

# 2014 全國大學院校 IC 設計競賽

## F 組 -- 電子系統層級設計(ESL)組

### 參考題型

為鼓勵國內大學院校學生從事系統層級設計研究，自 2014 年起 IC 設計競賽將新增系統層級設計組別 (F 組)。為了讓預計參加本組比賽的同學，能更進一步了解指定的工具軟體，及必須具備的操作知識，因此提供此一參考題型作為入門指引。

參考題型主要是以簡易的 ALU 範例來說明如何使用此次比賽指定的兩套工具軟體，即 Synopsys Platform Architect (PA)及 ARM DS-5。本範例包含 PA 及 DS-5 的基本操作，包括 PA 元件庫的建立、掛載元件至虛擬平台、DS-5/PA/Verilog 模擬器的軟硬體共同模擬等等。相關的技術細節將另於課程中加以說明。

關於 PA 系統架構分析功能之運用，並未列入此參考題型之操作練習，但正式比賽題目內可能會涵蓋此部分。請同學參考以下所列 PA 軟體內建之說明文件：

1. VPEXplorer Debug and Analysis User's Guide (PA\_VPEXplorer.pdf)
2. Virtual Prototype Analyzer and Manager User's Guide (PA\_VPAnalyzerGuide.pdf)

本文件內容分為題目說明、參考解答、參考解答操作指引、以及附錄等四個章節。

下載說明檔：

PA\_DS5\_ALU\_vxx.tar.gz - 執行參考題型模擬所需資料檔。

內含以下說明文件

Document\_vxx.pdf - 本文件。(xx:版本編號)

對本文件內敘述之設定與操作步驟有任何疑問，或需要進一步技術資訊，請聯絡 CIC 客服熱線並告知詢問 IC 設計競賽 F 組相關問題，CIC 客服人員會轉達您的問題給相關負責同仁。

TEL: (03)5773693 ext.886

E-mail: service@narlabs.org.tw

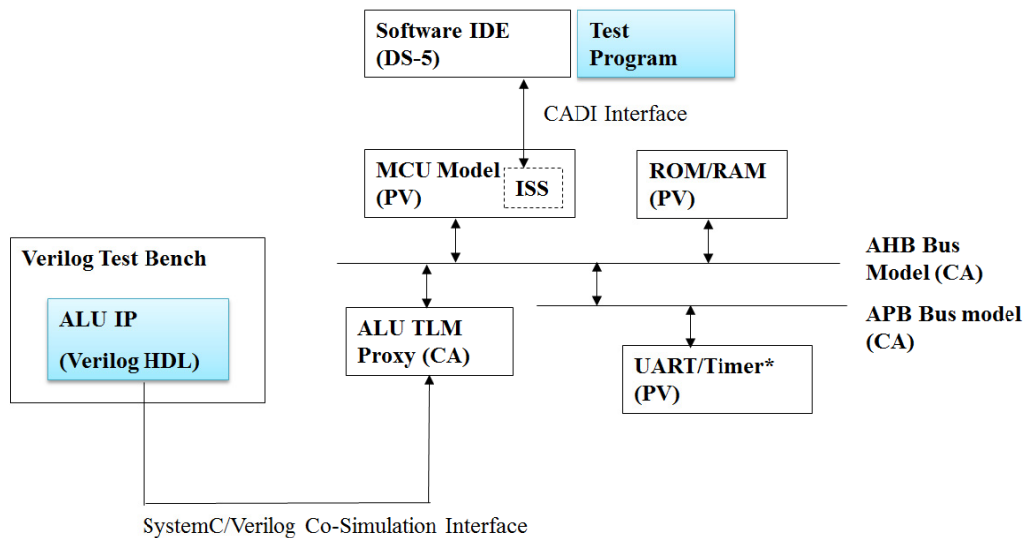
## 版 本 經 歷

版本	日期(年/月/日)	經 歷
1.0	2014/01/13	
1.01	2014/01/15	虛擬系統平台重組，並在 CentOS 6.5 VM 測試完成
1.02	2014/01/17	測試發現 ModelSim 相容性問題，從系統環境需求清單上移除
1.03	2014/01/23	將說明文件包裹於資料檔內 解決處理器模型在非執行模式時，DS-5 無法以 CADI 連接的問題
1.04	2014/01/28	參考解答操作指引加入“建立虛擬平台”一節 對參考解答操作指引之章節結構進行調整；內容敘述修訂 增加“dual_timer_test.axf”測試程式二元碼檔案
1.05	2014/02/19	修改 ARM Cortex M4 模型之 license 使用狀態 增加 ALU TLM 模型於 library/ALU_TLM/目錄 附錄追加“刪除不正常程序的方式”操作指引 附錄追加“啟動 PA 分析功能”練習

# 1. 題目說明

參賽者在限定時間內需完成符合規格之 ALU IP，並整合至主辦單位所提供之虛擬系統平台，完成電子系統層級(ESL)之軟硬體整合與驗證。

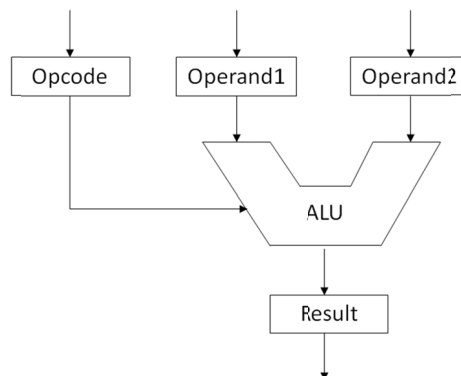
在此簡易參考範例中，虛擬系統平台規格如下圖所示。



\* 週邊元件在此範例並未使用

參賽者須分別完成下列 2 項硬體設計與軟體開發：

1. 以 Verilog 硬體描述語言設計一具備加法、減法、乘法功能之簡易算術邏輯單元(Arithmetic Logic Unit, ALU)，並整合至一具備處理器、記憶體及匯流排之 ARM-based 虛擬平台。ALU 架構圖如下：



2. 請撰寫一個在 ARM Cortex M 處理器上執行之程式，測試 ALU 運算之正確性。運算結果將輸出至 Platform Architect 軟體之操控台(console)視窗上。

## 1.1 硬體規格說明：簡易 ALU

ALU 具備以下輸入變數：

Opcode - 2 位元無號整數。

ALU 具備以下輸出變數：

Operand1, Operand2 - 均為 32 位元有號整數。

Result - 32 位元有號整數。

ALU 運算定義：

Opcode = 0            Result = Operand1 + Operand2

Opcode = 1            Result = Operand1 - Operand2

Opcode = 2            Result = Operand1 \* Operand2

Opcode = 3            Result = Operand2

ALU IP 相關暫存器之 Memory Map offset:

Opcode[1:0] :        0x0000

Operand1[31:0] :    0x0004

Operand2[31:0]:    0x0008

Result[31:0]:       0x000C

Opcode、Operand1、及 Operand2 之任何改變，應於下個 clock cycle 反映於 Result 值之改變。

## 1.2 軟體規格說明：簡易 ALU 測試程式

撰寫以 ARM DS-5 所提供的 Cortex M4 軟體框架(含開機設定)為基礎，透過讀寫 ALU 暫存器以進行功能測試的簡易程式。

測試程式參考架構：

```
for (i=0; i<m; i++)  
    for (j=0; j<n; j++)  
    {  
        set Opcode;  
        set Operand1 = i;  
        set Operand2 = j;  
        get Result;
```

}

參賽者僅需將測試內容撰寫成 C 函式，並於 Cortex M4 軟體框架中，main() 函式內的適當位置加以呼叫，即可完成測試程式，而不須撰寫或更動與 Cortex M4 架構相關的設定程式碼。

## 2. 參考解答

### 2.1 參考解答說明

本參考解答以參賽者已設計完成 ALU Verilog 程式碼(含 AMBA wrapper)為前提，引導參賽者如何利用 PA 依序進行以下步驟：

1. 將 ALU Verilog 程式碼包裝為虛擬平台內使用之元件。
2. 整合至已建立完成之虛擬平台。
3. 共同操作 PA、DS-5 及 INCISIV/ModelSim/VCS 其中任一款模擬器，進行軟體協同模擬。

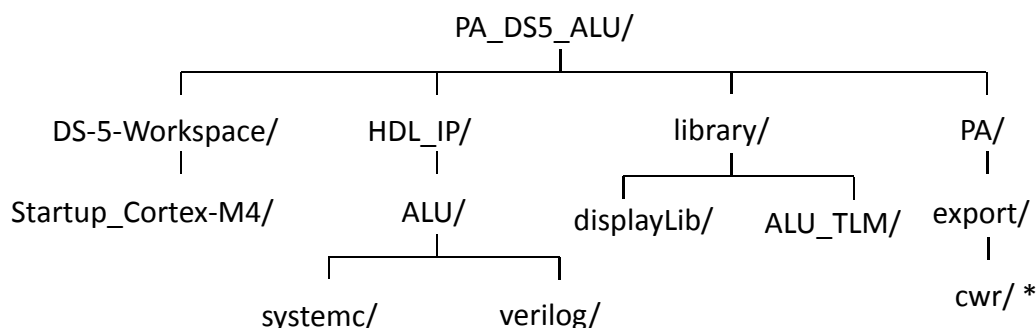
在此參考解答中，主辦單位已提供實作步驟所需之硬體與軟體原始碼，以方便讓參賽同學熟悉整個電子系統層級(ESL)之軟硬體整合與驗證流程。未來正式考題競賽時將不會給予參考設計，僅提供完成題目所必需之檔案或資料。因此除了需要自行動手完成的部分，也請同學注意此參考解答內已提供，但在比賽中可能需要自己撰寫或修改的程式碼或設定，例如虛擬平台整體及其中各別元件之連接方式與參數設定、ALU Verilog 程式碼中 AMBA wrapper 的撰寫格式、及 DS-5 軟體專案內程式碼的架構與變數宣告與設定方式等等。

### 2.2 參考解答安裝

A. 開啟 Terminal 程式，在範例放置目錄鍵入指令

tar zxvf PA\_DS5\_ALU\_v xx.tar.gz (xx:版本編號)，解開後得到 PA\_DS5\_ALU/ 目錄。

B. 範例的目錄結構如下：



**DS-5-Workspace/:** 放置測試軟體的目錄。

**HDL\_IP/:** 放置 ALU Verilog 程式碼的目錄。

**Library/:** 放置非 PA 內建元件的目錄。

**PA/**: 放置虛擬平台的目錄。

\* 雖然執行過模擬之後，各目錄底下都可能產生 cwr/ 目錄，但是 PA\_DS5\_ALU/PA/export/cwr/ 是放置了模擬參數設定的重要目錄，若刪除需要重新設定，請參考 3.2 節步驟 J 之敘述。

C. **PA\_DS5\_ALU/** 目錄內有兩個環境設定檔，pa\_setup.csh 及 run.csh。

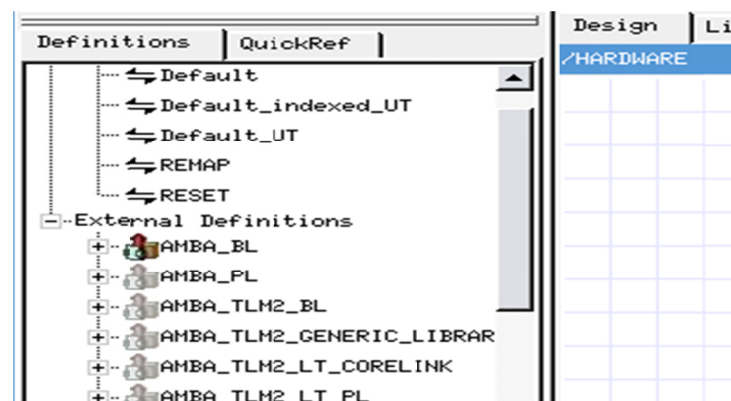
其中檔案 pa\_setup.csh 內容同於 PA 安裝目錄的 PlatformArchitec.csh，請針對下列兩項參數修改以符合您電腦的軟體安裝環境：

- PA 安裝路徑，即參數 SNPS\_VP\_HOME，以及
- HDL 模擬器環境參數，即 XX\_INSTALLED\_AT (XX 為 VCS/NC/MTI，擇一即可)，及下一行對應的軟體環境參數設定檔所在目錄，並將前方遮蔽符號 # 移除。

