논문 01-01-04

A New Test Algorithm for Bit-Line Sensitive Faults in High-Density Memories

고집적 메모리에서 BLSFs(Bit-Line Sensitive Faults)를 위한 새로운 테스트 알고리즘

Dong-Chual Kang*, Sang-Bock Cho*

(姜 東 哲*, 趙 相 福*)

Abstract

As the density of memories increases, unwanted interference between cells and coupling noise between bit-lines are increased. And testing high-density memories for a high degree of fault coverage can require either a relatively large number of test vectors or a significant amount of additional test circuitry. So far, conventional test algorithms have focused on faults between neighborhood cells, not neighborhood bit-lines. In this paper, a new test algorithm for neighborhood bit-line sensitive faults (NBLSFs) based on the NPSFs(Neighborhood Pattern Sensitive Faults) is proposed. And the proposed algorithm does not require any additional circuit. Instead of the conventional five-cell or nine-cell physical neighborhood layouts to test memory cells, a three-cell layout which is minimum size for NBLSFs detection is used. Furthermore, to consider faults by maximum coupling noise by neighborhood bit-lines, we added refresh operation after write operation in the test procedure(i.e., write→ refresh→ read). Also, we show that the proposed algorithm can detect stuck-at faults, transition faults, coupling faults, conventional pattern sensitive faults, and neighborhood bit-line sensitive faults.

요 약

메모리의 집적도가 올라갈수록 원치 않는 셀간의 간섭과 동시에 bit-line간의 상호 노이즈도 증가하게 된다. 그리고 높은 고장 검출율을 요구하는 고집적 메모리의 테스트는 많은 테스트 벡터를 요구하게 되거나 비교적 큰 추가 테스트 회로를 요구하게 된다. 지금까지 기존의 테스트 알고리즘은 이웃 bit-line의 간섭이 아니라 이웃 셀에 중점을 두었다. 본 논문에서는 NPSFs(Neighborhood Pattern Sensitive Faults)를 기본으로 한 NBLSFs(Neighborhood Bit-Line Sensitive Faults)를 위한 새로운 테스터 알고리즘을 제안한다. 그리고 제안된 알고리즘은 부가 회로를 요구하지 않는다. 메모리 테스트를 위해 기존의 5개의 셀 레이아웃이나 9개의 셀 레이아웃을 사용하지 않고 NBLSF 검출에 최소한 크기인 3개의 셀 레이아웃을 이용하였다. 더구나 이웃 bit-line에 의한 최대의 상호잡음을 고려하기 위해 테스트 동작에 refresh 동작을 추가하였다(예 write→ refresh→ read). 또한 고착고장, 천이고장, 결합고장, 기존의 pattern sensitive 고장, 그리고 이웃 bit-line sensitive 고장 등도 검출될 수 있음을 보여준다.

Keywords: Bit-Line Sensitive Faults(BLSFs), memory testing, Coupling Faults(CFs), Pattern Sensitive Faults(PSFs)

* 蔚山大學校 電子工學科 (Dept. of Electronic Eng., Univ. of Ulsan)

接受日: 2001年 2月16日, 修正完了日: 2001년 7月 4日

I. Introduction

The pattern-sensitive faults(PSFs) may be considered as the most general case of k-coupling faults, for k=n, where n represents all cells in the memory. A PSF can be defined as the susceptibility of the contents of a cell to be influenced by the contents of all other cells in the memory [1]-[3]. A group of cells that influences the base cell's behavior is called the neighborhood of the base cell. The PSFs are primarily caused by the high component densities of RAMs and unwanted interference(e.g., electromagnetic) between closely packed lines and signals. A variety of physical fault mechanisms are responsible for PSFs, making it difficult to define the logical fault models to represent them. The RAM testing for unrestricted PSFs would be very costly and impractical since it requires a test length of $(3N^2 + 2N)2^N$ [3]. As a result, the PSF models assume that interaction can occur only between the cells within certain physical proximity. Therefore, these are called the physical neighborhood pattern-sensitive faults (NPSFs). PSF' models are classified into ANPSFs(Active Neighborhood PSFs), PNPSFs(Passive Neighborhood PSFs), and SNPSFs(Static Neighborhood PSFs) [4] and [8]. If only interaction between cells is considered, Type-2 has better fault coverage than Type-1[5]. By the fact that the leakage is maximum when the symmetrically located cells contain the same bit patterns, a variant of 4-cell neighborhood can cover Type-2 [6].

