Exploring a Multiprocessor Design Space to Analyze the Impact of Using STT-RAM in the Memory Hierarchy

A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Nishant Ashok Borse

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
Master of Science Electrical Engineering

David J. Lilja

September, 2014

Acknowledgements

I would like to express my gratitude towards my advisor Prof. David Lilja for giving me this opportunity and direction to pursue my research and also for his continuous support, motivation and patience. His guidance was a huge help during the entire phase of researching towards my thesis. His mentoring gave me the required enthusiasm and encouragement to achieve this goal.

I would also like to thank the rest of my thesis committee : Prof. Kia Bazargan and Prof. William Cooper, for their insightful feedback, questions and recommendations.

In addition, I would thank my fellow lab mates William Tuohy and Cong Ma for enlightening me the first glance of research and also guiding and encouraging me throughout my work. It would be difficult to imagine my research having the right direction without the help of you guys. It was fun working together. I would also like to thank my friends in University of Minnesota; Rachit Agwania, Pushkar Nandkar, Samkit Jain and Prateek Khasgiwala.

Lastly, I would like to thank my family: my parents Ashok and Swati Borse for their immense support throughout and my brother Pushkar Borse for his encouragement.

Dedication

I would like dedicate my Thesis to my parents Ashok and Swati Borse who gave me this opportunity to study at graduate school.

Abstract

Spin-tronic memory is a promising technology and offers advantages due to its non-volatility and higher density. At the same time, based on device properties, there are trade-offs that decide the energy and performance penalty overhead. To decide these trade-offs its it imperative to understand the sensitivity of different parameters in the memory subsystem. In this work, we use a known statistical technique to analyze processor core and memory parameters for their sensitivity towards performance and energy for a Spin-tronic based memory hierarchy. We also study how does the sensitivity of processor core parameters like Re-order buffer, Load Store queue etc. vary when we replace a traditional SRAM memory with the new spin-tronic technology. Further, given a mix of different memory technologies and important processor core parameters, we use find the optimal configuration for delay, energy and area using the method of simulated annealing.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The need for a new generation memory technology is being extensively considered as more and more cores are integrated in todays CMPs. Current memory hierarchy is based on traditional SRAM technology. Though it has advantages of being fast, SRAM memories consume larger area given its low density bit cell as well as has high leakage. On chip cache take up most of the die area and the memory subsystem contributes heavily towards leakage and density. These are seemingly areas of concern and it is imperative to move towards alternatives that provide large capacity and low power caches. One of the promising alternatives is Spin Torque Transfer based Magnetic RAM (STT MRAM). STT MRAM based memory with its property of non volatility provides an excellent opportunity to replace SRAM in cache hierarchy and reduce leakage. It also provides higher density than SRAM which can be leveraged either by having a higher capacity cache or by reducing memory footprint thus allowing integration more cores into a single die. But STT MRAM comes with trade-offs with a higher bit write latency that leads to performance degradation. It also has a high write current resulting in a large dynamic write energy. Considering these trade-offs STT MRAM is being considered mostly as an alternative at the last level caches, although it can be a good fit for L1 caches if leakage and density is a major concern.

With a new technology being used in the memory subsystem, it is important to study the impact this technology has on various micro architecture parameters. Computer architects using this memory technology should know whether and how the design space

changes significantly at the micro architectural level. In this work, we look into important processor core and memory subsystem parameters and analyze their sensitivity towards performance and energy using the statistical technique of Plackett and Burman designs. Essentially, we identify important memory and processor bottlenecks for an STT MRAM memory based CMP. This is significant in selecting design space for the micro architectural parameters and further can be leveraged during various optimization phases. In addition, we analyze whether the sensitivity of these bottleneck parameters change as we replace STT MRAM with SRAM and should micro architectural design choices change in order to leverage trade-offs offered by the new memory technology. Our analysis show that sensitivity of bottleneck parameters towards performance as well as EDP fairly remain the same for a CMP with STT MRAM memory compared to SRAM. Although few parameters do show reduction in their significance across some benchmarks for performance and EDP but remain critical for others.

Further, this work looks into device choices that STT MRAM provides; especially the design options that trade-off density of a bit cell versus bit write time latency. Out of these STT MRAM bit cell devices, it is interesting to analyze which would be a better fit for L1 data and instruction cache. We consider a design space based on choices for memory technologies and important parameters from the earlier sensitivity analysis. Using these design space as a search space, we find the optimal configuration set for these processor parameters for minimum area, delay and energy using the method of simulated annealing. Our results indicate that the optimal configuration should have a STT MRAM based memory for both L1 data and Instruction caches and a low capacity STT MRAM based shared L2 cache for minimum ED^2AP. The solution also gives optimal values for other bottleneck processor parameters for individual benchmarks as well a global optimum across all benchmarks. This optimal set of parameters for processor and memory can further assist designers for appropriate parameter selection during further enhancements for STT MRAM based CMP.

Overall, this work provides a methodology for analysis and optimization for micro-architectural parameters which can be applied while studying new RAM technologies in general.

- Chapter 2 gives a background over the operation, design aspects and trade-offs of a STT MRAM bit cell. It also explains in brief the methodologies used in this work;

the statistical analysis technique of Plackett and Burman and the optimization method of simulated annealing.

- Chapter 3 briefly explains modeling for STT MRAM based array using CACTI and compares its metrics with SRAM.
- Chapter 4 describes the experimental setup for conducting the Plackett and Burman sensitivity analysis and explains the simulation methodology used for running experiments and obtaining statistics.
- Chapter 5 analyzes the most important bottleneck parameters for STT MRAM for performance and Energy Delay Product and compares it with the sensitivity results for SRAM.
- Chapter 6 describes the selection of design search space and the optimal results across it which gives a minimal solution for energy, delay and area.
- Chapter 7 provides conclusion and discussion.

Chapter 2

Background Work

2.1 STT MRAM Basics

Magnetic Random Access Memory is a new generation memory technology that promises to be an universal memory device due to its properties of non-volatility, zero standby leakage power and high density [5]. Work based in [6][7][8] [9] [10] points 4X gains in density. Unlike CMOS based memory which relies on storing information in the form of electric charge, Magnetic RAM uses magnetic storage. The most important component of a MRAM is the Magnetic Tunnel Junction (MTJ). Spin Torque Transfer MRAM (STT MRAM) is a new generation of MRAM and uses a similar MTJ for storing a binary bit. Figure 2.1 shows the conceptual view of a MTJ [6] [8]. An MTJ consists of three layers; a MgO tunnel barrier layer surrounded by two ferromagnetic layers. Out of the two ferromagnetic layers, the direction of one of the layers is kept fixed, while the direction of the outer layer is free and can be controlled by the current passing through it. The value of the bit stored is determined by whether these two layers are oriented in parallel or anti-parallel. Usually, two layers being in parallel represents a low resistance indicating a low '0' value stored, whereas the layers in anti-parallel represents high resistance and hence a high '1' value being stored. Thus, by controlling the direction of the current, binary data can be stored in the MTJ.

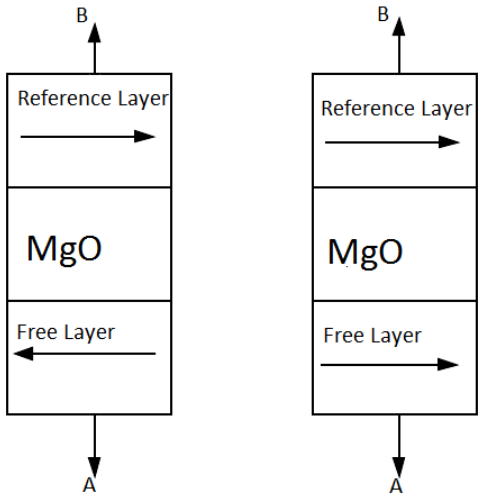

Figure 2.1: MTJ Conceptual View

2.1.1 STT MRAM bit cell

A STT MRAM bit cell typically consists of an NMOS access transistor in series with the MTJ. Figure 2.2 [6] [8] shows a schematic of a 1T-MTJ bit cell. In this schematic, a variable resistance represents the MTJ since the resistance across it changes gradually with the current flowing through it. The NMOS access transistor controls the current flowing through the MTJ. The Word Line (WL) is connected to the gate of the NMOS access transistor and functions similar to the traditional SRAM i.e. it turns on the access transistor during read and write operations. The source of the NMOS access transistor is connected to the Source Line (SL) whereas the Bit Line (BL) connects the free layer of the MTJ [6].

2.1.2 STT MRAM Operations

As discussed earlier, the direction of the free layer decides the value of the binary bit stored in the MTJ. Thus, by passing current through the device and sensing its value, the stored bit can be read. Usually, a small negative voltage is applied between BL and SL, typical value being -0.1V [6] [8]. On application of voltage, a small current flows through the MTJ which depends on the resistance between the layers. Using

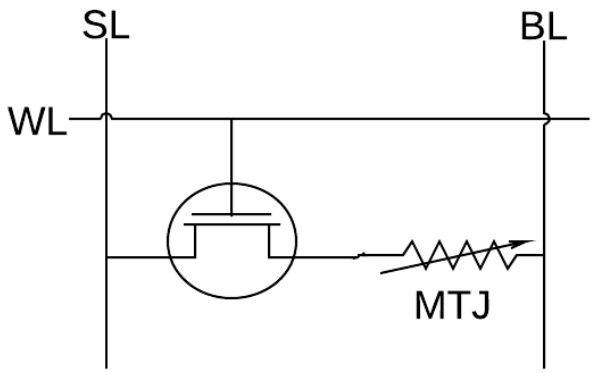

Figure 2.2: 1T-MTJ Bit Cell

sense amplifiers, these current values can be measured using reference signals and data is made available at the output. Also, since the voltage applied for read operations is much smaller than that for writing, read operations do not flip the bit stored inside the cell thus avoiding any destructive reads.

For the write operation, a substantially larger voltage has to be applied between SL and BL. Typically, to write a '0' a voltage of positive 1.0 V is applied between SL and BL whereas a negative 1.0 V is applied between the same to write a '1' [6] [8]. The duration of this write pulse is longer than that in the case of a read operation. This is since a large current is required to switch the direction of a free layer of the MTJ. Usually ,the size of the MTJ determines the amount of switching current which further determines the write pulse width.

2.1.3 Write Latency and Write Pulse Width Trade-off

As mentioned earlier, there exists a dependency between the write pulse width and the switching current through the MTJ. Figure 2.3 [4] gives the relationship between the two. The work in [11] categorizes the relation into three different regions of operations based on the pulse width. For Pulse width T $>$ 20 ns, the switching region is in thermal activation , for pulse width T $<$ 3ns, the region is in precessional switching, whereas the switching is in dynamic reversal for pulse width 3ns $<$ T $<$ 20ns. In

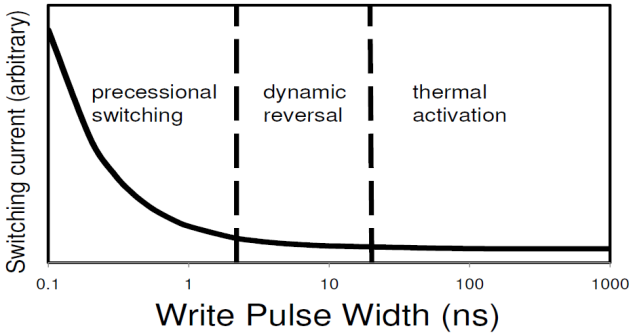

Figure 2.3: Switching Current vs. Write Pulse Width [4]

thermal activation, the switching current barely increases even with a large increase in the write pulse width. In this region a smaller pulse width can be used for better performance without any changes in the switching energy. As we move to the dynamic reversal region, the switching energy increase with reduction in write pulse width. In the precessional switching region, the switching current increases rather exponentially with even small reduction in the write pulse width. Since, the switching energy is a product of pulse width and the switching current, there would lie an optimum pulse width which would give a minimum energy with some performance degradation [4]. In our work, we concentrate the switching operation in the dynamic reversal region with pulse width varying from 3ns to 7ns.

2.1.4 Switching Current and Read Latency trade-off

For SRAM based memory arrays, read and writes are symmetrical in terms of their access latency. Usually these latencies are dominated by the H-tree routing over the macros, the predecoder and the decoder logics and the word line and bit line delays. In addition, there is Sense Amplifier latency in the read path, making the reads a bit longer than the writes, although both read and write access complete within same number of CPU cycles. For STT MRAM based arrays, inherently long write pulse width adds to the write access latency, thus leading to Asymmetric writes. As discussed earlier, the switching pulse width would be between 3ns to 7ns which is multiple times longer than the write path access latency. Thus, the write pulse latency dominates the write operations in an STT MRAM array. The read latency, on the other hand, remains

dominated by the routing delays and decoder logic and is same as that for a SRAM. [12] discusses a trade-off that allows to leverage asymmetric writes in STT MRAM. In essence, the write pulse width depends on the switching current flowing during the write operation. The switching current in turn decides the width of the NMOS access transistor, i.e. to pass higher switching current to MTJ the NMOS transistor need to be sized larger. Thus, by using a wider NMOS, a large current can be passed thus shortening the pulse width resulting in faster writes. But since NMOS is larger, the additional loading on the word lines and bit lines leads to slower read access latencies. Further, larger NMOS will reduce the density of the STT MRAM bit cell and increase the area footprint of the entire array. This means, H-tree routing will be comparatively longer which adds towards the read access latency. On the other hand, allowing for a longer pulse width leads to a smaller NMOS transistor, which further leads to less loading as well as a dense array thus making the read access faster. Thus, longer writes can be leveraged versus faster reads by choosing an appropriate NMOS transistor width. [12] further categorizes the workloads as favorable for either faster reads or faster writes for the STT MRAM based L2. They also introduce a dual-write speed scheme which classifies blocks for faster or slower writes and thus are sized accordingly. Such scheme improves the average write latency using smaller access transistors, thus saving area and also dynamic energy.

2.2 Sensitivity Analysis

While developing new processor architectures it is often required to explore the design space for new technologies or mechanisms. Simulators play an important role in exploring the design space due to its flexibility, cost and time. It is also imperative to properly select processor parameters to get simulation results so that new technology or mechanism is evaluated [1]. Also, we should identify important memory and processor parameters that dictate performance more than others, i.e. there should be a methodology for analyzing the sensitivity of the parameters towards performance.

There are various statistical techniques that can be applied for such analysis, broadly discussed in [13]. The most prominently used is the ANOVA technique. It is widely used to get the effect of each individual parameters as well as effect due to an interaction

between multiple parameters. ANOVA allows multiple values or types for parameters as inputs. The output is a percentage vector of all the parameters and their interactions. A higher percentage corresponding to a parameter or an interaction will indicate higher effectiveness towards the result, whereas a lower value indicates less impact. A more popular version of ANOVA is 2^N design, where N is the number of parameters. In this technique, a parameter can exist only in two values, usually in extremes. This is a faster way of determining effectiveness by using extreme bounds of parameters requiring less number of simulations. Although, ANOVA provides the effect of interactions along with effect of individual parameters, it is a 'one at a time' technique; it requires 2^N simulation for N different parameters. Such analysis requires a very long run time, for e.g. N =20 we will need 2097152 simulations. Hence, in order to cover a large number of parameters which is possible in case for processors, ANOVA seems less feasible given the large run time.

2.2.1 Plackett and Burman Design

Considering the run time, it is feasible to use saturated design techniques that would consume feasible simulation time. One such technique is the Plackett and Burman design [14]. The Plackett and Burman design requires N+1 simulations for N parameters, thus reducing the number of simulations to O (N). On the down side, the Plackett and Burman design only highlights effect of individual parameters ignoring effect due to the interaction of various other parameters. However, as analyzed in [1], effect on performance due to individual processor parameters is much more prominent that the effect due to interactions of parameters. Hence, Plackett and Burman provides to be a good statistical technique for our analysis.

Plackett and Burman design requires simulations that are in multiples of 4. Thus N parameters will need X simulations such that X is a next multiple of 4 greater than or equal to N+1. The configuration values of parameters for each such simulation is given by the Plackett and Burman design matrix.In the matrix, the rows represent different parameter configurations in the design whereas the columns represent the value for configurations. If in case, the number of parameters is such that there are more columns than parameters,then the extra columns are treated as dummy columns and do not affect the end results [1]. An illustration of such a matrix is shown in Table 2.1. The first row

of the design matrix is given by [14], whereas the next X-2 rows are obtained by circular right shift operations on preceding row. The last row of the design matrix is a row of minus ones as highlighted in Table 2.1. This example uses 7 parameters, thus needing 7+1=8 simulations which is also a multiple of 4. Each row in the matrix represents a simulation where '+1' and '-1' are high and low values of configurations. Here a high value refers to a parameter value that is higher than the higher end of the normal range values, whereas a low value refers to a parameter value that would be lower than the lower end of the normal range of values. For e.g. for a Reorder Buffer which usually has entries in the range of 32 to 160, we can select our low value as 16 and high value as 192. Thus, by varying values between these two extremes across simulations the Placket and Burman design gives the overall effect of that individual parameter. Since, high and low are the only values that can be input into this design, it is imperative to use a pragmatic high and low value and also observe the results taking these values into account.

A	**B**	**C**	**D**	**E**	**F**	**G**	**Execution Time**
+1	+1	+1	-1	+1	-1	-1	9
-1	+1	+1	+1	-1	+1	-1	11
-1	-1	+1	+1	+1	-1	+1	2
+1	-1	-1	+1	+1	+1	-1	1
-1	+1	-1	-1	+1	+1	+1	9
+1	-1	+1	-1	-1	+1	+1	74
+1	+1	-1	+1	-1	-1	+1	7
-1	**-1**	**-1**	**-1**	**-1**	**-1**	**-1**	4
65	**-45**	**84**	**-75**	**-75**	**73**	**67**	

Table 2.1: Plackett and Burman design matrix (X=8) [1]

As shown in Table 2.1, for each set of configuration a result effect is observed. For performance based analysis this result can be IPC or the Execution time. After obtaining the result for each configuration row, the effect of each parameter can be obtained by adding the product of result and value across all configurations. For example, in Table 2.1 the effect of parameter A is given by Effect (A) = (1)*9+(-1)*11+(-1)*2+(1)*1+(-1)*9+ (1)*74+(1)*7+(-1)*4 = 65

Thus, by computing effects of all parameters we can rank parameters based on their effect. Here the absolute value of the effect is considered without giving importance to

the sign. In our example effect of C is 84 followed by both D and E. The parameter C is the most important parameter followed by D and E. This means performance sensitive towards the value of C than others making it a critical bottleneck.

