12-9-2006

An Energy Efficient Data Cache Implementing 2-way LRC Architecture

Saibhushan Musalappa

AN ENERGY EFFICIENT DATA CACHE IMPLEMENTING

2-WAY LRC ARCHITECTURE

By

Saibhushan Musalappa

A Thesis
Submitted to the Faculty of
Mississippi State University
in Partial Fulfillment of the Requirements
for the Degree of Master of Science
in Electrical Engineering
in the Department of Electrical and Computer Engineering
Mississippi State, Mississippi

December 2006

AN ENERGY EFFICIENT DATA CACHE IMPLEMENTING

2-WAY LRC ARCHITECTURE

By

Saibhushan Musalappa

Approved

Yul Chu Assistant Professor of Electrical and Computer Engineering (Director of Thesis)	Robert Reese Associate Professor of Electrical and Computer Engineering (Committee Member)
Jerry Bruce Assistant Professor of Electrical and Computer Engineering (Committee Member)	Jim Harden Professor of Electrical and Computer Engineering (Graduate Program Director)

Roger King
Associate Dean of the College of Engineering

Name: Saibhushan Musalappa

Date of Degree: December 8, 2006

Institution: Mississippi State University

Major Field: Electrical Engineering

Major Professor: Dr. Yul Chu

Title of Study: AN ENERGY EFFICIENT DATA CACHE IMPLEMENTING 2-WAY LRC ARCHITECTURE

Pages in Study: 97

Candidate for Degree of Master of Science

Conventional level one data caches are widely used in high-performance microprocessors. Shrinking process parameters in chip fabrication technology allow a much larger number of devices on a chip with every new generation. This reduction in device size has led to an increase in the magnitude of leakage energy. Transistor level leakage energy research for sub-micron processes has shown that leakage can be as much as or greater than the dynamic energy for advanced circuit designs. Researchers have devised techniques to reduce leakage energy at the fabrication and circuit levels. Transitioning the idle circuits from operating voltage to a reduced voltage is one such circuit-level technique. The ELRU-SEQ replacement policy exploits this technique to control cache bank transitions. This thesis proposes a new cache architecture called 2-way Leakage Reduction Cache (LRC) that uses this replacement policy. The architecture employs xor-mapping function to reduce conflict misses.

DEDICATION

I dedicate this research to God, my parents Mr. Bhogi Bhushan and Mrs. Saileela, my sister Jayasree and my grandmother Seshmma.

ACKNOWLEDGEMENTS

I acknowledge Dr. Yul Chu, my major advisor. He has given the guidance and support throughout my graduate program. His insight and great knowledge directed me to my exploration in cache architectures.

I thank my committee members, Dr. Robert Reese and Dr. Jerry Bruce, for mentoring me through out my course work. The valuable knowledge they imparted helped me to gain the basics and insight necessary for this research.

I thank Dr. Norman Jouppi for his advice on Cacti cache power simulator, Dr. Kalla for advice and suggestions on LRU-SEQ policy, Dr. Krisztian Flautner for advice and suggestions on transition circuits, and Dr. Seznec for the CACHESKEW simulator.

My sincere thanks also go to Dr. Nicholas Younan and Mr. Lane, for their long-continued support during my research and course-work. I thank the department of Electrical and Computer Engineering for all the computing facilities and access to needed software.

I thank my friend Shivakumar Sundaram for his support and suggestions.

Finally, I thank my parents for their encouragement and support. Without their love and trust, this work would not have been possible.

TABLE OF CONTENTS

CHAPTER Page

APPENDIX Page

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVATIONS

ABBREVATIONS	
DM	Direct-mapped
SAC	Set-Associative Cache
CMOS	Complex metal oxide semiconductor
LRU	Least Recently Used
LRU-SEQ	Least Recently Used – Sequential
ELRU-SEQ	Extended Least Recently Used – Sequential
SEQ-DST	Sequential Distance
ESEQ-DST	Extended Sequential Distance
XOR	Exclusive OR
SPEC	Standard Performance Evaluation Corporation

LIST OF ACRONYMS

ACRONYMS	
Skew cache	Skewed associative cache
CS	cache size
LS	line size
Assoc.	associativity
Avg.	Average

CHAPTER I

INTRODUCTION

Power dissipation in advanced microprocessors has recently become a first order problem [1]. High power dissipation, mostly in the form of heat, degrades the performance of the whole computing system, increases cooling costs, and significantly reduces battery life for mobile systems.

1.1 Problem Statement

Cache memory is one of the main components of the microprocessor. It is used to bridge the performance gap between the processor and lower-level memory systems in terms of operating speeds. Faster processors need larger caches to function efficiently, and hence the instruction and data caches are allocated a larger portion of die area. These cache structures occupy more than 60% of the modern microprocessors' die area [2].

Table 1.1 Functional power dissipation percentages for a StrongARM processor [15]

Functional Unit	Power Dissipation (Percentage)
Instruction Cache	27
Instruction Unit	18
Data Cache	16
Clock	10
Instruction memory management unit	9
Execution Unit	8
Data memory management unit	8
Miscellaneous	4

Table 1.1 [15] shows the power dissipation percentage in a StrongARM processor that operates at 160MHz and 0.35um technology per functional unit. Instruction and data caches cause 43% of processor power dissipation. Table 1.2 shows the processor name, release date, operating frequency, number of transistors on-chip, and process technology for the Intel family of microprocessors [19]. The term "process technology" refers to the feature size of structures, such as gate length on a transistor. Process technology scaling has resulted in the realization of processor designs with millions of transistors on-chip, e.g., Intel Pentium 4 with a total transistor count of 42 million. Further process technology developments can increase this number well into the billions.

Table 1.2 Comparison of Intel family microprocessors

Processor Name	Release Date	Operating Frequency	Transistor Count	Process Technology
Intel 4004	1971	740 KHz	2300	10 um
Intel 40286	1982	12.5 MHz	134,000	1.5 um
Intel Pentium	1993	66 MHz	3.1 million	0.8 um
Intel Pentium 4	2000	1.4 GHz	42 million	0.18 um
Intel EE 955	2005	3.73 GHz	376 million	0.065 um

Smaller feature size also results in current flow (leakage energy) in the sub-threshold region that causes increased energy dissipation by the transistor. Leakage energy varies exponentially with key process parameters, such as gate length, oxide thickness, and threshold voltage. Due to this increase in leakage, the International Technology Roadmap for Semiconductors (ITRS) predicts leakage energy to be a major technology challenge for advanced process technologies. ITRS is a document that specifies technology challenges and needs of the semiconductor industry over the next 15 years. The Semiconductor Industry Association (SIA), the European Electronic Component Association (EECA), the Japan Electronics and Information Technology Industries Association (JEITA), the Korean Semiconductor Industry Association (KSIA), and the Taiwan Semiconductor Industry Association (TSIA) sponsor this roadmap [5].

With the number of transistors doubling every two years, the dynamic as well as static power dissipation of future integrated circuit designs will increase drastically. Table 1.2 lists the ITRS projections for the maximum allowable power dissipation in CMOS-based microprocessors [5]. Incessant increase in power dissipation by microprocessors and tight

power dissipation requirements set by ITRS provide an added impetus to developing a solution to control and reduce power dissipation.

Table 1.3 ITRS projection for maximum allowable power dissipation (W) in microprocessors [5]

Processor Type	2004	2009	2013	2018
High Performance	158	198	198	198
Cost Performance	84	114	138	168
Battery Powered	2.2	2.7	3	3

1.2 Motivation

Conventional cache memories dissipate power in two ways–dynamic power and static power. Dynamic power is the power dissipated when circuits switch. This power dissipation is calculated using the formula $P_{dynamic}$ α $C*V^2*f$, where C is a load capacitance, V is an operating voltage, and f is a frequency of operation. Static power is the power that dissipates (leaks), even when the circuit is inactive. The amount of power dissipated through leakage depends on device parameters, such as threshold voltage, gate size, and oxide thickness. With process technology scaling, these device parameters become small enough to cause quantum physical phenomena that allow current flow in unconventional paths in a transistor. Figure 1.1 shows the leakage paths in a CMOS transistor [17]. In Figure 1.1, important quantum physical phenomena that cause leakage current are reverse bias p-n junction leakage (I_1), sub-threshold leakage (I_2), oxide tunneling current (I_3), gate current due to hot-carrier injection (I_4), gate induced drain

leakage (I_5), and punch through (I_6). Due to these effects, the power dissipated by caches through leakage is significant and increases with technology scaling.

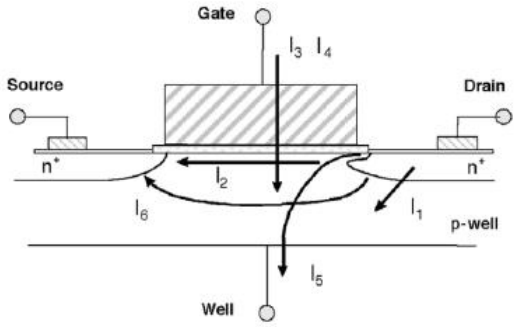


Figure1.1. Leakage phenomena in a CMOS transistor [17]

As cache memories occupy as much as 60% of processor die-area [2], leakage energy due to caches is significant as well. Studying the cache behavior, such as data placement and replacement, cache utilization during program execution (% of cache lines per bank that hold valid data), and cache access rate for each cache bank will help create a leakage energy control strategy at the architectural level. Though instruction and data caches are physically the same, their behavior is quite different. An instruction cache, during execution, can predict which instructions to fetch from memory due to the outcome of execution of instructions such as JUMP, GOTO, etc. Since all instructions between two successive branches occur sequentially, their execution is certain. The cache architecture exploits this capability to profile executing program and implement a leakage control strategy. In the case of data caches, such an approach is not feasible, because which data is needed for an operation is impossible to know until that instruction is executed. This inherent unpredictability of data makes data caches notoriously hard to profile and also

impedes the implementation of a leakage control strategy. Table 1.1 shows that a data cache consumes a significant portion (24%) of total processor power dissipation and calls for an effective way of cache memory accesses and storing data in a cache memory to control and reduce it in order to reach the goals set by ITRS in Table 1.3.

As shown in Table 1.2, process technology scaling is a driving factor in the miniaturization of device size that can be realized on a silicon wafer. Since more transistors can be fabricated per a unit chip area with smaller feature sizes, power density (power dissipated per unit chip area) of the chip increases drastically. Dynamic power dissipation of these smaller transistors must be reduced to keep the power density of the processor core within acceptable limits. Since dynamic power is proportional to the square of supply voltage, the operating voltages for these scaled down devices are also reduced. This reduction achieves a low switching energy per device. However, reducing the supply voltage without scaling the transistor threshold voltage (V_{th}) increases the switching time. The relationship between transistor propagation delay (T_{pd}) and its threshold voltage (V_{th}) is given by equation (1) where C is the load capacitance, V_{dd} is the supply voltage, V_{th} is the transistor threshold voltage, and K models the short channel effects.

$$Tpd \propto \frac{CV_{dd}^{\ 2}}{(V_{dd} - V_{th})^{K}} \tag{1}$$

Equation (1) demonstrates that reducing V_{dd} alone will make the transistor slower, which adversely affects its performance. Therefore, the transistor's threshold voltage should also be scaled proportionally [3]. As explained above, sub-threshold leakage

currents increase significantly with reduced threshold voltages [4]. Recent studies have shown that leakage power dissipation could exceed the dynamic power dissipation if the current trend of device scaling continues [5]. Researchers have presented many circuit level techniques, such as gated-V_{dd}, multi-voltage technology CMOS, dual-V_t CMOS, to reduce this leakage power [6][7][8]. These techniques reduce static power, but they also create a new type of power loss, i.e., transition power. Transition power loss occurs because activating and deactivating circuits consumes power. Kalla [10] reported significant power savings by controlling these transitions in a conventional set-associative cache [9].

As explained above, data caches are difficult to profile due to the random nature of data. However, locality analysis research has proved that every program has both temporal and spatial locality to some extent [20], and this property of spatial locality can aid the implementation of a leakage control strategy for caches. Temporal locality is the property of a code where a cache line once referenced will be referenced again in the near future. This type of locality is observed in program constructs like loops. Spatial locality principle dictates that if a data location is referenced, data locations with nearby addresses will tend to be referenced soon. Program constructs like arrays exhibit spatial locality. The LRU-SEQ replacement policy has successfully exploited this property in instruction caches. This replacement policy constrains cache line fills to the previously hit bank if the accessed cache line satisfies a specific criterion. Chapter two explains in detail the operation of the LRU-SEQ cache. This replacement policy has successfully

reduced cache power dissipation without significant loss of performance in terms of miss-rates [10].

Cache utilization is another important parameter that affects the miss-rate of a data cache. The skewed associative-cache has successfully dispersed data in caches, resulting in better cache utilization and reducing miss rates [11]. Chapter 2 also explains in detail the skewed associative-cache. For this research, I consider skewed associative-cache as the base model to investigate new cache architecture LRC (2-way skew/ELRU-SEQ) for data caches.

1.3 Organization

The rest of this thesis consists of 6 chapters. Chapter 2 discusses the conventional replacement policies and mapping functions. Chapter 3 explains the 2-way LRC cache architecture, flowchart, and operation. Chapter 4 describes the integrated power model and the simulation tools used and developed for the research. Chapter 5 explains the simulation methodology. Chapter 6 discusses the results and analyses obtained from the experiments, and Chapter 7 provides the conclusion. Appendix A explains the ELRU-SEQ control logic. Verilog code and behavioral waveform show the operation of the control logic. Appendix B lists the data generated for 2-way LRC cache architecture and compares with conventional cache architectures.

CHAPTER II

RELATED WORKS

Cache size, block size, associativity, mapping function, replacement policy, and write policy are the major factors in cache memory design, while miss rate, power dissipation and cache memory area are the key design metrics. The cache size, block size, and associativity can only be increased or decreased, whereas replacement policy and mapping function can be manipulated to meet the cache's design requirements as explained in the following sections.

2.1 Mapping Function

Conventional mapping function generates a unique index for each cache memory access. Data in the cache block is processed if the tag of the cache block at that index matches with the tag of the memory access or else replaced. The direct-mapped cache has a one-to-one relation between the index generated and the cache block. Due to this relation, a direct-mapped cache has a high conflict miss rate, but since only one cache block is accessed, the power dissipated per access is low. An n-way set associative cache(SAC) has a one-to-n relation between the index and the cache block. N-way SAC has significantly less miss-rate than DM cache. However, since n cache blocks are accessed for every cache data request, the power dissipated per access is high. A xor

mapping function, introduced by Seznec [11], reduces conflict misses significantly as explained below.

2.1.1 Xor Mapping Function

In conventional SAC, conflict miss occurs due to static mapping of memory addresses to predetermined locations and contributes greatly to total cache misses. The xor mapping function differs from conventional mapping functions because it generates the cache index differently.

To reduce conflict misses, Seznec [11] introduced a 2-way skewed-associative cache, which maps each address onto different cache lines by using xor mapping functions on each of the cache banks. Figure 2.1 shows the address division and memory address mapping in a 2-way skewed-associative cache. The xor functions disperse the addresses into multiple cache lines in both banks; this dispersion reduces conflict misses.

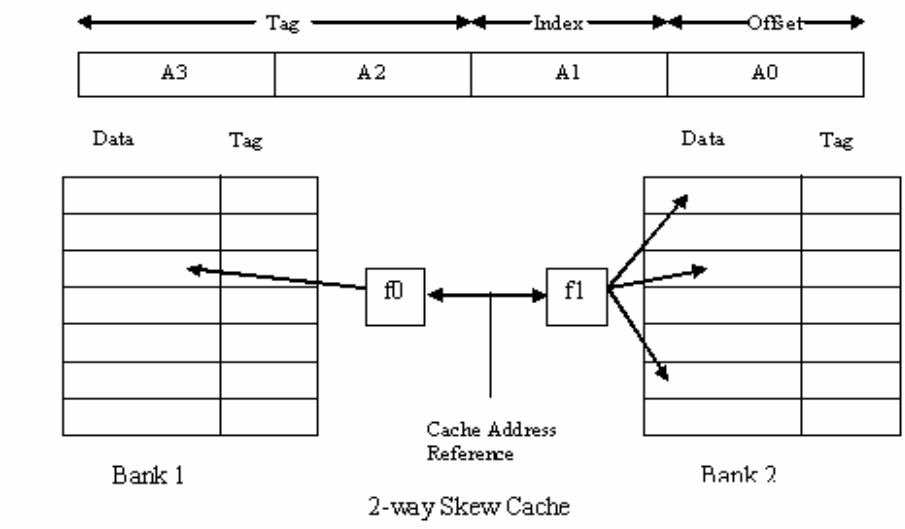


Figure 2.1. Address division and Mapping in a 2-way Skewed-associative Cache

An example that shows a 2-way skewed-associative cache operation is given below. The memory address splits into many parts, as shown in Figure 2.1. Consider the cache memory architecture to have 2 banks, with a cache size of 16KB and block size of 32B, which gives 64 cache lines on each bank. This architecture gives A1 and A2 8 bits each.

Let $f(R) = A2 \oplus shuffle\ (A1)$ and (2.1)

$f_1(R) = A2 \oplus CS[shuffle\ (A1)]$ (2.2)

be the two xor functions chosen for mapping. *CS* [shuffle (A1)] gives the circular shifted value of the shuffled bits of A1. Figure 2.2 shows the shuffling operation. Even bits accumulate toward the most significant locations, and odd bits toward the least significant locations. By this method, for example, the shuffling of A1 = (a0, a1, a2, a3, a4, a5, a6, a7) will generate a new bit sequence A1' = (a0, a2, a4, a6, a1, a3, a5, a7).

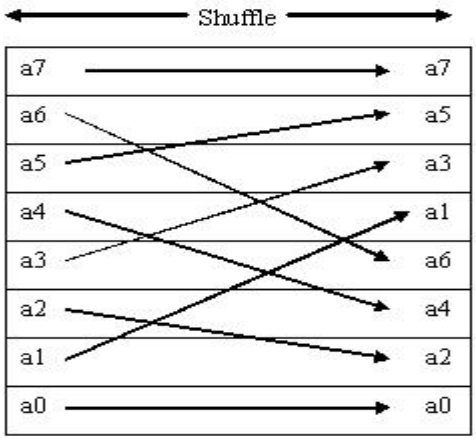


Figure 2.2. Shuffle operation for Skew Caches

The following example shows circular shift operation. Assuming the same cache architecture as above, the number of lines per each bank (L) in the cache is 256, which

gives Log_2 (L) as 8 bits. The following function circular shifts the bits A = (a0, a2, a4, a6, a1, a3, a5, a7) into A' = (a2, a4, a6, a1, a3, a5, a7, a0).

$$Z = ((A << 1) \,\&\, (L-1) + (A >> (L-1))) \tag{2.3}$$

1) A<<1 = (a0, a2, a4, a6, a1, a3, a5, a7, 0)
2) (A<<1)&(L-1) = (a2, a4, a6, a1, a3, a5, a7, 0)
3) (A>>(L-1)) = (0, 0, 0, 0, 0, 0, 0, a0)
4) Z = (a2, a4, a6, a1, a3, a5, a7, a0).

The shuffle and circular shift nets connect the incoming address to the xor module.

The propagation delays involved for a 2-way skewed-associative cache are not very different from that of a 2-way SAC, because the shuffle and circular shift operations do not involve any extra hardware logic. The only delay in the skewed-associative cache is due to the xor gates, and since all the xor gates operate in parallel, the delay is equal to one xor gate delay.

The advantages of using this mapping function are as follows: references with the same index are mapped to different locations, which reduces the conflict miss-rate; the xor function disperses data all over the cache bank, which improves cache line use; and the same cache index can be reproduced by using the memory reference index and tag information, since xor is commutative.

2.2 Replacement Policy

Replacement policy directs which block of data should be replaced to accommodate new data for a cache miss. Conventional replacement policies focus on cache miss reduction because replacing existing data with new data requires accessing lower-level

memory, which is costly in terms of processor cycles and power dissipation. Temporal locality and spatial locality are the two important properties of data that can be exploited to reduce cache misses. Temporal locality principle states that if a data location is referenced then it will tend to be referenced again soon [23]. Spatial locality principle states that if a data location is referenced then data locations with near by addresses will tend to be referenced soon. Temporal locality is exhibited by programs with constructs like loops and spatial locality is exhibited by programs with constructs like arrays. Conventional replacement policies, such as least recently used (LRU), most recently used (MRU), first in first out (FIFO), and not recently used (NRU) exploit temporal locality or spatial locality to optimize miss rate. LRU is biased toward exploiting temporal locality; when it replaces data, it always removes the data that has been used least recently. Therefore, LRU ensures that frequently used data is preserved in the cache. In conventional replacement policies, LRU produces the best miss rates. MRU exploits spatial locality by replacing data that is accessed most recently. This replacement ensures that data that is not yet accessed remains in the cache, which is spatially local to the data that has been recently accessed. On the other hand, FIFO follows a round robin method of replacing the earliest entry to make space for new data.

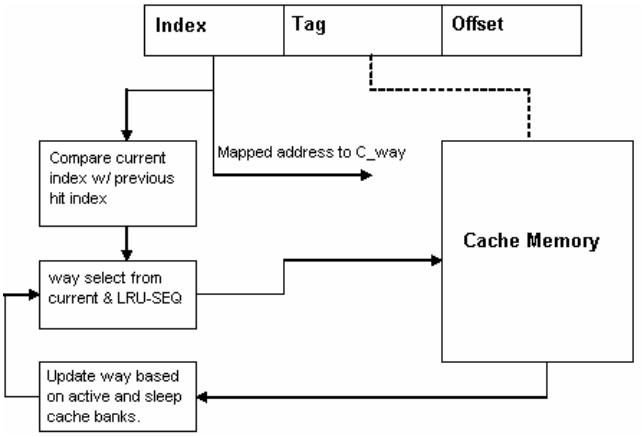


Figure 2.3. LRU-SEQ replacement policy block diagram

The LRU-SEQ replacement policy successfully exploits both temporal and spatial locality in instruction caches because it constrains sequential fills to the previously accessed bank. Therefore, it avoids unnecessary replacement, reduces consumed energy, and lowers miss rates [10]. The LRU-SEQ policy causes architectural level cache optimizations, such as control of bank transitions, by exploiting circuit level design enhancements—for instance: drowsy cache and transition circuit—to generate leakage and transition power savings. The distance parameter SEQ-DST is a numerical value that represents how far a current memory reference index can be from the previous hit reference to be considered spatially local. Therefore, programs with greater spatial locality can have a bigger SEQ-DST value than programs with lesser spatial locality. The SEQ-DST is fixed for single program environments, but can be programmed for multiple program environments. The SEQ-DST can be changed depending on the required power savings and the executing program's spatial locality. In this policy, the hit bank becomes previous way (P_way), and the hit line becomes previous line (P_line). On a miss, the

algorithm compares current index (C_line) with the previous hit line, and if the difference is within the SEQ-DST, then new data fills in the current bank. If the difference is not within the SEQ-DST, LRU replacement determines the best alternative. This replacement algorithm is used as the base model for my research.

