

## 메모리 테스트를 위한 BIST 기술

金大翊, 裴晟桓, 李昌基\*,  
李相泰, 田炳實

全北大學校 電子工學科  
西南大學校 電算情報學科\*

### I. 서 론

반도체 메모리 제조기술의 급속한 발전은 단일 칩당 비트수를 가속화시키면서 기억용량을 2~3년 주기마다 거의 4배씩 증가시키고 있다. 1Mbit 이상의 용량을 갖는 VLSI 메모리의 경우, 고집적화를 위한 미세화 패턴 공정으로 새로운 형태의 고장들이 발생되고, 비트 결합들의 상호 관계를 고려할 때 테스트 시간이 기하 급수적으로 증가하게 된다.<sup>[1]</sup> 일반적으로 테스트에 소비되는 시간은 제조 비용과 밀접한 관계를 갖고 있다. 즉, 테스트 시간이 증가할수록 그에 소비되는 비용이 증가하게 되어 비용절감에 커다란 장애 요인이 되고 있다. 이 문제를 해결할 수 있는 방안 중에 하나가 BIST (Built-In Self-Test) 기법이다.

BIST 기법의 기본 개념은 하드웨어가 그 자신을 스스로 테스트하도록 하는 것이다. 즉, 회로 자신의 테스트 가능성을 높이는데 그 목적이 있다. BIST 개념은 조합 회로에서 시작되었는데, 그 후 메모리(Random-Access Memory, Read-Only Memory, Programmable Logic Array 등)를 테스트하는데 응용하게 되었다.

메모리 테스트의 초기 단계에서는 ad hoc 방식<sup>[2]</sup>을 도입하였으며 이 방식의 고장 검출 범위는 제한적이었고 결정성이 자주 결여되었다. 이러한 결점을 해결하기 위해 고착 고장, 디코더 고장, 결합 고장, 패턴 감응 고장 등의 고장 모델이 소개되었다. 메모리 셀들간의 복잡한 관계성을 보완할 수 있는 고장 모델은 검출할 수 있는 테스트의 길이가 너무 커지기때문에 현실성 있는 테스트 길이를 갖도록 고장 모델을 단순화시켜왔다. 또한, 외부 테스트 장비를 이용한 기존의 테스트 환경에서 대용량 메모리의 실질적인 테스트 알고리즘의 길이는 메모리의 용량 N과 선형적인 관계를 갖고 있어야 한다고 생각하였다. 그러나 고집적 메모리에 있어서 근접 셀들 간의 간섭으로 인한 여러가지 다양한 물리적인 고장들의 효과적인 검출을 위해서 좀더 복잡한 고장 모델이 요구되어진다.

BIST 환경에서는 상대적으로 값싼 내부 테스트

장비가 칩을 동작시켜 테스트 신호를 조정하고, 칩의 상태를 비교하고 검사하여 메모리의 기능적 테스트를 수행함으로써 테스트 비용을 절감시키고 높은 고장 검출율을 갖는 복잡한 테스트 알고리즘이 적용될 수 있다. 10MHz 클럭을 사용하여 테스트 할 경우, 메모리 크기와 테스트 알고리즘의 길이에 따른 테스트 시간은 표 1과 같다. 표 1에서

볼 수 있듯이 대용량 메모리에서  $O(N^2)$ 의 길이를 갖는 테스트 알고리즘은 테스트에 요구되는 시간이 너무 크기때문에 사용할 수 없게 된다. 그러나, 테스트 길이가  $O(N^{3/2})$ 인 경우에는 메모리의 내부 구조를 얼마만큼 반영하여 BIST 회로를 설계하였는지에 따라 대용량 메모리 테스트에 적용시킬 수 있다.<sup>[3]</sup>

(표 1) 메모리 용량과 테스트 알고리즘의 길이에 따른 테스트 시간(클럭 : 10MHz)

테스트 길이 \ 메모리 용량	1 Mbit	4 Mbit	16 Mbit	64 Mbit	256 Mbit
N	0.1 초	0.4 초	1.6 초	6.4 초	25.6 초
$N \log_2 N$	2.1 초	8.8 초	38.4 초	2.8 분	11.9 분
$N^{3/2}$	1.7 분	13.3 분	1.8 시간	14.2 시간	4.7 일
$N^2$	1.2 일	18.5 일	9.9 개월	13 년	207.8 년

메모리 용량이 증가됨으로서 외부 테스트 장비를 사용할때 야기되는 테스트 비용의 증가를 줄이기 위해 BIST 기법의 필요성은 더욱 커지고 있다. 더욱이 적용되는 테스트 알고리즘이 상당히 복잡할지라도 BIST 기법은 병렬 테스트 방식과 라인 모드 테스트 방식 등을 이용하여 테스트 시간을 효과적으로 줄일 수 있다.

또한 칩내에 장착시킨 BIST 회로는 제조사의 테스트뿐만 아니라 다른 회로와 조합되어 있는 경우에도 테스트를 할 수 있는 장점을 갖고 있다. 즉, 적용되는 테스트 알고리즘의 길이가 상당히 짧을 경우 BIST 회로는 컴퓨터에 전원이 공급되는 동안에 CPU의 자체 테스트로써 메모리를 테스트할 수 있다.

본 고에서는 DRAM을 중심으로 하여 서술할 것이며, 먼저 메모리에 사용되는 고장 모델과 BIST 기법에 적용 가능한 테스트 알고리즘에 대하여 간략하게 살펴본 후, BIST 구현시 고려해야 할 몇가지 사항들을 제시하고 현재까지 학계 및 산업체에서 발표한 여러 논문 및 연구자료들을 토대로 하여 BIST 기법을 소개한다.

그리고 마지막으로 결론 및 앞으로의 연구 동향에 대하여 논한다.

## II. 고장 모델 및 테스트 알고리즘

### 1. 고장 모델

RAM에서는 여러 종류의 고장들을 생각할 수 있는데 여기에서의 셀(cell)은 데이터를 저장하는 장치이거나 레지스터로 볼 수 있다. 또한 선은 하나의 메모리 블록 내에서나 다른 블록으로 제어신호 또는 데이터를 전달하는 역할을 한다. 만일 메모리 칩 테스트에 있어 고장 발생 위치보다는 고장 검출에 더 관심이 있는 경우에는 고장 모델을 단순화할 수 있다. 기능적 테스트를 위해서는 어드레스 디코더, 메모리 셀 배열, 읽기/쓰기(read/write) 논리회로로 모델링할 수 있다. 즉, 어드레스 래치, 행 디코더 그리고 열 디코더를 어드레스 디코더로 묶는다. 쓰기 구동기, 감지 증폭기 그리고 데이터 저장기는 R/W 논리회로로 묶을 수 있다. 따라서 축소 기능 모델을 사용하여 여러 종류의 고장들을 단순화하면 고착 고장(Stuck-At Fault : SAF), 전이 고장(Transition Fault : TF), 결합 고장(Coupling Fault : CF), 패턴 감응 고장(Pattern-Sensitive Fault : PSF), 디코더와 R/W 논리회로에서의 고장으로 나눌 수 있다.

