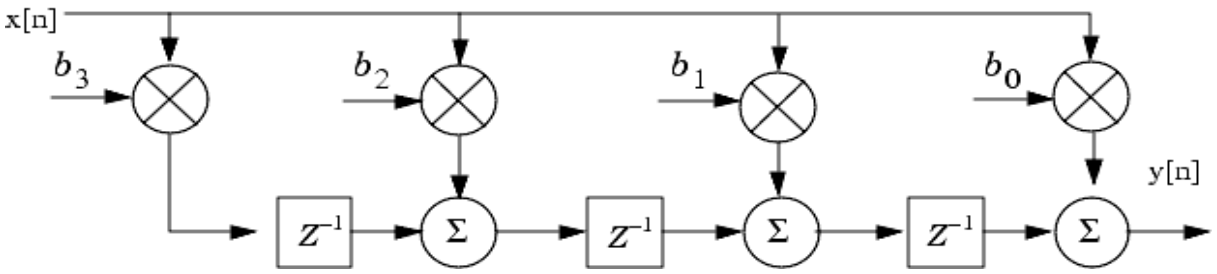
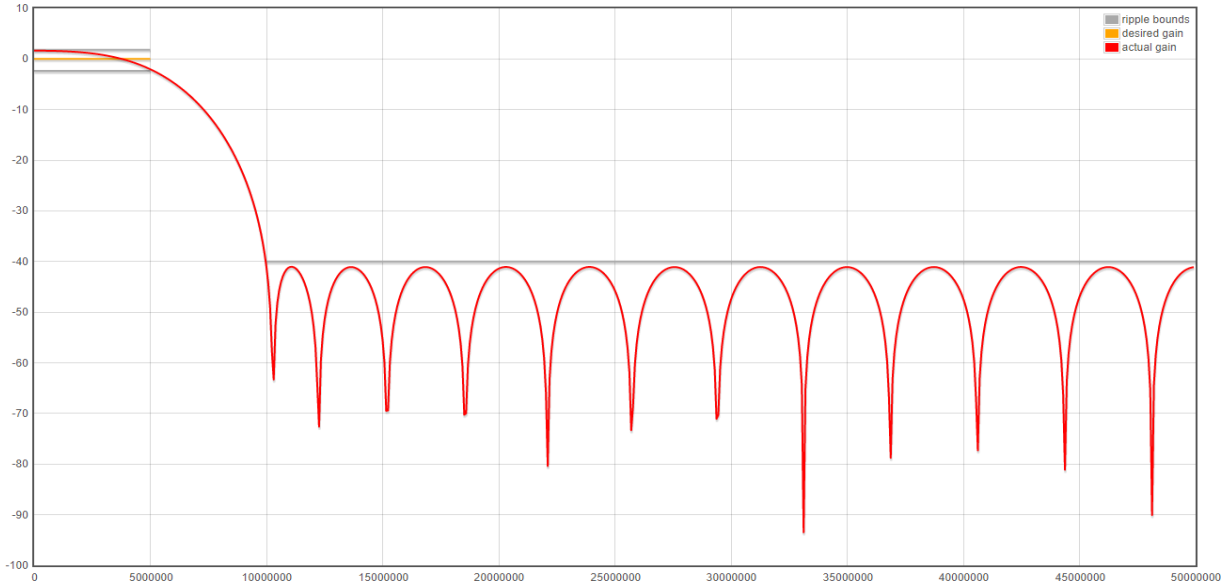
Lab09 FPGA
Low-pass FIR filter

Low-pass FIR filter

- FIR filter Block Diagram
- 數位低通濾波器
- 相較於類比低通濾波器來說製作難易度低、成本低。
- 硬體上只需要暫存器、乘法器和加法器即可實現濾波效果。

$$y[n] = b_0x[n] + b_1x[n-1] + b_2x[n-2] + b_3x[n-3]$$





FPGA

- 利用軟體運算FIR filter

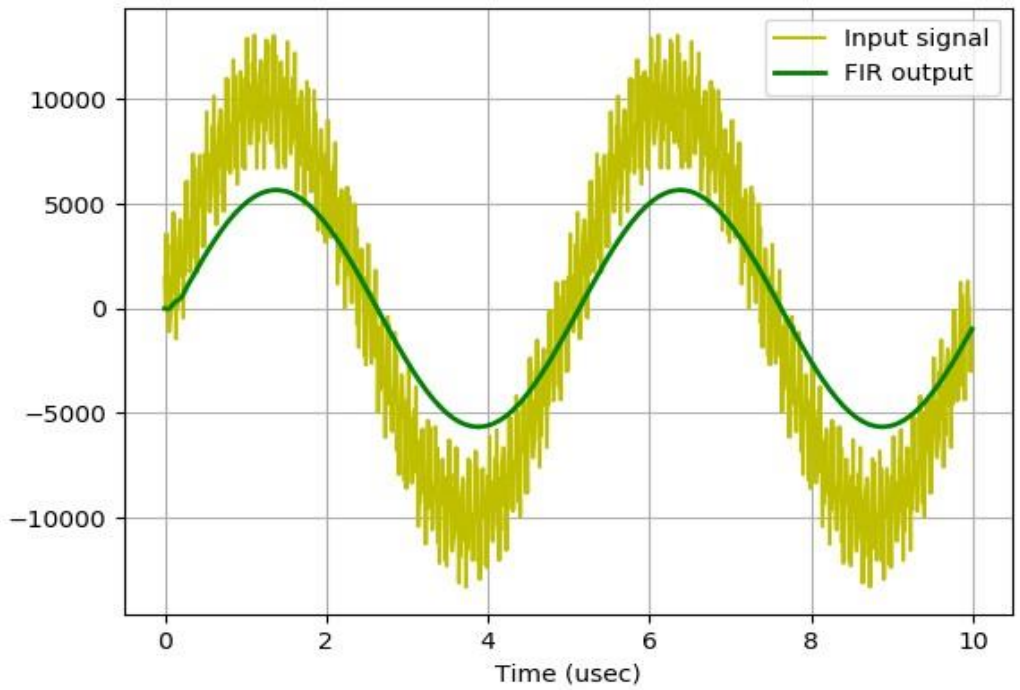
```
In [3]: from scipy.signal import lfilter

coeffs = [-255,-260,-312,-288,-144,153,616,1233,1963,2739,3474,4081,4481,4620,4481,4081,3474,2739,1963,1233,616,153,-144,-288,-31

import time
start_time = time.time()
sw_fir_output = lfilter(coeffs,70e3,samples)
stop_time = time.time()
sw_exec_time = stop_time - start_time
print('Software FIR execution time: ',sw_exec_time)

# Plot the result to notebook
plot_to_notebook(t,samples,1000,out_signal=sw_fir_output)

Software FIR execution time:  0.08370518684387207
```



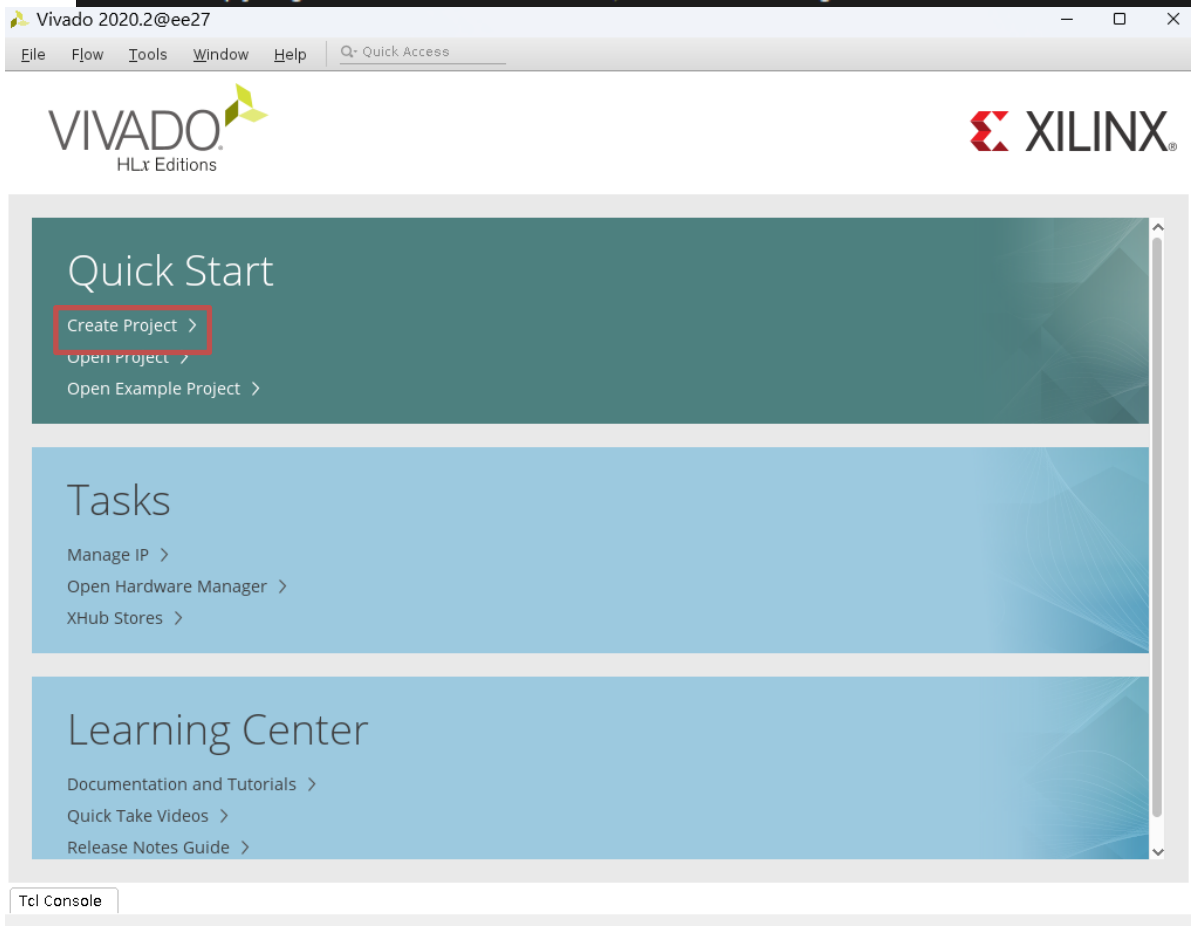
Block Design Flow

開啟vivado

- 在terminal輸入指令 vivado &

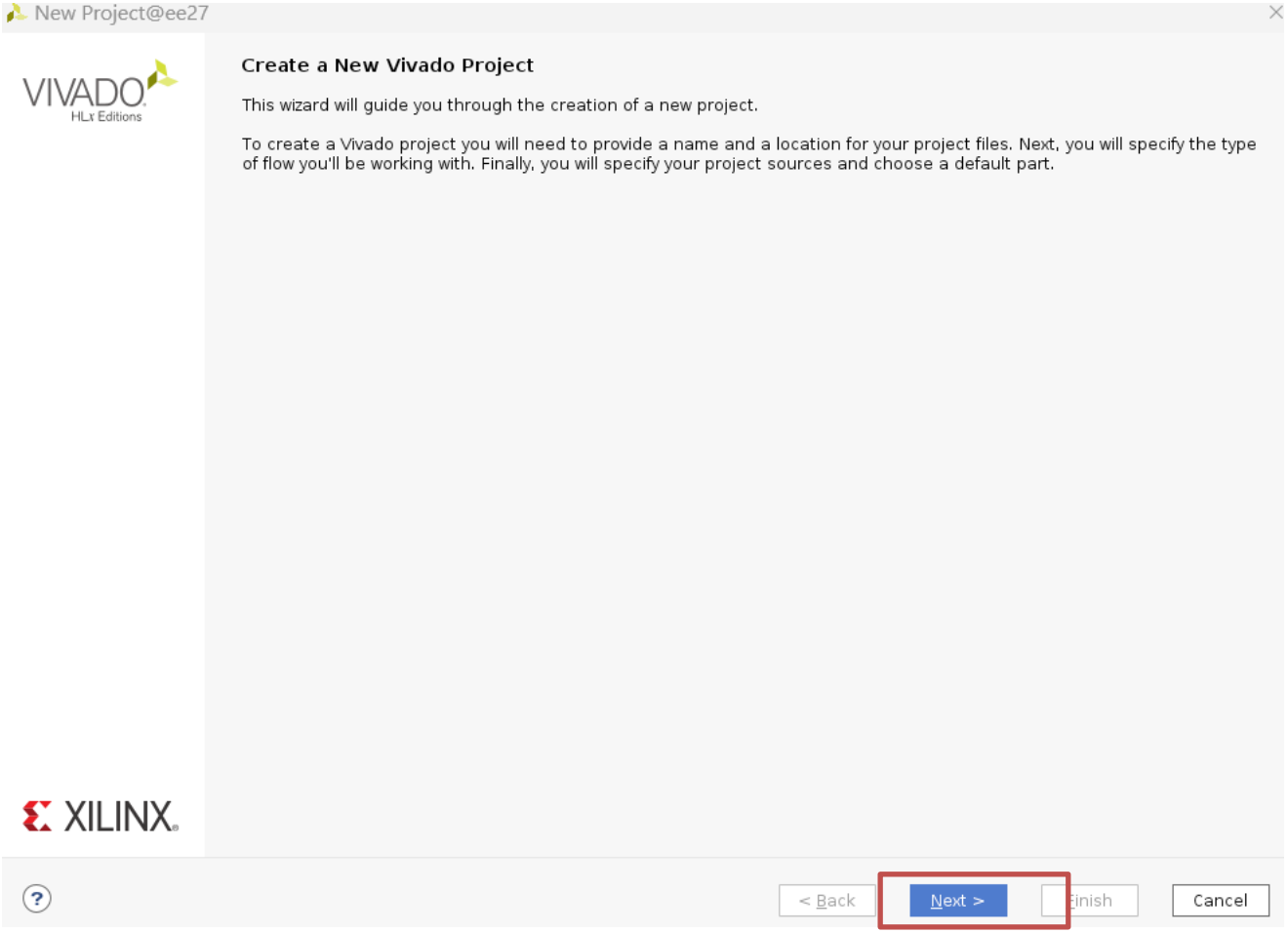
```
19:44 dcs175@ee27[~]$ vivado &
[1] 37161
19:44 dcs175@ee27[~]$
****** Vivado v2020.2 (64-bit)
  **** SW Build 3064766 on Wed Nov 18 09:12:47 MST 2020
  **** IP Build 3064653 on Wed Nov 18 14:17:31 MST 2020
    ** Copyright 1986-2020 Xilinx, Inc. All Rights Reserved.
```

