JOURNAL OF SEMICONDUCTOR TECHNOLOGY AND SCIENCE, VOL.15, NO.5, OCTOBER, 2015
http://dx.doi.org/10.5573/JSTS.2015.15.5.445

ISSN(Print) 1598-1657
ISSN(Online) 2233-4866

An Efficient Error Detection Technique for 3D Bit-Partitioned SRAM Devices

Heung Sun Yoon, Jong Kang Park, and Jong Tae Kim*

***Abstract*—As the feature sizes and the operating charges continue to be scaled down, multi-bit soft errors are becoming more critical in SRAM designs of a few nanometers. In this paper, we propose an efficient error detection technique to reduce the size of parity bits by applying a 2D bit-interleaving technique to 3D bit-partitioned SRAM devices. Our proposed bit-interleaving technique uses only 1/*K* (where *K* is the number of dies) parity bits, compared with conventional bit-interleaving structures. Our simulation results show that 1/*K* parity bits are needed with only a 0.024-0.036% detection error increased over that of the existing bit-interleaving method. It is also possible for our technique to improve the burst error coverage, by adding more parity bits.**

***Index Terms*—3D-integrated SRAM, soft error, EDC, ECC, EDAC**

I. INTRODUCTION

Semi-conductor integration technologies have faced various challenges to keep pace with Moore's Law and the More Moore domain. Technologies have been developed for reducing power consumption and enlarging IC transistor capacity by shrinking geometrical scaling. However, this technology has met the physical limitation of less than 10-nanometer processes, where the design and operation of small-scale transistors becomes seriously affected by reliability problems.

3D integration is an emergent technology to overcome this limitation. 3D integrated design is expected to offer advantages of high capacity, low power consumption, high performance, and heterogeneous system integration. 3D fabrication involves stacking two or more dies. The connections between dies should be maintained by means of very high density and a low-latency interface. Wire bonding has also been used for implementing physical interconnection in 3D integration, but Through Silicon Via (TSV) provides advantages in terms of performance and cost [1].

With scaling down of the physical structures of the transistor, the problem of reliability and yield has emerged, and have resulted in many efforts being made to ensure reliability. Errors in semiconductor memories can be classified into hard or soft errors. Hard errors occur either at manufacturing time, or in the field. Hard errors are permanent defects, unlike soft errors, which are transient or intermittent, but recoverable defects. There are several root causes for soft errors, such as power (or signal) supply noise coupling, and high-energy neutrons from cosmic rays colliding with particles in the atmosphere [2]. Despite the existence of only a few temporal errors, this phenomenon may result in fatal failure of the whole system. Error detecting codes (EDC) and error correcting codes (ECC) are used for detecting and correcting the soft errors, respectively. With the scaling down of the technology, state-of-the-art VLSI designs are required for low supply voltage and essentially the corresponding critical charges continue to be reduced. As a result, soft errors in memory will increase through radiation effects. In the past, these

Manuscript received Apr. 12, 2015, accepted Sep. 21, 2015
School of Electronic and Electrical Eng., Sungkyunkwan Univ., 300 Cheoncheon-dong Jangan-gu, Suwon, Gyeonggi-do 440-746, South Korea
E-mail : jtkim@skku.edu

problems were mainly the concern of aerospace and automotive applications, but consumer electronic devices might now be also vulnerable to these soft errors. Furthermore, the proportion of multiple-bit upset (MBU) is increasing in the smaller feature sizes [3-5].

Soft errors occur not only in planar components, but also in 3D-integrated components. We could apply conventional error detection and correction (EDAC), which is used for planar structures, to detect and correct soft errors for 3D-integraed SRAM. However, there is an opportunity for an efficient ECC technique that takes advantage of 3D integration. Existing study of ECC for 3D RAM has assumed that the high-energy particles stroke semiconductor devices from the top to bottom of the devices [6]. In this paper, we propose an efficient error detection technique for 3D-integrated bit-partitioned SRAM to save significant die area by reducing the number of parity bits compared with the conventional SRAM EDC scheme. A high-speed SRAM device can be utilized for the register file and cache memory system. Since the copy of the memory is always located in the lower memory hierarchy, it is not necessary to have an error correction feature in such SRAM devices. The proposed method in this paper can be applied to these applications. The main idea of our technique is to apply a two-dimensional bit-interleaving technique to 3D memory structures. This method provides an efficient solution, reduces the large amount of parity bits that protect the memory cells in different dies, while maintaining a burst error coverage in a die. In our simulation, the particle that on its trajectory of the device layers generates burst errors to adjacent memory cells, is refracted to random directions.

II. SRAM Structures and EDAC Techniques

This section briefly describes the primary 3D SRAM structures, including bit-partitioned technique, and the conventional EDAC technique for planar SRAM devices.

