



**(19) 대한민국특허청(KR)**  
**(12) 공개특허공보(A)**

(11) 공개번호 10-2022-0053871  
(43) 공개일자 2022년05월02일

(51) 국제특허분류(Int. Cl.)  
**G11C 29/00** (2006.01)

(52) CPC특허분류  
**G11C 29/838** (2013.01)  
**G11C 29/72** (2013.01)

(21) 출원번호 10-2020-0138122  
(22) 출원일자 2020년10월23일  
심사청구일자 2020년10월23일

(71) 출원인  
**연세대학교 산학협력단**  
서울특별시 서대문구 연세로 50 (신촌동, 연세대학교)

(72) 발명자  
**강성호**  
서울특별시 마포구 양화로 45, 101동 2102호 (서교동, 메세나폴리스)

**김대현**  
서울특별시 서대문구 가재울로2길 31-4, 2층 (남가좌동)

(74) 대리인  
**특허법인(유한)아이시스**

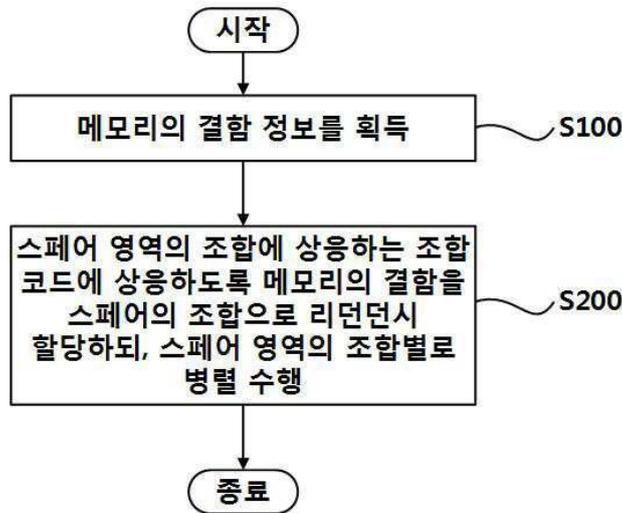
전체 청구항 수 : 총 16 항

(54) 발명의 명칭 **리던던시 분석 방법 및 리던던시 분석 장치**

**(57) 요약**

본 실시예에 의한 메모리의 결함 부분을 적어도 하나의 스페어(spare)로 대체하기 위한 리던던시 분석 방법은: 메모리의 결함 정보를 획득하는 단계 및 스페어의 조합에 상응하는 조합 코드에 상응하도록 결함을 스페어의 조합으로 리던던시 할당하는 단계를 포함하며, 스페어 영역의 조합으로 리던던시 할당하는 단계는, 스페어의 조합별로 병렬 수행(parallel processing)한다.

**대표도** - 도1



(52) CPC특허분류

*G11C 29/76* (2013.01)

*G11C 29/808* (2013.01)

이 발명을 지원한 국가연구개발사업

과제고유번호	1711110267
과제번호	2019R1A2C3011079
부처명	과학기술정보통신부
과제관리(전문)기관명	한국연구재단
연구사업명	중견연구자지원사업
연구과제명	인-메모리 컴퓨팅의 로버스트니스 향상을 위한 반도체 설계 기술
기여율	1/1
과제수행기관명	연세대학교
연구기간	2020.03.01 ~ 2021.02.28

---

## 명세서

### 청구범위

#### 청구항 1

메모리의 결함 부분을 적어도 하나의 스페어(spare)로 대체하기 위한 리던던시 분석 방법으로, 상기 분석 방법은:

상기 메모리의 결함 정보를 획득하는 단계 및

상기 스페어의 조합에 상응하는 조합 코드에 상응하도록 상기 결함을 상기 스페어의 조합으로 리던던시 할당하는 단계를 포함하며,

상기 스페어 영역의 조합으로 리던던시 할당하는 단계는, 상기 스페어의 조합별로 병렬 수행(parallel processing)하는 분석 방법.

#### 청구항 2

제1항에 있어서,

상기 분석 방법은,

복수의 스레드(thread)를 수행하는 병렬 컴퓨팅 장치로 수행되며,

상기 스페어 영역의 조합별로 각 스레드를 수행하여 리던던시 할당을 수행하는 분석 방법.

