Performance Study of Design Optimizations on STTMRAM L1 Cache

PROJECT REPORT

Submitted by

NARESH RAMAVATH

in partial fulfillment of the requirements

for the award of the degree of

MASTER OF TECHNOLOGY

Department Of Electrical Engineering

INDIAN INSTITUTE OF TECHNOLOGY MADRAS

JUNE 2021

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY MADRAS

2021

CERTIFICATE

This is to certify that this thesis (or project report) entitled ***“Performance Study of Design Optimizations on STTMRAM L1 Cache ”*** submitted by **NARESH RAMAVATH** to the Indian Institute of Technology Madras, for the award of the degree of **Masters of Technology** is a bona fide record of the research work done by him under my supervision. The contents of this thesis (or project report), in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma..

Dr. T. G. Venkatesh
Research Guide
Associate Professor
Department of Electrical Engineering
IIT Madras 600036

Acknowledgment

First and foremost, I would like to express my deepest gratitude to my guide, **Dr. T G Venkatesh**, Associate Professor, Department of Electrical Engineering, IIT Madras, for providing me an opportunity to work under him. I would like to express my deepest appreciation for his patience, valuable feedbacks, suggestions and motivations.

I convey my sincere gratitude to Shubhang Pandey, MS Scholar, IIT Madras, for all his suggestions and support during the entire course of the project. Throughout the course of the project he offered immense help and provided valuable suggestions which helped me in completing this project.

I would like to extend my appreciation to all my friends and for their help and support in completing my project successfully.

Abstract

Recently, the NVM Technology has received a lot attention as possible on-chip memories because of the very high density and low leakage powers. Researchers have investigated the PCM, ReRAM, Flash memory, STTMRAM, and many more to include these NVMs into the memory hierarchy and maximize their benefits. Even though PCM, ReRAM, and Flash Memory have shown promise, STTMRAM appears way too compatible, PCM faces problems in integration, and its very high write latency puts it at a disadvantage when the focus is on higher-level caches and both PCM and ReRAM are also plagued by severe endurance issues. Therefore, in this thesis, we focus on STTMRAM to be integrated along with the processor node technology. We consider it as a viable substitute to the SRAMs which are used. In the first part of our study, we look into the possibility of STTMRAM as a possible Main Memory solution. We perform our simulations on DRAMsim3 for synthetic workloads and observe the power values, after which we move closer to the processor, i.e., onto the chip, and explore STTMRAM as a viable L1 level cache. We use SST and NVsim Simulators against High-Performance Computing Workloads from the Rodinia Benchmarks Suite to assess this. We implement various cache design optimizations to find the best and worst optimization strategies in terms of power and performance when working against different replacement policies such as LRU, LFU, MRU, NMRU, and Random. We think such a rigorous study regarding STTMRAM as L1 cache under design optimizations has not been done, and it provides immense opportunity for future flexible caches and adaptive control policies to maximize the system's performance overall.

Contents

List of Figures

List of Tables

ABBREVIATIONS

PCM	Phase Change Memory
ReRAM	Resistive RAM
STTMRAM	Spin Transfer Torque Magnetic RAM
EDP	Energy Delay Product
Dyn Engy or DE	Dynamic Energy
FeRAM	Ferroelectric RAM
DDR	Double Data Rate

Chapter 1

Introduction

One of the significances of Moore's Law and performance enhancements is that processor speed in terms of instructions per second may be predicted to nearly double every two years. Memory capacity doubles every two years. However, memory latency, or how long it takes to perform an operation, has only improved approximately 1.1 times every two years. The processor's and main memory's speeds are increasing at an exponential rate. However, the difference between the diverging exponential grows exponentially. this problem is called the Memory wall problem. Our processors are getting far quicker than our memories, yet they still need to access memory every so many instruction since this disparity in speed has been rising extremely quickly over many years. We have been using caches as a sort of stairs for the memory wall. So our processors now are accessing fast caches, and only those rare accesses that are missing the cache will end up going to the slow main memory. major Obstacles in-memory System Design. The processor is much faster than the main memory.so memory speed cannot be increased beyond a certain point that is why we are coming up with so many techniques through which we can actually increase the speed of the memory. so the access time can be much much less.

Memory Hierarchy :A memory system is Organised into several levels by hierarchy which means it is divided into many levels.There are different levels of memory and the level that is closest to the processor is much faster, and which are a little further from the processor are slower.we incrementally add smaller, but faster memories each containing a subset of data stored in memory below it.Typical Hierarchy starting with closest to the processor, which is processor registers .then we have Level-1 Cache, typically divided into separate instruction count and data cache. we can have L2 Cache, L3 Cache, and main memory.

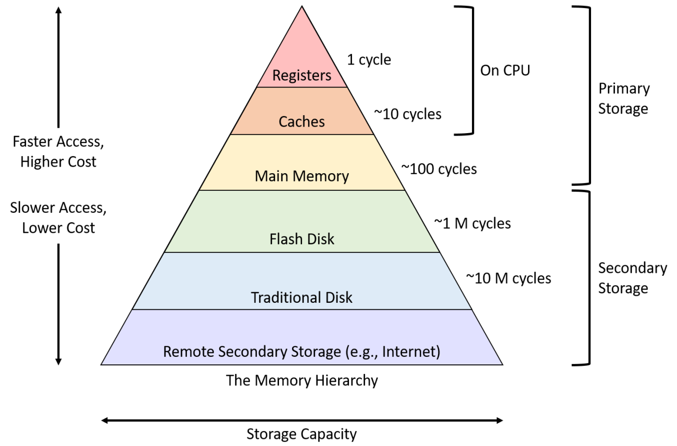


Figure 1.1: Complete Memory Hierarchy

As we move away from the processor the size increases, the cost slowly decreases, but at the same time, the speed also decreases.Cache memory works on the principle of Locality. Cache Memory Design as shown in the figure 1.2 is a high level block diagram which clearly indicates the major blocks in the design where we have the precharge block to hold the correct data in the cell, Address bits are managed and pass through the Row and Column Decoder to access the desired bit from a particular and number of blocks depend on the number of bits which can travel across the data bus at any given time. The timing and control block is responsible for decoding the address bits and timing the signals to units like Sense amplifer and Write Buffer and the Global Read and Write Block.

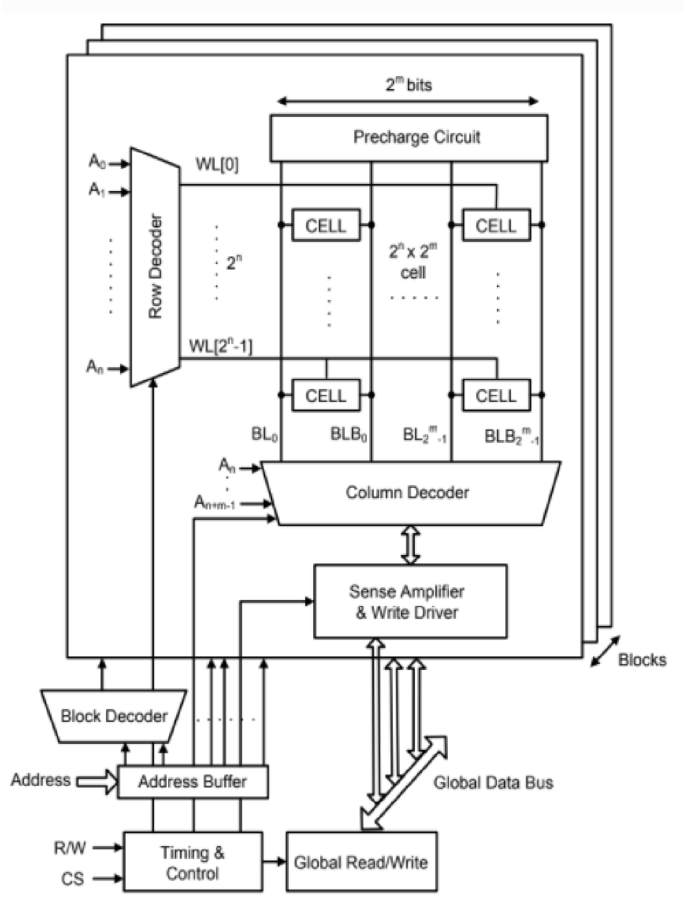


Figure 1.2: Cache Memory Design

STT-MRAM is one of the most promising nonvolatile memory technology with regards to its high density, low leakage power, less read latency compared to conventional 6T SRAM memory technology. STT-MRAM has few other characteristics like it is a non -volatile memory (retention time of cell to have the data is quite high), energy-efficient, and has very high endurance. Using the STT-MRAM memory technology in the case of hybrid memory(SRAM-STTMRAM) as L1 level Cache, to improve the performance of memory and can reduce the power consumption. when we have SRAM technology-based as an L1 cache, which is very fast, but in recent On-chip technologies, power consumption is one of the main reasons. As transistors scaling down significantly, with processor node technology, the leakage power kills our system .so this is the main drawback of SRAM-based memory technology. this is where STT-MARM will be integrated along with the SRAM or Vertical Integration of STT-MRAM into Memory Hierarchy in the on-chip cache memory since STT-MRAM is energy-efficient.

1.1 Motivation

STT-MRAM is one of the most promising nonvolatile memory technology with regards to its high density, low leakage power, less read latency compared to conventional 6T SRAM memory technology. STT-MRAM has few other characteristics like it is a non -volatile memory (retention time of cell to have the data is quite high), energy-efficient, and has very high endurance. Using the STT-MRAM memory technology in the case of hybrid memory(SRAM-STTMRAM) as L1 level Cache, to improve the performance of memory and can reduce the power consumption. when we have SRAM technology-based as an L1 cache, which is very fast, but in recent On-chip technologies, power consumption is one of the main reasons. As transistors scaling down significantly, with processor node technology, the leakage power kills our system .so this is the main drawback of SRAM-based memory technology. this is where STT-MARM will be integrated along with the SRAM or Vertical Inte-

gration of STT-MRAM into Memory Hierarchy in the on-chip cache memory since STT-MRAM is energy-efficient.

1.2 Aim

We Aim is to Study the performance of STT-MRAM when we have Integrated it (Integration can be either Vertical or Horizontal) in the memory Hierarchy to benefit from the power consumption, improve the computational power or speed at which instructions are serviced since existing memory technology are affected by leakage power consumption, especially for on-chip Cache memories. In today's world, Most of the applications are data-intensive such as social networking, weather monitoring, video games, online transaction, and video playback. the technology is growing up exponentially which results in the amount of digital data is produced increases exponentially. To store the information, improve the performance of the system, we need to have a memory Hierarchy, which will able to store the large volume of information that is produced as well as, should match up with the speed of the CPU and energy-efficient would be beneficial. So this is the reason we have decided to study the performance of STTMRAM Since it is energy efficient and non-volatile, having High endurance and has high computational power compared to all other Non Volatile memories.

