

Number of Events and Time to Failure Distributions for Error Correction Protected Memories

Pedro Reviriego, *Member, IEEE*, Lars Holst, and Juan Antonio Maestro, *Member, IEEE*

Abstract—Designing the memory system for an application has always been a complex task. This complexity has been increasing in the last years due the technology scaling, since memories are becoming more sensitive to soft errors. This issue is aggravated when the application is running in a harsh environment, e.g., in the presence of intense radiation sources. Therefore, the task to determine the optimal memory size and optimal protection mechanism is a challenge for designers to provide appropriate fault tolerance. In this paper, some classical reliability parameters, as the mean time to failure and the mean number of events to failure are revisited, but extending their scope with their probabilistic distributions. These can be of great help to designers when analyzing the consequences on reliability of certain decisions related to the memory size and protection codes.

Index Terms—Error correction codes, memory, reliability, soft errors, time to failure.

I. INTRODUCTION

RELIABILITY is a key concern for memory devices as data corruption can lead to system failure. Due to shrinking geometries and lower supply voltages, soft errors are becoming increasingly common in advanced memories [1]. Many of those errors are caused by radiation particles that hit the device and change the value of a memory cell [2]. To mitigate the effect of soft errors, a number of techniques ranging from the physical design of the device to system level techniques have been proposed [3]. In the case of memories, typically error correcting codes are used in each memory word so that when a soft error occurs, it can be corrected [4]. The most common type of codes used is single error correction–double error detection codes that can correct a single error per word and detect double errors to avoid silent data corruption [5]. More sophisticated codes that can perform double error correction (DEC) [6] or even triple error correction (TEC) have also been proposed [7]–[9]. These codes are often combined with scrubbing [7] to prevent error accumulation and increase reliability. Another major issue is multiple cell upsets (MCUs), where a single particle hit can upset more than one cell [1]. MCUs tend to affect cells that are physically close and can disturb several bits of the same word. To avoid this effect, interleaving is commonly used

in memories. With interleaving, cells that belong to the same word are physically distant in such a way that an MCU only affects one cell per word [10]. In that situation, MCUs appear as multiple single errors and the reliability of the memory can be accurately approximated by the single error case using an increased arrival rate (as shown in [10]). Therefore, in the rest of the paper, only single errors are considered, but the results also apply to a memory that suffers MCUs if an appropriate interleaving is used.

Memory reliability has been mostly studied in terms of the mean time to failure (MTTF) and mean number of events to failure (METF). Most of the existing work concentrates on SEC codes [5], [11]–[13] but some results have also been presented for codes that can correct multiple errors per word [9]. MTTF and METF are valuable indicators of memory reliability but cannot be used to accurately predict the probability of failure after a given time. For that, if the number of events and the time to failure are defined as a random variable, their distributions are needed.

The goal of this paper is to provide such distributions both in terms of number of events to failure and time to failure. To the best of the authors' knowledge, such distributions have not been presented anywhere else and its derivation is the main contribution of this paper.

These distributions can provide a value for the failure probability of the system after a certain time or after a certain number of events (bitflips) has occurred. These can then be used to extend existing reliability analysis and help the memory and system designers in the following.

- 1) Determine the optimal protection code that produces the lowest cost overhead while meeting the reliability constraints.
- 2) Calculate the largest memory that can be effectively protected with a certain protection code to meet the reliability constraints.
- 3) Deduce the maximum radiation level (through the event arrival rate) where a protected memory can work while meeting the reliability constraints.

In this paper, an exact analytical formulation of the distribution of the number of events to failure is introduced. Also, approximations for both the number of events and time to failure are proposed for the asymptotic case in which the number of words in the memory is large. These approximations are especially useful as memories are large in most designs.

The rest of the paper is organized as follows. In Section II the analytical solution and the different approximations for the distribution of the number of events and time to failure are

Manuscript received January 15, 2010; revised April 9, 2010; accepted June 11, 2010. Date of publication June 28, 2010; date of current version September 9, 2010. This work was supported by the Spanish Ministry of Science and Education under Grant AYA2009-13300-C03.

P. Reviriego and J. A. Maestro are with the Universidad Antonio de Nebrija, 28040 Madrid, Spain (e-mail: previrie@nebrija.es; jmaestro@nebrija.es).

L. Holst is with the Department of Mathematics, Royal Institute of Technology, 100 44 Stockholm, Sweden (e-mail: lholst@math.kth.se).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TDMR.2010.2055159

introduced. In that section some applications of the proposed approximations to memory design and selection are also discussed. Then in Section III simulation results are presented to evaluate the accuracy of the different approximations. This section also includes a discussion of the advantages and disadvantages of the different approximations. In Section IV the approximations are used in a few case studies to show how they can be applied to real designs. Finally, in Section V conclusions are presented. The mathematical derivations for the proposed solutions are covered in the Appendix.

II. EVENT AND TIME TO FAILURE DISTRIBUTIONS AND APPLICATIONS TO MEMORY DESIGN

Before introducing the proposed distributions, the assumptions used in the rest of the paper are summarized below:

- 1) The memory under study is formed by M words.
- 2) The memory is protected by a code capable of correcting L errors per word, so that a failure occurs when any given word has at least $L + 1$ errors.
- 3) Errors are produced at the memory following a Poisson process with error rate $\lambda * M$, where λ is the error rate per memory word and unit of time. This parameter also represents the mean value of the distribution.
- 4) Errors are uniformly distributed among the M cells.
- 5) T denotes the variable for the time to failure.
- 6) N denotes the variable for the number of events to failure.

In the rest of the section, the different approximations for the time to failure and the number of events to failure are introduced (as well as an exact solution for the latter). To keep the focus on the memory reliability problem, the mathematical derivations are presented at the end of the paper in the Appendix.

A. Time to Failure Distribution

The probability that a memory protected with a code capable of correcting L errors fails before t_x can be approximated when M is large as

$$P(T < t_x) \cong 1 - e^{-\frac{M}{(L+1)!} \cdot (\lambda \cdot t_x)^{L+1}}. \quad (1)$$

This will be denoted as the first approximation in the rest of the paper. The derivation of this approximation and the following ones are available at the end of the paper in the Appendix.