#### 청구항 3

제1항에 있어서,

상기 메모리는 복수의 로우(row)와 복수의 컬럼(column)을 포함하는 어레이(array) 형태로,

상기 스페어 영역은 상기 메모리의 로우를 대체하는 로우 스페어(row spare)와,

상기 메모리의 컬럼을 대체하는 컬럼 스페어(column spare)를 포함하는 분석 방법.

#### 청구항 4

제3항에 있어서,

상기 조합 코드는,

상기 리던던시 할당 수행 순서대로 상기 로우 스페어 영역에 상응하는 코드와, 상기 컬럼 스페어 영역에 상응하는 코드를 배치한 코드인 분석 방법.

#### 청구항 5

제1항에 있어서,

상기 결함 정보를 획득하는 단계 및

상기 결함을 상기 스페어 영역의 조합으로 리던던시 할당 하는 단계는 복수회 수행되는 분석 방법.

#### 청구항 6

제5항에 있어서,

상기 분석 방법은,

상기 분석 방법이 복수회 수행됨에 따라 리던던시 할당에 사용되는 상기 스페어의 개수 및 사용 가능한 스페어의 개수 중 어느 하나 이상을 계수(count)하는 분석 방법.

**청구항 7**

제4항에 있어서,  
 상기 분석 방법은,  
 상기 스페어의 모든 조합에서 계수된 상기 스페어 영역의 개수가 미리 정해진 값을 초과하거나,  
 사용 가능한 스페어의 개수가 미리 정해진 값 미만인 경우에는 상기 분석 방법을 종료하는 분석 방법.

**청구항 8**

제7항에 있어서,  
 상기 분석 방법은,  
 계수된 상기 스페어 영역의 개수가 적은 조합을 리던던시 할당 방법으로 출력하는 분석 방법.

**청구항 9**

메모리의 결함 부분을 적어도 하나의 스페어로 대체하기 위한 리던던시 분석 장치로, 상기 분석 장치는:  
 자동 테스트 장치(ATE, automatic test equipment)와 통신하여 결함 정보를 수신하는 통신부 및  
 복수의 스레드를 수행하는 복수의 코어들을 포함하는 병렬 컴퓨팅부를 포함하며,  
 상기 병렬 컴퓨팅부는 상기 스페어의 조합에 상응하는 조합 코드에 상응하도록 상기 메모리의 상기 결함을 상기 스페어의 조합으로 리던던시 할당하되, 상기 스페어의 조합별로 상기 스레드를 수행하여 리던던시 할당을 수행하는 분석 장치.

**청구항 10**

제9항에 있어서,  
 상기 결함 정보는  
 상기 메모리의 상기 결함 주소를 포함하는 분석 장치.

**청구항 11**

제9항에 있어서,  
 상기 메모리는 복수의 로우(row)와 복수의 컬럼(column)을 포함하는 어레이(array) 형태로,  
 상기 스페어은 상기 메모리의 로우를 대체하는 로우 스페어(row spare)와,  
 상기 메모리의 컬럼을 대체하는 컬럼 스페어(column spare)를 포함하는 분석 장치.

**청구항 12**

제11항에 있어서,  
 상기 조합 코드는,  
 상기 리던던시 할당 수행 순서대로 상기 로우 스페어에 상응하는 코드와, 상기 컬럼 스페어에 상응하는 코드를 배치한 코드인 분석 장치.

**청구항 13**

제9항에 있어서,  
 상기 분석 장치는  
 상기 자동 테스트 장치로부터 결함 정보를 복수회 수신하고,  
 상기 결함 정보를 수신할 때마다 상기 결함을 상기 스페어의 조합으로 리던던시 할당을 복수회 수행하는 분석

장치.

**청구항 14**

제13항에 있어서,

상기 분석 장치는

상기 결함을 상기 스페어의 조합으로 리턴던시 할당할 때마다 리턴던시 할당에 사용되는 상기 스페어의 개수 및 사용 가능한 스페어의 개수를 계수(count)하는 분석 장치.

