Soft error tolerant Content Addressable Memories (CAMs) using error detection codes and duplication

Juan Antonio Maestro a, Pedro Reviriego a,b, Sanghyeon Baeg b, Shijie Wen c, Richard Wong c

a Universidad Antonio de Nebrija, C/Príncipe, 55, 28040 Madrid, Spain
b School of Electrical Engineering and Computer Science, Hanyang University, Republic of Korea
c Component Engineering Group, Cisco Systems Inc., 170 W. Tasman Dr., San Jose, CA 95134, USA

A R T I C L E   I N F O

Article history:
Available online 28 October 2013

Keywords:
Error detection codes
Content Address Memories (CAMs)
Soft errors
Reliability

ABSTRACT

Soft Errors are becoming a major concern for modern computing systems. Memories are one of the elements affected by soft errors, which cause bitflips in some of the cells. A number of techniques such as the use of Error Correction Codes (ECCs), interleaving or scrubbing are utilized to mitigate the effects of soft errors on memories. Content Addressable Memories (CAMs) pose additional challenges, as many of those protection techniques are not applicable to CAMs. In this paper, a novel protection technique for CAMs is proposed, showing a convenient way to tackle false positives and negatives, and quantitatively studying the achieved benefit in reliability.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

Soft errors are becoming a critical issue in the design of modern computing systems [1]. These phenomena cause bit flips in memory cells, flip-flops or combinational gates. Those errors can lead to system failure and data corruption.

As memories are widely used in computing systems, bit flips caused by soft errors are a major concern [2]. Several techniques are used to ensure that those errors do not cause data corruption on memories [3]. For example, per-word Error Correction Codes (ECCs) are typically used so that a single bit flip on a memory bit can be detected and corrected. Scrubbing is also used to avoid error accumulation by periodically reading the memory words and correcting errors. As technology scales, radiation particles can corrupt more than one bit, a phenomenon known as Multiple Cell Upset (MCU). The cells affected by an MCU are typically close together [4] and are likely to corrupt more than one bit of the same word, causing an error that cannot be corrected by the ECC. To mitigate the effects of MCUs, interleaving is commonly used in memories. With interleaving, the bits of a logical word are physically apart, so that an MCU causes only single errors on multiple words that can be corrected by the ECC.

Content Addressable Memories (CAMs) are a special type of memories used in a variety of applications in computing and communications [5]. In particular they are commonly found in computer networking equipment where they are used to perform different task such as packet classification [6] or route lookup [7].

A CAM stores keys associated with values in pairs (key and value) and when a search key is presented to the memory, the address of the key that matches that search key is obtained. Then the associated value can be read using that address from a memory where the values are stored. To perform the search of the key that matches the input key, each memory word is compared in parallel with the input key. This requires additional logic compared to a standard memory. The overall structure of a CAM is shown in Fig. 1. The input data which is the key to be searched is compared to each memory word and if all bits match, the corresponding Match Line (ML) is activated. Then, a priority encoder computes the address of the first key that matches the input key and that address is used to obtain the value associated with the key.

Due to its nature, error correction codes are not appropriate for CAMs. For example, let us assume that an error correction code is used in each memory key and that there is one error in one bit of key $i$. Then, if we search for the value of that key $i$ (including the ECC bits) there will be a mismatch on the erroneous bit, and therefore the search would fail.

Interleaving is also difficult to implement in CAMs. An example of the bit placement of a memory using interleaving is shown in Fig. 2. The bits that belong to the same word are physically distant, but in a CAM all the bits in a word have to be connected by the corresponding matchline (see Fig. 1). So, when the bits of a word are physically distant, the matchline design and routing becomes almost impossible. This limits the applicability of interleaving in CAMs.

Given the limited applicability of existing memory protection techniques to CAMs, a number of alternative approaches have been proposed. For example, in [8], a modified matchline design is...
proposed such that one bit mismatch is allowed to produce a word match. This is combined with ECC in each key, so that single errors do not cause a failure in the search. In [9], the use of a parity bit and parity computation logic per key is proposed in order to detect errors immediately. A commonly used scheme is to add an independent ECC to the keys. This means that the key itself does not include the ECC bits that are stored in a conventional memory. Then periodically the key and ECC bits are used to detect and correct errors.

In this paper, an alternative technique to protect CAMs is introduced. The objective is that the proposed technique can be used on regular CAM devices without adding extra requirements.

2. Effects of soft errors in CAMs

A soft error that causes bit flips on a CAM key can produce two effects: a false negative and/or a false positive [1]. A false negative occurs when a search does not return a match because the associated key has been altered by a bitflip. On the other hand, a false positive occurs when an erroneous match is performed with an key affected by a bitflip (the bitflip has transformed a completely different word into the search key).

