

논문 2009-46SD-12-11

# 가상 구역에 따른 메모리 자가 치유에 대한 분석 알고리즘

( Analysis Algorithm for Memory BISR as Imagination Zone )

박재홍\*, 심은성\*, 장훈\*\*

( Jae-Heung Park, Eun-Sung Shim, and Hoon Chang )

## 요약

최근 VLSI 회로 직접도가 급속도로 증가함에 따라 하나의 시스템 칩에 고밀도와 고용량의 내장 메모리(Embedded Memory)가 구현되고 있다. 고장난 메모리를 여분 메모리(Spare Memory)로 재배치함으로써 메모리 수율 향상과 사용자에게 메모리를 투명하게 사용할 수 있도록 제공할 수 있다. 본 논문에서는 고장난 메모리 부분을 여분 메모리의 행과 열 메모리 사용으로 고장난 메모리를 고장이 없는 메모리처럼 사용할 수 있도록 여분 메모리 재배치 알고리즘인 MRI를 제안하고자 한다.

## Abstract

With the advance of VLSI technology, the capacity and density of memories are rapidly growing. In this paper we proposed MRI (Memory built-in self Repair Imagination zone) as reallocation algorithm. All faulty cells of embedded memory are reallocated into the row and column spare memory. This work implements reallocation algorithm and BISR to verify its design.

**Keywords** : BIRA, BISR, Embedded Memory

## I. 서론

VLSI 회로 직접도의 가속화로 인하여 하나의 시스템 칩에 고밀도 및 고용량의 내장 메모리가 구현되어지고 있다. 내장 메모리에 대한 수율은 전체 SoC (System-on-a-Chip)의 수율을 결정하는 중요한 요인이 되었으며, SoC 수율을 높이기 위해서는 SoC에 내장되어 있는 메모리를 효율적인 테스트함으로써 전체 SoC 수율을 증가시키는 것이 필수적이라 할 수 있다.

내장 메모리를 효율적으로 테스트하기 위해서는 메모리 스스로 고장 여부를 판단 할 수 있는 BIST (Built-In Self-Test) 회로를 구성하여야 하며, 이 구현

기술의 우수성과 효율성도 고려되어야 한다<sup>[1~2]</sup>. 그리고, 내장 메모리의 고장여부의 판단과 더불어 고장난 메모리를 복구할 수 있는 고장복구(BISR : Built-In Self-Repair)가 가능한 메모리를 사용함으로써 내장 메모리 및 SoC의 수율을 크게 개선시키는 노력이 이루어지고 있으며 그 사용이 널리 확대되고 있다.

BISR은 BIST와 BISR(Built-In Self-Diagnostics), BIRA(Built-In Redundancy Analysis) 등과 같은 여러 가지 메커니즘을 수반하고 있다. 일반적인 BIST의 구조는 테스트 회로를 테스트하기 위해 패턴을 자동적으로 생성해 테스트 회로에 자동적으로 입력시킨 후, 테스트 패턴과 테스트 후 결과 값이 사용자가 원하는 결과 값과의 동일 여부로 회로의 고장 유무를 판단하는 구조를 가지고 있다.

여분의 메모리 재배치를 위해 ATE(Automatic Test Equipment)를 이용한 여분의 메모리 분석 알고리즘들이 제안되어져 왔다<sup>[3~7]</sup>. BISR의 구조는 회로에 내장되어야 하는데 제안되었던 알고리즘들은 많은 비용이 소요되어 적용하기 어렵다. 또한 [5]에서는 여분의 메모리

\* 학생회원, 송실대학교 컴퓨터학과  
(Department of Computing, Soongsil University)

\*\* 정회원, 송실대학교 컴퓨터학부  
(School of Computing, Soongsil University)

※ 이 논문은 2009년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. 2009-0074893)

접수일자: 2009년10월5일, 수정완료일: 2009년12월2일

워드를 이용해 BISR 구조를 제안하였다. 이 방법은 여분 행 메모리(Spare Row Memory)만을 이용해 여분의 메모리를 분석하였다.

