10-28-2016

Voltage Droop Analysis and Mitigation in STTRAM-based Last Level Cache

Radha Krishna Aluru
University of South Florida, aluru@mail.usf.edu

Voltage Droop Analysis and Mitigation in STTRAM-based Last Level Cache

by

Radha Krishna Aluru

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Computer Science
Department of Computer Science and Engineering
College of Engineering
University of South Florida

Co-Major Professor: Swaroop Ghosh, Ph.D.
Co-Major Professor: Srinivas Katkoori, Ph.D.
Hao Zheng, Ph.D.

Date of Approval:
September 29, 2016

Keywords: Write, Bank, Sub-bank, Simulation

DEDICATION

This is to my parents, Venkata Rama Krishna Rao and Syamala Devi, to my dear sister Jaya Lakshmi Aluru and finally to the Computer Science & Engineering Department, University of South Florida. All of their belief in my capabilities and constant support helped to make everything of this possible.

ACKNOWLEDGMENTS

I would like to give thanks to my mentor and advisors, Dr. Swaroop Ghosh and Dr. Srinivas Katkoori, for handing me with the opportunity to be a contributing member of their research. Their continuous guidance and dedication has helped me excel in this field of study for the past year. I would like to heartfully thank my committee members Dr. Swaroop Ghosh, Dr. Srinivas Katkoori, and Dr. Hao Zheng. Dr. Ghosh helped me with my initial research that was published later on with his constant support and hours of guidance. Dr. Katkoori spent hours of time passing knowledge and experience suggesting corrections and modifications wherever and whenever needed through my work. Dr. Zheng spent a lot of time passing his wealth and knowledge and suggestions make better my work in the final stages. From computer science department, I would like to thank Gabriela Franco and Lashanda Lightbourne (Shanie) for helping all the time during my time in the CSE department. Their dedication to helping me was invaluable and greatly appreciated. I would like to thank my colleagues from the LOGICS lab Asmith De, Nasim Imtiaz Khan, Rekha Govindraj, Deepak Reddy Vontela, Hamid Motaman, and Nitin Rathi for a great experience and their valuable assistance during times. I especially would like to thank Nitin in helping with the basic start up support and Hamid in the initial phase. Asmith, Nasim, and Deepak were present all the time supporting and helping me. I would never forget the help from Rekha, especially on the day of my defense, it is highly appreciated. Last but not least, I thank my family and the department for all the opportunities they have provided to me.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Preferred especially for a Last Level Cache (LLC) due to its high retention and tolerance capabilities, Spin-Transfer Torque Random Access Memory (STTRAM) is an emerging and a promising Non-Volatile Memory (NVM) technology. To switch the magnetization of a Magnetic Tunnel Junction (MTJ), the amount of current needed is very high (~100µA per bit). For a full cache line (512-bit) write, this extremely high current results in a voltage droop in the conventional cache architecture. Due to this droop, the write operation fails especially when the farthest bank of the cache is accessed. In this thesis, we perform an analysis of the voltage droop across the STTRAM Last Level cache and then propose a new cache micro-architecture to mitigate the droop problem and make the write operations successful.

Instead of continuously writing the entire cache line (512-bit) in a single bank, the proposed architecture writes 64-bits in multiple physically separated locations across the cache. The voltage droop issue for crossbar memories such as Resistive RAM (ReRAM) has been pointed out but however, similar issue for STTRAM has never been investigated. In this study, we perform voltage droop analysis on the conventional STTRAM LLC while performing write/read operation with a simulation circuit model. Our investigation reveals that this problem exists for the write operation in a STTRAM LLC when we try to access the farthest bank in the cache. We propose a droop-mitigation Architecture which reduces the droop significantly. The effectiveness of this proposed architecture on the cache parameters such as latency and energy are compared with the conventional architecture for against various benchmarks. From the simulation results obtained (both circuit and micro-architectural), compared to the conventional architecture, the proposed architecture incurs 1.95% IPC and 5.21% energy for a 8MB last level cache.

CHAPTER 1

INTRODUCTION

Spin Transfer Torque Random Access Memory (STTRAM) [1] due to its features of non-volatility, low-power, low cost, and high-speed is considered as one of the best promising memory technology. The development of STTRAM emerged from the idea of the commercialization of Magnetic RAM (MRAM) to become the potential future memory technology. Hence, STTRAM can be called the energy-efficient modification of MRAM [2]. Here, the current induced spin-transfer torque is used for the switching of the magnetization. A low power and fast switching happens in STTRAM by the use of a current that is spin-polarized. Having the near density of DRAM, and a closer speed of SRAM, followed with a superb retention time and high level of endurance, a widely suitable candidate for the next universal memory is STTRAM [4-5].

1.1 Spin Transfer Torque Random Access Memory

An STTRAM cell is composed of a Magnetic Tunnel Junction (MTJ) which looks like a sandwich for storing the binary data. An MTJ is composed of two layers that are ferromagnetic. A tunnel/oxide barrier layer is present in-between separating both these layers. One of the two layers of ferromagnetic is called as Reference/Fixed layer and the other one is termed as Free layer. As the names suggest, the reference/fixed layer always has a fixed magnetic direction, and the free layer is a rotatable one so that, its magnetic direction can be in parallel or anti-parallel to the reference layer. The binary data (0,1) is represented using these two states.

Fig. 1.1 shows the cell schematic of an STTRAM. The, storage element is the Magnetic Tunnel Junction (MTJ) consisting of a pinned and a free magnetic layer.

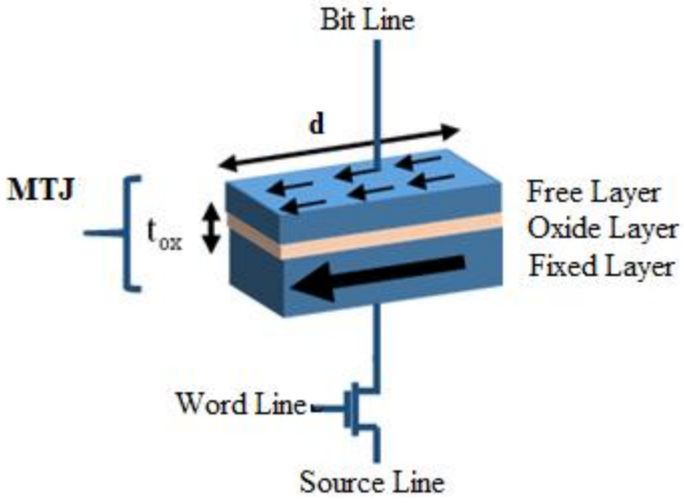

Figure 1.1 STT RAM bit cell showing an MTJ with different layers and the bitline, wordlines

When there is an anti-parallel magnetic orientation of the free layer compared with the fixed layer, the resistance of the MTJ is high and vice versa. Fig. 1.2 shows the two states of an MTJ parallel and anti-parallel. The active elements in magnetic random-access memory (MRAM) can be flipped using a spin induced Transfer torque [3]. A current that is induced from source-line to bitline and vice versa can be used in changing the MTJ configuration from parallel to anti-parallel and vice versa. The phenomenon of using a spin-torque for reversing of magnetization of the free layer results in the reduction in the write power when compared with the MRAM (conventional).

The magnetic orientation of the free layer stores the data in an MTJ. The restistance value of this orientation is used for the read cell operation. As in Fig 1.2, the low resistance of the MTJ happens in the parallel states (same direction) of the magnetic fields of the reference layer and the free layer. This is represented as a logical-0 in bitwise operations. When the anti-parallel states of these layers happen, the high MTJ resistance in the anti-parallel (opposite direction) is high and is represented in bitwise operations as a logical-1.

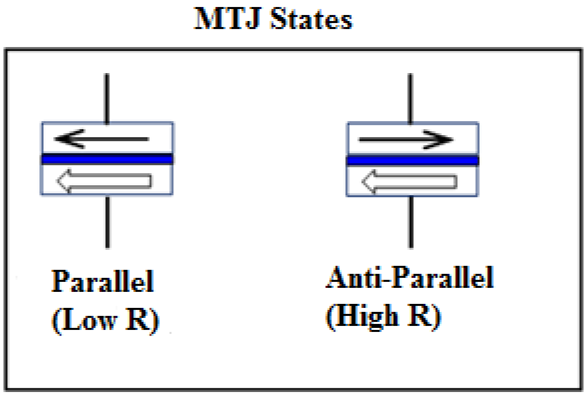

Figure 1.2 The two MTJ states parallel and anti-parallel

As discussed before, applying of a magnetic field or injection of an electrical current can be used to flip the ferro layer that is free. Injection of the current from a free layer to the reference layer, causes the in parallel switching of the layers resulting in the MTJ resistance to be low. This is represented as a 0-logic state. Similarly, the reverse direction induction of current will lead to anti parallel orientations of the layers leading to a high in resistance state for the MTJ. This is represented as a 1-logic state.

