

논문 2010-47SD-11-5

최적 수리효율을 갖는 다중 블록 광역대체 수리구조 메모리를 위한 자체 내장 수리연산회로

(A Built-in Redundancy Analysis for Multiple Memory Blocks with
Global Spare Architecture)

정 우 식*, 강 성 호*

(Woosik Jeong and Sungho Kang)

요 약

최근의 메모리 반도체에 있어서, 수율과 품질을 유지하기 위하여 불량셀은 반드시 수리가 필요하다. 대부분의 워드단위 입출력을 갖는 system-on-chip (SoC)를 포함한 많은 메모리가 다중 블록으로 구성되어 있음에도 불구하고, 기존의 대부분의 자체내장수리연산회로의 연구들은 단일블록을 대상으로 하였다. 워드 단위 입출력 메모리의 특성상 다중메모리 광역대체수리구조를 갖는 경우가 많다. 본 논문에서는 이러한 메모리를 대상으로 기존에 최적 수리효율을 갖는 대표적인 자체내장 수리연산회로인 CRESTA를 기본으로 하여, 보다 적은 면적으로 최적 수리효율을 낼 수 있는 알고리즘과 연산회로를 제안한다. 제안하는 자체내장수리 회로는 단위블록의 연산결과를 순차적으로 비교하여 워드단위 메모리의 제약조건을 만족시키는 최종 수리해를 구해내며, 기존의 회로보다 훨씬 빠른 시간 내에 최적의 수리 해를 구해 낼 수 있다.

Abstract

In recent memories, repair is an unavoidable method to maintain its yield and quality. Although many word oriented memories as well as embedded memories in system-on-chip (SOC) consists of multiple local memory blocks with a global spare architecture, most of previous studies on built-in redundancy analysis (BIRA) algorithms have focused on single memory block with a local spare architecture. In this paper, a new BIRA algorithm for multiple blocks with a global spare architecture is proposed. The proposed BIRA is based on CRESTA which is able to achieve optimal repair rate with almost zero analysis time. In the proposed BIRA, all repair solutions for local memory blocks are analyzed by local analyzers which belong to each local memory block and then compared sequentially and judged whether each solution can meet the limitation of the global spare architecture or not. Experimental results show that the proposed BIRA achieves much faster analysis speed compared to previous BIRAs with an optimal repair rate.

Keywords : 메모리, 자체내장수리연산회로, 테스트, 다중블록, 광역대체구조

I. 서 론

반도체 기술이 진보함에 따라 자료를 저장하는 메모리 소자의 대용량화, 고성능화가 빠르게 진행되고 있다. 최근의 메모리 소자의 양산에 있어서 수율과 품질을 만족시키기 위해서는 불량셀을 정상적인 예비 셀로 대체

하여 수리하는 방식의 채택이 불가피하다. 대부분의 SoC (System-on-a-chip)의 내부에 사용되는 내장메모리에 대해서는 외부의 비싼 테스트장치를 이용하여 테스트와 수리(Repair)를 진행하는 것이 많은 비용이 요구되어짐에 따라, 자체 내장 테스트 회로 (Built-in Self Test)를 통한 불량 정보 취득과 자체 수리연산 회로 (Built-in Redundancy Analysis)를 이용한 수리 방법이 많이 사용되어 왔다. 자체 수리연산 회로에 대한 많은 기존연구가 있었으나, 지금까지는 대부분 단일블록에

* 정회원, 연세대학교 전기전자공학과

(Department of Electrical and Electronic
Engineering, Yonsei University)