### 고착 고장 :

고착 고장은 다음과 같이 정의할 수 있다. 셀 또는 선이 고착되어서 논리 값이 항상 0(SA0 고장) 또는 1(SA1 고장)으로 머물러 있는 상태로 그 값이 변하지 않는다.<sup>[4]</sup> SA0 고장은 해당 셀이 1 값을 갖고 변하지 않기 때문에 “w1” 동작(논리 1 값을 쓰는 동작)에 의해서 검출할 수 있다. 즉 “w1” 동작 후에 읽기 동작은 정상적일 경우 1의 값을 요구하기 때문에 고장을 검출할 수 있다.

### 천이 고장 :

고착 고장의 특수한 경우로서 천이 고장을 들 수 있다. 이는 하나의 셀 또는 선이 0 → 1 천이를 경험하지 못할 때 상향 천이 고장, 1 → 0 천이를 경험하지 못할 때 하향 천이 고장이라고 한다.<sup>[4]</sup> 처음에 셀이 1 상태이었고 “w0”(논리 0 값을 쓰는 동작)에 의해서 0 상태가 된 후 어떤 값을 주어도 다른 상태로 천이되지 않는 고장은 셀이 0 상태일 때 “w1” 동작 후 해당 셀의 기대값이 1이지만 0이 읽힘으로 고장 검출이 가능하다.

### 결합 고장 :

결합 고장은 한번의 천이 쓰기 동작에 의해서 유발되며 이를 2-결합 고장(2-coupling fault)이라고 한다.<sup>[4]</sup> 이것은 2개의 셀이 관련되며 다음과 같이 정의한다. 한 셀이 0인 셀에 1을 쓰거나 1인 셀에 0을 쓰는 천이 쓰기 동작을 할 때 결합된 두 번째 셀의 내용이 변하는 경우이다. 이처럼 2개의 셀이 관련된 결합 고장은 일반적인 k-결합 고장의 특수한 경우에 해당된다. k-결합 고장은 k개의 셀들이 메모리 셀 배열 내에서 그들의 위치에 제약을 받지 않는 대단히 복잡한 고장이다.

제약된 k-결합 고장의 다른 형태는 inversion CF와 idempotent CF가 있다. 두가지 경우 모두 쓰기 동작에 의해서 검출된다.

#### - 반전 결합 고장(inversion coupling fault) :

반전 결합 고장(CFin)은 다음과 같이 정의된다. 하나의 셀에서 천이 쓰기 동작을 할 때 결합된 두 번째 셀이 원래 내용과 반대로 바뀌는 경우이며 셀 j의 값 천이가 셀 i의 결합 고장의 원인이 될 때 셀 i는 셀 j에 결합되었다고 하고 셀 i는 결합된

셀(coupled cell) j는 결합원 셀(coupling cell) 이라고 한다.<sup>[4]</sup> 이는 셀 j의 천이가 셀 i의 내용을 천이시키는 경우이다.

#### - 동행 결합 고장(idempotent coupling fault) :

동행 결합 고장(CFid)은 하나의 셀에서 천이 쓰기 동작을 할 때 결합된 두 번째 셀의 내용이 강제로 0 또는 1로 바뀌는 형태의 고장이다.<sup>[4]</sup>

#### 이웃 패턴 감응 고장 :

패턴 감응 고장은 다음과 같은 경우에 발생하는 데 한 셀의 내용이 메모리 배열 내의 다른 모든 셀의 내용에 따라 영향을 받는 것으로 이들 내용은 0과 1로 이루어져 있다.<sup>[4]</sup> PSF는 k-결합 고장의 가장 일반적인 경우로 생각할 수 있으며 k=n인 경우이다. 이웃셀은 전체 셀 중 특정 고장과 관련된 셀이며 기준셀(base cell)은 테스트 대상이 되는 셀을 말한다. 기준셀을 제외한 이웃을 근접 이웃(neighborhood)이라고 한다. N(Neighborhood) PSF는 PSF의 부분집합(subset)에 해당한다. 근접된 이웃은 기준셀에 전기적으로 가장 큰 영향을 미치고 이는 물리적으로 기준셀에 가장 근접한 셀이 된다.

#### - 능동 이웃 패턴 감응 고장 :

동적(active) NPSF라고도 하며 근접된 이웃 셀의 패턴 변화가 기준셀의 내용을 변화시킨다.<sup>[4]</sup> 이러한 변화는 근접 이웃셀 중 하나의 값이 천이될 때 일어난다.

#### - 수동 이웃 패턴 감응 고장 :

수동(passive) NPSF라고도 하며 임의의 근접 이웃 패턴에서 기준셀의 내용이 변할 수 없는 고장이다.<sup>[4]</sup>

#### - 정적 이웃 패턴 감응 고장 :

기준셀의 내용이 임의의 한 근접 이웃 패턴에 의해 임의의 한 상태로 유지 혹은 변화되도록 강요받는 고장 형태이다.<sup>[4]</sup>

#### - 행/열 가중치 감응 고장 :

이 고장 모델은 어떤 셀의 내용이 그 셀의 행과 열 이웃에 있는 셀들의 내용에 의해 영향을 받을 수 있다는 관점에 근본을 두고 있다. 전기적으로 같은 행이나 열에 있는 셀들은 어드레스와 리프레

쉬 회로를 서로 공유하고 있기 때문에 상호간에 간섭을 일으킬 수 있다. 만일 어떤 한 셀의 내용이 임의의 행/열 가중치 조합에 감응할때 이 고장이 그 셀에 발생된 것이다.<sup>[5]</sup>

**2. 테스트 알고리즘**

앞에서 소개한 메모리 고장모델들을 테스트하기 위한 다양한 알고리즘이 개발되어 왔다. 그러나 고장모델 모두를 만족시키는 테스트 알고리즘의 개발은 실제로 불가능하며 따라서 목적에 맞는 알고리즘을 선택하거나 이들을 조합해서 사용해야 한다. 여기에서는 BIST 기법에 적용할 수 있는 알고리즘을 중심으로 소개하겠다. 모든 테스트 알고리즘은 메모리 내의 셀에 적용할 임의의 순서로 열거된 R/W 순차열로 구성되어 있다.

