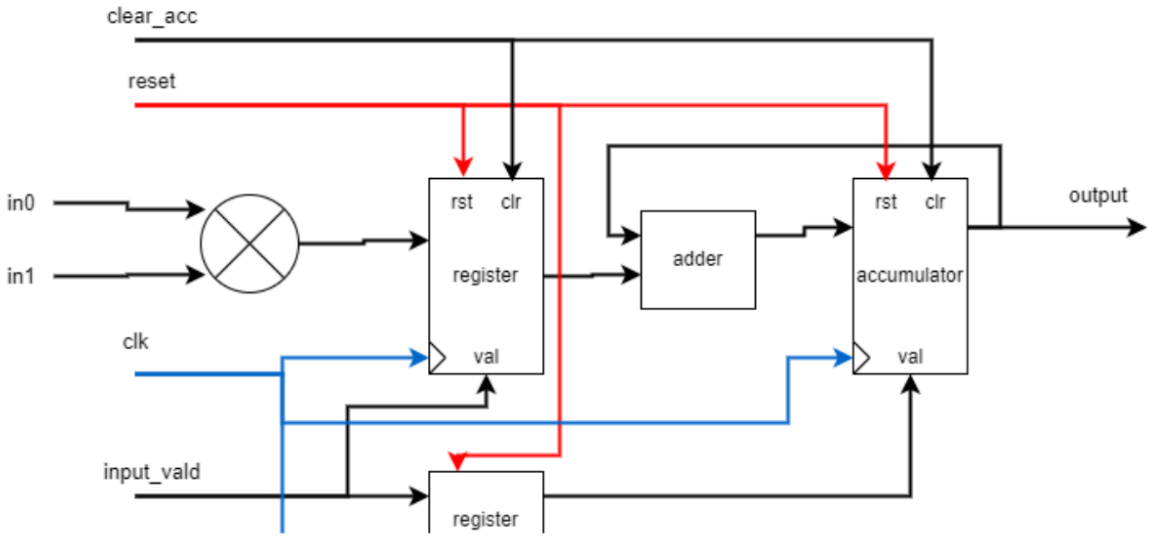
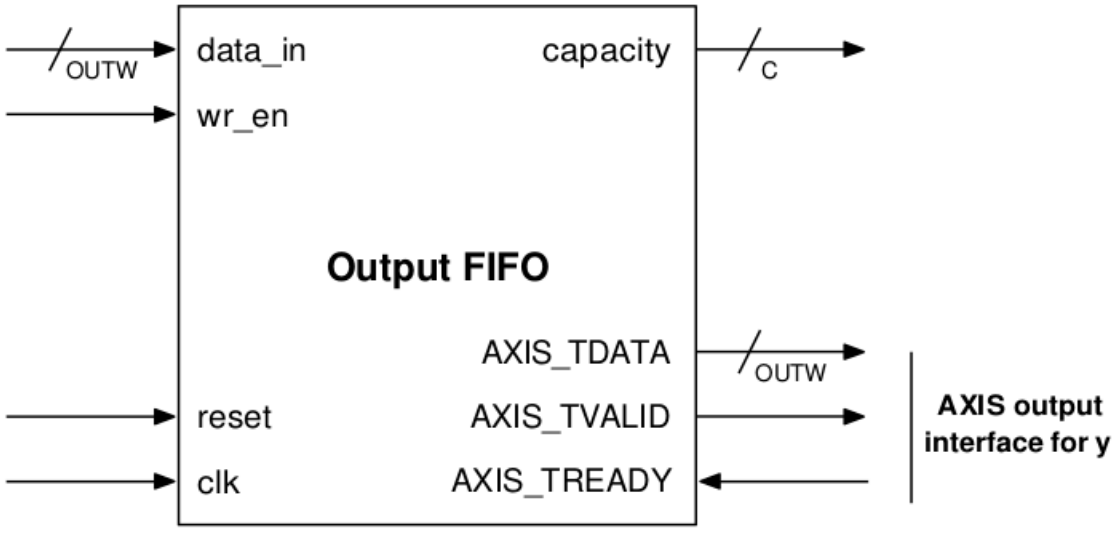
Assignment 5: Capstone Project

Project Part1: MAC



The RTL code for the multiplier and accumulator unit is provided in the file named “MAC.sv”.

Project Part 2: Output FIFO



The RTL code for this block is provided in the file named “Output_FIFO.sv”.

1. The basic form of the FIFO was discussed in slides, but here you needed to adapt that to
interface its output with AXI-Stream. Explain how you did that and how your logic works.

The AXI signals used in the FIFO are DATA,VALID and READY. The output AXIS_TVALID is the complement of the status signal “empty”. This means if the FIFO is not empty, it has valid data.

The AXIS_TREADY is an input signal and will be driven from the testbench.

The AXIS_TDATA is the output data of the FIFO.

Similarly the capacity logic is implemented which shows the current empty locations of the FIFO.

