2009 24th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems

# Error correction codes for SEU and SEFI tolerant memory systems

S. Pontarelli$^{\diamond\star}$, G.C. Cardarilli$^{\diamond}$, M. Re$^{\diamond}$, A. Salsano$^{\diamond}$
{pontarelli, salsano}@ing.uniroma2.it, {marco.re, g.cardarilli}@ieee.org
$^{\diamond}$ University of Rome ”Tor Vergata”, Via del Politecnico 1, 00191, Rome, ITALY
$^{*}$ (ASI) Italian Space Agency, Viale Liegi, 26, 00198 Rome, ITALY

**Abstract**

In this paper a modification of the Hsiao SEC-DED (Single Error Correction, Double Error Detection) code is presented. The proposed code is still a SEC-DED code, but it is also able to correct a byte erasure. This code has been developed to protect the memory chips of a spaceborne computer against SEU (Single Event Upset) and SEFI (Single Event Functional Interruption) faults. The code rate of our proposed code is the same of the Hsiao code and is particularly suitable for byte organized 64-bits memory systems. In fact, for these systems a (72,64) code can be constructed and a memory organization based on nine chips can be designed. The byte erasure correction allows to tolerate the occurrence of a SEFI fault in one of the memory chips without data loss.

## I. Introduction

Commercial Off The Shelf (COTS) memory chips are widely used in space application due to their superior performances with respect to the radiation hardened counterparts. The drawback of COTS memory chips is the susceptibility of these chips to the SEE (Single Event Effects) due to the ionizing radiations present in the space environment. These events can be transient or permanent faults.

Single Hard Error (SHE) [1] is a permanent fault, such as stuck bit in a memory device, that causes a permanent change to the operation of a device. Single Event Latchup (SEL) [2],[3] causes loss of device functionality due to the triggering of a parasitic *pnpn* structure and the consequent current flow can be destructive for the integrated circuit. An SEL may or may not cause permanent device damage, but requires power strobing of the device to resume normal device operations. If power consumption is low enough that no permanent damage occurs this fault can also be called a power-cycle soft error (PCSE)[4]. In fact, when the power is cycled, the thyristor is turned OFF and the chips return to its normal functional mode. Finally, between permanent faults we cite also Single Event Burnout (SEB)[5] - a condition which can cause device destruction due to a high current state in a power transistor and Single Event Gate Rupture (SEGR)[5] - in which the ionizing particle is able to form a conducting path in the gate oxide.

The occurrence of catastrophic failure in memory chip can be faced adding redundant spare chips. Spare memory chips can be used to substitute out of order memory chips that are switched off after the detection of a catastrophic failure. Another way to tolerate permanent fault is using EDAC code with error and erasure correction capabilities. [6],[7]. We recall that an erasure is an error for which is knowing the location inside the codeword. The difference between error and erasures is exploited in error correction algorithms because the redundancy used to correct erasures is less than the one needed to correct an error[8].

If a permanent fault in a memory chip is detected the codeword symbols stored in this memory are marked as erased, and the decoder is able to perform an erasure correction procedure to provide the corrected codeword.

Other types of SEE can be classified as soft errors. In this category fall Single Event Upset (SEU) and Multiple Bit Upset (MBU) [5],[9] that can change the value stored in one or in more memory locations. After the occurrence of one of these events the device continues to work properly even if the data stored in the location hit by the particle are corrupted. Another soft error is the Single Event Functional Interrupt (SEFI) that usually is due to a particle that hits the control circuitry of the memory chip. This effect is more likely in memory chips with complex control circuits such as DDR [9] and flash based memories[10]. In fact, the complexity growth of these circuits corresponds to an increase of the circuit area exposed to the particles flux.

Following the characterization given in [9] we divide this kind of error in soft and hard SEFI. When a particle hits the control circuits of the chip the memory can switch into an unknown state. A SEFI recovery operation can be performed to get the memory back into a known state allowing reloading of the control registers. If the recovery succeeds the error is called soft SEFI, while if it fails, then the only method of recovery is a power cycle, losing the data stored in the entire chip. This event can be classified as hard SEFI and is similar to the SEL induced PCSE[4]. To face the occurrence of these soft errors EDAC (Error Detection And Correction) codes are usually used to protect the memory context [11].