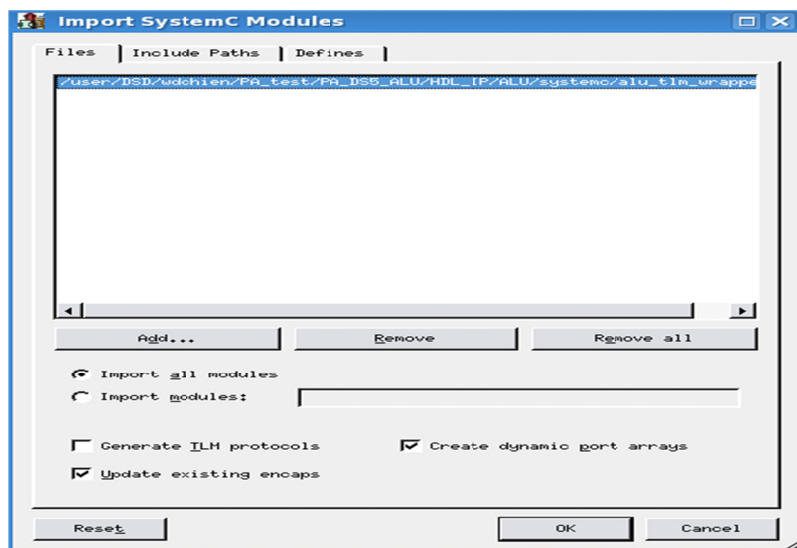
### 3. 參考解答操作指引

#### 3.1 將 ALU Verilog 程式碼包裝為 PA 元件庫

- A. 開啟 Terminal 視窗。在 PA\_DS5\_ALU/ 目錄執行指令 “source pa\_setup.csh” 設定環境參數。
- B. 切換目錄至 PA\_DS5\_ALU/HDL\_IP/ALU/。
- C. 執行指令“pct&”以啟動 Platform Creator (pct) 圖形介面環境。
- D. 在 pct 左側 “Definitions” 欄位的樹狀圖，點選打開 “External Definition” 及其下的”AMBA\_BL”(圖標會由灰轉彩色)。

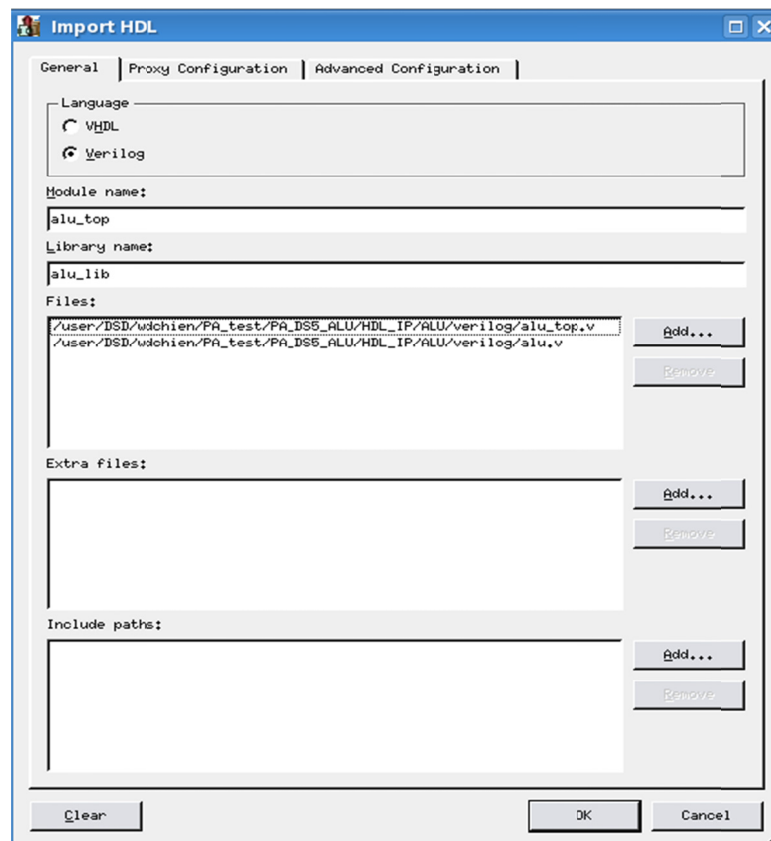


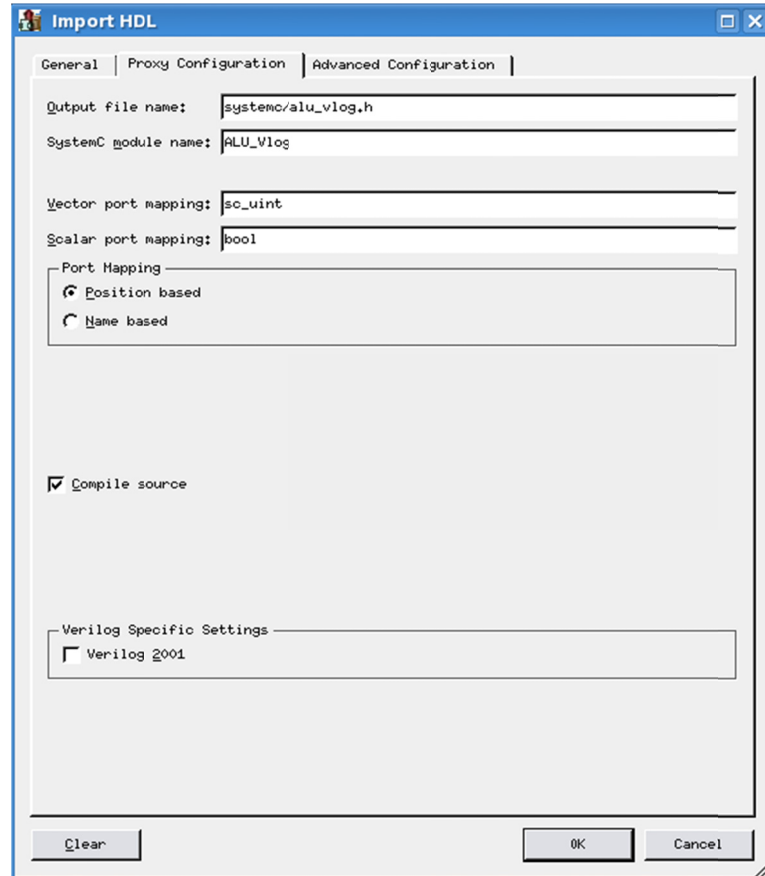
- E. 從系統選單執行 “Project -> Import SystemC Modules”, 叫出 SystemC 匯入視窗。在 “Files” 欄位，點按 “add” 鍵，加入檔案 PA\_DS5\_ALU/HDL\_IP/ALU/systemc/alu\_tlm\_wrapper.h。在 “Include Paths” 欄位，點按 “add” 鍵加入目錄 PA\_DS5\_ALU/HDL\_IP/ALU/systemc/。完成之後點選 “OK” 功能鍵。匯入成功會顯示訊息 “SystemC code import successfully finished.”。





- F. 此時在 “Definitions”欄會出現“ALU\_TLM\_Wrapper”圖標。點選並按滑鼠右鍵選“Rename”改成“ALU”。
- G. 從系統選單執行 “Project -> Import HDL”，叫出 HDL 匯入視窗。  
在”General”功能表進行以下設定：  
Language -> Verilog、  
Module name -> alu\_top、  
Library name -> alu\_lib、  
Files -> 加入 PA\_DS5\_ALU/HDL\_IP/ALU/verilog/alu.v 及 PA\_DS5\_ALU/HDL\_IP/ALU/verilog/alu\_top.v。  
在”Proxy Configuration”功能表進行以下設定：  
Output file name -> “systemc/alu\_vlog.h”、  
SystemC module name -> “ALU\_Vlog”、  
Compile source -> 勾選、  
其他設定不作變動。

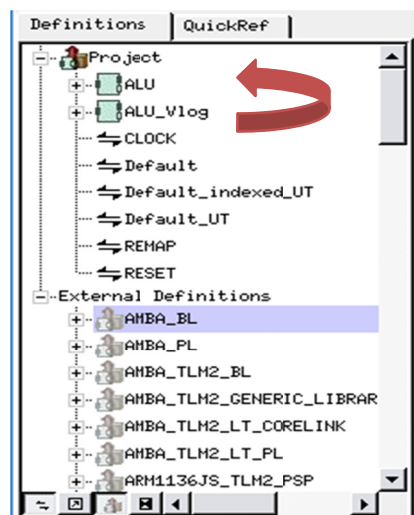




點選“OK”，匯入完成後“Definitions”欄會出現“ALU\_Vlog”圖標。

點選“Definitions”欄裏的“AMBA\_BL”圖標，按滑鼠右鍵選“Close”將其關閉。

H. 點選“ALU\_Vlog”圖標，按住滑鼠左鍵不放，將“ALU\_Vlog”拉進“ALU”裏。



I. 按滑鼠左鍵兩次點選“Definitions”欄裏的“ALU”圖標，畫面中間的工作區會直接切換到“Library Editor”。首先，在“Ports”欄按滑鼠右鍵分別新增兩個埠：

Port -> clock、Protocol -> CLOCK

Port-> reset、Protocol -> RESET

接著將右側“Encapsulation”選單，從“ALU\_TLM\_Wrapper”切換成“ALU\_Vlog”，在“Port”欄位會出現多個接腳的名稱。針對每個接腳，設定對應的“Mapped To”欄位值。

HADDR -> p\_AHB.HADDR

HCLK -> clock.pin

HRDATA -> p\_AHB.HRDATA

HREADYin -> p\_AHB.HREADYin

HREADYout -> p\_AHB.HREADYout

HRESETn -> reset.pin

HRESP -> p\_AHB.HRESP

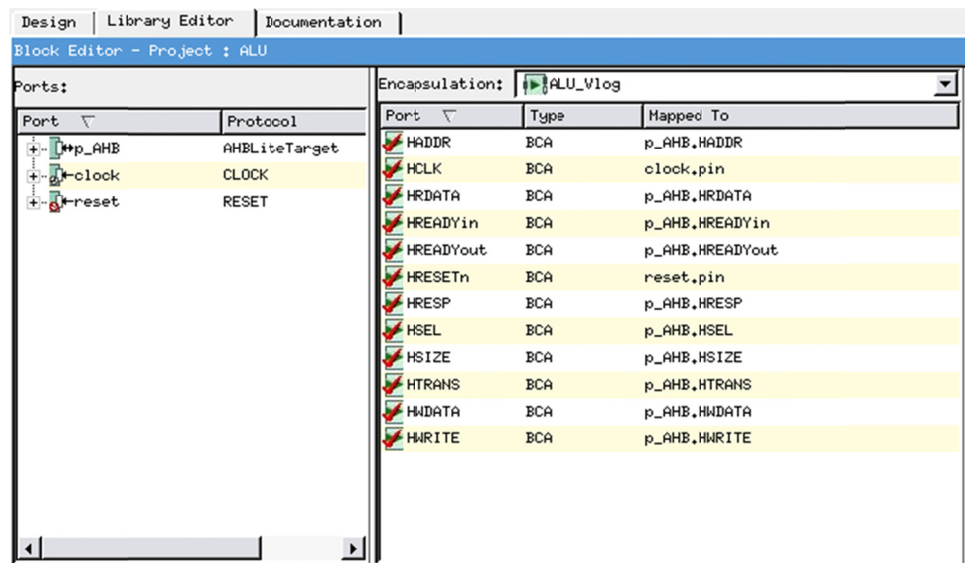
HSEL -> p\_AHB.HSEL

HSIZE -> p\_AHB.HSIZE

HTRANS -> p\_AHB.HTRANS

HWDATA -> p\_AHB.HWDATA

HWRITE -> p\_AHB.WRITE



J. 設定完成，儲存 project 檔。

在 pct 系統選單，選擇 File -> “Save Project As”；

在“Save Project As”設定視窗，設定

Project name -> “ALU”

File name -> alu.xml

點選“OK”完成儲存。

K. 離開 pct。使用任何文字編輯軟體，打開之前步驟中，pct 產生的檔案 PA\_DS5\_ALU/HDL\_IP/ALU/systemc/alu\_vlog.h，找出下列兩段文字

```
verilog_files.push_back(hpg_hdl_src_path + std::string("../alu_top.v"));
```

