Efficient Error Detection Codes for Multiple-Bit Upset Correction in SRAMs with BICS

PEDRO REVIRIEGO and JUAN ANTONIO MAESTRO
Universidad Antonio de Nebrija

Memories are one of the most widely used elements in electronic systems, and their reliability when exposed to Single Events Upsets (SEUs) has been studied extensively. As transistor sizes shrink, Multiple Bits Upsets (MBUs) are becoming an increasingly important factor in the reliability of memories exposed to radiation effects. To address this issue, Built-in Current Sensors (BICS) have recently been applied in conjunction with Single Error Correction/Double Error Detection (SEC-DED) codes to protect memories from MBUs. In this article, this approach is taken one step further, proposing specific codes optimized to be combined with BICS to provide protection against MBUs in memories. By exploiting the locality of errors within an MBU and the error detection and location capabilities of BICS, the proposed codes result in both a better protection level and a reduced cost compared with the existing SEC-DED approach.

Categories and Subject Descriptors: B.3.4 [**Memory Structures**]: Reliability, Testing, and Fault-Tolerance; B.7.3 [**Integrated Circuits**]: Reliability and Testing; E.4 [**Data**]: Coding and Information Theory

General Terms: Design, Reliability

Additional Key Words and Phrases: Fault tolerant memory, error correcting codes, high-level protection technique, protection against radiation

ACM Reference Format:

Reviriego, P. and Maestro, J. A. 2009. Efficient error detection codes for multiple-bit upset correction in SRAMs with BICS. ACM Trans. Des. Autom. Elect. Syst., 14, 1, Article 18 (January 2009), 10 pages, DOI = 10.1145/1455229.1455247 http://doi.acm.org/10.1145/1455229.1455247

1. INTRODUCTION

As technology scales, the occurrence of Multiple Bit Upsets becomes an increasingly important factor in the reliability of memories as discussed in Radaelli

This research was supported by the Spanish Ministry of Education and Science under project ESP2006-04163.

Authors' addresses: P. Reviriego and J. A. Maestro, Departmento de Ingenieria Informatica, Universidad Antonio de Nebrija, Calle Pirineos 55, 28040 Madrid, Spain; email: {previrie, jmaestro}@nebrija.es.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or direct commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.
© 2009 ACM 1084-4309/2009/01-ART18 $5.00 DOI 10.1145/1455229.1455247 http://doi.acm.org/10.1145/1455229.1455247

et al. [2005], Tipton et al. [2006] and Maiz et al. [2003]. The occurrence of multiple errors in a given event poses a challenge to Single Error Correction, Double Error Detection codes (SEC-DED) that have been traditionally used to protect memories. The most common approach to deal with multiple errors has been the use of interleaving in the physical arrangement of the memory cells, so that cells that belong to the same logical word are separated. As the errors in an MBU are physically close as discussed in Satoh et al. [2000], they will cause single errors in different words that can be corrected by the SEC-DED codes.

However, interleaving cannot be used, for example, in small memories or register files, and in other cases, its use may have an impact on floor-planning, access time and power consumption, as discussed in Dutta and Touba [2007]. For those reasons, the use of more sophisticated codes or the combination of different codes has been proposed in Neuberger et al. [2003] to deal with MBUs when the use of interleaving is not a valid option. More recently, codes that can correct multiple errors only when they are adjacent have also been proposed in Dutta and Touba [2007]. These codes are tailored to the specific patterns of errors in an MBU (again the errors will tend to be physically close) and therefore can achieve effective protection at a reduced cost.

An alternative approach to protect memories is the use of Built in Current Sensors (BICS) that are able to detect the occurrence of errors by detecting changes in the current, as proposed in Vargas et al. [1993] and in Lo [2002]. The sensors are placed in the columns of the memory block and they detect unexpected current variations on each of the memory bit positions. BICS, in conjunction with a per-word parity bit, has been used to protect memories against Single Event Upsets (SEUs) in Vargas and Nicolaidis [1994] and in Calin et al. [1995]. More recently, in Gill et al. [2005a], BICS in conjunction with SEC-DED codes has been proposed to deal with MBUs in memories as an alternative to the traditional combination of interleaving and SEC-DED codes. However, this approach does not fully exploit the locality of the errors within a given MBU and the error location capabilities of the BICS to optimize the protection. This optimization can be achieved with protection codes that are tailored to the specific problem of a memory protected with BICS that suffers MBUs. This is the objective of this paper in which such codes are proposed.

