WIPE: Wearout Informed Pattern Elimination to Improve the Endurance of NVM-based Caches

Sina Asadi, Amir Mahdi Hosseini Monazzah, Hamed Farbeh, Seyed Ghassem Miremadi

Presented by:

Sina Asadi

Dependable System Laboratory (DSL)

Department of Computer Engineering

Sharif University of Technology

Tehran, Iran 11155-9517

Outline

- Introduction
  - SRAM vs. NVMs
  - NVMs Challenges
- Related Work
  - Write Avoidance Techniques
  - Wear-Leveling Techniques
- Proposed WIPE Technique
  - Motivation
  - Proposed Technique
- Results
  - Endurance
  - Performance
  - Energy Consumption
- Conclusions

Introduction

- SRAMs are previlant memory technology in cache memories [1]!
  - Low access latecy
  - High scalability
  - Design simplicity
- Emerging challenges appear in SRAMs with technology scaling trend
  - High static energy
  - Low density
  - Reliability challenges
  - Etc.

Introduction (Cont.)

- A body of research has been triggered to find SRAMs alternative technologies
- NVMs are very promising between the alternative memory technologies [ITRS]
- In comparison with SRAMs, NVMs benefit from
  - Negligible static energy
  - Higher density
  - Higher scalability

But!
The main common drawbacks of all NVMs are endurance limitation and high write energy consumption

Related Work

- There are several studies tackle the mentioned NVMs challenges
  - From endurance point of view
  - From energy consumption point of view
- General approach to mitigate the NVMs endurance challenges
  - Reducing the number of write operations in NVM cells
- Cache write operations reduction techniques in literature
  - Write avoidance techniques [3][5-8]
  - Wear-leveling techniques [7][9][10]

Related Work – Write Avoidance

- Trying to eliminate write operations in the NVM cells
  - By eliminating the write operations in the cells that previously saved values are same as the new incoming values [11]!
  - By serving the write hot spot blocks from the memory cells with high endurance memory technology, i.e., SRAM [3][5][6][14]!
- These approaches can be implemented at
  - circuit-level [8]
  - architecture-level [3][5-10]

Related Work – Wear-Leveling

- Trying to distribute write operations across the NVM cells and wear out them uniformly!
- Considering an NVM-based associative cache
  - The literature techniques can be categorized in four groups
    - Bit-level [7]
    - Intra-block level [15]
    - Intra-set level [9][10]
    - Inter-set level [9]

WIPE: Motivation

- A number of data patterns in L2 cache are noticibly written more frequently!
- From the endurance point of view
  - These patterns have more contribution in wearout of NVM cells.

Examples of data patterns in different benchmarks

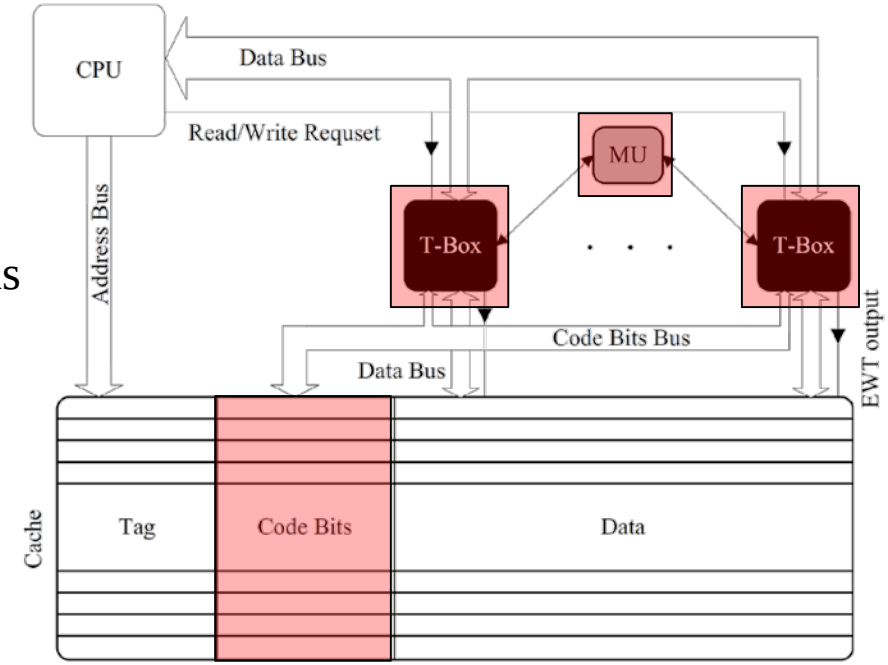
Benchmarks	32 bit data pattern (hexadecimal)				% of iterations in writes
h264ref	0	159	0	159	14
xalancbmk	0	77	184	176	10
cactusADM	102	102	102	102	9
lbm	28	113	199	28	30
soplex	63	240	0	0	15