내장 메모리의 행 혹은 열에 고장이 났을 경우 고장난 부분을 여분 행 혹은 열 메모리 하나로만 재배치하는 경우 보다 여분 메모리에 행과 열 모두를 이용해 고장난 부분을 재배치하는 것이 좀 더 많은 계산이 필요하지만 많은 부분에 대한 고장을 재배치 할 수 있는 점에서는 효율적이다<sup>[8~10]</sup>.

본 논문에서는 BIST를 통해 고장난 부분을 찾아낸 후 여분 행/열 메모리를 이용하여 내장 메모리의 고장난 부분을 재배치 할 수 있는 MRI(Memory built-in self Repair Imagination zone) 알고리즘을 제안한다. MRI 알고리즘은 내장 메모리를 가상의 메모리로 분할하고 분할된 각 가상 구역(Imagination Zone)에 주소를 임의로 할당 후 이를 고유의 메인 메모리로 가정하고 재배치한다.

고장난 메모리의 셀이 포함된 행/열 전체를 배치해야 할 경우, 고장이 포함된 행/열에 정상적인 셀들도 함께 재배치되어 재배치가 필요 없는 셀들이 고장난 셀로 인해 재배치가 됨으로써 여분의 메모리 공간이 낭비와 필요없는 연산 소요가 이루어지게 된다.

반면에 각각의 고장난 셀들만을 여분의 메모리로 재배치 할 경우는 정상적인 읽고/쓰기 때의 많은 연산 시간 오버헤드가 발생하게 된다. 따라서 본 논문에서는 적절한 4개의 가상 구역으로 나눈 알고리즘을 제안하고자 한다.

## II. 제안하는 MRI 알고리즘

내장 메모리에 고장난 셀을 포함하고 있는 행이나 열을 모두 재배치하는 방식은 고장이 발생하지 않은 셀 부분까지 여분 메모리에 재배치하는 경우가 발생하여 많은 여분 메모리의 손실을 초래하게 된다. 따라서 이러한 여분 메모리의 손실을 막고 재배치 효율을 극대화하기 위해서 내장 메모리를 4개의 가상 구역으로 분할하는 방식을 연구하였다. 각각의 가상 구역은 독립적인 내장 메모리로 가정하고 재배치 알고리즘을 적용시킨다.

### 1. BISR 구조

그림 1 은 전체적인 BISR의 구조를 보여주고 있다.

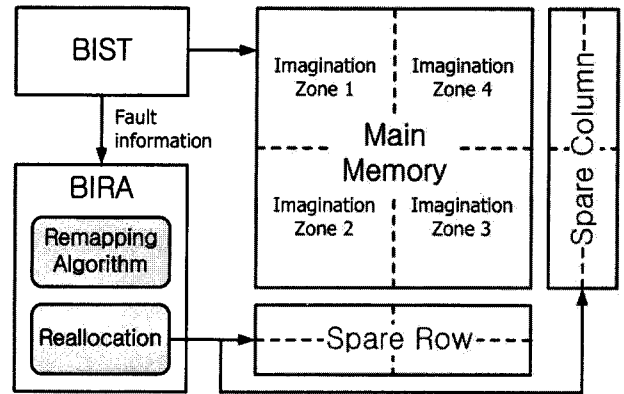


그림 1. BISR 구조  
Fig. 1. BISR Diagram.

BIST를 통해 메모리의 고장을 확인하게 되면, 그 주소와 고장 정보를 재배치 알고리즘에 전달한다. BIRA에서는 메모리를 가상 구역으로 나누고 각 구역에 대한 주소를 임의로 할당 하며 이들을 고유의 메인 메모리로 인식한다. 그리고 고장 정보를 이용하여 각 가상 구역에 대하여 재배치 MRI 알고리즘을 수행하게 된다.

