An Effective Test and Diagnosis Algorithm for Dual-Port Memories

Youngkyu Park, Myung-Hoon Yang, Yongjoon Kim, Dae-Yeal Lee, and Sungho Kang

This paper proposes a test algorithm that can detect and diagnose all the faults occurring in dual-port memories that can be accessed simultaneously through two ports. In this paper, we develop a new diagnosis algorithm that classifies faults in detail when they are detected while the test process is being developed. The algorithm is particularly efficient because it uses information that can be obtained by test results as well as results using an additional diagnosis pattern. The algorithm can also diagnose various fault models for dual-port memories.

Keywords: Dual-port memories, test algorithm, diagnosis, fault classification, fault model, fault dictionary, fault primitive.

Manuscript received Nov. 20, 2007; revised May 22, 2008; accepted June 27, 2008.

This work was supported by "System IC 2010" project of Korea Ministry of Knowledge Economy.

Youngkyu Park (email: hipyk@soc.yonsei.ac.kr), Myung-Hoon Yang (email: mhyang@soc.yonsei.ac.kr), Yongjoon Kim (email: yjkim@soc.yonsei.ac.kr), Dae-Yeal Lee (email: eto10@soc.yonsei.ac.kr), and Sungho Kang (phone: + 82 2 2123 2775, email: shkang@yonsei.ac.kr) are with the Department of Electical & Electronic Engineering, Yonsei University, Seoul, Rep. of Korea.

I. Introduction

Due to improvements in circuit design and manufacturing techniques, circuits are becoming more complex as the demand for memories with large capacity is increasing. Since highly integrated memories require a complex and precise design and process, they are more likely to have faults [1]. Therefore, it is very important to reproduce faulty memories through repair and to grasp the design and process problems.

Recently, dual-port memories have become popular [2], [3]. Since dual-port memories can be accessed simultaneously through two ports, they are used to process a large amount of data in a short time for image processing, microprocessors, network applications, and so on. Dual-port memories are composed of more complex circuitry than single-port memories. Because of these complexities, it has become harder and more expensive to test and diagnose dual-port memories than single-port memories. There are two objects of fault diagnosis. One is to obtain information about faulty sites, and the other is to obtain information about the type of faults. Fault sites are needed to reproduce faulty memories through the repair of a faulty cell. The information about the kinds of faults is useful to grasp the design and process problems. Unlike other logic circuits, since memories are accessed using addresses, fault sites are easily diagnosed from the address information. However, additional diagnostic algorithms or fault dictionaries are necessary to obtain information about the types of faults.

Memory fault diagnosis is classified into two types of procedures: detecting fault locations and identifying fault types. Information about defective cell addresses enables the detection of fault locations. Additional algorithms and fault dictionaries, however, are indispensable for diagnosing fault

types. Previous detection methods apply fault dictionaries to test algorithms to detect faults; hence, they are incapable of diagnosing various types of faults. Moreover, they show weaknesses in detecting fault types when glitches occur simultaneously at the memory cells of dual-ports capable of I/O data function. They also incur long testing times. Therefore, it is essential to develop a test algorithm which can detect a wide range of faults that occur in dual-port memories [4], [5]. Furthermore, such a test algorithm can generate a fault dictionary, and dual-port memories can be effectively tested by minimizing the length of additional diagnosis algorithms.

In this paper, we introduce a new test algorithm developed for fault detection in dual-port memories described in previous research. We propose a diagnosis algorithm based on this test algorithm. The proposed algorithm maximizes the efficiency of diagnosis by using both a fault dictionary and additional diagnosis algorithms. It effectively detects a wide range of fault types that occur in dual-port memories.

II. Fault Model

One difficulty of existing diagnosis algorithms is that they do not take into consideration various fault models. The diagnosis algorithm depends on the test algorithm. The diagnosis algorithm can be applied differently, depending on the kinds of fault models detected by a test algorithm. Therefore, we introduced a test algorithm in our previous work and now propose a new diagnosis algorithm based on that test algorithm which accounts for the previously omitted fault models.

In [6], test algorithms were proposed based on the fault models presented in [7]. That work suggested new fault models using inductive fault analysis (IFA), which considers spot defects such as opens, shorts, and bridges in memory cells. The reason for targeting these faults is that the fault models effectively consider all defects in dual-port memories. The probability of fault occurrence in dual-port memories is shown in Fig. 1. In this section, functional fault models for dual-port memories are explained, and target fault models are proposed.

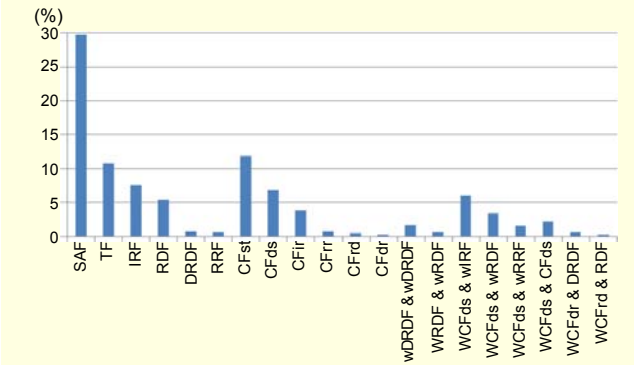


Fig. 1. Probability of fault occurrence.

1. Faults Related to One Port (1PFs)

The faults related to one port (1PFs) are classified into two types: those related to one cell (1PF1) and those related to two cells (1PF2). The 1PF1 faults can be caused by short or open nodes of a memory cell. Table 1(a) lists the kinds of 1PF1s. The notations of the fault primitives (FP) in Table 1(a) are from [7]. Let $\forall$ be any operation, w↑(w↓) be a rising (falling) transition, w1 (w0) be writing 1 (0) and r1 (r0) be reading 1 (0). An FP involving a 1PF1 is denoted by <S/F/R>. Also, S denotes the state or the operation sensitizing the fault, F denotes the data of the faulty cell, and R denotes the output result of the read operation. A 1PF2 fault occurs due to a short or interference between the nodes of two memory cells. There are two kinds of fault cells: aggressor and victim cells. Table 1(b) is a list of the kinds of 1PF2s. An FP involving a 1PF2 is denoted by <Sa;Sv/F/R>. The operation or the state of the aggressor cell sensitizing the fault is denoted by Sa, and the operation or the state of the victim cell sensitized by the fault is denoted by Sv.

