

國立交通大學

電機學院 電機與控制學程

碩士論文

嵌入式影像處理系統及 SURF 特徵擷取

Embedded Image Processing System and  
SURF Feature Extraction

研究生：徐尉嘉

指導教授：宋開泰 博士

中華民國一百零一年八月

# 嵌入式影像處理系統及 SURF 特徵擷取

## Embedded Image Processing System and SURF Feature Extraction

研 究 生：徐尉嘉

Student：Wei-Jia Hsu

指導教授：宋開泰 博士

Advisor：Dr. Kai-Tai Song



A Thesis

Submitted to College of Electrical and Computer Engineering

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master of Science

in

Electrical and Control Engineering

July 2012

Hsinchu, Taiwan, Republic of China

中 華 民 國 一 百 零 一 年 八 月

# 嵌入式影像處理系統及 SURF 特徵擷取

學生：徐尉嘉

指導教授：宋開泰 博士

國立交通大學 電機學院 電機與控制學程 碩士班

## 摘要

本論文主要目的在研製一個嵌入式影像處理卡，未來能將其應用在機器人上進行特徵擷取及各種的影像處理演算法。本影像處理卡之電路設計可以分成數位訊號處理器(DSP)及可程式邏輯閘陣列(FPGA)兩個模組。影像擷取採用低功耗及小尺寸的 CMOS 影像感測器，其輸出格式設定為 YUV4:2:2，擷取的影像尺寸則是 512x480 像素。DSP 採用德州儀器公司的 TMS320C6414TGLZ7，主要用於實現影像處理演算法；FPGA 採用 Altera 公司的 EP2C35F672C6，用於比較低階的影像處理及 CMOS 影像感測器控制。本論文成功地於所研製的影像處理卡中實現加速強健特徵(SURF)演算法，透過 USB 介面將所擷取的影像畫面及 SURF 描述子傳送到個人電腦顯示及進行特徵匹配。實驗結果顯示，即時影像擷取週期大約 33.5ms；當 SURF 描述子數量等於 64 個，計算時間大約 417.5ms；當 SURF 描述子數量為 199 個，計算時間大約是 985ms；依樣本模型在特徵擷取時設定的門檻值不同，特徵匹配率則由 70.2%變化到 20%。

# **Embedded Image Processing System and SURF Feature Extraction**

Student : Wei-Jia Hsu

Advisor : Dr. Kai-Tai Song

Degree Program of Electrical and Computer Engineering  
National Chiao Tung University

## **ABSTRACT**

This thesis aims to develop an embedded image processing board, which can be used to realize various image feature extraction and processing algorithms for robotic applications. The circuitry of this image board can be divided into two parts, one is for a digital signal processor (DSP) and the other for a field programmable gate array (FPGA). A low power, small size CMOS image sensor with YUV4:2:2 image format and 512x480 pixel size is adopted for image acquisition. The DSP module is mainly used for implementing image processing algorithms, while the FPGA module for lower level image processing and sensor control functions. In this work, the TMS320C6414TGLZ7 DSP from Texas Instruments Corporation and the EP2C35F672C6 FPGA from Altera Corporation were selected for the image board. In this work, speeded-up robust features (SURF) algorithm has been successfully realized on the developed image board. An USB interface with average throughput of 22MB/sec works to transfer acquired image frames and extracted SURF descriptors to a personal computer, where feature matching is executed. Experimental results show that the refresh period of frame grabbing is about 33.5ms. When the number of SURF descriptor is equal to 64, the computing time is about 417.5ms; when the number of SURF descriptor is equal to 199, the computing time is about 985ms. The typical matching rate varies from 70.2% to 20%, depending on the threshold of the patterns used in feature extraction.

## 誌 謝

首先，由衷的感謝指導教授 宋開泰博士對我的督促及耐心指導，讓我最終能順利完成論文。

另外，感謝專科時期的專題指導老師 黃寶強老師一直鼓勵我繼續攻讀研究所，如今能順利完成學業，總算沒辜負他的期望。

最後，感謝從學生時代就一路陪我走來的妻子楊婉琇。因為她的支持與辛苦地照顧家庭，讓我可以專心完成學業。



# 目 錄

中文摘要 .....	i
ABSTRACT.....	ii
誌 謝.....	iii
目 錄 .....	iv
圖目錄 .....	vii
表目錄 .....	x

## 第一章 緒論..... 1

1.1 研究動機.....	2
1.2 文獻回顧.....	3
1.2.1 低價的嵌入式彩色視覺系統[2].....	3
1.2.2 以FPGA實作適用於嵌入式視覺系統的影像處理架構[3].....	3
1.2.3 用於移動式機器人的即時手勢辨識系統[4].....	4
1.2.4 使用FPGA與CMOS影像感測器的影像系統[5].....	6
1.2.5 用於微型自主移動式機器人的即時影像辨識系統[6].....	7
1.2.6 基於DSP的嵌入式CMOS影像處理系統[7].....	7
1.3 問題描述與論文架構.....	8

## 第二章 嵌入式影像處理系統硬體電路設計..... 9

2.1 系統方塊圖.....	9
2.2 FPGA模組.....	11
2.2.1 FPGA配置方式.....	11
2.2.2 CMOS影像感測器介面.....	12
2.2.3 FPGA之記憶體介面.....	14
2.2.4 晶體震盪器(Oscillator).....	15
2.2.5 I/O擴充用介面.....	16
2.3 數位訊號處理器(DSP)模組 .....	16
2.3.1 外部記憶體介面(External Memory Interface, EMIF).....	16
2.3.2 記憶體映射圖(Memory Map)及介面訊號.....	17
2.3.3 內部周邊配置.....	19

2.3.4 開機模式.....	19
a) 無開機模式.....	19
b) ROM開機模式.....	20
2.3.5 外部裝置的通訊介面.....	21
a) 高速USB 介面.....	21
b) RS232介面.....	22
2.3.6 GPIO與中斷訊號.....	23
2.4 電源模組.....	23
2.5 設計總結.....	24
<b>第三章 SURF 演算法簡介.....</b>	<b>25</b>
3.1 積分影像.....	25
3.2 Fast-Hessian 偵測器.....	26
3.2.1 近似的 Hessian 行列式.....	26
3.2.2 尺度空間表示.....	28
3.2.3 興趣點定位.....	29
3.3 興趣點描述子.....	30
3.3.1 可重現的主方位.....	30
3.3.2 基於 Haar 小波響應和的描述子.....	31
3.4 快速匹配用的旗標.....	31
3.5 本章總結.....	32
<b>第四章 嵌入式影像處理系統之應用.....</b>	<b>33</b>
4.1 SURF 演算法實作系統簡介.....	33
4.1.1 FPGA 內部架構.....	33
4.1.2 DSP 程式架構.....	35
4.1.3 人機介面軟體.....	36
4.2 SURF 演算法相關參數定義.....	36
4.2.1 以 Octave 為基準的相關參數值.....	37
4.2.2 以箱形濾波器大小為基準的相關參數值.....	37
4.3 Avalon 介面.....	39
4.4 不同時脈領域之間的資料傳遞.....	41

4.5 積分影像功能實作.....	43
4.6 如何建立 DSP 的 Boot ROM.....	45
4.6.1 C6414T 的 ROM 開機程序.....	45
4.6.2 產生燒錄用的 Hex 檔.....	45
4.6.3 ROM 的燒錄程式.....	47
a) Hex 檔解析.....	47
b) 燒錄函式.....	48
4.7 PC 與 DSP 的通訊命令說明.....	49
4.8 PC 端人機介面的顯示架構.....	51
<b>第五章 實驗與討論.....</b>	<b>53</b>
5.1 即時影像擷取.....	54
5.2 動態 SURF 特徵擷取實驗.....	55
5.3 動態 SURF 特徵匹配實驗.....	62
5.4 OpenSURF 與本系統計算之 SURF 特徵比較.....	67
<b>第六章 結論與未來工作.....</b>	<b>71</b>
6.1 結論.....	71
6.2 未來工作.....	72
<b>參考文獻.....</b>	<b>74</b>
<b>附錄一、嵌入式影像處理系統電路圖.....</b>	<b>76</b>
<b>附錄二、CMOS 影像感測器(OV7649)電路圖.....</b>	<b>92</b>



## 圖 目 錄

圖 1-1 低價的嵌入式彩色視覺系統方塊圖.....	4
圖 1-2 分層式的視覺系統[3].....	4
圖 1-3 影像前處理器[3].....	4
圖 1-4 影像處理單元[3].....	5
圖 1-5 視覺系統方塊圖[4].....	5
圖 1-6 智慧型影像感測器架構[5].....	6
圖 1-7 智慧型影像感測器的方塊圖[5].....	6
圖 1-8 微型足球機器人的即時影像辨識系統組成[6].....	7
圖 1-9 基於 DSP 的嵌入式 CMOS 影像處理系統[7].....	8
圖 2-1 嵌入式影像處理系統方塊圖.....	9
圖 2-2 嵌入式影像處理系統實體圖.....	10
圖 2-3 FPGA 配置電路(AS、JTAG 模式).....	12
圖 2-4 影像感測器 OV7649 硬體重置訊號.....	13
圖 2-5 OV7649 每列的輸出時序圖(細部時間參數詳見[14]).....	13
圖 2-6 OV7649 每個影像畫面的輸出時序圖.....	13
圖 2-7 EMIF 輸出時脈的路徑選擇[12].....	17
圖 2-8 EMIF-B 與同步式 Slave FIFO 介面波形.....	22
圖 2-9 McBSP 使用串列埠模式連接 RS232 的方式.....	22
圖 2-10 McBSP 傳送 N, 8, 1 模式的 RS232 位元組的波形[24].....	23
圖 3-1 積分影像.....	25
圖 3-2 離散化並且裁剪後的高斯二階偏導數[9] (a)x 方向 (b)y 方向 (c)xy 方向 .....	27
圖 3-3 近似的高斯二階偏導數濾波器及權重值[9] (a)x 方向 (b)y 方向 (c)xy 方向 .....	27
圖 3-4 尺度空間示意圖。(a)傳統의影像金字塔 (b)SURF 的方法[1].....	28
圖 3-5 濾波器邊長[1].....	29
圖 3-6 3x3x3 非最大抑制[9].....	29
圖 3-7 Haar 小波濾波器[9]。(a)(b)分別用於計算 x、y 方向的響應值 .....	30
圖 3-8 主方位的搜尋。(a)至(c)表示以 $\pi/3$ 的扇形區域繞著興趣點旋轉一周 (b)表 示此興趣點的主方位.....	30
圖 3-9 以興趣點為中心之 SURF 描述子計算範圍.....	31

圖 4-1 SURF 演算法實作之系統方塊圖.....	33
圖 4-2 FPGA 內部架構.....	34
圖 4-3 DSP 程式之 SURF 描述子計算流程.....	35
圖 4-4 人機介面軟體畫面.....	36
圖 4-5 不同尺度下興趣點數量的直方圖[1].....	37
圖 4-6 箱形濾波器的取樣點。(a)x 方向 (b)xy 方向 (c)y 方向.....	38
圖 4-7 使用 Qsys 程式規劃 SOPC 模組.....	40
圖 4-8 配合 waitrequest 訊號的 Avalon-MM 讀寫時序[25].....	41
圖 4-9 亞穩態的圖解[26]。(a)訊號轉換發生在時脈之後，狀態不變 (b)訊號轉換 符合 tSU 與 tH，改變至新狀態 (c)訊號違反 tSU 或 tH，輸出處於亞穩態..	42
圖 4-10 簡單型的同步用暫存器鏈[26].....	42
圖 4-11 雙時脈 FIFO 讀/寫 CMOS 影像感測器輸出值的模擬波形.....	43
圖 4-12 即時計算影像感測器輸出點(a,b)的積分影像示意圖.....	44
圖 4-13 FPGA 計算積分影像的電路結構.....	44
圖 4-14 C6414T 的 ROM 開機程序.....	46
圖 4-15 產生燒錄用的 Hex 檔所需之相關設定與檔案.....	46
圖 4-16 將 out 檔轉換成 Hex 燒錄檔所需的 cmd 檔內容.....	47
圖 4-17 設定編譯後自動產生 Hex 燒錄檔.....	48
圖 4-18 ASCII-Hex 格式.....	48
圖 4-19 EP2IN 及 EP6IN 傳送 16 位元資料寬的位元組順序.....	50
圖 4-20 人機介面軟體的顯示架構.....	52
圖 5-1 本次的實驗環境.....	53
圖 5-2 CMOS 影像感測器的 VSYNC 與 HREF 輸出波形.....	54
圖 5-3 嵌入式影像處理系統即時擷取之畫面 (a)小乳牛 (b)混合物件 (c)書本...	55
圖 5-4 特徵擷取時軟體的操作流程圖.....	56
圖 5-5 不同門檻值之下小乳牛(亮背景)的 SURF 特徵描述子 (a)門檻值=50 (b)門 檻值=200 (c)門檻值=350 (d)門檻值=500.....	57
圖 5-6 不同門檻值之下小乳牛(暗背景)的 SURF 特徵描述子 (a)門檻值=50 (b)門 檻值=200 (c)門檻值=350 (d)門檻值=500.....	58
圖 5-7 不同門檻值之下混合物件的 SURF 特徵描述子 (a)門檻值=50 (b)門檻值= 200 (c)門檻值=350 (d)門檻值=500.....	58

圖 5-8 不同門檻值之下書本的 SURF 特徵描述子 (a)門檻值=50 (b)門檻值=200 (c)門檻值=350 (d)門檻值=500.....	59
圖 5-9 個別的折線圖表現不同場景的 SURF 特徵數量與計算時間 (a)小乳牛(亮 背景) (b)小乳牛(暗背景) (c)混合物件 (d)書本.....	61
圖 5-10 同一張折線圖表現不同場景的 SURF 特徵數量與計算時間.....	61
圖 5-11 在 PC 端進行特徵匹配時的軟體操作流程圖.....	62
圖 5-12 小乳牛的特徵匹配。右側均為門檻值=300 的樣本模型，左側為動態擷取 之特徵 (a)門檻值=50 (b)門檻值=200 (a)門檻值=350 (a)門檻值=500....	64
圖 5-13 混合物件的特徵匹配。右側均為門檻值=300 的樣本模型，左側為動態擷 取之特徵 (a)門檻值=50 (b)門檻值=200 (a)門檻值=350 (a)門檻值=500...	65
圖 5-14 書本的特徵匹配。右側均為門檻值=500 的樣本模型，左側為動態擷取之 特徵 (a)門檻值=50 (b)門檻值=200 (a)門檻值=350 (a)門檻值=500.....	67
圖 5-15 不同門檻值之下 SURF 特徵與匹配數量 (a)小乳牛 (b)混合物件 (c)書本...	68
圖 5-16 不同 SURF 實作法擷取簡單幾何圖形之特徵 (a)本系統，門檻值=100，特 徵數=226 (b)OpenSURF，門檻值=0.00015，特徵數=155.....	69
圖 5-17 不同 SURF 實作法擷取方形物件之特徵 (a)本系統，門檻值=100，特徵數 =182 (b)OpenSURF，門檻值=0.00004，特徵數=145.....	70



## 表 目 錄

表 1-1 視覺系統各區塊的效能[4].....	6
表 2-1 Cyclone II 的四種配置方式.....	12
表 2-2 連接於 FPGA 之 SSRAM 訊號說明.....	14
表 2-3 連接於 FPGA 之 SDRAM 訊號說明.....	15
表 2-4 連接於 FPGA 之 FLASH ROM 訊號說明.....	15
表 2-5 DSP 的 EMIF 介面特性摘要.....	16
表 2-6 本系統的 DSP 記憶體映射圖.....	17
表 2-7 連接於 DSP EMIF-A 之 SDRAM 訊號說明.....	18
表 2-8 連接於 DSP EMIF-B 之 FLASH ROM 訊號說明.....	18
表 2-9 連接於 DSP EMIF-B 之 Slave FIFO 訊號說明.....	19
表 2-10 C6414T 內部周邊配置說明.....	20
表 4-1 以 Octave 為基準的相關參數值.....	38
表 4-2 以箱形濾波器大小為基準的相關參數值.....	39
表 4-3 使用非同步憶記體時邏輯位址與 EMIF 位址的對應關係[28].....	49
表 4-4 FLASH ROM(Am29LV320DB)燒錄命令摘要.....	49
表 4-5 PC 送到 DSP 的命令封包內容.....	50
表 4-6 DSP 回應 PC 的封包內容.....	51
表 5-1 不同門檻值之下的 SURF 特徵數量及計算時間.....	60
表 5-2 不同門檻值之下的 SURF 特徵與匹配數量及計算時間.....	68
表 6-1 FPGA EP2C35F672 與 EP2C70F672 內部資源比較.....	73

## 第一章 緒論

基於個人電腦的影像處理系統大致上由四個部分組成：CCD 或 CMOS 影像感測器、影像擷取卡、個人電腦及相關的演算法。雖然隨著科技的進步，個人電腦系統的效能不斷提升，但是如果執行即時的影像處理演算法卻可能佔用太多電腦資源，使其它軟體功能效率降低。同時若考慮將影像處理功能整合到機器人平台上，則個人電腦還會出現一些不適用的缺點，例如：體積龐大、價格昂貴等。

一套適用於機器人的嵌入式影像處理系統至少要擁有以下特點：

- 價格合理，體積小，能完全獨立運作。
- 能夠連接影像感測裝置。
- 具有與其它控制裝置通訊的能力。
- 內含足夠的記憶體以暫存即時影像及運算過程的資料。
- 方便的開發環境(程式撰寫、除錯、燒錄等)。
- 運算效能高，能符合機器人的即時性需求。

根據以上原則，本論文設計出一套基於數位訊號處理器(Digital Signal Processor, DSP)及可程式邏輯閘陣列(Field Programmable Gate Array, FPGA)的嵌入式影像處理系統，並且在系統上實作快速強健特徵(Speeded-up Robust Features, SURF) [1]演算法，以驗證此系統設計的正确性與實用性，最終希望能將此系統裝載於機器人上進行影像特徵擷取及其它影像處理演算法。

近年來由於 FPGA 製程的進步，使得其內部的邏輯單元增加，效能也不斷提升，然而價格卻沒有因此而增加甚至更低，加上能快速地建立原型設計，因此許多大型或複雜的數位電路都會採用 FPGA 來做設計。在即時影像處理的應用中，當演算法需要大量規律性的計算時，例如：利用  $N \times N$  的遮罩(mask)對影像進行空間濾波，就可以利用 FPGA 內部的移位暫存器搭配硬體乘法器及加法器，使遮罩內的迴旋積分計算與影像感測器的輸出頻率同步。

德州儀器公司的數位訊號處理器可依用途區分出多種類別，例如：適用於馬達控制的 C2000 系列以及高效能的 C6000 系列等。此次設計採用的即是 C6000 系列中的定點式 DSP，操作頻率為 720MHz，此系列的 DSP 可適用在影像處理、專業的混音器或超音波系統等。另外德州儀器公司提供豐富的開發工具與程式庫，例如：DSP/BIOS 即時作業系統、整合式開發環境 CCStudio 及晶片支援程式

庫 CSL 等，可以輔助使用者進行影像處理演算法的開發。

CMOS 影像感測器的技術近年也是進步快速，它與 CCD 相比具有體積小、價格低、耗電少及可選擇要輸出的範圍等特色。尤其是它不需要額外的介面電路就可以和 FPGA 直接連接，所以做為本次實作 SURF 演算法時的影像輸入來源。

## 1.1 研究動機

嵌入式影像處理系統過去已經有相當多的研究發表，使用的硬體有微處理器、FPGA 或 DSP 等。在 Rowe 等人所設計的系統中[2]，使用微控制器進行簡單的彩色影像辨識，畫面更新率(Frame Rate)雖然可以達到即時的程度，但由於程式與資料記憶體不足而無法執行進一步的影像處理功能。McBader and Lee 的 FPGA 系統[3]已完成一些影像處理功能，但 FPGA 要執行動作仍需要一個外部的控制器做設定，原作者表示會將 RISC 處理器軟核嵌入 FPGA 中，使原系統更加完整。

過去的系統比較常見到使用單一主要元件進行設計，使得一項影像處理工作可能要搭配其它模組才可達成，如[3]的系統預計未來會在現有的 FPGA 加入一個 RISC 處理器的軟核。但是，基於微處理器的開發系統除了硬體裝置之外，重要的是軟體開發環境是否完善與便利，才能讓設計者專注於演算法的開發。因此本次設計的嵌入式影像處理系統採用 DSP 與 FPGA 的組合，希望藉由 DSP 能即時處理數位訊號、豐富的軟體開發資源及 FPGA 內部電路可彈性配置等特性，讓使用者擁有一個高效能的影像處理演算法設計平台。另外，由於將 DSP 與 FPGA 兩者整合於一片電路板中，使得體積縮小、重量減輕並且具有足夠的運算能力，相對於原本實驗室使用的工業電腦或不同開發板組成的系統，本系統更適合裝載於機器人平台上進行影像處理工作。例如：在移動式機械臂的影像伺服控制中，負責擷取目標物件的特徵點座標；或者在家用機器人上進行人臉表情辨識，使得人與機器的互動更人性化。

FPGA 的電路內容及影像演算法開發的過程中常需要不斷修改，因此所設計的系統也要能方便地更新程式。此系統的 DSP 與 FPGA 都能使用 JTAG 進行程式的下載。其中現有的 DSP 燒錄程式通常都要配合廠商特定的開發板，因此本次開發的 DSP 程式中將內含 FLASH ROM 的燒錄函式，在系統初次使用時仍必須先以 JTAG 將開發完成的程式碼載入 DSP，再用 PC 端的人機介面軟體將 DSP 的程式碼傳送到系統中由 DSP 燒入 FLASH ROM，如此才能使 DSP 脫離開發環境而獨立運作。以後要更新程式時，就可以直接由 PC 端的人機介面軟體進行燒錄。

## 1.2 文獻回顧

數位影像處理涵蓋的範圍廣泛，例如：相片修補、影像定位、物件辨識等，本論文僅著重於嵌入式影像處理系統的設計，本節將說明一些相似的系統設計。

### 1.2.1 低價的嵌入式彩色視覺系統[2]

圖 1-1 是一個嵌入式彩色視覺系統[2]方塊圖，以 75 MIPS 的 RISC 微控制器為主，連接 CMOS 影像感測器的輸出匯流排與同步訊號，然後利用 RS232 介面將處理後的影像資訊傳送給外部的處理器，外部處理器可用此資訊來控制其它裝置，例如：自走車。此系統的 CMOS 影像感測器輸出畫面最大為 352x288 個像素，每秒最多可輸出 60 個畫面。由於使用高效能的 RISC 微控制器，每秒可以完成 16.7 個畫面(80x143 像素)的基本彩色斑點追蹤(Color Blob Tracking)。除此之外，藉由電路板上的連接器可以將多組系統同時連接在同一個影像感測器的匯流排，因此可以對同一個影像畫面平行地進行不同的操作。此系統的軟體目前可執行以下功能：彩色斑點追蹤、色彩統計、雜訊濾波、影像感測器設定與展示功能(例如：控制標準的伺服機)。

### 1.2.2 以 FPGA 實作適用於嵌入式視覺系統的影像處理架構[3]

作者利用 FPGA 實作一個適用於智慧型嵌入式視覺系統的可程式平行架構。在影像大小為 256x256 像素之下，進行影像的中值濾波(Median Filtering)時可達到每秒 62.42 個畫面。文中將傳統視覺系統分成如圖 1-2 所示的七層，此論文主要專注於圖 1-2 方框裏面三層的功能實現，圖 1-3 則是此方框內部的細部圖解，作者將其稱為影像前處理器(Image Pre-Processor)。圖中 Original Buffers 暫存影像感測器的輸出值，Temp. Buffers 則是暫存前次處理的結果，DMA Channel 負責將二者其中之一的內容分配到 Image Processing Array 進行影像前置處理，結果則放到主控制器可以存取的 Image Memory；主控制器將待處理的指令寫入 Program Memory 後，監視 Main Controller 回應的狀態決定是否完成，最後由 Image Memory 讀回處理結果。圖 1-3 裏面的 Image Processing Array 是此論文提出的平行處理陣列，由 16 個影像處理單元(Image Processing Element, IPE)組成，如圖 1-4 所示。IPE 可視為執行某些特定影像處理演算法的小型 DSP，它接收由 Main Controller 送來的指令，可處理兩個 16 位元的補數資料，最後輸出一個 32 位元的結果。

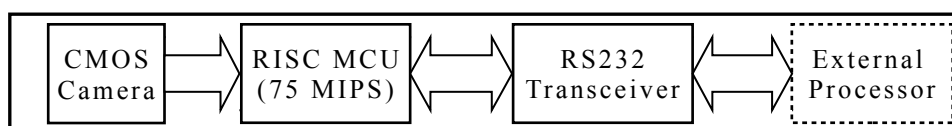


圖 1-1 低價的嵌入式彩色視覺系統方塊圖

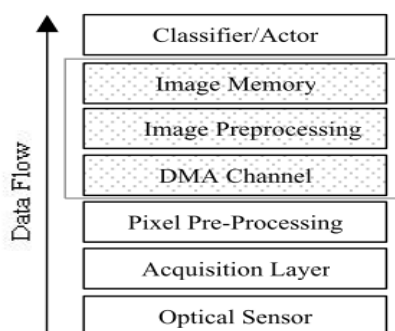


圖 1-2 分層式的視覺系統[3]

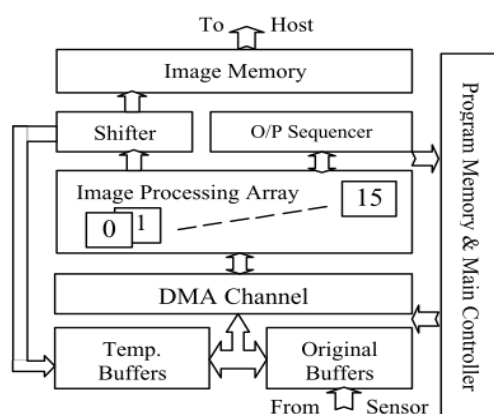


圖 1-3 影像前處理器[3]

### 1.2.3 用於移動式機器人的即時手勢辨識系統[4]

此系統藉由彩色 CMOS 影像感測器擷取手勢的影像，經過一連串의影像前置處理：像素讀取、RGB 轉 HSI、分割、濾波、壓縮等過程，然後將處理過的手勢影像置於畫面中央以增加後續神經網路判別的效率。系統架構如圖 1-5 所示，每個前置處理功能均為獨立的區塊，並且可平行地運作，整個手勢判別系統均包含於一顆 FPGA 內。圖 1-5 中各區塊的功能如下：

- **Pixel Read**：負責即時地讀取 CMOS 影像感測器輸出的 RGB 值。
- **RGB2HSI**：將 RGB 值轉換成各別獨立的 Hue(色相)、Saturation (飽和度)、Intensity(強度)三個分量。Intensity 使影像對環境中的光線變化具有較低的敏感度；另外 Hue 的原始數學定義含有反三角函數，使得運算困難而不利於硬體的實作，故此篇論文也介紹一個適合用硬體實作的演算法。

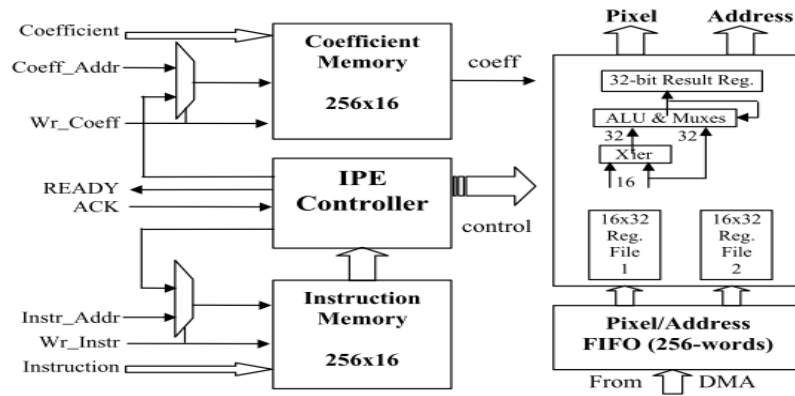


圖 1-4 影像處理單元[3]

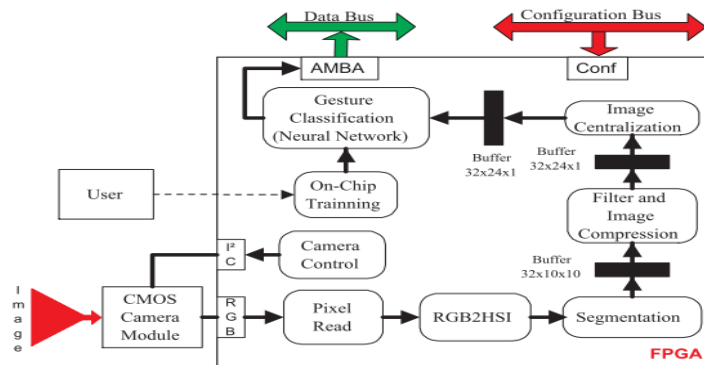


圖 1-5 視覺系統方塊圖[4]

- **Segmentation**：將 HSI 的影像轉換成二值化的影像。文中實作兩種方法，其一是以 H 分量為主的轉換，另一個則是將 HSI 分量各別轉換後再執行 AND 運算，最後採用效果較佳的 H 分量單獨轉換法。
- **Filter and Image Compression**：執行空間領域的濾波，並將原始 320x240 的影像壓縮成 32x24，用來減少輸入至神經網路的數量。
- **Image Centralization**：將手勢區域平移至影像的中心，目的是增加後續神經網路判別的效率。
- **Gestures Recognition**：利用 RAM-Based 之神經網路執行手勢辨識。此種神經網路屬於無權重式(weightless)，只能應用於二值化資料，故適合硬體實現。目前此系統可判別 7 種手勢，但可藉由系統內的訓練模組增加判別的手勢。

表 1-1 是使用 EP1S10F780C6 FPGA 時，系統中各區塊的效能。其中，OF:操作頻率，IS:影像大小(有標示\*\*者表示神經元的數量)，FS:每秒的畫面數量。由表中可看到由於 RGB2HSI 需使用到 320x240 次的除法，故此區塊的 FS 最低，而整個系統的 FS 則受限於此部分。

#### 1.2.4 使用 FPGA 與 CMOS 影像感測器的影像系統[5]

作者使用 System-On-Chip 方法設計一個智慧型影像感測器，用來執行除了高階決策之外的其它影像運算(early-vision)。基本架構如圖 1-6，包含主板 (STRATIX FPGA 模組)、CMOS 影像感測裝置、高速 USB2.0 通訊板。系統功能方塊圖如圖 1-7 所示。

表1-1 視覺系統各區塊的效能[4]

Blocks	OF (MHz)	IS	FS
Camera Control	269.11	-	-
Pixel Read	422.12	320x240	1374
RGB2HSI	16.88	320x240	31
Segmentation	109.79	320x240	1429
Filter and Compression	48.89	320x240	212
Image Centralization	18.50	32x24	3437
Neural Network	73.48	96**	382708

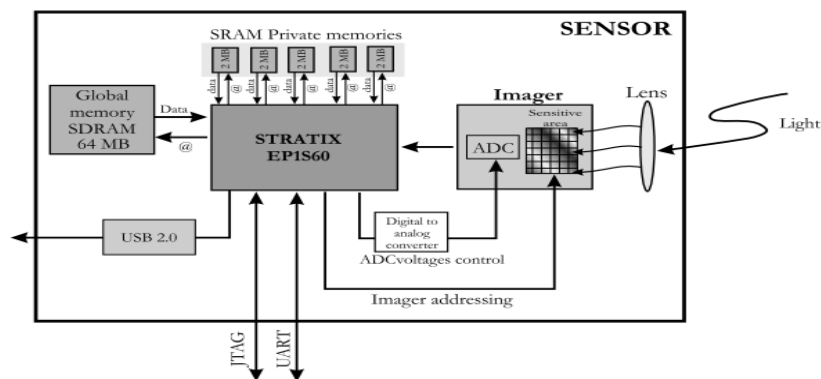


圖 1-6 智慧型影像感測器架構[5]

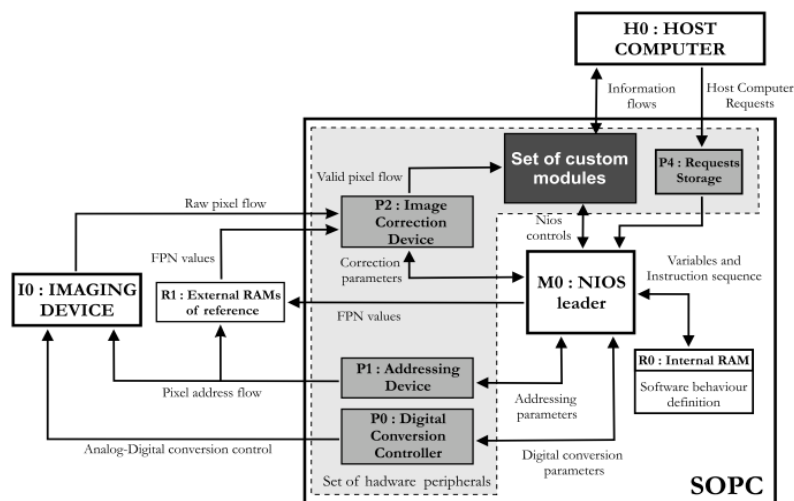


圖 1-7 智慧型影像感測器的方塊圖[5]

由於FPGA內部採用Altera公司的NIOS微處理器軟核，因此可以彈性地進行多種不同的影像運算，同時因為整個主要功能都包含在FPGA內部，使用者可以自訂不同的硬體周邊後加入原本的系統之中。根據作者所言，只要進一步的開發軟體工具，將有助於進行更複雜的視覺運算。

### 1.2.5 用於微型自主移動式機器人的即時影像辨識系統[6]

此論文使用Analog Devices公司的數位訊號處理器Blackfin BF533為運算平台，搭配CMOS影像感測器後裝載於微型足球機器人進行即時的影像辨識工作，系統元件如圖1-8所示。作者藉由適當安排DMA存取資料的順序，並使用自行設計的最佳化物件辨識演算法，使系統於預設的條件之下，在畫面大小為320x240像素時，可以用60畫面/秒的速率對簡單的球形物體進行辨識。作者認為目前因為使用固定式攝影機，使得機器人的視角受限於55度之內，所以計劃發展轉動式攝影機或使用微型立體攝影機來擷取三維物體的資訊。

### 1.2.6 基於DSP的嵌入式CMOS影像處理系統[7]

此系統由德州儀器公司的數位訊號處理器TMS320DM642，搭配外部記憶體、通訊介面(高速USB裝置與網路)及CMOS影像感測器所組成，系統元件如圖1-9所示。此系統擷取VGA影像的速率可達100畫面/秒。系統軟體以德州儀器公司的DSP/BIOS即時作業系統為基礎，實作了網路通訊、影像擷取、影像前處理與JPEG壓縮等功能。根據作者描述，目前在夜晚時的影像擷取效果不佳，往後除了改善此缺點之外，將專注於實現其它中階的影像處理功能並且改善目前的影像前處理運算。

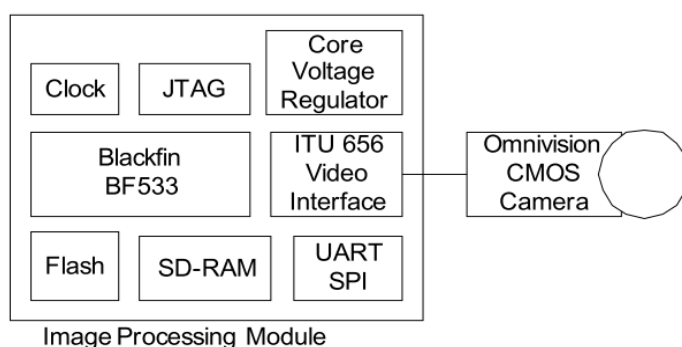


圖 1-8 微型足球機器人的即時影像辨識系統組成[6]

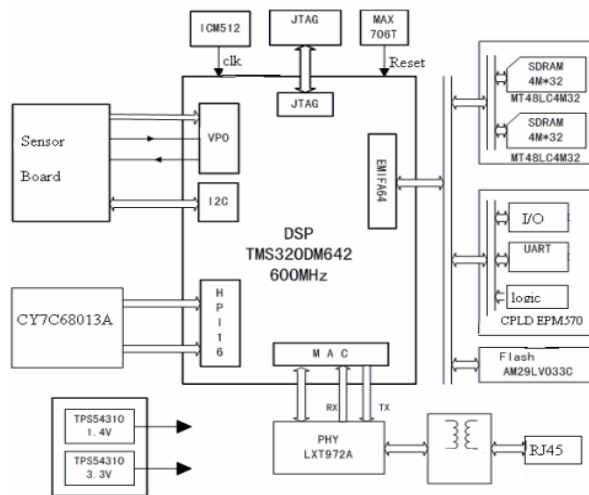


圖 1-9 基於 DSP 的嵌入式 CMOS 影像處理系統[7]

由本節內容看來，嵌入式影像處理可採用 MCU、DSP 或 FPGA 做為系統設計的基礎。由於 MCU 一般較偏向程序控制，並且運算效能與可連接的記憶體數量有限，對於處理高頻寬的即時影像資料顯得力有未逮，故本論文期望結合 DSP 強大的運算能力與 FPGA 的電路設計彈性，設計出一套適用於機器人的嵌入式影像處理系統。

### 1.3 問題描述與論文架構

物件辨識是機器人視覺中一項基本的功能，目的是找出感興趣的物體後再進一步地處理，例如：以視覺引導移動式機械手臂進行物件抓取，進行物件辨識之前則必須先找出相關的影像特徵，例如：形狀、大小或顏色等。在本實驗室以往的機器人系統中，大多使用工業電腦或結合不同種類的開發板來進行影像處理功能，使得體積龐大、重量加重，間接導致馬達驅動模組需要較大的功率與體積。因此，本論文結合 DSP 與 FPGA 設計出一套體積小但兼具高運算能力的嵌入式影像處理系統，希望裝載於機器人上進行影像處理。