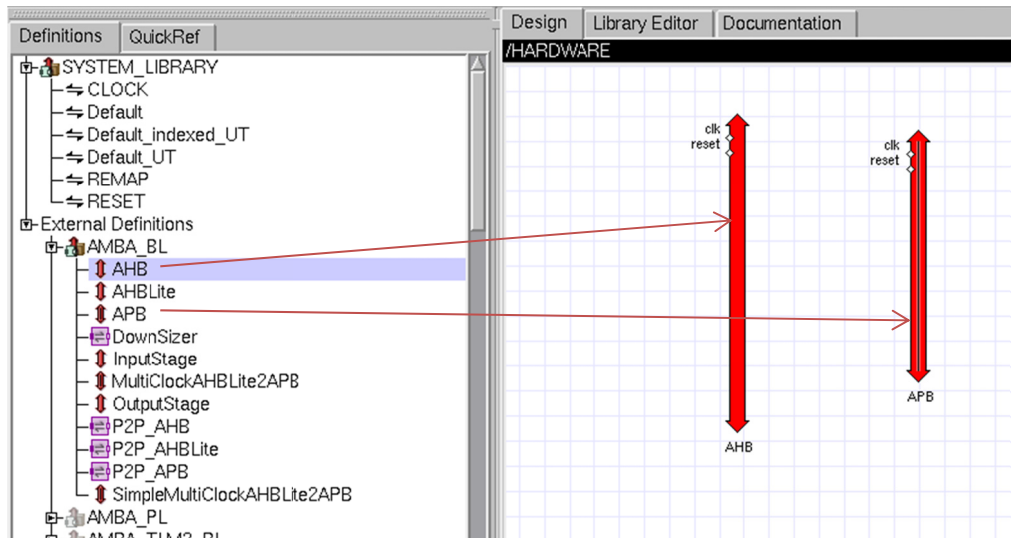
```
verilog_files.push_back(hpg_hdl_src_path + std::string("../g/alu.v"));
```

括號內...是指 ALU 兩個 \*.v 程式檔案在您電腦的絕對路徑，請修改成對應於 PA\_DS5\_ALU/PA/export/目錄的相對路徑。

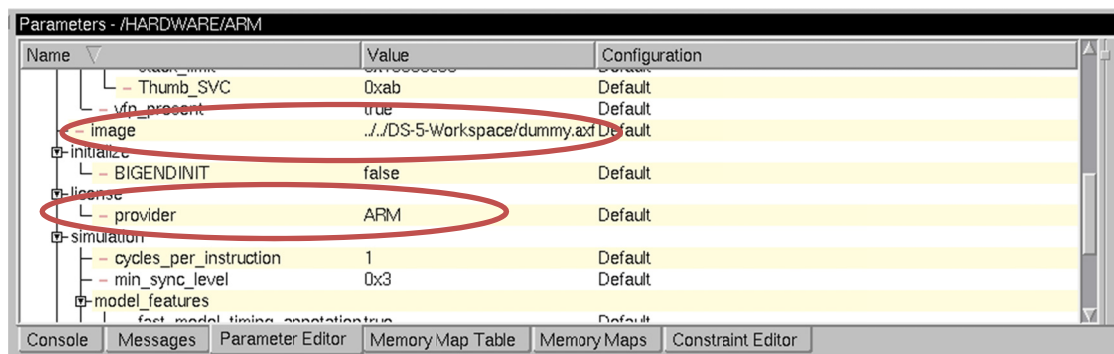
```
verilog_files.push_back(hpg_hdl_src_path +  
std::string("../HDL_IP/ALU/verilog/alu_top.v"));  
verilog_files.push_back(hpg_hdl_src_path +  
std::string("../HDL_IP/ALU/verilog/alu.v"));
```

## 3.2 建立虛擬平台

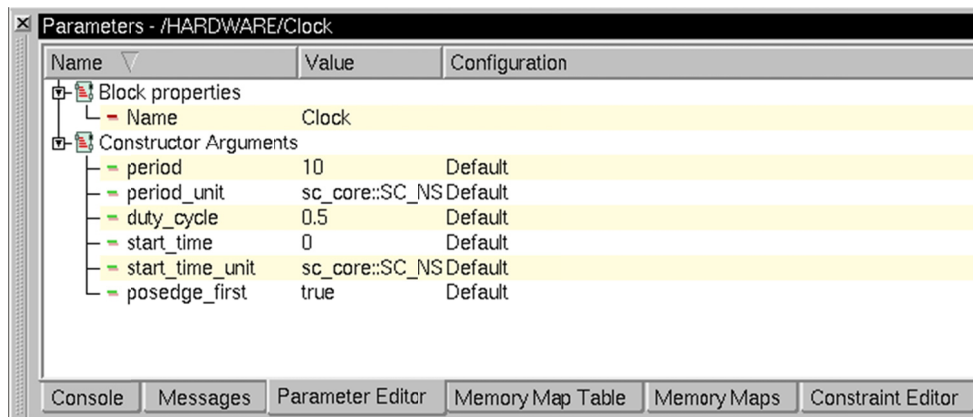
- A. 在 PA\_DS5\_ALU/目錄內，執行“source pa\_setup.csh”設定 PA 環境參數。
- B. 執行指令“pct&”以啟動 Platform Creator (pct) 圖形介面環境。
- C. 在 pct 左側 “Definitions” 欄位的樹狀圖，點選展開 “External Definition” 及其下的“AMBA\_BL”元件庫。按滑鼠左鍵拉出 “AHB” 及 “APB” 元件各一個至中央工作區(Design)。點選各元件並按滑鼠右鍵選“Rename”分別更名成 “AHB”及“APB”。



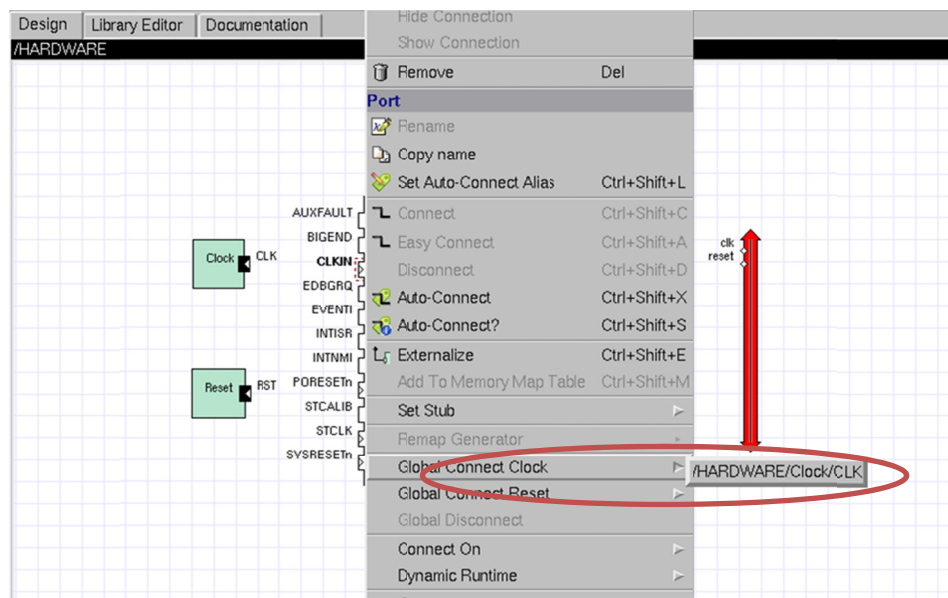
- D. 展開 “ARM\_CORTEX\_M4\_TLM2\_LT\_PSP”元件庫，按滑鼠左鍵拉出一個 “ARM\_CORTEX\_M4\_TLM\_LT” 元件至工作區，並更名成 “ARM”。點選 ARM 元件，切換到下方的“Parameter Editor”，修改參數 Extra properties -> image 為 ../../DS-5-Workspace/dummy.axf。將“license provider”由“Synopsys”切換為“ARM”。



- E. 展開 “GenericIPlib” 元件庫，按滑鼠左鍵分別拉出一個 “ClockGenerator” 與 “ResetGenerator”元件至工作區。分別更名成 “Clock” 與 “Reset”。點選 Clock 元件，切換到下方的 Parameter Editor，可以看見 period 參數為 10(ns)。因此虛擬平台之頻率為 100MHz。

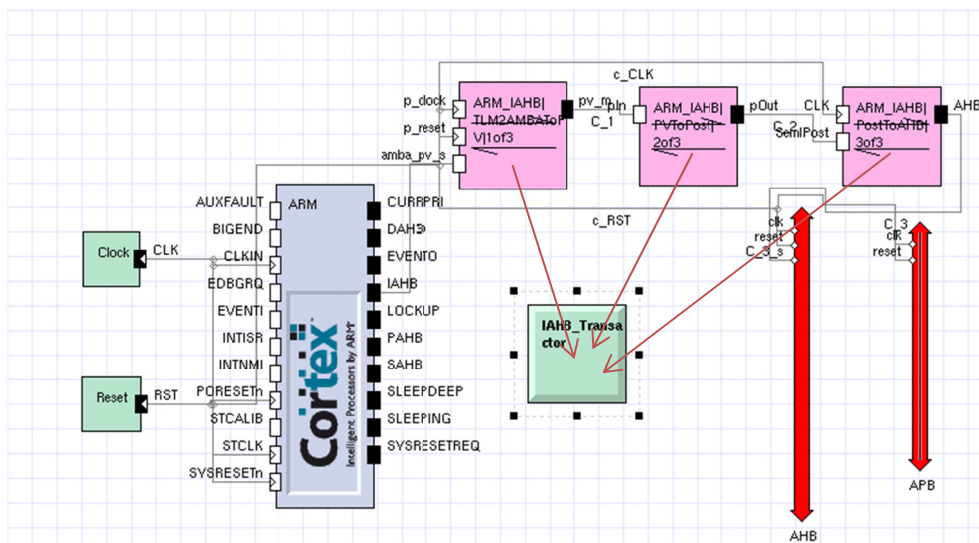


- F. 點選 ARM 元件的“CLKIN”埠，按滑鼠右鍵選 Global Connect Clock ->/HARDWARE/Clock/CLK，便可以接上 Clock 元件的 CLK 埠。以同樣方式將 ARM 元件的“STCLK”埠也接上 Clock 元件的 CLK 埠。

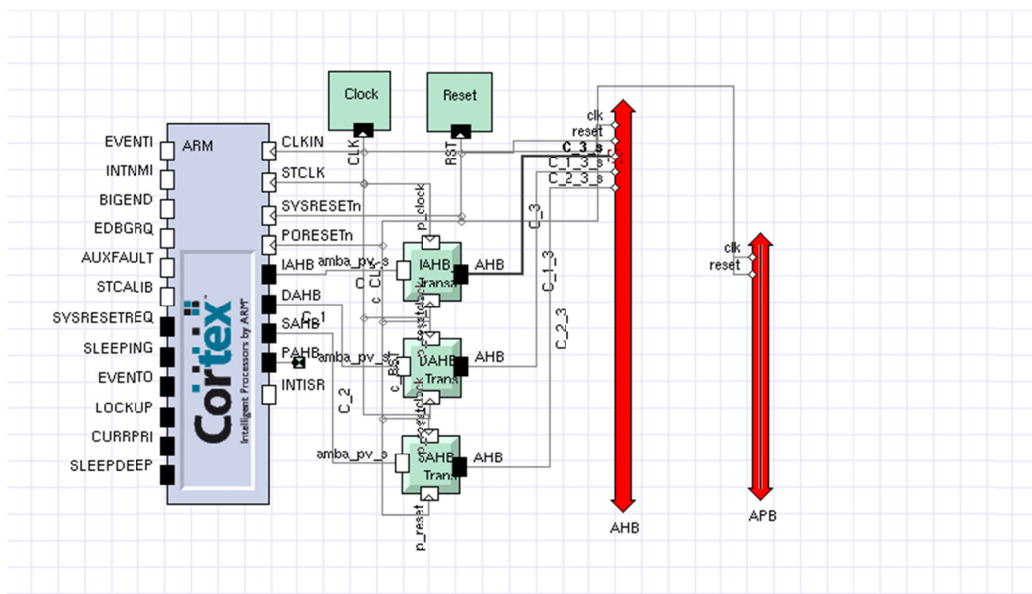


- G. 分別點選 AHB 及 APB 元件的 clk 埠，依照步驟 F 的方式接上 Clock 元件的 CLK 埠。
- H. 點選 ARM 元件的“SYSRESETn”埠，按滑鼠右鍵選 Global Connect Reset ->/HARDWARE/Reset/RST，便可以接上 Reset 元件的 RST 埠。以同樣方式將 ARM 元件的“PORESETn”埠也接上 Reset 元件的 RST 埠。
- I. 分別點選 AHB 及 APB 元件的 reset 埠，依照步驟 H 的方式接上 Reset 元件的 RST 埠。
- J. 按鍵盤 Ctrl 鍵不放，用滑鼠左鍵點選 ARM 元件的“IAHB”埠與 AHB 元件，接著放開 Ctrl 鍵，按滑鼠右鍵選擇“Easy Connect”，由於 IAHB 埠與 AHB 的傳輸協定不同，因此 PA 會自動接上適合的轉接器(transactors)。將各轉接器的 clock 及 reset 埠依步驟 F 及 H 的方式分別接上 Clock 元件及 Reset 元件。接著點選工作區空白處按滑鼠右鍵，選“Create Hierarchical Instance”並取名