1.3 Major contribution

We have Integrated STT-MRAM as the main memory in place of DRAM memory technology and Analyze the performance in terms of Dynamic energy parameters and power estimation since the disadvantage with DRAM memory are it is quite slow as information is stored in the form of charges on a capacitor and charge leakage with time needs to refresh when we perform the read operation. Energy

Overhead associated with refreshing the cells is almost close to 40 percentage of total energy consumption and leakage power becomes more and more, as transistors being scaled-down along with the processor node significantly. When we have integrated STTMRAM vertically in the memory Hierarchy as the main memory, we presented a comparative performance analysis between STTMRAM and DDR3 and simulated the results using DRAMsim3 with four synthetic workloads with changing read to write instructions ratio. from the results,we have determined that STTMRAM is able to perform close to the performance of DDR3 when we have a read-intensive synthetic workload since the overhead associated with write latency and write energy is high.

1.4 Outline Of Report

The rest of the thesis has the chapters arranged as: In Chapter 2, we give a background of the prevailing NVM Technologies and make a comparative study between them, explain the replacement strategies, followed by understanding of the STTRAM technology and Cache Design Optimizations. Also in this chapter we present a brief Literature Survey as well. Chapter 3 discusses about the Simulators and Benchmarks used. Chapter 4 presents all the simulations performed, its configurations and results which are obtained thereafter. Finally in Chapter 5 we discuss our Conclusions and present the possible future work.

Chapter 2

Background

2.1 Non Volatile Memories

In recent years, there has been a lot of interest in Non-Volatile Memory (NVM). With numerous benefits such as low density, non-volatile nature, and low standby power consumption, NVM is gaining a place in the memory hierarchy and may someday revolutionize our vision of computer design. Many NVMs have emerged, such as Flash memories Magnetoresistive random access memory(MRAM). Phase change random access memory(PCRAM). Resistive Random access memory(ReRAM). Ferroelectric Random access memory(FeRAM).

2.1.1 Flash Memory

In 1984, the Japanese Engineer, Fujio Masuoka, Invented an electrical storage medium at the Toshiba Corporation. As It would not require any energy to save data and could keep it for many years. It is used as a solid-state drive in laptops or desktop PCs. It is in our smartphones, memory cards, thumb drives, and so on. There exists a big variety when it comes to computer memory. They can be categorized as volatile and Nonvolatile memory storage mediums. Flash memory, however, is classified as Non-Volatile, it stores bits electrically and technically counts as an EEPROM or Electrically Erasable Programmable Read-only memory. It means that data can be programmed into and erased easily from the device. For example, EPROMS can be programmed electrically too, but are not electrically erasable.The fundamental building block of conventional flash memory is the Floating gate MOSFET or Flash cell. Floating-gate Mosfets have a structure similar to one of the normal MOSFETS and the only difference being a floating gate between the control gate and the substrate. this Insulated extra gate has the key function to store electrons thus enabling data storage. Let's suppose the Floating gate MOS stores a logic '1', which means that the cell is erased and no electrons are trapped in the floating gate. To read the data we apply the shown voltages to the respective terminals and therefore creating a conducting channel between source and drain. In addition, a current sensor is used to measure the flowing current and the flow of electrons is then being interpreted as a Logic "1".On other hand, A logic '0' is stored if electrons are trapped in the floating gate. Reading the data requires the same voltage as before. This time around we will not measure a current because the trapped electrons have raised the threshold voltage of the flash cell. therefore, only higher voltages at the control gate create a conductive channel.

We now know how logic '1' or 'o' is represented in a flash. How about Programming or erasing the flash cells?. To program a logic 'o', a conducting channel is formed .the relatively high voltages cause electrons to increase their velocity horizontally. Also, electrons interact with the vertical electric field of the control gate and letting certain electrons tunnel through the gate oxide near the drain terminal. As a result, trapping them in the floating gate. These electrons can not escape from there for decades, saving data without the need for energy. Assume we now want to erase the content of the cell, so it holds a logic '1'. We apply the required voltages to push the right amount of electrons out of the floating gate into the substrate. the method of programming and erasure has a destructive effect on the flash cell.

2.1.2 PCM

Phase change memory uses chalcogenide material to store the information in the form of resistance. It is actually a thin layer, made of germanium antimony tellurium material and two electrodes are used to wrap on both sides of chalcogenide alloy and in addition, we have used a heater. Phase change memory is a resistive non-volatile memory that utilizes different resistance to represent the bit information. This chalcogenide undergoes an electrically induced fast amorphous-to-crystalline phase transition. The quantity of heat generated by an applied electric pulse determines the phase. If the pulse width is brief but the voltage is large, the chalcogenide material changes state to amorphous (logic 0), indicating that the chalcogenide is made up of GST alloy and is heated above the melting point. On the other hand, A crystalline state will arise if the pulse width is too long and a low voltage is given to phase change memory (logic 1). The germanium antimony tellurium (GST) chalcogenide material is heated beyond the crystallization temperature (Tcrys) but below the melting temperature. The write performance is thus determined by the longer operation that is the SET. We can easily read without disrupting the material's phase after a phase has been created; the ratio between the resistance of the material in a SET and RESET phase is fairly large. In a single cell, we may store multi-bit information.

2.1.3 ReRAM

Resistive RAM (ReRAM) is a type of memory that stores its logical state in the form of a resistive state. ReRAM's properties allow it to be integrated into a wide range of application domains, including consumer electronics, personal computers, medical applications, space applications, and automotive applications. ReRAM is, in fact, compatible with traditional semiconductor fabrication processes.ReRAM is a promising technology since it consumes less power than PCM and has a higher density than MRAM. ReRAM falls within the category of memristor devices. A memristor is a two-terminal resistor with variable resistance. The resistance value varies with the amplitude and polarity of the applied voltage, as well as the duration of the voltage application. The fundamental feature of a memristor is that the resistance value does not vary when the power is switched off, giving it the attribute of non-volatility. In a memristor, a low resistance value is regarded as a logical 1, whereas a large resistance value is regarded as a logical 0. To read the value of a memory cell, the sensing amplifier compares the current flowing through the selected RERAM cell to the current flowing through a reference resistor with the same voltage applied to both. A ReRAM cell is a two-terminal Metal-Insulator-Metal (MIM) device made of an insulating or resistive material with two metal conductors separated by an oxide insulating layer. The fundamental principle underlying ReRAM is the Low Resistive State and High Resistive State of a conductive filament, with the goal of shunting the top and bottom electrodes (M) through the resistive oxide layer.

2.1.4 FeRAM

Ferroelectric ram is nothing but a random access memory that has a ferroelectric capacitor in its circuit and its hysteresis to achieve non-volatility. It is also widely known as FERAM. This FRAM is very similar to dram construction and, in functionality, it is similar to flash memory. The basic structure of FRAM is very much similar to dram although it contains ferroelectric material instead of dielectric material. To conduct read or write operations, FRAM comprises a word line, a bit line, one transistor, and a ferroelectric capacitor. We know that FRAM uses ferroelectric material instead of dielectric material, Ferroelectric material has a crystalline structure with an atom at the center. this atom has two equal and low energy states based on the position of the atom. The key to the working of FERAM is that the capacitance of a ferroelectric capacitor is changeable, allowing two states to be created for a memory cell. When the capacitor is not switched but an electric field is supplied, i.e. when there is no change in polarization, it operates normally linearly however If it switched, an additional charge was induced, which must have occurred as a result of the increased capacitance. An active element FET was employed to achieve this effect. To enable the individual cell to access a word line, a bit line is used in the FRAM memory cell. The read operation of FRAM memory necessitates a certain number of steps, which is quite similar to that of dynamic ram. The voltage on the bit line is simply compared to the voltage on the reference line. This reference is set two levels above unswitch voltages and two levels below switch voltages and After that, the sense amplifier works as a comparator, magnifying the difference to produce a logic ‘1’ or ‘0’. If we examine the sequences more closely, we can see that the cell is initially inactive when the bit line is low and the word line is low. The memory access sequence is then processed; the access procedure begins with voltages being applied to the word line. When applying a voltage across a capacitor, two scenarios must be considered, when the capacitor state switches and these voltages produce a field across a ferroelectric capacitor. It switches, and the switching operation causes a change that is shared by the bit line capacitance represented by C and the switched ferroelectric capacitor CS.

The resulting voltage on the bit line is, therefore, is proportional to the ratio of capacitances. There will be no additional charge induced if the capacitor state does not switch. It can be seen that the data within the cell can be changed during the read operation, which is a destructive process that necessitates the cell being rewritten if it is changed. The writing process in FRAM follows the same principles as the reading operation in that the control circuitry applies a field in the desired direction. Even if the power source is switched off after the data is written into the cell, the data will stay intact.

2.1.5 STTMRAM

In an STT-MRAM memory, information is stored in a magnetic tunneling junction element. Magnetic tunnel junctions are typically made up of one oxide layer and two ferromagnetic layers, which are distinguished by their magnetic orientation. One of the two ferromagnetic layers is known as a free layer because its magnetization direction is changeable, while the other ferromagnetic layer is known as a fixed layer because its magnetization direction is fixed. To change the direction of magnetization of a free layer, we must apply a sufficiently enough current across the cell. The amount of current required to write the cell is determined by a number of parameters, including the physical dimensions of the cell and the materials used, the operating temperature, and the length of the applied voltage signal.

2.2 Replacement Strategies

Which block should be replaced when there is an occurrence of a cache miss?

Usually, Cache has a finite capacity, and the program that we are going to execute will be generally larger than the size of cache memory. So During the execution of a program, we may have to bring in new data into cache memory and take out data that already there in the cache. What do we do when the cache is full?.let we say cache is Direct mapped ie for every block in main memory there exists a predefined location in the cache. When a new request(address) comes from the processor, upon searching in the cache, if we did not find it is called a cache miss then the block, when it is brought from the main memory has a predefined location in the cache. So whichever block when it is already existing in the predefined location has to be eventually taken out. There is no choice as for as direct mapping is concerned, wherever is the mapping location, to that location, we have brought, and whoever is there in the location has to be replaced. that is why the block replacement technique is very simple and it is a straightforward algorithm. And there is no choice as far as the design is concerned regarding which block will be replaced. this is the case with the direct-mapped cache.

Which block to be replaced when the cache is going to be set-associative cache? Consider the case where we have a CPU that is going to give us a 32-bit address. this 32-bit address is divided into 22 bit as tag and 8-bit index and the remaining bits are going to be offset. Using this 8-bit index, we are going to index it to One of the 256 sets and Cache is 4 way associative, and Unfortunately, we have found that none of the tag matching comparisons going to give a hit. That means the Given Address has encountered a miss. Once if it is a miss, we go all the way to the main memory and we going to bring a block of data and where this block of data going to reside, it has to reside in the same set because the set number is prefixed from the address. Now in the set, all the four ways are full and it is occupied by some other data. Out of these four blocks that are residing in the set, which is to be replaced. Block replacement algorithm Applicable only in the context of set-associative Cache, whenever there is a miss occurring on a given set index, out of all available blocks that are there in that particular set, One has to be picked. Block Replacement policy is purely restricted to which is the block that we are going to evict for replacement.