**Mscan 테스트 :**

가장 단순한 테스트 알고리즘으로 동작은 다음과 같다. 먼저 어드레스를 오름차순으로 메모리 전체에 0을 쓴다. 같은 방법으로 어드레스를 오름차순으로 기대값 0을 읽는다. 이때 SA1 고장을 검출할 수 있다. 계속해서 1을 쓰고 기대값 1을 읽음으로서 SA0 고장을 검출할 수 있다. 이 테스트 알고리즘은 메모리 어드레스 레지스터 또는 디코더에서 발생하는 SAF는 검출할 수 없지만, 메모리 배열, 메모리 데이터 레지스터, 그리고 R/W 논리회로에 대한 SAF를 검출할 수 있다. 테스트가 각 셀당 4 동작을 수행하기 때문에 테스트 길이는 4N이 된다.

**Marching 테스트 :**

이 테스트 알고리즘에서는 메모리를 오름차순으로 조사(scan)할 때에는, 현재 셀과 상위 어드레스를 갖는 셀 사이의 직접적인 결합이 후자를 읽을 때 검출되어 진다. 그리고 디코더 고장에 의한 상위 어드레스 셀에서 발생하는 고장도 검출할 수 있다. 또한, 메모리를 내림차순으로 조사할 때에는 하위 어드레스 셀에 대한 모든 영향을 찾아낸다. Marching 테스트는 모든 SAF와 디코더 고장을 검출할 수 있지만, 모든 단일 CF는 검출하지 못한다. 이 테스트의 길이는 14N이 된다.<sup>[6]</sup>

**Checkerboard 테스트 :**

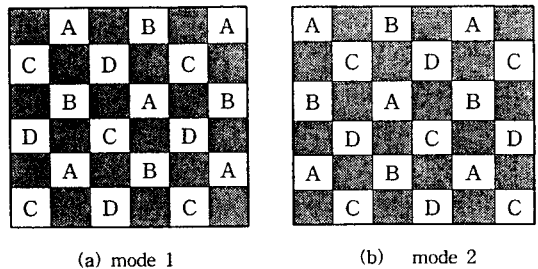
이 테스트 알고리즘은 그림 1에서 보여주는 바와 같이 메모리 배열에 check 무늬 식으로 0과 1을 쓰고 읽어서 테스트를 수행한다. 어드레스 디코더의 기능이 정상적일 때, 이웃셀 사이의 단락을 검출할 수 있다. 이것은 원래 DRAM(Dynamic RAM)의 리프레쉬 동작을 테스트 하기 위해 제안되었다. 테스트 길이는 각 스텝마다 2N 동작이 필요하므로 4 2N이 된다.<sup>[4]</sup>

1	2	1	2	1	2
2	1	2	1	2	1
1	2	1	2	1	2
2	1	2	1	2	1
1	2	1	2	1	2
2	1	2	1	2	1

(그림 1) Checkerboard 테스트를 위한 셀의 번호.

**Five-cell-neighborhood static PSF 테스트 :**

많은 알고리즘이 five-cell-NPSF를 검출하기 위해 제안었다. 모든 알고리즘은 메모리 배열을 tiling하는데 기본을 두고 있다. 그림 2에서 tiling 기법을 보여주고 있다. 음영 부분이 테스트 대상이 되는 기준셀이다. 각 기준셀들은 4개의 문자(A, B, C, D)에 의해 둘러 쌓여 있고 4개의 주변셀을 변화시키면서 그 때마다 기준셀의 내용을 읽어 보는 방식이다. 이 테스트 알고리즘은 기본적으로 SNPSF와 함께 SAF를 검출할 수 있다.<sup>[7]</sup>



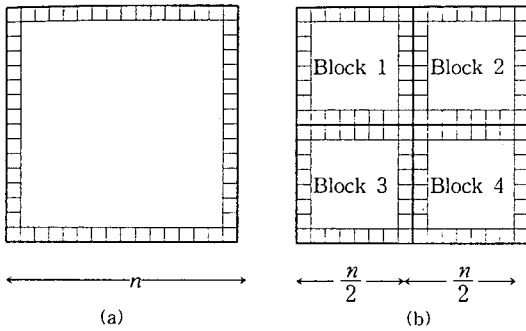
(a) mode 1

(b) mode 2

(그림 2) Five-cell-neighborhood PSF 패턴.

**행/열 가중치 감응 고장 테스트 :**

행/열 가중치 감응 고장을 테스트하기 위해 여러 알고리즘이 제안되었는데, 이들은 테스트 길이가  $O(N^3/2)$  이고 기본적으로 반복적인 분할을 통해서 테스트를 완성해 가는 “divide and conquer” 방법을 사용하였다.<sup>[3, 5, 7, 8]</sup> 첫번째로 가장자리에 위치한 셀들에 대한 테스트가 완료되고 그림 3(a)에 그 결과를 보여주었다. 그림 3(b)는 가장자리 셀들을 제외한 내부의 셀들을 테스트 하기 위해 메모리 전체를 4개의 블록으로 나누어서 각 블록을 비슷한 방법으로 테스트하며, 이때 각 블록의 가장자리 셀은 그림 3(a)에서처럼 테스트가 완료된 상태를 보여준다.



