"AVEGA"- The Memristor Crossbar Simulator

Nikhil Singh Shekhawat
Nikhil@eecs.ucf.edu

Dr Sumit Kumar Jha
Jha@eecs.ucf.edu

QMAS Lab at University of Central Florida

Abstract

The AVEGA simulator transpires from the Sanskrit word meaning onrush, boost and excitement. As the meaning suggests it is a tool for a large number of computations, and here it is applied for solving memristor crossbar networks. Currently investigations are being performed to model such a memristor that follows a desired hysteresis loop. AVEGA is the first step towards such a validation and verification method, which will capacitate researchers with ways to filter out useful designs, hence AVEGA will be a simulator capable of :

1. Simulating a large number of similar devices in a very short time.
2. Testing against different parameters concurrently
3. Testing models for the common process variations that occur in such devices
4. Adding noise to the parameters to check for a range close to proposed parameters
5. Trying all 2^n combinations and permutations of parameters

In this poster the simulator is directed towards solving a current issue which is of much importance to our lab, and which will bring about significant changes in the near future. The memristor is the fourth passive circuit element postulated by L.Chua and synthesized by HP Labs in 2008. As the Moore's law for transistors is coming to a halt, memristors are a promising option that can be used to extend Moore's law. The element itself possesses many attractive features such as its non-volatility, power efficiency and memory capabilities. Over the last few years there has been an explosion in the number of researchers who are delving into using memristor architecture for arithmetic operations and implementing logical functions. These designs for nanoscale memristor grids suffer from modelling deficiencies and scaling that instill in them convergence errors. It is imperative for these devices, to perform efficiently even in the presence of random noise. These all factors accumulate to a lot many computations that verify the design of a single memristor and to do this for a large set of memristors across a large set of parameters with noise added to them is what we are aiming to achieve. The results and inferences drawn in this poster are with respect to memristive devices following the Biolek model[1], although this can be extended for the computation of any device of similar chattels.

Objectives and Vision

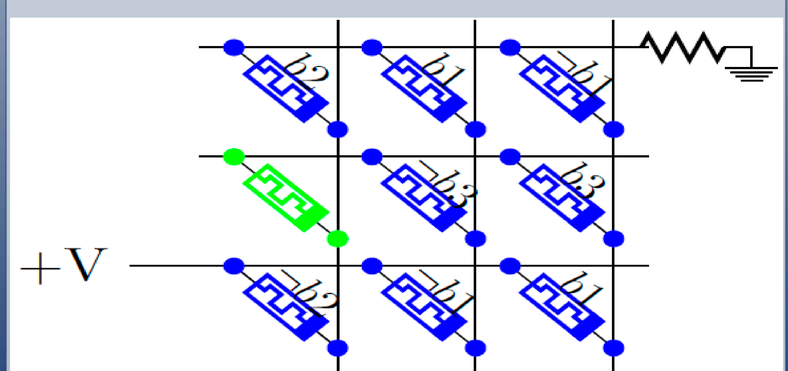
The memristor has demonstrated great potential as a memory and computing platform, and could serve as a next generation computing device. The only thing stopping this, is the synthesizing process, as no tool to validate designs exists. Our objective is to build a high performance CUDA - enabled massively parallel GPGPU validator, to explore the impact of stochastic process variation on the correctness and performance of memristor based designs for neuromorphic computing. This will guide us towards the more promising models and reject unworthy ones.

These details are essential and play a huge role in circuit element development. If they are overlooked, it might end up being catastrophic for a company or even for the world, as:

1. The Y2k bug that cost over $400 billion worldwide

2. The intel bug with the FDIV that resulted in the recall of the P5 processors.

These reasons support the need to administer the imperative process of design verification. To underpin this, we present a simulator to abolish any erroneous designs. The importance and demand for such tools, to simulate and analyze memristor designs are going to increase, as they facilitate design verifications while cutting efforts in terms of cost and time. The traditional HSPICE modeling for memristor doesn't scale well, there is a need to test and analyze these models in real time.



