Report\_dcs193 Final project 蔡東宏 110511277

1. **程式說明:**

這次的final project主要是做convolution，一開始先等filter\_valid拉起來，當filter\_valid拉起來時，讀取filter\_size, padding\_mode, acting\_mode, image\_mode以及filter的data，利用cntx, cnty去數現在存到第幾個位置，接下來等image\_valid拉起來時，讀取image的值，在讀取的同時判斷那個位置需不需要padding，要的話就將該位置進行padding(因為每次皆有歸0，所以不需要特別執行zero padding)，當讀了一定數量的值(可以計算convolution)時，便把compute\_vailid拉起來，當compute\_valid拉起來時，進行convolution的乘法，同時將compute\_valid傳給valid2，如此一來，下個cycle便可以進行加法，接著將加完的結果傳給下一個cycle進行activation function的處理，最後傳給下一個cycle判斷有沒有大於32767或有沒有小於-32768，同時將最後的結果輸出。

1. **架構圖:**

一張含有 文字, 圖表, 行, 繪圖 的圖片

自動產生的描述

一張含有 文字, 字型, 行, 數字 的圖片

自動產生的描述

(pipeline示意圖)

1. **優化過程:**

**A:共用乘法器和加法器:**一開始在做convolution(包括乘法和加法的時候)，我將filter是3\*3和5\*5的case分開討論，但後來發現這樣需要用到9+25個乘法器以及加法器，後來我試著將兩個case變成同一個case，我先將filter全部歸0，如此一來，便可將3\*3的case變得跟5\*5一樣，省下9個乘法器以及9個加法器的面積。

**B:減少存image的DFF:**一開始我使用12\*12個8bit的DFF去存image以及padding後的值(因為最大的情況是image\_size為8\*8且filter\_size是5\*5)，但後來我發現只需要7\*12個8bit的DFF，因為在算後面的時候，前面有些的image值已經用不到了，所以我可以當某一排的值已經用不到時，全部往上平移一格，如此一直重複，最多只需要用到7\*12的DFF去存image的值，如此可以大幅減少DFF的數量以及在做乘法的時候，二維陣列裡面的column不需要有變數去控制它，面積減少了許多。

一張含有 文字, 螢幕擷取畫面, 功能表 的圖片

自動產生的描述

**C:切pipeline:**我這次final project切pipeline的方式是我讓乘法一個cycle，加法一個cycle，過activation一個cycle，最後判斷有沒有超出邊界並輸出。如此切pipeline可以壓cycle time，讓一個cycle做的事情變少。

1. **結論:**

這次的final project主要為convolution，我嘗試了很多方法壓面積以及latency以及cycle time，透過共用硬體來減少面積，重複利用DFF來儲存image來減少DFF的數量，以及透過切pipeline來降低cycle time。