An alternative approximation that also applies for large M is given by

$$P(T < t_x) \cong 1 - e^{-M \cdot \left(1 - e^{-\lambda \cdot t_x} \cdot \sum_{k=0}^L \frac{(\lambda \cdot t_x)^k}{k!} \right)} \quad (2)$$

which will be denoted as the second approximation in the rest of the paper.

These two approximations enable a fast calculation of the probability of failure for any t_x .

Equations (1) and (2) can be used to guide the selection of the protection code (in terms of L) in a memory given M and the event arrival rate λ . This was previously done in [9] in terms of the MTTF, now it can be extended to cover the failure

probability for any given time interval t_x . This can be useful during the memory design phase to select a memory with the appropriate protection level for a given application.

Given L and λ , another application of (1), (2) is to determine the maximum value of M that would achieve a given reliability level. This can help a designer to dimension the memory of a system to ensure that reliability constraints are met.

Considering now that M and L are given then using (1), (2) the maximum λ that can be tolerated for a given reliability level can be determined. This is useful as it helps the designer to predict on which type of environments the memory can be used. For ground level systems this can translate into a maximum altitude while for space-borne systems it will dictate on which orbits the system can operate safely [7].

When memory scrubbing is used to improve reliability, a key parameter is the probability of failure in a scrubbing interval (t_s). Using (1) this probability becomes

$$P(T < t_s) \cong 1 - e^{-\frac{M}{(L+1)!} \cdot (\lambda \cdot t_s)^{L+1}} \quad (3)$$

or using (2)

$$P(T < t_s) \cong 1 - e^{-M \cdot \left(1 - e^{-\lambda \cdot t_s} \cdot \sum_{k=0}^L \frac{(\lambda \cdot t_s)^k}{k!} \right)}. \quad (4)$$

There is no need to use other approximations that are only applicable when there is a low number of events per scrubbing interval [5].

In summary, equations (1) and (2) prove to be very useful for memory reliability evaluation during the design stage of a memory or system.

B. Number of Events to Failure Distribution

The distribution of the number of events to failure can also be useful in different situations. For example during memory fabrication some cells may suffer defects that at the logical level appear as permanent failures [14]. These defects affect the production yield thus increasing the fabrication cost. Several techniques can be used to reduce the impact of those defects on yield. Examples are the use of fault repair techniques and error correction codes. When error correction codes are used it is interesting to know the probability of failure given a number of cells with defects. This would give information about the yield when error correction is applied.

To be able to select an appropriate error correction code for the memory, the distribution of the expected number of errors per part should be well characterized and stable in the manufacturing process. This is critical to ensure that the error correction code selected would provide a good yield.

Another situation where the number of events to failure would be useful is when the error arrivals do not follow a Poisson process but we have an indication of the number of expected errors after a given time (obtained for example from previous measurements in the same conditions). In this case the distribution of the number of events to failure would help us to accurately estimate the reliability of the memory.

The exact value of the probability of failure given N events can be obtained from the generating function of the random variable N that is given by

$$E(z^N) = 1 - (1 - z) \cdot \int_0^\infty e^{-v} \cdot \left(\sum_{j=0}^L \frac{\left(\frac{z}{M}\right)^j \cdot v^j}{j!} \right)^M \cdot dv \quad (5)$$

as from the definition of the generating function

$$E(z^N) = \sum_{k=1}^{\infty} z^k \cdot P(N = k). \quad (6)$$

And finally

$$P(N \leq n) = \sum_{k=1}^n P(N = k). \quad (7)$$

The details on how to compute $P(N = k)$ are given in the Appendix.

For large values of M two approximations can be used. The first one is related to (1) and given by

$$P(N > n) \cong e^{-\frac{n^{L+1}}{(L+1)! \cdot M^L}}. \quad (8)$$

While the second is related to (2) and given by

$$P(N > n) \cong e^{-M \cdot \left(1 - e^{-\frac{n}{M}} \cdot \sum_{k=0}^L \frac{\left(\frac{n}{M}\right)^k}{k!} \right)}. \quad (9)$$

In the next section the accuracy of the approximations presented is evaluated by simulation showing for which values of the parameters M and L are accurate.

III. NUMERICAL EVALUATION

A matlab script has been written to simulate the behavior of the memory and measure both the time to failure and number of events to failure. The script uses Poisson arrivals for the error events and a uniform distribution to select the word on which the error occurs, which is consistent with the assumptions made in Section II. The simulation is run until a new error occurs on a word that already has L errors, thus causing a failure. Then the time and number of events for the failure are recorded. A large number of simulations has been run to obtain a good approximation to the number of events and time to failure distributions.

This script has been used to validate the analytical results and check the accuracy of the derived approximations. The experiments have been divided into two groups, which work with regular and small memory sizes, respectively.

A. Regular Memory Sizes

In the first group of experiments, the approximation accuracy of the distributions of the time to failure [expressions (1) and (2)] and the number of events [expressions (8) and (9)] for regular values of M was checked. In the following, first approximation will be used to refer to expressions (1) and (8)

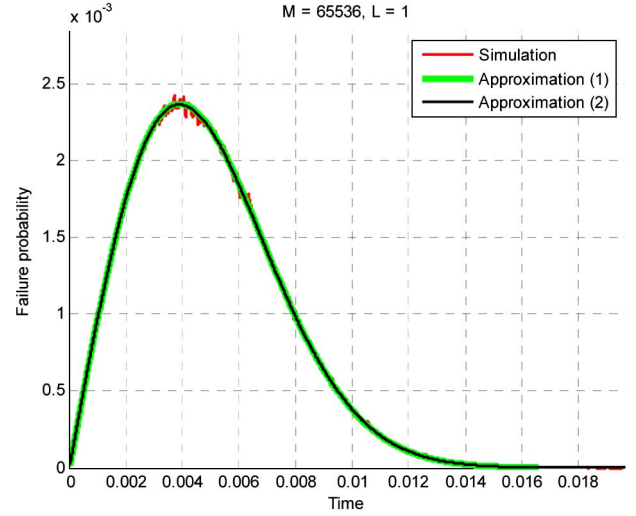


Fig. 1. Distribution of the time to failure obtained with approximations (1), (2) and simulation for $M = 64$ K and $L = 1$.

