COL215P Assignment 2.2

Machine Vision Through Neural Network

Control Path and Data Path Design

Chinmay Mittal, Dhruv Tyagi
2020CS10336, 2020CS10341
7 October 2022

Block Diagram (Netlist)

The complete circuit in the form of block diagram generated by vivid's open elaborated design method is shown on the next page. The individual components have been included and explained as required in the corresponding sections.

Additional Components for Data Path

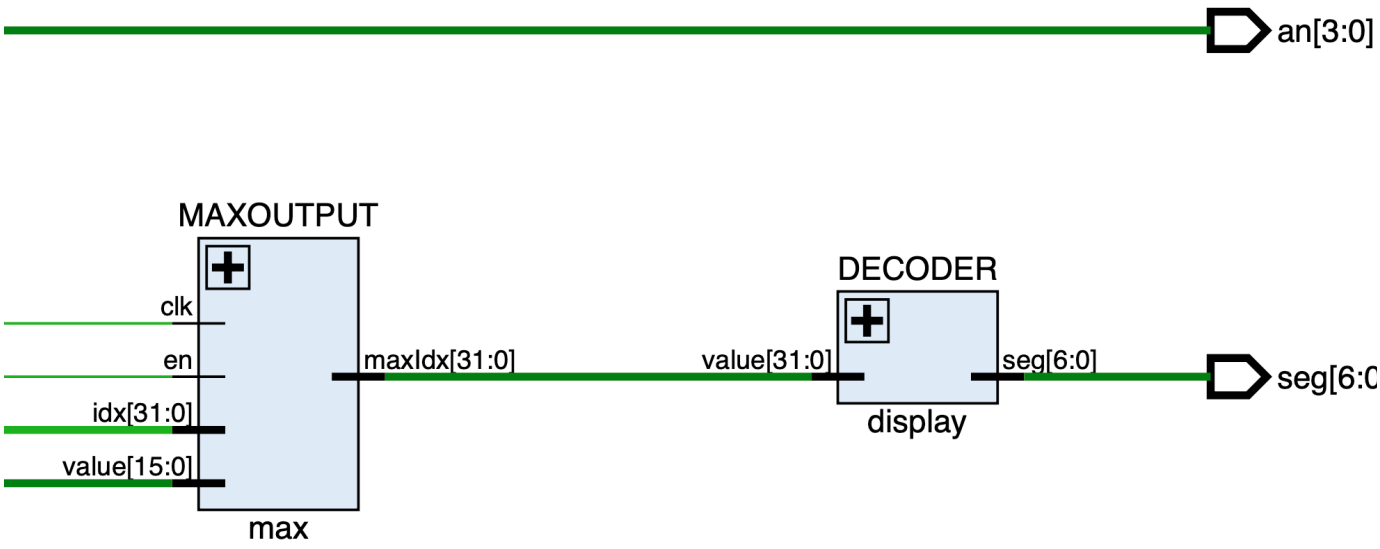
Decoder

This component will take as input the index to be displayed at the end of Neural Network computation and will output the cathode signals for the basis board.

The anode signals will be generated by the FSM which remain off when the computation is going on and when the computation is complete the anode signal for the lest most digit is made on.

MaxOutput

This component will be used to decide the prediction from the final layer. It compares activations of all 10 values one by one and keeps storing the best value. At the end the signal *maxIdx* will store the output of the neural networks as an integer which will be passed to the decoder.



Memory Architecture

Architecture design for memory components are explained below. Computational units are connected as per the block diagram shown above.

RAM1

For the first layer computation, RAM1 stores the image. Image is used repeatedly, so we have stored it in RAM1 to avoid multiple reads from ROM. ROM also has weight and biases stored and since we cannot perform 2 reads in one clock cycle from ROM, we have preloaded RAM1 with the image so that we are only reading one address from ROM in one clock cycle which we use to read the weights when we are simultaneously reading the image from RAM1. We also store 1 in this RAM which is multiplied with the bias. This ensures consistency in handling weights and biases where both can be treated similarly where weights can be multiplied with activations and biases can be multiplied with 1.

RAM2

Another RAM (called RAM2) is required where we store the output of first layer computation. This also stores final layer output. MaxOutput reads from this RAM to decide the final output state. We also store 1 in this RAM2 to handle the bias as explained above.

Register1

This register stores the activation values (16 bit) where the values are stored for computations for both the layers. It stores a value for computation so that other things can be loaded while computations are performed in the MAC unit. A multiplexor called reg1_select selects whether the input to this register is read from RAM1 or RAM2. This is required as RAM1 stores the activations for the first layer while RAM2 stores the activations for second layer.

Register2

This 8 bit register stores the weights and bias and is directly loaded from the ROM. The FSM decides the address of the ROM which has the corresponding weight or bias. This register is reused for both layers.