The design of an MTJ cell in a STT-RAM is so endurant. According to various studies, a thermal/temperature disturbance of at least 10 years is needed for the stored junction polarization upset in an MTJ [1]. Hence, non-volatility became the major advantage of STT-RAM, which means, indefinite data storage without a power supply. One more advantage of this non-volatility feature is that there is no need for the periodic refreshment of the data that is stored. This completely eliminates the concept of need for refresh power.

The various characteristics of an STTRAM include

- Highly scalable
- Non-volatility
- Low power consumption
- The read and write speeds on par with that of SRAM
- High levels of endurance
- Low footprint (4 F^2)
- Multi-level cell capability [7].

All of these characteristics make STTRAM one of the best alternatives to the existing SRAM to be adopted as the shared LLC by all the CPU cores in the current generation of growing multi-core processors.

1.2 STTRAM Read and Write Operations

As discussed in the previous section, based on the data that is stored, the MTJ resistance of the STTRAM changes. This results in a need for various sense and writing techniques for performing the reads and writes through the MTJ of the STTRAM. This is done by both Sense Amplifiers and the Write drivers. Their organization is as shown in Fig 1.3.

For reading the stored data in an MTJ cell, which is called by some as the operation of activation, between sense and bit lines, a small amount voltage is applied and, the amount of current flow is sensed. The read operations need a sense margin by a sufficient amount between the states of the resistance of the MTJ cells to that of the applied signal of reference. But some cases, the MTJ resistance variations will degrade this sense margin. This may result in the detection errors of the state of resistance causing Read errors.

On the other hand, write operation is a current-mode operation and not a voltage operation [4]. To flip the MTJ for changing its free layer magnetic orientation, a large amount of current is needed which is basically the write operation. The direction in which the current is applied decides, the parallel or anti-parallel states between the free and the reference layers to happen. The effort needed for the MTJ write operation is larger by a significant amount compared to that of reading of an MTJ. For this purpose, STT RAM uses large write drivers [4], one write driver for a single bit Global column. This makes write a slower and complicated operation compared to read in

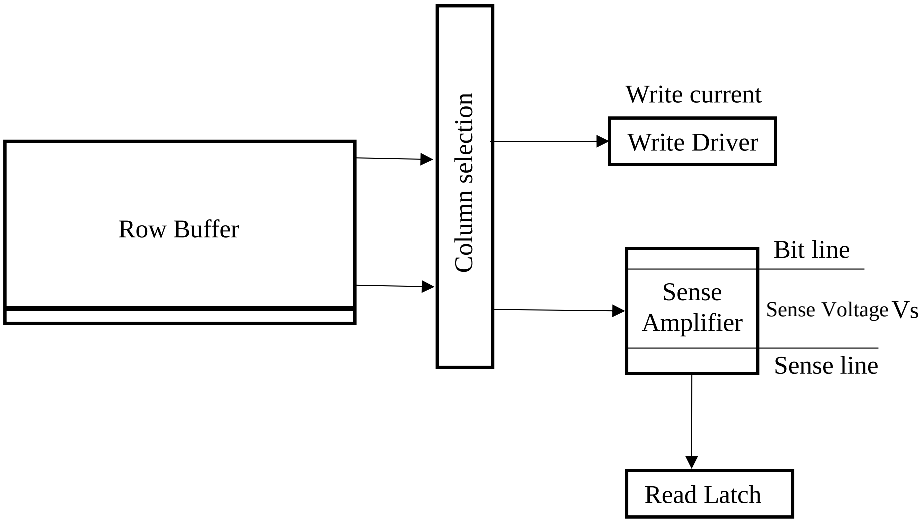

Figure 1.3 Organization of sense amplifier and a write driver

STTRAM. In addition, the various factors that affect the write speed of STT-RAM include fluctuations in the temperature and the variations of process within the die. This will delay the operation and a pulse longer than the normal needs to be reserved for ensuring the errors do not occur even if the operation is delayed by a cycle(s).

Spin-transfer torque lowers these write current requirements in STTRAM. However, the write current is still too high for most of the commercial applications [3]. In a conventional cache architecture, the high write current (~100µA per bit) might lead in a write failure due to the voltage droop that happens during the write operation. Fig. 1.4 shows the write operation in a conventional LLC of size 8MB with 8 banks (Bank 1 – 8) each of size 1MB. The entire cache line (512-bits) is written in one of the banks based on the address. The best case for droop is when we write to the bank close to the voltage source or regulator. Due to less interconnect resistance, the droop magnitude will be negligible resulting in correct write operation. However, the write to the farthest bank will experience considerable amount of droop (~200mV as per our estimate) which would

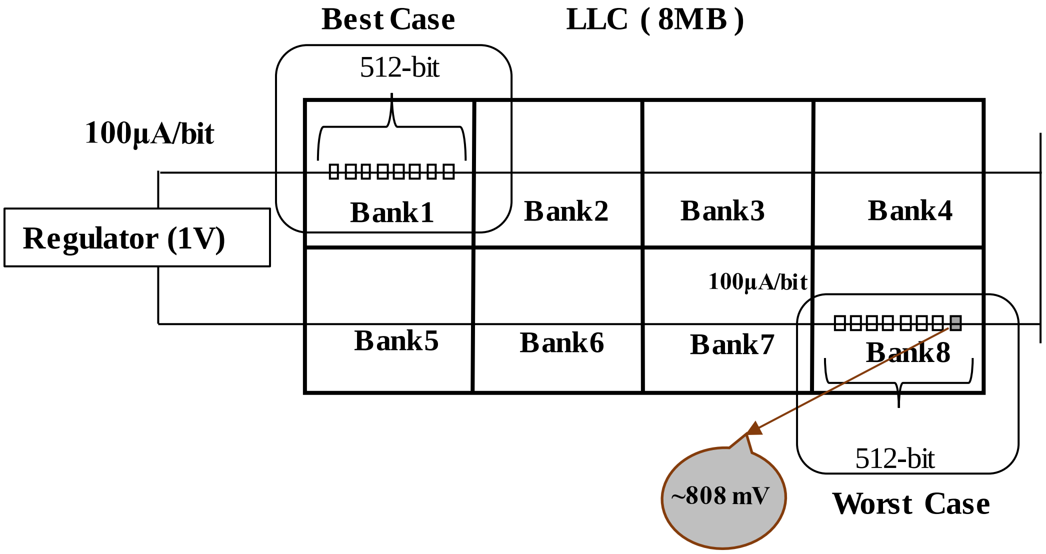

Figure 1.4 Conventional STT RAM LLC write operation showing the voltage droop/drop in worst case

result in a failure since the STTRAM will fail to flip for writing 0's and 1's. Even within the cache line the word which is farthest from the voltage source suffers the most. The primary reason for this failure is the high STTRAM switching current per bit and the need to write large number of bits simultaneously.

To overcome the above challenge, we propose a new LLC architecture which is shown in Fig. 1.5. Instead of writing the entire 512-bits in a single bank which draws significant current (~512x100µA) creating a large voltage droop for the last sequence of bits, we split the entire cache line into 8 parts (64*8) and write them in multiple physically separated locations across the cache. The approach reduces the current drawn per bank from 512xI_{write} to 64xI_{write} thereby reducing the droop and succeeding the write operation. In the proposed approach, the worst case bits only experience a maximum of 10% droop therefore the write operations succeed without any errors.

