Final Project Report

Student ID: 109060043 109062116 Name: 梁遠 潘勝元

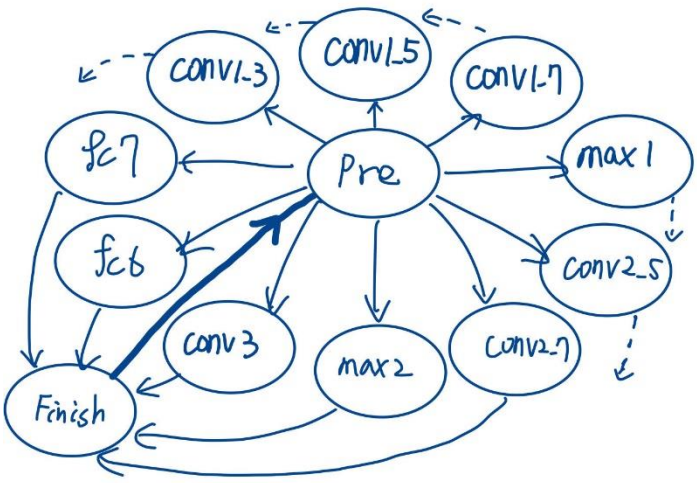
1. Design concept:

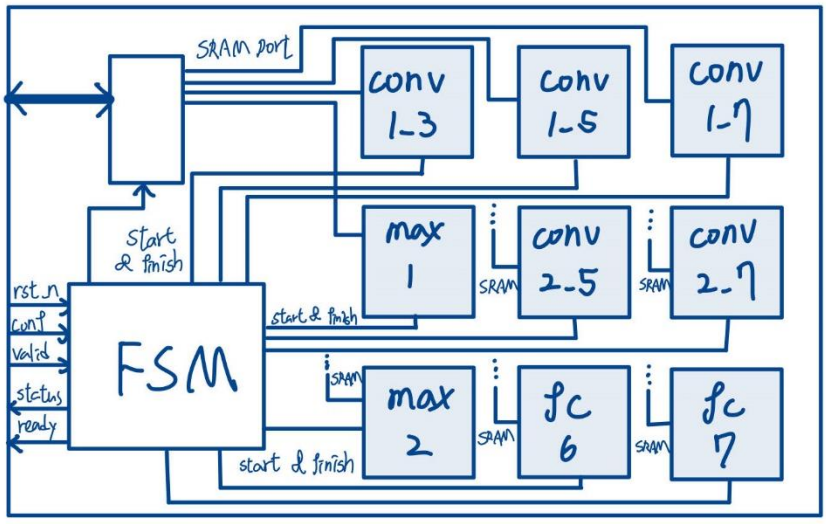
由於 spec 要求我們根據題意設計一個 reconfigurable accelerator, 因此我們的設計重點就是考慮更符合一般性的狀況，考量即使不依照六種測資的情形下資料的保障，並建立在符合上述要求上，盡可能提升 data reuse. 因此我們的設計有以下三個重點。

- 將 Convolutional 和 maxpool 分離完全獨立
  在 HW4 我們的設計都是卷積包含 maxpool 以利於輸出快速和減少 FF 的使用，但在 Final project 這樣設計的主要原因是因為想保證任何一個卷積層均能夠獨立執行，不受到隨後是否有 maxpool configugration 影響。
- 模型中所有 layer2 都會寫回 L2
  原先有考慮重複使用 hw4 的模組，將 pool 後的結果存在 L1，當 maxpool conf 輸入時，輸出 L1 存回 L2，但考量到如果存在與測資不同的情形，例如說 Conv1 後沒有 maxpool conf，純卷積的結果就不會被存回 L2，則以此加速器上層的處理器的角度來說，整個加速器就是 Black box, L1 無法被直接 access，就會遺失純卷積的結果，因此所有資料包含純卷積的結果都會被寫回 L2 SRAM。
- 額外的使用 activation sram.
  為了實踐上述均存回 L2 的設計，但又為了運算結束後符合 testbench goldan data 的資料排放，在 Conv1 和 Conv2 另外使用了 1024 到 1839 的 activation SRAM, 雖然更一般性的做法應該是寫回接續上層資料的位置，新 conf 訊號來之後再覆寫，但這會造成一些卷積運算和資料搬移變的相對複雜，最終就沒有這樣設計了。

○ RAE (top module)

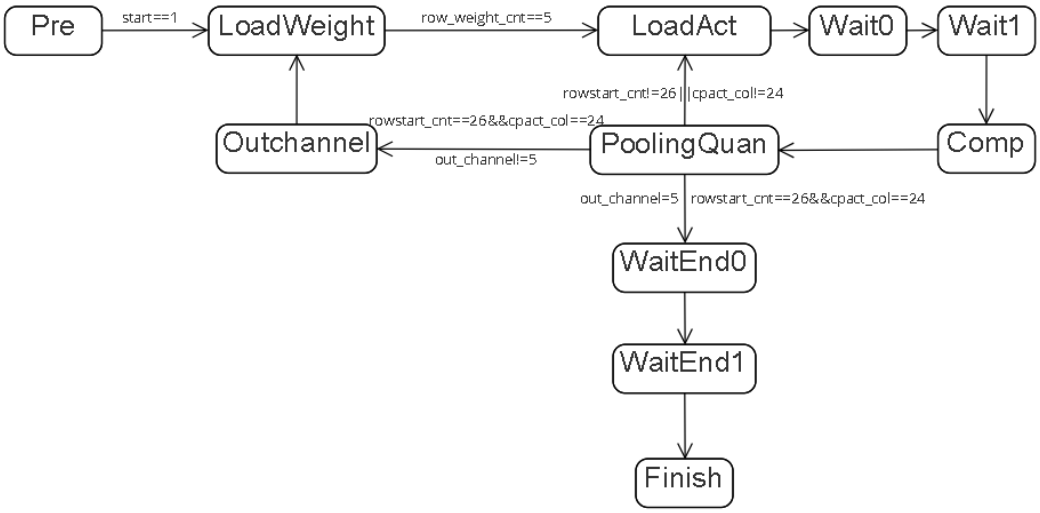
Initial state, 代表現在 RAE 是閒置的，正在等候 conf 訊號，當 conf 訊號來時，會切換至相對應的 state，然後將 SRAM 的存取 port (both weight and act)連接至相對應的模組和同時拉起該模組的 start 訊號。當模組的計算完成拉起其 finish 訊號時，跳至 finish state，將 status 拉成 finish，再重新回到 PRE。