Project Part 3: Input Memory Module

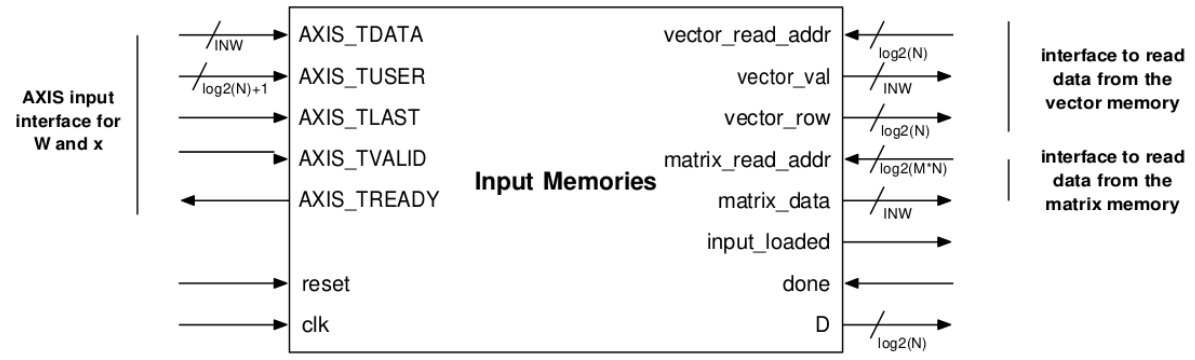
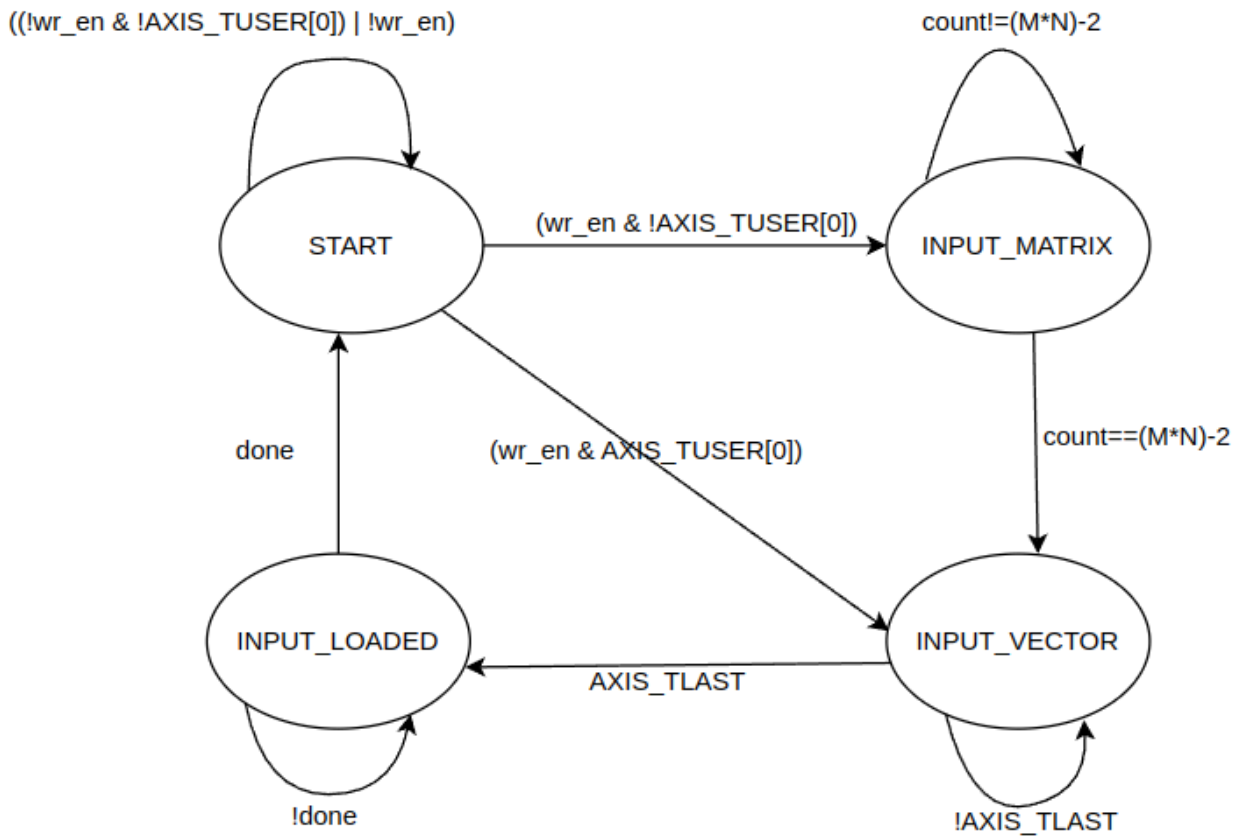


Figure 3.1. Input Memory module.

1. This part of the project required you to design a significant amount of control logic that
interacts with the AXI-Stream interface, the memories, the D register, and the input_ready and done signals. Carefully and thoroughly document this module including your control logic.