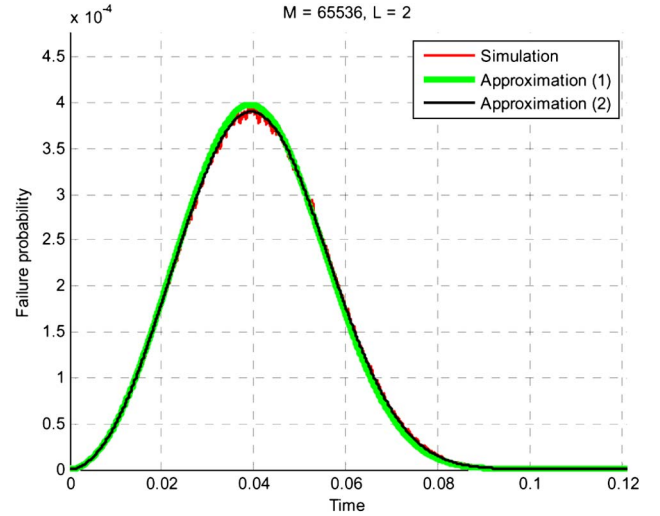


Fig. 2. Distribution of the time to failure obtained with approximations (1), (2) and simulation for $M = 64$ K and $L = 2$.

and second approximation to expressions (2) and (9). In these experiments, a memory size $M = 64$ K has been initially used. Since in most designs memories are larger than this size, and the accuracy of the approximations is better for larger values of M , the expected results can be considered a worst case. The results are shown in Figs. 1–3, for values of $L = 1, 2$ and 3 . An initial observation is that the accuracy of the first approximation gets worse with L . On the other hand, the second approximation is good, especially for L larger than one. To get accurate results by simulation, more experiments are required for larger memories, as there is more variability in the number of events to failure. For example, one million experiments were performed and it can be seen that for $L = 3$ the accuracy is not very good. To address this issue, the distribution obtained by simulation was smoothed by averaging nearby points. Additional experiments were done for $M = 1024$ K to evaluate the accuracy of the approximations for larger memories. The results, show a better matching, but there is still a small difference for the first approximation. This can be observed in Fig. 4 where the results

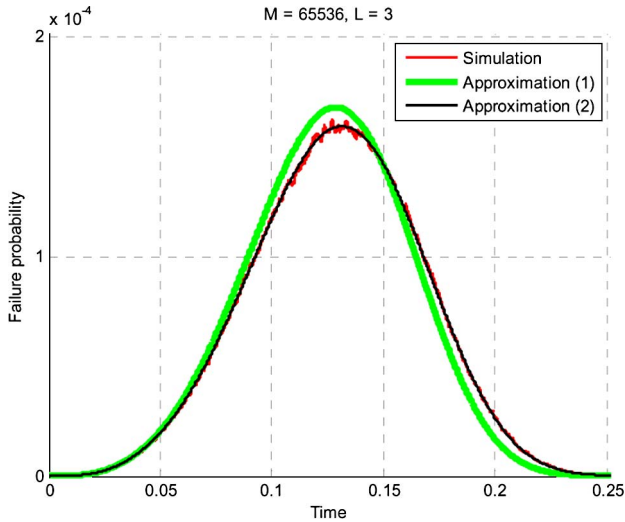


Fig. 3. Distribution of the time to failure obtained with approximations (1), (2) and simulation for $M = 64$ K and $L = 3$.

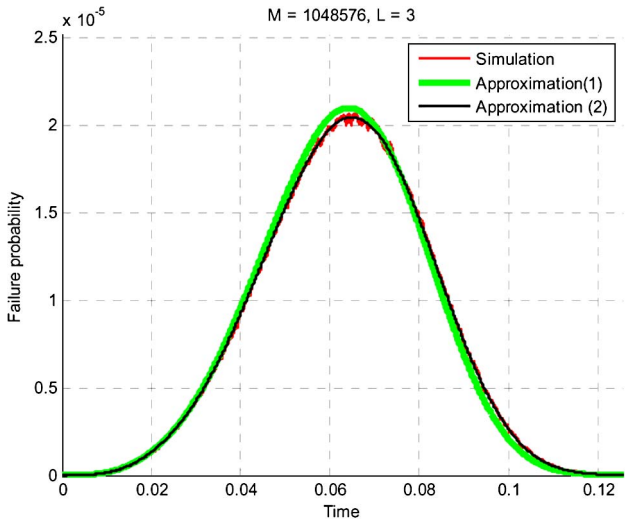


Fig. 4. Distribution of the time to failure obtained with approximations (1), (2) and simulation for $M = 1024$ K and $L = 3$.

for $L = 3$ are shown. In all cases the first approximation is conservative as observed in the previous experiment.

The approximations for the number of event distribution were also checked by simulation and the results were similar to those shown for the time to failure distributions.

B. Small Memory Sizes

The second group of experiments focuses on the theoretical distribution (5) of the number of events to failure. This can easily be computed for small memory sizes, and therefore the values $M = 64, 256$ and 1024 words were analyzed for different values of $L = 1, 2$ and 3 . Also, approximations (8) and (9), which have been previously studied in the first group of experiments, have been added, for comparison purpose. The results are shown in Figs. 5–7 for $M = 256$. In each case, the probability distribution is shown in the upper plot and the difference of the values obtained in simulation (and also using the two approximations) with the theoretical ones is illustrated

in the lower plot. They show that the theoretical distribution given by (5) accurately matches the simulation results. In fact, the two plots are practically identical in many of the figures. For these small values of M , approximation (8) is rather good for $L = 1$, but worse for $L > 1$. However, this approximation is conservative as it gives a lower number of events to failure than the one observed in simulation. This is useful, as it means that the use of the approximation would result in a safe design that exceeds the required reliability. Approximation (9) is more accurate when L is larger than 1, and therefore complements the first approximation. Similar results were obtained for other values of M (64 and 1024).