為 “IAHB\_Transactor”，接著點選所有轉接器並按滑鼠左鍵全部拖入 “IAHB\_Transactor” 包裝內。



- K. 依照步驟 J 的方式，把 ARM 元件的 “DAHB” 埠與 “SAHB” 埠分別以 Easy Connect 功能接上 AHB 元件、接好連接器的 clock 與 reset 埠、並建立 “DAHB\_Transactor” 與 “SAHB\_Transactor” 包裝，並把各自對應的連接器分別拖入。可以適當調整元件位置、大小及埠的排列，讓工作區較為整潔。
- L. 點選 ARM 元件的 “PAHB” 埠，按滑鼠右鍵選擇 Set Stub -> amba\_pv\_master\_stub。



- M. 切換 pct 下方的設定顯示分頁到 “Parameter Editor”，再分別點選 IAHB、DAHB、及 SAHB 接上 AHB 的接點（以上圖為例，即 C\_3\_S、C\_1\_3\_S、及 C\_2\_3\_S），設定以下參數。

IAHB: priority 0, default\_master true

DAHB: priority 1, default\_master false

SAHB: priority 2, default\_master false

- N. 展開“AMBA\_TLM2\_GENERIC\_LIBRARY”元件庫，按滑鼠左鍵分別拉出 2 個 “ambapv\_memory” 元件至工作區，並更名成 “ROM” 及 “RAM”。依序點選兩個元件，並在 Parameter Editor 修改參數“size”為“16M”、“BUS\_WIDTH”為“32”。依照步驟 J 的方式，將 ROM 元件與 RAM 元件的 ambapv\_s 埠分別以 Easy Connect 功能接上 AHB 元件、接好連接器的 clock 與 reset 埠、並建立 “ROM\_Transactor” 與 “RAM\_Transactor” 包裝，並把各自對應的連接器分別拖入。**請注意先設參數後接線的順序。**

- O. 點選工具列上的“Connection Tool”(折線狀圖標)，將 AHB 元件接上 APB 元件。Connection Tool 與 “Selection Tool”(箭頭狀圖標)的切換，在以下步驟將不會特別提及，請同學自行注意。

- P. 展開“PRIMECELL”元件庫，拉出一個“TIMER\_SP804\_Model”元件，更名為 “Timer”。從 GenericIPlib 元件庫拉出一個“PortBinder”元件，更名為 “NVIC\_Transactor”。

點選 Timer 元件的“TimClkEn1”埠，按滑鼠右鍵選 Set Stub -> PinDrive。接著點選產生出來的小方塊，將參數“stub\_value”改為“true”。以同樣方式設定 “TimClkEn2” 埠。

將 Timer 元件的 “TimClk” 埠及 “PCLK” 埠接上 Clock 元件的 CLK 埠；將 “PRESETn\_rst” 埠接上 Reset 元件的 RST 埠。

點選“NVIC\_Transactor”元件，從 Parameter Editor 修改參數“NR\_OF\_PORTS”為“2”、“DATA\_TYPE”為 “sc\_biguint<240>”；點選該元件的“OutPort”，修改參數“data\_width”為“240”。

將 Timer 元件的 “TimInt1” 埠接上 NVIC\_Transactor 元件的“InPort[0]”埠、“TimInt2” 埠接上 “InPort[1]” 埠。

將 Timer 元件的 “p\_APB” 埠接上 APB；將 NVIC\_Transactor 元件的 OutPort 埠接上 ARM 元件的“INTISR”埠。

- Q. 從系統選單選擇 Project -> Open Library File，開啟元件庫檔案 PA\_DS5\_ALU/library/displayLib/display.xml。展開 display 元件庫，拉進一個元件 display\_APB\_TLM 至工作區並更名為“Display”。

將 Display 元件的“din”埠接上 APB。

- R. 將下方設定顯示分頁切換到“Memory Map Table”。依序點選下列元件上的埠，並按滑鼠右鍵選擇 “Add to Memory Map Table”。

ARM->IAHB

ARM->DAHB

ARM->SAHB

ROM->ambapv\_s

RAM->ambapv\_s

APB (點選整個 APB 元件)



Display->din

Timer->p\_APB

可以發現 Memory Map Table 的橫軸列出所有主埠(initiator/master ports)，縱軸則是從埠(target/slave ports)。點選以下所列方格，按滑鼠右鍵選擇 Cell -> Remove Link。

(/HARDWARE/APB, /HARDWARE/ROM/ambapv\_s)

(/HARDWARE/APB, /HARDWARE/RAM/ambapv\_s)

(/HARDWARE/APB, /HARDWARE/APB)

(/HARDWARE/ARM/IAHB, /HARDWARE/Display/din)

(/HARDWARE/ARM/IAHB, /HARDWARE/Timer/p\_APB)

(/HARDWARE/ARM/DAHB, /HARDWARE/Display/din)

(/HARDWARE/ARM/DAHB, /HARDWARE/Timer/p\_APB)

(/HARDWARE/ARM/SAHB, /HARDWARE/Display/din)

(/HARDWARE/ARM/SAHB, /HARDWARE/Timer/p\_APB)

另外，將以下所列方格填入對應位址值。

(/HARDWARE/ARM/IAHB, /HARDWARE/ROM/ambapv\_s) : "0x0"

(/HARDWARE/ARM/IAHB, /HARDWARE/RAM/ambapv\_s) : "0x20000000"

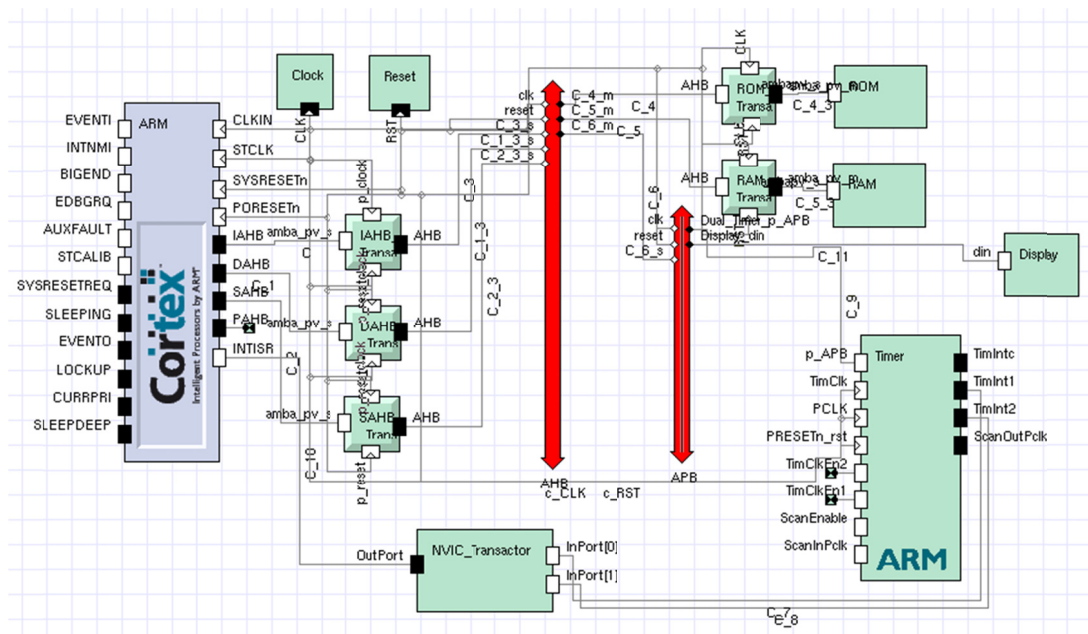
(/HARDWARE/ARM/IAHB, /HARDWARE/APB) : "0x41000000"

(/HARDWARE/APB, /HARDWARE/Display/din) : "0x0"

(/HARDWARE/APB, /HARDWARE/Timer/p\_APB) : "0x1000"

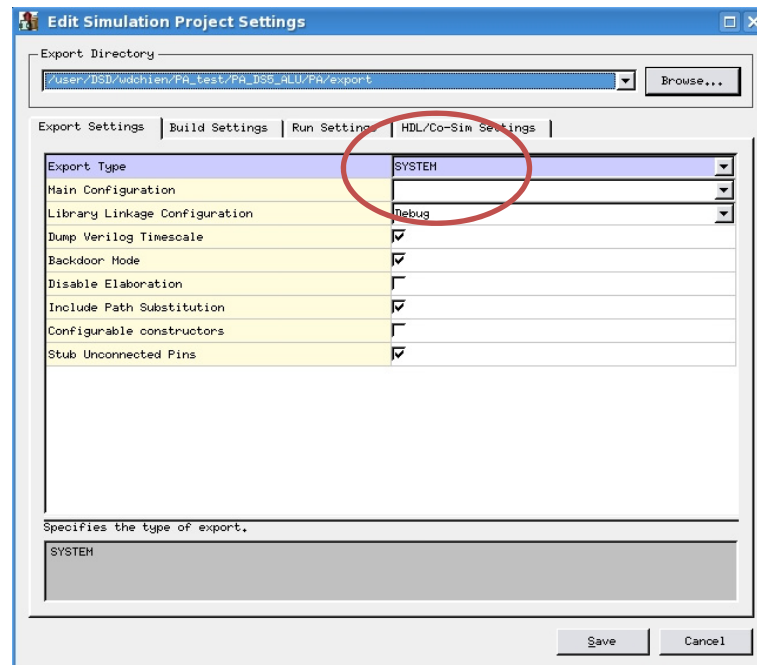
Memory Map Table - /HARDWARE

	/HARDWARE/ARM/IAHB	/HARDWARE/ARM/DAHB	/HARDWARE/ARM/SAHB	/HARDWARE/APB
/HARDWARE/ROM/ambapv_s	0x0	0x0	0x0	
/HARDWARE/RAM/ambapv_s	0x20000000	0x20000000	0x20000000	
/HARDWARE/APB	0x41000000	0x41000000	0x41000000	Remove link
/HARDWARE/Display/din	Remove link	Remove link		0x0
/HARDWARE/Timer/p_APB				0x1000

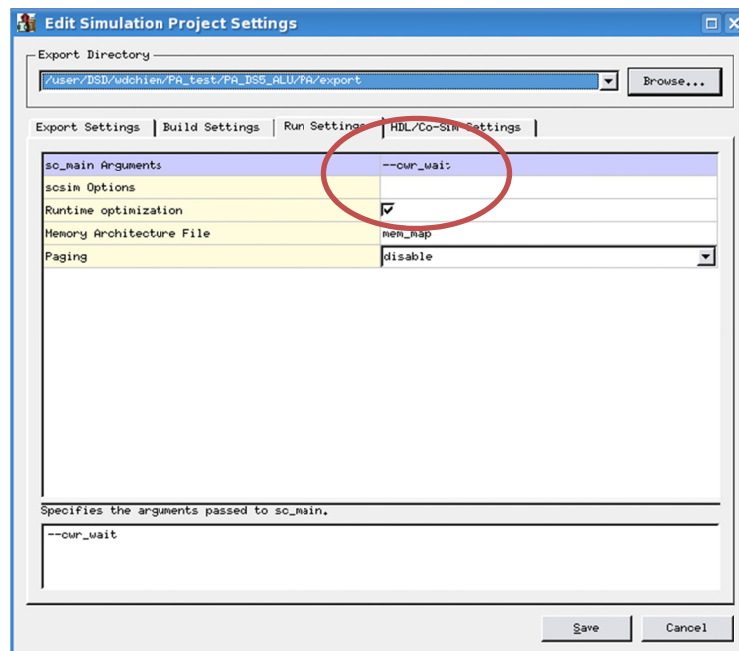


- S. 在功能列選 Simulation -> Project Settings。請檢查以下兩組參數之預設值，若遺失請加以設定：

Export Settings -> Export Type -> 選擇 "System"。



Run Settings -> sc\_main Arguments -> 加入 "--cwr\_wait"。



設定完成後，記得按 "Save"儲存修改過後的設定值。

- T. 從系統選單選擇 Check -> Check Hardware 進行檢查，若有任何錯誤或警告訊息，會顯示於下方設定顯示分頁的“Message”。出現“Hardware System is Ok”提示訊息，表示無重大設計錯誤（重要的埠忘記接線或參數設定錯誤、或是 memory map 尚未設定等等），可以進行模擬，但並非表示設計完全無誤。
- U. 點選左側所有彩色(開啟)的元件庫圖標，按滑鼠右鍵選擇“Close”以關閉元件庫。從系統選單選擇 File -> Save Project As 以儲存虛擬平台檔，“Project name”請設為“Project\_01\_HDL”，“File name”請設為“Project\_01\_HDL.xml”
- V. 按下工具列上的“Run Simulation”圖標(三角形播放圖示)啟動模擬。首先會進行編譯，PA 預設會把所有模擬所需檔案輸出至 PA\_DS5\_ALU/PA/export/目錄。PA 會先編譯虛擬平台，完成後會出現以下訊息：

