# Technical Research Report

SYSTEMS RESEARCH CENTER

*Supported by the National Science Foundation Engineering Research Center Program (NSFD CD 8803012), Industry and the University*

# The Reliability of Systems with Two Levels of Fault Tolerance: The Return of the "Birthday Surprise"

*by G.C. Yang and T. Fuja*

TR 90-49

# TECHNICAL RESEARCH REPORT

SYSTEMS RESEARCH CENTER

*Supported by the National Science Foundation Engineering Research Center Program (NSFD CD 8803012) and Industry*

Research support for this report has been provided by PYI Grant # NCR-8957623

# The Reliability of Systems with Two Levels of Fault Tolerance: The Return of the "Birthday Surprise"

*by G.C. Yang and T. Fuja*

SRC TR 90-49

# The Reliability of Systems With Two Levels of Fault Tolerance: The Return of the "Birthday Surprise"

Guu-chang Yang and Tom Fuja*
Department of Electrical Engineering
Systems Research Center
University of Maryland
College Park, MD 20742

**Abstract**

This paper considers the reliability of systems that employ fault tolerance at two different hierarchical levels. Specifically, it assumes the system consists of a two-dimensional array of components. Each component is reliable as long as it has been afflicted by no more than $t$ faults; when $t+1$ faults occur in a particular component, the component ceases to be reliable. Furthermore, the *system* remains operative as long no more than one component in any row is unreliable. By generalizing the techniques used to analyze the well-known "birthday surprise" problem of applied probability, we derive an approximation to the average number of faults needed until the systems fails. Applications include random access memory systems with chip-level and board-level coding as well as fault-tolerant systolic arrays.

* Supported in part by National Science Foundation grant NCR-8957623; also by the NSF Engineering Research Centers Program, CDR-8803012.

**Key Words**
Error control codes, fault tolerant systems, mean-time-to-failure, redundancy, reliability, systolic arrays.

## 1. Introduction

The two-dimensional array is a structure that pervades the study of information processing. Examples range from the conceptually simple – random access memory (RAM) arrays, in which each component is just a binary cell – to the complex, such as processor arrays.

In systems requiring a very high level of reliability, it is common to provide some kind of fault tolerance to such arrays. One way this can be achieved is by adding extra components and reconfiguring the array so as to logically remove the faulty cell(s). Reconfiguration techniques have been used to increase the yield of dynamic RAM chips since the introduction of the 64 K DRAM in 1979 [1]. Similar techniques for the design of highly reliable processor arrays have been extensively reported [2]-[3].

Another way that fault tolerance can be incorporated into such arrays is through the use of error control codes. For instance, it is common to arrange the chips making up a RAM system into an $M \times n$ array and form codewords from an $(n, k)$ code by taking one bit from each chip in a row of chips; as an example, if such a *board-level code* were capable of correcting any single error, then the system would continue to operate as long as no two errors "lined up" in the same positions on two different chips in the same row [4]-[5].

In some such systems, the demand for high reliability is so great that a second level of fault tolerance is needed; that is, each *component* in the array is required to maintain operation in the presence of faults. As an example, *on-chip* error control codes have been introduced as a means of increasing yield and enhancing reliability [6]-[7]; if both on-chip and board-level coding is used in a memory design, then there are two levels of fault tolerance. As another example, it has been suggested [8] that processor arrays requiring a very high degree of reliability can be organized as an "array of arrays", with fault tolerance at each of the two hierarchical levels. As a final example, the FTBBC computer designed at JPL for unmanned spacecraft allows not only faulty computers to be replaced with spares, but provides for internal redundancy so that memory modules, processors, etc., can be replaced within a computer before it is deemed "lost" [9].

This paper analyzes the reliability of such a "doubly redundant" system.

## 2. The Problem

Consider an $m \times n$ array of components (or *cells*). (See Figure 1.) Each component is fault tolerant in the sense that a particular component remains operative and reliable as long as no more than $t$ faults have occurred in that component; when $t+1$ faults have occurred in a cell, the cell ceases to be reliable. However, we assume that there is another, higher-level fault tolerance built into the array; as long as no more than *one* of the $n$ cells in any row is unreliable, the system continues to be operative.

We assume that when the system is placed in operation none of the $mn$ components contain faults. During the lifetime of the array faults occur uniformly over all of the components; that is, the first fault is equally likely to occur in any of the $mn$ components, as is the second fault, the third, etc. We assume further that the occurrence of the $i^{th}$ and $j^{th}$ faults are independent of one another for $i \neq j$.

The goal is to compute the average number of faults that can be tolerated until the system is unreliable – i.e., the average number of faults until two components in the same row have each suffered more than $t$ faults. We call such a number the *mean events to failure* (METF) of the system; furthermore, if the occurrence of faults in the components represent $mn$ independent Poisson processes, each with rate $\lambda$, then by Wald's identity the mean *time* to failure (MTTF) is given by

$$\text{MTTF} = \frac{1}{\lambda mn}\text{METF}. \tag{1}$$

To analyze this problem, we will employ techniques similar to those used by Klamkin and Newman in analyzing the well known "birthday surprise" problem.

In [10] Klamkin and Newman considered the following problem: Suppose you have a random experiment with $a$ equally likely outcomes; if you perform repeated, independent trials of the experiment, then what is the average number of trials until one of the outcomes has occurred $b$ times ? If we let $E(a, b)$ denote the desired expected value, then $E(365, 2) \approx 24$ is the solution to the "birthday surprise" problem, to wit: How many people, on average, do you need to interview before you find two who share a birthday ? (Blaum, Goodman, and McEliece [5] showed how results pertaining to the birthday surprise could be directly applied to the reliability of memory systems employing a single-error correcting board-level code.)

Klamkin and Newman showed that, for fixed $b$ and large $a$, $E(a, b)$ can be approximated by

$$E(a,b) \approx \sqrt[b]{b!}\,\Gamma\left(1+\frac{1}{b}\right) a^{1-(1/b)},$$

where $\Gamma(\alpha) \triangleq \int_0^\infty e^{-t} t^{\alpha-1} dt$. They accomplished this by representing the possible outcomes of the repeated trials as a polynomial and using a truncating operator to remove the terms of no interest; we will use a similar technique in analyzing our problem.

