

RAM 의 최소 테스트 패턴에 관한 연구 A Study on the Minimal Test Pattern of the RAM

김 철운*, 정 우성, 김 태성
전남대학교 전기공학과

Cheol-Woon Kim*, Woo-Sung Jung, Tae-Sung Kim
Dept. of Electrical Eng., Chonnam National University.

ABSTRACT

In this paper aims at studying the minimal test pattern of the RAM. This also propose a scheme of testing faults from the new fault model using the LLB. The length of test patterns are $6N(1-wsf)$, $9.5N(2-wsf)$, $7N(3-wsf)$, $3N(4-wsf)$ operations in N -bit RAM. This test techniques can write into memory cell the number of write operations is reduced and then much testing time is saved. A test set which detects all positive-negative static t -ws faults for $t=0,1,2,3,4$ and detects all pattern sensitive fault in memory array. A new fault model, which encompasses the existing fault model is proposed.

1. 서론

현재의 메모리 제조 기술은 갈수록 발전되어 좁은 면적에 더욱 많은 소자들을 배치할 수 있다. IC의 복잡도는 칩의 성능에 비례하여 높아지고 있으며 마이크로 프로세서와 같은 칩들의 테스트는 수백만개의 테스트 벡터가 필요하며, 테스트에 걸리는 시간은 생산단가와 비례한다. 그러므로 모든 칩은 보다 빠른 시간내에 정확한 테스트가 요구된다. 메모리 칩 테스트의 주요 연구는 최소의 테스트 패턴으로 모든 고장을 완전하게 검출할 수 있는 방법에 있다.[1] 이 논문에서는 기존의 테스트 방법보다 적은 테스트 패턴으로 RAM에서 발생할 수 있는 t -ws 고장을 테스트할 수 있는 방법에 대하여 연구 하였다.

II. 이론적 배경

메모리 테스트에서 가장 많이 알려진 고장은 다음과 같다. Stuck-at 고장은 메모리 셀의 입력단이나 출력단이 0 이나 1 로 고정되어 있는 고장이다. 따라서 한셀에 s -a 고장이 발생하면 그 셀은 어떠한 읽기나 쓰기 동작에 관계없이 0 또는 1 로 고정되어 아무런 동작을 할수 없게되는 고장이다.[1]

천이 고장은 메모리 셀에 쓰기 동작을 할 때 X 에서 X'로 천이가 불가능한 고장[3]이며, 결합 고장은 한 셀에 0 에서 1 (1 에서 0)로 쓰기 동작을 했을때 그 영향을 받아 다른

셀의 내용이 변하게 될때 서로 다른 두 셀에 발생하는 고장으로 결합고장이라고도 한다.[4] 그리고 메모리 셀에 고장이 발생했는데도 불구하고 고장이 아닌 것 처럼 동작하는 경우의 회로도 있는데 이를, 용장성 소자라고 한다. 테스트 패턴은 모든 가능한 고장을 완벽하게 검출할수 있어야 하며 또한 테스트에 드는 비용을 줄이기 위하여는 필요한 테스트 패턴을 간단하게 구할 수 있어야 한다. 따라서 유효한 테스트 패턴을 어떻게 단순하게 구하는가가 중요하다. RAM은 자료를 저장하거나 판독할때 자료가 저장된 주소에 관계없이 거의 동일한 접근시간이 걸리는 매우 빠른 기억장치로 컴퓨터의 주기억장치로 주로 사용되며 주로 반도체 기억장치가 사용되고 있다. RAM의 내부 구조는 $m \times n$ 개의 2진 저장 소자와 각 단어를 선택하는데 필요한 번지용 논리 소자로 구성 된다. 그림 1에 RAM의 구성을 나타 내었다.

