Memcomputing: the Cape of Good Hope

[extended special session description]

Yiyu Shi
ECE Dept., Missouri S&T
Rolla, MO, 65409, U.S.A.
yshi@mst.edu

Hung-Ming Chen
EE Dept., National Chiao Tung University
Hsinchu, Taiwan, 30010, R.O.C.
hmchen@mail.nctu.edu.tw

Energy efficiency has emerged as a major barrier to performance scalability for modern processors. On the other hand, significant breakthroughs have been achieved in memory technologies recently [1-4, 6]. As such, the fascinating idea of memcomputing (i.e., use memory for computation purposes) has drawn wide attention from both academia and industry as an effective remedy. Compared with conventional logic computing, memory array provides large set of parallel resources with high bandwidth, which can be configured to perform in-situ computing and information processing, leading to drastic reduction in processor-memory traffic. It will not only make computations more power- and speed-efficient, but also smarter. In addition, it exploits the advances in memory technologies (e.g., [8, 9]) and integration approaches (e.g. 3D integration [11-17]) to achieve better technology scalability. This special session includes three presentations that offer a broad-spectrum retreat on this hot topic.

The first presentation of this special session, titled *Memcomputing: a brain-inspired computing paradigm* and given by Prof. Pershin from University of South Carolina, focuses on a memcomputing paradigm that employs two-terminal electronic devices with memory (memelements), namely, memristive, memcapacitive or meminductive systems [1], to store and process information at the same physical location. Complex networks of such devices can be considered as massively-parallel processors (Figure 1) performing computation in an unconventional way [2]. In order to fabricate a viable memcomputing device, several criteria must be met. Specifically, the scheme requires: 1) Scalable massively-parallel architecture with combined information processing and storage. 2) Sufficiently long information storage times. 3) The ability to initialize memory states. 4) Mechanisms of collective dynamics, strong 'memory content'. 5) Ability to read the final result from the relevant memelements. 6) Robustness against small variations and noise.

In particular, the authors have investigated a representative memcomputing architecture based on two-dimensional networks of memristive devices [3,4]. The main advantage of this architecture is based on the analog parallel dynamics of many memristive elements. The authors show that such networks can efficiently solve various shortest-path optimization problems [4]. The presence of memory promotes self-organization of the network into the shortest possible path(s). One can introduce a network entropy function to characterize the self-organized evolution and show that the entropy decreases as the shortest-path solution emerges in an initially homogeneous network.

978-3-9815370-2-4/DATE14/©2014 EDAA

Additionally, the memristive networks have a remarkable ability to repair damaged solutions. This property is very similar to the self-healing ability of human brains.

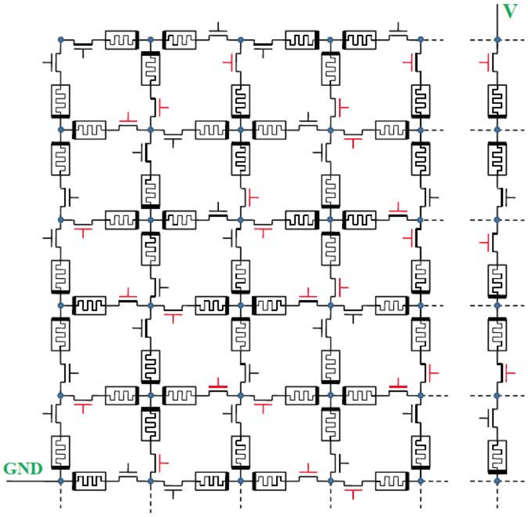


Figure 1. Illustration of a processor formed by massively-parallel memelements.

It is worth mentioning that similar considerations apply to networks of memcapacitors and meminductors, and networks with memory in various dimensions. Some work has already been done along these lines. For example, the recently developed concept of Dynamic Computing Random Access Memory [5] utilizes memcapacitors to store and process information *directly* in memory at low energy cost.

The second presentation of this special session [6] proposes a novel hardware accelerator framework that transforms high-density memory array into a configurable computing resource to accelerate variety of tasks – both compute- and data-intensive. Since energy-efficiency has emerged as a major barrier to performance scalability for modern processors, a computing paradigm that transfers data-intensive application kernels into the last level of memory (LLM) and performs computation within the LLM is imperative to overcome the bottleneck. Such an in-memory computing paradigm can be realized by implementing a reconfigurable computing fabric in the LLM to map data-intensive applications (Figure 2). It transforms the regular block-based organization of memory arrays into a fabric of reconfigurable computing resources connected through programmable interconnects. To enable the computing paradigm, however, the LLM technology should be compatible for logic integration. Fortunately, existing NAND/NOR flash memory as well as most emerging non-volatile memory technologies (e.g. resistive and spin-based) are logic-compatible, allowing the required transformation.

