



# 可調式晶片系統軟硬體開發平台之研製總計畫

陳少傑 熊博安 李宗演 程仲勝 游竹 嚴茂旭  
計畫編號：NSC-93-2215-E-002-032

## 一、摘要

本總計畫涵蓋下列五項研究項目：(一)可調式晶片系統平台之軟硬體共同設計，(二)可調式晶片系統平台之前置佈局評估回饋系統設計，(三)可調式晶片系統平台之實現：USB 及 Codec 週邊界面，(四)可調式晶片系統平台之實現：內嵌 FPGA，及(五)可調式晶片系統平台之實現：8051 及資料加密器。

本計畫之主要的學術創新將著重於各相關之設計方法論，現場可即時燒錄系統晶片 FPGA，及其資料加密之可調式系統設計平台之實現。由於在此一晶片系統平台中，各工具間彼此依賴，環環相扣，唯有進行整合性研究，才能建立完善的軟硬體設計開發平台與相關技術。而本計畫所完成的行動通訊派遣系統、無線即時生理監測系統，及其可調式軟硬

體設計開發平台，以及各子計畫所研發完成之軟體工具、Embedded FPGA、資料加密 SoC 單晶片等，將可提供技術轉移給航運業、物流業、醫療院所等。

## 二、計畫說明

### (一) 子計畫一：可調式晶片系統平台之軟硬體共同設計

第一階段計畫(NSC-92-2218-E-194-009)由熊博安教授(中正大學資訊工程系)執行。第二年計畫(NSC-93-2215-E-027-006)由李宗演教授(台北科技大學電子系)執行。

#### 1. 摘要

總計畫的設計目標是一個可調式 SoC 平台，同時支援行動通訊派遣系統 MMDS 以及無線生理監測系統(PSMS)。SoC 的設計需要整合相當多不同類型的數位與類比系

統矽智財到一顆晶片上，我們的目標又要系統是可重組式，因此我們需要針對可重組式晶片系統提出與實作一套共同設計方法論。

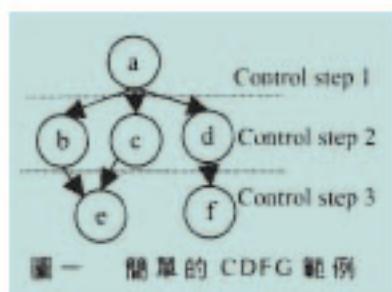
本子計畫在可調式晶片系統平台上完成之項目有軟體分割工具與軟體合成工具之設計，藉由多重分割演算法在可調式晶片系統(TSoC)上發展出一個軟體分割技術。為了提供人性化之人機介面，我們發展出圖形化之可調式晶片系統之軟體分割技術平台，設計者可經由此平台獲得符合系統規格之最佳軟體分割結果。此外，在軟體合成工具上，我們設計一個程式碼產生器演算法-RCGA 來產生在 TSoC 之微處理器語言。而硬體部分除透過 Xilinx Micro-Blaze 系統驗證板來做驗證及實現外，並進行燒錄到 FPGA 進行硬體合成。

#### 2. 研究成果

本節除了介紹如何利用基因演算法來解決軟硬體分割問題外，還會說明如何應用跨平台程式碼產生器演算法及開發圖形化人機界面來解決可調式晶片系統的軟硬體分割問題。

軟硬體分割是一個 NP-hard 問題，我們利用 *Control Data Flow Graph* (CDFG) 做為上層的描述系統行為的工具。CDFG 由節點 (node) 及 acyclic paths 構成，藉由 CDFG 可輕易描述系統的資料流程、控制步驟及是否同步。在此 CDFG 環境中，節點常被用來代表硬體或軟體，而 acyclic edge 則用來代表節點間的關係性。圖一是一個範例，它是 *Control Flow Graph* (CFG) 及 *Data Flow Graph* (DFG) 組成，CFG 用來顯示控制的順序，而 DFG 則用來說明節點 a 被 Control step 1 觸發後將會到達節點 b、c 或 d，因此，任一系統可經由 CDFG 描述出系統行為。圖一中的節點 a、b、c、d、e 及 f 代表一系列的 *Function Elements* (FEs)。

式(1)是基因演算法解軟硬體分割問題時所用的適應函式，其中的 x 包含硬體面積、硬體執行時間、軟體執行時間、軟體記憶體大小及功率消耗：



$$\text{fitness}(x) = \frac{1}{1 - \frac{1}{x}} \quad (1)$$

圖二是我們以基因演算法為基礎，而開發的軟硬體分割工具，使用者只要輸入 population size、crossover rate、mutation rate 及 number of generation 資料後，我們的工具會立即進行相關計算，接下來使用者還要再按下視窗右下角的 Running GA 按鈕後，程式就會進行基因演算法計算並顯示軟硬體分割的結果。

