고집적 메모리를 위한 새로운 테스트 알고리즘

강동철, 조상복
울산대학교 전기전자 및 자동화공학부
전화 : (052) 259-2202 / 팩스 : (052) 259-1686

A New Test Algorithm for High-Density Memories

Dong-Chual Kang, Sang-Bock Cho
School of Electrical Engineering and Automation, Univ. of Ulsan
E-mail : kangdc99@orgio.net, sbcho@uou.ulsan.ac.kr

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. From now on, conventional test algorithms have focused on faults between neighborhood cells, not neighborhood bit-lines. In this paper, a new algorithm for NPSFs, and neighborhood bit-line sensitive faults (NBLSFs) based on the NPSFs are proposed. Instead of the conventional five-cell and nine-cell physical neighborhood layouts to test memory cells, a three-cell layout which is minimum size for NBLSFs detection is used. 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 present properties of the algorithm, such as its capability to detect stuck-at faults, transition faults, conventional pattern sensitive faults, and neighborhood bit-line sensitive faults.

1. INTRODUCTION

The pattern-sensitive faults(PSFs) may be considered the most general case of k-coupling faults, for k=n, where n represents all cells in the memory.

This paper was partially supported by IDEC(IC Design Education Center).

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]-[4].

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].

However, those algorithms mentioned above mainly focused on interaction between cells, and partially could consider coupling noise by C_B, C_{BB}, $C_{Pull\text{-}Down}$, and C_{BW}. 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, while the time to test memory cells is increased. From that point of view, the five-cell and nine-cell physical neighborhoods which are typical tiling methods are also complicated to detect PSFs in the memory and 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. We also show that coupling faults(e.g., CF_{id} and CF_{in}) can be detectable within nine physical neighborhood cells [8].

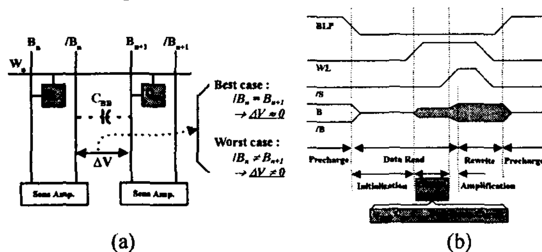
2. Neighborhood Bit-Line Sensitive Faults

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 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 $/B_n$.

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 to the maximum when the values of bit-lines are complement, and down to the minimum when those of neighborhood bit-lines are equal.

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 that the crosstalk happens, the data of a cell can maybe be changeable. As a result, C_{BB} becomes one of the most important factors to be able to delay the response for valid memory operations.

Figure 1(a) and (b) shows C_{BB} between bit lines and the timing diagram for normal operations. As shown in Figure 1(b), 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)

Figure 1. C_{BB} and charge sharing.

Notice that C_{BW} can be mostly disappeared 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. If C_C and C_B is about 30fF~40fF and 250~300fF, respectively, voltage level of bit lines is calculated by

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

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.

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}. 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 which should be also done by one read operation (i.e., one write → one refresh → one read operation).

Cells' data in the same word-lines is opposite one another, but neighbor bit-lines' data is same. It implies that the simultaneous consideration of both maximum leakage current between cells and maximum coupling noise by adjacent bit-lines is impossible.

Figure 2 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. That is, the 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, and so $|V_{H,L}|$ is getting smaller during charge sharing. Consequently sense amplifier cannot amplitude $|V_{H,L}|$ 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.

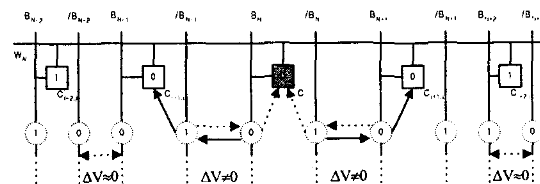


Figure 2. Maximum coupling noise during a refresh

operation.

3. ANBLSFs and PNBLSFs

In this section, new fault models are developed and an effective tiling method to detect them, as well as convention faults is introduced. As the density of memories gets high, test algorithms which consider the logical neighborhood are somewhat impractical because of coupling noise.

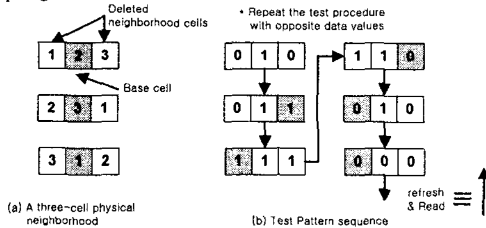


Figure 3. Test patterns according to label number.

Therefore, even thought 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 Figure 3(a), and the test pattern sequence to sensitize all faults by bit-line coupling noise is shown in figure 3(b). 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.

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.

4. ALGORITHM for NBLSFs detection

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

Figure 4 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 Figure 4. If there is an error after read operations, the error sign rises simultaneously.

```
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
```

Figure 4. Algorithm for NBLSFs detection.

The test complexity to detect NBLSFs is calculated by

2(case 1 and 2)$\times$2(read and write)$\times$24 test patterns n(memory size) + 2n('0' and '1'$\times$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).

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 $(k-1)\ 2^k$ per one base cell.

CFs(Coupling Faults) Detection

Assume that the base cell shown in Figure 5 is the coupling cell and others are the coupled cells. And then the deleted neighborhood cells are said to be coupled to

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

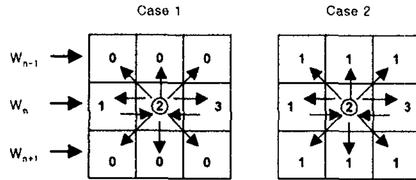


Figure 5. Coupling size

If label 2 is a base cell, cells interference within the tiling group which includes label 1, 2, and 3 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 (↑ or ↓) × 6cells × 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, so SNPSFs can be easily detected.

The most common faults which are 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 1 shows all kinds of faults which can be detected by the algorithm presented in this paper.

Table 1. Fault coverage of the algorithm.

number of neighborhood cells = k	3	SAFs, SOFs	X	SNPSFs ANPSFs PNPSFs	X X X
Test Patterns = ($k \times 2^k$)	24	TFs	X		
ANBLSFs PNBLSFs	X X	CF_{ins}, CF_{ids}	X	*Total test complexity with additonal test size*	110n (98n + 12n)

5. CONCLUSION

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 masked 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 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.

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 Tran. on Compt., 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, "Implementaion 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-Acess 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.