IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received April 30, 2019, accepted May 13, 2019, date of publication May 17, 2019, date of current version May 31, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2917195

A Novel Built-In Self-Repair Scheme for 3D Memory

TIANMING NI[1,2], HAO CHANG[3], YAO YAO[4], XUEYUN LI[4], AND ZHENGFENG HUANG[4]

[1]College of Electrical Engineering, Anhui Polytechnic University, Wuhu 241000, China
[2]Key Laboratory of Advanced Perception and Intelligent Control of High-end Equipment, Ministry of Education, Wuhu 241000, China
[3]Department of Computer Science and Technology, Anhui University of Finance and Economics, Bengbu 233030, China
[4]School of Electronic Science & Applied Physics, Hefei University of Technology, Hefei 230009, China

Corresponding author: Zhengfeng Huang (huangzhengfeng@139.com)

This paper was supported in part by the Anhui Polytechnic University Research Startup Foundation under Grant 2018YQQ007, in part by the National Nature Science Foundation of China under Grant 61574052, Grant 61704001, Grant 61874156, in part by the Anhui Provincial Natural Science Foundation under Grant 1808085QF196 and Grant 1908085QF272, and in part by the Key Projects of Natural Science Research of Universities in Anhui Province under Grant KJ2016A001.

ABSTRACT Three-dimensional (3D) memory products based on through silicon via (TSV) are widely developed to fulfill the ever-increasing demands of per unit area storage capacity. The yield is still one of the critical challenges for 3D memory. Redundancy technique is now widely used in industry to improve yield. How to reduce the overhead of redundancy by improving the utilization of redundancy is important to 3D memory. In this paper, we propose a row/column block-based mapping technique for 3D memory built-in self-repair scheme to improve the utilization of redundancy and low hardware overhead. Each row/column is divided into row/column block and the mapping can be performed at row/column-block level instead of the original row/column level. Therefore, more faulty cells can be clustered into the same row/column. Based on the proposed technology, a 3D-essential spare pivoting (ESP) algorithm is also proposed for the allocation of redundant rows and columns, and the area overhead of this algorithm is particularly low. The experimental results show that on an average the repair ratio of our proposed scheme is much better than the fault clustering technique by 12% and the redundancy-cost can reduce 23%.

INDEX TERMS 3D memory, built-in self-repair, yield.

I. INTRODUCTION

Three-dimensional (3D) memory has become a promising solution to address the unlimited demands for integration capabilities [1]–[9], [24]–[30]. 3D memory provides increased memory capacity with higher bandwidth, smaller latency, and lower power consumption by integrating multiple DRAM layers with short and dense through-silicon vias (TSVs). In spite of these tremendous advantages, yield is still one of the most critical issues in the 3D memory technology [8]. What's more, the yield will decrease as the number of layers stacked grows [9]. Inevitably, adding redundant row or column to replace the faulty ones plays an important role in yield enhancement.

On the one hand, various Built-In Self-Repair (BISR) and Built-Off Self-Repair (BOSR) [10]–[12] schemes have been proposed to minimize the redundancy analysis (RA) time as well as hardware overhead, and to improve the repair rate. On the other hand, various redundancy sharing strategies [15]–[17] including inter-die and global redundancy sharing methods have been proposed to increase utilization of the redundancy. In [18], a fault clustering technology is proposed which can share the "faulty cells" across the memory dies. It can cluster the faults from different memory layers to the same row/column. The redundancy utilization of this scheme is higher than the schemes mentioned above. However, this technique also has several limitations. Firstly, the fault clustering technique can only be performed while the rows/columns in the mapped layer are completely faultless, which greatly limits the clustering efficiency of the technology. Secondly, combined with the cyclic mapping method, it can achieve the highest repair rate but the hardware overhead is unacceptable.

To address these problems, a row/column block-based mapping technique for 3D memory BISR scheme is proposed in this paper. Using the Divided Word-Line (DWL) and

The associate editor coordinating the review of this manuscript and approving it for publication was Cihun-Siyong Gong.

2169-3536 © 2019 IEEE. Translations and content mining are permitted for academic research only.
Personal use is also permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

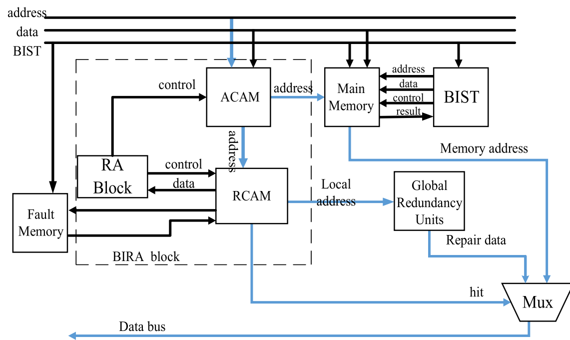


FIGURE 1. Schematic of our global BISR design.

Divided Bit-Line (DBL) techniques [13], [14], each row/ column of the memory array is divided into several blocks, and each layer exchanges mapping information based on blocks. This enables fault clustering more complete, and almost all faults can be clustered to the same row /column. In addition, a 3D-ESP algorithm based on the essential spare pivoting algorithm for redundancy analysis is proposed. Experiments show that the proposed method can effectively improve the yield of 3D memory. Compared with the previous methods, the proposed method can achieve higher repair rate with the same number of redundancy and achieve the same repair rate with the less area overhead.

