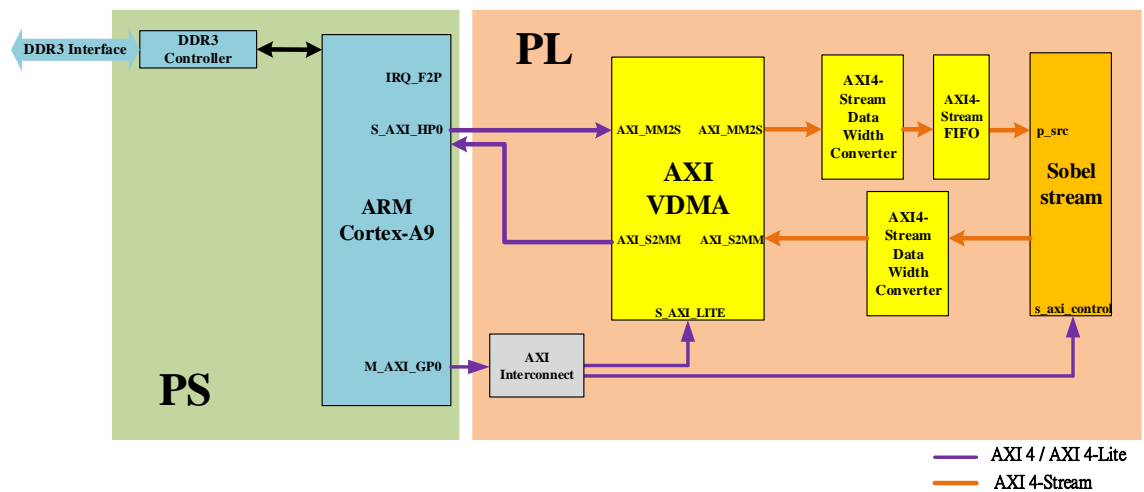


Lab 8 Sobel Filter and VDMA System

Design Introduction

本次利用一 Sobel Filter 硬體加速 SoC 平台做為舉例，系統架構圖如下。主要透過 Video DMA 做為輸出入傳輸。其中因為 Sobel 的資料接收/輸出寬度的不同，因此呼叫了 Data Width Converter IP 負責轉換資料寬度與負責平衡頻寬的 FIFO 做為傳輸上的 buffer。而本課程最後須使用 ILA 找到目標訊號，以此方式來了解 ILA 與實際 Debug 與 Trigger 訊號的方式。

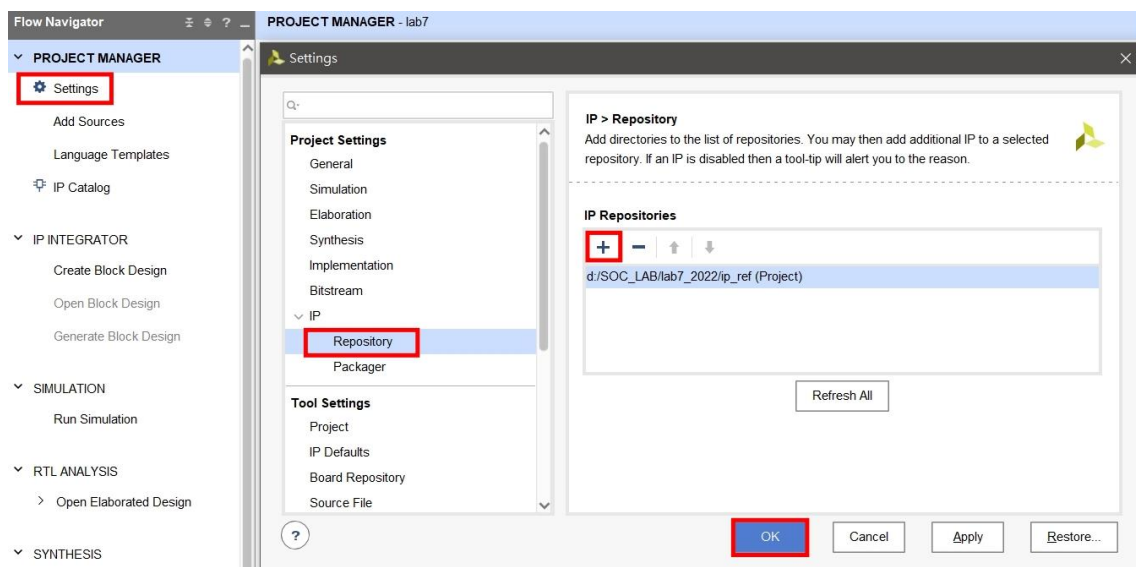
Design Overview



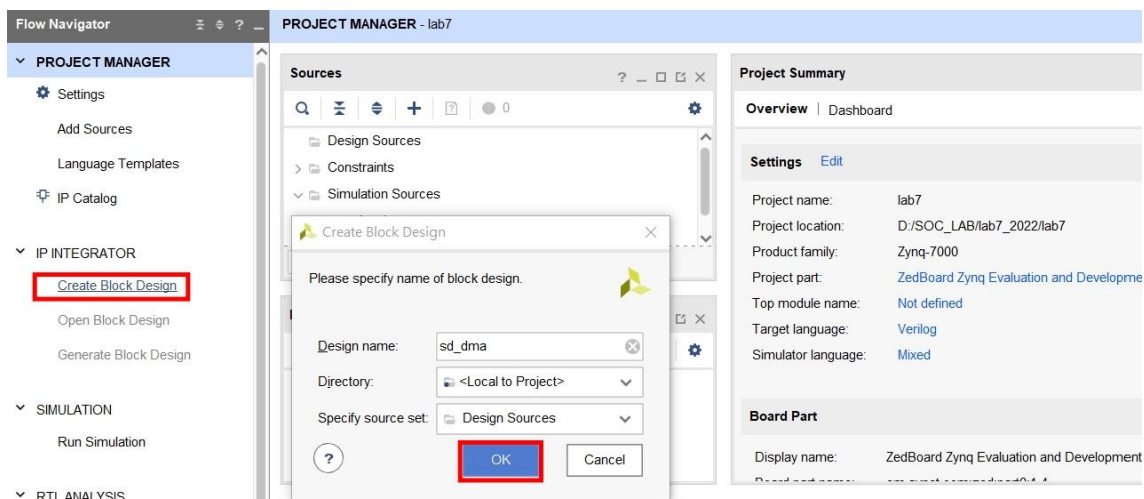
1. 建立專案

1.1 開啟 Vivado，選擇 Create Project→RTL Project→不用 Add Source、Constraint→選擇板子(Zedboard) →finish.

1.2 開啟 Setting→Repository，加入包好的 IP 專案 ip_ref 資料夾。

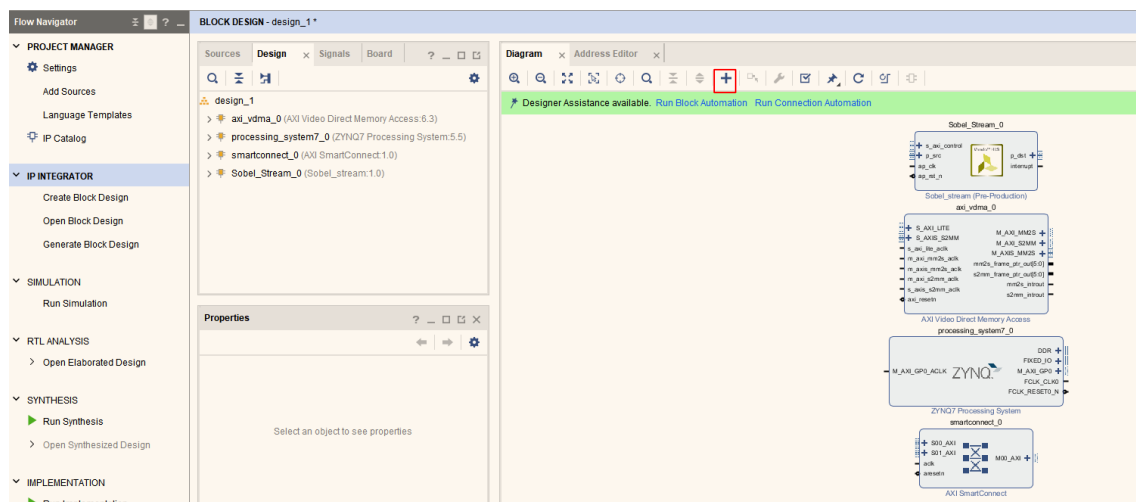


1.3 創建 Block Design，點選 Flow Navigator 的 Create Block Design.

