ECE 110 Chip Design Report
—
Coupled Izhikevich Neurons

Noah Williams

Oct 23, 2024

University of California, Santa Cruz
Department of Biomolecular Engineering and Bioinformatics

Baskin Engineering

Abstract. A linear system of two Izhikevich neurons was implemented for the Tiny Tapeout TT9 project [1]. Models followed the dynamical system for regular spiking cortical neurons proposed by Izhikevich [2]. Parameters were matched to known values, and adjusted to compensate for hardware limitations. Neurons were linearly connected such that the output of the first neuron was directly fed to the input of the second neuron. New spiking patterns were generated in this fashion, distinct from those observed with individual neurons. A Python program was developed to create a software model of Izhikevich neural networks and validate findings from hardware simulations[1].

Background. The Izhikevich neuron dynamical system was developed as a computationally efficient and biologically accurate reduction of the Hodgkins-Huxly (HH) model [2], [3]. Bifurcation theory was used to distill the HH four coupled dynamical equations into a system of two dynamical equations representing membrane potential (v) and membrane potential moderator (u), while maintaining transition state thresholds and spiking behavior.

$$\frac{dv}{dt} = 0.04v^2 + 5v + 140 - u + I$$
$$\frac{du}{dt} = a(bv - u)$$
$$if\ v \geq 30mV\ \{v \leftarrow c,\ u \leftarrow u + d\}$$

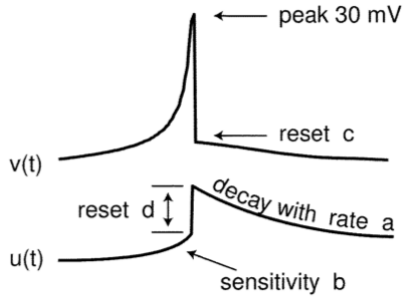

Fig. 1.: (Left) Izhikevich system of equations,
(right) visualization of parameters [2].

The membrane potential differential equation was determined by fitting spiking dynamics to rat cortical neuron population spiking data. Dimensionless parameters represent recovery time scale (a), coupling strength between u and v (b), reset membrane potential (c), and reset membrane potential modulator (d). Different combinations of these parameters yield various spiking patterns across excitatory and inhibitory cortical neurons, consistent with classifications from biological spiking data. This results in a highly flexible, biologically realistic model, with low computational requirements, making it an ideal candidate for the space-limited Tiny-Tapeout project.

[1] Initial submitted program was designed for validation of spiking behavior of a single neuron. This was later extended to enable multiple neurons.

Circuit[2]. The dynamical system (Fig. 1.) was used to develop a logic circuit for a single neuron, implemented in Verilog, and then extended to include two connected neurons. Parameters for the Izhikevich model were chosen from values known to produce Regular Spiking (a = .02, b = .2, c = -65, d = 8), and stored as registers [2]. Scaled fixed point floats (Q9.7) were used for high accuracy and adherence to biological values [4]. 16-bit internal calculations from 8-bit IO were achieved through concatenation of zeros to input, and precision dropped for output [5].

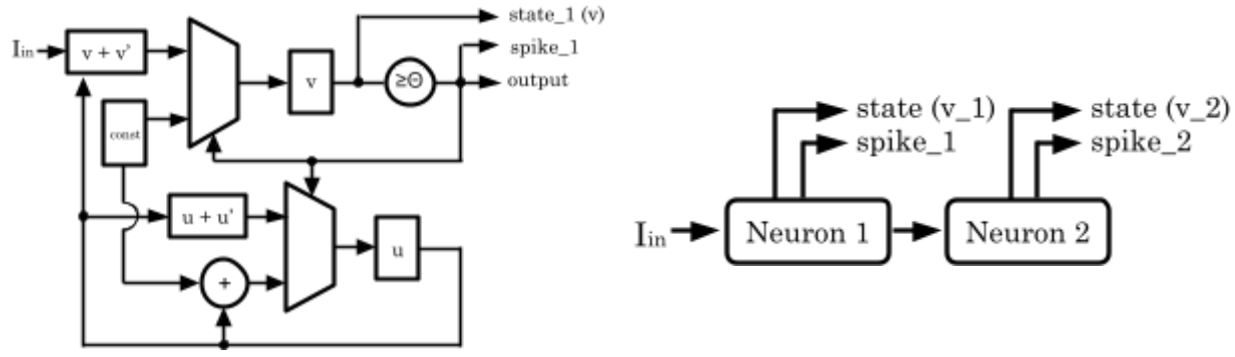

Fig. 2.: (Left) High-level logic circuit of a single Izhikevich neuron.
(Right) Black box circuit of Tiny Tapeout design.

