

論文93-30A-1-7

반도체 RAM의 결합고장을 검출하는 알고리즘

Algorithms for Detecting Coupling Faults
in Semiconductor RAM's

余定模*, 趙相福**

(Jeong Mo Yeo and Sang Bock Cho)

要約

기존의 알고리즘들이 완전히 검출하지 못하는 차수 2나 3의 연결된 2-결합고장을 검출하기 위하여 "알고리즘 다"가 제안되었다. 제한된 3-결합고장을 검출하는 "테스트 1*", "테스트 2*" 및 "알고리즘 라"가 제안되었다. "테스트 1*"는 제한된 3-결합고장을 검출하는 측면에서 기존의 알고리즘들보다 시간복잡도가 감소되었다. "테스트 2*" 및 "알고리즘 라"는 기존의 알고리즘들과 비교하여 시간복잡도가 감소되었고 개선된 고장 검출능력을 가진다. 그리고 요구하는 고장 검출정도에 따라 "알고리즘 라"를 순차적으로 수행시킬 수 있으며, 메모리를 병렬테스트하는 경우에 "알고리즘 라"를 수행시키면 시간복잡도가 상당히 개선된다. 비선형 번지순서를 발생시키고 두 번지부분으로 분할하여 반도체 RAM의 결합고장을 검출하는 MT(March Test)는 차수 3의 연결된 2-결합고장보다 복잡한 결합고장은 완전히 검출될 수 없다는 것이 입증되었다.

Abstract

"Algorithm DA" is proposed to test linked 2-CFs(2-Coupling Faults) with order 2 or 3 which are not perfectly detected in conventional algorithms. "Test 1*," "Test 2*," and "Algorithm RA" are proposed restricted 3-CFs. The time complexity of "Test 1*" is reduced in view of the detection of restricted 3-CFs. "Test 2*" and "Algorithm RA" have not only the reduced time complexity but also the improved fault coverage in comparison with conventional algorithms. And "Algorithm RA" can be applied step by step according to the degree of the fault coverage. If "Algorithm RA" is applied to the memory with parallel test, its time complexity is reduced considerably. It is proved that the MT(March Test) with nonlinear address sequences can not detect perfectly the CFs more complex than linked 2-CFs with order 3.

1. 서론

*正會員,釜山工科大学 電子計算學科

(Dept. of Computer Sci., Busan Nat.l Univ. of Tech.)

**正會員,蔚山大學校 電子工學科

(Dept. of Elec. Eng., Ulsan Univ.)

接受日字: 1992年7月 23日

(※이 연구는 '92년도 한국과학재단 연구비 지원에 의한 결과임. 과제번호: 921-0800-022-1)

반도체 RAM의 집적도가 증가하고 내부 동작전압이 낮아짐으로 인해 이전보다 고장 발생빈도가 높아졌을 뿐만 아니라 고장종류도 많아져서 고장검출이 복잡하고 어렵게 되어 고장 검출비용이 증가하였다. 따라서 테스트시간이 적고 우수한 고장 검출능력을 가지는 테스트를 요구하게 되었다.

반도체 RAM의 테스트는 DC 파라미터 테스트,

AC 파라미터 테스트 및 기능 테스트(functional test) 등으로 분류된다.⁷ 기능 테스트는 회로를 동작시키는데 영구적인 고장 즉 기능고장(functional faults)으로 논리레벨 측면에서 테스트하는 것을 말하며, 본 연구에서 취급하는 테스트이다. 기능고장에서 고착고장^{1,2,3}과 천이고장^{1,2,3}은 비교적 고장 모델이 간단하고 고장검출이 용이하다. 그러나 결합고장^{4,5,6,7,8}과 PSF(Pattern Sensitive Fault)^{7,8}는 고집적으로 인하여 많이 발생하고 그 고장모델이 복잡하여 고장검출 어려움의 정도가 비슷하다.⁸ 본 연구에서는 결합고장의 검출을 위한 테스트를 구한다.

Algorithm A¹는 번지해독기 고장과 판독/기록 회로(read/write logic)의 고장을 메모리셀 어레이의 고장으로 매핑시킴으로써 메모리셀 어레이의 고착고장과 2-결합고장만을 테스트하였다. Algorithm B⁴는 Algorithm A에서 검출하는 고장과 제한된 3-결합고장을 테스트하였다. 하지만 두 알고리즘에서는 역 결합고장(inversion coupling faults)과 연쇄 결합고장의 검출이 고려되지 않았다. Test A⁵는 고착고장, 천이고장, 2-결합고장 및 다중 호출고장(multiple access faults)의 네 고장 중에서 어느 하나의 고장이 존재하는 경우만 검출하는 최소의 MT(March Test)이고, Test B¹⁵는 다중 호출고장이 존재하지 않는 경우에 나머지 세 고장이 동시 존재하는 경우도 검출하였다. 하지만 Test A와 Test B에서도 역 결합고장과 연쇄 결합고장 그리고 3-결합고장이 고려되지 않았다. 36N test procedure⁶는 고착고장, 천이고장, 번지해독기 고장, 판독/기록 회로의 고장과 연결되지 않은 2-결합고장을 검출하는 테스트이고, 24Nlog₂N test procedure¹⁶는 2-결합고장과 제한된 3-결합고장까지 검출하는 테스트이다. 이 두 알고리즘에서도 역 결합고장과 연쇄 결합고장이 고려되지 않았다. A. J. Van de Goor는 역 결합고장을 고장모델로 분류⁷ 하였지만 이를 검출하는 알고리즘은 제시하지 않았다. 그리고 R. David는 역 결합고장까지 고려하여 테스트를 구성¹⁹ 하였지만, 이것은 마코브 체인 모델(Markov Chain Model)을 사용한 랜덤 테스트(random test)이었을 뿐만 아니라 연쇄 결합고장과 연결된 결합고장을 고려하지 않았다. "알고리즘 가"¹¹는 고착고장, 천이고장, 번지해독기 고장, 연결되지 않은 2-결합고장 및 연결되지 않은 제한된 연쇄 결합고장을 검출했다. 그리고 하나의 번지순서를 사용한 MT로서 검출할 수 있는 차수 2의 2-결합고장에는 한계가 있고,¹¹ "알고리즘 나"¹³는 그 한계까지 결합고장을 검출하였다. 이 두 알고리즘은 역 결합고장과 연쇄 결합고장을 고려하였으

나 차수 2의 연결된 2-결합고장을 완전히 검출하지 못하고 3-결합고장을 고려하지 않았다.

제 II 장에서는 본 연구에서 검출하고자 하는 반도체 RAM의 기능 고장모델을 설명하고, 제 III 장에서는 역 결합고장과 연쇄 결합고장까지 고려하여 기존의 알고리즘들에서 검출하지 못하는 차수 2나 3의 2-결합고장을 완전히 검출하는 테스트 알고리즘을 비선형 번지순서를 사용하여 구성하고 "알고리즘 다"로서 제안한다. 제 IV 장에서는 제한된 3-결합고장을 검출하는 "테스트 1*", "테스트 2*" 및 "알고리즘 라"를 구성하여 제안한다. 구성된 세 알고리즘 모두 역 결합고장과 연쇄 결합고장을 고려하였고, "테스트 2*"는 "테스트 1*"가 완전히 검출하지 못하는 연결되지 않은 2-결합고장을 검출하며, "알고리즘 라"는 "테스트 2*"가 완전히 검출하지 못하는 차수 2나 3의 연결된 2-결합고장을 검출한다. 그리고 구성된 세 알고리즘들의 3-결합고장 검출능력을 분석하고 기존의 알고리즘들과 비교하며, "알고리즘 라"의 순차적용과 병렬적용에 관하여 설명한다. 또한 비선형 번지순서 기법을 사용하는 MT가 검출하지 못하는 결합고장들이 있으므로 이를 규명한다.