WIPE: Motivation (Cont.)

- Frequent write patterns can be classified in to two groups:
  - Deterministic Write Patterns (DWPs)
    - These patterns are common between all of the workloads
    - Can be determined by static profiling
  - Non-deterministic Write Patterns (NWPs)
    - These patterns are only appeared in specific workloads
    - Can’t be determined by static profiling
    - May be changed during the execution of each workload!
- Both of DWPs and NWPs play an important role in wearing out of NVM cells!

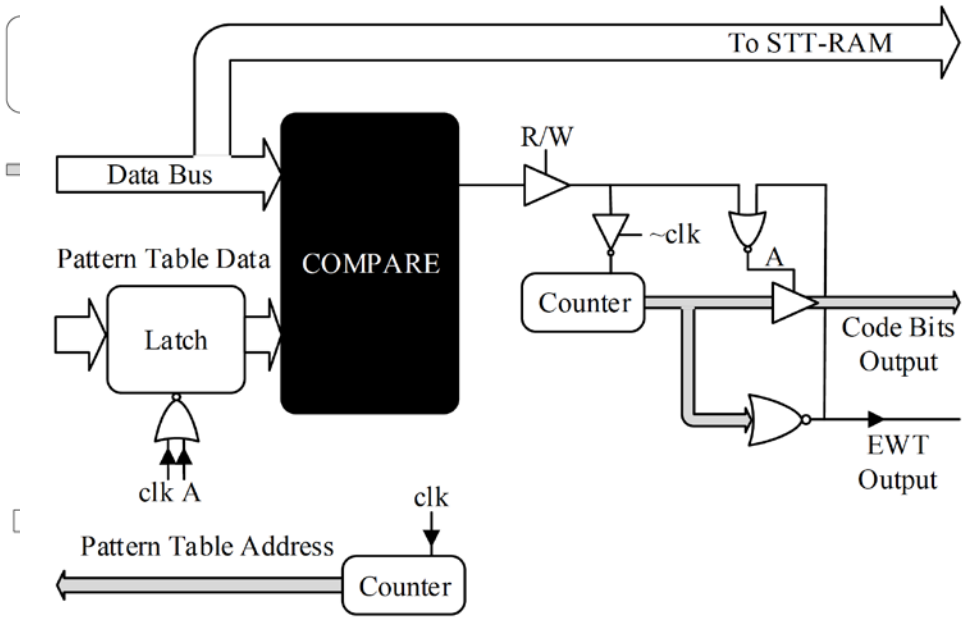
WIPE: How it works?

- We propose a Wearout Informed Pattern Elimination (WIPE) technique
  - Dynamically eliminates both DWPs and NWPs write patterns across the workloads
    - To improve the endurance of Last level Cache (LLC)
- WIPE enabled cache architecture
  - Benefits from Management Unit (MU)
    - To dynamically track the write patterns
  - Benefits from T-Boxes
    - To store the frequent write patterns



WIPE: T-Box

- A T-Box contains a small Pattern Table
  - Storing seven most frequent DWPs and NWPs in recent write operations!
- WIPE use one T-Box for each word width of LLC block width!