The article is organized as follows. The related work covering BICS for SRAM memory protection is presented in section 2. In section 3, the proposed codes are introduced and their correction capabilities when combined with BICS are discussed. Then, in section 4, the cost of the proposed codes and their protection capabilities are analyzed in detail and compared with the traditional SEC-DED codes. Finally the conclusions of the work are presented.

2. RELATED WORK

Built-in Current Sensors (BICS) were originally proposed as a mechanism for circuit testing, as discussed, for example, in Rubio et al. [1990]. In this case, the objective is to detect physical defects in a given device that may influence its functionality. This is useful during production to rule out defective parts.

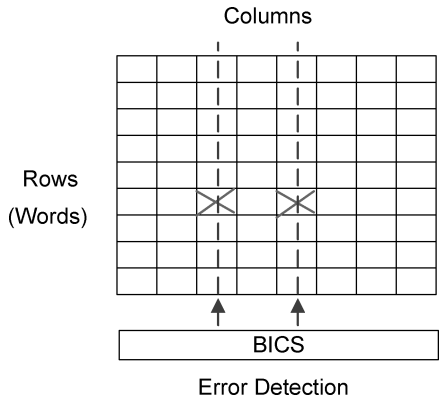

Fig. 1. Example of an MBU in a memory protected with BICS on the columns.

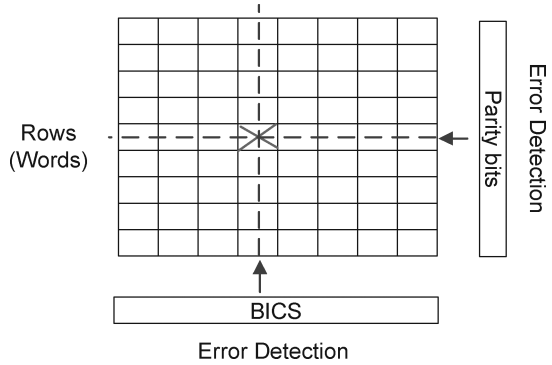

Fig. 2. Example of an SEU in a memory protected with BICS and a per-word parity bit.

In Vargas et al. [1993] the ability of these sensors to detect transient changes in digital circuits was used to identify the occurrence of SEUs in digital circuits. That work was extended by applying BICS for SRAM memory protection in Vargas and Nicolaidis [1994] and in Calin et al. [1995], where BICS are combined with a per-word parity bit to provide effective protection against SEUs. The BICS are placed on the vertical power lines of the memory and they provide the bit position on which a failure has occurred, but are not able to locate the exact word, as illustrated in Figure 1. When BICS are combined with a per-word parity bit as shown in Figure 2, the location of a SEU can be fully determined and therefore the error can be corrected.

More recently, in Gill et al. [2005a], BICS in conjunction with SEC-DED codes has been proposed to deal with MBUs in memories as an alternative to the traditional combination of interleaving and SEC-DED codes. This new application of BICS would be useful in situations where the use of interleaving is not possible or appropriate as discussed before. The combination of BICS with error protection codes can correct MBUs such as the one shown in Figure 3, where the BICS detect errors on the columns and the error protection code detects errors on the rows (words). In particular, for the MBU shown in

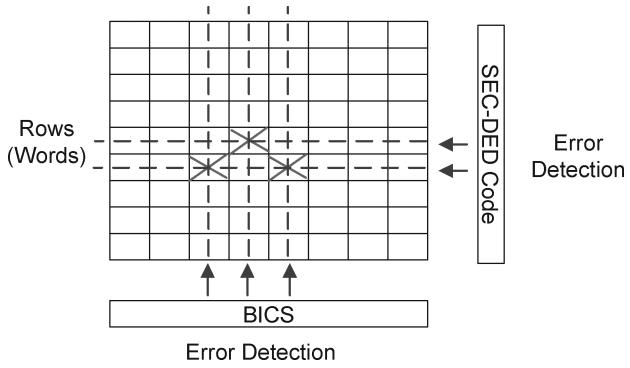

Fig. 3. Example of an MBU in a memory protected with BICS and SEC-DED codes.