(그림 3) 메모리 배열의 분할

- (a) 가장자리 셀들의 테스트 완료 후 메모리의 상태
- (b) 메모리의 분할 후 메모리의 상태

### III. BIST 회로의 구조 및 구현

#### 1. BIST 회로의 구조

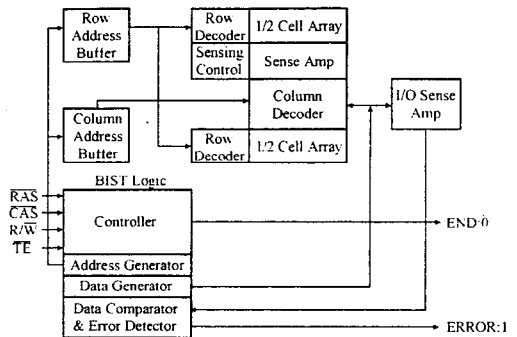
메모리 칩 제작자들은 엄격한 설계 법칙을 적용하여 칩당 메모리 셀 수를 증가시키고 메모리 액세스 시간을 최소화시키고 있다. 그러나 이런 상황은 BIST 회로를 구현하는 테스터에게 상당한 제약이 주게된다. 따라서 BIST 회로를 구현할 때 다음과 같은 사항을 최소화해야 한다.<sup>[9]</sup>

- BIST 회로가 점유하는 면적
- BIST 동작으로 인한 정상적인 메모리 동작 지연
- BIST에 요구되는 핀 수

RAM을 테스트할 경우, BIST 회로는 각 셀에 쓰여질 입력 데이터 패턴과 각 셀의 내용을 읽고 쓰는데(R/W) 필요한 어드레스 패턴 및 R/W 신호, 그리고 테스트 모드에서 BIST 회로의 각 부를 제어하는 신호들을 발생시켜야 한다. 즉, BIST 회로는 다음 네가지의 기본 기능을 수행해야 한다.<sup>[9]</sup>

- 메모리의 정상 모드와 테스트 모드의 전환 기능
- 패턴 발생 기능 (어드레스, 데이터)
- 비교 및 고장 검출 기능
- BIST 회로의 테스트 절차 제어 기능

BIST 회로는 앞에서 소개한 기본 기능을 수행하기 위해 제어부, 어드레스 발생부, 데이터 발생부, 데이터 비교 및 고장 검출부로 구성되고, 그림 4에 랜덤 논리 회로 구현 방식을 이용한 BIST DRAM의 기본 구조를 도시하였다.



(그림 4) BIST DRAM의 기본 구조(랜덤 논리 회로 방식).<sup>[10]</sup>

#### 제어부 :

테스트의 시작과 종료를 조정하고 테스트 알고리즘의 순서에 따라 BIST 회로의 각 부를 제어한다. 구현 방식에는 랜덤 논리 회로를 이용한 방식과 마이크로프로그램 ROM을 사용한 방식이 있다.<sup>[5]</sup> 랜덤 논리 회로 방식은 BIST 회로 내의 제어부를 주로 FSM(Finite State Machine)을 사용하여 구현하며 간단한 알고리즘을 적용할때 용이하다는 장점이 있다. 마이크로프로그램 ROM 방식은 알고리즘의 각 테스트 절차들을 마이크로코드화하여 ROM에 저장함으로써 BIST 회로의 각 부를 제어

하는 방식으로, 랜덤 논리 회로 방식보다 테스트 처리 속도는 느리지만 어떠한 테스트 알고리즘이라도 ROM에 프로그램함으로써 구현할 수 있다는 장점을 가지고 있다. 따라서 복잡한 알고리즘일수록 마이크로프로그램 ROM 방식의 BIST 구현이 더 실용적이다.

#### 어드레스 발생부 :

LFSR(Linear-Feedback Shift Register), 마이크로 프로세서, 또는 카운터를 사용하여 구현되며<sup>4)</sup> 테스트 대상이 되는 셀의 어드레스를 발생시킨다. LFSR을 사용할 경우에는 카운터를 사용하여 구현할때 보다 적은 면적을 차지하고 좀더 쉽게 자신을 테스트할 수 있다. 그러나 고장 검출 범위가 확률적으로 결정되는 단점이 있다. 마이크로 프로세서의 경우에는 높은 하드웨어 오버헤드와 낮은 수행 능력의 단점이 있으나 고장 수리 목적으로 여분의 행/열 사용을 고려할 수 있는 유연성을 갖고 있다. 카운터로써 구현된 어드레스 발생부는 쉽게 설계할 수 있으나 LFSR과 비교하여 좀더 많은 면적을 차지하게 되며 회로 자체 테스트를 수행하기 어렵다.

**데이터 발생부 :** 테스트 대상이 되는 셀에 써넣을 데이터를 발생시킨다.

#### 데이터 비교 및 고장 검출부 :

정의된 순간에 테스트 대상이 되는 셀의 데이터와 데이터 발생기에서 출력된 데이터(고장이 없을 경우의 데이터)를 비교하여 메모리 고장의 유,무를 판별하며, 결정론적인(deterministic) 비교 방식, 상호(mutual) 비교 방식, 다중 입력 쉬프트 레지스터, 그리고 마이크로 프로세서 분석 방식이 있다.<sup>4)</sup> 결정론적인 비교 방식은 같은 행에 있는 셀의 내용을 기준 데이터와 동시에 비교하여 고장을 판별한다. 이 방식은 다중 비트(multiple bit) 구조에서 사용할 수 있으며, 메모리의 정상적인 동작에서 이 회로에 의해 발생할 수 있는 신호의 지연을 야기하지 않고 낮은 하드웨어 오버헤드를 갖는다. 상호 비교 방식은 다중 어레이(multiple array) 구조에서 사용되며 어레이들로 부터 발생되는 출력들을 상호 비교한다. 이 방식 또한 메모리의 정상 동작시 지연 문제가 발생되지 않으며 차

지하는 면적이 적다. 다중 입력 쉬프트 레지스터 방식은 평가될 응답 데이터의 양을 줄이기 위해 압축 기술이 사용되며, 메모리의 정상 동작시에 하드웨어에 고장이 발생할 경우 고장 기호(signature)를 발생시켜 자체 테스트를 수행할 수 있는 장점이 있다. 그러나 구현시 많은 면적이 요구되며, 반복되는 응답 데이터 패턴이 예측될 경우에 고장 응답 신호가 정상적인 응답 신호와 같이 압축될 수 있는 마스킹(masking 또는 aliasing) 문제가 발생할 가능성이 적지않다. 마이크로 프로세서 분석 방식은 고장 수리의 목적으로 사용될 수 있으나 하드웨어의 오버헤드가 많고 수행 능력이 낮은 단점이 있다.

랜덤 논리 방식에서 BIST를 위한 추가 핀과 각 핀의 상태는 다음과 같다.<sup>10)</sup>

- TE(Test Enable) : BIST 모드와 정상 모드 동작을 구분시키는 핀으로, BIST회로의 동작이 실행되는 동안 1 상태를 유지한다.
- ERROR : BIST 모드에서 데이터 비교 회로에 의해 고장이 검출되면 1 상태가 되며, 고장이 검출되지 않으면 0 상태를 유지한다.
- END : BIST 동작 모드에서 1상태이며, 테스트가 종료되면 0상태가 된다.

