花費了相當多的時間，而關於辨識部分我們並未完成，對於進度的控制不良是我們的缺點，也是需要改進的地方。然而我們還是成功的在相機上實現人臉追蹤技術，此技術加上相機的很多優點，對於監視系統的改良是一大改進。我們感謝國科會與台大的贊助，也感謝台灣師大提供個人電腦與矽峰光電提供的相機使得計畫可以執行。

Reference:

- [1] D. J. Lan, Y. F. Ma, and H. J. Zhang, "A systemic framework of camera motion analysis for home video," in *Proc. Intl. Conf. Image Processing*, vol.1, pp.289-292, Sept. 2003
- [2] Nariman Habili, Cheng chew lim, Alireza Moini, "Segmentation of the face and hands in sign language video sequences using color and motion cues," *IEEE trans. on circuits and systems for video technology*, vol. 14, no. 8, pp.1080-1097, August 2004.
- [3] Kiran Challapali, Thomas Brodsky, Yun-Ting Lin, Yong Yan, Richard Yi Chen, "real-time object segmentation and coding for selective-quality video communications," *IEEE trans. on circuits and systems for video technology*, vol. 14, no. 6, pp. 813-824, June 2004.

晶片系統架構技術及開發平台研究之推動(3/3) -附屬計畫十

ARM-based SOC 平台以整合並加速影像擷取處理

ARM-based SOC platform to integrate and accelerate image capturing and processing

計畫編號：NSC 93-2215-E002-010 / 010

執行期限：93/8/1~94/7/31

計畫主持人：伍定武 講師 台灣師範大學工業教育系

計畫參與人員：李炘彥、曾文治

一、中文摘要

本研究設計一個含 37 個指令的 ARM7 CPU 與一個在同一實驗室中研發的 CMOS 影像擷取與影像處理單元(IPU)的 FPGA 設計[1]做結合，讓 ARM7 控制 IPU 的操作，IPU 實際上是將影像擷取(Image capturing)、迴旋(convolutions)與分類(sorting)的流程整合成一個 FPGA 電路，用三階段的管線(3-staged pipeline)的設計使三者幾乎可並行運作以增進速度，且影像 convolutions 與 sorting 會過濾 CMOS 影像感測器的 FPN (Fixed-Patten Noise) 與暗流(Dark current)雜訊而增進影像品質。ARM7 與 IPU 的 FPGA 連結可成為一個設計平台應用於 CPU 設計，控制介面設計，甚至 SoC 設計的教學。

Abstract

This research presents how an ARM7 was designed by repeated adding group- instructions and the system was verified in a self-developed FPGA board. The ARM7 was then connected to a CMOS image capturing and processing unit (IPU) implemented on other FPGA board. ARM7 now becomes a microcontroller for this IPU. IPU integrates image capturing, convolution and sorting in FPGA to perform 3-stage pipelined operations to speed up system operations. Convolution and sorting help further filter out the Fixed Patten Noise (FPN) and dark current noise in CMOS image sensor and result in better image qualities. The FPGA board with ARM7 and IPU could be used for teaching CPU design, controller design and a system-on chip (SoC) design since in FPGA version all circuits are in a single FPGA chip.

Keywords

ARM, CMOS Image Sensor, Image capturing, Image Processing, FPGA.

二、動機與目的

Peoples are usually familiar with using CPUs, such as 8051.80×86, ARM, MIPS. But very few of them think about designing them. This research presents a way by repeated adding more and more group-instructions to obtain a ARM7 CPU. So far an ARM7 CPU with 37 instructions was designed, simulated and the circuit was downloaded on a self-developed FPGA board for function verification. Arm7 was then connected with a CMOS image capturing and processing unit (IPU) implemented in the same lab on FPGA board [1]. This configuration let ARM7 be able to control the operations of IPU. Thus ARM7 becomes a microcontroller for IPU which sometimes is called image accelerator for its high speed operations. The connection of with ARM7 and IPU could be used for teaching CPU design, controller design and a system-on chip (SoC) design since all circuits are in a single FPGA chip.