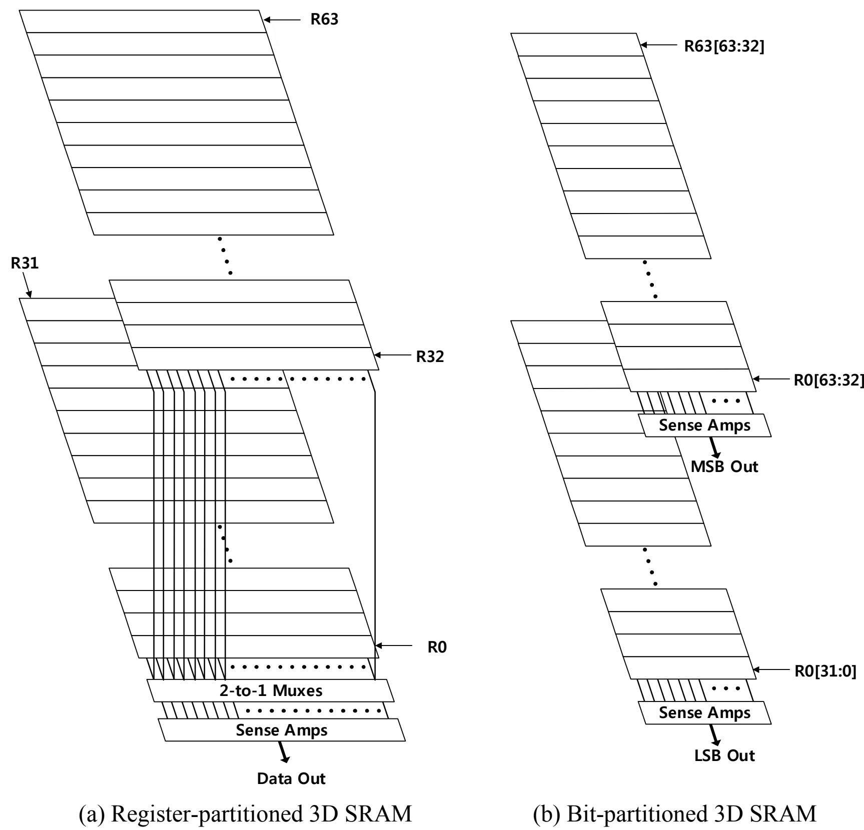
1. 3D-Integrated SRAM structures

3D-integrated SRAM designs can be classified into banked SRAM arrays and multi-ported SRAM arrays [7]. The memory banking technique divides the memory array into multiple sub-modules. A reduction in power consumption can be achieved by dividing the memory array into multiple banks and accessing only the bank that contains the required data. A previous study proposed four different designs of 3D multi-ported SRAM: register partitioning (RP), bit partitioning (BP), port splitting (PS), and hybrid configurations [7].

Consider the case of a 64-bit 64-entry register file. The register partitioned (RP) 3D SRAM array with a two-die stack consists of a bottom die that contains R0-R31, and a top die that contains R32-R63, as Fig. 1(a) shows. The vertical distance has been halved in the RP design. As a result, the length of the critical path is reduced, which also reduces the latency and power consumption. The row decoder height and overall footprint of the register file have also been halved. The bit-partitioned (BP) 3D SRAM design stacks higher-order and lower-order bits of the same register across different dies. Fig. 1(b) shows that the bottom die contains lower-order bits (0-31) of R0-R63, and the top die contains higher-order bits (32-63) of R0-R63. The horizontal distance (wordline) has been halved in BP. As a result, the gate loading (latency) and power consumption have also been reduced. The industry continues to scale down SRAM memory cells to improve their capacity. However, the capacity is significantly degraded by implementing multiple ports that support multiple read and write operations. If the number of ports is increased in a planar system, the area-per-bit is dominated by the wordlines and bitlines for implementing multiple read and write ports. Port splitting (PS) is one of the 3D SRAM designs. Each die contains bitlines, wordlines, and access transistors. Its footprint is fixed with an increased number of ports, by placing each port in its respective die. Accordingly, it is possible to obtain a large footprint reduction.

2. Soft errors and hardening technique

With the scaling down of transistors to the nanometer regime, soft errors occurs more often. As these soft errors may cause a fatal system error, we use various EDAC schemes to enhance the system reliability. The single event effect (SEE) occurs when a high-energy particle passes through a reverse biased PN junction. If a SEE occurs in a memory array, the stored bit may be flipped. This phenomenon is called a single event upset (SEU). If



(a) Register-partitioned 3D SRAM (b) Bit-partitioned 3D SRAM

Fig. 1. 3D Multi-ported SRAM Structures

several bits are flipped, burst errors occur in memory, which process is called a multiple-bit upset (MBU). The scaling down of size has increased the probability of MBU.

There are many studies for efficient EDC schemes to detect soft errors with small parity bits and latency. Error correcting and detecting codes are widely used in memory systems to increase reliability. Conventional commercial SRAM applies SECDED for ECC [9, 10]. SECDED is based on the Hamming code and Hsiao code [11]. It does not matter whether the parity bits are corrupted or not, because whenever parity bits are corrupted, this type of block code is able to detect and correct error. In spite of the ease of use, a large quantity of parity bits is required, together with a longer clock period, due to the integrity check of codes.

