Preliminaries of Implementing an SNN on an FPGA

Name: Vedika Jain
Supervisor: Prof. Sajad Ahmad Loan

Department of Electronics and Communication Engineering
Faculty of Engineering and Technology
Jamia Millia Islamia, New Delhi-110025

List of Contents

Introduction

Spiking Neural Networks (SNNs) represent a promising approach to neural computation, mimicking biological neural systems by using discrete spikes for communication between neurons. These networks are well-suited for FPGA implementation due to their parallel nature. This project focuses on the design of an SNN using Izhikevich neurons, exploring its high-level architecture, synaptic connectivity, and a simulation of its behavior with STDP learning rules. The goal is to provide a clear pathway toward FPGA realization in the future, ensuring scalability and performance optimization.

High Level Design

Neuron Model: Izhikevich Neuron

The Izhikevich model is a powerful, simplified spiking neuron model known for its biological realism and computational efficiency. The model includes two main equations governing the dynamics of the membrane potential v and the recovery variable u.

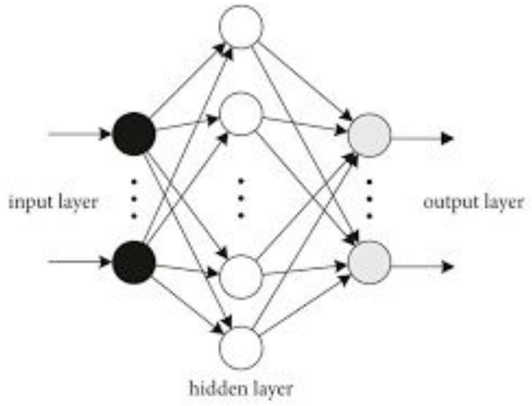
$$\frac{dv}{dt} = 0.04v^2 + 5v + 140 - u + I$$

$$\frac{du}{dt} = a(bv - u)$$

where I is the input current, and a, b, c, and d are constants governing the behavior of the neuron. Whenever v exceeds a threshold (set to 30mV), the neuron spikes and is reset. Upon firing, v is reset to a value c, and u is updated with a value d.

Synaptic Connectivity: Feedforward Network

The SNN is based on a simple feedforward architecture where each neuron in a layer connects to every neuron in the subsequent layer. There is no recurrent feedback between layers for now, but they can be added later.



Dataflow: Clock-Driven

The dataflow model chosen for this SNN is time-stepped simulation. This method ensures that each timestep is handled sequentially, with spikes communicated between neurons. Each timestep represents an iteration where neurons compute their membrane potential, check if they exceed the firing threshold, and propagate spikes to the next layer.

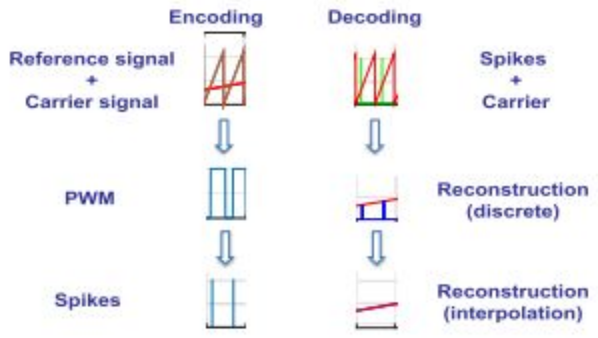
Training Algorithm: Base-2 STDP

The synaptic weights are updated via STDP (Spike-Timing Dependent Plasticity), where the strength of the connection depends on the relative timing of spikes between the pre- and post-synaptic neurons. STDP equations are made from exponential functions. Accuracy and hardware cost are the main concerns simultaneously when a large-scale implementation is targeted. The exponential function can be converted to a base-2 function.

$$\Delta w = \begin{cases} \Delta w^{+} = A^{+}.2^{-1.4375(\frac{\Delta t}{\tau^{+}}}), & \text{if } \Delta t \geq 0 \\ \Delta w^{-} = -A^{-}.2^{1.4375(\frac{\Delta t}{\tau^{-}}}), & \text{if } \Delta t \leq 0 \end{cases}$$

Encoding Scheme: PWM based Encoding

To encode input stimuli into spikes, PWM encoding is used. Pulse-width modulation (PWM) is a method of encoding the intensity of the input signal into the width of the pulses. This encoding allows the input data to be represented as a series of spikes, which can be processed by the SNN.