II. 반도체 RAM의 기능 고장모델

본 연구에서 테스트 대상회로는 반도체 RAM이고, 테스트하고자 하는 고장은 반도체 RAM의 결합고장을 중심으로 한 기능고장이다.

테스트 대상회로인 반도체 RAM의 회로는 복잡하나 동작에 영향을 미치지 않도록 RAM 모델을 축소시켜 번지해독기, 메모리셀 어레이 및 판독/기록 회로로 구성되고,⁸ 테스트 메모리는 1 워드가 1 비트인 N 개의 워드를 가진다고 가정한다. 그리고 번지해독기는 고장이 발생하였을 때 순서회로(sequential circuit)가 되지 않는다고 가정한다.^{11,18}

반도체 RAM의 테스트 알고리즘을 구하기 위해서는 우선 반도체 RAM의 기능 고장모델이 설정되어야 한다. 기능고장은 비교적 고장모델이 간단한 단순 기능고장(simple functional faults)과 고장모델이 복잡한 복합 기능고장(compound functional faults)으로 분류할 수 있다.

1. 단순 기능고장

단순 기능고장 중에서 고착고장과 천이고장은 가장 많이 발생하는 고장으로서 테스트에 필수적으로 포함시켜야 하는 고장이다. 그리고 반도체 RAM이 고집적화됨에 따라 결합고장이나 PSF가 발생하게 되어 이들의 검출도 고려하게 되었다. 또한 결합고장에

역 결합고장도 포함시켜야 할 것이다.

1) 고착고장(SAF : Stuck-At Fault)¹⁾²⁾³⁾

반도체 RAM의 세 부분의 회로에서 발생할 수 있으며, 어떤 선(lines)이나 셀이 0이나 1의 값으로 고정되어 있는 고장을 말한다.

2) 천이고장(TF : Transition Fault)¹⁾²⁾³⁾

반도체 RAM의 메모리셀 어레이와 판독/기록 회로에서 발생하며, 한 셀의 내용이 천이(transition)될 수 없는 고장을 말한다.

3) 결합고장(CF : Coupling Fault)³⁾⁴⁾⁵⁾⁶⁾⁷⁾⁸⁾

반도체 RAM의 선이나 셀들 사이에서 발생하고, 한 셀의 내용변화가 다른 셀의 내용에 영향을 미치는 고장을 말한다.

(1) 2-결합고장(2-CF : 2-Coupling Fault)

i 셀(번호가 i임을 뜻함)이 결합셀(coupling cell)이고 k 셀이 결합된셀(coupled cell)인 2-결합고장을 C_{ik} 라 하고, 발생 가능한 C_{ik} 를 표 1에 나타내었다.

표 1. i 셀과 k 셀 사이에서 발생 가능한 C_{ik}
Table 1. Possible C_{ik} 's between i Cell and k Cell.

Fault models	Possible C_{ik} 's
1-way asymmetric CFs	$(\uparrow i \Rightarrow \uparrow k), (\uparrow i \Rightarrow \downarrow k),$ $(\downarrow i \Rightarrow \uparrow k), (\downarrow i \Rightarrow \downarrow k)$
2-way asymmetric CFs	$(\uparrow i \Rightarrow \uparrow k), (\uparrow i \Rightarrow \downarrow k, \downarrow i \Rightarrow \uparrow k)$ $(\downarrow i \Rightarrow \downarrow k), (\downarrow i \Rightarrow \uparrow k, \uparrow i \Rightarrow \downarrow k)$
1-way symmetric CFs	$(\uparrow i \Rightarrow \uparrow k), (\downarrow i \Rightarrow \downarrow k)$
2-way symmetric CFs	$(\uparrow i \Rightarrow \uparrow k, \downarrow i \Rightarrow \downarrow k)$
1-way inversion CFs	$(\uparrow i \Rightarrow \downarrow k), (\downarrow i \Rightarrow \uparrow k)$
2-way inversion CFs	$(\uparrow i \Rightarrow \downarrow k)$

표 1에서 $(\uparrow i \Rightarrow \downarrow k)$ 와 같은 표시는 i 셀이 0에서 1로 변화하면 k 셀이 1에서 0으로 변화하는 것을 뜻한다. 결합셀의 \uparrow 표시는 양방향고장을 뜻하며, 결합된셀의 \downarrow 표시는 항상 셀의 상태가 변화하는 역 결합고장을 뜻한다. 대칭 결합고장은 비대칭 결합고장에 포함되므로 발생 가능한 서로 다른 C_{ik} 는 11 가지이다.

(2) k-결합고장(k-CF : k-Coupling Fault)

k-결합고장은 제한된 k-결합고장(restricted k-coupling faults)과 일반적인 k-결합고장으로 분류된다. 일반적인 k-결합고장의 완전한 검출은 불가능 뿐만 아니라 고장 발생확률이 적으므로 제한된 k-결합고장을 많이 검출한다.⁴⁾⁶⁾

s 셀이 0(1) 상태에 있을 때, i 셀의 내용이 변화하면 k 셀의 내용을 변화시키는 3-결합고장을 $C_{ik}(0)$ ($C_{ik}(1)$)이라 표시하고, $C_{ik}(0)$ 또는 $C_{ik}(1)$ 을 $C_{ik}(s)$

라 표시하기로 한다. $C_{ik}(s)$ 에서 i 셀을 결합셀, k 셀을 결합된셀, s 셀을 정지셀(static cell)이라 한다. 임의의 k-결합고장의 표시도 3-결합고장의 표시와 같은 방법으로 표시될 수 있다.

3-결합고장보다 복잡한 k-결합고장은 고장모델이 복잡하여 엄청난 테스트시간을 요구하고 고장 발생빈도가 적으므로 테스트의 타당성이 적다. 따라서 본 연구에서는 k-결합고장의 검출을 2-결합고장과 3-결합고장의 검출에 한정시킨다.

4) PSF(Pattern Sensitive Fault)

메모리 셀 어레이 또는 데이터 레지스터에서 발생하며, 주위 셀의 상태 또는 주위 셀의 내용변화에 의하여 셀의 내용이 변화하는 고장이다.

2. 복합 기능고장

반도체 RAM이 더욱 고집적화됨에 따라 단순 기능 고장들은 서로 영향을 주고 받음으로써 복합 기능고장을 형성하고 테스트의 구성을 어렵게 한다. 본 연구에서는 복합 기능고장 중에서 결합고장들 사이에서 발생하는 복합 기능고장만을 고려한다.

1) 연쇄 결합고장(CCF : Chained Coupling Fault)

한 셀의 내용변화가 연쇄적으로 다른 셀들에게 영향을 주어 결합고장의 효과가 파급되는 고장을 말한다.¹³⁾ 임의의 판독동작 이전에 결합고장의 전과가 소멸되는 연쇄 결합고장을 제한된 연쇄 결합고장(restricted chained coupling fault)이라 하고, 제한을 주지 않은 연쇄 결합고장을 일반적인 연쇄 결합고장이라 하며, 연쇄 결합고장에서 원천 결합셀과 결합된셀들로서 구성되는 통로를 연쇄통로(chained path)라 한다.¹⁴⁾ 그리고 셀을 판독할 때 제한된 연쇄 결합고장의 고장전과는 완전히 소멸되어 결합된셀들은 어떤 고정된 상태에 있게 될 것이므로 제한된 연쇄 결합고장의 연쇄통로상의 모든 결합고장은 단방향 결합고장으로 매핑된다.

2) 연결된 결합고장(LCF : Linked Coupling Fault)¹⁵⁾

두 개 이상의 결합 셀이 하나의 결합된 셀을 공유하여 결합고장이 발생할 때, 결합 고장들이 연결되어 있다(link)고 하고, 그 때의 고장을 연결된 결합고장이라고 한다. 하나의 연결된 결합고장에서 연결되어 있는 결합고장의 수를 연결된 결합고장의 차수(the order of a LCF)¹⁵⁾라 한다. 차수 2의 연결된 2-결합고장이 연결된 결합고장 중에서 가장 많이 발생하며 가장 기본적이다. C_{ik} 와 C_k 가 연결되어 형성된 차수 2의 연결된 2-결합고장을 C_{ik} 라 표시하며, C_{ik} 와 C_k 가 각각 11 가지 있으므로 서로 다른 C_{ik} 는 121

가지이다.