The remainder of this paper is organized as follows. Section 2 introduces the proposed row/column block-based mapping technique and 3D-ESP (essential spare pivoting) algorithm. Experimental results are shown in section 3. Finally, some conclusions are given in section 4.

II. PROPOSED APPROACH

In this section, we propose a row/column block-based mapping technique. To support this technique, a 3D essential spare pivoting (ESP) algorithm is also introduced, which is designed for allocating the redundant row/column after fault clustering.

A. PROPOSED 3D BISR WITH ROW/COLUMN BLOCK-BASED MAPPING ALGORITHM

The BISR structure proposed in this paper adopts global redundancy scheme as shown in Fig.1. The BIRA module consists of Built-In Self-Test (BIST), Redundancy Analysis (RA) block, Address Content Addressable Memory (ACAM), Redundancy Content Addressable Memory (RCAM), Fault Collection Registers (FCR) and spare memories called Global Redundant Units (GRUs). Generally, the BISR works as follows: firstly, RCAM stores the original location of the fault unit detected from the BIST module. Next, the RA module executes the row/column block mapping algorithm based on the information in RCAM, and modifies the information of RCAM and ACAM. After the mapping algorithm is executed, the RA module executes the 3D-ESP algorithm, which analyses all faults stored in RCAM and allocates redundant resources. Then the faulty information and corresponding repairing information are recorded in FCR.

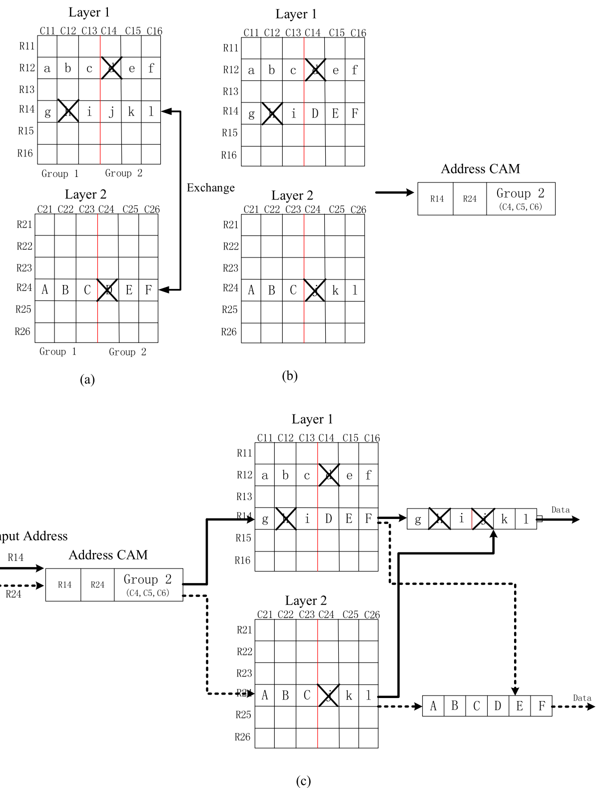


FIGURE 2. The principle of row/column block-based mapping: (a): Original location of faults; (b): Physical location of input data; (c): Memory accessing.

When the input address is decoded, ACAM will access the main memory and RCAM. If the input address is not found in RCAM, the content from the main memory will be the final output. Otherwise, RCAM will send a hit signal and control the multiplexer to select the data in the global redundant unit as the final output. The access process is marked with a blue line in the diagram.

Fig.2 is a principle diagram based on row/column block mapping. Fig.2(a) is the original location of the faults derived from BIST module. Because there are faults (R12, C14) and (R24, C24) in the same column in two layers, we adopt a row-block mapping strategy to divide each row of the array into two row groups. In order to cluster these two faults into the same layer, we only need to modify the mapping relationship of RB1 in Group 2 of R14 and R24 in ACAM, as shown in Fig.2(a). If we store the data "abcdef", "ghijkl" and "ABCDEF" in rows R12, R14 and R24 respectively, the physical location of the fault storage is shown in Fig.2(b). The actual physical location of "DEF" runs to Group 2 of R14, and the actual physical location of "jkl" runs to Group 2 of R24. Similarly, when memory access is required, as shown in Fig.2(c), given R14, since the Group2 of R14 has been mapped to the Group2 of R24, the data accessed is "gXiXkl", and the data accessed by R24 is "ABCDEF".

Although the actual physical location of the faults remains unchanged, logically, the faults in the second layer are mapped to the first layer, so that the faults "d" and "i" are

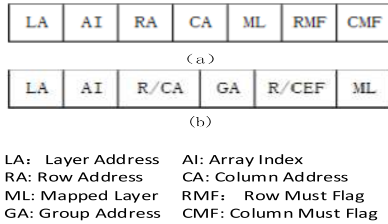


FIGURE 3. CAM structure: (a) Redundancy CAM; (b) Address CAM.

clustered in the same column. Thus, we only need two global redundant units to repair all three faults.

