

國立交通大學  
電機與控制工程研究所

碩士論文

在可規劃系統晶片硬體平台實現利用麥克風  
陣列進行真人語音純化與活動偵測



SOPC Implementation of Speech Purification  
and Voice Activity Detection System Using  
Microphone Array

研究生：許晏榮

指導教授：胡竹生博士

中華民國九十四年十月

在可規劃系統晶片硬體平台實現利用麥克風陣列進行真人語音純化與活動偵測

SOPC Implementation of Speech Purification and Voice Activity Detection System Using Microphone Array

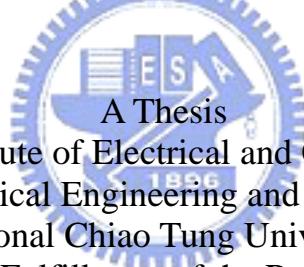
研究 生：許 晏 榮

Student : Yan-Jung Hsuei

指導 教授：胡 竹 生 博士

Advisor : Jwu-Sheng Hu

國立交通大學  
電機與控制工程學系  
碩士論文



Submitted to Institute of Electrical and Control Engineering  
College of Electrical Engineering and Computer Science  
National Chiao Tung University  
in partial Fulfillment of the Requirements  
for the Degree of Master  
in

Electrical and Control Engineering

June 2005

Hsinchu, Taiwan, Republic of China

中華民國九十四年十月

# 在可規劃系統晶片硬體平台實現利用麥克風陣列進行真人語音純化與活動偵測

研究生：許 晏 榮

指導教授：胡 竹 生 博士

國立交通大學電機與控制工程研究所碩士班



本論文針對室內吵雜環境提出一即時抑制噪音源與真人語音活動偵測系統。並提出一個符合實驗室規模、8通道之即時性麥克風陣列訊號處理實驗平臺，且已實作完成。為了讓系統能適應於噪音源的變化與環境的特徵，論文中設計出一適應性空間濾波器並實做完成，其中適應性空間濾波器的啟動由真人語音活動偵測結果決定。論文中之演算法皆實現於可規劃系統晶片硬體平台上。其中真人語音偵測由Nios微處理器完成，而適應性空間濾波器由自行設計的硬體進行加速並透過Avalon Bus和Nios微處理器進行溝通。由於此適應性空間濾波器的階數為彈性化設計，可以根據不同的環境狀況做調整，以達到較好的語音純化效果，由實驗證明，此系統能抑制干擾源的影響，有效提高訊號雜訊比。

# SOPC Implementation of Speech Purification and Voice Activity Detection System Using Microphone Array

Student : Yang-Rong Hsuei                  Advisor : Prof. Jwu-Sheng Hu

Institute of Electrical and Control Engineering

## ABSTRACT

A real-time speech purification and voice activity detection (VAD) system for noisy indoor environment is proposed in this thesis. The system contains a real-time eight channel microphone array signal processing platform. An adaptive spatial filter is also designed on the platform to provide the system with the ability of environmental characteristic and noise adaptation. All the algorithms are realized on a Nios embedded system-on-programmable-chip (SOPC) platform. The VAD algorithm is executed by the Nios processor and the adaptive filter is accelerated by a self-designed hardware, which is a customized peripheral. The communication between the Nios and processor and the customized peripheral is achieved by the Avalon Bus. Since the order of the spatial filter is flexible, the system can be adjusted for superior speech purification result. The experimental results verify that the system can suppress the effect of environmental noise and improve the SNR effectively.

## 誌 謝

對於本論文的完成，首先，感謝指導老師胡竹生教授兩年多來的辛勤指導。在這段日子裡，老師不僅教導我在學術上的知識，更教導我一個研究生應有的研究態度，解決問題的方法，讓我了解到研究生就是應該對自己的研究負責。

而在這些日子，必須感謝實驗室眾多學長、同學和學弟妹的陪伴與協助。感謝介呈與維瀚兩位語音者博班學長的協助，在理論及實務上都提供我很多的幫助。更謝謝立偉及宗敏學長給我許多寶貴的意見與鼓勵。也非常感謝語音組同伴佳興兄幫了我許多忙，一起在車上做實驗與參加比賽的甘苦日子。還有幾位一起奮鬥的同學，士奇、群棋、岑思和鏗元，有了你們的陪伴，讓研究生活添加了許多歡樂。另外也謝謝學弟妹們，朱木、螞蟻、鳥蕙、佩靜、耀賢和恆嘉，謝謝你們的陪伴。

另外，還有眾多好友們。謝謝欣瑋，有了妳在研究所生活最後那一段的陪伴，讓我在交大的日子劃下一個完美的句點。也謝謝跟我同系在專科時的摯友們，逸軒、阿德及恒展，有了你們的陪伴，在我的研究生活以外的世界更添色彩。

最感謝的就是我的家人，父親許明堂先生、母親張素越女士以及妹妹瓊文和淑娟，真摯的感謝你們給我一個無憂無慮的生活與溫暖的家，讓我能夠專心於學業上而無後顧之憂，真的最謝謝你們。謹以本論文向家人獻上最誠摯的謝意。

# 目 錄

摘要 .....	I
ABSTRACT .....	II
誌謝 .....	III
目錄 .....	IV
表列 .....	VII
圖列 .....	VIII
第一章 緒論 .....	1
1.1 研究動機 .....	1
1.2 研究目標 .....	2
1.3 文獻回顧 .....	3
1.4 論文貢獻 .....	5
1.5 論文架構 .....	5
第二章 適應性陣列訊號處理 .....	6
2.1 陣列訊號處理 .....	6
2.1.1 陣列式訊號處理簡介 .....	6
2.1.2 陣列型態：均勻線性陣列 (Uniform Linear Array) .....	8
2.1.3 均勻線性陣列空間響應 .....	9
2.1.4 均勻線性陣列特性 .....	10
2.2 語音活動偵測 (VOICE ACTIVITY DETECTION , VAD) .....	12
2.2.1 VAD 模擬 .....	16
2.3 適應性訊號處理 .....	18
2.3.1 適應性濾波器簡介 .....	18
2.3.2 適應性濾波器處理架構 .....	18
2.3.3 Least-Mean-Square (LMS) Algorithm .....	19
2.3.4 Normalize LMS Algorithm .....	21
2.4 適應性陣列訊號處理 .....	22
2.4.1 適應性陣列訊號處理簡介 .....	22
2.4.2 適應性空間濾波器 : Dahl's Algorithm .....	22
2.5 結合真人語音偵測與適應性陣列訊號處理 .....	25
2.5.1 結合真人語音偵測與適應性陣列訊號處理架構簡介 .....	25
2.5.2 結合真人語音偵測與適應性陣列訊號處理模擬 .....	26
第三章 實驗平台與軟硬體設計實現 .....	28

3.1 實驗平台架構簡介.....	28
3.2 前端麥克風陣列訊號擷取平台.....	29
3.2.1 聲音訊號放大及濾波電路.....	29
3.2.2 類比訊號擷取及轉換電路.....	31
3.2.3 陣列訊號擷取電路板.....	32
3.3 NIOS 嵌入式 SOPC 發展平台 .....	33
3.3.1 SOPC 系統概述與 NIOS 嵌入式 SOPC 發展平台簡介 .....	33
3.3.2 Nios CPU[22] .....	35
3.3.3 Avalon Bus 介面及使用者自訂之客製化周邊模組.....	37
3.3.4 DMA 控制器[22,23] .....	41
3.4 SOPC 單晶片系統軟硬體設計 .....	42
3.4.1 系統功能描述及系統方塊圖 .....	42
3.4.2 系統運算量估測及軟硬體配置 .....	44
3.4.3 系統硬體架構 .....	47
3.4.4 匯流排介面及暫存器檔案模組 .....	51
3.4.4.1 匯流排介面模組.....	51
3.4.4.2 暫存器檔案模組.....	52
3.4.5 資料擷取及流程控制模組 .....	53
3.4.5.1 資料擷取模組.....	53
3.4.5.2 流程控制模組.....	56
3.4.6 按鈕開關偵測與濾波器階數設定模組 .....	57
3.4.7 適應性空間濾波器模組 .....	59
3.4.7.1 八個頻道之 FIR 空間濾波器模組 .....	60
3.4.7.2 訊號能量計算模組.....	62
3.4.7.3 濾波器係數更新模組.....	64
3.4.7.4 適應性空間濾波器模組在 MATLAB 模擬 .....	66
3.4.8 記憶體模組 .....	70
3.4.8.1 欲純化語音記憶體模組(欲消除雜訊記憶體模組).....	71
3.4.8.2 資料記憶體模組.....	72
3.4.8.3 濾波器係數記憶體模組.....	73
3.4.9 Nios 微處理器系統軟體架構.....	74
3.5 實驗平台實際照片 .....	75
<b>第四章 實驗結果與分析 .....</b>	<b>76</b>
4.1 麥克風陣列於室內環境.....	76
4.1.1 空間濾波器結合真人語音活動偵測與訊號雜訊比關係 .....	77
<b>第五章 結論 .....</b>	<b>87</b>
5.1 總結 .....	87

5.2 未来展望 .....	88
<b>REFERENCE.....</b>	<b>89</b>



## 表 列

表 3-1：不同 Nios 軟核的比較-----	36
表 3-2：不同誤差值放大倍率的誤差值收斂情況對照表（256 階濾波器、16 位元動態訊號範圍、32 位元濾波器係數）-----	67
表 3-3：不同誤差值放大倍率的誤差值收斂情況對照表（32 階濾波器、10 位元動態訊號範圍、32 位元濾波器係數）-----	68
表 3-4：不同濾波器係數位元數的誤差值收斂情況對照表（256 階濾波器、 16 位元動態訊號範圍、 $2^{36}$ 誤差值放大倍率）-----	69



## 圖 列

圖 1-1：系統架構簡圖-----	2
圖 2-1：陣列模型-----	7
圖 2-2：均勻線性陣列-----	8
圖 2-3：均勻線性陣列之空間響應-----	10
圖 2-4：Grating Lobe 示意圖-----	11
圖 2-5：VAD 演算法流程圖-----	14
圖 2-6：VAD 模擬結果 N=6，有 Hangover-----	16
圖 2-7：VAD 模擬結果 N=6，無 Hangover-----	17
圖 2-8：VAD 模擬結果 N=0，無 Hangover-----	17
圖 2-9：適應性濾波器處理架構圖-----	18
圖 2-10：LMS 演算法方塊圖-----	20
圖 2-11：Dahl, s Algorithm 訊號擷取架構圖-----	23
圖 2-12：Dahl, s Algorithm 架構圖-----	24
圖 2-13：結合真人語音偵測與適應性陣列訊號處理架構圖-----	25
圖 2-14：人聲摻雜音樂聲-----	26
圖 2-15：真人語音與音樂混和訊號通過 Lower Beamformer 結果-----	27
圖 2-16：真人語音與音樂混和訊號通過 VAD 與 Lower Beamformer 結果-----	27
圖 3-1：語音純化系統架構圖-----	28
圖 3-2：聲音訊號放大及濾波電路架構圖-----	29
圖 3-3：聲音訊號放大及濾波電路圖-----	30
圖 3-4：頻率響應圖-----	30
圖 3-5：類比訊號擷取及轉換電路架構圖-----	31
圖 3-6：麥克風系統電路板-----	31
圖 3-7：SOPC 概念圖-----	33
圖 3-8：嵌入式 SOPC 發展平台-----	35
圖 3-9：SOPC Builder 的使用者介-----	36
圖 3-10：具可變動等待時序之從架構讀取傳輸-----	38
圖 3-11：固定間隔(latency=2)時序之從架構讀取傳輸-----	38
圖 3-12：串流時序之從架構讀取傳輸-----	39
圖 3-13：資料匯流排較小之從裝置客製化周邊-----	40
圖 3-14：資料匯流排較 Nios 大之從裝置客製化周邊-----	41
圖 3-15：搭配 DMA 控制器與客製化周邊之 SOPC 系統單晶片範示-----	42
圖 3-16：SOPC 單晶片系統方塊圖-----	45
圖 3-17：SOPC 單晶片系統硬體架構圖-----	47
圖 3-18：客製化周邊連結 Avalon Bus 之訊號示意圖-----	50
圖 3-19：客製化周邊硬體加速架構圖-----	51

圖 3-20：資料擷取控制器方塊圖-----	54
圖 3-21：資料擷取控制模組之狀態流程圖-----	56
圖 3-22：流程控制器模組之狀態流程圖-----	57
圖 3-23：按鈕開關偵測模組之狀態流程圖-----	58
圖 3-24：濾波器階數設定模組之狀態流程圖-----	58
圖 3-25：適應性空間濾波器模組方塊圖-----	59
圖 3-26：單一麥克風頻道濾波器運算單元-----	60
圖 3-27：8 頻道空間濾波器運算單元-----	61
圖 3-28：訊號能量計算模組運算單元-----	63
圖 3-29：濾波器係數更新模組運算單元 -----	65
圖 3-30：記憶體模組單元方塊圖-----	70
圖 3-31：欲純化語音或雜訊記憶體模組單元 -----	71
圖 3-32：資料記憶體模組單元-----	72
圖 3-33：濾波器係數記憶體模組單元-----	73
圖 3-34：Nios 微處理器系統軟體流程-----	74
圖 3-35：系統軟體之時序分析圖-----	75
圖 3-36：實驗平台實際照片-----	75
圖 4-1：實驗環境實際照片 -----	76
圖 4-2：實驗環境平面關係圖-----	77
圖 4-3：空間濾波器處理前，音樂雜訊-----	78
圖 4-4：空間濾波器處理後，音樂雜訊，階數=64-----	79
圖 4-5：空間濾波器處理前，音樂雜訊-----	79
圖 4-6：空間濾波器處理後，音樂雜訊，階數=128-----	80
圖 4-7：空間濾波器處理前，音樂雜訊-----	80
圖 4-8：空間濾波器處理後，音樂雜訊，階數=256-----	81
圖 4-9：空間濾波器處理前，音樂雜訊無預先錄製-----	82
圖 4-10：空間濾波器處理後，音樂雜訊無預先錄製，階數為 128-----	82
圖 4-11：空間濾波器處理前，音樂雜訊無預先錄製-----	83
圖 4-12：空間濾波器處理後，音樂雜訊無預先錄製，階數為 256-----	83
圖 4-13：空間濾波器處理前，白色雜訊-----	84
圖 4-14：空間濾波器處理後，白色雜訊，階數為 128-----	85
圖 4-15：空間濾波器處理前，白色雜訊-----	85
圖 4-16：空間濾波器處理後，白色雜訊，階數為 256-----	86

# 第一章 緒論

## 1.1 研究動機

環境中的語音訊號干擾源總是存在，例如冷氣機、電腦風扇、喇叭、密閉空間反射等等。當語音訊號遭到干擾時，若用於語音辨識中，辨識率會大為降低，若用於通訊中，通話品質也大受影響。因此若能設計出一語音輸入介面，降低環境中干擾源的影響，達到語音純化的效果，則在生活中將會有很大的應用面。

以往要消除干擾源的影響，往往要知道干擾源的特性，一般的處理方式是由時域、頻域去分析干擾源的特性，如果目前的干擾源在時域、頻域上沒有什麼固定的特徵，如電腦不斷的播放不同的音樂，那麼藉由時域、頻域去消除干擾源的影響，效能往往不理想。在論文中，我們利用麥克風陣列額取訊號在空間上的特徵來做處理，只要能對當時的環境訊號作適應性空間濾波 (Spatial Filter)，則可對不同角度入射的訊號有不同的增益，以降低干擾源對語音訊號的影響，達到提升訊噪比 (SNR) 的作用。除了適應性空間濾波的功能外，我們額外加入真人語音偵測 (Voice Activity Detection, VAD) 的功能，讓系統能依據真人語音的存在與否自動地適應性調整空間濾波器係數。另一方面，由於陣列訊號處理需要大量的乘累加矩陣運算，在 PC 上要實現如此高階且架構特殊的空間濾波器顯得沒有效率。此外，某些應用因空間限制不允許使用體積龐大的 PC (如車內環境)，故必需設計一個體積小而有效率的嵌入式系統。對於將來更希望此語音純化系統可以適用於各種吵雜環境之語音純化，作為語音輸入介面之前端處理，並結合語音辨識器作語者辨識的工作。適用範圍包括

1. 室內吵雜環境之語音辨識前處理
2. 車用語音介面導航系統之語音辨識前處理（車上所處噪音環境尤其複雜）

## 1.2 研究目標

本系統依目標區分為以下部分（如圖 1-1）

1. 選定真人語音偵測及適應性空間濾波器演算法。
2. 自行研製之麥克風陣列平台，包括八個通道之麥克風陣列及八個通道 A/D 擷取電路模組。透過它將八個通道之語音訊號傳輸至 Nios 嵌入式 SOPC(System On Programmable Chip)開發系統平台做語音純化之動作。
3. 實現可調階數之適應性空間濾波器及真人語音活動偵測於 Nios 嵌入式 SOPC 開發系統平台上，進行語音即時性純化處理，並期望輸出之語音資訊達到 16bit 的精準度，未來可做為語音辨識的前處理。

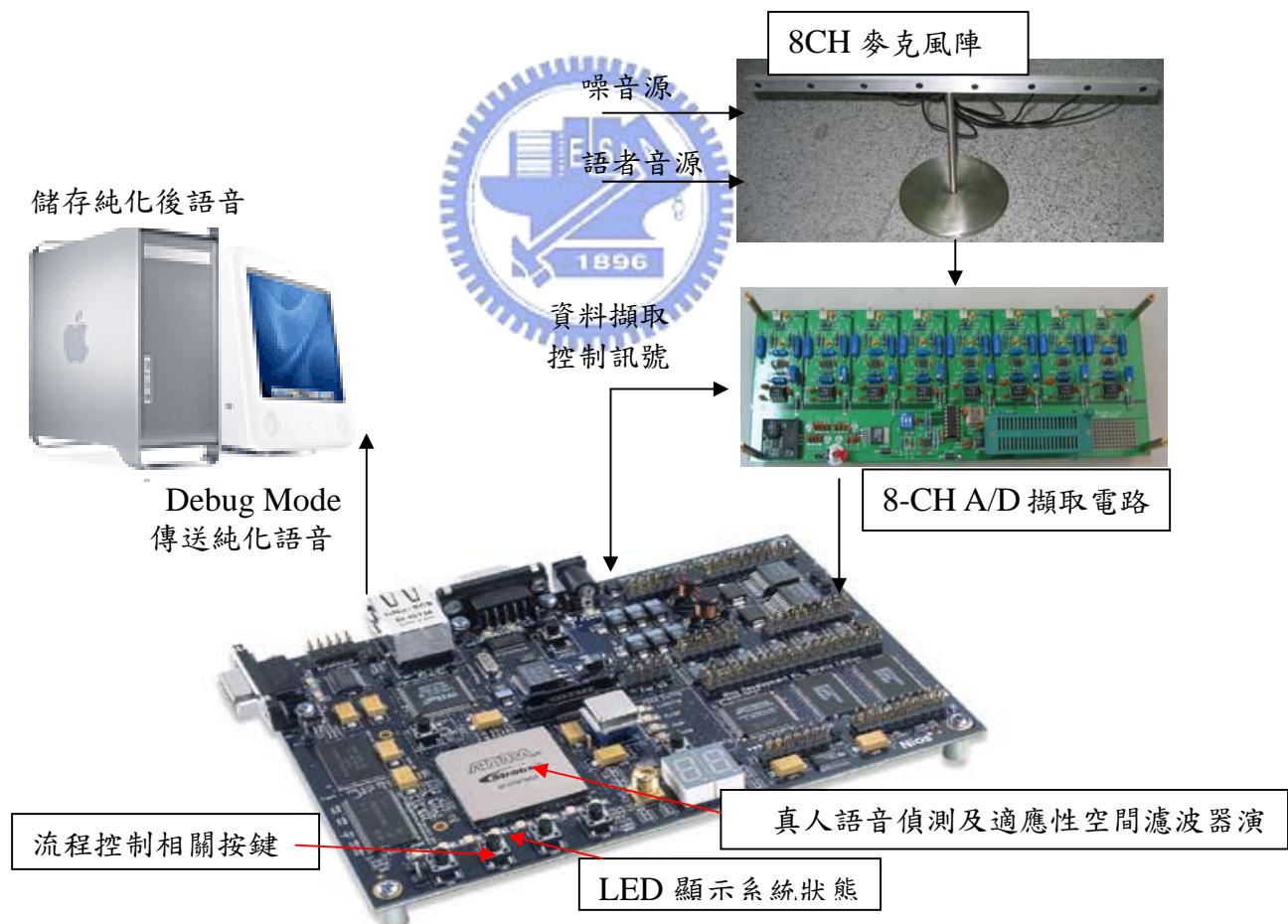


圖 1-1：系統架構簡圖

### 1.3 文獻回顧

陣列訊號處理技術早於第一次世界大戰時被提出並加以利用[1]，當時法國人 Sergent Jean Perrinm 用了兩組感測器，每組感測器由六組次感測器所組成，此發明是用來偵測敵機。之後，陣列訊號處理技術也被用於聲納[2]，陣列望遠鏡（如美國新墨西哥洲沙漠中的特大天線陣列（Very Large Array），它由 27 個碟形天線以 Y 字形分佈）等等。而早年的陣列訊號處理技術皆用於軍事或大型儀器上，直到最近，在電子元件普及與運算能力越來越強大的趨勢下，陣列訊號處理技術也慢慢走向消費性產品話，如麥克風陣列。

麥克風陣列可達到空間濾波的功能，一般而言稱之為 Beamformer[1]，Beamformer 用於麥克風陣列早用於第二次世界大戰[3]，接著慢慢衍生出諸如 Fourier Beamformer[4]、MVDR(Minimum Variance Distortionless Response Beamformer)[5][6]、Robust MVDR[7]、MCMV(Multiply Constrained Minimum Variance Beamformer)[8]、MMSE(Minimum Mean Square Error Beamformer)[9]、MSNR(Maximum SNR)[7]、ML(Maximum Likelihood Beamformer)[7] 等。在各種 Beamformer 中最簡單實現的技術為 Fourier Beamformer，它具有較高的 SNR，但是它需要較大的麥克風陣列才可以達到較好的效果，這是因為越多的麥克風可以形成較尖銳的 beam pattern，進而減少其他非聲源角度之干擾源影響。這樣的缺點會造成為了增加效果而必須一直擴大麥克風陣列的體積，因而提出了一種可以自動消除干擾源的 beamformer—MVDR，它除了可以將所量測出之聲源角度作完整聲音之接收，並且還可讓非聲源角度之聲音接收達到最低。此法跟 Fourier Beamformer 有相同之 SNR，然而卻增加了抑制干擾源的效果。然而，如果接收到的訊號是 coherence 或者是作聲源判斷時產生錯誤(pointing error)，MVDR 這方法所形成的效果將大打折扣，甚至會使得原本要接收之聲源變成完全沒有接收。接下來所提出之 Robust MVDR 便是加入 pseudo noise 以減少 pointing error 的影響。另外還有 MCMV 的方法，這

個方法需先計算出想要接收的角度以及干擾源的角度，Beamformer 的技術針對此聲源收音並且濾除其他方向之雜訊，則此系統將會變得更為實用，而這方面的系統複雜程度以及運算量相當的龐大，如何去利用 Beamformer 和 DOA 定義出想接收度，或者是不想接收的角度，然後產生一個 beam 於想要接收之角度，並且產生 null 於不想接收之角度，此法便可將不想接收的聲源消除，只是此法還需計算其他之角度，如此增加之計算量將是整體系統的負擔。

在國內，麥克風陣列的製作廠商幾乎沒有，而本實驗室 1999 年時，曾經以 Fourier beamformer 為基礎，設計一組 real-time spatial filter and DOA estimation system[10][11]，此系統包含一個 16 channel 的 microphone array 以及對應的 signal conditioner module、sampler module 以及 DSP Module，並能即時估測空間之聲源方向。並在 2004 年設計出以 USB1.1 為介面之語音純化系統[12]。

近年來，已有文獻探討麥克風陣列為基礎的聲音定位 IC[28]以及許多探討適應性濾波器各種電路架構的文獻[29-39]，其不外乎著重於低耗電、小面積或達到高速度的電路架構探討。但尚未發現以麥克風陣列為基礎做適應性空間濾波器結合真人語音偵測之語音純化 IC 相關文獻。而為了因應不同的實際噪音環境並提升陣列訊號處理之效果，往往需要更高階數且具階數彈性之適應性空間濾波器，故本論文將麥克風陣列為基礎之語音純化系統以客製化硬體方式實現於嵌入式發展平台，達到高效能、高擴充性之優勢。

## 1.4 論文貢獻

本論文已實作完成八通道麥克風陣列訊號擷取平台。在演算法上，將真人語音活動偵測（VAD）與空間濾波器（Beamformer）做結合，達到自動適應性調整空間濾波器功能。在硬體實現上，真人語音活動偵測（VAD）利用 Altera 所研製之 32-bit Nios CPU 進行運算；而運算量較大之可調階數空間濾波器則是利用客製化硬體方式進行加速並透過 Avalon Bus 和 Nios CPU 溝通。另外，此空間濾波器只需做小部分修改，即能和其他 32-bit CPU 之 Bus 相容，如 ARM 之 AMBA Bus。由實驗證明，麥克風陣列平台能夠在不同噪音環境下（音樂、人聲或白色雜訊．．．等等）純化語音，增加訊號雜訊比，將來可做為語音辨識之前處理，以增加語音辨識率。

## 1.5 論文架構

本篇論文包含了三個主要的部分，分別是即時性演算法的理論、實驗平臺的架構與實現與即時性演算法的驗證。底下將大致描述三個主要部分的內容：

第二章：將介紹陣列訊號處理概念、適應性訊號處理簡介、語音活動偵測演算法及適應性陣列訊號處理-Dahl's Algorithm。

第三章：介紹實驗平台架構以及演算法之軟硬體架構實現。

第四章：演算法在實驗平台的驗證

第五章：結論

## 第二章 適應性陣列訊號處理

### 2.1 陣列訊號處理

#### 2.1.1 陣列式訊號處理簡介

在傳統數位訊號處理研究中，大多著重於時域訊號的處理技巧，通常先將連續訊號進行取樣，接著通過濾波器以區分訊號中不同的成分，但當原始訊號與雜訊在頻譜上極相似時，一般的時域濾波器很難將原始訊號與雜訊分開，以單一麥克風為例子，若同時接收到兩人說話的聲音，因為兩人聲音頻帶重疊性很高的原因，時域訊號的處理方式很難將兩人聲音分開，因此若需要還原原始訊號，則需要對訊號進行空間取樣以獲得空間資訊[1]。

數個感應器排成特定的形狀，接收來自空間中所傳遞的訊號，並經過訊號處理，此技術稱為陣列訊號處理[1]。在陣列訊號處理領域中，依照其目的不同，大致可以將其研究領域分為兩大類，第一種類的研究著重於估測訊號的數量或在空間中的方位，此類研究一般來說稱為到達角估測（Direction of arrival estimation）。而另一種類的研究則是利用訊號的空間關係，希望能夠對不同方向的訊號作出不同的增益，以達到空間濾波的效果，藉以分離空間中不同方向聲源的訊號，這一類的研究一般稱之為波束形成（Beamforming），也就是一種空間濾波器（Spatial Filter）。

在陣列訊號處理理論中，基於兩個假設

- 窄頻訊號（Narrowband signal）
- 遠場平面波（Far field plane wave）

假設一陣列感應器排置如圖一所示， $s(t)$ 為原始訊號， $n(t)$ 為雜訊

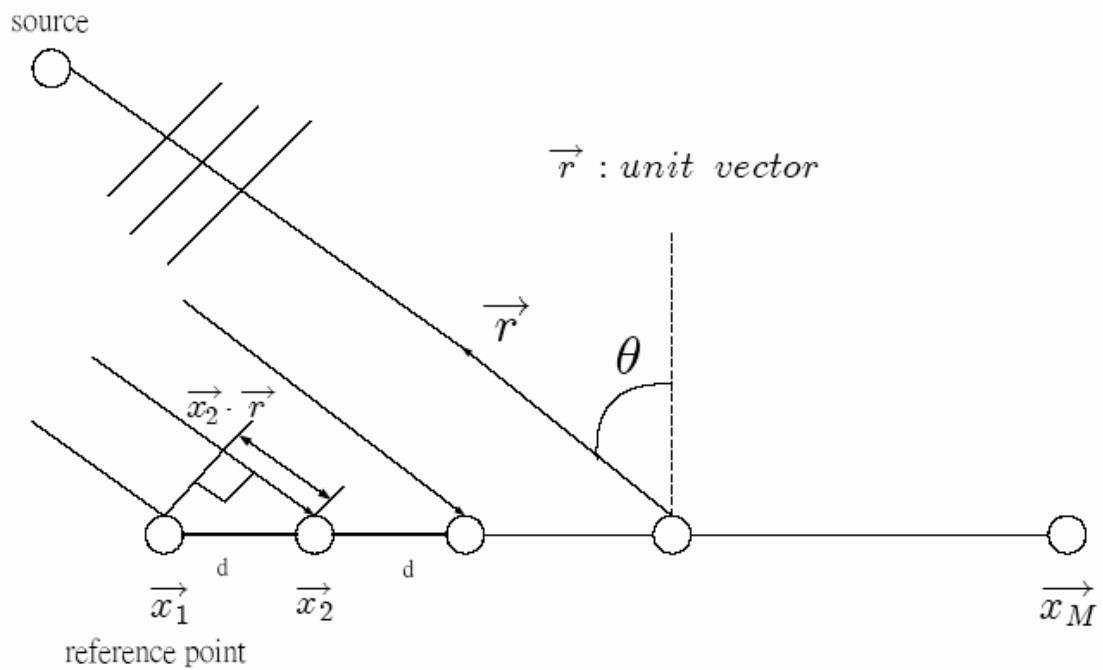


圖 2-1：陣列模型

則  $M$  個感應器輸出可寫成下列向量形式

$$x(t) = \begin{bmatrix} x_1(t) \\ \vdots \\ x_M(t) \end{bmatrix} = \begin{bmatrix} s(t) e^{j\omega_c \frac{\vec{x}_1 \cdot \vec{r}}{c}} \\ \vdots \\ s(t) e^{j\omega_c \frac{\vec{x}_M \cdot \vec{r}}{c}} \end{bmatrix} + \begin{bmatrix} n_1(t) \\ \vdots \\ n_M(t) \end{bmatrix}$$

$$= \begin{bmatrix} e^{jk_c \vec{x}_1 \cdot \vec{r}} \\ \vdots \\ e^{jk_c \vec{x}_M \cdot \vec{r}} \end{bmatrix} s(t) + \begin{bmatrix} n_1(t) \\ \vdots \\ n_M(t) \end{bmatrix} = a(\vec{r}) s(t) + n(t)$$

(2-1)

$k_c = \frac{\omega_c}{c} = \frac{2\pi}{\lambda_c}$   $k_c$  稱為 wavenumber 而  $\lambda_c$  為波長， $c$  為波速

$a(\vec{r})$  稱為 array manifold vector 包含了訊號傳遞到感應器之間時間關係

## 2.1.2 陣列型態：均勻線性陣列 (Uniform Linear Array)

不同的陣列型態會造成不同的空間響應，並會決定陣列的空間解析度，舉例來說，一維的陣列只能解析一維的空間維度，而二維的陣列就可解析二維的空間維度，論文中所實現的陣列型態屬於一維陣列的一部分，因此本章節將介紹屬一維陣列的均勻線性陣列。

均勻線性陣列 (Uniform Linear Array)，是指一組陣列感應器以線性方式排列，並且感應器之間的距離相等，其架構圖如圖 2-2 所示。

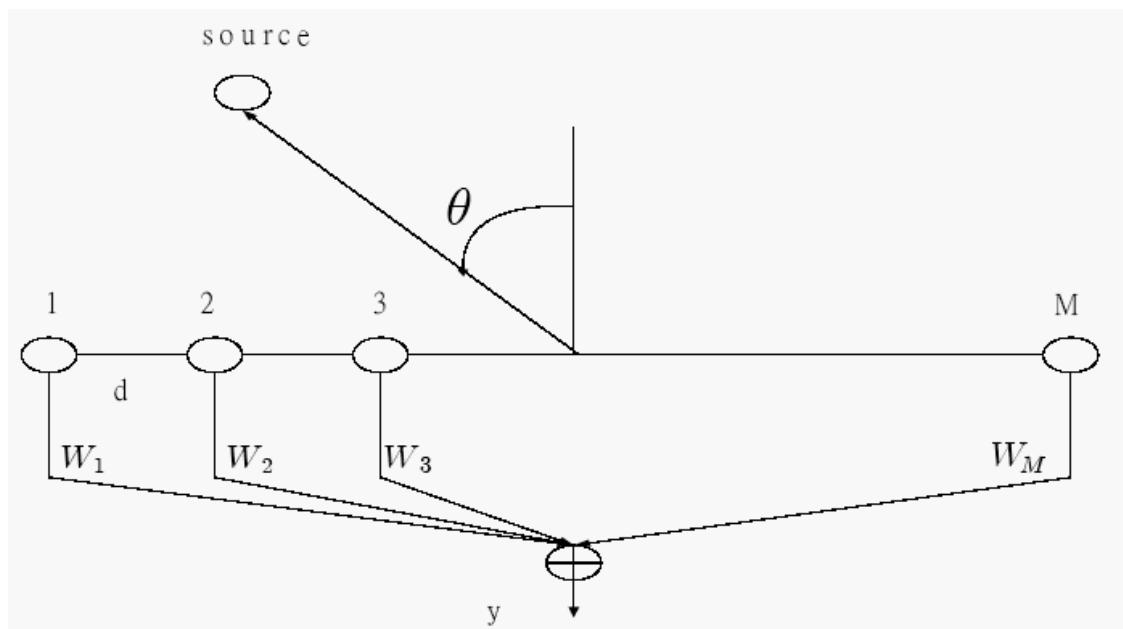


圖 2-2：均勻線性陣列

若以第一個感應器當作參考點，每個感應器對於訊號源相對角度皆為  $\theta$ ，則

第  $M$  個感應器收到的時間為訊號到達第一個感應器後延遲  $\frac{(M-1) \cdot d \cdot \sin \theta}{c}$ ，

因此均勻線性陣列的 Array manifold vector 可寫成如 (2-2) 式，均勻線性陣列的優點是容易實現且公式容易推導，運算量較其它多維陣列型態低，但缺點為只能對一維空間作解析。

$$a(\theta) = \begin{bmatrix} 1 \\ e^{jk_c d \sin \theta} \\ \vdots \\ e^{jk_c (M-1)d \sin \theta} \end{bmatrix} \quad (2-2)$$

