can be rewritten as

$$\Phi^* = d_1(\bar{d}\bar{e} + \bar{a}\bar{e}\bar{c} + \bar{a}\bar{b}e + \bar{a}\bar{b}cd + a\bar{b}\bar{c}\bar{d})$$

$$+ \bar{d}_1(\bar{a}be + bcd + b\bar{c}\bar{d}e + ad + ace).$$

Noting that the desired function is to be realized on node 1 of the circuit, the CF of the specification can be obtained [2], [3] as

$$\Phi_s = d_1 f + \bar{d}_1 \bar{f} + \text{DON'T CARE terms}$$

$$= d_1(\bar{a}\bar{b} + \bar{d}\bar{e} + \bar{a}\bar{c}\bar{e} + \bar{b}\bar{c}\bar{d})$$

$$+ \bar{d}_1(ad + be + ace + bcd)$$

$$+ a\bar{b}c\bar{d}e.$$

From [2] and [3], a circuit realizes its specification iff $\Phi^* \leqslant \Phi_s$. In other words, $\Phi^* \cdot \bar{\Phi}_s = 0$. That the latter relation is satisfied can be easily verified by inspection, since $\bar{\Phi}_s$ can be expressed as

$$\bar{\Phi}_s = [\bar{d}_1(\bar{a}\bar{b} + \bar{d}\bar{e} + \bar{a}\bar{c}\bar{e} + \bar{b}\bar{c}\bar{d}) + d_1(ad + be + ace + bcd)] \cdot \overline{(a\bar{b}c\bar{d}e)}.$$

VIII. CONCLUSIONS

We have shown how multiple valued characteristic functions can be used to extract functional descriptions of CSA networks from their structure and external signal constraints.

The algebraic manipulations can be programmed on a computer, for instance, by representing the CF's using cubical complexes [2], [15] extended for multiple values of the constituent variables. The technique described here can be used for formal verification, since it transforms a network into a series of logic functions. The analysis can be aided by applying the properties of Boolean equations [2]–[5], [16]. Our current work is in this direction [13], [14].

REFERENCES

[1] J. P. Hayes, "A unified switching theory with applications to VLSI design," *Proc. IEEE*, vol. 70, pp. 1140–1151, Oct. 1982.
[2] E. Cerny and M. A. Marin, "An approach to unified methodology of combinational switching circuits," *IEEE Trans. Comput.*, vol. C-26, pp. 745–756, Aug. 1977.
[3] E. Cerny, "Controllability and fault observability in modular combinational circuits," *IEEE Trans. Comput.*, vol. C-27, pp. 896–903, Oct. 1978.
[4] ——, "Unique and identity solutions of Boolean equations," *Digital Processes*, vol. 3, pp. 331–337, Fall 1977.
[5] ——, "Characteristic functions in multivalued logic systems," *Digital Processes*, vol. 6, pp. 167–174, 1980.
[6] E. Cerny, D. Mange, and E. Sanchez, "Synthesis of minimal binary decision trees," *IEEE Trans. Comput.*, vol. C-28, pp. 472–482, July 1979.
[7] E. Cerny and J. Gecsei, "Simulation of MOS circuits by decision diagrams," *IEEE Trans. Computer-Aided Design*, vol. CAD-4, pp. 685–693, Oct. 1985.
[8] R. E. Bryant, "An algorithm for MOS logic simulation," *LAMBDA/VLSI J.*, vol. 1, pp. 46–53, Fourth Quarter 1980.
[9] ——, "A switch-level model and simulator for MOS digital systems," *IEEE Trans. Comput.*, vol. C-33, pp. 160–177, Feb. 1984.
[10] ——, "Symbolic verification of MOS circuits," in *Proc. Chapel Hill Conf. VLSI*, 1985.
[11] M. Gordon, "Proving a computer correct with LCF-LSM hardware verification system," Computer Lab. Tech. Rep. 42, Cambridge University, 1983.
[12] D. L. Dill and E. M. Clarke, "Automatic verification of asynchronous circuits using temporal logic," in *Proc. Chapel Hill Conf. VLSI*, 1985.
[13] C. Berthet and E. Cerny, "Input-constrained memory elements in speed-independent circuits," in *Proc. Canadian Conf. VLSI (CCVLSI '86)*, Oct. 27–28, 1986.
[14] ——, "An algebraic model for asynchronous circuits verification," Publi. 571, Département d' informatique et de recherche opérationnelle, Université de Montréal, *IEEE Trans. Comput.*, to be published.
[15] D. Dietmeyer, *Logic Design of Digital Systems*, 2nd ed. New York: Allen and Bacon, 1978.
[16] S. Rudeanu, *Boolean Functions and Equations.* Amsterdam, The Netherlands, North-Holland, 1974.