Table 1. Target faults.

(a) Fault models involving one cell (1PF1)

Fault model	Fault primitives
SAF	<∀/0/->,<∀/1/->
TF	<w↑/0/->,<w↓/1/->
RDF	<r0/↑/1>,<r1/↓/0>
DRDF	<r0/↑/0>,<r1/↓/1>
IRF	<r1/1/0>,<r0/0/1>
RRF	<r0/0/?>,<r1/1/?>

(b) Fault models involving two cells (1PF2)

Fault model	Fault primitives
CFds	<wx; 1/↓/->, <rx; 0/↑/->, <rx; 1/↓/->, <wx; 0/↑/->
CFst	<1; 1/0/->, <1; 0/1/->, <0; 1/0/->, <0; 0/1/->
CFir	<0; r0/0/1>, <0; r1/1/0>, <1; r0/0/1>, <1; r1/1/0>
CFrr	<0; r0/0/?>, <0; r1/1/?>, <1; r0/0/?>, <1; r1/1/?>
CFdr	<0; r0/↑/0>, <0; r1/↓/1>, <1; r0/↑/0>, <1; r1/↓/1>
CFrd	<0; r0/↑/1>, <0; r1/↓/0>, <1; r0/↑/1>, <1; r1/↓/0>
CFtr	<0; w↓/1/->, <0; w↑/0/->, <1; w↓/1/->, <1; w↑/0/->

Table 2. Fault models related to two ports.

	Fault model	Fault primitives
2PF1	wDRDF & wDRDF	<r0; r0/↑/0>, <r1:r1/↓/1>
	wRDF & wRDF	<r1:r1/↓/0>, <r0:r0/↑/1>
2PF2v	wCFds & wCFds	<rx:rx; 0/↑/->, <rx:rx; 1/↓/->
2PF2a	wCFdr & wDRDF	<0; r0:r0/↑/0>, <1; r0:r0/↑/0>, <0; r1:r1/↓/1>, <1; r1:r1/↓/1>
	wCFrd & wRDF	<0; r1:r1/↓/0>, <1; r1:r1/↓/0>, <1; r0:r0/↑/1>, <0; r0:r0/↑/1>
2PF2av	wCFds & wRDF	<w0:r0/↑/1>, <w0:r1/↓/0>, <w1:r0/↑/1>, <w1:r1/↓/0>
	wCFds & wIRF	<w0:r0/0/1>, <w0:r1/1/0>, <w1:r0/0/1>, <w1:r1/1/0>
	wCFds & wRRF	<w0:r0/0/?>, <w0:r1/1/?>, <w1:r0/0/?>, <w1:r1/1/?>

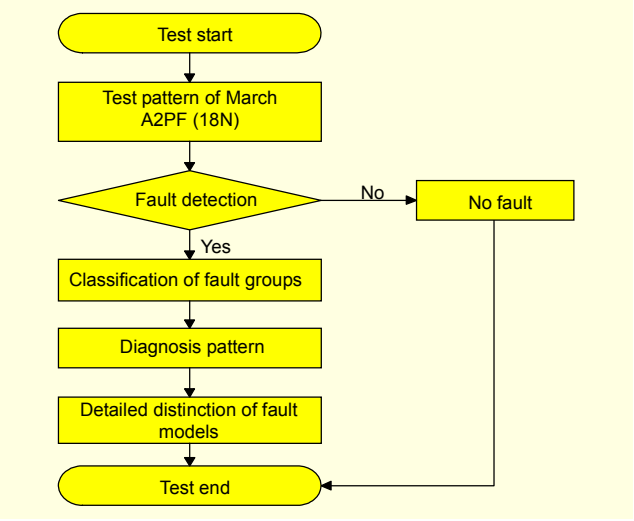


Fig. 2. Flowchart of test and diagnosis procedure.

2. Faults Related to Two Ports (2PFs)

The faults related to two ports (2PFs) may occur when one or two cells are accessed simultaneously through two ports. 2PFs occur as one fault uniting two weak faults and are classified into two types: those related to one cell (2PF1) and those related to two (2PF2). The class of 2PF2 is further divided into three groups according to the behavior of the victim and aggressor cells as shown in Table 2. A 2PF1 is denoted by <S1:S2/F/R>. Here, S1 and S2 denote the operations or the states of the v-cell sensitizing the fault. The symbol ':' denotes that S1 and S2 are applied simultaneously through two ports. A 2PF2a is denoted by <Sa:Sa;Sv/F/R> and <Sa;Sv:Sv/F/R>, a 2PF2v is denoted by <Sa;Sv:Sv/F/R>, and a 2PF2av is denoted by <Sa:Sv/F/R>. The target fault models are chosen based on the assumption that only the following operations can be executed in dual-port memories:

- two simultaneous read operations to the same cell,
- two simultaneous read operations to different cells,
- two simultaneous write operations to different cells, and
- simultaneous read and write operations to different cells.

Some faults in 2PF categories can occur when read and write operations are applied simultaneously to the same cell. There is an assumption that is impossible to read and to write simultaneously at the same cell; therefore, this scheme excludes such faults.

III. Test and Diagnosis Procedure

The diagnosis algorithm proposed in this paper detects fault types using the flowchart shown in Fig. 2. The test process begins with a memory test algorithm, and is implemented by applying the test patterns consisting of 18*N* of the March A2PF algorithm to memory cells. This process is followed by a diagnosis procedure that examines the existence of faults in dual-port memories. Using the response information and fault dictionaries obtainable from the test process during the diagnosis procedure, fault groups are classified from types A through X. After faults are detected during the diagnosis procedure, detailed information on the types of faults can be further analyzed with the application of additional diagnosis patterns. The absence of fault detection is considered to indicate that there are no defects in memory and so puts an end to the algorithm.

IV. Test Algorithm and Fault Dictionary

Diagnosis patterns depend on test patterns, because the targets of the diagnosis are the fault models detected by the test pattern. Test patterns have to detect all fault models and they must make them easy to diagnose. After a test pattern is selected, a fault dictionary is created by predicting test results. We introduce a test pattern that makes it easy to diagnose dual-port memories failures.

