Quality-Configurable Memory Hierarchy Through Approximation

Conference Paper · October 2017

CITATIONS
0

READS
95

3 authors, including:

Amir M. Rahmani
UC Irvine (USA) & TU Wien (Austria)
152 PUBLICATIONS **597** CITATIONS
SEE PROFILE

Nikil Dutt
University of California, Irvine
623 PUBLICATIONS **11,459** CITATIONS
SEE PROFILE

Special Session: Quality-Configurable Memory Hierarchy Through Approximation

Majid Shoushtari
Department of Computer Science
University of California, Irvine, CA
anamakis@uci.edu

Amir M. Rahmani
TU Wien, Vienna, Austria
University of California, Irvine, CA
amirr1@uci.edu

Nikil Dutt
Department of Computer Science
University of California, Irvine, CA
dutt@uci.edu

ABSTRACT

The memory subsystem is a major contributor to the overall performance and energy consumption of embedded computing platforms. The emergence of "killer" applications such as data-intensive recognition, mining, and synthesis (RMS) applications puts even more stress on the memory subsystem and exacerbates its energy consumption. Traditional mechanisms to ensure data integrity deploy overdesign (e.g., redundancy and error detection/correction) and/or guard-banding that consumes a significant part of the energy consumed in the memory subsystem. We explore opportunities for energy efficiency by exploiting the intrinsic tolerance of a vast class of approximate computing applications to some level of error in the on-chip memory hierarchy. We present two exemplars outlining the typical software and hardware mechanisms that are required for different components in the memory hierarchy, implemented in varying technologies such as SRAM and STT-MRAM.

KEYWORDS

Approximate Computing, Cross-layer Design, Memory Hierarchy

1 INTRODUCTION

Approximate computing trades accuracy for performance or efficiency, exploiting the fact that many applications are robust to some level of imprecision. Approximate computing is a prime candidate for exploiting cross-layer design, as these tradeoffs can be orchestrated across multiple levels of design abstraction. It is an accepted fact that neither software nor hardware alone can unlock the full potential of approximation. Unfortunately, contemporary design flows for approximate computing fail to exploit fully the performance and energy efficiency opportunities achievable by crossing this abstraction gap. The memory subsystem presents an opportunity to reason about imprecision, resilience and tradeoffs across the abstraction stack, as it is possible to trace the relationship between memory objects at different levels, from the lowest hardware abstraction (e.g., circuits and logic) up to the highest levels (e.g., applications) [12]. Accordingly, we focus on the memory subsystem and broadly discuss opportunities for quality-configurable approximation using two exemplars for quality-configurable caches.

CASES '17 Companion, Seoul, Republic of Korea

DOI: 10.1145/3125501.3125525

2 RESEARCH AND PRACTICES ON APPROXIMATE MEMORY MANAGEMENT

Previous research on approximate memory management can be classified into multiple categories based on a number of criteria as detailed in our technical report [24]. Here we summarize some representative approaches.

Depending the specific memory hierarchy component that is subject to approximate storage, different layers of the system abstraction come into play. Table 1 depicts the abstraction layers covered by recent efforts, as well as their design goals. It can be seen that many of the previous work leverage multiple layers of abstraction. While majority of approaches target reducing energy consumption as the main objective, some approaches attempt in increasing the performance and/or increasing the lifetime of the memory substrate. Approximations can be applied to memory structures abstractions such as L1/L2 cache [11] [22] [20] [19] [10] [15] [7] [13] [21], scratchpad memory [2], main memory [16] [3] [17], secondary storage [25] [4] [14] [6] [25] in a single-layer or cross-layer fashion. These techniques have three main objectives: making a trade-off between the quality of the generated output and (1) energy consumption [1] [2] [11] [16] [3] [22] [20] [10] [19] [17] [15] [9] [18] [7] [25] [21], (2) performance of the memory subsystem [5] [4] [22] [19] [15] [23] [14] [6] [25] [13] [8] [21], and (3) improving their lifetime [17] [4].