As the density of RAMs increases, the distance between neighborhood bit-lines gets close. But those algorithms mentioned above mainly focused on interaction between cells. And they partially could consider coupling noise by C_g, C_{BB}, $C_{Pull\text{-}Down}$, and C_{BW} shown in Fig. 1. C_{BB} among them is one of the most important parasitic capacitors which affect cells at the right time of read, write, and refresh operations [7].

In most cases, faults by C_{BB} can be masked by conventional fault models. Therefore, it is difficult for test algorithms using normal read/write operations of a memory to consider faults by maximum coupling noise.

That is why we propose a new algorithm.

The more neighborhood cells are considered, the more fault coverage is improved. However all NBLSFs are not likely to be detected because conventional algorithms are based only on the interaction between cells, not bit-lines. From that point of view, the five-cell and nine-cell physical neighborhoods which are typical tiling methods are complicated to detect PSFs in the memory. And they are not appropriate for testing bit-lines coupling noise.

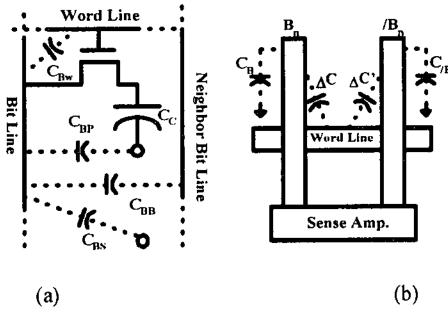
In this paper, a three-cell physical neighborhood to detect both NPSFs and NBLSFs(Neighborhood Bit-Line Sensitive Fault) is used, and refresh operations are added to the proposed test algorithm to consider maximum coupling noise between neighborhood bit-lines. We also show that coupling faults(e.g., CF_{id} and CF_{in}) can be detected within nine physical neighborhood cells [8].

II. Neighborhood Bit-Line Sensitive Faults

Fig. 1(a) shows representative coupling capacitors [7],[9]. These capacitors appear between neighbor elements as parasitic capacitance to cause noise. Even though they cannot change the data of a cell, they can delay the write/read operations of memory cells. If a word-line is fed with high voltage (5V~7V), bit-lines across the word-line raise the voltage level to 100mV~250mV by C_{BW}(coupling cap. between bit-lines and word-lines). This unexpected voltage can be mostly eliminated by the folded bit-lines as shown in Fig. 1(b) that is nowadays mostly used in high-density memories. But there are still noise factors coped with the difference between coupling capacitance of B_n and $/B_n$.

The coupling capacitance of bit-lines connected to cells is expressed by C_g(gate cap.) + C_{BW}. And the capacitance is larger than that of bit-lines which is not connected to cells. As a memory is highly densified, so the crosstalk between bit-lines also sharply increases by C_{BB}(Bit line - Bit line cap.) during refresh operations, as well as read/write operations. The crosstalk happens during charge sharing and sense amplification. The

crosstalk reaches the maximum when the values of bit-lines are complement. On the other hand, it does the minimum when those of neighborhood bit-lines are same.

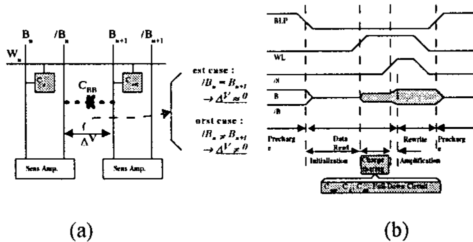


(a) (b)

Fig. 1. Parasitic capacitors and structure of the folded bit-line.
그림 1. 기생 커패시터와 folded bit-line의 구조.

