**國立清華大學**

**系統晶片設計**

**SOC Design**

一張含有 文字, 筆跡, 字型, 圓形 的圖片

自動產生的描述

**Lab 1**

**系所級:電子所二年級**

**學號:111063548**

**姓名:蕭方凱**

**指導老師:賴瑾教授**

目錄

[I. Xilinx Tools Installation 3](#_Toc146465850)

[1. Vitis Installation 3](#_Toc146465851)

[2. Vitis\_HLS Installation 3](#_Toc146465852)

[3. VIVADO Installation 3](#_Toc146465853)

[II. Vitis\_HLS 4](#_Toc146465854)

[1. Multiplication.cpp 4](#_Toc146465855)

[2. Multiplication.h 4](#_Toc146465856)

[3. MultipTester.cpp 5](#_Toc146465857)

[III. VIVADO 7](#_Toc146465858)

[1. 匯入vitis\_hls的IP (MULTIP\_2NUM): 7](#_Toc146465859)

[2. 加入ZYNQ7 Processing System並設定PLL Fabric clock為100MHz: 7](#_Toc146465860)

[3. 加入由vitis\_hls匯出的IP multip\_2num，並run block automation 8](#_Toc146465861)

[4. Run Connection Automation 8](#_Toc146465862)

[5. Create HDL Wrapper 8](#_Toc146465863)

[6. Generate FPGA Bitstream 9](#_Toc146465864)

[IV. PYNQ/Host Program 11](#_Toc146465865)

[1. 租借online FPGA 11](#_Toc146465866)

[2. 開啟jupyter notebook，並上傳Multip2Num.bit, Multip2Num.hwh, Multip2Num.ipynb 11](#_Toc146465867)

[3. 執行Multip2Num.ipynb 12](#_Toc146465868)

[V. Discussion 13](#_Toc146465869)

# Xilinx Tools Installation

## Vitis Installation

一張含有 文字, 螢幕擷取畫面, 軟體, 多媒體軟體 的圖片

自動產生的描述

## Vitis\_HLS Installation

一張含有 文字, 螢幕擷取畫面, 軟體, 多媒體軟體 的圖片

自動產生的描述

## VIVADO Installation

一張含有 文字, 螢幕擷取畫面, 軟體, 陳列 的圖片

自動產生的描述

# Vitis\_HLS

## Multiplication.cpp

一張含有 文字, 螢幕擷取畫面, 軟體, 多媒體軟體 的圖片

自動產生的描述

這是一c++語言，其功能為主要為將int32\_t型別的輸入參數n32In1, n32In2做乘積運算，並將運算結果指向pn32ResOut所指向的記憶體位置，而不是只有指向指標pn32ResOut。

