設計與實作可展延低功耗定製內嵌系統之即時作業系統

熊博安　國立中正大學資訊工程學系
計畫編號：NSC-94-2215-E-194-005

一、摘要

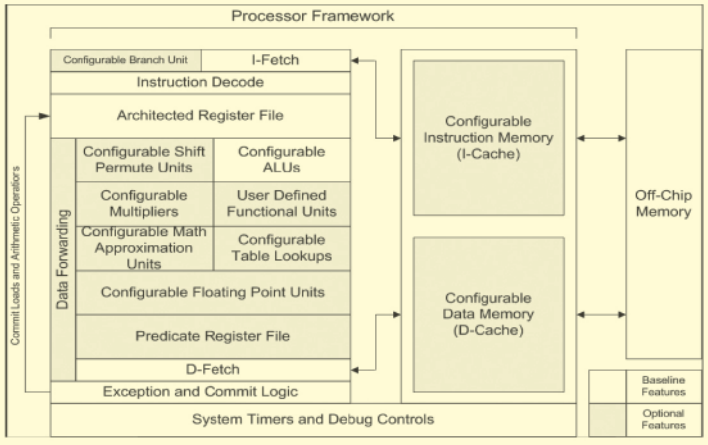
本計畫的主要目標是設計與實作一套專門針對具有原生 SIMD 指令集的微處理器例如 PLX 的開放程式碼、可行、有效率的可組態即時作業系統 (RTOS)。而我們的目標不僅是設計一套可行的 RTOS，而且是一個盡量利用 PLX 微處理器特性以及可在面積、效能與功耗間做取捨的作業系統。事實上，我們並非只有設計一套固定的 RTOS，而是設計一個根據 PLX 組態的機器定義 檔 (machine definition file) 自動生成該 PLX 組態專屬的 RTOS。我們所發展的 RTOS，目標也將支援多媒體的應用，例如 H.264 高效能視訊壓縮等。

相較於現有的 RTOS，我們所提出的 RTOS 模組將完全地支援 PLX ISA 的可展延與可組態的特性，因此可達成緊密整合、高效能與低功耗的目標。

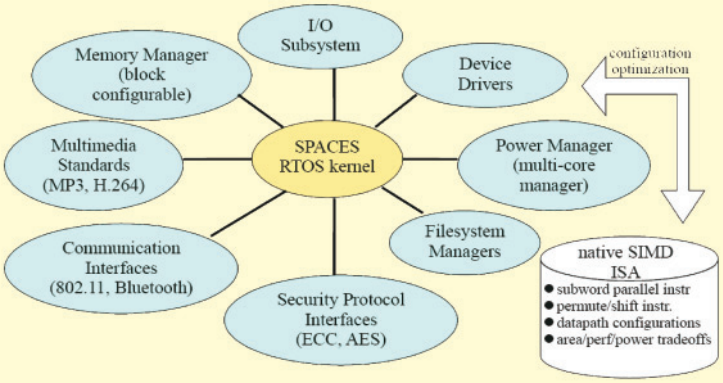
二、計畫緣由與目的

嵌入式系統經由硬體與軟體元件之間緊密結合可達成高效能與低功耗之特性。此類定製硬體與軟體元件的自動生成與評估工具之設計與研究是美國普林斯頓大學及麥迪遜威斯康辛大學所提出的計畫：「可展延低功耗之定製內嵌系統」(Scalable Power-Aware Custom Embedded Systems 簡稱 SPACES) 之設計的主要目標。與美方合作的我國團隊包含台大電子所陳少傑教授所提出的支援 H.264 視訊壓縮之定製低功耗晶片系統設計平台，以及本計畫所提出的設計與實作一個支援具有 subword-parallel 指令集之可組態內嵌處理器，如 PLX，之即時作業系統。

如圖一所示，PLX 是由普林斯頓大學電機系 Ruby Lee 教授所發展的小型具有 wordsize 可展延、平行次字彙 (subword-parallel，native SIMD) 指令集架構的微處理器。PLX 是第一個利用 subword parallelism、prefix



圖一　PLX Processor Framework



圖二　RTOS 的可配置模組化架構

execution 以及位元排列指令加速多媒體與資訊安全應用程式的微處理器。由於 PLX 擁有可組態 (configurable) 的特性，系統設計者可以根據目標應用的需求很容易地將 PLX 組態化。而雖然 PLX 有一個好的程式發展及架構探索環境，但是它仍缺一個實作平台以及一套可用於發展複雜多媒體與資訊安全應用的即時作業系統。