연결되지 않은 k-결합고장은 제한된 k-결합고장을 포함한다. 즉 제한된 k-결합고장을 구성하는 셀들의 집합들은 서로 분리되어야 하며, 연결되지 않은 k-결합고장을 구성하는 셀들의 집합들은 결합된셀들을 제외하고는 서로 분리되지 않아도 되기 때문이다.

III. 연결된 2-결합고장을 검출하는 테스트 알고리즘

MT 기법은 결정 테스트(deterministic test)의 일종으로 그 구성이 쉬울 뿐만 아니라 구현하기가 용이하므로 기존의 알고리즘들^[4,5,6,8]에서 고장 검출기법으로 많이 사용되었다. 본 연구에서도 결합고장의 검출을 위하여 MT 기법을 사용한다. MT는 ME(March Element)들로 구성되며 ME의 표기는 "알고리즘 가"^[13]에서 사용한 표기와 동일하게 한다.

하나의 번지순서를 사용한 MT가 검출할 수 있는 차수 2의 연결된 2-결합고장의 한계가 있다.^[13] 그 이유는 결합셀들이 연속적으로 결합된셀에 영향을 주어 고장효과를 소멸시키기 때문이다. 따라서 이런 고장들을 검출하기 위해서는 고장효과가 소멸되지 않도록 하여야 한다.

방법 a) MT가 다수의 번지순서에 의해 수행되도록 함으로써 결합셀들이 연속적으로 결합된셀에 영향을 주지 못하도록 하는 경우가 생기게 하여 고장효과가 소멸되지 않도록 한다.

방법 b) ME가 수행되는 번지가 변함에 따라 서로 다른 ME가 수행되도록 하여 고장효과가 소멸되지 않도록 한다.

방법 b는 번지가 달라짐에 따라 ME가 달라져서 MT의 성질에 위배된다. 따라서 방법 b에 의해 구성된 테스트는 MT가 아닐 뿐만 아니라 최적의 테스트를 구하기가 힘들다. 만약 테스트를 구한다 하더라도 이를 구현하기가 어렵다. 그러므로 방법 a를 이용하여 고장을 검출하는 테스트를 구한다.

다수의 번지순서를 발생시키는 방법에는 비선형 번지순서 기법이 있고, 이것은 기존의 알고리즘들^[4,6]에서 제한된 3-결합고장을 검출하기 위하여 사용되었다. 이 비선형 번지순서 기법을 연결된 2-결합고장의 검출에 이용할 것이다.

1. 비선형 번지순서의 발생과 성질

우선 비선형 번지순서를 정의하고 그 성질을 규명한다.

정의 1) 최하위 번지부터 최상위 번지까지의 순서로 구성된 번지순서를 선형 번지순서(linear address sequence) S_0 라 하고, S_0 의 각 번지를 k 비트씩 왼쪽 회전시켜 구성한 번지순서를 비선형 번지순서(nonlinear address sequence) S_k 라 한다.

정의 1의 선형 번지순서는 번지가 증가되는 방향이고, 선형 번지순서를 역으로 발생시키면 감소되는 선형 번지순서가 된다. 감소되는 선형 번지순서에서 다른 비선형 번지순서를 발생시킬 수 있다. 감소되는 선형 번지순서와 이 순서에서 발생된 비선형 번지순서의 성질은 증가되는 선형 번지순서와 이 순서에서 발생된 비선형 번지순서의 성질과 동일하므로 따로 고려하지 않는다.

정리 1) n-비트 번지로 구성되는 번지순서에는 1개의 선형 번지순서와 (n-1)개의 비선형 번지순서가 있다.

증명) 정의 1에 의하여 n-비트 번지로 구성되는 기본 번지순서는 1개 존재한다. S_0 를 n 비트 왼쪽 회전시키면 S_0 가 되기 때문에 n-비트 번지를 왼쪽 회전시킬 수 있는 방법은 (n-1)가지 있다. 따라서 비선형 번지순서를 발생시키는 방법은 (n-1)가지가 있다.

정의 2) 비선형 번지순서 S_k 의 모든 번지를 발생된 순서대로 배열하고 동일한 갯수로서 두 부분(partition)으로 나누었을 때, 먼저 발생된 번지부분을 S_k 의 상위 번지부분(top address partition)이라 하고, 나중 발생된 번지부분을 S_k 의 하위 번지부분(bottom address partition)이라 한다.

예를 들어 3-비트 번지의 기본 번지순서와 비선형 번지순서들을 그림 1에 나타내었다.

Partitions	S_0	S_1	S_2
Top address partition	0	0	0
	1	2	4
	2	4	1
Bottom address partition	3	6	5
	4	1	2
	5	3	6
	6	5	3
	7	7	7

그림 1. 3-비트 번지의 번지순서
Fig. 1. Address sequences of 3-bit address.

정리 2) k 번째 비트가 서로 다른 n 비트의 두 수는 S_{k+1} 상에서 서로 다른 번지부분에 있다. 여기서 k 는 0 또는 n 보다 적은 양의 정수이다.

증명) S_0 (또는 S_n)의 상위 번지부분과 하위 번지부분의 번지는 $(n-1)$ 번째 비트(MSB)에서 서로 다르다. 그리고 S_i 은 S_0 를 1 비트 왼쪽으로 회전시킨 것이므로 0 번째 비트부분(LSB)이 다르다. S_k 는 S_0 를 왼쪽으로 k 비트 회전시킨 것이므로 $(k-1)$ 번째 비트가 다르다. 따라서 k 번째 비트가 다른 두 수는 S_{k+1} 에서 서로 다른 번지부분에 있다. 4-비트 번지의 번지순서들의 비트 값은 그림 2와 같이 된다. ◆

Partitions	S_0	S_1	S_2	S_3
Top	0XXX	XXX0	XX0X	X0XX
Bottom	1XXX	XXX1	XX1X	X1XX

그림 2. 4-비트 번지에서 번지순서들의 비트 값
Fig. 2. Bit values of address sequences in 4-bit address.

보조정리) 서로 다른 두 번지는 적어도 한 개 이상의 번지순서에서 서로 다른 번지부분에 있다.

증명) 서로 다른 두 번지를 2진수로 표현하면 적어도 1 비트 이상이 서로 다르다. k 번째 비트가 서로 다르다면 정리 2에 의하여 S_{k+1} 상에서 서로 다른 번지부분에 있다. 따라서 서로 다른 두 번지는 적어도 한 개 이상의 번지순서에서 서로 다른 번지순서에 있다. ◆

정리 2를 역으로 말하면, k 번째 비트가 서로 같은 n -비트의 두 수는 S_{k+1} 상에서 서로 같은 번지부분에 있다. 즉 1 비트 이상이 동일한 n -비트의 두 수는 적어도 한 개 이상의 번지순서에서 동일한 번지부분에 있다. 만약 임의의 두 수가 동일한 번지부분에 있지 않을 조건은 두 수의 모든 비트가 달라야 한다. 모든 비트가 다른 두 번지는 가장 낮은 번지와 가장 높은 번지이다.

2. 연결된 2-결합고장의 검출

두 개의 2-결합고장, C_{ik} 와 C_{jk} 가 서로 연결되어 차수 2의 연결된 2-결합고장인 C_{jk} 가 형성되었다고 가정하자. 하나의 번지순서만을 사용한 MT가 검출할 수 없는 C_{ik} 는 모두 16 가지인데, ^[13] 이 고장들 모두 두 결합셀의 번지가 연속되어 오고 두 결합고장 중에서 하나 이상이 역 결합고장이다. 따라서 두 결합셀이 연속되지 않도록 셀순서를 재배치하여 테스트하는 경우에 대해 생각한다. 여기서 셀순서라 함은 ME를 수행시키는 셀의 순서를 의미한다. 셀순서를 재배치

하기 위해서는 하나의 번지순서로서는 불가능하므로 비선형 번지순서를 발생시켜 사용한다.

