國立中正大學

電機工程研究所

碩士論文

免授權個人進接通信系統第二層協定之處理程序之硬體智慧財產權(IP)設計

Hardware IP Design for PACS_UB Layer II Protocol Processing

研究生　：雷智偉

指導教授：葉經緯　博士

中華民國八十八年七月

目錄

第一章　前言

1.1　研究動機

1.2　論文概觀

第二章　免授權個人進接行動通信系統（PACS_UB）通信協定

2.1　個人行動通信系統（PCS_UB）簡介

2.2　PACS_UB 系統及架構

2.3　PACS_UB 各層連接介面

2.4　免授權個人進接行動通信系統第二層（L2）通信協定主要功能

2.4.1　系統廣播信號（SBC）

2.4.2　初始進接信號（IA）

2.4.3　訊框同步（Frame Sync）

2.5　免授權個人進接行動通信系統架構

2.5.1　手機（SU）端基本架構

2.5.2　基地台（RP）端基本架構

第三章　基本設計概念

3.1　通信協定模組不同設計方法

3.2　第二層通信協定系統方塊圖

3.2.1　標頭處理單元（HPE）

3.2.2　同步（SYNC）

3.2.3　串列、並列轉換（S2P_P2S）

第四章　軟、硬體通信協定設計之比較

4.1　硬體設計之優點

4.2　各模組硬體設計與軟體設計之比較

4.2.1 標頭處理單元（HPE）

4.2.2 同步（SYNC）

4.2.3 串列、並列轉換（S2P_P2S）

4.3　本章結論

第五章　硬體通信協定設計於智慧財產權（IP）之運用

5.1　智慧財產權（IP）簡介

5.2　智慧財產權（IP）於第二層通信協定之運用時機

5.2.1　標頭處理單元（HPE）

5.2.2　串列、並列轉換（S2P_P2S）

5.2.3　同步（SYNC）

5.3 本章結論

第六章 效益評估與測試方式

6.1 第二層（L2）通信協定晶片效能

6.2 晶片測試與驗證

第七章 結論及未來展望

附錄及參考文獻

第一章　前言

1.1 研究動機

低階個人通信系統是目前最受廣泛注目的通信系統之一，像是 DECT、CT2、PHS....等以低移動率、小細胞元為訴求的通信系統將成為都會區域通信的新寵，然而也因為如此，新的低階個人通信系統的通信協定也不斷孕育而生，於是產生了各個通信協定之間互相通聯時通信協定間不同無法有效整合的問題，所以通信協定引擎（Protocol Engine）的建立便成為可以解決不同通信協定之間的通信問題的一種解決方法。

通信協定引擎（Protocol Engine）的基本概念就是建立一個基本的電路或工作平台（Platform）給不同的通信系統使用，當運用於不同的通信協定時，僅需更改他的軟體，而不需再更改硬體設計，如此將使通信系統在不同的通信協定之間的變換變得更為簡便。

為了建立一個能使更多通信協定可以使用的平台鋪路，在這一篇論文中將先嘗試建立一些硬體電路，而這些電路是提供通信協定引擎（Protocol Engine）必需的電路；然而通信協定實在是太多了我們無法逐一了解，於是我們選擇以免授權個人進接行動通信系統第二層（PACS_UB Layer 2）通信協定出發，然後以第三代行動通信系統為目標，找出適合其使用的硬體。

1.2 論文概觀

本篇論文分為以下數章：

第一章，提出本論文的研究方向及論文大綱。

第二章，介紹免授權個人進接行動通信系統（PACS_UB）通信協定之整體環境，同時介紹第二層（Layer 2 , L2）通信協定之主要功能，最後簡介手機（SU）以及基地台（RP）的基本架構。

第三章，實現第二層（Layer 2 , L2）通信協定之主要功能的硬體方塊圖及架構。

第四章，於通信協定設計時，使用硬體或軟體設計時之優缺點比較。

第五章，使用硬體設計通信協定設計時，其重複使用之方式以及在硬體智慧財產權（Hardware IP）之運用。

第六章，效益評估與測試方式。

第七章，結論及未來展望。

最後，參考文獻及附錄。

第二章　免授權個人進接行動通信系統（PACS_UB）通信協定

個人通信服務（Personal Communication Service,PCS）之應用範圍相當廣泛，所以在其應用範圍內就有相當多的通信協定被制定。其中較常見的有 CT2、DECT、PHS，而個人進接行動通信系統（Personal Access Communication System，PACS）也就是其中的一種，而 PACS 系統已於 1996 年成為北美通信服務的標準，然而由於使用頻率及多工方式的不同，PACS 又區分為三種不同的通信協定分別為：個人進接行動通信系統（PACS Licensed）、免授權個人進接行動通信系統（PACS_UA）、免授權個人進接行動通信系統（PACS_UB），所以本篇論文就是以完成 PACS_UB 通信協定為主。

2.1　免授權個人進接行動通信系統（PACS_UB）通信協定簡述

PACS_UB 之主要特色簡述如下：

（一）接收靈敏度佳：PACS_UB 接收機靈敏度極佳（-101dBm），因此細胞涵蓋半徑較大，相同服務區面積所需之基地台數目約僅為 PHS 系統（靈敏度-90dBm）與 DECT 系統（靈敏度-83dBm）之 1/3 ~ 1/9。

（二）呼叫容量高：使用專屬的系統廣播通道（System Broadcast Channel， SBC），此設計使 PACS_UB 系統的呼叫容量遠較其他類似系統為高，同時也可縮短用戶接受來話的時間。具有可支援２０萬用戶之警訊/登錄區域（Alerting／Registration Area；ARA），其相較於其他的低階行動通信系統，可大幅降低登錄的話務。此外，PACS_UB 系統也提供緊急呼叫功能。

（三）具功率控制功能：PACS_UB 系統手機具備功率控制功能，可讓手機在遠離基地台或受到障礙物遮蔽時，能適當調高其訊號強度以確保無線鏈路的品質，此技術亦直接降低手機的功率損耗並延長持續通話時間。

（四）設備成本低：具有較短的時框及較低的通道傳送速率，使其之去回延遲（Round-trip Delay）短及頻選衰落（Frequency-selective Fading）效應低，這使得系統單元如手機、基地台與基地台控制站可免去諸如回音消除器、等化器及錯誤改正等複雜昂貴的電路，因而降低整體成本，並因具有較短的時框，能在連結出錯時，以較短的時間即可恢復通話品質。

（五）基地台間不需同步：上下鏈（Up／Down Link）間採 TDD 雙工方式，基地台之間不需同步，系統建設成本較低。

（六）通訊不中斷：通信鏈路品質惡化時，手機可及時切換至其他時槽或轉換基地台而不會造成通話中斷（切換時間為 0.05 秒）。

（七）自動頻率指配：PACS_UB 系統在基地台頻率指配上使用半固定方式（稱為 QSAFA），基地台可在短時間內自動完成頻率掃瞄、量測及選定等工作，免去人工指配頻率的困擾。

（八）用戶移動速率高：用戶移動速率可達每小時九十公里以上（PHS 與 DECT 系統各約為４０公里與４公里）。

（九）使用智慧型網路：使用智慧型網路架構，適合未來提供各種個人通信服務。

（十）提供語音/數據整合服務：PACS_UB 系統已完成數據通信協定設計，未來能夠提供語音和數據的整合通信，符合未來通訊發展趨勢。

2.2 PACS_UB 系統及架構

PACS_UB 系統架構及基本的功能描述於圖 2_1。在這個系統架構下包含了：固接用戶端（Fixed Subscriber Units，Fixed SU）、可移動用戶端（Portable Subscriber Units，Portable SU）、射頻端（Radio Ports，RP ）、射頻控制端（Radio Port Control Unit，RPCU）、使用用戶管理（Access Manager，AM）、網路交換功能

（Public Switched Telephone Network），而 RPCU、AM 及一些網路交換功能可以適當的整合在一個單元內。

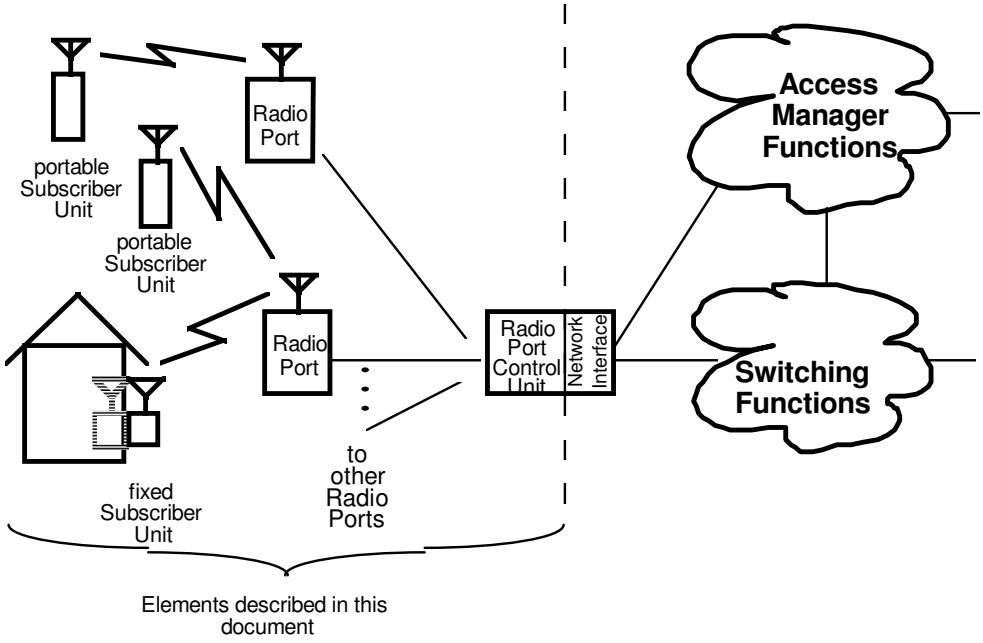

Figure 2_1 Function representation of PACS-UB system architecture

各單元的基本功能描述如下：

（一）固接用戶端（Fixed Subscriber Units，Fixed SU）：多為一般家用的固接式電話。

（二）可移動用戶端（Portable Subscriber Units，Portable SU）：也就是一般我們稱的手機，而（一）、（二）項就構成了 PACS_UB 最基本的用戶。

