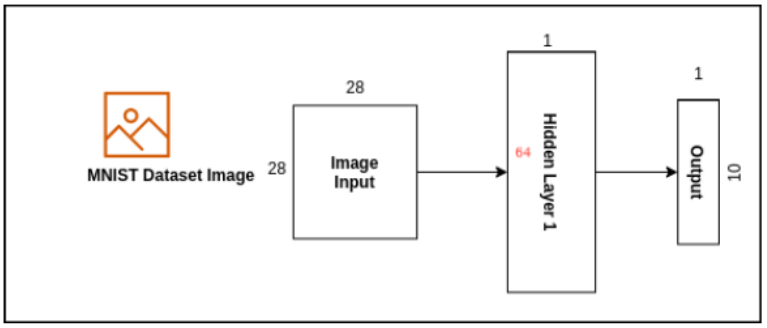
COL215P
Assignment 2 - Part 2
Machine Vision Through Neural Network Hardware

Shreya Arora(2019CS10401)
Ishita Chaudhary (2019CS10360)

In order to implement the control path for a 3-layer Multi-layer Perceptron (MLP) for the vision task, we have defined an FSM. The FSM sequences the data required starting from generating addresses of the data to reading the data, generating the cycle-wise signals to be sent to the datapath to control the modules in the design. This ensures that the execution is performed in a sequential and layer-wise manner. The states involved in our FSM design are described as follows:



State 1: In this state, we initialise the address that will be loaded for the subsequent computation operations. The Image addresses start getting stored from the address 0 and the weights and biases start getting stored from the address 1024.

Here, we also load the image, the weights, and the biases from the ROM (Read Only Memory) into the local memory. The **Input Image** of size 28x28 with each pixel of 8 bits is loaded from the addresses between 0-783 (784 8-bit wide words corresponding to the input image, and these words start getting stored from the address 0 (0000_{16})). The 50890 **Weights and Biases** are loaded from addresses between 1024-50899 (50816 weights and 74 biases of 8 bits each which start getting stored from the address 1024 (0400_{16})).

State 2: Now we perform the multiplication of the 8-bit weight and the 16-bit input/activation to generate a 24-bit output at layer 1, one column of weights in an iteration.

State 3: In this state, we truncate the output obtained from state 2 to form a 16-bit intermediate output using the shifter and accumulate the result of multiplication operation at layer 1.

State 4: In this state, we add the first bias, and pass the resulting vector to ReLU activation function and finally store these intermediate values in the RAM, which will then be read as input for the next layer.

State 5: Now we perform the multiplication operation for the 8-bit weight and the 16-bit input/activation to generate a 24-bit output at layer 2.

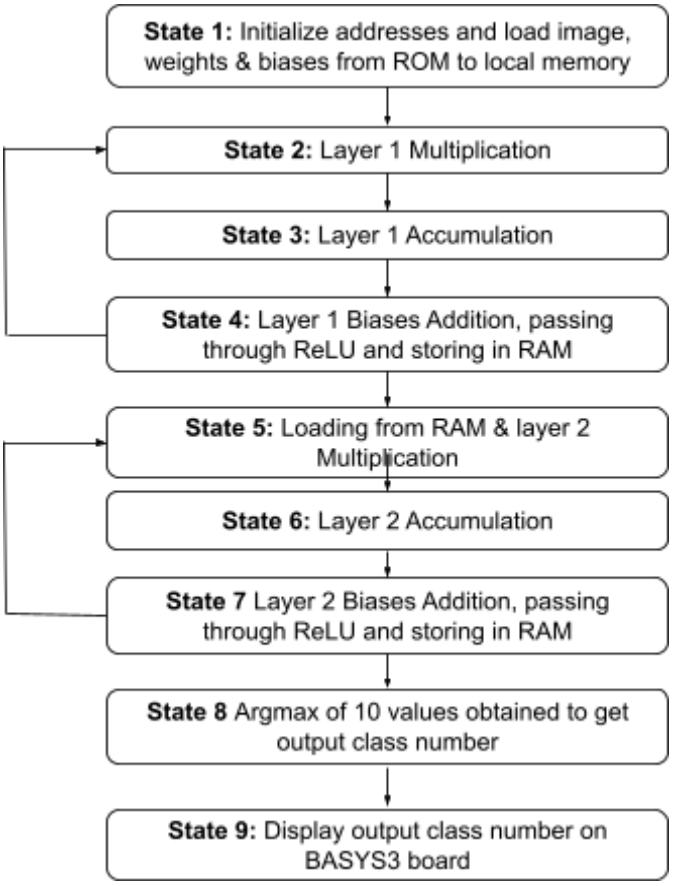
State 6: In this state, we truncate the output obtained in state 5 to form a 16-bit intermediate output using the shifter and accumulate the result of multiplication operation at layer 2.

