

Setting an Error Detection Infrastructure with Low Cost Acoustic Wave Detectors

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

The continuing decrease in dimensions and operating voltage of transistors has increased their sensitivity against radiation phenomena making soft errors an important challenge in future chip multiprocessors (CMPs). Hence, new techniques for detecting errors in the logic and memories that allow meeting the desired failures-in-time (FIT) budget in CMPs are required.

This paper proposes a low-cost dynamic particle strike detection mechanism through acoustic wave detectors. Our results show that our mechanism can protect both the logic and the memory arrays. As a case study, we also show how this technique can be combined with error codes to protect the last-level cache at low cost.

1. Introduction

The reliability, availability and serviceability (RAS) of systems to perform to customer expectations are strongly related to how the system is designed to respond to both hard and soft failures. 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. Therefore, a single radiation such as neutrons coming from outer space can cause a transient error [4]. Since these transient errors occur due to an incorrect charge or discharge of an intermediate capacitive node, they do not cause permanent failure in the hardware and hence are termed *soft errors* (SER) in the literature.

Constantly shrinking dimensions and operating voltages of transistors has increased their sensitivity against radiation phenomena making SER an important challenge in chip multiprocessors (CMPs). Moreover, the total failures-in-time (FIT) per chip will increase due to larger arrays and increased number of cores per area [18]. Hence meeting the

desired FIT budget for current and future CMPs is a major challenge.

Techniques for protecting memories against faults include the use of parity and error correcting codes. However, the capacity of caches like last-level cache (LLC) is growing. This leads to a rise in the probability of having multiple particle strikes since cache lines usually spend more time sitting in the cache before they get accessed again. As a consequence, designers are forced to use stronger codes, which implies higher costs in terms of area, power and latency. Another challenge to reduce the current FIT rate is protecting the logic. Since memories are already protected, unprotected logic elements are the current main contributors to the majority of the FIT budget. Therefore, designers have an acute need to protect them.

This paper proposes a low-cost dynamic particle strike detection mechanism through acoustic wave detectors. Instead of relying on codes or some kind of redundancy, we deploy a set of detectors on silicon that allows locating the exact position of particle strikes. The potential of this solution is twofold: (i) it can detect errors on all the current unprotected logic at a very low cost, and (ii) it can decrease the growing costs of protecting large memory arrays. Moreover, the proposed mechanism can stand alone or it can be integrated smoothly with other end-to-end error detection techniques.

In summary, the principal contributions of this paper are:

- We develop an architecture that detects and locates particle strikes on a processor based on acoustic wave detectors. We first introduce the structure of such detectors, and later propose the architecture to deploy them.
- We propose a new methodology that uses the acoustic wave detectors to precisely locate particle strikes. Our solution is based on measuring the time difference of arrival across different detectors, generate a set of hyperbolic equations, and solve them. We discuss the different trade-offs in terms of cost versus precision.

- We thoroughly evaluate our proposed architecture undertaking a case study of the LLC of CoreTMi7-like processor. Additionally, we propose a new solution that combines acoustic wave detectors with error correcting codes in such a way that we decrease the total cost of the protection mechanism while giving the same reliability levels.

The rest of the paper is structured as follows: Section 2 reviews some relevant related work. Section 3 explains how we can build acoustic wave detectors that detect particle strikes. Section 4 details our architecture that uses such detectors to locate particle strikes. We evaluate our proposed architecture in terms of coverage and overheads in Section 5. Section 6 presents a new protection mechanism for a LLC that combines our architecture with error detection and correction codes. Finally, a summary of main conclusions is presented in Section 7.

2. Related Work

In this section we review the basic works on soft error protection for memory arrays and logic.

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 [18]. Bit interleaving [16] 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 [13, 14, 21]. 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 [19, 26] techniques will be more effective.

Execution redundancy is a widely used technique to detect errors in the logic, either using the multithreading capabilities [20, 25] or the inherent hardware redundancy in CMPs [29]. Reis et al. proposed using hardware-software hybrid schemes which achieve fault tolerance by replicating instructions at the compiler level and using hardware fault detectors that make use of this redundancy [23, 24]. Replicating parts of the core has also been explored. DIVA [1] uses a simple in-order core as a checker for an out-of-order core.

3. Background: Radiation Interaction with Silicon Surface

Ultra-high energy particles from intergalactic sources interact with atmospheric nuclei and create a number of cascades of many nucleons such as neutrons, protons, pions, muons, etc. These particles strike silicon devices randomly in time and location. When the particles hit the silicon devices they generate electron hole pairs resulting into gen-

eration of charge. Neutrons are the dominating among all of the secondary particles and can corrupt a data bit stored in the memory (i.e., SRAM) or create a glitch in any gate in combinational logic. Since these errors have a non-permanent nature, they are termed soft errors.

In this section we introduce acoustic wave detectors as a method to detect such particle strikes. We first explain how particle strikes generate sound on the silicon surface. We then explain how we can detect that sound, and then how to build acoustic wave detectors in silicon with cantilevers.

3.1. Generation of Sound Waves

The primary interaction by which cosmic particles induce soft errors is the induction of silicon recoil [3]. When a high-energy particle collides with a silicon nucleus, it can transform enough energy to knock the nucleus from the lattice.

Recent studies [2, 3, 8, 12] show that particles with recoil energies of 10MeV or higher are capable of causing upsets in the circuits. When a cosmic ray collides with a silicon nucleus this energy is released in a very short span of time ($\leq 1ns$). This rapid recombination process results into a cloud of phonons spreading out of the impact site. Hence the cosmic ray is transformed into an intense sound wave as shown in the Figure 1(a). Such an acoustic wave travels at the speed of 10km/s on the silicon surface [7].