2.3 Transition Circuits

Leakage energy is the energy dissipated by a CMOS circuit when it is inactive. As explained in Chapter 1, leakage energy depends on the sub-threshold current and the operating voltage of the circuit. Research into reducing operating voltages for inactive circuits has revealed that a significant reduction in leakage energy is possible [18]. The operation of the LRU-SEQ replacement policy depends on transitioning the cache between normal voltage (1.7v for 0.18u technology) and reduced voltage (0.7v for 0.18u technology) levels. Transition circuits that can change the voltage level of the cache's SRAM cells source code makes this transition possible.

This research uses Krisztian Flautner's transition circuit design [18] in conjunction with the LRU-SEQ replacement policy to implement the LRC cache. Figure 2.4 shows the transistor level circuit, though not the pass transistors, word line, and bit lines of the drowsy bit.

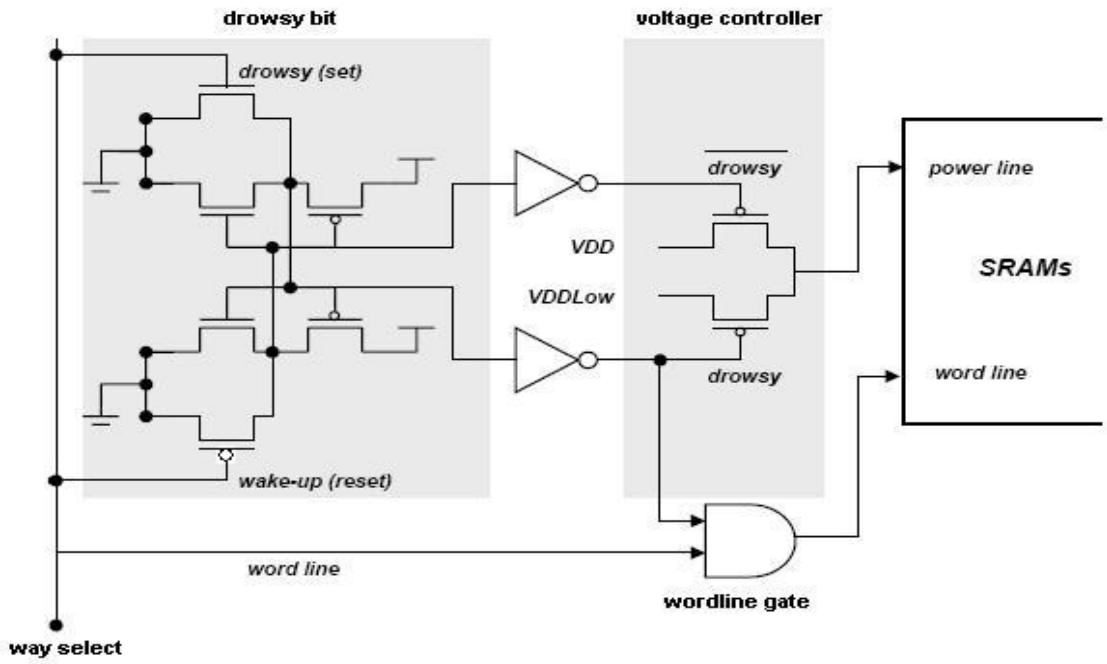


Figure 2.4 Transition circuit implementation [18]

The transition circuit shown in Figure 2.4 operates at two voltage levels: V_{dd}=1.7v and V_{ddLow}=0.7v. The output voltage net of this circuit interfaces with the cache's SRAM cells' source nodes. Way_select signal controls the output voltage of the circuit. The setting of the way_select signal transitions the cache bank from active to sleep state. The LRU-SEQ policy generates this way_select signal.

2.4 Drawbacks of LRU-SEQ

This section presents some of the drawbacks of the LRU-SEQ replacement policy and explains the rationale for using xor mapping function to generate cache memory index.

2.3.1 LRU-SEQ Replacement Policy

The main goal of LRU-SEQ replacement policy is to reduce static power dissipation. Forcing data replacement into an active cache bank if the spatial locality criterion is met achieves this goal. Consider the cache memory that executes the memory trace in Figure 2.5. Let instructions “a” and “c” have different indices in bank 0 and instructions “b” and “c” have the same index. Also, assume that instructions “a”, “b”, and “c” are all within the spatial locality criterion specified by the SEQ-DST parameter.

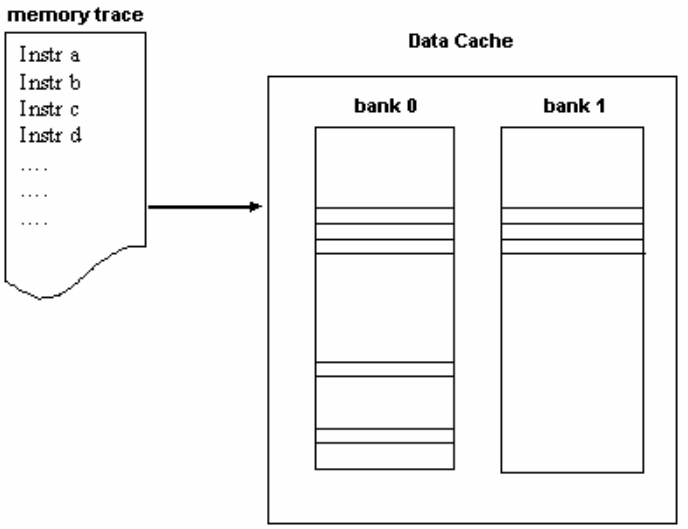


Figure 2.5 Block diagram for cache memory access

For the conventional set-associative mapping function, instructions “b” and “c” cause a conflict because they map to the same address while executing a loop. Therefore, LRU replacement policy places one of the instructions in the other bank, generating a bank transition. In the case of the LRU-SEQ replacement policy, the addresses are within spatial locality and, therefore, must be filled in the same bank. This forced data placement or replacement might result in a conflict miss. Conflict misses due to forced

data replacement can be remedied by employing the xor mapping function, as explained below.

2.3.2 XOR mapping function

As explained in section 2.1, xor mapping generates different indices for different banks in a cache by using both index and tag bits. For the example above, the instructions b and c, though they have identical index bits, differ in tag bits; thus, their skewed index will be different—instructions “a”, “b”, and “c” will be placed in the same bank without any conflicts. Research on skewed-associative caches proves that generating index from xor-ing two sub-parts of a memory address reduces conflict miss rate dramatically [11]. This method of index generationalso results in dispersion of indexes that are sequential in a conventional SAC and results in better use of cache address space

CHAPTER III

2-WAY LEAKAGE REDUCTION CACHE

This thesis proposes an energy efficient cache architecture called the 2-way leakage reduction cache (LRC) that successfully exploits more data localities and attains lower miss-rates than the SAC architecture. Controlling cache bank transitions (from active to sleep and vice versa) reduces leakage energy considerably. This reduction in leakage results in lower energy dissipation. The 2-way LRC architecture implements a xor-mapping function to generate cache indices and the ELRU-SEQ replacement policy to control the transitions of the cache banks. The xor functions map the cache index over different lines of the cache banks, depending on the tag and index bits of the address. These xor functions allows mapping of conflicting addresses to different lines of the cache bank and so reduces the conflict miss-rate. The ELRU-SEQ replacement policy, using the SEQ-DST parameter, controls the data replacement and cache bank transitions. This chapter presents the 2-way LRC architecture, its operation with a flow chart, and the hardware added to conventional cache architecture to implement the ELRU-SEQ replacement policy and xor-mapping function. Finally, it discusses the effect of additional hardware on the critical path and identifies the requirements to extend 2-way LRC into an n-way LRC.

3.1. Logic Diagram for 2-way LRC cache

As discussed in chapter 2, the LRU-SEQ replacement policy uses the SEQ-DST parameter to make a tradeoff between conflict misses and energy dissipation due to bank transitions: high SEQ-DST values (e.g., more than four) decrease transition energy but increase miss-rates, and lower values of SEQ-DST result in a reduced miss-rate with less transition energy savings. The xor-mapping function, as explained in chapter 2, effectively minimizes conflict misses. Figure 3.1 shows the 2-way LRC architecture. The xor-mapping hardware generates the skewed addresses from the incoming memory references. The ELRU-SEQ control logic generates the way_select signal that controls the activation and deactivation of cache banks. Section 3.2 explains in more detail the xor-mapping and ELRU-SEQ control hardware and discusses the 2-way LRC cache operation.

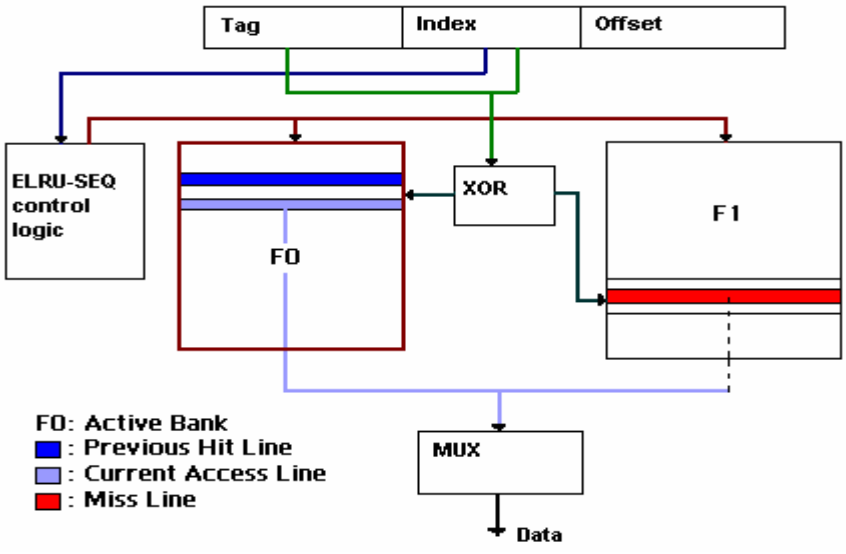


Figure 3.1 Logic diagram of 2-way LRC architecture

3.2. 2-way LRC cache operation

Exploiting spatial locality to reduce bank transitions without affecting miss-rate performance is the main goal of this cache architecture. The parameter SEQ-DST helps indentify spatial locality. When two cache indices are within SEQ-DST distance, then they are assumed to be in the sequential zone of the executing program. However, xor mapping generates cache indices different from set-associative mapping. This difference means that two cache indices in the same sequential zone specified by the SEQ-DST within a SAC will no longer be in that sequential zone of the skewed cache address space. So, program spatial locality and cache spatial locality would be entirely different for a skewed-associative cache.

To ensure that the ELRU-SEQ policy identifies the correct sequential zone inside the cache, changes to the cache architecture are incorporated. To preserve program spatial locality, the ELRU-SEQ logic stores P_{way} and P_{line} into the corresponding registers before the xor logic. This hardware is not in the critical path, and the xor address generation continues. P_{way} is the current active bank, and P_{line} is the previous cache line with a hit in the P_{way} bank. The flowchart in Figure 3.2 shows the algorithm used to implement the LRC cache architecture; the following explains the algorithm:

- Generate skewed indices from incoming memory reference.
- Check the active bank for tag match. Transition banks to make current bank active. If hit, increment the hit counter and update $P_{line} = C_{line}$ and $P_{way} = C_{way}$.

- Check the sleep bank for tag match. Transition banks to make current bank active. If hit, increment the hit counter and update $P_{line} = C_{line}$ and $P_{way} = C_{way}$.
- On miss, subtract the index part of the current memory reference (C_{line}) from the previous hit line index (P_{line}) and compare with SEQ-DST.
- If within the sequential zone, then select P_{way} (current active bank) for block replacement, from the corresponding register. Increment miss counter. Update $C_{way} = P_{way}$.
- Otherwise, replace the block selected by the LRU policy. Update $C_{way} = LRU_{way}$.

The LRC cache has the potential to substantially reduce miss-rates, just like its predecessor, due to its identification of logical sequential zones within the skewed-associative cache address space. The exploitation of spatial locality in data caches and xor-mapping functions' ability to reduce conflicts lead to theorizing that significant miss-rate enhancement can occur.

By controlling the SEQ-DST value, the size of the sequential zone can be altered—a lower SEQ-DST produces a smaller sequential zone and vice versa.The flowchart shows that ELRU-SEQ replacement policy will be executed only when the incoming cache index fails the spatial locality criterion (ABS (P_{line} - C_{line}) <= SEQ_DST). This condition controls the bank transitions of the cache banks. A large sequential zone forces data replacement into the current active bank more often than a small sequential zone, causing better energy savings for large values of SEQ-DST. However, forcing data into cache

lines also results in eviction of current data that increases miss-rate resulting in a tradeoff between energy savings and miss-rate performance.

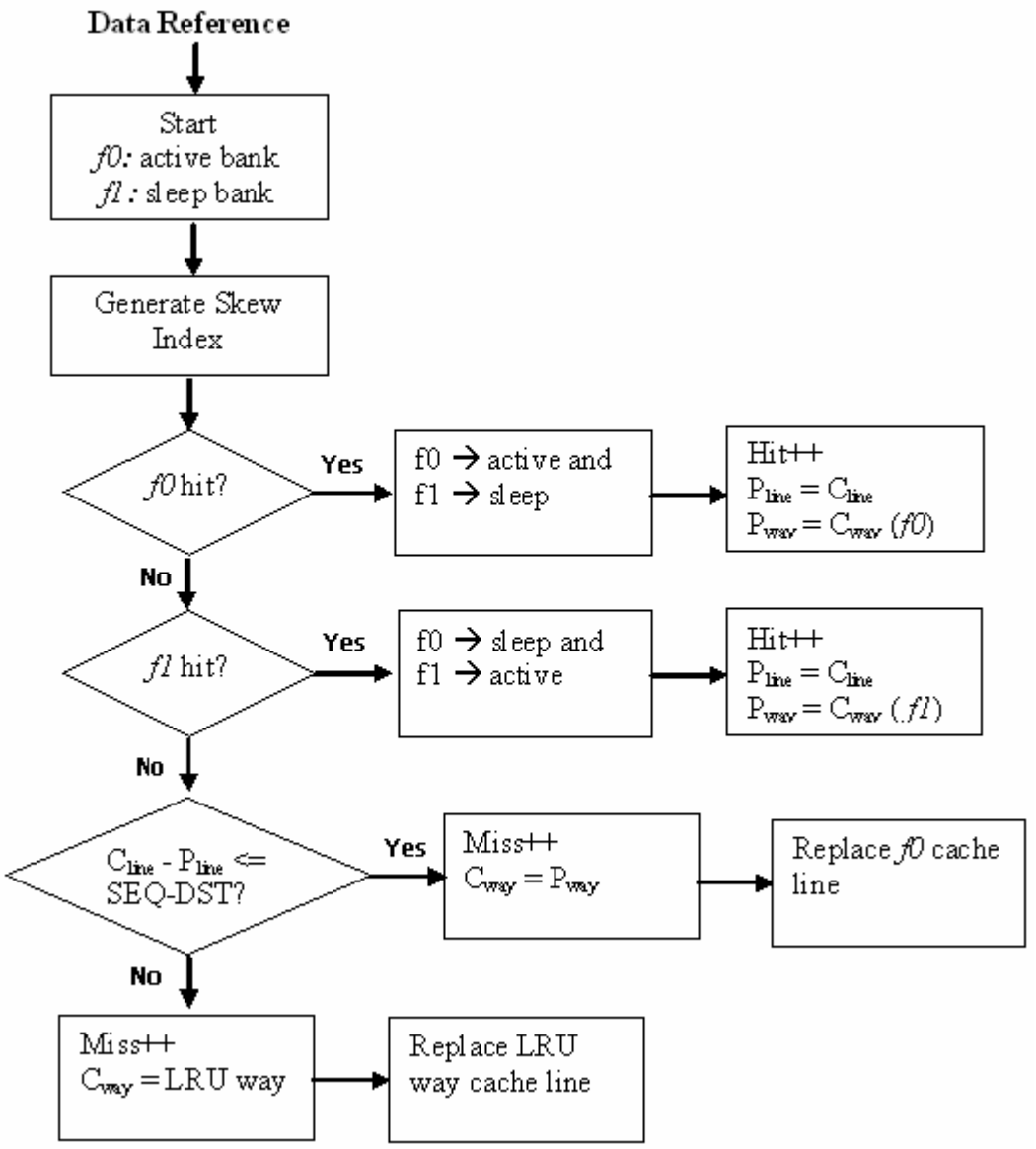


Figure 3.2 Flow chart for 2-way LRC cache architecture

3.3. Hardware Implementation of a 2-way LRC cache

Extra hardware added to conventional cache to model the LRC cache is as follows:

1) Register file (a set of static DFF's) to store the previously accessed index and current index of the instruction, current active bank
2) Logic to compare the current index and previous index (comparison and adder circuits) and generate the way_select signal
3) Multiplexer to issue the way depending on the way_select signal
4) xor mapping logic (a set of xor gates) to simulate a skewed-associative cache
5) Transition circuit to activate or deactivate the cache bank as directed by the way-select signal.

Cadence Spectre netlists simulate the transition logic, DFF's, comparison logic, xor gates, adder circuits, and multiplexers for 0.18um process technology, and their values are added to the base cache values obtained from Cacti 3.2. Figure 3.3 shows the LRC block diagram. Appendix A gives Verilog implementation of the extra hardware.

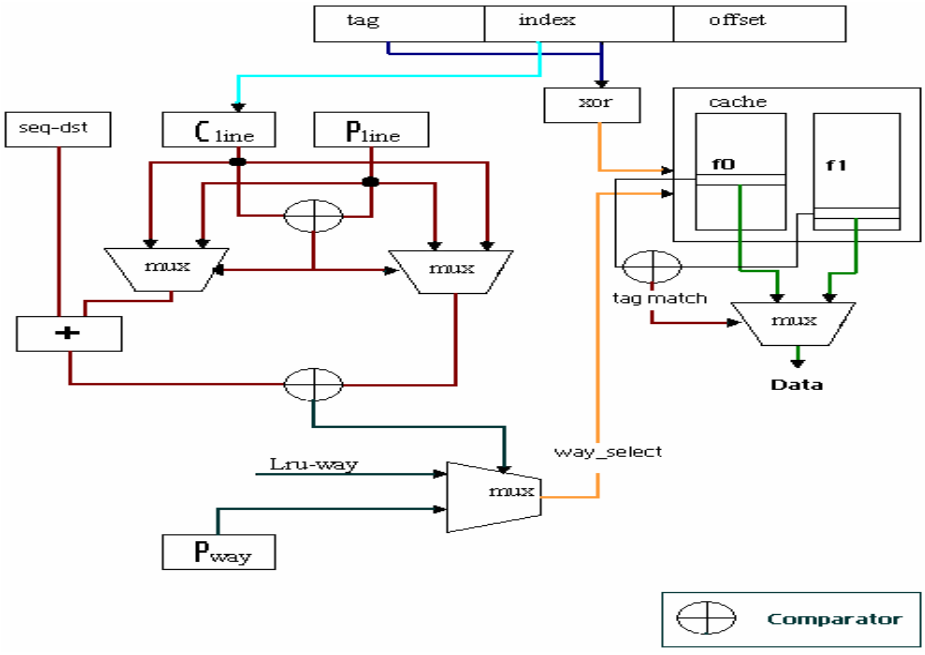


Figure 3.3 Extra Hardware for LRC cache architecture

Figure 3.3 shows the block diagram of the ELRU-SEQ control hardware incorporated into a generic cache design. This hardware executes whenever a cache miss occurs. As shown in the figure, the comparator compares the incoming memory reference (C_{line}) with the previous hit line (P_{line}). The outcome of this comparison drives the two 2-input multiplexers. If C_{line} is greater than P_{line}, then the adder circuit adds SEQ-DST to the P_{line} value; otherwise, to the C_{line} value. After the addition, the two values are compared again. The output of this comparison drives the way_select multiplexer. If C_{line} and P_{line} are within the SEQ-DST criterion, then ELRU-SEQ policy force-fills data from the next level memory into current active bank, and no bank transition will occur; otherwise data is replaced in the LRU bank. If the LRU bank is not the current active bank, then ELRU-SEQ policy transitions the current active bank to sleep mode, and transition the sleep bank to active mode, and the new data is filled into the cache line specified by the skew index.

3.4. Effect of extra hardware on the critical path

The algorithm accesses the xor address generation circuitry to generate the index of every cache memory reference. Because of this task, the propagation delay of the xor address generation circuitry extends the cache access time. The transition circuit, register file, adder circuits, and multiplexer are accessed only when a miss occurs. Since a cache miss initiates a data request to next level memory, the delay due to this path of the circuit is hidden in the L2 cache access times. Also, the power dissipated by the extra hardware

is quantitatively insignificant in comparison to cache dynamic, leakage, and transition power dissipation and does not affect the simulation results. Chapter 4 shows a sample execution cycle of the LRC cache and the calculations of the energy dissipation values for that case.

3.5. N-way LRC

The 2-way LRC cache architecture can be extended to an n-way LRC architecture with slight modifications. The xor address generation circuitry remains the same. The hardwired shuffling functions generate indices for additional banks—refer to chapter 2 for further explanation. For the replacement policy, since only one bank is active at any time, the SEQ-DST comparison does not change for an n-way LRC cache. However, the LRU_{way} part of the circuit becomes very complex. In a 2-way LRC, the LRU cache line can be identified with just one control bit. In a n-way LRC, the same operation requires a n-way comparator to compare $\log_2(n)$ control bits of each bank.

CHAPTER IV

POWER MODEL

This chapter conducts an analysis of the power dissipation of two-way LRC cache architecture, using the Cacti analytical model and Cadence Spectre.

4.1. Cacti 3.2

"Cacti is an integrated cache access time, cycle time, area, aspect ratio, and power model"[12]. This software models a generic cache whose parameters can be set from the command line. For a given cache organization, Cacti details all possible internal configurations and chooses the one with the best-weighted value. Since the software integrates the area model and power model, the area vs. power tradeoff can be compared for a wide set of cache configurations. This research uses a simulation of the cache architecture at 0.18um process technology.

