Memcomputing: computing *with* and *in* memory

Massimiliano Di Ventra[1]*, Fabio Traversa[1,2], Yuriy V. Pershin[3]

[1] Department of Physics, University of California, San Diego,

9500 Gilman Drive, La Jolla, California 92093-0407, USA

[2] Dipartimento di Elettronica e Telecomunicazioni, Politecnico di Torino, I-10129 Torino, Italy

[3] Department of Physics and Astronomy, University of South Carolina, Columbia, South Carolina 29208, USA

* Corresponding author. E-mail: diventra@physics.ucsd.edu

The need for a new computing paradigm

Present electronic systems used for computing (PCs, workstations and clusters) are based on the von Neumann architecture (see Figure 1, left panel) [1]. This computing paradigm employs the Turing machine concept [2], [3], and involves a significant amount of information transfer between a central processing unit (CPU) and memory, with concomitant limitations in the actual execution speed and large amounts of energy used to move data. Therefore, there is currently a surge of interest in unconventional computing approaches [4]-[11] that can outperform the present von Neumann one [4], [5]. It is clear that such alternatives have to fundamentally depart from the existing one in both their computational complexity as well as in the way they handle information. For at least a couple of decades, quantum computing [12] has been considered a promising such alternative, in view of its intrinsic massive parallelism afforded by the superposition principle of quantum mechanics. However, the practical realization of quantum computers seems still too far away from the present, and even near-future, technologies.

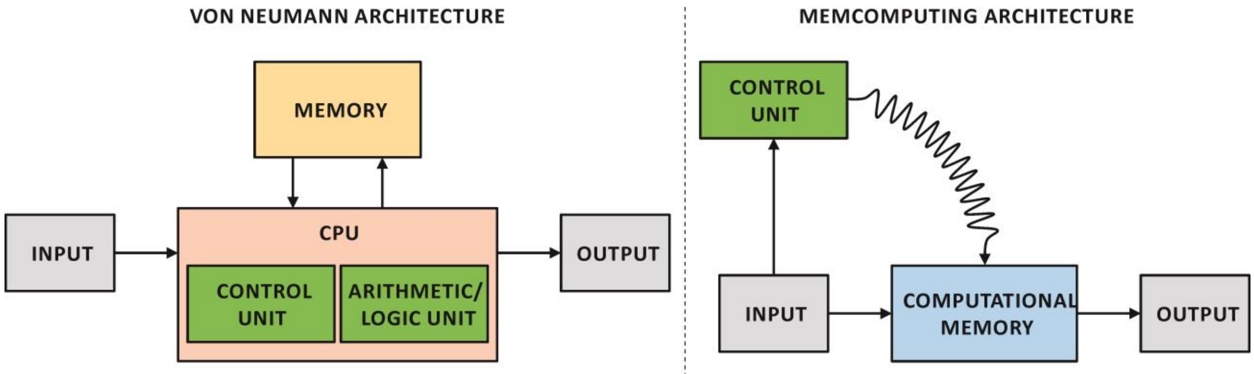

Figure 1. Von Neumann architecture versus the Memcomputing architecture.

In order to overcome the above mentioned limitations we then need to look for another paradigm, and the solid-state emulation of our own brain may provide the solution.

It is estimated that our brain uses only 10 to 25 Watts per day to perform about 10^{16} operations per second [13]. A supercomputer would require more than 10^{7} times that power to do the same amount of operations. And a computer does not even come close to performing such complicated tasks as pattern recognition, optimization problems, decision making, *etc*. we do in the noisy and unpredictable environment we live in, and in a massively-parallel way

How is it then possible that our brain is such a powerful computing machine and yet uses so little energy to operate? The answer definitely cannot come only from the number of computing elements (about 10^{11} neurons). Rather, it has to ultimately boil down to the fundamentally different way in which computation and information storage are accomplished in our nervous system. In fact, unlike our present (super-)computers, calculations in the brain are not performed in a CPU that is physically separated from the memory: our brain computes and stores information on the *same physical location*. This way of computing avoids the large amount of information transfer to/from the CPU and the memory, saving *both* energy *and* time.

Memcomputing

Can we realize this paradigm in the solid state? The answer is yes, with the available CMOS technology as well as materials or two-terminal systems that can hold information even in the absence of an external power source. These systems are resistors, capacitors and inductors with memory (memristors, memcapacitors, and meminductors, respectively) [14]. They can also be made using CMOS-compatible structures and devices thus offering unprecedented opportunities in electronics. In particular, all these standard and non-standard systems and devices allow precisely the paradigm we are looking for. We named this paradigm *memcomputing i.e., computing within memory* [4], [5], namely the ability to process information directly in/by the memory (see Figure 1, right panel for a schematic of a memcomputing architecture). This computing paradigm rests on the solid mathematical foundation of *universal memcomputing machines* [5]. We have indeed recently shown that these machines have the same computational power of non-deterministic Turing machines, thus allowing the solution of complex problems in polynomial time with polynomial resources.

