九十七學年度全國大學校院

嵌入式系統設計競賽

參賽書面報告書

AndesCore + PACDSP
Multi-core Development System

組　　別：系統軟體組

報名編號：　177

中華民國 98 年 3 月 28日

注意：

1. 為維護比賽公平性，請勿於報告中洩露比賽隊伍之身分。
2. 為方便評比，敬請依此格式撰寫報告。
3. 最多以十五頁為上限 (不含封面)，其重點放在參賽作品之功能與設計構想、特色與設計創意、系統軟硬體資源效能運用、系統驗證完整性及可能應用等。
4. 報告須以中文撰寫。

目 錄

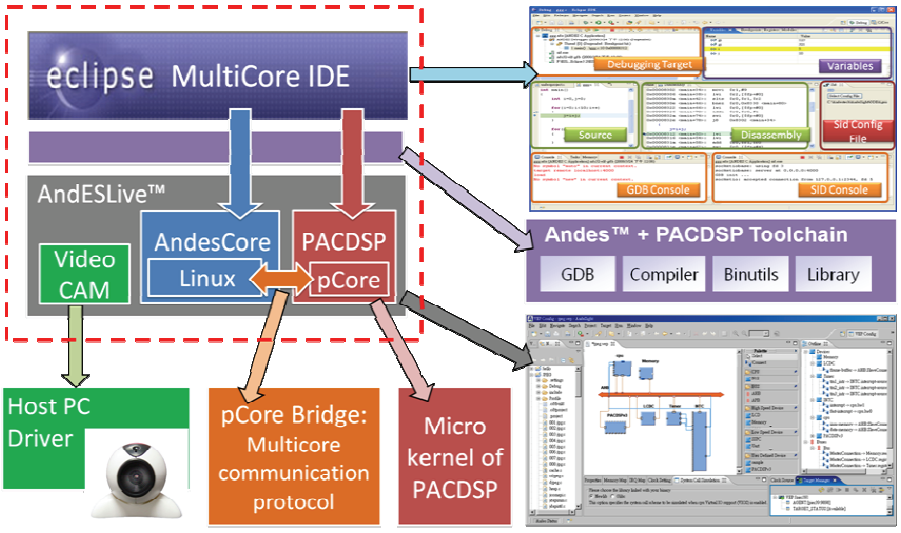
一、 摘要

隨著消費性數位產品在影音、遊戲、多媒體，以及無線通訊服務的發展與普及，產品推出的時程變的相當緊湊。為了瞭解架構上的配置與效能的差異，並且在架構開發初期提早進行應用軟體開發，我們需要有快速架構模擬平台工具，並能與軟體開發工具銜接，讓整體開發時間可以大幅縮減。

本次參賽作品將以晶心科技公司所提供之 ESL 工具「AndESLive™」為基礎，架構多核心的模擬環境(AndesCore + PACDSP*4)，其中本次比賽加入的模擬 IP 包括：Video Cam、Interrupt Control 及 PACDSP Core，應用程式包括：Video Cam, Interrupt program, and partitioned JPEG。並且預計移植 pCore 微核心到 PACDSP Core；在 AndesCore 上的 Linux 作業系統增加 PACDSP driver 並且架構 pCore Bridge 中介層與 pCore 進行資料傳遞。

另外再以 Eclipse™ 為基礎的 IDE 環境整合 Andes Development Toolkits 和 PACDSP Development Toolkits，提供一個多核心虛擬共通平台暨偵錯開發環境，可以開發多核心程式，控制並取得 AndESLive™上多顆處理器 IP 的模擬資料。

二、 作品功能與構想



本參賽作品為 AndesCore + PACDSP 之多核心開發環境，提供多核心 IDE, 多核心模擬器,視覺 I/O, 及作業系統層級溝通規範。作品的概念是要協助使用者能夠快速架構多核心模擬平台，並且有一個 IDE 可以開發程式、偵錯並操作多核心模擬環境，在本作品中的主要承襲之成果包括：

- Andes Core IP 及 AndESLive™ 模擬環境技術來自晶心科技
- Andes Compiler & GDB & Binutils 技術來自晶心科技

- PACDSP Core IP 模擬核心技術來自工研院晶片中心
- PACDSP Compiler & GDB & Binutils 技術來自相關科專計畫
- PACDSP pCore & pCore Bridge 技術來自相關科專計畫

