CMPEN 417 - FPGA Design
Spring 2021
Lab 2
DUE: Monday, March 1st, 11:59pm, on Canvas.

FIR Filter / 1D Convolution/Cross-Correlation

Description

In this lab you will use all of the topics we've talked about in lecture thus far. You will create a FIR filter module. The FIR filter will perform an 7 element filter on an input stream of data. The top-level module definition is provided below. You will create two versions of this module. The first will not be pipelined. The second will be inserting 3 pipeline stages into your design. You will then generate the timing report for each of these designs to show/writeup the differences and trade offs between the designs.

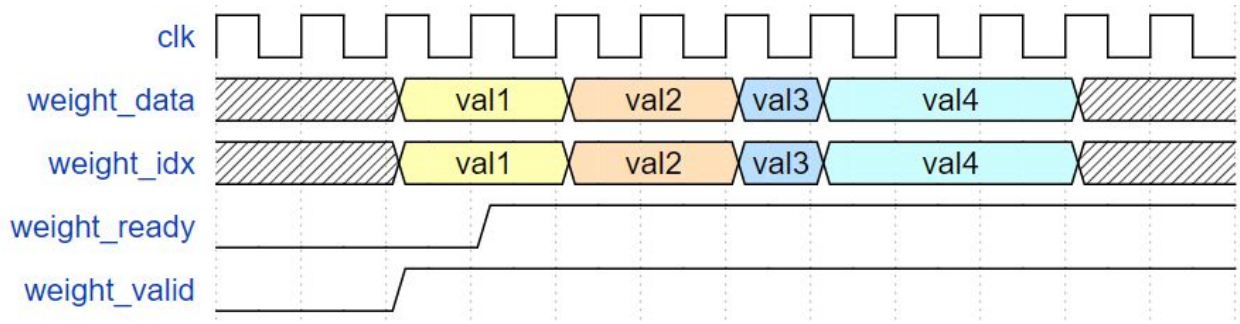
Extra Credit:

For 15 additional points you may also fully pipeline your design and show/writeup the differences between the other two designs as well.

Other Background (may be useful):
https://en.wikipedia.org/wiki/Finite_impulse_response

Functional Specification:

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).

Submission Requirements:

1. **Submit the block diagram for all of your modules and any submodules you create. (single pdf document)**
2. **Write-up of the differences between your pipelined version and non-pipelined version with screenshots inserted showing the timing for each of the designs (also as a pdf). As part of this writeup please include an "analysis" of the design (such as cycle latency, throughput etc. as well as which design you would choose under various (basic) design constraints and design goals (power, resource, speed, etc.)**

```
module fir_7_7
(
    input clk,
    input rst,
    input [7:0] weight_data,
    input [2:0] weight_idx,
    input weight_valid,
    output weight_ready,
    input [7:0] input_data,
    input input_valid,
    output input_ready,
    output [15:0] output_data,
    input output_ready,
    output output_valid
);


endmodule
```