ISSN (Print) : 0974-6846
Indian Journal of Science and Technology, *Vol 9(29), DOI: 10.17485/ijst/2016/v9i29/90864, August 2016* ISSN (Online) : 0974-5645

Improving the Reliability of Cache Memories using Identical Tag Bits

S. Manju Bharathi*, R.Vijay Sai and S. Saravanan

School of Computing, SASTRA University, Thanjavur – 613401, Tamil Nadu, India;
manjushanmugam92@gmail.com, vijaysai it.sastra.edu, saran@core.sastra.edu

Abstract

Cache memories are revealed to transitory error in tag bits and some of the efforts have been taken to decrease their susceptibility. In the advanced mechanisms of cache memories are most applicable components, because the soft errors are protected. The identical tag bit data is used to regain from the error in the tag bits. In this paper, to improve error protection capacity of the tag bits in caches, power efficient cache design is proposed by using Superlative Standard Techniques (SST) architecture to achieve power. To utilize the identical tag bits for transitory error protection, the proposed scheme is discussed by selecting the energy superlative standard techniques that decrease unwanted interior activities by reducing the dynamic switching power. In experimental method, results show that our proposed multilevel cache architecture sustains a performance of achieving dynamic power and reduces the power consumption up to 85% when applied to energy optimal technique.

Keywords: Cache Memories, Identical Tag Information, Superlative Standard Techniques (SST), Tag Bits, Transitory Error

1. Introduction

Transitory delusion generate using exterior transmission has turn in to essential concern in chip design. Tag bits in stored cache are revealing to transitory delusion. Caches contain major structure in chip design and it's more pregnable to the transient errors. Few efforts have been taken to reduce their susceptibility. Preventing the transient errors is becoming important and providing correct mechanism for given hardware circuits. Specifically, cache memories are susceptible through they operate at low voltage levels. Soft error conscious cache design has become increasingly crucial for reliable computing. The widely used Error Correction Code (ECC) has only limited capability in error detection and correction, while incurring nontrivial penalty in area or performance. The N modular idleness scheme is too costly for processors with stringent cost constraints. To improve consistency of cache tag formation, it needs more consistency aligned with errors when our data formation is confined.

To improve the multiplex error to framework in S2 cache established using idleness and soft errors are protected by some techniques. To evaluate result, this paper identifies Cache susceptibility Feature (CSF) is to prospect soft errors in stored cache to propagate in processor. Error modification and Error recognition technique. Here, a huge quantity of methods has projected to defend tags. In area transparency and error defense reporting, the above method is not used in adequate manner. The proposed scheme is discussing by selecting energy superlative standard techniques that decreases unwanted interior activities by reducing dynamic switching power. A shifter has encoder with small circuits and error correction code. When the conservative equality verifying bits are detected the error. To execute identical

* *Author for correspondence*

Tag first, shifter is an additional device added in detailed architecture. Identical bits are available in adjoining cache data, error can be corrected. Error detection codes and Error correction codes are mainly used to detect/correct the multiple errors and protect the memories. Soft errors are very important to ensure reliability.

2. Background and Related Work

New cache architecture is proposed to minimize the area and energy overheads of error deduction and error correction mechanism in set associative L1-caches. Simulation results for a four-way set associative cache shows the proposed architecture reduced in both the area and static power overheads of parity code and the dynamic energy overhead in comparison to conventional cache architecture. These reductions are achieved by using caches[1]. Caches with multicore architecture is described in paper[2]. On-Chip DRAM Cache for Simultaneous Miss Rate and Latency Reduction[3]. N-way associative cache model used for selected parameter is experimented. The energy consumption is compared with the fixed set associative cache of same –size[4.] The technique which is targeting to defending tag array in particular caches[5]. Experimental approach demonstrate Fast Tag needs an area, delay (ms), and dynamic power transparency is less than conservative technique which is presently determined. CSF is to evaluate the reliability for different stored caches. Caches May generating the memory hierarchy and processor. Computing the CSF based on the cache[6.] Our results shows based on errors from write down-throughout stored data able to be covered with not including or disturbing other mechanism. They also suggest two early strategies to get better consistency of write-reverse stored data without compromise elevated performance, hit latency and memory transmission Modal Cache achieves hit latency reduction[7]. The LRU cache policy is designed and then an approximate closed form expression. It's provided through the theory based on the function of item request rates and cache size. Finally, results of the extensive simulation are illustrated[8]. The architecture, has proposed to reduce the area overhead to handling bit errors from faulty cells[9]. They have proposed many techniques to this access time failure. Sub array-level parallel access minimizes the performance of cache bandwidth loss due to MLA. This result shows our Proposed L1 cache architecture sustains a performance hit of less than compared[10] to the conventional cache architecture with no access time failure[11]. The transistor N5 is connected to source voltage which is positive and assigned to be active during sleep mode and the transistor N6 delivers the cell to be directly grounded during enable state. So, these modes are analyzed by changing the voltage sources of the pass transistors and that is done to make less leakage[12]. Tag filter cache architecture is designed for increasing the Efficient of energy in cache design. In this cache tag bits are stored each way in lower order in an auxiliary bit array. Finally, Tag filter cache architecture reduces the consumption of power[13]. In presents a unique security compute which is applied to different category of memory flexibly[14]. The concept of scramble is utilized efficiently. An error in L2 caches based on redundancy is described in paper.

3. Detailed Architecture of Identical Tag

The existing method has a fundamental idea which is to be utilizing identical tag bits to correct imprecise tag bits in adjoining sets. By exploiting the same tag bit values, To strengthen exceedingly in error correcting ability of tag bits through utilizing the identical tag bit values by using only in overheads. To predetermine the location we required the extra bits information and also we can attain our goal, which denotes an accurate position of identical tag information in higher and also in subordinate position. These types of additional bits are also described as identical tag information bits and also called as ITI bits. Our existing system architecture is interconnected with tag element, which is more susceptible with system performance. Here they explain the existing design mention with tag bit data and ITI bits are higher and also subordinate data of original cache. It invalidate identical tag information bit directing to ejected stripe, and for the new tag bits we are generating the proper identical tag bits. Shifter and counter based approaches has implemented in our existing approach. Architecture of Identical Tag is shown in Figure 1.

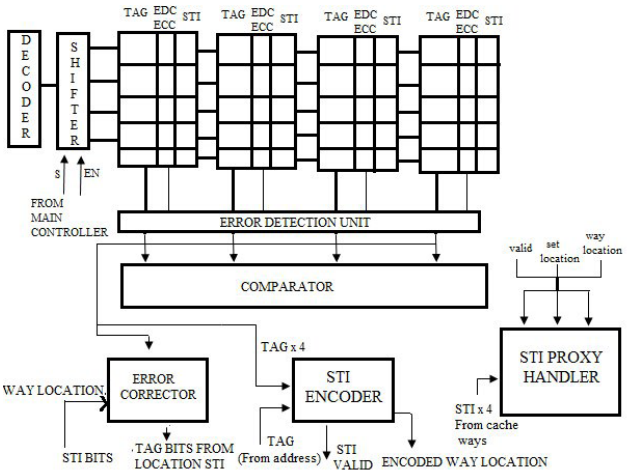

Figure 1. Architecture of Identical Tag.

3.1 Hardware Components

Existing architecture [ref your base paper no] is to utilize tag bits resemblance for transitory error security of tag bits, Hardware components considered with four simple method are enlarge with conventional cache structure. Representation of micro architecture which designed in our method called iTag (identical tag bits). To execute identical Tag first, shifter is an additional device which is used to access a higher or subordinate cache set. Even though, critical path is used to increase by using shifter device to access the cache, and also we develop another technique which is called counter based technique is discussed in this architecture next, to engender ITI bits we needed the ITI Encoder. Then, to transform ITI bits by higher and subordinate cache set by substitution is modified by using ITI substitution handler. Finally, an error rectification division is used to build for error revitalization.

3.1.1 Shifter Device

By using cache in a higher and subordinate set, it can be access when we varying output by the signals of cache decoder. By Engender ITI bit, shifter should shifts the signals in decoder which produce the output by both sides side in consecutive manner. In case of recovering error, signals projecting the output of decoder are shifts to find the location of ITI bits. So the outlay of error may ignore and transitory error rate is moderate.

3.1.2 Counter Technique

Counter performance is identical to the shifter operation; it rises or fall an index value of set on cache miss. Memory access stage is varying the index value of cache set in rising and falling time. During the implementation stage in multiplexer, it select the regenerated index value of cache set while occurring of error. This chosen set value is used by upcoming cycle to accessing higher and subordinate sets to obtain the identical tag bits. This method is entirely overlap cache miss process.

3.1.3 ITI Encoder

The cache miss information is comparing the tag bits by other tag bits placed in higher and subordinate sets in ITI encoder, whereas pipeline is due to the cache miss to engender ITI bits. If any identical tag bits in adjoining set, An ITI convincing bits are located to one side and ITI way position bit is generate by multiplexer. ITI set position bit can obtain by main controller

3.1.4 ITI Proxy Handler

ITI bits in higher or subordinate data should be simplified on cache proxy. Then ITI proxy handler corrects all ITI bits in adjoining sets. When the particular ITI bits are position to tag bits by cache line method, Then ITI convincing bit are engender new ITI bits by judging another identical tag bits.

