Reducing DUE-FIT of Caches by Exploiting Acoustic Wave Detectors for Error Recovery

Gaurang Upasani[†] Xavier Vera[♭] Antonio González[†♭]
[†] Dept. d'Arquitectura de Computadors, Universitat Politècnica de Catalunya, Barcelona
[♭] Intel Barcelona Research Center, Intel Labs, Barcelona
gaurang@ac.upc.edu {xavier.vera,antonio.gonzalez}@intel.com

Abstract—**Cosmic radiation induced soft errors have emerged as a key challenge in computer system design. New techniques for detecting errors in the logic and memories that allow meeting the desired failures in-time (FIT) budget in future chip multiprocessors (CMPs) are essential. This paper targets the DUE problem in write-back data caches. We analyze the cost of protection against single bit and multi-bit upsets into caches. Our results show that the proposed mechanism can reduce the DUE to "0" with minimum area, power and performance overheads.**

I. Introduction

The exponential growth rate of on-chip transistors, the lower voltages, and the shrinking feature size make current processors vulnerable to transient faults caused by particle strikes. They do not cause permanent failure in the hardware and hence are termed *soft errors* (SER) in the literature.

The soft error problem is projected to become a major challenge when designing future chip multiprocessors (CMPs). The total failures-in-time (FIT) per chip will increase due to larger arrays and increased number of cores per area [6]. Hence meeting the desired FIT budget for current and future CMPs pose a major challenge.

Architecturally, soft error detection and correction mechanisms create two categories of errors: silent data corruption *SDC* and detected unrecoverable errors *DUE*. Error detection codes (i.e. Parity) are used in memories to reduce the SDC-FIT. Error correction codes (i.e. Single error correction), are used to provide recovery in memory which can reduce the DUE-FIT rate. *DUE problem arises when the structure (i.e. parity protected L0 data cache) does not have error correction capacity but only error detection capacity.*

DUE is the largest contributor towards total soft error rate for write-back caches. As a consequence, designers are forced to use stronger codes (to provide detection as well as correction), which implies higher costs in terms of area, power and latency. Moreover, caches closer to the core (i.e. L0 cache) are usually protected only with parity per byte. However, to have correction capability each byte should be protected with ECC. Implementing ECC for every byte is complex and expensive. Hence, instead of providing ECC for each byte in a cache block, to reduce the cost of protection designers opt to protect cache block with ECC. But caches closer to the core have a lot of partial write operations. Having ECC at cache block level will result into an increase in read-modifies-writes operations. This incurs huge performance penalty.

Instead of relying on symptom based techniques (i.e. error codes) to detect errors, a new direction that is growing in interest by the research community is to detect the actual particle strike rather than its consequence. The proposed idea in [9] consists of deploying a set of detectors on silicon that would be in charge of detecting the particle strikes. By deploying acoustic wave detectors we guarantee to detect all the errors occurring on the caches, providing 0 SDC. Upon detection, a hardware or software mechanism would trigger the appropriate recovery action for correction.

This paper targets the DUE problem in data caches. To provide error correction, the system should be able to accurately locate the error. To achieve 0 DUE target, the architecture should be able to recover from all the errors that are detected. This can be done by exploiting the localization accuracy of acoustic wave detectors to detect and correct the errors. *Once the accurate location is found, we correct the error by flipping the bit.* And whenever this is not possible, to prevent the corruption of the architectural state, the solution takes advantages of the parity codes already deployed for error detection of hard errors.

The principal contribution of this paper is that it proposes a light-weight technique that uses acoustic wave detectors for error correction in data caches which is effective for both single and multi-bit upsets.

The rest of the paper is structured as follows: Section II explains how using acoustic wave detectors we can detect the errors in data caches. Section III details the DUE problem in L0 cache. In this section we evaluate the architecture in terms of DUE improvement we can achieve compared to number of detectors. We also evaluate how is it beneficial to combine traditional error detection techniques (i.e. parity) with detectors to improve DUE. In Section IV we extrapolate the idea and show how can it handle the case of multi-bit upsets. Section V reviews some relevant work in the same direction. Finally, a summary of main conclusions is presented in Section VI.

II. Acoustic Wave Detectors: Detection and Localization

The interaction of a high-energy particle with a silicon nucleus results into a cloud of phonons, transforming the cosmic energy to sound. The proposed architecture makes use of cantilever beam like structures as an acoustic wave detector to detect particle strikes through the sound they generate, as shown in Figure 1. The work done in [9] describes various properties of the detector such as the device dimensions, sensitivity etc. in detail.

