



# 空-時犁狀接收器之演算法及其軟體無線電 之實現

## 中文摘要

軟體無線電架構的發展，近幾年來已成為一個極受矚目的研究領域。在本報告中，吾人首先介紹軟體無線電(software radio)平台之概念，並整合寬頻分碼多重進接(W-CDMA)技術，發展出結合智慧型陣列天線訊號處理之空-時犁狀接收器(space-time RAKE receiver)。由模擬與實作結果顯示，在多重進接干擾(MAI)的環境下，此架構能有效降低誤碼率(BER)，提高系統效能。系統架構採用參數載入式設計，使其具備軟體無線電要求之可程式化與原件模組化特性，並可透過修改參數，輕易變更系統設定，以符合在不同環境下之通訊規格。吾人根據TMS320C6201 DSP 數位信號處理器應用模板來模擬及實現所提之架構，並配合其硬體特性，如平行運算處理單元，環狀定址模式與高速化多功能串列埠等，發展出一具備可程式與相容於第三代行動通訊系統之接收器基頻處理模組。

# **Algorithm and Software Radio Realization of Space-Time RAKE Receiver**

## **Abstract**

Recently, the development of software define radio has been an interesting research area in wireless communication signal processing. In this report, we first introduce the concept of software define radio. In the following, we propose a smart array antenna based space-time RAKE receiver for wideband CDMA (W-CDMA). It can be shown from simulation and implementation results that the proposed architecture can effectively reduce the bit-error-probability (BER) in the presence of multiple access interference (MAI). For the purpose of programmability and modularity, the proposed system architecture adopts the parameter-download-type scheme. By modifying the core parameter, we can reconfigure the proposed system easily to meet the different specifications. To realize the system, we exploit some of the hardware characteristics offered by TI TMS320C6201 DSP processor, such as parallel ALUs, circular addressing mode, McBSP device, and so on, to develop a programmable receiving backend compatible with 3GPP.

# 第一章

## 緒論

為滿足人類對通訊的需求，通訊科技在近年來有長足的進步。隨著第三代無線通訊的蓬勃發展，各式各樣通訊系統與通訊平台亦不斷的出現，但另一方面，許多舊一代的系統也仍持續運作中。如 GSM、IS-95、PHS、GPRS 與 IMT-2000 等系統，雖然許多組織曾嘗試著使規格統一，但實際上許多差異仍出現在不同的系統中，新舊世代的交換，並非一蹴可及。不同系統的同時運行，代表需要同時使用不同的設備端，造成使用上諸多不便。如何使單一設備端具備多模與多功能，並能快速切換到不同系統，為軟體無線電系統的基本概念。本報告中，吾人將以軟體無線電的概念設計第三代無線電系統之接收器架構，並分析在即時(Real-Time)的要求下，系統以硬體實現之可能性。

軟體無線電系統採用分層設計的方式，接收機在接收到訊號後，會將之轉成全數位化訊號，再經由下層的處理單元處理。數位化處理

原件的特點在於可透過下載程式碼的方式改變其內部功能，吾人可利用下載程式碼的方式，輕易改變不同的系統架構。但處理原件的處理速度為系統時限上一個主要的瓶頸，尤其是負責基頻處理的中央處理器部分。隨著處理器速度的日新月異，近年來以商業應用為主的軟體無線電系統已陸續出現。應用上，DSP 處理器與 FPGA 幾乎已成為設計軟體無線電系統的不二選擇。此兩種處理器皆有可程式化的特點。程式設計上，DSP 偏向於開發微處理器程式，FPGA 則類似晶片設計。整合多樣化周邊硬體之 DSP 處理器，具備運行獨立系統(stand-alone system)之能力，便利之發展環境，易於實作複雜之演算法架構。衡量 DSP 處理器以上特點，吾人選擇其作為開發系統之硬體平台。第二章中將簡介軟體無線電的基本概念，並討論實作上的相關考量。關於軟體無線電與實作上的的相關討論，可參考[1]-[7]。

為達到更高速的通訊要求，DS-CDMA[9]，[10] 術普遍使用於第三代行動通訊系統中。使用者利用不同的展頻碼，於相同的頻帶同時進行通訊，雖可有效利用頻寬，但也同時存在有多重進接干擾(MAI)。再者，於行動無線通訊環境中，訊號在行動端和基地台間傳遞時會因建築物 and 地形影響而產生多重路徑衰落，導致通道瞬間狀態隨時間改變。此外，由於與距離相關的路徑損失(path loss)、屏蔽效應(shadowing)的影響，產生遠近效應問題(near-far effect)。由於這些影響與強多重

存取進接干擾，導致上鍊路系統容量下降，需使用不同的技術以抑制干擾：如快速傳輸功率控制(TPC)的採用，可將訊號干擾功率比盡可能保持在事先定義的目標值[11]，[12]，藉以減少多重存取進接干擾。另一方面，以匹配濾波器(matched filter, MF)對訊號解出不同路徑及時間延遲的訊號，再利用犁耙(RAKE)接收器將其結合，降低多重路徑干擾影響。進一步可結合智慧型陣列天線與干擾消除機以有效消除多重存取進接干擾。第三章中將介紹空-時犁耙接收機基本架構，吾人以寬頻分碼多重進接(W-CDMA)[13]，[14]，[15]之規格實作接收機基頻處理部分使用軟體無線電的概念，搭配兩種不同形式的智慧型陣列天線架構，第五章中將分析此架構之效能。

第四章中介紹數位信號處理器。數位信號處理器(DSP processor)具備有低耗電與高運算速度等特性，可廣泛應用於通訊、控制、影像處理、語音處理與消費性電子等領域。TMS320C6201 DSP 處理器為德州儀器公司[19]發展之最新一代通訊 DSP 處理器。其採用超長指令架構(Very long instruction word, VLIW)設計，直另執行動作精簡優異。多運算單元可並行運作，搭配高速的運作時脈(200Mhz)，理論上可達到 1600 MIPS 的處理速度[21]。此 DSP 高速化的特點，已可應用於實際系統中，多樣化硬體周邊與高度可程式化，更適合用於快速驗證系統效能。

第五章中吾人將簡介程式架構與流程，並簡單說明程式設計之流程與考量。在講求處理速度的情況下，DSP 程式設計必須由配合硬體特性與程式最佳化的觀點出發[23]，[24] TMS320C6201 採用定點數運算架構，為求充分發揮硬體效能，吾人使用定點數演算法撰寫程式，對於資料動態範圍，運算精確度必須有適當處理。此章節中比較系統在不同架構下的效能表現，並驗證實作結果。吾人在第六章中作一總結，並討論多種進階系統實作方法。

## 第二章

### 軟體無線電系統概述

軟體無線電系統具備有多功能與可程式化的特點，透過運作軟體的置換，能切換至各種不同規格之系統。通訊系統所需之可適性訊號處理能力，在此系統架構中更可輕易實現。本章將簡介軟體無線電的基本架構，並說明現階段軟體無線電系統之發展狀況。

#### 2.1 軟體無線電架構

軟體無線電的概念為”可滿足不同環境下多模式、多功能通訊要求，具備可適性訊號處理、原件可程式化能力之全數位化通訊設備”。在此概念下，利用軟體控制方式以改變硬體特性的通訊設備，皆可廣泛視為軟體無線電系統[16]。軟體無線電系統的發展方式類似於開發軟體，系統中各個硬體原件模組可視為功能不同的物件(object)，視需要呼叫執行。因此，吾人可透過下載程式碼之方式來置換物件，亦即改



變系統架構。一個軟體無線電系統的基本架構如圖 2.1 所示，系統由不同的硬體模組組成，包含：天線(antenna)單元、多頻帶射頻(RF)單元、中頻(IF)濾波器、類比-數位轉換器(ADC)、數位-類比轉換器(DAC)、中央處理器等(CPU)。採用上述架構實作出的系統具有以下特點：

- 1、可藉由改變任一模組的方式達成改變系統架構的目的。
- 2、系統本身隨時可透過自行調整達到最佳化的目的。
- 3、系統可適用於任何規格的通訊系統。

然而，軟體無線電系統同時存在著幾個主要的問題。第一，處理速度。軟體無線電系統深受受處理器速度影響。因 DSP 處理器與 FPGA 晶片皆具有可重複程式化的特色，近代軟體無線電系統也大多以此兩種硬體作為主要的中央處理器。但長久以來，使用高速化的 FPGA 晶片或使是用容易開發的 DSP 處理器。一直是個值得爭議的地方。使用 DSP 處理器的好處在於其易於使用之特性，但對於需要及時運算與平行處理的場合，如解調與調變、解展頻等部分，DSP 處理器的晶片速度便略顯不足。使用 FPGA 晶片開發雖然可以提高處理速度，但為處理複雜的計算，FPGA 需要相當大的電路面積，造成產品體積過大的問題。一般而言，DSP 處理器適合開發複雜之演算法架構，FPGA 則適用於需快速處理速度之單元，表 2.1 為 FPGA 與 DSP 之比較表[19]，

[20]。第二，耗電問題。對於使用電池的行動設備而言，耗電量為不可忽略的因素。越高速的類比-數位轉換器其耗電量越大，複雜的中央處理器更是消耗大量電源。第三，射頻模組不易設計。多頻帶的使用造成對寬頻射頻模組的需求，提高設計之複雜度。

## 2.2 智慧型天線系統

智慧型天線系統為一個具有可適性特色的接收架構。透過改變對不同入射方向訊號權值(weights)之方式以形成不同的波束，可增加系統處理空間訊號之能力，降低由不同方向入射訊號所造成之干擾，如圖 2.2 與圖 2.3。軟體無線電系統可藉由與智慧型天線系統的搭配，提高系統的效能與容量。可適性陣列天線即為一智慧型天線系統的應用實例。透過可適性調整天線權值的方式，動態改變波束，使得同頻道干擾(cochannel interference, CCI)降到最低，並提高訊號-干擾比(signal to interference ratio, SIR)，因此可使用於多路徑干擾的環境。在可適性陣列天線中，因計算權值需佔用部分處理器時間，實作上處理器速度的問題必須列入考慮。固定式波束陣列天線基本上不屬於智慧型天線系統，因其波束方向視硬體設計而定，無法於系統運作時適應性更動。使用扇型多工陣列天線會增加系統硬體成本，但因不會對系統處理速度帶來影響，且能提升一定之效能，普遍使用於傳統接收器中。

吾人可稍加修改固定波束式陣列天線架構，使系統具備波束選擇的能力，提高系統效能。關於上述兩種架構的詳細討論，可參考第三章與[8]之說明。配合軟體無線電系統可變換模組之特性，一個系統可以同時具備多種智慧型天線架構，軟體無線電系統再視所處通訊環境之差異，選擇最適合之智慧型天線架構與其搭配[18]，基本架構示意圖如圖 2.4 所示。

## 2.3 參數載入式軟體無線電系統

一般的軟體無線電系統，如圖 2.5 所示，當吾人啟動通訊系統時，必須將程式碼載入到系統中，並啟動初始程式，此類型之軟體無線電系統稱為 full-download-type 軟體無線電系統。此系統的缺點在於下載程式碼過於龐大，使用者需花費長時間下載程式碼，且元件使用率過低。一個替代的方案[17]，如圖 2.6 所示，整個系統由一個中央參數控制器統籌管理。欲改變系統規格時，只需下載對應系統規格參數。其特點在於系統程式碼體積精簡，下載程式碼之時間極短以及提高系統核心程式碼重複利用率。

## 2.4 結論

在這個章節中，吾人簡介軟體無線電系統之基本概念，並介紹一些提高效能的方法與實作觀念。如智慧型天線的使用與 parameter-download-type 的系統架構。以下章節，吾人將分別針對這些架構作進一步的說明，並以 DSP 晶片實作驗證系統效能。

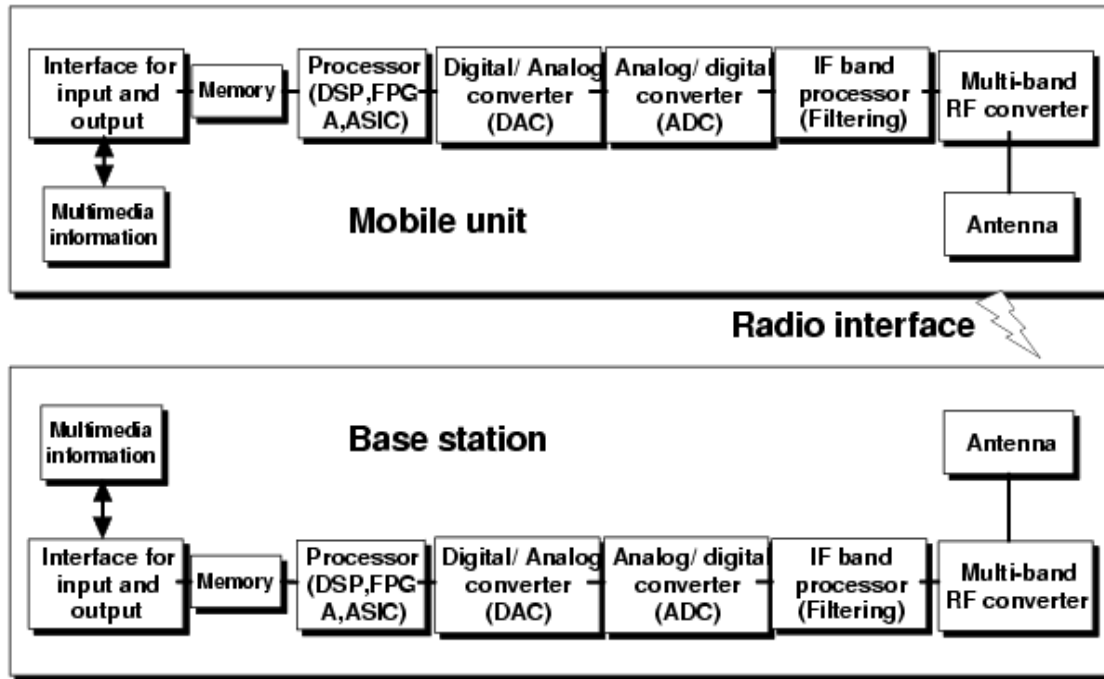


圖 2.1：軟體無線電系統架構

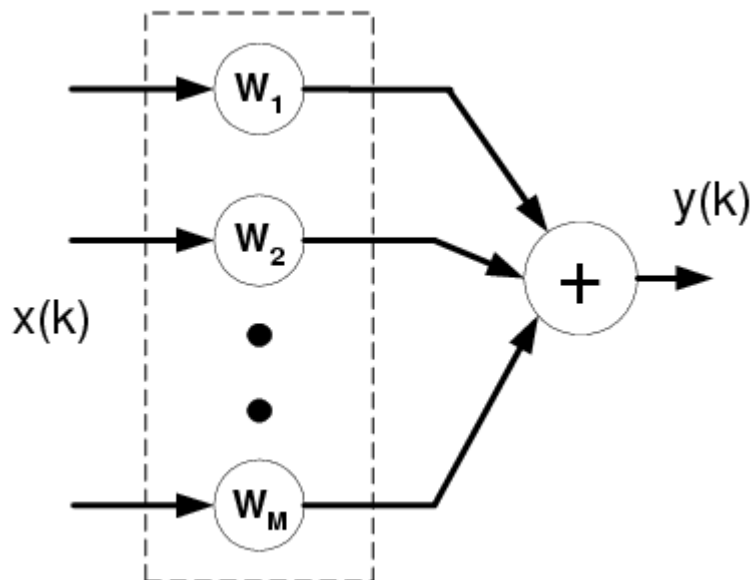


圖 2.2：空間分集結合器

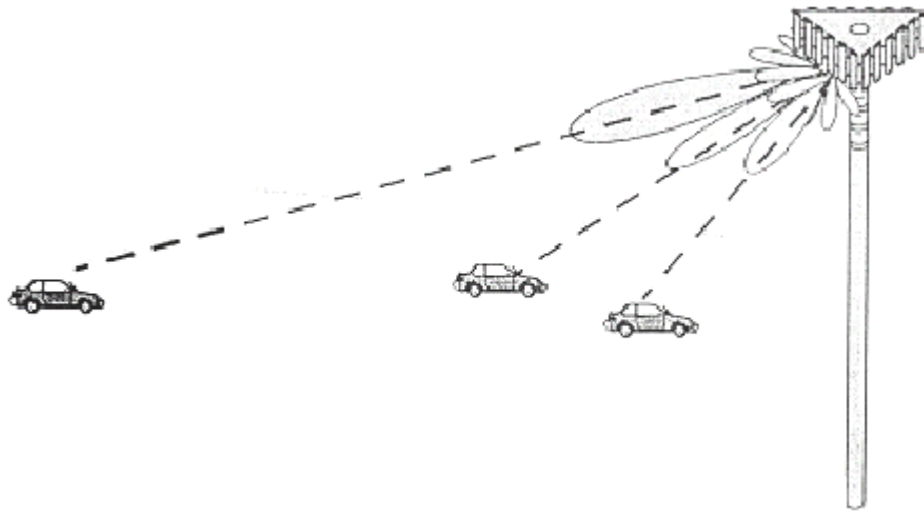


圖 2.3：波束干擾消除示意圖

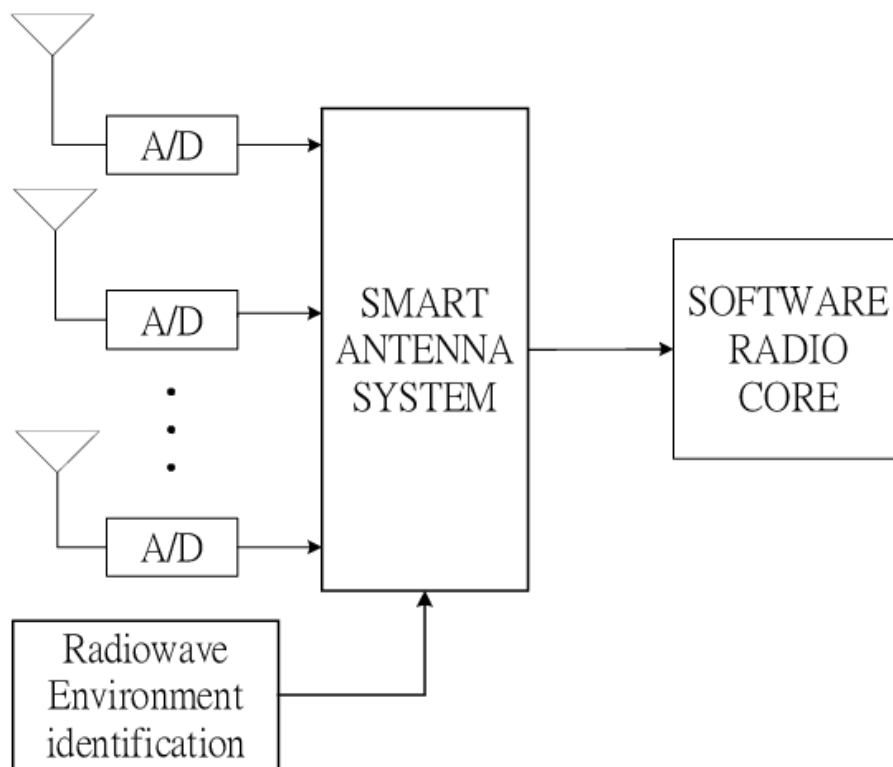


圖 2.4：使用智慧型天線之軟體無線電系統架構方塊圖

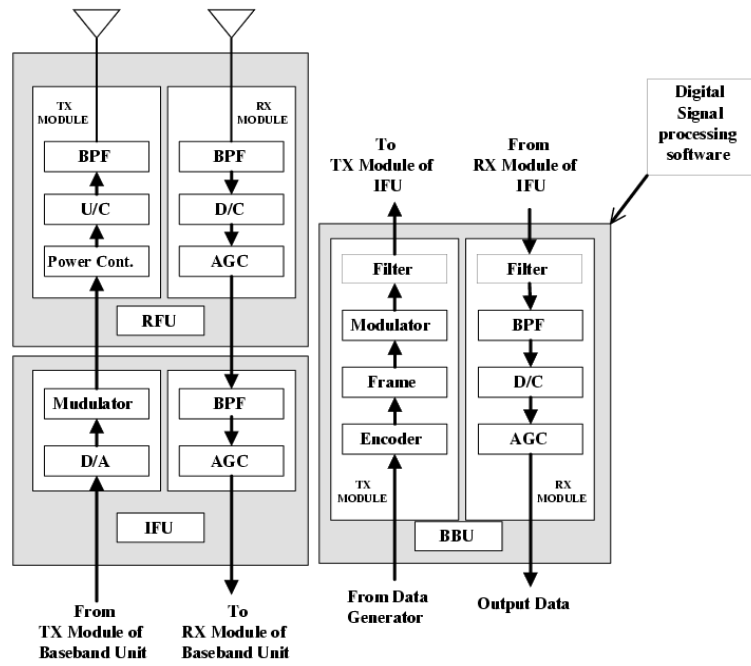


圖 2.5 : full-download-type 軟體無線電系統內部架構方塊圖

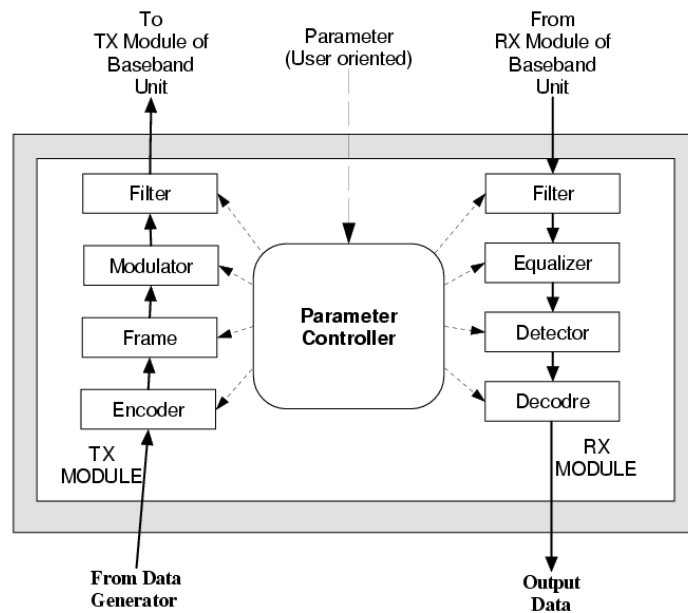


圖 2.6 : parameter-download-type 軟體無線電系統內部架構方塊圖

表 2.1：FPGA 與 DSP 比較表

	FPGA Chips	DSP Chips
程式語言	VHDL, Verilog	C/C++, Assembly
效能	高	依 DSP 時脈決定
適合設計元件	FIR filter, IIR filter, correlator, convolver, FFT	A signal processing program of sequential nature
耗電	低	高
平行處理能力	可平行處理	視作業系統設計而定



