Memristive Multistate Pipeline Register

Shahar Kvatinsky, Yuval H. Nacson, Yoav Etsion,
Avinoam Kolodny, and Uri C. Weiser
Department of Electrical Engineering
Technion – Israel Institute of Technology
Haifa 3200000, ISRAEL
skva@tx.technion.ac.il

Ravi Patel and Eby G. Friedman

Department of Electrical and Computer Engineering
University of Rochester
Rochester, New York 14627, USA

Abstract— **Over the past years, new memory technologies such as RRAM, STT-MRAM, and PCM have emerged. These technologies employ devices located within the metal layers of the chip, which are relatively fast, dense, and power efficient and can be considered as 'memristors'. In this paper, we present these emerging memory technologies as enablers to the era of memory-intensive computing, which brings interesting opportunities for novel architectural applications. As an example, we present the multistate pipeline register (MPR), a structure that stores the microarchitectural state of multiple threads. We show that MPR can eliminate the need to flush the pipeline upon a thread switch in Switch-on-Event (SoE) multi-threading machines. We call the new microarchitectural scheme, Continuous Flow Multi-Threading (CFMT), and compare the performance and power consumption against traditional SoE machines. Memristor-based CFMT exhibits an average performance improvement of 32%, while reducing power consumption by 8.5%, thereby significantly increasing the performance to energy ratio.**

Index Terms—**Memristor, Memristive device, STT-MRAM, PCM, RRAM, multithreading, CFMT, multistate register.**

I. INTRODUCTION

In recent years, new memory technologies (*e.g.*, RRAM, STT-MRAM, and PCM) have emerged. Although these memories have primarily been investigated as potential replacements for flash memory, these emerging memories have also exhibited the potential to replace other memory technologies, offering new characteristics for the memory hierarchy [1]. These emerging memory technologies are nonvolatile, relatively fast, low power, and dense. While the physical mechanisms are different, all of these technologies rely on resistance to represent the stored logical state and can therefore be referred to as *memristive devices* [2] (or for simplicity, as *memristors* [3])[1].

One interesting shared characteristic of these emerging memory devices is that these devices are fabricated between two layers of metal. Memristors can therefore be fabricated above the CMOS layers with good integration and extremely high density (the memristors can be stacked on several layers to further increase density). With many memory elements located above the CMOS logic, a significant change in computer architecture is possible – opening the era of *Memory Intensive Computing*. Memory intensive computing does not only mean larger and faster conventional memories (*e.g.*, a cache or register file). Memory intensive computing also includes new microarchitectures and novel memory elements, which use the "sea of memory" located above the standard CMOS circuits to enhance the performance of processors while reducing power dissipation.

In this paper, a new memory structure – the Multistate Pipeline Register (MPR) is presented to demonstrate a memory intensive architecture. An MPR stores the data of different instructions in the pipeline during execution, similar to regular pipeline registers. Unlike regular pipeline registers, an MPR can store the state of numerous different instructions from different threads. The high number of threads stored in a single MPR is achieved due to the high density of memristors and the short distance from the pipeline stages to the MPR. In this paper, MPRs are used to reduce the penalty of a thread switch on a Switch-on-Event multithreading (SoE) processor, enabling a new microarchitecture – Continuous Flow Multithreading (CFMT) [4]. CFMT achieves significantly higher performance than SoE, and comparable performance to Simultaneous multithreading (SMT), while keeping the complexity and power significantly lower than SMT and similar to SoE. CFMT shows on average 32% performance improvement as compared to Switch-on-Event multithreading and up to a 75% improvement for floating point benchmarks, while also reducing the energy by 8.5% [5].

II. RRAM-BASED MULTISTATE REGISTER

The introduction of massive amount of memory elements above the CMOS logic creates an opportunity to store data created at the CMOS level. The multistate register is a novel memory structure, used to store multiple bits within a single element. One set of bits is an active set, while the other sets are idle and stored for future usage. The multistate register is a synchronous storage element, when the procedure to change the active set is synchronized by a clock. The basic logic structure of a multistate register is shown in Figure 1.