## 2. BIST의 구현

지금까지 학계 및 산업체에서 BIST를 구현한 여러 논문이 발표되었다. 여기에서는 테스트 알고리즘, 테스트 구조, 그리고 제어회로와 같은 중요한 구현 형태를 살펴보기 위해 기존에 발표된 논문들의 내용을 소개한다.

### A Self-Testing Dynamic RAM Chip

제한된 PSF(restricted PSF) 테스트 알고리즘을 제안하였고 제어부는 랜덤 논리 방식을 채택하였다.<sup>11)</sup> 효과적인 테스트를 위해 메모리 어레이 셀을 순환 쉬프트 레지스터 형태로 재 구성하였고, 다중 비트를 동시에 테스트하기 위해 built-in self 발생기를 사용한 병렬 테스트 기법의 한 형태를 제안하였다. 여기에서 DRAM은 전체적 테스트 시간을 줄이기 위해 동일한 두개의 어레이들로 나누었으며, 이 어레이들은 병렬로 테스트된다. 테스트

패턴은 테스트 동안 메모리의 각 어레이에서 셀들이 순환 쉬프트 레지스터와 같이 동작하도록 재구성함으로써 발생되어진다. 한 어레이에서 하나의 행(워드라인)이 동작할때, 그 행에 포함되어 있는  $n$ 개 셀들의 내용은  $n$  비트라인으로 옮겨지고  $n$ 개의 센스 증폭기(sense amplifier)를 통과한 후, 같은 행에 있는 근접 셀들로 쓰여진다. 즉,  $n$ 개의 셀을 갖는 각 행은  $n$ -비트 쉬프트 레지스터와 같이 동작하게 된다. 이때 가장 오른쪽에 위치한 센스 증폭기로 부터 발생된 데이터는 플립플롭에 저장된 후, 다음 행의 가장 왼쪽에 있는 셀로 이동된다. 따라서, 한 어레이를 구성하는  $m$  행은  $mn$ -비트 쉬프트 레지스터의 형태를 이루게 된다. 마지막 행의 가장 오른쪽에 위치한 셀의 초기 내용을 저장함으로써 어레이는  $mn$ -비트 순환 쉬프트 레지스터를 구성하게 된다. 한 셀로부터 비트 값을 읽어서 같은 행에 있는 근접 셀로 그 값을 쓸수 있도록 하기위해 실제적으로 근접한 비트라인 사이에 통과 트랜지스터(pass transistor)를 위치시켜 센스 증폭기를 수정하였다. 내장된 비교 회로는 두 어레이에 대칭적으로 위치한 셀들의 결과를 비교하여 고장을 검출할 수 있도록 EXOR 게이트로 구성하였다. 구현한 회로는 두개의 어레이가 병렬로 테스트되고, 각각의 어레이에서 한 행에 포함된 셀들의 비트들이 동시에 액세스되기 때문에 MAMB (Multiple Array Multiple Bit) 테스트 구조로서 분류된다. 4K DRAM과 1M DRAM에서 면적에 대한 오버헤드는 각각 12%, 5%이다.

### New Parallel Test Approach for Large Memories

각기 다른 어레이에서 여러 비트라인들을 동시에 테스트하기 위해 병렬 기호 분석(parallel signature analyzer) 방법을 제안하였다.<sup>[12]</sup> 이 병렬 기호 분석기는 scan 모드, 쓰기 모드, 기호/읽기 모드로 동작된다. Scan 모드에서는 분석기가 외부 칩으로부터 특유한 형태의 패턴을 가져와서 비트들을 순서대로 조사한다. 또한 테스트 종료시 분석기의 내용을 조사할때에도 사용된다. 쓰기 모드에서는 분석기에 저장된 값이 병렬로 여러 비트라인들에 쓰여진다. 기호 모드의 경우에는 메모리 셀들

에 미리 쓰여진 내용들을 읽고 고장의 유,무를 결정하는 새로운  $k$ -비트 기호를 생성한다. 사실 이 논문에서 구현한 BIST 방식은 외부 칩에서 발생하는 데이터를 사용해야하기 때문에 완벽한 BIST라 할 수 없다. MAMB 구조를 사용하여 매우 빠른 테스트 시간을 이룰 수 있다. 그러나 데이터를 조사하기 위해 외부 테스트 장비가 요구되는 문제점을 갖고 있다. 또한 marching 테스트 알고리즘을 사용하기 때문에 고장 검출 범위도 적다. 256K DRAM에서 BIST 회로의 면적 점유율은 1.0~2.2%이고 64K SRAM(Static RAM)인 경우에는 1.8~2.9%이다.

### Built-In Self Testing RAM : A Practical Alternative

RAM에서 발생하는 5셀 이웃 패턴 감응 고장을 검출하기 위한 BIST 회로를 구현하였다.<sup>[13]</sup> 메모리 셀을 테스트하기 위해 요구되는 패턴을 그 패턴의 가중치에 따라 여러가지 다른 종류로 분류하였으며, 패턴을 적용하기 위해 최선의 테스트 길이를 갖는 알고리즘을 제안하였다. BIST의 제어부는 랜덤 논리 방식과, 간단하고 효과적인 명령 집합을 사용한 마이크로프로그램 ROM 방식을 사용하였다. SASB(Single Array Single Bit) 구조를 사용하였으며, 64K SRAM에 대한 오버헤드는 1.21%이고 1M SRAM의 경우에는 0.09%이다.

### Parallel Testing Technology for VLSI Memories

SAMB(Single Array Multiple Bit) 구조를 사용한 라인 모드 테스트를 제안하였다.<sup>[14]</sup> 라인 모드 테스트 방법은 한 워드라인에 연결된 모든 셀들을 동시에 테스트한다. 장착된 테스트 회로는 한 워드라인에 연결된 모든 셀들에 데이터를 동시에 써넣기 위한 병렬 쓰기 동작과 기대 데이터(expected data)와 셀로부터 읽은 데이터를 병렬로 비교하는 동작을 수행한다. 메모리 셀에 대한 테스트는 marching 알고리즘을 사용하였으며, 여러가지 테스트 알고리즘을 사용하여 디코더, 테스트 회로, 그리고 I/O 회로를 테스트하였다. 테스트 회로의 면적에 대한 오버헤드는 2M DRAM의 경우 1% 이하가 된다. 그러나 병렬 쓰기 동작은 셀에 적

용할 임의의 패턴만을 허용하기 때문에 셀간의 간섭에 따른 고장을 검출할 수 없는 문제점이 있다.

