

Homework 4 Report

Student ID: 112061533

Name: 潘金生

1. Design concept:

- Explanation of the overall hardware architecture and Block diagram.

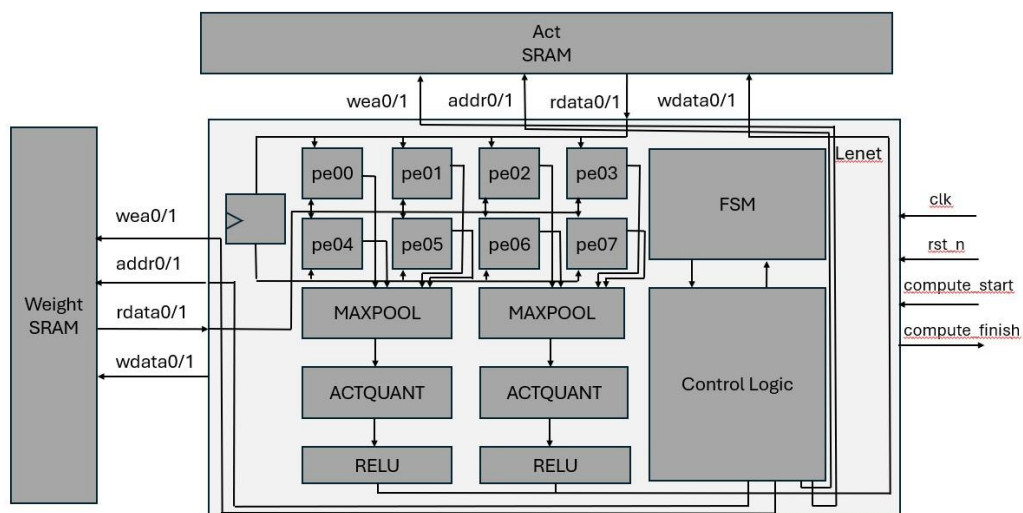
附圖為本次的 block diagram，這次主要分了五個 module 去撰寫，其中包括 top module Lenet 和 submodule pe, actquant, maxpool, relu，其中 Lenet 負責整合 submodule 和做訊號傳接，延遲訊號等處裡。

在本次作業中，採用了 8 個 PE(40 個 MAC)去實現 Lenet，這是為了在 convolution layer 時達到 data reusability 的效果，如同 spec 所說，可以一次 load 6*6 ifmap 去實現，讓 6 個 cycle 即可達到 8 筆 output 且經過 maxpool 後一次產生 2 筆 result，不過在 Fully-connected layer 時則只改成開 2 個 pe 去做運算。

除此之外，由於 scale 為正數，因此在本次作業中，我先進行 maxpool，在做 actquant, relu，由於 maxpool 後只會產生 2 筆 output，這樣一來，我只需要開 2 個 actquant 和 relu 就好，可以有效地節省面積。

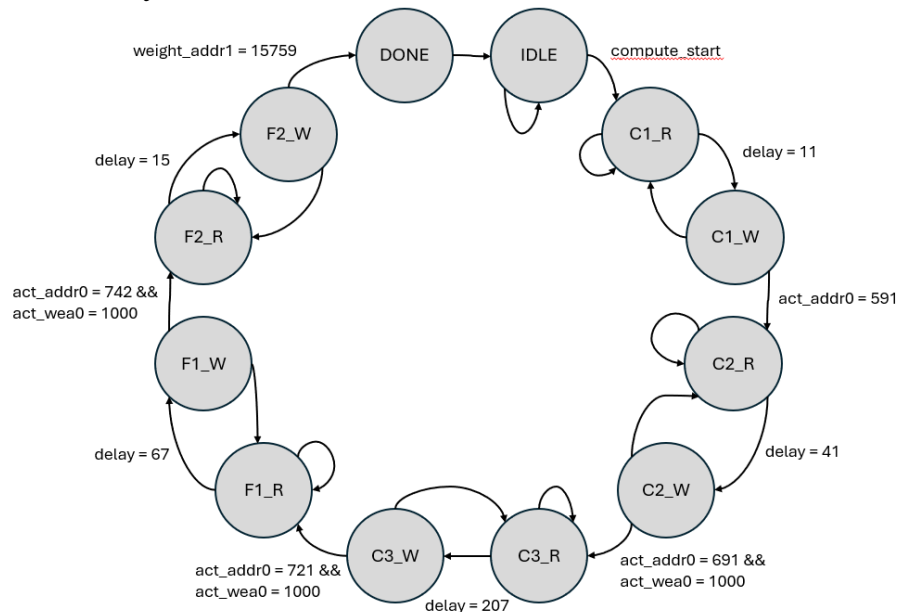
在本次作業中，clock period 也算在評分範圍內，因此在 pe, actquant 中，我認為 path 較長的 delay 我皆有切 pipeline，以達到較好的 clock period。

