Teaching the Von-Neumann Model with a Simulator

Martin Weinert, Jan Hendrik Krone, and Johannes Fischer

Technische Universität Dortmund, 44227 Dortmund, Germany
{martin.weinert,hendrik.krone,johannes.fischer}@cs.tu-dortmund.de

Abstract. An important goal of computer science education is the demystification of computing machinery. The Von-Neumann-model can help to achieve this goal, since it demonstrates how programs are executed.
Therefore we developed a simulator and an accompanying series of lessons on the Von-Neumann-model. Such teaching series are already being used in practice, but are often separated from related computer science topics. We connected our lessons to the concepts of sequential logic systems, (higher) programming languages and the IPO model.

Keywords: Von-Neumann-Model · IPO · Series of Lessons

1 Introduction

One of the core goals of computer science education is the demystification of computing machinery. The Von-Neumann-model contributes to this [7,6] by providing a simple, yet precise description of a computer, which allows students to understand how computers operate.

Because of the complexity of the topic, it is important to apply some simplifications through the teaching material. It is a common idea to achieve this by using simplified simulators. Such simulators are already in use. Examples are Johnny [1,2] and MOPS [4] for use in schools or Microsim for university education. The simulators were developed with several design principles in mind. MOPS, for example, used the IPO model (Input-Processing-Output) as an orientation [4]. Although the simulators are simplified and put emphasis on different aspects, they still have difficulties to effectively teach the Von-Neumann-cycle [3]. In order to address these difficulties and to be able to focus program execution as a teaching topic, we decided to develop and use a new simulator.

2 Simulator KUR2

An important design decision of the simulator is the arrangement of the components. In contrast to other simulators like Johnny, MOPS or Microsim, we consciously separated the input and output components and arranged them in a way that resembles the IPO model. The user enters programs in machine code,

J.-P. Pellet and G. Parriaux (Eds.): ISSEP 2023 Local Proceedings, pp. 221–224, 2023.
https://doi.org/10.5281/zenodo.8432030

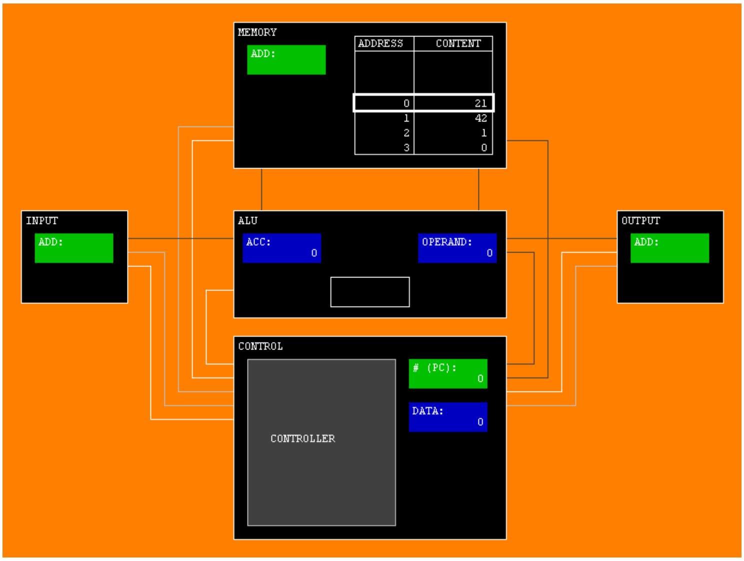


Fig. 1: Screenshot of the Simulator

which are then stored in the memory unit and executed afterwards. Due to the presentation of the internal registers, the flow of data can be followed.

We emphasized a very simplistic design, because we wanted to integrate the simulator into a story about John von Neumann. Therefore we did not want to include modern design elements into the graphical representation.

3 Lessons series and materials

In this section, we describe the lessons that we developed. They can be accessed on the authors' website [5] for further examination and usage.

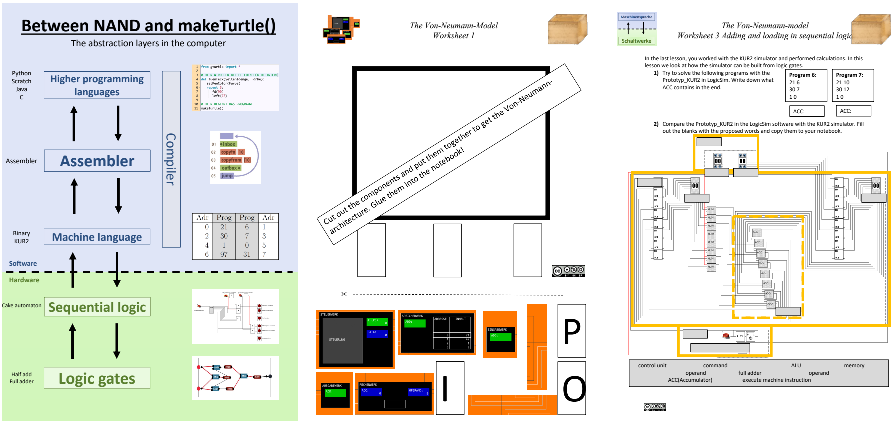
3.1 Series of Lesson

The series of lessons is composed of six lessons:

Lesson 1: The students learn how the simulator is structured by puzzling together its pieces on a worksheet.