- T-Box is updated by MU
- How write operation performs?
- How read operation performs?

Results: System Setup

- Simulation environment
  - Simulator: gem5
  - NVM: STT-RAM
  - Power and latency model
    - Cache: NVSim
    - Prepherals: Synopsys DC
  - Benchmarks: SPEC CPU2006
  - Considered baseline for evaluations
    - Conventional STT-RAM L2 cache

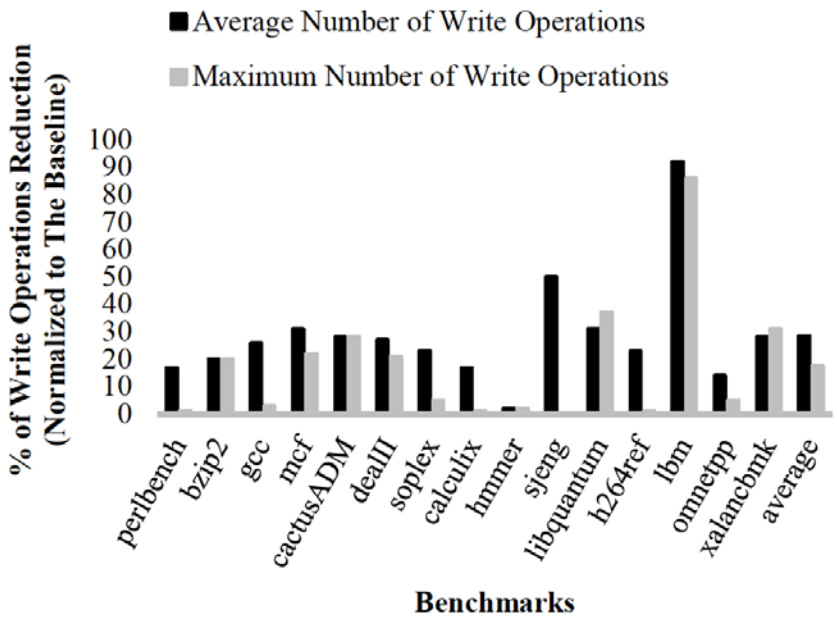
ARM

Simulation configurations

	Parameter	Value
L1 Caches (SRAM)	Frequency	1 GHz
	CPU Model	arm_detailed
	Associativity	4
	Size	32KB
	Block Size	64B
L2 Caches (STT-RAM)	Associativity	8
	Size	256KB
	Block Size	64B