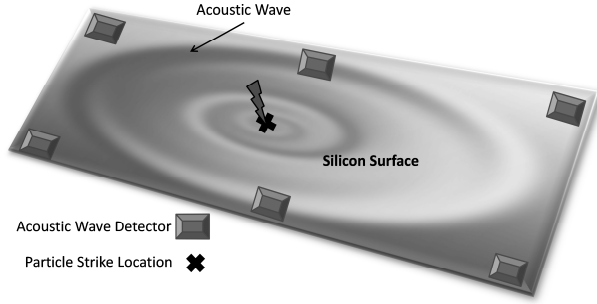
3.2. The micromechanical ears: Acoustic Wave Detectors

We propose to use cantilever like structures [10, 11] as an acoustic wave detector to detect particle strikes through the sound they generate. To be able to detect the impact of the cosmic particle, the cantilevers must perform two contradictory tasks:

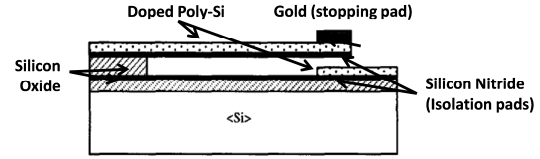
1. They must absorb as much energy as possible resulting due to the collision. This implies a thick pliable structure composed of a high density, high- Z^1 material, such as gold.
2. For efficient detection at a distance and to avoid thermal noise the pliable structure must maximally deflect for the given energy deposition. Thus, the levers should be light in weight and highly flexible.

Figure 1(b) shows the typical structure of an acoustic wave detector. These devices are rectangular structures of beams and plates on the silicon surface. A doped polysilicon grounding layer forms the lower plate of the sensing capacitor. Silicon oxide serves as the isolating layer between lever and substrate. The fabrication and placement of these

¹High impedance.



(a) Acoustic wave generated by particle strike



(b) Cantilever sensing device [10]

Figure 1. Transformation of particle strike into acoustic wave and structure to detect it

detectors on the surface of active silicon can be performed without much complications [11].

The particle strike is detected by the change in the capacitance of the gap between the cantilever and the ground pad of the detector structure shown in Figure 1(b). A simple capacitance detector can be designed based on a relaxation oscillator [5]. A simple microcontroller can be used for the same purpose. More accurate and faster capacitive detectors circuits can be constructed that are able to detect changes in capacitance on the order of 10 attofarads [30].

The length of the cantilever beam is very important in detecting the cosmic particle strike. Too long or very small lever dimensions would not be efficient in detecting the desired particle strikes. For instance, at 45nm technology, any particle strike that will result into a silicon recoil energy lesser than 10MeV will not induct enough charge to create an upset in the memory [2, 3]. Therefore, we need to size the cantilever accordingly, in such a way that it only detects particle strikes that result into a silicon recoil energy larger than 10MeV and therefore avoiding false positive detection.

The proposed cantilevers occupy an area of one square micron [12], which is roughly the area of one bit (a typical 6T SRAM cell) at 45nm. The cantilever is designed such that it detects particle strikes that generate silicon recoil with more than 10MeV energy. The cantilever can detect up to 0.3 mW/cm^2 , which is the sound wave peak power at a distance of 5mm from the source of the sound [12]. This means that our selected cantilever can cover an area of 78.5 square millimeters. This area is equivalent to the die area occupied by the last-level cache in a CoreTMi7 microarchitecture at 45nm technology [6].

In this work, the fundamental idea is to detect the particle strikes via mechanical deflection of acoustic wave detectors. The potential of the detectors will be exploited by: (i) detecting errors in the unprotected logic and therefore, reduce the silent data corruption (SDC) FIT rate, and (ii) deploying less number of detectors than the required parity/ECC bits and accurately localizing the particle strikes/bit flips in memory arrays.

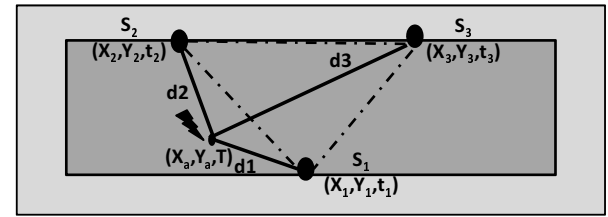


Figure 2. Strike detection

4. Detection and Localization of Particle Strikes

The previous section discussed the use of cantilevers for detecting the *existence* of particle strikes on the silicon surface. However, accurately *locating* the particle strike is somewhat more involved. In this section, we discuss how to use the acoustic wave detectors in order to precisely locate the particle strike. We will answer the following questions: (i) how many acoustic wave detectors are required to be able to locate the particle strike?, (ii) where should the acoustic wave detectors be placed?, (iii) what would the accuracy of the found location be?, and (iv) what would the latency in detecting the particle strike be?

4.1. Overview: Estimating the Location of the Particle Strike

Let us assume that one particle strikes at location (X_a, Y_a) . Therefore, a system of two equations is required to solve both unknowns. Unlike GPS, any apriori knowledge of the spatio-temporal information about the impacting particle strike is unavailable. This means that we do not know the actual time span between the particle strikes and the detectors trigger. The only information we have is the relative time difference of arrival (TDOA [28]) of the acoustic wave generated by the strike between the detectors. Hence, minimum three detectors are needed: with three de-

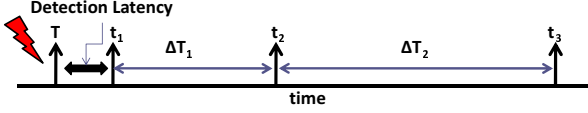


Figure 3. Timeline of the events following the particle strike

tectors we obtain two TDOA measurements, which opens the door to write the two required equations.

Hyperbolic position estimation. The estimation of the location is a three stage process. The first step is placing the acoustic wave detectors. They can be placed on or off the chip but on the same silicon surface. Notice that the coordinates of the acoustic wave detectors are known.

In the second stage we measure the TDOAs of the sound 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. Once the equations are formed, efficient algorithms are applied to produce a solution to these nonlinear equations [9]. At the end of the process the solution results in the estimated position of the particle strike.

4.2. Example

To better illustrate the particle strike detection and localization problem, a simple case of particle strike localization using 3 acoustic wave detectors is discussed.