4.2. Cadence Spectre

Figure 3.3 shows extra hardware added to the generic cache; I have implemented this hardware and the transition circuits in Cadence Spectre 0.18um technology. The source capacitance of the cache lines forms the capacitive load for power dissipation calculation of the transition circuit. All the cache configurations investigated using Cacti have the

extra hardware simulated. The power values obtained through Spectre simulations combine with those obtained from Cacti to produce the possible power dissipation values for LRC cache architecture. Mentor Modelsim simulation software establishes the behavioral functionality of this architecture. Appendix A gives a waveform showing the test vectors and the output of the LRC hardware.

4.3. Integrated Power Model

A cycle-level energy model aids in evaluating the energy advantages obtained by this power model, based on the equations below. Equations (4.1) and (4.2) give the energy dissipation by a conventional cache and of an LRC cache, respectively. Equation (4.3) gives the bank power of the LRC cache.

$$E_{CACHEL1} = cycles\_active * bank\_power + (dataset * (associativity - 1) + (dataset - cycles\_active)) * leakage\_percentage * bank\_power \quad (4.1)$$

$$E_{CACHEL1} = cycles\_active * bank\_power_{ELRU-SEQ} + (total\_transitions * transition\_energy * lines\_per\_bank) + (dataset - cycles\_active) * leakage\_percentage * bank\_power_{ELRU-SEQ} \quad (4.2)$$

$$Bank\_Power_{ELRU-SEQ} = bank\_power + associativity * XOR\_power + ELRU-SEQ\_power \quad (4.3)$$

Cycles_active: Total number of accesses to the cache. This number is the sum of accesses to each bank of the cache.

$Bank\_power_{LRC}$: Energy dissipated by the LRC cache bank per active access. It is the sum of bank energy generated by Cacti, energy dissipation of the extra hardware

generated by Cadence Spectre simulation. Bank energy is negligible when it is not accessed, since the bank is in sleep mode.

Dataset: Total number of instructions executed in the simulation run.

Leakage_percentage: The ratio of static energy dissipation to dynamic energy dissipation per bank. Leakage energy is assumed to be equal to dynamic energy for this research work. This assumption is valid for advanced processes such as 65nm and below where ITRS projections show an exponential increase in leakage energy exceeding the dynamic energy by many orders of magnitude. [21]

Total_transitions: Total number of transitions incurred during simulation run. This value is the sum of transitions of individual banks.

Transition_energy: Energy needed to transition bank from active to sleep and vice versa.

Bank_power: Conventional cache bank power per access. Cacti 3.2 power simulator, based on the input cache configuration parameters, generates these values.

4.4. Sample Power Dissipation Calculation

Equations (4.1), (4.2), and (4.3), when solved, reveal the total power dissipation by conventional and unconventional caches simulated for this research. Sample power dissipation calculations for two-way SA/LRU-SEQ cache architecture and for two-way LRC cache architecture are below. The cache configuration used for the example is:

Cache size = 8 KB, Block size = 16 B, Associativity = 2 way, Technology = 0.18um, Dataset = 10 million instructions, Lines per cache bank = 256.

From Cacti 3.2, energy per access per cache bank = 0.5671nJ

From Cadence Spectre, energy per access

Two 8-bit comparison circuit	1.74e-3nJ
Three 8-bit multiplexers	2.81e-3nJ
256 line Transition circuit	1.28nJ
8-bit adder	3.65e-3nJ
1-bit DFF	3.69e-4nJ
Two 8-bit DFF's	5.9e-3nJ
Total energy per access for an 8-bit LRC bank	0.5816nJ

Benchmark used for calculations = SPEC2000::GCC

Number of Load/Store instructions executed = 3085827

Miss count for two-way SA/LRU = 616618

Miss count for two-way Skew/LRU-SEQ = 608465

Number of Bank Transitions = 947297

4.4.1. Energy dissipation calculation for two-way SA/LRU cache

From equation (4.1),

Dynamic energy = instruction count * bank power = 3085827 * 0.5671nJ

= 0.00175 J

Static energy = (dataset * (associativity - 1) + (dataset - instruction count)) * leakage percentage * bank energy

= (10000000 * (2 – 1) + (10000000 – 3085827)) * 1 * 0.5671

= 0.00959 J

Total energy dissipation = Dynamic energy + Static energy = 0.00175 + 0.00959

= 0.0113 J

Average energy dissipation per access = 1.13nJ

4.4.2. Energy dissipation calculation for two-way LRC cache

From equation (4.2), Dynamic energy = instruction count * bank power

= 3085827 * 0.5816nJ = 0.00179 J

Static energy = (dataset – instruction count) * leakage percentage * bank power

= (10000000 – 3085827) * 1 * 0.5816nJ = 0.00402 J

Transition energy = transition count * transition energy per access

= 947297 * 1.28nJ = 0.001212 J

Total energy dissipation = Dynamic energy + Static energy + Transition energy

= 0.00179 + 0.00402 + 0.001212 = 0.007 J

Average energy dissipation per access = 0.7nJ

CHAPTER V

SIMULATION METHODOLOGY

This thesis evaluates the LRC architecture for an on-chip L1 data cache. It evaluates and compares the conventional cache architectures with direct mapped, 2-way, 4-way, 8-way, and fully associative mapping functions and random, LRU replacement policies with 2-way LRC architecture. The evaluations were based on trace-driven execution of Spec2000 benchmarks, compiled for the SimpleScalar PISA instruction set with no optimizations.

Cache sizes of 8, 16, and 32KB and line sizes of 8, 16, and 32B are used for functional and power simulations. Since the objective of this architecture is both to improve miss rates and reduce power dissipation, the data receive both functional and power simulations. Cacti 3.2 and Cadence Spectre perform power dissipation estimation. The Sim-safe program from SimpleScalar toolset and enhanced Cacheskew simulator performs the functional simulation. An enhanced Cacheskew simulator generates the miss-rates and cache bank transition information. Table 5.1 shows the benchmark programs used for simulation purposes. Figure 5.1 shows the simulation flow.

5.1. Functional Simulator

Cacheskew is a cycle level simulator using DINERO-like inputs. Seznec developed it [13] to simulate skewed-cache architectures.

To model the LRC cache, I enhanced this simulator in certain areas. Enhancements made to the simulator include

- Object-oriented design—cacheline class is defined, which simulates one line of a cache bank. The simulator instantiates this class C / (B*A) times where C is
- cache size, B is block size and A is associativity to generate the cache architecture.
- N-way skewed index generation capability—the enhanced Cacheskew simulator can generate indices using the equations (2.1) and (2.2) for N-way skewed-cache while the original software could generate only a 2-way skewed-cache.
- ELRU-SEQ replacement policy—code for simulating SEQ-DST criterion, bank placement and replacement, bank transitions from active to sleep and vice versa form the core of this replacement policy.

Each cacheline object records the tag, data, date of access, and number of accesses. Date of access determines the LRU index. The number of cache line accesses information helps investigate cache utilization and data dispersion capabilities of different mapping functions. Running the benchmark in sim-safe program of the SimpleScalar simulator

suite extracts the inputs. Data collected includes miss rate, bank transitions, and number of accesses to every cache line.

5.2. Benchmark Programs

This research uses Spec2000 integer and floating point benchmark programs. Table 5.1 shows the programs executed, their application area and their description.

Table 5.1 Spec2000 Benchmark Programs [14]

Benchmark	Application area	Description
171.swim	Weather prediction	Shallow-water model with 1335*1335 grid for solving water equations using finite difference approximations.
172.mgrid	Electromagnetism	Solver for a three dimensional potential field
173.applu	Fluid Dynamics/Math	Solves five coupled nonlinear partial differential equations on a 3-dimensional logically structured grid.
177.mesa	3-D graphics	A free graphics library similar to OpenGL
179.art	Image recognition / Neural networks	A neural network employed in recognizing objects in a thermal image.
183.equake	Seismology	A simulator for the study of propagation of elastic waves in large, highly heterogeneous valleys.
188.ammp	Computational Chemistry	A solver for ordinary differential equations defining the motions of atoms in a protein-inhibitor complex embedded in water.
301.apsi	Weather Prediction	Solver regarding temperature, wind velocity, and distribution of pollutants. Calculates statistics on temperature and pollutants in a grid.
175.vpr	Computer aided design	Performs placement and routing in field programmable gate arrays.
176.gcc	C language compiler	C compiler based on gcc version 2.7.2.2. Benchmark runs as a compiler with many optimizations enabled.

(Table 5.1 cont.)

181.mcf	Combinatorial optimization	Derived from a program used for single-depot vehicle scheduling in public mass transportation.
197.parser	Word processing	Syntactic parser of English based on link grammar.
254.gap	Group theory, interpreter	Implements a language and library designed mostly for computing in groups.
300.twolf	Computer aided design	Placement and routing package used in the process of creating lithography artwork for microchips.

5.3. Simulation Procedure

Figure 5.1 shows the flow chart that describes that execution of the simulation of LRC cache and extraction of power dissipation data. Simplescalar benchmark executable is the Spec2000 benchmark program cross-compiled for PISA instruction set. Running this executable in sim-safe generates a memory trace that is captured and converted to DINERO input format. This research used cache sizes from 8KB to 32KB with line sizes from 8B to 32B. These simulations used direct-mapped, 2-way, 4-way, 8-way, and fully associative conventional caches and 2-way skewed associative cache. A distance parameter value of 1, 2, 4, and 8 is used in the LRC cache simulations. The enhanced Cacheskew simulator generates miss-rate, cache line access information, and bank transition data. Additionally, Cacti performs simulations of each of the cache configurations to obtain cache bank energy consumption per access. Similarly, depending on the cache configuration, Cadence Spectre also helps calculate the energy dissipation by extra hardware for LRC cache.

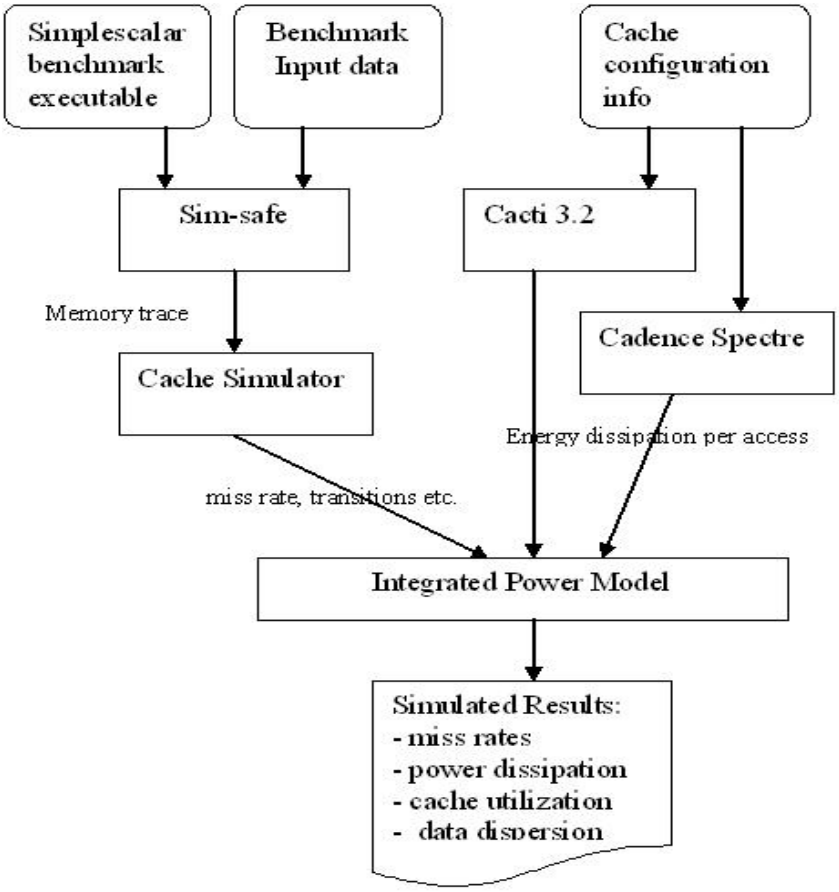


Figure 5.1 Simulation Flowchart

The equations described in section 4.3 calculate total power dissipated and relative power savings for the benchmark programs simulated.

CHAPTER VI

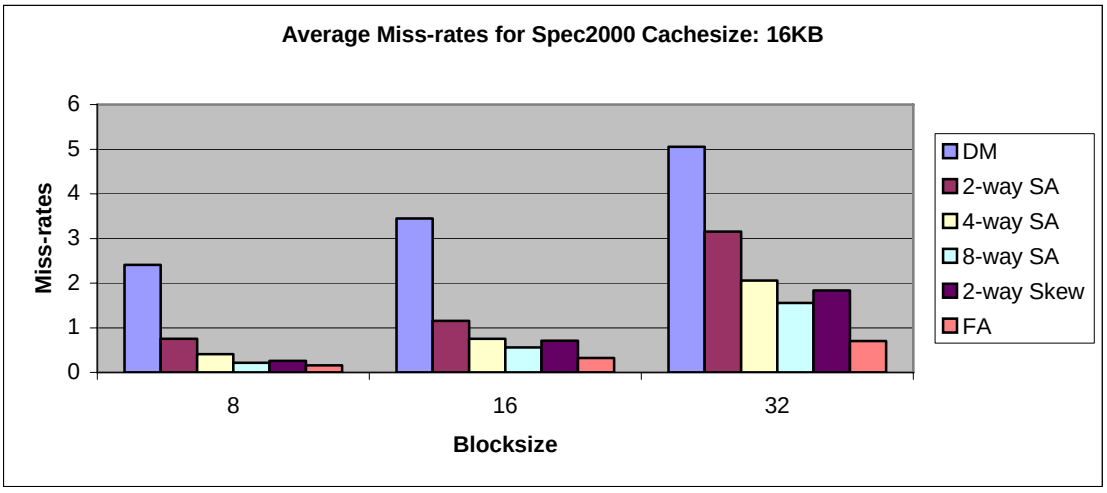
SIMULATION RESULTS

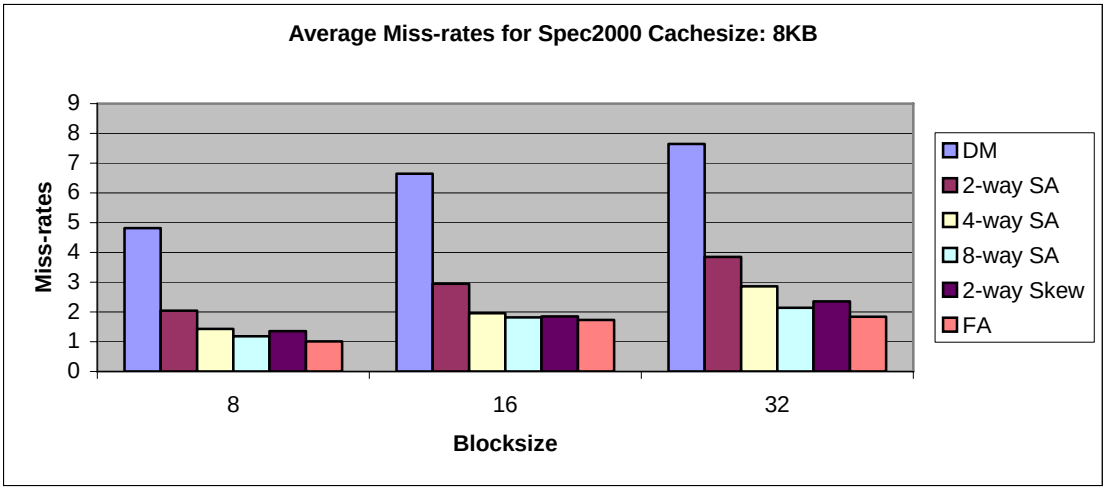
In the current research, improving L1 cache performance by reducing its conflict miss-rate and leakage energy dissipation has been the driving force behind developing the 2-way LRC cache architecture. This section presents the simulation results (miss-rates and energy dissipation) of different cache organizations along with the performance improvement of 2-way LRC cache architecture. The performance of 2-way LRC cache architecture is compared with direct-mapped, set-associative (SA) mapping functions as well as least recently used (LRU), random, and LRU-SEQ replacement policies. This thesis provides the results for those cache architectures with commonly employed block sizes of 8 bytes, 16 bytes and 32 bytes and cache sizes of 8 KB, 16KB and 32KB.

6.1 Cache Miss-rates

This section shows the miss-rates of the conventional cache architectures (DM: direct mapped, 2-way set-associative (SA), 4-way SA, 8-way SA, and FA: fully-associative) and compares them with that of the 2-way skewed-associative cache. As observed in Figure 6.1, 2-way skewed-associative caches (8KB, 16KB, 32KB) have miss-rates similar to those of 4-way SA caches (8KB, 16KB, 32KB), while direct-mapped caches generated the highest miss-rates. As explained in Chapter 2, the xor mapping function

distributes data efficiently across cache space and hence generates significantly less misses for a 2-way skewed-associative cache as compared to a 2-way SA cache





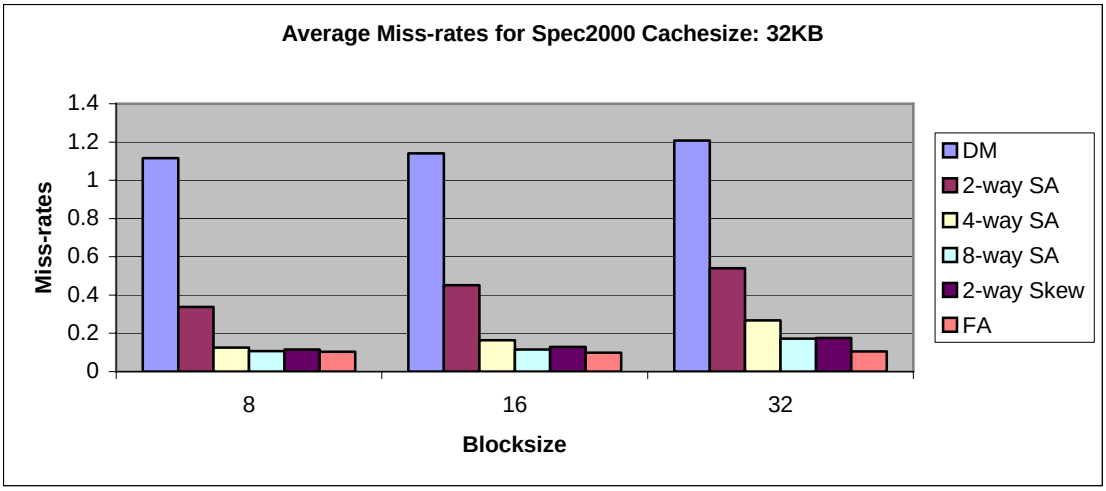
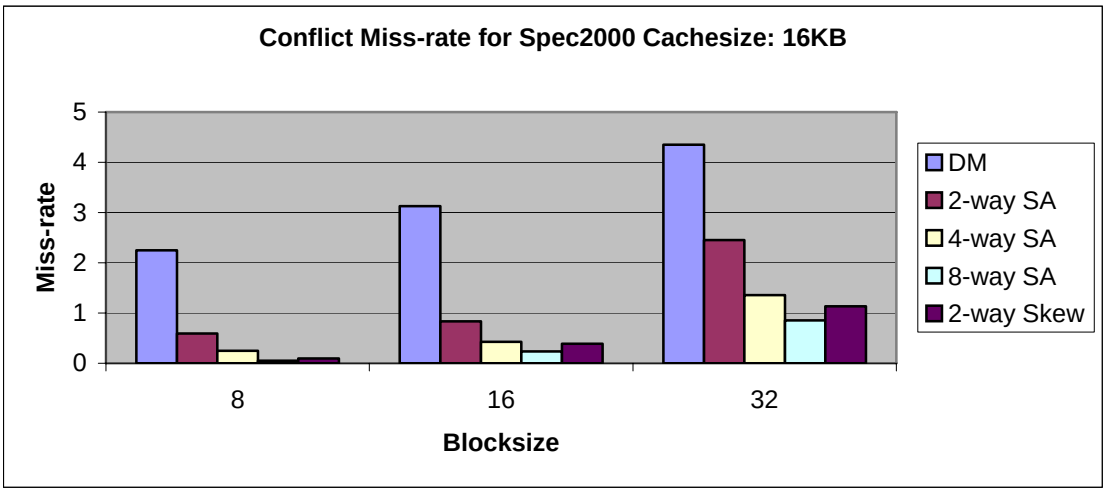
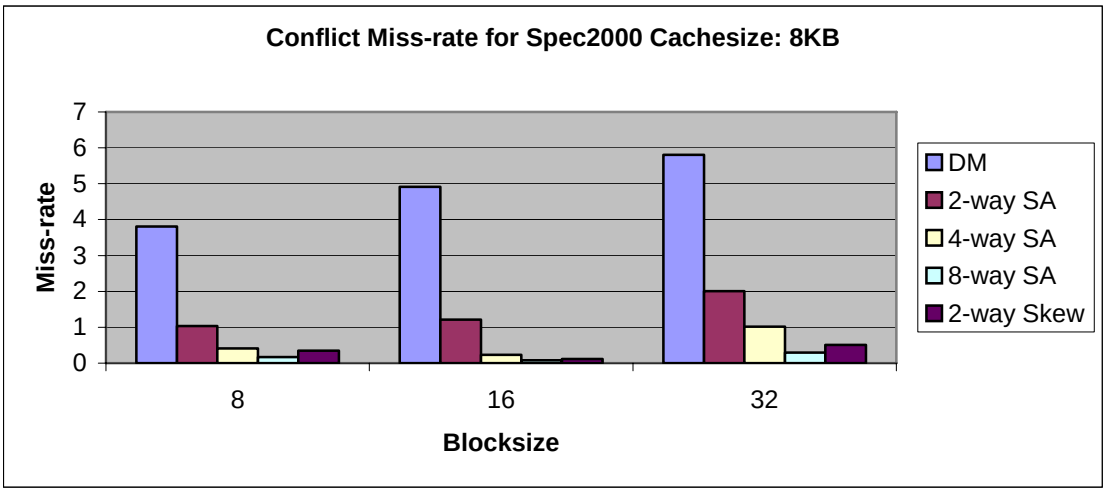


Figure 6.1 Miss-Rate comparison for Conventional cache architectures

6.2 Conflict Miss-rates

Figure 6.2 gives the conflict miss-rates of all the cache architectures used in Section 6.1. The direct-mapped cache produces the highest conflict misses, while the 8-way SA cache produces the lowest. In Figure 6.2, conflict misses increase with increase in block size. This behavior is expected since a bigger block size cache has less number of cache lines per bank than a smaller block size cache with similar cache size. With increase in block size, the number of cache lines per bank decreases. Increased competition for cache lines during data replacement due to a reduction in the number of cache lines per bank results in increased conflict misses. Also observed in Figure 6.2, 2-way skewed-associative caches (8KB, 16KB, and 32KB) with their xor mappings generate conflict misses similar to that of 4-way SA caches (8KB, 16KB, 32KB) like the Figure 6.1. Therefore, the xor mapping function can reduce conflict misses more effectively than conventional mapping functions, e.g., 2-way skew vs. 2-way SA or 4-way SA.