（三）射頻端（Radio Ports，RP ）：也就是一般所謂的基地台。

（四）射頻控制端（Radio Port Control Unit，RPCU）：數個基地台受一個 RPCU 控制，至於受控 RP 的數量則依建立細胞元的

大小決定。

（五）使用用戶管理（Access Manager，AM）：管理各用戶端進入網路的權利，特別是未經允許的使用者，像是未繳電話費的使用者。

（六）網路交換功能（Public Switched Telephone Network）：完成本網路與其他網路的連接，特別是在執行 ALT（Automatic Link Transfer）時使用。

2.3 PACS_UB 各層連接介面

在圖 2_2 中主要定義了 PACS_UB 通信協定中各層的連接介面，它共有三層：分別是 Layer1（L1）、Layer2（L2）、Layer3（L3）。Layer1 主要是實體層（Physical Layer）以及空中介面（Air Interface）的控制，Layer2 主要是資料連結層以及溝通模式（Acknowledge mode）的建立，Layer3 主要針對資料連結層和網路層以及通話流程（Call Flow）的控制。

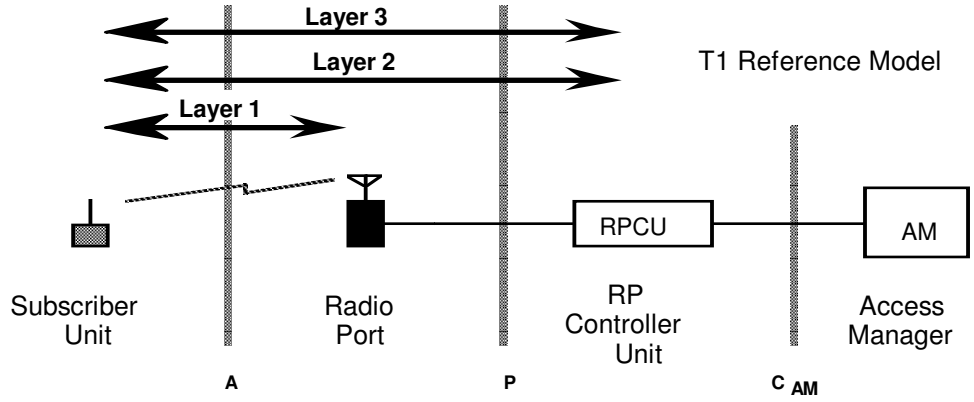

Figure 2_2　PACS-UB architecture and signaling layers

2.4　免授權個人進接行動通信系統第二層（L2）通信協定主要功能

第二層(L2)通信協定主要是維持手機端(SU)與基地台(RP)間的通連正常，縱使手機端（SU）不需通信也必需維持與基地台間的同步，所以手機端（SU）平時就是依靠廣播信號（SBC）來維持與基地台（RP）間的同步。以下圖 2_3 是以手機端（SU）為觀點的四個狀態，描述手機端（SU）如何維持與基地台間的通連。

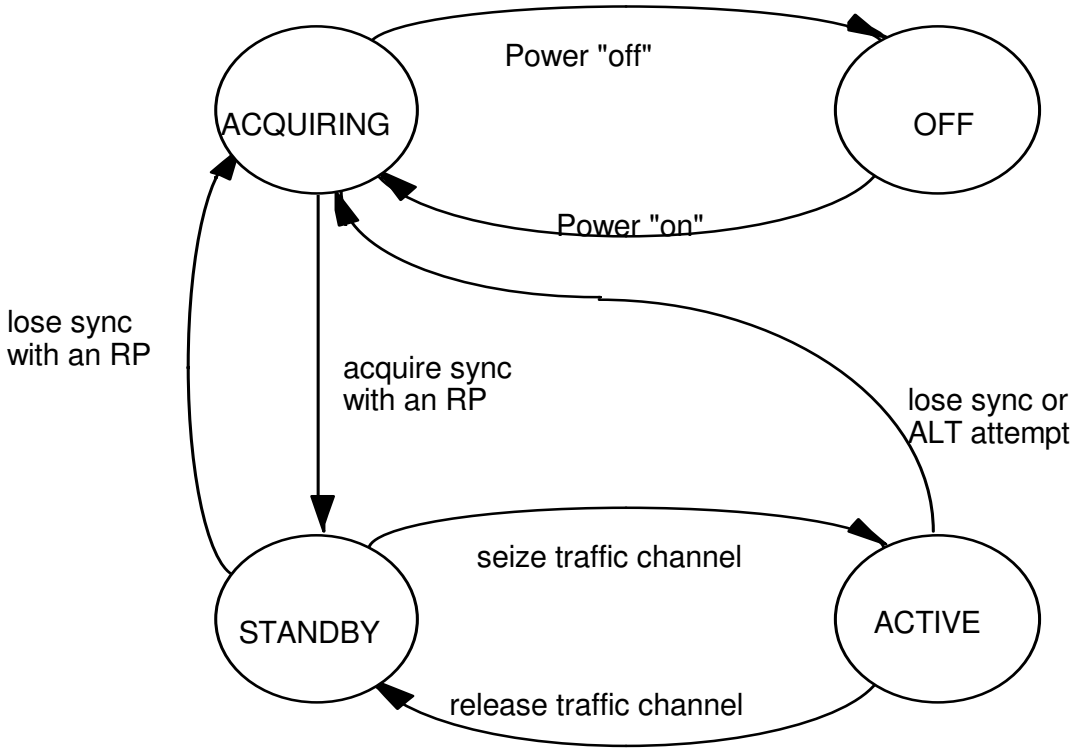

Figure 2_3　Layer 2 states in the SU

接著以下各節，要介紹的是第二層（L2）通信協定中最主要的功能，分別是：系統廣播信號（System Broadcast Channel，SBC）

、初始進接信號（Initial Access，IA）

、訊框同步信號（Frame Synchronization，Frame Sync）。

2.4.1 系統廣播信號（System Broadcast Channel，SBC）

系統廣播信號（System Broadcast Channel，SBC）主要支援以下三項功能：

（一）維持手機端（SU）與基地台（RP）間的同步。

（二）使手機（SU）有能力確認與其相通的基地台（RP）。

（三）並獲得足夠、正確且適當的訊息。

另外系統廣播信號（System Broadcast Channel，SBC）還區分為三個邏輯頻道（Logical Channel）分別是：

（一）振鈴頻道（Alert Channel，AC）：當一台手機（SU）要打電話時或對方打電話來時使用。

（二）系統資料頻道（System Information Channel，SIC）：接收RPCU 端或網路端的資料參數。

（三）進接要求頻道（Access Request Channel，ARC）：當手機端（SU）要求通話或基地台（RP）允許通話時使用。

2.4.2 初始進接信號（Initial Access，IA）

初始進接信號（Initial Access，IA）主要支援以下二項功能：

（一）提供手機端（SU）要求一個時槽（Time Slot）以提供通話時使用。

（二）基地台（RP）允許或不允許手機端（SU）使用該時槽（Time Slot）來通話。

2.4.3 訊框同步信號

（Frame Synchronization，Frame Sync）

主要是用來維持手機端（SU）與基地台（RP）間保持正確的同步、同時不能同步時通知其他單元做適當的處理。

2.5 PACS_UB 手機、基地台之基本架構

2.5.1 手機端（SU）之基本架構

下圖 2_4 是 PACS_UB 通信協定手機端（SU）之基本架構。而本篇論文主要要完成的就是手機端（SU）的第二層（L2）通信協定，也就是在下圖控制單元（Controller）的部分。

