

반도체 메모리의 Built-In Self Test (BIST)

趙 相 福

蔚山大學校 電子工學科

I. 서 론

반도체 기술의 급격한 발전으로 메모리의 집적도가 높아지고 있다. 현재 세계 반도체 시장 확보를 위해 세계 유수의 기업들이 활발한 경쟁을 벌이고 있으며, 메모리의 집적도를 높이기 위해 막대한 투자를 하고 있는 실정이며, 국내 반도체 회사에서도 세계 반도체 시장 확보를 위해 활발한 개발사업을 벌이고 있다. 이와 같이 메모리 집적도가 높아지고 있는 현재, 메모리를 테스트하는데 엄청난 경비가 소요될 뿐만 아니라 메모리 테스트 시간 또한 집적도 증가에 비례하여 증가하고 있으며, 메모리에서 발생 가능한 새로운 형태의 고장들이 생겨나고 있으며, 이에 대한 연구가 행해지고 있다. 메모리 테스트시 외부장비에 의해 테스트를 행하는 경우 상당한 테스트 시간이 요구되어 테스트 비용이 엄청나게 소요된다. 따라서 최근에는 테스트 회로를 내장한 built-in self test(BIST) 방식이 등장하여 테스트 비용과 시간을 절약하게 되었다.^[1-3] 한 예로 GALPAT사에서 n-bit RAM 에 대해 $4n^2+4n$ 의 테스트 복잡도 (complexity)를 가지는 방식으로 4M-bit RAM을 테스트 할 경우 (200nsec cycle time)에 162일이나 소요되었다.^[4] 또한 GALPAT 사에서 사용한 테스트 방법은 테스트 engineer의 경험에 의한 테스트 방법을 사용했기 때문에 대용량의 메모리에 적용하기에는 부적당한 방법들이었다. 최근 line mode 테스트, micro-programmable ROM을 이용한 테스트, ECC(error checking and correcting)등 여러가지 테스트 방식들이 제안되고 있다. ECC는 신뢰도가 가장 높은 장점을 가지나 부가회로 면적이 높다는 단점을 가지며 위의 방식들 모

두가 테스트 속도, 고장검출 범위, 부가회로 면적의 세 가지 조건을 모두 만족시키지는 못한다.

메모리 테스트시 고려되어지는 고장 모델은 stuck-at 고장, transition 고장, coupling 고장 및 PSF(pattern sensitive fault) 등이 있다. Coupling 고장이나 PSF의 검출은 어려우며 그 범위 또한 방대하다. 더욱이 메모리 집적도가 증가함에 따라 이러한 고장의 발생가능성은 더욱 증가하고 있으므로, 이들의 고장검출이 필수적이다. 그러나 이러한 고장은 각각 그 종류가 많고 복잡하며 완전히 이 고장들을 검출한다는 것도 불가능하다.^[5-10] 따라서 제한된 범위내에서 테스트 되어지는 것이 일반적이다.

본고에서는 첫째로 BIST를 중심으로 메모리 테스트에 적용된 방식들에 대해 살펴보고, 산업체 및 학계에서 발표한 방식들에 대한 연구동향을 분석해 보았다. 둘째로 메모리에서 발생빈도가 높아지고 있는 PSF 모델에 대한 BIST 방식을 소개하였으며 마지막으로 결론을 맺기로 한다.

II. 메모리 테스트에 적용된 방식 및 분석

초고집적 메모리 구성방식의 최신 경향은 메모리에서 필수적으로 고려되어야 하는 요소인 스피드, 고밀도, 전력소비, S/N비에 바탕을 두고 개발 되고있다. 또한 메모리의 테스트에서도 메모리의 용량이 증가함에 따라 테스트 시간의 단축과 고장범위를 고려하여 꾸준히 연구되고 있다. 본 절에서는 테스트와 관련한 메모리의 구성방식 및 동작특성에 관한 연구결과를 학계에 발표된 문헌을 참고로 하여 기술하였다. 표 1에 발표된 초

표 1. 발표된 초고집적 메모리의 주요특성

Maker 특징	Mitsubish LSI R & D Center	NEC	Toshiba	Matsushita (Semiconductor Research Center)
테스트 방식	Line mode test (1024 bit), Multi-bit-test (16 bit)	BIST with micro- programmable ROM, 16 bit parallel test	16bit parallel test	16bit parallel test
구 성 방 식	16M words×1b/ 4M words×4b (Metal mask op.)	16M words×1b/ 4M words×4b (Metal mask op.)	16M words×1b/ 4M words×4b (Bonding option)	16M words×1b/ 4M words×4b
Function mode	Fast page/ Static column page/ Nibble/ Serial	Fast page/ Static column page/ Nibble	Fast page/ Static column page/ Nibble mode (Bonding option)	Static column
Access time	$t_{RAC}=60ns$ $t_{CAA}=20ns, 15ns$	$t_{RAC}=55ns$	$t_{RAC}=45ns$	$t_{RAC}=65ns$ $t_{CAA}=35ns$
Memory cell구조	T-shaped Stacked capacitor	Stacked capacitor	Stacked trench capacitor	Trench capacitor, Surrounding high- capacitor cell
발표된 학술지명	ISSCC '89 pp.244-245	ISSCC '89 pp.246-247	ISSCC '89 pp.248-249	IEE J.SSC, Oct. 1988 pp.1104-1111
Design 룰	0.5 μm	0.55 μm	0.6 μm	0.5 μm