Conflict Miss-rate for Spec2000 Cachesize: 16KB
Miss-rate
0
1
2
3
4
5
8
16
32
Blocksize
DM
2-way SA
4-way SA
8-way SA
2-way Skew


Conflict Miss-rate for Spec2000 Cachesize: 8KB
Miss-rate
0
1
2
3
4
5
6
7
8
16
32
Blocksize
DM
2-way SA
4-way SA
8-way SA
2-way Skew

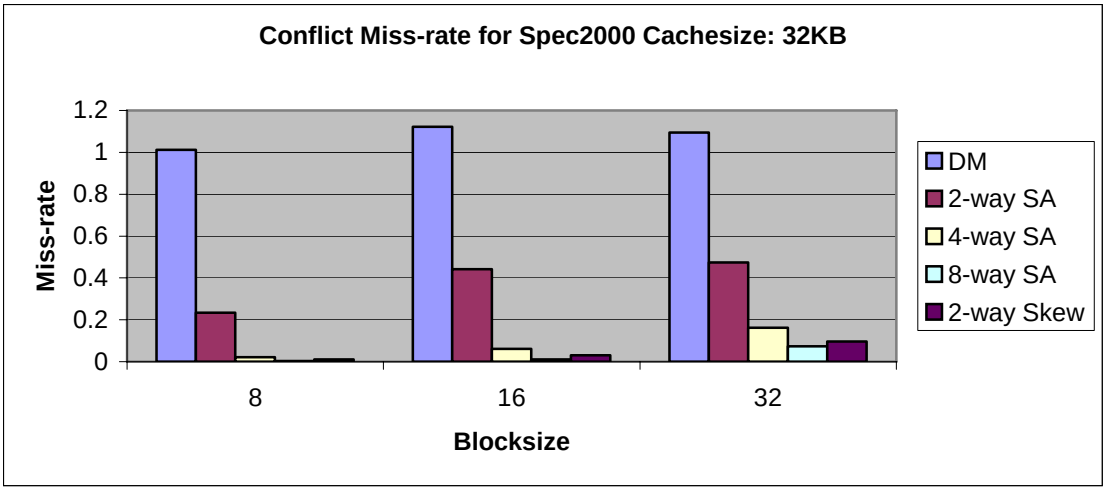
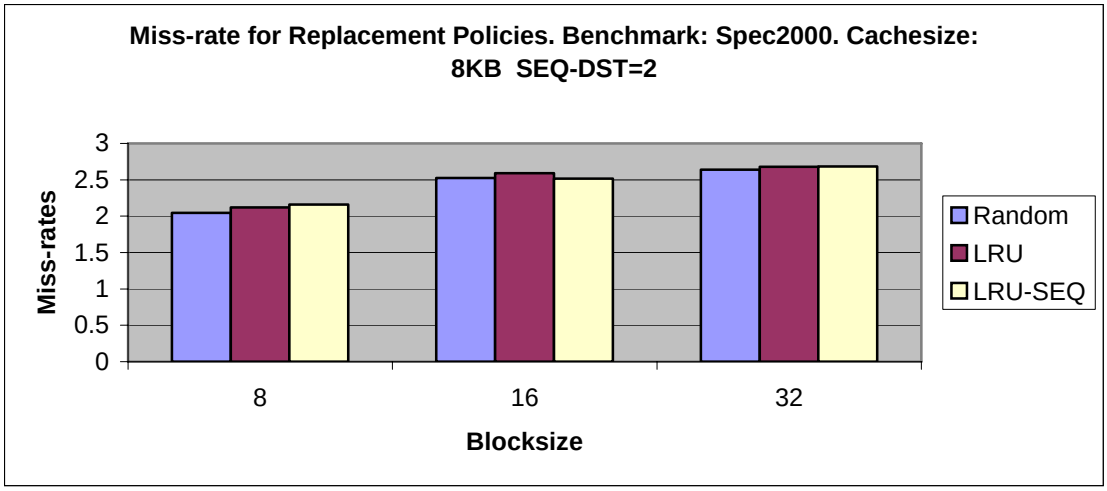
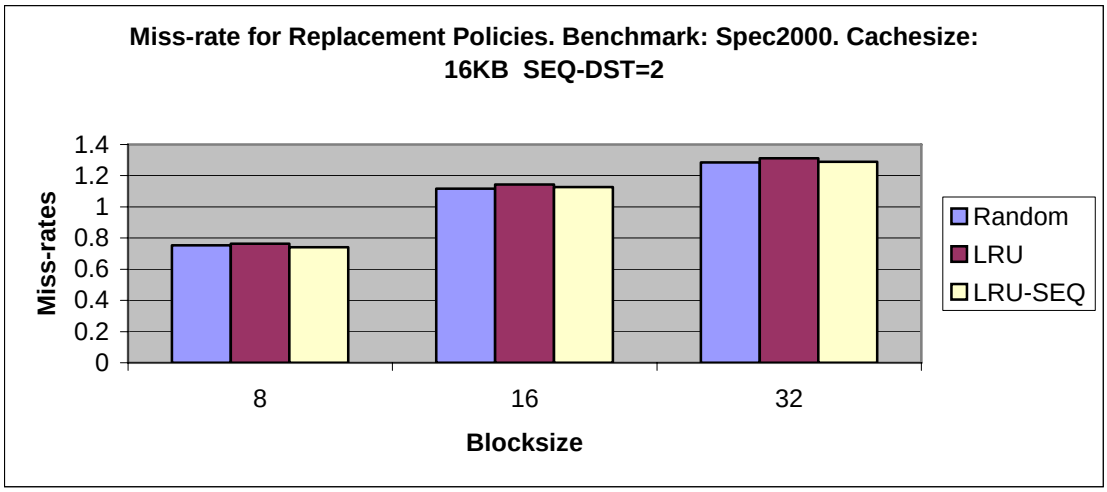


Figure 6.2 Conflict Miss-rates

6.3 Miss-rate Comparison for Replacement Policies

This section shows the miss-rates of the common replacement polices (LRU, random) and compares them with least recently used – sequential (LRU-SEQ) replacement policy. The cache uses 2-way SA mapping function. Consequently, the cache architecture is referred as 2-way SA/LRU-SEQ in following sections. LRU-SEQ policy, as explained in Chapter 2, uses spatial locality to force cache line fills into the currently active bank. The parameter SEQ-DST does this spatial locality assessment.





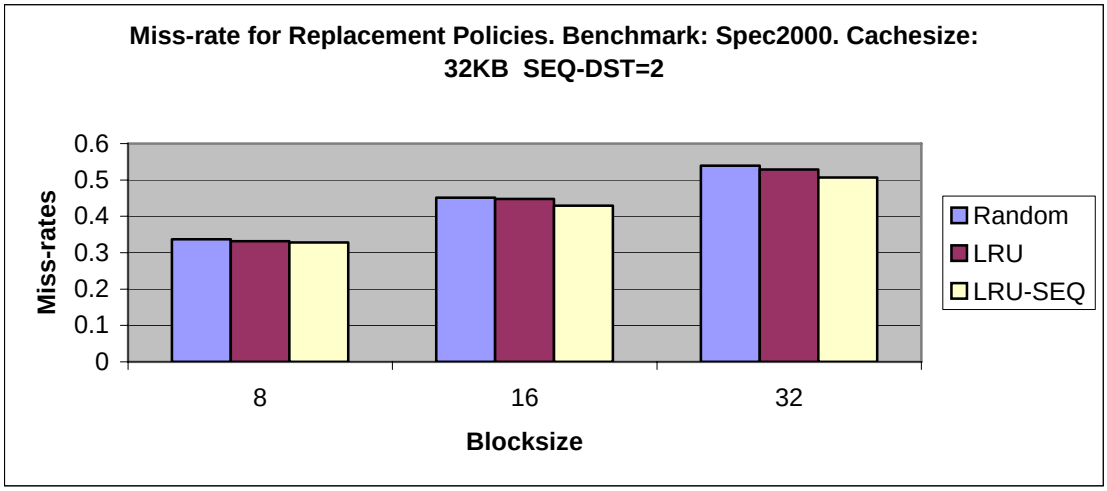


Figure 6.3 Miss-rates comparison for Replacement Policies

For this section, SEQ-DST is set at 2, i.e., in the case of a cache miss, if the cache line index is two lines above or below the previous hit line, data replacement is forced into current active bank. If the criterion fails, the LRU policy is used for data replacement instead.

Figure 6.3 shows that the LRU-SEQ replacement policy has miss-rates similar to that of the LRU policy. Miss-rates for both replacement policies increase progressively with increase in block sizes (8B, 16B, and 32B) for all cache sizes (8KB, 16KB, and 32KB). This increase in miss-rate occurs because LRU-SEQ policy falls back on the LRU policy for data replacement when SEQ-DST criterion fails. Since SEQ-DST was 2 for this set of simulations, only two cache lines above or below the previous hit line are considered to be within spatial locality criterion, significant percentage of memory references for programs without spatial locality will fail the criterion. Therefore, the LRU policy will be used to replace data. As seen in Figure 6.3, similar miss-rates for LRU and LRU-SEQ policies can be inferred that, on average, LRU policy is more frequently used than LRU-SEQ.

6.4 Energy Savings Comparison for Replacement Policies

As explained in Chapter 2, LRU-SEQ policy's main aim is to keep inactive cache banks in a sleep mode as long as possible. This goal is achieved by the use of the SEQ-DST parameter and by exploiting spatial locality. Since inactive banks do not dissipate conventional energy, this policy generates net savings when compared to conventional replacement policies.

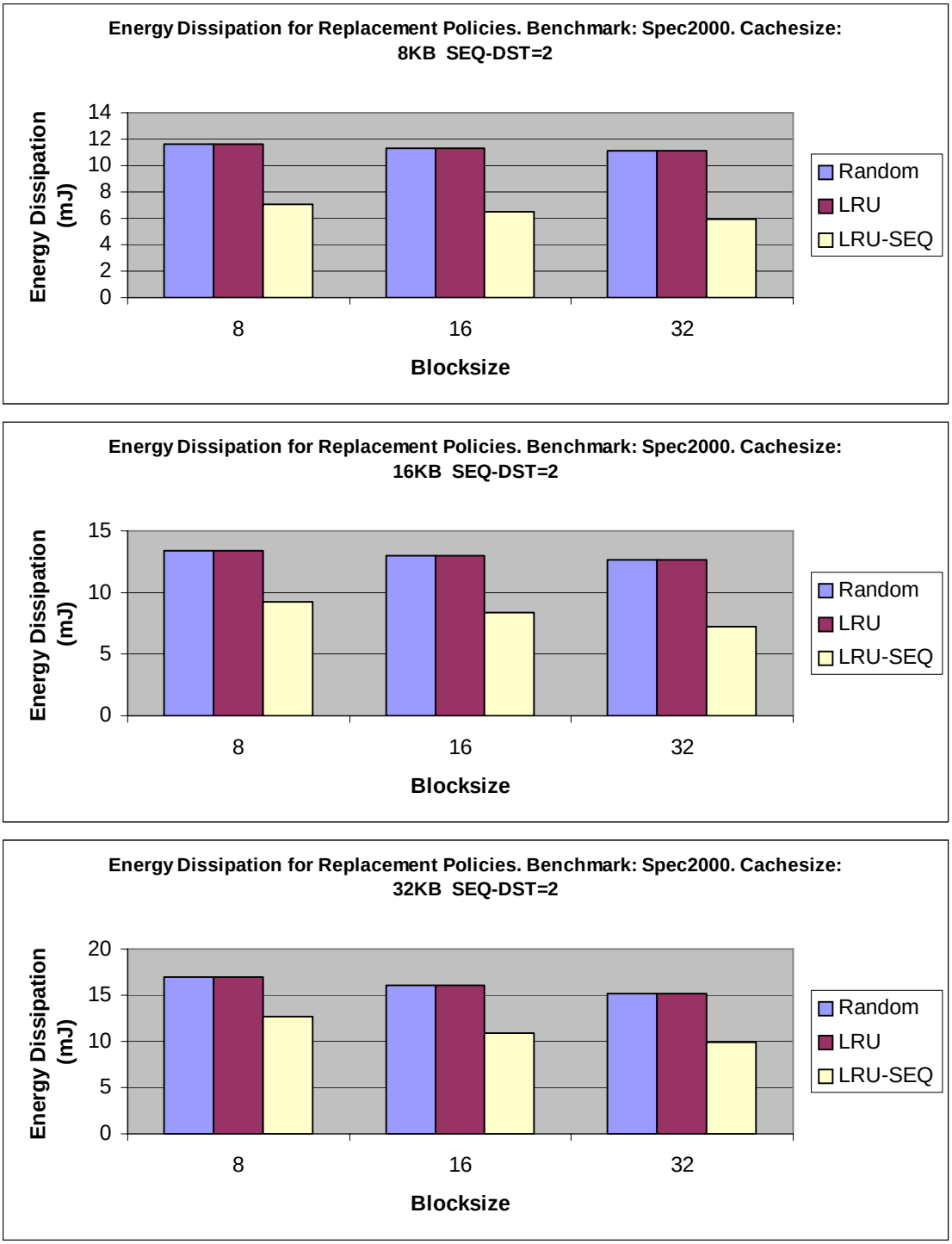


Figure 6.4 Energy Savings comparison for Replacement Policies

Figures 6.3 and 6.4 show that LRU-SEQ generates significant energy savings without causing considerable loss of performance. The amount of energy dissipated by caches (8KB, 16KB, and 32KB) increases with cache size increase for all replacement policies. Also a larger block size cache dissipates less energy than a smaller block sized cache of same cache size, which is an expected behavior. However, the LRU-SEQ replacement policy dissipates significantly less energy than the LRU and random replacement policies. The reason for such reduction in energy dissipation is the transitioning of inactive cache banks to a reduced voltage state (sleep state), which saves leakage energy. As shown by the example in Chapter 4, transitioning inactive cache banks saves energy, which is reflected in the plots shown in Figure 6.4. Figure 6.5 shows the average energy savings of the LRU-SEQ over the LRU. The following formula is used to calculate the average energy savings of the LRU-SEQ policy over the LRU replacement policy:

$$Avg.\ Savings\ of\ LRU\text{-}SEQ\ over\ LRU = 100*(Energy_{LRU} - Energy_{LRU\text{-}SEQ})/Energy_{LRU} \quad (3)$$

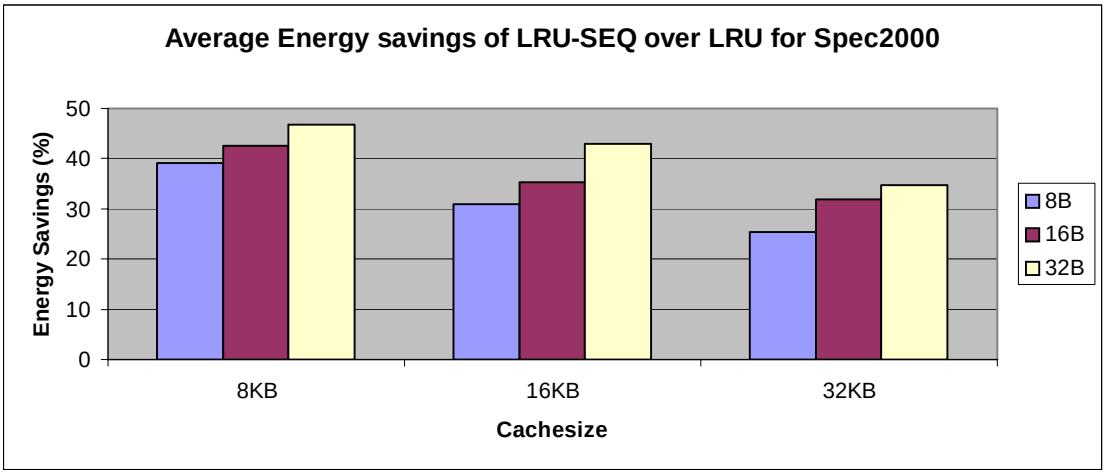


Figure 6.5 Average energy savings of the LRU-SEQ over the LRU

The 2-way LRU-SEQ saves on average a maximum of 46% for 8KB cache size and 32-byte block size cache architecture over the 2-way LRU.

6.5 LRU-SEQ Miss-Rate Comparison for different SEQ-DST values

The previous section observed the miss-rates and energy dissipation values achieved by the LRU-SEQ replacement policy with the SEQ-DST set at 2. This section shows the impact of varying SEQ-DST parameter on the miss-rate. SEQ-DST forces the cache line fills based on spatial locality criterion. If this parameter is set too high, then the forced fills start to cause conflict misses, which results in a high miss-rate. This section simulates the SEQ-DST with values at 2, 4, 8, and 16, and then plots the results.

As predicted, an increase in the SEQ-DST parameter causes an increase in miss-rates. Larger cache sizes generate fewer misses than smaller cache sizes. The plots in Figure 6.6 show an increase in miss-rate for each increase in the SEQ-DST parameter. An increase in miss-rate occurs when the SEQ-DST increases from 2 to 4 to 8, but the rate of increase is small. This behavior can be attributed to the fact that spatial locality distribution for each benchmark program is different. Hence, an increase in miss-rate due to an increase in the SEQ-DST does not occur for all benchmarks for exactly the same cache configuration. Figure 6.6 shows an increase in miss-rates for each increase in the SEQ-DST follows the pattern for increase in block sizes as well. The highest miss-rate occurs for the cache with the fewest number of cache lines per bank—8KB cache size and 32B block size—among the cache configurations simulated.

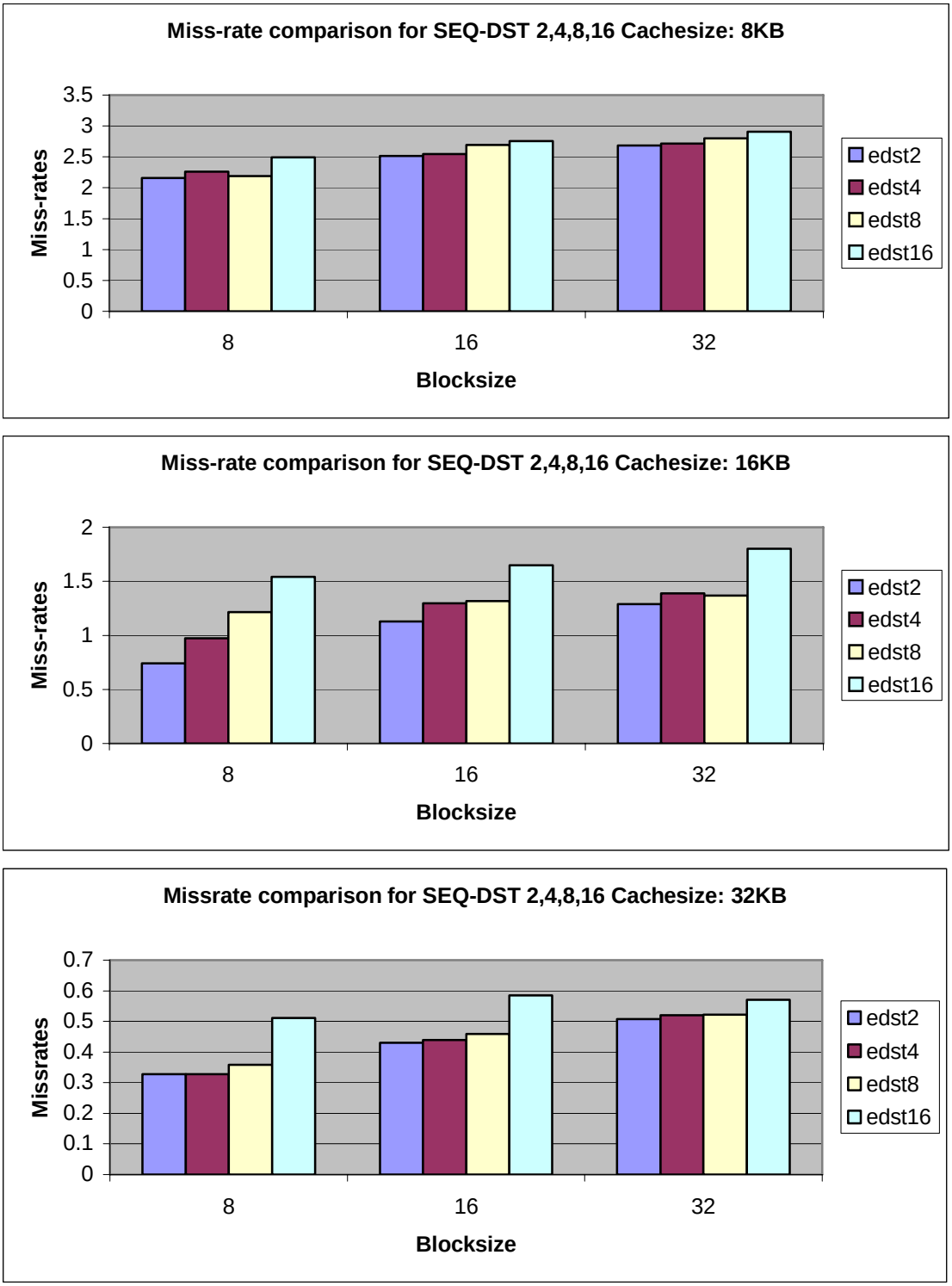


Figure 6.6 Miss-rates comparison for LRU-SEQ SEQ-DSTs

The following graphs show the miss-rates generated by some individual benchmark programs for each SEQ-DST value.