1. Test Algorithm

The test algorithm from [6] is used to detect faults that occur in dual-port memories. As for the dual-port memories, a special test is needed not only for fault detection in a single port, but also for the detection of faults that occur because of the simultaneous access of two ports. The algorithm shown in Fig. 3 is the March A2PF test algorithm proposed in [6]. March A2PF is capable of detecting all the faults of single-port and dual-port fault types using a single algorithm. It is a highly efficient algorithm that can

⇕(w0 : –);
①
⇑(r0 : r0, w1: r0,w1:r1,r1: r1); ⇑(r1:r1,w0:r1,w0: r0,r0:r0);
② ③ ④ ⑤ ⑥ ⑦ ⑧ ⑨
⇓(r0 : r0, w1: r0,w1:r1,r1: r1); ⇓(r1:r1,w0:r1,w0: r0,r0:r0);
⑩ ⑪ ⑫ ⑬ ⑭ ⑮ ⑯ ⑰
⇕(r0 : –);
⑱

Fig. 3. Test algorithm of dual-port memories.

$\Updownarrow_{c=0}^{C-1}(\Updownarrow_{r=0}^{R-1}(w0_{r,c} : -))$;
$\Uparrow_{c=0}^{C-1}(\Updownarrow_{r=0}^{R-1}(r0_{r,c} : r0_{r,c}, w1_{r,c} : r0_{r+1,c}, w1_{r,c} : r1_{r+1,c}, r1_{r,c} : r1_{r,c}))$;
$\Uparrow_{c=0}^{C-1}(\Updownarrow_{r=0}^{R-1}(r1_{r,c} : r1_{r,c}, w0_{r,c} : r1_{r+1,c}, w0_{r,c} : r0_{r+1,c}, r0_{r,c} : r0_{r,c}))$;
$\Downarrow_{0}^{c=C-1}(\Downarrow_{0}^{r=R-1}(r0_{r,c} : r0_{r,c}, w1_{r,c} : r0_{r,c-1}, w1_{r,c} : r1_{r,c-1}, r1_{r,c} : r1_{r,c}))$;
$\Downarrow_{0}^{c=C-1}(\Downarrow_{0}^{r=R-1}(r1_{r,c} : r1_{r,c}, w0_{r,c} : r1_{r,c-1}, w0_{r,c} : r0_{r,c-1}, r0_{r,c} : r0_{r,c}))$;
$\Updownarrow_{c=0}^{C-1}(\Updownarrow_{r=0}^{R-1}(r0_{r,c} : -))$;

Fig. 4. March A2PF algorithm including variation of row and column elements.

detect all kinds of faults in dual-port memories using a short test pattern length of 18*N*, where *N* is the number of memory cells.

March A2PF consists of 18 March elements (①-⑱), giving the total test pattern length of 18*N*. Each March element creates its own fault dictionary based on the detection results. A fault dictionary is created by using the results of each operation. Here, w0 (w1) and r0 (r1) denote write 0 (1) and read 0 (1). Also, ':' denotes that operations are applied simultaneously through the two ports on either side of the symbol. After the operations in parentheses '( )' are accomplished, addresses are changed. Here, '⇑', '⇓', and '⇕' denote change of addresses: '⇑' denotes increasing addresses, '⇓' denotes decreasing addresses, and '⇕' denotes either increasing or decreasing addresses. In this paper, '⇕' denotes increasing addresses at ① and decreasing addresses at ⑱. Figure 4 shows the increase/decrease factors of the row and column address in each March element of the test algorithm presented in Fig. 3. In Fig. 4, the March A2PF algorithm shown comprises several March elements, such as '$w1_{r,c}$:$r0_{r+1,c}$', '$w1_{r,c}$:$r0_{r,c-1}$', and so on. These March elements are used to detect the 2PF2av fault model shown in Table 2. If there is a 2PF2av fault, the March elements of the algorithm test and detect the victim cell adjacent to the aggressor cell. For example, if the victim cell is in the same row or column as the aggressor cell, the March element will be '$w1_{r,c}$:$r0_{r+1,c}$' or '$w1_{r,c}$:$r0_{r,c-1}$' respectively. Thus, the arrows of the March A2PF shown in Fig. 4 indicate an increase/decrease in the row and column of the memory cell.

Table 3. Fault dictionary of 1PFs.

		①	②	③	④	⑤	⑥	⑦	⑧	⑨	⑩	⑪	⑫	⑬	⑭	⑮	⑯	⑰	⑱	Fault group
1PFs : 1PF1																				
Stuck-at fault (SAF)																				
<∀/0/->					S	D	D						S	D	D					A
<∀/1/->		S	D						S	D	D						S	D	D	B
Deceptive read destructive fault (DRDF)																				
<r0/↑/0>										S	D							S	D	C
<r1/↓/1>						S	D							S	D					D
1PFs : 1PF2																				
State coupling fault (CFst)																				
<1;1/0/->	a>v				S	D	D													E
	a<v												S	D	D					F
<1;0/1/->	a>v								S	D	D									G
	a<v																S	D	D	H
<0;1/0/->	a>v				S	D	D													E
	a<v												S	D	D					F
<0;0/1/->	a>v	S	D						S	D	D									I
	a<v																S	D	D	H
Deceptive read destruction coupling fault (CFdr)																				
<0;r0/↑/0>	a>v									S	D									O
	a<v																	S	D	P
<0;r1/↓/1>	a>v					S	D													J
	a<v													S	D					K
<1;r0/↑/0>	a>v									S	D									O
	a<v																	S	D	P
<1;r1/↓/1>	a>v					S	D													J
	a<v													S	D					K

2. Fault Dictionary

For an efficient diagnosis, the information obtained during the test should be maximally utilized. Information about the existence (or nonexistence) of a fault and operations that cause the fault is obtained. To use the information, a fault dictionary must be created before the test. The fault dictionary is a list of expected test results for each fault that exists in a memory cell array. The fault dictionary depends on the test pattern. Information about the existence of faults and the locations where such faults are detected can be obtained from the use of test patterns in dual-port memory test algorithms. Such information includes prior formation of the fault dictionary based on the applicable test patterns. A fault dictionary refers to the formation of an individual detection list for the fault models considered by test patterns. Such a fault dictionary varies according to the patterns of the test algorithm applied to memory.