### 2. MRI 알고리즘

8x8의 크기를 갖는 내장 메모리를 BIST 실행 후 고장난 부분이 그림 2처럼 발생하였다고 가정한다.

내장 메모리의 주소와 워드 길이를 반으로 하여 내장

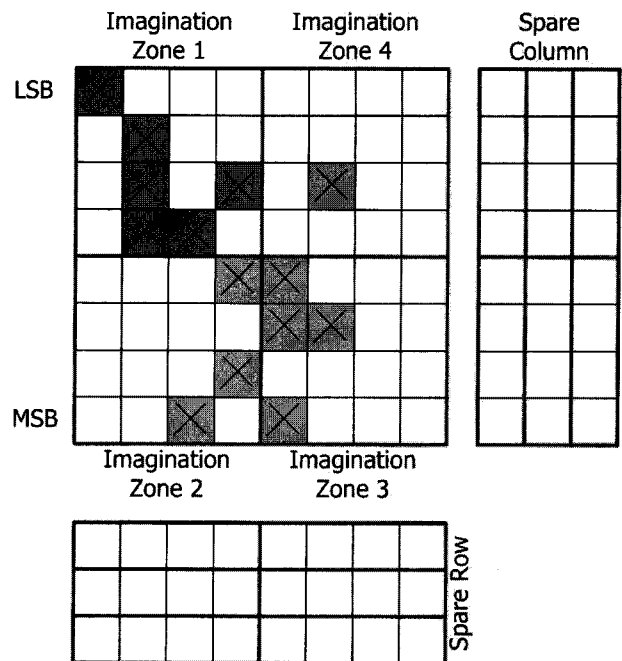


그림 2. 내장 메모리 BIST 후 고장 검출 부분  
Fig. 2. Fault detection sections after embedded memory BIST.

메모리를 4개의 가상 구역으로 나눈다. 여분 메모리의 구역은 행의 경우 워드 길이의 반으로 하고, 열 메모리는 주소의 반으로 구역을 결정한다. 여분 행/열 메모리는 재배치를 위하여 각각 6개씩의 가상 구역을 가진다고 가정한다.

재배치 MRI 알고리즘 순서는 아래와 같다.

- A. 내장 메모리의 주소와 워드를 각각 반으로 나누어 4개의 가상 구역을 만든다.
- B. MRI 알고리즘을 가상구역1에서부터 가상구역4로, LSB 주소에서 MSB 주소로 진행한다.
- C. 4개의 가상구역 중 가장 고장이 많은 가상 구역부터 재배치를 진행한다.
- D. 선택한 가상 구역에서 가장 고장이 많은 셀이 포함된 행/열을 재배치한다.
- E. 재배치된 셀과 같은 행/열에 있는 고장 셀들을 모두 재배치한다.
- F. 고장의 개수가 행과 열이 같으면, 행을 우선순위로 하여 여분 행 메모리에 재배치한다
- G. 남은 고장 셀은 재배치가 가능한 여분 행/열 메모리로 재배치한다.
- H. 하나의 가상 구역에 대하여 재배치가 마무리 되면 그 다음 고장 셀이 많은 구역을 위와 같은 순으로 재배치한다.

그림 2에서의 고장을 MRI 알고리즘으로 적용한다면, 가상 구역 1의 부분에서 가장 많은 고장이 검출되었기 때문에 처음으로 알고리즘을 시작한다.

- 1) 행과 열의 고장 개수를 판단하여 가장 많은 고장이 있는 곳을 찾아 여분 메모리에 재배치한다. 그림 3에서는 2번째 열의 고장 개수가 가장 많기 때문에 여분 열 메모리에 재배치시킨다.
- 2) 재배치시킨 2번째 열의 고장 위치를 포함하고 있는 행이 있는지 비교하여 만약 있다면, 여분 열 메모리로 재배치시킨다. 2번째 열의 고장난 부분은 모두 세 곳이며, row2와 row3, row4의 고장을 포함하고 있다. Row3은 column 2와 column 4의 고장을 가지고 있다. Column 2의 재배치에 따라 row3의 column 2는 재배치되었고, 나머지 column 4도 여분 열 메모리에 재배치되어야 한다.