1. **Random** Random policy which should generate a pseudo-random number generator.It is doesnot consider to take any advantage of Re-reference of blocks.

2. **Least Recently Used** When we have an associativity is equal to two,least recently used is equivalent to Not Most recently used.Let us say we have a 2-way associative cache and we are adding one extra bit,which is called as LRU bit ,whenever we are going to access a block,we will make LRU bit as one for that block and LRU bit of other block is eventually set to zero. Every access that we make into the particular set ,whichever block we are going to access ,we make LRU bit of the corresponding bit is one and other blocks LRU bit are cleared. If we implement this policy.at a time of replacment,we look into both the blocks(2-ways),which block is having an LRU bit as zero ,that was not recently used and other block which is having one in this LRU bit

,that is recently used .For associativity slightly bigger than two like four way associative cache or eight way associative cache ,single LRU bit won't work .LRU requires counters to store the information about the entries.

3. **Least Frequently Used** It is a block replacement policy, which evicts the blocks that are least frequently used when there is a cache overflow or when there is a Cache miss.

4. **Most Recently Used** It is a block replacement policy, which evicts the blocks that are most frequently used when there is the occurrence of a cache miss or Cache Overflow.

2.3 STTMRAM

A brief introduction being given about STTMRAM we now look deeper into the device in this section and understand about its working principle, read/write operations, and other features in greater detail.

2.3.1 Working Principle

STT-MRAM is a nonvolatile memory that uses magnetization direction in a ferromagnetic material for the data storage and magneto-resistance to read the stored data. STT-MRAM is a memory device that stores data by using electron spin in ferromagnetic materials. A ferromagnetic material that is highly susceptible to magnetization.

STT-MRAM has been called the ideal memory as no refresh is needed to retain the data which results in less power consumption. Why STT-MRAM required in the first place?. In a DRAM-based memory technology, It is a type of memory where the data stored in the form of charge in a capacitor, the cost of the dram cell is less compare to the cost of SRAM cells and it is used for high-density application.

The disadvantage with DRAM-based memory technology is that we need refresh cycles to store the data as data can be lost due to leakage of charge in a capacitor. In an SRAM-based memory technology, It is static memory and can regain a value for as long as the power supply is not interrupted.it is typically fast and used in high-speed memories such as Cache memory. The disadvantage of SRAM is that cost per memory cell is high In Flash kind of memories, It is a nonvolatile memory technology that can regain the data as long as possible even if the power supply is interrupted. The disadvantage of flash memories is that rate at which the data is stored is slow and it uses a lot of power.

STT-MRAM is an ideal device which combines the best feature of the three memory technology we have discussed so far. It combines the high density of DRAM with the speed of SRAM and the Nonvolatility of FLASH memory or hard disk, and all this while consuming very amount of power.

MRAM is well suited for military and space applications since it can withstand high radiation and function at severe temperatures. The magnetic tunnel junction is like the heart of MRAM. Two ferromagnetic layers are separated by a tiny dielectric barrier in the magnetic tunnel junction. If a bias voltage is placed between the two metal electrodes, electrons can tunnel through the barrier. In MTJ, The relative orientation of the magnetization of the ferromagnetic layers determines the tunneling current, which may be altered by applying a magnetic field.. this phenomenon is called tunneling magnetoresistance is abbreviated as TMR. The direction of polarization of the free magnetic layer is used for information storage.

2.3.2 Read and Write Operations

Read Operation:

STT-MRAM is consists of one transistor, and a magnetic tunnel junction is used to store the information in the form of resistance. During read Operation, a very

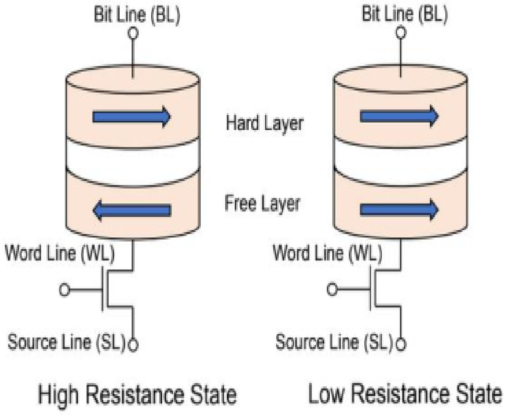


Figure 2.1: STT-MRAM CELL

small voltage is applied between the source line and bit line .the Transistor is in On state as the word line is set to VDD.the current flows through the Magnetic Tunnel Junction, and resistance due to the current is measured. If the resistance is high, and magnetization is in an anti-parallel direction. then the digital value '1' is measured. if the resistance is low and magnetization is in a parallel direction , then the digital value '0' is been read

Write Operation:

To write data into the cell, we must put a positive voltage between the source and bit lines to change the magnetization orientation of the free layer. And if the magnetization direction of the two layers is parallel. Hence value written into the cell will be '0'. To write '1' into the cell, if a high voltage difference is applied between the bit line and source line, the magnetization direction of the free layer will become opposite to that of the magnetization direction of the fixed layer. Hence the value written into the cell will be '1'.The amount of current required to write the cell is determined by the cell's physical size and the materials employed.

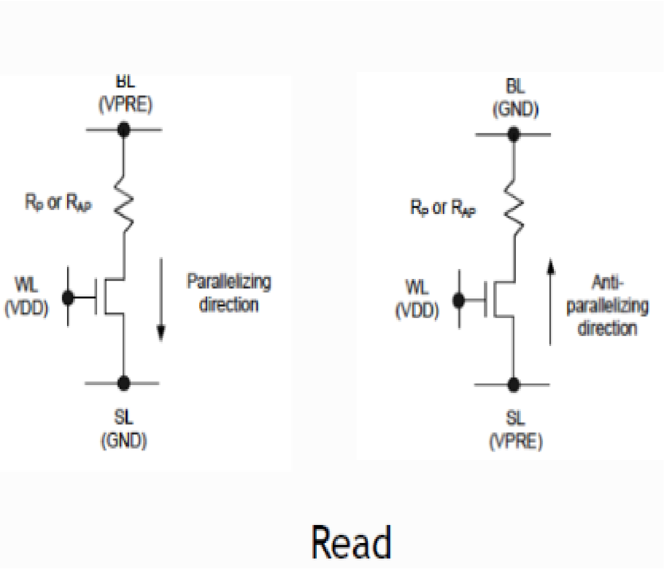


Figure 2.2: Read Operation of cell

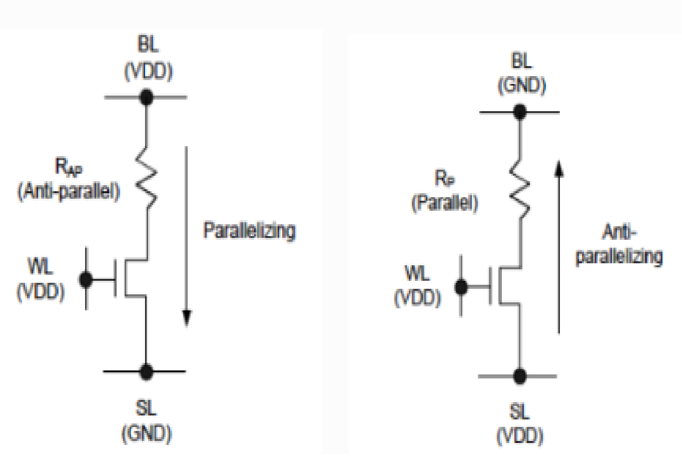


Figure 2.3: Write Operations of cell

2.3.3 STT-MRAM Endurance

The maximum number of times the cell can be written without degrading the performance of the cell. Typically the endurance is 10^{15} and is comparable to the endurance of DRAM and SRAM. Among all the non-volatile memories STT-MRAM has the best endurance. when we performing the write operation, since the write current required is quite high, due to high current, the performance of the cell degrades quickly after few numbers of such write operation and the lifetime of the cells reduces. Magnetic tunnel junction materials are exposed to thermal instability, which could lead to data loss. The probability of failure of cell functioning depends on the junction's implementation, the temperature, the time required between the write and the previous reading of data, and the number of memory cells. The thermal instability increases as the scale of technology decreases.

2.4 Comparison of STTMRAM with other NVM

The table 2.1 gives a brief overview of all the Memory technologies, based on cell size, read & write latency, write endurance, leakage power, dynamic energy and the maturity of the technology. It can be observed that based on all the tradeoffs present in the comparative study STTMRAM has one of the most promising opportunities to be the on-chip memory of the future.

Features	SRAM	DRAM	NAND Flash	STTMRAM	ReRAM	PCM	FeRAM
Cell Size(F^2)	120-200	60-100	4-6	6-50	4-10	4-12	6-40
Write Endurance	10^16	$> 10^{15}$	$10^4 - 10^5$	$10^{12} - 10^{15}$	$10^8 - 10^{11}$	$10^8 - 10^9$	$10^{14} - 10^{15}$
Read Latency	0.2-2ns	10ns	15-35μs	2-35ns	10ns	20-60ns	20-80ns
Write Latency	0.2-2ns	10ns	200-500μs	3-50ns	50ns	20-150ns	50-75ns
Leakage Power	High	Medium	Low	Low	Low	Low	Low
Dyn Engy(R/W)	Low	Medium	Low	Low/High	Low/High	Med/High	Low/High
Maturity	Mature	Mature	Mature	Test Chips	Test Chips	Test Chips	Manufact.

Table 2.1: NVM Characteristics [1], [2]

2.5 Literature Review

Rabiee et al. use STTMRAM as an L1 level cache and reduce the total number of writes by introducing a new cache coherency protocol. According to them among all the write requests to a cache, only the last one needs to reflect its result [3]. Retention time of STTMRAM is one of the open open research problems which is getting a lot of attention both from device physics perspective and also from the domain of computer architecture. Kyle et al. [4] propose LARS (Logically Adaptive Retention time STTMRAM) cache which can dynamically adapt to applications runtime requirements and it shows to reduce the average cache energy by 25%. Kyle Kuan et al. in their another paper [5] address the retention time issue with STTMRAM, this time the paper analyzes the effect of unused prefetches, which is responsible for the cache pollution, and propose two schemes Prefetch-Aware Retention time Tuning (PART) and Retention time-based Prefetch Control (RPC), which reduced the average cache energy and latency by 3.50% and 3.59%, respectively, and the hardware overhead by 54.55%.

