Memcomputing: storing and processing at the same time

Jhonatan S. Oliveira oliveira@cs.uregina.ca
Department of Computer Science
University of Regina
Regina, Canada

Editor: CS807 - Advanced Architecture - Dr. Gerhard and Trevor Tomesh

Abstract

Memcomputing was recently proposed as an alternative to turing paradigma of computation. The difference here is that the computation and the storage is done within the same architecture. In order to achieve that, memcopmuters use memelements, which are two-terminal electronics components with resistive, capacitive, and inductive characteristics. Memcomputers are mathematically proved able to solve NP-hard problems from turing computers.

Keywords: Memcomputing, post-silicon copmuting

1. Introduction

Commercial computer architectures are based on a turing machine (Turing, 1948). These computers can generally be split into two main computing tasks: storing and processing. In order to handle these tasks, different implementation patterns were proposed, for instance, the von-Neumann architecture. As technology evolved, storage became cheap and processing demand rose to exponential levels. That is, memories and processors might have different working speed. Thus, the mode of operation of these architectures imposes a limitations, which are collectively known as the *von-Neumann bottleneck* (Backus, 1978).

There are already some suggested alternatives to these limitations, including parallel computing and quantum computing. Parallel computing are implemented by using multiple core processors. While parallel computing can minimize the von-Neumann bottleneck issue in one computing unit, when considering scaling the whole workstation it might need specialized units such as *graphic processing units* (GPUs). These means more complexity on the infra-structure besides considering limitations of the specialized units themselves. On the other hand, quantum computers are promising in the sense that they are an unique computing platform with intrinsic massively parallel computing scheme. However, even considering the most recent improvements in the field, a practical quantum computing can not outperform a tradicional one yet.

An alternative computer that might outperform the current ones should be able to has at least two properties: first, an intrinsically massively-parallel architecture, and second, storage and computation are performed by the same basic units. Transistors can not be used for achieving these goals since they are active elements, that is they require power to perform tasks. Thus, the massive parallelization comes with the expense of considerable power consumption. Besides that, normal transistor can not store informations. Lastly, transistors have low density.

The best electronic element to achieve this goal are memelements. These are two terminal passive devices, that is do not require energy to work. Memelements have three different characteristics: resistive, capacitive, and inductive. These characteristics change accordingly to the current passing through them, as if the characteristic is remembering the current. Thus the name memelements. *Memcomputing* (Di Ventra and Pershin, 2012) is a non-turing paradigma for processing and storing information by using memelements.

The reminder of the paper follows. Section 2 presents the basic background on the field and relates memcomputing with other alternative to the conventional computer. The criteria to implement a memcomputing system are given in Section 3. Section 4 shows applications already using memcomputing. Conclusions are drawn in Section 5.

2. Background and Related Works

When defining a new computing paradigma it is necessary to establish crucial fatures, requirements, and implementation comments. Here we call them criteria of the computing system. Memcopmuting has 6 criteria as defined in (Di Ventra and Pershin, 2012). Also, (Di Ventra and Pershin, 2012) says that these criteria are close related to the quantum computing. Both quantum computing and memcomputing relies in massively parallel computation, but the mechanics behind these requirement are quite different. Quantum computing relies on a phenomenon called *superposition of states*, while memcomputing relies on the collective dynamic of simple, smaller and classical computers.

3. Memcomputing Criteria

This section is drawn from (Di Ventra and Pershin, 2012) where the criteria were first presented.

The first criteria is a scalable massively parallel architecture with combined information processing and storage. Here, the memelements are units of processing and storing information, while still working together as a whole system. A memelement can store information in two forms: its characteristic response, namely the voltage over the element when the input is applied, and in its intrinsic characteristic. For instance, a capacitor and inductor can store information on its electric and magnetic field, respectively.

Second criteria is the sufficient long information storage times. The memelements should hold their state, that is the stored information, at least longer than the required time for processing them. In order to approach that, memelements can make use of non-volatile memory cells. For instance, the ones using CMOS technology.

The third one is the ability of initialize memory state. This has to be with the programming desire. Relevant memelements can be be initialized using a provided mechanism for initialization. This can be done by applying an input that makes the memelements change their states to extreme points.

