設計自動化　64－6
(1992.10.22)

A Generalized Marching Test for Detecting Pattern Sensitive Faults in RAMs

橋本 昌寛　　藤原 英二

東京工業大学　工学部

東京都 目黒区 大岡山 2-12-1

あらまし

近年、半導体メモリの集積度は飛躍的に増大し、その結果、テスト時間が増大し、テストが困難となってきている。本論文では、ＲＡＭのテストパターンとして、パターンセンシティブ故障検出を目的とした一般化したマーチングパターンを提案している。これは、Ｎセルからなるメモリセルを距離ｄに対応してＢ個の集合に分割したセル集合に対しマーチングテストを行なうことで実現している。この分割数Ｂを変化させて、従来のＮ系パターンであるマーチングパターン($B=1$の場合)からN^2系のウォーキングパターン(Ｂ = Ｎの場合)までの一般化した新しいパターンを得ることができる。実用的な観点より、$B=2$の場合のパターンを提案している。

和文キーワード　　基底セル、近傍セル、静的パターンセンシティブ故障、
動的パターンセンシティブ故障、一般化マーチングテスト

A Generalized Marching Test for Detecting Pattern Sensitive Faults in RAMs

Masahiro HASHIMOTO　and　Eiji　FUJIWARA

Faculty of Engineering, Tokyo Institute of Technology

2-12-1, O-0kayama, Meguro-Ku, Tokyo 152, JAPAN

Abstract

Since semiconductor memory chip has become growing rapidly in its capacity, it requires a lot of time for testing. This paper proposes a generalized marching test for detecting pattern-sensitive faults in RAM chips. This new test executes the ordinary marching test in each memory cell-set obtained by dividing the whole chip having N cells into B cell-sets , in which each two cells has cell-distance d. As a result, by changing the number of set B, this new test includes the ordinary marching test for $B = 1$ and the ordinary walking test for $B = N$, and therefore this can be called a generalized marching test.

英文 key words　specified cell (SC), background cell (BC), static pattern sensitive faults (SPSF),
dynamic pattern sensitive faults (DPSF), generalized marching test

1 Introduction

With significant advance of the semiconductor technologies, the random access memory (RAM) chip has made rapid progress in its capacity. This, however, brings us a serious problem of chip testing, that is, an increase of test execution time.

In order to test RAM chips strictly and efficiently, many test algorithms [1] [2] have been proposed for various fault models. As memory chip circuit is composed of memory cell array, address decoder and read/write logic, there are three fault models such as memory cell array faults, address decoder faults and read/write logic faults. For these fault models, test algorithms are classified into those having test length $O(N^{0.5})$, $O(N)$, $O(N \log_2 N)$, $O(N^{1.5})$ and $O(N^2)$, where $O(N^k)$ denotes "of the order of N^k" and N is the capacity of RAM chip in bits.

It is almost evident that $O(N^{1.5})$ and $O(N^2)$ tests may be unacceptable for very high density RAM chips having capacity of 4 M-bit, 16 M-bit or more. From this standpoint, efficient test algorithms and new test scheme have been proposed; for example, $O(N^{0.5})$ test algorithms which test many cells simultaneously [3] and all cells on a word line simultaneously capable of detecting pattern sensitive faults (PSF) [4] [5] , and built-in self-test (BIST) scheme [6] [7] [2] using checkerbord pattern [8] and using of microcoded ROM [9] [10]. As for $O(N)$ test, the ordinary marching test having complexity of $10N$ operations is now one of the most widely used test algorithm, and several versions of it can be found in the literature [11] [12] such as a linear marching test algorithm having complexity of $30N$ operations [11] and an improved version having $14N$ operations by slightly modifying the fault model [12] . Another interesting $O(N)$ test detecting pattern sensitive faults (PSF) is a waltzing test having $31/3N$ operations whose memory cell array under test is divided into three areas, each having the same residue of the address divided by three [13] [14] [15] [16] . Similar idea of the test partitioning the memory cell array into several areas in which every two cells has at least Hamming distance D between their addresses gives $O(N)$ test for $D = 2$, $O(N \log_2 N)$ test for $D = 3$ and 4, $O(N(\log_2 N)^2)$ for $D = 5$ and 6, etc.[17]. These tests including the Walzing test, however, does not detect all coupling faults and multiple selection faults in the address decoder.

