Web Based Simulator of Superscalar RISC-V Processors

Jiri Jaros
Faculty of Information Technology
Brno University of Technology
Brno, Czech Republic
0000-0002-0087-8804

Michal Majer
Faculty of Information Technology
Brno University of Technology
Brno, Czech Republic
misa@majer.cz

Jakub Horky
Faculty of Information Technology
Brno University of Technology
Brno, Czech Republic
horkykuba@email.cz

Jan Vavra
Faculty of Information Technology
Brno University of Technology
Brno, Czech Republic
jv369@seznam.cz

***Abstract*—Mastering computational architectures is essential for developing fast and power-efficient programs. Our advanced simulator empowers both IT students and professionals to grasp the fundamentals of superscalar processors and HW-SW co-design. With customizable processor architecture, full C compiler support, and detailed performance statistics, this tool offers a comprehensive learning experience.**

***Index Terms*—Processor Simulator, Superscalar Processor, RISC-V, HW-SW Co-design, Web Application.**

I. Introduction

In the rapidly evolving field of computer architecture, a deep understanding of superscalar processors is crucial for both IT students and professionals, particularly those focusing on writing high-performance and power-efficient code. However, mastering the intricacies of these architectures is challenging, especially when existing educational tools fall short. Current simulators are often either too complex, lacking intuitive graphical interfaces, or they do not support essential features such as out-of-order execution [2], [3], [4], processor customization [1], memory and cache hierarchy, or detailed performance measurement.

The primary objective of our web-based simulator is to bridge this educational gap by providing IT students with an accessible and illustrative tool to explore and understand the architecture of superscalar RISC-V processors. The simulator is designed to visually demonstrate each phase an instruction undergoes within the processor pipeline, allowing students to identify potential bottlenecks and understand how different implementations of the same algorithm can impact execution time and power consumption.

By interacting with the simulator, students can experiment with different processor configurations and observe how various code optimizations influence performance. This hands-on approach aims to equip students with the knowledge and skills needed to answer critical questions: Given an algorithm, how should one design a processor and optimize the code for the best performance and smallest power consumption? By offering a user-friendly interface and comprehensive support for customization and performance analysis, our simulator seeks to enhance the learning experience and prepare students for the challenges of modern computing.

II. Proposed Solution and Key Features

To address the challenges associated with understanding and teaching superscalar RISC-V processors, we have developed a comprehensive web-based platform-independent simulator. Recognizing the need for both interactive and automated analysis, the simulator also includes a Java-based command-line interface (CLI) that allows for the benchmarking of large programs in a batch processing manner, catering to advanced users who require more extensive testing capabilities.

Key Features of the Proposed Simulator:

- **User-Friendly Interface:** The simulator features an intuitive web interface that visually presents each block and instruction in the processor pipeline. It includes comprehensive documentation and tutorials, making it accessible for students and educators alike.
- **Fully Configurable Processors:** Users can customize various processor parameters, including issue width, size of register fields, reorder, load/store buffers, branch predictors implementations, number of functional units, supported operations and corresponding delays. The simulator also allows for detailed configuration of cache memory settings such as size, associativity, cache line size, and replacement strategy. This flexibility enables users to explore different processor designs and understand their impact on performance and power consumption.
- **Forward and Backward Simulation:** The simulator supports both forward and backward instruction simulation, allowing users to step through the execution process in either direction. This feature aids in understanding

This work was supported by Brno University of Technology under project FIT-S-23-8141.

the flow of instructions and the effects of architectural decisions on execution.

- **GCC Compiler Interface:** Integrated with the GCC compiler, the simulator enables users to compile C code into assembly, offering various optimization levels. The interface includes syntax highlighting and links between C and assembly code, helping users understand how different coding strategies impact low-level operations.
- **Comprehensive Performance Statistics:** The simulator provides detailed performance metrics such as FLOPs, IPC, branch prediction accuracy, functional unit utilization, and cache hit rates. These metrics help users identify bottlenecks and optimize their code for better performance and efficiency.
- **Benchmark CLI**: For more advanced users, the simulator includes a command-line interface that allows for the benchmarking of complex programs in an automated, batch-processing manner.
- **Open Source**: The simulator's source code is available on GitHub at https://github.com/Sekky61/riscv-sim, encouraging collaboration and allowing users to modify and extend the tool according to their needs.

III. Evaluation

To validate the robustness of the code, extensive static unit testing was performed, achieving an 83% code coverage. This high level of coverage indicates that the majority of the codebase has been thoroughly tested, reducing the likelihood of bugs and ensuring stable operation.

In terms of performance, the simulator has been tested under dynamic conditions, supporting up to 100 concurrent users with a median latency of under 1.2 seconds. These tests were conducted on an Intel i5-8300 laptop with 16GB of RAM, demonstrating the simulator's ability to handle multiple users efficiently even on standard hardware.

IV. Conclusions

The web-based simulator for superscalar RISC-V processors represents a significant contribution to the field of computer architecture education and research. By providing an accessible, interactive platform, it enables a deeper understanding of complex processor architectures and fosters experimentation and innovation. Future work will focus on expanding the simulator's capabilities, including the addition of more advanced features such as vector units.

References

[1] M. B. Petersen, “Ripes Introduction”, 2022, available form: https://github.com/mortbopet/Ripes/blob/master/docs/introduction.md.

[2] H. Rajaei, “An Empirical Study for Multi-Level Cache Associativity”, in 2020 ASEE Virtual Annual Conference Content Access. 2020, no. 31008. DOI:10.18260/1-2–34115.

[3] S. Metzlaff, A. Vogelgsang, and N.Krezic-Luger, “DLX MIPS processor simulator”, University of Augsburg, 2013, available from: https://github.com/smetzlaff/openDLX.

[4] A. Castellanos, “Jupiter, RISC-V Assembler and Runtime Simulator”, 2022, available form: https://github.com/andrescv/Jupiter.