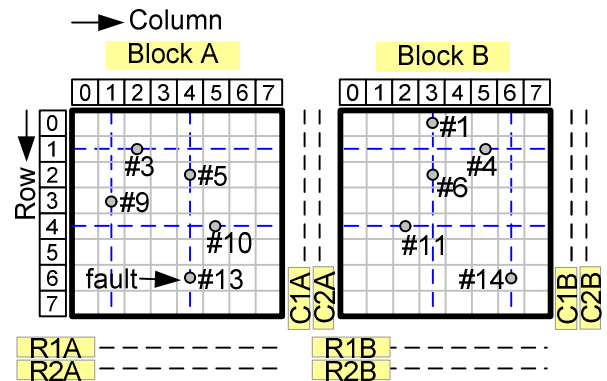
접수일자: 2010년8월23일, 수정완료일: 2010년10월9일

대한 수리해 (Repair solution)를 구하는 방법에 많이
 치중되어 왔다^[1~5]. 많은 정보를 빠른 시간에 처리해야
 하는 요구에 맞추어, 대부분의 메모리소자는 다수의 입
 출력 핀을 통해 동시에 많은 정보를 쓰고 읽을 수 있도
 록 되어 있다. 이러한 상황에서 각각의 입출력 핀을 통
 해 동시에 쓰여 지거나 읽혀지는 정보는 메모리소자 내
 에서 서로 다른 메모리 블록에 확보된 물리적으로 독립
 적인 저장 공간에 기록되어 진다. 또한 단품메모리소자
 중에서 범용메모리로 가장 많이 사용되는 DRAM
 (Dynamic Random Access Memory)이나 Flash 메모리
 등도 다수의 입출력 핀을 통해 동시에 정보를 쓰고 읽
 도록 되어 있다. 한 번에 동시 입/출력되는 정보를 담는
 메모리 셀들은 동일한 워드라인 주소를 공유하게 된다.
 불량 셀을 대체하여 수리하기 위한 예비 셀들도 이와
 같은 구조를 가짐에 따라서, 워드라인을 공유하는 다수
 의 블록에 대하여 동시에 대체되는 다중블록 동시 대체
 구조를 갖는 것이 유리하다. Program의 load와 연산영
 역으로 주로 사용되는 DRAM의 경우, 불량 셀이 하나
 라도 있게 되면 시스템의 오류를 야기하게 되므로, 자
 체내장 수리연산 회로의 수리효율이 높아야만 더 높은
 수율과 품질을 얻을 수 있게 된다. 기존에 발표된 단일
 블록용 자체내장 수리연산 회로 중 CRESTA는 완벽한
 수리효율을 가지는 것으로 알려져 있다^[1]. 하지만, 다중
 블록 동시 대체구조에 대해서는 아직까지 완벽한 수리
 효율을 갖는 자체내장 수리 연산 회로가 보고되지 않았
 다. 본 논문에서는 CRESTA를 기본으로 하여 다중블록
 동시 대체구조에 대해서 빠른 시간 내에 완벽한 수리효
 율을 갖는 자체내장 수리연산 회로를 제안하고자 한다.

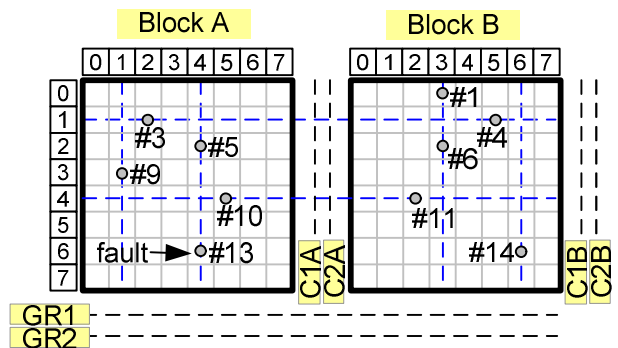
II. 다중 블록 메모리의 수리연산

1. 다중 블록 메모리의 예비셀 대체구조

우선 다중 블록을 갖는 메모리에서의 예비셀 대체구
 조에 대하여 설명하고자 한다. 메모리 소자의 예비 셀
 대체구조는 크게 1차원 예비셀과 2차원 예비셀, 그리고
 블록단위 예비셀로 나뉘게 된다^[4]. 최근의 대부분의 내
 장메모리를 포함한 상용 메모리들은 다양한 경우에서의
 불량에 대하여 대체가 가능한, 2차원 예비 셀 대체 구
 조를 많이 채택하고 있다. 메모리의 저장 공간은 x 주
 소와 y 주소의 조합으로 이루어지는 2차원 평면으로 볼
 수 있다. 이 경우, 같은 y 주소를 가지는 여러 불량 셀
 을 동시 대체 가능한 예비열 (column spare line)과 같



(a) 2 local column spares and 2 row spares per each block



(b) 2 local column spares per each block and 2 global row spares

그림 1. 다중 블록 메모리에서의 지역대체구조와 광역 대체구조의 예

Fig. 1. Example of multiple memory blocks with local spare architecture and global spare architecture.