Some memory structures use physical bit-interleaving for higher performance and multi-bit error detection. Multiple data words are stored along a single physical row of the cell array in a bit-interleaved fashion. *N*-way interleaving is designed to strengthen against burst error. For example, as Fig. 2 shows, it is possible to detect a

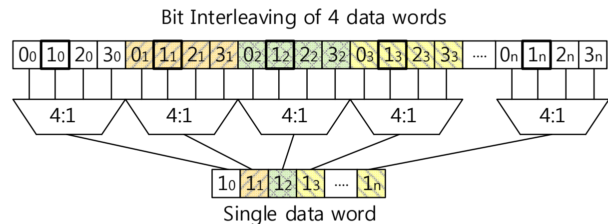


Fig. 2. Four-way interleaving

four-bit burst error by applying four-way interleaving. Each of four words have sparse layout. The adjacent four bits are all different words. Four-way interleaving has four parity bits for four data words. Each parity bit protects each word. Even if a four-bit burst error occurs, there is only one bit error in each word. Therefore, we can detect multiple errors, by checking multiple interleaved data and parity.

A previous study proposed two-dimensional (2D) coding of EDC to reduce the number of parity bits and hardware overhead [12]. 2D coding is the combination of horizontal per-word error coding with vertical column-wise coding; for example, four-way interleaved EDC-8

with vertical EDC-32. The 2D coding is able to correct any clustered multi-bit error that does not span more than 32 bits in both the horizontal and vertical directions. 2D coding detects errors by using horizontal parity bits and then correct errors by using vertical parity bits. Thus, all vertical parity bits should be read for correction.

III. A BIT-INTERLEAVING TECHNIQUE FOR 3D BIT-PARTITIONED SRAM

This section explains our efficient bit-interleaving technique for 3D bit-partitioned SRAM structures, which is based on the 2D bit-interleaving design for SRAM in our previous study [13].

1. 2D bit-interleaving for planar structures

Our previous work suggested 2D bit-interleaving technique for planar SRAM structures [13]. In this structure, each word is located horizontally off *N*-bit as well as vertically. Fig. 3 represents 2D four-way interleaving with 16 words of 16 bit length. Each word is distributed across four lines. It has a shifted structure every 4 lines, in order to protect data and parity bits together from any pattern of $N \times N$ burst errors. In Fig. 3, the large left-side number and small right-side number in a memory cell represent the word number and bit number of each word, respectively. The 1st, 5th, 9th, and 13th lines contain *word0*, *word1*, *word2*, and word3, respectively. The *p0*, *p1*, *p2*, and *p3* parity bits are regarding *word0*, *word1*, *word2*, and *word3*, respectively. As a result, the structure can detect four-bit burst errors with $1/N$ parity bits, in comparison with conventional four-way interleaving. But when reading one entire word, a four-line read operation is necessary. Thus, significant hardware and cycle overhead are needed for implementing 2D *N*-way interleaving in a planar SRAM.

0_{0}	1_{0}	2_{0}	3_{0}	0_{1}	1_{1}	2_{1}	3_{1}	0_{2}	1_{2}	2_{2}	3_{2}	0_{3}	1_{3}	2_{3}	3_{3}	p0
4_{0}	5_{0}	6_{0}	7_{0}	4_{1}	5_{1}	6_{1}	7_{1}	4_{2}	5_{2}	6_{2}	7_{2}	4_{3}	5_{3}	6_{3}	7_{3}	p4
8_{0}	9_{0}	10_{0}	11_{0}	8_{1}	9_{1}	10_{1}	11_{1}	8_{2}	9_{2}	10_{2}	11_{2}	8_{3}	9_{3}	10_{3}	11_{3}	p8
12_{0}	13_{0}	14_{0}	15_{0}	12_{1}	13_{1}	14_{1}	15_{1}	12_{2}	13_{2}	14_{2}	15_{2}	12_{3}	13_{3}	14_{3}	15_{3}	p12
1_{4}	2_{4}	3_{4}	0_{5}	1_{5}	2_{5}	3_{5}	0_{6}	1_{6}	2_{6}	3_{6}	0_{7}	1_{7}	2_{7}	3_{7}	0_{4}	p1
5_{4}	6_{4}	7_{4}	4_{5}	5_{5}	6_{5}	7_{5}	4_{6}	5_{6}	6_{6}	7_{6}	4_{7}	5_{7}	6_{7}	7_{7}	4_{4}	p5
9_{4}	10_{4}	11_{4}	8_{5}	9_{5}	10_{5}	11_{5}	8_{6}	9_{6}	10_{6}	11_{6}	8_{7}	9_{7}	10_{7}	11_{7}	8_{4}	p9
13_{4}	14_{4}	15_{4}	12_{5}	13_{5}	14_{5}	15_{5}	12_{6}	13_{6}	14_{6}	15_{6}	12_{7}	13_{7}	14_{7}	15_{7}	12_{4}	p13
2_{8}	3_{8}	0_{9}	1_{9}	2_{9}	3_{9}	0_{10}	1_{10}	2_{10}	3_{10}	0_{11}	1_{11}	2_{11}	3_{11}	0_{8}	1_{8}	p2
6_{8}	7_{8}	4_{9}	5_{9}	6_{9}	7_{9}	4_{10}	5_{10}	6_{10}	7_{10}	4_{11}	5_{11}	6_{11}	7_{11}	4_{8}	5_{8}	p6
10_{8}	11_{8}	8_{9}	9_{9}	10_{9}	11_{9}	8_{10}	9_{10}	10_{10}	11_{10}	8_{11}	9_{11}	10_{11}	11_{11}	8_{8}	9_{8}	p10
14_{8}	15_{8}	12_{9}	13_{9}	14_{9}	15_{9}	12_{10}	13_{10}	14_{10}	15_{10}	12_{11}	13_{11}	14_{11}	15_{11}	12_{8}	13_{8}	p14
3_{12}	0_{13}	1_{13}	2_{13}	3_{13}	0_{14}	1_{14}	2_{14}	3_{14}	0_{15}	1_{15}	2_{15}	3_{15}	0_{12}	1_{12}	2_{12}	p3
7_{12}	4_{13}	5_{13}	6_{13}	7_{13}	4_{14}	5_{14}	6_{14}	7_{14}	4_{15}	5_{15}	6_{15}	7_{15}	4_{12}	5_{12}	6_{12}	p7
11_{12}	8_{13}	9_{13}	10_{13}	11_{13}	8_{14}	9_{14}	10_{14}	11_{14}	8_{15}	9_{15}	10_{15}	11_{15}	8_{12}	9_{12}	10_{12}	p11
15_{12}	12_{13}	13_{13}	14_{13}	15_{13}	12_{14}	13_{14}	14_{14}	15_{14}	12_{15}	13_{15}	14_{15}	15_{15}	12_{12}	13_{12}	14_{12}	p15

Fig. 3. 2D four-way interleaved structure for EDC

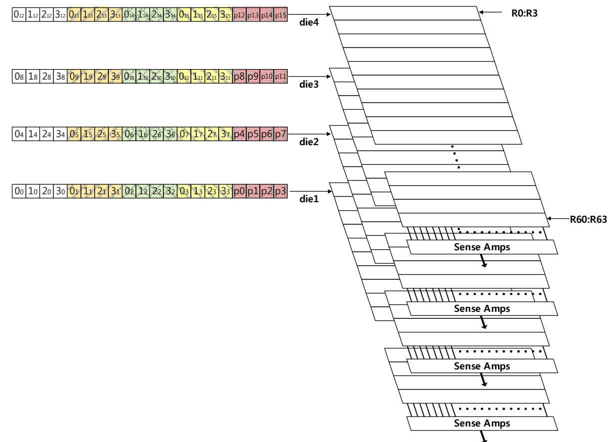


Fig. 4. 3D bit-partitioned SRAM with conventional bit-interleaving

2. 2D bit-interleaving for 3D bit-partitioned SRAM

We applied the aforementioned 2D bit-interleaving technique to 3D bit-partitioned SRAM. Multiple data words are stored in an interleaved fashion along a single physical row across all dies. 2D bit-interleaving in 3D bit-partitioned SRAM requires only *N* parity bits, where *L* bits of data are in a single physical row of the cell array, 2D *N*-way, and *K* dies, where each die contains $(L+N)/K$ bits. In the proposed method, there are *N* parity bits in a single physical row across all dies. In the case of conventional bit-interleaving, however, there are $N \times K$ parity bits in a single physical row across all dies. Thus, the total required number of parity bits in the proposed method to protect data words is only $1/K$ that of the conventional bit-interleaving.

Fig. 4 shows a four-die 64-bit 64-entry bit-partitioned SRAM in conventional four-way interleaving structure, where the bottom die stores the least significant 16 bits of data, the top die stores the most significant 16 bits, and each die contains four parity bits. For example, *p0*, *p4*, *p8*, and *p12* protect *R0* in *die1*, *die2*, *die3*, and *die4*,

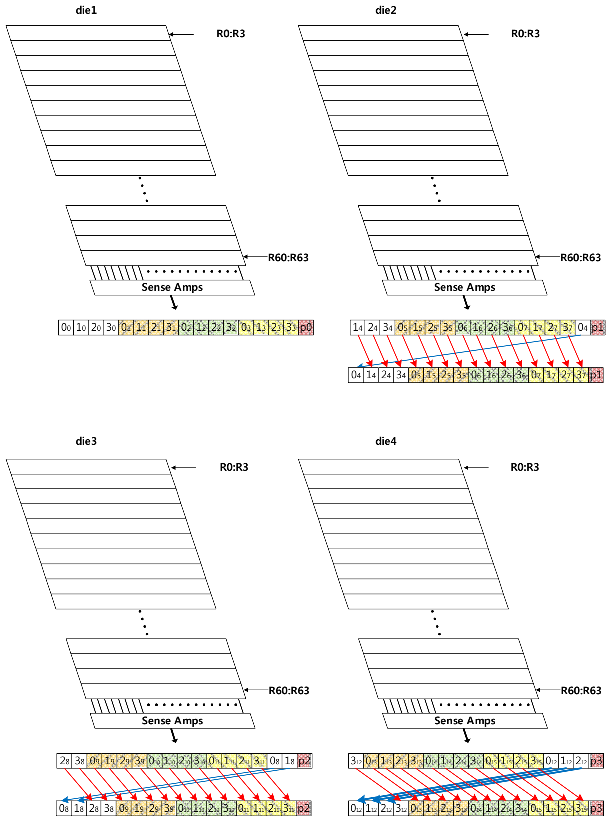


Fig. 5. 3D bit-partitioned SRAM with conventional 2D bit-interleaving

respectively. Thus, a total of 1024 parity bits are needed to protect 64×64 data bits.