Before introducing the fault clustering algorithm, we still use two Content Addressable Memories (CAMs) structures [18], as shown in Fig.3. In Fig.3(a), RMF = 1 means that the fault can only be repaired by redundant rows, and CMF = 1 means that the fault can only be repaired by redundant columns. RCAM records the information of the fault unit and the mapped layer, indicating the fault marked by LA, AI, RA and CA, which exchanges with the location marked by ML, AI, RA and CA. Compared with [18], the ACAM structure adds a BA, which is used to mark the block number to which the fault belongs. In ACAM, R/CEF = 0 denotes row block mapping, that is, ACAM denotes mapping address exchange between LA, AI, RA, BA-marked row blocks and ML, AI, CA, BA-marked row blocks. Similarly, R/CEF = 1 denotes column mapping, that is, ACAM denotes mapping address exchange between LA, AI, CA, BA-marked column blocks and ML, AI, CA, BA-marked column blocks.

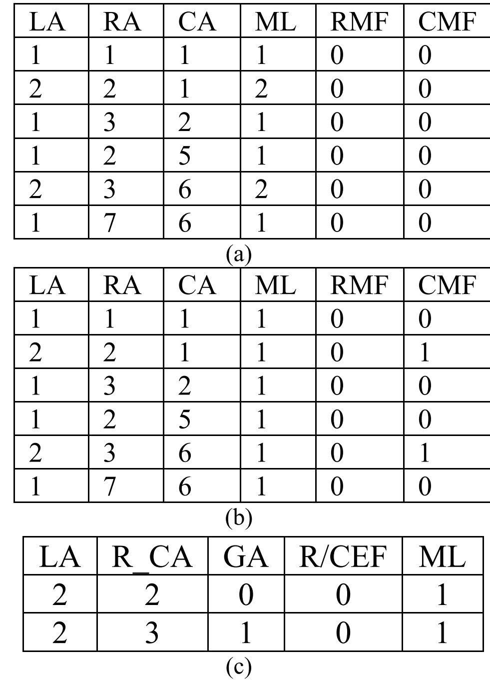
Algorithm 1 Row/Column Block-Based Mapping Technique

Input: RCAM, the number of groups: g, the number of faults: f_num

Output: ACAM

1. **For** each $RCAM_i \in RCAM$
2. **If**$(\exists 0 \leq i < f_num)$ such that $(LA_i == L_m)$ $\&\&(RMF_i != 1)$
3. $BA = CA_i/(n/g)$;
4. **If**$(\exists 0 \leq j < f_num)$ such that $(LA_j == L_n)$ $\&\& (CA_j == CA_i) \&\& (RA_j != RA_i)$
5. **If**$(\exists 0 \leq k < f_num)$ such that $(LA_k == L_n)$ $\&\& (RA_k == RA_i) \&\&(CA_k/(n/g) == BA)$
6. The row cannot mapping;
7. **Else** $\{ML_i = L_n; CMF_i = 1;$
8. Add the entry$\{L_m, RA_i, GA, 0, L_n\}$ into ACAM }
9. **If**$(\exists i < p < f_num)$ such that $(LA_p == L_m)$ $\&\&(RA_p == RA_i) \&\&(CA_p/(n/g) == BA)$
10. $ML_p = L_n$;
11. **If**$(\exists j < q \leq f_num)$ such that $(LA_q == L_n)$ $\&\&(CA_q == CA_i)$
12. $CMF_q = 1$;

The pseudocode of the proposed row/column block-based mapping algorithm is shown in Algorithm 1. Take row-based address exchange as an example. The array size is set to m ∗ n, assuming that the mapping layer and the mapping layer



LA	RA	CA	ML	RMF	CMF
1	1	1	1	0	0
2	2	1	2	0	0
1	3	2	1	0	0
1	2	5	1	0	0
2	3	6	2	0	0
1	7	6	1	0	0

(a)

LA	RA	CA	ML	RMF	CMF
1	1	1	1	0	0
2	2	1	1	0	1
1	3	2	1	0	0
1	2	5	1	0	0
2	3	6	1	0	1
1	7	6	1	0	0

(b)

LA	R_CA	GA	R/CEF	ML
2	2	0	0	1
2	3	1	0	1

(c)

FIGURE 4. The process of fault group clustered mapping. (a). RCAM status before mapping. (b). RCAM status after mapping. (c). ACAM status after mapping.

are Lm and Ln, respectively. Firstly, each item in RCAM is searched to find the fault unit (Ri, Cj) in the mapping layer. Ri and Cj represent columns i and j respectively, and the corresponding block number BA of the fault is calculated (lines 1-3). Then, all the fault units in the mapping layer are searched to find the fault (Rk, Cj) in the same row as the fault. If there is a fault unit in the BA block of Ri row in Ln, the row group mapping cannot be performed (lines 5-6). Otherwise, we map the row block BA of the Ri line in the Lm layer to the Ln layer (lines 7-8). Check if there are other failures in the row groups mapped by the Lm layer, and if so, change the ML flags of other failures mapped to the Ln layer to Ln (lines 9-10). Check if there are other failures in the Cj column in the Ln layer (except Rk, Cj), and if so, set the CMF flags for these failures to 1 (lines 11-12).