3.1.5 Error Corrector Sector

The tag bits without corrupting should obtain by adjoining cache set using ITI bits. To improve tag bits from errors. The accurate position having the identical tag bits may obtain with ITI way position bits. The corrupted tag bits should replace after fetching enticing tag bits.

3.1.6 Main Controller

To engender essential control signals, the main controller is lightly customized to sustain additional device. The main controller signals needs extra counter (or shifter) to contact adjoining sets.

4. Cache Design Architecture

To consummate the approach, we are modifying our existing architecture correlated to tag system, which should be more inventiveness to improve our system performance. The basic concept to proposed approach is utilizing identical tag bits are adjoining set to accurate imprecise tag bits. In case of utilizing the identical tag bit values, Error correcting potential is highly enhanced in our tag bits only when outlay are negligible. In our proposed architecture design, three cache design techniques are used by selecting optimal baseline method. This method is implemented to get better advancement in power by comparing our existing architecture. Selected design will optimize in architecture stage by applying low power techniques. The proposed method of the energy optimal design decreases the preventable interior activities so that the vibrant switching power is reduced. The transitory error corrects by error correction unit using identical tag bits from adjoining cache sets. Low level of energy, area and latency are determined by using extra devices. The convincing bit described whether particular tag bits have identical bits to adjoining set or not. The set position bits discuss a higher or subordinate set and finally, way position bits describing a particular cache way has an identical tag bits.

4.1 Proposed N-Way Cache Memory Micro-architecture

Selecting superlative standard cache concepts using an n-way set associative cache represents the micro architecture for our design. Each and every data access of memory address is separated in to three stages, they are offset bits, tag and set index. The preferred data point should contain the set to determine using decoder which laden in preferred address of given set index bits. The output lines of the decoder becomes elevated and nourishing using word line driver, then it's consist couple of cascaded inverters. It is initiating a specific cache set are correlate with n cache points. Sense amplifiers are read through tag and data array which is correlating with n cache lines corresponds to buffers. The corresponding data which is hit is send to processor by mux and output driver.

4.2 Pseudo Code for n Way Architecture

```
{*algorithm
In          : n no of input is assumed
Buffer      : Data signal is forwarded to CPU
*}
{*Initialize phaseb 1;phase 2;phase3;phase 4
Phase 1  : Decode index bit to determine cache set
phase 2  : Vaild bit set in decoded set compare with stored tag bits
Buffer   :  Valid bit set and loaded in sense ampliier buffer
            if any tag matches,
            then tag = cache hit

phase 3:
Hit       : It is hit,then the cache signal is asserted
Buffer    : It is forwarding to CPU
Repeat
  else tag = cache miss
  if tag doesn"t matches
  miss        : Data requested is forwarded
  Hierarchy : moved to next lower memory
Until matches to cache hit
phase 4: empty way
          Data line received from lower memory
          Valid bit is clear
          if no way is empty,a select way is evicted
          closed data
```

4.3 Pseudo Code Explanation

In the n-way architecture, number of inputs is assumed and data signal is forwarded to CPU in buffer and initializing the phases. Phase 1 is used to determine the decode index bit in cache set. Phase 2 is comparing the stored tag bit and the valid bit is set as one and loaded in to sense amplifier. If any tag matches, then it is denoting as tag is equal to cache hit. It is called as hit it is forwarding to CPU. If any tag doesn't match, then the tag is equal to cache miss. Finally, data line is received from lower memory and valid bit is clear. Figure 2 shows about N-WAY Cache Memory.

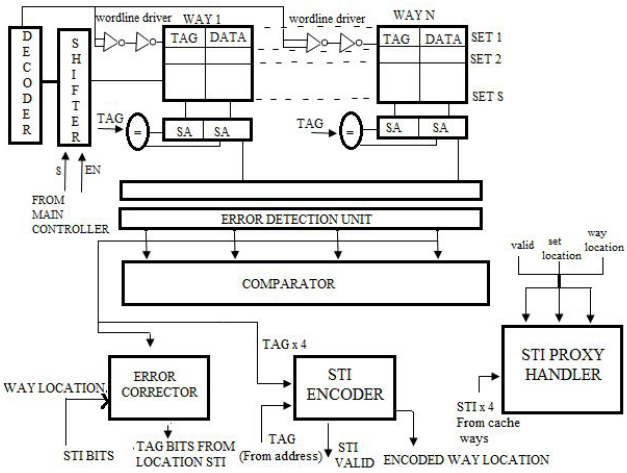

Figure 2. N-WAY cache memory.