To draw the analogy between our problem and that considered in [10], we think of each component as a person; a fault occurring in a component is equivalent to the component being "interviewed". Each row of components in Figure 1 corresponds to a group of people who share a birthday; thus, we're assuming that there are $m$ days in the year, and for each day there are $n$ people who have that day as their birthday.

To model the occurrence of faults, we assume that we begin interviewing people randomly; there are $mn$ people and every person is equally likely to be interviewed each time. We say a person has died if that person has been interviewed more than $t$ times – i.e., a component fails if more than $t$ faults have occurred in that component, rendering it unreliable. The system fails when two different people with the same birthday have died – i.e., when two component in the same row have become unreliable.

The goal is to compute the average number of faults ("interviews") until the system is unreliable. To this end, define $P_{i,N}$ to be the probability that there are exactly $i$ component failures after $N$ faults; also, define $Q_{i,N}$ to be the probability that the system has not failed after $N$ faults *given* that exactly $i$ component failures have occurred. Then

$$\begin{aligned}
&E[\text{Number of faults until the system fails}] \\
&\quad = \sum_{N=0}^{\infty} P\{\text{System has not failed after } N \text{ faults}\} \\
&\quad = \sum_{N=0}^{\infty} \sum_{i=0}^{m} P_{i,N} Q_{i,N} \\
&\quad = \sum_{i=0}^{m} \sum_{N=0}^{\infty} P_{i,N} Q_{i,N}.
\end{aligned} \tag{2}$$

Since no configuration of $i$ component failures is more likely than any other, we conclude that

$$Q_{i,N} = \begin{cases} \dfrac{\binom{m}{i} n^i}{\binom{nm}{i}}, & \text{if } 0 \leq i \leq m; \\ 0, & \text{if } i > m. \end{cases} \tag{3}$$

To calculate $P_{i,N}$ requires substantially more work. We begin by making the following trivial observation: If we let $p_i$ denote the probability that, after $N$ faults, the *first* $i$ components – or any particular $i$ components – have failed and the other $nm - i$ components have not failed, then

$$P_{i,N} = \binom{nm}{i} p_i. \tag{4}$$

This (once again) follows from the observation that no choice of $i$ components are more likely to fail than any other choice. Therefore, our problem consists of computing $p_i$.

To compute $p_i$ we generalize some of the techniques used in [10]. Suppose a random experiment is repeated $N$ times; each trial of the experiment has $a$ equally likely outcomes. Then the polynomial

$$P_{a,N}(x_1, x_2, \ldots, x_a) \triangleq (x_1 + x_2 + \ldots + x_a)^N,$$

when expanded, contains one term for each of the possible sequences of $N$ outcomes. For example, if $a = 3$ and $N = 2$, then

$$\begin{aligned} P_{3,2}(x_1, x_2, x_3) = {} & x_1x_1 + x_1x_2 + x_1x_3 + x_2x_1 \\ & + x_2x_2 + x_2x_3 + x_3x_1 + x_3x_2 + x_3x_3. \end{aligned}$$

Here, the first term in the sum represents the case where the first outcome occurs twice, the second term represents the case where the first outcome is followed by the second outcome, and so on; the nine terms represent the nine possibilities of two trials with three outcomes per trial.

We now introduce the *truncation operator* $T_{k_1,k_2,\ldots,k_a}$; if we are given a polynomial $f(x_1, x_2, \ldots, x_a)$, then $T_{k_1,k_2,\ldots,k_a}\{f(x_1, x_2, \ldots, x_a)\}$ is just the polynomial obtained by deleting all of the terms in $f(x_1, x_2, \ldots, x_n)$ that contain a power of $x_i$ greater or equal to than $k_i$ for $i = 1, 2, \ldots, a$. Using this definition, for instance,

$$T_{2,1,2}\{P_{3,2}(x_1, x_2, x_3)\} = x_1x_3 + x_3x_1.$$

More generally, the polynomial

$$T_{k_1,k_2,\ldots,k_a}\{P_{a,N}(x_1,x_2,\ldots,x_a)\}$$

contains one term for each of the sequences of outcomes such that the $i^{th}$ outcome occurs fewer than $k_i$ times for $i=1,2,\ldots,a$. If we now evaluate this polynomial at $x_i = 1/a$ for $i=1,2,\ldots,a$, then the value so obtained represents the probability that, in a sequence of $N$ trials in which each trial has $a$ equally likely outcomes, the $i^{th}$ outcome occurs fewer than $k_i$ times.

Consider the following shorthand notation: For a polynomial $f(x_1,x_2,\ldots,x_a)$ let $d \triangleq \deg[f]+1$ and

$$R_{i,k}\{f(x_1,x_2,\ldots,x_a)\} \triangleq T_{\underbrace{d,\ldots,d}_{i},\underbrace{k,\ldots,k}_{a-i}}\{f(x_1,x_2,\ldots,x_a)\}.$$

Then $R_{i,k}\{f(x_1,\ldots,x_a)\}$ is obtained by deleting all the terms in $f(x_1,\ldots,x_a)$ with a power of $x_j$ greater than or equal to $k$ for all $j \in \{i+1,i+2,\ldots,a\}$; the powers of $x_j$ for $j \le i$ are not taken into account in determining which terms are kept. Finally, define

$$T(N,i) \triangleq R_{i,t+1}\{(x_1+x_2+\ldots+x_{nm})^N\}_{\frac{1}{nm},\frac{1}{nm},\ldots,\frac{1}{nm}}.$$

As defined, $T(N,i)$ is just the probability that (after $N$ faults) no failures have occurred in the last $nm-i$ components; that is, $T(N,i)$ is the probability that there are between 0 and $i$ component failures, and they all occur within the first $i$ components. This implies

$$T(N,i) = \sum_{k=0}^{i} \binom{i}{k} p_k,$$

where $p_k$ is defined as in equation (4) – namely, $p_k$ is the probability that, after $N$ cell failures, a particular $k$ components have failed and the other $nm-k$ have not.

