Reliability Analysis of Memories Protected with BICS and a per-Word Parity Bit

PEDRO REVIRIEGO and JUAN ANTONIO MAESTRO
Universidad Antonio de Nebrija
and
CHRIS J. BLEAKLEY
University College Dublin

This article presents an analysis of the reliability of memories protected with Built-in Current Sensors (BICS) and a per-word parity bit when exposed to Single Event Upsets (SEUs). Reliability is characterized by Mean Time to Failure (MTTF) for which two analytic models are proposed. A simple model, similar to the one traditionally used for memories protected with scrubbing, is proposed for the low error rate case. A more complex Markov model is proposed for the high error rate case. The accuracy of the models is checked using a wide set of simulations. The results presented in this article allow fast estimation of MTTF enabling design of optimal memory configurations to meet specified MTTF goals at minimum cost. Additionally the power consumption of memories protected with BICS is compared to that of memories using scrubbing in terms of the number of read cycles needed in both configurations.

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, built-in current sensors

ACM Reference Format:

Reviriego, P., Maestro, J. A., and Bleakley, C. J. 2010. Reliability analysis of memories protected with BICS and a per-word parity bit. ACM Trans. Des. Autom. Electron. Syst. 15, 2, Article 18 (February 2010), 15 pages. DOI = 10.1145/1698759.1698768 http://doi.acm.org/10.1145/1698759.1698768

This work was supported in part by the Spanish Ministry of Science and Education under Grant AYA-2009-13300-C03-01, by the Regional Government of Madrid, and by the European Union FEDER program.

Authors' addresses: P. Reviriego and J. A. Maestro, Departamento de Ingeniería Informática, Universidad Antonio de Nebrija, Calle Pirineos 55, 28040 Madrid, Spain; email: {previrie, jmaestro}@nebrija.es; C. J. Bleakley, University College Dublin, Belfield, Dublin 4, Ireland; email: chris.bleakley@ucd.ie.

DOI 10.1145/1698759.1698768 http://doi.acm.org/10.1145/1698759.1698768

1. INTRODUCTION

For many years, soft errors have been a major concern for circuits that operate in harsh environments, such as space [Gosset et al. 1993]. Due to technology scaling, soft errors are also becoming an increasingly important factor in terrestrial applications [Normand 1996; Schrimpf et al. 2004]. One type of soft error is the Single Event Upset (SEU) [Nicolaidis 2005; May et al. 1978; Mavi et al. 2002]. These errors cause the value of a register or storage element to change. When an SEU affects a memory device, the error may lead to system failure. Given the broad use of memories in electronic systems, their reliability is of major concern. Consequently, the effect of SEUs has been widely studied in the literature.

Previous studies have focused on reliability analysis for memories protected with Single Error Correction (SEC) codes [Goodman et al. 1982; Blaum et al. 1988]. Typically, a code is applied to every memory word. The codes allow correction of single errors. Consequently at least two errors on the same word are needed to cause a failure. A commonly used complementary technique is scrubbing. In scrubbing, memory words are read periodically and any errors are corrected. In this way, the accumulation of errors over time is avoided, thus minimizing the probability of failure [Saleh et al. 1990; Goodman et al. 1991; Yang 1995]. More recent research focuses on the effects of Multiple Bit Upsets (MBUs) [Radaelli et al. 2005; Tipton et al. 2006; Maiz et al. 2003; Chugg et al. 2004] on memory reliability [Reviriego et al. 2007]. MBUs affect bits stored on physically adjacent memory cells. The most common approach to deal with MBUs in memories is to interleave bits such that the bits from a logical word are physically separated [Satoh et al. 2000; Tosaka et al. 2004]. This ensures that only one bit per word is affected by a single MBU event.

Another approach to protect memories is the use of Built-In Current Sensors (BICS). BICS were originally proposed as a mechanism for circuit testing [Rubio et al. 1990]. Circuit testing aims to detect physical defects in a device that may influence its functionality. It is used during production to identify and facilitate rejection of defective parts. However, BICS can also be used for memory protection. This possibility was first noted in Vargas et al. [1993] where the use of BICS was proposed as a means to identify the occurrence of SEUs in digital circuits. BICS used in combination with a per-word parity bit was subsequently proposed in Vargas et al. [1994] and Calin et al. [1995] for SRAM protection. In these proposals, BICS were placed on the power lines of the memory. When a SEU occurs, the per-word parity bit identifies the word in error and the BICS identify the bit position in error. Thus the location of a SEU can be determined and the error corrected. Usage of BICS in combination with per-word SEC codes has been proposed as a means to deal with Multiple Bit Upsets (MBUs) [Gill et al. 2005a]. Recently, specific error correction codes error to deal with MBUs in memories protected with BICS have also been developed [Reviriego et al. 2009] The implementation of efficient and reliable BICS has been addressed recently for advanced memory technologies (100nm) [Gill et al. 2005b; Neto et al. 2005]. These results are promising and indicate an increasing interest in BICS-based memory protection.