The way to get rid of the crosstalk is to twist bit-lines such as coaxial cables. By that way, the crosstalk between bit-lines mostly disappears. But at the right time of read/write/refresh operations, the crosstalk still appears. In worst case, the data of a cell can maybe change. As a result, C_{BB} becomes one of the most important factors to be able to delay the response for valid memory operations.

Fig. 2(a) and (b) shows C_{BB} between bit lines and the timing diagram for normal operations. Charge sharing are affected by C_g, C_{BB}, $C_{Pull\text{-}Down}$, and C_{BW}, and amplification by C_B, C_{BB}, and C_{BW}. Among those capacitors, C_{BB} and C_{BW} are common factors for two operations. C_{BW} is not likely to have an effect on neighbor bit-lines.



(a) (b)

Fig. 2. C_{BB} and charge sharing.
그림 2. C_{BB}와 전하 공유

Notice that C_{BW} can be removed by using the folded bit-lines, and a high-density memory means minimum distance between bit-lines, rather than between word-lines and bit-lines. On the other hand, C_{BB} sharply increased as the distance between neighbor bit-lines decrease. The crosstalk, voltage-drop, and voltage-rise are expected in each neighbor bit-lines. Consequently they affect the data of cells especially during high-speed read/write operations. To consider those unexpected effects at maximum, we add refresh operations in the test algorithm. This strategy is satisfactory for the maximum possibility to cause faults based on NPSF. During read/write operations and refresh operations, charge sharing always happens. At that time, voltage level of bit lines is calculated by

$$V_H = (C_B V_{CC} + C_G V_{CC})/(C_B + C_C)$$
$$\triangle V_H = V_H - V_{CC}/2 = (V_{CC}/2)/(1 + C_B/C_C)$$
$$V_L = (C_B V_{CC}/2)/(C_B + C_C)$$
$$\triangle V_L = V_L - V_{CC}/2 = -(V_{CC}/2)/(1 + C_B/C_C)$$

C_C and C_B is about 30fF~40fF and 250~300fF, respectively. In that, C_B is at maximum 7 times larger than C_C, and then the equation can be expressed again by

$$\triangle V_H = (V_{CC}/2)/(1 + C_B/C_C) = 2.5V/8 = 0.3V$$
$$\triangle V_L = -(V_{CC}/2)/(1 + C_B/C_C) = -0.3V$$

By the coupling capacitor, a new C_B is calculated by C_B+C_{BB}. Therefore, the increase of C_B results in the decrease of both $\triangle V_H$ and $\triangle V_L$. In the worst case, the contents of the cell can be changed. If the initial voltage for charge sharing is about less than ± 0.3V, the increase of C_B causes critical problem.

Now consider faults by the coupling capacitance between bit-lines. Both interaction between cells and coupling noise are equivalently considered. And nontransition write as well as transition is also important because coupling noise can appear in nontransition write operations. The B_n of a certain cell $C_{i,j}$ are well affected by adjacent bit-lines $/B_{n-1}$, not by adjacent cells. The coupling noise by neighbor bit-line $/B_{n-1}$ reflects on the cell $C_{i,j}$. Coupling noise around $C_{i,j}$ is activated by read/write/refresh operations of $C_{i-1,j}$

connected to $/B_{i-1}$ and $C_{i+1,j}$ connected to B_{i+1}. This is the main reason to determine tiling shape. In that, every neighborhood cells connected to the same bit-lines are not necessary to test faults caused by neighborhood bit-lines. Every cell in the same word-lines must be tested by proper test patterns. To test faults by coupling noise of bit-lines, we proposed fault models based on NPSFs and the way to test them. Read/write operations to test NPSFs have the period which is n(memory size) because the faults result from interference only between adjacent cells. On the other hand, to test NBLSFs, one write operation should be followed by one refresh operation. The refresh operation should be also followed by one read operation (i.e., one write → one refresh → one read operation).