The fundamental idea is to detect the particle strikes via mechanical deflection of acoustic wave detectors. The work in [9] proposes an architecture that not only detects but also locates particle strikes on a processor based on acoustic wave detectors. The architecture includes an analysis of various

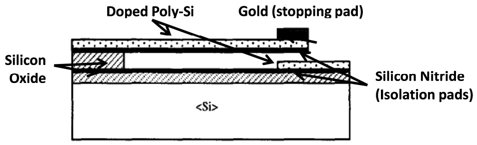


Fig. 1: Cantilever sensing device [9]

design space parameters such as (i) how many acoustic wave detectors are required to be able to accurately locate the particle strike?, (ii) where should the acoustic wave detectors be placed?, (iii) what would be the accuracy of the found location?, and (iv) what would be the detection latency?

The method requires a system of minimum 3 detectors. The estimation of the location is a three stage process. The first stage is about placing the acoustic wave detectors. They can be placed on or off the chip but on the same silicon surface at known coordinates. Second stage is about measuring the time difference of arrival (TDOA) of the sound wave between pairs of detectors through the use of time delay estimation. In the last stage, the estimated TDOAs are transformed into range difference measurements between the detectors. This gives a system of nonlinear hyperbolic equations. We linearize these equations using Taylor series expansion. Finally, by using iterative Gauss Newton interpolation method we solve the linearized hyperbolic equations. The discussion also models the effect of the sampling errors in the measurements of the TDOA. At the end of the process the solution results in the estimated position of the particle strike. We use Circular error probability (CEP) statistics to express the area of the error distribution of the final estimation of the position. Because of the sampling errors, the final outcome of the system level algorithm is an estimated coordinates of the location of particle strike and the CEP radius. This translates that the resolution of the final location can be equivalent to the area of a circle with CEP radius that can be mapped to one bit or multiple bits.

III. DUE estimation with acoustic wave detectors for L0 data cache

In this section we will explain how we can solve the DUE problem for L0 data caches by using acoustic wave detectors.

A. Acoustic wave detectors for DUE problem

As described in Section II acoustic wave detectors can be used to detect the errors on L0 data cache. However, to reduce the DUE we have to correct the error by flipping the erroneous bit. To flip the erroneous bit we need to know the exact location of the bit.

In this section we demonstrate the utility of the cantilever detectors by detecting and locating particle strikes in the L0 data cache of a Core^{TM}i7-like processor. The cache is 32KB, 4-way and has rectangular shape with the surface area of $1mm^2$. 1 bit SRAM cell area is equivalent to $0.65um^2$ [1], [2]. Monte-Carlo experiments consisting of 1048 randomly distributed particle strike locations in space and time have been performed.

The main objective of the experiments is to obtain the accurate location of the particle strikes using acoustic wave detectors incurring minimum area, power and performance penalty. Accurate localization of the error makes recovery easy. The accuracy of location depends on the several factors such as, (i) Number of detectors deployed on the cache (ii) Locations of detectors on the cache (iii) Number of TDOA equations that are used for localization algorithm and (iv) Sampling frequency.

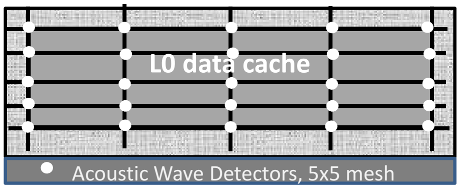


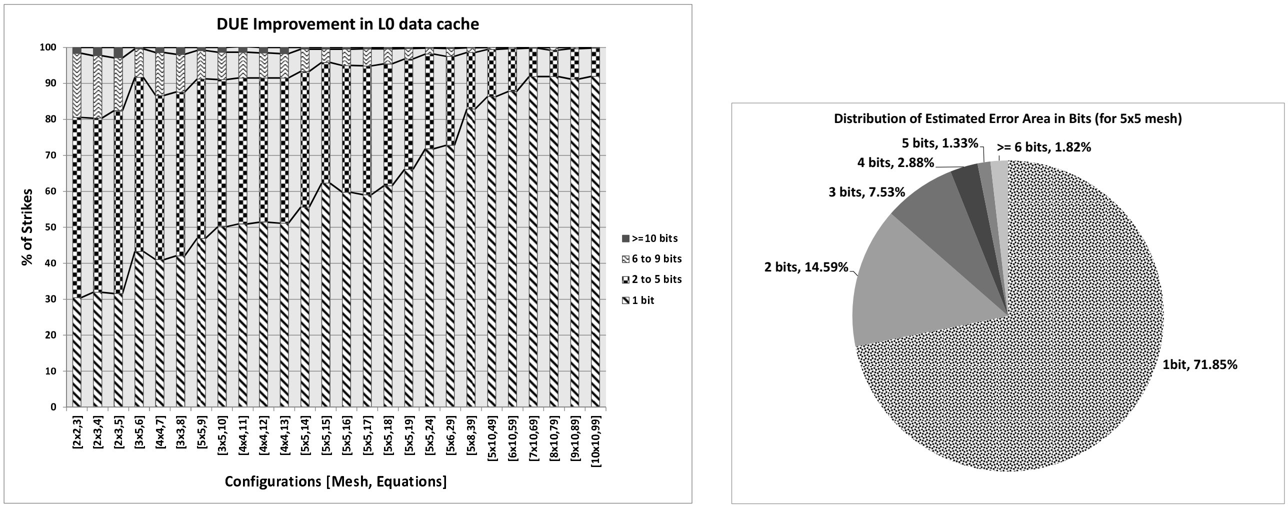
Fig. 2: Placement of detectors in a 5×5 mesh formation

After experimenting various configurations, we decided to put the detectors in a mesh formation on the surface of the cache. Figure 2 shows a 5×5 mesh. Each node in the mesh represents a detector.

From [9] we know that accuracy of the location can be improved by either increasing the sampling frequency or by solving more than 2 TDOA equations. Increasing sampling frequency reduces the effect of sampling noise and hence improves the accuracy of the estimated location. In this work, we have fixed the sampling frequency to 4GHz.

Recall from [9] that one detector can detect a particle strike occurring anywhere in the area of $78mm^2$, which is the area of a circle with a radius of 5mm (i.e. the detection range for one detector). The area of L0 data cache is $1mm^2$. Hence, for L0 data cache, in a mesh with N detectors we can built $N-1$ TDOA equations. Out of this $N-1$ equations, we choose the equations formed by the detectors that are closer to the source of the particle strike as they give more accurate estimation. Solving an overdetermined system of equations ($\geq$3 TDOA equations) reduces the effect of sampling noise and improves precision. To have the overdetermined system of equations we tried different mesh configurations starting from the most basic overdetermined system (3 TDOA equations) with 4 detectors in 2×2 mesh upto 99 TDOA equations (i.e. 100 detectors in a 10×10 mesh).

Figure 3(a), shows the best choices for the given number of TDOA equations that we solve, out of all the mesh configurations that can be used to construct those many TDOA equations. It summarizes the improvement in the DUE rate in each case. DUE improvement signifies that out of 1048 strikes, how many times we can locate the actual strikes within the area granularity of 1 bit. We can see that by increasing number of TDOA equations in solving the localization algorithm we significantly improve the DUE rate. As we keep solving more TDOA equations, the DUE improvement curve soon starts to saturate. Using more detectors increases the over all cost and complexity in solving the TDOA equations. Observing the cost of solution against the DUE improvement achieved, we conclude that the best trade-off for L0 data cache is obtained by configuring a 5×5 mesh with 25 detectors (shown in Figure 2) and solving for 24 TDOA equations. This configuration results into a **71.85%** improvement in DUE.



(a) DUE improvement in L0 data cache

(b) Quantification of error area for 5×5 mesh

Fig. 3: Assesment of DUE improvement in L0 data cache

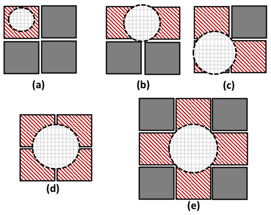


Fig. 4: Circular error area mapping to bits for (a)1-bit, (b)2-bits, (c)3-bits (d)4-bits and (e)5-bits

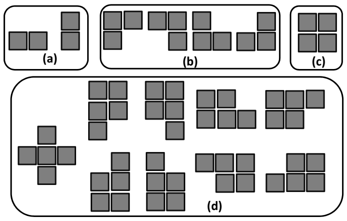


Fig. 5: Estimated error area granularity patterns for (a)2-bits, (b)3-bits, (c)4-bits and (d)5-bits

