為了將來能夠將整個 TCBNN 實作成硬體並放入 pynq 板子，先嘗試將簡單的神經網路做成硬體放入 pynq 中。

實驗:簡單神經網路以 vitis HLS 合成，並且做成 IP 放進 pynq 驗證成功。

首先要建構一個簡單的神經網路，我以 MNIST 為資料集，訓練三層神經網路，準確度為 97.9%。

Python 軟體

Dataset :MNIST

Accuracy:96.9

Neural network structure:

```
class NeuralNetwork(nn.Module):
  def __init__(self):
    super(NeuralNetwork, self).__init__()
    self.flatten = nn.Flatten()
    self.linear_relu_stack = nn.Sequential(
      nn.Linear(28*28, 64),
      nn.ReLU(inplace=True),
      nn.Linear(64,32),
      nn.ReLU(inplace=True),
      nn.Linear(32, 10),
    )
```

Vitis HLS

為了要將 Python 合成為硬體並放入 pynq 中交給 VITIS HLS 合成，我將神經網路架構寫成 C++格式。(如附檔)

在 pynq 當中，設計 IP 與 Host 的傳輸介面有 Axi-lite,Axi-master,Axi-stream 等等。由於 Axi-lite 一次傳輸的資料量太少，Stream 傳輸方式尚未實驗成功，所以這次實作內容是以 Axi-Master 的傳輸方式來實現。

Axi-master 傳輸介面可以以指標或者陣列的資料格式傳輸，在此有兩種傳輸模式[1]:

1.individual data transfer: 以傳輸的地址讀或寫單一筆資料。

2.burst mode: 以傳輸的地址讀或寫多筆連續的資料，連續的長度(burst-length)由使用者定義，以 C 的 memcpy 的方式實現。在 vivado 中，由於硬體規格的限制，單次傳輸資料的上限為 512bit。

本次實驗方式是以 burst mode 傳輸神經網路內的權重,偏差和輸出入資料。

將權重,偏差,以及輸入圖片先放進 pynq 板內的 global memory,再藉由 Host(pynq) 的 axi-lite 控制資料輸入，將圖片、權重傳入設計的 IP 內，當 IP 運算完成後，會將結果 output 到 global memory，最後將 output 結果再 Host 控制取出運算結果。

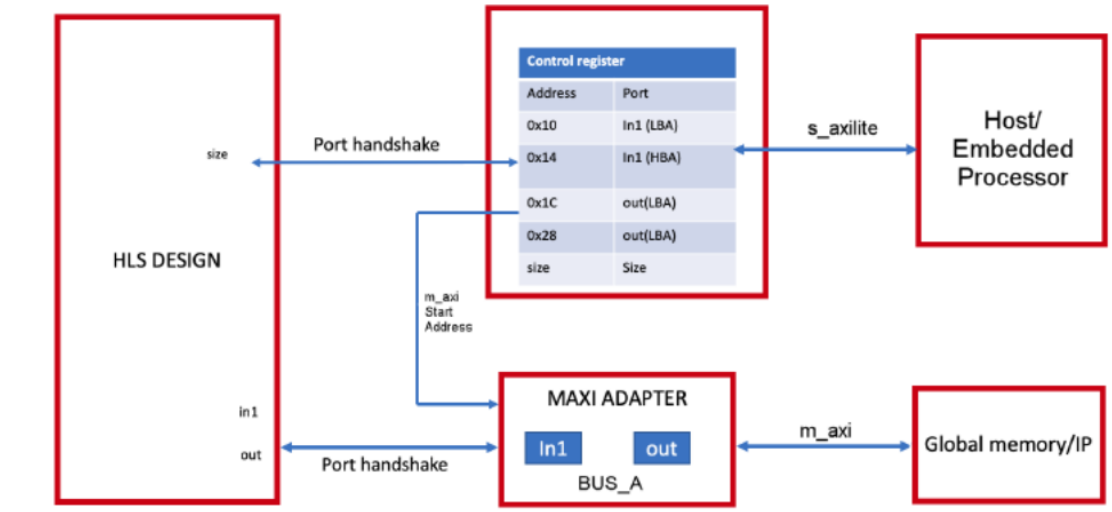

Figure 1 m_axi interface[1]

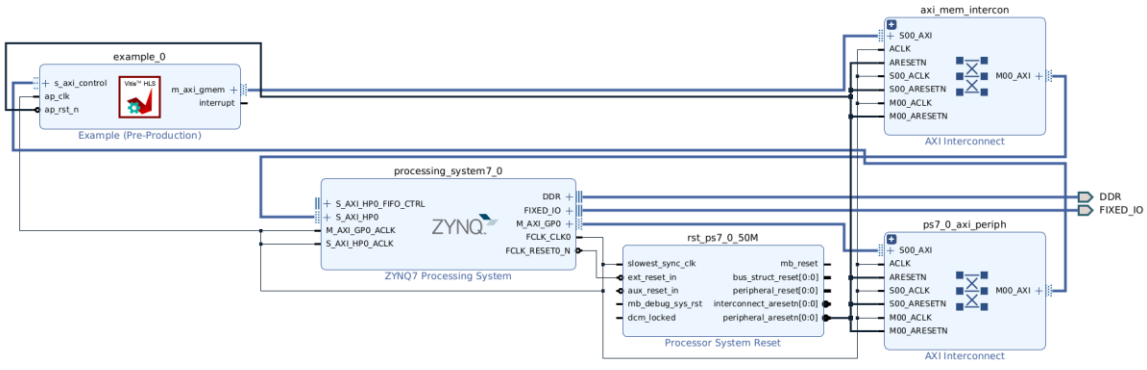

Figure 2 vivado diagram

Pynq 驗證

最後將 C++所設計的 kernel 包成 IP 丟進 vivado 做成 bi t stream 和 hwh 檔案傳到 pynq 板子上做驗證，看看實作結果是否與軟體算出來的結果是否一致，和比較速度。

我將資料以 np.save 的方式存成 npy 放入 pynq 中，比較兩種方式運算 10 張圖片所需要的時間。

	pynq 軟體	IP
Time(s)	39.0618462562561	0.13218092918395996
Accuracy(%)	96.9	96.8

加速 295.51 倍。

IP計算1000張圖片所需要的時間： 12.220770597457886(s)

```
start_time=time.time()
hit=0
for i in range(1000):
    img_buffer[:]=img_lib[i]
    ans=lable_lib[i]
    my_ip.register_map.CTRL.AP_START=1
    while(ol.example_0.register_map.CTRL.AP_DONE==0):
        pass
    if(ans==outcome_buffer.argmax()):
        hit=hit+1
    #print("i:",i+1,"acc",hit/1000)
end_time=time.time()
print(end_time-start_time)
print(hit/1000)
```

```
12.220770597457886
0.968
```

Figure 3 pynq outcome

結論

本次實驗成功將簡單神經網路以硬體方式實現，並且加速 295 倍。

接下來:

1. 繼續實驗使用 stream 傳輸資料格式
2. 開始嘗試將 AN code 加入神經網路做訓練
3. TCB 乘法代替位移以 C++方式硬體實現

Reference:

[1] https://www.boledu.org/textbooks/hls-textbook/io-interface/axi-master