而新增加以下功能，來完成本參賽作品：

- 新增 VideoCAM 模擬 IP 到 AndESLive™ 環境,可以進行立即影像輸出。
- 完成修改 PACDSP 單機執行模擬器變成相容 AndESLive™ 的多核心 PACDSP Core IP，包括：增加 Interrupt、增加記憶體架構、增加匯流排介面及修改模擬器程式架構。
- 完成 GDB 與 PACDSP Core IP 的溝通控制。
- 建構以 Eclipse 為基礎的多核心 IDE 開發工具介面，底層連結 Andes 及 PACDSP 各自的編譯器工具及 GDB。而 GDB 則控制 AndESLive™ 上的處理器 IP。
- 完成移植 pCore(DSP 微核心)及實現 pCore Bridge 多核心溝通介面至 AndESLive™ 環境。
- 完成展示多核心軟體開發及應用程式。

三、 系統軟硬體平台簡述與資源運用分析

以下簡述使用的軟體平台及技術，包括：AndeSight™ 及AndESLive™ 、PACDSP IP、Eclipse ™ IDE環境、及PACDSP微核心-pCore。

A. AndeSight™ & AndEsLive™

AndeSight™ 及AndESLive™ 是由晶心科技(Andes)所提供的整合開發環境，這套開發環境使硬體工程師和軟體工程師具有一樣的能力去製作和修改他們各自的系統模型。而且實現了應用軟體和硬體的同步開發，因而縮短了整合時間，此一虛擬平台加速開發進程並驗證用戶的使用體驗，進而大大降低計畫成本及提高產品研發之成功機率。此開發工具採用圖形用戶介面。該工具使用戶可以從一個包括Andes處理器和必要週邊設備(例如LCD控制器、UART、GPIO、I2C…等) 的資料庫中選擇需要的元件，因而快速製作和配置完整的虛擬模型。Tool chain包括編譯器、除錯器、組譯器、連結器以及除錯及最佳化的流程也使開發者受益匪淺。 這使得特定組件的軟體可以在不同抽象等級的模型中被執行和測試。 在一個通用的除錯環境下，同一個軟體還可以在FPGA原型以及最終晶片上的進行測試。

B. PACDSP IP

PACDSP為新一代的高效能數位訊號處理器，以VLIW和SIMD Instruction Set支援高平行度的運算需求。與Super Scale Processor相比，VLIW Processor降低了Multi-Issue架構在功率消耗上的傷害。而其Low Code Density 的問題，PACDSP則以Variable Instruction/Packet Length來加以處理。PACDSP 將Register File做適當的切割與分類，以較小的面積與Port數降低資料存取的功率消耗。除此之外也運用了特殊的定址方式減少Register File間的資料搬運，更進一步降低了處理器內部因資料搬運造成的功率消耗。

C. Eclipse™ IDE環境

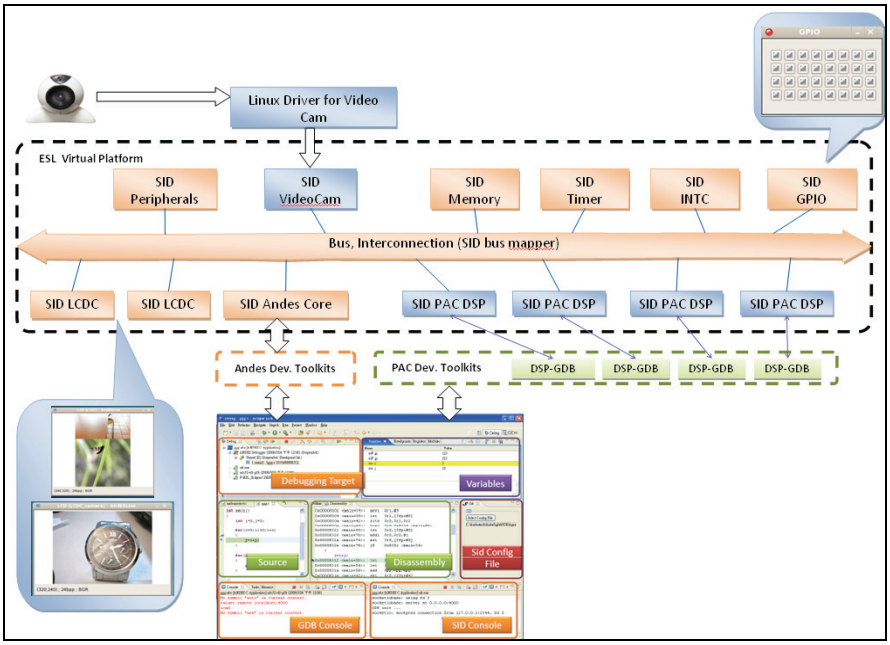
Eclipse是一個以模組為發展單位的open source 整合開發的GUI環境，支援多種不同的語言如C/C++、Cobol、Java等。在Eclipse platform裡，以plug-in方式的架構作為開發Eclipse的基本單位，使Eclipse整體架構較為簡單，容易維護與發展高效能的模組。為了實現雙處理器平台上的偵錯，我們利用單獨的偵錯器來對應不同的處理器，同時在每個不同的區塊之間，以不同的溝通介面連結，使系統達到整合的目的。在此IDE(Integrated Development Environment) 中整合 Andes Development Toolkits 和 PACDSP Andes Development Toolkits的部份，藉由一套定義好的tool chain格式，我們可以將各種tool chain整合在IDE上，包括Loader、Debugger、Assembler、Compiler、Linker，讓使用者可以透過此介面輕易的進行多核心應用程式的開發。

