ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

An ISO 3297: 2007 Certified Organization Vol. 5, Special Issue 4, March 2016

National Conference on Signal Processing, Instrumentation and Communication Engineering (SPICE' 16)

Organized by

Dept. of ECE, Mar Baselios Institute of Technology & Science (MBITS), Kothamangalam, Kerala-686693, India

Efficient Method for the Error Protection of Cache Memory by using the Similarity of Tag Bits

Mary Francy Joseph[1], Anith Mohan[2]

PG Student [VLSI &Embedded Systems], Dept. of ECE, College of Engineering College, Munnar, Kerala, India [1]

Assistant Professor, Dept. of ECE, College of Engineering College, Munnar, Kerala, India[2]

ABSTRACT: High performance processors make use of caches to increase the rate at which they can process the data. Soft errors can corrupt the information in the memory especially in cache memory which is the closest data storage to the CPU. Cache tag field are critical to correctness of cache access and to achieve high hit rate. Complex protection mechanism to tag bit information can degrade the performance due to longer cache access latency. Using the spatial locality of memory accesses error protection of tag bits is improved. The error correcting capability of tag bits is enhanced by exploiting the similarity of tag bit values. It is checked if neighbouring cache lines have the same tag bits as that of the data fetched, when data is fetched from main memory. Same tag bit values are stored as additional bits to be used later in the cache. Simulation of the proposed SIM tag approach is done by varying the number of ways in the cache. Results shows that proposed approach improves the error protection capability with negligible performance overheads.

KEYWORDS: Cache, Tag Bits, Soft Error.

I.INTRODUCTION

With continued technology scaling due to minimum feature size reduction, soft errors are becoming a major reliability issue in modern electronic systems. Soft errors also known as single event upsets occur when a radiation particle hits an electronic system [1], [2]. Cache memories are vulnerable to soft errors because they operate at low voltage and their size increase due to multi level cache hierarchy. An incorrect data value can easily propagate into processor registers and memory elements and cause computation failures. Soft error mitigation is an important consideration for cache design. As result data integrity checking is utilized in cache memories [3], [4].

Different techniques have been proposed to protect cache against soft errors. Protection is achieved using error detection and correction codes [5], [6]. Parity bit or SEC-DED (single error correct, double error detect) ECC code is used to protect a group of bits. Parity can detect errors but cannot correct errors depending on the cache policy. SEC-DED ECC code can detect all double bit errors and always recover data transparently for single bit errors. Mitigation of errors can also be done by introducing modifications in the manufacturing process. Due to area overhead, error protection coverage and latency overhead these schemes are not efficient.

Soft error on the cache tag bits can cause pseudo misses and pseudo hits. Pseudo hit occurs due to a hit that is actually a miss in the absence of a soft error. Pseudo miss is a miss that is actually a hit when there is no soft error. Complex error protection mechanisms against soft errors can degrade the performance due to longer delay. Caches have to operate with low latency. Providing high reliability with low latency mechanism is important to maintain system performance and system integrity.

In this paper the similarity of tag bits is exploited to improve the error protection capability of cache memory. There are many same tag bit values with those of other tag bits in adjacent cache sets. This is known as spatial locality of memory accesses. When a cache line is accessed or replaced a cache line with same tag bits with those of the accessed cache line is likely to be found in neighbouring adjacent cache sets. This is called similarity of tag bits. Error detection

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

An ISO 3297: 2007 Certified Organization Vol. 5, Special Issue 4, March 2016

National Conference on Signal Processing, Instrumentation and Communication Engineering (SPICE' 16)

Organized by

Dept. of ECE, Mar Baselios Institute of Technology & Science (MBITS), Kothamangalam, Kerala-686693, India

is done using error detecting codes like parity check bits. Error correction can be done using the similar tag bits in the adjacent cache sets. Erroneous tag bits are simply replaced with the correct tag bits in the adjacent cache sets.

The proposed SIM tag approach is evaluated with the conventional in cache replication (ICR) scheme. Both approaches are compared in terms of delay. The proposed SIM tag approach substantially reduces the cache access time. The cache memory should have as low delay as possible. Otherwise the system performance will be degraded.
This remaining part of the paper is organised as follows. Next section gives an overview of the conventional approaches for error protection. The proposed SIM tag approach is postulated in section III. Simulation results of the conventional approach and SIM tag approach is explained in section IV. Section V concludes the paper.

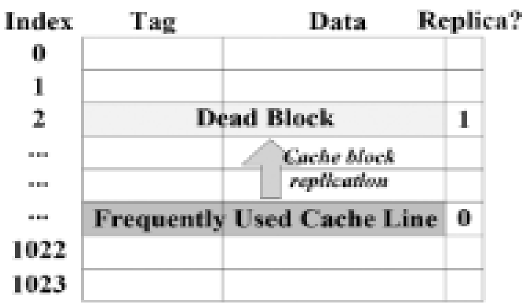
II. RELATED WORK

Soft errors which occur more often than permanent faults can corrupt the data and instruction in the memory. Errors in cache memory can easily propagate into registers and other elements of memory which eventually results in computation failures [7], [8]. Bit changes in the tag bits causes pseudo hits and pseudo misses. Pseudo hit occurs when the tag of the incoming address matches with the false cache tag field. Pseudo miss occurs when the tag field does not match an incoming reference address because this tag is affected by error. As a result error detection and correction is usually used in cache memories.

Different techniques have been proposed to protect caches against soft errors. Information redundancy is added to the original data to protect caches against errors. Error detection and correction codes such as parity and single error detection and correction (SEC-DED) code are used to protect cache memory. Parity check codes cannot correct errors. SEC-DED codes can correct single bit errors but cannot correct double bit errors. These codes are expensive way to maintain cache reliability. So it is not suitable under lo error rates. It is not suitable in terms of area requirement. Area of the check code is proportional to cache size. Latency for cache operation is another problem. Complex ECC code does not effectively utilize the pipeline of data path.

With continued technology scaling the rate multi bit errors are increased [9]. Interleaving can be used to mitigate multi bit errors, but it increases the latency and complexity of cache operations. Multiple strike flips causes multi bit errors in ECC code word. Cache scrubbing is another solution for multi bit errors. To combat against multi bit errors short scrubbing interval is required. But it needs extra cycles. This will increase the latency and power consumption.

Replication cache mechanism is an approach to mitigate multi bit errors [10]. In this scheme the replica for every write to the cache memory is stored into another cache and uses copies of this data for error correction. Its performance will be degraded because of increasing write-back rates resulting from evictions from the replication cache. Transient errors cannot be corrected and it will lead to wrong data consumption.

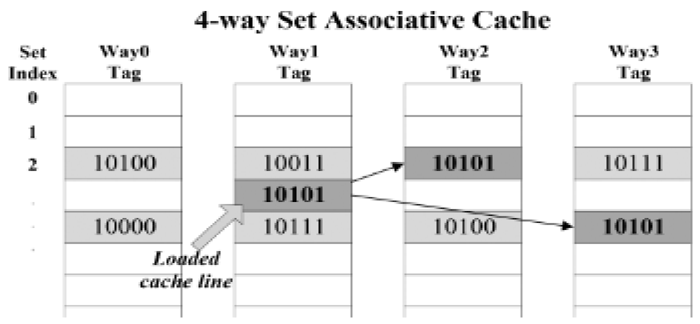


ICR has been proposed [11] to reduce the vulnerability of cache tag bits and Fig.1 shows its mechanism. In ICR scheme the frequently accessed cache lines are replicated into other cache lines which are predicted not to be accessed further and these are called dead blocks. Dead bocks in the data cache are recycled to maintain frequently used data bits against soft errors. Dead block miss prediction is used to avoid performance degradation. Replicated cache lines can be used to correct errors in tag bits in the active lines. ICR increases the miss rates because of dead block prediction technique and result in increased energy consumption and performance loss. Most of the previous works have been studied the effectiveness of tag bits in terms of the reliability of tag bits. The effectiveness in terms of energy performance and area overhead has not considered.

III. PROPOSED ARCHITECTURE

Caches reduce the average memory latency due to property of programs called principle of locality. This principle states that programs access certain regions of the data and code space more often than others. Instructions or data whose addresses are close together has more probability to be accessed around the same time. This is known as spatial locality.

The probability to find the same tag bits in neighbouring cache sets is high. When a cache line is accessed similar tag bits as those of the fetched line are found in the neighbouring cache sets. This is known as spatial locality of memory accesses. Fig.2 shows the spatial locality of tag bits.



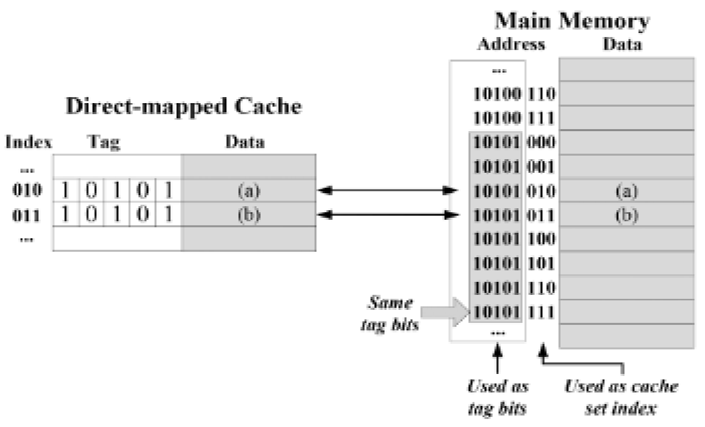
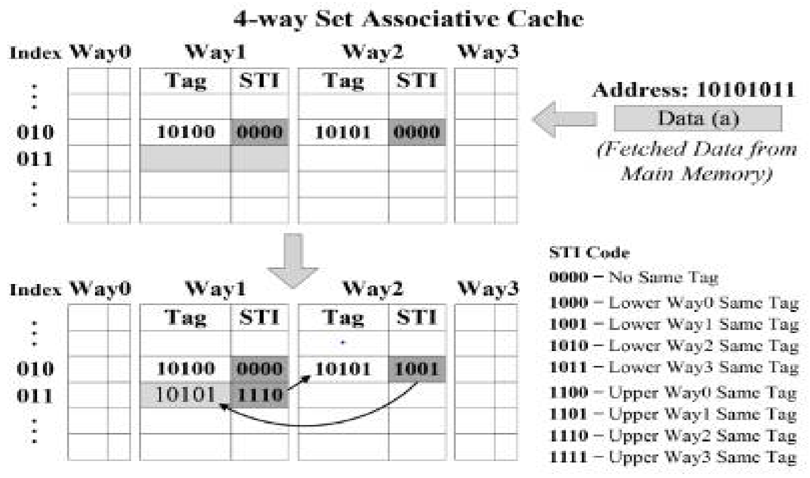


Fig.3 shows the tag bit similarity in detail. Eight bit main memory address and eight entry direct mapped cache are assumed in this example. Data (a) are accessed and transferred to the cache. The lower three bits of its address are used as cache set index and upper five bits are used as tag bits. The tag bits reflect upper side of memory address. So most of the tag bits are same in neighbouring cache entries. Data (b) located near data (a) are accessed with a high probability because of the spatial locality of memory accesses. There can be two different data with similar tag bits in the cache memory if both data (a) and data (b) are cached. So different data values with similar tag bits exist in the cache memory

The proposed scheme exploits the same tag bits in the neighbouring cache sets to correct tag bits with error. The error correcting capability of tag bits is improved by exploiting the similar tag bits. To point to the exact location of the same tag bits in the upper or lower set additional bits are required to encode location information. Extra bits are called same tag information (STI) bits. There are three parts for same tag information (STI) bits; a valid bit, a set location bit and way location bits. The valid bit denotes whether certain tag bits have the similar tag bits in the neighbouring set or not. The set location bit indicates an upper or lower set. Way location bits denote the specific cache way which has the same tag bits. Depending on the associativity of caches the length of way location bits can be different.

The detailed operation the technique is explained with an example and is shown in Fig.4. Four-way set associative cache is considered here.


4-way Set Associative Cache
Index
Way0
Way1
Way2
Way3
Tag
STI
010
011
10100
0000
10101
0000
Address: 10101011
Data (a)
(Fetched Data from Main Memory)
10101
1110
1001
STI Code
0000 – No Same Tag
1000 – Lower Way0 Same Tag
1001 – Lower Way1 Same Tag
1010 – Lower Way2 Same Tag
1011 – Lower Way3 Same Tag
1100 – Upper Way0 Same Tag
1101 – Upper Way1 Same Tag
1110 – Upper Way2 Same Tag
1111 – Upper Way3 Same Tag

Four bit STI is attached into each entry of the tag array. When data (a) are fetched from main memory upper and lower sets are denoted to find the similar tag bits. Upper set is found to have the same tag bits as newly fetched tag bits. STI bits of data (a) are filled with a proper STI code after checking the lower and upper sets. STI bit of existing line with whose tag value is same with the new tag bit value is also filled with a STI code. The STI code of the upper right side set were originally 0000. It indicates that there is no same tag and the corresponding tag bits were vulnerable to errors. STI bits are encoded to 1001 due to the same tag bits in the lower right side set after loading data (a). So the corresponding tag bits are protected by using the similarity of tag bits. Therefore both tag bits with similar values can be recovered from errors by fetching other intact tag bits. The tag bits of the evicted cache line may have been pointed by STI bits of other lines in the neighbouring cache sets. STI bits will point wrong tag bits if this is not handled properly.