**청구항 15**

제14항에 있어서,

상기 분석 장치는,

계수된 상기 스페어의 개수가 미리 정해진 값을 초과하는 경우 및

계수된 상기 사용 가능한 상기 스페어의 개수가 미리 정해진 값 미만인 경우 중 어느 하나 이상일 때에는 상기 분석 방법을 종료하도록 상기 자동 테스트 장치와 통신하는 분석 장치.

**청구항 16**

제14항에 있어서,

상기 분석 장치는

계수된 상기 스페어의 개수가 적은 조합을 상기 자동 테스트 장치에 출력하는 분석 방법.

**발명의 설명**

**기술 분야**

[0001] 본 기술은 리턴던시 분석 방법 및 리턴던시 분석 장치와 관련된다.

**배경 기술**

[0002] 메모리 밀도와 용량이 향상되면서 메모리 오류 발생 가능성이 높아졌다. 이 문제를 해결하기 위해 메모리 내의 리턴던시 메모리 셀을 이용하여 결함이 있는 셀을 대체하는 방법인 리턴던시 분석(redundancy analysis)이 널리 사용된다.

[0003] 다양한 리턴던시 분석 방식이 제안되었지만 여전히 긴 리턴던시 분석 대기 시간 또는 낮은 수리 속도와 같은 단점이 있다. 기존의 리턴던시 분석에서는 하나의 상품 메모리에 대해 리턴던시 분석을 수행하는 데 필요한 시간이 1 초 미만이나 수백만 개 이상의 모든 생성된 메모리를 테스트하고 수리해야하므로 단일 메모리에 대해 1 초 미만인 리턴던시 분석 지연 시간도 전체 메모리 생산 시간에 영향을 미친다. 따라서, 최적의 복구 속도를 얻지 못하면 복구 가능한 메모리가 폐기되어 메모리 생산 수율이 감소할 수 있다.

[0004] 한편 근래의 컴퓨터는 병렬 작업(thread)을 수행할 수 있는 복수의 코어들을 포함하며 높은 병렬 컴퓨팅 성능을 제공한다. 이러한 병렬 컴퓨팅은 다양한 분야에서 응용 프로그램 및 시뮬레이션의 성능을 향상시키는 데 널리 사용된다. 이러한 병렬 작업(thread)을 수행할 수 있는 복수의 코어들을 포함하는 장치로는 그래픽 프로세서(GPU, graphic processor unit)가 일반적이다.

**발명의 내용**

**해결하려는 과제**

[0005] 종래 기술에 의한 리턴던시 분석은 장시간이 소요되며, 오류 비트 맵을 저장할 수 있는 큰 용량의 저장 공간이 필요하다. 한 번에 테스트되는 메모리 수가 많을수록 메모리 오류 발생 확률이 높을수록 오류 정보를 저장하는 데 필요한 저장 공간은 더욱 커진다. 따라서 리턴던시 분석에 소모되는 시간을 줄이고 최적의 수리율을 달성하기 위한 효율적인 리턴던시 분석 방법이 필요하다.

[0006] 본 기술로 해결하고자 하는 과제 중 하나는 상술한 종래 기술의 난점을 해소하기 위한 것이다. 본 기술로 해결하고자 하는 과제 중 하나는 리턴던시 분석에 소모되는 시간을 줄이고 최적의 수리율을 달성할 수 있는 리턴던시 분석 방법 및 리턴던시 분석 장치를 제공하는 것이다.

**과제의 해결 수단**

[0007] 본 실시예에 의한 메모리의 결합 부분을 적어도 하나의 스페어(spare)로 대체하기 위한 리턴던시 분석 방법은: 메모리의 결합 정보를 획득하는 단계 및 스페어의 조합에 상응하는 조합 코드에 상응하도록 결합을 스페어의 조합으로 리턴던시 할당하는 단계를 포함하며, 스페어 영역의 조합으로 리턴던시 할당하는 단계는, 스페어의 조합별로 병렬 수행(parallel processing)한다.