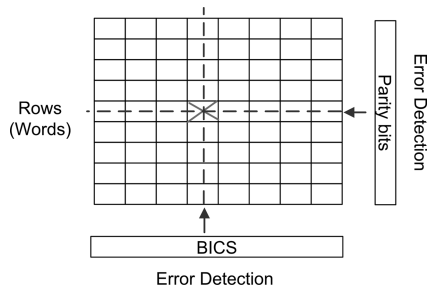

Fig. 1. Example of an SEU in a memory block protected with BICS and a per-word parity bit.

As mentioned before, the reliability of memories protected with scrubbing and SEC codes has been widely studied in the literature. However, only one work is available covering the reliability of memories protected with BICS and a per-word parity bit [Argyrides et al. 2008]. In this work it is assumed that the error detection performed by the BICS does not trigger a correction process and the errors accumulate in the memory. Therefore the analysis is similar to that of memories protected with Single Error Correction, Double Error Detection (SEC-DED) codes when scrubbing is not used [Blaum et al. 1988]. To the best of the authors' knowledge, there is no previous analysis of the reliability of memories protected with BICS when the error detection done by the BICS triggers a correction process. This seems to be an interesting configuration as it increases the reliability with little additional cost. Therefore the purpose of this article is to analyze the reliability of memories protected with BICS in terms of the MTTF when error detection by the BICS triggers a correction.

The rest of the article is organized as follows: in Section 2, the BICS-based memory architecture is described, providing the basis for the reliability models presented in Section 3. In Section 4, the models are validated with an extensive set of simulations, illustrating their use in selecting the optimal memory configuration. Finally, in Section 5 the conclusions of the work are presented.

2. MEMORY ARCHITECTURE

In this section, a memory architecture using BICS for protection against SEUs is presented to illustrate the failure mechanisms and provide a foundation for the reliability analysis presented in the following sections.

The memory architecture includes a per-word parity check and is composed of blocks that share BICS as illustrated in Figure 1. BICS identify the bit position of the error and the parity checks indicate the word in error. In this way, any bit error can be located and corrected. In Gill et al. [2005b] the proposed size for a block is 256 words, so a large memory will have a large number of blocks.

As detailed in Calin et al. [1995], once a SEU hits one of the blocks, the corresponding BICS will detect an error and trigger the correction process, for example, by issuing an interrupt to the processor. This process consists of sequentially reading the words in the memory block and for each word performing

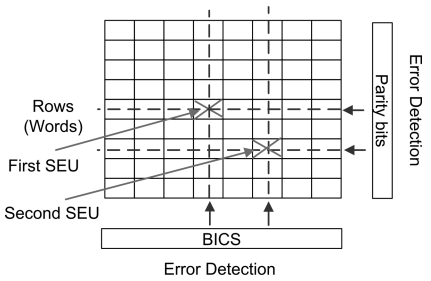

Fig. 2. Example of two SEUs causing a failure in a memory block protected with BICS.

a parity check to see if it has been affected by an SEU. Once the word in error is found, the bit for which the BICS detected a failure is inverted and the column error flag for that block is cleared, ending the correction process. Assuming that SEUs are randomly distributed over the block, the correction time will be random and uniformly distributed between the best case of the SEU occurring in the first word of the block and the worst case of the SEU occurring in the last word of the block.

Correction will fail if a second SEU hits the same block before the correction process finishes. If the second SEU occurs in the same word as the first, no error will be detected by the parity check. If the error hits a different bit position in a different word in the same block, correction will fail because the system will be unable to determine which column fault corresponds to which word parity error, as shown in Figure 2. Finally, if the second error hits the same column as the first one, correction will stop as soon as the first error is found and the second error would not be corrected. In summary, correction fails if a second SEU hits the same block before the first SEU has been corrected.

The proposed correction process could be modified to avoid failure in some cases. For example, the entire block could be checked so that multiple errors in the same column are corrected. Also, the order of arrival of column errors could be recorded and used to identify errors in cases where the second error hits a word after the word has been checked by the correction process but before the process for identifying the first error is complete. Those modifications would provide correction in a few cases but at the expense of increased complexity and correction time. Hence, they are not considered in this work. So far, the discussion has focused on a single block but a memory will normally consist of many of such blocks. During the correction process, another error may occur in a different block. In this case, correction of the second error can only start when the correction of the first completes. This complicates estimation of the correction time as it now depends on previous error arrivals. This will be discussed in more detail in the following section.

3. RELIABILITY MODELS

Following previous memory reliability models [Blaum et al. 1988; Saleh et al. 1990; Goodman et al. 1991] error arrivals are assumed herein to follow a Poisson process. The error arrival rate per memory block is denoted as λ and memories

composed of M blocks of size B words are considered. Finally, the average correction time for an individual block is denoted as t_c.