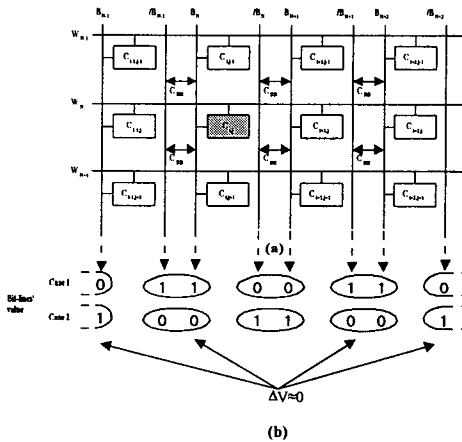


Fig. 3. Memory structure and the condition of minimum coupling noise.
그림 3. 메모리 구조와 최대 상호 노이즈의 조건.

Fig. 3 shows the condition of minimum coupling noise between adjacent bit-lines, whose values are all same. On the other hand, cells data in the same word-lines is opposite one another. It implies that the simultaneous consideration of both maximum leakage current between cells and maximum coupling noise by adjacent bit-lines is impossible.

Fig. 4 shows that crosstalk between bit-lines is to be maximum just after refresh operations, when the data values of adjacent bit-lines are opposite each other. On this condition, frequency characteristics are getting worse. In the refresh mode, $C_{i,j}$ are well affected by both $/B_{n-1}$ and B_{n+1} simultaneously. According to the refresh time and coupling noise, $C_{i,j}$ voltage may not go up or down to VDD or VSS respectively.

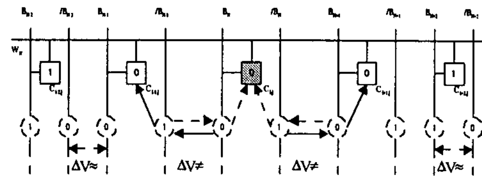


Fig. 4. Maximum coupling noise during a refresh operation.
그림 4. refresh 동작 동안에 최대 상호 노이즈.

That is, cell charge does not reach the expected value. The cell in the following read operation is also under coupling noise because of the leftover voltage of neighborhood bit-lines. Therefore $|\triangle V_{H,\ L}|$ is getting smaller during charge sharing. Consequently sense amplifier cannot amplitude $|\triangle V_{H,\ L}|$ in proper time and then the cell is not accessed for some time. That kind of faults is classified by DRFs(Data Retention Faults). And also data inversion may be expected because of severe coupling noise and a small amount of charge leaking away from the cell.

III. ANBLSFs and PNBLSFs

In this section, new fault models are also developed, introducing conventional faults. And an effective tiling method to detect them is proposed. As the density of memories gets high, test algorithms considering the logical neighborhood are somewhat impractical because of coupling noise.

Therefore, even though the memory arrays' physical and logical neighborhood are not identical, we consider physical neighborhood, rather than logical neighborhood.

The tiling method according to 3 different base cells is shown in Fig. 5(a). The test pattern sequence to sensitize all faults by bit-line coupling noise is shown in Fig. 5(b).

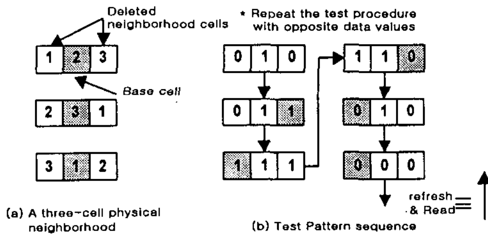


Fig. 5. Test patterns according to label number.
그림 5. 라벨 수에 따른 테스트 패턴.

As coupling noise reaches the maximum value in the refresh mode, one write operation of each test pattern should be followed by one refresh operation. And then in the read mode, the base cell is checked with the expected value.

But the possibility of faults only by neighborhood bit-lines may be low in practice. So we consider coupling noise under the condition of the maximum leakage current by cells. Fig. 6(a) shows the concept. That is, the base cell is surrounded with cells which contain the opposite data value. There are two cases. One is when the deleted neighborhood cells is "0" and the base cell is "1", the other is when the data values are opposite to those of case 1. Fig. 6(b) shows the tiling method over entire cells. Every cell has "0" or "1". Cells in even-lines are marked with a fixed pattern

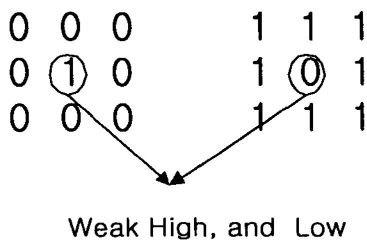
Table 1. Test patterns considering two cases.
표 1. 두가지 경우를 고려하는 테스트 패턴.

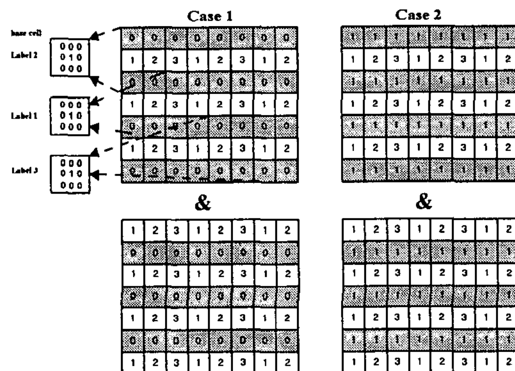
Case #	TP#	L1 L2 L3	TP#	L1 L2 L3	TP#	L1 L2 L3
Case 1	1	1 0 0	7	0 1 0	13	0 0 1
	2	1 1 0	8	0 1 1	14	1 0 1
	3	1 1 1	9	1 1 1	15	1 1 1
	4	1 0 1	10	1 1 0	16	0 1 1
	5	1 0 0	11	0 1 0	17	0 0 1
	6	0 0 0	12	0 0 0	18	0 0 0
Case 2	1	0 1 1	7	1 0 1	13	1 1 0
	2	0 0 1	8	1 0 0	14	0 1 0
	3	0 0 0	9	0 0 0	15	0 0 0
	4	0 1 0	10	0 0 1	16	1 0 0
	5	0 1 1	11	1 0 1	17	1 1 0
	6	1 1 1	12	1 1 1	18	1 1 1

which is a three-cell physical neighborhood as shown in the upper side of Fig. 6 (b).

Table 1 shows test patterns considering maximum leakage and bit-lines coupling noise.

There maybe pattern-sensitive faults(PSFs) which are classified by three categories, ANPSFs(Active Neighborhood PSFs), PNPSF(Passive Neighborhood PSFs), and SNPSFs(Static Neighborhood PSFs).
To sensitize all kinds of PSFs, a Eulerian sequence for ANPSFs and PNPSFs, and a Hamiltonian sequence for SNPSFs should be used. The Eulerian sequence shown in table 2 satisfies table 1. Therefore test pattern by the Eulerian sequence is used for both NPSFs and NBLSFs. And new fault models based on NPSFs are presented.





(a) (b)

Fig. 6. Tiling method considering maximum leakage current.
그림 6. 최대 누수 전류를 고려하는 타일링 방법.

Table 2. Test pattern by a Eulerian sequence.
표2. 오일러 순서에 의한 테스트 패턴

TP#	L1 L2 L3	TP#	L1 L2 L3	TP#	L1 L2 L3
1	0 0 0	9	1 0 1	17	0 1 1
2	0 0 1	10	0 0 1	18	0 0 1
3	0 1 1	11	0 0 0	19	1 0 1
4	0 1 0	12	1 0 0	20	1 1 1
5	1 1 0	13	1 1 0	21	1 1 0
6	1 1 1	14	0 1 0	22	1 0 0
7	1 0 1	15	0 1 1	23	0 0 0
8	1 0 0	16	1 1 1	24	0 1 0

The NBLSFs can be classified by the following two categories:

·ANBLSF(Active NBLSF) in which the base cell changes its contents as a result of changes in the pattern of the neighborhood bit-lines and the neighborhood cells. -- Use an Eulerian sequence

·PNBLSF(Passive NBLSF) in which the contents of the base cell cannot be changed due to the influence of an existing coupling noise in the neighborhood bit-lines and an existing pattern in the neighborhood cells. -- Use an Eulerian sequence

