2013 16th Euromicro Conference on Digital System Design

Memristor-based (ReRAM) Data Memory Architecture in ASIP Design

Matthias Hartmann*†, Praveen Raghavan†, Liesbet Van Der Perre*†, Prashant Agrawal*†, Wim Dehaene*†
*Department of Electrical Engineering (ESAT), KU Leuven, Kasteelpark Arenberg 10, B-3001 Heverlee, Belgium
†imec vzw, Kapeldreef 75, B-3001 Heverlee, Belgium

***Abstract*—Recently, multiple non-volatile emerging memories (NVMs) have been proposed and show promising properties to replace SRAM-based memories in future SoCs. However, these new emerging memories, such as STT-MRAM and ReRAM, provide new challenges for the processor design e.g. larger write latencies, higher power and lower endurance. In this paper, we propose a design method for memristor-based (ReRAM) memory architectures for embedded processors to address the effects caused by longer write latencies. We evaluate this method and present the design space for using ReRAM in the data memory of an wireless baseband processor. We propose architectural solutions for concealing the slow write speed of ReRAM and show their trade-offs in terms of performance with respect to different write latencies. We show that for single benchmarks the performance penalty caused by the ReRAM write latency can be reduced to 7% for the complete wireless communication benchmark suite. Morevoer, for single benchmarks the performance penalty can be eliminated completely.**

I. Introduction

Embedded memories have become increasingly dominating in terms of performance characteristics of current System-on-Chip (SoC) designs. Especially in mobile applications, the relative leakage power consumption is increasing significantly with each technology node. Multiple emerging, non-volatile memories, such as STT-MRAM [1], PCRAM [2] and ReRAM [3], have been proposed to decrease the leakage power, to improve the scaling with technology nodes and to increase the area density compared to traditional embedded SRAM memories. Nevertheless, these emerging non-volatile memories suffer from other drawbacks such as low write speeds, high write energy consumption, low endurance compared to SRAM.

Most research in the field of non-volatile embedded memories has so far been centered around the design of these new memories and the impact on circuit level [4]. It has been shown that 4Mb ReRAM modules can be implemented in a 65nm CMOS logic compatible process [5]. In addition, several circuit techniques have been investigated to improve the characterisitcs of these new emerging memories [6]. Complementary to this research centered around the cell and ciruit design of emerging memories, this paper focuses on the impact on the design of future embedded processors.

Other research groups have been focusing on the modeling of these new memory technologies ([7], [8]). Whereas these models can be used to study the highlevel impact of integrating these non-volatile memories on system-level, they primarily focus on the memories themselves and not on the complete processor. Moreover, these models do not study the possible impact on the datapath logic.

L1 data memories have higher requirements on the latency. They are critical for the design of future processors due to their high frequency access pattern and have a direct impact on the computational performance of the design. Nevertheless, they also suffer more from the drawbacks of the emerging memory technologies. To the best of our knowledge, this paper is the first to investigate L1 data memories based on ReRAM technology.

This paper discusses the latency challenges of the ReRAM memory technology with the focus on how these issues influence the integration of ReRAM-based memory within the data memory hierarchy of embedded processors. It proposes a method inluding architectural solutions in order to introduce ReRAM-based memories for all data memories including the timing-critical L1 data memories. Finally, we validate our method by presenting the design space for a high-speed, low-energy wireless baseband processor using ReRAM-based data memories.

The rest of this paper is structured as follows. Section II describes the ReRAM technology and its characteristics with reference to the data memory hierarchy of the processors. Section III introduces a method to efficiently embed ReRAM-based data memories within the memory hierarchy of an embedded processor. In Section IV and V, we evaluate the proposed method by presenting the design space for a high-speed, low-energy wireless baseband processor. Section VI concludes the paper.

II. ReRAM Technology

In this paper, the usage of the emerging ReRAM technology in the design of an application-specific instruction set processors (ASIP) is evaluated. Contrary to the typical 6T SRAM cell, in the ReRAM memory the databit is stored in a resistive element instead of four transistors. The 1T1R ReRAM cell consists of one access transistor and a resistor. The resistor is composed of a metal oxide layer between two metal electrodes. In the past, several metal oxide layers have been proposed (e.g. Hf [9]). The data bit is stored by configuring the resistor to a high or low resistive state. It has been shown that this type of emerging memory can provide fast write speeds with a good ratio between the on and off state of the memory [10] and low energy consumption [3].