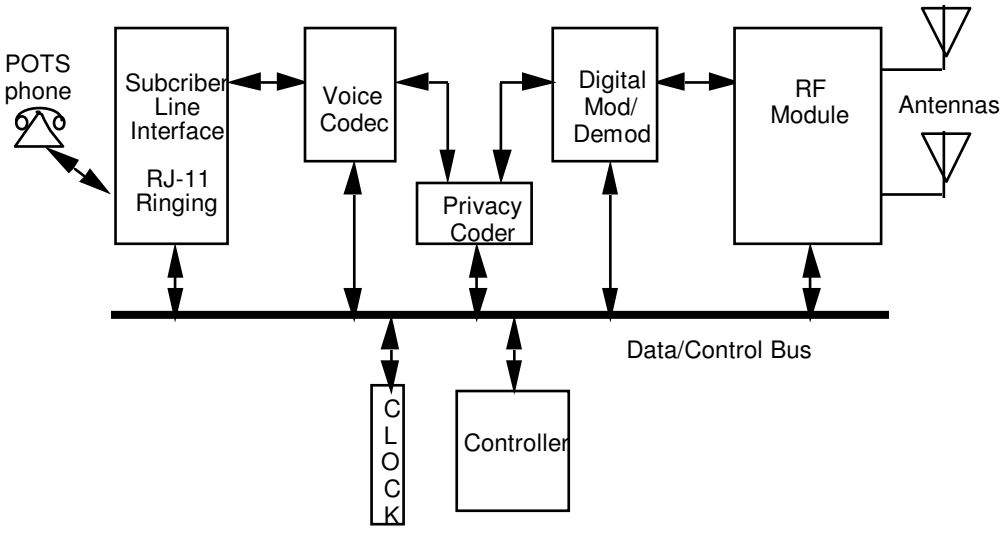

Figure 2_4　Subscribe Unit(SU) block diagram

2.5.2 基地台端（RP）之基本架構

圖 2_5 是 PACS_UB 通信協定基地台端（RP）之基本架構。因本篇論文不涉及 RP 端，所以不多做介紹。

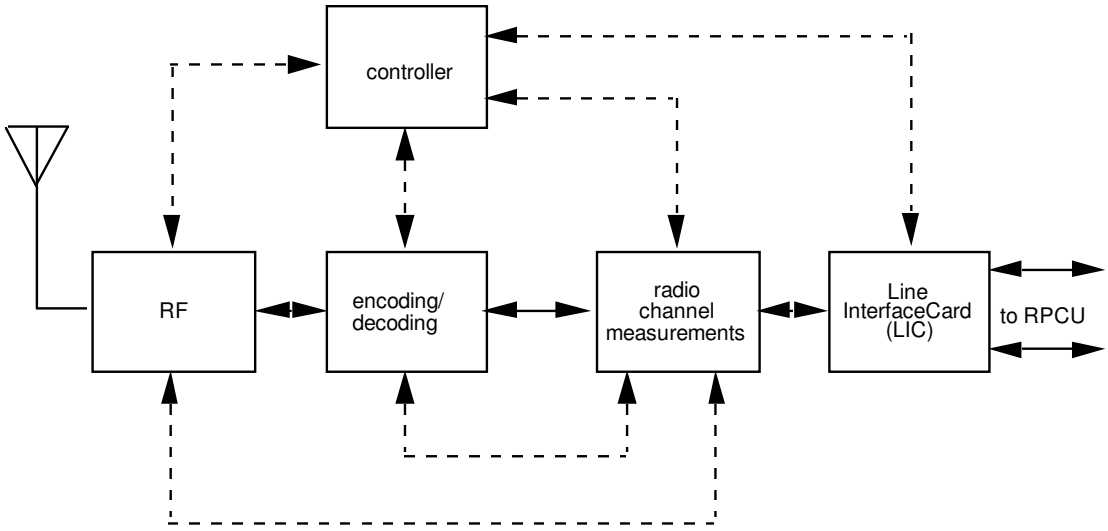

Figure 2_5 Radio Port (RP) block diagram

第三章　基本設計概念

3.1 通信協定模組不同設計方法

在一般的通信協定模組設計中，大多採用單晶片並運用軟體控制的方式，也就是先選定一個 CPU 作為工作平台，像是 8051 單晶片，然後將通信協定用組合語言完成，放入 8051 單晶片中，如此便完成一完整的通信協定控制。

但用上述的方法來完成可能沒有效率也浪費功率，因為畢竟 8051 單晶片不是專門針對通信協定模組設計的，所以有很多電路是多餘的，於是我們就想為通信協定模組量身訂作一個電路，並可以應用於不同的通信協定中，圖 3_1 中就是這樣的設計理念，我們先把適合通信協定的電路製作於一個模組（L2）中，然後選擇一個適當的 CPU 作為處理器，再將之作有效的整合。

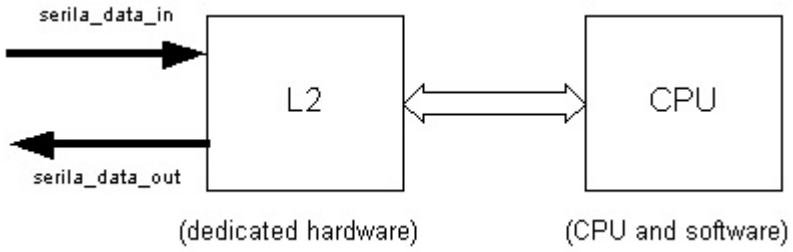

Figure　3_1　Protocol System Block Diagram：
Dedicated Hardware，CPU and Software

當我們把重複出現較多次並適合的功能製作在 L2 模組後，剩下的工作就是把未完成的功能再以軟體的方式放入 CPU 中，並完成完整的第二層通信協定（Layer2 Protocol）。

