

EDS 공정에서 Redundancy Analysis 시뮬레이션

서준호*, 이철기**

Redundancy Analysis Simulation for EDS Process

Jun-Ho Suh, Chil-Gee Lee

Abstract

It takes 2 or 3 months to manufacture memory device. Defect has to exist owing to hundreds of processes. If there are too many defects, the memory has to be rejected. But if there are a few defects, it will be more efficient and cost reducing for the company to use it by repairing. Therefore, laser-repair process is needed for such a reason and redundancy analysis is needed to establish correct target of laser-repair process. The equipment development company had provided the redundancy analysis and each development company had developed and provided separately. So, to analyze the similar type of defects, redundancy analysis time can be very different by the manufacture. The purpose of this research is to strengthen the competitive price and to apply correlation concept in business for reducing the redundancy analysis time to repair the defects

Key Words: redundancy analysis, correlation, EDS Process, Fail Bit map

* 본 연구는 한국과학재단 목적기초연구(R01-2000-00250)지원으로 수행되었음

** 성균관대학교 정보통신공학부 석사4기

*** 성균관대학교 정보통신공학부 부교수

1. 서론

반도체 기술은 최근 눈부신 기술혁신을 이루어 고도의 정보통신과 정보 처리기술 발전을 가져오고 있으며 그 응용범위는 산업뿐만 아니라 사회, 공공분야나 가정생활에 이르기까지 현대사회의 여러 분야에 확대되고 있다. 특히 우리의 경우 메모리 분야에서 급속한 발전을 이루고 있다.

메모리 분야에서의 칩(chip) 테크놀러지의 발전으로 인해 메모리 제품의 Life-Cycle은 점점 짧아지고 있으며, 신제품의 시장가격은 출하 이후 급속히 떨어지는 경향을 보이고 있으므로 매출과 이익의 극대화를 위해 고 수율의 제품을 가능한 빨리 시장에 출하하는 것이 반도체 업체들의 목표가 되고 있기에 반도체 공정에서의 개선은 필수적인 과제가 되어 가고 있다.

반도체 메모리 공정기간은 2~3개월 정도, 공정은 수백 가지에 이를 정도로 많기에 결함(defects)은 존재할 수밖에 없다. 치료(repair)를 할 수 없을 정도로 많은 결함이 있다면 어쩔 수 없겠지만 적은 결함이 발생한 경우에는 해당 다이 (die - wafer상에서의 device 기준 단위)를 제외시키는 것보다는 치료해서 사용하는 것이 메모리 생산 업체 입장에서는 보다 효율적이고 원가 절감 차원에서 필수적이다. 즉, VLSI(Very Large Scale Integration)와 WSI(Wafer Scale Integration)와 같은 통합(integration) 기술로 인해 큰 용량의 메모리 대량 생산이 가능하게 된 지금 Redundancy는 메모리 칩의 제조와 결함이 있는 셀(faulty cells)을 지닌 디바이스(device)를 치료하는데 광범위하게 사용되어져 왔다. 이런 유형의 메모리를 RRAM (Redundant Random Access Memory)라 부른다. Redundancy는 여유 행(spare rows)과 열 (columns)로 분할 된 여유 셀 (spare cells)로 구성되어지는데 여유 행과 열은 결함이 있는 셀이 있는 행과 열을 대체하는데 사용되어진다. Redundancy를 사용할 수 있게 하는 공정을 laser-repair라고 하는데 이 공정은 결함이 있는 디바이스의 수율을 증가시킴으로써 결국 비용 절감에 상당한 기여를 한다.

웨이퍼(wafer)상태의 테스트에서 RRAM을 치료하는데 많은 시도가 있었다. Repair Most는 메모리의 모든 행과 열의 결함 카운터(fault counters)를 사용하여 치료할 수 있는 행과 열의 위치정보(address)를 얻고자 하였다.[1] Fault-driven comprehensive redundancy algorithm [2]은 모든 가능한 조합으로부터 최적의 치료 결과(optimal repair-solution)를 구현하기 위하여 사용자 정의의 선호도(user-defined preferences)에 의존한다. Efficient spare allocation in reconfigurable arrays [3]에서는 RRAM의 치료를 위해서 새로운 2개의 알고리즘을 제시하였다. 첫 번째 알고리즘은 치료 공정에서 초기에 걸러내는(screening) branch-and-bound 접근방식(approach)이고 두 번째 알고리즘은 heuristic 기준(criterion)을 사용하는 것이다.

redundancy analysis 시물레이션은 기존의 redundancy analysis 알고리즘의 개념에서 벗어나 결함 유형별로 시물레이션 한 후 redundancy analysis를 진행함으로써 redundancy analysis에 소요되는 시간을 절약함으로써 원가 경쟁력 강화를 하는데 목적이 있다.