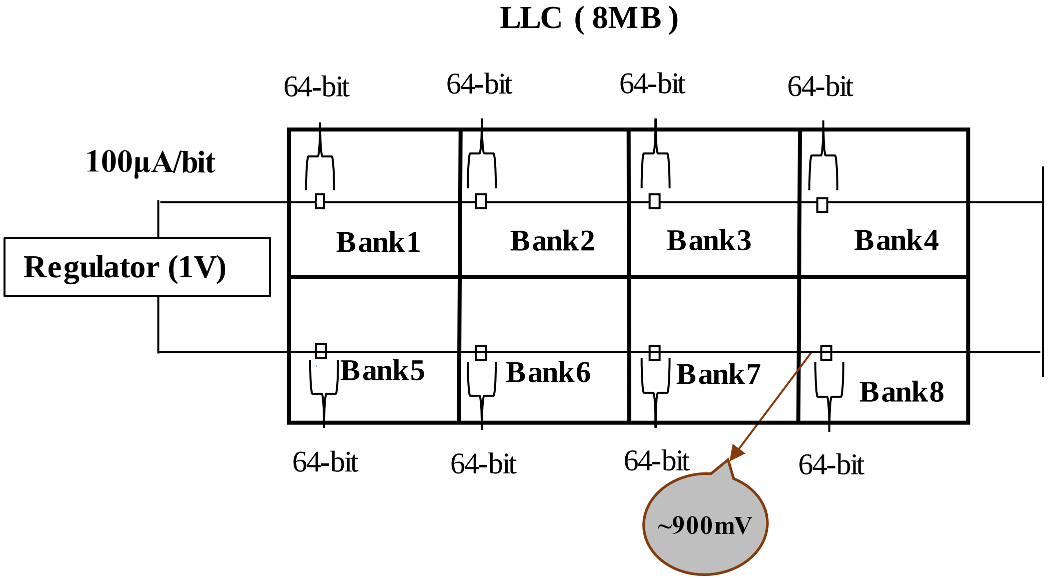

Figure 1.5 Proposed droop mitigating architecture showing the droop/drop to the last 64-bit of the cache line.

Note that, we still write to a single bank logically, however, the bank is spread across the cache physically to mitigate the droop.

1.3 Related Work

The voltage droop for crossbar memories such as Resistive RAM (ReRAM) has been pointed out [7][11]. The crossbar technology helps in achieving high density. However, crossbar faces some serious challenges especially when it comes to voltage drop and the sneak currents. As a result of the IR drop caused by the wire resistance and the sneaking currents, there will be different voltage drops across the cells for several cells within a mat of the Resistive RAM. A considerable loss of voltage occurs on the metal wires due to current passing through them and thus the voltage drop to the furthest ReRAM crossbar cell is decreased. The effective cell across voltage has a considerable amount of influence from the crossbar architecture's data pattern. The write latency on an overall is impacted by all these factors in a ReRAM [11].

However, similar issue for STTRAM has never been investigated. To the best of our knowledge, for the first instance, in thesis, we identify the voltage droop challenge for write operation in STTRAM and propose a novel micro-architectural solution for mitigation.

1.4 Contributions

The following are the main contributions made in this thesis:

- We perform voltage droop analysis on the conventional STTRAM LLC while performing write/read operation with a simulation circuit model.
- We propose a droop mitigation architecture which reduces the droop significantly.

- The impact of proposed architecture on the cache parameters namely latency and energy are compared with that of the conventional architecture for various benchmarks.

The rest of the thesis is organized as follows. In Chapter 2, we perform detailed voltage droop analysis on conventional LLC. In Chapter 3, we discuss the proposed droop mitigating architecture and its circuit analysis. In Chapter 4, we present the simulation set up and the simulation results. Finally, in Chapter 5, we draw the conclusions.

CHAPTER 2

VOLTAGE DROOP ANALYSIS IN A STTRAM LLC

In the following section, we describe the model of STTRAM Last Level Cache. Next, we describe the existing LLC cache bank architecture and droop analysis using the LLC model.

2.1 Last Level Cache

There is a hierarchy of cache levels in a typical processor. At the top of the hierarchy level, we have the caches that are Level 1 (L1) caches. These are small in size and the fastest in the operation. The size of the Level1 cache is typically 64 KB (32 KB instructions and 32KB remaining for the data). At the next level, the Level 2 or L2 cache have a size close to that of of 256 KB. Each of the CPU cores have a dedicated Level 1 and Level 2 caches for the case of a multiprocessor systems. The last or the third level of caches is the last-level cache (LLC) or the L3. Unlike the higher-level caches, which are private, this cache is shared. The size of the LLC is much larger than L1 & L2 and varies from 3 MB to over 32 MB, depending on the processor models [6].

There are two main reasons for this variation in the sizes. First is the different processor model associativities and second is the number slices/Banks used to build the cache forming the cache sets. The variations in the latency of an LLC are caused by different cache slice access times within the L3. These are classified into UCA (Unifrom Cache Access) and NUCA (Non-uniform Cache Access) models.

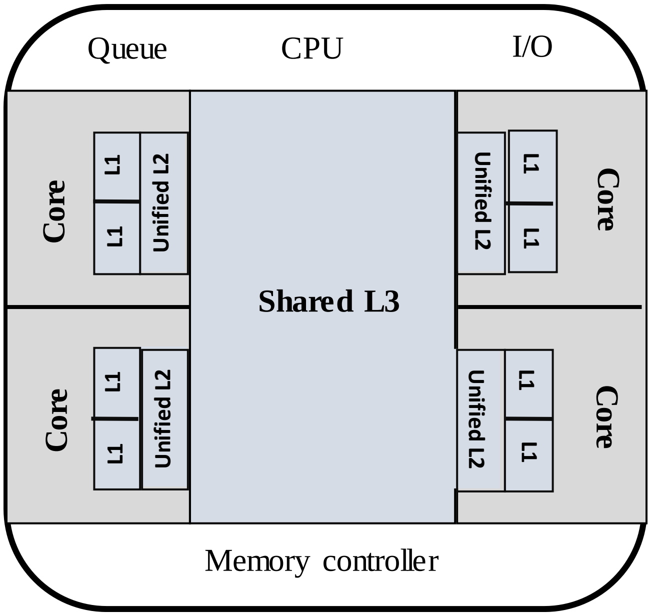

Figure 2.1 A multiprocessor CPU model with a shared LLC

The fixed-size units of memory stored by the cache are called Cache blocks/lines which typically are of the size of 64 Bytes (512 bits). Each of these lines map to a set in the cache. The associativity is defined by the number of these lines stored in each of the sets in a concurrent manner. Having an associativity value of *n* means, the cache is an *n*-way set-associative cache [5].

Cache read latency is considered as the amount of time CPU cycles required to read the cache line from one of the cache slices where its address is located.

Same goes with the write latency but this time we will be updating the Cache line information in that particular address belonging to a single Cache slice.

2.2 Model of STTRAM LLC

The entire cache is divided into multiple banks (can also be considered as the number of cache slices joined together to form the LLC). The entire cache line read/write is performed within this single bank [9]. Each bank is divided into a group of mats. A mat includes multiple ways (way *0* to *n*). The output cache-line is provided together by various mats in groups (e.g., 8 mats provide 64-bit each totaling to the cache line of 512 bits) [10]. Each mat contains a group of subarrays and a common pre-decoder is shared by them to provide the requested data or perform write operation.

For this study, we have considered an 8-MB LLC with 8-way set associativity. Each subarray refers to the associativity's (ways) from 0-7 to select from. Each of these ways consists of the rows and columns (global columns are muxed with local columns) to store the individual bits. The subarray size is 16KB. Each mat is composed of 8 subarrays (SA[7:0]) amounting to a total size of 128KB each. Each bank is composed of 8 mats (mat[7:0]) of total size 1MB. There are 8 such independent banks in the cache. The cache organization is shown in detail in Fig. 2.2.

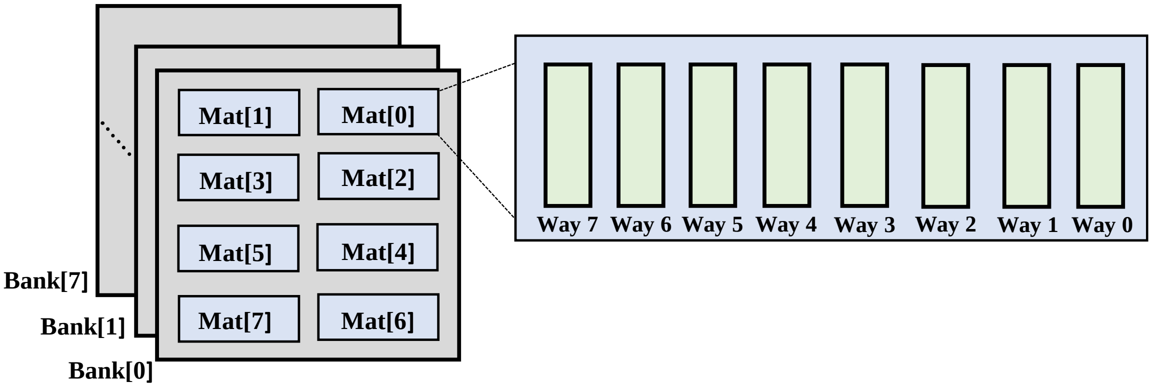

Figure 2.2 An 8MB LLC organization with banks, mats and ways/ subarrays

Fig. 2.3 shows a proposed subarray design consisting a total of 512 WLs and 512 local columns with 64 (32*2) global columns. The row decoders and column decoders are used to map the Global columns with the Local columns.