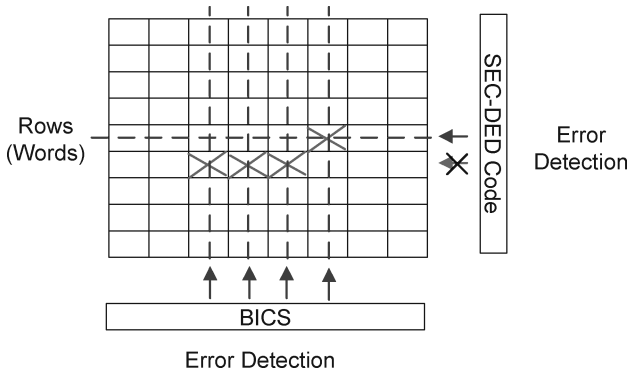

Fig. 4. Example of an MBU in a memory protected with BICS and SEC-DED codes.

that figure, the SEC-DED code can detect that there have been errors in two words since there are only two errors per-word in the worst case. Using the additional information from the BICS about the faulty columns, the errors could be located, and combining that information with the one from the SEC-DED codes, the errors could be corrected in most cases, as shown in Gill et al. [2005a].

However, for the MBU shown in Figure 4, the SEC-DED codes proposed in Gill et al. [2005a] cannot guarantee the error detection (since three errors are produced in the same word) and therefore they are of little use when dealing with MBUs that affect more than two bits per word. This is an important limitation, as MBUs that involve three or more bits have been reported for various radiation sources and process technologies in Radaelli et al. [2005] and in Maiz et al. [2003]. It is true, that in general, the occurrence of MBUs with larger number of errors is less likely and that they may be a reduced percentage of the total number of events as shown in Chugg et al. [2004], but even in that case they can have a large impact on the memory reliability, specially if scrubbing is used to increase the Mean Time to Failure (MTTF). This is so because a single MBU can cause the failure of the memory so that the MTTF due to this effect would be the average time to get such an event. For an event arrival rate λ and

a percentage of events with three or more errors p_{3+}, the MTTF would be:

$$MTTF = \frac{1}{\lambda \cdot p_{3+}}, \tag{1}$$

which compares with the MTTF for scrubbing when two events are needed to cause a failure and all possible two-event combinations do cause a failure. This, following Saleh et al. [1990], can be approximated for large memories as:

$$MTTF \cong \frac{2 \cdot M}{\lambda^2 \cdot t_s}, \tag{2}$$

where M is the number of words in the memory and t_s is the scrubbing period. It can be seen that Equation (1) will dominate the MTTF if

$$\frac{\lambda \cdot t_s}{2 \cdot M} \ll p_{3+}. \tag{3}$$

But in most cases $\lambda \cdot t_s \ll 1$ and $M \gg 1$ so that Equation (3) is true even for very small values of p_{3+}. The same reasoning applies to memories protected with BICS as in that case the correction takes place shortly after the error occurs which is similar to a small scrubbing period.

Finally, the practical applicability of all the protection techniques that use BICS depends strongly on the efficient and reliable implementation of the current sensors. Such implementations have been recently reported in Gill et al. [2005b] where a 100nm technology is used and the BICS reliability is evaluated across process, voltage, temperature and noise supply. The BICS is built with 27 transistors and shared among 256 memory cells so that the area and power overhead should be small. Another implementation is reported in Neto et al. [2005], also for a 100nm technology, which confirms the feasibility of BICS for advanced technologies.

3. PROPOSED CODES

Following the traditional assumptions when dealing with MBUs in memories, it can be assumed that the maximum distance between errors in an MBU is bounded by a certain value, as shown in Radaelli et al. [2005], Reviriego et al. [2007], and Satoh et al. [2000]. This maximum distance is used for example to select the interleaving pattern when this technique is implemented. In our case, we are interested in the maximum horizontal distance between the errors in an MBU. That is the difference between the values of the columns of the leftmost and rightmost errors in the MBU. As an example, for the MBU in Figure 3 that value is two while for the MBU of Figure 4 it is three. Let us define the maximum horizontal distance between the errors in any MBU as L-1. Given this assumption, all errors in a given word will occur in a group of up to L consecutive columns. Therefore, what is needed are codes that in conjunction with BICS can detect errors that affect up to L consecutive bits and also locate them within those L columns.

