# Exploiting Heterogeneous Parallel Programming for Developing an Educational Neuromorphic Computing Simulator

QMAS LAB

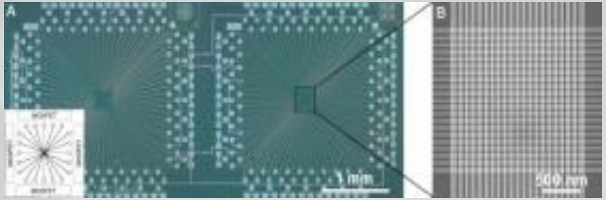
Faraz Hussain, Alvaro Velasquez, Sumit K. Jha, Nikhil Shekhawat

# Memristors

1. Memristors are relatively new circuit elements, that have tremendous potential, and can facilitate improvements in computing.
2. Qualities include nanoscale size, state retention, energy efficiency, non-volatility, high density arrangements

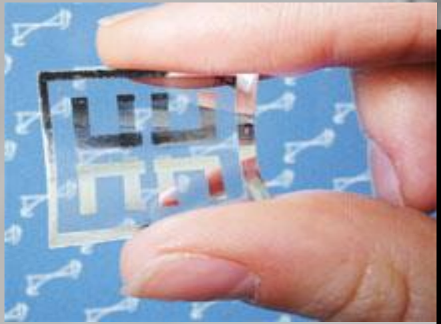
# Memristor crossbar memory

- Crossbar latches allow greater density hard drives with much greater speeds
- GALACA researches have prototyped memory using memristors that fit 100 gigabytes on one square centimeter (not cubic)
- This version of memristive memory is 10 times faster than the fastest memory today (DRAM)



# Applications

1. Instant on computers
2. Brain Neuron synthesis
3. Artificial Synapse
4. Neuromorphic Computing
5. Modeling machines for pattern recognition and A.I.



# Motivation

- Simulating a device with so many process variations, internal flux effects, on the scale of a million memristor crossbar grid is hard.
- There is currently no memristor simulation tool that can accurately simulate millions of memristors at once.

# How is it currently done

- Most of the tools available currently do not focus on memristors as a passive circuit element
- The memristor subcircuit has to be constructed using other circuit elements provided by these tools
- The best simulation currently is done using SPICE

# Problems with Spice simulation

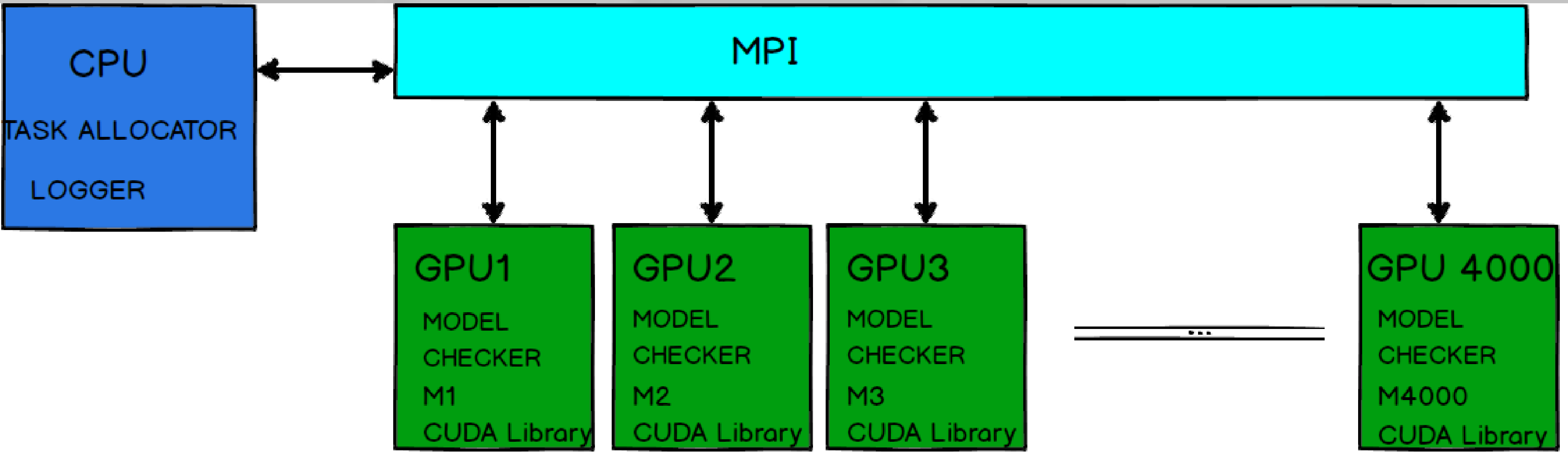
1. Scalability : Convergence errors are introduced into the memristor crossbar as the size of the crossbar increases
2. Dependency on other circuit elements as how they function in the given environment
3. Memristor model is an approximation using capacitors and other elements

# Approach Outline

1. Parallelize each memristor: The value of the memristor depends on the input voltage applied to it. Hence each calculation can be done independently. Efficient heuristics are needed for this.
2. Precise Mapping: Each equation and element in our are directly related to memristor as an independent device.