**Lemma 1:** If we define $p_i$ as above, then

$$p_i = \sum_{j=0}^{i} \binom{i}{j} (-1)^j T(N,i-j).$$

*Proof:*

$$\begin{aligned}
\sum_{j=0}^{i} \binom{i}{j}(-1)^j T(N,i-j) &= \sum_{j=0}^{i}\sum_{k=0}^{i-j} \binom{i-j}{k}\binom{i}{j}(-1)^j p_k \\
&= \sum_{k=0}^{i}\sum_{j=0}^{i-k} \binom{i-j}{k}\binom{i}{j}(-1)^j p_k \\
&= p_i + \sum_{k=0}^{i-1}\left(\sum_{j=0}^{i-k}\binom{i-j}{k}\binom{i}{j}(-1)^j\right) p_k \\
&= p_i.
\end{aligned}$$

This last equality follows from the identity

$$\sum_{v=0}^{\min(a,n-r)} (-1)^v \binom{a}{v}\binom{n-v}{r} = \binom{n-a}{n-r}$$

for positive integers $r$ and $n$ [11, p.65]. QED

We can now combine equations (2), (3), and (4) together with the results of Lemma 1 to obtain:

$$E[\text{Number of faults until the system fails}] = \sum_{i=0}^{m}\sum_{j=0}^{i}\binom{m}{i}\binom{i}{j} n^i(-1)^j \left[\sum_{N=0}^{\infty} T(N, i-j)\right]. \quad (5)$$

So we need an approximation for the term in brackets.

**Lemma 2:** For any integer $k$ such that $nm - k$ is very large,

$$\sum_{N=0}^{\infty} T(N,k) \approx nm \sqrt[t+1]{(t+1)!}\,\Gamma(1+\frac{1}{t+1})(nm-k)^{-1/(t+1)},$$

where $\Gamma(\cdot)$ is the gamma function. (Here, "$\approx$" means that as $nm - k$ gets large, the ratio of the two sides goes to unity.)

**Proof:** We begin by considering the power series

$$\begin{aligned}
&\sum_{N=0}^{\infty} R_{k,t+1}\{(x_1 + x_2 + \ldots + x_{nm})^N\}\frac{\tau^N}{N!} \\
&\quad = R_{k,t+1}\{\sum_{N=0}^{\infty}(x_1 + x_2 + \ldots + x_{nm})^N \frac{\tau^N}{N!}\} \\
&\quad = R_{k,t+1}\{e^{(x_1+x_2+\ldots+x_{nm})\tau}\} \\
&\quad = e^{(x_1+x_2+\ldots+x_k)\tau} \prod_{i=k+1}^{nm} S_{t+1}(x_i\tau),
\end{aligned}$$

where

$$S_i(x) \triangleq \sum_{j<i} \frac{x^j}{j!}.$$

If we multiply each side of this expression by $e^{-\tau}$, use the identity

$$\int_0^\infty \frac{t^N}{N!} e^{-\tau} d\tau = 1,$$

and evaluate the resulting polynomial at $x_i = 1/nm$ for all $i$, then we obtain

$$\sum_{N=0}^{\infty} T(N,k) = \int_0^\infty e^{-(1-(k/nm))\tau}[S_{t+1}(\tau/nm)]^{nm-k} d\tau.$$

Making the change of variable $s = (1 - (k/nm))\tau$, we obtain

$$\sum_{N=0}^{\infty} T(N,k) = \frac{nm}{nm-k}\int_0^\infty e^{-s}[S_{t+1}(s/(nm-k))]^{nm-k} ds.$$

But in [10] it is shown that

$$\lim_{n\to\infty} \frac{1}{n^{1-(1/k)}} \int_0^\infty [S_k(t/n)]^n dt = \sqrt[k]{k!}\Gamma(1+(1/k)).$$

This implies the lemma. QED.

Since $i-j$ in equation (5) never gets larger than $m \ll nm$, we can use the result of the lemma to obtain the following approximation:

$$\begin{aligned}&E[\text{Number of cell failures until the system fails}]\\ &\approx nm \sqrt[t+1]{(t+1)!}\Gamma(1+\frac{1}{t+1}) \sum_{i=0}^{m}\sum_{j=0}^{i} \binom{m}{i}\binom{i}{j} n^i (-1)^j (nm-i+j)^{-1/(t+1)}.\end{aligned} \tag{6}$$

To simplify (6) we use a Taylor series expansion as follows:

$$\begin{aligned}(nm-i+j)^{-1/(t+1)} &= (nm)^{-1/(t+1)}\left(1-\frac{i-j}{nm}\right)^{-1/(t+1)}\\ &= (nm)^{-1/(t+1)} \sum_{k=0}^{\infty} \frac{1}{k!}\left(\prod_{\ell=0}^{k-1}(\ell+\frac{1}{t+1})\right)\left(\frac{i-j}{nm(t+1)}\right)^k\\ &= \frac{(nm)^{-1/(t+1)}}{\Gamma(1/(t+1))} \sum_{k=0}^{\infty} \frac{1}{k!}\Gamma(k+\frac{1}{t+1})\left(\frac{i-j}{nm(t+1)}\right)^k.\end{aligned}$$

If we then substitute the first $i+1$ terms of the Taylor series expansion into equation (6) and use the identity [11, p. 65]

$$\sum_{k=0}^{n}(-1)^{n-k}\binom{n}{k}k^r = \begin{cases} 0, & \text{if } r<n;\\ n!, & \text{if } r=n,\end{cases}$$

and the fact that $\Gamma(1+\alpha) = \alpha\Gamma(\alpha)$, then we arrive at the following alternate approximation:

$$\text{METF} \approx \frac{(nm)^{t/(t+1)} \sqrt[t+1]{(t+1)!}}{t+1} F(m, \frac{1}{t+1}), \tag{7}$$

where

$$F(m,\epsilon) \triangleq \sum_{i=0}^{m}\binom{m}{i} m^{-i}\Gamma(i+\epsilon).$$

## 3. Simulations

In this section we compare equation (7) with simulation results.

### 3.1 A Generic Example

We now compare equation (7) with a simulation of the precise system that it was meant to approximate – an $m \times n$ array of $t$-fault tolerating components that is reliable as long as no more than one component in any row has suffered more than $t$ faults.

In Figures 2-3 we've graphed the approximation in equation (7) for the METF of a $m \times n$ system where $m$ is held constant at 100 and 1000 and $n$ varies; Figures 4-5 show the results when $n$ is held constant and

$m$ varies. There is close agreement between (7) and results obtained via simulation, especially at high values of $n$ and $m$. (This is to be expected, since we can only use the results of Lemma 2 when $m(n-1)$ is very large.)