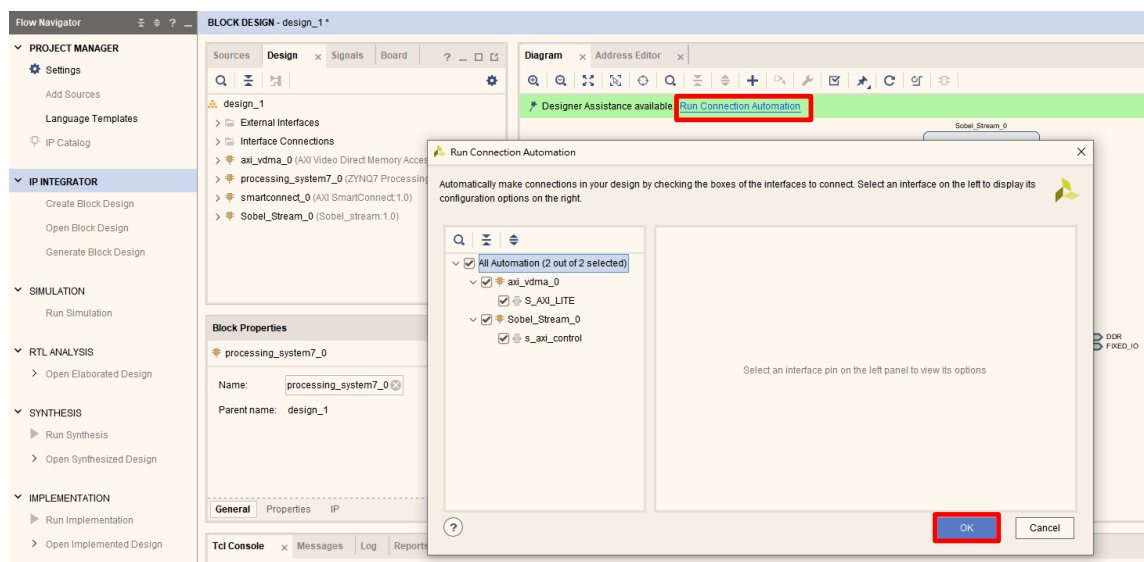


1.4 點選 Add IP，並加入以下 IP:

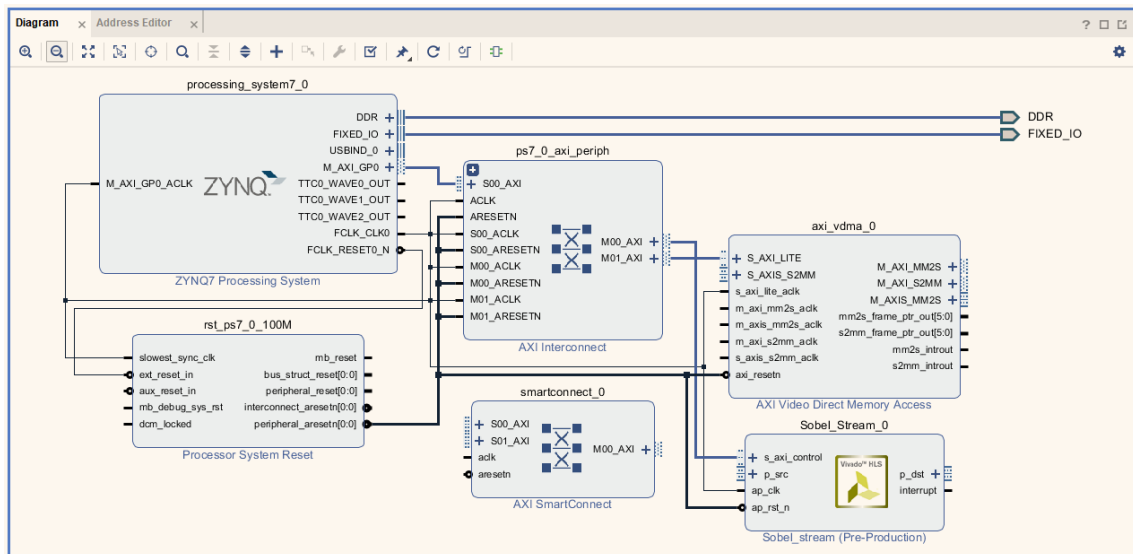
1. Zynq7 Processing System
2. AXI Video Direct Memory Access
3. AXI SmartConnect
4. Sobel_stream



1.5 點選 Run Block Automation 讓 CAD Tool 進行 IP 初始設定，再來點選 Run Connect Automation(全勾選)將基礎的線接上。

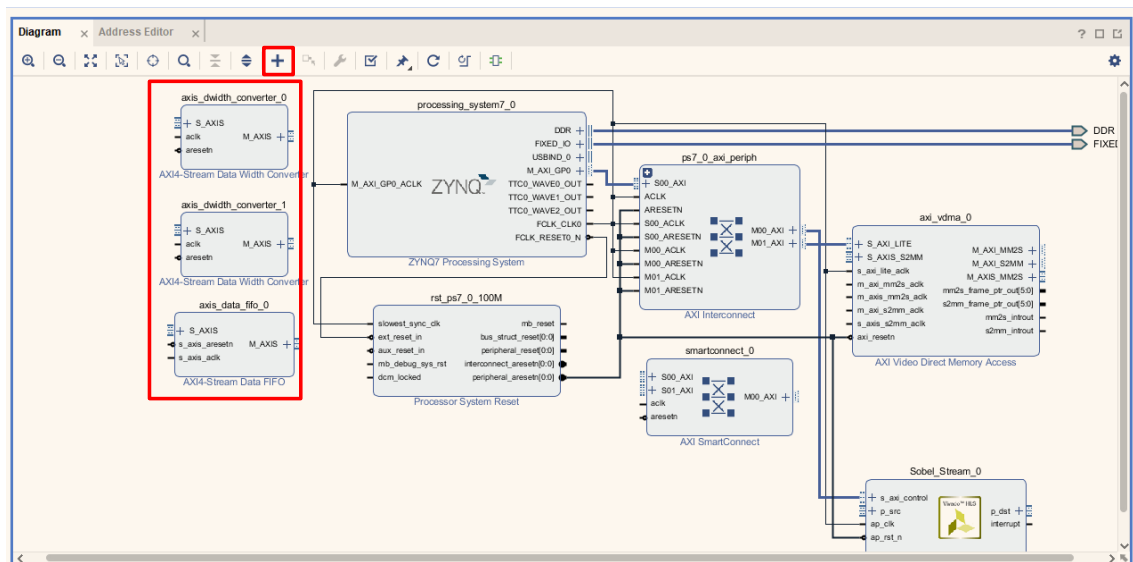


設定完成後應如下圖：



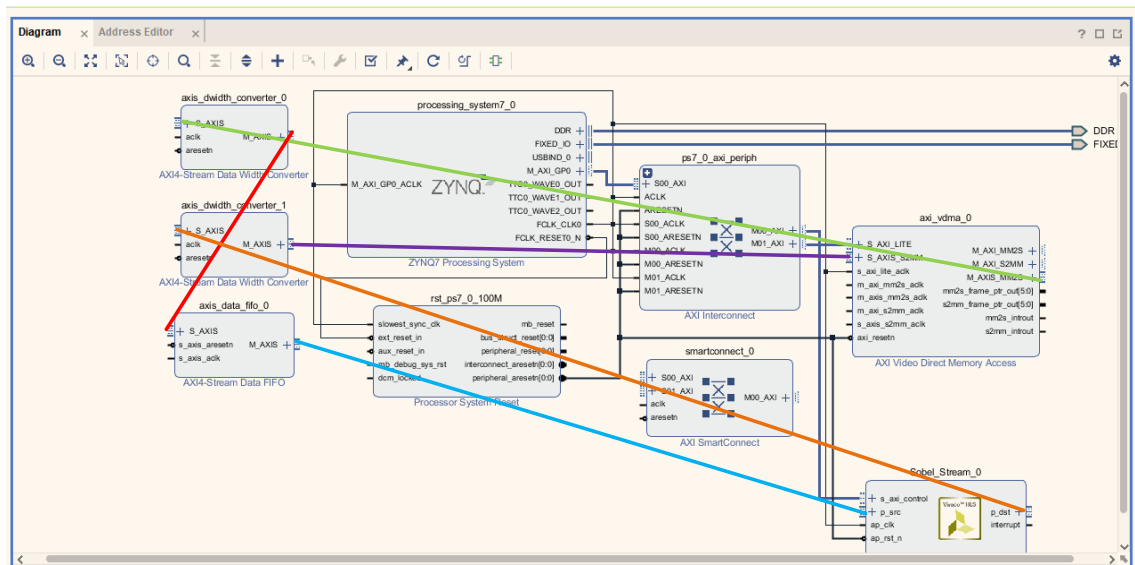
1.6 新增兩顆 AXI4-Stream Data Width Converter、AXI4-Stream Data FIFO

從 vdma 拉出資料後為 32bits，而 Sobel_Stream 輸出/入單筆資料為 8bits，因此輸出/入時需要 **AXI4-Stream Data Width Converter** 更動其 data 寬度。而同時思考到單一 clock 會從 ddr 內輸出 32bits 資料，而收取時單一 clock 僅只有 8bits 資料，此時產生頻寬不匹配。因此為了讓頻寬匹配尚需 **AXI4-Stream Data FIFO**。