我們所設計的可組態即時作業系統和其他不同的 RTOS 相比，本計畫提出的即時作業系統在 PLX 架構下有很好的延展性 (scalability) 及可重新組態的能力 (reconfigurability)。因此，我們所提出的 RTOS 並無法完全以現今已存在的 RTOS 所取代。此作業系統針對可組態的處理器，如 PLX，取得最佳的效能。

三、研究內容與成果現況

本計畫設計的可組態即時嵌入式作業系統採用模組化設計策略如圖二所示，其中實作的微核心僅含一個基本的即時排程器負責高效能低功耗多重執行緒之間的溝通與排程。其他的模組可視需要動態時載入。我們主要參考 eCos、uCOS-II 及嵌入式 Linux 作業系統的實作，來設計我們自己的一套可組態即時作業系統，以下簡稱 RTOS。

我們實作作業系統的平台主要是台大所開發的平台，包含 PLX 組譯器、SystemC-based PLX 指令集模擬器及 PLX-SoC 模擬器。且因為台大並未實作 C 語言編譯器，所以我們必須將 C 語言的程式經由 LCC compiler 編譯成 PLX 組語的程式後，由台大的組譯器轉成台大的 PLX 模擬器可以模擬的機器碼。但是整合的過程中遇到 LCC compiler 與台大組譯器不相容等相關問題，也都需要額外去解決。

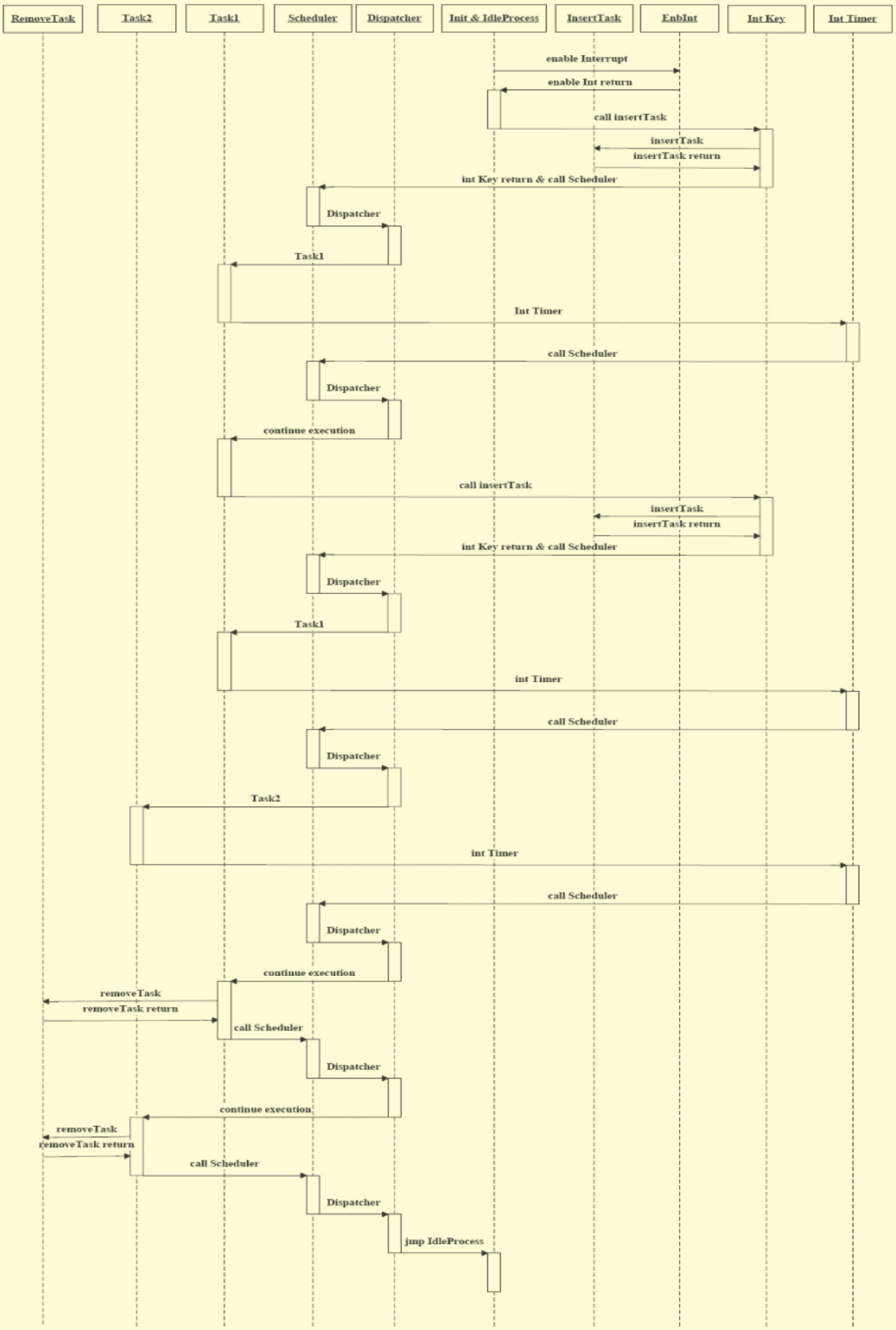
因為此計畫是預計與美方合作共三年，所以前一年多我們主要的重點除了建立實作平台之外，就是實作一個簡單的程序排程器、記憶體配置與管理的設計與實作、DCT/IDCT 轉換及 H.264 多媒體 codec 的移植。以下就分這四部份做詳細的介紹，我們所提的 RTOS 的其他部分如 power manager 等均需待往後一年多陸續開始設計與實作。

