OVERCOMING THE LIMITATIONS OF TRADITIONAL MEDIA FOR TEACHING MODERN PROCESSOR DESIGN

Peter Marwedel, Birgit Sirocic

Dept. of Computer Science,
University of Dortmund,
44221 Dortmund, Germany
peter.marwedel@udo.edu

ABSTRACT

Understanding modern processors requires a good knowledge of the dynamic behavior of processors. Traditional media like books can be used for describing the dynamic behavior of processors. Visualization of this behavior, however, is impossible, due to the static nature of books. In this paper, we describe a Java-based tool for visualizing the dynamic behavior of hardware structures, called RaVi (abbreviation for the German equivalent of "computer architecture visualization"). Available RaVi components include models of a microcoded MIPS architecture, of a MIPS pipeline, of scoreboarding, Tomasulo's algorithm and the MESI multiprocessor cache protocol. These models were found to be more useful than general simulators in classroom use. The Java-based design also enables Internet-based distance learning. Tools are available at //ls12.cs.uni-dortmund.de/ravi.

1. INTRODUCTION

The presented project aims at facilitating understanding the dynamics of modern processor architectures, thereby overcoming an important limitation of books. Videos tapes and video distribution techniques have made it possible to show non-interactive media elements to students. However, video tapes have to be accepted by teachers and users on an "as-is" basis. It is not possible to use instruction streams other than those employed for the production of the video. Also, it is not possible to modify hardware structures in order to see the effect of hardware changes on the dynamic behavior. In short, videos are very inflexible and cannot provide interactiveness (except the simple type of interactiveness possible with DVDs).

Providing this interactiveness, however, is difficult, since it requires the simulation of hardware structures. This can be a challenging task which cannot be solved within the time-frame available for preparing a course. Why not just use available hardware simulators? These simulators are frequently designed for optimum simulation speed and complex design projects. Ease of use, excellent visualization and portability have normally not been top goals for simulator design. Also, powerful simulators are typically proprietary and come at high costs, preventing their widespread deployment to classrooms and into the hands of students.

Therefore, we tried to design RaVi models for the simulation-based visualization of the dynamic behavior of hardware architectures. In contrast to available models, emphasis is on visualization.

The design of RaVi models was done in the context of the larger SIMBA project[1] [1].

This paper is structured as follows: section 2 describes the different target groups for RaVi models. Section 3 contains some information on the software technology used for implementing RaVi models. The following section contains a list of models developed (and made available) so far. A brief summary of the experiences with using RaVi in classrooms is included in section 5. The final section provides a conclusion.

2. TARGET GROUPS

A key idea for the RaVi project is to provide an added value for existing courses. As far as computer architecture is concerned, the books written by Hennessy and Patterson [2, 3] are used all over the world and constitute de-facto standards for computer architecture education. The goal is to enable the visualization of the dynamic behavior of the processors described in the book such that the additional material can be used in all courses based on the books. The focus is on topics for which understanding the dynamic behavior is essential and not easy.

Lecturers of computer architecture are the first target group for the RaVi project. Lecturers only require limited explanations of the models, since the models reflect the books mentioned as closely as possible. Experienced lecturers might want to modify the models. Simple changes can be incorporated with the schematic editor of the HADES simulation library [4], on which RaVi is based. The set of hardware primitives can be extended by non-standard hardware components. The simulation as well as the display behavior of non-standard hardware components has to be described in Java.

In order to simplify getting of first impression of RaVi without having to install software, RaVi models are available as applets on the Internet. This allows lecturers to quickly find out whether or not these models are appropriate for their classes.

Students make up a second group of potential users. Students require a different kind of explanations. Still, explanations can be kept to a minimum, since the books mentioned are assumed to be available. For this group of users, it is also important to be able to use the models without having to install software.

Using visualization of dynamics, we also want to improve the motivation of students to study processor architectures. In particular, we would like to attract students who are not initially focusing on computer engineering.

[1] We gratefully acknowledge the funding of the SIMBA-project by the German ministry of research and development (BMBF).

3. OBJECT-ORIENTED MODELING

In order to allow RaVi to be available on the Internet and not having to install software, RaVi models need to be implemented in Java. Authoring software does not provide the degree of interactiveness required.