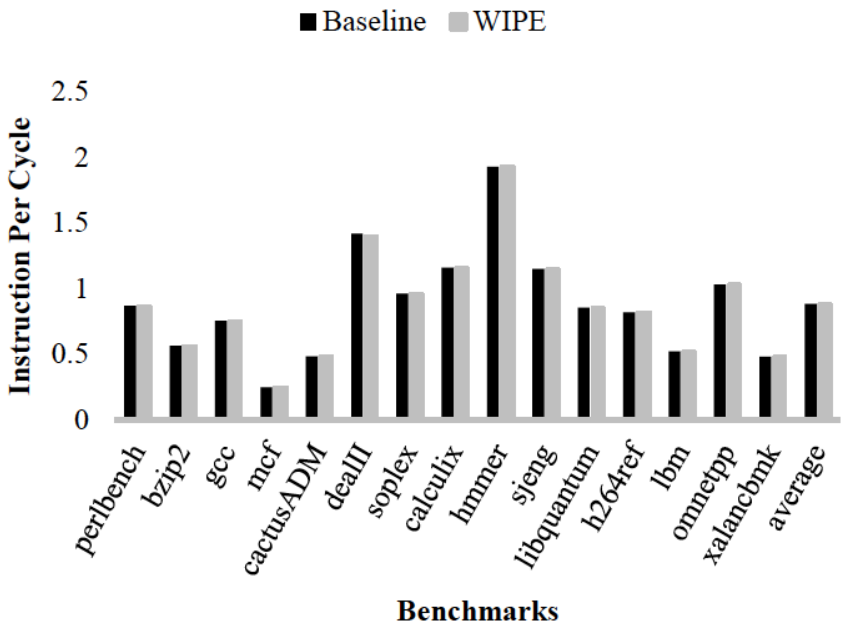
Results: Endurance

- We evaluate the endurance improvement in WIPE by two metrics:
  - The maximum number of write operations between all of the cache blocks
    - 18% improvement observed on average
  - The average number of write operations over all of the cache blocks
    - 30% improvement observed on average



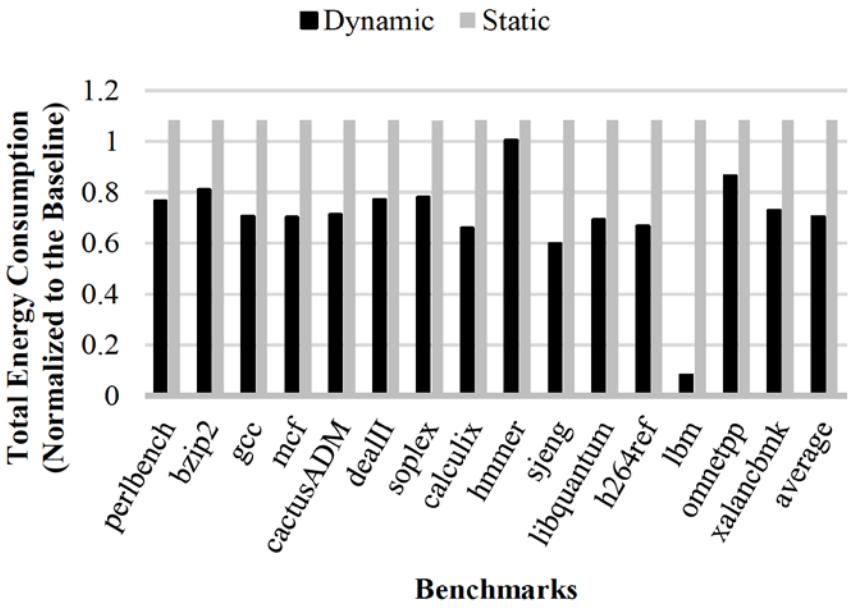
Results: Performance

- WIPE does not impose any clock cycle(s) to read/ write operations!
- The only performance penalty in WIPE
  - Cache flushing operations
    - During pattern table updates
- On average, WIPE imposes only 0.2% performance overhead to system!



Results: Energy and Area

- WIPE significantly reduces the number of write operations in NVM cells
  - Write operations consume significant energy in NVM cells
    - WIPE significantly reduces dynamic energy consumption!
    - 29% dynamic energy consumption reduction observed!
- Static energy consumption overheads of WIPE are 8.4%, on average
- WIPE's peripheral circuits impose 9% area overhead to L2 cache design



Simulation results: Conclusion

- We propose endurance improvement technique for L2 cache (LLC)
  - Called “Wearout Informed Pattern Elimination (WIPE)”
- WIPE snoops the write operations in NVM-based cache
  - Determines the frequent write patterns (DWPs and NWPs)
  - Prevents DWPs and NWPs to be written in NVM-based cache
- WIPE improves endurance of NVM-based cache by 30%, on average.
- WIPE improves dynamic energy of NVM-based cache by 29%, on average.
- The static energy consumption overhead of WIPE is about 8.4%.
- The area overhead of WIPE peripheral circuits and modules is 9%.
- Performance overhead of WIPE is negligible.

Thanks for your attention!