본 논문 구성은 다음과 같다. 2장에서는 반도체 공정에 대해 설명하고 3장에서는 Redundancy Analysis에 대해 설명하고 4장은 Correlation의 정의에 대해 설명한다. 5장에서는 Simulation을 수행하고 분석하였다. 마지막 6장에서는 연구결과와 향후 연구 방향을 제시하였다.

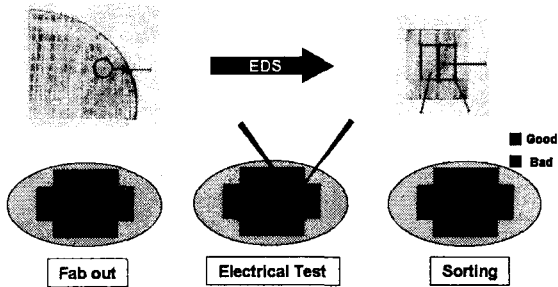
2. 반도체공정

반도체 공정은 크게 4가지 step으로 나눌 수 있다. FAB(Fabrication) 공정, EDS(Electrical Die Sorting) 공정, 조립(Assembly) 공정과 Package Test 공정이다.

FAB공정은 실제 Wafer에 회로를 만드는 공정으로 반도체 제조의 시작 공정이라 할 수 있다.

EDS 공정은 FAB에서 만들어진 웨이퍼를 전기적 검사를 통해 각각의 다이들을 점검하여 합격(Good) 또는 불합격(Bad)으로 분류(sorting)하

는 공정이다.



<그림 1> EDS 공정

<그림 1>에서 왼쪽의 Fab out은 다이의 동작 여부가 결정되지 않은 상황을 의미하고 중간의 Electrical Test는 전기적 신호를 probe (cf. 탐침의 일종)를 통해 웨이퍼 상의 다이 내에 있는 패드(pad)에 가하는 것을 표현한 것이다. Electrical Test에서 얻은 정보 - 주된 셀(main cell)의 결함 - 를 가지고 Laser-repair공정에서 여유 셀로 치료하게 된다. 최종적으로 오른쪽에 있는 sorting은 전기적 테스트를 통해 밝혀진 다이의 합격 또는 불합격상태를 확정시키는 과정의 표시이다.

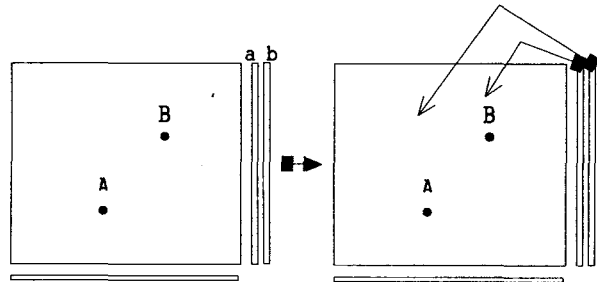
조립 공정은 EDS 공정에서 합격한 다이를 모듈(Module)로 구현하기 전의 칩(chip)으로 조립하는 공정이다. Package Test공정은 칩의 특성을 테스트하는 공정으로 EDS공정의 테스트보다 좀더 상세하고 열악한 조건에서 테스트를 진행하여 사용자에게 제품의 품질을 보장하게 된다.

3. Redundancy Analysis

3.1 Redundancy Analysis정의

EDS공정에서 웨이퍼 테스트를 마치면 주된 cell의 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 cell을 할당하게 되

는데 즉, 디바이스에 발생한 결함을 효율적으로 고치기 위해 디바이스가 가진 여유 cell을 할당하는 과정이 redundancy analysis이다.