The Reliability of Single-Error Protected Computer Memories

MARIO BLAUM, RODNEY GOODMAN, AND
ROBERT MCELIECE

Abstract—In this paper we study the lifetimes of computer memories which are protected with single-error correcting double-error detecting (SEC–DED) codes. We assume that there are five possible types of memory chip failures (single-cell, row, column, row–column, and whole chip) and, after making a simplifying assumption (the Poisson assumption) that has been substantiated experimentally, we derive a simple closed-form expression for the system reliability function. Using this formula, and chip reliability data from tables, we are easily able to compute the mean time to failure for realistic memory systems.

Index Terms—Error-correcting codes, RAM's, reliability, SEC–DED codes, semiconductor memory systems.

I. INTRODUCTION

All modern computers have memories built from VLSI RAM chips. Individually, these devices are highly reliable, and any single chip may function for decades before failing. However, when many chips are combined in a single large computer memory, the expected wait until at least one of them fails can be as small as a few hours, or even less. For this reason, almost all large computer memories are protected by *single-error correcting, double-error detecting* (SEC–DED) codes. In the usual jargon of coding theory, a SEC–DED code is just a shortened $d = 4$ Hamming code; the shortening is usually done in a hardware-efficient manner devised by Hsiao [7]. A recent paper by Chen and Hsiao [4] gives a good survey of how SEC–DED coding is actually implemented in computer memories, but we shall give a brief overview here.

Most often, the memory is organized as an $M \times n$ rectangular array of chips as shown in Fig. 1. The first k chips in each row are used to store information, and the remaining $r = n - k$ chips are *parity-check chips*, needed for the SEC–DED coding. As we shall see, this means that a (n, k) $d = 4$ Hamming code is being used to protect the memory. A typical small example is the standard one-megabyte board sold for VAX computers, which is organized as $M = 4$ rows of 64K RAM chips, with each row containing $k = 32$ data chips and $r = 7$ parity chips. The corresponding SEC–DED code is then a $(39, 32)$ $d = 4$ shortened Hamming code.

We assume that each chip has a one-bit wide external organization but is organized internally as an $l \times l$ square array of bits. For example, the standard 4164 64K chips have an external organization of 65536 single-bit locations, with $l = 256$. We also assume that the n bits stored in corresponding cells in one row of the memory form a codeword in the (n, k) code, as shown in Fig. 2.

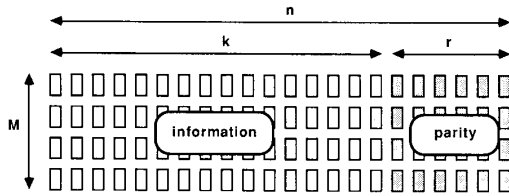We now discuss chip failures. By a chip "failure" we mean any

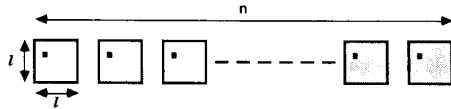Fig. 1.   The organization of memory chips.

Fig. 2.   Each bit in each codeword resides on a different chip.
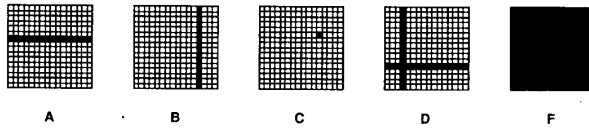

Fig. 3.   Five types of chip failure.

situation in which one or more of the bits written on a chip cannot be reliably read. These failures are traditionally classified as either *hard,* meaning that the affected memory cells are permanently damaged, or *soft,* meaning that the damage is only temporary.

Laboratory observation of real memories [1], [12], [13] shows that by far the most common type of chip failure is a soft error of a single cell on a chip. Such errors are caused by stray alpha particles which can, under certain circumstances, change a stored "1" to a "0" without otherwise damaging the chip [14]. However, several kinds of hard failure have been observed. A *single-cell failure,* for example, can also occur as a hard error. There are also several kinds of hard failures which cause multiple cell errors. A *row failure,* which can be caused by a failure of one of the chip's row drivers, causes all *l* cells in one row of the affected chip to fail. Similarly, a *column failure,* which can be caused by a failure of one of the chip's column sense amplifiers or column decoders, causes all *l* cells in a column of the chip to fail. A short circuit at a memory cell can cause a *row-column* failure. Finally, a catastrophic *whole chip failure* may occur, in which *all* the cells of a chip fail. All five of these failure types are illustrated in Fig. 3. (The letters *A, B, C, D,* and *F* will be referred to in Section II.)

