Taisir Hassan

☎ 647-667-3006 | ✉ taisir.hassan@uwaterloo.ca | linkedin.com/in/taisir-hassan | github.com/taisirhassan | taisirhassan.netlify.app

Education

University of Waterloo — Waterloo, ON

Candidate for BAsc in Honours Computer Engineering, Minor in Comb & Optimization — *Expected Graduation: May 2028*

- **Cumulative GPA:** 3.7
- **Relevant Courses**: Data Structures & Algorithms (C++), Digital Circuits (VHDL), Digital Computers (RISC-V ASM), Electronic Circuits, Linear Algebra, Numerical Methods (MATLAB)

Experience

Onboard Payload Embedded Software Engineering Co-op — Jan 2025 – Apr 2025

Telesat — *Ottawa, ON*

- Developed **containerization architecture** for satellite simulator, initially creating Ubuntu 22.04 development and runtime environments, then successfully overseeing migration to Ubuntu 24.04 while maintaining compatibility across **Versal FPGA** and **SX4000** hardware platforms
- Created **Debian packages** for simulator components in CI/CD pipeline, implementing **Ansible** automation for deployment and configuration management, improving environment consistency by **40%** and reducing setup time by **25%**
- Led **integration efforts** for simulated satellite components including **Ciena routers** and **modem banks**, implementing **SR-MPLS networking** with **VPP data plane** to enable proper traffic routing with segment routing between payload systems and ground segments
- Automated network management testing by developing scripts for **gNMI** client installation and **gRPC** execution against the simulator, reducing configuration time by **35%** and enabling consistent verification of satellite control system functionality

Electrical Team Member — May 2024 – Present

UW Rocketry — *Waterloo, ON*

- Developed and optimized **PWM** and **ADC drivers** with low-pass filter logic for the **PIC18F26K83** in **C**, reducing signal noise by approximately **15%** as measured with oscilloscopes and wave generators
- Implemented **RTOS-compatible I2C**, **UART**, and **IMU handler** for **STM32 HAL** in **C/C++**, while also creating a comprehensive unit test framework with **GTest** and **FFF** for embedded driver validation
- Engineered **APB bridge interface** and **SPI controller** in **SystemVerilog** for the **SRAD GNSS receiver** FPGA project, creating testbenches and CSR verification to ensure core satellite positioning functionality and reliable signal processing
- Led schematic design for **USB debug board PCB** featuring **12V boost converter** and **CAN-USB** communication with **Harwin** connectors

Autonomous Vehicle Embedded Systems Developer — December 2023 – Present

Watonomous Design Team — *Waterloo, ON*

- Engineered and debugged **ROS2** nodes for camera and LiDAR functionality in autonomous vehicles using **C++** and **Python**
- Optimized **Docker** containers for simultaneous operation of **FLIR camera** and **LiDAR**, resolving networking and **ROS DDS** compatibility issues
- Implemented **X11 forwarding** for real-time visualization of sensor data within containers, enhancing development and debugging capabilities

Data Quality Specialist — May 2023 – Sept 2023, May 2024 - Aug 2024

Cohere.ai — *Toronto, ON*

- Collaborated in the quality assurance of a state-of-the-art Large Language Model.
- Achieved an average task completion rate of **98%**, ensuring timely and accurate completion of text and code-based tasks.
- Decreased data discrepancies and inconsistencies by **25%** through meticulous data quality control measures

Junior Fullstack Developer — Jan 2022 – Apr 2022

Playfair Technologies — *Toronto, ON*

- Spearheaded the launch of a **React Native** mobile application, leading to a remarkable **50%** increase in user engagement. This initiative involved a strategic transition from existing applications. ensuring timely and accurate completion of text-based tasks
- Significantly improved application performance and response time by **30%**, achieved through meticulous optimization of back-end functionalities utilizing **Scala**.
- Enhanced the application's deployment process by integrating **Docker** containers, leading to a **40%** reduction in deployment time. This advancement contributed to quicker and more efficient software releases.

Projects

URA Research Project: Digital Circuit State Verification via De Bruijn Sequences | *Scala, Verilog, SystemVerilog*

- Developing a **Verilog** parser and synthesis engine to extract gate-level circuit descriptions and state transitions from hardware designs
- Implementing De Bruijn sequence generation algorithms to create optimal test sequences that guarantee complete state coverage through Hamiltonian cycle analysis
- Creating a verification framework that automatically generates test vectors to traverse all possible state combinations in digital circuit designs

Chrome Dinosaur Game ASIC | *Verilog, TinyTapeout*

- Led rendering system development for Chrome-inspired dinosaur game, designing sprite animations, scrolling background, and score display modules while coordinating cross-team integration for complete VGA implementation
- Architected ROM-based sprite and background rendering within 2-tile ASIC constraints, creating interfaces between Player Group, Image Buffer, and display systems for synchronized 640x480 output.

32-bit RISC-V Softcore Processor | *Verilog, GTKWave, Icarus Verilog*

- Designed and implemented RV32I base instruction set in **Verilog**, featuring 5-stage pipeline with hazard detection/forwarding logic
- Created test bench and assembly programs to validate instruction functionality, pipeline hazards, and branching logic
- Utilized **GTKWave** for waveform analysis and debugging of pipeline stages, ensuring correct instruction execution and timing

Traffic Light Controller | *VHDL, Quartus Prime*

- Designed a Traffic Light Controller using **VHDL** to model sequential logic and state machines, integrating pedestrian crossing requests with traffic light sequences for an FPGA implementation.
- Employed both **Moore** and **Mealy** finite state machine models to design a synchronized traffic control system, ensuring safe and efficient pedestrian and vehicular movement.
- Utilized **Quartus Prime** for VHDL code compilation, simulation, and FPGA programming, demonstrating proficiency in digital design tools and methodologies.

Technical Skills

Languages: C/C++, Java, Python, C#, Go, VHDL, SystemVerilog/Verilog, Rust, MATLAB, SQL, JavaScript/ Typescript, HTML/CSS
CAD/Engineering Software: Verilator, Yosys, KiCAD, AutoCAD, Simulink, Multisim, LTspice, Vivado, Quartus Prime, PSpice
Communication Protocols: I2C, SPI, UART, CAN
Frameworks/ Libraries: UVM, FreeRTOS, pandas, NumPy, Matplotlib, OpenCV, ROS, React, PyTorch, TensorFlow, React Native
Developer Tools: Git, CMake, Bash, Linux, Docker, Kubernetes,, GDB, GCC, Google Cloud Platform, VSCode, Visual Studio, CLion, Eclipse, CI/CD, Agile, Jira, Github, Gitlab, AWS (S3, Cloudfront, Terraform, X-Ray)