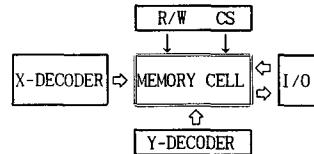


그림 1. RAM Organization

1. Pattern Sensitive 고장

소자 밀도의 증가로 메모리 셀 들이 더욱 근접함에 따라 셀 간에 전자기적 영향이나 커패시턴스, 전하 누설 등으로 인한 다른 셀 들의 간섭에 의해 정보가 소실될 수 있다. 즉 어떤 셀의 내용이 주위 다른 셀들의 상태의 조합인 어떤 패턴에 의해 바뀔 수 있다. 이러한 고장을 Pattern Sensitive Fault (PSF) 이라고 한다. 일반적인 PSF 를 검출하는 것은 많은 시간이 필요하기 때문에 메모리 설계나 역세스 방법 등을 고려하여 한 셀에 영향을 줄수 있는 다른 셀의 범위를 그림 2와 같이 주변 셀 4 개로 제한한다. 여기서 중앙의 셀을 기본 셀이라고 하고 상,하,좌,우의 주변 셀을 각각 N,E,W,S 로 표시 한다. 주변 셀 N,E,W,S 의 어떤 내용에 의해 기본 셀의 내용이 바뀌지 않거나, 바뀌는 고장을 Static PSF (SPSF) 라고 한다.[5]

주변 셀 N,E,W,S 의 패턴이 기본 셀의 내용을 무조건 바

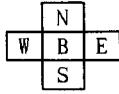


그림 2. neighbourhood of a cell.

꾸어 놓을때 그 고장을 Critical Pattern Fault (CPF) 이라 한다.[1][7] 지금까지는 메모리 테스트 패턴들은 고장모델의 정확한 정의 없이 다른 메모리 기능을 실험하다 발견적 방법으로 유도되었다. 그러나 그러한 테스트 방법들로는 근본적인 고장모델을 정의하기란 쉽지 않다. 표 1은 N bit RAM의 여러가지 테스트 방법들의 패턴수를 보여주고 있다.

표 1. Length of Test Pattern

Type of Test	Length of Test Pattern
Waking	$2N^2$
Galloping	$4N^2$
Column/Row Galloping	$3N^{3/2} + 6N$
Diagonal	$2N^{3/2} + 4N + 5N^{1/2}$
Sahgal's Test Procedure	$36N$
Thatte's Algorithm	$30N$
Galpat	$4N^2$

2. 고장모델의 정의

최근 발표된 논문들의 주요 연구방향은 고장모델을 정의한 후 그 모델을 사용하여 테스트 패턴을 생성하고 있다. 이 논문에서 제안한 고장모델은 다음 정의와 같다.

정의 1 : 어떤 Cell B 의 주변 CELL 이 B 주위 라고 할 수 있는 Cell 의 집합이라고 하면 인접한 주위의 Cell 4 개의 값은 N,E,W,S 로 한다.

정의 2 : 어떤 기본 cell B 의 주위에 있는 N,E,W,S Cell 들이 B 에 영향을 준다면 이를 PIF (Pattern Interference Fault) 이라고 말한다.

정의 3 : 기본 cell B 의 내용이 B 주위의 N,E,W,S 의 값에 의한 PIF 에 의해 고장이 발생했을 때 그 고장을 SPSF 라고 한다.

정의 4 : 만약 기본 cell B 의 내용들이 B 와 관계된 N,E,W,S 내용들이 변화할 때 고장을 일으킨다면 그 고장을 DPSF 라고 한다.[6]

이러한 정의들은 임의의 주변셀에 확대 해석 할 수 있다.[7] 패턴 NEWS=1110 이 positive 결합에 의해 기본 Cell 에 영향을 준다고 가정해 보자. Srin이 NEWS=1110 일때 마다 대응하는 기본 Cell B 는 논리 1 이 된다고 설명하였다.[4] 이러한 고장을 NPSF 라고 정의 한다. Hayes 에 의

해 사용되고 정의된 모델은 정적고장과 동적인 고장이 혼합되었다.[7] Srin이 32 N 쓰기과 32 N 읽기의 테스트 패턴을 생성 하고, 메모리의 SPSF 를 찾기 위해 다양한 조작을 했다. SUK 와 Reddy [6], hayes[7], 그리고 Seth 와 Narayanswamy[8] 에 의해 연구된 방법들도 매우 체계적이다.