Figure 2 displays three acoustic wave detectors (S_1 , S_2 and S_3) placed at known coordinates (X_1, Y_1) , (X_2, Y_2) and (X_3, Y_3) respectively on the surface of the cache. Now, let's assume that a particle strike occurs at an unknown time T at unknown location (X_a, Y_a) . As shown in Figure 2, d_1 , d_2 and d_3 are unknown absolute distances from the detectors S_1 , S_2 and S_3 . Once the strike has occurred, the ripples of phonons will traverse outward in a circular manner and the closest detector from the strike will trigger first. In this case S_1 will trigger at instance t_1 . After that, as the phonons traverse further, other detectors S_2 and S_3 will trigger at instances t_2 and t_3 respectively. A timeline of the events is shown in Figure 3.

4.3. Obtaining TDOA

Figure 4 shows a simple system which can measure the timing differences of the acoustic waves' arrival. The hardware consists of an asynchronous control (e.g., a multiple input OR gate) which generates a single signal out *Enable*. *Enable* is high whenever one of the triggered detector raises a flag, and activates the sequential counter that counts the

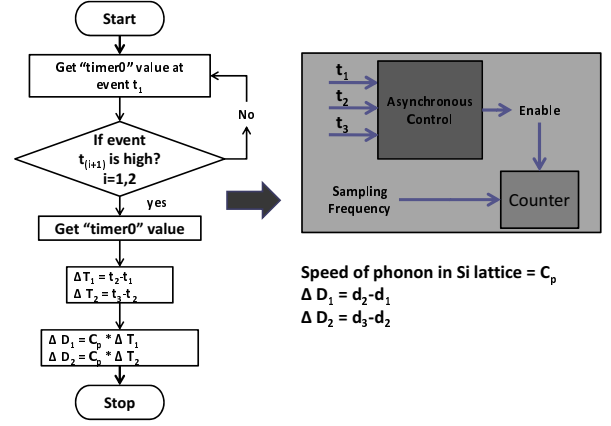


Figure 4. Strike detection hardware/firmware

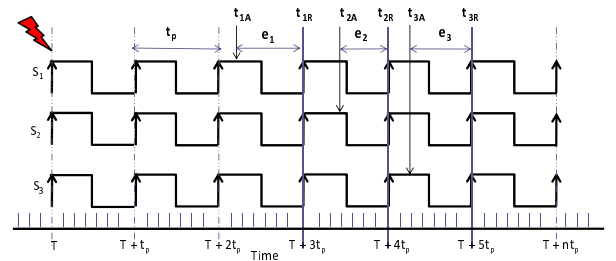


Figure 5. Sampling errors in the measurements of the time difference of the arrival at the acoustic wave detectors

number of clock pulses between two consecutive triggering detectors. The counter runs at the *sampling frequency*, which is a parameter of the design.

As the speed C_p at which acoustic waves traverse on the silicon surface is known (recall Section 3.1), using the measured timing differences of the arrival of the acoustic waves it is easier to compute the distance differences ΔD_i .

Errors in measurements. The effect of errors in the measurements of timing differences due to the sampling frequency cannot be ignored. We use the example depicted in Figure 5 to illustrate such case: the three detectors S_1 , S_2 and S_3 are in synch with each other and are being sampled at the rising edge of the clock with sampling period t_p . The actual arrival times of the acoustic wave generated due to particle strike at detectors S_1 , S_2 and S_3 are t_{1A} , t_{2A} and t_{3A} respectively. However, the signal will be read only at the rising edge of the clock pulse (i.e., at the instances t_{1R} , t_{2R} and t_{3R}) by the detectors. This introduces error in the measurements of the time differences.

Error can be modeled according to the next equation; let's assume a particle strike occurring at an unknown instance T and sampling period t_p . The sampling error e_s at

the acoustic wave detector S can be given by:

$$e_s = t_p - [(T + t_{iA}) \bmod (t_p)] \quad (1)$$

Notice that $e_s \in [0, t_p)$. Hence, the error in the time difference of arrival of the acoustic wave between detectors S_i and S_{i+1} is $e_{s_i} \in (-t_p, t_p)$.

4.4. Generating TDOA Equations

Once we know how to obtain the TDOA data, next step is generating the equations that describe the localization of the particle strike. We sort detectors based on their proximity to the source of the signal (i.e., the order in which they trigger), S_1 being the closest detector and S_n the furthest one. (X_a, Y_a) denotes the unknown source location and (X_i, Y_i) indicates the known location of the i^{th} detector.

A general model for the two dimensional (2-D) location estimation of a source using N detectors is adapted from [9], where the mathematical problem is to estimate (X_a, Y_a) given the detector positions and the TDOA readings. First, we define the squared euclidian distance between the source and the i^{th} detector:

$$D_{ia} = \sqrt{(X_i - X_a)^2 + (Y_i - Y_a)^2} \quad (2)$$

Next we derive the range difference ΔD_{ia} between detectors S_i and S_{i+1}

$$\begin{aligned} \Delta D_{ia} &= D_{ia} - D_{(i+1)a} \\ &= \sqrt{(X_i - X_a)^2 + (Y_i - Y_a)^2} \\ &\quad - \sqrt{(X_{i+1} - X_a)^2 + (Y_{i+1} - Y_a)^2} \end{aligned} \quad (3)$$

Now, we can set up our set of equations based on the TDOA measurements ΔT_{ia} between detectors S_i and S_{i+1}

$$\Delta D_{ia} = C_p * \Delta T_{ia} + e_{s_i}, \quad i = 1 \dots N - 1 \quad (4)$$

where C_p is the speed of the sound-wave on the silicon surface. Notice that if N is larger than 3, we will have an overdetermined system (i.e., more equations than unknowns).

4.5. Solving TDOA Equations

In this section we will explain how we solve the set of equations and estimate the location of the particle strike. A high-level algorithm is shown in Algorithm 1.

Lines 1-5 show the required inputs for solving the equations. The number and location of detectors, as well as the statistical distribution of the error measurements is information known at design time. The TDOA measurements are calculated online as explained in Section 4.3.

Algorithm 1 System level algorithm of hyperbolic location estimation

- 1: **INPUT:** Number of total detectors $\mapsto N$.
 - 2: **INPUT:** Locations of the detectors $\mapsto (X_i, Y_i)$, where $i = 1; 2; \dots; N$.
 - 3: **INPUT:** Range difference between receivers $\mapsto \Delta D_{ia}$, where $i = 1 \dots N - 1$.
 - 4: **INPUT:** Error in TDOA $e_{s_i} \in (-t_p, t_p)$.
 - 5: **INPUT:** Error covariance matrix $\mapsto R = [e_{s_i}]$.
 - 6: Identify triggered detectors.
 - 7: Generate hyperbolic equations.
 - 8: Linearization $\mapsto A\delta \cong Z + E$
 - 9: Gauss-Newton-Interpolation $[(X_v, Y_v), N, (X_i, Y_i), A, \delta, Z]$
 - 10: **while** $(\delta_x \neq 0, \delta_y \neq 0)$ **do**
 - 11: $[\delta_x, \delta_y] \mapsto LSQR((A), (Z))$
 - 12: $X_v \leftarrow X_v + \delta_x, Y_v \leftarrow Y_v + \delta_y$
 - 13: **end while**
 - 14: Compute $Q = [A^T R^{-1} A]^{-1}, CEP$
 - 15: **OUTPUT:** Area of Error Distribution
 - 16: **OUTPUT:** Radius of the circle(CEP), center (X_v, Y_v)
-

First step of the algorithm is generating the equations (lines 6-8). Equation 3 (and therefore, the set of equations 4) is nonlinear in nature. We opt to linearize these equations through Taylor-series expansion and retain the terms below second order [9].

The system of equations is solved by the iterative LSQR algorithm [22] (lines 9-13). In order to estimate the solution, we keep iterating until $\delta_x \mapsto 0$ and $\delta_y \mapsto 0$. Each new iteration is updated through $X_v \leftarrow X_v + \delta_x$ and $Y_v \leftarrow Y_v + \delta_y$.

Estimation of the error. Last step is calculating the error in the obtained position estimate (line 14). We use circular error probability (CEP) to express the area of the error distribution of the final estimation of the position [17]. Using the Rayleigh's method for approximating the CEP [17], it is possible to guarantee that actual strike location will always fall within a circle with the center at the obtained estimated location and the radius equal to the $3 * CEP$.

Mapping spatial multiple bit upsets. Spatial multi-bit errors occur when a particle strike affects adjacent bits [27]. Our scheme takes them into account in a very easy manner. We assume that a set of templates for the shape of the upsets caused by a particle strike are available. Then, we only need to map on top of the perimeter of the $3*CEP$ circle the templates, and therefore, extend the area of affected bits.

4.6. Runtime Calculation

We propose to generate and solve the equations in software. Once the first detector triggers, we stall the processor and obtain all TDOAs. Once all TDOAs are ready, we execute the algorithm to generate and solve the equations. This code is stored in firmware (along the position of all

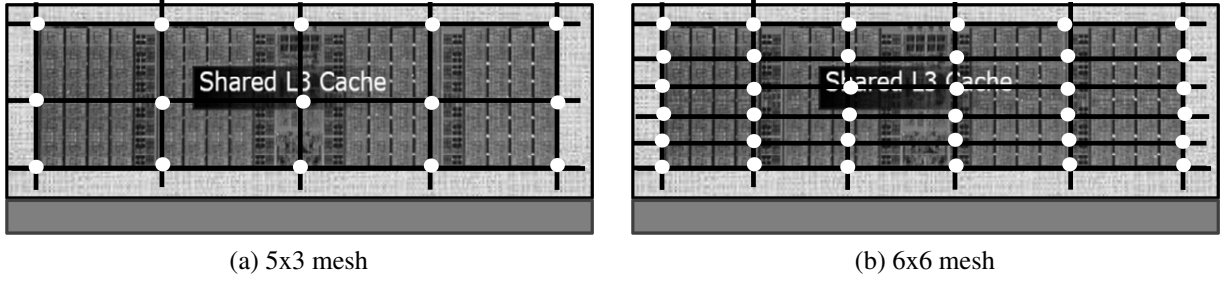


Figure 6. Placement of detectors in mesh formation

detectors) and is transparently run in any of the cores of the processor. The preferred option is to run it in one core that is not triggering the error to facilitate the error recovery if necessary, but it could also be run in the same core with some checkpointing. The impact on the performance of active tasks and user experience would be minimal, since generating and solving the equations takes around 0.1ms in a CoreTMi7 processor.

5. Locating Errors in the Last-Level Cache

In this section we demonstrate the utility of the cantilever detectors by detecting and locating particle strikes in the LLC of a CoreTMi7-like processor. The cache is 8MB, 16-way and has rectangular shape with the surface area of $78mm^2$. Monte-Carlo experiments consisting of 1048 randomly distributed particle strike locations in space and time have been performed.

Next, we will discuss the placement of the detectors, and how the number of detectors and the sampling frequency impact the error in the estimation.

5.1. Detectors Location

After trying different configurations, we have opted to place the detectors in a mesh.

Figure 6 shows the placement of acoustic wave detectors in mesh formations on LLC. Two formations, 5x3 and 6x6, are shown. Each node in the mesh represents an acoustic wave detector. For all the meshes $m \times n$ the area of cache is split into $m - 1$ equal parts along the X-axis and $n - 1$ equal parts along Y-axis.

