

Extending parallel programming education beyond the von Neumann architecture

The design of a high-performance educational simulator *aBCXsim* for memristor crossbars

Nikhil S. Shekhawat*, Sumit Kumar Jha[†]

Department of Electrical Engineering and Computer Science
University of Central Florida, Orlando, FL

Email: *nikhil@eecs.ucf.edu, [†]jha@eecs.ucf.edu

Abstract—Inspired by the lower energy-footprint of biological intelligence, traditional John von Neumann CPU and GPGPU high-performance computing is slowly being replaced by new parallel emerging computing architectures. Biological neurons are now known to be memristors, and hence, memristor-based neuromorphic computing is a particularly promising method for designing bio-inspired high-performance computational systems. There is currently no educational modeling and validation platform for exploring the design space of such memristive high-performance computing architectures.

In this paper, we present the design of an educational high-performance software infrastructure that will facilitate the design of networks of memristor-based nanoscale crossbars as high-performance computational systems. The simulator will enable students to analyze their own problem-specific designs of memristive systems, and train them in application-specific high-performance computing design.

Keywords—educational simulator; memristor; neuromorphic; GPGPUs; parallel programming.

I. INTRODUCTION

While tremendous strides have been made in the development of parallel algorithms and tools for numerical computing, the bulk of computer software continues to employ sequential implementations. As the clock-speed of CMOS processors approaches an energy-driven asymptotic limit, there is a necessity for the design of emerging non-CMOS inherently-parallel computer architectures that address specific custom applications.

University of Central Florida, besides being one of the largest universities in the country in terms of student population, is also home to the NVIDIA CUDA Teaching Center and the NSF/IEEE TCPP Early Adopter Program. Through these initiatives, traditional parallel programming models such as those based on message-passing and multi-core programming have been embedded into our algorithms and programming classes. However, the demise of Moore’s law requires that a holistic approach to parallel programming education include the design of high-performance parallel computing architectures. Unfortunately, it becomes difficult to teach the design of such parallel computing systems on emerging architectures without access to an educational simulator that can be used to rapidly prototype and evaluate candidate designs.

To the best of our knowledge, there is no existing educational software infrastructure for simulating memristive computing architectures. Currently, the most popular method of simulating memristor designs is by using general-purpose circuit simulators, such as SPICE and its variants. In our experience with senior undergraduates and first-year graduate students, it is not feasible to explore new parallel computer architectures using such a low-level representation. Further, traditional open-source simulators do not provide a simple mechanism to explore the impact of stochastic noise on the behavior of such circuits.

In this paper, we introduce the design of a new educational memristor crossbar simulator *aBCXsim* (a Bose-Chua crossbar simulator). We describe how our design choices have been motivated by our experience in teaching high-performance memristive computing to senior undergraduate and first-year graduate students.

II. *aBCXsim* AND HIGH-PERFORMANCE CROSSBAR COMPUTING

In 2008, HP Labs demonstrated a nanoscale memristor reaffirming a theoretical prediction made by Leon Chua about 40 years earlier [1], [2]. Based on symmetry arguments, Chua had suggested a new passive, two-terminal circuit element that would relate the two fundamental quantities of electric charge and magnetic flux linkage. It has since been shown that the British Indian polymath Sir J C Bose had built and recorded one quadrant of the pinched hysteresis loop of a two-terminal iron coherer in the early 20th century - the tell-tale sign of a memristor. In honor of the great experimentalist Bose and the brilliant theoretician Chua, we name the educational simulator as a Bose-Chua (cross) X-bar simulator or *aBCXsim*.

A fundamental property of a memristor is its ability to remember the amount of current previously flowing through it before being turned off [2]. Hence, it is not surprising that memristors are already being used to design high-density non-volatile storage devices [3]. This has been followed by a flurry of publications in the scientific and popular press investigating and analyzing the design, theory, and application of memristors and memristor-based computational systems [4]. Memristors have been used for performing arithmetic

operations [5], implementing logical operators [6], computer vision and image processing [7] among other applications [4].

The *aBCXsim* simulator has been designed as an educational simulator for the design and verification of high-performance memristor crossbar architectures. The validation of high-performance computer architectures built using nanoscale memristors requires that their performance be analyzed under different combinations of parameters and under different stochastic noise effects. The *aBCXsim* uses massively parallel GPGPUs, made available to us by the NVIDIA CUDA Teaching Center, to explore the effects of noise and parameter variations on memristive crossbar computing designs.