An improvement towards Plackett and Burman design is the foldover Plackett and Burman design. Like PB design the foldover design also exists in multiples of 4, but require 2X simulations instead of X, where X is the next multiple of 4 greater than or equal to N+1. Table 2.2 [1] shows an example of foldover PB design. Similar to PB, effects are computed for each parameter by adding the product of result across each configuration. The benefit of PB foldover design though is that the effects are more distinguishable and hence their ranking more prominent.

A	**B**	**C**	**D**	**E**	**F**	**G**	**Execution Time**
+1	+1	+1	-1	+1	-1	-1	9
-1	+1	+1	+1	-1	+1	-1	11
-1	-1	+1	+1	+1	-1	+1	2
+1	-1	-1	+1	+1	+1	-1	1
-1	+1	-1	-1	+1	+1	+1	9
+1	-1	+1	-1	-1	+1	+1	74
+1	+1	-1	+1	-1	-1	+1	7
-1	**-1**	**-1**	**-1**	**-1**	**-1**	**-1**	4
-1	-1	-1	+1	-1	+1	+1	17
+1	-1	-1	-1	+1	-1	+1	76
+1	+1	-1	-1	-1	+1	-1	6
-1	+1	+1	-1	-1	-1	+1	31
+1	-1	+1	+1	-1	-1	-1	19
-1	+1	-1	+1	+1	-1	-1	33
-1	-1	+1	-1	+1	+1	-1	6
+1	**+1**	**+1**	**+1**	**+1**	**+1**	**+1**	112
191	**19**	**111**	**-13**	**79**	**55**	**239**	

Table 2.2: Plackett and Burman design matrix with foldover(X=8) [1]

2.3 Optimal Processor Design Configuration

Plackett and Burman technique helps in identifying key processor parameters and also in understanding how ranks (and hence impact) of these key parameters migrate with architectural changes [1]. However, there is a need to define an optimal set of values

for these parameters for a simulator so that simulation results are not affected by improper values set for these key parameters. Also, with a new technology, it may be necessary to see how these optimal values migrate for key parameters [15]. Further, for or a mix of values, it is interesting to find the optimal set of parameters for minimal energy, delay or area. Various optimization techniques can provide an optimal solution for a discrete set of parameter values. A straight forward method is a one-at-a-time optimization algorithm [15]. In this technique, an optimal configuration is assumed initially and then parameters values are varied one at a time in steps until the optimum value for all the parameters is realized. Initially, first parameter is varied, keeping others constant till its optimal value is obtained, followed by varying the next parameter keeping the first parameter constant at the new optimal value and so on. Though this algorithm eventually converges to an optimal set and also avoids local minima, there are few drawbacks associated with it. It involves an experimenter's bias in deciding an initial optimal parameter set. The experimenter also decides the one-at-a-time order in which parameters are varied which may mislead the solution. But the most important drawback is that it allows only one parameter to be varied at a time and that it requires a substantial number of iterations to finally converge to a solution. Thus, optimization algorithm that is more heuristic so that it eliminates bias and also random would provide for a better alternative. Simulated Annealing (SA) is one such random heuristic optimization method discussed in the next section.

2.3.1 Simulated Annealing

SA is a global optimum search method which is iterative and random in nature and can be applied to a large discrete search space [16]. SA is metaheuristic in search; though it allows search in the direction where values give a better solution, it also accepts the search direction where the solution might be worse. A better or worse solution is defined by the cost of the objective solution being solved, compared to the cost obtained from a previous iteration. The decision to accept values that give a worse solution is based on a certain acceptance probability. This acceptance probability reduces with each search iteration, thus further reducing the chances of searching towards a worse solution. A conceptual parameter, Temperature, is defined in the SA method such that as the temperature varies gradually from initially being hot to being frozen towards the

end, the acceptance probability of a bad search path reduces. Thus, the temperature accompanies the probability of acceptance [17]. This allows the search optimization to avoid buckets of local minima and help converging towards a solution that point more towards a global minimum. This iterative process, though slow in converging, provides a more efficient solution than a typical brute-force method.

2.3.2 Simulated Annealing Algorithm

Simulated Annealing Algorithm mimics the metallurgical process of annealing in which controlled cooling is applied to materials. By slow, controlled cooling, the method allows for defects to eventually reach optimum point. It is an adaptation and an addition to the Metropolis algorithm.
The basic annealing algorithm for optimizing a set of discrete values for minimum cost can be stated in following steps:

I Select a random initial set of parameters S. The cost associated is cost(S). Initialize the Temperature to a 'hot' value T. Go to Step II.

II Make a search move by selecting a neighboring parameter set S' randomly. Go to III.

III Find the cost difference 'delta' between the new neighboring set and the current set of parameters i.e. $Delta = cost(S') - cost(S)$.
If Delta is negative, then the new set is a good search. In this case, select S' as the current optimum. S = S' and go to step 5. If the Delta is positive, then it is a bad search. In this case go to step IV.

IV Find the acceptance probability as $p = exp(-delta/T)$.
This search should be accepted if p ¡ uniform rand.number(0,1). If true, then S = S' and go to V. If not, then the search move is unacceptable. In this case, select a

new random configuration set S. Go to V.

V **Step 5**: Update temperature T . The temperature is reduced based on the annealing schedule. Repeat II until the temperature T reaches the freezing point.

Here the annealing schedule determines the cooling schedule that updates the temperature. This schedule is selected such that convergence is reached without too many iterations. The slower the cooling schedule means more closer is the result of the global optimum, but also means more number of iterations. Hence, cooling schedule has to be selected considering this trade-off. As T is updated chances of solutions with a higher delta being selected becomes less and less. Hence a bad move can be accepted only at a higher temperature and its chances of being accepted become less probable as the temperature is cooled down thus converging towards an optimal solution. Thus, SA provides a heuristic approach towards finding optimal set of processor parameters for a given mix of values. Although the solution may not always be the global optimum, it would at least be a neighbor of a global optimum.

Chapter 3

STT MRAM Cache Memory Modelling

STT-MRAM requires a similar sub-array interface structure as that of SRAM as similar word lines and bit lines are needed for selection. Further, a larger STT MRAM array is divided into set of sub-banks which are further divided into sub-arrays and thus have the same organization as that of an SRAM array. Thus, one can use a SRAM based array model for STT MRAM purposes. CACTI 6.5 [18] is a widely used modeling tool used to model SRAM array. We modify CACTI 6.5 to derive timing, energy and area information for a STT MRAM based array. Specifically, we modify CACTI 6.5 to consider i) zero standby leakage power for the bit cell ii) Bit cell area and aspect ratio for 1T-MTJ. iii) Access transistor sizing and its loading on the word and bit lines. iv) Effect of reduced bit cell area on the H-tree global routing. Further, we model both the data as well as the tag array as STT MRAM. The rational behind this that since writing of tag address bits and data array word happen in parallel, there is no additional delay penalty for writing tags. Another valid reason being that for higher associativity and larger caches, tag array forms a significant portion of overall leakage and area which can be potentially reduced by using STT MRAM.

3.1 Modelling STT MRAM bit cell

We use the bit cell values presented in [2] to model 1T-MTJ bit cell. Based on PTM models for 32 nm node, [2] derives values for cell size, switching current, switching time and write energy/ bit cell and these values are assumed for the rest of the work. Figure 3.1 [2] shows variation in switching time as a function of the cell area. As the switching time reduces, larger access transistors are required to accommodate the exponential increase in switching current. Hence, we see a steep increase in area for a change in switching time from 7 ns to 2 ns. For a pulse width approximately equal to 7 ns, the values are given in Table 3.1 [2].

Though, 7 ns pulse width may be appropriate for an L2 cache, it is an expensive option for an L1 cache given high rates of the CPU store requests. Thus, we model two versions of STT MRAM devices, Tech1 and Tech2, that have a write pulse latency of 3 ns and 7 ns respectively. Tech1 demands a larger NMOS access transistor and has a bit cell area of $30F^2$ [2]. Also, energy for both Tech1 and Tech2 assumed to be approximately same since a 2X increase in switching current will cause a 2X reduction in switching time for a conservatively assumed JC0 [2]. Further, based on the relation stated in [19] we scale the access transistors for Tech1 and Tech 2 based on the bit cell area. Though this is an approximation, it is sufficient to show a trend for timing and energy of these two device types.

In the rest of the section , we discuss the impact of STT MRAM on access latency, dynamic energy, leakage and area for 32nm node using CACTI and compare them with SRAM based arrays.

Parameter	**Value**
Cell Size	$10F^2$
Switching Current	50 uA
Switching Time	7 ns
Write Energy	0.3pJ/bit

Table 3.1: STT MRAM parameters for 32nm node [2]

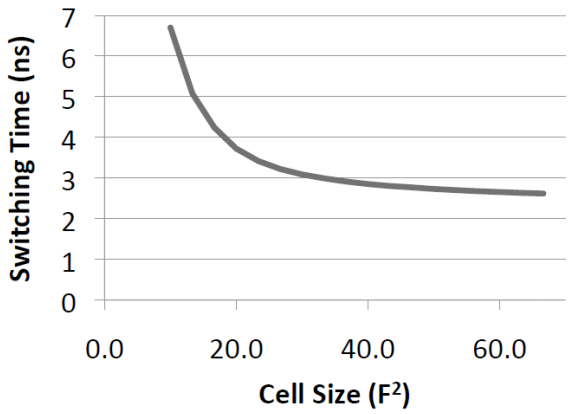

Figure 3.1: 1T-MTJ cell switching time as a function of cell area for 32nm node [2]

3.2 Latency Model

The read access latency for an array modeled using CACTI is based on the following:

1. H-tree input
2. Decoder + word line delay
3. Bit-line delay
4. Sense Amplifier delay
5. H-tree output delay

Using the device values discussed in the previous section, we use the modified CACTI to get the read access latencies for STT MRAM based arrays. Since a very small voltage (0.1V to 0.3V) is applied between the bit line and the sense line during the read operation ,the bit line delay is much higher for STT MRAM sub-arrays. Also,since STT MRAM has a single ended bit line, sensing happens using a reference signal to create a differential voltage and thus takes longer. Considering this, we add a 3X delay overhead to the sub-array bit line sensing delays based on the work in [20]. The tables Table 3.2, Table 3.3 and Table 3.4 shows the read access latencies of SRAM and STT MRAM for various sizes of L1 cache.

	32KB SRAM	**64KB SRAM**
Read Latency	0.502 ns	0.570 ns

Table 3.2: Read Acess Latency for SRAM based L1 Cache

	32KB Tech1	**64KB Tech1**	**128KB Tech1**	**256KB Tech1**
Read Latency	0.687 ns	0.821 ns	1.012 ns	1.311 ns

Table 3.3: Read Acess Latency for STT MRAM Tech1 based L1 Cache

The read latency for Tech1 is slightly higher than that of SRAM for 32 KB and 64 KB cache arrays, and increases even more with higher cache sizes. Thus replacing SRAM L1 with higher capacity Tech1 STT MRAM will lead to an additional cycle for the critical read latency despite density advantages. Tech2 STT MRAM has smaller bit cells and access transistor widths and thus shows faster reads even for higher capacity caches. Overall, we see approximately 1.5X-2X improvement in the read latencies for Tech2 compared to Tech1. This gain in read latency comes with a much higher write latency penalty of 7ns ,thus making it less attractive especially for write intensive workloads. The write latencies for Tech1 and Tech2 are shown Table 3.5. The long write pulse for Tech2 makes write operations skewed with 20X times longer latencies than reads. Tech1 writes are less skewed with write taking approximately 5X longer latencies than reads and this gap reduces for higher cache sizes.

3.3 Dynamic Energy Model

Dynamic read access energy consists of energy consumed in data array peripherals and tag access comparators but is highly dominated by H-tree input and output routing. The dynamic bit cell write energy is an additional overhead for STT MRAM arrays along with long write latency. We estimate the total dynamic write energy as a summation of peripheral write access energy obtained form CACTI and energy per bit cell from [2]. Further, write energy can be categorized as 'word write energy' while writing a 8 Byte word and 'fill write energy' while writing a entire cache line assumed to be 64 Bytes. SRAM has a negligible bit cell write energy which means that word write energy is approximately same as line fill energy obtained from CACTI. The Table 3.6 shows the CACTI based values for read and write dynamic energy for STT MRAM L1 cache and

	32KB Tech2	64KB Tech2	128KB Tech2	256KB Tech2
Read Latency	0.399 ns	0.399 ns	0.488 ns	0.627 ns

Table 3.4: Read Acess Latency for STT MRAM Tech2 based L1 Cache

Write Latency	32KB	64KB	128KB	256KB
Tech1	3.687 ns	3.821 ns	4.012 ns	4.311 ns
Tech2	7.399 ns	7.399 ns	7.488 ns	7.627 ns

Table 3.5: Write Latency for Tech1 and Tech2

its comparison with SRAM.

	Read Access Energy	Word Write Energy	Line Fill Energy
32KB SRAM	0.043 nJ	0.042 nJ	0.042 nJ
64KB SRAM	0.050 nJ	0.054 nJ	0.054 nJ
32KB Tech1	0.03 nJ	0.053 nJ	0.188 nJ
64KB Tech1	0.039 nJ	0.069 nJ	0.204 nJ
128KB Tech1	0.059 nJ	0.094 nJ	0.229 nJ
256KB Tech1	0.098 nJ	0.152 nJ	0.287 nJ
32KB Tech2	0.018 nJ	0.042 nJ	0.176 nJ
64KB Tech2	0.023 nJ	0.055 nJ	0.190 nJ
128KB Tech2	0.035 nJ	0.059 nJ	0.194 nJ
256KB Tech2	0.0530 nJ	0.0858 nJ	0.220 nJ

Table 3.6: Dynamic read and write energy for SRAM, Tech1 and Tech2 for 32nm

Since read energy is dominated by H-tree global routing, a dense array layout would reduce the dynamic read energy. For 32 KB and 64 KB, Tech 1 consumes 20% less energy for reads compared to traditional SRAM. But if a 32 KB SRAM has to be replaced with a 128 KB Tech1, it will lead to approximately 40% higher read energy. Tech 2, whereas, shows much lower read energy due to the more dense array. A 128 KB Tech2 consumes 18% less energy for reads compared to 32 KB SRAM. For smaller array sizes, the energy savings can be even more.

The write energies get expensive with STT MRAM. The word write energy for Tech1 increases by approximately 30% due to the bit cell energy overhead even though there are saving in the access energy during the writes. The increase is even greater in case of line fills where the write energy shoots up by 4X making them critical for energy consumption. Tech2 also hints at approximately 4X increase in line fills energy although word write energy only shows a small increase. Thus, STT MRAM comes with a huge

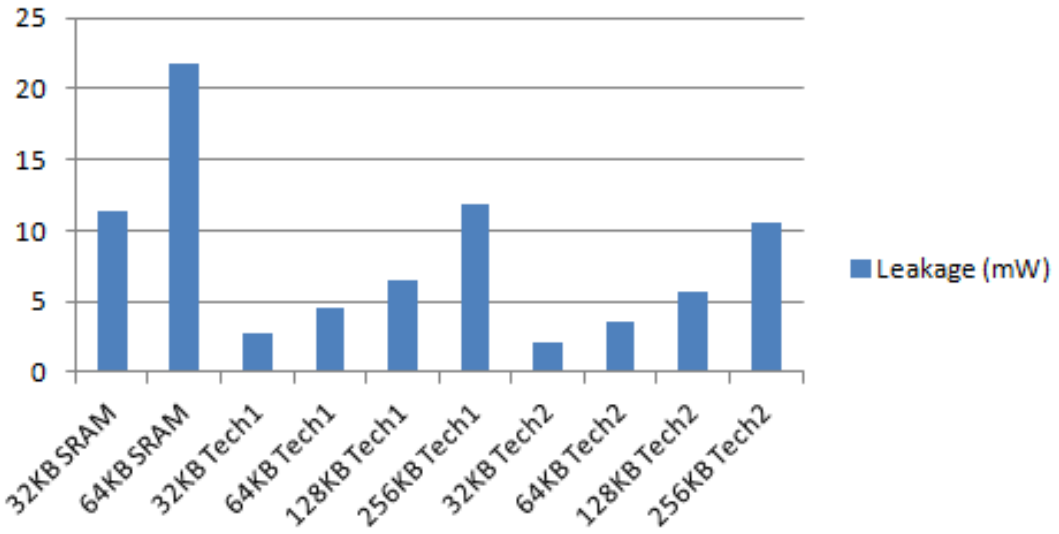

Figure 3.2: Leakage Power Savings with STT MTRAM L1 Caches at 32nm

write dynamic energy overhead although there can be savings in dynamic read energy if smaller capacity cache is preferred.

3.4 Leakage Model

The leakage power for STT MRAM based arrays is dominated by the CMOS peripheral leakage since STT MRAM bit cell is non-volatile and consumes negligible stand by power. In case of tag array, lower cache sizes has small contribution towards leakage but becomes more substantial as we move towards larger L2 caches. Figure 3.2 shows significant leakage savings that comes with STT MRAM. For 32 KB and 64 KB sizes, Tech1 and Tech2 show 4X and 5X savings in leakage respectively compared to SRAM. Leakage of 128 KB Tech1 and Tech2 is approximately 2X less than 32 KB SRAM and shows potential to replace high capacity dense STT MRAM caches with SRAM. It is also seen from Figure 3.2 that leakage savings from Tech2 are approximately 20% more than Tech1 pointing towards savings in global routing leakage.