3) 3번째 열의 고장도 마찬가지로이다. 가장 고장이 많은 부분에 다른 행과 열의 포함부분에 따른 재배치가 마무리 되면 내장 메모리의 주소가 가장 작은 부분부터 고장의 개수를 파악하여 재배치를 진행한다.

4) Row1의 고장은 1개이다. 즉, 행과 열의 개수가 같다. 이 경우는 행에 우선순위를 적용해 여분 행 메모리로 재배치한다.(그림 3)

두 번째로 고장 개수가 많은 가상 구역 3에 대해 재배치 MRI 알고리즘을 실행시킨다. 그림 4처럼 가상 구역 3에서 고장이 가장 많은 column 1부터 column 2까지 순차적으로 고장 메모리 셀들을 여분 열 메모리에 재배치시킨다.

세 번째로 가상 구역 2에 대해 재배치를 진행한다. 고장 개수가 많은 column 4에 대해 여분 열 메모리에 재배치를 시키고, row4, column 3에 위치한 고장 셀은

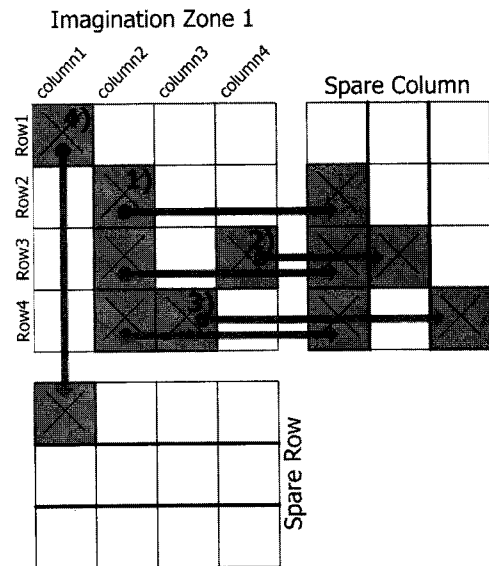


그림 3. 가상 구역 1의 여분 메모리 재배치  
Fig. 3. Spare memory relocation of imagination zone 1.

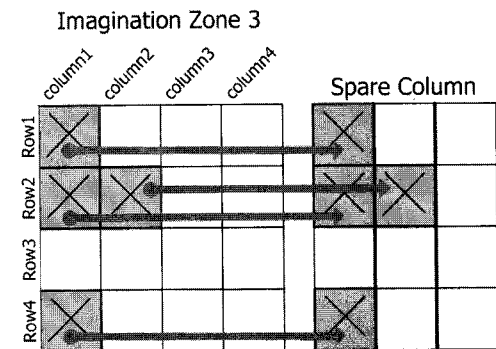


그림 4. 가상 구역 3의 여분 메모리 재배치  
Fig. 4. Spare memory relocation of imagination zone 3.

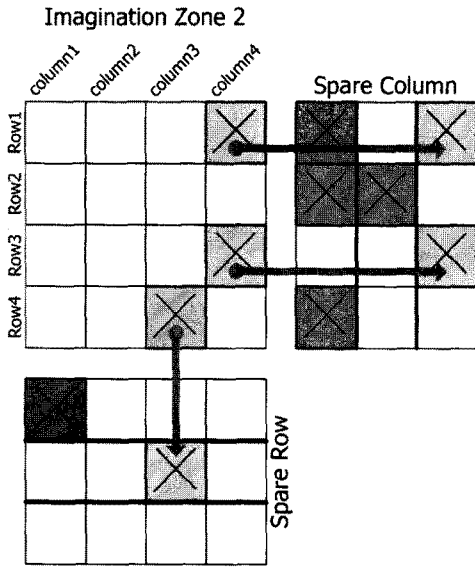


그림 5. 가상 구역 2의 여분 메모리 재배치  
 Fig. 5. Spare memory relocation of imagination zone 2.