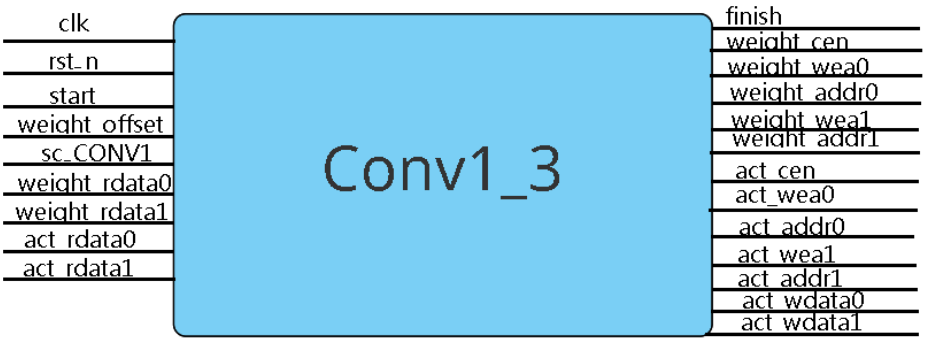




- Conv1 (filter=3)

Conv1_3 的設計架構是分成多個 state 分別去進行取得啟動訊號、從 sram 中取得 weight 和 activation 的值、進行運算、將運算結果輸出和結束訊號輸出這些動作。





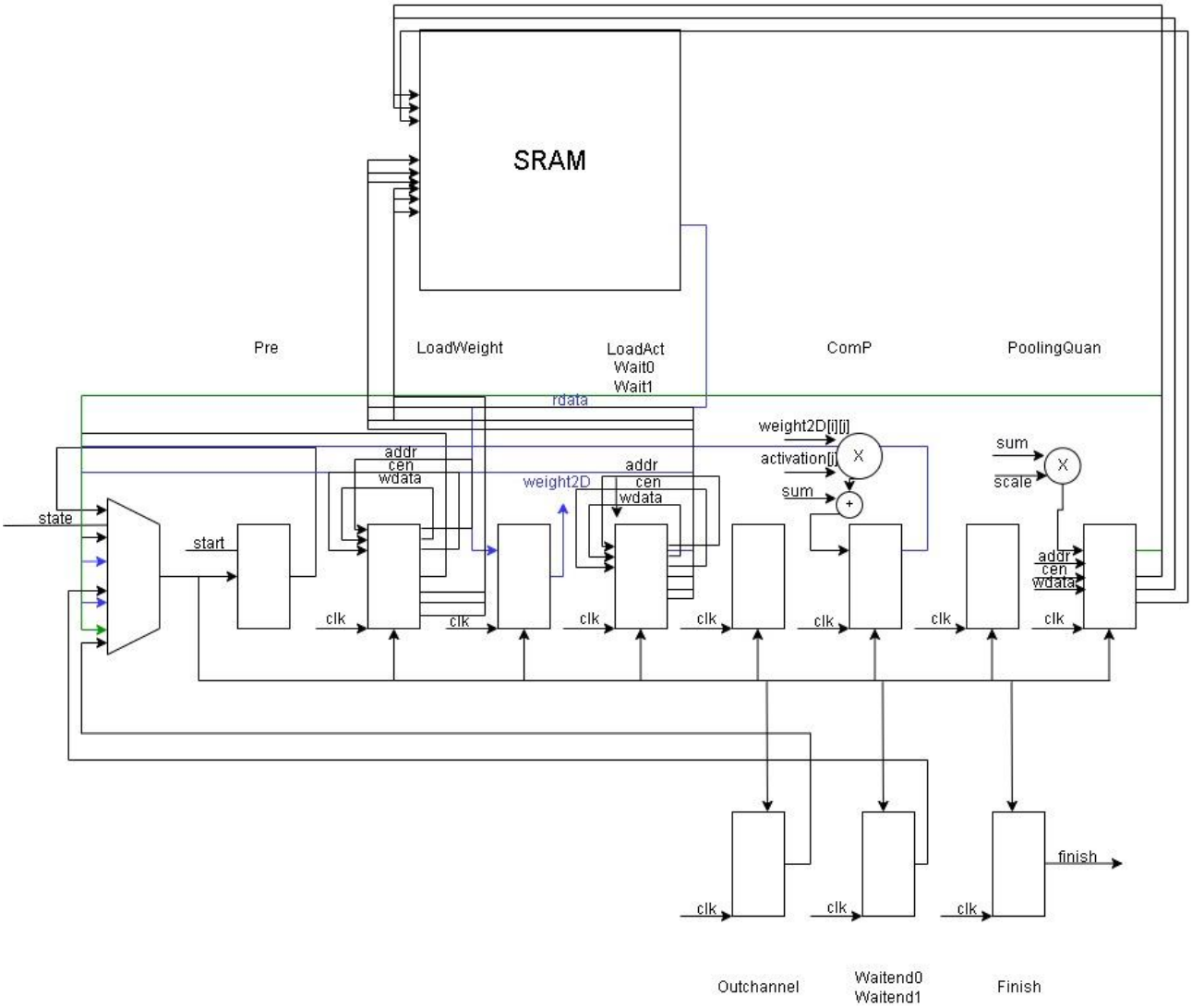
- Pre:
  負責接收要執行這個 module 的訊號並到下一個 state LoadWeight 開始執行。
- LoadWeight:
  每次將運算後要輸出的一個 channel 的量讀取進來，這個 module 是 3*3=9 筆資料，我們處理的方法是分兩個 cycle 要求把 4 個 address 的讀取資料，並在第一個 cycle 過 3 個 cycle 後開始讀取資料，但不使用最後一個 address 的資料。將資料都讀進 weight 後讓 state 進入 LoadAct。
- LoadAct、Wait0、Wait1:
  要求取得 activation 的值，我們取值的方法是 1 個 cycle 的 address 取同 1 row 的值並且計算，然後下 1 cycle 取下一個 row 的值並計算，直到計算出 2*4 的 output activation 後存到 sram address 1024 去再到下一個 column、row 開始計算直到計算完成，這 3 個 state 的目的是先要求要讀取各個 cycle 的值 LoadAct 傳至 Wait0，Wait0 傳至 Wait1，Wait1 傳至 ComP。

- Comp:
  執行以上的運算，在有 padding 時會在第一行時增加一個 activation 為 0 的 cycle 運算，最底層的部分也會將 activation 調為 0，左邊的 padding 處理方式是每次都計算 5 個值(包含 padding)，根據位置決定要輸出 padding 的位置還是要輸出前一個沒被存入的值並存取這次沒被存入的值，在計算出 2*4 的 output activation 時會傳至 PoolingQuan 這個 state。