The organization of the SEC-DED code (assuming "by-one" memory chips) guarantees that no chip failure, however catastrophic, can cause two errors in any *n* bit codeword, and so the memory system will survive any single-chip failure. In fact there are many possible combinations of multiple chip failures that can also be tolerated. Eventually, however, we expect that so many chip failures will have occurred that some one of the individual *n* bit codewords will have suffered two or more errors. When this happens, we declare a *memory system failure.*

If we start with a brand new memory system of the kind we have been describing, the time until a memory system failure occurs will be a random variable. In this paper we will derive accurate and easily evaluated estimates for two of the most important quantitative measures of this random variable, the *system reliability function* $R_{sys}(t)$ and the *mean time to failure* (MTTF). The reliability function $R_{sys}(t)$ represents the probability that the system will *not* have failed after *t* hours, and the MTTF represents the average length of time the system will function before a memory system failure occurs.

In the next section (Section II) we will describe the probabilistic model which is commonly used to describe the occurrences of the various types of chip failures, and see how it leads, via a Poisson

approximation, to a reasonably simple formula for $R_{sys}(t)$ and MTTF. In Section III we will give some useful approximations to the exact formula found in Section II. In Section IV we will make numerical comparisons of the predictions of our formulas to the results of computer simulation. The analytic predictions will be seen to be in very close agreement with the simulations, thereby justifying our confidence in the accuracy of the Poisson approximation made in Section II. Finally, in Section V, we will compare our results to those which have already appeared in the literature.

II. MODELS: FORMULAS FOR $R(t)$ AND MTTF

Our basic quantitative assumption about individual chip failures is that they are *exponentially distributed.* This means that the reliability of a given chip, i.e., the probability that it has not failed after *t* hours is equal to $e^{-\lambda t}$, where λ is a constant that must be found experimentally [11]. We will need to distinguish between the five types of chip errors depicted in Fig. 3, and so for future reference, we will use the following notation.

A: row failure
B: column failure
C: single-cell failure
D: row-column failure
F: whole chip failure.

We assume that, if a given chip fails, the conditional probabilities that the failure will be of type *A, B, C, D,* or *F* are *a, b, c, d,* and *f,* respectively. The probabilities *a, b, c, d,* and *f* also have to be determined experimentally. We further assume that failures on one chip are independent from failures on all other chips.

Given all these assumptions, it is in principle possible to calculate the *row reliability function* $R(t)$, defined as follows.

$R(t) = \text{Pr}$ {an uncorrectable combination of chip failures has
not yet occurred in the *i*th row of chips at time *t.*}

$$(1)$$

For example, if the only kind of chip failures were whole chip failures, we would have $a = b = c = d = 0, f = 1$ and

$$R(t) = e^{-\lambda n t} + n(1 - e^{-\lambda t})e^{-\lambda(n-1)t}, \qquad (2)$$

which is just the probability that the given row has suffered either zero or one whole chip failure after *t* hours [10]. Since the *M* rows are assumed to fail independently, the reliability of the entire system of *Mn* chips is

$$R_{sys}(t) = R(t)^M. \qquad (3)$$

The MTTF of system whose reliability function is $r(t)$ is well known to be given by the formula

$$\text{MTTF} = \int_0^\infty r(t) \, dt, \qquad (4)$$

and so for a computer memory system of the kind we are considering,

$$\text{MTTF} = \int_0^\infty R(t)^M \, dt. \qquad (5)$$

Thus, everything we are interested in depends in a simple manner on the row-reliability function $R(t)$.

Unfortunately, however, an exact formula for $R(t)$ proves to be extremely complicated. (For example, Mikhail *et al.* [15] give a recursive method for computing it when errors of types *A, B, C,* and *F* are present.) Thus, difficulty has led us to make the following simplifying assumption. We no longer view a row of chips as consisting of *n* separate chips, but "end on" so that the failures on all *n* chips are superimposed onto a single "protochip." We also make an important assumption about how protochip failures are distributed,
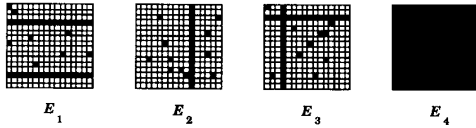
Fig. 4. The four tolerable failure configurations on the Poisson protochip.

which we call the *Poisson assumption:*

> *In each protochip, the failures of types A, B, C, D, and F form independent Poisson processes of intensities $an\lambda$, $bn\lambda$, $cn\lambda$, $dn\lambda$, and $fn\lambda$, respectively.*