Yazdanshenas et al. [6] use STTMRAM as Last Level Cache and address the challenges which are still prevalent in STTMRAM i.e. the limited write endurance and high write energy. They propose a coding method to exploit the principles of locality and reduce the overall number of writes to the LLC. [7] addresses the read latency issue in STTMRAM when used as an L1 Data Cache and provide certain micro-architectural modifications and appropriate code transformtion, such that the performance penalty comes within the tolerable limits. Kwon et al. in his paper [8] propose AWARE(Asymmetric Write Architecture with REdundant Blocks) as they bring in a possible solution to the write latency in the STTMRAM as a lower level memory. Here they identify redundant blocks, which are then preset so that faster write transitions can happen. It was observed that write latency significantly improved when it is written into redundant blocks compared to the conventional memory write methods. Another major aspect of Hybrid Caches are being explored and the paper by hongbin Sun et al. [9] pioneers the domain where they propose a Hybrid Cache involving SRAM and STTMRAM to mitigate the prone soft errors and reduce the overall power consumption by about 76% with only 2% performance degradation compared to the present caches. [10] discuss the tradeoff of the STTMRAM as both Cache and DRAM in terms of loss in latency performance but an immediate improvement in the power values. [11], [2] and [1] provide extensive survey and comparative study for all state of the art NVM technologies.

2.6 Design Optimization Target.

Using NVsim, this simulator gives us a lot of flexibility about Design Optimization such as accessing mode of the cache, power, energy, performance also it provides the best Solution for Optimization target in terms of estimated cell and total memory area, cache hit/miss latencies, read/write latencies, and precise power estimates of every block used, shown in fig 1.2. We have simulated the performance of STT-MRAM as L1 Cache for the following Cache Design Optimization.

2.6.1 Read/Write Latency Optimization

We can achieve an optimal solution for read/write latency using NVsim. Since NVsim models the optimization analyitcally it is challenging to estimate precisely the improvements in the memory design but these results as per the author have been validated with real hardware. We presume that for read latency optimization the authors might have improved the sense amplifiers, whereas for the write operation, the device latency is reduced such that the retention time gets lowered but the overall write latency is improved. Some additional peripheral circuits may also have been used to accelerate the read and the write operations, resulting in more Total area and also power consumption, these speculations can be verified from the understanding of the table 4.3 and table 4.2.

2.6.2 Read/Write Dynamic Energy Optimization

The focus in this optimization is to optimally improve the latency without incurring much cost on the Silicon Area. Optimal performance in terms of power and read latency can be achieved with Read Dynamic Energy Optimization as compared to Write, the read operation is realtively easy to handle. The direct consequence of Write Dynamic Energy Optimization is missing out on the endurance and the retention time benefits of STTMRAM. Therefore, the Write Dynamic Energy Optimization is more meticulously dealt with, and what we observe is from the table 4.3 and table 4.2 is that among all optimizations, this performs the worst and the issue has received a lot of attention to be addressed as a possible research problem [2].

2.6.3 Read/Write Energy Delay Product Optimization

Read/Write EDP is by far the best optimization available as observed from our simulated data in table 4.3 and 4.2 it takes care of the both delay performance as well as the power performance optimaly. As already mentioned that NVsim uses analytical models to achieve this information, it may be little inaccurate from the current state of the art technology, but it has validated it results against real systems and circuits. Therefore we can observe that the read/write EDP can ensure better leakage power values and area use at the loss of very low latency, compared to all its counter parts. 4.3

2.6.4 More Optimization Strategies:Leakage, Power and Area

So far the optimizations we have talked about the performance optimizations as its major focus but it is worth mentioning that all the NVM technologies provide a lot of opportunities in terms of optimizations in Leakage, Power and On chip area. These optimizations are hot topics of interest and are being addressed in the number of ways, a few of these works could be refered from [11].

Chapter 3

Simulators and Benchmarks

3.1 Simulators Used

Our work utilizes a number of simulators for the absolutle accuracy of our findings. We design our high level architecture with support of SST 10.1.0 to which we feed our data from the NVsim simulator. The Structural Simulation Toolkit (SST) [12] was developed to explore innovations in the field of computer architecture where the Instruction Set Architecture, microarchitecture, and memory interact with the programming model and communications system. We use the following two components - Ariel and MemHierarchy, from the Strcutural Simulation Toolkit.

Ariel(SST Processor Model)

Ariel is a core emulation component that uses a UNIX-pipe-based connection to PIN-tool-based profiling dynamically stream instructions from a running application. Instructions generate memory read/write requests at the core, then forwarded to a memory interface-based memory hierarchy. The Ariel Processor model allows multiple memory pools, like a DRAM and a Flash-memory pool, to be present in the system, with differing memory bandwidths or performance characteristics.

MemHierarchy(SST Cache and Main Memory Model)

MemHierarchy is Object Oriented-based memory hierarchy simulator component is used in the SST to provide a real time modern memory system model where it constructs the intra-node and inter-node directory-based cache coherency architecture. The SST model of hierarchical directory-based cache coherence architecture is almost in line with the state-of-art advance computer architectures like Core i7 from Intel, as it uses the MSI and MESI Cache Coherence Protocols.

The SST MemHierarchy model is made highly configurable for the ease of its users to simulate the advanced architectures and look around for further innovations. MemHierarchy uses actual interconnect models and actual cycle delays to simulate contention and latencies between the caches, directory controller, and memory to achieve higher accuracy.

3.1.1 NVsim

NVSim is a circuit level simulator which gives an estimate timing, power, area information of a particular Memory Technology like SRAM, NVM or DRAM based on user provided configuration details [13]. The simulator models the area, timing, dynamic energy and leakage power of Phase-Change Memory (PCM), Spin- Torque-Transfer RAM (STT-RAM), Resistive RAM (ReRAM) or memristor, Floating Body Dynamic RAM (FBDRAM) and Single-Level Cell NAND Flash. NVSim uses a well known simulator CACTI as its modelling principal [14].

3.1.2 DRAMsim3

DRAMSim3 [15] is a cycle accurate modern DRAM simulator which is capable of both performance modelling as well as thermal modelling at runtime. It models the timing paramaters and memory controller behavior for several DRAM protocols such as DDR3, DDR4, LPDDR3, LPDDR4, GDDR5, GDDR6, HBM, HMC, STT-MRAM. The simulator developed in C++ as an objected oriented model that includes a parameterized DRAM bank model, DRAM controllers, command queues and system-level interfaces to interact with a top level architecture simulators like GEM5 or SST or may even be used for trace workloads. In our work however we use it for synthetic trace workloads.

3.2 Benchmarks

3.2.1 Rodinia Benchmarks Suite

Rodinia Benchmarks Suite [16] is widely used for benchmark evaluations on GPUs and CPUs to explore the chances of performance improvement in heterogenous computer architectures. The Rodinia suite has the following features: • The benchmarks suite has four applications and five kernels. These application and kernels with the help of OpenMP for CPUs and with the CUDA API for GPUs can be performed as a part of parallel programming model. Parallel programming allows to utilize the chip resources more effectively and efficiently. • The workloads exhibit parallelism, various data access patterns and data-sharing characteristics.

- **Back Propagation:** Back Propagation is a popular algorithm used in the domain of neural networks that trains the weights of connecting nodes on layers of a network. The application has two phases: the Forward Phase, in which the activations are propagated from the input to the output layer, and the Backward Phase, in which the error between the observed and requested values in the output layer is propagated backwards to adjust the weights and bias values. In each layer, the processing of all the nodes can be done in parallel.

- **Computational Fluid Dynamics Solver:**The Computational Fluid Dynamics solver is an unstructured grid finite volume solver for the three-dimensional Euler equations for compressible flow. Effective memory bandwidth is improved by reducing total global memory access and overlapping redundant computation, as well as using an appropriate numbering scheme and data layout.

- **KMeans:** K-means is a clustering algorithm used in data mining, requiring very high data parallelism. In k-means, a data object is comprised of several values, called features. By dividing a cluster of data objects into K sub-clusters, k-means represents all the data objects by the mean values or centroids of their respective sub-clusters. The initial cluster center for each sub-cluster is randomly chosen or derived from some heuristic. The algorithm associates each data object with its nearest center based on some chosen distance metric in each iteration. The new centroids are the expectation values of all the data objects within each sub-cluster, respectively. The algorithm iterates until no data objects move from one sub-cluster to another.

- **Speckle Reducing Anisotropic Diffusion(SRAD):**SRAD (Speckle Reducing Anisotropic Diffusion) is a diffusion method for ultrasonic and radar imaging applications by solving partial differential equations (PDEs). The speckles which are nothing but the locally correlated noise are removed using this algorithm such that no harm is done to the original image features. SRAD consists of several pieces of work: image extraction, continuous iterations over the image (preparation, reduction, statistics, and computation and image compression. The sequential dependency between all stages requires synchronization after each stage.

3.3 Metrics of Study

It is very important to understand what are our metrics of study, to clearly identify and achieve our goals. In this section we discuss few metrics which are intensively used in computer architecture and based on these the final performance evaluation is done.

1. **Cache Hits and Misses:** Our evaluation majorly concentrates on the behaviour of STTMRAM as L1 level cache, therefore the study of cache hits and misses become absolutely necessary.

$$\text{Cache Hits} = \frac{\text{Number of accesses available in Cache}}{\text{Total number of accesses made to the Cache}} \tag{3.3.1}$$

$$\text{Cache Misses} = \frac{\text{Number of accesses not available in Cache}}{\text{Total number of accesses made to the Cache}} \tag{3.3.2}$$

Therefore the final relation can be established as

$$\text{Cache Hit rate} = 1 - \text{Cache Miss rate} \tag{3.3.3}$$

‘ This diredtly affects the Average Memory Access Latency(AMAT)

$$\text{AMAT} = \text{Hit Time} + \text{Miss Ratio} * \text{Miss Penalty} \tag{3.3.4}$$

2. **Execution Time(Total Cycles):** Performance evaluation is done majorly on the IPC and the CPI values, since we observe the performance for various design optimizations for the same workloads therefore with loss of generality we may say that knowing the number of cycles used for simulations would be sufficient for us, hence we only observe the total number of cycles required to process a particular workload.

3. **Total Power:** Power estimations are absolutely vital for any modern day architecture, customers always want higher performance for very low power consumptions. Since we are observing the power values for design optimizations for every read and write operation into the memory cell, we obtain a certain estimated power value, we then multiply it with the total read and write instruction count to get the required power estimates respectively. It is worth mentioning the dynamic power formula as it forms the key foundation for individual read/write access power consumption.

   $$P_{dynamic} = \alpha C V^2 f \tag{3.3.5}$$

   where α is the switching factor, C is the capacitance in the cell, V is the voltage applied and f is the frequency of operation.