### 2.1.3 均匀線性陣列空間響應

均匀線性陣列的架構如圖 2-2 所示，其中  $W$  指的是每個感應器輸出乘上的加權，而空間濾波器（Spatial Filter）指的就是將感應器輸出乘上各自加權值的線性組合，因此均匀線性陣列的總輸出可寫成如下形式：

$$p(\theta) = \sum_{i=1}^M W_i \cdot e^{jk_c(i-1)d \sin \theta} \quad (2-3)$$

此種線性組合的空間濾波器可稱為波束形成（beamforming），若將（2-3）式中的加權值都設為 1，則  $p(\theta)$  可化簡成如下所示：

$$\begin{aligned} p(\theta) &= \sum_{i=1}^M e^{jk_c(i-1)d \sin \theta} = \frac{e^{jk_c M d \sin \theta} - 1}{e^{jk_c d \sin \theta} - 1} \\ &= e^{jk_c \frac{(M-1)d}{2} \sin \theta} \frac{\sin\left(\frac{k_c M d}{2} \sin \theta\right)}{\sin\left(\frac{k_c d}{2} \sin \theta\right)} \end{aligned} \quad (2-4)$$

若將  $p(\theta)$  取 Magnitude 可得其 beampattern，如圖 2-3 所示。

從圖 2-3 可看出，不同角度入射的訊號會有不同的增益，而角度和增益的關係是由陣列的加權值所決定，因此波束形成（beamforming）就可達到空間率波的效果，而在波束形成理論中，就是用適當的方法去計算出加權值，將訊

號作空間濾波，就可得到想要的訊號。

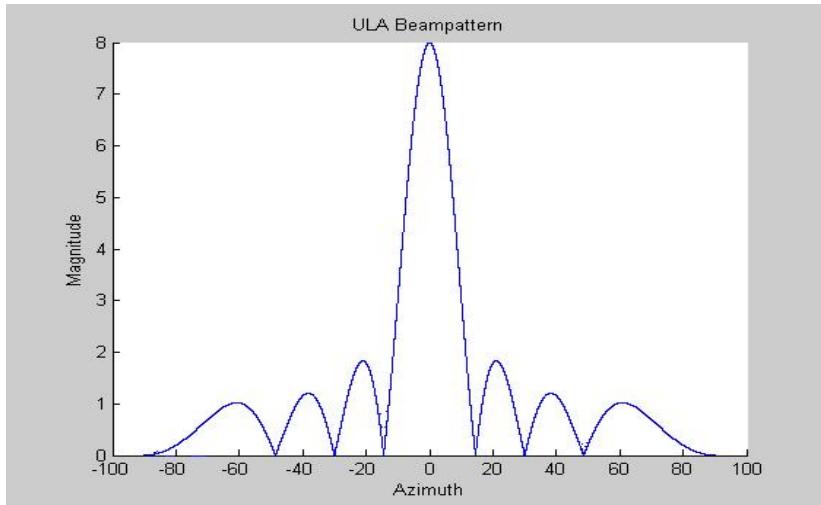


圖 2-3：均勻線性陣列之空間響應 ( $M=8$ , frequency=100Hz,  $d=10$ )



#### 2.1.4 均勻線性陣列特性

和時域濾波器一樣，空間濾波器也會有一些基本的特性，本章節將針對均勻線性陣列，介紹其基本特性[13]。

#### ■ Grating Lobe 問題

將 (2-4) 式取絕對值可得

$$|p(\theta)| = \left| \frac{\sin\left(\frac{k_c M d}{2} \sin \theta\right)}{\sin\left(\frac{k_c d}{2} \sin \theta\right)} \right| \quad (2-5)$$

由 (2-5) 式可看出  $|p(\theta)|$  對  $\sin \theta$  是一週期為  $\lambda_c/d$  的週期性的函式，關係圖如圖 2-4 所示。

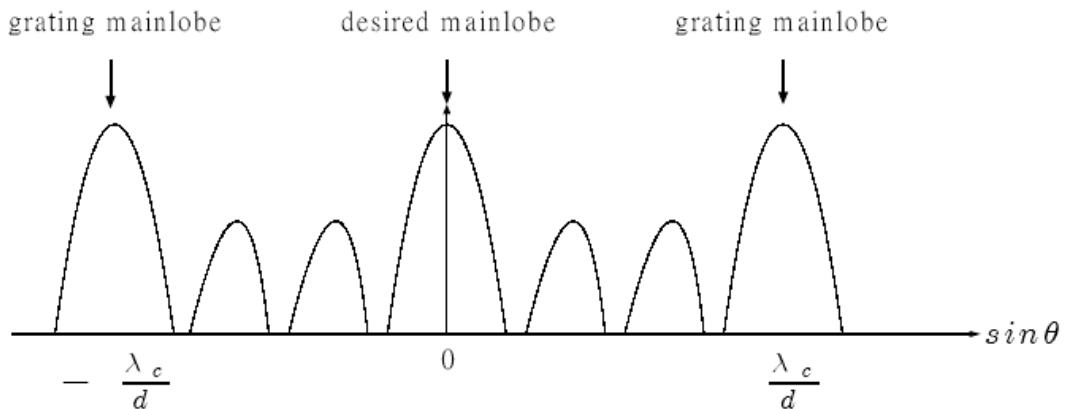


圖 2-4 : Grating Lobe 示意圖

在均勻線性陣列中，預期訊號的角度在 $\pm 90^\circ$ 間，而在這角度之間我們希望 Mainlobe 只會出現一次，如果 Mainlobe 出現兩次以上，則會照成不預期的訊號被接收近來。從圖 2-4 得知，Grating Lobe 發生在  $\sin \theta = \frac{\lambda_c}{d}$  的時候，因此若讓  $\frac{\lambda_c}{d} > 1$ ，則可避免在 $\pm 90^\circ$ 間出現兩個以上的 Mainlobe。而通常我們都會選取  $d = \frac{\lambda_c}{2}$ ，以避免 Grating Lobe 的問題。此現象類似於 Nyquist Sampling Theorem，取樣頻率必須是訊號頻率的兩倍以上。

## 2.2 語音活動偵測 (Voice Activity Detection , VAD)

語音活動偵測是用來判定是否有真人語音，近年來已廣泛用於通訊上達到節省能量耗損的目的。若用於語音辨識方面是屬於語音辨識的前處理，對辨識結果的影響很大，精確的語音活動偵測可降低噪音影響並提高辨識率。傳統的語音活動偵測大多使用語音能量或過零率（zero-crossing rate）等資訊來判別，本節將介紹的語音活動偵測演算法是使用長時間語音資訊（long-term speech information）來判別是否有真人語音[14]。

最常見的判定真人語音資訊為語音能量和過零率，雜訊及氣音的過零率都很高，語音能量都較低。例如，由歐洲電信標準協會（ETSI）所制定用於GSM（Global System for Mobile Communications）系統中的AMR（Adaptive Multi Rate）VAD判定方法就採用了能量、週期、頻譜失真等三種參數來判定[15][16]。另外由國際電信聯盟（ITU）所制定的G.729-VAD採用了全頻帶能量差、低頻帶能量差、頻譜失真和過零率四種參數來判定[17][18]。論文中使用的VAD演算法是使用長時間語音的資訊而非傳統瞬間音框（instantaneous frame）資訊，針對長時間語音資訊，定義出下列定義。

### ■ Long-Term Spectrum Envelope (LTSE)

若  $x(n)$  為一段包含有雜訊的語音訊號，而  $X(k,l)$  代表著  $x(n)$  中第  $l$  個音框第  $k$  個頻率的值，那麼  $N$  階的 LTSE 定義為：

$$LTSE_N(k,l) = \max\{X(k,l+j)\}_{j=-N}^{j=+N} \quad (2-6)$$

其  $LTSE_N(k,l)$  代表的意義為，從第  $l-N$  個音框到第  $l+N$  個音框，這  $2N+1$  個音框分別對其取頻譜絕對值(Amplitude Spectrum)後，在第  $k$  個頻率下，這  $2N+1$  個頻域絕對值內的最大值。而 LTSE 則代表了長時間語音資訊的意義，因為

LTSE 不只是對單一音框取值，而是針對  $2N+1$  個音框取最大值，這樣的好處是不容易忽略某些字頭的子音或是摩擦音。除了 LTSE 外，為了判定是否為真人語音，必須定義另一項定義 LTSD。

## ■ Long-Term Spectral Divergence (LTSD)

LTSD 的定義如 (2-7) 式：

$$\text{LTSD}_N(l) = 10 \log_{10} \left( \frac{1}{NFFT} \sum_{k=0}^{NFFT-1} \frac{LTSE^2(k, l)}{N^2(k)} \right) \quad (2-7)$$

其中 NFFT 代表了做 FFT (Fast Fourier Transform) 的點數，而  $N(k)$  代表了雜訊的頻譜絕對值平均，定義如 (2-8) 式：

$$N_K(k) = \frac{1}{2K+1} \sum_{j=-K}^{j=K} X(k, l+j) \quad (2-8)$$

從 (2-8) 式可看出， $N_K(k)$  代表在第  $k$  個頻率下，第 1 個音框及前後  $K$  個音框的頻譜絕對值平均， $X(k, l)$  和先前定義一樣，代表現階段語音的頻譜絕對值。因此 LTSD 的意義為：現階段長時間語音的頻譜能量與雜訊頻譜能量的比例關係，換句話說判定是否為真人語音是用了現階段語音能量的大小來判定，而此能量大小包含了長時間語音資訊，並非只有單一音框資訊。當 LTSD 大於某個臨界值則判定為真人語音，反之則非真人語音，而此臨界值  $\gamma$  定義如下：

$$\gamma = \begin{cases} \gamma_0 & E \leq E_0 \\ \gamma_0 + \frac{\gamma_1 - \gamma_0}{E_1 - E_0} (E - E_0) & E_0 < E < E_1 \\ \gamma_1 & E \geq E_1 \end{cases} \quad (2-9)$$

其中  $E_0$  和  $E_1$  代表了在最乾淨和最吵雜的情況下，雜訊的能量，而  $E$  是指現階

段雜訊的能量。 $\gamma_0$ 和 $\gamma_1$ 代表在最乾淨和最吵雜的情況下與 LTSD 比較的臨界值，因此 $E_0, E_1, \gamma_0$ 和 $\gamma_1$ 是先設定好的初始值。從（2-9）式可觀察出當

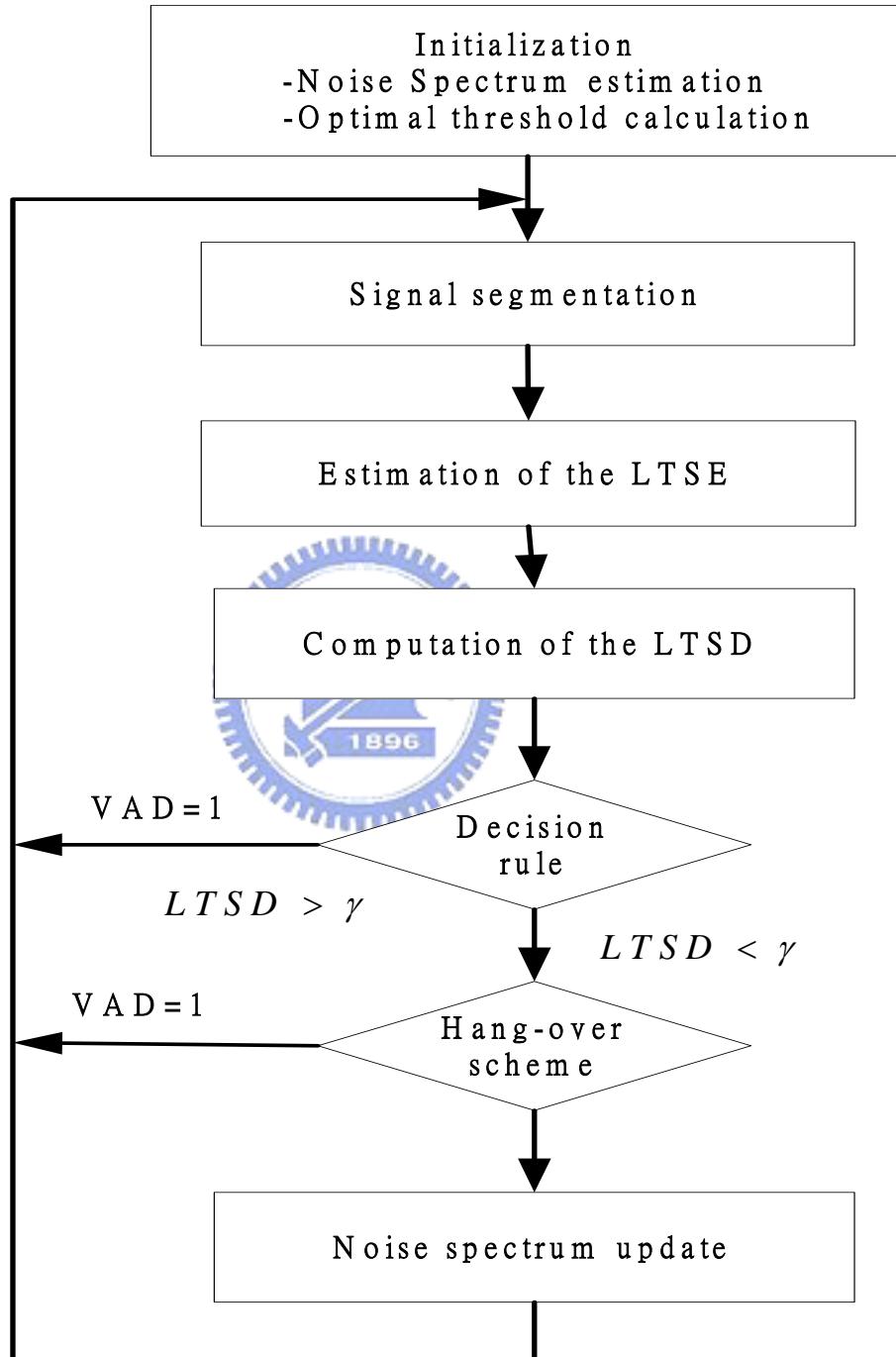


圖 2-5：VAD 演算法流程圖

現階段雜訊能量介於  $E_0$  和  $E_1$  時，則  $\gamma$  會依  $E - E_0$  在  $E_1 - E_0$  所佔的比例，作出  $\gamma_0$  的線性調整。而 VAD 演算法的流程如圖 2-5 所示：

## ■ VAD 演算法流程解說

1. 選定初始值  $E_0, E_1, \gamma_0$  和  $\gamma_1$  並進行環境雜訊初始化估測。
2. 將語音作切割，一個音框為 30ms，而音框和音框之間的交疊為 20ms。也就是每 10ms，就做一次 VAD 判斷。
3. 計算 LTSE 和 LTSD。
4. 將 LTSD 與  $\gamma$  作比較，若  $LTSD > \gamma$  則判定為真人語音，若  $LTSD < \gamma$ ，則經過 Hang-Over 機制。
5. 經過 Hang-Over 機制，若為非真人語音，則更新雜訊頻譜絕對值平均  $N(k)$ 。

Hang-Over 機制是為了延長字母尾音判定為真人語音的機制，因為字母尾音部分通常能量較小，容易被判定為非真人尾音，因此系統中加入 Hang-Over 機制，彌補字母尾音能量小的問題，而當雜訊能量越大時，所需要的延長時間就越多，補償原訊號被雜訊蓋過的部份。另外在更新雜訊頻譜絕對值平均  $N(k)$  方面，並非完全的更新，而是利用了適應性訊號處理的觀念，定義如下：

$$N(k, l) = \alpha N(k, l-1) + (1-\alpha)N(k) \quad (2-10)$$

其中， $k$  代表頻率， $l$  代表音框，從 (2-10) 式可看出， $N(k)$  的更新，除了有現階段  $N(k)$  的資訊外，也包含了上一個音框的  $N(k)$  資訊，而此權重  $\alpha$  可依照環境的雜訊變化速度自行調整。

## 2.2.1 VAD 模擬

本章節將上述 VAD 演算法，用 Matlab 模擬，先以取樣頻率為 16k Hz，用單一麥克風錄製 12 s 的語音，並對其作 VAD 的判定。圖 2-6 展示出有做長時間語音資訊以及 Hangover 的 VAD 結果，也就是 (2-6) 式取 N=6。圖 2-7 展示出有做長時間語音資訊但沒做 Hangover 的 VAD 結果，同樣的 N=6。而圖 2-8 展示出沒做長時間語音資訊及 Hangover 的 VAD 結果，其中 N=0。

圖 2-6、圖 2-7 及圖 2-8 的上半部分展示出 LTSD 與  $\gamma$  的關係，變動較大的為 LTSD，從圖中可觀察出，沒做長時間語音資訊 (N=0) 的 LTSD 變動率較大，這是因為每前進一個音框，LTSD 只由最近的一個音框決定值的大小，而不是由長時間的平均值來決定。比較三張圖可發現，沒做長時間語音資訊 (N=0) 的 VAD 結果（如圖 2-8）很容易將語音的字頭字尾部分忽略，並判定為非真人語音，而有做長時間語音資訊 (N=6) 的 VAD 就不容易忽略字頭字尾部分（如圖 6）。另外，即使有做長時間語音資訊 (N=6) 的 VAD，但沒有 Hangover 的機制，則會忽略字尾的部份（如圖 2-7）。

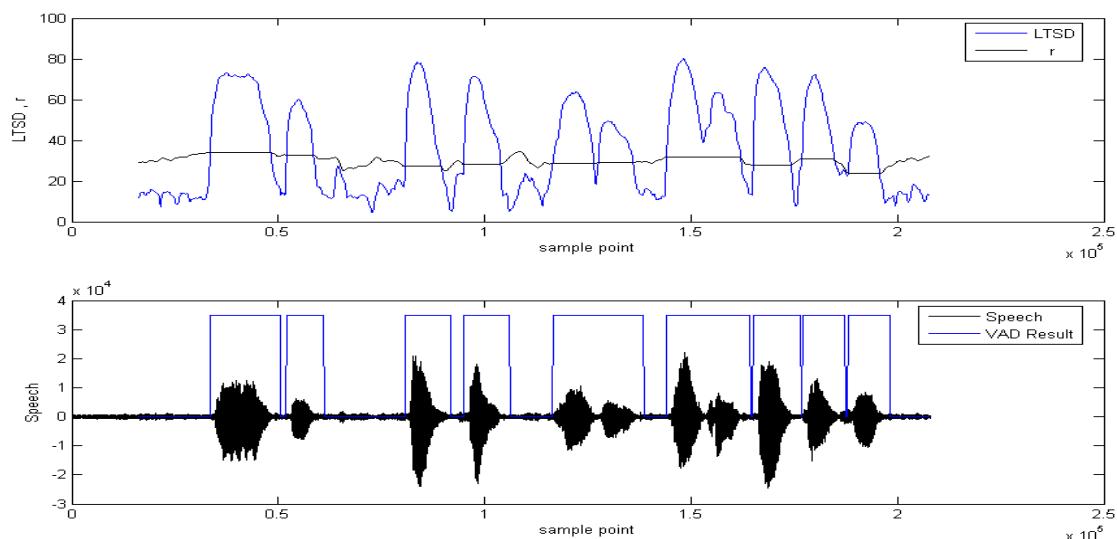


圖 2-6：VAD 模擬結果 N=6，有 Hangover

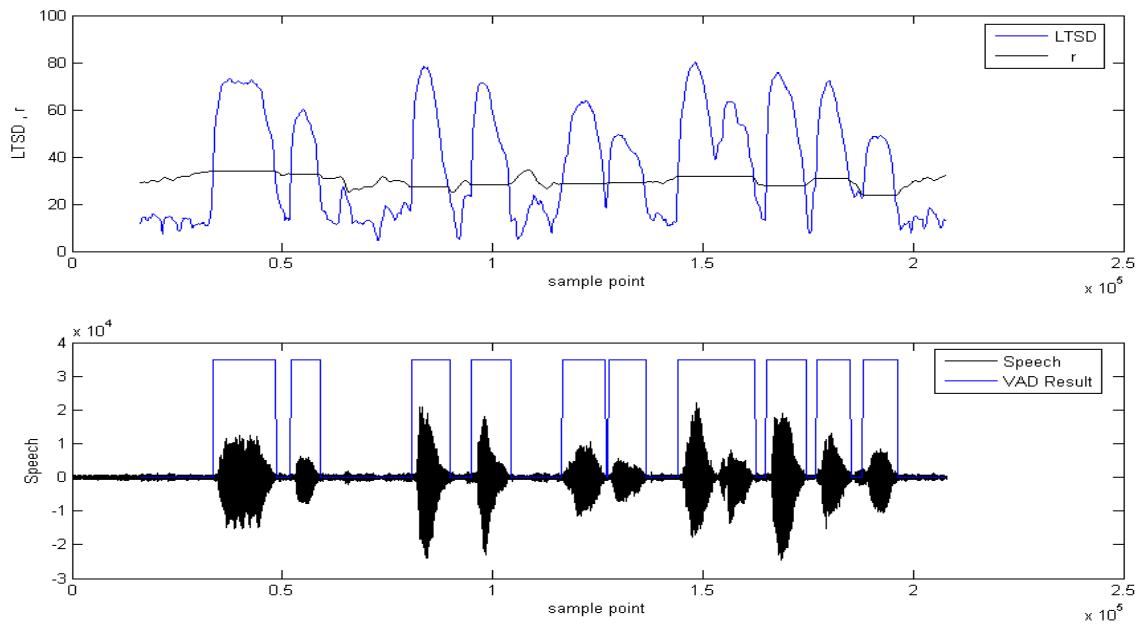


圖 2-7：VAD 模擬結果  $N=6$ ，無 Hangover

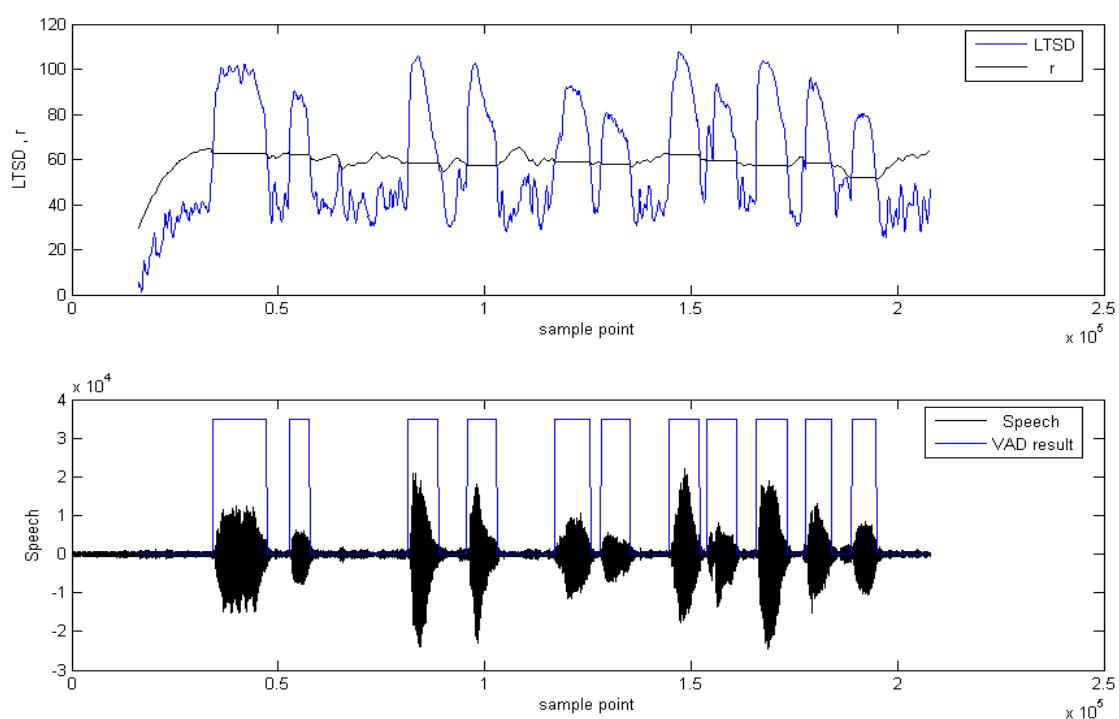


圖 2-8：VAD 模擬結果  $N=0$ ，無 Hangover

## 2.3 適應性訊號處理

### 2.3.1 適應性濾波器簡介

通常而言，濾波器的係數通常設計出來後皆為固定的，並不會自動的變動。而適應性濾波器指的是能根據輸入信號，用訊號處理的技巧來適應性地調整濾波器係數，讓濾波效果更能適應現在環境，以完成某些特定的需要。

### 2.3.2 適應性濾波器處理架構

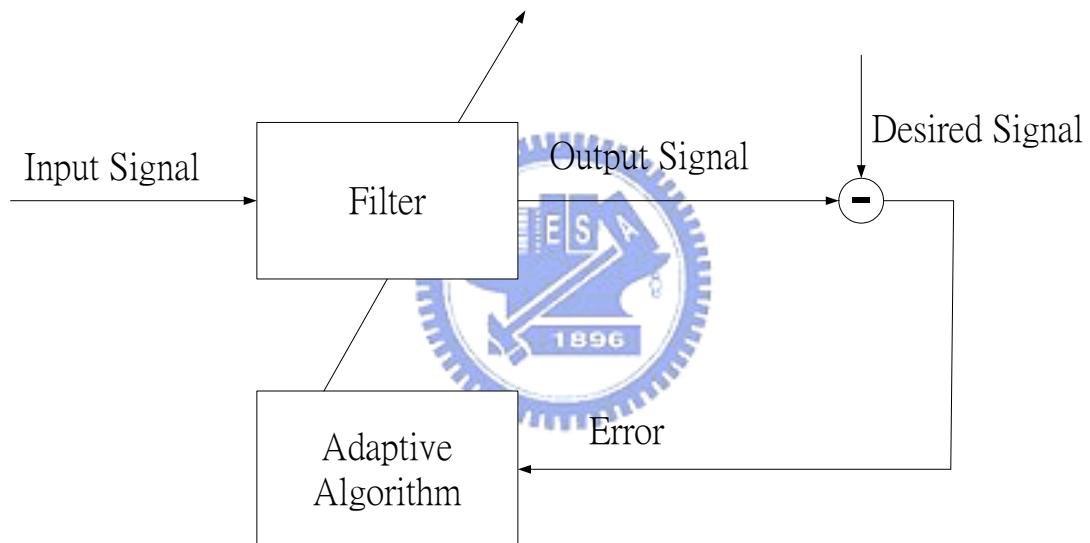


圖 2-9 適應性濾波器處理架構圖

適應性濾波器處理架構圖如圖 2-9 示，當訊號輸入適應性濾波器處理之後，輸出訊號與希望達成的訊號不同，產生誤差訊號，將誤差訊號代入適應性演算法，即可調整適應性濾波器的係數，如此經由誤差訊號及適應性演算法不斷的調整適應性濾波器的係數，係數會不斷的變動，最後達到某個穩定的值，此時系統輸出訊號與希望達成的訊號就會非常接近。

### 2.3.3 Least-Mean-Square (LMS) Algorithm

LMS 演算法指的是，找出一組權重  $W$  以達到誤差平方項最小[19]。其基本架構如圖 2-8 所示。假設希望達成的訊號為 zero-mean，其變異量為  $\sigma_d^2$

$$E\{d\} = 0, \sigma_d^2 = E|d|^2$$

而輸入訊號  $x$  為一組  $M \times 1$  向量，並定義其共異量矩陣和互共異量矩陣

$$R_x = E\{x^* x\}, R_{dx} = E\{dx^*\}$$

因此目標函數如 2-11 式所示

$$J(w) \equiv \min_w E\{d - xw\}^2 = E(d - xw)(d - xw)^* \quad (2-11)$$

2-11 式的意義就是找出一組  $W$  使誤差平方項最小，而  $W$  的找法則需用 Steepest-Descend Method，其標準式如下：

$$(\text{new guess}) = (\text{old guess}) + (\text{a correction term})$$

也就是

$$w_i = w_{i-1} + \mu p, \quad i \geq 0 \quad (2-12)$$

其中 2-12 式意義為從  $w_{i-1}$  出發，並前進  $\mu p$  的距離， $\mu$  為一個比重稱為 stepsize。

而  $p$  的選取必須從 2-11 式開始推導，將 2-11 式展開可得

$$J(w) = \sigma_d^2 - R_{dx}^* w - w^* R_{dx} + w^* R_x w \quad (2-13)$$

為了找一組  $W$  使  $J(w)$  最小，對 2-13 式取  $\nabla_w$  得

$$\nabla_w J(w) = w^* R_x - R_{dx}^* \quad (2-14)$$

因此，為了讓  $w$  往  $J(w)$  最低處的方向與強度前進，我們取

$$p = -[\nabla_w J(w_{i-1})]^* = R_{dx} - R_x w_{i-1} \quad (2-15)$$

2-12 式可寫為

$$w_i = w_{i-1} + \mu [R_{dx} - R_x w_{i-1}] \quad i \geq 0 \quad (2-16)$$

在實做上， $R_{dx}$  和  $R_x$  可用離散形式近似於瞬間值：

$$R_{dx} = d(i)x^*(i) \quad R_x = x^*(i)x(i) \quad (2-17)$$

所以 2-16 是可寫為：

$$w(i) = w(i-1) + \mu x^*(i)[d(i) - x(i)w(i-1)] \quad i \geq 0 \quad (2-18)$$

因此，LMS Algorithm 可整理如下：

$$\text{Filter out : } y(i) = x(i)w(i) \quad (2-18)$$

$$\text{Error function : } e(i) = d(i) - y(i) \quad (2-19)$$

$$\text{Update weight : } w(i) = w(i-1) + \mu x^*(i)e(i) \quad i \geq 0 \quad (2-20)$$

其方塊圖如下所示。

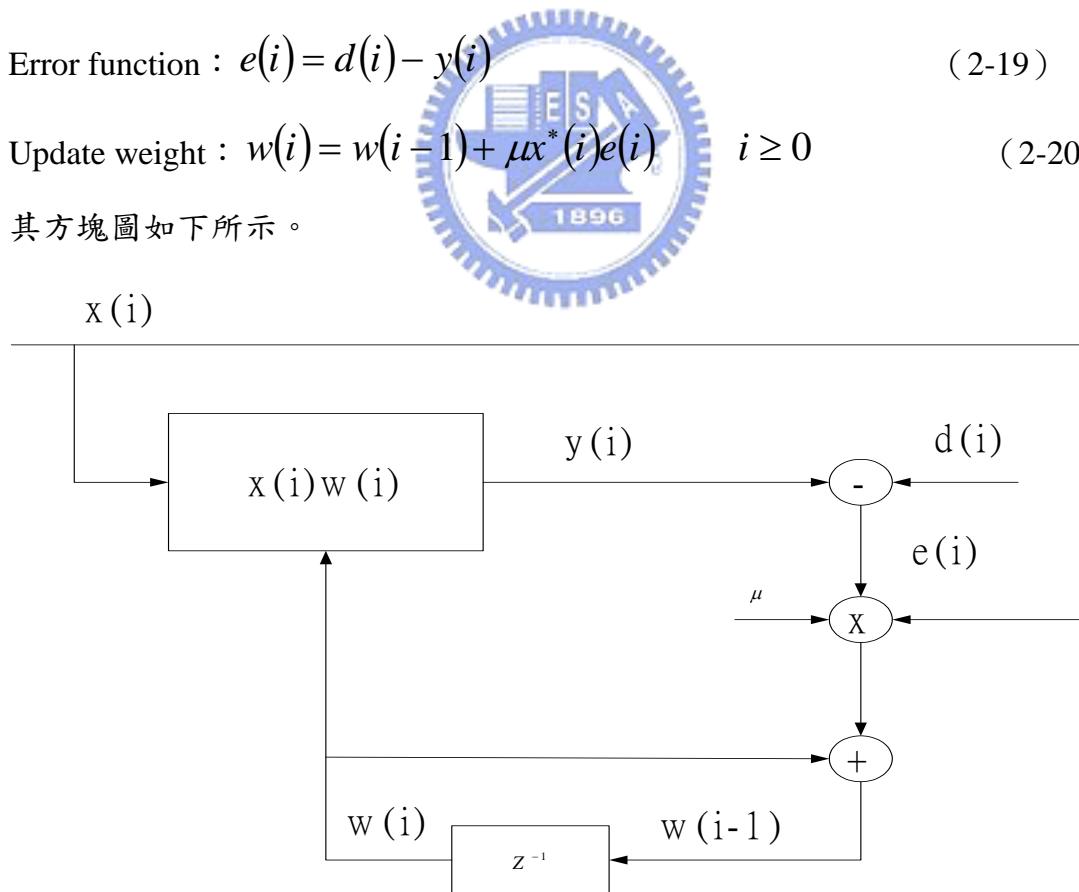


圖 2-10MS 演算法方塊圖

### 2.3.4 Normalize LMS Algorithm

在 LMS 演算法中，為了確保其收斂， $\mu$  的範圍必須為  $0 < \mu < \frac{2}{\lambda_{\max}}$ ， $\lambda_{\max}$

為  $R_x$  的最大特徵值，若所需濾波器階數愈高，則解  $R_x$  的特徵值就愈複雜，以實作方面來講，如此大的運算量會造成龐大的負擔，因此為了簡化其運算量，衍生出另一種演算法，Normalize LMS Algorithm：

$$\text{Filter out} : y(i) = x(i)w(i) \quad (2-21)$$

$$\text{Error function} : e(i) = d(i) - y(i) \quad (2-22)$$

$$\text{Update weight} : w(i) = w(i-1) + \frac{\alpha x(i)w(i)}{\gamma + x^*(i)x(i)} \quad i \geq 0 \quad (2-23)$$

