PSP-Cache: A Low-Cost Fault-Tolerant Cache Memory Architecture

Hamed Farbeh[1], Seyed Ghassem Miremadi[2]
Department of Computer Engineering
Sharif University of Technology, Tehran, Iran
[1]farbeh@mehr.sharif.edu, [2]miremadi@sharif.edu

***Abstract*—Cache memories constitute a large fraction of processor chip area and are highly vulnerable to soft errors caused by energetic particles. To protect these memories, most of the modern processors employ Error Detection Codes (EDCs) or Error Correction Codes (ECCs). EDCs/ECCs impose significant overheads in terms of area and energy; these overheads increase as a function of interleaving EDCs/ECCs to detect/correct multiple errors. This paper proposes a new cache architecture to minimize the area and energy overheads of EDCs/ECCs in set-associative L1-caches. Simulation results for a 4-way set-associative cache show that the proposed architecture reduces both the area and static power overheads of parity code by about 75% and the dynamic energy overhead by about 73% in comparison to conventional cache architecture. These reduction figures are about 68% and about 66%, respectively, for SEC-DED code. The above reductions are achieved without affecting the error coverage.**

Keywords— Cache Memory; Error Detection and Correction; Soft Errors; Fault Tolerance

I. INTRODUCTION

Radiation-induced soft errors in memory cells are the major source of system failures [1]. Cache memories constitute more than 60% of total chip area [2] and are thus the most probable components to soft errors [1]. Soft errors in cache memories can be in the shape of *Single Event Upsets* (SEUs) or *Single Event Multiple Bits Upsets* (SEMUs).

Due to scaling down the feature size as well as supply voltage in today's nanometer technology, the rate of SEUs and SEMUs significantly have increased; moreover, the rate of SEMUs is comparable to the rate of SEUs in today's nanometer feature size [1]. As reported in [3], the probability of SEMU caused by the particle strike is about 40% in 40nm feature size. In addition, employing low power techniques further accelerates the rate of SEMUs.

To protect cache memories against SEUs and SEMUs, Error Detection Codes (EDCs) and Error Correction Codes (ECCs) are widely employed in most of modern processors [4]. Parity code and Single Error Correction-Double Error Detection (SEC-DED) code are the most common EDC/ECC codes that are utilized to tolerate SEUs in cache memories [4]. To detect/correct SEMUs, EDCs/ECCs interleaving are mainly exploited in cache memories.

In conventional cache architecture, EDC/ECC bits are considered for each data word and on each access to read or write operation, EDC/ECC bits are checked or updated. Hence, EDCs/ECCs impose significant area and energy overheads and their overheads further increase due to EDC/ECC interleaving requiring extra redundant bits. The higher order of interleaving to increase the SEMUs coverage, the more overheads will be imposed.

To reduce the area and energy overheads of EDCs/ECCs, they can be employed in the granularity of a cache line instead of single word. Per-line EDC/ECC reduces the number of redundant bits; however, it requires the whole line access and *read before write* operation to check and update the redundant bits [2][5]. For these reasons, per-line EDC/ECC is mostly exploited in higher memory hierarchy, i.e., L2-cache and L3-cache; while its energy and performance overheads are not affordable in L1-caches. Hence, the preferred protection scheme in L1-caches is to exploit per-word EDC/ECC.

To protect L1-caches against SEMUs, employing interleaved EDC/ECC is inevitable. High area and energy consumption overheads of per-word interleaved EDC/ECC limits the applicability of wide interleaving in L1-caches. Designing a cache memory with high error detection/correction capability and low overheads is a major reliability challenge.

This paper proposes a low-cost EDC/ECC protected cache architecture, named *Per-Set Protected Cache (PSP-Cache)*, to minimize the number of redundant bits without reducing the protection capability of EDC/ECC. In this architecture, single EDC/ECC is utilized to corresponding data words of all cache ways instead of employing an EDC/ECC for data of each cache way. PSP-Cache is based on the fact that in a set-associative L1-cache, data words in all cache ways are accessed in parallel with tag comparison operation [6]. Since all cache ways are accessed in every cache access, the proposed architecture utilizes this inherently extra cache ways accesses for EDC/ECC checking and updating. Hence, the PSP-cache can take advantage of both per-word EDC/ECC for its lower energy consumption and performance overheads and per-line EDC/ECC for its smaller number of redundant bits.

