



<b>Title</b>	Reliability Analysis of Memories Protected with BICS and a per-Word Parity Bit
<b>Authors(s)</b>	Reviriego, P., Maestro, J.A., Bleakley, Chris J.
<b>Publication date</b>	2010-02
<b>Publication information</b>	Reviriego, P., J.A. Maestro, and Chris J. Bleakley. "Reliability Analysis of Memories Protected with BICS and a Per-Word Parity Bit." Association for Computing Machinery (ACM), February 2010. <a href="https://doi.org/10.1145/1698759.1698768">https://doi.org/10.1145/1698759.1698768</a> .
<b>Publisher</b>	Association for Computing Machinery (ACM)
<b>Item record/more information</b>	<a href="http://hdl.handle.net/10197/7063">http://hdl.handle.net/10197/7063</a>
<b>Publisher's statement</b>	© ACM, 2010. This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in ACM Transactions on Design Automation of Electronic Systems, {VOL 15, ISS 2, (2010)} <a href="http://doi.acm.org/10.1145/1698759.1698768">http://doi.acm.org/10.1145/1698759.1698768</a>
<b>Publisher's version (DOI)</b>	10.1145/1698759.1698768

Downloaded 2026-05-01 23:45:38

The UCD community has made this article openly available. Please share how this access benefits you. Your story matters! (@ucd\_oa)



© Some rights reserved. For more information

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

---

Pedro Reviriego, Juan Antonio Maestro and C.J. Bleakley

**This paper 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 paper allow fast estimation of MTTF enabling design of optimal memory configurations to meet specified MTTF goals at minimum cost.**

**Keywords: Radiation effects, single event upset, reliability, memory device, built-in current sensors.**

## I. Introduction

For many years, soft errors have been a major concern for circuits that operate in harsh environments, such as space [1]. Due to technology scaling, soft errors are also becoming an increasingly important factor in terrestrial applications [2],[3]. One type of soft error is the Single Event Upset (SEU) [4],[5],[6]. 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 [7],[8]. 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 [9],[10],[11]. More recent research focuses on the effects of Multiple Bit Upsets (MBUs) [12],[13],[14],[15] on memory reliability [16]. 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 [17],[18]. 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 [19]. 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 [20] 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 [21],[22] for SRAM protection. In these proposals, BICS were placed on the vertical 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 also been proposed

---

Sponsorship and author information come here after acceptance.  
This document is prepared in the regular or special-issue paper format.

as a means to deal with MBUs [23]. The implementation of efficient and reliable BICS has been addressed recently for advanced memory technologies (100nm) [24],[25]. These results are promising and indicate an increasing interest in BICS based memory protection.

As mentioned above, the reliability of memories protected with scrubbing and SEC codes has been widely studied in the literature. However, to the best of the authors' knowledge, there is no previous analysis of the reliability of memories protected with BICS. The purpose of this paper is to analyze the reliability of memories protected with BICS in terms of the MTTF.

The rest of the paper is organized as follows: in section II, the BICS-based memory architecture is described, providing the basis for the reliability models presented in section III. In section IV, the models are validated with an extensive set of simulations, illustrating their use in selecting the optimal memory configuration. Finally, in section V the conclusions of the work are presented.

## II. 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 Fig. 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 [24] the proposed size for a block is 256 words, so a large memory will have a large number of blocks.

Error! Not a valid link. [Fig. 1](#). Example of an SEU in a memory block protected with BICS and a per-word parity bit.

As detailed in [22], 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 a parity check to see if it has been affected by a 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 a 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 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 Fig. 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.

Error! Not a valid link. [Fig. 2](#). Example of two SEUs causing a failure in a memory block protected with BICS.

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. The effectiveness of this technique is assessed in section III. Also, the order of arrival of column errors could be recorded and used to identify errors in cases there the second error hits a word after the word has checked by the correction process but before the process for identifying the first error is complete. This modification would provide correction in a few cases but at the expense of increased complexity and correction time. Hence, it is 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.

## III. Reliability Models

Following previous memory reliability models [9],[10],[12], error arrivals are assumed herein to follow a Poisson process. The error arrival rate per memory block is denoted as  $\lambda$  and memories composed of  $M$  blocks of size  $B$  words are considered. Finally, the average correction time for an individual block is denoted as  $t_c$ .