## 第三章

### 空-時犁耙接收器系統模型

在這一章節，將介紹智慧型空-時犁耙接收器系統，智慧型陣列天線系統包含犁耙接收器。智慧型天線系統使用兩種不同架構，分別為：適應性陣列天線以及固定式波束陣列天線。最後將討論其在 3GPP 的規格下的性能。

#### 3.1 專用實體通道的展頻與調變

本章節重點將放在 3GPP 規格資料格式與調變方法。在 3GPP 規格裡的專用實體通道是由專用實體資料通道(DPDCH)和專用實體控制通道(DPCCH)組成。專用實體控制通道負責傳送第一層的控制訊息，專用實體資料通道則用於傳送第二層的專用資料。專用實體控制通道內含尋引位元之已知符號(供上鏈路通道估計使用)、功率控制指令(TPC)以及傳送格式指引位元，其結構如圖 3.1。每一個時框(frame)長度為 10 ms，含有 15 個時槽(slot)。圖 3.1 之參數  $k$  是用來決定位於上鏈路

專用實體資料通道內每一個時槽所容納的位元數。展頻係數  $SF$  ( $= 256/2^q$ )，範圍由 256 到 4。由於專用實體控制通道位元數固定，即使用不同的展頻係數，大部分專用實體資料通道和專用實體控制通道的傳輸速率仍不盡相同。表 3.1 和表 3.2 為各通道的格式，表 3.3 和表 3.4 則為尋引位元模型。

圖 3.2 為上鏈路專用實體通道展頻與調變示意圖，資料的調變採用雙通道 QPSK，亦即將  $I$  通道和  $Q$  通道當成兩個獨立的 BPSK 通道。當專用實體資料通道只有一個時，專用資料實體通道和專用實體控制通道分別使用兩個不同的展頻碼，並置於不同的通道中。若同時存在數個專用實體資料通道，則需使用額外的展頻碼，並可置於任一通道。兩展碼資料結合後，再利用特定的攪亂碼將展頻訊號  $I + jQ$  攪亂。關於上鏈路專用實體通道展頻碼，有一定的限制存在：相同的展頻碼不能重複使用，即使其用於不同的通道中。限制的理由是如果  $I$  通道和  $Q$  通道的資料使用相同的展頻碼，其在專用實體控制通道被偵測並進行通道相位估計前將無法分離出來。其次，展頻碼如果出自於同組碼樹，解展頻時會有碼樹結構的優勢，可以避免 chip level buffering。行動端可直接提供資料速度資訊，或透過專用實體控制通道 TFCI 欄位提供更準確的訊息，此資訊可使基地台動態選擇適當的展頻係數以進行資料解展頻。以圖 3.3 為例，通道展頻碼記

為  $C_{SF, codenumber}$ ，其中  $SF_{d,n}$  表示為第  $n$  個專用實體資料通道之展頻係數。專用實體控制通道之通道碼則由碼號為 1 之碼開始，依照展頻係數  $SF_c$  展開。

$$\begin{bmatrix} C_{2,1} \\ C_{2,2} \end{bmatrix} = \begin{bmatrix} C_{1,1} & \overline{C_{1,1}} \\ C_{1,1} & \overline{C_{1,1}} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (3.1)$$

$$\begin{bmatrix} C_{4,1} \\ C_{4,2} \\ C_{4,3} \\ C_{4,4} \end{bmatrix} = \begin{bmatrix} C_{2,1} & \overline{C_{2,1}} \\ C_{2,1} & \overline{C_{2,1}} \\ C_{2,2} & \overline{C_{1,1}} \\ C_{1,1} & \overline{C_{2,2}} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \quad (3.2)$$

⋮

$$\begin{bmatrix} C_{2^{n+1},1} \\ C_{2^{n+1},2} \\ C_{2^{n+1},3} \\ \overline{C_{2^{n+1},4}} \\ \vdots \\ C_{2^{n+1},2^{n+1}-1} \\ C_{2^{n+1},2^{n+1}} \end{bmatrix} = \begin{bmatrix} C_{2^n,1} & \overline{C_{2^n,1}} \\ C_{2^n,1} & \overline{C_{2^n,1}} \\ C_{2^n,2} & \overline{C_{2^n,2}} \\ C_{2^n,2} & \overline{C_{2^n,2}} \\ \vdots & \vdots \\ C_{2^n,2^n} & \overline{C_{2^n,2^n}} \\ C_{2^n,2^n} & \overline{C_{2^n,2^n}} \end{bmatrix} \quad (3.3)$$

訊號傳輸時，利用攪亂碼(scramble code)可區隔不同來源之訊號，在上鏈路方向分有長碼和短碼兩種。長碼被截成 1 個時框 10 ms 的長度，在碼片率(chip rate)為 3.84 Mcps 的情況下，1 個時框會有 38400 個碼片長。短碼長度為 256 個 chip 長，用於架構複雜之系統以

降低設計困難度，而較為複雜的長碼則適用於架構簡單之接收器。本報告採用短碼以降低設計複雜度。短碼  $S_v(n), n=0,1,\dots,255$ ，選自於  $S_2$  延伸碼族，可由  $z_v(n)$  推得。長度 255 切片的四進位序列  $z_v(n), 0 \leq v \leq 16,777,216$ ，則是由一個四進位序列  $a_r(n)$  和兩個二進位序列  $b_s(n)$  和  $c_t(n)$  進行四進位加法求得，其關係式如下：

$$z_v(n) = a_r(n) + 2b_s(n) + 2c_t(n) \pmod{4}, \quad n=0,1,\dots,254 \quad (3.4)$$

使用者索引  $v$  決定組成序列的索引值  $r, s$  和  $t$ ：

$$\begin{aligned} v &= t \cdot 2^{16} + s \cdot 2^8 + r \\ r &= 0, 1, 2, \dots, 255 \\ s &= 0, 1, 2, \dots, 255 \\ t &= 0, 1, 2, \dots, 255 \end{aligned} \quad (3.5)$$

$a_r(n)$  由遞迴產生式  $G_0$  (多項式為  $g_0(x) = x^8 + x^5 + 3x^3 + x^2 + 2x + 1$  產生，

其式為：

$$\begin{aligned} a_r(n) &= 3 \cdot a_r(n-3) + 1 \cdot a_r(n-5) + \\ &\quad 3 \cdot a_r(n-6) + 2 \cdot a_r(n-7) + 3 \cdot a_r(n-8) \pmod{4} \end{aligned} \quad (3.6)$$

序列  $b_s(n)$  由遞迴產生式  $G_1$  (多項式為  $g_1(x) = x^8 + x^7 + x^5 + x + 1$

產生，其式為：

$$b_s(n) = b_s(n-1) + b_s(n-3) + b_s(n-7) + b_s(n-8) \pmod{2} \quad (3.7)$$

序列  $C_i(n)$  由遞迴產生式  $G_2$  (多項式為  $g_2(x) = x^8 + x^7 + x^5 + x^4 + 1$  產生，其式為：

$$c_i(n) = c_i(n) + c_i(n-3) + c_i(n-4) + c_i(n-8) \pmod{2} \quad (3.8)$$

圖 3.4 為短攪亂碼產生器示意圖。

## 3.2 空-時犁耙接收機

適應性天線陣列可以藉由將波束消除 (beam nulling) 方向指向干擾源以壓制多重進接干擾源。其方法為將收進各別天線之訊號分別乘上適當的權值，以提升使用者的訊號干擾功率比 (SIR)。為求得適當之權值，不同系統間會有不同的處理方式，稍後詳述之。

### 3.2.1 主要架構

圖 3.1 為上鍊路的時框格式， $N_p(N_{pilot})$  個已知的尋引符號和  $N_d(N_{data})$  資料符號各佔用一組通道平行傳輸。圖 3.2 則為上鍊路展頻/調變圖解，資料通道和尋引通道使用不同的 OVSF 碼，並分置於

$I$  通道和  $Q$  通道進行調變。調變的訊號在乘上與所在細胞(cell)相關的攪亂碼後傳送出去。假設有  $k$  個使用者同時傳送，訊號經過多路徑傳輸通道被基地台以陣列天線接收後。第  $k$  個使用者傳送訊號  $s_k(t)$ ， $k=1,2,\dots,K$ ，表示成複數的型態：

$$s_k(t) = d_k(t)C_k(t) + j^* P_k(t) \quad (3.9)$$

其中  $d_k(t)$ 、 $C_k(t)$  和  $P_k(t)$  分別為符號波形、展頻資料波形以及尋引符號展頻波形。進一步可表示如下：

$$\begin{cases} d_k(t) = \sum_{m=-\infty}^{\infty} b_k(m)u(t/T - m) \\ C_k(t) = \sum_{n=-\infty}^{\infty} c_k(n)u(t/T_c - n) \\ P_k(t) = \sum_{n=-\infty}^{\infty} cp_k(n)u(t/T_c - n) \end{cases} \quad (3.10)$$

其中  $b_k(m)$  為資料符號， $c_k(n)$  為資料展頻碼，而  $cp_k(n)$  為尋引展頻碼。資料展頻碼和尋引展頻碼長度並不需要相同。在 3.10 式裡， $T$  為資料時間間隔， $T_c$  為展頻碼  $c_k(n)$  碼片時間間隔，當  $0 \leq t \leq 1$  時  $u(t)$  為 1，否則為 0， $P_g = T/T_c$  則為處理增益(processing gain)。若第  $k$  個使用者有  $L_k$  個可分解的路徑，其獨立複通道增益表示為  $a_{kl}(t)$ ，時間延遲為  $\tau_{kl}$ ， $l=1,2,\dots,L_k$ 。

圖 3.5 與 3.6 為處理第  $k$  個使用者在第  $l$  個路徑訊號之空-時犁耙接收機之方塊圖。接收到的複合訊號表示為：

$$\mathbf{r}(t) = \sum_{k=1}^K \sum_{l=1}^{L_k} \mathbf{a}(\theta_{kl}) \alpha_{kl}(t) [d_k(t)C(t - \tau_{kl}) + j^* P_k(t - \tau_{kl})] + \mathbf{n}(t) \quad (3.11)$$

其中  $\mathbf{a}(\theta_{kl})$  為第  $k$  個使用者在第  $l$  個路徑的空間特徵向量， $\mathbf{n}(t)$  則為背景(複數高斯)雜訊。需注意到在本報告中假定所有  $L_k$  個路徑訊號抵達第  $J$  根天線的延遲時間  $\tau_{kl}$  均為相同，通道增益則存在有些許不同。假定成立的原因乃因在載波頻率高達 2 GHz 的情況下，天線間的距離遠小於電磁波在一個碼片時間  $T_c$  所傳播的距離。圖 3.5 與圖 3.6 所示，空-時犁耙接收器是由匹配濾波器(matched filter, MF)、波束形成器和同調(coherent)時間犁耙接收機所組成。其中匹配濾波器是由一組相關器(correlator)合成，每一個相關器則是由乘法器和積傾濾波器(integrate-and-dump filter)組成。在乘法器乘法器運作時，接收訊號將會延遲一段欲解析路徑之預估路徑延遲時間，收到訊號再乘上一組由本地端產生之對應展頻碼，其後積傾器將於每符號間隔送出一積分值(即解展頻訊號)。因考慮  $L_k$  個路徑，每一根天線接收的訊號在經由匹配濾波器解展頻後，將分析出  $L_k$  個取樣。在處理尋引符號時，吾人假設其值皆為 1 (在 3GPP 的規格裡定義有尋引符號的格

式，本報告之模擬均遵循此規範)。則在時間  $t = mT + \tau_{kl}$ ，匹配濾波器會對於第  $J$  根天線第  $l$  根路徑輸出取樣值  $y_{kl(j)}$  (資料符號)和  $x_{kl(j)}$  (尋引符號)，其表示式為：

$$\begin{cases} y_{kl(j)}(m) = \frac{1}{T} \int_{mT+\tau_{kl}}^{(m+1)T+\tau_{kl}} r_{(j)}(t) C_k(t - \tau_{kl}) dt \\ x_{kl(j)}(m) = \frac{1}{T} \int_{mT+\tau_{kl}}^{(m+1)T+\tau_{kl}} r_{(j)}(t) P_k(t - \tau_{kl}) dt \end{cases} \quad (3.12)$$

其中  $r_{(j)}(t)$  為第  $j$  根天線所接收的訊號。將  $J$  根天線解展頻的訊號  $y_{kl(j)}$  和  $x_{kl(j)}$  取樣加權並相加後形成一接收波束場形，可提昇該根路徑的訊號雜訊比。相加合成結果值  $z_{kl}$  和  $p_{kl}$  (分別為第  $l$  根路徑波束形成器資料和尋引符號的輸出)，其式為：

$$\begin{cases} z_{kl}(m) = \sum_{j=1}^J y_{kl(j)}(m) w_{kl(j)}^*(m) \\ p_{kl}(m) = \sum_{j=1}^J x_{kl(j)}(m) w_{kl(j)}^*(m) \end{cases} \quad (3.13)$$

其中  $w_{kl(j)}$  為第  $j$  根天線的權值。為了在時間  $t = mT + \tau_{kl}$  時能同調結合  $L_k$  個波束使形成器輸出  $z_{kl}(m)$ ，需利用已知尋引通道符號以估計出複和之通道增益值。吾人所使用之估計方法為取樣平均估計法則，其式為：



$$s_{kl} = \frac{1}{N_p} \sum_{m=1}^{N_p} p_{kl}(m) \quad (3.14)$$

使用通道估計值將第  $j$  個波束形成器輸出  $z_{kl}(m)$  的隨機相位補除後，吾人可使用犁耙接收機將訊號同調結合，接收機輸出式如下：

$$z_k(m) = \sum_{l=1}^{L_k} z_{kl}(m) s_{kl}^* \quad (3.15)$$

### 3.3 權值估計

波束形成器權值之估計方式，在不同的架構下會有不同的考量，使用之演算法也不盡相同。以下吾人將分別說明使用於本報告中之兩種不同權值決定方式。

#### 3.3.1 適應性權值調整演算法(Weight adaptation algorithm)

可適性天線陣列架構使用可適性方式動態調整天線陣列之權值。權值參數  $w_{kl(j)}$  乃利用最小均方(least-mean-square algorithm, LMS)演算法更新，適應性權值調整演算法的目的在盡可能將波束形成器輸出的均方差(mean square error, MSE)縮小，運算時需有相對應之參考訊號存在，以產生第  $l$  個波束產生器的錯誤訊號(error signal)。此吾人

可以尋引符號作為參考訊號，經由估測值將暫時的參考資料決定出來：

$$\tilde{b}_k = \text{sign}\{z_k(m)\} \quad (3.16)$$

由上式，參考訊號：

$$\tilde{z}_{kl}(m) = \tilde{b}_k(m) * s_{kl} \quad (3.17)$$

錯誤訊號：

$$e_{kl}(m) = \hat{z}_{kl}(m) - z_{kl}(m) \quad (3.18)$$

最後，權值向量  $w_{kl}(m+1)$  使用正規化最小均方(NLMS)演算法求得為：

$$w_{kl(j)}(m+1) = w_{kl(j)}(m) + \mu \frac{y_{kl(j)}(m)}{\sum_{j=1}^J \|y_{kl(j)}(m)\|^2} \cdot e_{kl}^*(m) \quad (3.19)$$

### 3.3.2 固定式波束選擇器(Fixed-Beam selector algorithm)

使用固定式波束選擇器架構時，必須從對應至不同方向的候選權值中，

挑選出最適當的權值。選擇器以尋引通道為主要參考訊號，並利用接收能量大小決定適當的權值。所有權值的集合為：

$$\mathbf{W}_{kl} = \left[ \mathbf{w}_{kl}^{(1)} \quad \mathbf{w}_{kl}^{(2)} \quad \cdots \quad \mathbf{w}_{kl}^{(J)} \right] \quad (3.20)$$

其中

$$\mathbf{w}_{kl}^{(j)} = \left[ \omega_{kl(1)}^{(j)} \quad \omega_{kl(2)}^{(j)} \quad \cdots \quad \omega_{kl(J)}^{(j)} \right]^T \quad (3.21)$$

當訊號經過所有波束產生器後的輸出值為：

$$\mathbf{z}_{kl}(m) = \mathbf{y}_{kl}^T(m) \mathbf{W}_{kl} = \left[ z_{kl}^{(1)}(m) \quad z_{kl}^{(2)}(m) \quad \cdots \quad z_{kl}^{(J)}(m) \right] \quad (3.22)$$

其能量為：

$$\mathbf{E}_{kl}(m) = \left[ z_{kl}^{(1)}(m)^2 \quad z_{kl}^{(2)}(m)^2 \quad \cdots \quad z_{kl}^{(J)}(m)^2 \right] \quad (3.23)$$

波束選擇器再依能量值，從所有候選權值中選擇出最適合之第  $j$  組權值  $\mathbf{w}_{kl}^{(j)}$ ，此組權值則用於與下個資料時槽結合，估出所傳送訊息。

