Design and Implementation of Leaky Integrate-and-Fire (LIF) Neurons on FPGA

Ali Yazdanpanah
Supervisors: Mansoor Zolghadri Jahromi
Department of Electrical and Computer Engineering, Shiraz University

Abstract

This work presents the design, simulation, and FPGA realization of a compact spiking neural network based on the Leaky Integrate-and-Fire (LIF) model. The goal is to reproduce essential neuronal timing behavior within a resource-efficient digital flow that is suitable for teaching laboratories and small research prototypes. The implementation targets a Xilinx Artix-7 device and is developed in Vivado 2017.1 using Q4.12 fixed-point arithmetic.

We derive a discrete-time LIF update from the continuous RC model and adopt an exact-step formulation under piecewise constant input. The membrane state update uses a single fixed-point multiply and addition with explicit saturation, followed by thresholding, reset, and an optional refractory counter. All parameters are configurable, including leak factor, threshold, reset level, refractory duration, and synaptic delay. Although the core is current based for simplicity, the layer wrapper tracks separate excitatory and inhibitory conductances with geometric decay, which captures useful temporal filtering while keeping logic cost low.

The architecture is organized into three synchronous blocks. A Poisson spike generator produces presynaptic events via a 16-bit LFSR and comparator. The LIF neuron core performs the fixed-point update with saturating arithmetic and clean one-cycle spike outputs. An SNN layer adds per-synapse bounded delay lines, accumulates delivered weights into excitatory and inhibitory conductances, and computes the net drive for each neuron. Hooks for simple spike-timing dependent plasticity are included for future experiments, yet the baseline build focuses on deterministic timing, numerical robustness, and easy observability of internal signals.

Verification combines cycle-accurate RTL simulation with software references. Single-neuron benches confirm agreement in membrane trajectories, threshold crossings, resets, and refractory clamping. A two-neuron experiment driven by independent Poisson sources demonstrates coincidence detection and the expected concentration of output spikes where inputs overlap. A scene-driven suite with steps, idle windows, bursts, and constrained random segments tests delay delivery, conductance decay, and spike timing integrity across changing regimes. Throughput and latency are characterized under varying synaptic fan-in. Higher fan-in increases aggregate delivered events per second and modestly raises mean spike latency due to longer reduction paths. The pipeline can be retimed to flatten this curve without sacrificing throughput.

Synthesis on an Artix-7 XC7A12T using Vivado 2017.1 meets a 100 MHz clock with comfortable slack. The two-neuron reference top consumes 524 LUTs, 360 registers, and 10 DSP48E1 slices when learning arithmetic is enabled. Disabling plasticity reduces the DSP usage to one slice per neuron. The design uses no block RAM in the baseline configuration because synaptic delays map to short shift registers. Resource use scales linearly with neuron count when fan-in and delay bounds remain modest, which fits the constraints of low-cost boards used in coursework.

The prototype highlights several practical considerations for digital neuromorphic builds. Saturation prevents wraparound under bursts, Q4.12 provides an effective balance between precision and DSP footprint, and near-unity leaks are representable at 100 MHz for microsecond-scale dynamics. Limitations include fixed-point headroom under extreme bursts, the need for

additional routing and configuration fabric for large networks, and the omission of richer biophysical features such as adaptation. Planned extensions include on-chip plasticity experiments, lightweight AER routing for tiling, mixed-precision sweeps, and sensor-in-the-loop demonstrations.

In summary, the project bridges textbook LIF dynamics and synthesizable fixed-point RTL. It validates behavior against a software reference, characterizes performance under controlled traffic, and reports concrete utilization numbers on a small Artix-7 device. The result is a transparent and reproducible starting point for students and practitioners who want to explore spiking computation within a standard FPGA tool flow.

Keywords: Neuromorphic hardware, spiking neural networks, FPGA implementation, Leaky Integrate-and-Fire neuron, fixed-point arithmetic.