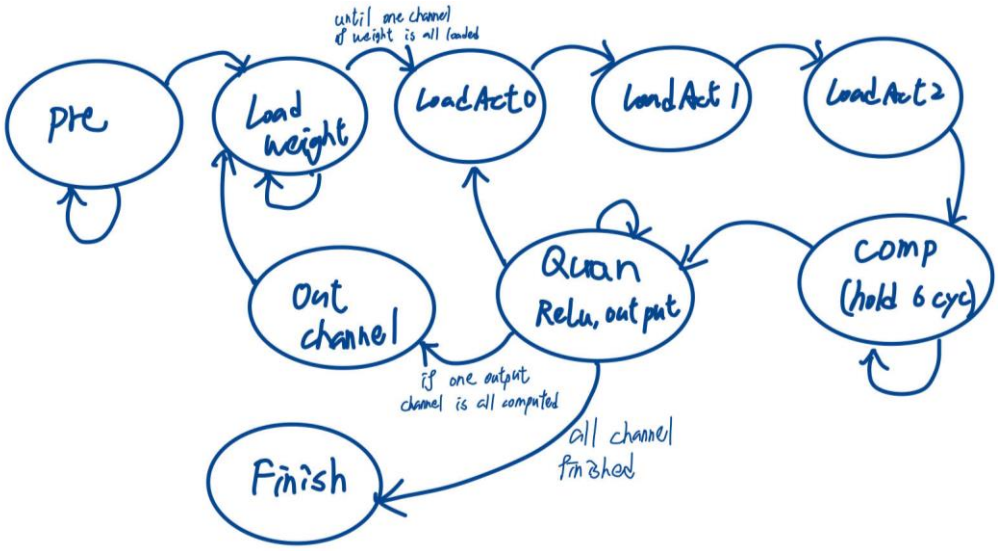
- PoolingQuan:
  將 2*4 的 activation 值進行 quantize 後輸出，如果這個 channel 輸出完成且有下一個 channel 就傳到 Outchannel 中，如果沒有下一個 channel 就傳至 Waitend0，如果這個 channel 還沒輸出完就傳到 LoadAct 繼續取值計算。
- Outchannel:
  將 channel 的值加 1 代表下一個 channel 並進入 LoadWeight。
- Waitend0、Waitend1:
  等待輸出 sram 的值輸出完成並傳給 Finish。
- Finish:
  表示計算完成並用 finish 表示。

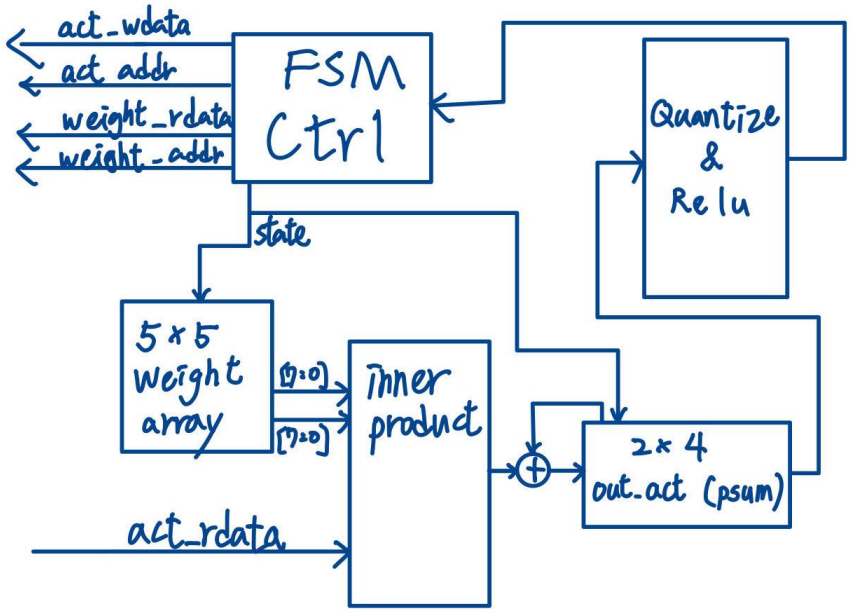
Analysis:

因為沒有使用 L1 buffer 所以會多花 cycle 去讀取資料。為了計算容易所以會放棄計算只有部分資料的區塊只計算完整的，所以會重複讀取資料多花 cycle。



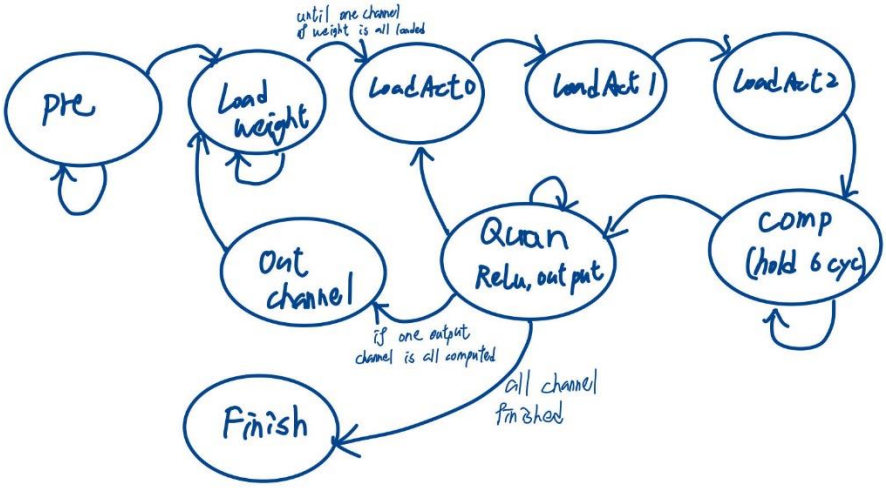
- Conv1 (filter=5)

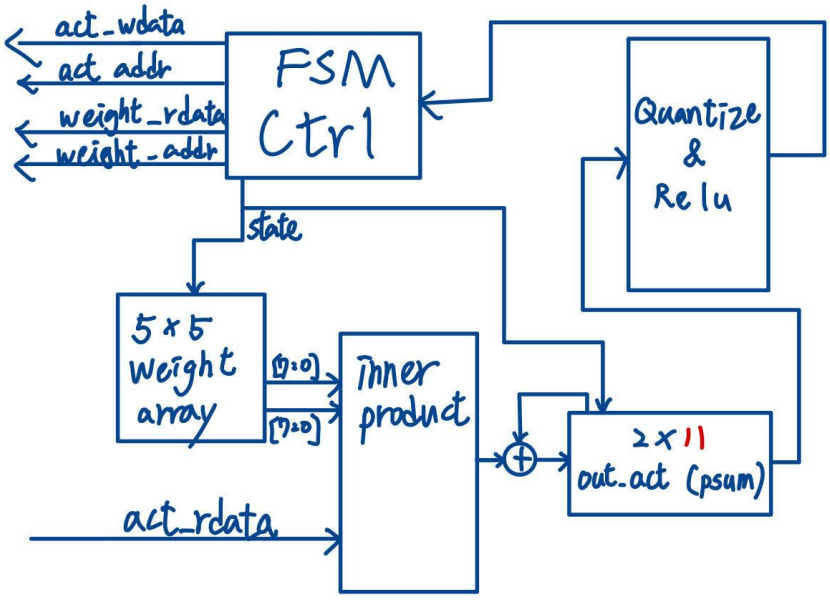




- Conv1 (filter=7)