[0008] 일 실시예로, 분석 방법은, 복수의 스레드(thread)를 수행하는 병렬 컴퓨팅 장치로 수행되며, 스페어 영역의 조합별로 각 스레드를 수행하여 리턴던시 할당을 수행한다.

[0009] 일 실시예로, 메모리는 복수의 로우(row)와 복수의 컬럼(column)을 포함하는 어레이(array) 형태로, 스페어 영역은 메모리의 로우를 대체하는 로우 스페어(row spare)와, 메모리의 컬럼을 대체하는 컬럼 스페어(column spare)를 포함한다.

[0010] 일 실시예로, 조합 코드는, 리턴던시 할당 수행 순서대로 로우 스페어 영역에 상응하는 코드와, 컬럼 스페어 영역에 상응하는 코드를 배치한 코드이다.

[0011] 일 실시예로, 결합 정보를 획득하는 단계 및 결합을 스페어 영역의 조합으로 리턴던시 할당 하는 단계는 복수회 수행된다.

[0012] 일 실시예로, 분석 방법은, 분석 방법이 복수회 수행됨에 따라 리턴던시 할당에 사용되는 스페어의 개수 및 사용 가능한 스페어의 개수 중 어느 하나 이상을 계수(count)한다.

[0013] 일 실시예로, 분석 방법은, 스페어의 모든 조합에서 계수된 스페어 영역의 개수가 미리 정해진 값을 초과하거나, 사용 가능한 스페어의 개수가 미리 정해진 값 미만인 경우에는 분석 방법을 종료한다.

[0014] 일 실시예로, 분석 방법은, 계수된 스페어 영역의 개수가 적은 조합을 리턴던시 할당 방법으로 출력한다.

[0015] 본 실시예에 의한 메모리의 결합 부분을 적어도 하나의 스페어로 대체하기 위한 리턴던시 분석 장치는: 자동 테스트 장치(ATE, automatic test equipment)와 통신하여 결합 정보를 수신하는 통신부 및 복수의 스레드를 수행하는 복수의 코어들을 포함하는 병렬 컴퓨팅부를 포함하며, 병렬 컴퓨팅부는 스페어의 조합에 상응하는 조합 코드에 상응하도록 메모리의 결합을 스페어의 조합으로 리턴던시 할당하되, 스페어의 조합별로 스레드를 수행하여 리턴던시 할당을 수행한다.

[0016] 일 실시예로, 결합 정보는 메모리의 결합 주소를 포함한다.

[0017] 일 실시예로, 메모리는 복수의 로우(row)와 복수의 컬럼(column)을 포함하는 어레이(array) 형태로, 스페어은 메모리의 로우를 대체하는 로우 스페어(row spare)와, 메모리의 컬럼을 대체하는 컬럼 스페어(column spare)를 포함한다.

[0018] 일 실시예로, 조합 코드는, 리턴던시 할당 수행 순서대로 로우 스페어에 상응하는 코드와, 컬럼 스페어에 상응하는 코드를 배치한 코드이다.

[0019] 일 실시예로, 분석 장치는 자동 테스트 장치로부터 결합 정보를 복수회 수신하고, 결합 정보를 수신할 때마다 결합을 스페어의 조합으로 리턴던시 할당을 복수회 수행한다.

[0020] 일 실시예로, 분석 장치는 결합을 스페어의 조합으로 리턴던시 할당할 때마다 리턴던시 할당에 사용되는 스페어의 개수 및 사용 가능한 스페어의 개수를 계수(count)한다.

[0021] 일 실시예로, 분석 장치는, 계수된 스페어의 개수가 미리 정해진 값을 초과하는 경우 및 계수된 사용 가능한 스페어의 개수가 미리 정해진 값 미만인 경우 중 어느 하나 이상일 때에는 분석 방법을 종료하도록 자동 테스트 장치와 통신한다.

[0022] 일 실시예로, 분석 장치는 계수된 스페어의 개수가 적은 조합을 자동 테스트 장치에 출력한다.

**발명의 효과**

[0023] 본 실시예에 의하면 리던던시 분석의 대기 시간을 감소시킬 수 있으며, 높은 수리율을 얻을 수 있다는 장점이 제공된다.