When applying 2D four-way interleaving in Fig. 3, four lines should be placed in each die. We can read out four lines in one access by applying the layout shown in Fig. 5. This means we can eliminate the cycle overhead of original 2D interleaving. Each die contains one parity bit, and conventional 2D bit-interleaving has a shifted structure to protect data and parity bits together. Thus, the data and parity bits should be rearranged after reading. This rearrangement of routing paths causes additional latency overhead. The critical path is the bold line in *die4* of Fig. 5, where the SRAM cell width is w, the number of bits in a line is L, and the critical path overhead is estimated as $(L\text{-}N\text{-}1)\times w$ for N-way interleaving.

Alternatively, Fig. 6 shows a four-die 64-bit 64-entry bit-partitioned SRAM with 2D four-way interleaving, where the bottom die stores the least significant 17 bits of data bits, and the top die stores the most significant 13 bits. Only the top die contains four parity bits. The parity bits *p0*, *p1*, *p2*, and *p3* cover *R0*, *R1*, *R2*, and *R3* across 4

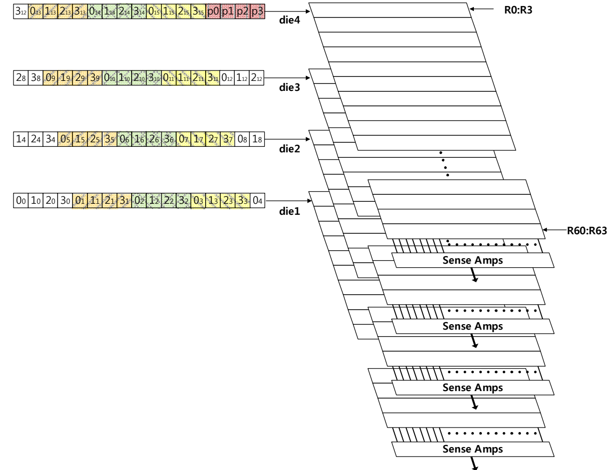


Fig. 6. 3D bit-partitioned SRAM with the proposed method

dies, respectively. A total of 256 parity bits are needed to protect 64×64 data bits. This EDC structure requires only one quarter the number of parity bits, as compared with that of conventional interleaving. Each die contains 17 bits including parity bits. Only *die4* has parity bits unlike in Figs. 4 and 5. A shifted structure caused the latency overhead of conventional 2D bit-interleaving in Fig. 5. However, we reduced the critical path overhead by placing all parity bits in the last die. All the rows shown in Fig. 6 can be concatenated into a single line that is exactly identical to the traditional interleaved structure depicted in Fig. 2.

The error detection capability of the proposed EDC scheme in a die is the same as that of the conventional bit-interleaving. When we read in 3D bit-partitioned SRAM, data and parity bits come out from all dies. A single data word and one parity bit are then extracted by using N-to-1 multiplexers. Errors can be detected by comparing parity bits and XOR output for all bits of a single data word. Since the parity bits are grouped in the vertical direction, a large amount of burst errors generated at different dies of the same row might not be covered by the proposed technique. We can see the related simulation results in Section 4.

IV. EXPERIMENTS

The advantage of the proposed EDC technique is that it only requires a small number of parity bits. Basically, it requires only $1/N$ parity bits, as compared with conventional N-way interleaving. This result originates

Table 1. Properties of 3D SRAM

TSV depth	100 μm
SRAM cell width	284 nm
Cache line	64 byte
Cache size	256 KB

Table 2. 3D SRAM die size

N, *K* (with *N*-way, *K* dies)	4	8	16
Logical die width (bit)	128	64	32
Logical die height (bit)	4,096	4,096	4,096
Physical die width (nm)	36,352	18,176	9,088
Physical die height (nm)	1,163,264	1,163,264	1,163,264

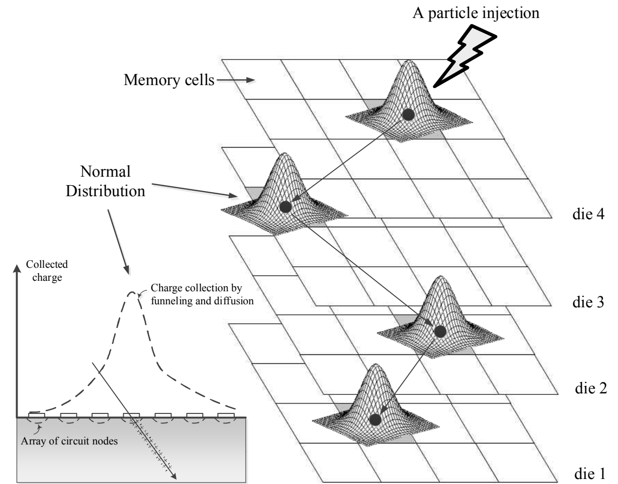


Fig. 7. The simulation procedure for multiple bit upsets due to a particle strike

from the enlarged parity bits coverage. In this section we show the comparative results for evaluating error detection probability of the proposed EDC.