We take the location of the fault in Fig.2 as an example to illustrate the execution process of the mapping algorithm based on row/column blocks. Given Lm = 2, Ln = 1. Taking row-based mapping as an example, we divide each row into two row groups, and the fault unit in RCAM is shown in Fig.4(a). We first find the fault of LA = 2 in RCAM, then find the fault in the second line (2,1), which belongs to row Group 1. Then we find the fault of the same column (1,1) in Layer 1 LA = 1, and check whether the row Group 1 of Layer 1 LA = 1 in RCAM has any fault. Because this row groups are not faulty, we exchange the index address of line Group 1 in Layer 1 and Layer 2 in ACAM, so we modify the ML and CMF flags corresponding to the fault (2,1), as shown in Fig4.(b), and add the exchange information to ACAM, as shown in Fig.4(c). Similarly, we continue to

ML	AI	RA	CA	FT

FIGURE 5. Structure of the FCR.

find the corresponding fault (3,6) of mapping layer LA = 2 in RCAM, which belongs to the row Group 2. And there happens to be a fault (7,6) of LA = 1 in RCAM. Since the fault of LA = 1 is not in Group2 of row 3, then the index address of Group 2 in row 3 of the two layers can be exchanged. ML of (3,6) in RCAM can be changed to 1, CMF is set to 1, and the exchanged information can be added to ACAM.

B. BUILT-IN REDUNDANCY ANALYSIS

In order to apply to row/column block mapping technology, this paper proposes an improved Built-In Redundancy Analysis (BIRA) algorithm based on the Essential Spare Pivoting (ESP) algorithm [18], which is called the 3D-ESP algorithm in this paper. In order to implement row/column block mapping technology and 3D global BISR architecture, we propose a global FCR structure. As shown in Fig.5, ML fields are added to traditional FCR. The FCR is located in the BISR layer and assigns the fault addresses remapped by all layers. In these fields, FT field represents the fault type, FT = 0 represents orthogonal single unit fault, FT = 1 represents row fault, FT = 2 represents column fault.

Algorithm 2 shows the pseudocode of the proposed BIRA algorithm which is used to determine whether the fault in the 3D memory can be repaired and how many GRUs are needed. When all faults in 3D memory have been clustered, we identify the types of faults stored in RCAM, and store the information in FCRs. Meanwhile, the FTs stored in the FCRs are also identified by comparing the next entry in redundancy CAM with those stored in the FCRs. According the values of RMF and CMF, we classify three kinds of conduct. If RMF = 1, the fault can only be repaired by the row. We only need to judge whether there is a fault stored in FCRs in the same row. If not, we can add the fault into FCRs and mark it as a row fault. In the same way, if CMF = 1, we only need to judge whether there is a fault stored in FCRs in the same column. If not, we can add the fault into FCRs and mark it as a column fault. If both RMF and CMF are equal 0, we need to judge whether there is a fault stored in FCRs in the same column or same column, if not, we can add the fault into FCRs and mark it as an orthogonal fault. In Algorithm 2, we show the case where both RMF and CMF are equal to 0, the other two cases are similar. The FCRs contains GRU entries, where the GRU denotes the number of global redundant units. If the FCRs do not have sufficient entries to store the fault information for next entry in RCAM, the repair_fail_flag signal is asserted and the faulty memory cannot be repaired successfully.

Algorithm 2 Proposed BIRA Algorithm

Input: (i) RCAM after fault clustering, (ii) Counts the total number of fault of 3D memory: f_num, (iii) Counts the number of Global Redundant Unit: GRU.

Output: (i) FCR, (ii) Counts the number of used GRUs: cont_f

```
1.  Clear FCRs
2.  If(RCAM[0].RMF==1)
3.    add {RCAM[0].ML, RCAM[0].RA, RCAM[0].CA,
    1} into FCR[0]
4.  Else if(RCAM[0].CMF==1)
5.      add {RCAM[0].ML, RCAM[0].RA, RCAM[0].CA,
    2} into FCR[0]
6.    Else
7.      add {RCAM[0].ML, RCAM[0].RA, RCAM[0].CA,
    0} into FCR[0]
8.  int cont_f = 1;
9.  For each RCAM[i](1 < i ≤ f_num)
10. If (RCAM[i].RMF==0) &&(RCAM[i].CMF==0)
11.   If there exist a j (0 =< j < cont_f) satisfy
    The following conditions {
12.
     If((FCR[j].ML==RCAM[i].ML)&&(FCR[j].
        RA==RCAM[i].RA&&FCR[j].FT==1)||
     (FCR[j].CA==RCAM[i].CA&&FCR[j].FT==2))
13.           Discard the faulty information;
14.         Else if((FCR[j].ML==RCAM[i].ML)
          &&( FCR[j].RA==RCAM[i].RA)&&(FCR[j].
          FT==0))
15.             FCR[j].FT = 1;
16.          Else if((FCR[j].ML==RCAM[i].ML)
          &&(FCR[j].CA==RCAM[i].CA)&&(FCR[j].
          FT==0))
17.                 FCR[j].FT = 2; }
18.     Else if(cont_f < GRU)
19.         FCR[cont_f].LA = RCAM[i].ML;
20.         FCR[cont_f].RA = RCAM [i].RA;
21.         FCR [cont_f].CA = RCAM [i].CA;
22.         FCR[cont_f].FT = 0; cont_f++;
23.         Else{repair_fail_flag = 1;}
24. If(repair_fail_flag = 1) return 0;
25. Elsereturn cont_f;
```

III. EXPERIMENTAL RESULTS

A. EXPERIMENT SETUP

The repair rate refers to the ratio of the number of repairable 3D memories to the total number of 3D memories. We use repair rate as a criterion to judge the repair ability of different repair strategies. The higher the repair rate is, the higher the yield is. Note that the repair rate relates to failure rate as well as redundant resources.