To sensitize all NBLSFs, one write operation must be followed by one refresh operation. And then a read operation can determine that there is a fault or not.

```
Test procedure in detail for NBLSFs is as follows.

Step 1: Initialize memory cells in odd word-lines with "0"
        //odd word-lines => even word-lines for step 6
Step 2: TP[1,2,3]:=TP#n['1,2,3 ']

Step 3:
        for row:=0 to n-1 do
          begin
            If row = even then //=>        row = odd for step 6
              begin
               //WRITE ACTION
                 for col:=0 to n-1 do
                  begin
                   If ( col = Label numbers whose values are changed) then
                     C[row,col]:= TP[col];
                 end;
                Refresh row ;
               // READ ACTION
                for col:=0 to n-1 do
                  begin
                    Data := Base cell
                    if (C[row,col]<>Data) then output('Error at cell', cell);
                 end;
              end;
          end;
Step 4: repeat step 2 and 3 until next TP#1;

Step 5: Initialize memory cells with "1" and repeat step 2, 3, and 4.

Step 6: Change "odd" into "even" and "even" into "odd",
        and then repeat step 1, 2, 3, 4, and 5
```

Fig. 7. Algorithm for NBLSFs detection.
그림 7. NBLSF 검출을 위한 알고리즘.

IV. Algorithm for NBLSFs Detection

The number of test patterns for PNBLSFs and ANBLSFs detection is calculated by 2^k and $(k-1)\times 2k$ respectively, and total test patterns required is $k\times 2^k$. But the controllability is a little difficult because one write operation should be followed by one refresh operation.

Fig. 7 shows test procedure for NBLSFs. Assume that there is one refresh operation after write operations for all the tiling groups in a word-line. Cells in the memory are initialized with "0" for case 1 and "1" for case 2. At first, cells in even word-lines are tiled and tested over case 1 and 2. Cells in odd word-lines are also done in turn. A refresh operation is followed by read operations in the algorithm shown in Fig. 7. If there is an error after read operations, the error sign rises simultaneously.

The test complexity to detect NBLSFs is calculated by

2(case 1 and 2)× 2(read and write)× 24 test patterns × n(memory size) + 2n("0" and "1"× n for even and odd tiling)= 98n.

PNBLSFs (Passive Neighborhood Bit-Line Sensitive Fault) Detection

To sensitize all the PNBLSFs, all possible states of cells should be written and directly read per one word-line.

The number of PNPs is given by 3×2^k(number of base cells × PNPs). Table 3 shows all of the PNPs for PNBLSFs detection.

ANBLSFs (Active Neighborhood Bit-Line Sensitive Fault) Detection

The way to detect ANBLSFs is the same with that to detect ANPSFs except for the test procedure that a write operation must be directly followed by a read operation.

The number of ANPs for ANPSF detection is given by $(k-1)\times 2^k$ per one base cell. Table 4 shows all of the ANPs for ANBLSFs Detection.

CFs(Coupling Faults) Detection

Assume that the base cell shown in Fig. 8 is the coupling cell and others are the coupled cells. And then the deleted neighborhood cells are said to be coupled to

Table 3. PNPs for PNBLSF detection.

표 3. PNBLSF 검출을 위한 PNP.

Base Cell	PNPs	Test Patterns	Base Cell	PNPs	Test Patterns	Base Cell	PNPs	Test Patterns
Label 1	- 0 0	TP#11-12,TP#22-23	Label 2	0 - 0	TP#23-24, TP#24-1	Label 3	0 0 -	TP#1-2, TP#10-11
	- 0 1	TP#18-19, TP#9-10		0 - 1	TP#2-3, TP#17-18		0 1 -	TP#14-15, TP#3-4
	- 1 0	TP#4-5, TP#13-14		1 - 0	TP#12-13, TP#21-22		1 0 -	TP#8-9, TP#7-8
	- 1 1	TP#15-16,TP#16-17		1 - 1	TP#6-7, TP#19-20		1 1 -	TP#5-6, TP#20-21

Table 4. ANPs for ANBLSF detection.

표 4. ANBLSF 검출을 위한 ANP.