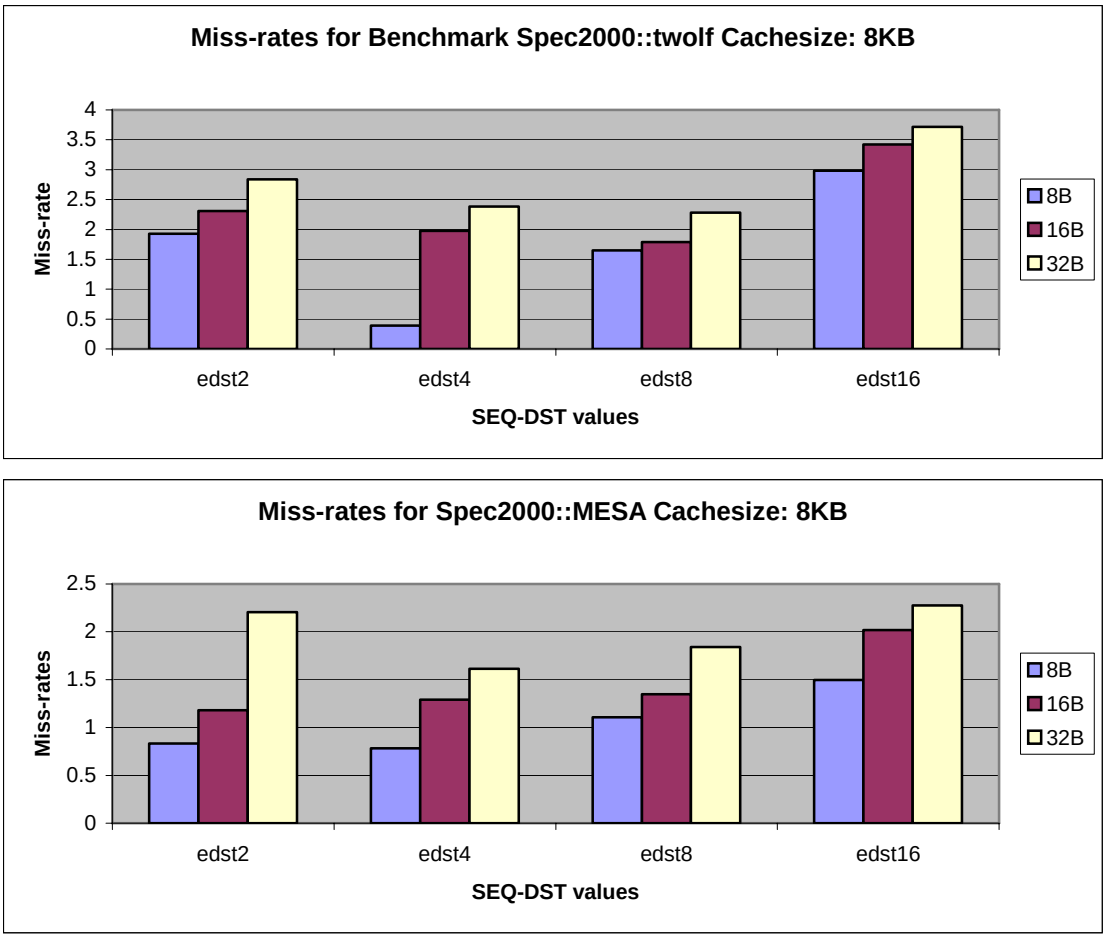


Figure 6.7 Miss-rate vs. SEQ-DST for twolf and mesa

Figure 6.7 shows two benchmark programs with the SEQ-DST parameter affecting miss-rates. The first plot shows the program twolf from SPEC2000 benchmark suite executed on a cache configuration of 8KB cache size and block sizes 8, 16, and 32 bytes. When the SEQ-DST is increased from 2 to 4, the miss-rate actually decreases for the 8-byte configuration, which can be inferred as better exploitation of spatial locality. But as

the distance increases, the miss-rates immediately increase. The same phenomenon occurs for another benchmark program, mesa. For the 16B block size characteristic of the mesa program, SEQ-DST=2 generates the least miss-rate, and when SEQ-DST increases, miss-rate increases. The entire benchmark suite, on average, exhibits this characteristic pattern of monotonous decrease in miss-rates as the SEQ-DST increases until SEQ-DST criterion itself causes more misses. The plots in Figure 6.6 show, on average, a maximum performance loss of up to 50% when the SEQ-DST is increased from 2 to 16.

6.6 LRU-SEQ Energy Savings Comparison for different SEQ-DST values

The previous section proves that an increase in the SEQ-DST causes an increase in miss-rates, but it is different for different benchmark programs. Since the purpose of the SEQ-DST parameter is to control the active and sleep states of cache banks, larger SEQ-DST values force cache fills to a currently active bank. Therefore, it invariably reduces the number of bank transitions and saves the transition energy of the active bank and leakage energy of the inactive bank. Because of this reduced number of transitions, an increase in the SEQ-DST parameter generates increased energy savings albeit at the expense of increased miss-rates. Figure 6.8 shows the average energy dissipation values for SPEC2000 benchmark suite for cache sizes of 8, 16, 32 KB and block sizes of 8, 16, and 32 bytes for the 2-way SA/LRU-SEQ cache architecture.

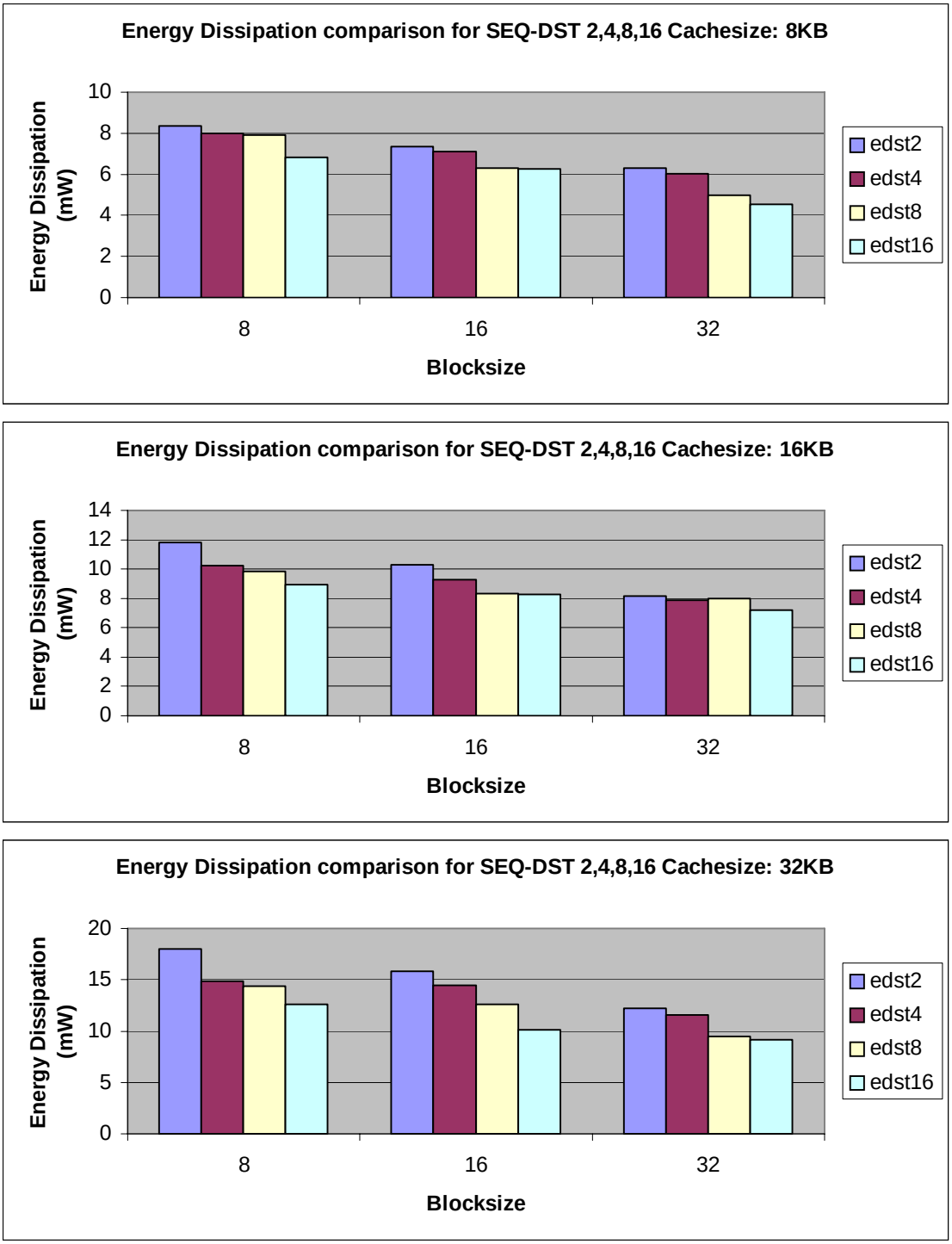


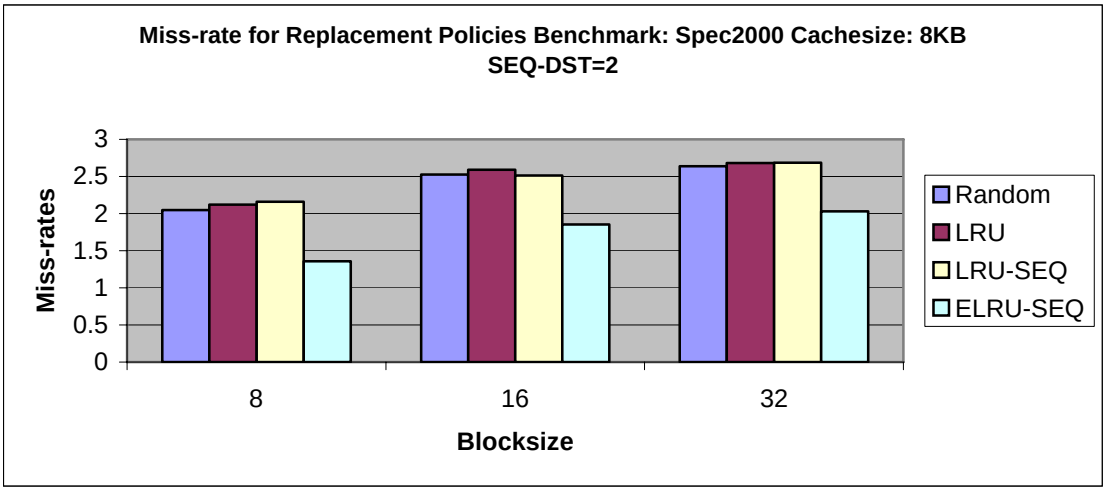
Figure 6.8 Energy Dissipation comparison for different SEQ-DST values

The plots in Figure 6.8 show that for the same cache size (8KB, 16KB, and 32KB) and block size (8B, 16B, and 32B), the energy dissipated by the cache for each increasing SEQ-DST (2, 4, 8, 16) progressively decreases. Also, for the same cache size, energy dissipation decreases with an increase in block size. The above observations lead to the conclusion that increases in SEQ-DST for the same cache configuration will force the LRU-SEQ policy to reduce bank transitions and keep the cache bank's state the same for longer durations, which generates energy savings. Increasing the block size decreases the number of lines per bank. Consequently, the number of cache lines that satisfies the SEQ-DST criterion increases, which results in reduced transitions that can be seen in increased energy savings. On average, energy savings of 33% occurs when SEQ-DST changes from 2 to 16.

6.7 Miss-rate Comparison between LRC and other Cache Architectures

This section compares the performance of conventional cache architectures with the 2-way LRC cache architecture. The 2-way LRC cache architecture, as explained in Chapter 3, has xor mapping functions for mapping its memory accesses to the cache address space and the ELRU-SEQ replacement policy for handling data replacement. Previous sections have proven that the xor mapping is better than the conventional mapping functions, such as the 2-way SA and the direct mapped. Also, the LRU-SEQ performs as well as the LRU replacement policy in terms of miss-rates and generates net energy savings when the SEQ-DST parameter is low. It can also generate increased energy savings when the SEQ-DST parameter is set high, but this high setting sacrifices miss-rate performance slightly.

By combining these two policies—the xor mapping function and the ELRU-SEQ replacement policy - the new 2-way LRC architecture achieves better miss-rate performance than the 2-way SA/LRU-SEQ. Energy savings are also significantly greater than what the 2-way SA/LRU generates. This result is an improvement over the LRU-SEQ policy that, while generating energy savings, does not improve LRU's miss-rates. Figure 6.9 shows the performances of the 2-way LRC cache architecture, which employs the 2-way xor mapping function and the LRU-SEQ replacement policy, and the 2-way SA/LRU, 2-way SA/random, and 2-way SA/LRU-SEQ cache architectures in terms of miss-rates.



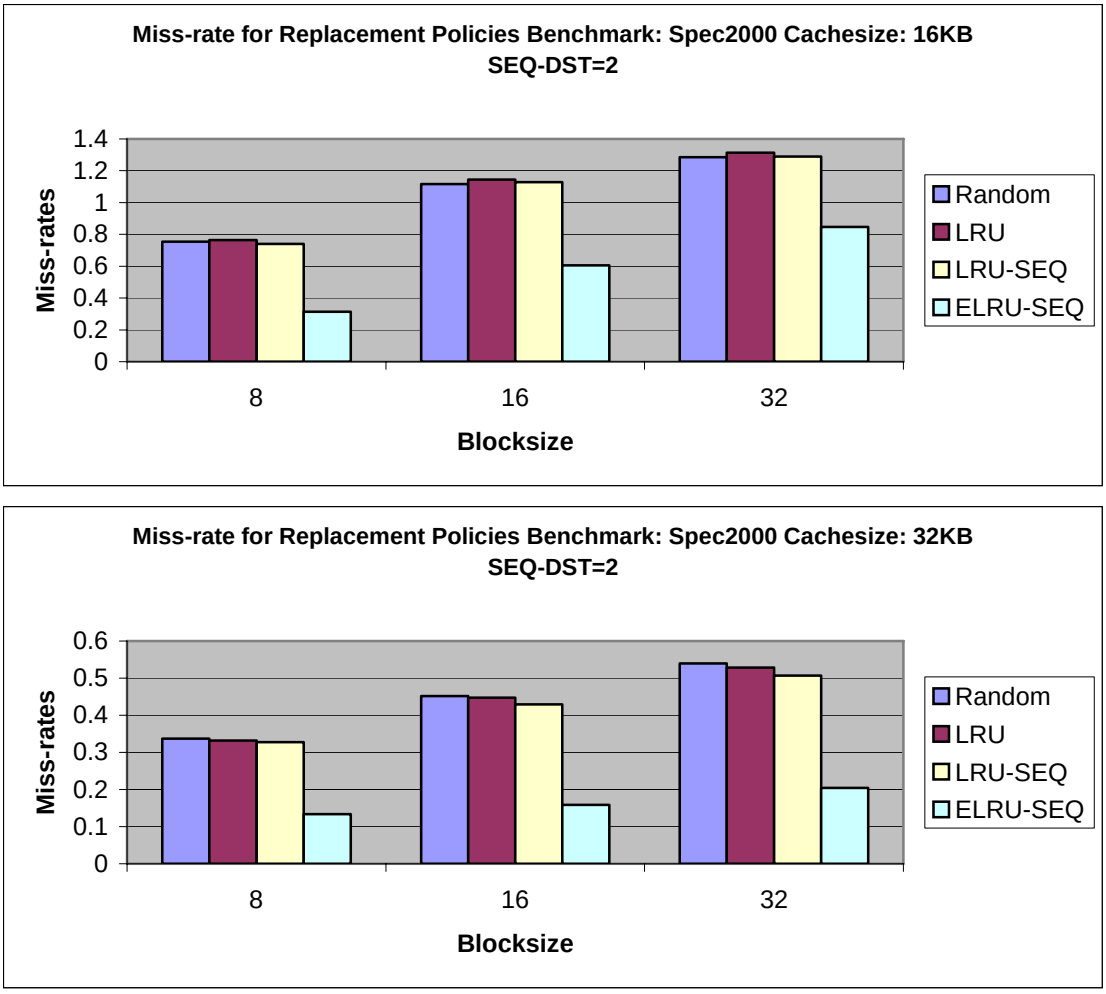


Figure 6.9 Miss-rate comparison of LRC and other architectures

The plots in Figure 6.9 show that the 2-way LRC cache architecture generates better miss-rates when compared with the conventional and the 2-way SA/LRU-SEQ cache architectures. As expected, the xor mapping function distributes data across cache lines better than conventional mapping functions and hence generates better miss-rates for all the simulated cache and block sizes. Figure 6.10 shows the average savings for the 2-way LRC cache architecture over the 2-way SA/LRU.

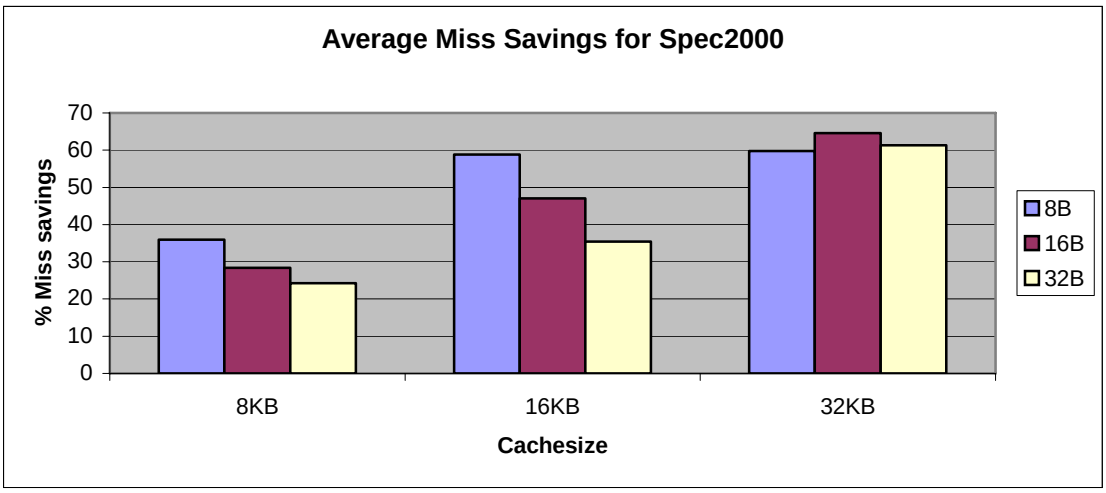


Figure 6.10 LRC vs. LRU miss-rate comparison

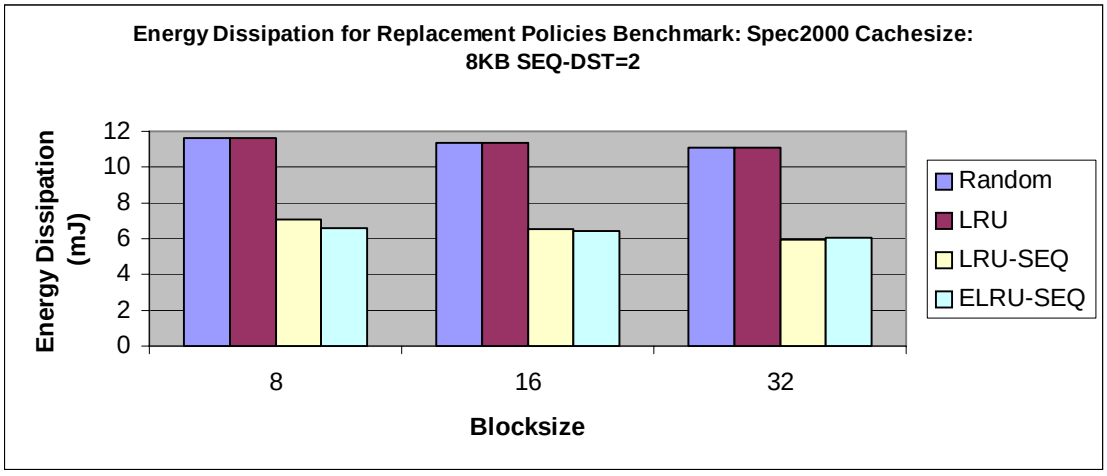
The formula used for calculating the average energy savings of 2-way LRC over LRU is:

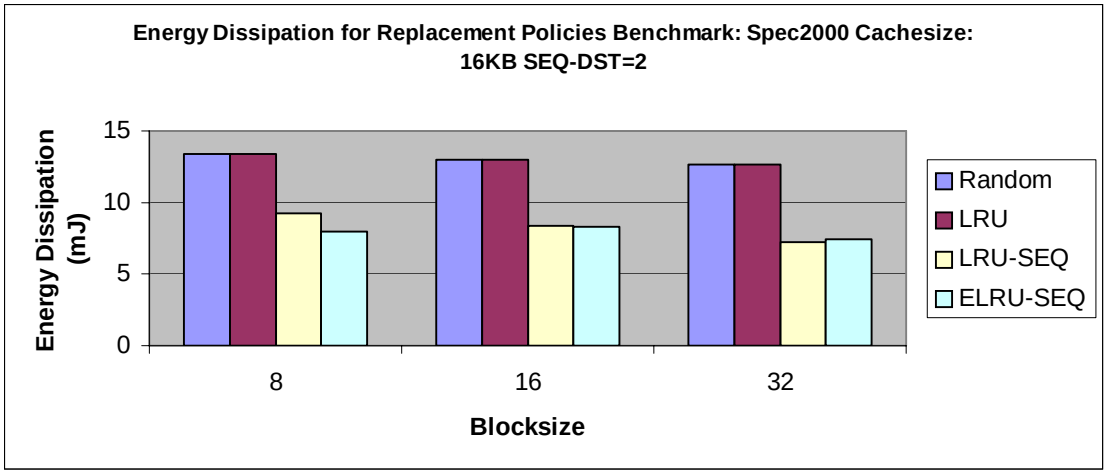
$$\%savings = 100*(Miss\text{-}rate_{LRU} - Miss\text{-}rate_{LRC})/Miss\text{-}rate_{LRU} \quad (4)$$

Figure 6.10 shows that miss-rate improvement on average of up to 64.56% can occur with the use of the 2-way LRC cache over the 2-way SA/LRU cache architecture. For cache sizes 8KB and 16KB, increasing the block size (8B, 16B and 32B) reduces the miss-rate improvement (% savings over LRU), which is due to a percentage increase in the number of cache lines per bank that satisfy the SEQ-DST criterion in the 2-way LRC cache. This trend of reduced miss-rate savings with increase in block size is not observed in 32KB cache for the 8B block size. This difference could be attributed to the greater availability of cache lines per bank within the 2-way SA/LRU cache. A decrease in miss-rate savings with increase in block size occurs again for the 16B and 32B block sizes for the 32KB cache size.

6.8 Energy Savings Comparison of LRC and other Cache Architectures

This section compares the energy dissipation in conventional cache architectures (2-way SA/LRU, 2-way SA/Random) and the 2-way SA/LRU-SEQ with that of the 2-way LRC cache architecture. Cache sizes of 8KB, 16KB and 32KB were used with block sizes of 8B, 16B, and 32B. Figure 6.11 shows that the 2-way LRC cache generates an average savings of up to 45.6% over the 2-way SA/LRU cache.





