Effectiveness and failure modes of error correcting code in industrial 65 nm CMOS SRAMs exposed to heavy ions (Postprint)

Authors: TONG Teng, WANG Xiao-Hui, Zhan-Gang Zhang, DING Peng-Cheng, LIU Jie, LIU Tian-Qi, SU Hong

Date: 2023-06-18T00:00:00+00:00

Abstract

Single event upsets (SEUs) induced by heavy ions were observed in 65 nm SRAMs to quantitatively evaluate the applicability and effectiveness of single-bit error correcting code (ECC) utilizing Hamming Code. The results show that the ECC did improve the performance dramatically, with the SEU cross sections of SRAMs with ECC being at the order of 10-11 cm2/bit, two orders of magnitude higher than that without ECC (at the order of 10-9 cm2/bit). Also, ineffectiveness of ECC module, including 1-, 2- and 3-bits errors in single word (not Multiple Bit Upsets), was detected. The ECC modules in SRAMs utilizing (12, 8) Hamming code would lose work when 2-bits upset accumulates in one codeword. Finally, the probabilities of failure modes involving 1-, 2- and 3-bits errors, were calcaulated at 39.39%, 37.88% and 22.73%, respectively, which agree well with the experimental results.

Full Text

Preamble

NUCLEAR SCIENCE AND TECHNIQUES 25, 010405 (2014)

Effectiveness and Failure Modes of Error Correcting Code in Industrial 65 nm CMOS SRAMs Exposed to Heavy Ions

TONG Teng[1],[2], WANG Xiao-Hui[1],[2], ZHANG Zhan-Gang[1],[2], DING Peng-Cheng[1],[3], LIU Jie[1], LIU Tian-Qi[1],[2], and SU Hong[1],†

[1]Institute of Modern Physics, Chinese Academy of Sciences, Lanzhou 730000, China
[2]University of Chinese Academy of Sciences, Beijing 100049, China
[3]Northwest Normal University, Lanzhou 730000, China

(Received July 23, 2013; accepted in revised form October 8, 2013; published online February 20, 2014)

Abstract: Single event upsets (SEUs) induced by heavy ions were observed in 65 nm SRAMs to quantitatively evaluate the applicability and effectiveness of single-bit error correcting code (ECC) utilizing Hamming Code. The results show that the ECC dramatically improved performance, with SEU cross sections of SRAMs with ECC on the order of 10^{-11} cm^2/bit—two orders of magnitude lower than those without ECC (on the order of 10^{-9} cm^2/bit). However, ineffectiveness of the ECC module was detected, including 1-, 2-, and 3-bit errors in a single word (not Multiple Bit Upsets). The ECC modules in SRAMs utilizing (12, 8) Hamming code cease to function when 2-bit upsets accumulate in one codeword. Finally, the probabilities of failure modes involving 1-, 2-, and 3-bit errors were calculated at 39.39%, 37.88%, and 22.73%, respectively, which agree well with the experimental results.

Keywords: Single event upsets (SEU), SRAM, Error correcting code (ECC), Hamming code, Effectiveness, Failure modes
DOI: 10.13538/j.1001-8042/nst.25.010405

Introduction

As technology scales downward in modern integrated circuits such as SRAM, the minimum charge needed to upset a device within a unit memory cell decreases while the influence of charge sharing on adjacent unit memory cells increases [1-5]. Consequently, advanced devices (especially deep-submicrometer) exhibit much greater sensitivity to energy deposition from heavy ion irradiation, which critically restricts their use in space applications.

Many methods have been proposed to mitigate single event upsets (SEUs) in advanced devices. Bit interleaving architecture is a commonly accepted approach to mitigate Multiple Bit Upsets (MBUs) in data words. In this architecture, bits in a data word are not physically adjacent but are interleaved with bits from other data words, transforming every MBU of physically adjacent memory cells into multiple single-bit upsets (SBUs) in different memory words.

Error correcting code (ECC) utilizing Hamming code is commonly found in many high-reliability and performance applications. As a relatively simple yet powerful ECC, it can correct single-bit errors anywhere within a codeword. Therefore, MBUs—which represent a major reliability problem in commercial and industrial electronics—can be transformed into multiple SBUs that appear as uncorrelated events relative to the ECC algorithm and subsequently be corrected [2, 5-7].