接下來的問題就是到底哪些電路要做到硬體中？哪些要用軟體來完成？這個問題我們在第四章另詳細分析，為使讀者更易了解分析內容，以下各節將先介紹 L2 模組的內容。

3.2 第二層通信協定系統方塊圖

下圖 3_2 是 L2 模組的系統架構圖（Architecture），在這個

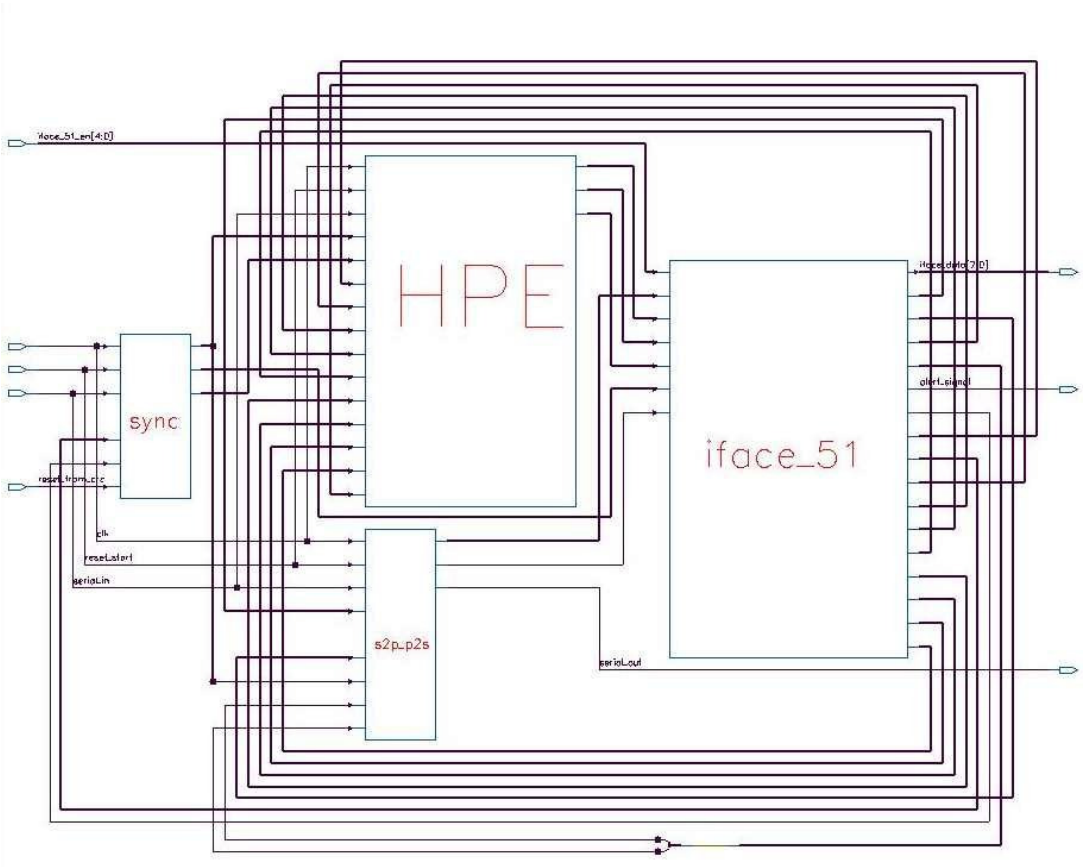

Figure 3_2　L2 Block Diagram

模組中，共有四個部分（Sub-module），分別是：

（一）標頭處理單元（Header Processing Element，HPE）：專門處理封包（Package）中的標頭（Header）部分。

（二）同步（Synchronization，Sync）：專門處理與基地台間（RP）同步的問題，同時提供整個系統的時序（Timing）。

（三）串列、並列轉換（Serial to Parallel、Parallel to Serial，S2P_P2S）：將資料作串、並列的轉換。

（四）8051 介面（Interface_51，Iface_51）：提供一個介面作為 L2 與 8051 間溝通時使用。

3.2.1 標頭處理單元

（Header Processing Element, HPE）

下圖 3_3 是 HPE 模組的系統架構圖（Architecture），在這個模組中，共有四個部分（Sub-module），分別是：

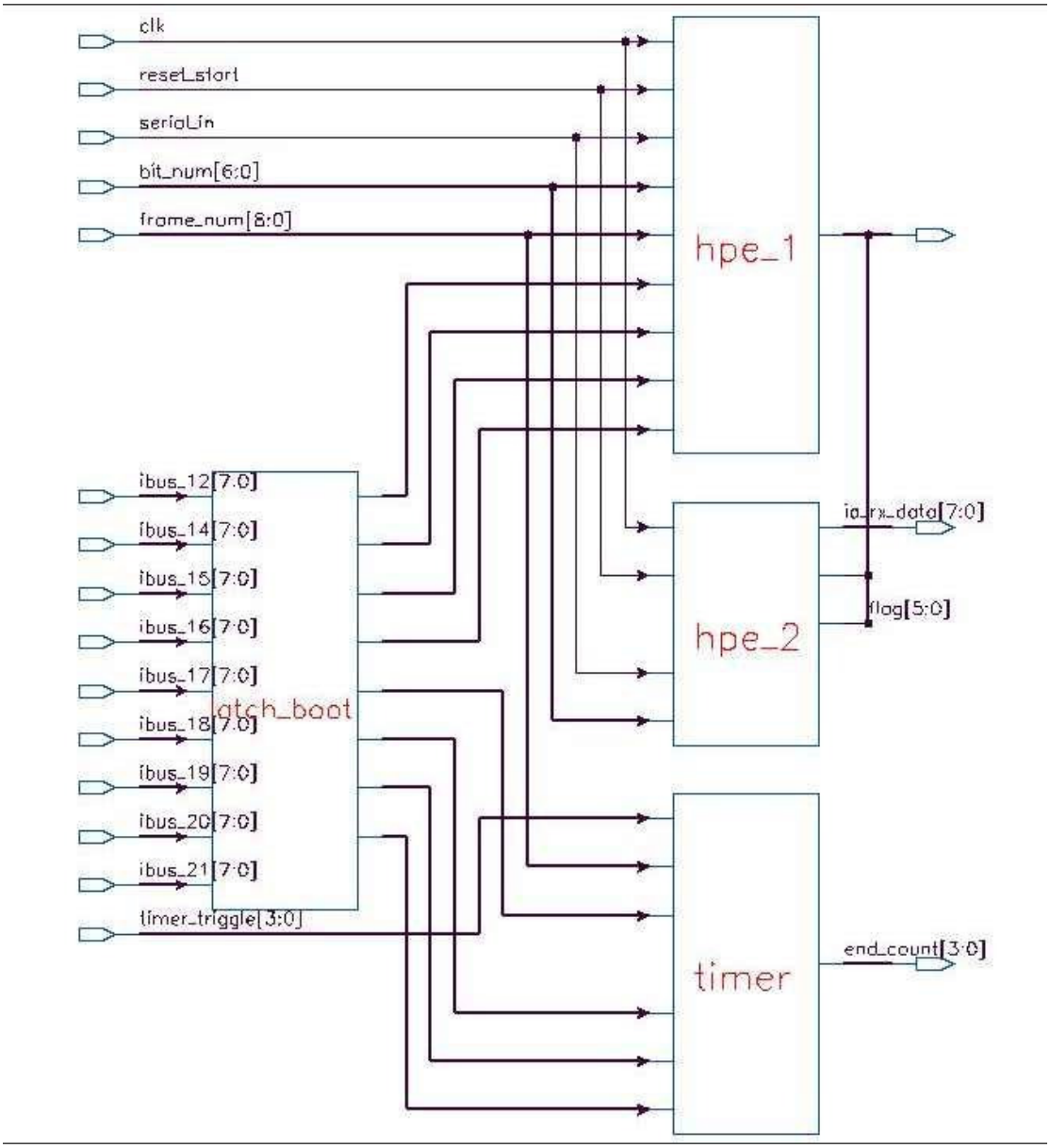

Figure 3_3　Header Processing Element（HPE）Architecture

（一）標頭處理單元之一（Header Processing Element_1，

HPE_1)：專門處理封包（Package）中的系統廣播信號（System Broadcast Channel）的標頭（Header）部分，而系統廣播信號（System Broadcast Channel）共分為四種標頭（Header）需要處理，有 SBC Header、AC Header、SIC Header、SIC Message Header。

（二）標頭處理單元之二（Header Processing Element_2，HPE_2)：專門處理封包（Package）中的初始信號（Initial Access)，而初始信號（Initial Access）共分為二種標頭（Header）需要處理，有 Access Confirm、Access Deny。

（三）計時器(Timer)：在無線通信中(不僅只有 PACS_UB Layer2）都需要很多的計時功能，因為當一組信號被送出後，發信端不可能無限制的等待，必須有一定的時間限制，所以當信號等待一段時間能無回應後，系統就由計時器（Timer）自動中斷，至於等待的時間長短就由 RPCU 端控制。在本模組中共提供 4 組 8 位元計時器（Timer)，供各層（Layer 1~3）使用。

（四）起始暫存器（Boot Latch，Latch_boot)：本暫存器共有 9 組 8 bits Latch，分別為提供 HPE_1、HPE_2、Timer 等模組中標頭（Header）值及計時值之存放處，若值有需要變動也是由此處修改。

3.2.2 同步（Synchronization, Sync）

下圖 3_4 是 SYNC 模組的系統架構圖（Architecture），在這個模組中，共有二個部分（Sub-module），分別是：

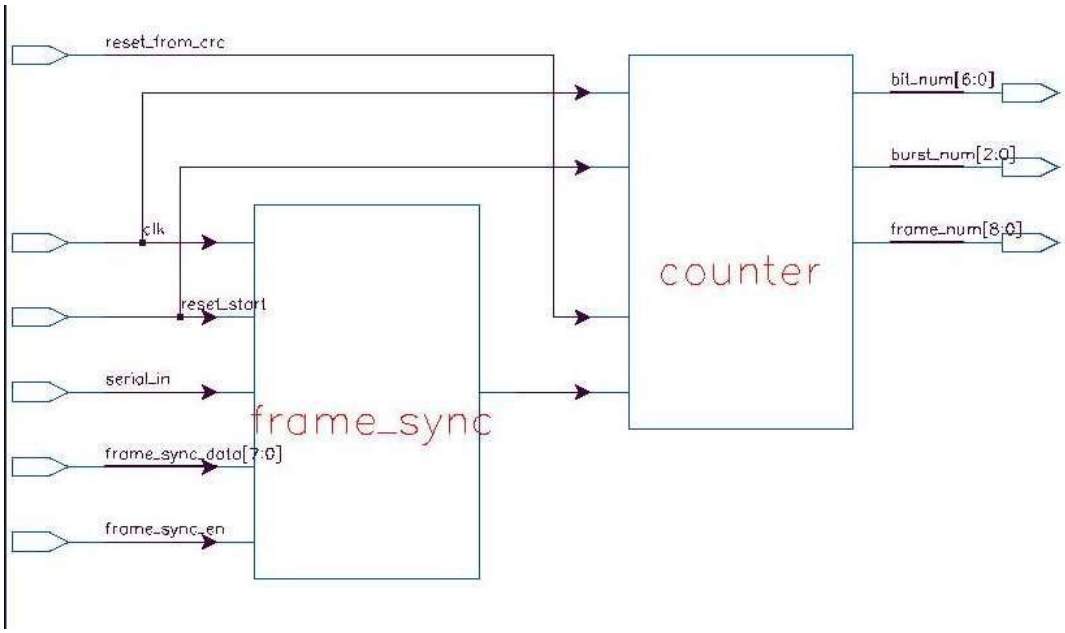

Figure 3_4　Synchronization（Sync）Architecture

（一）訊框同步處理（Frame Synchronization，Frame_Sync）：本模組共可處理 8 位元的串列資料，作為訊框同步（Frame Synchronization）使用。

（二）計數器（Counter）：本計數器（Counter）為整個系統之中心，所有各模組都需要由它提供時序。其共有三組計數器（Counter）分別為位元計數器 bit_counter（7 bits）、時槽計數器 burst_counter（3 bits）、訊框計數器 frame_counter（9 bits）。