C. Simulation Summary

The conclusions from the simulation experiments are:

- The exact formulation of the number of events to failure given by (5) has been validated for different values of L and M .
- Expressions (1) and (8) can be used as an accurate approximation to the number of events and time to failure, respectively, when M is large and L is small.
- Approximations (1) and (8) give conservative estimates for the number of events and time to failure and therefore even when they are not very accurate, they can be safely used.
- Approximations (2) and (9) are very accurate when M is large and can be used when increased accuracy is required at the cost of increased complexity in the computation. These approximations complement (1) and (8), as they are accurate for larger values of L .

IV. APPLICATION OF THE PRESENTED APPROXIMATIONS TO THE SYSTEM DESIGN PROCESS

To illustrate the application of the proposed approximations in real designs, some examples are considered in this section.

In the first example, a memory has to be selected for on-board operation in a spacecraft. During the mission, the error arrival rate can vary significantly and a conservative design should use the worst case value for this arrival rate. Let us assume that the worst case conditions are known and correspond to an Equatorial Orbit [7]. The memory size M is fixed because of the application requirements. The design has to guarantee a low probability of failure (below 10^{-4}) for the duration of the mission. Consider, for the discussion, that M is assumed to be 512Mword (with a width of 32 bits), $\lambda_{\text{bit}} = 2 \cdot 10^{-8}$ errors per bit and day (this would correspond to a typical Equatorial Orbit (>3000 Km), as reported in [7]) and the mission duration is 5 years. Then, using (1) and (2), the probability of failure can be computed for different protection codes (L). The results are shown in Table I and it can be observed that a code stronger than SEC ($L = 1$) is needed. Specifically, a protection code of at least $L = 3$ would have to be provided by the memory designer. But this code is able to protect a larger memory than the initial 512Mword. Using the same expression, a memory of, for example, 1 Gword could be used, respecting the protection requirements stated before. In this way, the application designer has the option to enhance

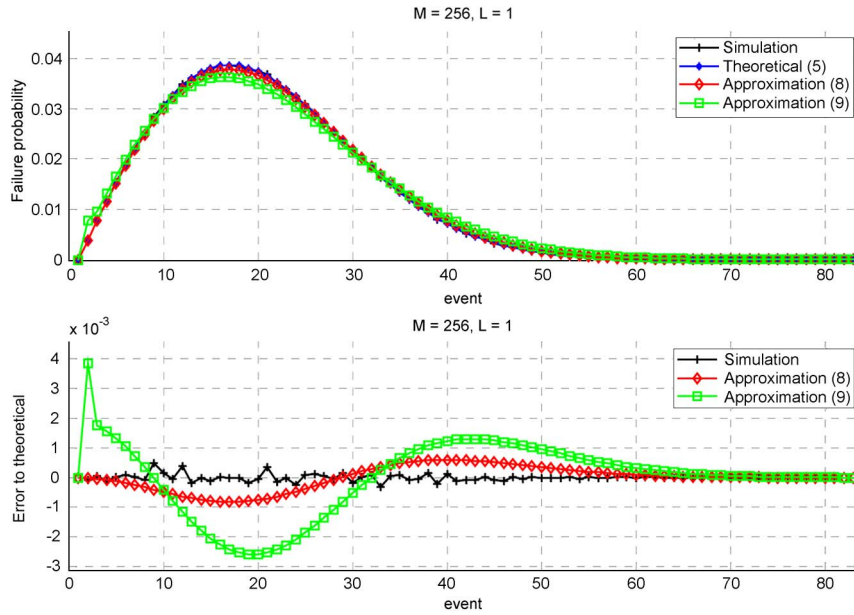


Fig. 5. Distribution of the number of events to failure obtained with the theoretical model (5), approximations (8) and (9) and simulation for $M = 256$ and $L = 1$.

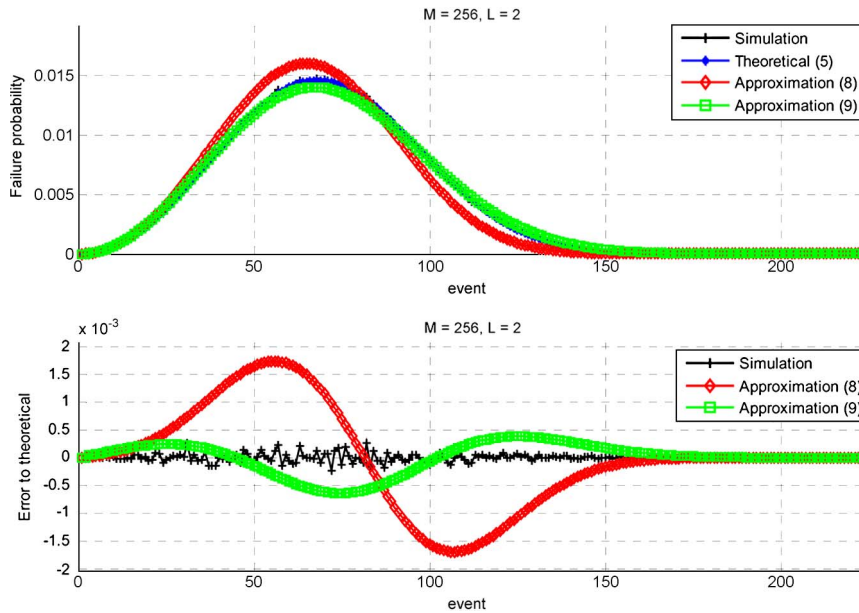


Fig. 6. Distribution of the number of events to failure obtained with the theoretical model (5), approximations (8) and (9) and simulation for $M = 256$ and $L = 2$.

the storage capability if needed. It should be noted that in this example a typical value of the arrival rate for the Equatorial Orbit has been used, but in a real design some safety margin should be added to ensure that the reliability objectives are met even if there are some deviations in the predicted error rate. This could be done by using a slightly larger rate in the calculations.