Chapter 4

Performance study of STTMRAM

4.1 Simulation Configuration Details

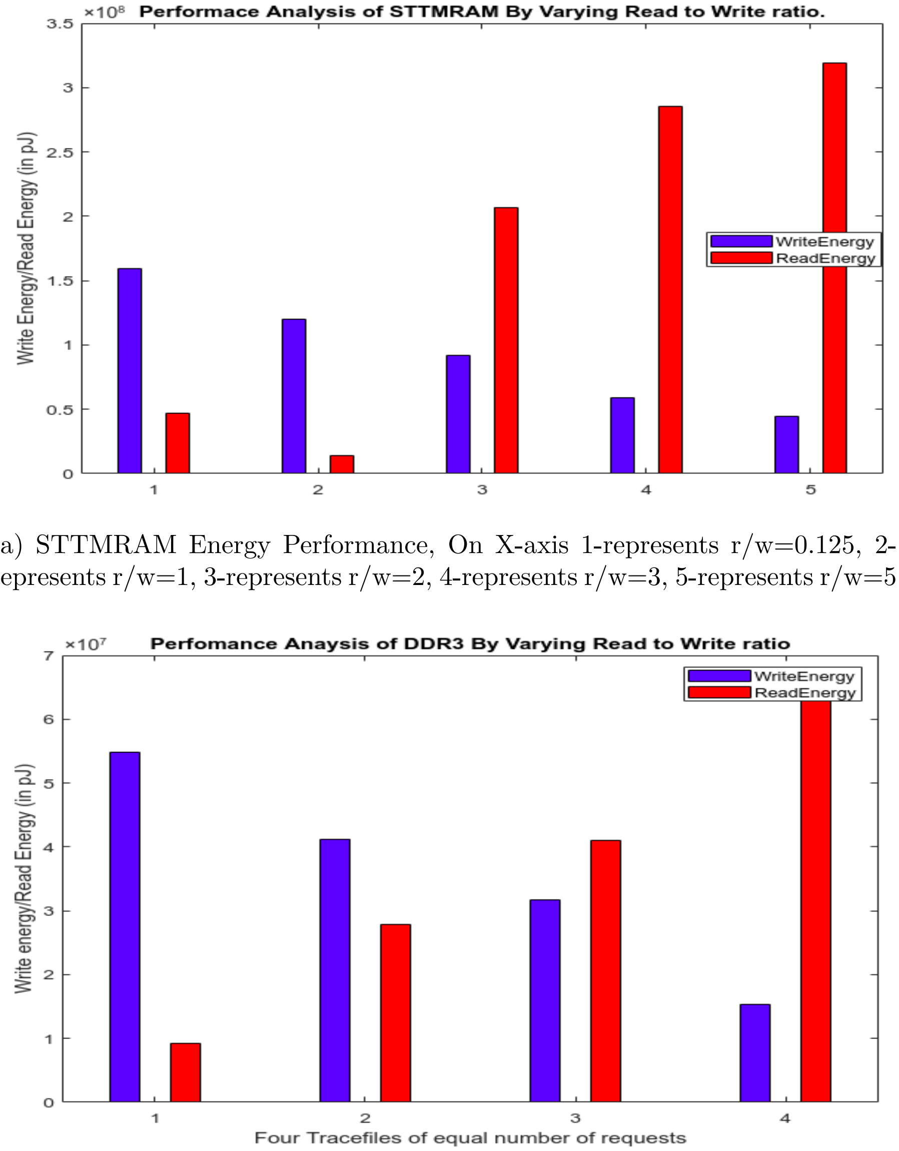
The top level architecture of our simulations have private L1 and L2 level caches for a multicore processor and then we have shared L3 level cache. For the first half of the study we use STTMRAM technology as the Main Memory then for the second half of our simulations where we explored the optimizations for STTMRAM as cache we use DDR3 as the conventional main memory. The simulator was run on a Linux Based OS(Ubuntu 18.04) with computer specs as Core i5-7th Gen Intel Processor and 8GB RAM. The following table 4.1 shares the precise information of the top level architecture.

Processor	4 core, OoO, 2.66GHz
L1 Cache	Private, 32KB, 2.66GHz, 2 way Assoc
L2 Cache	Private, 128KB, 2GHz, 8 way Assoc
L3 Cache	Shared, 4MB, 2GHz, 16 way Assoc
Cache Coherence	MSI
Cache BLock Size	64B
Main Memory	2GB
Bus Latency	50ps

Table 4.1: Top Level Architecture Configuration

4.2 Performance Study of STTMRAM as Main Memory

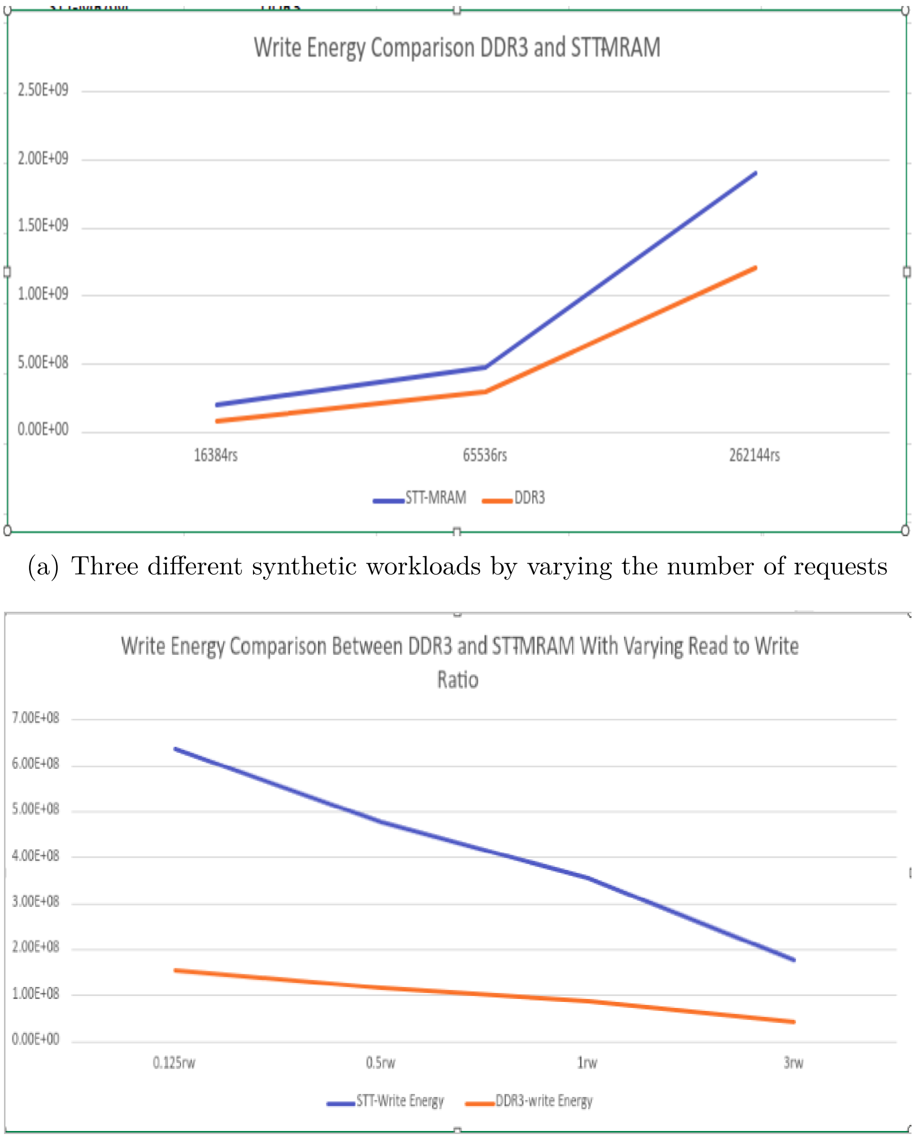
We are demonstrating the performance of STT-MRAM, when we replace STT-MRAM as DRAM memory technology in terms of its Dynamic Write energy and Dynamic read energy for a particular synthetic workload with changing the read to write instruction ratio. How the Energy parameters get effected for a particular workload when we are changing the read to write instruction ratio.



(a) STTMRAM Energy Performance, On X-axis 1-represents r/w=0.125, 2-represents r/w=1, 3-represents r/w=2, 4-represents r/w=3, 5-represents r/w=5

(b) DDR3 Energy Performance, On X-axis 1-represents r/w=0.125, 2-represents r/w=1, 3-represents r/w=2, 4-represents r/w=3

Figure 4.1: Write energy and read energy of a)STT-MRAM and b) DDR3 with respect to varying RW Ratio for a synthetic workload



(a) Three different synthetic workloads by varying the number of requests

(b) Four different synthetic workloads by varying the RW ratio

Figure 4.2: Write Energy Comparison between STT-MRAM and DDR3

4.3 STT-MRAM as L1 Cache

4.3.1 STTMRAM v. SRAM

In this section we implement the STTMRAM as an L1 cache and vary the replacement policies for different Design Optimizations. We make a comparative study of STTMRAM with SRAM as an L1 level Cache. Following are the configurations with which the study is made,

Optimization	Total Area	Read Lat	Write Lat	Read DE	Write DE	Leak Pow
SRAM	1830.977um^2	147.826ps	128.339ps	0.289pJ	0.227pJ	3.001mW
Read Latency	15551.041um^2	1.558ns	10.083ns	20.229pJ	40.863pJ	1.759mW
Write Latency	105426.495um^2	1.569ns	10.068ns	27.759pJ	44.522pJ	14.087mW
Read Dyn Eng	4432.653um^2	1.795ns	10.319ns	17.289pJ	36.405pJ	0.256mW
Write Dyn Eng	15409.324um^2	2.038ns	10.564ns	28.228pJ	36.291pJ	0.384mW
Read EDP	12108.053um^2	1.590ns	10.115ns	17.335pJ	36.323pJ	1.349mW
Write EDP	12108.053um^2	1.590ns	10.115ns	17.335pJ	36.323pJ	1.349mW

Table 4.2: Cache Tag Array Results Comparisons

Optimization	Total Area	Read Lat	Write Lat	Read DE	Write DE	Leak Pow
SRAM	25325.323um^2	0.177ns	0.162ns	31.351pJ	30.304pJ	41.712mW
Read Latency	63197.132um^2	1.587ns	10.110ns	299.458pJ	548.776pJ	17.329mW
Write Latency	63197.132um^2	1.587ns	10.110ns	299.458pJ	548.776pJ	17.329mW
Read Dyn Eng	41053.593um^2	1.706ns	10.240ns	244.494pJ	520.013pJ	8.259mW
Write Dyn Eng	116483.489um^2	3.085ns	11.630ns	709.623pJ	516.419pJ	3.966mW
Read EDP	41053.593um^2	1.706ns	10.240ns	244.494pJ	520.013pJ	8.259mW
Write EDP	41053.593um^2	1.706ns	10.240ns	244.494pJ	520.013pJ	8.259mW

Table 4.3: Cache Data Array Results Comparisons

The observations from table 4.2 and 4.3 are that even for various design optimizations in STTMRAM cache design, the conventional SRAM cache easily outperforms it in terms of latency and energy values. But compared to SRAM except Write Latency Design Optimization, all other design optimizations have extremely low leakage power and it can also be mentioned that the individual cell area of STTMRAM is extremely low, thereby providing really very high density values. The total area in the table 4.3 and table 4.2 consider all the additional blocks which are needed to ensure proper read and write into the STTMRAM which has already been disscussed in the section 2.3.2.

