Lab 11
Jupyter & GPIO

Department of Electrical Engineering
National Cheng Kung University

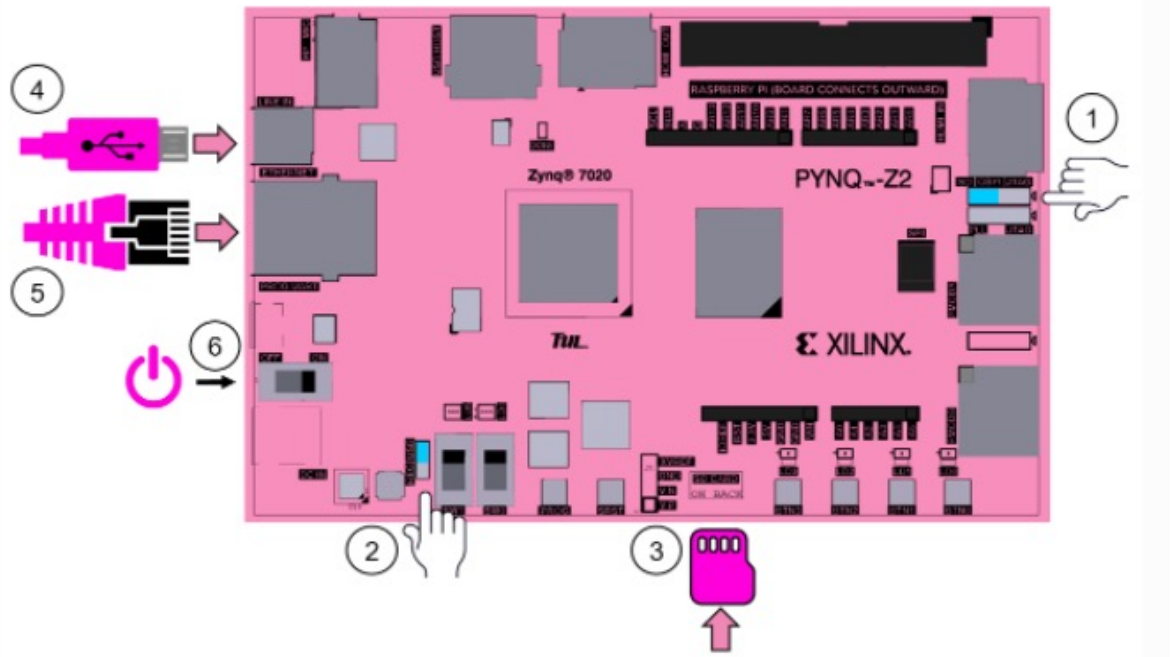
Outline

1. 什麼是PYNQ
2. Board setup
3. PYNQ overlay
4. 實驗一&實驗二
5. 課間檢查與結報內容
6. 參考資料

Board setup

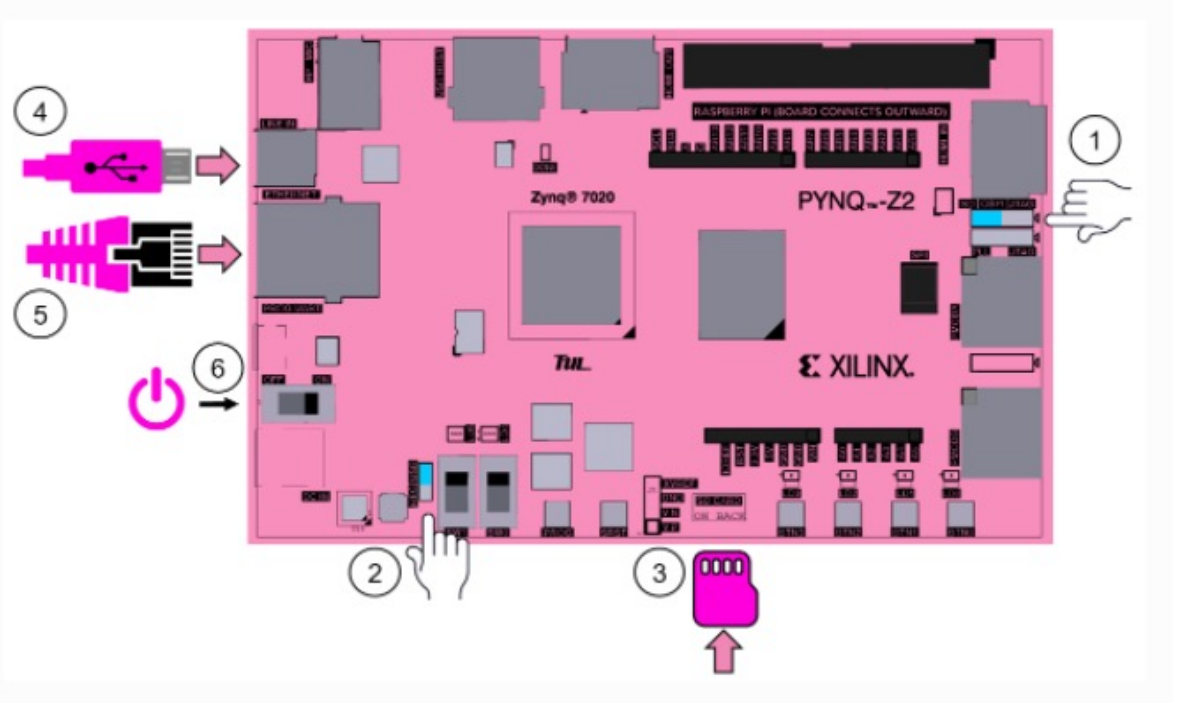
Board setup

- (1)設置開發版要從哪邊啟動(黑色小插座可拔插)
- (2)設置開發版供電方式，有USB和外接電源線兩種
- (3)將裝有 PYNQ-Z2 映像的Micro SD卡插入板下方的Micro SD卡插槽



Board setup (Cont.)

- (4) (5)將 USB 電纜及乙太線連接到 PC/筆記本電腦
- (6) 電源開關(開機後需檢查以下開機流程是否正常)



打開 PYNQ-Z2

- 紅色LED 將立即**亮起**以確認電路板已通電。幾秒鐘後，**黃色/綠色/完成**LED 將亮起，表明 Pynq® 設備正在運行
- 一分鐘後，會看到兩個**藍色 LED 和四個黃色/綠色** LED 同時閃爍。然後**藍色**LED 將打開和關閉，而**黃色/綠色**LED 保持亮起。
- 系統現已啟動並可以使用。

