Abstract

In education of Computer Architecture, microprogramming is a challenging topic, as many microarchitectures are not publicly disclosed, and those used in education are often outdated or too complex. This Bachelor's thesis aims to rework a microprogramming exercise unit for higher education. A microarchitecture is proposed, which can be used as an introduction to this topic and can usually be understood in a 90-minute tutorial. Furthermore, a simulator is developed, which implements this architecture, visualizes the relevant concepts, and provides testability for the students' microprograms.
In this study, the architecture, the deployment of the simulator, and the tutorial are evaluated. Additionally, the effectiveness of an exploratory format is tested, which de-emphasizes extended theoretical introductions and focuses on practical exercises, fostering an environment where students have to engage more actively with the tasks.

Data was collected from 154 undergraduate computer science students who had attended the respective practical exercise. 92% stated that they had understood why microprogramming is used, and the architecture was found to be intuitive by 90%, which means that the architecture is suitable for the purpose. The simulator enabled the students to perform better at writing microprograms, both in the tutorial and the graded homework. Students who used the simulator scored 15.8% more points in the homework, and 36.2% more of these students achieved full marks compared to those who did not use a simulator. In the final exam, a better score was observed as well. The instructor workload of grading simulator submission has also decreased by 75%. Through testability, more effective practical exercise time is enabled. By using an exploratory approach, the subjective feedback of the students has greatly improved.

The results of this work find a largely positive impact on both students and instructors, created by testability and a deeper involvement of the students.