Under this assumption, the row reliability function $R(t)$ is just the probability that at time t no cell on the protochip has suffered two or more errors. As we will see, this assumption greatly simplifies the formulas for $R(t)$ without introducing significant inaccuracies. For example, if we again consider a situation in which only whole chip failures occur, then under the Poisson assumption the number of whole chip failures in a given row is a Poisson process of intensity λn, and so the row-reliability function, i.e., the probability of zero or one whole chip failures after n hours is given by

$$R(t) = e^{-\lambda nt}(1 + \lambda nt). \qquad (6)$$

Formulas (6) and (2) are very similar, and for example if we compute the MTTF of a single row of chips using (2) and (4) we obtain

$$\text{MTTF} = \frac{1}{\lambda}\left(\frac{1}{n} + \frac{1}{n-1}\right),$$

whereas, if we use the Poisson assumption via (6) and (4) we get instead

$$\text{MTTF} = \frac{1}{\lambda}\left(\frac{2}{n}\right).$$

In Section IV we will give further comparisons between exact MTTF's and those obtained by the Poisson protochip method.

Using the Poisson protochip, we can now derive a formula for the row-reliability function $R(t)$, the probability that the memory system is still working after t hours. It will be convenient to classify the various tolerable combinations of protochip failures into the following four categories.

E_1: only row and single-cell failures
E_2: only column and single-cell failures
E_3: one row–column failure and single-cell failures
E_4: one whole chip failure.

These "tolerable failure configurations" are shown in Fig. 4.

We note that the configurations E_1 and E_2 are not disjoint, so that we also introduce E_{12}, defined as

$$E_{12} = E_1 \cap E_2: \text{only single-cell failures.}$$

Then the row-reliability function defined by (1) is

$$R(t) = \Pr\{E_1 \cup E_2 \cup E_3 \cup E_4\}$$
$$= \Pr\{E_1\} + \Pr\{E_2\} - \Pr\{E_{12}\} + \Pr\{E_3\} + \Pr\{E_4\}. \qquad (7)$$

We now proceed to calculate the five probabilities in (7).

We begin by calculating $\Pr\{E_1\}$, which is the probability that the protochip has suffered no errors of type B, D, or F, and that the errors of type A and C are "tolerable," i.e., that no cell on the protochip has suffered two or more errors. To do this, we focus our attention on a single row of cells on the protochip, say row i. Since each protochip has l rows, our Poisson assumption implies that the row failures in this particular row form a Poisson process of intensity $an\lambda/l$. Therefore, the probability that there have been no row failures

in row i at time t is

$$\exp(-an\lambda t/l),$$

and the probability that there has been exactly one row failure in this row is

$$\exp(-an\lambda t/l)(an\lambda t/l).$$

To simplify the notation, from now on we will use the parameter x defined by

$$x = \lambda nt, \qquad (8)$$

so that the probability of zero and one failures in row i at time t are given by

$$e^{-ax/l} \text{ and } e^{-ax/l}(ax/l),$$

respectively. Next we focus on a particular cell within the ith row, say cell (i, j). Since each chip contains l^2 cells, our Poisson assumption implies that the single-cell failures in this particular cell form a Poisson process with intensity $cn\lambda/l^2$. Therefore, the probabilities that the (i, j)th cell has suffered zero or one single-cell errors at time t are

$$e^{-cx/l^2} \text{ and } e^{-cx/l^2}(cx/l^2),$$

respectively. It follows that the probability that at time t no cell in row i has suffered a single-cell error is

$$(e^{-cx/l^2})^l = e^{-cx/l},$$

while the probability that no cell in row i has suffered more than one single-cell error is

$$\left(e^{-cx/l^2}\left(1 + \frac{cx}{l^2}\right)\right)^l.$$

We now wish to calculate the probability that no cell within the ith row has suffered two or more errors of type A or C. This probability is the sum of the following two probabilities:

• *The row has not failed and there is at most one single-cell error in each of the l cells.*

• *The row has failed and there are no single-cell errors in any of the l cells.*

This probability is, therefore,

$$e^{-ax/l}\left(e^{-cx/l^2}\left(1 + \frac{cx}{l^2}\right)\right)^l + e^{-ax/l}(ax/l)(e^{-cx/l}). \qquad (9)$$

Finally, the probability $\Pr\{E_1\}$ is the probability that the type A and C errors in all l rows of the protochip will be tolerable, which is just the lth power of (9), multiplied by the probability that there are no errors of type B, D, or F, which is $\exp(-(b + d + f)x)$. After some algebra we find that this product is

$$\Pr\{E_1\} = e^{-x}\left(\left(1 + \frac{cx}{l^2}\right)^l + \frac{ax}{l}\right)^l. \qquad (10)$$

By replacing "rows" with "columns" in the preceding argument, we find that

$$\Pr\{E_2\} = e^{-x}\left(\left(1 + \frac{cx}{l^2}\right)^l + \frac{bx}{l}\right)^l. \qquad (11)$$

