Taha Khokhar

taha.khokhar@mail.utoronto.ca | (306) 570-9786 | ON, Canada | linkedin.com/in/taha-khokhar2t3| github.com/MTahaK

Education

University of Toronto, Toronto, ON — September 2019 – April 2024

Bachelor of Applied Science: Computer Engineering

Skills and Abilities

- **Coding Languages/Libraries:** C/C++, Python, (System) Verilog, Ruby, Pytorch, TensorFlow, Perl
- **Tools:** Git, GitHub, Perforce, Crontab, Unity, ANTLR4
- **Technical Skills:**
  - Experienced in building gaming computers.
  - Proficient with Unix systems.

Work Experience

AMD - ASIC Design & Verification Engineering Intern — May 2022 – April 2023

- Launched, monitored, and maintained regression tests for **AMD's** VCN decoder **ASIC** architecture. Maintained weekly linting reports.
- Managed linting tool updates and compatibility
- Managed and monitored job deployment to ensure test queue health and load.
- Identified race conditions in end-to-end sanity testbenches, and developed safeguards in **Python** against them to ensure accurate testing and reporting of test results.
- Developed test flow systems in **Python & Ruby** to implement constant end-to-end test uptime, and further improved uptime by a factor of **5 over a two-week period**.
- Built automatic test failure reporting systems in **Python, Ruby, and C-Shell scripting**, **reducing average response time by a factor of 3**.
- Automated a variety of verification flows using **Python, Ruby, and C-Shell scripts** to **ensure constant update and report** of passes and failures.
- Analyzed and reported performance metrics on several encode-decode **ASIC** architectures.
- Analyzed performance data between iterations of leading **ASIC** architecture and between current leading **AMD market ASIC** architecture.
- Maintained storage server and drive health, developed **Python, C-Shell, & Ruby scripts** to report and automatically clean up bloated directories. Deployment vastly improved response to storage blow-ups from **at least one day to within an hour.**

Personal & Class Projects

Capstone: Acceleration of Raytracing via a Fixed-Point FPGA Raytracer — September 2023 - Present

- Developing a fixed-point raytracer using FPGA hardware connected to a host machine for control as a Capstone project. Implementing in **C** and **System Verilog.**
- FPGA module will include intersection & shading module with multiple dedicated units to take advantage of parallelism, and a Bounding Volume Hierarchy traversal module using parallelized traversal cores.
- Implementing raycasting and ray-traversal of Bounding Volume Hierarchy on FPGA in **System Verilog.**
- Implementing control of operation and drawing of pixels in **C** on host PC.

Compiler for Simple C-Based Language — September 2023 - Present

- Developing a compiler for a small **C**-like language called SmallC. Will implement in **C++.**
- Built a simple lexer using **ANTLR4.**

Implementation of a Memory-Mapped Peripheral — February 2022

- Built a Peripheral Controller working on an Avalon bus in **C, Verilog,** and built using **Quartus.**
- Programmed I/O devices to the peripheral as memory-mapped I/O in **C.**

Multi-Threaded Text-Conferencing Application — November 2021

- Built a webserver using a socket programming library in **C.** Webserver handled sending, receipt, and broadcasting of chat rooms that it could create.
- Built client-side architecture in **C** to push messages to webserver, and to request for and join chat rooms created on the server.
- Instantiated multi-threaded system on server in **C** to process data from multiple users simultaneously.

Social Media Bot Detection (Machine Learning, NLP) — June 2021 – August 2021

- Developed an **NLP**-based **machine learning** program in **Python** (using **Pytorch**) in a team of three to classify Twitter posts as being made by a human or a bot-controlled account.
- Wrote, tested, and tuned primary and baseline models. Baseline followed basic classifier architecture. Primary models achieved highly improved success over the baseline.
- Primary implementations utilize LSTM-type and GRU-type RNNs and GloVe embeddings. Achieved final accuracy of 79.35% on, limited due to the only relevant public dataset available sourcing Tweets up to only 2011.
- Identified various factors that affect model performance, such as biases in the data (ex: bot Tweets often had URLs).

Recreation of 'Battle City' (1985) in C and ARM — April 2021

- Programmed a low-level recreation of 'Battle City' (1985) in **C** and **ARM**.
- Rendered game graphics directly using double-buffering and game logic. Implemented in **C.**
- Built game input with **ARM** interrupts implemented in **C**.

Engineering Software Communication & Design — January 2021 – April 2021

- Developed a graphical navigation system in **C++**. Worked in a three-person team.
- Created graphical interface using **GTK** in **C++**, used OpenStreetMap data for system.
- Programmed an A* Pathfinding implementation in **C++** with complexity O(n*logn) and a greedy multi-Dijkstra implementation in C++ to solve a travelling salesman problem variation with complexity O(k*n*logn).

Engineering Design Team Projects — September 2019 - April 2020

- Worked in six-person team to develop two conceptual engineering design solutions for two clients in each four-month period. The second involved constant interaction between the client and the team.
- September-December 2019: Designed a Hydraulic Height-Adjustable Dock to combat docking complications during floods in Ward's Island in the Toronto Islands.
- January-April 2020: Worked with Minoru Tanaka of the JCCC to design a Laser Alarm System to assist in the avoidance of jams upon deployment of the JCCC's Kobayashi Hall Hussey-MX3 Telescopic Seating System.

Relevant Courses

Operating Systems
Computer Networks
Computer Hardware
Algorithms & Data Structures
Software Design & Communication
Fundamentals of Machine Learning
Introduction to Machine Learning
Compilers & Interpreters
Software Engineering
Computer Systems Programming
Computer Security
Engineering Strategies & Practices