This paper proposes a new $O(N)$ test based on the ordinary marching test. The proposed test, called generalized marching test (GMT), has a characteristic of detecting both static pattern sensitive faults (SPSF) and dynamic pattern sensitive faults (DPSF) as well as detecting other faults such as cell stack faults, coupling faults, decoder faults, etc.. This also has a characteristic of including the walking test and, of course, the ordinary marching test with changing the number of partition in the memory cell array.

In the next Section, we define the fault models and definitions. In Section 3 we propose the GMT capable of detecting SPSF and DPSF, each having cell-distance d. In Section 4 we show an evaluation of the proposed GMT from testing time and fault detection capabilities points of view.

2 Fault models and Definitions

Memory chip circuit is organized mainly from memory cell array, address decoder and read/write logic, as shown in Figure 1. Based on these circuits, there are three fault models shown below.

1. Memory cell faults
   - Cell stack faults
   - Coupling faults
     A pair of cells is said to be coupled if a transition in one of them changes the contents of the other cell from 0 to 1 or 1 to 0.
   - Pattern sensitive faults (PSF)
     A cell is said to have a pattern sensitive fault (PSF) [18] if its content is altered by a pattern of 0's and 1's, or changing the contents from 0 to 1 and/or from 1 to 0 in a group of other memory cells. A cell is said to be a static pattern sensitive fault (SPSF) if its content is altered when a certain pattern of 0's and 1's exists in the neighborhood cells. A cell is said to be a dynamic pattern sensitive fault (DPSF) if its content is altered because of a change in its neighborhood pattern. Often the neighborhood is allowed to take only the position that physically surrounds the base cell. The restricted neighborhoods considered are five-cell, nine-cell physical neighborhoods [19] [20] [21] [22] [23] and a broader row/column neighborhoods [24].
2. Address decoder faults
   - Nonselection faults
   - Erroneous selection faults
   - Multiple selection faults
3. Read/write logic faults
   - Stack faults of output lines of the sense amplifier logic or write driver logic

Most faults occurring in the address decoder and the read/write logic can be mapped to faults in the memory cell array. Therefore these faults will be detected by tests for the memory cell array.

Definition 1
Cell-distance between two cells, e.g., cell 1 and cell 2, is defined as the following:

$$d(cell_1, cell_2) = |x_1 - x_2| + |y_1 - y_2|$$

where (x_1, y_1) and (x_2, y_2) are denoted as their positions in the orthogonal memory array, as shown in Figure 2 . ▮

Definition 2
A cell being tested is called a *base cell*,or a *specified cell (SC)*, and its remaining cells are called *background cells (BC)*. ▮

Definition 3
The cell area in which every cell has cell-distance d from the SC is called *neighborhood cells*. If the contents of the SC are affected and then inversed only by the contents of its neighborhood cells, this kind of faults is called a *static pattern sensitive faults having cell-distance d (SPSF-d)*. If the contents of the SC are potentially affected and then inversed only by changes in the contents of the neighborhood cells, this kind of faults is called a *dynamic pattern sensitive faults having cell-distance d (DPSF-d)*. ▮

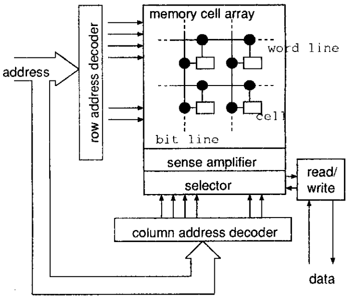


Figure 1: Memory chip circuit

3 Generalized marching test (GMT)

Figure 3 shows test sequence of the ordinary marching test algorithm. Step 1 in the figure shows initialization of the memory array having N cells by writing ' 0 ' (' 1 ') to all memory cells. Step 2 indicates the marching test procedure to read and verify ' 0 ' (' 1 '), and then to write ' 1 ' (' 0 ') at each address from zero to N, i.e., with incresing address order. Step 3 also shows the marching test procedure with inverse address order, i.e., with decending address order, from N to zero, and with using complement of the data operated in the Step 2. These procedures detect cell stack faults, coupling faults and address decoder faults. The above test sequence, however, detects limited cases of the PSF, because not all 4 neighborhood cells have opposite data to the SC, as shown in Figure 4 .