은 x 주소를 가지는 여러 불량 셀들을 동시 대체 가능
 한 예비행 (row spare line)을 모두 가진 메모리 대체
 구조를 2차원 대체구조라고 한다. 그림 1a는 각 블록의
 불량만을 대체 가능한 2개의 지역 예비열(local spare
 column line)과 2개의 지역 예비행(local spare row
 line)을 갖는 2개의 블록으로 구성된 메모리의 예이다.
 그림 1b는 각 블록의 불량만을 대체 가능한 2개의 지역
 예비열과 모든 블록을 동시에 대체하는 2개의 광역 예
 비행(global spare row line)을 갖는 2개의 블록으로 구
 성된 메모리의 예이다. 그림 1a와 그림 1b의 예 에서는
 예비행이 지역 대체만 가능한가 아니면 광역 대체가 가
 능한가의 차이만 있을 뿐, 불량 발생 정보를 포함하여
 그 외는 모두 동일한 상황을 가진다.

2. 지역대체 구조의 수리연산

그림 1a와 같은 지역대체구조를 갖는 메모리에 대하

여 완벽한 수리효율을 가지며 최적해를 가장 빨리 구해 내는 수리연산 알고리즘은 CRESTA이다^{1,4}. CRESTA는 메모리 블록이 가용한 예비행과 예비열의 개수에 대하여 발생될 수 있는 모든 경우의 수를 동시에 연산하도록 되어 있다. 어떠한 단일 메모리 블록에 대하여 예비행과 예비열의 수를 각각 R_S 와 C_S 라고 할 때, 수리 가능한 모든 경우의 수는

$$\{(R_S+C_S)!/(R_S!*C_S!)\} \quad (1)$$

이며, 이진 검색 트리 (binary search tree)로 표현 가능하다². 그림 1a의 2개의 블록은 각각 2개의 예비열 (=2)과 2개의 예비행(=2)을 가지므로 존재하는 모든 경우의 수는 $\{(2+2)!/(2!*2!)\}$, 즉 6이 된다. 그림 2는 지역 대체구조를 가지는 그림 1a의 메모리 예에 대한 이진 검색 트리를 나타낸다. 예비행과 예비 열을 각각 R과 C라고 하면, 그림 1a의 예에 대해서는 그림 2의 이진 검색 트리에 대해 RRCC, RCRC, RCCR, CRRC, CRRC, CCCR의 6개의 가지(branch)가 존재한다. 하나의 가지가 소유하는 마디(node)의 개수는 (R_S+C_S) 이며, 그림 2의 경우 각각의 가지는 4개의 마디를 가진다^{2,4}.

CRESTA는 그림 2의 이진검색트리를 그대로 하드웨어로 구현한 것이어서 모든 가지가 동시에 병렬적으로 동작한다. CRESTA의 주소 저장방식은 이미 결정되어 있는 이진 검색트리를 따라서 불량주소의 행 주소(row address)와 열 주소(column address)중 어느 것도 이미 저장된 주소와 일치되는 것이 없을 경우에는 비어있는 가장 위쪽의 마디에 정보를 저장한다. 만약, 불량 주소가 기존에 저장된 주소 정보와 일치하게 되면 아무것도 저장하지 않는다. 해당 메모리에 대한 모든 테스트가 진행되는 동안 불량이 발생될 때마다 이러한 과정을 지속적으로 거치게 된다. 모든 테스트가 종료된 시점에서, 지금까지 발생한 모든 불량정보를 모두 저장하고 모자라지 않은 가지가 하나 이상이면 해당 메모리 블록은 수리가능(Repairable)판정이며, 그러한 가지들의 마디에 저장된 불량주소 정보가 수리해 (Repair solution)가 될 수 있으며, 복수의 수리해가 존재 할 수 있다. 모든 가지가 수리불가능한 경우에 한하여 해당블록은 수리불가능(Unrepairable)판정이 된다. 그림 1a의 블록A에 발생된 불량셀의 주소를 발생 순서에 따라 (행,열)의 형태로 표시하면 (1,2) -> (2,4) -> (3,1) -> (4,5) -> (6,4) 이다. 그림 2의 블록A의 첫 가지(branch)는 RRCC이다. 블록A에 대한 첫 불량 주소는 (1,2)이며, 마디순서에 따