In our simulation, different number of faults are injected into each memory block at random locations using the Polya-Eggenberger distribution [15], [20], [21]. When faults are injected, the probabilities of faulty rows, faulty column and orthogonal single faults are set to be 15%, 15%, and 70%, respectively. After fault injection, fault clustering technique is performed using the method in [18] and the method proposed

TABLE 1. Repair rate comparisons of different 3D memory redundancy schemes.

Memory Size	number of Redundancy per layer	repair rate under different layers/%														
		4-layer					6-layer					8-layer				
				Proposed					Proposed					Proposed		
		[16]	[18]	a=2 b=2	a=4 b=4	a=8 b=8	[16]	[18]	a=2 b=2	a=4 b=4	a=8 b=8	[16]	[18]	a=2 b=2	a=4 b=4	a=8 b=8
256*256	3	16.2	32	37.1	40.3	40.8	10.5	27.1	32.9	37.8	38.4	6.6	24	32.5	37.5	40.4
	4	37.6	61.2	66.6	69.4	69.6	34.6	60.6	69.1	73.3	73.5	31.9	62.4	70.1	74.1	76.2
	5	63.5	81.3	84.9	85.7	88.8	62.9	83.5	88.9	90.2	91.6	66	87.8	91.6	91.9	94.3
	6	80.7	93.5	94.8	95.7	95.9	83.2	95.4	96.7	97.6	97.8	85.4	96.7	98	98.3	98.7
	7	92	97.9	98.5	98.8	99	94.2	98.4	99.3	99.4	99.5	96.1	99.4	99.6	99.7	99.8
512*512	8	39.8	64.5	71	77.5	79.2	38.2	62.7	75.7	76	81.2	31.9	65	76.2	82.3	83.3
	9	53.5	76.8	81.8	86.3	87.2	52.4	78.6	86.8	87.8	90.8	50.2	81.3	87.9	92.4	92.9
	10	66.7	84.6	89.8	91.4	91.7	67.4	87.2	93.5	93.6	94.5	67.6	90.6	94.1	96.1	96.9
	11	76.5	91.4	94.5	95	95.7	78.9	93.6	96.4	97.4	98.1	80.7	96	97.5	98.1	99
	12	83	95.7	97.1	97	97.5	86.6	97.2	98.1	99	99.5	89.6	98.4	98.9	99.4	99.6
1024*1024	19	63.7	85.7	90.2	91.3	91.6	63.1	88.2	93.3	94.5	95.3	62.4	90	93.6	96.7	96.9
	20	69.7	88.5	92.8	93.4	93.5	70.5	92.5	94.2	95.6	96.2	72	93.6	95.6	98	98.6
	21	75.8	91.2	94.5	95.2	95.1	77	94	96.5	96.9	97.5	79	96.1	97.5	98.4	98.6
	22	79.8	93	96.2	96.5	96.2	82.9	96	97.3	97.9	98.5	84.8	97.7	97.9	98.1	98.8
	23	84.3	94.3	97.3	97.7	97.4	86.2	97.5	98.5	98.8	99.1	88.4	98.5	98.6	99.4	99.5
2048*2048	36	60.1	81.6	87.8	88.6	88.7	60	85.8	91.9	93.2	94.5	58.8	85.6	95.2	95.4	95.5
	38	65.6	85.6	91.3	91.5	91.2	68	89.4	94.2	95.5	96.6	65.8	90.6	98	98.4	98.8
	40	72	88.3	94.4	93.5	93.4	74.4	93.4	96	97.5	97.9	73.2	93.8	98.4	98.5	98.7
	42	76.8	91.3	95.9	94.7	95.7	80.8	95.6	97.1	98.4	98.5	81	96	98.6	98.7	99
	44	81.4	93.9	97.1	96.4	97.1	86.1	97	98.2	99.1	99.4	86.4	97.4	98.9	99	99.6

in this paper, and 3D-ESP algorithm proposed in this paper is used to carry out redundant analysis and repair. In [18], three mapping schemes are proposed: pairwise mapping, multi-to-one mapping, and cyclic mapping. The efficiency of pairwise mapping is low since it can only one to one mapping. And the overhead of cyclic mapping is so large that can't be afforded. Therefore, the experiment in this paper adopts a multi-to-one mapping scheme.

In addition, the size and number of layers of 3D memory arrays also affect the experimental results. In order to evaluate the proposed strategy, the fault repair results from global redundancy sharing scheme [16], fault clustering scheme [18] and the proposed scheme are simulated. The simulation results are shown in Table 1. The failure rates are set to 2.5%, a and b represent the number of row blocks and column blocks respectively. In each case, 1000 groups of experiments are carried out, and the average values are filled in Table 1.

B. RESULTS AND ANALYSIS

From Table I, it can be seen that the repair rate of the proposed scheme is greater than that of the global redundancy sharing scheme [16] and the fault clustering scheme [18] in any case, which shows that the repairability of the proposed scheme is better than that of the previous two schemes. When the memory sizes and layers are the same, the more redundant resources each layer has, the smaller the repair rate difference between the two schemes is. This shows that when the number of redundant resources is relatively small, this method can show more obvious advantages, and the more rows/columns are divided, the higher the repair rate is. However, as the number of rows/columns increases, the repair rate also shows a downward trend. In addition, with the increase of stacking layers, compared with the previous two schemes, the repair rate of the proposed scheme increases much more.