3.1 Partition of the memory cell array for detecting PSF

In the proposed marching test algorithm, i.e., the generalized marching test (GMT), memory cell array having N cells is partitioned into B sets, i.e., C_1, C_2, ... ,C_B, and every two cells in each set has cell-distance at least $d+1$. The GMT is started by choosing the set of SC, called SC-set, in the B sets. In this case, N/B cells in the SC-set having cell-distance at least $d+1$ between each other are distributed in the memory cell array under test . The memory cells except the ones in the SC-set are included in the neighborhood cells. By taking proper values of B and d, the GMT can detect PSF having cell-distance d. For example, partitioning the memory cell array into 5 sets enables to detect PSF having cell-distance 2 because every two cells in a set has cell-distance at least 3. This is shown in Figure 5 . The number shown in each memory cell array, in this figure, shows the set number and the cells hatched with oblique lines are included in the SC-set, e.g., C_1, and others, C_2, C_3, C_4, C_5, are included in the neighborhood sets, called *BC-sets.*

Definition 4
When the memory cell array is partitioned into B sets, C_1, C_2, ... C_B, *cell group* G is defined as the set of cells satisfying the following condition:

1. Group G has B cells, each included in a different set.
2. Every two cells in G has cell-distance less than or equal to d. ∎

Figure 6 shows examples of G for $d = 1$, 2 and 3. In this, a set of cells having different set numbers makes G. Minimum number of partition into sets is determined uniquely for odd number of d and for even number of d. Figure 7 shows the shapes of G for these cases.

Theorem 1
Minimum number of partition B for detecting PSF having cell-distance d is determined by the following equations:
For $d = 2i$ $(i > 0)$

$$\begin{aligned} B &= \frac{d^2}{2} + d + 1 \\ &= 2i^2 + 2i + 1 \qquad (3-1) \end{aligned}$$

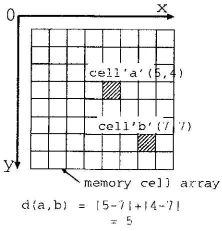


Figure 2: Cell-distance

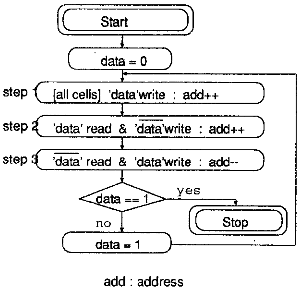

Start
data = 0
step 1
[all cells] 'data'write : add++
step 2
'data' read & $\overline{\text{'data'}}$write : add++
step 3
$\overline{\text{'data'}}$ read & 'data'write : add--
data == 1
yes
no
Stop
data = 1
add : address

Figure 3: Ordinary marching test algorithm

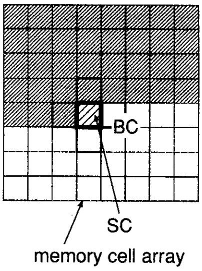


Figure 4: PSF in the ordinary marching test

For $d = 2i + 1$ $(i > 0)$

$$\begin{aligned} B &= \frac{d^2}{2} + d + \frac{1}{2} \\ &= 2i^2 + 4i + 2 \qquad (3-2) \end{aligned}$$

∎

Proof
This can be proved with using the method of mathematical induction.
For $d = 2i$:
(i) For $i = 0$, this makes $B = 1$ which satisfies the minimum number of partition.
(ii) For $i = k$, this makes $d = 2k$. Then, $B = 2k^2 + 2k + 1$ is assumed to be satisfied.
(iii)For $i = k + 1$, this satisfies $d = 2k + 2$ which