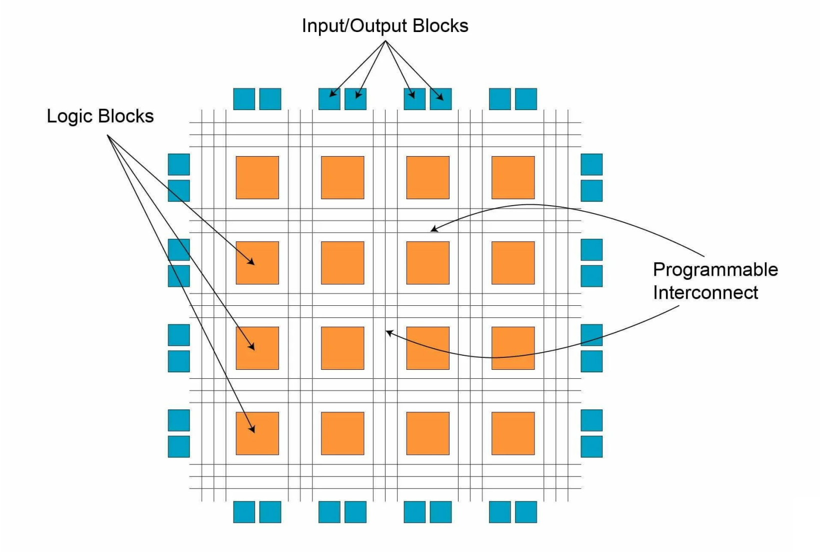

Encoding
Decoding
Reference signal + Carrier signal
Spikes + Carrier
PWM
Reconstruction (discrete)
Spikes
Reconstruction (interpolation)

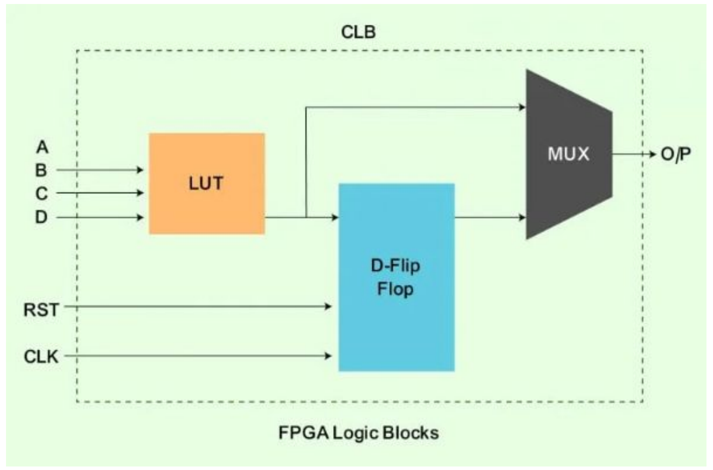
FPGA Architecture

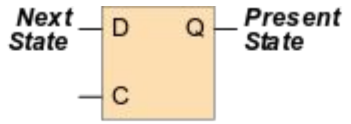
A basic FPGA architecture consists of thousands of fundamental elements called configurable logic blocks (CLBs) surrounded by a system of programmable interconnects, called a fabric, that routes signals between CLBs. Input/output (I/O) blocks interface between the FPGA and external devices.

A CLB gives the FPGA its ability to accept different hardware configurations. CLBs can be programmed to perform almost any logic function. The individual CLB contains a number of discrete logic components including look-up tables (LUTs) and flip-flops.