Nevertheless, the write speeds, achieved with the ReRAM technology, are still too slow to allow single cycle writes in high-speed ASIPs due to their clock speed of 500MHz and

`/13 $26.00 © 2013 IEEE
DOI 10.1109/DSD.2013.138

higher. The write speed of ReRAM memories can be improved by increasing the set/reset voltage applied to the ReRAM module [3]. However, reaching single cycle write speeds will require voltages that not only significantly increase the energy consumption of the memory, but also cause integration issues to embed the memory in a SoC. In order to be able to integrate ReRAM memories in future SoC in terms of their operating voltages, processor designers will need to adapt their designs to multi-cycle write operations. Moreover, additional implementation efforts will also be required on the circuit-level of the memory module. In this paper we will focus on the first issue (multi-cycle write latencies).

Another advantage the ReRAM technology is its lower area footprint compared to SRAM. The ReRAM cell has an area footprint of $36F^2$ compared to the $240F^2$ of the SRAM cell. Moreover, the ReRAM read energy is expected to be lower than SRAM, and the ReRAM cell itself will cause no leakage current. Therefore, the ReRAM memory model will have a smaller leakage power only caused by the periphery of the ReRAM model. Nevertheless, an accurate energy model of the ReRAM is not available at this point. For this reason, this paper will not focus on the energy trade-off space of the proposed ReRAM solutions. We are currently working on an accurate energy model for ReRAM and will investigate the energy trade-off space in our future work.

III. Design Methodology

Memories can be classified based on their hierarchy, their access patterns and their content, whereas each class of memory has different requirements on the actual memory implementation. Off-chip memory has low duty cycle accesses, mostly in burst mode. Therefore, the latency of off-chip memories is less important, but leakage power and area consumption play an important role in the memory design. This paper does not address on off-chip memories, but focuses on on-chip memories, which have a higher constraint on latency scaling with their level in the hierarchy.

Current processors require a high clock frequency due to high computational requirements, and the longer write latencies of ReRAMs will therefore result in multi-cycle write operations. Therefore, processors with ReRAM memories should not only be able to handle multi-cycle write operations either in hardware or in software by the compiler, but also need to implement architectural solutions to prevent a performance drop due to store operations.

Figure 1 shows the proposed methodology for the design of processors with ReRAM-based data memories. Depending on the type of memory different architectural solutions are feasible to ensure the required performance of the processor. Nevertheless, these solutions are restricted to preserve the interface to the datapath, which depends also on other factors (e.g. the exploited data-level parallelism) and a redesign of the datapath is linked to a significant design effort.

Step I: For memories that are **not latency constrained**, a simple drop in replacement of the SRAM memory with ReRAM memory is proposed. Wider interfaces can solve possible throughput issues of the ReRAM modules. Possible conflicting store operation to the ReRAM can be handled by a stall mechanism. Typically, these type of memories can be found in higher levels of the data memory hierarchy (**L2 and higher**).

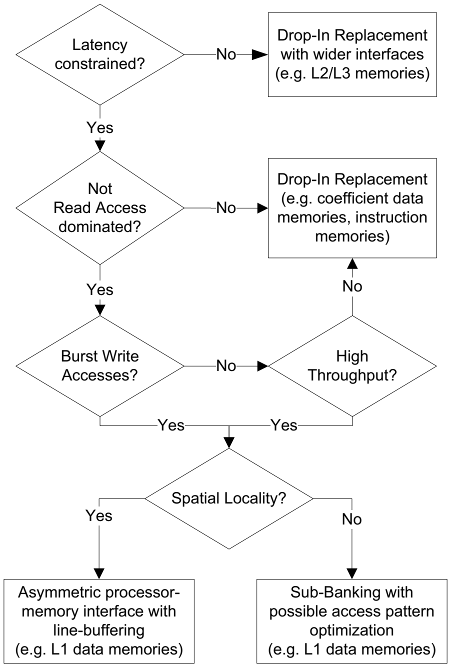
Latency constrained?
No
Drop-In Replacement with wider interfaces (e.g. L2/L3 memories)
Yes
Not Read Access dominated?
No
Drop-In Replacement (e.g. coefficient data memories, instruction memories)
Yes
No
Burst Write Accesses?
No
High Throughput?
Yes
Yes
Spatial Locality?
Yes
No
Asymmetric processor-memory interface with line-buffering (e.g. L1 data memories)
Sub-Banking with possible access pattern optimization (e.g. L1 data memories)