In this hardening approach, ECC modules can be used in high-reliability and performance applications to resolve SBUs when combined with bit interleaving architecture in advanced process node devices. To observe and compare SEUs induced by heavy ions in SRAMs of different process nodes and to quantitatively evaluate the applicability and effectiveness of single-bit ECC utilizing Hamming code in advanced process SRAMs, we irradiated four SRAMs from ISSI company with a ^{12}C ion beam. Two of them, manufactured via 130 nm and 150 nm processes, represent the most advanced process devices in their SRAM families without ECC modules, while the other two are 65 nm process SRAMs with ECC modules. Some interesting results were obtained.

Experimental Background

Four industrial SRAMs produced by high-performance CMOS technology were irradiated at normal incidence in vacuum by ^{12}C beams from the Heavy Ion Research Facility in Lanzhou (HIRFL). The ^{12}C ions had an effective linear energy transfer (LET) value of 1.8 MeV$\cdot$cm^2/mg. Table 1 shows the information for the SRAMs under test. The IS2ME is a 2 Mbit SRAM organized as 131,072 words by 16 bits with ECC, 65 nm process node; the IS4ME is a 4 Mbit SRAM organized as 262,144 words by 16 bits with ECC, 65 nm process node; the IS2M is a 2 Mbit SRAM organized as 131,072 words by 16 bits without ECC, 150 nm process node; and the IS4M is a 4 Mbit SRAM organized as 262,144 words by 16 bits without ECC, 130 nm process node. The first two SRAMs with ECC are the main objects of observation, and the other two serve as comparative devices. All four industrial SRAMs belong to the IS61WV series made by ISSI company, and the ECC functions described in this application are implemented using Hamming code, a relatively simple yet powerful ECC that can correct all single-bit errors in one codeword.

The SRAMs were tested using a data pattern of all “1” (blank pattern) at a voltage of 3.3 V, and the operating frequency was set at 20 MHz throughout. Under static test mode, the devices were written prior to beam exposure and read periodically throughout the beam shot (this technique is often referred to as multiple-read) [1, 8, 9]. Error data occurring during the test were stored in another RAM (referred to as mirrored RAM relative to the SRAM under test) operating in the test system, serving as reference data for the next read cycle. The test flow applied (Fig. 1 [Figure 1: see original paper]) distinguishes SBU, MBU, and SEL. All upset events were recorded with a timestamp and bitmap location.

Results Analysis and Discussion

A. The High Efficiency of ECC Module

SEU cross sections of the four SRAMs are shown in Fig. 2 [Figure 2: see original paper]. The SRAMs without ECC module are much more sensitive to irradiation than devices with ECC module. The SEU cross sections of SRAMs without ECC module are on the order of 10^{-9} cm^2/bit, while they are 10^{-11} cm^2/bit for SRAMs with ECC module. However, the technology used to produce the IS2ME and IS4ME in 65 nm process is more advanced than that of IS2M (150 nm process node) and IS4M (130 nm process node). With technology scaling, the number of upsets per chip increases due to higher circuit density and sensitivity. Therefore, the sharp contrast between the two data groups should be attributed to the high efficiency of the ECC module.

B. The Ineffectiveness of ECC Module

Only 1-bit upsets in a data word were detected in devices without ECC module in this experiment. Upset events involving 1-, 2-, and 3-bit errors occurred in devices with ECC module. Fig. 3 [Figure 3: see original paper] shows the measured and theoretical results of bits per upset event distribution (percentages over total events). We will discuss the results with special attention to the distinction between the terms "upset" and "error" in the following text— "upset" refers to the actual change that occurred in a memory cell, while "error" refers to the data finally read out from memory.

1. The Fundamental Reason The results show that the fundamental reason for the problem is that a 2-bit upset in a codeword causes the disablement of the ECC module utilizing (12, 8) Hamming code. For discussing the experimental results, we make the following assumptions:

2. Parsing the Problem First, considering the beam energy of ^{12}C and the bit interleaving architecture, the normal incidence ion beams do not affect adjacent memory cells simultaneously. Therefore, MBUs are not expected to occur in a codeword at any time in this experiment [2, 5–7].

Second, the static mode used in this test means that only one write operation occurs in a test cycle, while the ECC module does not correct or rewrite the memory itself [1] but only corrects the "error" bit(s). When data are read out through the ECC module, the memory remains in upset status until a new write command arrives with new data. Therefore, if other bit(s) upset occurs in the same word, the ECC module utilizing Hamming code, which can only correct one-bit errors, will lose function. Thus, the disablement of the ECC module is an accumulation effect caused by several SBUs in a word at different times. On the other hand, as the ECC functional block diagram shows (Fig. 4 [Figure 4: see original paper], presented in the datasheet of devices with ECC module), the circuit structure of the ECC module utilizes the (12, 8) Hamming code in this application.

