A METHOD TO COMPUTE SEU FAULT PROBABILITIES IN MEMORY ARRAYS WITH ERROR CORRECTION

Gokhan Gercek
Lockheed Engineering and Sciences Company

S28-61
44849

Abstract:

With the increasing packing densities in VLSI technology, Single Event Upsets (SEU) due to cosmic radiations are becoming more of a critical issue in the design of space avionics systems. In this paper, a method is introduced to compute the fault (mishap) probability for a computer memory of size M words. It is assumed that a Hamming code is used for each word to provide single error correction. It is also assumed that every time a memory location is read, single errors are corrected. Memory is read randomly whose distribution is assumed to be known. In such a scenario, a mishap is defined as two SEUs corrupting the same memory location prior to a read. The paper introduces a method to compute the overall mishap probability for the entire memory for a mission duration of T hours.

I. INTRODUCTION

The radiation effects in spacecraft electronics evolving into a more significant problem with advances in semiconductor technology. The minituraztiation trends in microelectronics technology have created a new set of radiation problems for the designers of space avionics. As explained in [1], space radiation is a significant cause of errors in space borne memory devices. There are various ways to deal with radiation related problems. These can be avoidance, hardening, fault tolerance and SEU tolerance [2]. Avoidance is about, given a choice, operating in a less severe radiation environment. Another way to reduce the effects of the radiation is a technique called hardening. Hardening involves both processing changes which affect material and junction properties and circuit changes which reduce or eliminate degradation and failure mechanisms. Fault tolerance is associated with redundancy and voting mechanisms to reduce or eliminate radiation caused (and sometimes due to other reasons) errors and failures. The final technique, the SEU tolerance can be also considered in fault tolerance category. SEU tolerance is about those methods, tools and designs that would reduce SEUs or their effects [3], [4], [5].

There are several aspects of SEU related problems. First, SEUs create no significant damage to the circuit but only transient error conditions. This is mostly due to the fact that effects of SEUs are confined to (albeit not exclusively) bistable flip-flop storage elements. Secondly, SEUs mostly affect single bit storages and therefore single error correction techniques are accepted as a sufficient method of dealing with SEUs.

Despite the fact that error detection and correction mechanisms are quite effective in dealing with SEUs, one must remember the cumulative effects of SEUs in such designs. The cumulative effect of the SEUs refer to the situations where number of SEUs can cause multiple error conditions in a given word in a memory array every time. Obviously, this becomes a more pronounced problem when SEU rates are higher. It must be considered in all designs when the risk (i.e. the probability of occurrence times the cost incurred from the occurrence) is fairly high due to a SEU failure. The errors induced by space radiation are known as Single Event Upsets (SEUs).

Accumulation of errors in memory arrays with error detection and correction circuits can be reduced by deploying periodic "refresh" cycles (**scrubs**) where each memory cell is read and if it is in error corrected. By selecting sufficiently small refresh cycle durations, the probability of SEU error accumulation can be minimized. Another way of improving the SEU immunity in memory arrays is refreshing memory locations during the accesses. Everytime the program executing in the CPU accesses the memory, an error checking is performed on the contents of the memory word and when an error is found the contents of the memory location is refreshed.

The refresh approach can be costlier in CPU performance since periodic refreshes steals cycles from useful CPU operations. Memory accesses for refreshes introduce additional wait states resulting in slower CPU operations.

In this paper a memory array M words is considered. It is assumed that memory contents are refreshed overtime the memory is accessed. Furthermore, a simplifying assumption is that the memory locations are accessed for reading only. This is due to the fact that when a memory location is written into, the errors in pre-write state are irrelevant since they can not cause any failures.

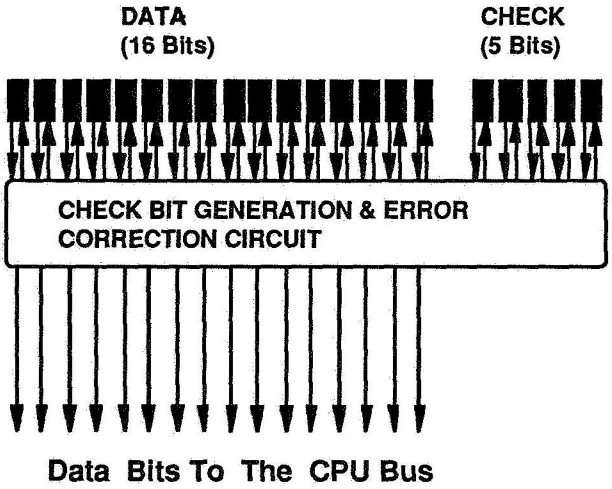

Figure 1. Word organization with check and data bits.

The access pattern to memory locations in a memory array is random in general. Therefore a memory access probability distribution is introduced to model the randomness. A bi-rectangular distribution is assumed for the derivations. However, the analysis can be carried out for any type of distribution without loss of generality.