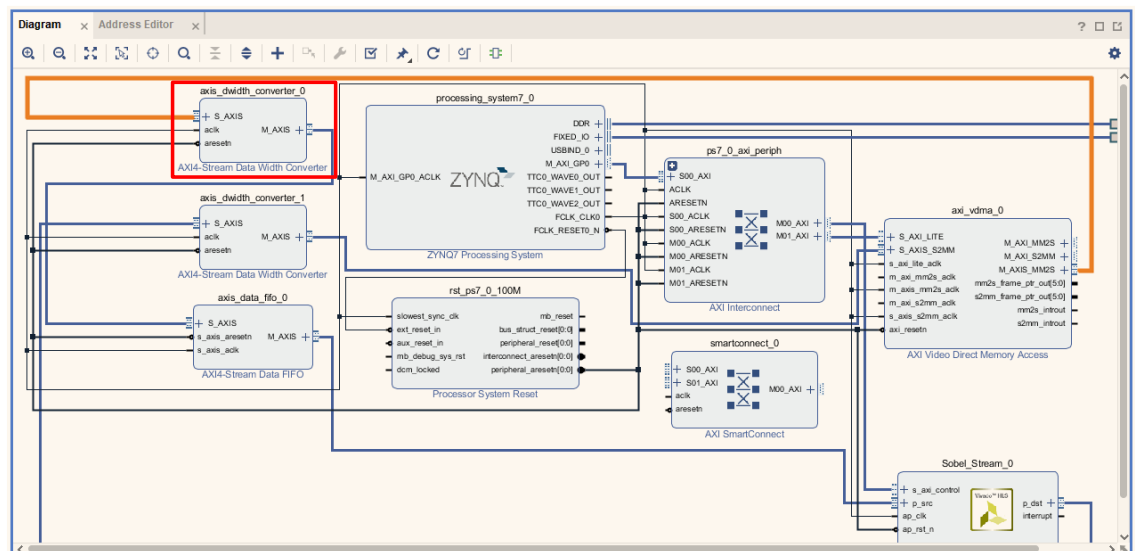
1.7 AXI-Stream Data Flow 連接

於 Block Diagram 中，將 AXI-Stream 資料流依照下圖連接。

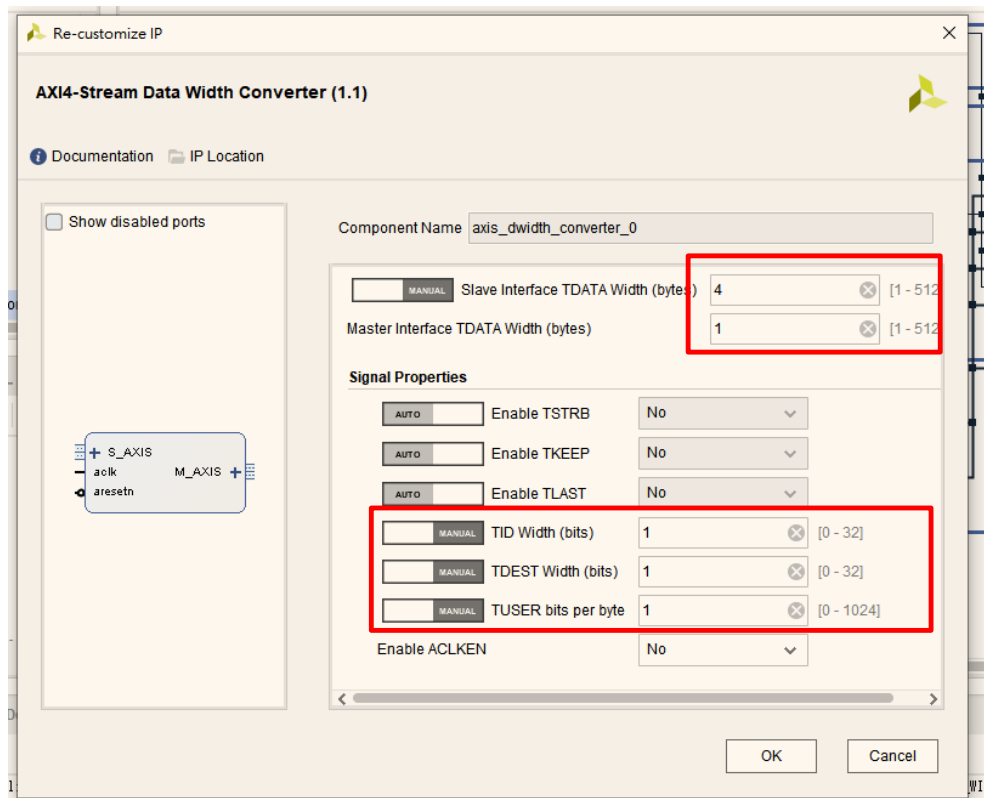


1.8 設定 Data Width Converter

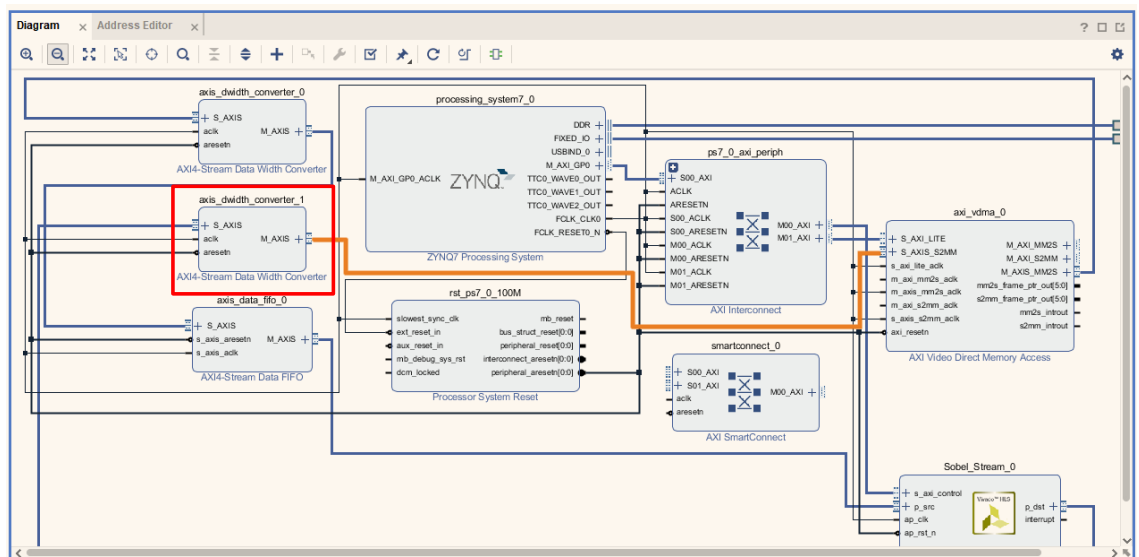
1. 點開輸入端口的 Data Width Converter 設定。



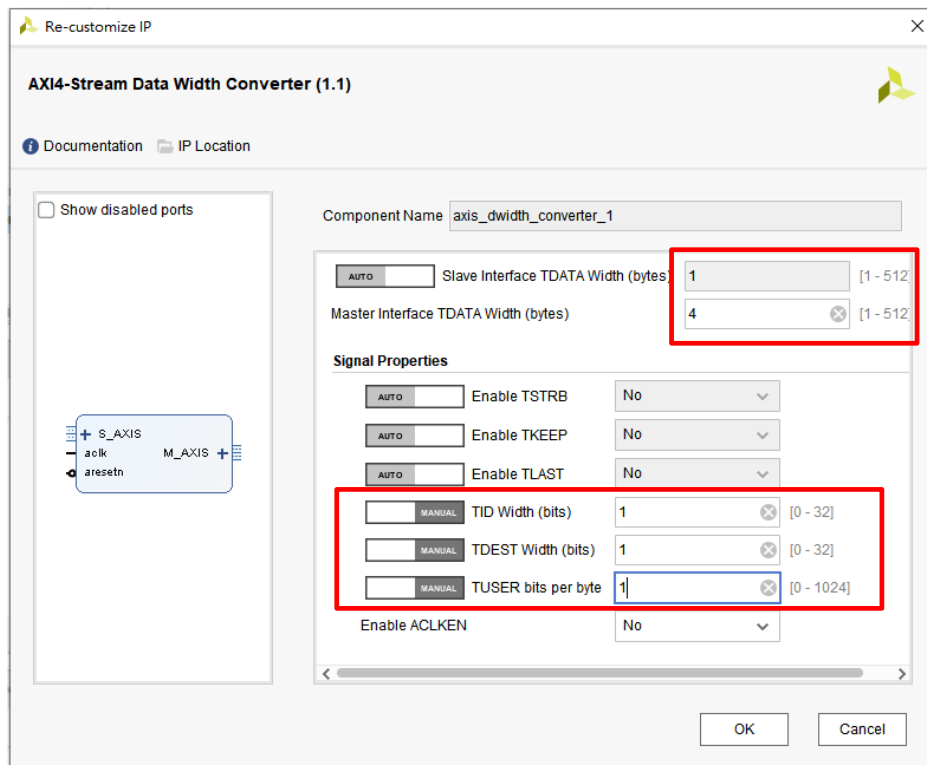
2. 並且依照下方圖片說明更改設定。



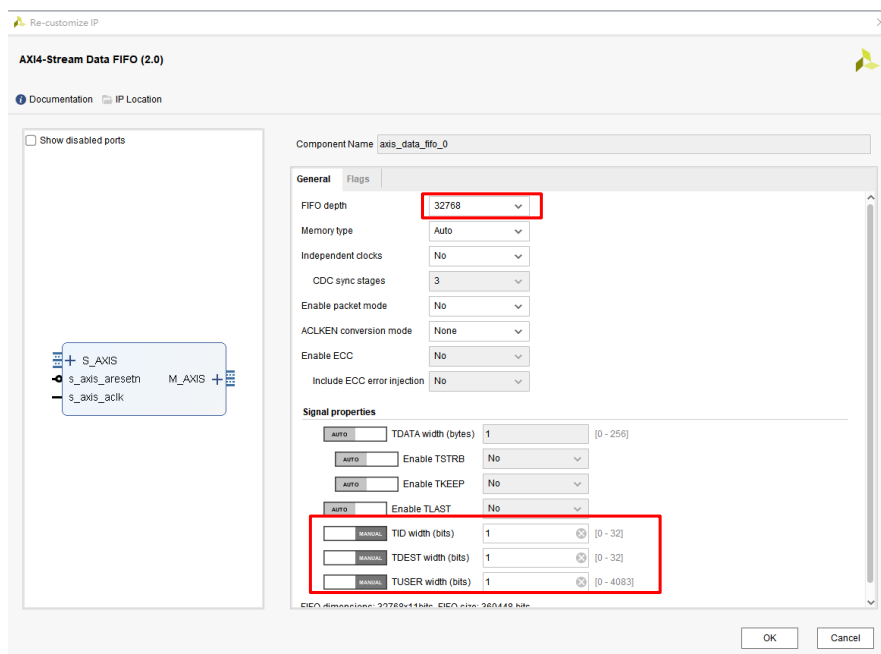
3. 點開輸出端口的 Data Width Converter 設定。



4. 並且依照下方圖片說明更改設定。

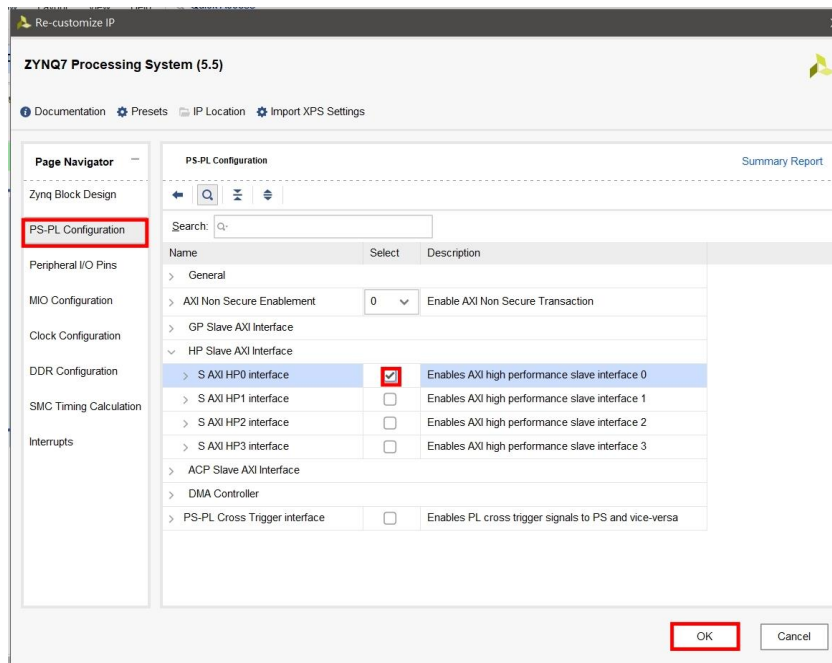