Fig. 1. Proposed Design Methodology

Step II: Latency-constrained memories, which are typically **L1 memories**, can be further split according to their access patterns. If the data memory is **primarily accessed by reads** such as in coefficient memories, then a ReRAM module with the same bandwidth as the SRAM module can be used. The only write operations to these kind of memories occur during the initialization phase of the processor, which in general does not have a high latency constraint.

Step III: The most performance critical memories regarding the utilization of ReRAM technology inside processors are **L1 data memories** that have **burst write accesses**. In this step, write bursts are defined as multiple write accesses to the memories in sequential cycles, but they are not required to have sequential addresses. For this category of memories, a drop in replacement of the SRAM module by a ReRAM-based module with a stall mechanism will cause a significant drop in the performance of the processor. In order to resolve this issue, we propose to consider the spatial locality of the write accesses, which will be done in Step V.

Step IV: In Step IV, memories with non-burst write accesses are addressed. Depending on their access frequency, these memories can either use architectural solutions proposed in Step II or Step V. If the L1 data memory requires a **write throughput** that is higher than the ReRAM module can provide, then the memory is treated the same as a memory with burst write accesses (step III). In the other case, the ReRAM memory can be integrated using the architecural solution of

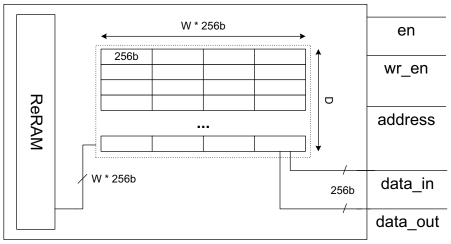

Fig. 2. Datapath with a ReRAM module using line-buffering

step II causing less hardware overhead. Since in the latter case the write accesses are distributed over time, the write operations won't overlap and no significant performance drop will be observed.

Step V: In this last step of the proposed methodology, the **spatial locality** of the memory accesses is considered. If the accesses patterns provide spatial locality, then we propose to implement a line-buffering mechanism for the ReRAM memory module as shown in Figure 2. The processor will read/write an element of a single line from/to the line-buffer, thereby using a narrow interface matching the datapath of the processor. The memory will read and write complete lines to or from the linebuffer, thereby using a wider interface increasing the throughput to the memory.

If the access patterns have no spatial locality, then we propose to use a sub-banking scheme and thereby distributing the store operations to different ReRAM memory instances each with their own stall mechanism and write latency. The efficiency of the sub-banking can be improved, if the designer has knowledge of the access pattern of the application at design time.

IV. Experimental Setup

In order to validate our methodology, several ReRAM-based solutions were implemented and compared to an SRAM-based reference design. The wireless baseband processor used for the validation is optimized to support multiple wireless standards such as WLAN and LTE cat4. It implements one part of a software-defined radio platform requiring a low latency, high throughput datapath in order to meet the strict timing deadlines. The processor design targets a 1GHz clock frequency. The processor has two vector data memories with a 256bit interface for the input and output data as well as a third data memory for local stack memory. In this paper, we will focus on replacing the vector data memories, since these memories are more critical to the performance of the SoC.

A. Application of proposed method

Following the proposed methodology of Section III, the memories were analyzed. Both of them are situated at the lowest level of the data memory hierarchy and are connected directly to the datapath of the processor. Therefore, these memories are performance critical for the SoC and have a latency constraint (**step I**). The next step shows that both memories have balanced write/read patterns and are not primarily used for read-only data (**step II**).

Moreover, the profiling of application domain in terms of access patterns also showed that the writes are either occurring in burst mode (**step III**) or the exploited parallelism in the processor requires a high throughput towards the memory (**step IV**). Therefore, a drop in replacement of the SRAM module with a ReRAM module will not be a valid solution for the given performance constraints. In the last step, the data locality in the application domain is analyzed. Profiling shows that most of the data is written into fixed output buffers in a sequential way with minimal offsets. Therefore, providing a sufficient amount of data locality to enable the efficient usage of a linebuffer with a depth of four (**step V**).