It is interesting to compare the results in Table I with the MTTF computed for the same cases using the existing expressions [9]. The MTTF values are shown in Table II. It is clear that we need the MTTF to be larger than the mission duration something that occurs for $L = 2$ and $L = 3$. However, the MTTF alone does not provide enough information to accurately predict the failure probability during the mission duration.

Another alternative would be to use scrubbing with SEC ($L = 1$). As an example, if scrubbing is performed once per hour, the probability of failure for the duration of the mission drops to approximately 0.008. A scrubbing period of less than a minute (approximately 40 s) is needed to meet the specified probability of failure. In this case (1) and (2) are used to calculate the probability of failure in a scrubbing period. From this, the probability of failure for the duration of a mission can be easily derived (e.g., [5]).

In this example, it has been shown that the expressions derived in this paper are useful for both the memory and the application design perspective. It can also be seen that both approximations provide very similar values for the probability of failure.

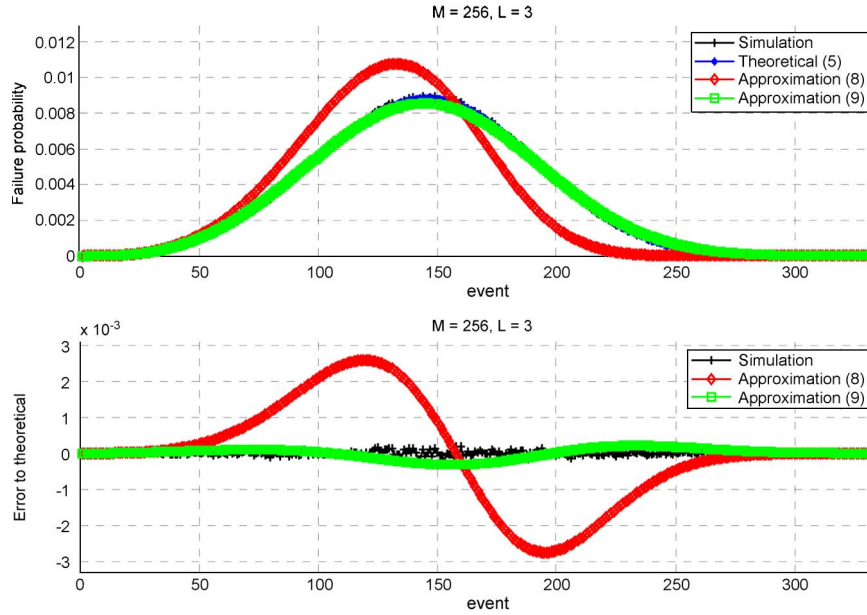


Fig. 7. Distribution of the number of events to failure obtained with the theoretical model (5), approximations (8) and (9) and simulation for $M = 256$ and $L = 3$.

TABLE I
MEMORY FAILURE PROBABILITIES FOR THE FIRST EXAMPLE

L	Probability of Failure (1)	Probability of Failure (2)
1	1	1
2	0.1329	0.1328
3	0.0000416	0.0000415

TABLE II
MTTFs FOR THE FIRST EXAMPLE

L	MTTF
1	0.23
2	8.55
3	56.42

In the second example, the reliability of a server that operates on a terrestrial environment is considered. In this case, $M = 4$ Gwords (with a width of 32 bits) and $L = 1$ (SEC is used). Then, considering that λ has been previously estimated, the probability that the server fails before a given time T can be computed using (1) or (2). This can be useful when designing computer clusters, as it allows the designer to know the degree of fault tolerance that will be required in advance [15]. For example, if an error rate $\lambda_{\text{bit}} = 2.4 \cdot 10^{-11}$ errors per bit per day is considered, then the probability of failure in years would be 0.000169 and other effects [15] will dominate the failure rate of servers. However, for larger memories the situation may be different. In this context, the proposed approximations could be used to predict the memory size at which failures caused by soft errors will become the dominant factor. In this particular case failures caused by soft errors would start to be significant only for very large memories in the range of terawords. Therefore, SEC seems to be sufficient unless the error rate increases significantly in the future. However, as the cost of scrubbing can be negligible in many cases, it may be interesting to implement it to provide an additional safety margin.

Finally, let us consider a situation in which the number of errors after a given time is known. This could correspond, for example, to the number of defects after fabrication. Let us assume a memory with 1Mwords that is protected with DEC and that has 1000 defective cells. Then, using (8) and (9), the probability of failure can be computed, giving values of $1.516 \cdot 10^{-4}$ and $1.515 \cdot 10^{-4}$, respectively. These values would ensure that the production yield is close to 100% even in the presence of that number of defective cells. However, if we consider an increase in the number of errors to 10 000 defective cells per part, the failure probability would rise to 0.141 and 0.140 according to (8) and (9). This level of defects would significantly affect the yield.

V. CONCLUSION

In this paper, the number of events and time to failure distributions of memories protected with error correcting codes have been studied. An exact formulation for the number of events to failure has been presented. For large memories, two approximations have been proposed which enable a fast computation of the reliability of a memory. These approximations can be applied to both the number of events to failure and the time to failure, and are useful as in most cases memories are large. The accuracy of the approximations has been studied through a number of simulation experiments that cover a wide range of values for memory size and protection code type. The first approximation is more accurate for memories protected with single error correction (SEC) codes, while the second is better for memories protected with double or triple error correction codes. It is also worth mentioning that the first approximation is conservative and therefore can be safely used to estimate the reliability in a design.

Also, a number of examples of applications of the approximations presented have been discussed. The examples illustrate the usefulness of the approximations in the memory design or

in the memory selection at the system level to ensure a given reliability target. In summary, the proposed approximations enable an accurate evaluation of the reliability of a memory.