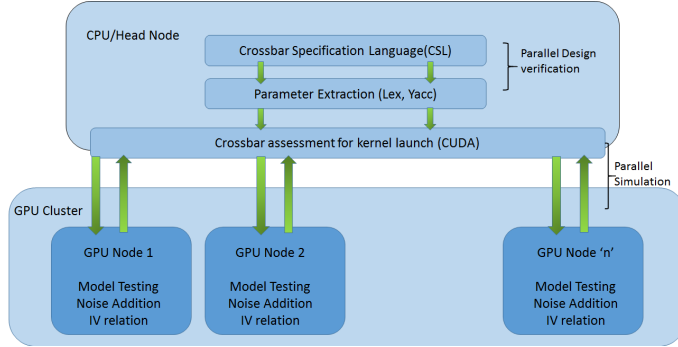


Fig. 1: *aBCXsim*: a Bose-Chua (Cross) X-bar Simulator

The design of the *aBCXsim* simulator consists of four components:

- Crossbar specification language: One of our goals is to permit educational users to express their new memristive crossbar designs in an intuitive formalism. Hence, we have designed a high-level input language capable of expressing networks of nanoscale memristor crossbars.
- Stochastic crossbar simulator: The module parses the description of the computing system written down in the crossbar specification language, and creates multiple copies of the crossbar - taking into account parameter variations and stochastic noise effects. It then applies Kirchoff's laws to solve each such crossbar circuit at a given snapshot of time.
- GPGPU-based memristor simulator: Structurally equivalent memristors in the crossbars are then passed onto the multiple cores of a GPGPU where their device dynamics are evaluated.
- Visualization engine: Using hints provided in the crossbar specification language, statistical summaries of the currents and voltages at the desired locations of the memristor crossbar are plotted.

Figure 1 gives an overview of our *aBCXsim* educational simulator. The simulator produces 3 orders of magnitude improvement in runtime over the sequential HSPICE memristor model [8].

III. PEDAGOGICAL SUMMARY

Our design of the *aBCXsim* simulator targets two distinct groups of users:

- Undergraduate students: The simple and intuitive crossbar specification language enables students with little background in circuit theory to define new parallel memristive computing architectures and simulate them. The simulation kernel and the specification language can be modified by the instructors for their application-specific use. This will enable students from different disciplines, such as biomedical engineering and nanosciences, to use GPGPU parallel computing and bring their domain expertise to the design of next-generation non-von Neumann computing architectures. It will also increase the awareness of parallel programming across disciplines. Undergraduate students familiar with C can also modify the CUDA kernel function performing the memristor dynamics computation. This will help introduce parallel programming and next-generation memristive computing to computer science majors at a very early stage.
- Early-stage graduate students: The *aBCXsim* education tool will enable graduate students in computing, semiconductors, biology, and nanosciences to verify their memristor models and study their process variations under noise by examining the results of the simulation of a network of crossbars of memristors. Thus, our tool provides an opportunity to educate a variety of graduate students about parallel programming using a parallel simulator.

The *aBCXsim* toolset can be used to teach undergraduate students the basics of designing and analyzing parallel memristive crossbar networks using high-performance GPGPUs, while giving others a platform to delve into using parallel programming by modifying the simulator code. The crossbar specification language provides a programming language like environment for educational end-users to design and simulate nanoscale memristive circuits. This enables new users to get hands-on practical experience in designing parallel memristive architectures and provides advanced users a tool to sharpen their analysis and validation skills by taking advantage of the stochastic analysis and visualization capabilities of *aBCXsim*.

REFERENCES

- [1] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, "The missing memristor found," *Nature*, vol. 453, no. 7191, pp. 80–83, 2008.
- [2] L. Chua, "Resistance switching memories are memristors," *Applied Physics A*, vol. 102, no. 4, pp. 765–783, 2011.
- [3] Jason Mick, "How Silicon Valley's Best-Kept Secret, Crossbar, Beat HP to the Market w/RRAM," Online: DailyTech, August 2013, Accessed: September 18, 2014.
- [4] P. Mazumder, S. M. Kang, and R. Waser, "Memristors: devices, models, and applications," *Proceedings of the IEEE*, vol. 100, no. 6, pp. 1911–1919, 2012.
- [5] F. Merrikh-Bayat and S. B. Shouraki, "Memristor-based circuits for performing basic arithmetic operations," *Procedia Computer Science*, vol. 3, pp. 128–132, 2011.
- [6] J. Borghetti, G. S. Snider, P. J. Kuekes, J. J. Yang, D. R. Stewart, and R. S. Williams, "Memristive switches enable stateful logic operations via material implication," *Nature*, vol. 464, no. 7290, pp. 873–876, 2010.
- [7] C. K. K. Lim, A. Gelencser, and T. Prodromakis, "Computing image and motion with 3-d memristive grids," in *Memristor Networks*. Springer, 2014, pp. 553–583.
- [8] N. Shekhawat and S. K. Jha, "Analysis and validation of memristor crossbars using GPGPUs," *NVIDIA GPU Technology Conference (GTC)*, 2015.