而#pragma HLS INTERFACE s\_axilite是一種介面類型，將函式連接到AXILite介面，用於FPGA與外部通信的協議。

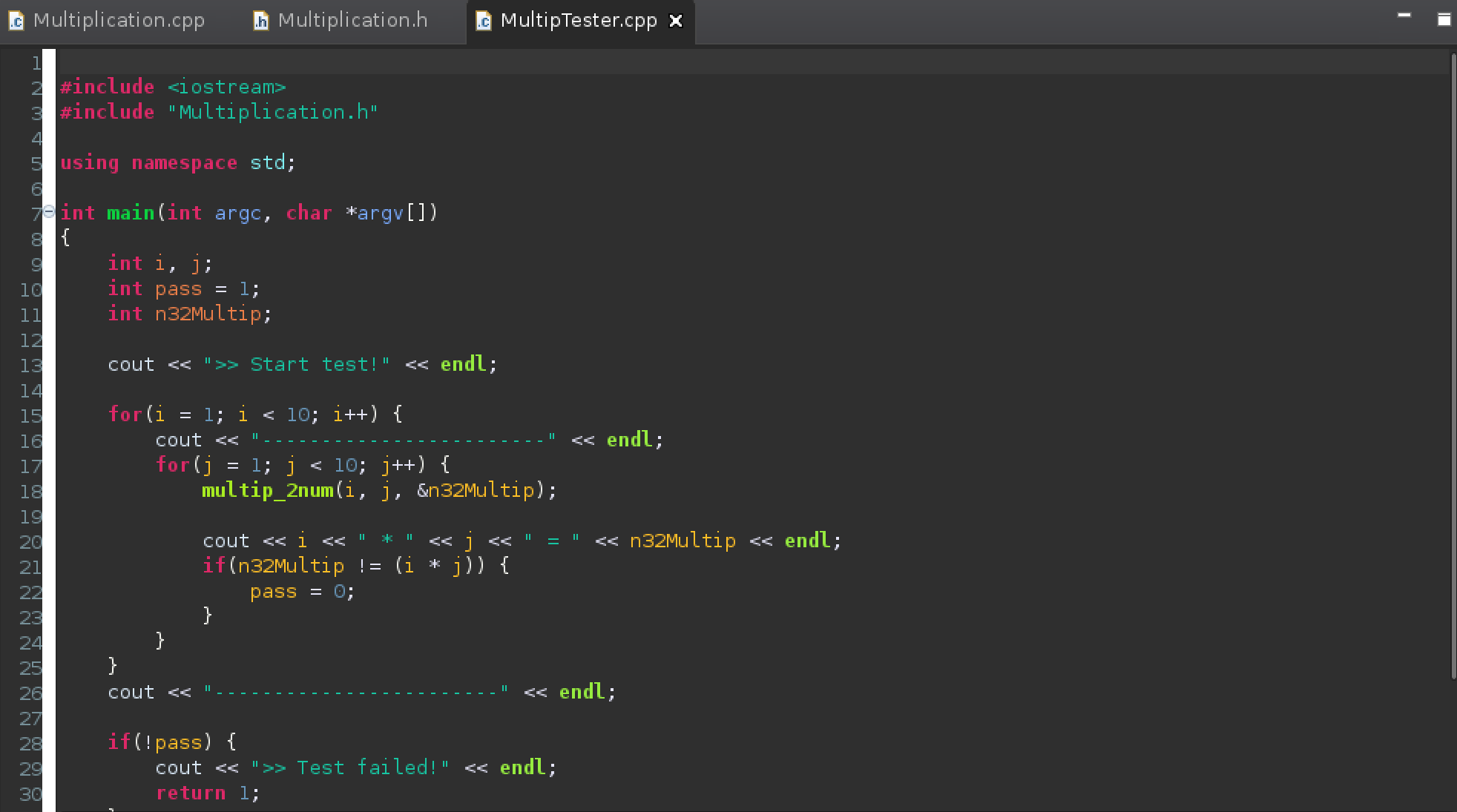
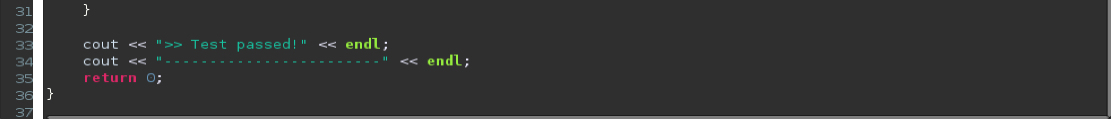
Multiplication.h

一張含有 文字, 螢幕擷取畫面, 軟體, 陳列 的圖片

自動產生的描述

此圍標頭檔，標頭檔是程式與程式之間的介面，以便多個.cpp程式檔案可以共享和使用這些元素，而ifndef&define兩行是為確保.h標頭檔只會執行一次。

MultipTester.cpp



此為c++語言所寫出類似testbench的功能，透過兩個for迴圈搭配I, j

變數得到輸入pattern，接著去驗證multip\_2num函式(輸入i,j)的運算結果是否等於i\*j，只要有一個不相等，pass就會變成0，導致顯示Test failed。

當三份檔案(multipicaiton.cpp, multiplication.h, multiptester.cpp)都準備好後，便可以開始執行simulation→synthesis→cosimulation。要特別注意的是在跑cosimulation協同模擬時，必須要將「#pragma HLS INTERFACE ap\_ctrl\_none port=return」註解，查了一些資料以後，推測是因return有可能會被當成控制訊號，進而導致模擬結果錯誤。Cosimulation只支持三種類型的ap\_ctrl\_none設計，如下:

1. combinational designs : 純粹的組合邏輯設計，不涉及時序或流水線處理
2. pipelined design with II of 1: 具有流水線架構的設計，其中 II 設置為1，表示每個時鐘週期只有一個操作被執行。這可以用於一些需要時序控制的設計
3. designs with array streaming or hls\_stream or AXI4 stream ports:

表示設計使用了特定的流式數據傳輸機制，例如 Vivado HLS 中的數組流或 hls\_stream，或者是 AXI4 流介面

執行完simulation→synthesis→cosimulation後，會產生三個檔案，csynth.rpt, multip\_2num\_csim.log, multip\_2num\_csynth.rpt。

1. csynth.rpt

此report內容為C++ 代碼轉換為硬體的結果和相關資訊，包含SW I/O Information, HW Interfaces等，舉例來說，SW I/O Information用於描述和記錄軟體的輸入和輸出（I/O）相關信息，HW Interfaces則是硬體設計中的各種接口資訊。

1. multip\_2num\_csim.log

此log檔類似執行完testbecnch(multiptester.cpp)後的輸出，其主要是模擬輸出並檢查模擬是否正確。

1. multip\_2num\_csynth.rpt

與csynth.rpt不同，此report檔是針對特定函數multip\_2num的報告，其內容包含時序分析(timing, latency)、硬體資源使用情況等。

完成IP設計後，最後export rtl以供vivado做使用，Vivado 會需要匯入由 Vitis HLS匯出的IP，另外，由C++語言轉換的RTL code會出現在〝solution1/impl/Verilog〞中的multip\_2num.v。

# VIVADO

## 匯入vitis\_hls的IP (MULTIP\_2NUM):

一張含有 文字, 螢幕擷取畫面, 陳列, 軟體 的圖片

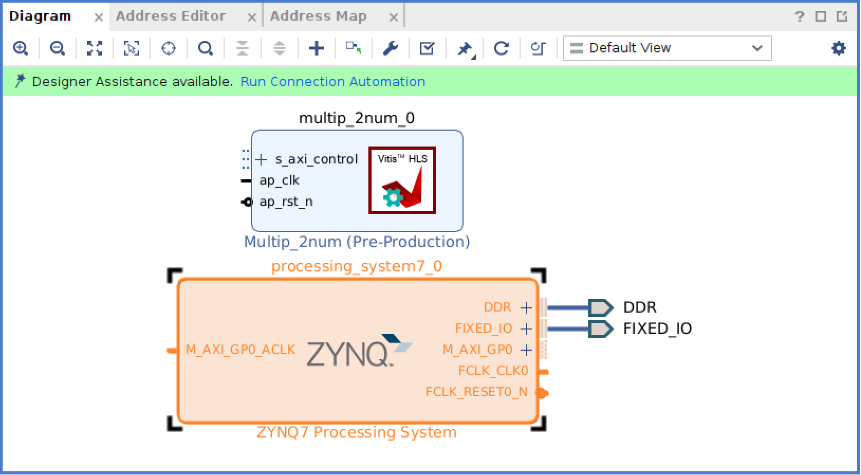
自動產生的描述

加入ZYNQ7 Processing System並設定PLL Fabric clock為100MHz:

|  |  |
| --- | --- |
| 一張含有 文字, 螢幕擷取畫面, 軟體, 陳列 的圖片  自動產生的描述 |  |

ZYNQ是一種可編程SOC，整合了 ARM 雙核 cortex-A9處理器和 Xilinx 7 系列 FPGA 架構，內部包含Processing System及Programmable Logic。至於上方設定的PLL Fabric clock，是指可程式邏輯區域中的相位鎖定迴路所產生的時鐘訊號，由processing system產生並提供給programming logic使用，當提高frequency時候，在沒有timing violation的情況下，可提高IP Core在硬體上的運行速度。