These objectives are often dependent on the memory technology used. Different memory technologies have different limitations, for example, SRAM and (e)DRAM consume high leakage and refresh power, respectively, while NVMs have high write energy/latency

Table 1: Approximate Memory Techniques

Related Work	Abstraction Layer					Goal		
	Application/ Algorithm	Compiler/ ISA	OS / Runtime	Architecture	Device	Energy	Perf.	Lifetime
[11]			■	■	■	■		
[1]			■	■	■	■		
[10]				■	■	■		
[16]			■	■		■		
[4]		■	■		■		■	■
[22]					■	■	■	
[2]			■	■	■	■		
[3]			■	■		■		
[5]		■		■			■	
[20]				■		■		
[23]	■						■	
[18]				■		■		
[19]				■		■	■	
[15]				■	■	■		
[9]				■		■		
[17]				■		■		■
[7]				■		■		
[14]	■				■		■	
[6]	■					■		
[25]			■	■			■	
[13]	■			■			■	
[8]	■				■	■	■	
[21]		■		■		■	■	

and/or low write endurance. To reduce energy consumption of these memories and improve the lifetime of NVMs, approximate storage techniques sacrifice data integrity by reducing supply voltage in SRAM [11] [15] and refresh rate in (e)DRAM [16] [3] [9] and by relaxing or skipping read/write operation in NVMs [25] [17] [4] [2] [1] [22] [7] [14] [6].

These objectives are achieved by a number of general strategies, namely: (1) precision scaling [20] [18], (2) approximating load operations [5] [19] [21], (3) skipping store operations [5] [17] [13], (4) using faulty or unprotected memory substrate [22] [4] [10] [15] [7] [14] [6] [25], (5) tweaking technology-dependent reliability-energy knobs [1] [11] [16] [3] [2] [4] [9].

3 EXEMPLARS OF QUALITY-CONFIGURABLE MEMORIES

3.1 Relaxed Cache

Relaxed cache [11] is a technique for saving leakage energy in SRAM-based caches. In Relaxed Cache a subset of the cache ways are protected (using error detection/correction and/or proper supply voltage settings). Other cache ways (called relaxed ways) operate with a lower supply voltage and therefore may become faulty. The programmer identifies noncritical data objects using suitable annotations, and the virtual addresses of these objects are kept in a table. On a cache miss, if the missed data block is noncritical, a victim block is selected from the relaxed ways. A critical data item is always stored in a fault-free block.

To control the degree of approximation, any cache block that has more than a certain acceptable number of faulty bits (called AFB) is disabled and power-gated. AFB along with VDD are two knobs used in this work to determine the effective capacity of the cache and also bound the overall error exposed to software. The experimental results show that Relaxed Cache save significant amount of cache leakage energy (up to 74%) while still generating acceptable quality results.

3.2 QuARK

QuARK [1] is a hardware/software approach for trading reliability of STT-MRAM caches for energy savings in the on-chip memory hierarchy of systems running approximate applications. QuARK utilizes fine-grained actuation knobs to efficiently control reliability-energy trade-offs for individual accesses of concurrently running applications.

Compared to the related work on SRAM caches [11] that uses cache-way-level knobs, QuARK presents a more fine-grained actuation capability enabling it to offer the following advantages: i) the knob actuations do not affect any other cache block unlike the actuation in [11] which requires flushing all the affected blocks, ii) multiple applications with different degrees of reliability can share the same cache without affecting each other's guaranteed level of reliability, and iii) the reliability level requested for a piece of data can be changed at runtime, if the quality of output is not satisfactory. The simulation results demonstrated up to 40% energy savings over a fully-protected STT-MRAM L2 cache, with acceptable quality loss.

4 CONCLUSIONS AND FUTURE DIRECTIONS

In this paper, we presented state-of-the-art approximate memory management approaches and discussed opportunities for energy savings in the on-chip memory hierarchy. We believe the gains from approximation can be maximized with cross-layered techniques that can match the possibilities in one layer with opportunities in the other layers. However, to realize such opportunities, we need to cope with open challenges such as: i) how to develop programming models to capture the resiliency of data objects of a program, ii) how to automate classification of data objects based on their resiliency to memory errors, iii) how to achieve a holistic approach to guide approximation across the entire memory hierarchy, iv) how to orchestrate memory approximation knobs at different levels of abstraction, and v) how to control the approximation process of the memory subsystem to bound errors propagated to applications.