5. Experimental Setup

In this sector, our experimental results illustrate that the setup of proposed SST technique is to get better consistency of identical tag bits. Comparing identical tag technique architecture with superlative standard caches by reducing dynamic power consumptions and performance of area coverage. Delay is increased by 0.02% in the proposed architecture technique. Dynamic power and area is reduced by 85% using comparison of Identical Tag Bits. Comparing an Area, Power and Delay of Identical Tag Information Bits with Superlative Standard Technique is shown in Table 1.

Table 1. Comparing an area, power and delay of identical tag information bits with superlative standard techniques

	Identical Tag (Existing system)			N-Way Architecture (Proposed system)		
Logic	Area	Dynamic Power(mw)	Delay (ms)	Area	Dynamic Power(mv)	Delay (ms)
6 Bit Counter	88	0.004429	0.10	88	0.004420	0.10
2 Bit Mux	12	0.000728	0.03	12	0.000728	0.03
19 Bit Mux	117	0.00789	0.05	117	0.00789	0.05
Comparator	76.97	0.001539	0.04	76.97	0.001539	0.04
Parity Encoder	29.57	0.00547	0.05	29.57	0.00547	0.05
Over All Integration	284	0.0156	0.14	183	0.0015	0.16

6. Conclusion

Our paper utilizes identical tag bits to develop tag bits reliability. Dynamic power and area have been verified by RTL compiler using cadence tools for architecture demonstration.

7. Reference

1. 1. Wang S, Hu J, Ziavras S. TRB: Tag Replication Buffer for enhancing the reliability of the cache tag array. IEEE Transactions Very Large Scale Integer. (VLSI) Systems. 2012 Apr; 20(4):643–54.
2. 2. Sai RV, Saravanan S, Anandkumar V. Implementation of a novel data scrambling based security measure in memories for VLSI circuits. Indian Journal of Science and Technology. 2015 Dec; 8(35):1–6. DOI: 10.17485/ijst/2015/v8i35/86798.
3. Hameed F, Bauer L. architecting on-chip dram cache for simultaneous miss rate and latency reduction. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 2016 Apr; 35(4):651–64.
4. Subha S. An algorithm for variable cache ways. 2013 International Conference on Advances in Technology and Engineering (ICATE); 2013Jan. p. 1–3.
5. Gulur N, Mehendale M, Manikantan R, Govindarajan R. Bi-modal DRAM cache: Improving hit rate, hit latency and bandwidth microarchitecture (MICRO). 2014 47th Annual IEEE/ACM International Symposium; 2014 Dec. p. 38–50.
6. Hong S, Kim S. Designing a resilient L1 cache architecture to process variation-induced access-time failures computers. IEEE Transactions on Computers. 2016 Jan; PP(99):1–1.
7. Farbeh H, Miremadi SG. PSP-cache: A low-cost fault-tolerant cache memory architecture. 2014 Design, Automation and Test in Europe Conference and Exhibition (DATE); 2014 Mar. p. 1–4.
8. Reviriego P, Pontarelli S, Ottavi M, Maestro JA. Errors Device. 2014 Sep; 14(3):935–7.
9. Seongil O, Kwon S, Son YH, Park Y, Ahn JA. cidr: a cache inspired area-efficient dram resilience architecture against permanent faults. Computer Architecture Letters. 2015 Jan–Jun; 14(1):17–20.
10. Valls JJ, Sahuquillo J, Ros A, Gomez ME. Approach Parallel; 2015 Mar. p. 182–9.
11. Jiang L, Guoqing W, Tao H, Jianya C, Yunjie L. Modeling the sojourn time of items for in-network cache based on LRU policy. China Communications. 2014 Oct; 11(10):88–95.
12. Saravanan S, Vennelakanti S. Design and analysis of low power memory built in self test architecture for SoC based design. Indian Journal of Science and Technology. 2015 Jul; 8(14):1–5. DOI: 10.17485/ijst/2015/v8i14/62716.
13. Hajimiri H, Mishra P, Bhunia S. Dynamic cache tuning for efficient memory based computing in multicore architectures. 2013 26th International Conferences on VLSI Design and the 12th International Conference on Embedded Systems; 2013 Jan. p. 49–54.
14. Bhattacharya K, Ranganathan N, Kim S. A framework for correction of multi-bit soft errors in L2 caches based on redundancy. IEEE Transactions Very Large Scale Integer (VLSI) Systems. 2009 Feb; 17(2):196–206.