#### A Novel Approach for Testing Memories Using A Built-In Self Testing Technique

테스트 시간을 단축시키기 위해 테스트 모드 동안 동시에 많은 메모리 셀에 액세스하도록 여분의 입력으로써 디코더를 제어하게 하는 방식을 제안하였다.<sup>[15]</sup> 메모리 구조로는 NMOS 기술을 기본으로 한 SRAM을 채택하였다. 추가된 입력단과 몇 가지 로직 게이트를 사용하여 다중 셀들과 셀 분할이 동시에 이루어질 수 있도록 디코더를 수정하였다. 또한 EXOR 게이트를 사용하여 고장을 검출하기 위해 읽기 회로내에 있는 센스 증폭기에 두개의 관측점을 두었다. 여기에서 제안한 테스트 알고리즘은 메모리 셀에서 발생 가능한 SPSF를 검출하기 위한 16 4-비트 그레이 코드 패턴, DPSF를 검출하기 위한 64 4-비트 천이 패턴, 그리고 ESSF(Electrical neighborhood SPSF)를 검출하기 위한 테스트 패턴으로 구성되어 있고, 어드레스 디코더 고장을 찾아내기 위한 알고리즘으로 이루어졌다. BIST RAM 칩은 제어 가능한 디코더, 메모리 어레이, R/W 회로와 앞에서 소개한 두개의 관측점으로 이루어진 테스트 가능한 RAM (TRAM)과 테스트 발생기로 구성되었다. 이 칩은 모드 선택, 클럭, 테스트 종료, 그리고 테스트 고장을 나타내기 위해 4개의 추가핀이 필요하다.

#### Built-In Test Processor for Self-Testing Repairable Random Access Memories

대용량 RAM의 테스트와 고장 수리(repair)를 목적으로 built-in 프로세서를 사용하여 BIST를 구현하였다.<sup>[16]</sup> 고장 수리는 지능적인 자체 테스트 개념을 이용하여 고장 위치와 수리 계획을 위한 계산등이 요구된다. 테스트 프로세서는 높은 유연성이 요구되며, 새로운 메모리 제작 기술이 발달되고 새로운 종류의 고장과 고장 모델이 발생될 경우 좀더 복잡한 테스트 알고리즘을 적용시켜야하기 때문에 여러가지 다른 메모리 종류와 구조에 좀더 쉽고 빠른 적응성을 지원해야 한다. 테스트 기능은 제작시 뿐만 아니라 시스템의 유지 서비스 동안에도 적용되어야 한다. 여기에서 제안한 테스트 프

세서의 주된 요소들은 RAM 셀과 ROM 셀 그리고 디코더이며, 테스트 프로그램을 위해 사용된 ROM의 용량은 512×14 비트이다. SAMB 구조를 사용하였으며 1M RAM의 경우 BIST 회로의 면적에 대한 오버헤드는 약 5%이다.

#### A 60-ns 4-Mbit CMOS DRAM with Built-In Self Test Function

일본의 도시바에서 발표한 4M DRAM을 위한 BIST 회로에 대한 논문으로서 checkerboard 테스트 패턴과 그에 대한 보수 패턴을 구현하였고, MAMB 테스트 구조로서 분류된다.<sup>[17]</sup> RAM은 8개의 어레이로 나뉘었고 그중 2개는 R/W 사이클에서 동작한다. 각각의 어레이로부터 8비트가 동시에 액세스되어 읽어진 데이터 패턴과 기대 데이터 패턴이 데이터 비교기를 통해 비교되어지고 고장 유, 무가 판별된다. 제어부는 랜덤 논리 회로로써 구현되었으며 BIST 회로의 점유율은 1% 이내이다. 그러나 채택한 checkerboard 테스트 알고리즘의 낮은 고장 검출 범위에 따른 문제점을 갖고 있다.

#### Parallel Testing for Pattern-Sensitive Faults in Semiconductor Random Access Memories

같은 워드라인에 연결된 셀들이 동시에 액세스될 수 있도록 병렬 테스트 기법을 제안하였다.<sup>[18]</sup> 테스트 모드에서 다중 비트라인들이 선택되어지고, 데이터 값이 같은 워드라인에 있는 다중 비트들에 쓰여지도록 하기위해 디코더를 수정하였다. 읽기 모드에서 다중비트 비교기는 비트라인들의 출력들을 동시에 비교한다. 256K RAM에서 하드웨어 오버헤드는 약 0.4%로 추정하였다.

#### A Built-In Self-Test Algorithm for Row/Column Pattern Sensitive Faults in RAM's

행/열 패턴 감응 고장 테스트 알고리즘이 랜덤 논리 방식과 마이크로프로그램 ROM 방식으로 구현되었으며, 두 방식 모두 SASB 테스트 구조를 사용하였다.<sup>[5]</sup> 이 SASB 구조의 문제점은 테스트 시간이 상당히 소모된다는 것이다. 이 문제점은 MASB(Multiple Array Single Bit)와 MAMB 구



조를 사용하여 해결할 수 있다. 여기에서 제안한 테스트 알고리즘의 테스트 길이는  $10N3/2$ 이고, 4M RAM에 대한 오버헤드는 0.8% 정도이다.

### Serial Interfacing for Embedded-Memory Testing

Embedded RAM의 직렬 인터페이스 기법을 소개하였다.<sup>[19]</sup> Embedded RAM은 어드레스, 데이터, 그리고 R/W 제어가 칩의 입, 출력핀을 통하여 직접적으로 조절할 수 없는 형태로서 BIST 기법을 적용할 수 있는 좋은 응용 예가 된다. 구현 방식은 앞에서 소개한 DRAM의 자체 테스트 방식과 비슷하게 데이터를 하나의 메모리 셀에서 다른 셀로 이동시키는 방법을 이용하였다. 두 가지 방식 모두 MAMB 구조를 사용하였지만, DRAM의 자체 테스트의 경우에는 한 어레이 내에서 데이터를 이동시키고 병렬로 두 어레이가 테스트된다. 그러나, 이 논문에서 제안한 방식은 한 어레이 뿐만 아니라 다른 어레이로 데이터를 이동시켜서 테스트한다. 즉, 데이지 체인(daisy-chain) 방식을 이용하여 한 어레이의 마지막에 있는 데이터를 다른 어레이의 처음에 위치한 셀로 옮긴다. 이것은 BIST 회로와 RAM 블럭사이에 적은 수의 상호연결선을 필요로 하기 때문에 BIST 회로가 다중 어레이들 사이에서 좀더 쉽게 공유될 수 있다. 따라서 상호연결선에 기인하는 오버헤드를 줄일 수 있게 된다.