One of the more used EDAC code is the Hsiao SEC-DED code, originally proposed in [12] that is able to correct one bit error and to detect two erroneous bits. Therefore, this code is particularly suitable for correcting SEUs. In this paper we modify the Hsiao code by a suitable ordering of the columns composing the parity check matrix of the code in order to guarantee that a byte erasure can be corrected by the code. By adding this feature the Hsiao code is also able tolerate the occurrence of soft SEFI and PCSE if a suitable organization of the memory is chosen. If we store in different chips the different bytes composing the Hsiao codeword, a soft SEFI or a PCSE produces an error in a know byte of the codeword. Therefore can be treated as erasure and corrected by our proposed code. Finally, we notice that the problem discussed in this paper was born for space applications [11] but is nowadays applicable also to terrestrial environment, in particular for servers and workstations[4]. In fact, the last generation memory chips are sensitive to atmospheric neutrons that are present in the terrestrial environment. Many papers [13],[14][15] report the SEE caused by neutrons on memory chips.

The rest of the paper is divided as follows: the section II shows how to construct our proposed code, while section III describes the memory organization of a space computer using this code and the erasure correction algorithm. Section IV computes the Bit Error Rate (BER) achievable with the proposed code for a commercial SDRAM used in a satellite positioned in a geosynchronous orbit. The obtained data are compared with the BER of a Reed Solomon code used in a memory system for space applications [16]. Finally, in Section V we report the conclusions of this work.

## II. Hsiao code with byte erasure correction

The Hsiao code is a SEC-DED code with a minimum number of ones in the parity check matrix. The minimum number of ones allows implementing the encoder with the minimum number of xor gates. The rules for composing the parity check matrix $\mathbf{H}$ are:

1) all the columns are different from $\mathbf{0}$.
2) Each column is different from the others. This allows identifying a single error. In fact, computing the syndrome vector $\mathbf{S}$ for a single error ($\mathbf{e}^T = (0 \dots e_i \dots 0)$) we obtain $\mathbf{S} = \mathbf{He} = \mathbf{h}_i$, where $\mathbf{h}_i$ is the $i$ column di $\mathbf{H}$ and corresponds to the location in which the error is occurred.
3) Each column had an odd number of ones. With this rule the sum of two column is composed by a even number of ones. Therefore the syndrome of a double error $\mathbf{e}^T = (0 \dots e_i \dots e_j \dots 0)$ is the sum of two columns of $\mathbf{H}$. This syndrome is different from zero due to rules 1 and 2. The syndrome is also different from a syndrome corresponding to a single error because in the case of single error the number of ones in the syndrome is odd, here is even.

Now we describe the parity check matrix of a (64,72) Hsiao code. $\mathbf{H}$ is a $8 \times 72$ matrix. Eight columns are composed only by a 1 and forms the separable part of the code. The $\mathbf{H}$ matrix can therefore be written as:

$$\mathbf{H} = [\mathbf{P}|\mathbf{I}_8] = \begin{pmatrix} h_{11} & h_{12} & \dots & h_{1k} & 1 & 0 & \dots & 0 \\ h_{21} & h_{22} & \dots & h_{2k} & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ h_{(n-k)1} & h_{(n-k)2} & \dots & h_{(n-k)k} & 0 & 0 & \dots & 1 \end{pmatrix} \tag{1}$$

where $\mathbf{P}$ is related to the generator matrix $\mathbf{G}$ by the following equation.