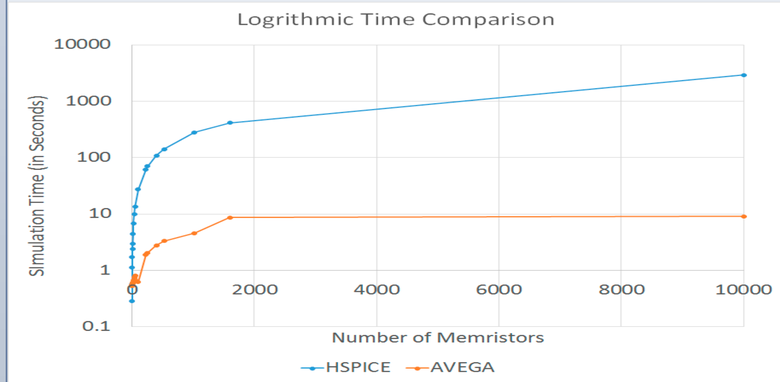
Results

AVEGA— This memristor crossbar simulator using CUDA on GPUs was able to realize significant improvement in the run time for computing values of ten thousand memristors, to the extent that it ran 5 orders of magnitude faster than the HSPICE model.

It was tested against various parameters and many followed the hysteresis curve as predicted by Biolek[1] which asserts the correctness of the simulator.

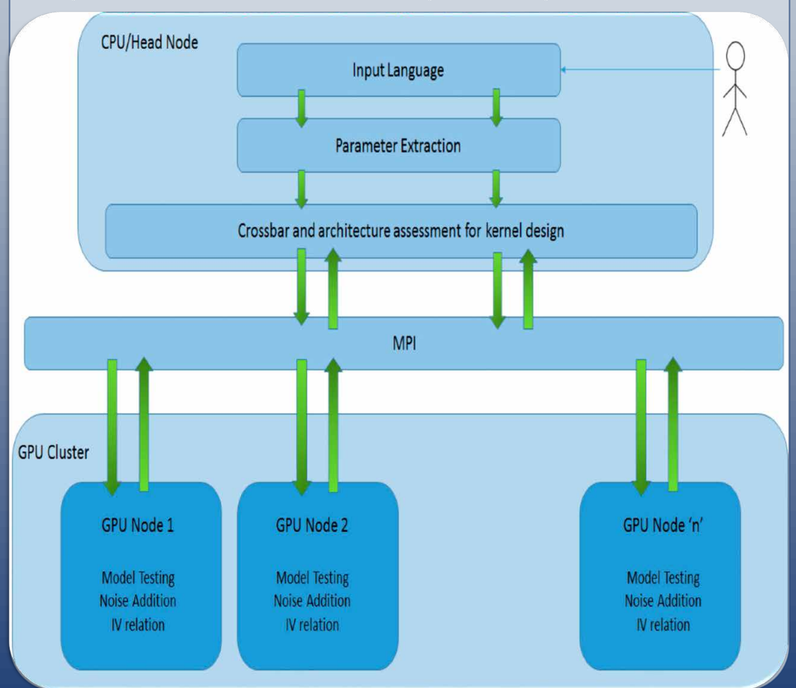
Hence the advantages of using AVEGA are:

1. Faster runtime
2. Accurate simulation, and hysteresis match
3. Simultaneous validation over a larger parameter test space with process variations and noise