Table 4. Fault dictionary of 2PFs.

		①	②	③	④	⑤	⑥	⑦	⑧	⑨	⑩	⑪	⑫	⑬	⑭	⑮	⑯	⑰	⑱	Fault group
2PFs : 2PF1																				
wRDF & wRDF																				
<r0;r0/↑/1>			SD							SD	D							SD	D	B
<r1:r1/↓/0>						SD	D							SD	D					A
2PFs : 2PF2a																				
wCFds & wCFds																				
<r0;r0;0/↑/->	a>v		SD							SD	D							SD	D	I
	a<v																	SD	D	H
<r0;r0;1/↓/->	a>v		S			D	D													E
	a<v										S			D	D					F
<r1;r1;0/↑/->	a>v						S			D	D									G
	a<v														S			D	D	H
<r1;r1;1/↓/->	a>v					SD	D													E
	a<v													SD	D					F
2PFs : 2PF2v																				
wCFdr & wDRDF																				
<0;r0;r0/↑/0>	a>v									S	D									O
	a<v																	S	D	P
<0;r1;r1/↓/1>	a>v					S	D													J
	a<v													S	D					K
<1;r0;r0/↑/0>	a>v									S	D									O
	a<v																	S	D	P
<1;r1;r1/↓/1>	a>v					S	D													J
	a<v													S	D					K
2PFs : 2PF2av																				
wCFds & wIRF																				
<w0;r0/0/1>	a>v								SD											Q
	a<v																SD			R
<w0;r1/1/0>	a>v							SD												S
	a<v															SD				T
<w1;r0/0/1>	a>v			SD																U
	a<v											SD								V
<w1;r1/1/0>	a>v				SD															W
	a<v												SD							X

In this study, faults are detected using the March A2PF algorithm without any extra test pattern. The list of conditions for fault detection is generated through segmentation by FPs of fault models. This paper proposes a new fault dictionary that leads to a detailed display of fault results according to the location of the aggressor and victim cells for the purpose of identifying fault types. Since the diagnosis algorithm proposed in this paper takes various fault models of single-port memory and dual-port memories into consideration, it offers information which is more accurate and detailed than that of the existing dictionary. The same fault model can take different test responses depending on the location relationship between FPs, aggressor cells, and victim cells. Accordingly, the new fault dictionary offers precise and detailed information for diagnosis. The fault dictionary of 1PFs and 2PFs is shown in Tables 3 and 4, respectively. In these tables, S denotes operations that sensitize faults, and D denotes operations that detect faults. The actual dictionary is marked 0 or 1 at each read operation. Operations that do not detect faults are denoted by 0, and operations that detect faults are denoted by 1. We mark S and D in this paper for emphasis. Tables 3 and 4 display the fault dictionary that indicates the conditions for fault detection in March elements of the test algorithm by the FPs of the fault model considered in the March A2FP dual-port memories algorithm. The test results offer information about the location of faults occurring in the test algorithm; however, the faults detected in the same operations cannot be identified based on such test results. Therefore, faults detected in the same operations are divided into fault groups. This paper presents several dictionaries created using fault models. The fault dictionary groups detected faults derived from the same operations together, enumerating fault groups from A to X. The full fault dictionary for dual-port memories is shown in appendix.

For example, the fault dictionary for stuck-at faults is created using the following method. A datum in a cell which has a stuck-at fault cannot be changed through any operations that are applied to the memory cell. The stuck-at fault model is divided into two fault primitives, stuck-at zero <∀/0/-> and stuck-at one <∀/1/->. Here, <∀/0/-> denotes a fault in which the cell is stuck at the value 0. If a circuit has a fault primitive <∀/0/->, the write 1 operations of ④ and ⑫ sensitize the fault, and the read 0 operations of ⑤, ⑥, ⑬, and ⑭ detect the fault. Therefore, ④ and ⑫ are marked S, and ⑤, ⑥, ⑬, and ⑭ are marked D. A fault in which the cell is stuck at the value 1 is denoted by <∀/1/->. Therefore ①, ⑧, and ⑯, are marked S and ②, ⑨, ⑩, ⑰, and ⑱ are marked D. Group A is formed of <∀/0/-> faults, and group B is formed of <∀/1/-> faults.

V. Diagnosis Algorithm

Fault types can be detected by applying additional diagnosis patterns to the 24 fault groups. The following conditions are used in identifying FPs by fault models in each group.

- Whether the fault is sensitized using one port or two ports.
- Whether the fault is involved with one cell or two cells.
- The conditions of the aggressor and victim cells that sensitize faults (to distinguish coupling faults).

Table 5. Distinguishing features of coupling faults.

Faults	Conditions		Data of the victim cell and output results of the read operation (F:R)
	Aggressor cell	Victim cell	
CFds	Read or write operation	Specific state	Same
CFst	Specific state	Specific state	Same
CFir	Specific state	Read operation	Different
CFrr	Specific state	Read operation	Different
CFdr	Specific state	Read operation	Same at first read operation and different at second read operation
CFrd	Specific state	Read operation	Same
CFtr	Specific state	Write operation	Same

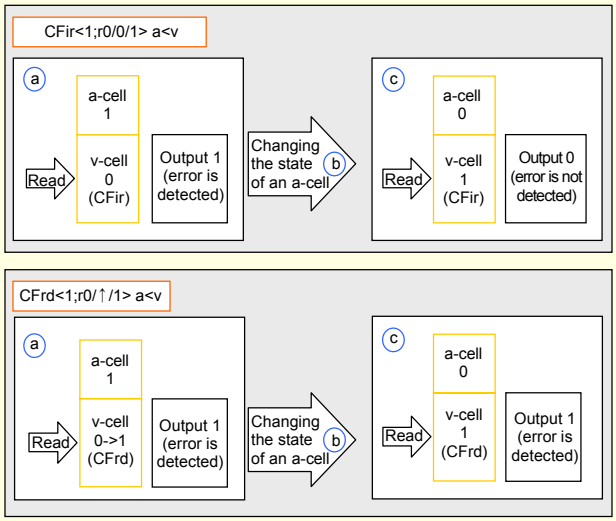


Fig. 5. Process which distinguishes CFir and CFrd.