三、研究方法與成果

3.1 ARM7 System Configuration

ARM7 system shown in Fig. 2-1 includes a register bank (REG) containing 16 32-bit registers, a main memory (MEM), a data processing unit (DTU), and control unit (CON) for organizing and sequencing the system operations [2,3]. Fig. 2-1 also shows the datapath between these 4 units for the execution of 28 DPU instructions (16 for alu, 6 for shifting, 6 for multiply), not including the 4 condition bits of NZCV (Negative, Zero, Carry, Overflow). If the condition bits are included (as many other computers do) the maxima number of instructions may reach 112 (28×4).

Fig. 2-2 shows the inside of the DPU in Fig. 2-1a, the DPU includes an ALU, a barrel shifter and a multiplier. The detailed circuit of ALU, shifter and multiplier can be self-designed or be obtained form existing libraries. Cond is a special 16 input multiplex

selected by NZCV. This DPU structure is obtained according to the following instruction formats

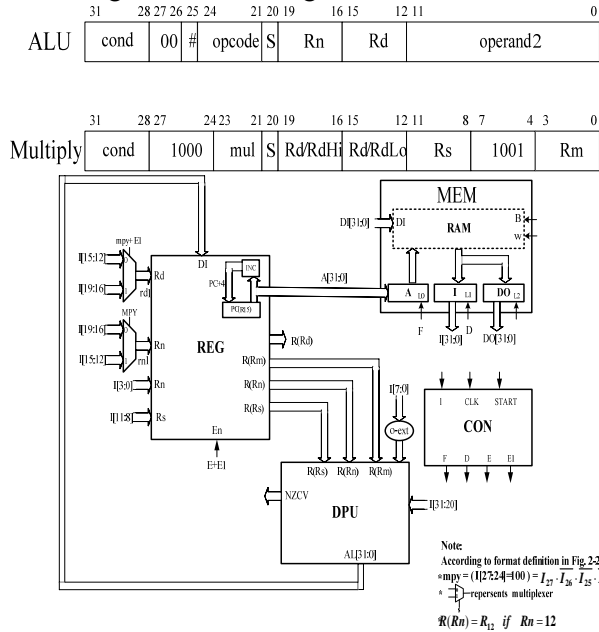


Fig. 2-1 ARM7 System Configuration for 28 DPU

instructions

Fig. 2-1 is a system that can execute 28 DPU instructions. Increasing system complexity results if two more groups of instruction, branch (2 instructions) and data transfer (7 instructions), are added. The total number of instructions are 37(=28+2+7). Next section briefly describes ARM7 with 37 instructions.

3.2 Added Branch and Data Transfer Instructions and Control Unit

3.2.1 Branch and Data Transfer Circuit

Fig. 3-1 shows the block diagram, datapath and control for the added branch and data-transfer instruction. The increasing complexity can be found in datapath as well as in control unit CON. rd2, m2 and en3 that input to REG are 4x1 multiplexer for 4-bit bus not shown for simplicity.

Branch instructions are used to change the program execution sequence, just by changing the values of program counter (R15) will accomplish it. Data transfer instructions are concerned with the single or multiple data transfer between registers and memory as shown in Figure 3-1. Two buses: a gray-dotted bus in the Figure DO [31:0] from memory to DI terminal of REG block and the other dotted bus R(Rd) from REG block to the DI[31:0] terminal of MEM were added, a shift control block, for identifying which registers were to be transferred, and other associate circuits added for data transfer operations were included in Figure 3-1.