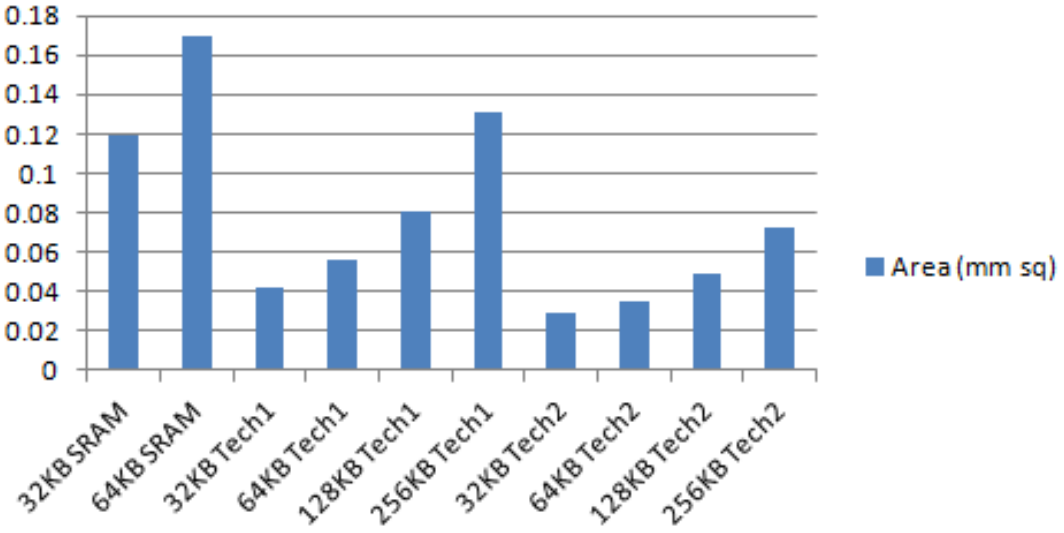

Figure 3.3: Area density improvements with STT MRAM at 32nm

3.5 Density Model

Along with leakage, density of the bit cells is an attractive feature for STT MRAM that can be leveraged upon. A STT MRAM bit cell has a much lower footprint than the traditional 6T-bit cell SRAM. CACTI assumes a SRAM bit cell of area $146F^2$ where F is the feature size. Tech1 and Tech2 assumes a bit cell of area $30F^2$ and $10F^2$ respectively. It means the bit cell density is approximately of the order 5X and 15X for Tech1 and Tech2 respectively. Although the bit cell is high,the peripheral circuitry being CMOS cuts into the overall density gains for the entire array. Figure 3.3 points the overall density gains for entire cache including the tag arrays. Tech1 gives 3X density gains whereas Tech2 provides a more substantial 4X-5X gains. We can either leverage this potential by replacing SRAM with a large capacity STT MRAM or using similar capacity STT MRAM cache and reduce memory footprint and thus die cost.

3.6 Modeling L2 Cache for STT MRAM

As seen in the earlier sections, STT MRAM arrays provide a substantial savings in leakage and area. Typically, an L2 cache occupy a larger memory footprint, especially

for CMP's where L2 is shared between multiple cores, and thus consume high standby power. We model STT MRAM based L2 cache as Tech2 in our work as it provides more area and leakage gains than Tech1. Also, taking into account the long interconnect latency between L1 and L2 and also long L2 access latency, the impact of long write latency incurred with Tech2 is subsided. Table 3.7 gives comparison between SRAM and Tech2 for L2 cache at 32nm. Leakage savings with STT MRAM are approximately 12X for 512 KB and 1 MB cache. Leakage is dominated by long global routes and thus increases as we move to 4 MB L2. Tag array leakage also forms a substantial portion of standby power as capacity goes up. Similar gains are seen with density for STT MRAM. Read energy is dominated by global routing and reduces as the array shrinks. The write energy for L2 line fills goes up by 2X compared to SRAM. This dynamic write energy penalty, though expensive, is less than the 4X increase for L1 cache. Also by compromising write pulse latency with Tech2, faster read accesses are achieved.

	Read Energy	**Write Energy**	**Area**	**Leakage**	**Delay**
512KB SRAM	0.123 nJ	0.133 nJ	1.890 mm^2	167.09 mW	1.888 ns
1MB SRAM	0.129 nJ	0.141 nJ	3.954 mm^2	338.45 mW	3.395 ns
512KB Tech2	0.09 nJ	0.239 nJ	0.155 mm^2	12.85 mW	1.694 ns
1MB Tech2	0.127 nJ	0.263 nJ	0.349 mm^2	28.01 mW	2.385 ns
2MB Tech2	0.136 nJ	0.270 nJ	0.908 mm^2	83.79 mW	3.164 ns
4MB Tech2	0.206 nJ	0.337 nJ	1.233 mm^2	117.69 mW	3.211 ns

Table 3.7: Comparison between SRAM and STT MRAM Tech2 L2 Cache at 32nm

Chapter 4

Experimental Setup

This chapter describes the experimental setup for conducting the Plackett and Burman sensitivity analysis and explains the simulation methodology used for running experiments and obtaining statistics.

4.1 Simulation Methodology

In this work we run PARSEC [3] benchmark on a gem5 simulator [21]. We simulate a four-processor CMP with shared memory; each core being out-of-order. Cache coherence is modeled using a 2-level MESI protocol with inclusion. The first level consists of private data and instruction L1 caches while the second level is shared between four cores. Due to long runtime of these workloads and the fact that these have to be ran iteratively multiple times, we use a prefer sampling opposed to a full run. To maintain accuracy while reducing runtime, we use the technique described in SMARTS [22]. Further, we create multiple checkpoints spaced between equal intervals using simple atomic CPU model of Gem5 simulator for each benchmark. These checkpoints are compiled from source using [23] such that simulations runs only on the ROI. This allows for running multiple simulations on non-overlapping regions of the application. We use GNU parallel [24] to run multiple simulations is parallel on different cores, thus allowing to run the entire workload in a relatively short period of time and further increasing simulator throughput. Further, for each checkpoint we collect samples by

running detailed out-of-order simulations in intervals and running simple timing in-order for pushing simulations between these detailed simulations. These are switched-CPU simulations where simulation statistics are only collected for detailed out-of-order simulations but the dynamic structures like caches are kept active during simple in-order timing simulations.These sampled results are verified to be fairly accurate compared with the full benchmark simulation runs.

The performance metrics, cache events and rates are collected from these samples of detailed runs and thus are representative of the corresponding application benchmark. We use IPC as the metric to estimate performance impact of a configuration. IPC is computed for each core by taking harmonic mean of sampled IPCs for intervals across benchmark. The system IPC is reported as the sum of IPCs across all four cores. We apply a similar computation technique to get processor and memory events from samples which are used for energy computation. Such simulation strategy seems pragmatic while analyzing performance and energy impact of various processor configurations where iterative runs are required.

4.2 Simulating STT MRAM memory

We use Ruby memory of Gem5 [21] simulator to model the memory sub-system. Ruby provides a detailed simulation model for cache hierarchies including a detailed interconnection network, cache policies and coherence protocols. We model a two level cache hierarchy based on a MESI coherence protocol with strict inclusion. Since Ruby does not model separate read and write latencies, it thus cannot directly model the asymmetric write operation for STT MRAM. We modify the Gem5 simulator to explicitly take into account the additional write latency for cache write events such as CPU stores and line fills for L1 and L2 caches. While this gives the impact of the long writes on the CMPs performance, it also models impact on dynamic energy consumption due to various cache update events that get more expensive with write energy of STT MRAM. Cache updates in the STT MRAM hierarchy are affected by coherent data sharing between processors, replacements and inclusion policies and thus we gather and analyze events like load and store misses, evictions and coherence transfers.

4.3 Simulation, Benchmarks and Parameters

As described in Chapter 2, we use Plackett and Burman (PB) designs to analyze the impact of processor and memory parameters so as to select parameter values for simulations. In doing so we identify important processor core and memory parameters which are bottleneck to performance. Further, PB provides designers insight into the impact the STT MRAM memory will have on these important parameters. As discussed in the earlier sections,we model four core CMP using gem5 O3CPU ALPHA simulator which is loosely based on Alpha 21264 [25]. The main memory is held at 1 GB across all simulations and the processor clock is 2 GHz across all cores. As stated earlier, PARSEC suite is used which characterizes a parallel workload. The list of PARSEC benchmarks [3] is provided in Table 4.1. Of the sim-small, sim-medium and sim-large input set available with parsec, sim-medium input set is used assuming it to be substantial workload and also considering the simulation time.

Benchmark	**Description**	**Problem Size**
blackscholes	calculates portfolio price using Black-Scholes PDE	16,384 options
bodytrack	computer vision, tracks 3D pose of human body	4 frames, 2,000 particles
canneal	synthetic chip design, routing	200,000 netlist elements
dedup	pipelined compression kernel	31 MB
facesim	physics simulation, models a human face	372,126 tetrahedra, 1 frame
ferret	pipelined audio, image and video searches	64 image queries, 13,787 images
fluidanimate	physics simulation, animation of fluids	100,000 particles, 5 frames
freqmine data	mining application	500,000 transactions
streamcluster	kernel to solve the online clustering problem	8,192 input points
swaptions	computes portfolio prices using Monte-Carlo simulation	32 swaptions
vips	image processing, image transformations	2,336 2,336 pixels
x264	H.264 video encoder	32 frames

Table 4.1: Description of the PARSEC workload with sim medium input set that is used for this work [3]

Table 4.2 gives the list processor core parameters along with their low and high values between which configurations are varied. The parameter selection method used in this work is roughly based on [1]. The processor core parameter values are selected considering the normal range of values used in commercial processors[26][27][28] [29] in addition to the architectural configuration for ALPHA 21264[30][25] and other processors [26]its implementations considered in various works. The values listed in the table are not those necessarily found in commercial processors, but as discussed earlier are

slightly higher and lower than normal values [1]. As a caveat, the selection of these high and low ranges should be considered while analyzing the results. The Branch Predictor accuracy is varied between local and tournament predictor where the 'low' local predictor has 2K history table and a 'high' tournament predictor has a global table with 8K entries in addition to the local predictor and 8K chooser entries . We assume a constant mis-prediction penalty set by Gem5 and a constant RAS size of 16. Due the mechanics of Plackett and Burman design some processor parameters cannot be varied independently of other parameters in the design[1]. We vary the Load-Store queue and Instruction Queue as a function of ROB values. If not done so, there is a possibility of ROB being 16 entries whereas LSQ and the Issue Queue may have 64 entries. This configuration will not make sense since ROB limits the number of in-flight instructions and thus both LSQ and Issue Queue will have contain a maximum of 16 instructions. Thus such scenarios should be avoided which otherwise may lead to meaningless results. The Issue queue parameter points to the in-flight scheduled integer and floating point instructions. The physical registers parameter represents both instruction and floating physical registers each which means both integer and physical registers are varied simultaneously from 64 to 256 each. A minimum if 64 is simulator set by which we are limited to although we would prefer a smaller size as the 'low' value and this limitation should be considered while analyzing the results.

We assume a 4-way out-of-order processor and thus all the parameter values are chosen for a 4-way issue. The issue width, dispatch width and commit width are kept constant to four. Varying these widths will affect the selection for most of the processor parameters as it drastically changes the number of in-flight instructions. For example, ROB has a different low and high value for 8-way issue or a 2-way issue compared to a 4-way issue and will result in ambiguous conclusions requiring guess work. Further, we also assume a constant parameter values for Integer and Floating point ALU's. This is done for two reasons. i)We know that number of ALU's is an important parameter and depends on the issue width of the processor. We can thus fix the value of ALU knowing that it is an important parameter while investigating significance and migration of other parameters. Secondly,fixing the number of ALUs will not affect the sensitivity conclusions for other parameters. Thus results can be drawn for analysis for assumed

number of ALUs. This work presents a methodology and especially concentrates on impact on memory subsystem due to STT MRAM along with other important processor core buffer structures. Table 4.3 shows the fixed ALU values selected for a core based on utilization of a single threaded workload.

Parameter	**Low Value**	**High Value**
ROB Entries	16	192
LSQ Entries	0.25*ROB	ROB
IQ Entries	0.25*ROB	ROB
Branch Predicator Accuracy	Local	Tournament
BTB Entries	128	2048
BTB associativity	2-way	16-way
Interger & FP Physical Registers	64	256

Table 4.2: Processor core parameters with Plackett and Burman values

Functional Unit	**Count**
Int ALU	4
Int MUL/DIV	1
FPU	2
FP MUL/DIV	1

Table 4.3: Functional Unit values

Table 4.4 lists the memory subsystem parameters along with their high and low values. Since gem5 Ruby models a cache line/block size that is constant across L1D, L1I and L2 cache, we have a common cache line size parameter that represents the cache line size across the entire hierarchy. L1D and L1I cache size are varied from 16 kB to 256 KB whereas, L2 cache size is varied from 256 kB to 8 MB. L2 minimum size of 256 KB is taken considering its a shared L2 amongst four private L1D and L1I caches with forced inclusion. The cache access latencies are varied along with the cache sizes and hence there are no separate parameters representing cache access latency. The latency values are obtained from CACTI as discussed in Chapter 3. Thus high and low cache values models effect of array access latency along with the effect of cache capacity which is practically more appropriate. The replacement policy is kept constant to pseudo LRU due its efficiency over LRU. The L2 Banks parameter varies the L2 bandwidth that models the contention at the internetwork connection between L2 and private L1

caches. We model the memory bandwidth by varying number of channels that allows more bytes to be transfered.

Parameter	Low Value	High Value
L1-I size	16 kB	256 kB
L1-I associativity	direct	16-way
L1-D size	16 kB	256 kB
L1-D associativity	direct	16-way
Line Size	16 Bytes	256 Bytes
L2 size	256 kB	8 MB
L2 associativity	4-way	32-way
L2 Banks	1	8
Memory Latency	200 cycles	20 cycles
Memory Bandwidth	1-channel	8-channel
I-TLB Size	64 Entries	256 Entries
D-TLB Size	64 Entries	256 Entries

Table 4.4: Memory Subsystem parameters with Plackett and Burman values

4.4 Methodology for Energy Calculation

Plackett and Burman design can also be used to obtain set of important processor parameters which are sensitive towards energy. The design matrix described in the previous section can be applied to get the ranks for the impact on energy by varying same set of high and low values. Further, true significance of the parameter should also be determined by the energy consumed along with its performance. We use the Energy Delay Product as the cost metric and apply the PB design to find the sensitivity of the processor parameters towards EDP.

Delay is obtained from the system CPI in the manner described in the section of simulation methodology. The dynamic and leakage power is obtained from McPAT [31]. McPAT is a integrated framework for area and power modeling and models multicore Alpha 21264. Whereas leakage power is only based on the processor configuration, the dynamic power needs runtime stats like event counts, accesses and look-ups. McPAT supports an interface where simulation statistics can be provided to get dynamic power consumption of processor core parameters. We gather processor stats regarding event counts for each benchmark from gem5 and provide it to the McPAT interface to get

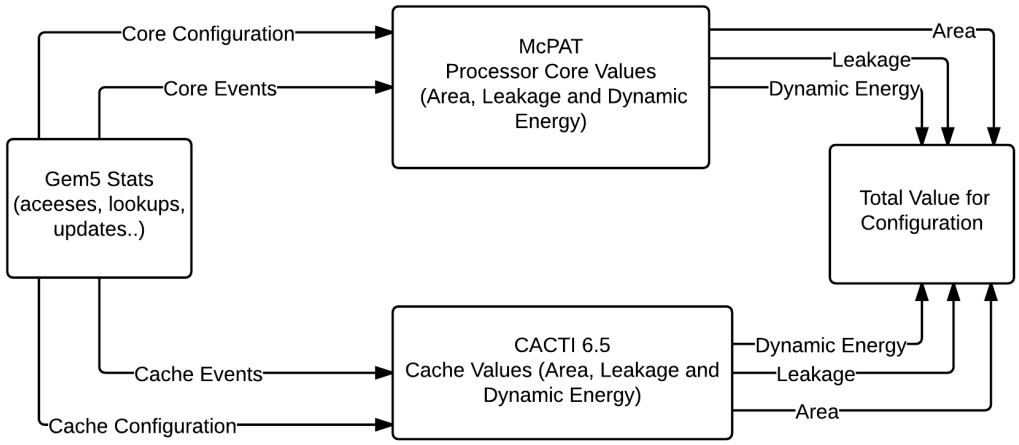

Figure 4.1: Methodology to estimate dynamic energy for a configuration along with area and leakage

dynamic power information. The dynamic energy and leakage values computed from CACTI give the energy for the cache hierarchy. Based on these statistics, we obtain EPI (Energy per Instruction) as our energy metric and compute the Energy Delay product as EPI*CPI. Figure 4.1 shows the methodology for energy calculation.

Chapter 5

Plackett and Burman Design Results and Analysis

In this chapter we analyze the results from the Plackett and Burman design experiment discussed in Chapter 4. Initially, based on the parameter selection, we determine the most important processor parameters for traditional SRAM memory. We later apply the same design experiment to determine bottleneck parameters for a STT MRAM based memory and analyze the impact of the enhancement in memory technology on the significance of the parameters. Further, considering the overall performance vs. energy overhead, we analyze the sensitivity of the processor parameters towards EDP when STT MRAM is assumed. We also classify PARSEC workload into groups based on their similarity on stressing processor parameters.

