

# Mahmoud Abumandour

+ 1 (788) 320-8958 | BC, Canada | mahmoud\_abumandour@sfu.ca | [Website](#) | [Linkedin](#) | [GitHub](#)

## EDUCATION

<b>Simon Fraser University</b>	<b>Sep 2024 – Present</b>
<i>Ph.D. in Computer Science (GPA: 4.11/4.33)</i>	<i>BC, Canada</i>
<b>Simon Fraser University</b>	<b>Sep 2022 – Aug 2024</b>
<i>Master of Science in Computer Science (GPA: 4.07/4.33)</i>	<i>BC, Canada</i>
Thesis: <b>Resilient Neural Networks at the Edge: Uncovering and Mitigating Bit-Flip Vulnerabilities</b>	
<b>Mansoura University</b>	<b>Sep 2017 – Jul 2022</b>
<i>Bachelor of Science in Computer and Communication Engineering (GPA: 3.96/4.0)</i>	<i>Mansoura, Egypt</i>
Ranked first over a class of 180 students	

## EXPERIENCE

<b>Simon Fraser University</b>	<b>BC, Canada</b>
<i>Graduate Research Assistant</i>	<b>Sep 2022 – Present</b>
<ul style="list-style-type: none"><li>Conducted research in machine learning, system, and hardware security</li><li>Designed a black-box bit-flip attack which degrades modern LLMs to 0% accuracy using less than 25 bit flips</li><li>Developed a model stealing framework using symbolic execution to infer model architectures from inference binaries</li><li>Designed a bit-flip protection for LLMs, eliminating white-box bit-flip attacks on GPUs</li></ul>	
<b>Simon Fraser University</b>	<b>BC, Canada</b>
<i>Teaching Assistant</i>	<b>Jan 2023 – Present</b>
<ul style="list-style-type: none"><li>Conduct tutorials and lab sessions, grading assignments and exams, and providing support during office hours</li><li>Courses: Intro to Computer Systems, Principles of Compiler Design, Computer Architecture, Distributed Systems</li></ul>	
<b>Intel Corporation</b>	<b>Santa Clara, CA (Remote)</b>
<i>CPU Architecture Intern</i>	<b>Jan 2024 – May 2024</b>
<ul style="list-style-type: none"><li>Researched, modelled, and assessed CPU front-end features, including instruction prefetching and caching</li><li>Performed workload analysis to categorize based on instruction cache footprint and branch behavior</li><li>Conducted comparative studies between functional and cycle-accurate simulators to identify sources of miscorrelation</li></ul>	
<b>Google Summer of Code (RTEMS)</b>	<b>Remote</b>
<i>Student Developer</i>	<b>May 2022 – Sep 2022</b>
<ul style="list-style-type: none"><li>Achieved 8x speedup over the previous release notes generator by using a multi-threaded architecture</li><li>Automated release data fetching from RTEMS bug tracker and Markdown to RST &amp; PDF generation</li></ul>	
<b>Master Micro</b>	<b>Cairo, Egypt</b>
<i>Software Engineering Intern</i>	<b>Oct 2021 – Feb 2022</b>
<ul style="list-style-type: none"><li>Designed a range format for an EDA design lookup table file, reducing query time by 50% over a binary format</li></ul>	
<b>Google Summer of Code (QEMU)</b>	
<i>Student Developer</i>	<b>May 2021 – Aug 2021</b>
<ul style="list-style-type: none"><li>Implemented multi-core, multi-level cache performance emulation of user-space and full-system workloads</li><li>Improved the system call tracing by making its reports more script-friendly for post-processing</li></ul>	

## PUBLICATIONS

---

- **Mahmoud Abumandour**, Srinija Ramichetty, Guru Venkataramani, and Alaa R. Alameldeen. 2025. *WeightSentry: Real-Time Bit-Flip Protection for Deep Neural Networks on GPUs*. In Hardware and Architectural Support for Security and Privacy 2025 (HASP 2025)

## TALKS

---

- *Position-Adaptive Temporal Sparsity for KV Caches in Long-Context LLMs*. In Sparsity, the Key Ingredient from HPC to Efficient LLMs, Co-located with the International Symposium on Microarchitecture (MICRO) 2025

## PROJECTS

---

- **[The Kyoto Compiler and Fuzzer](#)**: Designed the Kyoto Programming Language and its compiler. Used LLVM for code generation and analysis. Implemented a grammar-based rust fuzzer for the compiler.
- **[Symbolic Execution Engine \(Mnemosyne\)](#)**: Built a QEMU binary tracing plugin and a symbolic execution engine
- **[Fuzzing with RISC-V Emulation](#)**: Developed a RISC-V 64-bit functional emulator for userspace fuzzing. Increased test generation throughput linearly with available resources by mitigating kernel overhead of native execution
- **[Database Engine \(RheaDB\)](#)**: Implemented a disk-oriented DBMS with SQL support, in-memory pool caching, B+ Tree indexing, and JDBC driver
- **[AES Encryption Core](#)**: Designed a low-power AES encryption core for FPGA. Reduced area and power consumption by more than 80% over a high-throughput pipelined design
- **[Hyperthreaded, Software-Interlocked RISC Processor](#)**: A multi-threaded five-stage pipelined RISC core for FPGA and a custom assembler with software interlocking, achieving 5x more throughput over single-threaded execution

## SKILLS

---

**Programming Languages:** C++, C, x86 Assembly, Rust, Python, Bash Scripting, Java

**Tools:** Ghidra, Gem5, LLVM, Git, Docker, Valgrind, perf, PyTorch, Tensorflow

**Platforms:** Linux, QEMU, FPGA, ARM Cortex M4, Raspberry Pi

**Hardware Design Tools:** Xilinx Vivado, ModelSim, SystemVerilog, VHDL

## OPEN-SOURCE CONTRIBUTIONS

---

- **RISC-V Newlib**: Profiled and optimized the newlib standard library implementation for RISC-V. Used QEMU, Spike, Gem5, and a RISC-V Raspberry Pi board for benchmarking.
- **SerenityOS**: Defined a global OS versioning API. Increased user-space utilities POSIX compliance. Improved the SerenityOS DBMS SQL support
- **QEMU**: Modernized the usage of locking and memory allocation APIs by using scope-based locks and automatically freed allocations. Redefined plugins' configuration interface adhering to modern QEMU standards