本論文共分為六章。第一章是緒論。首先介紹此論文的研究動機，接著回顧一些相關的文獻；第二章說明嵌入式影像處理系統的硬體電路設計。基本上將整個系統分成 FPGA 與 DSP 兩個模組個別說明，另外則說明系統的電源需求；第三章簡介 SURF 演算法。先介紹 SURF 演算法能快速計算的基礎-積分影像，接著說明 Fast-Hessian 興趣點偵測器，最後是 SURF 描述子與快速匹配用的旗標；第四章是此嵌入式影像處理系統的應用。主要是在系統中實作 SURF 演算法，並說明相關的參數與使用的技術；第五章展示實作時的畫面擷圖與討論；第六章則是結論與未來的工作。

## 第二章 嵌入式影像處理系統硬體電路設計

### 2.1 系統方塊圖

本嵌入式影像處理系統主要分成DSP與FPGA兩個群組，DSP使用德州儀器的TMS320C6414TGLZ7高效能數位訊號處理器；FPGA採用Altera公司平價化的EP2C35F672C6 (Cyclone II系列產品)，兩者之間藉由DSP的EMIF-A匯流排及GPIO與中斷訊號進行資料傳遞；影像輸入採用CMOS彩色影像感測器，系統內包含多種不同用途的記憶體，可用來緩衝影像畫面、暫存運算資料或儲存程式碼等；系統對外的通訊介面包含一組高速USB與兩組RS232，圖2-1所示為系統的方塊圖。圖2-2是此系統製作完成後的實體圖及主要零件的位置。

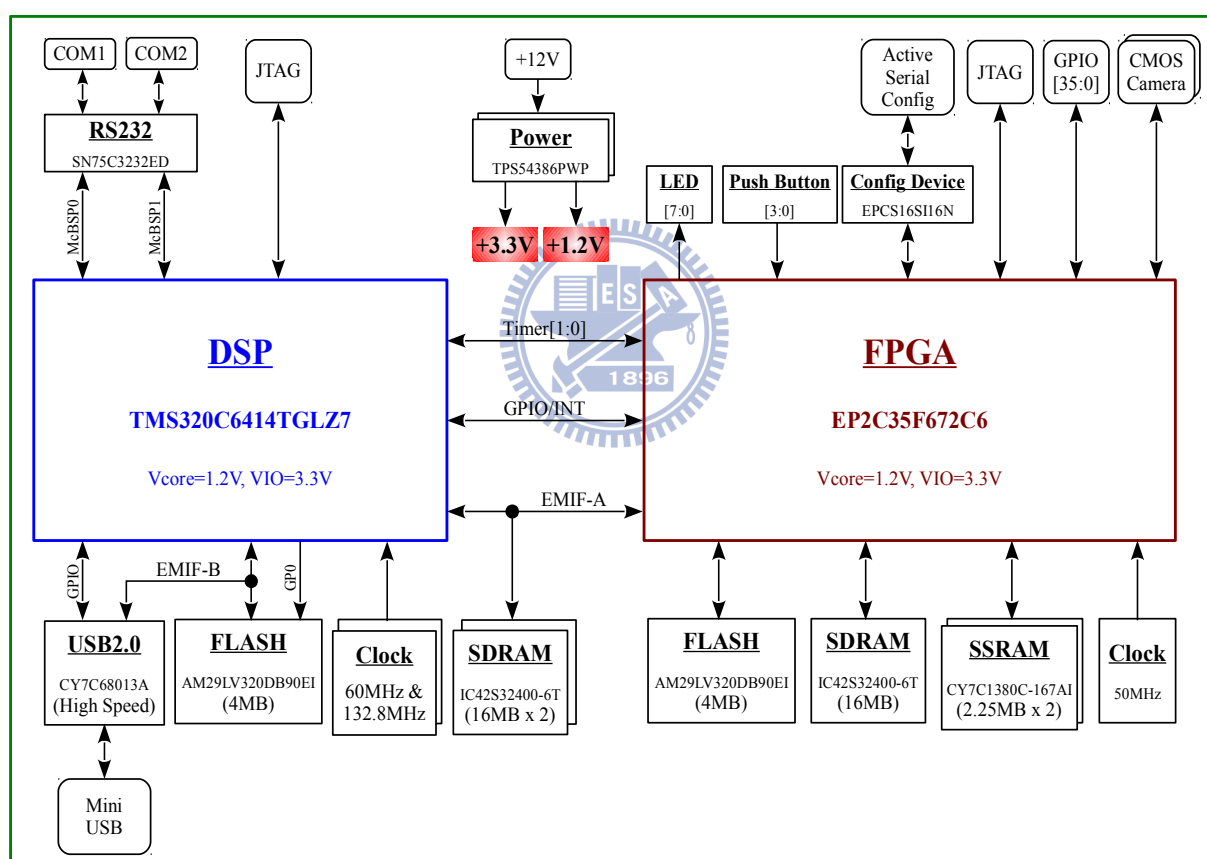


圖2-1 嵌入式影像處理系統方塊圖

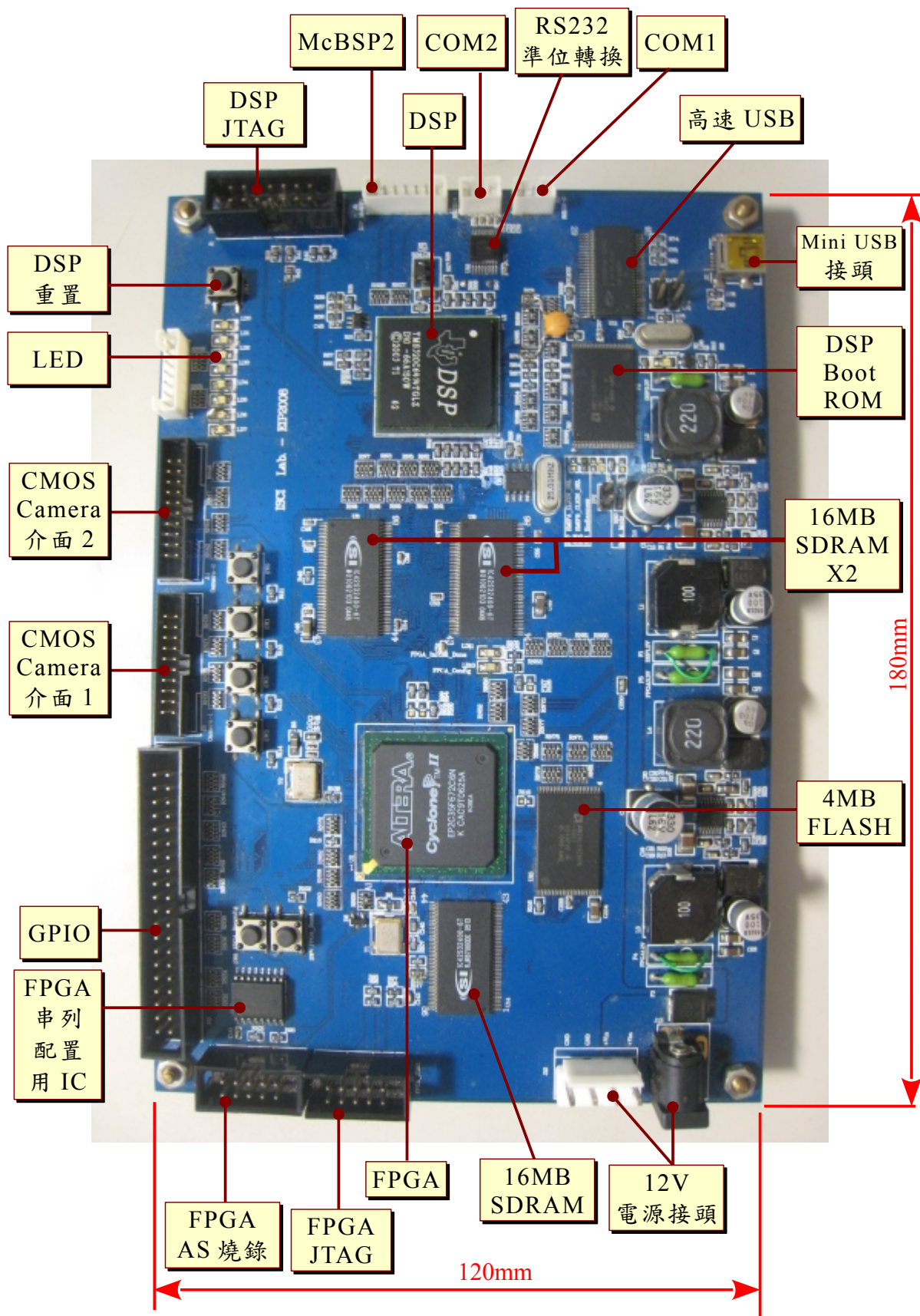


圖2-2 嵌入式影像處理系統實體圖

## 2.2 FPGA模組

本系統的FPGA型號為EP2C35F672C6，屬於Altera公司的平價化產品線，但是它仍擁有一些數位訊號處理與連接多種外部記憶體的能力，因此選用來當做DSP的協同處理器(Coprocessor)。以下列出此FPGA的部分特性，更詳細的特性請參閱Cyclone II Device Handbook[13]。

- 高密度架構，具有33216個邏輯元素。
- 內含472.5Kbit記憶體，可用做內部的FIFO、Dual-port RAM等。
- 內建35個18×18位元的硬體乘法器，最快速度可達250MHz。
- 支援多種的IO標準，例如：單端3.3V的LVTTTL/LVCMOS、高速差動訊號LVDS等。
- 可連接多種外部的記憶體，例如：SDRAM、DDR、DDR2、SSRAM、FLASH ROM等。

以下各小節將介紹FPGA的配置方式與相關的周邊元件。

### 2.2.1 FPGA配置方式

Cyclone II系列的FPGA配置(Configuration)方式有四種，由接腳MSEL1與MSEL0在上電重置(Power-On Reset, POR)或強制重新配置時的狀態決定，如表2-1所示。其中JTAG模式具有最高優先權，在此模式下MSEL1與MSEL0的狀態將被忽略，以下為配置模式的說明：

- **主動式串列模式(Active Serial Mode)**：使用串列式記憶體進行配置，例如：EPCS16、EPCS64等。在串列式記憶體正常動作之前，必須先使用下載線燒錄配置資料，FPGA在上電後會自動由此串列式記憶體讀取配置資料。
- **被動式串列模式(Passive Serial Mode)**：使用其它微處理器、下載線或增強型的配置用記憶體，例如：EPC8、EPC16等。
- **JTAG模式**：使用下載線的JTAG接腳、其它微處理器或標準測試與燒錄語言(STAPL)。此模式適合在開發初期對FPGA進行燒錄，可以避免對配置用記憶體頻繁地燒錄而降低其壽命。

圖2-3為FPGA的配置電路，本系統僅使用AS與JTAG模式，其中AS模式的速度可由電阻設定，預設為40MHz Fast AS模式。

表2-1 Cyclone II的四種配置方式

配置方式	MSEL1	MSEL0
20MHz 主動式串列模式, AS	0	0
被動式串列模式, PS	0	1
40MHz 快速主動式串列模式, Fast AS	1	0
JTAG 模式	-	-

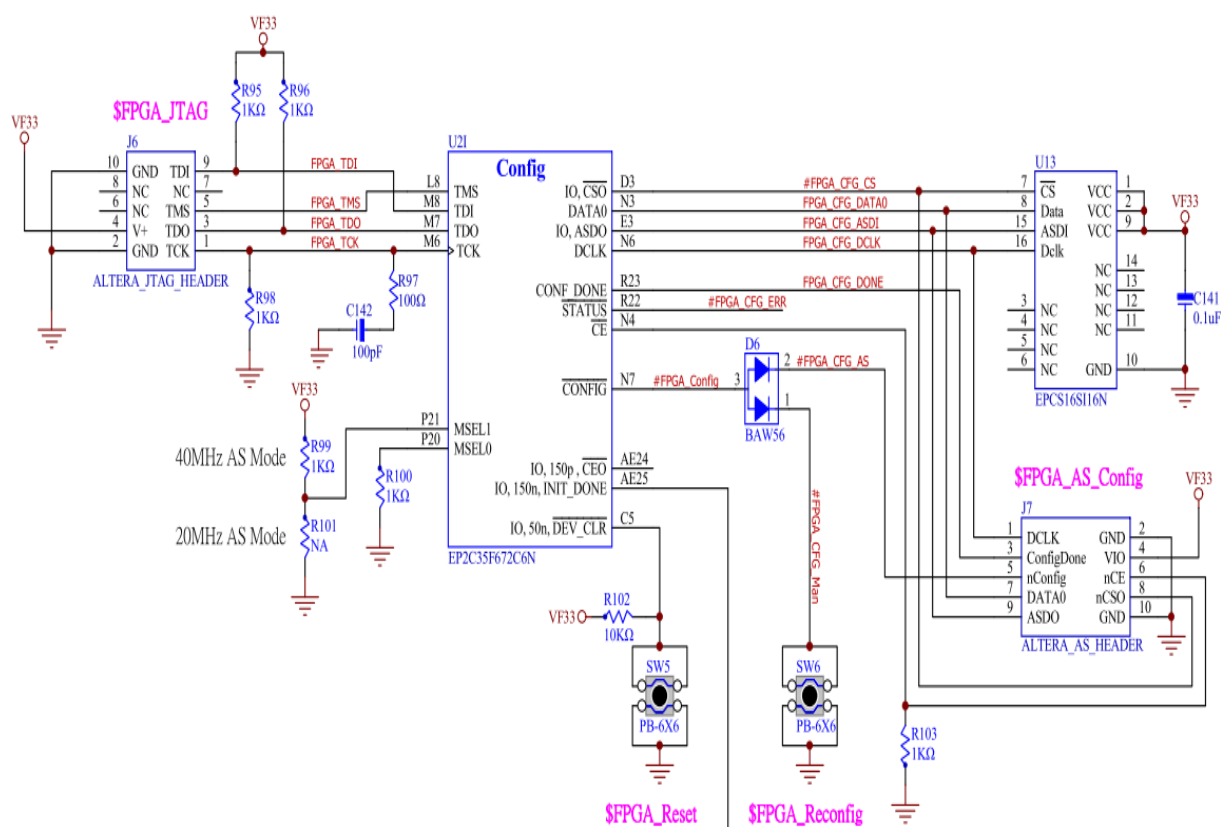


圖2-3 FPGA配置電路(AS、JTAG模式)

## 2.2.2 CMOS影像感測器介面

影像感測器主要有電荷耦合元件(Charge-coupled Device, CCD)與CMOS二種。本系統使用OmniVision公司的OV7649 CMOS影像感測器，基本參數如下：

- 最大影像尺寸640x480 (VGA格式)。
- 最大影像傳輸率，在VGA模式下可達30fps；QVGA模式則可達到60fps，因此能符合即時的要求。
- 核心與類比電壓為2.5V，IO電壓可使用3.3V，與FPGA的介面無需再另外做電壓準位的轉換。
- 輸出格式：YUV/YCbCr4:2:2、RGB4:2:2、Raw RGB Data。
- 使用標準的SCCB介面存取內部暫存器，可設定相關的操作模式。

為了使感測器進入穩定的操作狀態，在開始使用前必須先進行重置(Reset)動作，OV7649有二種重置方法：

- **硬體重置**：令RESET接腳為高電位並維持1毫秒，等待4096個外部時脈後感測器會進入穩定的狀態，如圖2-4所示。
- **軟體重置**：將0x80寫到暫存器0x12。此動作應執行二次，第一次寫入後要如圖2-4等待4096個外部時脈後再寫第二次。

OV7649的輸出訊號主要有PCLK、VSYNC、HREF、D[7:0]。所有的訊號變化都是發生在PCLK的負緣，因此FPGA必須在PCLK的正緣時取樣D[7:0]。在本次的實作中，採用OV7649的YUV4:2:2輸出格式，此格式屬於2:1的水平取樣，預設輸出順序為 $U_0Y_0V_0Y_1U_1Y_2V_1Y_3\cdots$ ，可組合成像素值 $[Y_0U_0V_0]$ 、 $[Y_1U_0V_0]$ 、 $[Y_2U_1V_1]$ 、 $[Y_3U_1V_1]\cdots$ ，相當於每4個輸出可組合成2個像素值。由於OV7649的輸出資料寬度為8-bit，因此在預設的影像大小(640X480)時，每一列會有1280 Bytes輸出。圖2-5為每列的輸出時序圖，圖2-6則是每個影像畫面的時序圖。



圖2-4 影像感測器OV7649硬體重置訊號

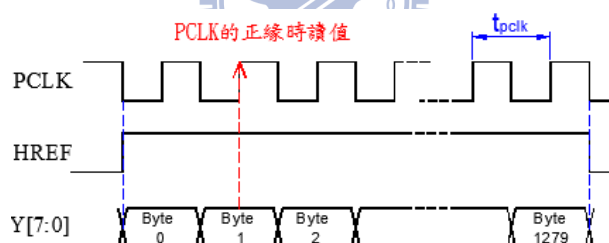


圖2-5 OV7649每列的輸出時序圖(細部時間參數詳見[14])

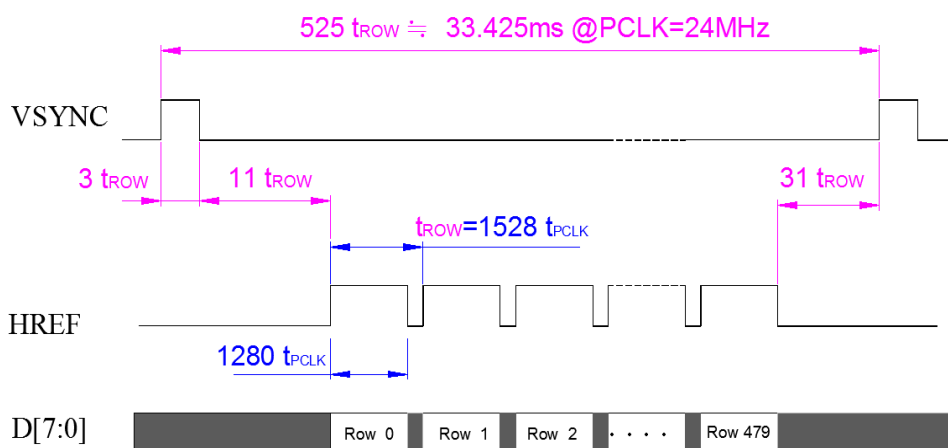


圖2-6 OV7649每個影像畫面的輸出時序圖

圖2-6中，HREF是水平同步訊號，當此訊號為高電位時OV7649將輸出影像資料，即圖中的Row0、Row1、... Row479，細部的時序如圖2-5所示。VGA畫面每列有640個像素，當輸出格式為YUV4:2:2時，每列會有1280筆影像資料輸出，加上換列的時間後HREF的週期 $t_{ROW}$ 則為1528個 $t_{PCLK}$ 。VSYNC是垂直同步訊號，在每個畫面開始時OV7649會輸出一個寬度為 $3t_{ROW}$ 的高準位脈波，其週期等於 $525t_{ROW}$ ，當 $PCLK=24MHz$ 時，畫面更新週期大約是33.425 毫秒，相當於每秒30個畫面。

### 2.2.3 FPGA之記憶體介面

FPGA模組包含三種記憶體，分別為：4.5MB SSRAM、16MB SDRAM及4MB FLASH ROM。為了能專注於其它影像功能的設計，將使用Altera公司提供的嵌入式周邊IP進行記憶體控制，屆時自行設計的IP只要符合Altera公司標準的Avalon介面時序即可存取記憶體內容。表2-2、表2-3、表2-4是各記憶體型號、使用的嵌入式周邊IP及訊號說明。

表2-2 連接於FPGA之SSRAM訊號說明

記憶體型號	CY7C1380C-167AC (SSRAM)	
需要的嵌入式周邊IP	Generic Tri-State Controller 及 Tri-State Conduit Bridge	
FPGA周邊IP訊號	SSRAM訊號	SSRAM訊號說明
tcm_address_out [20:2]	A[18:0]	記憶體位址
tcm_outputenable_n_out	$\overline{OE}$	輸出致能
tcm_byteenable_n_out [3:0]	$\overline{BW}_A$ 、 $\overline{BW}_B$ 、 $\overline{BW}_C$ 、 $\overline{BW}_D$	配合 $\overline{BWE}$ 訊號，選擇DQ[31:0]之中要寫入的是那幾個位元組
tcm_begintransfer_n_out	$\overline{ADSC}$	當此訊號為低電位，則位址會被門鎖至SSRAM的內部暫存器
tcm_write_n_out	$\overline{BWE}$	位元組寫入致能
tcm_data_out [31:0]	DQ[31:0]	寫入或讀出SSRAM的資料
tcm_chipselect_n_out	$\overline{CE1}$	晶片致能
--	CLK	同步用的時脈輸入，由FPGA內的PLL提供
--	$\overline{ADSP}$	未使用，維持高電位
--	$\overline{ADV}$	未使用，維持高電位
--	DQP[3:0]	額外的4位元資料，未使用
--	MODE	在電路預設為低電位，表示使用Linear Burst傳輸
--	ZZ	在電路預設為低電位，表示不進入睡眠模式
--	$\overline{GW}$	未使用，維持高電位
--	$\overline{CE2}$	在電路預設為高電位，晶片選擇僅由 $\overline{CE1}$ 控制
--	$\overline{CE3}$	在電路預設為低電位，晶片選擇僅由 $\overline{CE1}$ 控制

表2-3 連接於FPGA之SDRAM訊號說明

記憶體型號	IC42S32400-6T (SDRAM)	
需要的嵌入式周邊IP	SDRAM Controller	
FPGA周邊IP訊號	SDRAM訊號	SDRAM訊號說明
addr[11:0]	A[11:0]	記憶體位址
ba[1:0]	BS[1:0]	選擇內部的記憶體庫
cas_n	$\overline{\text{CAS}}$	行位址觸發
cke	CKE	時脈訊號致能
cs_n	$\overline{\text{CS}}$	晶片選擇
dq[31:0]	DQ[31:0]	寫入或讀出SDRAM的資料
dqm[3:0]	DQM[3:0]	輸出與輸入的資料遮罩
ras_n	$\overline{\text{RAS}}$	列位址觸發
we_n	$\overline{\text{WE}}$	寫入致能
--	CLK	同步用的時脈輸入，由FPGA內的PLL提供

表2-4 連接於FPGA之FLASH ROM訊號說明

記憶體型號	AM29LV320DB90EI (FLASH ROM)	
需要的嵌入式周邊IP	Generic Tri-State Controller 及 Tri-State Conduit Bridge	
FPGA周邊IP訊號	FLASH訊號	FLASH訊號說明
tcm_address_out[20:0]	A[20:0]	記憶體位址
tcm_outputenable_n_out	$\overline{\text{OE}}$	輸出致能
tcm_reset_n_out	$\overline{\text{RESET}}$	晶片重置
tcm_write_n_out	$\overline{\text{WE}}$	寫入致能
tcm_data_out[15:0]	DQ[15:0]	寫入或讀出FLASH ROM的資料
tcm_chipselect_n_out	$\overline{\text{CE}}$	晶片致能
--	$\overline{\text{WP/ACC}}$	寫入保護或加速燒錄時間，預設為高電位
--	$\overline{\text{READY/BUSY}}$	未使用

#### 2.2.4 晶體震盪器(Oscillator)

FPGA的輸入時脈為50MHz，再使用內部的鎖相迴路(PLL)產生較高的操作時脈。例如：本系統目前在FPGA內部使用120MHz的操作頻率，同時將此時脈做為外部SSRAM與SDRAM的同步時脈。

### 2.2.5 I/O擴充用介面

FPGA群組使用一個腳距為2.54mm的簡易牛角公座，保留總共36條I/O接腳，可做為日後擴充用。

## 2.3 數位訊號處理器(DSP)模組

本系統使用德州儀器(Texas Instruments, TI)公司的高效能定點式數位訊號處理器TMS320C6414TGLZ7，此系列產品主要應用於數位影像、通訊設備等，核心電壓1.2V，IO電壓則是3.3V，操作頻率為720MHz。TI為數位訊號處理器提供了完整的開發工具，包括整合式的開發環境 Code Composer Studio(CCS)、即時的作業系統DSP/BIOS、控制與配置DSP內部周邊的晶片支援程式庫 (Chip Support Library, CSL)、最佳化的C語言編譯器與組合語言組譯器等等，藉由這些工具的輔助，可以大幅減化開發的困難度。

### 2.3.1 外部記憶體介面(External Memory Interface, EMIF)

EMIF是DSP用來連接外部裝置的介面，如：非同步記憶體(SRAM、FLASH等)或同步式記憶體(SDRAM, SSRAM...)，本系統同時用來連接FPGA。C64x系列的EMIF分成EMIF-A與EMIF-B，表2-5是其特性的摘要說明。EMIF的時脈輸入來源是由EMIF-B的位址線決定(見2.3.3節)，透過EMIF的全域控制暫存器(GBLCTL)可以設定兩個時脈輸出訊號(如圖2-7)，此輸出之時脈用來同步化外部的裝置，例如：SSRAM、SDRAM或FIFO等。

表2-5 DSP的EMIF介面特性摘要

項 目	EMIF-A	EMIF-B
匯流排寬度	64-bit	16-bit
可定址空間(MBytes)	1024	256
支援的外部裝置寬度	8-, 16-, 32-, or 64-bit	8-, or 16-bit
記憶體空間數量	4 (CE0~CE3)	
控制訊號型式	多工、共用	
脈波輸出頻率	● 外部的ECLKIN接腳 ● CPU時脈/4 ● CPU時脈/6	
支援的記憶體類型	FLASH、SBSRAM、ZBT SRAM、FIFO及非同步記憶體等	

在本系統中，EMIF-A的輸出時脈是由外部的振盪電路產生，分別供給SDRAM(ECLKOUT1)與FPGA(ECLKOUT2)做為同步時脈之輸入，SDRAM的同步頻率為132.8MHz，FPGA的同步頻率則是66.4MHz；EMIF-B的輸出時脈則由CPU時脈除頻而來，目前由ECLKOUT2供應30MHz時脈給USB之Slave FIFO介面。

### 2.3.2 記憶體映射圖(Memory Map)及介面訊號

C64x採用記憶體映射的方式來存取內/外部的裝置，例如：EMIF-A的暫存器位於0x01800000~0x0183FFFF。系統中DSP的外部裝置映射到EMIF-A與EMIF-B的記憶體空間，表2-6為本系統DSP外部裝置的位址及範圍。表中Slave FIFO表示USB裝置CY7C68013A的操作模式。另外，由於FIFO的讀寫都是對同一個位址，因此其大小欄位相當於資料寬度。

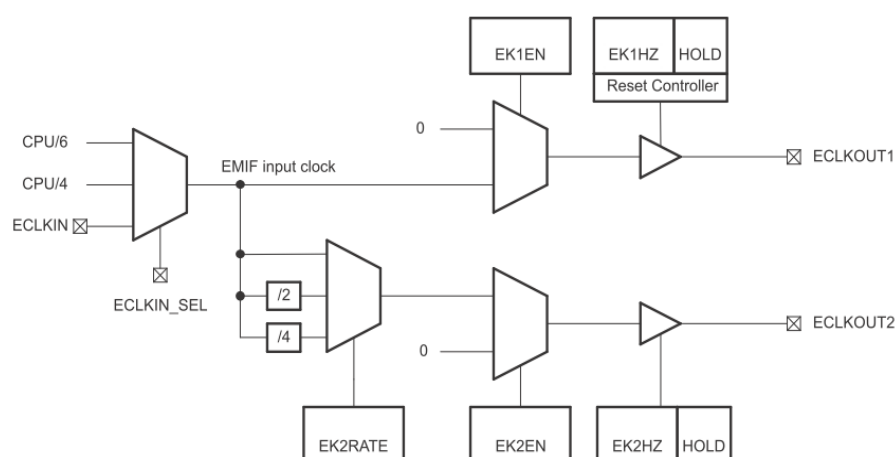


圖2-7 EMIF輸出時脈的路徑選擇[12]

表2-6 本系統的DSP記憶體映射圖

EMIF 記憶體空間	外部裝置	十六進制位址範圍	大小 (bytes)
EMIF-B CE1	FLASH ROM	6400 0000~643F FFFF	4M
EMIF-B CE3	Slave FIFO 輸入端點2	6C00 0000~6C00 0001	2
	Slave FIFO 輸入端點6	6C00 1000~6C00 1001	2
EMIF-A CE0	SDRAM	8000 0000~81FF FFFF	32M
EMIF-A CE1	FPGA 內部的影像FIFO	9000 0000~9000 0003	4
	FPGA 內部的積分影像FIFO	9000 0004~9000 0007	4
EMIF-A CE2	FPGA保留空間	A000 0000~AFFF FFFF	256M
EMIF-A CE3	FPGA保留空間	B000 0000~BFFF FFFF	256M

由於DSP的EMIF介面已經內建多種標準記憶體的控制時序，只要在使用前透過相關的暫存器調整時序參數即可，因此在選用DSP的記憶體時也儘量選擇標準的產品，以簡化電路的設計。表2-7、表2-8、表2-9分別是DSP與SDRAM、FLASH ROM與Slave FIFO的訊號連接方式與說明。

表2-7 連接於DSP EMIF-A之SDRAM訊號說明

記憶體型號	IC42S32400-6T (SDRAM) 共2顆，資料匯流排寬度為64	
EMIF-A訊號	SDRAM訊號	SDRAM訊號說明
AEA[14:3]	A[11:0]	記憶體位址
AEA[16:15]	BS[1:0]	選擇內部的記憶體庫
$\overline{\text{ARE}}/\text{SDCAS}/\text{SADS}/\text{SRE}$	$\overline{\text{CAS}}$	行位址觸發
ASDCKE	CKE	時脈訊號致能
$\overline{\text{ACE0}}$	$\overline{\text{CS}}$	晶片選擇
AED[63:0]	DQ[31:0] X 2	寫入或讀出SDRAM的資料
ABE[7:0]	DQM[3:0] X2	輸出與輸入的資料遮罩
$\overline{\text{AOE}}/\text{SDRAS}/\text{SOE}$	$\overline{\text{RAS}}$	列位址觸發
$\overline{\text{AWE}}/\text{SDWE}/\text{SWE}$	$\overline{\text{WE}}$	寫入致能
AECLKOUT1	CLK	同步用的時脈輸入

表2-8 連接於DSP EMIF-B之FLASH ROM訊號說明

記憶體型號	AM29LV320DB90EI (FLASH ROM)	
EMIF-B訊號	FLASH訊號	FLASH訊號說明
BEA[20:1]	A[19:0]	記憶體位址
GP0	A[20]	由於EMIF-B的位址線不足，故由一般輸出腳將此FLASH ROM分成兩個區塊
$\overline{\text{AOE}}/\text{SDRAS}/\text{SOE}$	$\overline{\text{OE}}$	輸出致能
--	$\overline{\text{RESET}}$	晶片重置，由外部重置晶片控制
$\overline{\text{AWE}}/\text{SDWE}/\text{SWE}$	$\overline{\text{WE}}$	寫入致能
BED[15:0]	DQ[15:0]	寫入或讀出FLASH ROM的資料
BCE1	CE	晶片致能
--	$\overline{\text{WP}}/\text{ACC}$	寫入保護或加速燒錄時間，預設為高電位
--	$\overline{\text{READY}}/\text{BUSY}$	未使用

表2-9 連接於DSP EMIF-B之Slave FIFO訊號說明

DSP周邊型號	CY7C68013A (操作於Slave FIFO模式的高速USB)	
EMIF-B及GPIO 訊號	Slave FIFO 訊號	Slave FIFO 訊號說明
BEA[12:11]	FIFOADR[1:0]	選擇內部的輸入或輸出端點(2, 4, 6, 8)
$\overline{AOE}/\overline{SDRAS}/\overline{SOE}$	$\overline{SLOE}$	輸出致能
$\overline{ARE}/\overline{SDCAS}/\overline{SADS}/\overline{SRE}$	$\overline{SLRD}$	讀取致能
$\overline{AWE}/\overline{SDWE}/\overline{SWE}$	$\overline{SLWR}$	寫入致能
BED[15:0]	FD[15:0]	寫入或讀出Slave FIFO的資料
$\overline{BCE3}$	$\overline{SLCS}$	晶片致能
BECLKOUT2	IFCLK	同步用的時脈輸入
GP1	FLAGA	輸入端點2的FIFO已填滿旗標
GP2	FLAGB	輸入端點6的FIFO已填滿旗標
以下三個訊號是PC的命令封包被USB轉成串列方式傳到DSP時所需的訊號		
GP11	PA3	封包起始訊號
GP4	PA1	同步用脈波
GP10	PA0	命令資料

### 2.3.3 內部周邊配置

C6414T的內部周邊配置主要是根據EMIF-B位址線在重置時的電壓準位來決定。由於C6414T不含PCI功能，故PCI\_EN接腳必須為低電位，使得相關共用接腳的周邊自動被致能，包括Host-Port Interface (HPI)、GPIO[15:9]及McBSP2。另外，程式開發時也必須符合硬體的配置，例如：CSL程式庫的選用要與Endian模式一致、FPGA使用EMIF-A介面傳送資料時也必須知道其時脈頻率。表2-10說明DSP內部周邊配置用之接腳，表中的預設值表示本系統設定的操作模式。

### 2.3.4 開機模式

C6414T的開機模式有三種，分別為無開機模式、ROM開機模式與HPI開機模式，本系統僅使用無開機模式及ROM開機模式二種。

#### a) 無開機模式

此模式適用於開發階段，使用JTAG介面的模擬器進行開發與除錯，程式的載入與執行由開發環境控制。

## b) ROM開機模式

為了讓嵌入式影像處理系統脫離開發環境而獨立運作，DSP開發完成後必須將程式碼燒錄到外部的FLASH ROM，再將開機模式設為ROM開機，上電重置後C6414T會自動載入EMIF-B CE1的前1K Bytes程式碼然後開始執行。由於DSP的程式碼大部分都會超過1K Bytes，故此區塊通常均存放自行撰寫的二次開機載入程式(Secondary Bootloader)，再由此程式將其它程式節區搬移到執行用記憶體。

表2-10 C6414T內部周邊配置說明

配置用 接腳名稱	功 能 說 明
BEA20	資料排列的順序 0: Big Endian mode 1: Little Endian mode (預設)
BEA[19:18]	開機模式 00: 無開機，使用JTAG模擬器開發時用(預設) 01: HPI開機 10: 使用連接在EMIF-B CE1空間的8-bit FLASHROM開機 11: 保留
BEA[17:16]	EMIF-A時脈輸入來源 00: DSP的AECLKIN接腳 (預設) 01: CPU時脈/4 10: CPU時脈/6 11: 保留
BEA[15:14]	EMIF-B時脈輸入來源 00: DSP的BECLKIN接腳 01: CPU時脈/4 10: CPU時脈/6 (預設) 11: 保留
BEA13	PCI EEPROM 自動初始化，此功能僅用在C6415T與C6416T，使用C6414T時不可改變此接腳預設的pulldown狀態
BEA11	UTOPIA致能，此功能僅用在C6415T與C6416T，使用C6414T時不可改變此接腳預設的pulldown狀態。
BEA[9:7]	僅用在C6415T與C6416T，使用C6414T時不可改變這些接腳預設的pulldown狀態。
HD5	HPI的匯流排寬度 0: 使用16-bit寬的HD[15:0] 1: 使用32-bit寬的HD[31:0]

本系統使用的FLASH ROM型號為AM29LV320DB90EI，除了儲存程式碼也可用來儲存執行時需要的參數資料。由於EMIF-B的位址線不足，故使用GP0來將FLASH ROM空間分成二個區塊，在DSP上電重置時GP0預設為低電位，因此在開機階段使用的是FLASH ROM的前半個區塊。另外FLASH ROM在電路上設定為Word(16-bit)模式，因此讀寫時必須使用其Word模式的命令[17]。但是C6414T在開機自動載入階段預設只能使用8-bit ROM，所以要將程式碼燒寫入FLASH ROM時必須使用8-bit的資料型別，此時邏輯位址bit[0]對應到EMIF-B的BEA[1]再連接至FLASH ROM的位址A[0]。

### 2.3.5 外部裝置的通訊介面

本系統與外部裝置的通訊介面有二種。第一種是高速USB介面，主要是與PC或其它主控制器通訊，用來接收命令或上傳影像及特徵資料；第二種則是利用DSP的McBSP周邊模擬RS232的介面，可用來連接機器人的馬達控制模組等。

#### a) 高速USB 介面

此介面使用Cypress公司的EZ-USB FX2LP高速USB控制器，型號為CY7C68013A，連接於DSP的EMIF-B匯流排。為了達到最大的傳輸頻寬，因此將CY7C68013A設定為同步式Slave FIFO模式，圖2-8是寫入時的波形。圖中的時間單位為ns，時間參數則是Slave FIFO所需的參數，括號內的數值是它的最小值，其它數值則是相對於原始BECLKOUT2或IFCLK正緣的時間。另外，假設IFCLK在內部反相後與BECLKOUT2的時間延遲可以忽略。

Slave FIFO的同步時脈IFCLK由EMIF-B的第二組時脈BECLKOUT2提供，由於在電路設計初期預設是使用非同步的傳輸，後來由於頻寬不足必須改用同步式的Slave FIFO，導致時間參數 $t_{SWR}$ 與 $t_{SFA}$ 無法滿足要求，因此最後在BECLKOUT2的串聯電阻後面再並聯一個150pF左右的電容形成一組RC延遲電路，讓原始的BECLKOUT2與輸入到IFCLK的訊號相差大約5.4ns，如此 $t_{SWR}$ 即可符合時序，但是 $t_{SFA}$ 則會落在臨界值。因此為了保險起見，在切換不同的輸入端點後(改變FIFOADDR)，要先讀取相同CE3空間中虛設的位置，讓FIFOADDR提早一個IFCLK週期被設定好。

除了前述的IFCLK問題之外，因為CY7C68013A的資料手冊中只有標示IFCLK正緣變化後到資料輸出的最大時間，並未明確指出它的最小時間，因此無法確認是否符合EMIF-B的讀取時序。考量到實際傳輸影像或特徵時才需要大量

的頻寬，因此只有在DSP上傳時使用同步式Slave FIFO介面；當USB控制器接收到PC的命令後會先轉換成串列資料，再利用GPIO及中斷訊號將命令送到DSP。

#### b) RS232介面

本系統預留二個RS232的介面，用來與機器人的馬達控制器或其它裝置通訊。由於C6414T不直接支援標準UART，因此本系統利用McBSP0與McBSP1來模擬RS232介面。McBSP可以被設定成兩種模式來達到此目的：串列埠模式或一般IO模式[24]。一般IO模式並非限定使用McBSP，在[24]裏有提供相關的範例程式。本系統將McBSP的電路設計成串列埠模式，連接方式如圖2-9所示。

