SYSTEM EFFECTS OF SINGLE EVENT UPSETS

A. M. Finn
United Technologies Research Center
Silver Lane
East Hartford, CT 06108

Abstract

Single Event Upsets (SEUs) pose a serious threat to computer reliability and longevity. SEU effects are found at sea level, in airborne avionics, and in space. At the system level, SEUs in processors are controlled by replication and voting, watchdog processors, and tagged data schemes. SEUs in memory subsystems are controlled by periodically scrubbing words protected by an Error Control Code (ECC). The rate of memory scrubbing affects the performance and reliability of the entire computer system. There are tradeoffs between using radiation hardened semiconductors, scrubbing rates, and ECC capabilities. Previous tradeoff analyses have used simplified analytic models.

The system effects of SEUs may be evaluated by Markov modeling. Markov modeling has been extensively used for modeling processor redundancy; here it is also used for memory subsystems. A modeling methodology is presented which extrapolates chip transient and permanent failure rates to the system level, allows evaluation of alternative ECCs, and permits sensitivity analyses. The results for an example memory subsystem show that scrubbing effectiveness may be relatively insensitive to scrubbing rate.

1.0 Introduction

Radiation poses a serious threat to computer reliability and longevity. A single high-energy particle may cause a transient state change, i.e. a Single Event Upset (SEU), in computer memory or combinational logic circuits. SEU effects are found at sea level [24, 31], in airborne avionics [19, 20], and in space [1, 2, 5, 7, 10, 12, 14, 23, 24, 25, 26, 27, 28, 32]. The effect of SEUs on chips has been extensively investigated and programs, e.g. CREME [21], have been written to predict a chip's SEU rate. SEU effects may be minimized at the integrated circuit level by changes to the process technology. These changes tend to increase circuit area and decrease speed of operation.

At the system level, SEUs in processors are controlled by fault-tolerance techniques such as replication and voting, watchdog processors, and tagged data schemes [13,16,30]. SEUs in memory subsystems are controlled by use of error control codes (ECCs) [4,17,21] and a process called *scrubbing.* The scrubbing process periodically reads each word in the memory. If the number of faulty digits in a word is less than or equal to the number the ECC can correct, then the digits are corrected and the word is written back to memory. If the number of faulty digits exceeds the ECC's capability, the errors cannot be corrected and the memory has failed. Fault-tolerance to memory failures requires either physical redundancy via replication or temporal redundancy via checkpoint rollback schemes. In most aerospace applications physical redundancy is undesirable because mass, volume, and power are at a premium.

The rate at which SEUs are scrubbed from memory affects the performance and reliability of the entire computer system. Infrequent scrubbing leads to an accumulation of faults and increases the probability of exceeding the ECC's capability. Conversely, frequent scrubbing uses memory cycles that might otherwise be used by the operating system or an application program.

There is a recognized tradeoff between using ECCs and scrubbing or using lower density, higher power, radiation-hardened semiconductors to achieve reliability [7,32]. Previous analyses of the tradeoffs between the use of simple ECCs, the additional hardware for the ECC, failure due to that additional hardware, and the system impact have been based on simplified analytical models; detailed analytical models are intractable.

This paper introduces the idea of Markov modeling for SEU effects. Markov modeling allows extrapolation of chip failure rates to the subsystem and system level, allows more sophisticated tradeoff evaluations, and permits sensitivity analyses. The remainder of this paper is organized in three parts. Section 2 provides background about the SEU problem, expected SEU failure rates, and SEU control techniques. Section 3 introduces the use of Markov modeling techniques for memory subsystems and develops one model in detail. Section 4 presents the modeling results and generalizes the applicability of the modeling techniques.

Appeared in Computers in Aerospace VII Conference,
Monterey, CA, October 3-5, 1989

2.0 Background

Semiconductors, at sea level or in space, may encounter many different irradiating particles, e.g., photons (as gamma rays or x-rays), charged particles (as electrons, protons, alpha particles, or other ions), and neutrons. These irradiating particles cause ionization as they pass through a semiconductor. If the ionization deposits a charge that exceeds the noise margin of a circuit, an SEU occurs. An SEU is a state change in a memory cell or a logic circuit. Circuits with small noise margins, e.g. memory cells of small geometry, are particularly susceptible to SEUs. In general, SEUs are transient faults and do not cause permanent damage, although the total radiation dose causes a threshold voltage shift and eventual failure.

Both processors and memories may fail due to SEUs. High density, dynamic random access memory (DRAM) is particularly susceptible to SEUs. Static RAM, processor flip-flops, and combinatorial logic are also susceptible, but to a lesser extent. Hardware susceptibility to SEUs increases with decreasing device size, and with increasing radiation intensity.

2.1 SEU Fault Rates

Semiconductor memory is particularly susceptible to SEUs because it is composed of densely packed, minimum geometry devices. Protection of memory from SEUs is therefore of critical concern. Sufficient experimental and empirical evidence has been generated to gauge the occurrence rates and effects of SEUs. For instance, over a two-year period a total of 72 SEUs occurred in a satellite memory. The memory consisted of 48K bytes of 1K bit static RAM chips. Multiple faults occurred from single particles: 9 double faults, 1 triple fault, and 1 quadruple fault [5]. As another example, a one megabit NMOS DRAM in the 90% worst-case solar spectrum is expected to have upset rates between 0.3 /chip/hour and 17 /chip/hour [25].