$$\mathbf{G} = \left[\frac{\mathbf{I}_k}{\mathbf{P}}\right] \tag{2}$$

With $n-k = 8$ we have $\binom{8}{3} = 56$ columns with three ones. All these columns are used to construct the Hsiao code. The last 8 columns that must be used to construct $\mathbf{P}$ must have 5 ones. Now, we divide the $\mathbf{H}$ matrix and the corresponding codeword in blocks of eight bits, corresponding to the byte stored in different memory chips. The (72,64) is divided in 9 bytes and the $\mathbf{H}$ matrix can be written as:

$$\mathbf{H} = [\mathbf{H}_1|\mathbf{H}_2|\mathbf{H}_3|\mathbf{H}_4|\mathbf{H}_5|\mathbf{H}_6|\mathbf{H}_7|\mathbf{H}_8|\mathbf{H}_9]$$

where each $\mathbf{H}_i$ is an $8 \times 8$ matrix and $\mathbf{H}_9 = \mathbf{I}$.

The codeword can be written as $\mathbf{c}^T = (\mathbf{c}_1^T, \mathbf{c}_2^T, \mathbf{c}_3^T, \mathbf{c}_4^T, \mathbf{c}_5^T, \mathbf{c}_6^T, \mathbf{c}_7^T, \mathbf{c}_8^T, \mathbf{c}_9^T)$, where each block $\mathbf{c}_i$ is a byte and the dataword is $\mathbf{d}^T = (\mathbf{c}_1^T, \mathbf{c}_2^T, \mathbf{c}_3^T, \mathbf{c}_4^T, \mathbf{c}_5^T, \mathbf{c}_6^T, \mathbf{c}_7^T, \mathbf{c}_8^T)$. The last byte $\mathbf{c}_9$ is the check symbol and is computed as $\mathbf{c}_9 = \mathbf{Pd}$. Now, let us suppose that we have side channel information on the codeword that localizes an erasure in the $i$-th byte composing the codeword. Computing $\mathbf{Hc} = \mathbf{0}$ we obtain:

$$\mathbf{Hc} = \mathbf{H}_1\mathbf{c}_1 + \dots \mathbf{H}_i\mathbf{c}_i + \dots \mathbf{H}_8\mathbf{c}_8 + \mathbf{c}_9 = \mathbf{0}$$

This equation is equivalent to:

$$\sum_{\substack{j=1 \\ j \neq i}}^{9} \mathbf{H}_j\mathbf{c}_j = \mathbf{H}_i\mathbf{c}_i \tag{3}$$

Note that the left-hand side of the equation is error free, because the error is confined in the $i$-th byte. We are able to recompute the correct value of $\mathbf{c}_i$ from equation (3) if $\mathbf{H}_i$ is invertible. In this case the correct value of $\mathbf{c}_i$ is:

$$\mathbf{c}_i = \mathbf{H}_i^{-1}(\sum_{\substack{j=1 \\ j \neq i}}^{9} \mathbf{H}_j\mathbf{c}_j) \tag{4}$$

From the above discussion the necessary and sufficient condition for correcting a byte erasure in the Hsiao code is that the all the 9 $8 \times 8$ $\mathbf{H}_i$ matrices are invertible. It must be noticed that we do not want to change the algorithm for selecting the columns forming the parity check matrix, but we want only to change the order in which such columns are arranged to obtain invertible $\mathbf{H}_i$ matrices.

We have developed an algorithm that searches a $\mathbf{H}$ matrix composed by all invertible $\mathbf{H}_i$ submatrices. The algorithm start with a Hsiao Parity Check Matrix $\mathbf{H}$ and analyze the first submatrix $\mathbf{H}_1$. While the determinant of $\mathbf{H}_1$ is zero, (*i.e.* is not invertible) a column of $\mathbf{H}_1$ is exchanged with a column of another submatrix randomly chosen. When the submatrix becomes invertible the algorithm continues with the next submatrix. We notice that the exchange of column is allowed only between submatrices that are not yet invertible. The algorithm is presented here (Alg. 1):