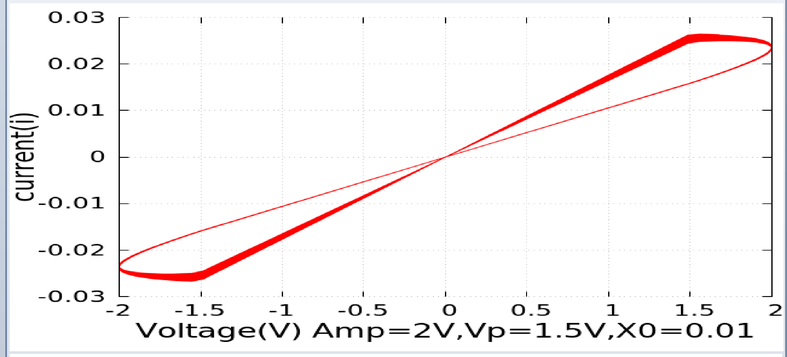
Experiment

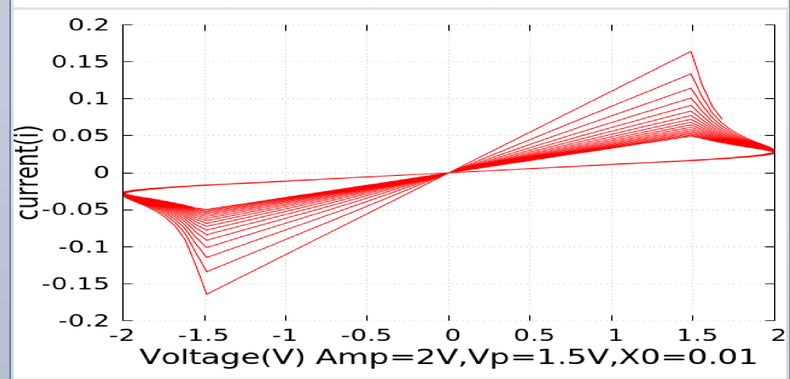
AVEGA tests a memristor model that relies on previous resistance and voltage. Each model is tested against various parameters to check which set of parameters follow the characteristic hysteresis curve. The simulator takes the input design and parameters in a defined language from the user. Based on this design, the simulator creates the memristor grid model which is then rigorously tested over various parameters using Gaussian noise and other functions for stochastic processes. The researcher using this tool will be given a large test result set, and then the best results that match the expected result on the basis of RMS error can also be logged. As the set of parameters is large, this has to be done faster than the traditional HSPICE method. The project can use any HPC of 'n' GPGPU nodes that simultaneously tests and analyzes these models.



Conclusion and Future Work

AVEGA can be used for testing, enumeration, searching, matching and similar operations over a large set of data or parameters. Here it has been used for the testing and matching of memristive devices over a set of parameters with noise added to them. The following images are the hysteresis for the same parameters but different magnitudes of random noise added to them:





Who all would want to use AVEGA?

People who are not that familiar with CUDA programming and the underlying GPU architecture, will be assisted by AVEGA as the simulator through device query and thread assessment will try to optimize the code. Therefore the user might only write the input language and the following benefits will be provided without the user's participation:

- No Performance Tuning Required – The simulator through device query and search space assessment, automatically finds the optimal kernel launch configuration to deliver the best performance.
- High Performance – The kernel launch will be modified according to the GPU's capabilities.
- Performance and Software Portability – The simulator will run on any Nvidia GPU as long as the GPU supports CUDA. The kernel is launched according to the GPU's capacity and it is optimized by device query and thread assessment.
- Error Handling – Every time the simulator is not able to complete the assigned task an error log is made to help the user understand what went wrong.
- Plug and Play Use – The user might only give the input language as the input and watch the magic unfold.

There are a lot of areas where a high amount of computation is required, for example gene matching and protein folding. The enormous compute capability displayed here with respect to the memristor crossbars can be applied/extended for:

1. Verifying memristor crossbar designs for neuromorphic computing
2. SAT solving using heuristics coupled with exhaustive enumeration
3. Memristor circuit flow, simulating the flow of current in a highly dense memristor crossbar circuit

References

[1] Zdenek Biolek, Dalibor Biolek, and Viera Biolkova. Spice model of memristor with nonlinear dopant drift. Radioengineering, 18(2):210–214, 2009.