

Work in Progress - MythSim: The Mythical Simulator for Real Students

Jason Vroustouris¹ and Mitchell D. Theys²

Abstract - MythSim is a cross-platform control-code simulator being used at the University of Illinois at Chicago. This mythical 8-bit processor gives students experience with concepts in computer architecture. MythSim has been redesigned from the ground up based on the changing needs of instructors and students. An improved multi-windowed interface, color-coding system and reference card allows students to learn the architecture quickly so they can focus on developing and debugging their programs. To facilitate future development, the source code is hosted online in a distributed development environment. Feedback from instructors and students has been positive. This paper presents improvements made in the new version, feedback from students and the support we provide for instructors.

Index Terms - computer architecture, computer science, curriculum, java, microcode, simulator.

OVERVIEW

It is becoming easier for students to write programs without knowing much about the hardware that is used to execute their software. Modern languages are built with many layers of abstraction that shield hardware details from the programmer. In addition, the trend has been to automate the programming process with IDE's that have integrated help systems, code completion, and automatic compilation utilities. As a result, students are unable to easily integrate a low level concept like control code programming into their understanding of the computer.

MythSim [1] is an open-source control-code simulator designed by students and instructors at the University of Illinois at Chicago (Figure 1). The simulator is a central part of the undergraduate computer architecture curriculum in the Computer Science and Electrical and Computer Engineering departments. In the Computer Science Department the simulator is used in a two-semester sequence that covers digital logic, datapath design and assembly programming. As a final project the students' implement a simple assembly language in microcode. By writing their own control code they gain a fundamental understanding of how the computer works.

IMPROVEMENTS TO MYTHSIM

Many improvements to the original simulator were made based on feedback from students and the instructors running the courses. The most notable of these are the multi-window display, color-coding system, and reference card.

The multi-window display accommodates a text editor and allows for expansion. The original one-window simulator [2, 3] did not leave room for a text editor. Now the simulator windows can fit around a text editor as seen in Figure 1. The one-window design also made it difficult to add new components without cluttering the interface. Now students can create plug-in windows that can be displayed and hidden as needed.

The color-coding system gives a different color to the different numeric values in the simulator. Current microcode values are shown in red, register values are shown in blue, and memory values are shown in green. These colors subtly reinforce connections to related information across the sub-windows.

The reference card was created to aid students while programming. One side has a list of statements in the microcode language and their meaning. The other side has a complete diagram of the datapath and control unit. Students can clearly see the relationship between microcode and hardware. They can quickly start interpreting example code and writing their own code.

FEEDBACK FROM STUDENTS

The feedback received from students showcases the effects MythSim has had on the learning experience. One student liked the ability to see the state of values at any time in a program. After using MythSim they started using a visual debugger when doing their programming in C++. Other feedback involved the ease of portability of the new simulator. One student was proud to have been carrying around MythSim and his data files on a thumb drive. This allowed him to seamlessly move from his computer at home to computers in the school labs and always have the most up-to-date version of his code at his fingertips.

Some of the most important feedback came when problems occurred. For instance, after the class was given a major assignment, a flaw was found in the simulator display. This was a problem for students who were getting their first impression of the simulator. Fortunately we had created a

¹ Jason Vroustouris, University of Illinois at Chicago, Department of Computer Science, jasonv@jasonv.com

² Mitchell D. Theys, University of Illinois at Chicago, Department of Computer Science, mtheys@uic.edu.

discussion board where students could post the bugs found in the software. The postings on the board had already stated the nature of the problem and a workaround to help solve the problem. Using a closed-source application would not have allowed such a quick resolution of the problem. In this case the students were able to solve the problem themselves.

SUPPORT FOR INSTRUCTORS

The MythSim design team is committed to supporting instructors who use the simulator. The instructors provide important feedback that helps guide the development process. Whether reporting bugs, writing plug-ins or getting questions answered, instructors are encouraged to visit our website and join our community.

Since MythSim is used in courses at The University of Illinois at Chicago it has been tested in a real learning environment. We know that instructors spend considerable time creating lectures, assignments, and exam questions that are based on this type of software. When software changes, we know that the instructor may have to quickly learn the new software, train the teaching assistants, create new tutorials, exam questions, and homework problems [4]. To insure that the instructor has more control of the content they are presenting, MythSim is distributed under an open source

license [5]. This license ensures that the version a curriculum is built around will always be available.

CONCLUSION

For more information on MythSim please visit our website <http://www.mythsim.org/> [6]. The application, source code, and documentation are available for download. These files are also hosted on SourceForge [7] to provide mirrors of the data, and enable bug tracking and distributed development.