B. Combining Error Codes with Acoustic wave detectors

If an L0 data cache is protected with only acoustic wave detectors in 5×5 mesh, 71.85% of the times we can exactly locate the upset bit, we call this $P_{1bit_{AWD}}$. A further quantification as shown in Figure 3(b) reveals that for 14.59%, 7.53%, 2.88% and 1.33% of the times we can locate the error at the granularity of 2 bits, 3 bits, 4 bits and 5 bits respectively. We call them $P_{2bit_{AWD}}$, $P_{3bit_{AWD}}$, $P_{4bit_{AWD}}$ and $P_{5bit_{AWD}}$ respectively.

$$DUE_{(AWD)} = P_{1bit_{AWD}} = 71.85\% \tag{1}$$

By using only acoustic wave detectors in L0 data cache we can improve the DUE by **71.85%** as shown in Equation 1.

Interestingly, we noted that the granularities of error area (i.e. circular area with CEP radius) obtained by acoustic wave detectors are mapped to bits in specific patterns as shown in Figure 4. The circle in the Figures 4(a-e) show the estimated error area obtained by localization algorithm. The bits that are overlapped or intersected by this circle are also shown in Figure 4. For single bit upsets, one of the bits covered by this circular area is erroneous. Using this mapping, we show all the possible error area patterns[1] for bit granularities of 2 to 5 bits in Figure 5.

Because of this characteristic, *we can further improve the DUE if we can exactly isolate the erroneous bit out of the error area granularities of 2-5 bits by combining acoustic wave detectors with error codes*. To detect hard errors already parity codes can be deployed for each block or for every byte in a block. Now we will see how we can take advantage of combining acoustic wave detectors with parity codes.

1) Acoustic Wave Detectors + Parity per Block: Let's assume that each cache block is protected by parity bits. Figures 6(a-e) show the error area granularity from 2-5 bits obtained by acoustic wave detectors.

In the case of **2-bit patterns**, we assume that 2-bit patterns shown in Figure 5(a) are equiprobable(i.e. probability of having each of them is 50%). If both the bits are located in the same cache block as shown in case 1 of Figure 6(a), we will not be able to locate the exact bit. However, if the 2 bits are located as shown in case 2 of Figure 6(a) we will be able to locate the exact bit that was upset. This means that out of 2 cases involving 2-bit error area granularity we can always detect the patterns, that are similar to case 2. **Parity per block can improve the 2-bit contribution towards DUE by further $50\% \times \mathbf{P_{2bit_{AWD}}}$.**

Likewise, in the case of **3-bit patterns** all the 3 bits are located in two different blocks as shown in all the cases of Figure 6(b). We will be able to locate the error only when the erroneous bit is the only bit lying in a different cache block out of the 3 bits of error area. Again we consider all the 4 cases shown in Figure 5(b) are equiprobable(i.e. probability of having each case is 25%). Furthermore, we can detect the exact location of the error only when the error is in specific 1

[1] Not to be confused with multi-bit upset patterns

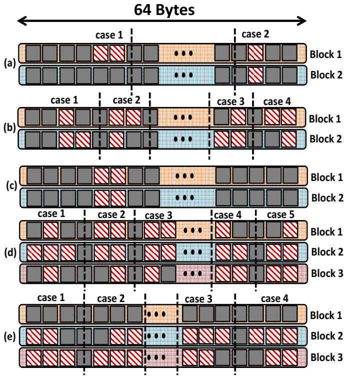


Fig. 6: Parity per block for (a) 2-bit, (b) 3-bit, (c) 4-bit and (d,e) 5-bit patterns

bit out of 3 bits in each case. This means that we can improve DUE for each case of Figure 6(b) by $(1/3) \times 25\%$. **This yields an overall improvement for the 3-bit contribution towards DUE by $\mathbf{34\% \times P_{3bit_{AWD}}}$.**

In the case of **4-bit pattern** as shown in Figure 6(c) it is not possible to locate the exact erroneous bit.

Figures 6(d) and (e) show the **5-bit patterns**. Here also we consider that all the patterns shown in Figure 5(d) are equiprobable. Hence, each can occur with a probability of 11.12%.

Similar calculation in the case of 5-bit pattern, shows that for the case 1 of Figure 6(d) when the strike is either in the bit that is in block 1 or block 3, it is possible to locate the exact error. This means we can correct the error if it is only in either of the two bits out of the 5 possible bits. The probability of locating exact error in case 1 of Figure 6(d) like patterns is $(2/5) \times 11.12\%$.