**도면의 간단한 설명**

- [0024] 도 1은 본 실시예에 따른 리던던시 분석(redundancy analysis) 방법의 개요를 도시한 순서도이다.
- 도 2는 본 실시예에 따른 리던던시 분석(redundancy analysis) 장치의 개요를 도시한 블록도이다.
- 도 3은 자동 테스트 장치와 리던던시 분석 장치 사이에서 메모리 테스트 및 리던던시 분석(RA)이 이루어지는 과정을 개요적으로 도시한 도면이다.
- 도 4는 자동 검사 장치(ATE)와 리던던시 분석 장치의 동작을 개요적으로 도시한 도면이다.
- 도 5는 결함 정보로부터 병렬 컴퓨팅 부가 리던던시 분석을 수행하는 과정을 개요적으로 도시한 도면이다.
- 도 6은 예비 부품 수에 따른 RA의 분석 시간과 정규화 된 수리율을 도시한 도면이다.
- 도 7은 정규화된 수리율의 분석 시간을 도시한 도면이다.

**발명을 실시하기 위한 구체적인 내용**

- [0025] 이하에서는 첨부된 도면들을 참조하여 본 실시예에 의한 리던던시 분석(redundancy analysis) 방법 및 리던던시 분석 장치를 설명한다. 도 1은 본 실시예에 따른 리던던시 분석(redundancy analysis) 방법의 개요를 도시한 순서도이고, 도 2는 본 실시예에 따른 리던던시 분석(redundancy analysis) 장치의 개요를 도시한 블록도이다. 도 1 및 도 2를 참조하면, 본 실시예에 의한 리던던시 분석 방법은 메모리의 결함 정보를 획득하는 단계(S100) 및 스페어(spare)의 조합에 상응하는 조합 코드에 상응하도록 메모리의 결함을 스페어의 조합으로 리던던시 할당(redundancy allocation)하는 단계(S200)를 포함하며, 스페어의 조합으로 리던던시 할당하는 단계는, 스페어의 조합별로 병렬 수행(parallel processing)된다.
- [0026] 본 실시예에 의한 리던던시 분석 장치(10)는 메모리를 테스트하는 자동 테스트 장치(automatic test equipment, ATE)와 통신하여 결함 정보를 수신하는 통신부(100) 및 복수의 스레드를 수행하는 복수의 코어들(210)을 포함하는 병렬 컴퓨팅부(200), 병렬 컴퓨팅부(200)는 스페어 영역의 조합에 상응하는 조합 코드에 상응하도록 메모리의 결함을 스페어 영역의 조합으로 리던던시 할당하되, 스페어 영역의 조합별로 각 스레드를 수행하여 리던던시 할당을 수행한다.
- [0027] 병렬 컴퓨팅부(200)는 복수의 코어(210)들을 포함하며, 이들을 이용하여 복수의 연산 스레드(thread)를 병렬로 처리할 수 있다. 자동 테스트 장치(ATE)는 많은 수의 메모리를 동시에 테스트하므로 각 대상 메모리에서 얻은 많은 양의 결함 정보를 처리할 필요가 있다. 본 실시예와 같이 병렬 연산을 수행할 수 있는 병렬 컴퓨팅부(200)를 사용하여 효율적인 리던던시 분석을 수행할 수 있다. 일 실시예로, 병렬 컴퓨팅부(200)는 복수의 코어(210)를 포함하는 그래픽 프로세싱 유닛(GPU, graphic processing unit)일 수 있다. 다른 실시예로, 병렬 컴퓨팅부(200)는 복수의 중앙 처리 장치(CPU)일 수 있다.
- [0028] 통신부(100)는 메모리를 검사하는 자동 검사 장치(ATE)와 통신하여 리던던시 분석에 필요한 데이터를 송수신한다. 자동 검사 장치(ATE)는 검사 대상 메모리에 결함이 발견되면 결함 정보를 리던던시 분석 장치(10)로 전송한다. 리던던시 분석 장치(10)는 병렬 컴퓨팅부(200)가 아이들 상태일 때 리던던시 분석 결과를 자동 검사 장치(ATE)로 전송한다.
- [0029] 일 실시예로, 자동 검사 장치(ATE)와 통신부(100) 사이에서 통신되는 데이터는 결함 정보(fault information), 리던던시 분석 상태 및 리던던시 분석 결과를 포함한다. 결함 정보는 검사 대상 메모리 중 결함이 있는 메모리의 수, 해당 메모리에 있는 결함 행 및 열 주소를 포함한다. 리던던시 분석 상태는 리던던시 할당으로 결함을 수리하는 방법, 남은 스페어 수 및 수리된 행 또는 열의 주소를 포함한다. 리던던시 결과 데이터는 수리후 남은 스페어의 총 수와 수리 상태를 포함할 수 있다. 일 실시예로, 통신부(100)는 자동 검사 장치(ATE)와 송수신하는 데이터를 일시적으로 저장하는 버퍼(미도시)를 더 포함할 수 있다.
- [0030] 병렬 컴퓨팅부(200)는 병렬 컴퓨팅을 통해 많은 계산을 수행할 수 있으므로 결함 정보를 저장하지 않고 새로운 결함이 감지될 때마다 리던던시 할당(redundancy allocation)의 모든 사례를 동시에 검사할 수 있고, 이를 통하여 결함을 동적으로 분석할 수 있다.