RAM 의 인접한 기억소자 사이의 캐패시턴스와 전류 누설등의 원인으로 SPSF 는 발생 된다.

III. 메모리 테스트

본 연구의 목적은 RAM 의 고장을 테스트 하는데 있어 가장 간단하고 빠르게 테스트 할 수 있는 최소의 테스트 패턴을 얻는데 있다. 메모리 용량이 칩의 소자 밀도에 비례하여 증가되므로 기존의 테스트 방법인 전체검사법이나 의사-전체검사법들은 많은 시간이 걸리고 비용도 많아 비실용적이다. 이러한 테스트 시간은 너무 많이 걸리며, 고밀도의 메모리 칩을 빠르게 테스트하기 위해서는 복잡한 테스트 장비와 많은 비용이 든다. 테스트 방법으로는 Built in Self Test (BIST) 가 있는데 이 방법은 어떠한 특별한 테스트 장비도 필요치 않으며 칩 내부에 테스트 벡터를 생성할 수 있는 회로를 내장해 놓으므로 자체적으로 고장을 비교하여 검출하는 효율적인 방법이다. 그동안 값비싼 테스트 장비를 사용하는 많은 기억소자들에 BIST 를 적용할 수 있다. 과거에는 모든 테스트 방법이 단지 비용 절감을 위한 목적으로 되어있어 알고리즘의 실행을 위해서는 상당한 양의 하드웨어와 소프트웨어가 요구되었다. 그러나 최근의 연구 과제는 최소화 또는 최소화해 가까운 테스트 방법의 연구와 간단한 테스트 생성 방법에 대한 연구로 요약된다. 테스트 방법은 칩의 구조와 테스트 생성 알고리즘이 요구된다. 그러므로 이 방법을 사용한다면 테스트 생성 알고리즘은 복잡하지 않는 하드웨어에 의해 실행하는 것이 바람직하며 테스트 검증 하드웨어도 간단한 것이 좋다.

정의 5 : SWSF (static weight sensitive fault) 는 주변셀에 있는 1 의 값(weight)에 의해 기본셀의 내용을 변화시키는 고장이다.

정의 6 : t-ws 고장은 기본셀의 주변셀 news 가 1 로 할당될때 발생하는 ws 고장의 여러 가지 형태를 말한다.

정의 7 : P(N) SWSF 는 주변셀이 1(0)일 때 기본셀의 내용이 0→1 (1→0) 으로 각각 변화하는 고장이다.

정의 8 : DWSF (dynamic weight sensitive fault) 는 주변셀의 쓰기 동작중 news 의 내용이 변화될 때 기본셀의 내용이 바뀌는 고장이다.

정의 9 : P(N) DWSF 고장은 주변셀에 쓰기 동작중 기본셀의 내용이 0→1 (1→0)로 바뀌는 고장이다.

정의 10 : 테스트 패턴의 수는 읽기와 쓰기의 동작 수로 정의한다.

정의 11 : Local Lower Bound (LLB)는 메모리 내용이 모두

0 (1) 으로 부터 시작하고 0 (1) 으로 끝나는 테스트 패턴을 말한다.

IV. t-ws 고장 테스트

PSF 가 WSF 처럼 고장모델이 될 수 있다는 것은 앞에서 정의하였다. $t=0,1,2,3,4$ 일때 고장을 검출할 수 있는 테스트 집합은 메모리의 모든 PSF 를 검출할 수 있다. $P(N)$ WSF 를 검출하기 위해서는 기본 셀의 내용이 0(1), 그리고 모든 가능한 weight 패턴이 주변 셀에 포함되어야 만 한다. 이 상태가 모든 PSF 를 검출할 수 있는 상태이다. DWSF 검출도 비슷한 이론이 정해질 수도 있다. 이제 메모리 배열 내의 SWSF 를 보자. N bit 는 2 차원 배열로 구성된 $m \times n$ bit 라고 가정한다.

1. 1-ws 테스트

그림 3에 각 기본 셀은 4개의 서로 다른 1-ws 패턴으로 되어 있다. 그러므로 각 기본 셀은 네가지 패턴을 순서적으로 네번 읽어야 만 한다. 주어진 모든 1-ws 패턴은 주변 셀에 1이 꼭 쓰여져야 한다. 그러나 만약 1이 0으로 바뀌면 모든 메모리 셀은 지워지고 테스트는 끝나게 된다. 그러므로 두번의 쓰기 동작과 네번의 읽기 동작이 각 셀이 필요하므로 $4N + (8 \times N/4) = 6N$ 이 필요하다.

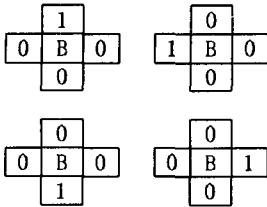


그림 3. all 1-ws pattern

* 결과 1 : 1-ws 고장 검출 LLB (local lower bound) 는 $6N$ 의 테스트 패턴에 의해 검출된다.

그림 3을 사용하여 증명하자. 모든 셀이 0으로 비워있다고 가정하고 시작하고 표 2의 순서에 의해 모든 1-wsf 는 검출된다.

표 2. 1-ws 순서

Step	Write	Read	Write
1	N→1	news	N→0
2	E→1	news	E→0
3	W→1	news	W→0
4	S→1	news	S→0

단계 5-8) 은 NEWS 와 news 를 바꾸어 반복하면 된다.

2. 2-ws 테스트

그림 4에 기본 셀은 6개의 서로 다른 2-ws 패턴으로 되어 있다. 그러므로 각 셀은 6번씩 읽어야 만 한다. 또한 weight 는 14개 이므로 2-ws 고장 테스트 패턴을 구하기 위

한 쓰기 동작의 전체 수는 $14 \times N/4$ 즉 $3.5N$ 이 되므로 읽기 와 쓰기 동작의 전체 수는 $6N + 3.5N = 9.5N$ 이다.

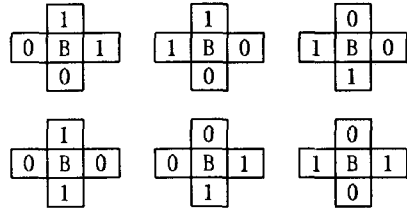


그림 4. all 2-ws pattern

* 결과 2 : 2-ws 고장 검출 LLB 는 $9.5N$ 의 테스트 패턴에 의해 검출된다.

그림 4를 사용하여 증명하자. 모든 셀이 비워져 있다고 가정하고 다음 표 3에 의해 2-wsf 는 검출된다.

표 3. 2-ws 순서

Step	Write	Read	Write
1	S, N→1	news	S→0
2	W→1	news	N→0
3	E→1	news	E→0
4	S→1	news	W→0
5	E→1	news	S→0
6	N→1	news	E, N→0

7) 에서 12) 단계는 N,E,W,S 와 n,e,w,s 를 바꾸어 반복하면 된다.

3. 3-ws 테스트

4개의 주변셀 중 3개가 1로 쓰여진 경우의 서로 다른 패턴 4개가 기본 셀 이므로 4번의 읽기 동작이 필요하다. 쓰기 동작은 셀의 값이 모두 0으로 시작하고 끝날 때 다시 모두 0으로 되어야 하며 12개의 $N/4$ 셀이 되므로 전체 동작수는 $4N + 12 \times (N/4) = 7N$ 이 필요하다.

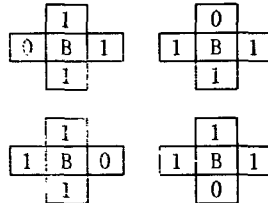


그림 5. all 3-ws pattern

* 결과 3 : 3-ws 고장 검출 LLB 는 $7N$ 의 테스트 패턴에 의해 검출된다.

그림 5를 사용하여 증명하자. 모든 셀이 비워져 있다고 가정하고 다음 표 4에 의해 모든 3-wsf 는 검출된다.

표 4. 3-ws 순서

Step	Write	Read	Write
1	E, W, S→1	news	E→0
2	N→1	news	S→0
3	E→1	news	W→0
4	S→1	news	N, E, S→0

5) 에서 8) 단계는 NEWS 를 news 로 바꾸어 반복한다.

4. 4-ws 테스트

기본 셀의 각 셀은 적어도 한번은 읽어야 한다. 주변 셀은 0 으로 쓰고 0 의 상태로 끝나도록 써야 하므로 적어도 1N 의 읽기 동작과, 2N 의 쓰기 동작이 요구된다.

A	B	A	B	
B	A	B	A	
A	B	A	1	
B	A	1		1
			1	

그림 6. all 4-ws pattern

• 결과 4 : 모든 4-ws 고장검출의 LLB 는 3N 의 테스트 패턴에 의해 검출된다.

그림 6을 이용하여 증명하자. 모든 셀이 0으로 되어 있다고 가정하고 다음 표 5에 의해 모든 4-wsf 는 검출된다.

표 5. 4-ws 순서

Step	Write	Read	Write
1	A(NEWS)→1	-	-
2	-	B(news)	-
3	-	-	A(NEWS)→0

4) 에서 6) 까지의 각 단계는 A 와 B를 바꾸어 반복한다. 읽기동작의 수는 1N 이고 쓰기 동작의 수는 8개의 N/4 셀이 되므로 전체 테스트 패턴 수는 $1N + 8 \cdot N/4 = 3N$ 이다.

V. 결론

이 논문에서는 RAM 의 테스트 방법을 개선 하기 위해 t-ws 고장을 검출하기 위한 LLB 최소 테스트 패턴에 대하여 연구하였다. 이것은 기존의 고장 테스트 방법들 보다 최소의 테스트 패턴으로서 빠르게 생성되며, 생성 과정에 있어 구조적이고 체계적인 방법을 사용하였다. 이와 같은 방법을 사용하면 N bit RAM 에서의 1-ws 고장을 검출할 수 있는 최소 테스트 패턴은 6N, 2-ws 고장에서는 9.5N, 3-ws 고장에서는 7N, 4-ws 고장에서는 3N 으로 된다. 제안된 테스트 패턴은 표1 과 비교 했을 때 대단히 감소 되었음을 알 수 있으며, 테스트 시간의 감소로 칩의 생산성과 경제성을 함께 높일 수 있다.

VI. 참고문헌

- [1] L. Wang and E. McCluskey, "Complete Feedback Shift Register Design for Built-In Self Test", Proc.of Int'l conf. Computer Aided Design, pp. 56-59, 1986.
- [2] Edward J. McCluskey, "Built-In Self Test Techniques", IEEE Design & Test, pp. 21-28, April, 1985.
- [3] Edward J. McCluskey, "Built-In Self Test Structures", IEEE Design & Test, pp. 29-36, April, 1985.
- [4] Srin, V. P., "API Tests for RAM Chips", IEEE Computer, Vol, 10, No. 7, July 1977, pp. 32-36.
- [5] Suk, D. S. and Reddy, S. M., "Test Procedures for a class of Pattern-Sensitive faults in Semiconductor Random-Access Memories", IEEE Trans. Comp., Vol. c-29, No. 6, June 1980, pp. 419-429.
- [6] Hayes, J. P., "Detection of Pattern-Sensitive Faults in Random-access Memories", IEEE Trans. Comp., Vol. C-24, No. 2, February 1975, pp. 150-157.
- [7] Hayes, J. P., "Testing Memories for Single Cell Pattern-Sensitive Faults", IEEE Trans Comp., Vol. c-29, No. 3, March 1980, pp. 249-254.
- [8] Seth, S. C and Narayanswamy, k, "A Graph Model for Pattern-Sensitive Faults in Random-Access Memories", IEEE Trans. Comput., Vol. C-30, December 1981, pp. 973-977.