

DCS HW05 Report

109511207 電機 13 蔡宗儒

(一) 如何設計作業

切 Pipeline，順序如下：

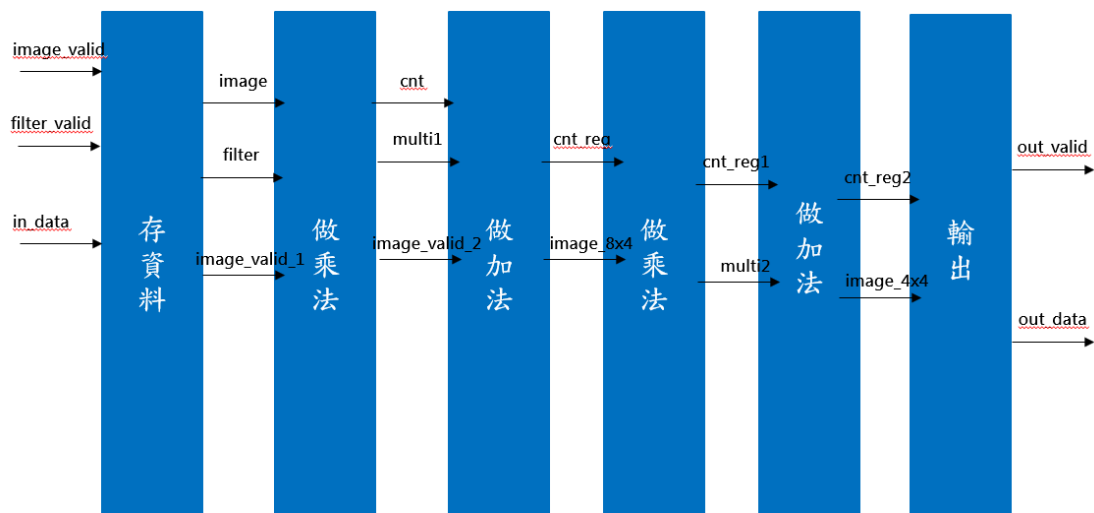
存資料 → 做乘法 → 做加法 → 做乘法 → 做加法 → 輸出

而我原先的切法是：

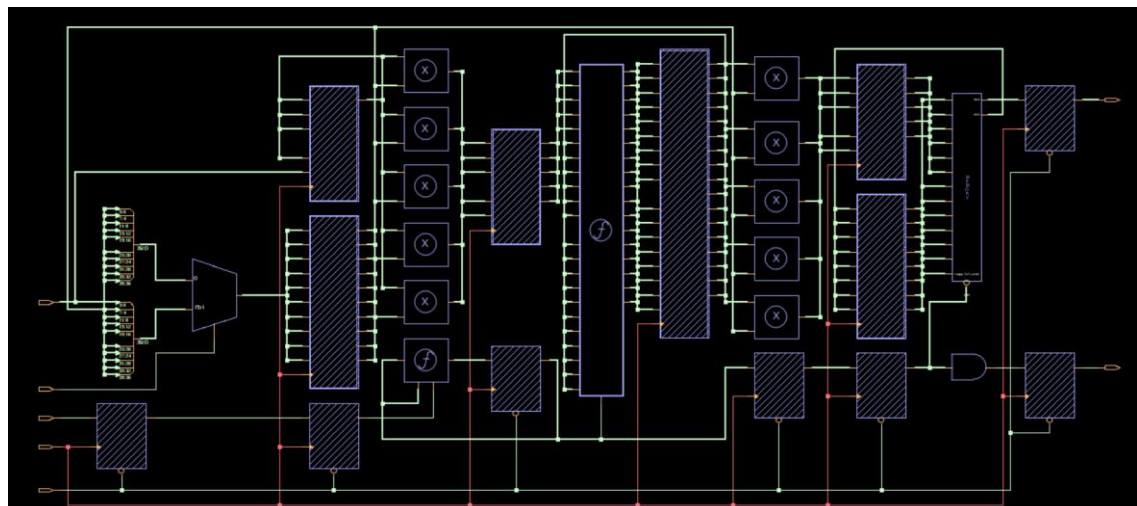
存資料 → 做 Convolution1 → 做 Convolution2 → 輸出

但用原本的切法的話，Clk Cycle 會壓不下去，因為 Convolution 裡面做的運算有乘有加，就會導致運算時間較長，Clk Cycle 就自然無法壓低。所以後來想想就把 Convolution 的乘法跟加法拆開來做，就能把 Clk Cycle 降下去，雖然這樣 Latency 會較大，且 Area 也會上升，但上升幅度都不多，反而 Clk Cycle 從 5.5ns 降到了 2.8ns，Performace 提升了將近一倍。

(二) 架構圖



附上 nSchema 圖



(三) 遇到的困難與解決方法

如何存那麼多資料

助教給的架構圖看起來要用很多 Array，但是仔細思考過發現其實根本不需要用那麼多空間來存資料。就拿第一層的 8×8 陣列來舉例，實際上每個 Cycle 中，Image 會需要用到的值都只有那五個要做 Convolution 的值，所以其實可以用 Shift Register 的概念。如下圖，每次把 Image 新的值推進去 image[4] 中，把舊的不需要的 image[0] 捨棄，其他四個值則都往 index 為 0 的方向上推。

```
always_ff @(posedge clk ) begin
    image[0] <= image[1];
    image[1] <= image[2];
    image[2] <= image[3];
    image[3] <= image[4];
    image[4] <= in_data;
end
```

以此類推，後面的 8×4 陣列可以只用 17 個位置存完， 4×4 陣列可以只用 8 個位置就存完。所以我這次作業用了大量的 Shift Register，這麼做除了好操作外也少了很多複雜的判斷條件，更不用寫很多很大的 Case 然後用 cnt 判斷，也因此省下了不少面積，可以說是一舉多得。

有趣的發現

做一連串的加法時用括號將他們隔開面積會有稍微的不同，不同的括法也會有不同的結果，推測應該是 EDA Tool 看到括號內的東西會優先做一些優化。如下圖

```
image_8x4_comb[16] = (multi1[0] + multi1[1]) + (multi1[2] + multi1[3]) + multi1[4];
```

(四) 心得

這次的作業感覺很像整個學期的總複習，用到了 Shift Register、Signed 的處理、寫 Pattern(這次 Pattern 比 Lab 難寫一點)和測值，花了我大概一整天來做，有比之前花的時間少一點，這可能意味著我有進步。這次跟上次一樣和同學交換了測值來互相測試我們的 Design 對不對，雖然測了有 16000 筆了，但還是沒有很心安，擔心會有一些沒想到的錯誤。

(五) Area

```
Combinational area:          35841.960109
Buf/Inv area:                2893.968102
Noncombinational area:      33370.445332
Macro/Black Box area:       0.000000
Net Interconnect area:      undefined (No wire load specified)

Total cell area:             69212.405440
Total area:                  undefined
```