This can be achieved with a code that performs the parity computations shown in Equation (4) for an N-bit memory word.

$$P_j = b_j \oplus b_{j+L} \oplus b_{j+2L} \cdots \oplus b_{j+L \cdot floor(\frac{N-j}{L})} \qquad j = 1, \ldots, L \tag{4}$$

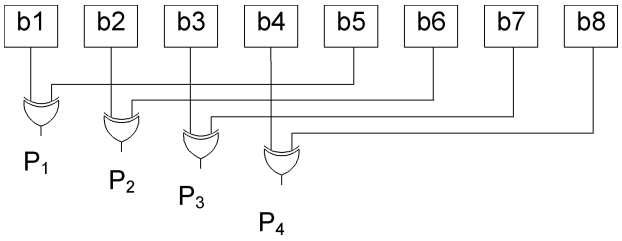

Fig. 5. Example of the proposed code parity checks for L = 4 and N = 8.

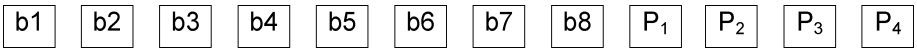

Fig. 6. Example of the proposed code parity bits arrangement for L = 4 and N = 8.

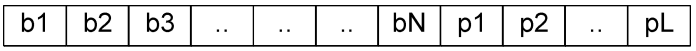

Fig. 7. Memory word organization for the proposed error detection code for N/L integer.

From the definitions of the P_j, failures in consecutive bits of up to L bits will only affect one bit in any given P_j. The proposed parity checks are illustrated for L = 4 and N = 8 in Figure 5, in which it can be seen that any possible MBU pattern with L less or equal than four will affect only one bit per-parity check. As the BICS would provide the up to L columns in which a failure has occurred, the correcting process becomes trivial. For each word check, if there is a parity error, then find for that parity error the bit on which the BICS has detected an error. Once that bit is located, invert the bit to correct the error.

A final detail is that errors can also affect the cells that store the parity bits. This can pose a problem if a parity bit and one of the bits involved in its computation are both affected by errors in an MBU. This will cause a situation in which there is no parity error and the failure is not even detected by the system. To avoid this type of effect, the P_j bits have to be carefully arranged. This is shown in Figure 6 for the example presented in Figure 5.

In general, this can be achieved when N is divisible by L using the memory word organization shown in Figure 7. In this case Equation (4) becomes

$$\begin{aligned} P_j &= b_j \oplus b_{j+L} \oplus b_{j+2L} \ldots \oplus b_{j+N-L} \qquad j = 1, \ldots, L-1 \\ P_L &= b_L \oplus b_{2L} \oplus b_{3L} \ldots \oplus b_N. \end{aligned} \tag{5}$$

And therefore the distance between the last bit involved in a parity check and the corresponding parity bit using the memory word organization shown in Figure 7 is L.

In the rest of the cases, a similar reasoning can be used by noticing that the last bit b_N will be used for a given parity check that we denote as p_k. Then this p_k should be placed at the rightmost position to ensure that it is at a distance L of b_N. Then the following rightmost position should be used for the parity

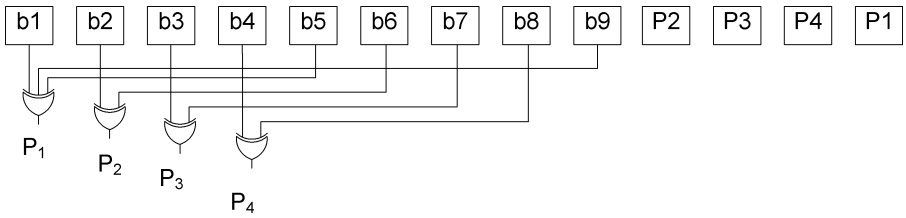

Fig. 8. Example of the proposed code parity checks and bits arrangement for L = 4 and N = 9.

