COL215 Assignment 2

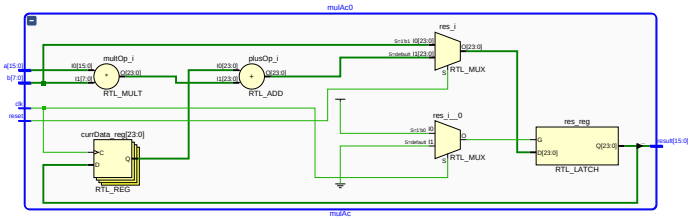
Tanmay
2020CS10399

Shivam
2020CS10383

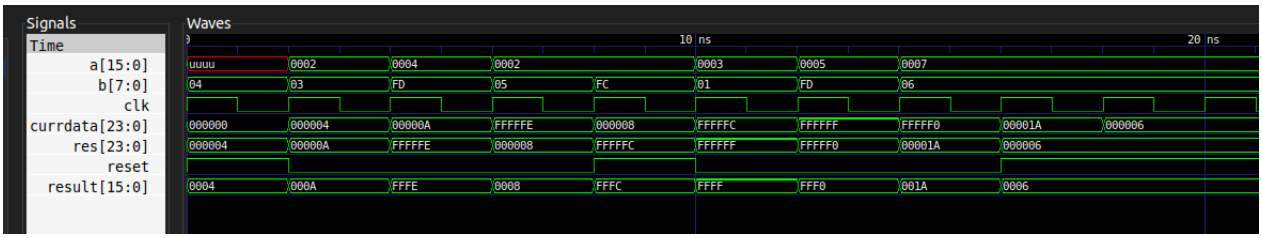
Modules used in assignment:

dataType.vhdl:
Idea is to create data type which is dynamic in size, can be used to save vector and matrix and can be passed through ports
∴ smallVector data type defines as array (natural range <>) of std_logic_vector(7 downto 0);
And vector data type defines as array (natural range <>) of std_logic_vector(15 downto 0);

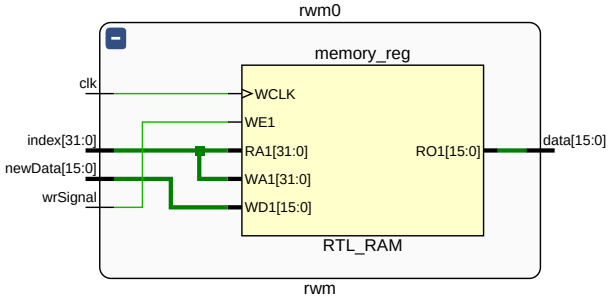
mulAc.vhdl:
This module is for multiply and accumulate.
It takes in input clk as std_logic, reset as std_logic, a as std_logic_vector(15 downto 0) and b as std_logic_vector(7 downto 0) and outputs the result.
It maintains an accumulated result of all the multiplications.
When the reset signal is on, it resets the accumulated value to input bias.
Block Diagram for mulAc:



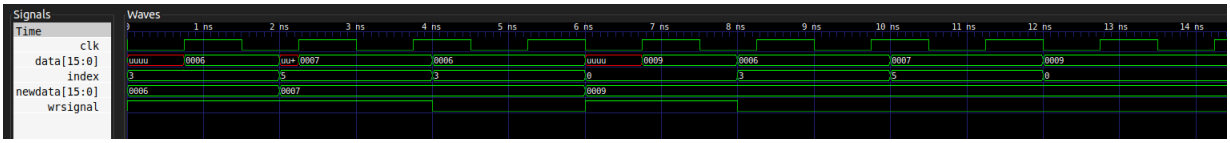
Waveform for testbench:



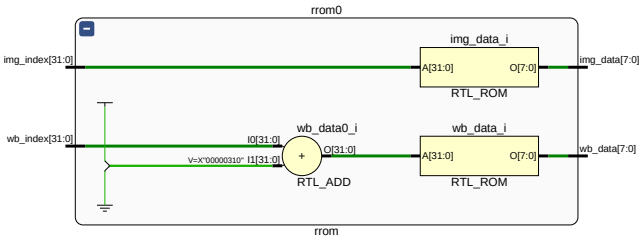
rwm.vhdl:
This module defines the entity Read-Write memory.
It takes input clock as std_logic, index as integer, wrSignal for write signal as std_logic, newData as std_logic_vector(15 downto 0), and gives, data as output as std_logic_vector(15 downto 0)
Block Diagram for rwm:



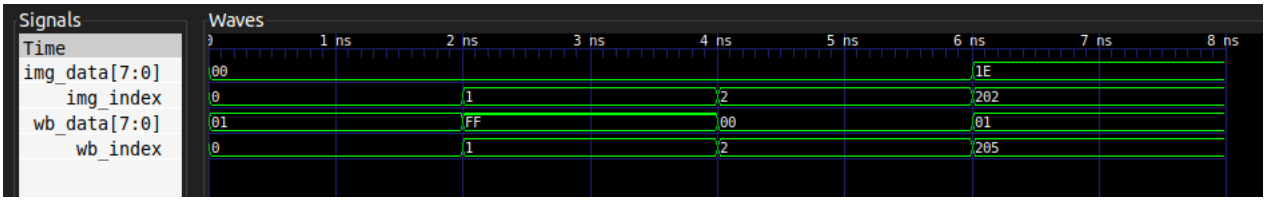
Waveform for testbench:



rom.vhdl:
This module defines the entity ROM.
It initialies the rom by taking input from mif file name of image file and weight_bias file with their sizes.
It also takes input image index and weight_bias index and outputs the image_data and weight_bias data.
Block Diagram for rom:

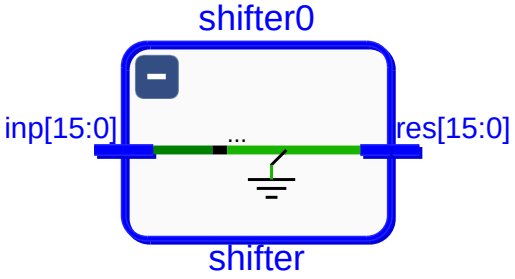


Waveform for testbench:

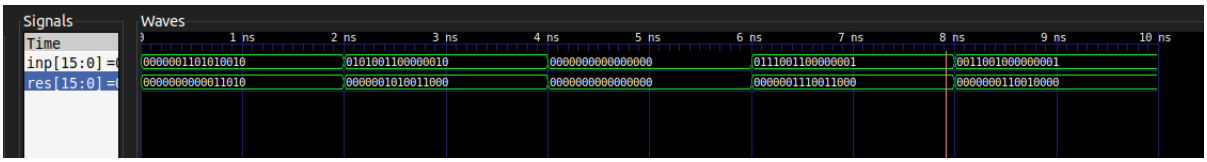


relu.vhdl:
This module defines the entity relu.
It takes in inp as std_logic_vector(15 downto 0) and checks the inp(15). If inp(15) = '1' ⇒ number is negative and assigns x"0000" to res otherwise res ⇐ inp.
When wrsignal is '1' then data ⇐ newData otherwise data remains same.
EP wave signal of the testbench:

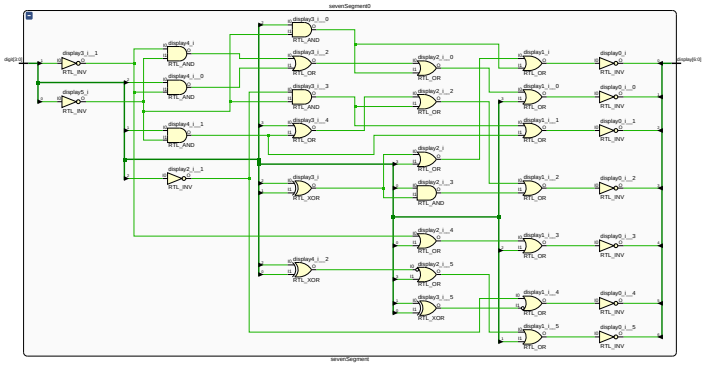
shifter.vhdl
This module defines the entity shifter.
It takes input a std_logic_vector(15 downto 0) and shifts it by 5 bits and output std_logic_vector(15 downto 0) such that input is shifted by 5 bits to the right. Also signed is not needed to be maintained as the positive value after passing through relu is only passed in shifter.
Block Diagram for shifter:



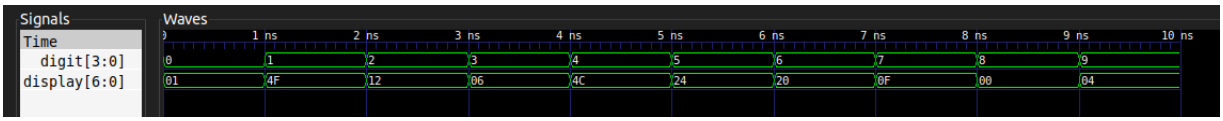
Waveform for testbench:



sevenSegment.vhdl
This module decodes the binary form of number to seven segment display.
It takes input std_logic_vector(3 downto 0) as input number and output std_logic_vector(6 downto 0) seg display.
Block Diagram for sevenSegment:

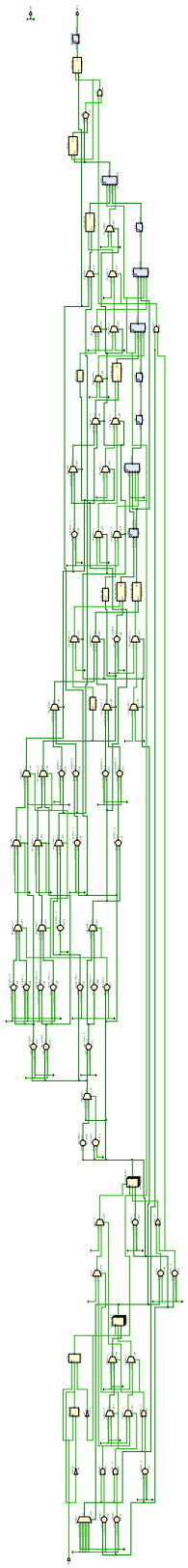


Waveform for testbench:



Idea of the design:

The main idea of the design is to import the mif files into the rom and let other process wait till the mif file is loaded completely.
rrom takes in mif file paths and their sizes and imports them into a vector datatype signal.
This is state 00 if FSM and it waits atmost for 10000 clock cycles.
Next FSM state is compute the layer 1 of neural network. Finding 1 value of layer1 take 785 clock cycle.
For calculating i^{th} value layer1 result, first of all mulAc is reset with the value bias. Then after every clock tick, two corresponding values of img_data and weight1 are multiplied and accumulated in the mulAc result.
Therefore, after 785^{th} clock tick, i^{th} value of layer1 is calculated and then passed through relu and shifter to finally store in the rwm.
In the third state of FSM, layer2 value is calculated using layer1 output and weight2 and bias2.
Each value of layer2 is calculated in 65 clock cycles.
In first value, mulAc is reset to the bias2 value and in every clock cycle, corresponding layer1 and weight1 values are multiplied and accumulated in mulAc.
Therefore, after 65^{th} clock tick, i^{th} value of layer2 is calculated and then passed through relu to finally store in the rwm1.
In the last FSM state, maximum of layer2 output value is calculated and the digit corresponding to maximum value is passed through sevenSegment decoder to output the digit on the board.
Block Diagram for complete design:



Waveform for testbench for imagedata_digit7.mif: