Data Integrity Evaluations of Reed Solomon Codes for Storage Systems

G.C. Cardarilli, M. Ottavi, S. Pontarelli, M.Re, A. Salsano
{ottavi,pontarelli,salsano}@ing.uniroma2.it
{marco.re, g.cardarilli}@ieee.org
Department of Electronic Engineering
University of Rome “Tor Vergata”
Via Del Politecnico 1 00133 Rome, ITALY

Abstract

This paper introduces a very flexible approach for the evaluation of Bit Error Rate (BER) attainable on storage systems which use Reed Solomon codes. These evaluations are based on the use of a Markov model to evaluate the probabilities of having an uncorrectable codeword. Differently from previous literature, the reported approach can take into account the impact of both erasures and random errors allowing smaller degree of approximation and better evaluation of BER improvement related to the introduction of scrubbing techniques. The flexibility of the proposed method is finally shown by applying it to different cases of interest.

I. Introduction

In the past, magnetic tape recorders have been used to store the large amount of data generated by the on board data collection instruments of satellits. While a tape recorder can provide several gigabytes of non-volatile storage, its mechanical and electromechanical parts have insufficient operational flexibility and reliability for the space missions planned for the future. On the other hand, the rapid growth in capacity of semiconductor memory devices, quadrupling every three years, now permits the development of Solid State Mass Memories (SSMM), which are competitive with tape recorders due to their higher reliability and better performances [1]. It must be noticed that the requirements of low latency time, high throughput and storage capabilities of a SSMM can not be satisfied by space qualified components, therefore the use of Commercial Off The Shelf (COTS) components is mandatory. COTS components offer better performances in terms of storage capabilities, speed and power consumption with respect to the space qualified ones. However, the drawback of these components is the lack of resistance to both transient and permanent faults which can occur in the space environment. To cope with the occurrence of these faults many techniques have been developed. In particular, memory elements are usually strengthened with a combined usage of error detection and correction codes (EDAC) [2], spare elements and scrubbing techniques [3][4].

EDAC codes improve both the reliability (i.e. the probability that the system will perform its required function for a specified period of time) of the storage system and the data integrity (i.e. the probability that the data in the memory system are stored correctly for a specified period of time); in this paper a class of maximum distance separable EDAC codes known as Reed-Solomon codes is examined. RS codes guarantee a high level of data integrity, are widely used both for transmission lines and storage system and give a high level of flexibility allowing to choose the appropriate dataword and codeword lengths.

The use of spare elements improves the reliability of the memory allowing to cope with the occurrence of permanent faults with an ideal zero delay substitution with the spare element. Finally, the scrubbing technique [3] which basically consists in periodically reading the content of the memory and correcting the possible errors, is useful to improve the data integrity reducing the accumulation of SEUs in the memory. Summarizing, the reliability of a memory system using these techniques is closely related to the occurrence of permanent faults (e.g. stuck-at), while the data integrity is mainly related to the occurrence

Proceedings of the 19th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'04)
1550-5774/04 $ 20.00 IEEE

of transient faults (e.g. SEU), which can modify the value of the data stored in the memory elements, even if also partially to the occurrence of permanent faults. In fact, the occurrence of a permanent fault has a twofold impact on data integrity:

- the fault can cause the loss of data stored in the memory element affected by the fault,
- the correction capabilities of the used EDAC code are degraded because the code must correct the errors due to both permanent and transient faults.

The goal of this paper is to evaluate the data integrity of a memory system with respect to various design parameters such as the scrubbing frequency, the correction capabilities of the chosen EDAC code (related to the dataword and codeword length), and the rate of occurrence of permanent and transient faults with a very flexible Markov modeling which accounts for more cases of those in presented in the literature [5].

The paper is organized as follows: Section II provides the basics of RS codes, while Section III illustrates the method used to model the memory system for data integrity estimation. Section IV describes the evaluation results obtained for a set of typical satellite application, and finally in Section V some conclusions are drawn.

II. Background

Reed Solomon codes [2] are widely used both for data transmission and storage systems. A Reed Solomon code RS(n,k) is defined by the two integer values n and k where n represents the number of symbols of m bit (with $n \leq 2^m - 1$) composing a codeword and k represents the number of symbols composing the related dataword. A RS(n,k) code is able to correct up to $2 \cdot er + re \leq n - k$ where er is the number of erasures, re is the number of random errors. For data transmission a random error occurs when a symbol of the received codeword differs form the transmitted one in an unknown location of the codeword. Instead, an erasure occurs when the channel side informations available from the receiver allows to localize the erroneous symbol in the codeword. For a memory system the following assumptions can be done:

- Transient faults (e.g. SEU) can occur in a unknown location of a codeword, therefore they can be seen as random errors.
- Permanent faults (e.g. stuck-at 0/1) can be easily localized in a memory system either with self-checking circuits or with on-line testing, therefore can be assumed as erasures.

The localization of permanent faults is mandatory to exploit the error correction capabilities of the RS codes. In fact, until the permanent fault is not localized, the error correction algorithm assumes this error as a random error degrading the error correction capabilities of the code. Instead, when the permanent fault is localized the capabilities of the RS code of correcting an error which occurs in a known location can be fully exploited. The localization of permanent faults can be reached by using different methods, although a complete survey of these methods is out of the scope of this paper. Such an example a permanent fault in a memory chip can be detected monitoring the quiescent supply current (I_{ddq}) [6] or starting a fault localization procedure when the RS decoder detects an erroneous codeword. A localization procedure consists in rewriting the correct codeword in the same location and reading it back, if an error is detected again in the same location the location is assumed to be affected by a permanent fault (i.e. an erasure).

RS codes are suitable for highly reliable memory systems also for their reconfiguration capabilities. For example, if a memory system is initially configured with RS(18,16) code and a permanent failure occurs in a memory package, the code can not correct any random error. However, if the memory system is reconfigured to the usage of RS codes with higher number of symbols, the re correction capability can be recovered. For instance the memory can be reconfigured to a RS(144,128) code which is able to correct 8 erased symbols er and is also able to correct up to 4 random symbol errors re. Otherwise, if a fixed RS code is used, the memory module will be in permanent fault and can be substituted with a spare module.

As an example the reconfiguration from a RS (36,32) to the RS (144,128) is reported. Starting from a RS(36,32) coding scheme, after a certain period of time the check procedure detects three permanent package failures (three erasures). All the data stored in the memory module are converted from RS(36,32) to RS(144,128): i.e. four 36 byte codewords are read, decoded and the 128 data bytes are coded into a codeword of 144 bytes preserving the data stored in the memory. The drawback in the use of longer codewords is the performance degradation of the memory system in terms of latency. In fact, the decoding latency depends on the codeword length.

Finally, as for the random errors, to cope with their accumulation in a codeword a technique known as scrubbing can be applied [3]. Memory scrubbing basically consists in periodically reading a codeword, correcting the possible erroneous symbols and rewriting the corrected codeword in the same memory location, improving the data integrity of the memory. This technique has three main drawbacks:

1) Hardware overhead due to the logic circuitry needed to perform the operation
2) Increase of the average memory access time with the increase of scrubbing frequency
3) Increase of power consumption with the increase of scrubbing frequency.

The second and third drawbacks provide the reasons for which the scrubbing effect should be very accurately evaluated with respect to requested memory performances. A correct balance of scrubbing frequency to the application specifications allows reducing the impact of excessive scrubbing frequency.

The parameters needed to evaluate the data integrity of a memory system based on RS codes can be extracted starting from the architectural issues briefly described above. These evaluations are useful to tune the different project parameters in order to obtain the level of performances required by the application and can be performed using a model based on the Markov chains, as described in the next section.

III. Markov Modeling

The behavior of a RS coded memory system implementing the scrubbing technique can be described with a Markov model [7]. A Markov model is particularly suitable for obtaining data integrity evaluations of this dynamically reconfigurable system in terms of BER at a given time T. The states $S(er, re)$ of the Continuous Time Markov Chain (CTMC) which can be associated to the system temporal evolution, can be uniquely identified with the numbers er and re, which represent the number of random errors and erasures in different symbols of a codeword at time T. We introduce the start state at $T = 0$ (or Good state) $G = S(0, 0)$ in which $er = re = 0$ and the unrecoverable error state (or Fail state) $F = S(er, re)$ as a state in which we have $2 \cdot er + re > n - k$. The transition between the states represents the rate of occurrence of transient and/or permanent faults, or the occurrence of a scrubbing operation. Considering λ as the SEU rate affecting a single bit of a symbol, λ_e as the permanent fault rate per symbol and T_{sc} as the scrubbing operation period, the following transition rates can be defined:

- $r_{re}(er, re) = m \cdot \lambda \cdot (n - er - re)$ is the transition rate from the state $S(er, re)$ with er erasures and re random errors to the state $S(er, re + 1)$ with er erasures and $re + 1$ random errors.
- $r_{er}(er, re) = \lambda_e \cdot (n - er - re)$ is the transition rate from the state $S(er, re)$ with er erasures and re random errors to the state $S(er + 1, re)$ with $er + 1$ erasures and re random errors.
- $r_{er*}(re) = \lambda_e \cdot (re)$ is the transition rate from the state $S(er, re)$ with er erasures and re random errors to the state $S(er + 1, re - 1)$ with $er + 1$ erasures and $re - 1$ random errors. In this case the permanent fault affects a symbol previously affected by a random error.
- $r_{sc} = \frac{1}{T_{sc}}$ is the transition rate from the state $S(er, re)$ with er erasures and re random errors to the state $S(er, 0)$ with er erasures and 0 random errors. In this case the scrubbing operation can rewrite correctly all the symbols in the codeword affected by the random errors, but cannot rewrite the correct values in the symbols affected by the erasures as the erasures represent permanent faults.

In the previous definitions it has been assumed that the probability that a bit flip affects an already affected symbol is negligible and so it can be omitted. Starting from these assumptions and from the definitions of the rates, a Markov chain as reported in Fig. 1 can be drawn.

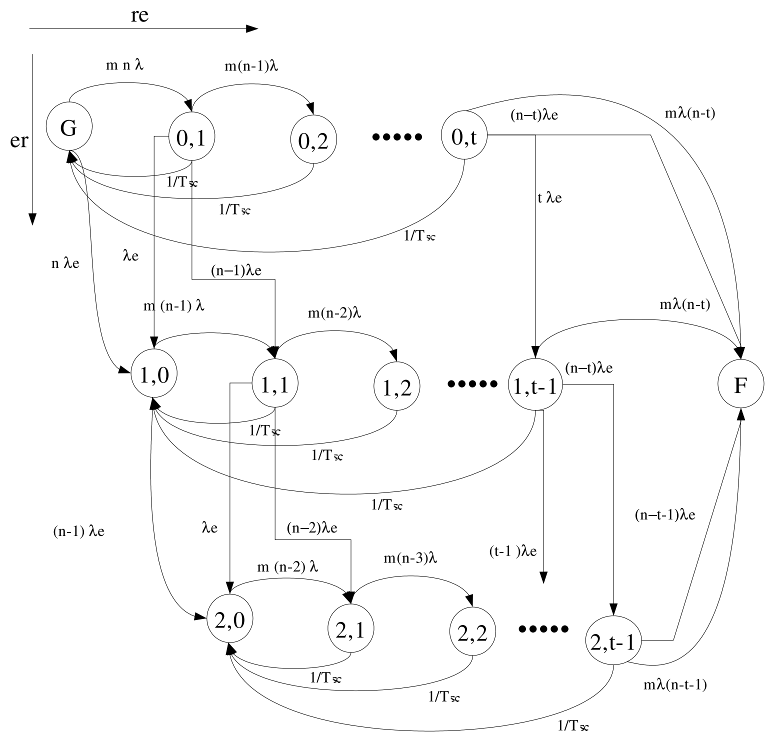

Fig. 1

MARKOV MODEL OF A RS CODE

In figure 1 we use an RS(n,k) code able to correct up to t random errors and a fixed T$_{SC}$ period. Because of the correction capability of the code as long as the code is correctable the number of erroneous bits is 0. Instead, when the codeword is uncorrectable, the number of wrong bits can be roughly assumed as the number of bits composing $n-k$ symbols which is the Hamming distance between two codewords. Thus the BER of the memory system can be defined as the probability of the codeword of being uncorrectable, i.e. the probability of being in the F state of the Markov chain $P(F)$, times the number of bits that are composing the Hamming distance with the closest symbol. Therefore:

$$BER = m \cdot (n-k) \cdot P(F) \tag{1}$$

The value of $P(F)$ can be obtained from the solution of the Markov model. An n-state Markov model leads to a system of n-coupled differential equations. These equations can be represented with vector notation. If we order the state $S(er, re)$ from 0 (the G state) to n (the F state), and assume the vector $P(t) = [P_{S(0)}(t), P_{S(1)}(t), .., P_{S(n)}(t)]$, where $P_{S(i)}(t)$ is the probability of being in the state $S(i)$ at time t, the system of differential equations is given by $P'(t) = \mathbf{A}P(t)$, where the matrix $\mathbf{A}$ is composed of the transition rates given above. In particular, the generic element $a_{i,j}$ with $i \neq j$ represents the transition rate from state i to state j, while the element $a_{i,i}$ represents the rate related to the probability of permanence in the i-th state, i.e. it can be defined as $a_{i,i} = -\sum_{j \neq i} a_{i,j}$. The obtained solution is time dependent, thus the BER of the system at time t can be written as

$$BER(t) = m \cdot (n-k) \cdot P_{S(n)}(t) \tag{2}$$

IV. Analysis and Evaluations

Starting from equation (2) some evaluation of the BER obtained with different RS codes and architecture choices can be done. For the transient faults we assume that in interplanetary space a background rate of $7.3 \cdot 10^{-7}$ errors/bit/day can be considered, which occasionally increases up to $1.7 \cdot 10^{-5}$ errors/bit/day during solar flares. The rate of permanent faults depends on the reliability of the memory chips and can be evaluated using the models given in [1], [8]. For the evaluation reported in this section we can compute the element a$_{i,i}$ of the matrix **A** starting from the transition rate r_{re}, r_{er} r_{er*} r_{sc} given in the previous section. In particualr for the RS(36,32) code the matrix **A** is reported below:

$$\begin{pmatrix}
a_{1,1} & 0 & 0 & 0 & 0 & \frac{1}{T_{sc}} & 0 & 0 & \frac{1}{T_{sc}} & 0 \\
n\lambda_e & a_{2,2} & 0 & 0 & 0 & \lambda_e & \frac{1}{T_{sc}} & 0 & 0 & 0 \\
0 & (n-1)\lambda_e & a_{3,3} & 0 & 0 & 0 & \lambda_e & \frac{1}{T_{sc}} & 0 & 0 \\
0 & 0 & (n-2)\lambda_e & a_{4,4} & 0 & 0 & 0 & \lambda_e & 0 & 0 \\
0 & 0 & 0 & (n-3)\lambda_e & a_{5,5} & 0 & 0 & 0 & 0 & 0 \\
nm\lambda & 0 & 0 & 0 & 0 & a_{6,6} & 0 & 0 & 0 & 0 \\
0 & (n-1)m\lambda & 0 & 0 & 0 & (n-1)\lambda_e & a_{7,7} & 0 & 2\lambda_e & 0 \\
0 & 0 & (n-2)m\lambda & 0 & 0 & 0 & (n-2)\lambda_e & a_{8,8} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & (n-1)m\lambda & 0 & 0 & a_{9,9} & 0 \\
0 & 0 & 0 & (n-3)m\lambda & (n-4)(\lambda_e + m\lambda) & 0 & (n-2)m\lambda & (n-3)(\lambda_e + m\lambda) & (n-2)(\lambda_e + m\lambda) & 0
\end{pmatrix}$$

where $a_{i,i} = -\sum_{j \neq i} a_{i,j}$.

In the following three cases are studied

1) comparison of RS(18,16) and RS(36,32) in terms of BER(t) obtained with no scrubbing and variable SEU rate
2) comparison of RS(18,16) and RS(36,32) in terms of attainable BER(t) during solar flares with different T_{sc} periods
3) analysis of RS(36,32)in terms of BER(t) considering permanent faults occurrence

In the evaluation related to case 1) and 2) we assume that data are stored in the memory for 2 days ($T_{st} = 48h$) and therefore we estimate the BER during this interval.

The first evaluation has been done on the RS codes RS(18,16) and RS(36,32) without scrubbing, without permanent fault and with the rate of transient fault $\lambda \in [7.3 \cdot 10^{-7}, 1.7 \cdot 10^{-5}]$. In Figure 2 the BER of the two codes are reported.

The following consideration can be done using the obtained data:

1) with the minimum SEU rate the RS(36,32) code provides a BER smaller than 10^{-12} without scrubbing .
2) the use of RS(18,16) code without scrubbing should be avoided because its BER is up to 10^{-10} even with the minimum SEU rate.
3) the use of the RS(36,32) code without scrubbing can be considered if we assume that during the mission of the satellite the frequency of solar flares is negligible.

In Figure 3 the attention is focused on the behavior of the RS(36,32) and RS(18,16) codes during solar flares. The parameter λ is fixed to $1.7 \cdot 10^{-5}$, which is the maximum SEU rate, and the BER with different T_{sc} is evaluated in order to tune the scrubbing frequency with respect to the specific space environment. The result of the BER evaluations of the two codes are reported with different scrubbing frequencies varying from 1 up to 4 times every hour. It must be noticed that also in this case no permanent fault is assumed to affect the memory (i.e. $\lambda_e = 0$).

The reported data show that the RS(18,16) code can not guarantee a low BER (less than 10^{-12}) during the solar flares even if the scrubbing technique is applied. Therefore in this case the use of the RS(36,32) code is mandatory. Using s RS(36,32) code the scrubbing frequency necessary in order to keep the BER below 10^{-12} can be lower than 1 times per hour.

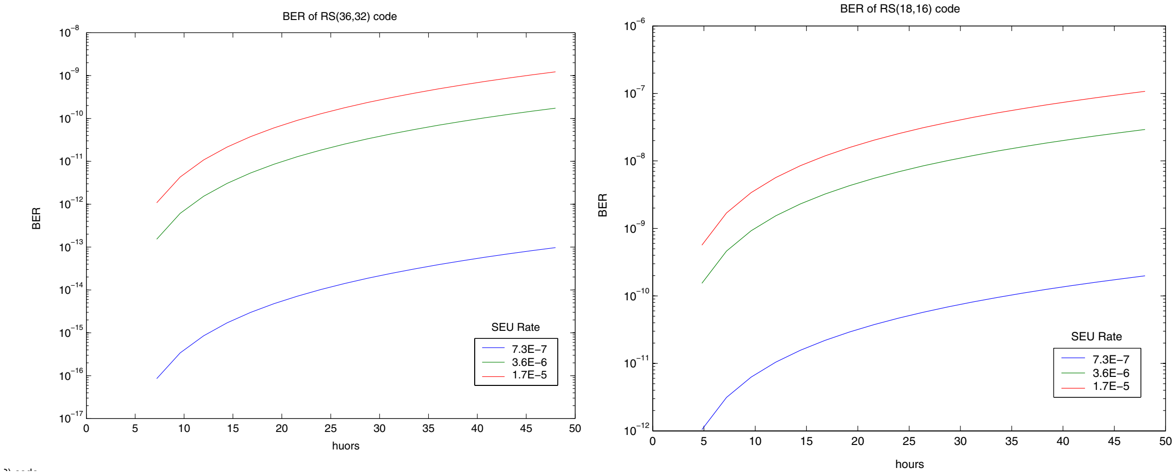

Fig. 2

BER OF RS(36,32) AND RS(18,16) CODES

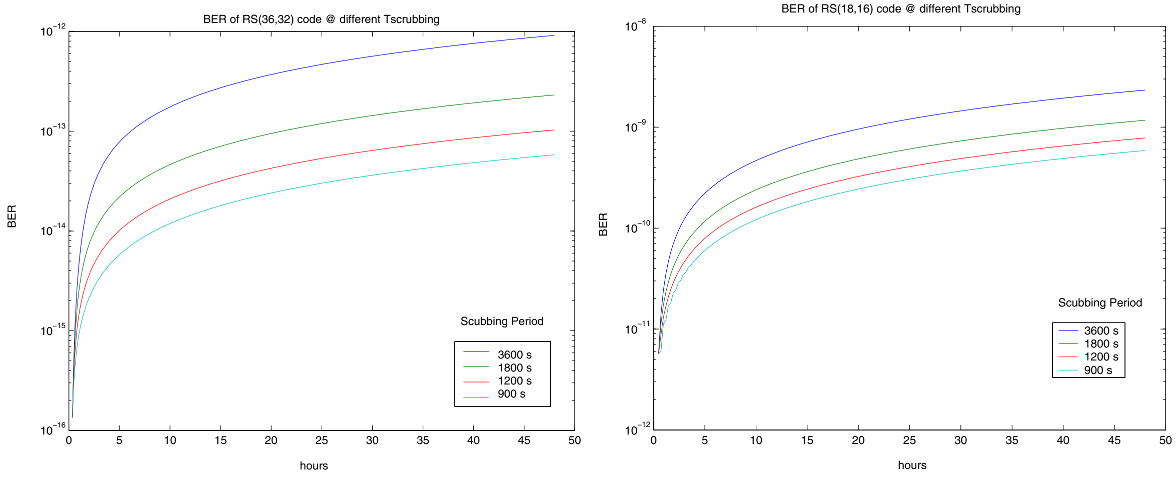

Fig. 3

BER OF RS(36,32) AND RS(18,16) CODES WITH DIFFERENT T$_{sc}$

The last evaluation reported takes into account also the behavior of the RS code if some permanent faults occur. In Figure 4 the result of the codes RS(36,32) with different permanent failure rates are reported. The scrubbing period is 1000 sec and the permanent failure rate is in the range $\lambda_e \in [1 \cdot 10^{-11}, 1 \cdot 10^{-4}]$. Note that in this case the considered data are assumed to be permanently stored into the memory for all the mission lifetime. Therefore the reported evaluations are done for a storage time period of about 24 months which equals the mission time. An evaluation of the attainable BER with a storage time $T_{st} = 48h$ like in the previous cases, could be done with suitable modifications of the Markov model in order to take into account the additional transitions related to $1/T_{st}$.

The above reported cases studied in this section can be seen as some examples of the evaluations that can be performed exploiting the high flexibility of the proposed method. In particular, the dependency

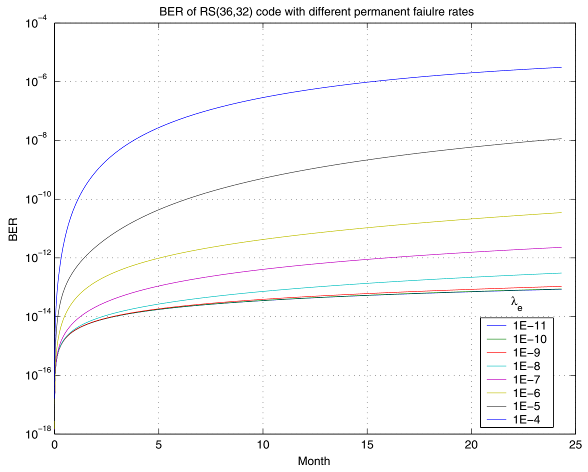

Fig. 4

BER OF RS(36,32) CODE WITH DIFFERENT λ_e

of the BER versus different parameters can be easily analyzed in order to select the suitable architecture solutions for the specific application.

V. Conclusions

A method to evaluate the BER of a memory system which uses Reed Solomon codes and scrubbing techniques has been proposed. The use of a Markov model allows evaluating the data integrity provided by a memory system with respect to a chosen RS(n,k) code and a set of both permanent and temporary faults. In particular the model takes into account the degradation of the data integrity caused by the occurrence of a permanent fault and allows to consider the time dependence of the BER. Moreover the the proposed model is suitable to evaluating the improvement of the BER obtained with various architectural choices (e.g. scrubbing techniques but also use of cold spare memory modules). Some evaluations obtained with the proposed method are reported in order to show its high flexibility which permits to use it to forecast the behavior of a memory system in different operating conditions.

References

[1] G.C. Cardarilli, A. Leandri, P. Marinucci, M. Ottavi, S. Pontarelli, M. Re, and A. Salsano, "Design of a fault tolerant solid state mass memory," *IEEE Transactions on Reliability*, vol. 52, no. 4, pp. 476 – 491, December 2003.

[2] R. E. Blahut, *Theory and Practice of Error Control Codes*, Addison-Wesley Publishing Company, 1983.

[3] A. M. Saleh, J.J. Serrano, and J.H. Patel, "Reliability of scrubbing recovery-techniques for memory systems," *IEEE Transactions on Reliability*, vol. 39, pp. 114 – 122, April 1990.

[4] G. C. Yang, "Reliability of semiconductor rams with soft-error scrubbing techniques," in *Computers and Digital Techniques IEE Proceedings*, September 1995, vol. 142, pp. 337 – 344.

[5] F. Labeau, C. Desset, B. Macq, and L. Vandendorpe, "Approximating the protection offered by a channel code in terms of bit error rate," in *European Signal Processing Conference*, Rhodes, Greece, 1999.

[6] J. M. Soden, "Iddq testing for submicron cmos ic technology qualification," in *IDDQ Testing, Digest of Papers, IEEE International Workshop on*, 5-6 Nov 1997, pp. 52 – 56.

[7] L. Schiano, M. Ottavi, and F. Lombardi, "Markov models of fault-tolerant memory systems under seu," in *IEEE International Workshop on Memory Technology, Design and Testing*, 2004.

[8] *MIL-HDBK 217*.