Maker 특징	Oki	Hitachi	IBM (General tech. division)	Mitsubish (LSI R & D Center)	NTT
테스트 방식	16bit parallel test	8bit parallel test	Error check. & correct.(ECC)	ECC Technique	ECC
구 성 방 식	16M words×1b/ 4M words×4b/ 2M words×8b	4M words×1b/ 1M words×4b (Bonding option)	16M words×1b	16M words×1b	2M words×8b
Function mode	Fast page/ Static column/ Serial(×1,×4,×8)	Static column page/ Fast page/ Nibble/ Byte (Metal mask op.)	Fast page/ Static column	Fast page/ Static column	—
Access time	$t_{RAC}=60ns$	$t_{RAC}=65ns$ $t_{CAA}=30ns$	$t_{RAC}=50ns$	$t_{RAC}=45ns$ (simulated)	80ns
Memory cell구조	Buried stacked capacitor	Stacked capacitor	Trench capacitor	— —	Isolation mer- ged vertical capacitor
발표된 학술지명	IEEE J. SSC, Oct. 1989 pp.1176-1183	ISSCC '87 pp.18-19	ISSCC '90 pp.232-233	IEEE J. SSC, Feb.1990 pp.11-17	ISSCC '87 pp.22-23
Design 룰	0.55 μm	0.8 μm	0.5 μm	—	0.7 μm

표 2. 년도별 scheme 비교분석

년 도	테스트 방식	area overhead				algorithm	테스트 복잡도	게재지명	발표자
		64K	256K	1M	2M 4M				
85	Signature Analyzer	1.8-2.9%				Marching	O(N)	Proc.Int'l Test Conf., Comp. Society pp.462-470	Sridhar
87	On Chip Compact Test	1.21% (SRAM)				SPSF	O(N)	IEEE Design & Test of Comp. vol. 4, no.1, Feb. pp.42-51	Saluja, Sug and Kinoshita
	Parallel Test	0.32% (SRAM)			1% (DRAM)	Marching		Proc.Int'l Test Conf., Comp. Society Press no.798 pp.1078-1084	Inoue et al.
	Built-In Processor	0.09% (SRAM)				Not specified		Proc.Int'l Test Conf., Comp. Society Press no.798 pp.1078-1084	Muler and Ritter
89	Parallel Test	0.4% (RAM)				PSF	0(n ^{1/2})	IEEE Trans. Comp. vol.38, no.3,pp.394-407	Mazumder and Patel
90	Row/Column Weigh-sensitive				0.8% (RAM)	Row/Column PSF test	O(n ^{3/2})	IEEE J.Solid State Circuits vol.25,no.2. pp.514-524	M.Franklin, K.K.Saluza and K.Kinoshita
	Micro Program ROM	1% (for 16M bit DRAM)				Marchng	O(N)	IEEE J.Solid State Circuits vol.25,no.4, pp.903-911	T.Takeshima et al.

고집적 메모리의 주요 특성을 테스트 방법, 구성 방식, 동작특성등을 고려하여 각 회사별로 나타내었다. 또한 표2는 학계 및 기업체에서 발표한 BIST scheme에 대해 년도별로 비교 분석하였다.

1. Line Mode Test

Mitsubishi사는 ISSCC '89에 기존의 16-bit parallel 테스트 모드와 결합하여 LMT(line mode test)방식을 발표하였다.^[1] 이는 MPR(multi purpose register)과 비

교기(comparator)가 각 메모리의 main I/O 와 sub I/O 사이에 위치하여 1024-bit의 메모리 셀을 한꺼번에 테스트하는 방식이다. 그림 1에 MPR과 비교기를 나타내었다.

그림 2의 (a)는 LMT시에 1024개의 메모리 셀에 write 동작이 수행됨을 나타내며, (b)는 (a)의 write 동작후에 메모리 셀에 저장되었던 값을 line read하여 기대한 값과 동일한지 여부를 판별하여 동일하지 않을 때는 error flag 신호가 발생하는 원리를 보여주고 있다.