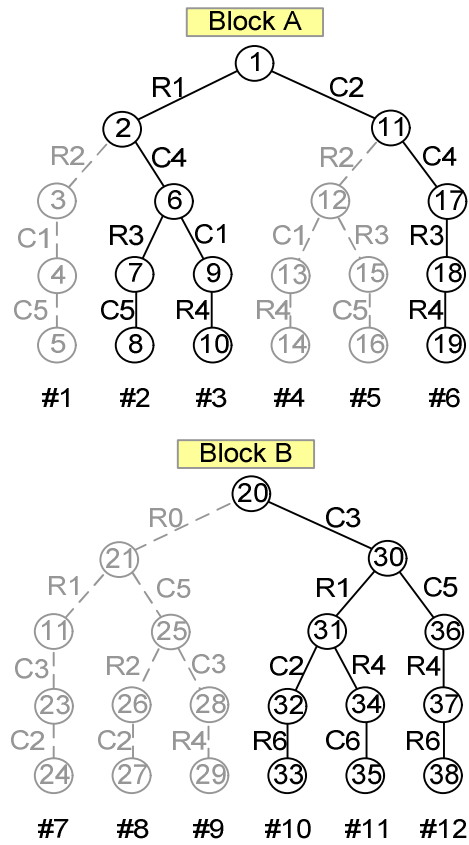


그림 2. 지역대체구조를 갖는 그림 1의 예에 대한 이진 검색 트리

Fig. 2. Binary search tree for the memory example of Fig. 1a with a local spare architecture.

라서 행주소인 R1이 제일 처음에 기록된다. 두 번째 주소인 (2,4)는 이미 저장된 R1과 행주소가 일치하지 않으므로, R2가 기록된다. 동일하게 (3,1)과 (4,5)도 행주소가 일치하지 않으므로, 순서에 따라 열주소로 C1과 C5가 저장된다. 이제 더 이상 주소 저장 공간이 없는 상태에서 마지막 불량 주소인 (6,4)도 기존 저장주소와 일치하지 않으므로, 블록A의 첫 번째 가지는 수리가 불가능 하다. 그림 2를 보면, 블록 A는 #2, #3, #6의 가지들이, 블록 B에 대해서는 #10, #11, #12의 가지들이 수리해가 존재한다.

3. 광역대체 구조의 수리연산

그림 1b는 예비열 광역대체구조를 갖는 다중블록 메모리를 나타낸다. CRESTA의 수리연산 알고리즘을 그림 1b에 적용하여 모든 경우의 수를 나타내려면, 다중 블록에 속한 모든 예비열과 예비행을 고려하여야 한다. 이 경우 2개의 예비행($R_S=2$)과 4개의 예비열($C_S=4$)을 가지게 되며, 식(1)에 따라서 모두 $(2+4)!/(2!*4!)=15$ 개의

가지가 존재하며, 각각의 가지는 6개의 마디를 갖게 된다. 지역대체구조에서는 CRESTA가 테스트 종료와 동시에 수리가능여부 및 최적 수리해를 구할 수 있는 것과는 달리, 광역대체 구조에 대해서는 예비열은 해당 메모리 블록에서만 사용되어야 한다는 제한조건을 테스트 종료 후에 추가로 판정하면 올바른 수리해를 구할 수 있다. 또 하나의 단점은 예비행과 예비열의 개수 증가에 따라 면적이 급격하게 증가하게 된다는 것이다. 그림 1a와 그림 1b의 총 마디수는 각각 $48 \{=6(\text{가지}) \times 4(\text{마디}) \times 2(\text{블록})\}$ 과 $90 \{=15(\text{가지}) \times 6(\text{마디})\}$ 으로서, 지역대체구조인 그림 1a에 비해서 광역대체구조를 갖는 그림 1b가 약 2배 이상 많은 저장 공간을 필요로 하게 된다.