5. 設定 AXI4-Stream Data FIFO 內容如下。

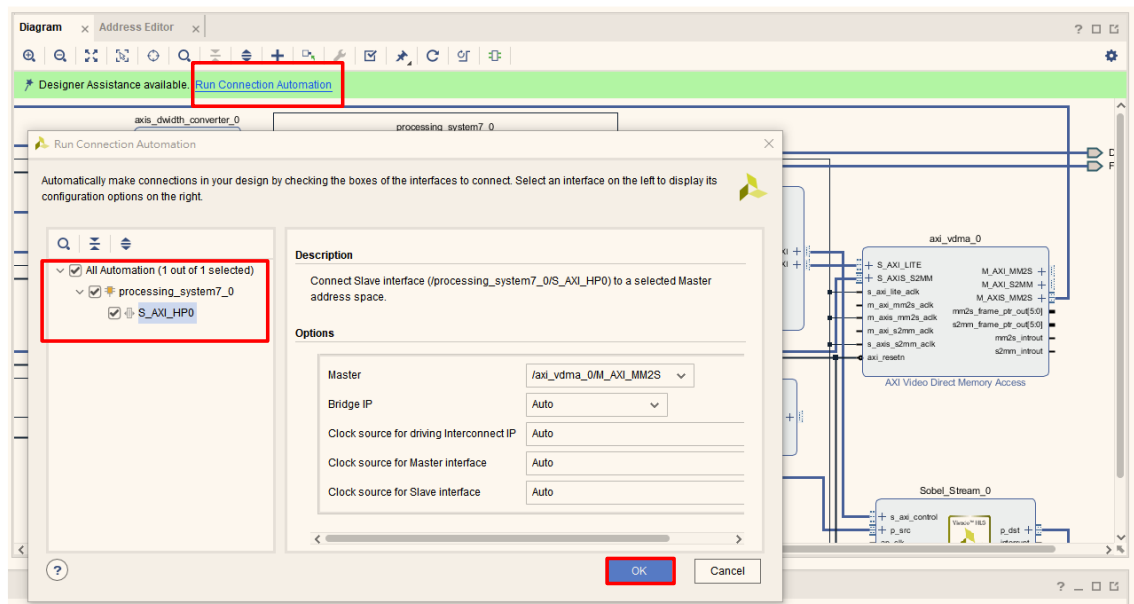


1.9 PS 端開啟與 DMA 進行資料交換的 AXI HP Interface.

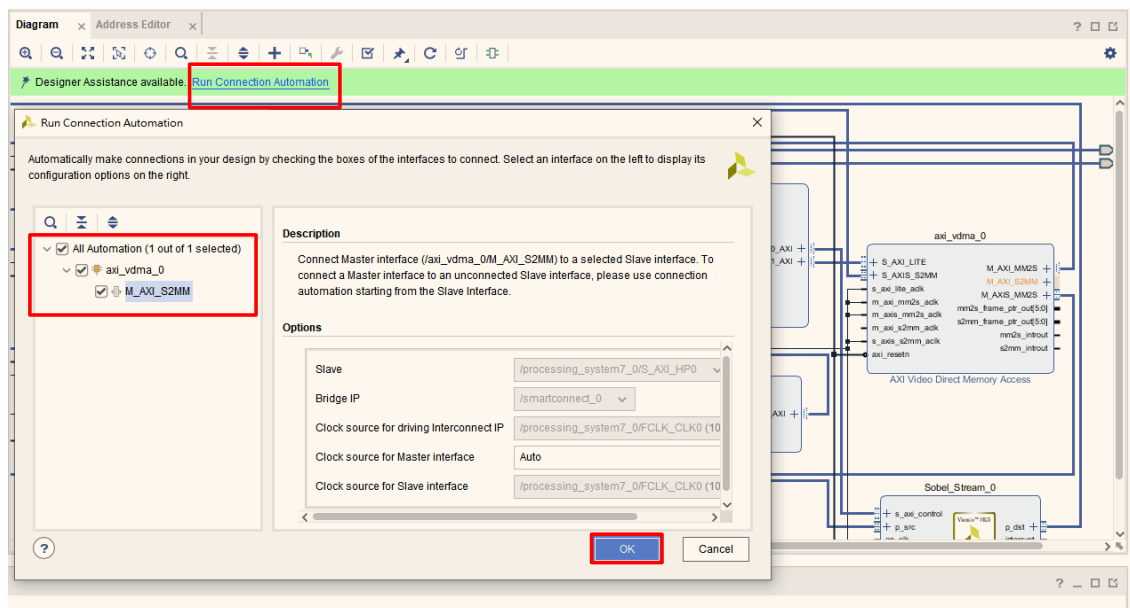
至 PS-PL Configuration 選單，將 HP Slave AXI Interface 中 HP0 勾選，完成 PS 與 VDMA 的基礎設定.



點選 Run Connection Automation.

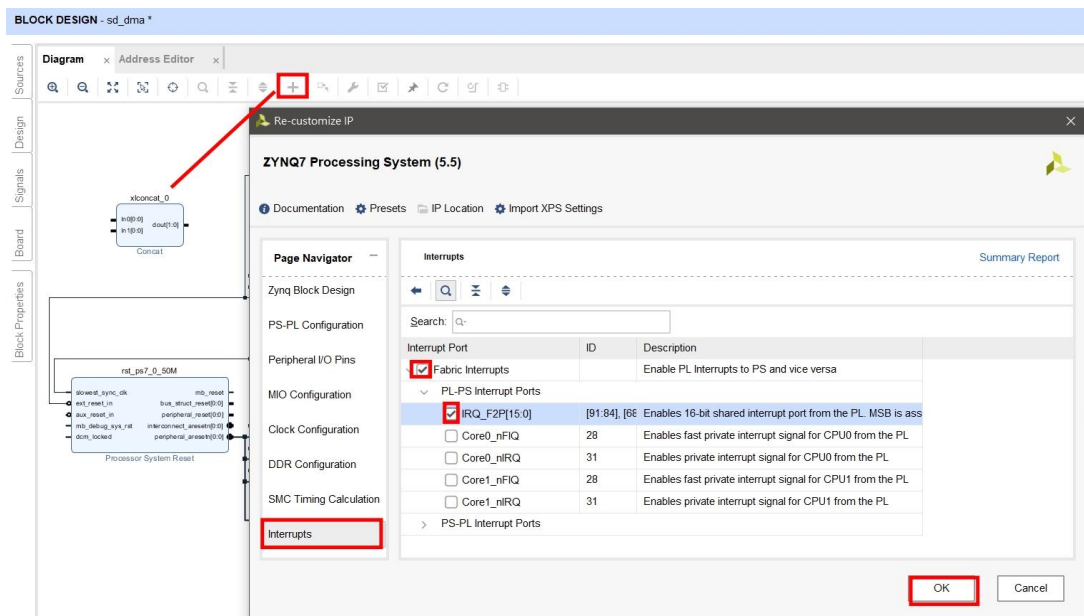


再點選 Run Connection Automation.

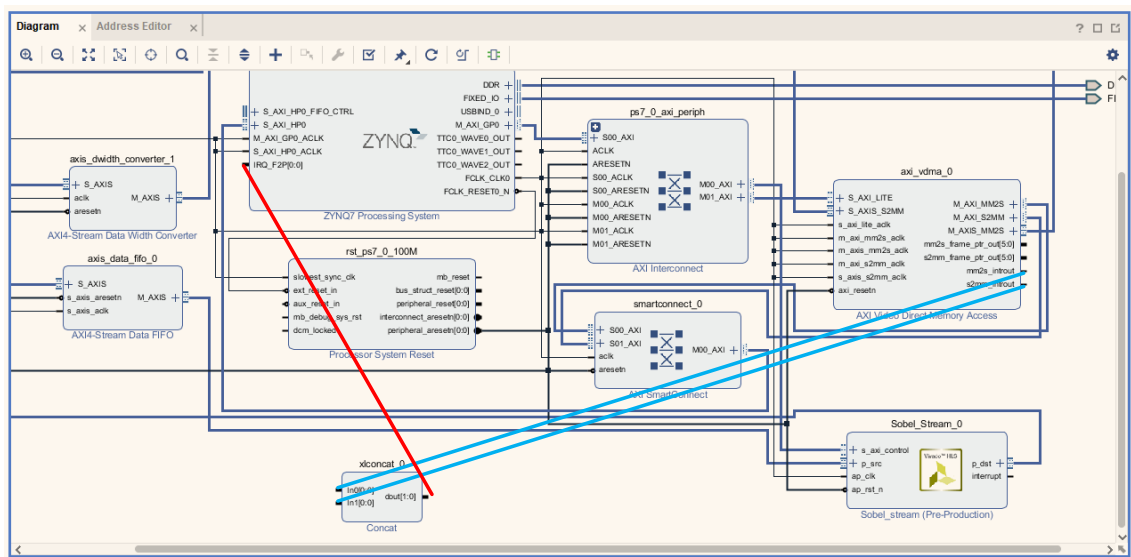


1.10 點選 Add IP 並加入 Concat，接著開啟 PS 的 IP 設定，點選 Interrupts 的 IRQ_F2P Port.