### A 55-ns 16-Mb DRAM with Built-in Self-Test Function Using Microprogram ROM

16M DRAM의 BIST를 위해 마이크로코드를 사용하여 marching 테스트와 checkerboard scan R/W 패턴을 구현하였다.<sup>[20]</sup> 장착된 ROM은  $18 \times 10$  비트로서 구성되었으며, BIST 회로가 차지하는 오버헤드는 약 1%이다.

### A BIST Scheme Using Microprogram ROM for Large Capacity Memories

LSI 메모리 테스트에 적용할 N 패턴, N2 패턴, 그리고 데이터 보유 테스트를 마이크로프로그램 ROM 방식을 사용하여 BIST를 구현하였다.<sup>[21]</sup>  $16 \times 16$ 비트 ROM을 갖는 BIST 회로는 테스트 길이가  $O(N2)$ 인 galloping 테스트 알고리즘을 구현하였으며, 16M DRAM에 대한 면적의 오버헤드는

2% 이내가 된다. 또한 BIST 회로 자체에 대한 테스트 방안을 제안하였다.

### 고장위치 검출 가능한 BIST용 패턴 발생회로의 설계

테스트 시간을 단축시킬 수 있고 일반적인 고장 모델을 검출할 수 있는 대용량 메모리 테스트에 적용가능한 알고리즘을 BIST 회로에 적용하기 위해, 테스트시 요구되는 어드레스와 데이터를 발생시키는 패턴 발생부와 이 회로에서 발생하는 정보로써 고장의 유, 무 및 그 위치를 검출해 주는 고장위치 검출기를 설계하였다.<sup>[9]</sup> 여기에서는 열 가중치 감응 고장 모델을 제시하여 이 고장을 검출할 수 있는 알고리즘을 제안하였다. BIST 회로의 제어부는 랜덤 논리 방식이며, SASB 구조를 사용하였다. 또한 테스트 시간을 단축시키기 위해 병렬 테스트 기법을 이용하였다. 1M DRAM의 경우, BIST 회로가 차지하는 면적 비율은 약 0.89%이다. 그러나 행의 가중치를 고려하지 않아 고장 검출 범위가 다소 낮은 문제점이 있다.

### A Design of BIST Circuit for Testing ULSI DRAM

초고집적 메모리의 테스트 방법과 이를 BIST 방식을 이용하여 회로를 구현하는 방법에 대해 연구하였다.<sup>[10]</sup> 높은 고장 검출율을 가지면서 충분히 실용 가능한 테스트 시간을 갖는 행/열 가중치 감응 고장 알고리즘을 수행하는 회로를 BIST 기법과 하드웨어 기술 언어(VHDL)을 이용하여 설계하였다. 여기에서 BIST 회로의 제어부는 랜덤 회로 방식이고 사용된 구조 또한 SASB이며, 이 구조의 단점은 MASB 구조로서 보완할 수 있다. 제안한 알고리즘의 길이는  $9N3/2$ 이고 'divide and conquer' 방법으로 테스트를 수행한다. BIST 회로의 오버헤드는 64M DRAM의 경우 0.0055%이고 256M DRAM일때에는 0.0015%를 갖는다.

### Industrial BIST of Embedded RAMs

이탈리아의 한 통신업체에서 Embedded RAM의 BIST를 위해 선형적인 복잡도를 갖는 알고리즘을 채택하여 비트 기반(bit-based) 메모리가 아닌 워드 기반(word-based) 메모리로 확장하여 테스트 발생회로를 구현하였다.<sup>[22]</sup> RAM을 설

계하는 사람이 칩의 테스트 능력과 신뢰도를 높이기 위해 기존에 라이브러리에 저장해 놓은 BIST 셀을 사용할 수 있는 환경을 조성하는데 목적이 있다. 즉, 구현되어 저장될 BIST 셀은 디자인 룰과 파라미터에 의존하지 않도록 설계되어야 한다. 선택한 테스트 알고리즘은 RAM의 실제적인 구조를 고려하지 않고 독립적으로 적용시킬 수 있는 장점이 있다. 여기에서는 SAF와 CF만을 선택하였고 PSF는 고려하지 않았다. BIST 셀의 오버헤드는 1K×8 RAM에서 약 9%를 차지하게 된다.

#### IV. 결 론

본 고에서는 초고집적 메모리에서 테스트 시간을 단축시킬 수 있고 신뢰도를 높일 수 있는 BIST 기법에 대해 살펴보았다. 테스트 시간은 테스트 비용에 많은 영향을 주기 때문에 단축된 테스트 시간은 비용을 절감시킬 수 있다고 판단된다. 메모리 고장 모델의 경우, 고장을 수리할 목적으로 메모리 칩내에서 여분의 행과 열을 이용한 재구성 방법의 영향을 고려하여 고장 모델을 설정해야 할 것이며, Ga-As를 이용한 새로운 방식의 고속 RAM에서는 지연 고장(delay fault) 또한 고려해야 할 것이다. 현재 메모리에서 발생 가능한 모든 고장들을 빠른 시간내에 검출할 수 있는 알고리즘과, 전체회로에 대한 BIST 회로가 차지하는 면적 오버헤드를 줄이기 위해 많은 연구가 진행되고 있다. 일반적으로 지금까지 발표된 BIST 구현은 고장을 진단할 수 없다. 그러나 메모리 칩의 재구성과 다중칩 메모리 모듈의 수리를 도모하기 위해 BIST를 이용한 진단이 필요하게 될 것이다. 그리고 BIST 구현시 발생하는 추가핀에 대한 연구가 필요하고, 마지막으로 기능적 고장 모델로 표현할 수 없는 결점(defect)으로 인한 신뢰도 문제를 해결하기 위해 Iddq(Steady-state power Supply current) 방식과 같은 전류 테스트 기술에 기반을 둔 BIST 방안을 제시해야 할 것이다.