The flip-flops and logic gates of a processor are also susceptible to SEUs. The evidence to date indicates that fault rates for microprocessors are commensurate with the rates of high-density RAM. For example, using the CREME program [2] the upset rate for several single-chip microprocessors was shown to be between 1.2 10^{-4}/hour and 8.4 10^{-4}/hour (784 km 98° orbit, 1g/cm^2 Al shielding, solar minimum weather). For the worst case solar flare, the rates increased to between 3 and 18 upsets per hour [12]. As another example, the on-chip RAM of the INMOS Transputer was found to contribute almost 95% of the observed SEUs. Even if a Transputer had a protected off-chip RAM, the expected fault rate is 1.5/day in the processor during a worst-case solar flare (900 km 98° orbit) [28].

The composite worst-case solar flare particle environment imposes an extreme demand on aerospace computer design. Solar flares may increase the nominal upset rate by three or four orders of magnitude [22]. These events are at their worst for only a few hours, but may be above nominal for a few days [1]. Further, there are factors of two to three uncertainty in our knowledge of the basic interplanetary heavy ion environment and the benefits of geomagnetic shielding [1,22]. As memory density increases, faults due to direct ionization by protons in the radiation belts may become the limiting factor on space mission longevity.

2.2 SEU Control

Single event upsets may be controlled by radiation hardening (a fault-avoidance technique) or by radiation tolerance (a fault-tolerance technique). Radiation hardening techniques typically involve integrated circuit process changes. Shielding, a fault-avoidance technique may be counter productive due to bremsstrahlung radiation and nonlinear energy deposition rates [14]. Fault-tolerance techniques include the use of ECCs for memories and processor registers; replication with voting for ALUs and processors; and watchdogs, checkpoint-rollback, and memory reloading for software execution.

An example process change for radiation hardness is the addition of cross-coupled resistors in CMOS memory cells [3]. Device hardness may be increased by reducing the gate oxide thickness, by making the p-well as small as possible, and by increasing the size of metal interconnections. All of these techniques have drawbacks in chip area and speed of operation.

As opposed to SEU fault-avoidance, where the fault occurrence is postponed, it is possible to apply fault tolerance techniques to handle the inevitable failures. Of particular interest for memories is the use of ECCs. An ECC is described by the triple (n,k,d), where d is the minimum Hamming distance between all pairs of codewords. ECCs incorporate redundancy by encoding k digits of information (a data word) into an n digit codeword, where $k < n$. Depending on the amount of redundancy, the code may correct c errors and detect an additional d errors. The codewords, rather than the data words, are stored in memory. As a codeword is read, a decoder uses the redundancy to detect any errors or to reconstruct the original data word by correcting any errors. The code fails when the decoder does not detect an error or incorrectly corrects an error.

An ECC provides protection for some prespecified number of faults. However, faults may occur and accumulate sufficiently to defeat the code, depending on the radiation environment and the frequency of accessing a specific word in memory. A fault which has not been detected is called latent. If the number of latent faults in a word are correctable they are called passive latent faults. More latent faults than this are called active latent faults; they will cause an error when the word is read. Fault latency may be controlled by periodically accessing all memory locations, correcting faults if necessary. The periodicity required is determined by reliability modeling and simulation as demonstrated in the following sections.

The on-chip use of ECCs has been proposed to reduce the apparent SEU susceptibility. For example, using 256 Kbit DRAMS with an on-chip single-bit ECC and scrubbing the entire memory in 1.25 seconds, it was possible to scrub the effects of up to 5 particles /cm^2/second. An on-chip ECC is not sufficient by itself, there were some events, such as pull down of whole bit lines, that affected many bits simultaneously and many of the single ion tracks affected multiple bits [32].

Memory chips may fail in complex manners and for reasons other than SEUs. Empirical data suggests that a permanent fault rate for a chip is between approximately 10^{-7}/hour and 10^{-8}/hour [18]. When there is a significant probability of an entire chip failing, ECCs are developed over *digits* of the width of the chip [6,9]. That is, ECCs for digits are developed by working over the Galois field GF(2^w), where a digit is *w* bits. For example, there is a (17,15,3) code over GF(2^4) which can be shortened to a (10,8,3) code. The (10,8,3) code encodes 8 digits, of 4 bits each, into 10 digits, of 4 bits each. This digit correcting code is distance 3 over GF(2^4). It can correct a word with 1 faulty digit or detect (but not correct) a word with 2 faulty digits. With only n-k=2 redundant digits (8 redundant bits) it is not possible to develop a code that can correct single digits and detect double-digit errors.

Other redundancy techniques are possible for SEU control. At a higher level than simply protecting the memory, N-modular redundancy and voting may detect (mask for *N > 2)* single event upsets. If fault-tolerance is mandated for a control system, then single event upsets may already be controlled provided that the system fault-model includes the manifestation of SEUs.

The effects of SEUs on program flow in TI's SBR9000 microprocessor, ignoring the effect of SEUs on data, are 96% detectable by illegal opcode detection, invalid opcode address detection, invalid read/write address detection, and unused/nonexistent memory detection [16]. The effects of SEUs on program flow, ignoring the effect of SEUs on data, is 85% to 90% detectable by containment set modifications to `typical' control system software and addition of a watchdog timer [13,30].