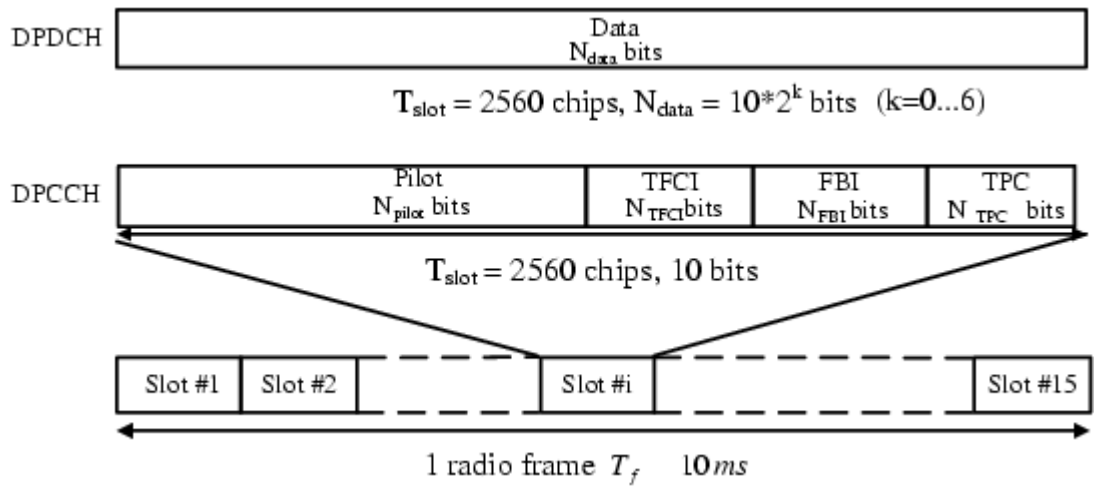


圖 3.1：上鏈路專用實體通道(DPDCH)規格

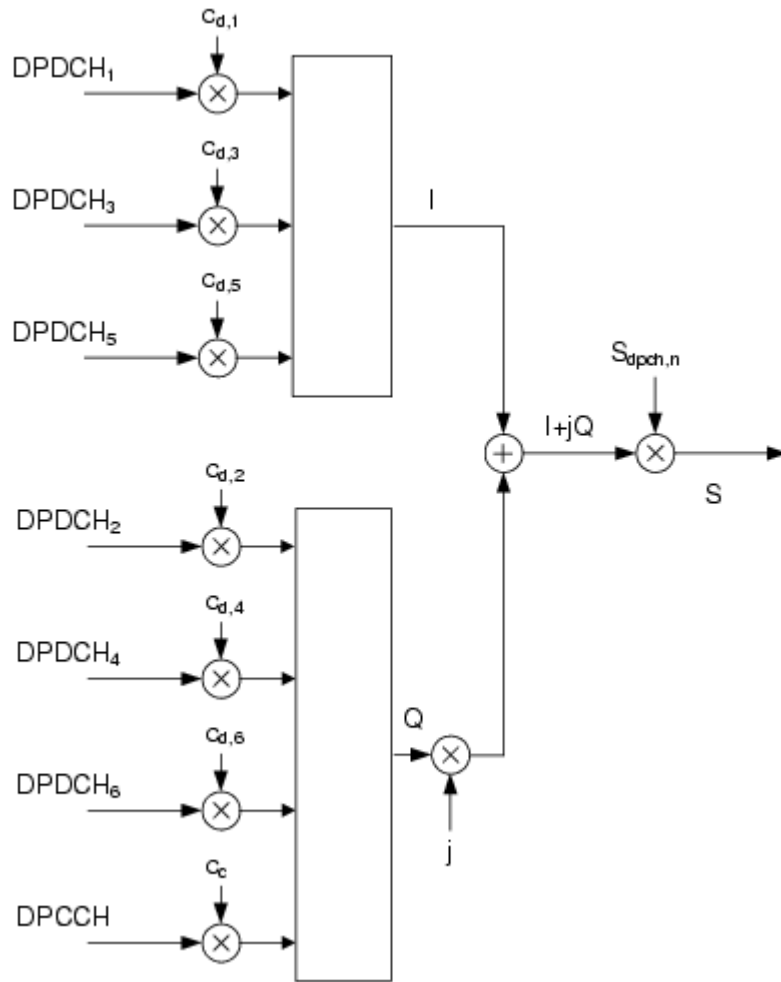


圖 3.2：上鏈路專用實體通道展頻與調變

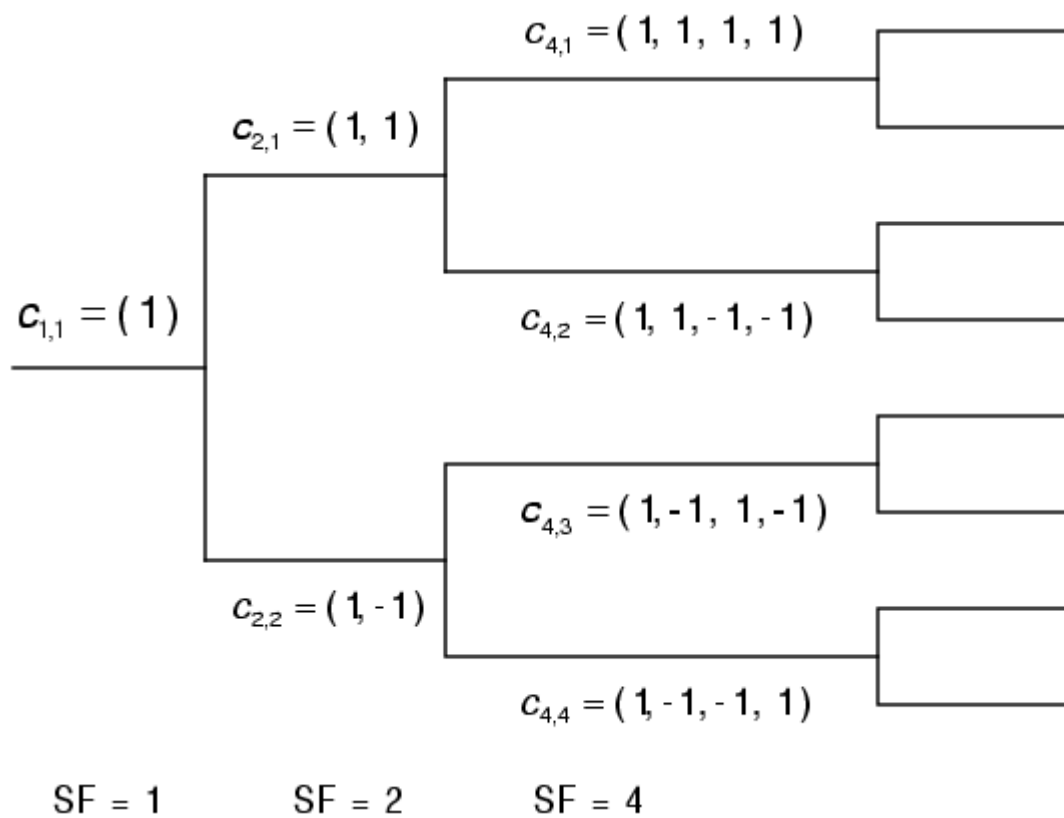


圖 3.3：正交可變展頻係數碼碼樹示意圖

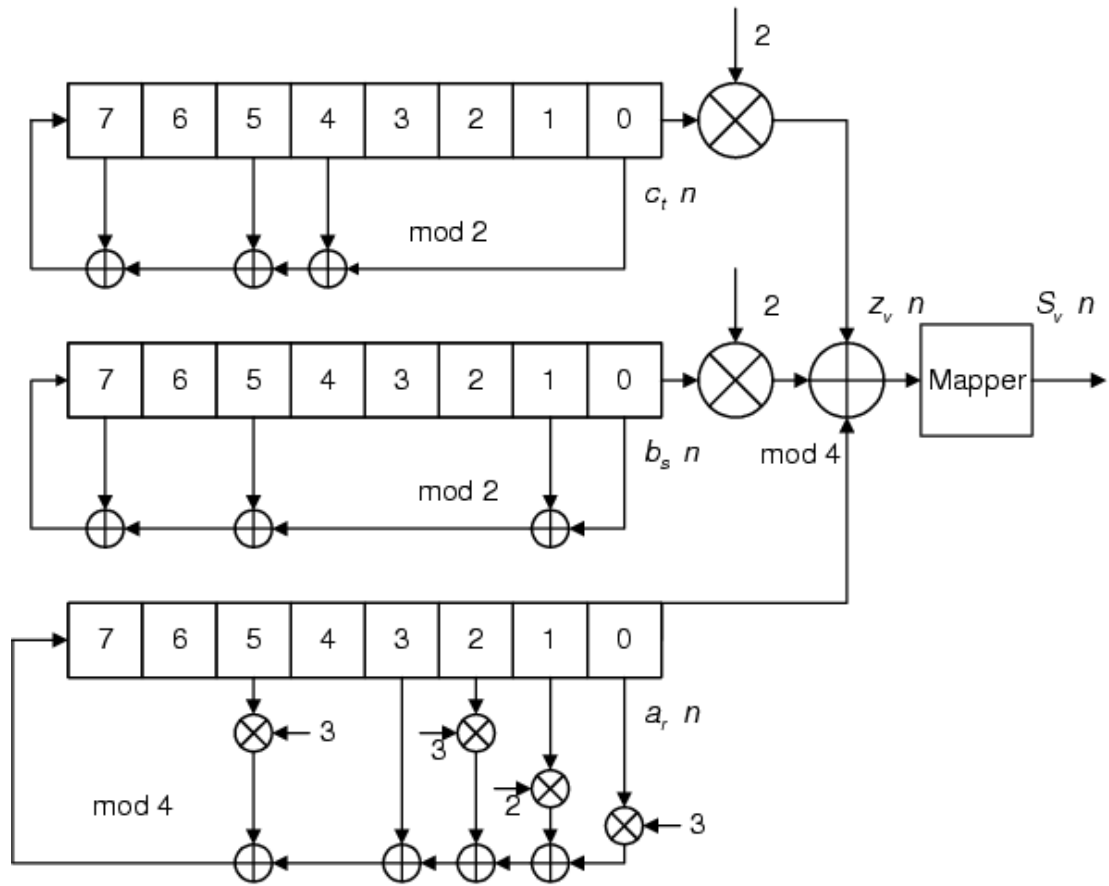


圖 3.4：短攪亂碼產生器示意圖

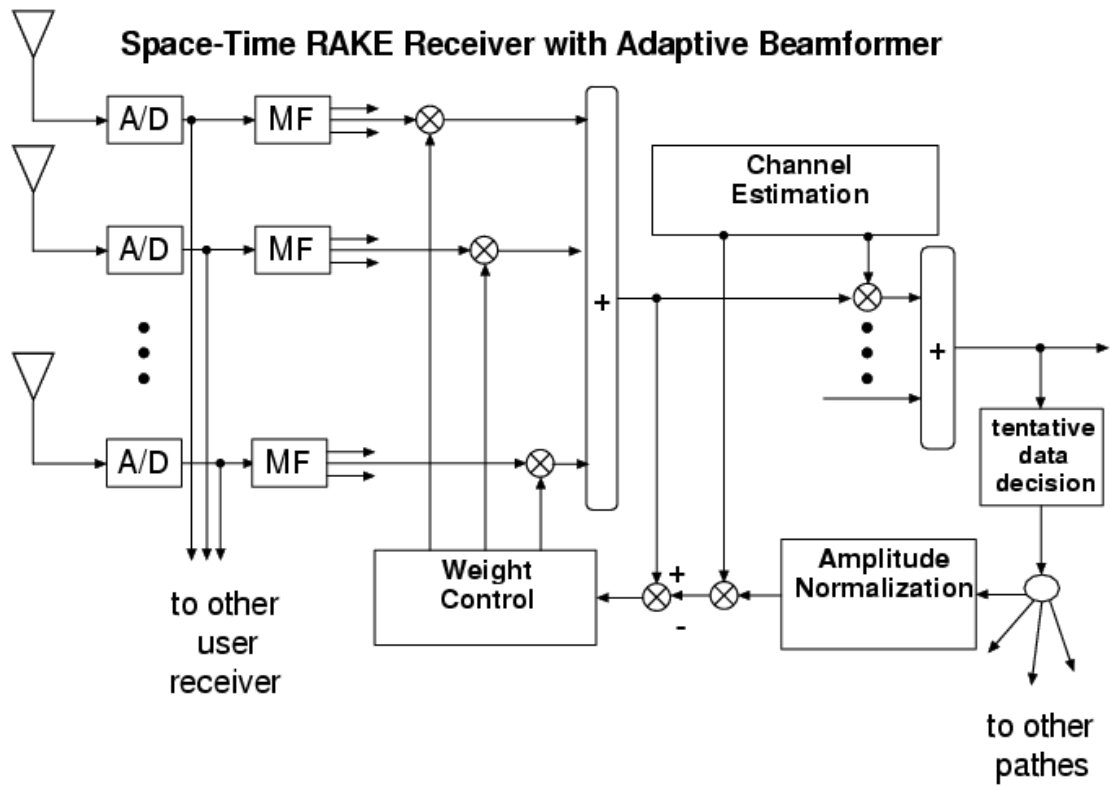


圖 3.5：空-時犁耙接收器架構圖(可適性陣列天線)



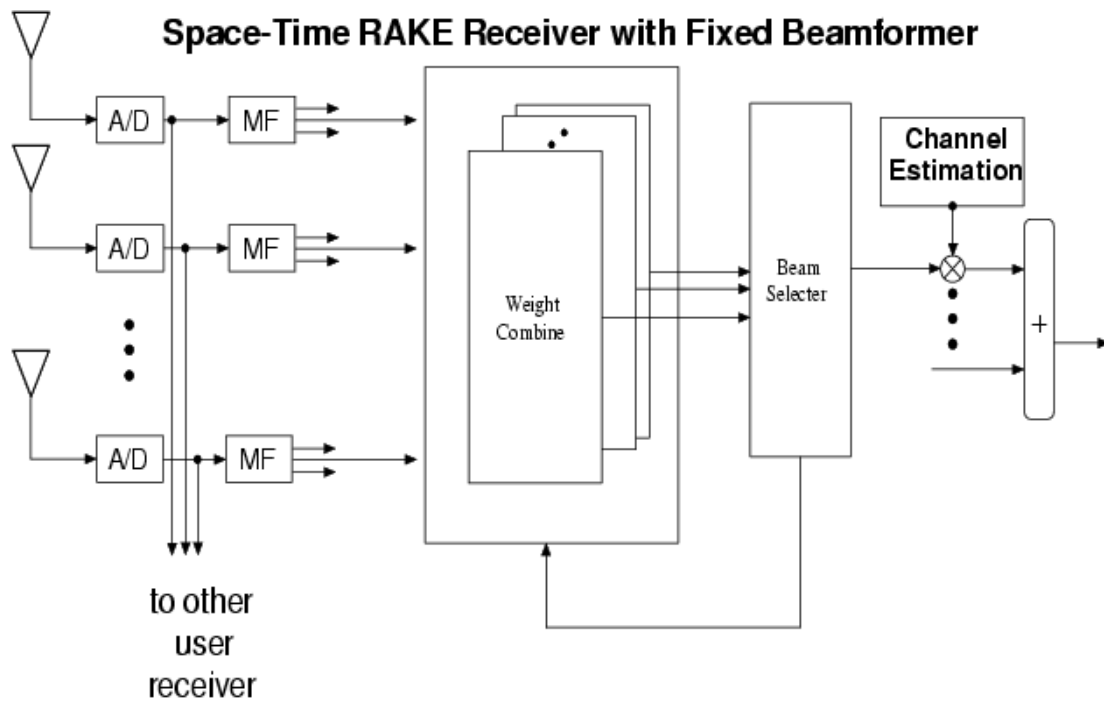


圖 3.6：空-時犁耙接收器架構圖(固定式陣列天線)

表 3.1：上鏈路專用實體通道(DPDCH)規格

時槽 格式 #i	通道位 元速率 (kbps)	通道符 號速率 (ksps)	展頻 係數	位元 /時框	位元 /時槽	$N_{data}$
0	15	15	256	150	10	10
1	30	30	128	300	20	20
2	60	60	64	600	40	40
3	120	120	32	1200	80	80
4	240	240	16	2400	160	160
5	480	480	8	4800	320	320
6	960	960	4	9600	640	640

表 3.2：上鏈路專用實體通道(DPDCH)規格

時槽 類型 #i	通道 位元 速率 (kbps)	通道 符號 速率 (ksps)	展頻 係數	位元 /時框	位元 /時槽	$N_{pilot}$	$N_{TPC}$	$N_{TFCI}$	$N_{FBI}$	時槽 傳送 數 /時框
0	15	15	256	150	10	6	2	2	0	15
0A	15	15	256	150	10	5	2	3	0	10-14
0B	15	15	256	150	10	4	2	4	0	8-9
1	15	15	256	150	10	8	2	0	0	8-15
2	15	15	256	150	10	5	2	2	1	15
2A	15	15	256	150	10	4	2	3	1	10-14
2B	15	15	256	150	10	3	2	4	1	8-9
3	15	15	256	150	10	7	2	0	1	8-15
4	15	15	256	150	10	6	2	0	2	8-15
5	15	15	256	150	10	5	1	2	2	15
5A	15	15	256	150	10	4	1	3	2	10-14
5B	15	15	256	150	10	3	1	4	2	8-9

表 3.3：上鏈路專用實體控制通道(DPCCH)尋引位元模型

	$N_{pilot} = 3$			$N_{pilot} = 4$				$N_{pilot} = 5$				$N_{pilot} = 6$						
位元 #	0	1	2	0	1	2	3	0	1	2	3	4	0	1	2	3	4	5
時槽 #0	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	0
1	0	0	1	1	0	0	1	0	0	1	1	0	1	0	0	1	1	0
2	0	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1
3	0	0	1	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0
4	1	0	1	1	1	0	1	1	0	1	0	1	1	1	0	1	0	1
5	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	0
6	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1	0	0
7	1	0	1	1	1	0	1	1	0	1	0	0	1	1	0	1	0	0
8	0	1	1	1	0	1	1	0	1	1	1	0	1	0	1	1	1	0
9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
10	0	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1
11	1	0	1	1	1	0	1	1	0	1	1	1	1	1	0	1	1	1
12	1	0	1	1	1	0	1	1	0	1	0	0	1	1	0	1	0	0
13	0	0	1	1	0	0	1	0	0	1	1	1	1	0	0	1	1	1
14	0	0	1	1	0	0	1	0	0	1	1	1	1	0	0	1	1	1

表 3.4：上鏈路專用實體控制通道(DPCCH)尋引位元模型 (續)

	$N_{pilot} = 7$							$N_{pilot} = 8$							
位元 #	0	1	2	3	4	5	6	0	1	2	3	4	5	6	7
時槽 #0	1	1	1	1	1	0	1	1	1	1	1	1	1	1	0
1	1	0	0	1	1	0	1	1	0	1	0	1	1	1	0
2	1	0	1	1	0	1	1	1	0	1	1	1	0	1	1
3	1	0	0	1	0	0	1	1	0	1	0	1	0	1	0
4	1	1	0	1	0	1	1	1	1	1	0	1	0	1	1
5	1	1	1	1	1	0	1	1	1	1	1	1	1	1	0
6	1	1	1	1	0	0	1	1	1	1	1	1	0	1	0
7	1	1	0	1	0	0	1	1	1	1	0	1	0	1	0
8	1	0	1	1	1	0	1	1	0	1	1	1	1	1	0
9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
10	1	0	1	1	0	1	1	1	0	1	1	1	0	1	1
11	1	1	0	1	1	1	1	1	1	1	0	1	1	1	1
12	1	1	0	1	0	0	1	1	1	1	0	1	0	1	0
13	1	0	0	1	1	1	1	1	0	1	0	1	1	1	1
14	1	0	0	1	1	1	1	1	0	1	0	1	1	1	0

表 3.5：短攪亂碼產生對應表

$z_v(n)$	$S_v(n)$
0	$+1 + j$
1	$-1 + j$
2	$-1 - j$
3	$+1 - j$

## 第四章

### TI TMS320C6201 DSP 通訊晶片簡介

TMS320C6201 為德州儀器公司發展之最新一代通訊 DSP 處理器，採用先進的超長指令架構(VLIW)設計，具 8 個可並行運作之運算單元與多樣化的硬體周邊，為一功能強大之 DSP 處理器。其架構[22]可參考圖 4.1。為使用此 DSP 處理器做為開發系統之主要硬體平台，吾人有必要瞭解其內部架構與周邊原件，以求充分發揮硬體特性。

#### 4.1 CPU 結構

TMS320C6201 DSP 晶片(以下簡稱 DSP)包含運算單元、控制暫存器與相關控制電路。

##### 4.1.1 運算單元

DSP 共有兩組運算單元組(1 組與 2 組)，各含四個運算單元，分別稱

為.L、.S、.M、.D 運算單元，功能如下：

.L 運算單元 - 邏輯、計數器。

.S 運算單元 - 位元移位(bit shift)、程式碼跳躍(branch)

.D 運算單元 - 線性(linear)、環狀(circular)定址器。

.M 運算單元 - 乘法器。

#### 4.1.2 CPU 內部暫存器

內部暫存器一功能可大致區分為兩大類：

##### 一般暫存器群組

DSP 中共有兩組一般暫存器群組 (A 群與 B 群)，各有 16 個 32 位元之暫存器。基本上，A 群暫存器提供運算單元組 1 組使用，B 群暫存器則提供運算單元組 2 組使用。但 DSP 亦提供特殊的存取路徑，如圖 42，使得運算單元組 1 可以使用 B 群暫存器，反之亦然。暫存器的長度為 32 個位元，透過兩個暫存器的結合，可延伸至 40 個位元。結合方式可參考表 4.1 及圖 4.3。

##### 控制暫存器群組

DSP 中總共有十個控制暫存器，分別用以儲存或設定 DSP 內部的狀態。此十個控制暫存器皆需要使用特殊的指令加以控制。以下簡要說明各控制暫存器的功能：



### 1. AMR (定址模式控制暫存器)

可控制暫存器儲存間接位址的定址方式，預設為線性定址模式 (linear mode)，亦可設定成環形定址模式(circular mode)。

### 2. CSR (控制狀態暫存器)

控制狀態暫存器為儲存控制及狀態位元之暫存器。可以顯示諸如是否運算溢位，DSP 及 DMA 之 word 或 half-word 之位元組儲存方式為 little endian 或 big endian。

### 3. IER、IFR、ISR、ICR (中斷控制及狀態暫存器)

此群暫存器組為儲存及設定中斷(INT1-INT15，NMI)是否致能，可分別致能或清除各個中斷。

### 4. ISTP (中斷向量位置暫存器)

儲存中斷向量服務程式(interrupt service routine)位址與中斷的優先順序。

### 5. IRP (中斷回復位置暫存器)

儲存一般中斷服務程式的回復位址。

### 6. NRP (NMI 中斷向量位置暫存器)

儲存 NMI 中斷服務程式的回復位址。

### 7. PCE1 (指令執行位址)

儲存執行狀態之指令位址。

## 4.2 記憶體架構

TMS320C6201 記憶體分配主要可分為程式記憶體(program memory)、資料記憶體(data memory)、內部周邊原件空間記憶體(internal-peripheral space memory)以及透過記憶體控制介面(EMIF)控制的外部記憶體四種。使用 32-bit 定址方式，總共可以存取至 4 Gbytes 的記憶體空間。

### 4.2.1 記憶體映射模式(memory map)

TMS320C6201 使用記憶體映射的控制方式，將所有的記憶體及暫存器位址映射至一個單一的記憶體空間，透過此單一定址空間，即可直接對所有記憶體與周邊控制暫存器作存取，TMS320C201 提供了兩種型態的映射方式(map 0 與 map 1)，如表 4.2。相異之處在於其對記憶體區塊的排列方式。資料放置的方式亦有兩種不同的格式，可以選擇 word 或 half-word 中低位元組的置放位置。將低位元組置放於低位址的作法稱為 little-endian，反之將低位元組置放於高位址之作法則稱為 big-endian，如圖 4.4。

## 4.2.2 內部程式記憶體

TMS320C6201 總共有 64K 之內部程式記憶體，可等於 2K 個 256-bit FP(Fetch Packet)或等於 16K 個 32-bit 指令。內部記憶體共有四種工作模式。大致上分成一般程式記憶體與快取記憶體兩種類型。

## 4.2.3 內部資料記憶體

TMS320C6201 總共有 64K 之內部程式記憶體，內部資料記憶體分成兩區塊，每塊 32Kbytes，分成 4 組，每組含 4K 個 16-bit half-word。記憶體儲存方式需對齊 2-byte 邊界或 4-byte 邊界限制。DSP 同時可對不同塊之不圖組記憶體作讀取動作。存取同組之記憶體則會產生 wait state。DSP 與 DMA 可同時對資料記憶體作存取動作，但如果欲對同組記憶體存取，則由 DMA 通道之 PRI 位元決定其優先次序。

## 4.2.4 內部周邊原件空間

內部周邊原件空間包含了外界記憶體控制介面(External Memory Interface, EMIF)及記憶體直接存取控制器(Direct Memory Access, DMA)的控制暫存器組。

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

EMIF 之主要功能為提供各種控制及時序介面信號，以連接各類型記憶體原件，擴充 DSP 可使用之記憶體。EMIF 可使用之記憶體原件包括同步原件如：同步靜態記憶體(SBSRAM)、同步動態記憶體(SDRAM)、以及非同步原件如：非同步靜態記憶體(ASRAM)、ROM、FIFO 等。EMIF 對外控制外接記憶體的使用，對內則負責對提出記憶體使用要求的原件提供必要的服務，如圖 4.5。