State 7: In this state, we add the second bias, and pass the results obtained through ReLU activation function, we store these values in the RAM.

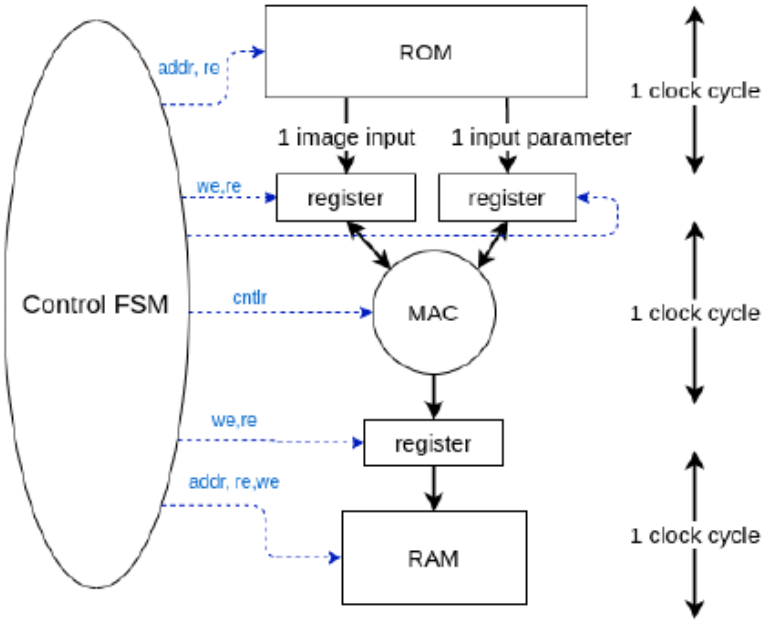
State 8: Finally, we take the index of the maximum of these 10 values that we have obtained to inference the given image, and on the basis of this, we give the output class number to be displayed on the seven segment on BASYS3 board.

The output number is given as an input to seven segement decoder to display the output class number as obtained in state 8 on the BASYS3 board.

FSM Diagram



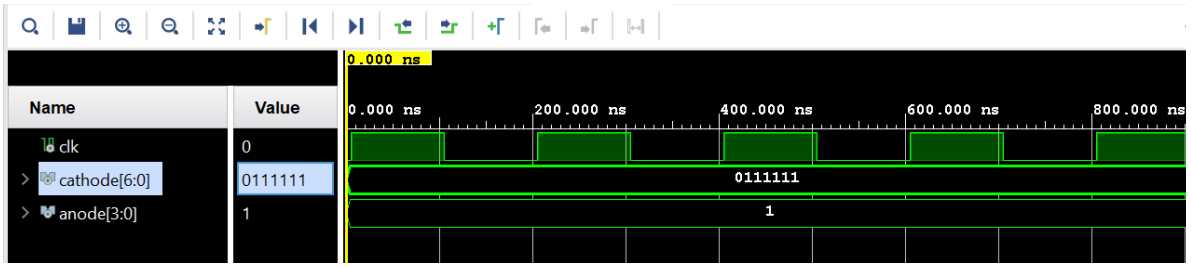
Datapath



Simulation Waveform

We ran the fsm_tb.vhd (testbench) for "imgdata_digit.7mif" file and got the following waveform:

At the beginning:



Final waveform (cathode value=1111000 corresponding to display 7 on FPGA board):