D. PACDSP微核心-pCore

PAC 使用了微核心系統 (pCore v3.0）和用來跟 AndesCore 溝通的 IPC library，再搭配 PACC 編譯器與 Binutils。pCore 所提供的 API 使我們能輕易的讓 AndesCore 與 PAC 溝通，與控制 PAC 端的程式碼。

四、 系統實作內容

A. 系統架構

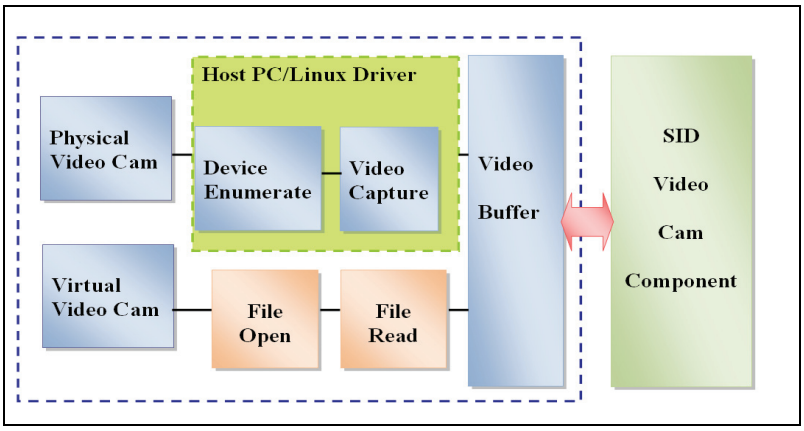


透過ESL整合工具來發展模擬多核心環境，將Video Cam元件和PACDSP元件加入此模擬環境，並放入原本Andes所提供之元件(Andes Core, Memory, LCDC, Bus)，而不同核心之間的溝通方式可以透過Shared Memory或者Interrupt來溝通，最後使用LCD元件來顯示應用軟體之結果，而在IDE中整合了Andes Development Toolkits和PACDSP Andes Development Toolkits，提供多核心虛擬共通平台暨偵錯開發環境，上圖為此競賽中所模擬的實做系統架構圖。

B. Multi-core PACDSP IP on AndESLive™

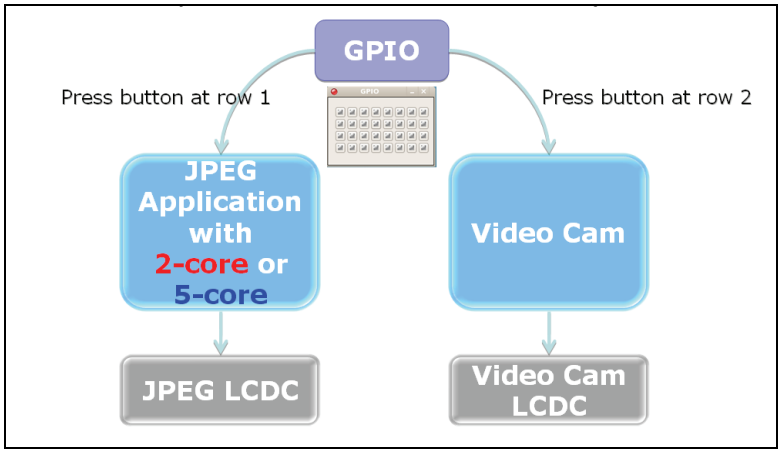
我們首先取得ITRI STC的PACDSP模擬器，該模擬器接受編譯後的執行檔然後獨立執行。但在AndESLive™模擬環境中，每一個要模擬的硬體都是一個獨立的元件(component)，而模擬的環境透過一個configuration檔案來表示描述架構，在這描述檔中會包括所要使用到的元件(CPU, Memory, Bus, INTC, Timer…)，和每一個元件之間的關係，包括PIN腳的連接或BUS的連接等等相對應的關係。而原本PACDSP模擬器的記憶體存取方式與匯流排介面都不符合AndESLive™所需。因此我們做了增加記憶體架構、增加Interrupt、增加匯流排介面及修改模擬器程式架構來完成多核心PACDSP IP。