Based on the time structure of the cyclotron and the upstream scanning magnets, the incident ions have uniform temporal and spatial distribution in the used flux range, thus each SBU can be deemed an independent random event. In independent random events, if the upset probability is p (p 1), the probability that r bit(s) upset occurs in an n-bit codeword is P (r) = C(n,r) p (1-p)$^{n-}$ n!/(r!(n-r)!) p . From the results of IS2M and IS4M, about 200 ions could cause 1-bit upset in order of magnitude; assuming this probability is suitable for IS2ME and IS4ME, we have $p = 5 \times 10^{-3}$. Then, the probability of two and three SBUs occurring at different times in one codeword is:

$P_{12}(2)$ = C(12,2) p^2 $(1\text{-}p)^{10}$ = 12!/(2!10!) $(5 \times 10^{-3})^2 = 3.3 \times 10^{-4}$

$P_{12}(3)$ = C(12,3) p^3 $(1\text{-}p)^9$ = 12!/(3!9!) $(5 \times 10^{-3})^3 = 1.1 \times 10^{-6}$

The results of these equations show a probability difference of two orders of magnitude between r = 2 and r = 3. Thus, three or more SBUs occurring at different times in one codeword is of very low probability, hence their omission in this experiment.

3. Analysis Results Figure 5 [Figure 5: see original paper] shows a basic memory architecture of an ECC module utilizing Hamming code [10]. Table 2 shows the common relationship between syndrome vector and single-error location.

Assuming an 8-bit data word vector D = $D_7D_6D_5D_4P_3D_3D_2D_1P_2D_0P_1P_0$ and a 4-bit check word vector P, the syndrome vector S can be generated from the data word and check word as [11]:

$S_0 = D_0 \quad D_1 \quad D_3 \quad D_4 \quad D_6 \quad P_0$
$S_1 = D_0 \quad D_2 \quad D_3 \quad D_5 \quad D_6 \quad P_1$
$S_2 = D_1 \quad D_2 \quad D_3 \quad D_7 \quad P_2$
$S_3 = D_4 \quad D_5 \quad D_6 \quad D_7 \quad P_3$

The corresponding (12, 8) parity matrix is:

When an 8-bit data word is written in SRAM, the ECC module generates a 4-bit check word to compose a 12-bit codeword and stores it in the memory cell. After irradiation, when the data word is read out from the memory cell through the ECC module, the module generates a syndrome vector S = $(S_3S_2S_1S_0)$ according to the codeword.

In the parity matrix, each column vector represents the position of each bit (D where u = 0, 1, ···, 7 or P where v = 0, 1, 2, 3) in the codeword; 0 means the bit does not participate in the formation of S (k = 0, 1, 2, 3), while 1 means it does participate. How then does a 2-bit change in the codeword generate S $\neq$ (0000), and how does S point to an error in Table 2? The method to find the failure modes is discussed as follows:

First, if neither of the 2 upset bits participates in S , then S = 0 0 = 0, pointing to "no error."

Second, if both upset bits participate in S , then S = 1 1 = 0, also pointing to "no error."

Third, if only one upset bit participates in S , then S = 1 0 = 1 or S = 0 1 = 1; the value of the corresponding S is always 1, so the ECC module spots an "error" and performs a "correction" operation.

Consequently, the S value is associated with the status of the 2 upset bits participating in S through an XOR operation between S and the 2 upset bits.

For example, if the 2-bit upset involves D_3 and P_0, they will not affect the value of S_0 (as both participate in it) or S_3 (as neither participates in it). However, $S_1 = P_1 \quad D_0 \quad D_2 \quad D_3' \quad D_5 \quad D_6$ and $S_2 = P_2 \quad D_1 \quad D_2 \quad D_3' \quad D_7$ will result in $S = (S_3S_2S_1S_0) = (0110)$, which can be understood simply as:

This syndrome vector points to an "error" position at D_2 according to Table 2. The ECC module then "corrects" the correct value of D_2 to an erroneous value, while the real upset bit D_3 is read out as "correct" data, leading to 2-bit errors in D_3D_2. In other words, if the data written in is "FF," the data read out is "F3" as an error to be detected.

Therefore, the problem-solving method can be simplified as the following procedures: (1) extract two column vectors from the parity matrix of Eq. (5) (2-bit upsets occurring in the same bit in a codeword do not affect S, hence this condition is omitted); (2) perform an XOR operation on them as in Eq. (6); (3) produce the syndrome vector S; (4) find the "error" position where S points; (5) analyze the relationship between the "error" and the "upset" ; and (6) achieve a statistical analysis of failure modes including 1-bit, 2-bit, and 3-bit errors read out from the SRAMs.