Under the above conditions, the method to detect fault types is carried out as follows. First, defects are detected by applying diagnosis patterns to FPs of the fault group through just one port. If faults are detected, they are associated with a single port (1PFs), and if faults are not detected, they are marked as dual-port faults (2PFs). Second, faults are detected by applying diagnosis patterns only to the defective cells. If faults are detected, such faults are associated with one cell (1PF1 or 2PF1); otherwise, they are associated with both cells (1PF2 or 2PF2). Finally, some faults cannot be detected through the two methods mentioned above. These are all coupling faults. Coupling faults are those associated with two cells and can be distinguished by the locations of the aggressor and victim cells. Identification of such coupling faults can be accomplished based on the existence or the nonexistence of changes in cell values at the time of fault detection. Therefore, the detailed fault types are detected by applying corresponding diagnosis patterns to each fault group after identifying the appropriate fault group of each fault using fault dictionaries, which are in turn based on the test results. Table 5 lists the distinguishing features of each coupling fault.

For example, both CFir and the CFrd are sensitized by read operations at the victim cell and the specific state of the aggressor cell. Their test results are also the same. However, the data of the cell that has a CFrd fault changes, but that of the cell that has a CFir fault does not. Faults are generally distinguished according to the following processes:

ⓐ certifying an error caused by the read operation at the v-cell,
ⓑ changing the state of an a-cell, and
ⓒ recertifying the error of the v-cell.

Figure 5 depicts an example of the process which distinguishes

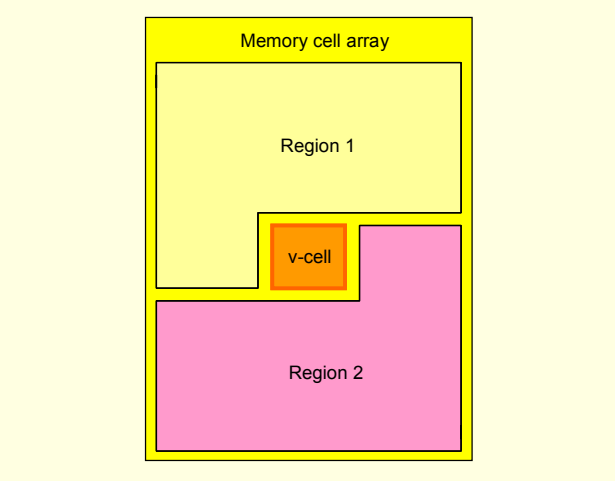


Fig. 6. Partitioning of the memory cell array.

CFir<1;r0/0/1> (a<v) and CFrd<1;r0/↑/1> (a<v). Process ⓐ is a read operation at a v-cell and the aggressor cell contains a 1 at the same time. Therefore, both CFir and CFrd faults are detected as a result of the read operation. At that time, the data of the v-cell is not changed by CFir, but the data of the v-cell is changed by CFrd. Then, the data of the aggressor cell is changed by process ⓑ. Process ⓒ, which is a read operation, is applied to the v-cell. Both CFir and CFrd are not sensitized by the operation, since the aggressor cell contains a 0. The faults are thus distinguished from each other by the three processes, since the result of CFir is a 0 but the result of CFrd is a 1.

The memory cell array can be divided into 3 regions as shown in Fig. 6: region 1, the faulty cell, and region 2. The faulty cell is a single cell. Region 1 includes cells that have a smaller address than that of the faulty cell, and region 2 includes cells that have a larger address than that of the faulty cell. Sometimes, different

Table 6. Fault group E.

Fault group	Fault number (FN)	Fault primitive
E	1	CFst <1; 1/0/-> a>v CFst <0; 1/0/-> a>v
	2	CFds <w1; 1/↓/-> a>v CFds <r1; 1/↓/-> a>v
	3	CFir <0; r1/1/0> a>v CFir <1; r1/1/0> a>v
	4	CFrr <0; r1/1/?> a>v CFrr <1; r1/1/?> a>v
	5	CFrd <0; r1/↓/0> a>v CFrd <1; r1/↓/0> a>v
	6	CFtr <0; w↑/0/-> a>v CFtr <1; w↑/0/-> a>v
	7	wCFds & wCFds <r0:r0; 1/↓/-> a>v wCFds & wCFds <r1:r1; 1/↓/-> a>v
	8	wCFrd & wRDF <0; r0:r0/↑/1> a>v wCFrd & wRDF <1; r1:r1/↓/0> a>v

Table 7. Diagnosis elements applied at each region of diagnosis pattern for group E.

	DE 1	DE 2	DE 3	DE 4	DE 5	DE 6	DE 7
Region 1	-	-	-	-	-	-	-
V-cell	-	w1, r1	-	r1	w1	r1	-
Region 2	w1	-	r1, w1	-	-	-	w0

	DE 8	DE 9	DE 10	DE 11	DE 12	DE 13
Region 1	-	-	-	-	-	-
V-cell	r0	-	r1	w1	r1	r1
Region 2	-	w1	-	w1	-	-

operations are applied to each region for diagnosis, or some operations are applied only to one or two regions.

For example, fault group E can be diagnosed using the features above. As shown in Table 6, fault group E, which has 18 fault primitives, is classified into 8 kinds of fault numbers (FN). They can be distinguished by the following 13 additional diagnosis elements (DEs).

[DE 1] w1 at region 2 through one port: for a-cell of FN 6

[DE 2] (w1, r1) at v-cell through one port: detection of FN 6

[DE 3] (r1, w1) at region 2 through one port: a-cell of FN 1, 2, 3, 4, and 5

[DE 4] r1 at v-cell through one port: detection of FN 1, 2, 3, 4, and 5

[DE 5] w1 at v-cell through one port: for v-cell of FN 1, 3, 4, and 5

[DE 6] r1 at v-cell through one port: detection of FN 1, 3, 4, and 5

[DE 7] w0 at region 2 through one port: for a-cell of FN 3

[DE 8] r0 at v-cell through one port: detection of FN 3

[DE 9] w1 at region 2 through one port: for a-cell of FN 5

[DE 10] r1 at v-cell through one port: detection of FN 5

[DE 11] w1 at v-cell and w1 at region 2 through one port: for a-cell of FN 1

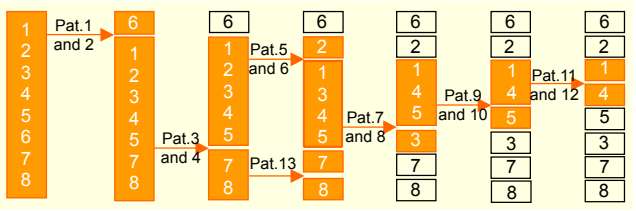


Fig. 7. Fault classifications by additional diagnosis pattern for fault group E.