表一是我們提出的跨平台程式碼產生器演算，它主要是用來產生可適用在不同

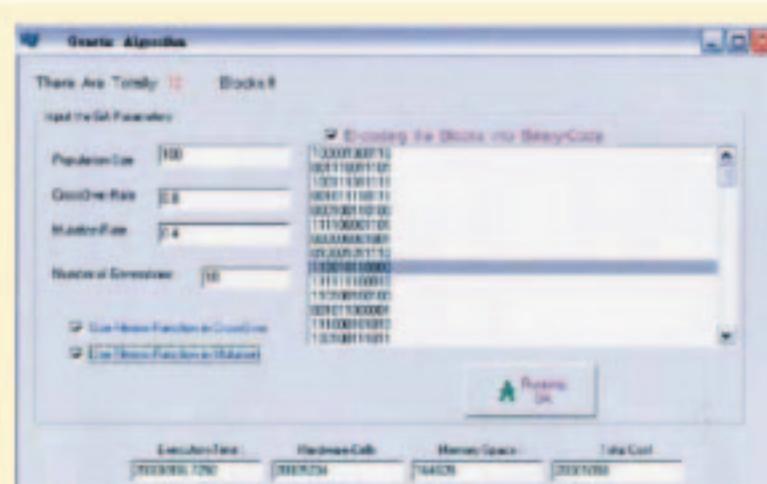
平台的 C 語言程式碼，步驟(1)開啟來源檔及指定輸出檔名，步驟(2)進行 I/O、中斷及計時/計數啟能等初始化設定步驟(3)設定 interrupt time 及 input/output 硬體參數，步驟(4)到(6)進行來源檔剖析，步驟(7)轉換來源檔為可在硬體平台執行格式，步驟(8)進行參數映射，步驟(9)透過 XML 樹及 Binary 樹進行函數轉換。

## (二) 子計畫三：可調式晶片系統平台之前置佈局評估回饋系統設計

計畫編號：NSC-93-2215-E-212-001，主持人：程仲勝教授（大葉大學電機工程學系）。

### 1. 摘要

本整合計畫擬設計之 MMDS/PSMS 系統在設計之初有些功能模組為 hard IPs



圖二 軟硬體分割工具的 GUI



表一 Code Generation Algorithm

<b>Procedure Code_Generation</b>			
Begin			
Open_file();	(1)		
Initialization();	(2)		
Parameter_Setting();	(3)		
//Set the interrupt, vector, timer, I/O port type, etc.			
While (file is not ending)	(4)		
GetToken(); //get the variable from the file	(5)		
if (token = keyword)//include, int, char, etc. (6)			
{if (Token == Selection) //if, case, else...etc. Selection(); //parsing if, else, and get the condition else if (Token == Loop) //while, for ... Loop(); //get the loop times			
		//and ending condition else if (Token == HeadFile) //include Headfile(); //get the head file else if (Token == Declare) Declare(); //store the variable, array, and //subroutine. else Comment(); //end of if else Variable(); End;//end of While	(7)
		Statement();	(8)
		Translation();	(9)
		End;//end of Begin	
		End;//end of Procedure	

，有些為 soft IPs，有些為本整合計畫欲設計之子系統。設計如此複雜系統耗時甚鉅且各功能模組需不斷修改以符合系統目標，因此對於後端實體設計者而言，必須等待較長系統設計時間才能真正著手相當費時的電路佈局工作。另一方面，為了達到佈局最佳化以保障系統效能的目標，有些佈局問題根本解決之道可能必須追溯至系統階層修正軟硬體共同設計（HW/SW codesign）或修改其他子計畫所設計之系統模組。因此為了縮短整個系統佈局時間及達到佈局與系統模組設計最佳化，正本清源的方法為在佈局前之系統設計時即嘗試模組平面規劃並評估晶片面積、繞線延遲、及繞線擁擠程度等佈局最佳化重要因子，而後回饋相關訊息給系統設計者以修正系統設計，使得爾後能有效率

地獲得正確的實體設計結果。

綜上所述，本子計畫三擬設計行動通訊演進系統及無線即時生理監測系統（MMDS/PSMS）SoC 之前置佈局評估回饋系統。本計畫並非實際對 MMDS/PSMS 做電路實際佈局，而是利用經過初步系統階層之軟硬體共同設計所得之模組如：8051 core，可能以 hard/soft IP core 呈現之 GPS、GIS 及即時嵌入式系統以及欲由其他子計畫設計之建構於 FPGA 之可重組態功能單元、週邊硬體界面與 AES 加解密等模組尚未設計完全前預先快速嘗試此類不確定模組之平面規劃以求得可能之模組平面規劃型態及其相對應之未來佈局重要評估因子如晶片面積、繞線擁擠程度、重要訊號繞線長度、溝通頻繁模組之群聚性等。之後再將所得資

訊作一分析評估，並將評估結果回饋給其他系統階層子計畫之設計者以作為其先前設計時所可能忽略之後端佈局環境參數，藉此改善不當系統設計及其對後端實體設計所帶來不良之影響。

## 2. 研究成果

在 92 年度計畫中我們已採用模擬退火方式及可切割（slicing）結構做為不確定模組平面規劃的程式主體。一開始模擬退火演算法使用一個任意的可切割平面圖當作初始解並伴隨著一個能表示此可切割平面圖的波蘭表示式（polish expression）。接著對此初始波蘭表示式執行一連串的移動（move）以求得最佳的平面圖。移動的策略採用傳統的方式進行，如交換波蘭表示式中兩相鄰運算元、交換水平及垂直分割兩運算子順序及交換相