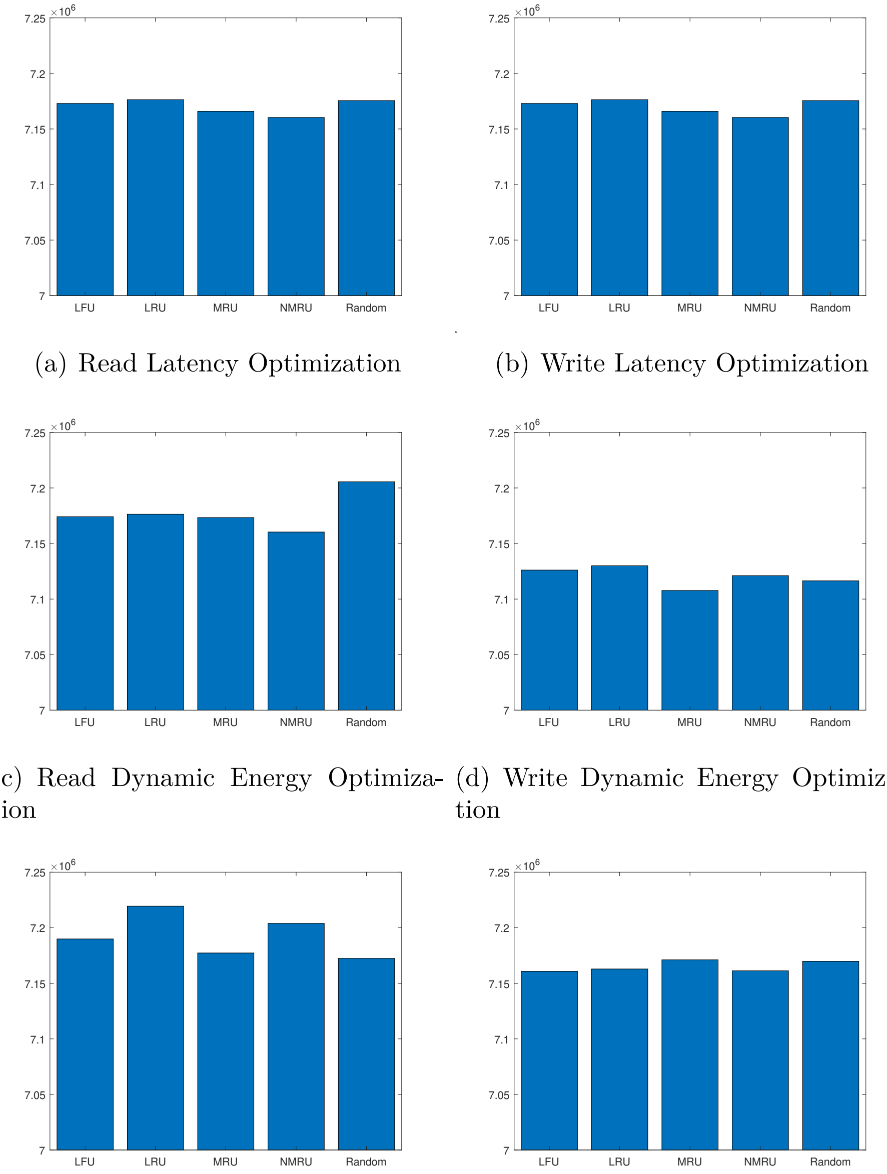
4.3.2 Study of Replacement Strategies for Design Optimizations

Optimization	Hit Latency	Miss Latency	Write Latency
Read Latency	1.587ns	1.558ns	10.110ns
Write Latency	1.587ns	1.569ns	10.110ns
Read Dynamic Energy	1.795ns	1.795ns	10.319ns
Write Dynamic Energy	3.805ns	2.038ns	11.630ns
Read EDP	1.706ns	1.590ns	10.240ns
Write EDP	1.706ns	1.590ns	10.240ns

Table 4.4: Optimal Cache hit latency,cache miss latency and cache write latencyfor each design Optimization Technique when we have integrated STT-MRAM asL1 Cache

Optimization	LRU	NMRU	MRU	Random	LFU
Read Latency	7176391	7160360	7165922	7175560	7172979
Write Latency	7176391	7160360	7165922	7175560	7172979
Read Dynamic Energy	7176391	7160360	7173375	7205569	7174151
Write Dynamic Energy	7130013	7121101	7107733	7116430	7126116
Read EDP	7219421	7203912	7177298	7172415	7189913
Write EDP	7162893	7161284	7171148	7169811	7160789

Table 4.5: Cache Hits under different Replacement Policy for each Optimization under Kmeans Workload for STTMRAM



(a) Read Latency Optimization

(b) Write Latency Optimization

(c) Read Dynamic Energy Optimization

(d) Write Dynamic Energy Optimization

(e) Read EDP Optimization

(f) Write EDP Optimization

Figure 4.3: Cache Hits for different Replacement Policies in STTMRAM for Kmeans Workload

Optimization	LRU	NMRU	MRU	Random	LFU
Read Latency	1854002	1853796	1824561	1851726	1854224
Write Latency	1853396	1853811	1841739	1852288	1854275
Read Dynamic Energy	1853028	1853967	1841609	1852097	1854008
Write Dynamic Energy	1853859	1853870	1841943	1852258	1854004
Read EDP	1853450	1853742	1825772	1852557	1854396
Write EDP	1853726	1853908	1833823	1852362	1853988

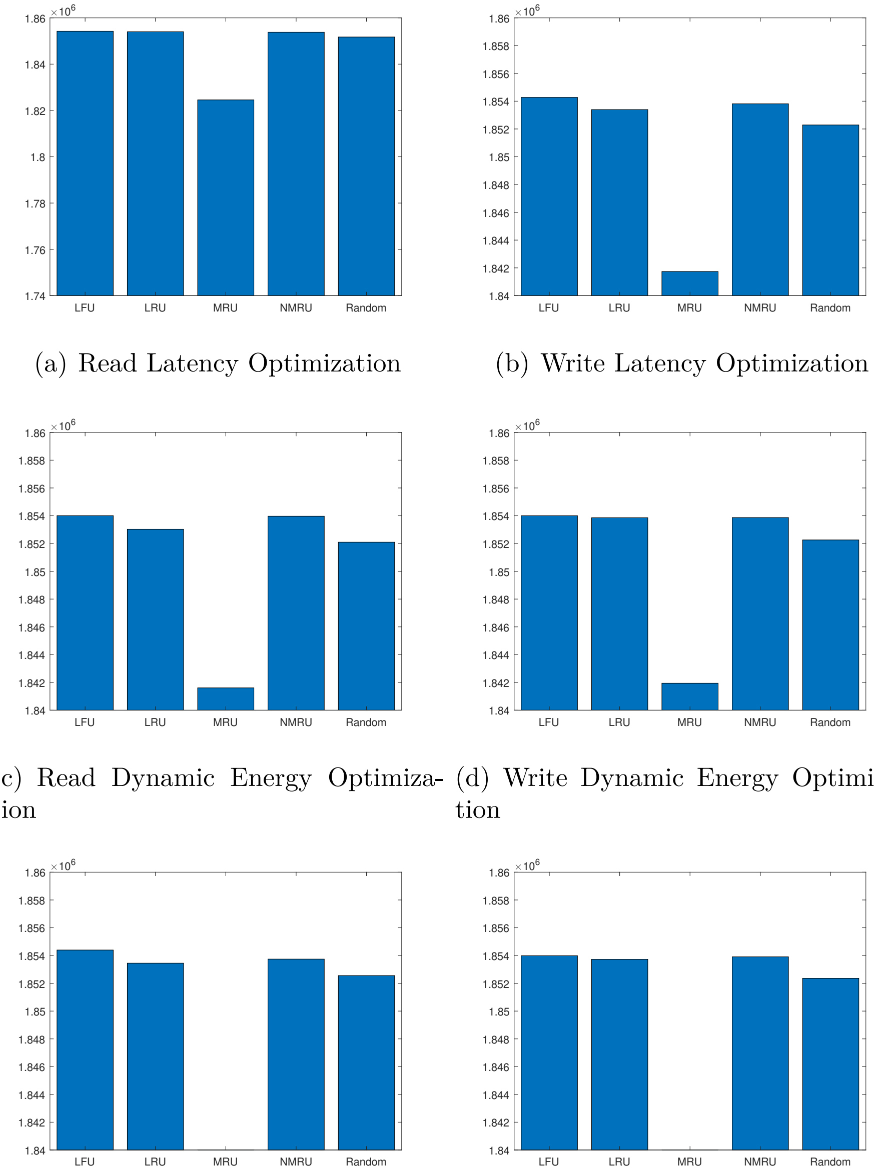
Table 4.6: Cache Hits under different Replacement Policy for each Optimization under Back Propagation Workload for STTMRAM

Optimization	LRU	NMRU	MRU	Random	LFU
Read Latency	7104260	7108082	7080062	7105473	7129944
Write Latency	7098568	7131850	7109994	7120323	7108389
Read Dynamic Energy	7156704	7169960	7099244	7164258	7160556
Write Dynamic Energy	7178632	7190392	7141356	7205525	7207396
Read EDP	7184850	7187292	7093230	7177985	7156968
Write EDP	7184850	7187292	7093230	7177985	7156968

Table 4.7: Cache Hits under different Replacement Policy for each Optimization under Kmeans Workload for SRAM

Optimization	LRU	NMRU	MRU	Random	LFU
Read Latency	1853893	1854117	1743060	1852190	1854491
Write Latency	1853963	1853826	1841518	1852388	1854143
Read Dynamic Energy	1853975	1854312	1843628	1851911	1853827
Write Dynamic Energy	1854042	1853333	1841708	1852138	1854150
Read EDP	1853972	1853501	1843537	1852328	1854390
Write EDP	1853847	1853780	1843422	1852093	1854261

Table 4.8: Cache Hits under different Replacement Policy for each Optimization under Back Propagation Workload for SRAM



