<Write your own unique Title that describes your submission using general as well as specific terms>

**Seth Levine**

Department of Electrical and Computer Engineering

University of Central Florida

Orlando, FL 32816-2362

*Abstract*—Abstract is a 100-word to 200-word summary of the entire paper. It should explain that the objective of the paper is to evaluate some of the fundamental metrics for selected *cache configuration*. List the various types of metrics that you located in the papers which you selected. Name only a couple of the most interesting cache designs that you examined.

Keywords—Keywords are terms used in the paper. For example: SRAM, Non-Volatile Memory, etc.

# Introduction

Write an overview of the topic of cache configuration that you have investigated in a few paragraphs. Explain why multilevel caches are important. Identify various types of cache block placements such as direct-mapped, full-associative, or set-associative. In addition, mention which of the device technologies are volatile or non-volatile (e. g., SRAM and eDRAM are volatile but STT-RAM is non-volatile).

Write a paragraph to explain how the contents of cache are accessed in direct-mapped and set-associative cache strategies. Identify the different fields and/or hardware components utilized inside most caches. Illustrate the cache implementation with a sketch of the cache design similar to a diagram from lecture slide. Write a paragraph to explain the terms: miss ratio and hit ratio.

Finish with a paragraph containing a few sentences that mention what the forthcoming sections of this paper are. Such as, in Section 2 there are ten cache configurations spanning from the year 2000 until today that are reviewed.

# Literature Review

In this section, review the data for the metrics identified in the introduction with a sufficient number of papers. For original work in this class not present elsewhere on the web, pick **a total of 4 baseline computer systems** as listed on the next page from the first group of references. Search each paper to locate the desired metric that you wish to determine for each computer system. Next, do the same for a total of 4 other computer systems such as those listed in the second section of references. You will receive extra credit if you complete an additional 2 rows for another paper from the baseline list and another paper from the comparison list. You can click on the link in webcourses project resources in the lecture of course to open the electronic version, then type the term that you are looking to find in the search box, or use CTRL-F to search, for example for cache configuration, L1/L2/LLC, or processor. To organize your results chronologically, group these by 5 year intervals: so you have one paragraph for 2015 – 2010, one paragraph for 2009 to 2004, etc.

In each paragraph, identify which types of technology node and fabrication technology are popular. Use a sentence to describe each system by listing the benefits of each fabrication technology compared to others. For example, “Over past decade, STT-RAM technology has received significant attention as an alternative replacement for SRAM-based cache design [3]. Results have shown …”

# Data Analysis

In this section you can use the values for the metrics that you located in the papers to fill out Table 1. From these, you should be able to draw some plots (Figures) using Mircosoft excel based on the table and then just discuss the trends briefly. You should refer within the narrative text to the each table and figure to guide the reader as to what each contains.

**Metrics covered by various papers which are suitable for plotting:**

* Area overhead: STT-RAM vs. SRAM/eDRAM (if provided in the paper)
* Energy consumption: STT-RAM vs. SRAM/eDRAM (if provided in the paper)
* IPC: STT-RAM vs. SRAM/eDRAM (if provided in the paper)
* Cache latency: STT-RAM vs. SRAM/eDRAM (if provided in the paper)

1. <Write a Caption in your own words below each Figure.>

# Conclusion

A single conclusion paragraph should be provided at the end to discuss some trends you observed. The questions to be answered in this study are: Can you identify some of the cache designs mentioned in Module-11? Or if not, state what you did find, etc.

##### References

References should be numbered consecutively within square brackets [1]. The sentence punctuation in the text that cites, for example, use of the second reference would be noted at the end of the sentence as [2]. Finally, make sure all your references are numbered in sequence for [1] to [8], and that each of them actually appears as a reference used inside the text itself.

Finally, **delete or replace all of the red text** in the entire document before making your final submission.

Some **baseline designs** that are familiar to the graders and clearly list the metrics to write about:

1. S. E. Crawford and R. F. DeMara, "Cache coherence in a multiport memory environment," *in Proceedings of the Second International Conference on Massively Parallel Computing Systems (MPCS-95)*, pp. 632-642, Ischia, Italy, May 2-6, 1995.
2. N. Khoshavi, X. Chen, J. Wang and R. F. DeMara, “Bit-Upset Vulnerability Factor for eDRAM Last Level Cache Immunity Analysis,” *Proceedings of 17th International Symposium on Quality Electronic Design (ISQED 2016),* Santa Clara, CA, USA, March 15 - 16, 2016.
3. X. Chen, N. Khoshavi, J. Zhou, D. Huang, R. F. DeMara, J. Wang, W. Wen and Y. Chen, “AOS: Adaptive Overwrite Scheme for Energy-Efficient MLC STT-RAM Cache,” *53rd Design Automation Conference,* Austing, TX, USA, 2016.
4. B. Motlagh, and R. F. DeMARA. "Performance of Scalable Shared-Memory Architectures" Journal of Circuits, Systems, and Computers 10.01n02 (2000): 1-22.
5. M. Lin, et al. "ASTRO: Synthesizing application-specific reconfigurable hardware traces to exploit memory-level parallelism" *Microprocessors and Microsystems 39.7* (2015): 553-564.