## 4.3 內部周邊原件

TMS320C6201 提供的多樣周邊原件，使其本身除了訊號處理功能外，透過配邊原件的整合，能發展出更多功能更為強大的應用。以下吾人將簡介幾個重要的週邊原件與其運作控制方式。

### 4.3.1 主電腦介面(Host Interface Ports, HPIs)

HPI 為一個 16 位元的並聯輸出入埠，提供主電腦(HOST 端)與 DSP 進行資料交換的介面，其架構圖如圖 4.6。透過此介面，主電腦可以由 DMA 的輔助通道經由 HPI 對 DSP 內部記憶體空間存取資料。HPI

有三個暫存器，分別為：

HPI 控制暫存器(HPI control register，HPIC)

HPI 位址暫存器(HPI address register，HPIA)

HPI 資料暫存器(HPI address register，HPID)

HPIC 可同時被主電腦與 DSP 存取，為兩者溝通訊號的主要暫存器。

其中的 DSPINT 位址用來讓主電腦對 DSP 產生中斷信號，HINT 則可令 DSP 對主電腦產生中斷訊號。主電腦對 DSP 存取時，需先設定 HPIC，將位址寫入 HPIA，並透過 HPID 讀取資料。

#### **4.3.2 直接記憶體存取(Direct Memory Access，DMA)**

TMS320C6201 DMA 控制器共有四個獨立的通道與一個連接 HPI 的輔助通道，可以不透過 DSP 的控制，直接對所有的映射記憶體位址 DMA 存取。其控制暫存器主要可分成兩大類：一為 DMA 通用控制暫存器，另一類則為個別通道控制暫存器。個別通道暫存器主要決定 DMA 每讀取一筆資料時的資料大小，來源位址與目的地位址，讀寫同步事件模式，DMA 啟動與停止、中斷產生。通用暫存器則可設定來源位址與目的地位址的資料移動方向，使用於 DSP 自動載入之情況時，可將每一傳送資料區塊所含傳送資料框數目及資料框資料筆數再重新載入某一通道之傳送計數暫存器中。一般常見的 DMA 動作方

式為區塊資料搬移，設定流程如下

- 1、設定搬移資料區塊大小。
- 2、目的地位址與來源位址。目的地位址與來源位址的增減方式，即下一筆資料的排列位址與前一筆資料位址的相關位址。
- 3、設定是否產生資料框同步訊號，設定讀寫同步訊號來源，同步訊號來源如表 \Z\ref{tbl:dma\_sync}\Z 所示，可使 DMA 資料讀寫配合周邊元件的動作。
- 4、設定是否產生中斷訊號，及中斷訊號產生之狀況。
- 5、啟動 DMA

### 4.3.3 中斷選擇器與中斷源(Interrupt Selector and External Interrupts)

TMS320C6201 的 16 個中斷服務來源除了重置與 NMI 對應至預設的重置與 NMI 中斷服務常式外，其餘的 INT4-INT15 中斷皆可透過中斷選擇器，自由選擇外接之中斷源。TMS320C6201 的外部中斷源共有 16 個，當系統重置時，系統會依中斷選擇器之預設值設定中斷源與內部中斷之對應關係，如表 4.4 所示。

### 4.3.3 高速串聯埠介面(Multichannel Buffered Serial Ports , McBSPs)

TMS320C6201 共有兩個高速串聯埠介面，可分別作高速的全雙工高速傳輸，其架構如圖 4.7 所示。TMS320C6201 之高速串聯埠提供多種不同傳輸方式，可直接連接各種通信標準介面如：T1/E1、MVIC、ST-BUS、IOM-2、AC97、IIS、SPI 等。並可支援最高達 128 通道之多通道傳輸模式。其內部提供之取樣信號產生器，可程式化產生本身所需的時脈信號與同步信號。多個時脈與信號框同步信號接腳，則可使其使用或輸出供外部設備使用之時脈與信號框同步信號。

TMS320C6201 高速串聯埠一般的工作方式為:

#### 接收模式

在接收端接收到設定的信號框同步信號後，啟動接收模式並將資料由 DR 腳位讀入，先送入 RSR，次移入 RBR，然後送入 DRR。此時高速串聯埠會對 DSP 設定接收信號(RRDY)與對 DMA 發出讀入事件信號(REVT)，分別通知 DSP 或 DMA 將資料讀入。

#### 傳送模式

在傳送端接收到設定的信號框同步信號後，此時高速串聯埠會對 DSP 設定傳送信號(XRDY)與對 DMA 發出寫出事件信號(XEVT)，分別通

知 DSP 或 DMA 將資料寫入 DXR，資料先移入 XSR，後由 DX 腳位送出。詳細時序圖可參考圖 4.8。



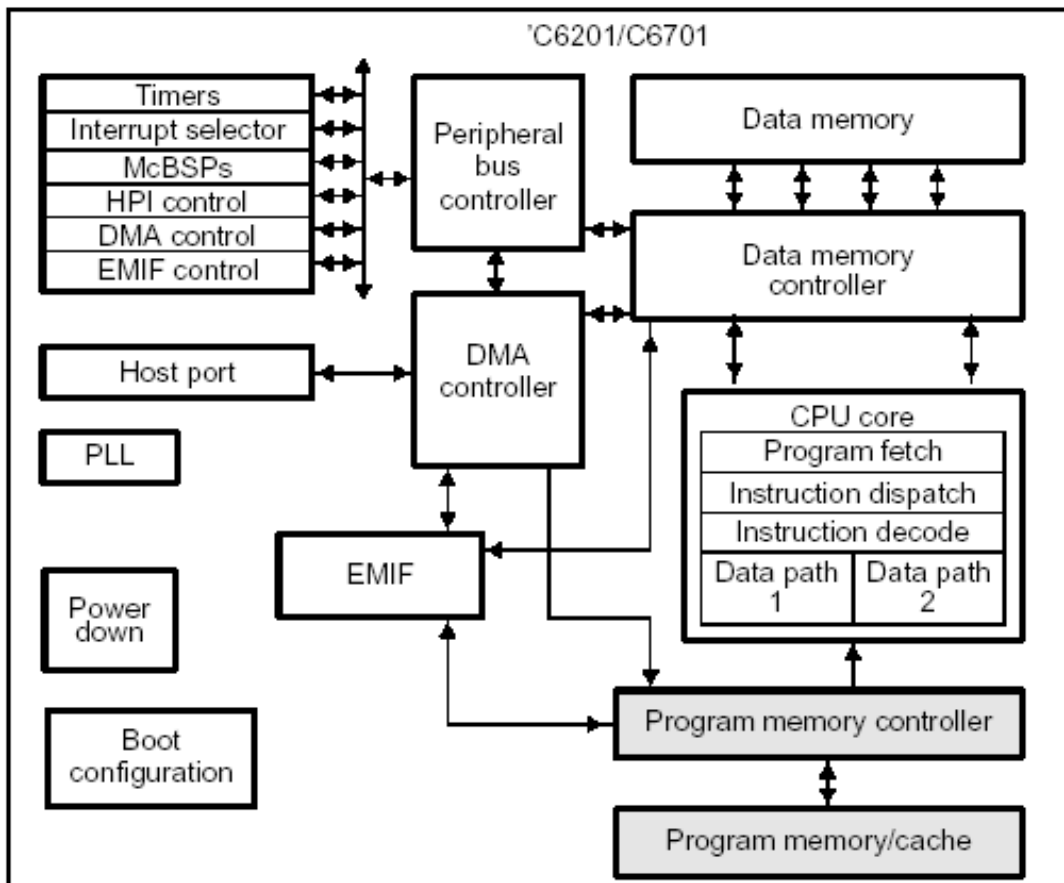


圖 4.1 : TMS320C6201 系統架構方塊圖

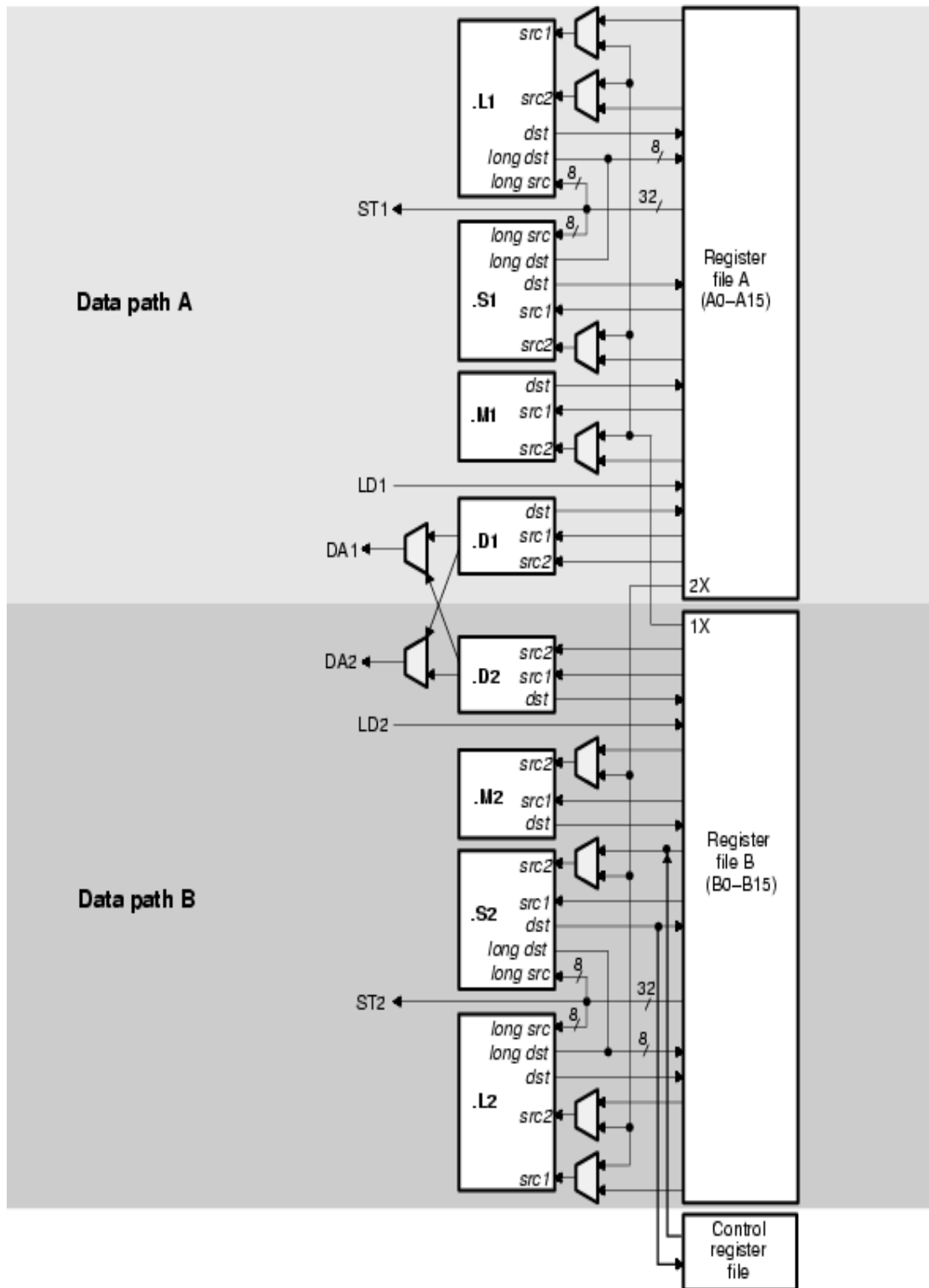


圖 4.2 : TMS320C6201 資料存取路徑

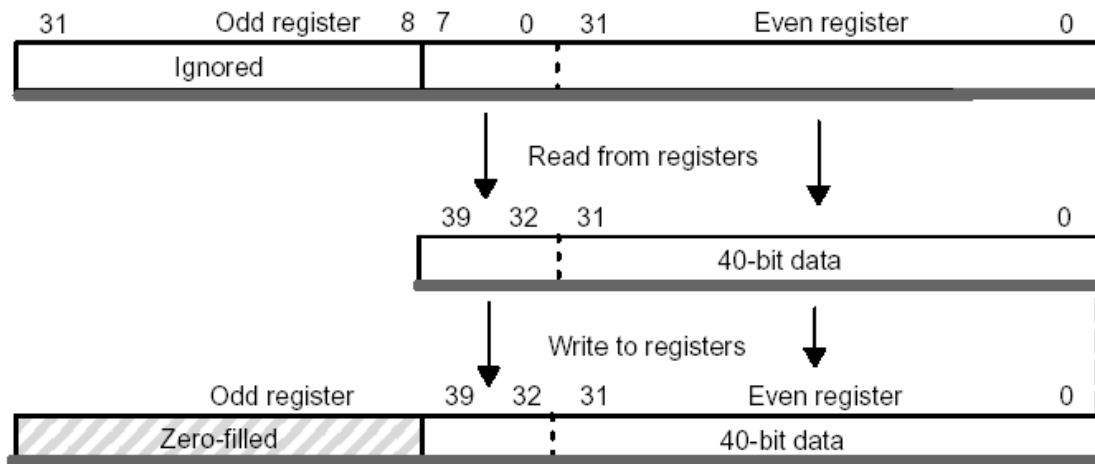


圖 4.3：TMS320C6201 暫存器之結合方式

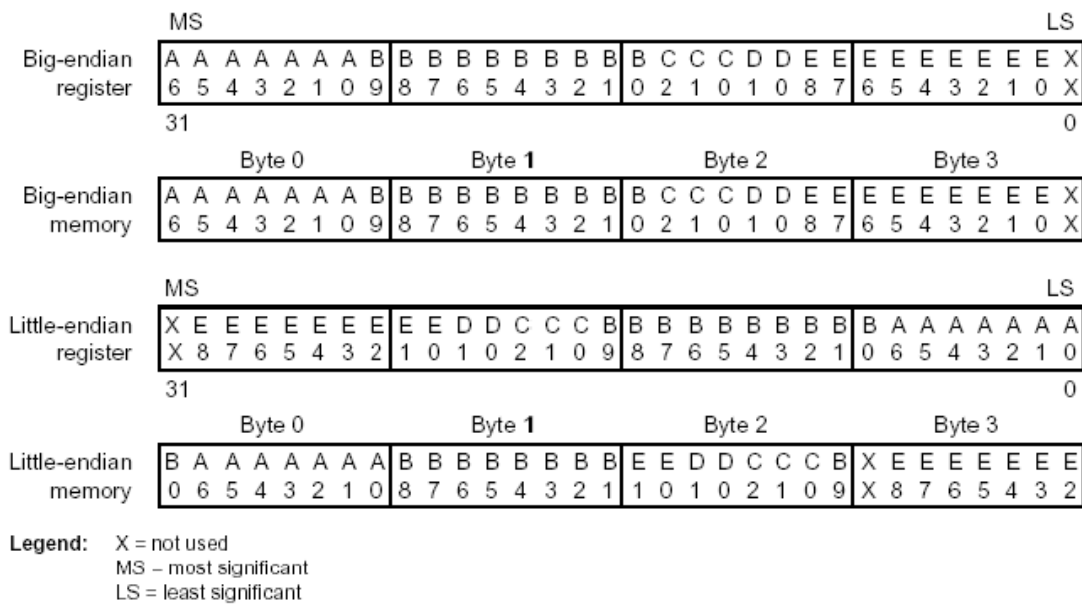


圖 4.4：Big-endian 與 Little-endian 資料排列方式

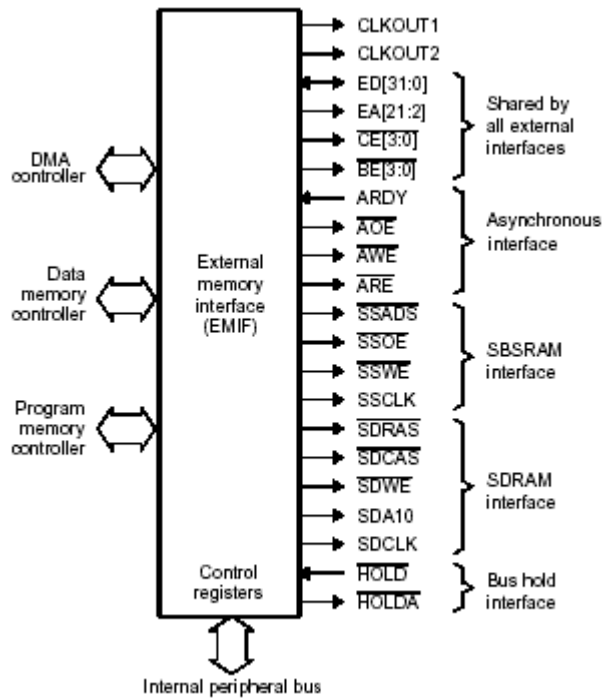


圖 4.5：TMS320C6201 外界記憶體介面方塊圖

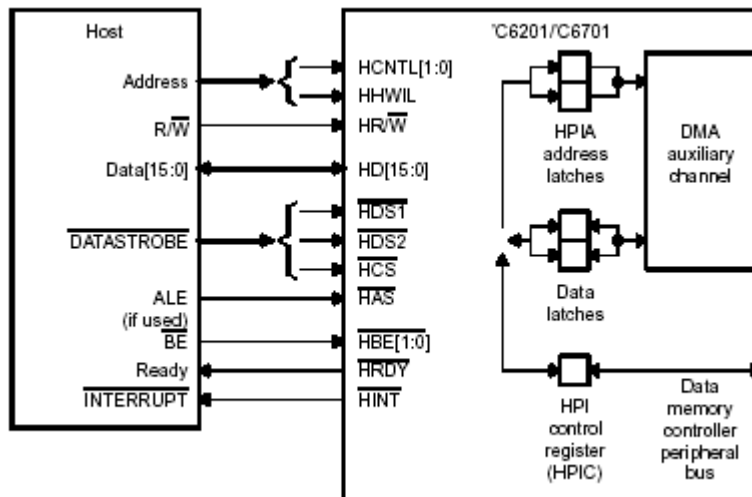


圖 4.6：TMS320C6201 主電腦介面方塊圖

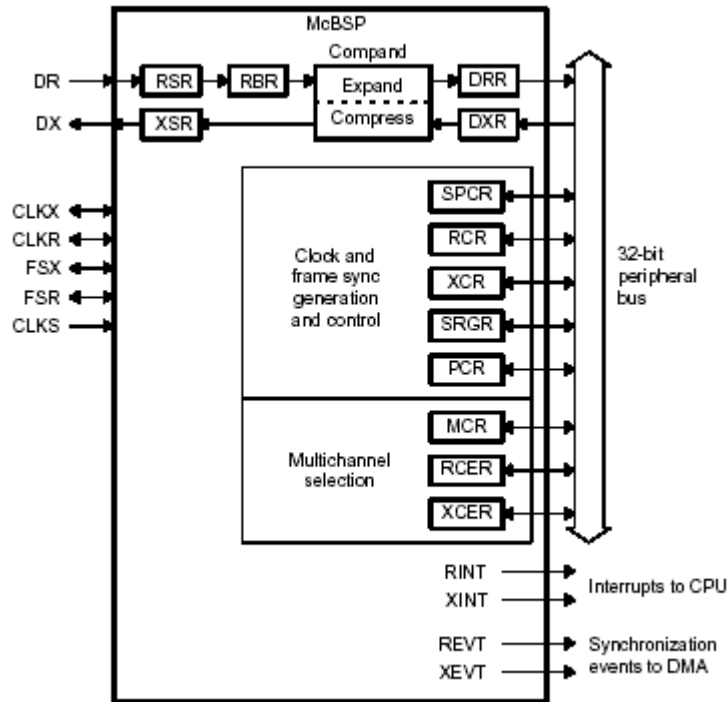


圖 4.7：TMS320C6201 高速串聯埠介面方塊圖

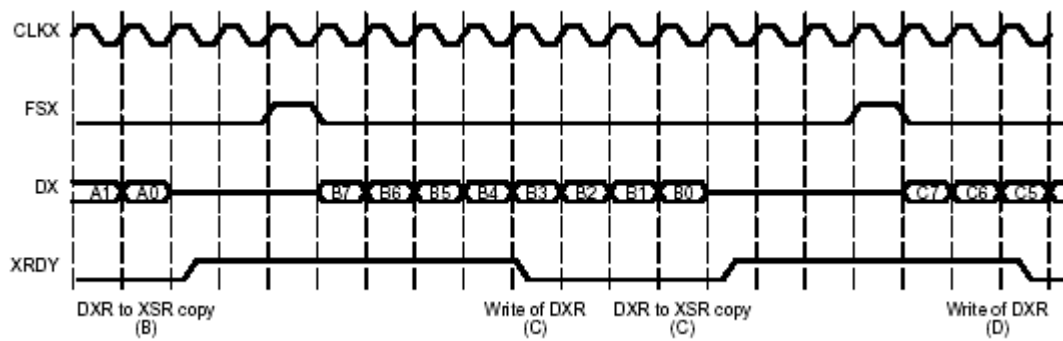


圖 4.8：TMS320C6201 高速串聯埠介面標準運作時序圖

表 4.1：TMS320C6201 暫存器分群及結合法

群組 A	群組 B
A1:A0	B1:B0
A3:A2	B3:B2
A5:A4	B5:B4
A7:A6	B7:B6
A9:A8	B9:B8
A11:A10	B11:B10
A13:A12	B13:B12
A15:A14	B15:B14

表 4.2 : TMS320C6201 記憶體映射表

Address Range (Hex)	Size (Byte)	Description of Memory Block	
		MAP 0	MAP 1
00000000-0000FFFF	64K	External memory interface CE 0	Internal program RAM
00010000-003FFFFFFF	4M-64K	External memory interface CE 0	Reserved
00400000-00FFFFFFF	12M	External memory interface CE 0	External memory interface CE 0
01000000-013FFFFFFF	4M	External memory interface CE 1	External memory interface CE 0
01400000-0140FFFF	64K	Internal program RAM	External memory interface CE 1
01410000-017FFFFFFF	4M-64K	Reserved	External memory interface CE 1
01800000-0183FFFF	256K	Internal peripheral bus EMIF registers	
01840000-0187FFFF	256K	Internal peripheral bus DMA controller registers	
01880000-018BFFFF	256K	Internal peripheral bus HPI register	
018C0000-018FFFFFFF	256K	Internal peripheral bus McBSP 0 registers	
01900000-0193FFFF	256K	Internal peripheral bus McBSP 1 registers	
01940000-0197FFFF	256K	Internal peripheral bus Timer 0 registers	
01980000-019BFFFF	256K	Internal peripheral bus Timer 1 registers	
019C0000-019FFFFFFF	256K	Internal peripheral bus interrupt selector registers	
01A00000-01FFFFFFF	6M	Internal peripheral bus (reserved)	
02000000-02FFFFFFF	16M	External memory interface CE 2	
03000000-03FFFFFFF	16M	External memory interface CE 3	
04000000-7FFFFFFF	2G-64M	Reserved	
80000000-803FFFFFFF	64K	Internal data RAM	
80400000-FFFFFFFF	2G-64K	Reserved	