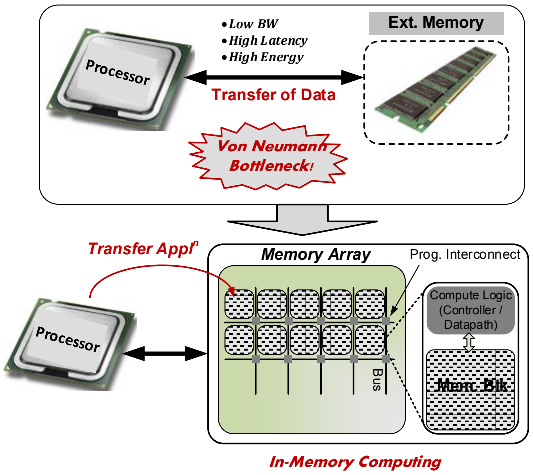


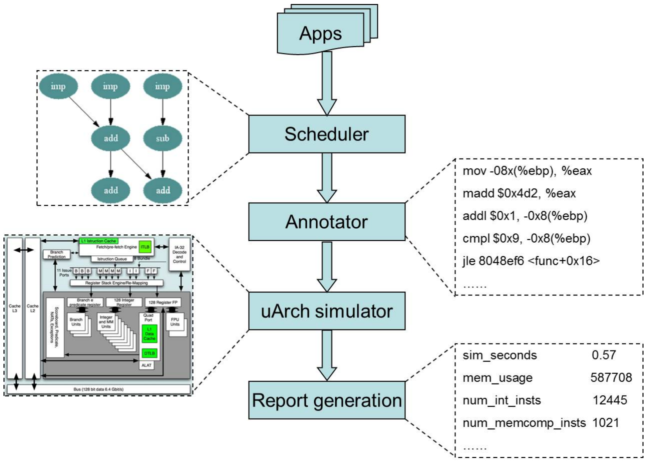
Figure 2: In traditional computing systems, Von Neumann bottleneck is a barrier to improving energy-efficiency for data-intensive applications. An in-memory computing framework can serve dual purpose of storage and computing on demand to mitigate this bottleneck.

As such, the authors exploit the block-based architecture of nanoscale memory to create a spatially connected array of lightweight processors, each of which uses a memory block as its local memory. The proposed framework provides some unique advantages for hardware acceleration compared to conventional accelerators: 1) memory array provides large set of parallel resources with high bandwidth, which can be configured to perform computing in spatio/temporal manner leading to dramatic reduction in processor-memory traffic; 2) many complex functions in the domains of security, signal processing, communication, and informatics are suitable for mapping to memory as large multi input/output lookup tables (LUTs); 3) it brings the computing engine close to the data, thus drastically minimizing the von Neumann bottleneck; 4) finally, it exploits the advances in memory technologies and integration approaches e.g. 3D integration to achieve better technology scalability compared to alternative reconfigurable accelerator platforms.

To enable hardware acceleration through memcomputing, a software framework is also needed for application mapping as well as automatic extraction of application kernels that are amenable to mapping into the proposed framework. Towards this, the authors also propose an efficient application mapping process that can efficiently map complex application kernels from their control and data flow graph (CDFG) into the proposed framework. Simulation results for several common applications show that the proposed computing approach provides over 91X improvement in energy efficiency compared to software execution which is 5-10X better compared state-of-the-art reconfigurable accelerator platforms while achieving significantly lower hardware overhead.

The third presentation of this special session [10] introduces MSim, a general yet open-to-public cycle accurate simulation platform that aims at motivating further studies in memcomputing. Although there have already been a few studies in advanced memcomputing technologies, the validation of their results are still performed by various in-house tools. The lack of open simulation platform has created an invisible barrier for those who newly enter this research field, and also make it difficult memcomputing technologies to be compared in a fair manner.

To address this problem, the simulation platform presented by the authors consists of the following tool chains: 1) A scheduler that statically schedules the operations to be computed in-memory. 2) An annotation engine that extends the existing instruction set architectures (ISAs) to support new memcomputing operations. 3) A cycle-accurate microarchitecture level simulator engine based on Gem5 [7] that accepts the annotated application for detailed simulation. 4) A report engine that provides user friendly simulation results. An overview of the platform structure is shown in Figure 3.



```
mov -08x(%ebp), %eax
madd $0x4d2, %eax
addl $0x1, -0x8(%ebp)
cmpl $0x9, -0x8(%ebp)
jle 8048ef6 <func+0x16>
......
```

```
sim_seconds            0.57
mem_usage              587708
num_int_insts          12445
num_memcomp_insts 1021
......
```

Figure 3. The structure of the cycle accurate simulation platform for memcomputing introduced in [3].