與 LMS 演算法比較，Normalize LMS 演算法只有在更新權重的部分不一樣，原有的  $\mu$  被  $\frac{\alpha}{\gamma + x^*(i)x(i)}$  所取代，其中， $0 < \alpha < 2$ ， $\gamma$  為一個微小的數，目的是確保分母項不為零，如此即可確保 Normalize LMS 演算法收斂，而且如此的運算即不用解  $R_x$  的特徵值，讓運算量降低許多，但在硬體實現上，除法仍會消耗較大的硬體資源。

## 2.4 適應性陣列訊號處理

### 2.4.1 適應性陣列訊號處理簡介

在作陣列訊號處理時，會假設兩條件：

- 窄頻訊號 (Narrowband signal)
- 遠場平面波 (Far field plane wave)

當此兩條件成立時，系統數學式子會簡化許多，空間濾波器的設計也較為簡單，但若感應器陣列所收到的訊號並非遠場平面波，則空間濾波器的設計會變的非常複雜，因此為了簡化空間濾波器的設計方法，則將陣列訊號處理結合了適應性訊號處理的觀念。因為適應性訊號處理只須知道希望達到訊號在空間上的特徵，則可利用演算法去調整適應性濾波器；若將此觀念用於陣列訊號處理，則只須先用感應器陣列得知希望達到訊號的空間特徵，在利用適應性訊號處理演算法來設計「適應性空間濾波器的係數」，於是將適應性觀念用於空間濾波器中。如此，就算感應陣列所收到的訊號並非遠場平面波，但只要知道訊號在空間的特徵，那麼即可利用適應性空間濾波器來專門接收某方向的訊號，並且不斷地作適應性調適，使誤差訊號愈來愈小。

### 2.4.2 適應性空間濾波器：Dahl's Algorithm

本章節將介紹用於麥克風陣列的適應性空間濾波器設計方法，稱作 Dahl's Algorithm。依據適應性訊號處理的觀念，必須先得到希望達到訊號的特性，而 Dahl's Algorithm 的訊號擷取架構圖如圖 2-11 示。

Dahl's Algorithm 的訊號擷取架構圖分兩部分來操作，首先利用 M 個麥克風，在安靜的環境下錄製希望達到的訊號，也就是特定方向的語音訊號，再將此訊號儲存至記憶體。第二步驟就是錄製固定干擾源，也就是希望

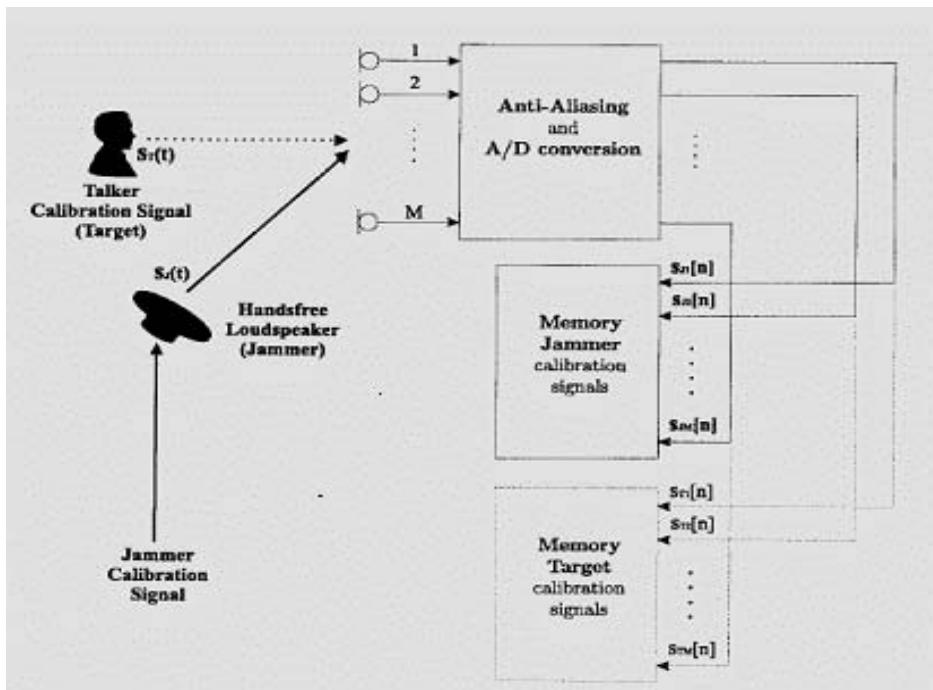


圖 2-11 Dahl's Algorithm 訊號擷取架構圖

空間濾波器濾掉的訊號，並將此固定干擾源儲存至記憶體。舉例來說，若環境中有人的講話聲和喇叭所撥放的音樂聲，則 Dahl's Algorithm 的操作方式為先用麥克風陣列在安靜環境下錄製幾秒鐘人講話的聲音，秒數可自己設定，接下來也在安靜環境下錄製幾秒鐘喇叭所撥放的音樂聲，這樣則完成 Dahl's Algorithm 的預錄部分。

而 Dahl's Algorithm 架構圖如圖 2-12 示，此架構用虛線分為兩部分，上半部分為將麥克風陣列收到的訊號乘上空間濾波器的係數作為輸出，而下半部分則為空間濾波器係數的更新。更新空間濾波器係數方式為，將麥克風陣列即時錄製到的訊號與希望達到的訊號和固定干擾源作相加，相加的結果當作 LMS Algorithm 的輸入，再利用 LMS Algorithm 去調變空間濾波器係數，係數會不斷變動，最後收斂到某一範圍，如此適應性空間濾波器的輸出訊號會和希望達到的訊號誤差最小，也就是說空間濾波器在希望達到訊號的方向增益最高，而固定干擾源的方向增益會被壓低，達到濾除干擾源的

效果。

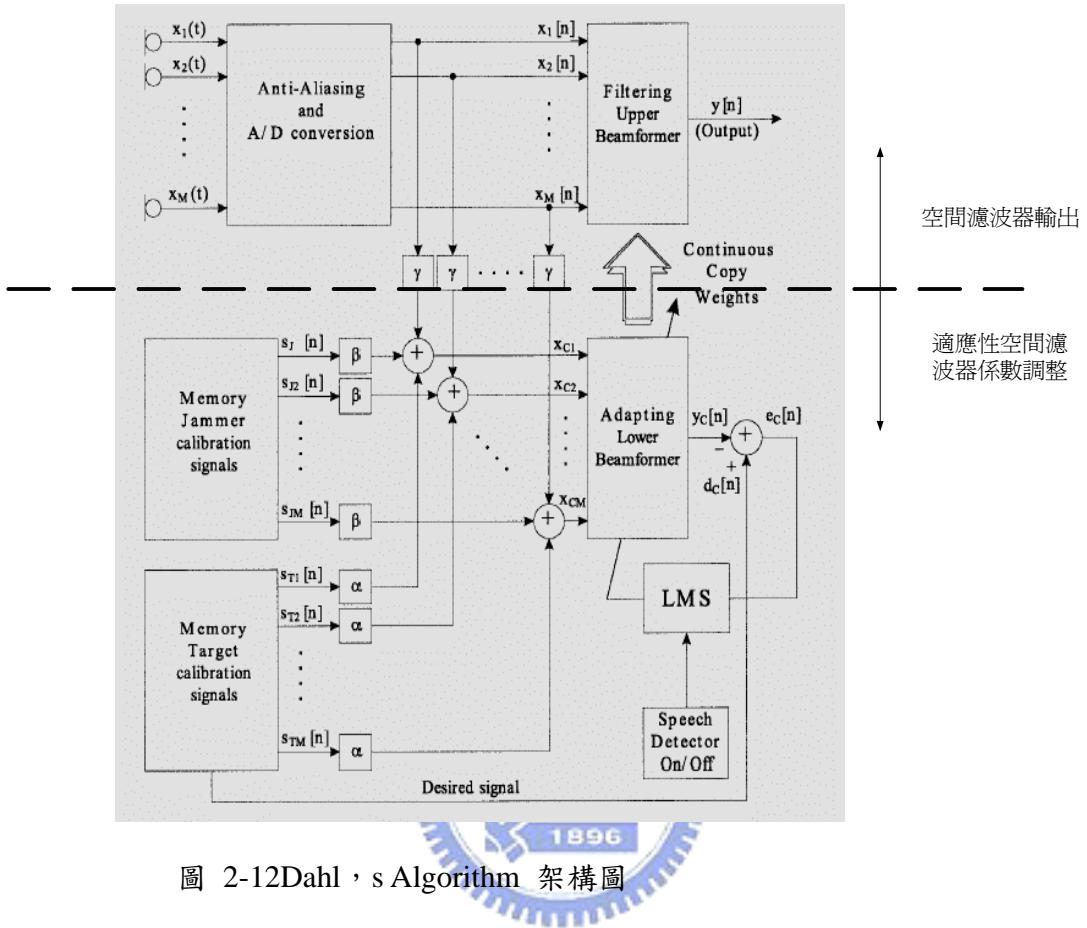


圖 2-12Dahl, s Algorithm 架構圖

在 Dahl's Algorithm 中，適應性空間濾波器調適和空間濾波器輸出這兩部分不可同時進行，這部份由 Speech Detector 判斷進行適應性空間濾波器調適或是空間濾波器輸出。當判斷為語音時則進行空間濾波器輸出進行空間濾波器輸出時；反之若判斷為非語音則進行適應性空間濾波器調適。而當干擾源方向改變或有新的干擾源發生，則必須重新啟動適應性空間濾波器係數調整的功能並關閉空間濾波器輸出，調整出適合新干擾源方向的空間濾波器係數。

## 2.5 結合真人語音偵測與適應性陣列訊號處理

### 2.5.1 結合真人語音偵測與適應性陣列訊號處理架構簡介

本章節將介紹結合真人語音偵測與適應性陣列訊號處理架構，其架構圖如圖 2-13 示：

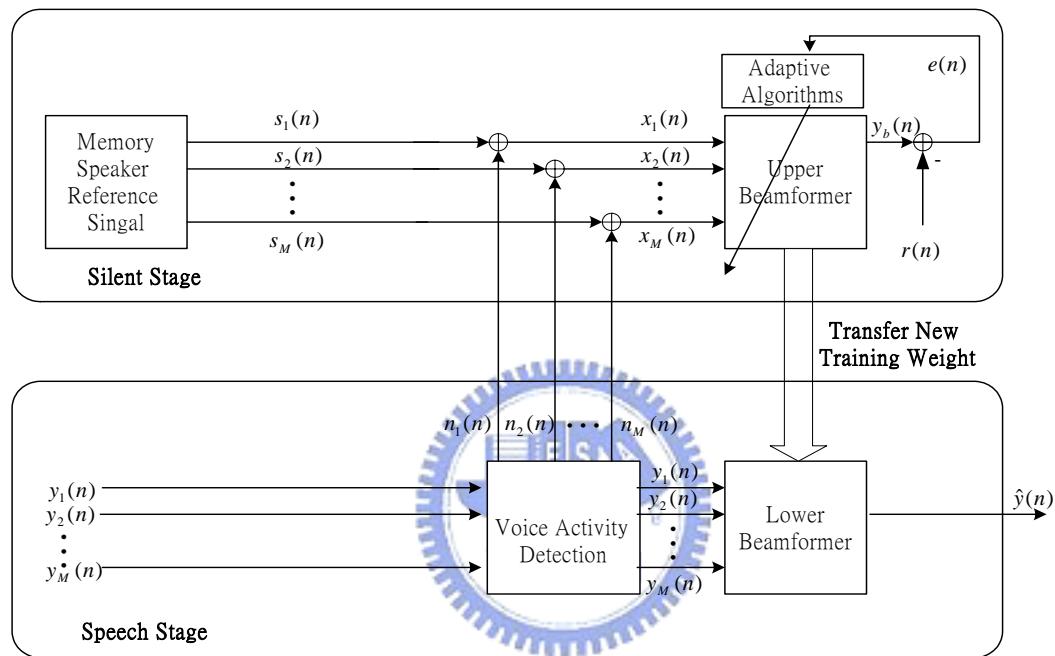


圖 2-13 結合真人語音偵測與適應性陣列訊號處理架構圖

圖 2-13 結合 VAD 與 Dahl's Algorithm 架構圖，此架構可由 VAD 的判定來分為兩部分。第一部分稱為 Silent Stage，第二部分稱為 Speech Stage。當聲音訊號經過 VAD 的判定，若為非真人語音，此時系統會進入 Silent Stage，此時透過 Normalize LMS 的方法適應性地調整 Upper Beamformer 係數。若聲音訊號經過 VAD 的判定為真人語音，此時系統會進入 Speech Stage，Silent Stage 中的適應性訊號調整將會被關閉，並將 Upper Beamformer 係數傳遞給 Lower Beamformer，讓真人語音通過空間濾波器的處理，為純化後之語音。

另外，在系統剛啓動的前 12 個語音訊框，也就是前 140ms，必須先進行外部雜訊環境之初使化估測，此時欲純化之真人語音不能出現，否則將會使

雜訊能量誤判，造成往後的 VAD 判斷錯誤的情形。

### 2.5.2 結合真人語音偵測與適應性陣列訊號處理模擬

本章節將展示將聲音訊號於 2.5.1 節所敘述架構中的模擬結果，圖 2-14 為一真人語音與音樂混合之訊號，圖 2-15 為將此混合訊號通過 Lower Beamformer(濾波器階數為 128)後的結果。圖 2-16 為將此音訊通過 VAD 與 Lower Beamformer 的結果。原混合訊號如圖 2-14 的 SNR 約為 8db，而通過 Lower Beamformer 後之訊號如圖 2-15 約為 17db，故 Lower Beamformer 將混合訊號 SNR 比提高了約 9db。



圖 2-14：真人語音與音樂混合之訊號

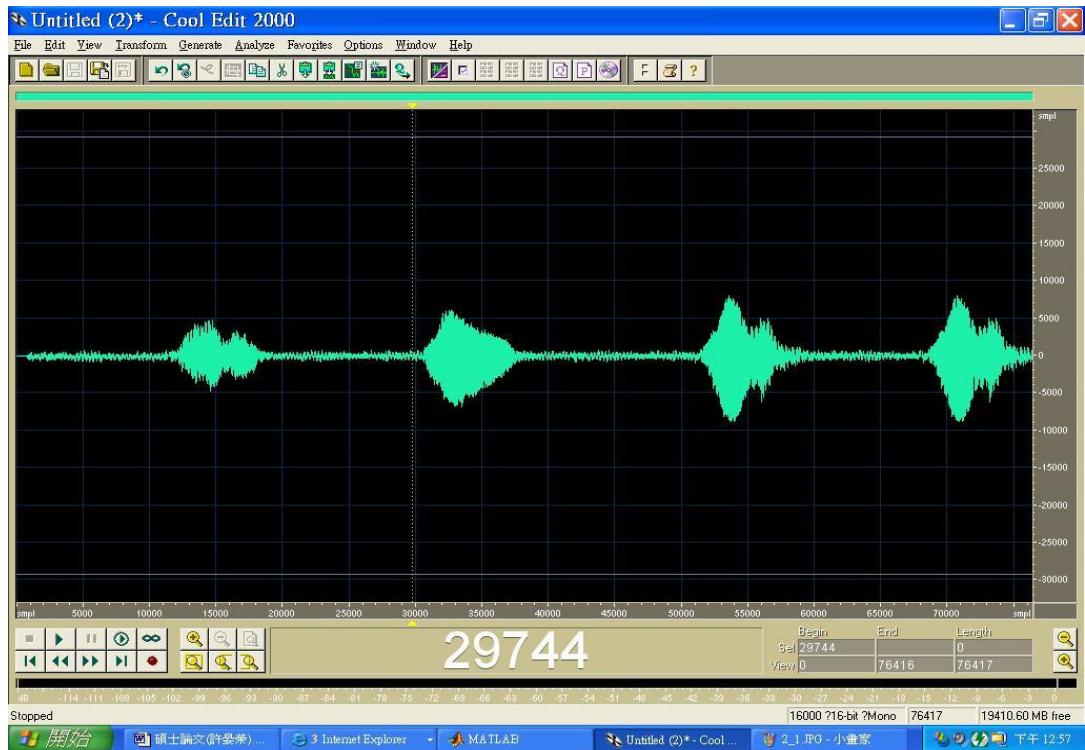


圖 2-15：真人語音與音樂混和訊號通過 Lower Beamformer 結果

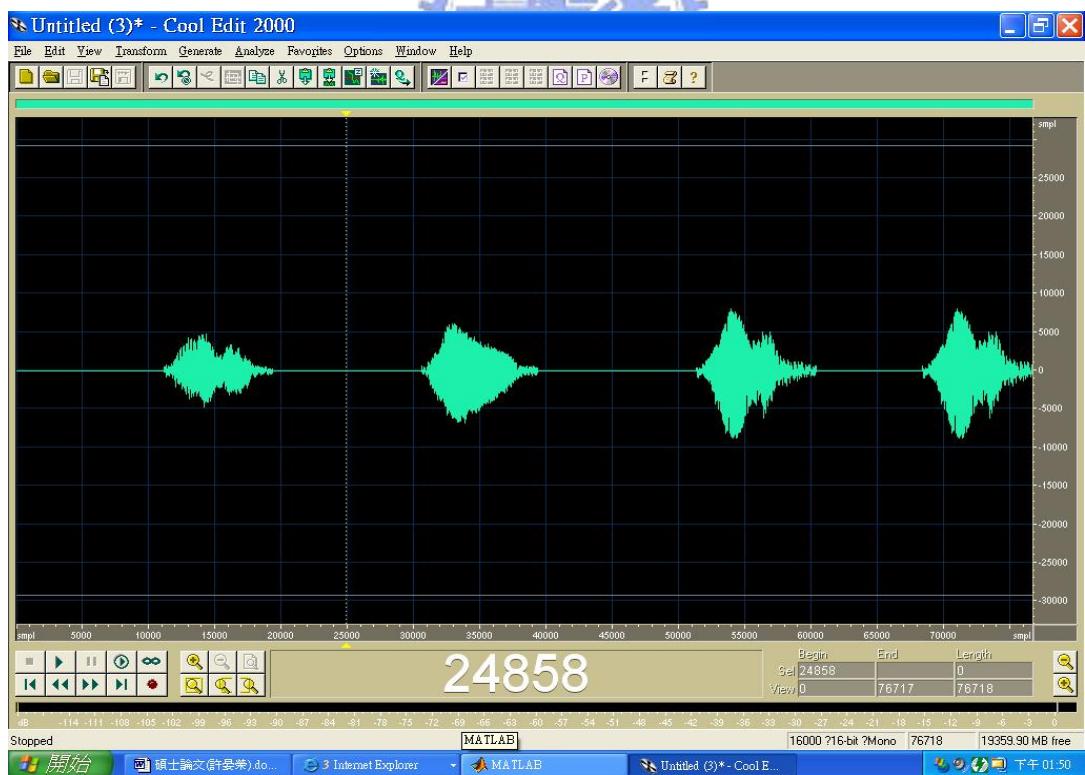


圖 2-16：真人語音與音樂混和訊號通過 VAD 與 Lower Beamformer 結果

## 第三章 實驗平台與軟體設計實現

本章節將針對系統所用到的軟、硬體部份，先簡略介紹實驗平台架構，接著依序由系統硬體架構的設計及運作方式作說明，最後針對 Nios 處理器的系統軟體部分作說明。

### 3.1 實驗平台架構簡介

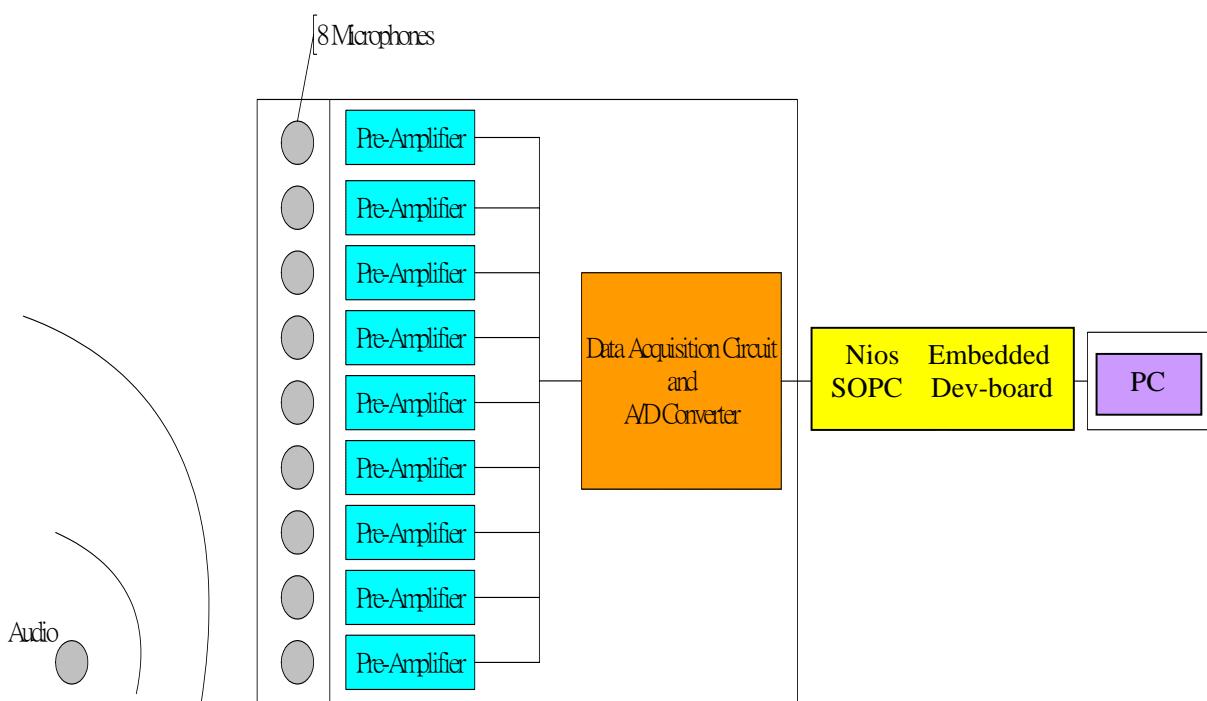


圖 3-1：語音純化系統架構圖

以下介紹實現語音純化系統所需之實驗平台，平台架構方塊圖如圖 3-1 所示。首先，聲音訊號經由 8 通道麥克風陣列所擷取，接著通過訊號放大電路與濾波器並依序將八個頻道之聲音訊號透過 A/D (Analog to Digital) 轉換器轉換為數位形式，數位形式的資料經由 Nios 嵌入式 SOPC 發展平台進行擷取並進行語音純化演算法即時處理，最後經由平台上的 debug 介面傳輸至 PC。

以下分為二大部分介紹實驗平台：

- 前端麥克風陣列訊號擷取平台
- Nios 嵌入式 SOPC 發展平台

## 3.2 前端麥克風陣列訊號擷取平台

### 3.2.1 聲音訊號放大及濾波電路

聲音訊號放大及濾波電路的目的是用來放大麥克風所收到的訊號並濾調高頻及低頻的雜訊，每顆麥克風都有各自的放大及濾波電路，8 組放大及濾波電路構造及功能皆相同，放大及濾波電路架構圖如圖 3-2 所示：

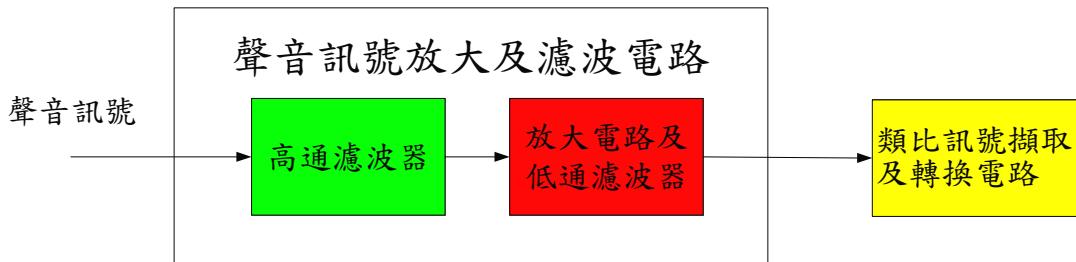


圖 3-2：聲音訊號放大及濾波電路架構圖

聲音訊號經由電容式麥克風轉為電壓訊號後，必須先經過高通濾波器，以濾掉低頻雜訊及直流訊號，而高通濾波器的 3 dB 點設於 80Hz 的地方。訊號經過高通濾波器後還是一個非常小的電壓，因此必須經過一放大電路來放大電壓訊號，以供後端的 A/D 來取樣，而本電路的取樣頻率為 16 kHz，所以必須將訊號通過低通濾波器來避免 Aliasing 問題，而電路中低通濾波器的 3 dB 點設定於 6 kHz。聲音訊號放大及濾波電路圖展示於圖 3-3，本放大電路為一兩級的 OP 放大器，工作電壓介於 5V 和 -5V 之間，並採用負回授的形式，圖 3-3 的放大倍率為 60dB。而電路的頻率響應圖由 P-SPICE 所模擬如圖 3-4 所示。

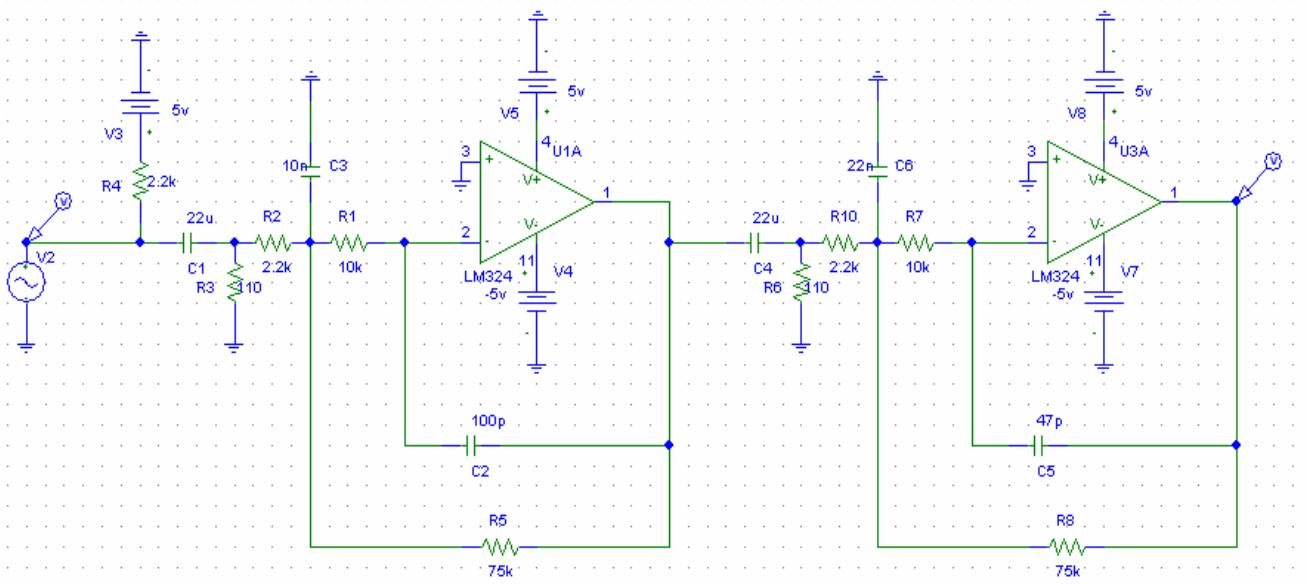


圖 3-3：聲音訊號放大及濾波電路圖

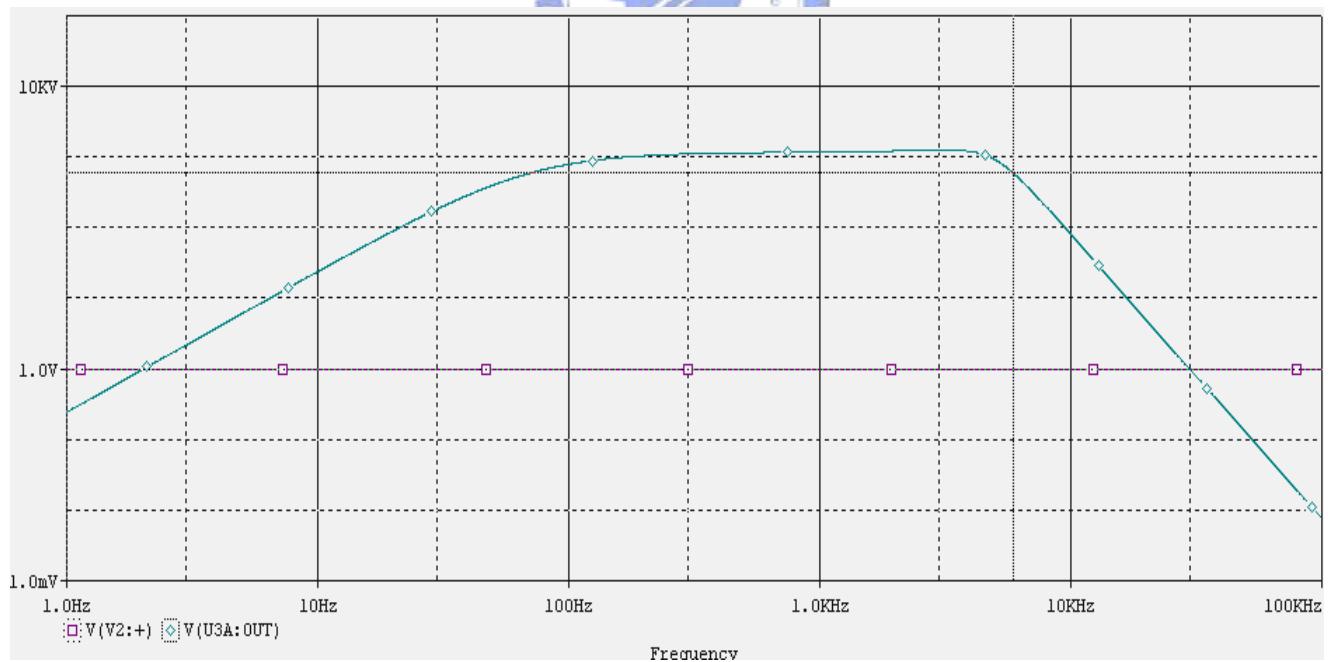


圖 3-4：頻率響應圖

### 3.2.2 類比訊號擷取及轉換電路

類比訊號擷取及轉換電路的目的是將放大倍率後的類比聲音訊號，轉換為數位訊號（16 位元），電路架構圖如圖 3-5 所示，為了節省功率消耗，電路只用了一個類比數位轉換器(A/D Converter)轉換器，因此，8 通道的類比聲音訊號和類比數位轉換器之間需要一個切換器(Switch)，將 8 通道的類比聲音訊號輪流切換給類比數位轉換器做轉換。

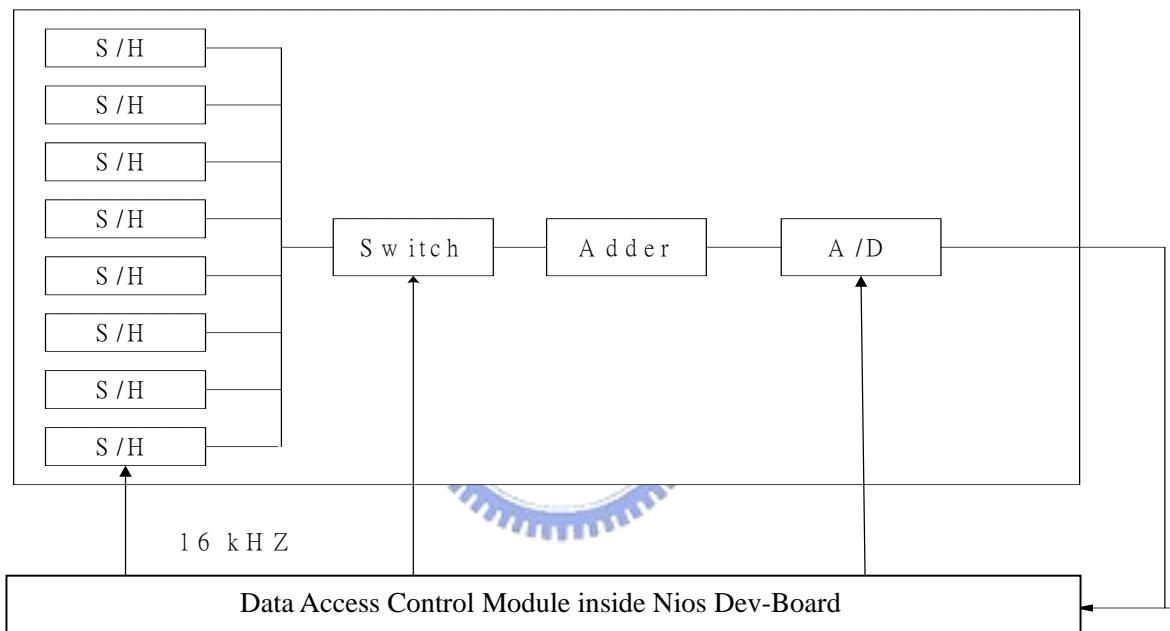


圖 3-5：類比訊號擷取及轉換電路架構圖（S/H：Sample and Hold）

其中類比數位轉換器只能供給正電壓，但取樣／保持電路（S/H Circuit）輸出會含有負電壓，因此切換器和類比數位轉換器之間必須加一個加法器，讓類比數位轉換器的輸入皆在 0 伏以上。

而取樣／保持電路、切換器和類比數位轉換器的時序，由 Nios 嵌入式發展平台的資料存取控制電路所控制，本系統的取樣頻率為 16 kHz，因此取樣／保持電路的工作頻率為 16 kHz，最後類比數位轉換器所轉換出來的 16 位元資料，由 Nios 嵌入式 SOPC 發展平台的資料存取控制模組接收。