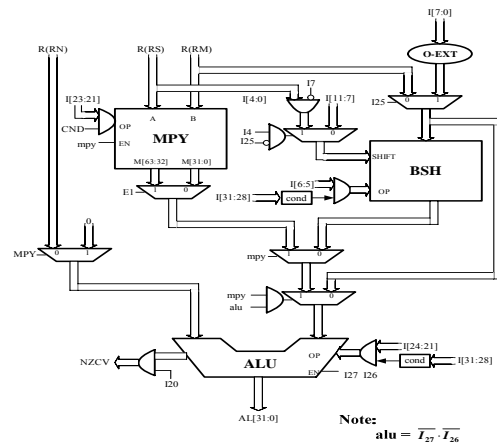


Fig. 2-2 DPU block diagram

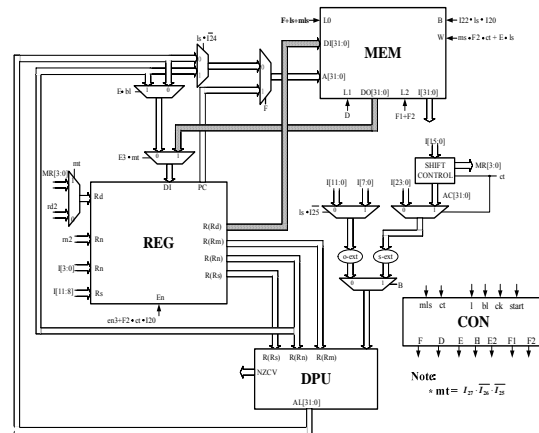


Fig.3-1 Block diagram of Arm7 contains branch, data-transfer and DPU instructions.

3.3 Control Unit

When more instructions were added, the control unit became larger and more complicated. The control unit (CON) at lower right corner in Fig. 3-1 has more inputs and outputs than that in Fig. 2-1. It actually is a state machine realization of the state diagram shown in Fig. 3-2. Most of the instructions in ARM use only 3 states during instruction execution, namely F(fetch), D(decode), E (execute). Long(l) multiplication(64 bit product) and Branch Link(bl) instruction, both needed extra E1 or E2 state, respectively. The data transfer included Load/Store (ls), multiple Load/ Store (mls) need extra states F1 and F2. So the inputs of the CON block are the decoded instruction signals l, bl, ls, mls as mentioned above, and ct represented the multiple register transfer counts, the outputs are the states F, F1, F2, D, E, E1, E2 as shown the enlarged block in Fig. 3-2 for easy reading.

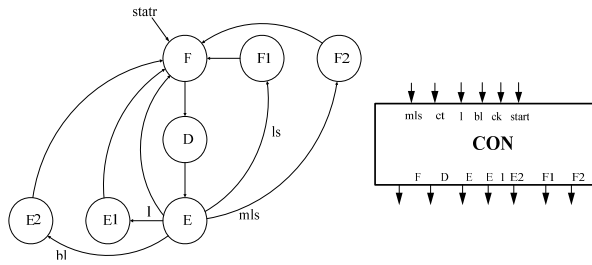


Fig. 3-2 Control state diagram and Enlarged Control unit block diagram from Figure 3-1.

3.4 Platform Board Demonstration

Fig. 3-3 Showed the self-develop FPGA board with an 0.8 million gate count XILINX FPGA chip in the center. Each of the 36 instructions in ARM7 was verified in the board. Several programs such as sorting, arithmetic calculations, counters were keyed in for verification. Fig. 3-4. Shows the board was running a 24-hour clock program which was keyed in through the key pad at lower right corner and the total 24-hour clock circuit was loaded through parallel port into FPGA chip. The 7-segment showing 030903 was the snap shot at the 3 o'clock 9 minutes and 3 seconds while 24-hour clock program was running.