C. Video Cam IP on AndESLive™



首先要模擬的硬體是Video Cam元件，上圖為此元件之架構圖，一開始先透過在Host PC/Linux中安裝Video Cam的Driver，連結到實體的Video Cam，然後在透過Buffer的方式讓Virtual Video Cam抓到實體Video Cam所截取的Video，並在此平台中提供虛擬平台與實體Video Cam的連結，最後讓此Video顯示在模擬的LCD元件中。

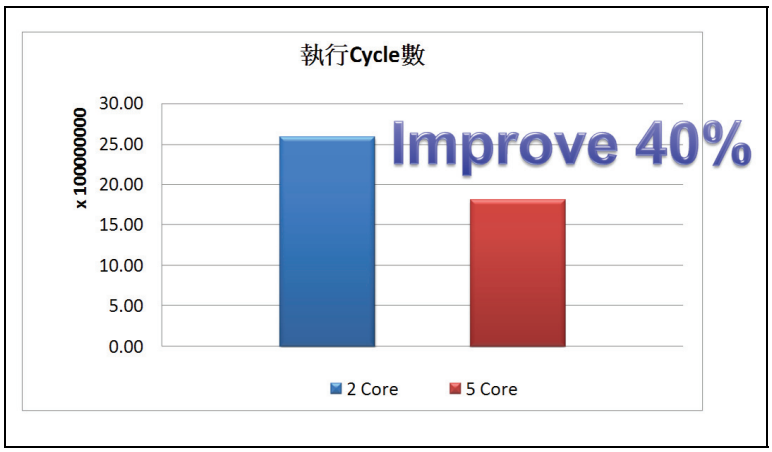
D. 多核心應用程式實做



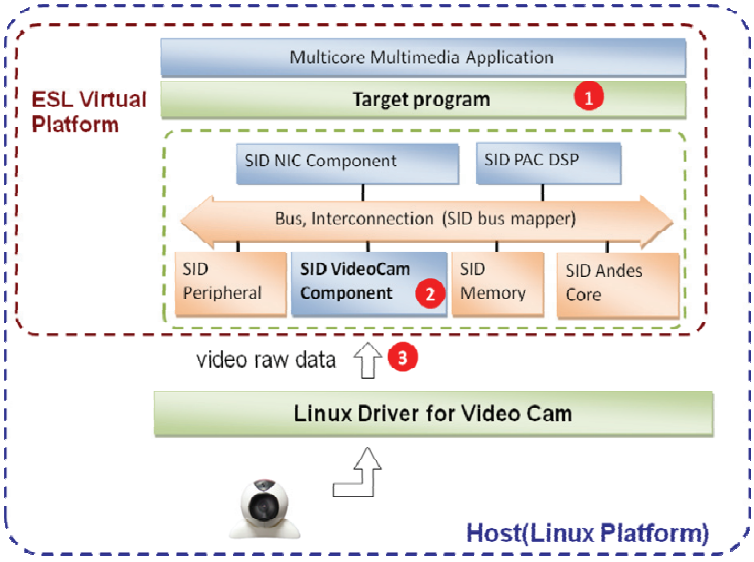
上圖為我們在此架構中所要展現的多媒體應用程式，這個應用程式包括兩個主要的程式，一個是JPEG，一個是Video Cam，使用者可以透過GPIO按鈕來控制程式執行的Task，當按下GPIO第一排按鈕的時候，Task會切換到JPEG應用程式(本競賽展示的是5顆Core版本)，而當按下第二排按紐的時候，Task會切換到Video Cam做截取畫面的動作，以下針對兩個應用程式做更詳細的介紹。

在JPEG部份，我們有實做兩顆Core(1*AndesCore+1*PACDSP)版本及五顆Core(1*AndesCore+4*PACDSP)版本，在兩顆Core的部份，在每次處理的

六個8*8的Block中，將其中4個丟給PACDSP去做IDCT運算，而剩下兩個交給AndesCore做運算；而五顆Core版本中，則是將四個Block平均分配到四顆PACDSP上，也就是一顆PACDSP處理一個Block，而剩下兩個Block一樣交給AndesCore來處理。接著透過我們系統所提供的Profile功能，來分析五顆Core版本和兩顆Core版本效能上的差異，下圖為兩個版本的執行Cycle數，可以看到五顆Core版本所花的指令數相對於兩顆Core版本少了40%，也就是增進了40%的效能。



有了JPEG應用程式後，為了展示我們平台適用軟體開發的優點，因此我們加入了Video Cam到此平台中，並同時展示兩個應用程式一起執行的情況，當我們提供Video Cam的模擬元件後，使用者只需要在架構描述檔上多增加Video Cam的描述，就可以撰寫程式來控制模擬的Video Cam，以下是Video Cam程式架構:



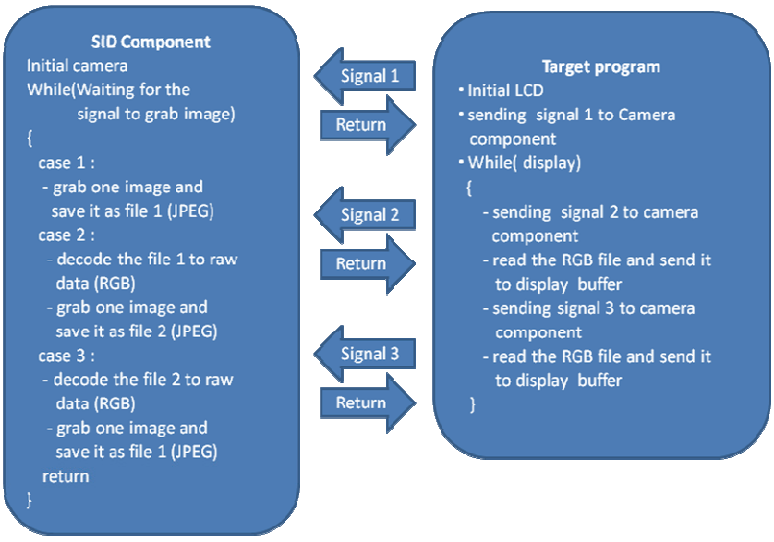
(1) 1這個部份可以看成是在ESL虛擬平台上跑的一支程式，主要的動作有接受使用者的輸入而產生相對應的反應，比如輸入s來開始截取影

像的動作。當Target program得知使用者輸入s以後，他會Trigger SID Video Cam component (2號)。

(2) 當SID Video Cam component接受到由Target program傳來的訊號時，他會產生相對應的反應。比如開始截取影像，則他會啟動(3號) 也就在Host端的Linux camera driver去截取影像，然後通知Target program說影像已準備完程，可以去Image buffer裡把他拿到LCDC上去顯示。

(3) 3這個部份其實是Camera在Linux上的Driver，SID component 可以透過這個Driver連結到實體Camera去截取影像。

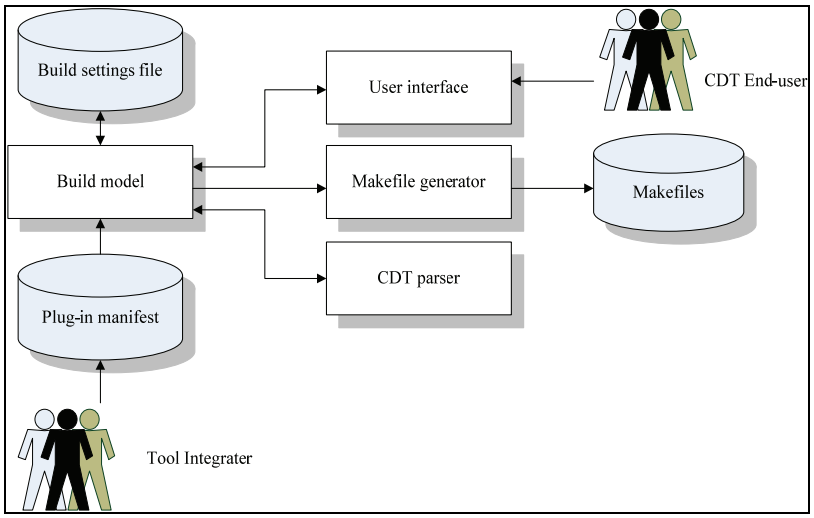
以下是程式流程 ：



E. 多核心虛擬共通平台暨偵錯開發環境

Eclipse 是一個以模組為發展單位的 open source 整合開發環境，支援多種不同的語言如 C/C++、Cobol、Java 等。在 Eclipse platform 裡，以 plug-in 方式的架構作為開發 Eclipse 的基本單位，使 Eclipse 整體架構較為簡單，容易維護與發展高效能的模組。藉由 CDT(C Development Tool)的支援，Eclipse platform 得以完成 C/C++ 程式語言的支援。在 CDT 的架構中，managed build system 乃是與程式編組譯流程控制相關的重要的一環。

Managed build system 包含了許多基本的元件，在這些元件的共同運作下完成整個應用程式的編組譯動作。在 managed build system 中包含了一個核心的 build model，如附圖六所示，build model 連結了三個 internal client：使用者介面(user interface)、makefile 產生器(makefile generator)、CDT 剖析器(CDT parser)。Makefile generator 能夠產生適用於 compiler 的 makefile，藉由使用者對編組譯流程選項的選擇，build model 能夠藉由 external client 所提出的 build setting files 及 plug-in manifest 來產生正確的建構流程。