(a) Read Latency Optimization

(b) Write Latency Optimization

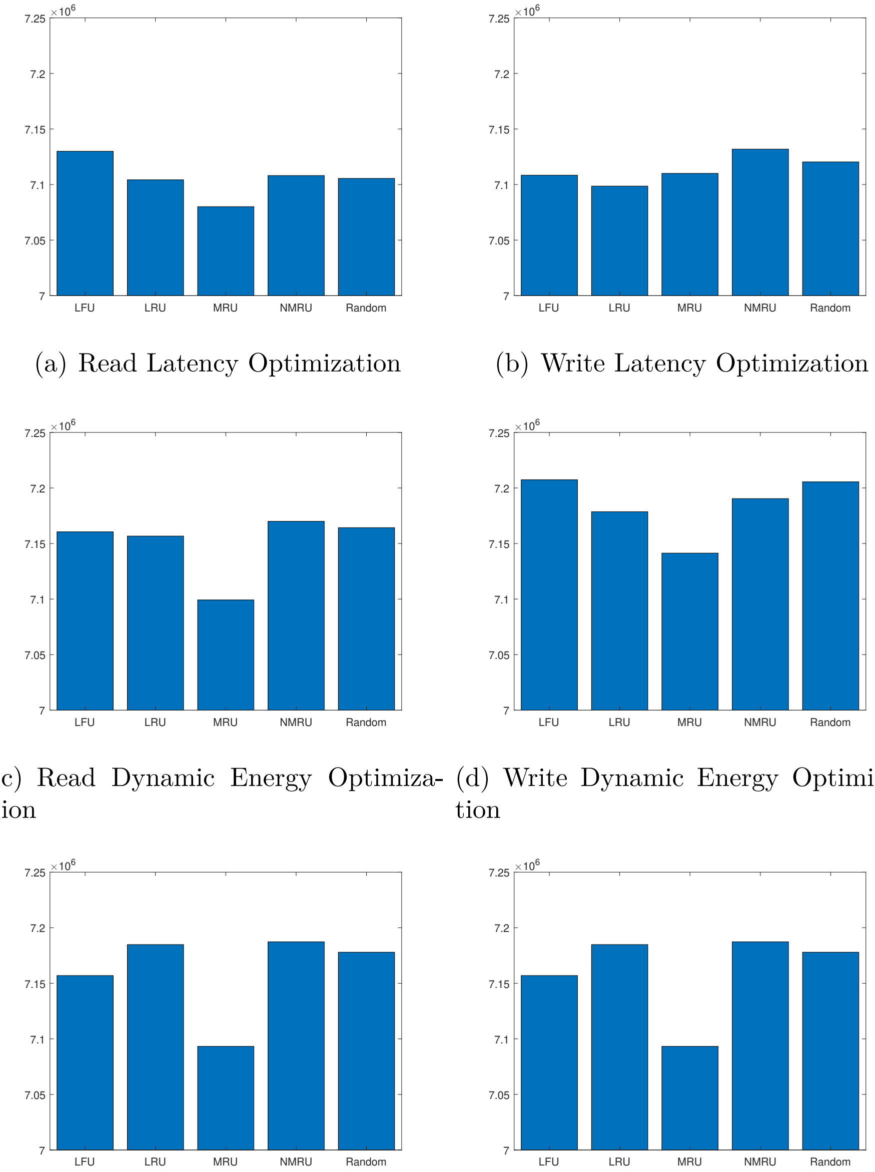
(c) Read Dynamic Energy Optimization

(d) Write Dynamic Energy Optimization

(e) Read EDP Optimization

(f) Write EDP Optimization

Figure 4.4: Cache Hits for different Replacement Policies in STTMRAM for Back Propagation Workload



(a) Read Latency Optimization

(b) Write Latency Optimization

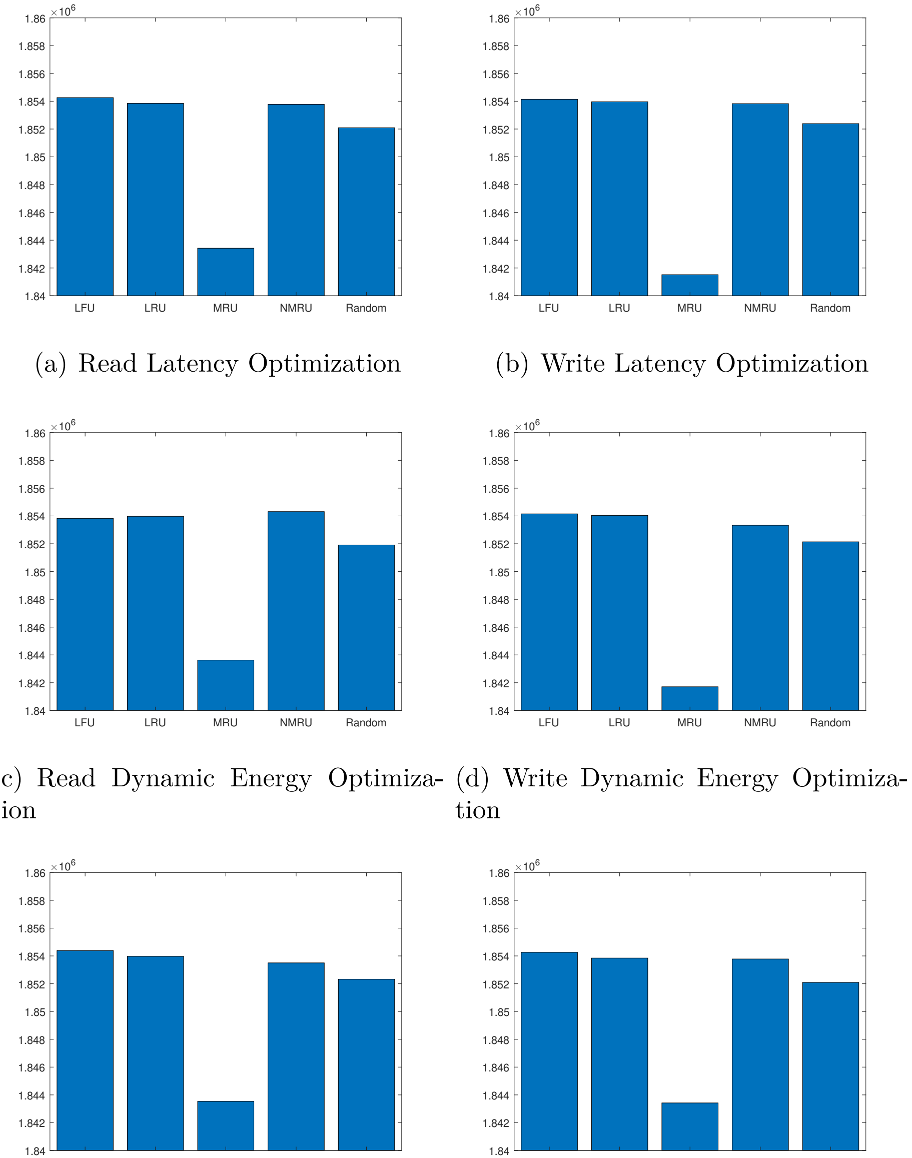
(c) Read Dynamic Energy Optimization

(d) Write Dynamic Energy Optimization

(e) Read EDP Optimization

(f) Write EDP Optimization

Figure 4.5: Cache Hits for different Replacement Policies in SRAM for Kmeans Workload



(a) Read Latency Optimization

(b) Write Latency Optimization

(c) Read Dynamic Energy Optimization

(d) Write Dynamic Energy Optimization

(e) Read EDP Optimization

(f) Write EDP Optimization

Figure 4.6: Cache Hits for different Replacement Policies in SRAM for Back Propagation Workload

From the Figure 4.6 and Figure 4.4 When compared to all other replacement policies, the performance of Cache in terms of cache hits under each Optimization Target for Replacement policy ”Least Frequently Used ” is superior for a given Workload. In terms of cache hits, STT-performance MRAM’s as an L1 cache under the replacement ”Most recently used” is the worst. The architectural elements of cache memory, such as cache size and associativity, will determine cache performance in terms of cache hits.

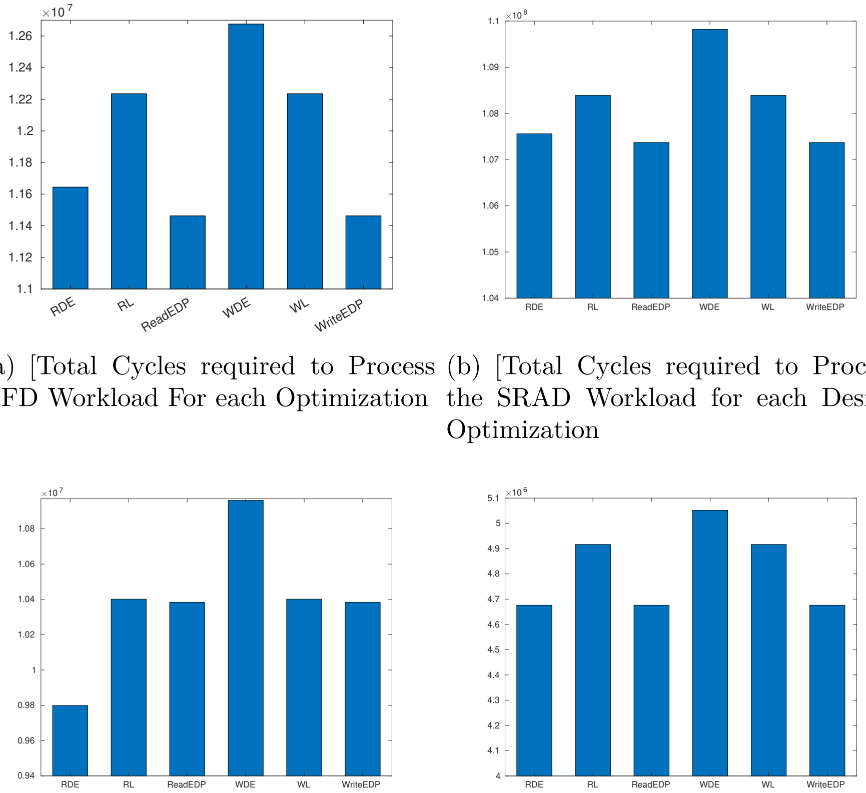
In a multi-core architecture with STT-MRAM as the L1 cache, For a particular OptimizationTarget, cache hits from Figure 4.6may not provide us with the best conclusion about cache performance.So, for a given workload, the OptimizationTarget takes less time (in total cycles) to Process the real workloads and latencies necessary to execute Cache hits and misses, allowing us to determine whether optimization approach performs better.

The figure 4.7shows that when we set the OptimizationTarget to READ Dynamic Energy and READ Energy Dealy, the performance improves because the workload is processed in fewer cycles. When using Write Dynamic Energy as a DesignTarget, more Cycles (Total Cycles) are required to process the given workload’s instruction count.

The performance of STT-MRAM as an L1 cache is given in Figure 4.8 for various Design Targets. Cache performance with Read Dynamic Energy and READ Energy Delay Product as a Design Target is superior since the workload is processed in less cycles. Because the cache hit, miss, and write latency are all minimal under these Design Targets.

STT-MRAM was utilized when When four different workloads were employed as an L1 cache, we looked at the overall energy consumption of read and write requests for each Cache Design Optimization Target. For each Cache Design Optimization, the performance of STT-MRAM is measured in terms of Total Energy Consumption for all Workloads.

