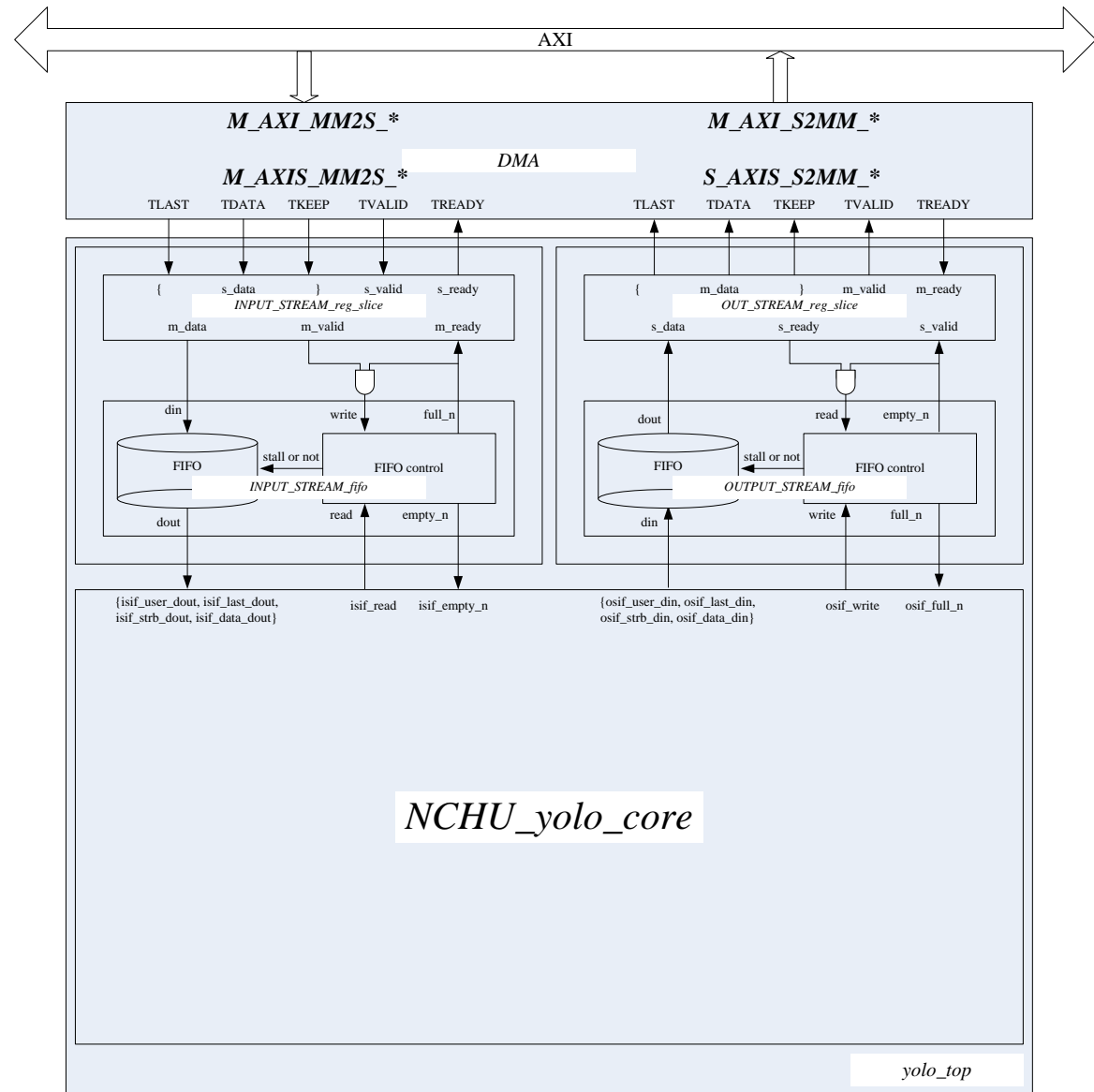


# SoC Lab6 AXI DMA 及自訂 IP 環境建立

Author: 蕭毓寬

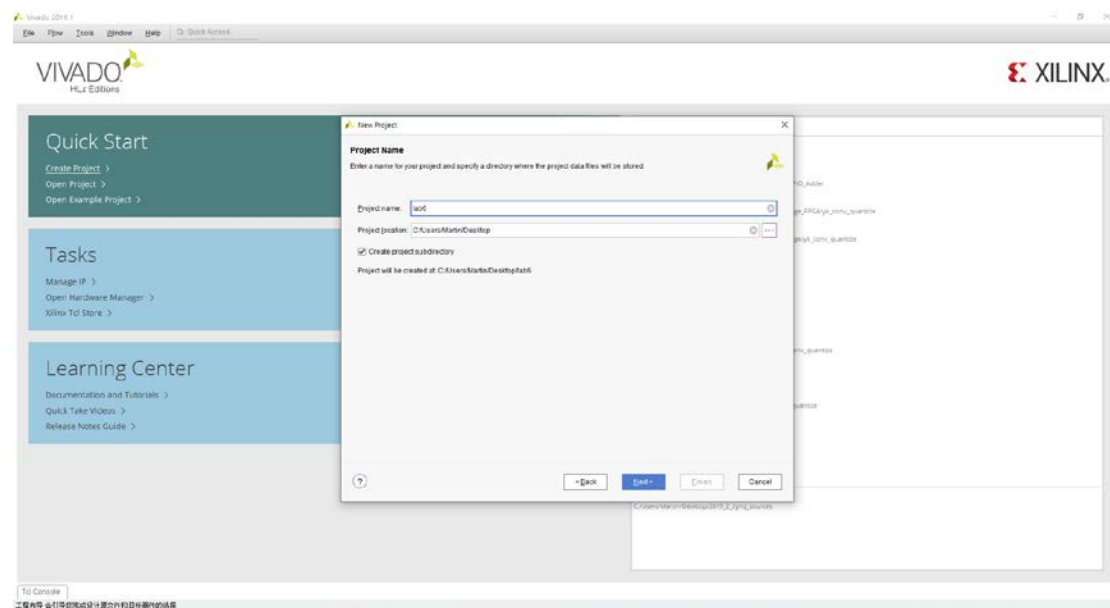
2021.05.11

本 Lab 將教學如何建立一個 PS 端經由 AXI DMA 與內含 FIFO 的 IP 間做資料的傳遞簡易環境。

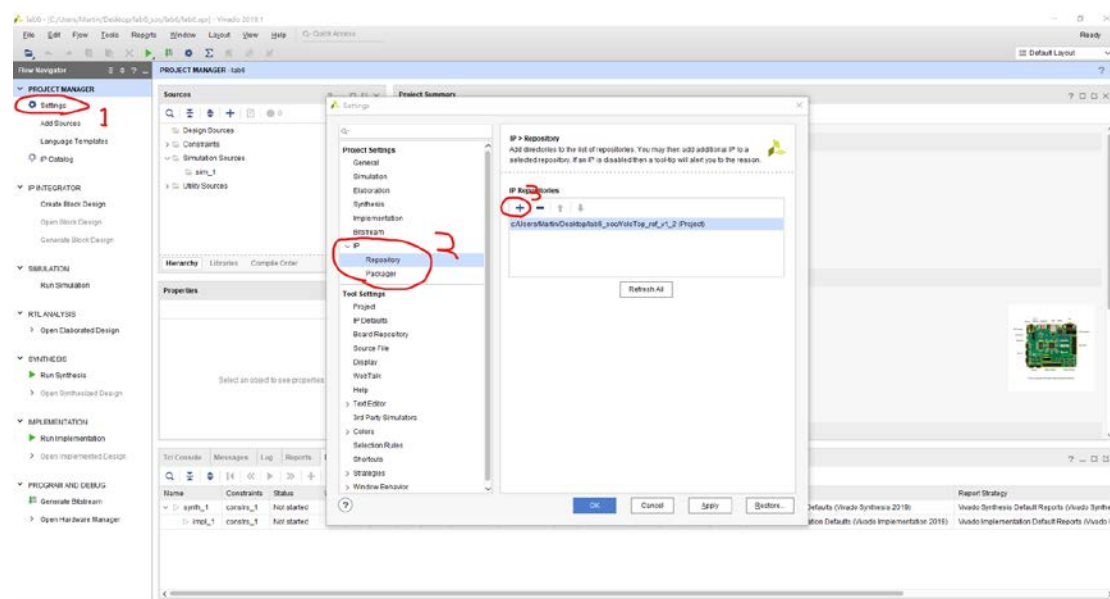


yolo\_top 為本 lab 作為範例的自訂 IP，其以包含 Input FIFO 以及 Output FIFO，方便同學們能簡易的與 AXI DMA 進行溝通。