CMPEN 417 - FPGA Design
Spring 2021
Lab 3
DUE: Monday, March 15th, 11:59am, on Canvas.

Memory Interface / 2D Convolution and Average Pooling w/ Clamp Function

Description

In this lab you will use all of the topics we've talked about in lecture thus far. The groups will be split into two assignments (each part is approximately equal difficulty). Groups that are odd numbered (as per the group sign up sheet will implement Part A of this lab. Groups that are even numbered will implement Part B. Additionally Part A modules are meant to communicate with Part B modules. Each odd group will collaborate with the group that is one higher in number from their own group (e.g. group 1 will work with group 2). Each even group will work with the group number one below their own.

NOTE: EVEN THOUGH YOUR GROUP WILL ONLY IMPLEMENT ONE PART OF THIS LAB READ BOTH PARTS TO BETTER UNDERSTAND THE TASK.

Background:

https://www.paperswithcode.com/method/average-pooling
https://towardsdatascience.com/intuitively-understanding-convolutions-for-deep-learning-1f6f42faee1

Part A

For part A of the lab you will implement a 2D filter on a 64 x 64 8-bit (grayscale) image. The filter size will be 3x3 with a stride of 2 in both the X and Y directions. Do not worry about border handling. This module will have to accept weights (in a similar fashion to lab 2). Data from the computation will come from reading out of a provided memory module. The output after the computation will be passed to a Part B module (in a similar fashion to Lab 2). Module definition for Part A will be as follows.

```
module part_a
(
    input clk,
    input rst,

    output [12:0] read_address,
    input [7:0] read_data,
    output read_data_ready,
    input read_data_valid,

    input [7:0] weight_data,
```

```
    input [3:0] weight_idx,
    input weight_valid,
    output weight_ready,

    output [7:0] out_data,
    output out_data_valid,
    input out_data_ready
);
```

Part B

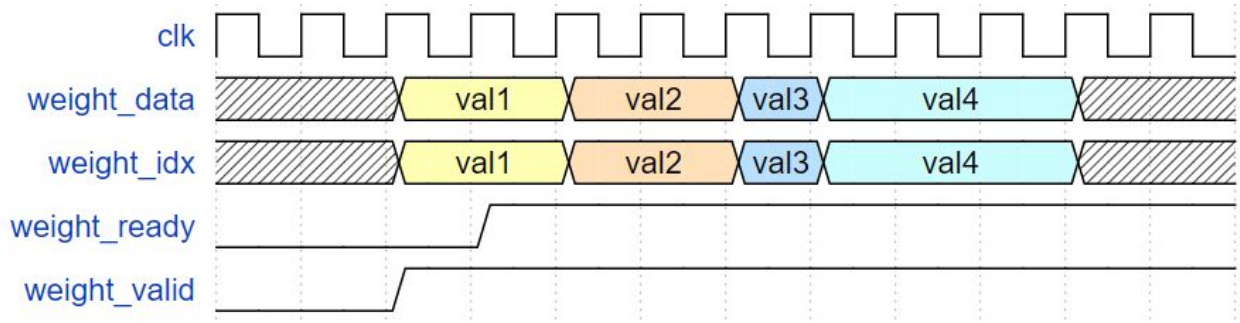
For part B of this lab you will implement an average pooling function on a 32x32 8-bit image with a window size of 2x2. Additionally from the output of this pooling calculation you will also implement a clamping filter with a configurable high and low clamp. Part B module definition provided below. For input you will be receiving input from the part A module similar to the way input was transmitted in Lab 2.

```
module part_b
(
    input clk,
    input rst,

    output [12:0] write_address,
    output [7:0] write_data,
    input write_data_ready,
    output write_data_valid,

    input [7:0] clamp_low,
    input [7:0] clamp_high,
    input clamp_valid,
    output clamp_ready,

    input [7:0] in_data,
    input in_data_valid,
    output in_data_ready
);
```

Functional Specification:

Part A Weights

Weights will be loaded through the weight_data and weight_idx port with the value on weight_idx indicating which weight it represents (0-6). Both weight_data and weight_idx will signal on the weight ready/valid signals.



Input data will be a constant stream of data that comes in signaled using standard ready/valid signals. Output data will also be signaled on ready/valid.

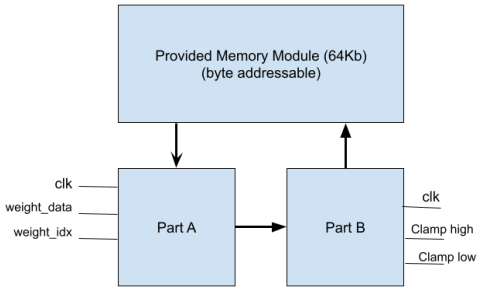
When reset is high the entire pipeline should clear (set to 0).

Part B Clamp (High/Low)

Clamp low and clamp high values will be loaded simultaneously at the corresponding ready/valid signals. On reset clamp low should be 0 and clamp high should be 255.

Block Diagram

***note: ready/valid/reset and other sideband signal are not shown**



Submission Requirements:

1. **Submit the block diagram for all of your modules and any submodules you create. (single pdf document)**
2. **Show a waveform of modules part a and part b communicating.**
3. **Submit all verilog files.**