3.1 Simple Model

A simple model for the MTTF can be derived assuming that the arrival rate is such that $\lambda \cdot M \cdot t_c \ll 1$. We refer to this as the low arrival rate case. In this case, most of the time, the memory will be in one of two states when an error arrives: (i) there is no error in the memory or (ii) there is one single error. This is so because the probability of having k events on an interval t_c is defined as

$$P_a(k) = \frac{(\lambda \cdot M \cdot t_c)^k}{k!} \cdot e^{-\lambda \cdot M \cdot t_c} \tag{1}$$

Given the assumption $\lambda \cdot M \cdot t_c \ll 1$, the following expression holds.

$$P_a(0) \gg P_a(1) \gg P_a(2)..... \tag{2}$$

Under these conditions, most failures occur when an error arrives before a single previous error has been corrected. This will happen with probability

$$P_f \cong P_a(1) \cdot \frac{1}{M} \cong \lambda \cdot t_c, \tag{3}$$

where the second term is the probability that the second error falls in the same block as the first and therefore causes a failure. From Eq. (3) the Mean Events to Failure (METF) can be calculated as

$$METF_{BICS} \cong \sum_{i=1}^{\infty} [(1 - P_f)^{i-1}] = \frac{1}{P_f} = \frac{1}{\lambda \cdot t_c}. \tag{4}$$

Using the well-known relationship between the MTTF and the METF for Poisson processes, we obtain

$$MTTF_{BICS} = \frac{METF}{\lambda \cdot M} \cong \frac{1}{\lambda^2 \cdot M \cdot t_c}. \tag{5}$$

which is similar to the traditional expression for memories protected with scrubbing [Saleh et al. 1990]

$$MTTF_{scrubbing} \cong \frac{2 \cdot B}{\lambda^2 \cdot M \cdot t_s}. \tag{6}$$

The main difference is the B factor (block size). This is related to the fact that in scrubbing the second error has to fall in the word in which the first error occurred in order to cause a failure (considering SEC protection). In the case of BICS protection, failure occurs if the second error falls on the same block of B words where the first error occurred.

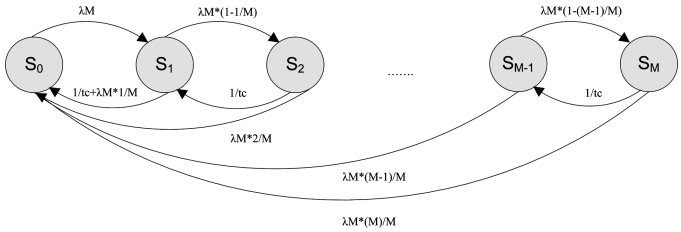

Fig. 3. Proposed Markov model for the BICS protected memory.

The low error arrival rate assumption also ensures that the probability of errors accumulating in the memory while previous ones are being corrected is very low. Therefore, this effect can be neglected.

3.2 Markov Model

A more elaborated model can be used in situations where the low arrival rate assumption is not valid. In this case, memory behavior can be represented by the Markov model shown in Figure 3. The states correspond to different numbers of errors in the memory: S_0 represents zero errors, S_1 represents one error, and so forth. A transition to a new state is caused by an error arrival or correction of an existing error. For an arrival, the transition is to a state that has one more error, if the new error occurs in a block that has no previous errors, or is to the initial state S_0 if the error falls in a block that has an existing error. This latter transition models a failure as a restart of the system. For a correction, the transition is always to the state that has one less error. It should be noted that the correction time is independent of the accumulated number of errors.

Solving the Markov model provides the probability of finding the memory in each state. These probabilities can be used to calculate the probability of failure on the arrival of a new event as

$$P_f = \sum_{i=1}^{M} \left(P(S_i) \cdot \frac{i}{M} \right). \tag{7}$$

From which, following similar reasoning to the one used in the derivation of the simple model, the MTTF can be derived as

$$MTTF = \frac{1}{\lambda \cdot M \cdot \sum_{i=1}^{M} \left(P(S_i) \cdot \frac{i}{M} \right)}. \tag{8}$$

For the Markov model to be applicable, the distribution of arrival times and correction durations should be exponential. In our case this is true for arrival times, as a Poisson distribution has been assumed. However, it is not valid for correction durations. These are uniformly distributed between the best and worst case, as discussed before. Therefore, the Markov model only provides an