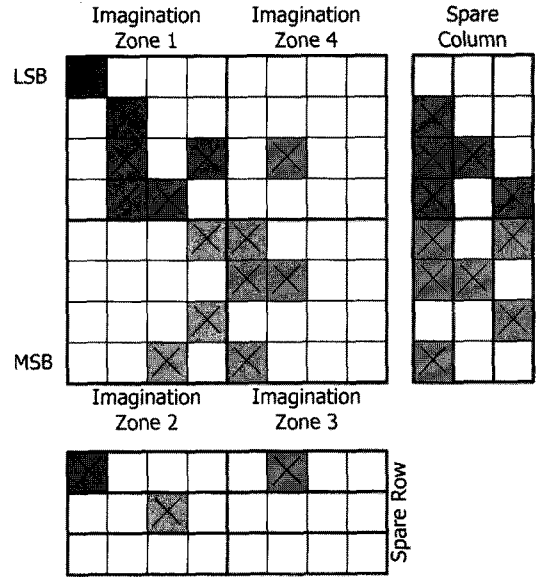


그림 7. MRI 알고리즘을 이용한 재배치  
 Fig. 7. Relocation using MRI algorithm.

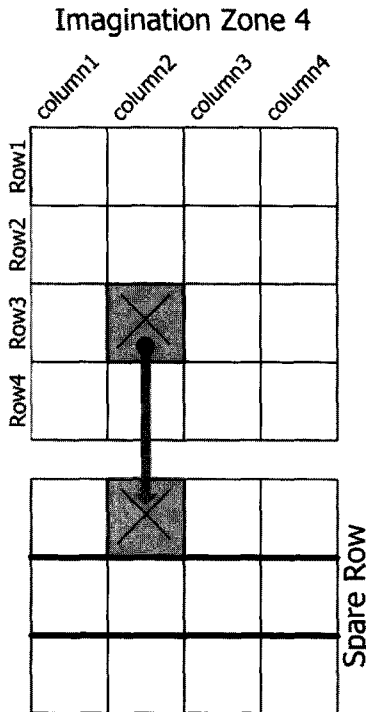


그림 6. 가상 구역 4의 여분 메모리 재배치  
 Fig. 6. Spare memory relocation of imagination zone 4.

여분 열 메모리의 공간이 모두 할당되었기 때문에 여분 행 메모리에 재배치한다.(그림 5)

마지막으로 고장의 개수가 가장 적은 가상 구역 4의 고장 셀 하나는 여분 행 재배치 우선순위에 의해 여분 행 메모리에 그림 6처럼 재배치를 진행한다.

각각의 가상 구역에 대한 재배치를 위와 같이 실행했을 경우 그림 7과 같은 결과를 얻을 수 있다.

### 3. 재배치가 적용된 메모리 동작

그림 7처럼 MRI 알고리즘을 통해 재배치가 이루어진 내장 메모리에 읽기/쓰기가 발생했을 경우 BIRA의 Reallocation 모듈을 통해 여분 행/열 메모리로 재배치가 된 셀인지를 판별한다. 만약 재배치가 이루어진 셀에 접근을 시도한다면, BISR은 내장 메모리와 여분 메모리의 데이터를 통합(읽기 시점), 분산(쓰기 시점)시켜주는 역할을 수행함으로써 외부 시스템 혹은 사용자로 하여금 고장이 없는 메모리를 사용하는 것처럼 보이게 된다. 그림 8은 재배치 구조를 보여주고 있다.