Some **comparison designs** you could contrast to:

1. A. Jog, A. K. Mishra, C. Xu, Y. Xie, V. Narayanan, R. Iyer, and C. R. Das, “Cache Revive: Architecting Volatile STT-RAM Caches for Enhanced Performance in CMPs,” *in Proceedings of 49th Annual Design Automation Conference (DAC)*. 2012, pp. 243–252.
2. A. Jaleel, M. Mattina, and B. Jacob, “Last Level Cache (LLC) Performance of Data Mining Workloads on a CMP-a Case Study of Parallel Bioinformatics Workloads,” *in Proceedings of 12th International Symposium on High Performance Computer Architecture (HPCA)*, 2006, pp. 88–98.
3. Z. Sun, X. Bi, H. H. Li, W.-F. Wong, Z.-L. Ong, X. Zhu, and W. Wu, “Multi Retention Level STT-RAM Cache Designs with a Dynamic Refresh Scheme,” *in Proceedings of 44th annual IEEE/ACM International Symposium on Microarchitecture*. 2011, pp. 329–338.
4. M.-T. Chang, P. Rosenfeld, S.-L. Lu, and B. Jacob, “Technology Comparison for Large Last-level Caches (L 3 Cs): Low-leakage SRAM, Low Write-energy STT-RAM, and Refresh-optimized eDRAM*,” in Proceedings of 19th International Symposium on High Performance Computer Architecture (HPCA)*, 2013, pp. 143–154.
5. Z. Sun, X. Bi, and H. Li, “Process variation aware data management for stt-ram cache design,” *in Proceedings of the 2012 ACM/IEEE International Symposium on Low Power Electronics and Design, ISLPED,* 2012, pp. 179–184.
6. M. R. Jokar, M. Arjomand, and H. Sarbazi-Azad, “Sequoia: High-Endurance NVM-Based Cache Architecture,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2016.
7. D. Chandra, et al. “Predicting inter-thread cache contention on a chip multi-processor architecture” *11th International Symposium on High-Performance Computer Architecture*, 2005.
8. M. K. Qureshi, and Y. N. Patt. “Utility-based cache partitioning: A low-overhead, high-performance, runtime mechanism to partition shared caches” *Proceedings of the 39th Annual IEEE/ACM International Symposium on Microarchitecture*, 2006.
9. X. Wu, et al. “Hybrid cache architecture with disparate memory technologies” *ACM SIGARCH computer architecture news*. Vol. 37. No. 3. ACM, 2009.
10. J. Huh, et al. "A NUCA substrate for flexible CMP cache sharing." *IEEE transactions on parallel and distributed systems* 18.8 (2007): 1028-1040.
11. M. Maghsoudloo, H. Zarandi “Design space exploration of non-uniform cache access for soft-error vulnerability mitigation” *Microelectronics Reliability* 55.11 (2015): 2439-2452.
12. M. K. Qureshi, D. Thompson, and Y. N. Patt. “The V-Way cache: demand-based associativity via global replacement” *32nd International Symposium on Computer Architecture (ISCA'05)*, 2005.
13. Manikantan, Raman, Kaushik Rajan, and Ramaswamy Govindarajan. "Probabilistic shared cache management (PriSM)." *ACM SIGARCH computer architecture news. Vol. 40. No. 3*, 2012.
14. Parihar, Raj, et al. "Protection, utilization and collaboration in shared cache through rationing." URL http://www. cs. rochester. edu/u/cding/Documents/Publications/tr-ration. pdf (2013).
15. Wang, Jianxing, et al. "A coherent hybrid SRAM and STT-RAM L1 cache architecture for shared memory multicores." *19th Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2014.
16. Ahn, Junwhan, Sungjoo Yoo, and Kiyoung Choi. "DASCA: Dead write prediction assisted STT-RAM cache architecture." *IEEE 20th International Symposium on High Performance Computer Architecture (HPCA)*, 2014.
17. Yazdanshenas, Sadegh, et al. "Coding last level STT-RAM cache for high endurance and low power." *IEEE computer architecture letters* 13.2 (2014): 73-76.
18. Joo, Yongsoo, and Sangsoo Park. "A hybrid PRAM and STT-RAM cache architecture for extending the lifetime of PRAM caches." *IEEE computer architecture letters* 12.2 (2013): 55-58.
19. Kim, Namhyung, and Kiyoung Choi. "Exploration of trade-offs in the design of volatile STT–RAM cache." *Journal of Systems Architecture* (2016).
20. Li, Qingan, et al. "Compiler-assisted STT-RAM-based hybrid cache for energy efficient embedded systems." *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 22.8 (2014): 1829-1840.
21. Mao, Mengjie, et al. "Coordinating prefetching and STT-RAM based last-level cache management for multicore systems." *Proceedings of the 23rd ACM international conference on Great lakes symposium on VLSI*, 2013.
22. Li, Jianhua, et al. "Low-energy volatile STT-RAM cache design using cache-coherence-enabled adaptive refresh." *ACM Transactions on Design Automation of Electronic Systems (TODAES)* 19.1 (2013): 5.
23. Mao, Mengjie, et al. "Prefetching techniques for STT-RAM based last-level cache in CMP systems." *19th Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2014.
24. Zhang, Yaojun, et al. "Read performance: The newest barrier in scaled STT-RAM." *IEEE Transactions on Very Large Scale Integration (VLSI)* *Systems* 23.6 (2015): 1170-1174.
25. Syu, Shun-Ming, Yu-Hui Shao, and Ing-Chao Lin. "High-endurance hybrid cache design in CMP architecture with cache partitioning and access-aware policy." *Proceedings of the 23rd ACM international conference on Great lakes symposium on VLSI*, 2013.