To compute $\Pr\{E_{12}\}$, we note that this case corresponds to no errors of types A, B, D, or F, and at most one error in each cell of the

protochip. Thus,

$$\Pr\{E_{12}\} = e^{-(a+b+d+f)x} \cdot \left(e^{-cx/l^2}\left(1+\frac{cx}{l^2}\right)\right)^{l^2}$$

$$= e^{-x}\left(1+\frac{cx}{l^2}\right)^{l^2}. \tag{12}$$

In order to calculate $\Pr\{E_3\}$, note that if there is a row–column failure and if there are no failures of types A, B, or F, we can tolerate zero or one single-cell failures in the unaffected $(l-1)^2$ cells but no errors in the $2l-1$ cells affected by the row–column failure. Hence,

$$\Pr\{E_3\} = e^{-(a+b+f)x} \cdot e^{-dx}(dx) \cdot (e^{-cx/l^2})^{2l-1}$$

$$\cdot \left(e^{-cx/l^2}\left(1+\frac{cx}{l^2}\right)\right)^{(l-1)^2}$$

$$= e^{-x}dx\left(1+\frac{cx}{l^2}\right)^{(l-1)^2}. \tag{13}$$

To calculate $\Pr\{E_4\}$, we note that if there is a whole chip failure, there can be no further failures of any kind. Therefore,

$$\Pr\{E_4\} = e^{-(a+b+c+d)x} \cdot e^{-fx}fx$$

$$= e^{-x}fx. \tag{14}$$

Finally, to compute the row-reliability function $R(t)$, we combine (7) with (10), (11), (12), (13), and (14), and obtain the following somewhat intimidating expression.

$$R(t) = e^{-x}\left(\left(\left(1+\frac{cx}{l^2}\right)^l+\frac{ax}{l}\right)^l+\left(\left(1+\frac{cx}{l^2}\right)^l+\frac{bx}{l}\right)^l\right.$$

$$\left.-\left(1+\frac{cx}{l^2}\right)^{l^2}+dx\left(1+\frac{cx}{l^2}\right)^{(l-1)^2}+fx\right). \tag{15}$$

This expression is our main result. The rest of the paper will be devoted to exploring its consequences.

III. Asymptotic Approximations and Special Cases

Although the expression (15) for $R(t)$ is simple enough for numerical work, it is possible to give approximations to it that will yield additional insight into the problem. For example, in most modern chips the number l (the number of storage cells per row or column) is quite large, and this suggests that the limiting behavior of (15) as $l \to \infty$ may be interesting to consider. In fact, this limit is given by the formula

$$R_\infty(t) = e^{-x}(e^{(a+c)x}+e^{(b+c)x}+e^{cx}(dx-1)+fx). \tag{16}$$

This formula is simple enough to integrate explicitly, and so by (5) we find that the MTTF for one row of chips is approximately

$$\int_0^\infty R_\infty(t)\,dt = \frac{1}{\lambda n}\left(\frac{1}{1-a-c}+\frac{1}{1-b-c}-\frac{1}{1-c}+\frac{d}{(1-c)^2}+f\right). \tag{17}$$

Experimentation with (16) and (17) indicates that these approximations are quite accurate if $\max(a+c, b+c)$ is not too close to 1. Such a restriction is understandable, since if, e.g., $c = 1$, an $l = \infty$ chip would have an infinite MTTF, since the probability of any given cell position being hit twice would be zero.

Another interesting case to consider is the case of a large number of rows. (A CRAY-1, for example, has $M = 1024$ rows, each consisting of 72 1K ECL RAM chips.) In this case we can exploit the classic theory of asymptotic analysis [3] and find an asymptotic

formula for the integral in (5). Omitting the details, we find that the result is of the form

$$\text{MTTF} \sim \frac{1}{\lambda nM}(\sqrt{M} \cdot K_1 + K_2). \tag{18}$$

In (18) the number $1/\lambda nM$ represents the mean time between chip failures (there are nM chips and each one has an MTTF of $1/\lambda$). The other term, viz., $\sqrt{M}\cdot K_1 + K_2$ represents the asymptotic value of the *mean number of events to failure* (METF), which is the average number of chip failures which occur before a system failure occurs. The constants K_1 and K_2 in (18) are determined as follows. If we call the bracketed term in (15) $r(x)$ and expand it as a polynomial in x up to terms of degree 3, we find that

$$r(x) = 1 + x + r_2 x^2 + r_3 x^3 + \cdots, \tag{19}$$

where

$$r_2 = \frac{c^2}{2}\cdot\frac{(l^2-1)}{l^2}+(a+b)c\cdot\frac{(l-1)}{l}+\frac{(a^2+b^2)}{2}$$