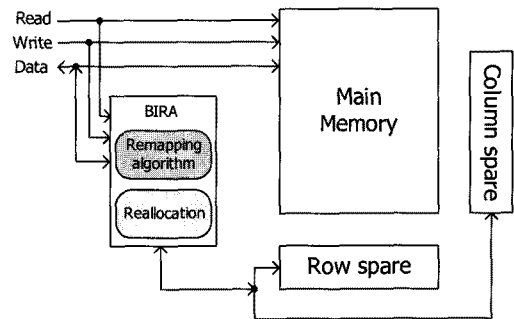


그림 8. 재배치 구조  
 Fig. 8. Relocation diagram.

### III. 실험 결과

본 논문에서 제안한 MRI 알고리즘을 아래의 그림 9

에서와 같이 의사 코드를 통해 알고리즘 재배치율을 검증하였다.

앞에서 설명한 것처럼, 내장 메모리 및 여분 메모리를 Creat\_imagination\_Zone을 이용하여 가상 구역 4개를 만든다. 그리고, 메모리 자체 테스트를 이용하여 메모리 고장이 가장 많은 곳의 가상 구역을 검색한다(Search\_for\_the\_Most\_Faulttest\_Cell\_Zone). 가장 고장이 많은 가상 구역에서부터 여분 메모리에 재배치를 하게 된다. 내장 메모리에서 가장 고장이 많은 가상 구역에서 가장 고장이 많은 행 또는 열을 여분 메모리로 재배치한다. (Faulttest\_cell\_relocation\_row\_or\_column\_spare\_memory).

다른 고장난 셀들 중에서 가장 고장이 많은 셀이 포함된 셀의 위치나 이미 여분 메모리에 재배치된 셀들 중에 일치한 위치에 고장난 셀들 포함된 셀들은 모두 여분 메모리에 재배치한다(Faulty\_Cell\_Row\_or\_Column\_Spare\_Memory\_Included\_with\_Relocation\_Cell). 또한 고장난 내장 메모리의 행과 열의 개수가 같다면, 여분 행 메모리로 재배치한다(A\_Number\_of\_Row\_or\_Column\_is\_All\_the\_same). 위와 같은 조건에 부합되지 않는다면, 행 또는 열 메모리에 재배치를 진행 한다(Relocation\_Row\_Column\_Spare).

고장난 메모리의 셀이 재배치될 공간보다 많이 발생하게 되면, 여분 메모리로 재배치 할 수 없게 된다. 따라서 고장난 메모리의 셀이 여분 메모리의 공간보다 적거나 혹은 일치 하다는 가정하에 실험을 진행하였다.

메모리의 고장이 발생한 후 하나의 가상 구역을 기준으로 순차적으로 가상 구역을 변경하면서 MRI 알고리즘을 실행 했을 경우 재배치율을 아래의 표 1에 나타내었다. 실험에 적용된 메모리의 크기는 메인 메모리 64Mbyte, 여분 행/열 메모리 각각 8Mbyte 이다.

Pseudocode of The MRI Algorithm

```

01: Creat_Imagination_Zone();
02: Search_for_the_Most_Faulttest_Cell_Zone();
03: FOR(the_Most_Faulttest_Cell_Zone: 0<Next_Fault_Cell_Zone: 4--)
04: Faulttest_Cell_Relocation_Row_or_Column_Spare_Memory();
05: IF(Faulty_Cell_Row_or_Column_Spare_Memory_Included_with_Relocation_Cell)
    THEN
06:   All_Faulty_Cell_Is_Relocate_to_Row_or_Column_spare_Memory();
07: IF(A_Number_of_Row_or_Column_is_All_the_same) THEN
08:   The_Priority_Order_Row();
09: ELSE
10:   Relocation_Row_Column_Spare();
    
```

그림 9. MRI 의사코드

Fig. 9. Pseudocode of the MRI algorithm.

MBIST을 이용하여 10%의 메모리 고장이 검출된 후 MRI 알고리즘을 적용시켰다.