It should be noted that all of the curves representing equation (7) were generated in orders of magnitude *less* CPU time than it took to generate even one of the simulation points.

### 3.2. A Two-Level Coding Example

The 256K and and 1M dynamic RAM chips manufactured by Micron Technologies employ an on-chip (12, 8) single-error correcting Hamming code [7]; that is, each 1M chip contains $2^{17}$ 12-bit words, of which eight bits per word are data and four are redundancy. (The 256K chip contains $2^{15}$ such codewords.) The four bits of redundancy ensure that a codeword will be decoded correctly as long as no more than one error affects the codeword.

We now suppose that we use a chip employing such a (12, 8) chip-level code along with an $(n, k)$ single-error correcting *board-level* code; that is, the memory is made up of an $M \times n$ array of chips and gives correct results as long as there are not two bits failed in the same relative position on two different chips in the same row of chips. (See Figure 6.) We can approximate such a system as an $m \times n$ array of codewords, where $m = M \cdot N$ and $N$ is the number of codewords per chip; we furthermore make the assumption that the system fails when two codewords in the same row of this $m \times n$ array each suffer more than one error.

Note that this is a pessimistic assumption; when two errors occur in a (12, 8) codeword the decoder will make a third error, meaning three of the twelve bits in the codeword estimate are in error; therefore, it is quite possible to have two codewords each be decoded incorrectly and *still* not have two bit failures that "line up" with one another. (If one assumes that the locations of the three failed bits in each 12-bit codeword are uniformly distributed, then the probability that they *don't* line up is $\binom{9}{3}/\binom{12}{3} \approx 0.382$.) Essentially we are assuming that, when more than one bit in a chip-level codeword fails, then the whole codeword (or at least more than half of the codeword) has failed.

Figure 7 shows the approximation for $n = 39$ and varying values of $m$. (We picked $n = 39$ because it corresponds to a (39, 32) single-error correcting, double-error detecting extended Hamming code which might be used for a memory system with 32-bit words.) Also shown are simulation values which show the METF of such a system when bounded distance decoding is performed – i.e., when the decoder actually maps the retrieved 12-tuple onto the codeword that is at a distance of at most one away. As expected, the approximation is significantly lower than the simulation, owing to the pessimistic assumption; we find that equation (7) underestimates the METF by about 30%.

## 4. Implications of the Approximation

In this section we consider the implications of the approximation given by equation (7).

Suppose $m$ and $t$ are fixed; then equation (7) indicates that the METF increases like $n^{t/(t+1)}$ for increasing $n$. For systems in which the faults form independent Poisson processes, each with rate $\lambda$ – i.e., for systems where equation (1) applies – this means that the mean time to failure decreases like $n^{-1/(t+1)}$ for increasing $n$.

Alternatively, suppose $n$ and $t$ are fixed. Then equation (7) implies that the METF increases like $m^{t/(t+1)} F(m, \frac{1}{t+1})$. For reasonable values of $m$ – i.e., $m \leq 10^5$ – $F(m, \epsilon)$ grows approximately like a polynomial in $\log(m)$. For instance, the following formulae – generated by Mathematica – provide good approximations for $F(m, \frac{1}{2})$ and $F(m, \frac{1}{3})$ for $10 \leq m \leq 10^5$:

$$F(m, \frac{1}{2}) \approx 0.536 + 4.533x - 1.463x^2 + 0.413x^3 \big|_{x=\log_{10}(m)}$$

and

$$F(m, \frac{1}{3}) \approx 3.168 + 1.572x - 0.0514x^2 + 0.0922x^3 \big|_{x=\log_{10}(m)}.$$

Finally, suppose that the "row level" fault tolerance is obtained by spare switching; that is, in a row of $n$ components there are $k = n - 1$ that are in use at any given time. Suppose further that the total number of components that must be in use at any given time is fixed at a constant $c$, meaning $c = mk$; thus there is a total of $c + m$ components in the array, of which $m$ are redundant ($1 \leq m \leq c$). The *rate* of such a system is given by

$$R(m) = \frac{c}{c+m}.$$

Furthermore, if we once again assume that the faults occurring on the chips form $mn$ independent Poisson processes, each with rate $\lambda$, then the mean time to failure of such a system is given by

$$\mathrm{MTTF}(m) \approx \frac{\sqrt[t+1]{(t+1)!}}{\lambda(t+1)(c+m)^{1/(t+1)}} F(m, \frac{1}{t+1}).$$

Now it is obvious that $R(m)$ is a monotone decreasing function of $m$; that is, the highest rate is obtained with a single row of $n = k + 1$ components. It is less obvious that $\mathrm{MTTF}(m)$ is maximized by setting $m = c$ – that is, by having $c$ rows, each with one active and one spare component. One way to reconcile these two conflicting design considerations is to choose $m$ so as to maximize the rate while guaranteeing a specified MTTF – i.e., by defining

$$m(\tau) \triangleq \min\{m : \mathrm{MTTF}(m) \geq \tau\}.$$

Figure 8 shows $m(c\lambda\tau)$ for the case $t = 1$ and $c = 10^4$. We choose to normalize by $1/c\lambda$ because that's the mean time to failure of a system containing $c$ components when there is *no* fault tolerance; thus, $c\lambda\tau = 100$ means that $\tau$ is 100 times the MTTF of such an unprotected system. As an example, if $\lambda = 10^{-4}$ faults per hour, then the MTTF of a system with ten thousand cells with no redundancy is one hour. If a MTTF of 500 hours is required for the ten-thousand-cell system, then Figure 8 indicates that the smallest $m$ satisfying that condition is $m = 108$. However, since 108 does not divide $10^4$, our best design would be a $125 \times 81$ array; such an array would have a rate $R = 10000/10125 \approx 0.988$ and a MTTF of 517 hours.

## 5. Conclusion

The approximation derived in this paper offers a simple and low-computational way of estimating the METF (and thus the MTTF) of a wide variety of fault-tolerant systems.

## References


[1] R.P. Cenker *et al.,* "A fault tolerant 64K dynamic random access memory," *IEEE Transactions on Electron Devices,* vol. ED-26, pp. 853-860, June 1979.