3.0 System Modeling

The applicability of Markov modeling to evaluating processor redundancy and fault-tolerance techniques has been extensively studied, see e.g. Trivedi [29]. This section introduces the use of Markov modeling techniques for SEUs in memory subsystems. A Markov model is a directed graph where the nodes represent system states and the arcs represent transition rates between the states. The probability of a state transition for a pure Markov model is assumed to depend only on the current state. This is equivalent to assuming that failure rates are constant and that failure occurrence is a Poisson process.

A Markov model has been developed for a generic memory subsystem. The generic memory subsystem is assumed to be RAM, M_w chips wide by M_d chips deep as shown in Figure 1. Each chip is C_w bits wide by C_d bits deep as shown in Figure 2. There are M_w digits of C_w bits each in a word. If a chip is organized *N* x 1, the digit is one bit; if organized *N* x 4, the digit is a nibble; if organized *N* x 8, the digit is a byte. Because chips may fail for reasons other than SEUs, the memory is assumed to be protected by a digit ECC capable of correcting *c* errors and detecting *d* errors in units of C_w.

The Markov model is shown in Figures 3 and 4. The states of the Markov model are labeled with the two tupple (h, s) representing the number of permanent (hard) and transient (soft) faults in the memory. These are latent faults, i.e. probabilistically no word has more faults than the ECC can correct. When a new fault occurs, there is some probability that a word will have more faults than the ECC can correct, i.e. a latent fault has become active. When this happens there is a transition (shown in to upper right of Figure 4, but not in Figure 3) to a failure state. The assumption of immediate transition is conservative. An active latent fault does not cause an error until the faulty word is read.

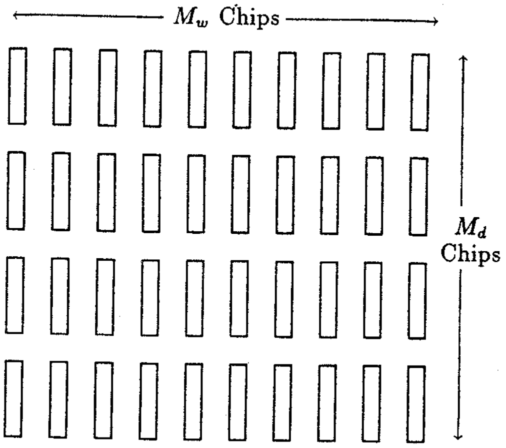

Figure 1: Memory Board Array of Chips

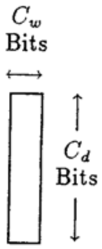

Figure 2: Chip Dimensions

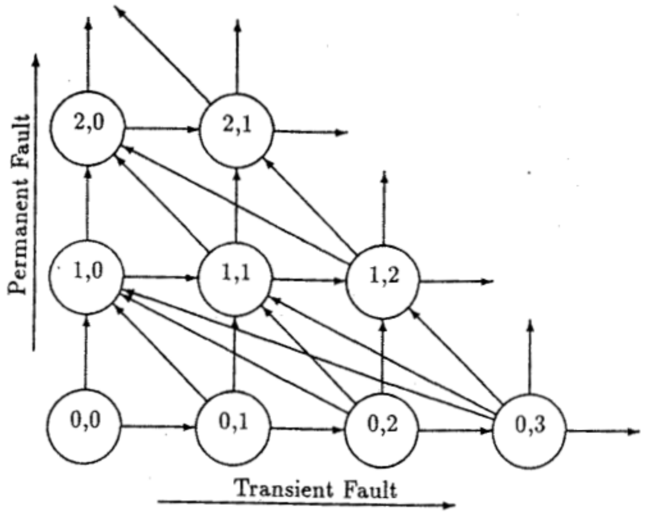

Figure 3:Memory Subsystem Markov Model
(Without Recovery or Failure Transitions)

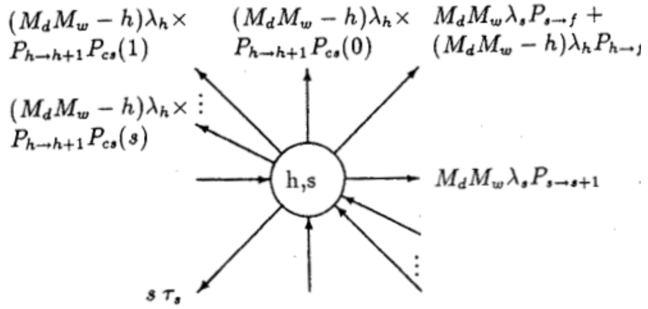
$(M_dM_w - h)\lambda_h\times$ $P_{h\to h+1}P_{cs}(1)$
$(M_dM_w - h)\lambda_h\times$ $P_{h\to h+1}P_{cs}(0)$
$M_dM_w\lambda_sP_{s\to f}$ + $(M_dM_w - h)\lambda_hP_{h\to j}$
$(M_dM_w - h)\lambda_h\times$ ⋮ $P_{h\to h+1}P_{cs}(s)$
h,s
$M_dM_w\lambda_sP_{s\to s+1}$
$s\,\tau_s$
⋮

Figure 4: State Transition Rates

In the model, horizontal transitions to the right occur when a digit, not previously faulty, is affected by an SEU. Transitions to a state with one less transient error (shown as diagonal transitions to the lower left in Figure 4, but not shown in Figure 3) occur when transient faults are scrubbed. When an SEU occurs it may further corrupt an already faulty digit or it may corrupt a new digit. (Conservatively it is assumed that subsequent faults do not repair the effects of previous faults.) If the fault corrupts an already faulty digit, there is no change to the system, and therefore no change to the model. If a new digit is corrupted, the memory may accumulate a new faulty digit, or a passive latent fault may become active, as when the capability of the ECC is exceeded.