Pynq 連網

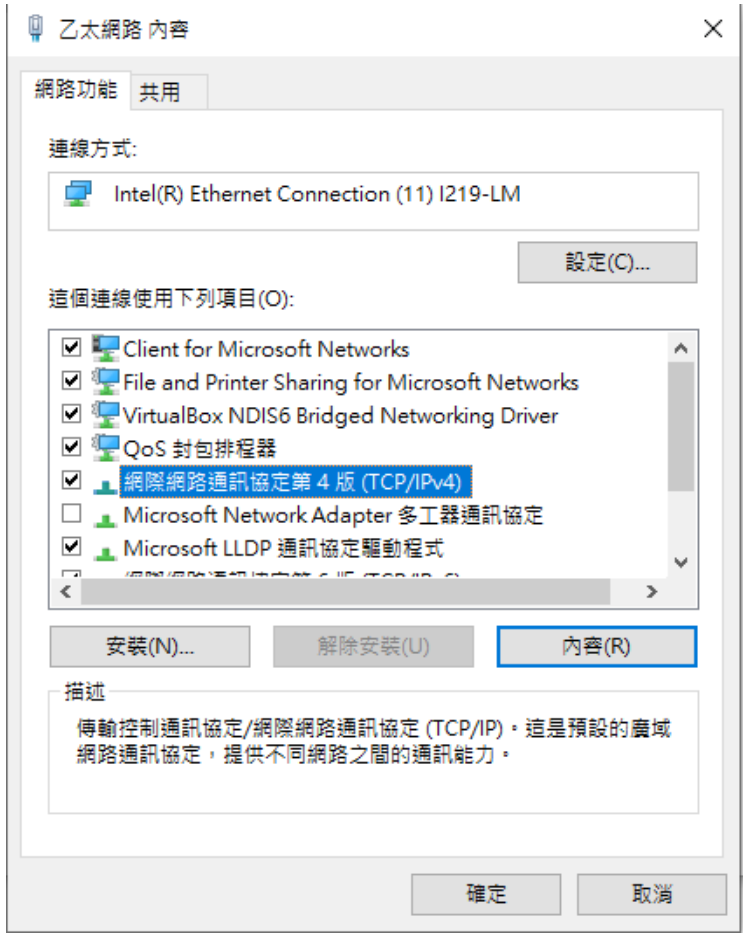
- 為了讓開發板可以在網路上實作，須將Pynq連上專用的Jupyter網頁

分配靜態IP

- 打開設定->網路和網際網路->變更介面卡選項
- 找到與電腦連接的開發版的網路（此例為2)

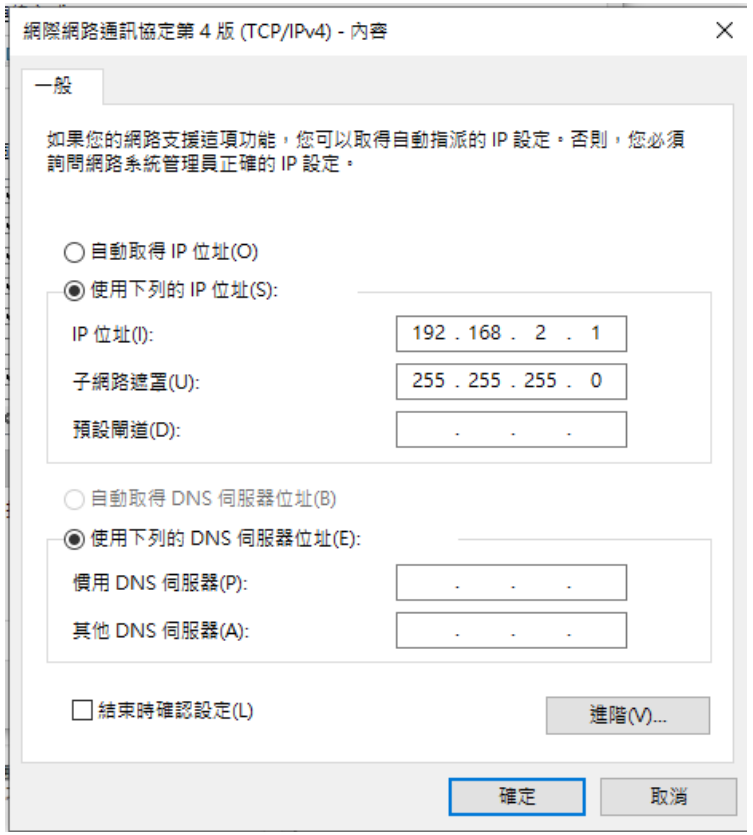
分配靜態IP (Cont.)