False positives can be effectively removed using error detection codes. To discuss the use of these codes, let us first introduce some definitions.

Let us define \( n \) as the total number of bits per CAM key. Of those, \( n-k \) bits will contain the original data pattern and \( k \) will be redundant bits to implement the error detection code. For a given code, let us define \( d_{\text{min}} \) as the minimum Hamming distance provided among any two \( n \)-bit data words.

Then, if an error detection code of minimum distance \( d_{\text{min}} \) is used in a CAM, errors of up to \((d_{\text{min}} - 1)\) bits in a key cannot cause a false positive with any valid search pattern. This is a direct consequence of the error detection capabilities of a block code of distance \( d_{\text{min}} \).

In its simplest form, error detection can be implemented with a parity bit which provides a minimum distance \( d_{\text{min}} \) of 2. For a CAM, this is combined with another code for error correction. However, as discussed in the introduction, MCUs are becoming increasingly common, and as interleaving is not an option for CAMs, multiple errors per word will occur. To deal with multiple errors, more sophisticated codes can be used. For example, Single Error Correction Double Error Detection (SEC-DED) codes [12], which are typically used to protect memories, are a good option. Those codes have a minimum distance \( d_{\text{min}} \) of 4. In a CAM, the codes are used only for error detection and no attempt to correct errors is made. Therefore, a SEC-DED code will detect all errors that affect three or less bits in a key. Using SEC-DED codes can be effective against the most common type of MCUs, those with three or less errors. In this case, the cost is higher than for parity bits, but still a small percentage of the number of bits in each key, as illustrated in Table 1.

So far, we have shown how error detection codes can be effective to avoid false positives caused by single or multiple bit flips on CAMs. For some applications in which a false negative does not cause a failure but simply a slower processing (for example in a write-through cache or in a route lookup in an Ethernet switch), this protection may be sufficient.

However, in applications where false negatives must be avoided, additional protection mechanisms are needed. For single bit errors, there are two different situations to be considered: (i) when negatives are mostly due to soft errors (most of the search keys should be in the memory in normal conditions) or (ii) when after a negative the CAM is not used for a large number of cycles because of the slower processing associated with the negative.

In both cases, assuming that error detection codes with \( d_{\text{min}} > 2 \) are used to deal with false positives, then false negatives can be effectively removed by searching all patterns at distance of one of the original pattern. If one of those patterns matches a key, then that was the key affected by a soft error. If none of those patterns returns a match, then the negative was not caused by a soft error (meaning that the pattern was not initially in the memory). For a CAM with \( n \)-bit keys, this requires \( n \) additional accesses. This additional time may be acceptable when this situation rarely occurs, or if the CAM is not going to be used during those cycles anyway (as discussed before).

A drawback of this scheme is that the detection capability for false positives is reduced, as now any pattern at a distance of one can also return a match. This reduction occurs when there are at

![Fig. 1. Block diagram of a content addressable memory.](image)

![Fig. 2. Example of bit placement in a memory that implements interleaving.](image)

<table>
<thead>
<tr>
<th>Data bits</th>
<th>Check bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>16</td>
<td>6</td>
</tr>
<tr>
<td>32</td>
<td>7</td>
</tr>
<tr>
<td>64</td>
<td>8</td>
</tr>
<tr>
<td>128</td>
<td>9</td>
</tr>
<tr>
<td>256</td>
<td>10</td>
</tr>
</tbody>
</table>
least two keys affected by soft errors which are at a distance of one. In this case, one of the affected keys can be retrieved as a false positive of the other one.

The previous algorithm requires a number of cycles to perform the correction process. For applications in which false negatives are not tolerated and this extra delay cannot be allowed, additional redundancy must be added. In the next section, a novel technique is proposed to handle this situation.

3. A novel technique for CAM protection

When extra levels of reliability are required, more redundancy has to be used. Classical approaches require the triplication of the information in order to protect against soft errors. However, this can produce an unacceptable overhead, hence requiring alternative solutions. Therefore, our proposal is to use a redundancy level lower than the one produced by triplication, which can be applied to regular CAMs.

In this way, a feasible solution would be to duplicate the keys in the memory in such a way that if one copy is affected by errors, the other one still produces a match. This would eliminate the possibility of false negatives. In order to avoid false positives, each key would have error detection codes, as discussed in the previous section.

Also, in order to protect against MCUs, the location of the copies should be carefully chosen to avoid having both of them affected by the same MCU. A scheme like the one shown in Fig. 3 could be used to ensure a large physical distance between copies, therefore avoiding the local effect of MCUs. A simple relationship between the addresses of the two copies can be used. In this case, the most significant bit of the address identifies the copy, while the rest of the bits would be the same in both copies.