Register3

This register is used to store the value of MAC computation. Whenever a computation for a particular activation is completed the corresponding value is written to this register from where it is eventually written to RAM2. This register is reused for both the layers. We pass the output of the MAC through the comparator (ReLU) and shifter before writing it to this register.

FSM Description

There are a total of 5 states which are described below. Each state has different counters which when reach their maximum value, the state gets updated. The states have been named as per convenience as there is no logic in numbering the states as they have been.

State 0

State 0 reads the image from the ROM before computation begins. It also stores a '1' which gets multiplied by the bias and gets accumulated in the MAC unit.

State 1

State 1 corresponds to computation of the hidden layer. In this state, in each cycle we read one value from the ROM (weight/bias) and RAM1 (activation corresponding to image) which are then written in register 2 and register 1 respectively. MAC also keeps performing the computations synchronised with the reading of the weights and biases. When the first weight is multiplied, we enable the 'first' signal in the MAC denoting that we have started calculating a new activation. Input and output index signals are maintained so as to ensure that we are reading the right values (weights, biases, activations) and writing them in the write place.

State 3

State 3 is used is writes a '1' to RAM2 corresponding to the biases for second layer computation. Then we go to State 2.

State 2

This state is similar to the previous state except for the fact that the second layer is begin computed in this state. The only difference being that we write the values from this State in RAM2 again after reading the activations from RAM2 itself whereas in the first layer computation we read the activations from RAM1 and then wrote them in RAM2.

State 7

This state is used to compute the final answer by finding out the maximum values of all the final layer activations. We read the activations from RAM2 and send them to MaxOutput. At the end of this state, the computation is finished and we move to the terminal state 6

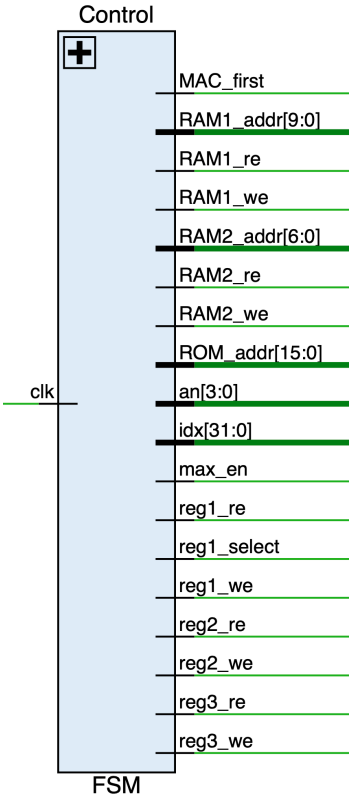
State 6

This state is the terminal state the anode value for the right most digit is switched on and the decoder displays the prediction of the Neural Network.

FSM Architecture and Control Signals

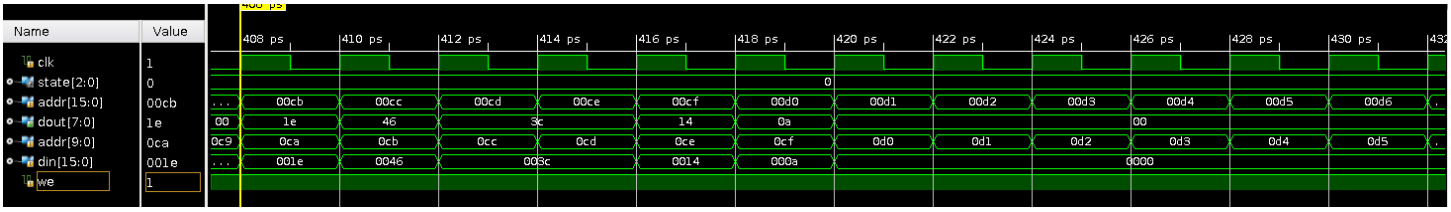
The following control signals have been used for controlling the Finite State Machine (FSM):

- MAC_first: If enabled, erases the old accumulated values.
- RAMi_addr: Data Address for the RAMi (i is 1 or 2).
- RAMi_re: Read enable for RAMi.
- RAMi_we: Write enable for RAMi.
- ROM_addr: Data address for the Read Only Memory.
- an: Controls the anode to enable the display.
- idx: Checks which index to compare in the maxoutput unit.
- max_en: Enable the maxoutput unit.
- regi_re: Enables the read for register i (i is 1,2,3).
- regi_we: Enables the write for register i.