- 點選create project



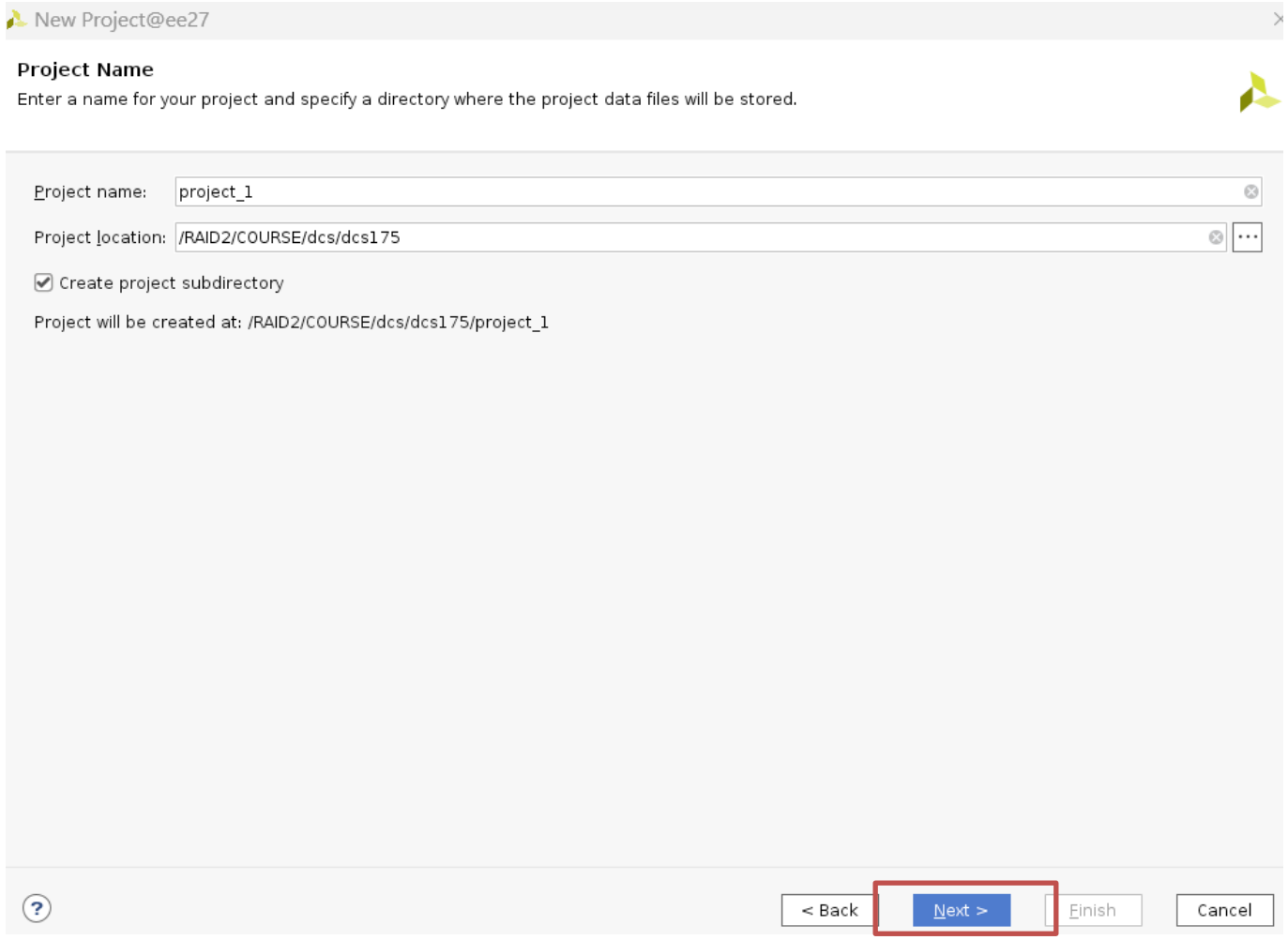
開啟vivado

- 點選next



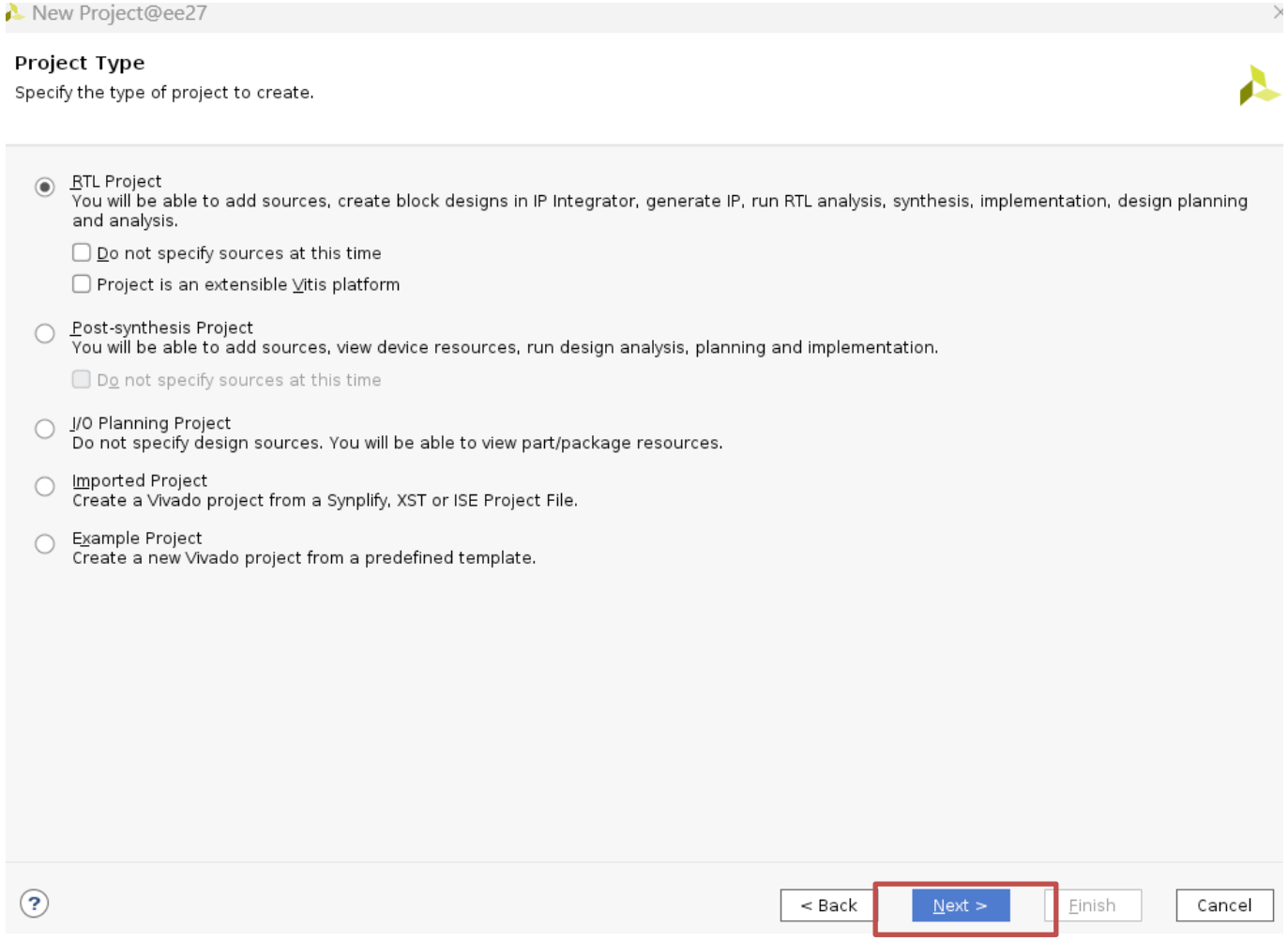
開啟vivado

- 點選next



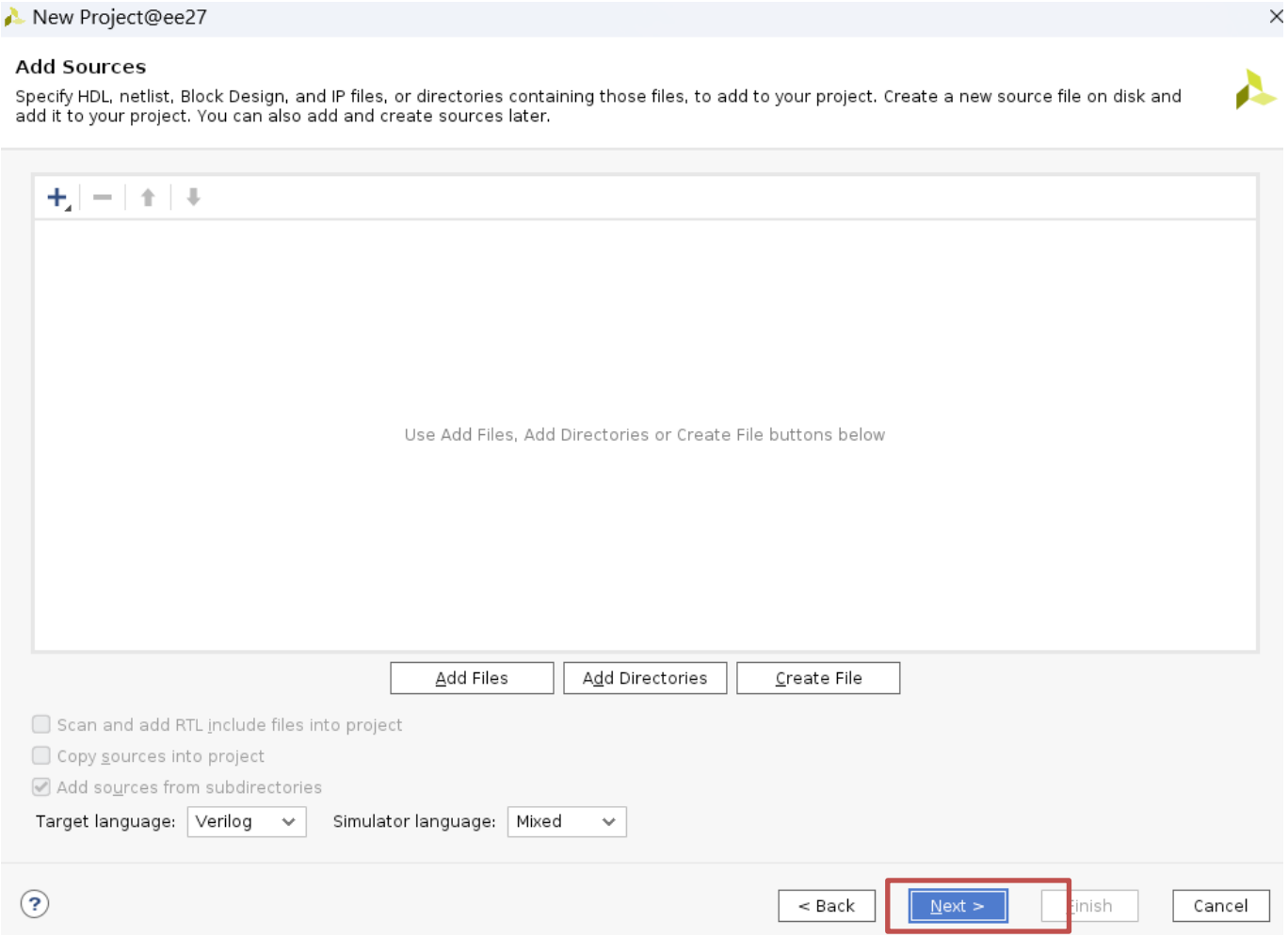
開啟vivado

- 點選next



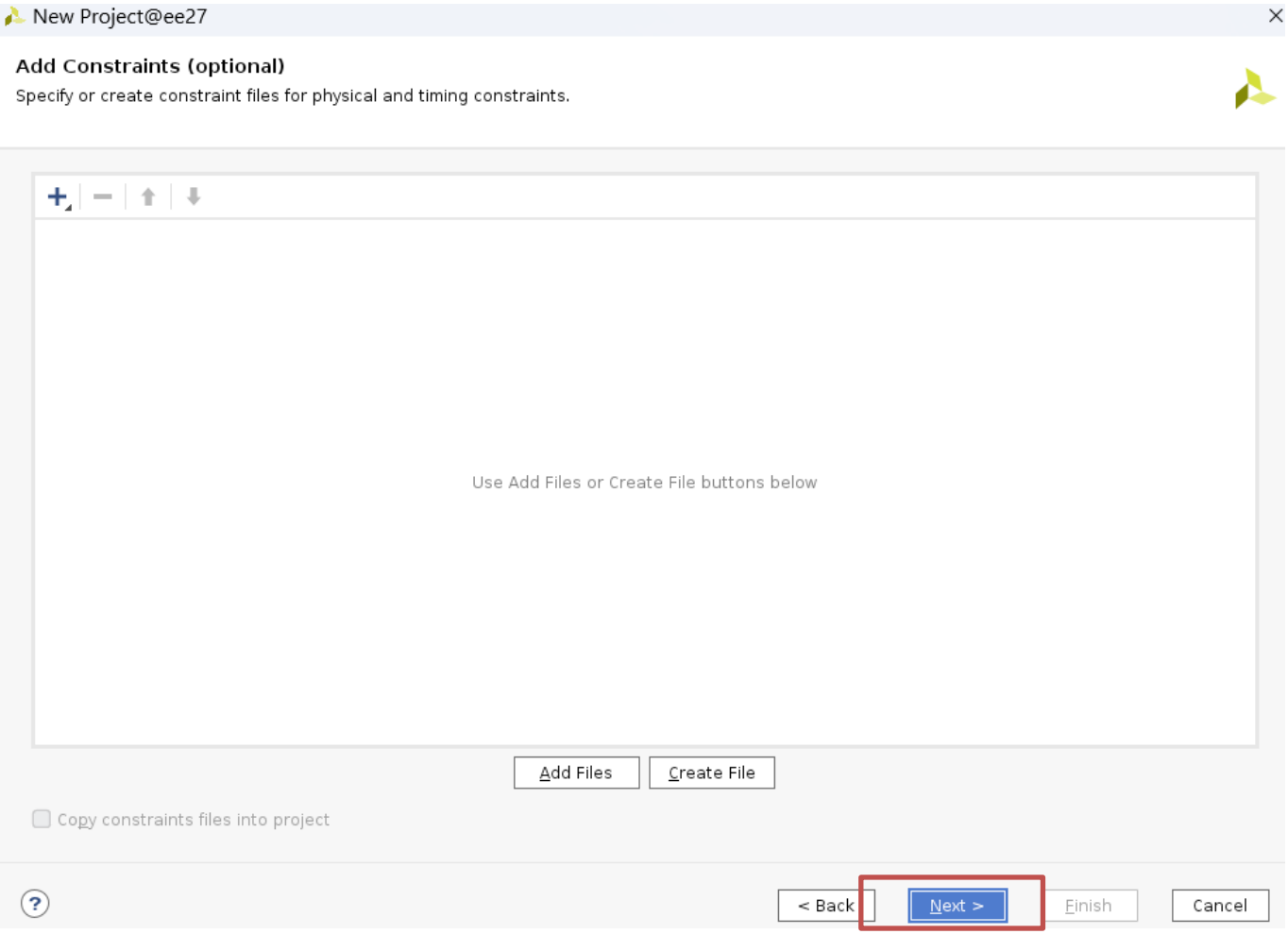
開啟vivado

- 點選next



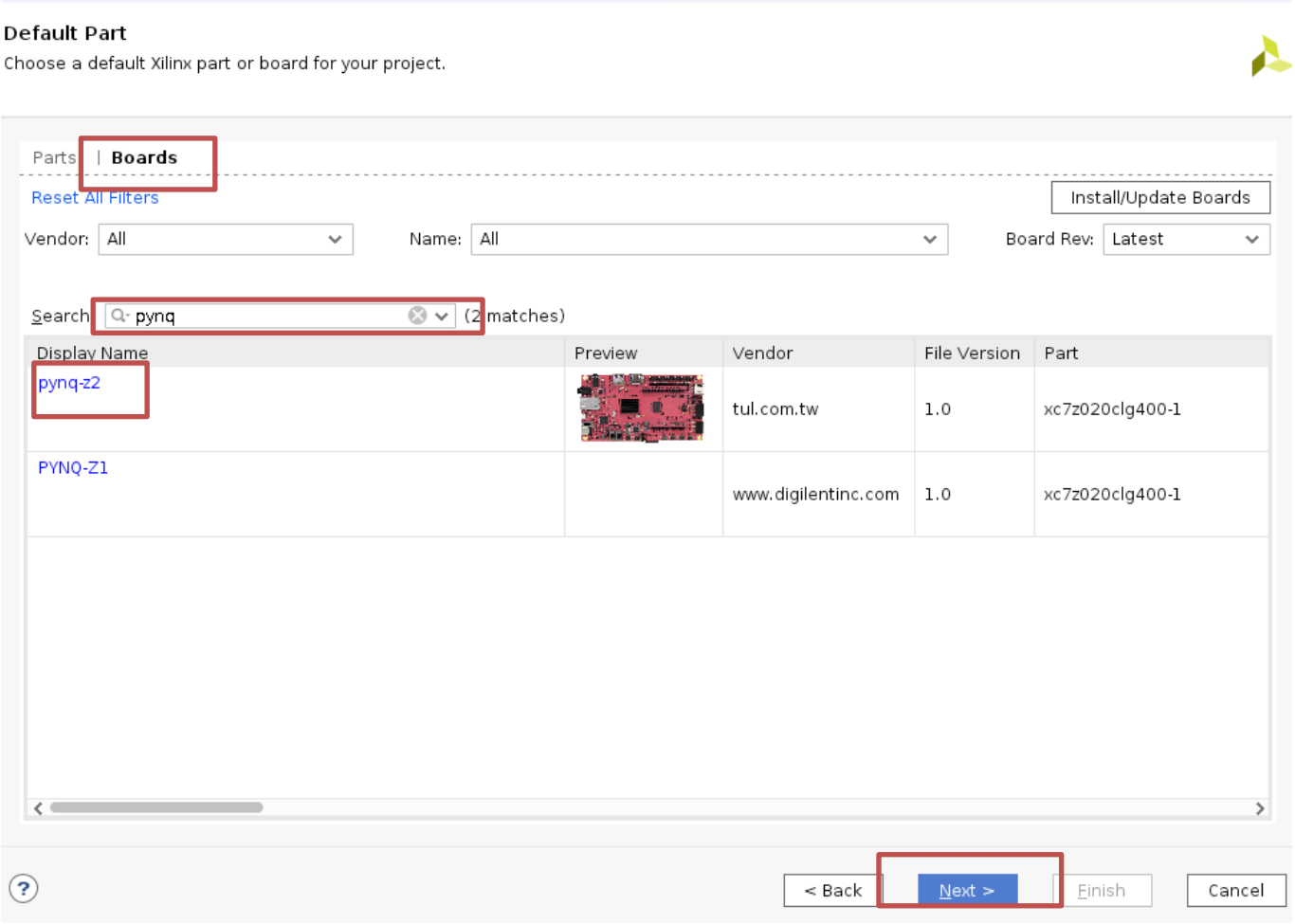
開啟vivado

- 點選next



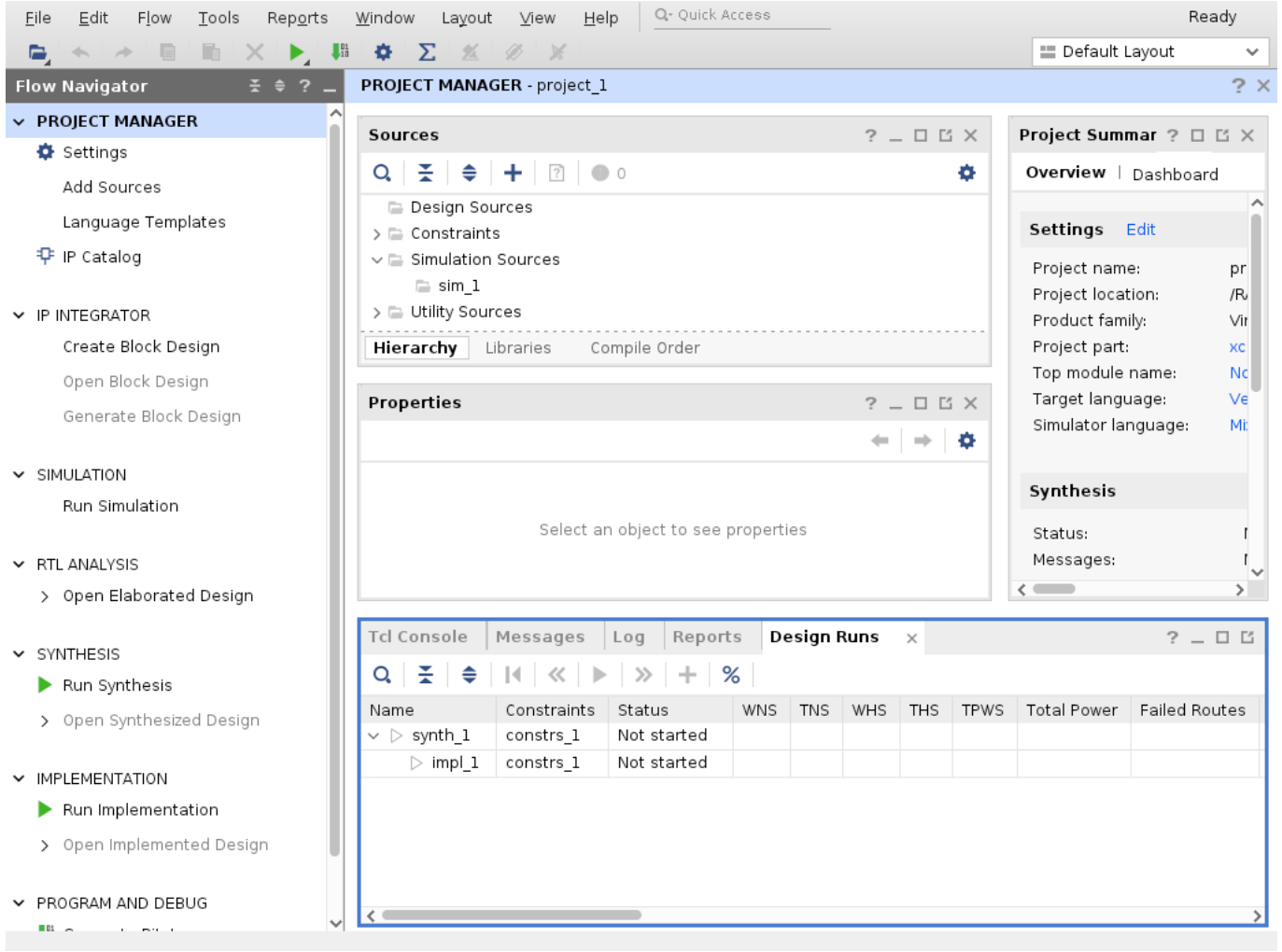
開啟vivado

- 點選boards
- 在search 搜尋pynq
- 點選pynq-z2
- 點選next

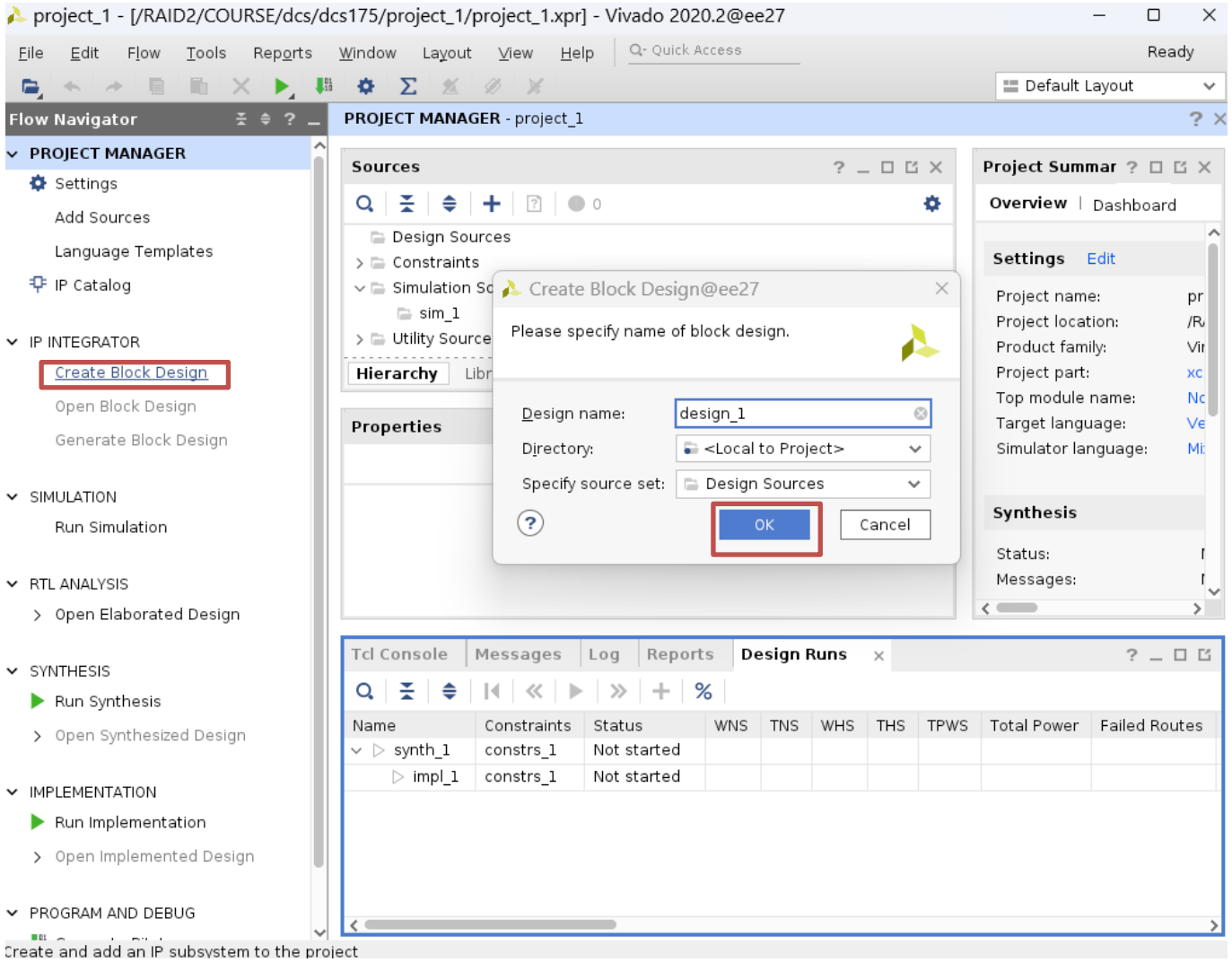


開啟vivado

- 點選finish後進入vivado

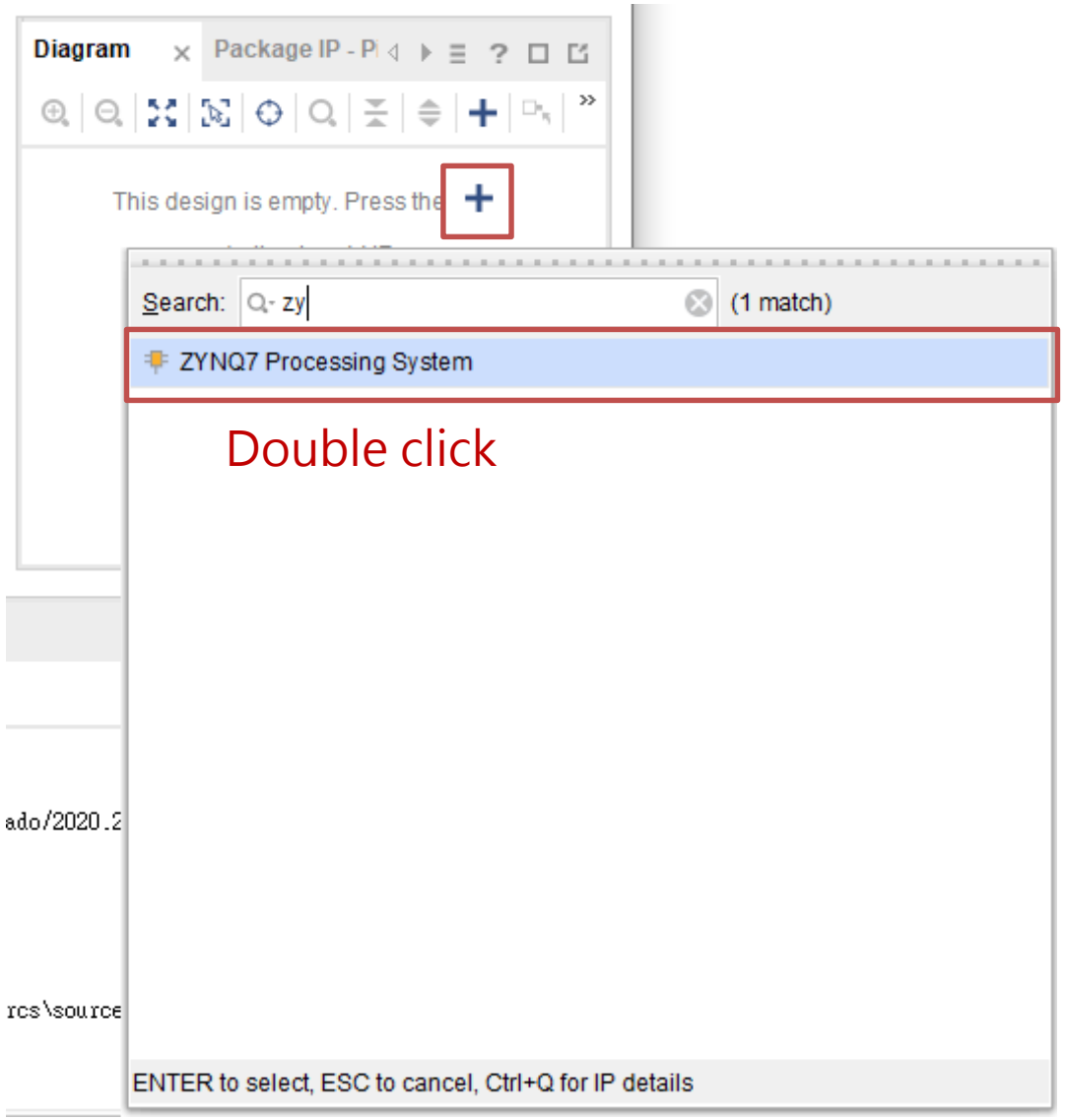


創建overlay


project_1 - [/RAID2/COURSE/dcs/dcs175/project_1/project_1.xpr] - Vivado 2020.2@ee27
File Edit Flow Tools Reports Window Layout View Help
Quick Access
Ready
Default Layout
Flow Navigator
PROJECT MANAGER
Settings
Add Sources
Language Templates
IP Catalog
IP INTEGRATOR
Create Block Design
Open Block Design
Generate Block Design
SIMULATION
Run Simulation
RTL ANALYSIS
Open Elaborated Design
SYNTHESIS
Run Synthesis
Open Synthesized Design
IMPLEMENTATION
Run Implementation
Open Implemented Design
PROGRAM AND DEBUG
PROJECT MANAGER - project_1
Sources
Design Sources
Constraints
Simulation Sc
sim_1
Utility Source
Hierarchy
Properties
Create Block Design@ee27
Please specify name of block design.
Design name: design_1
Directory: <Local to Project>
Specify source set: Design Sources
OK
Cancel
Project Summar
Overview
Dashboard
Settings Edit
Project name:
Project location:
Product family:
Project part:
Top module name:
Target language:
Simulator language:
Synthesis
Status:
Messages:
Tcl Console
Messages
Log
Reports
Design Runs
Name Constraints Status WNS TNS WHS THS TPWS Total Power Failed Routes
synth_1 constrs_1 Not started
impl_1 constrs_1 Not started
Create and add an IP subsystem to the project