Managed build system 概觀圖

除錯器(debugger)在整合發展環境裡扮演相當重要的角色，它提供一個控制系統的方式，讓使用者能夠依據需求來操控系統，進而從系統獲得所需要的資訊。舉例來說，使用者能夠指定讓系統執行一個指令並且停止下來，接著使用者便能觀察目前系統的暫存器(registers)內容，甚至可以更動暫存器的內容。藉由除錯器所提供控制系統的方法，使用能從得到的資訊判斷程式是否按照使用者的需求來執行，使執行結果錯誤的程式能夠有效的縮小可能的錯誤範圍，提供使用者具體的方向來解決程式或系統的問題。

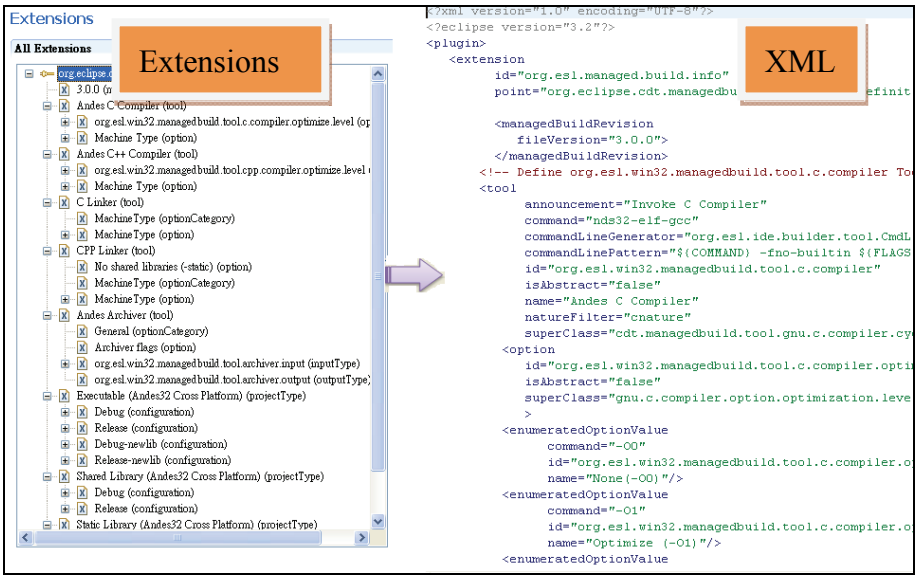
以目前最廣為流傳的除錯器 GNU debugger (GDB)為例，為了使除錯器的發展能保有最大的可移植性，GDB 將 debugger 劃分為兩部分：符號端(Symbol side)及目標端(Target side)。在 GDB 中，目標向量(Target Vector)負責的處理的部分界於上述兩者之間。其發展概念類似於物件導向程式中的物件，利用相同對物件的操控方法，重覆利用相同的上層程式碼，簡化開發時抽換底層機制的複雜度。一個目標向量包含有，與目標平台相同暫存器數量大小的空間，負責傳輸暫存器的內容及儲存其內容；反組譯(Disassmble)硬體機械碼，提供解譯目標平台上應用程式內容的方法；除此之外，並負責控管中斷點(Breakpoint)及監視點(Watchpoint)的數量與使用情形。

利用這樣的架構劃分，使 GDB 能夠同時兼備有對多重語言的支援與多重平台的支援，讓 GDB 在不同平台的移植上更便利。針對嵌入式系統(embedded system)上的除錯，為了讓 GDB 能夠被使用在嵌入式系統的除錯，GDB 提出了遠端串列傳輸協定(Remote Serial Protocal, RSP)，利用精簡化的單一指令封包，控制接在本端機器上的遠端硬體。在遠端串列傳輸協定裡，所有的傳輸內容都利用 ASCII 碼來做為傳輸格式。在協定的設計上，