[2] I. Korn (Editor), *Defect and Fault Tolerance in VLSI Systems,* Chapter 9: "Reconfigurable Arrays," Plenum Press, 1989.

[3] M. Chean and J. Fortes, "A Taxonomy of Reconfiguration Techniques for Fault-Tolerant Processor Arrays," *Computer,* pp. 55-69, January 1990.

[4] C. Chen and M. Hsiao, "Error-correcting codes for semiconductor memories: A state-of-the-art review," *IBM Journal of Research and Development,* vol. 28, March 1984.

[5] M. Blaum, R. Goodman, and R. McEliece, "The reliability of single-error protected computer memories," *IEEE Transactions on Computers,* vol. 37, No. 1, pp. 114-119, January 1988.

[6] J. Yamada *et al.,* "A sub-micron 1 Mbit dynamic RAM with a 4-bit-at-a-time built-in ECC circuit," *IEEE Journal of Solid State Circuits,* vol. SC-19, pp. 627-633, October 1984.

[7] J.E. O'Toole, T.M. Trent, and W.D. Parkinson, "256K dynamic error corrected RAM," memorandum, Micron Technologies.

[8] Y.-X. Wang and J. Fortes, "On the Analysis and Design of Hierarchical Fault-Tolerant Processor Arrays," *Defect and Fault Tolerance in VLSI Systems,* Plenum Press, 1989.

[9] D. Rennels, "Fault-tolerant computing – Concepts and examples," *IEEE Transactions on Computers,* vol. C-33, no. 12, December 1984.

[10] M.S. Klamkin and D.J. Newman, "Extensions of the birthday surprise," *Journal of Combinatorial Theory,* vol. 3, pp. 279-282, 1967.

[11] W. Feller, *An Introduction to Probability Theory and Its Applications (Volume I),* Third Edition, John Wiley and Sons, 1950.

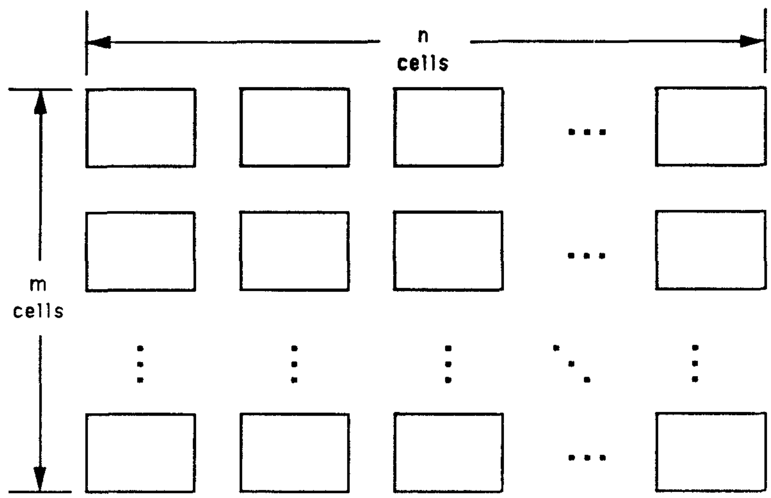


Figure 1: An m x n array of components. Each component is reliable as long as it has been afflicted by no more than t faults; the system is reliable as long as no row contains more than one unreliable component.