**Algorithm 1**: Hsiao Parity Check Matrix With Invertible Submatrices Search Algorithm

```
Input: an Hsiao Parity Check Matrix $\mathbf{H}$
Output: $\mathbf{H}_1 \dots \mathbf{H}_9$ with $det(\mathbf{H}_i) \neq 0$ and $\mathbf{H} = [\mathbf{H}_1|\mathbf{H}_2|\mathbf{H}_3|\mathbf{H}_4|\mathbf{H}_5|\mathbf{H}_6|\mathbf{H}_7|\mathbf{H}_8|\mathbf{H}_9]$
begin
    for ($i = 1$; $i < 8$; $i$++) do
        while $det(\mathbf{H}_i) \neq 0$ do
            choose a column of $\mathbf{H}_i$;
            choose a column of $\mathbf{H}_j$ with $j > i$;
            exchange the two columns;
        end
    end
    while $det(\mathbf{H}_8) \neq 0$ and $det(\mathbf{H}_9) \neq 0$ do
        choose a column of $\mathbf{H}_8$;
        choose a column of $\mathbf{H}_9$;
        exchange the two columns;
    end
end
```

The second while cycle is different from the first one since we need to check that the last two submatrices are simultaneously invertible. Otherwise, we cannot change the columns on the last submatrix if it is not invertible when all the other column has been fixed. Even if the algorithm is not optimized it provides very quickly the $\mathbf{H}$ matrix for the (64,72) code. The submatrices are reported in the following equation.

$$\mathbf{H}_1 = \begin{pmatrix} 0&1&0&0&0&1&0&1\\ 0&0&0&1&0&0&0&0\\ 0&0&0&0&1&0&0&1\\ 0&0&1&0&0&0&1&0\\ 1&0&0&1&1&1&0&0\\ 1&0&1&0&1&0&1&1\\ 1&1&1&1&0&0&0&0\\ 0&1&0&0&0&1&1&0 \end{pmatrix} \mathbf{H}_2 = \begin{pmatrix} 1&0&0&0&0&0&0&1\\ 1&0&0&0&0&0&1&0\\ 0&0&1&1&1&1&0&0\\ 0&1&0&0&0&0&0&0\\ 1&1&0&0&0&1&1&1\\ 0&1&0&1&1&0&0&0\\ 0&0&1&0&1&0&0&1\\ 0&0&1&1&0&1&1&0 \end{pmatrix} \mathbf{H}_3 = \begin{pmatrix} 0&1&0&0&0&0&0&0\\ 0&0&0&0&1&1&0&0\\ 1&0&1&0&0&0&1&1\\ 1&1&1&1&0&1&0&1\\ 0&0&0&1&0&0&1&1\\ 0&0&1&0&0&0&0&0\\ 0&1&0&1&1&1&1&0\\ 1&0&0&0&1&0&0&0 \end{pmatrix}$$

$$\mathbf{H}_4 = \begin{pmatrix} 1&0&1&0&0&0&0&0\\ 0&1&1&1&1&1&1&1\\ 0&0&0&1&0&0&1&1\\ 1&0&1&0&1&1&0&0\\ 0&1&0&1&0&1&0&0\\ 0&1&0&0&1&0&0&0\\ 0&0&0&0&0&0&0&1\\ 1&0&0&0&0&0&1&0 \end{pmatrix} \mathbf{H}_5 = \begin{pmatrix} 0&1&1&0&1&1&0&1\\ 1&0&1&0&0&0&0&0\\ 1&1&0&0&0&0&0&0\\ 0&0&0&0&1&0&0&0\\ 0&0&0&0&1&0&1&0\\ 1&0&0&1&0&1&1&1\\ 0&1&0&1&0&1&0&0\\ 0&0&1&1&0&0&1&1 \end{pmatrix} \mathbf{H}_6 = \begin{pmatrix} 1&1&0&1&0&0&0&0\\ 0&0&0&0&1&0&1&0\\ 0&1&1&0&0&0&0&0\\ 0&0&1&1&0&1&0&0\\ 1&0&0&0&0&1&0&1\\ 1&0&0&1&1&0&1&0\\ 0&0&1&0&1&0&0&1\\ 0&1&0&0&0&1&1&1 \end{pmatrix}$$