5.1 Sensitivity Analysis

In this section, we evaluate biggest performance bottlenecks for a four wide out-of-order four core CMP simulated using gem5 running PARSEC suite. We conduct analysis using traditional SRAM as the baseline memory subsystem. As discussed in Chapter 4, we consider a total of 19 processor and memory subsystem parameters (X=20) and vary them between selected 'low' and 'high' values. Further, the PB design with foldover is simulated that needs 2X simulations for better accuracy of results. Table 5.1 show results for PB with foldover using SRAM memory that are based on simulations results

Parameter	stream	vips	swap	fluid	canneal	dedup	body	ferret	face	x264	rtview	Total
Line size	1	1	4	4	3	1	2	1	2	1	1	21
LSQ Entries	2	2	6	3	2	3	4	5	1	2	2	32
L2 assoc	3	3	3	2	8	2	3	3	7	3	3	40
ROB Entries	8	7	1	1	10	4	1	2	11	6	5	56
BPred Type	4	4	11	9	5	5	8	8	4	4	4	66
Mem Latency	6	6	13	5	1	8	9	12	3	7	7	77
L1D size	5	5	12	8	9	6	7	7	9	5	6	79
L1Iassoc	9	9	8	6	13	7	6	6	13	9	8	94
BTB assoc	12	11	2	7	14	9	5	4	16	10	10	100
Mem B/W	7	8	15	11	4	10	11	13	5	8	9	101
Physical reg	13	13	7	10	15	11	10	9	17	11	11	127
I-TLB	10	10	18	13	6	14	14	16	6	12	12	131
L2 size	16	14	5	12	18	12	12	10	19	14	14	146
L1D assoc	11	12	19	15	7	15	15	18	8	13	13	146
DTLB	17	16	9	14	19	13	13	11	18	15	15	160
Issue Queue	14	15	17	17	11	18	18	19	10	16	16	171
L2 B/W	19	18	10	16	17	16	16	14	15	18	18	177
BTB size	15	17	16	19	12	19	19	17	12	17	17	180
L1Isize	18	19	14	18	16	17	17	15	14	19	19	186

Table 5.1: Plackett and Burman design results for all Processor and Memory parameters with SRAM based Memory hierarchy

for 40(2X) configurations. For each benchmark, parameters are assigned a rank based on their sensitivity towards performance; the most significant parameter having $rank = 1$. The ranks are assigned based on the total weight computed for IPC as described in Chapter 2. For each parameter, the rank is summed across all the benchmarks to get an overall summation of ranks. The sum total indicates the most and the least significant parameters across all the benchmarks. The parameters are sorted according to the ascending order of their sums.

Line size turns to be the most significant parameter closely followed by number of LSQ entries and they both remain significant across all benchmarks. The L2 cache associativity remains significant across benchmarks barring canneal and facesim and can thus be considered a significant bottleneck. The Re-order buffer as well ranks most significant across four workloads and even though its rank drops down for others, we consider it as an important bottleneck. Thus processor performance is most sensitive to these four parameters and they form the most important processor bottlenecks as shown in Table 5.1.

Secondly, we consider ranks 5 to 10 (till Memory Bandwidth) as parameters with some degree ('medium') of sensitivity but not bottlenecks given their comparably lower ranks. We limit this group till Memory Bandwidth as there is a substantial difference between

its sum and that of the next parameter (Physical Registers). The L1 Data cache size, L1 Instruction cache associativity, BTB associativity and Memory Bandwidth show some degree of significance for individual benchmarks. Parameters below Physical registers ($rank = 11$) can be declared as those with a much lower significance.

Line size as an important bottleneck may not come as a surprise since it dictates a high degree of spatial locality that can be leveraged across both levels of cache. A higher line size has a prefetching effect across most of the benchmarks that reduces miss rates. In addition, line size also affects sharing in a coherent setup where multiple cores can operate on different words in the same line. The Load Store queue also ranks as a significant micro-architectural bottleneck parameter. Most of the PARSEC benchmarks are memory bound and on average consists of approximately one-third of instructions as load and stores and hence load speculation, which leverages load store dependencies, may improve performance with higher entry LSQ.

L2 associativity is a crucial bottleneck and hints at high conflicts between lines in a set of a shared L2. The fact that L1D assoc is relatively less significant, points that L2 associativity bottleneck arises due to different processors sharing lines mapped to a set. L2 associativity ranks third in significance across all benchmarks except for canneal and facesim. These workloads have high capacity miss rates since the large program data block does not fit inside L2 irrespective of the number of ways. This fact is corroborated with L2 size being almost least significant for these benchmarks.

Although the Re-order Buffer shows extreme sensitivity towards performance for swaptions, fluidanimate, bodytrack and ferret, it is much less sensitive for streamcluster, canneal, vips and facesim which hints that ROB stalls is not as frequent for a multi-threaded workload running on four cores. Further, these workloads have high miss rates [32] which can be a mix of speculative and non-speculative misses. The speculative misses is a function of ILP's aggressiveness and leads to eviction of useful data from the cache thus leading to performance drop. Thus, out-of-order memory access for these benchmarks maybe counter productive which leads to degradation in IPC. The branch prediction accuracy is significant for these benchmarks and indicates frequent flushing of the pipeline leading to less ROB full events. Branch prediction accuracy, though overall significant, ranks much lower for some benchmarks like swaptions, fluidanimate and ferret and suggests that varying the prediction accuracy do not necessarily impact

performance and a less accurate local predictor can be sufficient.

The L2 cache size shows a pretty low sensitivity towards performance for PARSEC benchmarks. The PARSEC sim medium working set used for this analysis has a considerable working set thus application block does not fit in L2 often even for a larger capacity. Swaptions is an exception since it has a small working set of 512 KB and thus L2 size plays a significant role on performance since the data set would fit in for larger caches. Higher miss rates and a lower significance for L2 means Main memory plays an important role for PARSEC. Ferret shows similar trend as swaptions though less significant. Memory latency ranks fifth and is sensitive across most benchmarks especially canneal, streamcluster, freqmine and facesim which have a higher miss rates. It can be seen that memory latency drops in significance for swaptions and ferret since the working set fits well in L2.

L1 Data size is impacted by degradation due to access latency and gains due to the larger capacity as it is varied from low to high value. Though L1D size ranks relatively higher among the list of memory based parameters, it does not come across as a crucial bottleneck across any benchmark having the least rank of 5 for streamcluster and vips and dropping down to 12 for swaptions. This hints that although a higher capacity L1D comes with performance improvements it is offset by additional access latency reducing the significance of capacity a bit.

The L1I cache size and BTB size show least significance throughout and can be considered to be non sensitive towards performance. This means a higher capacity L1I cache or BTB may not necessarily bring improvement in performance. Although, there is a scope of interaction between parameters than PB does not consider. L1I associativity and BTB associativity figure in top ten important parameters can there is certainly some degree of interaction with L1I cache size and BTB size that may impact performance.

5.2 Impact of STT MRAM on Sensitivity

Chapter 3 outlines the effect of STT MRAM technology on latency, density and energy. The write latency and write energy overhead is been a huge concern, and to address these problems researchers have proposed several techniques such as reducing the retention time [33] [34] [4], modifying cache hierarchy by using a mix of structures with different

properties [10] [9] [35][34], implementing policies to limit write operations to high-power structures [36][9][37][38] [39], and using hybrid cache architectures [40][41]. Though only some of the research work looks into STT MRAM as an option for L1 cache with most limiting their scope to LLC. Also, most research works limit their analysis to IPC and access events changes. While these key metrics provide some insight to effect of STT MRAM on performance, computer architects need to identify other important metrics those affect overall systems performance to understand the actual impact of STT MRAM memory. Considering that it would be rather difficult to evaluate the overall impact of all metrics, we suggest the PB methodology to understand the big picture impact of STT MRAM memory. We use the PB design matrix to gather the significant parameters with STT MRAM hierarchy and by comparing their ranks with those for SRAM memory, the actual effect of the technology can be determined. Separately, we can observe how does the new memory technology differ in stressing processor parameters for each benchmark. This would provide a big picture idea to computer architects; whether there is a need to re-look certain important aspects of processor design and policies, or would traditional design choices would suffice.

We consider the same set of parameter values and ranges while simulating configurations with STT MRAM memory. We consider L2 cache with a 7 ns write pulse width (Tech2) described in Chapter 3) and L1 Data and Instruction caches with 3 ns pulse width (Tech1).The results of PB design with foldover assuming STT MRAM Memory is shown in Table 5.2.

Initial conclusions that can be drawn is that with STT MRAM memory, only first four parameters (till LSQ Entries) are significant across benchmarks and thus have high sensitivity. We base on the large difference between the sum of ranks for fourth parameter (LSQ Entries) and fifth parameter (L1 Data Cache Size). Similarly, we consider the next five parameters (L1D cache size to Main Memory latency) as a set with medium significance towards performance. On comparison with SRAM results, we find that though the order of the ranks have changed, the top four parameters remain the same and hence have the same set of most important performance bottlenecks. Further, the set of parameters with medium significance shrinks as memory bandwidth drops down in rank below physical registers. A better comparison is accomplished by verifying how much the sum of ranks have changed for the high and medium parameters

Parameter	stream	vips	swap	fluid	canneal	dedup	body	ferret	face	x264	rtview	Total
Line size	1	1	8	3	3	1	4	2	2	1	1	27
L2 assoc	3	3	4	2	7	2	2	3	6	2	2	36
ROB Entries	8	7	1	1	10	3	1	1	10	6	3	51
LSQ Entries	2	2	11	6	2	4	8	8	1	3	4	51
L1D size	5	5	12	8	9	5	9	7	8	4	5	77
BPred Type	4	4	13	9	5	6	11	10	4	5	7	78
L1I assoc	9	9	6	5	12	7	5	5	12	7	6	83
BTB assoc	12	10	2	4	14	8	3	4	14	10	9	90
Mem Latency	6	6	16	10	1	10	12	13	3	8	11	96
Physical Reg	13	11	5	7	15	9	6	6	16	11	8	107
Mem B/W	7	8	17	13	4	11	13	15	5	9	10	112
L2 size	16	14	3	11	18	12	7	9	18	14	13	135
I-TLB	10	12	19	16	6	14	17	18	7	13	15	147
D-TLB	17	15	7	12	19	13	10	11	19	15	12	150
L1D assoc	11	13	18	17	8	15	16	19	9	12	14	152
L1I size	18	19	10	15	16	17	14	14	15	18	16	172
L2 B/W	19	18	9	14	17	16	15	12	17	19	17	173
Issue Queue	14	16	15	19	11	18	18	17	11	17	19	175
BTB size	15	17	14	18	13	19	19	16	13	16	18	178

Table 5.2: Plackett and Burman design results for all Processor and Memory parameters with STT-MRAM based Memory hierarchy

compared to SRAM. Table 5.3 gives a migration in average sum of ranks across all benchmarks for parameters with high and medium sensitivity. We consider the rank migration of important parameters since they carry a higher weight and thus variation in their sum of ranks would affect their sensitivity as bottlenecks much more than for less significant parameters. Negative or positive differences indicate that the sensitivity of the parameter has increased or decreased respectively.

Further, across important parameters, we consider rank difference of 1.5 or more to be of a substantial impact. The rational behind this lies in the analysis of weight of effects for parameters across configuration. The ranking system for Plackett and Burman is based on these total effects as discussed in Chapter 2. It is observed that for parameters with medium sensitivity, their total effects are clustered together with no significant difference in values making ranks highly susceptible to simulation noise. Further, STT MRAM has a inherently less IPC than SRAM causing the effects to vary across all configurations. Thus, even though some parameters show migrations in ranks, their respective effects may not be impacted as much. Only for parameters with overall migration of 1.5 or more showed an impact on overall effects.

Table 5.3 highlights substantial migration of two parameters, LSQ and Memory

Parameter	SRAM Avg.	STT MRAM Avg.	Migration
Line size	1.90	2.45	0.54
LSQ Entries	**2.90**	**4.63**	**1.72**
L2 assoc	3.63	3.27	-0.36
ROB	5.09	4.63	-0.45
BP Type	6	7.09	1.09
Mem Latency	**7**	**8.72**	**1.72**
L1D size	7.18	7	-0.18
L1Iassoc	8.54	7.54	-1
BTB assoc	9.09	8.18	-0.90
Mem B/W	9.18	10.18	0.99

Table 5.3: Effect of STT MRAM memory on average ranks of processor parameters across all PARSEC benchmarks

Latency. Although, there is drop in rank for LSQ, it remains as sensitive for streamcluster, canneal, facesim and vips hinting that the reduction in significance is not across all workloads. It significance, as a bottleneck, reduces for others with ferret, bodytrack, fluidanimate and rtview showing reductions in sensitivity. For these workloads, memory dependence prediction like load speculation provided by a larger LSQ proves to be less effective and that out-of-order execution of loads and stores fails to achieve greater ILP. Figure 5.1 shows some reduction in IPC gain for STT MRAM when LSQ Entries are increased. Speed-up is used a metric since we are comparing the change in impact on performance not merely performance. We see loss in speed-up for bodytrack, dedup, rtview and ferret which co-relates with the increase in ranks. Also the reduction in speedup is just around 2 to 4 % and remains the same for other workloads. Thus LSQ reduces in significance but still remains a high sensitivity parameter.

Significance of main memory latency, which shows medium sensitivity for SRAM, drops on average. This increase in rank is substantial across fluidanimate and rtview and to some extent across bodytrack and dedup. For these workloads sensitivity seem to shift from medium to low. Memory latency continues to be an important bottleneck for benchmarks with high miss rates i.e. canneal, facesim, vips and facesim. Figure 5.2, which shows reductions in main memory accesses for STT MRAM compared with SRAM, confirms the trend seen in significance. Decrease of around 10 % is seen for rtview and around 5% for bodytrack and dedup. Others show marginal decrease in

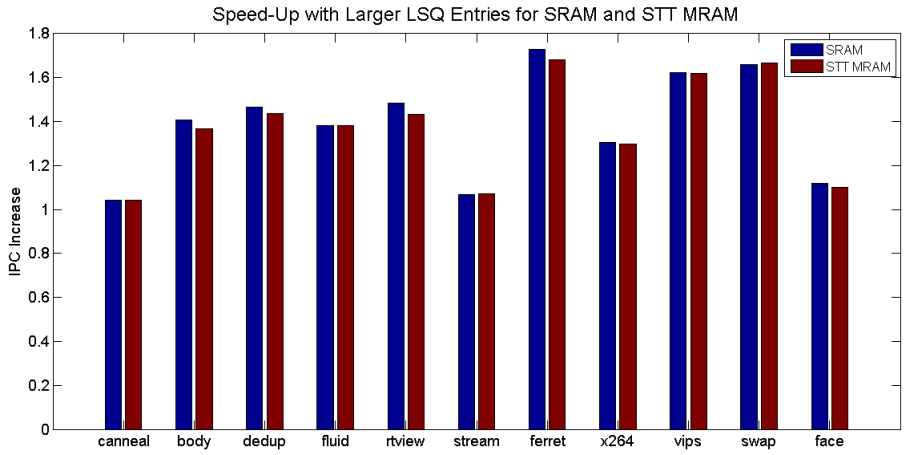

Figure 5.1: Variation in speed-up when LSQ is changed from 4 to 64 Entries for a 64 Entry ROB

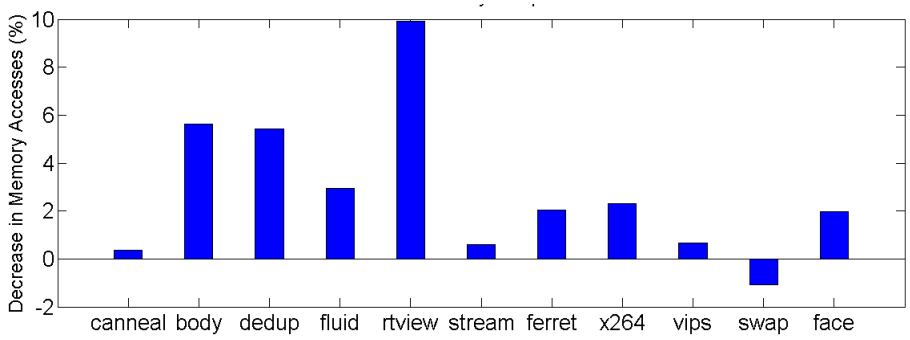

Figure 5.2: Percentage Reductions in Main Memory Requests when STT MRAM is used in cache hierarchy

memory requests. This reduction in access may also explain decrease in significance for memory bandwidth though the rank variations are really marginal. It is possible that asymmetric writes leads to different pattern of data sharing, replacements, invalidations and other cache updates and leads to less access of the main memory. Alternately, STT MRAM tries to fit the data into L2 cache more than it did in case of SRAM as hinted by substantial improvement in significance of L2 cache size across bodytrack where its rank jumps form 12 to 7. Thus, STT MRAM reduces sensitivity of memory latency towards performance for some workloads though it remains an important bottleneck for those with a higher miss rates.

Of the cache hierarchy related parameters, we find impact of cache line size and L2 associativity to be high whereas that L1D cache size and L1I cache associativity as medium. Other parameters like L2 and L1I cache size and L1D associativity to be of lesser significance. Importantly, there is no significant migration observed for these parameters with STT MRAM technology and thus the design choices with respect allocation and replacement policies, sharing patterns and protocols need not change significantly than those currently used for SRAM as far as performance is concerned.