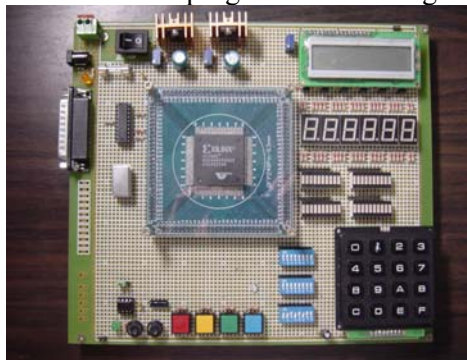


Fig. 3-3 Self-made platform board

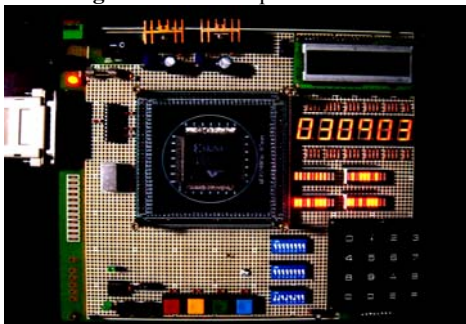


Fig.3-4 Self-made platform board snap shot at 03:09:03.during program running

Section 2 through 4 are briefly described the add-up approach to the design and verification of 36-instruction ARM7 CPU. This approach is working so far and can be continued. But in the following sections, the focus is in the application of the ARM7 CPU in image capturing and processing by interfacing a image processing unit.

3.5 Image Capturing and Processing

The circuit to perform image capturing and processing is called IPU (Image Processing Unit). It includes image capturing, 2-D convolution and 2-D sorting as shown in Fig. 5. Image capturing is an self-developed FPGA version of 8051 controller program. 2-D convolution and 2-D sorting are based on the FPGA version of image processing algorithms from Crookes[4,5] and Maheshwari[6], respectively.

Capturing, convolution and sorting were usually done in series. They are now integrated in a single FPGA chip (occupied about 70k gate-count) for 3 state pipelined operations, to increase operation speed as well as to help further filter Fixed Pattern Noise (FPN) and dark current noise in CMOS image sensor.

IPU needs an external RAM to store the processed image for being used in PC or LCD display. An MEM (Memory management Unit) is used to manage the external memory access. The connection between IPU, MMU and PC/LCD are also shown in Fig 3-5.

After IPU finishing the operation of initialization, “ic-go” control is enabled from MMU, to start the image capturing operations. As shown in Fig. 3-5, Signal “valid” is enabled from comparing xpos and ypos with the stored frame size, then “enable” signal is activated for sending the input image data to 2-D convolver if the image data are within the frame size, meanwhile image capturing is in busy state, which means capturing, convolution and sorting are in working condition.

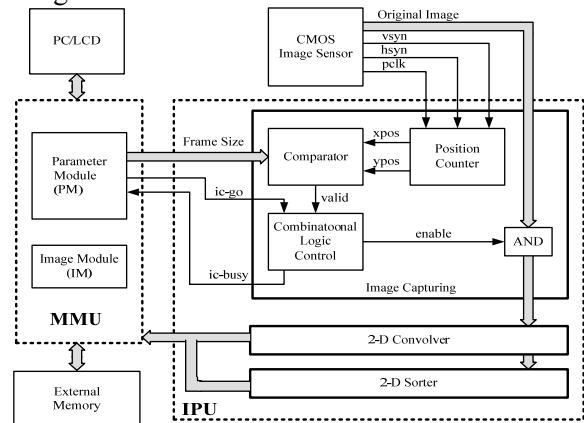


Fig.3-5 Internal and external connection of Image Processing Unit (IPU)

3.6 ARM7 as Microcontroller or SoC Based Design

The CMOS image capturing and filtering device can work independently, it can also be connected to the self-developed ARM7 microcontroller as shown in Fig. 3-6 Fig. 3-7shows the experiment result by using photo of Lena as an original image.

In Fig. 3-8 an ARM7 used up to 600k gate-count out of an 800k gate-count FPGA-2 chip, IPU used up to 70k gate-count out of a 300k gate-count FPGA-1 chip, IPU can be merged to the 800k chip as shown in Fig. 3-8 since both are using FPGA design. This architecture becomes a kind of SoC design in FPAG version.