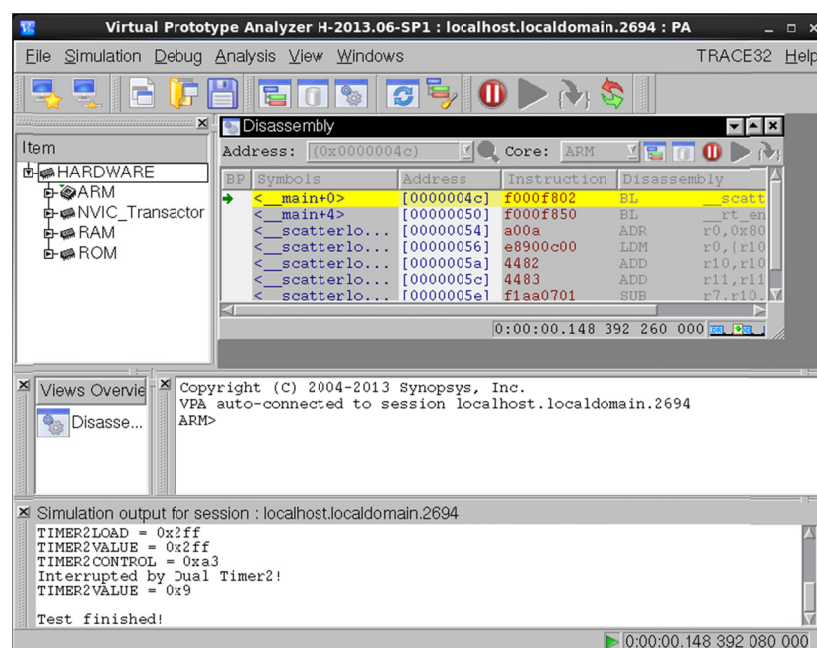
```
#####
GDB stubs allow attaching a GDB to a core.
Take care of using a GDB that was configured for the appropriate target
architecture.

Use this command to attach to a core:
(gdb) target remote dsd01.cic.org.tw:PORT-ID

The following cores are waiting for GDB to attach:
PORT-ID      CPU-ARCH      INSTANCE_NAME
12346  ARMv7-M      HARDWARE.ARM0
#####
```

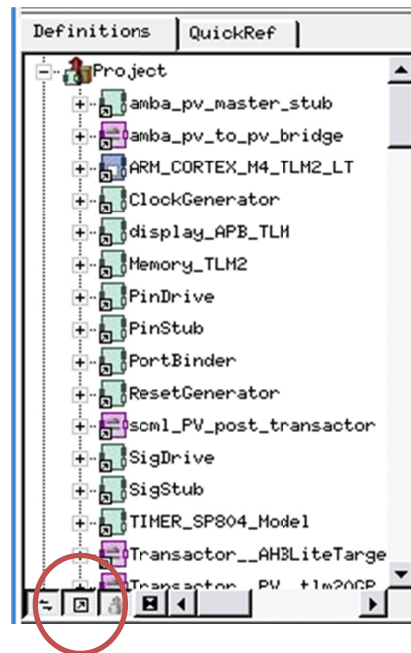
此時 PA 硬體模擬部分已經準備完成，等候軟體除錯工具連接其處理器模型。

- W. 回到 Terminal 視窗，鍵入“vpa&”指令以啟動 Virtual Prototype Analyzer (vpa)。Virtual Prototype Analyzer 啟動後會自動接上正在執行的虛擬平台。從系統選單選擇 Debug-> Load Image 以載入測試軟體二元碼，並指定軟體二元碼檔案為 PA\_DS5\_ALU/DS-5-Workspace/dual\_timer\_test.axf。載入完成後點按工具列上的“Continue the simulation”圖標(三角形播放圖示)，測試軟體會持續執行至結束，並將結果顯示於 vpa 下方的 Simulation output 欄位。

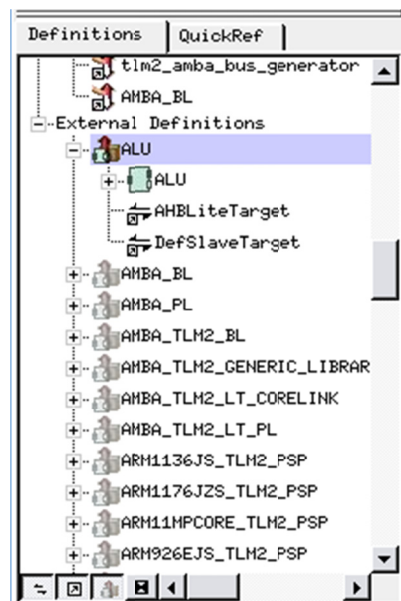


### 3.3 將 ALU 元件掛入虛擬平台

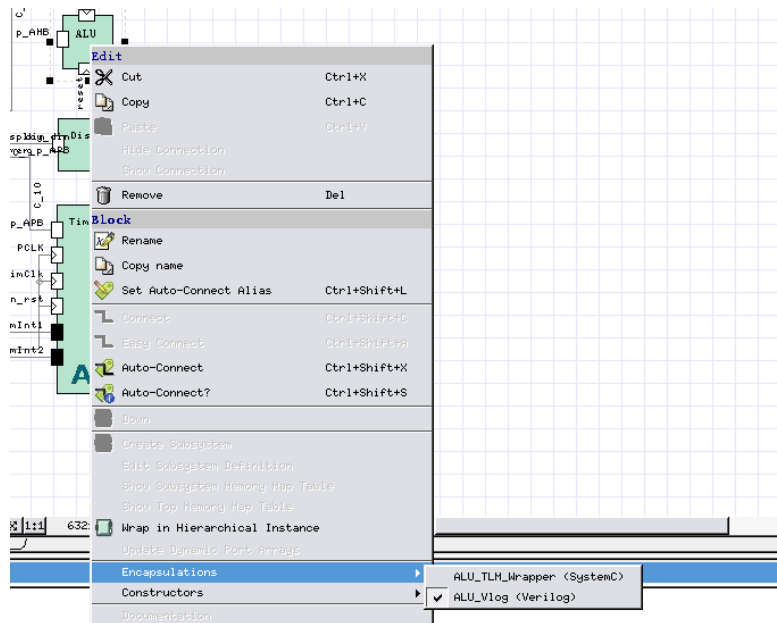
- A. 在 PA\_DS5\_ALU/目錄內，執行“source pa\_setup.csh”設定 PA 環境參數。
- B. 執行 pct&。
- C. 在功能列，選擇 File -> Open Project，開啟虛擬平台計畫檔 PA\_DS5\_ALU/PA/Project\_01\_HDL.xml。載入後，虛擬平台會出現在中央的工作區(Design)。
- D. 在“Definitions”欄左下角，點按左數第二個“Show references”鍵，原本只顯示少數資訊的“Project”樹狀圖項目便會顯示所有使用的元件。



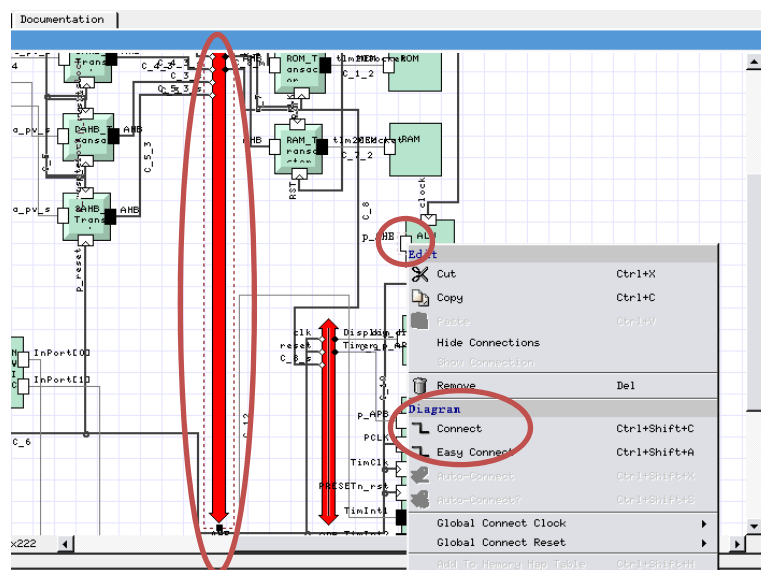
- E. 在功能列，選擇 Project -> Open Library File，並且開啟檔案 PA\_DS5\_ALU/HDL\_IP/ALU/alu.xml，在“Definitions”欄的“External Definitions”項目下會出現“ALU”元件庫。



- F. 展開“ALU”元件庫，按滑鼠左鍵拉取一個 ALU 元件至工作區。點選此元件按滑鼠右鍵，選 “Rename”將元件更名為“ALU”；同樣按滑鼠右鍵選 “Encapsulations”，從“ALU\_TLM\_Wrapper (SystemC)”切換至“ALU\_Vlog (Verilog)”。



- G. 點選 ALU 元件的 “clock”埠，按滑鼠右鍵選 Global Connect Clock -> /HARWARE/Clock/CLK，便可以接上 Clock 元件的 CLK 埠。點選 ALU 元件的 “reset”埠，按滑鼠右鍵選 Global Connect Reset-> /HARWARE/Reset/RST，便可以接上 Reset 元件的 RST 埠。
- H. 在工具列中選用 “Selection Tool”(箭頭狀圖示)。按住鍵盤“Ctrl”鍵不放，以滑鼠左鍵同時點選“AHB”(整體)與“ALU”元件的“p\_AHB”埠，選完後放開 Ctrl 鍵，並按滑鼠右鍵選擇 “Connect”完成連接。請注意步驟 F 和步驟 H 的順序，先切換再接線。



- I. 點選 ALU 元件的“p\_AHB”埠，按滑鼠右鍵選“Add to Memory Map Table”。將 pct 下方設定顯示分頁從“Console”切換到“Memory Map Table”。點選方格 (/HARDWARE/APB, /HARDWARE/ALU/p\_AHB)，按滑鼠右鍵選 “Remove Link”；接著在方格 (/HARDWARE/ARM/IAHB, /HARDWARE/ALU/p\_AHB) 填寫 “0x40001000”，三個未設定位址的方格會同時設定完成。

Memory Map Table - /HARDWARE				
	/HARDWARE/ARM/IAHB	/HARDWARE/ARM/DAHB	/HARDWARE/ARM/SAHB	/HARDWARE/APB
/HARDWARE/ROM/ambapv_s	0x0	0x0	0x0	
/HARDWARE/RAM/ambapv_s	0x20000000	0x20000000	0x20000000	
/HARDWARE/APB	0x41000000	0x41000000	0x41000000	
/HARDWARE/Display/din				0x0
/HARDWARE/Timer/p_APB				0x1000
/HARDWARE/ALU/p_AHB	0x40001000	0x40001000	0x40001000	