加入ZYNQ CPU

- CPU是拿來做資料間傳輸的控制，
- 拿來運行電腦下過來的指令。



加入ZYNQ CPU

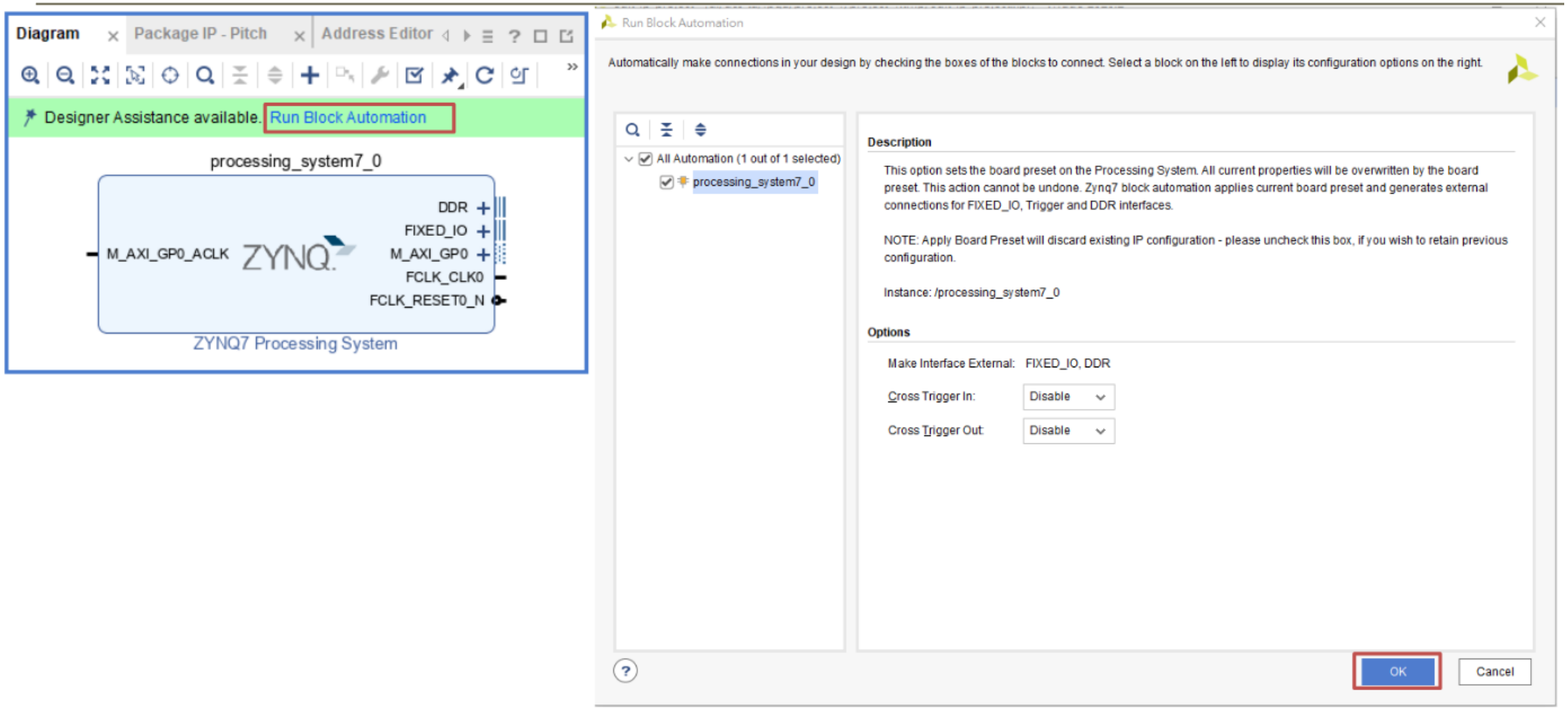
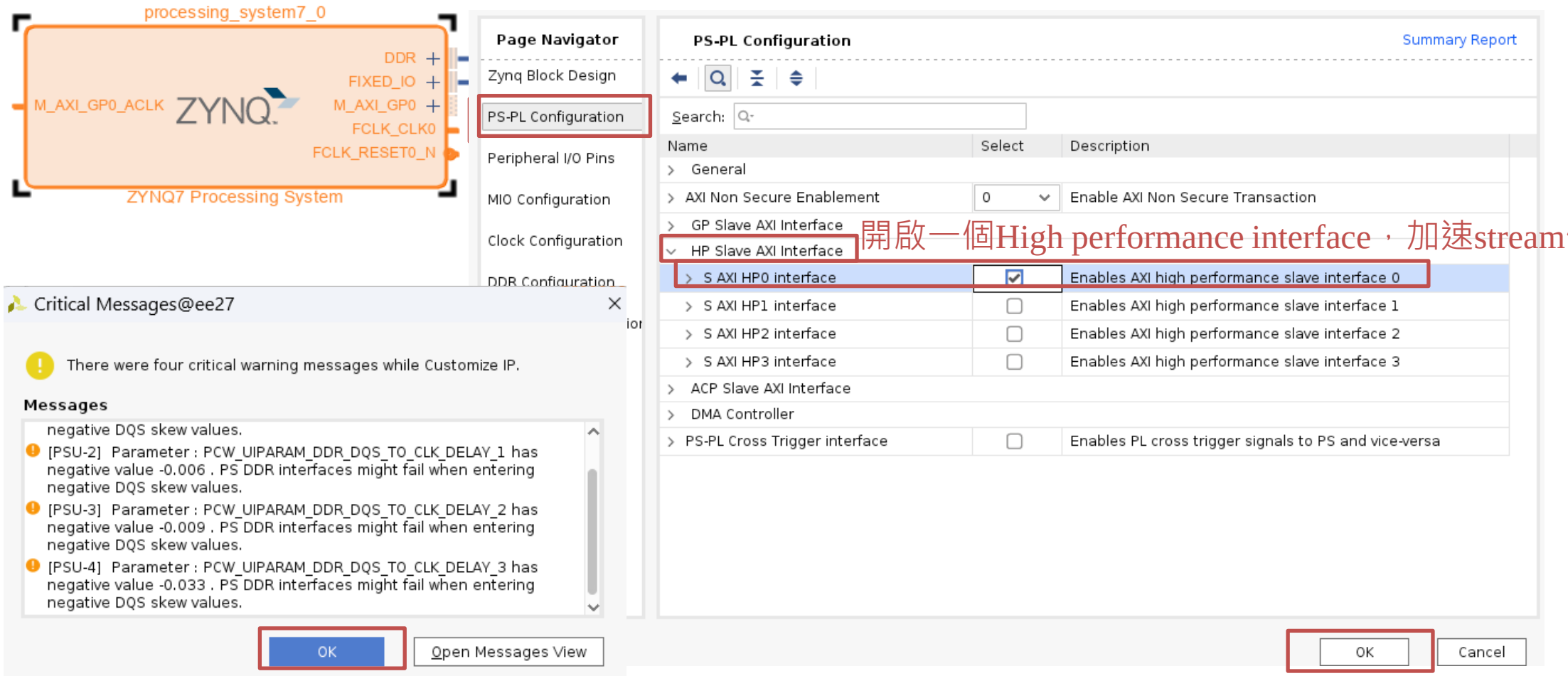


Diagram
Package IP - Pitch
Address Editor
Designer Assistance available.
Run Block Automation
processing_system7_0
M_AXI_GP0_ACLK
ZYNQ
DDR
FIXED_IO
M_AXI_GP0
FCLK_CLK0
FCLK_RESET0_N
ZYNQ7 Processing System
Run Block Automation
Automatically make connections in your design by checking the boxes of the blocks to connect. Select a block on the left to display its configuration options on the right.
All Automation (1 out of 1 selected)
processing_system7_0
Description
This option sets the board preset on the Processing System. All current properties will be overwritten by the board preset. This action cannot be undone. Zynq7 block automation applies current board preset and generates external connections for FIXED_IO, Trigger and DDR interfaces.
NOTE: Apply Board Preset will discard existing IP configuration - please uncheck this box, if you wish to retain previous configuration.
Instance: /processing_system7_0
Options
Make Interface External: FIXED_IO, DDR
Cross Trigger In: Disable
Cross Trigger Out: Disable
OK
Cancel

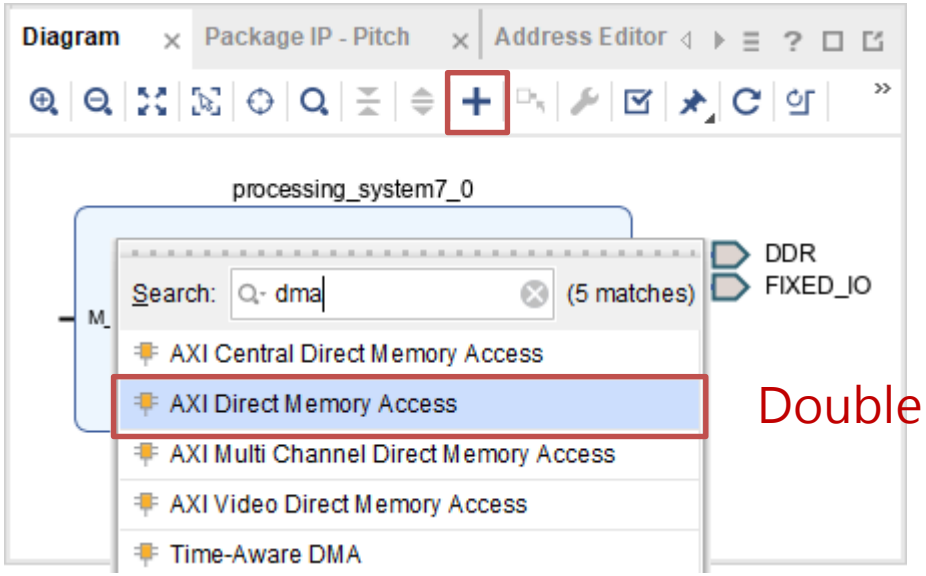
ZYNQ7 Processing System setup

- 左鍵點兩下ZYNQ7的Block



加入DMA

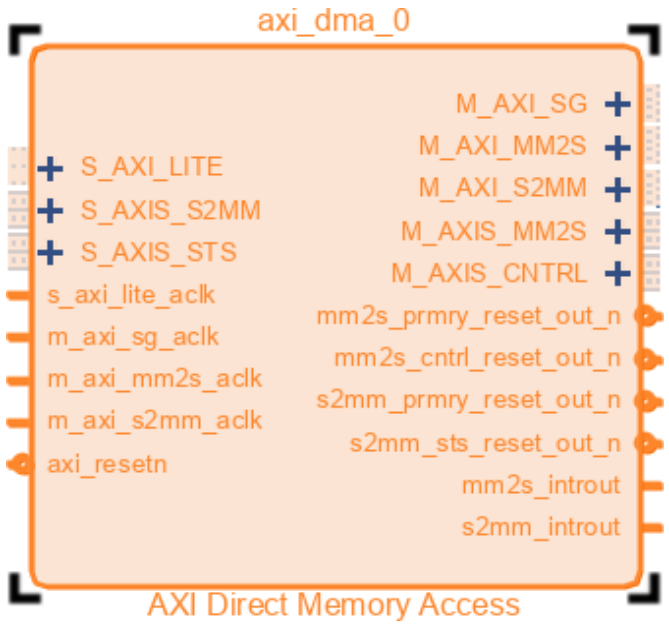
- DMA是用來與PYNQ板上的記憶體做溝通橋梁，
- 分配記憶體空間、存取記憶體中的資料，
- 取資料給你的Design、你的Design再寫資料回記憶體。