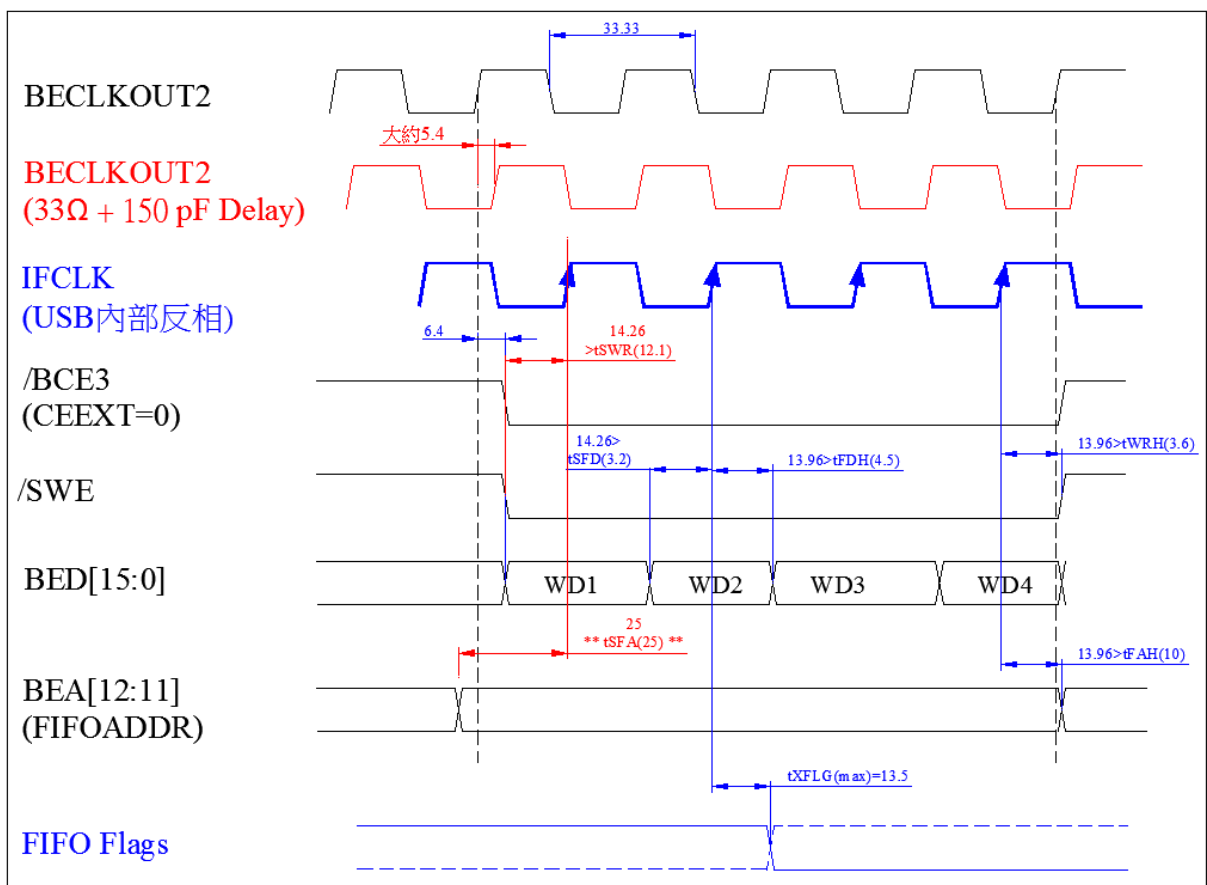


圖2-8 EMIF-B與同步式Slave FIFO介面波形

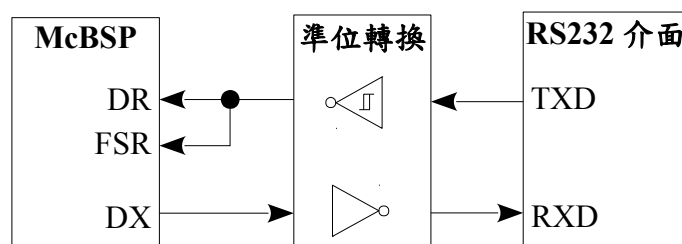


圖2-9 McBSP使用串列埠模式連接RS232的方式

如圖2-9所示，因為RS232的TXD訊號包含資料位元與資料框(frame)的訊息，故將此訊號連接到McBSP的資料輸入(DR)與資料框的同步化輸入(FSR)，RXD訊號則連接到McBSP的資料傳送(DX)。

使用McBSP模擬RS232時，每個資料位元以16-bit字組表示，相當於McBSP的取樣頻率等於鮑率的16倍。軟體必須將待傳的每個位元擴展成16-bit字組，同時將收到的每16-bit字組壓縮成一個資料位元。另外，RS232的起始位元是一個負緣訊號，因此McBSP的資料框同步也要設為負緣觸發。以RS232常用的N, 8, 1傳輸模式(無同位元，8個資料位元，1個停位元)為例，可以將每個傳送的位元組分成二個部分，第一部分包含起始位元與8個資料位元，每個位元都以16-bit字組表示；第二部分則是將一個停止位元用2個8-bit位元組表示，當使用1.5個停止位元時，則可改用3個8-bit位元組表示，圖2-10所示為McBSP傳送N,8,1模式的RS232位元組的波形。

### 2.3.6 GPIO與中斷訊號

C6414T的GPIO用於接收經由USB裝置傳來的串列式命令或監控Slave FIFO的狀態。另外，當FPGA內的FIFO被填滿後，會發出一個觸發EDMA的訊號將FIFO的內容搬移到DSP內部。



### 2.4 電源模組

此次設計的嵌入式影像處理系統最終希望能使用在機器人平台，因此系統的電源輸入預設為12V，但是系統中DSP與FPGA的核心電壓都是1.2V，IO電壓則是3.3V，由於輸入與輸出之間電壓差太大，如果使用一般的線性穩壓器，會因為穩壓器本身的功率消耗太高，而需要另外處理散熱問題。因此，最後選用德州儀器公司的降壓型交換式DC-DC電壓轉換器，型號為TPS54386PWP。此轉換器的電壓輸入範圍為4.5~28V，最大輸出電流是3A，有2組輸出電壓，電壓值則可以使用外部電阻做調整。

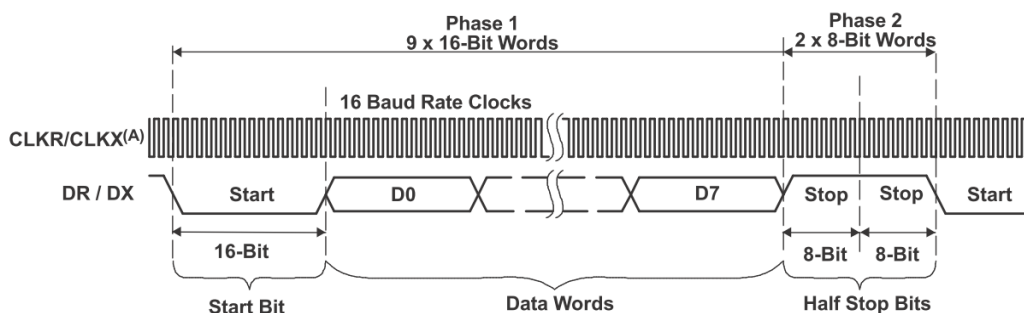


圖2-10 McBSP傳送N, 8, 1模式的RS232位元組的波形[24]

## 2.5 設計總結

本章主要在說明FPGA與DSP兩個模組的硬體電路設計。在電路設計時，首先要考慮到FPGA電路內容與DSP程式的下載與燒錄方式。在演算法開發初期會有一段除錯的期間，此時利用JTAG介面進行下載，可以避免對FPGA的配置用記憶元件及DSP的開機記憶體頻繁地燒錄而降低其使用壽命，另外使用JTAG下載的時間較短也可以減少除錯的等待時間。當開發完成後，為了使整個系統可以獨立運作，也必須提供燒錄介面將最後的電路內容與DSP程式燒錄至記憶體。FPGA部分使用的是AS介面加上串列式配置用記憶體；DSP程式則是利用USB介面加上自行撰寫的燒錄函式將其燒錄到FLASH ROM。

由於DSP已經將一些標準型記憶體的控制時序設計在EMIF介面中，使用者只要設定相關的暫存器就可以調整其時序參數；FPGA廠商或其第三方協力廠也有提供一些記憶體控制器的IP，同樣可經由參數設定而調整記憶體的控制時序。有鑑於此，在記憶體選用時我們儘量選擇DSP或FPGA支援的記憶體類型，或者參考現有開發板使用的記憶體型號。如此可避免記憶體不相容而花費多餘時間做時序的調整，從而專注於影像演算法的開發與設計。



### 第三章 SURF 演算法簡介

在不同影像畫面中找出某些點的相關性在電腦視覺應用裏是一個基本的步驟。要達到此目的，第一步要在影像中找出某些具有獨特性的點，例如：轉角、T 型交接處等，此步驟一般稱為興趣點偵測(Interest Point Detecting)。第二步則是收集興趣點鄰近像素的資訊然後組成一個特徵向量，例如：光強度的變化梯度等，此向量稱為描述子(Descriptor)。最後一步則是針對不同影像中的描述子進行匹配的動作，其中一種做法是比較各特徵向量之間的距離，當距離在某一個門檻值之內則視為相同的點。

SURF(Speeded-Up Robust Features)[1]是由 Bay 等人所提出，主要包含興趣點偵測器(Interest Point Detector)及特徵描述子(Feature Descriptor)。另外，在運算過程中 Hessian 矩陣的跡(the trace of the Hessian matrix)，也可在匹配階段先剔除正負號不同者，避免直接進行耗時的距離計算，以加快整體匹配的速度。SURF 主要強調的是尺度與平面旋轉的不變性，若與其它功能相近的演算法相比，在重覆性、獨特性與強健性方面也都能獲得相當不錯的效果，除此之外，SURF 的另一項主要優點就是運算快速。

為了在機器人平台擷取即時影像特徵，因此在本次設計的嵌入式影像處理系統實作 SURF 演算法，期望藉由硬體電路的特性，減少特徵擷取的時間。例如：以 FPGA 設計影像積分的功能，可以在接收完一個畫面後同時將積分影像計算完成。本章將對 SURF 演算法相關的技術做說明。

#### 3.1 積分影像

SURF 演算法能夠達到快速運算的主要原因就是使用積分影像來計算相關的迴旋積分。圖 3-1 所示為積分影像的示意圖，假設  $I(x,y)$  表示影像中某一點  $\mathbf{x}=(x,y)$  的函數值(例如：光的明亮度)，則積分影像定義如(3-1)式。

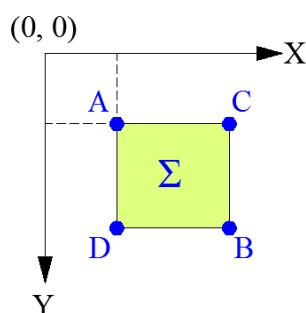


圖 3-1 積分影像

$$H(\mathbf{x}) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(i, j) \quad (3-1)$$

一旦積分影像計算完成，往後要計算影像中直立矩形區域(如圖 3-1 的  $\Sigma$ )的積分值只要三個加法運算即可，如(3-2)式，此時不論矩形區域的大小為何，運算時間將接近常數值。

$$H_{\Sigma} = H_A + H_B - (H_C + H_D) \quad (3-2)$$

### 3.2 Fast-Hessian 偵測器

一個二維影像的線性尺度空間(linear scale space)表示法，定義成高斯核心  $G(x, y, \sigma)$  與原始影像  $I(x, y)$  的迴旋積分：

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (3-3)$$

其中，高斯核心  $G(x, y, \sigma)$  的定義如下式：

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \quad (3-4)$$

另外，若  $\mathbf{x}=(x, y)$  表示影像中的任一點，則 Hessian 矩陣  $H$  可定義成  $\mathbf{x}$  與尺度  $\sigma$  的函數：

$$H(\mathbf{x}, \sigma) = \begin{bmatrix} L_{xx}(\mathbf{x}, \sigma) & L_{xy}(\mathbf{x}, \sigma) \\ L_{xy}(\mathbf{x}, \sigma) & L_{yy}(\mathbf{x}, \sigma) \end{bmatrix} \quad (3-5)$$

式中各矩陣元素分別定義如下，

$$L_{xx} = \frac{\partial^2}{\partial x^2} G(\sigma) * I \quad (3-6)$$

$$L_{yy} = \frac{\partial^2}{\partial y^2} G(\sigma) * I \quad (3-7)$$

$$L_{xy} = \frac{\partial^2}{\partial x \partial y} G(\sigma) * I \quad (3-8)$$

上面三個式子分別表示在尺度  $\sigma$  之下對高斯核心  $G(x, y, \sigma)$  取  $x$ 、 $y$  與  $xy$  三個方向的二階偏導數後再與原始影像進行迴旋積分，圖 3-2 則是在某個尺度下將其離散化並且裁剪後的圖形。SURF 的興趣點偵測器就是以 Hessian 行列式為基礎，先定義近似的高斯二階偏導數，再配合前述的積分影像來快速計算 Hessian 行列式，最後在不同尺度之下以 3x3x3 非最大抑制法(Non-Maximum Suppression)尋找影像中的興趣點。

#### 3.2.1 近似的高斯行列式

前述離散化之後的高斯二階導數若直接與影像進行迴旋積分運算將會相當耗

時，為了能發揮積分影像快速計算的特性，因此 SURF 作者提出一種箱形濾波器 (Box Filter) 來近似高斯二階導數，使得計算不同尺度下的 Hessian 行列式只要改變 Box Filter 的大小，同時維持 Box Filter 結構的比例即可，圖 3-3 即為近似的高斯二階偏導數濾波器及權重值。圖中 9x9 的箱形濾波器屬於尺度空間裏最低的尺度(最高的空間解析)，近似於高斯函數的尺度參數  $\sigma=1.2$ ，其它的尺度也可以使用(3-12)式由箱形濾波器的邊長求得。運用這三種結構的箱形濾波器於不同的尺度空間，可以得到各尺度空間裏  $x$ 、 $y$  與  $xy$  軸近似的高斯二階導數，再配合積分影像做迴旋積分運算可得到(3-6)到(3-8)式的近似值： $D_{xx}$ 、 $D_{yy}$  及  $D_{xy}$ ，最後由(3-5)式可以得到近似的 Hessian 行列式值如(3-9)式，此近似值表示在尺度空間中某一點  $\mathbf{x}=(x, y, \sigma)$  的斑點響應(Blob Response)。

$$\det(H_{\text{approx}}) = D_{xx}D_{yy} - (wD_{xy})^2 \quad (3-9)$$

式中  $w$  是為了使原本的高斯核心與近似的高斯核心維持能量守恒(energy conservation)，其值定義為：

$$w = \frac{|L_{xy}(1.2)|_F |D_{yy}(9)|_F}{|L_{yy}(1.2)|_F |D_{xy}(9)|_F} = 0.912... \approx 0.9 \quad (3-10)$$

其中  $|\cdot|_F$  表示弗羅貝尼烏斯範數(Frobenius norm)。雖然  $w$  的值會隨尺度而變，但 SURF 作者實驗後認為保持常數對結果並不會產生太大的影響。

在尺度空間裏對每一點求得上述的斑點響應後，再利用對應的濾波器大小將其正規化即可組成多張的斑點響應圖(blob response map)，將用來尋找興趣點。

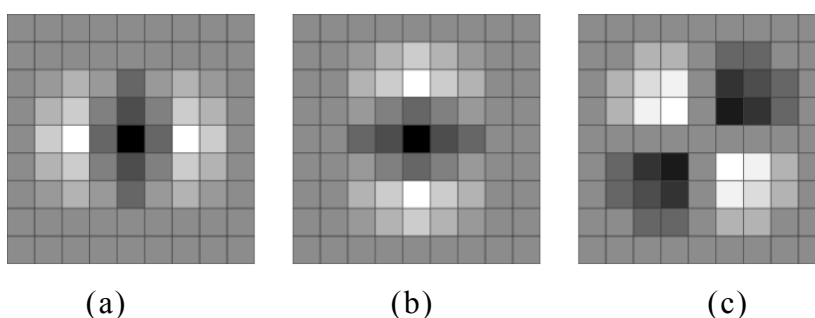


圖 3-2 離散化並且裁剪後的高斯二階偏導數[9] (a)x 方向 (b)y 方向 (c)xy 方向

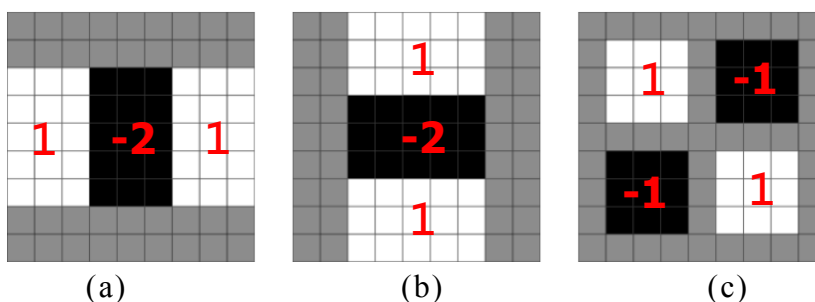


圖 3-3 近似的高斯二階偏導數濾波器及權重值[9] (a)x 方向 (b)y 方向 (c)xy 方向

### 3.2.2 尺度空間表示

為了在不同的影像尺度下尋找興趣點的相關性，必須使用尺度空間的表示法。在電腦視覺領域裏，一般以影像金字塔來實現尺度空間，但此方法必須迭代地將輸入影像與高斯核心做迴旋積分後再進行縮減取樣(Downsampling)，因此相當耗時。有別於傳統的影像金字塔，SURF 使用積分影像與前述的箱形濾波器，藉由改變濾波器的大小來建立多尺度空間。由於箱形濾波器的計算時間與其尺寸無關，而且可以同時對輸入影像進行運算，因此可提高運算速度。圖 3-4 是兩種尺度空間的示意圖。

尺度空間由多個 Octave 組成，每個 Octave 則包含多張不同尺度的斑點響應圖(Blob Response Map)，尺度的設定則是依據箱形濾波器的大小。濾波器尺寸變化時要保留中心像素，假設  $OctaveId=0,1,2, \dots$  表示濾波器所處的 Octave 編號， $MapId=0,1,2, \dots$  表示 Map 編號，則由下式可計算出濾波器邊長：

$$L_{Filter}=3\times[1+(MapId+1)\times 2^{OctaveId+1}] \quad (3-11)$$

在 SURF 的原始定義中， $9\times 9$  的濾波器用於尺度空間的最底層 ( $OctaveId=0, MapId=0$ )，其對應近似的高斯二階導數尺度為  $\sigma=1.2$ ，因為濾波器尺寸不同時仍需維持相同的比例，由濾波器的邊長可以算出對應的近似尺度為

$$\begin{aligned} \sigma_{approx} &= \frac{\text{最底層濾波器尺度}}{\text{最底層濾波器邊長}} \times \text{已知的濾波器邊長} \\ &= \frac{1.2}{9} \times \text{已知的濾波器邊長} \\ &= 0.1333 \times \text{已知的濾波器邊長} \end{aligned} \quad (3-12)$$

圖 3-5 為 Octave 數量等於三，斑點響應圖等於四的時候，濾波器的邊長圖示。由圖中可看出濾波器邊長的增加量會隨著 Octave 而加倍。

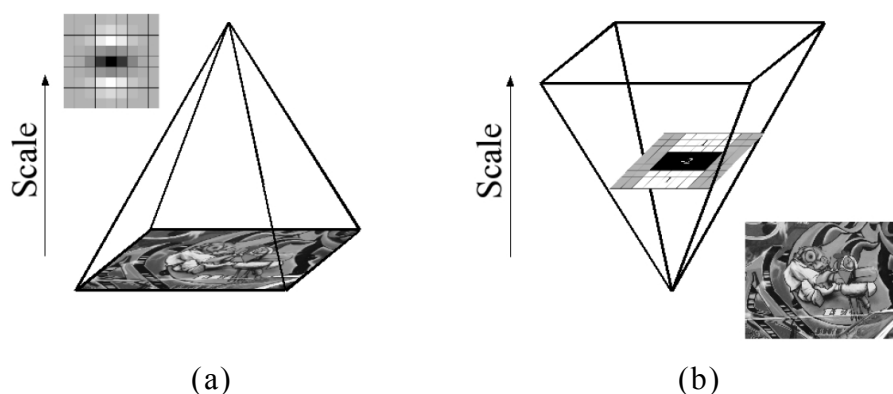


圖 3-4 尺度空間示意圖。(a)傳統的影像金字塔 (b)SURF 的方法[1]

### 3.2.3 興趣點定位

找出影像中的興趣點分成三個步驟。首先，比較候選點的 Hessian 行列式值與預先設定的門檻值(Threshold)，找出行列式值大於門檻值的點。第二步，使用 3x3x3 的非最大抑制(Non-Maximal Suppress)法在尺度空間中比較候選點與其它 26 個相鄰的點，如圖 3-6 所示。圖中的 X 符號表示候選點，如果它的 Hessian 行列式值大於圖中其它 26 個點，則表示此點為區域性的極值。另外，若此點確實為極值，則其鄰近的下一個取樣點(行、列、Map)必定不會是區域性的極值，可跳過比較以加快計算時間[10]。最後，則是利用鄰近像素的 Hessian 行列式值以內插法求得在空間與尺度下的次像素精度(Sub-pixel Accuracy)，此法以 Brown[11] 所提出的擬合 3 維的二次多項式完成。假設  $\mathbf{x}=(x, y, \sigma)$  為尺度空間中的一點，則 Hessian 函數  $H(\mathbf{x})$  的泰勒展開式表示為

$$H(\mathbf{x}) = H + \frac{\partial H}{\partial \mathbf{x}}^T \mathbf{x} + \frac{1}{2} \mathbf{x}^T \frac{\partial^2 H}{\partial \mathbf{x}^2} \mathbf{x} \quad (3-13)$$

取上式的導數後令其為 0，則興趣點在尺度空間的位置為

$$\hat{\mathbf{x}}(x, y, \sigma) = -\frac{\partial^2 H^{-1}}{\partial \mathbf{x}^2} \frac{\partial H}{\partial \mathbf{x}} \quad (3-14)$$

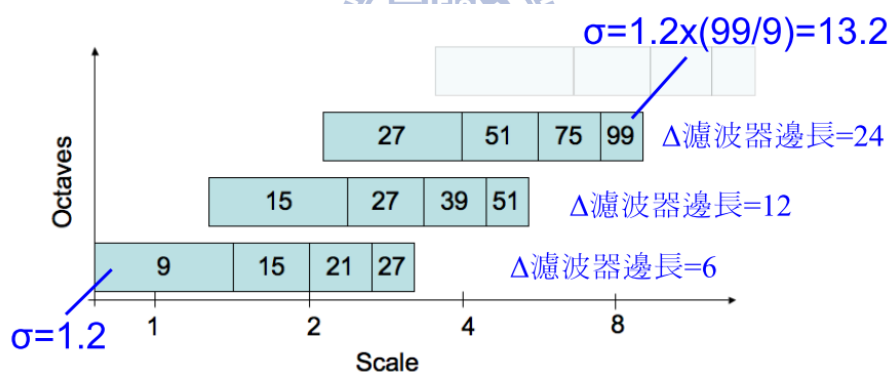


圖 3-5 濾波器邊長[1]

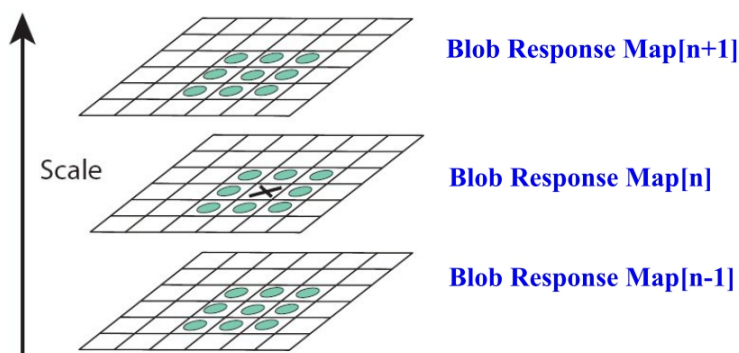


圖 3-6 3x3x3 非最大抑制[9]

### 3.3 興趣點描述子

SURF 的描述子是基於興趣點周圍像素的光強度分佈情形。使用圖 3-7 所示之 Haar 小波濾波器配合積分影像來計算  $x$ 、 $y$  方向的響應值，每個興趣點使用 64 維的向量來描述，可以使描述子的計算及特徵匹配速度加快。描述子的計算主要分成二部分。首先，尋找可重現的主方位，接下來以此方位為基準在一個與尺度相關的視窗中計算出 64 維的描述子。前述所有的計算都是相對於興趣點被偵測到的尺度，如此才能達到尺度的不變性。

#### 3.3.1 可重現的主方位

尋找興趣點可重現的主方位目的是為了達到影像的旋轉不變性。步驟如下：

- 以興趣點被偵測到的尺度做為取樣的間距  $s$ 。
- 令 Haar 濾波器的邊長為  $4s$ ，以興趣點為中心在半徑  $6s$  的範圍內計算每個取樣點在  $x$ 、 $y$  方向的 Haar 小波響應，同時乘上標準差  $\sigma=2s$  的高斯係數。
- 以角度為  $\pi/3$  的扇形區域繞著興趣點以微小的角度增量旋轉一周，同時在每個角度時分別累加  $x$ 、 $y$  方向的 Haar 小波響應後組成一個新的向量。圖 3-8 表示此搜尋過程，扇形中的箭頭即為 Haar 小波響應組成的向量。
- 最後，比較所有的向量大小，最大者即指定為興趣點的主方位。

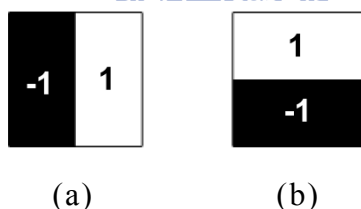


圖 3-7 Haar 小波濾波器[9]。(a)(b)分別用於計算  $x$ 、 $y$  方向的響應值

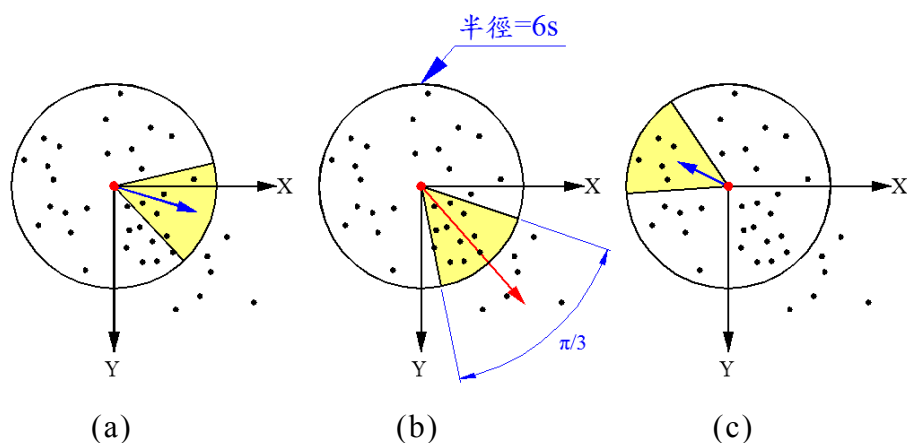


圖 3-8 主方位的搜尋。(a)至(c)表示以  $\pi/3$  的扇形區域繞著興趣點旋轉一周 (b)表示此興趣點的主方位

### 3.3.2 基於 Haar 小波響應和的描述子

興趣點的描述子是由平行與垂直於前述主方位的 Haar 小波響應和所組成。以下是計算描述子的步驟，圖 3-9 為邊長  $20s$  的 SURF 描述子計算範圍示意圖。

- 以興趣點被偵測到的尺度做為取樣的間距  $s$ 。
- 以興趣點為中心，建立一個以前述的主方位為水平軸( $X'$ )且邊長為  $20s$  的虛擬方塊，做為計算 Haar 小波響應的範圍。
- 將此虛擬方塊再分成  $4 \times 4$  個方形子方塊，每個子方塊裏則包含有  $5 \times 5$  個取樣點。
- 使用圖 3-7 所示，邊長為  $2s$  的 Haar 小波濾波器與標準差  $\sigma=3.3s$  的高斯權重係數，計算子方塊裏 25 個取樣點在  $X'$  與  $Y'$  方向的 Haar 小波響應值，記為  $dx$  與  $dy$ 。
- 將每個子方塊裏的響應值及其絕對值相加，形成一個 4 維向量  $\mathbf{v}=(\sum dy, \sum dx, \sum |dy|, \sum |dx|)$ ，整個虛擬方塊裏總共可得到一個  $4 \times 4 \times 4=64$  維的向量。
- 計算此 64 維向量的單位向量後即為 SURF 描述子。

### 3.4 快速匹配用的旗標

在計算前述的 Hessian 行列式時，可以同時得到 Hessian 矩陣的跡(trace)。在匹配階段可以先判斷兩個描述子的跡之正負號是否相同而不必直接進行向量的比對，因此可加快匹配階段的速度。另外，由於跡在計算 Hessian 行列式時就可順便求得，所以不必另外花費計算時間。

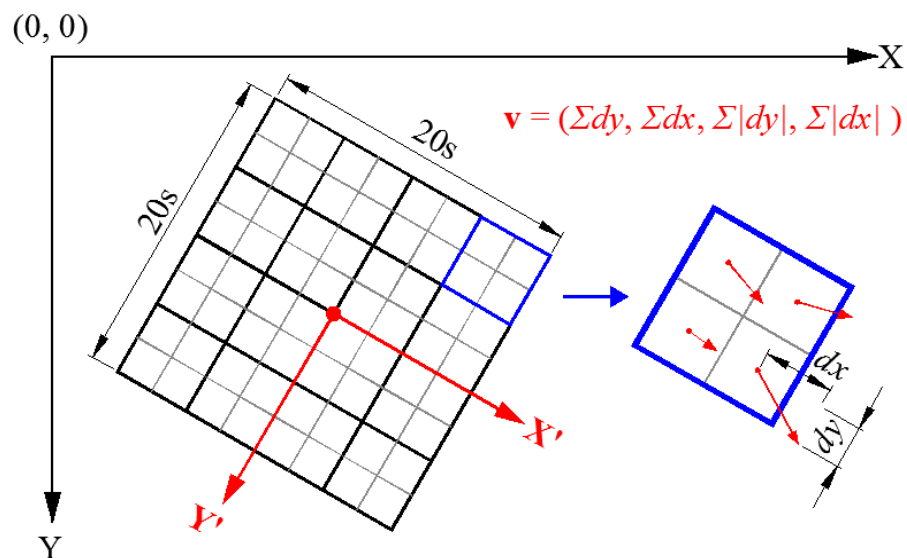


圖 3-9 以興趣點為中心之 SURF 描述子計算範圍

### 3.5 本章總結

SURF 演算法包含興趣點偵測器及描述子。興趣點偵測器是在多尺度空間下利用  $3 \times 3 \times 3$  非最大抑制法尋找局域性的極值，由於使用積分影像及近似的高斯二階偏導數濾波器，因此可以加快計算速度。SURF 描述子則是先找出可重現的主方位，再以此方位為基準，計算  $20s \times 20s$  方形區域內的 Haar 小波響應和以建立描述子向量。另外，在特徵匹配階段使用計算過程中得到的 Hessian 矩陣的跡之正負號，則可以先快速地判斷兩個特徵向量是否匹配。



## 第四章 嵌入式影像處理系統之應用

本章說明應用此次設計之嵌入式影像處理系統實作 SURF 演算法的相關事項，主要目的是驗證硬體電路功能是否能符合最初之設計要求。

### 4.1 SURF 演算法實作系統簡介

SURF 演算法實作主要分成三大部分：CMOS 攝影機、嵌入式影像處理系統及 PC 端之人機介面軟體，圖 4-1 是本次實作之系統方塊圖。FPGA 接收 CMOS 攝影機的影像輸出同時計算積分影像；DSP 執行 SURF 演算法；人機介面軟體負責顯示影像及 SURF 特徵描述子，另外可選擇是否使用最近鄰居演算法(Nearest Neighbors Algorithm)進行特徵匹配。嵌入式影像處理系統與 PC 之間使用高速 USB 連接，通訊方式可參考 4.7 節之 PC 與 DSP 的通訊命令說明。

#### 4.1.1 FPGA 內部架構

圖 4-2 所示為 FPGA 的內部架構。由於本次實作之目的主要是驗證系統電路功能，待得到實驗結果後可再進行演算法的最佳化，因此 FPGA 目前只負責 SURF 演法中積分影像之計算。其它功能則包含 CMOS 攝影機控制、影像流量調節、記憶體讀寫及 DSP 介面等。為了避免不同時脈領域之間資料傳輸發生錯誤，FPGA 與 DSP 及 CMOS 攝影機之間均使用雙時脈 FIFO 做為介面。另外，實作中使用 Altera 公司標準化的 Avalon 介面組成一個小型的 SOPC 系統，目的是在取得影像並且積分完成後，將兩者內容分別暫存在外部的 SSRAM，之後再將其讀出放入 DSP 介面的雙時脈 FIFO 中。

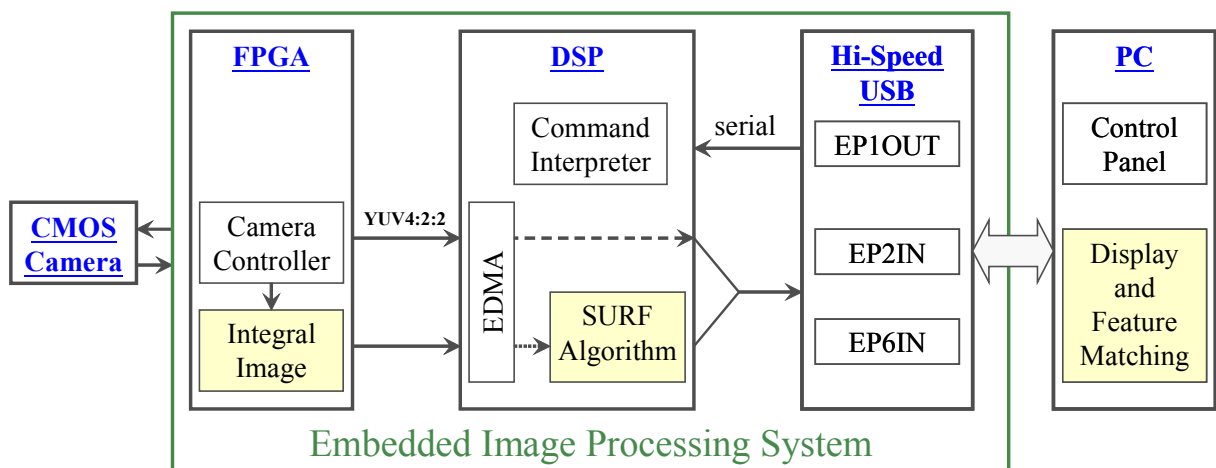


圖 4-1 SURF 演算法實作之系統方塊圖

以下說明 FPGA 從取像到寫入 DSP Bridge 的步驟：

- Camera Controller 根據 CMOS 攝影機的輸出時序，擷取其 YUV4:2:2 的輸出影像後放入雙時脈 FIFO。
- 當 Frame Grabber and Integral Image 模組偵測到雙時脈 FIFO 有資料後，便讀出並且計算積分影像，最後使用 Avalon 介面將原始影像及積分影像分別寫入外部的 SSRAM。
- Flow Control 模組負責調節影像流量，並決定 Image Uploader 是否可以讀取外部 SSRAM 的影像資料。Flow Control 與小型 SOPC 的動作相當於一個外部 FIFO。
- 當 Image Uploader 偵測到已有資料寫入 SSRAM 後，開始利用 Avalon 介面將資料讀出，然後分別寫入 DSP Bridge 的兩個雙時脈 FIFO。
- DSP Bridge 模組會計數寫入雙時脈 FIFO 的數量，每當填滿一列影像資料後就會送出一個訊號去觸發 DSP 的 EDMA 模組，讓其將整列資料讀回 DSP。

FPGA 內部除了上述的影像擷取與積分相關模組之外，為了減少 FPGA 資源的使用，將 DSP 的 Timer0 與 Timer1 當成一般 IO 連接到 CMOS 攝影機的 SCCB 介面，如此可以在 DSP 撰寫攝影機內部暫存器的讀寫函式。

