**Recommendation Letter**

It is a pleasure to write this recommendation for Bhavana Manikyanahalli Srinivasegowda. I have known Bhavana since March 2025, when she enrolled in my ECE 410/510 course on *Hardware for AI/ML* at Portland State University. Over the course of ten weeks, she demonstrated initiative, a willingness to learn, and the ability to engage with technically challenging material. I recommend her without hesitation for roles or opportunities that involve hardware-software integration, AI systems, or digital design.

Bhavana’s weekly submissions reflect a steady engagement with a broad range of topics in the course. In Weeks 1 and 2, she explored how large language models like ChatGPT can aid in chip design and examined performance bottlenecks in Python programs. She followed this by implementing and visualizing basic neural networks like perceptrons to solve logic problems—an effort that showed both conceptual understanding and practical execution.

In the middle part of the course, she focused on reinforcement learning and GPU acceleration. She tested the FrozenLake Q-learning algorithm in both Python and CUDA and wrote additional CUDA programs like SAXPY and Fibonacci generators. These exercises helped her develop benchmarking and profiling skills as she compared different implementations. The hands-on nature of these activities built her confidence in performance-oriented computing.

In Weeks 5 to 7, she expanded her work to system-level design, looking at systolic arrays for sorting, modeling spiking neurons in Verilog, and simulating crossbar-based matrix-vector multiplication in SPICE. She also worked on an SPI interface and used the cocotb framework to bridge Verilog with Python-based testbenches. These efforts required careful integration of hardware and software—an area where she showed good technical independence.

In the final phase of the course, Bhavana delved into neuromorphic computing and edge AI hardware. She studied commercial architectures like BrainChip’s Akida and modeled memristive devices. Her GitHub repository provides a timestamped record of her weekly progress, covering topics from basic neural models to SPI co-simulation and genetic algorithm acceleration. The repository includes organized folders and descriptive commits, such as Week9 pushed, Challenge 25, and Recommendation Letter modified, indicating consistent documentation of her evolving project work.

For her final project, she designed a hardware accelerator for a genetic algorithm. She identified compute-heavy sections in the algorithm and implemented them in Verilog. To test the design, she used cocotb for co-simulation and synthesized the hardware using OpenLane. She iteratively improved timing and area metrics, benchmarking her hardware against a software baseline. While she used large language models for occasional code suggestions, the integration and testing decisions reflected her own understanding and methodical approach.

In conclusion, she has shown the ability to stay focused, work independently, and apply tools effectively across both software and hardware contexts. She approached problems pragmatically and followed through with project flow. Based on her performance and contributions, I believe she deserves good grades for this course and recommend her for any opportunity that values thoughtful system-level design and learning agility.