1) 연결된 2-결합고장을 검출하는 테스트 알고리즘의 구성

“알고리즘 가”의 각 ME를 선형 번지순서의 한 번지부분에서 수행시킨 후, 선형 번지순서의 다른 번지부분을 판독하는 ME를 수행시킨다. 이와 같은 과정을 모든 비선형 번지순서에서 반복 수행시킨 것이 “알고리즘 다”이다. “알고리즘 가-1”을 사용하면 “알고리즘 다-1”을 구성할 수 있고, “알고리즘 가-2”를 사용하면 “알고리즘 다-2”를 구성할 수 있다. “알고리즘 다-1” 또는 “알고리즘 다-2”를 “알고리즘 다”라 한다. 다음은 “알고리즘 다-1”을 나타낸 것이다.

“알고리즘 다-1”

단계 1 : 모든 셀을 x 값으로 기록하여 초기화한다.
단계 2 : 다음 MT를 선형 번지순서에서 수행시킨다.

단계 3 : 비선형 번지순서를 발생시켜 각 번지순서에서 단계 2를 반복 수행시킨다.

정리 3) “알고리즘 다”는 고착고장, 천이고장, 번지해독기 고장, 연결되지 않은 2-결합고장, 연결되지 않은 제한된 연쇄 결합고장 및 4보다 적은 차수의 연결된 2-결합고장을 검출한다.

증명) a) “알고리즘 다”는 “알고리즘 가”를 포함하므로 고착고장, 천이고장, 번지해독기 고장, 연결되지 않은 2-결합고장 및 연결되지 않은 제한된 연쇄 결합고장을 검출한다.

b) 차수 2의 연결된 2-결합고장의 검출을 증명한다. i 셀, j 셀 및 k 셀로 형성되는 C_{ik} 가 형성되고, i 셀과 j 셀을 서로 다른 번지부분에 있게 하는 번지순서를 S_k 라 가정하자. 정리 2의 보조정리에 의해 S_k 는 적어도 하나 이상 존재한다. k 셀은 S_k 의 어느 한 번지부분에서 i 셀(또는 j 셀)과 함께 존재하고, j 셀(또는 i 셀)은 다른 번지부분에 존재하게 될 것이므로 j 셀(i 셀)이 k 셀에 영향을 주어 k 셀이 판독될 때 C_{ik} 가 검출된다.

c) 차수 3의 2-결합고장의 검출을 증명한다. i, j 및 l 셀이 결합셀이고 k 셀이 결합된 셀인 차수 3의 연결된 2-결합고장, C_{ijk} 가 발생하였다고 가정하자. 결합셀들 중에서 임의의 두 셀을 서로 다른 번지부분에 있게 하는 하나의 번지순서가 정리 2의 보조정리에 의해 하나 이상 존재한다. 예를 들어 i 셀과 j 셀

이 서로 분리되어 위치될 수 있는 방법에는 그림 3과 같이 네 가지 경우가 있다. ①(④)의 경우, 한 번지부분에서 i 셀(j 셀)의 내용이 변화하여 k 셀의 내용을 변화시키면 다른 번지부분의 k 셀을 판독할 때 C_k 가 검출된다. ②(③)의 경우, $C_k(C_{ik})$ 가 연결되지 않은 2-결함고장의 형태가 되므로 C_{ik} 가 검출된다. 따라서 차수 3의 연결된 2-결함 고장이 검출된다. ◆

Cases	One partition	The other partition
①	i	j, l, k
②	i, l	j, k
③	i, k	j, l
④	i, l, k	j

그림 3. i 셀과 j 셀이 서로 다른 번지부분에 있을 때, i, j, l 및 k 셀이 위치할 수 있는 모든 경우

Fig. 3. Possible locations of i, j, l, and k cell when both i and j are located separately.

“알고리즘 다”는 정리 3의 고장 이외에도 많은 고장을 검출한다. 제한된 연쇄 연결결함고장은 소속된 모든 결함셀들이 다른 고장과 연결되어 있어야 할 뿐만 아니라 소속되는 모든 연결고장이 고장효과가 항상 소멸되어야 검출되지 않는다. 따라서 이와 같은 경우는 극히 적고 “알고리즘 다”는 모든 비선형 번지순서에서 수행되므로 제한된 연쇄 연결결함고장이 거의 모두 검출된다. 또한 “알고리즘 다”가 모든 비선형 번지순서에서 수행됨에 따라 차수 4 이상의 연결된 2-결함고장이나 제한된 3-결함고장의 많은 부분이 검출될 수 있다. “알고리즘 다”의 시간복잡도는 $12.5N \log_2 N + N$ 이다.

2) 시뮬레이션

“알고리즘 다”가 “알고리즘 가”를 포함하고 있으므로 고착고장, 천이고장, 번지해독기 고장, 연결되지 않은 2-결함고장 및 연결되지 않은 제한된 연쇄 결함고장의 검출은 명백하다. “알고리즘 다”는 차수 2나 3의 연결된 2-결함고장을 연결되지 않은 2-결함고장으로 변화하여 연결되지 않은 2-결함고장을 검출함으로써 이들 연결고장을 검출한다. 따라서 차수 2나 3의 연결된 2-결함고장에 대한 시뮬레이션은 “알고리즘 다”에서 이와 같은 변화과정이 잘 이루어지는가를 확인함으로써 충분하다. SUN 4 SPARK 시스템에서 프로그램으로써 이를 확인하였다.

기존의 알고리즘들¹¹⁾은 차수 2나 3의 연결된 2-결함고장을 완전히 검출하지 못한다. “알고리즘 다”는 이들 고장을 완전히 검출할 뿐만 아니라 연쇄 결

함고장을 고려하였으므로 고장 검출능력이 증가하였다. 하지만 “알고리즘 다”는 비선형 번지순서 기법을 사용하였으므로 시간복잡도가 증가하였다.

IV. 제한된 3-결함고장을 검출하는 테스트 알고리즘

고집적 반도체 RAM에서 3-결함고장이 발생할 수 있다. 일반적인 3-결함고장의 완전한 검출은 엄청난 테스트시간이 필요하므로 거의 불가능할 뿐만 아니라 고장 발생정도가 적으므로 제한된 3-결함고장을 검출한다.¹²⁾

1. 제한된 3-결함고장을 검출하는 테스트 알고리즘의 구성

기존의 알고리즘들¹¹⁾은 비선형 번지순서 기법을 사용하여 제한된 3-결함고장을 검출하였다. 본 연구에서도 제한된 3-결함고장을 검출하는 테스트 알고리즘을 비선형 번지순서 기법을 사용하여 구성하기로 한다.

1) 테스트 1*

하나의 번지순서를 두 개로 분할하여 하나의 번지부분에 s 셀이 있고 다른 하나의 번지부분에 C_{ik} 가 형성되어 구성된 $C_{ik}(s)$ 를 검출하도록 한다.

한 번지부분의 s 셀을 0과 1의 각각의 상태에 두었을 때, 다른 번지부분에서 “알고리즘 가”를 수행시킨다. 그리고 이와 같은 과정을 모든 번지순서에서 수행시킨 테스트를 “테스트 1*”라 한다. “테스트 1*”는 여러가지로 구성될 수 있다. 다음은 “알고리즘 가-1”을 이용한 하나의 “테스트 1*”를 나타낸다. 각 ME 앞의 t는 상위 번지부분에서 ME를 수행시키는 것을 뜻하고, b는 하위 번지부분에서 ME를 수행시키는 것을 뜻한다.

“테스트 1*”의 예

단계 1 : 모든 셀을 x 값으로 기록하여 초기화한다.