<그림 2> 여유 cell로 치료하는 과정

<그림 2> 행쪽 여유 cell이 한 개, 열쪽 여유 cell을 두 개 가진 디바이스의 경우에 결함(A,B)를 고치기 위해 여유 cell (a,b)가 사용되고 있음을 보여 준다. 만약에 A, B 결함이 행쪽으로 수평하게 발생하였다면 행쪽 여유 cell 한 개로 치료하는 것이 열쪽 여유 cell 두 개로 치료하는 것 보다 효율적이기 때문에 두개의 열쪽 여유 cell로 치료하지 않고 행쪽 여유 cell 한 개로 치료하였을 것이다. 참고로 결함이 한 개만 발생하였을 경우 행쪽 여유 cell 한 개가 사용되거나 열쪽 여유 cell 한 개가 사용되어도 마찬가지일 것이다. 이 경우 우선도(priority) 개념을 적용해서 대개 여유 cell이 많이 있는 행 또는 열 방향을 우선 사용하게 된다.

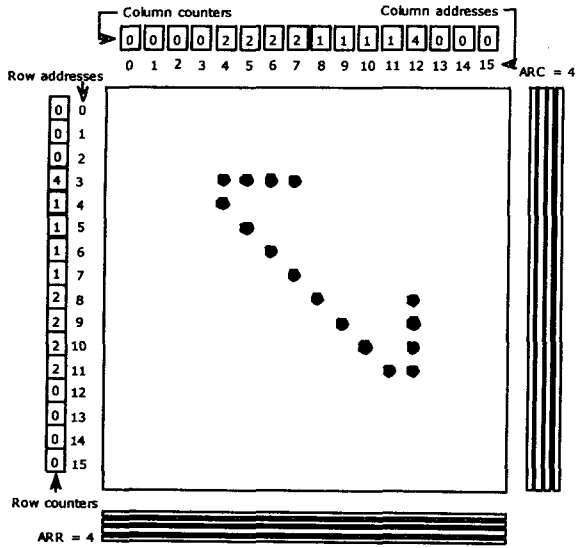
3.2 Redundancy Analysis 알고리즘

RRAM의 치료를 위해 많은 redundancy analysis 알고리즘이 있지만 가장 우선적으로 사용되어지는 것이 repair-most 알고리즘이다. [4]

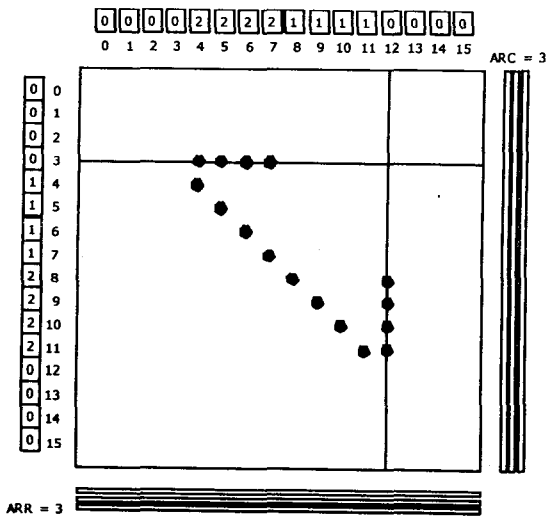
repair-most 알고리즘은 먼저 각각의 행과 열의 카운터에 결함의 개수를 저장한다.

<그림 3>에서 행과 열의 카운터에 결함의 값이 저장되어 있음을 볼 수 있다.

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



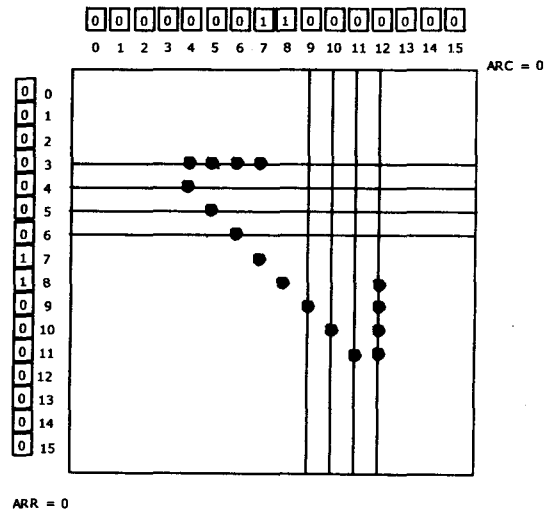
<그림 3> 결함의 발생 예



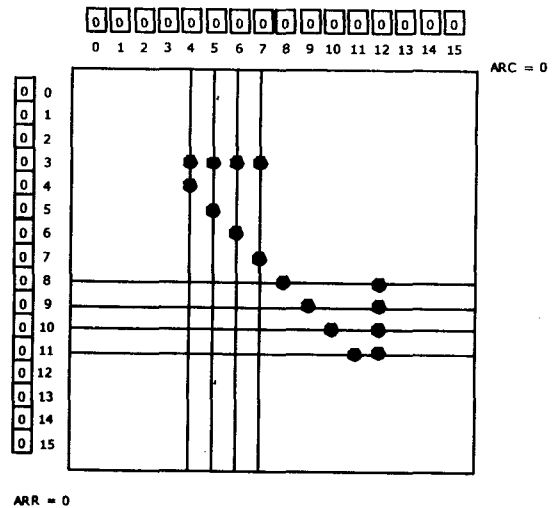
<그림 4> 카운터 값이 가장 높은 행과 열에 여유 cell을 할당하여 치료

repair-most 알고리즘은 <그림 4>에서처럼 가장 카운터 값이 높은 행과 열을 우선적으로 치료한다. 이 과정을 여유 cell을 다 사용하거나 결함이 모두 치료될 때까지 반복한다. 대개

repair-most 알고리즘으로 일반적인 결함을 치료할 수 있으나 <그림 5>는 repair-most 알고리즘으로 결함을 모두 치료하지 못하는 경우를 보여준다.



<그림 5> repair-most 알고리즘으로 치료하지 못한 예



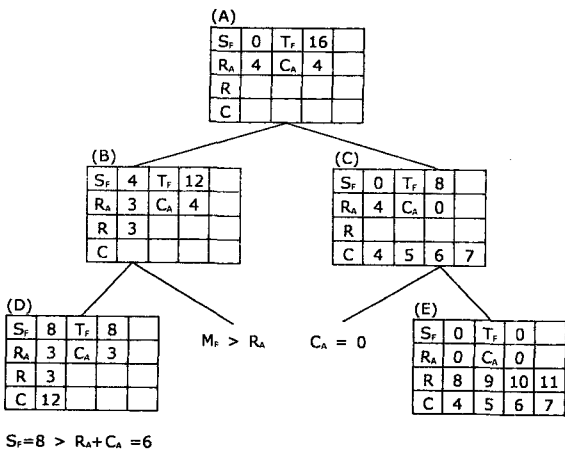
<그림 6> Fault-Driven 알고리즘

이러한 경우 <그림 6>에서처럼 fault-driven 알고리즘을 사용하면 치료할 수 있다.

Fault-Driven 알고리즘은 두 단계로 구성되어 있다.[5] 첫 단계는 결함이 존재하는 방향과 같은 여유 cell로 대체되어야 하는 특정 행 또는 열을 결정하는 forced-repair 분석이다. 두 번째 단계는 forced-repair에 의해 사용되어지지 않은 여유 cell을 사용하여 forced-repair 단계 이후 남아 있는 결함에 대한 치료 방법을 결정하는 sparse-repair 분석이다.

Fault-Driven 알고리즘은 <그림 6>처럼 치료하기 위해 $2^{(T_F+1)} - 1$ 개의 레코드(record)가 생성된다. FLCA(Fault Line Covering Approach) 알고리즘은 <그림 6>처럼 치료하는데 Fault-Driven 알고리즘의 레코드보다 적은 $2^{(\frac{T_F - S_F}{\min} + 1)}$ - 1의 레코드만 생성될 뿐이다.[6]

여기서 R_A 는 행의 여유 cell의 개수, C_A 는 열의 여유 cell의 개수, T_F 는 전체 결함 수, S_F 는 단독 결함 (single fault)의 수를 의미한다.



<그림 7> FLCA 알고리즘에 의한 결과

FLCA는 단지 2번의 실행만으로 치료 해법을 얻을 수 있음을 <그림 7>에서 보여준다. <그림 7>에서 부(parents) B와 C는 각각 오직 한 개의 자(descendant)를 가지고 있음에 주목하여야 한다. ($M_f > R_a$ 이고 $C_a = 0$ 이므로) <그림 7>에서 부

(parent) D는 $S_f > R_a + C_a$ 이므로 사용될 수 없다. 그러므로 치료 해법은 <그림 7>에서 parent E의 record로 주어진다

4. Correlation 정의

독립적인 random 변수 X와 Y의 평균은 random 변수의 central tendency에 대한 측정 방법이 될 수 있는데 아래와 같은 성질을 가지고 있다.

- ① $E(cX) = cE(x)$ (c는 상수)
- ② $E(X+Y) = E(x) + E(Y)$
- Var(X)라고 표시되는 random 변수 X의 분산은 평균에 대한 random 변수의 dispersion의 measure인데 아래와 같이 표현된다.
- ③ $Var(X) = E[(X-u)^2] = E(X^2) - u^2$
(u는 평균값)
- 또한 아래와 같은 성질을 가지고 있다.
- ④ $Var(cX) = c^2 Var(X)$ (c는 상수)
- ⑤ $Var(X+Y) = Var(X) + Var(Y)$
(X,Y가 독립적일 때)

Cov(X,Y)라고 표시되는 random 변수 X와 Y의 covariance는 X와 Y의 의존적인 측정값이고 아래와 같이 표현된다.

$$\begin{aligned} \text{⑥ } Cov(X,Y) &= E\{[X - E(X)][Y - E(Y)]\} \\ &= E(XY) - E(X)E(Y) \end{aligned}$$

일반적으로

- ⑦ $Var(X-Y) = Var(X) + Var(Y) - 2 Cov(X,Y)$
X와 Y가 독립적이면
- ⑧ $Var(X-Y) = Var(X) + Var(Y)$

Cor(X,Y)라고 표시되는 random 변수 X와 Y의 correlation은 다음과 같다.

$$Cor(X,Y) = \frac{Cov(X,Y)}{\sqrt{Var(X)Var(Y)}}$$

또한 correlation은 $-1 \leq Cor(X,Y) \leq 1$ 로 표시될 수 있다. [7]

여기서 $Cor(X,Y)$ 값이 1에 근접할수록 유사도

가 높은 것을 의미한다. 본 논문에서 유사도가 높다는 것은 비교 대상이 되는 fail 유형들이 거의 흡사하여 치료될 수 있는 가능성이 높다는 것을 의미한다.

5. Redundancy Analysis Simulation

5.1 RA 시물레이션의 목적

RA시물레이션은 기존의 RA알고리즘으로 이미 분석된 치료 방법을 correlation기법을 사용하여 적용함으로써 RA 소요시간을 최소한으로 줄이는 것이 목적이다. 즉, RA시물레이션을 통해 산출된 correlate limit을 사용하여 디바이스의 결함을 치료하는 것이 기존의 RA 알고리즘을 사용하여 결함을 치료하는 것 보다 시간상으로 적게 소요됨으로써 메모리의 생산비용을 줄일 수 있다는 것을 보여준다.

5.2 실험계획

개발 환경으로는 Visual C++ / Window 98 을 사용했다. RA 시물레이션 구현의 효율화를 위해 열쪽 여유 cell만 있고 행쪽 여유 cell은 없는 것으로 하고 열쪽 여유 cell의 개수는 2개로 한다. 디바이스 크기는 행으로 1K (1024 bit), 열로 1K (1024 bit). 즉, 1M bits로 한다. 시물레이션의 input요소는 fail bit map과 임의의 correlate limit이고 최초의 correlate limit은 0.0으로 한다. RA시물레이션을 통해 치료할 수 있으면서 유사도의 기준이 되는 최적의 correlate limit을 산출한다.

5.3 Fail Bit map 정보

파일 안에 있는 16진수 한 개는 32bit의 fail bit 정보를 표시한다. 즉, 16진수 '9'는 이진수로 1001. 즉, MSB에 8bit, LSB에 8bit의 결함이 있고 가운데 2x8 bit에는 결함이 없음을 의미한다

-----1-----2-----3---
12345678901234567890123456789012



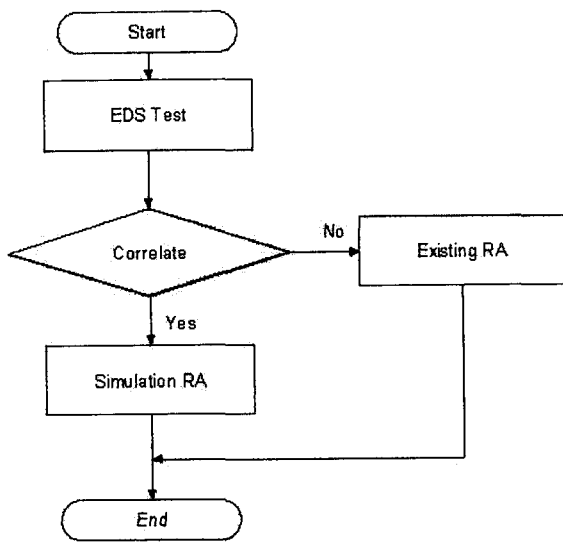
<그림 8> Fail Bit map 정보

이번 실험에서 디바이스 크기를 1024 x 1024로 가정하였기 때문에 fail bit map 정보 파일은 행쪽으로 32개의 16진수 숫자들의 집합, 열쪽으로 32개의 16진수 숫자들의 집합으로 이루어진 정사각형 구조를 가진다. <그림 8>은 fail bit map 정보 file의 한 예인데, 행에서 2번째 열에서 1번째에 'A'라는 정보가 있음을 알 수 있다. 이것을 해석해 보면 'A'는 이진수로 1010을 의미한다. 즉, 16진수 정보를 통해 불량이 발생한 위치 정보와 함께 불량의 양도 알 수 있다. 단, 한정된 문자 개수로 1M 디바이스의 fail 정보를 표시하여야 하기 때문에 8bits (1byte) 의 fail 정보의 압축은 어쩔 수 없다고 하겠다. <그림 8>에서 보면 열쪽으로는 32개의 문자가 있지만 행쪽으로는 7개의 문자밖에 없다. 그러나 실제상으로는 행쪽도 32개의 문자가 있다. 표시만 7개의 문자로 표현한 것뿐이다.

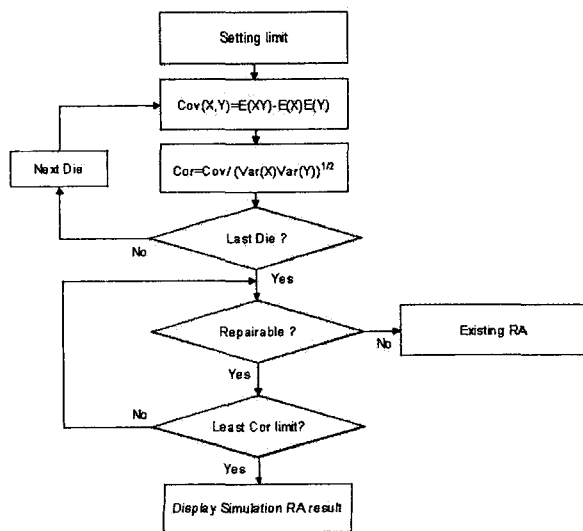
5.4 Correlate 과정

실험 계획에서 여유 cell이 2개인 경우의 디바이스만 고려하기로 하였기에 열쪽 방향의 2개 여유 cell로 치료할 수 있는 모든 경우를 데이터베이스에 저장한다. <그림 11>은 2개의 여유 cell로 치료할 수 있는 모든 경우의 수를 그림으로 표현한 것이다. 비교 대상 디바이스의 테스트 후 분석된 fail bit map의 fail 유형과 <그림 11>

의 데이터베이스에 저장된 fail 유형을 비교하여 평균, Covariance, 분산을 구한 후 Correlation 값을 추출해 낸다. 상기의 비교 과정 중 여유 cell의 필요 개수가 3개 이상이면 치료할 수 없기에 repairable flag를 거짓으로 분류하여 RA를 진행하지 않도록 함으로써 시간을 절약한다.



<그림 9> Redundancy Analysis Simulation의 전체 flow

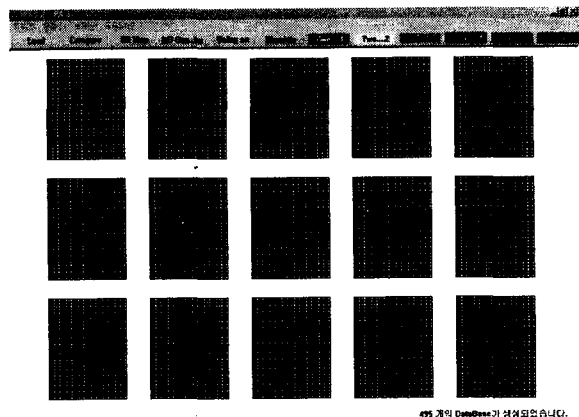


<그림 10> Correlate의 상세 flow

<그림 9>와 <그림 10>의 flow에서는 correlate 과정에서 이전에 설정된 correlate limit보다 적으면서 치료될 수 없으면 기존 RA과정으로 분기되는 것을 보여준다. 다른 관점에서 보면 이전에 설정된 correlate limit보다 크면 (유사도가 높으면) RA 시뮬레이션으로 분기되어 최종적으로 치료할 수 있으면서 최소의 correlate limit을 추출해내는 과정을 보여 준다. 여기서 correlate limit의 최초 값은 0.0이나 각 다이 별로 평균, covariance, 분산, correlation 값을 구하고 치료여부를 확인한 후 최적의 correlate limit (치료할 수 있으면서 correlate limit이 가장 작은 값)이 얻어진다. 가장 작은 값의 correlate limit을 구하는 이유는 가능한 많은 경우의 fail 유형을 치료할 수 있는 유사도의 기준을 설정하기 위해서이다. 만약 유사도가 높은 즉, correlation 값이 1에 근접한 기준을 설정하게 되면 조금 유사도가 떨어져도 치료할 수 있는 fail 유형의 경우를 버리게 되는 현상이 일어난다.

5.5 RA 시뮬레이션의 진행 순서

- 2개의 여유 cell로 치료할 수 있는 모든 경우의 수에 해당하는 정보를 생성 및 데이터베이스에 저장시킴.



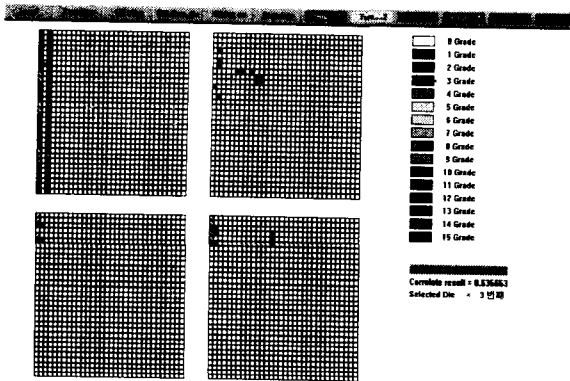
<그림 11> 생성된 치료할 수 있는 모든 경우의 데이터베이스 Maps

-테스트 후 얻어진 Fail Bit map 정보를 가진 파일을 로딩(loading)

-Simulate button을 눌러 데이터베이스에 저장된 fail 유형과 fail bit map 정보를 가진 파일의 fail 유형에 대한 Correlate process를 진행

5.6 시물레이션 실험 결과

시물레이션 결과를 분석하기 위해서 우측 상단에 있는 x2 button과 One...1, Two...2, Three...3, Four...4 button을 사용하여 개별적으로 map을 선택하면 우측 하단에 선택된 map의 Correlate 결과를 표시한다. 그리고 Repairable이면 녹색, Unrepairable이면 노란색으로 표시된다. 또한 오른쪽 상단에 있는 색별 단계(grade)는 fail bit수와 각 결합 위치에서의 fail 유형을 의미한다. (1단계인 빨간색은 LSB에서만 8 bits fail 되었음을 의미하고, 7 단계인 녹색은 MSB에서만 8 bits fail 되었음을 의미한다. 15 단계인 검정색은 4x8 bits fail 즉, 32 bits fail이 발생하였음을 의미한다.)



<그림 12> 시물레이션 결과

<그림 12>에서 보여지는 시물레이션 결과를 보면 4개의 fail bit map 중 치료가능하면서 correlate 값이 최소 값을 가지는 것은 좌측 하단에 있는 map (즉, 3번째 Correlate 값이 0.635653 인 것)임을 알 수 있다. 다시 말하면 치료할 수 있는 최적의 Correlate limit이 산출

되었음을 알 수 있다.

1 번째의 Correlate Value = 1.000000 R
2 번째의 Correlate Value = 0.645531 U
3 번째의 Correlate Value = 0.594451 U
4 번째의 Correlate Value = 0.544803 U
5 번째의 Correlate Value = 0.651499 R
6 번째의 Correlate Value = 0.593629 R
7 번째의 Correlate Value = 0.580835 R
8 번째의 Correlate Value = 0.609243 R
9 번째의 Correlate Value = 0.544803 U
10 번째의 Correlate Value = 0.613858 U
11 번째의 Correlate Value = 0.552430 U
12 번째의 Correlate Value = 0.526863 U

위 자료는 12 다이의 fail 유형과 데이터베이스에 저장된 fail 유형과의 시물레이션 결과를 나타낸다. <그림 12>에서 우측 하단 부분만을 보여준 것이다. (12개의 map을 표현한 그림은 map이 작게 표현되어 식별이 어려운 관계로 우측 하단 부분만 보여줌)

시물레이션 결과를 보면 <그림 11>의 데이터베이스에 저장된 fail 유형과 유사하여 치료할 수 있는 경우는 1번째 (1.0), 5번째 (0.651499), 6번째 (0.593629), 7번째 (0.580835), 8번째 (0.609243)이고 이중 최소의 correlate limit은 7번째의 0.580835이다. 1 번째 Correlate value가 1이 나온 것은 <그림 11>의 데이터베이스에 저장된 fail 유형 495개 data 중 어느 한 경우와 동일한 경우에 해당하기 때문이다.

6. 결론

시물레이션 결과로 얻어진 치료할 수 있으면서 최소의 correlate 값을 가지고 EDS의 Redundancy Analysis에 적용하면 거의 0 에 가까운 RA time을 얻을 수 있다. 기존의 RA 알고리즘은 매년 fail 유형을 분석해서 최적의 RA

결과를 산출해야 하지만 RA 시뮬레이션은 이미 fail유형을 데이터베이스에 저장해 둔 다음 유사도가 높은 최적의 RA 결과를 산출하기 때문이다. 기존의 RA process는 최종적으로 치료할 수 없는 경우도 main cell의 fail 유형의 분석이 모두 끝난 후에 알 수 있다. 그러나 RA 시뮬레이션은 이미 fail 유형의 분석이 끝나 있기 때문에 Correlate 결과만 만족하면 바로 fail 유형 분석을 마칠 수 있다. EDS에서 Test time을 비롯한 RA process time은 비용에 직결된다. 반도체 산업의 특성상 메모리의 용량은 갈수록 높아져 가고 상대적으로 용량당 단가는 낮아질 수밖에 없다. 그리고 용량이 높아져 감에 따라 fail 유형도 더욱 다양해 질 수밖에 없기에 RA 시뮬레이션이 RA process time 절감에 새로운 대안이 될 수 있다고 하겠다.

참고문헌

- [1] Memories and redundancy techniques in *IEEE ISSCC Dig. Tech. Papers*, pp. 80-87, 1981
- [2] J. R. Day, A fault-driven comprehensive redundancy algorithm for repair of dynamic RAMs *IEEE Design & Test*, vol. 2, no. 3, pp.33-44, 1985
- [3] S-Y. Kuo and W. K. Fuchs, Efficient spare allocation in reconfigurable arrays, *IEEE Design & Test*, vol. 4, pp. 24-31, 1987
- [4] M. Tarr, D. Boudreau, and R. Murphy, "Defect Analysis System Speeds Test and Repair of Redundant Memories," *Electronics*, Jan. 12, 1984, p.175
- [5] John R. Day, "A Fault-Driven Comprehensive Redundancy Algorithm," *IEEE Design & Test*, vol. 2, No. 3, pp. 35-44, 1985.
- [6] Fabrizio Lombardi and W. K. Huang, "Approaches for the repair of VLSI/WSI RRAMs by Row/Column deletion," *Proc. 18th Int. Symp. on Fault-Tolerant Computing*, pp. 342-347.
- [7] Averill M. Law, W. David Kelton : Simulation Modeling and Analysis, Third Edition, Chap4. 235-260 (2000)

● 저자소개 ●



서준호

1995 연세대학교 세라믹공학학사

2002 성균관대학교 정보통신공학부 석사4기

현재 삼성전자(주) 근무

관심분야 : 컴퓨터 시뮬레이션, 객체지향 모델링, 전문가 시스템



이철기

1980 성균관대학교 전자공학과 졸업

1979~1983 한국방송공사 근무

1985 Arizona State University 전기 및 컴퓨터 공학과 석사

1990 University of Arizona 전기 및 컴퓨터 공학과 박사

1990~1995 삼성전자 수석연구원

1995~현재 성균관대학교 전기전자 및 컴퓨터 공학부 조교수

관심분야 : 컴퓨터 시뮬레이션, 객체지향 모델링, 공장자동화, 전문가 시스템