From Figure **??** depicts the total energy consumption for each Optimization Target.The energy usage for all instructions is the least of all Design Optimizations



(a) [Total Cycles required to Process CFD Workload For each Optimization

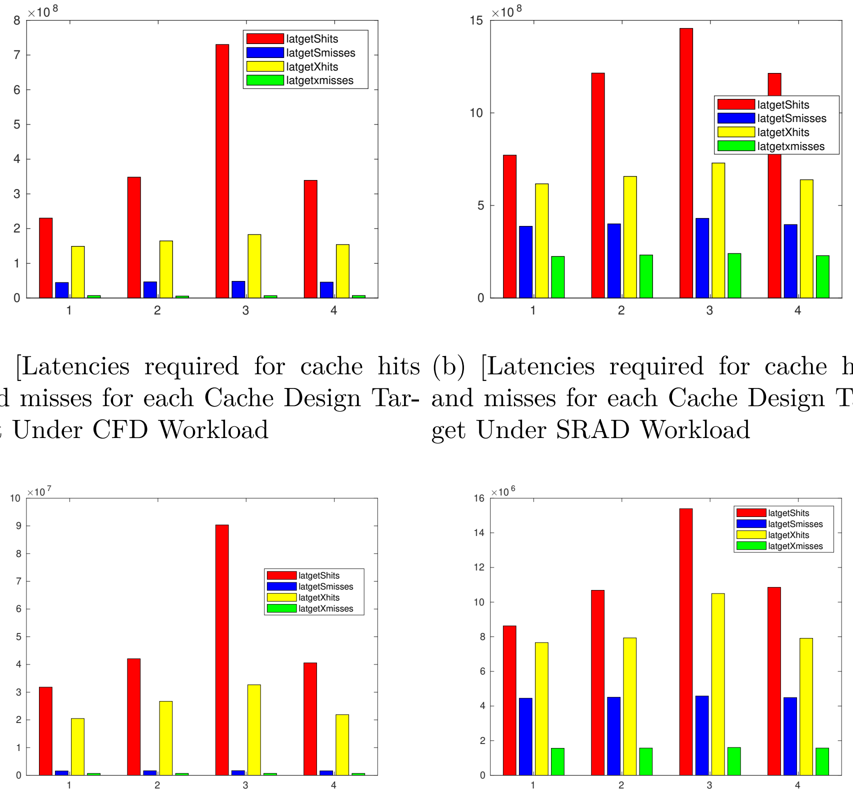
(b) [Total Cycles required to Process the SRAD Workload for each Design Optimization

(c) Total Cycles required to Process the kmeans Workload for each Design Optimization

(d) Total Cycles required to Process the Backprop Workload for each Design Optimization

Figure 4.7: Analyzing the Computational power of STT-MRAM as L1 Cache using Four Workloads

when using read Energy Delay Product Optimization and write Energy Delay Product. While compared to all other Design Optimizations, the energy consumption for all instructions is higher when using Write Dynamic Energy Design Optimization.



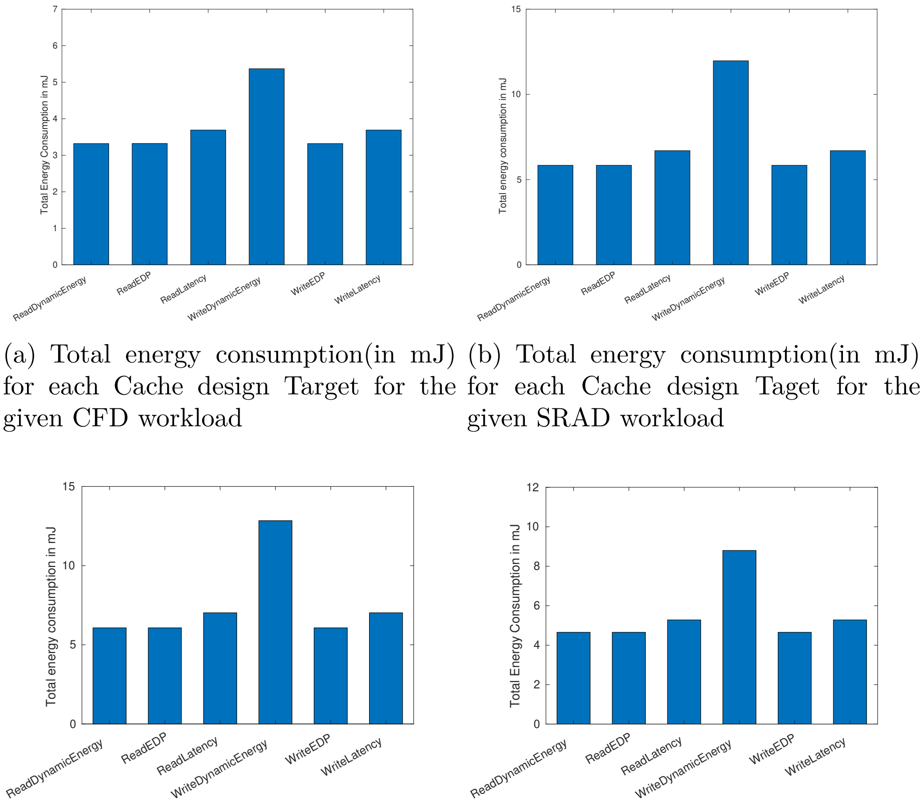
(a) [Latencies required for cache hits and misses for each Cache Design Target Under CFD Workload

(b) [Latencies required for cache hits and misses for each Cache Design Target Under SRAD Workload

(c) Latencies required for cache hits and misses for each Cache Design Target Under KMEANS Workload

(d) Latencies required for cache hits and misses for each Cache Design Target Under Backprop Workload

Figure 4.8: Performance Evaluation of STT-MRAM in terms of Cache hit Latency and Cache miss Latency as L1 Cache using Four Workloads



(a) Total energy consumption(in mJ) for each Cache design Target for the given CFD workload

(b) Total energy consumption(in mJ) for each Cache design Taget for the given SRAD workload

(c) Total energy consumption(in mJ) for each Cache design Optimization for the given Kmeans workload

(d) Total energy consumption(in mJ) for each Cache design Optimization for the given Backprop workload

Figure 4.9: Energy Consumption under each Workload

Chapter 5

Conclusions & Future Works

In this thesis, we Investigated the performance of STT-MRAM at different levels of a memory hierarchy. When we analysed the performance of STT-MRAM as the main memory technology and performed simulations, we observed that its performance in terms of energy consumption is exceptionally good because When we have read-intensive workloads, the overhead associated with write operations and write energy consumption is the major disadvantage. As a result, STT-MRAM must be horizontally integrated alongside DRAM-based main memory technology. As a result, read-intensive tasks may be routed to STT-MRAM while write operations are handled by DRAM-based main memory technology.Among all non-volatile memories that we have, STT-MRAM has the least read latency, high density, the lowest leakage power, infinite endurance, and the best write latency. As a result, it is a viable option to replace the present SRAM-based cache memory.In a multicore system with a private L1 cache as STT-MRAM cell-based cache for each core, the performance of cache hits under replacement policies was analyzed using the Rodinia benchmarks suite. Figures illustrate cache hits with all of these replacement strategies for each Cache Optimizationtarget. Cache performance, on the other hand, cannot be determined simply on cache hits and cache misses since cache hits and misses are dependent on cache associativity and cache size. So, first and foremost, we are assessing the performance of STT-MRAM with various Cache Optimization targets in terms of computing power and overall energy consumptions utilising four workloads.

Bibliography

[1] J. Boukhobza, S. Rubini, R. Chen, and Z. Shao, "Emerging nvm: A survey on architectural integration and research challenges," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 23, no. 2, pp. 1–32, 2017.

[2] S. Mittal and J. S. Vetter, "A survey of software techniques for using non-volatile memories for storage and main memory systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 5, pp. 1537–1550, 2015.

[3] F. Rabiee, M. Kajouyan, N. Estiri, J. Fluech, M. Fazeli, and A. Patooghy, "Enduring non-volatile l1 cache using low-retention-time sttram cells," in *2020 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*. IEEE, 2020, pp. 322–327.

[4] K. Kuan and T. Adegbija, "Energy-efficient runtime adaptable l1 stt-ram cache design," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 6, pp. 1328–1339, 2019.

[5] ——, "A study of runtime adaptive prefetching for sttram l1 caches," in *2020 IEEE 38th International Conference on Computer Design (ICCD)*. IEEE, 2020, pp. 247–254.

[6] S. Yazdanshenas, M. R. Pirbasti, M. Fazeli, and A. Patooghy, “Coding last level stt-ram cache for high endurance and low power,” *IEEE computer architecture letters*, vol. 13, no. 2, pp. 73–76, 2013.

[7] M. P. Komalan, C. Tenllado, J. I. Gomez Perez, F. T. Fernández, and F. Catthoor, “System level exploration of a stt-mram based level 1 data-cache,” in *2015 Design, Automation Test in Europe Conference Exhibition (DATE)*, 2015, pp. 1311–1316.

[8] K.-W. Kwon, S. H. Choday, Y. Kim, and K. Roy, “Aware (asymmetric write architecture with redundant blocks): A high write speed stt-mram cache architecture,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 4, pp. 712–720, 2014.

[9] H. Sun, C. Liu, W. Xu, J. Zhao, N. Zheng, and T. Zhang, “Using magnetic ram to build low-power and soft error-resilient l1 cache,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 20, no. 1, pp. 19–28, 2012.

[10] P. Chi, S. Li, Y. Cheng, Y. Lu, S. H. Kang, and Y. Xie, “Architecture design with stt-ram: Opportunities and challenges,” in *2016 21st Asia and South Pacific Design Automation Conference (ASP-DAC)*. IEEE, 2016, pp. 109–114.

[11] S. Mittal, J. S. Vetter, and D. Li, “A survey of architectural approaches for managing embedded dram and non-volatile on-chip caches,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 6, pp. 1524–1537, 2014.

[12] A. F. Rodrigues, K. S. Hemmert, B. W. Barrett, C. Kersey, R. Oldfield, M. Weston, R. Risen, J. Cook, P. Rosenfeld, E. Cooper-Balis *et al.*, “The structural simulation toolkit,” *ACM SIGMETRICS Performance Evaluation Review*, vol. 38, no. 4, pp. 37–42, 2011.

[13] X. Dong, C. Xu, Y. Xie, and N. P. Jouppi, “Nvsim: A circuit-level performance, energy, and area model for emerging nonvolatile memory,” *IEEE Transactions*

on Computer-Aided Design of Integrated Circuits and Systems, vol. 31, no. 7, pp. 994–1007, 2012.

[14] S. Li, K. Chen, J. H. Ahn, J. B. Brockman, and N. P. Jouppi, "Cacti-p: Architecture-level modeling for sram-based structures with advanced leakage reduction techniques," in *2011 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2011, pp. 694–701.

[15] S. Li, Z. Yang, D. Reddy, A. Srivastava, and B. Jacob, "Dramsim3: a cycle-accurate, thermal-capable dram simulator," *IEEE Computer Architecture Letters*, vol. 19, no. 2, pp. 106–109, 2020.

[16] S. Che, M. Boyer, J. Meng, D. Tarjan, J. W. Sheaffer, S.-H. Lee, and K. Skadron, "Rodinia: A benchmark suite for heterogeneous computing," in *2009 IEEE international symposium on workload characterization (IISWC)*. Ieee, 2009, pp. 44–54.