III. 제안하는 수리연산회로

1. 수리연산회로의 주소저장구조

앞장에서 살펴 보았 듯이, 다중블록 광역대체구조에서의 수리해를 구하기 위해서는 테스트 종료 후 추가적인 연산이 필요하다. 수리연산회로의 주소저장구조의 형태에 따라서 테스트 종료 후 수행하는 추가연산이 달라지게 된다. 불량 주소를 저장하기 위한 공간은 빠른 비교를 하기에 적합한 CAM(Content Addressable Memory)을 주로 사용한다^[6]. 하지만, CAM은 보통의 로직회로에서 정보저장용으로 사용되는 레지(Latch)나 레지스터(Register)보다 많은 면적을 차지하므로, 결국 주소저장을 필요로 하는 data의 비트수를 적게 하여 면적부담을 최소화 하는 것이 필요하다. 본 논문에서는 소모 면적을 최소화하기 위해서, 그림 1a와 같은 지역대체구조와 동일하게 주소를 저장하며, 이때 기존의 CRESTA에서 중복되는 저장 공간을 제거하여 훨씬 면적 효율을 높인 RCRESTA를 접목하였다^[5]. 이를 바탕으로 광역대체 구조에 적합한 수리해만을 선별하는 방식을 제시하여 G-CRESTA (Global CRESTA)를 새롭게 제안한다.

그림 3은 제안하는 수리연산회로의 주소저장구조를 나타낸다. 이와 같은 구조를 가짐으로써 기존의 CRESTA보다 작은 공간으로 CRESTA와 동일한 결과를 얻을 수 있다. 그림 3b는 주소저장을 위한 각각의 CAM의 개념적인 구조를 나타내며, 새로 발생한 불량 주소와 이미 저장된 불량주소를 비교하기에 효율적이다. 각각의 Sub-analyzer는 이진연산트리의 하나의 가지(branch)를 나타낸다. 각각의 CAM중 R(C)로 시작하

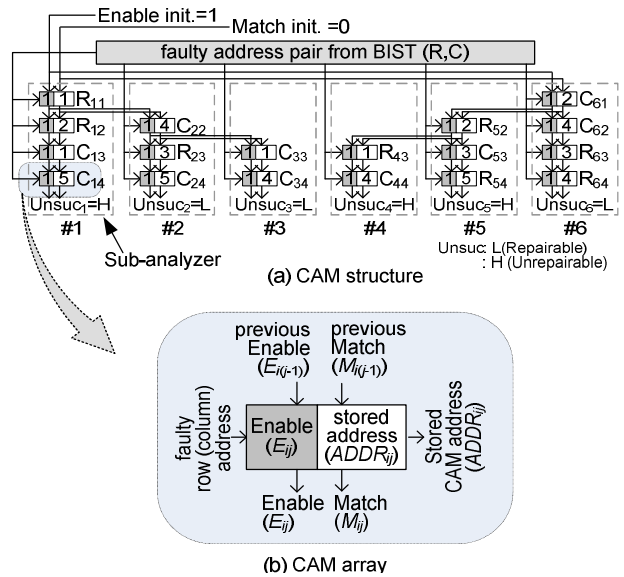


그림 3. 제안하는 수리연산회로의 주소저장구조
Fig. 3. Fault collection CAM structure of the proposed BIST.