그림 3 에는 발표한 16-M DRAM에 적용된 LMT의 흐름도를 나타내었다. 첫번째로 random 테스트 데이터가 MPR에 세트된다. 그리고 난후, 한 사이클에서 선택되어진 하나의 워드라인에 연결 되어있는 1024 비트를 동시에 write 동작을 행한다. 모든 메모리셀에 대한 random 테스트 data의 write 동작은 16K 사이클 동안에 완료된다. 최종적으로 16K read 사이클 안에서 기대된 패턴과 메모리 셀(각 워드라인에 대해 1024 비트)에 저장된 random 테스트 데이터와 비교한 후 테스트

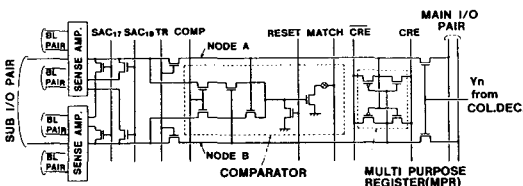
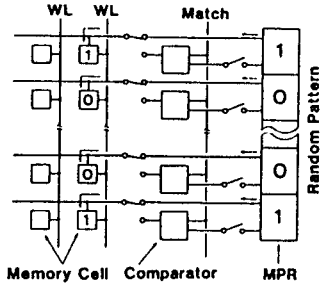
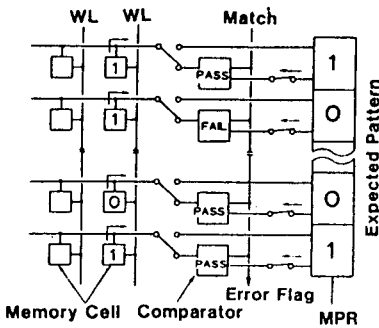


그림 1. MPR과 비교기



(a) LMT에서의 Write 동작



(b) Line read 후의 parallel compare 원리

그림 2.

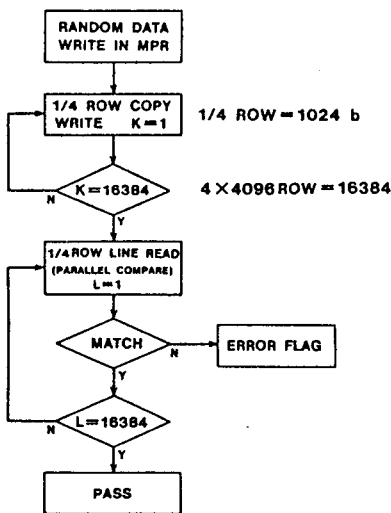


그림 3. LMT의 흐름도

트 동작을 완료한다.

이 LMT의 테스트 패턴은 MPR에 의해 기존에 사용된 "0" 또는 "1"의 제한된 값 외에도 flexible 테스트 패턴을 인가할 수 있는 특징을 가지고 있다. 또한 area overhead는 MPR과 비교기가 효과적으로 레이아웃 되므로 0.5%에 지나지 않는다.

2. Micro-Programmable ROM을 이용한 Test 방식

NEC사는 16-bit 병렬 테스트(parallel test)와 칩 내부에서 마이크로 프로그래밍(micro-programming)이 가능한 ROM을 내장한 self-test 방식의 16-Mbit DRAM을 발표하였다.^[2] 그림 4는 BIST(built-in self test) 회로를 내장한 DRAM의 블록 다이어그램을 보여주고 있다. 메모리를 구성하고 있는 기존의 회로에 self-test 회로를 구동시키는 테스트 클럭 발생기(test clock generator), ROM에 어드레스를 지시하는 프로그램 카운터(program counter), 테스트 되어지는 메모리 셀의 어드레스를 지시하는 어드레스 카운터(address counter), 메모리 셀에 인가될 테스트 데이터를 발생시키는 데이터 발생기(data generator), 그리고 데이터 발생기에서 출력된 테스트 데이터를 메모리 셀에 write 한 후, 저장되어 있던 값을 read하여 비교하는 기능을 하는 데이터 비교기(data comparator)가 부가되어 있다.

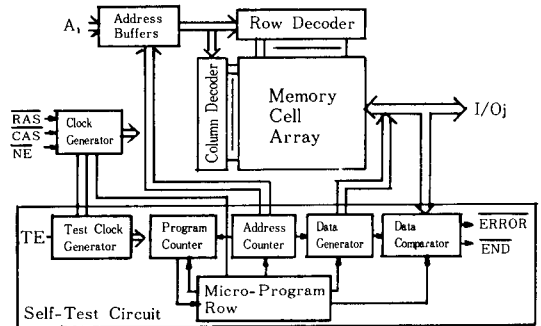


그림 4. 마이크로 프로그래밍 가능한 ROM을 내장한 BIST DRAM의 블록 다이어그램

Microinstruction과 micro-code에 의한 테스트 프로시쥬어를 그림5에 보이고 있다. (a)는 본 연구진에 의해 1차년도에 수행되었던 형태의 marching 테스트에 대한 프로시쥬어이며, (b)는 기존의 checkerboard 테스트의 프로시쥬어를 나타내고 있다. YX는 메모리 셀의 어드레스이며 PC는 ROM의 어드레스이다. 각 스텝에