Each Global column provides a single bit that constitutes to the over all 512-bit cacheline.

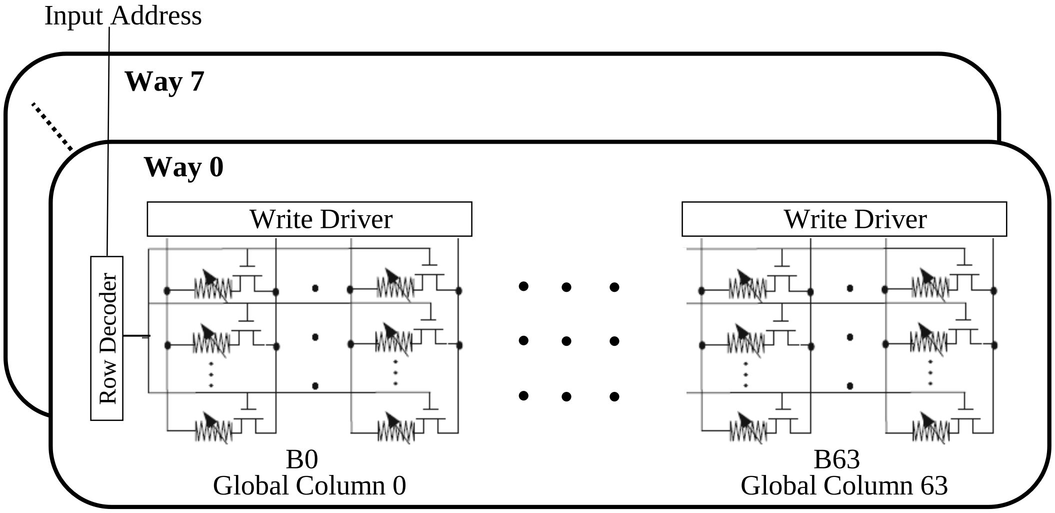

Figure 2.3 Subarray/way architecture of STTRAM showing the STTRAMs, the organization of bits, the global columns and the write drivers for a write operation

2.3 Circuit Simulation with Droop

We have used circuit model of an 8MB cache and simulated the effective model of a cache line (512-bit) write. We write 64-bit in each of the 8-subarrays (belonging to way 7) in each of the 8 mats within a single independent bank. The values of the effective resistance and capacitance of

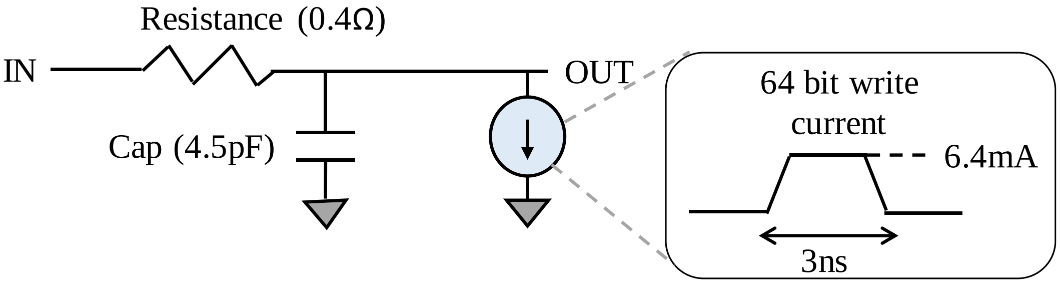

Figure 2.4 STTRAM 1-bit write operation with effective resistance 0.4 Ω in parallel to a 4.5pF capacitance and a pulse current load of 0.1*64mA

the power supply which provides the current needed to write the 512 bits (64 bits across 8 mats) are shown in Table 2.1.

Table 2.1 Equivalent calculated values for the LLC cache model

# of bits	Resistance	Capacitance	Write current (Load)
1	0.4 Ω	4.5 pF	100 μA
64	25.6 Ω	0.29nF	6.4 mA

Fig. 2.4 shows the circuit model of the write operation with a calculated effective values. Fig. 2.5 shows a circuit model of the cache with 8 banks to perform the write operation in the 8 mats of the last and the farthest bank. The idle banks are represented by shaded blocks. The unshaded bank represents an equivalent cache line being written into it. The supply voltage is assumed to be 1V.

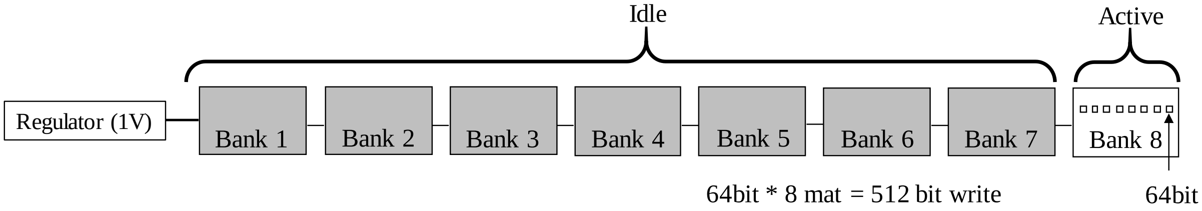

Figure 2.5 STTRAM cache circuit model with write in the last/farthest bank

Fig. 2.6 shows the corresponding voltages at each of these mats to perform the write operation from mat1 to mat8. As we can see from the plot that the voltage keeps on drooping down and the last 64-bit of the cache line at mat8) receives ~0.79V to perform the write operation. At such low voltage, the STTRAM fails to switch states resulting in a write failure. This is due to high write current of the STTRAM and inability of the conventional LLC bank architecture to provide reliable supply current.

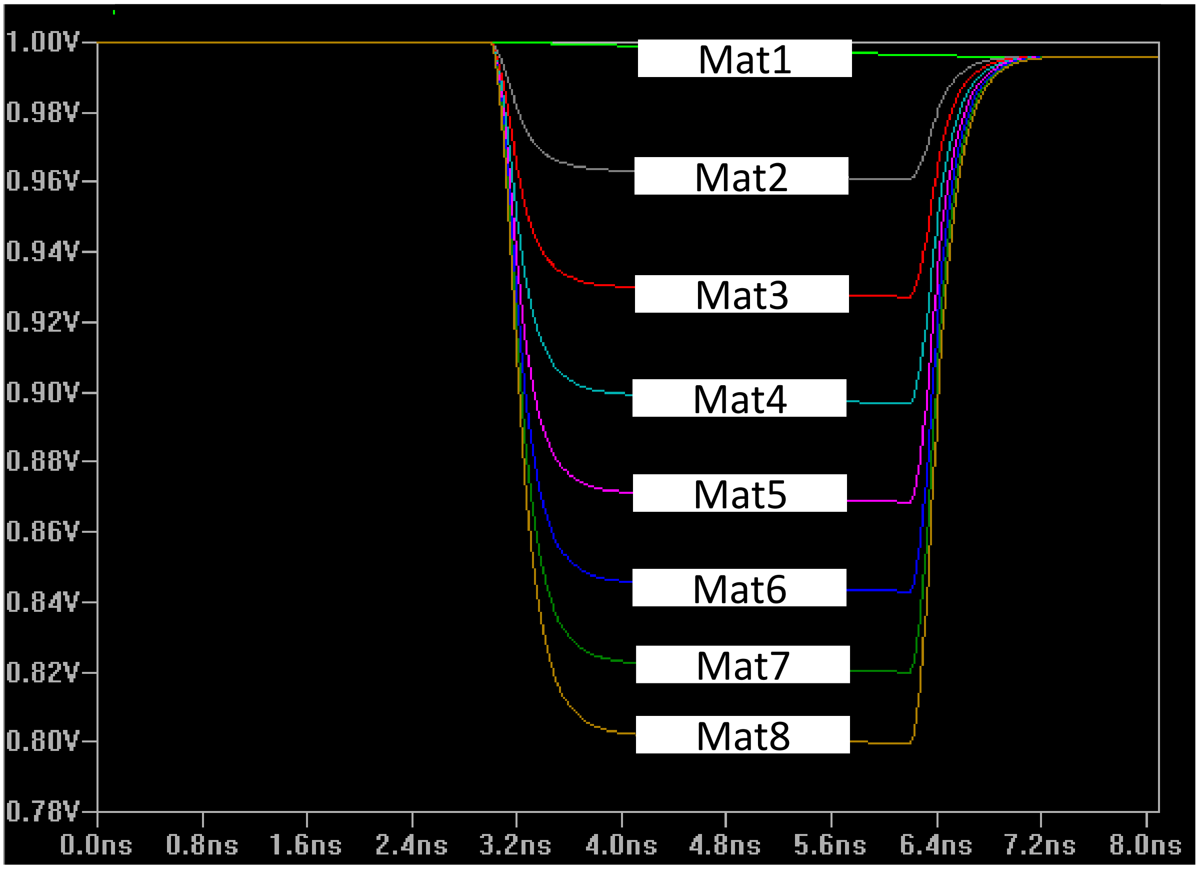

Figure 2.6 Voltage plot showing the available voltage for each of the 64-bit write operations in each mat