$$\mathbf{H}_7 = \begin{pmatrix} 1&1&0&1&1&0&0&1\\ 0&0&1&1&1&0&1&1\\ 1&1&1&0&0&0&0&1\\ 0&1&1&0&0&1&1&0\\ 1&0&0&0&0&0&0&0\\ 0&0&0&0&1&0&0&0\\ 0&0&0&1&0&1&0&0\\ 0&0&0&0&0&1&1&0 \end{pmatrix} \mathbf{H}_8 = \begin{pmatrix} 0&0&1&0&0&1&1&0\\ 1&1&0&0&0&1&0&0\\ 0&0&0&1&1&1&1&0\\ 1&0&0&1&1&0&0&1\\ 0&1&1&1&1&1&1&1\\ 1&1&1&1&0&1&1&1\\ 1&1&1&0&1&0&1&1\\ 1&1&1&1&1&0&0&1 \end{pmatrix} \mathbf{H}_9 = \begin{pmatrix} 1&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0\\ 0&0&0&1&0&0&0&0\\ 0&0&0&0&1&0&0&0\\ 0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&1 \end{pmatrix}$$

The proposed code is used to protect a 64-bit memory for a space computer (like the Maxwell SCS750[16] that uses a PowerPC 750FX with a 64-bit data bus), as it will be discussed in the next section.

## III. Memory organization

In this section we describe the memory organization of a space computer using the (64,72)SEC-DED code defined in Section II. The 72-bits codeword is composed by 9 bytes stored in different memory chips. The use of the SEC-DED code is preferable with respect to a Reed Solomon (11,8) code as it was used in [16], for two reasons: the former is that the SEC-DED coding and decoding procedures are simpler and faster than the RS counterparts, the latter is that it requires only 9 memory chip with respect to the 11 memory chips of the RS code. The schema of the processor/memory interface with a SEC-DED codec is shown in Fig. 1.

The blocks depicted in Fig. 1 are:

1) the processor
2) the (64,72)SEC-DED codec
3) a Scrubbing Block
4) nine 8-bits memory chips
5) nine Power Controllers (PW controller)

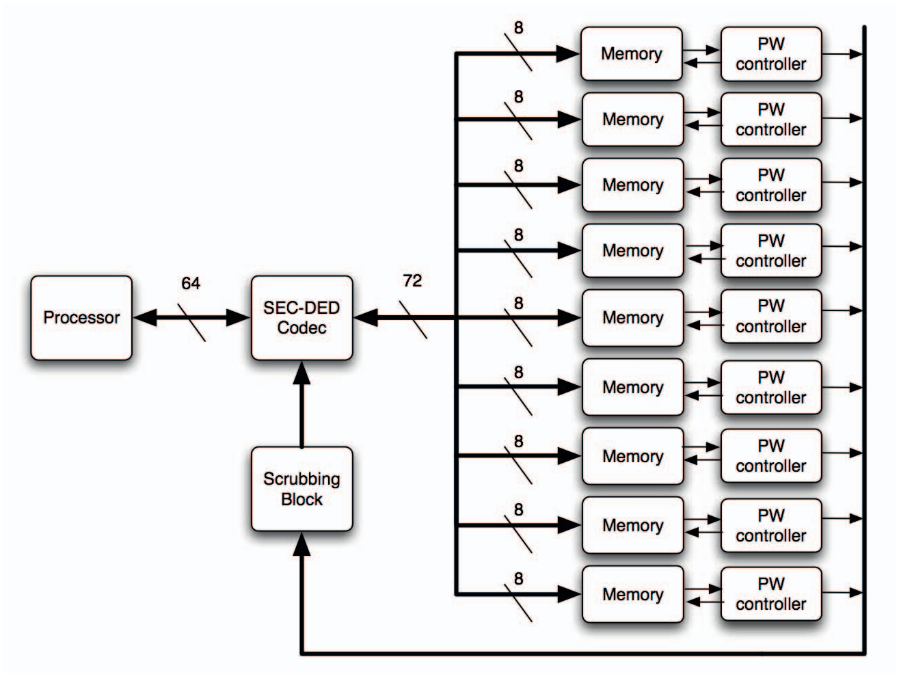

Fig. 1. Processor/memory interface with a SEC-DED codec

The processor is connected to the memory chips using the SEC-DED codec that codifies the data that will be written in the memory and decodes and eventually corrects the data coming from the memory.