5.3 Workload Characterization

By identifying the bottlenecks for a memory technology, it can be determined whether these system aspects have an impact on the performance for a given workload. As we observe in the previous section, since various PARSEC benchmarks stress processor and memory parameters differently, only a select few would be good candidates for certain enhancements. Workloads can be broadly categorized as memory bound or compute bound. Work in[32] use real hardware to profile PARSEC to understand bottlenecks for CMP designs. PARSEC has been classified in different ways such as by the difference in application types, Integer vs. Floating point, size of the working set, as read vs. write intensive and scalability as discussed by the work in [3][32]. Thus, based on such variety of classifications it is difficult to say if two different benchmarks are similar. We use the method described in [1] which classifies benchmarks based on the degree in which they stress processor parameters. Under this method, similarities between two benchmarks in described by comparing their Plackett and Burman design ranks. For a benchmark, ranks for all the parameters can be considered as a rank vector. Further, to determine similarities between any two benchmarks, we find the Euclidean distance between their corresponding rank vectors. For rank vector $X = [x_1, x_2, .., x_{n-1}, x_n]$ for benchmark X and rank vector $Y = [y_1, y_2, .., y_{n-1}, y_n]$ for benchmark Y, where n is the number of parameters, the euclidean distance between the X and Y is given by:

$$Dist = [(x_1 - y_1)^2 + (x_2 - y_2)^2 + ... + (x_{n-1} - y_{n-1})^2 + (x_n - y_n)^2]^{1/2} \tag{5.1}$$

The smaller the distance, the greater is the similarity although it is up to the reader to select the similarity threshold. These distances are represented in form of hierarchical

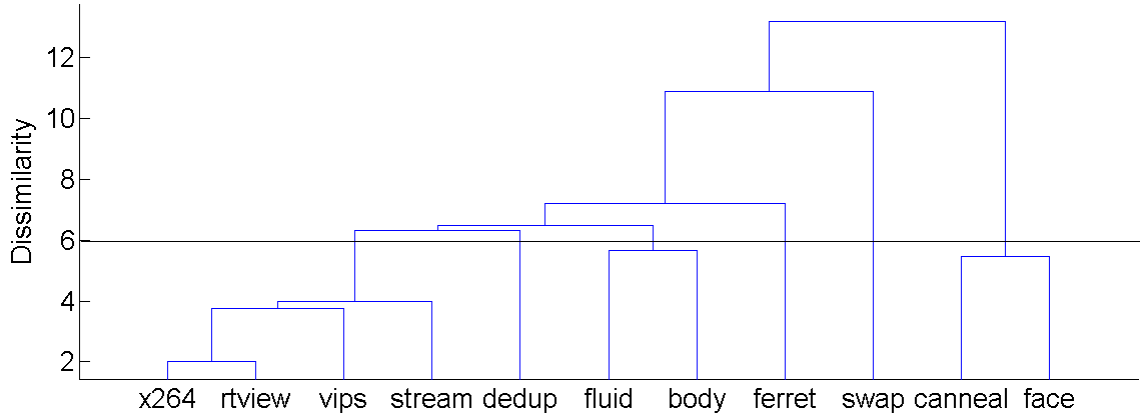

Figure 5.3: Cluster Diagram of Euclidean Distances showing Similarities and Dissimilarities between PARSEC benchmarks on stressing Processor Parameters

clusters as shown in Figure 5.3. This cluster diagram, also known as a dendrogram, uses the euclidean distances between benchmarks. Figure 5.3 gives similarities between PARSEC benchmarks based on both processor core and memory parameters when STT MRAM technology is assumed. The similarity threshold is drawn arbitrarily at 6 and is used a reference to group benchmarks. As seen, x264 and rtview has the least dissimilarity and thus together form a cluster. Further, streamcluster and vips are more similar to x264 and rtview than other benchmarks and these four form another cluster. Similarly, fuildanimate and bodytrack equally stress processor parameters and so does canneal and facesim and both form a separate cluster. Swaptions shows a large dissimilarity than other benchmarks by the way it is impacts parameters and forms a unique group.

Based on this clustering, representative groups are shown in Table 5.4. These groups aim at classifying PARSEC in a unique way as described in [1] and that benchmarks grouped together may have different IPCs or miss rates. In addition, this classification aims at providing a representative benchmark for a set of benchmarks so that design space exploration can be completed more efficiently without compromising simulation time. Instead of running the entire PARSEC benchmarks redundantly, initial simulations to evaluate design policies can be carried out using any one benchmarks from its representative group.

Group	Benchmarks
I	x264, rtview
II	streamcluster, vips, x264, rtview
III	bodytrack, fluidanimate
IV	canneal, facesim
V	dedup
VI	ferret
VII	swaptions

Table 5.4: PARSEC workloads grouped on their effect on Memory Subsystem Parameters with STT MRAM hierarchy

5.4 Sensitivity of Parameters Towards Energy Delay Product

5.4.1 Sensitivity for SRAM Memory

Just as STT MRAM affects the performance with its inherent long write latency, it also impacts the write energy with the high write current during writes. For example, L1D,L1I and L2 caches sizes, their associativity as well as their line sizes may have an additional impact on energy since they determine the miss rates, allocation and data sharing. The change in the cache update events will impact dynamic energy with a higher penalty and thus creating a case to study the impact of parameters on the dynamic energy. Thus for a complete analysis, the energy impact for STT MRAM should be considered along with its impact on performance. Further, it is an interesting exercise to see what are the bottleneck parameters for energy for a STT MRAM based processor. As discussed in Chapter 4, we use Energy Delay product as a metric to evaluate the overall significance of parameters in the design space. We ignore the contribution of leakage energy in this analysis since it may hide the impact of parameters such as associativity and block sizes which do not have a leakage overhead but do affect the dynamic energy. If standby power is considered, its impact may be misleading. Further, since we do not take off chip energy into account, only a secondary impact of main memory latency and bandwidth is taken into account i.e. manner in which main memory impact dynamic energy in the on-chip cache hierarchy.

Although we use Plackett and Burman analysis for to get total effect of a parameter

Parameter	stream	vips	swap	fluid	canneal	dedup	body	ferret	face	x264	rtview	Total
Line size	High	High	Med	Med	High	High	High	High	Low	High	Low	37
BTB assoc	Low	Med	High	High	Med	Med	High	High	Med	Med	High	40
L2 assoc	High	Med	Med	Med	Med	High	High	High	Low	High	Med	40
LSQ Entries	High	High	Med	Med	High	Med	High	Med	Low	High	Low	45
ROB Entries	Med	Med	High	High	Med	High	High	High	Low	Med	Low	45
BPred Type	High	Med	Med	Med	High	Med	Med	Med	Low	Med	High	48
L1D size	High	Med	Med	Med	Med	Med	Med	Med	Low	Med	Med	56
L1I assoc	Med	Med	Med	High	Med	Med	Med	Med	Low	Med	Med	56
D-TLB	Low	Low	High	High	Low	Med	Med	Med	High	Low	High	59
Physical Reg	Low	Med	Med	High	Med	Med	Med	Med	Med	Med	Low	61
Mem Latency	High	Med	Low	Low	High	Med	Med	Low	Med	Med	Low	67
Mem B/W	High	Med	Low	Low	High	Med	Med	Low	Low	Med	Low	72
L2 size	Low	Low	High	Med	Low	Med	Med	Med	Med	Low	Low	76
L1D assoc	Med	Med	Low	Low	Med	Low	Low	Low	Low	Low	Med	90
I-TLB	Med	Low	Low	Low	Med	Low	Low	Low	Med	Low	Med	90
L1I size	Low	Low	Med	Low	Low	Low	Low	Low	High	Low	Med	91
Issue Queue	Low	Low	Low	Low	Low	Low	Low	Low	Med	Low	High	96
L2 B/W	Low	Low	Med	Low	Low	Low	Low	Low	Med	Low	Low	100
BTB size	Low	Low	Low	Low	Low	Low	Low	Low	Low	Low	Low	110

Table 5.5: Plackett and Burman design results for all parameters showing sensitivity towards EDP with SRAM based Memory hierarchy

across configurations, we generalize the ranking system for EDP analysis. The reason being, the individual effects are observed to be much more clustered and have similar values across more number of parameters than those in case of performance analysis. This is expected since as parameters are varied form low to high value the decrease in delay is offset by increase in performance leading to multiple configurations converging around values in the same range for total effect. In order to separate important parameters from the less important ones, we group parameters as 'high', 'low' and 'medium' for each benchmark. This grouping is based on identifying jumps in the values of the effects which makes it easier to categorize. Further, to get the total sum of ranks across benchmarks, each rank group is given weight; $high = 1, medium = 5, low = 10$. Parameters are sorted based on the this total sum of ranks. Thus, the final PB results table consists of parameters ranked as 'high', 'medium' and 'low' sorted in an ascending order of calculated sum of ranks.

Table 5.5 gives the PB results for SRAM based memory showing sensitivity of parameters towards EDP. Cache Line size continues to be the most significant parameter and ranks high most benchmarks indicating a larger impact of performance than energy consumption. Also, higher line size reduces number of miss events and thus energy consumed by filling those cache lines. L2 cache associativity, re-order buffer and LSQ

are observed to be significant across most benchmarks. This means that the performance benefits of a larger number of ways or larger number of buffer entries overshadow their energy overheads for these parameters. Branch Target Buffer (BTB) associativity turns out to be a bottleneck for EDP even though its value is much lesser sensitive towards performance. BTB are expensive structures in terms of energy and its consumption increases with higher associativity which needs parallel lookups. Hence, in this case energy overheads of BTB associativity dominates its performance benefits. Further, we observe that Branch Predictor type, L1I associativity and L1D cache size have a 'medium' sensitivity towards EDP across most workloads and can be categorized as parameters of medium significance, similar as in case of performance. D-TLB and number of physical registers, which had a much lower sensitivity towards performance, show a medium significance for EDP. This is not surprising since D-TLB lookups and writes consume large energies which gets higher with larger sizes. Physical registers consists of register files and are one of the more expensive on-chip structures. Further, larger physical registers also imply more entries in the free list and related tables thus having a higher impact on energy.

We consider the secondary effects of the main memory on energy i.e. its impact on events in the cache hierarchy and not the energy cost of memory itself. Main memory latency and bandwidth remain highly significant for canneal and streamcluster given their high miss rates. It also shows medium to low significance across other benchmarks which was similar for performance. Similarly, L2 size is highly sensitive for swaptions as in case of performance. It does show medium sensitivity across fluidanimate, bodytrack, dedup ,ferret and facesim which is higher than it was for performance. This is since, access energy for lower array is substantially smaller and since larger sizes do not necessarily provide performance improvements, the EDP gets sensitive towards smaller sizes.

5.4.2 Sensitivity for STT MRAM Memory

Performance analysis for STT Memory showed that the bottlenecks remain the same for both SRAM and STT MRAM hierarchy with some drop in significance in LSQ and Memory Latency for only few workloads. The cache hierarchy parameters like line size, array size and associativity did not show much migration indicating that the additional write latency, though affects performance, does change the significance

Parameter	stream	vips	swap	fluid	canneal	dedup	body	ferret	face	x264	rtview	Total
BTB assoc	Low	Med	High	High	Med	High	High	High	Med	Med	Med	40
L2 assoc	High	High	Med	Med	Med	High	Med	High	Low	High	Med	40
ROB Entries	Med	Med	Low	Med	Med	High	High	High	Med	Med	Med	48
Line size	High	High	Low	Low	High	High	Med	High	Low	High	Low	51
BPred Type	High	High	High	Low	High	Med	Med	Med	Low	Med	Low	54
D-TLB	Low	Low	High	High	Low	Med	High	Med	High	Low	High	55
L1D size	High	Med	Med	Med	Med	Med	Med	Med	Low	Med	Med	56
Physical Reg	Low	Med	Low	High	Med	Med	Med	Med	Med	Med	High	57
L1I assoc	Med	Med	Med	Med	Med	Med	Med	Med	Low	Med	Med	60
LSQ Entries	High	High	Low	Low	High	Med	Med	Med	Low	Med	Low	63
L2 size	Low	Low	Med	Med	Low	Med	High	High	Med	Low	Med	67
Mem Latency	High	Med	Low	Low	High	Med	Low	Low	Med	Low	Low	77
Mem Bandwidth	High	Med	Low	Low	High	Med	Low	Low	Low	Low	Low	82
L1I size	Low	Low	Med	Med	Low	Low	Low	Low	High	Low	High	82
I-TLB	Medium	Med	Med	Low	Med	Low	Low	Low	Med	Low	Low	85
L1D assoc	Med	Med	Med	Low	Med	Low	Low	Low	Low	Low	Low	90
Issue Queue	Low	Low	High	Low	Low	Low	Low	Low	Med	Low	Med	91
L2 bandwidth	Low	Low	Med	Low	Low	Low	Low	Low	Med	Low	Med	95
BTB size	Low	Low	Low	Low	Low	Low	Low	Low	Low	Low	Low	110

Table 5.6: Plackett and Burman design results for all parameters showing sensitivity towards EDP with STT MRAM based Memory hierarchy

of these parameters. Though dynamic write energy impacts the overall cache energy consumption, it is interesting to see if this overhead vary the sensitivity of parameters for EDP. We perform PB analysis to get sensitivity for STT MRAM based processor parameters towards EDP using the same ranking system applied for SRAM.

Table 5.6 shows the results for STT MRAM and Table 5.7 shows the migration of ranks compared to SRAM. We find that expect for Line size and LSQ, sensitivity of the important SRAM parameters remain the same towards EDP. Cache line size, which was the most important for SRAM, drops by a significant margin; it goes from 'high' to 'medium' for bodytrack and 'medium' to 'low' for swaptions and fluidanimate. It should be noted that the significance of line size towards performance reduced only marginally for STT MRAM. A higher reduction for EDP comes from the fact that higher cache line sizes consume a much larger energy during line fills and replacement events for STT MRAM due to larger bit cell energy. This implies that for the same rate of misses, line size have a high dynamic energy overhead. Hence, even though there are performance improvements with higher line size, a higher energy penalty will be levied for line updates. This migration in rank and lower significance makes sense since most of the research work identifies lower size of line size as essential and efforts have been taken by changing line allocation policies to reduce number of cache line writes to

reduce energy. The LSQ drops in significance for EDP as it did in case of performance indicating that the performance drop dominates the product. Similar trend is observed for memory latency and bandwidth which reduce in significance.

Parameter	SRAM Avg.	STT MRAM Avg.	Migration
Line size	**3.36**	**4.63**	**1.27**
BTB assoc	3.63	3.63	0
L2 assoc	3.63	3.63	0
LSQ Entries	**4.09**	**5.72**	**1.63**
ROB Entries	4.09	4.36	0.27
BPred Type	4.36	4.90	0.54
L1D size	5.09	5.09	0
L1I assoc	5.09	5.45	0.36
D-TLB	5.36	5	-0.36
Physical Reg	5.54	5.18	-0.36
Mem Latency	6.09	7	0.90
Mem B/W	6.54	7.45	0.90
L2 size	6.90	6.09	-0.81
L1D assoc	8.18	8.18	0
ITLB	8.18	7.72	-0.45
L1I size	8.27	7.45	-0.81
Issue Queue	8.72	8.27	-0.45
L2 B/W	9.09	8.63	-0.45
BTB size	10	10	0

Table 5.7: Effect of STT MRAM memory on average ranks of processor parameters towards EDP across all PARSEC benchmarks

We observe that apart from cache line size, parameters for L1D, L1I and L2 caches do not show any major migration hinting that their impact on EDP remains the same with STT MRAM memory. L2 associativity and cache line size continues to be most important memory parameter for STT MRAM although there is some drop in the later. L1D cache size and L1I associativity show a medium degree of significance for EDP. This also means that STT MRAM does not substantially affect the sharing patterns, miss rates and invalidations; it merely delays the transactions in memory without significantly affecting the stress on architectural parameters.

To conclude, bottleneck parameters remain fair and squarely similar to those of SRAM towards both performance and EDP though some migration across few benchmarks has

been observed for LSQ and Main Memory latency. Migration for cache line size for EDP points to expensive line fills for larger sizes which offsets performance gains. With similar bottlenecks, computer architects can continue to concentrate currently implemented architectural design decisions towards designing processor with STT MRAM memory.

Chapter 6

Optimal Processor Configuration

This chapter concentrates on a methodology which aims at realizing an optimal configuration from a given design space. We first define a design space by identifying key processor parameters based on Plackett and Burman analysis. Further, we select different memory technology options for first level cache memory and add them to the design space. Finally, we apply the method of simulated annealing to find the global optimum results from the defined set of memory and processor parameters for energy, delay and area.

6.1 Optimization Procedure

For any optimization methodology, first step is to define a configuration using a set of variables which would form a cost function whose solution needs to be minimized. Optimization process is further based on assigning different values for these variables iteratively from a set of variables. Thus, the next step is defining a vector space for each variable across which optimal solution would lie. Both steps form the overall part of defining an appropriate design space. This section explains how are memory and processor parameters and their corresponding vector set is selected to define the design space for optimization.

6.1.1 Design Search Space for Cache Hierarchy

Chapter 2 discussed two different flavors of STT MRAM technology; Tech1 with a 3ns write pulse and Tech2 with a 7ns write pulse but faster reads than Tech1. Both provide area and leakage benefits over SRAM at the expense of performance degradation and increase in dynamic write energy; Tech1 and Tech2 show 5X improvement in leakage and 3X-4X improvement in density with 5X more line fill energy. Although STT MRAM based memory may not seem a good contender to replace SRAM in L1 cache based on performance as a sole metric, it can be considered as a suitable alternative for embedded applications where savings in power and area are prime and degradation in performance can be compromised. Further, higher density can be leveraged to fit in larger data into L1 thus reducing performance degradation. Alternatively, we can replace SRAM with equal capacity STT MRAM reducing the on-chip memory footprint. PB analysis results have shown that L1 Instruction cache is of low significance for both performance and EDP. Hence, having a high capacity L1I cache may not necessarily give performance improvements. L1D cache size, although showed some degree of significance, the sensitivity could be towards either higher capacity or faster access latency. Hence, a 4X large capacity L1 private cache ,made possible by replacement with STT MRAM, may not necessarily give apparent performance benefits. Based on these facts we keep the deign space for both L1D and L1I cache limited to STT MRAM with maximum 2X capacity as that of SRAM. Table 6.1 shows the design space selected for first level cache hierarchy. Since, STT MRAM Tech2 shows more density and leakage savings than Tech1, we consider a higher capacity 128 KB cache. Also, Tech2 has a faster access latency that translates to 2 CPU cycle access time which turns out to be a cycle faster than equal capacity SRAM. Further, we assume a standard 4-way associativity for both L1D and L1I cache across all configurations.

We assume L2 cache to be of STT MRAM (Tech2). Table 3.7 shows over 10X improvement in density and leakage which are reasons strong enough to replace SRAM in L2. In addition, unlike L1, L2 writes happen only on replacements and allocations on misses and hence a higher write latency should not impact performance dramatically as it does for L1. Further, the given the long interconnection latency between L1 and L2 (13 cycles in gem5), write pulse latency tends to get absorbed for L2 unlike L1. Thus, our design space for L2 cache consists only of STT MRAM memory with 7ns write

Capacity	Technology	CPU Cycle Access Time
32 KB	SRAM	3 cycles
64 KB	SRAM	3 cycles
32 KB	Tech1 (3ns)	3 cycles
64 KB	Tech1 (3ns)	3 cycles
32 KB	Tech2 (7ns)	2 cycles
64 KB	Tech2 (7ns)	2 cycles
128 KB	Tech2 (7ns)	2 cycles

Table 6.1: Design Space Vector for L1D and L1I Cache Size

pulse. Further, L2 cache size was found to be of medium to low significance towards EDP for STT MRAM which again means that leveraging density benefits with larger capacity may not lead to a lower EDP cost. Thus we limit the maximum capacity for L2 in the design space to 4 MB. In addition, L2 associativity is observed to be a significant bottleneck for both performance and EDP. Thus, we consider the interaction between the capacity and associativity, we vary the number of L2 ways for different capacities of L2 cache. Table 6.2 shows the vector defined for L2 cache. The aim behind considering L2 as a variable for optimization is to verify if performance benefits are significant for higher capacity or if settling for low footprint L2 would be a better deal for 4 core CMP that which would also give energy benefits.