ROM ADDRESS	MICRO-CODED TEST PROCEDURE
0) Clear	000000010
1) W(0). Inc YX. IF YX=MAX THEN Inc PC.	1010000100
2) R(0).	0000001000
3) W(1). Inc YX. IF YX=MAX THEN Inc PC ELSE Dec PC.	1110010100
4) R(1).	0000011000
5) W(0). Inc YX. IF YX=MAX THEN Inc PC ELSE Dec PC.	1110000100
6) Dec YX.	0001000000
7) R(0).	0000001000
8) W(1). Dec YX. IF YX=0 THEN Inc PC ELSE Dec PC.	1101010100
9) R(1).	0000011000
10) W(0). Dec YX. IF YX=0 THEN Inc PC ELSE Dec PC.	1101000100
11) Halt.	0000000001

(a) Marching 테스트

12) Clear. EXOR D.	0000100010
13) W(0). EXOR D. Inc YX. AVD IF YX=MAX.	1010100100
14) R(0). EXOR D. Inc YX. AVD IF YX=MAX.	1010101000
15) W(1). EXOR D. Inc YX. AVD IF YX=MAX.	1010110100
16) R(1). EXOR D. Inc YX. AVD IF YX=MAX.	1010111000
17) Halt.	0000000001

(b) Checkerboard 테스트

그림 5. Microinstruction과 microcode에 의한 테스트 프로시쥬어

해당되는 operation 코드의 한 워드는 10-bit로 구성되어 있다.

3. Parallel 테스트

메모리의 용량이 증가할 수록 이에 대한 테스트시간도 증가하게 된다. 단순한 쓰기과 읽기 동작인 경우 $2N \cdot t$ (N :memory density, t :cycle time)의 시간이 되며, 간단한 Galloping pattern을 인가하는 경우 $2N^{3/2} \cdot t$ 테스트 시간을 필요로 한다. 이에 반해 복수개의 메모리 셀에 지정한 값을 쓰고 읽어내는 테스트 동작을 수행하게 되면 테스트 시간이 크게 줄게 된다. 현재 각社에서 발표되는 초고집적 메모리의 테스트 방식은 기존의 1-M DRAM과 4-M DRAM의 parallel 테스트 방식에 기초를 두고 있다.

1) Multi-bit-test 방식

85년 Mitsubishi사는 1-M bit DRAM에 multi-bit test mode(MBT)를 적용하였다.^[11] 그림 6에는 이에 대한 블럭 다이어그램을 보여주고 있다. 이 MBT 모드는 TE 단자에 high 레벨을 인가하므로써 구동되게 된다. D_{in} 단자의 입력 데이터는 최상위 비트 어드레스(RA_9, CA_9)만 다른 row와 column 어드레스를 갖는 메모리 셀에 쓰여지게 된다. 즉 4개로 분할된 subarray에(256K) 최상위 비트만이 다른 4개의 메모리 셀에 동시에 데이터를 write 동작을 수행하고 이를 다시 read하여 4개의 메모리 셀에 저장된 값이 동일 한지 여부를 판별하게

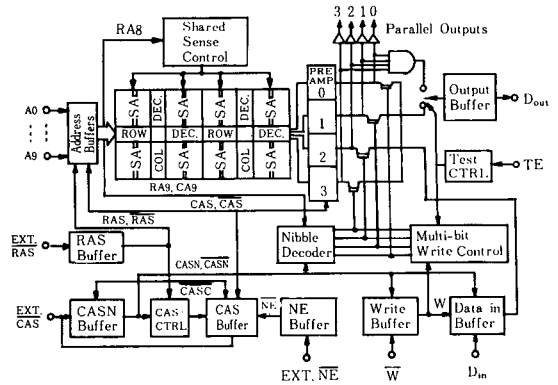


그림 6. MBT(multi-bit-test)모드를 갖는 DRAM의 블럭 다이어그램

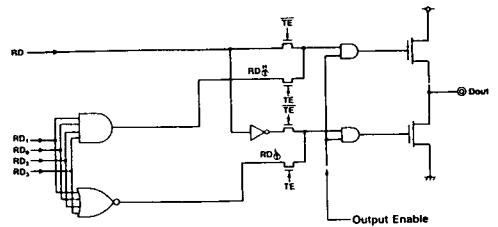


그림 7. MBT의 회로도

되는 것이다. 만일 읽어낸 4개의 데이터가 동일하지 않다면 메모리 셀이 고장(fail)이란 상태를 표시하게 된다.