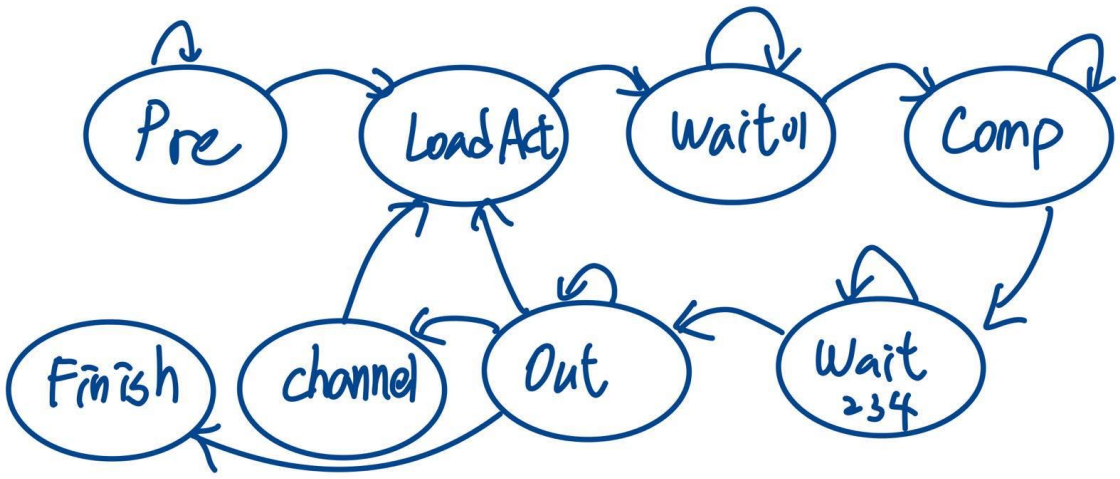
2*11 的 reg array 是基於每次計算 4 個 word 的 input act, 加兩側 padding，共 18*8bit，得出 12*8bit 的輸出，但不會兩側同時 padding，因此只需 11 個 reg 儲存 psum 即可。

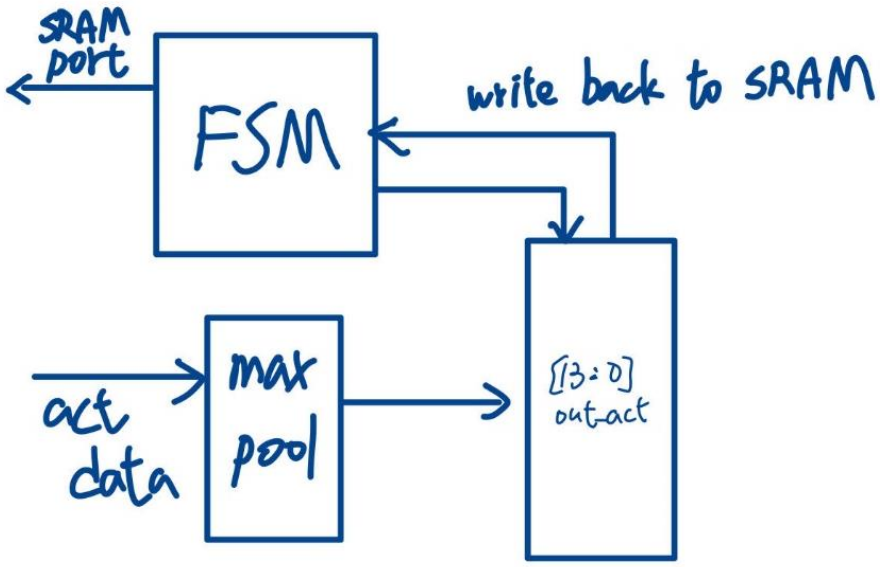




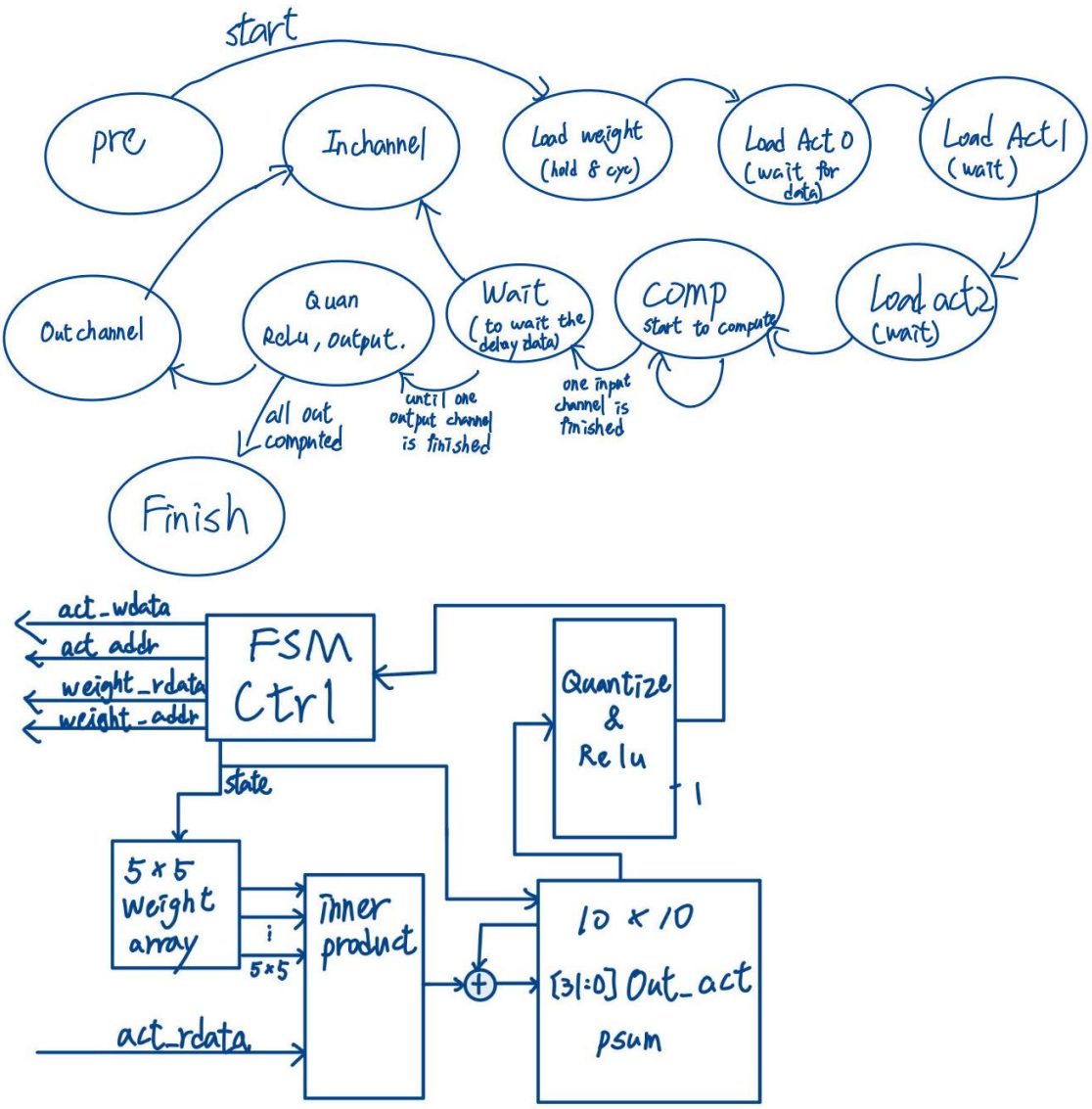
- Max1

  將存在 1024 以後的 conv1 output maxpool 後存回 image 後面(256or196)



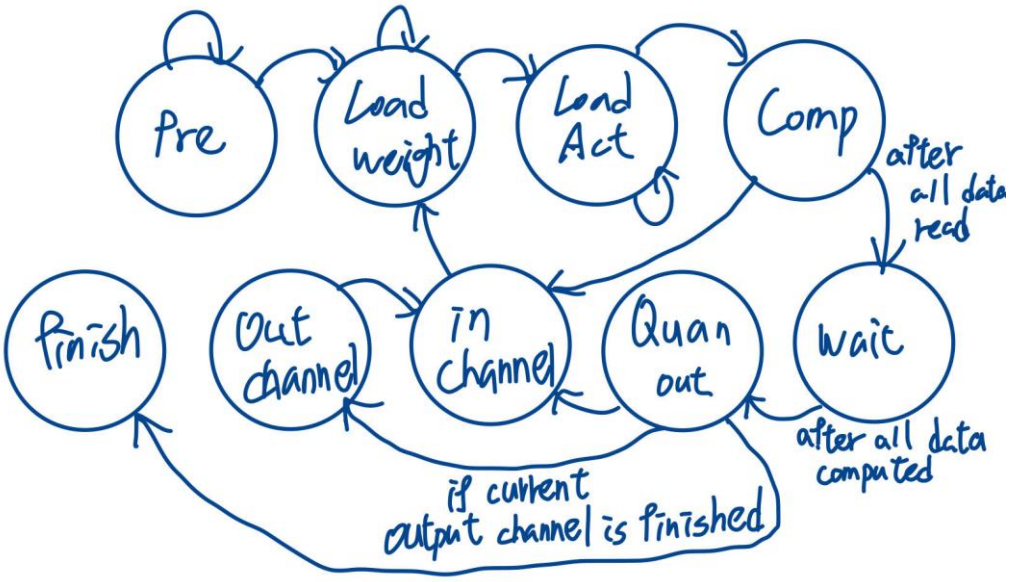


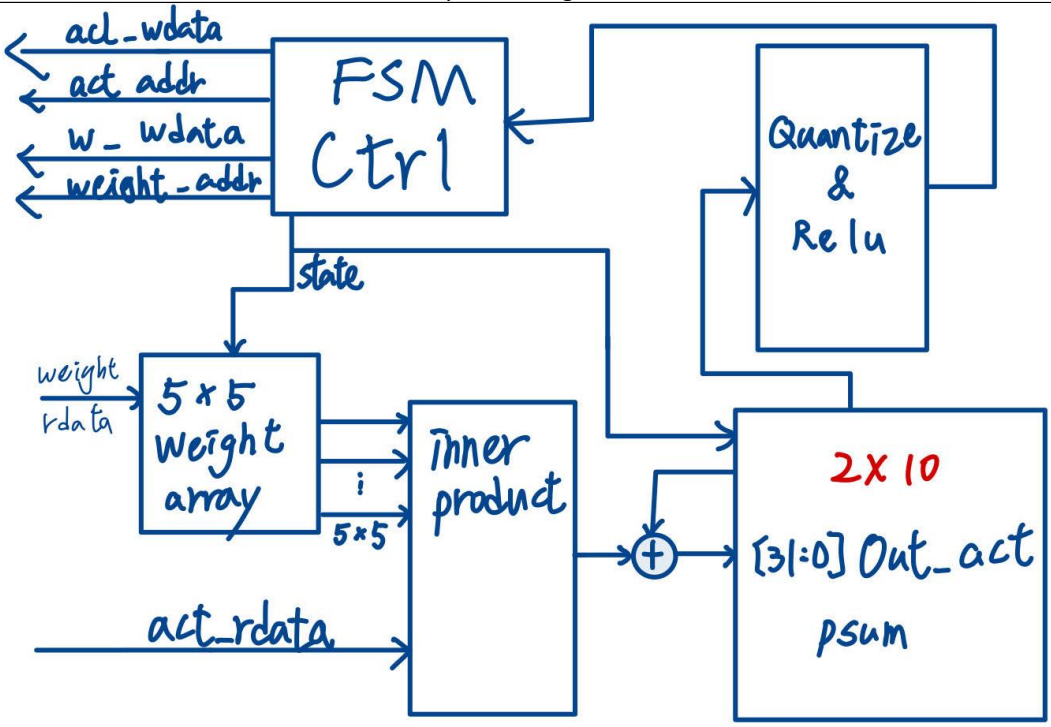
- Conv2 (filter=5)



- Conv2 (filter=7)

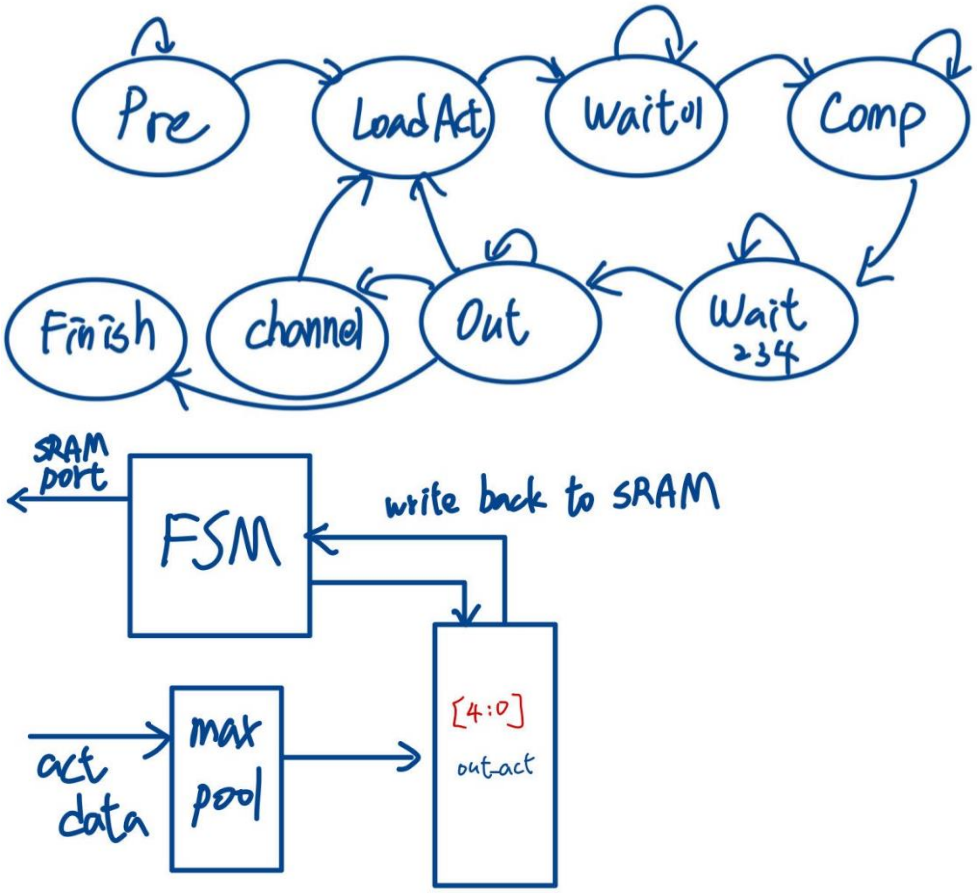
每次循環計算完 2*10(即兩個 row)後輸出，不同於 conv25，計算整個 output channel 才輸出，以節省 reg 儲存 psum 的需求。





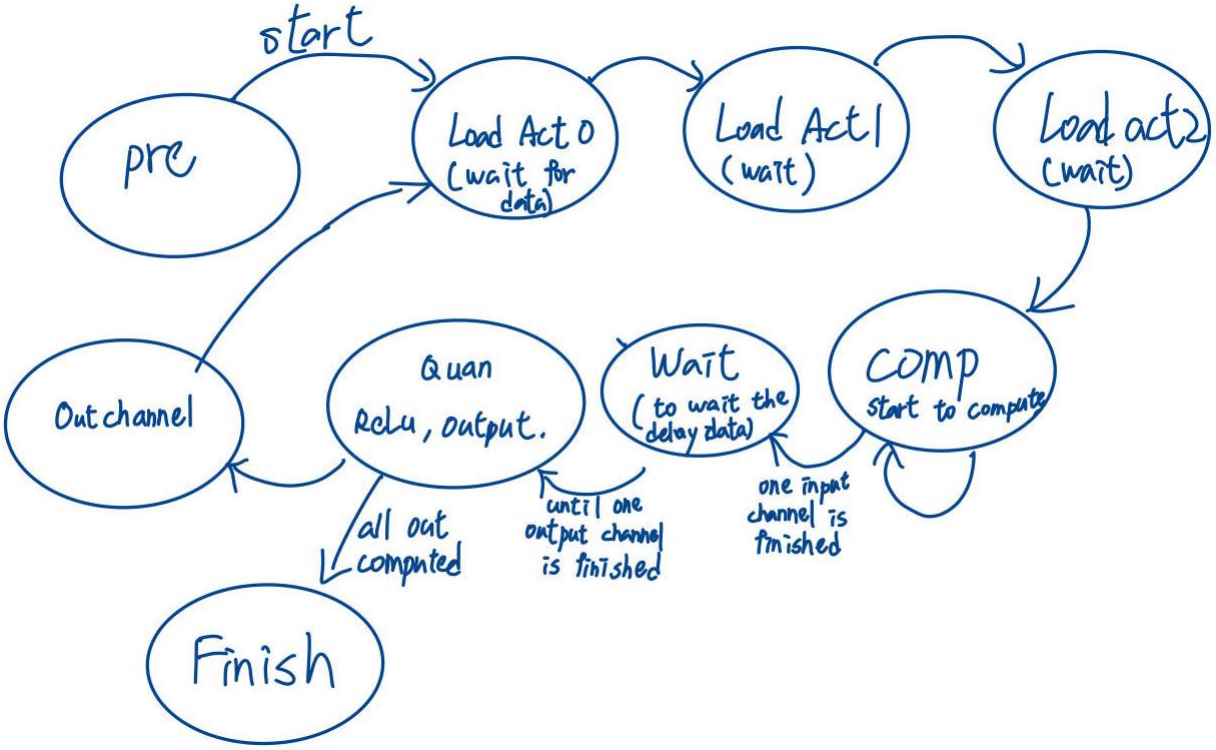
- Max2

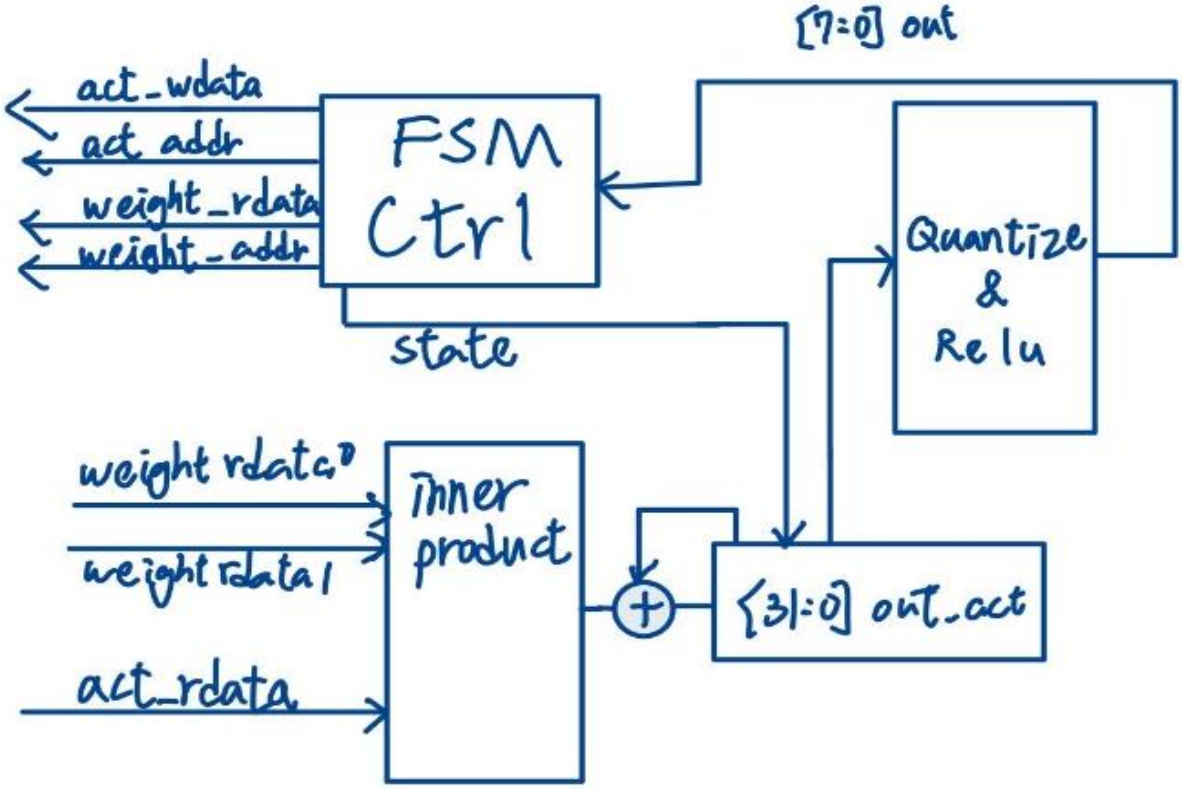
  設計與 max1 相近，但針對的是 conv2 output 的 data arrangement。