Although it is possible to design a CMOS SRAM-based multistate register (or any other conventional memory technology), emerging memory technologies enable high capacity multistate registers due to the high density and low leakage. SRAM-based multistate register has high area, while the equivalent RRAM-based multistate register requires a relatively small area overhead. For example, a 64 state multistate register of SRAM based on a 22 nm CMOS process is 83 times larger than RRAM-based multistate register. An RRAM-based multistate register is shown in Figure 2 [6]. In this circuit, the inactive bits are stored within the RRAM layer, where compatibility with existing digital circuits is used to store the active bits within standard CMOS D flip flops. The switching procedure between the active set and idle sets is achieved in two phases – the previously active set is initially written into the memristor layer following by reading of the data from the desired idle set into the CMOS layer.

A multistate register can be used for different purposes. In this paper, the application of multistate pipeline registers (MPR) is described and demonstrated. In pipeline registers, the state of the instruction from the preceding pipeline stage is stored and transferred to the next pipeline stage. In an MPR, additional instructions are also stored within the MPR in background. The basic functionality of the pipeline is therefore unchanged. An opportunity exists to use the

[1] Although the use of the term memristor for these emerging memory elements is still under debate, in this paper, we refer to any device that is nonvolatile, dense, and resistance-based as a memristor.

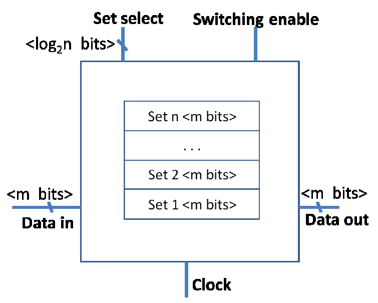

Figure 1: The logic structure of a multistate register. The size of each set is *m* bits, while *n* states are stored. The multistate register is synchronized by a clock and can switch the active set.

MPR in multithreaded processors without the need for additional buffers. In the next section, a novel microarchitecture – Continuous Flow Multithreading (CFMT) [4], is described and evaluated based on a memristive MPR. In CFMT, an MPR stores multiple machine states of different threads, where a single thread is active at a time, enabling higher throughput computing.

III. Continuous Flow Multithreading

Using MPRs as pipeline registers in multithreaded processors can enhance performance by minimizing the switch penalty. In conventional pipelines, the pipeline registers are located between pipeline stages to store the state of the predecessor instructions before moving the state to the next pipeline stage. Conventional pipeline registers are replaced by MPRs, as shown in Figure 3. The use of MPRs instead of regular registers saves the state of the stalled threads in addition to the state of the active thread. The mechanism of thread switching is therefore different from a conventional SoE. Rather than flushing the pipeline, the states of the consecutive instructions are stored within the MPRs, locally near the relevant pipeline stage. Furthermore, rather than refilling the pipeline, instructions from the new active thread are read from the MPRs, significantly reducing the thread switch penalty to the time required to read data from an MPR. Additionally, the novel switching mechanism can contribute to power preserving if reading and writing to the MPR is lower in energy than refilling the entire pipeline and replaying the flushed instructions.

Lowering the thread switch penalty also enables new events to trigger a thread switch. With a conventional SoE MT, it is worthwhile to switch threads on events, when the latency is longer than the time required to refill the pipeline. With CFMT, the condition changes and it is effective to switch threads on events when the latency is longer than the time to switch an active thread within an MPR. This condition enables additional MCEs that in a conventional SoE MT stall the pipeline. This improvement further increases the performance of the processor.

In CFMT, the controller acts similar to a simple conventional SoE MT. The simplicity of CFMT is achieved since an in-order mechanism is used and only a single thread is active within the pipeline at any particular time. While the control mechanism is simple and the energy is low, CFMT offers significantly higher utilization and performance that is comparable to state-of-the-art multithreading mechanisms as an SMT.

IV. Conclusions

Emerging memory technologies are enabling the era of memory intensive computing. Memory intensive architectures use novel memory elements to store data not stored in conventional architectures to enhance performance, while reducing energy.