### 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

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

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

$$P_a(0) \gg P_a(1) \gg P_a(2) \dots \quad (2)$$

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

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

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 (3) the Mean Events to Failure (METF) can be calculated as

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

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

$$MTTF_{BICS} = \frac{METF}{\lambda \cdot M} \cong \frac{1}{\lambda^2 \cdot M \cdot t_c} \quad (5)$$

which is similar to the traditional expression for memories protected with scrubbing [9]:

$$MTTF_{scrubbing} \cong \frac{2 \cdot B}{\lambda^2 \cdot M \cdot t_s} \quad (6)$$

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.

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.

## 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 Fig. 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.

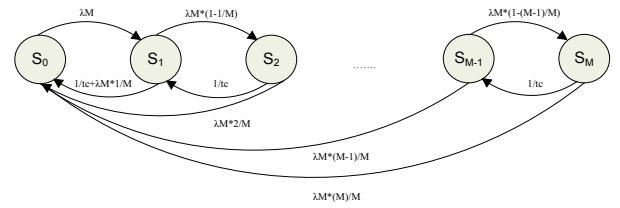


Fig. 3. Proposed Markov Model for the BICS protected memory.

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

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

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

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

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

## 3. Alternative correction algorithms

The correction process can be modified in order to handle some multiple error conditions. As mentioned previously, instead of ending the process once an error has been detected

and corrected, it may be beneficial to check the remainder of the block and handling any the errors that may be present in the same column (errors in other columns cannot be univocally detected). With this technique, several errors could be corrected in one process but the correction time increases (doubles on average), which could have negative effects.

The gain obtained from this technique can be assessed as follows. Let us assume a memory with  $N$  columns. As now an arrival interval  $t'_c = 2 \cdot t_c$  has to be considered, then the probability of failure given by (3) becomes

$$P'_f \cong P'_a(1) \cdot \frac{1}{M} \cdot \frac{N-1}{N} \cong 2 \cdot P_f \cdot \frac{N-1}{N} \quad (9)$$

Finally, from (5)

$$\begin{aligned} MTTF'_{BICS} &= \frac{METF'}{\lambda \cdot M} \cong \frac{1}{\lambda^2 \cdot M \cdot t'_c} \cdot \frac{N}{N-1} \\ &= \frac{MTTF}{2} \cdot \frac{N}{N-1} \end{aligned} \quad (10)$$

It can be observed that this alternative correction technique results in a lower MTTF. The conclusion from this analysis is that sophisticated error correction algorithms should be carefully analyzed before implementation to ensure that they really improve reliability.

### III. 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.

#### 1. Simple and Markov model validation

The first set of experiments was conducting using a per-block arrival rate  $\lambda$  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 Fig. 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 Fig. 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.

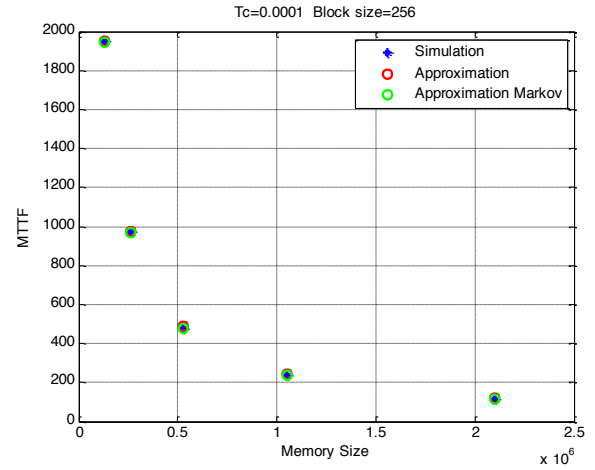


Fig. 4. MTTF simulation results and model estimates for the first experiment.

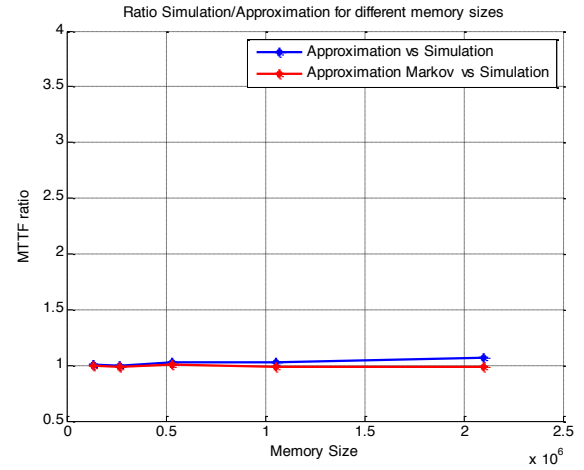


Fig. 5. MTTF ratio for model estimates and simulation results for the first experiment.

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 Fig. 6 and Fig. 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 the correction times are not exponentially distributed. Nevertheless, it provides a more reliable and more conservative estimate than the simple model.

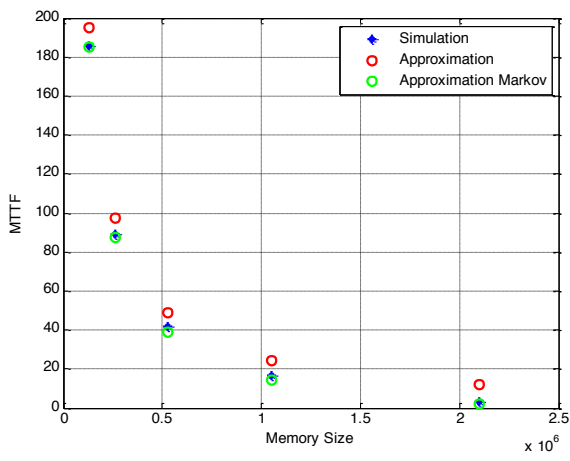


Fig. 6. MTTF simulation results and model estimates for the second experiment.

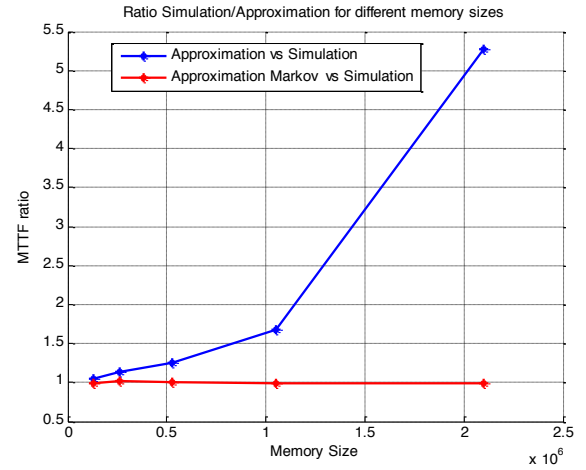


Fig. 8. MTTF ratio model estimates and simulation results for the second experiment with exponentially distributed correction time.

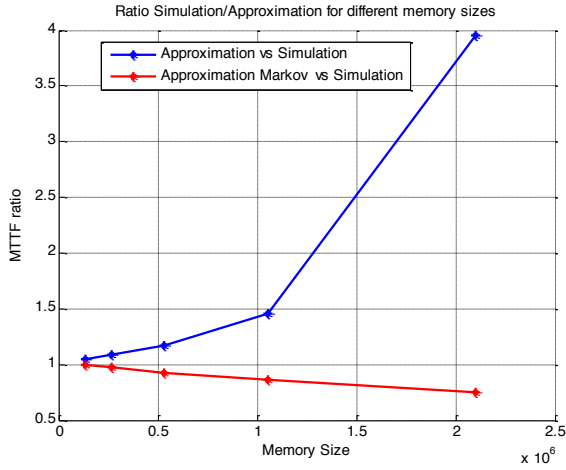


Fig. 7. MTTF ratio for model estimates and simulation results for the second experiment.

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 Fig. 8 where it can be seen that the Markov model estimates accurately match the simulation results.

## 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 equation (5) is rewritten using the per-word arrival rate  $\lambda'$ , the memory size in words  $S$  and correction time  $t_{c\_nom}$  normalized to a reference block size  $B_{nom}$ , we obtain:

$$MTTF \cong \frac{B_{nom}}{(\lambda')^2 \cdot B^2 \cdot S \cdot t_{c\_nom}} \quad (11)$$

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 Fig. 9, Fig. 10, and Fig. 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.

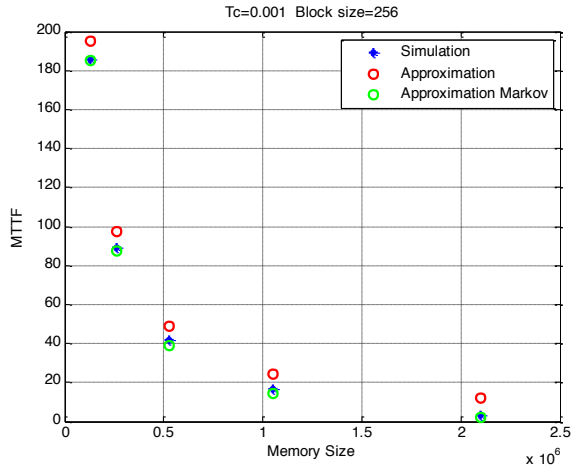


Fig. 9. MTTF simulation results and model approximations for the third experiment with block size 256.

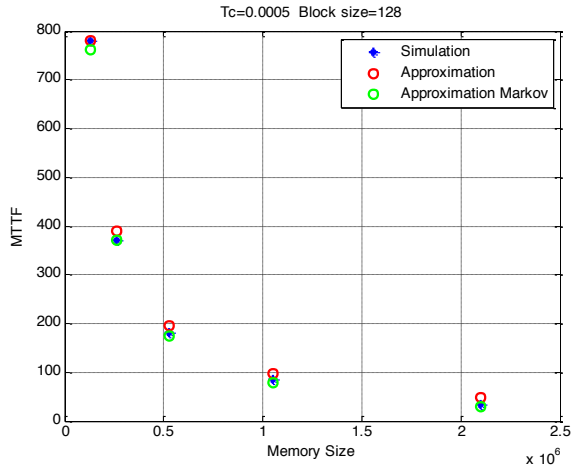


Fig. 10. MTTF simulation results and model approximations for the third experiment with block size 128.

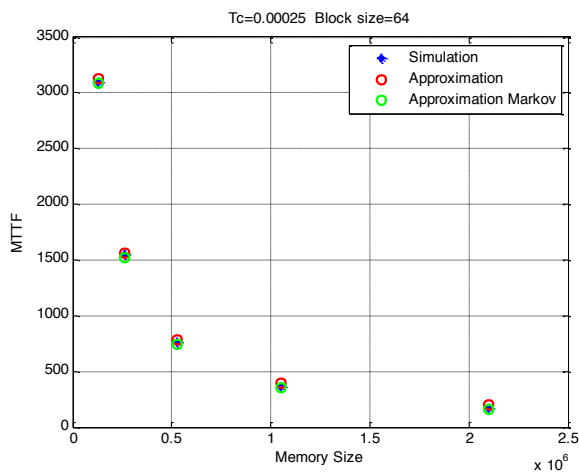


Fig. 11. MTTF simulation results and model approximations for the third with block size 64.

### 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 expressions (5) and (6) are equal. This leads to the following relation between  $t_s$  and  $t_c$ :

$$t_s = 2 \cdot B \cdot t_c \quad (12)$$

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 block. Therefore:

$$RC_{Scrubbing} = \frac{M \cdot B}{t_s} \quad (13)$$

b) For BICS: on average  $B/2$ , each time an error arrives with rate  $\lambda \cdot M$ :

$$RC_{BICS} = \lambda \cdot M \cdot \frac{B}{2} \quad (14)$$

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

$$r = \frac{RC_{Scrubbing}}{RC_{BICS}} = \frac{\frac{M \cdot B}{t_s}}{\lambda \cdot M \cdot \frac{B}{2}} = \frac{2}{\lambda \cdot t_s} \quad (15)$$

If expression (12) is applied to ensure both implementations have the same reliability then the read cycle ratio becomes:

$$r = \frac{2}{\lambda \cdot t_s} = \frac{1}{\lambda \cdot B \cdot t_c} \quad (16)$$

Since it was assumed that  $\lambda \cdot M \cdot t_c \ll 1$ , then (as in most cases  $M > B$ ):

$$r = \frac{RC_{Scrubbing}}{RC_{BICS}} = \frac{2}{\lambda \cdot t_s} = \frac{1}{\lambda \cdot B \cdot t_c} \gg 1 \quad (17)$$

And therefore,

$$RC_{Scrubbing} \gg RC_{BICS} \quad (18)$$

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.

In Fig. 12, the ratio  $r$  is depicted for several values of the block size  $B$ . It can be seen that the smaller the block size is, the less read cycles are required by the BICS approach since it has to check less words in the faulty block to identify the error.

It can be noted in expression (17) that both  $t_c$  and  $\lambda$  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 ( $\lambda$ ). 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.

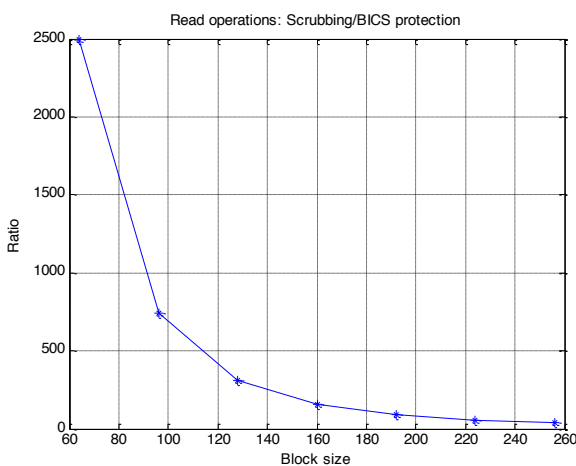


Fig. 12. Ratio of the read cycles in scrubbing compared to BICS.

## V. Conclusions

The reliability of memories protected with BICS and a per-word parity bit were analyzed in this paper. 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

- [1] C. A. Gossett, B. W. Hughlock, M. Katozzi, G. S. LaRue, and S. A. Wendler, "Single event phenomena in atmospheric neutron environments", IEEE Transactions on Nuclear Science, vol. 40, pp. 1845–1856, 1993.
- [2] E. Normand, "Single event upset at ground level," IEEE Transactions on Nuclear Science, vol. 43, 1996, pp. 2742–2750.
- [3] R. D. Schrimpf and D. M. Fleetwood, "Radiation effects and soft errors in integrated circuits and electronic devices", World Scientific Publishing, 2004. ISBN: 981-238-940-7.
- [4] M. Nicolaidis, "Design for Soft Error Mitigation", IEEE Transactions On Device And Materials Reliability, Vol. 5, No. 3, September 2005.
- [5] T. C. May and M. H. Wood, "A new physical mechanism for soft errors in dynamic memories," in Proc. 16th Annual Int. Reliability Physics Symp., 1978, pp. 33–40.
- [6] D. G. Mavi and P. H. Eaton, "Soft error rate mitigation techniques for modern microcircuits," in Proc. 40th Annual Reliability Physics Symp., 2002, pp. 216–225.
- [7] R.M.F. Goodman and R.J. McEliece, "Hamming codes, computer memories and the birthday surprise", Proc of the 20<sup>th</sup> Allerton Conference on Communication, Control and Complexity, Oct. 1982 Pages 672-679.
- [8] M. Blaum, R. Goodman and R. McEliece, "The reliability of single-error protected computer memories", IEEE Transactions on Computers, Volume 37, Issue 1, Jan. 1988, Pages: 114 – 119.
- [9] A.M. Saleh, JJ. Serrano and JH Patel, "Reliability of scrubbing recovery-techniques for memory systems", IEEE Transactions on Reliability Volume 39, Issue 1, April 1990, Pages: 114 - 122.
- [10] R.M. Goodman and M. Sayano, "The reliability of semiconductor RAM memories with on-chip error-correction coding", IEEE Transactions on Information Theory Volume 37, Issue 3, Part 2, May 1991 Pages: 884 - 896.
- [11] G.C. Yang, "Reliability of semiconductor RAMs with soft-error scrubbing techniques", IEE Proceedings Computers and Digital Techniques, Volume 142, Issue 5, Sept. 1995 Pages: 337 – 344.
- [12] D. Radaelli, H. Puchner, S.Wong and S. Daniel, "Investigation of multi-bit upsets in a 150 nm technology SRAM device", IEEE Transactions on Nuclear Science, Volume 52, Issue 6, Dec. 2005.
- [13] A. D. Tipton, J. A. Pellish, R. A. Reed, R. D. Schrimpf, R. A. Weller, M. H. Mendenhall, B. Sierawski, A. K. Sutton, R. M.

- Diestelhorst, G. Espinel, J. D. Cressler, P. W. Marshall and G. Vizkelethy, "Multiple-Bit Upset in 130 nm CMOS Technology" IEEE Transactions on Nuclear Science, Volume 53, Issue 6, Part 1, Dec. 2006 pp. 3259 – 3264.
- [14] J. Maiz, S. Hareland, K. Zhang and P. Armstrong, "Characterization of multi-bit soft error events in advanced SRAMs", IEEE International Electron Devices Meeting, 2003. IEDM'03 Technical Digest, Dec. 2003, pp. 21.4.1 - 21.4.4.
- [15] A.M. Chugg, M.J. Moutrie and R. Jones, "Broadening of the variance of the number of upsets in a read-cycle by MBUs", IEEE Transactions on Nuclear Science, Volume 51, Issue 6, Part 2, Dec. 2004, Pages: 3701 – 3707.
- [16] P. Reviriego, J.A. Maestro and C. Cervantes, "Reliability Analysis of Memories Suffering Multiple Bit Upsets", IEEE Transactions on Device and Materials Reliability, Vol. 7, No 4, December 2007, pp. 592-601.
- [17] S. Satoh, Y. Tosaka and S.A. Wender, "Geometric effect of multiple-bit soft errors induced by cosmic ray neutrons on DRAM's", IEEE Electron Device Letters, Volume 21, Issue 6, Jun 2000, Pages: 310 – 312.
- [18] Y. Tosaka, H. Ehara, M. Igeta, T. Uemura, H. Oka, N. Matsuoka and K. Hatanaka, "Comprehensive study of soft errors in advanced CMOS circuits with 90/130 nm technology", IEEE International Electron Devices Meeting, 2004. IEDM'04 Technical Digest, Pages: 941- 944.
- [19] A. Rubio, J. Figueras and J. Segura, "Quiescent current sensor circuits in digital VLSI CMOS testing" Electronics Letters Volume 26, 1990, pp.1204 – 1206.
- [20] F. Vargas, M. Nicolaidis and B. Courtois "Quiescent current monitoring to improve the reliability of electronic systems in space radiation environments" IEEE International Conference on Computer Design (ICCD), 1993, pp 596-600.
- [21] F. Vargas and M. Nicolaidis "SEU-tolerant SRAM design based on current monitoring" Twenty-Fourth International Symposium on Fault-Tolerant Computing, 1994, pp. 106 – 115.
- [22] T. Calin, F.L. Vargas, and M. Nicolaidis, "Upset-tolerant CMOS SRAM using current monitoring: prototype and test experiments" International Test Conference 1995, pp. 45 – 53.
- [23] B. Gill, M. Nicolaidis and C. Papachristou, "Radiation induced single-word multiple-bit upsets correction in SRAM" 11th IEEE International On-Line Testing Symposium, IOLTS, 2005, pp. 266 – 271.
- [24] B. Gill, M. Nicolaidis, F. Wolff, C. Papachristou and S. Garverick "An efficient BICS design for SEUs detection and correction in semiconductor memories" Design, Automation and Test in Europe, 2005, pp. 592 – 597.
- [25] E. H. Neto, I. Ribeiro, M. Vieira, G. With and F.L. Kastensmidt, "Evaluating Fault Coverage of Bulk Built-in Current Sensor for Soft Errors in Combinational and Sequential Logic" Integrated Circuits and Systems Design, 18th Symposium on, 2005 pp. 62 – 67.