Reliable Neuromorphic Computation

Background information

As Deep Learning, using artificial neural networks (ANN), is continuing to gain more prominence, it becomes evident that the conventional von Neumann computing systems are not quite the right hardware for implementing them. It is because ANNs are data intensive and in von Neumann architecture memory units and computation unit are placed apart from each other. Therefore, data should move between memory and processing units frequently which not only dissipates a considerable amount of energy, but also elongates the process time significantly. In-memory computing tries to close the gap by performing the computation at the very same site that data resides.

Our research focus is on the platforms that leverage memristors for realization of in-memory computation. Memristors are the fourth fundamental circuit element, alongside resistors, capacitors, and inductors, that can perform as memory devices by storing data as their resistance state rather than electric charge (memristor: memory resistor). Organizing them in a crossbar structure (Fig. 1 a), dense memories can be built out of them. What is more important though is that by exploiting simple circuit laws – Ohm’s law, and Kirchhoff’s Current Summation law (KCL) – they can execute several complex arithmetic and logic operations in a single step, among which vector-matrix-multiplication (VMM) is the most appealing one.

Figure 1. a) memristor crossbar with transistor in series to address sneak-path. b) the developed memristor-based computation in-memory simulator
Since the device is in its infancy, neither is it mass produced nor is a good device model like that of transistors available. There are some research groups, e.g. IBM, HP, UMASS, UMichigan, etc. who make devices though. Considering that, we have built our own functional simulator that mimics the behavior of memristor crossbars and that of their necessary driving circuitry (Fig.1 b). This enabled us to develop our architecture on top of it. However, the simulator only replicates the behavior of a perfect platform that has no non-idealities which is not the case in real world. Your task as explained below is to investigate the non-idealities, integrate them to the simulator, and to find solutions to address those imperfections. There is high potential to publish the achieved results in a peer reviewed paper.

**Student Task Description**

In this project you would be working on our developed simulators (the first two links in the attachments). Your task would be to find the non-idealities associated with memristor based computation in-memory systems, e.g. device-to-device variability, radio telegraph noise, line resistance, etc. (the third link in the attachments), modeling and integrating them to our simulator that does not consider any non-idealities, as well as finding and testing as many as solutions to address the imperfections that you have integrated. Below you can find a proposed break down of tasks.

- Understanding the basics of memristor based in-memory computation
- Understanding the developed architecture and its codes
- Literature review and search for non-idealities in such platforms
- Introducing non-idealities to the developed architecture
- Finding solutions for addressing the errors caused by introduced non-idealities and testing the ideas

**Pre-requisite:**

- Excellent C/C++ skills

**Helpful Skills:**

- Knowledge about VLSI
- Work independently
- Critical approach to the problem

You will be helped to get to know the topic, by receiving the necessary documents and regular meetings.

**Attachments with some important background information:**