鄰運算元與運算子等。所不同的是每次在階層式由下往上聚集模組形成平面圖的過程中，不是使用只能將具有確定寬高模組群集的運算，而是使用能接受模組寬、高機率分佈串列並能對其寬高值及相對應機率加以處理的運算。如此使得每次群集後所形成較大的模組亦有其寬、高機率分佈串列。至於模擬退火的成本函數（cost function）則定義為整個平面圖寬、高期望值的乘積與寬、高變異數的乘積之線性組合以嘗試降低平面圖面積及增加估計的可靠性。除了可切割結構為主的平面規劃外，我們也將不確定模組平面規劃擴展至以 CBL (Corner Block List)、SP (Sequence Pair) 及 BSG (Bounded Slice-line Grid) 等表示法來表示不可切割 (nonslicing) 的平面圖結構，以求得具有更小估計面積的平面規劃結果，形成一套較完整的不確定模組面積最佳化平面規劃系統。

在 93 年度計畫中，首先給定一個從 92 年度所求得之平面圖為初始平面圖。對此初始平面圖，我們先估計出各連接線長度及繞線擁擠程度，接著漸進地局部改進初始平面圖（亦即重新調整模組間的相對位置）以減少連線長度及降低繞線擁擠程度，在減少連線長度方面，

由於平面規劃時尚不知詳細繞線路徑，因此無法得知實際繞線長度，但可估算出每條連接線長度，據此可控制過長網列連接線。而在降低繞線擁擠程度方面，可藉由減少平面規劃區域中單位網列繞線密度來達成。以下定義所欲解決問題的目標為：

- (1) 估計晶片面積最小。
- (2) 若有  $K$  條最長連接線其線長有  $\%p$  以上的機率為  $L$  單位長度，則將  $K$  值、 $p$  值、 $L$  值盡可能減小或找出其對降低連線長度有意義之組合值。
- (3) 將平面圖切割成若干大小一致的單位，若其中網列繞線密度最高的單位個數為  $M$  且其密度為  $D$ ，則盡可能將  $D$  值及  $M$  值減小。

由上得知，考慮降低不確定模組區塊間連接線長度與繞線擁擠程度的平面規劃問題是一個同時進行多項因數的最佳化問題。必須注意的是，為了減少連接線過長問題有時反而會使某些繞線區域密度增大，反之亦然。故上述問題是一極為複雜的最佳化問題。

針對以上問題，我們已針對可切割式結構的初始平面圖加以改進，且已採取模擬退火演算法來改進不確定

模組區塊間連接線長度與繞線擁擠程度。至於演算法中尋找下一個可行解的策略，我們採用由下而上 (bottom-up) 的樹狀結構搜尋策略及模組兩兩交換技巧來達成，其主要步驟有二：

- 步驟 1：針對欲改進平面圖  $G$ ，找出  $K$  條最長連接線（其線長有  $\%p$  以上的機率為  $L$ ）及  $M$  個繞線密度最高的單位格子（其密度為  $D$ ）。
- 步驟 2：針對平面圖  $G$ ，反覆找出可交換位置的模組。若交換後，重新計算之  $K$ 、 $p$ 、 $L$ 、 $M$ ，及  $D$  值優於舊值且晶片面積不大於前值，則將其交換。以下為目前利用平面圖  $G$  相對應之可切割樹來找出可交換位置模組的方法：
  - (1) 找出所有具有相同父節點的兩個樹葉節點加以交換。交換順序可依其父節點在可切割樹由左至右、由右至左的順序或隨機方式交換其所屬子節點。
  - (2) 交換方式與順序同(1)，但交換的節點由樹葉節點往上一層（即樹葉節點之父節點）。依此類推，我們可由下至上將可切割樹的每一層節點依序兩兩交換以求得不同的平面圖。



另一方面，若初始平面圖為一不可切割式結構，我們亦可採取模擬退火演算法來改進不確定模組區塊間連接線長度與繞線擁擠程度，在模擬退火演算法中，根據其伴隨能表示此不可切割平面圖的表示式，我們執行一連串的可行移動（feasible move）以求得最佳的平面圖。至於移動策略將視採取何種表示式而定。而模擬退火的成本函數則定義為整個平面圖寬、高期望值的乘積、繞線總長度與總繞線密度之線性組合以嘗試降低平面圖面積、訊號繞線長度及繞線擁擠程度等重要佈局評估因子及增加估計的可靠性。

### (三)子計畫五：可調式晶片系統平台之實現：USB 及 Codec 週邊界面 (II)

計畫編號：NSC-93-2215-E-197-003 主持人：游竹教授  
(宜蘭大學電子工程學系)

#### 1. 摘要

系統晶片本身要配合目標應用系統（含軟硬體）的發展，必須要有一個良好的設計環境，包括 Hardware-Software Codesign 工具、Electronic Design Automation (EDA) 工具，以及發展上述工具時用於測試之可調式系統設計平台及研究載具等。於此一設計環境中，各工

具間彼此依賴，環環相扣，唯有進行整合性研究，才能建立完善的軟硬體設計開發平台與相關技術。

本子計畫依照先前規劃第一年完成擴充界面軟式智財晶片（soft IP）設計，像是 RS-232、USB 界面等主要的周邊 IC，而第二年度實現其相對應的硬式智財晶片（hard IP），也將提供本「可調式晶片系統（System-On-Chip）平台」各相關子計畫間的重要資訊，以利他們有實際的依據做適當的修正。另外，第二年度的後半年也規劃完成語音 Codec、顯示模組控制器等軟式智財晶片設計與實現。由於在不同應用系統中可能會有不同需求的周邊界面，因此我們將配合子計畫六的「可調式晶片系統平台之實現：內嵌 FPGA」的邏輯結構及子計畫四「可調式晶片系統平台之嵌入式即時作業系統及軟體合成」做適當的設計調整，以得到較佳的電路架構。接著本年度我們也將搭配子計畫三進行晶片佈局，以得到面積、繞線區域擁擠程度、重要訊號繞線長度、溝通頻繁模組之群聚性問等之最佳化。此外，本計畫所有的周邊 IC 的硬體實現成本將經由子計畫一「可調式晶片系統平台之軟硬體共同設計」的最佳化結果做為依據，設計其最佳