APPENDIX

In this Appendix, the derivations for the approximations of the distributions of the number of events and time to failure and for the exact distribution of the number of events to failure introduced in Section II are presented.

To that end, let us define W_i as the random variable of the time to failure of memory word i . From the assumptions, errors arrive at each word following a Poisson process with arrival rate λ . Therefore, W_i is $\Gamma(L + 1, \lambda)$ distributed (where Γ is the Gamma distribution) and its probability density function is given by

$$P(W_i = t) = \frac{(\lambda \cdot t)^L \cdot e^{-\lambda \cdot t} \cdot \lambda}{L!} \quad (10)$$

$$T = \min(W_1, \dots, W_M). \quad (11)$$

Now, for a number $\mu_M > 0$

$$\begin{aligned} P(T > \mu_M) &= P(\min(W_1, \dots, W_M) > \mu_M) \\ &= \left(\int_{\mu_M}^{\infty} \frac{(\lambda \cdot t)^L}{L!} \cdot e^{-\lambda \cdot t} \cdot \lambda \cdot dt \right)^M \\ &= \left(1 - \int_0^{\mu_M} \frac{(\lambda \cdot t)^L}{L!} \cdot e^{-\lambda \cdot t} \cdot \lambda \cdot dt \right)^M. \end{aligned} \quad (12)$$

For a fixed $x > 0$ and $\mu_M = \sqrt[L+1]{(L+1)!/M} \cdot (x/\lambda)$, as M tends to infinity

$$\begin{aligned} \lim_{M \rightarrow \infty} \left(M \cdot \int_0^{\mu_M} \frac{(\lambda \cdot t)^L}{L!} \cdot e^{-\lambda \cdot t} \cdot \lambda \cdot dt \right) \\ = \lim_{M \rightarrow \infty} \left(M \cdot \int_0^{\mu_M} \frac{(\lambda \cdot t)^L}{L!} \cdot \lambda \cdot dt \right) \\ = x^{L+1}. \end{aligned} \quad (13)$$

Therefore, when M tends to infinity, and using the equality $\lim_{x \rightarrow \infty} (1 + (1/x))^x = e$, (12) converges to

$$\lim_{M \rightarrow \infty} P(T > \mu_M) = \lim_{M \rightarrow \infty} \left(1 - \frac{x^{L+1}}{M} \right)^M = e^{-x^{L+1}} \quad (14)$$

which is in fact approximation (1).

From (14), it follows that when M tends to infinity,

$$\lim_{M \rightarrow \infty} P(T < \mu_M) = 1 - e^{-x^{L+1}} \quad (15)$$

$$\lim_{M \rightarrow \infty} P\left(\lambda \cdot \sqrt[L+1]{\frac{M}{(L+1)!}} \cdot T = x\right) = (L+1) \cdot x^L \cdot e^{-x^{L+1}} \quad (16)$$

which gives the probability density function of the time failure of the memory.

Let us now define the random variables $Z_i/(\lambda M)$ as the times between the errors in N . Given the Poisson assumption, Z_i are exponentially distributed with mean 1 and the random variables Z_i and N are independent. Then, the time to failure T can be expressed as

$$T = \sum_{k=1}^N \frac{Z_k}{\lambda \cdot M} \quad (17)$$

from which

$$M \cdot T = \frac{N}{\lambda} \cdot \left(1 + \frac{\sum_{k=1}^N (Z_k - 1)}{N} \right). \quad (18)$$

When N tends to infinity the above equation can be written as

$$M \cdot T \cdot \lambda = N \quad (19)$$

which can be rewritten as

$$\lambda \cdot \sqrt[L+1]{\frac{M}{(L+1)!}} \cdot T = \frac{N}{\sqrt[L+1]{(L+1)!} \cdot M^L}. \quad (20)$$

Therefore from (16)

$$\lim_{M \rightarrow \infty} P\left(\frac{N}{\sqrt[L+1]{(L+1)!} \cdot M^L} = x\right) = (L+1) \cdot x^L \cdot e^{-x^{L+1}} \quad (21)$$

which is in fact approximation (8).

Let us now consider the probability that given N errors in the memory, k of them have fallen on a particular word. This is given by the binomial distribution

$$P_j(k) = \binom{n}{k} \frac{(M-1)^{n-k}}{M^n} \quad (22)$$

which, using the Poisson approximation to the binomial distribution [16], can be expressed as

$$P_j(k) \cong \frac{e^{-\alpha} \cdot \alpha^k}{k!} \quad (23)$$

where α is given by n/M .

Then assuming that the distributions of the errors on the different cells are independent (which is another approximation) the probability that given N events there is no word with more than k errors is given by

$$P(N > n) \cong \left(e^{-\frac{n}{M}} \cdot \sum_{k=0}^L \frac{\left(\frac{n}{M}\right)^k}{k!} \right)^M \quad (24)$$

or alternatively by

$$P(N > n) \cong \left(1 - e^{-\frac{n}{M}} \cdot \sum_{k=L+1}^{\infty} \frac{\left(\frac{n}{M}\right)^k}{k!} \right)^M. \quad (25)$$

Then, making

$$x = \frac{1}{e^{-\frac{n}{M}} \cdot \sum_{k=L+1}^{\infty} \frac{\left(\frac{n}{M}\right)^k}{k!}}. \quad (26)$$

The following is obtained:

$$P(N > n) \cong \left(\left(1 - \frac{1}{x}\right)^x \right)^{\frac{M}{x}}. \quad (27)$$

When M is large, if $1/x$ is not close to zero then $P(N > N)$ will be close to zero. On the other hand if $1/x$ is close to 0 then

$$\begin{aligned} P(N > n) &\cong \lim_{x \rightarrow \infty} \left(\left(1 - \frac{1}{x}\right)^x \right)^{\frac{M}{x}} \\ &= e^{-\frac{M}{x}} \\ &= e^{-M \cdot \left(e^{-\frac{n}{M}} \cdot \sum_{k=L+1}^{\infty} \frac{\left(\frac{n}{M}\right)^k}{k!} \right)} \\ &= e^{-M \cdot \left(1 - e^{-\frac{n}{M}} \cdot \sum_{k=0}^L \frac{\left(\frac{n}{M}\right)^k}{k!} \right)}. \end{aligned} \quad (28)$$

This is in fact approximation (9), which is valid for large values of M .

