Shakil Ahmed

Dhaka, Bangladesh

+8801786751384 | shakil.000024@gmail.com | ahmedshakill.github.io | github.com/ahmedshakill | linkedin.com/in/shakil-ahmed-a06103174

Personal Profile

A computer science bachelor passionate about Compilers LLVM/MLIR, GPU, RISCV.

Education

Shahjalal University of Science and Technology *Sylhet, Bangladesh*

BSc in Computer Science and Engineering *December, 2017 - April, 2023*

- **CGPA : 3.24**
- **Total Credit Completed : 161**
- **Graduation Date : April, 2023**

Work Experience

Dynamic Solution Innovators *Dhaka*

Assistant Software Engineer *April 2023 - January 2025*

- Our team verifies RISC-V chip design through software simulation. I worked on benchmarking the cycle and instruction count of the design using Dhrystone and Coremark.
- Studied about porting freertos to RISC-V and Symmetric Multi Processing with freertos.
- Improved and maintained the HAL as the hardware features kept updating. HAL targeted a simulated hardware.
- Had an exposure with ROS2 and Gazebo simulator. I programmed and simulated a car with C++ and ROS2.

Google Summer of Code 2022 *Remote*

Intern *June 2022 - Oct 2022*

- Perticipated in GSoC 2022 as a code contributor and worked in the ***Enzyme*** project to enable LLVM's new Pass Manager support for Enzyme. My work focused on enabling Enzyme pass to be used with New Pass Manager of LLVM side by side with the Legacy Pass Manager.

PRs

HEIR involves Homomorphic Encryption and MLIR

- ***https://github.com/google/heir/pull/1058***
- ***https://github.com/google/heir/pull/789***
- ***https://github.com/google/heir/pull/991***
- ***https://github.com/google/heir/pull/765***

Projects

- **Load Store Analysis Pass:** An LLVM based compiler pass to find aliasing load store instructions inside a function. Got familiar with AliasAnalysis passes such as MemoryDependenceAnalysis, MemorySSA. Source is available ***here***.
- **RISC-V Simulator:** A RISC-V simulator written using C++17 following OOP principles. It is able to fetch, decode and execute load-store, arithmetic and logical shift, arithmetic operations and conditional branching instructions without pipeline support. Source is available ***here***.
- **Chip8 Emulator for Android:** It is an emulator that emulates chip8 platform written in modern C++ using NDK. Also used SDL2 for graphics support. Used an array of std::function to create a table of methods where methods correspond to each opcode. At runtime opcode from game ROM are read and corresponding method is selected from array and invoked allowing dynamic dispatch. Games written for chip8 platform can be played and enjoyed in android using this emulator. Source available ***here*** .
- **Asynchronous Web-Server and Client:** Asynchronous multi-threaded Web Server using BOOST ASIO, C++17 and CMake which serves multiple clients simultaneously. The server compiles and runs c++ code provided over http request and returns the result. The Web-Server required using CRTP (Curiously recurring template pattern) where a templated object keeps reference to itself to increase the life time of itself (scenarios like, socket waiting to serve a client), asynchronous programming, lambda functions and smart pointers with other language fundamentals. I hosted the server on heroku in a docker container. Source is available ***here***.
- **Modernization of WinBGIm library:** Improved WinBGIm graphics library targeting GCC 64bit compiler for windows. The last release build by Borland was around 20 years ago and thus is now incompatible with 64bit MinGW compiler. I gave a new build. Also integrated this into a project with CMake support which is easily usable and thus helpful for beginners in graphics as it requires minimal setup. Source is available ***here***.

Interests

Working at the hardware software interface, firmware and HAL development. High Performance Computing, Concurrent & Parallel Systems, Compilers & Programming Languages, OOP & FP in C++, template meta-programming, Rust.
Also interested in Mechanized Program Verification, Functional Programming, and Category Theory due to increasing safety requirements and complexity in software.

Skills

Programming	C/C++, Haskell
ISA	RISC-V, X86, ARM
Testing	Catch2, Google Test
Domains	LLVM, MLIR, FreeRTOS
Tools	CMake, Gradle, Bazel, GDB, Git, vim, tmux, LaTeX
Soft Skills	Problem-solving, Time Management, Teamwork, Documentation

Languages

English	Professional proficiency
Bengali	Native proficiency
Hindi	Bilingual proficiency
Spanish	B1 Intermediate proficiency