The scrubbing block will avoid the accumulation of errors inside the memory. The scrubbing procedure requires reading each codeword stored inside the memory and rewrites the read (and eventually corrected) codeword in the same memory location. In such way the probability that two SEUs corrupt the same codeword, generating an uncorrectable error, is drastically reduced, since the two SEUs should occur in an time interval less than the one between two consecutive scrubbing operations.

Finally, the power controller blocks perform two main operations:

1) the first operation is the IDDQ-based Erasure Detection [7], [17],[18]. Basically, they monitor the occurrence of Hard SEFI/ SEL by current sensing.
2) The second operation starts when a PCSE fault is detected. The memory chip is power-cycled and a signal is sent to the Scrubbing Block. The power cycle erases the context of the entire memory chip, corresponding to an erasure in all the codeword stored in the memory chips. All the memory must be scrubbed, and all the codeword will present an erasure in the location corresponding to the memory chip affected by the PCSE.

Therefore, the scrubbing block gives to the SEC-DED code the erasure location. The SEC-DED codec decodes the codeword by using equation (4) described in Section II and thus recovers the memory context after the PCSE.

## IV. BER Evaluation

Now we evaluate the effectiveness of our solution against SEU and hard SEFI. We suppose to use a DDR-II SDRAM memory of 4 GB for the processor and we want to obtain a BER less than 1E-12.

The SEE data related to the radiation environment are taken from [9]. The memories taken into account are:

1) Micron 512Mx8
2) Hynix 512Mx8
3) Samsung 512Mx8
4) Samsung 1Gx8

From the data in [9] the Micron SDRAM show a SEL sensitivity for a LET of 68.5 $MeV/(mg/cm^2)$, while the other memories do not present SEL.

For Samsung and Infineon memory chips both Soft and Hard SEFIs were recorded throughout the various tests. The reported cross-sections for SEU are about 5E-10 $cm^2/bit$ and 3E-11 $cm^2/bit$ respectively.

Finally, the Elpida chips shows an extremely low number of SEUs (the cross section is 3E-12 with a LET of 68.5 $MeV/(mg/cm^2)$ and no MBUs have been reported.

The Table I summarize these results.

| **Manufacturer** | **SEL** | **cross-section** $cm^2/bit$ | **Hard SEFI** |
|---|---|---|---|
| Samsung | NO | 5E-10 | YES |
| Infineon | NO | 3E-11 | YES |
| Micron | YES | - | YES |
| Elpida | NO | 3E-12 | YES |

TABLE I
Summary of radiation Test Results for DDR-II

The collected data allow excluding the Micron chips from the list of candidate devices due to their high sensitivity to SEL.

From the surviving list the Elpida chips are the one providing the best behavior in terms of SEU rate.

The absence of MBU [9] makes the choice of the SEC-DED suitable for correction od SEU, while the occurrence of SEFI will be faced as described in the previous sections. From the above consideration, we conclude that the Elpida SDRAM is suitable for our solution.

If the Elpida chip is positioned in the GEO orbit the SEU rate calculated by CREME96 [19] is 1.8E-11 ([#events/bit/day]).

From the above data the BER of a 4GB memory has been computed using a SEC-DED(72,64) Hamming code with different scrubbing rates. The results are reported in Table II.

