Convolutional Neural Networks (CNNs) on FPGA Using SIMBA-like architecture (Parallel Vector MACs)

To Do:

1. Establish the UART communication between the system and the FPGA and ensure the read-write in BRAM.

2. You will be provided the trained model for the MNIST dataset classification (Note that images in the MNIST dataset are 28x28 in size).

MNIST dataset

The model will have the following layers and parameters:

a. Convolutional layer (32 3x3 filters) with ReLU activation
   [Input Size: 28x28; Output Size: 26x26x32]

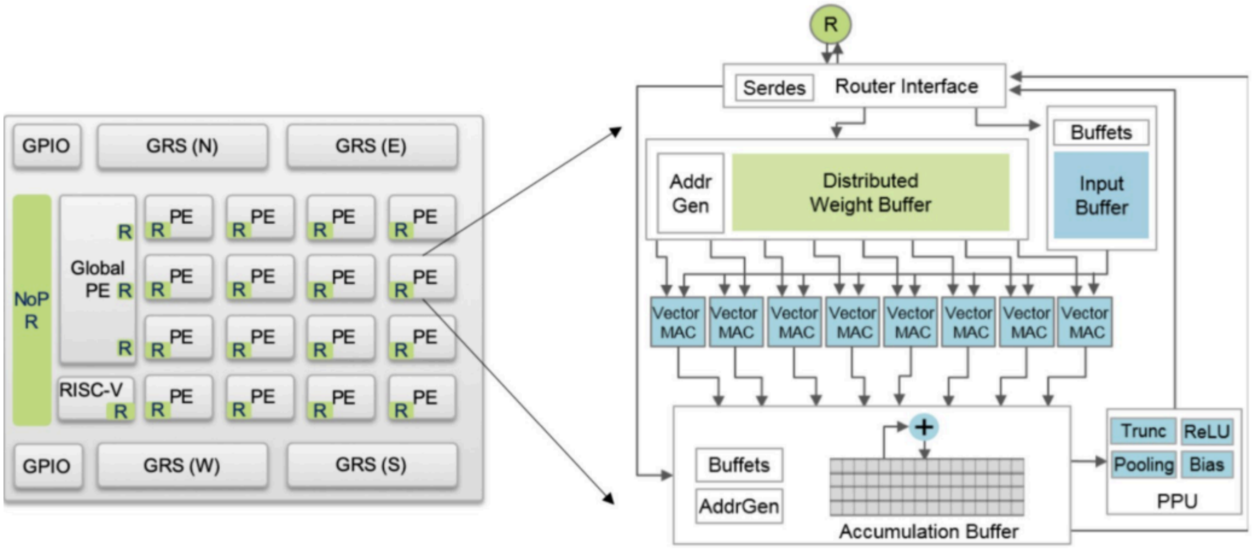
b. MaxPooling Layer (2x2 filter)
   [Input Size: 26x26x32; Output Size: 13x13x32]

   c. Fully Connected Layer (64 neurons)
      [Size after flattening: 5408; Output Size: 64]

   d. Output Layer (10 classes for MNIST)

3. The group will first code up the network in Python (NOT Pytorch). (This should be completed by the end of the second week)

4. Once the Python code is verified, you can start writing the Verilog code. **Note** that you will be performing a SIMBA-like implementation for the neural network.



Simba Chiplet

Simba PE

SIMBA Processing Element (PE) with Vector MACs

5. You can deploy the trained model for inference on FPGA. Your final goal would be image classification for the MNIST dataset.

Note that You are supposed to use a bit-width of 8 bits for all the layers.

Timeline:

For submission purposes, you can create a GitHub repository where you can push the codes and update the documentation after every week.

Week-1 submission: 13th March

Week-1 meeting: 14th March

Week-2 submission: 21th March

Week-2 meeting: 22th March

Week-3 submission: 5th April

Week-3 meeting: 6th April

Week-4 submission: 12th April

Week-4 meeting: 14th April