- Conv3

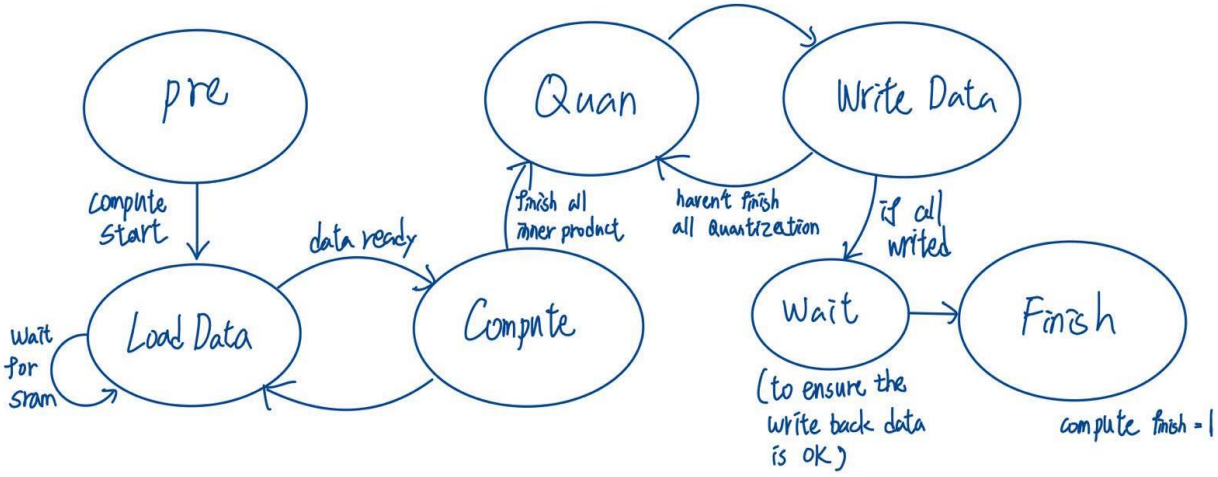
第三層卷積層的運算因為 ifmap 和 weight filter 大小相同，因此可以被理解成 120 個 activation 和 weight i (0<=i<120)的內積，內積的兩向量長為 400 (=16*5*5)，因此設計就可以大幅簡化。