A. Detailed Architecture

Four simple components are added to the cache structure to exploit the similarity of tag bits for protection tag bits against soft errors. The detailed architecture is shown in Fig.5. Shifter is added to access upper or lower set. STI bits are generated using STI encoder. On replacements STI bits in the upper or lower sets are modified using STI replacement handler. For error correction error correction unit is used.

1) Shifter: Upper or lower set are accessed by changing the output signals of the cache decoder. Shifter is added for this purpose. On cache misses, shifter shifts the output signals to the left and right to generate STI bits. Decoder output signals are shifted for error recovery. Shift direction is determined by STI set location bit.
2) STI Encoder: STI encoder compares the tag bits of the cache missed data with other tag bits in the lower or upper sets. Thus STI bits are generated. STI valid bits is set to one when there are matching tag bits in a neighbouring set. Using a multiplexer way location bits are generated. STI set location bit is obtained from the main controller.
3) STI Replacement Handler: STI bits in the upper or lower set must be updated on cache replacements. STI replacement handler checks all STI bits in neighbouring sets. If certain STI bits point to the tag bit values of the replaced cache line ten their STI valid bits are invalidated and new STI bits are generated by finding other same tag bits..
4) Error Corrector: Uncorrupted tag bits should be obtained from neighbouring cache sets using STI bits if the same tag bits exist. Thus the erroneous tag bits are corrected. The location having the same tag bits is obtained from the STI way location bits. Tag bits with error are replaced with them after fetching right tag bits.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

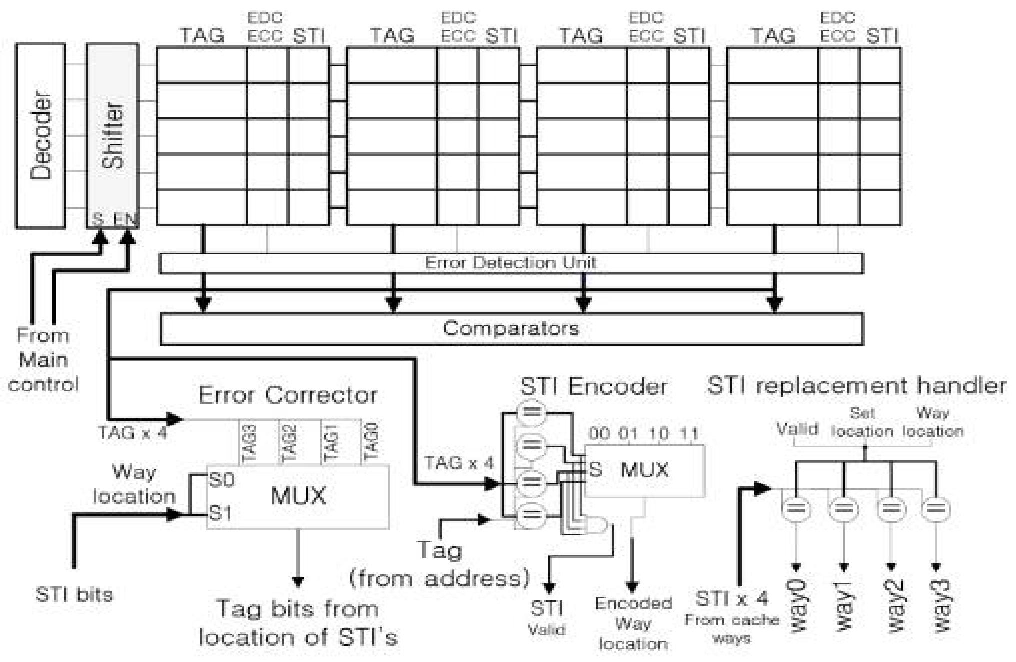
International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

An ISO 3297: 2007 Certified Organization Vol. 5, Special Issue 4, March 2016

National Conference on Signal Processing, Instrumentation and Communication Engineering (SPICE' 16)

Organized by

Dept. of ECE, Mar Baselios Institute of Technology & Science (MBITS), Kothamangalam, Kerala-686693, India



5) Main controller: To generate controller signals main controller is modified. On cache misses or tag bit errors the pipelines are stalled. At that time the main controller will signal the shifter to access the neighbouring sets.