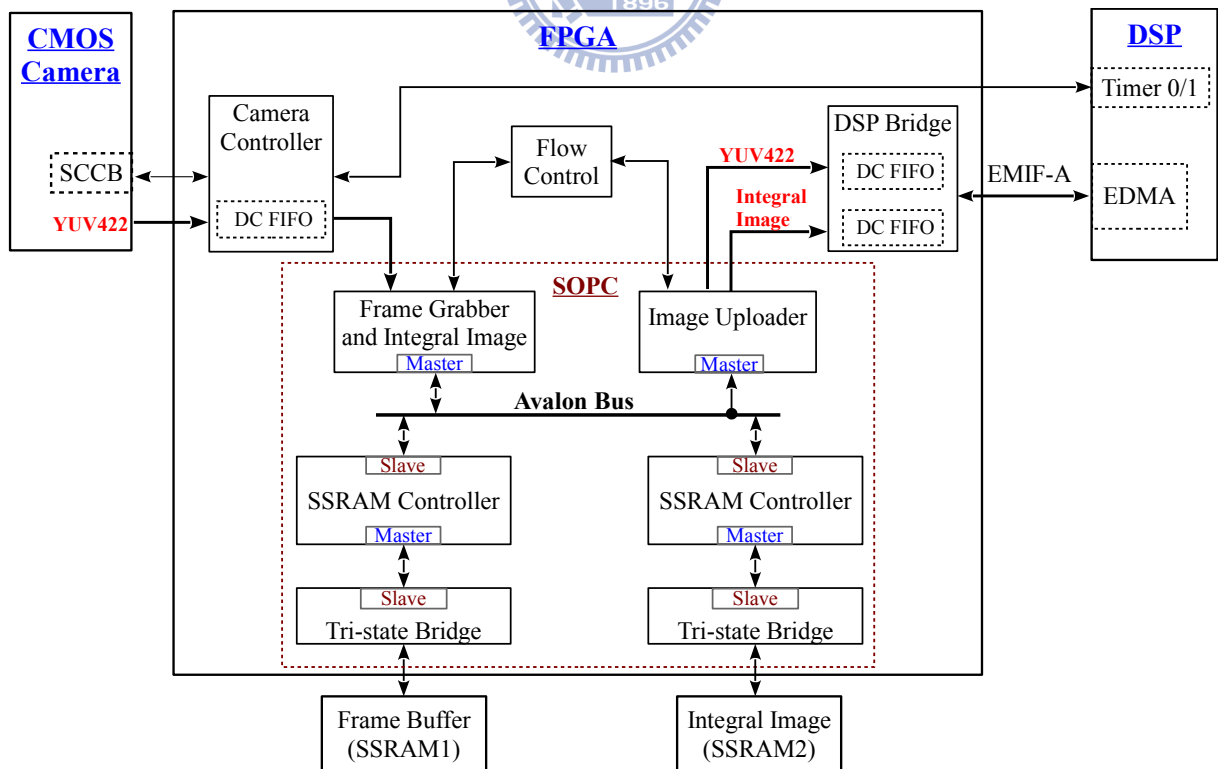


圖 4-2 FPGA 內部架構

#### 4.1.2 DSP 程式架構

DSP 程式以德州儀器公司的即時多工核心 DSP/BIOS 為基礎，可分成『命令解釋器』與『取像及 SURF 演算法實作』兩大部分。當人機介面軟體要傳送命令或燒錄 DSP 的程式碼時，會先傳到高速 USB 的 EP1OUT 緩衝區，此時高速 USB 內的 8051 微控制器會將內容轉成串列方式送到 DSP，然後 DSP 在中斷副程式接收完命令後送出一個信號(semaphore)給執行命令用的 Task。另外的『取像及 SURF 演算法實作』可以分成三個 Task 及一個 EMDA 中斷副程式，計算流程如圖 4-3 所示，說明如下：

- 開始取像或計算 SURF 描述子後，DSP 會先讓 FPGA 離開重置狀態，之後 FPGA 開始在每次接收完一列影像資料後觸發 DSP 之 EDMA 模組，使其將影像資料讀到 SDRAM，完成後再產生一個 EDMA 中斷。
- 在 EDMA 中斷裏，若是收到一列原始影像，則送出 semUpload 信號給 Uploader 將影像上傳到人機介面軟體；若是收到一列積分影像，則送出 semFindKeyPoints 信號給 Find Interest Points 開始尋找興趣點。
- 在 Find Interest Points 這個 Task，會先計算所有 Hessian 行列式的值，接著在 Octave 物件裏以 3x3x3 非最大抑制法尋找興趣點，找到後則以內插法求取空間及尺度下的次像素精度(Sub-pixel Accuracy)，最後再送出一個 semCalcFeatures 信號給 Calc. SURF Descriptors。
- 在 Calc. SURF Descriptors 裏則是計算興趣點的主方位及 SURF 描述子，然後將結果放入 FeatureQueue，再送出 semUpload 信號給 Uploader 將結果上傳到人機介面軟體。

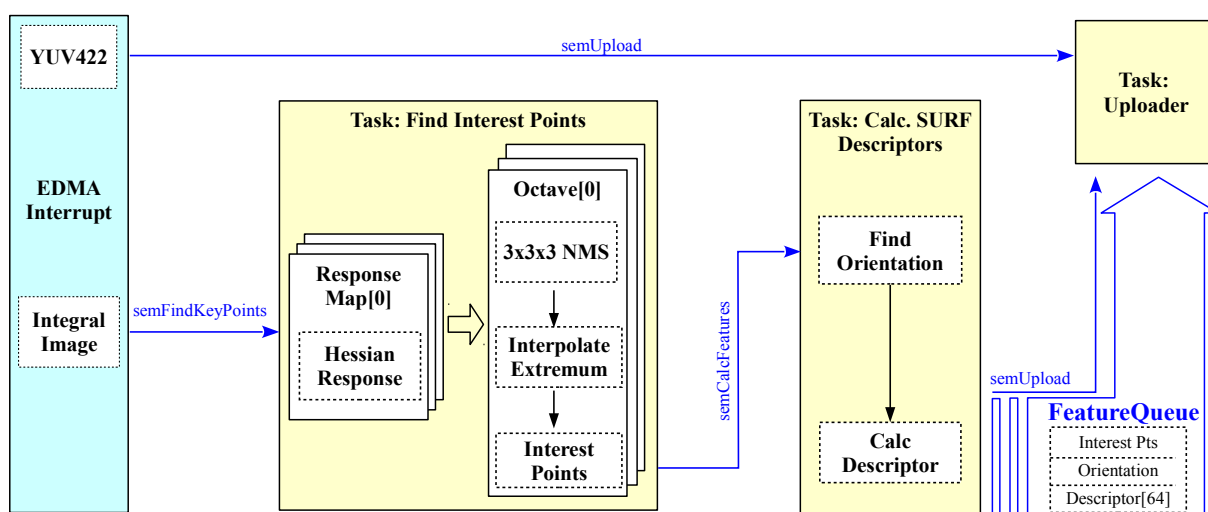


圖 4-3 DSP 程式之 SURF 描述子計算流程

### 4.1.3 人機介面軟體

此部分使用 C++ Builder 6.0 進行開發，取像及擷取特徵的操作畫面如圖 4-4 所示，顯示的架構如 4.8 節之說明。目前具備的功能有：

- 透過高速 USB 操作嵌入式影像處理系統，例如：取像、擷取特徵及燒錄等。
- 顯示及儲存影像畫面及 DSP 計算後的 SURF 描述子。
- 使用最近鄰居演算法進行動態 SURF 特徵匹配並顯示結果。
- 分析 DSP 專案轉換後的 Hex 檔，傳送到嵌入式影像處理系統進行燒錄。

### 4.2 SURF 演算法相關參數定義

考量到 USB 傳送資料的封包單位與 FPGA 內部的 FIFO 深度，實作時定義影像每一列有 512 個像素，每張影像畫面有 480 列。另外，如圖 4-5 所示，大部分的興趣點都可以在較低尺度的空間被偵測到，因此實作時僅選擇三個 Octave，每個 Octave 包含四張斑點響應圖 (Blob Response Map)。最後一個尺度值相當於 13.2，尺度值的計算可參考 (3-12) 式及圖 3-5。

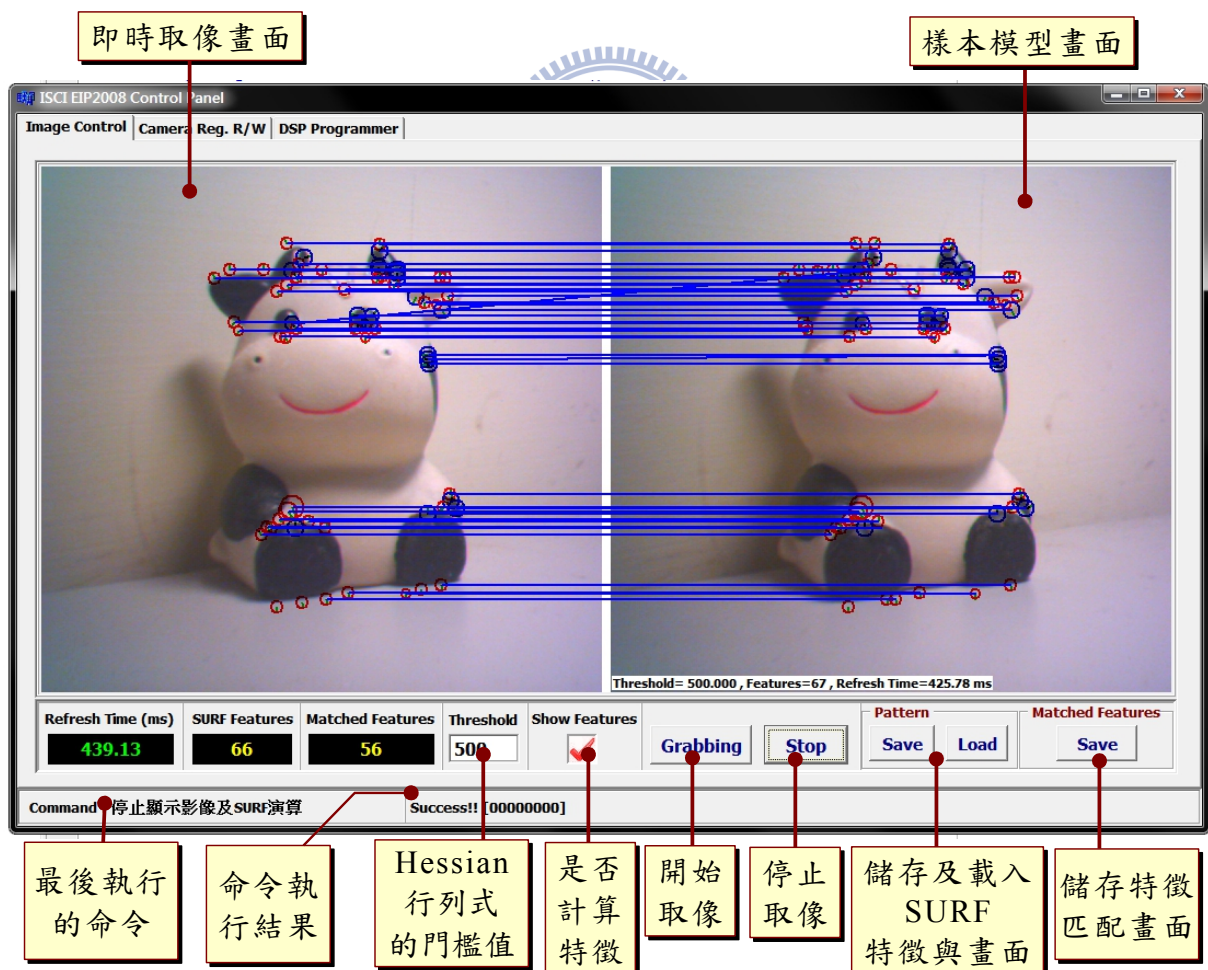


圖 4-4 人機介面軟體畫面

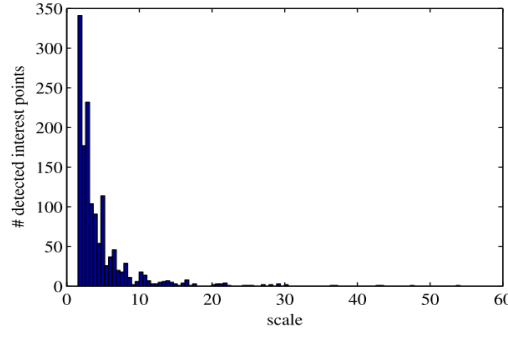


圖4-5 不同尺度下興趣點數量的直方圖[1]

#### 4.2.1 以 Octave 為基準的相關參數值

在同一個 Octave 之下以 3x3x3 非最大抑制法尋找興趣點時，箱形濾波器的中心點與取樣間距相同，尺度則依箱形濾波器的大小而不同。濾波器的邊長公式如 (3-11) 式，不同 Octave 取樣的間距為

$$Step = Step_{Initial} \times 2^{OctaveId} \quad (4-1)$$

公式中  $Step_{Initial}$  表示初始的取樣間距，本次實作程式設為 2； $OctaveId$  則是 Octave 的索引編號 0, 1, 2 ...。為了讓不同 Octave 間的取樣點重疊，使用取樣間距的最小公倍數  $Step_{LCM}$  來計算每個 Octave 在原始影像中的第一個取樣點座標：

$$Y_0 = X_0 = Step_{LCM} \times \left[ \text{INT} \left( \frac{L_{Filter, max} + 1}{2 \times Step_{LCM}} \right) + 1 \right] \quad (4-2)$$

最後一個取樣點座標則為：

$$Y_{Last} = Image\ Height - Y_0 \quad (4-3)$$

$$X_{Last} = Image\ Width - X_0 \quad (4-4)$$

由前面的公式可算出同一個 Octave 裏每個斑點響應圖的行數及列數：

$$MapRows = \frac{Y_{Last} - Y_0}{Step} + 1 = \frac{Image\ Height - 2 \times Y_0}{Step} + 1 \quad (4-5)$$

$$MapCols = \frac{X_{Last} - X_0}{Step} + 1 = \frac{Image\ Width - 2 \times X_0}{Step} + 1 \quad (4-6)$$

表 4-1 是當 Octave 數量等於三，斑點響應圖數量等於四的相關參數值。

#### 4.2.2 以箱形濾波器大小為基準的相關參數值

除了尺度會隨著箱形濾波器大小而變(參考(3-12)式)，要計算濾波器積分值的時候取樣點也不同。圖 4-6 是以參數的方式分別表示計算  $D_{xx}$ 、 $D_{xy}$  與  $D_{yy}$  時濾波器取樣點相對於中心點的偏移值，圖中的  $(x_c, y_c)$  表示濾波器的中心點， $X1 \sim X4$  與  $Y1 \sim Y4$  表示要計算影像積分值的時候使用之座標， $Lobe$  及  $Offset1 \sim Offset6$  則是取樣點相對於中心點  $(x_c, y_c)$  的偏移值。

表 4-1 以 Octave 為基準的相關參數值

OctaveId	0	1	2
濾波器最大邊長( $L_{Filter, max}$ )	27	51	99
取樣間距(像素)	2	4	8
第一個取樣點 ( $XY0$ )	16	32	56
垂直方向最後一個取樣點 ( $Y_{Last}$ )	464	448	424
水平方向最後一個取樣點 ( $X_{Last}$ )	496	480	456
每個斑點響應圖的總列數	225	105	47
每個斑點響應圖的總行數	241	113	51

取樣間距的最小公倍數

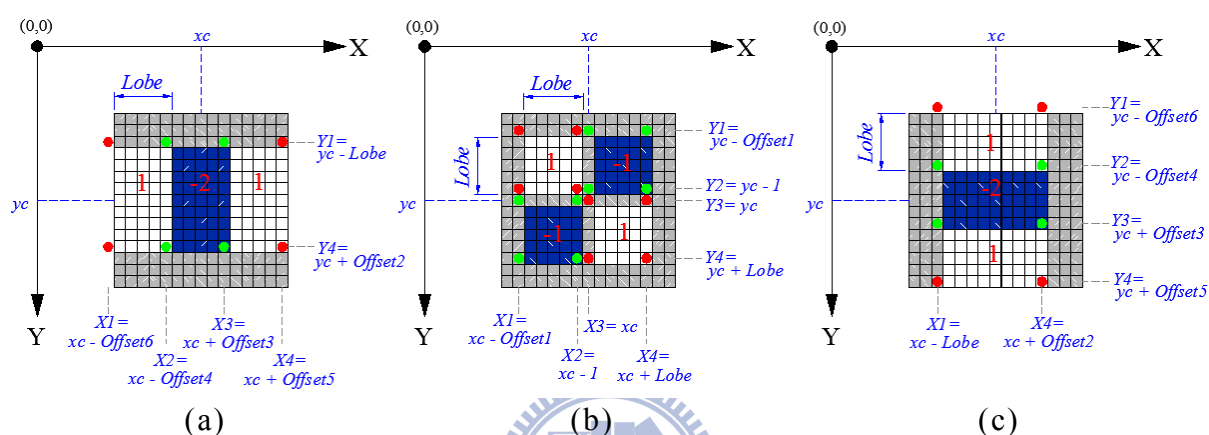


圖 4-6 箱形濾波器的取樣點。(a)x 方向 (b)xy 方向 (c)y 方向

圖 4-6 中，偏移值參數的計算公式如下列：

$$Lobe = INT\left(\frac{L_{Filter}}{3}\right) \quad (4-7)$$

$$Offset1 = Lobe + 1 \quad (4-8)$$

$$Offset2 = Lobe - 1 \quad (4-9)$$

$$Offset3 = INT\left(\frac{Lobe}{2}\right) \quad (4-10)$$

$$Offset4 = INT\left(\frac{Lobe + 1}{2}\right) \quad (4-11)$$

$$Offset5 = INT\left(\frac{L_{Filter}}{2}\right) \quad (4-12)$$

$$Offset6 = INT\left(\frac{L_{Filter} + 1}{2}\right) \quad (4-13)$$

表 4-2 則是當 Octave 數量等於三，斑點響應圖數量等於四的時候，利用公式(4-7)到(4-13)計算出與濾波器大小相關的參數值。

表 4-2 以箱形濾波器大小為基準的相關參數值

OctaveId	MapId	$L_{Filter}$	Scale	$\Delta Scale$	Lobe	Offset1	Offset2	Offset3	Offset4	Offset5	Offset6
0	0	9	1.2	0.8	3	4	2	1	2	4	5
0	1	15	2		5	6	4	2	3	7	8
0	2	21	2.8		7	8	6	3	4	10	11
0	3	27	3.6		9	10	8	4	5	13	14
1	0	15	2	1.6	5	6	4	2	3	7	8
1	1	27	3.6		9	10	8	4	5	13	14
1	2	39	5.2		13	14	12	6	7	19	20
1	3	51	6.8		17	18	16	8	9	25	26
2	0	27	3.6	3.2	9	10	8	4	5	13	14
2	1	51	6.8		17	18	16	8	9	25	26
2	2	75	10		25	26	24	12	13	37	38
2	3	99	13.2		33	34	32	16	17	49	50

### 4.3 Avalon 介面

Avalon 介面是 Altera 公司設計做為 FPGA 內部元件互連用的標準介面，具有以下特性：

- 提供標準化的元件介面時序規格，使各元件能夠獨立開發。
- 有圖形化的工具程式 Qsys(舊版為 SOPC Builder)能快速地建立基於 Avalon 介面的模組。
- Altera 或其它第三方廠商有提供免費或付費的元件以節省系統開發的時間，讓設計者能專心於其它功能的實現。
- 可使用現有的元件與外部零件連接，例如：SDRAM 控制器等。

Avalon 總共分成以下七種介面類型：

- **Avalon Streaming Interface(Avalon-ST)**：適用於單向的資料傳送，包含多路的封包或 DSP 資料傳輸。
- **Avalon Memory Mapped Interface(Avalon-MM)**：以位址為基準的主從式存取介面。
- **Avalon Conduit Interface**：不屬於 Avalon 介面的其它訊號，可用來連接 Avalon 模組與 FPGA 內部訊號或外部接腳。
- **Avalon Tri-State Conduit Interface(Avalon-TC)**：使用多工方式連接外部零件及 FPGA，可減少 FPGA 的接腳用量及 PCB 的線路。



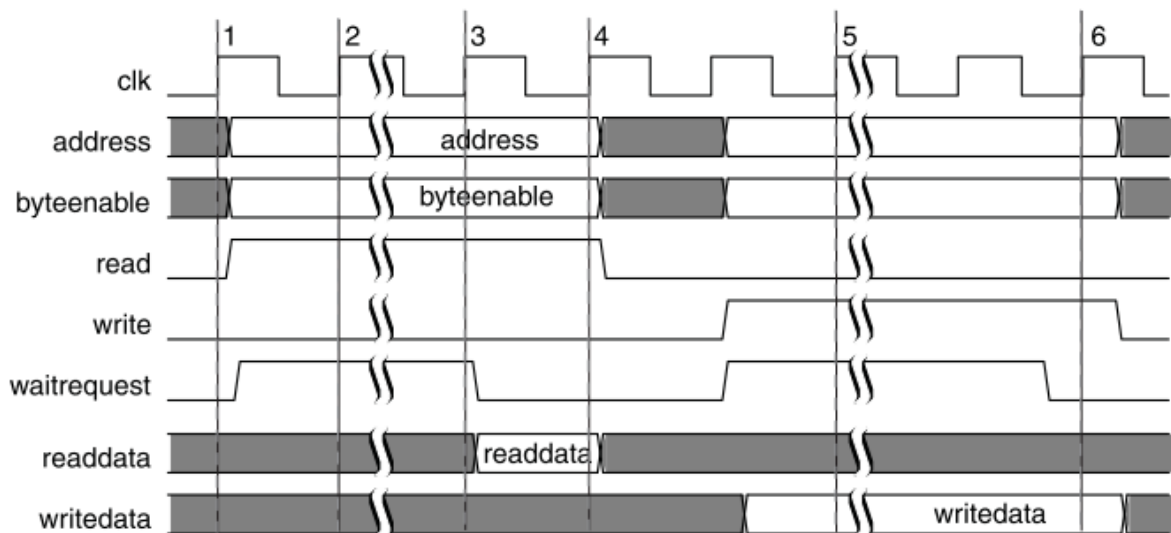


圖 4-8 配合 waitrequest 訊號的 Avalon-MM 讀寫時序[25]

#### 4.4 不同時脈領域之間的資料傳遞

當訊號在不同時脈領域之間傳遞時，對接收端而言就算是非同步訊號，如果接收端沒有進行訊號的同步則可能發生亞穩態(metastability)。所謂亞穩態是指暫存器無法在預定的時間達到確定的狀態，因此無法預期它的輸出電位，此狀態可能在接收端的時脈領域中一直傳遞下去而造成接收的資料錯誤。亞穩態的概念可以用圖 4-9 來表示，當訊號的轉換符合暫存器的設置時間  $t_{SU}$  與保持時間  $t_H$ ，則輸出會處於新的狀態，如圖 4-9(b)；若訊號的轉換發生於時脈的邊緣及保持時間之後，則輸出維持舊的狀態，如圖 4-9(a)；當訊號的轉換違反  $t_{SU}$  與  $t_H$ ，則輸出將如圖 4-9(c)處於亞穩態，此時將違反時脈至輸出的延遲時間  $t_{CO}$ 。

改善亞穩態的問題，可以從 FPGA 的製程著手，也可以藉由適當的設計來達成。對單一訊號而言，其中一種方式如圖 4-10 所示在接收端串聯多個同步用暫存器鏈，此時發送端與接收端的第一級暫存器之間不能包含組合邏輯，以避免組合邏輯產生的短暫脈衝雜訊(glitch)被接收端的第一級暫存器捕捉而在接收端的時脈領域中傳遞。在不同時脈領域傳送多位元資料的一種方式則是使用雙時脈 FIFO，除了可以匹配讀/寫端的流量之外，也可以混用不同的讀/寫寬度。雙時脈 FIFO 的設計仍然需要考慮訊號的同步機制，為了專注於整個系統功能的設計可以使用 Altera 公司提供的 DCFIFO 元件，使得寫入與讀取的動作都是相對於各別的時脈。不過在設定雙時脈 FIFO 的參數時仍需要指定同步用的時脈數量，越多階的同步時脈可得到較高的操作頻率，但時間的延遲也越大。

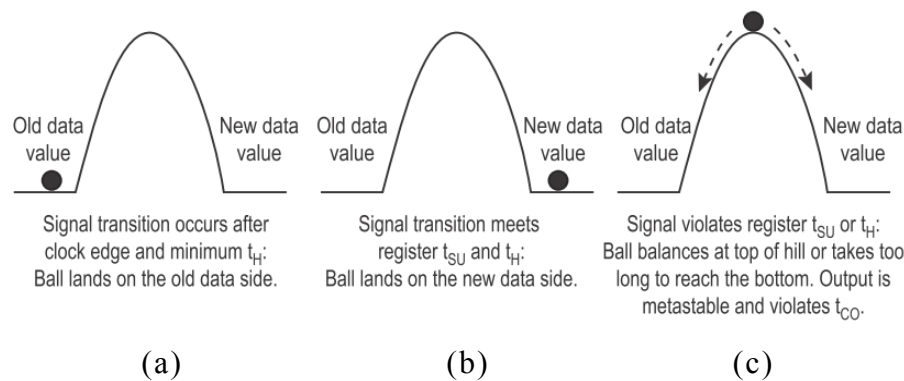


圖 4-9 亞穩態的圖解[26]。(a)訊號轉換發生在時脈之後，狀態不變 (b)訊號轉換符合  $t_{SU}$  與  $t_H$ ，改變至新狀態 (c)訊號違反  $t_{SU}$  或  $t_H$ ，輸出處於亞穩態

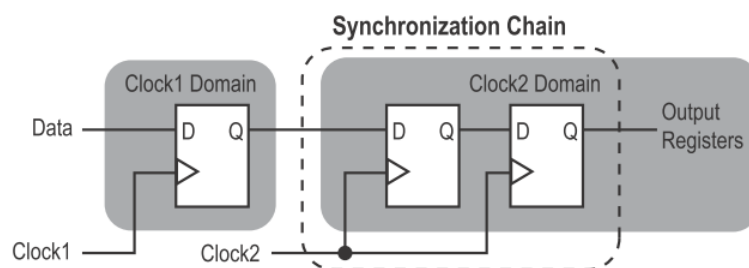


圖 4-10 簡單型的同步用暫存器鏈[26]

本次自行設計的 IP 元件有兩個地方需要考慮到不同時脈的資料傳遞。一處是 FPGA 讀取 CMOS 影像感測器的輸出值，另一處則是 DSP 透過 EMIF-A 讀取 FPGA 的影像資料時。圖 4-11 是使用雙時脈 FIFO 讀寫 CMOS 影像感測器輸出值的模擬波形，其中，CMOS 影像感測器的輸出寬度為 8 位元，而 FPGA 內部讀取時則使用 32 位元為單位。時序模擬時 CMOS 影像感測器內部使用 32 位元變數，每 4 個 PCLK 加 1，由於雙時脈 FIFO 使用 8 位元轉 32 位元的資料寬度，因此在 FIFO 讀取端的值也會是 32 位元累增的值。寫入端的時脈由外部 CMOS 影像感測器輸出的 PCLK 提供，控制訊號  $WrReq$  與  $WrData$  依 OV7649 的定義在 PCLK 的負緣變化，FIFO 的狀態  $WrFull$  則會在正緣變化；讀取端時脈由 FPGA 內部的 SysClk 提供，控制訊號  $RxReq$  在 SysClk 的正緣變化，FIFO 的狀態  $RxEmpty$  與輸出  $RxData$  則在負緣變化。

以下將圖 4-11 分成 7 個步驟說明：

- ① 在每次 PCLK 負緣時，CMOS 影像感測器將輸出 8 位元的  $WrData$ 。圖中顯示的是模擬變數的  $bit[7:0]$ ，數值為 12。
- ② 在 PCLK 的正緣時，因為  $WrReq=1$  而且  $WrFull=0$ ，所以  $WrData$  的值被寫入雙時脈 FIFO。此刻為模擬變數的  $bit[7:0]$ ，數值為 12。

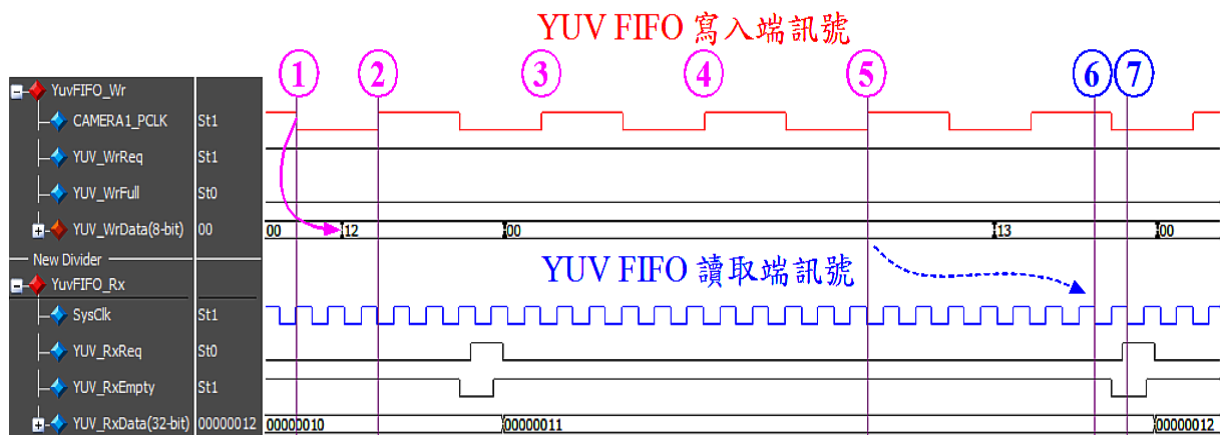


圖 4-11 雙時脈 FIFO 讀/寫 CMOS 影像感測器輸出值的模擬波形

- ③ 在 PCLK 的正緣時，因為 WrReq=1 而且 WrFull=0，所以 WrData 的值被寫入雙時脈 FIFO。此刻為模擬變數的 bit[15:8]，數值為 0。
- ④ 在 PCLK 的正緣時，因為 WrReq=1 而且 WrFull=0，所以 WrData 的值被寫入雙時脈 FIFO。此刻為模擬變數的 bit[23:16]，數值為 0。
- ⑤ 在 PCLK 的正緣時，因為 WrReq=1 而且 WrFull=0，所以 WrData 的值被寫入雙時脈 FIFO。此刻為模擬變數的 bit[31:24]，數值為 0，此位元組寫入後剛好是讀取端的一個字組(32 位元)。
- ⑥ 由於步驟 5 已寫完一個字組，經過雙時脈 FIFO 的同步化機制延遲後，FIFO 在 SysClk 的負緣輸出狀態 RxEmpty=0，表示有資料可以讀取。在下一個 SysClk 的正緣時因為 RxEmpty=0，因此令 RxReq=1，表示要求讀取 FIFO 的內容。
- ⑦ FIFO 在 SysClk 的負緣時偵測到 RxReq=1，因此將 CMOS 感測器最後一次寫入的字組數值 00000012(十六進位)輸出至 RxData。

#### 4.5 積分影像功能實作

CMOS 影像感測器的輸出順序為 8 位元的  $U_0Y_0V_0Y_1U_1Y_2V_1Y_3\cdots$ ，由於計算積分影像時只需要考慮光強度值  $Y_n$ ，而且 FPGA 內部是以 32 位元為讀取單位，因此相當於每 4 個影像感測器的輸出值要計算 2 次積分。考量到影像感測器以『串列』的方式依序輸出光強度值，為了使用 FPGA 達到即時計算之目的，假設在某一點  $\mathbf{x}=(a,b)$  時光強度值為  $I(a,b)$ ，則原來的積分影像公式修改如下：

$$\begin{aligned}
H(\mathbf{x}) &= \sum_{i=0}^{i \leq a} \sum_{j=0}^{j \leq b} I(i, j) \\
&= \sum_{i=0}^{i \leq a} \sum_{j=0}^{j \leq (b-1)} I(i, j) + \sum_{k=0}^{k \leq (a-1)} I(k, b) + I(a, b) \\
&= H_p + \text{RowSum} + I(a, b)
\end{aligned} \tag{4-14}$$

上式中  $H_p$  表示由零點至點  $(a, b-1)$  的矩形積分值， $\text{RowSum}$  則是當前影像感測器的輸出列  $b$  從 0 到第  $a-1$  行的累加值，圖 4-12 為此公式的示意圖。

使用 FPGA 實作積分影像的計算時，使用如圖 4-13 右側的移位暫存器來儲存每次計算後的積分值，移位暫存器的總長度等於影像的寬度，當影像感測器輸出 VSYNC 訊號時表示每張影像的起點，可用此訊號來清除移位暫存器，最後一級的輸出則相當於上面公式中的  $H_p$ 。圖 4-13 左側的暫存器則用來儲存每列光強度的累加值，其輸出相當於公式中的  $\text{RowSum} + I(a, b)$ ，當影像感測器的 HREF 訊號為低電位時(見圖 2-6)，表示每列輸出的間隔時間，用來重置此暫存器的輸出。積分影像則是上面兩者的相加值。

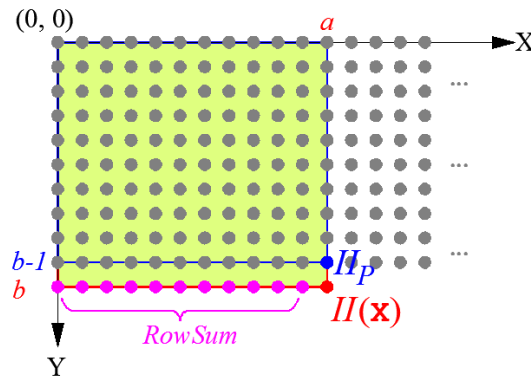


圖 4-12 即時計算影像感測器輸出點  $(a, b)$  的積分影像示意圖

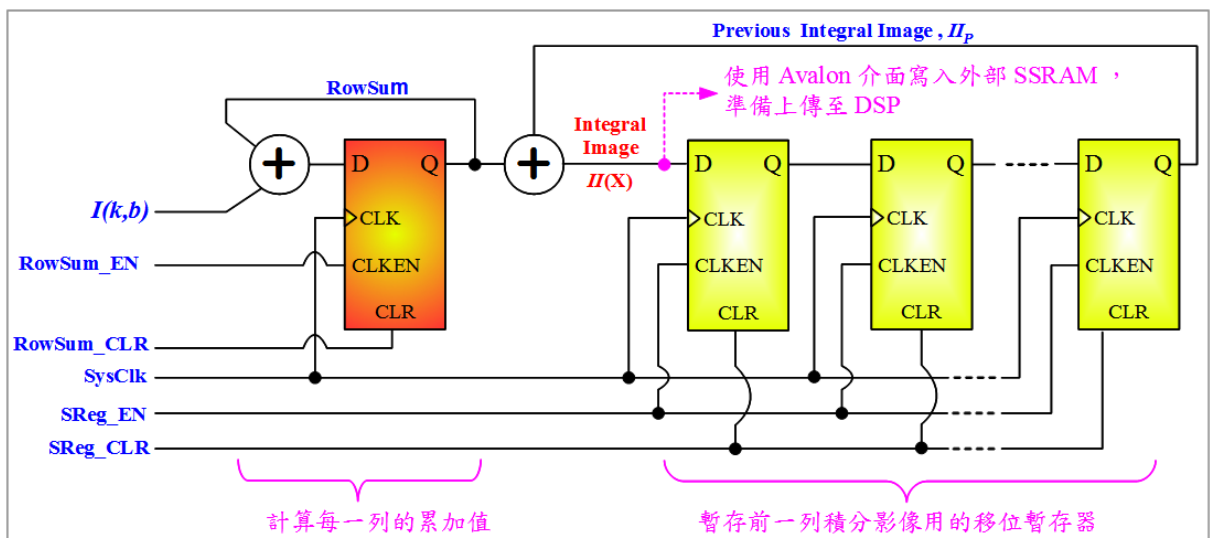


圖 4-13 FPGA 計算積分影像的電路結構

## 4.6 如何建立 DSP 的 Boot ROM

DSP 程式在開發階段通常使用 JTAG 模擬器進程式碼下載與除錯，實際應用時則需要 DSP 在每次送電後都能開始獨立運作，因此必須將開發完成的程式碼燒錄到電路板上的 FLASH ROM。要達到 ROM 開機之目的，首先要撰寫二次開機載入程式(Secondary Bootloader)並將它加入原本的 DSP 專案中，程式開發完成後再使用德州儀器公司提供的 hex6x 工具程式將專案編譯後產生的 COFF 檔(.out)轉換成燒錄用的 Hex 文字檔，最後則是將 Hex 檔的內容燒錄到外部的 FLASH ROM。

### 4.6.1 C6414T 的 ROM 開機程序

德州儀器公司有多種的 DSP 產品線，而且不同型號的 DSP 其 ROM 開機程序也不盡相同。以 C6414T 為例，在上電重置後只能使用 8-bit ROM 開機模式，從 EMIF-B 的 CE1 複製 1K Bytes 的資料到內部記憶體的位置 0，由於實際應用上 DSP 程式必定會超過此大小，故此部分通常只儲存另外撰寫的二次開機載入程式(Secondary Bootloader)，負責把其它程式區段複製到執行程式的記憶體，完成後便跳到專案程式的進入點開始執行，如圖 4-14 所示。

### 4.6.2 產生燒錄用的 Hex 檔

為了產生燒錄用的 Hex 檔，要做一些對應的設定及增加 2 至 3 個檔案到專案中，圖 4-15 是 DSP 開發環境 CCStudio 的專案列表與 DSP/BIOS 配置的部分畫面，以下將其分成三個部分說明。

- ① C6414T 的內部記憶體有 1M Bytes，因為這次的 DSP 專案中有使用到德州儀器公司提供的即時作業系統 DSP/BIOS，因此配置記憶體時要將此區域分成二部分：

- ◆ **BOOT\_ISRAM**：儲存 ROM 開機時硬體自動由 EMIF-B 的 CE1 空間複製來的 1K(0x00000400)Bytes 資料，起始位址是 0x00000000。此區的 create a heap in this memory 選項必須取消，同時 space 要設為 code/data。
- ◆ **ISRAM**：起始位址緊接於 BOOT\_ISRAM 之後(0x00000400)，長度則是 1M - 1K = 1047552(0x000FFC00)。由於本專案使用此區做為程式與主要的資料記憶體，因此 space 一樣設為 code/data，同時要勾選 create a heap in this memory，給程式中動態配置記憶體時使用。