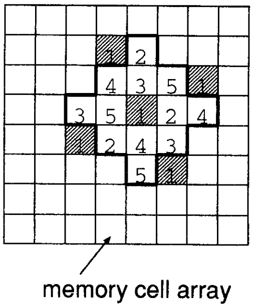


Figure 5: An example of the partition for detecting PSF having cell-distance 2

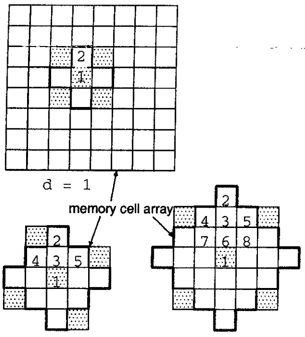


Figure 6: Examples of G for d = 1, 2 and 3

makes G_{2k+2}, i.e., G having $d = 2k + 2$, larger in its size by one than G_{2k}. This can be proved in the following. For $d = 2k$, cell a and cell b whose cell-distance is d can be selected in G_{2k}, and they are located at the most exterior in G_{2k}. If cell a (or cell b) is located at the interior in G_{2k}, i.e., not existed at the edge of G_{2k}, then another cell c included in G_{2k} is existed at the outer side of cell a (or cell b). This shows that the cell-distance between c and b (or a) is at least $d + 1$ and this finally induces contradiction to the definition of G. Therefore, cell a (or cell b) is located at the most outer side in G_{2k}. If cell a' adjacent to outer side of cell a and cell b' adjacent to outer side of cell b are existed, i.e., cell a', cell b' $\notin G_{2k}$, cell-distance between them is $d+2$. Therefore, G_{2k+2} which includes cell a' and cell b' is larger in size by one than G_{2k}. In general, G is in the shape of lozenge, and therefore the minimum number of cells in G_{2k+2}, i.e., B_{2k+2}, is larger than that of G_{2k}, i.e., B_{2k}, by the number of cells in the surroundings of G_{2k}.

$$\begin{aligned} B_{2k+2} &= B_{2k} + 4(k+1) \\ &= 2k^2 + 2k + 1 + 4k + 4 \\ &= 2(k+1)^2 + 2(k+1) + 1 \end{aligned}$$

Therefore, this holds for $i = k + 1$.
For $d = 2i + 1$, this can be proved in the same way as the previous case of $d = 2i$.

(Q.E.D.)

3.2 GMT for static-PSF

Based on the partition mentioned in the Theorem 1, the GMT is performed according to the following algorithm.

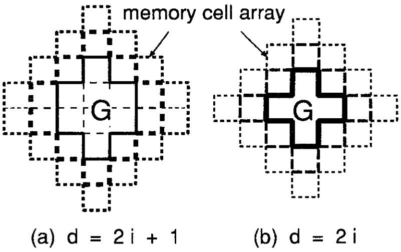
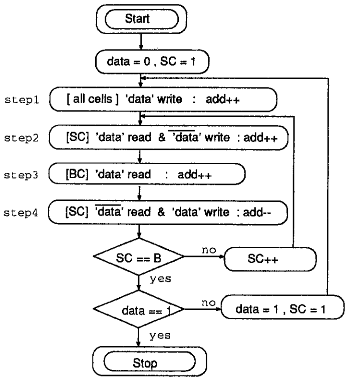


Figure 7: Cell area of G



Start
data = 0 , SC = 1
step1
[all cells] 'data' write : add++
step2
[SC] 'data' read & '$\overline{\text{data}}$' write : add++
step3
[BC] 'data' read : add++
step4
[SC] '$\overline{\text{data}}$' read & 'data' write : add--
SC == B
no
SC++
yes
data == 1
no
data = 1 , SC = 1
yes
Stop

add : address
B : number of partitions
BC : Background-Cells
SC : Specified-Cells

Figure 8: Algorithm 1 for detecting SPSF-d

Algorithm 1

Step 1: Data ' 0 ' is written in all cells of the memory array.

Step 2: Read and verify ' 0 ' and then write ' 1 ' with increasing address order in the SC-set, e.g., C_i.

Step 3: Read and verify ' 0 ' in each BC-set.

Step 4: Read and verify ' 1 ' and then write ' 0 ' with decending address order in C_i.

Step 5: Change the SC-set from C_i to C_{i+1} and perform the above steps from 2 to 4 until $i = B$.

Step 6: Inverse the data from ' 0 ' to ' 1 ' and perform the above steps from 1 to 5.

Figure 8 shows the above algorithm.

Theorem 2
Algorithm 1 can detect static-PSF having cell-distance d, i.e., SPSF-d. ∎

Proof
Let the SC-set be C_i. Every two cells in C_i has cell-distance at least $d + 1$. Marching test procedure always writes the opposite data in every cell in C_i to the contents of all cells in BC-sets. From this, Algorithm 1 detects SPSF-d.

(Q.E.D.)

It is apparent that algorithm 1 can detect cell stack faults, nonselection faults in the address decoder as well as SPSF-d. It also detects erroneous selection faults, multiple selection faults in the address decoder and coupling faults if they occur in a set, because this algorithm performs marching test in each set.

Theorem 3
For given B and N, the memory access number of the algorithm 1 is shown in the following:

Number of write access: $6N$

Number of read access: $2(B + 1)N$ ∎

This can be easily proved and therefore omitted.

This theorem says that for $B = 1$ the proposed test algorithm is identical with the ordinary marching test, and for $B = N$ it is identical with the walking test. We can say from this that the proposed test algorithm is a generalized marching test, because it includes from the $O(N)$ test to the $O(N^2)$ test, and therefore we can get the test having any fault detection capability of SPSF by changing the value of B.

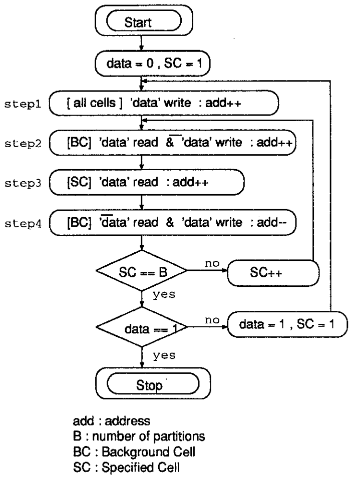


Figure 9: Algorithm 2 for detecting DPSF-d

3.3 GMT for dynamic PSF

Based on the memory array partition in Theorem 1, another GMT detecting DPSF is proposed in the following algorithm.

Algorithm 2

Step 1: Data ' 0 ' is written in all cells of the memory array.

Step 2: Read and verify ' 0 ' and then write ' 1 ' with increasing address order in each BC-set.

Step 3: Read and verify ' 0 ' in C_i.

Step 4: Read and verify ' 1 ' and then write ' 0 ' with decending address order in each BC-set.

Step 5: Change the SC-set from C_i to C_{i+1} and perform the above steps from 2 to 4 until $i = B$.

Step 6: Inverse the data from ' 0 ' to ' 1 ' and perform the above steps from 1 to 5. ▮

Figure 9 shows the above algorithm.

Theorem 4
Algorithm 2 can detect dynamic PSF having cell-distance d, i.e., DPSF-d. ▮

Proof
Every two cells in C_i has cell-distance at least $d+1$. Marching test procedure always writes the opposite data in every cell in the BC-sets to the contents of the cells in C_i. From this, Algorithm 2 detects DPSF-d.

($Q.E.D.$)

Algorithm 2 can also detect cell stack faults, nonselection faults in the address decoder as well as DPSF-d. It also detects erroneous selection faults, multiple selection faults in the address decoder and coupling faults if they occur in a set, because this algorithm performs marching test in each set.

Theorem 5
For given B and N, the memory access number of the algorithm 2 is shown in the following:

Number of write access: $2(2B-1)N$

Number of read access: $2(2B-1)N$ ▮

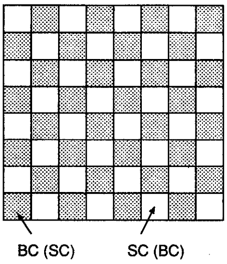


Figure 10: Memory cell array partition for practical GMT with $B = 2$

This can be easily proved and therefore omitted.

Changing the number B gives any fault detection capability of the DPSF-d. That is, if B is larger in number, then stronger disturbance is given to each SC, because every SC is surrounded by larger number of BC's having opposite data to the contents of SC. Compared to the Algorithm 1, this requires larger memory access number for testing and from this we have to choose an appropriate value of B from the viewpoint of testing time.