In the model, vertical transitions occur when a permanent fault occurs. A permanent fault may mask previous SEUs, therefore the vertical transitions may be diagonal to the upper left. That is, if a permanent fault occurs in a chip that has j transient faults, then the next state will be that of one more permanent fault and j fewer transient faults.

In the model, the transient and permanent faults are known to be independent because permanent faults supersede transient faults. Further, the sum of all faults in a word are less than or equal to the correctable number, c, because the transitions into a non-failed state are conditioned by the probability that the number of faulty digits is less than c.

The detailed Markov model state transitions are shown in Figure 4. The probability of a digit, word, chip, etc. being faulty is the probability mass function of a Bernoulli random variable. For instance, given *s* SEU induced faulty memory digits, the probability that a given word contains *k* faulty digits, $P_{ws}(s\ k)$, is

$$P_{ws}(s,k) = \binom{s}{k}\left(\frac{1}{M_dC_d}\right)^k\left(1-\frac{1}{M_dC_d}\right)^{s-k}$$

$$0 \le k \le min(M_w, s)$$

where M_dC_d is the number of words in the memory. Given that a word contains *k* SEU faulty digits, the probability that a given digit contains *j* faults, $P_{ds}(k, j)$, *is*

$$P_{ds}(k,j) = \binom{k}{j}\left(\frac{1}{M_w}\right)^j\left(1-\frac{1}{M_w}\right)^{k-j}$$

$$0 \le j \le 1$$

where M_w is the number of digits in a word and j is limited to 0 or 1 because any number of SEUs in a digit means that the digit is faulty.

The probability that a given chip contains *k* SEU induced faulty digits, $P_{cs}(s\ k)$, is

$$P_{cs}(s,k) = \binom{s}{k}\left(\frac{1}{M_wM_d}\right)^k\left(1-\frac{1}{M_wM_d}\right)^{S-k}$$

$$0 \le k \le min(C_wC_d, s)$$

where M_wM_d *is* the number of chips in the memory. Finally, given *h* permanently faulty chips, the probability that a given word contains *k* permanent faults $P_{wh}(h, k)$, is

$$P_{wh}(h,k) = \binom{h}{k}\left(\frac{1}{M_w}\right)^k\left(1-\frac{1}{M_w}\right)^{h-k}$$

$$0 \le k \le min(M_w, h)$$

where M_w is the memory width in chips.

The Markov model state transition equations, Figure 4, may be derived from the definitions of $P_{ws}(s\ k)$, $P_{ds}(k\ j)$, $P_{cs}(s\ k)$, and $P_{wh}(h\ k)$. For example, SEUs occur at the rate $M_dM_w\lambda_s$, where λ_s is expressed per chip per unit time. The probability that the memory accumulates a new SEU, $P_{s\text{->}s+1}$, is the probability that for any possible number of permanent faults in the word the digit affected by the SEU does not have a permanent fault; and that for any number of transient faults in the word, such that the word has less than c total faults, the digit affected by the SEU is not already faulty because of an SEU. The other state transition equations are derived similarly.

$$
\begin{aligned}
P_{s\to s+1} &= \sum_{i=0}^{h} P_{wh}(h,i)\, P_{dh}(i,0) \sum_{\substack{j=0\\ i+j<c}}^{s} P_{ws}(s,j)\, P_{ds}(j,0)\\
P_{s\to f} &= \sum_{i=0}^{h} P_{wh}(h,i)\, P_{dh}(i,0) \sum_{\substack{j=0\\ c\leq i+j}}^{s} P_{ws}(s,j)\, P_{ds}(j,0)\\
P_{h\to h+1} &= \sum_{i=0}^{h} P_{wh}(h,i) \sum_{\substack{j=0\\ i+j<c}}^{s} P_{ws}(s,j)\\
P_{h\to f} &= \sum_{i=0}^{h} P_{wh}(h,i) \sum_{\substack{j=0\\ c\leq i+j}}^{s} P_{ws}(s,j)
\end{aligned}
$$

By definition,

$$P_{s\to s+1} + P_{s\to f} = 1$$

and

$$P_{h\to h+1} + P_{h\to f} = 1.$$

3.1 Modeling Assumptions

A number of conservative assumptions have been made in developing this model. The assumption of immediate transition to a failure state is conservative because an error does not occur until a faulty word is read. Theoretically, execution of the operating system and application programs may scrub SEUs from a memory. If a faulty word is never read, or is written over, the memory subsystem would not actually fail. However, analysis of memory access patterns indicate a highly irregular distribution and long fault latencies [11]. Over the time scales of interest, a small portion of the memory will be accessed extensively, a small portion will not be accessed at all, and a large portion will be accessed infrequently. Memory locations that are infrequently accessed by normal operation may contain active latent faults which will cause an error when read.