MBT 회로를 그림 7에 나타내었다. 테스트 모드에서는 D_{out} 단자의 상태가 high-impedance 인지를 판별하므로써 테스트 대상이 된 메모리셀이 고장인지를 알게 된다. 읽어낸 값이 동일하지 않으면 D_{out} 이 high-impedance 상태가 된다.

2) Texas Instru.의 16-bit parallel 테스트

Texas Instrument사는 4-Mb DRAM($1M \times 4bit$)을 발표하였다.^[8] 여기에서는 16-bit parallel 테스트 모드를 도입하여 256-Kb DRAM이 테스트 되어지는 것과 같은 테스트 시간을 얻었다.

테스트 모드에서는 4개의 메모리 sub-array가 여기(active)되어, 이들 각 어레이가 동시에 4개의 비트를 어드레싱 하도록 디자인하여 내부적으로는 16-bit가 paralle 테스트를 위해 동시에 액세스 가능하게 하였다. 그림 8에는 paralle read와 write 동작을 하는 기능 다이어그램을 나타내었다.

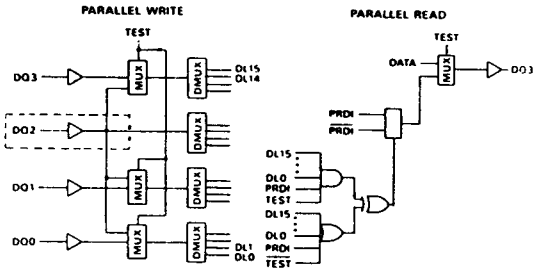


그림 8. 16-bit parallel 테스트의 기능 다이어그램

입력부 DQ2에서 출력된 데이터는 parallel write 동작 동안에 16개의 내부 데이터 라인을 구동시킨다. 이들 데이터 라인들은 parallel read 동작 동안에 기대되는 값과 동일하지 여부를 비교하게 된다. 만일 모든 데이터 라인들의 값이 기대되는 값과 동일하면 기대되는 값이 DQ3에 출력되어 되고, 동일하지 않다면 DQ3에 기대되는 값의 보수의 값이 출력되게 된다.

3) 8-bit parallel 테스트

그림 9에는 Mitsubishi사에서 1987년에 발표한 90ns 4M-bit DRAM에 내장된 테스트 모드 동작을 위한 회로를 나타내었다.^[12] 일반적인 read/write 동작에서는, 클럭 ϕ_T 가 I/O 버스와 각 2-Mbit 블록을 연결하는 트랜스퍼 게이트를 turn on 시킨다. 그렇게한 다음에 normal preamplifier (PA₁-PA₄) 4개가 동작하게 되고 이에 적절하게 $\times 1$ 또는 $\times 4$ 데이터가 읽혀지게 된다. 테스트 모드에서의 read 동작은 클럭 ϕ_T 가 4개의 I/O 버스라인 쌍(bit-line pair)을 차단하고, 각 512K 메모리 subarray에서 나온 8 비트의 데이터가 4개의 normal preamplifier와 test preamplifier(PA₅-PA₈)에 의해 동시에 증폭된다. PA₅-PA₈ 를 거치고 난 후의 증폭된 결과는 논리적으로 두개의 여분의 버스라인을 통

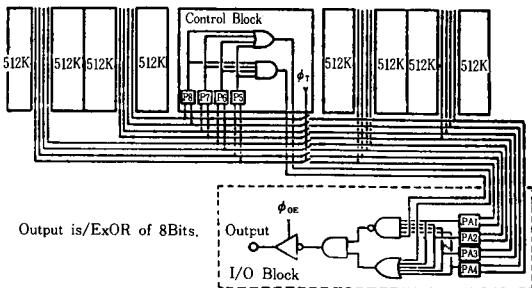
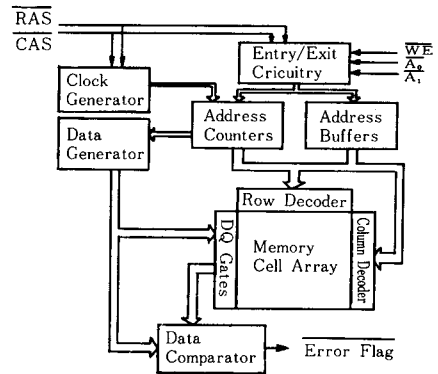