1. Simulation setup

The simulation model for evaluating error detection capability considers particle propagation and the burst error model. We assume that when a high-energy particle is propagated in the 3D SRAM structure and the particle undergoes refraction in a random direction. We define the 3D SRAM properties, as summarized in Table 1, to determine the spatial structure of bit-partitioned 3D SRAM for simulating particle propagation. We reference the 3D SRAM properties to a state-of-the-art 3D SRAM system in ITRS 2013. We assume that the SRAM cell width and height are the same. Selecting the number of dies determines the 3D SRAM physical layout, based on Table 1. For example, the physical die size of a four die stack 3D SRAM is 1163264 nm × 36352 nm (die height × die width).

We simulated three cases in which *N* and *K* were equal to 4, 8, and 16, respectively. The physical die size changes according to *K*. Table 2 denotes each die size of SRAM in terms of bits and nanometers. We simulated the path of the particle propagation within the physical die size in terms of nanometers and error injection within cell logical die size in terms of bits. In a 3D bit-partitioned SRAM design, cache size and cache line determine die height. When we determine *K*, only die width changes. We simulated particle propagation with a spatial 3D bit partition layout determined by the parameters listed in Tables 1 and 2.

As shown in Fig. 7, if a high-energy particle strikes any spot in the die, the highest probability of soft error occurs at that spot. The probability decreases with distance from the spot. In our simulation, we model the probability of error occurrence as a probability density function (PDF) that is a normal distribution. The struck spot represents the mean of the normal distribution. We injected burst errors around the normal distribution, every time the incidence and reflected particle strikes to a die. If the applied protection technique detects up to $n \times n$ burst errors, the normal distribution is then set to make 98% of burst errors less than $n \times n$ errors. Thus, 98% of burst errors in a die are expected to be detected for the given EDC structures.

After the first strike, a high-energy particle was refracted in a random direction. If we know the next spot, we can determine a vector from the spot at which it first struck (origin spot) to the next spot. We generate the vector by randomly choosing a certain propagation angle θ (0~360°) and Φ (0~180°), where θ is the angle between x axis and projection on the x-y plane, and Φ is the angle between the vector and projection on the x-y plane. We can determine the next spot from the origin spot, and the certain propagation direction given by θ and Φ. We continue this process until the particle is left out of the device, and repeatedly run the particle simulation

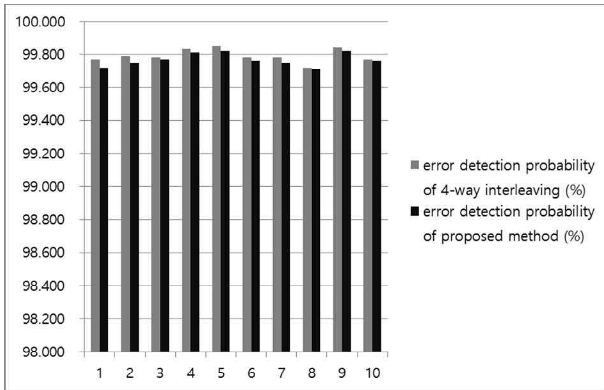


Fig. 8. Detection probability of 2D four-way interleaving of four-die SRAM

Table 3. Comparison of EDCs

EDC scheme (where N=K)	Parity bits (bit)	Detection probability(%)	Difference of detection probability(%)
Conventional four-way interleaving	65,536	99.791	0.024
Proposed four-way interleaving	16,384	99.767	
Conventional eight-way interleaving	262,144	99.771	0.036
Proposed eight-way interleaving	32,768	99.735	
Conventional 16-way interleaving	1,048,576	99.819	0.033
Proposed 16-way interleaving	65,536	99.786	

100,000 times to obtain an error detection probability for the given EDC structures.

2. Results and Discussion

We present the simulation results for various sets of parameters determined by the simulation model in Section 4.1. First of all, we compared the difference in error detection probability of conventional bit-interleaving and the proposed method.

Fig. 8 represents the simulation result of four-way interleaving four-die 3D bit-partitioned SRAM design with the parameters given in Tables 1 and 2. The gray and black bars denote error detection probability when applying conventional 1D four-way interleaving and the proposed method, respectively. Each bar implies 10,000 simulated particle propagations. The difference between the conventional and proposed detection probability is small enough to be negligible. In summary, for a total of 100,000 simulated particle propagations, the difference is only 0.024%. To implement conventional four-way interleaving, 65,536 parity bits are required to protect 256 KB. But only 16,384 parity bits are required to implement the proposed method. Applying the proposed method saves 49,152 bit SRAM cells with a 0.024% detection probability loss. 49,152 bits are an estimated 3964.4037 μm^2 in terms of area.

N-way interleaving can detect N burst errors. Selecting a larger N to protect larger burst errors may affect the error detection probability in the proposed method, because the occurrence of more burst errors means the particles have more energy. Thus, the probability of two or more errors occurring in the same parity-bits coverage will be larger. We experimented with N equal to 4, 8, and 16. Table 3 denotes each required number of parity bits and detection probabilities to protect 256 KB. Each result implies 100,000 simulated particle propagations. The parity bits of the proposed method require only $1/K$ parity bits of the conventional four-way interleaving.