- 右鍵選擇內容->網際網路通訊協定(TCP/IPV4)



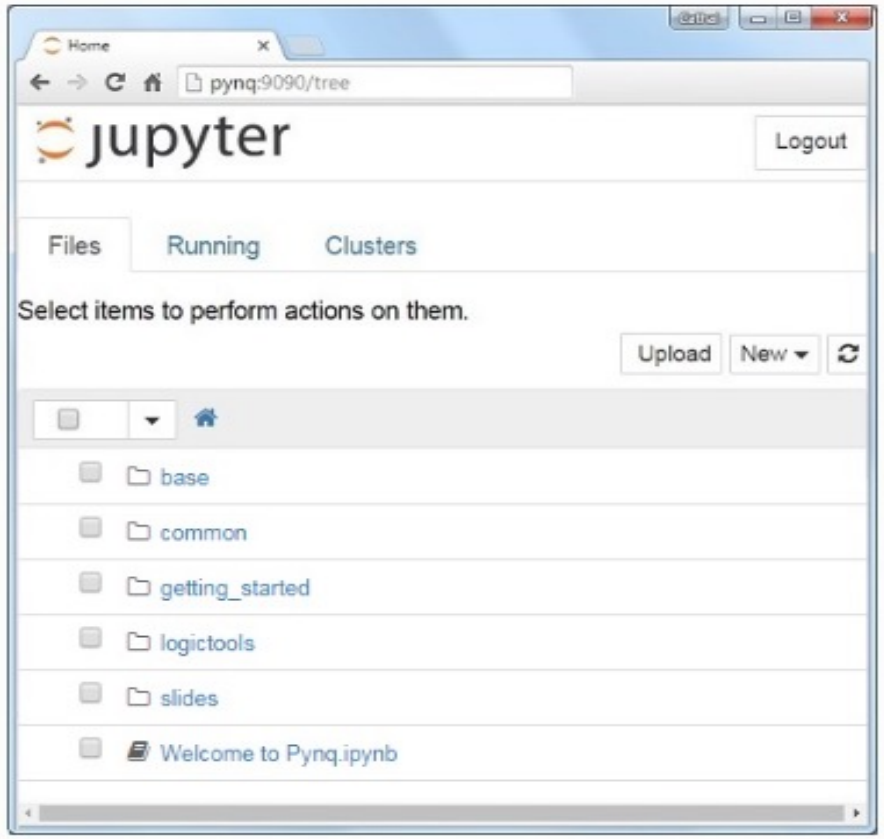
分配靜態IP (Cont.)

- 點選內容->將IP位址設為 192.168.2.x，其中 x 為 0-255（不包括 99，因為這已經被開發板佔用），遮罩設為255.255.255.0
- 單擊確定



連接到 Jupyter 筆記本

- 設置完成後，打開web瀏覽器網址輸入192.168.2.99
- 打開後會看到一個登入螢幕，帳號密碼均為Xilinx



Jupyter Notebook

Jupyter Notebook

- Jupyter notebook是一個介於IDE(Pycharm, Spider)以及Editor(Sublime text, Atom, VScode, 記事本)之間的工具，可以讓使用者在瀏覽器中撰寫及執行程式也有筆記的功能，並且支援多種語言。

Jupyter Notebook (Cont.)

- 創建一個ipynb檔

Jupyter Notebook (Cont.)

- 創建一個ipynb檔

工具列

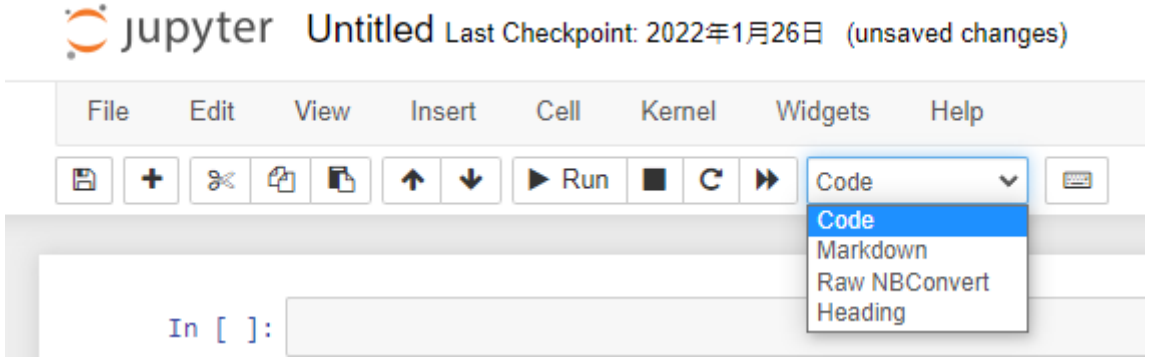
程式執行區塊(Cell)

Jupyter Notebook (Cont.)