The rest of this paper is organized as follows. Background and motivations are explained in Section II. In Section III, the proposed architecture is presented. The simulation system setup and the results are demonstrated in Sections IV. Finally, Section V concludes the paper.

978-3-9815370-2-4/DATE14/©2014 EDAA

II. BACKGROUND AND MOTIVATIONS

SEMU is one of the types of Multiple Bit Upsets (MBUs) that is also referred as *Spatial MBU* in the literature [7]. The second type of MBUs is *Temporal MBU* that occurs by striking multiple particle over the time to a word [7] and is managed by memory scrubbing [7]. The probability of Temporal MBUs in L1-caches is negligible in comparison to Spatial MBUs and it is not a reliability threat; hence Spatial MBUs (SEMUs) is the major MBU challenges in L1-caches.

EDCs/ECCs can be applied to memories in different granularities. For cache memories, EDCs/ECCs are mainly employed in two granularities: 1) *per-word EDC/ECC*, in which an EDC/ECC is employed for every cache word; 2) *per-line EDC/ECC*, in which an EDC/ECC is employed to every cache line that contains a bunch of words.

Employing an EDC/ECC to larger data blocks reduces the total number of redundant bits concatenated to the memory; hence, per-line EDC/ECC imposes significantly lower area overhead in comparison to per-word EDC/ECC; lower area overhead leads to lower static power overhead and even lower dynamic power overhead. On the other hand, per-line EDC/ECC require a whole line access for every read or write operation to check or update the EDC/ECC of the line. Accessing the whole line instead of a single word requires more dynamic energy for accessing and checking/updating the EDC/ECC; accessing the larger data block which require a more complex EDC/ECC circuitry also may increase the access latency of the memory.

For L1-caches that data are accessed in granularity of a word or even a byte, energy consumption and performance overheads of per-line EDC/ECC are not affordable. Per-line EDC/ECC in L1-cache needs *read before write* operation [5] for each write operation and the also requires whole line access for each read operation. Hence, per-word EDC/ECC is employed in conventional L1-caches architecture.

The area and energy overheads of per-word EDC/ECC imposed by redundant bits are not a challenge when they are applied to protect the memory against SEUs. Employing parity code per byte or SEC-DED code per 64-bit word imposes about 12.5% area and energy overheads [8]. However, to detect or correct SEMUs, it is necessary to use interleaved EDC/ECC or employ more powerful ECC codes, e.g., Double Error Correction-Triple Error Detection (DEC-TED), Single Nibble Correction-Double Nibble Detection (SNC-DND) and Reed-Solomon (RS) codes [4]. However, for L1-caches that are protected by per-word EDC/ECC, overheads of interleaving are significantly higher than per-line EDC/ECC in L2-cache.

Set-associative caches are introduced to increase the hit ratio of the cache memories [9]. However, the these caches require a more complex cache controller circuit in comparison to the direct-mapped caches which may lead to increase in cache access latency [9]. In a K-way set-associative L1-cache, to decrease the access latency, tags of all cache ways in a set are simultaneously accessed and compared to the requested address [9]. Moreover, a conventional technique to minimize the latency of the cache is to access all K ways in parallel with tag comparison operation [8], which is known as *Fast Access* [6].

The question that arises here is that is it possible to apply the EDC/ECC to larger granularities in L1-caches without imposing the per-line EDC/ECC overheads? On the other word, is it possible to employ the EDC/ECC code to a data block larger than the data access granularity in L1-cache without requiring extra data accesses to check/update the EDC/ECC?