All cases in Table 3 are K (where K is the number of dies) equal to N. The difference of detection probability (when $N = 8, 16$) slightly increased, compared with $N = 4$. According to the simulation results, the detection probability of the proposed method is slightly smaller by as much as 0.024% - 0.036%, compared with the conventional method. Given the fact that when assuming that less than or equal to N burst errors always occur, conventional bit-interleaving is detectable at 100%, 0.024% - 0.036% is not a large loss. According to the simulation results, when we apply a normal distributed error occurrence model, the error detection probability of conventional bit-interleaving is 99.791% (where $N = K = 4$). If we do not assume that the burst errors always occur less than or equal to N, conventional bit-interleaving has a 0.209% detection fail probability. The 0.024% detection probability error is acceptable, compared with the 0.209% detection fail probability in conventional bit-interleaving.

Because the particle injection rate to the target memory device, equals to 1.0 in this work, the detection error percentage should be lowered in real-world application. For example, multiplying neutron flux = 56.15n/m^2/s by effective injection rate = 2.2E-5, results in 0.0012n /m^2/s as referred to in [14]. For the proposed

16-way interleaving case in Table 3, 99.786% detection probability can be converted to about 0.1 FIT (Failure-In-Time) / device, where 1 FIT equals to 1 error per 1 billion hours. Without the error detection feature, 45.67 FIT / device can be observed in this case.

Regarding the design overheads, the proposed technique does not require actual rearrangement of memory bits. In case of conventional 2D bit-interleaving (Fig. 3), bit positions must be rearranged before 4-to-1 multiplexer. In spite of the shifted structure in the proposed method, the word and parity layout become still consecutive. For example, as shown in Fig. 6, the last bit of die 1 is 0_4 and the first bit of die 2 is 1_4. In this way, the order in bits keeps consecutive along to different dies. Four lines in four different dies are concatenated to a long line before 4-to-1 multiplexers and the resultant structure is the same as the one in Fig. 4.

Because the number of 4-to-1 multiplexers and the required operations are the same as the ones of the conventional 4-way interleaving (Figs. 2 and 4), the expected power consumption is not much different from each other. Static power consumption due to the reduced parity bits can be further decreased in the proposed method.

It is clear that more parity bits should be needed to protect SRAM cells from larger burst errors. When conventional bit-interleaving is applied, the required parity bits increase dramatically to protect larger area coverage. However, applying the proposed method reduces the growing number of required parity bits. Thus, when *N* and *K* are larger, the saved area or the cost for the proposed method will be larger. As Table 4 shows, we compared the simulation results of the proposed method by varying *N* and *K*. When *N*, *K* = 16, the saved parity bits are 983,040 bits. The number of saved parity bits increase dramatically, compared with *N* = *K* = 4. A total of 983,040 bits are estimated to cover 79,288 μm^2 in terms of area. The detection fail probability of the proposed method is 0.214%, which is an increase of 18.233%, compared with conventional bit-interleaving.

Our work saved (*K*-1)/*K* of the parity bits, but the probability for error detection fails increased by 11.4833% - 18.232%, compared to conventional bit-interleaving. To improve detection probability, we can use more redundant parity bits in contrast with the case of Table 4. We realized that using twice as many parity

Table 4. Effects of the proposed method

N-way, *K* dies (where *N*=*K*)	Saved parity bits (bit)	Saved area (μm²)	Increased detection fails probability (%)
4	49,152	3,964.4037	11.4833
8	229,376	18,500.5507	15.7205
16	983,040	79,288.0742	18.232

Table 5. Effects of the proposed method with enhanced detection probability

N-way, *K* dies (where *N*=*K*)	Saved parity bits (bit)	Difference of detection probability (%)	Increased detection fails probability (%)
4	32,768	0.001	0.5025
8	196,608	0.005	2.3697
16	917,504	0.004	2.3121

bits enhanced detection probability compared with the original proposed method for separately protecting odd dies and even dies. Adjacent dies are protected by each other's parity bits. Table 5 summarizes our simulation results. The simulation results represent the effects of the proposed method in enhancing the detection probability compared with conventional bit-interleaving. The saved parity bits and saved area of the proposed method with enhanced detection probability should be reduced, compared with the original proposed method. However, the probability for detection fails is almost the same as for conventional bit-interleaving. This results indicate that the proposed technique provides an efficient solution for burst errors with a reduction in redundant memory bits, whereas the detection capability of multiple bit upsets can be retained, compared to the conventional *N*-way interleaving structures.

In this paper, we have focused on a 3D arrangement of the message word and redundant bits to protect in a stacked SRAM device. We mainly considered constructing the efficient memory word-parity structure that arranges individual bits and their redundant parity bits. Basically, the redundant bit generation and checking in this paper is based on a simple parity generation. This can be replaced and expanded by any well-known protection techniques such as Hamming, Shao code, RS code, more complex EDAC codes for the purpose and requirements in the target memory system.

However, SRAM-based cache structure (ex. L1 cache, fast executable code segments copied from FLASH memory) usually requires 1 clock cycle in the latency.

Since the original copy of the content is also located in the lower memory hierarchy, it is not necessary to have an error correction feature for such SRAM devices.

CONCLUSION

3D integration has emerged in recent years to overcome the physical limitation of sub-10 nanometers processing. As we scale down to the nanometer regime, reliability will emerge as a first-class design constraint. The design of modern microprocessors and system-on-a-chips will demand large and reliable embedded cache memory.