（一）即時嵌入式排程器之實作

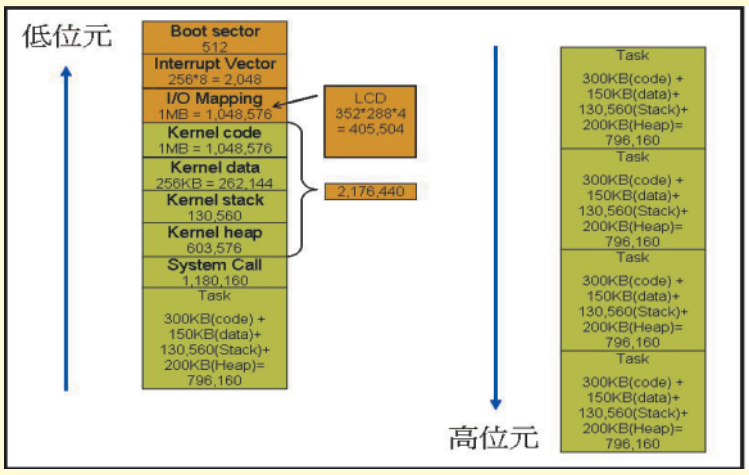
目前計畫中，是在基於 SystemC 的 PLX 指令集模擬器上使用 PLX 組語實作一套全新的優先權排程器，且以 round-robin 方式作為優先權排程的 backup policy。採用組語撰寫排程器的主要原因是希望做到更低階的控制，例如：context switching、interrupt service routine、task preemption 等。

在圖三中，我們使用順序圖 (sequence diagram) 表示兩個 tasks 之間的 preemption 及 round-robin 執行情況。以下為更詳細的執行流程。這個流程已經用 PLX 組語實作完成，並且已驗證無誤。

1. 初始化所有 registers 值。
2. Enable interrupt，執行 Idle Process(最低優先權)。
3. 利用 key interrupt 插入優先權較高的 Task1。
4. 呼叫 Scheduler 尋找最高優



圖三　Task Preemption and Round-Robin 執行流程



圖四　記憶體配置情況

先權的 task，將其 task address 插入 CurTask 中。

5. Dispatcher 載入 CurTask (Task1) 並執行。
6. Timer interrupt，呼叫 Scheduler 尋找下一個最高優先權 task，將其 address 載入 CurTask。
7. 目前只有一個最高優先權 Task1，由 Dispatcher 載入 CurTask (Task1) 並執行。
8. Task1 執行時，利用 key interrupt 插入和 Task1 同樣高優先權的 Task2，並儲存 Task1 的 context。
9. 呼叫 Scheduler，選擇最高優先權的 task 執行。
10. 每次 Timer interrupt，呼叫 Scheduler，儲存 context，依 Round-Robin 方式執行最高優先權的 tasks。
11. Task1 執行完畢，呼叫 RemoveTask 及 Scheduler。
12. Scheduler 選擇最高優先權 Task2，Dispatcher 載入執行。
13. Task2 執行完畢，呼叫 RemoveTask 及 Scheduler。
14. Scheduler 發現高優先權的 tasks 都執行完畢。
15. 跳回 Idle Process，等待新增 task。

（二）記憶體規劃與管理

目前記憶體規劃，主要是建構在 PLX 的指令集架構上。由於 PLX 並未提供 Memory Management Unit (MMU) 的支援，因此記憶體在管理上主要朝向無 MMU 支援的機制發展。