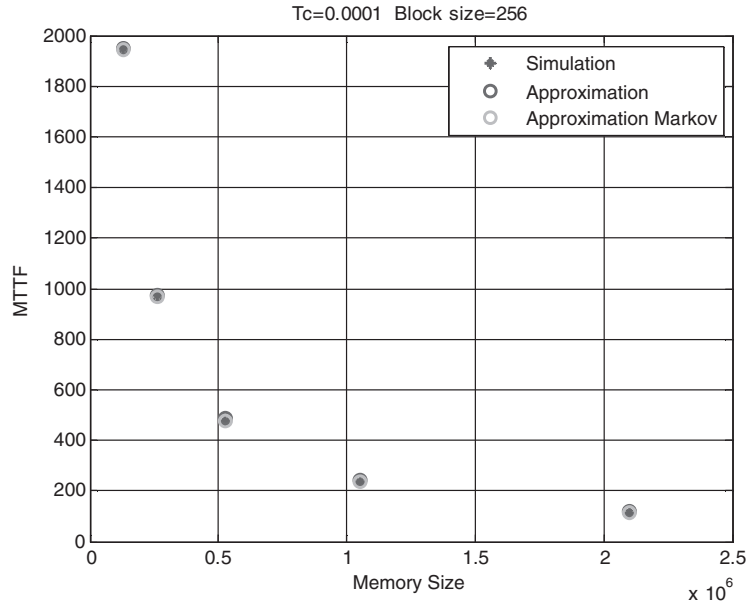

Fig. 4. MTTF simulation results and model estimates for the first experiment.

approximation that can be used to obtain an estimate of the MTTF in cases for which the simple model is not valid.

4. SIMULATION RESULTS

To evaluate the accuracy of the proposed models, an extensive set of simulation experiments was conducted. In the simulations, errors were inserted following a Poisson process while for the correction time a uniform distribution with mean t_c was used. Corrections were performed one at a time so that errors on different blocks can accumulate, as discussed before.

4.1 Simple and Markov Model Validation

The first set of experiments was conducted using a per-block arrival rate λ of 0.1 per time unit, a block size $B = 256$, and an average correction time $t_c = 0.0001$ time units. The results for different memory sizes are shown in Figure 4. In this case, the maximum value of $\lambda \cdot M \cdot t_c$ is 0.082, so the simple model is valid and works reasonably well. This is better appreciated in Figure 5 where the ratio of the MTTF given by the models and the results obtained by simulation are shown. In this case, the simple model is sufficient and there is little advantage in using the Markov model.

For the second set of experiments, the average correction time is increased to 0.001 so that now $\lambda \cdot M \cdot t_c$ increases to 0.82 for the largest memory sizes. In this case, as shown in Figure 6 and Figure 7, the simple model overestimates the MTTF as error accumulation in different blocks is not captured. The Markov model also deviates somewhat from the simulation results due to the fact that

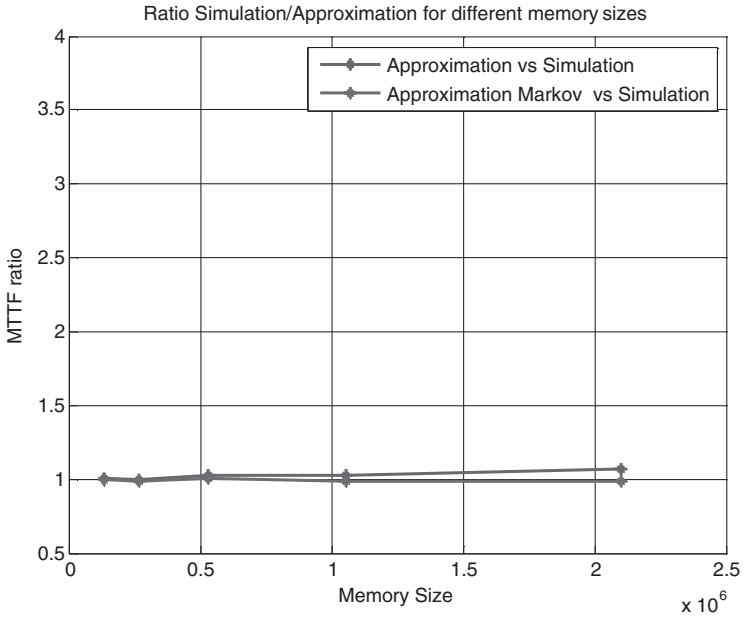

Fig. 5. MTTF ratio for model estimates and simulation results for the first experiment.

