Week 6 Challenge - Challenge #19: Implement a binary LIF neuron

# Overview

This document summarizes the complete conversation and implementation process for the Week 6 Challenges (ECE 410/510 Spring 2025). It includes all user queries, code solutions, visualization enhancements, and error handling in a progressive and tutorial-friendly format.

# Challenge Descriptions

## Challenge #19: Binary LIF Neuron

Objective: Implement a binary Leaky Integrate-and-Fire (LIF) neuron model in Python.  
Method: Used leak factor λ and threshold θ to accumulate input and simulate neuron spikes.  
Result: Successfully visualized neuron inputs, spikes, and internal potential values using matplotlib.

## Challenge #20: Crossbar Matrix-Vector Multiplication

Objective: Simulate a 4x4 and later an 8x8 resistive crossbar performing analog matrix-vector multiplication.  
Method: Converted resistance to conductance (G = 1/R), then used dot product to compute output currents.  
Result: Verified output with printed values and plotted both conductance matrix and output currents.

# User Prompts and Solutions

## Q1: Can you give me the solutions for the above challenges provided and guide me through how to run them on any of the platform

A1: Provided Verilog and SPICE solutions along with run instructions using Icarus Verilog and ngspice.

## Q2: Give it to me in python

A2: Converted both challenges to Python implementations. Used NumPy and Matplotlib.

## Q3: is this the expected output?

A3: Verified matrix-vector multiplication results and confirmed correctness of the output.

## Q4: Would you like me to package both solutions into a GitHub-ready folder with README and requirements.txt?

A4: Packaged all Python code with README, requirements.txt, and explanation.

## Q5: yes

A5: Generated downloadable zip archive containing all solution files.

## Q6: help me how to upload it to git

A6: Gave full GitHub upload tutorial including git init, add, commit, and push commands.

## Q7: Can I get a graph for cross\_bar matrix is that possible?

A7: Enhanced crossbar\_sim.py to include heatmap and bar chart using matplotlib.

## Q8: yes

A8: Extended crossbar from 4x4 to 8x8 and provided visual representation for conductance and output.

## Q9: Can you give documentation of this chat also include all the prompts I have asked...

A9: You are reading the generated documentation.

# Vibe Coding Highlights

• Used progressive Python scripting to match learning goals from hardware to software simulation.  
• Introduced leak dynamics and threshold reset logic in LIF neurons.  
• Mapped analog matrix-vector ops with digital approximation using NumPy.  
• Enhanced user understanding by visually mapping conductance strength to crossbar behavior.  
• Upgraded models from 4x4 to 8x8 with zero issues thanks to vectorized implementation.