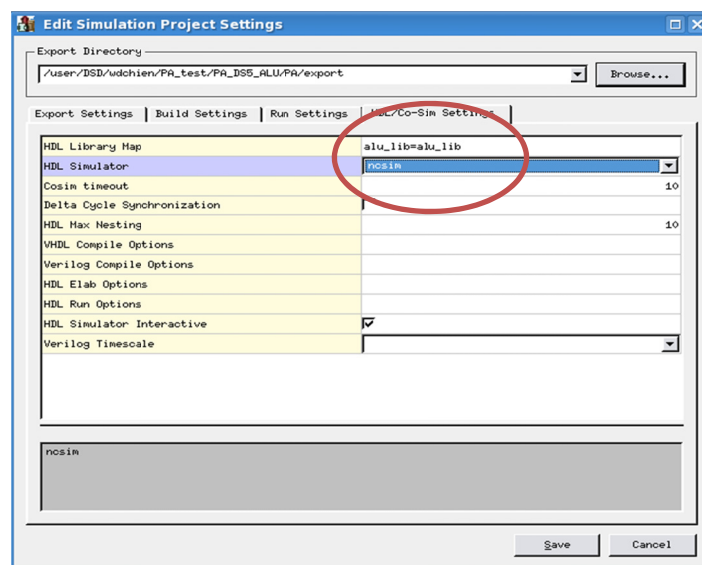
- J. 在功能列選 Simulation -> Project Settings。請檢查以下三組參數之預設值，若遺失請加以設定：

Export Settings -> Export Type -> 選擇 "System"。

Run Settings -> sc\_main Arguments -> 加入 “--cwr\_wait”。

HDL-Co-Sim Settings -> HDL Library Map -> 加入 alu\_lib=alu\_lib。其設定格式為 lib\_name=lib\_path。左側是前述 3.1 節(包裝 HDL IP)，執行步驟 G 時所設定之名稱；右側是 co-sim 時放置該 HDL library 的目錄名稱。如果有兩個以上 HDL IP，格式為 lib\_name\_A=lib\_path\_A lib\_name\_B=lib\_path\_B (中間以空白鍵分隔)。

此外 HDL Simulator 可以選擇 NCSIM (INCISIV)或 VCS (ModelSim 目前測試不正常，不建議使用)。另外請勾選 “HDL Simulator Interactive”以方便在 co-sim 時對 HDL simulator 進行設定。若您使用 VCS，請在 "Verilog Compile Settings" 加入 "-debug\_all" 參數。設定完成後，記得按 "Save"儲存修改過後的設定值。



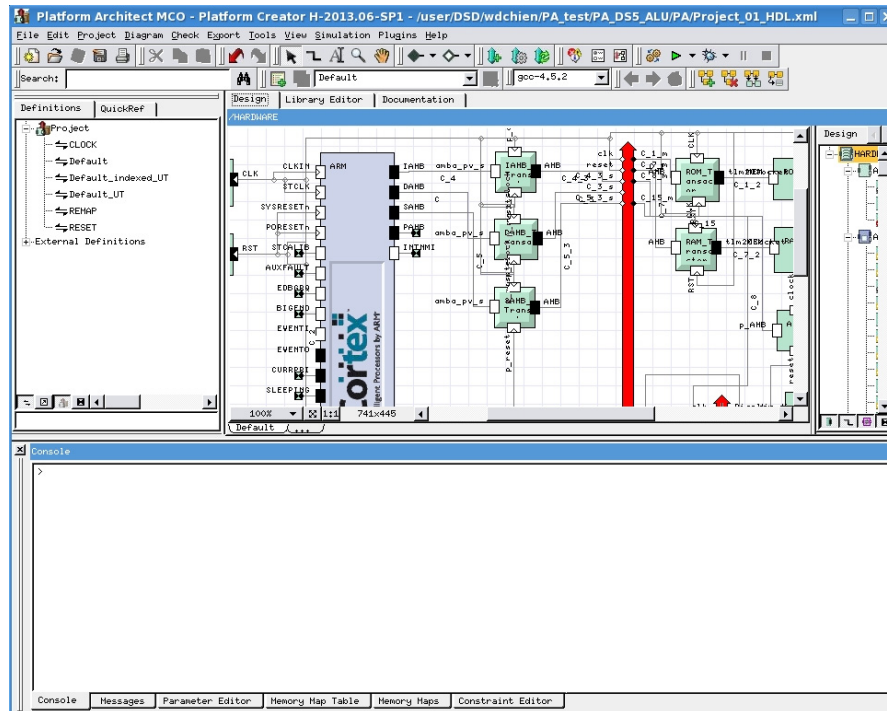
- K. 在功能列選 File -> Save Project，以儲存更改設計後的虛擬平台。關閉 pct。



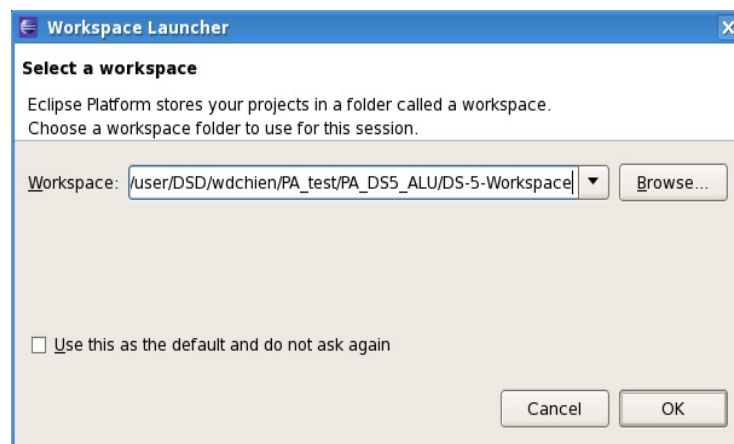
### 3.4 啟動全系統模擬與軟體除錯

啟動 Platform Architect 及 HDL 模擬器

- A. 執行指令"source run.csh&"，該檔案會先設定 PA 環境參數，接著啟動 pct 並載入虛擬平台 PA\_DS5\_ALU/PA/Project\_01\_HDL.xml。



同時也會啟動 DS-5 主程式。請在 Workspace Launcher 按“Browse”功能選擇 PA\_DS\_ALU/DS-5-Workspace 目錄。



- B. 按 pct 上的 Run Simulation。PA 預設會把所有模擬所需檔案輸出至 PA\_DS5\_ALU/PA/export/目錄。PA 會陸續編譯虛擬平台，啟動並將控制權交

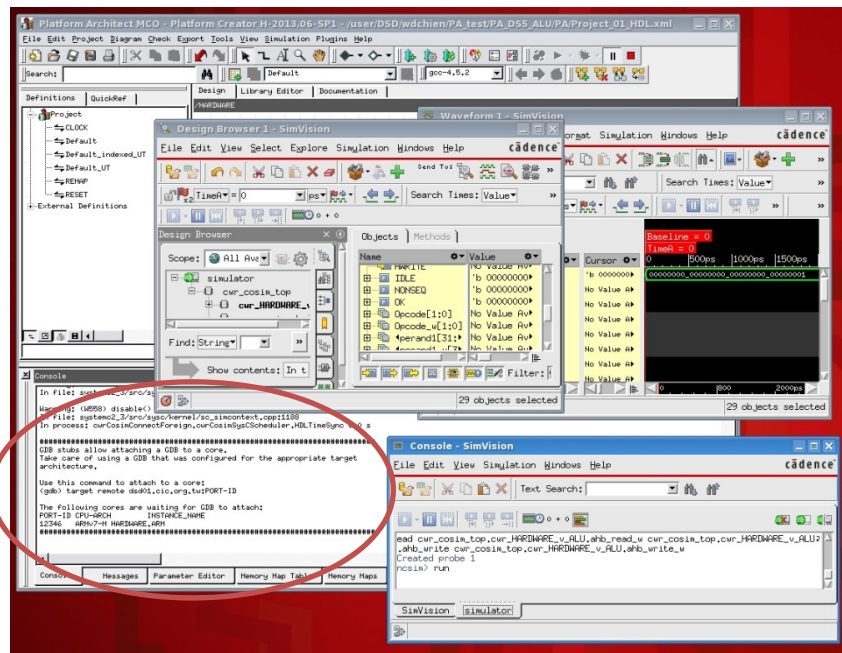
由 HDL simulator (範例為 NCSIM; HDL 分析及編譯需要較長時間, 請稍待), HDL simulator 設定完並繼續執行(在 NCSIM 為 Run/Continue 按鍵)後, 控制權回到 PA。依 pct Console 上的提示, 輸入指令::scsh::c, 接者會出現以下訊息:

```
#####
GDB stubs allow attaching a GDB to a core.
Take care of using a GDB that was configured for the appropriate target
architecture.

Use this command to attach to a core:
(gdb) target remote dsd01.cic.org.tw:PORT-ID

The following cores are waiting for GDB to attach:
PORT-ID      CPU-ARCH      INSTANCE_NAME
12346        ARMv7-M       HARDWARE.ARM0
#####
```

此時 PA 硬體模擬部分已經準備完成, 等候 DS-5 以 CADI 介面連接其處理器模型。

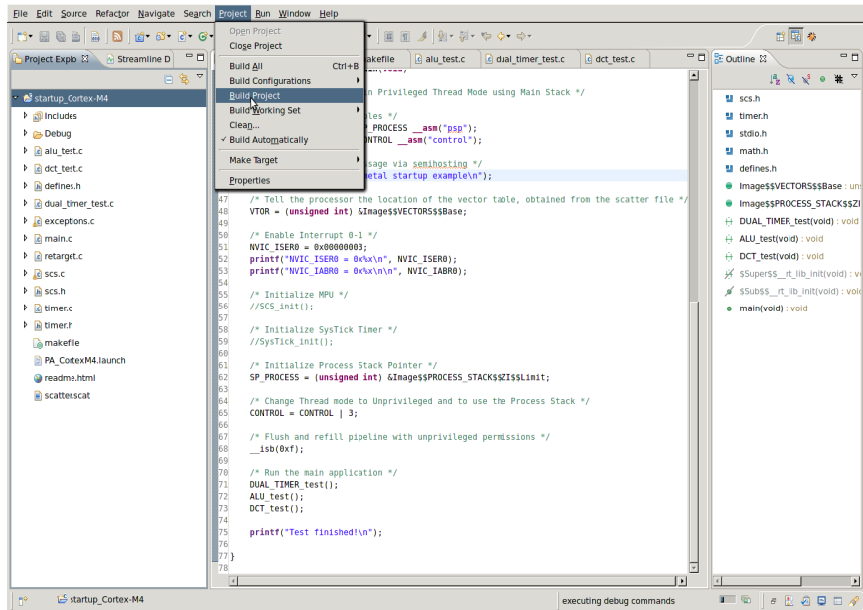


DS-5 除錯軟體必須透過一組描述 target 的資料, 才能辨識如何連接 target 是何種介面, 這個資料便是 Configuration Database。DS-5 軟體中並無 PA 相關的 Configuration Database, 因此, 使用者在第一次使用 DS-5 連接 PA 前, 必須先匯入 PA 的 Configuration Database 至 DS-5。請依序執行下列步驟以進行設定。

C. 請先切換至 C/C++ 版面, 在功能列選擇 Window -> Open Perspective -> C/C++。

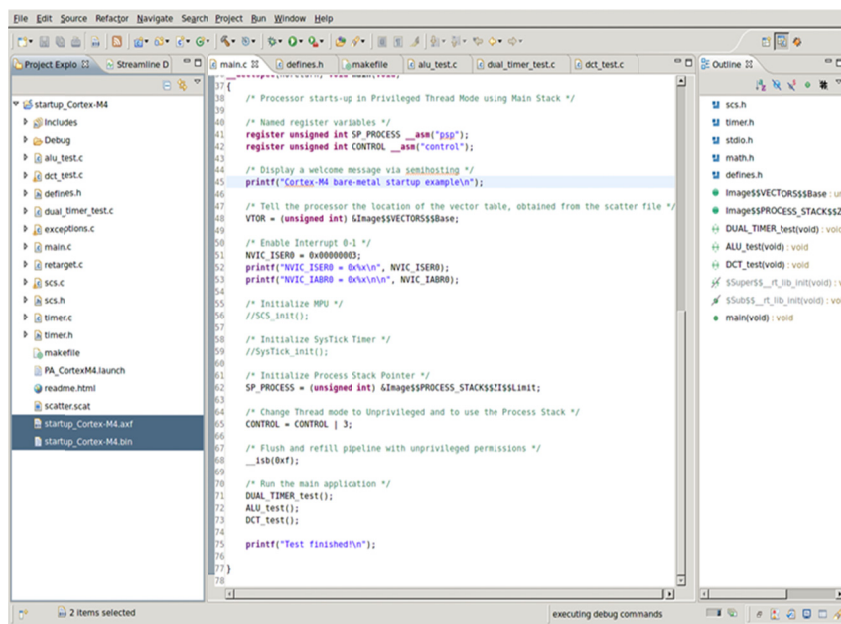
於左側 Project Explorer 中選擇 startup\_Cortex-M4 專案,