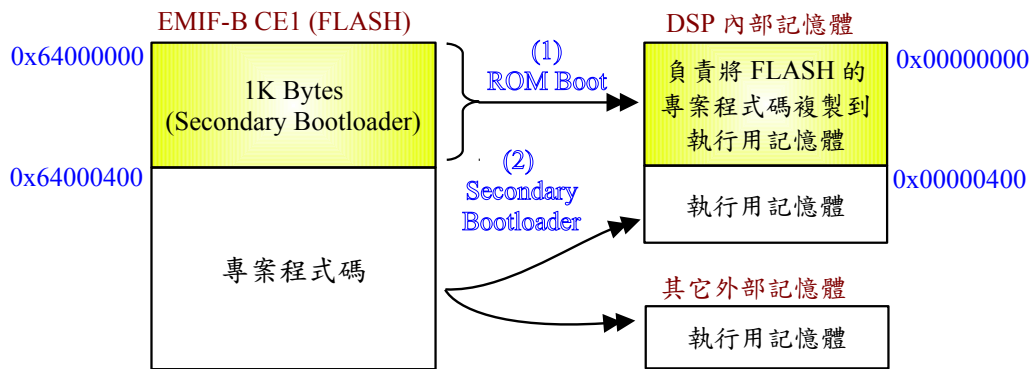


圖 4-14 C6414T 的 ROM 開機程序

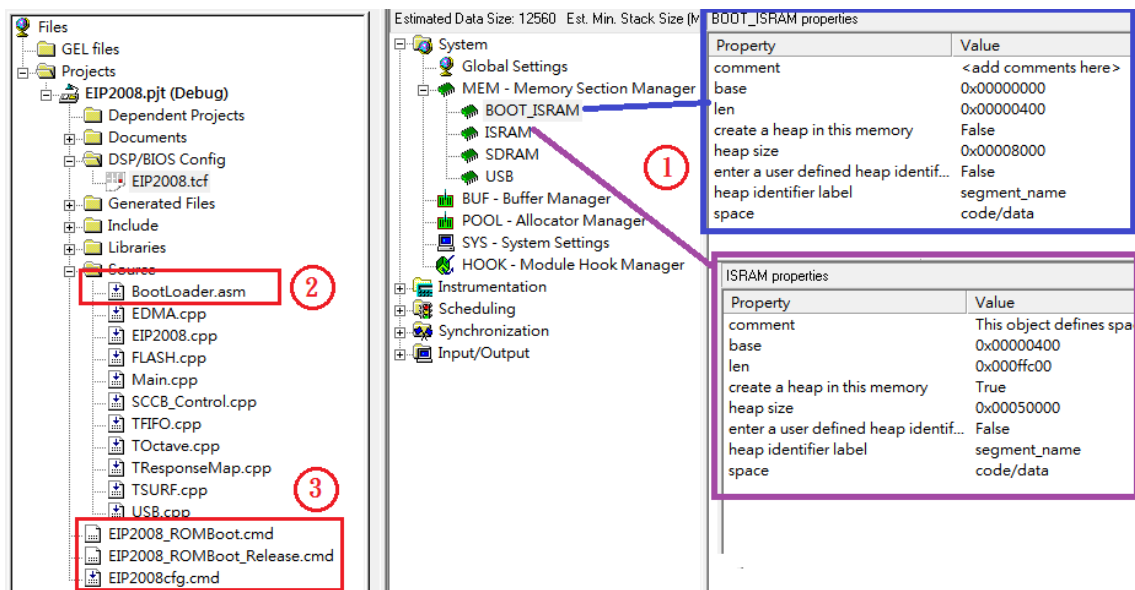


圖 4-15 產生燒錄用的 Hex 檔所需之相關設定與檔案

- ② BootLoader.asm 即前述的二次開機載入程式，在 ROM 開機時包含於前述的 1K Bytes 裏被複製到 BOOT\_ISRAM 之後開始執行。此部分使用[20]書附光碟第六章的範例檔 boot.asm，負責將 DSP 專案的其它節區由 FLASH ROM 複製到 ISRAM。
- ③ 連結命令檔，主要用來定義程式節區、記憶體配置與連結命令等。其中的 EIP2008cfg.cmd 是由 DSP/BIOS 配置程式自動產生，另外二個 cmd 檔則是當作工具程式 hex6x 的輸入檔，目的是將專案編譯後的 COFF 檔(.out)轉成要燒錄的.hex 檔，這二個檔案內容幾乎相同，只是裏面指定的.out 檔分別為 Debug 版或 Release 版。因為 DSP 專案中只能有一個 cmd 檔，因此要將這二個 cmd 檔在編譯時排除，作法是在檔名處按滑鼠右鍵選擇 File Specific Options...，然後將 Exclude file from build 選項打勾。圖 4-16 是使用 Debug 版的.out 檔轉換成 Hex 燒錄檔所需的 cmd 檔內容。

```

/*****
1. 設定 C6x Compile 後的 *.out 輸出檔
2. 指定 hex6x 要輸出的 *.hex 檔 (-o) 及格式 (-a)
3. 指定 hex6x 要輸出的 *.map 檔
*****/
".\Output\Debug\EIP2008.out"

-a -o .\Output\EIP2008.hex
-map .\Output\EIP2008_hex.map

/*****
設定 FLASH ROM 相關的資訊，並使用 -boot 相關的選項
其中的 .boot_load 要與 BootLoader.asm 裏設定的 .sect 相同
*****/
-memwidth 8
-boot
-bootorg 0x64000400
-bootsection .boot_load 0x64000000

```

圖 4-16 將 out 檔轉換成 Hex 燒錄檔所需的 cmd 檔內容

上列三個部分只需設定或撰寫一次，以後只要燒錄前在 DOS 命令列執行下列命令，就可以將最終編譯後產生的.out 檔轉換成 Hex 燒錄檔：

除錯版程式碼轉換：*hex6x Source\EIP2008\_ROMBoot.cmd*

發行版程式碼轉換：*hex6x Source\EIP2008\_ROMBoot\_Release.cmd*

如果擔心燒錄前忘了轉換成最新的 Hex 檔，也可以使用 Project→Build Options... 叫出圖 4-17 的畫面，將上述命令依程式碼的版本擇一加入 Final build steps。如此一來，每次編譯專案後就會自動將.out 檔轉換成 Hex 燒錄檔。

#### 4.6.3 ROM 的燒錄程式

由於德州儀器公司或第三方配合廠商提供的燒錄程式都是配合其開發的電路板，為了將本專案的 DSP 程式燒入自行設計的嵌入式影像處理系統，就必須撰寫 FLASH ROM 的燒錄程式。燒錄程式分成 Hex 檔解析與燒錄函式二部分：

##### a) Hex 檔解析

Hex 檔解析的工作由 PC 端的人機介面軟體負責。Hex 檔屬於一種『文字檔』，包含多種格式，例如：Intel Hex、ASCII-Hex、Motorola-S 等等。目前我們使用 hex6x 的版本是 v6.0.8，當如圖 4-16 的 cmd 檔內設定輸出格式使用 -a 選項時指定的即是 ASCII-Hex 格式。如圖 4-18 所示，當程式碼的位址不連續時，就會有一列以 \$A 開頭的 32 位元位址，接下來每一列的內容都是位址連續的 8 位元

資料，每列的資料個數可能不同但仍然是連續位址，除非再次遇到\$A開頭的位址。Hex檔解析之目的就是把以字元表示的十六進制位址與資料轉換成二進制的數值，然後由USB傳送到DSP進行FLASH ROM的燒錄。

## b) 燒錄函式

燒錄函式位於DSP程式中，負責接收PC端解析Hex檔之後傳來的資料，然後對FLASH ROM進行燒錄與比對。FLASH ROM連接於EMIF-B匯流排上，而程式裏的邏輯位址與EMIF-B實體位址之間的對應關係會因為所連接的記憶體資料匯流排寬度而不同。因為C6414T在ROM開機階段固定使用8位元的非同步記憶體時序，所以兩種位址的對應關係如表4-3所示，再參考圖2-8 EMIF-B與FLASH ROM的位址連接方式，可得到邏輯位址與FLASH ROM位址剛好對齊於第0個位元，即程式中的位址值可以直接對應到FLASH ROM的位址。

FLASH ROM的讀寫操作是以命令方式進行，而命令位址與電路中設定的資料匯流排寬度有關，雖然DSP在ROM開機時使用的是8位元模式，但電路設計時FLASH ROM是定義為16位元模式，因此對FLASH ROM讀寫時要使用它的Word命令。完整的命令定義可參考[17]的表14，表4-4僅列出目前燒錄時有使用到的命令。

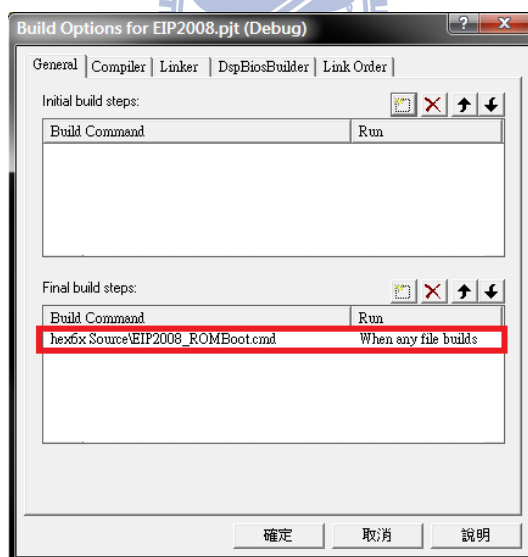


圖 4-17 設定編譯後自動產生 Hex 燒錄檔

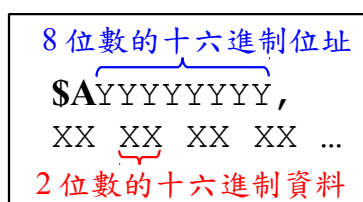


圖 4-18 ASCII-Hex 格式

表 4-3 使用非同步憶記體時邏輯位址與 EMIF 位址的對應關係[28]

		EA Line																			
C620x/C621x/ C670x/C671x	EA	EA	EA	EA	EA	EA	EA	EA	EA	EA	EA	EA	EA	EA	EA	EA	EA	EA	EA	EA	
	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	
C64x EMIFA	EA	EA	EA	EA	EA	EA	EA	EA	EA	EA	EA	EA	EA	EA	EA	EA	EA	EA	EA	EA	
	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	
C64x EMIFB	EA	EA	EA	EA	EA	EA	EA	EA	EA	EA	EA	EA	EA	EA	EA	EA	EA	EA	EA	EA	
	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	
Mtype Width		Logical Byte Address																			
x64	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	
x32	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	
x16	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	
x8	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	

表 4-4 FLASH ROM(Am29LV320DB)燒錄命令摘要

命令	操作週期數量	週 期											
		1		2		3		4		5		6	
		Addr	Data	Addr	Data	Addr	Data	Addr	Data	Addr	Data	Addr	Data
讀取	1	RA	RD										
燒錄	4	555	AA	2AA	55	555	A0	PA	PD				
區段抹除	6	555	AA	2AA	55	555	80	555	AA	2AA	55	SA	30

表 4-4 中，Addr 與 Data 欄位的數值均為十六進制。RA：要讀取的位址、RD：讀取的內容、PA：要燒錄的位址、PD：要燒錄的內容、SA：區段的位址，由 FLASH ROM 的位址線 A[20:12]指定。Am29LV320DB 屬於 Bottom Boot Block 型式的 FLASH，詳細的 SA 與區段編號或位址範圍的關係請參考[17]的表 4。

#### 4.7 PC 與 DSP 的通訊命令說明

PC 與 DSP 之間使用高速 USB 裝置 CY7C68013A 的 Slave FIFO 及 GPIO 進行通訊。USB 使用端點(Endpoint)做為資料傳輸的通道，IN/OUT 方向是以 PC 的觀點定義，例如 EP1OUT 表示由 PC 送出資料。當 PC 要送命令給 DSP 時，會先送到 USB 的 EP1OUT 端點緩衝區，然後由 USB 裝置內部 8051 微控制器將命令封包以串列形式送到 DSP；DSP 則以中斷的方式，逐位元地將命令讀入後執行。表 4-5 是 PC 送給 DSP 的命令封包內容。其中 x=0 或 EP2 AutoIn Words=0 時，長度等於 256Words 或 512 Bytes。另外，讀寫 DSP 的 FLASH ROM 時，N 的最大值是依 hex6x 工具程式轉換後的 Hex 檔內容決定，目前 N=24。

當 DSP 接收到命令並且執行完後，會由 EP6IN 回應命令執行結果；如果是

連續取像同時進行 SURF 演算法，則 EP2IN 傳回影像畫面並且由 EP6IN 傳回 SURF 特徵描述子。EP2IN 與 EP6IN 規劃為同步式 Slave FIFO 模式，它的資料匯流排寬度設定為 16 位元寬，而 PC 端的 API 函式則是以 byte 型式傳回收到的資料，傳送的示意圖與資料順序如圖 4-19 所示。表 4-6 則是 DSP 由 EP6IN 回應給 PC 的封包內容。在進行 DSP 程式燒錄時，由於 EP6IN 回傳到 PC 是以 16 位元為單位，所以若表中的 N 為奇數，最後一筆回傳的 Code 陣列必須補齊為 16 位元，格式為 {0x00, Code[N-1]}。如果 Code 陣列可正常寫入 FLASH，則最後一個字組的高位元組為字元 T、低位組為 0；若燒錄過程中有錯誤，則高位元組為 0、低位元組為錯誤的 Code 陣列的索引值。

表 4-5 PC 送到 DSP 的命令封包內容

	Host -> USB EP1OUT -> Serial Out -> DSP						
	Tx Packet[]						
	CMD ID	EP6 AutoIn Words	Data Field				
Packet Index →	0	1	2	3	4	5	6 ... N+5
設定 Slave FIFO 的IFCONFIG暫存器 (主要是IFCLK的來源)	0x10	1	IFCONFIG	EP2 AutoIn Words			
設定 CMOS Camera的暫存器	0x11	1	Addr	WrData			
讀取 CMOS Camera的暫存器	0x12	1	Addr				
抹除 DSP 外掛FLASH 的sectors - Sector Addr[20:12] = SA<8? SA : (SA-7)*8 - 每完成一個Sector就回應	0x20	1	起始的 Sector Num	Length			
燒錄 DSP 程式到外掛的FLASH ROM - 燒錄前要先抹除有用到的Sectors - 預設A[20]=GP0=0	0x21	$\text{INT}(\frac{N+1}{2})+3$	Flash Addr [7:0]	Flash Addr [15:8]	Flash Addr [23:9]	Code Length (N)	Code[0] : Code[N-1]
讀取 DSP 外掛的FLASH ROM內容	0x22	$\text{INT}(\frac{N+1}{2})+2$	Flash Addr [7:0]	Flash Addr [15:8]	Flash Addr [23:9]	Code Length (N)	
停止取像 - 同時 Reset FPGA (不含Camera)	0x40	1					
連續取像 + SURF演算 - 傳回Feature的Descriptio結構	0x41	X	Threshold Byte[0]	Threshold Byte[1]	Threshold Byte[2]	Threshold Byte[3]	Get Feature?

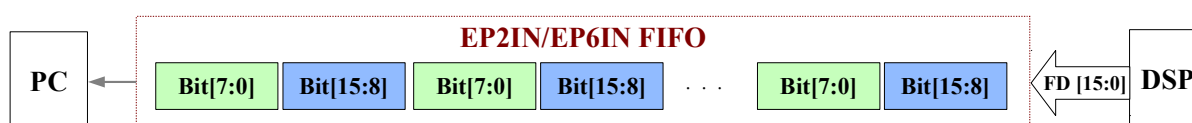


圖 4-19 EP2IN 及 EP6IN 傳送 16 位元資料寬的位元組順序

表 4-6 DSP 回應 PC 的封包內容

Host ←- USB EP2IN/EP6IN ←- Slave FIFO ←- DSP							
EP6IN : Rx/Response Packet[]							
		Data Field					
Packet Index →		0	1	2	3	$4 \dots \text{INT}(\frac{N+1}{2}) \times 2 + 3$	$4 \dots \text{INT}(\frac{N+1}{2}) \times 2 + 4$
設定 Slave FIFO 的IFCONFIG暫存器 (主要是IFCLK的來源)		IFCONFIG	EP2 AutoIn Words				
設定 CMOS Camera的暫存器		Addr	WrData				
讀取 CMOS Camera的暫存器		Addr	RdData				
抹除 DSP 外掛FLASH 的sectors - Sector Addr[20:12] = SA<8? SA : (SA-7)*8 - 每完成一個Sector就回應		已處理的 Sector Num	'T'=OK				
燒錄 DSP 程式到外掛的FLASH ROM - 燒錄前要先抹除有用到的Sectors - 預設A[20]=GP0=0		Flash Addr [7:0]	Flash Addr [15:8]	Flash Addr [23:9]	Code Length (N)	Code[0] : Code[N-1]	Error Code Index
讀取 DSP 外掛的FLASH ROM內容		Flash Addr [7:0]	Flash Addr [15:8]	Flash Addr [23:9]	Code Length (N)	Code[0] : Code[N-1]	
停止取像 - 同時 Reset FPGA (不含Camera)		7	1				
連續取像 + SURF演算 - 傳回Feature的Descriptio結構		Descriptor結構 ...					

#### 4.8 PC 端人機介面的顯示架構

PC 端的畫面顯示主要使用 C++ Builder 的 TBitmap 物件，先在背景畫好影像及特徵畫面(PaintBoxBuf)後再顯示到螢幕中。顯示架構如圖 4-20 所示，使用者啟動影像或 SURF 特徵擷取及匹配後，顯示的步驟如下：

- 當高速 USB 接收到 DSP 傳來的 YUV4:2:2 原始影像後先畫在 Frame 物件；若是收到 SURF 描述子則畫在 Feature 物件。如果要儲存 SURF 特徵，要先停止取像再將最後的 Frame 及 Feature 內容存入檔案中。另外，當每次接收完一個畫面後才進行後續 b 至 d 的步驟。
- 先如圖中的步驟①將 Pattern 畫到 PaintBoxBuf，若 Pattern 從未由檔案中載入，表示僅單純取像或擷取 SURF 特徵，則 Pattern 的左右兩邊均為空白；若要進行匹配而將樣本模型檔載入 Pattern 中，則 Pattern 的左側維持空白，右側則是待匹配的樣本模型畫面。此步驟相當於清除 PaintBoxBuf 中前一個即時擷取之影像畫面及特徵匹配結果。
- 如圖中的步驟②將 Frame 畫到 PaintBoxBuf，若要擷取或匹配特徵再如步驟③將 Feature 也畫到 PaintBoxBuf。
- 如果要匹配特徵，會先將匹配結果畫在 PaintBoxBuf，最後再如步驟④將 PaintBoxBuf 顯示在螢幕中。

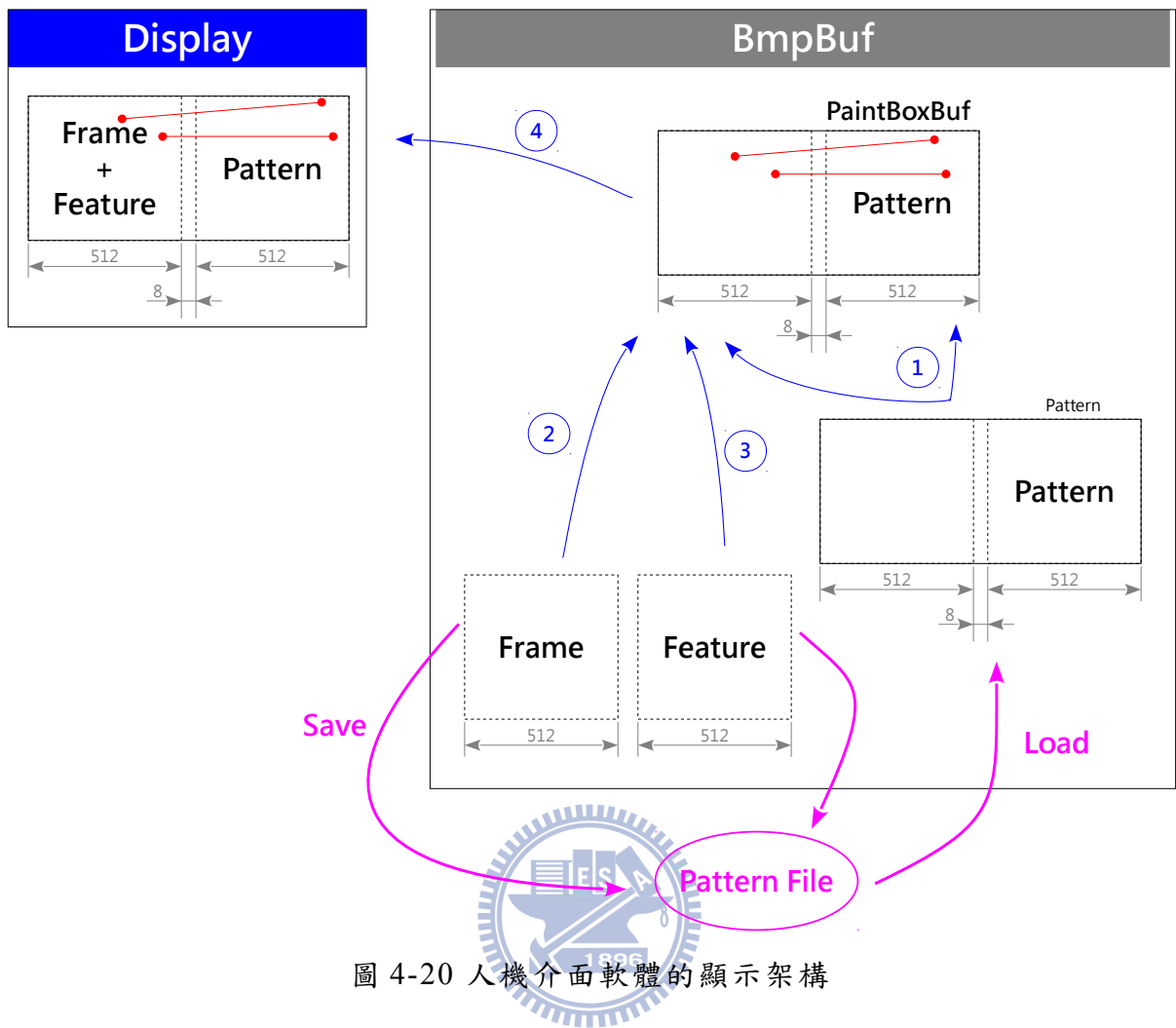


圖 4-20 人機介面軟體的顯示架構

## 第五章 實驗與討論

本章使用此次設計之嵌入式影像處理系統實作 SURF 演算法，主要目的是驗證硬體電路功能是否符合預期。圖 5-1 為本次的實驗環境，電腦主機為 Intel Core i7-2600 @3.4GHz、16GB RAM，作業系統使用 Win7 x64，人機介面軟體使用 C++ Builder 6.0 設計，CMOS 影像感測器使用 OmniVision 的 OV7649，FPGA 擷取的影像大小固定為 512x480 像素。本章各節將分別實驗即時影像擷取、動態擷取 SURF 特徵與動態匹配 SURF 特徵，最後則是本系統與 OpenSURF[9]擷取之特徵比較。

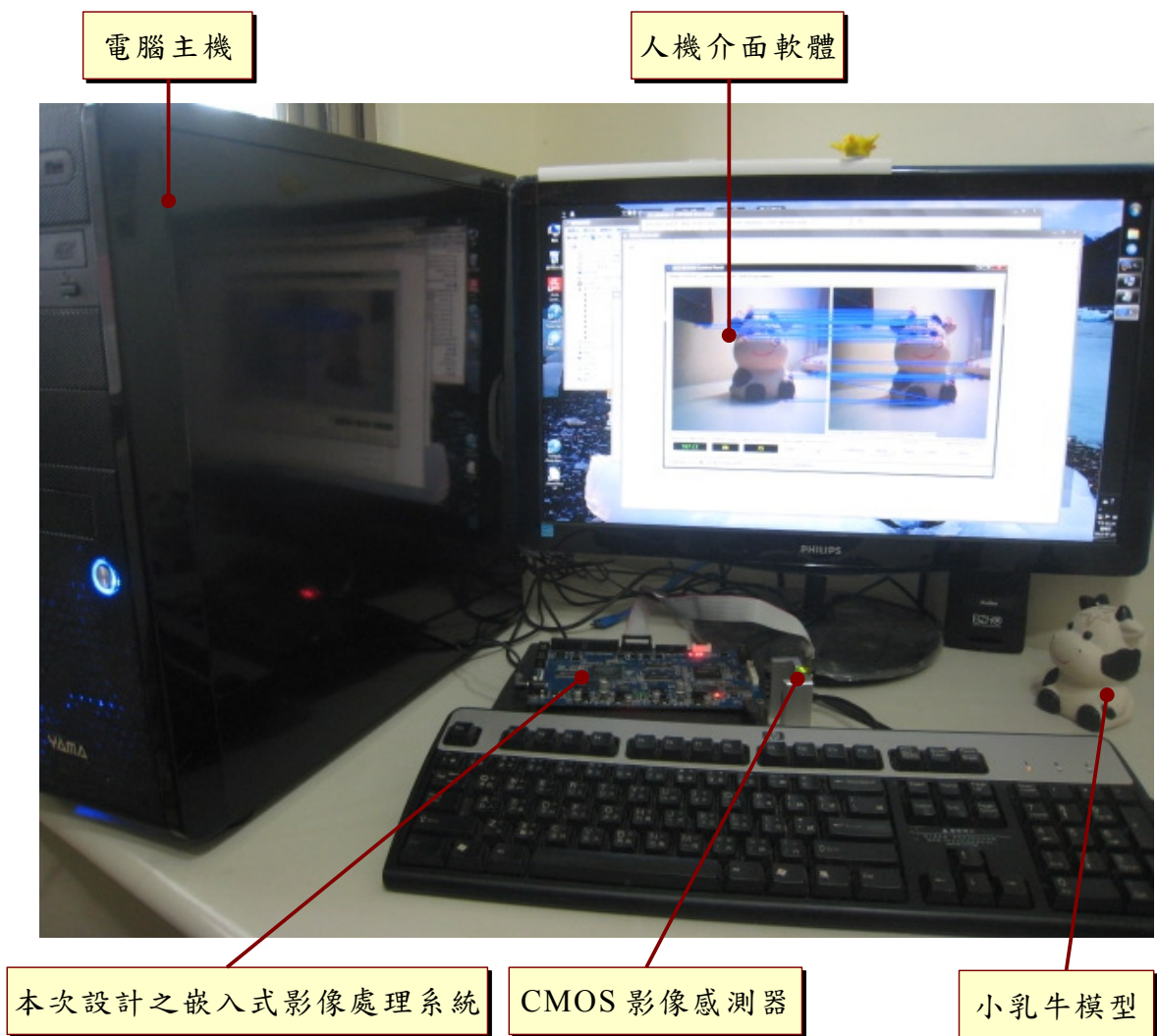


圖 5-1 本次的實驗環境

## 5.1 即時影像擷取

系統實作之功能方塊圖如圖 4-1 所示。即時影像之擷取流程如下：

- 以 FPGA 即時地擷取 CMOS 感測器的輸出值，同時進行影像積分後分別存入外部的 SSRAM 緩衝區。
- 將 SSRAM 緩衝區的原始及積分影像依序讀入 FPGA 內部的雙時脈 FIFO，每一列完成後則送出一個訊號觸發 DSP 之 EDMA。
- EDMA 將原始及積分影像整列讀回 DSP。原始影像會直接經由高速 USB 介面傳回 PC。如果有指定計算 SURF 描述子，則計算後也會立即傳回 PC。
- 人機介面軟體接收完一個畫面的影像及 SURF 描述子後立即顯示於螢幕。如果要儲存影像畫面及 SURF 描述子，必須先暫停取像再將最後一張畫面儲存成 bmp 格式檔；若取像時有指定計算 SURF 描述子，軟體會同時將描述子儲存成 ini 格式檔。ini 檔中，每個節區(section)表示一個描述子，內容包含興趣點被偵測到的  $x$ 、 $y$  座標、尺度及描述子的主方位與 64 維之特徵向量。

一旦即時影像能正常顯示，表示嵌入式影像處理系統的主要元件如：DSP、FPGA、記憶體、資料傳輸路徑及 CMOS 影像感測器等電路可以正常動作。圖 5-2 是使用示波器量測 CMOS 影像感測器的 VSYNC 與 HREF 輸出波形。在感測器輸入時脈等於 24MHz 的時候，示波器的 VSYNC 周期與圖 2-6 相近，所以此次設計之 CMOS 影像感測器電路確實可正常動作。圖 5-3 所示的三個場景則是在人機介面軟體即時顯示時擷取下來存檔之畫面，也是後續要進行動態 SURF 特徵擷取與匹配要使用的畫面。圖中有顯示存檔當時的畫面更新周期，三張圖平均約為 33.5ms 也與圖 2-6 相近，故可確認此次設計之嵌入式影像處理系統的硬體電路亦可符合預期動作。

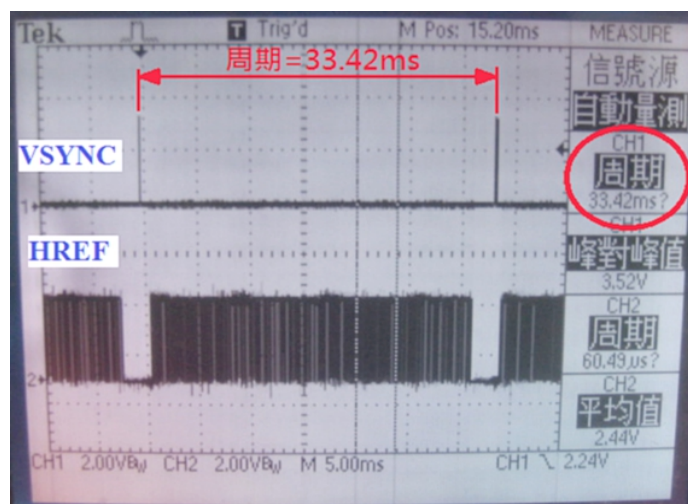


圖 5-2 CMOS 影像感測器的 VSYNC 與 HREF 輸出波形



(a)



(b)



(c)

圖 5-3 嵌入式影像處理系統即時擷取之畫面 (a)小乳牛 (b)混合物件 (c)書本

## 5.2 動態 SURF 特徵擷取實驗

在動態 SURF 特徵擷取的實驗中，將使用前一節的三種物件畫面個別進行 SURF 特徵的擷取實驗，分別為小乳牛(亮背景)、小乳牛(暗背景)、混合物件及書本四種，其中同一個小乳牛會在兩種不同的光線背景中進行擷取。在完成電源、USB 及 CMOS 影像感測器的連接後，即可由人機介面操作特徵擷取的實驗，圖 5-4 即為特徵擷取時軟體的操作流程圖。

擷取 SURF 特徵描述子之前，必須使用如圖 3-6 所示之  $3 \times 3 \times 3$  非最大抑制法尋找興趣點，此方法是比較候選點與周圍 26 個點的 Hessian 行列式值，當候選點的行列式值大於預設之門檻值(Threshold)及周圍的 26 個點，則視為興趣點。其中，門檻值用來事先過濾候選點，當候選點的行列式值大於此門檻值才會與其它

26 個點比較以決定是否為興趣點，否則將直接捨去。當門檻值設得越大，興趣點的數量會減少，從而使 SURF 特徵數量也減少；當門檻值設得越小，特徵數量則會變多，最佳化之門檻值則必須依不同的應用而做適當的調整。本次的動態擷取 SURF 特徵實驗都是在相同的畫面之下，藉由調整不同的門檻值以動態地得到畫面中的 SURF 特徵，主要目的是觀察門檻值、SURF 特徵數量及計算時間之關係。另外，人機介面軟體是在每次接收完一個畫面的影像及 SURF 特徵描述子之後才更新畫面及計算時間，所以由人機介面中就可以得到不同門檻值對應的 SURF 特徵數量及計算時間(近似於畫面更新時間)。圖 5-5 到圖 5-8 分別是小乳牛(亮背景)、小乳牛(暗背景)、混合物件及書本在四種不同門檻值之下的 SURF 特徵描述子擷取畫面，在(a)至(d)畫面中的左下角可以看到門檻值依序是 50、200、350、500，同時也有存檔時的 SURF 特徵數量及畫面更新時間。

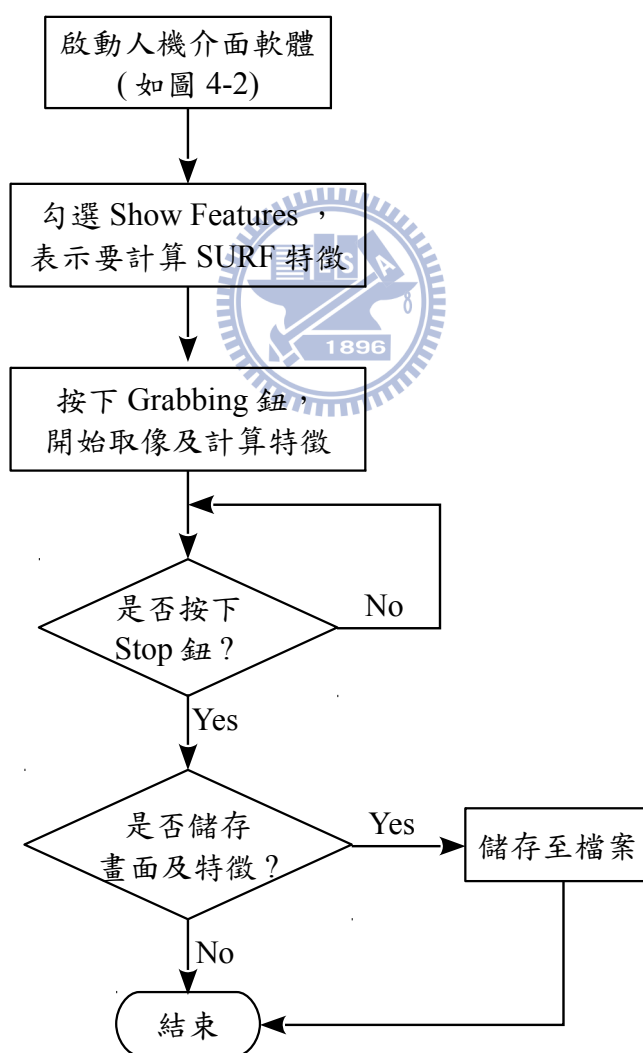
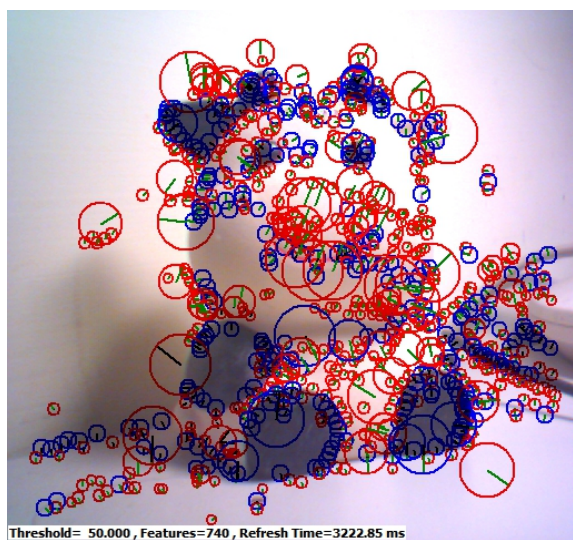
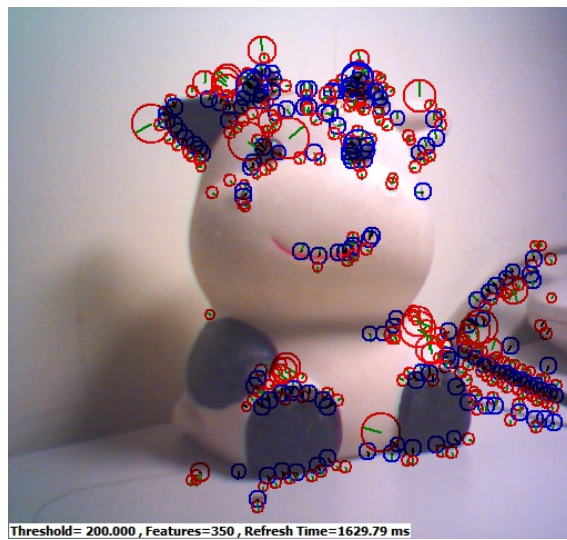


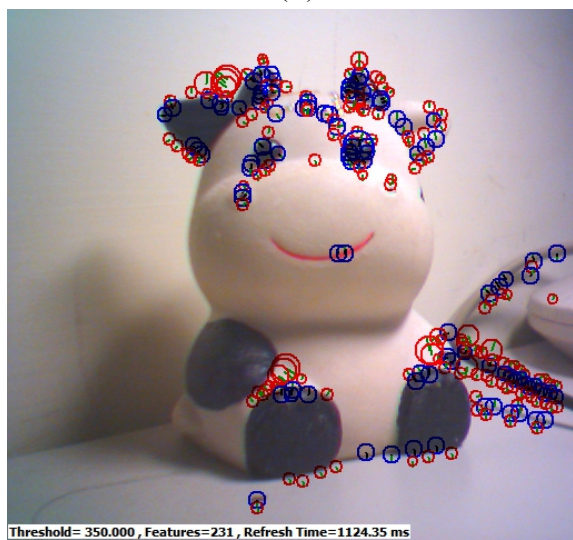
圖 5-4 特徵擷取時軟體的操作流程圖



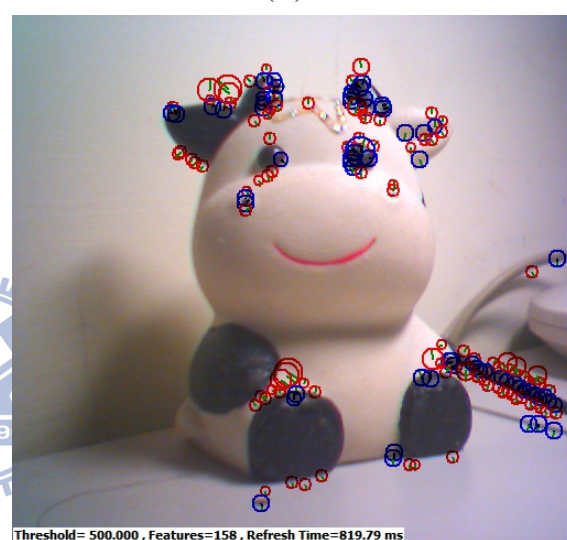
(a)



(b)

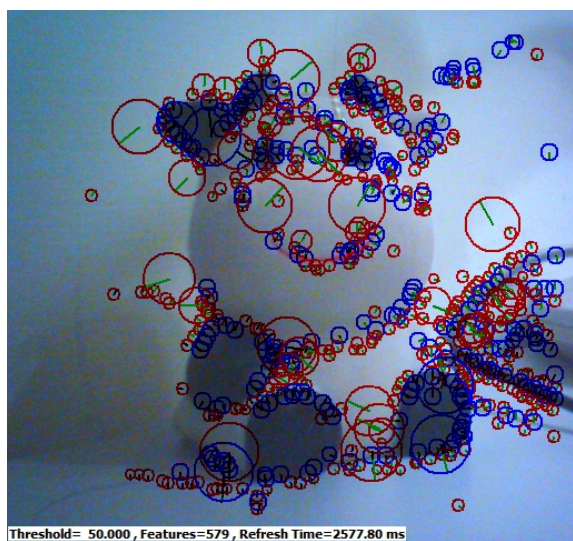


(c)