yolo\_top 與 DMA 透過 AXIS 協定溝通，期末 project 同學們可修改 yolo\_core 實踐想完成的 RTL 電路，yolo\_core 透過 isif\_empty\_n/isif\_read 交握訊號向 INPUT\_STREAM\_FIFO 拿資料，及 osif\_full\_n/osif\_write 交握訊號對 OUTPUT\_STREAM\_FIFO 寫出資料。Lab 所提供的 yolo\_core 功能僅 bypass 訊號。

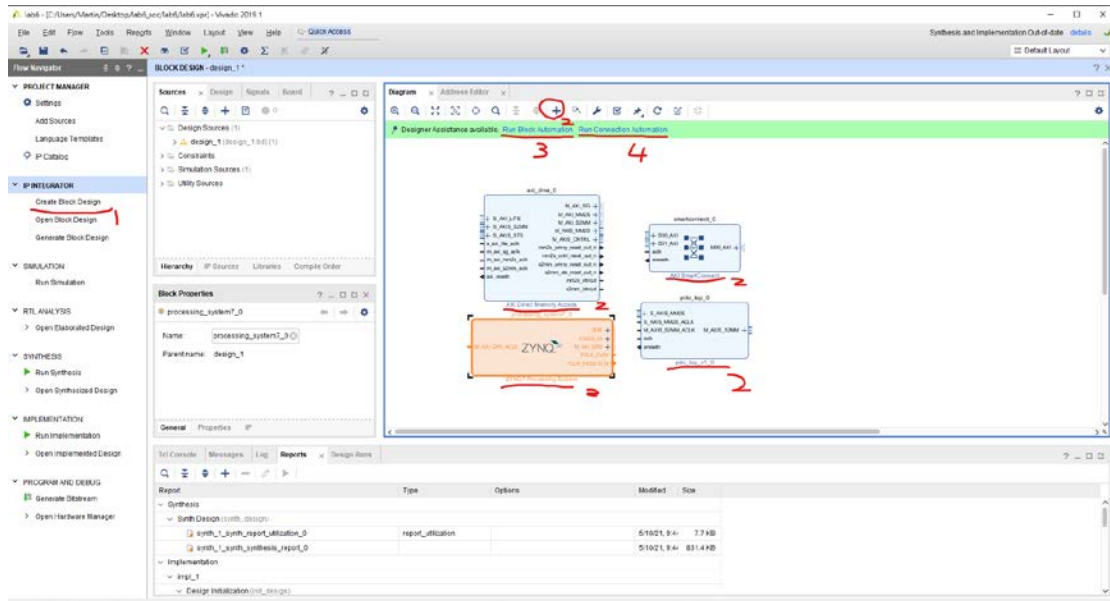
## 1. 建立 Vivado 專案



首先點擊 Creat Project 建立新的專案(lab6)