3.2.3 串列、並列轉換（S2P_P2S）

下圖 3_5 是 S2P_P2S 模組的系統架構圖（Architecture），在這個模組中，共有二個部分（Sub-module），分別是：

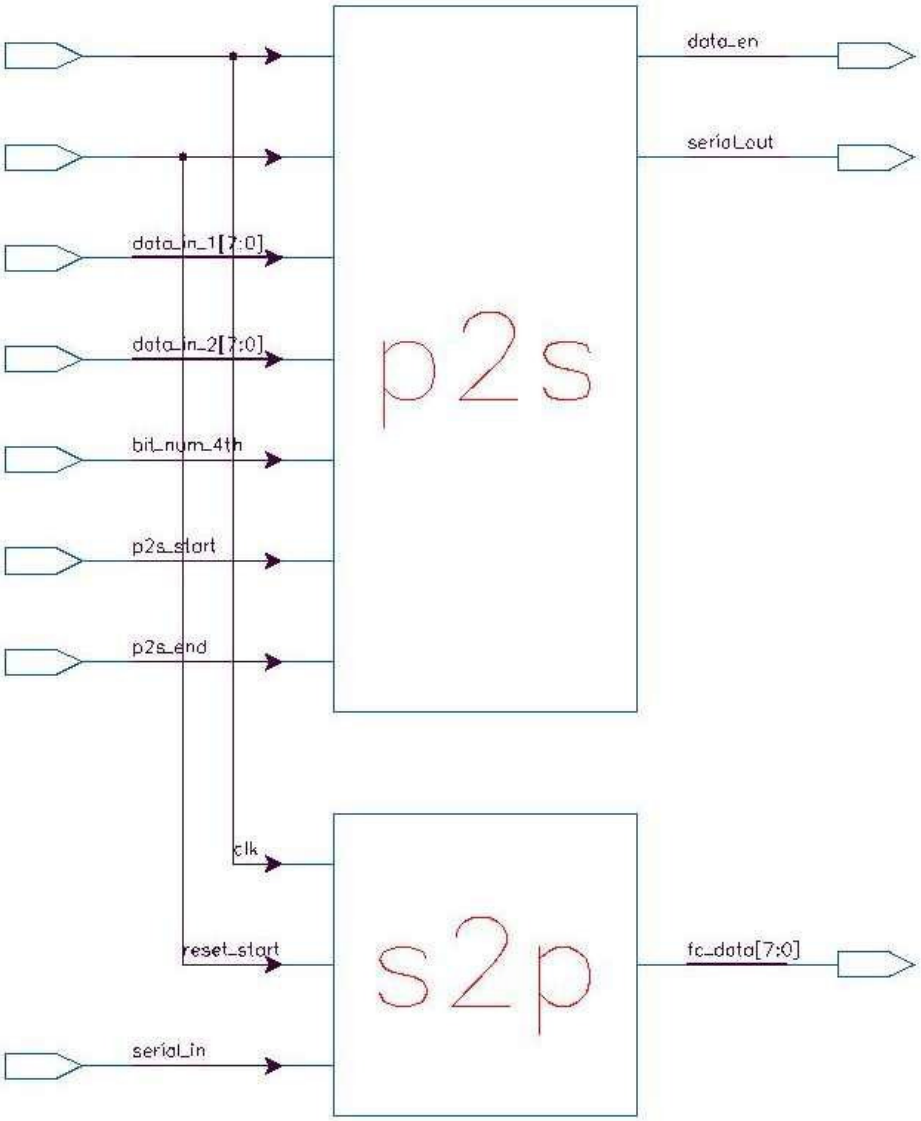

Figure 3_5　Serial to Parallel，Parallel to Serial（S2P_P2S）Architecture

（一）串列對並列轉換（Serial to Parallel，S2P）：將串列資料轉換為 8 位元並列資料。

（二）並列對串列轉換（Parallel to Serial，P2S）：將 8 位元並列資料轉換為串列資料。

第四章 軟、硬體通信協定設計之比較

4.1 硬體設計之優點

在前一章中已經介紹了使用軟體、硬體設計之比較，也強調了本篇論文要添加硬體來設計通信協定系統，但其中究竟有什麼好處呢？大致說來最主要的優點有兩點，一是可以有更低的功率，一是更有效率，也就是用更快的速度來處理通信協定系統。

首先是低功率的部分，在現在的通信系統中，我們都希望製作出來的系統是低功率的，特別是手機的部分。若單純只用 8051 和軟體來作通信協定系統，那大部分的功率都將消耗在 8051 這個大晶片上，我們的想法是用一塊面積較小，消耗功率也較小的硬體（L2）來處理通信協定系統，如此一來可以有效降低功率消耗。

接著是效率的部分，如果在第二層（L2）通信協定系統中能夠用較快的方式來處理，那麼我們就有較多的時間來處理其他剩餘各層的通信協定，如此一來即可提昇通信協定系統的效率；另外在同步技術上，如果用軟來處理勢必有時序的問題，也就是必需設計軟體計數器，但若用硬體來處理就較無此問題，同時也較容易控制。

4.2 各模組硬體設計與軟體設計之比較

接下來的各節就是要比較運用硬體設計比運用軟體設計的好處，我們將按照標頭處理單元（HPE）、同步（SYNC）、串並列轉換（S2P_P2S）等各模組（Sub-module）的順序一一比較。

4.2.1 標頭處理單元（Header Processing Element，HPE）

標頭處理單元（Header Processing Element，HPE），主要是處理封包（Package）中的標頭（Header）部分，而在通信協定系統中每個封包（Package）都有其固定的標頭（Header），所以我們只要抓住正確的時序就可以處理標頭（Header）的問題。

從上一段的敘述看來，處理這種問題似乎也可以用軟體來處理，那為何又要大費周章的使用硬體來處理呢？以下我們先用一個例子來作比較，看完這個例子我們就可以比較出使用硬體設計的好處了。

下面的例子是在某一個封包（Package）中，有一 6 位元的標頭（Header）需要處理，我們分別使用軟體及硬體來完成，軟體的部分是採用 8051 作為平台並 8051 組合語言來完成，使用 25MHz 作為工作頻率；而硬體的部分就是用 L2 模組（module）中的標頭處理單元（Header Processing Element，HPE）來處理，使用 384KHz

作為工作頻率，這個工作頻率是規格表（Specification）所規定的。

（1）運用軟體設計的例子：使用 8051 組合語言。

我們先假設封包（Package）中標頭（Header）的值已由 8051 的串列阜（Port）輸入處理完畢並以存入記憶體中，然後由記憶體中取出處理，程式如下所示：

```
MOV     A,(mem_1)
AND     A,(001111)    // get 6 bits header data
CMP     A,(mem_2)     // compare with original header data
JPZ     A             // jump to process contend, if get correct
                      // header.
```

在上述程式中共使用了 4 個指令，以 8051 而言每一指令共需要 2 個機械週期（Machine Cycle），每個機械週期（Machine Cycle）需要 12 個脈衝（Clock Cycle），若是以 25MHz 為工作週期，那麼執行完這 4 個指令共需要 3.84 微秒（uS），而若 8051 以平均功率 41mW 來計算的話，執行完這 4 個指令共需消耗功率 157.44 mW；接下來我們來看運用硬體設計的例子。

（2）運用硬體設計的例子：

硬體的部分主要是採用 L2 模組（module）中的標頭處理單元（Header Processing Element，HPE）來處理，使用 384KHz 作為工作頻率，而對於標頭處理（Header Processing）我們不需要再由

8051 再作多餘的處理，8051 只需要將標頭處理單元（Header Processing Element，HPE）處理完結果讀入即可，其硬體架構如下：

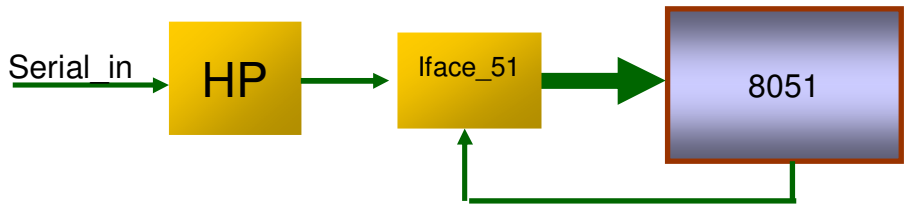

Figure 4-1 HPE sub-module implemented by dedicated hardware

最後我們針對效率及功率作一比較，若使用軟體來處理的話共需耗費時間 3.84uS，消耗功率 157.44mW，若使用硬體來處理的話共需耗費時間 2.6uS，消耗功率 36.4mW，所以很明顯的不論是在效率或是在功率上硬體都比軟體好，詳見下表。

	Software	Dedicated hardware
Efficiency	2 machine cycle/inst. =24 clock cycle/inst. =960ns(25 MHz) total=3.84us	total=2.6us (save CPU time for L3)
Power	Ave. power=41mW/ns Total power=157.44mW	Ave. power = 14mW/ns total=36.4mW

Table 4-2 Comparison dedicated hardware with software design

4.2.2 同步（Synchronization，Sync）

本模組（module）專門處理與基地台間（RP）同步的問題，同時提供整個系統的時序（Timing），針對此一部份的優缺點比較，我們將區分為兩個部分來作比較：

（一）正確性：如果我們以軟體計數器來並使用 8051 來處理的話，將有兩種方法，一是將 8051 的工作頻率降低至 384KHz 以符合規格（Specification），二是能維持 8051 的工作頻率為 25MHz，然後使用軟體計數器運作，但使用以上兩種方法均有其缺點。如果將 8051 的工作頻率降低至 384KHz 的話，雖然可以抓住正確的時序（Timing），但整個 8051 的處理速度明顯的降慢，根本無法處理複雜的通信協定；如果將 8051 的工作頻率訂為 25MHz，雖然可保有較高的工作速率，但是得耗費較多的工作時間在軟體計數器上，而且消耗較多的功率在執行軟體計數器上，同時使用 25MHz 的工作頻率要用軟體調出 384KHz 的時脈也不是一件容易的事。所以最好的方法能是使用硬體來處理，我們先設計一塊硬體電路就是同步（Synchronization，Sync）然後設定 384KHz 的工作頻

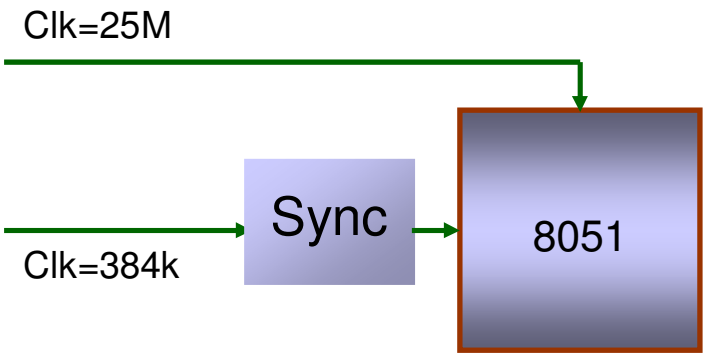

Figure 4_3　Sync sub-module implemented by dedicated hardware

率，而 8051 則維持 25MHz 的工作頻率，如此一來同步

（Synchronization，Sync）模組只要將結果也就是正確的時序（Clocking）送給 8051 即可，如此一來就可以得到正確的時序（Clocking）又不會降低 8051 的工作速率了。硬體架構如圖 4_3 所示。