The table report also the BER of a Reed Solomon RS(11,8) code. This code is able to correct a single byte error and to detect a double byte error. We compare the performance achievable with the two codes. The RS(11,8) had a similar behavior with respect to SEU, while is able to correct also MBU that occurs in the same byte. We notice that this feature is useless in our case in which the selected memory does not present MBU [9]. Instead, this code is able to correct a SEFI if the same Processor/memory interface depicted in Fig. 1 is used.

| **EDAC** | **Scrubbing Rate** | **BER [#events /day]** |
|---|---|---|
| SEC-DED (72,64) | 72 hours | 3.9E-10 |
| SEC-DED (72,64) | 1 hour | 5.4E-12 |
| SEC-DED (72,64) | 10 min | 9E-13 |
| RS (11,8) | 72 hours | 6.1E-10 |
| RS (11,8) | 1 hour | 7.8E-12 |
| RS (11,8) | 10 min | 1.3E-12 |

TABLE II
BER for 4GB DRAM Memory

From the results reported in Table II the SEC-DED(72,64) code is able to fulfill the requirements if a suitable scrubbing rate is used. Comparing the BER of our proposed code with the BER of the RS code we notice that the latter had a worse BER than the SEC-DED code of about the 40%. This is due to the number of memory chips used for the two codes. The SEC-DED code requires 9 chip, while the RS code requires 11 chips, therefore the occurrence of a double error is more likely in the latter case. Finally, we evaluate the probability that two subsequent SEFIs produce an uncorrectable error in the processor memory. Considering the worst day GEO condition the SEFI occurrence evaluated by CREME96 [19] is 1.2E-11 SEFI/sec. The Hard SEFI condition requires a power cycle for the memory chip affected by the SEFI. After this power cycle, the scrubbing procedure is started. The DDR-II memories taken into account had an operating frequency of 533 MHz and the number of codeword to be scrubbed is 4GigaByte/8byte = 512M. Supposing two clock cycle for read and rewrite a codeword the scrubbing time is about 2 seconds. Therefore, we suppose that two subsequent SEFIs produce an uncorrectable error is they occur in an interval less than 2 seconds. The probability to had a SEFI in 2 seconds is $p_{SEFI} = 2.4E - 11$. If we bad 9 chips as described in Fig. 1 the probability of two SEFI in two seconds is:

$$p_U = 9 \cdot 8 \cdot p_{SEFI}^2 \cdot (1 - p_{SEFI})^7 \tag{5}$$

This is the probability of an uncorrectable error in the memory due to the SEFI occurence. Using a SEFI rate of 1.2E-11 SEFI/sec for a single chip, we obtain $p_u$=1E-20.

## V. Conclusion

In this paper a SEC-DED code with erasure correction capability is presented. The code is used to protect the memory of a processor for space applications by using a suitable memory organization. The memory organization is able to detect hard SEFI and SEL by current monitor. After the detection of such faults these PCSE are tolerated by using a scrubbing procedure that exploit the erasure correction capabilities of our code. Finally, some evaluations of the BER obtainable with the proposed approach are provided by using as reference a commercial SDRAM characterized for space application. The evaluations are performed supposing that this memory system is used in a satellite positioned in a geosynchronous orbit.

## References

[1] G.M. Swift, D.J. Padgett, and A.H. Johnston, "A new class of single event hard errors [DRAM cells]," *IEEE Transactions on Nuclear Science*, vol. 41, no. 6 Part 1, pp. 2043–2048, 1994.

[2] A.H. Johnston, "The influence of VLSI technology evolution on radiation-inducedlatchup in space systems," *IEEE Transactions on Nuclear Science*, vol. 43, no. 2 Part 1, pp. 505–521, 1996.

[3] T.E. Page Jr, J.M. Benedetto, and R.M. Syst, "Extreme latchup susceptibility in modern commercial-off-the-shelf (COTS) monolithic 1M and 4M CMOS static random-access memory (SRAM) devices," in *IEEE Radiation Effects Data Workshop, 2005*, 2005, pp. 1–7.

[4] C.W. Slayman, "Cache and memory error detection, correction, and reduction techniques for terrestrial servers and workstations," *IEEE Transactions on Device and Materials Reliability*, vol. 5, no. 3, pp. 397–404, 2005.