Extracting two column vectors from the parity matrix of Eq. (4), the total number of error types is C(12,2) = 66. Tables 3–5 list details of the failure modes and error types.

Case 1: When 2-bit upset occurs in both check bits (Table 3)

In this case, the ECC module makes a wrong operation. The number of error types is C(4,2) = 6, and all failure modes result in 1-bit errors.

Case 2: When 1-bit upset occurs in check word and 1-bit upset in data word (Table 4)

In this case, the ECC module makes a wrong operation. The number of error types is C(4,1) × C(8,1) = 32, of which 20 result in 1-bit errors and 12 result in 2-bit errors. The failure modes include both 1-bit and 2-bit errors.

Case 3: When 2-bit upset occurs in data word (Table 5)

In this case, the ECC module makes a wrong operation. The number of error types is C(8,2) = 28, of which 13 result in 2-bit errors and 15 result in 3-bit errors. The failure modes include both 2-bit and 3-bit errors.

Therefore, the total number of 1-bit error types is 6 + 20 = 26, giving a probability of 26/66 = 39.39% among all error types. The total number of 2-bit error types is 12 + 13 = 25, giving a probability of 25/66 = 37.88%. The total number of 3-bit error types is 15, giving a probability of 15/66 = 22.73%. Table 6 shows that the theoretical probabilities of failure modes including 1-, 2-, and 3-bit errors agree well with the experimental results.

Thus, the inherent factor causing failure modes of the ECC module in this experiment is the inability of the (12, 8) Hamming code to handle 2-bit upsets in one codeword.

Conclusion

The results demonstrate both the effectiveness and ineffectiveness of ECC modules utilizing (12, 8) Hamming code in 65 nm process node SRAMs. The ECC module clearly works to harden the advanced process node SRAMs. The failure modes including 1-, 2-, and 3-bit errors in a data word have been analyzed, and the essential factor causing these failure modes is the inability of the (12, 8) Hamming code to handle 2-bit upsets in one codeword. The measured bits-per-upset event distribution agrees well with theoretical calculations.

Several mitigation approaches can be employed if much higher reliability is required. Periodic memory scrubbing is often used to improve device performance, and a scrubbing operation will be conducted on SRAMs exposed to heavy ions in our laboratory to observe the relationship between scrub rates and bit error rate (BER). If more redundancy is acceptable, triple-bit-correcting Golay code or Triple Modular Redundancy (TMR) may be employed.

This research on 65 nm SRAMs may provide a reference for manufacturers in their choice of reinforcement models and algorithms, and for users in their selection of device application environments and methods.

Acknowledgments

The authors thank LIU Xin and ZHAO Fa-Zhan from the Institute of Microelectronics, Chinese Academy of Sciences, for discussions on the failure modes of Hamming code, and the staff of the HIRFL accelerator for experimental assistance.

References

[1] Lawrence R K and Kelly A T. IEEE Trans Nucl Sci, 2008, 55: 3394-3400.
[2] Heidel D F, Marshall P W, Pellish J A, et al. IEEE Trans Nucl Sci, 2009, 56: 3499-3504.
[3] Schrimpf R D, Weller R A, Mendenhall M H, et al. Nucl Instrum Meth B, 2007, 261: 1133-1136.
[4] Liu J, Duan J L, Hou M D, et al. Nucl Instrum Meth B, 2006, 245: 342-345.
[5] Bajura M A, Boulghassoul Y, Naseer R, et al. IEEE Trans Nucl Sci, 2007, 54: 935-945.
[6] Radaelli D, Puchner H, Wong S, et al. IEEE Trans Nucl Sci, 2005, 52: 2433-2437.
[7] Juan Antonio Maestro and Pedro Reviriego, Study of the Effects of MBUs on the Reliability of a 150 nm SRAM Device, DAC' 08 Proceedings of the 45th annual Design Automation Conference, p.930-935, California, USA, June 8-13, 2008.

[8] Measurement and Reporting of Alpha Particle and Terrestrial Cosmic Ray-Induced Soft Errors in Semiconductor Devices, JESD89A, 2006, p.10.
[9] Palomo F R, Morilla Y, Mogollón J M, et al. Nucl Instrum Meth B, 2011, 269: 2210-2216.
[10] Nicolaidis M. Soft errors in modern electronic systems, Germany, Springer, 2011, p.207.
[11] Tam S. Single error correction and double error detection, Xilinx, XAPP645 (v2.2), 2006.

Note: Figure translations are in progress. See original paper for figures.

Source: ChinaXiv —Machine translation. Verify with original.