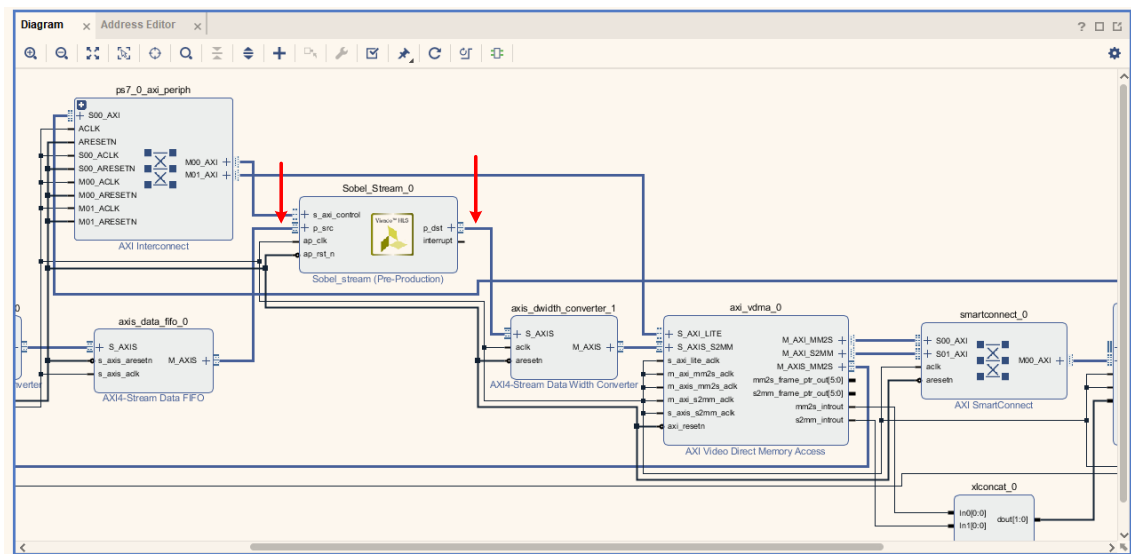
*Concat-IP 作用在當 VDMA 搬完資料時，MM2S 與 S2MM 皆會產生 Interrupt 訊號(結束訊號)，而 Concat-IP 則負責將兩訊號 Concat 並接往 PS 端。

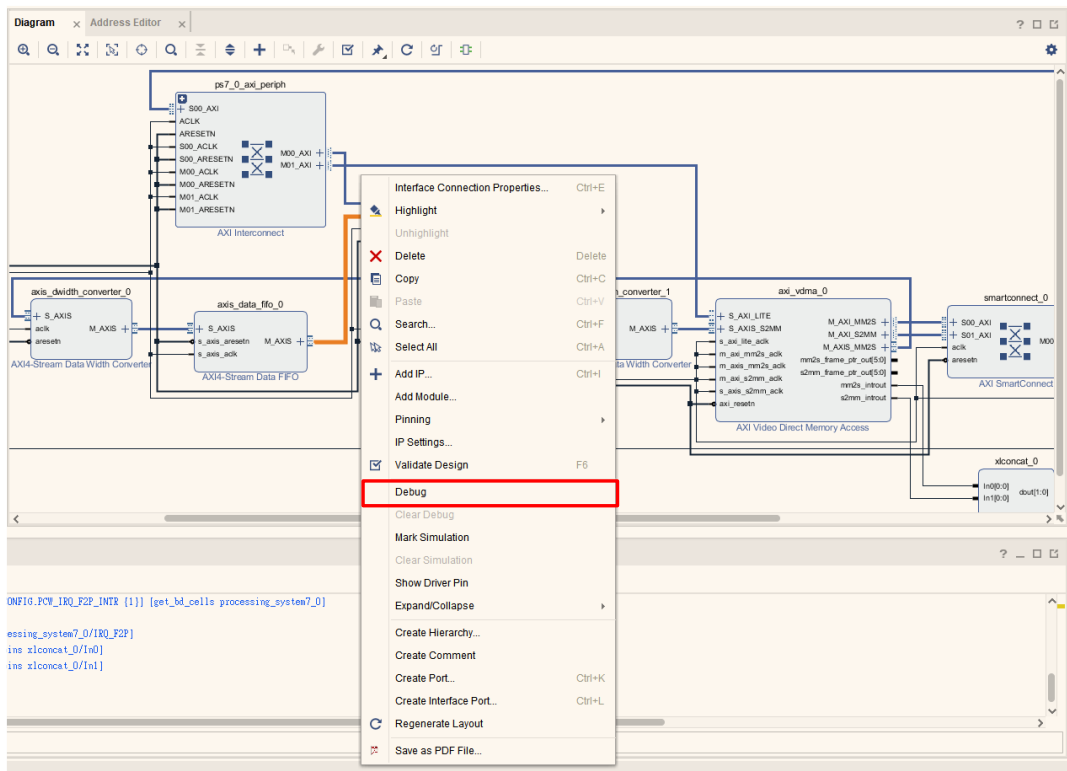


接著進行 Concat 與 PS、VDMA 的接線，如下圖。

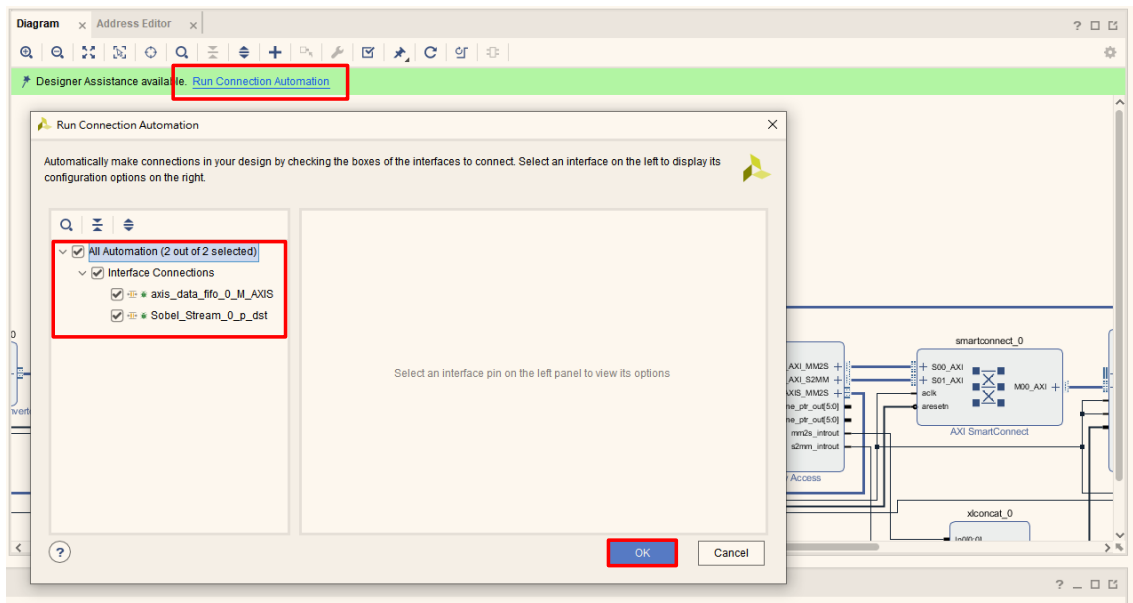


1.11 再來加入 Debug IP，點選 Add IP 並加入 ILA (Integrated Logic Analyzer).
 在 Sobel_stream 的 IP 前後線路，對兩條線分別右鍵選取 Debug。

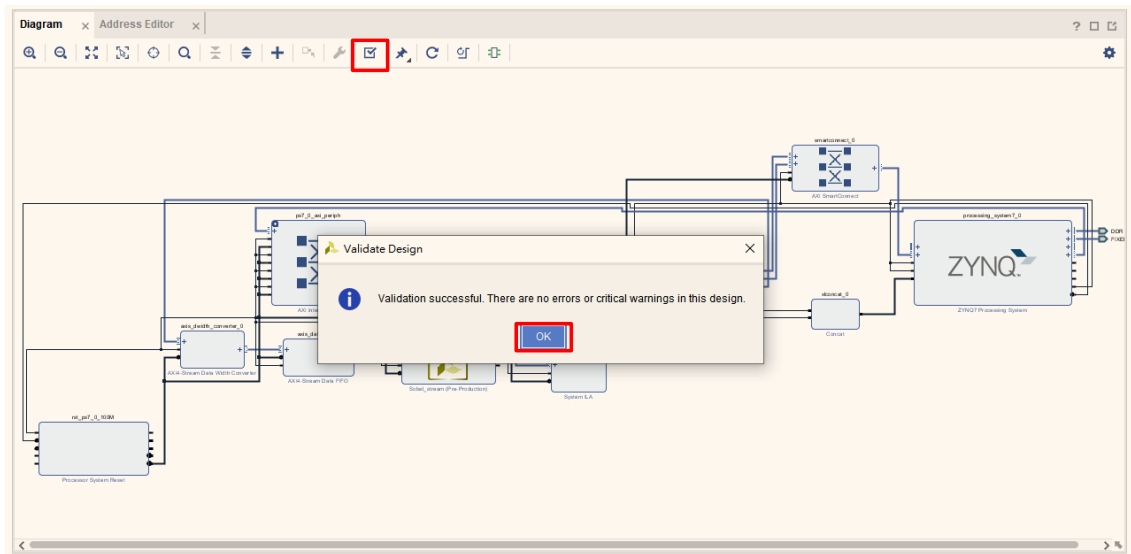




點選 Run Connection Automation.