3.4 Practical GMT

The GMT having $d = 1$, that is, $B = 2$ from the Eq.(3-2), is considered to be most practical, because every SC is adjacent to 4 BC's and therefore it is directly and most strongly affected by these 4 adjacent BC's, and also this test algorithm has small number of memory accesses, that is, $12N$. Figure 10 shows the partition of the memory cell array in this case. The test for DPSF-1 is identical with that for SPSF-1, and therefore it has advantages of both GMT's.

4 Evaluation

Table 1 lists the major RAM test algorithms including the proposed GMT and compares from the viewpoint of testing time, i.e., memory access number, and fault detection capabilities. This says that the GMT with $B = 2$ has very high fault detection capabilities, almost comparable to the walking test, and has the highest fault detection capabilities among the $O(N)$ tests of the checkerboard test, the waltzing test and the ordinary marching test. Figure 11 shows the relation between the memory access number for testing and the memory capacity in bits for various test algorithms including the proposed GMT.

5 Conclusion

This paper has proposed a new RAM test algorithm, called a generalized marching test (GMT), capable of detecting both static and dynamic pattern sensitive faults. The pattern sensitive faults (PSF) having cell-distance d is newly defined and algorithms detecting these PSF are proposed. Especially, the GMT for static-PSF includes the ordinary marching test (with $B = 1$) and the walking test (with $B = N$). The GMT with $B = 2$ is the most appropriate one for practical use having small testing time.

Future study remains in autonomous pattern generation of the GMT from BIST standpoint.

Table 1: Fault detection capabilities of GMT

Test Algorithm	Memory access number	Address Decoder faults		Memory cell array faults			
		non selection	Maltiple Erroneous	Cell stack	Coupling	PSF	
						static	dynamic
Checkerboard	4N	◎	×	◎	×	△	×
GMT(B=1) [Ordinary Marching]	10N	◎	◎	◎	◎	×	×
Waltzing	$\frac{31N}{3}$	◎	○	◎	○	△	△
GMT(B=2) [Practical GMT]	12N	◎	◎	◎	◎	△	△
Butterfly	$8N^{1.5}+2N$	◎	◎	◎	◎	○	△
Galloping	$4N^{2}+2N$	◎	◎	◎	◎	○	△
GMT(B=N) using Algorithm 1 [Walking]	$2N^{2}+6N$	◎	◎	◎	◎	◎	△
GMT(B=N) using Algorithm 2	$8N^{2}-4N$	◎	◎	◎	◎	◎	◎

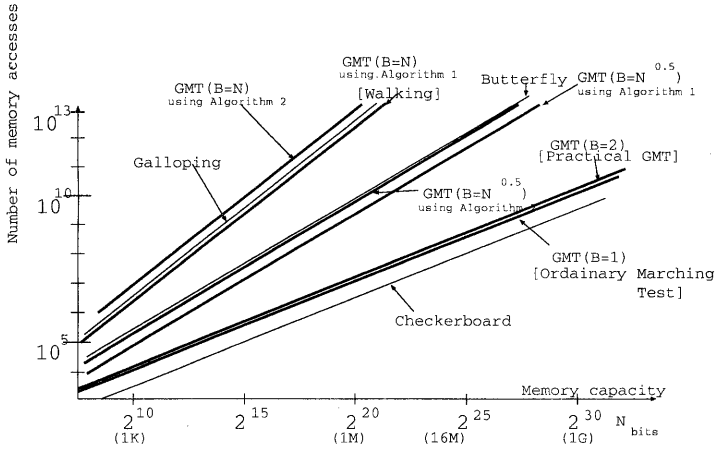


Figure 11: Number of memory accesses for GMT

References

[1] H.Tamamoto, "Design for Testability in Memories" (in Japanese), J. IPS, 30, 12, pp.1467-1472, Dec. 1989.

[2] M.Franklin and K.K.Saluja, "Built-in Self-Testing of Random-Access Memories", IEEE Computer, pp.45-56, October 1990.

[3] P.Mazumder and J.K.Patel, "Parallel Testing for Pattern-Sensitive Faults in Semiconductor Random-Access Memories", IEEE Trans. Comput., 38, 3, pp.394-407, March 1989.

[4] Y.Miura, H.Tamamoto and Y.Narita, "A Built-in Testing for Functional Testing in Semiconductor Random Access Memories", (in Japanese) Trans. IEICE, J69-D, 10, pp.1416-1423, October 1986.