成本的硬體架構。

#### 2. 研究成果

##### (1) 通用序列匯流排 (USB)

USB 為通用序列匯流排 (Universal Serial Bus) 之簡稱，為電腦與週邊設備間廣被採用的通訊介面。在 USB 1.1 版的規格，規範了低速 (1.5 Mbps) 與全速 (12Mbps) 介面，在 USB2.0 版增加了高速 (480Mbps) 介面。USB 使用兩條銅導線傳輸差動訊號，並使用 NRZI 的編碼方式，可確保傳送與接收端同步而不需額外的時脈訊號。

圖三為本設計的 USB 硬體架構方塊圖，此架構分為功能介面、控制端點單元、USB 協定層 (SIE)、實體層、收發單元及記憶體 (FIFO) 等為主要，其中功能介面將提供 USB 的功能控制，未來也可接前述的 8051 微控制器達成 SCSI 指令轉換成，以便符合 USB Mass Storage Class 驅動程式的規範。另外架構中的裝置 (Device)，未來可以是 ROM 或其他周邊，如果是 ROM 它就相當於市售的隨身碟。

##### (2) RS-232 電路設計

圖四為本計畫所使用的 RS-232 電路架構圖，其中我們根據 RS-232 的規範，將整個電路規劃成傳輸電路及接

收電路，由於此部分將整合於本計畫的 SOC 晶片設計內，故其介面需設計符合我們所規劃 I/O 匯流排，以易於日後的整合。除了 RS-232 外，未來也將計畫納入 USB 的介面，以使我們的系統更加完整。此外，對於 RS-232 的電路設計主要是在於傳送及接收單元的時序控制器，因此只要清楚 RS-232 的時序規範，整個 RS-232 的晶片設計就等同於兩個有限狀態機的實現。

### (3) 語音編解碼器

語音編解碼器主要是用於提供系統泛用型的語音界面，我們將此主體分成兩個部分，即「串列傳輸介面」及「串列介面 codec」，串列傳輸介面負責將位元形式的語音送入系統如圖五所示，而串列介面 codec 則是實質接喇叭及麥克風，由於此部分混合類比信號處理及數位信號處理，因而有一定的複雜度，初期為了降低語音編解碼器的複雜度，我們將只設計串列傳輸介面，而串列介面 codec 將採系統外接形式，目前市面常用的 IC 為 TP3054，此 IC 為具 a-Law 及 μ-law PCM Codec，並含有 ADC/DAC 及濾波器電路，詳實的電路架構可參照其 IC 規格。

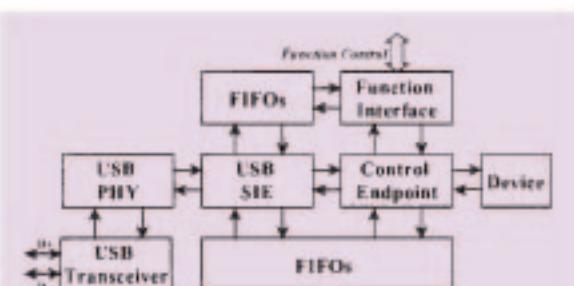
### (4) 顯示模組控制器

為了提供總計畫中的「行動通訊派遣系統」及「無線生理監測系統」充當資訊檢視之用，我們設計一個顯示模組控制器介面。如圖六所顯示，此一模組能提供簡單的 640 × 480 VGA 模式顯示，未來可以直接連結中小尺寸的 LCD 顯示器，然而這方面的電路設計重點主要是在於同步信號與輸出時序的產生及控制。另外，由於一般顯示記憶體的空間需要很大，因此我們將採同一般的 PC 顯示卡作法，將顯示用記憶體採外接形式。

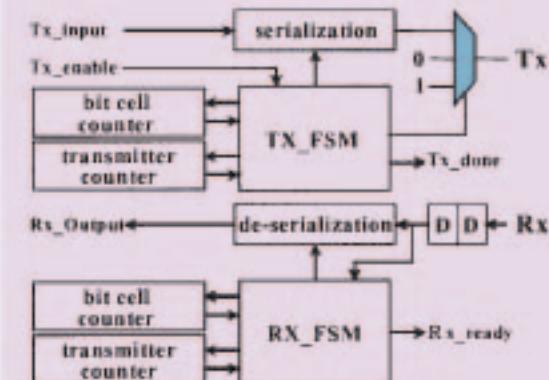
以避免將來晶片實現的時候佔據過大的晶片面積而顯的不實際。

### (四) 子計畫六：可調式晶片系統平台之實現：內嵌 FPGA (II)

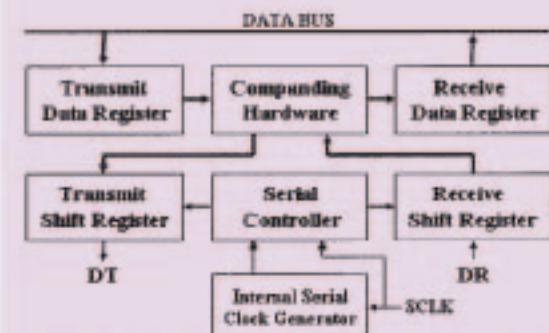
計畫編號：NSC-93-2215-E-018-007，主持人：嚴茂旭副教授（彰化師範大學資訊工程學系）。