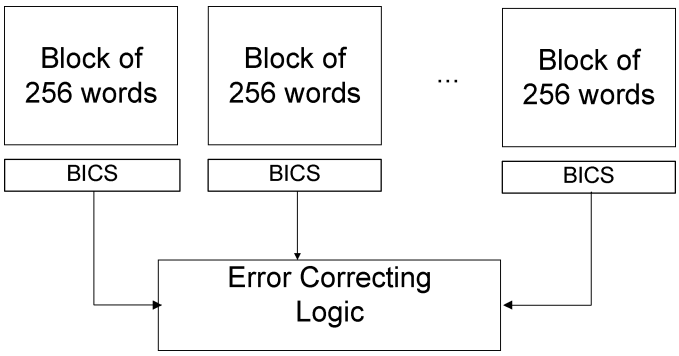

Fig. 9. Example of memory implementation using BICS for protection.

check bit that includes bit b_{N-1} and so on. With this strategy the parity bits are always at distance L of the bits used to compute them. As an example the approach is illustrated in Figure 8 for L = 4 and N = 9.

In a memory protected with the proposed codes and BICS, the correction will be done as follows. The current sensors will be shared across a number of cells, for example 256 as reported in Gill et al. [2005b] and we assume that the correction logic is shared among a number of blocks, as illustrated in Figure 9. The detail of each block is shown in Figure 10 where also an MBU is illustrated showing in red the affected cells and in green the BICS that will detect an error. When an error such as the one shown in Figure 10 occurs, the sensors will issue an error signal that triggers the correction process. Then the block of words that corresponds to the sensors that have triggered a failure are read word by word. For each of them the parity bits are recalculated, and in the case of error the bit (again, by design there can only be one error in each parity check) for which the corresponding BICS has reported an error is inverted. In the example of Figure 10, p_1 and p_L will produce an error in word 255 and p_2 in word 256 and using the information of the faulty columns given by the BICS the errors are corrected.

4. COST AND PROTECTION ANALYSIS

In a general case the cost of the proposed method is: L bits per-word to store the parity bits, ((N/L)-1)*L xor gates to compute the parity checks and the logic needed to combine the BICS information with the parity checks and perform

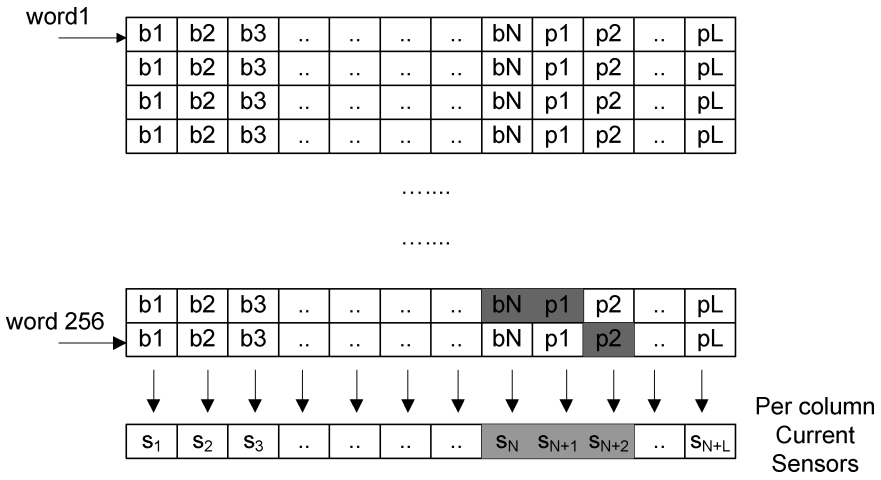

Fig. 10. Detail of a block implementation.