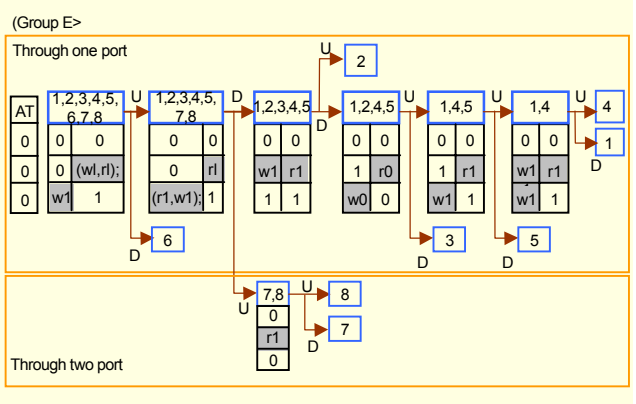


Fig. 8. Additional diagnosis process for fault group E.

[DE 12] r1 at v-cell through one port: detection of FN 1

[DE 13] r1 at v-cell through two ports: detection of FN 7

Table 7 shows a diagnosis pattern which is applied at the memory cell for group E. Each column indicates the involved diagnosis elements. The diagnosis elements are operations which are applied to the 3 regions. A blank space means that no operation occurs in the region. All operations are applied to the memory cell in increasing address order. Figure 7 indicates that the processes of the 8 fault primitives are distinguished by 13 patterns. For example, the fault primitives {1, 2, 3, 4, 5, 6, 7, 8} are classified into {1, 2, 3, 4, 5, 7, 8} and {6} by elements 1 and 2. Fault primitives {1, 2, 3, 4, 5, 7, 8} are classified into {1, 2, 3, 4, 5} and {7, 8} by elements 3 and 4. Then, fault primitives {1, 2, 3, 4, 5} are classified into {1, 3, 4, 5} and {2} by elements 5 and 6.

Figure 8 presents all the information about the additional diagnosis pattern, order of application, diagnosis procedure, and the maximum diagnosis pattern length for fault group E. The maximum diagnosis pattern length is 6N+8. In the figure, "AT" means the state of each of the 3 regions after the test procedure. The numbers 1 to 8 are the fault numbers, and the boxes marked with numbers are divided based on the diagnosis patterns below. Each diagnosis element in the diagnosis pattern consists of 3 boxes which list the operations applied to each of the 3 regions. Each column, which consists of 3 boxes, refers to a single diagnosis element. The arrows marked "U" indicate

Table 8. Classified fault results of fault diagnosis algorithm.

	Faults	Appropriate fault primitives (Fault groups - fault numbers)
1PF1s	SAF, TF, RDF, IRF, RRF	A-1, B-1
	DRDF	C-1, D-1
1PF2s	CFst	E-1, F-1, G-1, H-1, I-1
	CFds	E-2, F-2, H-2, I-2, J-1, K-1, L-1, M-1, N-1, O-1
	CFir	E-3, F-3, G-2, H-3, I-3
	CFrr	E-4, F-4, G-3, H-4, I-4
	CFrd	E-5, F-5, H-5, I-5
	CFdr	J-2, K-2, O-2, P-1
	CFtr	E-6, F-6, G-4, H-6
2PF1	wDRDF & wDRDF	A-2, B-2
	wRDF & wRDF	C-2, D-2
2PF2v	wCFds & wCFds	E-7, F-7, G-5, H-6, I-7
2PF2a	wCFdr & wDRDF	J-3, K-3, O-3, P-2
	wCFrd & wRDF	E-8, F-8, H-8, I-7
2PF2av	wCFds & wRDF	Q-1, R-1, S-1, T-1, U-1, V-1, W-1, X-1
	wCFds & wIRF	Q-2, R-2, S-2, T-2, U-2, V-2, W-2, X-2
	wCFds & wRRF	Q-3, R-3, S-3, T-3, U-3, V-3, W-3, X-3

Table 9. Performance comparison.

	[8]	[9]	[10]	[11]	[12]	March A2PF
Pattern length / Faults	17N	9N	17N	16N	Test pattern+ $19N+6M^2+3M+15$	24N+8
1PF1	D	D	D	D	D	D
1PF2	P	P	P	P	P	D
2PF1	I	I	I	I	I	D
2PF2a	I	I	I	I	I	D
2PF2v	I	I	I	I	I	D
2PF2av	I	I	I	I	D	D

D: distinguishable, I: indistinguishable, P: partially distinguishable

the faults that do not detect an error by the operations in the diagnosis element, and the arrows marked "D" indicate the faults that do detect an error.

VI. Performance Evaluation

The test patterns for tests and diagnoses that include all the fault models of dual-port memories must have the length of $18N$, while additional diagnosis pattern have the minimum length of 2 and the maximum length of $6N+8$. The use of as many as $24N+8$ patterns makes it feasible to determine fault types in more detail. Table 8 shows the results of fault identification by the fault models of the diagnosis algorithm. In addition, the fault group and fault number of each fault are displayed. They cannot be distinguished by any diagnosis pattern because the effects of the faults are identical.

The new algorithm is compared with existing fault diagnosis algorithms in Table 9. Only 1PFs fault models are taken into consideration in the diagnosis algorithms of [8]-[11]. After an examination of the length of each algorithm pattern, the types of faults are determined by comparing their responses from test patterns, specifically $17N$ ($12N+5N$), March-CW pattern+$9N$, $17N$, and $16N$ ($12N+4N$). Nevertheless, such determination takes only some of the faults, such as stuck-at faults and coupling faults, into consideration. The diagnosis algorithm of [12] is a fault diagnosis algorithm for dual-port memories. The pattern length of this algorithm, (not including the test) is $19N+6M^2+3M+15$ (where M is the number of columns). Such a length is even longer during execution of the test and diagnosis when the test algorithm is added to that length.