圖三 USB 硬體架構方塊圖



圖四 RS-232 系統架構方塊圖



圖五 語音編解碼器系統架構方塊圖

#### 1. 摘要

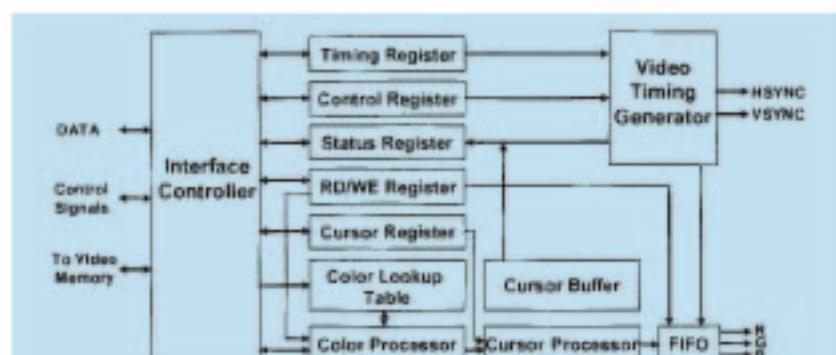
本計畫之研究總目標為建立一個「可調式 SoC 平台」。首先我們將以此新一代的「可調式 SoC 設計開發平台及環境」，建立一個具資料加密能力，且支援「行動通訊派遣系統」及「無線即時生理監測系統」。所以「可調式 SoC 平台」的軟硬體，應具有彈性、易擴展等特性，以

符合未來的新的應用及標準。  
 ·降低開發新 SoC 的成本。  
 延長 SoC 產品的壽命，提高  
 SoC 產品的效能。

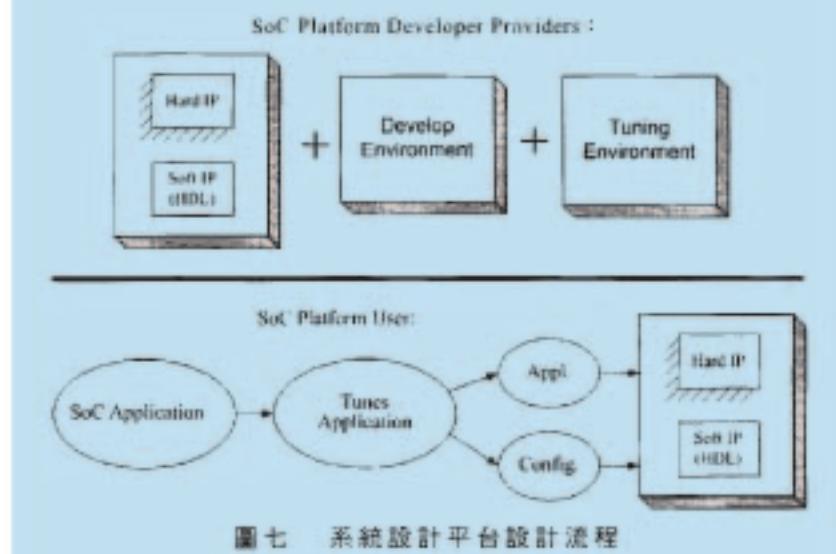
本計畫針對可重組式 SoC 系統首先提出一個同時具有「即時重組」(Real-Time Reconfigurable) 及「執行時間重組」(Run-Time Reconfigurable; RTR) 技術之 Embedded FPGA。此「即時重組」、「執行時間重組」技術可提昇 Embedded FPGA 的函數密度 (Function Density)、降低 FPGA 整體硬體成本，因此結合此兩技術之 Embedded FPGA，將加強可調式 SoC 平台的實用性。

## 2. 研究成果

一般典型 SoC 平台上包含許多 IP，有微處理器、cache、memory、bus、peripheral、FPGA 等。這些 IP 都具備有一些參數，以提供設計者針對特定 SoC 產品之晶片大小、價格、能量等需求，「調整」(Tune) 這些參數來完成設計。因此設計者必須有一方法，以找到這些適當參數後，重組 (Reconfigurable) SoC 以符合系統規格要求，如圖七所示，所以 SoC 平台要配合目標應用系統 (含軟硬體) 的發展，必須要有一個良好的設計環境，包括 Hardware-Software Codesign



圖六 顯示模組控制器系統架構方塊圖



圖七 系統設計平台設計流程

工具、Electronic Design Automation (EDA) 工具，以及發展上述工具時用於測試之「系統設計平台」及研究載具等。

此可調式系統設計平台中將提供參數式 IP，這些參數式 IP 是以 HDL 形式或重組 IC 形式顯示，而在每個設計軟體將加入評估和可調性 (Evaluation and Tuning)。因此設計者於 SoC 設計流程步驟中，將所設計架構可不斷的評估和調整至最佳化後，才重組 (Reconfigurable) SoC 來滿足系統規格的要求。

