Special Issue on General Computer Architecture Simulators

*Computer Architecture Simulators Are Not as Good as the Real Thing
– They Are Better!*

Welcome to this special issue of the *Journal of Educational Resources in Computing* (JERIC). This issue is focused on the topic of *general* computer architecture simulators with both the terms “computer architecture” and “simulation” interpreted in their broadest sense. The simulators in this issue are designed as teaching tools; but there is an equivalently rich field of computer architecture simulators designed for research purposes. The next issue of this journal will be another special issue on the closely related topic of *specialized* computer architecture simulators for educational purposes. JERIC is the premiere electronic journal in the area of computer science education, and as such provides the novel capability for downloading both the papers and the corresponding simulation software in a manner that, I believe, advances the forefront of academic publication.

The genesis for dedicating back-to-back special issues on this topic was an ITiCSE 2000 working group which studied the challenges in teaching computer architecture. One of this working group’s findings was the survey result that confirmed that many CS instructors who teach computer architecture topics have low confidence in their own understanding of the material. This is completely understandable, given that the scope of computer architecture stretches across a wide range of topics (digital logic to computer organization, to assembly language, to memory hierarchy, to parallel processing, and beyond) and requires a disparate skill set (discrete mathematics, programming, design, electronics). For the complete report, see Cassel et al. [2001].

Simulators, the most important development in the teaching of computer architecture, came to the rescue. Simulators provide educators with a powerful tool to communicate complex systems to students by visually displaying the organization and underlying simultaneous events during program execution.

The title of this guest editorial makes a bold assertion. But how can a simulation be better than the real thing? Consider these five points:

- 1) Simulators can highlight important general concepts while eliminating the unimportant proprietary details of real machines.
- 2) Simulators can operate at multiple levels of abstraction, from digital logic to microcode, to assembly language, to operating systems, in order to support integrated bottom-to-top or top-to-bottom approaches (depth) that are feasible in only the most well-equipped laboratories of real machines, since completely different types of test equipment are required.
- 3) Simulators are available with minimal resources, many are freely accessible via the Internet. Laboratories for real machines are resource intensive, and equipment is easily outdated due to its ever-shortening lifecycle.

- 4) User-friendly simulators are available for a wide range of machines (from hypothetical machines to extinct machines, to real advanced machines), exposing students to many different computer architectures (breadth), while gaining competence on a single real machine may take a significant part of one course.
- 5) Simulators facilitate both active learning (students program, debug, and execute their own system software) and asynchronous learning (simulators are available at any time to anyone with access to a computer).

As evidence for these five points, this special issue provides an overview of the state-of-the-art in general computer architecture simulators for educational purposes. I have ordered the articles in this issue approximately from novice to intermediate complexity. The articles range from hypothetical machine simulators (Knob & Switch, Little Man Computer, OAMulator, PIPPIN) to real machine simulators (MIPS/RTLSim, SPARC/ALFA-1, x86/EasyCPU) to simulators appropriate for CS0 (Knob & Switch, Little Man Computer, OAMulator, PIPPIN), to simulators adaptable to CS1 (Postroom Computer) to a simulator specifically designed for architecture design (CPU Sim).

This issue begins with PIPPIN, a widely used visual simulator associated with a popular textbook. This hypothetical CPU simulator is integrated with a language translator, Rosetta, to provide an integrated top-down view of machine operation. Very much like PIPPIN, the OAMulator also provides the low-level functionality of one-address assembly language programming integrated with a compiler. The Knob & Switch Computer simulator is notable for (1) its use of “cognitive hooks” to facilitate active learning and (2) its modular design, which enables incremental exposure to different levels of machine abstraction. The creator of CPU Sim has a decade of experience with his award-winning software. CPU Sim has continued to evolve through different versions into a unique simulator that allows students to design and test their own architectures.

The article “Three Simulator Tools for Teaching Computer Architecture...” shows the educational range of several representative computer architecture simulators from K12 (EasyCPU), to undergraduate CS0 (Little Man Computer), to advanced undergraduate and graduate studies (RTLSim). The Postroom Computer, which is also based on the Little Man Computer paradigm, is a simulator notable for its flexibility: it can be used to model an extensible range of instruction set architectures and memory models. The final article introduces ALFA-1, based on the SPARC architecture, with a DEVS simulation environment that allows students to comprehend machine behavior by building and analyzing performance through different layers of abstraction.

An underlying thread running through all these articles is a global consensus on the use of computer architecture simulators for educational purposes. While the articles accepted for this issue and the next one are more USA-centric than I would like, we did have the worldwide participation of authors whose home institutions are located in Argentina (3), Spain (3), Germany (2), New Zealand (2), Canada (1), Israel (1), and the United Kingdom (1). It should also be noted that a significant percentage (~35%) of the peer reviewers were from home institutions outside of the USA.

Lastly, it has been my privilege to serve as guest editor for this issue. It has been a delightful experience, and I thank JERIC and ACM for giving me the opportunity. Of course, any journal issue such as this one is really the result of the efforts of many people. I count about 60 people, including the JERIC co-editors-in-chief, Lillian (Boots)

Cassel and Ed Fox, Deborah Knox of the Computer Science Teaching Center <<http://www.cstc.org/>>, the authors, peer-reviewers, and Jono Hardjowirogo of ACM Press. There were also many fine papers and software simulators submitted to this issue, which, unfortunately, we could not include here. It was a difficult selection process, and we both thank and encourage the authors of papers that are not included in this issue to continue their work and resubmit to future issues of JERiC. After releasing an international call-for-papers, we received 33 submissions and accepted the 13 paper/simulator pairs in this issue and the issue to follow. After an initial cut, each submission that met the call-for-papers requirements received at least three independent peer reviews.

I hope you enjoy what we have assembled and that it inspires many ideas for teaching and research!

WILLIAM (BILL) YURCIK
Illinois Wesleyan University
February 2002

CASSEL, L., KUMAR, D., BOULDING, K., DAVIES, J., HOLLIDAY, M., IMPAGLIAZZO, J., PEARSON, M., WOLFFE, G. S., AND YURCIK, W. 2001. Distributed expertise for teaching computer organization & architecture. *ACM SIGCSE Bull.* 33, 2, 111-126.