그림 9. Memory subarray와 테스트 회로

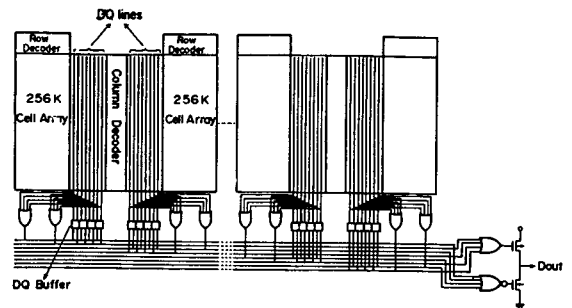
하여 I/O block에 전달된다. 그리고 난후 I/O 블록에서 8 비트의 데이터가 exclusive-OR 된다. 8개의 데이터 비트가 동일하지 않을때는 'low'가 출력되며, 동일할때는 'high'가 된다. 테스트시의 write 동작은 ϕ_T 가 turn on 되어 8개의 데이터가 각각의 512K subarray 에 있는 8개의 메모리 셀에 동시에 쓰여지게 된다.

4) Toshiba의 16-bit parallel 테스트

ISSCC '87에 Toshiba사에서는 self 테스트 동작이 가능한 4-Mb DRAM을 발표하였다.^[13] 그림 10의 (a)에서는 기존의 RAM에 컬럼 어드레스 카운터(column address counter), 테스트 패턴 발생기(test pattern generator) 그리고 데이터 비교기(data comparator)가 부가된 것을 보이고있다. (b)에서는 테스트 회로의 구성도를 보이고 있다.



(a) Self-test 기능이 내장된 DRAM의 블록 다이어그램



(b) 테스트 회로의 구성도

그림 10.

III. Pattern Sensitive Faul(PSF)테스트

본절에서는 메모리의 집적도가 증가함에 따라 고장

확률이 증가하며, 고장 검출이 필수적인 NPSF (neighborhood PSF)를 검출가능한 BIST scheme을 소개한다. 메모리 셀과 셀사이에서 전자기적인 영향으로 전류 누설등에 기인한 고장을 PSF라 한다. 이러한 고장은 메모리셀 전체에 대해서 고려해야 하므로 그 종류가 복잡하여 실제적인 고장모델로서는 부적당하며 이를 검출하기도 어렵고 상당한 테스트 시간을 요구하여 테스트 비용이 증가하므로 실제로 가장 많이 발생하는 제한된 PSF(restricted PSF or NPSF)에 대해 고려하는 것이 효과적이다.

1. NPSF 모델

NPSF를 검출하기 위해 테스트하는 셀을 기본셀 (base cell)이라하고, 기본셀을 포함한 이웃셀을 neighborhood cell이라 하며, 기본셀이 제외된 이웃셀을 deleted neighborhood cell이라고 한다.

NPSF의 고장 모델은 아래와 같다.

(1) Active NPSF (ANPSF) or dynamic NPSF

이웃셀의 내용변화로 인해 기본셀의 내용이 변화되는 고장을 말한다. 이러한 고장을 검출하기 위해서는 deleted neighborhood cell의 내용을 변화시키는 모든 가능한 pattern에서 각 셀을 0 과 1 상태에서 읽어야 한다.

(2) Passive NPSF (PNPSF)

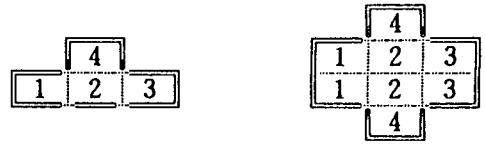
이웃셀의 어떤 특정한 패턴으로 인해 기본셀의 내용을 변화시켰음에도 불구하고, 기본셀의 내용이 변화하지 않는 고장을 말한다. 이러한 고장을 검출하기 위해서는 deleted neighborhood cell의 모든 pattern에 대해 각 셀의 "0" 과 "1" 의 상태에서 쓰고 읽어야 한다.

(3) Static NPSF (SNPSF)

이웃셀의 어떤 특정한 패턴으로 인해 기본셀의 내용이 어떤 특정한 값으로 되는 고장을 말한다. 이러한 고장을 검출하기 위해서는 deleted neighborhood cell의 모든 pattern에 대해 각 셀을 0 과 1 상태에서 각각 읽어야 한다.

2. 고장 검출 알고리즘

NPSF를 검출하기 위해 전체의 메모리셀을 일정한 형태로 tiling 하고 나서 메모리셀의 tile 내의 지정된 셀 번지에 따라 Eulerian path에 의해 발생한 테스트 패턴을 write한다. 그림 11에 tiling 형태를 나타내었다. 그림 11과 같은 tile 형태를 전체 메모리셀에 적용하면 그림 12의 set A와 같으며 NPSF 검출효율을 높이기



(a) 기본 tile 모양 (b) 고장강도를 높인 tile모양

그림 11. NPSF 검출을 위한 tile 형태

Set A

Word line	Bit line							
	1	2	3	4	5	6	7	8
1	1	2	3	4	1	2	3	4
2	1	2	3	4	1	2	3	4
3	3	4	1	2	3	4	1	2
4	3	4	1	2	3	4	1	2
5	1	2	3	4	1	2	3	4
6	1	2	3	4	1	2	3	4
7	3	4	1	2	3	4	1	2
8	3	4	1	2	3	4	1	2

Set B

Word line	Bit line							
	1	2	3	4	5	6	7	8
1	1	2	3	4	1	2	3	4
2	3	4	1	2	3	4	1	2
3	3	4	1	2	3	4	1	2
4	1	2	3	4	1	2	3	4
5	1	2	3	4	1	2	3	4
6	3	4	1	2	3	4	1	2
7	3	4	1	2	3	4	1	2
8	1	2	3	4	1	2	3	4

그림 12. Tiling 된 8x8 메모리 셀 어레이

위해 전체 메모리셀의 tile을 아래로 한 line씩 shift하여 똑같은 test 동작을 수행한 것이 set B 이다.

이러한 set A, B에 대해 Eulerian path를 적용하여 2x4x2' (=128)개의 테스트 패턴을 메모리셀의 tile에 인가한다.

이에 대한 알고리즘을 다음에 나타내었다.

<<알고리즘>>

- ㉠ 첫번째 테스트 패턴을 발생하여 메모리셀 전체의 tile에 write
- ㉡ tile내의 셀번지 1에 대해
- ㉢ 테스트 패턴을 write한다.
- ㉣ 모든 셀을 읽어본다.
- ㉤ 4개의 tile에 대한 ㉢-㉣까지를 수행 완료 했으면 다음 test 패턴을 발생하고 ㉡으로 간다. 그렇지 않으면 tile번지를 1증가시키고 ㉢-㉣을 반복수행한다.

㉔ 128개의 패턴을 모두 발생했으면 테스트를 종료한다. 그렇지 않으면 ㉒-㉔을 반복 수행한다.

3. 고장 검출범위

가) SNPS;

그림 11의 tile내의 2번 cell에 테스트 패턴을 write 한 후 read해 봄으로써 기본셀에 쓰기 동작이 수행 안 되는 고장을 검출 할 수 있다. 모든 tiling 번지의 셀에 대해 위와 같은 방법으로 SNPSF를 검출할 수 있다.

나) PNPSF

가)의 방식으로 PNPSF 또한 검출할 수 있다.

다) DNPSF 혹은 ANPSF

2번셀에 write후 모든 tiling 번지의 셀을 read 해 봄으로써 기본셀의 변화로 인한 주변셀의 고장을 검출할 수 있다.

라) Stuck-at고장 및 transition 고장은 위의 고장을 테스트 함으로써 cover 가능하다.

마) PSF는 coupling 고장의 확장된 형태이므로 이에 대한 언급은 하지 않는다.

4. BIST Scheme

NPSF 테스트 알고리즘을 효과적으로 수행하기 위한 BIST scheme의 블럭다이어그램은 그림 13과 같다. 이 scheme에는 기존의 RAM에 부가적으로 어드레스 생성기(address generator logic:AGL),테스트 패턴 발생기(test pattern generator:TPG) 와 병렬 writer(parallel writer:PW) 그리고 에러 검출단(error

detector)이 필요하게 된다. 전체 BIST RAM의 블럭다이어그램 및 전체 회로도를 그림 13에 나타내었다. 그림 13은 #P개의 page로 구성된 메모리에 대해 제안한 BIST scheme의 구성도를 나타낸 것이다.

테스트 신호가 인가되면 클럭발생기(clock generator)에 의해서 열 선택기(column selector)와 행 선택기(row selector)를 제어한다. 동시에 테스트 패턴을 발생하는 기본 클럭을 인가하게 된다.

IV. 결 론

전술한 바와 같이 메모리 테스트에 적용된 여러가지 BIST 방식들에 대해 살펴 보았다. 현재의 메모리 개발 추세에 부응하여 외부 장비에 의한 테스트 보다 BIST에 의한 메모리 테스트가 필수적이라 생각된다. 또한 메모리의 테스트 시간을 줄이기 위해 병렬 테스트 방식을 취하지 않을 수 없으리라 생각된다. 그러나 병렬테스트시 발생하는 실질적인 문제는 본고에서는 생략하였으나, 각 테스트 알고리즘에 따라서 병렬테스트 방법을 적용할 수도 있고 그렇지 않을 수도 있다. 더 나아가서는 병렬테스트시 실제 메모리 상에서 발생하는 문제점들이 엄연히 존재하지만, 이러한 문제점들을 해결하기 위해 실제공정 기술진과 테스트 연구진의 공동연구가 이루어져야 할 것이며, 메모리 테스트 시간 및 비용의 축소를 위해서는 병렬테스트를 적용하여야 할것으로 기대된다.