It is worth mentioning that obviously $P(N > n) = 1$ for $N = 1, \dots, L$ and for small values of M it may be worth using those values instead of the approximations.

When M is large, (28) can be used to estimate the distribution of the time to failure by approximating N with λMt as shown below

$$P(T > t_x) \cong P(N > \lambda \cdot M \cdot t_x) = e^{-M \cdot \left(1 - e^{-\lambda \cdot t_x} \cdot \sum_{k=0}^L \frac{(\lambda \cdot t_x)^k}{k!} \right)} \quad (29)$$

which is approximation (2).

The derivation of the exact probability density function of the number of events to failure starts by noting that

$$\begin{aligned} P(T > t) &= P(\min(W_1, \dots, W_M) > T) \\ &= P(W_1 > t) \cdot \dots \cdot P(W_M > t) \\ &= \left(\sum_{j=0}^L \frac{(\lambda \cdot t)^j}{j!} \cdot e^{-\lambda t} \right)^M. \end{aligned} \quad (30)$$

For an exponential random variable with mean 1

$$E(e^{-s \cdot Z}) = \int_0^{\infty} e^{-s \cdot t} \cdot P(Z=t) \cdot dt = \int_0^{\infty} e^{-s \cdot t} \cdot e^{-t} \cdot dt = \frac{1}{1+s}. \quad (31)$$

From (17)

$$e^{-s \cdot M \cdot T} = e^{-s \cdot \frac{\sum_{k=1}^N Z_k}{\lambda}}. \quad (32)$$

Taking the expectation in both sides of the equation

$$E(e^{-s \cdot M \cdot T}) = E \left(e^{-s \cdot \frac{\sum_{k=1}^N Z_k}{\lambda}} \right). \quad (33)$$

The right side of (33) depends on random variables Z_k and N which are independent and can be solved first for the random variables Z_k using (31)

$$E \left(e^{-s \cdot \frac{\sum_{k=1}^N Z_k}{\lambda}} \right) = E \left(\left(\frac{1}{1 + \frac{s}{\lambda}} \right)^N \right). \quad (34)$$

Also for the left size of (33), using the product rule for the derivative of the product of two functions, we obtain

$$\begin{aligned} E(e^{-s \cdot M \cdot T}) &= \int_0^{\infty} e^{-s \cdot M \cdot t} \cdot \frac{\partial P(T < t)}{\partial t} \cdot dt \\ &= \int_0^{\infty} e^{-s \cdot M \cdot t} \cdot -\frac{\partial P(T > t)}{\partial t} \cdot dt \\ &= 1 - \int_0^{\infty} s \cdot M \cdot e^{-s \cdot M \cdot t} \cdot P(T > t) \cdot dt \end{aligned} \quad (35)$$

which, using (30), can be reduced to

$$E(e^{-s \cdot M \cdot T}) = 1 - \int_0^{\infty} s \cdot M \cdot e^{-s \cdot M \cdot t} \cdot \left(\sum_{j=0}^L \frac{(\lambda \cdot t)^j}{j!} \cdot e^{-\lambda t} \right)^M \cdot dt. \quad (36)$$

Therefore

$$E \left(\left(\frac{1}{1 + \frac{s}{\lambda}} \right)^N \right) = 1 - \int_0^{\infty} s \cdot M \cdot e^{-s \cdot M \cdot t} \cdot \left(\sum_{j=0}^L \frac{(\lambda \cdot t)^j}{j!} \cdot e^{-\lambda t} \right)^M \cdot dt. \quad (37)$$

Also making $u = \lambda Mt$

$$E \left(\left(\frac{1}{1 + \frac{s}{\lambda}} \right)^N \right) = 1 - \frac{1}{\lambda} \cdot \int_0^{\infty} s \cdot e^{-(1 + \frac{s}{\lambda}) \cdot u} \cdot \left(\sum_{j=0}^L \frac{\left(\frac{u}{M}\right)^j}{j!} \right)^M \cdot du. \quad (38)$$

Finally, changing to $z = 1/(1 + s/\lambda)$ and $v = u/z$

$$E(z^N) = 1 - (1 - z) \cdot \int_0^\infty e^{-v} \cdot \left(\sum_{j=0}^L \frac{\left(\frac{z}{M}\right)^j \cdot v^j}{j!} \right)^M \cdot dv \quad (39)$$

which is the generating function of N presented in (5) from which its probability density function is extracted from

$$E(z^N) = \sum_{k=1}^\infty z^k \cdot P(N = k). \quad (40)$$

To simplify the numerical evaluation of (39) the polynomial in the right side can be expressed as

$$\left(\sum_{j=0}^L \frac{\left(\frac{z}{M}\right)^j \cdot v^j}{j!} \right)^M = \sum_{i=0}^{L \cdot M} c_i \cdot \left(\frac{z \cdot v}{M}\right)^i \quad (41)$$

where c_i is the coefficient for the i th term that is computed by developing the left side of (41).

Then, using the following equality: $n! = \int_0^\infty x^n \cdot e^{-x} \cdot dx$, expression (39) can be written as

$$E(z^N) = 1 - (1 - z) \cdot \sum_{i=0}^{L \cdot M} c_i \cdot i! \cdot \left(\frac{z}{M}\right)^i. \quad (42)$$

Finally, using (40), the probability that $N = n$ is given by the n th term in (42), which can be extracted as

$$P(N = n) = c_n \cdot \left(\frac{n!}{M^n}\right) - c_{n-1} \cdot \left(\frac{(n-1)!}{M^{n-1}}\right). \quad (43)$$

This can be easily computed using a symbolic tool.