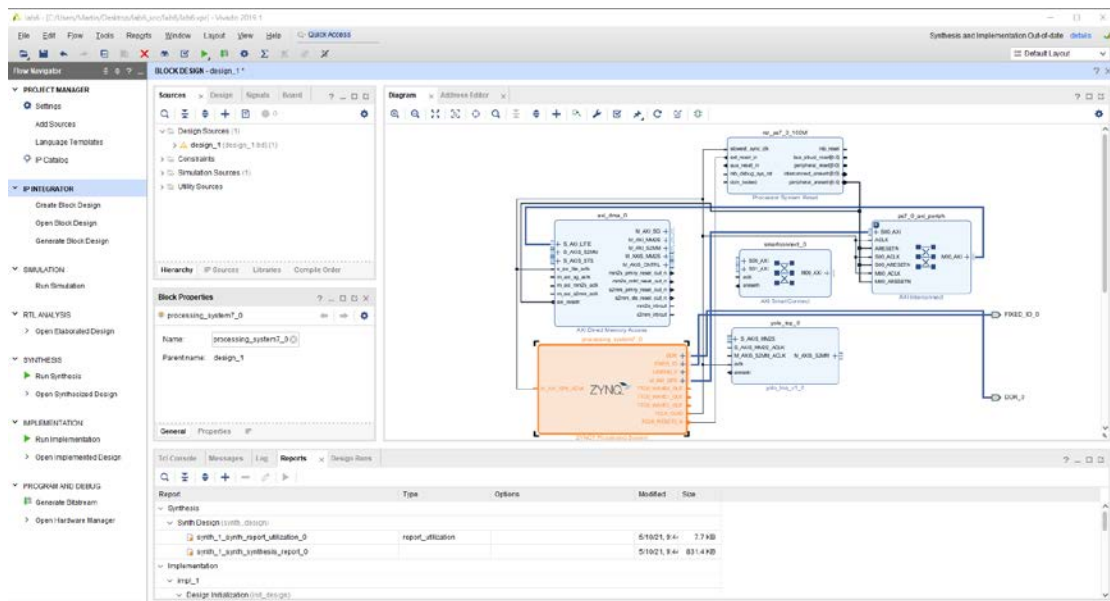


接著進入 Setting 內去引入 lab6 所提供的 YoloTop\_ref 資料夾以將本次實驗所要用的 yolo\_top IP 包進來使用。

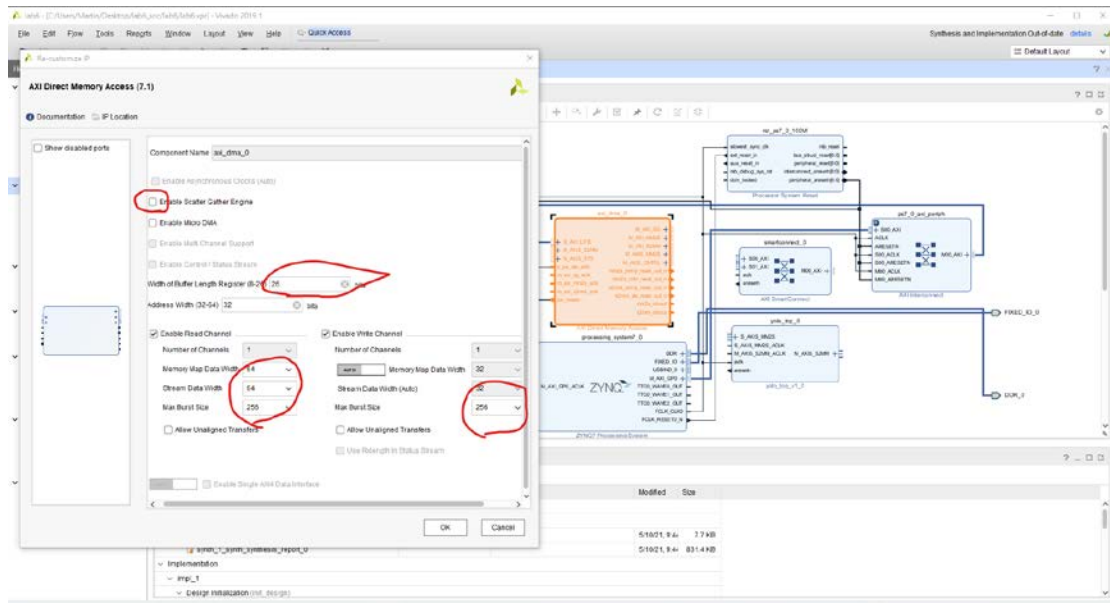


Create Block Design 以並加入 ZYNQ7 Processing System、AXI Direct Memory Access、AXI SmartConnect (以上三個為建立 PS 端及 DMA 用以跟自訂 IP 座溝通的相關 IP)，且加入 yolo\_top\_v1\_0 (此為自訂 IP)。

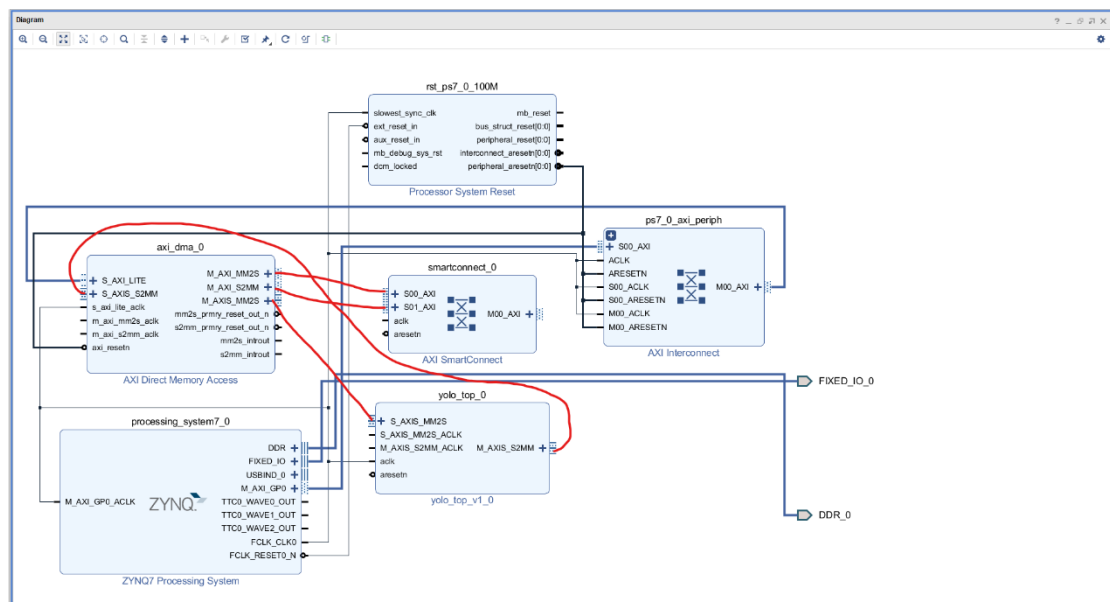
加入完成先點 Run Block Automation 讓 CAD tool 幫我們生成相對應需要生成的 IP，生成完畢再點 Run Connection Automation 讓 tool 幫我們把一些基礎線接上。



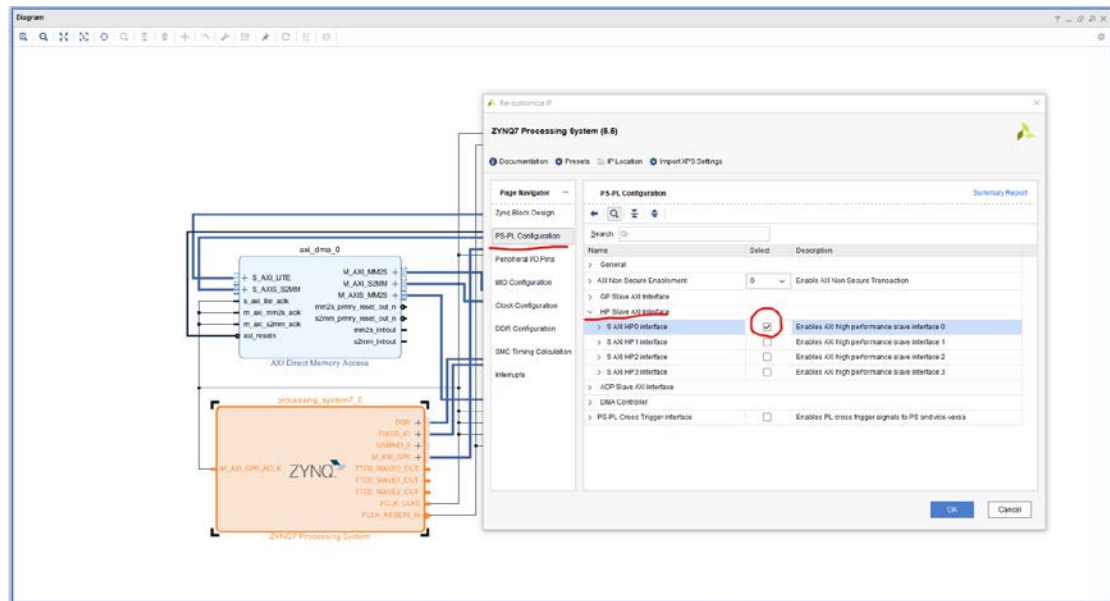
完成後應呈現如上圖所示。



接著我們需要設定 DMA 的參數用以跟 yolo\_top IP 接口匹配，取消勾選 Enable Scatter Gather Engine 並將參數設定成如上圖所示。