#### 참 고 문 헌

- [1] Y. You, "Testing of memories with tolerable defects," *Int'l Conf. on Elect., Info. and Comm.*, pp. 214-217, Yanji, China, Aug. 23, 1991.
- [2] M.M. Breuer and A.D. Friedman, *Diagnosis and Reliable Design of Digital Systems*, Rockville, MD : Computer Science Press, 1976.
- [3] M. Franklin, K.K. Saluja, "Built-in Self-Testing of Random-Access Memories," *IEEE Comput.*, pp. 45-56, Oct. 1990.
- [4] A.J. Van de Goor, *Testing Semiconductor Memories : Theory and Practice*, John Wiley & Sons, 1991.
- [5] M. Franklin, K.K. Saluja and K. Kinoshita, "A Built-In Self Test Algorithm for Row/Column Pattern Sensitive Faults in RAM's," *IEEE Journal of Solid-State Circuits*, vol. 25, no. 2, pp. 514-524, April 1990.
- [6] A.J. van de Góor, and C.A.Verruijt, "An Overview of Deterministic Functional RAM Chip Testing," *ACM Computing Surveys*, vol. 22, pp. 5-33, 1990.
- [7] K. Kinoshita, K.K. Saluja, "Built-in testing of memory using an on-chip compact testing scheme," *IEEE Trans. Comput.*, vol. SC-35, No. 10, pp. 862-870, Oct. 1986.
- [8] Chang-Ki Lee, Byoung-Sil Chon, "Advanced Test Algorithm for Row/Column Pattern Sensitive Fault in RAMs," *Proc. Int. Conf. on VLSI & CAD*, pp. 377-380, Nov. 1993.
- [9] 김대익, 이창기, 전병실, "고장위치 검출 가능한 BIST용 패턴 발생회로의 설계," 한국

- 통신학회 논문지, 1993. 10
- [10] D. Kim, et al., "A Design of BIST circuit for Testing ULSI DRAM," *Int'l Conf. on Elect., Info. and Comm.*, pp. II-40-43, Yanji, China, Aug. 1995.
- [11] Y. You and J.P.Hayes, "A self-testing dynamic RAM chip," *IEEE Journal of Solid-State Circuit*, vol.SC-20, No.1, pp. 428-435, Feb. 1985.
- [12] T. Shridhar, "A New Parallel Test Approach for Large Memories," *Proc. Int'l Test Conf.*, pp. 462-470, 1985.
- [13] K.K. Saluja, S.H. Sng, and K. Kinoshita, "Built-In Self Testing RAM : A Practical Alternative," *IEEE Design & Test of Comput.*, vol. 4, pp. 42-51, Feb. 1987.
- [14] J. Inoue et al., "Parallel Testing Technology for VLSI Memories," *Proc. Int'l Test Conf.*, pp. 1066-1071, 1987.
- [15] K.T. Le and Kewal K. Saiuja, "A Novel Approach for Testing Memories Using A Built-in self Testing Technique," *Proc. Int. Test Conf.*, pp. 830-839, 1986.
- [16] H.C. Ritter and B. M ller, "Built-in Test Processor for Self-testing Repairable Random Access Memories," *Proc. IEEE Int. Test Conference*, pp. 1078-1084, 1987.
- [17] T. Ohsawa et al., "A 60-ns 4-Mbit CMOS DRAM with Built-In Self-Test Function," *IEEE Journal of solid-state circuits*, vol. 22, no. 5, pp. 663-668, Oct. 1987.
- [18] P. Mazumder and J.K. Patel, "Parallel Testing for Pattern-Sensitive Faults in Semiconductor Random-Access Memories," *IEEE Transactions on Computers*, C-38(3), pp. 394-407, Mar. 1989.
- [19] B. Nadeau-Dostie et al., "Serial Interfacing for Embedded-Memory Testing," *IEEE Design & Test of Comput.*, vol. 7, pp. 52-63, Apr. 1990.
- [20] T. Takeshima et al., "A 55-ns 16-Mb DRAM with Built-in Self-Test Function Using Microprogram ROM," *IEEE Journal of Solid-State Circuits*, vol. 25, No. 4, pp. 903-911, Aug. 1990.
- [21] Hiroki Koike et al., "A BIST Scheme Using Microprogram ROM for Large Capacity Memories," *Proc. Int. Test Conf.*, pp. 815-822, 1990.
- [22] Paolo Camurati et al., "Industrial BIST of Embedded RAMs," *IEEE Design & Test of Comput.*, pp. 86-95, Fall 1995.

## 저자 소개



金大翊

1969年 1月 23日生

1991年 2月 전북대학교 공과대학 전자공학과 졸업(공학사)

1993年 2月 전북대학교 대학원 전자공학 석사과정 졸업(공학석사)

1995年 2月 전북대학교 대학원 전자공학 박사과정 수료

주관심분야: 메모리 테스트, VLSI설계



裴晟桓

1970年 10月 12日生

1993年 2月 전북대학교 공과대학 전자공학과 졸업(공학사)

1995年 2月 전북대학교 대학원 전자공학 석사과정 졸업(공학석사)

1995年~현재 전북대학교 대학원 전자공학 박사과정

주관심분야: 메모리 테스트, 병렬처리



李昌基

1961年 7月 15日生

1988年 2月 전북대학교 공과대학 전자공학과 졸업(공학사)

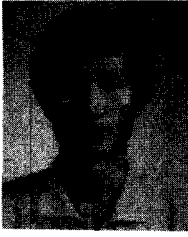
1990年 2月 전북대학교 대학원 전자공학 석사과정 졸업(공학석사)

1994年 8月 전북대학교 대학원 전자공학(공학박사)

1995年 3月~현재

서남대학교 전산정보학과 전임강사

주관심분야: 메모리 테스트, 신경회로망



## 李 相 泰

1954年 12月 6日生

1977年 2月 아주대학교 전자공학과 졸업(공학사)

1992年 2月 전북대학교 대학원 전자공학 석사과정 졸업(공학석사)

1993年~현재 전북대학교 대학원 전자공학 박사과정

1981年~1985年

국제상사 컴퓨터 사업본부 근무

1985年~현재

표준과학연구원 전산센터근무

주관심분야: 테스트알고리즘, 병렬처리, 광대역통신망



## 田 炳 實

1945年 2月 14日生

1967年 2月 전북대학교 공과대학 전기공학과 졸업(공학사)

1969年 2月 전북대학교 대학원 전자공학 석사과정 졸업(공학석사)

1974年 전북대학교 대학원 전자공학(공학박사)

1979年

미국 Univ. of Notre Dame 전기공학과 객원교수

1986年

전북대학교 전자계산소장

1994年~현재

전북대학교 도서관장

1971年~현재

전북대학교 공과대학 전자공학과 교수

주관심분야: VLSI설계, 병렬컴퓨터, 지능망, ATM