A permanent fault is assumed to cause all of the bits in an entire chip to be faulty. This is a conservative assumption since single bits in a chip may suffer permanent faults (the Voyager 11 is a case in point [15]). Removal of this assumption is discussed in Section 4.1.

An assumption has been made that each transient fault affects only one digit and that the faults are uniformly distributed in the memory. Removal of this assumption is discussed in Section 4.1.

An assumption has been made that scrubbing is effectively a continuous process. That is, if there are s. SEUs in the memory, on average one SEU will be encountered in *1/s* of the memory scrub time. Therefore one SEU will be removed at s times the scrub rate, where the scrub rate is the inverse of the scrub time. The scrub rate is expressed as number of complete passes through memory per hour. If the scrub rate is much larger than the SEU rate, then scrubbing might be modeled as a batch process. Variations on modeling the scrubbing process are discussed in Section 4.1.

4.0 Results

Programs have been written to automatically generate the Markov model of Figure 3 with the transition rates of Figure 4. The program output is a model description in the input syntax of the Markov model solver PAWS (Pale Approximation With Scaling) from NASA Langley Research Center [8]. The program input includes the maximum number of transient and permanent errors (this bound is later checked to ensure numerical accuracy); the memory organization M_w, M_d, C_w, C_d; and the ECC capability *c*, and *d*. The results are produced symbolically in λ_s, λ_h, and the scrubbing rate τ_s.

Reliability analyses have been performed for various ECCs by allowing time to vary for fixed values for λ_s, λ_h, and τ_s. Sensitivity analyses have been performed for various ECCs by fixing a value of time and individually varying for λ_s, λ_h, and τ_s. The sensitivity analysis demonstrates the tradeoffs possible between the additional memory chips for various ECCs, the increased probability of failure due to those additional chips, and the effectiveness of scrubbing.

A crucial question in modeling high-reliability systems is that of model validation. In the results reported here, the programs, mathematics, and models are simple enough that they may be verified by exhaustive testing and inspection. As with any modeling, however, the results are only as good as the input parameters and model assumptions. In this case the critical parameters are the time it takes to scrub the memory, τ_s the SEU fault rate per chip, λ_s and the permanent fault rate per chip, λ_h.

Consider the case of a memory composed of 64K x 4 static RAM chips ($C_d = 2^{16}$, $C_w = 4$). Suppose the memory is 256K words deep and wide enough for a 32-bit word plus ECC redundancy ($M_d = 4$, $M_w > 8$). From the results of Section 2.1, SEU bit error rates (BERs) between 10^{-9}/day and 10^{-4}/day have been used. This is λ_s between approximately 10^{-5}/chip/hr and 1/chip/hr. From MIL-HDBK-217E, the expected permanent failure. rates, λ_h, are between approximately 10^{-7}/chip/hr and 10^{-6}/chip/hr [18]. Memory scrubbing is only desirable if it does not interfere excessively with other computations. The range of interest is between once per hour and once per second.

To limit the state space of the model, an upper bound was assumed on the maximum number of permanent and transient faults that would ever be simultaneously present in the memory. This bound was arbitrarily set at ten and fifty, respectively. This assumption was verified by investigating the probability that the model was ever in a state with the maximum number of faults. This probability was numerically negligible for all of the results reported here.

The probability of failure as a function of time for various ECCs is shown in Figure 5. For this simulation, the bit error rate was 10^{-4}/day, i.e. λ_s = 1.09/hour; the permanent failure rate was 10^{-6}/hour; and the scrubbing rate was 60/hour, i.e. one complete pass through memory each minute. The reliability improvement from using double and triple digit ECCs at this scrubbing rate is obvious.

Surprisingly, as shown in Figure 6, the memory reliability is relatively insensitive to the scrubbing rate (10 hour mission time, λ_s and λ_h as before). The diminishing returns for faster scrubbing may be seen from left to right in Figure 6. Changing the scrubbing rate from 7 to 700, the single digit ECC memory reliability improves a factor of 27; the double digit ECC memory reliability improves a factor of 10; the triple digit ECC memory reliability improves a factor of 4. The better the ECC, the slower the scrubbing needed to achieve the maximum reliability. Very low scrubbing rates have not been simulated where the numerical error due to the model bounds is significant.

A sensitivity analysis, varying the permanent fault rate, for a single digit ECC, is shown in Figure 7 (1 minute scrubbing, other parameters as before). The graph may be divided into three regions: above the left dashed line the probability of failure is dominated by double transient faults, $P_{fail} \propto \lambda_s^2$, below the right dashed line the probability of failure is dominated by double permanent faults, $P_{fail} \propto \lambda_h^2$,and between the dashed lines both permanent and transient faults contribute to the probability of failure $P_{fail} \propto \lambda_s. \lambda_h$.

The same sensitivity analysis as Figure 7, but for a double digit ECC, is shown in Figure 8. Again, the graph may be divided into three regions: above the left dashed line the probability of failure is dominated by triple transient faults, $P_{fail} \propto \lambda_s^3$, below the right dashed line the probability of failure is dominated by triple permanent faults $P_{fail} \propto \lambda_h^3$ and between the dashed lines the dominant failure mode is two permanent failures and one transient failure $P_{fail} \propto \lambda_s \lambda_h^2$. A distinct region where $P_{fail} \propto \lambda_s^2 \lambda_h$ is not found; it is an unlikely failure mode.