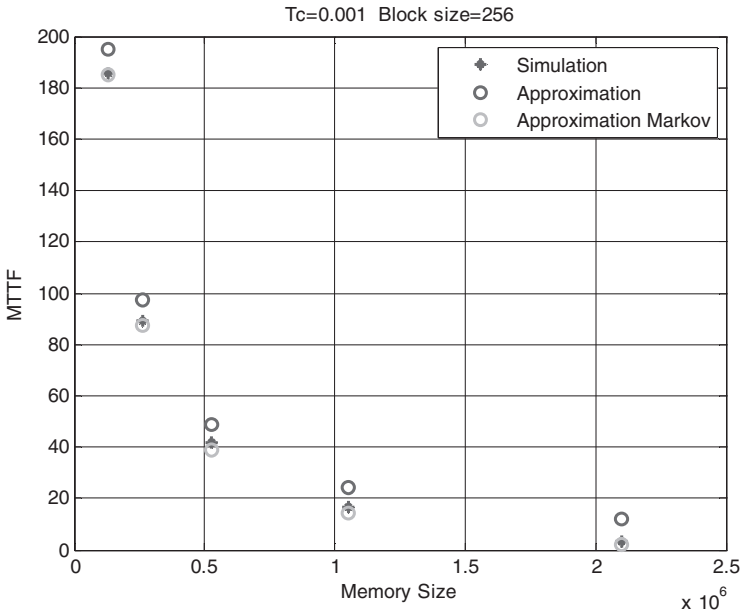

Fig. 6. MTTF simulation results and model estimates for the second experiment.

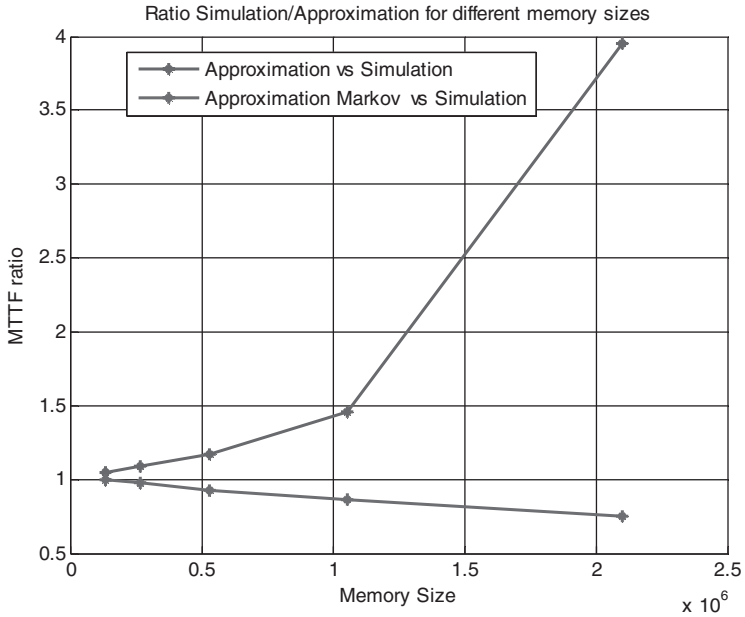

Fig. 7. MTTF ratio for model estimates and simulation results for the second experiment.

the correction times are not exponentially distributed. Nevertheless, it provides a more reliable and more conservative estimate than the simple model.

To perform a sanity-check for the Markov model, a set of simulations assuming exponentially distributed correction durations was conducted using the configuration employed in the second experiment. The results are plotted in Figure 8 where it can be seen that the Markov model estimates accurately match the simulation results.

4.2 Effect of Block Size on Reliability

Once the models were validated, experiments were conducted to determine how MTTF varies with the main design parameter for BICS protection: block size, B. Block size has a direct impact on the area overhead of protection as the smaller the block size, the larger the number of blocks needed for a given memory size. The number of current sensors grows linearly with the number of blocks, as does the area overhead. On the other hand, larger block sizes increase the probability of two errors falling in the same block and causing a failure. So, MTTF should decrease with the block size. Block size also has an effect on average correction duration: the smaller the block, the shorter the correction time. This can be easily seen in the simple model. If Eq. (5) is rewritten using the per-word arrival rate λ', the memory size in words S, and correction time t_{c_norm} normalized to a reference block size B_{norm}, we obtain

$$MTTF \cong \frac{B_{norm}}{(\lambda')^2 \cdot B^2 \cdot S \cdot t_{c\_norm}}, \tag{9}$$