Practical realizations

We have recently proposed a simple and practical realization of memcomputing [6] that utilizes easy to build memcapacitive systems [15]. We have named this architecture dynamic computing random access memory (DCRAM) (see Fig. 2). We have shown that DCRAM provides *massively-parallel* and *polymorphic* digital logic, namely it allows for different logic operations with the same architecture, by varying only the control signals. In addition, by taking into account realistic parameters, its energy expenditures can be as low as a few fJ per operation. DCRAM is also fully compatible with CMOS technology, can be realized with current fabrication facilities, and therefore can really serve as an alternative to the present computing technology.

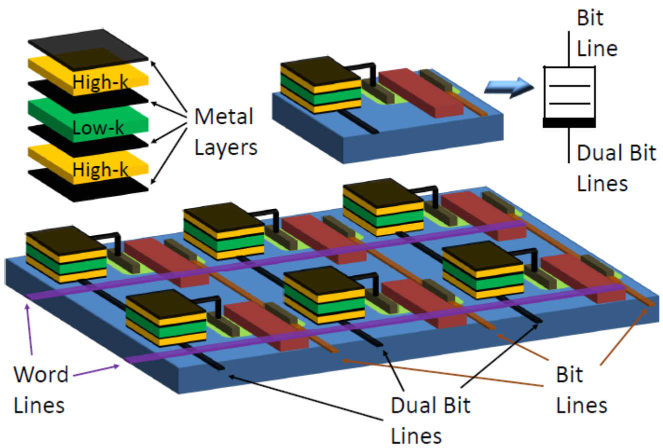

Figure 2. Scheme of a DCRAM architecture.

Conclusions

In conclusion, we have introduced the concept of computing *with* and *in* memory: *memcomputing*. This new computing paradigm can be realized in the solid state with available systems and materials and it is compatible with CMOS technology. It provides a solution to the time and energy constraints of traditional von Neumann architectures, while offering a powerful new computational tool for solving complex problems that currently require an exponentially large number of resources and time.

This work is a collaborative effort between CMRR and the Physics Departments at UCSD and the University of South Carolina. It is partially supported by NSF grant ECCS-1202383.

References

[1] J. von Neumann, *First draft of a report on the edvac*, Annals of the History of Computing, IEEE, **15**, 4, 27–75, 1993.
[2] A. M. Turing, On *computational numbers, with an application to the entscheidungsproblem*, Proc. of the London Math. Soc., **42**, 230–265, 1936.
[3] A. M. Turing, *The Essential Turing: Seminal Writings in Computing, Logic, Philosophy, Artificial Intelligence, and Artificial Life, Plus The Secrets of Enigma*. Oxford University Press, 2004.
[4] Di Ventra, M., Pershin, Y. V., *The parallel approach*, Nature Physics, **9**, 200, 2013.
[5] Traversa, F., Di Ventra, M. *Universal Mamcomputing Machine*, arXiv:1405.0931, 2014.
[6] Traversa, F. L. Bonani, F. Pershin, Y. V. Di Ventra, M., *Dynamic Computing Random Access Memory*, Nanotechnology, **25**, 285201, 2014.
[7] Pershin, Y. V. Di Ventra, M. *Experimental demonstration of associative memory with memristive neural networks*, Neural Networks, **23**, 881, 2010.
[8] Borghetti, J., Snider, G. S., Kuekes, P. J., Yang, J. J., Stewart, D. R., Williams, R. S., *Memristive switches enable `stateful' logic operations via material implication Nature*, Nature, **464**, 873–876, 2010.
[9] Pershin, Y. V., Di Ventra, M., *Solving mazes with memristors: a massively-parallel approach*, Phys. Rev. E, **84**, 046703, 2011.
[10] W. Maass, T. Natschlager, and H. Markram, "*Real-time computing without stable states: A new framework for neural computation based on perturbations*," Neural Comput., **14**, 2531–2560, 2002.
[11] Guest editor A. Adamatzky, *Special Issue: Unconventional Computing*, International Journal of General Systems, **43**,7, 2014
[12] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information.* Cambridge Series on information and the Natural Sciences, Cambridge University Press, 10th Aniversary ed., 2010.
[13] E. R. Kandel, J. H. Schwartz, T. M. Jessell, Principles of neural science, McGraw-Hill, 2000.
[14] M. Di Ventra, Y. Pershin, and L. Chua, *Circuit elements with memory: Memristors, memcapacitors, and meminductors*, Proceedings of the IEEE, **97**, 1717–1724, 2009.
[15] J. Martinez-Rincon, M. Di Ventra, and Yu. V. Pershin, *Solid-state memcapacitive system with negative and diverging capacitance*, Phys. Rev. B, **81**, 195430, 2010.