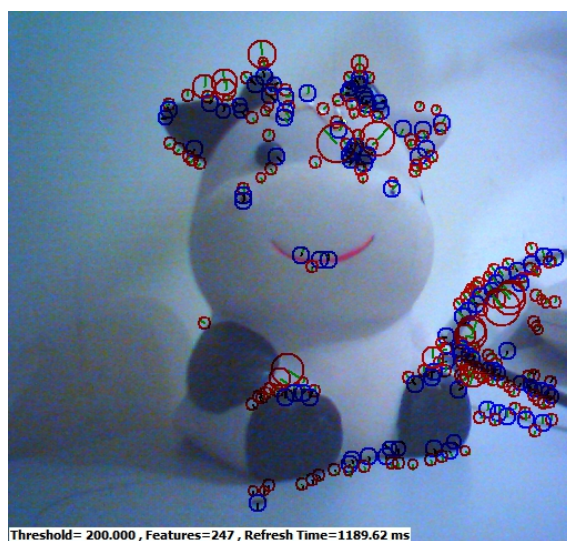


(d)

圖 5-5 不同門檻值之下小乳牛(亮背景)的 SURF 特徵描述子 (a)門檻值=50 (b)門檻值=200 (c)門檻值=350 (d)門檻值=500



(a)



(b)

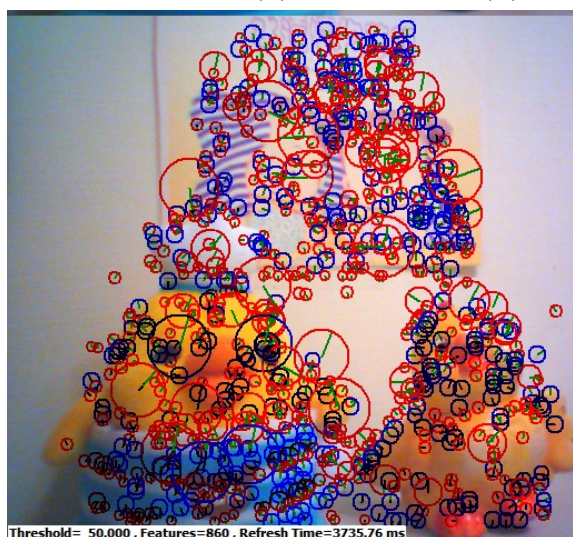


(c)

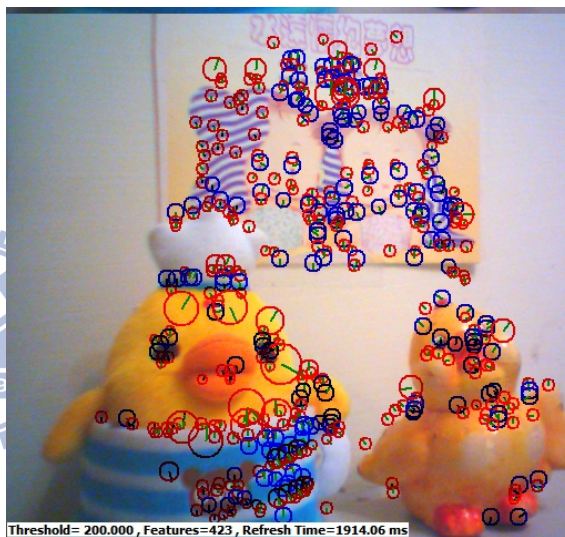


(d)

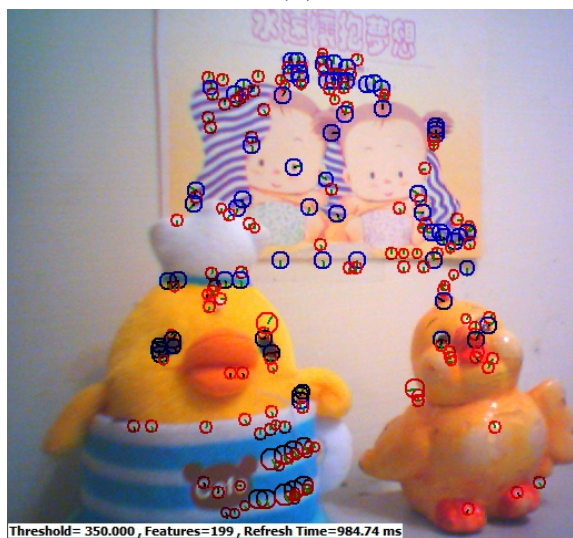
圖 5-6 不同門檻值之下小乳牛(暗背景)的 SURF 特徵描述子 (a)門檻值=50 (b)門檻值=200 (c)門檻值=350 (d)門檻值=500



(a)



(b)



(c)



(d)

圖 5-7 不同門檻值之下混合物件的 SURF 特徵描述子 (a)門檻值=50 (b)門檻值=200 (c)門檻值=350 (d)門檻值=500

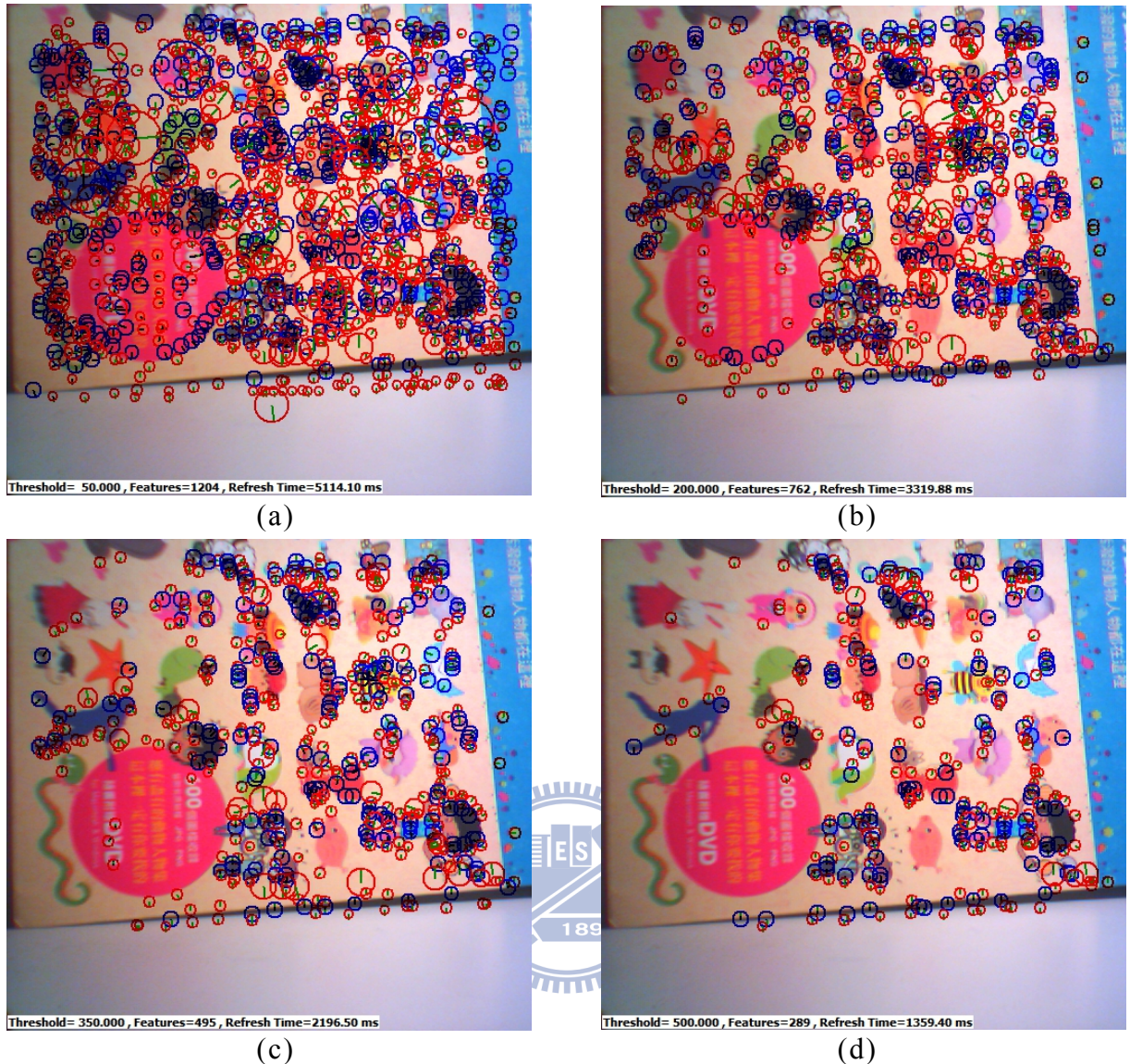


圖 5-8 不同門檻值之下書本的 SURF 特徵描述子 (a)門檻值=50 (b)門檻值=200 (c)門檻值=350 (d)門檻值=500

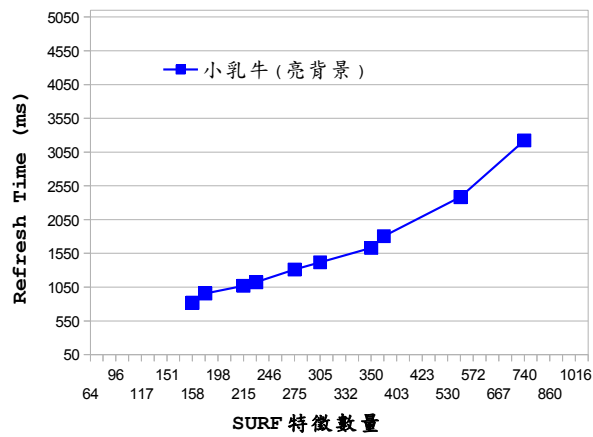
表 5-1 是上述物件擷取時的 SURF 特徵描述子數量及計算時間(畫面更新時間)的統計表，由表中可以看到門檻值越高得到的 SURF 特徵越少，同時計算時間也相對地比較短。但是如果和單純取像的畫面更新時間相比，加入 SURF 計算後速度卻是明顯地降低許多，其原因應該是 DSP 與 FPGA 之間的資料匯流排頻寬被佔用所致。DSP 與 FPGA 之間的資料匯流排 EMIF-A 的同步時脈頻率是 66.4MHz，匯流排寬度是 32 位元，理論上傳輸頻寬可達 265.6M Bytes/sec，但是由於 DSP 的 SDRAM 與 FPGA 同時連接在 EMIF-A，當 DSP 要由 FPGA 逐列地讀取原始影像與積分影像時，已經佔用一段 EMIF-A 匯流排的時間，加上 DSP 內部 L2 記憶體不足以儲存整張積分影像，所以還是必須先將它暫存到 SDRAM，此時又需要

佔用 EMIF-A 匯流排的頻寬；重要的是，進行 3x3x3 非最大抑制尋找興趣點、計算 Haar 響應與存取 SURF 特徵時會對 SDRAM 進行大量『非連續性』的讀寫，才會使得整體速度降低許多。要解決加入 SURF 計算後整體速度太低的問題，可以先對記憶體的存取進行最佳化，或者將尋找興趣點的工作安排在 FPGA 內部進行，以減少 DSP 對 SDRAM 的非連續性讀取。最理想的情形則是進一步地分析 SURF 演算法，分別找出 DSP 與 FPGA 最適合的運算工作。

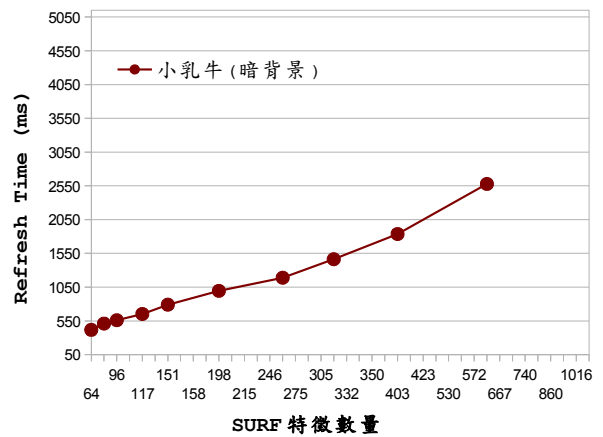
圖 5-9 是將表 5-1 中不同擷取畫面的 SURF 特徵數量與計算時間以個別的折線圖來表現，由圖中可以看到在目前的系統中，雖然擷取的對象不同但是 SURF 計算時間都會隨著特徵數量而增加。圖 5-10 則是再將圖 5-9 的四張折線圖畫在同一張圖中，如此可以清楚地看出其變化趨勢的一致性，顯示目前的嵌入式影像處理系統仍具有一定的穩定性。

表 5-1 不同門檻值之下的 SURF 特徵數量及計算時間

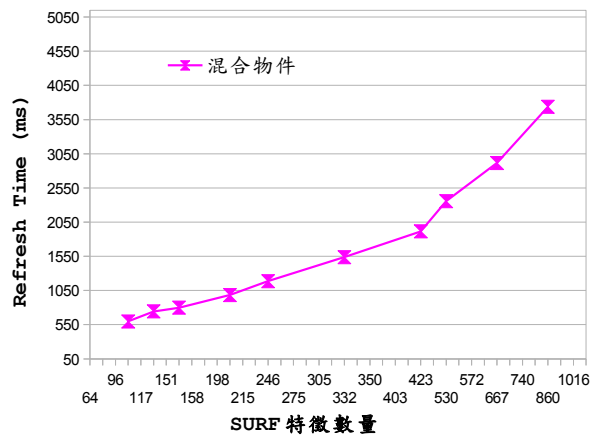
Threshold	小乳牛(亮背景)		小乳牛(暗背景)		混合物件		書本	
	Features	Refresh Time (ms)	Features	Refresh Time (ms)	Features	Refresh Time (ms)	Features	Refresh Time (ms)
500	158	819.79	64	417.53	106	597.40	289	1359.40
450	190	956.83	83	510.44	138	743.14	344	1566.93
400	215	1070.47	96	560.25	153	798.56	410	1865.94
350	231	1124.35	117	652.65	199	984.74	495	2196.50
300	275	1311.36	151	790.49	246	1187.24	572	2535.21
250	305	1417.04	198	995.02	332	1538.33	670	2944.27
200	350	1629.79	247	1189.62	423	1914.06	762	3319.88
150	399	1804.21	316	1463.32	530	2357.72	881	3784.77
100	538	2383.76	403	1836.97	667	2914.52	1016	4349.98
50	740	3222.85	579	2577.80	860	3735.76	1204	5114.10



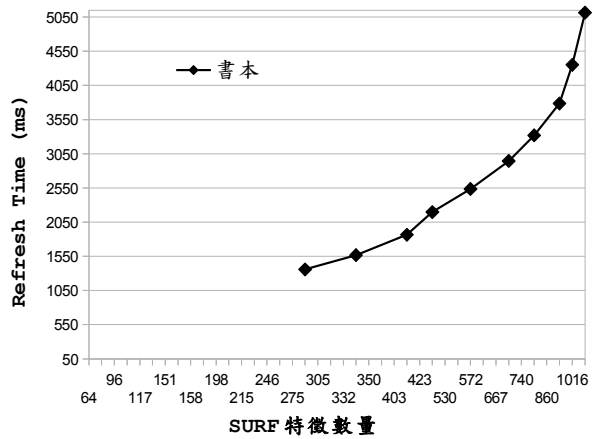
(a)



(b)



(c)



(d)

圖 5-9 個別的折線圖表現不同場景的 SURF 特徵數量與計算時間 (a)小乳牛(亮背景) (b)小乳牛(暗背景) (c)混合物件 (d)書本

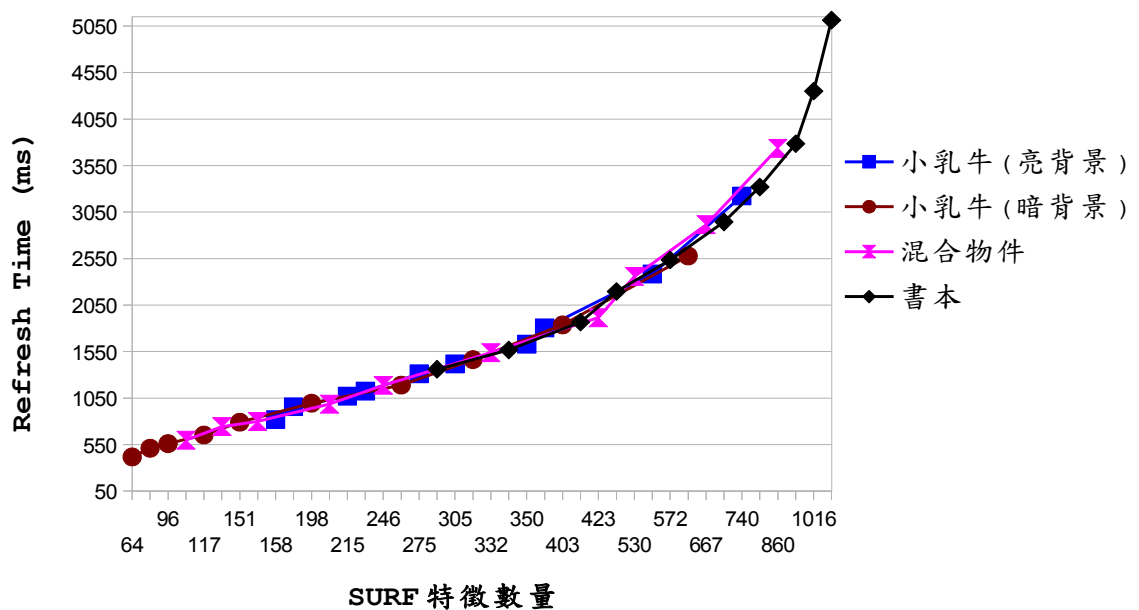


圖 5-10 同一張折線圖表現不同場景的 SURF 特徵數量與計算時間

### 5.3 動態 SURF 特徵匹配實驗

特徵匹配是在 PC 中使用最近鄰居演算法(Nearest Neighborhood Algorithm)來匹配由影像處理卡動態擷取及預先擷取並儲存之 SURF 特徵描述子，特徵擷取的方法如 5.2 節所述。本實驗使用 64 維的 SURF 描述子向量，假設  $V_i=[v_1, v_2, \dots, v_{64}]$  及  $P_j=[p_1, p_2, \dots, p_{64}]$  分別表示動態影像及樣本模型中的某一個 SURF 特徵向量，則兩者之間的距離(或相似性)可用(5-1)式來表示。

$$d_{ij}=\sqrt{\sum_{k=1}^{64}(v_k-p_k)^2} \quad (5-1)$$

以最近鄰居演算法匹配動態特徵  $V_i$  與樣本模型中的特徵向量時，步驟如下：

- 計算  $V_i$  與  $P_j$  的距離  $d_{ij}$ ，其中  $j=1 \dots M$ ， $M$  是樣本模型中特徵向量的總數。
- 將所有  $d_{ij}$  排序，找出最小的前二個(最近鄰居及次近鄰居)。
- 根據[29]作者的建議，採用最近鄰居及次近鄰居  $d_{ij}$  之比值來剔除可能的錯誤配對，當比值越接近 1，比較可能是背景或雜訊造成的錯誤配對。在本實驗中取比值為 0.65 進行篩選，若比值小於 0.65 則視為匹配成功。

實驗時，首先用 5.2 節的方法動態地擷取畫面中的 SURF 特徵，再將畫面及 SURF 特徵描述子儲存在檔案中。接著載入此畫面及特徵描述子檔案做為樣本模型並顯示在 PC 端人機介面軟體的右側，然後使用影像處理卡動態地擷取畫面中的 SURF 特徵描述子並且傳送到 PC 端進行特徵匹配。圖 5-11 所示為特徵匹配時 PC 端的軟體操作流程圖，圖中的取像及計算特徵在影像處理卡執行後將結果傳至 PC，特徵匹配及其它動作則在 PC 端完成。

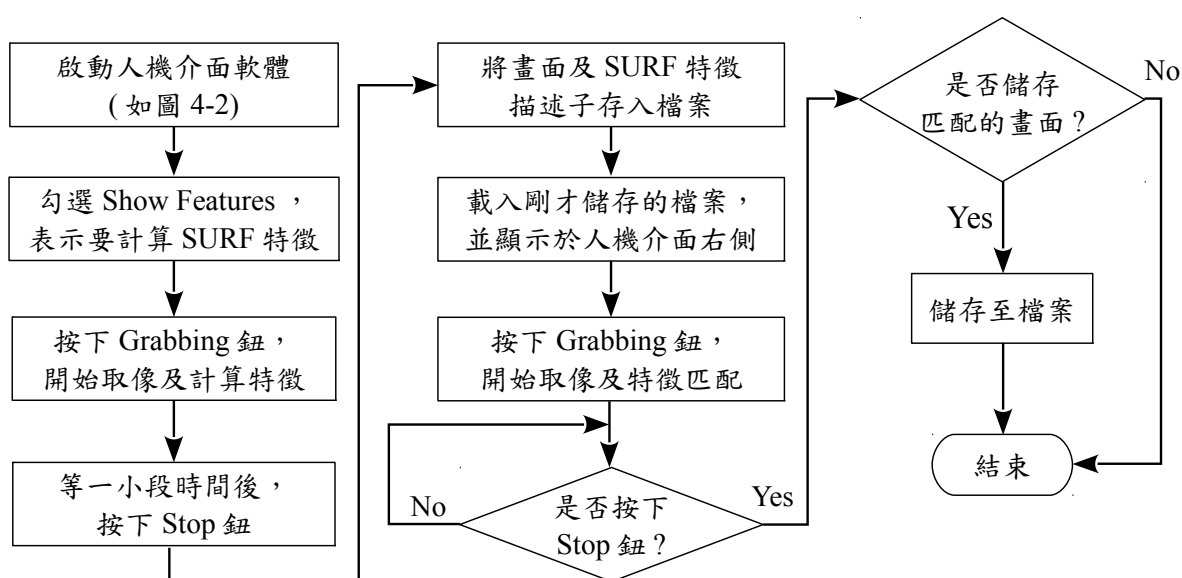
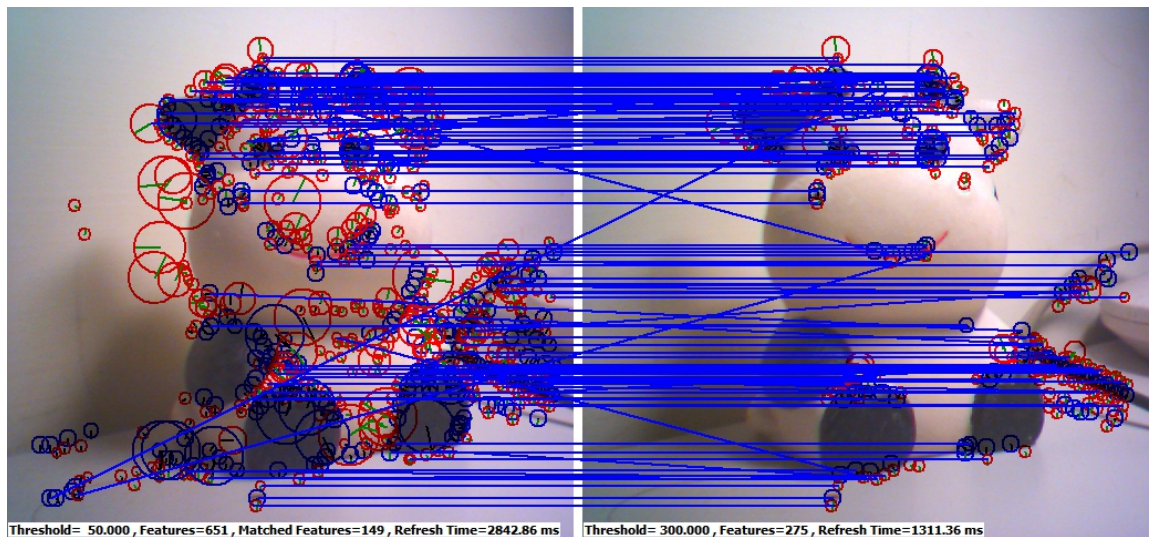
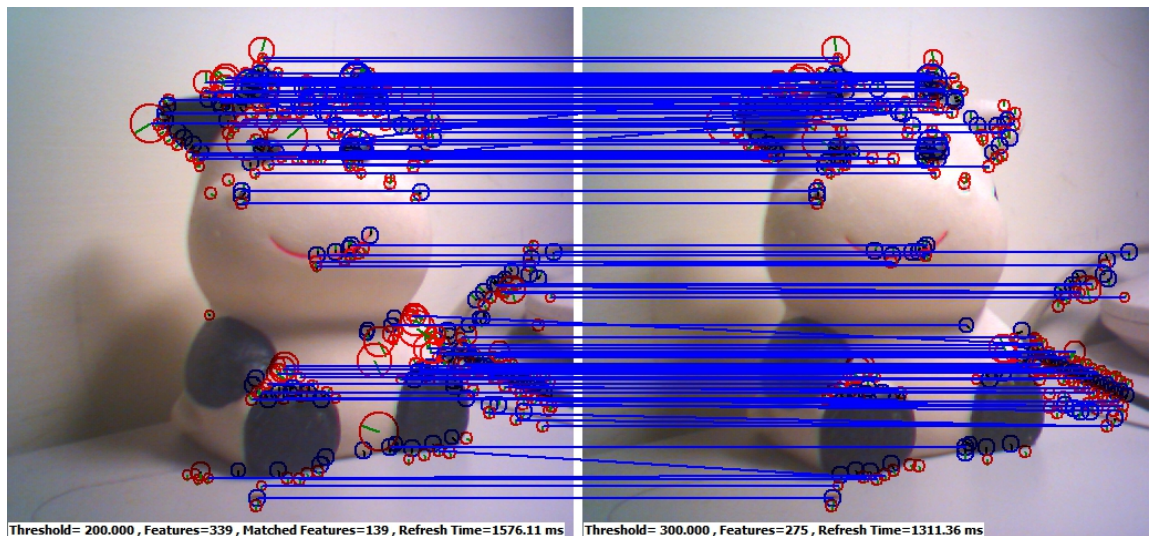


圖 5-11 在 PC 端進行特徵匹配時的軟體操作流程圖

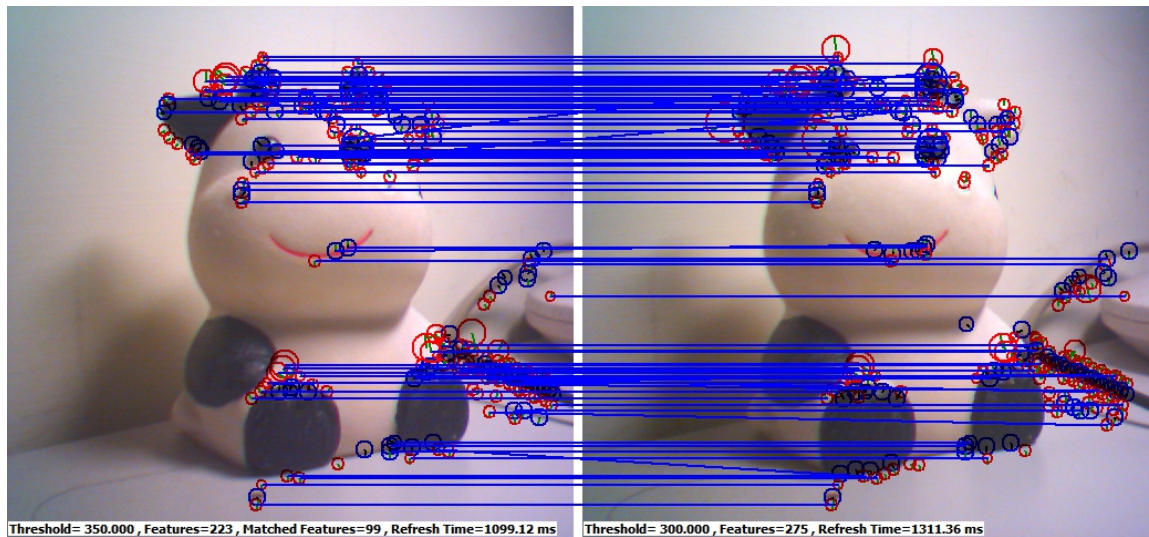
圖 5-12 至圖 5-14 是在不同門檻值之下小乳牛、混合物件與書本的特徵匹配。樣本模型均放在圖的右側，其中小乳牛與混合物件的樣本模型使用之門檻值為 300，書本的門檻值則是 500；動態擷取之特徵畫面置於圖的左側，使用的門檻值依序是 50、200、350 與 500，各圖中同樣有存檔時各畫面的特徵數量與匹配的數量等。由圖 5-12 至圖 5-14 來看，當動態擷取特徵時門檻值設的越大，則如前節的結果一樣得到的特徵數量也比較少，同時匹配的數量也會越少，但是錯誤的匹配也同時減少。因為在實驗時物件的狀態都保持相同，所以兩個畫面中相同特徵的連線應該是接近水平的藍色線，若圖中有傾斜的線可以明顯地判斷為匹配錯誤。目前看來錯誤匹配的數量還有改善的空間，例如：改用更強健的匹配演算法等。



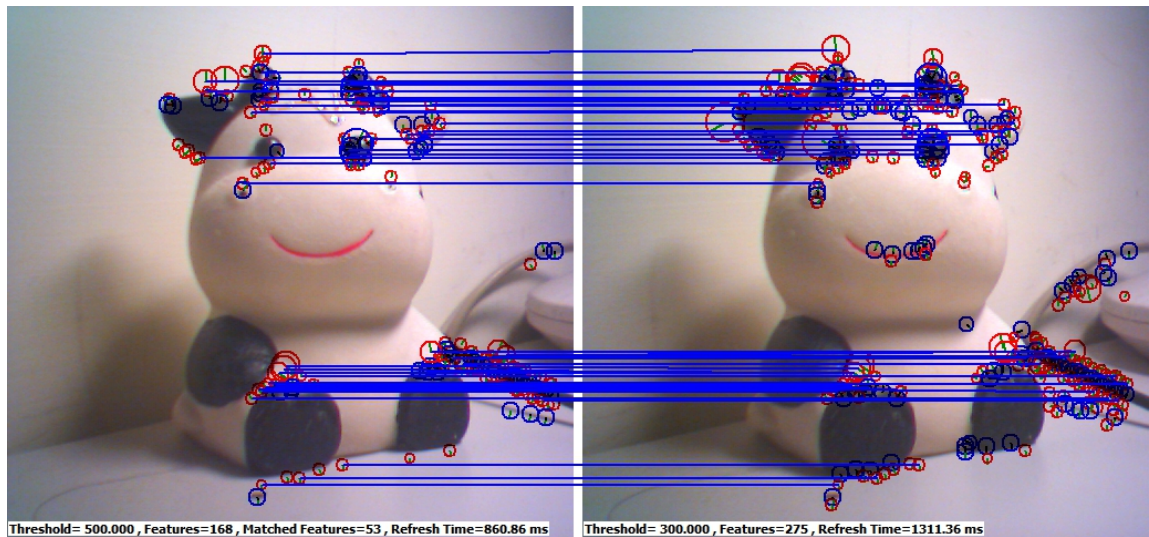
(a)



(b)

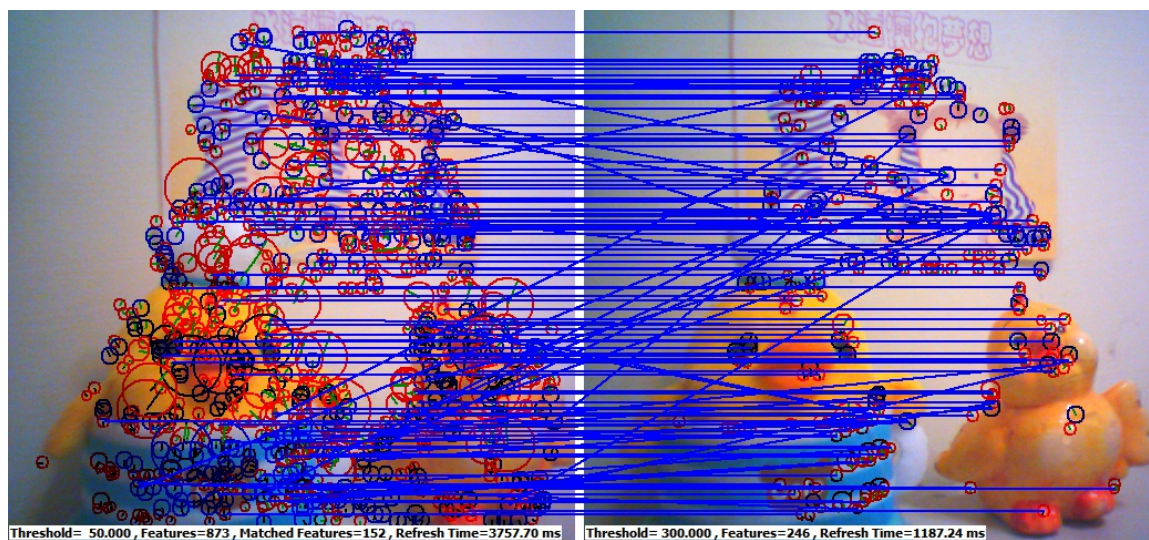


(c)

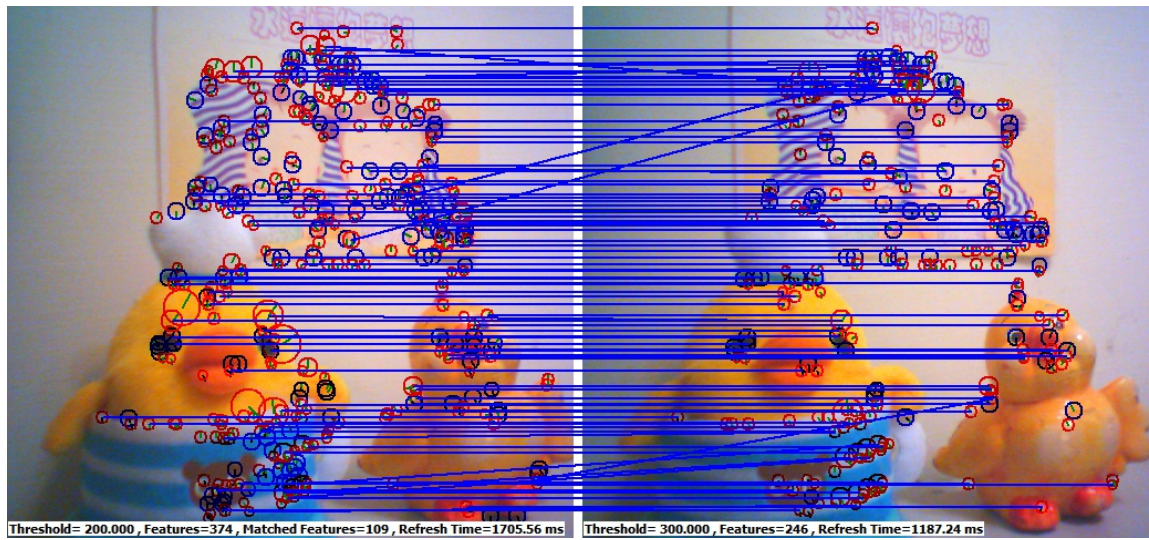


(d)

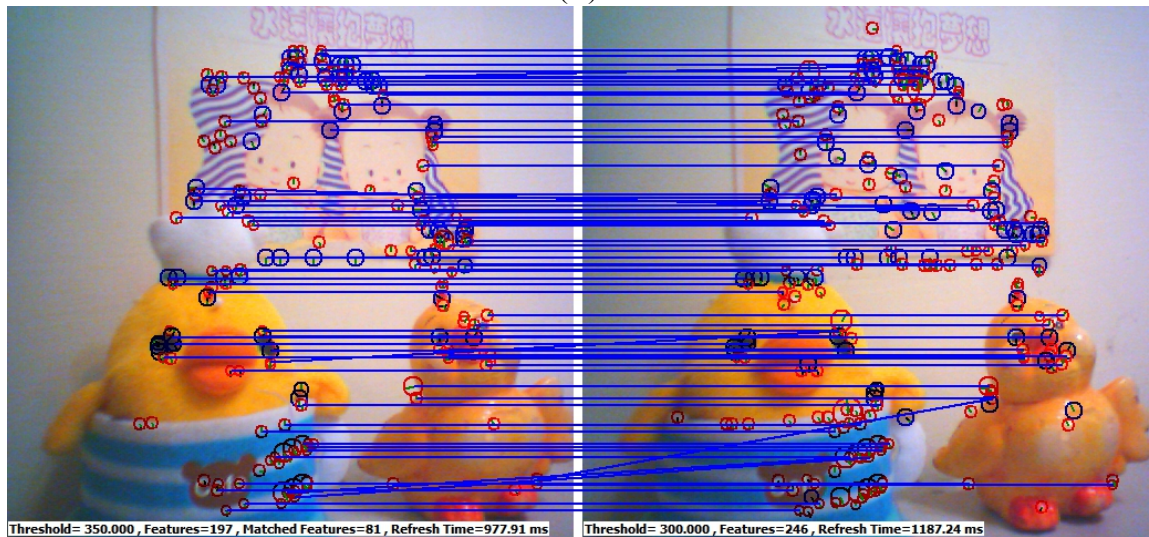
圖 5-12 小乳牛的特徵匹配。右側均為門檻值=300 的樣本模型，左側為動態擷取之特徵 (a)門檻值=50 (b)門檻值=200 (a)門檻值=350 (a)門檻值=500