The algorithm proposed in this paper is capable of distinguishing various faults in greater detail than other algorithms, and can detect all faults associated with single- and dual-port fault models. Moreover, it is capable of both testing and diagnosis with patterns of the maximum length of $24N+8$ (test pattern $18N$ + maximum diagnosis pattern $6N+8$).

VII. Conclusion

This paper proposed a diagnosis algorithm that can effectively test dual-port memories and identify fault types through the diagnosis processes. This algorithm verifies the existence (or nonexistence) of defects by conducting tests with the March A2PF test pattern, which has length $18N$, and creates fault dictionaries based on such results. Next, such faults are classified into fault groups using the fault dictionary, and the types of faults are further analyzed by applying diagnosis patterns appropriate for each fault group. The diagnosis pattern length has been minimized with the use of a fault dictionary, which may be created using test results. The algorithm proposed here is highly efficient and takes all faults occurring in dual-port memories into consideration.

Appendix. Fault Dictionaries of 1PFs and 2PFs.

Tables A and B show the fault dictionary according to the March elements that detect the fault models (1PFs, 2PFs) in the dual-port memories using the March A2PF algorithm.

Table A. Fault dictionary of 1PFs.

Fault model			March element	Fault group
1PFs : 1PF1				
TF	<w↑/0/->		⑤, ⑥, ⑬, ⑭	A
	<w↓/1/->		②, ⑨, ⑩, ⑰, ⑱	B
RDF	<r0/↑/1>		②, ⑨, ⑩, ⑰, ⑱	B
	<r1/↓/0>		⑤, ⑥, ⑬, ⑭	A
IRF	<r0/0/1>		②, ⑨, ⑩, ⑰, ⑱	B
	<r1/1/0>		⑤, ⑥, ⑬, ⑭	A
RRF	<r0/0/?>		②, ⑨, ⑩, ⑰, ⑱	B
	<r1/1/?>		⑤, ⑥, ⑬, ⑭	A
1PFs : 1PF2				
CFds	<w0;0/↑/->	a>v	②, ⑨, ⑩	I
		a<v	⑰, ⑱	H
	<w0;1/↓/->	a>v	⑥	J
		a<v	⑭	K
	<r0;0/↑/->	a>v	②, ⑨, ⑩	I
		a<v	⑰, ⑱	H
	<r0;1/↓/->	a>v	⑤	L
		a<v	⑬	M
	<w1;0/↑/->	a>v	②	N
		a<v	⑩	O
	<w1;1/↓/->	a>v	⑤, ⑥	E
		a<v	⑬, ⑭	F
	<r1;0/↑/->	a>v	②	N
		a<v	⑩	O
	<r1;1/↓/->	a>v	⑤, ⑥	E
		a<v	⑬, ⑭	F
CFir	<0;r0/0/1>	a>v	②, ⑨, ⑩	I
		a<v	⑰, ⑱	H
	<0;r1/1/0>	a>v	⑤, ⑥	E
		a<v	⑬, ⑭	F
	<1;r0/0/1>	a>v	⑨, ⑩	G
		a<v	⑰, ⑱	H
	<1;r1/1/0>	a>v	⑤, ⑥	E
		a<v	⑬, ⑭	F
CFrr	<0;r0/0/ ?>	a>v	②, ⑨, ⑩	I
		a<v	⑰, ⑱	H
	<0;r1/1/ ?>	a>v	⑤, ⑥	E
		a<v	⑬, ⑭	F
	<1;r0/0/ ?>	a>v	⑨, ⑩	G
		a<v	⑰, ⑱	H
	<1;r1/1/ ?>	a>v	⑤, ⑥	E
		a<v	⑬, ⑭	F
CFrd	<0;r0/↑/1>	a>v	②, ⑨, ⑩	I
		a<v	⑰, ⑱	H
	<0;r1/↓/0>	a>v	⑤, ⑥	E
		a<v	⑬, ⑭	F
	<1;r0/↑/1>	a>v	②, ⑨, ⑩	I
		a<v	⑰, ⑱	H
	<1;r1/↓/0>	a>v	⑤, ⑥	E
		a<v	⑬, ⑭	F
CFtr	<0;w↓/1/->	a>v	⑨, ⑩	G
		a<v	⑰, ⑱	H
	<0;w↑/0/->	a>v	⑤, ⑥	E
		a<v	⑬, ⑭	F
	<1;w↓/1/->	a>v	⑨, ⑩	G
		a<v	⑰, ⑱	H
	<1;w↑/0/->	a>v	⑤, ⑥	E
		a<v	⑬, ⑭	F

Table B. Fault dictionary of 2PFs.