본 논문에서 제안하는 MRI 알고리즘은 고장 검출 후 고장이 가장 많이 존재하는 구역에서부터 MRI 알고리즘을 실행하여 재배치한다. 하지만 고장의 개수와 상관없이 순차적으로 구역에 따라 MRI 알고리즘을 적용시켰을 경우는 아래의 표 1과 같이 모든 경우에 있어서 100% 고장 재배치율을 기대할 수 없다.

구역의 순서에 상관없이 MRI 알고리즘을 실행 시켰을 경우 대부분의 고장을 재배치하였지만, 몇몇은 재배치 할 수 없음을 확인하였다. 평균 고장 재배치율은 99.5%이다.

고장을 재배치 할 수 없었던 이유는 여분 메모리의 한쪽 공간으로만 물리는 현상과 가장 많은 고장 셀에 대한 여분 행/열 메모리의 확보가 어려워 재배치 할 수

표 1. 가상 구역 순서에 따른 재배치율  
Table 1. Relocation ratio as zone sequence.

1번째 MRI 적용	2번째 MRI 적용	3번째 MRI 적용	4번째 MRI 적용	재배치율
1	2	3	4	100 %
1	2	4	3	100 %
1	3	2	4	100 %
1	3	4	2	100 %
1	4	2	3	98.2 %
1	4	3	2	100 %
2	1	3	4	100 %
2	1	4	3	100 %
2	3	1	4	100 %
2	3	4	1	100 %
2	4	1	3	95.1 %
2	4	3	1	100 %
3	1	2	4	100 %
3	1	4	2	99.2 %
3	2	1	4	100 %
3	2	4	1	100 %
3	4	1	2	100 %
3	4	2	1	100 %
4	1	2	3	100 %
4	1	3	2	100 %
4	2	1	3	97.6 %
4	2	3	1	100 %
4	3	1	2	100 %
4	3	2	1	100 %

표 2. 가상 구역 2개의 동시 처리에 따른 재배포율  
Table 2. Relocation ratio as 2 zone processing the same time.

1번째 MRI 적용		2번째 MRI 적용		재배치율
1	2	3	4	
1	3	2	4	99.2 %
1	4	2	3	98.2 %
2	3	1	4	100 %
2	4	1	3	95.1 %
3	4	1	2	94.3 %

없었다. 이것을 해결하기 위해서는 여분 메모리의 용량을 늘리는 방법을 선택 할 수 밖에 없다.

가상 구역 2개를 동시에 적용시켰을 경우 가상 구역 하나씩 처리할 때보다는 빠른 처리 속도에 대한 이득은 얻을 수 있으나, 표 2에서와 같이 100% 고장을 재배포치 할 수는 없었다.

이것은 동시에 같은 여분 메모리에 각기 다른 고장 메모리 셀을 재배포하려 할 경우, 충돌로 인한 재배포 문제로 더 이상 재배포 할 수 없는 상황을 야기 시킨다. 이러한 상황을 피하려면 순차적 가상 구역에 대한 재배포 치만이 해결 가능하다.

#### IV. 결 론

본 논문에서는 SoC에서 내장 메모리가 고장이 발생했을 경우를 대비하여 여분 메모리를 두어 내장 메모리에 고장이 발생한 셀을 여분 메모리에 재배포 할 수 있도록 MRI 알고리즘을 제안하였다.

내장 메모리 및 여분 메모리에 가상 구역을 나누어 각 영역에서 고장이 가장 많은 셀을 중심으로 행 또는 열에 여분 메모리로 재배포 후 MRI 알고리즘 조건에 따라 각각 상황에 맞게 여분 메모리로 재배포한다.

이 알고리즘의 효율성을 실험결과를 통해 확인하였으며, 본 논문에서 제안한 재배포 알고리즘인 MRI는 사용자로 하여금 고장이 없는 메모리를 사용하는 것처럼 고장난 메모리를 효율적으로 사용할 수 있도록 한다. SoC 안에서의 내장 메모리 수율을 증가시켜 전체적인 SoC의 수율 증가를 기대할 수 있을 것이라 판단된다.