We have evaluated different mesh configurations. For that experiment, we have opted for the most basic over-determined system of 3 equations. Therefore, we need to construct a mesh that guarantees that for all possible particle strikes, at least 4 detectors trigger (recall that only the detectors that are placed within 5mm of the particle strike will be able to detect the strike). Our studies show that the minimum configuration is a mesh with 15 detectors, 5x3. In

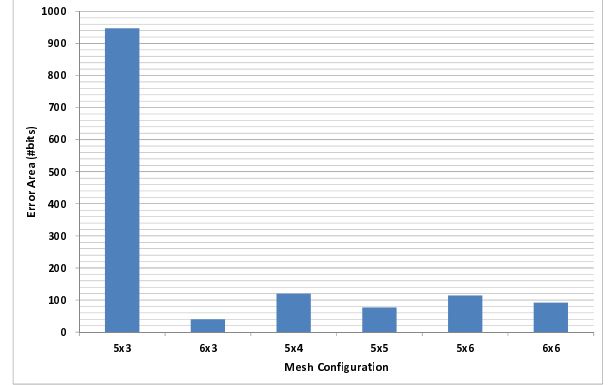


Figure 7. Worst-case error area for different mesh configurations when building 3 equations

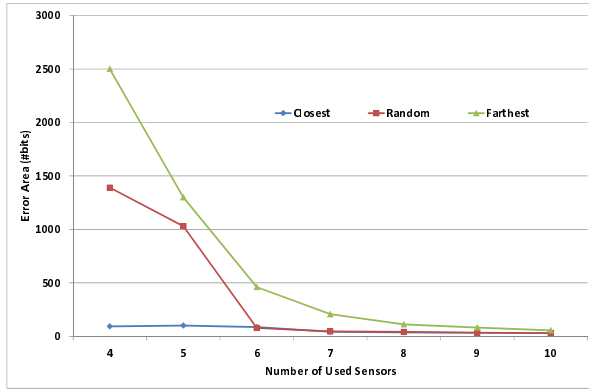
those configurations where more than 4 detectors trigger, we take the first four detectors.

Figure 7 shows how the number (and placement) of detectors impact the error area (in terms of $3 \times \text{CEP}$). As one can see, using only 15 detectors yields a large error area of 947 bits, which is a $3 \times \text{CEP}$ radius of 17 bits. However, when we change to a 6x3 mesh, area is extremely reduced to a radius of 3 bits. It is also interesting to note how increasing the number of detectors does not increase the quality of the solution, since solution is more affected by the location of the detectors. For instance, using 36 detectors through a 6x6 mesh yields a $3 \times \text{CEP}$ radius of 5.4.

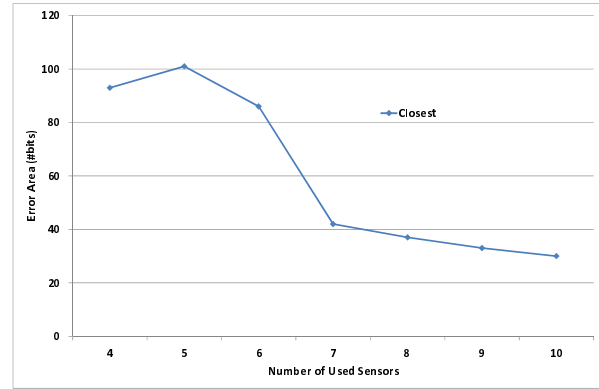
5.2. Effect of Number of Equations on Accuracy

In this section, we assess the impact of the number of equations on the $3 \times \text{CEP}$ error area. For that purpose, we choose a 6x6 mesh because it guarantees that at least, 10 detectors detect the particle strike. We also assume a 2GHz sampling frequency.

Figure 8(a) shows the obtained error area for the 6x6 mesh. We show results for three different algorithms that se-



(a) Different algorithms



(b) Closest detectors algorithm

Figure 8. Worst-case error area with the selection of different set of detectors (4 to 10) from a given 6x6 mesh

lect the detectors when more detectors than necessary trigger: (i) choosing the closest, (ii) the farthest and (iii) choosing randomly. Result shows that for all the choices, selecting the closest detectors is the most accurate option. This is because the nearest detectors are placed at locations where it was possible to generate better TDOA measurements between two detectors and the LSQR method could reach to a more accurate solution.

Once we select the closest algorithm, we can observe that increasing the number of equations has a very important impact on the error area. It can be seen that for the closest selection of detectors for the given 6x6 mesh ($M=36$) increasing the used detectors from 4 to 10, the error area reduces by a factor of 3 (see Figure 8(b)).

We show in Table 1 the best configurations observed; we consider different mesh configurations and number of equations for each of them. Third column of the table shows the minimum number of detectors that trigger upon the particle strike. Fourth column shows the number of detectors used to set up the equations. Last column shows the worst-case error observed for the 1048 particle strikes. Although the best error area is obtained by setting a 6x6 mesh and using 10 detectors, the complexity of setting and solving the equations makes it too expensive. Therefore, we conclude that the best trade-off is obtained by setting a 5x3 mesh using 5 acoustic wave detectors and setting up 4 different equations.

5.3. Effect of Sampling Frequency on Accuracy

The effect of altering the sampling frequency over the final error area is also studied thoroughly. Figure 9(a) shows the impact of sampling frequency on the worst-case error area for all *best* configurations described in Table 1. The results indicate that doubling the frequency from 2GHz up

to 4GHz reduces the error area by 3.5x.

We detail the best configuration described in previous section (a 5x3 mesh employing 5 detectors) in Figure 9(b). We can see that increasing the sampling frequency reduces the error area; doubling the frequency from 2GHz up to 4GHz reduces the worst-case error area from 38 bits down to 11 bits (i.e., a radius of 3.4 bits down to 1.8 bits).

5.4. Error Area Granularity

Due to the errors in the TDOA measurements caused by the sampling frequency, the location of the particle strike is given as estimated (X, Y) coordinates and an estimation of the error area ($3 \cdot \text{CEP}$) that contains the actual location of particle strike. So far we have discussed the errors in terms of bits. However, these errors can be easily mapped to bytes or a cache lines. For instance, the worst-case error of 38 bits that we estimated for a 5x3 mesh employing 5 detectors at 2GHz sampling frequency would map in the worst-case on 8 different cache lines, assuming that there is no space between them (38 bits is a $3 \cdot \text{CEP}$ radius of 3.4 bits). It is also interesting to note that we can detect whether the particle strike occurred on the cache or it struck on the logic elsewhere on the die area.