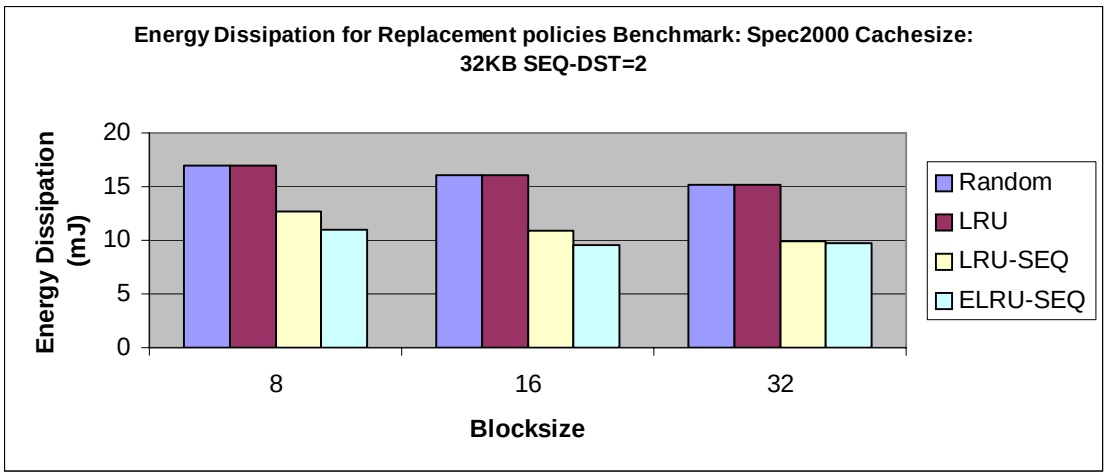


Figure 6.11 Energy Dissipation comparison of LRC with other cache architectures

The 2-way LRC cache dissipates less energy for smaller block sizes (8B and 16B), but for larger block sizes (32B), the energy dissipation is similar to that of the 2-way SA/LRU-SEQ cache. This difference can be attributed to the way that the LRU-SEQ implements a bank transition in case of a cache miss. When a cache miss occurs, the LRU-SEQ replacement policy first checks if the cache access's index lies within the spatial locality of the previous cache hit for that bank. If the index satisfies the spatial locality criterion, then the new data is placed in the active bank and current contents in that cache line are evicted. This operation does not result in a bank transition. However, if the index does not satisfy the spatial locality criterion, then the LRU replacement policy decides which bank the new data should be placed in. For the 2-way SA/LRU-SEQ cache architecture, if the current active cache line generates a conflict miss, then the new data is placed in the next bank, which results in a bank transition. The xor mapping function, due to its better data dispersion capability, offers the LRU replacement policy cache locations

in a current active bank with less probability of conflict misses (as proven in sections 1 and 2) than the 2-way SA mapping function. Due to this higher availability of cache lines per bank, the 2-way LRC successfully finds empty cache lines for data placement more often than the 2-way SA/LRU-SEQ. Consequently, the 2-way LRC cache generates fewer bank transitions and, hence, less energy dissipation.

An increase in the block size in a 2-way LRC cache increases the number of cache lines that satisfy the SEQ-DST criterion. Because of increased block size, the LRU-SEQ replacement policy is used more frequently, resulting in reduced transitions and so the 2-way LRC cache performs similar to that of a 2-way SA/LRU-SEQ cache in terms of energy dissipation. Since xor mapping function distributes data more effectively among the cache lines, the miss-rates observed for the 2-way LRC cache are significantly better than the 2-way LRU-SEQ cache.

Figure 6.12 shows the average energy savings generated by the 2-way LRC architecture over the 2-way SA/LRU. The formula used for calculating the average energy savings is:

$$Avg.\ Savings\ of\ LRC\ over\ LRU = 100*(Energy_{LRU} - Energy_{LRC})/Energy_{LRU} \quad (5)$$

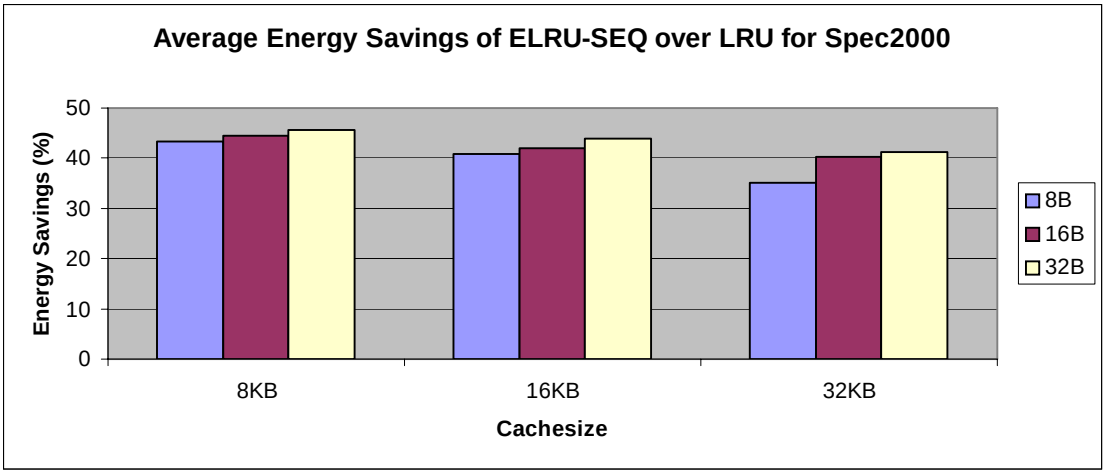
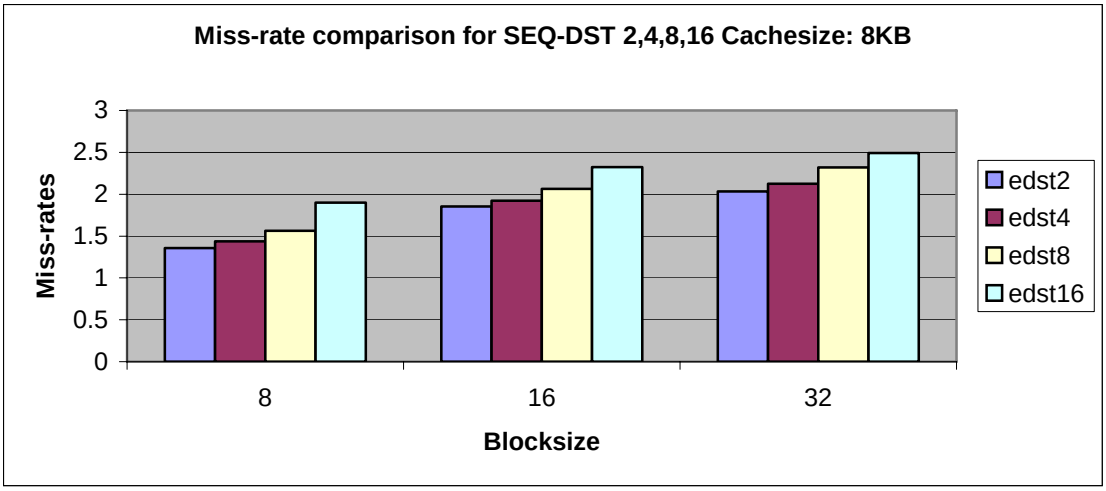
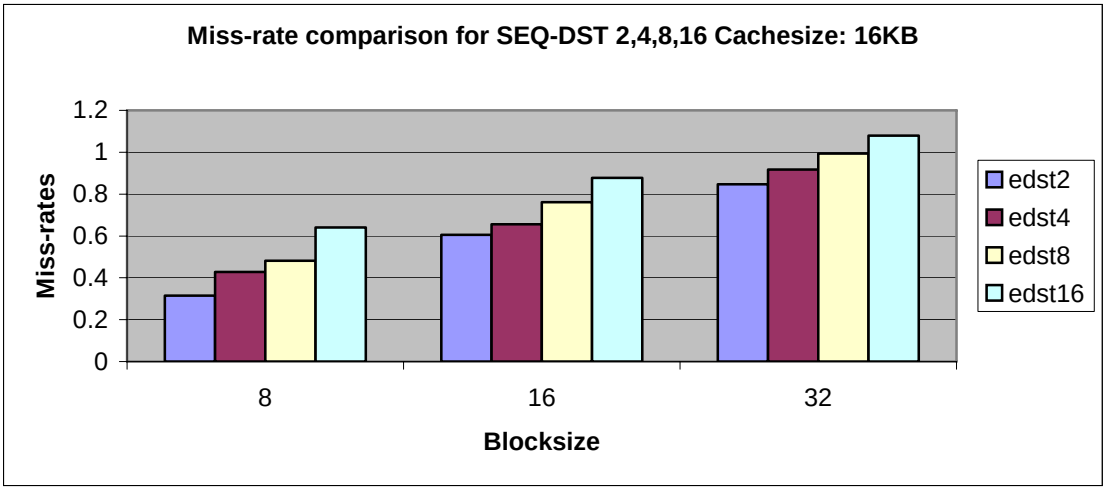


Figure 6.12 Average LRC energy savings over LRU

6.9 LRC Miss-rate Comparison for different SEQ-DST Values

This section compares the performance of the 2-WAY LRC architecture with the multiple SEQ-DST values. Cache size of 8, 16, 32KB and block size of 8, 16, 32B are used for cache configuration, and the SEQ-DST values of 2, 4, 8, 16 are used for simulations. The same pattern of increasing miss-rates with an increase in SEQ-DST occurs here as well. This correlation was expected, since constraining cache fills to a current active bank would eventually generate conflict misses and, consequently, an increase in miss-rates. Figure 6.13 shows this general trend of loss in performance due to an increase in SEQ-DST from 2 to 16.The maximum performance loss is 53.78%. As with the LRU-SEQ, a general increase in miss-rates cannot be attributed to an increase in miss-rates for all benchmarks at the same cache configurations and SEQ-DST values.


Miss-rate comparison for SEQ-DST 2,4,8,16 Cachesize: 8KB
Miss-rates
3
2.5
2
1.5
1
0.5
0
8
16
32
Blocksize
edst2
edst4
edst8
edst16


Miss-rate comparison for SEQ-DST 2,4,8,16 Cachesize: 16KB
Miss-rates
1.2
1
0.8
0.6
0.4
0.2
0
8
16
32
Blocksize
edst2
edst4
edst8
edst16

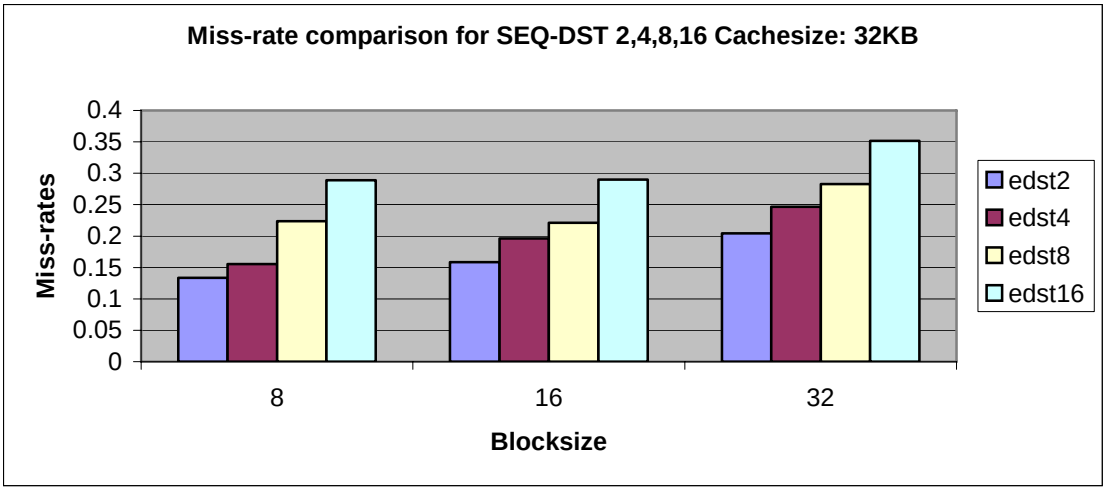
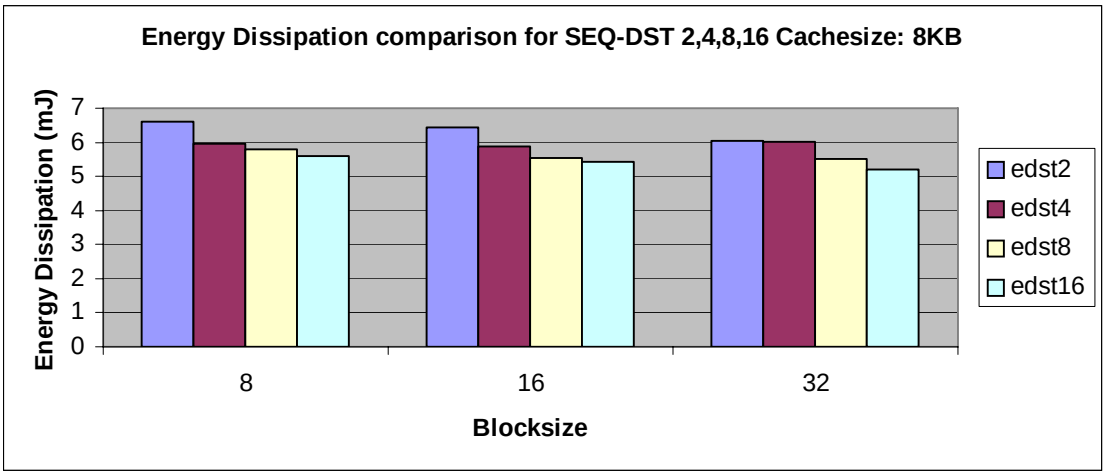


Figure 6.13 LRC Miss-rate vs. SEQ-DST

6.10 LRC Energy Savings Comparison for different SEQ-DST values

This section shows the performance of the 2-WAY LRC cache architecture in terms of energy dissipation when using different values of SEQ-DST. Cache size of 8, 16, 32KB and block size of 8, 16, 32B are used for the cache configuration and SEQ-DST of 2, 4, 8, 16 are used for the simulations.



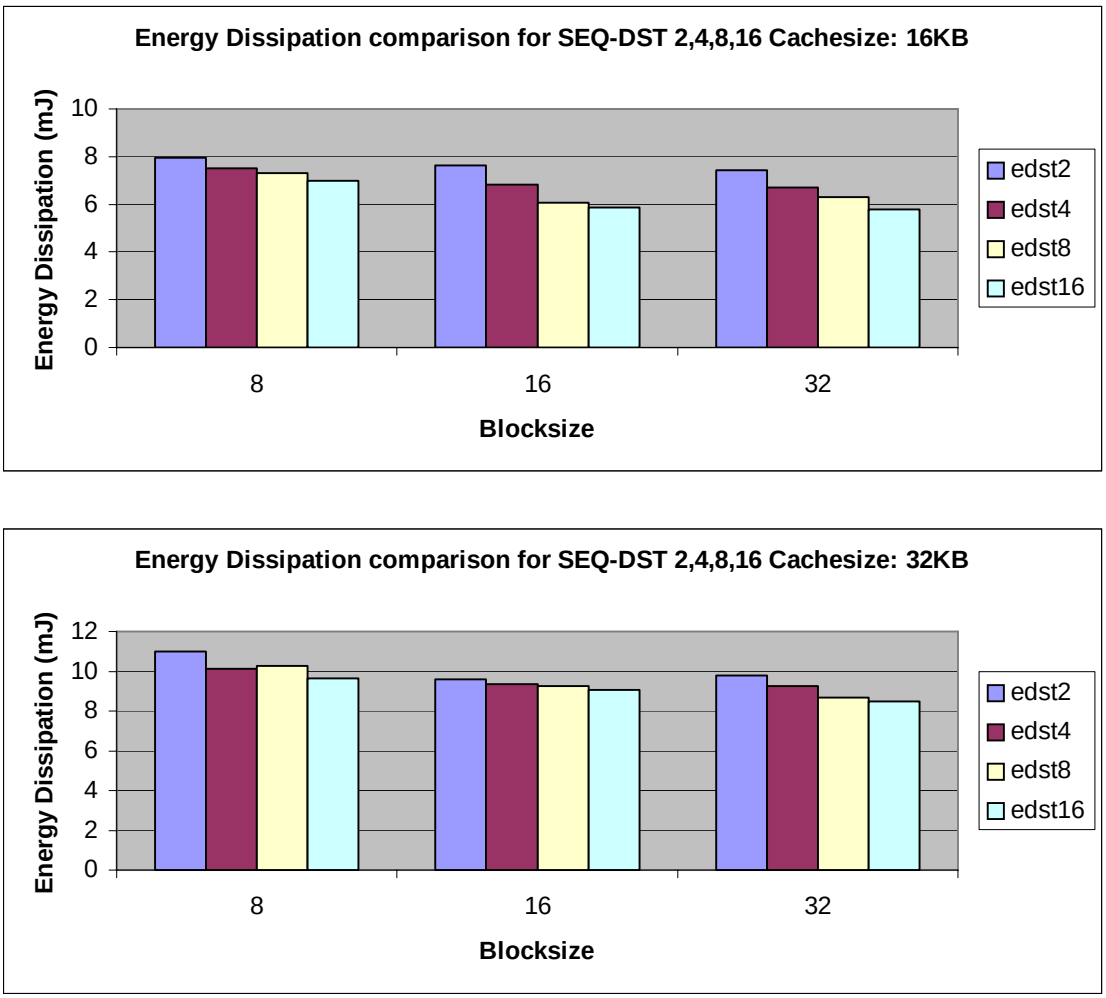


Figure 6.14 LRC Energy Dissipation with varying SEQ-DST values

Figure 6.14 shows that energy dissipation decreases with an increase in SEQ-DST value. The average energy savings with a maximum value of 21.55% occurs when SEQ-DST is increased from 2 to 16.

CHAPTER VII

CONCLUSIONS AND FUTURE WORK

7.1 Summary of Results

The goal of this research is to develop a cache architecture that will reduce cache miss-rate and also improve power dissipation by controlling leakage power loss. The cache miss-rate and power dissipation are the performance metrics used for this research. An optimal cache scheme would maintain the miss-rates while reducing power dissipation. This thesis shows the drawbacks of existing mapping functions and replacement policies and proposes new cache architecture LRC. Using unconventional mapping xor mapping functions and ELRU-SEQ replacement policy achieved better miss-rates and power savings.

Simulation results show that 2-way xor-mapping function generates miss-rates comparable to a 4-way SA mapping. ELRU-SEQ replacement policy generates miss-rates similar to that of LRU replacement policy for SEQ-DST value of 2 while saving power up to 46%. By increasing SEQ-DST value from 2 to 16, a power savings of up to 33% are seen while miss-rate performance reduces by 50%. Using xor mapping and ELRU-SEQ replacement policies together, the LRC cache architecture has been able to generate miss-rate and also power dissipation improvement. A miss-rate performance increase of 64.56% and power dissipation savings of 45.6% has occurred, on average.

With increase of SEQ-DST from 2 to 16, power savings are improved further by 21.55%. But with increased SEQ-DST, miss-rate performance reduces by 53.78% on average.

The disadvantages of the LRC architecture is the increase in access times due to the addition of xor logic for xor mapping function and the cost of extra hardware needed to implement the LRC architecture in terms of additional gates. As explained in chapter 3, this increase in access times amounts to one xor gate delay. With power dissipation in advanced processors becoming a major problem, the advantages for this cache architecture far outweigh the disadvantages.

7.2 Future Work

Currently, the LRC cache architecture is developed for uni-processors. Further extensions in future could develop LRC architecture for multi-processor environments. In this research, the SEQ-DST parameter was kept constant through out the simulation for each benchmark. However, the spatial and temporal localities of programs vary during execution. An evaluation of this architecture by dynamically changing the SEQ-DST parameter by observing conflict misses can improve LRC performance. More sophisticated mapping mechanisms for *f0* and *f1* should be investigated to improve conflict miss-rates. Additionally, an attempt to reduce the hardware used by LRC can be made in order to reduce the implementation cost. Also, extended evaluation of LRC for different hardware implementation technologies must occur.

REFERENCES

[1] D. Kudithipudi, S. Petko, E. John, "Cache leakage power analysis in embedded applications," *Proceedings of the 47th Midwest Symposium on Circuits and Systems*, Hiroshima, Japan, 2004, pp. 517-520.

[2] J. Montanaro, R. T. Witek, K. Anne, A. J. Black, E. M. Cooper, D. W. Dobberpuhl, P. M. Donahue, J. Eno, W. Hoeppner, D. Kruckemyer, T. H. Lee, P. C. M. Lin, L. Madden, D. Murray, M. H. Pearce, S. Santhanam, K. J. Snyder, R. Stephany, S. C. Thierauf, "A 160-MHz, 32-b, 0.5-W CMOS RISC Microprocessor," *IEEE Journal of Solid-State Circuits*, Vol. 31, Issue 11, Nov. 1996, pp. 1703 - 1714

[3] *Digital Integrated Circuits*, J. M. Rabaey, Englewood Cliffs, NJ, Prentice-Hall, 1996.

[4] S. Borkar, "Design challenges of technology scaling," *IEEE Micro*, vol.19, Issue 4, July 1999, pp.23-29,

[5] Semiconductor Industry Association. 2005. International Technology Roadmap for Semiconductors 2005 edition. http://www.itrs.net/Common/2005ITRS/Home2005.htm. (Accessed: January 2006)

[6] J. T. Kao, A. P. Chandrakasan, "Dual-threshold voltage techniques for low-power digital circuits", *IEEE Journal of Solid-State Circuits*, Vol. 35, Issue 7, July 2000, pp. 1009 – 1018

[7] M. Powell, Yang Se-Hyun, B. Falsafi, K. Roy, T. N. Vijaykumar, "Gated-V/sub dd/: A circuit technique to reduce leakage in deep-submicron cache memories", *Proceedings of the 2000 International Symposium on Low Power Electronics and Design*, Rapallo, Italy, 2000, pp. 90-95.

[8] Seongmoo Heo, K. Barr, M. Hamplon, K. Asanovic, "Dynamic fine-grain leakage reduction using leakage-biased bitlines," *Proceedings of the 29th annual international symposium on computer architecture*, Anchorage, AK, May 2002, pp. 137 – 147.

[9] Todd C. Mowry, Monica S. Lam, Anoop Gupta, "Design and evaluation of a compiler algorithm for prefetching," *Proceedings of the architectural support for programming languages and operating systems V*, Boston, MA, October 1992, pp. 62–73.