는 것은 예비행(열)을 저장하는 것을 의미하며, $R(C)_{ij}$ 는 i 번째 가지의 j 번째 마디를 나타낸다. 그림 3a에 표현된 것과 같이 각각의 가지의 마지막 CAM의 Match(M_{ij})값이 "L"값을 가지는 경우에 대해서 수리가능 판정이 된다. 이 그림에서는 그림 2a의 블록A에 대한 저장 모드를 나타낸다.

2. 최종 수리연산 판정 회로

일단 단일 메모리 블록에 대해서 지역대체구조와 같은 형태로 불량주소가 저장되면, 테스트 종료와 함께 단일 메모리 블록에 속한 가지(branch)들에 대한 수리가능여부가 판정이 된다 (1차 판정). 이를 가지고 광역대체구조에 맞는 최종 수리 연산해를 구하기 위해서는 다음과 같은 구속조건을 만족하여야 한다. 모든 블록의 수리를 위해서는 각 블록에 대하여 지역대체구조로 수리가능판정이 된 가지의 수리해를 블록 당 하나씩 조합하여 최종 수리해를 구하여야 한다. 또한 그렇게 만들어진 최종 수리해에서 사용된 예비행 들의 총합이 광역대체구조에서 사용가능한 총 예비행의 수보다 같거나 작아야 한다 (2차 판정). 이러한 제약사항을 알기 위해서는 각각의 가지에 해당하는 마디 중 예비행(spare row)의 정보를 블록별로 비교하여야 한다. 모든 가지에 대한 예비행의 조합을 순서대로 비교하는 것은 블록의 개수가 늘어나거나, 예비행의 개수가 많은 메모리 블록의 경우에 시간이 너무 많이 걸리게 된다. 반면에 시간을

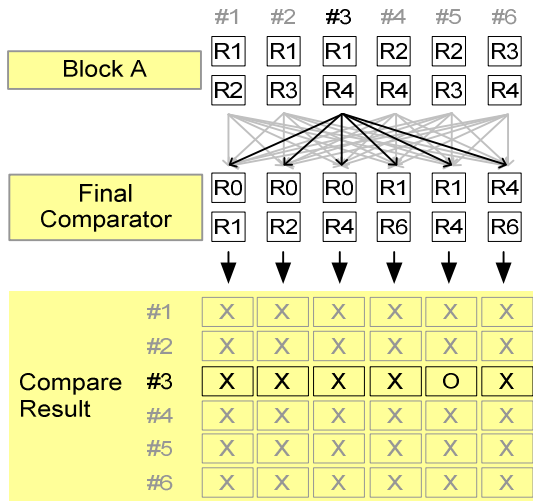


그림 4. 제안하는 최종수리가능여부 판정의 예
Fig. 4. The example of the proposed final repairable judgement.

줄이기 위해서, 모든 조합의 경우를 한 번에 처리하기에는 이를 회로로 구성하기 위한 부담이 너무 커지게 된다. 결국, 처리시간과 면적의 사이에서 적절한 절충점을 찾아야 한다. 본 논문에서는 하나의 블록에 해당되는 가지의 개수 만큼을 동시에 처리할 수 있는 기본 비교 단위로 설정하였다.

그림 4는 그림 2와 같이 1차판정이 완료 후 2차 판정을 하는 방식을 보여준다. 그림 4의 Final Comparator는 모두 12개의 CAM으로 구성하였으며, 여기에는 블록 B의 예비행 주소를 그대로 복사한다. 이후 블록 A의 각각의 가지의 예비행 주소를 순서대로 Final Comparator에 저장된 블록 B의 예비행 주소와 비교하여 저장하도록 한다. 블록 A의 가지를 #1에서 #6의 순으로 차례로 하나씩 Final Comparator에 비교하여 저장공간이 모자라지 않는 경우는, 블록 A의 세번째 가지(#3)와 Final Comparator의 5번째 (즉, 블록 B의 5번째 가지 #11)의 조합뿐이다. 결국, 블록 A의 세 번째 가지와 블록 B의 5번째 가지의 조합을 통해서 그림 1b와 같은 광역 대체 구조 메모리를 수리하기 위한 최종 수리해가 하나 존재 함을 알 수 있다. 수리해는 광역 예비행 (R1, R4)과 블록A의 예비열 (C1, C4) 그리고 블록 B의 예비열 (C3, C6)이다. Final Comparator로 블록 B를 복사하기 위해서는 예비행의 개수(=2) 만큼 클럭이 필요하며, 각각의 가지 수 만큼을 비교할때도 각각 예비행의 개수(=2) 만큼의 클럭이 필요하여 도합 14클럭이 소요됨을 알 수 있다.