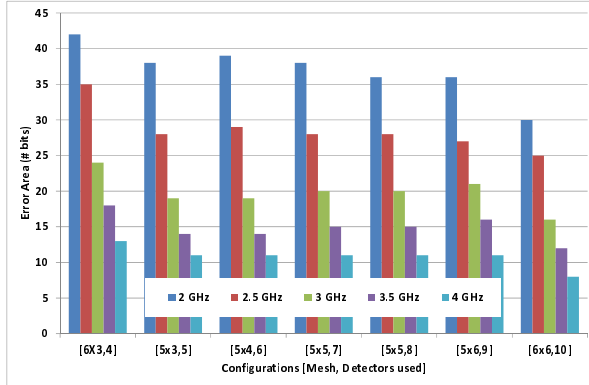
5.5. Detection Latency

Detection latency can be defined as the time until the first detector triggers following a particle strike. Smaller detection latency will make it easier to contain the errors, since the sooner we detect the error, the sooner we can take the right actions.

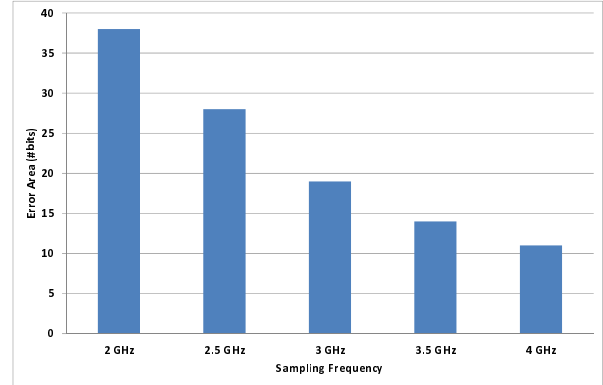
As it is discussed in section 3.1, the sound wave traverses the silicon lattice at 10km/sec. This means that if only one acoustic wave detector was used, in the worst-case a particle

Mesh Configuration	#Detectors in Mesh	Minimum #Detectors for each strike	#Detectors used for algorithm	Worst Error (#bits)
5×3	15	5	5	38
6×3	18	4	4	42
5×4	20	6	6	39
5×5	25	8	7	38
5×5	25	8	8	36
5×6	30	9	9	36
6×6	36	10	10	30

Table 1. Configuration of mesh and number of detectors used in solving equations (best choices)



(a) Impact of sampling frequency on error



(b) Impact for best configuration

Figure 9. Impact of sampling frequency on error area

strike occurring at 5mm away would be detected in 500ns (or 1000 cycles in a processor running at 2GHz).

Figure 10(a) shows the worst-case latency observed for the different mesh configurations. As one can observe, adding more acoustic wave detectors significantly helps in reducing the detection latency. Therefore, we have considered the option of adding, on top of the detectors deployed for precise estimation of location, a set of detectors to minimize the detection latency.

We show the results in Figure 10(b), where the 5×3 mesh is considered for estimating the location. We observe that the number of detectors required to reduce the worst-case detection latency increases exponentially. The sweet point is adding an extra 23×7 mesh (161 extra detectors), which allows reducing the detection latency down to 100 cycles.

5.6. Summary and Overheads

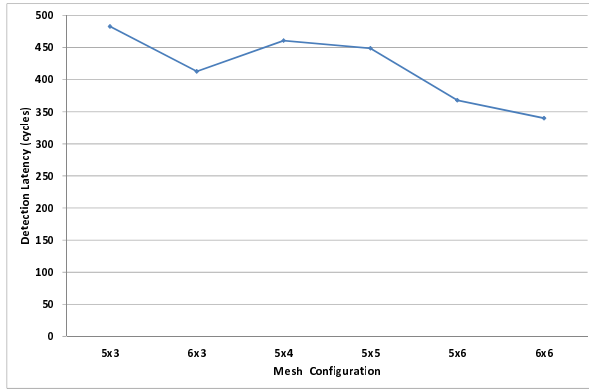
The results confirm that the overdetermined system of equations (i.e., when using more than 3 detectors and setting more than 2 equations) reduces the worst-case error area by a huge margin. We have also shown the impact of the sampling frequency on the behavior of the error area. Increasing sampling frequency reduces the sampling error in the measured TDOA. Raising sampling frequency from 2 GHz to 4 GHz, reduces the sampling errors by a factor of 2; this

reduction reflects into a worst-case error area reduction of 3.4x.

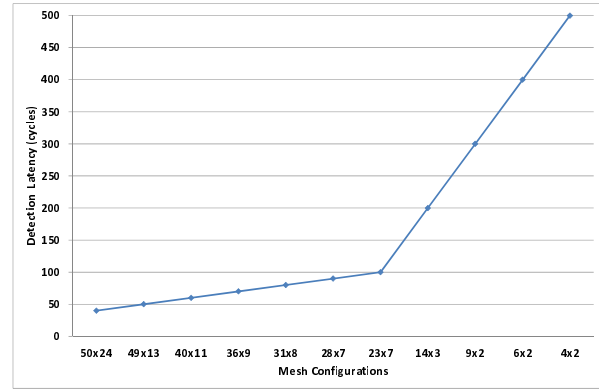
Overall, our results confirm that increasing the sampling frequency is more effective than increasing the number of equations. For instance, a system that uses 3 equations (e.g., 6×3 mesh) sampling at 4 GHz is a better option than a system using 9 equations (e.g., 6×6 mesh) with the sampling frequency of 2 GHz.

Finally, we have also discussed the impact of the number of detectors on the detection latency. We have concluded that the most effective design is the one that uses two independent meshes: a small mesh for precise location of the strike, and a somewhat larger mesh for detection latency. The optimum configuration for the LLC, is to set a 5×3 mesh and an overdetermined system of equations of 4 equations, which gives a worst-case error area of 38 bits. We also add a 23×7 mesh for detection latency, resulting in a latency of 100 cycles for a processor running at 2GHz.