四、結論

Since FPGA is reconfigurable the microcontroller CPU can be tailored to fit a special purpose. For example, if we have a 600k FPGA chip that can just hold the ARM7 of this design and no room for putting in 70 k gate-count image accelerator. Since all calculations are made in image accelerator circuits. Then it is possible to take out the 32x32 multiplier (occupied about 150k gate-count) which is currently unused and made room available for 70k gate-count image accelerator to be included in.

ARM7 embedded with many device controllers in a FPGA chip shown inside the dash-lined area in Fig. 3-8 demonstrates a kind of SoC architecture design. It might gradually become a platform for CPU design capable of adding controllers for controlling special accelerators or for interfacing the devices, such as image processing, mechantronic, nano-optics.

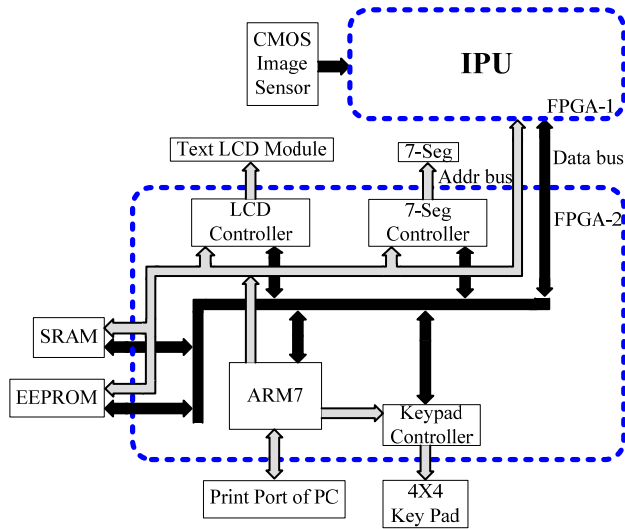
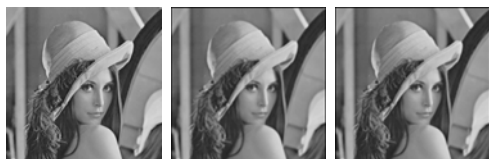
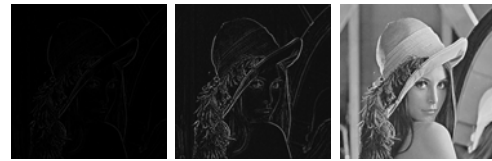


Fig. 3-6 ARM7 and IPU in sperate FPGA board



(a) Original Image (b) Averaging filter (c) Gaussian filter



(d) High pass filter (e) Enhance from d (f) a plus d
Fig. 3-7 Photo of Lena through this experiment