### 3.2.3 陣列訊號擷取電路板

電路實作設計時，將聲音訊號放大及濾波電路和類比訊號擷取及轉換電路結合在一起，用 Protel 軟體佈局出其電路圖，其印刷電路板實際照片如圖 3-6 所示：

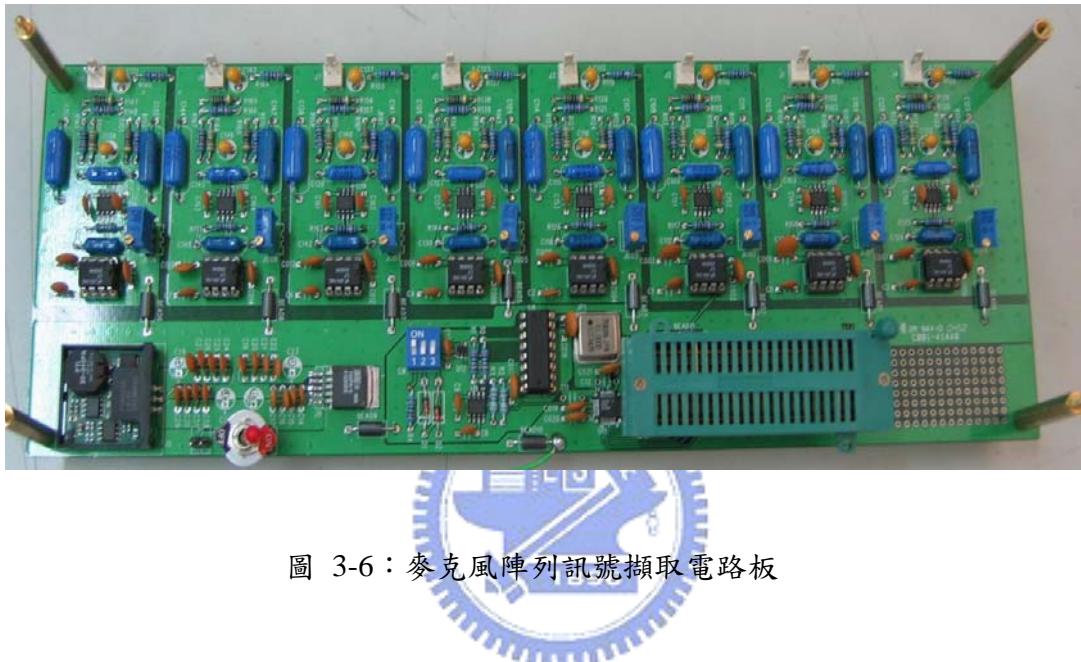


圖 3-6：麥克風陣列訊號擷取電路板

此印刷板電路為一四層板架構，長×寬為 27 公分×10 公分，工作電壓為 5 伏特，其麥克風放大倍率和 A/D 取樣範圍皆為可調。

### 3.3 Nios 嵌入式 SOPC 發展平台

#### 3.3.1 SOPC 系統概述與 Nios 嵌入式 SOPC 發展平台簡介

##### SOPC 系統概述

SOPC(System-On-a-Programmable-Chip)系統是一套由美商 Altera 開發的「可規劃邏輯單晶片系統」。其開發套件中的 SOPC Builder 軟體，可以讓使用者在圖形介面下定義自己所需的 Nios 微處理器，並可將該微處理器透過 Avalon Bus 的機制，結合使用者以 VHDL、Verilog、AHDL 等硬體描述語言所撰寫完成的數位邏輯電路。使得高整合性的系統可在單顆 FPGA 上完成。

如圖 3-7 所示，一個複雜的系統如 System A 所示，可能由微控器 (CPU) 、數位訊號處理器(DSP) 、可規劃邏輯陣列 (FPGA) 、常用的 I/O(例如 GPIO、UART、SPI · · 等等) 、及外部記憶體等等組成。若系統經過適當的規劃，微控器、數位訊號處理器、及常用的 I/O 皆可以整合於單顆 SOPC 如 System B ，無形中降低了許多成本、複雜度及耗電量。

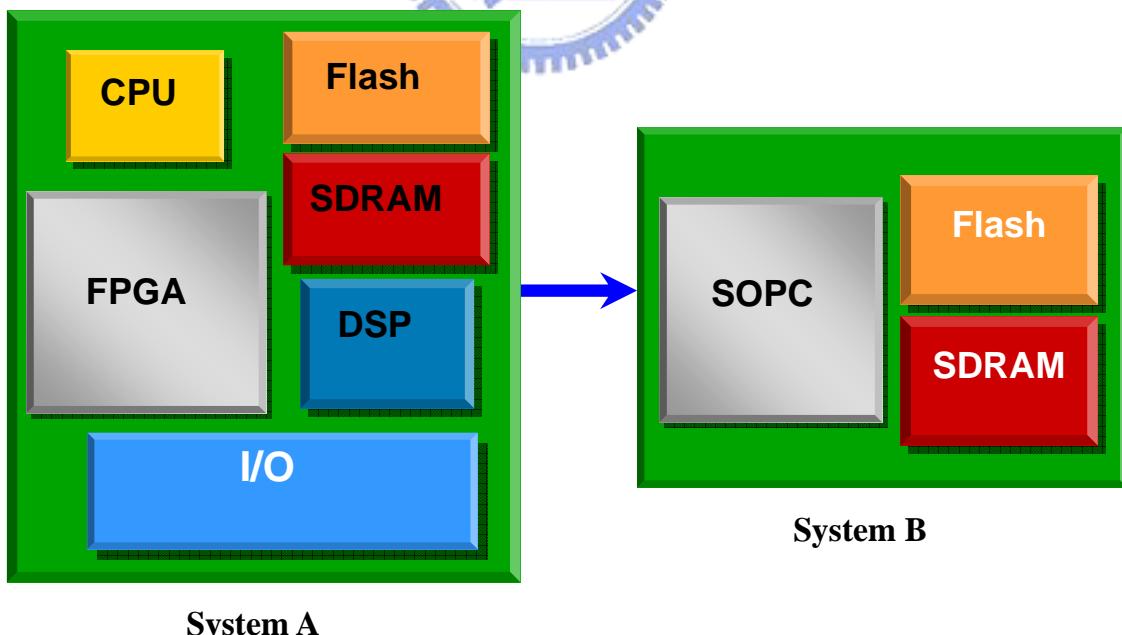


圖 3-7：SOPC 概念圖

## Nios 嵌入式 SOPC 發展平台簡介[27]

這是一套由 Altera 公司針對其產品 Stratix1S40 晶片所計的 SOP 發展電路平台，圖 3-8 為發展平台的外觀圖，除了 Stratix 晶片以外，該發展平台具有以下幾個特點：

- 1Mbyte 的 SRAM 記憶體及 16Mbyte 的 SDRAM 記憶體可供 Nios 處理器運用
- 8Mbytes 的 Flash 記憶體供 Nios 處理器存放硬體規劃程式、C、Assemble 開發的軟體
- Type I CompactFlash cards 接頭
- 10/100 網路實體層及媒體存取控制器(PHY/MAC)及網路接頭(RJ-45)
- 兩組 RS-232 連接頭
- 3.3V、5V 可規劃接腳各 1 組(每組 40 根接腳)，以及 3.3V、5V 電源提供周邊電路使用
- USB Blaster(用於 debug)連結的 JTAG 插槽
- 50 MHz 石英振盪器及外部 CLOCK 輸入接腳
- 用於 debug 的 Mictor 插槽
- 4 個使用者可定義的 push 開關
- 兩組七段顯示器及 8 個使用者定可定義的 LED



而 Stratix1S40 晶片包含下列幾項重要資源

- 41250 Logic Elements
- 428Kbyte On-Chip-Memory
  - ✓ 384 個 M512 RAM Blocks(32 \* 18 bits(max.))
  - ✓ 183 個 M4K RAM Blocks(128 \* 36 bits(max.))
  - ✓ 4 個 M-RAM Blocks(4k \* 144 bits(max.))
- 14 個 DSP Blocks 可規劃為乘累加器、乘法器、加法器等等使用

- 12 個 PLL(Phase Lock Loop)可供使用

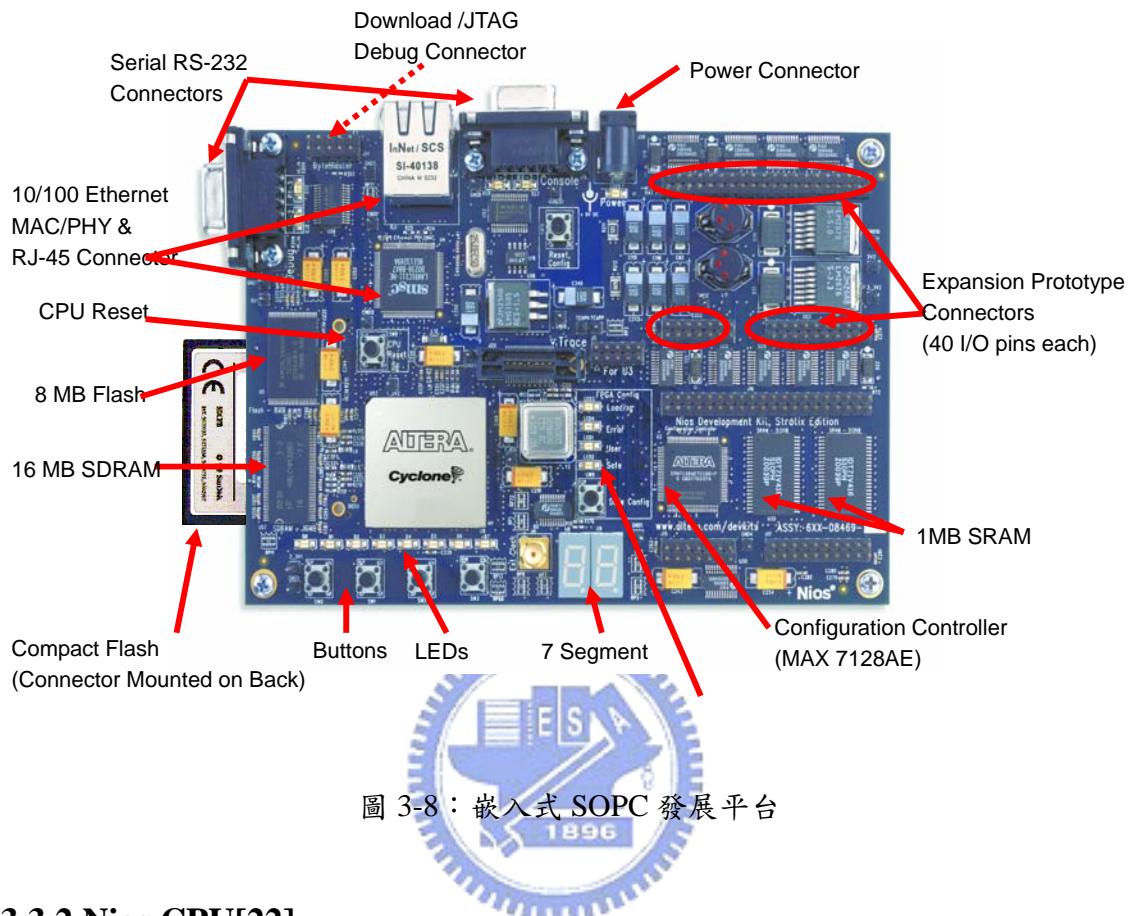


圖 3-8：嵌入式 SOPC 發展平台

### 3.3.2 Nios CPU[22]

Nios 微處理器是由美商 Altera 公司所開發的軟核(SoftCore)微處理器，使用者可以依據系統的運算需求，透過如圖 3-9 所示的 SOPC Builder GNU 開發工具，規劃使用者所需的處理器。其中可以設定成 16 或 32bit 的處理器，另外可以透過 AVALON Bus 的機制，擴充其他週邊如 DMA 控制器、外部記憶體控制器、SPI/UART 通訊控制器、網路通訊模組控制器及使用者自定之週邊等等。

另外，Nios 的軟核可以分為三種如表 3-1 所示，可以根據需求規劃一適合自己系統的微處理器，使用上頗具有彈性。若系統有大量的算術邏輯運算或對於速度上非常要求，則可以規劃為快速 (Fast) 版本。若系統只需很低的算術邏輯運算量，則可以規劃為經濟 (Economy) 版本，節省硬體成本。否

則可以選擇介於兩者的標準版 (Standard)。此外，資料或指令快取記憶體容量以及是否以除法電路加速都可以選擇。

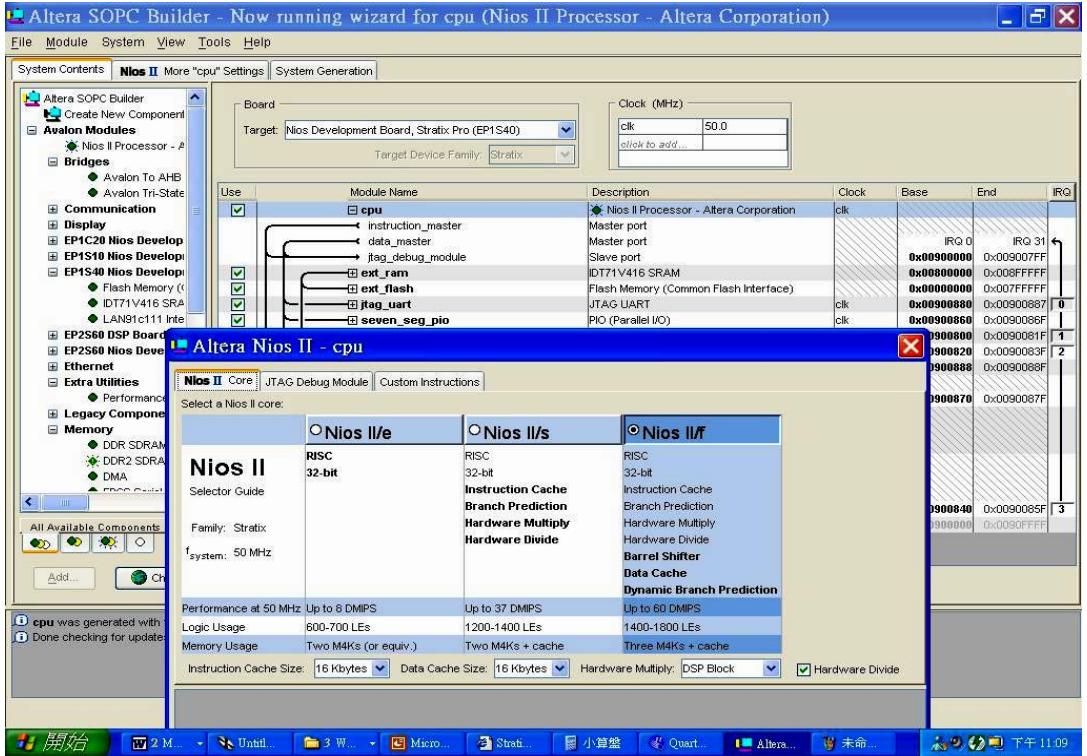


圖 3-9：SOPC Builder 的使用者介面

	Nios II /f Fast	Nios II /s Standard	Nios II /e Economy
<b>Pipeline</b>	<b>6 Stage</b>	<b>5 Stage</b>	<b>None</b>
<b>H/W Multiplier &amp; Barrel Shifter</b>	<b>1 Cycle</b>	<b>3 Cycle</b>	<b>Emulated In Software</b>
<b>Branch Prediction</b>	<b>Dynamic</b>	<b>Static</b>	<b>None</b>
<b>Instruction Cache</b>	<b>Configurable</b>	<b>Configurable</b>	<b>None</b>
<b>Data Cache</b>	<b>Configurable</b>	<b>None</b>	<b>None</b>
<b>Logic Usage (Logic Elements)</b>	<b>1400 - 1800</b>	<b>1200 – 1400</b>	<b>600 – 700</b>

表 3-1：不同 Nios 軟核的比較

### 3.3.3 Avalon Bus 介面及使用者自訂之客製化周邊模組

#### Avalon Bus 介面[26]

在 SOPC 嵌入式系統單晶片內，Nios 微處理器利用 Avalon Bus 連結其所有周邊裝置，其支援多樣的主從周邊架構，可以同時將二個以上的主裝置 (Master) 及從裝置 (Slave) 掛在 Avalon Bus 上進行任意方向之資料交換或控制交握。只要事先定義好此裝置為主裝置或從裝置，以及相關特定傳輸所需的控制訊號線，利用 SOPC Builder GNU 軟體，就可以自動產生如下所需的匯流排架構，包括以下幾個重要項目：

- 汇流排仲裁器
- 周邊或記憶體裝置之位址解碼
- 資料路徑多工
- 匯流排寬度自動估算
- 等待狀態的產生
- 中斷控制訊號



而 Avalon Bus 的傳輸區分為

- 主 (Master) 裝置傳輸
  - ✓ 用來向匯流排啟動傳輸的裝置
  - ✓ 傳輸時必須遵循從架構的等待 (waitrequest) 訊號
- 從 (Slave) 裝置傳輸
  - ✓ 用來被動回應匯流排的傳輸要求

以傳輸特性區分則有：

- 等待 (wait state) 傳輸：適用於某些速度較慢之周邊，包括固定或變動性等待，如圖 3-10 所示為具可變動等待時序之從架構讀取傳輸，當 clk 上緣而 waitrequest 訊號為 1 時，才代表從裝置已送出有效資料，主裝置

可以開始讀取資料。

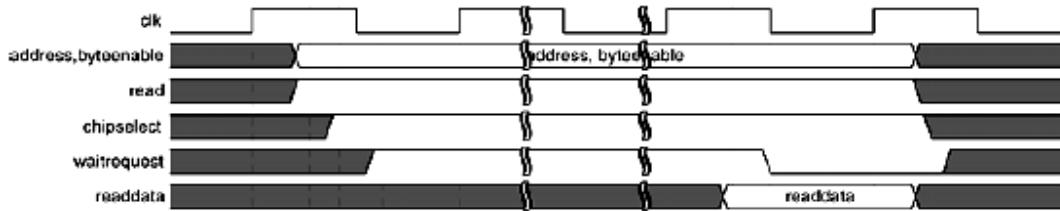


圖 3-10 具可變動等待時序之從架構讀取傳輸

- **間隔 (latency) 傳輸：**適用於某些需要固定或變動性等待且連續資料傳輸的周邊，而周邊也必須支援連續資料的傳輸。與等待傳輸不同的是，間隔傳輸在啓動一次連續資料的傳輸後只需做一次的等待，不會影響傳輸效能；而等待傳輸則是啓動傳輸後，每筆資料皆需要等待一次，會大大影響系統傳輸效能。如圖 3-11 所示為固定間隔 ( $\text{latency}=2$ ) 時序之從架構讀取傳輸，當  $\text{clk}$  上緣且  $\text{waitrequest} = 0$  時，則代表有效資料會在兩個 cycle 後出現；與等待傳輸不同的是，間隔傳輸在往後的每個 cycle 資料皆會持續出現，如圖 3-11 的  $\text{data2}$  至  $\text{data3}$ 。

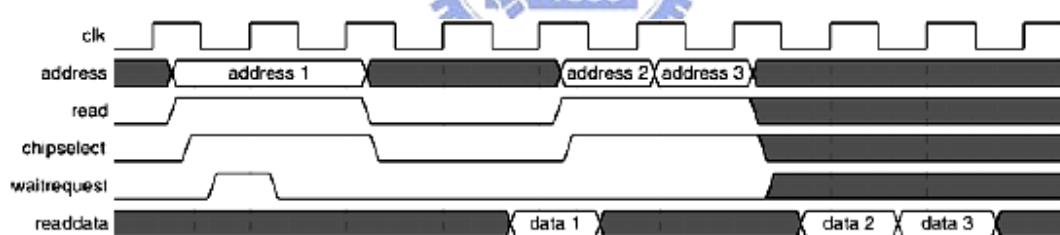


圖 3-11 固定間隔 ( $\text{latency}=2$ ) 時序之從架構讀取傳輸

- **串流 (streaming) 傳輸：**適用於需大量而連續的資料傳輸，但周邊必須支援連續資料串流的傳輸。如圖 3-12 為串流時序之從架構讀取傳輸，當  $\text{clk}$  上緣且  $\text{dataavailable} = 1$  時，則代表有效資料已經由周邊送出；當周邊已無資料時，可送  $\text{dataavailable}=0$  告訴主裝置停止擷取資料。另外  $\text{endpacket}$  則為一使用者自訂的控制訊號，方便其他特殊之用途。



圖 3-12 串流時序之從架構讀取傳輸

### 使用者自訂之客製化周邊模組

當使用者需要用到某些特殊或相當費時的運算，而 Nios 的運算效能又無法符合系統需求。此時我們可以自行設計一個客製化周邊，專門處理這些複雜運算。進行設計時，周邊不論是從裝置或主裝置，或是任何一種傳輸特性，皆需符合 Avalon Bus 之協定。在完成客製化週邊設計並定義好所需和 Avalon Bus 溝通的所有訊號線後，利用 SOPC Builder GNU 裡的 Component Generation 圖形化介面程式，自動幫你產生所需的元件，最後經過編譯，會自動產生符合系統需求的 Avalon Bus 架構。

另外，客製化周邊和 Nios CPU 若資料寬度不一致，會造成位址對應的錯誤，位址對應方式有以下兩種：

- 動態位址對應 (Dynamic-Address-Alignment)
- 自然位址對應 (Native-Address-Alignment)

如圖 3-13 所示為從裝置的客置化周邊(8 bit)的資料匯流排較 Nios(32 bit)為小之示意圖，Nios 在動態位址對應與自然位址對應的結果分別如下

- Dynamic Address Alignment
  - ✓ Load from Base + 0x0: dd cc bb aa
  - ✓ Load from Base + 0x4: uu uu uu ee
- Native Address Alignment
  - ✓ Load from Base + 0x0: uu uu uu aa

- ✓ Load from Base + 0x4: uu uu uu bb
- ✓ Load from Base + 0x8: uu uu uu cc

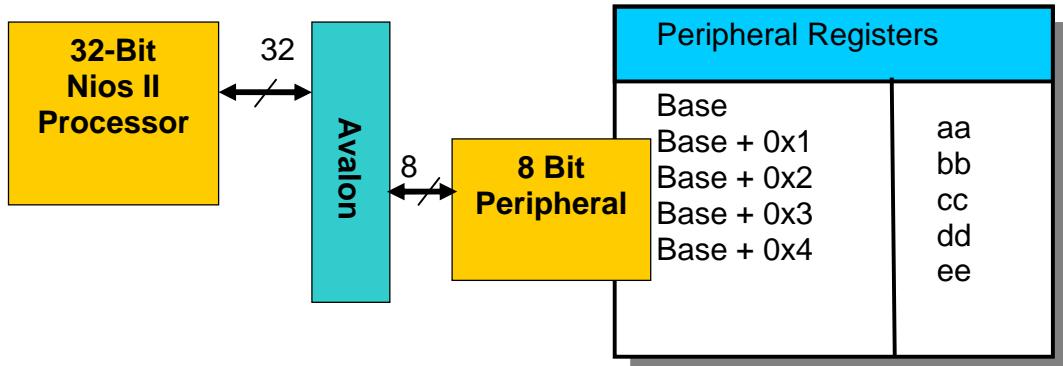


圖 3-13 資料匯流排較小之從裝置客製化周邊

如圖 3-14 所示為從裝置客製化周邊的資料匯流排較 Nios 為大之示意圖，Nios 在動態位址對應與自然位址對應的結果如下，其中自然位址對應 (Native Address Alignment) 在裝置客製化周邊的資料匯流排較小的情形下，客製化周邊資料最高的 4 個位元組會有資料遺失的情況。

#### ➤ Dynamic Address Alignment

- ✓ Load from Base + 0x0: 33 22 11 00
- ✓ Load from Base + 0x4: 77 66 55 44
- ✓ Load from Base + 0x8: bb aa 99 88

#### ➤ Native Address Alignment

- ✓ Load from Base + 0x0: 33 22 11 00
- ✓ Load from Base + 0x4: bb aa 99 88
- ✓ Load from Base + 0x8: ?? ?? ?? ??

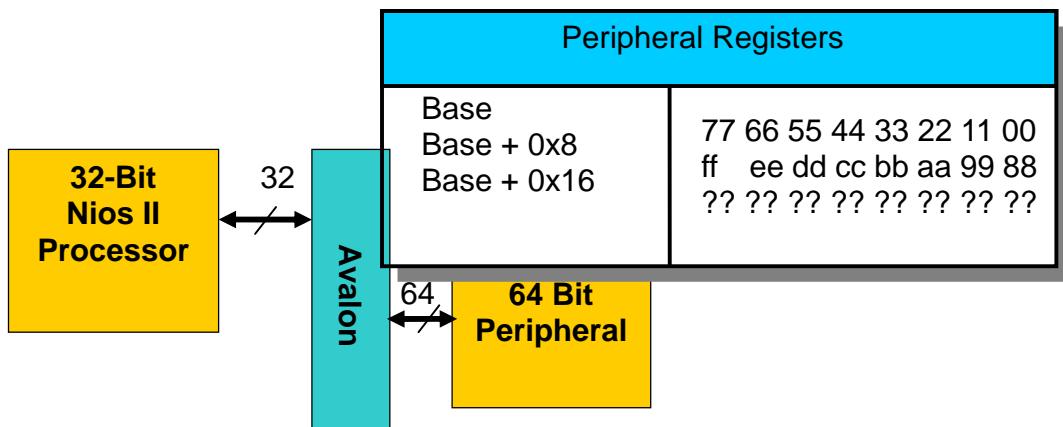


圖 3-14 資料匯流排較 Nios 大之從裝置客製化周邊

### 3.3.4 DMA 控制器[22,23]

當 Nios 微處理器對外面的周邊裝置或記憶體需要時常大量資料的搬移時，可以藉由 Altera 所提供的 DMA 控制器來進行大量連續資料的搬移，而此 DMA 控制器根據目的地及來源地的不同，區分為三種不同傳輸

- 由記憶體至周邊裝置的傳輸
- 由周邊裝置至記憶體的傳輸
- 兩塊記憶體之間的傳輸

此外，若 SOPC 系統單晶片中使用 DMA 控制器並搭配客製化周邊進行加速，可以使系統的效能提升不少，本論所實現之語音純化系統就是以如此的架構為基礎來設計。如圖 3-15 所示，Nios 微處理器和客製化周邊加速器同時都在工作。當中 Nios 可以呼叫 DMA 控制器並使用其串流的傳輸方式，將客製化周邊（Accelerator）處理完的資料送到 On-Chip 或 Off-Chip 的資料記憶體以進行下一步資料的處理。

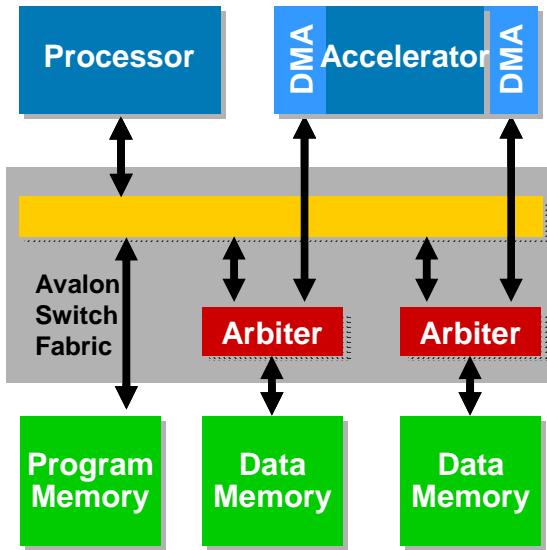


圖 3-15 搭配 DMA 控制器與客製化周邊之 SOPC 系統單晶片範示意圖

## 3.4 SOPC 單晶片系統軟硬體設計

### 3.4.1 系統功能描述及系統方塊圖

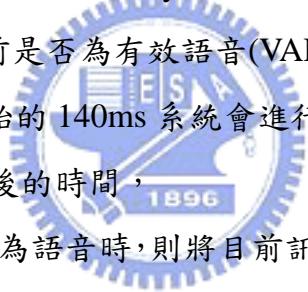
#### ➤ 系統功能

- ✓ 純化特定方向語者之語音，並消除其他方向之噪音源（包括已知及未知噪音源）
- ✓ 利用發展平台按鍵可以更改欲純化語者之語音，以及欲消除已知之噪音源
- ✓ 利用發展平台按鍵可以更改空間濾波器之階數，以適應不同噪音環境特性
- ✓ 透過 debug mode 將純化之語音傳送至 PC

此系統使用適應性空間濾波器，搭配真人語音活動偵測（Voice Activity Detection，VAD）以決定是否更新環境雜訊，或者直接將語音通過空間濾波器以去除雜訊效應，達到特定方向之語音純化的效果。最後純化之訊號透過

debug mode 傳送至 PC，如圖 3-16 所示。

➤ 系統執行步驟如下

1. 此系統工作前，先將欲純化之語音、欲消除噪音兩種聲源儲存起來（利用 B1 按鍵），用以做為適應性陣列訊號處理所需之參考訊號。
2. 選擇此系統中，適應性空間濾波之階數。利用 B2 按鍵增加階數，而 B3 按鍵則減少階數，可選擇階數為 32、64、96、128、160、192、224、256 等八種選擇。當環境變動快時，適合較少階數以利於環境更新速度；當環境變動慢時，則適合較高階數以達到較好的濾波效果。而不同的空間頻道效應，可嘗試選擇不同階數，以達到較好的語音純化效果。
3. 按下 B4 按鍵系統開始運做。
4. 初使化語音活動偵測（Voice Activity Detection，VAD）內所需之雜訊環境參數，並用它來判斷目前是否為有效語音(VAD Block)，以決定是否更新空間濾波器之係數。故開始的 140ms 系統會進行雜訊環始估測初始化，此時真人語音不能出現。往後的時間，  
i. 若進入之訊號判斷為語音時，則將目前訊號通過最後更新的一組空間濾波器係數，此係數可以將預錄雜訊及新進雜訊增益降至最低。  
ii. 若進入之訊號判斷為非語音時，則進行空間濾波器係數更新的動作，使得新係數可以適應新的環境雜訊。
5. 純化後之語音，透過 debug mode 傳回 PC。
6. 若在系統開始運作後又欲更改空間濾波器階數，則按 B2 回至 2.。
7. 若在系統開始運作後欲更改預錄欲純化之語音、及欲消除噪音兩種聲源，則按 RESET 鍵回至 1.。

如圖 3-16 所示為語音純化 SOPC 單晶片系統方塊圖，以下簡略說明各區塊功能：

➤ 資料擷取控制器：負責與 8CH A/D 擷取電路板做交握，以求能抓取到正

的資料並送至記憶體及暫存器模組。

- 系統流程控制器：在正確的時間將資料從記憶體模組送至執行單元（包括適應性空間濾波器及真人語音活動偵測）以及在正確的時間將運算結果搬運至所屬暫存器模組，其中也包括和 Nios 的交握控制訊號，為整個系統之控制核心。
- 匯流排介面及資料、控制相關暫存器檔案模組：負責產生客製化周邊連接 Avalon Bus 所需的資料、位址及相關控制訊號線。而資料控制相關暫存器檔案模組則用來存放 Avalon Bus 所需存取的資料。
- 按鈕偵測及濾波階數設定控制器：用來偵測平台按鈕是否被按下，以及濾波器階數設定的控制器。
- 適應性空間濾波器：包括 Dahl's Algorithm 適應性訊號處理所有的運算單元，分為濾波器輸出與適應性濾波器係數更新等區塊。
- 記憶體模組：用來儲存麥克風陣列 8 個頻道的資料、空間濾波器係數、以及欲純化語音和欲消除環境噪音等資料。
- 真人語音活動偵測器 (VAD)：判斷目前訊號是否為真人語音訊號，以選擇要進行空間濾波器輸出或空間濾波器係數更新的動作。
- LED 顯示系統狀態：用來顯示系統操作時的系統狀態。

### 3.4.2 系統運算量估測及軟硬體配置

#### 運算量估測

本系統需要做適應性空間訊號處理以及語音活動偵測，其中語音為八個通道且皆為 16KHz 取樣頻率，以下為各個主要區塊運算量估計，共需約 170MIPS 運算量。

- 適應性空間訊號處理部分：每一個取樣資料進來皆需進行以下運算

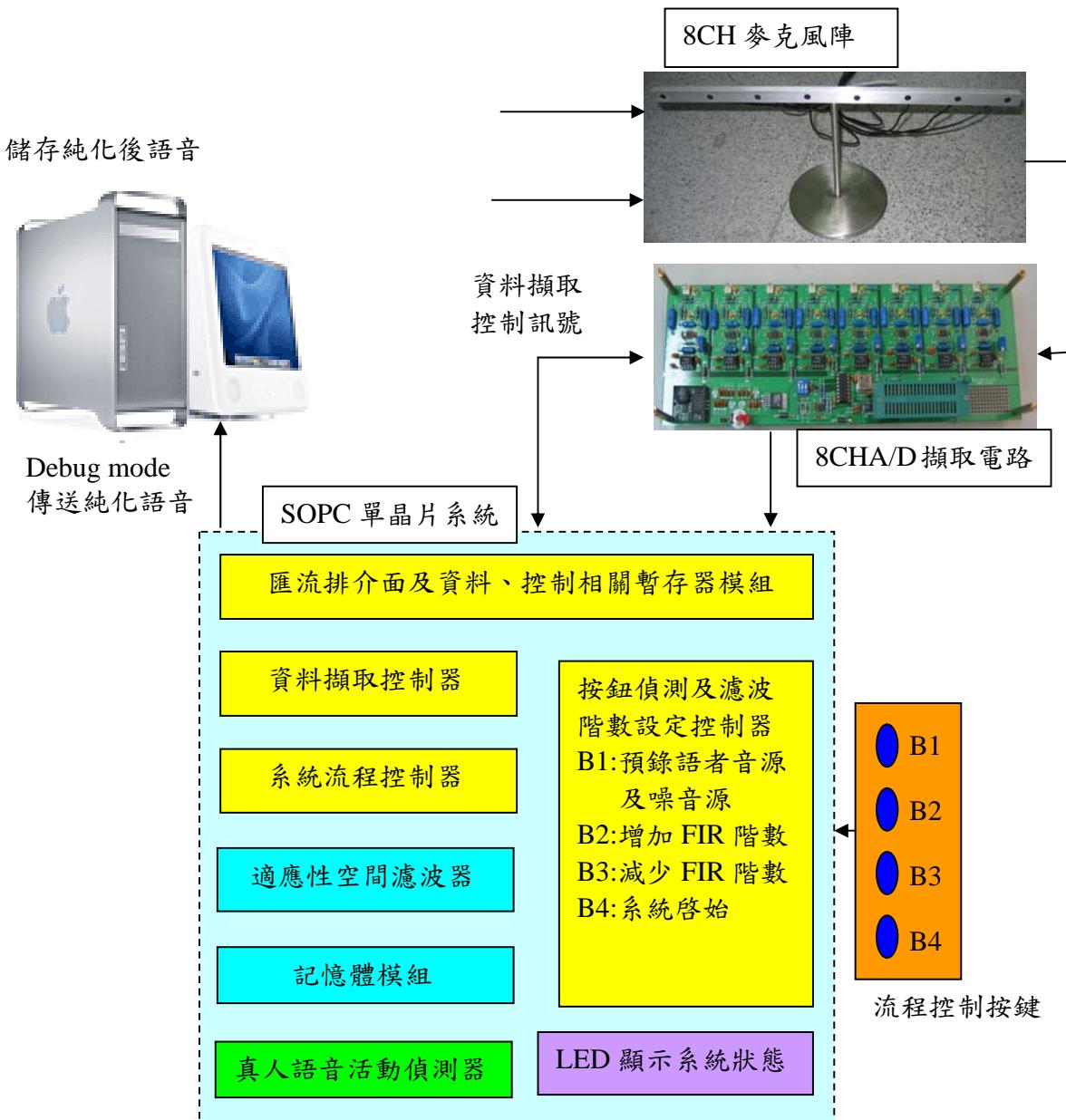


圖 3-16 SOPC 單晶片系統方塊圖

- ✓ 濾波器區塊：系統中，FIR 濾波器最多為 256 階，八個頻道。所以在 $(1/16K)\text{sec}$  內需執行 2048 個乘累加運算，也就是需要約 66MIPS 的運算量（規劃使用 NIOS 外部硬體的乘累加器進行加速，且用 pipeline 架構，故乘累加運算平均只需一個 cycle 即可完成）。
- ✓ 係數更新區塊：系統中，係數更新需要在 $(1/16K)\text{sec}$  內，執行 2049