# Architecture Outline

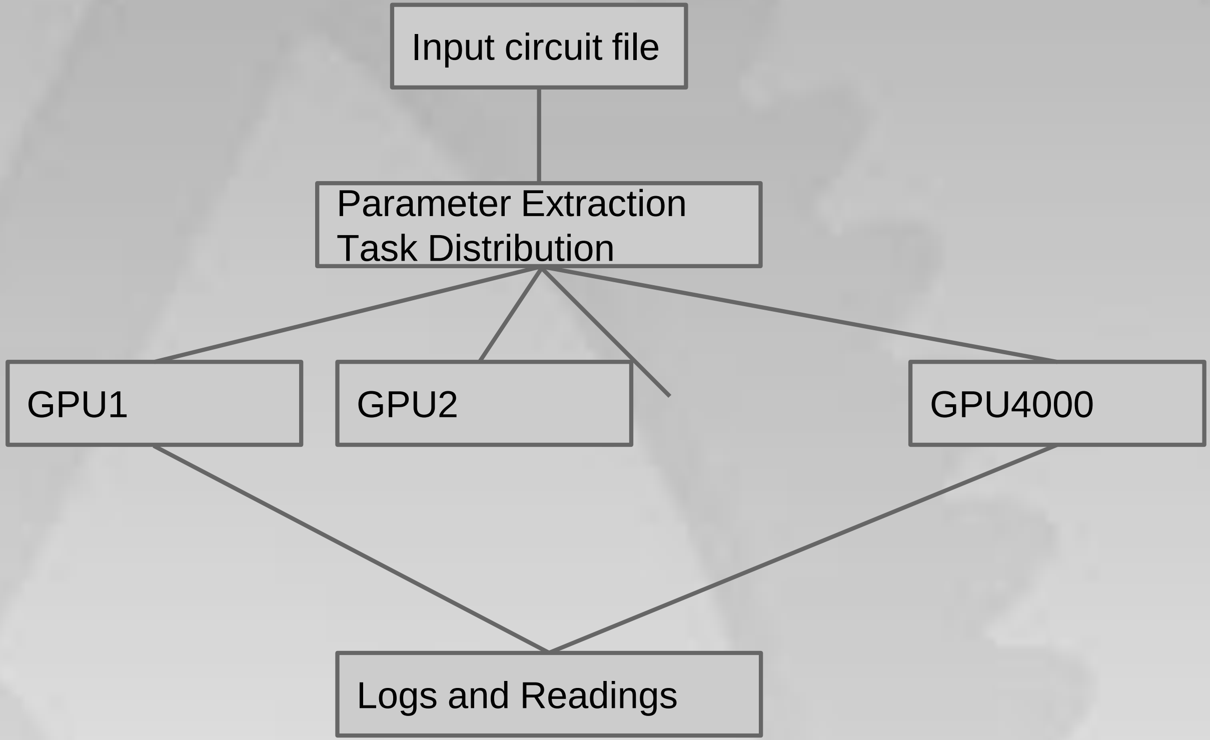
For a crossbar of over a million memristors



# Heuristics

1. The input applied at a wire will push voltage into all the memristors on that wire.
2. These memristors can be calculated in parallel, and the ones that are turned on or remain on will transfer the voltage to the corresponding wire on the other end.
3. This can be parallelized by setting each wire on a GPU and further each memristor on a thread

# Activity Diagram




Recursive calls till the time memristors are switching

# Current Work

1. Parallelization of these calculations for each memristor
2. Using shared memory to optimize each thread, in terms of memory access for parameter access while calculating
3. Validation and verification of the output values. First with HSPICE model(Taha) and then with real memristor models(AFRL).

# SPICE vs Simulator

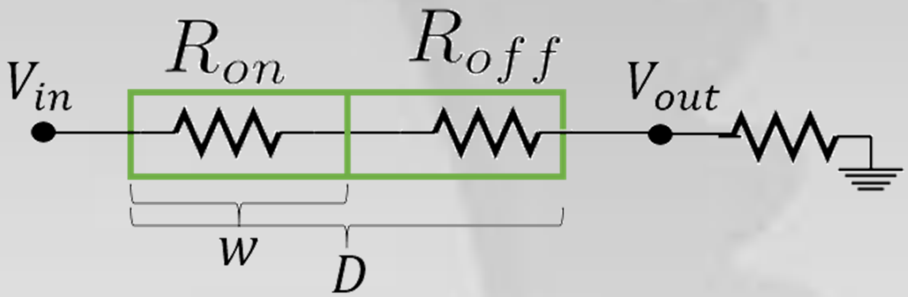
| | |
|---|---|
| Spice has a very broad element set that can be used to build circuits | Will focus only on memristors as the only element in the circuit(at least for the initial version) |
| Not scalable and shows convergence errors | Will be highly scalable |
| Spice supports parameters, functions and circuit modeling and manipulation | Will only support memristor parameter definition, along with grid structure definition |

# Neuromorphic Computing

- Seeks to mimic the brain as a parallel computing platform.
- Neural network algorithms are inherently parallel.
- Traditional von Neumann architectures do not implement these algorithms efficiently at the hardware level.
- Need a low-power architecture that mitigates the transfer and latency issues of traditional architectures.

# Memristors as Synaptic Weights

- The state of a memristor is determined by the amount of charge that has been applied to it.
- A voltage bias causes the state of a memristor to change.
- The memristor's resistance is uniquely determined by its state. Thus, this state can act as a weight for an input voltage so that the output voltage is a function of $V_{in}$ and state.



# Neuromorphic Computing Using Memristors