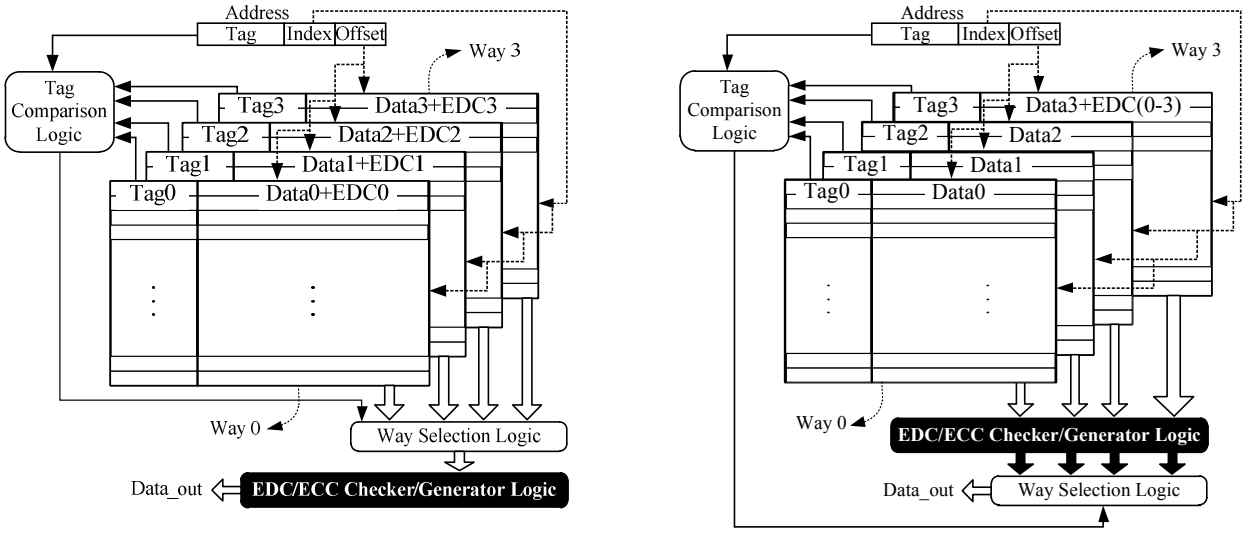
L1-cache *Fast Access* mode can answer this question. In a K-way set-associative L1-cache, the requests for a data is in the granularity of a byte, double-byte or generally a word. However, to minimize the cache access latency, K data words are accessed simultaneously in the cache data arrays. Based on this fact, in the next section we propose a new scheme to employ the EDC/ECC to larger data blocks in L1-caches that significantly reduces the EDC/ECC overheads.

III. THE PROPOSED CACHE ARCHITECTURE

In this section, we describe the proposed low-overhead EDC/ECC protected cache architecture, named *Per-Set Protected Cache (PSP-Cache)*. As discussed in Section II, both *per-word* and *per-line* EDC/CC architectures impose significant overheads to detect/correct SEMUs. This paper proposes a new architecture for set-associative caches that shares a single EDC/ECC among all cache ways without decreasing the protection capability of the code. The aim of this architecture is to minimize the number of redundant bits for L1-caches in order to reduce the overheads of memory protection mechanism. The main idea of PSP-Cache is to exploit single code for data of all cache ways instead of using a code for data of each cache way.

In a set-associative L1-cache that operates in Fast Access mode, for each access to the cache memory, all cache ways are accessed in parallel to tag comparison operation in order to eliminate the way access latency [6]. Accessing all cache ways instead of accessing the requested way improves the performance of the cache at the cost of increasing the cache dynamic energy consumption. PSP-Cache exploits these inherently-imposed extra accesses to minimize the overheads.

In PSP-Cache architecture, all data that are accessed simultaneously are protected by single code, instead of assigning a code for each of them. Fig. 1 depicts an abstract view of the conventional cache and the proposed cache architecture for a 4-way set-associative cache. According to Fig. 1, data of all 4 ways are accessed based on the *index* and *offset* fields of the requested data address. As illustrated in Fig. 1 (a), in conventional cache architecture, these data are applied to *Way Selection Logic* and the output of *Tag Comparison Logic* selects one of these data as the response of the request to the cache. The selected data is then applied to *EDC/ECC Checker/Generator Logic* and then delivered to data bus. In PSP-Cache architecture that single code is assigned to the data of all cache ways, as depicted in Fig. 1 (b), EDC/ECC Checker/Generator Logic operates on all accessed data. Hence, this unit is located prior to Way Selection unit and the output of EDC/ECC Checker/Generator Logic is fed to the Way



(a) Conventional cache architecture

(b) Proposed cache architecture

Fig. 1. Conventional EDC/ECC protected cache and proposed cache architecture

Selection Logic. The output of EDC/ECC Checker/generator Logic is data of all cache ways without EDC/ECC.

Parity code is the most conventional error detection code used in L1-caches. Instruction-caches (I-caches) and write-through Data-caches (D-Caches) are mostly protected by parity codes. In parity code, the number of redundant bits needed to detect up to a specific number of bit errors is independent of the data length. Therefore, the number of redundant bits that is required to protect all N cache ways in PSP-Cache is equal to the number redundant bits of single cache way in conventional parity-protected cache.