We have used Hspice [12] to simulate the STTRAM flipping phenomenon for the write-1 and write-0 at various supply voltages. The STTRAM model consists of a MTJ verilogA with adjustable parameters such as temperature, spin-polarization, conductance etc. We implemented this model in a 22nm technology and performed the simulation of write operation (both bit-0 and bit-1) at varying supply voltages. Figs. 2.7 and 2.8 show the plots of write latency with supply voltage. From the plots, the STTRAM write latency is increased as the supply voltage is reduced from 1V. For write-1, the latency increases by ~150ps at 0.9V. At further scaled down voltages, this delay rises exponentially. For write-0, less than ~0.9V will result in failure. At 0.9V, the latency increases by ~160ps.

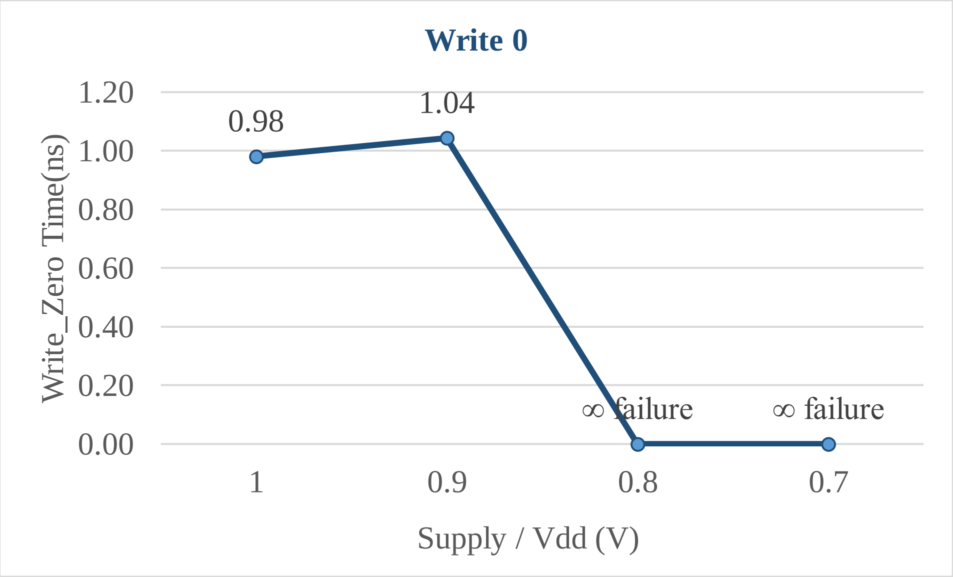

Figure 2.7 Plot showing the result obtained from Hspice simulations of the STTRAM write time for logical 0

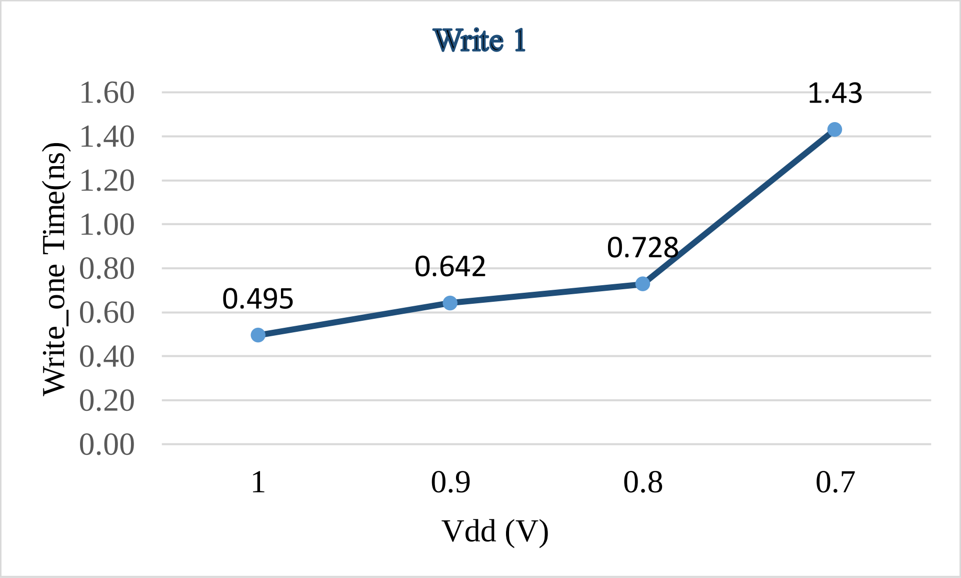

Figure 2.8 Plot showing the result obtained from Hspice simulations of the STTRAM write time for logical 1

CHAPTER 3

DROOP MITIGATING BANK ARCHITECTURE

In this Chapter, we present the proposed LLC architecture for overcoming the droop for a successful write operation. We also provide a detailed circuit analysis of the new proposed model.

3.1 Architecture Model

Since the conventional architecture results in voltage droops we propose a new bank architecture which distributes the current drawn during write operations at different physical

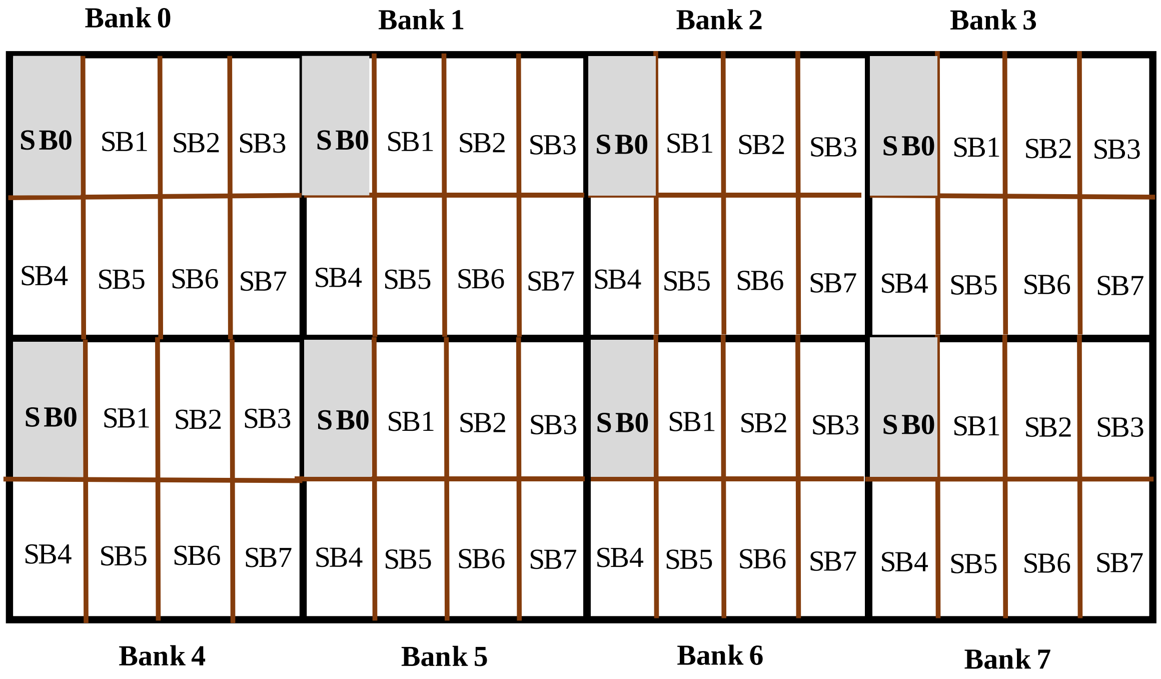
Bank 0
Bank 1
Bank 2
Bank 3
SB0 SB1 SB2 SB3 SB0 SB1 SB2 SB3 SB0 SB1 SB2 SB3 SB0 SB1 SB2 SB3
SB4 SB5 SB6 SB7 SB4 SB5 SB6 SB7 SB4 SB5 SB6 SB7 SB4 SB5 SB6 SB7
SB0 SB1 SB2 SB3 SB0 SB1 SB2 SB3 SB0 SB1 SB2 SB3 SB0 SB1 SB2 SB3
SB4 SB5 SB6 SB7 SB4 SB5 SB6 SB7 SB4 SB5 SB6 SB7 SB4 SB5 SB6 SB7
Bank 4
Bank 5
Bank 6
Bank 7

Figure 3.1 Droop mitigating physical bank architecture of 8MB LLC

locations. Therefore, the effective droop at a particular location is reduced. Fig. 3.1 shows the physical bank architecture for the droop mitigating LLC. Each of these physical banks divided into 8 sub-banks (SB0–SB7) represented in shaded color. Similar color sub banks located at various different physical locations throughout the cache together represent a complete logical bank i.e., though separated physically, each of these sub-banks are still logically continuous in terms of physical addresses. For example, consider the sub-bank SB0 (grey) which is distributed in 8 different locations forming a continuous logical bank, bank0.