III. 실험

광역대체구조를 갖는 다중블록 메모리에 대하여, 기존의 CRESTA 알고리즘과 제안하는 G-CRESTA의 특성을 비교하기 위하여 필요 정보저장 비트수와 최종연산에 필요한 클럭수를 비교하였다. 수리가능 확률 (Repair rate)에 대해서는 CRESTA와 제안하는 방법이 서로 차이가 없이 존재하는 모든 수리해를 찾을 수 있음을 실험을 통해 알 수 있었다. 자체수리연산회로의 면적에 있어서, 정보저장에 필요한 비트수 만큼의 CAM을 필요로 하며 이것이 면적의 대부분을 차지하게 되므로 많은 논문에서 면적비교를 이것으로 대신하여 왔다^[2~4]. 그림 5는 단위블록이 1024행과 1024열을 가지는 두 개의 블록으로 구성된 메모리에 대하여 예비행과 예비열의 개수를 변화시키며 필요정보 저장공간을 비교한 것이다. CRESTA에 비하여 제안하는 알고리즘이 실험구간을 종합하여 평균 약 3.5배정도 적은 용량을 가지며 예비행(열)의 개수가 증가할수록 차이는 커지는 것을 알 수 있다. 그림 6은 단위블록이 1024행과 1024열을 가지며 두 개의 예비행과 두 개의 예비열을 가지는 상황에서 블록의 개수를 변화시키며 필요정보 저장공간을 비교한 것이다. 그림 5의 결과와 유사하게, 제안하는 방법이 기존의 CRESTA에 비해서 실험구간 평균 약 11.9배 적은 용량을 가짐을 보인다. 그림 5와 그림 6의 결과를 통하여, 다중블록 광역 대체구조에서는 예비행(열)의 증가보다 블록의 개수 증가가 면적증가에 훨씬 크게 반영되는 것을 알 수 있다.

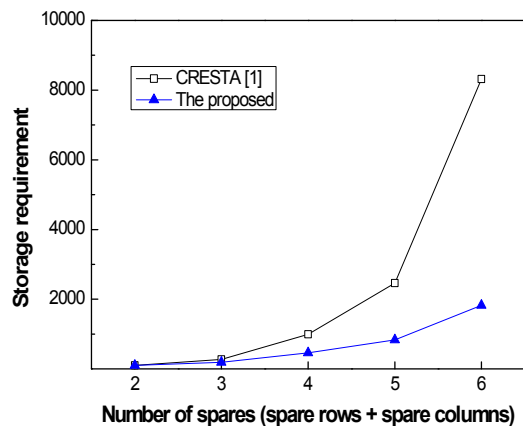


그림 5. 예비행(열)의 개수의 변화에 따른 필요 정보저장공간의 비교 (1024x1024 두블록)
Fig. 5. Comparison of storage requirement with different number of spares for two 1024 by 1024 memory blocks.

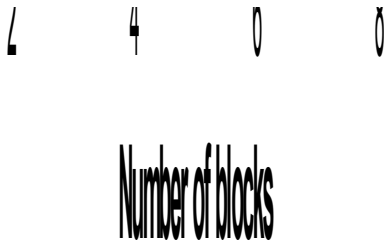


그림 6. 블록의 개수의 변화에 따른 필요 정보저장공간의 비교 (1024x1024, 예비행=2, 예비열=2)
 Fig. 6. Comparison of storage requirement with different number of blocks for 1024 by 1024 memory with two spare rows and two spare columns.