$$\cdot\frac{(l-1)}{l}+cd\cdot\frac{(l^2-2l+1)}{l^2}$$

and

$$r_3 = \frac{c^3}{6}\cdot\frac{(l^4-3l^2+2)}{l^4}+\frac{(a+b)c^2}{2}\cdot\frac{(l^3-2l^2+1)}{l^3}$$

$$+\frac{(a^2+b^2)c}{2}\cdot\frac{(l^2-3l+2)}{l^2}+\frac{(a^3+b^3)}{6}$$

$$\cdot\frac{(l^2-3l+2)}{l^2}+\frac{c^2d}{2}\cdot\frac{(l^3-4l^2+5l-2)}{l^3}.$$

The constants K_1 and K_2 in (18) are then given by

$$K_1 = \sqrt{\frac{\pi}{2(1-2r_2)}} \tag{20a}$$

$$K_2 = \frac{2(r_3-r_2)+\frac{2}{3}}{(1-2r_2)^2}. \tag{20b}$$

The difference between the asymptotic formula (18) and the true value (5) is guaranteed to go to zero, if a, b, c, d, f, and l are fixed and $M \to \infty$. However, (18) usually gives remarkably accurate answers for small values of M, often even for $M = 1$, as we shall see in the next section.

We now consider two special cases, viz., $f = 1$ and $c = 1$. When $f = 1$ (all failures are whole chip failures), the reliability function $R(t)$ in (15) becomes simply

$$R(t) = e^{-x}(1+x),$$

which is exactly the same as the $l = \infty$ approximation (16). Therefore, the MTTF for one row of chips is given exactly by (17), i.e.,

$$\text{MTTF} = \frac{2}{\lambda n}.$$

Since there are n chips, each with MTTF equal to $1/\lambda$, this formula simply reflects the fact that the system will fail as soon as two whole chip failures have occurred. (We already observed this in Section II.) More interesting is what happens when the number of rows M is large. In this case the system will fail as soon as one of the M chip rows has suffered two errors. If we interpret the occurrence of a chip

failure in the ith row as the arrival of a "person" whose "birthday" occurs on the ith day of an M day year, we see that the expected number of chip failures needed to cause a system failure is the same as the expected number of people we need to interview before we find two with the same birthday. It is known that this "birthday surprise" number is given by [8]

$$B(M) = M \cdot \int_0^\infty (e^{-x}(1+x))^M \, dx. \tag{21}$$

This is in agreement with the theory developed in Section II, since with $f = 1$ the formula (15) simplifies to $R(t) = e^{-x}(1 + x)$, and so by (5) the MTTF is

$$\text{MTTF} = \frac{1}{\lambda n} \cdot \int_0^\infty (e^{-x}(1+x))^M \, dx$$

$$= \frac{1}{\lambda nM} \cdot B(M).$$

Furthermore, since $r(x) = 1 + x$ in this case, in the expansion (19) we have $r_2 = r_3 = 0$, and so the asymptotic formula (18) becomes

$$\text{MTTF} \sim \frac{1}{\lambda nM} \left(\sqrt{\frac{\pi M}{2}} + \frac{2}{3} \right). \tag{22}$$

What this means is that the term $\sqrt{\pi M/2} + 2/3$ is an approximation to the METF, which in this case is simply the mean number of whole chip failures before system failure, as well as the "birthday surprise" number $B(M)$:

$$B(M) = \text{METF} \sim \sqrt{\frac{\pi M}{2}} + \frac{2}{3}. \tag{23}$$

The approximation given in (23) is very accurate, too, as the following table shows.

M	Exact $B(M)$ [from (21)]	Approx. $B(M)$ [from (23)]
1	2.000	1.920
2	2.500	2.439
4	3.219	3.173
8	4.245	4.212
16	5.704	5.680
32	7.774	7.756
⋮		
365	24.616	24.611

The $M = 365$ entry shows that on a planet with a 365-day year, one needs to interview between 24 and 25 people, on the average, before finding two with the same birthday. Alternatively, in a 365-row computer memory in which only whole chip failures occur, and in which each row is SEC–DED protected, the memory will tolerate between 24 and 25 chip failures, on the average, before failing.