The 3D memory using the proposed technique with a different number of blocks with a fixed error rate set to 2.5% is shown in Fig.6. As the number of redundancy increases, the repair rate of all proposed techniques with different blocks obtains higher repair rate. As shown in Fig.5 the proposed repair technique under different block constraints obtains the repair raise in different gradients. On average, the fault clustering technique with two blocks, four blocks, eight blocks, sixteen blocks can approximately improve the repair rate by 8%, 11%, 12%, 13% compared to the original fault clustering technique. The more the number of blocks, the higher the repair rate, but the costs have increased accordingly. The other experiments each row and each column are divided into four blocks in row/column block-based mapping technique, considering the cost and effectiveness.

Fig.7 shows the repair rate comparison by varying the number of redundancies with a fixed error rate set to 2.5%, and the three schemes are compared: the original, fault

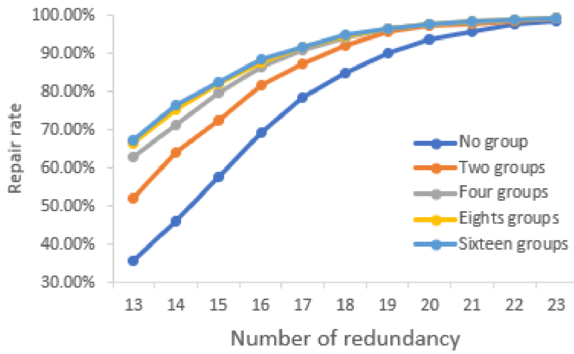


FIGURE 6. Repair ratio of row/column block-based mapping with a different number of block varying the redundancy.

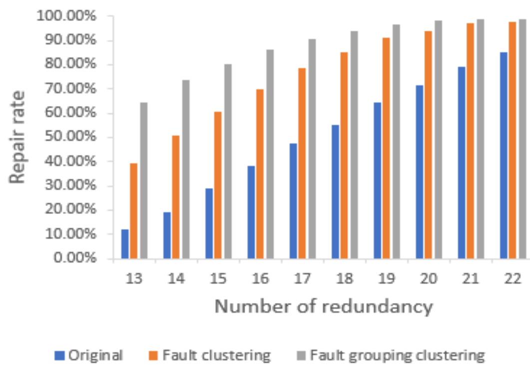


FIGURE 7. Simulation results on repair ratio varying the number of redundancy.

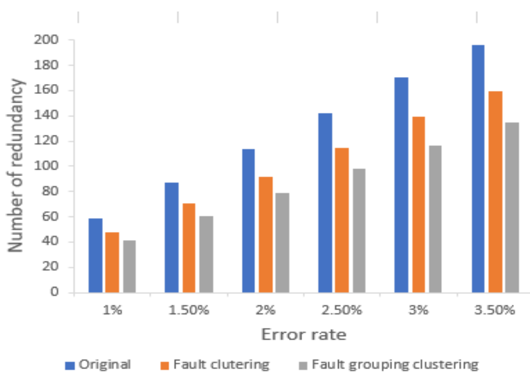


FIGURE 8. Simulation results on the number of redundancy varying error rate.

clustering and row/column block-based mapping technique. Obviously, the row/column block-based mapping technique outperforms the others. In the lowest number of redundancy, it can approximately improve the repair rate by 53%, 25% compared to original scheme and fault clustering technique, respectively. On average, the technique can approximately improve the repair rate by 38%,12% compared to the original scheme and fault clustering technique, respectively. As the number of redundancy increases, the improvement of row/column block-based mapping technique than fault clustering technique on repair rate decreases. Obviously, row/column block-based mapping technique can obtain the optimal repair rate with the same number of redundant units.

Fig.8 shows the number of redundancy comparison by varying the error rate with a target 99% yield. With the ramping error rate, the required redundancy of original scheme rises sharply while the row/column block-based mapping technique rises the most gently. In three schemes, the proposed technique can efficiently save the requirement of redundancy. As the error rate increases, the proposed scheme can save more redundancies compared to the original scheme. With the highest error rate, the proposed scheme 23% redundancy-cost reduction over the clustering scheme.

C. COST ANALYSIS

The increment of area overhead refers to the ratio of hardware overhead to total overhead of the proposed scheme compared with reference [18]. There are two main sources of hardware overhead in this paper. One is the area overhead caused by the modified ACAM structure and FCR structure; the other is the area overhead caused by the use of word line segmentation and bit line segmentation. Assume each row/column is divided into 8 blocks, and ACAM adds 3 bits per unit (BA filed), FCR adds 3 bits per unit (ML field), assuming average 15 FCR units, then the total additional hardware overhead of the FCR is: 8*64*15*3 = 23040bit. The two structures increase 32256 bits in total, and the area cost is less than one thousandth, so the area of this item is almost negligible.

In addition, word segmentation uses one AND gate for each row group, bit segmentation uses two PMOS for each column group. These gate devices use Nangate 45nm technology library, and the NAND gate area is 0.798 μm^2, the INV gate area is 0.532 μm2, and the NMOS area is 0.266 μm2. Therefore, if each row and column is divided into two parts, the area overhead of each chip layer will be increased by 488112.128 μm^2 (1024*64* [4* (0.798 + 0.532)+4*2*0.266] = 488112.128 μm^2). According to [18], the density of DRAM chip is about 27.9Mb/mm^2. For each layer of 64Mbit chip in the experiment, it occupies an area of 2293906.81μm^2, and the area cost of using word line and bit line segmentation technology accounts for about 10.64%. Similarly, the hardware overhead of 4 row/column blocks and 8 row/column blocks is 21.31% and 42.56% respectively.