A lookup table (LUT) determines what the output is for any given input(s). A flip-flop is a circuit that has two stable states and can be used to store state information.

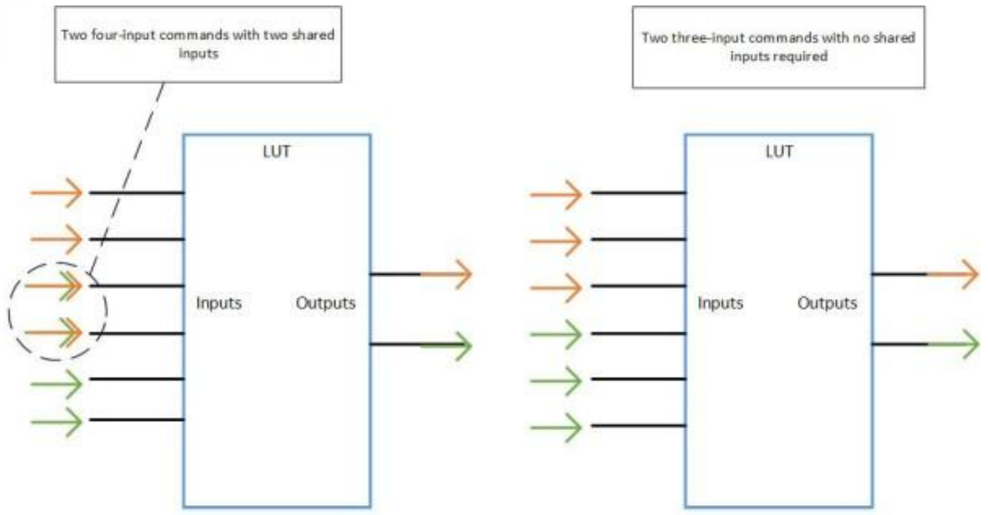






Data input to a memory device is called the *Next State*

Output from a memory device is called the *Present State*


Two four-input commands with two shared inputs
LUT
Inputs
Outputs
Two three-input commands with no shared inputs required
LUT
Inputs
Outputs

Conclusion

- SNNs represent a promising approach to neural computation. These networks are well-suited for FPGA implementation due to their parallel nature.
- The Izhikevich neuron strikes a balance between biological realism and computational efficiency.
- Feedforward architectures are simple to implement and clock-driven dataflow works well with FPGA’s parallel processing capabilities.
- The main challenge in implementing STDP is the exponential term that is required to store and retrieve spike timing. The Based-2 model offers affordability as well as precision.

Future Work

The endeavour will be to design a neural network that can be used for temporal processing applications.

- BRAM blocks on the FPGA can be used for memory requirements.
- Look-Up Tables and/or DSP Slices can be used for arithmetic-heavy blocks like Neural Processing Units.
- IO blocks will handle communication with external devices.
- Clocking resources will ensure correct timing and synchronization.

References

[1] Pham, Quoc Trung, et al. "A review of SNN implementation on FPGA." 2021 international conference on multimedia analysis and pattern recognition (MAPR). IEEE, 2021.

[2] Wang, Kuanchuan, et al. "Comparison and Selection of Spike Encoding Algorithms for SNN on FPGA." IEEE Transactions on Biomedical Circuits and Systems 17.1 (2023): 129-141.

[3] Arriandiaga, Ander, et al. "Pulsewidth modulation-based algorithm for spike phase encoding and decoding of time-dependent analog data." IEEE Transactions on Neural Networks and Learning Systems 31.10 (2019): 3920-3931.

[4] Gomar, Shaghayegh, and Majid Ahmadi. "Digital realization of PSTDP and TSTDP learning." 2018 International Joint Conference on Neural Networks (IJCNN). IEEE, 2018.

[5] Izhikevich, Eugene M. "Simple model of spiking neurons." IEEE Transactions on neural networks 14.6 (2003): 1569-1572.