수율향상을 위한 EDS 공정에서의 CRA 시뮬레이션

한영신*

요약

메모리의 용량이 급속도로 증가함에 따라 결함의 빈도 또한 높아지고 있다. 결함이 있다면 어쩔 수 없겠지만 적은 결함이 발생한 경우에는 해당 다이를 사용안하는 것보다 수리해서 사용하는 것이 메모리 생산 업체 입장에서는 보다 효율적이고 원가 절감 차원에서 필수적이다. 기존의 RA process는 최종적으로 치료할 수 없는 경우도 main 셀의 결함 유형의 분석이 모두 끝난 후에 알 수 있다. 그러나 CRA(Correlation Repair Algorithm) 시뮬레이션은 이미 fail 유형의 분석이 끝나 있기 때문에 Correlate 결과만 만족하면 바로 결함 유형 분석을 마칠 수 있다. EDS(Electrical Die Sort)에서 Test time을 비롯한 RA process time은 비용에 직결된다. 반도체 산업의 특성상 메모리의 용량은 갈수록 높아져 가고 상대적으로 용량 당 단가는 낮아질 수 밖에 없다. 그리고 용량이 높아져 감에 따라 fail 유형도 더욱 다양해 질 수밖에 없기에 CRA 시뮬레이션이 RA process time 절감에 새로운 대안이 될 수 있다고 하겠다.

CRA(Correlation Repair Algorithm) Simulation in EDS for Yield Improvement

Shin-Young Han*

ABSTRACT

As the density of memory chips increases, the probability of having defective components is also increased. However, in case of a small number of defects, it is desirable to reuse a defective die after repair rather than to discard it, because reuse is an essential element for memory device manufactures to cut costs effectively. As for the conventional RA process, it was impossible to know whether a defect could be repaired before the result of the main cell type analysis was obtained. However, in the CRA simulation, a database of each fail type analysis is already in place. Therefore by correlating the data with the fail type we can reduce the whole process of the analysis. In EDS redundancy analysis, time spent in tests and in the RA process is directly connected to cost. Due to the technological developments in the semiconductor industry, memory volume is increasing and the unit price per volume is decreasing. As bigger volume means more various fail types, The CRA simulation will be an effective alternative to save time in the RA process.

Key Words : EDS, Database, Semiconductor Manufacturing Yield, Simulation, CRA

* 성균관대학교 정보통신공학부(✉hanys@skku.edu)
· 제1저자(First Author) : 한영신 · 교신저자(Correspondent Author) : 한영신
· 접수일(2011년 2월 16일), 수정일(1차 : 2011년 3월 17일), 게재확정일(2011년 3월 22일)

Ⅰ. 서 론

VLSI기술 혁신은 반도체 메모리의 고집적화와 대용량화를 이루었지만 불량 발생의 개연성도 상대적으로 증가되어왔다[1]. 메모리의 용량이 급속도로 증가함에 따라 결함의 빈도 또한 높아지고 있으며 메모리 생산에 있어 중요한 요소 중 하나인 수율도 낮아지는 문제가 발생한다. 이 문제를 해결하고 수율을 높이기 위해 메모리를 수리하는 것이 중요한 역할을 차지하게 된다. DRAM의 경우 수많은 미세 셀 중 한 개라도 결함이 있으면 메모리로서 제구실을 하지 못하므로 불량품으로 처리된다. 하지만 DRAM의 집적도가 증가함에 따라 확률적으로 소량의 셀에만 결함이 발생할 확률이 높은데도 이를 불량품으로 폐기한다는 것은 수율을 낮추는 비효율적인 처리 방식이다.

따라서 이 경우 미리 DRAM내에 설치해둔 예비 메모리 셀을 이용하여 불량 셀을 대체시킴으로써 수율을 높이는 방식을 채용한다[2,3,4,5,6]. 본 논문에서 제안한 시뮬레이터는 기존의 RA알고리즘 개념에서 벗어나 결함 유형별로 CRA(Correlation Repair Algorithm)를 진행함으로써 RA에 소요되는 시간을 절약함으로써 메모리의 생산비용을 줄일 수 있다는 것을 보여주는 것이 목적이다.

Ⅱ. 관련 연구

EDS공정에서 웨이퍼 테스트를 마치면 주된 셀의 어느 부분에서 결함이 발생하였는지를 알 수 있다 (즉, 결함의 발생 위치, 행과 열의 주소를 알 수 있음). 이 정보를 바탕으로 redundancy analysis는 다이가 가진 여유 셀을 할당하게 되는데 즉, 디바이스에 발생한 결함을 효율적으로 고치기 위해 디바이스가 가진 여유 셀을 할당하는 과정이 redundancy analysis이다.

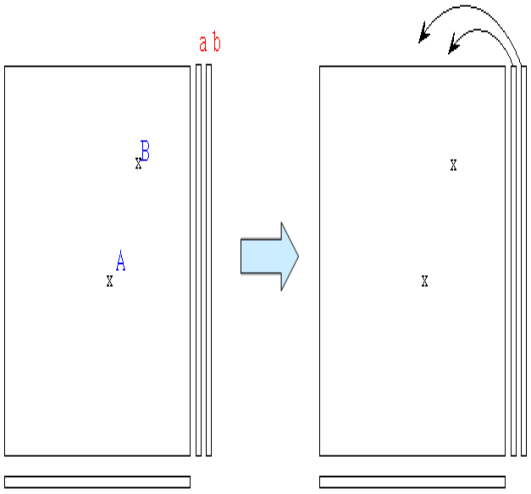

그림 1. 여유 셀로 할당하는 과정
Fig. 1. Repair process with spare cells

그림 1에서 행 방향과 열 방향에 여유 셀을 각각 가진 디바이스의 경우에 결함(A, B)을 고치기 위해 여유 셀 (a, b) 가 사용되고 있음을 보여 준다. 만약에 A, B 결함이 행 방향으로 수평하게 발생하였다면 행 방향 여유 셀 한 개로 치료하는 것이 열 방향 여유 셀 두 개로 치료하는 것 보다 효율적이기 때문에 두개의 열 방향 여유 셀로 치료하지 않고 행 방향 여유 셀 한 개로 치료하였을 것이다. 참고로 결함이 한 개만 발생하였을 경우 행 방향 여유 셀 한 개가 사용되거나 열 방향 여유 셀 한 개가 사용되어도 마찬가지일 것이다. 이 경우 우선순위 개념을 적용해서 대개 여유 셀이 많이 있는 행 또는 열 방향을 우선 사용하게 된다.

RRAM의 치료를 위해 많은 redundancy analysis 알고리즘이 있지만 가장 우선적으로 사용되어지는 것이 repair-most 알고리즘이다[4]. Repair-most 알고리즘은 먼저 각각의 행과 열의 카운터에 결함의 개수를 저장한다.

그림 2에서 행과 열의 카운터에 결함의 값이 저장되어 있음을 볼 수 있다.

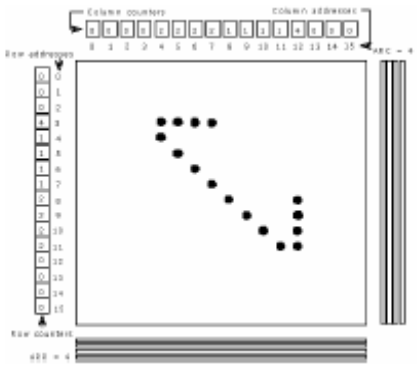

그림 2. 결함 발생 예
Fig. 2. Example of the occurrence of a defect

ARC (available-redundant-columns)와 ARR (available-redundant-rows)은 여유 셀의 개수를 표시하는 카운터이다.

Repair-most 알고리즘은 그림 3에서처럼 가장 카운터 값이 높은 행과 열을 우선적으로 치료한다. 이 과정을 여유 셀을 다 사용하거나 결함이 모두 치료될 때까지 반복한다. 대개 repair-most 알고리즘으로 일반적인 결함을 치료할 수 있으나 그림 4는 repair-most 알고리즘으로 결함을 모두 치료하지 못하는 경우를 보여준다. 이러한 경우 그림 5에서처럼 fault-driven 알고리즘을 사용하면 치료 할 수 있다.

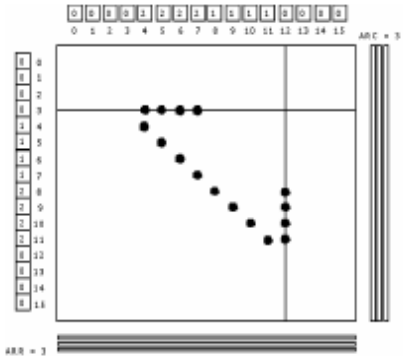

그림 3. 카운터 값이 높은 행과 열을 우선적으로 치료
Fig. 3. Repairing a row a column with the highest value

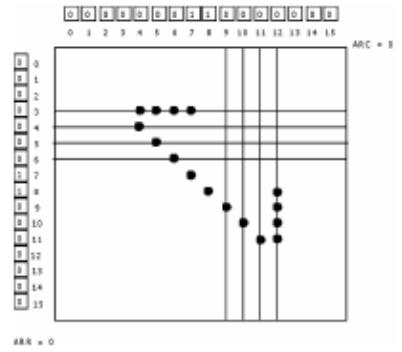

그림 4. Repair-most 알고리즘으로 결함을 모두 치료하지 못하는 경우
Fig. 4. Cells cannot be repaired even with the repair-most algorithm

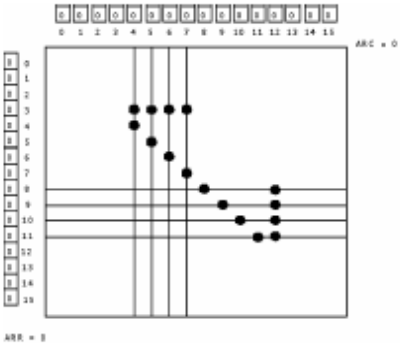

그림 5. Fault-driven 알고리즘
Fig. 5. Fault-driven algorithm

Fault-Driven 알고리즘은 두 단계로 구성되어진다 [5]. 첫 단계는 결함이 존재하는 방향과 같은 여유 셀로 대체되어야 하는 특정 행 또는 열을 결정하는 forced-repair 분석이다. 두 번째 단계는 forced-repair 에 의해 사용되어지지 않은 여유 셀을 사용하여 forced-repair 단계이후 남아있는 결함에 대한 치료 방법을 결정하는 sparse-repair 분석이다.

Fault-Driven 알고리즘은 그림 5처럼 치료하기 위해 레코드(record)가 생성된다. FLCA(Fault Line

Covering Approach) 알고리즘은 그림 6처럼 치료하는데 Fault-Driven 알고리즘의 레코드보다 적은 레코드만 생성될 뿐이다[8].

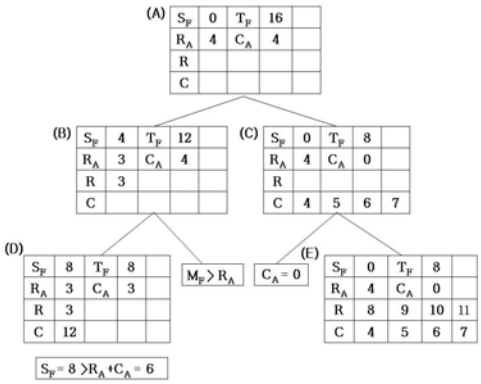

그림 6. FLCA 알고리즘 결과
Fig. 6. Result of the FLCA algorithm

여기서 CA는 ARC, RA는 ARR, TF는 fault RAM에 담겨 있는 전체 결함 수, SF는 1개의 결함으로 이루어진 결함들의 수, MF는 행과 열의 카운터에 담긴 결함 수중에 가장 큰 값을 의미한다. FLCA는 forced repair analysis 후 남아있는 결함들과 fault RAM의 정보를 CA, RA, TF, SF로 분류한 후 큐에 저장한다. FLCA는 단지 2번의 실행만으로 치료 해법을 얻을 수 있음을 그림 6에서 보여준다. 그림 6에서 부(parents) B와 C는 각각 오직 한 개의 자 (descendant)를 가지고 있음에 주목하여야 한다 (MF>RA이고 CA=0이므로). 그림 6에서 부(parent) D는 SF > RA + CA이므로 사용될 수 없다. 그러므로 치료 해법은 그림 6에서 parent E의 레코드로 주어진다.