In order to evaluate the performance of the ReRAM modules with multi-cycle writes, different architectural solutions of the processor were modeled and profiled with several kernels of the supported wireless standards. Firstly, we designed a reference design "SRAM" using SRAM memories. The second design "Drop In" replaces the data memories with ReRAM modules with the appropriate stall mechanism. In the third design "2 Banks", both memories utilize a sub-banking scheme with two memory banks using the least significant bit of the address as bank selector. Contrary to that, the design "2 Banks optimized" utilizes an optimized sub-banking scheme for the FFT kernel. The last two designs use a line buffer to mask the write latency of the ReRAM module. Each line in this buffer consists of four 256bit words. In the design "line buffer", a read from the processor to an address that is not in the line buffer will cause a stall cycle in order to fetch this data from the ReRAM memory. In the case of the last design "line buffer with read bypass", the read can bypass the line buffer, which will eliminate the additional stall cycle at the expense of additional hardware and energy.

For each design a SystemC model was implemented and simulated concurrently with the SystemC model of the processor. After the simulation the content of both data memories was verified with the reference output in order to ensure correct stall handling and ReRAM writes.

B. ReRAM memory modeling

A model of the available, inhouse silicon measurements of the ReRAM technology [3] was used to obtain the performance characteristics the ReRAM memory model. For the timing specifications, this paper assumes that both the SRAM and the ReRAM instances can execute a read operation in 1 clock cycle. Nevertheless, the ReRAM has a ten times longer write latency. This specification needs to be handled by the implemented control mechanism.

V. Results

Figure 3 is showing the performance results in terms of cycle counts for each of the designs. The SRAM reference solution is represented with a relative performance of 100%. As predicted, the "Drop In" design results in a significant performance penalty up to a factor of five for single benchmarks and a factor of three totalled overall. This is caused by stalls due to the longer write latency of ReRAM and the burst accesses to the read/write memories of the processor. Both designs using sub-banking decrease the performance overhead compared to the "Drop In" solution, but the overall

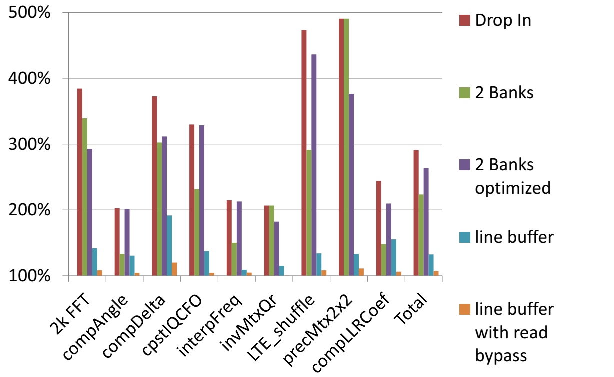

Fig. 3. Performance of different ReRAM solutions and benchmarks relative to the reference SRAM solution (100%)

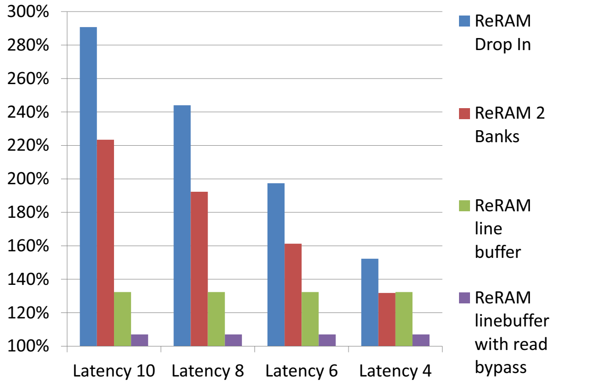

Fig. 4. Performance of different ReRAM solutions depending on the assumed ReRAM write latency in clock cycles

execution time is still factor 2.2 (and 2.6 respectively) longer than the reference SRAM design. In addition, for some specific benchmarks we observe no improvement due to non-beneficial memory access patterns. Moreover, an optimized sub-banking for a specific kernel will improve performance in this kernel (2k FFT) significantly, but might cause a larger overhead overall (Design "2 Banks optimized").