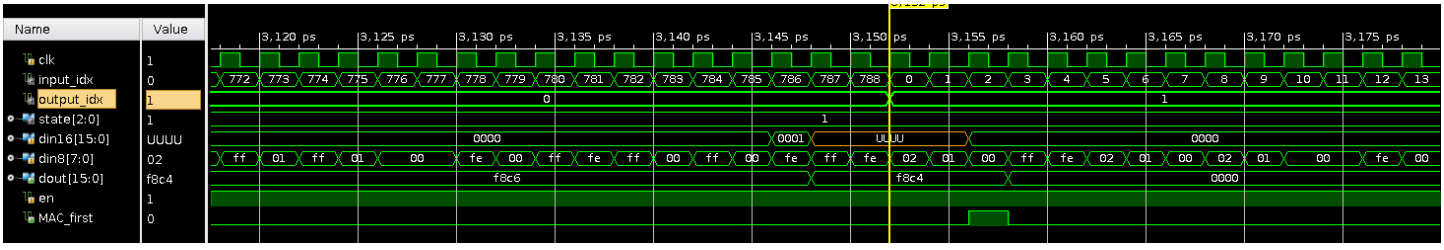


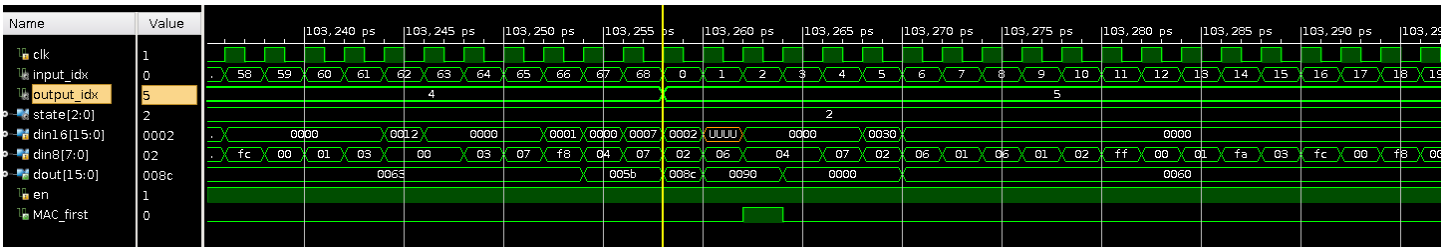
Simulations

Simulation Corresponding to State 0 where we are reading image data from ROM to local memory (RAM1).

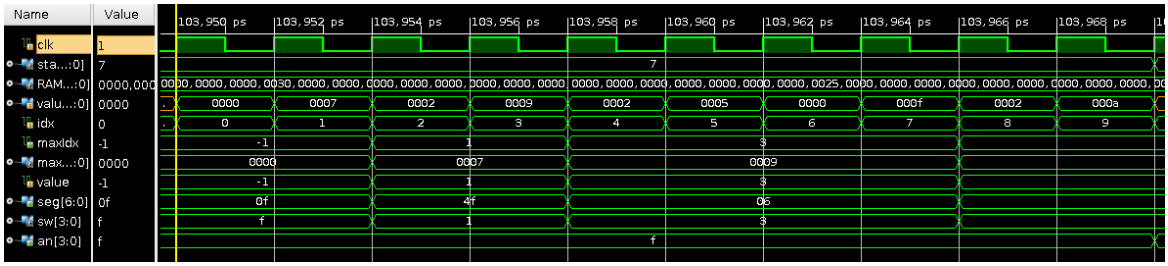


Computations for Layer 1 and Layer 2 in the MAC unit are shown below (first image is for layer 1 and second is for layer 2).

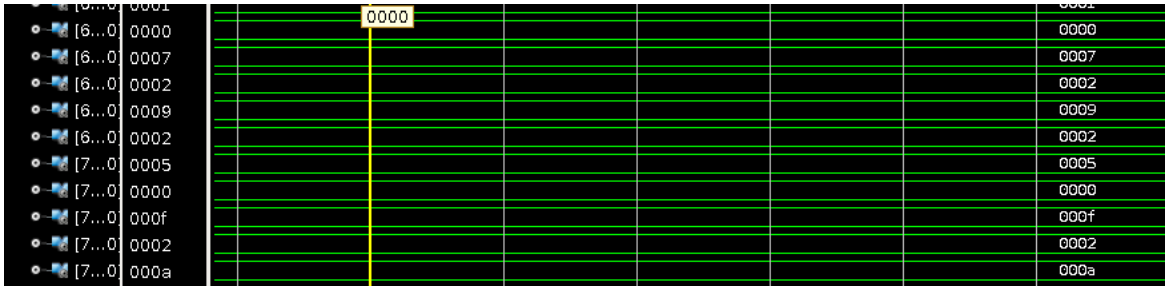




Maxoutput unit computations with clock cycle:



Final layer in RAM2 after all computations:



Final output computed after the maxoutput unit performs its computations: