

論文2002-39SD-3-3

이중 포트 메모리를 위한 고장 진단 알고리즘

(Fault Diagnosis Algorithm for Dual Port Memories)

朴漢源*, 姜成昊*

(Hanwon Park and Sungho Kang)

요 약

현재 다양한 분야에서 이중 포트 메모리의 사용이 증가함에 따라서 이중 포트 메모리의 고장을 진단하기 위한 효율적인 고장 진단 알고리즘의 필요성이 증대되고 있다. 따라서 본 논문에서는 이중 포트 메모리에서의 효율적인 고장 진단 알고리즘을 제시하여 이중 포트 메모리에서 발생하는 거의 모든 종류의 고장에 대한 진단을 가능하게 한다. 또한 진단 과정에서 착오를 일으키지 않고 다양한 고장 모델을 구별하며 고장과 관련된 위치를 정확하게 확인하는 것이 가능하다. 새로운 진단 알고리즘을 사용함으로써 이중 포트 메모리에서의 고장 진단 과정은 효과적으로 수행될 수 있으며 이전의 다른 연구들과의 성능 평가를 통해 효율성을 확인할 수 있다.

Abstract

As dual port RAMs are widely used in the various applications, the need for an efficient algorithm to diagnose faults in dual port RAMs is increased. In this paper we propose an efficient algorithm that can diagnose all kinds of faults in dual port RAMs. In addition, the new algorithm can distinguish various fault models and locate the position related to each fault. Using the new algorithm, fault diagnosis for dual port RAMs can be performed efficiently and the performance evaluation with previous approaches proves the efficiency of the new algorithm.

Key words : *dual port memory, fault model, fault diagnosis.*

I. 서 론

메모리의 용량과 집적도의 급속한 증가로 인해 필연적으로 메모리 생산 기술 뿐 아니라 테스트 관련 기술의 개발에 대한 필요성이 대두되었다. 특히 메모리 수리 과정을 통해 재생산을 하기 위해서는 테스트를 통해 검출된 고장을 분석하고 고장의 종류와 위치를 명확하게 결정할 수 있는 고장 진단 방법의 개발이 필수적이다. 하지만 지금까지의 연구는 대부분 메모리를 테

스트하기 위한 방법들을 다루어 왔다^[1-5].

또한 현재 다양한 분야에서 널리 사용되고 있는 이중 포트 메모리에 대한 고장 모델과 테스트 방법에 대한 연구는 어느 정도 발표된 바가 있다^[6-8]. 하지만 이중 포트 메모리에서의 고장 진단 방법에 관한 연구는 전혀 이루어지지 않아 이에 관한 연구의 필요성이 절실한 상황이다. 따라서 본 논문에서는 이중 포트 메모리에서의 효율적인 고장 진단 방법을 제안한다.

지금까지 메모리에서의 고장 진단에 대한 기존의 방법들은 다음과 같은 범위 내에서 이루어져 왔다. 우선^[9]에서는 메모리에서의 결합 고장(coupling fault : CF)과 고착 고장(stuck at fault : SAF)의 위치를 확인하기 위한 진단 과정이 제안되었으나 다양한 고장 모델을 고려하지 못하여 실제로 메모리에서 고장이 발생할 경우 이를 진단하기가 어렵다. [10]에서는 메모리에서

* 正會員, 延世大學校 電氣電子工學科

(Dept. of Electrical and Electronic Eng., Yonsei Univ.)

※ 본 연구는 2001년도 시스템집적반도체기반기술개발사업의 과제 연구비에 의하여 연구되었음.

接受日字:2001年5月28日, 수정완료일:2002年2月7日

고장을 진단하기 위한 마치 테스트(march test)가 제안되었으나 새로운 고장 모델에 근거해 고장을 진단하는 방법이 사용되었기 때문에 메모리 테스트에서 일반적으로 사용되는 기능적 고장 모델에 대한 분석과 진단이 이루어지지 않는다. [1]에서는 메모리에서 진단 과정을 수행할 수 있는 내장된 자체 테스트(Built-in self test : BIST)구조가 제안되었다. 하지만 이 경우 진단 과정에서 고려하는 고장 모델들에 대한 분석이 이루어지지 않고 있으며 BIST에서 스캔(scan)을 이용하여 단지 고장이 검출되는 셀의 위치만을 확인할 수 있다. 또 [12]에서는 메모리 테스트 알고리즘에서 각각의 고장 모델에 따라 알고리즘에서의 고장 검출 여부에 대한 결과에 대한 표를 작성하고 이를 실제의 테스트 결과와 비교하여 메모리의 셀에서 고장이 존재할 경우 고장의 종류를 분석한다. 이 경우 다양한 종류의 고장들에 대한 분석이 가능하지만 결합 고장의 경우 피결합 셀(coupled cell)의 위치만을 확인하고 결합원셀(coupling cell)의 위치를 확인하지 못하므로 고려하는 고장 모델에 관련된 모든 셀을 진단하지 못한다. [13]에서는 제안된 마치 테스트를 초기에 사용하여 단일 셀 고장(single cell fault)과 다중 셀 고장(multiple cell fault)를 구별하고 결합원셀을 찾기 위한 패턴을 사용하여 결합원셀의 확인이 가능하지만 결합원셀을 찾기 위한 패턴을 가하는 어드레스 시퀀스가 고장 모델에 관계없이 일정하므로 비효율적인 단점이 있으며 NPSF에 대한 진단은 이루어지지 않는다. 그런데 이러한 기존의 연구들은 모두 단일 포트 메모리에서의 고장들에 대한 진단 방법에 국한되어 이중 포트 메모리에서의 고장 진단 방법은 제안되지 않았다. 그러므로 이중 포트 메모리의 구조와 동작을 고려한 고장 모델들에 근거한 고장 진단이 불가능하다는 문제점이 있다.

따라서 본 논문에서는 이중 포트 메모리에서 일반적으로 사용되는 고장 모델들에 근거하여 테스트에서 고장이 검출될 경우 그 고장의 종류와 고장과 관련된 모든 셀의 위치를 확인할 수 있는 진단 방법을 제안한다. 이는 각각의 고장 모델들을 위해 새로운 진단 패턴을 제안하고 기존의 테스트 알고리즘과 통합함으로써 가능하게 하였다. 이러한 이중 포트 메모리에 대한 진단 방법에 근거하여 단일 포트 메모리에 대한 진단 방법도 분석한다. 또 일반적인 진단 알고리즘뿐 아니라 테스트 알고리즘의 수행 결과를 이용한 진단 방법을 제시하여 테스트 정보의 분석을 이용한 진단 방법의 확

보를 가능하게 한다.

본 논문은 다음과 같이 구성되어 있다. II장에서는 진단 과정에서 고려하는 고장 모델들에 대해 설명한다. III장에서는 고려하는 고장 모델들의 진단을 위한 패턴들을 분석한다. IV장에서는 이중 포트 메모리에서의 고장을 고려한 진단 알고리즘을 제안하고 이에 따라 단일 포트 메모리에서의 진단 방법 및 테스트 정보의 분석을 이용한 진단 방법을 제안한다. V장에서는 기존의 제안된 메모리 진단 방법들과 본 논문에서 제안한 진단 방법들을 비교한 성능 평가 비교 및 결과를 보여준다. 그리고 VI장은 결론으로써, 본 논문의 연구 내용을 요약한다.

II. 고장 모델

메모리 테스트를 위해서는 발생할 수 있는 물리적 결함을 모델링하여 고장 모델의 정의하고 테스트를 통해 이들을 검출할 수 있어야 한다. 따라서 테스트에서 고장이 검출될 경우 효율적인 고장 진단을 위해서는 고려해야 할 고장 모델을 결정하여야 한다. 이중 포트 메모리는 두 개의 포트를 통해 메모리에 접근하여 데이터를 입출력할 수 있는 구조이므로 일반적인 단일 포트 메모리에서의 고장 뿐만 아니라 새로운 고장 모델이 필요하다. 본 연구에서는 이중 포트 메모리에서의 고장을 단일 포트 관련 고장과 이중 포트 관련 고장으로 나누어 다음과 같은 고장 모델들을 고려하기로 한다.

1. 단일 포트 관련 고장(1PF)

(1) 고착 고장, 천이 고장

고착 고장은 하나 또는 그 이상의 메모리 셀이나 선이 고착되어서 값이 0 혹은 1로 항상 고정되어 있는 상태의 고장을 말한다. 따라서 이러한 고장을 검출하려면 메모리 셀에서 0과 1의 값을 읽는 동작을 수행하여야 한다. 그리고 천이 고장은 고착 고장의 특수한 경우라고 할 수 있는데 셀이 0에서 1로 혹은 1에서 0으로의 천이 쓰기 동작을 수행하지 못하고 실패하는 고장을 말한다. 그러므로 이 고장을 검출하기 위해서는 0에서 1로의 천이 쓰기 동작과 1에서 0으로의 천이 쓰기 동작을 수행한 후 각각의 값들을 읽는 동작을 수행해야 한다.

(2) 결합고장

두 개 또는 그 이상의 셀들 사이에서의 비정상적인 연결로 인해 발생하는 고장으로 한 셀에서의 동작이

다른 셀에 영향을 끼쳐 오동작을 발생시키는 고장을 말한다. 하나의 셀에서 천이 쓰기 동작을 할 때 결합된 두 번째 셀의 내용이 강제로 0 또는 1로 바뀌는 형태의 고장이다. 4가지의 다른 형태의 고장이 존재할 수 있다. 즉 셀 i 가 피결합셀, j 가 결합원셀이라고 할 때, 셀 $\langle j ; i \rangle$ 에서 $\langle \uparrow;0 \rangle$, $\langle \uparrow;1 \rangle$, $\langle \downarrow;0 \rangle$ 그리고 $\langle \downarrow;1 \rangle$ 이 존재할 수 있다. 예를 들어 $\langle \uparrow;0 \rangle$ 의 경우 결합원셀 j 에 0에서 1로 상승 천이 쓰기 동작을 수행할 경우 피결합셀 i 의 값이 강제로 0으로 바뀌게 되는 고장을 의미한다. 따라서 이 고장은 고착 고장이나 천이 고장과 달리 테스트에서 고장이 검출되는 셀 외에도 고장과 관련된 셀이 존재하게 된다. 따라서 진단 과정에서 고장이 검출되는 셀 뿐만 아니라 고장과 관련된 셀 모두를 확인하는 과정이 필요하다.

(3) NPSF

셀의 내용이 메모리 배열 내의 다른 셀들의 내용에 따라 영향을 받는 것으로 과도한 누설 전류나 커패시터 결합 노이즈(capacitively coupled noise)등에 의해서 발생한다. 이러한 패턴 감응 고장의 원인들은 메모리가 고집적 됨에 따라 셀간의 간격이 좁아져 더욱 빈번히 발생하게 되어 대용량 메모리의 큰 문제가 되고 있다. 이러한 NPSF는 기준셀 주위의 이웃한 여러 셀들의 특정 패턴이나 천이 쓰기 동작이 기준셀에 영향을 끼쳐 오동작을 일으키는 형태로 나타난다. 이러한 고장의 경우 여러 가지로 세분화될 수 있는데 각각 이웃셀들의 특정 패턴에서의 고착 고장, 천이 고장, 결합 고장과 유사한 고장의 특성을 갖는다.

(4) 어드레스 디코더 고장

어드레스 디코더 고장은 주소 디코더에서 발생하는 고장을 나타낸다. 이들 고장은 크게 다음과 같이 분류될 수 있다.

- 고장 1 : 특정한 주소를 가지고 어떤 메모리 셀도 접근할 수 없는 고장
- 고장 2 : 특정한 셀이 접근 불가능한 고장
- 고장 3 : 특정한 주소가 여러 개의 셀을 동시에 접근하는 고장
- 고장 4 : 특정한 셀이 여러 개의 주소로 접근이 가능한 고장

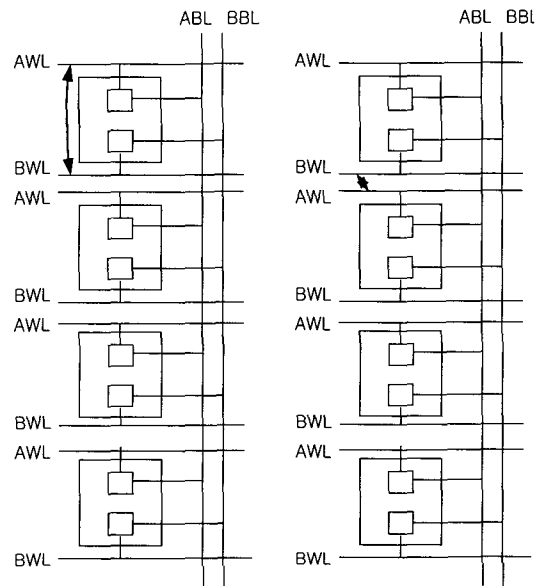
2. 이중 포트 관련 고장(2PF)

(1) Inter port fault

이중 포트 메모리는 두 개의 포트 A, B를 통해 독립

적으로 메모리 셀 어레이에 접근하여 데이터를 동시에 입출력하는 것이 가능하다. 이를 위해 포트 A와 포트 B를 위한 워드 라인이 별도로 존재하게 된다. 따라서 이러한 이중 포트 메모리의 구조상 서로 다른 포트 A와 B의 워드 라인들간의 단락이나 커패시턴스 결합 등의 고장이 존재할 수 있는데, 이를 inter-port fault라 한다^[14]. 이는 다음과 같은 형태로 매핑(mapping)될 수 있다.

- ◆ 같은 행 내의 OR-type(AND-type) 워드 라인 단락
 - ◆ 다른 행 내의 OR-type(AND-type) 워드 라인 단락
- 이러한 서로 다른 포트들의 워드 라인들간에 존재할 수 있는 단락은 동일한 행 내에서의 경우와 서로 인접한 행들의 경우를 포함하며 서로 떨어져 있는 행들간의 단락은 여기에서 고려하지 않는다. 동일한 행 내에서의 고장과 이웃한 행들간의 고장을 나타내면 그림 1과 같다.



(a) 동일한 행에서의 고장 (b) 이웃한 행들간의 고장
 그림 1. Inter-port fault의 예
 Fig. 1. Example of inter port fault.

(2) AF2

이중 포트 메모리에서는 서로 다른 포트들을 통해 메모리 셀에 접근하므로 각각의 포트에 대한 어드레스 디코더가 별개로 존재하여서 동시에 다른 어드레스 디코더를 통해 다른 어드레스의 메모리 셀에 접근하여 데이터를 읽고 쓰는 동작을 수행할 수 있게 된다. 따라서 기존의 단일 포트 메모리의 어드레스 디코더 고장

위의 다른 고장 모델이 필요하며 이중 포트 메모리에서의 어드레스 디코더 고장은 하나의 포트와 관련된 어드레스 디코더 고장 AF1과 두 포트의 어드레스 디코더 관련 고장 AF2로 나눌 수 있다. 여기서 AF1은 하나의 포트의 어드레스 디코더에서의 고장을 나타낸다. 따라서 이러한 형태의 고장의 경우 단일 포트 메모리에서의 고장과 같고 테스트 방법 또한 동일하다. 반면 AF2는 두 포트의 어드레스 디코더의 라인들간의 단락이나 간섭에 의해 발생하는 고장으로 일반적인 단일 포트 메모리에서의 고장과는 차이가 존재한다^[15].

III. 진단 패턴

II장에서는 이중 포트 메모리에서의 고장 진단을 위해 고려할 고장 모델들에 대해 살펴 보았다. 이 장에서는 고려하는 고장들에 대해 진단 과정 수행을 위해 필요한 패턴들을 분석한다. 우선 이 고장이 하나의 셀에만 국한된 종류의 단일 셀 고장인지 여러 셀과 관련이 있는 다중 셀 고장인지 여부를 규명할 수 있어야 하며 또 일반적인 단일 포트 관련 고장인지 이중 포트 관련 고장인지도 규명되어 이중 포트 메모리에서 발생하는 고장에 대해 그 종류와 위치에 대한 분석이 명확하게 이루어질 수 있어야 한다.

1. 단일 포트 관련 고장 진단 (PTN-1PF)

(1) PTN-ST : 고착 고장, 천이 고장을 위한 패턴

$$W_{ij}(0), R_{ij}(0), W_{ij}(1), R_{ij}(1) \quad W_{ij}(0), R_{ij}(0)$$

여기서 W_{ij} 는 셀 (i, j)에 쓰기 동작을 수행하는 것을 의미하고 R_{ij} 는 우선 셀 (i, j)의 값을 읽는 동작을 의미한다. 고장셀 (i, j)에 위와 같이 0과 1을 쓰고 읽는 동작을 수행한다. 이러한 동작들을 수행함으로써 고장이 결합 고장인지 고착 고장나 천이 고장과 같은 단일셀 고장인지를 알 수 있다. 우선 위의 동작들을 수행할 때 고장이 검출되지 않는다면 고착 고장나 천이 고장과 같은 단일셀 고장이 아니므로 결합 고장이나 NPSF와 같은 다중 셀 고장에 대한 진단 과정이 필요하다. 이와는 달리 만일 여기서 $R_{ij}(0)$ 이나 $R_{ij}(1)$ 이 실패하여 고장이 검출된다면 고착 고장, 천이 고장과 같은 단일 셀 고장으로 진단된다. 하지만 이 경우 고장이 발생한 순간 셀 (i, j)의 이웃셀들의 특정 패턴들 때문에 위의 동작들이 실패하는 passive, static NPSF일 수도 있으므로 진단 과정에서 착오가 있을 수 있다. 따라서 셀 (i,

j)의 이웃셀들의 값을 바꾸어 준 후 위의 동작들을 다시 수행함으로써 고착 고장, 천이 고장과 같은 단일셀 고장인지 NPSF인지가 결정되므로 진단 과정은 완료된다.

(2) PTN-C : 결합 고장을 위한 패턴

(1)의 과정에서 고장이 검출되지 않은 경우들에 대해 아래와 같은 패턴들을 가해 주는 진단 과정을 수행한다. 이 과정에서는 메모리에서 존재하는 고장이 결합 고장일 경우 고장의 원인이 되는 결합원셀의 위치를 확인한다.

$$W_{ij}(0) \uparrow (W_n(0), W_n(1), R_{ij}(0), W_n(0), R_{ij}(0))$$

$$W_{ij}(1) \uparrow (W_n(0), W_n(1), R_{ij}(1), W_n(0), R_{ij}(1))$$

여기서 W_{ij} 는 셀 (i, j)에 쓰기 동작을 수행하는 것을 의미하며 W_n 은 셀 (i, j)를 제외한 모든 셀들에 쓰기 동작을 수행하는 것을 의미한다. 우선 셀 (i, j)에 0이나 1의 값을 쓰고 다른 셀들에 0을 쓴 후 다시 1을 쓰는 동작을 수행한 후 셀 (i, j)의 값을 읽어서 확인해 본다. 그리고 다시 0을 쓰고 셀 (i, j)의 값을 읽어서 확인한다. 이러한 동작을 어드레스를 변화시키면서 반복적으로 수행하는 중에 고장이 검출되지 않는다면 결합 고장이 아니므로 NPSF의 검출에 대한 진단을 수행하여야 한다. 만일 위의 동작을 수행하는 중 고장이 검출된다면 피결합셀 뿐 아니라 결합원셀의 위치도 확인되고 생각할 수 있다. 하지만 이 경우 결합원셀이 기준셀의 이웃셀중에 하나일 경우에는 고장이 발생한 순간 셀 (i, j)의 이웃셀들이 특정한 값을 가진 상태에서만 위의 동작이 실패하는 것일 가능성이 있으므로 결합 고장이 아니라 active NPSF일 수도 있다. 따라서 1.1의 경우와 마찬가지로 이웃셀들의 패턴을 바꾸어 준 후 위의 패턴들을 다시 수행함으로써 결합 고장인지 active NPSF인지를 결정하고 진단 과정을 완료하게 된다.

(3) PTN-N : NPSF를 위한 패턴

(1)과 (2)의 과정을 통해 검출되지 않는 고장일 경우 아래와 같이 좀 더 특수한 형태의 고장인 NPSF에 대한 진단 과정을 수행하게 된다. 여기서 테스트에서 고장이 검출된 셀 (i, j)는 기준셀이 되고 상하 좌우의 인접한 셀들이 이웃셀이 된다. 이러한 NPSF의 검출을 위해 필요한 패턴들은 다음과 같다.

① Active NPSF

$$W_{ij}(D)$$

$$APPLY(ANPSF)$$

$$R_{ij}(D)$$

여기서 D는 데이터 0 혹은 1을 의미하며 APPLY (ANPSF)는 active NPSF의 검출을 위해 필요한 패턴 들을 이웃셀들에 가하는 것을 의미한다. 필요한 패턴은 이웃셀들에 하나의 천이 쓰기 동작을 포함한 가능한 모든 상태를 의미한다. 따라서 이 경우에는 기준셀에 0 과 1의 값을 쓰고 이웃셀들에서 하나의 천이 동작을 포함한 모든 패턴들에 대해 각각 기준셀에 대해 0과 1 을 읽는 동작을 수행하여 검출 가능하다.

② Passive NPSF

APPLY(PNPSF)

$W_{ij}(0), R_{ij}(0), W_{ij}(1), R_{ij}(1) W_{ij}(0), R_{ij}(0)$

passive NPSF 고장의 경우는 active NPSF의 경우와 달리 이웃셀들의 모든 천이 쓰기 동작을 고려할 필요 없이 가능한 모든 상태들에 대해 각각 위의 동작을 기준셀에 수행하고 읽기 동작을 수행하여 값을 확인하면 된다. 따라서 APPLY(PNPSF)는 이웃셀들에 천이 동작을 고려할 필요없이 가능한 모든 상태를 적용하면 된다.

③ Static NPSF

$W_{ij}(D)$

APPLY(SNPSF)

$R_{ij}(D)$

이 경우도 passive NPSF와 마찬가지로 이웃셀들에 가능한 모든 상태를 가하는 패턴이 필요하다. 기준셀에 0과 1의 값을 쓰고 이웃셀들의 모든 패턴들에 대해 각 각 기준셀에 대해 0과 1을 읽는 동작을 수행하여 검출 가능하다.

(4) PTN-A : 어드레스 디코더 고장을 위한 패턴

(1)에서 (3)까지의 과정을 통해서 고장의 종류와 위치가 진단되지 않는다면 어드레스 디코더 고장일 가능성이 있고 이를 확인하기 위해 어드레스 디코더 고장에 대한 테스트를 수행한다. 이 고장은 메모리에서 어드레스 디코더 부분에서 관련된 고장으로 대부분의 경우 메모리 셀에서의 고장으로 매핑(mapping)된다. 다음과 같은 조건을 만족하는 테스트를 수행할 때 어드레스 디코더 고장을 검출할 수 있다.

$$\uparrow(R(D), \dots, W(\bar{D}))$$

$$\downarrow(R(\bar{D}), \dots, W(D))$$

앞의 진단 과정들에서는 위와 같은 패턴들을 모두 인가하지 않았기 때문에 모든 어드레스 디코더 고장을

검출할 수는 없다. 그리고 모든 어드레스 디코더 고장을 진단하기는 어려우므로 앞에서 말한 모든 고장들에 대한 진단 패턴으로도 검출이 되지 않을 경우에 한해 위에서 말한 조건을 만족하는 가장 짧은 길이의 MATS+ 알고리즘^[16]을 이용하여 어드레스 디코더 고장에 대한 진단을 수행한다.

2. 이중 포트 관련 고장 진단 (PTN-2PF)

(1) PTN-IP : Inter port fault를 위한 패턴

이중 포트 메모리의 서로 다른 포트들의 워드 라인들간의 단락인 inter port fault는 동일한 행내의 고장과 이웃한 행들간의 고장으로 나뉘어지게 되며 이 고장들은 and type 과 or type이 존재하게 된다. 진단 과정에서는 먼저 동일한 행내에서의 고장에 대한 진단 패턴을 인가하고 이를 통과할 경우에는 이웃한 행들간의 고장에 대한 패턴을 가하여서 inter port fault에 대한 고장의 종류와 위치를 진단하게 된다.

우선 동일한 행 내에서 inter port fault의 진단을 위한 패턴은 다음 표 1과 같다.

표 1에서 A와 B는 이중 포트 메모리에서의 서로 다른 포트를 의미하며, (i, j-1)이나 (i, j-2)는 메모리에서의 어드레스를 의미한다. 따라서 $W_A(0)_{i,j-1}$ 는 포트 A를 통해 어드레스 (i, j-1)인 셀에 0을 쓰는 동작의 수행을 의미하며 $R_A(0)_{i,j-1}$ 는 포트 A를 통해 어드레스 (i, j-1)인 셀로부터 0의 값을 읽는 동작을 수행하는 것을 의미한다. 동일한 행 내에서의 고장의 경우 OR type을 예로 들어 진단 패턴을 설명하면 다음과 같다.

표 1. 동일한 행에서의 inter port fault에 대한 진단 패턴

Table 1. Diagnosis patterns for inter port fault at same row.

| | 동일한 행 내에서의 고장을 위한 패턴 |
|----------|--|
| OR type | $W_A(0)_{i,j-1}, W_A(1)_{i,j-2}, \begin{pmatrix} R_A(0)_{i,j-1} \\ R_B(x)_{i,j-2} \end{pmatrix}$ |
| AND type | $W_A(1)_{i,j-1}, W_A(0)_{i,j-2}, \begin{pmatrix} R_A(1)_{i,j-1} \\ R_B(x)_{i,j-2} \end{pmatrix}$ |

우선 A 포트를 통해 어드레스 (i, j-1)인 셀에 0을 쓰고 (i, j-2)인 셀에 1을 쓴 후 A 포트를 통해 셀 (i, j-1)의 값을 읽는 동작을 수행하고 동시에 포트 B를 통해서 셀 (i, j-2)의 값을 읽는 동작을 수행한다. 고장이 존재하지 않을 때는 $R_A(0)_{i,j-1}$ 의 수행시 정상적으로

기대했던 0의 값을 읽게 된다. 하지만 만약 셀 (i, j-2)에 포트 A의 워드 라인과 포트 B의 워드 라인 사이에 단락 고장이 존재할 경우에는 B포트를 통해 어드레스 (i, j-2)의 값을 읽는 $R_B(x)_{i,j-2}$ 동작을 수행할 때 단락 고장 때문에 B 포트의 워드 라인 뿐만 아니라 A 포트의 워드 라인까지 선택되게 된다. 따라서 $R_B(x)_{i,j-2}$ 와 동시에 수행하는 $R_A(0)_{i,j-1}$ 동작을 수행할 때 고장이 존재하지 않을 경우 정상적으로 0의 값을 읽게 되지만 고장이 존재하는 경우는 A 포트 (i, j-1)의 워드 라인 뿐만 아니라 (i, j-2)에 해당하는 워드 라인도 선택되므로 $R_A(0)_{i,j-1}$ 의 수행 결과가 0이 아닌 1이 되게 된다. 따라서 어드레스 (i, j-2)에 해당하는 행 내에서 OR type의 inter port fault는 진단이 이루어지게 된다. AND type 고장의 경우도 이와 유사하게 표 1과 같은 패턴으로 진단 과정이 수행될 수 있다.

이러한 동일한 행 내에서의 고장의 경우와 달리 이웃한 행들간의 고장의 경우는 고장과 관련된 위치가 하나의 셀이나 행과 관련된 것이 아니고 두 개의 이웃한 행과 관련이 있으므로 이를 위한 별도의 진단 패턴이 필요하게 된다. 이웃한 두 행들간의 고장에 대한 진단을 위한 패턴은 다음 표 2와 같다.

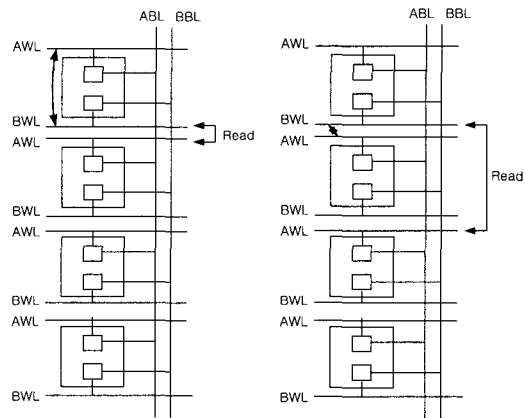
표 2. 이웃한 행에서의 inter port fault에 대한 진단 패턴

Table 2. Diagnosis patterns for inter port fault across the adjacent rows.

| | 이웃한 행 내에서의 고장을 위한 패턴 |
|----------|--|
| OR type | $W_A(0)_{i,j}, W_A(1)_{i,j-1}, W_A(1)_{i,j-2}, \begin{pmatrix} R_A(0)_{i,j} \\ R_B(x)_{i,j-2} \end{pmatrix}$ |
| AND type | $W_A(1)_{i,j}, W_A(0)_{i,j-1}, W_A(0)_{i,j-2}, \begin{pmatrix} R_A(1)_{i,j} \\ R_B(x)_{i,j-2} \end{pmatrix}$ |

표 2에서 알 수 있듯이 이웃한 행에서의 inter port fault에 대한 진단 패턴은 동일한 행 내에서의 고장에 대한 패턴과는 차이가 있다. OR type을 예로 들어 설명하면 다음과 같다. 우선 A 포트를 통해 셀 (i, j)에 0을 쓰고 셀 (i, j-1)에 1을 쓴다. 그리고 셀 (i, j-2)에도 포트 A를 통해 1을 쓰는 동작을 수행한다. 그리고 A 포트를 통해 셀 i에서 0을 읽는 $R_A(0)_{i,j}$ 동작과 B 포트를 통해 (i, j-2) 셀을 읽는 $R_B(x)_{i,j-2}$ 동작을 동시에 수행하게 된다. 고장이 존재하지 않는 경우에는 A 포트를 통한 읽기 동작에서 정상적으로 기대했던 0의 값을

읽게 된다. 그러나 만약 셀 (i, j-1)과 셀 (i, j-2)의 행들 사이에 단락이 존재한다면 $R_B(x)_{i,j-2}$ 동작을 수행할 때 B 포트의 셀 (i, j-2)의 행의 워드라인이 선택될 때 j-1과 j-2 사이에 존재하는 단락 때문에 A 포트 j-1의 워드 라인도 선택되게 된다. 따라서 $R_A(0)_{i,j}$ 동작을 $R_B(x)_{i,j-2}$ 동작과 동시에 수행할 때 고장이 존재하지 않는 경우는 정상적으로 0의 값을 읽게 되지만 고장이 존재하는 경우에는 j-1에 해당하는 A 포트의 워드 라인도 선택되게 되므로 A 포트를 통해 0의 값을 읽는 동작을 실패하고 1의 값을 읽게 되어 서로 다른 행들간의 inter port fault에 대한 진단이 이루어지게 된다. AND type 고장의 경우도 이와 유사하게 수행될 수 있다. 동일한 행에서와 서로 다른 행에서의 inter port fault에 대한 각각의 진단 패턴의 수행의 예는 그림 2와 같다.



(a) 동일한 행 내에서의 (b) 이웃한 행들간의 고장 고장

그림 2. Inter-port fault에 대한 진단 패턴 수행의 예 Fig. 2. Example of diagnosis procedure for the inter port fault.

(2) PTN-AF2 : 이중 포트 관련 어드레스 디코더 고장을 위한 패턴

이중 포트와 관련된 어드레스 디코더 고장인 AF2는 두 개의 어드레스 디코더에서 일어날 수 있는 단락이나 간섭에 의한 것으로 다양한 종류의 결함이 존재할 수 있지만 그 고장들 중 일부는 검출을 위해 두 개의 포트를 사용할 필요가 없고 하나의 포트만을 통해 일반적인 AF1에 대한 패턴들만으로 검출이 가능하다는 특징을 가진다. 따라서 이중 포트 관련 어드레스 고장인 AF2에서 AF1에 대한 패턴으로 검출 가능한 고장들

을 제외하고 두 개의 포트를 모두 사용하여 검출하기 위해 특별한 패턴이 필요한 고장들은 다음 표 3과 같다.

표 3. AF2의 간략화된 형태
Table 3. Simplified form of AF2.

| Fault | Name |
|---|---------|
| If $A_a(x) : A_b(\bar{y})$ then $C_a(x) : C_b(y)$ | Fault E |
| If $A_a(x) : A_b(z) z \neq y$ then $C_a(x) : C_b(z) : C_b(y)$ | Fault F |
| If $A_a(x) : A_b(y)$ then $C_a(x) : C_b(\emptyset)$ | Fault G |

표 3에서 나타낸 AF2의 표기는 다음과 같다.

- ◆ $A_a(x)$ - 포트 a를 통해서 어드레스로 행 x를 선택하는 것을 의미한다.
- ◆ $C_a(x)$ - 포트 a를 통해 행 x의 셀에 접근한 것을 의미한다.
- ◆ $A_b(\bar{y})$ - 포트 b를 통해 어드레스로 행 y가 선택되지 않는 것을 의미하며 이 경우 y 외의 다른 행은 선택될 수 있다.
- ◆ $A_a(x) : A_b(y)$ - a 포트를 통해서 행 x를 선택하고 동시에 b 포트를 통해서 행 y를 선택하는 것을 의미한다.
- ◆ $C_a(x) : C_b(y)$ - a 포트를 통해서 행 x의 셀에 접근하고 b 포트를 통해서 행 y의 셀에 접근하는 것을 의미한다.
- ◆ $C_b(\emptyset)$ - b 포트를 통해서 어떤 셀에도 접근이 되지 않는 경우를 의미한다.

표 3의 AF2들을 검출하기 위해서는 모든 어드레스들의 조합을 생성할 필요가 있다. 따라서 모든 어드레스 조합을 통해 고장을 유발하여서 모든 종류의 AF2를 검출하기 위한 패턴은 그림 3과 같다. 그림 3을 통해 행 디코더의 단락을 통해 발생할 수 있는 모든 종류의 고장들을 검출 가능하고 열 디코더의 경우도 그림 3과 유사하게 검출 가능하다.

이중 포트 메모리에서의 어드레스 디코더 고장에 대한 진단 과정은 크게 AF1에 대한 과정과 AF2에 대한 과정으로 나뉘어지게 된다. 우선 AF1에 대한 진단 과정은 1.4에서 설명한 PTN-A를 하나의 포트를 활성화하지 않은 채 다른 포트에 대해 단일 포트에 대한 패턴들을 가해 주는 방식으로 두 포트 모두에 대해 진단

과정을 수행하게 된다. 그리고 AF2의 진단을 위해서는 그림 3의 패턴 PTN-AF2를 사용하는데 이 경우 두 포트 모두에 대해 동시 접근이 이루어지게 된다. 따라서 이와 같은 과정을 통해 이중 포트 메모리에서 모든 AF에 대한 진단이 가능하게 된다.

```

select two columns c1 and c2 ; c1≠c2 ;
for all r // r∈{0, 1, ..., R-1}
{ w0r,c1 : n ; // c1을 0으로 초기화
  n : w0r,c2 ; // c2를 0으로 초기화
}
for(r1=0 ; r1<R ; r1++)
{ for(r2=0 ; r2<R ; r2++)
  { r0r1,c1 : r0r2,c2 ; // 고장 검출
    w1r1,c1 : w1r2,c2 ; // 고장 유발
    r1r1,c1 : n ; // 고장 검출
    n : r1r2,c2 ; // 고장 검출
    w0r1,c1 : n ;
    n : w0r2,c2 ;
  }
}
for all r
{ r0r,c1 : - ; // 고장 검출
}
    
```

그림 3. 이중 포트 메모리에서 AF2를 위한 패턴
Fig. 3. Patterns for AF2 of dual port RAMs.

IV. 이중 포트 메모리의 고장 진단 알고리즘

II장과 III장을 통해 본 연구에서 고려하는 고장 모델을 결정하였고 고장의 효율적인 진단을 위한 패턴들을 알아보았다. 이 장에서는 각각의 고장들에 대한 진단 패턴을 바탕으로 이중 포트 메모리에서의 고장을 위한 전체 진단 과정을 제안한다.

1. 이중 포트 메모리의 진단 과정

우선 이중 포트 메모리를 위한 전체 진단 과정을 위해서는 단일 포트 관련 고장 진단 과정인 PTN-1PF를 각각의 포트를 통해 수행한다. 그리고 이중 포트 관련 고장 진단 과정인 PTN-2PF를 수행한다. 이중 포트 메모리에서의 전체적인 진단 방법은 다음 그림 4와 같다.

여기서 1PF는 단일 포트와 관련된 고장들을 의미하고 2PF는 이중 포트와 관련된 고장을 의미한다. 1PF에는 고착 고장, 천이 고장, 결합 고장, NPSF, 어드레스 디코더 고장 등이 속하고, 2PF에는 inter-port fault, 어드레스 디코더 고장이 속한다. 또 PTN - 1PF은 단일 포트 관련 고장인 1PF에 대한 진단 패턴을 의미하며

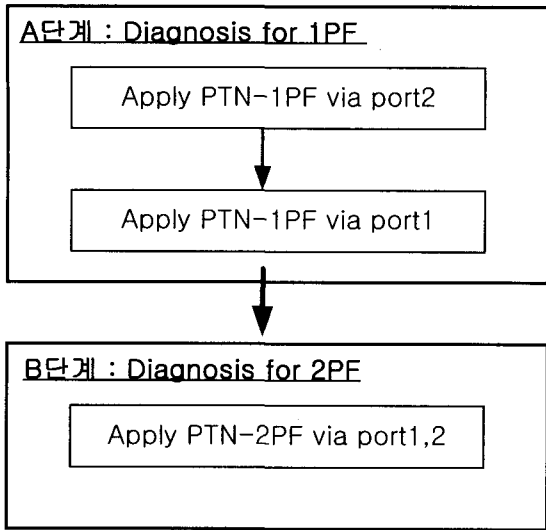


그림 4. 이중 포트 메모리의 개략적인 전체 진단 과정
Fig. 4. Summarized diagnosis procedure of dual port RAMs.

PTN - 2PF는 2PF에 대한 진단 패턴을 의미한다. 따라서 A 단계를 통해 우선 이중 포트 메모리에서 단일 포트 관련 고장들은 모두 진단이 가능하다. 그리고 단일 포트 관련 고장에 대한 진단 과정인 A 단계를 통해서도 진단이 이루어지지 않을 경우에는 이중 포트 관련 고장에 대한 진단 과정인 B 단계를 통해 진단 과정을 수행한다. 그러므로 그림 4와 같은 과정을 수행함으로써 이중 포트 메모리에서의 모든 고장들에 대한 진단이 이루어지게 된다.

2. 이중 포트 메모리를 위한 고장 진단 알고리즘

(1) 단일 포트 관련 고장을 위한 알고리즘

앞에서 이중 포트 메모리에서 발생하는 모든 고장에 대한 패턴들과 개략적인 진단 과정을 알아보았다. 여기에서는 그림4의 이중 포트 메모리 진단 과정에서 A 단계를 위한 단일 포트 관련 고장들에 대한 상세한 알고리즘을 제안한다. 단일 포트 관련 고장들에 대한 알고리즘은 다음 그림 5와 같다.

그림 5에서는 고착 고장, 천이 고장, 결합 고장, NPSF, 어드레스 디코더 고장 등의 단일 포트 관련 고장들에 대한 진단 패턴을 사용하여 고장 진단을 수행하게 된다. 이러한 과정을 각각의 포트들을 통해 수행함으로써 이중 포트 메모리에서 하나의 포트와 관련된 고장들에 대한 진단이 이루어지게 된다.

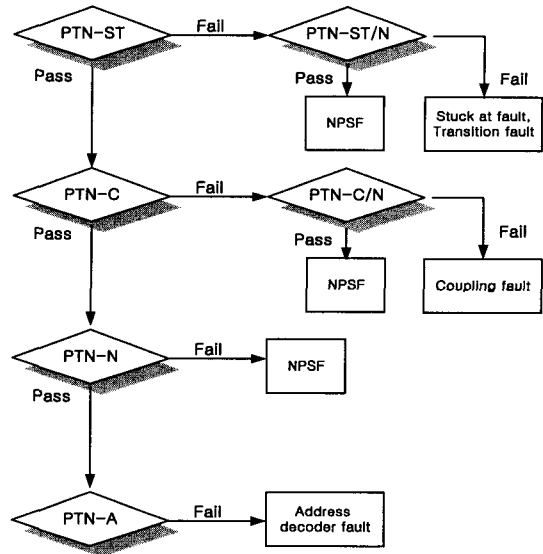


그림 5 단일 포트 관련 고장을 위한 진단 알고리즘
Fig. 5. Fault diagnosis algorithm for 1PF.

(2) 이중 포트 관련 고장을 위한 알고리즘

이중 포트 메모리에서는 그림 5와 같은 하나의 포트와 관련이 있는 고장들에 대한 알고리즘의 수행을 통해서 진단이 이루어지지 않는 고장이 존재한다. 이는 두 개의 포트와 관련된 고장으로 본 논문에서는 II장에서 설명한 바와 같이 inter port fault와 AF2를 고려한다. 이러한 이중 포트 관련 고장들에 대한 알고리즘은 다음 그림 6과 같다.

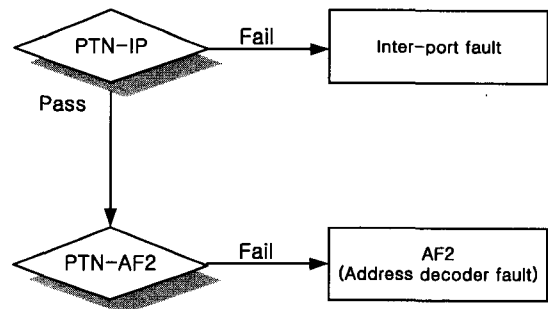


그림 6. 이중 포트 관련 고장을 위한 알고리즘
Fig. 6. Fault diagnosis algorithm for 2PF.

그림 6에서는 이중 포트 관련 고장을 위한 진단 패턴인 PTN-IP와 PTN-AF2를 사용하여 고장 진단을 수행한다. 이러한 과정을 수행함으로써 이중 포트 메모리에서 두 개의 포트와 관련된 고장들에 대한 진단이 이루어지게 된다.

(3) 이중 포트 메모리의 고장 진단 알고리즘

앞의 2.1과 2.2에서는 하나의 포트와 관련된 고장들과 두 개의 포트와 관련된 고장들의 진단을 위한 알고리즘을 제안하였다. 이를 바탕으로 이중 포트 메모리에서 고장 진단을 위한 알고리즘은 다음 그림 7과 같다. 그림 7의 진단 과정을 차례대로 설명하면 다음과 같다. 우선 PTN-ST의 패턴은 단일 셀 고장인지 여부를 판단하기 위해 사용된다. PTN-ST의 과정을 통과할 경우 단일 셀 고장이 아닌 다중 셀 고장으로 예측되어 PTN-C의 과정을 수행하게 되며 실패할 경우에는 단일 셀 고장으로 예상되지만 NPSF일 가능성이 존재하므로 위에서 설명한 것처럼 이웃셀의 패턴을 바꾸어 주는 PTN-ST/N을 통해 고착이나 천이 고장인지 NPSF인지 진단을 완료하게 된다. 다음으로 PTN-ST를 통과한 경우에 대해 PTN-C의 패턴을 인가하게 되는데 이를 통해 결합 고장을 발생하게 하는 결합원셀을 찾을 수 있다. PTN-C의 과정에서 실패하는 경우 역시 결합 고장 외에 NPSF일 가능성이 있으므로 이웃셀의 패턴을 바꾸어 주는 PTN-C/N를 통해 결합 고장인지 NPSF인지를 진단하는 과정을 완료하게 된다. 그리고 PTN-C의 과정을 통과할 경우에는 NPSF에 대한 테스트인 PTN-N의 과정을 수행하게 된다. 여기서 실패할 경우에는 NPSF로 진단 과정이 완료되며 통과할 경우에는 어드레스 디코더 고장에 대한 테스트인 PTN-A의 과정을 수행하게 된다. 진단 과정의 마지막 과정인 PTN-A에서 실패할 경우는 어드레스 디코더 고장으로 진단이 완료된다. 하지만 만약 이 과정에서도 통과하게 된다면 이중 포트 관련 고장들로 예측될 수 있으므로 PTN-IP와 PTN-AF2의 과정을 통해 이중 포트 관련 고장에 대한 진단을 수행하게 된다. 그림 7의 고장 진단 과정에서 PTN-ST, PTN-C, PTN-N, PTN-A는 단일 포트 관련 고장들에 대한 진단 패턴들로 PTN-IPF를 의미한다. III장에서 보여준 바와 같은 이러한 패턴들을 이용한 과정들을 각각의 포트를 통해 수행함으로써 단일 포트 고장들의 경우 진단 과정이 완료된다. 하지만 메모리에서의 고장이 단일 포트 관련 고장이 아닌 두 포트와 관련된 고장이라면 이러한 과정들로만 진단 과정이 완료되지 않는다. 따라서 이중 포트 관련 고장들에 대한 진단 패턴인 PTN-IP와 PTN-AF2를 이용한 과정을 통해 고장의 진단 과정을 수행하게 된다. 우선 PTN-IP를 이용하여 동일한 행에서 혹은 이웃한 행에서 단락이 존재할 경우 그러한 고장의 종류 및 위

치에 대한 진단이 이루어지게 된다. 만약 이 과정에서 진단이 이루어지지 않을 경우에는 두 포트의 어드레스 디코더와 관련된 PTN-AF2를 이용한 과정을 수행하여 최종적으로 진단 과정을 완료하게 된다. 따라서 이러한 과정들을 통해 이중 포트 메모리에서 고장이 발생할 경우 대부분의 고장에 대해 그 종류와 고장이 검출되는 셀 뿐만 아니라 고장과 관련된 셀 등의 고장의 위치까지 진단하는 것이 가능하다.

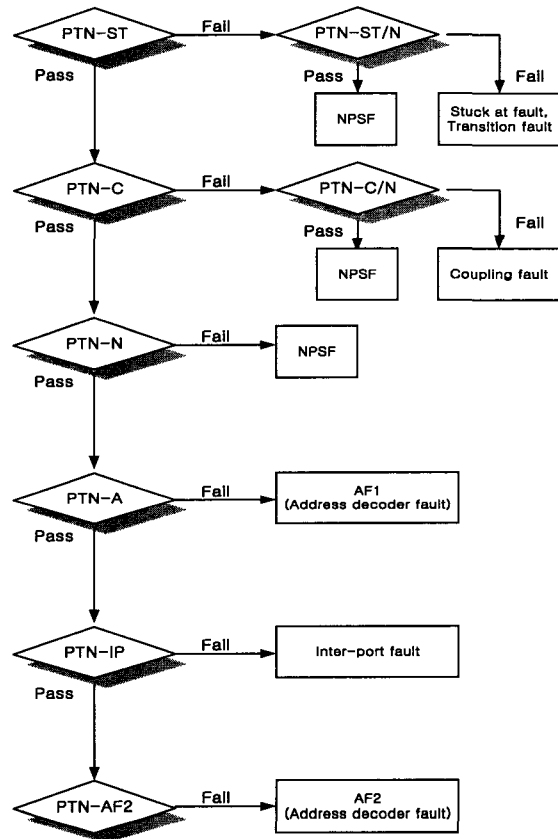


그림 7. 이중 포트 메모리 고장 진단 알고리즘
Fig. 7. Fault diagnosis algorithm for dual port RAMs.

3. 테스트 정보의 분석을 이용한 고장 진단 방법

앞에서 살펴본 고장 진단 방법은 테스트 방법들에 대한 정보가 없는 경우로 테스트 알고리즘을 수행하는 과정이나 그에 따른 결과들에 대해 알지 못하는 경우를 말하며 이 경우 단지 고장이 검출된 셀만을 파악한 상태에서 진단 과정에 착수하게 된다. 이러한 경우에는 테스트에서 고장이 검출된 셀을 대상으로 III장에서 설명된 진단 패턴들을 적절한 시퀀스로 가해 주어서 IV

장의 2결과 같은 진단 알고리즘을 통해 검출되는 고장의 종류를 분류하고 고장과 관련된 모든 셀들을 확인할 수 있음을 살펴보았다. 이 절에서는 테스트 방법에 대한 정보의 분석을 이용한 고장 진단 방법을 알아본다. 테스트 알고리즘을 수행하는 과정이나 각 셀에서의 결과값에 대한 정보를 얻을 수 있는 경우를 말한다. 이러한 경우에는 메모리를 테스트하는 알고리즘에 따른 분석을 바탕으로 알고리즘 수행 시 각 셀에서의 결과를 진단 과정에 이용함으로써 고장의 예측 가능성을 증대시킨다.

메모리 테스트를 위한 알고리즘의 각각의 읽기 동작에서 고장이 없을 경우와 고장이 존재하는 경우 또 그 고장이 어떠한 종류의 고장 모델인지에 따라서 알고리즘에서 읽기 동작을 수행하였을 때 나타나는 결과가 달라질 수 있다. 따라서 고려하는 각각의 고장 모델에 따라 테스트 알고리즘의 읽기 동작에서 나타나는 결과들이 분석된다면 테스트 알고리즘을 메모리에 가했을 때 나타나는 결과를 관찰하는 것으로서 메모리에서의 고장에 대한 예측의 가능성이 높아지므로 진단 과정 또한 일반적인 진단 과정과 다른 형태로 변화될 수 있다. 이러한 경우에는 읽기 동작에서 고장이 검출되어 테스트 알고리즘이 중지되는 시점의 마치 요소(march element)에서 최초로 발생하여 검출될 수 있는 고장에 대한 분석이 필요하다.

여기에서는 메모리의 테스트 알고리즘으로 임의의 마치 요소를 가지는 알고리즘이 사용된 경우 고장이 발생했을 때 진단하는 방법에 대해 알아본다. 우선 이를 위해서는 진단 과정에서 고려하는 각각의 고장 모델들과 이를 검출하는 마치 요소들에 대한 분석이 필요하다. 그리고 테스트에서 고장이 검출된 부분의 마치 요소에 따라 진단 과정을 수행해야 한다. 이러한 테스트 정보를 이용한 고장 진단 방법은 다음 그림 8과 같이 나타낼 수 있다.

그림 8의 테스트 정보의 분석을 이용한 고장 진단 방법을 위해서 진단 과정에서 고려하는 각각의 고장 모델을 테스트 시 검출할 수 있는 패턴은 다음과 같이 표 4와 같이 나타낼 수 있다. 표 4에는 진단 과정에서 고려하는 고장들과 이를 테스트 알고리즘에서 검출하는 테스트 패턴들을 나타낸다. 그러므로 마치 형태의 테스트 알고리즘을 통해 고장을 검출할 경우 고장을 검출한 읽기 동작이 포함되는 마치 요소들 중 표 4에 포함되는 것들에 따른 고장으로 예측될 수 있고 그에

따른 진단 과정을 우선적으로 수행해야 할 필요가 있다.

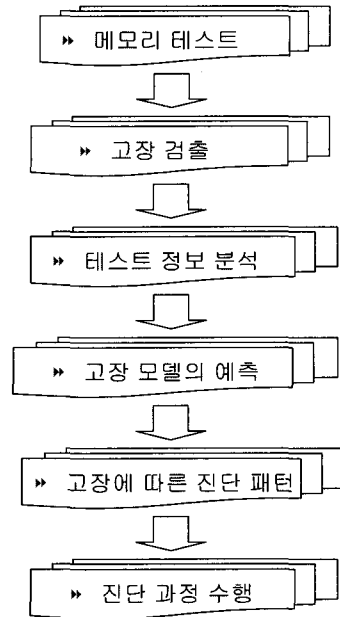


그림 8. 테스트 정보의 분석을 이용한 고장 진단 방법
Fig. 8. Fault diagnosis by analyzing test information.

표 4. 각각의 고장 모델들을 검출하는 마치 요소들

Table 4. March elements detecting each fault model.

| 고장 모델 | 마치 요소 | |
|----------|----------------|----------------|
| SA0 | ↓(R1) | |
| SA1 | ↓(R0) | |
| TF↑ | ↓(R0,W1) ↓(R1) | |
| TF↓ | ↓(R1,W0) ↓(R0) | |
| 결합 고장 | ↑:0(a<v) | ↓(R0,W1) ↓(R1) |
| | ↑:0(a>v) | ↑(R0,W1) ↓(R1) |
| | ↓:0(a<v) | ↑(R1,W0) |
| | ↓:0(a>v) | ↓(R1,W0) |
| | ↑:1(a<v) | ↑(R0,W1) |
| | ↑:1(a>v) | ↓(R0,W1) |
| | ↓:1(a<v) | ↓(R1,W0) ↓(R0) |
| | ↓:1(a>v) | ↑(R1,W0) ↓(R0) |

진단 과정을 수행하기 위해 각 고장들에 대한 변형된 진단을 위한 패턴들은 표 5와 같이 나타낼 수 있다. 따라서 마치 알고리즘을 이용한 테스트를 수행하는 중 고장이 검출되면 고장을 검출하는 마치 요소에 따라

표 4와 같은 고장 모델들에 대한 진단 과정이 필요하며 이에 따라 표 5과 같이 각각의 고장 모델들에 대해 우선적으로 진단 패턴을 사용하여야 한다.

표 5. 고장 모델들에 따른 변형된 진단 패턴들
Table 5. Modified diagnosis patterns for each fault model.

| 고장 모델 | 변형된 진단 패턴 |
|-------|--|
| SA0 | $W_{ij}(1), R_{ij}(1)$ |
| SA1 | $W_{ij}(0), R_{ij}(0)$ |
| TF ↑ | $W_{ij}(0), R_{ij}(0), W_{ij}(1), R_{ij}(1)$ |
| TF ↓ | $W_{ij}(1), R_{ij}(1), W_{ij}(0), R_{ij}(0)$ |
| 결합 | ↑:0(a<v) $W_{ij}(1) \uparrow(W_n(0), W_n(1), R_{ij}(1))$ |
| | ↑:0(a>v) $W_{ij}(1) \downarrow(W_n(0), W_n(1), R_{ij}(1))$ |
| | ↓:0(a<v) $W_{ij}(1) \uparrow(W_n(1), W_n(0), R_{ij}(1))$ |
| | ↓:0(a>v) $W_{ij}(1) \downarrow(W_n(1), W_n(0), R_{ij}(1))$ |
| 고장 | ↑:1(a<v) $W_{ij}(0) \uparrow(W_n(0), W_n(1), R_{ij}(0))$ |
| | ↑:1(a>v) $W_{ij}(0) \downarrow(W_n(0), W_n(1), R_{ij}(0))$ |
| | ↓:1(a<v) $W_{ij}(0) \uparrow(W_n(1), W_n(0), R_{ij}(0))$ |
| | ↓:1(a>v) $W_{ij}(0) \downarrow(W_n(1), W_n(0), R_{ij}(0))$ |

따라서 테스트 과정이나 결과에 대한 정보를 이용한 고장 진단 과정은 그림 5와 7의 진단 알고리즘을 수행함에 있어 각각의 일반적인 진단 패턴들 대신 표 4와 표 5에 의해 그림 8에서 나타내는 바와 같이 예측되는 고장 모델들에 따른 변형된 진단 패턴들을 사용해서 좀 더 간략화될 수 있으며 진단 과정에 소요되는 시간을 줄일 수 있다.

V. 성능 평가 비교 및 분석

이 장에서는 본 연구를 통해 제안한 진단 방법을 기존의 제안된 진단 방법들과 비교하기 위해서 진단 가능한 고장 모델, 진단 가능 범위, 진단 대상 메모리 등의 측면에서 진단 능력을 분석한다. 그리고 이를 바탕으로 기존의 진단 방법인 [12], [13]과의 성능 비교를 통해 본 논문에서 제안한 방법의 이중 포트 메모리에 대한 효율성을 확인한다.

[12]의 방법은 메모리의 진단을 위한 별도의 패턴을 사용하지 않고 테스트 알고리즘의 수행 결과 분석을

통해 고장의 종류를 분석한다. 메모리 테스트 알고리즘에서 각각의 고장 모델이 존재할 경우에 따른 각각의 마치 요소에서의 고장 검출 여부에 대한 결과를 표를 통해 작성하고 이를 테스트 결과와 비교하여 메모리의 셀에서 고장이 존재할 경우 각 셀에서의 고장의 종류를 분석한다. 이러한 [12]는 고려하는 고장의 종류를 모두 구별하는 진단이 가능하지만 고려하는 고장이 고착 고장, 결합 고장일 경우에 국한된 결과일 뿐이고 다양한 고장들에 대한 실제적인 진단이 불가능하다는 문제점이 있다. 반면 제안된 방법은 고착 고장, 천이 고장, 결합 고장, 어드레스 디코더 고장, NPSF, inter port fault 등 이중 포트 메모리에서의 다양한 고장들에 대한 진단이 가능하다는 장점이 있다. 그리고 [12]의 경우 고려하는 고장의 종류를 구별하는 것은 가능하지만 고장과 관련된 셀을 찾는 진단 과정이 이루어지지 않으므로 고장의 위치를 명확하게 진단할 수 없다. 따라서 결합 고장이 존재할 경우 종류는 구별할 수 있지만 고장의 위치를 확인할 수 없다. 이에 반해 제안된 방법에서는 고장이 검출되는 셀 뿐 아니라 고장과 관련이 있는 모든 셀을 확인함으로써 고장의 종류 뿐 아니라 고장의 발생 위치까지도 명확하게 진단할 수 있다.

그리고 [13]에서 제안된 방법은 제안된 초기 마치 테스트를 수행하여 단일 셀 고장인지 다중 셀 고장인지를 구별하고 결합원셀을 찾기 위한 패턴을 사용한다. 따라서 피결합셀외에 결합원셀을 찾아낼 수 있다. 하지만 이러한 방법의 경우에는 진단 과정의 수행을 위해 반드시 초기 마치 테스트를 수행하는 과정이 포함되어야 하며 모든 경우의 NPSF에 대한 진단이 불가능하다는 단점이 있다. 그리고 [12]와 [13]의 방법은 모두 이중 포트 메모리에서의 고장 진단이 불가능하다는 단점이 있다.

본 연구에서 제안된 방법에서는 이중 포트 메모리의 대부분의 고장 모델의 종류 분석이 가능하고 고장이 검출되는 셀 뿐만 아니라 다양한 고장의 종류 및 고장의 원인이 될 수 있는 셀과 같이 고장과 관련된 모든 셀의 확인을 통해 고장의 위치에 대한 진단이 가능하다. 또 일반적인 진단 과정 뿐 아니라 메모리 테스트 정보의 분석을 이용한 진단 패턴의 변형 및 진단 방법을 확보하였다. 또한 [12]와 [13]과 같은 기존의 진단 방법들이 단일 포트 메모리에서의 고장에 대한 진단만을 다루었지만 본 연구에서는 이중 포트 메모리에서 발생할 수 있는 고장들에 근거하여 이중 포트 메모리

표 6. 다양한 진단 방법들의 비교
Table 6. Comparison of various diagnosis method.

| 방법들 | [12] | [13] | 제안된 방법 |
|-----------|---------------------------|--|--|
| 고장 모델 | 고착 고장, 천이 고장, 결합 고장 | 고착 고장, 천이 고장, 어드레스 디코더 고장, 결합 고장 | 고착 고장, 천이 고장, 어드레스 디코더 고장, 결합 고장, NPSF Inter-port fault |
| 진단 범위 | · 고장 관련 셀 불가능 | · 고장 관련 셀 가능 | · 고장 관련 셀 가능 |
| 제한 요소 | · 고장 관련 셀에 대한 진단 불가능 | · 초기 마치 테스트의 필요 · 모든 경우의 NPSF에 대한 진단은 불가능 | · 고장 관련 모든 셀에 대한 진단으로 인한 소요 시간 증가 ($O(N^2)$) |
| 진단 대상 메모리 | · 단일 포트 메모리 | · 단일 포트 메모리 | · 단일 포트 메모리 · 이중 포트 메모리 |

에서의 효과적인 고장 진단을 가능하게 한다. 본 연구에서 제안한 메모리 진단 방법과 기존의 제안된 진단 방법들의 종합적인 성능 비교 평가 결과는 표 6과 같다.

VI. 결 론

메모리 테스트를 통해 고장이 검출될 경우 고장 셀을 여분의 셀로 교체하는 수리 과정을 통해 수율을 향상시키는 것이 메모리 생산에서 중요한 요소이다. 그러므로 내장형 고용량 메모리를 위한 내장된 자체 수리 기법 (Built in self repair : BISR)의 개발을 위한 기반 기술을 확보하고 메모리의 수리를 통해 수율을 향상시키기 위해서는 메모리 테스트와 더불어 검출된 고장의 종류와 위치를 명확하게 결정하기 위한 진단 방법의 연구가 필수적이다.

그러나 기존의 진단 방법들은 단일 포트 메모리의 경우에 국한되어 있어 이중 포트 메모리에서 발생할 수 있는 고장들을 고려하지 못하여 고장의 종류 및 위치를 효과적으로 진단 할 수 없다. 따라서 본 연구에서는 이중 포트 메모리에서 존재할 수 있는 다양한 고장 모델을 고려하는 효율적인 고장 진단 방법을 제안하였다.

본 연구에서는 이중 포트 메모리에서 발생 가능한 대부분의 고장 모델들을 고려하고 이에 근거하여 각각의 고장들에 대한 진단을 위한 패턴들을 분석하였다. 또 이를 바탕으로 테스트에서 고장이 검출될 경우 고

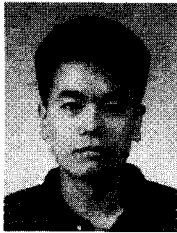
장의 종류를 구별하는 것을 가능하게 하는 진단 방법을 제안하고 단지 테스트에서 고장이 검출되는 셀 뿐만 아니라 고장과 관련이 있을 수 있는 셀들을 포함하여 고장과 관련된 모든 셀 들을 확인함으로써 이중 포트 메모리에서의 고장의 종류 및 위치의 분석 방식을 확보하였다. 또한 단일 포트 메모리를 위한 고장 진단 방법과 테스트 정보의 분석을 이용한 고장 진단 방법도 제시하였다. 따라서 제안된 고장 진단 알고리즘을 통해 이중 포트 메모리 및 단일 포트 메모리에서 발생하는 거의 모든 종류의 고장들에 대한 진단이 가능함을 알 수 있다. 이를 통해 이중 포트 메모리에서 고장의 발생 특성을 분석 방법을 확보하는 것을 가능하게 하고 고장이 검출된 메모리의 효율적인 재생산을 위한 진단 과정을 가능하게 하였다. 그리고 제안된 고장 진단 방법과 이전의 다른 연구들과의 성능 평가는 본 논문의 이중 포트 메모리에 대한 고장 진단 방법의 효율성을 보여준다.

참 고 문 헌

[1] J. Otterstedt, D. Niggemeyer, T. W. Williams, "Detection of CMOS address decoder open faults with March and pseudo random memory tests," Test Conference, 1998. Proceedings., International, 1998, Page(s): 53~62.

- [2] Syng-Jyan Wang, Chen-Jung Wei, "Efficient built-in self-test algorithm for memory," Asian Test Symposium, 2000. (ATS 2000). Proceedings of the Ninth, 2000, Page(s): 66~70.
- [3] Chih-tsunHuang, Jing-Reng Huang, "A programmable built-in self-test core for embedded memories," Design Automation Conference, 2000. Proceedings of the ASP-DAC 2000. Asia and South Pacific, 2000, Page(s): 11~12.
- [4] M. Azimane, A. L. Ruiz, "New short and efficient algorithm for testing random-access memories," Electronics, Circuits and Systems, 1998 IEEE International Conference on, Volume: 1, 1998, Page(s): 541~544.
- [5] V. Kim, T. Chen, "Assessing defect coverage of memory testing algorithms," VLSI, 1999. Proceedings. Ninth Great Lakes Symposium on, 1999, Page(s): 340~341.
- [6] H. Yokoyama, H. Tamamoto, Wen Xiaoqing, "Built-in random testing for dual-port RAMs", Memory Technology, Design and Testing, 1994., Records of the IEEE International Workshop on, 1994, Page(s): 2~6.
- [7] T. Matsumura, "An efficient test method for embedded multi-port RAM with BIST circuitry", Memory Technology, Design and Testing, 1995., Records of the 1995 IEEE International Workshop on, 1995, Page(s): 62~67.
- [8] A. J. van de Goor, S. Hamdioui, "Fault models and tests for two-port memories", VLSI Test Symposium, 1998. Proceedings. 16th IEEE, 1998, Page(s): 401~410.
- [9] M. F. Chang, W. K. Fuchs, J. H. Patel, "Diagnosis and repair of memory with coupling faults," computers, IEEE Transactions on, volume.38, No.4, April 1989, Page(s): 493~500.
- [10] Lin Shen, B. F. Cockburn, "An optimal march test for locating faults in DRAMs," Memory Testing, 1993., Records of the 1993 IEEE International Workshop on, 1993, Page(s): 61~66.
- [11] Chin Tsung Mo, Chung Len Lee, Wen Ching Wu, "A self-diagnostic BIST memory design scheme," Memory Technology, Design and Testing, 1994., Records of the IEEE International Workshop on, 1994, Page(s): 7~9.
- [12] C. F. Wu, C. T. Huang, "Error catch and analysis for semiconductor memories using march tests," Computer-Aided Design, 2000. ICCAD 2000. Digest of Technical Papers. 2000 IEEE/ACM International Conference on, 2000, Page(s): 468~471.
- [13] T. J. Bergfeld, D. Niggemeyer, E. M. Rudnick, "Diagnostic testing of embedded memories using BIST", Design, Automation and Test in Europe Conference and Exhibition 2000. Proceedings, 2000, Page(s): 305~309.
- [14] J. Zhao, S. Irrinki, M. Puri, F. Lombardi, "Detection of inter-port faults in multi-port static RAMs", VLSI Test Symposium, 2000. Proceedings. 18th IEEE, 2000, Page(s): 297~302.
- [15] S. Hamdioui, A. J. Van De Goor, "Address decoder faults and their tests for two-port memories", Memory Technology, Design and Testing, 1998. Proceedings. International Workshop on, 1998, Page(s): 97~103.
- [16] A. J. van de Goor, Testing Semiconductor Memories: Theory and practice, J. Wiley & Sons, 1991.

저 자 소 개



朴 漢 源(正會員)

1976년 4월 18일 생. 1999년 : 연대
전기공학과 졸업. 1999년~현재 연
대 전기전자공학과 석사과정



姜 成 昊(正會員)

1963년 4월 13일 생. 1986년 2월 :
서울대 공대 제어계측공학과 졸업.
1988년 5월 : The University of
Texas at Austin. 전기 및 컴퓨터공
학과 졸업(석사). 1992년 5월 : The
University of Texas at Austin 전
기 및 컴퓨터공학과 졸업(공학). 미국 Schlumberger
연구원. Motorola 선임 연구원. 현재 연대 공과대학 전
기전자공학과 부교수