METHODS FOR PROTECTION OF COMPUTER MEMORIES

M. BLAUM, R. M. F. GOODMAN, and R. J. McELIECE
Electrical Engineering Department
California Institute of Technology, Pasadena, California

ABSTRACT

In this paper we discuss two methods for increasing the lifetime of a computer memory with multiple error-correction-coding (ECC): adding s rows of spare chips with single error correction, versus double error correction. Comparing the two methods in terms of the number of redundant chips, we conclude that sparing is better when the number of rows of chips is large enough. However, if complexity of implementation is considered, 2-ECC may be superior.

1. INTRODUCTION

A computer memory is an M by n array of by-one memory chips, such that the first k columns are information chips and the last $n-k$ columns are redundant chips. Thus k is equal to the width of the computer word. The redundant chips are used to encode and protect the information in general through a single-error-correcting double-error-detecting (SEC–DED) code [2]. Internally, each chip is an ℓ by ℓ array of cells, each of which stores a 0 or a 1. Chip failures can be hard or soft, and we assume that the soft errors which generally only affect a single cell can be easily removed by periodic scrubbing. In the case of hard errors, several types of chip failures can occur: single-cell failures, row failures, column failures, and catastrophic chip failures. Regardless of how bad a chip failure is, the code will correct it whenever no more than one chip in a row is affected. However, it is expected that after a certain amount of time, an uncorrectable pattern will have occurred. In that case, we say that a memory failure has occurred.

A natural problem is to find the mean time between failures (MTBF) in a memory with a SEC–DED code, *i.e.*, how long we have to wait on the average, until an uncorrectable pattern occurs. This problem has been extensively studied ([1]–[3]). These studies show that if we define the coding gain (CG) as the ratio of MTBF with SEC–DED to the MTBF without SEC–DED, then this CG may be as large as several thousands. In this paper we are concerned with methods of producing even higher reliability memory systems. Two methods are of interest, adding spare chips to a SEC–DED code and performing spares switching, or alternatively implementing double error correction (DEC–TED).

2. SPARES SWITCHING

One way of further extending the lifetime of the computer memory is by adding s rows of spare chips to the M by n array which also has 1–ECC. The spare chips act in such a way that each time a chip failure occurs, a connection to a spare chip in the corresponding column is made. In practice, this means that the chip is replaced by one of the s spare chips in the corresponding column. Hence, s failures per column are tolerated before errors are actually corrected continuously.

Call $(CG)_s$ the coding gain when s rows of spare chips are added in this way. We have found the following upper and lower bounds for $(CG)_s$:

$$(CG)_0 + \frac{k}{n} B_2(n, s+1) \leq (CG)_s \leq (CG)_0 + ks \tag{1}$$

where $B_2(n, s+1)$ is the average number of balls we have to place in n cells, until either two cells have $s+1$ balls each or one cell has $s+2$ balls. An expression for $B_2(n, s+1)$ is given by ([1]). Notice that $(CG)_0$ is the coding gain when 1–ECC is implemented with no spares.

3. DOUBLE ERROR CORRECTION

An alternative way of further extending the lifetime of the computer memory is to implement 2–ECC, that is, adding more parity check chips so that we have a double-error-correcting triple-error-detecting (DEC–TEC) code.

Let us call $(CG)_0'$ the coding gain for the memory with a DEC–TEC code. The memory protected by a DEC–TED code can be considered to be an M by n' array of chips, where n' is the number of columns needed to implement the DEC–TED code. If we assume that only cell row and column failures occur, and that they occur with the same frequency, we obtain the following results:

$$(CG)_0' = \frac{k}{n'} \int_0^\infty e^{-M\ell x} (1 + x + \frac{x^2}{4})^M \, \ell dx \tag{2}$$

and as $M \to \infty$,

$$\frac{(CG)_0'}{(CG)_0} \sim \frac{n}{n'} \sqrt{\ell} \tag{3}$$

For example, if $\ell = 256$ (64K chips), $k = 32$, $n = 39$, $n' = 39$, $n' = 45$, and M a large number, the increase in coding gain when DEC–TED is implemented with respect to the memory with SEC–DED is roughly 14 times.

4. REDUNDANCY COMPARISON

Let us now compare the redundancy of the two methods. Assume that M is large enough so that our asymptotic approximations hold. We wish to estimate the number

of spare chips that we have to add such that $(CG)_s \geq (CG)'_0$. If this is less than the number needed for 2-ECC, then sparing is better. For the coding gain condition to hold, we have from Equation (1):

$$(CG)_0 + \frac{k}{n} B_2(n, s+1) \geq (CG)'_0 \tag{4}$$

Replacing (3) in (4) and given that ([1]):

$$(CG)_0 \sim \frac{k}{n} \sqrt{\pi} \sqrt{M} \tag{5}$$

we have the following inequality on s:

$$\left(\frac{n}{n'} \sqrt{\ell} - 1\right) \sqrt{\pi} \sqrt{M} \leq B_2(n, s+1) \tag{6}$$

Using our typical example with $k = 32$, $n = 39$, $n' = 45$, and $\ell = 256$, we obtain:

$$(22.8)\sqrt{M} \leq B_2(39, s+1) \tag{7}$$

Using Equation (7), and performing numerical integration to find $B_2(39, s+1)$, we can find the minimum number of spare rows s that satisfies the inequality for fixed M. Let us call this value $s(M)$. Now, if we add $s(M)$ spare rows, we are adding $39s(M)$ chips, while if we implement a DEC-TED code, we have to add $6M$ chips. Thus if $39s(M) < 6M$ or $s(M) < 2/13M$, we conclude that sparing is better.

The table below compares these values. It can be seen that $s(M)$ grows very slowly. For M around 50, the result is inconclusive, but for $M > 60$ sparing is clearly better.

M	$s(M)$	$2/13M$
50	8	7.7
60	9	9.2
80	10	12.3
100	11	15.4
200	14	30.8
400	19	61.5

However, spares switching is much more complex to implement, and so 2-ECC may turn out to be superior.

References

(1) M. Blaum, "Ph.D. Thesis," Caltech, 1984.

(2) R. M. F. Goodman and R. J. McEliece, "Lifetime analyses of error-control coded semiconductor RAM systems," Proc. IEEE, v. 129, part E (32), 81-85 (May 1982).

(3) R. M. F. Goodman and R. J. McEliece, "Hamming codes, computer memories, and the birthday surprise," Proc. 20th Allerton Conference on Communication, Control and Computering, 672-679 (1982).