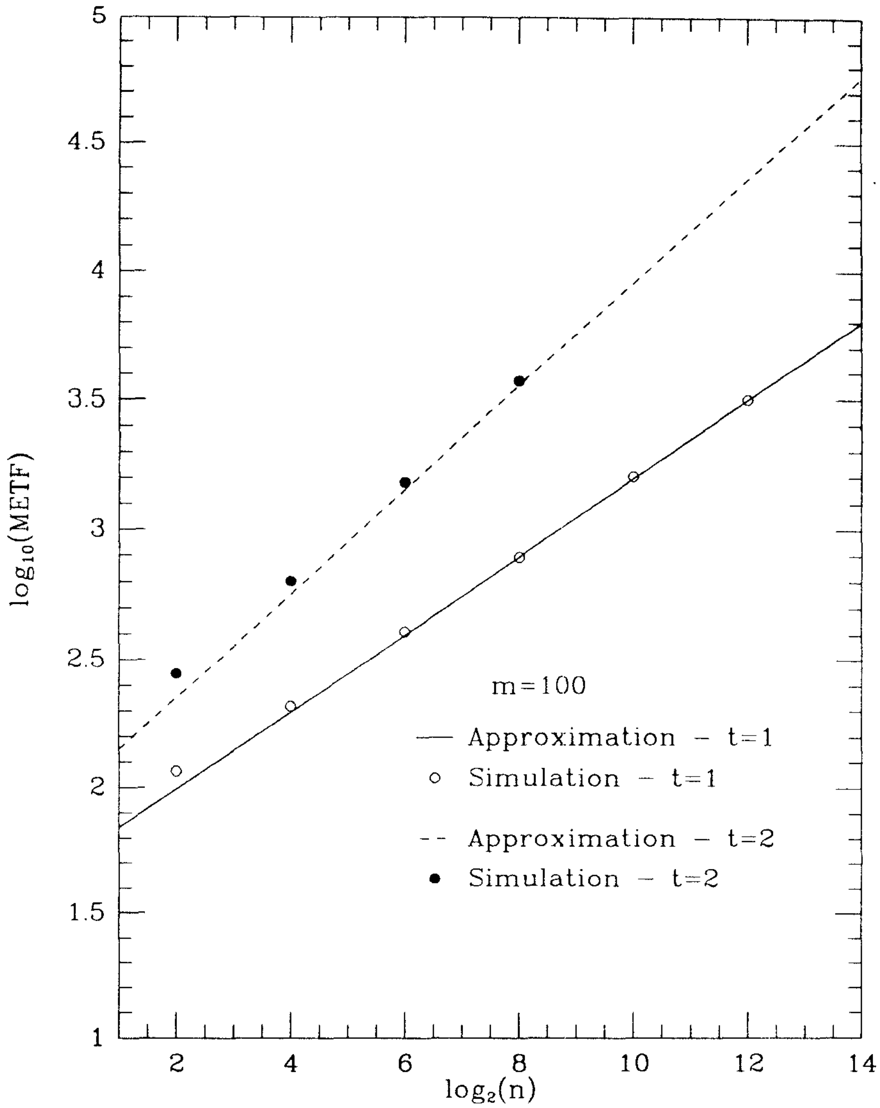


**Figure 2: The METF of a 100 x n array for varying n -- approximation and simulation.**

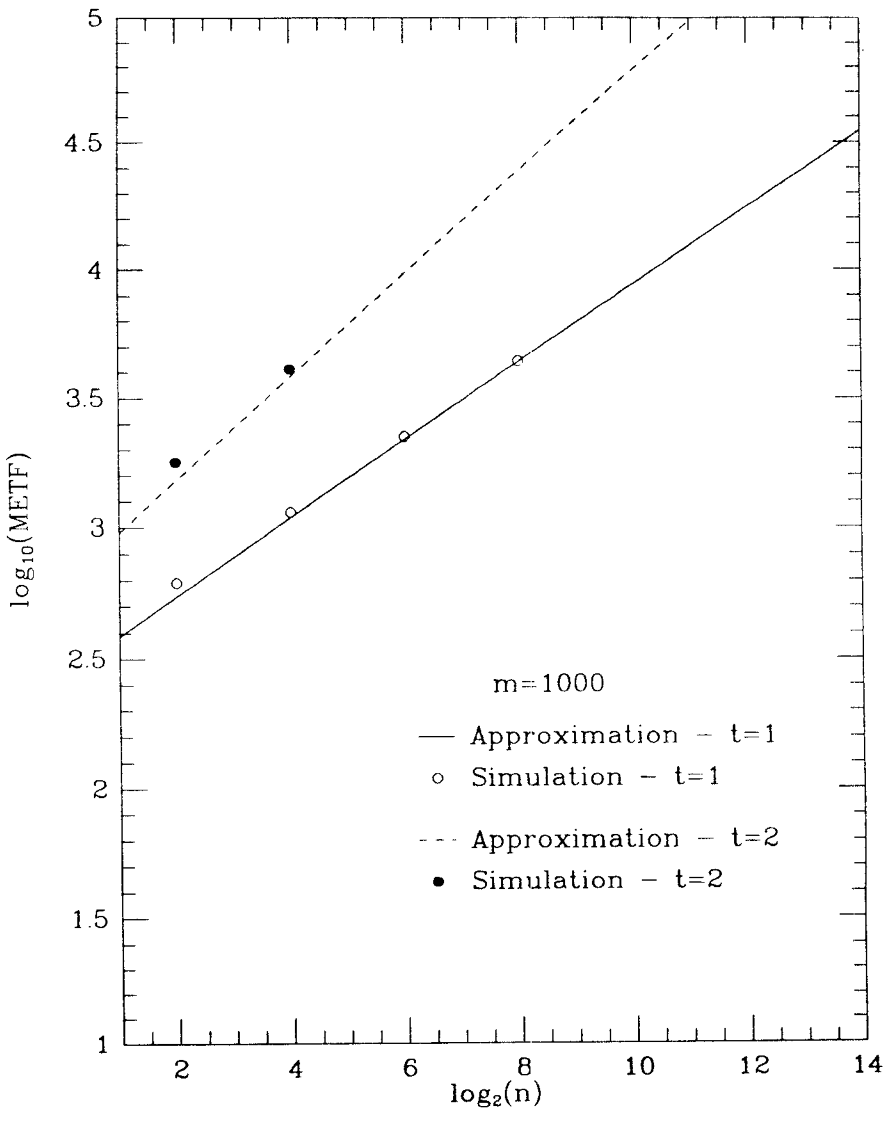


**Figure 3: The METF of a 1000 x n array for varying n -- approximation and simulation.**

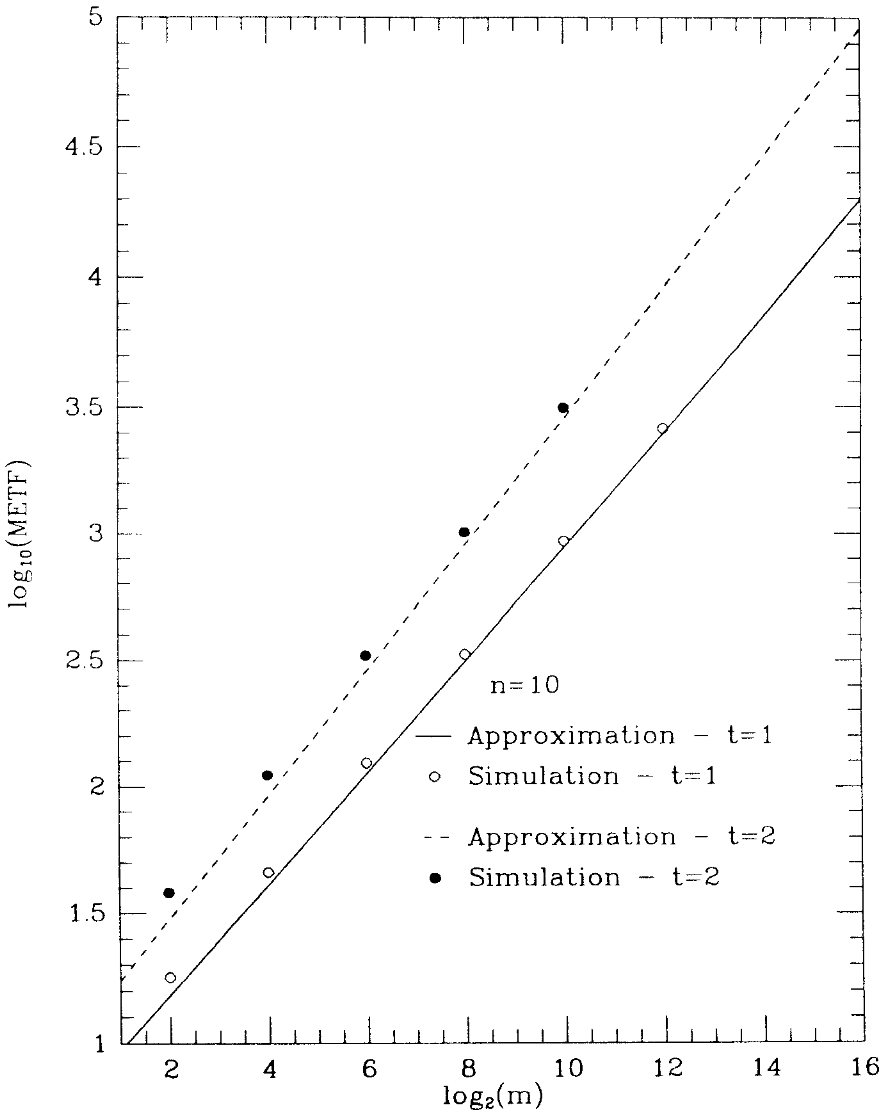