Capacity	Technology	Associativity
512 KB	Tech2 (7ns)	8-way
1 MB	Tech2 (7ns)	8-way
2 MB	Tech2 (7ns)	8-way
512 KB	Tech2 (7ns)	16-way
1 MB	Tech2 (7ns)	16-way
2 MB	Tech2 (7ns)	16-way
4 MB	Tech2 (7ns)	16-way

Table 6.2: Design Space Vector for L2 Cache

6.1.2 Processor Core Parameter Selection

The vector set defined for L1D, L1I and L2 cache can be given as an input to a optimization solver giving minimum cost memory configuration. Although, the main goal

for this optimization exercise is to find the optimal cache hierarchy across technologies and capacities, other important processor parameters also need be added in the mix of variables to get the overall minimum cost configuration. The optimal values for these EDP bottlenecks should also be figured with the goal of leveraging this information for processor parameter selection for future simulations using STT MRAM designs. Hence, we define variables vector sets for important bottleneck parameters from PB results for EDP and add them to the design space matrix.

Plackett and Burman results from Table 5.6 identifies key processor parameters that are sensitive towards performance as well as for energy. We consider limited parameters in our design space since more number of parameters exponentially increase the iterations required to converge towards optimal solution and hence would consume longer simulations. Thus for brevity reasons we limit the design space only to the top bottleneck parameters from the PB results. All other parameters are set to standard values found in commercial processors. Table 6.3 shows the selected bottleneck processor parameter vectors that forms the complete matrix of values along with L1 and L2 cache vectors. We consider BTB associativity along with BTB size with the assumption of interaction between the two even though the later shows hardly any significance. LSQ Entries are varied along with ROB entries and are assumed to be half the number of ROB entries. These parameters are combined for faster convergence and also to avoid scenarios where LSQ entries are larger then ROB entries which would be meaningless. We consider local and tournament predictors with different history table entries as shown.

ROB	LSQ	BTB Entries/Ways	BPred Type	Predictor Table Entries	Line Size
16	8	256/4-way	Local	Local: 1K	16 B
32	16	512/4-way	Local	Local: 2K	32 B
48	24	1024/4-way	Tournament	Local:1K Global:2K Chooser:2K	64 B
64	32	512/8-way	Tournament	Local:1K Global:4K Chooser:4K	128 B
96	48	1024/8-way	Tournament	Local:2K Global:2K Chooser:2K	256 B
128	64	1024/16-way	Tournament	Local:2K Global:4K Chooser:4K	512 B
160	80	2048/16-way	Tournament	Local:1K Global:8K Chooser:8K	-

Table 6.3: Design Space Vectors for Processor Parameters

6.1.3 Defining Cost Function

The cost function is defined so that all three processor design aspects , i.e. delay, energy and area are considered. We obtain the energy, leakage and area values for processor

parameters from McPAT as described in Chapter 4 and the cache values for SRAM and STT MRAM (Tech1 and Tech2 both) are obtained from CACTI 6.5 and its modified version as derived in Chapter 3. We use the product of Energy, Area and $Delay^2$ as the metric which defines the cost across configurations. Delay is squared so that performance is given an higher weight. Further, since we are looking for optimal configuration for a core in a CMP, we consider energy, leakage and performance contribution of a configuration across all four cores. We give area a lower weight by only considering area cost for a single core. Since L2 is shared amongst fore cores, only a quarter of its area cost is considered. Thus a optimal processor configuration would give performance and energy benefits throughout the entire CMP at the same time reducing die area.

6.1.4 Simulated Annealing Algorithm

The design search space defined for the set of parameters is made available entirely to find the optimal solution. Since this is a multi parameter optimization, we use the simulated annealing algorithm that brings in an iterative improvement and allows acceptance of higher cost values thus reducing possibilities of getting stuck in a local minimum [42]. Chapter 2 discusses the base simulated annealing algorithm and the effect of the annealing schedule on convergence. We apply the same algorithm to search for the solution that gives a global minimum or a neighbor of a global minimum. The flow of the simulated annealing technique that is applied for searching optimal processor and memory configuration in shown in figure 6.1. The algorithm begins with selecting a randomly generated configuration from the vector space rather than an initially user provided configuration set. This is done to remove any initial guess or an experimenter's bias. Each iteration performs a move, where a neighboring configuration is selected. A move is accepted if the cost for the new neighboring configuration is lower than previous or if the difference in cost is lesser than the acceptance probability defined by the current temperature. Throughout the process, the best solution is stored and updated till termination. This solution eventually is the global minimum obtained form the annealing process. The initial temperature is set to 1.0 and is cooled by 0.04 after every iteration. The annealing schedule is selected experimentally based on observing convergence rate of different schedules towards global minimum. This schedule requires 250 iterations till the temperature turns cold where the annealing process ends. It was

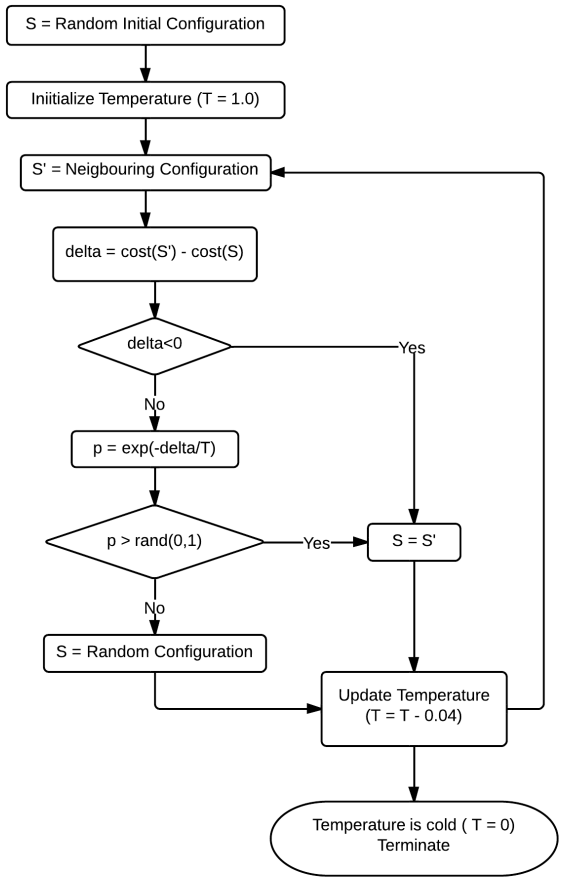

Figure 6.1: Basic Simulated Annealing Algorithm used to find Optimal Configuration Solution which gives a Global Minimum

found that 250 iterations were sufficient for the defined search space to converge towards global minimum and have enough iteration thereafter confirming that its was not a local minima trap.

The simulation methodology applied is similar to the one discussed in Chapter 4 where we gather statistics from samples running from multiple checkpoints in parallel. The difference though is we apply a simulation schedule to the number of samples gathered which goes in hand with the annealing schedule. Initially, since a larger acceptance probability allows moves even in bad direction and thus flexible, we can use less number of samples during this phase to gather simulation statistics and increment the sample

size periodically such that more accurate data is gathered towards convergence. This helps in simulation time across benchmarks without sacrificing on quality of solution.

6.2 Optimization Result

6.2.1 Minimal solution for ED^{2}AP

The optimal configuration is obtained across each workload in PARSEC suite by running the process for them individually. Table 6.4 gives the optimal set of parameters for each benchmark. This solution either a global minimum or a its neighbor.

Bench	ROB	LSQ	BTB	BPred Type	L1D cache	L1I cache	Line Size	L2 cache
body	48	24	256/4-way	Local:1K Global:2K	32 KB Tech1	32 KB Tech2	64 B	512 KB/ 8-way
canneal	48	24	256/4-way	Local:1K Global:2K	32 KB Tech2	64 KB Tech2	256 B	512 KB/ 16-way
dedup	32	16	256/4-way	Local:2K	32 KB Tech1	32 KB Tech2	256 B	512 KB/ 16-way
ferret	48	24	256/4-way	Local:1K	32 KB Tech2	32 KB Tech2	128 B	512 KB/ 16-way
fluid	32	16	512/4-way	Local:1K Global:2K	32 KB Tech1	32 KB Tech2	32 B	512 KB/ 8-way
rtview	32	16	512/8-way	Local:1K	32 KB Tech1	64 KB Tech2	64 B	512 KB/ 16-way
stream	32	16	256/4-way	Local:1K	32 KB Tech2	64 KB Tech2	512 B	512 KB/ 16-way
swap	48	24	256/4-way	Local:1K Global:2K	128 KB Tech2	32 KB Tech2	32 B	512 KB/ 16-way
x264	48	24	512/4-way	Local:1K	32 KB Tech2	32 KB Tech2	512 B	512 KB/ 8-way
vips	32	16	256/4-way	Local:1K Global:2K	32 KB Tech1	32 KB Tech2	512 B	512 KB/ 16-way
facesim	16	8	512/8-way	Local:1K Global:4K	32 KB Tech1	64 KB Tech2	512 B	512 KB/ 16-way

Table 6.4: Global Optimal Configuration giving Minimal Solution for ED^{2}A Product

As discussed, the annealing schedule was selected for appropriate convergence within limited number of runs. Figure 6.2 shows the convergence trend for one of the benchmarks. The optimal solution is observed before sufficient iterations till the temperature runs cold and hence indicating that the result is not a local minima trap.

All the workloads converges towards a minimum size L2 of 512 KB Table 6.4. As seen from PB analysis increasing L2 size does not necessary gives significant performance improvements as the data seldom fits even in a larger capacity L2 which is inclusive. These area, leakage and access energy reductions with smaller L2 points to multi-fold savings in cost. Hence, a smaller L2 points to an optimum even in a four core CMP that can reduce on-chip memory footprint and leakage savings. Also, we see solutions across most workloads settling for a 16-way associativity.

An important observation across optimal solutions is the convergence towards STT MRAM for L1 data cache. STT MRAM brings in performance degradation and also higher dynamic write energy on stores and line fills. But if runtime energy including

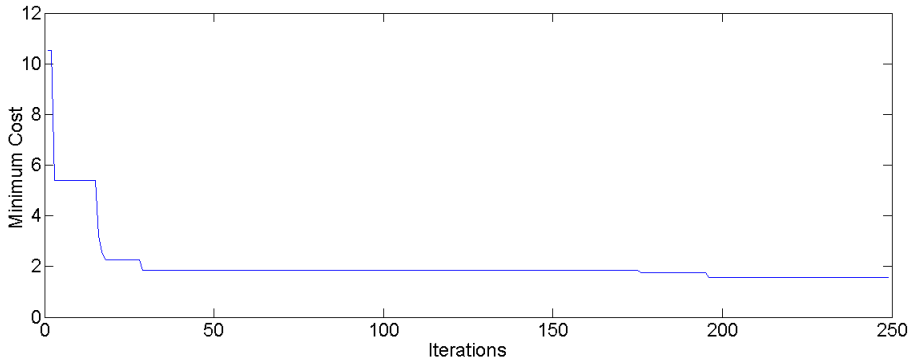

Figure 6.2: Convergence Rate towards the Optimal Solution observed for Fluidanimate for the Selected Annealing Schedule

leakage is a concern and area savings necessary, STT MRAM is seen to be a optimal solution for minimum ED^2AP. Secondly, for different available sizes, almost all results indicate a minimal 32 KB data cache as optimal for PARSEC. This confirms that leveraging density benefits by fitting in a higher capacity cache may not be a good design choice. A 64 KB capacity incurs more peripheral overhead for both access energy and leakage which seems to off-set the reduction in miss rates that is offers. An exception is swaptions which shows good performance with larger capacity L1D. As discussed in previous analysis, swaptions shows to fit more data in on-chip cache hierarchy and benefits more with larger cache than others. The most interesting observation is the convergence towards Tech2 device based STT MRAM, which has a big 14 cycles write pulse, across roughly half the number of workloads. Tech2 offers better density, lower leakage and access energy than Tech1. Importantly, it takes one cycle less to access the cache with Tech2 thus reducing the read access latency which is one of the most critical path in a processor design. Thus, for workloads that have a read/write skewed more towards reads and for those that have less percentage of store instructions, STT MRAM Tech2(7 ns) offers a better solution as the improvement in highly frequent access latency tries to balances the 14 cycle large write pulse delay at the same time providing savings in area, leakage and access energy. Table 6.5 gives the ratio for number of loads for every store instructions for each benchmark. The Load and Store percent gives the percentage that these instructions form out of overall committed instructions. Streamcluster and x264 are highly skewed towards load instructions with stores showing

real small percentage of total instructions making a faster access Tech2 a feasible option as seen from the results. On the other hand, table shows that dedup, facesim and rtview have a balanced ratio for loads vs. stores where stores form a much higher percentage of total instructions compared to others. Degradation with higher write latency of Tech2 would be way more larger for these workloads and hence they show convergence towards Tech1 as seen from the results. Thus for such applications, a larger write pulse can be traded with smaller access transistor devices. This an important observation since a good amount of research aims at reducing write pulse at the expense of bit-cell size and write current.

Benchmark	**Load/Store Ratio**	**Load Percent**	**Store Percent**
streamcluster	10.92	27.67	2.53
x264	7.06	19.8	2.8
canneal	3.74	13.31	3.55
vips	3.55	16	4.49
fluidanimate	5.46	25.96	4.75
swaptions	4.05	21.11	5.2
ferret	4.37	23.03	5.26
bodytrack	3.12	19.57	6.27
dedup	2.71	20.28	7.46
facesim	2.26	26.94	11.83
rtview	1.96	27.56	14.05

Table 6.5: Table showing frequency of CPU loads vs. stores for PARSEC benchmarks

Equally important result from this optimization procedure is preference across all benchmarks for STT MRAM Tech2(7 ns) for L1I cache. Instruction cache does not have any direct CPU writes as it is accessed only for instruction fetches. The write latency (and write energy) will act only when there are line fills on misses. But since, these from a real small fraction of total instruction cache access, the faster access provided by Tech2 gives better performance. Other savings in cost makes STT MRAM Tech2 as a strong contender to replace SRAM as L1I cache.

Cache Line size, which is important bottlenecks shows varied solutions. A higher cache line size (256 B and 512B) is obtained as a minimal cost solution for most benchmarks, showing that its performance impact is much larger than the high line fill energy for STT MRAM that grows with bigger lines. Exceptions are swaptions, bodytrack ,fluidanimate and rtview. For these, the gains in performance may not necessarily off-set dynamic energy associated with it. This confirms the findings in PB analysis for EDP where we see a drop in significance of line size for bodytrack, swaptions and fluidanimate. Thus,

optimal solution for line size is application specific in case of STT MRAM and even though most PARSEC benchamarks show preference over higher line size, there are few whose cost is impacted drastically if a high cache line is used.

Re-order Buffer and LSQ were found to be important bottlenecks in processor design. The minimal configurations obtained shows optimal ROB size to be between 32 to 48 entries. Correspondingly, LSQ should be around 16 to 24 entries for minimal cost. Thus for a four core CMP, the ability to extract ILP from threads running across each core is limited and than that increasing entries further to larger values would probably incur higher area and energy penalties without benefiting the performance much. Branch Target Buffer also proves to be expensive in terms of overall cost and most solutions points to lower entry and lesser associativity structure. The results for branch predictor are spread between local and tournament predictor though the prediction accuracy gains get limited after 2K entries of global and chooser entries of a tournament predictor.

6.2.2 Minimal Solution for Delay

Though Table 6.4 gives the optimal considering impact on delay, energy and area, it will be interesting to know what was the optimal if only performance is considered. Alternately, the trade-off between performance vs. area and energy can be understood by obtaining the best set of parameters for performance and knowing what was lost in gaining area and energy benefits for each parameter. Thus, we apply simulated annealing process for getting the global minimal set of parameters for minimum delay for each benchmark. Table 6.6 gives the values for the optimal set across individual workloads.