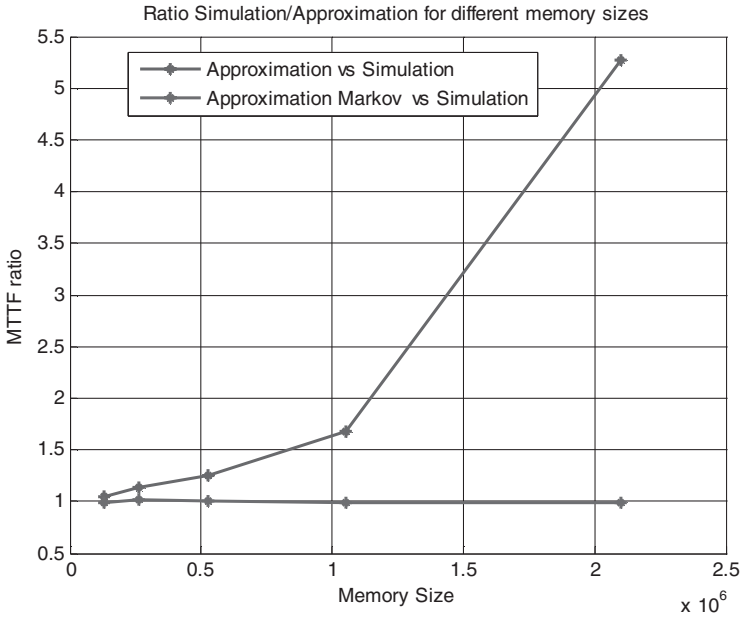

Fig. 8. MTTF ratio model estimates and simulation results for the second experiment with exponentially distributed correction time.

from which, the dependency of MTTF on block size can be clearly seen. To illustrate this, a third set of simulations was conducted in which MTTF was evaluated for various block sizes with a constant memory size, correction duration, and event arrival rate. The results are shown in Figure 9, Figure 10, and Figure 11. It can be observed that MTTF is increased by approximately a factor of four each time the block size is reduced by two, as predicted by the model.

4.3 Effect of Block Size on Power Consumption

In a final experiment, the power consumption of the BICS approach was compared to that of the traditional scrubbing process. The number of read cycles was used as a figure of merit for this study. In order to make the results comparable, the conditions for the experiment were selected such that the reliability of both techniques was the same. The techniques have the same reliability when Eqs. (5) and (6) are equal. This leads to the following relation between t_s and t_c.

$$t_s = 2 \cdot B \cdot t_c \tag{10}$$

Notice that since t_c is a function of the block size B, t_s depends on B^2.

The number of read cycles was the following.

(a) For scrubbing: one read cycle every t_s units of time for each memory word. Therefore the number of read cycles per unit of time, $\mathrm{RC}_{Scrubbing}$, can be computed as

$$RC_{Scrubbing} = \frac{M \cdot B}{t_s}. \tag{11}$$

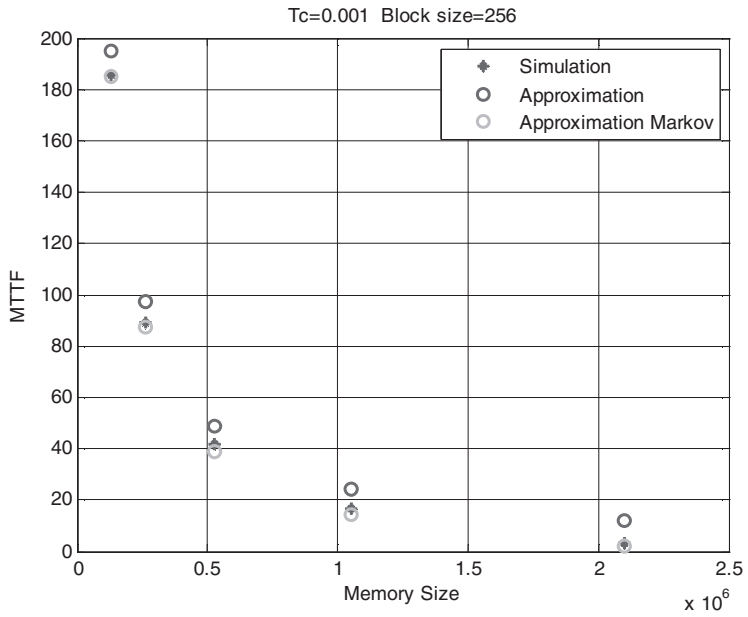

Fig. 9. MTTF simulation results and model estimates for the third experiment with block size 256.

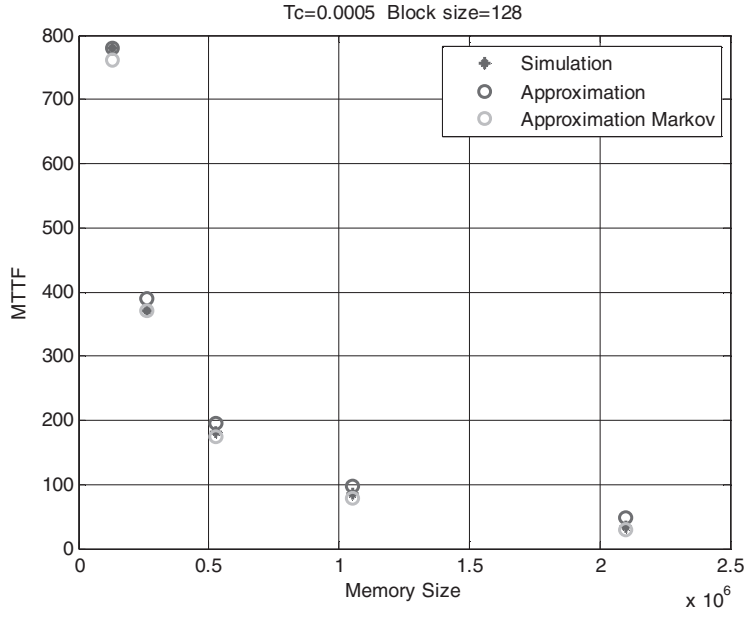

Fig. 10. MTTF simulation results and model estimates for the third experiment with block size 128.