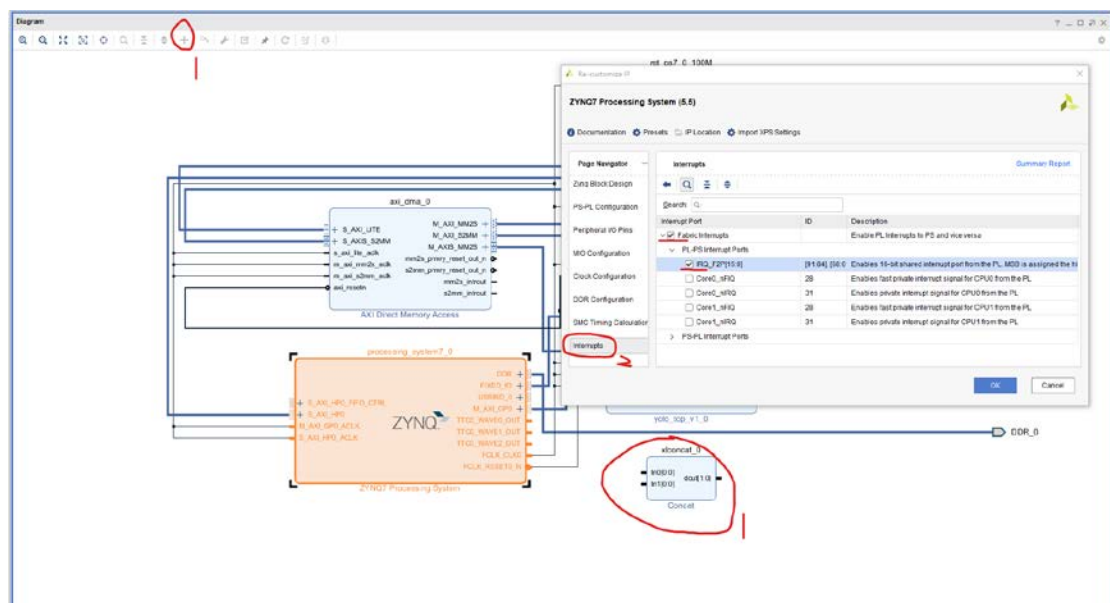


將 yolo\_top 的輸入(S\_AXIS\_MM2S)與輸出(M\_AXIS\_S2MM)與 DMA 如圖的對應接口連接，並將 DMA 的 M\_AXIS\_MM2S、M\_AXIS\_S2MM 與 AXISmart Connect 接上，然後記得要把 yolo\_top 的 aresetn 接上 PS 端的 FCLK\_RESET0\_N，如此便完成了 yolo\_top 與 DMA 溝通的接腳設定。

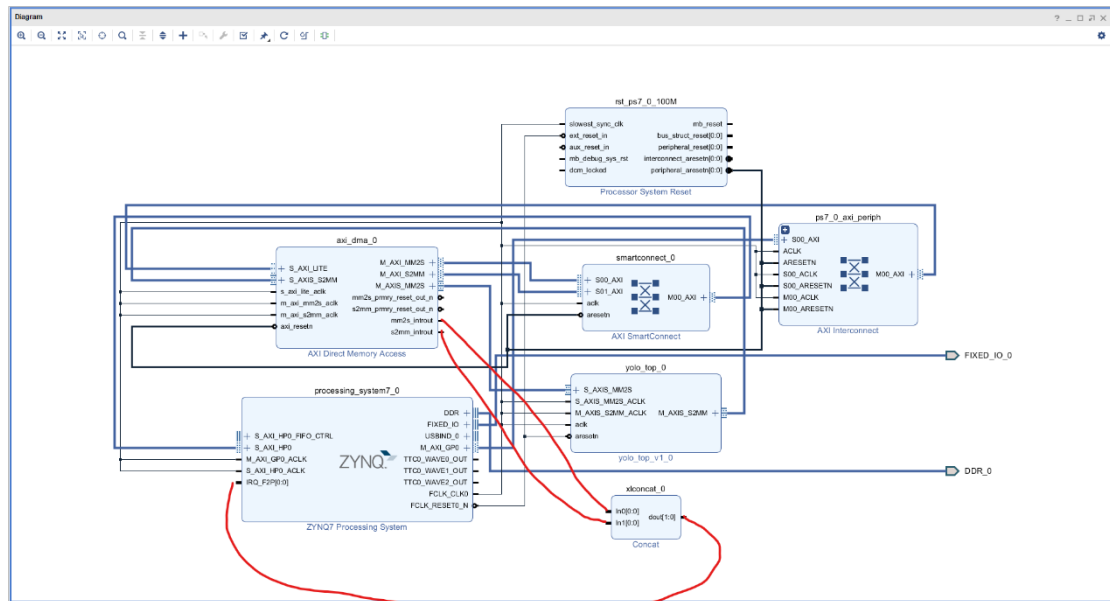


再來換 PS 端的設定，點開 ZYNQ7 Process System，在 PS-PL Configuration 中把 HP Slave AXI Interface 中的 HP0 勾選。按完 OK 後上面會再出現一個 Run Connection Automation，點下去之後讓他自動把 SmartConnect 與 PS 的接線接好。