加入由vitis\_hls匯出的IP multip\_2num，並run block automation



Run Connection Automation

一張含有 文字, 螢幕擷取畫面, 數字, 陳列 的圖片

自動產生的描述

一張含有 文字, 字型, 數字, 行 的圖片

自動產生的描述

執行完connection autimation後可切到address editor查看Memory map

Create HDL Wrapper

這一部非常重要，Create HDL Wrapper在 FPGA 或 ASIC 設計流程中創建一個RTL的封裝（wrapper）模塊，該封裝模塊將自行設計的模塊與FPGA 或 ASIC 的外部界面進行連接。

Generate FPGA Bitstream

一張含有 文字, 螢幕擷取畫面, 軟體, 數字 的圖片

自動產生的描述

Generate完成後會產生兩個檔案，Multip2Num.bit及Multip2Num.hwh，以提供FPGA做使用。

1. Multip2Num.bit:

.bit 文件是 FPGA 的位流（Bitstream）文件，包含了將硬體設計配置到 FPGA 的資訊，包括邏輯閘配置、連接、clock設定等。一張含有 文字, 電子產品, 螢幕擷取畫面, 軟體 的圖片

自動產生的描述

1. Multip2Num.hwh:

.hwh 文件是硬體描述文件（Hardware Description File），它包含了有關 FPGA 設計的元數據（Metadata），不包含實際數據。這些元數據可能包括設計的版本、配置信息、接口定義等。

一張含有 文字, 螢幕擷取畫面, 字型, 軟體 的圖片

自動產生的描述輸入cp ./vivado\_lab1/vivado\_lab1.runs/impl\_1/design\_1\_wrapper.bit ./Multip2Num.bit

cp ./vivado\_lab1/vivado\_lab1.gen/sources\_1/bd/design\_1/hw\_handoff/design\_1.hwh ./Multip2Num.hwh後，會在soc\_lab1目錄下得到兩檔案，分別為Multip2Num.bit及Multip2Num.hwh，如下圖:

一張含有 文字, 螢幕擷取畫面, 軟體, 多媒體軟體 的圖片

自動產生的描述

# PYNQ/Host Program

## 租借online FPGA

一張含有 文字, 螢幕擷取畫面, 軟體, 多媒體軟體 的圖片

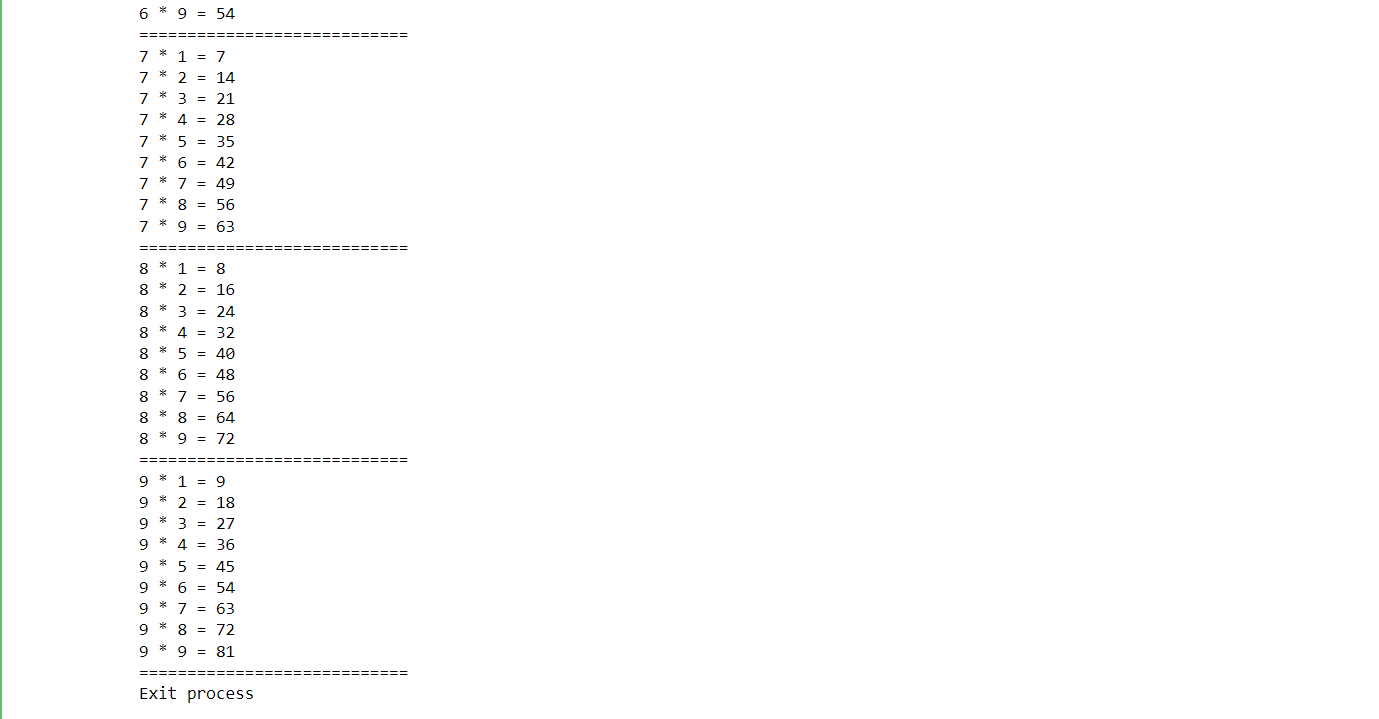
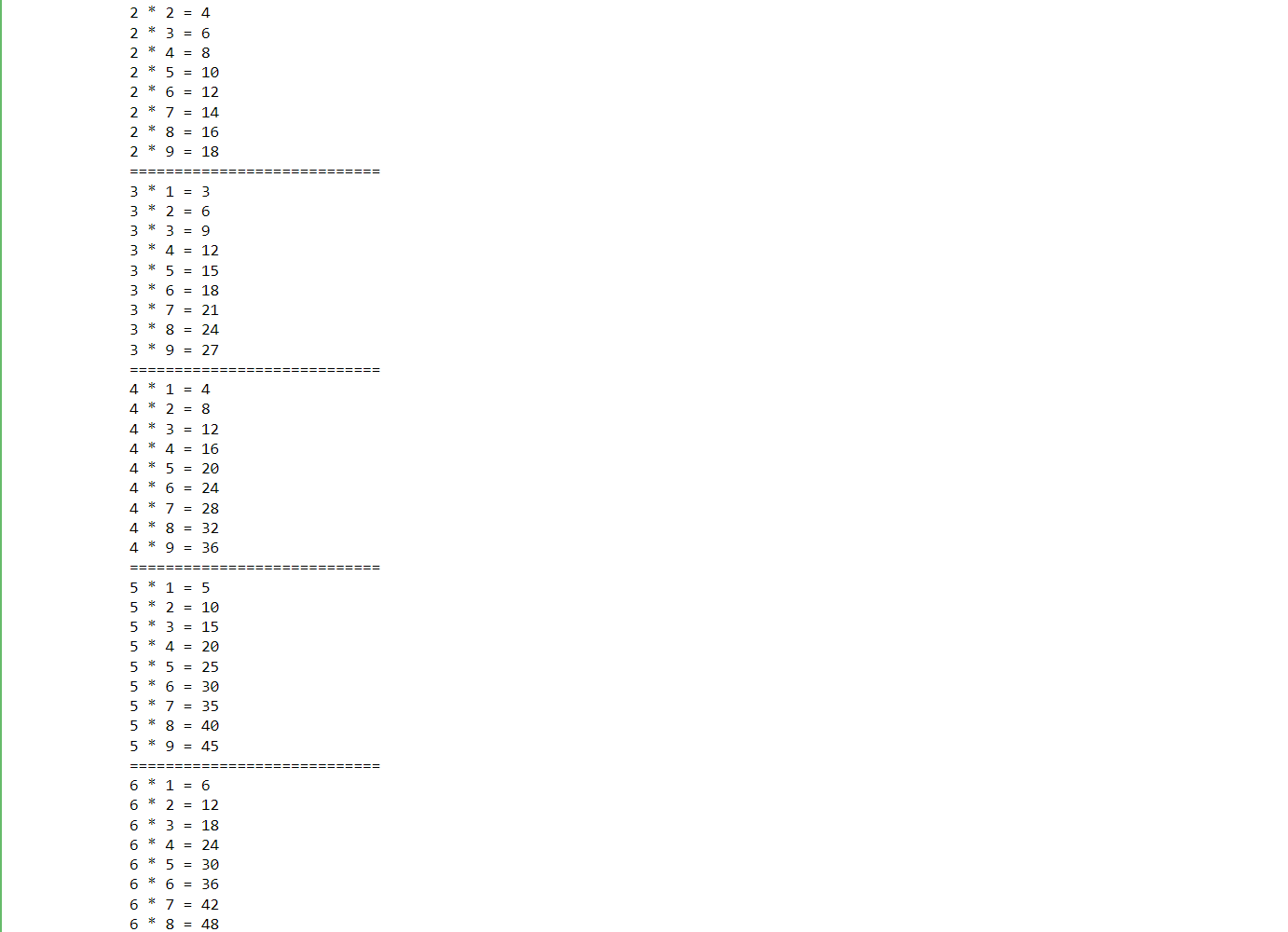
自動產生的描述