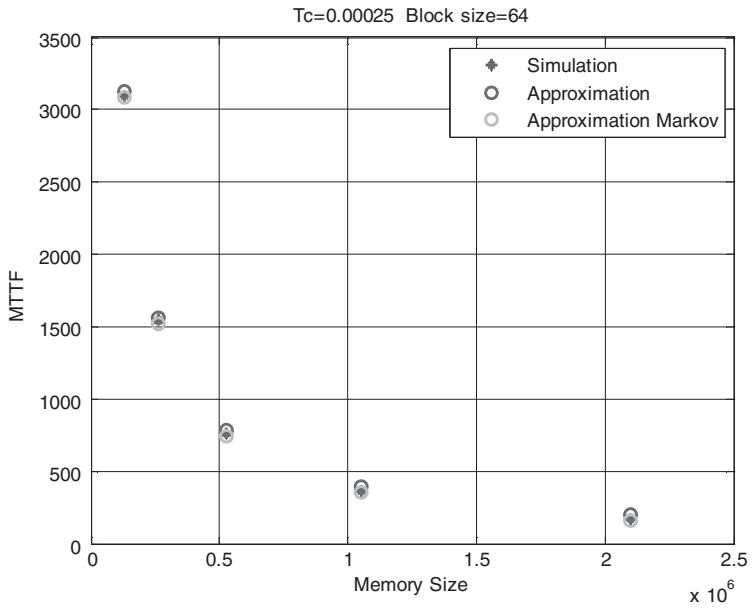

Fig. 11. MTTF simulation results and model estimates for the third with block size 64.

(b) For BICS: on average $B/2$, each time an error arrives with rate $\lambda \cdot M$. Therefore the number of read cycles per unit of time, RC_{BICS}, can be computed as

$$RC_{BICS} = \lambda \cdot M \cdot \frac{B}{2}. \tag{12}$$

Combining Eqs. (11) and (12), the ratio of read cycles is

$$r = \frac{RC_{Scrubbing}}{RC_{BICS}} = \frac{\frac{M \cdot B}{t_s}}{\lambda \cdot M \cdot \frac{B}{2}} = \frac{2}{\lambda \cdot t_s}. \tag{13}$$

If Eq. (10) is applied to ensure both implementations have the same reliability then the read cycle ratio becomes

$$r = \frac{2}{\lambda \cdot t_s} = \frac{1}{\lambda \cdot B \cdot t_c}. \tag{14}$$

Since it was assumed that $\lambda \cdot \mathrm{M} \cdot \mathrm{t}_c \ll 1$, then (as in most cases $M > B$):

$$r = \frac{RC_{Scrubbing}}{RC_{BICS}} = \frac{2}{\lambda \cdot t_s} = \frac{1}{\lambda \cdot B \cdot t_c} \gg 1 \tag{15}$$

and therefore,

$$RC_{Scrubbing} \gg RC_{BICS}. \tag{16}$$

This means that the approach based on BICS is effective in terms of reducing the number of read cycles used for protection, leaving more bandwidth for data operations and consuming less power.

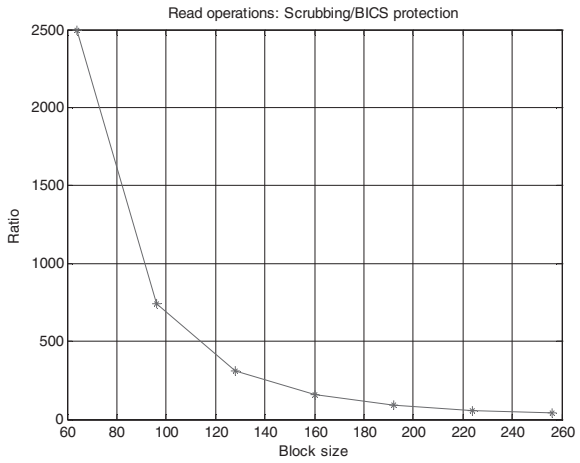

Fig. 12. Ratio of the read cycles in scrubbing compared to BICS.

In Figure 12, the ratio r is depicted for several values of the block size B. It can be seen that, for the BICS approach, the smaller the block size is, the fewer read cycles are required since less words in the faulty block need to be checked in order to identify the error.