以上是大致 PS、DMA 與自訂 IP 的建立，接著要再多加一點小東西

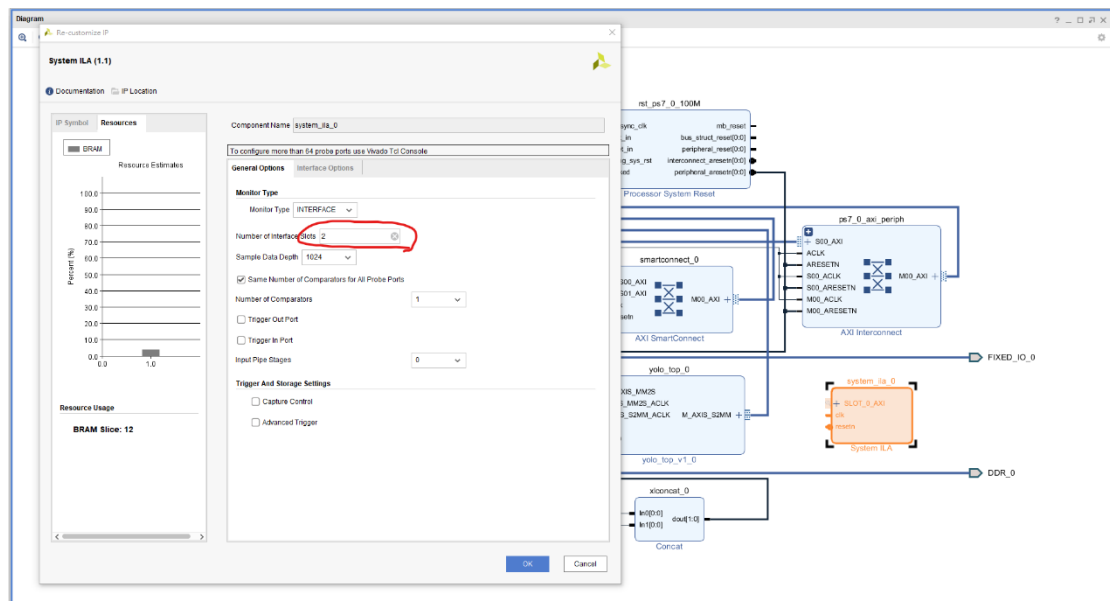


一樣去 Add IP 的地方叫出一個 Concat，並且在點開 PS 端，將裡面 Interrupts 的 IRQ\_F2P 角打開，我們用它來跟 PS 端去做 DMA TX 及 RX 的訊號接收。



照著上圖把它接起來~~

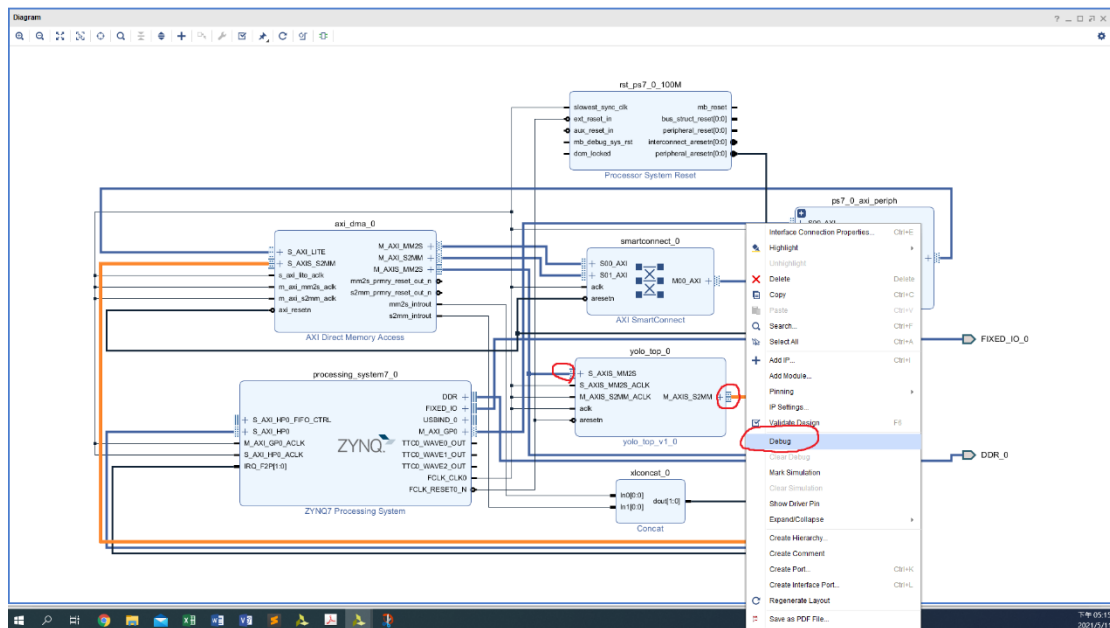
這樣的設定完畢之後其實已經可以在板子上運行了，然而各位在寫自己 IP 的時候一定會需要檢查 wave 運作方式是否與自己想要的相同，所以我們引進一個 System ILA 的 IP 讓我們可以通過 Vivado 觀察板子上運作時的波型。



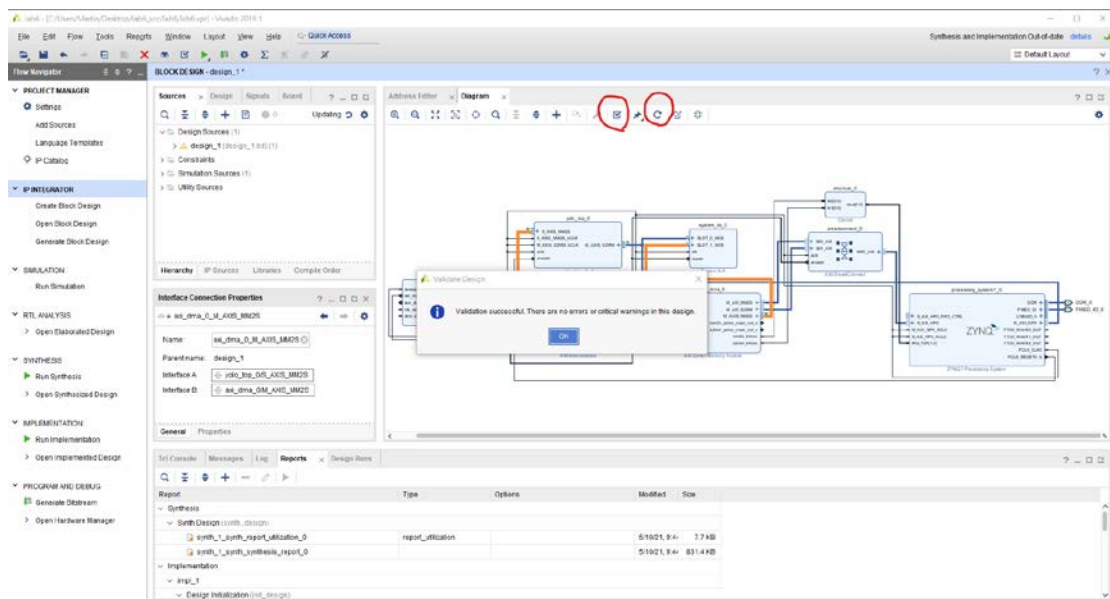
在 Add IP 引入 System ILA，然後 Number of Interface Slots 設定成 2 個。