Bench	ROB	LSQ	BTB	BPred Type	L1D cache	L1I cache	Line Size	L2 cache
body	160	80	1024/16-way	Local:2K Global:4K	32 KB SRAM	64 KB Tech2	256 B	2 MB/ 16-way
canneal	96	48	512/4-way	Local:2K Global:2K	64 KB Tech1	32 KB Tech2	256 B	4 MB/ 16-way
dedup	128	64	1024/4-way	Local:2K Global:4K	64 KB SRAM	64 KB Tech2	512 B	1 MB/ 8-way
ferret	96	48	1024/8-way	Local:1K Global:4K	64 KB SRAM	32 KB Tech2	512 B	2 MB/ 16-way
fluid	160	80	2048/16-way	Local:2K Global:4K	32 KB SRAM	128 KB Tech2	512 B	2 MB/ 8-way
rtview	128	64	2048/16-way	Local:1K Global:2K	32 KB SRAM	128 KB Tech2	64 B	4 MB/ 16-way
stream	160	80	1024/16-way	Local:1K Global:2K	64 KB SRAM	128 KB Tech2	512 B	4 MB/ 16-way
swap	96	48	1024/4-way	Local:1K Global:4K	64 KB SRAM	32 KB Tech2	128 B	4 MB/ 16-way
x264	128	64	1024/4-way	Local:1K Global:8K	64 KB SRAM	32 KB Tech2	512 B	2 MB/ 16-way
vips	96	48	1024/4-way	Local:1K Global:8K	64 KB SRAM	64 KB Tech2	512 B	4 MB/ 16-way
face	128	64	2048/16-way	Local:2K Global:4K	64 KB SRAM	64 KB Tech2	512 B	4 MB/ 16-way

Table 6.6: Global Optimal Configuration giving Minimal Solution for Delay

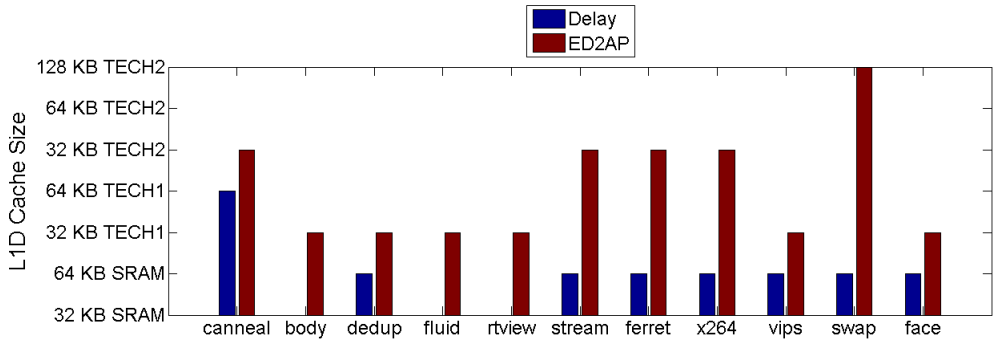

Figure 6.3: Optimal Result for the L1 Data Cache for each benchmark for Delay and ED^2AP

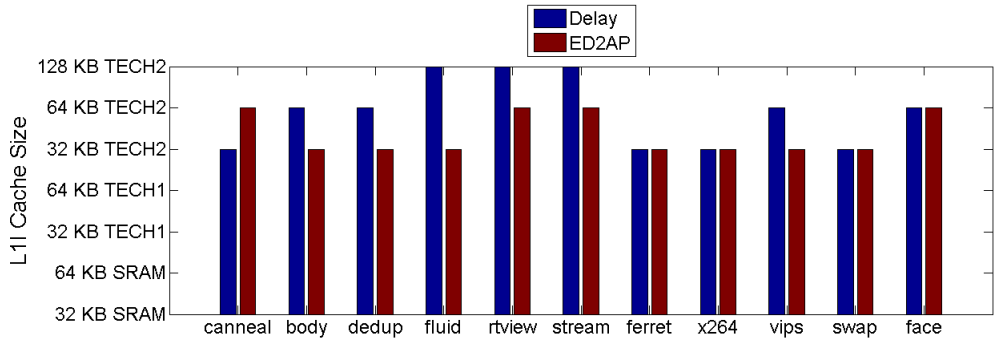

Figure 6.4: Optimal Result for the for L1 Instruction Cache for each benchmark for Delay and ED^2AP

Apparent and expected observations from the minimal solutions, for each benchmark to optimize performance, is that larger capacities(entries) ROB, LSQ, BTB and Branch Predictors give less cost in delay. The solution for L2 as well settles for a large capacity high associativity cache. Solutions also converge towards larger cache line sizes except for rtview. L1 Data cache converges towards SRAM for all (except canneal) as it avoids performance loss which STT MRAM brings in with extra write latency. Canneal inherently has a low IPC and that results have shown that extra write latency hardly degrades performance. Thus, canneal would not necessarily have an optimal value for L1D cache and the algorithm may choose a configuration with any value for L1D as the minimal solution. Again an important conclusion is that with L1I cache converging towards STT MRAM Tech2 that provides a faster cycle access. This also confirms STT

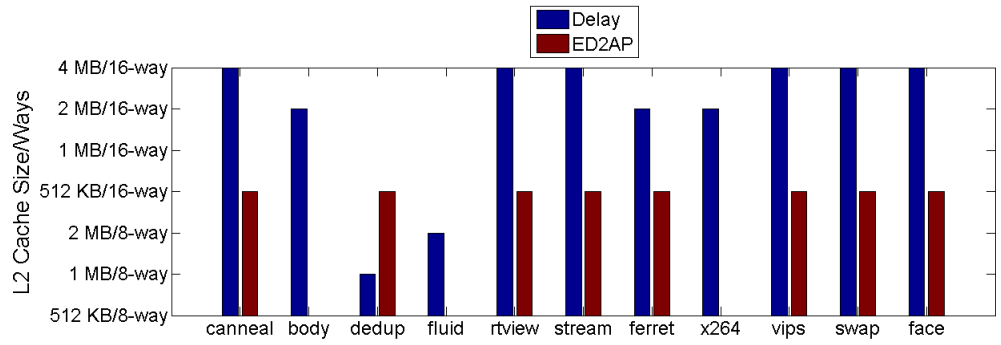

Figure 6.5: Optimal Result for the L2 cache for each benchmark for Delay and ED^{2}AP

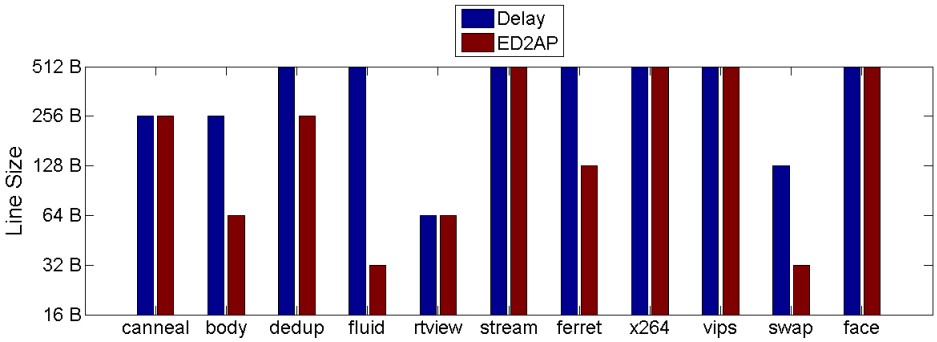

Figure 6.6: Optimal Result for the Cache Line Size each for benchmark for Delay and ED^{2}AP

MRAM Tech2 as an ideal choice for L1I cache since it brings in overall benefits with performance, area and energy.

A graphical comparison of how parameters scale, when the cost is changed from delay to ED^{2}AP, can be obtained by comparing each design parameter from the respective solution vectors for delay and ED^{2}A for each benchmarks. Figures 6.7,6.8 and 6.9 show how architectural parameters scale when area and energy impact is considered. Figure 6.3 points shifts in technology from SRAM to STT MRAM with ED^{2}AP as cost. As discussed, the area, leakage and read access energy reductions overcome the increase in delay and dynamic write energy. The L1I values fairly remains the same for ED^{2}AP as seen in Figure 6.4 indicating STT MRAM Tech2 as optimal throughout. The Line size for bodytrack, fluidanimate and swaptions drop in values though it remains constant for others as shown in 6.6. L2 as discussed shows convergence for smaller capacity 512

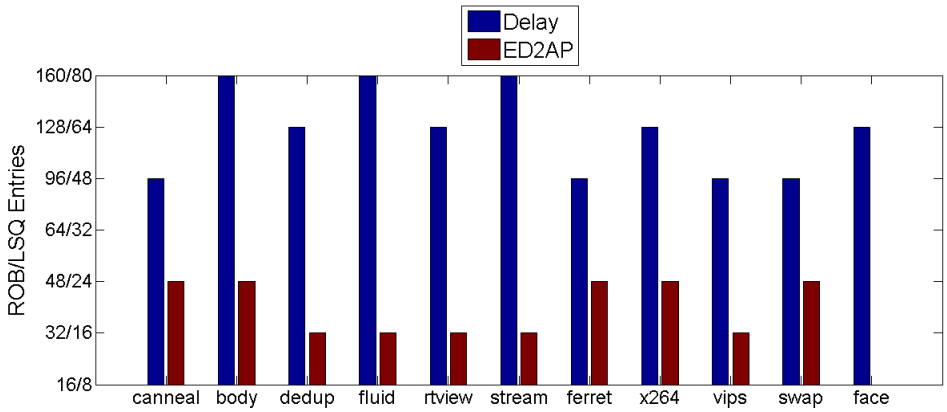

Figure 6.7: Optimal Result for the Number of Re-order Buffer Entries for each benchmark for Delay and ED^{2}AP

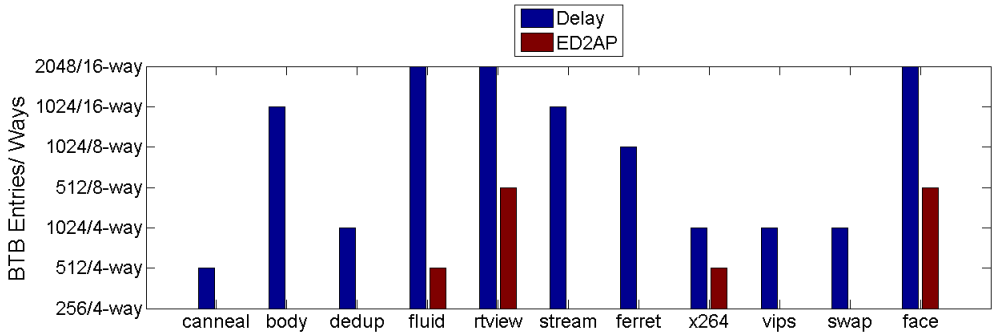

Figure 6.8: Optimal Result for the Branch Target Buffer for each benchmark for Delay and ED^{2}AP

KB cache as seen in 6.5.

6.2.3 Global Optimal Configuration across all Benchmarks

The convergence obtained for individual benchmarks gives a range in which a optimal value for each of the parameters should lie for minimal cost. Though solutions for L2, L1I cache and BTB points to a similar values for optimal solution, L1D cache and Line size show varied results which depends on the characteristics of the application. Essentially it is important to know a global optimal configuration that would give best

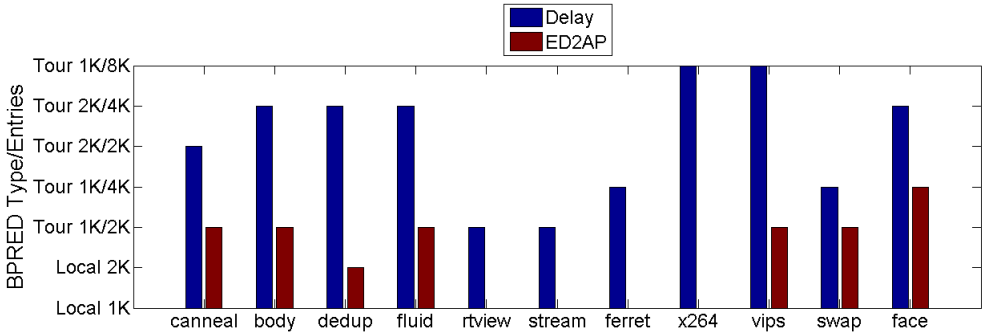

Figure 6.9: Optimal Result for Branch Predictor Type for each benchmark for Delay and ED^{2}AP

solution across all applications. We need to find that towards which memory technology, from the mix of SRAM, Tech1 and Tech2, the global optimal will point and what capacity should fit in as a first level on-chip cache in a CMP. This will further help in appropriate standard parameter selection for running simulations especially for gem5 using PARSEC.

The ED^{2}AP cost function for the global optimal is computed by taking harmonic means of IPCs and other rates and events across all benchmarks. Thus the cost function is the representative for the entire PARSEC suite. We apply the similar optimization procedure using simulated annealing across the same search space to find the configuration with minimal solution. Table 6.7 gives the global minimum across all workloads. L2 converges towards 512 KB and 16-way associativity which was seen consistently for most individual benchmarks as well. Both L1I and L1D caches give solution for small capacity 32 KB STT MRAM Tech2 (7 ns) and Tech1 (3 ns) respectively. Thus, for CMPs running parallel workloads, have a small capacity on-chip caches offers a better solution. Exploiting density advantages offered by STT MRAM, even for a L2 shared amongst four cores, by having higher capacity caches does not seem to be an ideal design choice. For L1D cache, though few individual workloads did show better solution with a Tech2 device, when all applications are combined as one a smaller write latency Tech1 device gives the best solution. Whereas, a Tech2 device is a universal solution for L1I cache as it gives performance benefits with faster fetches and thus promises an ideal replacement for SRAM.

Global solutions for processor core parameters like ROB and BTB settle for similar values as seen across individual benchmarks. A low capacity tournament predictor gives the best cost as it shows more accuracy with both local and global predictions and consumes less cost compared to those with large entries. Thus, for a CMP running multi-thread, smaller architectural parameters are seen to be sufficient since there is a limited ILP and speculation to be exploited.

ROB	LSQ	BTB	BPred Type	L1D cache	L1I cache	Line Size	L2 cache
48	24	256/4-way	Local:1K Global:2K	32 KB Tech1	32 KB Tech2	256 B	512 KB/ 16-way

Table 6.7: Global Optimal Configuration for ED^2AP across all Benchmarks

The global minimum configuration observed though may give the best solution, it is important to know how does other low cost configurations vary i.e. configurations that give cost higher than the minimal solution but still form a group of low cost configurations overall. This trend would show the range each parameter variable can have in achieving minimum ED^2AP. Importantly, will give an idea of which parameters, from the set of seven parameters, dictate the overall cost and which have less impact on the cost function.

The resultant cost represents solution for values corresponding to seven different parameters, it can be assumed to be a point lying in a 7-dimensional space. This point, or a set of points representing group of minimum costs, lying in a 7-dimensional space can be represented by parallel lines using parallel coordinates. Parallel coordinates (parallelcoords) [43] is used to plot a matrix of multivariate data with rows as observations and columns as variables. This allows to plot the observations with minimum cost and see the corresponding sequence of parameter values to see which how parameters vary towards giving minimum cost. Figure 6.10 gives the multivariate plot for seven processor parameter values that give low cost solution including the global minimum. We consider top 20 of overall low cost configurations observed during the optimization across all benchmarks. The parallel lines correspond to each parameter variable and the index corresponds to the value taken by the parameter. For ease of representation, we show index number for the parameter values and not its actual value. The indices corresponds to the corresponding vector index in Tables 6.1, 6.2 and 6.3. It is evident that L2 size dictates the cost function with all the low cost configurations having a L2

size of 512 KB. Importantly, observations show that L2 associativities of both 8-way and 16-way contribute equally towards the cost though 16-way gives the best solution. Line size are observed to be ranging from 64 Bytes to 512 Bytes. This is expected as higher line size get expensive even though they give lesser delay and thus values in these ranges contribute equally towards the overall cost.

Sequences for L1I size are narrowed down to STT MRAM Tech2 32 KB and 64 KB caches and again proves that this technology is ideal in replacing SRAM in L1I cache. L1D cache sequences are also narrowed down to STT MRAM devices showing observations of both Tech1 and Tech2 though the minimum is given by 32 KB Tech1 device. Also, none of the minimum cost configurations show SRAM as a choice for L1D and L1I.

We find that sequences for branch predictors spread over entire range and means that the even though a small capacity tournament predictor gives the best solution, the overall cost does not increase dramatically if larger capacity local and global tables are employed. We also find more sequences passing through indices '1' and '2' which corresponds to local predictor with 2K entries and a tournament predictor with 1K and 2K local and global table entries respectively. Sequences for BTB mostly pass through indices '0', '1' and '3' which corresponds to 256 entries/4-way, 512 entries/4-way and 512-entries/8-way justifying the earlier conclusion of low capacity/associativity BTB structures being sufficient. ROB/LSQ sequences are found mostly to be in a range of index '1' to '4' (32/16 Entries to 128/64 Entries). To conclude, though simulated annealing gives the optimum processor configuration, for design space exploration it is important to provide the range in which parameters should lie without affect the overall cost. This helps designers in deciding which parameters should be kept at their optimal values while which others can be varied through a specified range thus allowing for flexibility.

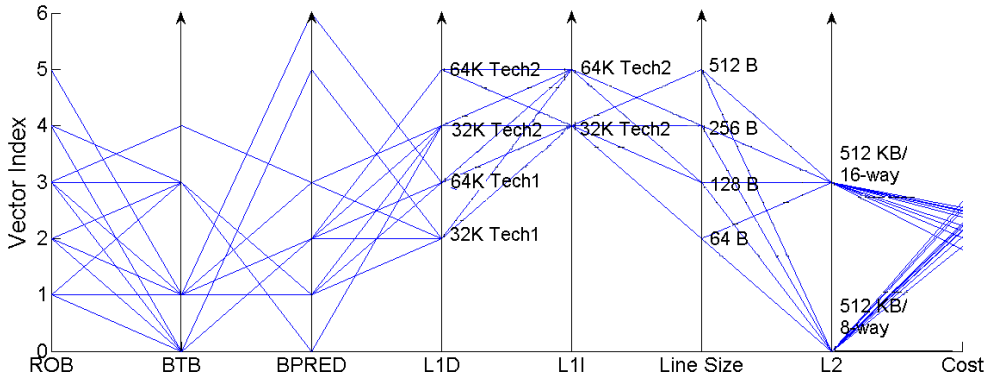

Figure 6.10: Parallel Coordinates showing Multivariate plot of Parameter values for ED^{2}AP Minima

Chapter 7

Conclusion and Discussion

Over the years, STT MRAM is seen to be a promising replacement for last level caches (LLC) and with continuous progress with the device technology it can be a good future alternative for first level caches. The impact of extra write pulse and dynamic write energy is discouraging researchers to replace STT MRAM as an alternative for L1 cache especially the data cache where the frequency of CPU writes is high. But the memory subsystem, especially the on-chip memory, is increasingly turning out to be a bottleneck mainly due to its high leakage consumptions and large area. With more and more cores been integrated in CMPs these days, addressing the issue of decreasing memory footprint will essentially allow for reducing the die cost. Hence if the performance degradation can be balanced with substantial reductions in area and energy, STT MRAM may be thought of as a good alternative to SRAM based caches especially for embedded applications that can afford such compromise.