IV. RESULTS AND DISCUSSION

The SIM tag approach and ICR approach are implemented using verilog and synthesized for the 90nm technology using cadence virtuoso tool. Simulation results of the proposed system are shown in figure. SIM tag architecture is compared with the conventional ICR approach. The comparison in terms of delay is shown in table 1. The number of ways in the cache has been varied from 2 to 16. The tag length increases by a bit each time the number of ways is doubled. From the results it is clear that delay savings are larger when the number of ways in the cache increases. The proposed approach shows larger savings when the number of ways in the cache increases. This is a key factor as ECC implementations in caches are limited by delay.

TABLE I: DELAY ESTIMATES (IN NANOSECONDS)

	Conventional ICR approach	**SIM tag approach**
2 Way	0.40	0.22
4 Way	0.40	0.23
8 Way	0.41	0.24
16 Way	0.41	0.25

V. CONCLUSION

With continuous technology scaling and increase of cache size the transient error rates are increasing. So it is important to provide error detection and correction mechanism for semiconductor memories especially for cache memories. SEC-

DEC code or parity code has been widely used to mitigate transient errors in cache memories. But these techniques are reluctant choice because of performance overhead and additional energy consumption due to ECC encoding/decoding. Similarity of tag bits is used to improve the error protection of tag bits in this paper. When an error is detected using parity check codes, the error can be corrected if the same tag bits are present in neighbouring cache sets. The erroneous tag bits are simply replaced with the correct tag bits in the neighbouring cache sets. The proposed architecture is compared with the conventional ICR approach. The proposed SIM tag approach can substantially reduce the delay overhead needed to implement the protection which is a more relevant factor in caches.
.

References

[1] S. Kim and A. Somani, "Area efficient architectures for information integrity in cache memories," in *Proc. Int. Symp. Comput. Archit.*, 1999, pp. 246–255.

[2] B. Gold, M. Ferdman, B. Falsafi, and K. Mai, "Mitigating multi-bit soft errors in L1 caches using last-store prediction," in *Proc. Workshop Archit. Support Gigascale Integr.*, 2007, pp. 1–8.

[3] S. Mukherjee, J. Emer, T. Fossum, and S. Reinhardt, "Cache scrubbing in microprocessors: Myth or necessity," in *Proc. Int. Symp. Dependable Comput.*, 2004, pp. 37–42.

[4] C. Slayman, "Cache and memory error detection, correction, and reduction techniques for terrestrial servers and workstations," *IEEE Trans. Device Math. Rel.*, vol. 5, no. 3, pp. 397–404, Sep. 2005.

[5] N. Wang and S. Patel, "ReStore: Symptom-based soft error detection in microprocessors," *IEEE Trans. Dependable Sec. Comput.*, vol. 3, no. 3, pp. 188–201, Jul. 2006.

[6] O. Ergin, O. Unsal, X. Vera, and A. Gonzaez, "Exploiting narrow values for soft error tolerance," *IEEE Comput. Archit. Lett.*, vol. 5, no. 2, pp. 1–12, Dec. 2006.

[7] T. Austin, "DIVA: A reliable substrate for deep submicron microarchitecture design," in *Proc. 32nd Annu. Int. Symp. Microarchitecture*, 1999, pp. 196–2007.

[8] Weaver, J. Emer, S. S. Mukherjee, and S. K. Reinhardt, "Techniques to reduce the soft errors rate in a high-performance microprocessor," in *Proc. 31st Annu. Int. Symp. Comput. Archit.*, Jun. 2004, pp. 264–275.

[9] L. Hung, M. Goshima, and S. Sakai, "Mitigating soft errors in highly associative cache with CAM-based tag," in *Proc. IEEE Int. Conf. Comput. Des.*, Oct. 2005, pp. 342–347.

[10] W. Zhang, "Replication cache: A small fully associative cache to improve data cache reliability," *IEEE Trans. Comput.*, vol. 54, no. 12, pp. 1547–1555, Dec. 2005.

[11] W. Zhang, S. Gurumurthi, M. Kandemir, and A. Sivasubramaniam, "ICR: In-cache replication for enhancing data cache reliability," in *Proc. Int. Conf. Dependable. Syst. Netw.*, 2003, pp. 291–300.