A similar "birthday analysis" can be made for the case $c = 1$. In this case only single-cell failures occur; it is as if there were l^2M independent rows of 1×1 chips. The number of single-cell failures which can be tolerated before a system failure occurs is thus $1/\lambda nM$ times the "birthday number" $B(l^2M)$. This too is consistent with our theory, since with $c = 1$, the formula (15) becomes

$$R(t) = e^{-x} \left(1 + \frac{x}{l^2} \right)^{l^2},$$

and so by (5)

$$\text{MTTF} = \frac{1}{\lambda n} \cdot l^2 \cdot \int_0^\infty (e^{-x}(1+x))^{l^2M} \, dx$$

$$= \frac{1}{\lambda nM} \cdot B(l^2M).$$

The asymptotic formula (18) is even more accurate in this case, since with $r(x) = (1 + x/l^2)^{l^2}$ we have

$$r_2 = \frac{c^2}{2} \cdot \frac{(l^2 - 1)}{l^2}$$

$$r_3 = \frac{c^2}{6} \cdot \frac{(l^4 - 3l^2 + 2)}{l^4}$$

which means that the asymptotic formula (18) works out to be

$$\text{MTTF} \sim \frac{1}{\lambda nM} \left(l\sqrt{\frac{\pi M}{2}} + \frac{2}{3} \right), \tag{24}$$

exactly the same as (22) except that M is replaced with l^2M. In this case it is typically quite difficult to integrate $R(t)^M$ accurately, and the approximation (24) provides the only reliable way of obtaining accurate values for the MTTF.

The relationship between MTTF's and "birthday surprises" was originally noted in [6] and [14].

IV. NUMERICAL EVALUATION OF THE MTTF FORMULAS

In this section we will illustrate our results numerically. The plan is to take three sets of values for the parameters a, b, c, d, f, and l, as reported in the literature, and for various values of M to compute the METF (which differs from the MTTF by the factor $1/\lambda Mn$, as we explained in the last section) using four methods. The first method is direct Monte Carlo simulation. This method is very slow, but it makes no use of our Poisson assumption and provides a valuable check of the accuracy of our other methods. The second method is direct integration of the expression (15):

$$\text{METF} = M \cdot \int_0^\infty (R(x))^M \, dx.$$

The third method is direct integration of the $l = \infty$ approximation of the row-reliability function, given in (16):

$$\text{METF} = M \cdot \int_0^\infty (R_\infty(x))^M \, dx.$$

The final method is to use the two-term asymptotic approximation given by (18), viz.,

$$\text{METF} = \sqrt{M}K_1 + K_2.$$

The sets of parameters are taken from [9], [11], and [15], as summarized in the following table.

	a	b	c	d	f	l
[9]	0.01646	0.01646	0.85343	0	0.11365	128
[11]	0.047	0.047	0.893	0.013	0	128
[15]	0.12	0.18	0.35	0	0.35	64

Our numerical results for the corresponding METF's are given in Figs. 5–7. Each of the numbers in the "simulation" columns represents the average number of (simulated) chip failures before (simulated) system failure for 40 000 Monte Carlo trials, reported to the nearest tenth.

We see in very case that our "exact" expression (15) gives results that are indistinguishable from the simulation results. The $l = \infty$

	Simulation	exact(15)	$l = \infty$(16)	M large(18)
$M = 1$	8.3	8.458	8.662	5.142
$M = 2$	8.9	8.900	9.023	6.260
$M = 4$	9.8	9.710	9.783	7.842
$M = 8$	11.4	11.283	11.328	10.079
$M = 16$	14.1	13.997	14.032	13.243
$M = 32$	18.4	18.200	18.234	17.717

Fig. 5. Numerical estimates for METF, using data from [9].

	Simulation	exact(15)	$l = \infty$(16)	M large(18)
$M = 1$	20.9	20.774	25.122	20.367
$M = 2$	26.3	26.286	30.770	25.905
$M = 4$	33.8	34.058	39.145	33.737
$M = 8$	45.2	45.067	51.263	44.813
$M = 16$	60.9	60.671	68.589	60.477
$M = 32$	83.9	82.773	93.224	82.630

Fig. 6. Numerical estimates for METF, using data from [11].

	Simulation	exact(15)	$l = \infty$(16)	M large(18)
$M = 1$	2.8	2.793	2.826	2.506
$M = 2$	3.4	3.359	3.384	3.163
$M = 4$	4.2	4.225	4.248	4.092
$M = 8$	5.5	5.496	5.521	5.406
$M = 16$	7.3	7.326	7.356	7.263
$M = 32$	9.9	9.934	9.972	9.890

Fig. 7. Numerical estimates for METF, using data from [15].

approximation is good, but not as good, and is consistently high. This is because with $l = \infty$ single-cell errors become negligible, as do, for example, pairs of row errors. The asymptotic estimate is always low, because the complete asymptotic expansion of the METF involves positive terms which we have neglected.

We feel that these results justify our confidence in the accuracy of the Poisson approximation. An independent verification of the accuracy of the Poisson approximation, based on a mathematical analysis, was given in [2].