因為 PLX 架構缺乏 MMU 的支援，虛擬記憶體機制無法實作，task 須直接存取實體記憶體位址，並負責管理其取得的記憶體。這與 uC/OS-II 的記憶體管理機制頗為相符，因此我們參考 uC/OS-II 的管理機制並應用在我們的記憶體管理上。因為 PLX ISA 並未提供 IO instruction，因此我們必須切割一塊實體記憶體空間當成週邊 IO mapping 的部份，用以控制週邊設備。

根據台大 PLX 平台的設計有 8MB 的記憶體可以讓我們使用，因此針對目前的 PLX 架構所做的記憶體配置規劃如圖四所示。從低位元組開始，512 B 記憶體空間給 Boot Sector，1 MB 多的空間配置給 Interrupt Vector Table (IVT) 和 IO Mapping 等。剩下的部份則依序配置 OS 的 code size（即圖四中的 kernel code 及 kernel data 的部分）和 OS stack、heap 部分，及 OS 相關的 System call 部分。另外，剩下的部份再配置給 tasks 的程式碼、資料及 task 所需的 stack、heap 空間。而我們的 OS 也會另外提供相關的系統呼叫 (system call)，例如：mem_init()、mem_create()、mem_get()、mem_put() 及 mem_query() 等函式用來配置或釋放記憶體，並提供查詢記憶體目前使用的狀況。

（三）DCT/IDCT

為了使設計的 RTOS 應用於 H.264 視訊壓縮，我們實作了離散餘弦轉換（Discrete Cosine Transformation, DCT）以及 離散餘弦反轉換 (Inverse DCT, IDCT) 分別應用在 H.264

視訊編碼、解碼過程，並測試 4*4 的矩陣在經過 DCT 和 IDCT 計算後仍一致。

DCT 的功用為將「空間域數位影像資料」轉換成「頻率域」；反之 IDCT 便是將「頻率域數位影像資料」轉換成「空間域」。而在實作的過程因 PLX 指令集和 LCC compiler 並不支援浮點數的運算，因此將相關浮點數部份改為 fixed point 表示，再加以運算。

（四）H.264 Codec的移植

本計畫的最終目標是要實作一套即時嵌入式作業系統可以支援多媒體應用，例如視訊等。

影像（video）是高度仰賴 real-time 呈現的一種媒體型式，在播放時只要有任何的不順暢，人眼就很容易察覺出。影像利用人類視覺暫留的現象，在視覺保留的十六分之一秒內，出現下一張畫面已達到畫面連續。這個「畫面連續」需求，讓許多 real-time 的研究都想將它納入考慮。

而就目前的影像編碼方式來說，H.264/AVC 是一個熱門且值得研究的編 / 解碼方式，因此我們的作業系統也將 H.264 的支援納入考量。為了選擇適合的 H.264 IP，我們選用了 H.264 JM (Joint Model)，JM 未經最佳化的程式內容，對我們來說反而是種最精簡的 IP。在諸多的 JM 版本中，我們選用了 JM7.6 版來實作。原因是此版釋出的時間，也差不多是 H.264 整個協定達到完整的時刻；而在其後所釋出的 JM，則是加入了其他 H.264 增加部份（如藍光 DVD），這些增加部份對於我們的實作來說，反而增加系統設計的複雜度。

在實作過程中，由於 PLX 指令集不支援浮點數運算，而且負責將 C 語言編譯成 PLX 指令集的 LCC compiler 也不支援。所幸這些涉及浮點數運算的部份，大多是 H.264 JM 中負責統計的部份，因此將它修改 / 移除並不會造成系統上的影響。

我們接下來的目標，是讓系統能支援 H.264 同時編碼及解碼的應用，就像目前的 3G 影像電話一樣，可同時將影像編碼傳給對方，又同時接受對方的影像並解碼。

四、結論與應用範圍

本研究計畫針對 PLX 微處理器設計架構，實作一套可組態即時嵌入式作業系統。此作業系統根據不同的微處理器組態，做模組化的重組，以使其能有最佳與最緊密的軟硬體搭配。

而此模組化的設計策略，也使得當 PLX 被實作在一個可動態且部分重新組態（partially reconfigurable）的 SoC 上會獲得很大的應用優勢。因為計畫目前只執行一年多，故目前完成排程器的設計、記憶體空間的規劃、DCT/IDCT 轉換應用及 H.264 的程式碼移植。其他模組的設計與實作將在接下來的一年多內完成。

作者簡介

熊博安
國立中正大學資訊工程學系教授
國立台灣大學電機工程博士
專長：軟硬體共同設計、系統設計與正規驗證、即時嵌入式系統、晶片系統、平行與分散式系統
電話：(05) 2720411 轉 33119