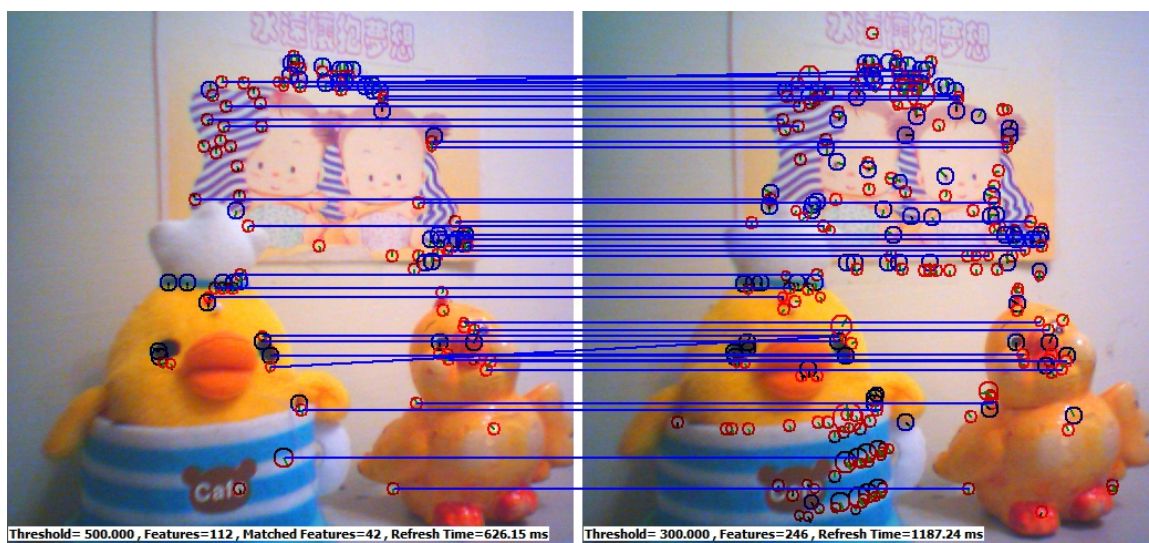
(a)



(b)

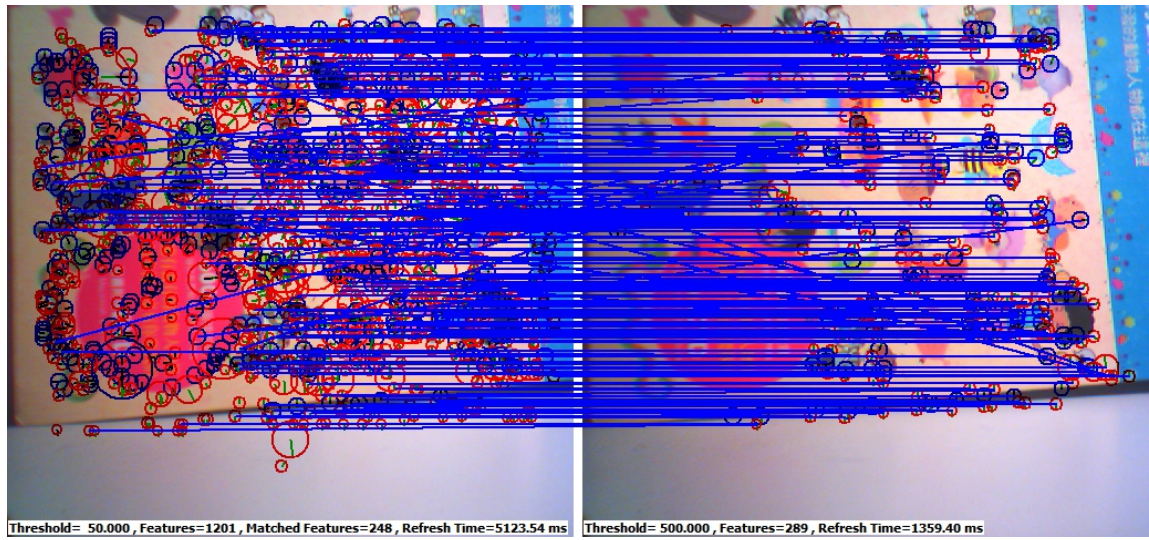


(c)

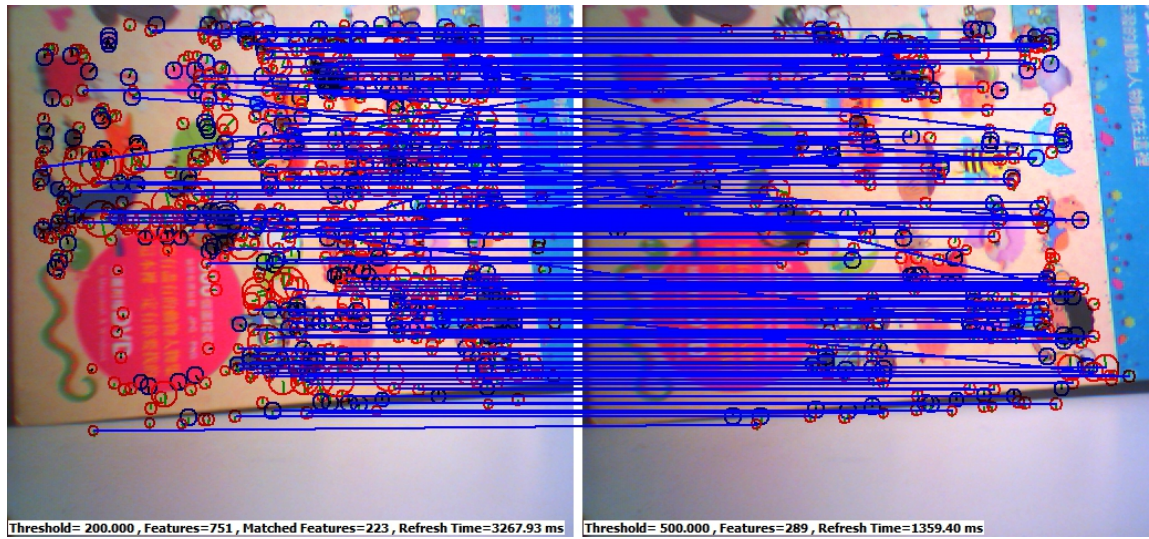


(d)

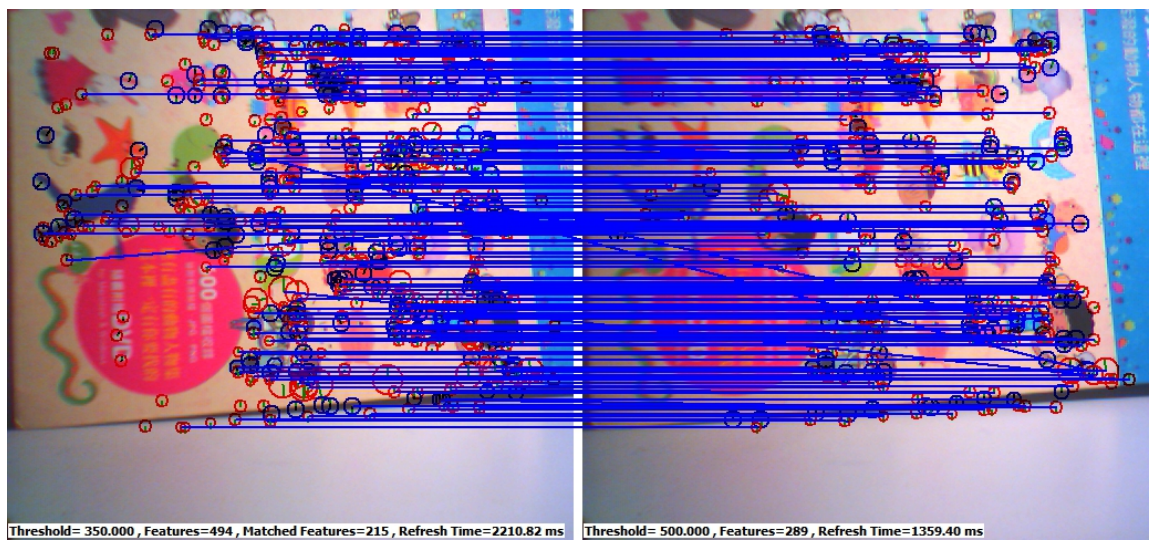
圖 5-13 混合物件的特徵匹配。右側均為門檻值=300 的樣本模型，左側為動態擷取之特徵 (a)門檻值=50 (b)門檻值=200 (a)門檻值=350 (a)門檻值=500



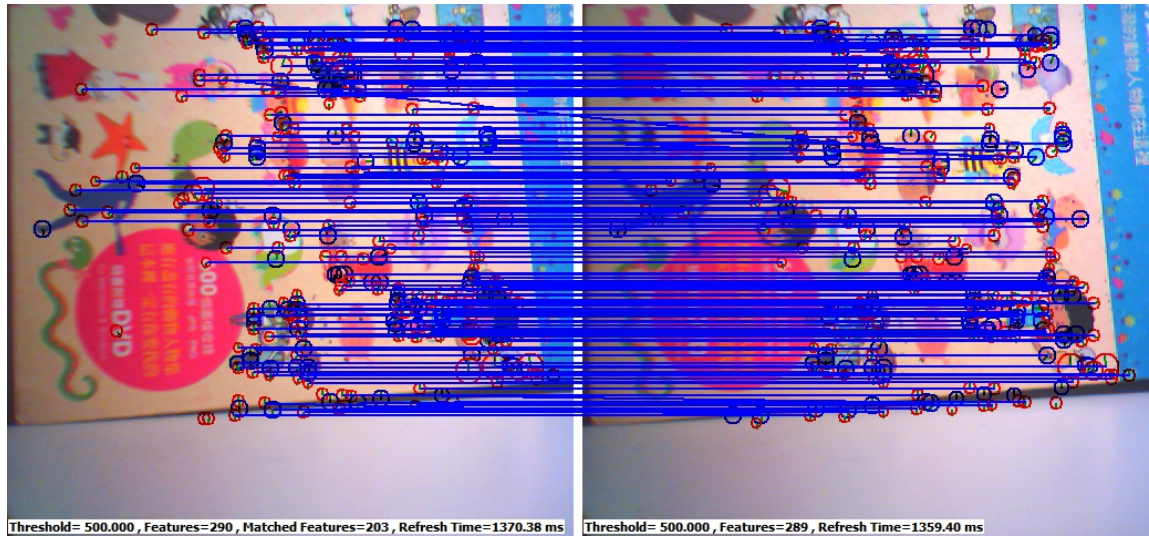
(a)



(b)



(c)



(d)

圖 5-14 書本的特徵匹配。右側均為門檻值=500 的樣本模型，左側為動態擷取之特徵 (a)門檻值=50 (b)門檻值=200 (a)門檻值=350 (a)門檻值=500

表 5-2 是上述圖形在不同門檻值之下進行動態匹配時的結果統計。表中上側藍綠色背景內是匹配時的樣本模型資料，特徵數量與更新時間同前節的說明，匹配的數量則是上列圖形中左側的動態畫面與右側的樣本模型在不同門檻值之下進行匹配所得的結果。圖 5-15 是將表 5-2 的資料以折線圖表示不同門檻值之下特徵與匹配數量的趨勢。由圖可看出特徵數量隨著門檻值降低而增加，但是匹配的數量並不會隨著特徵數量的增加而變化太多，這是因為樣本模型的特徵數量已經是固定值，故匹配的數量不會大於此值。另外，匹配的數量與樣本模型的特徵數量有關，當同一個樣本模型的特徵數量越多，樣本模型的特徵數量越少，但是匹配率會比較高，例如表 5-2 中書本的樣本模型門檻值為 500，當動態擷取的特徵門檻值也是 500 的時候，匹配率大約 70.2%；但是另外兩種樣本模型設為比較低的 300，即使動態擷取的特徵門檻值也是 500 時，匹配率卻只有 17%~19 %。另外，對同一個物件而言，雖然將動態擷取的特徵門檻值設低可得到比較多的特徵，但是由於特徵的獨特性較差，可能會產生比較多的錯誤匹配。

#### 5.4 OpenSURF 與本系統計算之 SURF 特徵比較

SURF 具有多種實作的方式，例如：原始作者的方法、OpenSURF[9]及 OpenCV 等。其中，OpenSURF 是一個開源專案，並且作者也有提供完整的說明文件[9]，另外參考文獻[30]也是在其軟體中整合 OpenSURF 程式庫以進行車子廠牌的辨識。本節將採用 OpenSURF 與本影像處理卡擷取之 SURF 特徵做比較，

表 5-2 不同門檻值之下的 SURF 特徵與匹配數量及計算時間

樣本 模型 動態 擷取	小乳牛(亮背景)			混合物件			書本		
	Threshold	300		Threshold	300		Threshold	500	
	Features	275		Features	246		Features	289	
Threshold	Features	Matched	Refresh Time (ms)	Features	Matched	Refresh Time (ms)	Features	Matched	Refresh Time (ms)
500	168	53	860.86	112	42	626.15	290	203	1370.38
450	184	62	928.53	130	49	714.15	352	220	1603.85
400	202	98	1001.74	150	62	798.42	407	219	1841.85
350	223	99	1099.12	197	81	977.91	494	215	2210.82
300	260	129	1224.99	259	102	1240.45	582	228	2563.68
250	305	129	1437.43	302	104	1404.5	668	223	2935.87
200	339	139	1576.11	374	109	1705.56	751	223	3267.93
150	395	137	1800.53	480	115	2134.02	879	241	3775.73
100	481	143	2155.27	624	136	2757.26	1014	235	4349.25
50	651	149	2842.86	873	152	3757.7	1201	248	5123.54

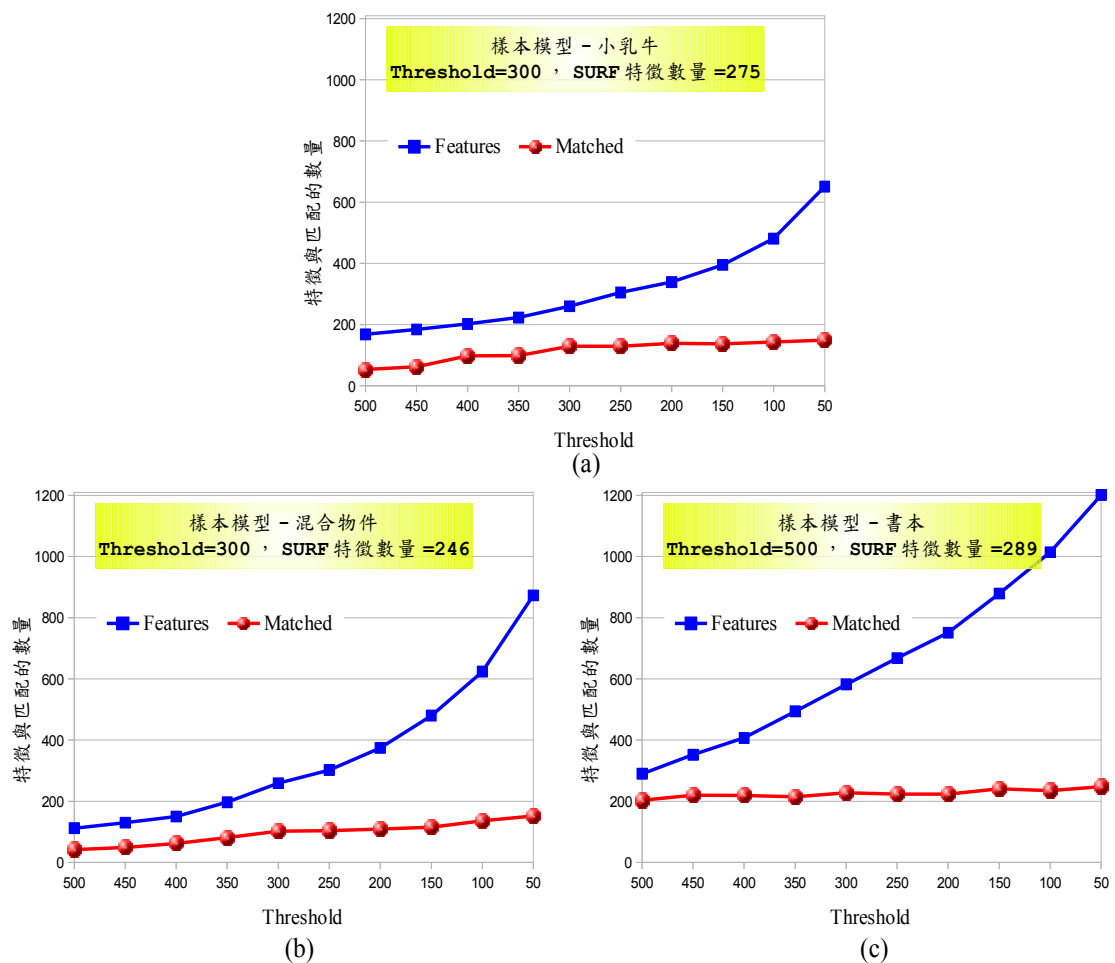


圖 5-15 不同門檻值之下 SURF 特徵與匹配數量 (a)小乳牛 (b)混合物件 (c)書本

目的是評估本影像處理卡擷取之 SURF 特徵的正確性。如果兩者找出的影像特徵分佈情形相似，表示本系統擷取之 SURF 特徵無誤，但仍需進一步將特徵描述子應用於影像特徵辨識等演算法，才能由辨識的成功率真正地確認目前系統擷取之特徵的正確性。

實驗的進程序，首先利用前述之動態特徵擷取實驗，將本系統擷取之特徵畫面及原始影像畫面分別存檔，接著將原始影像畫面輸入 OpenSURF 程式中進行計算，最後分別顯示結果及進行比較。如圖 5-16(b)及圖 5-17(b) 為 OpenSURF 擷取之特徵；圖 5-16(a)及圖 5-17(a)則是本系統擷取之特徵。

由於本影像處理卡計算及傳回人機介面軟體的影像格式為 YUV4:2:2，但是使用 C++ Builder 的 TBitmap 物件顯示畫面及存檔時，必須先將畫面轉換為 RGB 格式，因此兩個演算法的輸入影像會因為轉換公式而不盡相同。另外，為了避免過於複雜的物件畫面使兩個輸入影像產生其它未知的轉換誤差，因此選擇如圖 5-16 及圖 5-17 的簡單圖形畫面以方便進行比較，擷取的門檻值及特徵數量如圖下方之說明。由圖 5-16 及圖 5-17 可看出，因為前述的轉換誤差及電路中存在的雜訊等因素，使兩種 SURF 實作方式得到的特徵點位置並非完全相同，但偵測到之特徵分佈趨勢則大致相同，例如：特徵大致沿著物件的邊緣分佈。未來還需要將本系統擷取之 SURF 特徵應用於其它影像辨識等演算法，才能藉由辨識的成功率來判斷本系統擷取之特徵的正確性。

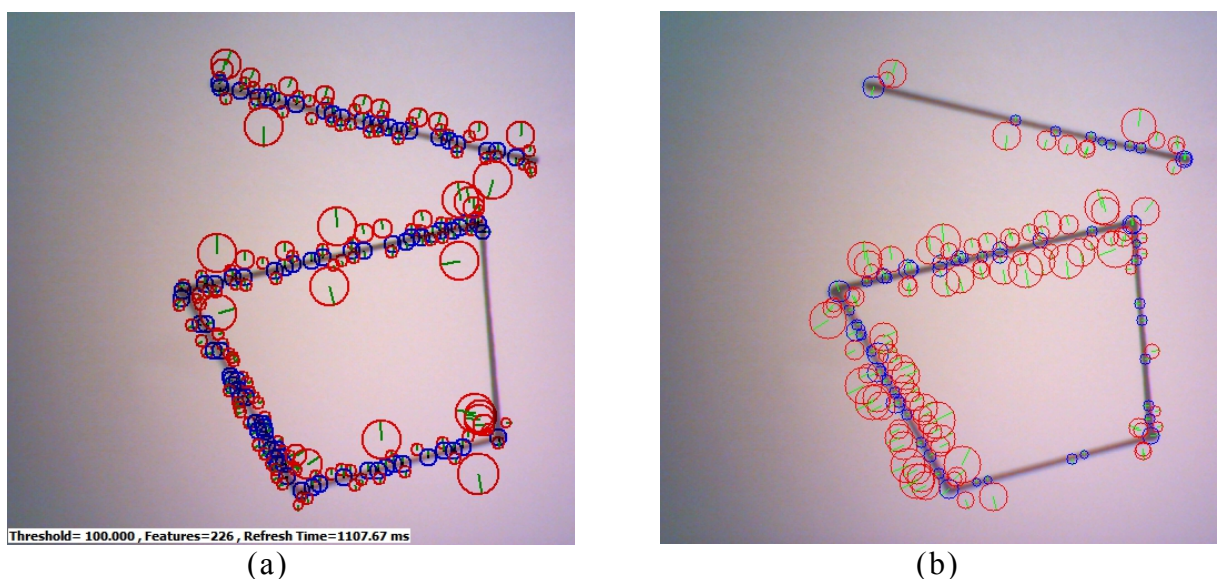
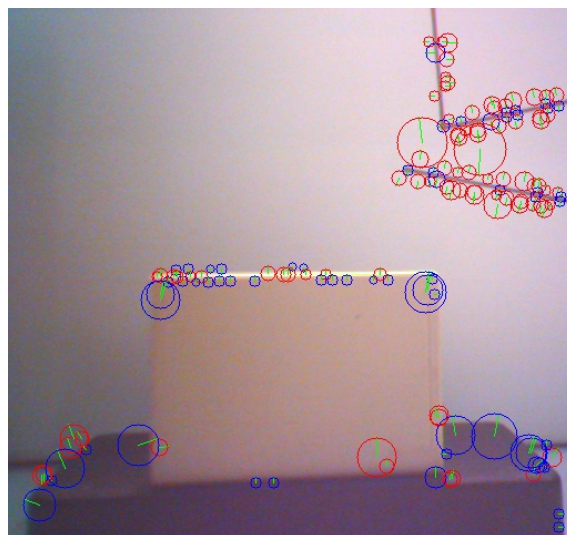


圖 5-16 不同 SURF 實作法擷取簡單幾何圖形之特徵 (a)本系統，門檻值=100，特徵數=226 (b)OpenSURF，門檻值=0.00015，特徵數=155



(a)



(b)

圖 5-17 不同 SURF 實作法擷取方形物件之特徵 (a)本系統，門檻值=100，特徵數=182 (b)OpenSURF，門檻值=0.00004，特徵數=145



## 第六章 結論與未來工作

### 6.1 結論

現今個人電腦的效能強大，只要搭配影像擷取卡及 CCD 影像感測裝置就可以開發影像處理之演算法。然而，若針對機器人視覺的應用而言，此種配置的體積大、價格高並且耗費能源。有鑑於此，本論文設計一套基於數位訊號處理器(DSP)與可程式邏輯閘陣列(FPGA)之嵌入式影像處理系統，藉著 DSP 的高運算效能及 FPGA 彈性的電路配置能力，達到嵌入式之影像處理功能，例如：擷取影像特徵等。除此之外，本系統具有一組高速 USB 及二組 RS232 介面可用來與外部裝置通訊：高速 USB 操作於巨量型傳輸時，最大可達 22MB/sec 的數據傳輸率，因此適用於頻寬大之即時影像畫面及影像特徵等資料傳輸；RS232 則可以與機器人馬達控制系統等裝置溝通。

在影像演算法的開發過程中常需要反覆地修改與驗證，因此系統內的 DSP 及 FPGA 都各自具有 JTAG 介面方便進程式及電路內容下載。此方法能快速地下載並除錯，並且不會反覆地燒錄非揮發性的程式記憶體而減低其壽命，只是 JTAG 下載方式必須配合開發環境，所以僅適合用於開發階段。當 DSP 程式及 FPGA 的電路內容開發完成後，為了能脫離開發環境而獨立運作於機器人平台，必須要提供燒錄介面將最終的程式及電路內容儲存在非揮發性的記憶體：FPGA 可以使用主動式串列介面將電路內容燒錄至配置用記憶體；DSP 的 FLASH ROM 則使用自行撰寫的燒錄程式由 USB 介面將程式碼傳送至 DSP 進行燒錄。另外，DSP 及 FPGA 的製造廠商(德州儀器公司及 Altera 公司)均有提供付費或免費之程式庫及 IP 模組，也有豐富的應用資料可供參考，可以讓設計者專注於演算法的開發。

在本嵌入式影像處理系統完成之後，為了驗證硬體電路動作是否符合預期，因此在系統中實作快速強健特徵(Speeded-up Robust Features, SURF) [1] 演算法，並且設計一套人機介面軟體用來控制此系統、顯示影像畫面與 SURF 特徵描述子，最後以此環境搭配另外設計的 CMOS 影像感測器電路進行三項實驗，分別為即時影像擷取、動態 SURF 特徵擷取及動態 SURF 特徵匹配。在即時影像擷取實驗中，由 FPGA 擷取大小為 512x480 像素的影像畫面，先存入外部的 SSRAM 緩衝區後再讀入 FPGA 內部的雙時脈 FIFO，在每一列結束後 FPGA 就會送出一個訊號觸發 DSP 的 EDMA 控制器，當 EDMA 將一列影像資料接收完後，DSP 會立即再將此列資料由高速 USB 傳送至人機介面軟體顯示。此實驗結果可正確地顯示影像畫

面，畫面的更新週期平均約 33.5ms，故可確認硬體電路的設計已能符合預期的功能。

在動態 SURF 特徵擷取及匹配的實驗中，影像的擷取流程與即時影像擷取實驗相同，只是在每一列取像後會由 FPGA 進行即時的影像積分，再經由 EDMA 傳送至 DSP 執行 SURF 演算法，當尋找到興趣點並計算 SURF 描述子後，立即由高速 USB 傳回人機介面軟體進行顯示及匹配。SURF 演算法的執行時間隨著特徵數量而增減，當特徵數量等於 64 個，計算時間大約 417.5ms；當特徵數量等於 199 個，計算時間大約 985ms。此時間與單純取像相比差距甚大，原因應該是 DSP 與 FPGA 之間的資料匯流排使用尚未達到最佳化，使得 DSP 對其 SDRAM 進行過多『非連續性』的讀寫，例如：3x3x3 非最大抑制尋找興趣點、計算 Haar 響應及存取 SURF 特徵等，因此使得 SURF 特徵的計算速率降低許多。雖然如此，但由圖 5-10 可看出即使不同的偵測對象，計算時間隨特徵數量增加的變化曲線相當一致，可知此系統仍具有一定的穩定性。

特徵匹配的實驗，是先擷取一個 SURF 特徵畫面後存檔，再由人機介面載入當成樣本模型，然後再動態擷取 SURF 特徵畫面進行匹配，匹配使用的是最近鄰居演算法。匹配的數量與樣本模型的門檻值有關，當同一個樣本模型的門檻值越高，樣本模型的特徵數量越少，但是匹配率會比較高，例如表 5-2 中書本的樣本模型門檻值為 500，當動態擷取的特徵門檻值也是 500 時，匹配率大約 70.2%；但是另外兩種樣本模型設為 300，即使動態擷取的特徵門檻值也是 500 時，匹配率卻只有 17%~19 %。另外，對同一種物件而言，雖然將動態擷取的特徵門檻值設低可得到比較多的特徵，但是由於特徵的獨特性較差，可能會產生比較多的錯誤匹配。

## 6.2 未來工作

由實驗結果來看，在本系統實作 SURF 演算法之效能有待改善，因此提出下列四項做為未來的改善方向：

1. 分析各項運算的時間，使 DSP 記憶體及 EMIF 介面的使用達到最佳化。
2. 本系統最初的電路規劃主要希望能運用在機器人進行嵌入式影像處理，故 FPGA 選用 Altera 公司低消耗功率及高性價比的 Cyclone II 系列，型號為 EP2C35F672C6。但電路設計時有先預留 FPGA 的升級空間，可以選擇同系

列中最高等級並且封裝同樣是 672-Pin FineLine BGA 的 EP2C70F672C6。  
表 6-1 是兩者的內部資源比較，由表中可看出 2C70 的邏輯單元是目前的 2 倍多，記憶體位元大約是 2.38 倍，嵌入式乘法器的數量大約是 4.28 倍。

3. 以目前實作的 SURF 演算法來看，尋找興趣點的程序具有規律性，但興趣點的產生與畫面有關而沒有一定的規則。因此，可以進一步將 Fast-Hessian 興趣點偵測器以 FPGA 來實現，當找到興趣點之後先存入雙時脈 FIFO，再由 DSP 讀出後進行 SURF 描述子的計算，如此可以讓 DSP 與 FPGA 採用類似 Pipeline 方式同步進行運算以提高計算效能。除此之外，由於 CMOS 影像感測器是逐列地輸出影像資料，而計算 SURF 描述子最大需要以興趣點為中心計算 20s X 20s 方形區域內的 Haar 小波響應，此時 CMOS 影像感測器的輸出卻尚未達到可計算 Haar 小波響應的列數，故 FPGA 無法在偵測到興趣點後立即計算描述子。因此將描述子的計算放在 DSP，而 DSP 與 FPGA 之間儲存興趣點的 FIFO 正好可當做緩衝區。
4. 在電路設計方面，考慮將 DSP 與 FPGA 模組分成兩個獨立的電路板，兩者之間以 Piggyback 的方式連接。如此不但可以進一步縮小尺寸，而且可以個別對 FPGA 或 DSP 進行更新與升級，甚至也能夠改變組合的方式，例如：使用近年發展快速的 ARM 處理器等。

最後，此系統並沒有限定只能實現 SURF 演算法，初期只是先以 SURF 來評估並驗證系統功能。因此往後還需多做不同的演算法實作，藉以收集更多的實驗數據，或者另外研究演算法實現於本系統的評估方式。如此一來，就可以有準則來判斷此系統是否適用於新的演算法。

表 6-1 FPGA EP2C35F672 與 EP2C70F672 內部資源比較

	EP2C35F672	EP2C70F672
<b>LEs</b>	33,216	68,416
<b>M4K RAM blocks</b>	105	250
<b>Total RAM bits</b>	483,840	1,152,000
<b>Embedded Multipliers</b>	35	150
<b>PLLs</b>	4	4

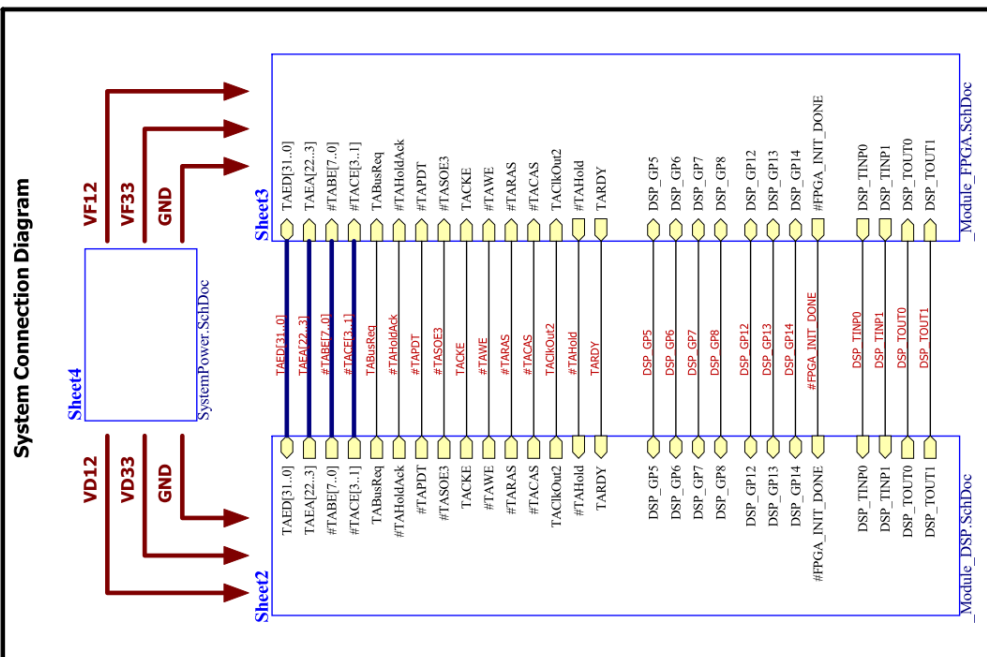
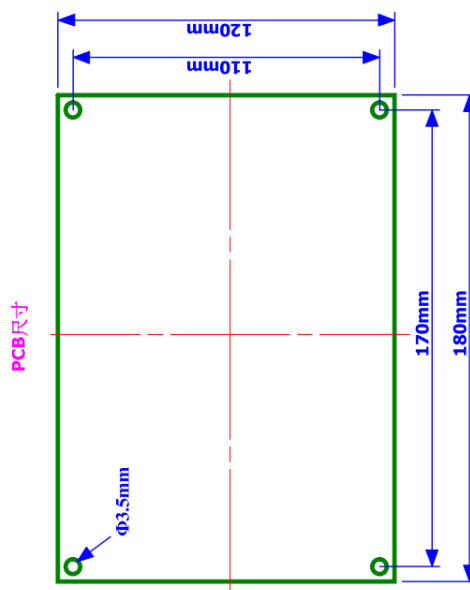
## 参考文献

- [1] Herbert Bay, Andreas Ess, Tinne Tuytelaars and Luc Van Gool, "SURF: Speeded Up Robust Features," *Computer Vision and Image Understanding*, Vol.110, No.3, 2008, pp.346-359.
- [2] Anthony Rowe, Charles Rosenberg and Illah Nourbakhsh, "A Low Cost Embedded Color Vision System," in *Proceedings of IROS 2002*, Switzerland, 2002, pp.208-213.
- [3] Stephanie McBader and Peter Lee, "An FPGA Implementation of a Flexible, Parallel Image Processing Architecture Suitable for Embedded Vision Systems," in *Proceedings of the International Parallel and Distributed Processing Symposium*, Nice, France, 2003, pp.228-232.
- [4] Vanderlei Bonato, Adriano K. Sanches, M.M. Fernandes, João M. P. Cardoso, E.D.V. Simoes and Eduardo Marques, "A Real Time Gesture Recognition System for Mobile Robots," in *Proceedings of International Conference on Informatics in Control, Automation, and Robotics*, Setúbal, Portugal, 2004, pp.207-214.
- [5] P. Chalimbaud and F. Berry, "Design of an Imaging System based on FPGA Technology and CMOS Imager," in *Proceedings of IEEE International Conference on Field-Programmable Technology*, Brisbane-Australia, 2004, pp407-411.
- [6] Stefan Mahlknecht, Roland Oberhammer and Gregor Novak, "A Real-time Image Recognition System for Tiny Autonomous Mobile Robots," in *Proceedings of the 10th IEEE Real-Time and Embedded Technology and Applications Symposium*, Toronto, Canada, 2004, pp.324-330.
- [7] Fen Xu, Jian-Jun Zeng and Yun-Long Zhang, "Design of a DSP-based CMOS Imaging System for Embedded Computer Vision," in *Proceedings of IEEE Conference on Cybernetics and Intelligent Systems*, Chengdu, China, 2008, pp430-433.
- [8] P.A. Viola and M.J. Jones, "Rapid Object Detection Using a Boosted Cascade of Simple Features," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Hawaii, 2001, pp.511-518.
- [9] Christopher Evans, "Notes on the OpenSURF Library," <http://www.chrisevansdev.com/computer-vision-opensurf.html>, 2009.
- [10] Alexander Neubeck and Luc Van Gool, "Efficient Non-Maximum Suppression," in *Proceedings of the 18th International Conference on Pattern Recognition*, Hong Kong, 2006.
- [11] M. Brown and D.G. Lowe, "Invariant features from interest point groups," in *Proceedings of British Machine Vision Conference*, Cardiff, Wales, 2002, pp.656-665.

- [12] Texas Instruments Inc., "TMS320C6000 DSP External Memory Interface (EMIF) Reference Guide," SPRU266E, 2008.
- [13] Altera Corporation, "Cyclone II Device Handbook," Volume 1, 2006.
- [14] OminVision Technologies, Inc. "OV7649 Color CMOS VGA(640x480) CAMERACHIP Datasheet," 2004.
- [15] OminVision Technologies, Inc. "OmniVision Serial Camera Control Bus(SCCB) Functional Specification," Ver. 2.2, 2007.
- [16] Texas Instruments Inc., "TMS320C6000 Chip Support Library API Reference Guide," SPRU401J, 2004.
- [17] Spansion Inc., "Am29LV320D Datasheet," 2005.
- [18] Texas Instruments Inc., "Creating a Second-Level Bootloader for FLASH Bootloading on TMS320C6000 Platform With Code Composer Studio," Application Report, SPRA999A1, 2006.
- [19] Texas Instruments Inc., "TMS320C6000 Tools: Vector Table and Boot ROM Creation," Application Report, SPRA544D, 2004.
- [20] 盧怡仁, 蔡偉和, "單晶片於數位信號處理的應用-以 TMS320C6000 的開發平台為例" 文魁資訊, 台北, 民國九十六年。
- [21] Cypress Semiconductor Corp., "EZ-USB FX2LP™ USB Microcontroller-High Speed USB Peripheral Controller," Rev. R, 2010.
- [22] Cypress Semiconductor Corp., "EZ-USB® Technical Reference Manual," Rev. C, 2010.
- [23] Cypress Semiconductor Corp., "Cypress CyAPI Programmer's Reference," 2011.
- [24] Texas Instruments Inc., "TMS320C620x/C642x McBSP: UART," Application Report, SPRA633C, 2008.
- [25] Altera Corporation, "Avalon Interface Specifications," ver. 11.0, 2011.
- [26] Altera Corporation, "Understanding Metastability in FPGAs," ver. 1.2, 2009.
- [27] Texas Instruments Inc., "Creating a Second-Level Bootloader for FLASH Bootloading on TMS320C6000 Platform With Code Composer Studio," Application Report, SPRA999A1, 2006.
- [28] Texas Instruments Inc., "TMS320C6000 EMIF to External Flash Memory," Application Report, SPRA568A, 2002.
- [29] D. G. Lowe, "Distinctive Image Features from Scale-invariant Keypoints," *International Journal of Computer Vision*, 2004, pp.91-110.
- [30] Štefan Badura and Stanislav Foltán, "Advanced scale-space, invariant, low detailed feature recognition from images - car brand recognition," in *Proceedings of the International Multiconference on Computer Science and Information Technology*, Wisla, 2010, pp.19-23.