- Cells:

  用來呈現文字或是程式碼的區塊，主要可分為兩種。

  - Code cell: 撰寫的程式碼
  - Markdown cell: markdown格式的文字



Jupyter Notebook (Cont.)

- Code cell:

```
In [1]: print("Hellow world!")

        Hellow world!
```

- Markdown cell:

```
# This is Heading

This is markdown
```

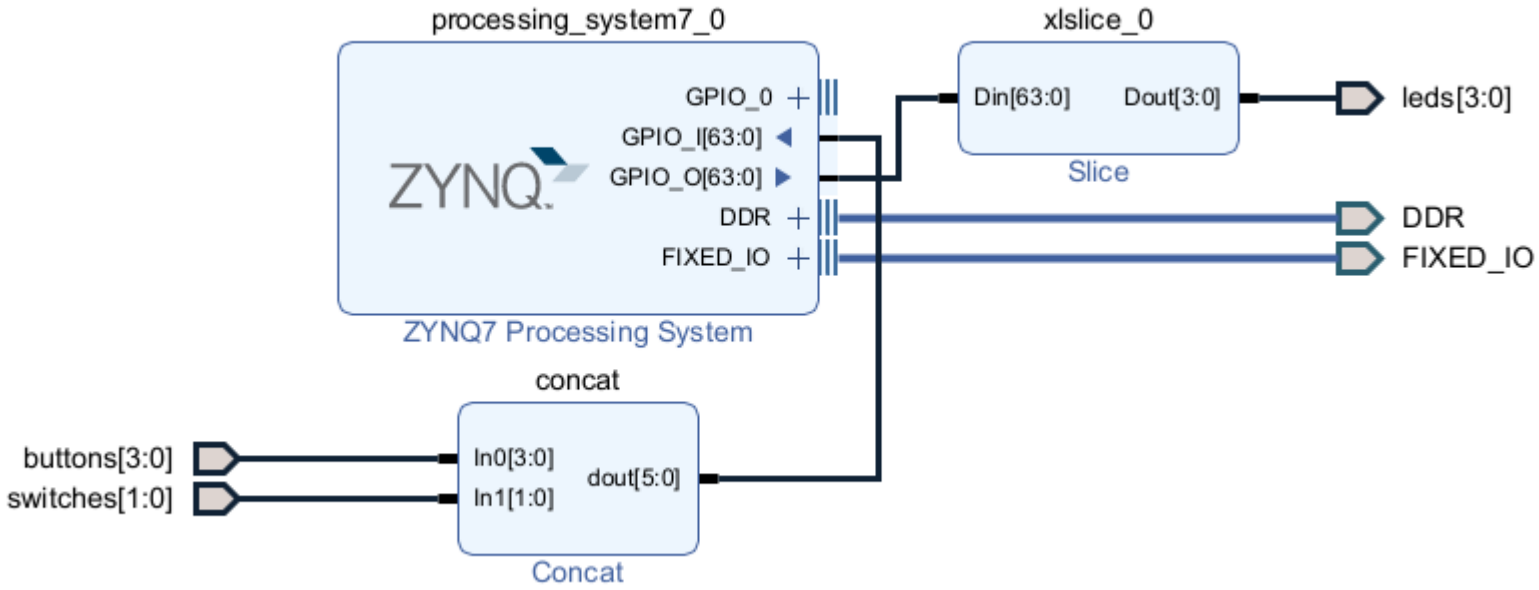
Run cell

This is Heading

This is markdown

實作題(一)