Fault model			March element	Fault group
2PFs : 2PF1				
wRDF&wRDF	<r0;r0/↑/1>		②,⑨,⑩,⑰, ⑱	B
	<r1:r1/↓/0>		⑤, ⑥, ⑬, ⑭	A
wDRDF&wDRDF	<r0;r0/↑/0>		⑩, ⑱	C
	<r1:r1/↓/1>		⑥, ⑭	D
2PFs : 2PF2a				
wCFds & wCFds	<r0:r0;0/↑/->	a>v	②, ⑨, ⑩	I
		a<v	⑰, ⑱	H
	<r0:r0;1/↓/->	a>v	⑤, ⑥	E
		a<v	⑬, ⑭	F
	<r1:r1;0/↑/->	a>v	⑨, ⑩	G
		a<v	⑰, ⑱	H
	<r1:r1;1/↓/->	a>v	⑤, ⑥	E
		a<v	⑬, ⑭	F
2PFs : 2PF2v				
wCFdr & wDRDF	<0;r0:r0/↑/0>	a>v	⑩	O
		a<v	⑱	P
	<0;r1:r1/↓/1>	a>v	⑥	J
		a<v	⑭	K
	<1;r0:r0/↑/0>	a>v	⑩	O
		a<v	⑱	P
	<1;r1:r1/↓/1>	a>v	⑥	J
		a<v	⑭	K
wCFrd & wRDF	<0;r1:r1/↓/0>	a>v	②, ⑨, ⑩	I
		a<v	⑰, ⑱	H
	<0;r0:r0/↑/1>	a>v	⑤, ⑥	E
		a<v	⑬, ⑭	F
	<1;r1:r1/↓/0>	a>v	②, ⑨, ⑩	I
		a<v	⑰, ⑱	H
	<1;r0:r0/↑/1>	a>v	⑤, ⑥	E
		a<v	⑬, ⑭	F
2PFs : 2PF2av				
wCFds & wRDF	<w0:r0/↑/1>	a>v	⑧	Q
		a<v	⑯	R
	<w0:r1/↓/0>	a>v	⑦	S
		a<v	⑮	T
	<w1:r0/↑/1>	a>v	③	U
		a<v	⑪	V
	<w1:r1/↓/0>	a>v	④	W
		a<v	⑫	X
wCFds & wIRF	<w0:r0/0/1>	a>v	⑧	Q
		a<v	⑯	R
	<w0:r1/1/0>	a>v	⑦	S
		a<v	⑮	T
	<w1:r0/0/1>	a>v	③	U
		a<v	⑪	V
	<w1:r1/1/0>	a>v	④	W
		a<v	⑫	X
wCFds & wRRF	<w0:r0/0/?>	a>v	⑧	Q
		a<v	⑯	R
	<w0:r1/1/?>	a>v	⑦	S
		a<v	⑮	T
	<w1:r0/0/?>	a>v	③	U
		a<v	⑪	V
	<w1:r1/1/?>	a>v	④	W
		a<v	⑫	X

References

[1] S.J. Wang and C.J. Wei, "Efficient Built-in Self-Test Algorithm for Memory," *Proc. of IEEE Asian Test Symposium*, 2000, pp. 66-70.

[2] S. Bahl and B. Singh, "A Novel Method for Silicon Configurable Test Flow and Algorithms for Testing, Debugging and Characterizing Different Types of Embedded Memories through a Shared Controller," *Proc. of IEEE Memory Technology, Design, and Testing*, 2004, pp. 78-83.

[3] W. Ji *et al.*, "Multi-port Memory Design Methodology Based on Block Read and Write," *Proc.s of IEEE Int'l Conference on Control and Automation*, 2007, pp. 256-259.

[4] C.F. Wu *et al.*, "Simulation-Based Test Algorithm Generation and Port Scheduling for Multi-port Memories," *Proc. IEEE Design Automation Conference*, 2001, pp. 301-306.

[5] A. Benso et al., "Automatic March Tests Generation for Multi-port SRAMs," *Proc. IEEE Electronic Design, Test, and Applications*, 2006, pp. 385-392.

[6] Y.K. Park et al., "An Efficiency Testing Algorithm for Realistic Faults in Dual Port Memories," *Trans. the Korean Institute of Electrical Engineers*, 2007, pp. 72-85.

[7] S. Hamdioui and A.J. Van de Goor, "Thorough Testing of Any Multiport Memory with Linear Tests," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 21, 2002, pp. 217-231.

[8] T.J. Bergfeld, D. Niggemeyer, and E.M. Rudnick, "Diagnostic Testing of Embedded Memories Using BIST," *Proc. IEEE Design Automation and Test in Europe Conference*, 2000, pp. 305-309.

[9] C.W. Wang *et al.*, "A Built-in Self-Test and Self-Diagnosis Scheme for Embedded SRAM," *Proc. IEEE the Ninth Asian Test Symposium*, 2000, pp. 45-50.

[10] J.F. Li et al., "March-Based RAM Diagnosis Algorithms for Stuck-at and Coupling Faults," *Proc. IEEE Int'l Test Conference*, 2001, pp.758 -767.

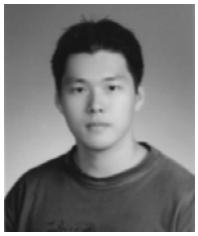
[11] V.A. Vardanian and Y. Zorian, "A March-Based Fault Location Algorithm for Static Random Access Memories," *Proc. IEEE Inte'l Workshop on Memory Technology, Design, and Testing*, 2002, pp. 10-12.

[12] H.W. Park and S.H. Kang, "A Diagnosis Algorithm for Dual-Port Memories," *Trans.* the *Korean Institute of Electrical Engineers*, 2001, pp. 192-200.

Youngkyu Park received the BS degree in electronic engineering from Hoseo University, Korea, in 2004, and the MS degree in electrical and electronic engineering from Yonsei University, Seoul, Korea, in 2007. He is currently working toward a PhD degree in electrical and electronic engineering at Yonsei University. His current research interests include VLSI design and testing, memory test algorithms, memory BIST, and design for testability.

Myung-Hoon Yang received the BS and MS degrees in electrical and electronic engineering from Yonsei University, Seoul, Korea, in 1996 and 1998, respectively. He was a research engineer with the System LSI Division, Samsung Electronics, Ltd. from 1998 to 2004. He is currently working toward a PhD degree in electrical and electronic engineering at Yonsei University. His current research interests include VLSI design, VLSI CAD, and VLSI testing and design for testability.

Yongjoon Kim received the BS and MS degrees in electrical and electronic engineering from Yonsei University, Seoul, Korea, in 2002 and 2004, respectively. He is currently working toward a PhD degree in electrical and electronic engineering at Yonsei University. His current research interests include SoC design, as well as testing and design for testability.



Dae-Yeal Lee received the BS and MS degrees in electrical and electronic engineering from Yonsei University, Seoul, Korea, in 2006 and 2008, respectively. His research interests include VLSI design, VLSI CAD, and VLSI testing and design for testability. He is currently working as a research engineer with the Memory Division of Samsung Electronics, Ltd.

Sungho Kang received the BS degree from Seoul National University, Seoul, Korea, and the MS and PhD degrees in electrical and computer engineering from University of Texas at Austin. He was a post-doctorial fellow at University of Texas at Austin; a research scientist at the Schlumberger Laboratory for Computer Science, Schlumberger Inc.; and a senior staff engineer at Semiconductor Systems Design Technology, Motorola Inc. Since 1994, he has been a professor with the Department of Electrical and Electronic Engineering, Yonsei University, Seoul, Korea. His current research interests include VLSI design, VLSI CAD, and VLSI testing and design for testability.