開啟jupyter notebook，並上傳Multip2Num.bit, Multip2Num.hwh, Multip2Num.ipynb

一張含有 文字, 數字, 行, 字型 的圖片

自動產生的描述

## 執行Multip2Num.ipynb



# Discussion

1. **為何#pragma HLS INTERFACE ap\_ctrl\_none port=return不註解的話會無法cosimualtion?**

ANS:

當使用interface protocol ap\_ctrl\_none時，不會將block-level I/O protocol添 加到設計中，只會有clock, reset, data ports。**沒有ap\_done信號，接收來自 ap\_return port的數據的consumer block將無法知道數據何時有效。**

1. **為何run block automation後，ZYNQ7會多出兩個port接腳(DDR, FIXED\_IO)?**

ANS:

DDR port通常用於連接外部的 DDR SDRAM 存儲器，當運行Run Block Automation時，自動生成 DDR port以支援 DDR 存儲器的連接和操作。FIXED\_IO port通常與 FPGA 的固定引腳（Fixed I/O Pins）有關，此固定引腳通常用於連接到外部硬體設備，如顯示器、按鈕、LED等，生成 FIXED\_IO 端口的目的是確保可以設置這些引腳的功能並連接。

1. **Run connection automation後，多出來的components是什麼?**

ANS:

多出來的component是processor system reset以及AXI interconnect，Processor System Reset是一個用於控制和管理處理器（通常是 ARM Cortex-A 等）的組件。負責處理器的初始化和重置，以確保處理器能夠正確執行軟體。AXI Interconnect 是一個用於連接不同硬體模塊的組件，確保數據在不同block之間可靠地流動。

1. Bitstream是什麼?.hwh檔案是什麼?

ANS:

Bitstream是FPGA 設計的二進制配置檔案，其可將 FPGA 中的可編程邏輯閘陣列配置為特定硬體功能，使FPGA得以實現特定的硬體功能。

Comment:

透過這次lab1，我學會了一套從軟體語言(c++)到FPGA上模擬的流程，使用軟體而非硬體去編成的好處在於，當電路功能很大時用軟體去描述會比硬體來的更方便更快速一些，透剁vitis\_hls層層轉換輸出得到IP，再匯入進去vivado配置clock, connections, gate, etc.，最後產生bitstream及.hwh檔案，即可在FPGA上面進行模擬觀察電路功能是否符合預期，有無正常運作。

在做LAB的過程有許多名詞是之前沒看過的，例如前面的ap\_ctrl\_none，其用意是什麼?Processing System及Programming Logic之間的關係又是什麼?有許多問題還是需要一定的基礎知識才較好理解，每個step的目的也希望能更加瞭解而不是只照個講義做，這堂課給予的實作機會讓我充分了解自己還有哪部分需要加強，希望未來可以不斷精進，更了解SOC的各種細節及設計概念。