III. CRA 시뮬레이션 실험 및 결과

3.1 제안하는 CRA 시뮬레이션

기존의 RA 알고리즘은 지수적인 복잡도를 갖거나 필요로 하는 기록인자(record)가 많은 단점을 가지고 있다. 또 최적의 RA결과를 산출하기위해 여러 번 결함 유형을 분석하여야 하지만 CRA는 이미 결함 유형을 데이터베이스에 저장해 둔 다음 유사도가 높은 최적의 RA 결과를 산출하기 때문에 기존 RA 알고리즘의 보완 방법이 될 수 있다고 하겠다. EDS에서 테스트 타임을 비롯한 RA 소요시간은 비용에 직결된다. CRA시뮬레이션은 여유 셀로 치료할 수 있는 모든 경우의 수를 데이터베이스에 저장하고 비교 대상 디바이스의 테스트 후 분석된 결함 패턴의 결함 유형과 데이터베이스에 저장된 결함 유형을 비교하여 평균, covariance, 분산을 구한 후 correlation 값을 추출해 낸다. 시뮬레이션과정은 다음 3단계로 이루어진다.

- 여유 셀로 치료할 수 있는 모든 경우의 수에 해당하는 정보를 생성 및 데이터베이스에 저장시킴.
- 테스트 후 얻어진 결함 패턴 정보를 가진 파일을 로딩(loading)
- 데이터베이스에 저장된 결함 유형과 결함 패턴 정보를 가진 파일의 결함 유형에 대한 CRA 진행

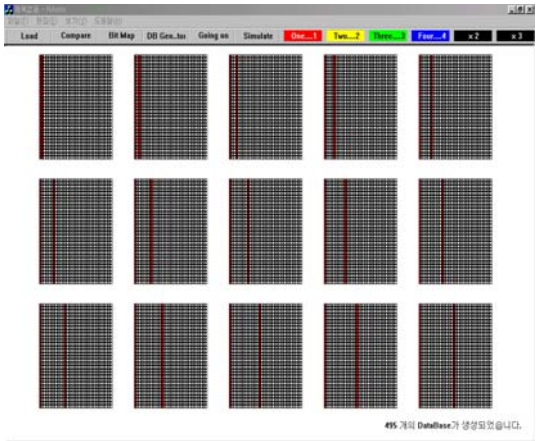

그림 7. 데이터베이스 맵
Fig. 7. Database Maps

데이터베이스에 저장된 결함 유형과 결함 패턴 유형을 비교하여 CRA를 진행한다. CRA는 수리가 가능

하면서 correlation value을 추출한다. 그림 7 은 생성된 치료할 수 있는 모든 경우의 데이터베이스 Maps 과 그림 8에서 CRA을 통해 치료할 수 있으면서 유사도의 기준이 되는 최적의 correlation value를 산출한다. 이번 실험에서 디바이스 크기를 각각 다르게 가정하였기 때문에 결함 패턴 정보 파일은 행 방향으로 열 방향으로 이루어진 정사각형구조를 가진다.

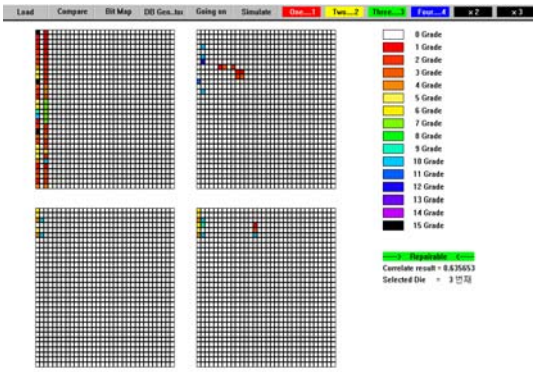

그림 8. 시뮬레이션결과
Fig. 8. Correlation value result

3.2 실험 방법

지금까지 RA는 장비 개발 업체에서 제공하는 경우가 대부분 이었고 각 장비 업체별로 RA알고리즘을 개발하여 제공하여 왔기 때문에 동일한 결함 유형에 분석하는 RA시간이 각 장비 업체 별로 다른 경우가 대부분이었다. 이에 본 실험에서는 랜덤한 10,000개의 결함 패턴을 5, 10, 20개의 군집유형으로 나누어 두고 CRA(Correlation Repair Algorithm)을 시뮬레이션 한다. 개발 환경으로는 Visual C++를 사용했다. CRA를 시뮬레이션하기위한 조건은 다음과 같다.