並製作 SoC 晶片。因此，於此 SoC 設計流程中，各工具間彼此依賴，環環相扣，唯有進行整合性研究，才能建立完善的軟硬體設計開發平台與相關技術。

藉由「可調式 SoC 平台」的軟硬體設計，我們可完成一個 MMDS/MSPS 的可重組 SoC (Reconfigurable SoC)，其內包含許多可重組化的硬體單元。一個「可調式 SoC 平台」之可重組化硬體，係包含了一個內嵌式 (Embedded) 微處理機、內嵌式記憶體 (Memory)、USB/PCI/Ether-

net，及一個或多個的可重組化（Reconfigurable）單元。微處理機以執行演算法及控制系統為主，而可重組化單元則可分為幾個種類，分別是以處理資料算術邏輯運算為主的可重組化硬體（Reconfigurable Hardware），以處理資料傳輸為主的可重組化匯流排或連線（Reconfigurable Bus or Interconnection），以及可重組化記憶體（Reconfigurable Memory）與可重組化輸入輸出（Reconfigurable I/O）。其中，可重組化硬體是一個可重組化 SoC 中重要的單元，扮演著協同處理器（Coprocessor）的角色用以加速特定的運算，提高系統效率。

本子計畫六第一年已完成所有包含內嵌式 FPGA 的架構之 Soft IP 研發，第二年將所完成的 Soft IP 轉換為 Hard IP。此內嵌式 FPGA 架構之主要研究重點有：

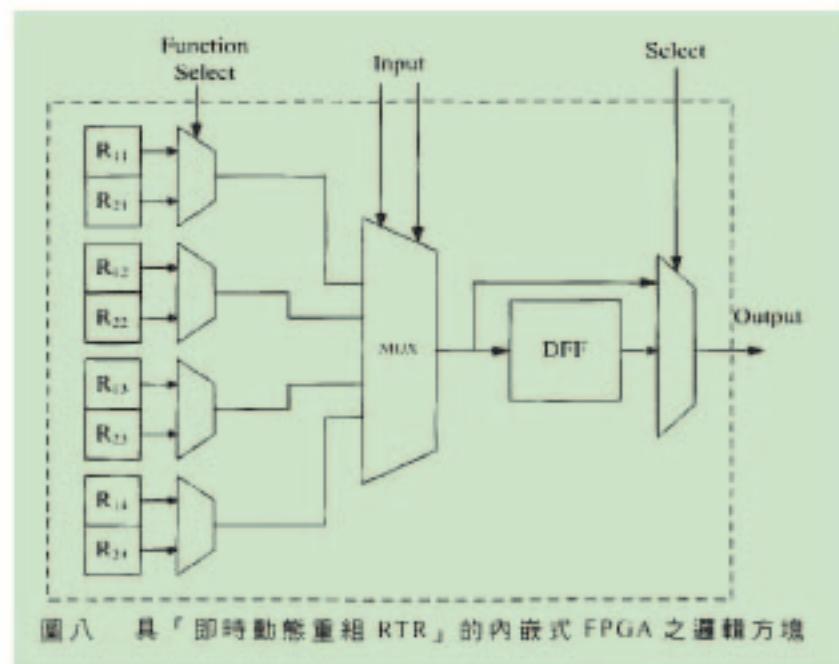
- (1) 設計內嵌式 FPGA、微處理機及記憶體間的匯流排架構。
- (2) 內嵌式 FPGA 之區域 RTR 架構。
- (3) 即時（Real Time）重組之 RTR 技術。
- (4) 內嵌式 FPGA 之統線網路架構。
- (5) 內嵌式 FPGA 之邏輯模組架構。

#### (6) 內嵌式 FPGA 之輸入/輸出架構。

具有動態重組 RTR 機制之內嵌式 FPGA，可提昇內嵌式 FPGA 整體的計算函數密度。但內嵌式 FPGA 的動態重組 RTR 是需要一些時間的，即使是採用高速的平行重組介面，動態重組時間仍須要約數毫秒（ms）。若採用簡單、低速之串列重組介面機制，動態重組所需的時間就更長了。對於即時 SoC 系統，當要動態重組內嵌式 FPGA 時，即時 SoC 系統必須等待內嵌式 FPGA 重組完成，這將無法滿足即時可重組 SoC 之系統需求。因此本計畫是第一個針對即時可重組 SoC 系統，提出即時動態重組 RTR 機制之內嵌式 FPGA，如圖八所示。此內嵌式 FPGA 將可大幅提升整體可重組

SoC 系統的速度。

一般而言，SoC 產品中微處理機執行的程式都是「固定不變」的。因此我們可很容易將下一個即將被微處理執行的大量運算或迴圈，事先將「內嵌式 FPGA」重組，以實現此大量運算或迴圈，也就是以「硬體來實現軟體」。如此將可降低系統執行軟體所需的能量 Energy (= 功率 Power × 時間 Time)，加快執行時間、提高系統硬體的彈性。在此 Embedded FPGA 的作用如同一個協同處理器（Coprocessor），且由於 Embedded FPGA 所減少軟體的執行時間（Time），遠小於 Embedded FPGA 所增加的功率（Power）消耗，所以 Energy (= Power × Time) 是相當小。因此結合 Embedded FPGA 的可重組 SoC 是有相



圖八 具「即時動態重組 RTR」的內嵌式 FPGA 之邏輯方塊

當的能量效率。由於此內嵌式 FPGA 具有此種可即時切換執行函數概念，故可同時支援 MMDS/PSMS 可重組 SoC 產品的開發，及提供「可調式晶片系統平台」之硬體可調性。