[10] P. Kalla, X. S. Hu, J. Henkel, "LRU-SEQ: A Novel Replacement Policy for Transition Energy Reduction in Instruction Caches", *Proceedings of the International Conference on Computer Aided Design*, San Jose, CA, November 2003, pp. 518-522.

[11] F. Bodin, A. Seznec, "Skewed associativity improves program performance and enhances predictability," *IEEE Transactions on Computers*, Vol. 46, Issue 5, May 1997, pp. 530 – 544

[12] Premkishore Shivakumar, Norman P. Jouppi. 2001. CACTI 3.0: An Integrated Cache Timing, Power, and Area Model. http://www.hpl.hp.com/research/mmsl/publications/adv/cacti3.pdf. (Accessed: September 2005)

[13] Seznec, Andre, Jerome Hedouni. 11 September 1997. The CACHESKEW simulator. http://www.irisa.fr/caps/PROJECTS/Architecture/CACHESKEW.html (Accessed: October 2005)

[14] Simplescalar official website. http://www.simplescalar.com. (Accessed: October 2005)

[15] M. J. Flynn, P. Hung, "Microprocessor design issues: Thoughts on the road ahead," *IEEE Micro*, Vol. 25, Issue 3, May 2005, pp. 16 – 31

[16] SPEC official website. http://www.spec.org/cpu2000/. (Accessed: October 2005)

[17] K. Roy, S. Mukhopadhyay, H. Mahmoodi-Meimand, "Leakage current mechanisms and leakage reduction techniques in deep-submicrometer CMOS circuits," *Proceedings of the IEEE*, vol. 91, Issue 2, Feb 2003, pp. 305-327

[18] Nam Sung Kim, Krisztian Flautner, David Blaauw, Trevor Mudge, "Drowsy Instruction Caches: Leakage Power Reduction using Dynamic Voltage Scaling and Cache Sub-bank Prediction," *Proceedings of the 35th Annual IEEE/ACM International Symposium on Microarchitecture*, Istanbul, Turkey, November 2002, pp. 219 – 230

[19] Intel Corp. April 19 2005. Moore's law 40th anniversary. http://www.intel.com/pressroom/kits/events/moores_law_40th/. (Accessed: December 2005)

[20] Q. G. Samdani, M. A. Thornton, "Cache resident data locality analysis," *Proceedings of the 8th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, San Francisco, CA, August 2000, pp. 539 – 546

[21] N. S. Kim, T. Austin, D. Baauw, T. Mudge, K. Flautner, J. S. Hu, M. J. Irwin, M. Kandemir, V. Narayanan, "Leakage current: Moore's law meets Static Power," *IEEE computer*, Vol. 36, Issue 12, December 2003, pp. 68 – 75

[22] Patterson, David, and John L. Hennessy. 2005. Computer Organization and Design: The Hardware / Software Interface. San Francisco, CA: Morgan Kauffman Publishers

APPENDIX A

VERILOG IMPLEMENTATION OF 2-WAY LRC EXTRA HARDWARE

Cacti 3.2 and Cadence Spectre have generated the energy dissipation values for the hardware of the 2-way LRC architecture. Cacti 3.2 instantiates a generic cache model based on the input parameters like cache size, block size, associativity etc. Cadence Spectre netlists realize the additional hardware needed to provide the ELRU-SEQ control logic and transition circuits. Verilog HDL behavioral model simulates the functional aspect of the ELRU-SEQ control logic.

A. Algorithm

The algorithm used to generate *way_select* signal is below. Using this algorithm, a way_select occurs when locality criteria is not met and vice versa. This way_select signal controls the multiplexer that selects between LRU replacement bank and current active bank for data replacement.

```
if (current access > previous hit) {
        temp = previous hit + edst;
        if (current access > temp) {
                way_select = '1';
        }
        else {
                way_select = '0';
        }
}
else {
        temp = current access + edst;
if (previous hit > temp) {
                way_select = '1';
        }
        else {
                way_select = '0';
        }
}
```

B Verilog Code

```
/*
--- Verilog model to implement an 8-bit LRC extra hardware.
--- Author:    Saibhushan Musalappa,
               ECE Department,
               Mississippi State University
--- Latest Update: 3/12/06
*/
// main module
module elruseq (seqout, dff_out2, dff_out1, add_out, a, b, clk, reset, edst);
        output seqout;
        input[7:0] a, b;
        input[7:0] edst;
        input clk, reset;
        wire gt1, ngt1, gt2;
        wire[7:0] mux1, mux2;
        output[7:0] dff_out1, dff_out2, add_out;

        gteq_8_bit G1 (, gt1, a, b);
        not (ngt1, gt1);
        mux8 M1 (mux1, a, b, gt1);
        mux8 M2 (mux2, a, b, ngt1);
        reg8 R1 (dff_out1, mux1, clk, reset);
        reg8 R2 (dff_out2, mux2, clk, reset);
        adder8 A1 (add_out, dff_out1, edst);
        gteq_8_bit G2 (, gt2, dff_out2, add_out);
        reg1bit R3 (seqout, gt2, clk, reset);

endmodule

//mux declaration
module mux8 (y, a, b, sel);
        input[7:0]          a, b;
        input sel;
        output[7:0]         y;

        mux2 M7 (y[7], a[7], b[7], sel);
```

```
        mux2 M6 (y[6], a[6], b[6], sel);
        mux2 M5 (y[5], a[5], b[5], sel);
        mux2 M4 (y[4], a[4], b[4], sel);
        mux2 M3 (y[3], a[3], b[3], sel);
        mux2 M2 (y[2], a[2], b[2], sel);
        mux2 M1 (y[1], a[1], b[1], sel);
        mux2 M0 (y[0], a[0], b[0], sel);

endmodule

module mux2 (y, a, b, sel);
        input a, b, sel;
        output y;
        wire nsel, n1, n2;

        not (nsel, sel);
        nand (n1, a, nsel);
        nand (n2, b, sel);
        nand (y, n1, n2);

endmodule

//adder declaration
module adder8 (sum, a, b);
        input[7:0] a, b;
        output[7:0] sum;
        reg[7:0] sum_int;

        assign sum = sum_int;

        always @ (a or b)
                sum_int = a + b;

endmodule

// 8-bit comparator declaration
module gteq_8_bit (yeq, ygt, a, b);
```

```
    output yeq, ygt;
    input[7:0] a, b;
    wire m1gt, m1eq, m2gt, m2eq;
    wire p1;

    gteq_4_bit M1 (m1eq, m1gt, a[7:4], b[7:4]);
    gteq_4_bit M2 (m2eq, m2gt, a[3:0], b[3:0]);

    and (p1, m1eq, m2gt);
    and (yeq, m1eq, m2eq);
    or (ygt, m1gt, p1);
endmodule

module gteq_4_bit (yeq, ygt, a, b);
    output   yeq, ygt;
    input[3:0]          a, b;
    wire[3:0]           c, d, e, f;

    xor      (c[3], a[3], b[3]);
    xor      (c[2], a[2], b[2]);
    xor      (c[1], a[1], b[1]);
    xor      (c[0], a[0], b[0]);

    not (d[3], c[3]);
    not (d[2], c[2]);
    not (d[1], c[1]);
    not (d[0], c[0]);

    and (yeq, d[3], d[2], d[1], d[0]);

    not (e[3], b[3]);
    not (e[2], b[2]);
    not (e[1], b[1]);
    not (e[0], b[0]);

    and (f[3], a[3], e[3]);
    and (f[2], a[2], e[2], d[3]);
```

```
        and (f[1], a[1], e[1], d[3], d[2]);
        and (f[0], a[0], e[0], d[3], d[2], d[1]);

        or (ygt, f[3], f[2], f[1], f[0]);
endmodule

// 1-bit register declaration
module reg1bit (dout, din, clk, reset);
        input din, clk, reset;
        output dout;
        reg dreg;

        assign dout = dreg;

        always @ (posedge reset or negedge clk)
                begin
                        if (reset)
                                dreg <= 1'b0;
                        else
                                dreg <= din;
                end
endmodule

// 8-bit register declaration
module reg8 (dout, din, clk, reset);
        input[7:0] din;
        input clk, reset;
        output[7:0] dout;
        reg[7:0] dreg;

        assign dout = dreg;

        always @ (posedge reset or posedge clk)
                begin
                        if (reset)
                                dreg <= 8'b0;
                        else
```

```
                                        dreg <= din;
                        end
    endmodule


    // ---------- End of Module declarations ---------- //


/*
--- test bench for ELRU-SEQ control hardware
--- Author:          Saibhushan Musalappa,
                     ECE Department,
                     Mississippi State University
--- Latest Update: 7/05/06
*/

module test_elru_seq;

reg clk, reset;
reg [7:0] edst, a, b;
wire seqout;
wire [7:0] dff_out1, dff_out2, add_out;

elruseq dut (seqout, dff_out2, dff_out1, add_out, a, b, clk, reset, edst);

initial // clock generator
        begin
                clk = 0;
                forever #50 clk = !clk; // clock pulse 100 time units
        end

initial    // Test stimulus
  begin
    reset = 1;
    a = 45;
    b = 40;
    edst = 2;
    #25 reset = 0;
    #200 b = 42; a = 40;
    #300 a = 45;
    #700 a = 43;
    #1000 a = 38;
    #1300 a = 44;
    #1500 b = 47;
    #1700 b = 46;
    #2000 $finish;
  end

initial
    $monitor($stime,, reset,, clk,, edst,, a,, b,, dff_out1,, dff_out2,, add_out,, seqout);

endmodule

// ---------- End test bench ---------///
```

C Simulation Output

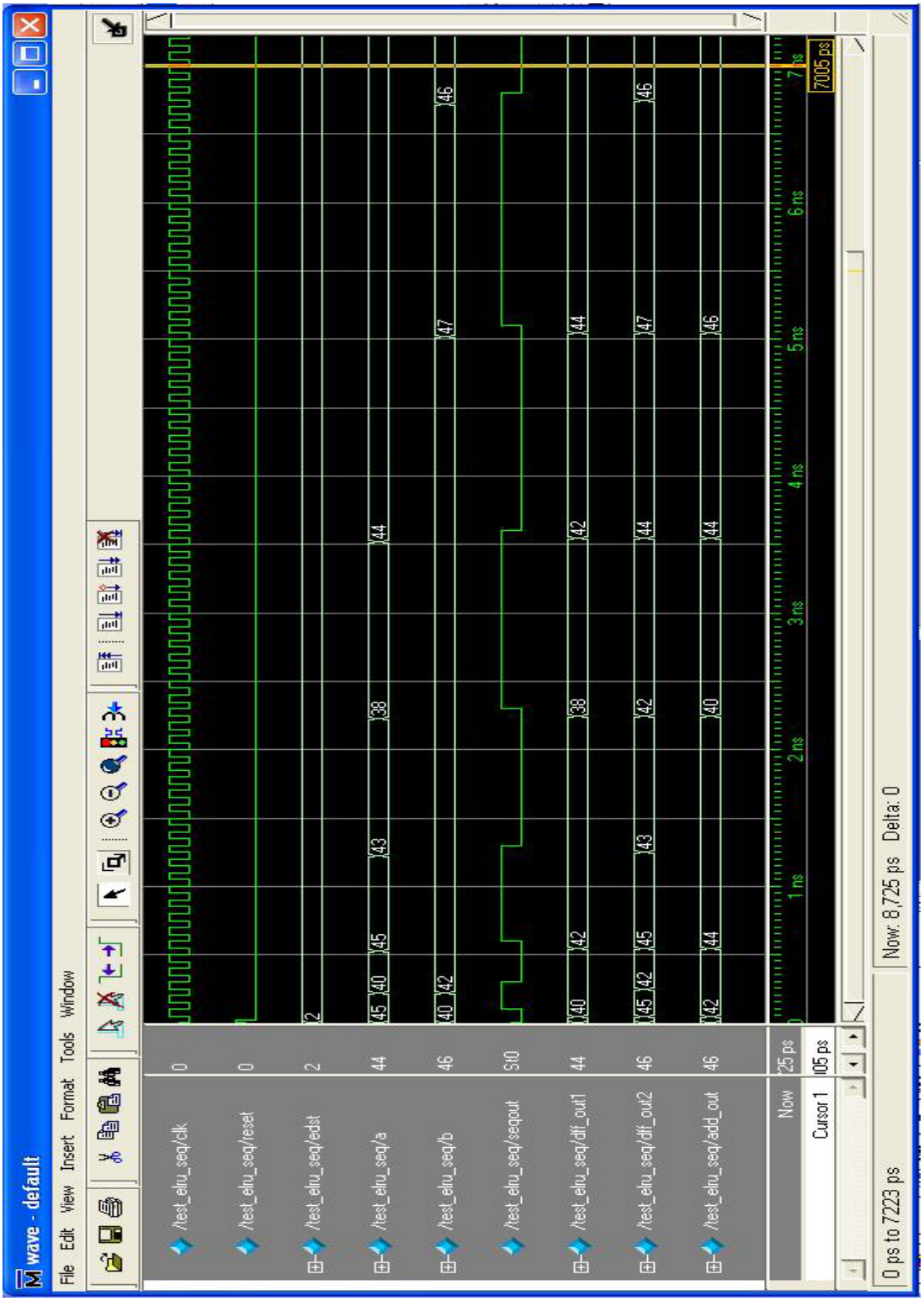


Figure A.1. ELRU-SEQ control hardware operation

Figure A.1 shows the operation of ELRU-SEQ control logic with SEQ-DST parameter set at 2. When previous hit index (a) and current memory reference (b) are within SEQ-DST value (2 for this simulation)—as when a=43 and b=42—the seqout signal goes low and new data is filled into current active bank; otherwise seqout signal goes high and data is replaced in the LRU_{bank}.

APPENDIX B

SPEC 2000 MISS-RATES AND ENERGY DISSIPATION RESULTS

B.1 Detailed 2-way LRC Miss-rates for Spec2000

Table B1. Miss-rates for DM, 2-way SA/LRU, 2-way SA/Random, 2-way LRC

CACHE SIZE: 8KB

8B	DM	Random	LRU	LRC
ammp	1.7128	0.7694	1.0574	0.0871
applu	0.3563	0.2369	0.2276	0.2052
apsi	0.0138	0.0135	0.0133	0.0135
art	0.0424	0.0186	0.0170	0.0173
equake	4.9230	0.9326	0.7808	0.1363
gap	5.0649	3.4025	3.5749	1.1465
gcc	22.7824	17.7205	18.1557	14.2465
mcf	0.0149	0.0150	0.0140	0.0143
mesa	4.3642	0.8428	0.9888	0.1783
mgrid	0.1162	0.0917	0.0836	0.0822
parser	0.0954	0.0828	0.0779	0.0745
twolf	18.3559	0.4313	0.4321	0.0935
A.M	**4.8202**	**2.0465**	**2.1186**	**1.3579**
16B				
ammp	2.1671	1.4625	1.9818	0.2499
applu	0.3416	0.2518	0.2425	0.1810
apsi	0.0111	0.0111	0.0106	0.0107
art	0.0391	0.0149	0.0139	0.0144
equake	5.0383	1.3147	1.1442	0.1809
gap	5.3365	4.0120	4.0127	2.5397
gcc	22.3880	19.8813	19.9823	18.5625
mcf	0.0124	0.0127	0.0116	0.0117
mesa	3.9015	1.2585	1.3426	0.1523
mgrid	0.1073	0.0835	0.0771	0.0761
parser	0.1083	0.0796	0.0756	0.0708
twolf	19.1463	1.9428	2.1767	0.2063
A.M	**4.8831**	**2.5271**	**2.5893**	**1.8547**

Table B1 (continued)

32B	DM	Random	LRU	LRC
ammp	2.9101	1.9265	2.4405	0.4242
applu	0.4249	0.2176	0.2067	0.1636
apsi	0.0083	0.0081	0.0078	0.0084
art	0.0454	0.0117	0.0105	0.0112
equake	5.0933	2.2683	2.2686	1.0031
gap	5.0594	4.1402	4.0624	3.5768
gcc	20.9611	18.7627	18.5875	18.0736
mcf	0.0093	0.0095	0.0088	0.0090
mesa	6.0048	1.8262	1.9067	0.7647
mgrid	0.0882	0.0695	0.0654	0.0618
parser	0.1158	0.0758	0.0694	0.0659
twolf	15.3225	2.3463	2.5372	0.2173
A.M	**4.6703**	**2.6385**	**2.6810**	**2.0316**

CACHE SIZE: 16KB

8B	DM	Random	LRU	LRC
ammp	0.3250	0.0864	0.0782	0.0771
applu	0.2316	0.2033	0.1946	0.1861
apsi	0.0134	0.0136	0.0132	0.0132
art	0.0363	0.0174	0.0169	0.0170
equake	1.6801	0.1738	0.1329	0.0332
gap	2.6133	0.8635	0.8687	0.2283
gcc	14.6052	7.0841	7.2503	2.8367
mcf	0.0139	0.0139	0.0133	0.0134
mesa	1.1531	0.2837	0.3283	0.1663
mgrid	0.0917	0.0804	0.0734	0.0748
parser	0.0704	0.0631	0.0595	0.0542
twolf	8.0510	0.1592	0.1412	0.0745
A.M	**2.4071**	**0.7535**	**0.7642**	**0.3146**

Table B1 (continued)

16B	DM	Random	LRU	LRC
ammp	0.3422	0.1409	0.1231	0.1202
applu	0.1952	0.1757	0.1704	0.1523
apsi	0.0107	0.0109	0.0105	0.0104
Art	0.0329	0.0145	0.0135	0.0137
equake	1.7949	0.1677	0.1271	0.0269
gap	3.0889	1.3876	1.3887	0.4347
gcc	14.7640	9.7914	9.8445	6.1417
mcf	0.0110	0.0109	0.0104	0.0105
mesa	1.1233	0.3828	0.4069	0.1388
mgrid	0.0793	0.0693	0.0631	0.0592
parser	0.0851	0.0588	0.0539	0.0480
twolf	8.0106	1.1822	1.5153	0.1095
A.M	**2.4615**	**1.1161**	**1.1439**	**0.6055**
32B				
ammp	0.9699	0.2081	0.1971	0.0511
applu	0.1610	0.1399	0.1329	0.1176
apsi	0.0078	0.0080	0.0076	0.0076
art	0.0392	0.0109	0.0102	0.0104
equake	1.8245	0.4990	0.5776	0.0212
gap	3.0606	1.6493	1.6315	0.9021
gcc	15.0234	10.8878	10.8762	8.5864
mcf	0.0080	0.0079	0.0074	0.0079
mesa	0.9787	0.5117	0.5354	0.3180
mgrid	0.0628	0.0548	0.0505	0.0467
parser	0.0891	0.0504	0.0467	0.0415
twolf	5.6380	1.4049	1.6795	0.0567
A.M	**2.3219**	**1.2861**	**1.3127**	**0.8473**

Table B1 (continued)

CACHE SIZE: 32KB

8B	DM	Random	LRU	LRC
ammp	0.0845	0.0788	0.0751	0.0754
applu	0.2178	0.1870	0.1840	0.1859
apsi	0.0133	0.0132	0.0132	0.0132
art	0.0173	0.0174	0.0169	0.0170
equake	0.7169	0.1738	0.1322	0.0323
gap	1.0972	0.2895	0.2965	0.1139
gcc	10.0959	2.8606	2.8567	0.5809
mcf	0.0135	0.0136	0.0133	0.0134
mesa	0.4914	0.1682	0.1623	0.3816
mgrid	0.0790	0.0746	0.0697	0.0700
parser	0.0584	0.0538	0.0505	0.0513
twolf	0.5034	0.1131	0.1116	0.0671
A.M	**1.1157**	**0.3370**	**0.3318**	**0.1335**
16B				
ammp	0.0711	0.0639	0.0600	0.0594
applu	0.1776	0.1478	0.1463	0.1487
apsi	0.0105	0.0105	0.0104	0.0104
art	0.0139	0.0143	0.0135	0.0136
equake	0.7108	0.1671	0.1261	0.0259
gap	1.2789	0.5472	0.5681	0.0980
gcc	10.4161	4.1091	4.1065	1.2242
mcf	0.0105	0.0106	0.0104	0.0104
mesa	0.4591	0.1369	0.1313	0.1332
mgrid	0.0658	0.0617	0.0567	0.0550
parser	0.0716	0.0452	0.0404	0.0408
twolf	0.4095	0.1041	0.0990	0.0829
A.M	**1.1413**	**0.4515**	**0.4474**	**0.1585**

Table B1 (continued)

32B	DM	Random	LRU	LRC
ammp	0.0563	0.0480	0.0447	0.0461
applu	0.1385	0.1127	0.1087	0.1091
apsi	0.0077	0.0076	0.0076	0.0076
art	0.0105	0.0110	0.0102	0.0103
equake	0.7039	0.1588	0.1194	0.0196
gap	1.3880	0.6274	0.6398	0.1553
gcc	9.8165	5.2236	5.1464	1.8866
mcf	0.0075	0.0076	0.0073	0.0075
mesa	0.3150	0.1013	0.0963	0.0978
mgrid	0.0501	0.0456	0.0423	0.0404
parser	0.0610	0.0344	0.0311	0.0293
twolf	0.4202	0.0931	0.0894	0.0445
A.M	**1.0813**	**0.5393**	**0.5286**	**0.2045**

B.2. Detailed 2-way LRC energy dissipation values for Spec2000

Table B2. Energy values for DM, 2-way SA/LRU, 2-way SA/Random, 2-way LRC

CACHESIZE: 8KB

8B	DM	LRU	LRC	Random
ammp	0.0042	0.0116	0.0062	0.0116
applu	0.0042	0.0116	0.0060	0.0116
apsi	0.0042	0.0116	0.0072	0.0116
art	0.0042	0.0116	0.0063	0.0116
equake	0.0042	0.0116	0.0069	0.0116
gap	0.0042	0.0116	0.0073	0.0116
gcc	0.0042	0.0116	0.0073	0.0116
mcf	0.0042	0.0116	0.0058	0.0116
mesa	0.0042	0.0116	0.0064	0.0116
mgrid	0.0042	0.0116	0.0058	0.0116
parser	0.0042	0.0116	0.0066	0.0116
twolf	0.0042	0.0116	0.0073	0.0116
A.M (mJ)	**4.2257**	**11.6474**	**6.6050**	**11.6474**

Table B2 (continued)