.

1. <CPU Cache comparistion table>

| ***Parameters for the below techniques (Year)*** | Processor | |  | | | | | | | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Level 1 (L1) for Instruction (I) or Data (D) | | | | | Level 2 (L2) | | | | | Level 3 (L3) or Last Level Cache (LLC) | | | | |
| # of cores | Freq. | Capacity | Set Assoc. | Device  Tech. | # of CL | Protocol | Capacity | Set Assoc. | Device  Tech. | # of CL | Protocol | Capacity | Set Assoc. | Device  Tech. | # of CL | Protocol |
| Khoshavi [2] (2016) | 8 | 3GHz | 32KB | 8-way | SRAM | 512 | MESI | 512KB | 8-way | SRAM | 8192 | MESI | 96MB | 16-way | eDRAM | ~1.5M | WB |
| Sun [8] (2011) | 4 | 2GHz | 32KB | 4-way | SRAM | 512 | N/A | 256KB | 8-way | SRAM | 4096 | N/A | 4MB | 16-way | STT-RAM | 65536 | N/A |
| C. Bienia [6] (2016) | 4 | 2GHz | 32KB | 4-way | SRAM | 512 | N/A | 1MB | 16-way | STT-RAM | 16384 | N/A | N/A | N/A | N/A | N/A | N/A |
| Chen [3] (2016) | 4 | 3.3 GHz | 32KB | 8-way | SRAM | 512 | WB | 4MB | 8-way | STT-RAM | 65536 | WB | N/A | N/A | N/A | N/A | N/A |
| Motlagh [4] (2000) | 8 | N/A | N/A | N/A | SRAM | N/A | N/A | 512KB-1MB | N/A | SRAM/STT-RAM | 8192-16384 | N/A | N/A | N/A | eDRAM | N/A | N/A |
| Lin [5] (2015) | 2 | 800 MHz | 32KB | N/A | N/A | 512 | MOESI | 512KB | N/A | N/A | 8192 | MOESI | N/A | N/A | N/A | N/A | N/A |
| Jaleel [7] (2006) | 8 | N/A | 32KB | 4-way | N/A | 512 | N/A | 256KB | 8-way | N/A | 4096 | N/A | 4MB-64MB | 16-way | DRAM | 65536-  ~1M | N/A |
| Chang [9] (2013) | 8 | 2GHz | 32KB | 8-way | SRAM | 512 | MESI | 256KB | 8-way | SRAM | 4096 | MESI | 32MB | 16-way | SRAM/DRAM/STT-RAM | 524288 | N/A |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

“CL”= Cache line

Calculation for “# of CL” columns:

Manually compute the number of cache lines given the capacity value as listed in capacity column, assuming the cache line size is always 64 Bytes

Protocol column = {Write Back (WB), Write Through (WT), MESI, MOESI, Not Available (N/A)}