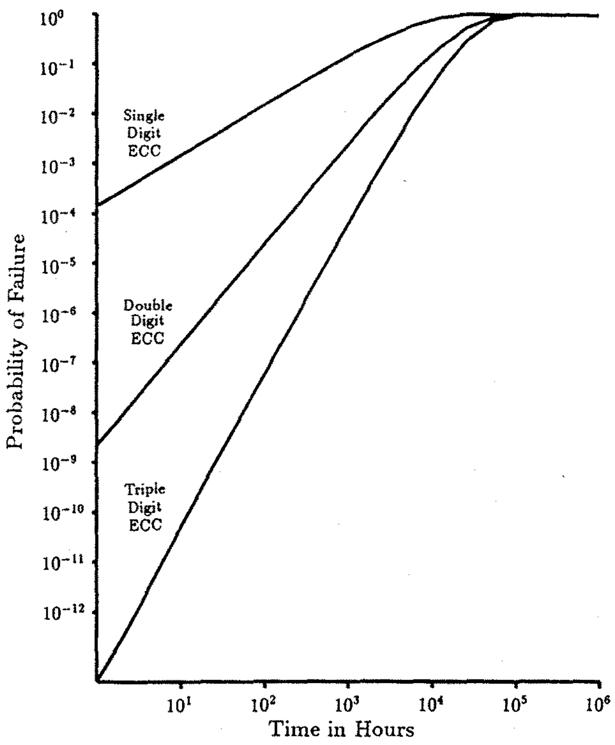

Figure 5: Probability of Failure vs. Time

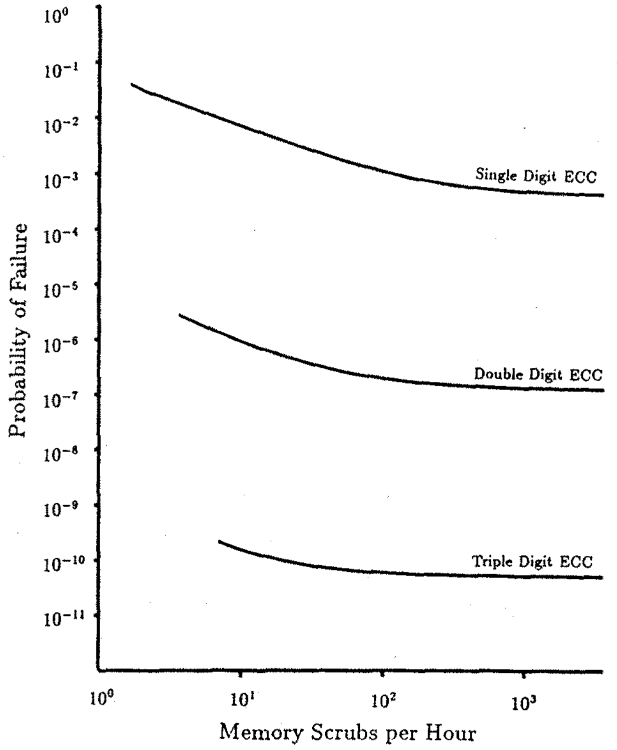

Figure 6: Scrubbing Sensitivity for Several ECCs

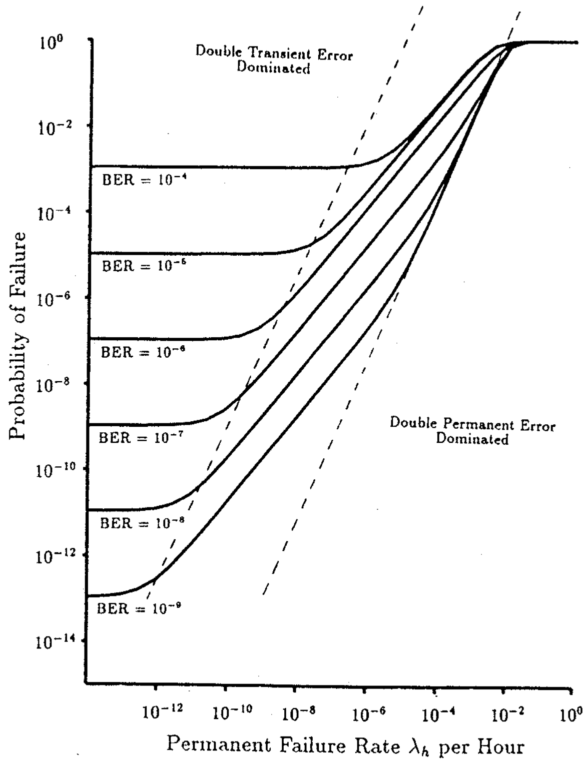

Figure 7: Failure Modes, Single Digit ECC

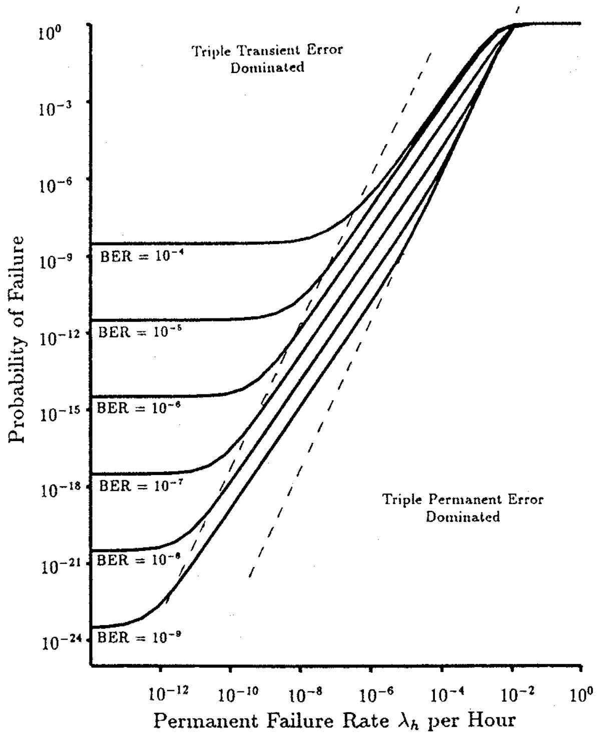

Figure 8: Failure Modes, Double Digit ECC

4.1 Model Extensions

There are several possible extensions and future work associated with this study. The present model is predicated on single digit faults resulting from ionizing radiation. However, an SEU is not necessarily a single upset; for instance, multiple bit faults occur due to single ionizing particles [5,32]. Another failure mode in high-density memories which leads to multiple bit faults is alpha particle induced charge transfer between cells [10]. The present model might be extended to analyze this, provided that realistic multiple digit fault rates could be established. The present model also assumes that permanent faults affect the entire memory chip. The model could be extended to remove this restriction if the permanent fault rates could be decoupled into individual digit fault rates, row or column fault rates, and entire chip fault rates.

The assumption of a constant rate scrubbing process may not be realistic depending on the implementation. A fast, infrequent scrubbing might be better modeled, depending on the actual mechanism, as a batch process where faults accumulate and then are all removed at once. In this case, the recovery transitions would be back to the zero transient fault state. It is simple to change the model to study this mode of operation.

In some cases it might be of interest to distinguish detectable from undetectable memory failures, provided the ECC has correctability c strictly less than detectability d. Two types of failure states would be required, one for detected, but uncorrectable, errors and another for undetected incorrect correction. The fault occurring state transitions would not be conditioned by the probability of a word having less than c errors. The transition to the failure state, detected or undetected, would occur when the scrubbing process (or other program activity) read a word with between c and d or greater than d faults, respectively. It would be necessary to probabilistically characterize program and operating system activity.