# 附錄一、嵌入式影像處理系統電路圖

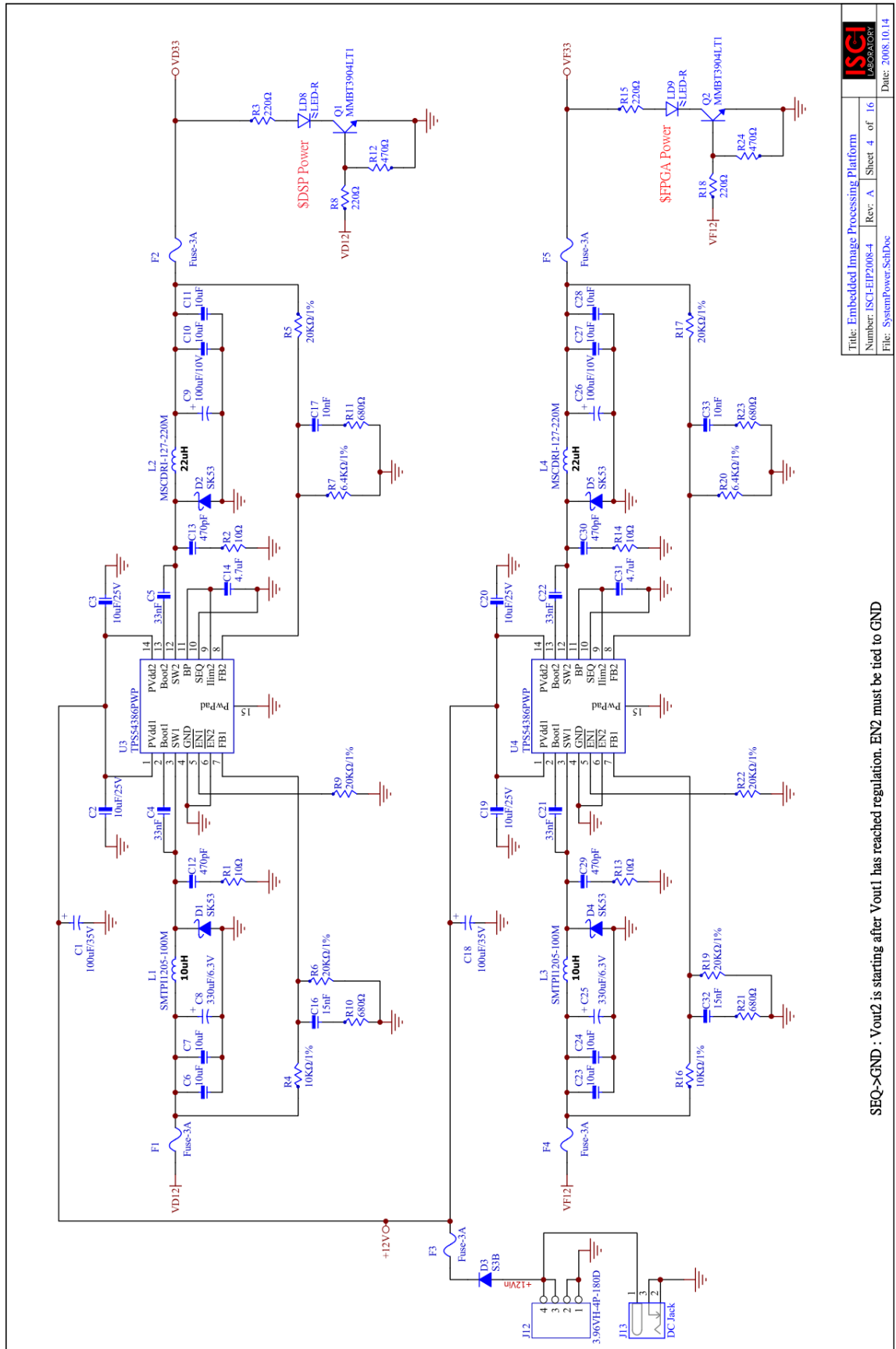
- Layout 注意事項:
1. 粉紅色文字用於提醒PCB設計者的注意事項
  2. 文字前面有\$ 符號者,表示要將其置於PCB的文字層, (不包含\$ 符號)
  3. 黑色文字用於電路功能的提示, PCB設計者可忽略
  4. 簡易牛角座(2.0mm或2.54mm)的第1Pin, 均向PCB內側
  5. PCB尺寸大致如下, 最後尺寸依Placement後決定, (越小越好)

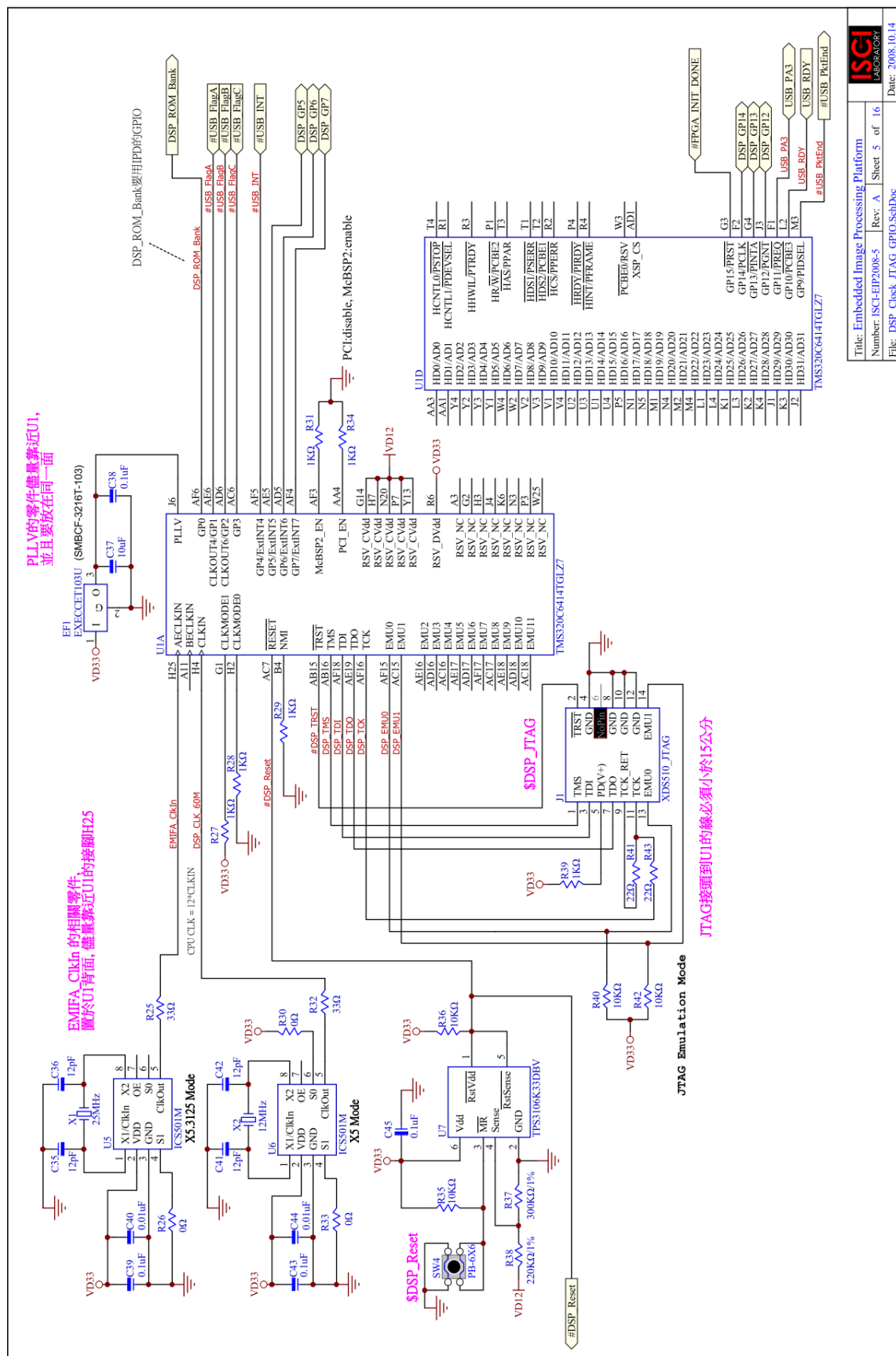


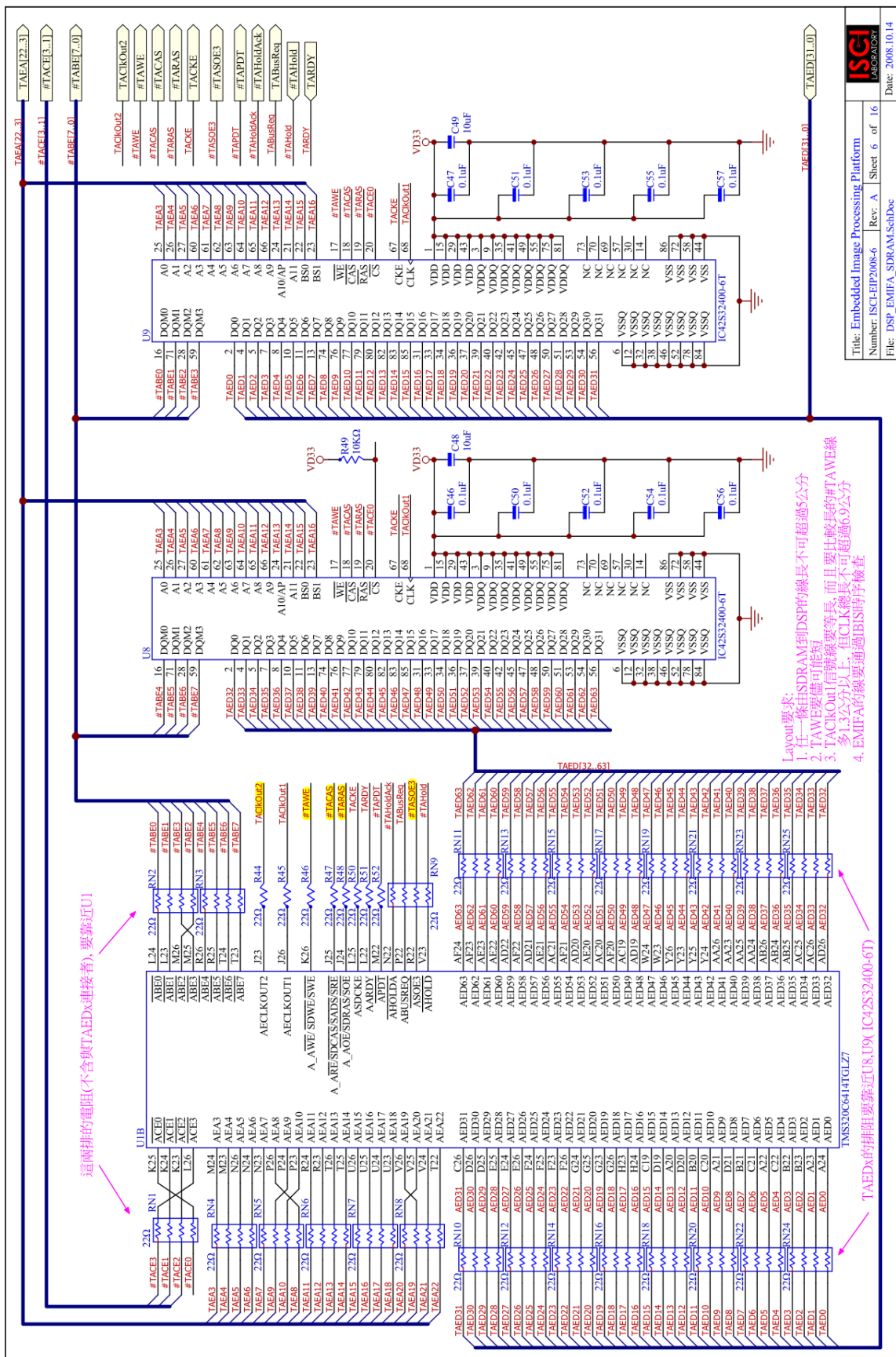
[illegible]

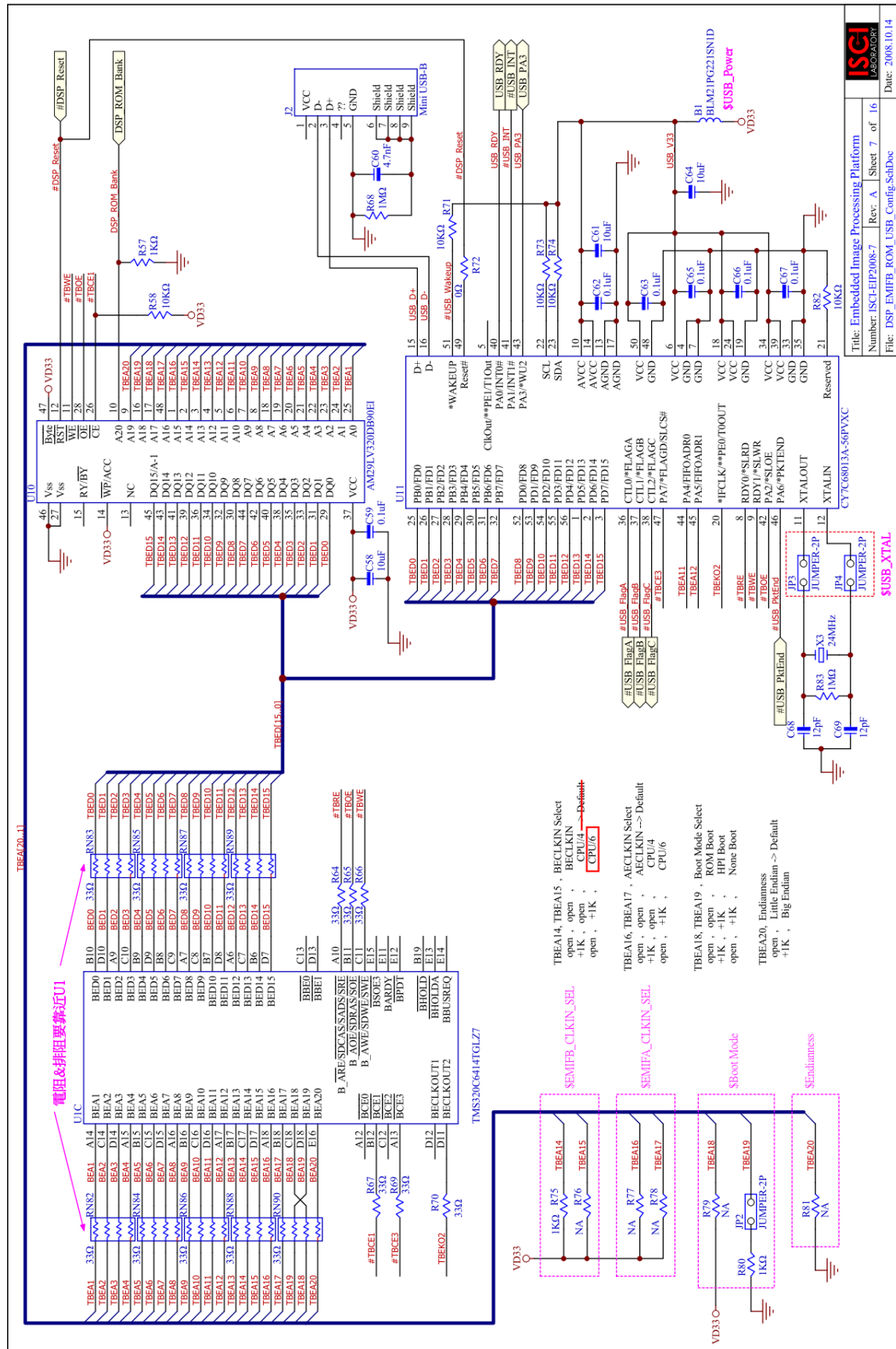
[illegible]

<div style="display: flex; justify-content: space-between; align-items: center;"> <div> <p>Title: Embedded Image Processing Platform</p> <p>Number: ISCI-EIP2008-3</p> <p>File: _Module_FPGA_SchDoc</p> </div> <div> <p>Rev: A</p> </div> <div> <p>Sheet 3 of 16</p> </div> </div>			 <p>Date: 2008.10.14</p>

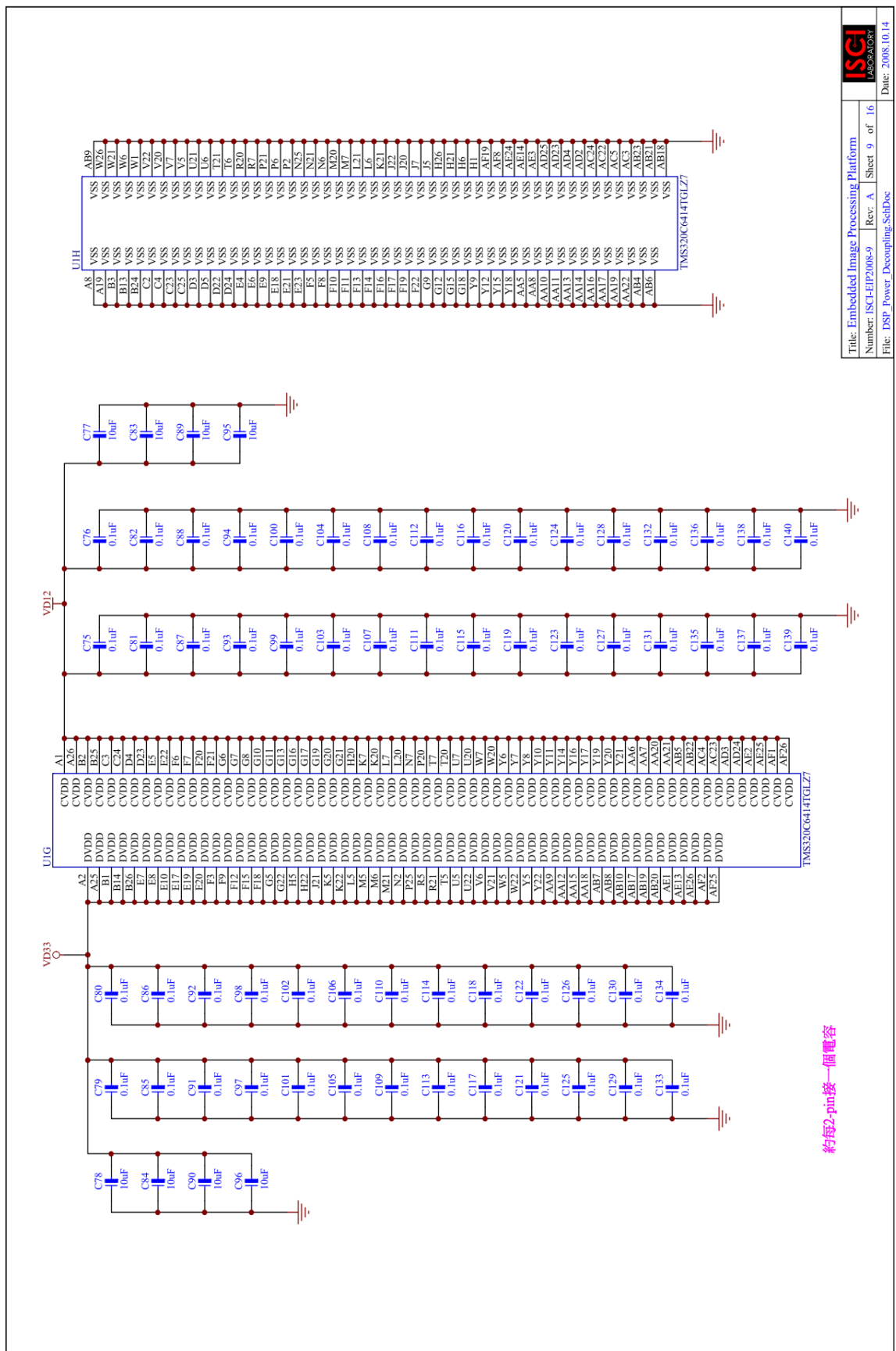




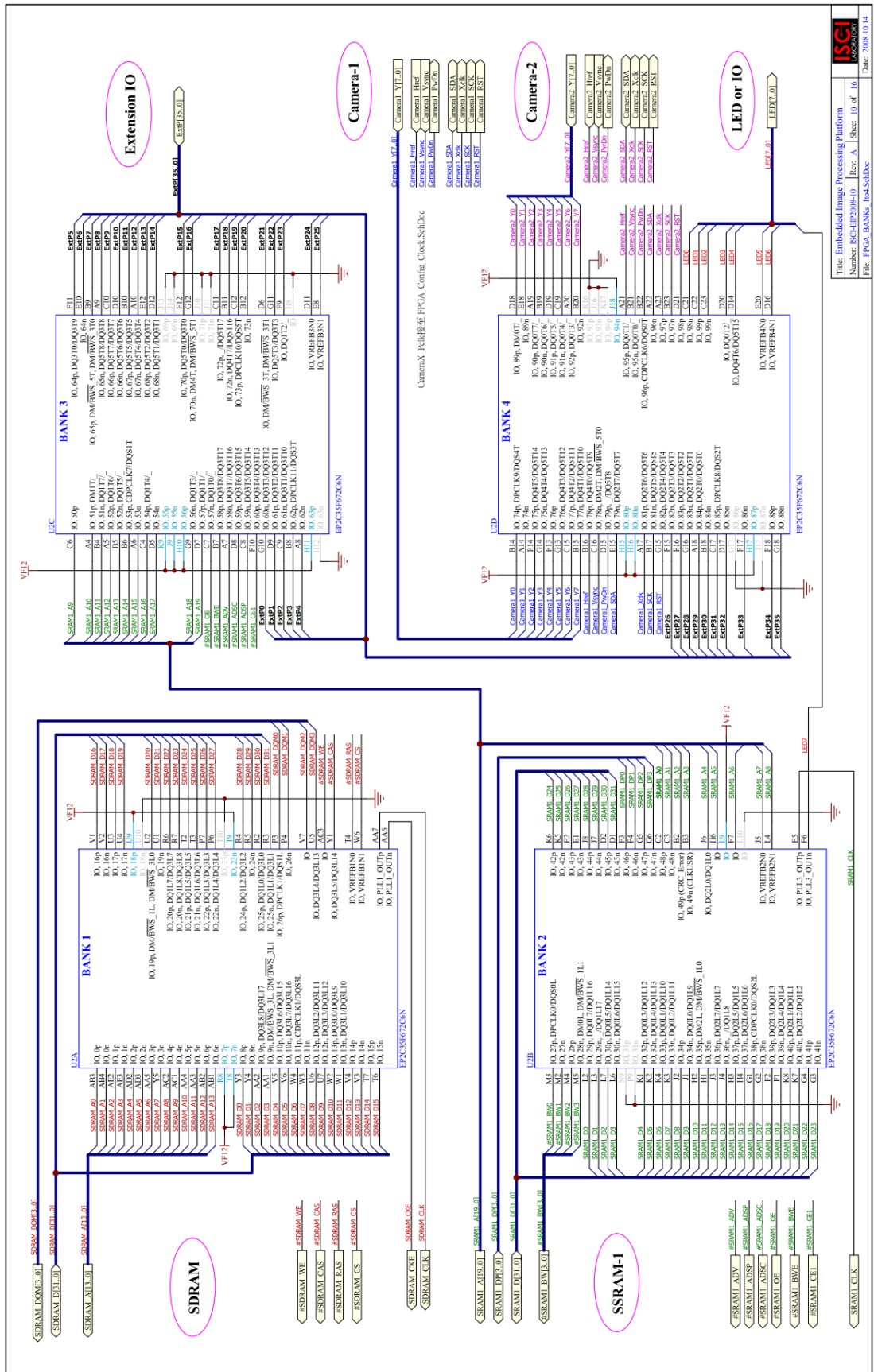


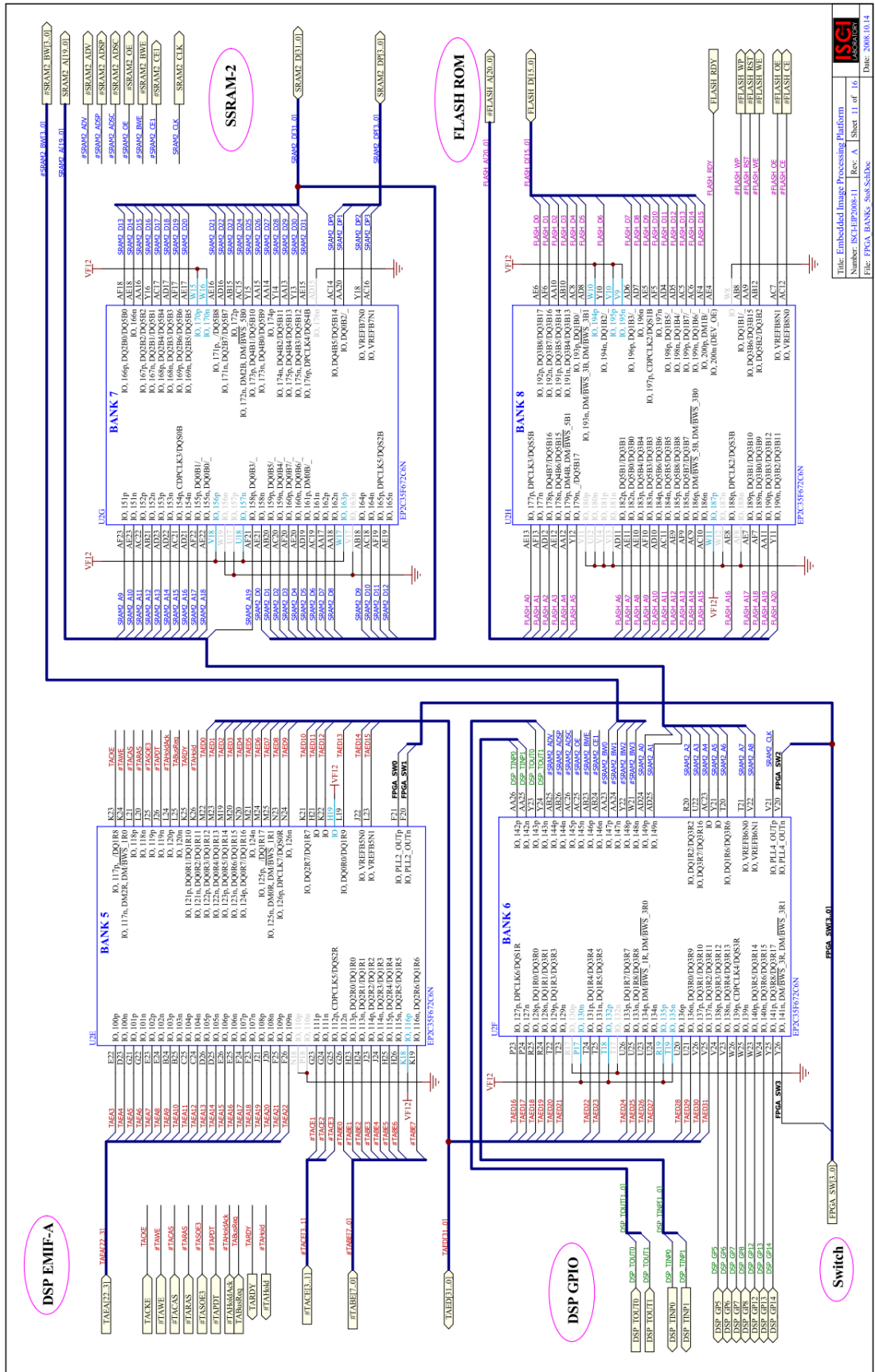




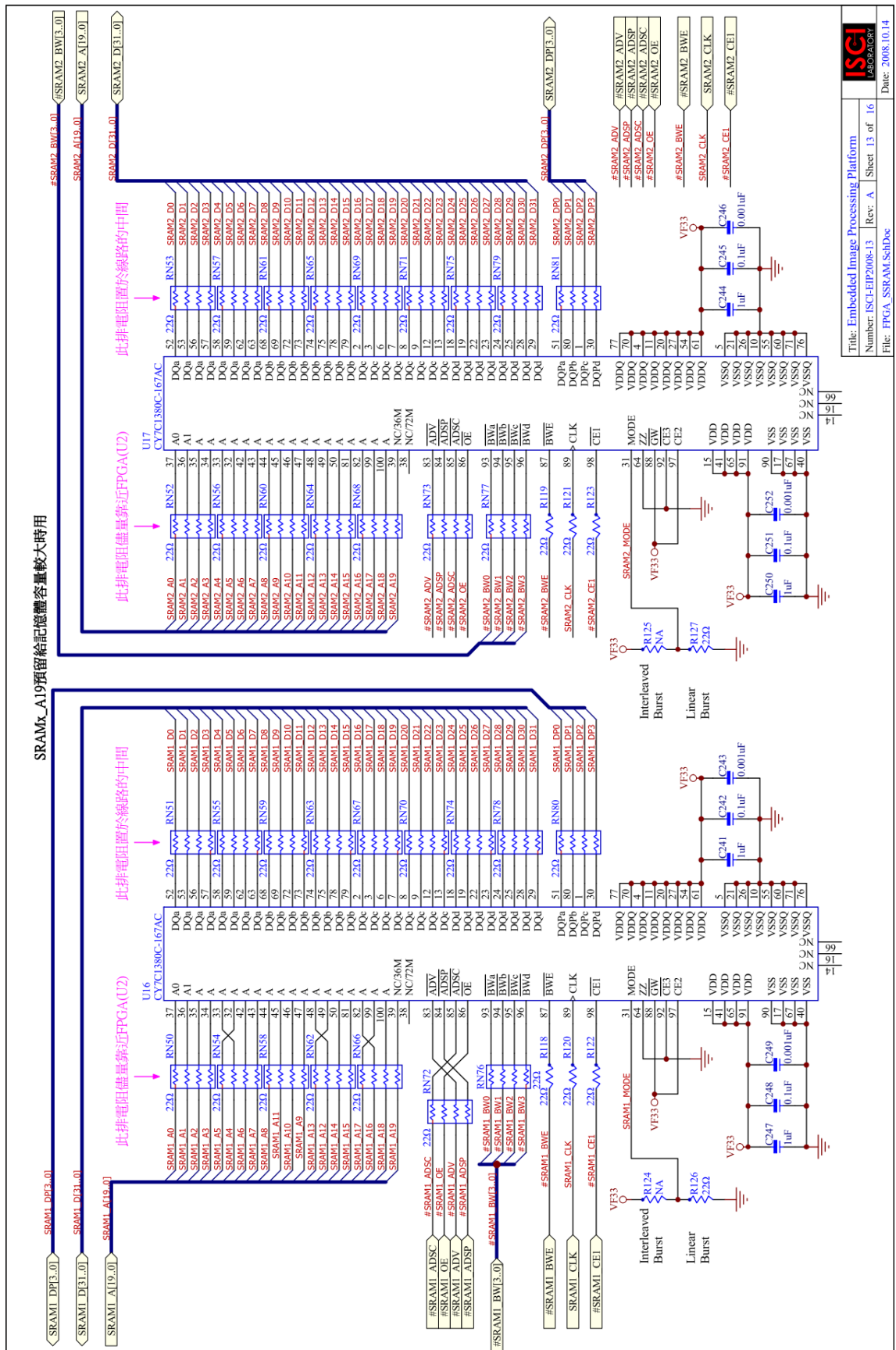


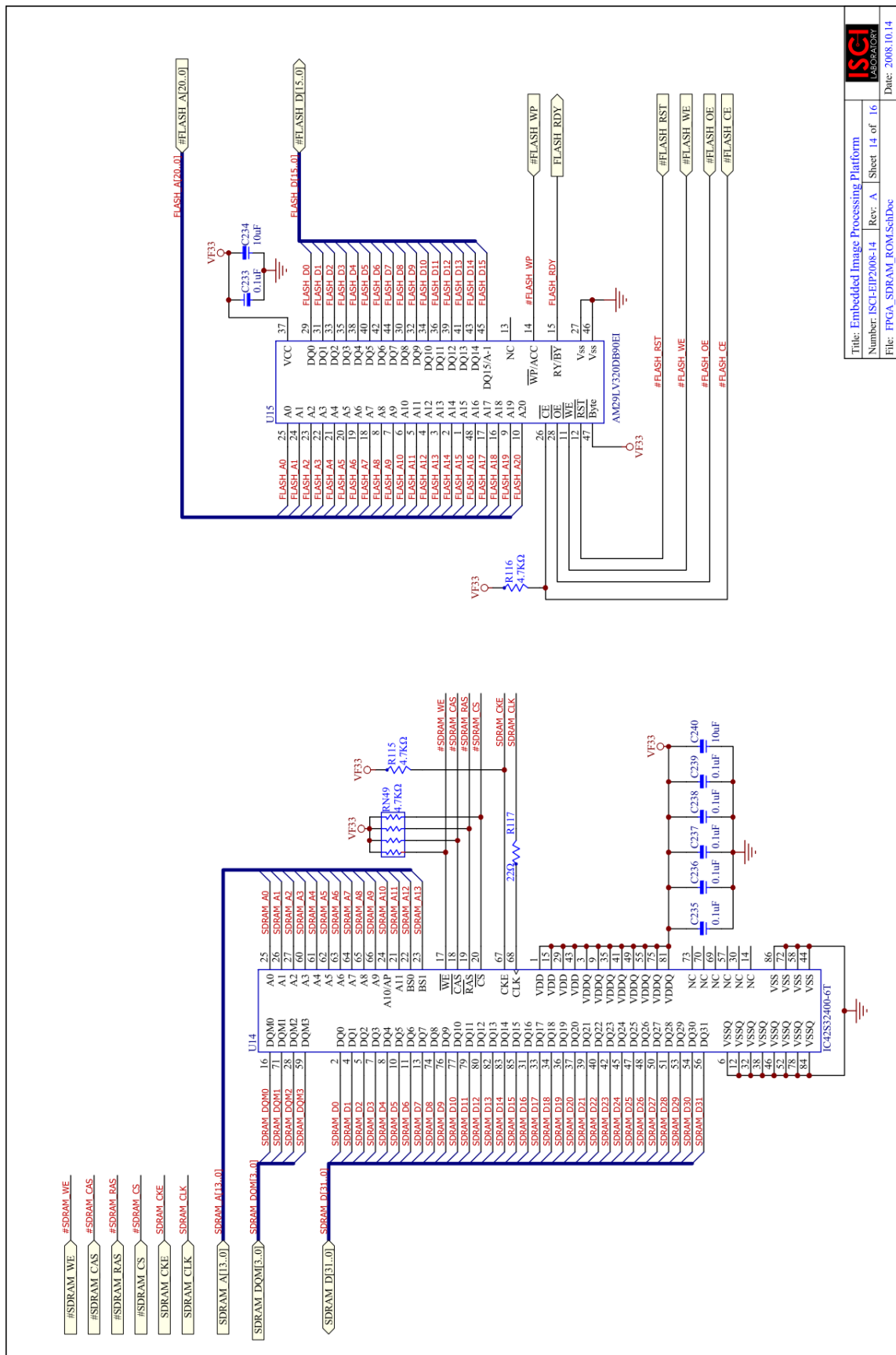
約每2-pin接一個電容



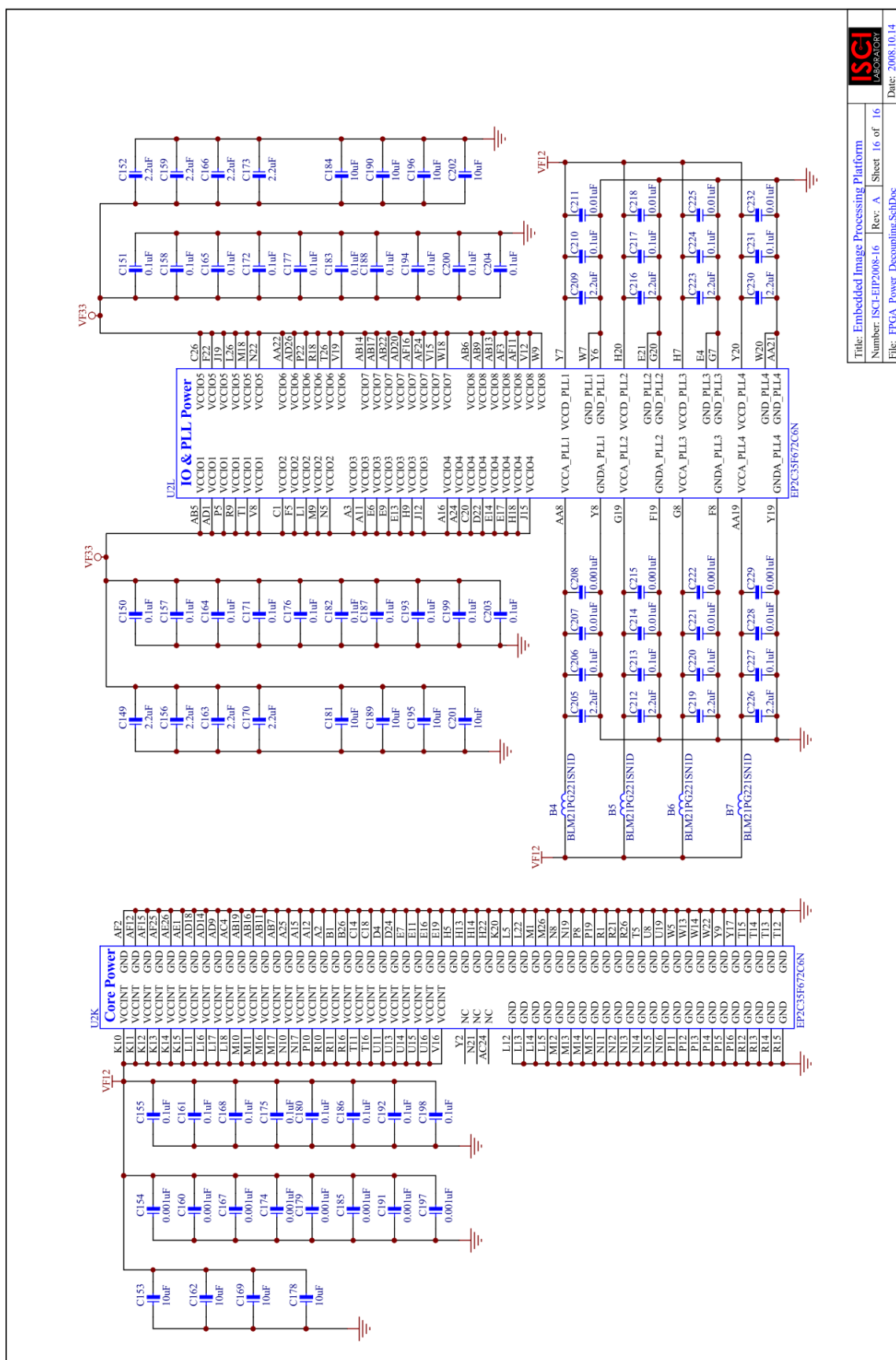












## 附錄二、CMOS 影像感測器(OV7649) 電路圖