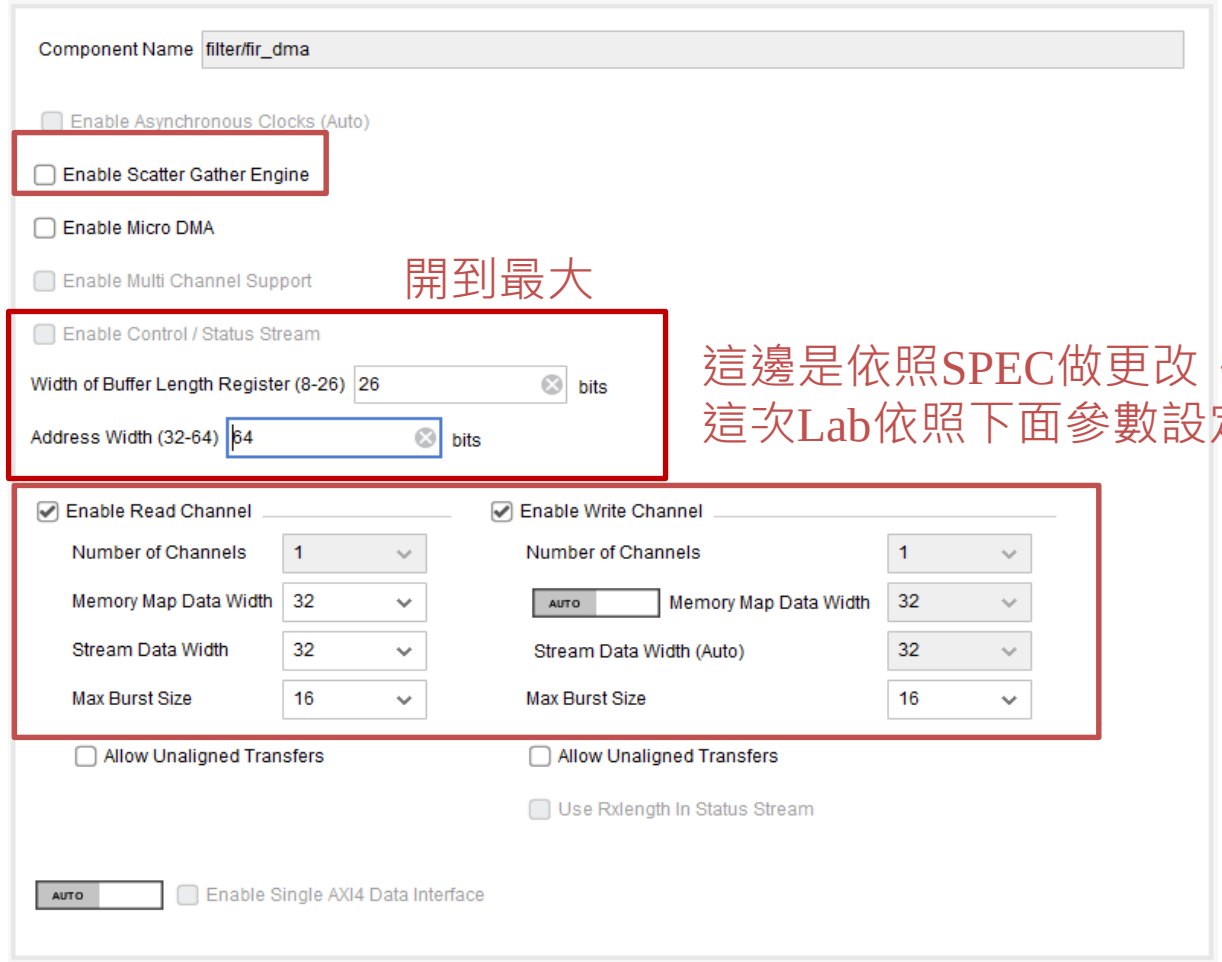


Double click

AXI DMA setup

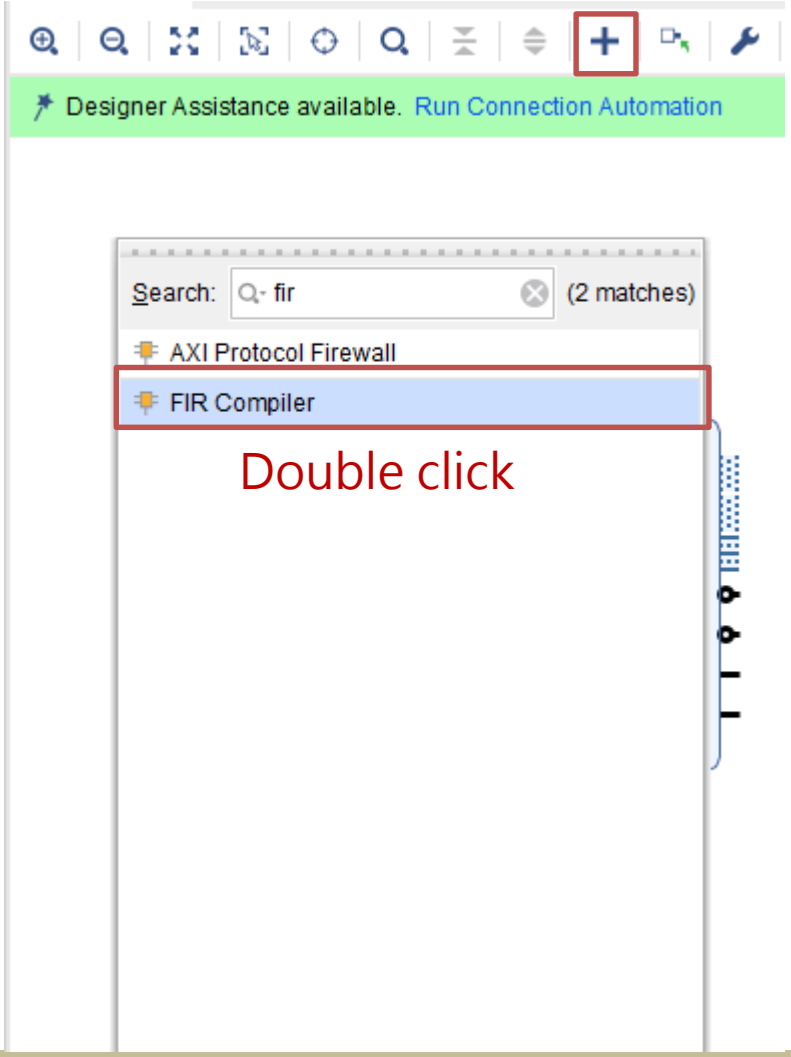
- 左鍵點兩下DMA的Block





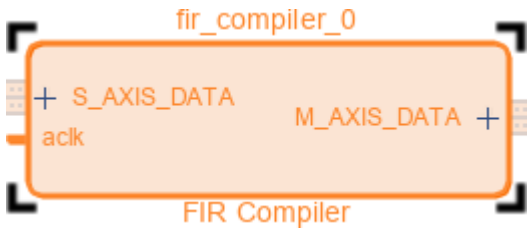
加入Project的IP

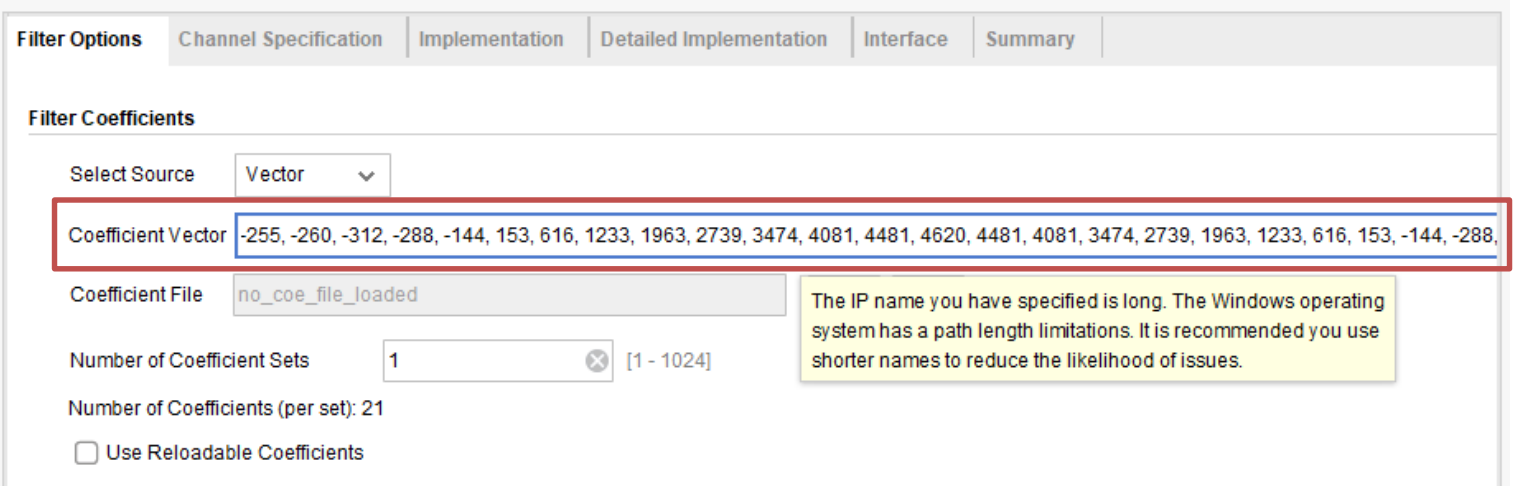
- 將寫好的.sv接上AXI4介面後包成IP，在這裡呼叫。
- 這次Lab使用內建FIR即可。



FIR setup

- 點兩下FIR的block





- 本次要做Sampling Rate 100MHz、
  Pass Band 0~5MHz、Stop Band 10M~50MHz的FIR LP Filter
- Coefficient Vector改成下面的數列:
  -255, -260, -312, -288, -144, 153, 616, 1233, 1963, 2739, 3474, 4081, 4481, 4620, 4481, 4081, 3474, 2739, 1963, 1233, 616, 153, -144, -288, -312, -260, -255

FIR setup

Filter Options | Channel Specification | Implementation | Detailed Implementation

Hardware Oversampling Specification

Select Format	Frequency Specification	
Sample Period (Clock Cycles)	1	[1.0 - 1.0E7]
Input Sampling Frequency (MHz)	100	[1.0E-6 - 189952.0]
Clock Frequency (MHz)	100	[0.390625 - 742.0]

Filter Options | Channel Specification | Implementation | Detailed Implementation | Interfa

Data Path Options

AUTO	Input Data Type	Signed
MANUAL	Input Data Width	32 [2 - 47]
AUTO	Input Data Fractional Bits	0 [0 - 32]
	Output Rounding Mode	Non Symmetric Rounding Up
	Output Width	32 [1 - 47]

Output Fractional Bits : 0

Filter Options | Channel Specification | Implementation | Detailed Implementation | Interface | Summary