단계 2 : 다음 MT를 선형 번지순서에 수행시킨다.

$$\overline{tRxWx'}; \overline{tRx'Wx}; \overline{tRxWx'}; \overline{tRx'Wx}; tR; \overline{bRxWx'}; \overline{bRx'Wx}; \overline{bRxWx'}; \overline{bRx'Wx}; bRx;$$

단계 3 : 모든 셀을 x' 값으로 기록하여 다시 초기화한다.

단계 4 : 다음 MT를 선형 번지순서에서 수행시킨다.

$$\overline{tRx'Wx}; \overline{tRxWx'}; \overline{tRx'Wx}; \overline{tRxWx'}; tR; \overline{bRx'Wx}; \overline{bRxWx'}; \overline{bRx'Wx}; \overline{bRxWx'}; bRx;$$

단계 5 : 비선형 번지순서를 발생시켜 각 번지순서에서 단계 2부터 4까지를 반복 수행시킨다. 단 하나의 번지순서에서의 수행이 끝나고 다음 번지순서를 수행시킬 때는 x를 x'로 바꾸고 x'를 x로 바꾼다.

정리 4) "테스트 1*"는 고착고장, 천이고장 및 제한된 3-결합고장을 검출한다.

증명) "테스트 1*"는 모든 셀에 0과 1을 기록하고 판독하므로 고착고장을 검출하고, 모든 셀을 상태변화시키고 판독하므로 천이고장을 검출한다. 어떤 번지부분에 s 셀이 있고 다른 번지부분에서 C_{ik}가 발생하여 3-결합고장인 C_{ik}(s)가 형성되었다고 가정하자. s 셀이 0과 1의 각각의 상태에 있을 때, 다른 번지부분에 형성되는 C_{ik}는 그 번지부분에서 수행되는 "알고리즘 가"에 의해 검출된다. 즉 C_{ik}(s)가 검출된다. C_{ik}(s)를 형성시켰을 때, 대부분의 C_{ik}(s)는 수행되는 번지순서에서 발생되어 검출되지만 일부의 C_{ik}(s)는 검출되지 않는다. 검출되지 않는 C_{ik}(s)들을 분리되는 셀들의 집합속에 포함시키지 않으면, 분리되는 집합들로 구성되는 C_{ik}(s)들은 모두가 검출되므로 제한된 3-결합고장이 검출된다. ◆

"테스트 1*"가 수행되는 모든 번지부분에서 형성되는 연결되지 않은 2-결합고장은 "알고리즘 가"가 수행됨으로 인해 검출된다. 하지만 수행되는 어떠한 번지부분에서도 형성되지 않는 연결되지 않은 2-결합고장이 있다. 예를 들어 임의의 두 번지가 동일한 번지부분에 존재하지 않을 경우는 정리 2의 역의 경우로서 두 번지의 모든 비트가 다른 경우이다. 즉 가장 낮은 번지의 셀과 가장 높은 번지의 셀로 구성되는 2-결합고장은 수행되는 어떠한 번지부분에서도 구성되지 않아 검출되지 않는다. 따라서 "테스트 1*"는 연결되지 않은 2-결합고장을 완전히 검출하지 못한다. 하지만 이와 같은 경우는 오로지 한 가지뿐이므로 "테스트 1*"는 연결되지 않은 2-결합고장을 거의 모두 검출한다. "테스트 1*"의 시간복잡도는 19Nlog₂N+N이다.

2) 테스트 2*

"테스트 1*"는 연결되지 않은 2-결합고장을 완전히 검출하지 못한다. 따라서 제한된 3-결합고장을 검출할 뿐만 아니라 연결되지 않은 2-결합고장을 완전히 검출하도록 테스트 알고리즘을 구성한다.

"테스트 1*"의 MEs를 재배치하여 제한된 3-결합고장을 검출하도록 하고, 연결되지 않은 2-결합고장을 검출할 수 있도록 재구성한다. 상위(또는 하위) 번

지부분 다음에 하위(또는 상위) 번지부분이 오도록 구성된 번지순서에 "알고리즘 가"를 적용시키고, 다시 초기화하여 하위(또는 상위) 번지부분 다음에 상위(또는 하위) 번지부분이 오도록 구성된 번지순서에 "알고리즘 가"를 수행시킨다. 이와 같은 과정을 모든 번지순서에 수행시킨 테스트를 "테스트 2*"라 한다. "테스트 2*"는 여러가지로 구성될 수 있으며, 다음은 "알고리즘 가-1"을 이용한 하나의 "테스트 2*"를 나타낸다.

"테스트 2*"의 예

단계 1 : 모든 셀을 x 값으로 기록하여 초기화한다.
단계 2 : 다음 MT를 선형 번지순서에 수행시킨다.

$$\overline{tRxWx'}; \overline{bRxWx'}; \overline{tRx'Wx}; \overline{bRx'Wx};$$

$$\overline{bRxWx'}; \overline{tRxWx'}; \overline{bRx'Wx}; \overline{tRx'Wx}; Rx;$$

단계 3 : 모든 셀을 x' 값으로 기록하여 다시 초기화한다.

단계 4 : 다음 MT를 선형 번지순서에서 수행시킨다.

$$\overline{bRx'Wx}; \overline{tRx'Wx}; \overline{bRxWx'}; \overline{tRxWx'};$$

$$\overline{tRx'Wx}; \overline{bRx'Wx}; \overline{tRxWx'}; \overline{bRxWx'}; Rx;$$

단계 5 : 비선형 번지순서를 발생시켜 각 번지순서에서 단계 2부터 4까지를 반복 수행시킨다. 단 한 번지순서의 수행이 끝나고 다음 번지순서를 수행시킬 때는 x를 x'로 바꾸고 x'를 x로 바꾼다.

정리 5) "테스트 2*"는 고착고장, 천이고장, 번지해독기 고장, 연결되지 않은 2-결합고장, 연결되지 않은 제한된 연쇄 결합고장 및 제한된 3-결합고장을 검출한다.

증명) a) "테스트 2*"는 "알고리즘 가"를 포함하므로 고착고장, 천이고장, 번지해독기 고장, 연결되지 않은 2-결합고장 및 연결되지 않은 제한된 연쇄 결합고장을 검출한다

b) 제한된 3-결합고장은 3-결합고장을 형성하는 셀들의 집합들이 서로 분리되어 있다. 즉 어떤 한 셀이 제한된 3-결합고장으로 인하여 내용이 변하고 나면, 다른 셀의 내용이 변한다 하더라도 그 셀의 내용은 변하지 않아 고장효과가 소멸되지 않으므로 "테스트 1*"와 "테스트 2*"에 있는 각 ME의 수행은 서로 독립적으로 작용한다. 그리고 "테스트 2*"는 "테스트 1*"의 모든 MEs를 포함하고, 각 ME가 수행될 때 다른 번지부분의 상태가 서로 동일하므로 정리 4에 의해 "테스트 2*"는 제한된 3-결합고장을 검출한다. ◆

즉 조합(combination)을 나타낸다. C_{ik} 와 s 셀이 서로 다른 번지부분에 존재하여 형성된 $C_{ik}(s)$ 를 $P_{ik}(s)$ 라 하자. $P_{ik}(s)$ 가 구성된 세 알고리즘들에서 형성될 수 있는 방법의 수, M_T 는 (2) 식과 같다. 여기서 K 는 16이며 알고리즘의 구성방법에 따라 달라진다.

$$M_T = (N/2)C_2 * (N/2) * 2 * n * K \quad (2)$$

M_T 에서 서로 다른 $P_{ik}(s)$ 의 수를 M_R 이라 하면, M_T 는 M_R 보다 훨씬 크다.