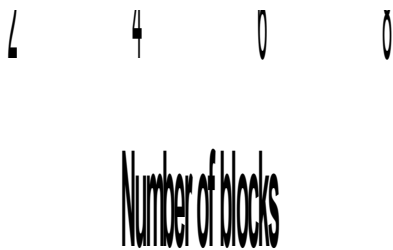


그림 7. 블록의 개수의 변화에 따른 최종 수리확정에 필요한 클럭의 비교 (1024x1024, 예비행=2, 예비열=2)
 Fig. 7. Comparison of clock cycles of final repair judgement with different number of blocks for 1024 by 1024 memory with two spare rows and two spare columns.

그림 7은 두 개의 알고리즘에 대하여 최종 수리해를 구하기 위해 필요한 클럭수를 비교한 것이다. 필요 클럭수가 적을수록 생산성에 이득이 되는 것은 당연하다. 실험 결과를 통해 제안하는 알고리즘이 기존의 CRESTA보다 실험구간 평균 약 15.7배 빠른 시간 안에 최종 수리해를 구하는 것을 알 수 있다.

IV. 결 론

다중블록 광역대체구조를 갖는 메모리에 대하여 완

벽한 수리효율을 가지며, CRESTA보다 실험구간 평균 약 3.5배 이상 적은 면적과 15배 이상 빠른 시간 안에 최종 수리해를 구할 수 있는 G-CRESTA 자체 내장 수리연산 알고리즘을 제안하였다. 이를 통해 고집적 고속의 메모리에 대한 효율적인 자체 내장 수리 방안이 될 수 있을 것이다.

참 고 문 헌

- [1] T. Kawagoe, J. Ohtani, M. Niuro, T. Ooishi, M. Hamada, and H. Hidaka, "A built-in self repair analyzer (CRESTA) for embedded DRAMs" in *Proc. International Test Conference*, pp.567-574, Oct. 2000.
- [2] C.-T. Huang, C.-F. Wu, J.-F. Li, and C.-W. Wu, "Built-in Redundancy Analysis for Memory Yield Improvement," *IEEE Trans. Reliability*, vol. 52, pp.386-399, Dec.2003.
- [3] P. Öhler, S. Hellebrand, and H.-J. Wunderlich, "An Integrated Built-in Test and Repair Approach for Memories with 2D Redundancy" in *Proc. European Test Symposium (ETS)*, pp.91-96, May 2007.
- [4] W. Jeong, I. Kang, K. Jin, and S. Kang, "A Fast Built-in Redundancy Analysis for Memories With Optimal Repair Rate Using a Line-Based Search Tree" in *IEEE Trans. Very Large Scale Integration (VLSI) systems*, vol.17, pp. 1665-1678, Dec. 2009
- [5] W. Jeong, T. Han and S. Kang, "An Advanced BIRA Using Parallel Sub-analyzers for Embedded Memories" *Proceedings of International SOC Design Conference (ISOCC 2009)* pp. 249-252, Nov 2009.
- [6] K. Pagiamtzis, and A. Sheikholeslami, "Content-Addressable Memory (CAM) Circuits and Architectures: A Tutorial and Survey," *IEEE Journal of Solid-State Circuits*, vol. 41, no.3, pp.712-727, Mar. 2006.

 저 자 소 개



정 우 식(학생회원)
 1997년 고려대학교 제어계측
 공학과 학사 졸업.
 1999년 고려대학교 전자공학과
 석사 졸업.
 1999년~현재 하이닉스반도체
 근무 중.

2010년 현재 연세대학교 전기전자공학과
 박사과정.

<주관심분야 : 반도체, SoC 설계, 테스트>



강 성 호(평생회원)
 1986년 서울대학교 제어계측
 공학과 학사 졸업.
 1988년 The University of Texas,
 Austin 전기 및 컴퓨터
 공학과 석사 졸업.
 1992년 The University of Texas,
 Austin 전기 및 컴퓨터
 공학과 박사 졸업.

1992년 미국 Schlumberger Inc. 연구원.

1994년 Motorola Inc. 선임 연구원.

2010년 현재 연세대학교 전기전자공학과 교수.

<주관심분야 : SoC 설계, SoC 테스트>