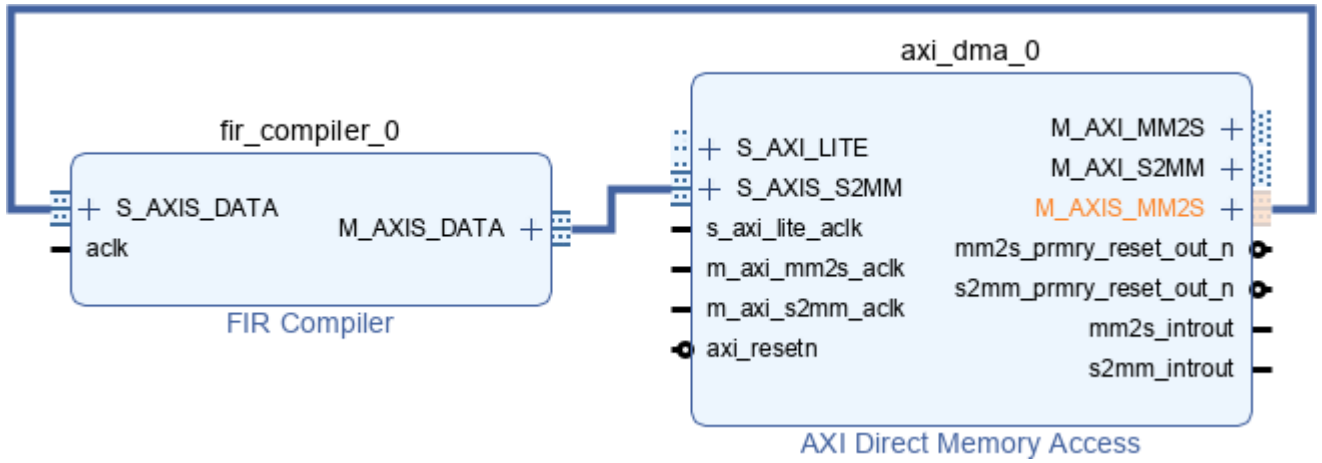
Data Channel Options

TLAST Packet Framing

☑ Output TREADY ☑ Input FIFO

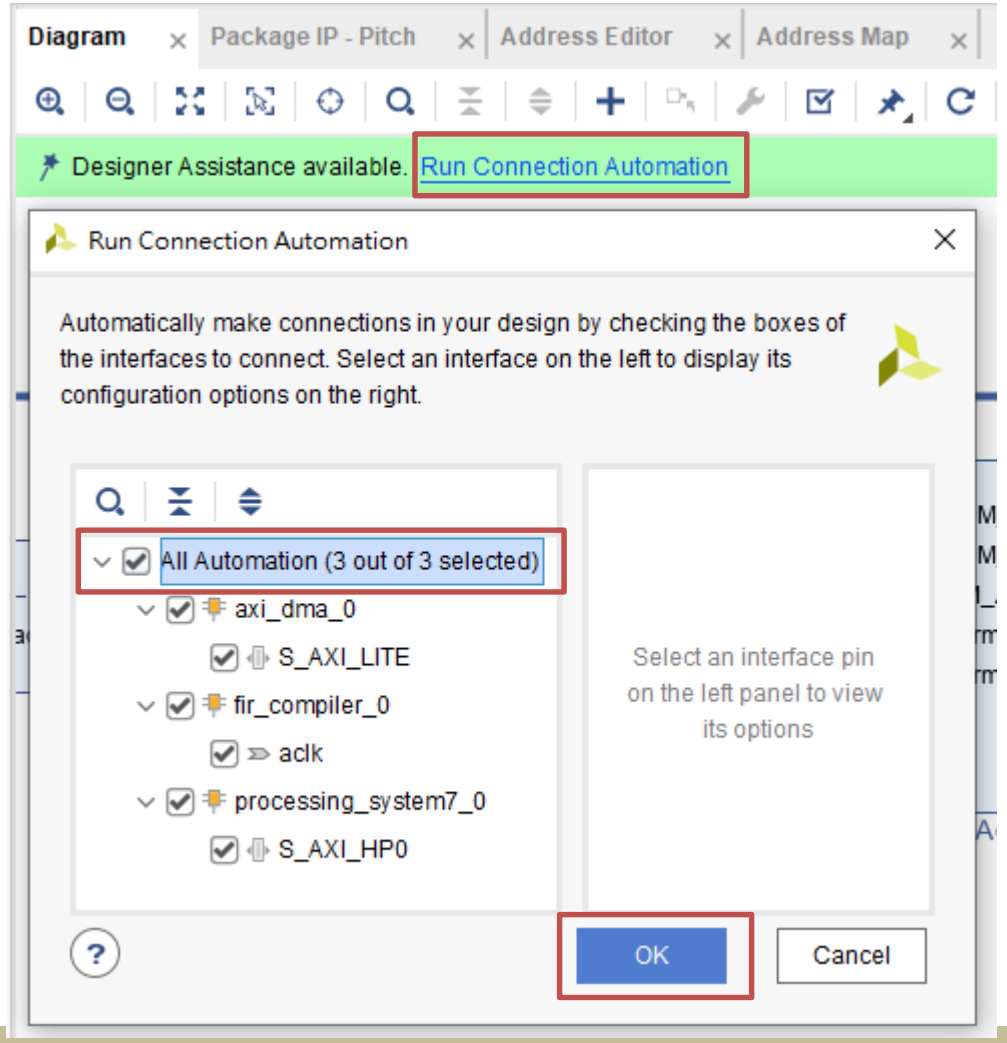
click ok

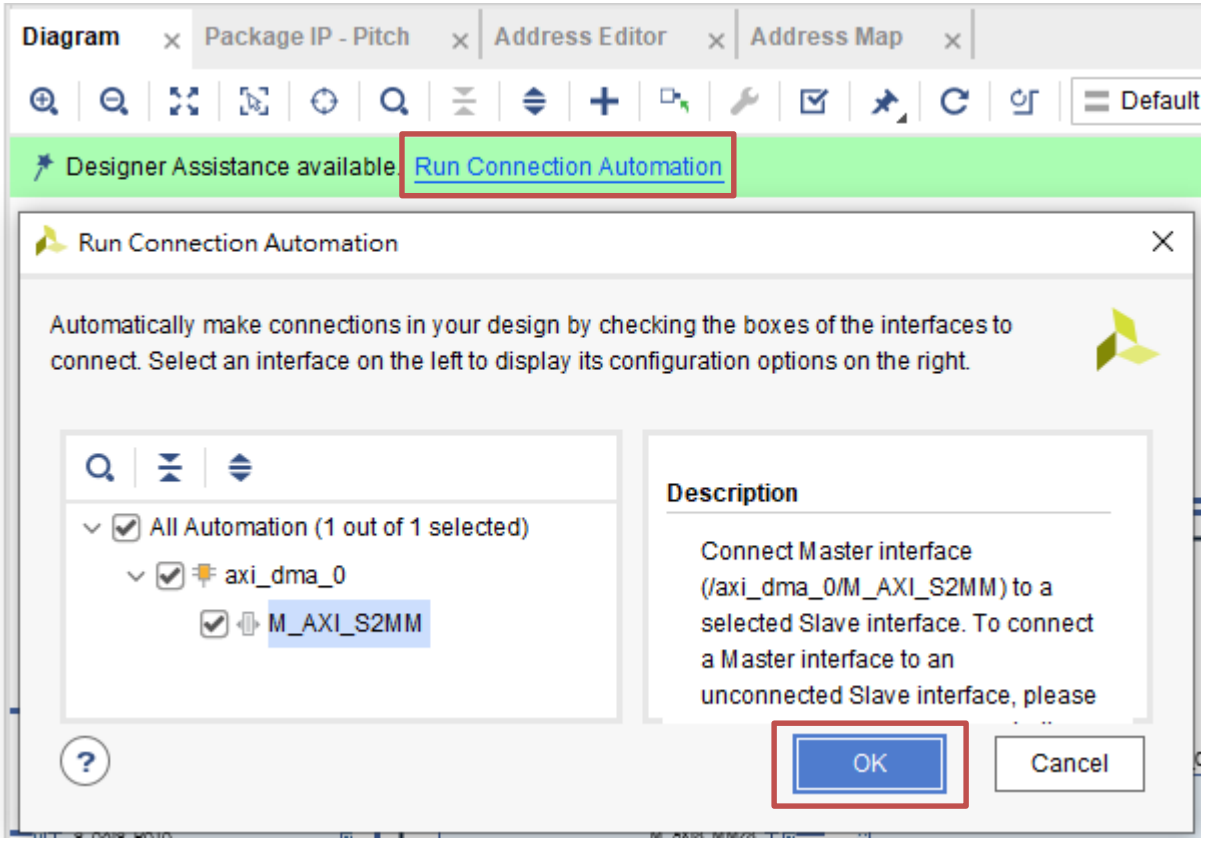
- 將FIR與DMA相連接，對著Port長按拖曳到另一個Port即可連接。



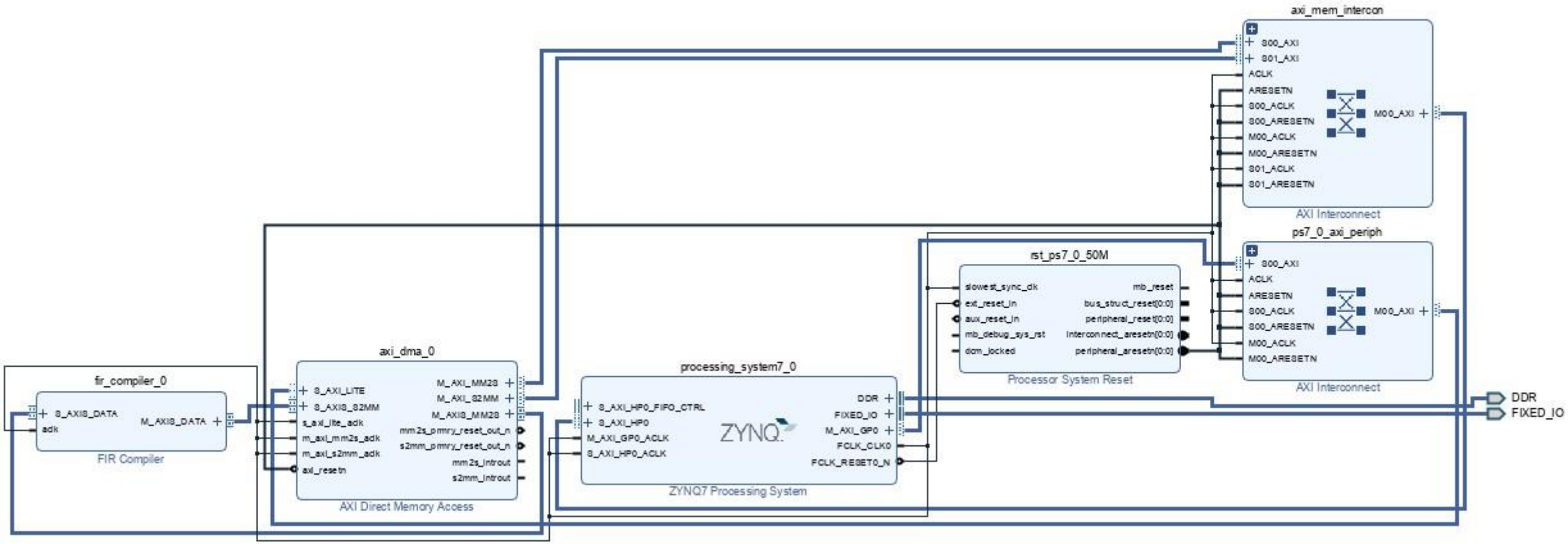
Run Connection Automation

- 跑兩次Run Connection Automation



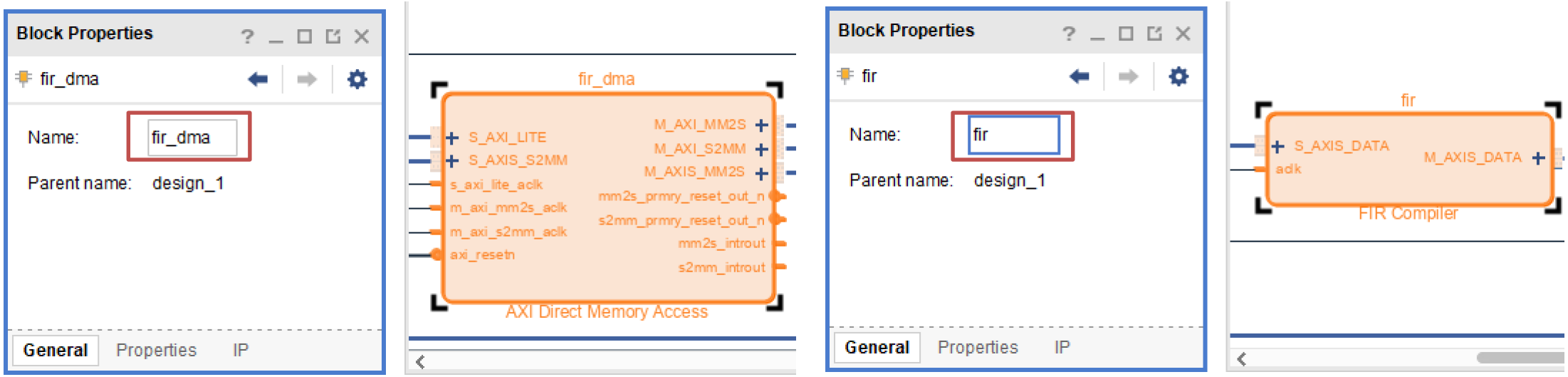


- 至此為止，Block Diagram會大致長這樣

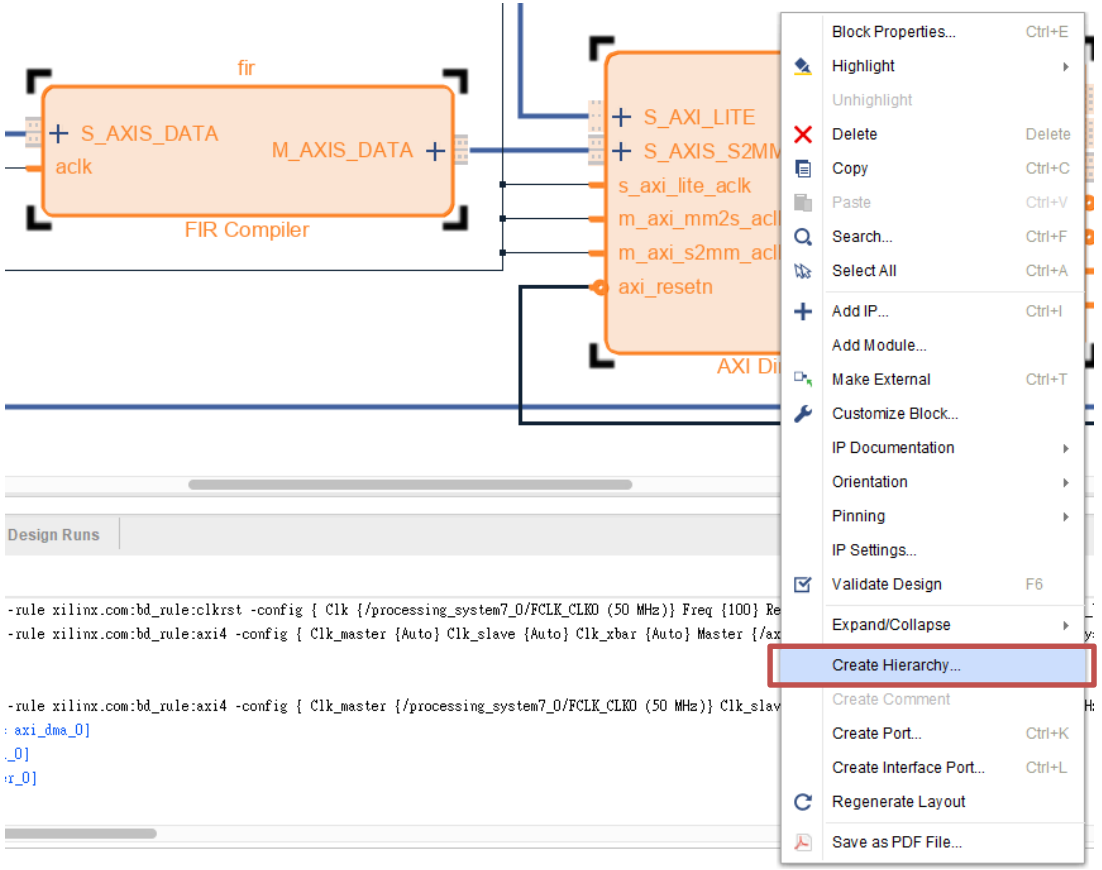


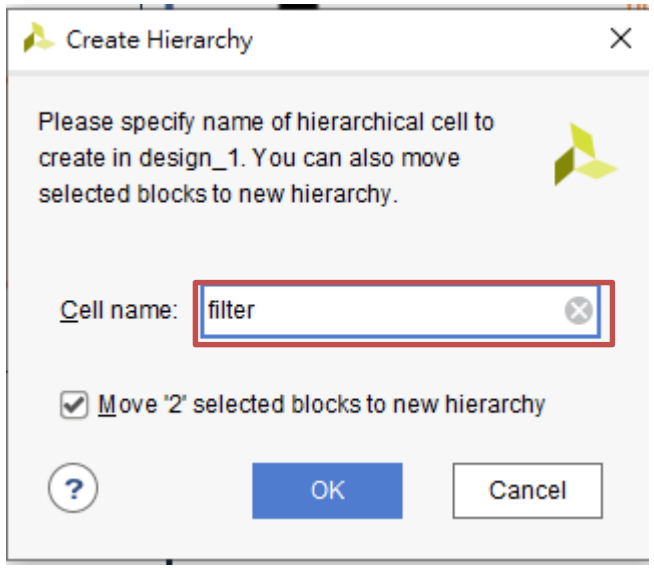
Rename

- 將fir_compiler_0和axi_dma_0的名稱改成fir和fir_dma，
- 以便在Jupyter notebook呼叫module。

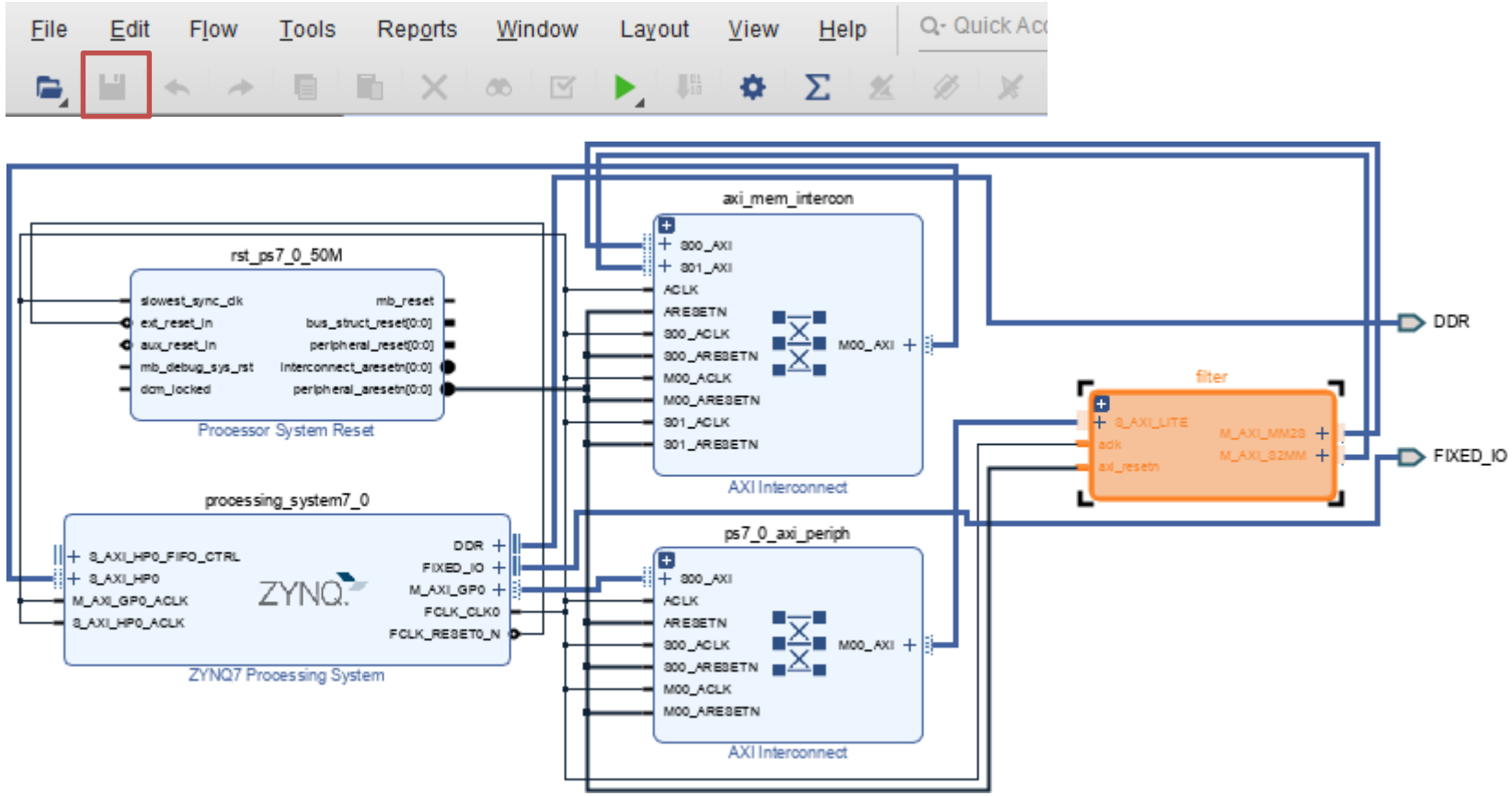


- Ctrl+左鍵，同時選取fir和fir_dma→右鍵→ Create Hierarchy
- 將fir和fir_fma包在一起，取名叫filter。