정의 3) 다수의 번지순서를 사용한 M_T 로써 k -결합고장을 검출할 때, 발생 가능한 모든 k -결합고장의 수(M_T)에 대한 M_T 가 검출하는 모든 k -결합고장의 수(M_R)의 비를 k -결합고장의 형식패턴비(formal pattern ratio of k -CF)라 하고, M_T 에 대한 M_T 가 검출하는 서로 다른 k -결합고장의 수(M_R)의 비를 k -결합고장의 실패턴비(real pattern ratio of k -CF)라 한다. ◆

k -결합고장의 형식패턴비는 k -결합고장의 검출을 위해 k -결합고장을 중복형성시키는 정도를 나타내고, k -결합고장의 실패턴비는 k -결합고장의 검출에 필요한 k -결합고장의 형성정도를 나타낸다. 형식패턴비가 크면 클수록 k -결합고장을 검출하기 위한 패턴을 많은 부분에서 발생시킨다. 실패턴비는 1을 초과할 수 없고, 1이면 연결되지 않은 k -결합고장을 모두 검출하게 된다. 따라서 형식패턴비는 적을수록 실패턴비는 1에 가까울수록 효과적인 테스트 알고리즘이다.

구성된 세 알고리즘들에서 M_T 는 (1) 식과 같고, M_T 는 (2) 식과 같으므로 3-결합고장의 형식패턴비 R_F 는 (3) 식으로 표현되고, 3-결합고장의 실패턴비 R_R 은 (4) 식으로 표현된다.

$$R_F = M_T/M_T = n * N / (8 * (N - 1)) * K \quad (3)$$

$$R_R = M_R/M_T \quad (4)$$

정리 7) 구성된 세 알고리즘들에서 메모리용량이 커질수록 3-결합고장의 실패턴비는 1에 접근하나 결코 1이 될 수 없다.

증명) 구성된 세 알고리즘들에서 3-결합고장의 R_F 와 R_R 의 값을 SUN 4 SPARK 시스템에서 구하여 표 2에 나타내었다. 표 2에서 메모리용량이 증가하면 3-결합고장의 실패턴비가 1에 접근하는 것을 알 수 있다. 그리고 가장 낮은 번지의 i 셀과 가장 높은 번지의 k 셀이 C_{ik} 를 형성하고 다른 임의의 s 셀이 정지셀이 되어 3-결합고장인 $C_{ik}(s)$ 가 구성되었다고 가정하자. 임의의 s 셀이 0과 1의 각 상태에서 C_{ik} 의

모든 경우가 구성될 수 있어야 $C_{ik}(s)$ 를 모두 검출할 수 있다. 하지만 C_{ik} 의 모든 경우가 발생하도록 M_T 가 구성될 수는 있으나 i 셀이나 k 셀이 결코 같은 번지부분에 있을 수 없으므로 C_{ik} 의 각 경우가 임의의 s 셀이 0과 1 상태에서 구성되는 것이 보장되지 않는다. 따라서 $C_{ik}(s)$ 가 완전히 검출되지 않아 구성된 세 알고리즘들에서 3-결합고장의 실패턴비는 결코 1이 될 수 없다.

표 2에서는 C_{ik} 가 하나의 번지부분에서 구성된다고 가정하였지만 두 번지부분에서 C_{ik} 가 구성되어 형성되는 많은 $C_{ik}(s)$ 가 구성된 세 알고리즘들에서 검출된다. 따라서 표 2의 R_R 값은 실제로 더욱 커질 것이므로 메모리용량이 커질수록 R_R 은 빠른 속도로 1에 접근한다.

보조정리) 메모리용량이 클수록 구성된 세 알고리즘들이 연결되지 않은 3-결합고장을 검출할 수 있는 능력은 증가한다.

증명) 정리 7에 의하여 메모리용량이 커질수록 3-결합고장의 실패턴비가 1에 접근하므로 구성된 세 알고리즘들이 연결되지 않은 3-결합고장을 검출하는 정도는 증가한다. ◆

표 2. 구성된 세 알고리즘들에서 각 비트수에 따른 R_F 와 R_R 의 값

Table 2. Values of R_F and R_R for Each of the bits on Three Algorithms Proposed

No. of bits	R_F	R_R
3	0.857142857	0.71428572
4	1.066666666	0.76190475
5	1.290322580	0.80645161
6	1.523809523	0.84587813
7	1.763779527	0.87926509
8	2.007843137	0.90659256
9	2.254403131	0.92839736
10	2.502443792	0.94547903
11	2.751343429	0.95868068
12	3.000732600	0.99951136
13	3.250396776	0.99987789
14	3.500213636	0.99998474
15	3.750114444	0.99999809

16-비트(64k) 이상의 번지를 가진 메모리에 대한 3-결합고장의 실패턴비는 0.999999 이상이므로 구성된 세 알고리즘들은 64k 이상의 메모리에 있는 연결되지 않은 3-결합고장을 모두 검출한다고 할 수 있다.

Algorithm B¹와 $24N \log_2 N$ test procedure⁶도 비선형 번지순서 기법을 사용하여 테스트 알고리즘들이 구성되었으므로 본 연구에서 구성된 세 알고

리듬들과 같은 정도로 연결되지 않은 3-결합고장을 검출한다. 하지만 이 알고리즘들은 연결되지 않은 3-결합고장을 고려하지 않았을 뿐만 아니라 고장 검출 능력을 분석하지 않았다.

(2) 연결된 3-결합고장의 고장 검출능력

어떤 고집적 반도체 RAM을 구성된 세 알고리즘들로서 테스트할 때, 3-결합고장의 실패턴비가 1에 가까우므로 실패턴비를 1이라 가정하여 연결되지 않은 3-결합고장이 모두 검출된다고 하자. 그리고 어떤 3-결합고장인 $C_{ik}(s)$ 가 다른 2-결합고장과 연결되어 차수 2나 3의 연결된 결합고장이 발생하였다고 가정하자. "알고리즘 라"에서는 s 셀의 0과 1 상태에서 모든 경우의 C_{ik} 를 발생시키고 연결되지 않은 3-결합고장이 검출된다고 하였으므로 $C_{ik}(s)$ 는 2-결합고장으로 작용한다. 따라서 가정된 고장을 검출하기 위해서는 차수 2나 3의 연결된 2-결합고장을 검출하면 된다. "알고리즘 라"는 차수 2나 3의 연결된 2-결합고장을 검출하므로 가정된 고장을 검출한다. 하지만 "테스트 1*"나 "테스트 2*" 또는 기존의 알고리즘들¹⁾은 차수 2나 3의 연결된 2-결합고장을 검출하지 못하여 가정된 고장을 검출하지 못한다.

어떤 3-결합고장이 2-결합고장들과 연결되어 차수 4 이상의 결합고장을 형성하였을 때는 기존의 알고리즘들¹⁾과 구성된 세 알고리즘들에서 완전히 검출되지 못한다. 또한 어떤 3-결합고장이 다른 3-결합고장과 연결된 고장도 완전히 검출되지 아니한다. 하지만 이들 고장은 고장확율이 적을 뿐만 아니라 엄청난 테스트비용으로 인하여 반도체 RAM의 테스트에서 무시하여도 좋을 것이다.

5) 제한된 3-결합고장을 검출하는 알고리즘들의 비교

"테스트 1*"가 고착고장, 천이고장 및 제한된 3-결합고장의 검출은 정리 4에 의하여 명백하다. "테스트 2*"는 "알고리즘 가"를 포함하고 있어 고착고장, 천이고장, 번지해독기 고장(AF's : Address decoder Faults), 연결되지 않은 2-결합고장 및 연결되지 않은 연쇄 결합고장을 검출하고, 그리고 "테스트 1*"의 각 MEs를 포함하고 있어 제한된 3-결합고장을 검출하는 것이 명백하다. "알고리즘 라"는 "알고리즘 다"를 포함하고 있어 "알고리즘 다"가 검출하는 모든 고장을 검출하고, 그리고 "테스트 2*"를 포함하고 있어 제한된 3-결합고장을 검출하는 것이 명백하다. 따라서 "테스트 1*", "테스트 2*" 및 "알고리즘 라"의 고장검출에 관한 시뮬레이션은 필요하지 않다.