There may be applications in which the cost of doubling the number of keys in the CAM may not be acceptable. In those cases, some protection can be achieved by using the keys that are not used at a given instant to store duplicates of existing keys. In the cases where the number of keys is smaller than half the capacity of the CAM, then the previous discussion about avoiding false negatives applies. However, if the occupancy occasionally becomes larger than 50% of the CAM, there will only be space to duplicate some of the keys and false negatives would be reduced but not completely avoided.

For example, if the occupancy is 60%, there will be duplicates for only 40% of the keys, thus, on average, one out of three keys will not be protected against false negatives. If keys to be duplicated are randomly selected, false negatives would affect keys regardless of how often they are searched. However, if only the most commonly searched keys are duplicated, false negatives would concentrate on the keys that are less frequently used, thus reducing the number of searches that suffer a false negative. For example, if 20% of the keys concentrate 80% of the searches, then even if only those are duplicated, false negatives could be avoided in 80% of the cases. Duplication of 20% of the keys can be done when the CAM occupancy is up to 80%.

A further complication of this approach is how to keep track of which keys are being used for duplicates and which hold a unique copy. The study of algorithms to manage the duplication and replacement policies in this configuration is left for future study.

4. Reliability analysis

In this section, the reliability of the proposed technique will be put in perspective, using the Mean Time To Failure (MTTF) as a figure of merit.

4.1. False positives

First, the case of false positives will be studied. Since, as previously commented, all patterns are at a distance of \( d_{\text{min}} \) then at least \( d_{\text{min}} \) bits in the same word should be affected by soft errors in order to produce a failure. Therefore, up to \( (d_{\text{min}} - 1) \) bitflips can be tolerated, which is equivalent to the situation in which words are protected with a redundancy code able to correct \( (d_{\text{min}} - 1) \) errors.

The MTTF of this situation has been studied in [13], and for a memory with \( M \) words can be approximated by:

\[
\text{MTTF}_{\text{memory}} = \frac{e^{\sqrt{(d_{\text{min}} - 2)}/C_2}}{\lambda (1 + \frac{1}{e^{\sqrt{(d_{\text{min}} - 2)/C_2}}}) 
\]

where \( \lambda \) is the error arrival rate for the whole memory.

However, this is a conservative approach, since in the CAM situation these errors may not lead to false positives, for example if the pattern is not searched, or if the valid key has a higher priority than the false positive and therefore the latter would not be retrieved in the search process. So, expression (1) is a lower bound of the CAM reliability when affected by false positives:

\[
\text{MTTF}_{\text{CAM \ false \ positives}} > \text{MTTF}_{\text{memory}}
\]

If MCUs are present then the analysis above is not valid, as a single MCU can cause multiple bit errors on a single word. In that case, the reliability analysis is similar to that of a memory with a redundancy code able to correct several errors (as described before), but that does not use interleaving. Several approximations to describe the MTTF of this situation can be found in [14].

4.2. False negatives

When duplication is used, a false negative can only occur if the two copies are corrupted. In this case, the analysis of the reliability resembles that of a memory protected with a single error correction code.

Let us assume that there are \( i \) corrupted keys at a given time in a CAM with duplicated keys. Then the probability that a new bitflip causes a false negative is \( i/(2M) \), where \( M \) is the maximum number of keys in the CAM and \( 2M \) is the size. If that false negative does not cause a failure, then it can affect a new key or an already corrupted key. In the first case, there will be \( i + 1 \) corrupted keys, while in the second one there will be only \( i \).

These probabilities are similar to those in a memory protected with SEC-DED that has \( 2M \) words. However, in this case, a new
error always causes either a failure or increases the probability of failure for the following events, something that does not happen in the CAM scenario. This means that the MTTF of the memory protected with SEC-DED (obtained from (1) by making \(\delta_{\text{min}} = 2\)) can be used as a lower bound for the reliability of the CAM with duplication:

\[
\text{MTTF}_{\text{CAM}}^{\text{false neg}} = \frac{\sqrt{2} \cdot \Gamma(1 + \frac{1}{2}) \cdot M^2}{\lambda} \]  

For MCUs, when duplication ensures that an MCU cannot affect both copies of an key, the reliability analysis can be approximated using the MTTF expression in [15]. In this way, the reliability of a CAM suffering MCUs can be bounded by the reliability of a CAM suffering single errors, with a modified event arrival rate.

5. Simulation results

Simulation experiments have been conducted in order to measure how a CAM would handle false negatives in the presence of single errors. In this way, several CAMs (up to 4M words) whose size is double the expected number of keys have been considered. For each CAM, 100,000 simulation experiments have been recreated, and the Mean Time To Failure measured. An event arrival rate of \(\lambda = 1\) has been used (1 soft error per memory and day). The experimental results have been then compared with the theoretical reliability prediction provided by expression (3). The results and also the relative error are shown in Table 2. Both simulation and theoretical results are very similar, which indicates that expression (3) is a good model to predict reliability in the mentioned environment.