**Figure 4: The METF of an m x 10 array for varying m -- approximation and simulation.**

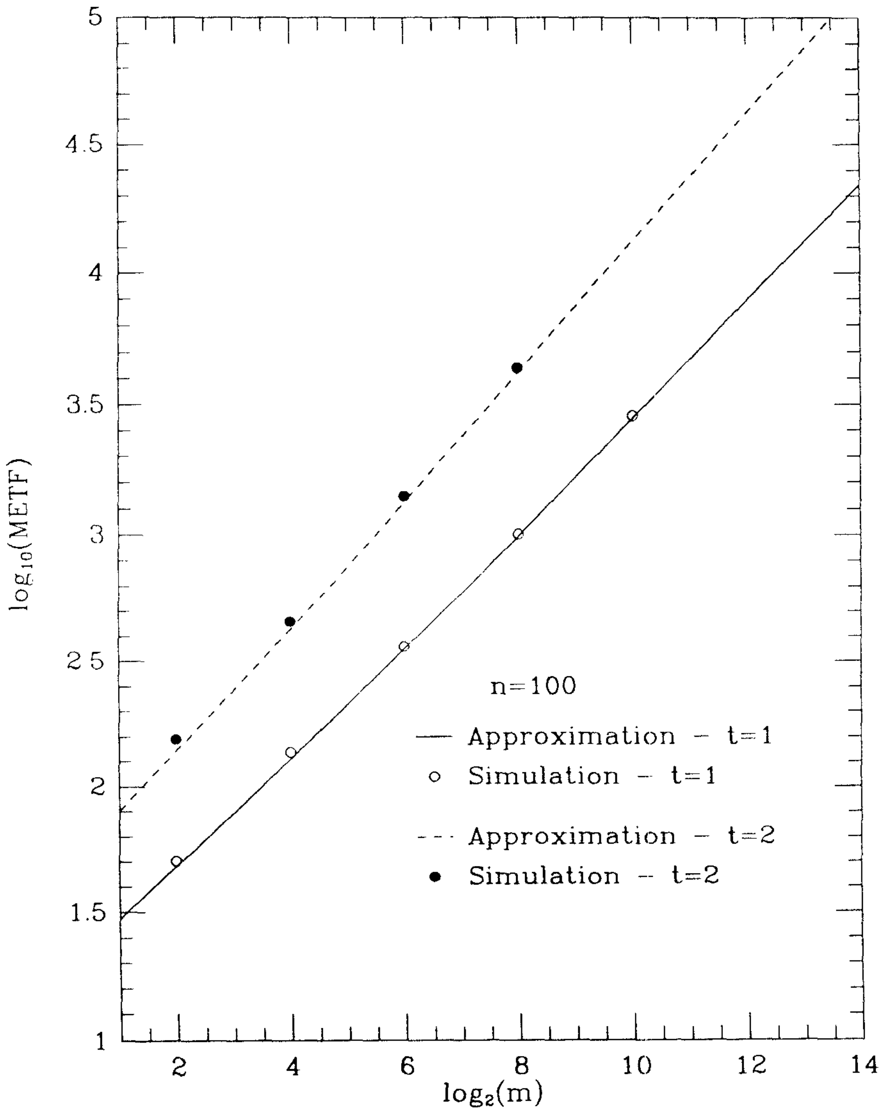


**Figure 5: The METF of an m x 100 array for varying m -- approximation and simulation**

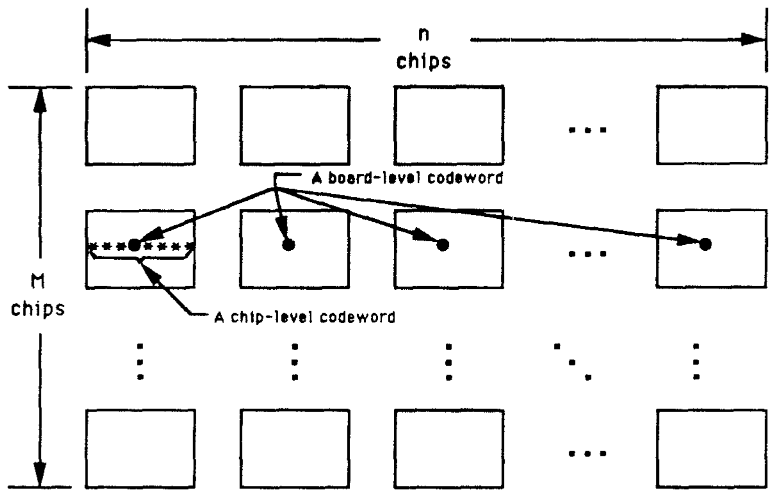


Figure 6: An **M** x n array of chips on a board; the stars indicate a chip-level codeword, the darkened circles a board-level codeword.