表 4.3 : TMS320C6201 DMA 同步事件列表

Event Number(Binary)	Event Acronym	Event Description
00000	None	No synchronization
00001	TINT0	Timer 0 interrupt
00010	TINT1	Timer 1 interrupt
00011	SD_INT	EMIF SDRAM timer interrupt
00100	EXT_INT4	External interrupt pin 4
00101	EXT_INT5	External interrupt pin 5
00110	EXT_INT6	External interrupt pin 6
00111	EXT_INT7	External interrupt pin 7
01000	DMA_INT0	DMA channel 0 interrupt
01001	DMA_INT1	DMA channel 1 interrupt
01010	DMA_INT2	DMA channel 2 interrupt
01011	DMA_INT3	DMA channel 3 interrupt
01100	XEVT0	McBSP 0 transmit event
01101	REVT0	McBSP 0 receive event
01110	XEVT1	McBSP 1 transmit event
01111	REVT1	McBSP 1 receive event
10000	DSPINT	Host processor to DSP interrupt
10001	XEVT2	McBSP 2 transmit event
10010	REVT2	McBSP 2 receive event
Other	Reserved	



表 4.4：TMS320C6201 預設中斷向量對應值

CPU In- terrupt	Related INTSEL field	INTSEL Reset Value	Interrupt Acronym	Interrupt Description
INT4	INTSEL4	00100b	EXT_INT4	External interrupt pin 4
INT5	INTSEL5	00101b	EXT_INT5	External interrupt pin 5
INT6	INTSEL6	00110b	EXT_INT6	External interrupt pin 6
INT7	INTSEL7	00111b	EXT_INT7	External interrupt pin 7
INT8	INTSEL8	01000b	DMA_INT0	DMA Channel 0 Interrupt
INT9	INTSEL9	01001b	DMA_INT1	DMA Channel 1 interrupt
INT10	INTSEL10	00011b	SD_INT	EMIF SDRAM timer interrupt
INT11	INTSEL11	01010b	DMA_INT2	DMA Channel 2 interrupt
INT12	INTSEL12	01011b	DMA_INT3	DMA Channel 3 interrupt
INT13	INTSEL13	00000b	DSPINT	Host port to DSP interrupt
INT14	INTSEL14	00001b	TINT0	Timer 0 interrupt
INT15	INTSEL15	00010b	TINT1	Timer 1 interrupt

## 第五章

### 實作過程分析與結果

在講求實用為主的真實系統中，效率乃為一非常重要之考量因素。如何設計出正確且執行效率高的系統，軟體與硬體間的配合是不可或缺的。此章中，吾人將嘗試以前一章節所介紹之 TI TMS320C6201 DSP 處理器實現提出之智慧型犁耙機收器基頻處理端，並驗證與分析在軟體無線電架構基礎下系統實現之可能性。

#### 5.1 發展環境簡介

開發環境依功能可分成 DSP 硬體系統與控制程式碼兩部分。

##### 5.1.1 硬體發展環境

硬體開發環境使用 Blue Wave 公司製作之 PCI/C6600 TBC 發展板，PCI/C6600 的架構如圖 5.1 所示。此發展板之特色如下：

- 1、處理器：TBC 板上共使用兩顆 運算速度為 200Mhz 之 TI TMS320C6201 DSP 處理器，雙 CPU 可互相配合，提供更強大的運算能力。
- 2、匯流排架構：匯流排之分佈如圖 5.1，使用專用之 PCI 匯流排連接各個版上元件，對外透過 PCI-PCI 橋接器與 PC 端之 PCI 匯流排連接，對內則使用 PLX PCI9080 DSP-PCI 橋接器與 DSP 溝通。每顆 DSP 晶片並擁有獨立匯流排，用以連接專屬 DSP 本身之週邊設備，如 SDRAM 與控制用暫存器等。共享匯流排 A (Shared Bus A)與共享匯流排 B (Shared Bus B)則提供 DSP 存取共享記憶體 A (Shared SRAM Bank A)及共享記憶體 B (Shared SRAM Bank B)之路徑。單顆 DSP 晶片對外之介面可以歸納為如圖 5.2 所示。
- 3、記憶體配置：除了單顆 DSP 晶片配置的內部 64K bytes 資料記憶體與 64K bytes 程式記憶體外，每顆 DSP 晶片另配置 16M bytes 之 SDRAM。另有兩組容量皆為 1M byte 之共享記憶體(使用 SRAM)。
- 4、中斷：PCI/C6600 提供許多彈性的中斷，包含 DSP 與主電腦端間之中斷、DSP 與 DSP 間之中斷以及 TBC 板上

元件發出之中斷，如表 5.1 所示。

### 5.1.2 軟體發展環境

DSP 程式可以使用 C 語言或組合語言撰寫而成，組合語言格式又可分成一般線性組合語言與 DSP 專用組合語言兩種。一般而言，使用組合語言開發完成之程式碼執行效率會高於使用 C 語言開發之程式碼。但 TI 提供之編譯器效能已十分接近使用組語開發之程式碼，故程式發展一般為顧及時效性與方便性，會先以 C 語言撰寫[23]，[24]，[26]，再針對程式瓶頸處以組合語言加以改寫。TI 之 C 編譯器遵循 ANSI C 標準，並支援 ANSI C 執行時期函式庫。因硬體架構的緣故，資料型態部分如表 5.3，與一般常見的資料型態略有差異。

編譯器可以產生組合語言原始檔、執行檔之目的檔、置放於 ROM 中之初始程式檔處理器執行檔與等四種輸出。TI 本身提供一套功能強大的整合式發展介面 - Code Composer IDE，吾人可於此系統上撰寫、編譯程式碼，並進行除錯的工作。程式開發流程圖可參考圖 5.3。

## 5.2 定點 DSP 晶片程式設計上之相關考量

由於 TMS320C6201 為一個使用定點數運算單元的 DSP 處理晶片，故

在設計程式架構時必須考慮到使用定點數設計上時的一些問題，如動態範圍、量化、及定點算數運算等。

### 5.2.1 定點數與浮點數

一般所稱之定點數與浮點數表示法，乃是指其儲存一個數字的方式。

IEEE-734 浮點數的表示法較為複雜，如圖 5.4 所示。實際值為：

$$X = (-1)^{-2} \times 2^{(e-127)} \times (1 + .m) \quad (5.1)$$

最小值為  $-(2 - 2^{-23}) \times 2^{127}$

最大值為  $(2 - 2^{-23}) \times 2^{127}$

動態範圍  $\sim \pm 10^{38.53}$

解析度為  $2^{-149}$

定點數的表示方式較為直接，因其以整數型態來儲存數值。一個有號之數值表示方式常記錄成  $A(m,n)$ ，其中最高位元用以表示正負號， $m$  為整數部分之位元數， $n$  為小數點部分之位元數。以 16 位元定點數為例，常用的表示方式稱為 Q-15 格式，即為  $A(0,15)$ ，以所有有效位元代表小數，如圖 5.5 所示。實際值為：

$$X = (-1)^s + \frac{f}{2^{15}} \quad (5.2)$$

除最高位元外，其餘數字代表小數點部分之值。在有號數情況下：

最大值表示成 0x7fff，實際值為 0.9997。

最小值表示成 0xffff，實際值為 -1。

動態範圍為  $\frac{1}{2^{-15}} = 32768$ 。

精確度為  $\frac{1}{2^{15}} = 0.000030518$ 。

Q-15 之環狀示意圖可參考圖 5.6。因 \$Q-15\$ 只能表示小數。儲存成 Q-15 格式之前，需先對數值作正規化的動作，並另外記錄正規化係數之值大小，用於運算後回復原正確值。由精確度可知，最小的差異值決定因素在於  $n$  位元數的多寡，若欲表示值接近此數量級，則必須考慮以更多的位元數來表示小數部分，以降低量化誤差。

## 5.2.2 定點數計算

浮點數處理器因擁有特殊的浮點運算單元與相關暫存器，可以對浮點數值直接以硬體處理，並獲得較為精確的結果。但定點數運算[25]並非就不能執行浮點運算，只是需要利用軟體的方式完成[29]，故效能會有很大的差異。

### 定點數加法器

定點數加法器運算如圖 5.7 所示，運算動作可於一個時脈內完成。定點數加法需考慮到溢位(overflow)問題，即運算結果超出所能儲存位元大小。此時需根據所需結果的形式決定使用更多的位元儲存或是選擇以飽和(saturation)方式處理。

### 定點數乘(除)法器

定點數乘法器運算如圖 5.8 所示，16 位元乘法運算動作可於兩個時脈內完成。定點數乘法需考慮到截斷(truncation)問題，以 16 位元相為例，需使用 32 位元儲存結果，如圖 5.9 所示。若不需保留精確度，則可直接捨棄多餘位元。

## 5.2.3 程式碼最佳化

DSP 程式執行效率決定了系統效能。為達成程式碼最佳化的目的，使用正確的工具幫助調整程式架構為必要的動作。TI 提供一個程式發展流程建議，如圖 5.10。由於吾人主要以 C 語言開發程式，如何寫出有效率的程式，TI 提供了幾點建議[24]：

- 1、使用編譯器內附參數：TI C 編譯器本身提供了許多可增加效能的參數，並提供不同等級的最佳化效果，一般程式皆可藉由調整此參數達到一定程度的最佳化。
- 2、降低程式相關性：程式相關性對程式的效能影響很大，但

降低程式相關性牽涉到演繹法則(algorithm)的問題，並非能真正得到有效的解決方案。若能有效降低程式相關性，則程式執行時會增加平行處理的次數，有效增加程式的效能。

- 3、展開迴圈(unroll loop)：迴圈程式也是拖慢效能的原因之一，因迴圈架構處無法有效平行化，且因會有多次 branch 發生，會造成許多延遲。若迴圈數目不多，則應以展開方式處理。
- 4、使用線性組合語言改寫：以線性組合語言針對程式瓶頸處加以改寫，為最佳化最有效之作法，若能自行計算程式執行所需時脈數，搭配指令的應用，則可達到最高的執行效率。

### 5.3 系統架構

此系統可依功能大致區分成三個部分：主電腦端(HOST)，DSP 傳送端與 DSP 接收端。主電腦負責傳送與接收 DSP 資料的工作，DSP 傳送端負責將資料透過高速串聯埠送至 DSP 接收端處理。整個資料輸出入流程如圖 5.11 所示。發送訊號的產生，吾人採用 PC 端模擬的方式。將欲傳送資訊先行轉換成(1, -1)的位元信號後，再通過以 Matlab



模擬之通道，產生接收端模擬訊號。為使接收端 DSP (DSP 1)接收資料模式能接近實際系統，吾人使用另一顆 DSP (DSP 0)為資料傳送器，隨時對 DSP 1 提供資料。DSP 0 本身需維護一個發送暫存記憶體，並建立與 PC 端的連線，以確保發送暫存記憶體隨時保持足夠的資料量。DSP 1 則為主要接收機程式部分，本身維護一個時槽長度的暫存器，作為外界輸入資料的存放位址。當資料解碼後，可選擇透過 Code Composer IDE 驗證資料，或將資料送回 PC 端完成後續的處理工作。以下個章節分別詳述其細部功能與實作方式。

### 5.3.1 主電腦端

主電腦端負責個程式的控制部分，包括下載 DSP 程式碼執行檔與初始設定、下載系統參數、傳輸資料與接取資料。主電腦端在控制 DSP 之前，需先取得 TBC 板的控制指標(handle)，再取得 DSP 的控制指標，才能直接透過指標控制 DSP。取得 DSP handle 後，DSP 必須先設定成重置狀態(RESET)，以進行初始動作。初始步驟依序為：

- 1、下載 DSP 程式碼：下載程式碼為所有重置動作的第一步驟，取得必要 handle 後，便可以對所欲下載的 DSP 處理器下載程式碼。此時 DSP 應處於重置狀態，等待解除狀態後，DSP 會從跳進程式進入點開始執行。於重置到解

除重置（此時 DSP 開始動作）期間的初始化動作包含中斷程式的安置與參數的載入等。

2、系統參數載入：由於使用 parameter-download-type 的觀念實作，吾人將所有系統相關參數獨立成一個如下之結構

(Structure)：

```
*****  
Structure  
*****/  
struct param_table_t  
{  
/* 3gpp Spec. */  
unsigned int slot_num;  
unsigned int slot_length;  
unsigned int sf_I ;  
unsigned int sf_Q ;  
unsigned int scramble_length;  
unsigned int data_length;  
unsigned int pilot_length;  
  
/* System architecture & simu. param.*/  
unsigned int ant_num;  
unsigned int path_num;  
unsigned int frame_num;  
float tpc_ref;  
float mu;  
unsigned int architecture;  
unsigned int data_norm;  
  
/* chanel */  
unsigned int max_delay;  
};
```

3、在下載 DSP 程式碼後，透過下載系統參數的方式，可動態調整系統的特性。系統參數的位址為 0x80000020，總長度為 0x40。詳細結構可參考表 5.2。

4、傳輸資料與接取資料：因主電腦無法直接得知 DSP 執行時期(run-time)的變數資訊，必須由 DSP 端提供相關數值。因 DSP 與主電腦端無 mailbox 暫存器存在，此處使用的作法稱為 virtual-mailbox-interrupt，亦即 DSP 於發出資料存取請求時，必須於特定位址(需於 DSP 內部保留一特定位址空間存放 mailbox 值)寫入相關參數，再發出 mailbox 中斷。主機端在收到此一中斷信號後，會先判斷訊號屬於何種要求，再讀取 mailbox 內容，並進一步根據其內容作讀寫的動作。資料傳輸流程如圖 5.12。因使用中斷方式交換資訊，必須於主程式安置對應之中斷程式，一旦接收到中斷信號後(在此為 DSP 之 HINT 信號)，中斷程式即可被呼叫執行。

### 5.3.2 DSP 傳送端

DSP 傳送端使用 TBC 板上編號為 DSP 0 之 DSP 處理器。主要功能為傳送資料至 DSP 接收端或從主電腦端接收資料。為能持續應付 DSP 接收端的資料傳輸要求。DSP 0 部分規劃出兩個時框容量的暫存區，一個暫存區為作用中時框，另一個暫存區則預先存入下一次需使用之時框。當 DSP 接收端要求資料時，傳送端必須負責將資料送至接收

端，在真實系統中，DSP 資料輸出、入標準介面是以高速並聯埠 (McBSP) 為輸出、入介面。實作的方式則將 DSP 接收端與 DSP 傳送端以 McBSP 相連。在 TBC 版的設計中，每顆 DSP 處理器中的 McBSP 皆被連接到 TBC 板上獨立的輸出接腳，接腳的排列形式如圖 5.13 所示。若以特殊的訊號線直接連接兩端接腳，則一端的框同步訊號輸出腳位與時脈訊號輸出腳位會被對應到另一端的框同步訊號輸入腳位與時脈訊號輸入腳位，資料輸出腳位與資料輸入腳位會被對應到另一端的資料輸入腳位與資料輸出腳位，可以使傳送端為主控端 (Master)，接收端為受控端 (Slave)，兩邊互相同步傳接資料[28]。

McBSP 的設定流程如圖 5.14，以此程序設定 McBSP 後，當接收端進入接收模式時，會持續等待傳送端的框同步訊號，準備接收訊號。雙邊時脈以傳送端的時脈為基準，以確保資料傳送時的同步正確。McBSP 資料傳輸速率受內部時脈產生器之除頻器與 DSP 時脈控制。

為達成 3.84Mcps 要求，除頻值 CLKGDV 須設定成：

$$\frac{f}{\text{ChipRate}} = \frac{200\text{Mhz}}{3.84\text{Mhz}} = 52.083 \quad (5.3)$$

上式中， $f_{\text{DSP}}$  為 DSP 處理器時脈速度。CLKGDV 則應設定為：

$$\text{CLKGDV} = 52 - 1 = 51 \quad (5.4)$$

實際傳送速率：

$$\frac{200\text{Mhz}}{52} = 3.8462 \text{ Mhz}$$

### 5.3.3 DSP 接收端

DSP 接收端使用 TBC 板上編號為 DSP 1 的 DSP 處理器，為系統主架構部分。資料輸出、入的部分採用 polling 的方式，每次向傳送端要求一個時槽的資料量。傳送端與接收端的溝通以信箱暫存器(mailbox register)為主。本地端欲接收資料，需先設定輸出信箱暫存器(out-going mailbox register)，TBC 板則會對另端 DSP 發出信箱暫存器寫入信號，並發出一中斷信號。主程式可分成：

- A. Despreading and descramble
- B. Adaptive-beam Rake Receiver
- C. Fixed-beam Rake Receiver

原理大致已於第三章中討論過，在此僅簡述其實作時之考量：

#### A. Despreading and descramble

時槽長度為 2560，攪亂碼長度為 256，專用實體控制通道碼與專用實體資料通道碼長度分別為 256 與 32。因解攪亂碼不影響原資料長度，

可以攪亂碼長度 256 為單位，同時解出兩通道資料。解碼發方式以 if 判斷式與加減法為主，減少乘法器使用，以減少運算所需時脈數。天線數為 4 與路徑數為 2 為例，總共需重複解出 8 次。此處最作佳化程式分析設計則可大幅增加速度。

### **B. Adaptive-beam Rake Receiver**

程式使用前述理論實作，主要運算部分在於可適性天線權值求解，以定點數實作時，需考慮到動態範圍過大的情形。吾人以 16 位元 short 格式 Q-15 儲存量化輸入訊號，亦即訊號正規係數為 6，最大值為  $2^6 - 2^{-9} = 63.998$ ，精細度(量化誤差)為  $2^{-9} = 0.0019531$ 。式 3.19 後半部運算更新值，因動態範圍過大，為保存最大精確值，則另以 32 位元儲存。考慮到速度與乘法運算結果，更新後權值仍以 16 位元儲存，忽略掉  $2^{-15}$  以下之運算誤差。正規化的除法運算因誤差過大，需另行設計。此處則以 TI 內附之 `_divf()` 函式[27]來處理除法部分。`_divf()` 為一模擬浮點數運算之單精度定點數函示，吾人需犧牲部分效能，以換取精確度。

### **C. Fixed-beam Rake Receiver**

程式使用前述理論實作，主要運算部分在於權值選擇器的部分，權值以 Q-15 格式儲存，權值參數檔則另以文字檔形式儲存於程式內，已便隨時更動。因運算量簡單，執行效率比 Adaptive-beam Rake Receiver 高。程式結果統計將列於後表。

## 5.4 結果與效能分析

在此章節中，吾人嘗試比較各種系統的模擬結果，並顯示 DSP 實作結果，使用之環境變數如表 5.4 所示。使用可適性空時接收機的模擬結果如圖 5.15 所示。使用固定波束式空時接收機的模擬結果如圖 5.16 所示，兩者的在相同訊號干擾比的環境下之比較如圖 5.17 所示。使用可適性陣列天線在第十個時框結束時的波束圖形為圖 5.18，可見得已有效消除干擾訊號。在採用固定式天線陣列時，由圖 5.19 中可以看出其波束方向均分於  $-60^{\circ} \sim 60^{\circ}$  間。圖 5.20 為使用可適性陣列天線架構下，實作結果與模擬結果波束形成圖的比較，在此可看見第二條路徑之波束圖形略為不同，可見使用定點數運算已發生些許誤差，其第一個使用者之第一條路徑訊號錯誤值收斂圖如 5.21 所示，由圖可見權值在第一個時框結束後已大致收斂。程式之效能與統計資料(使用最佳化參數 -o3, -mt, -pm)可參考表 5.6，不同的最佳化程度對執行效能會有不同的影響，如表 5.7 與圖 5.22 所示，差異最大者為解展頻速度，此處應為程式執行瓶頸。另一需消耗大量時脈之程式為資料輸出入控制函示，因所使用之 PCI/C6600 周邊函示需配合版上周邊硬體控制，採用 polling 接收資料方式也會浪費等待時間，有此執行結果應屬合理。由表可知使用可適性陣列天線的效果較佳，但執行

