Lu Jiongjia(Avimitin)

avimitin@gmail.com · https://github.com/Avimitin · https://crates.io/users/avimitin · https://sh1mar.in

Work Experience

Arch Linux RISC-V, Remote Mar, 2022 – May, 2023

ISCAS, Package Maintainer

- Actively contributed to the Arch Linux RISC-V project, submitting 189 pull requests (PRs) to enhance the software ecosystem for the RISC-V platform.
- Testing and analyzing for Arch Linux RISC-V, strengthening and expanding its infrastructure while establishing automated build and test environments to improve system stability.
- Proactively championed upstream submission of RISC-V platform-related modifications, fostering close communication with the community to drive widespread support for the RISC-V ecosystem.

Buddy MLIR, Remote May, 2023 – May, 2024

ISCAS, RVV Dialect and Infra maintainer

- Engineered and deployed a comprehensive RISC-V Vector (RVV) infrastructure for PyTorch, enabling end-to-end validation on baremetal RVV hardware.
- Leveraged Buddy Compiler infrastructure to successfully deploy and execute the Bert ML model on a baremetal RVV machine, showcasing Buddy MLIR capabilities for supporting multiple backend tasks.
- Contribute comprehensive documentation for MLIR Sparse Tensor implementation, including illustrative examples, to enhance developer understanding and facilitate broader adoption.
- Developed and maintained the build system infrastructure, ensuring seamless integration of new tools and libraries, and reproducibility.

Wafer Compiler, Remote May, 2023 – Dec, 2024

ISCAS, Cryptography Dialect developer

- Engineered the MD5, SM3 MLIR crypto Dialect, providing high level abstraction of crypto implementation for users.
- Develop the Wafer Compiler to provide simutaneous execution of multiple crypto operation on CPU/GPU.

Chip Design and Simulation, Remote Jan, 2025 – Present

HNIIC, Hardware Enginner

- Developed and maintained a formal RISC-V ISA model by creating a code generation framework in Sail and authoring key specifications.
- Engineered a comprehensive verification environment by implementing a Rust-based simulation framework for co-simulation and a differential testing framework to ensure model accuracy against a golden reference.

Related Projects

chipsalliance/t1, RISC-V Vector implementation chipsalliance/t1

- CI/CD Infrastructure Management: Led development and maintenance of the CI/CD infrastructure for Project T1, ensuring seamless service deployment, server configuration, and ongoing system optimization.
- Simulation Test Automation: Spearheaded the creation of a simulation test automation framework for Project T1, driving daily regression testing and backend verification to ensure the RTL reliability.
- Performance Analysis Framework: Engineered the probe-based TestBench module for Project T1, drafting the performance analysis and inspection framework to identify bottlenecks and optimize compiler efficiency.
- Rocket Core Development: Engineered the rocket core DPI and Verilator emulator, establishing foundational support for integrating Project T1 with the Rocket Chip.
- Compiler Infrastructure: Led the design and implementation of basic system library for Project T1, enabling the execution of vectorized workloads and expanding the T1 emulator capabilities.
- Authored and maintained RISC-V ISA specifications by developing a code generation framework that translated opcodes into a formal hardware model with the Sail language.

- Engineered a simulator framework in Rust to enable seamless interaction and co-simulation between software testbenches and the Sail RISC-V hardware model.
- Implemented a differential testing framework to validate hardware model correctness by comparing processor states against a golden reference model (Spike), significantly enhancing verification accuracy.

Skills

- Programming Languages: multilingual (not limited to any specific language), especially experienced in **Rust Scala Ruby Lua TypeScript Nix**, comfortable with OCaml Haskell Zig C++ (in random order).
- Experience Neovim user, have full experience on Neovim plugin development and development environment deployment.
- Linux development: experienced Linux distro user and developer, Know how Linux distro works and know what is the best practice for maintaining Linux infrastructure.
- Rust developement: familiar with Rust development, know some implementation detailed like the Async Coroutine, Higher Kinded Types
- Nix developement: experienced with Nix packaging and reproducible build. Famililar with Nixpkgs library and packages. Daily driven project with Nix flakes
- React Development: familiar with how to start and implement a typical CSR or SSR frontend project, with React and Rust as the tech stack.
- Tools: editor-agnostic, have experience with team tools like GitHub, Gitee, Slack, GitLab and more.