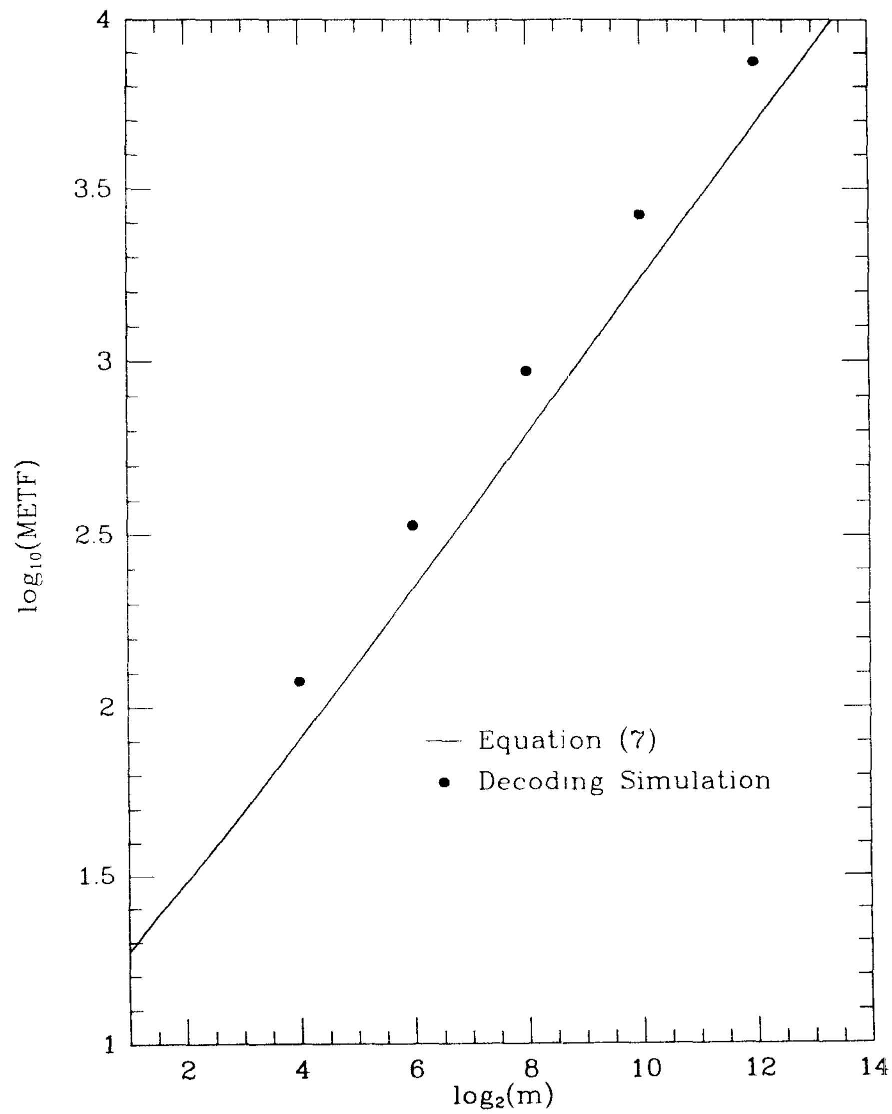


**Figure 7:** The METF of an m x 39 array of single-error correcting codewords; the solid line is computed from the (pessimistic) assumptions of equation (7), while the simulations assume bounded distance decoding.

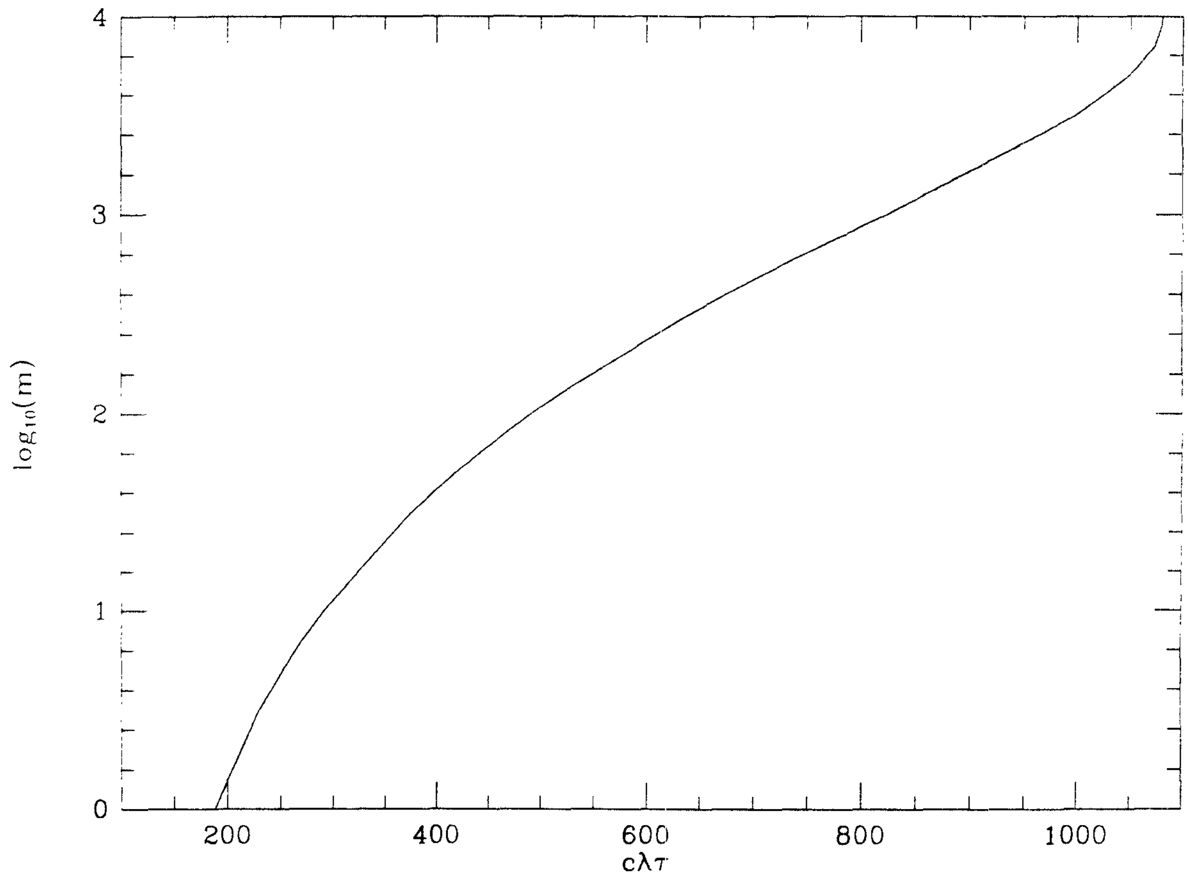


Figure 8: A plot of $m(c\lambda\tau)$ - the value of m that maximizes the rate of an m x n system subject to MTTF $\geq c\lambda\tau$. Here, n=k+1 and c=mk is constant at $c=10^4$.