- 本實驗目的是展示如何使用 PYNQ 的 Zynq PS GPIO。PS GPIO是來自PS的simple wires，在programmable logic 也不需要controller來操控。而且他多達 64 個 PS GPIO 可用，它們可用於將簡單的控制和數據信號連接到 PL 中的 IP 或外圍設備。
- 利用buttons及switch當作輸入將輸出結果呈現在LED上

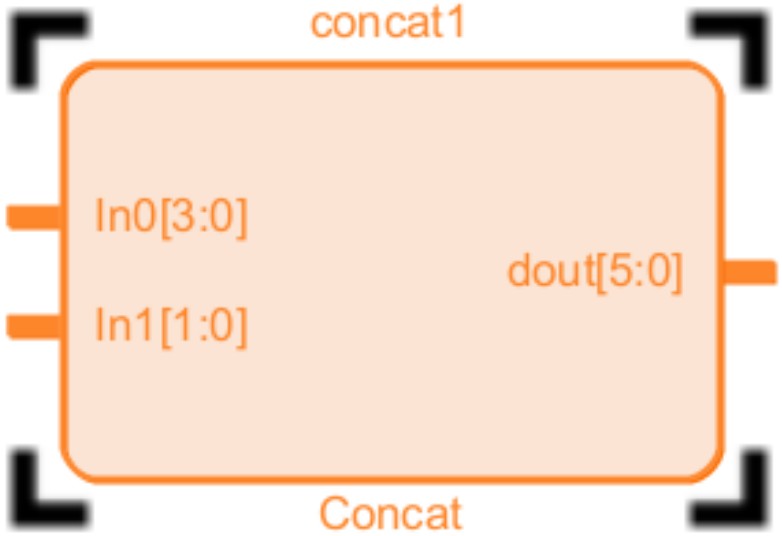


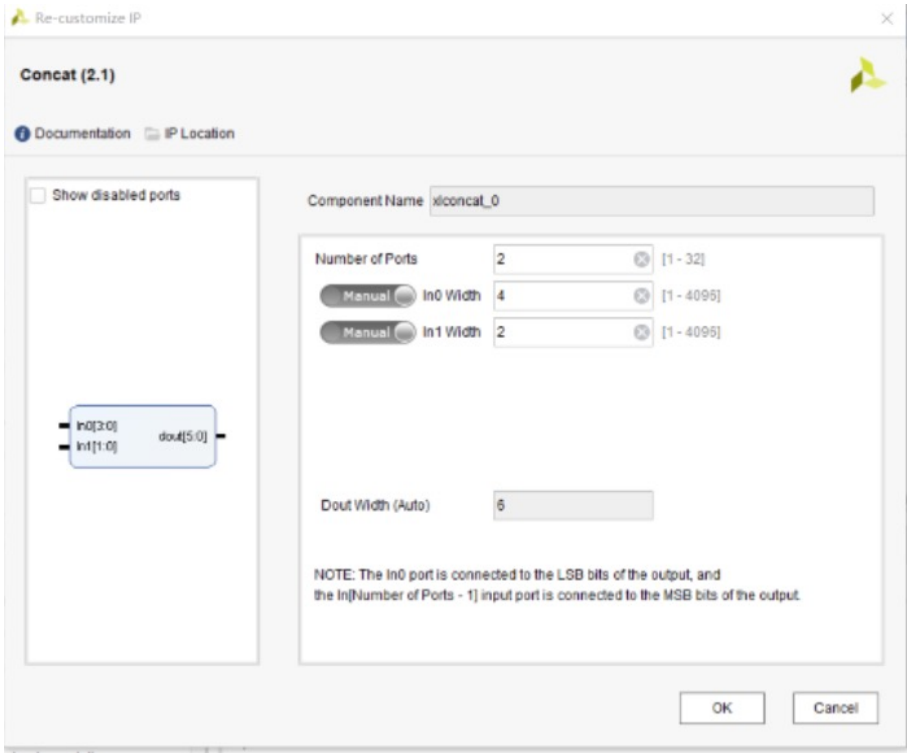
ZYNQ

- ZYNQ，簡單來說，就是兩大模塊的組合，可想成虛擬開發版
  - PS(Processing System)處理系統
  - PL(Programmable Logic)可編程邏輯