The fourth criteria is a mechanism of collective dynamics. The basic idea here is to promote the collaborative characteristic of the architecture. That is, the current state of an element depends on the current state of some other elements. For instance, in a memelement circuit with resistive characteristics changing one element voltage affect other ones, since the resistance changes affecting the voltage in subsequent memelements.

The fifth criteria is the ability to read the final result. After a processing is done in a memelement architecture, it is desired the possibility of reading the result of it. But the reading process can not modify the result state itself. This can be done by choosing an input voltage that the memelements does not change at that level or change little enough to do not be considered.

The sixth and last criteria is the robustness against small imperfections and noise

4. Applications

In order to test the abilities proposed for memcomputers, (Di Ventra and Pershin, 2012) did an experiment with the problem of finding the shortest path. The problem of the shortest path is the procedure of finding a way to connect two points in a matrix via the shortest path of points between them. It is known that this problem has complexity of logarithmic time Ahuja et al. (1990).

The proposed memcomputer architecture is a square matrix made of memelements with resistive characteristic, as shown in Figure 1. Here, the architecture is a non-programable computer with memelements connected in a square format to each other. It is worth mentioning that this memcomputer is specifically projected to solve the shortest path, that is it is not programmable for other functionalities other than that. Also, notice that each memelement has a switch attached to it, which provides the possibility of writing and reading individually. This is requires for initialization purposes as well as reading the result after the computation.

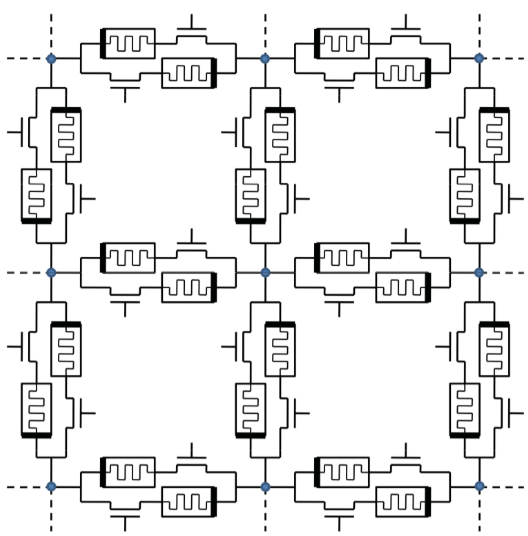

Figure 1: A memcomputer proposed in (Di Ventra and Pershin, 2012) for solving the shortest path problem.

In this computer, we give the inputs by activating a subset of points in this grid, that is by applying a voltage in a subset of memelements. The processing part is the collective evolution of the states of the memelements after the input is applied. Lastly, the result of the processing step is the final state of all memelements.

The shortest path problem can be be computed using the memcomputer in Figure 1 by applying a constant voltage to two points the square. These are the required input. The memelements closer to the application of the voltage is affected by the electron current around. Therefore, the memelement changes its resistance by lowering it down, that is the memelement become active. The closest memelement activates itself similarly, forming a reinforcement path. In this way, a path from the first application point to the second one is formed. The final state that can be read from the memcomputer is a straight line of active memelements from the first point to the second one, which is the shortest path in this square.

5. Conclusions

Memcomputers are non-Turing paradigma of computers. They use memelements as unit of processing and storage. One application of memcomputers is shown by solving the shortest path problem. Memcomputers are also known for solving NP-hard problems in polynomial time (Traversa et al., 2014).

References

Ravindra K.; Ahuja, Kurt Mehlhorn, James Orlin, and Robert E Tarjan. Faster algorithms for the shortest path problem. *Journal of the ACM*, 37(2):213–22, 1990.

John Backus. Can programming be liberated from the von neumann style? - A functional style and its algebra of programs. *Commun. ACM*, 21(8):613–641, August 1978. ISSN 0001-0782. doi: 10.1145/359576.359579. URL `http://doi.acm.org/10.1145/359576.359579`.

M Di Ventra and Y V Pershin. Memcomputing: a computing paradigm to store and process information on the same physical platform. *arXiv.org*, (4):200–202, November 2012.

Fabio L Traversa, Chiara Ramella, Fabrizio Bonani, and Massimiliano Di Ventra. Memcomputing NP-complete problems in polynomial time using polynomial resources and collective states. November 2014.

Alan M. Turing. Intelligent machinery. *Cybernetics: Key Papers*, 4, 1948.