IV. CONCLUSION

In this paper, the row/column block-based mapping technique is proposed for 3D memory BISR. The mapping can be performed at row/column blocks level between different layers instead of the traditional row/column level. Experimental results show that the proposed technique achieves 12% and 38% higher repair rate compared to the fault clustering scheme and original scheme [16] in multi- to-one mapping scheme. In addition, the redundancy-cost reduction of our proposed scheme is much better than [16] by 23% with the same number of redundancy. The 3D-ESP algorithm suitable for row/column block-based mapping technique is proposed in this paper to perform redundancy analysis. The area overhead of this algorithm is particularly low. The technique

solves the problem of large power overhead, high access delays and low yield with 21.312% increased area overhead.

REFERENCES

[1] T. Ni, H. Liang, M. Nie, X. Xu, A. Yan, and Z. Huang, "A region-based through-silicon via repair method for clustered faults," *IEICE Trans. Electron.*, vol. E100-C, no. 12, pp. 1108–1117, Dec. 2017.

[2] T. Ni *et al.*, "Vernier ring based pre-bond through silicon vias test in 3D ICs," *IEICE Electron. Express*, vol. 14, no. 18, p. 11, Jul. 2017.

[3] T. Ni, H. Chang, X. Zhang, H. Xiao, and Z. Huang, "Research on physical unclonable functions circuit based on three dimensional integrated circuit," *IEICE Electron. Express*, vol. 15, no. 23, p. 10, Dec. 2018.

[4] T. Ni, H. Chang, H. Qi, and Z. Huang, "A novel in-field TSV repair method for latent faults," *IEICE Electron. Express*, vol. 15, no. 23, p. 10, Dec. 2018.

[5] T. Ni, H. Chang, X. Sun, X. Xia, and Z. Huang, "An enhanced time-to-digital conversion solution for pre-bond TSV dual faults testing," *IEICE Electron. Express*, vol. 16, no. 3, p. 10, Feb. 2019.

[6] T. Sekiguchi, K. Ono, A. Kotabe, and Y. Yanagawa, "1-Tbyte/s 1-Gbit DRAM architecture using 3-D interconnect for high-throughput computing," *IEEE J. Solid-State Circuits*, vol. 46, no. 4, pp. 828–837, Apr. 2011.

[7] J.-S. Kim *et al.*, "A 1.2 V 12.8 GB/s 2 Gb mobile wide-I/O DRAM with 4×128 I/Os using TSV based stacking," *IEEE J. Solid-State Circuits*, vol. 47, no. 1, pp. 107–116, Jan. 2012.

[8] M.-F. Chang, P.-F. Chiu, W.-C. Wu, C.-H. Chuang, and S.-S. Sheu, "Challenges and trends in low-power 3D die-stacked IC designs using RAM, memristor logic, and resistive memory (ReRAM)," in *Proc. IEEE 9th Int. Conf. ASIC (ASICON)*, Xiamen, China, Oct. 2011, pp. 299–302.

[9] H.-H. S. Lee and K. Chakrabarty, "Test challenges for 3D integrated circuits," *IEEE Design Test*, vol. 26, no. 5, pp. 26–35, Sep. 2009.

[10] H.-H. Liu *et al.*, "A built-off self-repair scheme for channel-based 3D memories," *IEEE Trans. Comput.*, vol. 66, no. 8, pp. 1293–1301, Aug. 2017.

[11] S.-K. Lu, C.-L. Yang, Y.-C. Hsiao, and C.-W. Wu, "Efficient BISR techniques for embedded memories considering cluster faults," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 18, no. 2, pp. 184–193, Feb. 2010.

[12] J. Kim, W. Lee, K. Cho, and S. Kang, "Hardware-efficient built-in redundancy analysis for memory with various spares," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 3, pp. 844–856, Mar. 2017.

[13] M. Yoshimoto *et al.*, "A divided word-line structure in the static RAM and its application to a 64 K full CMOS RAM," *IEEE J. Solid-State Circuits*, vol. SSC-18, no. 5, pp. 479–485, Oct. 1983.

[14] A. Karandidar and K. K. Parhi, "Low power SRAM design using hierarchical divided bit-line approach," in *Proc. Int. Conf. Comput. Design*, Oct. 1998, pp. 82–88.

[15] L. Jiang, R. Ye, and Q. Xu, "Yield enhancement for 3D-stacked memory by redundancy sharing across dies," in *Proc. Int. Conf. Comput.-Aided Design*, San Jose, CA, USA, Nov. 2010, pp. 230–234.

[16] X. Wang, D. Vasudevan, and H.-H. S. Lee, "Global built-in self-repair for 3D memories with redundancy sharing and parallel testing," in *Proc. IEEE Int. 3D Syst. Integr. Conf. (3DIC)*, Osaka, Japan, Jan. 2011, pp. 1–8.

[17] J. Lee, K. Park, and S. Kang, "Yield enhancement techniques for 3D memories by redundancy sharing among all layers," *ETRI J.*, vol. 34, no. 3, pp. 388–398, Jun. 2012.