ACKNOWLEDGMENT

We acknowledge financial support by the Marie Curie Actions of the European Union's H2020 Programme.

REFERENCES

[1] A. M. Monazzah et al. 2017. QuARK: Quality-configurable Approximate STT-MRAM Cache by Fine-grained Tuning of Reliability-Energy Knobs. In *Proc. of ISLPED*.
[2] A. Ranjan et al. 2015. Approximate Storage for Energy Efficient Spintronic Memories. In *Proc. of DAC*.
[3] A. Raha et al. 2017. Quality Configurable Approximate DRAM. *IEEE Trans. Comput.* (2017).
[4] A. Sampson et al. 2013. Approximate Storage in Solid-state Memories. In *Proc. of MICRO*.
[5] B. Thwaites et al. 2014. Rollback-free Value Prediction with Approximate Loads. In *Proc. of PACT*.
[6] D. Jevdjic et al. 2017. Approximate Storage of Compressed and Encrypted Videos. In *Proc. of ASPLOS*.
[7] F. Sampaio et al. 2015. Approximation-aware Multi-Level Cells STT-RAM Cache Architecture. In *Proc. of CASES*.
[8] G. P. Arumugam et al. 2015. Novel Inexact Memory Aware Algorithm Co-design for Energy Efficient Computation: Algorithmic Principles. In *DATE*.
[9] K. Cho et al. 2014. eDRAM-based Tiered-Reliability Memory with Applications to Low-power Frame Buffers. In *Proc. of ISLPED*.
[10] K. Lee et al. 2006. Mitigating Soft Error Failures for Multimedia Applications by Selective Data Protection. In *Proc. of CASES*.
[11] M. Shoushtari et al. 2015. Exploiting Partially-Forgetful Memories for Approximate Computing. *IEEE Embedded Systems Letters* (2015).
[12] N. Dutt et al. 2014. Multi-Layer Memory Resiliency. In *Proc. of DAC*.
[13] O. Kislal et al. 2016. Cache-Aware Approximate Computing for Decision Tree Learning. In *Proc. of IPDPSW*.
[14] Q. Guo et al. 2016. High-Density Image Storage Using Approximate Memory Cells. In *Proc. of ASPLOS*.
[15] S. Ganapathy et al. 2015. Mitigating the Impact of Faults in Unreliable Memories for Error-resilient Applications. In *Proc. of DAC*.
[16] S. Liu et al. 2011. Flikker: Saving DRAM Refresh-power Through Critical Data Partitioning. In *Proc. of ASPLOS*.
[17] Y. Fang et al. 2012. SoftPCM: Enhancing Energy Efficiency and Lifetime of Phase Change Memory in Video Applications via Approximate Write. In *Proc. of ATS*.
[18] Y. Tian et al. 2015. ApproxMA: Approximate Memory Access for Dynamic Precision Scaling. In *Proc. of GLSVLSI*.
[19] J. S. Miguel et l. 2014. Load Value Approximation. In *IEEE/ACM International Symposium on Microarchitecture*.
[20] J. S. Miguel et l. 2015. DoppelgÄNger: A Cache for Approximate Computing. In *Proc. of MICRO*.
[21] J. S. Miguel et l. 2016. The Bunker Cache for Spatio-value Approximation. In *Proc. of MICRO*.
[22] F. Oboril, A. Shirvanian, and M. Tahoori. 2016. Fault Tolerant Approximate Computing using Emerging Non-volatile Spintronic Memories. In *Proc. of VTS*.
[23] Martin Rinard. 2013. Parallel Synchronization-Free Approximate Data Structure Construction. In *5th USENIX Workshop on Hot Topics in Parallelism*.
[24] Majid Shoushtari and Nikil Dutt. 2017. *A Survey of Techniques for Approximate Memory Management.* Technical Report CECS-TR-17-03. Center for Embedded and Cyber-physical Systems, University of California, Irvine.
[25] X. Xu and H. H. Huang. 2015. Exploring Data-Level Error Tolerance in High-Performance Solid-State Drives. *IEEE Transactions on Reliability* (2015).