本子計畫內嵌式 FPGA 研究完成後，除了在各年度所實現之 IP、EDA 工具、晶片架構會有所創新外，而本子計畫所完成的架構，與 EDA 工具將能夠技術轉移給產業界，應用於實際 SoC 之設計上，使得 SoC 產品更具有彈性、易擴展，以符合未來的新的應用及標準，降低開發新產品的成本。

#### (五) 子計畫七：可調式晶片系統平台之實現：8051 及資料加密器 (II)

計畫編號：NSC-93-2215-E-002-034，主持人：陳少傑教授（台灣大學電子工程研究所）。

##### 1. 摘要

本計畫擬設計一具資料加密能力之可調式 SoC 硬體平台，俾支援行動通訊派遺系統 MMDS 及無線生理監測系統 PSMS 之應用，並證明此可調式平台之實用性。在本計畫中，硬體方面我們將開發 8051 微控器、Compact Flash 存取介面、無線通訊模組介面、AES 加解密模組以及一個

實際可用的可調式 SoC 平台 (Tunable SoC Platform)。

我們由 8051 與 AES 模組的設計開始，包含對 8051 指令集的分析、ALU 的設計、AES 演算法的研究，研讀 USB2.0、Mass Storage Class、ATAPI、SCSI、MMC、RPC、Compact Flash、IEEE802.3、IEEE802.11b、IEEE802.15 等相關規格與標準，制定系統架構與介面。我們在第一年的計畫中，已完成在 FPGA 板上驗證 Soft IPs 及完成 8051 微控器、Compact Flash 存取介面、USB 介面、Bluetooth Transceiver，以及 AES 加解密器之 Hard IPs 模組開發。第二年計畫的重點是將第一年已於 FPGA 驗證的 IP 實際整合於晶片中，已完成晶片有 USB2.0 Compact Flash Reader 及 Bluetooth low-IF receiver 晶片。

##### 2. 研究成果

可攜式電子產品，諸如隨身碟、數位相機及 MP3 播放器等，在我們的日常生活中越來越普遍。在這些產品中，數位聲音影像壓縮及大量儲存裝置是主要的關鍵技術。這些產品的主要功能，就是將聲音影像壓縮並儲存至大量儲存裝置，或從大量儲存裝置讀回解壓縮再播放。大量儲存裝置可以是硬碟

或是快閃記憶體。硬碟的儲存容量較大，但需要較大功率且抗震性較差，故快閃記憶體在可攜式電子產品仍是主要應用。

快閃記憶體的存取速度通常是系統效能的瓶頸。現今數位邏輯及有線傳輸通訊皆可輕易達到每秒數百萬次的操作頻率，但快閃記憶體只能達到每秒數十萬次的操作頻率。而且在儲存媒介上需要實施檔案系統管理以提高記憶體空間利用率，這對作業系統上是一個很大的負擔。

可攜式電子產品還需要有效率的數位傳輸通訊系統。舉例而言，由於數位相機記憶空間有限，每次照像後必須將照片傳輸到個人電腦加以保存。若使用簡易的傳輸系統，則傳輸影像需時數小時，不符合可攜式產品使用便利性的需求。以現今的科技技術，通用序列匯流排 (USB) 是較適合的傳輸系統。在本可調式平台上，尚需連接到全球定位系統 (GPS)，它使用的傳輸系統是 RS232，需連接到影像編解碼器，使用的傳輸系統是 I2C，及連接到語音編解碼器，使用的傳輸系統是 I2S。另外還需要影像擷取裝置的介面，在此使用 CMOS SENSOR 的數位影像。

在可攜式電子產品，資

料安全性是一個日受重視的問題。若是產品遺失時不希望別人獲悉所儲存的私密資料，資料必須經過加密。常用的加密演算法有非對稱加密的 RSA ( Rivest, Shamir, and Adleman )，及對稱加密的 DES ( Data Encryption Standard ) 與 AES ( Advanced Encryption Standard )。RSA 較為複雜，需要較高運算能力與功率，DES 較容易被破解，AES 相較之下是較適合的折衷方式。

如圖九所示，本可調式平台內包含了兩個 8051 微處理器核心，其中之一作為系統的功能控制器，另一個用以處理檔案系統。這個 8051 核心使用精簡指令集處理器 ( RISC ) 的方法重新設計，以解決一些原來 8051 的複雜指令集處理器 ( CISC ) 所造成的效率損失。

除了影音功能以外，可攜式電子產品還可應用在健

康照護上，以隨時偵測血氣血糖等需隨時注意的健康風險指標。本平台還包含了一些生理訊號感測器 ( SENSOR ) 的介面，由於各種生理訊號的處理方式差異很大，此部份需要做成可重規劃的數位訊號處理核心。

在實作上，重新規劃是透過通用非同步序列傳輸匯流排 ( UART ) 將資料載入晶片內部的 RAM 以調整數位訊號處理核心的功能，這傳輸不需 CPU 的輔助，並且在重新規劃期間不能影響 CPU 的功能。