And as shown in other cases of Figure 6(d), it is possible to locate the exact error only when the erroneous bit is in a different block and it is the only bit of the 5 bits. This means we can correct the error if it is in only one specific bit out of the 5 possible bits. The probability of locating exact error bit in case 2,3,4 and 5 of Figure 6(d) is $(1/5) \times 11.12\%$ each. As they are all equiprobable the improvement is $(4/5) \times 11.12\%$.

Also in the occurrence of patterns shown in all the cases of Figure 6(e) it is not possible to locate the exact bit that was upset. As each block contains two or more bits that can be erroneous.

Putting it all together, **for 5-bit pattern, parity per block on top of acoustic wave detectors can increase the DUE improvement by $\mathbf{(2/5) \times 11.12\% + (4/5) \times 11.12\%}$ giving overall DUE improvement of $\mathbf{14\% \times P_{5bit_{AWD}}}$.**

$$\begin{aligned} DUE_{(AWD+Parity_{block})} =& P_{1bit_{AWD}} + 50\% \times P_{2bit_{AWD}} + \\ & 34\% \times P_{3bit_{AWD}} + 14\% \times P_{5bit_{AWD}} \\ =& 81.89\% \end{aligned} \tag{2}$$

Hence, deploying *parity per block + acoustic wave detectors* in L0 data cache will improve the DUE by $\mathbf{81.89}\%$ as calculated in Equation 2.

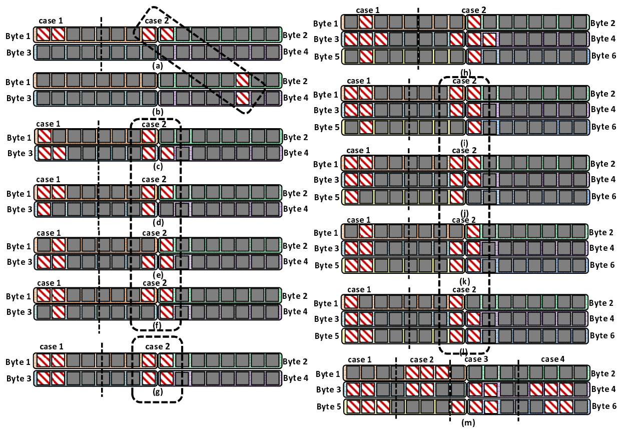


Fig. 7: Parity per byte for (a,b) 2-bit, (c-f) 3-bit, (g) 4-bit patterns and (h-m) 5-bit patterns

2) Acoustic Wave Detectors + Parity per Byte: Now we will see the case when each byte in a cache block is protected by parity bits along with acoustic wave detectors. A cache block in L0 data cache of a Core^{TM}i7-like processor has 64 Bytes. Figures 7(a-m) show all the possible cases for locating the erroneous bit for 2-bit, 3-bit, 4-bit patterns and 5-bit patterns.

As it is obvious that if all the estimated error bits are in the same byte, we will not be able to locate the exact bit with upset. But if the bits are in different bytes it is possible to locate the exact erroneous bit. All **2-bit patterns** are shown in Figures 7(a) and (b). For the patterns as in the case 1 of Figure 7(a), as both the error area bits are in the same byte we cannot locate the upset bit. But for the patterns similar to case 2 of Figure 7(a) or patterns similar to Figure 7(b) both the bits are into two different bytes and as we have parity at byte level, we can exactly pin-point the upset bit out of the two bit error area.

For a 64 byte block the probability of having 2-bit pairs, in which both bits are in different bytes as shown in case 2 of Figure 7(a) is 12.3% (i.e. 63 pairs out of 511 total possible combinations). Which also yields probability of having patterns like case 1 of Figure 7(a) to 87.7%. We know that the 2-bit patterns shown in Figure 5(a) are equiprobable(i.e. each of them have probability of 50%). This concludes that the probabilities of having patterns like case 1 and case 2 of Figure 7(a) are 43.85% and 6.15% respectively and the probability of having patterns similar to Figure 7(b) is 50%. This implies that 56.15% of the times we can exactly pin-point the upset bit for 2-bit error area granularity. Hence, **parity per byte helps improving the 2-bit DUE rate by $\mathbf{56.15\% \times P_{2bit_{AWD}}}$.**

Figures 7(c-f) show the **3-bit patterns**. For each 3-bit pattern there are two possibilities, either these 3 bits are spread over 2 different bytes (i.e. case 1 of Figure 7(c)) or all the 3 bits are in 3 different bytes (i.e. case 2 of Figure 7(c)). Probability of having patterns similar to case 1 and case 2 of Figure 7(c) is 87.7% and 12.3% respectively. Moreover, all the 4 possibilities of 3-bit granularities, shown in Figure 5(b) are equiprobable each with the probability of 25%.