표 3에 제한된 3-결합고장을 검출하는 알고리즘들의 시간복잡도와 고장 검출능력을 나타내었다.

"테스트 1*"는 Algorithm B⁶⁾보다 2-결합고장을 검출하지 못하지만 제한된 3-결합고장을 검출하는 측

면에서 시간복잡도가 감소되었고, 24Nlog₂N test procedure⁶⁾보다 고장 검출능력이 증가하였으며, 시간복잡도가 감소되었다. 그리고 "테스트 1*"는 기존의 알고리즘들에서 고려되지 않았던 역 결합고장을 고려하였을 뿐만 아니라 대용량 RAM의 연결되지 않은 3-결합고장을 검출한다.

"테스트 2*"는 Algorithm B보다 연결된 2-결합고장을 적게 검출하나 천이고장과 연결되지 않은 연쇄 결합고장을 검출하고 시간복잡도가 감소되었다. 그런데 Algorithm B에서 역 결합고장을 고려하지 않았으므로 "테스트 2*"에서도 역 결합고장을 고려하지 않는다면 "테스트 2*"는 Algorithm B와 같은 수준으로 연결된 2-결합고장을 검출한다. 그리고 "테스트 2*"는 24Nlog₂N test procedure보다 고장 검출능력이 증가하였을 뿐만 아니라 시간복잡도가 감소하였다. 또한 "테스트 2*"는 기존의 알고리즘들¹⁾에서 고려하지 않았던 역 결합고장 및 연쇄 결합고장을 고려하였으며, 대용량 RAM의 연결되지 않은 3-결합고장을 검출한다. "테스트 2*"는 동일한 시간복잡도로서 "테스트 1*"보다 고장 검출능력이 우수하다.

표 3. 3-결합고장을 검출하는 알고리즘들의 시간복잡도 및 고장 검출능력
Table 3. Time Complexity and Fault Coverage of Algorithms for Detecting 3-CFs.

Algorithms	Time complexity	Fault coverage
Algorithm B ⁶⁾	32Nlog ₂ N+N	SAFs, AFs, 2-CFs, & Restricted 3-CFs (not including inversion CFs & CCFs)
24Nlog ₂ N test procedure ¹⁾	24Nlog ₂ N+N	Restricted 3-CFs (not including inversion CFs & CCFs)
TEST 1*	19Nlog ₂ N+2N	SAFs, AFs, TFs, Restricted 3-CFs, & Unlinked 3-CFs of large memories
TEST 2*	19Nlog ₂ N+2N	SAFs, AFs, TFs, Unlinked 2-CFs, Unlinked Restricted CCFs, Restricted 3-CFs, & Unlinked 3-CFs of large memories
Algorithm RA	21.5Nlog ₂ N+2N	SAFs, TFs, Unlinked 2-CFs, Unlinked Restricted CCFs, 1 linked 2-CFs with order less than 4, Restricted 3-CFs, Unlinked 3-CFs of large memories, & Linked faults of large memories when a 3-CF is linked with 1 or 2 2-CFs

"알고리즘 라"는 Algorithm B에서 고려하지 않았던 천이고장과 연결되지 않은 제한된 연쇄 결합고장을 검출한다. 그리고 Algorithm B는 역 결합고장까지 포함한 차수 2나 3의 연결된 2-결합고장을 완전히 검출하지 못하지만 "알고리즘 라"는 이를 검출한다. 따라서 "알고리즘 라"는 Algorithm B보다 고장 검출능력이 증가하였다. 그리고 시간복잡도도 Algorithm B보다 감소하여 개선되었다.

"알고리즘 라"는 24Nlog₂N test procedure보다 고장 검출능력이 증가하였고, 시간복잡도가 감소되었다. 또한 "알고리즘 라"는 대용량 RAM의 연결되지

않은 3-결합고장을 검출하고, 대용량 RAM에서 3-결합고장이 하나 또는 두 개의 2-결합고장과 연결된 결합고장도 검출한다. "알고리즘 라"는 "테스트 1*"나 "테스트 2*"보다 시간복잡도는 다소 증가하였으나 고장 검출능력은 월등히 증가하였다.

2. "알고리즘 라"의 순차적용 및 병렬적용

표 4. "알고리즘 라"의 단계적 적용
Table 4. Step by Step Application of "Algorithm RA".

Steps applied	Time complexity	Fault coverage
to step 2	13.5N	SAFs, AFs, TFs, Unlinked 2-CFs, & Unlinked CCFs
to step 3	12.5Nlog2N+Ni	all Above faults & Linked 2-CFs with order less than 4
all	21.5Nlog2N+2N	all Above faults & Restricted 3-CFs

"알고리즘 라"는 "알고리즘 가" 및 "알고리즘 다"를 포함하고 있으므로 요구되는 고장 검출정도에 따라 단계별로 수행시킬 수 있다. 표 4는 "알고리즘 라"를 단계별로 수행시킨 표이다.

"알고리즘 라"의 단계 2까지 수행시키면 반도체 RAM에서 가장 많이 발생하는 고착고장, 친이고장, 번지해독기 고장, 연결되지 않은 2-결합고장 및 연결되지 않은 연쇄 결합고장 등을 13.5N의 시간복잡도로써 검출할 수 있고, 단계 3까지 수행시키면 4보다 적은 차수의 연결된 2-결합고장까지 검출할 수 있으며, 모두 수행시키면 제한된 3-결합고장까지 수행시킬 수 있다. 따라서 반도체 RAM의 집적도나 공정도의 신뢰성 또는 생산성 등에 따라 "알고리즘 라"를 단계별로 수행시킬 수 있다.

대용량의 메모리는 대부분 분할되어 병렬테스트되고 있다. 메모리를 m 개씩 q(N = mq) 개로 분할하여 "알고리즘 라"를 수행시키는 경우를 생각해 보자. 이때 "알고리즘 라"의 시간복잡도는 21.5Nlog2N+2N에서 21.5mlog2m+2m으로 감소한다.

표 5는 메모리용량에 따라 메모리의 제한된 3-결합고장을 검출하는 알고리즘들의 시간복잡도를 나타낸 것이다.

표 5에서 보면 "알고리즘 라"가 기존의 알고리즘들보다 시간복잡도가 개선되었을 뿐만 아니라 병렬테스트하게 되면 더욱 개선되는 것을 알 수 있다. 64 Mbits의 메모리를 Algorithm B로 테스트하면 833N의 시간복잡도를 요구하고, "알고리즘 라"로서 테스트하면 561N의 시간복잡도를 요구하는 반면, 메모리를 64 등분하여 병렬테스트하도록 해서 "알고리즘 라"를 수행시키면 6.75N의 시간복잡도를 요구할

뿐이다.

표 5. 제한된 3-결합고장을 검출하는 알고리즘들의 시간복잡도

Table 5. Time Complexity of Algorithms for Detecting Restricted 3-CFs.

Algorithms	Time complexity(The no. of partitions)		
	4 Mbits	16 Mbits	64 Mbits
Algorithm B***	705N(1)	769N(1)	833N(1)
24Nlog2N test procedure**	565N(1)	613N(1)	661N(1)
Algorithm RA	475N(1)	518N(1)	561N(1)
Algorithm RA(parallel test)	24.3N(16)	27N(16)	6.75N(64)

3. 비선형 번지순서 기법을 사용한 MT의 결합고장 검출한계

비선형 번지순서 기법을 사용하여 MT를 구성하더라도 결코 검출할 수 없는 결합고장이 있다. 이를 규명하고자 한다.

정리 8) 비선형 번지순서 기법을 사용한 MT는 차수 3의 연결된 2-결합고장보다 복잡한 결합고장은 완전히 검출할 수 없다.