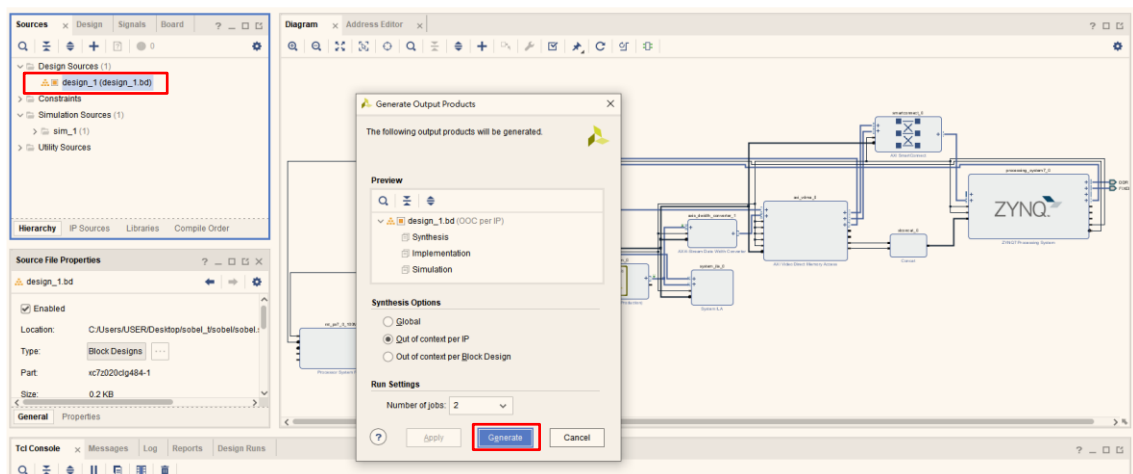


1.12 檢查 Design 目前有無問題，按下 Validate Design.

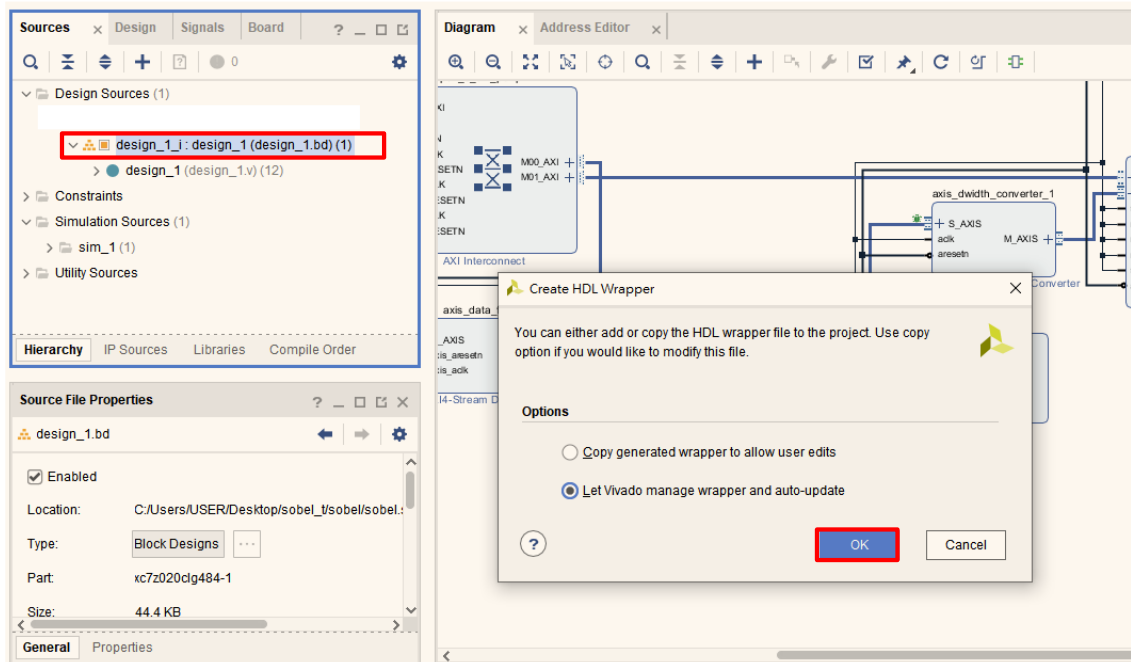


1.13 進行 Generate Output Products

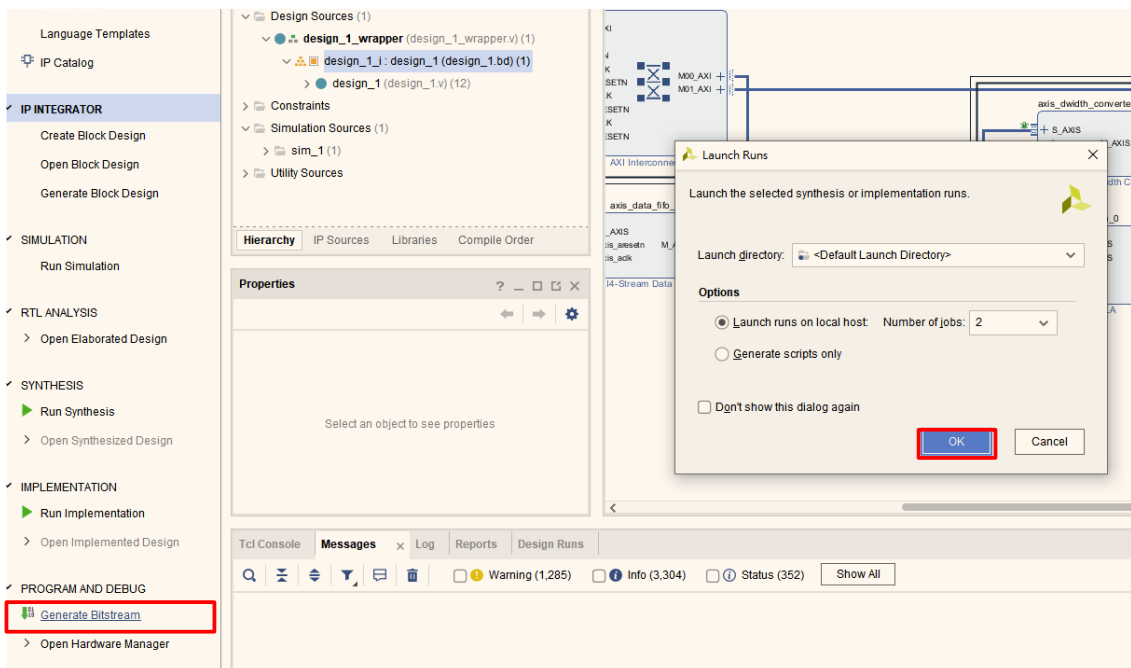
1. 對 Source window 的 Block Design 右鍵點選 **Generate Output Products**.



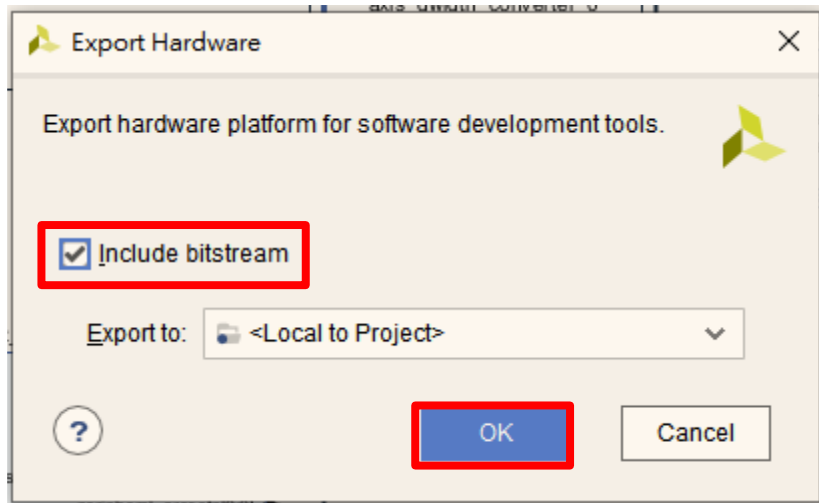
2. 對 Source window 的 Block Design 右鍵點選 **Create HDL Wrapper**.



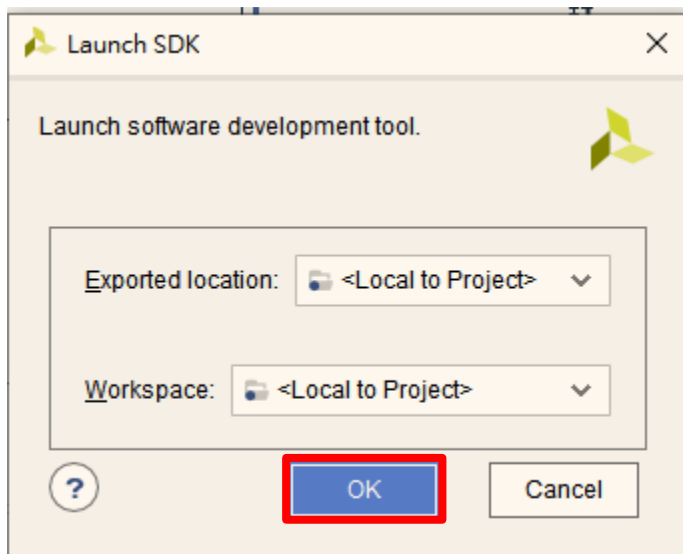
3. 於 Flow Navigator 處選擇 **Generate Bistream**.



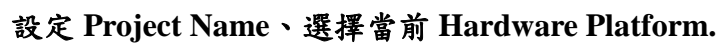
1.14 接著匯出 Bistream，File→Export.. →Export Hardware，跳出窗格後選 Include bistream.