（二）可移植性：

如果用軟體計數器來作為時脈的控制，那在不同的 CPU 系統中我們就必須重新調整軟體計數器的延長時間（Delay Tme），每移植至一個新的 CPU 系統我們就得重新調整一次，如此的調整動作將會非常的繁瑣，也不實際，但若我們以硬體來完成，不管我們移植至任何的 CPU 系統中，同步（Synchronization，Sync）模組仍使用 384KHz 的工作頻率，而新的 CPU 系統就使用它自己的工作頻率，如此設計在移植中也多了許多便利性。

4.2.3 串列、並列轉換（Serial to Parallel、Parallel to Serial，S2P_P2S）

在一般的 CPU 系統中我們需要串列、並列轉換（Serial to Parallel、Parallel to Serial，S2P_P2S）模組，因為如果不使用此模組，而使用 CPU 系統中的資料匯流排（Data Bus）16 或 32 位元那

樣大的匯流排來控制 1 位元的串列資料，實在是太浪費了，所以我們需要這個模組的設立；但如果我們用 8051 來處理而不用 S2P_P2S 模組的話也是會有問題，8051 單晶片的 UART 模組是用來控制串、並列轉換的一個單元，它為了兼具一般性的使用者使用將其鮑率（Baud rate）訂為 8 bps、16 bps、32 bps 等固定常用的工作速率，也就是說使用者只有這幾種鮑率(Baud rate)的選擇，像是 PACS_UB 必須使用 384KHz 這樣特殊的頻率時，就只能靠內部運用中斷信號來調整了，如此既不方便又不實際。最後還是用 S2P_P2S 模組的硬體設計較好。

4.3 本章結論

以上比較顯示添加部分硬體確實比全用軟體設計要好，在下表中我們分別對各模組的效能、功率及同步的容易程度綜合整理如下：打勾（✓）的部份表示使用硬體較佳、沒打勾（-）的部份表示無從比較或不相上下。

	Efficiency	Power	Synchronization
HPE	✓	✓	-
SYNC	✓	-	✓
S2P_P2S	✓	-	✓

Table 4_4 Comparison of L2 sub-module with software design

所以在效率的部分使用硬體設計的各模組均比使用軟體設計

來的好，在功率的部分使用硬體設計的 HPE 模組明顯的功率比較低，而 SYNC 和 S2P_P2S 模組則也和用軟體設計相當，在同步的部分 SYNC 和 S2P_P2S 模組明顯的比較好，而 HPE 模組則無法比較，所以整體而言使用硬體設計有較佳的特性。

第五章 硬體通信協定設計於智慧財產權（IP）之運用

在前一章我們已將軟、硬體的通信協定設計方式作一比較，我們也知道使用硬體設計有較好的優點，但使用硬體來設計通信協定，是否每變換一個通信協定就得重新設計一個新的硬體呢？如此一來將無謂的浪費，所以我們引進智慧財產權（Intellectual Property ,IP）的觀念，為使硬體設計能夠重新使用。

5.1 智慧財產權（IP）簡介

當我們希望我們所作的任何設計都能夠被重複的被使用（Design Reuse），這就是智慧財產權（IP）的基本觀念。以有相關的組織訂定了設計的規則及方式，為使智慧財產權（IP）設計及使用更有效率，因為這不是本篇論文要完成的部分，所以對智慧財產權（IP）詳細的運作程序就不多作介紹。

以下各節，將針對各模組的智慧財產權（IP）設計及可重複使用的時機作一介紹。

5.2 智慧財產權（IP）於第二層通信協定之運用時機

在本節中，我們針對 L2 模組，將其中可茲運用的情形作一介

紹：

（一）運用於不同的通信系統中：現今我們在無線電網路系統中，可以發現有非常多不同的通信協定，如果每一個不同的通信協定就要設計一個硬體的話，是非常不經濟的，針對這個情形我們將每個模組的設計，設計成動態的方式，如此一來在不同的通信協定中也可以使用；同時我們也考慮到在第三代行動通信中的運用，因為在第三代行動通信中，封包（Package）、標頭（Header）都是動態且不固定長度，所以我們做了一些特殊的設計，來滿足第三代行動通信。

（二）運用於不同的控制系統中（CPU System）：雖然 L2 模組被設計與 8051 連結，但是我們也希望能夠被運用於不同的 CPU 系統，所以針對此一需求，我們將匯流排（Bus）的寬度作成可調整的，以滿足此一需求。

所以接下來的各節將針對此二項之運用作更深入的描述。

5.2.1 標頭處理單元（HPE）

HPE 電路模組主要是主動比較（Active Correlation）的電路型態，而非被動比較（Passive Correlation）的電路型態。

如圖 4_1 所示是 HPE 中的核心電路，我們首先介紹電路的功能，然後再介紹如何做動態的通信協定運用的控制。

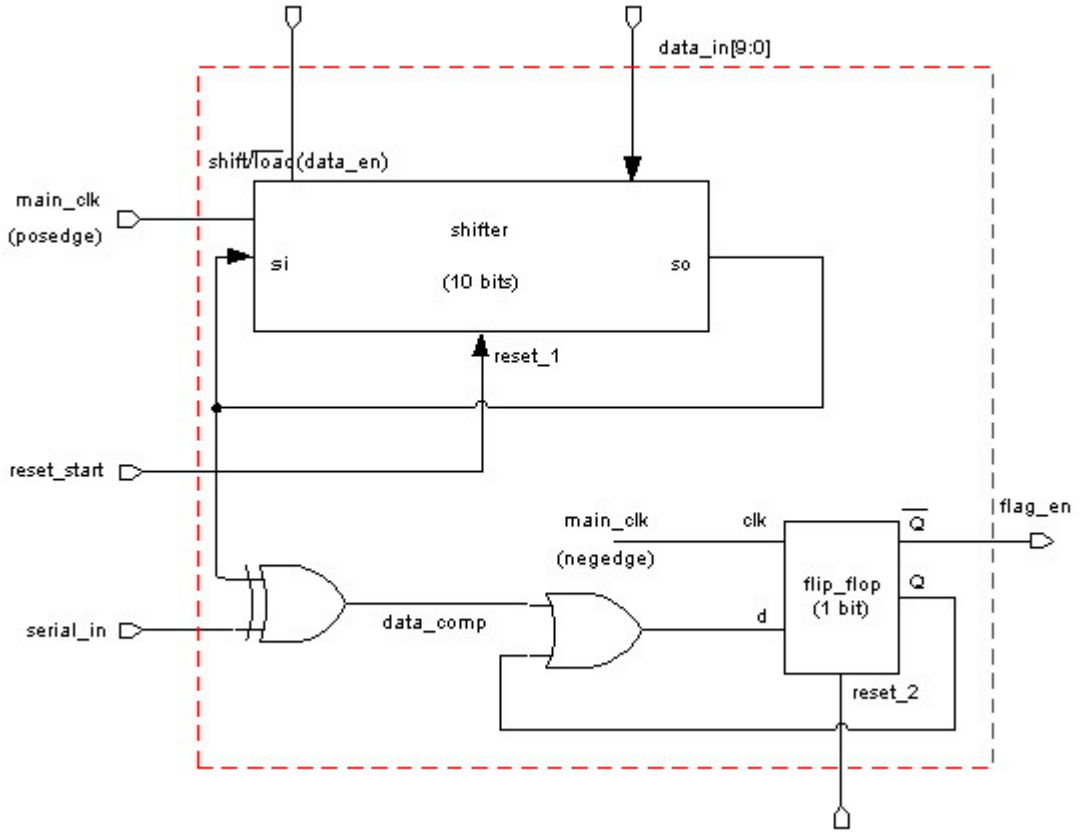

Figure 4_1　Circuit in HPE

HPE 電路模組（詳見上圖）共分為三個主要的部分：

（一）移位暫存器（Shift Register）：這個部分主要是個 10 位元的移位暫存器（Shift Register），為將正確且需要被比較的標頭（Header）值存入，然後以串列的方式輸出，將串列資料與串列輸入的資料作一比較。

（二）比較器（Comparator）：這個部分主要是個互斥或閘（XOR Gate），作為比較器（Comparator）使用，也就是將移位暫存器（Shift Register）串列輸出的資料與從 Layer 1 串列輸入（從 Serial_in 端）的資料作一比較，若是有一個位元錯誤，我們就斷定這個標頭

（Header）值錯誤，因為標頭（Header）值必需完全正確才算符合要求。

（三）鎖住電路（Lock Circuit）：這個電路由一個或閘（OR Gate）和一個 1 位元的正反器（Flip_Flop）所構成，主要是為了鎖住資料用，因為比較器比較的結果如果其中的 1 位元有錯，那後面的資料縱使都對也是無用，所以一旦比較的值有 1 位元錯就將其鎖住並送出錯誤訊息。

接下來將說明如何控制本電路及如何在動態標頭（Header）中應用，移位暫存器（Shift Register）中有 2 根主要的控制接腳，一是 data_in【9：0】端，一是 shift/load 端，data_in【9：0】端主要是將標頭（Header）值輸入，而 shift/load 端主要是控制移位暫存器（Shift Register）是工作於移位或是要載入資料，所以只要在正確的時間點控制 shift/load 端，就可以和串列輸入值作一比較。而 Serial_in 端主要就是輸入從 Layer 1 來的串列信號，兩信號透過比較器就可作一比較，若標頭（Header）值不足 10 位元，只要將要比較的資料，往高位元端輸入，同時控制 shift/load 端即可。