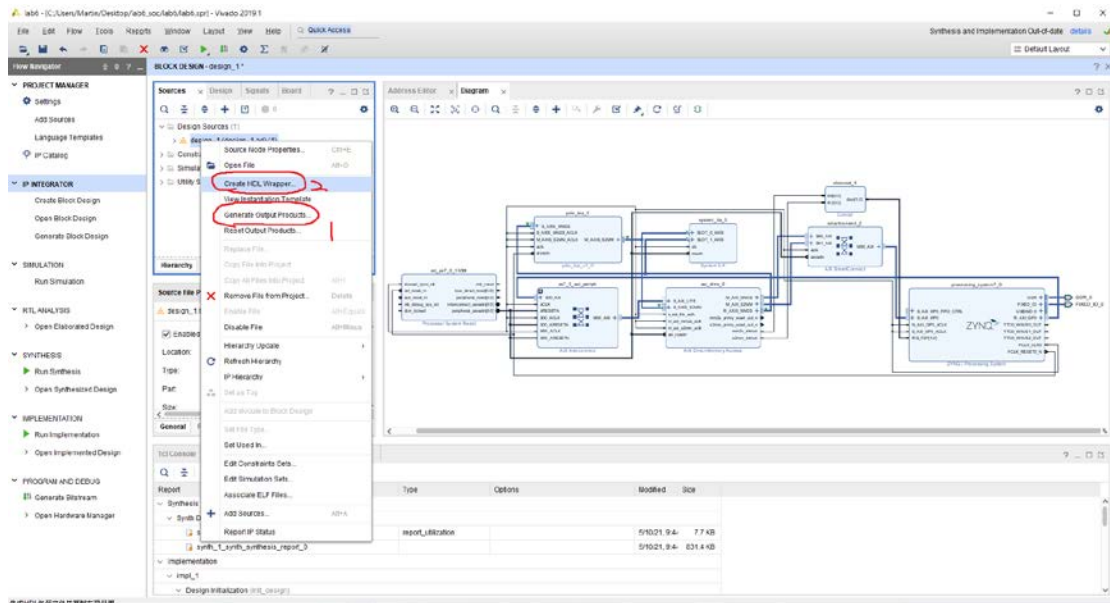


要能透過 Vivado 看波型我們還需要在 S2MM 及 MM2S 兩條線上點右鍵，選擇 Debug，然後就會看到有隻小甲蟲在接腳上。



最後可以按上面的 ReGenerate Layout 讓它排列整齊一點，然後按下 Validate Design 檢查有沒有問題，沒問題會跳出上面的 successful。



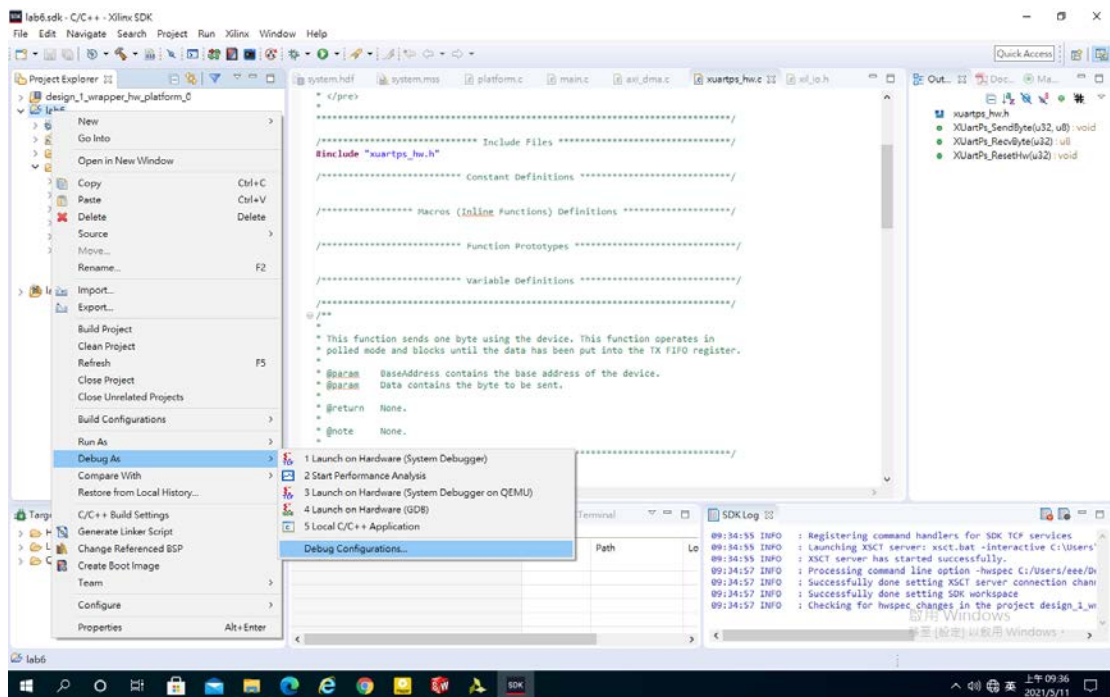


像之前 Lab 一樣，先 Generate Output Products 後點 Create HDL Wrapper 讓它生成一份 RLT 的外包。再來就點下 Generate Bitstream 並祈禱自己一切順利吧哈哈~(有問題要注意前面步驟是不是漏了甚麼)

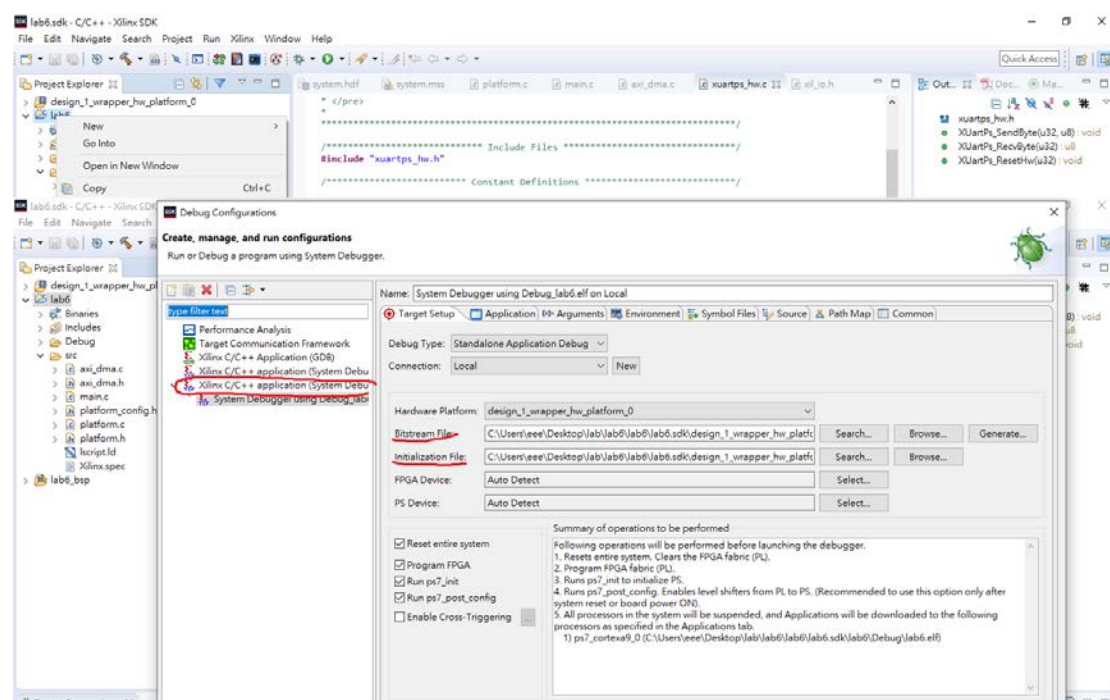
## 2. 以 SDK 做 SoC 的操作

請先跟之前的 Lab 一樣建立個新的 C code Project 以及 bsp，並將 lab6 提供的 src 中的 C code import 進專案中，接著右鍵對 bsp 按 Re-Built 和 C code 專案按下 Build Project。