[5] Y.Miura, H.Tamamoto and Y.Narita, "A Built-in Concurrent Testing for Semiconductor Random Access Memories by Concurrently Testing Cells on a Word Line", (in Japanese) Trans. IEICE, J70-D, 6, pp.1116-1125, June 1987.

[6] S.K.Jain and C.E.Stroud, "Built-in Self Testing of Embedded Memories", IEEE Design and Test of Computers, 3, 5, pp.27-37, Oct. 1986.

[7] K.K.Saluja, S.H.Sng and K.Kinoshita, "Built-in Self-Testing RAM: A Practical Alternative", IEEE Design and Test of Computers, 4, 1, pp.42-51, Feb. 1987.

[8] T.Ohsawa, et al., "A 60-ns 4-Mbit CMOS DRAM with Built-In Self-Test Function", IEEE J. Solid-State Circuit, SC-22, 5, pp.663-668, Oct. 1987.

[9] K.Kinoshita and K.K.Saluja, "Built-in Testing of Memory Using an On-Chip Compact Testing Scheme", IEEE Trans. Comput. C-35, 10, pp.862-870, Oct. 1986.

[10] T.Takeshima, et al., "A 55 ns 16 M DRAM", Proc. 1989 ISSCC, pp.246-247, Feb. 1989.

[11] R.Nair, S.M.Thatte and J.A.Abraham, "Efficient Algorithms for Testing Semiconductor Random-Access Memories", IEEE Trans. Comput., C-27, 6, pp.572-576, June 1978.

[12] D.S.Suk and S.M.Reddy, "A March Test for Functional Faults in Semiconductor Random Access Memories", IEEE Trans. Comput., C-30, 12, pp.982-985, Dec. 1981.

[13] J.Knaizuk,Jr. and C.R.P.Hartmann, "An Algorithm for Testing Random Access Memories", IEEE Trans. Comput., C-26, 4, pp.414-416, April 1977.

[14] J.Knaizuk,Jr. and C.R.P.Hartmann, "An Optimal Algorithm for Testing Stuck-at Faults in Random Access Memories", IEEE Trans. Comput., C-26, 11, pp.1141-1144, Nov. 1977.

[15] M.Kishi, "Testing Method Using the Walzing Pattern for IC Memory Devices" (in Japanese), Trans. IECE, 60-D, 12, pp.1031-1038, 1977.

[16] P.H.Bardell,Jr. and W.H.McAnney, "Self-Test of Random Access Memories", Proc. 1985 Int. Test Conference, pp.352-355, 1985.

[17] T.Ishikawa and K.Matzuzawa, "Memory Test-Pattern Considering Hamming Distance between Addresses" (in Japanese), Trans. IECE, J64-D, 9, pp.807-814, 1977.

[18] J.P.Hayes, "Detection of Pattern-Sensitive Faults in Random-Access Memories", IEEE Trans. Comput., C-24, 2, pp.150-157, Feb. 1975.

[19] V.P.Srini, "Fault Location in a Semiconductor Random-Access Memory Unit", IEEE Trans. Comput., C-27, 4, pp.349-358, April 1978.

[20] J.P.Hayes, "Testing Memories for Single-Cell Pattern-Sensitive Faults", IEEE Trans. Comput., C-29, 3, March 1980.

[21] D.S.Suk and S.M.Reddy, "Test Procedures for a Class of Pattern-Sensitive Faults in Semiconductor Random-Access Memories", IEEE Trans. Comput., C-29, 6, pp.419-429, June 1980.

[22] K.K.Saluja and K.Kinoshita, "Test Pattern Generation for API Faults in RAM", IEEE Trans. Comput., C-34, 3, pp.284-287, March 1985.

[23] P.D.Jong and A.J.van de Goor, "Test Pattern Generation for API Faults in RAM", IEEE Trans. Comput., C-37, 11, pp.1426-1428, Nov. 1988.

[24] M.Franklin, K.K.Saluja and K.Kinoshita, "A Built-in Self-Test Algorithm for Row/Column Pattern Sensitive Faults in RAM's", IEEE J. Solid-State Circuit, 25, 2, pp.514-524, April 1990.