Error Correction Codes (ECCs) employed in cache memories are based on Hamming code [10]; and SEC-DED code, that is capable to correct single bit errors and detect two-bit errors, is the most conventional ECC in write-back D-caches. To improve the correction capability, SEC-DED interleaving is exploited. The number of redundant bits of ECCs increases logarithmically as the data length increase. For example, in SEC-DED code, a 32-bit data and a 64-bit data require 7 and 8 redundant bits, respectively. Therefore, the number of redundant bits that is required to protect all N cache ways in PSP-Cache is equal to the number redundant bits of single cache way in conventional SEC-DED protected cache plus *logN*.

The main features of the proposed architecture are as follows:

- A negligible modification on cache architecture is required to implement PSP-Cache;
- It is applicable to the tag array of cache memories in addition to data array. Moreover, both D-cache and I-cache can take advantages of this architecture;
- It is independent of cache protection granularity. Hence, all set-associative caches with per-X-bit EDC/ECC protection, when X is between a single byte to the cache line length, can be transformed to PSP-Cache architecture;
- The efficiency of the proposed architecture improves by increasing the cache associativity.

IV. Analysis and Evaluation

The proposed architecture is analyzed and evaluated in terms of energy consumption, area, and reliability. Conventional parity and SEC-DED protected cache are considered as the baseline and CACTI tool [11] has been used to model the cache architecture. A 16 Kbyte cache, in 65 nm technology, that operates in Fast Access mode is evaluated. In the following subsections different aspects of PSP-Cache are evaluated.

A. Energy Consumption and Area Overheads

There are two sources of the energy and area overheads imposed by EDC/ECC code: 1) overheads of redundant bits, and 2) overheads of EDC/ECC checker/generator circuit. In L1-caches, redundant bits are the major source of area and energy overheads. The static power and area overheads of checker/generator circuit are negligible (less than 1%) in comparison to the overheads of redundant bits. Redundant bits also are the dominant overhead of dynamic energy.

Table I shows the number of redundant bits required to protect cache memory for baseline cache and PSP-Cache. The granularity of data protection is 64-bit word and five EDC/ECC codes are considered for cache protection. According to Table I, the reduction in the number of redundant bits in PSP-Cache is proportional to the cache associativity. For parity code, PSP-Cache reduces the number of redundant bits in 2-way, 4-way and 8-way set associative caches by 50%, 75% and 87.5%, respectively. In SEC-DED code, this feature is reduced by 43.75%, 68.75% and 82.81% for 2-way, 4-way and 8-way set associative caches, respectively. The reduction in the static power and area overheads of the proposed

TABLE I. NUMBER OF REDUNDANT BITS REQUIRED TO PROTECT CACHE FOR DIFFERENT CACHE ASSOCIATIVITY AND PROTECTION CODES

EDC/ECC type	Number of redundant bits			
	Baseline cache	**PSP-Cache**		
		2-way cache	4-way cache	8-way cache
1-bit parity per word	N	$\frac{N}{2}$	$\frac{N}{4}$	$\frac{N}{8}$
2-bit interleaved parity per word	$2N$	N	$\frac{N}{2}$	$\frac{N}{4}$
4-bit interleaved parity per word	$4N$	$2N$	N	$\frac{N}{2}$
8-bit interleaved parity per word	$8N$	$4N$	$2N$	N
SEC-DED (72,8)	$8N$	$\frac{9N}{2}$	$\frac{10N}{4}$	$\frac{11N}{8}$
N : Number of words in the cache				

architecture are according the above mentioned values in comparison to conventional cache architecture.

In PSP-Cache, the reduction in dynamic energy overhead is slightly lower than the area and static power reduction. The reason is that the checker/generator logic imposes more dynamic energy than that of baseline cache. Fig. 2 illustrates the dynamic energy overhead of PSP-Cache for different cache architecture normalized to baseline cache. For parity code, PSP-Cache reduces the overhead by 49%, 73% and 86% for 2-way, 4-way and 8-way set-associative caches, respectively. These reductions are 43%, 66% and 79% for the SEC-DED code.

B. Reliability Analysis

In this subsection, the effect of PSP-Cache on the error detection/correction capability of the EDC/ECC code is explored. We consider both SEUs and SEMUs for reliability analysis.

SEUs: both simple parity code and interleaved parity code are capable to detect single bit errors (SEUs) regardless of the data length. The same is correct for the SEC-DED code to correct SEUs. Hence, the proposed architecture does not hurt the protection capability of the EDC/ECC code.