Table I. Cost of the Proposed Codes

N	Method in Gill et al. [2005a]	Proposed L = 2	Proposed L = 3	Proposed L = 4
8	4 bits, 14 xor	2 bits, 6 xor	3 bits, 5 xor	4 bits, 4 xor
16	6 bits, 48 xor	2 bits, 14 xor	3 bits, 13 xor	4 bits, 12 xor
32	7 bits, 96 xor	2 bits, 30 xor	3 bits, 29 xor	4 bits, 28 xor

the correction. The xor gates for encoding and decoding and the additional correction logic can be shared among a number of words and therefore it makes sense to analyze the required bits and the xor gates and correction logic independently. The costs for different memory word sizes and MBU sizes are shown in Table I, where the cost for the traditional SEC-DED codes proposed to be combined with BICS in Gill et al. [2005a] is also shown (note that the cost details for those codes is given Dutta and Touba [2007]). The cost of the correction logic has not been included as there are different alternatives to implement it and it is therefore hard to make a fair comparison, but in any case the correction logic will be more complex in the case of the codes proposed in Gill et al. [2005a].

In order to compare the protection provided by the proposed codes versus the one provided by SEC-DED codes in Gill et al. [2005a], it must be noted that SEC-DED codes will be unable to detect some of the errors that affect three or more bits of the same word as already discussed. Therefore those errors will not be corrected. In that regard, the approach proposed in Gill et al. [2005a] only guarantees the correction of double horizontal MBUs (L = 2). If a parity bit is added to the SEC-DED codes as suggested in Gill et al. [2005a], correction of triple horizontal MBUs is guaranteed at an extra cost. Therefore the technique proposed in Gill et al. [2005a] can be compared with the cases L = 2 and L = 4. In the first case the proposed technique will provide a slightly lower protection and in the last case a superior one.

The relative cost comparison for the proposed codes versus the SEC-DED codes proposed in Gill et al. [2005a] is shown in Table II, for different memory

Table II. Relative Cost of the Proposed Codes versus the Traditional SEC-DED Codes

N	Proposed L = 2		Proposed L = 4	
	Bits	xor	Bits	xor
8	50%	42%	100%	29%
16	33%	29%	66%	25%
32	28%	31%	57%	29%

word sizes. It can be seen that the proposed codes result in a significant reduction in complexity when compared with traditional SEC-DED codes, even for L = 4 (that would provide a superior level of protection). The maximum logic depth of the parity checks is also reduced, which in conjunction with a simpler correction logic results in a reduction in the time needed to correct an error which would be another advantage of the proposed approach.

This cost reduction is a direct consequence of designing the codes for this particular application in which BICS provide useful information to locate errors, and errors in an MBU are assumed to occur physically close.

One aspect in which the proposed approach will be weaker than the one proposed in Gill et al. [2005a] is in the case of consecutive SEUs that affect distant bits of the same word. In this case the BICS would detect the error immediately and the correction would take place closely after the first error has occurred and therefore a failure would occur only if two SEUs arrive very close in time. However, the probability of two events occurring very close in time would be negligible in most environments and is in fact the assumption behind the scrubbing techniques that are widely used in memory systems as shown in Saleh et al. [1990]. Assuming that all combinations of two SEUs would cause a failure (which is a conservative assumption) the MTTF will depend on the probability that a second SEU arrives before the correction takes place. For the approach proposed in Gill et al. [2005a], assuming MBUs with two errors per word, two events will cause a failure if in total they affect three or more bits in a word. Therefore, the MTTF will depend on the probability that a second event arrives before the correction takes place and that between both events they affect three or more bits so that only a few of the possible two event combinations would cause a failure. So, in summary the use of the SEC-DED codes would result in a larger MTTF for the particular case of L = 2. However, for the general case where L > 2, the proposed codes would provide a superior protection level since the approach in Gill et al. [2005a] is unable to deal with MBUs that involve more than two errors per-word as discussed before.

5. CONCLUSIONS

In this article, efficient codes have been proposed to protect memories against MBUs in combination with BICS. The codes exploit the locality of the errors in an MBU and the capabilities of the BICS to achieve a reduction in the implementation cost and an increase of the protection level against MBUs compared with existing solutions.

Future work will focus on a comprehensive analysis of the implementation of the proposed codes in conjunction with BICS in a memory prototype.

REFERENCES

CALIN, T., VARGAS, F. L., AND NICOLAIDIS, M. 1995. Upset-tolerant CMOS SRAM using current monitoring: prototype and test experiments. In *Proceedings of the International Test Conference*. 45–53.