For patterns similar to case 1 of Figure 7(c) we will

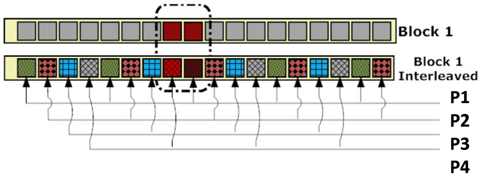


Fig. 8: Bit interleaved parity with degree of interleaving:4

be able to locate the exact upset bit if the upset is in the one bit that is in a different byte from the other two. This means that we can improve DUE for case 1 of Figure 7(c) by $(1/3)\times(87.7\%)\times 25\%$. However, We can exactly pin-point the erroneous bit in the patterns similar to case 2 of Figure 7(c) and this can improve DUE by $(12.3\%)\times 25\%$. Summing it up for all 4 possibilities shown in Figures 7(c-f) we conclude, **parity per byte helps improving the 3-bit DUE rate by $\mathbf{41.5\%\times P_{3bit_{AWD}}}$.**

For **4-bit pattern**, as can be seen in Figure 7(g) there are two possibilities. If the pattern bits are spread as shown in the case 2 over 4 different bytes in 2 rows it is possible to correct the upset. Or if they are spread as shown in the case 1 it is not possible to find the upset bit with the help of parity per byte. **Parity per byte helps improving the 4-bit DUE rate by $\mathbf{12.3\%\times P_{4bit_{AWD}}}$.**

Similar observation for **5-bit patterns** of Figures 7(h-m) reveal that for 5-bit patterns shown in Figure 7(h) we can locate the upset for case 1 only in only 2 bits out of 5 and the probability of having 3 bits in the same byte in a 64 byte block is 75.3% (i.e. 384 out of 510 total combination of triplets in a block). This results into the probability to locate the upset for case 1 as $(2/5)\times(75.3\%)$ and for case 2 as we can locate 3 bits out of 5, the probability is $(3/5)\times(24.7\%)$. Again all the 9 possibilities of 5-bit granularities as shown in Figure 5(d) are equiprobable each with the probability of 11.12%. This yields the joint probability for 5-bit patterns shown in case 1 and case 2 of Figure 7(h) as $((2/5)\times(75.3\%)+(3/5)\times(24.7\%))\times 11.12\%$. Similarly, we can correct all the upsets in all case 2 like patterns of Figures 7(i-l), but we can correct only 1 upset out of 5 possible locations in all possibilities similar to case 1 like patterns in Figures 7(i-l). This results into a probability of $(4\times(12.3\%)+(4/5)\times(87.7\%))\times 11.12\%$. Also for Figure 7(m) the probability of locating the upset is $(4/5)\times(24.7\%))\times 11.12\%$. **Parity per byte improves the 5-bit DUE rate by $\mathbf{20.5\%\times P_{5bit_{AWD}}}$.**

$$\begin{aligned}DUE_{(AWD+Parity_{byte})} =& P_{1bit_{AWD}}+56.13\%\times P_{2bit_{AWD}}+\\ & 41.5\%\times P_{3bit_{AWD}}+12.3\%\times P_{4bit_{AWD}}\\ & 20.5\%\times P_{5bit_{AWD}}\\ =&83.8\%\end{aligned} \tag{3}$$

Summing up, *Parity per byte + acoustic wave detectors* for L0 data cache will result into $\mathbf{83.8}\%$ improvement in DUE as shown in Equation 3.

C. Acoustic wave detectors and bit interleaving

Now, consider the L0 cache bits are parity protected and physically interleaved. Usually the degree of interleaving of parity protected bits of L0 data cache is in the range of 4 to 16 [4], [10]. Let's assume, every byte of an L0 data cache protected with bit interleaved parity and the degree of interleaving is 4 along with acoustic wave detectors as shown in Figure 8. This combination will make sure that all the bits in all the patterns of Figure 5 are associated with a different parity code. This implies that with interleaving degree of 4 it is possible to exactly locate the upset bit in 2-5 bit error area patterns of Figure 5.