- 샘플 패턴 : 10,000개의 메모리 어레이(array) 사용
- 디바이스별 크기 (n × n) : 16 × 16, 32 × 32, 64 × 64, 128 × 128, 256 ×256, 512 × 512, 1024 × 1024 의 정사각형 어레이 사용
- Line redundancy 개수(ARR=ARC) : 각각의 디바이스별로 정해 실험
- CRA(Correlation Repair Algorithm)의 입력요소 : 랜덤 결함 패턴

3.3 실험 결과

EDS에서 테스트 시간을 비롯한 RA소요시간은 비용에 직결된다. 그러므로 본 실험결과는 각 디바이스별 크기와 패턴개수에 따라 10,000개의 결함 패턴을 랜덤하게 5, 10, 20개의 군집으로 나누어 CRA알고리즘으로 시뮬레이션 했을 때 수행시간을 실험하였다. 본 실험에서는 디바이스 사이즈가 커지면 데이터베이스 용량이 많아져 군집화 하는 방법을 썼다. 임의로 선택된 K개의 군집을 중심으로 시작하여 각 결함패턴은 가장 가까운 중심의 군집에 할당된다. 새로운 군집에 할당된 결함패턴들의 새로운 중심을 계산하며 결함패턴들이 다른 군집으로 재 할당 되지 않거나 SE (square-error) 값이 감소하지 않을 때까지 위의 과정 반복하여 선택과정에서 수렴에 많은 시간이 소요되는 것을 개선시킨다. 그림 9는 각 디바이스별 CRA 알고리즘 시뮬레이션 결과를 나타낸다.

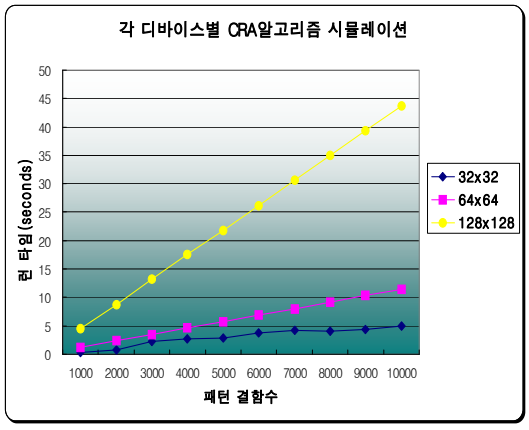

그림 9. 각 디바이스별 CRA 알고리즘 시뮬레이션
Fig. 9. CRA Algorithm Simulation

기존 RA 알고리즘의 수행속도를 보면 불량 셀의 개수에 의해 민감하게 영향을 받아 불량 셀의 수가 늘어나면 수행속도가 많이 걸리는 반면에 본 연구에서 제안하는 알고리즘은 단지 패턴의 수와 셀의 사이즈 크기에 따라 영향을 받기 때문에 CRA은 사이즈 크기를 i이라 했을때 O(i)의 시간 복잡도를 가지고, 또한 사이즈 크기를 j 이라 했을 때 O(j)의 시간 복잡도를 가지므로 제안된 CRA의 효율성을 확인 할 수 있었다. 또한 CRA은 모든 결함의 종류를 구별 하는 것이 가능하여 기존보다 효율적인 불량 분석 업무를 진행할 수 있으며 수율 및 품질 향상에 기초가 될 수 있을 것이다.

Ⅳ. 결 론

반도체 제조 기술의 발달은 더욱 작고 복잡한 회로의 구현을 가능하게 하여, 단위 면적당 메모리 셀 수를 증가시켰다. 메모리의 집적도 또한 계속적으로 증가되고 있다. 그러나 이러한 메모리칩의 평면적 수직적 축소는 복잡하고 정밀한 제조 공정을 요구하게 되고, 이러한 공정을 통하여 완성된 메모리 제품의 신뢰도 및 품질 보장을 위하여 점점 더 복잡하고 정교하고 긴 시간을 요하는 테스트가 필요하게 되었다. 64M 이상의 메모리에서는 테스트가 전체의 생산비용의 40%이상을 차지하게 되어 반도체 메모리에 있어서 테스트 및 수리는 반도체의 가격 경쟁력을 결정하는 중요한 기술이 될 것 이다.