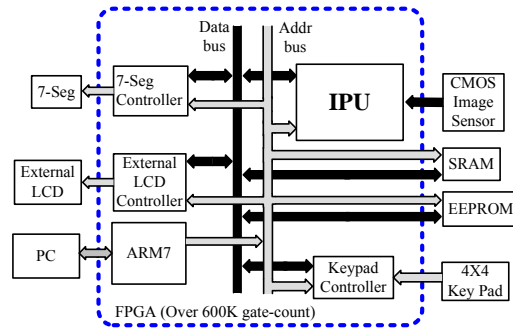


Fig. 3-8 ARM7 and IPU merge in a single FPGA chip

五、References

- [1] Chi-Jeng Chang, Zen-Yi Huang, Hsin-Yen Li, Kai-Ting Hu, and Wen-Chih Tseng, "Pipelined Operation of Image Capturing and Processing," Processing of 5th IEEE Conference on Nanotechnology, TU-PS5-1, July 11-15, 2005.
- [2] Steve Furber, "ARM System-on-Chip Architecture (2nd Edition)," Maidstone, Kent, USA: Addison-Wesley, 2000.
- [3] Seal, D. ARM Architectural Reference Manual, 2nd edition, Addison Wesley, 2000.
- [4] Crookes D., Benkrid K., Bouridane A., Alotaibi K., and Benkrid A., "Design and implementation of a high level programming environment for FPGA-based image processing," Vision, Image and Signal Processing, IEE Proceedings, vol. 147, Issue: 4, pp. 377-384, Aug. 2000.
- [5] Benkrid K., Crookes D., and Benkrid A., "Towards a general framework for FPGA based image processing using hardware skeleton," Parallel Computing vol. 28, Issue: 7-8, pp. 1141-1154, Aug, 2002.
- [6] Maheshwari R., Rao.S.P., and Poonacha P.G., "FPGA implementation of median filter," Tenth International Conference on VLSI Design, pp. 523-524, June. 1997.

肆、舉辦十五場 SoC 訓練課程

每週二下午二至五時在台灣大學電機二館 105 會議室舉行

1. 94.02.23: HW-SW Codesign Overview 主講者: 陳少傑教授
2. 94.03.02: HW-SW Co-design Models, Architectures, and Spec. Lang 主講者: 熊博安教授
3. 94.03.09: Function/Architecture Co-design 主講者: 李宗演教授
4. 94.03.16: High Level Synthesis of Loop Accelerators 主講者: 林永隆教授
5. 94.03.23: SLD for Reconfigurable System 主講者: 顧孟愷教授
6. 94.03.30: Design for Manufacturability in Sub-130nm Technologies 主講者: 張克正教授
7. 94.04.06: On-Chip Bus - The Backbone of SoC Integration 主講者: 黃俊達教授
8. 94.04.13: Architecture Synthesis: A SW/HW Partitioning Tool 主講者: 蘇培陞博士
9. 94.04.20: SoC Design for Ubiquitous Computing 主講者: 王駿發教授
10. 94.04.27: Baseband Design for MC-CDMA Mobile Communication 主講者: 關志達教授
11. 94.05.04: UWB Baseband Design 主講者: 李鎮宜教授
12. 94.05.11: Low-Power High-Speed Content Addressable Memories 主講者: 王進賢教授
13. 94.05.18: Sensor Network and its Applications 主講者: 黃寶儀教授
14. 94.05.25: Electronic System Level Verification Methodology 主講者: 陳哲生博士
15. 94.06.01: SoC Hardware-Software Codesign 主講者: 蘇泓萌博士

伍、舉辦五場 SoC 人才培訓海外講座

1. 93.8.5-93.8.13: 北區 SOC 人才培育推動小組邀請美國 University of California, Santa Barbara 之 Prof. Margaret Marek-Sadowska 來台。分別於 8 月 6 日在台灣大學及 8 月 9 日在成功大學演講 “Placement, White Space, and Interconnect Density”。
2. 93.10.21-93.10.22: 北區 SOC 人才培育推動小組分別於九十三年 10 月 19 日至 10 月 20 日(星期二、三) 在國立成功大學及於九十三年 10 月 21 日至 10 月 22 日(星期四、五)在國立台灣大學博理館 101 演講廳舉辦了兩場各為期兩天之 SoC 人才培訓講座: “System-Level Modeling for System-on-a-Chip Design”。講者為美國 IBM TJ Watson Research Center 之 Dr. Reinaldo Bergamaschi, 美國 Synopsys 之 Dr. Yirng-An Chen, 法國 INPG Grenoble, France 之 Dr. Ahmed Jerraya, 及德國 University of Tübingen 之 Prof. Wolfgang Rosenstiel。
3. 94.1.31: 北區 SOC 人才培育推動小組及台大電子所於九十四年 1 月 31 日(星期一)在國立台灣大學博理館 101 演講廳舉辦了一場 “前瞻多媒體通訊處理器設計研討會”, 講者為德國 Aachen University of Technology 及 CoWare Inc. 之 Dr. Heinrich Meyr, 講題為 “Application-specific Processors (ASIP) for Multimedia and Wireless Communications: Design, Cost, and Energy - Efficiency vs. Flexibility”。