速度較慢。使用固定式陣列天線的效果較差，但執行速度較快。記憶體使用配置表則列於表 5.5。



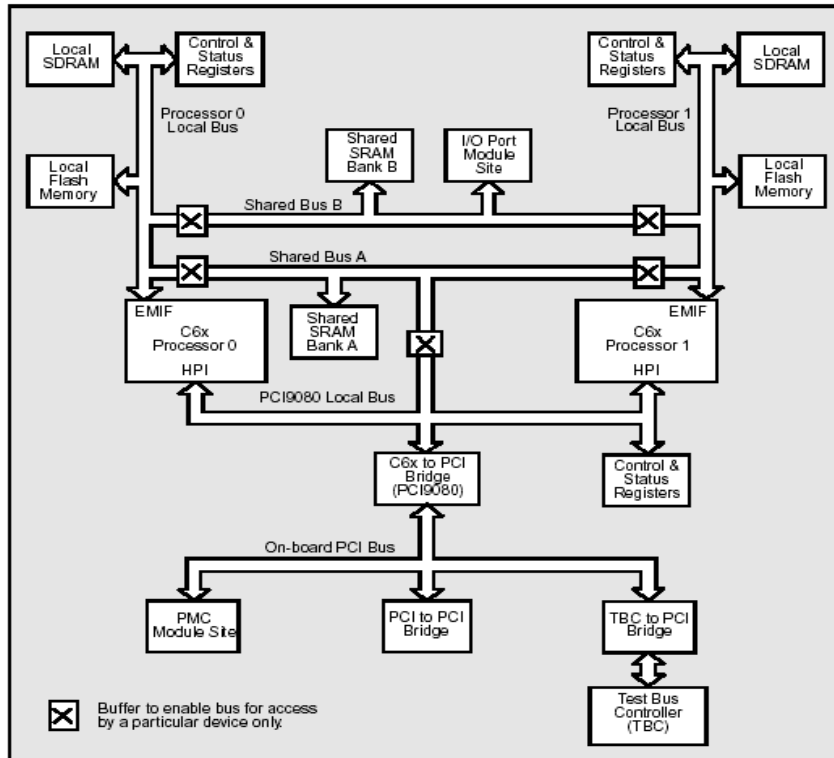


圖 5.1 : PCI/C6600 方塊圖

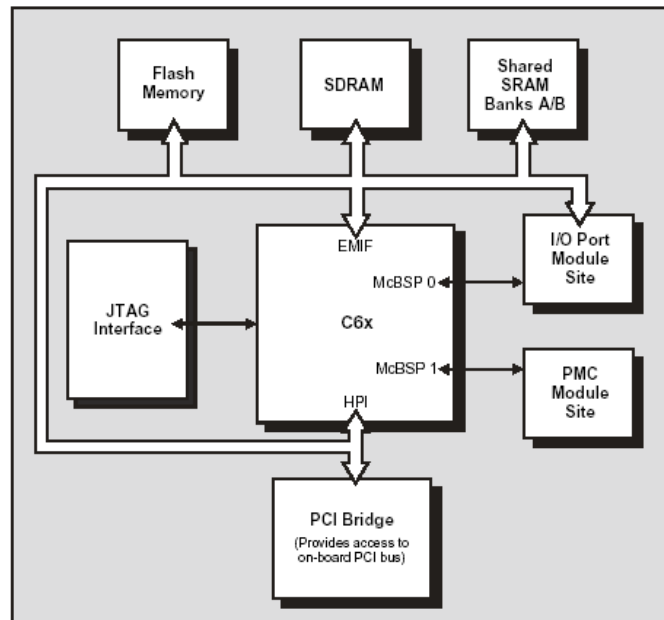


圖 5.2 : PCI/C6600 DSP 處理器聯外界面方塊圖

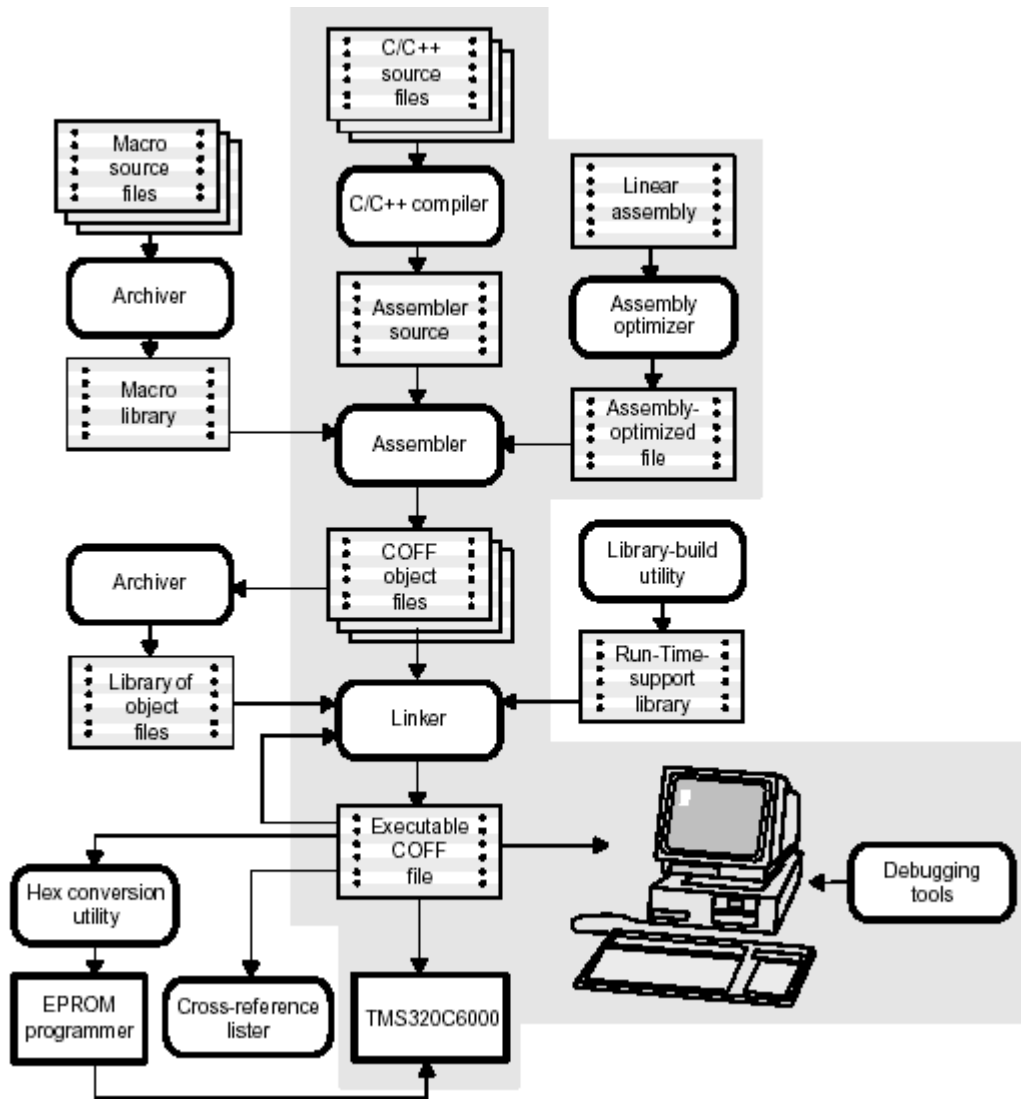


圖 5.3 : TMS320C6x 軟體開發設計流程圖

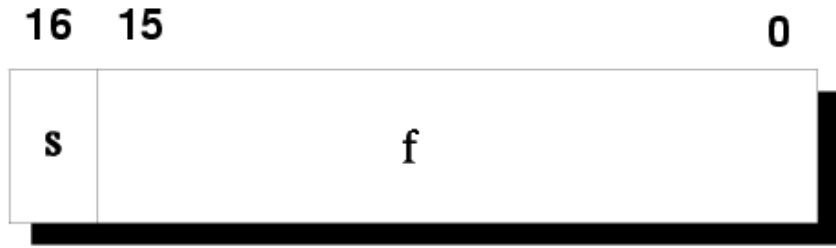


圖 5.4：單精度浮點數格式圖



圖 5.5：16 位元定點數格式圖

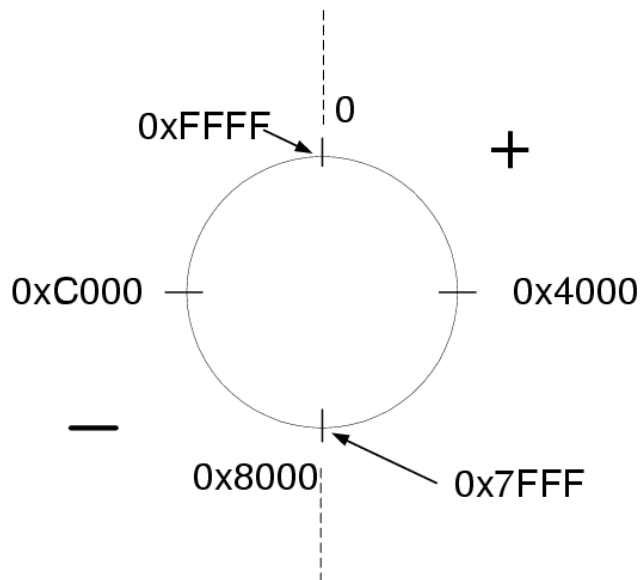


圖 5.6：16 位元定點數表法環型示意圖

ADD .L2X A1,B1,B2

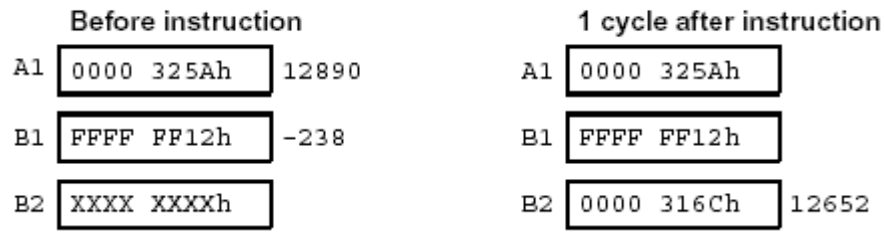


圖 5.7：TMS320C6201 加法器運算範例

MPY .M1 A1,A2,A3

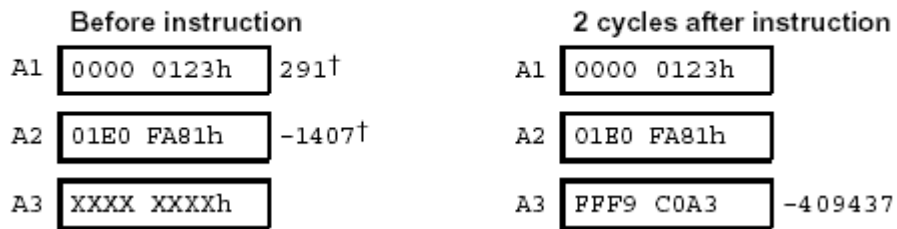


圖 5.8：TMS320C6201 乘法器運算範例

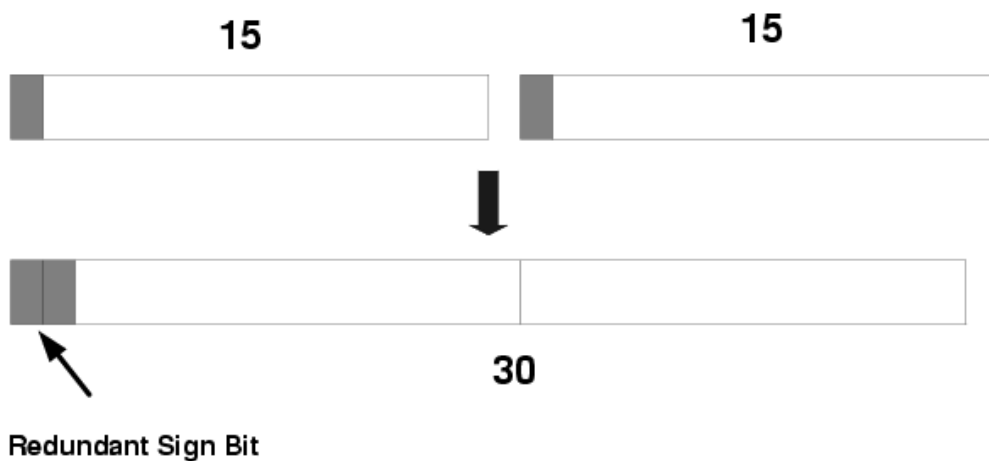


圖 5.9：定點數乘法運算結果

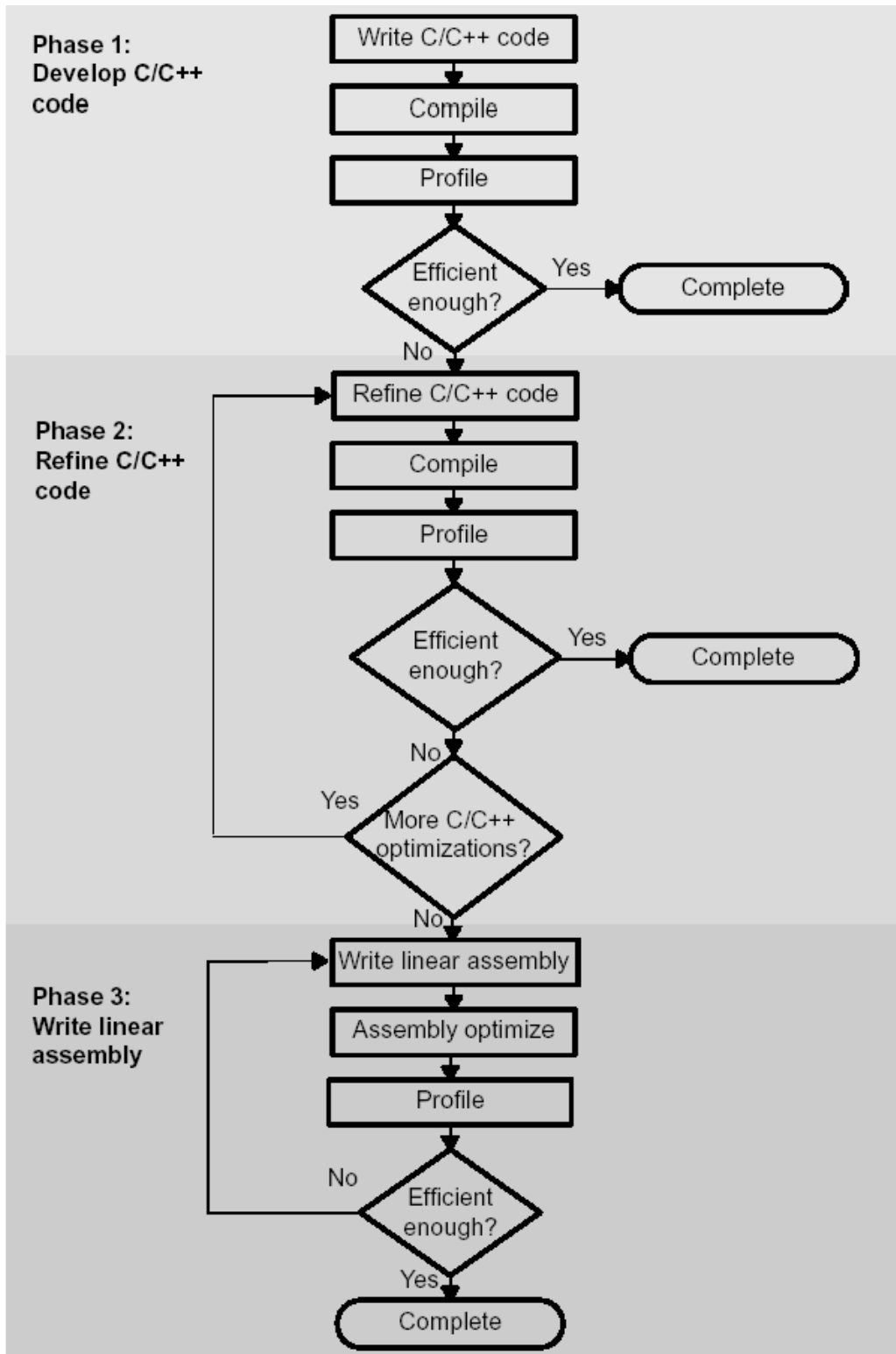


圖 5.10 : TMS320C6201 程式最佳化流程建議圖

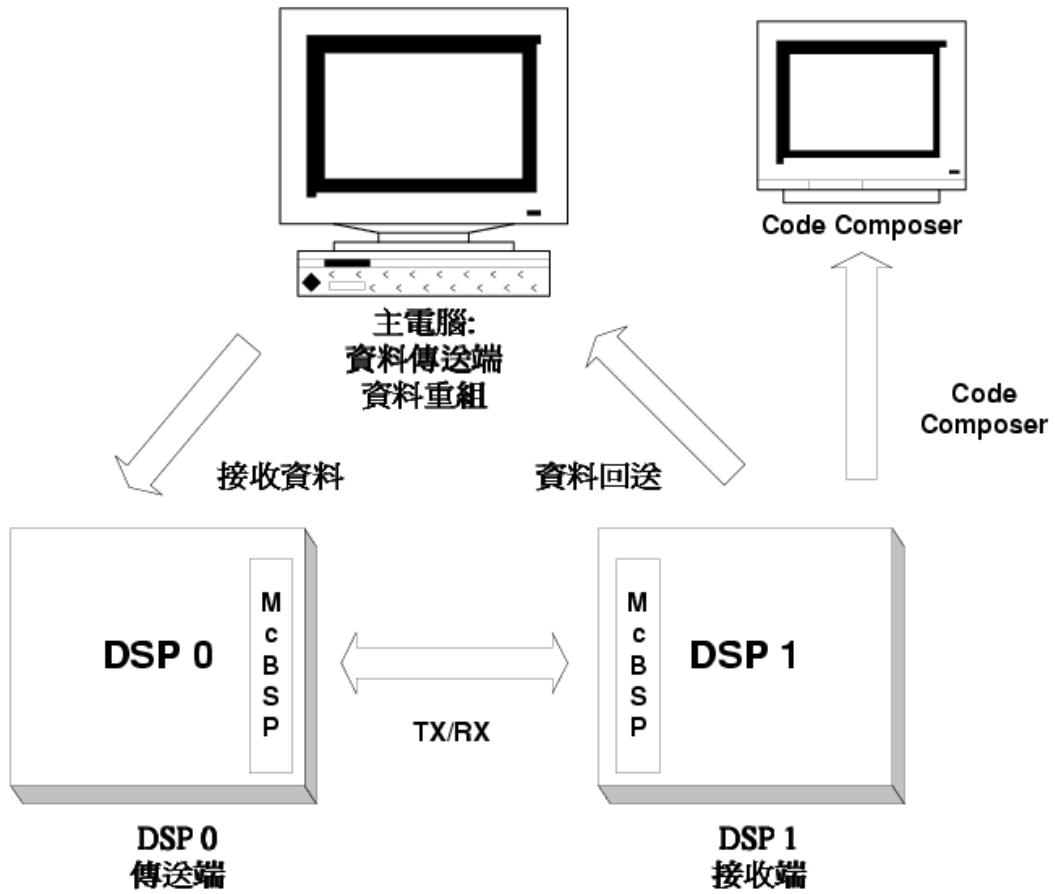


圖 5.11：接收器程式發展環境示意圖

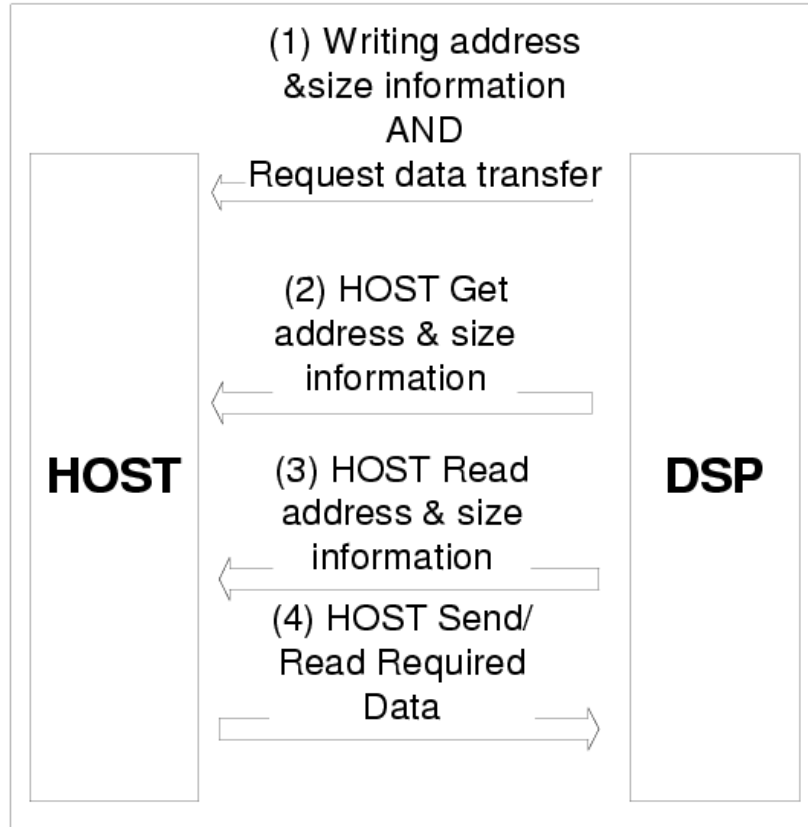


圖 5.12：DSP 與主電腦資料交換步驟流程圖

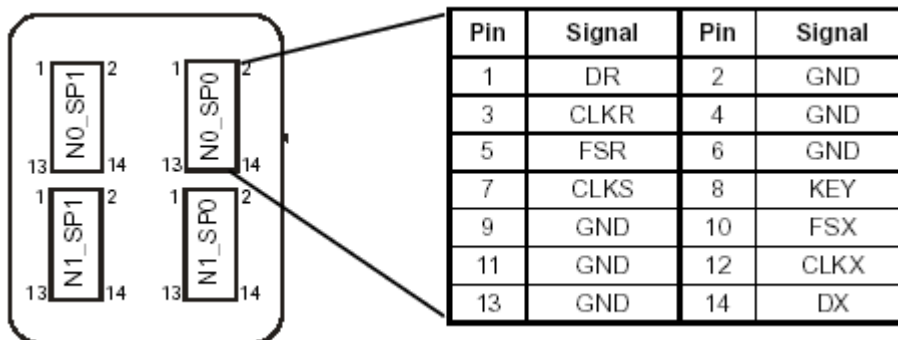