#### 참 고 문 헌

- [1] Jone W.B., Huang D.C., Wu S.C., and Lee K.J., "An efficient BIST method for distributed small buffers," IEEE Transactionson Very Large Scale Integration Systems, vol. 10, pp. 512-515, August 2002
- [2] Hamdioui S., and van de Goor A.J., "Thorough testing of any multiport memory with linear tests," IEEE Transactionson Computer-Aided Designof Integrated Circuits and Systems, vol. 21, pp. 217-231, February 2002
- [3] Chin Long Wey, and Lombardi F., "On the Repair of Redundant RAM's," IEEE Transactionson Computer-Aided Design of Integrated Circuits and Systems, vol. 6, pp. 222-231, March 1987
- [4] Wei Kang Huang, Yi Nan Shen, and Lombardi F., "New approaches for the repairs of memories with redundancy by row/column deletion for yield enhancement," IEEE Transactionson Computer-Aided Design of Integrated Circuitsand Systems, vol. 9, pp. 323-328, March 1990
- [5] Schober V., Paul S., and Picot O., "Memory built-in self-repair using redundant words," Proceedings of International Test Conference, pp. 995-1001, 30 October-1 November 2001
- [6] 최호용, 서정일, 차상록, "2차원 여분 메모리를 이용한 내장 메모리의 자가치유회로 설계," 대한전자공학회논문지, 제44권 SD편, 제12호, 54-60쪽, 2007년 12월
- [7] 강일권, 강성호, "Sign Bit을 사용한 고효율의 메모리 자체 수리 회로 구조," 대한전자공학회 논문지, 제 44권 SD편, 제 12호, 85-92쪽, 2007년 12월
- [8] Horiguchi M., Etoh J., Aoki M., Itoh K., and Matsumoto T., "A flexible redundancy technique for high-density DRAMs," IEEE Journal of Solid-State Circuits, vol. 26, pp. 12-17, January 1991
- [9] Ilyoung Kim, Zorian Y., Komoriya G., Pham H., Higgins F.P., and Lewandowski J.L., "Built in self repair for embedded high density SRAM," Proceedings of International Test Conference, pp. 1112-1119, 18-23 October 1998
- [10] Heon Cheol Kim, Dong Soon Yi, Jin Young Park, and Chang Hyun Cho, "A BISR (built-in self-repair) circuit for embedded memory with multiple redundancies," International Conference on VLSI and CAD, pp. 602-605, 26-27 October 1999.

저 자 소 개



박 재 흥(학생회원)  
 1999년 숭실대학교 컴퓨터학부  
 학사 졸업.  
 2002년 숭실대학교 대학원  
 컴퓨터학과 석사 졸업.  
 2002년~현재 숭실대학교 대학원  
 컴퓨터학과 박사 과정.

<주관심분야 : 메모리 테스트, VLSI 설계 및 테  
 스트, 컴퓨터구조, 임베디드 시스템>



장 훈(정회원)  
 1987년 서울대학교 전자공학과  
 학사 졸업.  
 1989년 서울대학교 대학원  
 전자공학과 석사 졸업.  
 1993년 University of Texas at  
 Austin 박사 졸업

1993년 Motorola Inc., AMCU Design Center,  
 Senior Staff Engineer.

1994년~현재 숭실대학교 컴퓨터학부 정교수  
 <주관심분야 : 컴퓨터구조, 메모리 테스트, VLSI  
 설계 및 테스트, 임베디드 시스템>



심 은 성(학생회원)  
 2003년 한서대학교 컴퓨터  
 정보학과 학사 졸업.  
 2005년 숭실대학교 대학원  
 컴퓨터학과 석사 졸업.  
 2005년~현재 숭실대학교 대학원  
 컴퓨터학과 박사 과정.

<주관심분야 : 메모리 테스트, VLSI 설계 및 테  
 스트, 컴퓨터 구조, 임베디드 시스템>