The memory modeling, or simulation results, demonstrated here could be incorporated into a system Markov model. The two-dimensional memory model could be extended to *n* dimensions for redundant processors, buses, etc. The transitions between the new states would depend on the system fault-tolerance techniques. A disadvantage of this approach is explosion of the number of model states. A more appropriate, hierarchical approach would be to model the system with the memory as a single device whose failure rate was determined by simulations such as those presented here.

5.0 Conclusions

A methodology for the reliability modeling of memory subsystems in a radiation environment has been introduced. It is based on probabilistically conditioning the state transitions of a Markov model. The methodology allows tradeoff analyses for the use of Error Control Codes (ECCs) and scrubbing to control radiation induced Single Event Upsets (SEUs). The methodology is sufficiently general that it may model (or may be easily extended to model) many different memory organizations, both logical and physical. The simulation results may be hierarchically incorporated into system level Markov models.

A model for a hypothetical 256K word memory was developed to demonstrate the modeling technique. The simulation results showed that large reliability improvements are possible from using double and triple digit ECCs, provided that scrubbing is rapid enough. The results also showed that memory reliability may be relatively insensitive to the scrubbing rate, e.g., a 2 orders of magnitude change in scrubbing rate for a triple digit ECC resulted in only a factor of 4 change in reliability. Sensitivity analyses demonstrated that, for a given set of parameters, one failure mode may dominate the probability of memory failure.

The software tool PAWS [8] in combination with custom model generation programs has been found to be very efficient and useful for high-level reliability modeling, analysis, and evaluation.

6.0 References

[1] J. Adams and A. Gelman, `The Effects of Solar Flares on Single Event Upset Rates', IEEE Trans. Nuc. Sci., v. NS-31, n. 6, Dec. 1984, pp. 1212-1216.

[2] J. Adams, Cosmic Ray Effects on Microelectronics, Part IV, NRL MR 5901, Dec. 31, 1986.

[3] J. Andrews, et al., `Single Event Upset Error Immune CMOS RAM', IEEE Trans. Nuc. Sci., v. NS29, Dec.. 1982.

[4] R. Blahut, The Theory and Practice of Error Control Codes, Addison-Wesley, revised, 1984.

[5] J. Blake, and R. Mandel, `On-Orbit Observations of Single Event Upsets in Harris HM-6508 1K RAMS', IEEE Trans. Nuc. Sci., v. NS-33, n. 6, Dec. 1986, pp. 1616-1619.

[6] D.C. Bossen and M.Y. Hsiao `A System Solution to the Memory Soft Error Problem', IBM J. Res. Dev., v. 24, n. 3, May 1980, pp. 390-397.