- Module pe: 乘加運算器，用來執行 weight 和 activation 的乘加運算，共有 5 個乘法和 6 個加法器，且中間有切 2 個 cycle。
- Module Maxpool: 比較 4 筆 input data 的大小，產生 1 筆最大的輸出，中間有切一個 cycle。
- Module Actquant: 用來乘上 scale 和進行 shift 16 bits 且 clamp 大小為(-128,127)中間範圍，且中間有切 1 個 cycle。
- Module Relu: 小於 0 的數值歸 0，大於 0 保持原值。
- Lenet: 用來控制 address 和 wea 參數，判斷合適需要讀入數值，和寫回輸出，並且其中的 FSM 來判斷目前為什麼 state(layer)，且要做什麼事。



- Finite-state machine diagram.

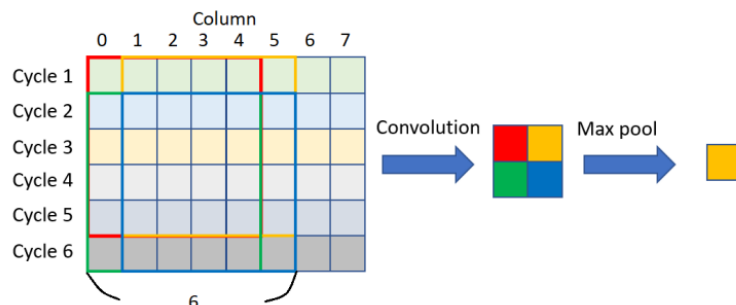
FSM 設計的部分，基本上就是判斷地址，這裡的 delay 是代表在讀取時所需要的 cycle，公式如下：讀取的 data 數 + 等待 2 個 clk(sram 讀取 data 會有延遲)+pipeline cycle 數，其中，在 Fully-connected layer 時，由於我是一次寫四筆，因此他需要等待的 delay 較長，並且在 delay 中，還需留一個 cycle 讓 pe 歸 0 重新運算，因此 fully-connected layer 的公式為：讀取的 data 數+ 3*等待 clk + 等待 2 個 clk + pipeline cycle 數，其餘層_R(讀取層)以此類推，接著再判斷何時要跳入_W(寫出層)，並且判斷 addr 到何時需要跳入下一個 layer。



- Explanation of dataflow (and data reuse) for convolution and fully connected layers.

Convolution Layer:

- Spatial reuse:** Activation 採用了 spatial reuse，一次可以讀取 8 筆資料，並將他們給全部的 pe 所使用。
- Temporal reuse:** 由於在讀取 weight 的時候，無法給下一排所使用，因此這裡採用 weight buffer 的方式，在下一個 clk 的時候，給下排使用。



Fully-connected Layer:

在 Fully-Connected Layer 時，採用一次寫四筆的輸入，其中會 delay1 個 clk，最後在一次四筆輸出過去，這樣可以避免一次 write 時需要等待的 cycle，節省所要花費的 clk 數。

2. Spyglass Report:

a. Could not find clocks for all the flops

這部分要加入 SGDC constraint，來讓 Design Compiler 找到 clocks，不過我有去檢查波型，發現我所設計 FF 有正常運作，因此在這裡忽略它。

```

MOROSIMPLE REPORT:
##### FATAL MESSAGES #####
ID      Rule      Alias      Severity  File      Line  Wt      Message
-----
[2]     Av init01          Fatal      ../hdl/lenet.v  1    10    Could not find clocks for all the flops. Please add clock SGDC constraint to the design

```

b. Advanced Lint Policy not run due to unavailability of auto verify license feature:

軟體廠商授予與我們的 license 限制，可以忽略。

```

MOROSIMPLE REPORT:
##### FATAL MESSAGES #####
ID      Rule      Alias      Severity  File      Line  Wt      Message
-----
[0]     Av license01        FATAL      N.A.      0    2    'Advanced Lint Policy' not run due to unavailability of Auto Verify license feature

```

3. Result

本次作業採用 clock period 1.75ns 去合成跑在 1.98ns 的 gate-level simulation 來避免 setup/hold time 問題。

Item	Description	Unit
RTL simulation	PASS	---
Gate-level simulation	PASS	---
Gate-level simulation clock period	1.98	ns
Gate-level simulation latency	25857	cycles
Total cell area	48599.688779	μm^2

4. Others (optional)

謝謝助教在討論區的積極回覆，幫助我解決很多問題。