[18] T. Li, Y. Han, X. Liang, H.-H. S. Lee, and L. Jiang, "Fault clustering technique for 3D memory BISR," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, Lausanne, Switzerland, Mar. 2017, pp. 560–565.

[19] C.-T. Huang, C.-F. Wu, J.-F. Li, and C.-W. Wu, "Built-in redundancy analysis for memory yield improvement," *IEEE Trans. Rel.*, vol. 52, no. 4, pp. 386–399, Dec. 2003.

[20] C. H. Stapper, "On a composite model to the IC yield problem," *IEEE J. Solid-State Circuits*, vol. 10, no. 6, pp. 537–539, Dec. 1975.

[21] R.-F. Huang, J.-F. Li, J.-C. Yeh, and C.-W. Wu, "A simulator for evaluating redundancy analysis algorithms of repairable embedded memories," in *Proc. Int. Workshop Memory Tech., Design, Test. (MTDT)*, Isle Bendor, France, Jul. 2002, pp. 68–73.

[22] G. H. Loh, "3D-stacked memory architectures for multi-core processors," in *Proc. Int. Symp. Comput. Archit.*, Beijing, China, 2008, pp. 453–464.

[23] *PTM. 45 nm Predictive Technology Model*. [Online]. Available: http://ptm.asu.edu

[24] L. Jiang, F. Ye, Q. Xu, K. Chakrabarty, and B. Eklow, "On effective and efficient in-field TSV repair for stacked 3D ICs," in *Proc. IEEE Design Autom. Conf. (DAC)*, Austin, TX, USA, 2013, pp. 1–6.

[25] W.-H. Lo, K. Chi, and T. Hwang, "Architecture of ring-based redundant TSV for clustered faults," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, Grenoble, France, Mar. 2015, pp. 3437–3449.

[26] Y.-H. Chen, C.-P. Chiu, R. Barnes, and T. Hwang, "Architectural evaluations on TSV redundancy for reliability enhancement," in *Proc. IEEE Design, Automat. Test Eur. Conf. Exhib. (DATE)*, Lausanne, Switzerland, Mar. 2017, pp. 566–571.

[27] S. Wang, T. Kim, Z. Sun, S. X.-D. Tan, and M. B. Tahoori, "Recovery-aware proactive TSV repair for electromigration lifetime enhancement in 3-D ICs," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 26, no. 3, pp. 531–543, Mar. 2018.

[28] S. Wang, K. Chakrabarty, and M. B. Tahoori, "Defect clustering-aware spare-TSV allocation in 3D ICs for yield enhancement," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, to be published.

[29] C. Hao and L. Huaguo, "Pulse shrinkage based pre-bond through silicon vias test in 3D IC," in *Proc. IEEE VLSI Test Symp. (VTS)*, Napa, CA, USA, Apr. 2015, pp. 1–6.

[30] S. Deutsch and K. Chakrabarty, "Contactless pre-bond TSV test and diagnosis using ring oscillators and multiple voltage levels," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 33, no. 5, pp. 774–785, May 2014.

TIANMING NI was born in 1991. He received the B.S. degree in integrated circuit design and integrated system from the Tianjin University of Technology, Tianjin, China, in 2013, and the Ph.D. degree in integrated circuit and system from the Hefei University of Technology, Hefei, China, in 2018. He joined the College of Electrical Engineering, Anhui Polytechnic University, in 2018. He has been a member of the Key Laboratory of Advanced Perception and Intelligent Control of High-end Equipment, Ministry of Education, China, since 2019. His research interests include 3D IC testing and fault tolerance. He served as a publication Chair on the organizing committee of the 27th IEEE Asian Test Symposium, in 2018.

HAO CHANG received the B.S. degree in computer science and technology from the Anhui University of Finance & Economics, Bengbu, China, in 2004, and the M.S. and Ph.D. degrees in computer application technology from the Hefei University of Technology, Hefei, China, in 2007 and 2015, respectively. He is currently an Associate Professor with the Department of Computer Science and Technology, Anhui University of Finance and Economics. His recent research interests include 3D ICs integration and test, built-in self-test, and fault tolerance.

YAO YAO received the B.S. degree in computer science and technology from the Anhui University of Science and Technology, Huainan, China, in 2016. She is currently pursuing the M.Sc. degree with the Department of Electronic Science & Applied Physics, Hefei University of Technology, Hefei, China. Her current research interests include fault-tolerant and reliability design of 3D ICs.

XUEYUN LI received the B.S. degree in integrated circuit design and integrated system from the Hefei University of Technology, in 2018. She is currently pursuing the master's degree with the School of Electronic Science & Applied Physics, Hefei University of Technology. Her research interests include fault-tolerant and approximate computing.

ZHENGFENG HUANG received the Ph.D. degree in computer engineering from the Hefei University of Technology, in 2009. He joined the Hefei University of Technology, as an Assistant Professor, in 2004, and has been an Associate Professor, since 2010. He was a Visiting Scholar with the University of Paderborn, Germany, from 2014 to 2015. He has been a Professor with the Hefei University of Technology, since 2018. His current research interest includes design for soft error tolerance/mitigation. He is a member of the Technical Committee on Fault Tolerant Computing which belongs to the China Computer Federation. He served on the organizing committee of the IEEE European Test Symposium, in 2014.