As a memory array of M words with D data bits and C check bits is considered (i.e. total word length L=D+C). Figure 1 shows a word organization example with D=16 and C=5. The check bits are assumed to be capable of correcting single bit errors (such as Hamming code). It is also assumed that an SEU can not upset more than one bit of storage at a given time [1]. We define a "Mishap" as an error condition with more than one error accumulating in a memory location prior to a refresh. The reason for using the word "mishap" instead of "failure" is that, not every mishap can result in a failure necessarily. For example, if a memory array has some words which may never be accessed during the scrub period, then the Mishap can not result in a failure. We also assume a memory access rate of k times (randomly) per second. k can be taken roughly as the MIPs rating of the processor. We denote Λ as the SEU upset rate (upsets per unit time) for the entire memory. Thus the SEU arrival rate per word becomes $\lambda = \Lambda / M$ which is assumed to be Poisson distributed. We define the time unit, t_u, as a **quantum** which is the access time to the memory. Thus $t_u = 1/k$.

II. MEMORY PROFILE MODEL

Since the CPU accesses memory locations in a random manner, we define a memory access distribution profile or simply memory profile as the probability distribution of accessing any one of the M memory locations at a given time. Figure 2 shows the bi-rectangular distribution adopted for the subsequent analyses. Note that this is a discrete distribution with the independent variable being the address of a memory location. Although we use the bi-rectangular distribution for analytical simplicity, it can be shown that the analysis can be extended to any other type of distribution.

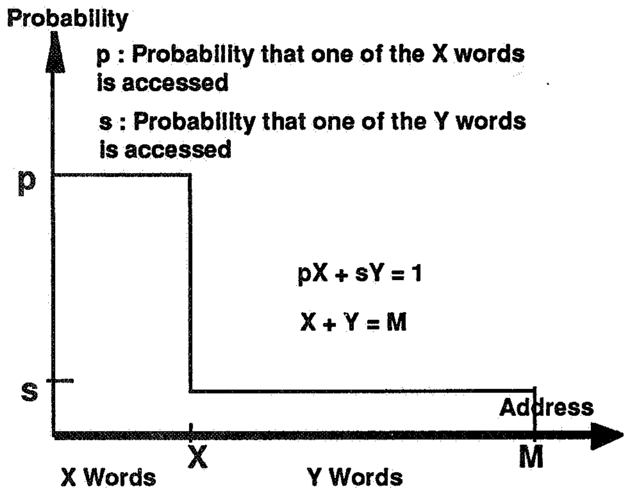
Probability
p : Probability that one of the X words is accessed
s : Probability that one of the Y words is accessed
p
pX + sY = 1
X + Y = M
s
Address
X
M
X Words
Y Words

Figure 2. Bi-rectangular memory access probability distribution profile.

The memory profile in Figure 2 can be interpreted as follows: During a given access to the memory, there is "p' probability that we will read from a particular memory location between addresses 1 and X and there is q=1-p probability that we will not read from that particular location. We define Y = M - X. The asymmetric profile reflects the fact that certain parts of the memory (i.e. first X words or "X" type) are more frequently accessed than the next Y words or "Y" type words. Similarly during a given access, we have probability s that a "Y" type memory will be read and a probability r=1-s that particular location will not be read. Note also that due to conservation of probability, pX+sY=1.

III. ANALYSIS

Consider a X type memory location "A" in the memory profile. Assume that "A" is just accessed. For our subsequent analyses we will call the interaccess time for a given memory location the location inter read time (LIRT). The probability that "A" will be accessed again after N quanta is:

$$P\{ \text{ LIRT for A} = N \} = P_N = p\, q^{N-1} \qquad (1)$$

As Equation (1) suggests, the LIRT of a given memory location is geometrically distributed.

Now let's consider the probability of two or more SEUs striking this memory location during the LIRT of N quanta. Note that if two or more SEUs corrupt the memory location, this would result in a mishap.

$$P\{ \text{two or more SEUs in N quanta} \} = 1 - P\{ 0 \text{ SEU in N quanta} \} - P\{ 1 \text{ SEU in N quanta} \}$$

Since SEU arrivals are assumed to be Poisson distributed with parameter λ,

$$P\{ 0 \text{ SEU in N quanta} \} = e^{-\lambda N}$$

$$P\{ 1 \text{ SEU in N quanta} \} = \lambda N e^{-\lambda N}$$

Thus

$$P\{ \text{two or more SEUs in N quanta} \} = 1 - e^{-\lambda N} - \lambda N e^{-\lambda N}$$

or

$$P\{ \text{Mishap in N} \} = 1 - e^{-\lambda N} - \lambda N e^{-\lambda N} \qquad (2)$$

And probability of success in N quanta will then be:

$$P\{ \text{Success in N} \} = 1 - P\{ \text{Mishap in N} \}$$

$$P\{ \text{Success in N} \} = 1 - e^{-\lambda N} - \lambda N e^{-\lambda N} \qquad (3)$$