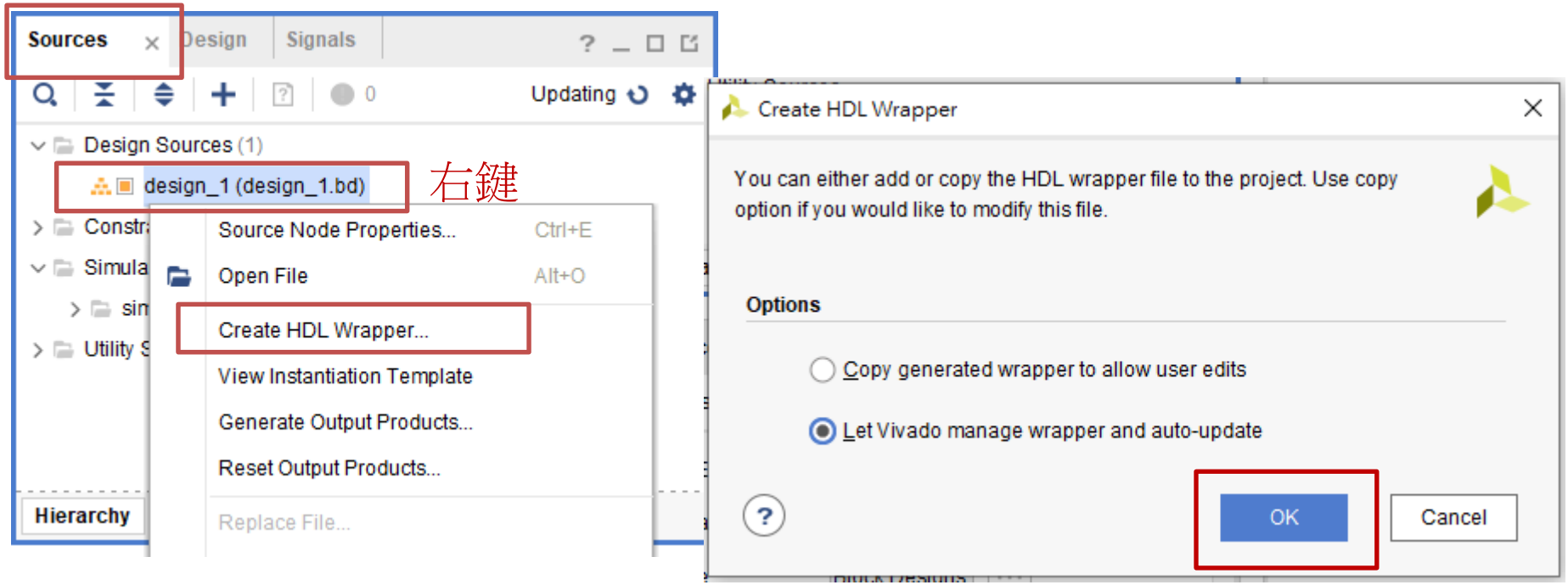




- 至此為止，Block Diagram會大致長這樣，可以先存檔。

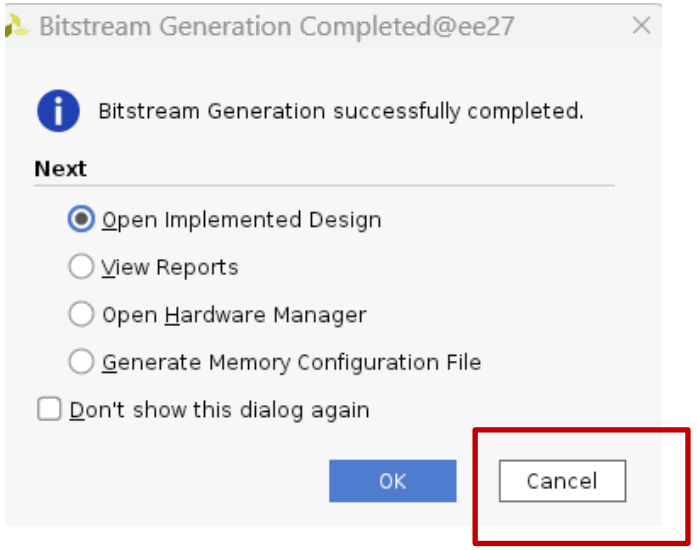


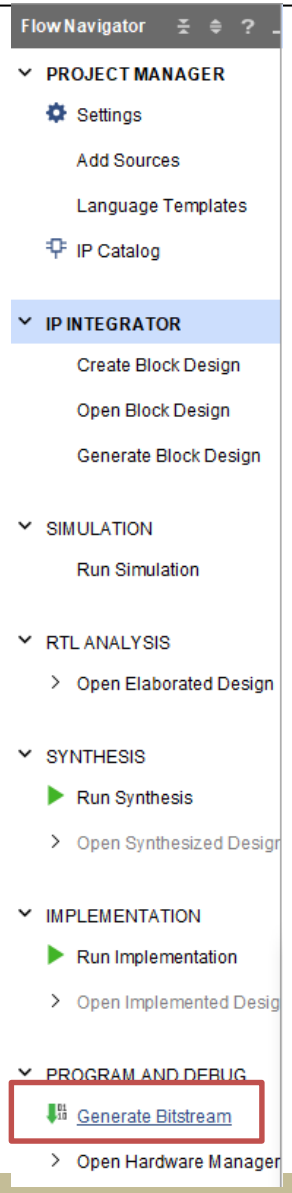
Generate HDL wrapper



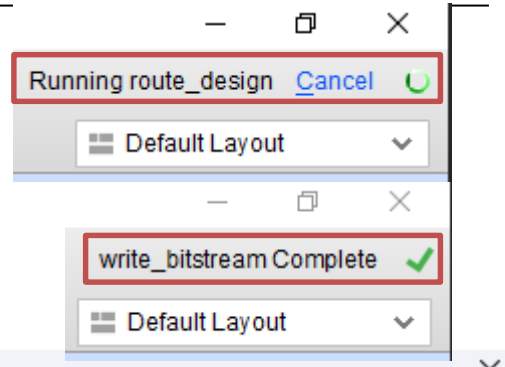
Generate Bitstream

- 這個會跑一段時間，
右上角沒有在跑東西之後
再往下一步。

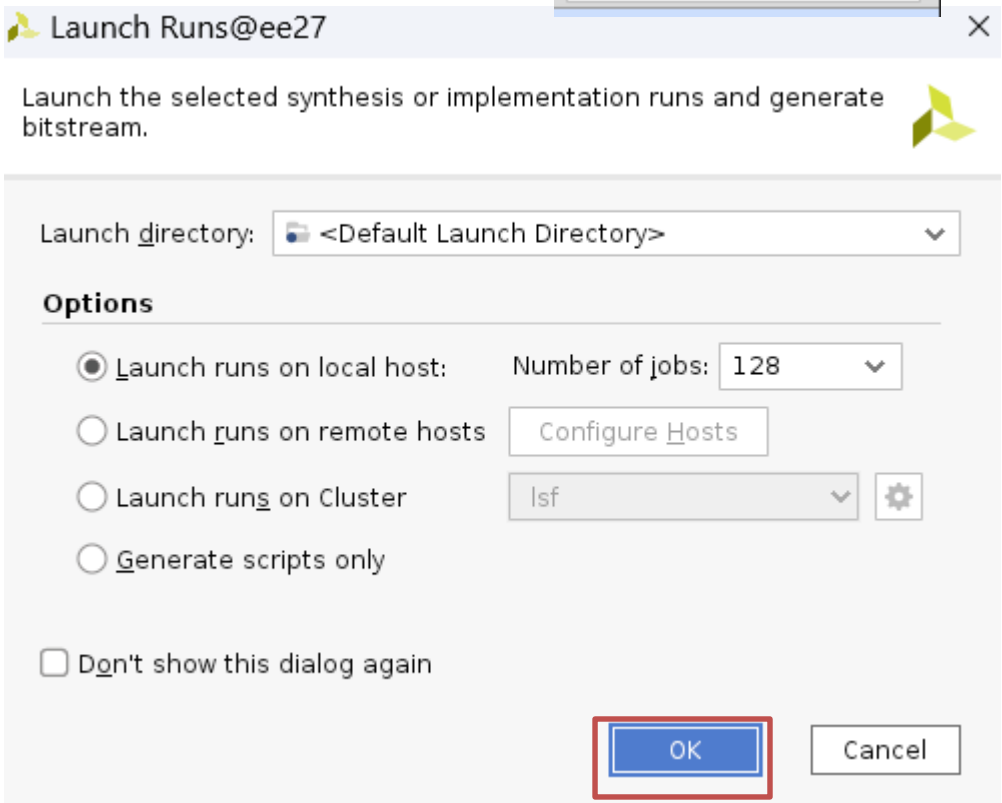




代表還在跑

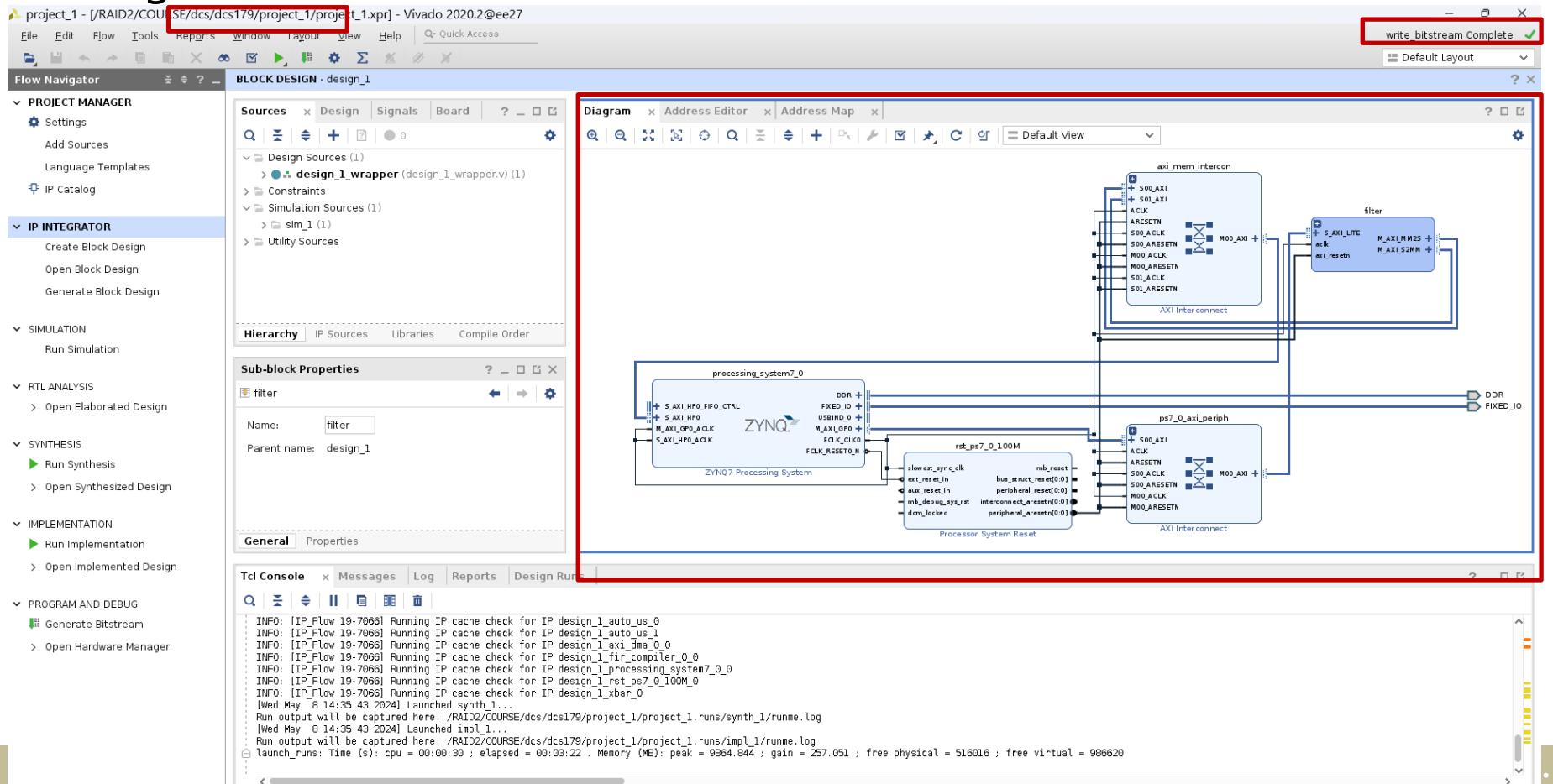


跑完了



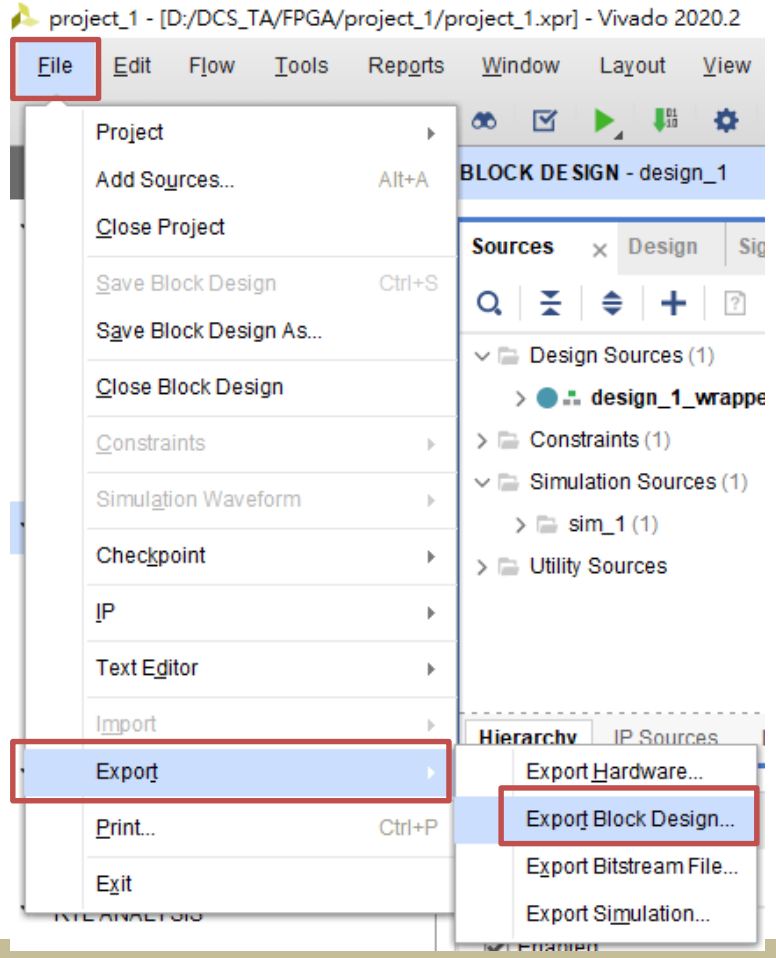
截圖vivado畫面 (demo依據之一)

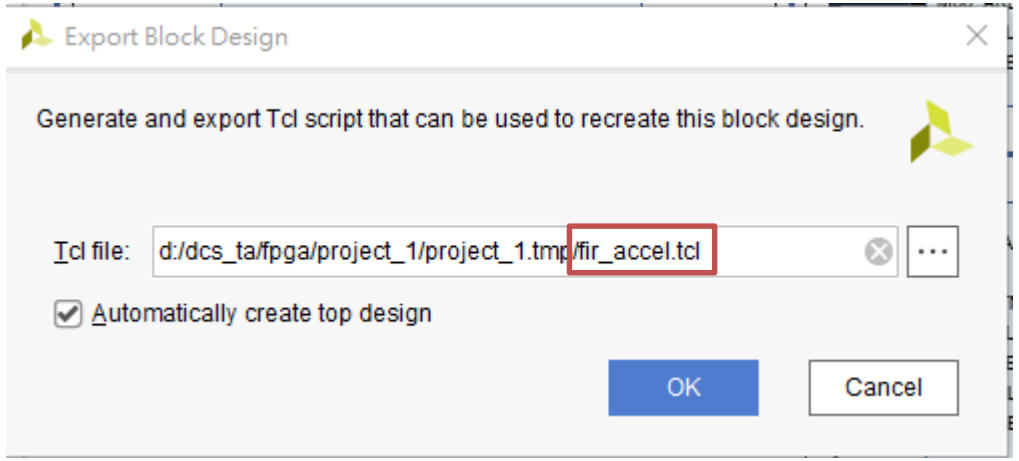
- 截圖須包含
  1. 工作站帳號
  2. Write _bitsream complete 勾勾
  3. 完整Diagram



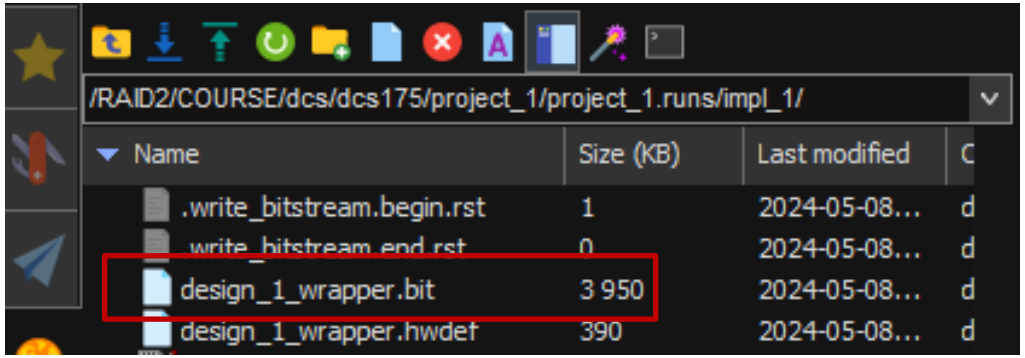
Generate tcl file

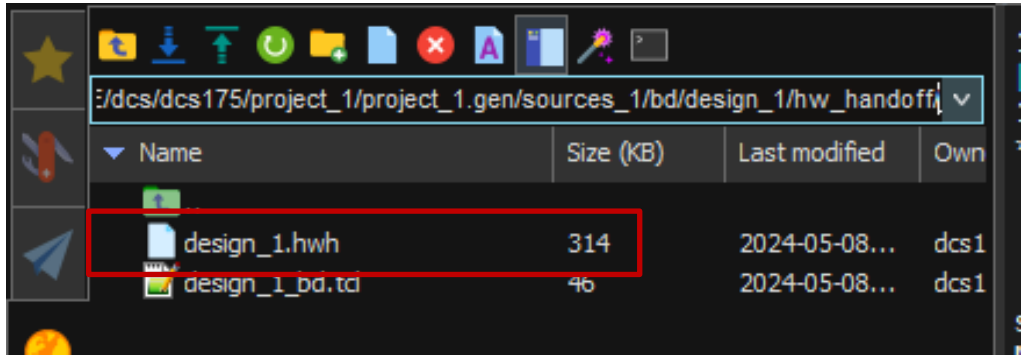
- 用Jupyter叫module overlay的時候，
- PYNQ需要讀tcl檔取得Hardware Design。