When a memory hierarchy is replaced with a new technology, it is essential to understand how the impact of micro-architectural parameters change. Considering the same design choices maybe not be productive in oder to exploit the potential of new technologies such as STT MRAM. Thus, we present a methodology essentially to study new memory technologies from a micro-architectural point of view that analyzes whether significant changes in the processor core and memory subsystem are necessary to leverage optimal performance. We apply the well-know statistical technique of Plackett and Burman designs which is used by computer architects to study architectural bottlenecks. Initially, we find the sensitivity of the processor core and memory parameters

towards performance for SRAM and compare how the sensitivity changes when STT MRAM is replaced in the cache hierarchy. It is found that critical processor bottlenecks; LSQ, cache line size, L2 associativity and Re-order buffer remain the same for STT MRAM. Although, sensitivity of LSQ towards performance reduces for few benchmarks indicating less potential exploiting load speculation and load/store dependencies for these workloads. Sensitivity of memory latency towards performance also reduces for some workloads in case of STT MRAM as observed in reduction in main memory requests across them. The significance of the cache hierarchy parameters like the capacity, line size and associativity does not show substantial migration. Computer architects, thus, can continue with the current state of art cache design policies and protocols aimed at improving performance while using STT MRAM in CMPs. As an addition, this work also demonstrates a way of categorizing the PARSEC workload by grouping benchmarks based on their similarity in stressing processor parameters. This grouping allows for more simulation time, as experiments can be run on representative benchmarks in each group, and thus deeper exploration during initial processor design phases for CMPs.

We apply similar PB design experiment to find the important bottleneck parameters towards EDP which allows to analyze the energy overhead associated with performance improvement. Top bottleneck parameters for STT MRAM are observed to be similar to those for SRAM although we find migration of sensitivity in case of cache line size and the LSQ. The reduction in significance for LSQ, observed for few benchmarks, is compounded with the impact observed towards performance which suggests that number of LSQ entries can be relaxed for STT MRAM for such applications though it remains an important bottleneck overall. The drop is significance for the cache line size, observed across some workloads, comes into effect as line fills get expensive w.r.t energy with increase in line size. This is in sync with the research efforts undertaken currently which aims at allocation and replacement policies across the cache hierarchy to reduce line fills. Other cache parameters across L1 and L2 does not show change in their sensitivities. This means, STT MRAM does not dramatically change patterns in which cache updates and sharing that takes place across four cores. Thus, STT MRAM does not offer any leverage which architects can exploit towards reducing EDP across memory and that existing design choices should suffice. Our analysis also conclude that

processor core parameters will be equally stressed for STT MRAM memory encouraging similar micro-architectural design decisions as in case of SRAM memory. Importantly, this work identifies processor and memory bottlenecks for both energy and performance for a CMP having STT MRAM hierarchy running PARSEC benchmark.

In this work, we further consider two distinct types of STT MRAM bit cell design which are based on the trade-off between the write pulse width and the bit cell area. We model STT MRAM based L1 and L2 caches for delay, energy, leakage and area by modifying CACTI to analyze the values of these metrics for these two STT MRAM device technologies on comparison with SRAM. It is found that read access latency for arrays with Tech2 devices are a cycle faster than SRAM and STT MRAM Tech1. This is since it uses a highly dense array reducing global routing delay. Tech2 also consumes 50 % less access energy than SRAM whereas Tech1 takes 20% less energy, although both consume similar energy for line fills which is 4X higher than that for SRAM making line fills highly expensive. Further, Tech1 and Tech2 based array give 4X to 5X reductions in leakage and 3X to 5X gains in density respectively. Tech2 shows promising alternative for L2 where impact of a 7 ns write latency is not as large as it would be for L1 cache where there is a high frequency of CPU stores.

The most important contribution of this work is to realize the optimal memory configuration for a CMP. With the mix of alternatives for L1D and L1I cache, SRAM, STT MRAM Tech1 and Tech2, the best suited device technology should have minimum cost across delay, area and energy. Also, for a STT MRAM based L2 shared across four cores, best cache configuration that replaces SRAM should have should be able to balance higher capacity that STT MRAM allows with peripherals overheads. Further, important processor core parameter like re-order buffer, LSQ, Branch predictors etc., as identified from PB analysis for EDP, can benefit from the optimization in on-chip memory, i.e. reduced energy and area may allow for increased capacity for these structures allowing performance improvements. Thus, the optimal configuration for these processor values should be known to see if indeed a larger structure would benefit the overall cost in energy, delay and area for these processor parameters in a CMP. Thus, considering interaction between memory and processor parameters, we find the optimal configuration for individual benchmarks for these set of parameters that gives the minimum cost of ED^2AP using the optimization procedure of simulated annealing.

We further repeat this method by running all PARSEC benchmarks as one application which would give a global optimal solution across all benchmarks.

The results shows that a low capacity 512 MB STT MRAM with 16-way associativity as an optimum L2 configuration across benchmarks allowing significant savings in on-chip memory footprint and low peripheral leakage and energy. A more important result is that STT MRAM Tech2 (7 ns) proves to be an ideal for L1I cache with benefits across performance, area and energy thus an ideal contender in replacing SRAM. This is since, Tech2 provides a faster access cycle for CPU fetches. For L1D cache, the global solution converges to Tech1 STT MRAM with 32 KB capacity. A larger capacity data cache does not provide enough performance benefits to overcome peripheral overheads. Interestingly, for some workloads that have higher ratio of loads vs. stores, Tech2 STT MRAM seems to be an ideal choice since its extra write latency does not substantially degrade the performance in addition to gains due to less access energy, leakage and area. This in general provides a good alternative in device research which is currently aimed at reducing write pulse at the expense of write current and bit-cell area. Applications which are more read intensive like video processing can manage with higher write latency data cache and can benefit with smaller write current.

Further, for a four-core CMP, the processor core values show best solutions for smaller capacities. This means, reduction in memory footprint does not have to be necessarily leveraged with larger entry ROB, LSQ, BTB etc. and that a better choice would be aiming at integrating more cores on chip by reducing area and energy per core. Alternatively, we can have fabricate more dies from wafers which would be economically beneficial.

Finally this work presents a methodology for exploring design space when a new memory technology is considered as a replacement. This methodology initially identifies bottlenecks and compares if stress on any parameter is potentially impacted. Further, these results can be used to create a design search space and an optimization procedure such as simulated annealing can be applied to find the optimal configuration that would provide minimal delay, area and leakage.

References

[1] J.J. Yi, D.J. Lilja, and D.M. Hawkins. A statistically rigorous approach for improving simulation methodology. In *High-Performance Computer Architecture, 2003. HPCA-9 2003. Proceedings. The Ninth International Symposium on*, pages 281–291, Feb 2003.

[2] Xiaochen Guo, Engin Ipek, and Tolga Soyata. Resistive computation: avoiding the power wall with low-leakage, stt-mram based computing. In *In Proc. of the 37th Annual Intnl. Symp. on Computer Architecture*, 2010.

[3] Christian Bienia, Sanjeev Kumar, Jaswinder Pal Singh, and Kai Li. The PARSEC benchmark suite: characterization and architectural implications. *PACT '08: Proceedings of the 17th international conference on Parallel architectures and compilation techniques*, pages 72–81, 2008.

[4] A. Jog, A.K. Mishra, Cong Xu, Yuan Xie, V. Narayanan, R. Iyer, and C.R. Das. Cache revive: Architecting volatile stt-ram caches for enhanced performance in cmps. In *Design Automation Conference (DAC), 2012 49th ACM/EDAC/IEEE*, pages 243–252, June 2012.

[5] M. Sharad, R. Venkatesan, A Raghunathan, and K. Roy. Domain-wall shift based multi-level mram for high-speed, high-density and energy-efficient caches. In *Device Research Conference (DRC), 2013 71st Annual*, pages 99–100, June 2013.

[6] Xiangyu Dong, Xiaoxia Wu, Guangyu Sun, Yuan Xie, H. Li, and Yiran Chen. Circuit and microarchitecture evaluation of 3d stacking magnetic ram (mram) as a universal memory replacement. In *Design Automation Conference, 2008. DAC 2008. 45th ACM/IEEE*, pages 554–559, June 2008.

[7] Guangyu Sun, Xiangyu Dong, Yuan Xie, Jian Li, and Yiran Chen. A novel architecture of the 3d stacked mram l2 cache for cmps. In *High Performance Computer Architecture, 2009. HPCA 2009. IEEE 15th International Symposium on*, pages 239–249, Feb 2009.

[8] M. Hosomi, H. Yamagishi, T. Yamamoto, K. Bessho, Y. Higo, K. Yamane, H. Yamada, M. Shoji, H. Hachino, C. Fukumoto, H. Nagao, and H. Kano. A novel nonvolatile memory with spin torque transfer magnetization switching: spin-ram. In *Electron Devices Meeting, 2005. IEDM Technical Digest. IEEE International*, pages 459–462, Dec 2005.

[9] Ping Zhou, Bo Zhao, Jun Yang, and Youtao Zhang. Energy reduction for stt-ram using early write termination. In *Computer-Aided Design - Digest of Technical Papers, 2009. ICCAD 2009. IEEE/ACM International Conference on*, pages 264–268, Nov 2009.

[10] Sang Phill Park, Sumeet Gupta, Niladri Mojumder, Anand Raghunathan, and Kaushik Roy. Future cache design using STT MRAMs for improved energy efficiency: devices, circuits and architecture. In *DAC '12: Proceedings of the 49th Annual Design Automation Conference*. ACM Request Permissions, June 2012.

[11] Zhitao Diao, Zhanjie Li, Shengyuang Wang, Yunfei Ding, Alex Panchula, Eugene Chen, Lien-Chang Wang, and Yiming Huai. Spin-transfer torque switching in magnetic tunnel junctions and spin-transfer torque random access memory. *Journal of Physics: Condensed Matter*, 19(16):165209, 2007.

[12] Wei Xu, Hongbin Sun, Xiaobin Wang, Yiran Chen, and Tong Zhang. Design of last-level on-chip cache using spin-torque transfer ram (stt ram). *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 19(3):483–493, March 2011.

[13] David J. Lilja. *Measuring Computer Performance: A Practitioner's Guide*. Cambridge University Press, September 2005.

[14] R.L. PLACKETT and J.P. BURMAN. *The design of optimum multifactorial experiments*. 1946.

[15] Joshua J. Yi, Hans Vandierendonck, Lieven Eeckhout, and David J. Lilja. The exigency of benchmark and compiler drift: Designing tomorrow's processors with yesterday's tools. In *Proceedings of the 20th Annual International Conference on Supercomputing*, ICS '06, pages 75–86, New York, NY, USA, 2006. ACM.

[16] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by Simulated Annealing. *Science*, 220:671–680, May 1983.

[17] Meijuan Gao and Jingwen Tian. Path planning for mobile robot based on improved simulated annealing artificial neural network. In *Natural Computation, 2007. ICNC 2007. Third International Conference on*, volume 3, pages 8–12, Aug 2007.

[18] L.P. Hewlett-Packard Development Company. Cacti 6.5, 2009.

[19] Y. Jiang A. Klemm J.P. Wang J. Kim, H. Zhao and C.H. Kim. Scaling Analysis of In-plane and Perpendicular Anisotropy Magnetic Tunnel Junctions Using a Physics-Based Model. Device Research Conference (DRC), June 2014.

[20] Ki Chul Chun, Hui Zhao, J.D. Harms, Tae-Hyoung Kim, Jian ping Wang, and C.H. Kim. A scaling roadmap and performance evaluation of in-plane and perpendicular mtj based stt-mrams for high-density cache memory. *Solid-State Circuits, IEEE Journal of*, 48(2):598–610, Feb 2013.

[21] 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. *SIGARCH Comput. Archit. News*, 39(2):1–7, August 2011.

[22] Roland E Wunderlich, Thomas F Wenisch, Babak Falsafi, and James C Hoe. SMARTS: accelerating microarchitecture simulation via rigorous statistical sampling. In *ISCA '03: Proceedings of the 30th annual international symposium on Computer architecture*. ACM, June 2003.

[23] Mark Gebhart, Joel Hestness, Ehsan Fatehi, Paul Gratz, and Stephen W. Keckler. Running parsec 2.1 on m5. Technical report, The University of Texas at Austin, Department of Computer Science, October 2009.

[24] O. Tange. Gnu parallel - the command-line power tool. *;login: The USENIX Magazine*, 36(1):42–47, Feb 2011.

[25] R. E. Kessler. The alpha 21264 microprocessor. *Micro, IEEE*, 19(2):24–36, Mar 1999.

[26] K.C. Yeager. The mips r10000 superscalar microprocessor. *Micro, IEEE*, 16(2):28–41, Apr 1996.

[27] M. Tremblay and J.M. O'Connor. Ultrasparc i: a four-issue processor supporting multimedia. *Micro, IEEE*, 16(2):42–50, Apr 1996.

[28] S.P. Song, M. Denman, and J. Chang. The powerpc 604 risc microprocessor. *Micro, IEEE*, 14(5):8–, Oct 1994.

[29] Jason Robert Carey Patterson. Modern Microprocessors. `http://www.lighterra.com/papers/modernmicroprocessors/`, 2012. [Online; accessed May 2014].

[30] D. Leibholz and R. Razdan. The alpha 21264: a 500 mhz out-of-order execution microprocessor. In *Compcon '97. Proceedings, IEEE*, pages 28–36, Feb 1997.

[31] Sheng Li, Jung Ho Ahn, Richard D. Strong, Jay B. Brockman, Dean M. Tullsen, and Norman P. Jouppi. Mcpat: An integrated power, area, and timing modeling framework for multicore and manycore architectures. In *Proceedings of the 42Nd Annual IEEE/ACM International Symposium on Microarchitecture*, MICRO 42, pages 469–480, New York, NY, USA, 2009. ACM.

[32] Major Bhadauria, Vincent M Weaver, and Sally A McKee. Understanding PARSEC performance on contemporary CMPs. *Workload Characterization, 2009. IISWC 2009. IEEE International Symposium on*, pages 98–107, 2009.

[33] C.W. Smullen, V. Mohan, A. Nigam, S. Gurumurthi, and M.R. Stan. Relaxing non-volatility for fast and energy-efficient stt-ram caches. In *High Performance Computer Architecture (HPCA), 2011 IEEE 17th International Symposium on*, pages 50–61, Feb 2011.

[34] Zhenyu Sun, Xiuyuan Bi, Hai Helen Li, Weng-Fai Wong, Zhong-Liang Ong, Xiaochun Zhu, and Wenqing Wu. Multi retention level STT-RAM cache designs

with a dynamic refresh scheme. In *MICRO-44 '11: Proceedings of the 44th Annual IEEE/ACM International Symposium on Microarchitecture*. ACM Request Permissions, December 2011.

[35] Yusung Kim, Sumeet Kumar Gupta, Sang Phill Park, Georgios Panagopoulos, and Kaushik Roy. Write-optimized reliable design of STT MRAM. In *ISLPED '12: Proceedings of the 2012 ACM/IEEE international symposium on Low power electronics and design*. ACM Request Permissions, July 2012.

[36] M. Rasquinha, D. Choudhary, S. Chatterjee, S. Mukhopadhyay, and S. Yalamanchili. An energy efficient cache design using spin torque transfer (stt) ram. In *Low-Power Electronics and Design (ISLPED), 2010 ACM/IEEE International Symposium on*, pages 389–394, Aug 2010.

[37] Kon-Woo Kwon, S.H. Choday, Yusung Kim, and K. Roy. Aware (asymmetric write architecture with redundant blocks): A high write speed stt-mram cache architecture. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 22(4):712–720, April 2014.

[38] Zhenyu Sun, Hai Li, and Wenqing Wu. A dual-mode architecture for fast-switching STT-RAM. In *ISLPED '12: Proceedings of the 2012 ACM/IEEE international symposium on Low power electronics and design*. ACM Request Permissions, July 2012.

[39] Junwhan Ahn, Sungjoo Yoo, and Kiyoung Choi. Dasca: Dead write prediction assisted stt-ram cache architecture. *High Performance Computer Architecture (HPCA2014), 2014 IEEE 20th International Symposium on*, February 2014.

[40] Xiaoxia Wu, Jian Li, Lixin Zhang, E. Speight, and Yuan Xie. Power and performance of read-write aware hybrid caches with non-volatile memories. In *Design, Automation Test in Europe Conference Exhibition , 2009. DATE '09.*, pages 737–742, April 2009.

[41] Amin Jadidi, Mohammad Arjomand, and Hamid Sarbazi-Azad. High-endurance and performance-efficient design of hybrid cache architectures through adaptive line

replacement. In *ISLPED '11: Proceedings of the 17th IEEE/ACM international symposium on Low-power electronics and design*. IEEE Press, August 2011.

[42] M.N. Gurcan, B. Sahiner, H.-P. Chan, L. Hadjiiski, and N. Petrick. Optimal selection of neural network architecture for cad using simulated annealing. In *Engineering in Medicine and Biology Society, 2000. Proceedings of the 22nd Annual International Conference of the IEEE*, volume 4, pages 3052–3055 vol.4, July 2000.

[43] Natick Massachusetts United States The MathWorks, Inc. Matlab and statistics toolbox release 2014a, 2014.

Appendix A

Glossary and Acronyms

Care has been taken in this thesis to minimize the use of jargon and acronyms, but this cannot always be achieved. This appendix defines jargon terms in a glossary, and contains a table of acronyms and their meaning.

A.1 Glossary

- **Instruction per Cycle** (**IPC**) – Performance metric thats gives the number of instructions committed per CPU cycle. It is a measure of throughput of the processor.
- **Speed-Up** – Improvement in performance relative to the baseline value.
- **Region of Interest** (**ROI**) – The main working set in an application workload that reflects its actual behavior without running into issues related to cold cache and sequential overheads.

A.2 Acronyms

Table A.1: Acronyms

Acronym	Meaning
STT MRAM	Spin Transfer Torque Magnetic Random Access Memory
ROI	Region of Interest
B/W	Bandwidth
PB	Plackett and Burman
L1D	L1 Data Cache
L1I	L1 Instruction Cache
IPC	Instructions Per Cycle
EDP	Energy Delay Product
ED^2AP	Energy $Delay^2$ Area Product