The circuit takes input for clock, reset, and an 8-bit input current. Sequential logic with a flip flop checks for reset state upon each positive edge clock cycle. If the condition is met, v and u are set to 0. Otherwise, v and u are updated from the state equations. Combinational logic is used to check for spiking, assign spike output, update u and v after spike conditions, and calculate next states for v and u. While no spike has occurred, the next states for u and v are calculated from current states, stored in registers, and used to update values. If membrane potential surpasses a defined threshold, a 1 bit spike is assigned to output. Upon spiking, a multiplexer switches the states and assigns u and v to reset values stored in registers.
8-bit membrane potential (state) is output directly to the next neuron as input, along with a 1-bit output for spiking. Both 8-bit values and 1-bit spikes are output per neuron, making a total of 4 output values for the chip design.

[2] The official submission includes mistakes that are worth discussing. (1) membrane potential for individual neurons is better modeled by: dv/dt=0.04v^2+4.1v + 108, b=-0.1 [2]. (2) To achieve spikes with hardware implementation, parameters were changed to a = .02, b = .15, c = -65, d = 10. (3) A direct connection was used between the two neurons; it should have been implemented with a variable proportion controlled by unused IO ports. (4) There should have been a combined spike output wire for network spiking. (5) The two neurons should have been set up as an oscillatory network.

Simulation. Validation of the circuit was determined by developing a benchmark Python program, then testing with cocotb and VS Code Surfer waveform viewer [6]. Thresholds for testing were heuristically determined through experimentation with the Python implementation. Three points were found and categorized as no spiking, some spiking, and frequent spiking (see Fig. 4, Supplementary Materials).

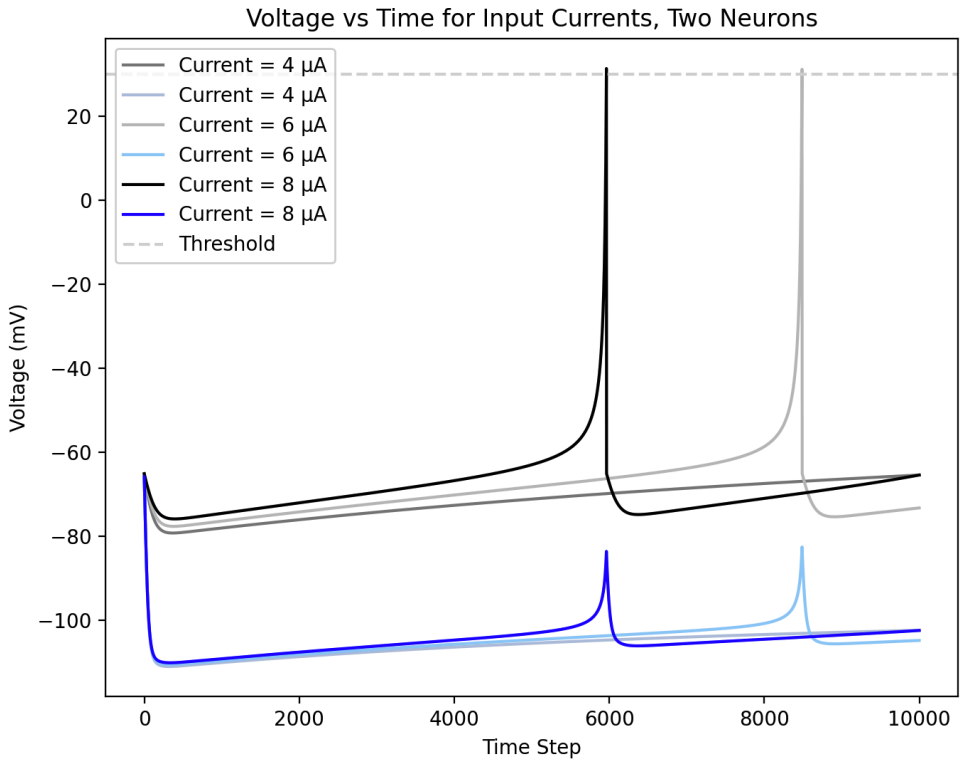

Fig. 3. Python simulation of directly coupled Izhikevich neurons. Values determined heuristically. First neuron represented in greyscale, second neuron by shades of blue. Figure created by author.