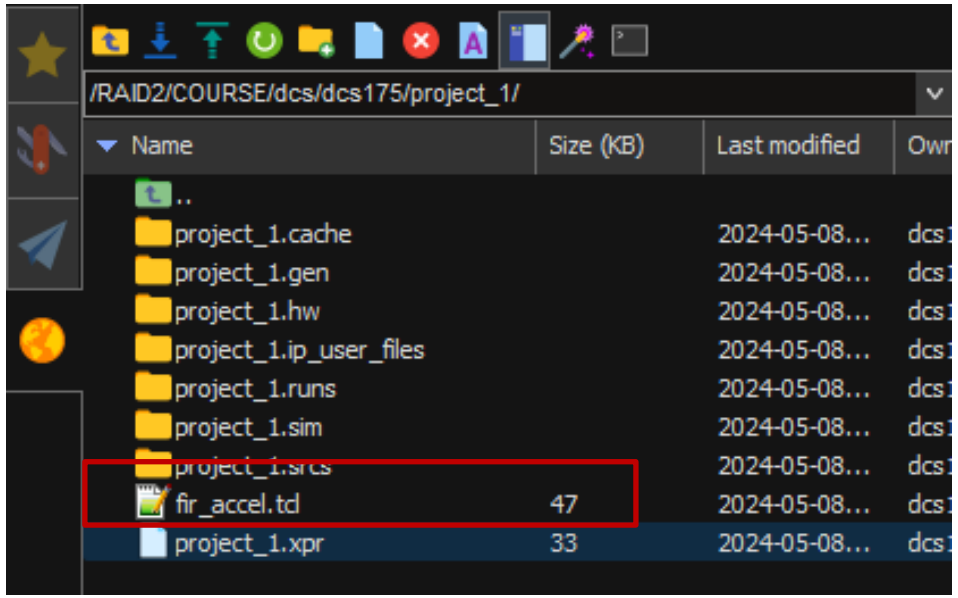




- 根據下面的路徑找到design_1_wrapper.bit、design_1.hwh 和 fir_accel.tcl，下載這三個檔案到自己的電腦，並將design_1_wrapper.bit、design_1.hwh重新命名成fir_accel.bit、 fir_accel.hwh







Connect to jupyter notebook

在表單上登記FPGA使用時間

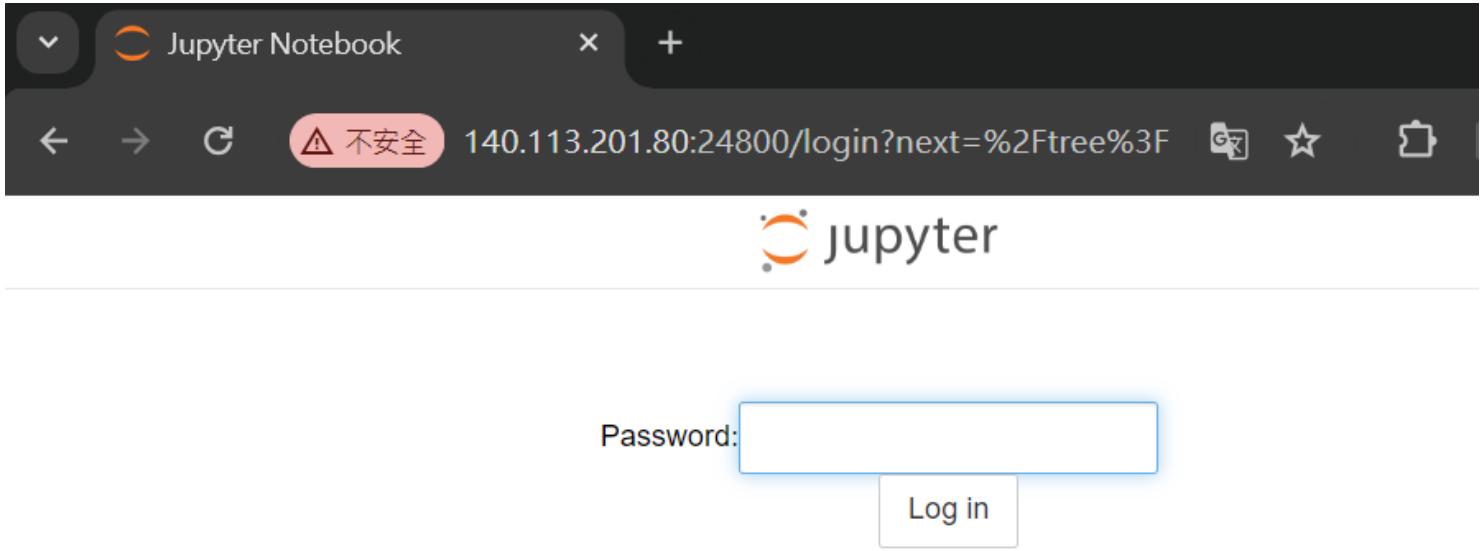
- 因FPGA數量只有48個，請同學輪流使用
- 請同學在連接上jupyter notebook前先確認板子是沒人使用的，避免跑到一半的結果被其他人洗掉

- 請同學先到下面網址登記使用時間，一個時段為2小時，如果在登記時段結束時還沒完成才可再登記下個時段
- https://docs.google.com/spreadsheets/d/1GCMX8VIO5hEjZ77t3UhFzwIPOfHLOxZT1oMgRcgrvYo/edit?usp=sharing

- 另外若板子有連線問題請在表單的註記欄打X，並先用其他板子

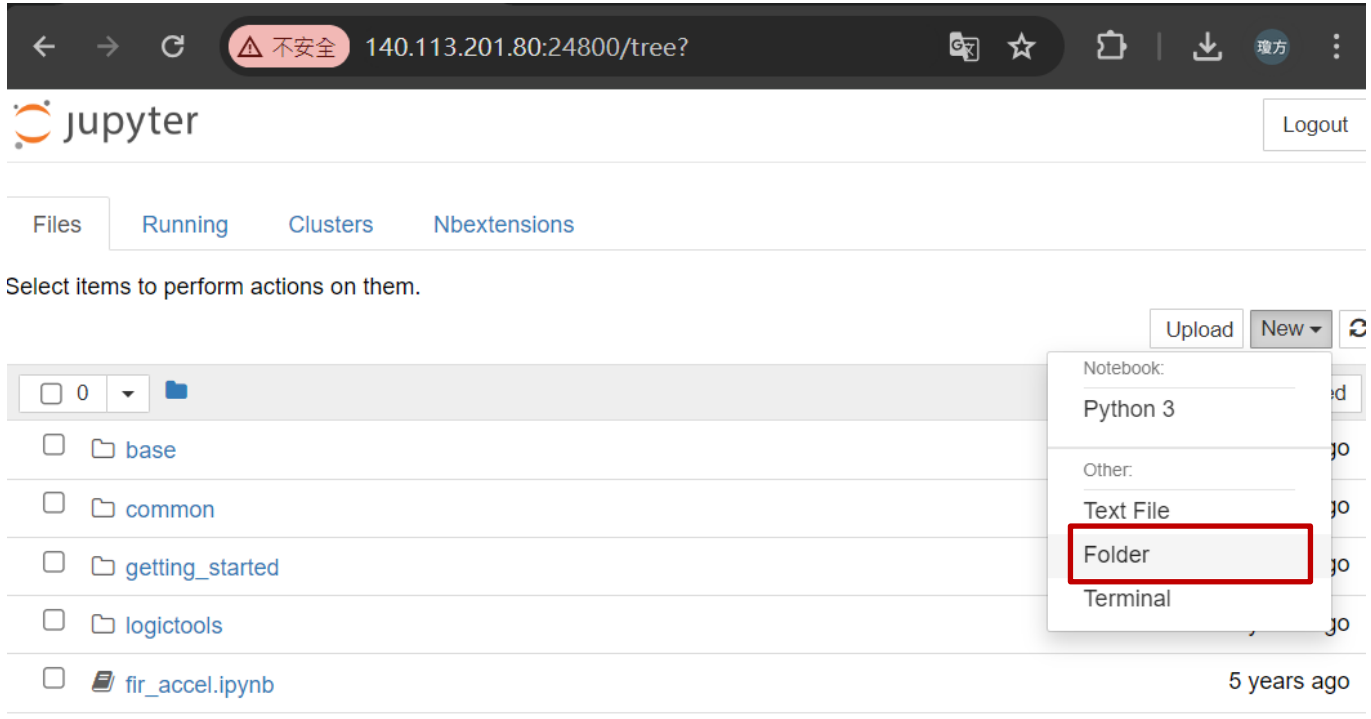
開啟Jupyter Notebook

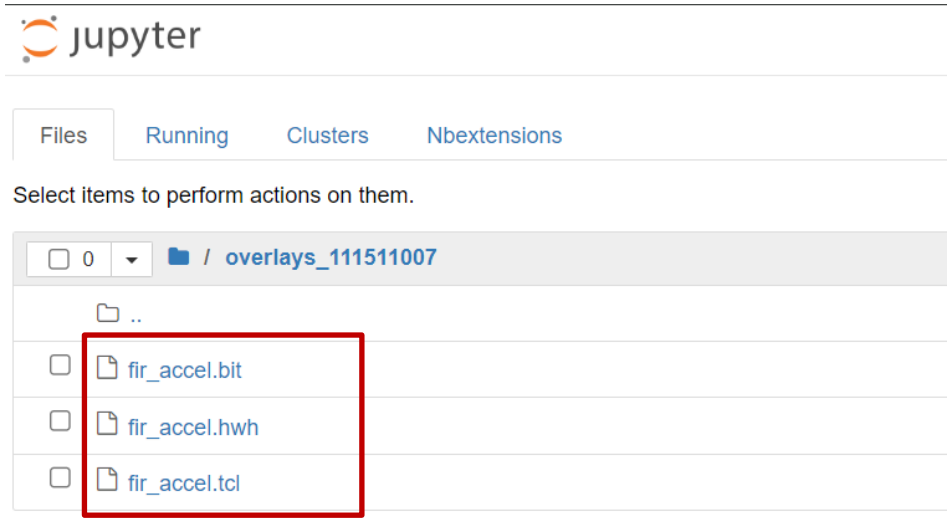
- 打開瀏覽器 → 輸入 140.113.201.80:板子編號
- 密碼: xilinx



上傳檔案到jupyter notebook

- 創建資料夾，命名為overlays_學號
- 將剛剛下載的三個檔案上傳至 overlays_學號 資料夾





fir_accel.ipynb

```
from pynq import Overlay
import pynq.lib.dma

# Load the overlay
overlay = Overlay('/home/xilinx/pynq/overlays/fir_accel/fir_accel.bit')

# Load the FIR DMA
dma = overlay.filter.fir_dma
```

//讀入Overlay

```
from pynq import Xlnk
import numpy as np

# Allocate buffers for the input and output signals
xlnk = Xlnk()
in_buffer = xlnk.cma_array(shape=(n,), dtype=np.int32)
out_buffer = xlnk.cma_array(shape=(n,), dtype=np.int32)

# Copy the samples to the in_buffer
np.copyto(in_buffer,samples)

# Trigger the DMA transfer and wait for the result
import time
start_time = time.time()
dma.sendchannel.transfer(in_buffer)
dma.recvchannel.transfer(out_buffer)
dma.sendchannel.wait()
dma.recvchannel.wait()

stop_time = time.time()
hw_exec_time = stop_time-start_time
print('Hardware FIR execution time: ',hw_exec_time)
print('Hardware acceleration factor: ',sw_exec_time / hw_exec_time)

# Plot to the notebook
plot_to_notebook(t,samples,1000,out_signal=out_buffer)

# Free the buffers
in_buffer.close()
out_buffer.close()
```

//xlnk申請記憶體空間，指標到虛擬記憶體

//分配記憶體空間給in_buffer和out_buffer

//將input資料寫入in_buffer

//開始進行DMA傳輸

//等待DMA傳輸結束

//釋放記憶體空間

fir_accel.ipynb

Hardware FIR implementation

In the following code blocks, we test out the hardware FIR implementation and measure it's performance.

```
from pynq import Overlay
import pynq.lib.dma

# Load the overlay
overlay = Overlay('/home/xilinx/jupyter_notebooks/overlays_111511007/fir_accel.bit')

# Load the FIR DMA
dma = overlay.filter.fir_dma
```

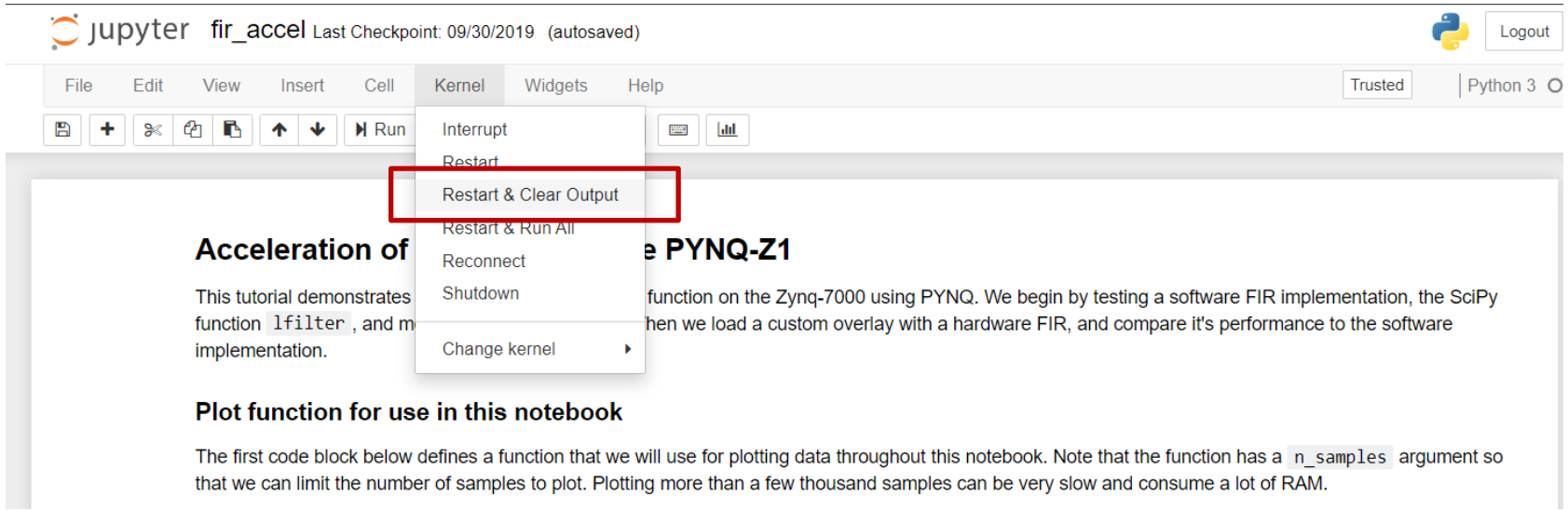
將路徑修改成你的資料夾名稱

fir_accel.ipynb

- 依序跑過所有的code並截下Software FIR filter using SciPy和Hardware FIR implementation的執行結果(兩張截圖)
- 兩個都須包含執行的execution time和input/output的圖形

注意事項

- 在連線到jupyter notebook前請先登記時段，且不要在非登記時段使用
- 產生檔案後記得要上傳到jupyter notebook
- 跑完結果後請將ipynb結果清除，並刪除你上傳的overlays_學號 資料夾



Demo

- Demo需要上傳兩個檔案到E3：vivado產生的bit檔，以及包含vivado畫面截圖、jupyter notebook兩張截圖的report，命名規則如下
  - dcsxxx.bit
  - dcsxxx_report.pdf
- Report可以只貼三張截圖就好，但請註明哪一張圖是對應哪個結果，沒註記會斟酌扣分
- Vivado 畫面截圖要求請參考p30， jupyter notebook截圖要求請參考p39，未達要求demo不通過
- 檔名錯誤扣5分，缺交任一檔案則不會通過demo
- 1de: 5/10 23:59