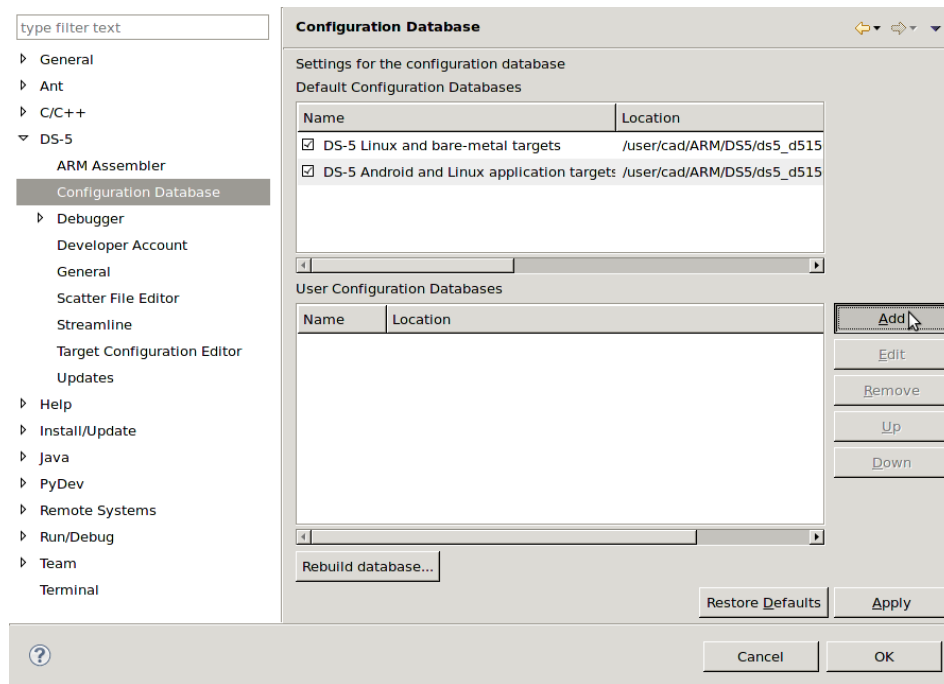


在功能列選擇 Project -> Build Project。

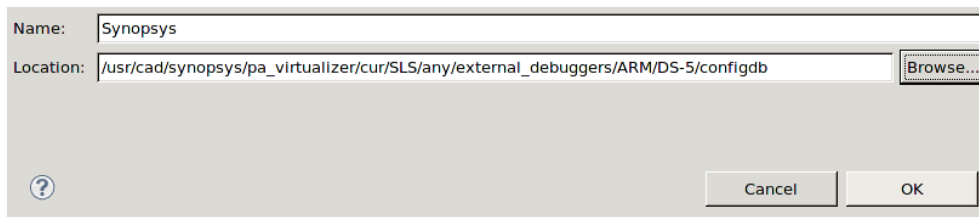
編譯成功後，在 Project Explorer 中您將會看到 start\_Cortex-M4.axf 及 startup\_Cortex-M4.bin 兩個檔案產生。



D. 打開 Configuration Database 視窗，在功能列選擇 Window -> Preference -> DS-5 -> Configuration Database。按下 Add 按鍵以增加 User Configuration Database。



將新增資料庫命名為 Synopsys，並選擇您安裝的 PA 目錄底下的 SLS/any/external\_debuggers/ARM/DS-5/configdb。



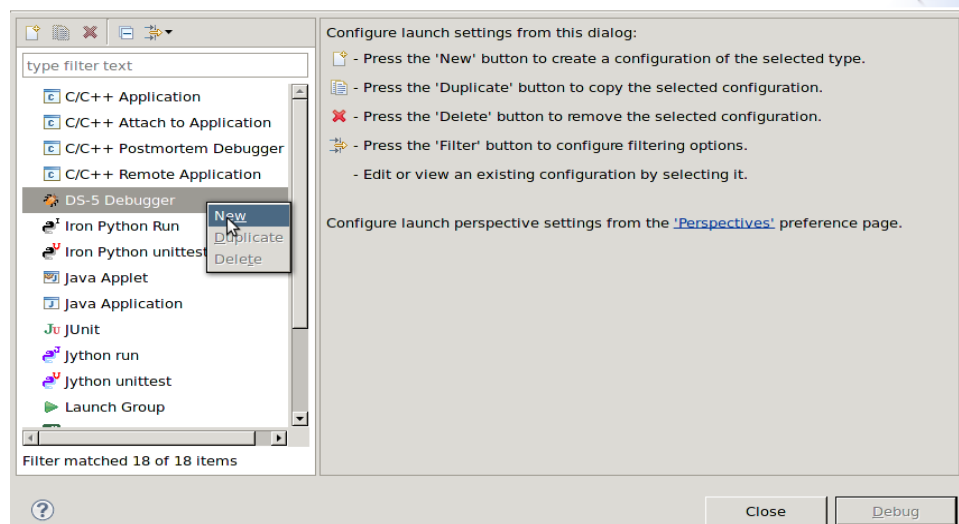
點選 Rebuild database 圖標，重建完後按 OK 按鈕離開，DS-5 便有 PA 的 Configuration database，可以設定 Debugger 連線。

- E. Configuration database 設定完成之後，接著可以建立一個特定的 DS-5 debugger configuration 以連接 PA。

在 DS-5 系統選單選擇 Run -> Debug Configurations，於 DS-5 Debugger 上按

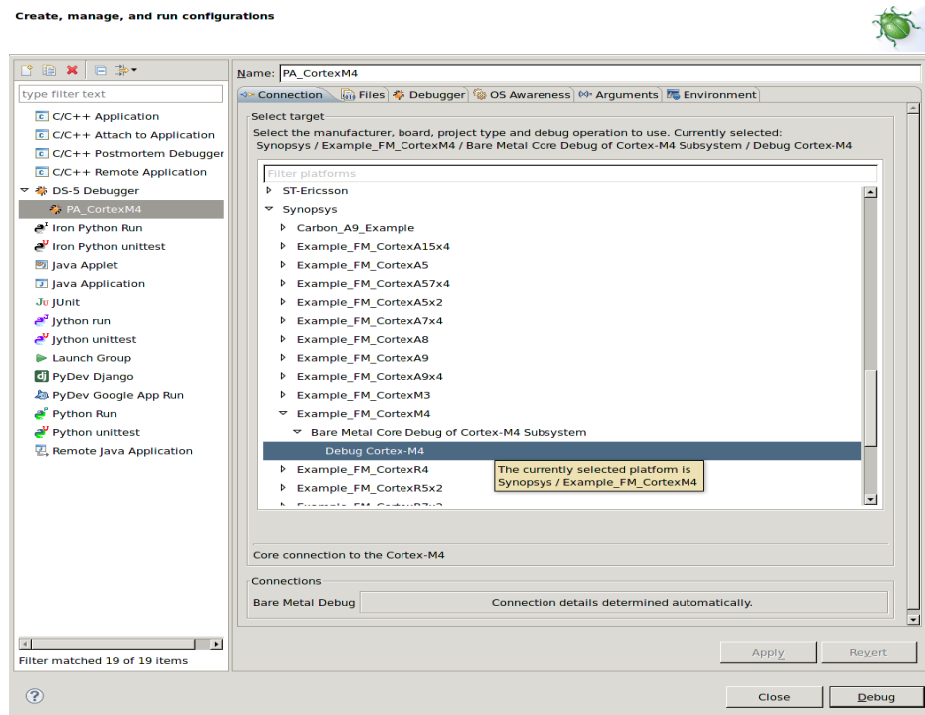
#### Create, manage, and run configurations

Create, edit or choose a configuration to launch a DS-5 debugging session.

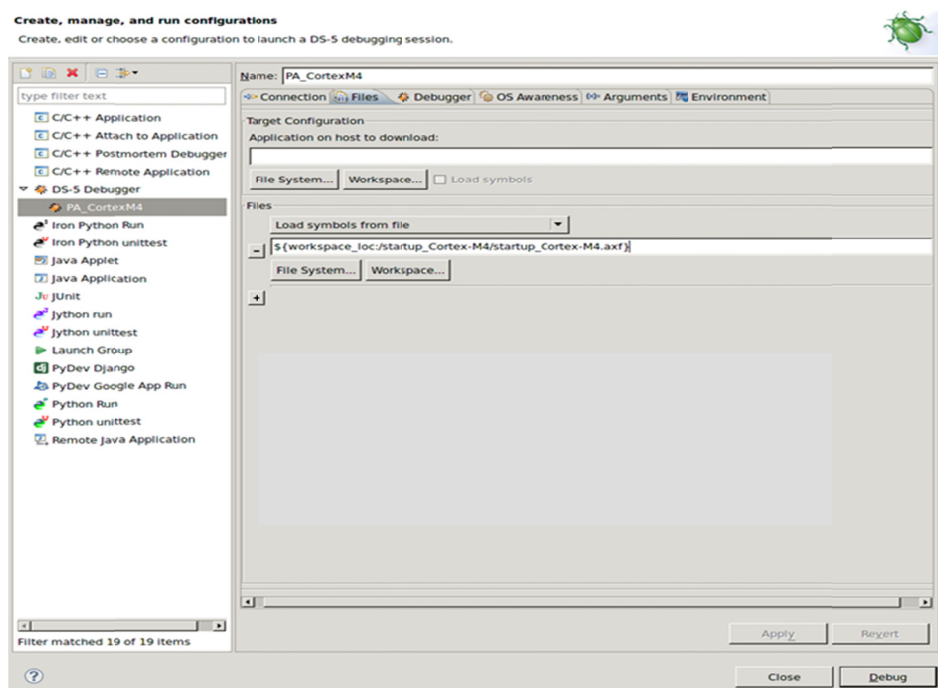


滑鼠右鍵，選擇 New，

並改名為 PA\_CortexM4，於 Connection 分頁中，選 target 為 Synopsys -> Example\_FM\_CortexM4 -> Bare Metal Core Debug of Cortex-M4 Subsystem -> Debug-M4。



於 Files 分頁中，設定您要載入的 symbol file，本例為  
\${workspace\_loc:/startup\_Cortex-M4/startup\_Cortex-M4.axf} (亦即  
PA\_DS5\_ALU/DS-5-Workspace/startup\_Cortex-M4/startup\_Cortex-M4.axf)。