Overheads. The proposed solution will make use of two different meshes. The 5×3 mesh will be used to obtain the TDOA. In that case, the hardware mechanism explained in Section 4.3 will consist of 15 detectors (i.e., roughly 15 bits area), and a 2-level OR tree to generate the *Enable* signal. The tree will use 6 3-input OR gates and 2 2-input OR gates. The worst-case TDOA is 765 clock pulses. Hence, a 10-bit counter is necessary.



(a) Regular configuration



(b) Adding detectors for latency

Figure 10. Worst-case detection latency for a processor running at 2GHz

We will also use a 23x7 mesh to minimize the detection latency. On one hand, it requires 161 detectors (i.e., roughly 161 bits area). On the other hand, we will need a 4-level OR tree to generate the detection signal. Such tree is composed of 66 3-input OR gates and 28 2-input OR gates. Notice that in this case we do not require a counter since we only want to signal the presence of the strike.

6. Case Study: Reducing Error Detection Codes Complexity

In this section, we describe how the implementation proposed in Section 5 would interact with the normal operation of a processor and which are the most important challenges for achieving high levels of error protection and error containment. Later, we combine our mechanism with error detection and correction codes and compare it with a mechanism that solely uses error detection and correction codes.

6.1. Cache Protection with Acoustic Wave Detectors

In order to detect particle strikes in the cache, we use the configuration outlined in Section 5.6: a 5x3 mesh, an overdetermined system of equation of 4 equations, and a 23x7 mesh for reducing the detection latency. Assuming a 2GHz *sampling frequency* the worst-case error area is 38 bits (it spans 8 cache lines). The worst-case detection latency is 100 cycles. Once the first detector triggers, the processor stalls and our system calculates the location of the particle strike.

Notice that particle strike rate with recoil energy $\geq 10MeV$ is not very high [2, 3, 8, 12]. Hence, the probability of consecutive particle strikes in a time span of 100 cycles is practically zero.

Reaction upon a particle strike. Once we know the estimate of the localization of the particle strike and the error area, it is time to take the appropriate actions to provide, when possible, fine-grain error detection, error correction and error containment. The challenges are:

1. We need to provide recovery capabilities; if the particle strike has occurred on a dirty line the detectors cannot recover them since the location accuracy is not at a bit level.
2. We need to provide containment; if a read to a cache line or eviction of a dirty cache line happens during the worst-case 100 cycles detection latency, the error may propagate through the architectural state.

Next, we will consider the case when a cache is protected only with acoustic wave detectors, and the more reasonable case when they are deployed with protection codes.

6.2. Standalone Acoustic Wave Detectors

Once the particle strike has been localized, the error area would be in the worst-case 8 cache lines. This means that we would have 8 potential cache lines where the particle could have hit. We propose to go line by line within the error area provided by the localization algorithm and *clear* them: since we assume that protection code is not used, we invalidate the cache lines *just in case*. If any of the lines is dirty, no recovery would be possible and we would need to throw a machine check architecture (MCA) exception.

Techniques such as early write back [15] may help in providing recovery by minimizing the number of dirty cache lines. However, the impact on performance is out of the scope of this paper and we leave this evaluation for future work.

Error containment is somewhat more involved. In the worst-case, detectors would trigger 100 cycles after the par-

ticle has hit the cache. This means that any data (assuming the cache does not have error codes) leaving the cache may have a bit flip. For cache lines being evicted, this can be easily solved using a victim buffer that delays write to main memory for 100 cycles. On the other hand, data being served to the processor would reach the head of the reorder buffer much earlier than those 100 cycles. A good option to contain the error would be stalling the commit of the load instruction (with its corresponding impact on performance) or enabling checkpoint mechanisms. Again, this study is out of the scope of this paper and we leave it for future analysis.

Next, we will explore combining acoustic wave detectors with regular codes. This combination of protection mechanisms will improve error recovery, error containment and protection against hard faults.

6.3. Acoustic Wave Detectors with Error Codes

The baseline implementation is the same as explained in the previous section: once the error is localized, we would go line by line within the error area provided by the localization algorithm and *clear* them. Unlike the previous case, now we have the option of using the error code present in the cache line to clearly identify which is the line affected. If the code offers the correction, we would correct the cache line (the benefits would be similar to those of cache scrubbing [19, 26]). If code only offers detection, we would still need to invalidate the affected cache line.

Combining detectors with error codes offers two other benefits: (i) error codes allow us to contain the error when the cache line is evicted or read before the detectors trigger, and (ii) they allow us to identify if an error is caused by a hard fault or particle strike. If a cache line is read or evicted and the code triggers, we will wait up to 100 cycles. If the error is caused by a particle strike, a detector will trigger. Otherwise, it is a hard fault. In either case, correction will be provided by the code when possible.

As one can see, using *Error Codes+Detectors* we can detect all particle strikes, since detectors trigger timely and therefore, latent particle strikes do not accumulate. In general, error containment is achieved when the number of hard faults in the cache line is strictly less than the error code detection capability (1 for double error detection, 2 for triple error detection). Error correction (of dirty lines) is achieved when the number of hard faults in the cache line is strictly less than the error code correction capability (0 for single error correction, 1 for double error correction).

The approach of *Error Codes+Detectors* is able to detect *all* temporal particle strikes that cause bit upsets (i.e., with recoil energy $\geq 10\text{MeV}$), whereas in the case of only *Error Codes* the detection is limited by their detection ca-

pability. Moreover, *Error Codes+Detectors* provides better error containment.

Interestingly, in a scenario where there is presence of 1 hard fault, SECCDED codes with detectors provide the same detection level as DECCDED, at a much cheaper cost in area and latency.

7. Conclusions

This paper presents a novel architecture that provides particle strike detection with minimal hardware overhead at no performance cost. We have introduced how to implement acoustic wave detectors with cantilevers that can easily be implemented in silicon. Then, we have proposed a system based on TDOA measurements and hyperbolic equations to precisely locate the particle strike. Finally, we have discussed the impact of all design parameters on the error area estimate.

We have shown how using acoustic wave detectors also enables relaxing the requirements on error codes for protecting SRAM arrays. In future work, we plan to combine them with mechanisms such as early writeback and checkpointing and assess the trade-offs in terms of error detection, recovery and performance cost. We will also explore how to exploit their detection capabilities to cover the unprotected logic elements in processors.