Fig. 3.2 shows the inside look of a sub-bank which only contains a single mat unlike 8 mats before.

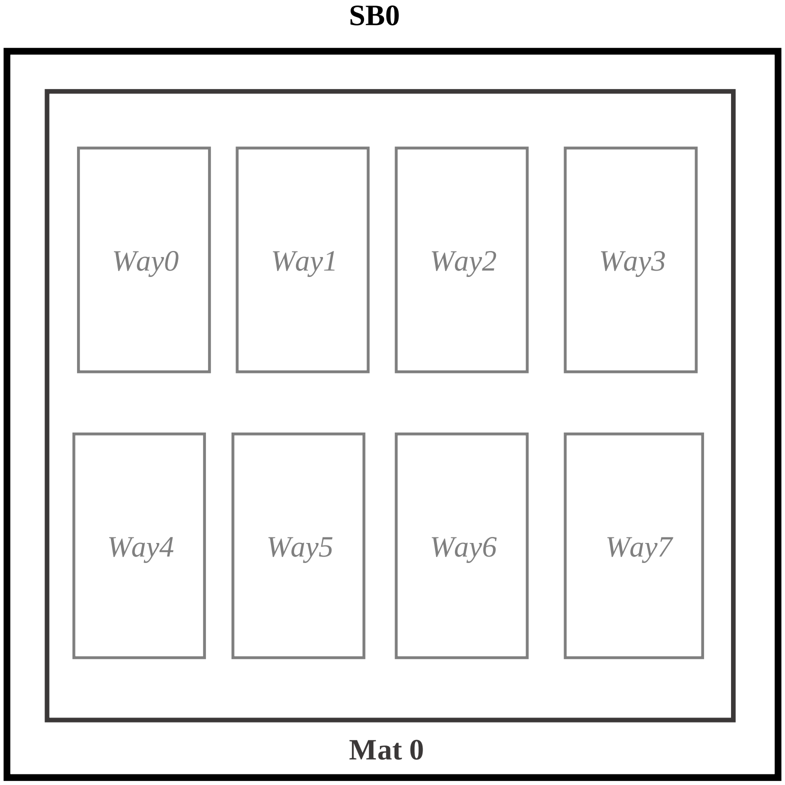

Figure 3.2 Inside look of the sub-bank(SB0) which now contains only a single mat (mat0) and the 8 subarrays/ways inside it.

The internal structure of this mat with the subarrays/ways remains the same. Likewise, each of these 8 sub-banks now contribute 64-bit each forming the entire cache 512-bit cache line. Each of the 64-bit write occurs at different physically separated locations hence mitigating the droop and allowing the write operation to succeed.

3.2 Circuit Analysis of the Proposed Model

We have created a circuit model of the proposed 8MB LLC as shown in Fig. 3.3. The shaded portions in each of the banks represent an equivalent mat/sub-bank with the 64-bit being written in them. The remaining unshaded portions in each of the banks are the idle sub-banks. The supply voltage is kept at 1V.

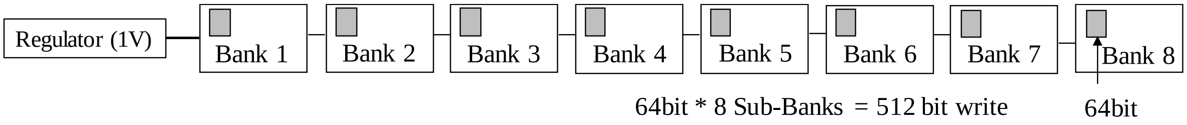

Figure 3.3 The cache circuit model with write operation in the droop mitigating architecture

Fig 3.4 shows the values of the corresponding voltages received at each of these. sub-banks to perform the write operation from bank1-bank8. We can observe that the voltage droop is reduced greatly and last 64-bit of the cache line in bank8 receives close to 0.9V to perform the write operation. Even though the write latency increases, the failure could be avoided.

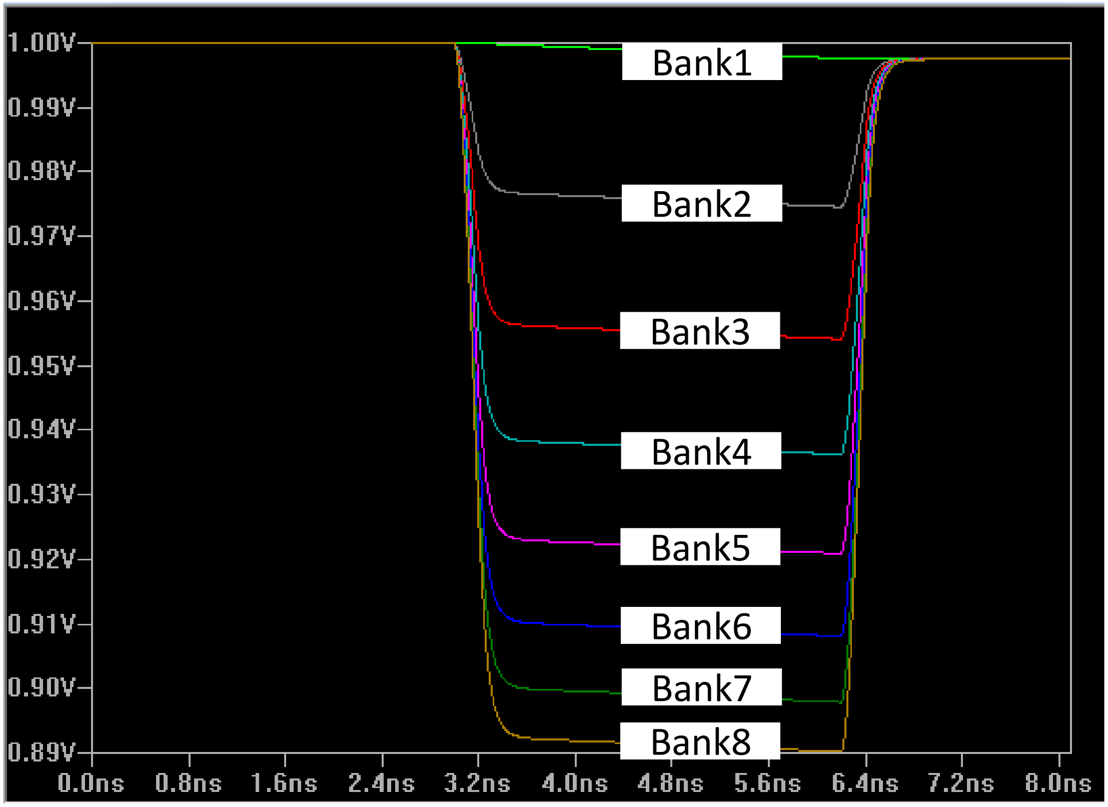

Figure 3.4 Voltage plot showing the available voltage for each of the 64-bit write operations in each of the banks (i.e. sub-banks)

CHAPTER 4

MICRO ARCHITECTURE EVALUATION AND RESULTS

In this Chapter, we will describe the simulation setup and present the micro-architectural simulation results.

4.1 Simulation Setup

We have used Cacti [13] [14] to simulate a model of the LLC with our proposed droop mitigating bank architecture. A footprint of size 4F$^{\wedge}$2 was used to model the STTRAM [14]. We have calculated the values of read/write latency and the dynamic read/write energy per access to this cache model in a system frequency of 1GHz using a 8MB shared LLC. The processor configuration details are provided in Table 4.1.