Finally, the two designs using line-buffering reduce the performance penalty of ReRAM memories to 30% for the Design "line buffer" and only 7% for the design with a read bypass compared to all benchmarks. For single benchmarks such as "invMtxQr" the overhead can be completely eliminated for the last design. Nevertheless, both designs also require a significant amount of additional logic which will results in poorer area and energy efficiencies of these designs. As previously mentioned both of these metrics are out of the scope of this paper due to the still ongoing research on circuit design and processing options for embedded ReRAM memory modules. Future research will address the tradeoff space between the proposed designs and energy and area consumption of the SoC.

As previously discussed in Section II, the ReRAM write latency can be decreased by increasing the operational voltage of the ReRAM memory module. Depending on the development of future ReRAM cells, the actual write latency of ReRAM cells might be reduced. On the other hand, embedded systems with lower clock frequencies will also not require a write latency of ten cycles for ReRAM modules. In Figure 4, we investigated the effect of different ReRAM write latencies on the tradeoff space for the complete benchmark.

The results show that the performance overheads for the Designs "Drop In" and "2 Banks" are reduced when the write latency is lowered. Nevertheless, the designs using line-buffering show no significant improvement. These results show that for SoC with a lower clock frequency a similar performance using sub-banking rather than the more expensive line buffering can be achieved.

VI. Conclusion

In this paper, we propose a methodology to utilize ReRAM technology for the data memories of processors and propose solutions for the challenges of this emerging memory technology. The methodology is evaluated on a set of benchmarks for a wireless baseband processor and the paper shows trade-off points in terms of performance. For some design points, the performance overhead caused by the introduction of ReRAM can be reduced to 7% on average and even completely elimintated for specific benchmarks. Furthermore, the paper shows that SoC with more relaxed timing constraints will suffer smaller penalties and enabling the usage of less expensive micro-architectural solutions masking the ReRAM write latency.

References

[1] M. Hosomi *et al.*, "A novel nonvolatile memory with spin torque transfer magnetization switching: spin-ram," in *Electron Devices Meeting, 2005. IEDM Technical Digest. IEEE International*, dec. 2005, pp. 459 –462.

[2] S. Raoux *et al.*, "Phase-change random access memory: A scalable technology," *IBM Journal of Research and Development*, vol. 52, no. 4.5, pp. 465 –479, july 2008.

[3] B. Govoreanu *et al.*, "10x10nm2 hf/hfox crossbar resistive ram with excellent performance, reliability and low-energy operation," in *Electron Devices Meeting (IEDM), 2011 IEEE International*, dec. 2011, pp. 31.6.1 –31.6.4.

[4] M.-F. Chang *et al.*, "Circuit design challenges in embedded memory and resistive ram (rram) for mobile soc and 3d-ic," in *Design Automation Conference (ASP-DAC), 2011 16th Asia and South Pacific*, jan. 2011, pp. 197 –203.

[5] ——, "A 0.5v 4mb logic-process compatible embedded resistive ram (reram) in 65nm cmos using low-voltage current-mode sensing scheme with 45ns random read time," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2012 IEEE International*, feb. 2012, pp. 434 –436.

[6] P. Zhou *et al.*, "Energy reduction for stt-ram using early write termination," in *Computer-Aided Design - Digest of Technical Papers, 2009. ICCAD 2009. IEEE/ACM International Conference on*, nov. 2009, pp. 264 –268.

[7] Y. Xie, "Modeling, architecture, and applications for emerging memory technologies," *Design Test of Computers, IEEE*, vol. 28, no. 1, pp. 44 –51, jan.-feb. 2011.

[8] X. Dong and others., "Nvsim: A circuit-level performance, energy, and area model for emerging nonvolatile memory," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 31, no. 7, pp. 994 –1007, july 2012.

[9] Y. Chen *et al.*, "Highly scalable hafnium oxide memory with improvements of resistive distribution and read disturb immunity," in *Electron Devices Meeting (IEDM), 2009 IEEE International*, dec. 2009, pp. 1 –4.

[10] M.-F. Chang *et al.*, "Circuit design challenges in embedded memory and resistive ram (rram) for mobile soc and 3d-ic," in *Design Automation Conference (ASP-DAC), 2011 16th Asia and South Pacific*, jan. 2011, pp. 197 –203.