[5] E. Normand, "Single-event effects in avionics," *IEEE Transactions on nuclear science*, vol. 43, no. 2 Part 1, pp. 461–474, 1996.

[6] W.D. Armitage, J.C. Lo, M.T. Center, and R.I. Coll, "Erasure error correction with hardware detection," in *Defect and Fault Tolerance in VLSI Systems, 1999. DFT'99. International Symposium on*, 1999, pp. 293–301.

[7] G.C. Cardarilli, F. Lombardi, M. Ottavi, S. Pontarelli, M. Re, and A. Salsano, "A Comparative Evaluation of Designs for Reliable Memory Systems," *Journal of Electronic Testing*, vol. 21, no. 4, pp. 429–444, 2005.

[8] R.E. Blahut, "Theory and practice of error control codes," *Reading, Massachusetts*, 1984.

[9] R. Harboe-Sorensen, F.X. Guerre, and G. Lewis, "Heavy-Ion SEE Test Concept and Results for DDR-II Memories," *IEEE Transactions on Nuclear Science*, vol. 54, no. 6 Part 1, pp. 2125–2130, 2007.

[10] F. Irom and D.N. Nguyen, "Single Event Effect Characterization of High Density Commercial NAND and NOR Nonvolatile Flash Memories," *IEEE Transactions on Nuclear Science*, vol. 54, no. 6 Part 1, pp. 2547–2553, 2007.

[11] W.K.S. Walker, C.E.W. Sundberg, and C.J. Black, "A reliable spaceborne memory with a single error and erasure correction scheme," *IEEE Transactions on Computers*, vol. 100, no. 28, pp. 493–500, 1979.

[12] M.Y. Hsiao, "A class of optimal minimum odd-weight-column SEC-DED codes," *IBM Journal of Research and Development*, vol. 14, no. 4, pp. 395–401, 1970.

[13] P.E. Dodd, M.R. Shaneyfelt, J.R. Schwank, and G.L. Hash, "Neutron-induced soft errors, latchup, and comparison of SER test methods for SRAM technologies," in *Electron Devices Meeting, 2002. IEDM'02. Digest. International*, 2002, pp. 333–336.

[14] P.E. Dodd, M.R. Shaneyfelt, J.R. Schwank, and G.L. Hash, "Neutron-induced latchup in SRAMs at ground level," in *2003 IEEE International Reliability Physics Symposium Proceedings, 41st Annual*, 2003, pp. 51–55.

[15] J. Baggio, D. Lambert, V. Ferlet-Cavrois, C. D'hose, K. Hirose, H. Saito, J.M. Palau, F. Saigne, B. Sagnes, N. Buard, et al., "Neutron-induced SEU in bulk and SOI SRAMs in terrestrial environment," in *2004 IEEE International Reliability Physics Symposium Proceedings, 2004. 42nd Annual*, 2004, pp. 677–678.

[16] R. Hillman, G. Swift, P. Layton, M. Conrad, C. Thibodeau, and F. Irom, "Space processor radiation mitigation and validation techniques for an 1,800 MIPS processor board," in *Radiation and Its Effects on Components and Systems, 2003. RADECS 2003. Proceedings of the 7th European Conference on*, 2003, pp. 347–352.

[17] F. Vargas and M. Nicolaidis, "SEU-tolerant SRAM design based on current monitoring," in *Fault-Tolerant Computing, 1994. FTCS-24. Digest of Papers., Twenty-Fourth International Symposium on*, 1994, pp. 106–115.

[18] J. M. Soden, "Iddq testing for submicron cmos ic technology qualification," in *IDDQ Testing, Digest of Papers, IEEE International Workshop on*, 1997, pp. 52–56.

[19] AJ Tylka, JH Adams Jr, PR Boberg, B. Brownstein, WF Dietrich, EG Fluckiger, EL Peterson, MA Shea, DF Smart, and EC Smith, "CREME-96; A revision of the cosmic ray micro-electronics code," *IEEE Trans. Nucl. Sci*, vol. 44, pp. 2150–2160, 1996.