Table 4.1 Processor configuration

Processor	Alpha, O3, 4 cores, 2GHZ, Detailed CPU
SRAM L1 Cache	Private, I-cache=32KB, D-cache=64KB, 64B Cacheline, 2-cycle Read/Write latency and Write back.
SRAM L2 Cache	Private, Size=2MB, 64B Cacheline, 8 cycle Read/Write latency and Write back.
STTRAM LLC/L3 Cache	Shared, Size=8MB, 8 banks, 8ways, 64B Cacheline, Write back, Read/Write latency based on the Architecture Model
Main Memory	4GB, DDR3, 200-cycle latency

Table 4.2 details the read and write latency values that were considered for the conventional STTRAM LLC and calculated for our proposed droop mitigating LLC.

Table 4.2 The LLC latency values of both the conventional and the proposed architecture

	Latency(Cycles)	
	Read	Write
Conventional STTRAM LLC	6 cycles	9 cycles
Droop mitigating STTRAM LLC	10 cycles	13 cycles

In terms of latency, the inter-latency between the mat and new bank/sub-bank becomes 0, since each sub-bank contains only a single mat in the proposed architecture. The inter-mat latency i.e., the latency between subarrays and the mat remains the same as conventional architecture. The latency within a single subarray due to wordlines, bitlines, and the write drivers also remain the same. However, the delay/latency to access the bank is greatly increased as for every access, the farthest cache slice is reached. Extra latency of horizontal and vertical hops is paid in terms of cycles to mitigate the droop and make the writes successful. Similar is true for access energy in toggling these hops.

For the conventional STTRAM, we used the Non-uniform Cache Access (NUCA) model offered by Cacti to calculate the results. It considers the average value of all the horizontal and vertical latencies to reach the banks/cache slices [14]. In proposed model, we use Uniform Cache Access (UCA) where we toggle every cache slice every access thus paying more in terms of hop latencies.

4.2 Simulation Results

We have used Gem5 [15] [16] to plug-in values of latency and energy obtained from Cacti. Figs. 4.1 and 4.2 show the normalized comparison of these obtained results with the conventional architecture. Using these values in modified Gem5, we ran various benchmarks from SPLASH suite for both the conventional LLC and the proposed LLC. The simulations are run in Full System mode against benchmarks like barnes, fft, fmm, ocean, raytrace, fmm, radiosity, volrend, and water-nsquared. We have used Mcpat [17] tool to plug-in these benchmark stats obtained from Gem5 and generate the values of dynamic and leakage energies of the LLC.

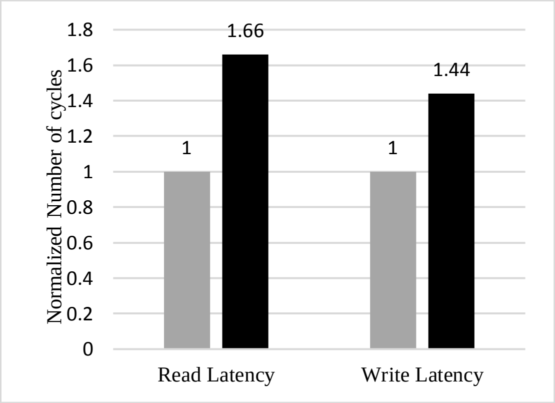

Figure 4.1 Normalized read/write latency comparison

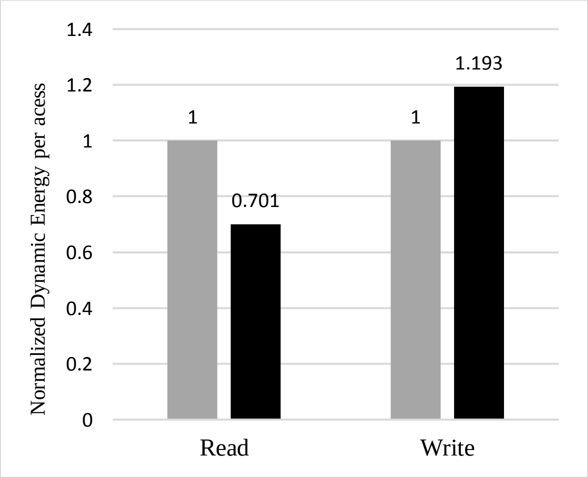

Figure 4.2 Normalized dynamic read/write energy per access comparison

Figs. 4.3, 4.4, and 4.5 show the comparison between droop mitigating architecture with conventional architecture with respect to IPC and energy. The proposed architecture is clearly results in minor (an average of 1.96%) overhead in terms of IPC and 5.21% energy overhead. The reason for this is the nature of the non-volatile memories, where the bit write/read time takes the dominant part of the latency while impact of the hop latencies in a cache is very small. The benefit of proposed architecture is observed from ~50% improvement in worst case droop (~100mV droop compared to ~200mV droop in conventional architecture).