Base Cell	ANPs	Test Patterns	Base Cell	ANPs	Test Patterns	Base Cell	ANPs	Test Patterns
Label 1	0 - 0	TP#23-24, TP#24-1	Label 2	- 0 0	TP#11-12,TP#22-23	Label 3	- 0 0	TP#11-12,TP#22-23
	0 - 1	TP#2-3, TP#17-18		- 0 1	TP#18-19,TP#9-10		- 0 1	TP#18-19, TP#9-10
	1 - 0	TP#12-13,TP#21-22		- 1 0	TP#4-5, TP#13-14		- 1 0	TP#4-5, TP#13-14
	1 - 1	TP#6-7, TP#19-20		- 1 1	TP#15-16,TP#16-17		- 1 1	TP#15-16,TP#16-17
	0 0 -	TP#1-2, TP#10-11		0 0 -	TP#1-2, TP#10-11		0 - 0	TP#23-24, TP#24-1
	0 1 -	TP#14-15, TP#3-4		0 1 -	TP#14-15, TP#3-4		0 - 1	TP#2-3, TP#17-18
	1 0 -	TP#8-9, TP#7-8		1 0 -	TP#8-9, TP#7-8		1 - 0	TP#12-13,TP#21-22
	1 1 -	TP#5-6, TP#20-21		1 1 -	TP#5-6, TP#20-21		1 - 1	TP#6-7, TP#19-20

the base cells. The notation <↑ ;↓ > means that an ↑ transition in the base cell causes a ↓ transition in the coupled cells.

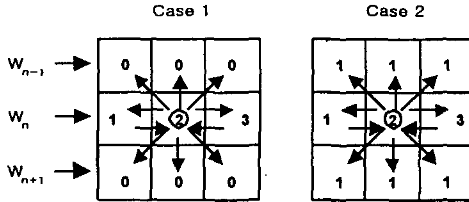


Fig. 8. Coupling size
그림 8. 커플링 크기.

If label 2 is a base cell, a tiling group includes label 1, 2, and 3. cells interference within the *tiling* group can be considered by the Eulerian sequence. But 6 cells in W_{n-1} and W_{n+1} are independent of the transitions of the base cell. Therefore, whenever there are a ↑ (0→ 1) (or ↓ (1→ 0)) transition in the base cell, read operations over 6 cells are performed. As a result of that, CF_{ins}(Inversion Coupling Faults) which inverts the contents of neighborhood cells can be detected in a nine-cell physical neighborhood. CF_{ids}(Idempotent Coupling Faults) which forces the contents of neighborhood cell to a certain value, 0 or 1 can be also detected. There are four possible CF_{ids} which are <↑ ;0>, <↑ ;1>, <↓ ;0>, and <↓ ;1>. During case 1, read operations after a ↑ (or ↓) transition in the base cell can make <↑ ;1> and <↓ ;1> detected. And In case of case 2, <↑ ;0> and <↓ ;0> are also detected after read operations.

The additional test size is calculated by

$$2(\uparrow or \downarrow) \times 6\,\text{cells} \times \text{n(memory size)} = 12n.$$

NPSFs (Neighborhood Pattern Sensitive Fault) Detection

ANPSF(Active NPSFs) and PNPSFs(Passive NPSFs) can be detected by the algorithms for ANBLSF and PNBLSFs. As the Eulerian sequence includes test patterns for SNPSFs(Static NPSFs) detection, SNPSFs can be easily detected.

SAF(Stuck-At Faults), SOFs(Stuck-Open Faults), and TFs(Transition Faults) can be easily detected because every cell in the memory has a ↑ (or ↓) transition and read operations are followed. Table 5 shows all kinds of faults which can be detected by the algorithm presented in this paper.