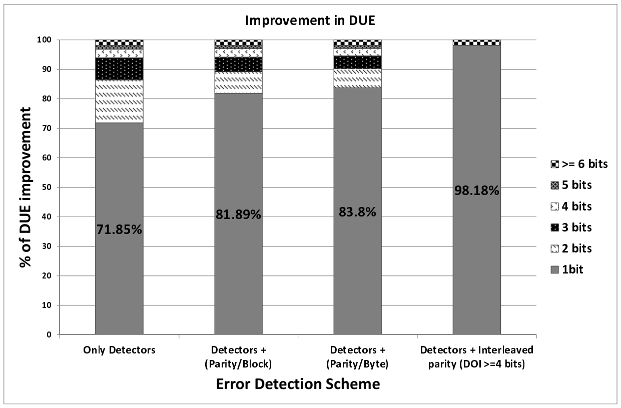


Fig. 9: Improvement in DUE for L0 data cache

Combining physical bit interleaving with $DOI = 4$ and acoustic wave detector will improve the DUE to $\mathbf{98.18}\%$.

Figure 9 sums up the improvement in the DUE achieved by using only acoustic wave detectors, and combining acoustic wave detectors with parity per block and parity per byte scheme. It also shows the improvement in DUE by combining interleaving of parity protected bits with acoustic wave detectors.

D. Cost of Protection

The area overhead includes 25 detectors (area of 25 memory bits) and a control circuit (consists of a counter and a few logic gates). Because of smaller dimensions of L0 data cache and denser mesh, the detection latency is 14.5ns for 5 × 5 mesh with 25 detectors. This means that we need to provide containment; if a read to a cache line or eviction of a dirty cache line happens during the 14.5ns detection latency, the error may propagate through the architectural state. The latency in solving 24 equations is 10ns, once the error is detected we stall the processor so this delay is harmless. The detectors are passive in nature and do not consume power and the control circuit is trivial and adds minimal power overhead. Overhead in combined approach, such as parity per block, parity per byte and bit interleaving adds to the overall cost of protecting the L0 data cache.

IV. Handling Multi-bit Upsets

In this section we will see how acoustic wave detectors can improve DUE for multi-bit upsets. We consider the multi-bit upset patterns studied in [4]. Figure 10(a) shows the 2-bit upset patterns and Figure 10(b) shows 3-bit upset patterns. As we have already seen in Section III, for the case of single bit upsets the acoustic wave detector can locate the bit at the granularity of 1 bit (best case) or 5 bits (worst case).

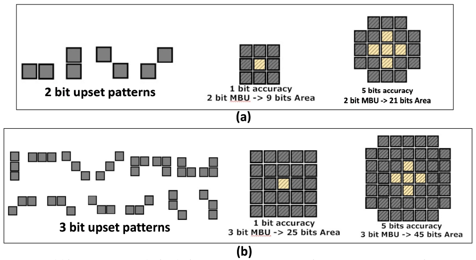


Fig. 10: Handling multi bit upsets using acoustic wave detectors (a) 2 bit MBU (b) 3 bit MBU

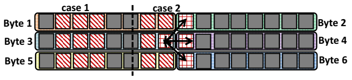


Fig. 11: 2 bit MBU for 1 bit error area granularity and parity per byte

Now in the case of 2-bit MBUs, as shown in Figure 10(a) to be able to cover all 2-bit upsets the single bit error area mask will be transformed into an area mask of 9 bits. Similarly, the 5-bit error mask will now be transformed into an area of 21 bits. The same scenario for 3-bit MBUs, as shown in Figure 10(b) will require the area masks of 25 bits and 45 bits for the error area accuracy of 1 bit and 5 bits respectively.

This implies that using only acoustic wave detectors to point out the exact locations of upsets in 2 and 3 bit MBUs is not possible. Also the combination of *acoustic wave detectors + parity per block* cannot locate the exact locations of the upset bits.

Figure 11 shows the scenario for the combination of *acoustic wave detectors + parity per byte*. Undertaking similar exercise as done in the case of single bit upsets earns, **a DUE improvement for 2-bit MBUs by of $(3/8) \times 24.7\%$ when the error area granularity of acoustic wave detector is 1 bit**. It is worth mentioning here that *acoustic wave detectors + parity per byte* cannot detect any 2-bit MBU when the error area granularity of acoustic wave detector is 5-bits. Also, this combination is ineffective against 3-bit MBUs.