국내 반도체 회사에서는 메모리 테스트의 BIST화에 대한 관심을 기울리해서는 안되리라 생각되며, 메모리의 BIST에 대한 연구가 활성화되면, 관련 분야 (규칙 구조를 가지는 array 회로)에 대한 파급효과가 클 것으로 기대된다. 또한 모든 system이 one ship 화 되어 가는 현재 추세를 감안할 때, 테스트 회로의 내장에 대한 연구는 필수적이라 생각된다.

參 考 文 獻

[1] S.Fuji ,M.Ogihara, M.Shimizu, et al., "A 45-ns 16-Mbit DRAM with triple-well structure," *J. Solid-State Circuits*, vol.SC-24, pp.1170-1175, Oct. 1989.
 [2] T.Mano, T.Matsumura, J.Yamada, et al., "Circuit Technologies for 16Mb DRAMs," *ISSCC Digest of Technical Papers*, pp.22-23, 1987.

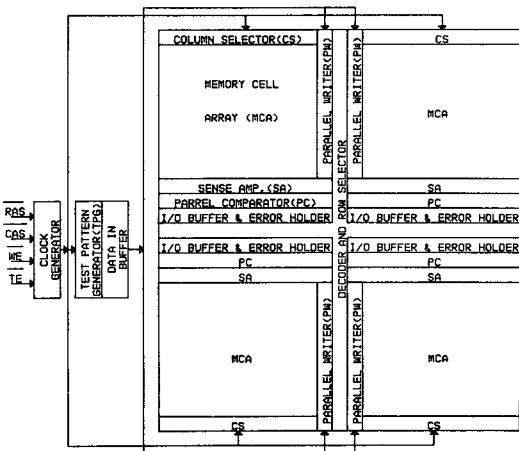

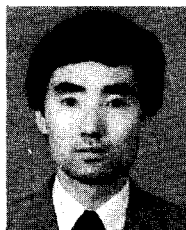


그림 13. BIST RAM의 전체 블럭 다이어그램

- [3] K.Shimohigashi, K.Kimura, Y.Sakai, et al., "A 65ns CMOS DRAM with a Twisted Driveline Sense Amplifier," ISSCC Digest of Technical Papers, pp.18-19, 1987.
- [4] K.Arimoto, K.Fujishima, Y.Matsuda, et al., "A 60ns 3.3V 16Mb DRAM," ISSCC Digest of Technical Papers, pp.244-245, 1989.
- [5] D.S. Suk and S.M. Reddy, "A march test for functional faults in semiconductor random access memories," *IEEE Trans. on Comput.*, 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 Comput.*, vol.C-34, no.2, pp.110-116, Feb.1985.
- [7] A.H.Shah, CP. Wang, R.H.Womack, et al., "A 4-Mbit DRAM with trench-transistor cell," *J. Solid-State Circuits*, vol.SC-21, pp.618-626, Oct. 1986.
- [8] P.D. Jong and AD.J. Van De Goor, "Test pattern generation for API faults in RAM," *IEEE Trans. on Comput.*, vol. C-37, no.11, pp.1426-1428, Nov. 1988.
- [9] A.J. Van De Goor and C.A. Verruit, "An overview of deterministic functional RAM chip testing", *ACM Computing Surveys*, vol. 22, no.1, pp. 5-33, Mar. 1990.
- [10] J.M Yeo, and et. al., "Algorithms for the detection of coupling faults in RAMs" *Proceedings of KITE Fall Conference*, vol 14, no.2, pp. 594-597, Sep. 1991.
- [11] M.Kumanoya, K.Fujishima, H.Miyatake, et al., "A reliable 1-Mbit DRAM with a multi-bit-test-mode," *J. Solid-State Circuits*, vol.SC-20, pp.909-913, Oct. 1985.
- [12] K.Mashiko, M.Nagatomo, K.Arimoto, "A 90ns 4Mb DRAM in a 300 mil DIP," ISSCC Digest of Technical Papers, pp.12-13, 1987.
- [13] T.Ohsawa, T.Furuyama, Y.Watanabe, "A 60ns CMOS DRAM with Built-In Self Test," ISSCC Digest of Technical Papers, p.286-287, 1987. 

筆者紹介



趙相福

1955年 6月 10日生

1979年 2月 한양대학교 공과대학 전자공학과 (학사)

1981年 2月 한양대학교 대학원 전자공학과 (석사)

1985年 2月 한양대학교 대학원 전자공학과 (박사)

1986年 3年 ~ 현재 울산대학교 전자공학과 부교수

주관심 분야: 초고집적 RAM의 테스트 및 Testable Design,
VLSI/ULSI 설계 및 테스트, ASIC 설계 및 테스트