Concat

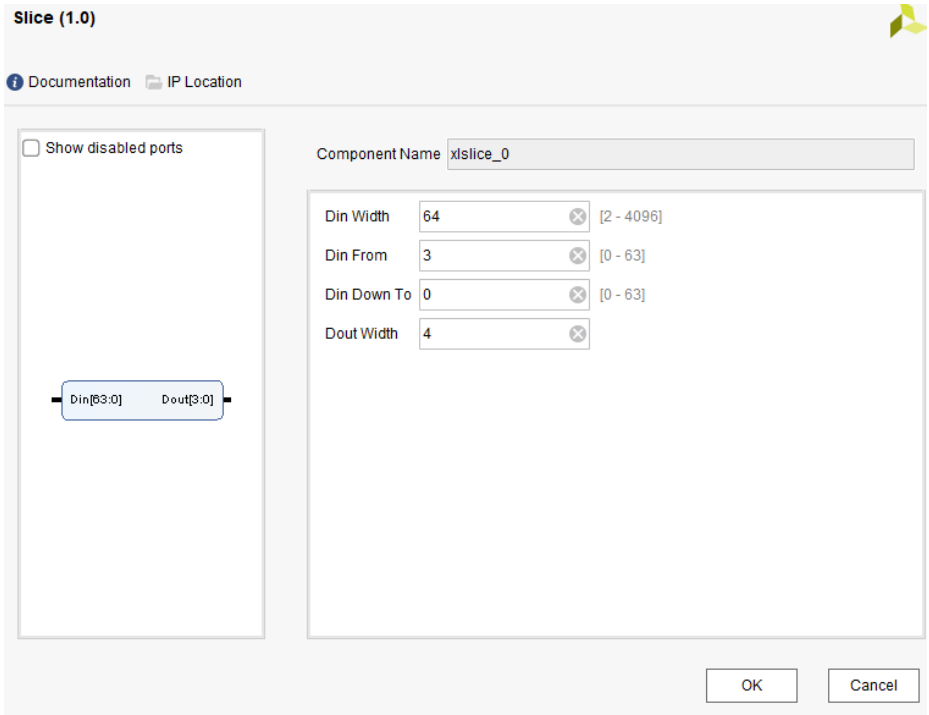
- 這個IP的實現的就是bit拼接的功能。這裡要將1個4bit和1個2bit的數據拼接為一個6bit數據。因此，Number of Ports設置為2，輸入端口的位寬都設置為4和2。





Slice

- Slice的功能正好和Concat功能相反。該功能是從一個多位寬的數據中提取1位或多位。輸入數據是64位，現在從中取低4位，故Din From一欄填3，Din Down To一欄填0，此時Dout Width會自動更新為4。



- 這次實驗主要是著重在操作Jupyter notebook，故我們已幫忙處理好bitstream轉換的相關作業

- 請在Jupter網頁中創建一個資料夾
- 將lab11-1中所有檔案上傳到該資料夾
- 在Jupyter中打開1_ps_gpio.ipynb檔案
- 然後逐次執行

實驗結果

- 當switch打開0/1,還有button0,1,2同時按住再執行Run便會得到以下結果

```
In [7]: print(f"Button0: {button0.read()}")
        print(f"Button1: {button1.read()}")
        print(f"Button2: {button2.read()}")
        print(f"Button3: {button3.read()}")

        print("")
        print(f"Switch0: {switch0.read()}")
        print(f"Switch1: {switch1.read()}")
```