<table>
<tr><td>number of neighborhood cells = k</td><td>3</td><td>SAFs, SOFs</td><td></td><td rowspan="2">SNPSFs
ANPSFs
PNPSFs</td><td rowspan="2"></td></tr>
<tr><td>Test Patterns = (k X 2^k)</td><td>24</td><td>TFs</td><td></td></tr>
<tr><td>ANBLSFs
PNBLSFs</td><td></td><td>CF_{ins}, CF_{ids}</td><td></td><td>Total test complexity with additonal test size</td><td>110n
(98n + 12n)</td></tr>
</table>

Table 5. Fault coverage of the algorithm.
표 5. 알고리즘의 고장 검출 범위

V. Conclusions

As the density of a memory increases, coupling noise between bit-lines as well as interference between cells sharply increases. In most cases, such coupling noise is not detected by conventional test algorithms, or ignored. To achieve higher fault coverage including faults by bit-line coupling noise, we proposed new fault models for bit-line coupling noise and method to test them. And we also proved that the test procedure was proper method to test NBLSFs including several conventional faults. The test complexity is 98n. The proposed tiling shape can consider maximum leakage current of base cells and coupling noise by neighborhood bit-lines during refresh operation simultaneously. The fault coverage by a three-cell physical neighborhood is lower than that by Type-1 and Type-2 which are conventional tiling groups for NPSFs detection. But one of the most important factors, coupling noise by bit-lines was considered and the test method for it was introduced in this paper. We make sure that this new tiling method is more effective and practical than conventional ones in high-density memories whose coupling noise level by neighborhood bit-lines is high.

Acknowledgement

This work was supported by IC Design Education Center(IDEC) and Ministry of Informations and Communications.

References

[1] T.C.Lo and Mark R. Guidry, " An Integrated Test Concept for Switched-Capacitor Dynamic MOS RAM's," IEEE J. Solid-State Circuits, Vol. SC-12, pp.693-703, December 1977.

[2] J.P.Hayes, "Testing Memories for Single-Cell Pattern-Sensitive Faults," IEEE Trans. Comput., Vol. C-29, No. 3, March 1980.

[3] J.P.Hayes, "Detection of Pattern - Sensitive Faults in RAMs," IEEE Trans. Comput., Vol. C-24, No. 2, pp. 150-157, Feb. 1975.

[4] D.S.Suk and S.M.Reddy, "Test Procedures for a Class of Pattern-Sensitive Faults in Semiconductor Random-Access Memories," IEEE Trans. Comput., Vol. C-29, No. 6, pp. 419-429, June 1980.

[5] D.C.kang and S.B.Cho, "Implementation of a Built-in Self Test Circuit for High Density Memory Test Using a New Tiling Method," ITC-CSCC'98, pp.1781-1784, July 1998.

[6] P.Mazumder and J.K.Patel, "Parallel Testing for Pattern-Sensitive Faults in Semiconductor Random-Access Memories," IEEE Trans. Comput., Vol. 38, No. 3, pp.394-407, March 1989.

[7] H.J.Yoo , " DRAM Design", IDEC, pp. 13-71, 1996.

[8] A.K.Sharma, "Semiconductor Memories," IEEE PRESS, pp. 151-154, 1996.

[9] S.Chou et al., A 60-ns 16Mbit DRAM with a minimized sensing delay caused by bit-line stray capacitance, IEEE J. Solid State Circuits, Vol. 24, No. 5, pp.1176-1183, October 1989

저 자 소 개

姜 東 哲 (正會員)

1971년 5월 20일생
1997년 울산대학교 전자공학 졸업(공학사)
1999년 울산대학교 전자공학 대학원 졸업(공학석사)
2001년 현재 울산대학교 전자공학과 대학원 박사과정수료.
주관심 분야는 테스트 및 테스트 용이한 설계, 메모리 테스트, 임베디드 메모리, 자동차 전자회로 설계

趙 相 福 (正會員)

1955년 6월 10일생. 1979년 2월 한양대학교 전자공학과 졸업(공학사).
1981년 2월 동 대학원 전자공학과 졸업(공학석사). 1985년 2월 동 대학원 전자공학과 졸업(공학박사). 1994년 8월~1995년 8월 Univ. of Texas, Austin 초빙학자.
1986년 3월~현재 울산대학교 교수. 전기전자정보시스템연구소 소장.
주관심 분야는 ASIC 설계, 자동차 전자회로 설계, 비젼 시스템 개발, 테스트 및 테스트 용이한 설계, 메모리테스트 등임