圖 5.13：PCI/C6600 高速串列埠介面接腳圖接腳圖

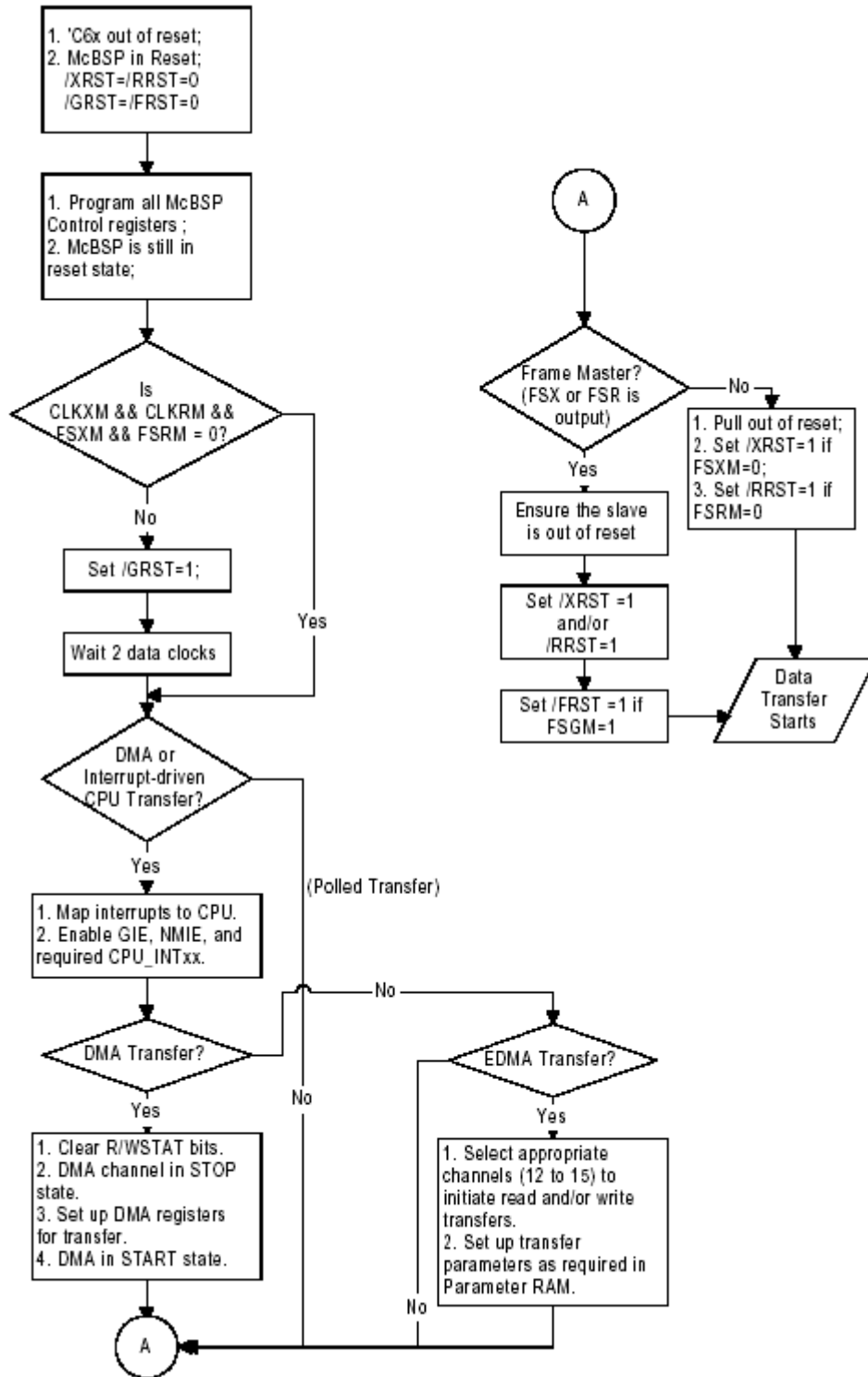


圖 5.14：高速串列埠介面 Master-Slave 傳輸模式設定流程圖



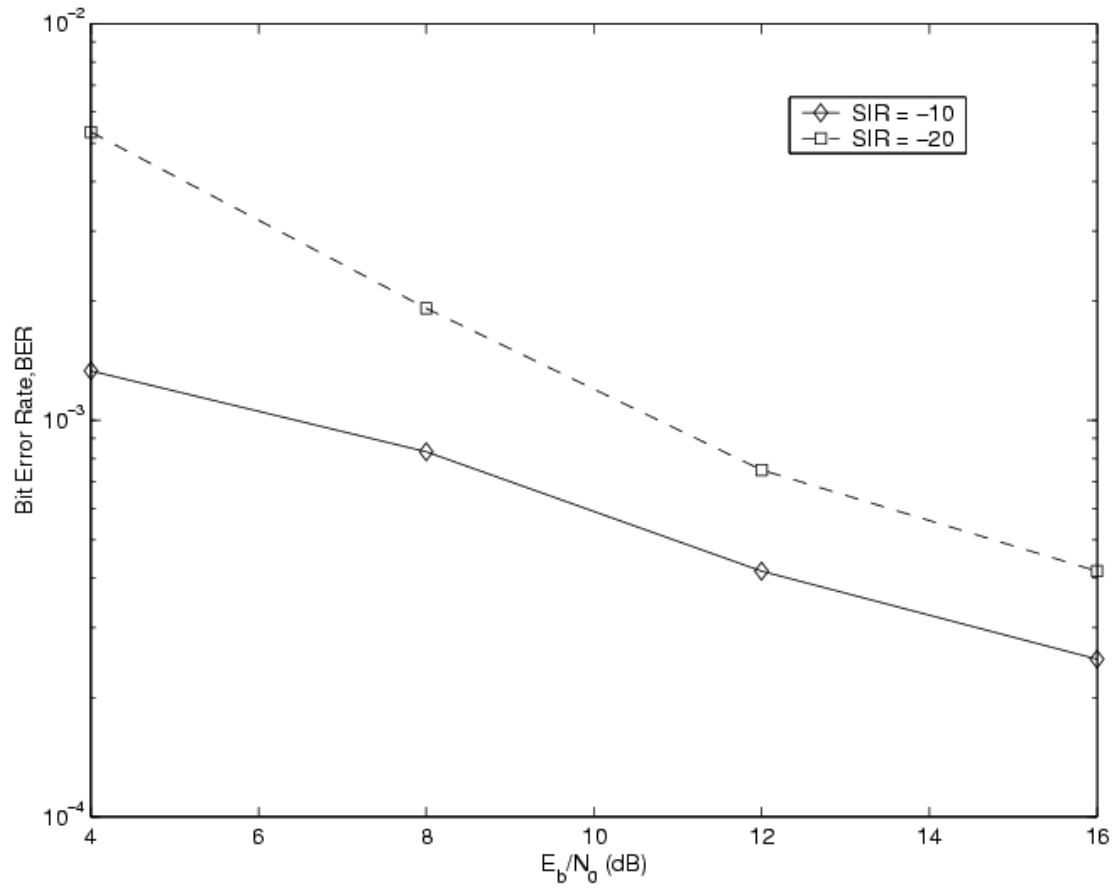


圖 5.15：Adaptive-beamforming 架構下 BER 與  $E_b/N_0$  之關係圖。(使

用參數: adaptive beamforming,  $J = 4$ , user = 2, path = 2)}

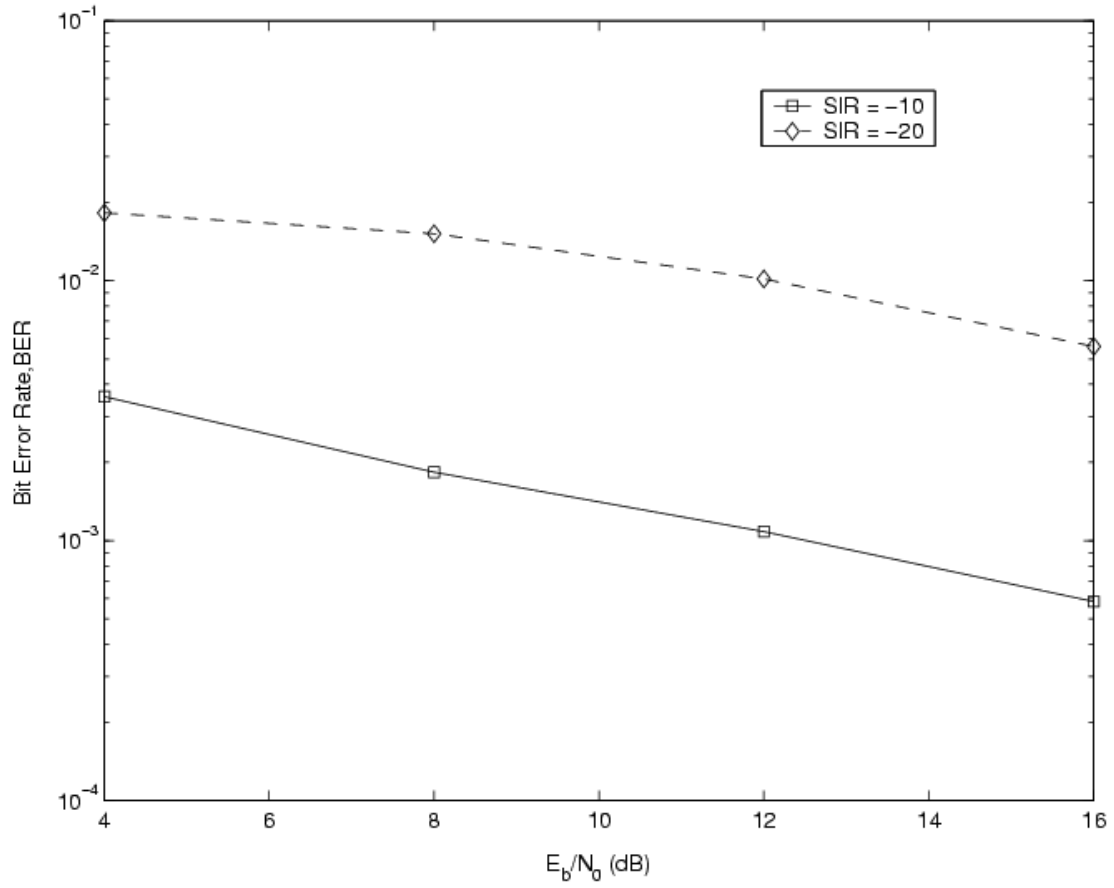


圖 5.16 : Fixed-beamforming 架構下 BER 與  $E_b/N_0$  之關係圖。(使用

參數:  $J = 4$ ,  $user = 2$ ,  $path = 2$ )

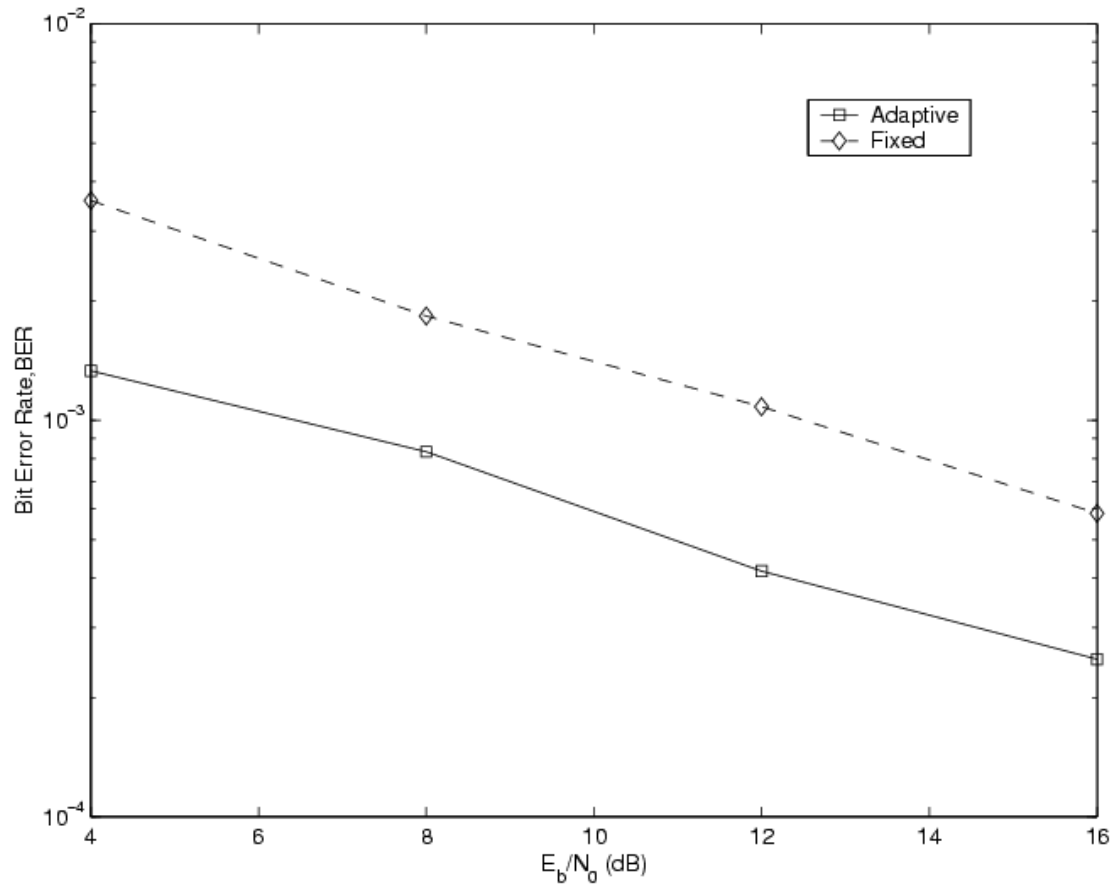


圖 5.17 : Fixed-beamforming 與 Adaptive-beamforming 的效能比較圖。

(使用參數:  $J = 4$ , user = 2, path = 2, SIR = -10 dB)

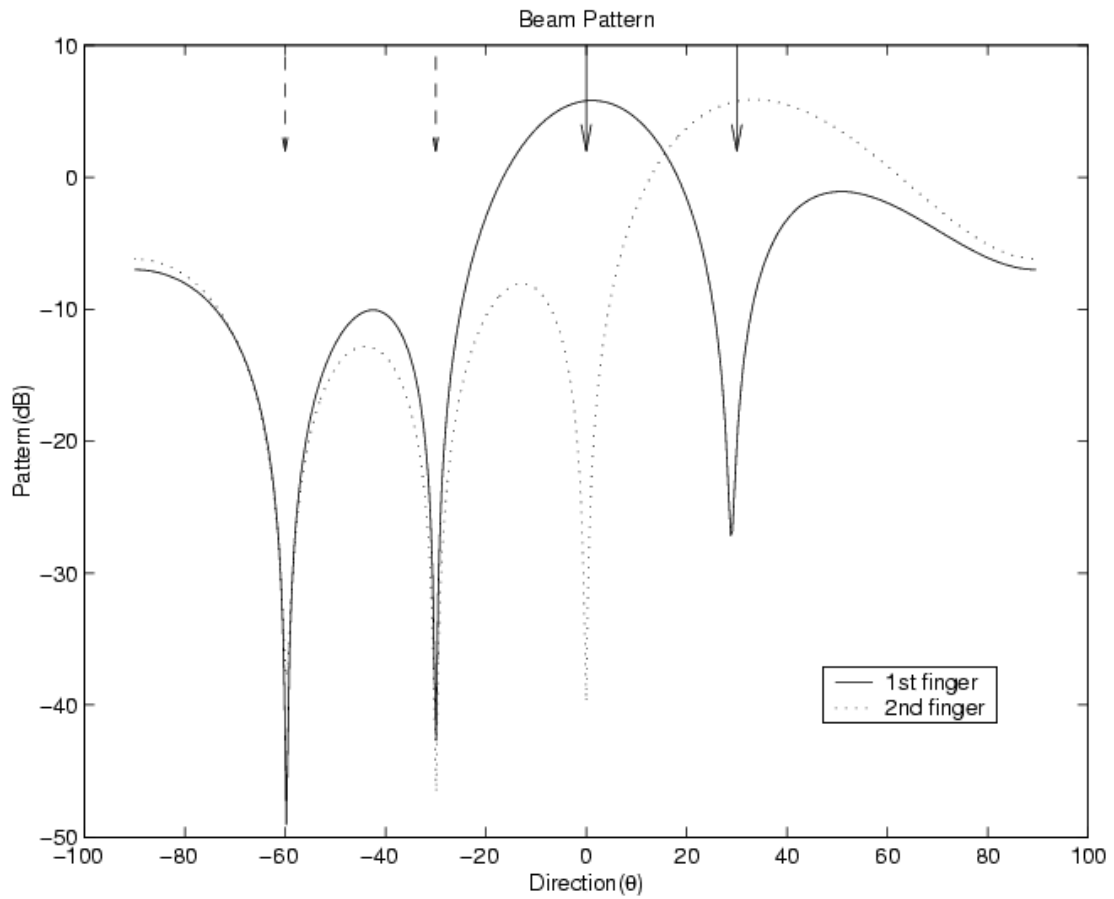


圖 5.18 : Adaptive-beamforming 之波束圖形(使用參數:  $J = 4$ , users = 2, path = 2, SIR = -10 dB,  $E_b / N_0 = 12$  dB, frames = 10)

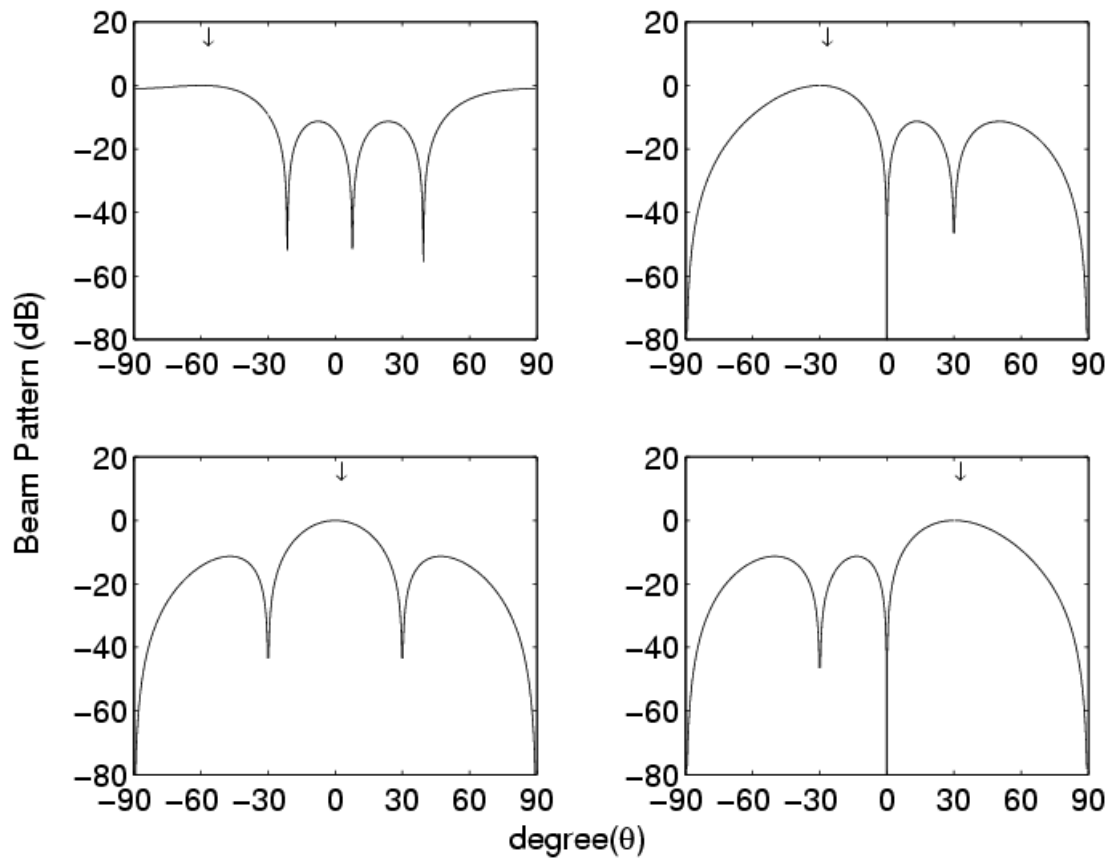
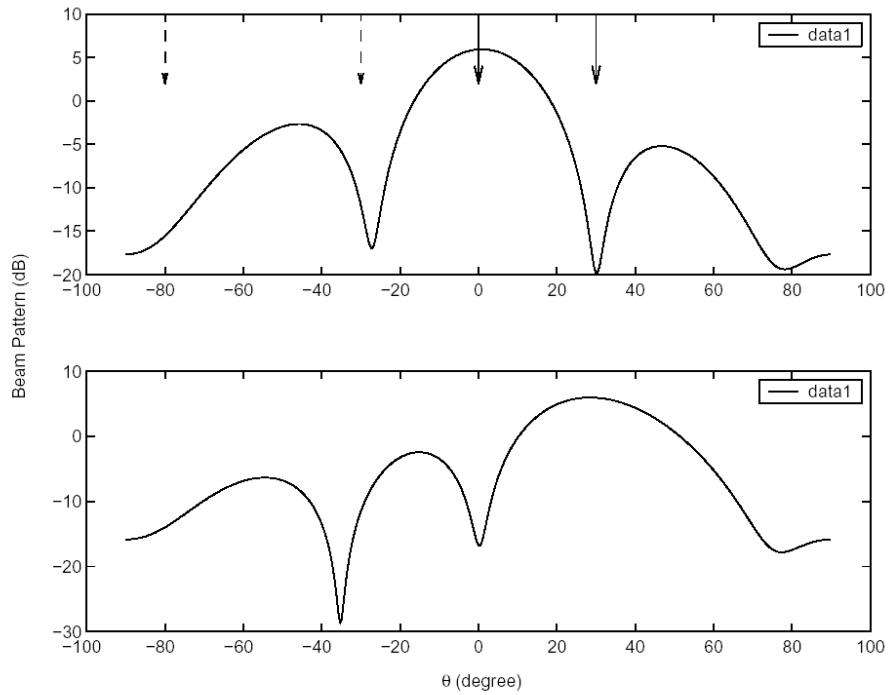
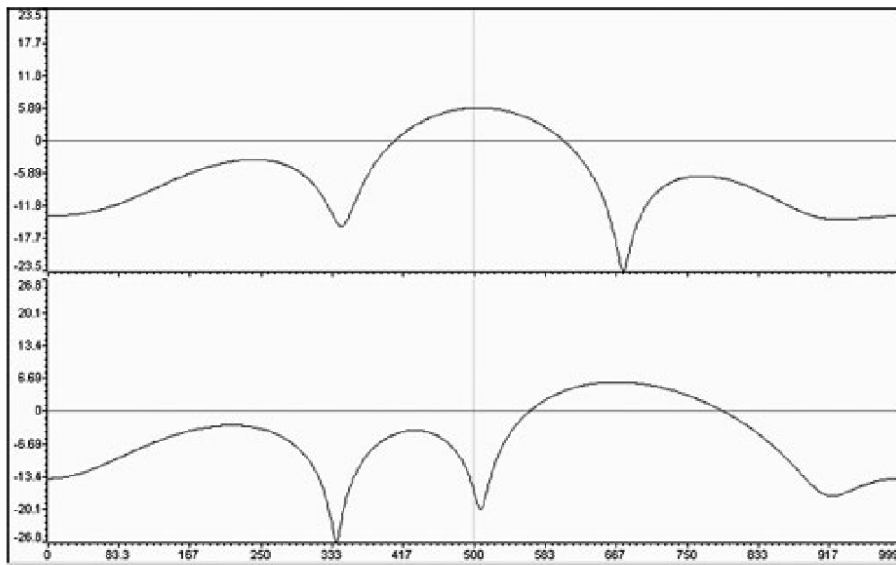


圖 5.19 : Fixed-beamforming 四個方向之波束圖形(使用參數:  $J = 4$ ,  
 $users = 2$ ,  $path = 2$ ,  $SIR = -10$  dB,  $E_b / N_0 = 12$  dB)

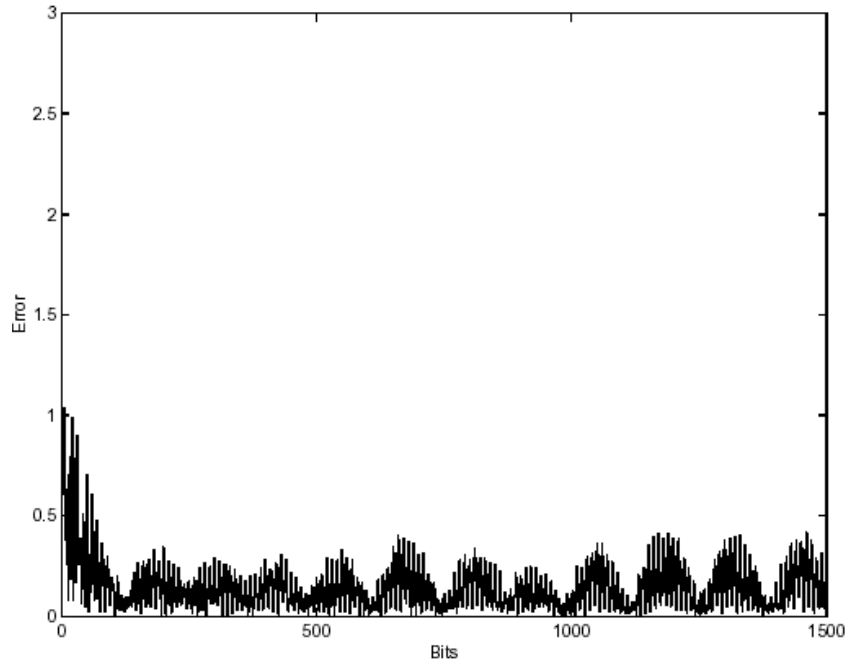


(a) C simulation result

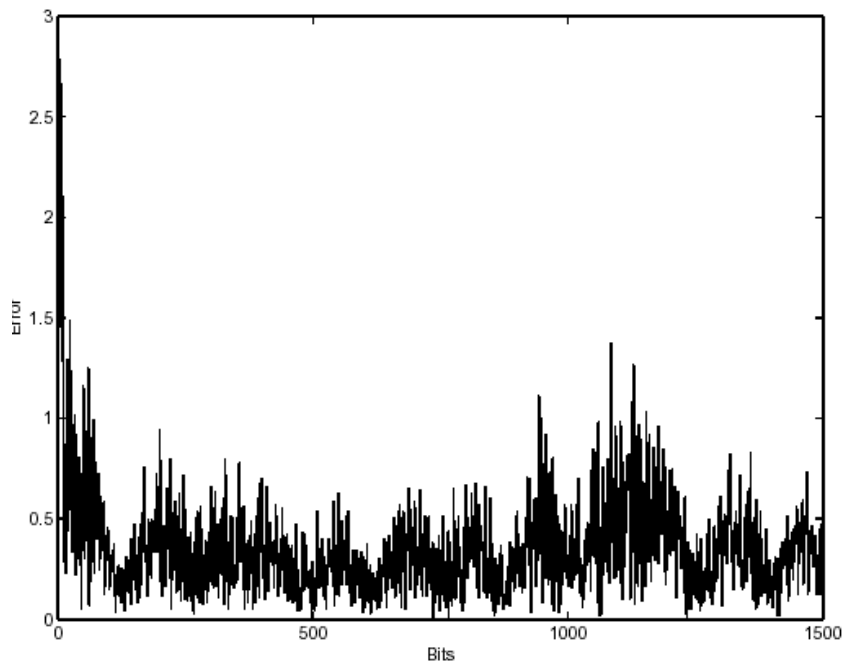


(b) DSP implementation result

圖 5.20：實作結果與模擬結果波束比較圖(可適性陣列天線)。(a)模擬結果 (b)DSP 硬體實作結果。(使用參數:  $J = 4$ ,  $user = 2$ ,  $path = 2$ ,  $SIR = -10$  dB,  $E_b / N_0 = 12$  dB,  $frames = 10$ )



(a) C simulation result,  $k = 1, l = 1$



(b) DSP implementation result,  $k = 1, l = 1$

圖 5.21：實作結果與模擬結果錯誤值比較圖(可適性陣列天線)。(a)模擬結果 (b)DSP 硬體實作結果。(使用參數:  $J = 4$ ,  $user = 2$ ,  $path = 2$ ,  $SIR = -10$  dB,  $E_b / N_0 = 12$  dB,  $frames = 10$ )

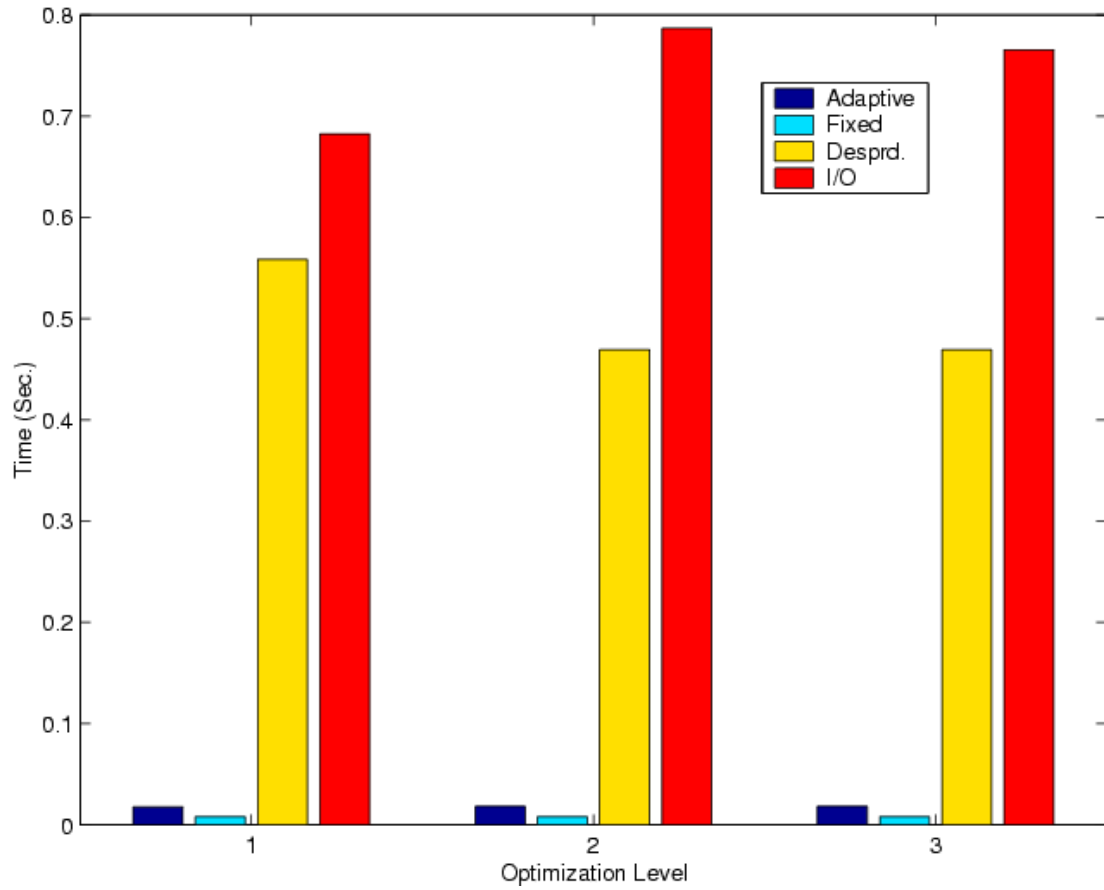


圖 5.22：程式執行所需時間與最佳化關係圖。(使用參數:  $J = 4$ ,  $user = 2$ ,  $path = 2$ ,  $SIR = -10$  dB,  $E_b/N_0 = 12$  dB,  $frames = 1$ , 最佳化參數: -pm, -mt)



表 5.1：PCI/C6600 中斷對應表

Interrupts from:	Interrupts to:				
	HOST PC	C6x DSPs	I/O Port	PMC	PCI9080
HOST PC		√	×	×	×
C6x DSPs	√		√	√	√
I/O Port	×	√		×	×
PMC	√	√	×		×
SRAM A/B	×	√	×	×	×
PCI9080	×	√	×	×	
Incoming Mailbox	×	√	×	×	×

表 5.2：接收器系統參數設定表

變數	預設值	位址	說明
slot_num	15	0x80000020	時槽數目
slot_length	2560	0x80000024	時槽長度
sf_I	32	0x80000028	DPDCH通道碼長度
sf_Q	256	0x8000002C	DPCCH通道碼長度
scramble_length	256	0x80000030	擾亂碼長度
data_length	80	0x80000034	DPDCH資料長度
pilot_length	10	0x80000038	DPCCH資料長度
ant_num	4	0x8000003C	天線個數
path_num	2	0x80000040	路徑數目
frame_num	10	0x80000044	總時框個數
mu	0.01	0x80000048	步進值
architecture	ADAPTIVE	0x8000004C	架構

表 5.3：TMS320C6201 資料型態列表

資料型態	大小 (位元)
<b>char</b>	8
<b>short</b>	16
<b>int</b>	32
<b>long</b>	40
<b>float</b>	32
<b>double</b>	64

表 5.4：模擬環境參數列表

時框格式	時槽	尋引	10 Symbols
		資料	80 symbols
	時框	10 ms	
傳送端	切片速率	3.84 Mcps	
	展頻係數	32 for data	
		256 for pilot	
		256 for scramble	
通道	萊利衰減	$f_d = 80$ Hz	
	多重路徑數	2	
接收端	架構	空-時智慧型犁耙接收機	

表 5.5：接收器程式記憶體配置表

	DSP0	DSP1
0x00000000-0x00000020	Interrupt service routines	
0x00000020-0x00010000	Program code	
0x02000000-0x02400000	1st Input/Output buffer	1st Output buffer
0x02400000-0x02800000	2nd Input/Output buffer	2nd Output buffer
0x03000000-0x03400000		Input buffer
0x80000000-0x80000020	Parameter table	
0x80000020-0x80000040	Mailbox-type Interrupt message	
0x80000040-0x80010000	Data	

表 5.7：接收器程式效能資料統計表(A) (使用參數: J = 4, users = 2, path = 2, SIR = -10 dB,  $E_b/N_0 = 12$  dB, frame = 1, 最佳化參數: -O3, -pm, -mt)

	Adaptive	Fixed
主程式時脈數	363,918	176,507
主程式執行時間 (秒)	0.0182	0.00825
解展頻時間 (秒)	0.4694	
程式碼長度	89,146 Kbytes	
參數長度	84 Bytes	
誤碼率 (BER)	-26.383 dB	-24.089 dB

表 5.8：接收器程式效能資料統計表(A) (使用參數:  $J = 4$ ,  $users = 2$ ,  $path = 2$ ,  $SIR = -10$  dB,  $E_b/N_0 = 12$  dB,  $frame = 1$ , 最佳化參數: -pm, -mt)

	<b>-O1</b>	<b>-O2</b>	<b>-O3</b>
主程式 (Adaptive) 執行時間 (秒)	0.01777	0.0182	0.0182
主程式 (Fixed) 執行時間 (秒)	0.008104	0.00825	0.00825
解展頻時間 (秒)	0.5586	0.4694	0.4694
I/O 函示時間 (秒)	0.6825	0.7866	0.7657
主程式 (可適性) 時脈數	355451	363919	363918
主程式 (固定式) 時脈數	162084	176489	176507
解展頻時脈數	11171021	9388901	9388923
I/O 函示時脈數	13649810	17635061	15314340
程式碼大小	83,926KB	89,146KB	89,146KB

## 第六章

### 結論

軟體無線電系統為未來通訊系統的一個主要的發展趨勢，彈性化的系統設計觀念與物件導向式的系統開發方法，提供了許多比起傳統單一功能系統更吸引人的特點。軟體無線電系統最早出現於 1990 年代初期，雖然它的概念簡單易懂，但實際上，一直到廿世紀結束，許多的技術仍未能配合軟體無線電的要求，如多頻帶射頻單元效能、類比-數位轉換器速度、中央處理器的執行時脈等。在此篇報告中，吾人嘗試整合軟體無線電之概念，以 DSP 處理器實現第三代行動通訊空-時犁耙接收機基頻端。為使系統可依通訊環境適時的調整架構，吾人結合智慧型陣列天線，並使用兩種不同之陣列天線處理架構。因軟體無線電具備可程式化與全數位化的等特點，結合智慧型天線系統為軟體無線電系統的一個主要技術。使用多重碼進接系統雖可以提高容量，但因所有使用者共用同一頻帶，訊號間干擾情況嚴重。應用波束形成技術可有效降低多重進接干擾，提高訊號雜訊比，提升系統效

能。但也因多根天線的使用，系統的設計複雜度與成本必將提高。可適性陣列天線系統本身即能動態調整對準訊號入射方向，當系統進入穩態，效能有顯著提升。吾人提出之扇型多工固定式波束的系統，使用傳統固定式天線，並搭配一軟體選擇器使其具備可程式化的特點。但受限於固定天線指向範圍，由執行結果顯示，其效能比適性天線來得差。但因其快速之執行速度與易於實作，提供另一種介於速度與效能之間的選擇。此兩種架構則可透過參數的選擇而任意切換，吾人可視情況隨時調整系統效能。

使用數位信號處理器開發系統，需考慮到程式的可實行性。此時程式的效能成為唯一指標。除了系統演算法架構考量外，如何達到即時處理的要求為系統實現之必要條件。因此，程式碼的最佳化與演算法的精簡，並配合處理器特性，為設計系統時不可忽略的課題。在訊號於真實世界由類比-數位轉換器進入到系統中後，所有的資料型態皆以自然型態之定點數儲存。設計上必須重新分析資料統計特性，以確保定點數運算法結果正確。處理定點數運算，除了量化誤差外，不當且大意的設計必定造成巨大的錯誤。吾人所建立之驗證環境為一純數位化環境，所有資料接由主電腦輸出並傳送至 DSP 處理器處理。由結果顯示，大量資料的輸出入對此系統並非問題，DSP 程式碼則仍受限於 DSP 處理速度，為系統之瓶頸。程式碼效能無法提升許多種

原因，如程式碼架構不佳，演算法過於費時等。但另一個重要因素來自於傳統程式開發方式-單緒的執行流程。對於需及時運算的系統來說，此架構將浪費大部分處理時間於等待狀態中，無法有效運用系統資源。此外，運算採用定點數運算雖可大幅增加速度，但因使用固定正規係數與固定長度儲存量化資料與演算結果，不易控制動態範圍大之資料，對於量化誤差之影響無法完全排除。但系統改進之空間仍大，如：

#### **硬體方面：**

吾人可配合實際智慧型天線輸出入介面與高速類比-數位轉換器之使用以處理真實訊號。並針對程式瓶頸處如解展頻運算提供更高速的專屬解碼、解展頻運算晶片，使 DSP 處理器能專職於執行演算架構。

#### **軟體方面：**

搭配即時作業系統(Real-Time OS)之使用，輔助吾人正確管理 DSP 系統程式之排程、資料輸出入、分配記憶體與 DSP 處理器可用資源，減少系統閒置時間(idle time)，真正發揮 DSP 處理器效能。定點數運算需配合資料特性採用動態正規係數，並改進演算架構，以使系統能處理更複雜之訊號。

## 參考文獻

- [1] J. Mitola, "The software radio architecture," *IEEE Commun. Mag.*, pp. 26-38, May 1995.
- [2] T. Turetti and D. Tennenhouse, "Estimating the computational of software GSM base station," *Proc. IEEE ICC'97*, Montreal, Canada, June 1997.
- [3] J. Kennedy and M. C. Sullivan, "Direction finding and smart antennas using software radio architectures," *IEEE Commun. Mag.*, pp. 62-68, May 1995.
- [4] W. Tuttlebee, "The impact of software radio," presented at the Software Radio Workshop, Brussels, Belgium, May 1997.
- [5] V. Bose, M. Ismert, M. Welborn, and J. Guttag, "Virtual radios," *IEEE J. Select. Areas Commun*, vol. 17, no. 4, pp. 591-602, April 1999.
- [6] V. G. Bose, A. B. Shah, and M. Ismert, "Software radio for wireless networking," *IEEE Infocom'98*, San Francisco, CA, April 1998.
- [7] J. Mitola, **Software radio architecture**, A Wiley-Interscience Publication, 2000.
- [8] J. C. Liberti, Jr. and T. S. Rappaport, **Smart Antenna for Wireless Communications: IS-95 and Third Generation CDMA Application**, Prentice-Hall, 1999.
- [9] Special Issue, IMT-2000: Standards Efforts of the ITU, *IEEE Pers. Commun.*, vol. 4, Aug. 1997.



- [10] A. J. Viterbi, **DS-CDMA, Principles of Spread Spectrum Communications**, Addison-Wesley, 1995.
- [11] A. Duel-Hallen, J. Holtzman, and Z. Zvonar., "Multiuser detection for CDMA systems," *IEEE Personal Communications*, pp.46-58, April 1995.
- [12] R. Lupas and S. Verdu., "Near-far resistance of multiuser detectors in asynchronous channels", *IEEE Trans. Commun.*, vol.38, no.4, pp.496-508, April 1990.
- [13] F. Adachi *et al.*, "Coherent multi-code DS-CDMA mobile radio access," *IEICE Trans. Commun.*, vol. E79-B, Sept. 1996, pp. 1316-25.
- [14] U.C. Fiebig *et al.*, "Design study for a CDMA-based third generation mobile radio system," *IEEE JSAC*, vol. SAC-12, May 1994, pp. 733-43.
- [15] 3rd Generation Partnership Project(3GPP), *Technical Specification*, TSS1.11, TS S1.13.
- [16] Kohno, "Structures and theoris of software antennas for software defined radio," *IEICE Trans. Commun.*, vol.E83-B, no.6, pp.1189-1199, June 2000.
- [17] H. Haradaa, Y. Kamio and M. Fujise, "Multimode software radio system by parameter controlled and telecommunication component block embedded digital signal processing hardware," *IEICE Trans. Commun.*, vol.E83-B, no.6, pp. 1217-1236, June 2000.
- [18] Y. Karasawa, Y. Kamiya, T. Inoue and S. Denno, "Algorithm diversity in a software antenna," *IEICE Trans. Commun.*, vol.E83-B, no.6, pp. 1229-1246, June 2000.

- [19] Texas Instruments Corp. (www.ti.com), 2000.
- [20] Xilinx Corp. (www.xilinx.com), 2000.
- [21] **TMS320C6201, TMS320C6201B Digital Signal Processors Data Sheet**, sprs051, Texas Instruments, March 2000.
- [22] **TMS320C6000 Peripherals Reference Guide**, spru190d, Texas Instruments, March 2000.
- [23] **TMS320C6000 Programmer's Guide**, spru198d, Texas Instruments, March 2000.
- [24] **TMS320C6000 Optimizing C Compiler User's Guide**, spru187g, Texas Instruments, March 2000.
- [25] **TMS320C6000 Assembly Language Tools User's Guide**, spru186g, Texas Instruments, March 2000.
- [26] **TMS320C6x C Source Debugger User's Guide**, spru186g, Texas Instruments, March 2000.
- [27] **TMS320C62x DSP Library Programmer's Reference**, spru402e, Texas Instruments, March 2000.
- [28] S. Anjanaiah, "TMS320C6000 McBSP Initialization," Application Report, spra488, Nov. 1998.
- [29] E. Biscondi, "C Implementation of the TMS320C62xx Intrinsic Operators," Application Report, spra616, Nov. 1998.
- [30] "Implementation of a W-CDMA Rake Receiver on a TMS320C62x DSP Device," Application Report, spra680, July 2000.