16B	DM	LRU	LRC	Random
ammp	0.0040	0.0113	0.0061	0.0113
applu	0.0040	0.0113	0.0060	0.0113
apsi	0.0040	0.0113	0.0057	0.0113
art	0.0040	0.0113	0.0062	0.0113
equake	0.0040	0.0113	0.0065	0.0113
gap	0.0040	0.0113	0.0068	0.0113
gcc	0.0040	0.0113	0.0069	0.0113
mcf	0.0040	0.0113	0.0078	0.0113
mesa	0.0040	0.0113	0.0061	0.0113
mgrid	0.0040	0.0113	0.0057	0.0113
parser	0.0040	0.0113	0.0067	0.0113
twolf	0.0040	0.0113	0.0068	0.0113
A.M (mJ)	**3.9620**	**11.3428**	**6.4463**	**11.3428**
32B				
ammp	0.0038	0.0111	0.0058	0.0111
applu	0.0038	0.0111	0.0059	0.0111
apsi	0.0038	0.0111	0.0056	0.0111
art	0.0038	0.0111	0.0066	0.0111
equake	0.0038	0.0111	0.0061	0.0111
gap	0.0038	0.0111	0.0066	0.0111
gcc	0.0038	0.0111	0.0065	0.0111
mcf	0.0038	0.0111	0.0056	0.0111
mesa	0.0038	0.0111	0.0060	0.0111
mgrid	0.0038	0.0111	0.0056	0.0111
parser	0.0038	0.0111	0.0060	0.0111
twolf	0.0038	0.0111	0.0064	0.0111
A.M (mJ)	**3.8491**	**11.1088**	**6.0431**	**11.1088**

Table B2 (continued)

CACHESIZE: 16KB

8B	DM	LRU	LRC	Random
ammp	0.0054	0.0134	0.0079	0.0134
applu	0.0054	0.0134	0.0071	0.0134
apsi	0.0054	0.0134	0.0096	0.0134
art	0.0054	0.0134	0.0071	0.0134
equake	0.0054	0.0134	0.0081	0.0134
gap	0.0054	0.0134	0.0085	0.0134
gcc	0.0054	0.0134	0.0097	0.0134
mcf	0.0054	0.0134	0.0067	0.0134
mesa	0.0054	0.0134	0.0078	0.0134
mgrid	0.0054	0.0134	0.0067	0.0134
parser	0.0054	0.0134	0.0073	0.0134
twolf	0.0054	0.0134	0.0089	0.0134
A.M	**5.3844**	**13.4247**	**7.9504**	**13.4247**
16B				
ammp	0.0051	0.0130	0.0073	0.0130
applu	0.0051	0.0130	0.0072	0.0130
apsi	0.0051	0.0130	0.0065	0.0130
art	0.0051	0.0130	0.0071	0.0130
equake	0.0051	0.0130	0.0081	0.0130
gap	0.0051	0.0130	0.0087	0.0130
gcc	0.0051	0.0130	0.0087	0.0130
mcf	0.0051	0.0130	0.0107	0.0130
mesa	0.0051	0.0130	0.0075	0.0130
mgrid	0.0051	0.0130	0.0116	0.0130
parser	0.0051	0.0130	0.0082	0.0130
twolf	0.0051	0.0130	0.0082	0.0130
A.M (mJ)	**5.0589**	**12.9751**	**8.3242**	**12.9751**

Table B2 (continued)

32B	DM	LRU	LRC	Random
ammp	0.0047	0.0127	0.0068	0.0127
applu	0.0047	0.0127	0.0066	0.0127
apsi	0.0047	0.0127	0.0092	0.0127
art	0.0047	0.0127	0.0081	0.0127
equake	0.0047	0.0127	0.0073	0.0127
gap	0.0047	0.0127	0.0072	0.0127
gcc	0.0047	0.0127	0.0081	0.0127
mcf	0.0047	0.0127	0.0063	0.0127
mesa	0.0047	0.0127	0.0070	0.0127
mgrid	0.0047	0.0127	0.0064	0.0127
parser	0.0047	0.0127	0.0083	0.0127
twolf	0.0047	0.0127	0.0078	0.0127
A.M (mJ)	**4.7337**	**12.6707**	**7.4313**	**12.6707**

CACHESIZE: 32KB

8B	DM	LRU	LRC	Random
ammp	0.0074	0.0169	0.0096	0.0169
applu	0.0074	0.0169	0.0103	0.0169
apsi	0.0074	0.0169	0.0142	0.0169
art	0.0074	0.0169	0.0093	0.0169
equake	0.0074	0.0169	0.0107	0.0169
gap	0.0074	0.0169	0.0122	0.0169
gcc	0.0074	0.0169	0.0141	0.0169
mcf	0.0074	0.0169	0.0085	0.0169
mesa	0.0074	0.0169	0.0107	0.0169
mgrid	0.0074	0.0169	0.0086	0.0169
parser	0.0074	0.0169	0.0118	0.0169
twolf	0.0074	0.0169	0.0118	0.0169
A.M (mJ)	**7.4046**	**16.9375**	**10.9824**	**16.9375**

Table B2 (continued)

16B	DM	LRU	LRC	Random
ammp	0.0069	0.0160	0.0094	0.0160
applu	0.0069	0.0160	0.0083	0.0160
apsi	0.0069	0.0160	0.0080	0.0160
art	0.0069	0.0160	0.0088	0.0160
equake	0.0069	0.0160	0.0105	0.0160
gap	0.0069	0.0160	0.0108	0.0160
gcc	0.0069	0.0160	0.0126	0.0160
mcf	0.0069	0.0160	0.0080	0.0160
mesa	0.0069	0.0160	0.0095	0.0160
mgrid	0.0069	0.0160	0.0081	0.0160
parser	0.0069	0.0160	0.0098	0.0160
twolf	0.0069	0.0160	0.0111	0.0160
A.M (mJ)	**6.8670**	**16.0431**	**9.5780**	**16.0431**

32B	DM	LRU	LRC	Random
ammp	0.0062	0.0152	0.0085	0.0152
applu	0.0062	0.0152	0.0081	0.0152
apsi	0.0062	0.0152	0.0133	0.0152
art	0.0062	0.0152	0.0113	0.0152
equake	0.0062	0.0152	0.0094	0.0152
gap	0.0062	0.0152	0.0113	0.0152
gcc	0.0062	0.0152	0.0109	0.0152
mcf	0.0062	0.0152	0.0076	0.0152
mesa	0.0062	0.0152	0.0088	0.0152
mgrid	0.0062	0.0152	0.0076	0.0152
parser	0.0062	0.0152	0.0104	0.0152
twolf	0.0062	0.0152	0.0101	0.0152
A.M (mJ)	**6.2051**	**15.1713**	**9.7669**	**15.1713**

B.3. Detailed 2-way LRC miss-rate vs. SEQ-DST values for Spec2000

Table B3. 2-way LRC miss-rate values for SEQ-DST = 2, 4, 8, and 16

SEQ-DST: 2

8KB	8B	16B	32B
ammp	0.0871	0.2499	0.4242
applu	0.2052	0.1810	0.1636
Apsi	0.0135	0.0107	0.0084
Art	0.0173	0.0144	0.0112
equake	0.1363	0.1809	1.0031
Gap	1.1465	2.5397	3.5768
Gcc	14.2465	18.5625	18.0736
Mcf	0.0143	0.0117	0.0090
mesa	0.1783	0.1523	0.7647
mgrid	0.0822	0.0761	0.0618
parser	0.0745	0.0708	0.0659
Twolf	0.0935	0.2063	0.2173
A.M	**1.3579**	**1.8547**	**2.0316**
16KB			
ammp	0.0771	0.1202	0.0511
applu	0.1861	0.1523	0.1176
Apsi	0.0132	0.0104	0.0076
Art	0.0170	0.0137	0.0104
equake	0.0332	0.0269	0.0212
Gap	0.2283	0.4347	0.9021
Gcc	2.8367	6.1417	8.5864
Mcf	0.0134	0.0105	0.0079
mesa	0.1663	0.1388	0.3180
mgrid	0.0748	0.0592	0.0467
parser	0.0542	0.0480	0.0415
Twolf	0.0745	0.1095	0.0567
A.M	**0.3146**	**0.6055**	**0.8473**

Table B3 (continued)

32KB	8B	16B	32B
ammp	0.0754	0.0594	0.0461
applu	0.1859	0.1487	0.1091
Apsi	0.0132	0.0104	0.0076
Art	0.0170	0.0136	0.0103
equake	0.0323	0.0259	0.0196
Gap	0.1139	0.0980	0.1553
Gcc	0.5809	1.2242	1.8866
Mcf	0.0134	0.0104	0.0075
mesa	0.3816	0.1332	0.0978
mgrid	0.0700	0.0550	0.0404
parser	0.0513	0.0408	0.0293
Twolf	0.0671	0.0829	0.0445
A.M	**0.1335**	**0.1585**	**0.2045**

SEQ-DST: 4

8KB			
ammp	0.2327	0.2482	0.6119
applu	0.2261	0.2016	0.1793
Apsi	0.0137	0.0106	0.0081
Art	0.0176	0.0330	0.0109
equake	0.0373	0.4487	1.1202
Gap	1.5537	2.7173	3.6652
Gcc	14.6144	18.5833	18.2648
Mcf	0.0143	0.0123	0.0090
mesa	0.2855	0.2804	1.1087
mgrid	0.0852	0.0755	0.0617
parser	0.0744	0.0738	0.0672
Twolf	0.0976	0.3882	0.3896
A.M	**1.4377**	**1.9227**	**2.1247**

Table B3 (continued)

16KB	8B	16B	32B
ammp	0.0793	0.1199	0.0797
applu	0.1878	0.1765	0.1274
Apsi	0.0133	0.0104	0.0077
Art	0.0170	0.0137	0.0105
equake	0.4397	0.0268	0.6043
Gap	0.4106	0.5500	0.9502
Gcc	3.5498	6.6066	8.6454
Mcf	0.0134	0.0107	0.0077
mesa	0.1684	0.1398	0.4297
mgrid	0.0749	0.0622	0.0488
parser	0.0561	0.0484	0.0408
Twolf	0.1282	0.1087	0.0592
A.M	**0.4282**	**0.6561**	**0.9176**
32KB			
ammp	0.0756	0.0602	0.0453
applu	0.1865	0.1474	0.1102
Apsi	0.0132	0.0104	0.0076
Art	0.0170	0.0137	0.0103
equake	0.0328	0.0259	0.0195
Gap	0.1692	0.1812	0.2497
Gcc	0.8943	1.5900	2.2662
Mcf	0.0135	0.0105	0.0075
mesa	0.2736	0.1340	0.0974
mgrid	0.0706	0.0563	0.0410
parser	0.0525	0.0413	0.0299
Twolf	0.0677	0.0834	0.0720
A.M	**0.1556**	**0.1962**	**0.2464**

Table B3 (continued)

SEQ-DST: 8

8KB	8B	16B	32B
ammp	0.2675	0.3845	0.5466
applu	0.2444	0.2316	0.2055
Apsi	0.0138	0.0106	0.0081
Art	0.0181	0.0238	0.0114
equake	0.0379	0.7606	1.9256
Gap	1.9952	2.8691	3.6551
Gcc	15.0681	18.8361	18.3699
Mcf	0.0145	0.0127	0.0092
mesa	0.5045	0.9147	1.7479
mgrid	0.0841	0.0803	0.0631
parser	0.3362	0.0711	0.0688
Twolf	0.1909	0.5907	1.2187
A.M	**1.5646**	**2.0655**	**2.3191**
16KB			
ammp	0.0799	0.0792	0.2272
applu	0.1935	0.1750	0.1268
Apsi	0.0133	0.0104	0.0076
Art	0.0171	0.0139	0.0105
equake	0.0345	0.5599	0.4274
Gap	0.5280	0.6778	1.1610
Gcc	4.5206	7.0211	9.0370
Mcf	0.0135	0.0107	0.0078
mesa	0.1689	0.3582	0.7118
mgrid	0.0759	0.0622	0.0493
parser	0.0577	0.0526	0.0419
Twolf	0.0760	0.1197	0.1207
A.M	**0.4816**	**0.7617**	**0.9941**

Table B3 (continued)

32KB	8B	16B	32B
ammp	0.6028	0.0604	0.0457
applu	0.1889	0.1521	0.1121
Apsi	0.0133	0.0104	0.0076
Art	0.0172	0.0138	0.0108
equake	0.0329	0.0264	0.0195
Gap	0.1985	0.1946	0.2794
Gcc	1.1476	1.7877	2.6918
Mcf	0.0136	0.0105	0.0076
mesa	0.2753	0.2423	0.0988
mgrid	0.0715	0.0577	0.0417
parser	0.0524	0.0418	0.0309
Twolf	0.0686	0.0574	0.0461
A.M	**0.2235**	**0.2213**	**0.2827**

SEQ-DST: 16

8KB	8B	16B	32B
ammp	0.5923	0.4352	1.4410
applu	0.2476	0.2463	0.2618
Apsi	0.0136	0.0106	0.0085
Art	0.0275	0.0151	0.1033
equake	0.5677	2.6460	2.4270
Gap	2.1778	2.9687	3.7359
Gcc	15.4110	19.1899	18.4876
Mcf	0.0150	0.0130	0.0097
mesa	0.9385	0.5895	1.9583
mgrid	0.0870	0.0818	0.0637
parser	0.0811	0.0755	0.0713
Twolf	2.6375	1.5997	1.3357
A.M	**1.8997**	**2.3226**	**2.4920**

Table B3 (continued)

16KB	8B	16B	32B
ammp	0.1688	0.0638	0.8733
applu	0.2023	0.1687	0.1297
Apsi	0.0134	0.0106	0.0076
Art	0.0172	0.0139	0.0104
equake	0.4417	0.4842	0.7332
Gap	0.7128	1.0164	1.2365
Gcc	5.0095	7.5030	9.2839
Mcf	0.0136	0.0108	0.0080
mesa	0.8869	0.6825	0.3935
mgrid	0.0788	0.0644	0.0499
parser	0.0606	0.0569	0.0454
Twolf	0.0779	0.4602	0.1736
A.M	**0.6403**	**0.8779**	**1.0787**
32KB			
ammp	0.6330	0.0603	0.2227
applu	0.1884	0.1478	0.1289
Apsi	0.0133	0.0104	0.0076
Art	0.0172	0.0139	0.0111
equake	0.2107	0.0264	0.1975
Gap	0.1915	0.3074	0.3684
Gcc	1.6010	2.3905	2.8745
Mcf	0.0137	0.0106	0.0076
mesa	0.4020	0.3519	0.2080
mgrid	0.0718	0.0574	0.0433
parser	0.0529	0.0426	0.0326
Twolf	0.0708	0.0583	0.1180
A.M	**0.2889**	**0.2898**	**0.3517**

B.4. Detailed 2-way LRC energy vs. SEQ-DST values for Spec2000

Table B4. 2-way LRC energy dissipation values for SEQ-DST = 2, 4, 8, and 16

SEQ-DST: 2

8KB	8B	16B	32B
ammp	0.0062	0.0061	0.0058
applu	0.0060	0.0060	0.0059
Apsi	0.0072	0.0057	0.0056
Art	0.0063	0.0062	0.0066
equake	0.0069	0.0065	0.0061
Gap	0.0073	0.0068	0.0066
Gcc	0.0073	0.0069	0.0065
Mcf	0.0058	0.0078	0.0056
mesa	0.0064	0.0061	0.0060
mgrid	0.0058	0.0057	0.0056
parser	0.0066	0.0067	0.0060
Twolf	0.0073	0.0068	0.0064
A.M (mJ)	**6.6050**	**6.4460**	**6.0430**
16KB			
Ammp	0.0079	0.0073	0.0068
Applu	0.0071	0.0072	0.0066
Apsi	0.0096	0.0065	0.0092
Art	0.0071	0.0071	0.0081
Equake	0.0081	0.0081	0.0073
Gap	0.0085	0.0087	0.0072
Gcc	0.0097	0.0087	0.0081
Mcf	0.0067	0.0107	0.0063
Mesa	0.0078	0.0075	0.0070
Mgrid	0.0067	0.0116	0.0064
Parser	0.0073	0.0082	0.0083
Twolf	0.0089	0.0082	0.0078
A.M (mJ)	**7.9500**	**8.3240**	**7.4310**

Table B4 (continued)

32KB	8B	16B	32B
ammp	0.0096	0.0094	0.0085
applu	0.0103	0.0083	0.0081
Apsi	0.0142	0.0080	0.0133
Art	0.0093	0.0088	0.0113
equake	0.0107	0.0105	0.0094
Gap	0.0122	0.0108	0.0113
Gcc	0.0141	0.0126	0.0109
Mcf	0.0085	0.0080	0.0076
mesa	0.0107	0.0095	0.0088
mgrid	0.0086	0.0081	0.0076
parser	0.0118	0.0098	0.0104
Twolf	0.0118	0.0111	0.0101
A.M (mJ)	**10.9820**	**9.5780**	**9.7670**

SEQ-DST: 4

8KB			
ammp	0.0062	0.0061	0.0058
applu	0.0062	0.0060	0.0056
Apsi	0.0058	0.0057	0.0056
Art	0.0062	0.0060	0.0067
equake	0.0066	0.0063	0.0061
Gap	0.0075	0.0064	0.0063
Gcc	0.0071	0.0067	0.0064
Mcf	0.0080	0.0078	0.0077
mesa	0.0061	0.0061	0.0059
mgrid	0.0058	0.0057	0.0056
parser	0.0065	0.0063	0.0062
Twolf	0.0069	0.0065	0.0062
A.M (mJ)	**5.9723**	**5.8643**	**6.0262**

Table B4 (continued)

16KB	8B	16B	32B
ammp	0.0075	0.0070	0.0068
applu	0.0075	0.0072	0.0065
Apsi	0.0096	0.0065	0.0092
Art	0.0069	0.0070	0.0065
equake	0.0079	0.0076	0.0073
Gap	0.0087	0.0090	0.0076
Gcc	0.0092	0.0084	0.0079
Mcf	0.0067	0.0107	0.0063
mesa	0.0079	0.0071	0.0068
mgrid	0.0067	0.0115	0.0064
parser	0.0075	0.0079	0.0079
Twolf	0.0088	0.0081	0.0080
A.M (mJ)	**7.8960**	**8.1750**	**7.2560**
32KB			
ammp	0.0095	0.0088	0.0084
applu	0.0092	0.0083	0.0081
Apsi	0.0085	0.0080	0.0133
Art	0.0093	0.0088	0.0081
equake	0.0103	0.0100	0.0091
Gap	0.0115	0.0094	0.0091
Gcc	0.0134	0.0119	0.0109
Mcf	0.0085	0.0165	0.0076
mesa	0.0102	0.0094	0.0089
mgrid	0.0086	0.0181	0.0076
parser	0.0115	0.0098	0.0103
Twolf	0.0114	0.0095	0.0101
A.M (mJ)	**10.1382**	**9.3697**	**9.2767**

Table B4 (continued)

SEQ-DST: 8

8KB	8B	16B	32B
ammp	0.0062	0.0060	0.0058
applu	0.0059	0.0059	0.0056
Apsi	0.0072	0.0057	0.0056
Art	0.0070	0.0066	0.0058
equake	0.0065	0.0062	0.0059
Gap	0.0071	0.0066	0.0063
Gcc	0.0070	0.0066	0.0062
Mcf	0.0080	0.0078	0.0077
mesa	0.0063	0.0060	0.0059
mgrid	0.0058	0.0057	0.0056
parser	0.0066	0.0066	0.0058
Twolf	0.0069	0.0064	0.0064
A.M (mJ)	**5.8027**	**5.5323**	**5.5041**
16KB			
ammp	0.0073	0.0071	0.0068
applu	0.0074	0.0066	0.0065
Apsi	0.0067	0.0065	0.0063
Art	0.0068	0.0069	0.0068
equake	0.0075	0.0076	0.0072
Gap	0.0092	0.0086	0.0076
Gcc	0.0091	0.0082	0.0077
Mcf	0.0067	0.0107	0.0063
mesa	0.0074	0.0070	0.0070
mgrid	0.0067	0.0065	0.0063
parser	0.0078	0.0074	0.0074
Twolf	0.0085	0.0077	0.0074
A.M (mJ)	**7.5950**	**7.5740**	**6.9400**

Table B4 (continued)

32KB	8B	16B	32B
ammp	0.0095	0.0087	0.0084
applu	0.0089	0.0083	0.0078
Apsi	0.0142	0.0080	0.0076
Art	0.0088	0.0081	0.0079
equake	0.0099	0.0098	0.0088
Gap	0.0101	0.0104	0.0099
Gcc	0.0127	0.0118	0.0102
Mcf	0.0085	0.0165	0.0076
mesa	0.0102	0.0093	0.0089
mgrid	0.0085	0.0081	0.0076
parser	0.0104	0.0092	0.0100
Twolf	0.0112	0.0110	0.0095
A.M (mJ)	**10.2450**	**9.2305**	**8.6778**

SEQ-DST: 16

8KB			
ammp	0.0061	0.0059	0.0057
applu	0.0061	0.0058	0.0058
Apsi	0.0072	0.0057	0.0056
Art	0.0069	0.0059	0.0057
equake	0.0063	0.0061	0.0059
Gap	0.0066	0.0066	0.0065
Gcc	0.0068	0.0064	0.0061
Mcf	0.0080	0.0078	0.0077
mesa	0.0062	0.0060	0.0059
mgrid	0.0058	0.0082	0.0056
parser	0.0065	0.0064	0.0057
Twolf	0.0065	0.0064	0.0061
A.M	**5.5942**	**5.4268**	**5.2027**

Table B4 (continued)

16KB	8B	16B	32B
ammp	0.0072	0.0067	0.0069
applu	0.0074	0.0066	0.0069
Apsi	0.0096	0.0065	0.0063
Art	0.0068	0.0069	0.0065
equake	0.0073	0.0074	0.0072
Gap	0.0088	0.0081	0.0071
Gcc	0.0088	0.0079	0.0075
Mcf	0.0110	0.0065	0.0063
mesa	0.0073	0.0068	0.0071
mgrid	0.0067	0.0115	0.0063
parser	0.0072	0.0072	0.0073
Twolf	0.0083	0.0076	0.0070
A.M (mJ)	**6.9890**	**7.4800**	**6.8760**
32KB			
ammp	0.0094	0.0082	0.0084
applu	0.0089	0.0083	0.0078
Apsi	0.0085	0.0080	0.0076
Art	0.0088	0.0087	0.0076
equake	0.0096	0.0094	0.0088
Gap	0.0109	0.0093	0.0092
Gcc	0.0125	0.0110	0.0098
Mcf	0.0085	0.0165	0.0076
mesa	0.0098	0.0090	0.0086
mgrid	0.0086	0.0181	0.0076
parser	0.0093	0.0090	0.0102
Twolf	0.0112	0.0097	0.0088
A.M (mJ)	**9.6569**	**9.0431**	**8.5046**