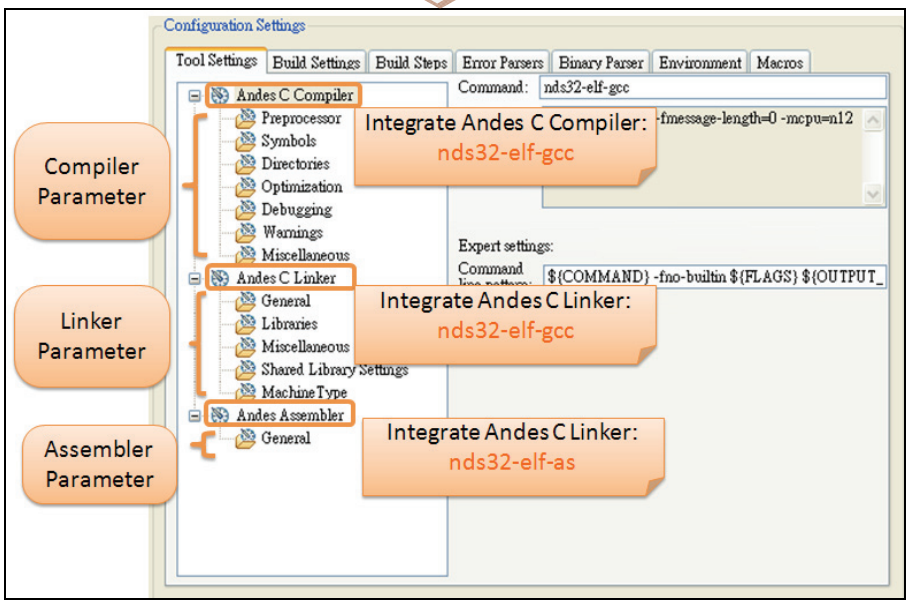
考慮了利用串列傳輸線傳輸封包時，資訊誤傳的可能性。

為了實現雙處理器平台上的偵錯，將利用單獨的偵錯器來對應不同的處理器，同時在每個不同的區塊之間，以不同的溝通介面連結，使系統達到整合的目的。

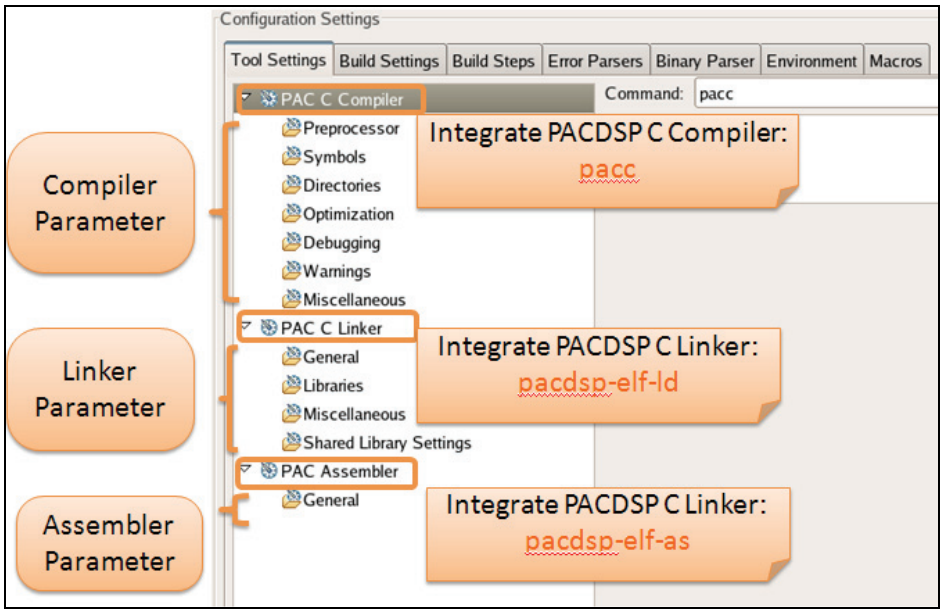
實作內容，在整合 Andes Tool-Chain 及 PACDSP Tool-Chain 的部份，我們利用 CDT plugin 上既有的延伸點 (extension point): “org.eclipse.cdt.managedbuilder.core.buildDefinitions” 來做延伸，將我們既有的 tool-chain: ANDES 跟 PACDSP 整合在我們的 IDE 上，這部份主要透過修改 XML 檔來達成如下圖所示。