個乘累加以及 2048 個加法與一個除法，這部份約需 100MIPS 的運算量（規劃使用 NIOS 外部加法器及乘累加器和除法器進行加速，且用 pipeline 架構，故乘累加及加法運算平均只需一個 cycle 即可完成，而除法用 pipeline 大約需 15 個 cycle）。

- **語音活動偵測部份：**系統中，一個語音訊框（frame）為 30msec，而每 10msec 就有一個新的語音訊框產生。故每 10msec 的間隔就需要判斷一次 VAD 的結果，時間上的緊迫性及運算量皆少於適應性空間訊號處理部分。這部份需作 512 點 Real FFT、13 個語音 frame 中每個頻率最大值並求其平方和、雜訊更新、及二個 log 等等... 運算，共約 5MIPS 的運算量。若以 NIOS 處理器(使用 FAST 版本的軟核心架構且頻率為 50Mhz) 實現，模擬上平均約使用百分之四十的效能。

#### 演算法的運算配置



在適應性空間訊號處理及語音活動偵測兩個部分在硬體資源的配置如下：

適應性空間訊號處理部分的性能需求，必需在每個( $1/16K$ )sec 需要作大量乘累加運算和加法運算，另外資料為八個通道同時進來（Avalon BUS 頻寬只有 32bit），若以單顆 NIOS 來實現比較沒有效率且來不及，故此部份以外部客製化週邊進行硬體平行運算來加速。

語音活動偵測部份的性能需求，只需在每隔 10msec 做出一次結果，時間上不如適應性空間濾波器急迫，且運算中包含很多複雜的資料交換及初使化條件判斷和流程控制，用 NIOS 來做處理顯得較符合硬體經濟效益且整個系統比較有彈性(如演算法需做小部份調整)。

### 3.4.3 系統硬體架構

本論文在 Nios 嵌入式 SOPC 發展平台實現即時語音純化系統，其系統硬體架構圖如圖 3-17 所示，整個系統硬體皆在 50MHz 的速度下動作，以下針對各個區塊做介紹

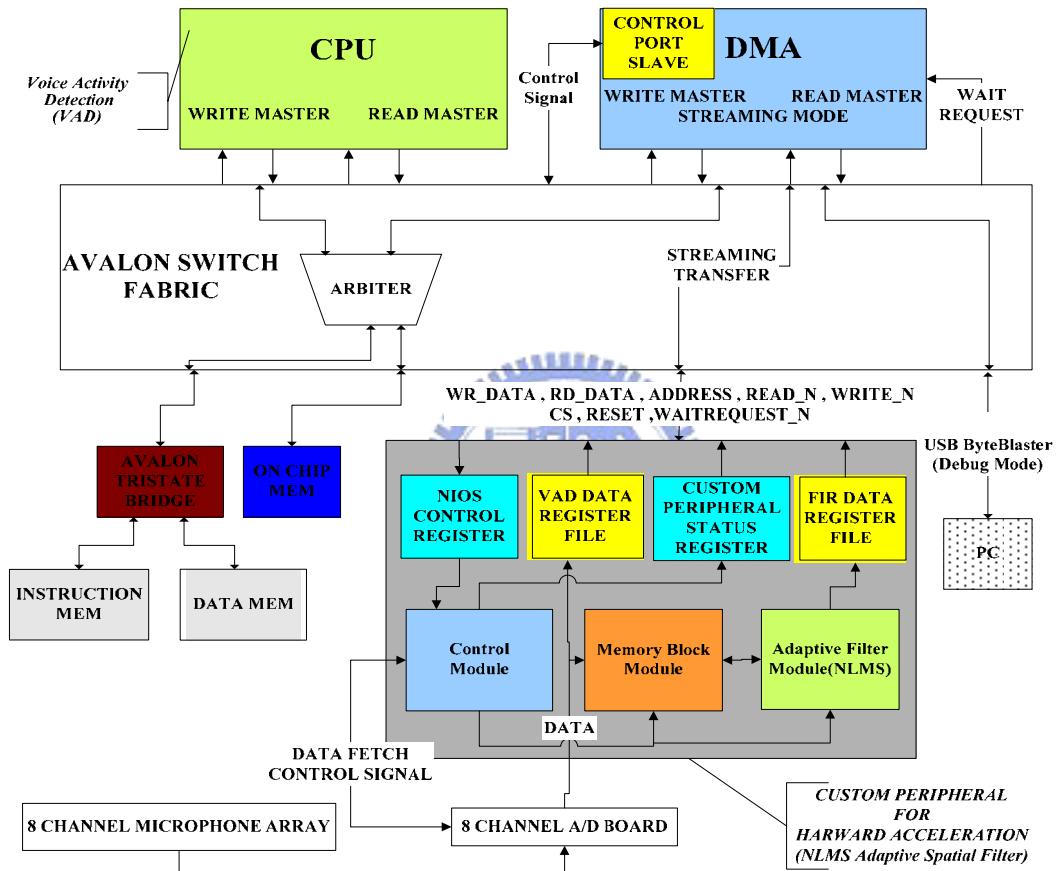


圖 3-17 SOPC 單晶片系統硬體架構圖

- NIOS 微處理器：規劃為一個 32bit 並為 Fast 版本的微處理器以達到較好的運算效能，負責真人語音活動偵測演算法(VAD)的運算，以及取得由客製化周邊運算之純化語音結果。

- 客製化周邊電路 (Custom Peripheral for Hardware Acceleration)：此周客製化周邊為一從 (Slave) 客製化周邊，利用客製化周邊電路進行適應性空間濾波器的硬體加速，詳細的客製化周邊硬體加速架構圖如圖 3-19 所示，包括下列模組：
- ✓ 適應性濾波器模組 (Adaptive Filter Module)：用來實現適應性空間濾波器，其使用的演算法為 NLMS 架構，區分為下列三個模組
    - ✧ 空間濾波器模組 (Spatial Filter Module)：當 VAD 判斷為真人語音時做濾波器輸出用途；相反地，當 VAD 判斷為非真人語音時，則用於計算係數更新所需之誤差訊號。
    - ✧ 資料能量計算模組 (Data Power Computation Module)：當 VAD 判斷為非真人語音時，用於計算係數更新所需之資料（為欲純化語音、欲消除雜訊、及目前進來訊號的總和）能量。
    - ✧ 濾波器係數更新模組 (Filter Weighting Update Module)：當 VAD 判斷為非真人語音時，利用誤差訊號及資料能量的結果，計算出新的八組 FIR 空間濾波器係數（每個頻道皆有一組相對應的空間濾波器）。
  - ✓ 記憶體模組 (Memory Block Module)：區分為下列三個模組
    - ✧ 資料記憶體 (Data Dual-Port 4K-RAM)：用來暫時儲存麥克風陣列輸入之資料。為一個雙埠記憶體，共有八塊資料記憶體分別儲存一個頻道之資料。
    - ✧ FIR 係數記憶體 (FIR Weight Triple 4K-RAM)：為一個三埠的記憶體，一個為輸入埠，另外兩個為輸出埠。共有八塊係數記憶體，每塊記憶體分別用來暫存一組 FIR 濾波器係數。
    - ✧ 欲純化語音記憶體 (Pure Signal M-RAM)：用來儲存預錄的欲純化真人語音，為一雙埠的記憶體，可儲存 250ms 的語音資料。

- ◆ 欲消除雜訊記憶體 (Noise M-RAM)：用來儲存預錄的欲消除環境雜訊，為一雙埠的記憶體，可儲存 250ms 的雜訊。
- ✓ 控制模組 (Control Module) 包括：
  - ◆ 流程控制模組 (Flow Control Module)：用來產生外部麥克風系統電路板的控制存取訊號、Nios 嵌入式發展平台的按鍵及 LED 顯示等 IO 介面，以及客製化周邊各個模組間的控制流程。
  - ◆ 資料擷取模組 (Data Access Module)：負責與 8CH A/D 擷取電路板做交握，以求能抓取到正的資料並送至記憶體及暫存器檔案模組。
  - ◆ 按鈕開關偵測模組 (Push-Bottom Detection Module)：用來偵測按鈕行為，包括避免彈跳的機制 (de-bounce)。
  - ◆ 濾波器階數設定模組 (Filter Tap Number Setting Module)：用來設定適應性空間濾波器的階數。
- ✓ 匯流排 (Avalon Bus) 介面及暫存器檔案 (Register File) 模組：
  - ◆ 匯流排 (Avalon Bus) 介面模組：如圖 3-18 所示為客製化周邊連結 Avalon Bus 所需的訊號示意圖。各訊號含義將於後面做詳細說明。
  - ◆ 濾波器資料暫存器檔案 (Filter Data Register File)：儲存由客製化周邊做空間濾波器輸出（純化後）之語音。
  - ◆ 真人語音活動偵測暫存器檔案 (VAD Data Register File)：儲存欲由 NIOS 進行真人語音活動偵測運算之資料，於八個聲音頻道中任選其中一個用於判斷之。
  - ◆ Nios 微處理器控制暫存器 (Nios Control Register)：NIOS 用來控制客製化周邊動作流程的暫存器。

◆ 客製化周邊狀態暫存器(Custom Peripheral Status Register): NIOS  
用來判斷客製化周邊目前所處狀態的暫存器。

- DMA 控制器模組：用來存取客製化週邊的暫存器檔案模組，並設計為具間隔且串流 (Latency & Streaming) 的傳輸方式。傳輸方式已於 3.3.3 Avalon Bus 一章節介紹，為一個可以進行連續資料存取的傳輸方式。其中 waitrequest\_n 則是進行間隔讀取時所需的交握手號（判斷 Register File 是否有無資料可讀取）。
- 系統中的 On-Chip-Memory：用來儲存程式中執行頻率較高的程式碼及資料變數，以增加資料存取速度。

以下章節將詳盡介紹各個模組之設計說明

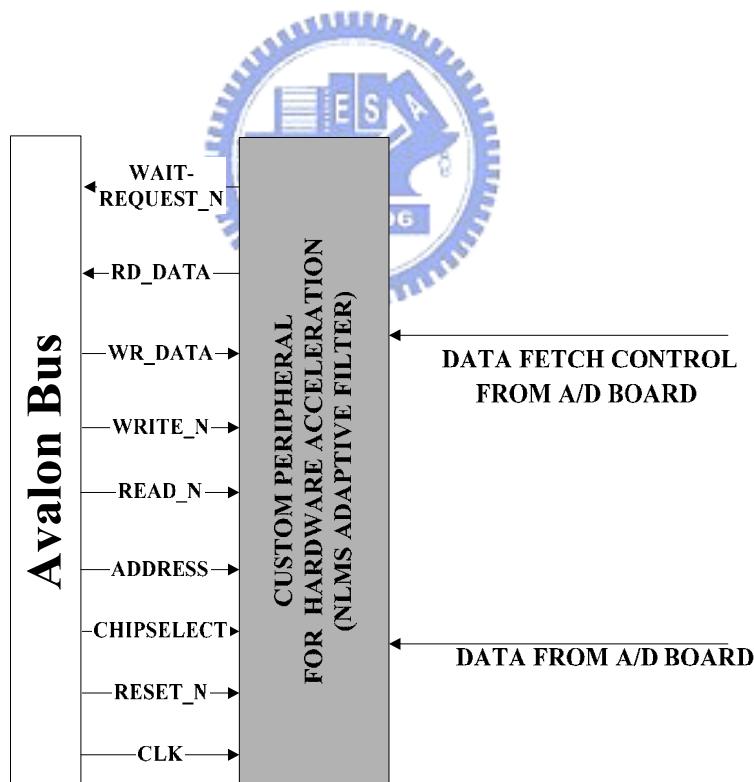


圖 3-18：客製化周邊連結 Avalon Bus 之訊號示意圖

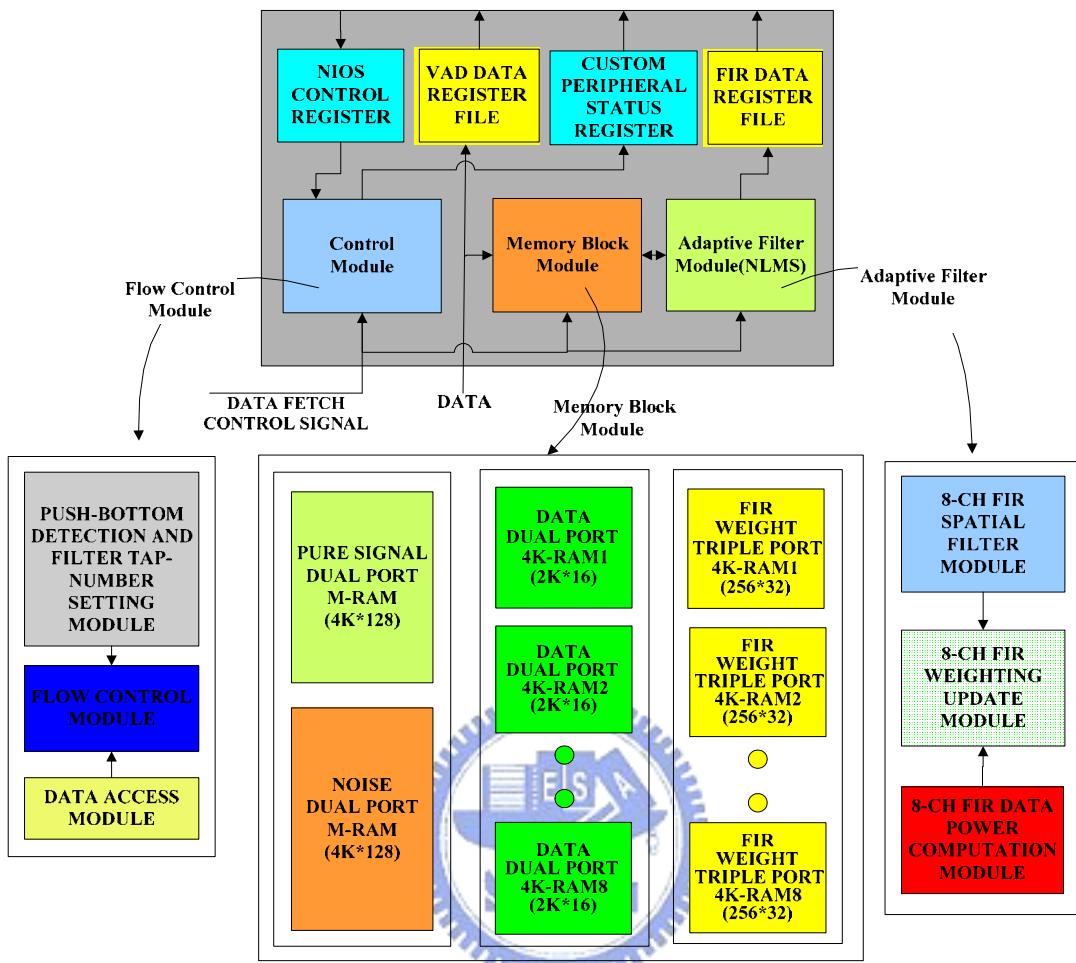


圖 3-19：客製化周邊硬體加速架構圖

### 3.4.4 匯流排介面及暫存器檔案模組

#### 3.4.4.1 匯流排介面模組

如圖 3-18 所示為此客製化周邊（適應性空間濾波器）連結 Avalon Bus 之訊號示意圖，以下介紹各個訊號之用途。

- CLK：為提供客製化周邊的頻率，和系統頻率相同為 50MHz。
- RESET\_N：為提供客製化周邊重置的作用，當 Nios 微處理器被重置時，

也同時送一個重置訊號給客製化周邊。為低準位動作非同步訊號。

- WR\_DATA：為客製化周邊至 Avalon Bus 的資料方向，提供 Nios 讀出資料的資料匯流排，為 32bit 的資料寬度。
- RD\_DATA：為 Avalon Bus 至客製化周邊的資料方向，提供 Nios 寫入資料的資料匯流排，為 32bit 的資料寬度。
- CHIPSELECT：當 Avalon Bus 要對客製化周邊做存取動作時，設為高準位，表示致能客製化周邊。
- WRITE\_N：當 Avalon Bus 要對客製化周邊做寫入動作，並且 Avalon Bus 資料已於匯流排端準備好時，設為低準位。
- READ\_N：當 Avalon Bus 要對客製化周邊做讀取動作，並且客製化周邊的資料已於匯流排端準備好時，設為低準位。
- ADDRESS：用來選擇所要存取客製化周邊中，暫存器檔案的位址。此客製化周邊共有四個暫存器檔案，故使用一組兩個位元的多工器用以選擇任何一個暫存器檔案。
- WAITREQUEST\_N：為使用間隔傳輸所需要的控制交握訊號，用來顯示客製化周邊在做傳輸時資料是否已就緒完畢。當 waitrequest\_n 為 0 時，表示客製化周邊的資料尚未就緒，Avalon Bus 會自動產生等待時序，直到 waitrequest\_n 為 1 時，才進行存取的動作。

#### 3.4.4.2 暫存器檔案模組

- 濾波器資料暫存器檔案 (Filter Data Register File)：為一個深度 512、寬度 32 的 FIFO (First In First Out) 所組成。設計上，FIFO 的 FULL (表示 FIFO 已滿) 訊號為 1 且被寫入時，或是 FIFO 的 EMPTY (表示 FIFO 已

空) 訊號為 1 且被讀出時，都會將 waitrequest\_n 設為 0，使得 Avalon Bus 產生等待時序，直到上述情形解除。

- 真人語音活動偵測暫存器檔案 (VAD Data Register File)：為一個深度 512、寬度 32 的 FIFO (First In First Out) 所組成。和 Avalon Bus 溝通的方式則與濾波器資料暫存器檔案相同。
- Nios 微處理器控制暫存器 (Nios Control Register)：為一個深度 4、寬度 32 的 FIFO (First In First Out) 所組成。和 Avalon Bus 溝通的方式則與濾波器資料暫存器檔案相同。
- 客製化周邊狀態暫存器 (Custom Peripheral Status Register)：為一個深度 4、寬度 32 的 FIFO (First In First Out) 所組成。和 Avalon Bus 溝通的方式則與濾波器資料暫存器檔案相同。

由於上述四個暫存器檔案資料寬度皆設計為 32bit，與 Nios 的資料匯流排有同樣的寬度，故在位址對應方面採用自然位址對應 (Negative -Address-Alignment) 方式即可。



### 3.4.5 資料擷取及流程控制模組

#### 3.4.5.1 資料擷取模組

資料擷取模組包含於客製化周邊內，功能之一為用來控制 8 Channel A/D Board 所需的資料擷取訊號。如圖 3-20 所示為資料擷取控制器方塊圖，相關控制訊號說明如下

- Hold：用來控制八顆取樣保持 (S/H, Samle & Hold) IC，在同一個時間點取樣然後保持八個麥克風的訊號。
- Switch：用來控制類比開關 (Analog Switch)，3 個 bit，分別控制擷取八顆取樣保持 IC 的訊號。

- Convert：用來控制數位類比轉換器（A/D Converter）開始轉換的訊號。
- CS：用來控制數位類比轉換器（A/D Converter）是否致能的訊號。
- Busy：用來偵器數位類比轉換器（A/D Converter）是否已經轉換結束的訊號。
- RD：用來控制數位類比轉換器（A/D Converter）開始讀取轉換結束之數位訊號。
- Data：為數位類比轉換器（A/D Converter）的資料匯流排，共 16bit。

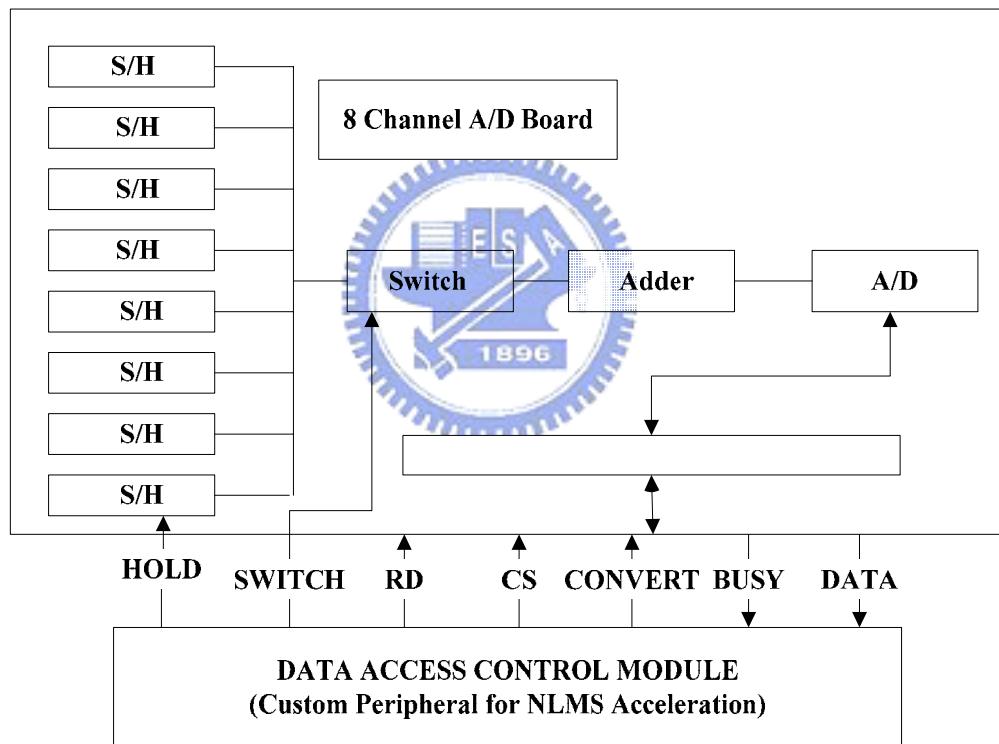


圖 3-20：資料擷取控制器方塊圖

如圖 3-21 所示為資料擷取控制器系統狀態流程圖，為一有限狀態機 (FSM)。除了負責控制 8 CH A/D Board 以擷取麥克風陣列訊號以外（數位類比轉換器、取樣／保持電路、類比開關等等控制），還包括

- 取樣頻率：16KHz 取樣頻率所需的計時器。
- 記憶體模組及暫存器檔案寫入：檢查流程控制器相關寫入控制旗標，以決定是否將擷取的麥克風陣列訊號寫入真人語音活動偵測暫存器檔案 (VAD Data Register File)、資料記憶體 (Data Dual-Port 4K-RAM)、欲純化語音記憶體(Pure Signal M-RAM)或欲消除雜訊記憶體 (Noise M-RAM) 等等。
- 階數重設程序：檢查流程控制模組的階數重設控制旗標。若被設定則
  1. 寫入 0x0F0F0F0F 至客製化周邊狀態暫存器 (Custom Peripheral Status Register)，當 Nios 微處理器從客製化周邊狀態暫存器讀到此值，則做重置動作。
  2. Nios 寫入 0x0F0F0F0F 至 Nios 微處理器控制暫存器 (Nios Control Register)，以通知資料擷取模組，Nios 已完成重設。
  3. 初使化所有暫存器檔案，並設定階數重設結束旗標，通知流程控制模組，已完成階數重設相關程序。
- 通知 Nios 讀取真人語音活動偵測暫存器檔案：當真人語音活動偵測暫存器檔案 (VAD Data Register File) 已儲存 160 筆資料，則寫入 0x0F0F0F0F 至客製化周邊狀態暫存器 (Custom Peripheral Status Register)。當 Nios 微處理器從客製化周邊狀態暫存器讀到此值，則以串流的方式從語音活動偵測暫存器檔案讀取 160 筆資料，並進行 VAD 運算。
- 通知 Nios 讀取濾波器資料暫存器檔案：檢查流程控制器的濾波器輸出旗標是否被設定。若是，則寫入 0x0F0F0F0F 至濾波器資料暫存器檔案 (Filter Data Register File)。當 Nios 微處理器從客製化周邊狀態暫存器讀到此值，則以串流的方式從濾波器資料暫存器檔案讀取 160 筆資料，其為濾波器輸出結果。

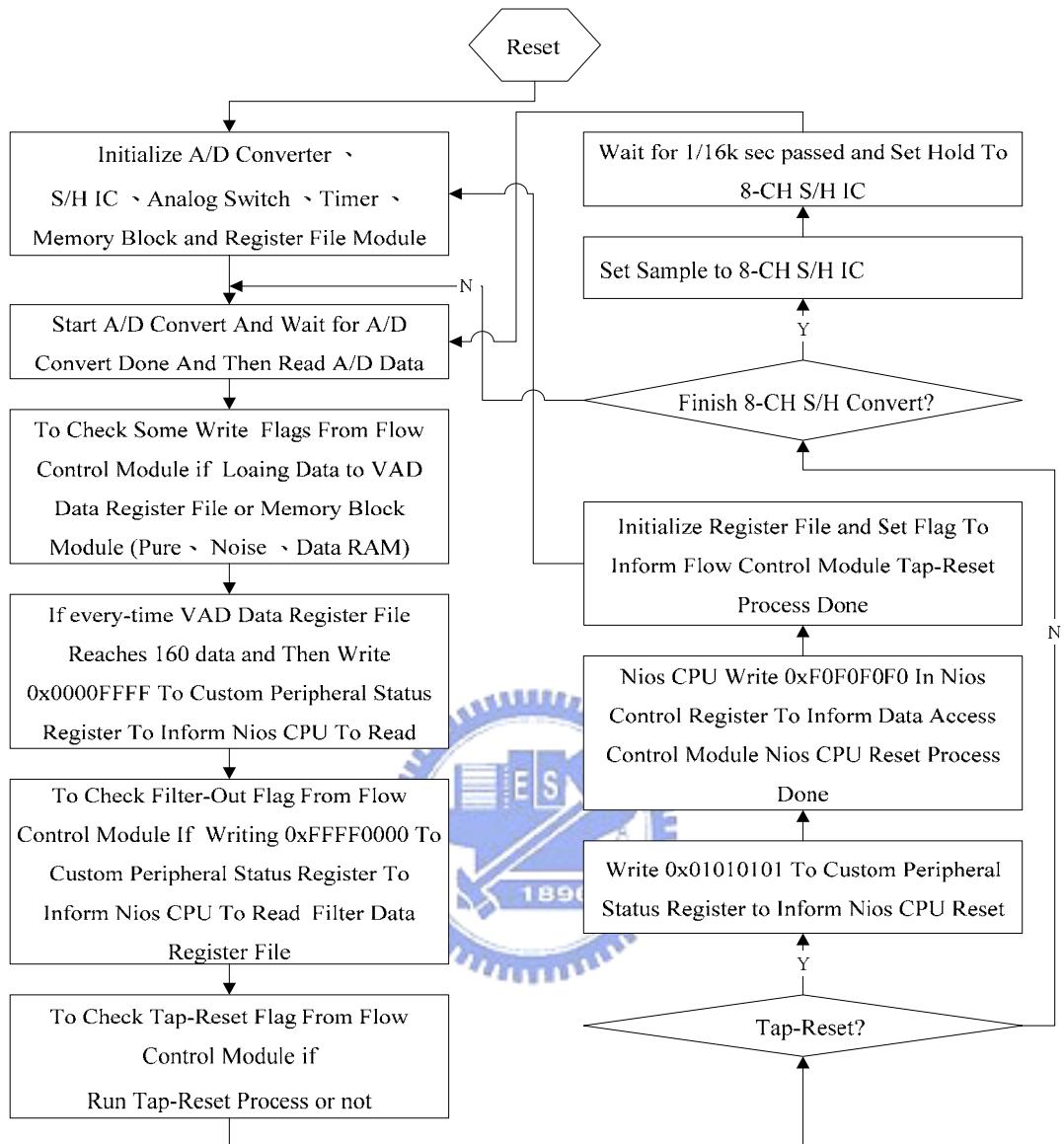


圖 3-21：資料擷取控制模組之狀態流程圖

### 3.4.5.2 流程控制模組

如圖 3-22 所示為流程控制器模組之狀態流程圖，為一有限狀態機。用以控制 I/O 介面操作、適應性空間濾波器運算時資料流程以及與資料擷取控制模組間的溝通。圖 3-22 中，B1-B4 表示為 SOPC 平台的四個使用者定義按鈕。而 LED1-LED8 為八個顯示燈，顯示操作時系統狀態。

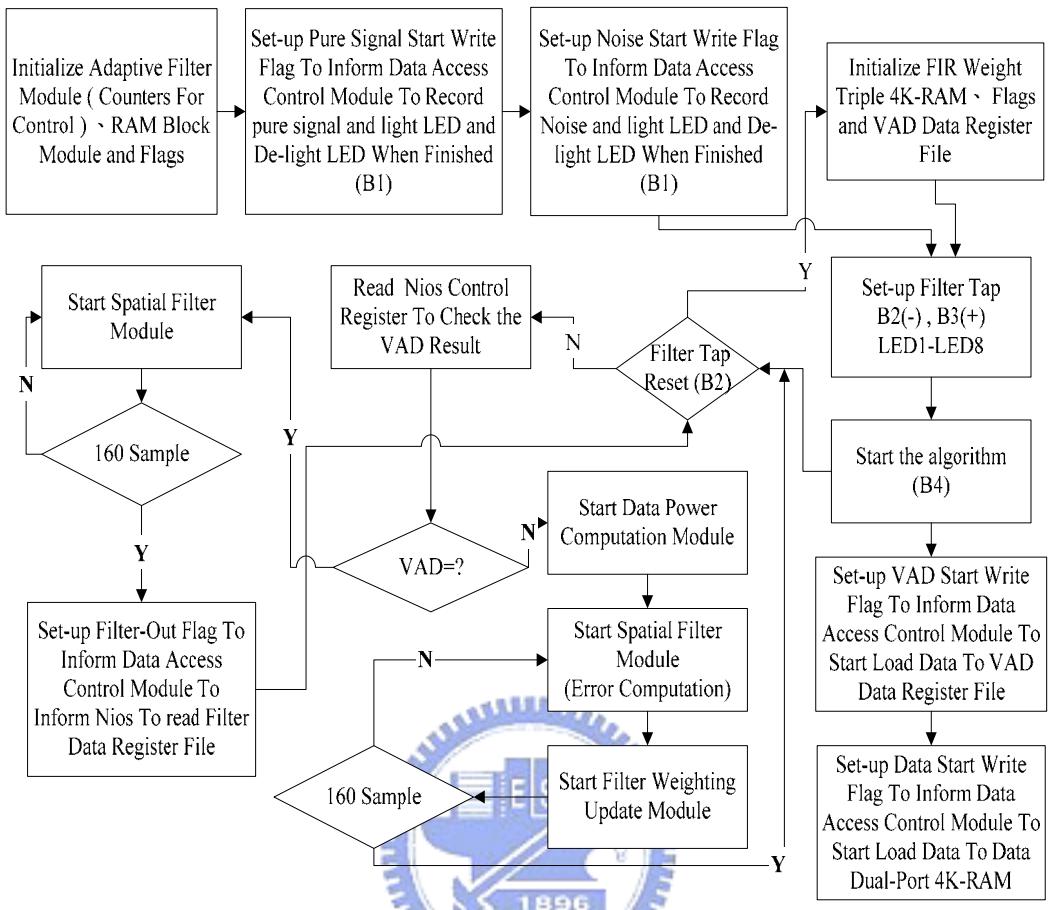


圖 3-22：流程控制器模組之狀態流程圖

### 3.4.6 按鈕開關偵測與濾波器階數設定模組

如圖 3-23 為其中一個按鈕開關偵測模組之狀態流程圖，四個按鈕 B1-B4 皆為一樣。有限狀態機的初使狀態為等待按鈕按下，當偵測到連續被按下 100usec 時才判斷為按下的狀態，以避免彈跳現象；接著等待按鈕放開並超過 100usec 時才判斷為放開的狀態，往後依此循環。

如圖 3-24 為濾波器階數設定模組之狀態流程圖。有限狀態機的初使狀態將濾波器階數設為 32 階。當偵測到由流程控制器模組 (Flow Control Module) 設定的開始旗標 (Start Set-up Flag) 則進入到按鈕偵測狀態。當 B2 被按下時，若此時階數大於 32，則將階數減 32，並等待 B2 放開，回到按鈕偵測狀態。當 B3 被按下時，若此時階數小於 256，則將階數加 32，並等待 B3 放開，回

到按鈕偵測狀態。當 B3 被按下時為結束設定，並送結束旗標給流程控制器模組，等待下一個開始旗標的來到。

此濾波器階數設定模組共有八種階數可選擇，範圍為 32~256 間，以 32 為一個間隔。在設定的同時，會將階數顯示於 LED 上，若階數為 32 則 LED1 會亮起，若為 64 則 LED2 亮起，依此類推，當階數為最大的 256 則 LED8 會亮起。