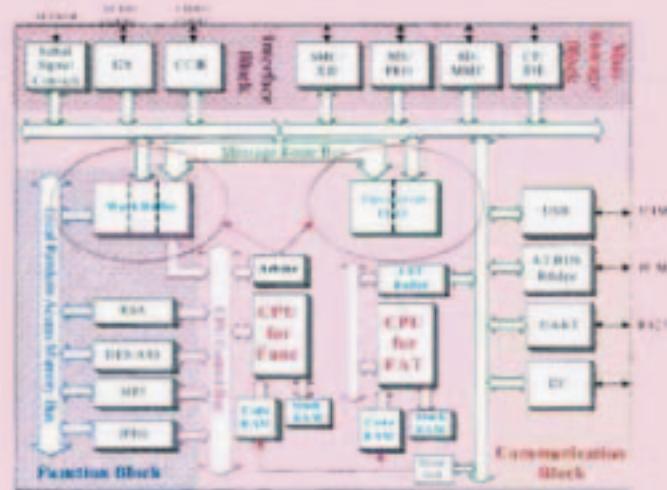
#### (1) 訊息轉送匯流排設計

圖九顯示本 SoC 系統平台的架構，其中，Interface Block、Mass Storage Block 及 Communication Block 的邏輯閘會被放在各角落靠近它們接腳的位置，過長的接線會降低系統的效率與雜訊

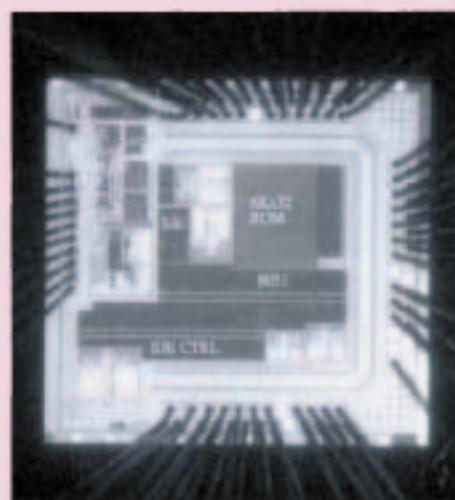
忍受力。但因所有功能區塊間的通訊都是循序資料，而不像 RAM 般的隨機存取時，可以考慮將通訊訊息都包裝成封包。封包的觀念來自數位通訊，例如網路上傳送的是 TCP/IP 格式的封包，封包中包含了傳送來源與目的地的資訊，不同的封包間有適當的區隔標記，則許多來源與目的之間的傳輸皆可在網路間傳送。

當訊息被包裝成封包後，訊息就可以被緩衝轉送。當傳輸路徑太長，可以在路徑中間加一或多級緩衝暫存器，將封包先暫存起來再轉送至下一級。如果將數個端點的訊息先集中至緩衝暫存器再轉送，則可大量減少點對點傳輸所需之接線，此時緩衝暫存器所扮演的功能如同網路上的路由器。

#### (2) 通用序列匯流排設計



圖九 SoC 系統平台



圖十 USB 讀卡機晶片



通用序列匯流排 (USB) 是目前個人電腦與設備間最常使用的有線通訊介面，其速度可達每秒 480 百萬位元 (480Mb/s)。通用序列匯流排是由三個部份組成。第一部份為實體層 (PHY)，它是以類比電路技術設計的工作在 480MHz 頻帶的有線傳輸模組，它的主要挑戰在降低符號間干擾 (ISI)。第二部份為會議層，它負責平行低速資料與序列高速資料的轉換，低雜訊干擾的編解碼。

以及通訊協定的維護。第三部份為應用層，包含 CPU、RAM、ROM 及其它達成晶片應用所需的功能方塊。

圖十是本 SoC 平台中 USB 讀卡機設計部份的晶片，它的功能是透過 USB 界面存取 Compact Flash 記憶卡的資料，它包含 USB、8051 CPU、大量儲存裝置與快閃記憶體界面的功能，但不包含影像、聲音、與加解密功能方塊。它的製程是 SMIC

0.18um，晶片面積是 2200um × 2000um，消耗功率 70mA × 3.3V (含快閃記憶卡)，有關 USB 讀卡機的電路，已由 FPGA 原型驗證並撰寫於第一年報告中，於此不再詳述。

### 三、計畫成果自評

本計畫進行相當順利，原計畫書中所預計完成之各工作項目均已達成，過去兩年來本研究團隊已有一百多篇論文發表。

#### 作者簡介



陳少傑  
國立台灣大學電子工程  
研究所教授  
美國南美以美大學電機  
工程博士  
專長：無線電路設計；軟  
硬體共同設計  
電話：(02)33663647  
傳真：(02)23638247

#### 作者簡介



李宗演  
國立台北科技大學電子  
工程系副教授  
國立台灣大學電機工程  
博士  
專長：軟硬體共同設計；  
嵌入式軟體合成  
；FPGA 系統設計  
電話：(02)27712171 轉 2251  
傳真：(02)23638247



游竹  
國立宜蘭大學電子工程  
學系副教授  
國立台灣大學電機工程  
博士  
專長：積體電路設計；影  
像編碼  
電話：(03)9357400 轉 651  
傳真：(03)9369507



程仲勝  
私立大葉大學電子工程  
學系助理教授  
國立台灣大學電機工程  
博士  
專長：積體電路實體設計  
；電子設計自動化  
電話：(04)8511888 轉 2186  
傳真：(04)8511245



嚴茂池  
國立彰化師範大學資訊  
工程學系副教授  
國立台灣科技大學電子  
工程博士  
專長：超大型積體電路；現  
場可程式邏輯陣列  
電話：(02)24622192 轉 6652  
傳真：(02)24623249