integrate extension point and corresponding XML file

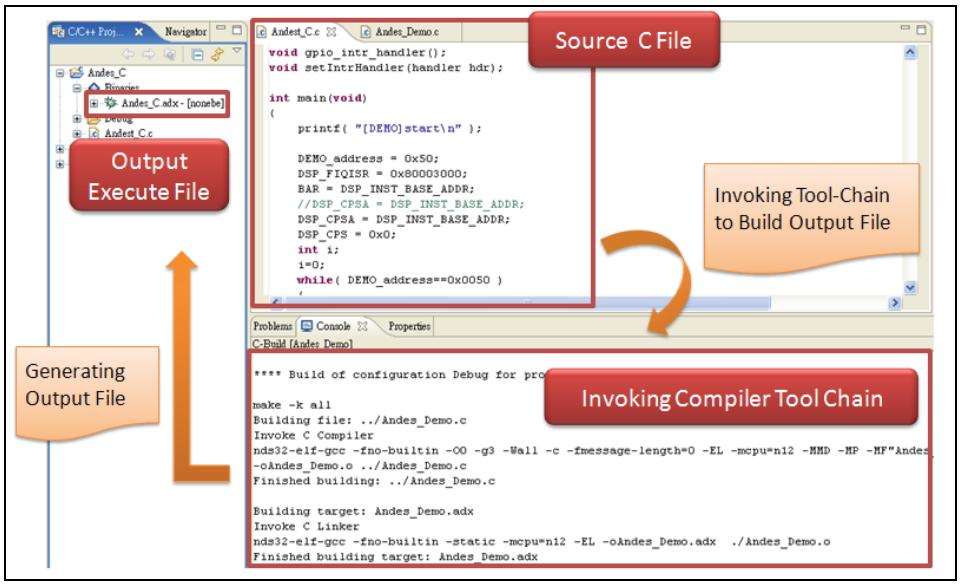


integrating andes tool view



integrating pacdsp tool view

藉由一套定義好的 tool chain 格式，我們可以將各種 tool chain 整合在 ide 上，最後顯示的界面如上面兩張圖所示，而最後 tool-chain 執行起來編譯的過程如下圖所示，我們呼叫了 andes 的 tool-chain，完整的編譯出一個可執行的 binary file 出來。



編譯過程

F. 系統驗證

在系統驗證部份，我們採用了C90做PACDSP元件的測試，我們使用了兩種參數：-O0和-O1，在733個測試程式中，全部都通過測試，通過率達100%；我們也測試記憶體讀寫的功能，包括了AndesCore存取Simulator Share Memory和PACDSP Local Memory，還有PACDSP存取Simulator Share Memory和PACDSP Local Memory。

在Interrupt部份，我們也做了從AndesCore發Interrupt給PACDSP和從PACDSP發Interrupt給AndesCore的功能性測試；另外在Simulator中的元件，包括LCDC和GPIO也都測試了其功能性。

五、 預期功能與目標

此競賽預期的目標是快速提供模擬平台，並適合於應用軟體開發；只要我們提供足夠的模擬元件，使用者確定系統架構後，只要撰寫架構描述檔就可以快速提供模擬平台；我們同時也在軟體開發部份，提供所需tool chain，包括Loader、Debugger、Assembler、Compiler、Linker，甚至是在需要OS的情況下提供pCore及pCore Bridge，方便使用者撰寫應用軟體。

此競賽中預期的結果將展現整體模擬平台外，並利用應用多核心應用程式來展示整個系統的模擬結果，包括JPEG和Video Cam，加上IDE整合多核心開發環境，提供使用者可以同時開發多核心程式的環境，達到我們所預期的快速提供模擬平台並適合於應用軟體開發。

六、 設計創意

在我們這套系統中，最重要的設計是使用者可以藉由提供的多核心 IDE 以單一介面開發多核心應用程式。使用者不但可以開發獨立程式，也可以開發與作業系統相依的應用程式，包括 MPU 作業系統與 DSP 作業系統。我們更提供一個快速調整架構的多核心模擬器，只要透過系統描述檔案來描述想要架構的系統，即可馬上得到完整的模擬環境。使用者如果想要對模擬的硬體做參數的微調，藉此觀察不同參數對效能所造成的影響，也只要透過此系統描述檔案的設定就可以得到最適合此使用者的硬體規格。

七、 可能遭遇之困難點及解決方法

我們在此系統中所遇到困難是由於 AndESLive™目前在多核心溝通同步的設計，模擬器執行的概念是每一個 step 訊號，CPU 就自己執行使用者設定的 cycle 數。當我們以單顆實體 CPU 模擬五顆 CPU 的時候，程式其實是以 sequential 依序執行，所以如果五顆 CPU 都將此 Step cycle 設定一個很大的值的時候，此方法會加速模擬效率，但五顆核心之間的溝通點可能會比較鬆散。

如果使用者需要很精準的同步時，為了解決此問題，只要將五顆 CPU 每個 step 執行多少 cycle 的參數都設到最小，就可以得到正確的溝通同步，但是模擬

的速度就會相對變慢。

所以當使用者操作多核心開發環境時，可以依照所專注的項目進行架構微調，以得到正確的資訊。

八、 結論

此競賽預期的目標是提供多核心開發環境，適合於多核心應用軟體開發。我們提供足夠的模擬元件，使用者只要確定系統架構後，撰寫架構描述檔就可以快速得到模擬平台。在軟體開發部份，提供所需 tool chain，包括 Loader、Debugger、Assembler、Compiler、Linker，甚至是在需要 OS 的情況下提供 pCore 及 pCore Bridge，方便使用者撰寫應用軟體。此競賽中所呈現的結果將展現整體模擬平台外，並利用多核心應用程式來展示整個系統的模擬結果，加上 IDE 整合多核心開發環境。