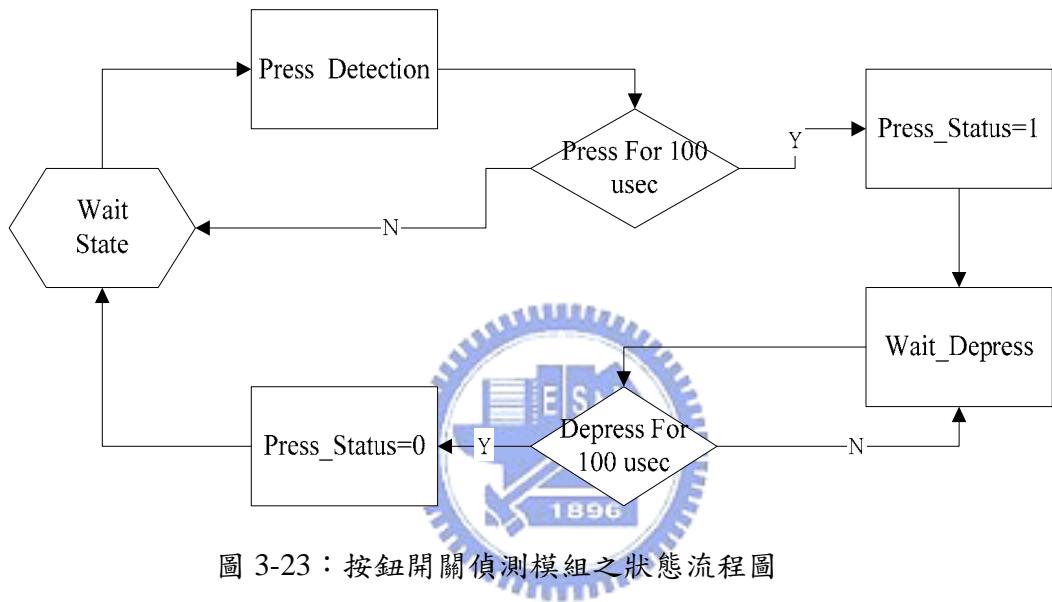


圖 3-23：按鈕開關偵測模組之狀態流程圖

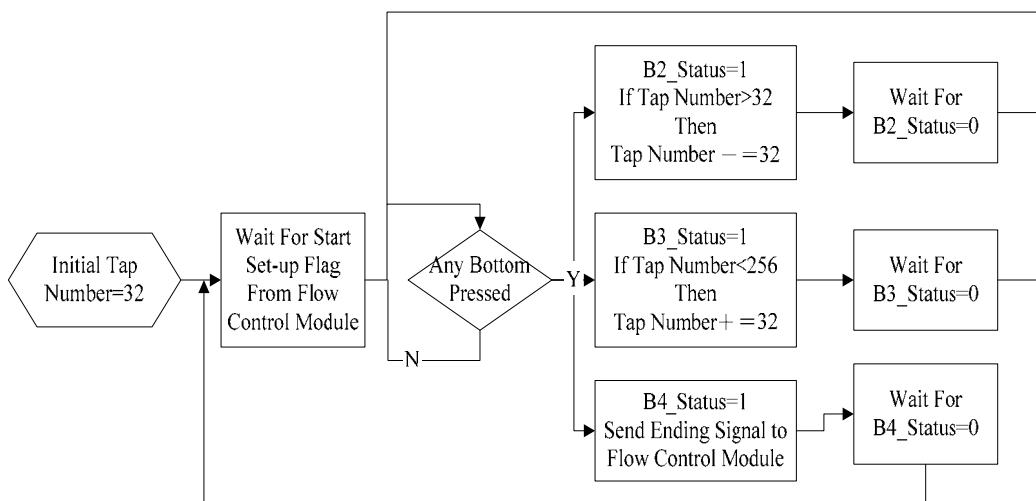


圖 3-24：濾波器階數設定模組之狀態流程圖

### 3.4.7 適應性空間濾波器模組

如圖 3-25 所示為適應性空間濾波器模組方塊圖。為整個客製化周邊之運算單元，所實現的適應性空間濾波器為 NLMS 演算法架構。此適應性空間濾波器以 NLMS 的方程式區分為三個部分實現，分別為

- 8-CH Spatial Filter Module：當真人語音時，做濾波器輸出  $y(i)$ ；若非真人語音時則運算出  $e(i)$  誤差訊號。
- 8-CH FIR DATA POWER COMPUTATION MODULE：當非真人語音時，需要計算訊號的能量  $x^*(i)x(i)$ ，用於濾波器係數更新之用。
- 8-CH FIR WEIGHTING UPDATE MODULE：當非真人語音時，將前兩個模組運算出的  $e(i)$  誤差訊號及訊號能量  $x^*(i)x(i)$ ，用來計算新的濾波器係數為  $w(i) = w(i-1) + \frac{\alpha x(i)e(i)}{\gamma + x^*(i)x(i)}$ 。由於麥克風實際上會有雜訊的存在顯少 0 有的情況，故在此將  $\gamma$  設為 0 並且將  $\alpha$  設為 1，最後方程式簡化為  $w(i) = w(i-1) + \frac{x(i)e(i)}{x^*(i)x(i)}$ 。

以下介紹各個模組之電路設計說明

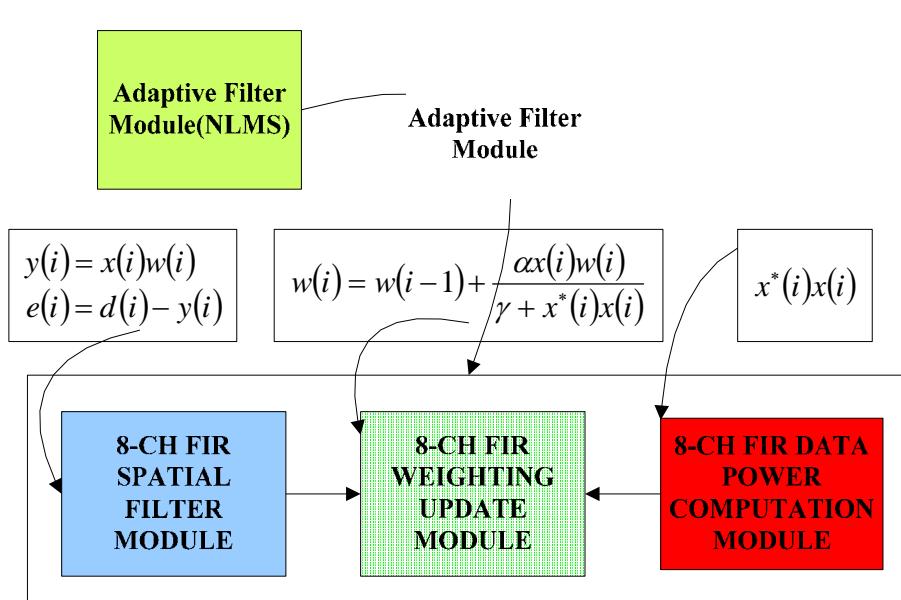


圖 3-25：適應性空間濾波器模組方塊圖

### 3.4.7.1 八個頻道之 FIR 空間濾波器模組

由於是八個頻道的麥克風陣列，所以總共需要八組 FIR 濾波器，相加的結果即為最後的輸出。且為了使濾波器設計為具階數彈性的架構，故最簡易且省運算單元的做法為，每組乘累加運算器作為單一麥克風頻道的運算單元如圖 3-26 所示。總共需要八組乘累加運算器，可以平行運算八個頻道的資料，而最後只需一個八個輸入的加法器，即可構成一組八個頻道的空間濾波器。

如圖 3-26 所示為單一麥克風頻道濾波器運算單元，當 Select 為 1 時，則將輸入訊號乘上濾波器係數做為濾波器輸出之用；當 Select 為 0 時，則將輸入訊號加上欲純化語音和雜訊兩個訊號，再與濾波器係數相乘做為與欲純化相比較之訊號。而濾波器係數選擇 32bit 之原因是為了增加解析度，這是由於進行濾波器係數更新運算的過程中（3.2.7.3 章節），為了提高誤差的收斂效果，而將小數點做位移的動作（低位元部份截去或補零），此位移會造成濾波器係數較大的位元數，於後面的 3.2.7.4 章節裡將進行模擬並做說明。

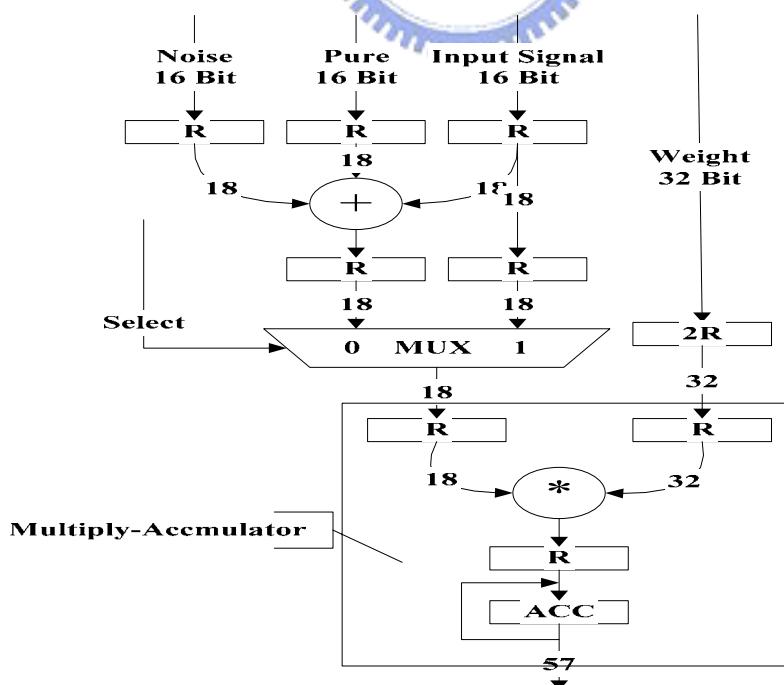


圖 3-26：單一麥克風頻道濾波器運算單元

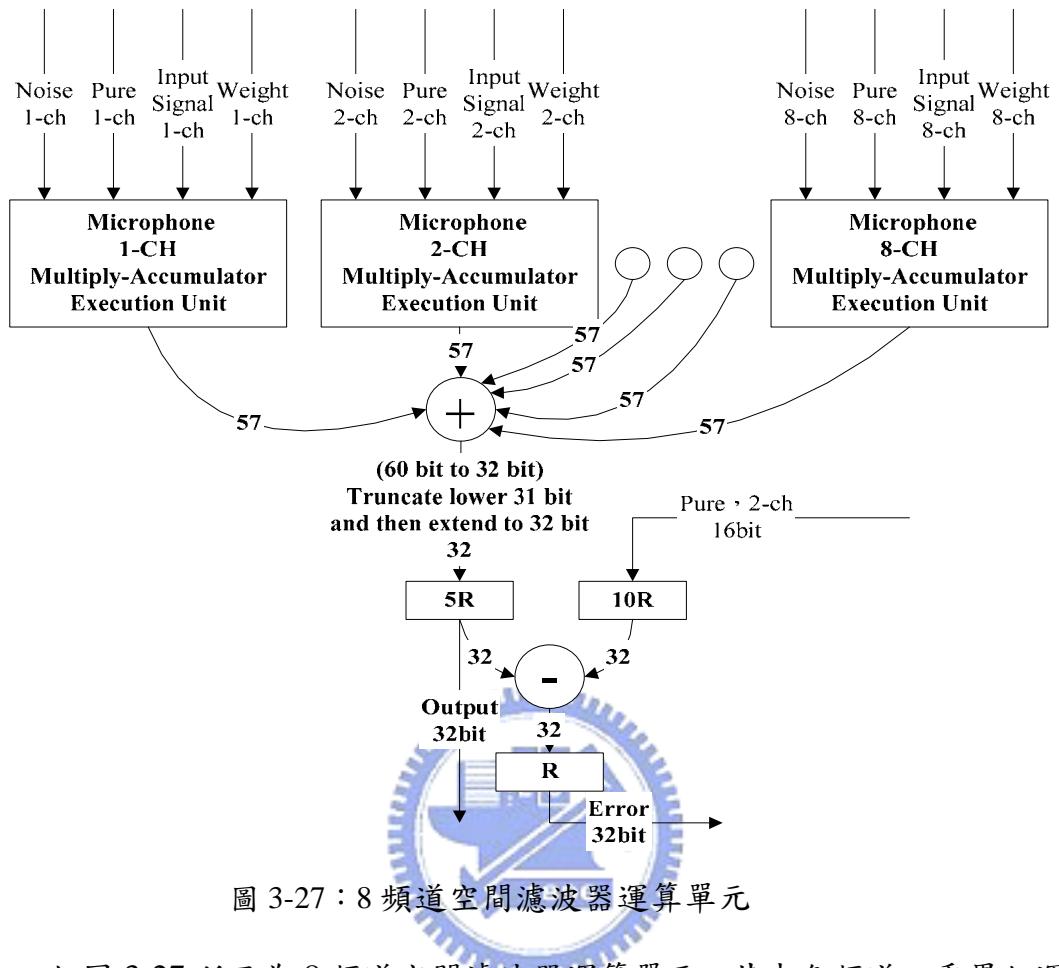


圖 3-27：8 頻道空間濾波器運算單元

如圖 3-27 所示為 8 頻道空間濾波器運算單元，其中各頻道之乘累加運算結果為八個輸入端加法器的輸入，加法器輸出原為 60 位元，先將較低的 31 個位元截去(truncate)，再將剩餘的 29 位元進行符號位元擴充(signed-extension)而成為 32 位元，此為濾波器係數更新模組運算過程有進行小數點移位的動作，故在此要小數點位移回來，原因將在 3.4.7.3 及 3.4.7.4 章節做說明。當為有效真人語音訊號時，此加法器輸出即為濾波器輸出結果。當不為真人語音訊號時，則將此訊號與欲純化真人語音訊號（選擇第二頻道）相減，所得結果即為濾波器係數更新模數所需要之誤差訊號值。另外，在濾波器模組的所有訊號皆為的 2 補數的格式，包括所有的運算過程如乘累加器及加（減）法器等等。模組內除了清除訊號為非同步外，其餘皆為同步設計。而模組內的 clk\_en 用於致能模組之用。

### 3.4.7.2 訊號能量計算模組

如圖 3-28 所示為訊號能量計算模組運算單元。當判斷為非真人語音而進行濾波器係數更新時，用於計算八個頻道訊號的能量。如圖 3-28 所示，8-1 的多工器前端八個頻道輸入訊號，為從 8 頻道空間濾波器模組拉過來的訊號。利用多工器 1(MUX1)可以選擇目前要計算哪一個頻道的能量。而經過多工器 2(MUX2)的路徑 B，為用來累加八個頻道的訊號能量。而經過多工器 2(MUX2)的路徑 A，為用來計算每個時間點八個頻道的訊號能量。

運作流程為如下（假設選擇為 32 階濾波器）：

1. 一開始前面的 32 個時間點，先由路徑 B 計算 32 個時間點八個頻道訊號累積的能量，同時路徑 A 將每個時間點的八個頻道訊號累積能量儲存至 FIFO。
2. 當第 33 個時間點的訊號進來時，只需先由路徑 B 加入第 33 個時間點的八個頻道訊號能量，然後再經由路徑 A 將第 1 個時間點的八個頻道訊號能量扣除掉，於是運算出新的訊號累積能量。

相關控制訊號為，CLK\_EN 用來控制輸入訊號是否能通過多工器 1；Channel\_Select 用來選擇要讓哪個頻道通過多工器 1；Acc\_Sel 用來選擇路徑 A 或 B 通過多工器 2；Clr\_Acc1 用來清除累加器 1；Add\_Sub\_Acc2 用來選擇累加器 2 要做加法或減法，而 Clk\_En\_Acc2 用來選擇是否致能累加器 2；Rd\_FIFO 及 Wr\_FIFO 分別用來控制 FIFO 的讀和寫；另外有一個模組致能及清除訊號，控制整個模組是否被致能或清除。

而在訊號能量計算模組的所有訊號皆為的 2 補數的格式，包括所有的運算過程如累加器及加（減）法器等等。模組內除了清除訊號為非同步外，其

餘皆為同步設計。

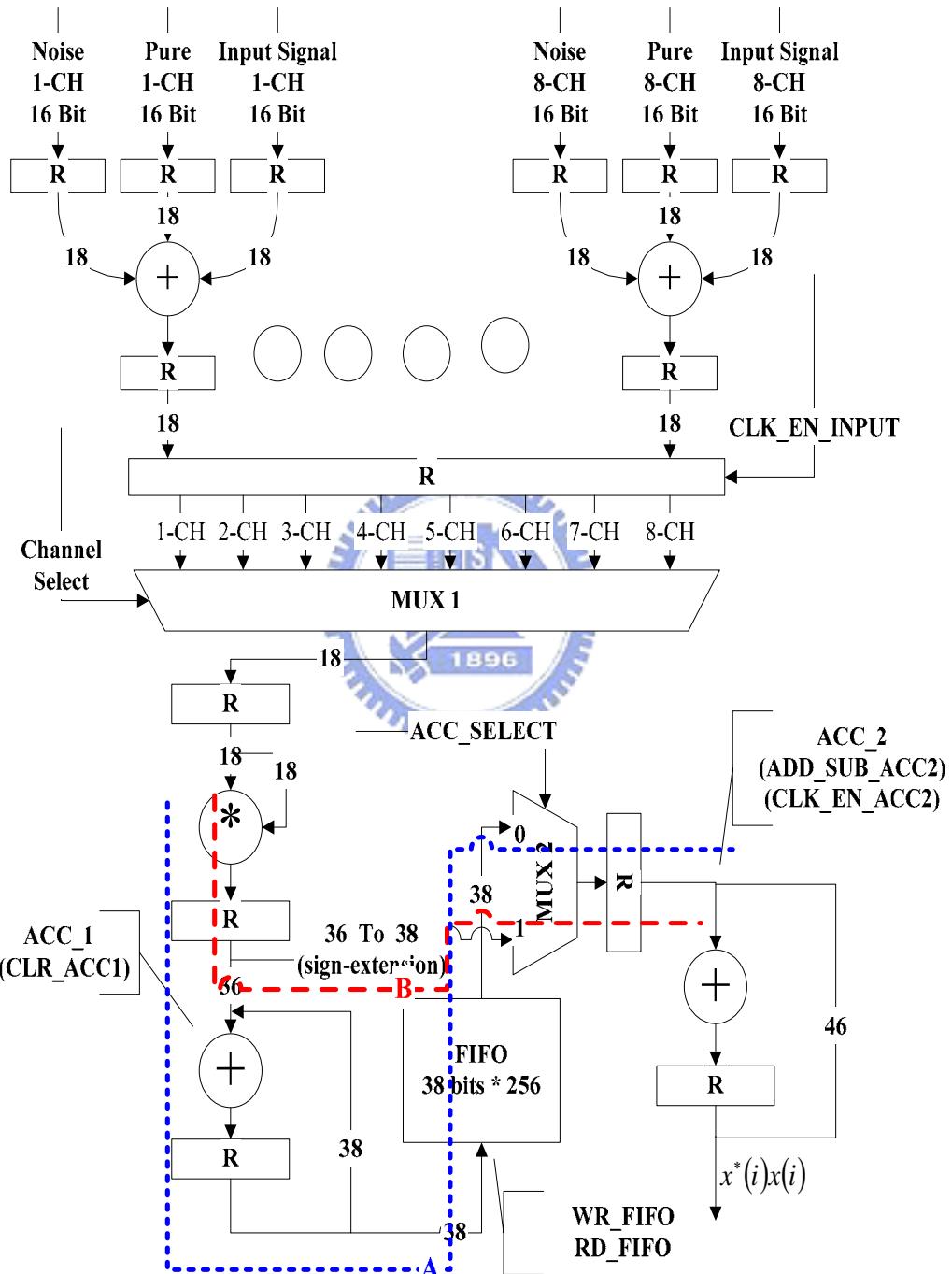


圖 3-28：訊號能量計算模組運算單元

### 3.4.7.3 濾波器係數更新模組

如圖 3-29 為濾波器係數更新模組運算單元。當判斷為非真人語音而進行適應性調整濾波器係數時，用於更新濾波器的係數。如圖 3-29 所示，運算流程如下

1. 兩個訊號輸入源誤差訊號 (Error) 及能量訊號(Power)分別為空間濾波器運算單元及訊號能量計算模組運算單元輸出的結果。經由 Matlab 模擬結果 (3.4.7.4 章節)，誤差訊號 (Error) 遠遠小於能量訊號(Power)，若直接將兩個訊號相除必為 0，於是將誤差訊號在低位元部分加上 36 個位元的零後再除以能量訊號，再將商數限制為 21 位元的 2 補數的範圍內。
2. 將誤差訊號 (Error) 及能量訊號(Power)相除的商數和輸入訊號相乘。而此輸入訊號為空間濾波器運算單元中，將純化語音、雜訊及輸入訊號等三個訊號相加的結果儲存於濾波器係數更新模組中的 FIFO。相乘的結果為 39 位元，再將較低的 5 個位元截掉(Truncation)，並限制為 32 位元 2 的補數範圍。在此若截掉太多位元將影響誤差收斂效果，將於後面說明。
3. 最後將上述結果與前一個時間點的濾波器係數相加，完成濾波器係數的更新，係數為一 32 位元 2 的補數範圍。而在相加的過程中，為了避免發生因溢位而造成適應性更新係數的方向發生錯誤而造成系統不穩，我們將結果限制在 32 位元 2 補數的正負最大值。

另外在 3.4.7.1 節中的圖 3-27 中，提到此 8 頻道各別的濾波器運算單元乘累加器的輸出，並經過加法器相加的結果原本為 60 位元。但最後先截去較低的 31 位元並經過有號位元擴充為 32 位元。這是因為我們將濾波器係數更新模組中，輸入的誤差訊號在低位元加上 36 個零，並且在過程中截去較低的 5 位元之後再限制大小在 32 位元的範圍（在此若截掉太多將失去訊號解析度，

太少又將造成運算單元過大)，於是還留下 31 位元的位移量，故最後在濾波器運算單元乘累加器輸出的結果中位移回來。以上截去（Truncation）或加上零（Zero-Padding）位元的行為均事先在 Matlab 模擬過，可以達到不錯的誤差收斂效果，將於 3.4.7.4 做說明。

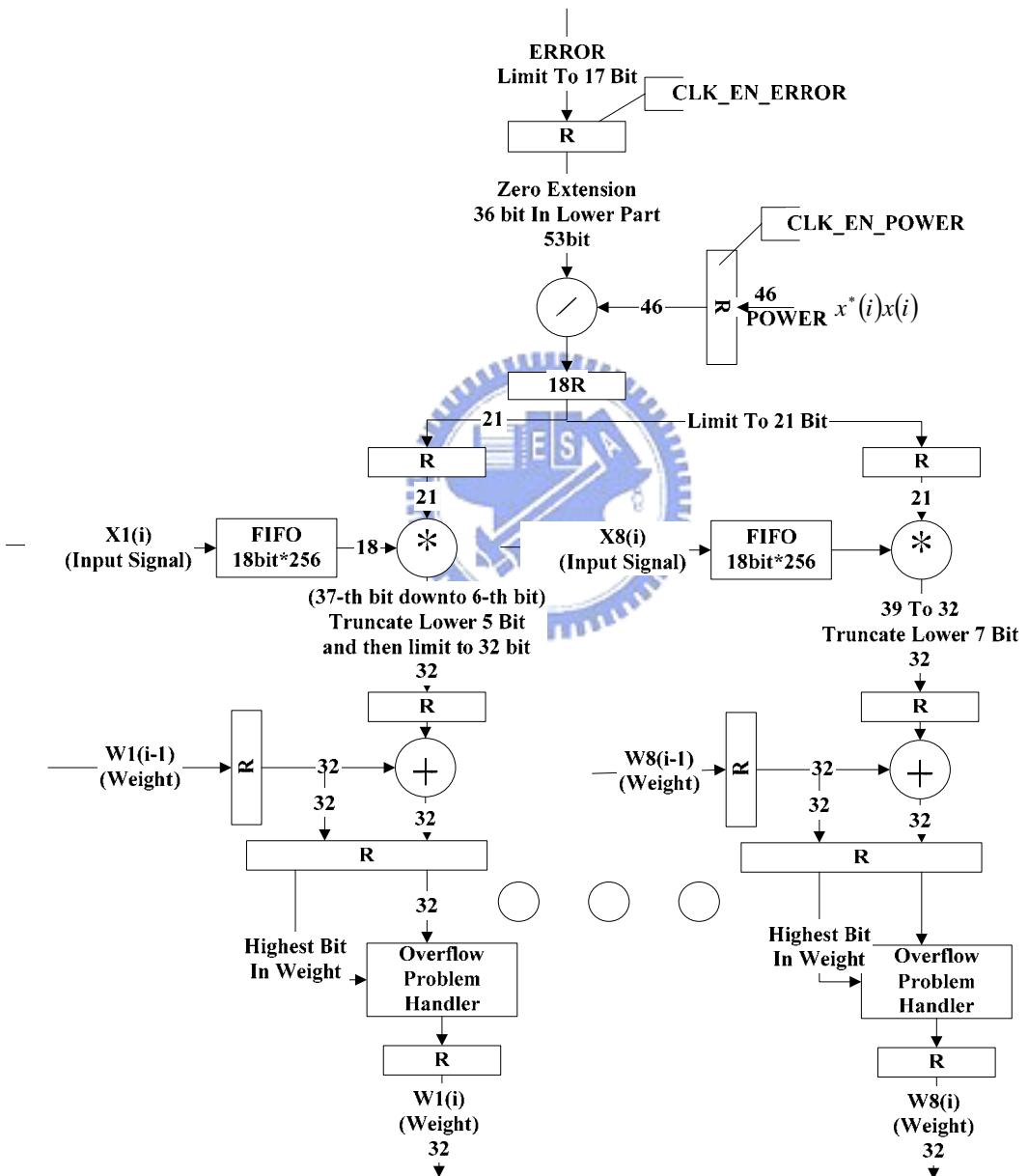


圖 3-29：濾波器係數更新模組運算單元

### 3.4.7.4 適應性空間濾波器模組在 MATLAB 模擬

本章節將 3.4.7.1 至 3.4.7.3 章節所設計的適應性空間濾波器（NLMS）進行模擬測試，以確保電路中所有的運算過程皆能保有 NLMS 應有的效果，也就是誤差的收斂效果。

測試環境為，先預錄一段真人語音做為欲純化語音，長度為 4096。接著錄製一段噪音源，空間中存在於任何形式的雜訊，但我們使用白色雜訊（white noise），這是由於 NLMS 的推導過程中皆假設雜訊為白色雜訊，如此才能印證所設計的電路是否在理論的基本假設下仍能作用，且我們保証訊號在運算過程中皆保持 16 位元以上的解析度。我們針對以下四種不同因素來評估其影響誤差值收斂的效果，並選擇其中一組有最好的收斂效果做為電路的設計依據。

- 
- 濾波器階數：選擇 256 及 32 兩種極端情況，在進行濾波器係數更新時，它會直接影響訊號總能量的動態範圍，其影響誤差值除以訊號能量的解析度而間接影響誤差值的收斂效果，而解析度由誤差值的放大倍率所決定。
  - 聲音訊號的大小：選擇最大的 16 位元及 10 位元兩種不同訊號動態範圍，影響的層面和濾波器階數相同，會直接造成訊號能量大小成平方比的影響而使誤差收斂效果的不同。
  - 誤差值放大倍率：分別以 2 的 24 次方、30 次方、36 次方等 3 種放大倍率進行模擬。由於誤差值大小級數和訊號能量差距太大，故必須先將它放大，再與訊號能量相除，以提高訊號的數值解析度，其間接影響誤差收斂效果。
  - 濾波器係數位元數：在此我們分別選擇 24 位元、30 位元、36 位元來進行模擬，越大的濾波器位元數，可以降低訊號在進行小數點移動時造成

訊號的數值解析度變差的影響。

### 情況一：256 階濾波器、16 位元動態訊號範圍、32 位元濾波器係數

情況一會造成最大的訊號能量。表 3-2 為在 256 階濾波器、16 位元動態訊號範圍、32 位元濾波器係數的情況下，在不同的誤差值放大倍率中，誤差值收斂情況。

如表 3-2 所示，越大的誤差值放大倍率有越好的穩態收斂效果，但所耗的除法器將越大，而在  $N=24$  的情況下，誤差值沒有收斂的效果，這是因為情況一所造成的訊號能量很大的關係，造成  $\left[ \left( \text{誤差值} * 2^N \right) / \text{訊號能量} \right]$  的解析度很差間接影響誤差收斂效果。

在收斂時間上， $N$  為 36 及 30 兩種情況的收斂速度皆一樣。而在  $\left[ \left( \text{誤差值} * 2^N \right) / \text{訊號能量} \right]$  的動態範圍方面，越大的誤差值放大倍率造成越大的動態範圍，所需表示數值的總位元數就越多，在此由於訊號能量很大，故動態範圍相對較小，沒有溢位的問題，但卻有因為太小的動態範圍造成訊號數值解析度下降，間接影響誤差值收斂效果。

誤差值放大倍率 $2^N$ 誤差值收斂情況	$N=36$	$N=30$	$N=24$
誤差值(穩態)	1462	1655	無穩態
$\left[ \left( \text{誤差值} * 2^N \right) / \text{訊號能量} \right]$ 之最大絕對值	11689	183	3
誤差收斂時間 (sample point) 〔Settling Time, 90%〕	40960	40960	無穩態

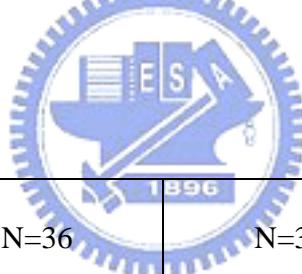
表 3-2：不同誤差值放大倍率的誤差值收斂情況對照表（256 階濾波器、16 位元動態訊號範圍、32 位元濾波器係數）

## 情況二：32 階濾波器、10 位元動態訊號範圍、32 位元濾波器係數

情況二會造成最小的訊號能量，及較小的誤差值（由於動態訊號範圍較小）。表 3-3 為在 32 階濾波器、10 位元動態訊號範圍、32 位元濾波器係數的情況下，在不同的誤差值放大倍率中，誤差值收斂情況。

由於情況二的  $[(\text{誤差值} * 2^N) / \text{訊號能量}]$  動態範圍較大，所以訊號沒有數值解析度的問題，故在三種誤差值放大倍率中，穩態誤差值及誤差收斂時間皆為相同。但要注意的是  $[(\text{誤差值} * 2^N) / \text{訊號能量}]$  動態範圍較大，所需表示數值的總位元數為最多，在  $N=36$  情況下，必需有 21 位元來表示  $[(\text{誤差值} * 2^N) / \text{訊號能量}]$  的動態範圍。

由情況一及情況二的分析，我們選擇  $N=36$ ，也就是較大的誤差值放大倍率，以達到較好的誤差收斂效果，而這些操作也同時保有訊號 16 位元的解析度。



誤差值放大倍率 $2^N$	$N=36$	$N=30$	$N=24$
誤差值(穩態)	73	73	73
$[(\text{誤差值} * 2^N) / \text{訊號能量}]$ 之最大絕對值	555023	8672	136
誤差收斂時間 (sample point) [ Settling Time, 90% ]	12288	12288	12288

表 3-3：不同誤差值放大倍率的誤差值收斂情況對照表（32 階濾波器、10 位元動態訊號範圍、32 位元濾波器係數）

### 情況三：256 階濾波器、16 位元動態訊號範圍、 $2^{36}$ 誤差值放大倍率

從情況一及情況二的分析，我們選擇  $N=36$  為誤差值放大倍率以確保較好的收斂效果。接著我們分析在誤差值放大倍率為  $2^{36}$ 、256 階濾波器、16 位元動態訊號範圍的情況中，在不同濾波器係數位元所造成的誤差收斂情況。

如表 3-4 所示，在濾波器係數位元數為較大的 32 位元時，在穩態誤差值收斂及誤差收斂時間兩方面皆有較好的效果，而  $\left[ \left( \text{誤差值} * 2^N \right) / \text{訊號能量} \right]$  動態範圍則與濾波器係數位元數沒有相關性。

在此我們選擇 32 位元的濾波器係數，以達到好的誤差值收斂效果。

濾波器係數位元數 M 誤差值收斂情況	M=32	M=28	M=24
誤差值(穩態)	1462	1477	1500
$\left[ \left( \text{誤差值} * 2^N \right) / \text{訊號能量} \right]$ 之最大絕對值	11689	11689	11689
誤差收斂時間 (sample point) $\left[ \text{Settling Time}, 90\% \right]$	40960	49152	57344

表 3-4：不同濾波器係數位元數的誤差值收斂情況對照表（256 階濾波器、16 位元動態訊號範圍、 $2^{36}$  誤差值放大倍率）

**結論：**綜合三種情況的分析，在此適應性濾波器的設計中，為了在不同的濾波器階數及聲音訊號大小皆能達到較好的誤差值收斂效果，我們選擇  $2^{36}$  的誤差值放大倍率及 32 位元的濾波器係數，並以 21 位元來表示  $\left[ \left( \text{誤差值} * 2^N \right) / \text{訊號能量} \right]$  的動態範圍。

### 3.4.8 記憶體模組

如圖 3-30 所示為記憶體模組單元方塊圖。共分為以下模組

- 欲純化語音記憶體模組：共使用一塊 M-RAM，以記錄八個頻道各 4K 的語音資訊 (0.25 秒)。
- 欲消除雜訊記憶體模組：與欲純化語音記憶體模組一樣，共使用一塊 M-RAM，以記錄八個頻道各 4K 的雜訊資訊 (0.25 秒)。
- 資料記憶體模組：共使用八塊資料記憶體，每塊資料記憶體使用 8 塊 4K-RAM，個別暫存八個麥克風頻道訊號。
- 濾波器係數記憶體模組：共使用八塊濾波器係數記憶體，每塊濾波器係數記憶體使用 2 塊 4K-RAM，個別暫存八組濾波器的係數。