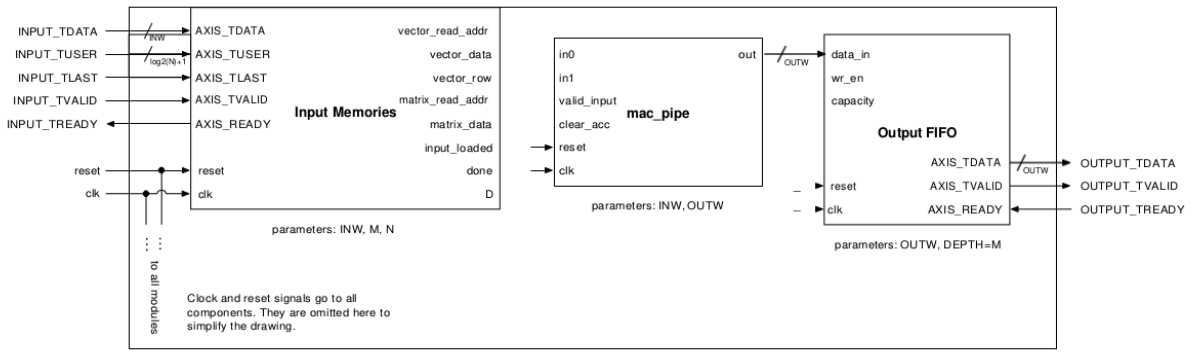


Initially we are at the start state. Then depending upon the inputs we go to either the INPUT_MATRIX state or INPUT_VECTOR state. Note that we go to the INPUT_VECTOR_STATE only when we want to reuse the already stored matrix. In these states we store the matrix and vector in the memories. When AXIS_TLAST is 1 which indicates the last entry of the vector we go to the INPUT_LOADED state. In this state MAC performs the calculations and once the data is written to FIFO and then read from FIFO the done signal is asserted which takes the controller back to the START state.

Describe how your system's hardware changes when you change the parameters INW, M and N. Be specific about how the hardware components in your design will change as you change these parameters.

If we change the parameters M and N we will need to use large memories to store the matrix and the vector. Similarly most of the counters used are counting till either M or N. Changing these parameters will result in more flip flops being used and hence more combinational circuitry. More hardware will result in increased power consumption. The latency of the operation will also decrease as only one element is either written or read in one cycle.

Project Part 4: MSpVM



The RTL code of the top module is named “mems_tops.sv”.

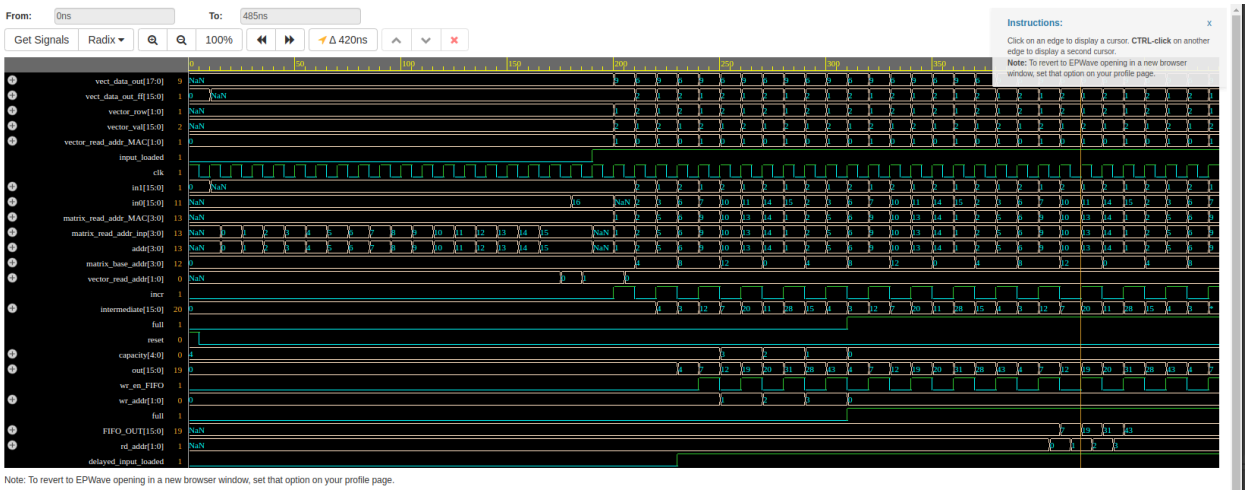
The EDA link is provided

https://www.edaplayground.com/x/Z9Pv

The final results in the form of screenshots are shown below

The inputs I provided for this test are

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{bmatrix} \quad \begin{bmatrix} 0 \\ 2 \\ 1 \\ 0 \end{bmatrix}$$



As we can see the 3rd last signal named “FIFO_OUT” shows the output 7,19,31 and 43 which is the correct answer.