As an example of memory intensive computing, the combination of a novel memory structure, multistate pipeline register (MPR), with a novel microarchitecture Continuous Flow Multithreading (CFMT) exhibits a 40% performance improvement

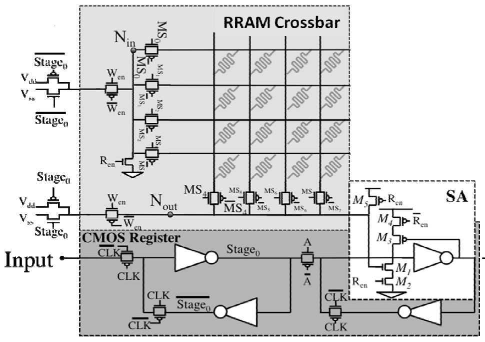

Figure 2: RRAM-based single bit multistate register.

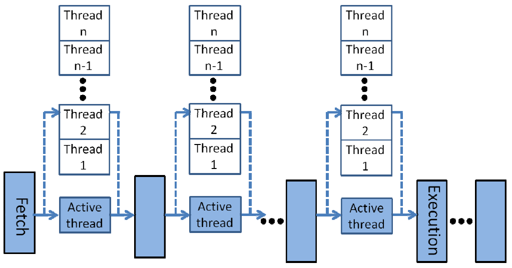

Figure 3: Continuous Flow Multithreading structure. A multistate pipeline register (MPR) is located between each two pipeline stages instead of a conventional pipeline.

with a reduction in energy, supporting the use of CFMT in low power machines.

CFMT is a single example of a memory intensive architecture. Numerous other applications of MPR and other memory elements based on emerging memory technologies are possible, including logic with memristors [7-11]. These novel architectures will improve both performance and energy and extend CMOS by adding complementary technology to CMOS.

References

[1] S. Kvatinsky, E. G. Friedman, A. Kolodny, and U. C. Weiser, "The Desired Memristor for Circuit Designers," *IEEE Circuits and Systems Magazine*, Vol. 13, No. 2, pp. 17-22, second quarter 2013.

[2] L. O. Chua and S. M. Kang, "Memristive Devices and Systems," *Proceedings of the IEEE*, Vol. 64, No. 2, pp. 209- 223, February 1976.

[3] L. O. Chua, "Memristor – The Missing Circuit Element," *IEEE Transactions on Circuit Theory*, Vol. 18, No. 5, pp. 507-519, September 1971.

[4] S. Kvatinsky *et al.*, "Memristor-based Multithreading," *IEEE Computer Architecture Letters*, 2013 (in press).

[5] S. Kvatinsky, *et al.*, "On the In-Die 3D Integration of Memory in CMOS Metal Layers and Its Implications on Processor Microarchitecture," unpublished.

[6] R. Patel, E. G. Friedman, A. Kolodny, and S. Kvatinsky, "Multistate Register Based on Resistive RAM (ReRAM) – A Novel Digital Circuit," *IEEE Transactions on Very Large Scale Integration* (in review).

[7] S. Kvatinsky, *et al.*, "Memristor-based Material Implication (IMPLY) Logic: Design Principles and Methodologies," *IEEE Transactions on Very Large Scale Integration (VLSI)* (in press).

[8] Y. Levy, *et al.*, "Logic Operation in Memory Using a Memristive Akers Array," *Microelectronics Journal* (in press).

[9] S. Kvatinsky, *et al.*, "MAGIC – Memristor Aided LoGIC," *IEEE Transactions on Circuits and Systems II: Express Briefs* (in review).

[10] S. Kvatinsky, E. G. Friedman, A. Kolodny, and U. C. Weiser, "Memristor-based IMPLY Logic Design Flow," *Proceedings of the IEEE International Conference on Computer Design*, pp.142-147, October 2011.

[11] S. Kvatinsky, *et al.*, "MRL – Memristor Ratioed Logic," *Proceedings of the International Cellular Nanoscale Networks and their Applications*, pp. 1-6, August 2012.