Development of Teaching Computer Application

Mohammed Mahmudul Hasan

February 25, 2026

1 Problem statement

Teaching the internal working principles of a computer processor to first-semester students is a demanding task [2]. Modern processors are highly optimized and abstract away many fundamental mechanisms, making it difficult for beginners to understand how simple assembly programs are executed step by step [6]. Teaching computers and processor simulators aim to reduce this complexity by providing simplified models that expose internal states and execution processes [4]. However, many existing solutions are either based on outdated technology or lack the flexibility required to explore more complex execution scenarios [7].

At Technische Universität Chemnitz, the Lehrcomputer 1 (LC1) has traditionally been used to introduce students to basic processor concepts. While it serves as a valuable educational tool, its simulator is limited in terms of memory size, execution flexibility, and extensibility. Similar constraints can also be observed in other teaching processor models, which often focus on fixed architectures and predefined behavior, making it difficult to adapt or extend them for deeper architectural exploration [3].

The objective of my part of this thesis is to design and implement a processor architecture and control simulation as part of a modern, web-based Teaching Computer Application. The central problem addressed here is how to model a processor in a way that clearly represents instruction execution while remaining flexible, extensible, and suitable for educational use [7].

A key challenge is bridging the abstraction gap between low-level digital components and high-level program execution. The processor simulation must integrate registers, memory, arithmetic logic units, and buses into a coherent model that reflects how instructions are processed internally. At the same time, the instruction cycle—covering instruction fetch, decoding, execution, memory access, and write-back—must be represented explicitly and executed step by step [1]. This allows students to observe how individual operations and control decisions contribute to the overall behavior of the processor.

Another important aspect is the simulation of timing and control behavior. Instead of treating instruction execution as an instantaneous process, the simulation should reflect the temporal progression of states based on the behavior of underlying components [5]. This is essential for helping students understand how complex instruction execution emerges from simpler combinational and sequential mechanisms controlled by a control unit.

Finally, the processor simulation must provide structured access to its internal state, including register contents, memory values, current instructions, and control signals. Well-defined interfaces are required to support visualization and interaction layers while allowing the processor model to evolve independently [3]. Addressing these challenges forms the foundation for a flexible and pedagogically effective teaching computer that supports deeper understanding of processor architecture.

References

[1] John D Carpinelli. "The relatively simple CPU simulator". In: *Computers in Education Journal* 12.2 (2002), pp. 20–26.

[2] Jean-Luc Dekeyser and A Shadi Aljendi. "Adopting new learning strategies for computer architecture in higher education: case study: building the S3 microprocessor in 24 hours". In: *Proceedings of the Workshop on Computer Architecture Education*. 2015, pp. 1–8.

[3] Jovan Djordjevic, Bosko Nikolic, and Aleksandar Milenkovic. "Flexible web-based educational system for teaching computer architecture and organization". In: *IEEE Transactions on Education* 48.2 (2005), pp. 264–273.

[4] Sabah Al-Fedaghi. "Conceptual Modeling for Computer Organization and Architecture". In: *arXiv preprint arXiv:2103.01773* (2021).

[5] Roberto Giorgi and Gianfranco Mariotti. "WebRISC-V: A 64-bit RISC-V Pipeline Simulator for Computer Architecture Classes". In: *arXiv preprint arXiv:2504.03722* (2025).

[6] Nathan Gober et al. "The championship simulator: Architectural simulation for education and competition". In: *arXiv preprint arXiv:2210.14324* (2022).

[7] A Yavuz Oruc et al. "CodeAPeel: An Integrated and Layered Learning Technology For Computer Architecture Courses". In: *arXiv preprint arXiv:2104.09502* (2021).