The lower bound was used as the test value to ensure the system was not spiking when the threshold was not reached. The next value was slightly above the minimum value to induce spiking and tested the sensitivity of the circuit. Finally, a third value (not shown in the Python program) was introduced to ensure no overflow errors occurred with maximum input current. In between test inputs was a reset, to ensure that the reset system of the circuit was functional, and to clear residual membrane potential. Cocotb was used to generate Python test scripts that outlined each of the four tests, and waveforms were viewed in VS Code Surfer.

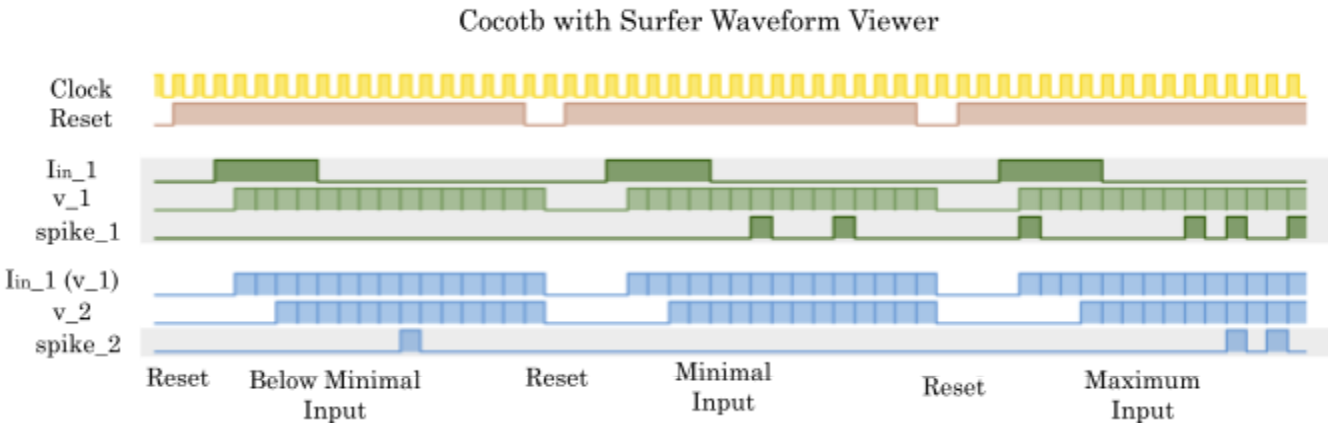

Fig. 3. Cocotb test viewed in VS Code Surfer. Results compared to graphs generated from Python script. Figure created by author.

Benchmark results matched somewhat to output from the Python simulation. It was difficult to accurately time when the spikes occurred, making it impossible to use assertions in the benchmark (leading to the belief of an error in the circuit or Python program). The Python code does not take clock cycles into account which could be the reason for discrepancies. The Verilog implementation was much more sensitive than the Python implementation, which indicated that either the parameter b was incorrect, or an issue with decay rates. The behavior of the second neuron was difficult to debug and should not have spiked with below-minimal input. Issues aside, Fig. 3 demonstrates a somewhat partially successful implementation of two coupled Izhikevich neurons, which mostly spike when expected from the benchmark program.

References.

[1] W. Noah, *Tiny Tapeout Izhikevich Nueron System*. [Online]. Available: https://github.com/nomuwill/tt_um_nomuwill

[2] E. M. Izhikevich, "Simple model of spiking neurons," *IEEE Trans. Neural Netw.*, vol. 14, no. 6, pp. 1569–1572, Nov. 2003, doi: 10.1109/TNN.2003.820440.

[3] "Hodgkin-Huxley Model." [Online]. Available: https://neuronaldynamics.epfl.ch/online/Ch2.S2.html

[4] "Q Number Formats." [Online]. Available: https://en.wikipedia.org/wiki/Q_(number_format)#cite_note-TI_2003-1

[5] "Hodgkin-Huxley Tiny Tapeout." [Online]. Available: https://github.com/jeshraghian/tt05-hodgkin-huxley

[6] *cocotb*. [Online]. Available: https://docs.cocotb.org/en/stable/