V. A SURVEY OF RELATED WORK: CONCLUSION

The literature contains many papers devoted in whole or part to the subject of this paper, including two survey papers ([9] and [16]). In this section we will attempt to describe how our work adds to what is already known.

The earliest work on ECC memory reliability [10] deals only with type F chip failures, i.e., whole chip failures. Later models, including those in [9] and [11], extended the types of failure modes to include types A, B, C, and D, but as pointed out in [16], it is implicitly assumed in these models that the failure types are "nested." That is, there is a hierarchy of failure types, such that each type is a subset of the previous type. For example, single-cell, row, and whole chip failures are nested, but no nested hierarchy can contain both row and column failures. Since one row and one column failure in a row of chips will cause a memory system failure, it is important to have a model that handles "crossed" failure types, e.g., failure types A and B simultaneously, as is done in [15]. However, [15] does not consider failure type D. Indeed, our model is to our knowledge the only one that handles all five failure types A, B, C, D, and F simultaneously.

We believe that the key innovation of our paper, however, is the introduction of the Poisson approximation. As we have seen, this approximation allows us to obtain simple formulas for the system reliability without sacrificing significant accuracy. And although our main formula (15) may seem excessively complex, when compared to the corresponding formulas in [9], [11], and [15], it is very simple indeed. As we have demonstrated in Section IV, it can be easily programmed to give fast and accurate reliability estimates that can be used by memory system designers.

REFERENCES

[1] J. M. Ayache and M. Diaz, "A reliability model for error corrective memory system," *IEEE Trans. Reliability*, vol. R-28, Oct. 1979.
[2] M. Blaum, Caltech Ph.D. Thesis, 1985, appendix 2.
[3] N. G. de Bruijn, *Asymptotic Methods in Analysis*. Amsterdam, The Netherlands: North-Holland, 1961.
[4] C. L. Chen and M. Y. Hsiao, "Error-correcting codes for semiconductor memory applications: A state-of-the-art review," *IBM J. Res. Develop.*, vol. 28, pp. 124–134, Mar. 1984.
[5] R. M. F. Goodman and R. J. McEliece, "Lifetime analyses of error-control coded semiconductor RAM systems," *Proc. IEE*, part E, vol. 3, pp. 81–85, 1982.
[6] ———, "Hamming codes, computer memories, and the birthday surprise," in *Proc. 20th Allerton Conf. Commun., Control, Comput.*, 1982, pp. 672–679.
[7] M. Y. Hsiao, "A class of optimal minimum odd-weight-column SEC-DED codes," *IBM J. Res. Develop.*, vol. 14, pp. 395–401, July 1970.
[8] M. S. Klampkin and D. J. Newman, "Extensions of the birthday surprise," *J. Comb. Theory*, vol. 3, pp. 279–282.
[9] D. Y. Koo and H. B. Chenowith, "Choosing a practical MTTF model for ECC memory chip," in *Proc. 1984 IEEE Reliability Maintainability Symp.*, pp. 255–261.
[10] L. Levine and W. Meyers, "Semiconductor memory reliability with error detecting and correcting code," *Computer*, vol. 9, pp. 43–50, Oct. 1976.
[11] D. Marston, "Memory system reliability with ECC," Intel Appl. Note AP-73, Intel Corp., 1980.
[12] T. C. May, "Soft errors in VLSI: Present and future," *IEEE Trans. Components, Hybrids, Manuf., Technol.*, vol. CHMT-2, pp. 377–387, Dec. 1979.
[13] T. C. May and M. H. Woods, "A new physical mechanism for soft errors in dynamic memories," in *Proc. 1978 Int. Reliability Phy. Symp.*, Apr. 1978, pp. 33–40.
[14] R. J. McEliece, "The reliability of computer memories," *Scientif. Amer.*, vol. 252, pp. 88–95, Jan. 1985.
[15] W. F. Mikhail, R. W. Bartoldus, and R. A. Rutledge, "The reliability of memory with single-error correction," *IEEE Trans. Comput.*, vol. C-31, pp. 560–564, 1982.
[16] R. A. Rutledge, "Models for the reliability of memory with ECC," in *Proc. 1985 IEEE Reliability Maintainability Symp.*, pp. 57–62.

FISHNET: A Distributed Architecture for High-Performance Local Computer Networks

YONG J. KANG, JAMES H. HERZOG, AND JOHN SPRAGINS

Abstract—FISHNET (Facilities Integrated in a Shared Habitat NETwork) addresses the problem of effectively integrating a wide variety of computers, terminals, memory devices, and computer peripherals in a local environment. High performance is achieved by effectively separating the high volume node-to-node data communications and the low-