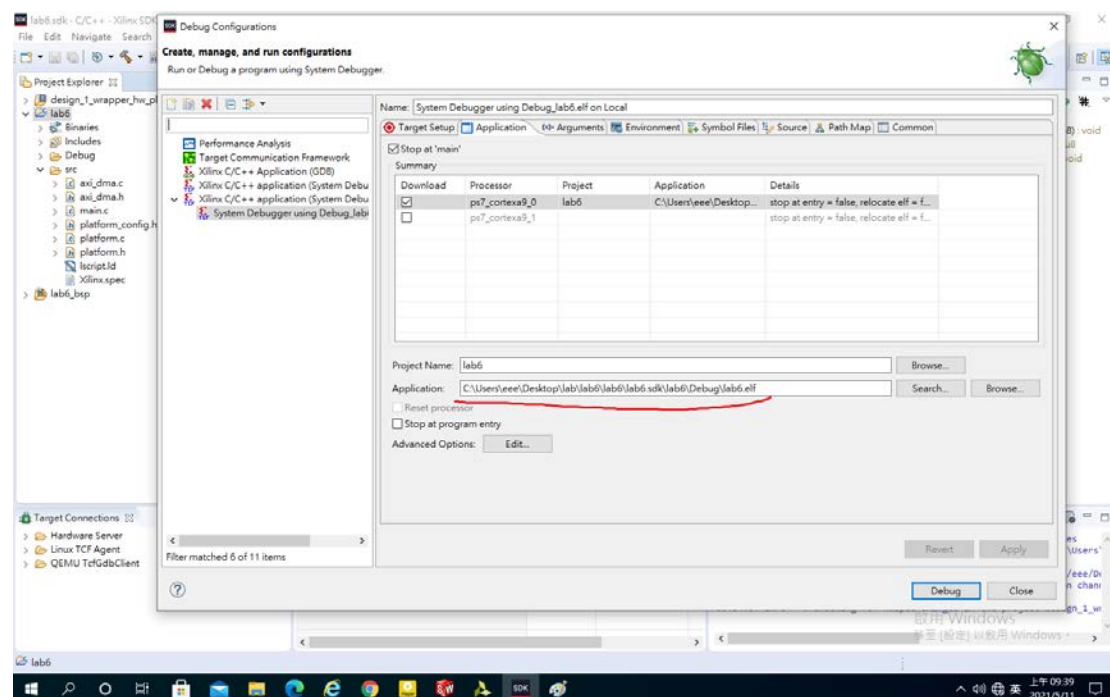
這次操作板子的方式跟之前不同，因為要讓各位順勢學會怎麼 debug。



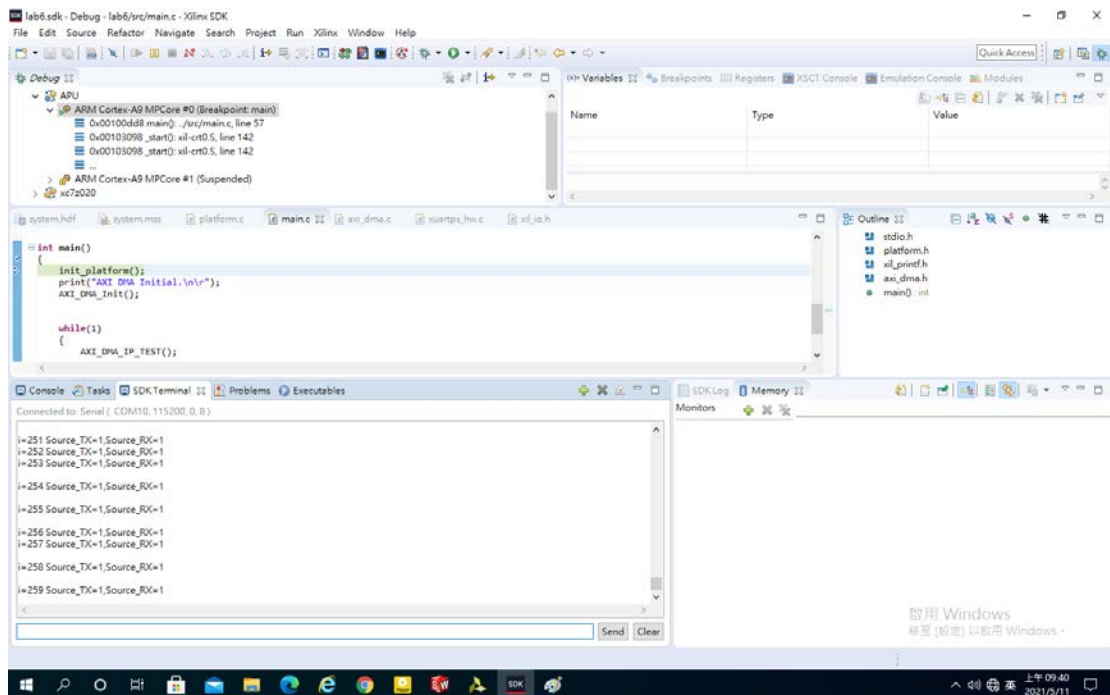
右鍵點選專案，選擇 Debug As 中的 Debug Configuration。