Finally, it is also worth noticing that for the case $L = 1$ the exact formulation is reduced to

$$P(N > n) = \prod_{i=0}^{n-1} \left(1 - \frac{i}{M}\right). \quad (44)$$

REFERENCES

[1] R. C. Baumann, "Soft errors in advanced computer systems," *IEEE Des. Test. Comput.*, vol. 22, no. 3, pp. 258–266, May/June. 2005.
 [2] R. D. Schrimpf and D. M. Fleetwood, *Radiation Effects and Soft Errors in Integrated Circuits and Electronic Devices*. Singapore: World Scientific, 2004.
 [3] M. Nicolaidis, "Design for soft error mitigation," *IEEE Trans. Device Mater. Rel.*, vol. 5, no. 3, pp. 405–418, Sep. 2005.
 [4] R. W. Hamming, "Error detecting and correcting codes," *Bell Syst. Tech. J.*, vol. 29, pp. 147–160, 1950.
 [5] A. M. Saleh, J. J. Serrano, and J. H. Patel, "Reliability of scrubbing recovery-techniques for memory systems," *IEEE Trans. Rel.*, vol. 39, no. 1, pp. 114–122, Apr. 1990.
 [6] R. Naseer and J. Draper, "DEC ECC design to improve memory reliability in sub-100 nm technologies," in *Proc. 15th IEEE ICECS*, 2008, pp. 586–589.
 [7] M. A. Bajura, Y. Boulghassoul, R. Naseer, S. DasGupta, A. Wituski, F. Sondeen, J. Stansberry, D. Draper, J. Massengill, and L. W. Damoulakis, "Models and algorithmic limits for an ECC-based approach to hardening sub-100-nm SRAMs," *IEEE Trans. Nucl. Sci.*, vol. 54, pt. 2, no. 4, pp. 935–945, Aug. 2007.
 [8] C. L. Chen and M. Y. Hsiao, "Error-correcting codes for semiconductor memory applications: A state-of-the-art review," *IBM J. Res. Dev.*, vol. 28, no. 2, pp. 124–134, 1984.

[9] J. A. Maestro and P. Reviriego, "Selection of the optimal memory configuration in a system affected by soft errors," *IEEE Trans. Device Mater. Rel.*, vol. 9, no. 3, pp. 403–411, Sep. 2009.
 [10] P. Reviriego, J. A. Maestro, and C. Cervantes, "Reliability analysis of memories suffering multiple bit upsets," *IEEE Trans. Device Mater. Rel.*, vol. 7, no. 4, pp. 592–601, Dec. 2007.
 [11] R. M. Goodman and M. Sayano, "The reliability of semiconductor RAM memories with on-chip error-correction coding," *IEEE Trans. Inf. Theory*, vol. 37, pt. 2, no. 3, pp. 884–896, May 1991.
 [12] M. Blaum, R. Goodman, and R. McEliece, "The reliability of single-error protected computer memories," *IEEE Trans. Comput.*, vol. 37, no. 1, pp. 114–119, Jan. 1988.
 [13] G. C. Yang, "Reliability of semiconductor RAMs with soft-error scrubbing techniques," *Proc. Inst. Elect. Eng.—Comput. Digit. Tech.*, vol. 142, no. 5, pp. 337–344, Sep. 1995.
 [14] T. P. Haraszti, *CMOS Memory Circuits*. Berlin, Germany: Springer-Verlag, 2000.
 [15] L. A. Barroso and U. Hözlze, *The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines*. Durham, NC: Morgan and Claypool, 2009.
 [16] W. Feller, *An Introduction to Probability Theory and Its Applications*. New York: Wiley, 2001.



Pedro Reviriego (M'03–A'03–M'04) received the M.Sc. and Ph.D. degrees (Honors) in telecommunications engineering from Technical University of Madrid, Madrid, Spain, in 1994 and 1997, respectively.

From 1997 to 2000, he was an R&D Engineer at Teldat, Madrid, working on router implementation. In 2000, he joined Massana to work on the development of 1000BaseT transceivers. During 2003, he was a Visiting Professor at University Carlos III, Leganés, Madrid. From 2004 to 2007, he was Distinguished Member of Technical Staff with LSI Corporation working on the development of Ethernet transceivers. He is currently with Universidad Antonio de Nebrija, Madrid. His research interests are fault tolerant systems, performance evaluation of communication networks and the design of physical layer communication devices. He has authored numerous papers in international conferences and journals. He has also participated in the IEEE 802.3 standardization for 10GBaseT.



Lars Holst received the Ph.D. degree from Uppsala University, Uppsala, Sweden, in 1972.

From 1971 to 1988, he was Senior Lecturer at Uppsala University. He had the Chair as professor in mathematical statistics at the Royal Institute of Technology, Stockholm, Sweden, from 1989 until his retirement in 2007. His research interests are in different parts of probability theory often with a combinatorial flavour. He has authored books and numerous research papers in international journals. He was involved in the leadership of the Royal Institute of Technology as dean and other commissions of trust. As Professor Emeritus he continues doing some research on problems he finds interesting.



Juan Antonio Maestro (M'07) received the M.Sc. degree in physics and the Ph.D. degree in computer science from Universidad Complutense de Madrid, Madrid, Spain, in 1994 and 1999, respectively.

He has served both as a Lecturer and a Researcher at several universities, such as Universidad Complutense de Madrid; Universidad Nacional de Educación a Distancia (Open University), Madrid; Saint Louis University, Madrid; and Universidad Antonio de Nebrija, Madrid, where he currently manages the Computer Architecture and Technology Group. His current activities are oriented to the space field, with several projects on reliability and radiation protection, as well as collaborations with the European Space Agency. He is the author of numerous technical publications, both in journals and international conferences. Aside from this, he has worked for several multinational companies, managing projects as a Project Management Professional and organizing IT departments. His areas of interest include high level synthesis and cosynthesis, signal processing and real-time systems, fault tolerance, and reliability.