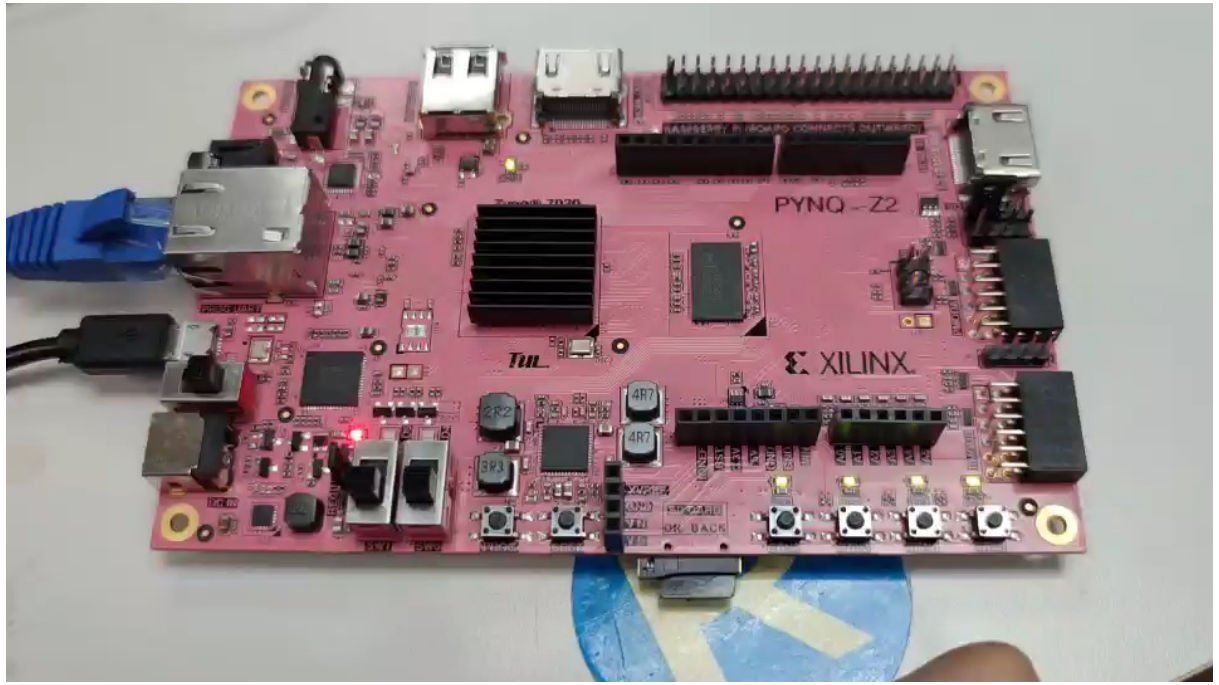
```
Button0: 1
Button1: 1
Button2: 1
Button3: 0

Switch0: 1
Switch1: 1
```

實驗成功影片

實驗二

- 請參考助教提供的檔案及官方所提供的base overlay，以python撰寫一支程式來控制控制LED、RGB LED的亮燈方式。



實驗二 (Cont.)

- 程式要求:
  - 程式執行最初先將4個LED燈全部亮起
  - button[0]: 控制RGB LED照顏色順序亮燈
  - button[1]: 控制LED以右至左的方式依序亮燈
  - button[2]: 控制LED以左至右的方式依序亮燈
  - button[3]: 結束程式

Base Overlay

- buttons[3:0]
- leds[3:0]
- rgbleds[5:4]
- color_idx: 0~7

Function

- 讀取button值:
  - base.buttons[idx].read()
- 控制亮燈:
  - base.leds[idx].on(), base.leds[idx].off()
  - base.leds[idx].toggle()
  - base.regleds[idx].write(color_idx)
- 使用time library
  - from time import sleep
  - sleep(sleep_time)

課間檢查與結報內容

課間檢查與結報內容

- 課間檢查
  - 實驗結果
- 結報內容
  - 實驗心得

参考資料

- **PYNQ-Z2 設置指南**
- **https://blog.csdn.net/CSD_N_csdn/article/details/106383090**