REFERENCES

- [1] Jason Vroustouris; "MythSim: The Mythical Simulator for Real Students", Undergraduate Research Project, UIC, 2004
- [2] Rolf Hettelsater, "MythSim: A Detailed Computer Architecture Simulator", MS project report, UIC, 1997
- [3] Jon Solworth, "Computer Architecture", manuscript in progress, 2003
- [4] Mitchell D. Theys and Patrick A. Troy, "Lessons learned from teaching computer architecture to computer science students," Frontiers in Education 2003, November 2003.
- [5] GNU General Public License - <http://www.gnu.org/licenses/gpl.txt>
- [6] MythSim Website - <http://www.mythsim.org/>
- [7] Source Forge - <http://www.sourceforge.net/>

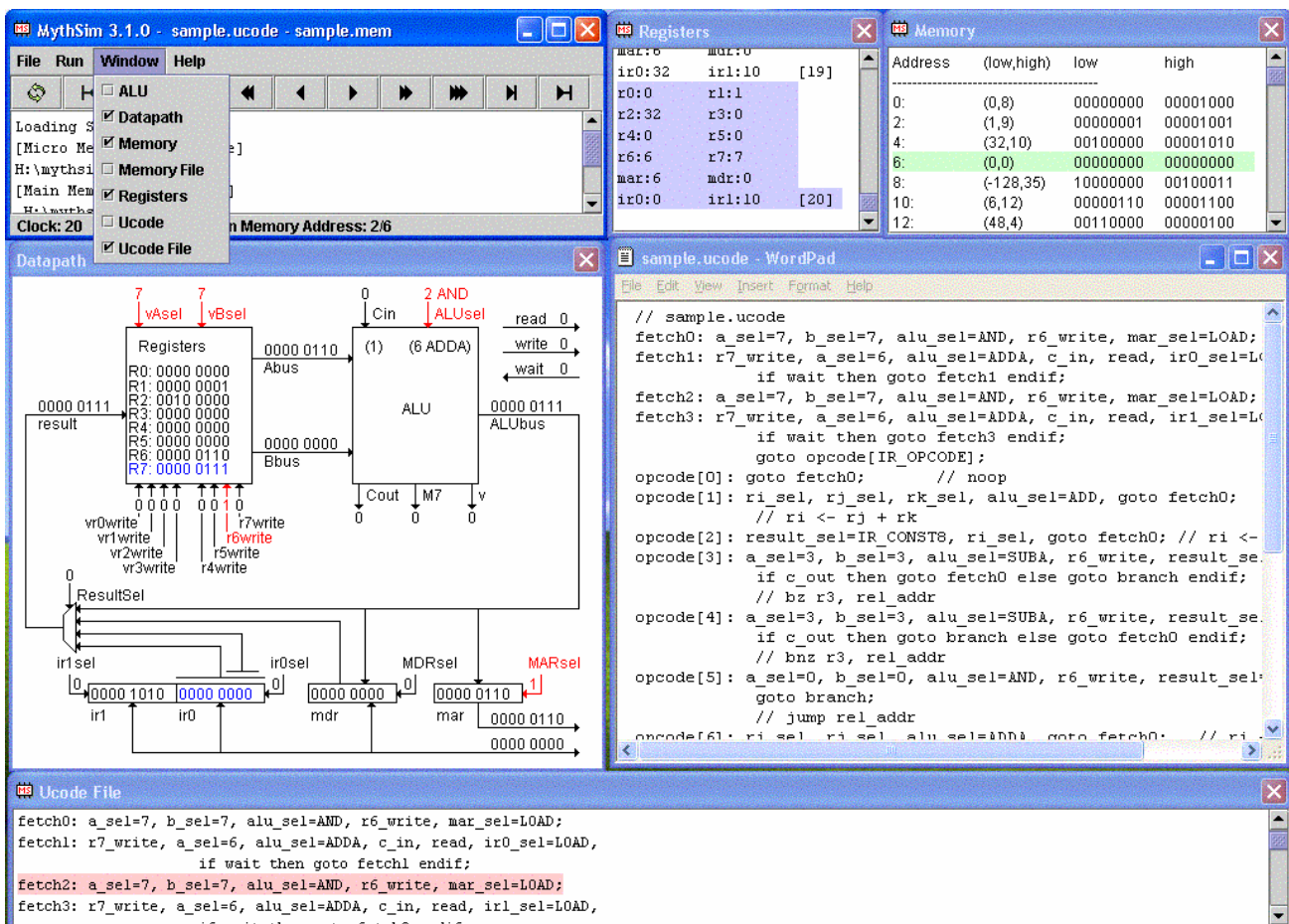


FIGURE 1
Debugging With MythSim and WordPad on Windows