기존의 RA process는 최종적으로 치료할 수 없는 경우도 메인 셀의 결함 유형의 분석이 모두 끝난 후에 알 수 있었다. 그러나 CRA시뮬레이션은 이미 결함 유형의 분석이 끝나있기 때문에 correlate 결과만 만족하면 바로 결함 유형 분석을 마칠 수 있다. 반도체 산업의 특성상 메모리의 용량은 갈수록 높아져 가고 상대적으로 용량 당 단가는 낮아질 수밖에 없고 용량이 높아져 감에 따라 결함 유형도 더욱 다양해 질 수밖에 없기에 CRA 시뮬레이션이 RA소요시간 절감에 새로운 대안이 될 수 있다.

참고문헌

[1] C.-W. Wang, C.-F. Wu, J.-F. Li, C.-W. Wu, T. Teng,K. Chiu, and H.-P. Lin. "A built-in self-test and selfdiagnosis scheme for embedded SRAM." In Proc. *Ninth IEEE Asian Test Symp.* (ATS), pages 45 - 50, Taipei, Dec. 2000.

[2] C.-F. Wu, C.-T. Huang, C.-W. Wang, K.-L. Cheng, and C.-W. Wu. "Error catch and analysis for semiconductor memories using March tests." In Proc. *IEEE/ACM Int. Conf.* Computer-Aided Design (ICCAD), pages 468 - 471, San Jose, Nov. 2000.

[3] R. Rajsuman, "Design and test of large embedded memories", an overview, *IEEE Design & Test of Computers*, pp. 16-27, 2001.

[4] Chih-Tsun Huang, Chi-Feng Wu, Jin-Fu Li, Cheng-Wen Wu, "Built-In Redundancy Analysis for Memory Yield Improvement" *IEEE TRANSACTIONS ON RELIABILITY*, VOL. 52, NO. 4, DECEMBER 2003

[5] Shyue-Kung Lu and Chih-Hsien Hsu, "Fault Tolerance Techniques for High Capacity RAM", *IEEE TRANSACTIONS ON RELIABILITY*, VOL. 55, NO. 2, JUNE 2006

[6] Kawagoe, T., et al.: "A built-in self-repair analyzer (CRESTA) for embedded DRAMs". Proc. 2000 *Test Conf.*, ITC, Atlantic city, NJ, USA, pp. 567 - 573, 2000.

[7] Huang, C.-T., Wu, C.-F., Li, J.-F., and Wu, C.-W.: "Built-in redundancy analysis for memory yield improvement", *IEEE Trans.* Reliab., 52, (4), pp. 386 - 399, 2003.

[8]Y. Zorian and S. Shoukourian, "Embedded-Memory Test and Repair: Infrastructure IP for SoC Yield," *IEEE Design & Test*, vol. 20, no. 3, pp. 58-66, 2003.

[9] R.-F. Huang et al., "A Simulator for Evaluating Redundancy Analysis Algorithms of Repairable Embedded Memories," Proc. *IEEE Int'l Workshop Memory Technology, Design and Testing* (MTDT 02),

IEEE CS Press, pp. 68-73, 2002.
[10] Youngshin Han and Chilgee Lee, "RRAM Spare Allocation in Semiconductor Manufacturing for Yield Improvement," *KES LNAI* 3215, pp. 95 - 102, 2004.

저자소개

한영신(Young-Shin Han)

1997년 이화여자대학교 전산정보 (공학석사)
2004년 성균관대학교 전기전자 컴퓨터공학과(공학박사)

2004년 9월 : 이화여자대학교 컴퓨터그래픽스/가상현실 연구센터 박사후 연구원
2005년 3월 : 성결대학교 멀티미디어학과 전임강사
2007년 8월 : U of A ACIMS센터 Visiting Scholar
2009년3월~현재: 성균관대학교 정보통신공학부 반도체시스템공학과 계약교수
※ 관심분야 : 모델링& 시뮬레이션, 온톨로지, 가상현실