It can be noted in Eq. (15) that both t_c and λ depend on B: the larger the size block is, the more time is needed to correct it (t_c), and the higher error arrival rate per block (λ). This, together with the explicit dependence on B, makes r have a growth order of $O(1/B^3)$. This may seem to imply that a smaller block size would always be beneficial, since the number of read cycles used by the BICS approach would be reduced. However, this is a simplistic conclusion because a smaller block size would necessitate a larger number of blocks and of BICS. This would, in turn, increase the area of the circuit and the power consumed by the BICS themselves. In fact, for large values of r, the BICS power consumption may be the more relevant factor, as the correction power consumption is negligible compared to that of scrubbing.

5. CONCLUSIONS

The reliability of memories protected with BICS and a per-word parity bit was analyzed in this article. Two models were presented which enable quick evaluation of the MTTF. These models allow designers to select the optimal configuration to meet a given reliability level.

The models were validated using a wide set of simulation experiments that illustrate their applicability. The influence of the size of the memory blocks that share BICS on memory reliability was studied. Finally, a comparison of the number of read cycles required for a given MTTF level using scrubbing and BICS protection was conducted. The results show the potential savings of the BICS-based approach.

Further work will concentrate on the evaluation of more sophisticated correction algorithms and their impact on reliability.

REFERENCES

ARGYRIDES, C., VARGAS, F., MORAES, M., AND PRADHAN, D. K. 2008. Embedding current monitoring in h-tree RAM architecture for multiple seu tolerance and reliability improvement. In *Proceedings of the 14th IEEE International Online Testing Symposium*. 155–160.

BLAUM, M., GOODMAN, R., AND MCELIECE, R. 1988. The reliability of single-error protected computer memories. *IEEE Trans. Comput. 37*, 1, 114–119.

CALIN, T., VARGAS, F. L., AND NICOLAIDIS, M. 1995. Upset-Tolerant CMOS RAM 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.

GILL, B., NICOLAIDIS, M., AND PAPACHRISTOU, C. 2005a. Radiation induced single-word multiple-bit upsets correction in SRAM. In *Proceedings of the 11th 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 Conference on Design, Automation and Test in Europe*. 592–597.

GOODMAN, R. M. F. AND MCELIECE, R. J. 1982. Hamming codes, computer memories and the birthday surprise. In *Proceedings of the 20th Allerton Conference on Communication, Control and Complexity*. 672–679.

GOODMAN, R. M. AND SAYANO, M. 1991. The reliability of semiconductor RAM memories with on-chip error-correction coding. *IEEE Trans. Inform. Theory 37*, 3, 884–896.

GOSSETT, C. A., HUGHLOCK, B. W., KATOOZI, M., LARUE, G. S., AND WENDLER, S. A. 1993. Single event phenomena in atmospheric neutron environments. *IEEE Trans. Nucl. Sci. 40*, 1845–1856.

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 (IEDM'03). Tech. Digest,* 21.4.1.

MAVI, D. G. AND EATON, P. H. 2002. Soft error rate mitigation techniques for modern microcircuits. In *Proceedings of the 40th Annual Reliability Physics Symposium.* 216–225.

MAY, T. C. AND WOOD, M. H. 1978. A new physical mechanism for soft errors in dynamic memories. In *Proceedings of the 16th Annual International Reliability Physics Symposium*. 33–40.

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 18th Symposium on Integrated Circuits and Systems Design*. 62–67.

NICOLAIDIS, M. 2005. Design for soft error mitigation. *IEEE Trans. Device Mater. Reliabil. 5*, 3.

NORMAND, E. 1996. Single event upset at ground level. *IEEE Trans. Nucl. Sci. 43*, 2742–2750.

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 Mater. Reliabil. 7*, 4, 592–601.

REVIRIEGO, P. AND MAESTRO, J. A. 2009. Efficient error detection codes for multiple bit upset correction in SRAMs with BICS. *ACM Trans. Des. Autom. Electron. Syst. 14*, 1, 18:1.

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

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

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.

SCHRIMPF, R. D. AND FLEETWOOD, D. M. 2004. *Radiation Effects and Soft Errors in Integrated Circuits and Electronic Devices.* World Scientific Publishing.

TIPTON, A. 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.

TOSAKA, Y., EHARA, H., IGETA, M., UEMURA, T., OKA, MATSUOKA, H. N., AND HATANAKA, K. 2004. Comprehensive study of soft errors in advanced CMOS circuits with 90/130 nm technology. In *Proceedings of the IEEE International Electron Devices Meeting (IEDM'04). Tech. Digest,* 941.

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. In *Proceedings of the 24th International Symposium on Fault-Tolerant Computing*. 106–115.

YANG, G. C. 1995. Reliability of semiconductor RAMs with soft-error scrubbing techniques. *Proc. IEEE Comput. Digit. Techniq. 142*, 5, 337–344.

Received January 2009; revised October 2009; accepted November 2009