The overall goal of MSim is to provide an open yet flexible infrastructure for various memcomputing studies. It can either be directly used to evaluate the performance of memcomputing technologies in conventional computer systems, or as the starting point to explore other architecture level or computing paradigm innovations with minimal development efforts.

In summary, we hope that the three presentations in this special session can inspire more follow-up works in the field of memcomputing, an emerging technology that can potentially change our computing paradigm in the near future.

REFERENCES

[1] M. Di Ventra, Y. V. Pershin and L. Chua, “Putting Memory into Circuit Elements: Memristors, Memcapacitors and Meminductors,” Proceedings of the IEEE 97, pp. 1371, 2009

[2] M. Di Ventra and Y. V. Pershin, “The Parallel Approach,” Nature Physics 9, pp. 200-202 2013

[3] Y. V. Pershin and M. Di Ventra, “Solving Mazes with Memristors: a Massively-Parallel Approach,” Phys. Rev. E 84, 046703, 2011

[4] Y. V. Pershin and M. Di Ventra, “Self-organization and Solution of Shortest-path Optimization Problems with Memristive Networks,” Phys. Rev. E 88, 013305, 2013

[5] F. L. Traversa, F. Bonani, Y. V. Pershin, M. Di Ventra, “Dynamic Computing Random Access Memory,” arXiv:1306.6133

[6] S. Bhunia, R. Puri and S. Paul, Energy-Efficient Hardware Acceleration through Computing in the Memory, in *Proc. of Design, Automation & Test in Europe*, 2014.

[7] N. Binkert et al. "The gem5 simulator." ACM SIGARCH Computer Architecture News 39.2 (2011): 1-7.

[8] Y. Wang, et al. "Design of low power 3D hybrid memory by non-volatile CBRAM-crossbar with block-level data-retention." Proceedings of the 2012 ACM/IEEE international symposium on Low power electronics and design. ACM, 2012.

[9] L. Wang, et al. "A novel memristor-based rSRAM structure for multiple-bit upsets immunity." IEICE Electronics Express 9.9 (2012): 861-867.

[10] C. Zhang, P. Deng, H. Geng, J. Liu, Q. Zhu and Y. Shi, "MSim: A General Cycle Accurate Simulation Platform for Memcomputing Studies," in Proc. of Design, Automation & Test in Europe, Germany, 2014.

[11] G. Luo, Y. Shi, and J. Cong, "An Analytical Placement Framework for 3D ICs and Its Extension on Thermal-Awareness," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 32(4), pp. 510-523, 2013.

[12] C.-L. Lung, Y.-S. Su, H.-H. Huang, Y. Shi, and S.-C. Chang, "Through-Silicon Via Fault-Tolerant Clock Networks for 3D ICs," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 32(7), pp. 1100-1109, 2013.

[13] C. Zhang, D. Ma, C. Li and Y. Shi, "Runtime Self-Calibrated Temperature-Stress Co-Sensor for 3D Integrated Circuits," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2014.

[14] P.-W. Luo, C. Zhang, Y.-T. Chang, L.-C. Cheng, H.-H. Lee, B.-L. Sheu, Y.-S. Su, D.-M. Kwai, and Y. Shi, "Benchmarking for Research in Power Delivery Networks of Three-Dimensional Integrated Circuits," in Proc. of International Symposium on Physical Design, pp. 17-24, Lake Tahoe, 2013

[15] T. Wang, C. Zhang, J. Xiong and Y. Shi, "Eagle-Eye: A Near-Optimal Statistical Framework for Noise Sensor Placement," in Proc. of International Conference on Computer-Aided Design, pp. San Jose, 2013

[16] C. Zhang, M. Jung, S.-K. Lim and Y. Shi, "Novel Crack Sensor for TSV-based 3D Integrated Circuits: Design and Deployment Perspectives," in Proc. of International Conference on Computer-Aided Design, San Jose, pp. 371-378, 2013

[17] U. Tida, C. Zhuo and Y. Shi "Through-Silicon-Via Inductor: Is it Real or Just a Fantasy?" in Proc. of Asia and South Pacfic Design Automation Conference, pp. 837-842, Singapore, 2014