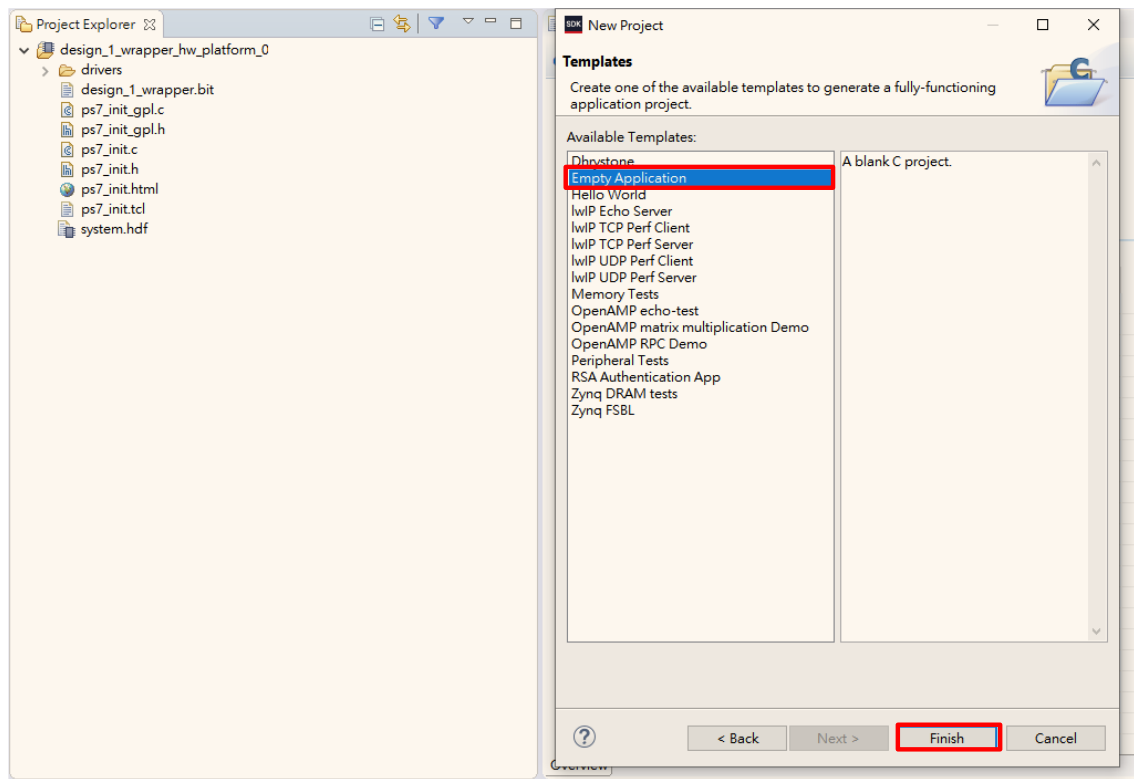
1.15 開啟 SDK，File→Launch SDK.



2.1 開啟 SDK 後，建立新 Project.

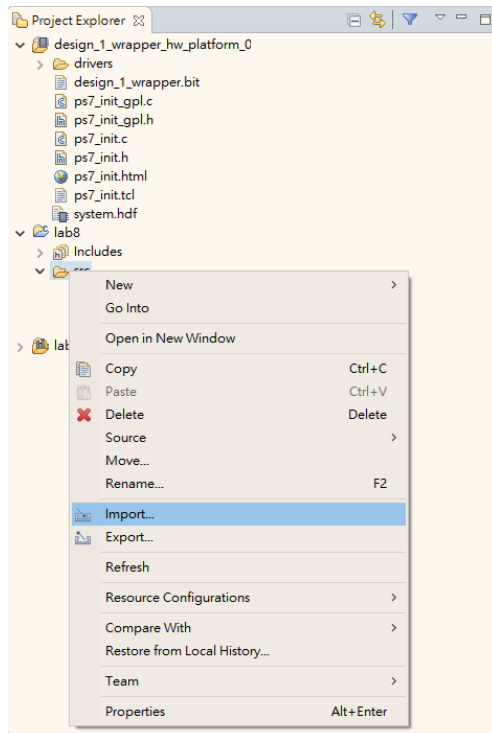


建立 Empty Application

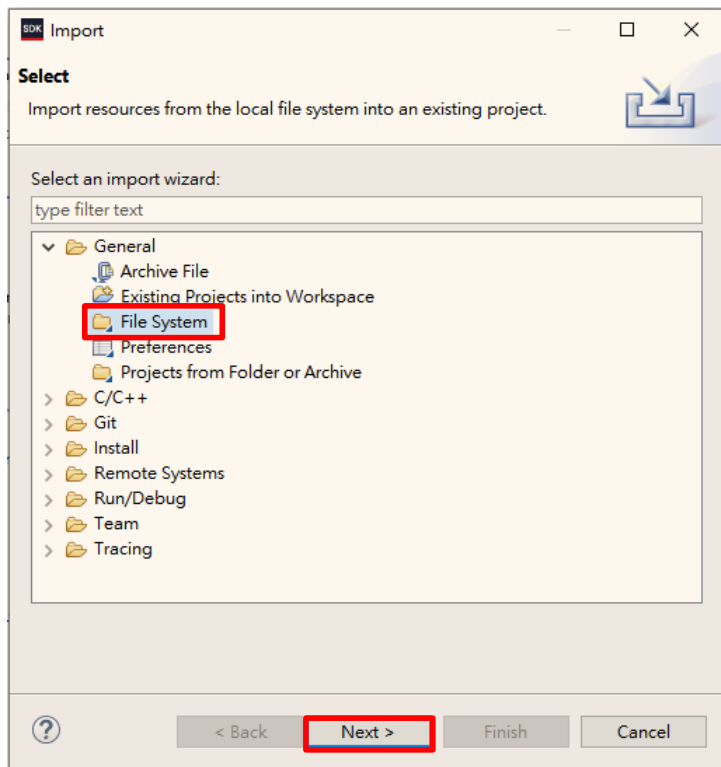


2.2 Load 進本 Lab 所需的 C code

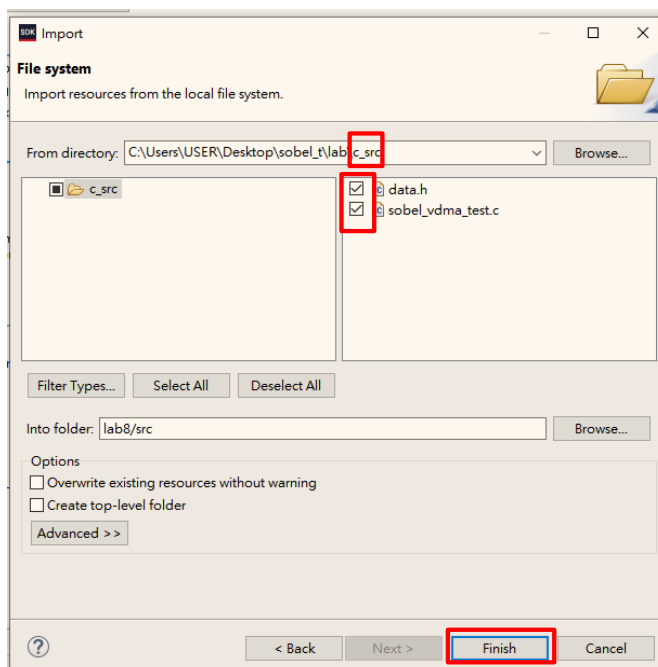
1. 在 Project Explorer 下，對新增的 lab8 Project 下的 src 資料夾右鍵點選 Import.



2. 選擇 File System

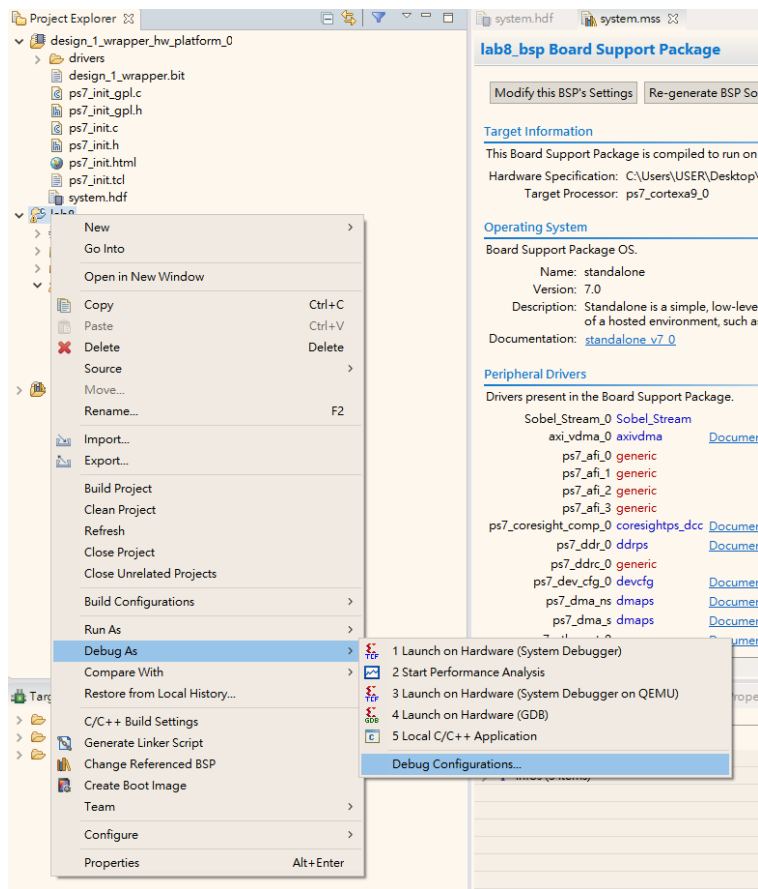


3. 將 lab 資料夾中的 c_src 檔案整包內 c code load 進來



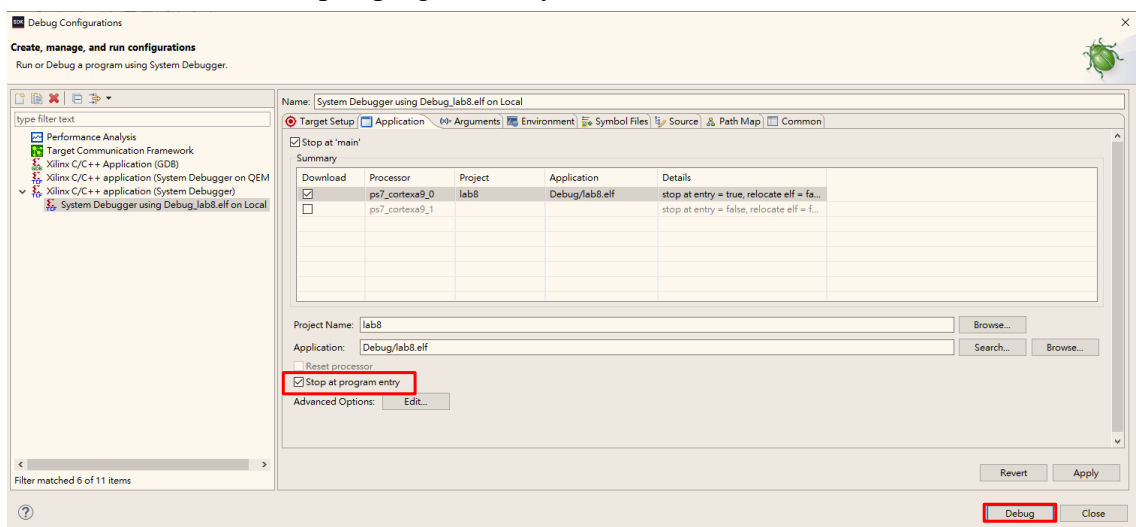
2.3 進行 Debug Mode

1. 對 Project Explorer 下的 lab8 右鍵點選 Debug As→Debug Configuration.