Acoustic wave detectors + bit interleaving is very effective in improving DUE by locating both bits in 2-bit MBU and all 3 bits in 3-bit MBU. This can achieve **98.18%** DUE improvement for 2-bit and 3-bit MBUs. However, in adapting *Acoustic wave detectors + bit interleaving*, the minimum required degree of interleaving to be able to locate all bits in the given MBU pattern of Figure 10 increases with the increase in the number of bits required to be located. Increasing degree of interleaving increases the cost and the complexity of the solution.

Table I summarizes the minimum required degree of interleaving for adapting *acoustic wave detectors + bit interleaving*. In the L0 data cache, to be able to **correct 98.18% 2-bit and 3-bit MBUs** the optimum solution is to have acoustic wave detector with bit interleaved parity with degree of interleaving 8.

TABLE I: Minimum required degree of interleaving(DOI)

MBU type	Area gran. bits(AWD)	MBU area mask(#bits)	Min. required DOI
2 bits	1	9	4
	5	21	6
3 bits	1	25	6
	5	45	8

V. Related Work

In this section we review the most relevant works on soft error protection for data caches.

The most effective method of dealing with soft errors in memory components is to use codes for error detection and correction. Parity, SECDED and DECTED are examples of such codes [6]. SECDED and DECTED are effective but incur large area, power and delay overheads. There have been proposals for storing the stronger codes into memory and only a smaller part of the code in the cache [10]. In [5] authors show how to use parity codes for error correction. The recovery from errors in dirty data is more complex and expensive.

Bit interleaving [4] can be used to demote the spatial multi-bit fault to several single-bit faults, then simple encoding techniques can correct the several single-bit faults separately [3], [8]. Temporal multi-bit fault is the cumulative effect of several single-bit faults in a period of time. For temporal multi-bit errors, cache scrubbing [7] techniques will be more effective.

VI. conclusion

This paper presents an architecture that uses acoustic wave detectors to detect and precisely locate the particle strikes with minimal hardware overhead incurring zero performance cost. We have shown how combining acoustic wave detectors with parity codes and interleaving can significantly improve the DUE for data cache, for both single and multi-bit upsets. We conclude that by using only acoustic wave detectors we can improve the DUE in the case of single bit upsets by 71.85%. For multi-bit upsets by combining *acoustic wave detectors with bit interleaving* we can improve the DUE for 2 and 3 bit MBUs by 98.18%.

References

[1] I. Corporation, *Intel's Nehalem data sheet*, Intel Corporation.

[2] S. R. et. al., "A 65-nm dual-core multithreaded Xeon®processor with 16-MB L3 cache," *IEEE Journal of Solid-State Circuits*, vol. 42, no. 1, pp. 17–25, 2007.

[3] M. K. Kim J, Hardavellas N, "Multi-bit error tolerant caches using two-dimensional error coding," in *Proceedings of International Symposium on Microarchitecture (MICRO)*. Washington, DC, USA: IEEE Computer Society, 2007.

[4] Z. K. Maiz J, Hareland S, "Characterization of multi-bit soft error events in advanced srams," in *IEEE International Electron Devices Meeting, 2003. IEDM'03 Technical Digest*. Los Alamitos, CA, USA: IEEE Computer Society, March 2003, pp. 21–24.

[5] M. Manoochehri, M. Annavaram, and M. Dubois, "Cppc: correctable parity protected cache," in *Proceedings of the 38th annual international symposium on Computer architecture(ISCA)*, 2011.

[6] S. Mukherjee, *Architecture Design for Soft Errors.*, 1st ed., 2009.

[7] S. Mukherjee, J. Emer, T. Fossum, and S. Reinhardt, "Cache scrubbing in microprocessor," in *Proceedings of International Symposium on Pacific Rim Dependable Computing (PRDC)*, 2004.

[8] C. P., "Two-dimensional parity checking," in *Proceedings of International Symposium on Microarchitecture (MICRO)*. Washington, DC, USA: IEEE Computer Society, 1961.

[9] G. Upasani, X. Vera, and A. González, "Setting an error detection infrastructure with low cost acoustic wave detectors," in *Proceedings of the 39th International Symposium on Computer Architecture (ISCA)*, 2012.

[10] D. H. Yoon and M. Erez, "Memory mapped ecc: low-cost error protection for last level caches," in *Proceedings of the 36th annual international symposium on Computer architecture(ISCA)*, 2009.