以上四種記憶體將於以下章節做完整設計說明

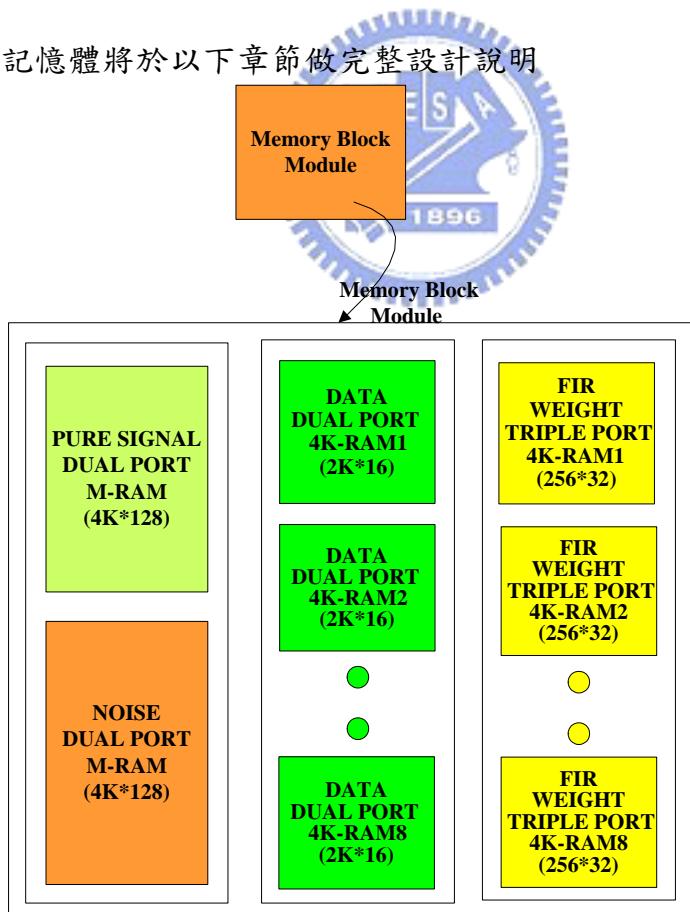


圖 3-30：記憶體模組單元方塊圖

### 3.4.8.1 欲純化語音記憶體模組(欲消除雜訊記憶體模組)

如圖 3-31 為欲純化語音或雜訊記憶體模組單元。有一個輸出埠（接至空間濾波器運算單元中的欲純化語音或雜訊輸入埠）和一個輸入埠（接至 8 CH A/D Board 的資料匯流排），皆為 128(16Bit\*8)位元寬度，分別儲存八個頻道各 16 位元的資料。其他訊號說明如下：

- CLK 及 CLK\_EN：分別為此同步記憶體的時脈頻率及記憶體致能訊號。
- WR\_EN：記憶體寫入致能訊號。
- WR\_BYTE\_SEL：選擇要寫入訊號的位元組位置，共有八個位元組。
- WR\_ADDR：記憶體寫入位址，為一個 12 位元的計數器的輸出。而 CNT\_EN 及 SCLR 分別控制此計數器之計數致能及清除的功能。
- RD\_ADDR：記憶體讀出位址，為兩個計數器輸出相加的結果。其中一個 12 位元計數器，在其輸出資料的最高位元加上一個零 (Zero -Padding) 後為 13 位元的 Base\_Addr；另一個 8 位元計數器，在其輸出資料的最高位元加上五個零(Zero-Padding)後為 13 位元的 Offset\_Addr。最後由一個 13 位元無號加法器將 Base\_Addr 及 Offset \_Addr 兩個訊號相加，並去掉最高位元為 RD\_ADDR 記憶體讀取位址。

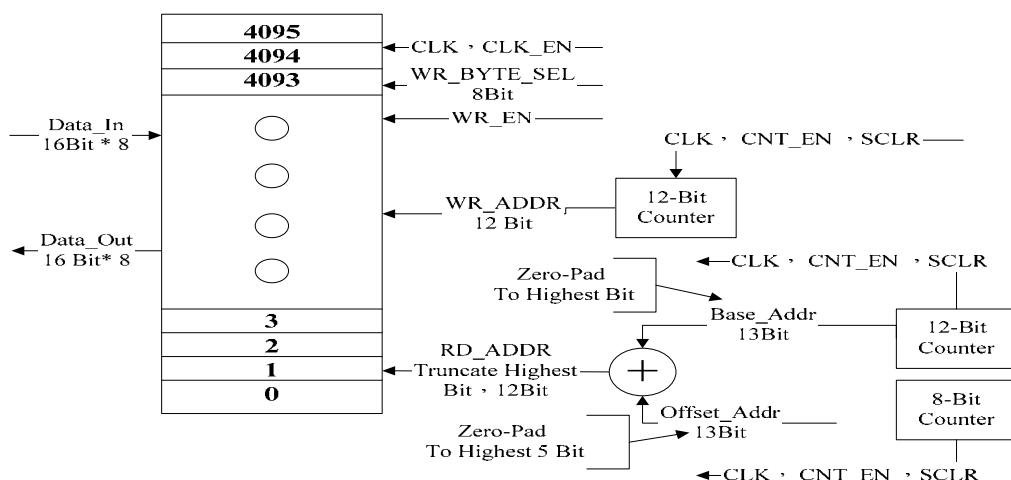


圖 3-31：欲純化語音或雜訊記憶體模組單元

### 3.4.8.2 資料記憶體模組

如圖 3-32 為資料記憶體模組單元。共有八個輸出埠（接至空間濾波器運算單元中的訊號輸入埠）和八個輸入埠（接至 8 CH A/D Board 的資料匯流排），皆為 16 位元寬度，分別儲存八個頻道各 16 位元的資料。其他訊號說明如下：

- CLK 及 CLK\_EN：分別為此同步記憶體的時脈頻率及記憶體致能訊號。
- WR\_EN 及 RD\_EN：記憶體寫入致能及讀取致能訊號。
- WR\_ADDR：記憶體寫入位址，為一個 11 位元的計數器的輸出。而 CNT\_EN 及 SCLR 分別控制此計數器之計數致能及清除的功能。
- RD\_ADDR：記憶體讀出位址，為兩個計數器輸出相加的結果。其中一個 11 位元計數器，在其輸出資料的最高位元加上一個零 (Zero -Padding) 後為 12 位元的 Base\_Addr；另一個 8 位元計數器，在其輸輸出資料的最高位元加上四個零(Zero-Padding)後為 12 位元的 Offset\_Addr。最後由一個 12 位元無號加法器將 Base\_Addr 及 Offset\_Addr 兩個訊號相加，並去掉最高位元為 RD\_ADDR 記憶體讀取位址。

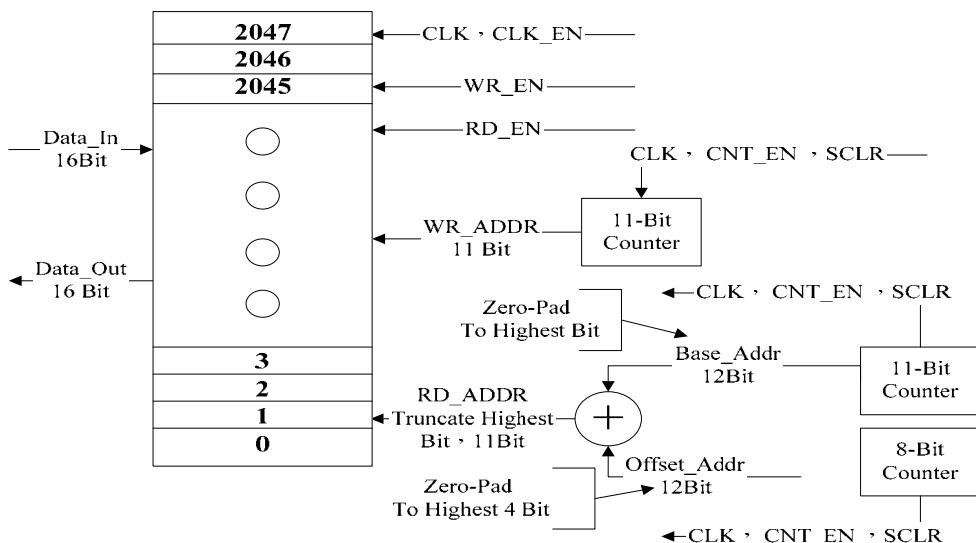


圖 3-32：資料記憶體模組單元

### 3.4.8.3 濾波器係數記憶體模組

如圖 3-33 為濾波器係數記憶體模組單元。共有十六個輸出埠，(其中 A 埠接至濾波器係數更新模組運算單元中的濾波器係數  $w(i-1)$  輸入端；B 埠接至空間濾波器運算單元中的濾波器係數輸入埠) 和八個輸入埠 (接至濾波器係數更新模組運算單元中的濾波器係數  $w(i)$  輸入端)，皆為 16 位元寬度，分別儲存八個頻道的濾波器係數。其他訊號說明如下：

- CLK 及 CLK\_EN：分別為此同步記憶體的時脈頻率及記憶體致能訊號。
  - WR\_EN：記憶體寫入致能訊號。
  - RD\_EN\_A 及 RD\_EN\_B：記憶體 A 埠及 B 埠讀取致能訊號。
  - WR\_ADDR：記憶體寫入位址，為一個 8 位元的計數器的輸出。而 CNT\_EN 及 SCLR 分別控制此計數器之計數致能及清除的功能。
  - RD\_ADDR：記憶體讀出位址，為一個 8 位元的計數器的輸出。而 CNT\_EN 及 SCLR 分別控制此計數器之計數致能及清除的功能。

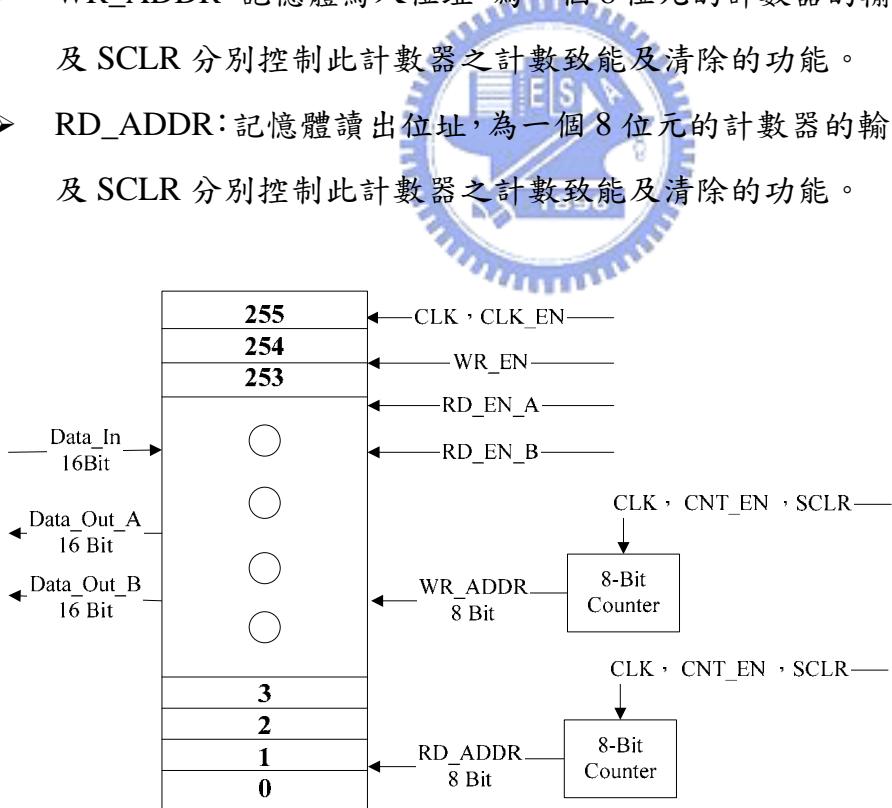


圖 3-33：濾波器係數記憶體模組單元

### 3.4.9 Nios 微處理器系統軟體架構

如圖 3-34 所示為 Nios 微處理器系統軟體流程圖。而圖 3-35 為系統軟體之時序分析圖。

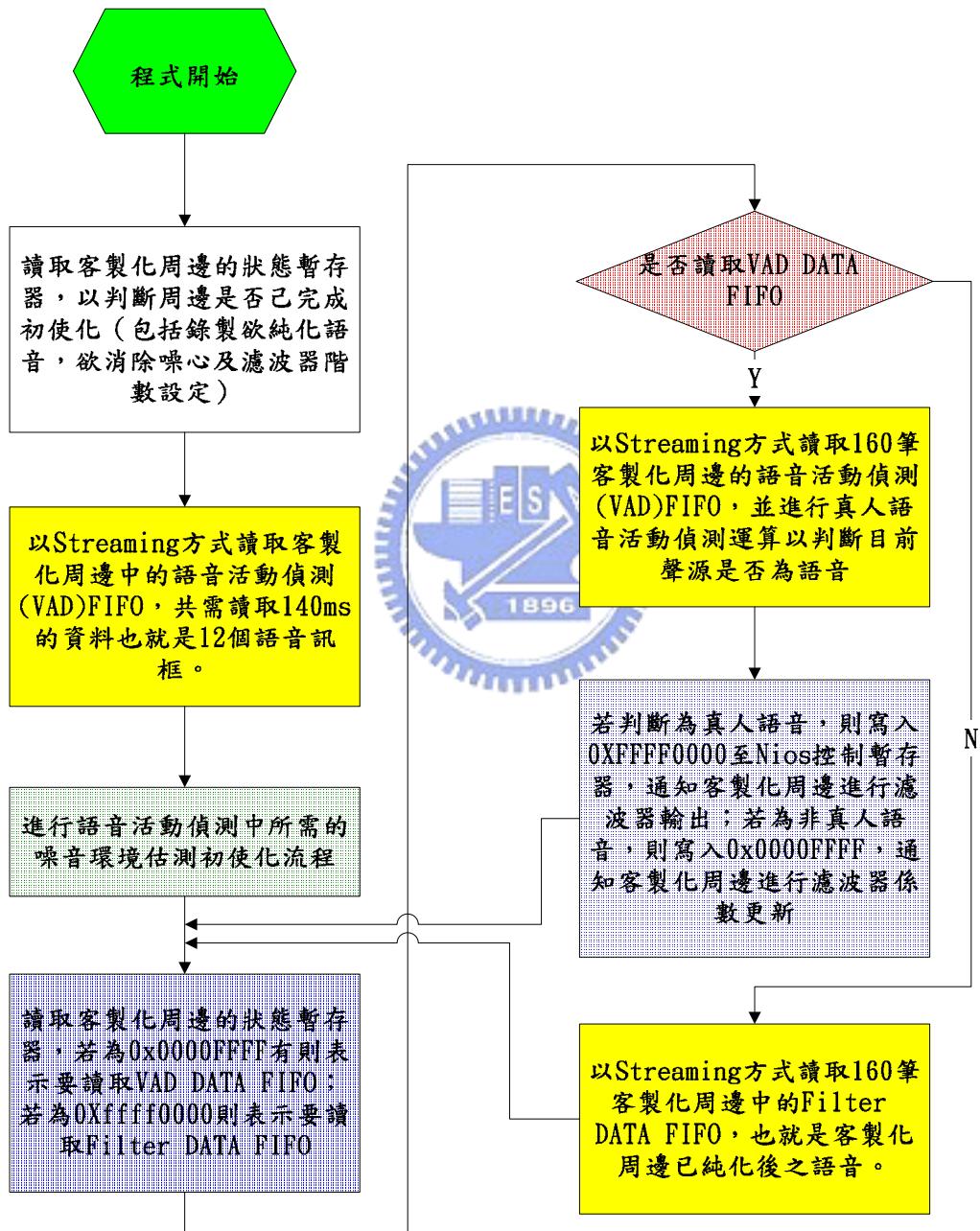


圖 3-34：Nios 微處理器系統軟體流程圖

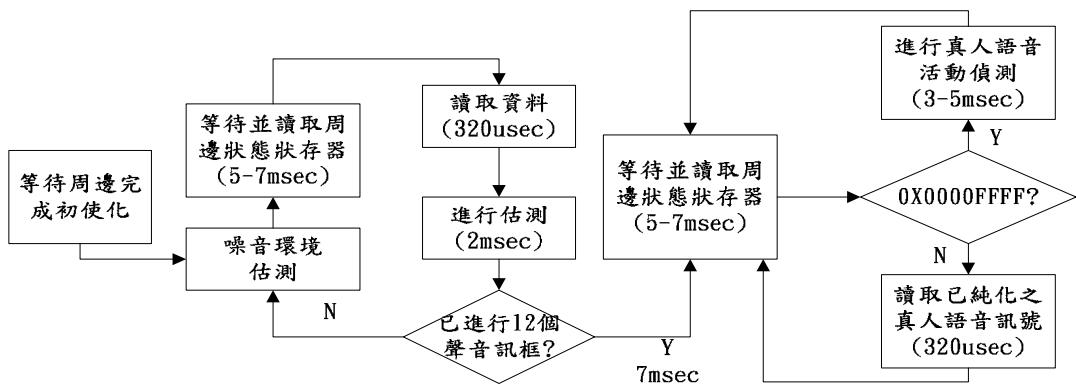


圖 3-35：系統軟體之時序分析圖

### 3.5 實驗平台實際照片

圖 3-36 為實驗平台實際照片，包含了：

1. 麥克風陣列（間距為 6cm）
2. 系統電路板
3. Nios 嵌入式 SOPC 發展平台平台
4. PC

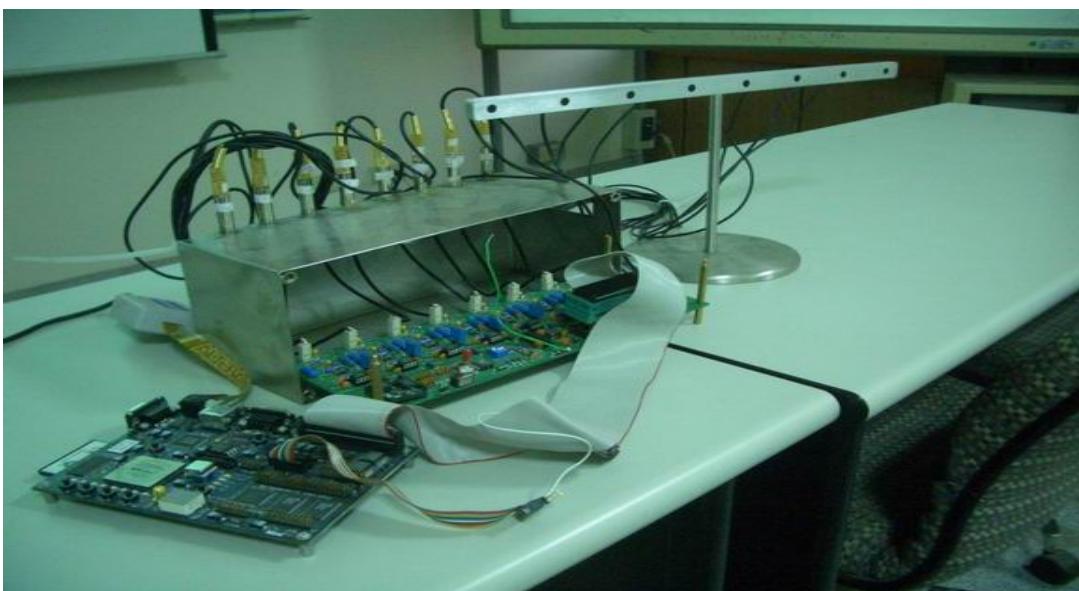


圖 3-36：實驗平台實際照片

## 第四章 實驗結果與分析

本章節將介紹將麥克風陣列平台於不同噪音環境以及不同濾波器階數情況下測試的結果，針對此語音純化系統所能改善的訊號雜訊比做探討，而實驗環境則在室內環境。

### 4.1 麥克風陣列於室內環境

圖 4-1 為實驗環境的實際照片，實驗環境中有兩個喇叭，一個喇叭用來撥放人聲，另一個為撥放音樂聲，而圖中的電腦為用來將純化後之語音以 debug mode 方式儲存下來。而圖 4-2 為實驗環境的平面關係圖，其中我們選擇兩個喇叭與麥克風陣列的角度差為 30 度，使得兩個聲源在空間上的分佈具有差異性。



圖 4-1：實驗環境實際照片

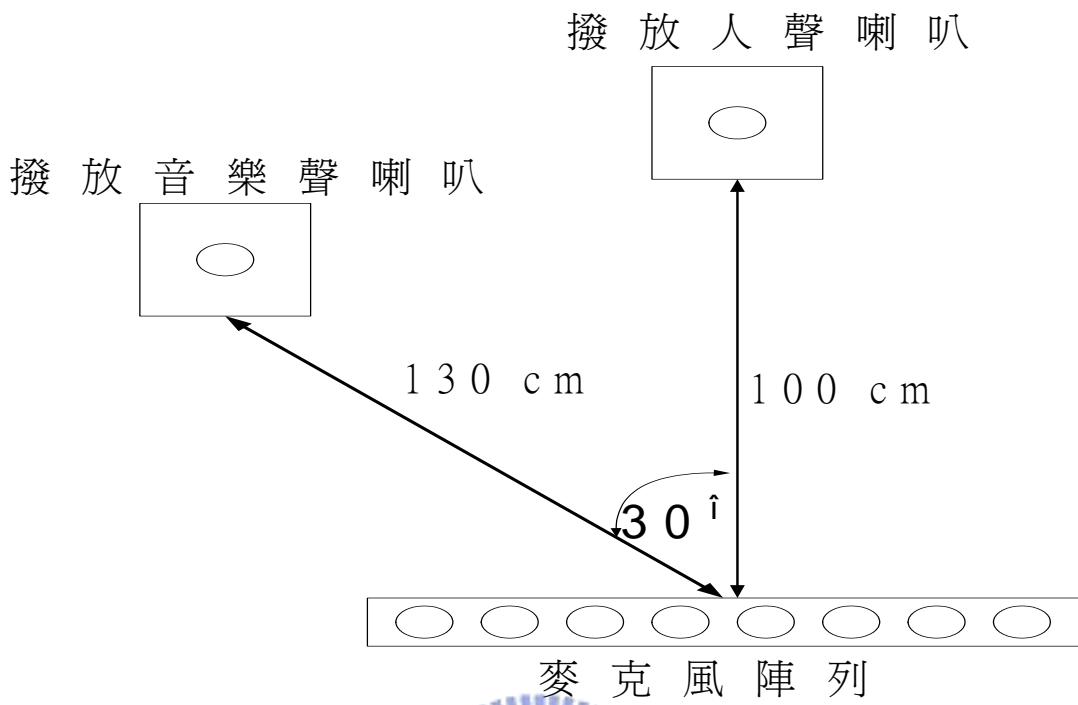


圖 4-2：實驗環境平面關係圖



#### 4.1.1 空間濾波器結合真人語音活動偵測與訊號雜訊比關係

首先，將人聲錄製一小段語音為“喂! 123”，並在下列二種情況下撥放測試其訊號雜訊比：

1. 音樂聲
2. 白色雜訊

實驗中 SNR 的計算方式如下：

$$10 \log \left( \frac{\sum_{i=M}^N x^2(i)}{N - M + 1} \right) \quad (4-1)$$

假設雜訊為第  $M_1$  到第  $N_1$  筆，而語音加雜訊為第  $M_2$  到第  $N_2$  筆，其 SNR

為

$$10 \log \left( \frac{\sum_{i=M_2}^{N_2} x^2(i)}{N_2 - M_2 + 1} \right) - 10 \log \left( \frac{\sum_{i=M_1}^{N_1} x^2(i)}{N_1 - M_1 + 1} \right) \quad \text{dB} \quad (4-2)$$

測試一：測試在音樂聲（流行歌曲：孫燕姿-奔）雜訊環境下，且有預先錄製音樂聲雜訊，在不同濾波器階數下訊號雜訊比的變化。

#### ➤ 空間濾波器為 64 階

圖 4-3 中音樂聲為-32.2dB，而人聲摻雜音樂聲為-17.4 dB，因此 SNR = 14.8 dB。圖 4-4 為經過 64 階空間濾波器的處理，音樂聲為-38.7 dB，而人聲摻雜音樂聲為-16.2 dB，因此 SNR=22.5 dB。故經過 64 階空間濾波器處理後，SNR 比提升了 7.8dB。

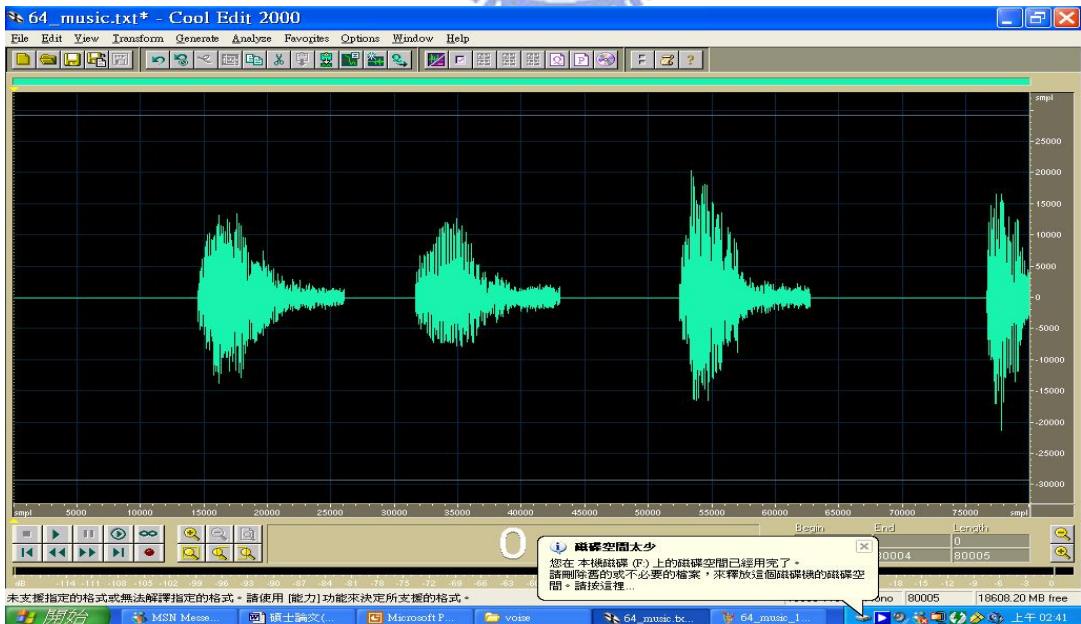


圖 4-3：空間濾波器處理前，音樂雜訊

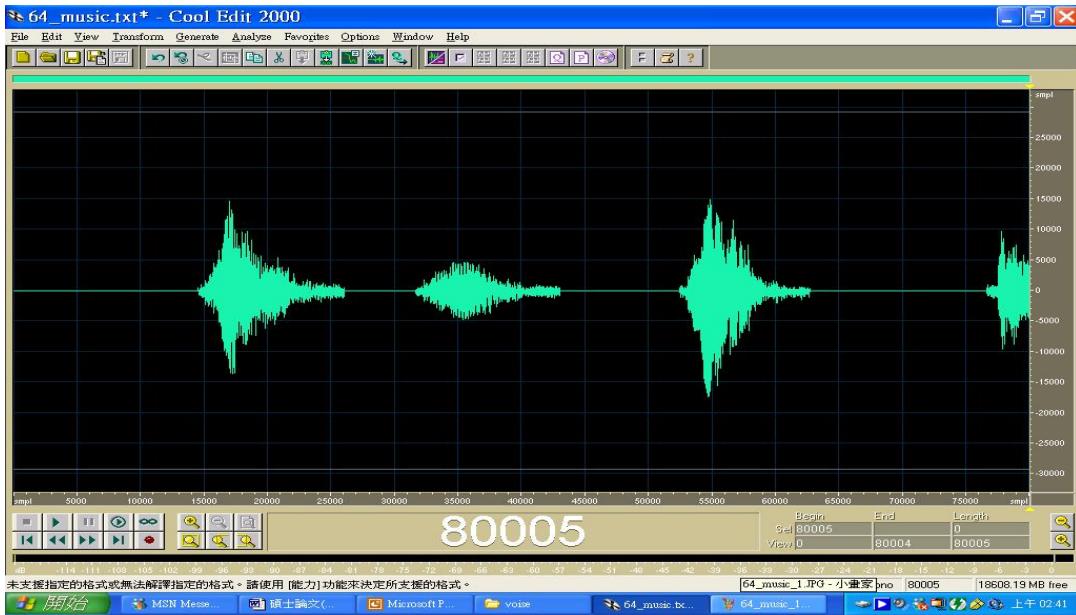


圖 4-4：空間濾波器處理後，音樂雜訊，階數=64

#### ➤ 空間濾波器為 128 階

圖 4-5 中音樂聲為-32.2dB，而人聲摻雜音樂聲為-15.9 dB，因此  $\text{SNR} = 16.3 \text{ dB}$ 。圖 4-6 為經過 128 階空間濾波器的處理，音樂聲為-38.2dB，而人聲摻雜音樂聲為-17.4 dB，因此  $\text{SNR}=20.8 \text{ dB}$ 。故經過 128 階空間濾波器處理後，SNR 比提升了 4.5dB。

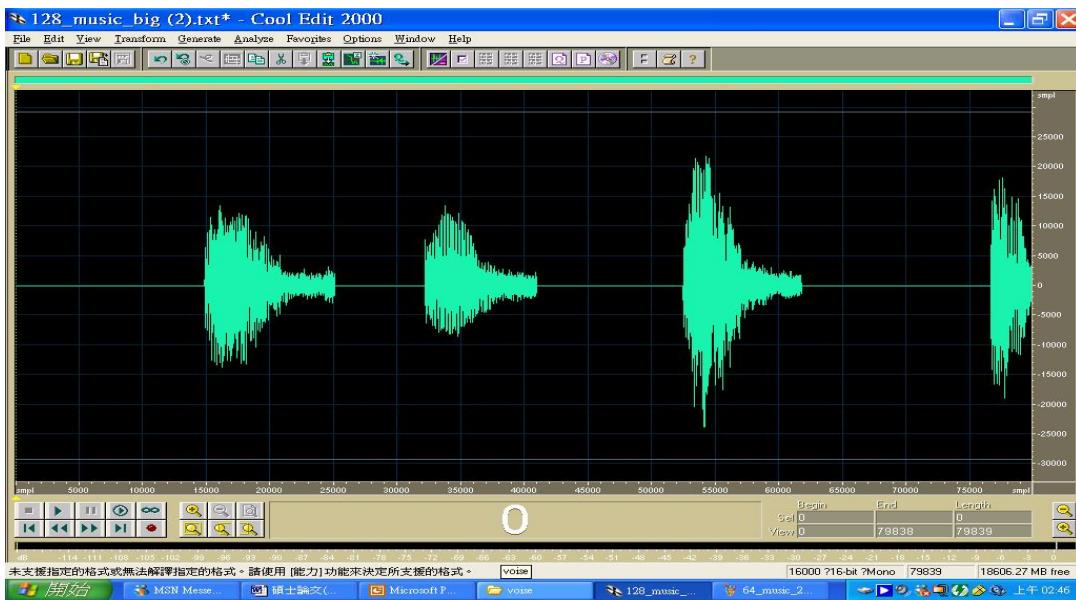


圖 4-5：空間濾波器處理前，音樂雜訊



圖 4-6：空間濾波器處理後，音樂雜訊，階數=128

#### ➤ 空間濾波器為 256 階

圖 4-7 中音樂聲為 $-32.5\text{dB}$ ，而人聲摻雜音樂聲為 $-15.4\text{ dB}$ ，因此  $\text{SNR} = 17.1\text{ dB}$ 。圖 4-8 為經過 256 階空間濾波器的處理，音樂聲為 $-42.8\text{dB}$ ，而人聲摻雜音樂聲為 $-16.3\text{ dB}$ ，因此  $\text{SNR}=26.5\text{ dB}$ 。故經過 256 階空間濾波器處理後， $\text{SNR}$  比提升了  $9.4\text{dB}$ 。