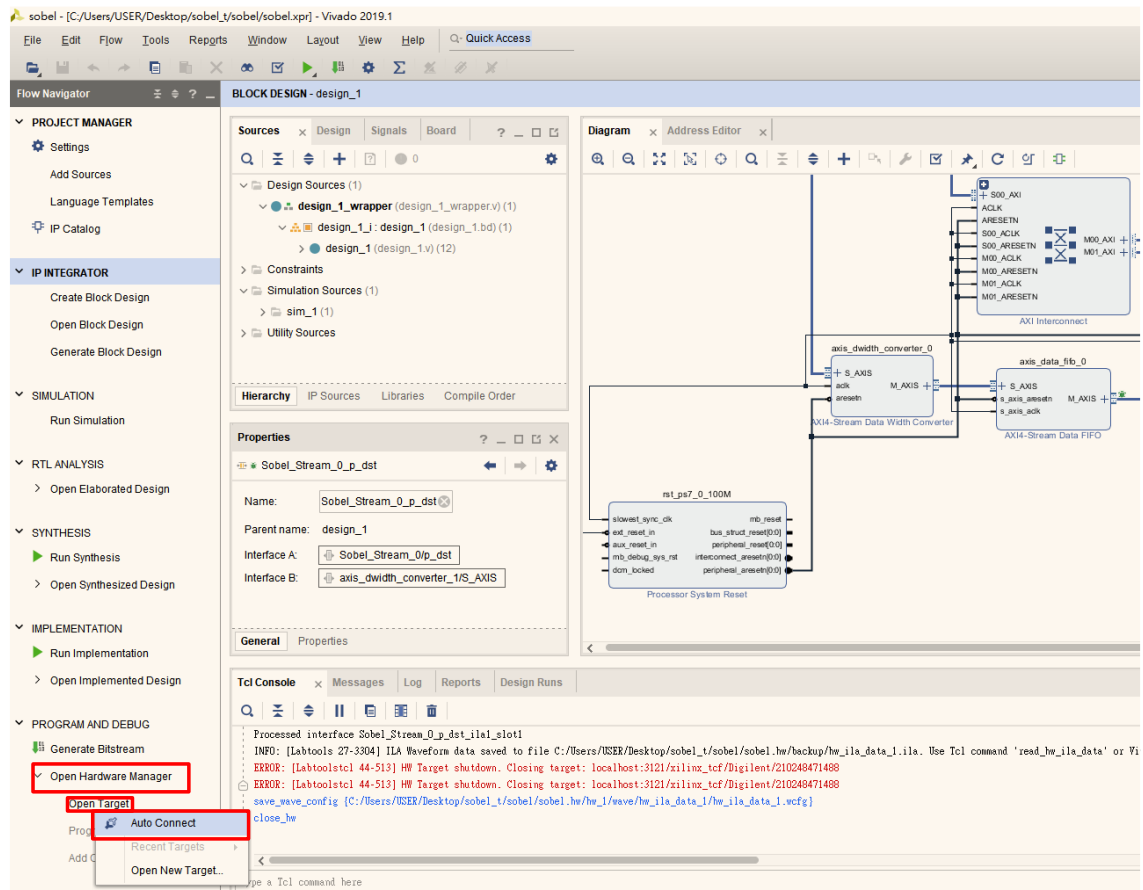
2. 於 Debug Configurations 窗格點選 System Debugger using Debug_lab8.elf on Local，接著以下步驟設定檔案絕對位置。

- 勾選下方選項 “Stop at program entry”



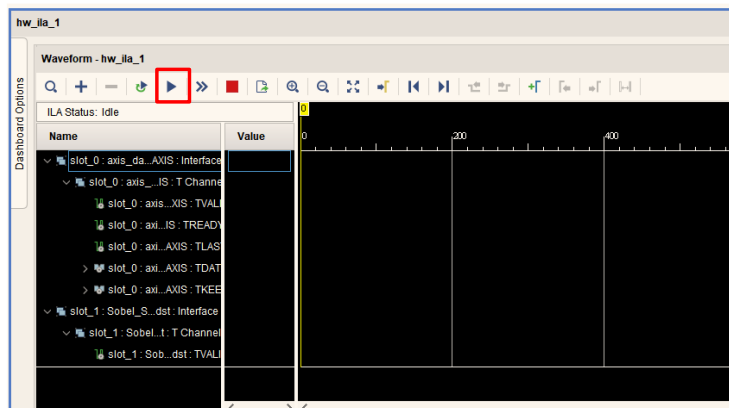
2.4 Vivado 端 Hardware Manager 設定

1. Flow Navigator 下的 Open Hardware Manager→Open Target→Auto Connect，打開 ILA 介面。



2.5 找尋訊號

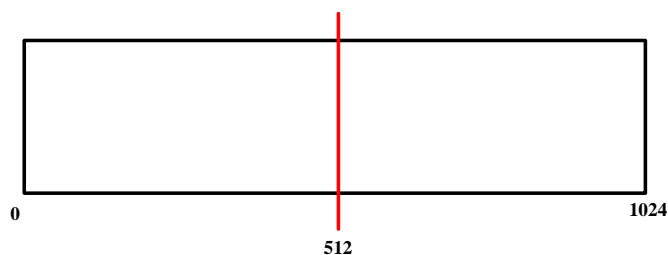
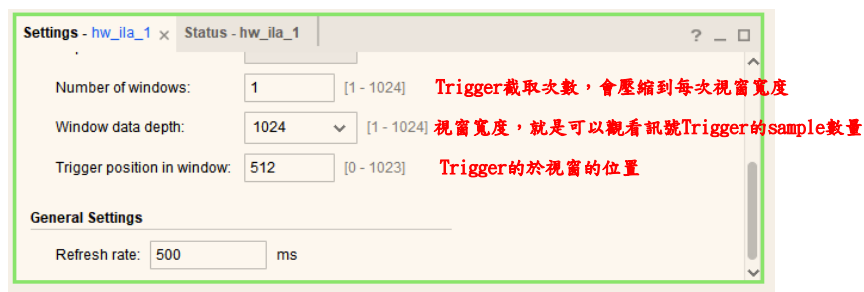
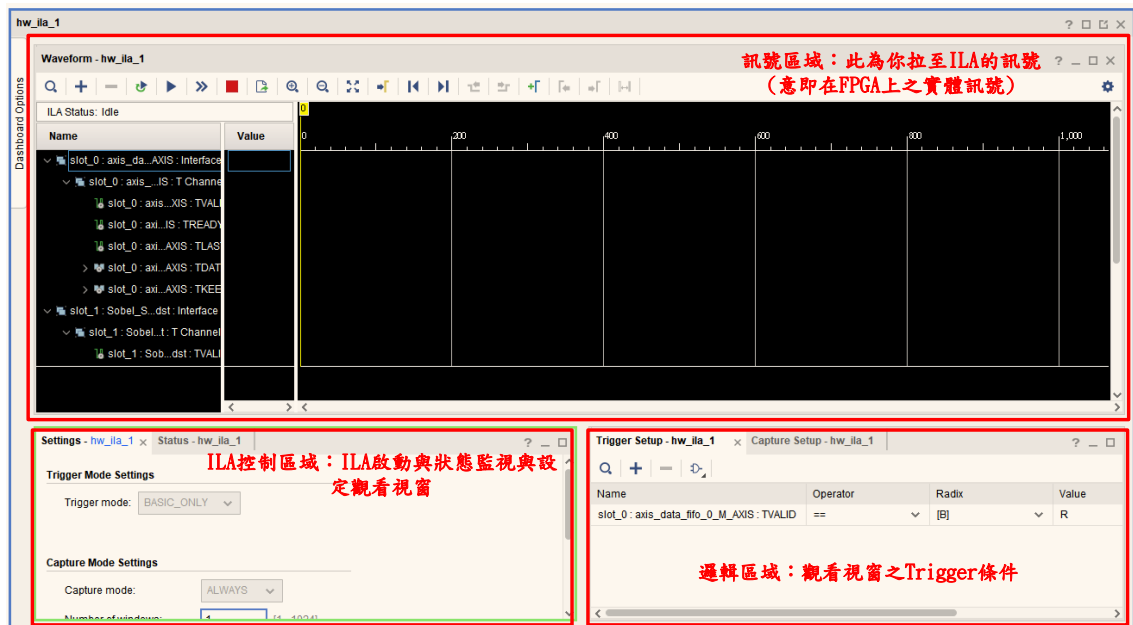
2.3 部分已經開啟 SDK Debug 介面，並且尚未開啟程式，當 ILA 的定義完成後，先行點選 ILA 上之播放鍵，接下來就可以在 Debug 介面執行程式，以讓 ILA 抓取訊號。



今日驗收：

找出 Sobel 輸出資料第一筆資料，並且 Trigger 須於第一個於 Window 的 Sample。

ILA 面板說明：



增加參考訊號