Since LIRT is geometrically distributed as given by Equation 1, the expected value of LIRT is 1/p. This means that an X type of location is read on the average once every 1/p quanta. Thus

$$E\ \{\ LIRT_X\ \} = 1/p \qquad (4)$$

In Equation (3), the probability of success is given under the assumption that the $LIRT_X$ is given as N. Since $LIRT_X$ geometrically distributed, the average probability of success during $LIRT_X$ can be found by:

$$P_s = P\{\ average\ success \sin\ LIRT_X\}$$

$$P_s = P\{\ average\ success \sin\ LIRT_X\} = \sum_{N=1}^{\infty} (e^{-\lambda N} + \lambda N\, e^{-\lambda N}\ )\, pq^N \qquad (5)$$

Equation 5 can be separated into two infinite series and each can be individually computed to yield:

$$P_s = \frac{pe^{-\lambda}}{1-qe^{-\lambda}} + \frac{p\lambda\, e^{-\lambda}}{(1-qe^{-\lambda})^2} \qquad (6)$$

It should be noted that Equation 6 is a computation sensitive equation since the numbers involved are very small (e.g. $p \approx 10^{-6}$, $q \approx 1 - 10^{-6}$ and $\lambda \approx 10^{-15}$). If Equation 6 is computed using a typical set of numbers with a calculator, the resulting value for P_S would likely to be 1.0 due to the computation sensitivity of Equation 6.

In order to facilitate the computational problem, we can introduce the following form for P_S:

$$P_s = 1 - \varepsilon_x \qquad (7)$$

In Equation 7, ε_X is a very small number which represents the probability of mishap during an average $LIRT_X$. By using first order Taylor series approximation it can be shown that:

$$P_s = 1 - \frac{\lambda^2}{p} \qquad (8)$$

or equivalently

$$\varepsilon_x = \frac{\lambda^2}{p} \qquad (9)$$

Now let's assume that all memory locations are scrubbed every T many quanta. For a given memory location of type X, the probability of success in T quanta is:

$$P_{sT} = (\,1 - \varepsilon_x\,)^m \qquad (10)$$

Where m=T/(1/p) or the number of average size LIRTs in T. The probability that all the locations of type X survives during T quanta is:

$$P_{sT} = (\,1 - \varepsilon_x\,)^{mX} \qquad (11)$$

Since ε_x is a very small number and mX is a very large number Equation 11 can be approximated as:

$$P_{sTX} = 1 - mX\varepsilon_x \qquad (12)$$

Equation 12 is the survival probability for the first X words of the memory for a duration of T. By using similar arguments, for the Y type locations, the survival probability can be found as:

$$P_{sTY} = 1 - nY\varepsilon_Y \qquad (13)$$

In Equation 13, n=Ts , s being the probability of accessing a Y type location at a given read. ε_Y is defined in a similar way as ε_x in the following way:

$$\varepsilon_Y = \frac{\lambda^2}{s} \qquad (14)$$

The survival probability for the entire memory can then be computed as:

$$P_{sT} = (1 - nY\varepsilon_Y)\ \ (1 - mX\varepsilon_x)$$

or

$$P_{sT} = (1 - nY\varepsilon_Y - mX\varepsilon_x + mnXY\varepsilon_x\varepsilon_Y) \tag{15}$$

and the probability of a mishap in the entire memory for a duration of T can then be found as:

$$P_{mishapT} = nY\varepsilon_Y + mX\varepsilon_x - mnXY\varepsilon_x\varepsilon_Y \tag{16}$$

Example: Let's assume a memory of 250 KWords with a word size of 32 bits (without the checkbits), a memory profile as shown in Figure 2, an access rate of 5 million reads per second (i.e. quanta = 0.2 µsec.) and an SEU arrival rate of 10^{-15} upsets/word/quantum. Let's also assume that the memory is never scrubbed during the entire mission which lasts 30 days (i.e. T=720 hours). Using the analysis given in the paper, the probability of a mishap during the entire mission can be computed as $P_{\mathrm{mishapT}} = 5\ 10^{-12}$.

IV. CONCLUSION

It is shown that SEU reliability of memory arrays with single error correction feature is predictable when a memory profile can be associated with the memory access patterns. Although the derivation is performed for a bi-rectangular profile, it is possible to extend the approach to general profile models. In case periodic scrubs are used, the analyses yield the result for one scrub cycle. The mishap probability for the entire mission can then be found by multiplying the number of scrubs in a mission with the mishap probability in one scrub cycle.

REFERENCES

[1] J. Bradford and P. Bail, "Cosmic ray effects in spaceborene electronics" Proc. Third Digital Avionics Systems Conf., Forth Worth, TX, Nov. 1979.

[2] R. D. Rasmussen, "Spacecraft Electronics Design for Radiation Tolerance", Proceedings of the IEEE, Vol. 76, No. 11, November 1988.

[3] G. Burdick, E. Kopf and D. Meyer, " The Galileo Single Event Upset Solution and Risk Assessment", Proc. Ninth Ann. Rocky Mountain Guidance and Control Conf. February 1986.

[4] P. Lewkowicz and L. Richter, "Single event upsets in spacecraft digital systems," Proc. Int. Telemetering Conf., October 1984.

[5] M. Sievers and G. Gilley, " An architectural approach to tolerating radiation effects," IEEE Ann. conf. on Nuclear and Space Radiation Effects Short Course, Monterey, CA, July 1985.