CHUGG, A. M., MOUTRIE, M. J., AND JONES, R. 2004. Broadening of the variance of the number of upsets in a read-cycle by MBUs. *IEEE Trans. Nucl. Sci. 51*, 6, 3701–3707.

DUTTA, A. AND TOUBA, N. A. 2007. Multiple bit upset tolerant memory using a selective cycle avoidance based SEC-DED-DAEC code. In *Proceedings of the IEEE VLSI Test Symposium,* 349–354.

GILL, B., NICOLAIDIS, M., AND PAPACHRISTOU, C. 2005a. Radiation induced single-word multiple-bit upsets correction in SRAM. In *Proceedings of the IEEE International OnLine Testing Symposium*. 266–271.

GILL, B., NICOLAIDIS, M., WOLFF, F., PAPACHRISTOU, C., AND GARVERICK, S. 2005b. An efficient BICS design for SEUs detection and correction in semiconductor memories. In *Proceedings of the Design, Automation and Test in Europe.* 592–597.

LO, J. 2002. Analysis of a BICS-only concurrent error detection method. *IEEE Trans. Comput. 51*, 3, 241–253.

MAIZ, J., HARELAND, S., ZHANG, K., AND ARMSTRONG, P. 2003. Characterization of multi-bit soft error events in advanced SRAMs. In *Proceedings of the IEEE International Electron Devices Meeting.* 21.4.1–21.4.4.

NETO, E. H., RIBEIRO, I., VIEIRA, M., WIRTH, G., AND KASTENSMIDT, F. L. 2005. Evaluating Fault Coverage of Bulk Built-in Current Sensor for Soft Errors in Combinational and Sequential Logic. In *Proceedings of the Symposium on Integrated Circuits and Systems Design*. 62–67.

NEUBERGER, G., DE LIMA, F., CARRO, L., AND REIS, R. 2003. A multiple bit upset tolerant SRAM memory. *ACM Trans. Desi. Automat. Electr. Syst. 8*, 4, 577–590.

RADAELLI, D., PUCHNER, H., WONG, S., AND DANIEL, S. 2005. Investigation of multi-bit upsets in a 150 nm technology SRAM device. *IEEE Trans.Nucl. Sci. 52*, 6, 2433–2437.

REVIRIEGO, P., MAESTRO, J. A., AND CERVANTES, C. 2007. Reliability analysis of memories suffering multiple bit upsets. *IEEE Trans. Device Materials Reliabil. 7*, 4, 592–601.

RUBIO, A., FIGUERAS, J., AND SEGURA, J. 1990. Quiescent current sensor circuits in digital VLSI CMOS testing. *Electron. Lett. 26*, 15, 1204–1206.

SATOH, S., TOSAKA, Y., AND WENDER, S. A. 2000. Geometric effect of multiple-bit soft errors induced by cosmic ray neutrons on DRAM's. *IEEE Electron Device Lett. 21*, 6, 310–312.

SALEH, A. M., SERRANO, J. J., AND PATEL, J. H. 1990. Reliability of scrubbing recovery-techniques for memory systems. *IEEE Trans. Reliability 39*, 1, 114–122.

TIPTON, D., PELLISH, J. A., REED, R. A., SCHRIMPF, R. D., WELLER, R. A., MENDENHALL, M. H., SIERAWSKI, B., SUTTON, A. K., DIESTELHORST, R. M., ESPINEL, G., CRESSLER, J. D., MARSHALL, P. W., AND VIZKELETHY, G. 2006. Multiple-bit upset in 130 nm CMOS technology. *IEEE Trans. Nucl. Sci. 53*, 6, 3259–3264.

VARGAS, F., NICOLAIDIS, M., AND COURTOIS, B. 1993. Quiescent current monitoring to improve the reliability of electronic systems in space radiation environments. In *Proceedings of the IEEE International Conference on Computer Design (ICCD),* 596–600.

VARGAS, F. AND NICOLAIDIS, M. 1994. SEU-tolerant SRAM design based on current monitoring. *Proceedings of the International Symposium on Fault-Tolerant Computing,* 106–115.

Received December 2007; revised April 2008, July 2008; accepted July 2008