By applying 2D bit-interleaving in 3D bit-partitioned SRAM, we greatly reduced the number of parity bits. Only $1/K$ parity bits are needed with a 0.024-0.036% detection probability loss, compared with conventional bit-interleaving. By reducing the number of parity bits with negligible errors for detection fails, our work shows significant savings in cost. Furthermore, by using more redundant parity bits, we enhanced the detection probability.

In future works, beyond the burst error detection, we will consider and test several error correction techniques to 3D SRAM designs.

ACKNOWLEDGMENT

This work was supported by IDEC (Integrated circuit Design Education Center).

REFERENCES

[1] M.-Ch. Tsai, T.-C. Wang, and T. T. Hwang, “Through-Silicon Via Planning 3-D Floorplanning,” IEEE Trans. on VLSI, Vol. 19, No. 8, pp.1448-1457, 2011.

[2] R.C. Baumann, “Soft errors in commercial integrated circuits,” Int’l J. of High Speed Electronics and Systems, Vol. 14, No. 2, pp.299-309, 2004.

[3] S.M. Abbas, S. Lee, S. Baeg, S. Park, “An Efficient Multiple Cell Upsets Tolerant Content-Addressable Memory,” IEEE Trans. on Computers, Vol. 63, No. 8, pp.2094-2098, 2014.

[4] E. Ibe, H. Taniguchi, Y. Yahagi, K. Shimbo, T. Toba, “Impact of Scaling on Neutron-Induced Soft Error in SRAMs From a 250 nm to a 22 nm Design Rule,” IEEE Trans. on Electron Devices, Vol. 57, No. 7, pp.1527-1538, 2010.

[5] L.-X. Huang, H.-G. Xie, S.-L. Niu, “Monte Carlo Simulation of Scaling Effect on SEI and MBU Cross Sections by High Energy Protons,” Int’l Workshop on Monte Carlo Codes & MCNEG 2007 meeting, 2007.

[6] L.-J. Chang, Y.-J. Huang, and J.-F. Li, “Area and reliability efficient ECC scheme for 3D RAMs," IEEE Int’l Symp. on VLSI-DAT, pp.1-4, 2012.

[7] K. Puttaswamy, and G.H. Loh, “3D-integrated SRAM components for high-performance micro-processors,” IEEE Trans. on Computers, Vol. 58, No. 10, pp. 1369-1381, 2009.

[8] P. Reed, G. Yeung, and B. Black. “Design aspects of a microprocessor data cache using 3D die interconnect technology,” IEEE Int’l Conf. on Integrated Circuit Design and Technology, pp. 15-18, 2005.

[9] J. Borkenhagen, and S. Salvatore, “5th Generation 64-bit PowerPC-Compatible Commercial Processor Design," IBM White Paper, 1999.

[10] W. Bryg,and J. Alabado, “The ultrasparc t1 processor-reliability, availability, and serviceability,” Whitepapers: UltraSPARC Processors Documen-tation, 2005.

[11] M.-Y. Hsiao, “A Class of Optimal Minimum Odd-weight-column SEC-DED codes,” IBM J. of Research and Development, Vol. 14, No. 4, pp.395-401, 1970.

[12] J. Kim, N. Hardavellas, M. Ken, B. Falsafi, and, J.C. Hoe, “Multi-bit error tolerant caches using two-dimensional error coding,” Proc. of 40th IEEE/ACM Int’l Symp. on Microarchitecture, pp.197-209, 2007.

[13] S. Kwon, H.S. Choi, J.K. Park, and J.T. Kim, “Radiation-Induced Soft Error Detection Method for High Speed SRAM Instruction Cache,” J-KICS, Vol. 35, No. 6, pp.948-953, 2010.

[14] J.K. Park and J.T. Kim, “An Evolutionary Approach to the Soft Error Mitigation Technique for Cell-Based Design,” Advances in Electrical and Computer Eng., Vol. 15, No. 1, pp.33-40, 2015.

Heung Sun Yoon received a BS degree in Electric and Electronics Engineering in 2010 and received a MS degree in department of Electrical and Computer Engineering in 2015 from Sungkyunkwan University, Korea. From 2015, he joined SanDisk as an engineer. His current research interests include soft error analysis and tolerance techniques, design methodology, and embedded systems.

Jong Kang Park received BS and MS degrees in Electric, Electronics and Computer Engineering in 2001, 2003 and Ph.D. degree in Electric and Electronics Engineering from Sungkyunkwan University, Korea in 2008. From 2008 to 2013, he was with Samsung Electronics where he designed touch sensor ICs. He is now a research professor, School of Electronic and Electrical Engineering, Sungkyunkwan University. His current research interests include the sensor data acquisition, embedded system and soft error analysis and tolerance techniques for VLSI designs.

Jong Tae Kim is a Professor at the School of Electronic and Electrical Engineering, Sungkyunkwan University, where he has been since 1995. He received the BS degree in electronics engineering from Sungkyunkwan University in Korea in 1982 and the MS and PhD degrees in electrical and computer engineering at the University of California, Irvine, in 1987 and 1992, respectively. From 1991 to 1993 he was with the Aerospace Corporation in Elsegundo, California. He was a full-time lecturer at Chunbuk National University in Korea from 1993 to 1995. His research interests include SoC design and design methodology, embedded systems, and multi-core processor architecture.