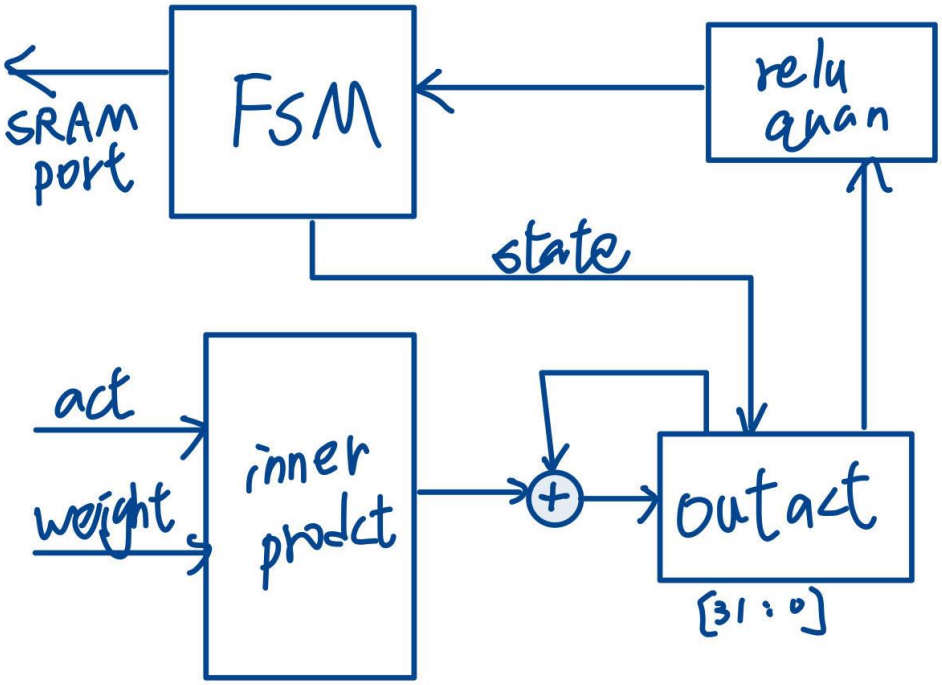




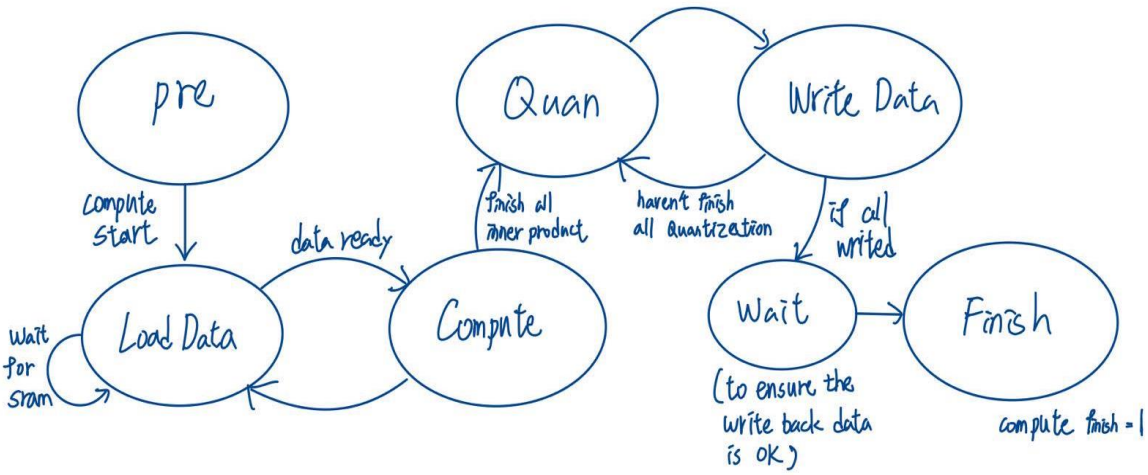
- Fc6

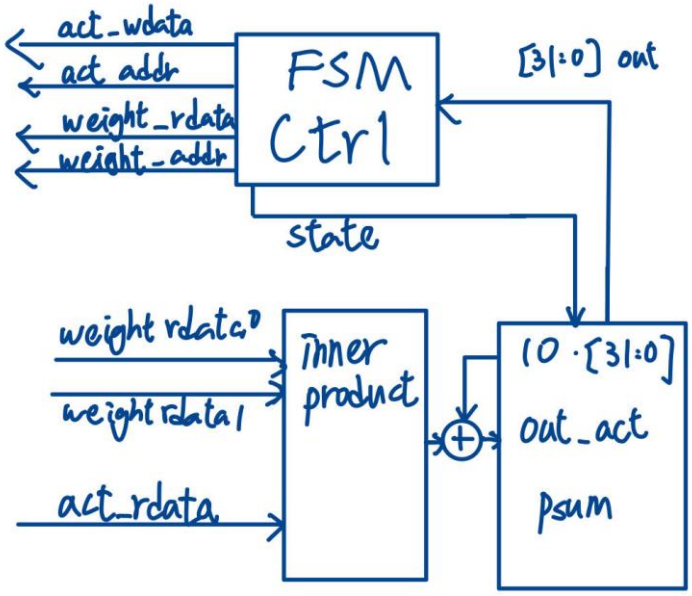
  每次僅計算與輸出一個 out act，以利於節省 reg 使用。





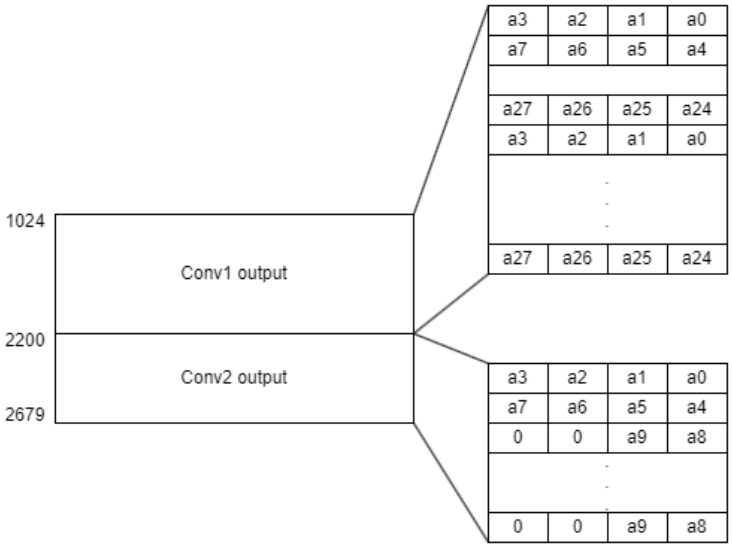
- Fc7





2. Data arrangement

在這裡對於根據 spec 所要求的資料排放不多做贅述，僅描述另外使用的 activation (1024 以後)的部分。



3. Result

	Lenet0	Lenet1	Lenet2	Lenet3	Lenet4	Lenet5	Average
RTL simulation	PASS	PASS	PASS	PASS	PASS	PASS	
Gate-level simulation	PASS	PASS	PASS	PASS	PASS	PASS	
Cycles	21552	21157	30288	29892	20364	29100	**25392.2**
L2 activation access	8187	8199	10779	10791	8151	10743	**9475**
L2 weight access	17044	18400	22212	23568	17632	22800	**20276**
Gate-level sim clock period	4.1 ns						
Total cell area	837582 μm^2						

Lenet0

```
Reset System
Computation is finished, start validating result...
====================
Image             [PASS]
Conv 1 activation [PASS]
Conv 2 activation [PASS]
Conv 3 activation [PASS]
FC   1 activation [PASS]
FC   2 activation [PASS]
>>> Congratulation! All result are correct
  [Pre-layout gate-level simulation]
Clock Period: 4.10 ns,Total cycle count:       21552 cycles
L2 Weight Buffer Access times:                  8187
L2 Actication Buffer Access times:             17044
====================
Simulation finish
```

Lenet1

```
Reset System
Computation is finished, start validating result...
====================
Image             [PASS]
Conv 1 activation [PASS]
Conv 2 activation [PASS]
Conv 3 activation [PASS]
FC   1 activation [PASS]
FC   2 activation [PASS]
>>> Congratulation! All result are correct
  [Pre-layout gate-level simulation]
Clock Period: 4.10 ns,Total cycle count:       21156 cycles
L2 Weight Buffer Access times:                  8199
L2 Actication Buffer Access times:             18400
====================
Simulation finish
```

Lenet2

```
Reset System
Computation is finished, start validating result...
====================
Image             [PASS]
Conv 1 activation [PASS]
Conv 2 activation [PASS]
Conv 3 activation [PASS]
FC   1 activation [PASS]
FC   2 activation [PASS]
>>> Congratulation! All result are correct
  [Pre-layout gate-level simulation]
Clock Period: 4.10 ns,Total cycle count:       30288 cycles
L2 Weight Buffer Access times:                 11258
L2 Actication Buffer Access times:             22227
====================
Simulation finish
```

Lenet3

```
Reset System
Computation is finished, start validating result...
====================
Image             [PASS]
Conv 1 activation [PASS]
Conv 2 activation [PASS]
Conv 3 activation [PASS]
FC   1 activation [PASS]
FC   2 activation [PASS]
>>> Congratulation! All result are correct
  [Pre-layout gate-level simulation]
Clock Period: 4.10 ns,Total cycle count:       29892 cycles
L2 Weight Buffer Access times:                 11270
L2 Actication Buffer Access times:             23583
====================
Simulation finish
```

Lenet4

```
Reset System
Computation is finished, start validating result...
====================
Image             [PASS]
Conv 1 activation [PASS]
Conv 2 activation [PASS]
Conv 3 activation [PASS]
FC   1 activation [PASS]
FC   2 activation [PASS]
>>> Congratulation! All result are correct
  [Pre-layout gate-level simulation]
Clock Period: 4.10 ns,Total cycle count:       20364 cycles
L2 Weight Buffer Access times:                  8151
L2 Actication Buffer Access times:             17632
====================
Simulation finish
```

Lenet5

```
Reset System
Computation is finished, start validating result...
====================
Image             [PASS]
Conv 1 activation [PASS]
Conv 2 activation [PASS]
Conv 3 activation [PASS]
FC   1 activation [PASS]
FC   2 activation [PASS]
>>> Congratulation! All result are correct
  [Pre-layout gate-level simulation]
Clock Period: 4.10 ns,Total cycle count:       29100 cycles
L2 Weight Buffer Access times:                 11222
L2 Actication Buffer Access times:             22815
====================
Simulation finish
```

4. Others (optional)

另外補充一下，我們的設計因為為了滿足前後有 buffer 的設計要求，有改動部分 testbench，以正確計算總 cycle count，testbench 有提供於附件，謝謝助教，麻煩你們了。