6. Conclusions

In this paper, a novel protection technique based on the use of errors detection codes and duplication has been proposed to increase the reliability of Content Addressable Memories (CAMs). The proposed technique can provide MCU protection both for false positives (through the use of error detection codes) and false negatives (using duplication), something that simpler techniques, as using only parity bits, cannot achieve. The overhead involved in this technique is a little more than duplication of the original CAM size, which is convenient when compared with traditional redundancy mechanisms as triplication.

The reliability analysis of the proposed techniques in terms of MTTF has also been discussed, showing that existing reliability models can be applied. This is convenient, as it simplifies the evaluation of the protection techniques in a given design.

The presented technique can protect effectively against MCUs, which are becoming increasingly common as technology scales. Previous approaches have focused on the single bit error case and they provide little or none protection against MCUs.

Finally, the technique presented in this paper can be directly applied to regular CAMs, and no specific design changes are required, what makes this approach very convenient for existing devices.

References


J. Antonio Maestro received the M.Sc. degree in physics and the Ph.D. degree in computer science from Universidad Complutense de Madrid, Madrid, Spain, in 1994 and 1999, respectively. He has served both as a Lecturer and a Researcher at several universities, such as the Universidad Complutense de Madrid; the Universidad Nacional de Educación a Distancia (Open University), Madrid; Saint Louis University, Madrid; and the Universidad Antonio de Nebrija, Madrid, where he currently manages the Computer Architecture and Technology Group. His current activities are oriented to the space field, with several projects on reliability and radiation protection, as well as collaborations with the European Space Agency. Aside from this, he has worked for several multinational companies, managing projects as a Project Management Professional and organizing support departments. He is the author of numerous technical publications, both in journals and international conferences. His areas of interest include high-level synthesis and coynthesis, signal processing, and real-time systems, fault tolerance, and reliability.

Pedro Reviriego received the M.Sc. and Ph.D. degrees in telecommunications engineering from the Technical University of Madrid, Madrid, Spain, in 1994 and 1997, respectively. From 1997 to 2000, he was an R&D Engineer with Teldat, Madrid, working on router implementation. In 2000, he joined Massana to work on the development of 1000BaseT transceivers. During 2003, he was a Visiting Professor with the University Carlos III, Leganés, Madrid. From 2004 to 2007, he was a Distinguished Member of Technical Staff with the ISI Corporation, working on the development of Ethernet transceivers. He is currently with the Universidad Antonio de Nebrija, Madrid. He is the author of numerous papers in international conference proceedings and journals. He has also participated in the IEEE 802.3 standardization for 10 GBaseT. His research interests include fault-tolerant systems, performance evaluation of communication networks, and the design of physical layer communication devices.
Sanghyeon Baeg received the Ph.D. degrees in electrical and computer engineering from the University of Texas at Austin, Texas, U.S.A in 1994. From 1994 to 1997 he was a Staff Researcher at Samsung Electronics Co., Korea. In 1997, he joined Cisco Systems, Inc. San Jose, CA and worked as a Hardware Manager. Since 2004, he has been working as Professor at Hanyang University, Korea in Dept. of Electronics and Communication Engineering. His work has focused on test and reliability issues for memory devices, low power SRAM/CAM, and VLSI DFT implementation and methodologies. He currently serves as the president of Institute of Semiconductor Test of Korea.

Shi-Jie Wen received his Ph.D in Material Engineering from University of Bordeaux I in 1993. He joined Cisco Systems Inc., San Jose, CA in 2004, where he has been engaged in IC component technology reliability assurance. His main interest is in silicon technology reliability, such as SEU, WLR, and complex failure analysis, etc. He is a member of DFR, SEU core teams in Cisco. Before Cisco, he worked in Cypress Semiconductor where he was involved in the area of product reliability qualification with technology in 0.35 um, 0.25 um, 0.18 um, 0.13 um and 90 nm.

Richard Wong is a Staff Engineer at Cisco Systems Inc in San Jose, CA since 2006. He received his M.S. degree in electrical engineering from Santa Clara University and his B.S. degree in chemical engineering from UC Berkeley. He is currently responsible for IC component technology reliability assurance. His areas of interest include Single Event Upset, Electro Static Discharge, Wafer Level Reliability, failure analysis and reliability modeling. Prior to Cisco, he had worked on technology and reliability for ASICs, FPGAs, TCAMs and memories. He has 18 patents and has authored or coauthored over 80 papers.