NeuroFPGA

Implementing a Spiking Neural Network on an FPGA

Evan Lang, Vivek Patel, Raunak S Rajpal

Background

- Neuromorphic computers mimic the structure of the human brain using artificial neurons.

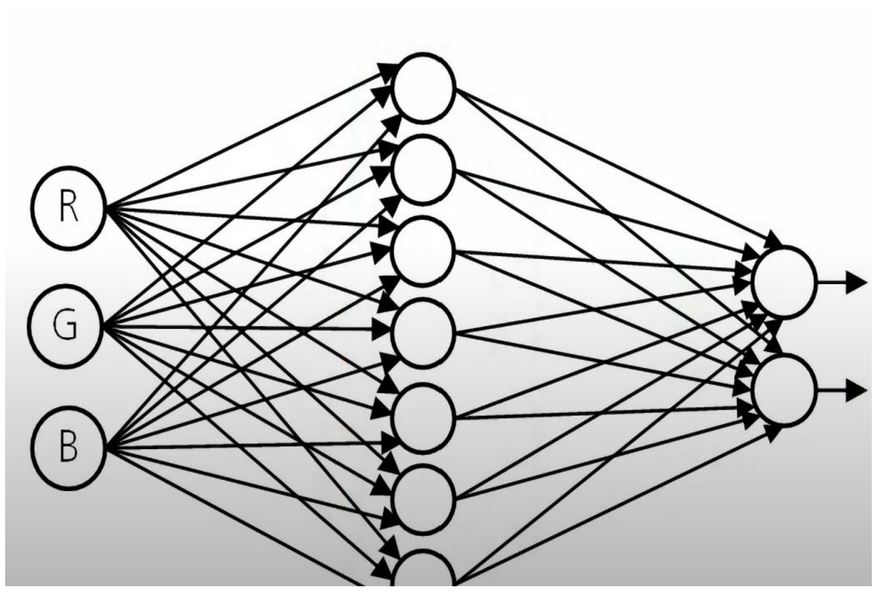
- Spiking Neural Networks (SNNs) are a specific artificial networks that mimic biological neurons by communicating information in short pulses.

$$\tau \frac{du(t)}{dt} = -[u(t) - u_{rest}] + RI(t),$$

Neural Network

- The architecture of the neural network includes three inputs which are RGB values, 8 hidden nodes, and 2 output values which are blue and yellow.

- When none of the outputs are activated a third category is introduced to indicate no color detected.

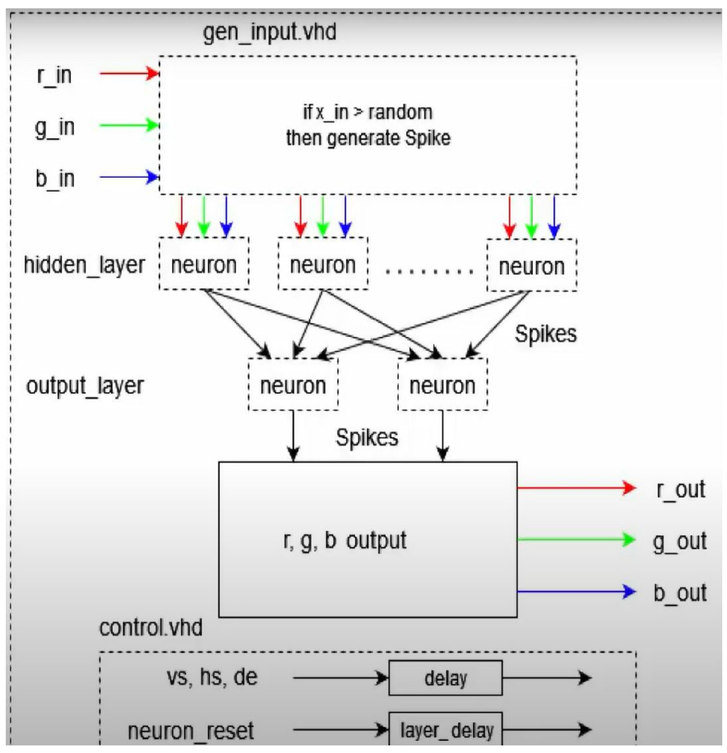


Objectives

- Simulate a SNN on a consumer grade FPGA by implementing an open source FPGA Design.

- The FPGA will be able to detect blue and yellow in given images.

Block Diagram



FPGA

- The low cost and consumer grade FPGA chosen for this project is the Intel Cyclone 10.

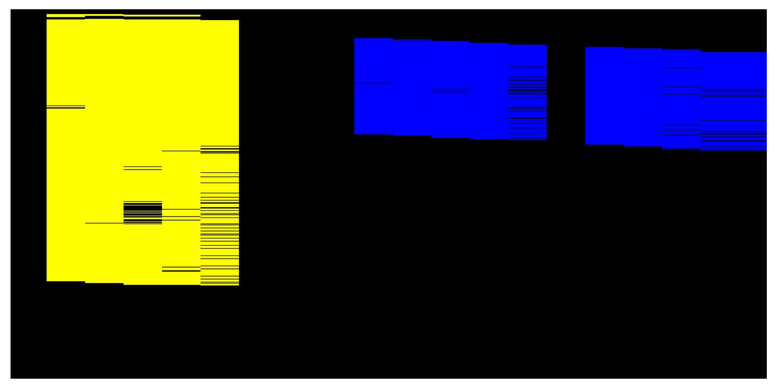
- This FPGA allows us to utilize Questa software to simulate the FPGA.

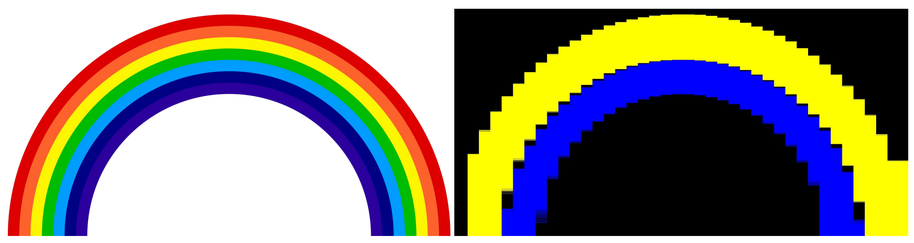
Training the SNN

- The neural network is trained externally via Python and MATLAB using an ReLu activation function.

- Images are manually labelled, assigning individual colors to a grayscale label.

Results





- Works great in ideal scenarios (color's clearly separated)
- Struggles with similar colors in close proximity

Conclusion

- The SNN can properly identify colors
- More training is required for edge cases and gradients
- Cyclone 10 LP FPGA can easily handle the SNN requirements, and leaves room to expand the network with more nodes

Future Works

- Expand color detection to include red, orange, green, and purple

- Improve accuracy of detection with expanded library of training data

References

- "Spiking Neural Network on FPGA" by FPGA Remote Vision Lab

- Exploring Neuromorphic Computing Based on Spiking Neural Networks: Algorithms to Hardware: by Rathi et al