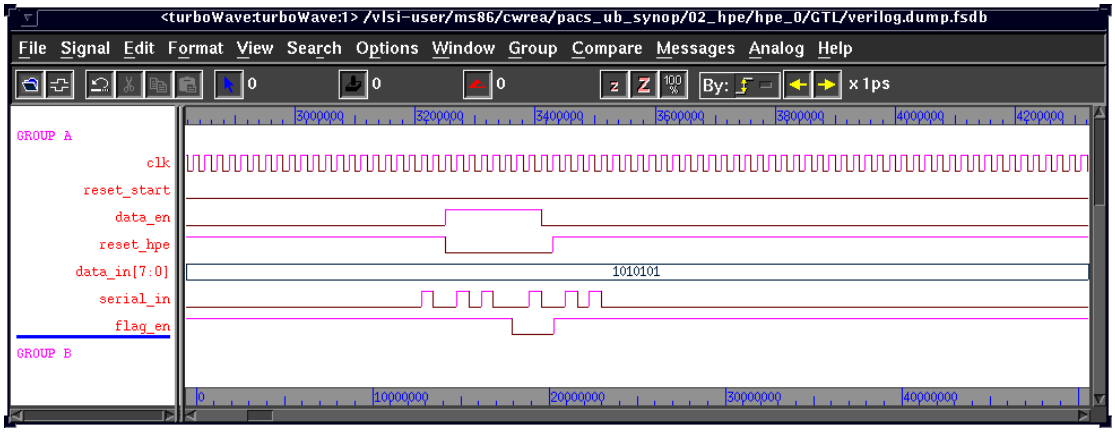

Figure 4_2　Waveform on Correct Header

圖 4_2、4_3 就是本電路作模擬（Simulation）的波形圖，輸入的標頭(Header)值為 8 位元的資料，圖 4_2 是輸入正確的標頭(Header)值，圖 4_3 是輸入不正確的標頭（Header）值。所以我們可以從輸出端（Flag_en 端）看出圖 4_2 有信號輸出，而圖 4_3 則無信號輸出。

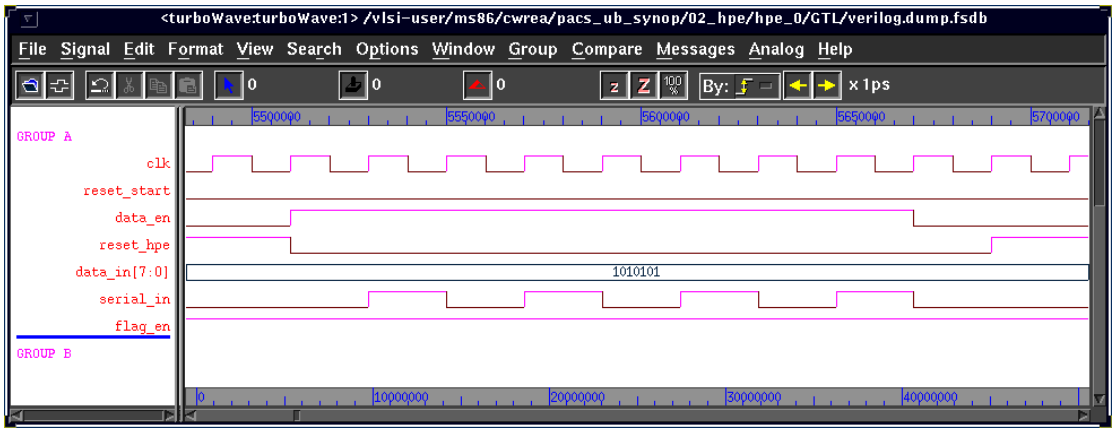

Figure 4_3　Waveform on Wrong Header

所以由波形圖我們可以明顯的看出，我們只要控制 shift/load 端及 reset_hpe 端就可以很容易的作動態的通信協定控制；如此一來這個 HPE 模組，不僅可以在 PACS_UB L2 可以運用，也可以在其他不同的通信系統中應用，而我們要做的只要適當的控制標頭（Header）

值即可。

5.2.2 串列、並列轉換（S2P_P2S）

串列、並列轉換（S2P_P2S）模組共提供了 4 種不同的匯流排（Bus）寬度，分別為 8、16、24、32 位元，為了就是滿足不同的 CPU 系統來使用。

目前我們的原型設計是使用 8051 來作控制器，所以只需要 8 位元的匯流排（Bus）寬度，但若是改用其他的 CPU 系統我們只要作一簡單的調整就可以了，像是用 ARM 這樣的 CPU 來作控制器以取代 8051，我們只要把匯流排（Bus）寬度改為 32 位元就好了。

所以在不同的 CPU 控制系統中，串列、並列轉換（S2P_P2S）模組就可以不斷的重複被使用。

5.2.3 同步（SYNC）

同步（SYNC）模組內部的計數器（Counter）是由三組不同的計數器構成，且是按照 PACS_UB 的規格及時序完成，當我們如此設計時，我們就不需要因為變換 CPU 控制系統，而重新調整時序，所以如此以來就可重複的被使用。

5.3 本章結論

不論是標頭處理單元（HPE）、串列、並列轉換（S2P_P2S）、同步（SYNC）都可以在不同的時機重複的被使用，不論是不同的通信系統中或是不同的 CPU 控制系統中，如此一來這樣的設計就可以重複的被使用，也就符合了智慧財產權（IP）的要求。

第六章　效益評估與測試方式

在這一章中我們將要介紹這一顆 IC 的效能（Performance）和測試環境以及如何測試它。

6.1　第二層（L2）通信協定晶片效能（Performance）

首先，介紹這一顆 IC 的設計流程（Design Flow），我們主要是採用細胞元設計（Cell Base Design）的流程，利用 verilog 硬體描述語言（HDL）作為主要的設計語言，並用 Synopsis 作為邏輯合成工具，並通過 GTL 模擬（Gate Level Simulation）；接著使用 Cadence 來作自動佈局（Auto Place and Rout）以產生實體佈局（Physical Layout），當然也必須通過前置模擬（Pre-sim）和後置模擬（Post-sim），使用的模擬軟體是 Time-mill and Power-mill。

接著要介紹的是所使用的細胞庫（Cell Library），它是採用本實驗室自行開發的 0.35um 製程且低功率的細胞庫（Cell Library），因為低功率的設計可以使通信系統節省更多的能源並增家使用的時間。

最後我們要介紹本積體電路的整體效能表現，它是採用 TSMC1P4M-0.35um 的製程，它的核心大小（Core Size）為 1.872mm^2×1.873 mm^2，體積算是相當的小，相關佈局的位置請參閱下頁，圖 6_1、圖 6_2。

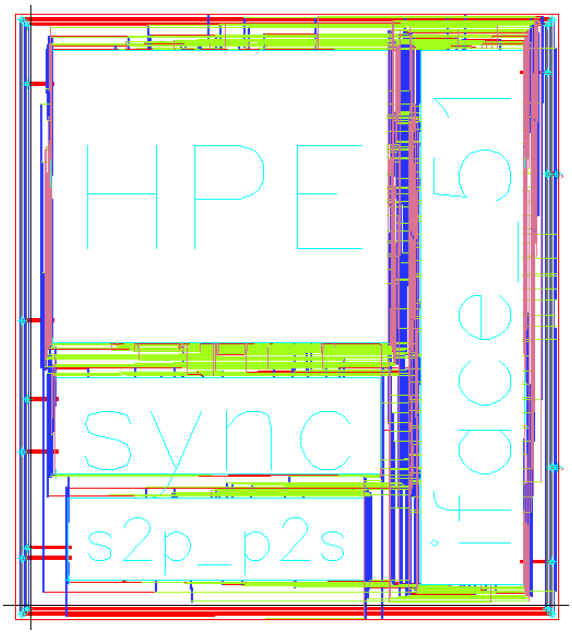

Figure 6_1　Floorplan of L2

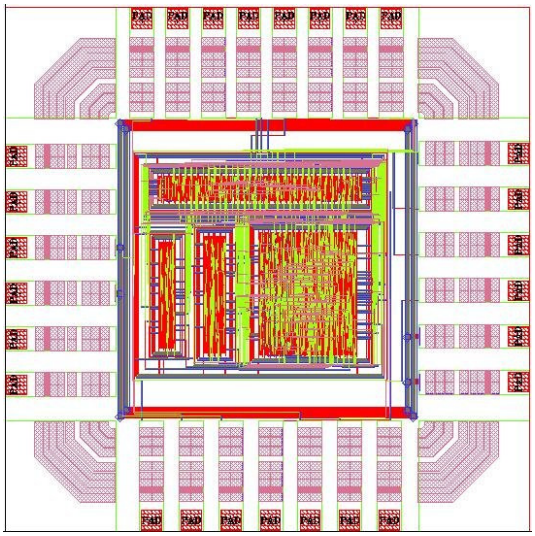

Figure 6_2　Physical layout of L2

另外使用的外部偏壓是 3.3V，並使用的外部頻率是 384KHz，這頻率是由規格表所規定的，並且由系統所提供，而其他相關的參數請參閱表 6_3。

Technology	TSMC1P4M
Number of transistors	11K
External clock rate	12.288MHz , 384KHz
External power supply	3.3V
Power consumption	21mW
Core area	1.872×1.873 mm^2

Table 6_3　Chip Feature

6.2　晶片測試與驗證

晶片的測試環境如圖 6_4 所示，它是由兩個模擬的部分所構

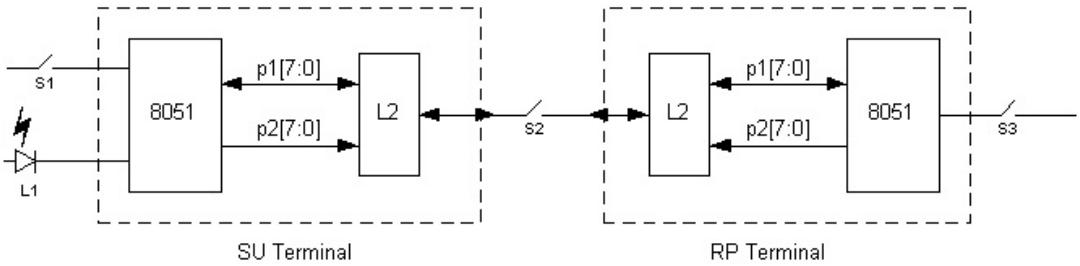

Figure 6_4　Block diagram of L2 testing and verification

成的，主要有 4 顆 IC，左邊虛線的部分是用戶端（SU），由本篇論文所製作的晶片(L2)和 8051 組成，右邊虛線的部分是射頻端(RP)，也是相同的 2 顆 IC，接線的方式也一致，唯一不同的是 8051 內部的控制程式而已，測試的方法就是用用戶端（SU）與射頻端（RP）互相通信，然後產生不同的情況來測試，我們將會利用三個開關 S1、S2、S3，來測試訊框(Frame Sync)無法同步、系統廣播信號(SBC)、初始信號（Initial Access），接下來我們將介紹這三種情形測試的情況。

（一）訊框（Frame Sync）無法同步測試：訊框（Frame Sync）無法同步通常是指射頻端（RP）的訊號比用戶端（SU）快或慢，而照成兩端無法同步，但這樣的情形我們無法在這樣的電路中模擬，所以我們利用 S2 開關來作控制，形成斷訊的情形，類似無法同步，詳細控制情形見下一段敘述。

（二）系統廣播信號（SBC）測試：此時射頻端（RP）會不斷的送出廣播信號（SBC），而用戶端（SU）則負責接收，我們會控制開關 S2，如果 S2 接上（On）則表示用戶端（SU）可以順利接收射頻端（RP）的資料；如果 S2 不接上（Off）則表示用戶端（SU）無法接收射頻端（RP）的資料，像是用戶端（SU）離開射頻端（RP）的

通信範圍或是遭遇雜訊的干擾都有可能，總之這時 Layer 2 會發出信號告知 Layer 1，Layer 1 則會處置像是重新作同步、重新調整時槽（Time Slot）甚至更換通信頻率都有可能。

（三）初始信號（Initial Access）測試：當用戶端（SU）接收完系統廣播信號（SBC）一切無誤後，就可以準備開始初始信號（Initial Access），首先要控制的是用戶端（SU），將 S1 開關接上（On）後用戶端（SU）就會送出初始信號（Initial Access），這個信號必需耗掉有 2 個訊框（Frame），所以當射頻端（RP）接收到信號後，就可以決定是否提供時槽（Time Slot）給用戶端（SU）使用，控制此一命令的開關就是 S3，所以如果 S3 接上（On）則用戶端（SU）接收到的訊息就是允許使用其中的一個時槽（Time Slot）來通信，但如果 S3 不接上（Off）則用戶端（SU）接收到的訊息就是不允許使用其中的任何一個時槽來通信，也許是系統忙碌（Busy）或是其他的原因。

當然除了上述這些狀況外，其實還有很多狀況我們無法一一測試，但是這些測試情形已足夠滿足大部分的功能，並測出大部分的錯誤，表 6_5 顯示在作不同的測試時所使用的開關。

	S 1	S 2	S 3
Frame_Sync		✓	
SBC		✓	
IA	✓		✓

Table 6_5 L2 Module Hardware Testing and Verification

第七章　結論及未來展望

在通信協定引擎（Protocol Engine）的設計中必需掌握兩個重點，一是一般性，一是低功率。在一般性方面，我們希望找出更多的功能，而這些功能是在大部分的通信協定中都能用到的，然後又適合將其製作成硬體，像是這一篇論文中所提出的幾個模組，標頭處理單元（HPE）、同步（SYNC）、串列、並列轉換（S2P_P2S）...等等，當然還有其他適合製作成硬體的模組，所以這些部份就非常適合成為通信協定引擎（Protocol Engine）的一部份，最後再將這些部份跟 CPU 整合在一起，就是一個很好的一個通信協定引擎（Protocol Engine）；在低功率的部分，我們希望能找出更省電的方式來完成通信系統，因為一個可移動的通信系統如果省電的話，就可以延長使用的時間，如此一來的話就可解決一部份電源供應的問題。

在這一篇論文中我們已經改善了 L2 的部分，也提出了一些有效應用的方法，但在通信協定引擎（Protocol Engine）中不是只有 L2 的部分而已，還有 CPU 的控制部分，而這也是可以好好改善的一部份，未來我們希望根據 L2 的部分然後重新設計 CPU 的部分，而這個 CPU 必需符合幾個條件，就是能夠滿足且提供足夠的指令集給通信協定（Protocol）使用，且在這樣的條件下整合出來的通信協定

引擎（Protocol Engine）面積最小功率最低，這樣就是一個相當好的通信協定引擎（Protocol Engine）了。

最後，我們期待更多的無線通信技術能夠被發展出來，使整個無線通性系統能夠更具實用性。

附錄一

Comparison of Deferent Protocol System

Parameter	PHS	PACS
Operating frequency	1895.0-1906.1 MHz	1850-1895MHz 1930-1975MHz
Duplexing	TDM/TDMA/TDD	TDM/TDMA/FDD
Channel spacing	300kHz	300kHz
Transmit power	80mW handset 80mW private base-station 160mW-4W max. public base-station	200mW handset 800mW base-station
Modulation	π/4 QPSK	π/4 QPSK
Channel bit rate	384kb/s	384kb/s
Voice channels per base-station	4	8
Frame duration	5mS	2.5mS
Channel assignment	DCAa	Fixed or QSAFAb
Voice coding	G.726 ADPCM	G.726 ADPCM

Parameter	PACS_UA	PACS_UB
Operating frequency	1920-1930MHz	1920-1930MHz
Duplexing	TDM/TDMA/TDD	TDM/TDMA/TDD
Channel spacing	300kHz	300kHz
Transmit power	53mW handset and base-station	53mW handset and base-station
Modulation	π/4 QPSK	π/4 DQPSK
Channel bit rate	384kb/s	384kb/s
Voice channels per base-station	4	8
Frame duration	5mS	2.5mS
Channel assignment	Etiquette rules	Etiquette rules
Voice coding	G.726 ADPCM	G.726 ADPCM

附錄二

8051 interface with L2(Layer 2) module

Channel number	Sub_module name	Pin name	Direction	
IBus_0	S2p	Fc_data[7:0]	L2→CCU	
IBus_1	HPE_1	Flag[5:0]	L2→CCU	
IBus_2	HPE_2	la_rx_data[7:0]	L2→CCU	
IBus_3	Timer	End_count[3:0]	L2→CCU	
IBus_4	Counter	Burst_num[2:0]	L2→CCU	+
IBus_5	P2s	Data_en	L2→CCU	*
IBus_6	P2s	Data_in_1[7:0]	CCU→L2	
IBus_7	P2s	Data_in_2[7:0]	CCU→L2	
IBus_8	Timer	Timer_triggle[3:0]	CCU→L2	
IBus_9	P2s	P2s_start,p2s_end	CCU→L2	+
IBus_10	AC	Alert signal	CCU→L1	*
IBus_11	Frame_sync	Frame_sync_en	CCU→L2	*#
IBus_12	Latch_boot	Lat_boot_en[7:0]	CCU→L2	#
IBus_13	Frame_sync	Frame_sync_data[7:0]	CCU→L2	#
IBus_14	Latch_boot	Header_0[7:0]	CCU→L2	#
IBus_15	Latch_boot	Header_1[7:0]	CCU→L2	#
IBus_16	Latch_boot	Header_2[7:0]	CCU→L2	#
IBus_17	Latch_boot	Header_3[7:0]	CCU→L2	#
IBus_18	Latch_boot	Timer_0[7:0]	CCU→L2	#
IBus_19	Latch_boot	Timer_1[7:0]	CCU→L2	#
IBus_20	Latch_boot	Timer_2[7:0]	CCU→L2	#
IBus_21	Latch_boot	Timer_3[7:0]	CCU→L2	#

Note:

" r " index reverse direction , L2→CCU.

" * " index only one pin.

" + " index less 8 pins.

" # " index add latch at output.

參考文獻：

【1】 Personal Access Communications System Unlicensed(version B) Air Interface Standard. J-STD-014B

【2】 A.S. Krishnakumar, The Programmable Protocol VLSI Engine(PROVE). IEEE Transaction on Communication 1992.

【3】 A.S. Krishnakumar, The Programmable Protocol VLSI Engine(PROVE). IEEE Transaction on Communication 1994.

【4】 Kiyoshi Kobayasshi, Low-Power and High-Quality Signal Transmission Baseband LSIC for Personal Communications.