於 Debugger 分頁中，設定 Run Control 為 Connect only，並指定 debugger 執行的命令

interrupt

restore \${workspace\_loc:/startup\_Cortex-M4.bin} binary 0x0

set \$PC=0

delete breakpoints

b main

其中

Interrupt:停止 CPU 運行；

restore file binary 0x0: 將 rom 檔回存到 target 位置 0x0 起始的地方；

set \$PC=0: 設定 PC 為 0x0；

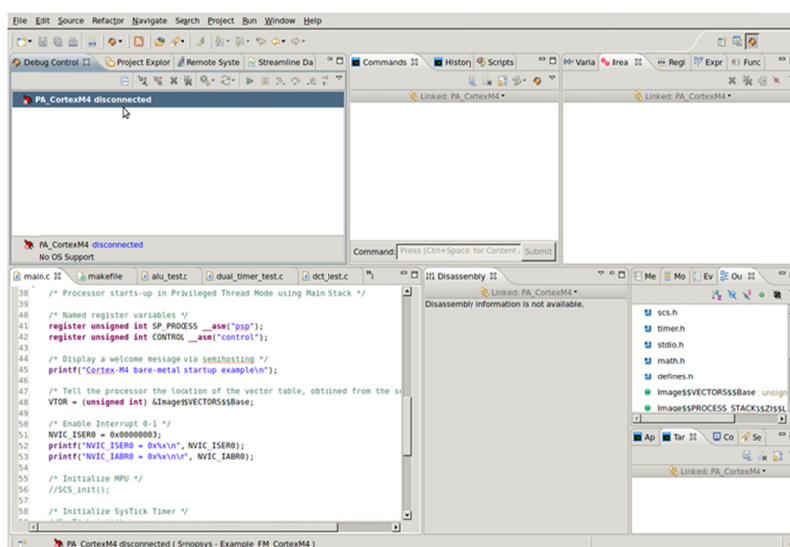
delete breakpoints: 將前一次 debugger 中設定的中斷點清除；

b main:設定中斷點至 main function。

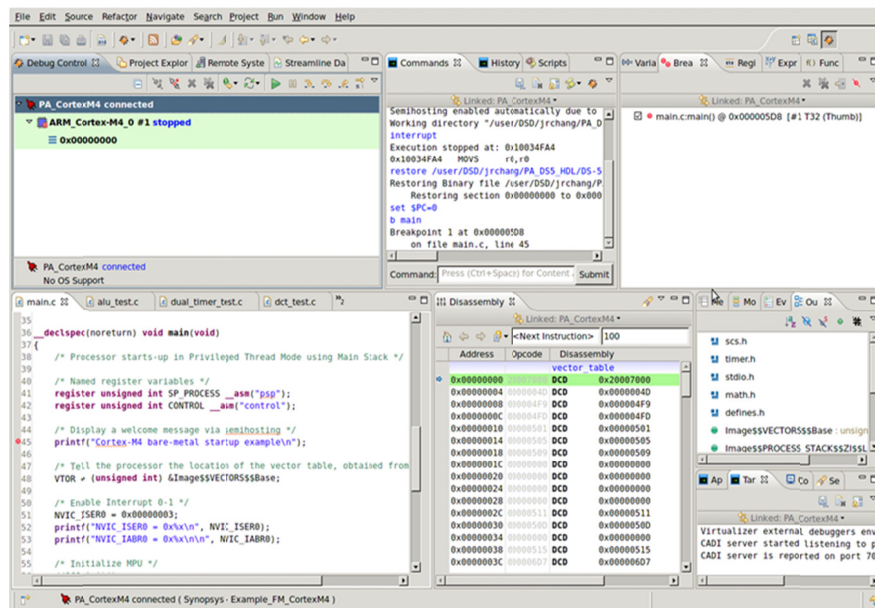
上述指令會在 DS-5 連接 target 時執行，會使得 CPU 停止，並設定下一個執行點為 0x0、同時把 rom 檔案存回 target 0 的位置，故下一次 CPU 執行時，就是執行 rom 檔。CPU 執行至 main() function 時就會停止執行。設定好 Debug Configuration 後，就可在 DS-5 的 DS-5 debug 面板中使用 PA\_CortexM4 來連接 PA 並進行開發及除錯。

當 PA 軟體及 HDL simulator 啟動完成後，利用 DS-5 連接並除錯

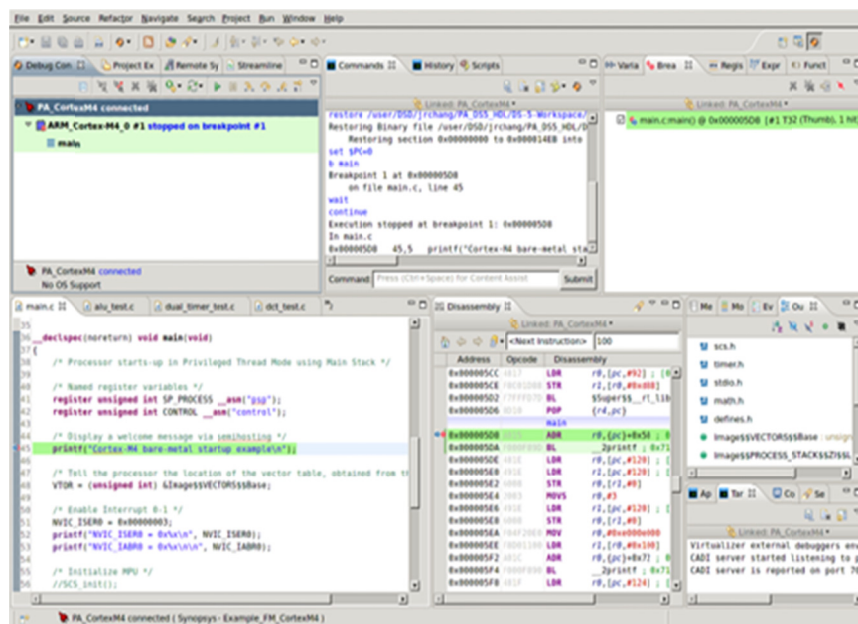
- F. 在系統選單選擇 Window -> Open Perspective -> DS-5 Debug 以切換至 DS-5 Debug 版面。於左上方 Debug Control 視窗中可看到 PA\_CortexM4 設定，點選後按滑鼠右鍵選 “Connect to Target”連接 PA 平台。



G. DS-5 連接 PA 後會將 startup\_Cortex-M4.axf 及 startup\_Cortex-M4.bin 載入 PA 平台，設定停止點至 main() function，並將 CPU 停在 0x0 的地方。



使用 F8 或是 Debug Control 的 Continue 按鍵執行，程式將會停在 main() function。



利用 F5、F6 或是 Debug Control 的 Step、Next 按鍵單步執行程式，便會看到程式執行，相關資訊將利用 semi-host 的機制列印至 PA 的 Console 視窗。

- [illegible]



## 4. 附錄

### 4.1 系統環境需求

硬體：

具備 4GB 以上記憶體之 x86/64 電腦。因 DS-5 設計上的限制，一部電腦只能同時執行一組 PA/DS-5 project，因此請儘量不要使用伺服器。

作業系統：

- 升級過的 RHEL 5 Linux 或完全相容之 Linux 版本
- 建議使用 CentOS 6.5 32/64 位元版

Synopsys Platform Architect (PA)版本:

- H-2013.06SP1

ARM DS-5 版本:

- d5.150018

HDL 模擬工具版本 (已確認相容之版本)

- Cadence INCISIV: 12.20.012
- Synopsys VCS: G-2012.09-SP1

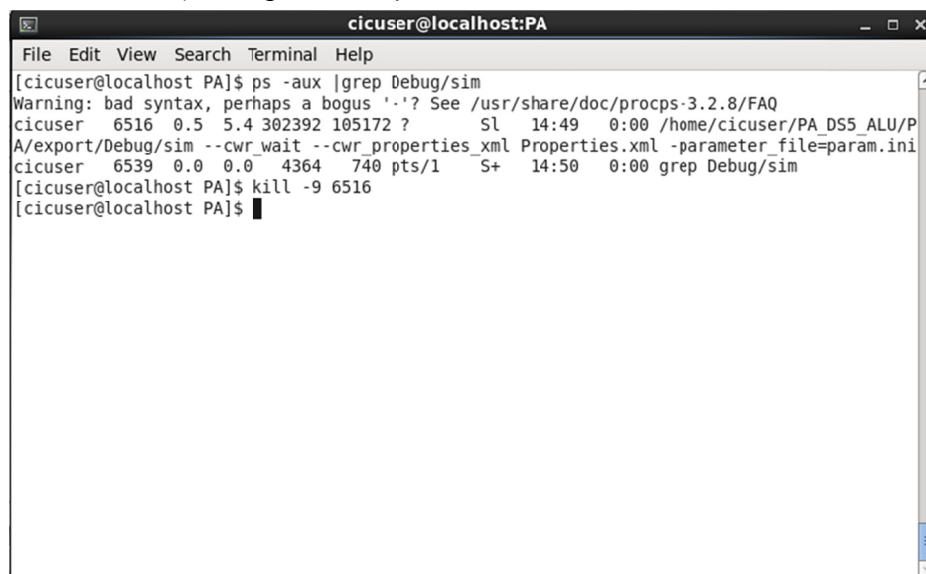
HDL 除錯工具版本

- Synopsys Verdi/nLint:2013.07

### 4.2. 刪除不正常程序的方式

#### A. 刪除不正常的 PA 程序

在 Terminal 鍵入指令 `ps -aux | grep Debug/sim`，找出 process id。接著鍵入指令 `kill -9 xxxx (xxxx: process id)`。



```
cicuser@localhost:PA
File Edit View Search Terminal Help
[cicuser@localhost PA]$ ps -aux | grep Debug/sim
Warning: bad syntax, perhaps a bogus '..'? See /usr/share/doc/procps-3.2.8/FAQ
cicuser  6516  0.5  5.4 302392 105172 ?        Sl   14:49   0:00 /home/cicuser/PA_DS5_ALU/P
A/export/Debug/sim --cwr_wait --cwr_properties_xml Properties.xml -parameter_file=param.ini
cicuser  6539  0.0  0.0  4364   740 pts/1    S+   14:50   0:00 grep Debug/sim
[cicuser@localhost PA]$ kill -9 6516
[cicuser@localhost PA]$
```

B. 刪除不正常的 DS-5 程序

在 Terminal 鍵入指令 `ps -aux | grep model_shell`，找出 process id。接著鍵入指令 `kill -9 xxxx (xxxx: process id)`。



### 4.3. 啟動 PA 分析功能 (選擇)

- A. 在 PA\_DS5\_ALU/ 目錄執行指令 `source pa_setup.csh` 以設定環境參數，切換至 PA\_DS5\_ALU/PA/ 目錄，執行指令 `pct&` 啟動 PA 主程式 (Platform Creator)；從 pct 功能選單 File->Open Project->Project\_01\_HDL.xml 開啟虛擬平台檔案。
- B. 從 pct 功能選單 Simulation->Project Settings->Build Settings 設定參數。將“Debug Level”設為“Maximum (-g3)”，啟動“Enable Instrumentation”。
- C. 在工具列上，按下“Debug”圖標(蟲狀圖案)右側選單，選擇“Debug in SystemC Explorer”，啟動 SystemC Explorer。
- D. 啟動 SystemC Explorer 後，在該軟體的工具列上按下“Suspend”圖標。這時從系統選單選擇 Analysis -> Configure 叫出 Configure Analysis 視窗。接著勾選“Bus Analysis”並按下其後方的“...”圖標，此時會出現“Configure Attributes”視窗。修改“Interval”參數為 "100 ns"，接著按“OK”結束設定。
- E. 在 Terminal 視窗執行指令“`vpa&`”啟動 Virtual Prototype Analyzer，啟動後 vpa 會自動接上目前正在進行的模擬。在 vpa 系統選單選擇 Debug -> Load Image，會出現“Select Image to Load”視窗，設定“Image file”為 PA\_DS5\_ALU/DS-5-Workspace/dual\_timer\_test.axf，接著按“Load Image”載入測試軟體。
- F. 回到 SystemC Explorer。連續點按兩次工具列上的“Resume Suspended Simulation”圖標，測試軟體便開始執行。當 Console 輸出顯示軟體執行完成後，先按下 Suspend 圖標，再按下“Terminate current simulation”圖標結束模擬並把分析資料寫入資料庫。
- G. 重新回到 pct 並再次執行 debug，但是此次選擇“Debug in VPExplorer”。啟動 VPExplorer 後，從系統選單選擇 Analysis -> Configure 叫出 Configure Analysis 視窗。從左側樹狀功能列表清單中選擇 Software Analysis->HARDWARE.ARM，右側會出現連串的分析選項。勾選“Function Trace”接著按“OK”結束設定。
- H. 重複步驟 5，啟動 vpa 並載入軟體。
- I. 回到 VPExplorer。展開左側“Design Objective/Monitor Name”樹狀列表清單下的 Current Simulation->HARDWARE，確認 BusAnalysis 及 ARM->Function Trace 選項都有勾選。點按工具列上的“Resume Suspended Simulation”圖標，測試軟體便開始執行。當 Console 輸出顯示軟體執行完成後，先按下“Suspend current simulation”圖標，再按下“Stop current simulation”圖標結束模擬並把分析資料寫入資料庫。
- J. 從 Design Objective/Monitor Name 清單，以滑鼠左鍵擊點兩次 BusAnalysis 及 Function Trace 選項，便可將分析資料送到右側的“Chart View”視窗。透過 Function Trace 和 BusAnalysis 的對照，可以觀察軟體執行於不同階段時，

系統的運行效能。請交互以下列方式選定適當觀察區間：

- 拉動卷軸
- 在圖表上選擇觀察起點，按滑鼠左鍵不放直到觀察終點再放開
- 按放大及縮小圖標(含有+/-的放大鏡)，

並按圖表左側每個觀察項目選項的上下箭頭標示，以切換觀察項目。

- K. 若在模擬結束後檢視觀察結果，請在 Terminal 執行“vpexplorer&”啟動 VPEXplorer，再從系統選單 File->Open Simulation Results 選擇 \*.cwrrs 檔案(預設放在 export/analysis\_results/latest)。