Hendrich, with his design of the HADES system [4], has designed a class library which can serve as the starting point for developing hardware models. HADES comprises a schematic editor and a simulator kernel for simulating hardware structures. It supports the IEEE 1164 logic value system and deterministic simulation of clocked parallel hardware structures. The class library includes classes for common hardware elements. The schematic editor can be used to create instances of these classes. Hence, each hardware component is an instance of the corresponding Java class representing its type. This is a nice example of exploiting object orientation in hardware design. Each instance comes with methods for simulating its behavior as well as with methods for displaying itself on the screen and for showing and modifying its properties. Instances can be connected using drag and drop. For memory models, it is possible to interactively modify memory contents.

By default, HADES uses color to visualize signal values. Unfortunately, in each clock step many signals change their values in the actual architecture. However, many of these are redundant since they are not loaded into any memory element during that clock step. More precisely, many line segments used in the architecture schematics are not used to carry relevant information. In order to avoid confusing changes of the colors of almost all line segments, an algorithm had to be developed which identifies those line segments that carry relevant information. Starting from memory inputs that are enabled for a certain clock step, paths to selected memory outputs are found. Line segments on selected paths are marked for visualization. All other line segments will be displayed with a light color, representing an undefined value. This algorithm facilitates understanding the dynamic behavior of architectures.

Interestingly, the power consumption of real processors could be reduced if wires corresponding to redundant lines segments were gated.

4. MULTIMEDIA COMPONENTS DESIGNED IN RAVI

The following multimedia models are currently available in RaVi:

- *A microcoded version of the MIPS processor architecture.*
- *Several variants of MIPS pipeline:* Using these variants, the effect of bypassing, pipeline stalls and early branch detection can be demonstrated.
- *The MESI protocol for multiprocessor caches:* Models include a simple model for a single cache line as well as a model with several caches lines per cache.
- *Dynamic instruction scheduling:* Score-boarding as well as the Tomasulo algorithm are available as RaVi models.

5. EXPERIENCES

RaVi models have been used in three courses so far. In all three cases, the demonstration of the dynamic behavior was preceeded by a Powerpoint-based explanation of the essentials, including simple Powerpoint animations. In the first course, the results were somewhat mixed since the RaVi models were not well integrated into the lecturer's presentation. For the second course, integration into the course was very much improved and visualization of dynamics was very much appreciated. With some fine-tuning of the models, the level of appreciation was further increased for the third course. Starting with the third course, students started asking for their own copy of the models.

We have also tried to evaluate the effect of using RaVi on the level of understanding reached by the students. Towards this end, students were partitioned into two groups in the first course. One one group of students was exposed to RaVi models, while the second group was not. Unfortunately, other factors influenced the results obtained much more than the use of RaVi components. We plan to repeat this evaluation for the current version of RaVi models since the comments are now much more encouraging. In general, it seems to be extremely difficult to measure the effect of new presentation techniques on the knowledge of the students.

RaVi models have been well received by colleagues attending a demonstration.

Developing stable models that are accepted by the students required more effort than the numbers that are typically mentioned for the design of "standard" multimedia components. Almost two person-years have been spent on the models that are available so far. Additional effort is required for generating all explanations required for the various target groups.

We found that the techniques employed in HADES extend beyond the scope of computer architecture, since the component symbols used for the schematic as well as their behaviors can be freely defined in Java.

6. CONCLUSION

We have developed models for the interactive visualization of the dynamic behavior of processor-based systems. According to our experience, developing these models in Java proved being very useful since this enabled excellent visualization. Internet-based applications are an additional benefit of this approach. After some fine-tuning, RaVi multimedia models were well-received by students and colleagues.

REFERENCES

[1] S. Schubert et al., "SIMBA home page", *http://ddi.cs.uni-dortmund.de/simba*

[2] J. L. Hennessy and D. A. Patterson, "Computer Organization – The Hardware/Software Interface", *Morgan Kaufmann Publishers Inc.*, 2nd ed. 1997

[3] J. L. Hennessy and D. A. Patterson, "Computer Architecture – A Quantative Approach, *Morgan Kaufmann Publishers Inc.*, 3rd ed. 2002

[4] N. Hendrich, "A Java-based Framework for Simulation and Teaching", *Proceedings of the 3rd European Workshop on Microelectronics Education*, EWME 2000, Kluwer Academic Publishers, p. 285-288