左側選擇 System Debugger，確認紅線處的檔案的位置(用 Browse 去確定絕對路徑)，該打勾的也確定一下

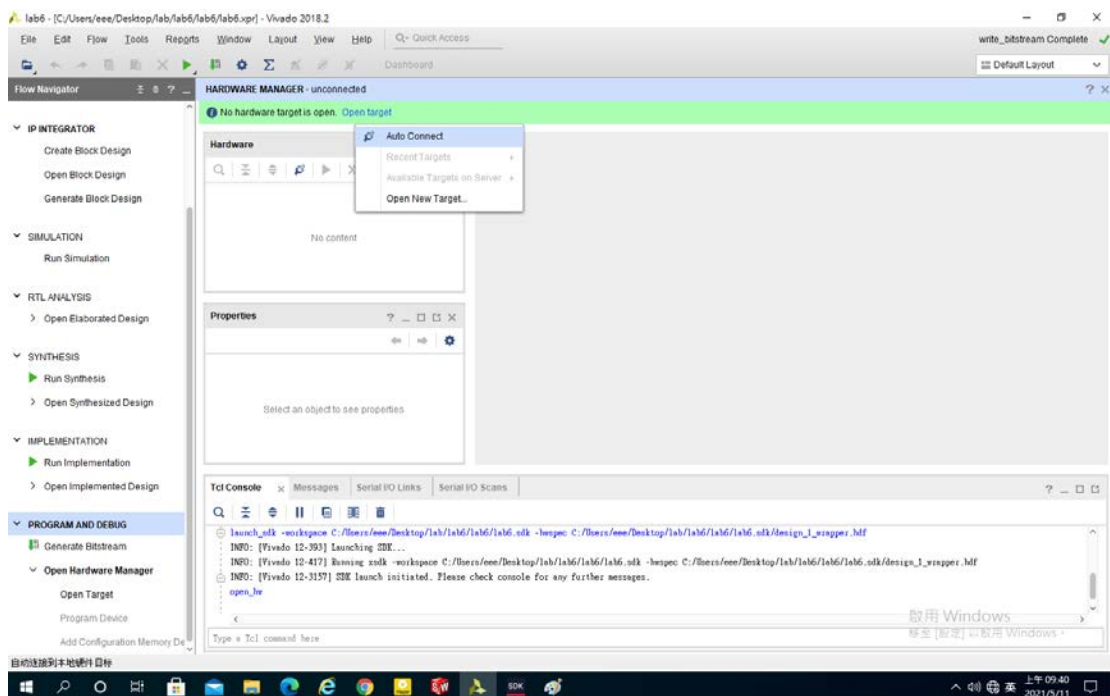


Application 處也確定紅線處的路徑並確認 ps7\_cortexa9\_0 是打勾的，就能按下 Debug 了。

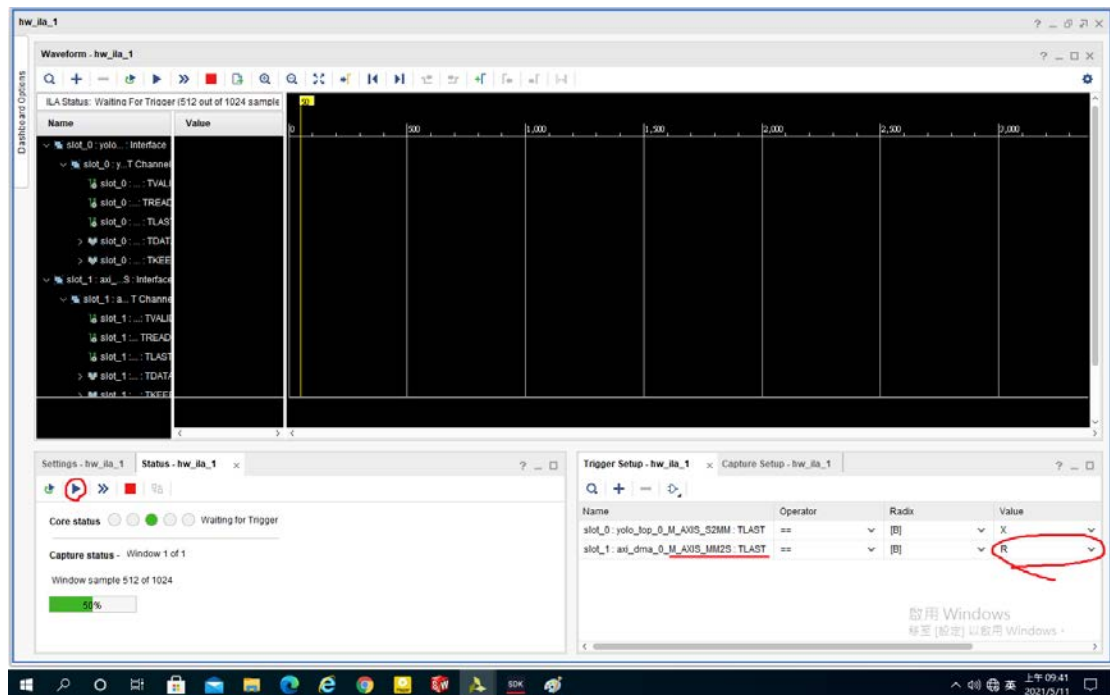


這邊就是 Debug 的畫面，可以像是一般軟體開發一樣設定程式的中斷點，各位可以自己玩玩看。不設定中斷點按下上方的綠色播放鍵就可以看 Terminal 跑出畫面了(Terminal 也跟之前 lab 一樣記得接 COM)。

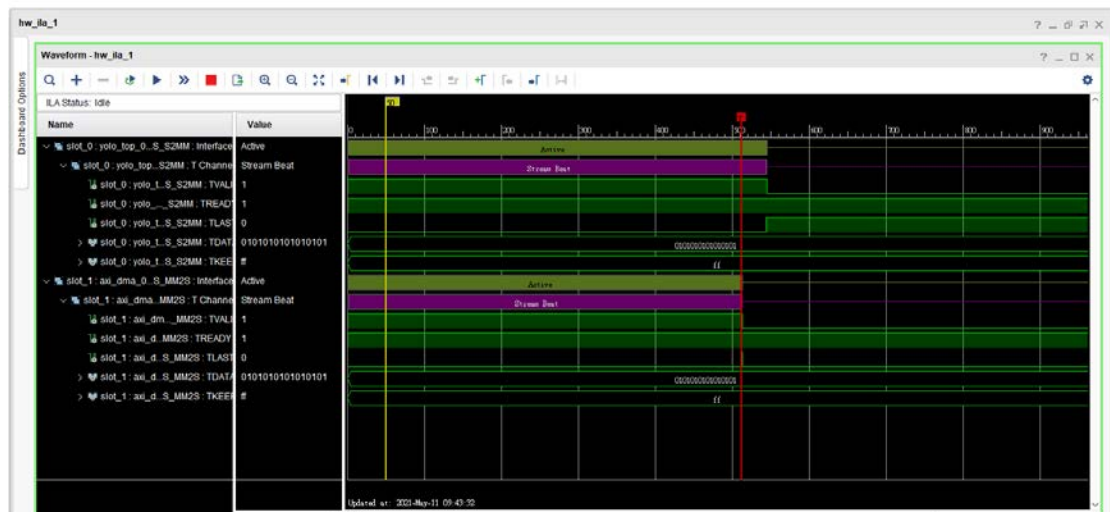
同時間的可以用 Vivado 監看波型，ILA 的用途就是在這，請大家再重新燒操作一次 Debug AS...，但這次先不要按下綠色播放鍵。



在 Vivado 左邊最下面 Open Hardware Manager 然後 Auto Connect



右下 Trigger Setup 的部分把 MM2S 的 TLAST 加進來，Value 設定為 R(0 升 1)當作 Trigger，然後按下左邊的播放鍵。這時再重新去執行剛剛 SDK 那邊的程式執行。



然後就會看到這邊能跑出波型啦~