Lesson 2: The students interact with the simulator for the first time. They simulate first programs and learn how to write simple programs for basic calculations.

Lesson 3: The students see how the functions of the simulator can be implemented with sequential logic. They interact with a logic simulator, which shows how some of the functions of KUR2 can be performed with logic gates. They learn how sequential logic and the Von-Neumann-model are connected.



(a) Poster (b) Worksheet IPO-VNM (c) Sequential logic

Fig. 2: Materials used in the lessons

Lesson 4: The students learn about jumps. First they formulate that programs are stored as a linear sequence of commands. Then they see that execution can deviate from this sequence through (conditional) jumps. At the end of the lesson the students write down these insights as rules of the von-Neumann-model.

Lesson 5: The students see that programs and data are stored in the same memory. They also learn how the memory is structured and addressed. These insights are again written down as rules of the model.

Lesson 6: Finally there is a last lesson where the students can choose from three activities. Those activities introduce connections to external (virtual) devices and show how text and pixel displays can be accessed or how a simple guess the number game can be played with the simulator. The activities are intended to teach that the machine is independent of the problem.

3.2 Supporting materials and key features

During the development of the series of lessons some aspects emerged as especially important. The first of those aspects came from the realization that the first contact with the simulator was quite overwhelming for the students. We tried to make it easier for the students by introducing the puzzle (see fig. 2b) in the first lesson, where they just put the simulator together and relate it to the IPO model.

Next we tried to make it easier to see the connections to related topics and build a coherent overall picture. To this end we created a poster (see fig. 2a) that shows the different layers of abstraction and their connections. This poster was presented in the computer lab to be visible at all times. These connections were also emphasized by adding the activity where the students would see the simulator implemented with logic gates (see fig. 2c).

Finally we tried to tie all lessons together with an engaging backstory. The idea is that someone has found a box containing John von Neumanns old belongings. Since the notebook that contains information on the Von-Neumann-model and the simulator KUR2 in partly destroyed and unreadable, the students have to reconstruct its contents through the lessons and worksheets.

4 Future work

Our observations during the lessons indicate that the connection of the Von-Neumann-model to underlying topics like sequential logic and the IPO model, and to emphasize how it fits into the broader context might be substantial for effective teaching. We will investigate how this affects the development of mental models of computing machinery in the future.

The aforementioned connection was mainly established with the poster that shows the connections between the Von-Neumann-model and other topics, as well as the simulator itself. We will look for additional ways to improve the connections.

All of the materials and the simulator are free to use for educational purposes. The materials and simulator can be accessed on the authors' website [5]. The simulator is implemented in Java and a JavaScript version is currently in development. The source code can be accessed via GitHub [8].

References

1. Dauscher, P.: Johnny - A Simulator of a Simple von-Neumann Computer (2012), https://sourceforge.net/projects/johnnysimulator/
2. Dauscher, P.: Aufbau und Funktionsweise eines Von-Neumann-Rechners - Ein möglicher Unterrichtsgang mit dem Open-Source-Simulator Johnny. LOG IN **33**(2) (2013)
3. Göbel, L., Hellmig, L.: Die Von-Neumann-Prinzipien erleben - Ein enaktiver Zugang zu einem abstrakten Thema. LOG IN **39**(1) (2019)
4. Haase, M.: MOPS Modellrechner mit PseudoAssembler. http://www.viktorianer.de/info/mops.html (2013)
5. Krone, J.H., Weinert, M.: Teaching materials for KUR2 (2023), https://ls11-www.cs.tu-dortmund.de/staff/weinert/kur
6. Lautebach, U.: Vom Gatter zum Compiler: Im Unterricht durch sieben Abtraktionsebenen. In: Gallenbacher, J. (ed.) Informatik allgemeinbildend begreifen. Gesellschaft für Informatik e.V., Bonn (2015)
7. Sorva, J.: Notional machines and introductory programming education. ACM Transactions on Computing Education **13**(2), 1–31 (jun 2013)
8. Weinert, M.: Code repositories for KUR2 (2019), https://github.com/kur2