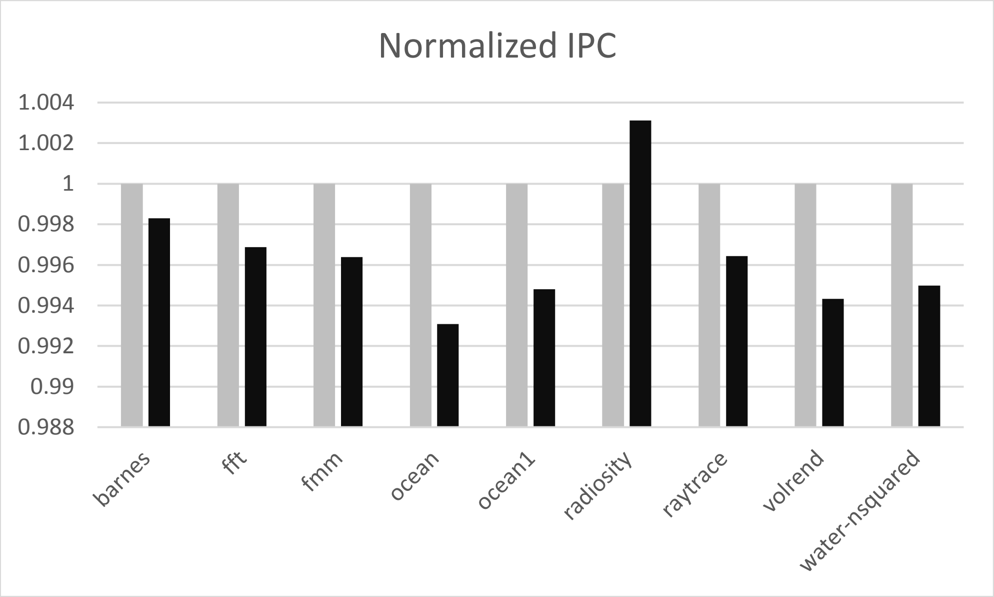

Figure 4.3 Normalized comparison of IPC

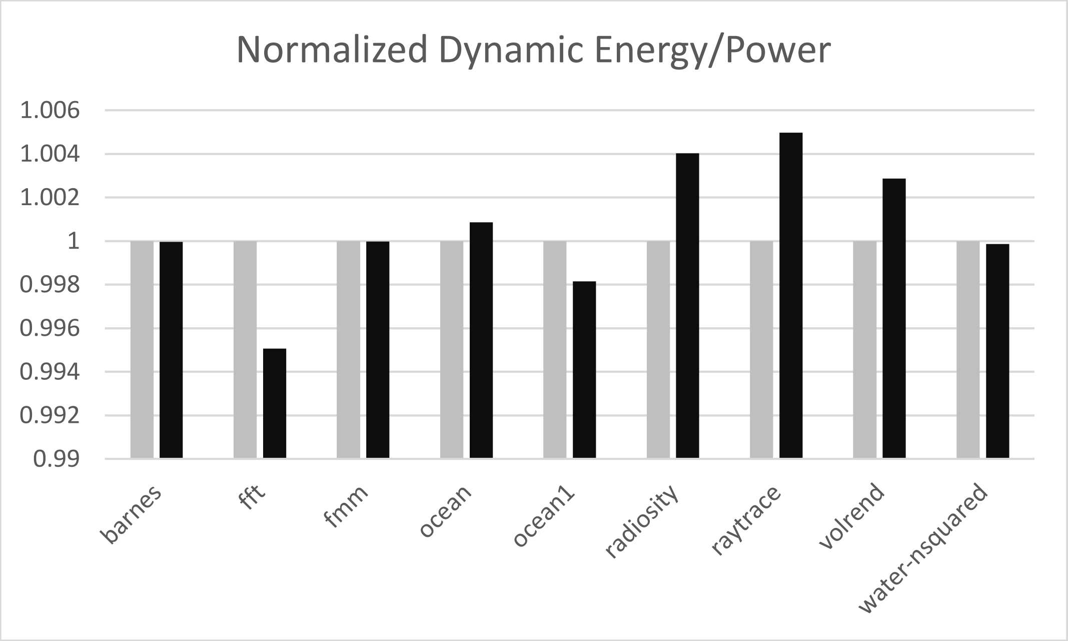

Figure 4.4 Normalized comparison of dynamic energy of various benchmarks

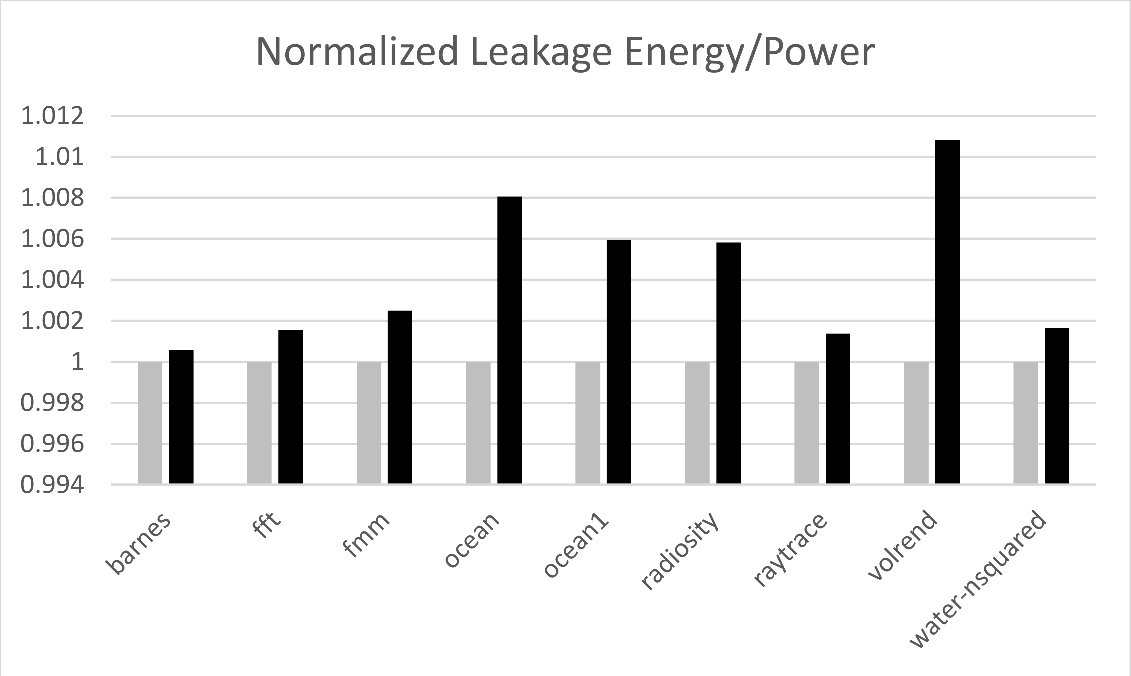

Figure 4.5 Normalized comparison of leakage energy of various benchmarks

CHAPTER 5

CONCLUSIONS

Spin-Transfer Torque Random Access Memory (STTRAM) is a dense memory with desirable features such as non-volatility, scalability, low power consumption, and unlimited endurance. These features make it especially preferred for LLC to cope up with the increasing processor speeds and on-chip memory demands due to technology scaling. However, the amount of current required for the reorientation of the magnetization for the write operation at present is very high. When a full cache line write is performed, this high current causes voltage droop resulting in write operation to fail especially for farthest bank/cache. We have proposed a new droop mitigating bank architecture of LLC to mitigate the droop and enable successful write operation. Instead of continuously writing in a single bank, we write the cache line in multiple different locations across the cache slices. Circuit and micro-architectural simulation results show that the proposed approach reduces voltage droop with negligible overhead in latency and energy.

REFERENCES

[1] Driskill-Smith, Alexander. "Latest Advances and Future Prospects of STT-RAM." *Non-Volatile Memories Workshop,* 2010.

[2] Zhu, Jian-Gang. "Magnetoresistive Random Access Memory: The Path to Competitiveness and Scalability." *Proceedings of the IEEE Proc. IEEE* 96, no. 11 (2008): 1786-798.

[3] An Chen, James Hutchby, Victor Zhirnov, George Bourianoff, John Wiley & Sons “Emerging Nanoelectronic Devices.”, *Nov 12, 2014*.

[4] Kultursay, Emre, Mahmut Kandemir, Anand Sivasubramaniam, and Onur Mutlu. "Evaluating STT-RAM as an Energy-efficient Main Memory Alternative." *2013 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, 2013: 256-267.

[5] Chen, E., D. Apalkov, Z. Diao, A. Driskill-Smith, D. Druist, D. Lottis, V. Nikitin, X. Tang, S. Watts, S. Wang, S. A. Wolf, A. W. Ghosh, J. W. Lu, S. J. Poon, M. Stan, W. H. Butler, S. Gupta, C. K. A. Mewes, Tim Mewes, and P. B. Visscher. "Advances and Future Prospects of Spin-Transfer Torque Random Access Memory." *IEEE Trans. Magn. IEEE Transactions on Magnetics 46*, no. 6 (2010): 1873-878.

[6] Yarom, Y., Qian Ge, Fangfei Liu, Ruby B. Lee and Gernot Heiser. "Mapping the Intel Last-Level Cache." *NICTA, ICT Research*.

[7] Niu, Dimin, Cong Xu, Naveen Muralimanohar, Norman P. Jouppi, and Yuan Xie. "Design Trade-offs for High Density Cross-point Resistive Memory." *Proceedings of the 2012 ACM/IEEE International Symposium on Low Power Electronics and Design – ISLPED*, 2012: 209-214.

[8] Tabrizi, Farhad. "Non-volatile STT-RAM: A True Universal Memory." *Flash Memory Summit, Santa Carla, CA, USA August* 2009.

[9] Muralimanohar, Naveen and Rajeev Balasubramonian. "CACTI 6.0: A Tool to Understand Large Caches." *Tech. Rep. HP Labaratories*, 2009/4/21.

[10] Motaman, Seyedhamidreza and Swaroop Ghosh. "Adaptive Write and Shift Current Modulation for Process Variation Tolerance in Domain Wall Caches." *IEEE Transactions on Very Large Scale Integration (VLSI) Systems IEEE Trans. VLSI Syst.* 24, no. 3 (2016): 944-53.

[11] Xu, Cong, Dimin Niu, Naveen Muralimanohar, Rajeev Balasubramonian, Tao Zhang, Shimeng Yu, and Yuan Xie. "Overcoming the Challenges of Crossbar Resistive Memory Architectures." *2015 IEEE 21st International Symposium on High Performance Computer Architecture (HPCA)*, 2015: *476-488*.

[12] "Hspice Simulator Tool. 2016 Synopsis, Inc." https://www.synopsys.com/tools/Verification/AMSVerification/CircuitSimulation/HSPICE

[13] Sheng Li, Ke Chen, Jung Ho Ahn, Jay B. and Norman P. Jouppi. “CACTI-P: architecture-level modeling for SRAM-based structures with advanced leakage reduction techniques.” *ICCAD '11 Proceedings of the International Conference on Computer-Aided Design, Pages 694-701.*

[14] Muralimanohar, N., R. Balasubramonian, and N. Jouppi. "Optimizing NUCA Organizations and Wiring Alternatives for Large Caches with CACTI 6.0." *40th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO 2007)*, 2007: 3-14.

[15] "Sims, M5." http://www.m5sim.org/Documentation

[16] Nathan Binkert, Bradford Beckmann, Gabriel Black, Steven K. Reinhardt, Ali Saidi, Arkaprava Basu, Joel Hestness, Derek R. Hower, Tushar Krishna, Somayeh Sardashti, Rathijit Sen, Korey Sewell, Muhammad Shoaib, Nilay Vaish, Mark D. Hill and David A. Wood. “The gem5 simulator.” *ACM SIGARCH Computer Architecture News archive, Volume 39 Issue 2, May 2011: 1-7.*

[17] Sheng Li, Jung Ho Ahn, Richard D. Strong, Jay B., Dean M. and Norman P. Jouppi. “McPAT: an integrated power, area, and timing modeling framework for multicore and manycore architectures.” *MICRO 42 Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture, New York, 2009: 469-480.*