SEMUs: SEMUs occurs in adjacent bits and interleaved code is exploited to protect the cache against them. Since PSP-Cache concatenates the data words that have no adjacency (data words of different cache ways), it does not hurt the protection capability of interleaved EDC/ECC in comparison to conventional cache.

Accordingly, PSP-Cache that extremely reduces the number of redundant bits to protect cache memory has no impact on the reliability of the cache memory and does not decrease the error detection and correction capability of the conventional cache protection methods.

V. CONCLUSIONS

The overheads of EDCs/ECCs in terms of energy consumption and area are the major challenges in using

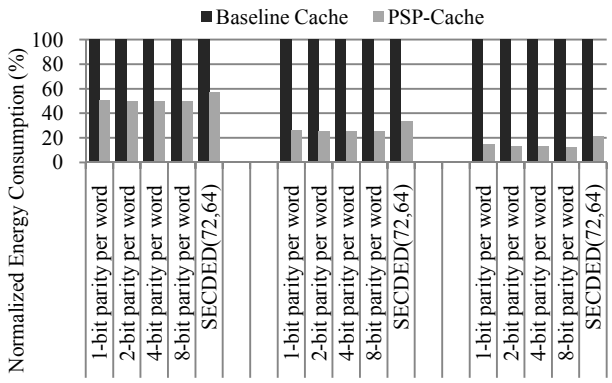


Fig. 2. Normalized dynamic energy for baseline cache and PSP-Cache.

EDCs/ECCs in L1-cache. This paper proposed a cache architecture, named *PSP-Cache*, that significantly decreases the overheads of EDC/ECC by reducing the number of redundant bits needed to protect the cache lines. Our evaluations show that for a 4-way set associative cache, the static power and area overheads of PSP-cache for parity code are about 75% lower than conventional cache; this value is 68.75% for a SEC-DED code. Moreover, the proposed architecture reduces the dynamic energy overhead of parity code and SEC-DED code by about 73% and 66%, respectively.

REFERENCES

[1] Z. Ming, X. L. Yi, L. Chang, and Z. J. Wei, "Reliability of memories protected by multibit error correction codes against MBUs," IEEE Transactions on Nuclear Science, vol. 58, no. 1, pp. 289-295, 2011.

[2] M. Manoochehri, M. Annavaram, and M. Dubois, "CPPC: correctable parity protected cache," in Proceedings of International Symposium on Computer Architecture, pp. 223-234, 2011.

[3] A. Dixit and A. Wood, "The impact of new technology on soft error rates," in Proceedings of IEEE International Reliability Physics Symposium, pp. 486-492, Apr. 2011.

[4] S. Kim and A. K. Somani, "Area efficient architectures for information integrity in cache memories," Proc. International Symp. on Computer Architecture, pp. 246-255, 1999.

[5] J. Kim, N. Hardavellas, K. Mai, B. Falsafi, and J. Hoe, "Multi-bit error Tolerant Caches using Two-dimensional Error Coding," in Proceedings of the 40th Annual IEEE/ACM International Symposium on Microarchitecture, pp. 197-209, 2007.

[6] D. avid Tarjan, S. Thoziyoor, N. P. Jouppi, CACTI 4.0, HP Laboratories Palo Alto, HPL-2006-86, June 2, 2006.

[7] S. S. Mukherjee, J. Emer, T. Fossum, and S. K. Reinhardt, "Cache Scrubbing in Microprocessors: Myth or Necessity?," in Proceedings of 10th IEEE Pacific Rim International Symposium on Dependable Computing, pp. 37-42, 2004.

[8] C. W. Slayman, "Cache and memory error detection, correction, and reduction techniques for terrestrial servers and workstations," IEEE Transactions on Device and Materials Reliability, vol. 5, no. 3 pp. 397-404, 2005.

[9] D. A. Patterson, and J. L. Hennessy, Computer organization and design: the hardware/software interface, Morgan Kaufmann, 2008.

[10] R. W. Hamming, "Error Detecting and Error Correcting Codes," Bell System technical journal, vol. 29, no. 2, pp. 147-160, 1950.

[11] S. Thoziyoor, N. Muralimanohar, J. H. Ahn, and N. P. Jouppi, "Hp labs cacti v5.3," CACTI 5.1, HP Laboratories, Tech. Rep. HPL-2008-20, 2008.