증명) a) 차수 4의 연결된 2-결합고장의 검출에 대하여 생각한다. 연결되어 있는 결합고장들이 서로 고장효과를 소멸시키면 해당고장은 검출되지 아니한다. 예를 들면 3-비트 번지의 메모리에서 0 번지 셀이 결합된셀이고 1, 2, 5, 6 번지 셀들이 결합셀들이며, 연결된 고장 모두가 양방향 역 결합고장인 차수 4의 연결된 2-결합고장은 검출되지 아니한다. 따라서 차수 4의 연결된 2-결합고장은 완전히 검출되지 아니한다.

b) a)에 의하여 차수 4의 연결된 2-결합고장이 완전히 검출되지 않으므로 차수 4 이상의 연결된 2-결합고장은 완전히 검출되지 아니한다.

c) 정리 7에 의하여 연결되지 않은 3-결합고장이 완전히 검출되지 아니하므로 이보다 복잡한 3-결합고장은 완전히 검출되지 아니한다. ◆

정리 8에 의하여 비선형 번지순서 기법을 사용한 MT로써 결합고장을 검출하려고 하는 경우에 차수 3의 연결된 2-결합고장보다 복잡한 결합고장을 완전히 검출할 수 없다. 따라서 이를 완전히 검출하기 위해서는 더 많은 번지부분으로 나누든지 다른 방법을 사용해야 할 것이므로 이들을 검출하는 테스트의 시간복잡도는 크게 증가될 것으로 예측된다

V. 결론

기존의 결합고장을 검출하는 결정 테스트들

(deterministic tests)은 역 결합고장과 연쇄 결합고장을 고려하지 않았다. 본 연구에서는 이들을 고려하여 테스트를 구하였다.

"알고리즘 다"는 고착고장, 천이고장, 번지해독기 고장 및 연결되지 않은 2-결합고장을 검출할 뿐만 아니라 기존의 알고리즘들에서 고려되지 않았던 연결되지 않은 제한된 연쇄 결합고장과 차수 2나 3의 연결된 2-결합고장을 완전히 검출한다. 이보다 복잡한 고장들도 많은 부분에서 검출한다. "알고리즘 다"의 시간복잡도는 $12.5N\log_2 N + N$ 이다.

제한된 3-결합고장까지 검출하는 "테스트 1*", "테스트 2*" 및 "알고리즘 라"를 구성하였다. "테스트 1*"의 시간복잡도는 $19N\log_2 N + N$ 이고, 고착고장, 천이고장 및 제한된 3-결합고장을 검출하며, 거의 모든 연결되지 않은 2-결합고장과 고집적 RAM의 연결되지 않은 3-결합고장을 검출한다. 제한된 3-결합고장을 검출하는 측면에서 "테스트 1*"는 기존의 알고리즘들보다 개선되었다. "테스트 2*"는 "테스트 1*"가 검출하는 고장, 번지해독기 고장, 연결되지 않은 2-결합고장 및 연결되지 않은 제한된 연쇄 결합고장을 검출하므로 "테스트 1*"와 기존의 알고리즘들보다 고장 검출능력이 증가하였고, 시간복잡도가 $19N\log_2 N + N$ 로서 기존의 알고리즘들보다 개선되었다. "알고리즘 라"는 "테스트 2*"가 검출하는 고장과 4보다 적은 차수의 연결된 2-결합고장을 검출한다. 그리고 고집적 RAM의 연결되지 않은 3-결합고장을 거의 모두 검출하고, 하나의 3-결합고장이 하나 또는 두 개의 2-결합고장과 연결되어 있는 경우의 고장도 검출한다. "알고리즘 라"는 기존의 알고리즘들보다 고장 검출능력이 증가하였고, 시간복잡도가 $21.5N\log_2 N + 2N$ 으로서 기존의 알고리즘들보다 개선되었다. 또한 "알고리즘 라"는 "테스트 1*"나 "테스트 2*"보다 시간복잡도가 약간 증가하였지만 고장 검출능력이 월등히 증가하였다.

반도체 RAM의 집적도나 신뢰성 또는 생산성에 따라 요구하는 고장 검출능력이 달라지는 경우 "알고리즘 라"를 단계적으로 수행시킬 수 있다. 또한 메모리를 분할하여 병렬테스트하는 경우에 "알고리즘 라"를 수행시키면 시간복잡도가 상당히 개선되는 것을 알 수 있다. 예를 들어 64 Mb의 메모리를 64 등분하여 병렬테스트하는 경우에 "알고리즘 라"를 수행시키면 $6.75N$ 의 시간복잡도를 가진다.

비선형 번지순서 기법을 사용한 MT는 차수 4 이상의 연결된 2-결합고장보다 복잡한 결합고장은 완전히 검출할 수 없다는 것을 본 연구에서 입증하였다.

参 考 文 献

- [1] J. Knaizuk, JR., and C. R. P. Hartmann, "An Algorithm for Testing Random Access Memories," *IEEE Trans. on Computers*, vol. C-26, pp. 414-416, Apr. 1977.
- [2] J. Knaizuk, JR., and C. R. P. Hartmann, "An Optimal Algorithm for testing Stuck-at Faults in Random Access Memories," *IEEE Trans. on Computers*, vol. C-26, no. 11, pp. 1141-1144, Nov. 1977.
- [3] R. Nair, "Comments on 'An Optimal Algorithm for Testing Stuck-at Faults in Random Access Memories,'" *IEEE Trans. on Computers*, vol. C-28, no. 3, pp. 258-261, Mar. 1979.
- [4] R. Nair, S. M. Thatte, and J. A. Abraham, "Efficient Algorithm for Testing Semiconductor Random Access Memories," *IEEE Trans. on Computers*, vol. C-27, no. 6, pp. 572-576, June 1978.
- [5] D. S. Suk and S. M. Reddy, "A March Test for Functional Faults in Semiconductor Random Access Memories," *IEEE Trans. on Computers*, vol. C-30, no. 12, pp. 982-985, Dec. 1981.
- [6] C.A. Papachristou, and N. B. Sahgal, "An Improved Method for Detecting Functional Faults in Semiconductor Random Access Memories," *IEEE Trans. on Computers*, vol. C-34, no. 2, pp. 110-116, Feb. 1985.
- [7] M. S. Abadir and H. K. Reghbaty, "Functional Testing of Semiconductor Random Access Memories," *ACM Computing Surveys*, vol. 15, no. 3, pp. 175-198, Sept. 1983.
- [8] A. J. Van De Goor and C. A. Verruijt, "An Overview of Deterministic Functional RAM chip Testing," *ACM Computing Surveys*, vol. 22, no. 1, pp. 5-33, Mar. 1990.
- [9] R. David, A. Fuentes, and B. Courtois, "Random Pattern Testing Versus Deterministic Testing of RAM."

IEEE Trans. on Computers, vol. 38, no. 5, pp. 637-650, May 1989.

- [10] J. Savir, W. H. McAnney, and S. R. Vecchio, "Testing for Coupled Cells in Random-Access Memories," *IEEE Trans. on Computers*, vol. 40, no. 10, pp. 1177-1180, Oct. 1991.
- [11] D. S. Suk, "Functional and Pattern Sensitive Fault Testing Algorithms for semiconductor Random access Memories," ph. D. dissertation, Univ. of Iowa, July 1978.
- [12] J. Savir, W. H. McAnney, and S. R. Vecchio, "Testing for coupled cells in random-access memories," in *proc. 1989. Int. Test Conf.*, pp. 439-451, Aug. 1989.
- [13] J. M. Yeo and S. B. Cho, "The limit of the march test method and algorithms (on detecting the coupling faults of semiconductor memories)," *KITE*, vol. 29, no. 8, pp. 99-109, Aug. 1992.

— 著 者 紹 介 —

余 定 模(正會員) 第29卷 A編 第8號 參照
 현재 부산공업대학 전자계산학과
 교수.

趙 相 福(正會員) 第29卷 A編 第8號 參照
 현재 울산대학교 전자공학과 교수