[7] J. Browning, et al., `Single Event Upset Rate for a 16-K CMOS SRAM', IEEE Trans. Nuc. Sci., v. NS-32, n. 6, Dec. 1985, pp. 4133-4139.

[8] R.W. Butler and P.H. Stevenson, The PAWS and STEM Reliability Analysis Programs, NASA-TM100572, Mar. 1988.

[9] C. Chen and M. Hsiao, `Error-Correcting Codes for Semiconductor Memory Applications: A State-of-the-Art Review', IBM J. Res. Develop., v. 28, n. 2, March 1984, pp. 124-134.

[10] J-S. Chern, et al., `Alpha-Particle-Induced Charge Transfer Between Closely Spaced Memory Cells', IEEE Trans. Elect. Dev., v. ED-33, n. 6, June, 1986, pp. 822-834.

[11] R. Chillarege and R.K. Iyer, `An Experimental Study of Memory Fault Latency', IEEE Tram. Comp., v. TC-38, n. 6, June 1989, pp. 869-874.

[12] R. Harboe-Sorensen, et al., `The SEU Risk Assessment of Z80A, 8086, and 80C86 Microprocessors Intended for use in a Low Altitude Polar Orbit', IEEE Trans. Nuc. Sci., v. NS-33, n. 6, Dec. 1986, pp. 1626-1631.

[13] M.Z. Khan and J.G. - Tront, `Detection of Upset Induced Execution Errors in Microprocessors', Phoenix Conf. Comp. and Comm., to appear.

[14] S.E. Kerns, et. al, `The Design of Radiation Hardened ICs for Space: A Compendium of Approaches', Proc. IEEE, v. 76, n. 11, Nov. 1988, pp. 1470-1509. .

[15] R. Laeser, et al., `Engineering Voyager 2's Encounter with Uranus', Sci. Am., v. 255, n. 5, Nov. 1986, pp. 36-45.

[16] K. Li, et al., `An HDL Simulation of the Effects of Single Event Upsets on Microprocessor Program Flow', IEEE Trans. Nuc. Sci., v. NS-31, n. 6, Dec. 1984, pp. 1139-1144.

[17] A. Michelson and A. Levesque, Error-Control Techniques for Digital Communications, John Wiley and Sons, 1985.

[18] -, Reliability Prediction of Electronic Equipment, MIL-HDBK-217E, Oct. 27, 1986.

[19] D.K. Nichols, et al., `Recent Trends in Parts SEU Susceptibility from Heavy Ions', IEEE Trans. Nuc. Sci., v. NS-34, n. 6, Dec. 1987, pp. 1332-1337.

[20] B.D. Nordwall, `Avionics Vulnerable to Particles from Solar Flares, Cosmic Rays', Aviation Week and Space Technology, July 25,1988, pp. 50-51.

[21] W. Peterson, and E. Weldon, Error-Correcting, Codes, 2nd ed., MIT Press, MA, 1982.

[22] E. Petersen and J. Langworthy, `Suggested Single Event Upset Figure of Merit', IEEE Trans. Nuc. Sci., v. NS-30, n. 6, Dec. 1983, pp. 4533-4539.

[23] R.D. Rasmussen `Spacecraft Electronics Design for Radiation Tolerance', Proc. IEEE, v. 76, n. 11, Nov. 1988, pp.1527-1534.

[24] G.A. Sai-Halasz, `Cosmic Ray Induced Soft Error Rate in VLSI Circuits', IEEE Elect. Dev. Lett., v. EDL-4, n. 6, June 1983, pp. 172-174.

[25] P. Shapiro, et al., Calculation of Cosmic-Ray Induced Soft Errors and Scaling in VLSI Devices, NRL MR 4864, Aug. 26, 1982.

[26] M. Shoga, et al., `Verification of Single Event Upset Rate Estimation Methods with On-Orbit Observations', IEEE Trans. Nuc. Sci., v. NS-34, n. 6, Dec. 1987, pp. 1256-1261.

[27] J.R. Srour and J.M. McGarrity `Radiation Effects on Microelectronics in Space', Proc. IEEE, v. 76, n. 11, Nov. 1988, pp. 1443-1457.

[28] J. Thomlinson, et al., `The SEU and Total Dose Response of the INMOS Transputer', IEEE Trans. Nuc. Sci., v. NS-34, n. 6, Dec. 1987, pp. 1803-1807.

[29] K.S. Trivedi, Probability and Statistics with Reliability, Queuing, and Computer Science Applications, Prentice-Hall, Englewood Cliffs, New Jersey, 1982.

[30] J. Tront, et al., `Software Techniques for Detecting Single-Event Upsets in Satellite Computers', IEEE Trans. Nuc. Sci., v. NS-32, n. 6, Dec. 1985, pp. 4225-4228.

[31] J.F. Ziegler and W.A. Lanford, `The Effect of Sea Level Cosmic Rays on Electronic Devices', J. Appl. Phys., v. 52, n. 6, June 1981, pp. 4305-4312.

[32] J.A. Zoutendyk, et al., `Single-Event Upset (SEU) in a DRAM with on-Chip Error Correction', IEEE Trans. Nuc. Sci., v. NS-34, n. 6, Dec. 1987, pp. 1310-1315.