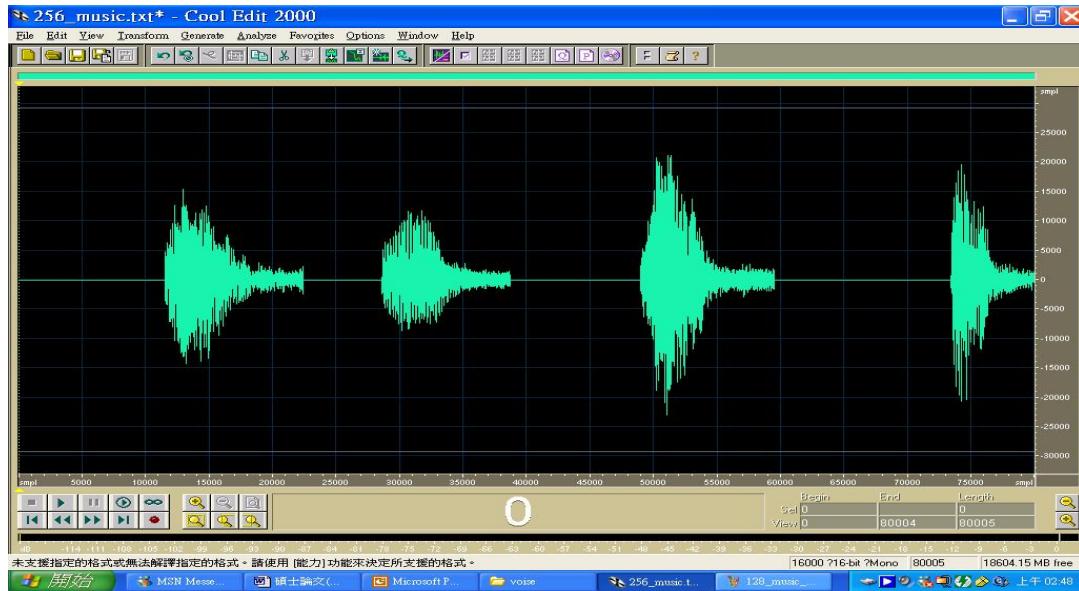


圖 4-7：空間濾波器處理前，音樂雜訊

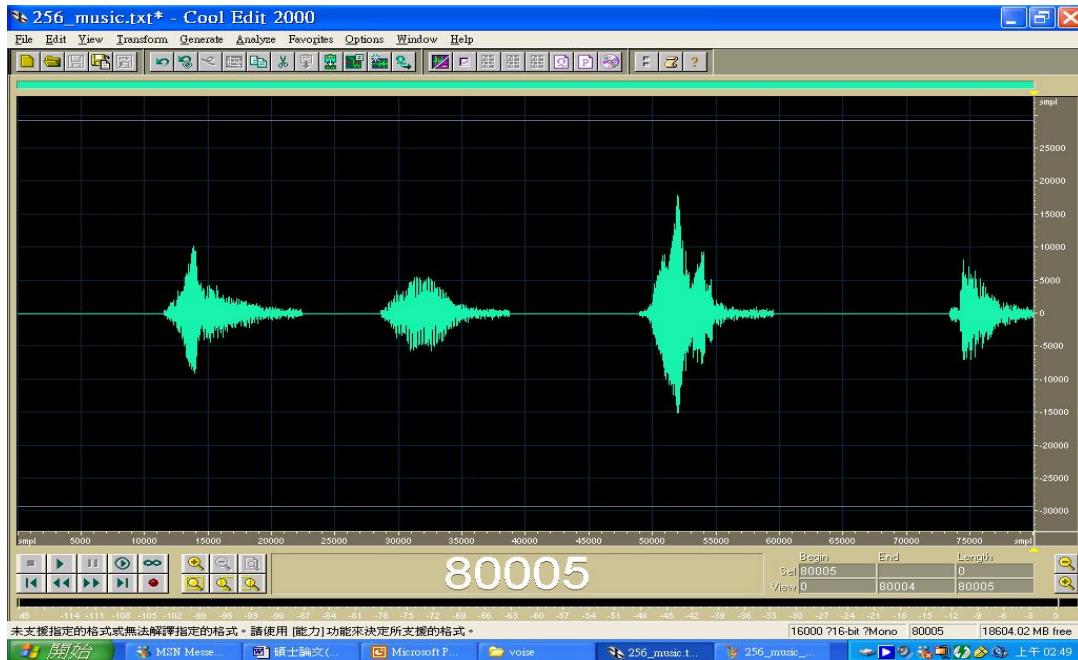


圖 4-8：空間濾波器處理後，音樂雜訊，階數=256

### 測試一總結：

以上在音樂聲的雜訊環境下，三種濾波器階數皆有一定程度的 SNR 比提升，其中又以最高的 256 階有最好的 SNR 比提升效果。

**測試二：**測試在音樂聲（流行歌曲：孫燕姿-奔）雜訊環境下，但沒有預先錄製音樂聲雜訊，在不同濾波器階數下訊號雜訊比的變化。

### ➤ 空間濾波器為 128 階

圖 4-9 中音樂聲為-32.4dB，而人聲摻雜音樂聲為-15.1 dB，因此  $SNR = 17.3 \text{ dB}$ 。圖 4-10 為經過 128 階空間濾波器的處理，音樂聲為-42.5dB，而人聲摻雜音樂聲為-17.3 dB，因此  $SNR=25.2 \text{ dB}$ 。故經過 128 階空間濾波器處理後，SNR 比提升了 7.9dB。

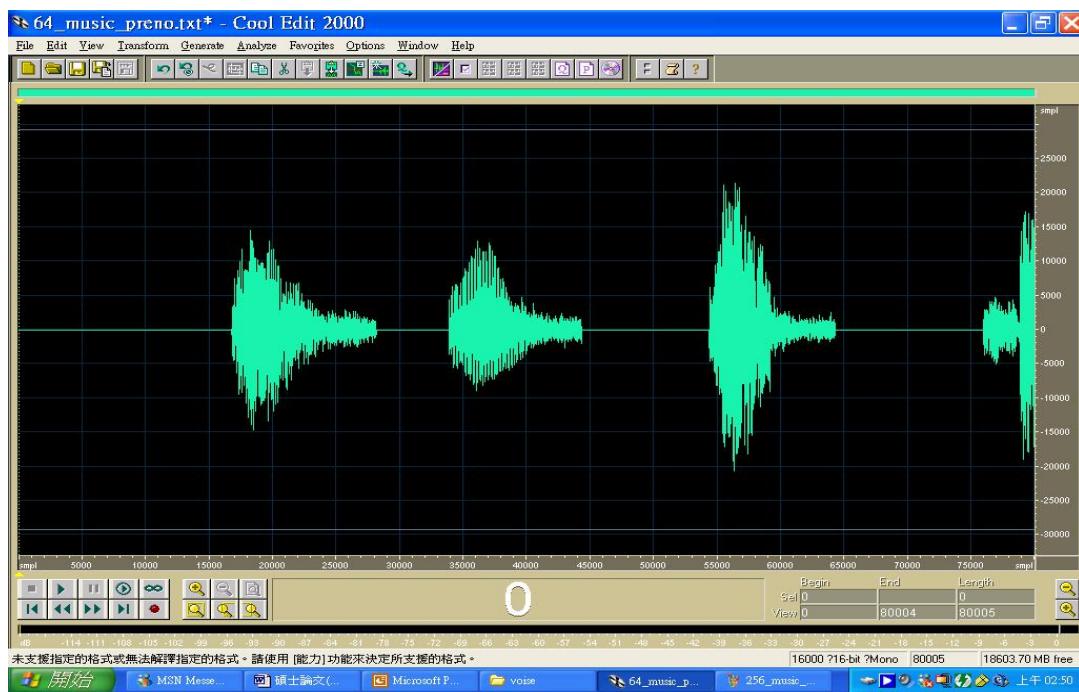


圖 4-9：空間濾波器處理前，音樂雜訊無預先錄製

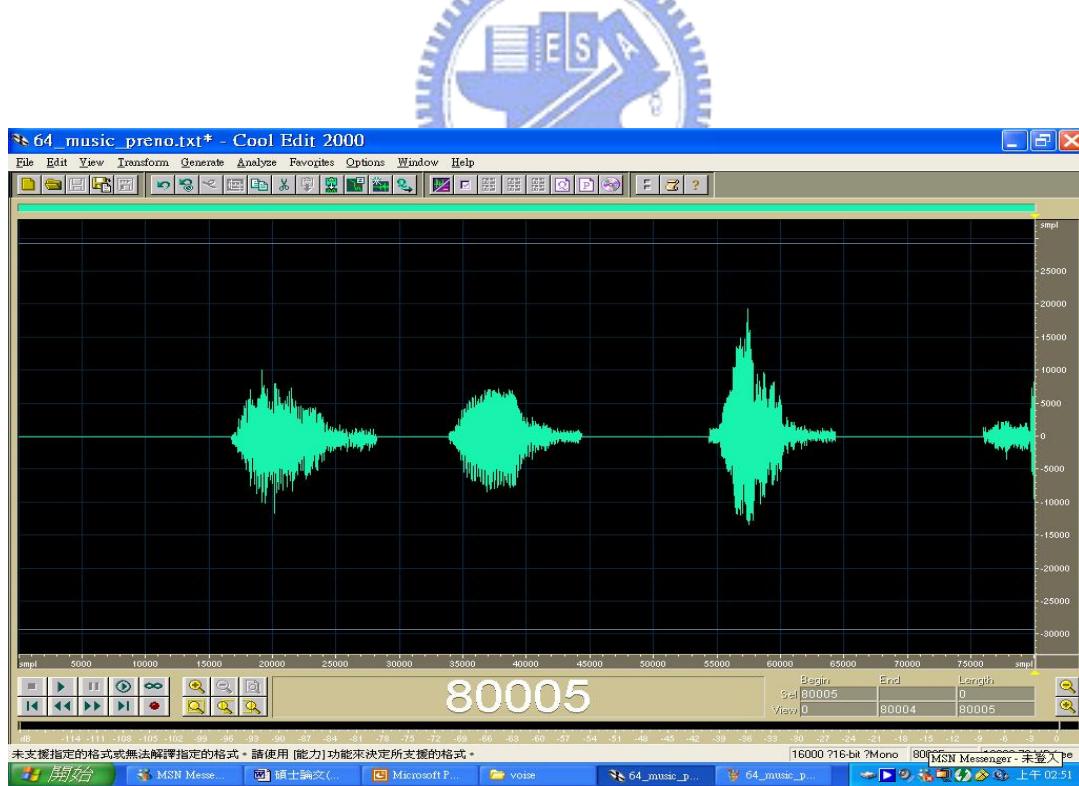


圖 4-10：空間濾波器處理後，音樂雜訊無預先錄製，階數為 128

## ➤ 空間濾波器為 256 階

圖 4-11 中音樂聲為-31.5dB，而人聲摻雜音樂聲為-15.4 dB，因此 SNR = 16.1 dB。圖 4-12 為經過 256 階空間濾波器的處理，音樂聲為-41.8dB，而人聲摻雜音樂聲為-16.3 dB，因此 SNR=25.5 dB。故經過 256 階空間濾波器處理後，SNR 比提升了 9.4dB。

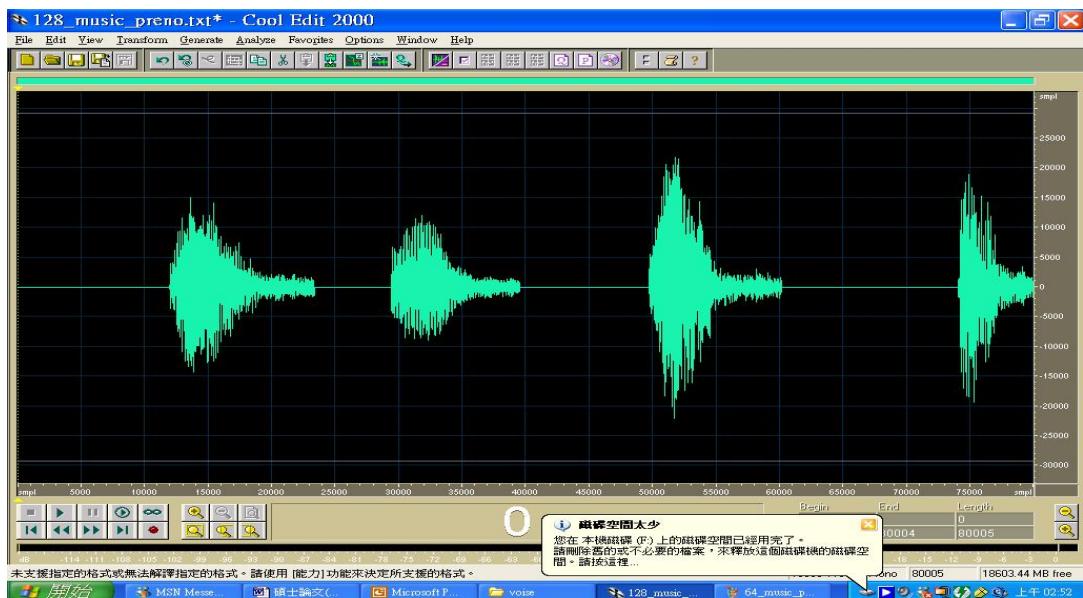


圖 4-11：空間濾波器處理前，音樂雜訊無預先錄製

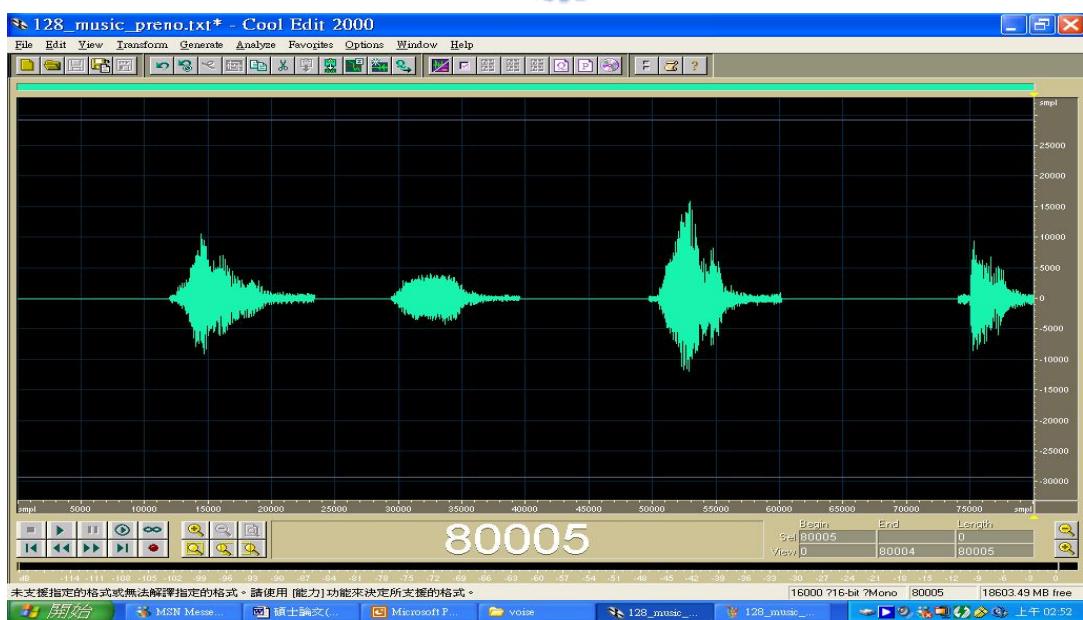


圖 4-12：空間濾波器處理後，音樂雜訊無預先錄製，階數為 256

## 測試二總結：

由上面的測試可得知，即使雜訊無預先錄製的情況下，仍可透過適應性濾波器係數調整（當真人語音判斷為零時），將新的雜訊考慮進來，達到抑制雜訊的效果。而上面測試得知，256 階所得到的雜訊壓制效果比 128 階好，兩者之間的 SNR 比差了 1.5dB。

## 測試三：測試在白色雜訊環境下，且有預先錄製白色雜訊，在不同濾波器階數下訊號雜訊比的變化。

### ➤ 空間濾波器為 128 階

圖 4-13 中白色雜訊為-38.5dB，而人聲摻雜白色雜訊為-15.7 dB，因此 SNR = 22.8 dB。圖 4-14 為經過 128 階空間濾波器的處理，白色雜訊為-44.8dB，而人聲摻雜白色雜訊為-14.3 dB，因此 SNR=30.5 dB。故經過 128 階空間濾波器處理後，SNR 比提升了 7.7dB。

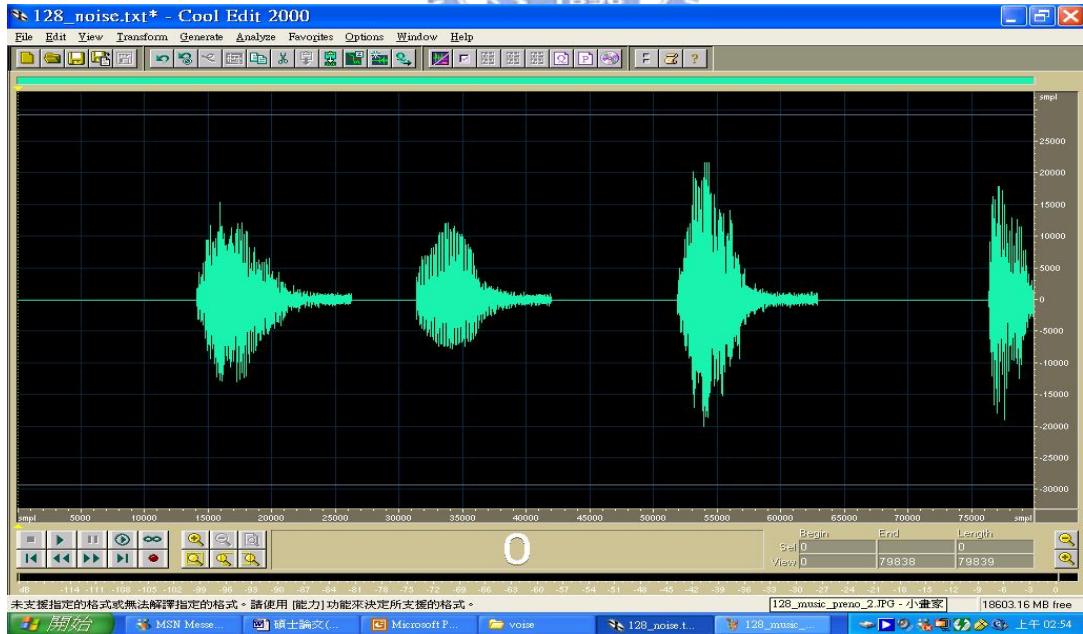


圖 4-13：空間濾波器處理前，白色雜訊

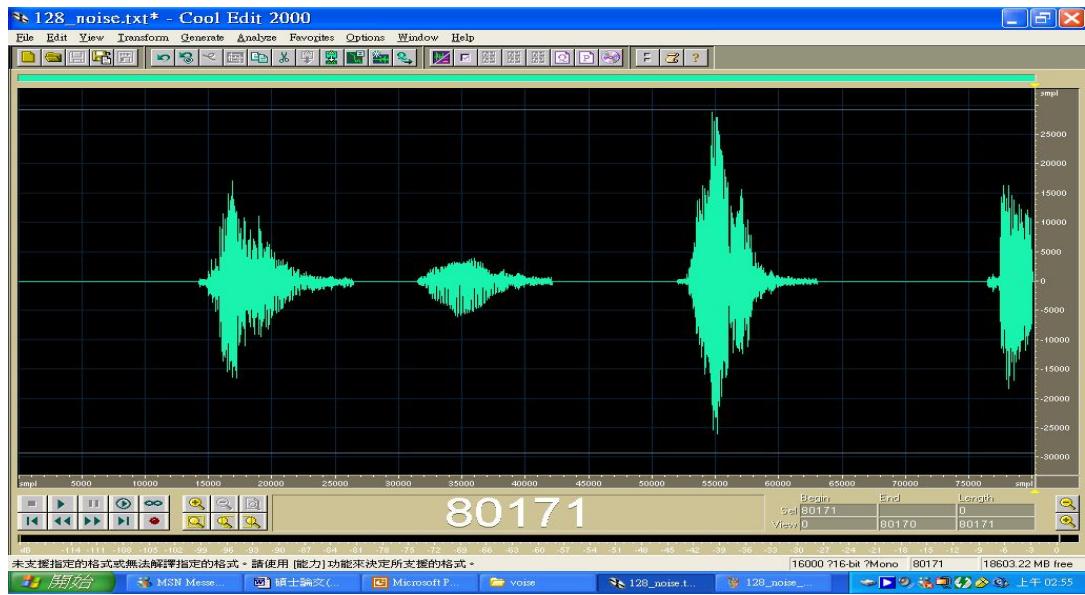


圖 4-14：空間濾波器處理後，白色雜訊，階數為 128

#### ➤ 空間濾波器為 256 階

圖 4-13 中白色雜訊為-38.2dB，而人聲摻雜白色雜訊為-18.5 dB，因此 SNR = 19.7 dB。圖 4-14 為經過 256 階空間濾波器的處理，白色雜訊為-45.8dB，而人聲摻雜白色雜訊為-14.8 dB，因此 SNR=31 dB。故經過 256 階空間濾波器處理後，SNR 比提升了 11.3dB。

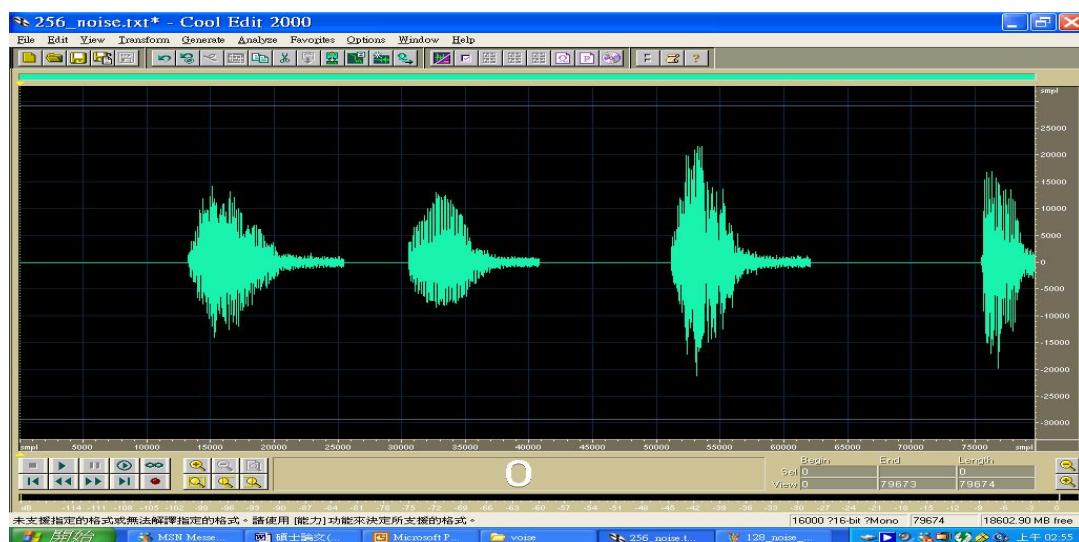


圖 4-15：空間濾波器處理前，白色雜訊

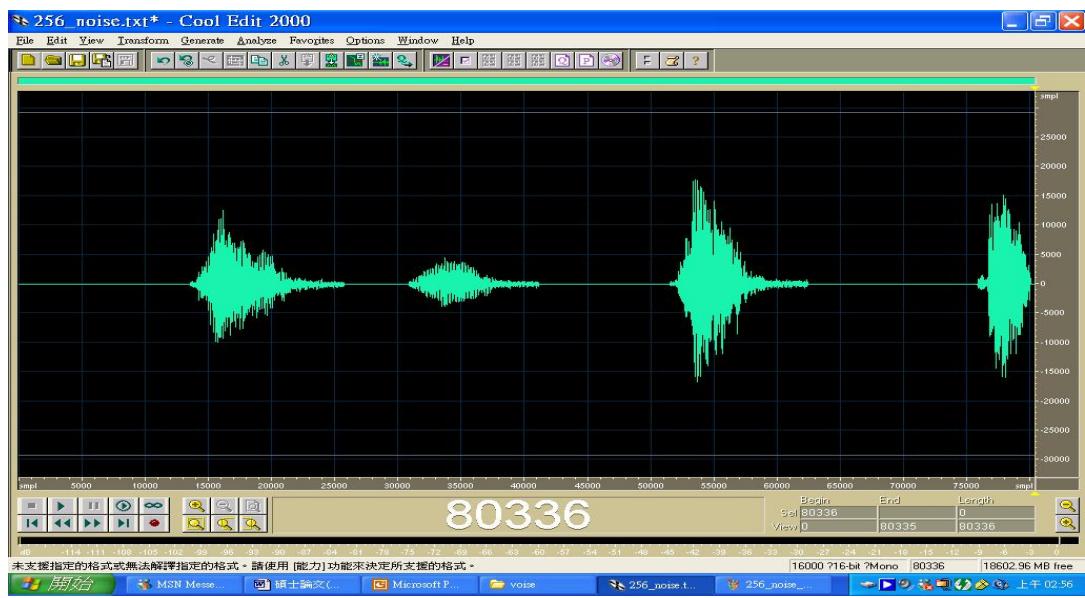


圖 4-16：空間濾波器處理後，白色雜訊，階數為 256

### 測試三總結：

由上面測試可知，對於白色雜訊，空間濾波器仍有效的抑制雜訊。而上面測試得知，256 階所得到的雜訊壓制效果比 128 階好，兩者之間的 SNR 比差了 2.5dB。



# 第五章 結論

## 5.1 總結

本論文已實作完成一嵌入式 SOPC 系統之 8 通道及時性麥克風陣列訊號處理平台，並在此平台實作完成真人語音偵測（VAD）與適應性空間濾波器整合，可使用於室內，測試結論為：

1. 於吵雜環境下使用麥克風陣列，其訊號雜訊比值較使用單一麥克風高。
2. 空間濾波器階數的提高可以有提升其 SNR 的機會。
3. VAD 與空間濾波器的結合可應用於噪音源位置變動的環境，若只使用空間濾波器則噪音源位置改變，則須手動方式重新啟動 LMS，使其重新調整空間濾波器係數，若加入 VAD 則當沒真人語音時，系統自動調整空間濾波器係數。
4. VAD 結合空間濾波器適用於噪音源變動的環境。



而本論文所實現的嵌入式 SOPC 單晶片系統軟硬體設計則具有以下特色

➤ 具有彈性且平行處理的 DSP 電路架構設計：

由於此語音純化系統中使用適應性空間濾波器，屬於多通道訊號處理。所以在硬體設計上採用 8 個 MAC 的架構，可以平行處理 8 個階數可任意變動的濾波器，並且在濾波器輸出及濾波器係數更新上共用一組硬體，以達到具彈性且省硬體資源的 DSP 電路架構。在效能上，不同濾波器階數可以適應不同噪音環境的特性（如噪音環境變化較為劇烈時採用較低階以達到快速的收斂性；反之則選擇高階以達到較小的穩態誤差）。

➤ 以客製化周邊的硬體架構進行軟硬體整合：

此語音純化系統使用客製化周邊的設計進行硬體加速（以達到需要大量乘累加運算的效能），且將運算量較低的語音活動偵測（VAD）交由 NIOS 運算，並透過客製化周邊的 Control & Status register 來達到軟硬體間的溝通。如此的系統架構，

即使在不同的匯流排傳輸架構上，也可以快速的移植。

➤ 使用 streaming 傳輸方式，搭配 On-Chip-Memory 的使用，增進系統效能：

配合 SOPC Builder 已有的 DMA Controller，客製化周邊以 streaming 的傳輸方式增加系統效能。並且使用.section 假指令，將 NIOS 程式中運算量大的區塊放置於 On-Chip-Memory，更進一步增加系統效能。

## 5.2 未來展望

在做 LMS 演算法時，必須先針對雜訊作出 zero-mean 及 Gaussian 的假設，若雜訊本身並非 zero-mean 及 Gaussian，則 Normalize LMS 處理效果會有限，但現實生活中並非所有接收到的雜訊皆為 zero-mean 及 Gaussian，因此未來日子可考慮其他適應性訊號處理演算法，像  $H^\infty$  等。而在本系統的 SOPC 硬體電路為求得高的誤差收斂效果，運算單元因此耗掉比較多的硬體資源，往後期望在 DSP 硬體技術及演算法上花費更多功夫，以求達到改善效能及省硬體的目標。並且在嵌入式 SOPC 系統發展平台上，實現嵌入式 Ethernet 網路傳輸，並進行即時語音辨識的工作。



## Reference

- [1]D. Johnson and D. Dudgeon, *Array Signal Processing: Concepts and Techniques*, Prentice Hall, Englewood Cliffs, New Jersey, 1993.
- [2]J. L. Flanagan, L. Landgraf, D. J. McLean, "Matched-filter processing of hydrophone array", *J. Acousr. Soc. Am.* 42,1165(A)(1967).
- [3]Barry D. Van Veen and Kevin M.Buckley,"Beamforming:A Versatile Approach to Spatial Filtering,"*IEEE ASSP MAGAZINE* April 1988.
- [4]Zoltowski, M., "High resolution sensor array signal processing in the beamspace domain: novel techniques based on the poor resolution of Fourier beamforming," *Spectrum Estimation and Modeling*, 1988., Fourth Annual ASSP Workshop on , pp. 350 –355, 1988
- [5]Byung-Chul Kim; I-Tai Lu , "High resolution broadband beamforming based on the MVDR method," *OCEANS 2000 MTS/IEEE Conference and Exhibition* , Volume: 3 , pp. 1673 –1676, 2000
- [6]Marciano, J.S., Jr.; Vu, T.B., "Reduced complexity MVDR broadband beamspace beamforming under frequency invariant constraint," *Antennas and Propagation Society International Symposium*, 2000. IEEE , Volume: 2 , pp. 902 –905, 2000
- [7]Pillai, S. Unnikrishna, *Array signal processing*, 1989
- [8]Ta-Sung Lee; Tsui-Tsai Lin , "Coherent interference suppression with complementally transformed adaptive beamformer," *Antennas and Propagation, IEEE Transactions on* , Volume: 46 Issue: 5 , pp. 609 –617, May 1998
- [9]Gollamudi, S.; Yih-Fang Huang , "Optimally combined nonlinear MMSE beamforming and interference cancellation for CDMA communications," *Personal Wireless Communications*, 2000 IEEE International Conference on , pp. 474 –478, 2000
- [10]黃佑霖,"應用延遲濾波器之麥克風陣列訊號處理",交大電控碩士論文,Jun 1999
- [11]陳界全,"即時聲源追蹤與空間濾波器設計",交大電控碩士論文,Jun 2000
- [12]康創閔,"應用於個人電腦環境之即時語音純化系統設計",交大電控碩士論文,Jul 2004
- [13]Ta-Sung Lee, *Array Signal Processing*,(class note)
- [14]Javier Ramírez , José C. Segura , Carmen Benítez , Ángel de la Torre and Antonio Rubio , "Efficient voice activity detection algorithms using

- long-term speech information,” Speech Communication, Volume 42, Issues 3-4, April 2004, Pages 271-287
- [15]European Digital Cellular Telecommunications System; Half rate speech; Voice Activity Detection (VAD), ETSI GSM 06.42 (ETSI 300-581-6), 1995.
- [16]European Digital Cellular Telecommunications System; Half rate speech; Half rate speech transcoding, ETSI GSM 06.20 (ETSI 300-581-2), 1995.
- [17]ITU-T G.729, Coding of Speech at 8kbit/s Using CS-ACELP, March, 1996.
- [18]A. Benyassine, E. Shlomot, H. Y. Su, D. Massaloux, C. Lamblin, and J. P. Petit, “ITU recommendation G.729 annex B: A silence compression scheme for use with G.729 optimized for V.70 digital simultaneous voice and data applications,” IEEE Commun. Mag., vol. 35, pp. 64–73, Sept. 1997.
- [19]Ali H. Sayed, Fundamentals of Adaptive Filtering
- [20]Dahl, M.; Claesson, I., “Acoustic noise and echo cancelling with microphone array,” Vehicular Technology, IEEE Transactions on , Volume: 48 Issue: 5 , Sept.1999 Page(s): 1518 –1526
- [21]Quartus II Handbook
- [22]Nios II Processor Reference Handbook , ALTERA Corp.
- [23]Nios II Software Development Handbook , ALTERA Corp.
- [24]Stratix Device Handbook, Volume 1 , ALTERA Corp.
- [25]Stratix Device Handbook, Volume 2 , ALTERA Corp.
- [26]Avalon Interface Specification , ALTERA Corp.
- [27]Nios Development Board Reference Manual, Stratix Professional Edition , ALTERA Corp.
- [28]David Halupka ; Nebu John Mathai ; Parham Arrabi ; Sheikholeslami “Robust Sound Location in 0.18um CMOS” IEEE TRANSACTIONS ON SIGNAL PROCESSING 53 (6): 2243-2250 JUN 2005
- [29]Chin-Liang Wang ”Bit-Serial VLSI Implementation of Delayed LMS Adaptive FIR Filters” Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on [see also Circuits and Systems II: Express Briefs, IEEE Transactions on]Volume 41, Issue 11, Nov. 1994 Page(s):768 – 772
- [30]Chin-Liang Wang; Ching-Chia Chen; Che-Fu Chang”A digit-serial VLSI architecture for delayed LMS adaptive FIR filtering”Circuits and Systems, 1995. ISCAS '95., 1995 IEEE International Symposium on Volume 1, 28 April-3 May 1995 Page(s):545 - 548 vol.1
- [31]Allred, D.; Krishnan, V.; Huang, W.; Anderson, D.” Implementation of an LMS adaptive filter on an FPGA employing multiplexed multiplier architecture” Signals, Systems and Computers, 2003. Conference Record of the Thirty-Seventh Asilomar Conference on Volume 1, 9-12 Nov. 2003 Page(s):918 - 921 Vol.1
- [32]Rissa, T.; Uusikartano, R.; Niittylahti, J.” Adaptive FIR filter architectures for run-time reconfigurable FPGAs” Field-Programmable Technology, 2002. (FPT). Proceedings. 2002 IEEE International Conference on 16-18 Dec. 2002

Page(s):52 – 59

- [33]Dukel, B.; Rizkalla, M.E.; Salama, P." Implementation of pipelined LMS adaptive filter for low-power VLSI applications" Circuits and Systems, 2002. MWSCAS-2002. The 2002 45th Midwest Symposium on Volume 2, 4-7 Aug. 2002 Page(s):II-533 - II-536 vol.2
- [34]Huang, W.; Krishnan, V.; Allred, D.; Heejong Yoo" Design analysis of a distributed arithmetic adaptive FIR filter on an FPGA" Signals, Systems and Computers, 2003. Conference Record of the Thirty-Seventh Asilomar Conference on Volume 1, 9-12 Nov. 2003 Page(s):926 - 930 Vol.1
- [35]Allred, D.J.; Heejong Yoo; Krishnan, V.; Huang, W.; Anderson, D.V. "A novel high performance distributed arithmetic adaptive filter implementation on an FPGA" Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on Volume 5, 17-21 May 2004 Page(s):V - 161-4 vol.5
- [36]Herzberg, H.; Haimi-Cohen, R. "A systolic array realization of an LMS adaptive filter and the effects of delayed adaptation" Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on] Volume 40, Issue 11, Nov. 1992 Page(s):2799 – 2803
- [37]Lan-Da Van; Wu-Shiung Feng "Efficient systolic architectures for 1-D and 2-D DLMS adaptive digital filters" Circuits and Systems, 2000. IEEE APCCAS 2000. The 2000 IEEE Asia-Pacific Conference on 4-6 Dec. 2000 Page(s):399 – 402
- [38]Lan-Da Van; Wu-Shiung Feng "An efficient systolic architecture for the DLMS adaptive filter and its applications" Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on [see also Circuits and Systems II: Express Briefs, IEEE Transactions on Volume 48, Issue 4, April 2001 Page(s):359 - 366
- [39]Yi, Y, Woods, R, Ting, LK, et al." High speed FPGA-based implementations of delayed-LMS filters" JOURNAL OF VLSI SIGNAL PROCESSING SYSTEMS FOR SIGNAL IMAGE AND VIDEO TECHNOLOGY 39 (1-2): 113-131 JAN-FEB 2005