8. Acknowledgements

This work has been partially supported by the Spanish Ministry of Education and Science under grant TIN2010-18368, the TRAMS project of the FP7 program of the European Commission under agreement 248789, the Generalitat of Catalunya under grants 2009SGR1250 and FI-DGR-2010, and Intel Corporation.

References

- [1] T. Austin. DIVA: a reliable substrate for deep submicron microarchitecture design. In *Proceedings of International Symposium on Microarchitecture (MICRO)*, 1999.
- [2] R. Baumann. Silicon amnesia: a tutorial on radiation induced soft errors. In *International Reliability Physics Symposium (IRPS)*, 2001.
- [3] R. Baumann. Soft errors in advanced semiconductor devices-part i: the three radiation sources. *IEEE Transactions on Device and Materials Reliability*, 1(1):17–22, 2001.
- [4] R. Baumann. Soft errors in advanced computer systems. In *Proceedings of IEEE Design and Test of Computers*, pages 258–266, Los Alamitos, CA, USA, 2005. IEEE Computer Society.
- [5] L. K. Baxter. *Capacitive Sensors: Design and Applications*. John Wiley and Sons, 1996.

- [6] I. Corporation. *Intel's Nehalem data sheet*. Intel Corporation.
- [7] B. C. Daly, T. B. Norris, J. Chen, and J. B. Khurgin. Picosecond acoustic phonon pulse propagation in silicon. *Phys. Rev. B*, 70:214307, Dec 2004.
- [8] A. Dixit and A. Wood. The impact of new technology on soft error rates. In *Proceedings of the International Reliability Physics Symposium (IRPS)*, 2011.
- [9] W. Foy. Position-Location Solutions by Taylor-Series Estimation. *IEEE Transactions on Aerospace Electronic Systems*, 12:187–194, Mar. 1976.
- [10] M. Hammig. The design and construction of a mechanical radiation detector. In *Proceedings of IEEE Nuclear Science Symposium*, pages 803–805, Dept. of Nucl. Eng., Michigan Univ., Ann Arbor, MI, 1998. IEEE.
- [11] M. Hammig. Nuclear radiation detection via the detection of pliable microstructures. In *Proceedings of Nuclear Instruments and Methods in Physics Research*, pages 278–281, Los Alamitos, CA, USA, 1999. Elsevier Science.
- [12] E. Hannah. Cosmic ray detectors for integrated circuit chips. United States Patent Number 7309866B2, December 2007. Available online (17 pages).
- [13] S. S. Hung L D, Goshima M. Zigzag-hvp: A cost-effective technique to mitigate soft errors in caches with word-based access. In *IPSIJ Digital Courier*, Washington, DC, USA, 2006. IEEE Computer Society.
- [14] 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, 2007. IEEE Computer Society.
- [15] L. Li, V. Degalahal, N. Vijaykrishnan, M. Kandemir, and M. Irwin. Soft error and energy consumption interactions: a data cache perspective. In *Proceedings of the International Symposium on Low Power Electronics and Design (ISLPED)*, 2004.
- [16] 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*, pages 21–24, Los Alamitos, CA, USA, March 2003. IEEE Computer Society.
- [17] C. McMillan and P. McMillan. Characterizing rifle performance using circular error probable measured via a flatbed scanner. Creative Commons Attribution-Noncommercial-No Derivative Works, December 2008.
- [18] S. Mukherjee. *Architecture Design for Soft Errors*. 1st edition, 2009.
- [19] 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.
- [20] S. Mukherjee, M. Kontz, and S. Reinhardt. Detailed design and evaluation of redundant multithreading alternatives. In *Proceedings of International Symposium on Computer Architecture (ISCA)*, 2002.
- [21] C. P. Two-dimensional parity checking. In *Proceedings of International Symposium on Microarchitecture (MICRO)*, Washington, DC, USA, 1961. IEEE Computer Society.
- [22] C. C. Paige and M. A. Saunders. Lsq: An algorithm for sparse linear equations and sparse least squares. *ACM Trans. Math. Softw.*, 8:43–71, March 1982.
- [23] G. Reis, J. Chang, N. Vachharajani, R. Rangan, and D. August. SWIFT: Software implemented fault tolerance. In *Proceedings of the International Symposium on Code Generation and Optimization (CGO)*, 2005.
- [24] G. Reis, J. Chang, N. Vachharajani, R. Rangan, D. August, and S. Mukherjee. Design and evaluation of hybrid fault-detection systems. In *Proceedings of the 32nd International Symposium on Computer Architecture (ISCA)*, 2005.
- [25] E. Rotenberg. AR-SMT: A microarchitectural approach to fault tolerance in microprocessors. In *Proceedings of International Symposium on Fault-Tolerant Computing (FTC)*, page 84, 1999.
- [26] A. Saleh, J. Serrano, and J. Patel. Reliability of scrubbing recovery techniques for memory systems. *IEEE Transactions on Reliability*, 39(1):114–122, 1990.
- [27] N. Seifert, P. Slankard, M. Kirsch, B. Narasimham, V. Zia, B. C. Brookresonand A. Voand S. Mitraand B. Gill, and J. Maiz. Radiation-induced soft error rates of advanced cmos bulk devices. In *Proceedings of International Reliability Physics Symposium*, pages 217–225, Los Alamitos, CA, USA, March 2006. IEEE Computer Society.
- [28] G. Shen, R. Zetik, and R. Thoma. Performance comparison of toa and tdoa based location estimation algorithms in los environment. *Proceedings of Workshop on Positioning, Navigation and Communication (WPNC)*, pages 71–78, 2008.
- [29] K. Sundaramoorthy, Z. Purser, and E. Rotenberg. Slipstream processors: improving both performance and fault tolerance. In *Proceedings of the ninth international conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2000.
- [30] M. William, O. Roger, and M. Daniel. Capacitance bar sensor. United States Patent US4947131, August 1990.