- [0031] 종래 기술에서 필수적인 정적 또는 하이브리드 리던던시 분석에 필수적인 결함 비트 맵은 본 실시예에서 필요하지 않다. 또한 결함이 감지되면 리던던시 분석을 수행하므로 테스트 시퀀스가 완료된 후 추가 분석 시간이 필요하지 않다. 나아가, 본 실시예에 의하면 가능한 모든 중복할당 사례를 검사할 수 있으므로 종래 기술에 비하여 복구 속도가 향상된다.
- [0032] 도 3은 자동 테스트 장치(ATE)와 리던던시 분석 장치(10) 사이에서 메모리 테스트 및 리던던시 분석(RA)이 이루어지는 과정을 개요적으로 도시한 도면이다. 도 3을 참조하면, 자동 테스트 장치(ATE)에 검사 대상 메모리를 장착하여 메모리 테스트를 시작한다. 자동 테스트 장치(ATE)는 검사 대상 메모리를 테스트하여 결함을 검출하면, 결함 정보를 형성하여 리던던시 분석 장치(10)에 제공한다.
- [0033] 리던던시 분석 장치(10)는 결함 정보를 수신하면 리던던시 분석을 수행한다. 리던던시 분석된 결과는 버퍼(미도시)에 저장될 수 있으며, 리던던시 분석 장치(10)가 아이들 상태에 있을 때 리던던시 분석의 중간 결과를 자동 테스트 장치(ATE)로 출력한다. 이어서, 자동 테스트 장치(ATE)가 새로운 결함을 검출하여 결함 정보를 제공하면 리던던시 분석 장치(10)는 새로운 결함을 포함하여 리던던시 분석을 수행하여 중간 결과를 자동 테스트 장치(ATE)에 전송한다.
- [0034] 메모리 테스트 종료 후, 자동 테스트 장치(ATE)는 입력된 리던던시 분석 결과로부터 메모리 결함을 수리할 수 있는 최적의 방법을 선택하여 메모리의 결함을 수리한다. 일 예로, 자동 테스트 장치(ATE)는 가장 적은 수의 스레드 수를 가지는 수리 방법을 선택하여 메모리 결함을 수리할 수 있다. 도시되지 않은 다른 실시예에 의하면, 주어진 개수의 행 스레드와 열 스레드들로 메모리 테스트의 결과에 의하여 확인한 결함들을 수리할 수 없는 경우에는 리던던시 분석을 진행하지 않고 조기 종료할 수 있다.
- [0035] 종래 기술은 메모리 테스트가 완료되면 자동 테스트 장치(ATE)가 결함 정보를 형성하여 리던던시 분석 장치(10)에 제공하여 리던던시 분석이 수행되었다. 이러한 경우에는 메모리 테스트 이후에 비로소 리던던시 분석이 수행되므로 메모리 테스트 및 리던던시 분석 완료까지 소요되는 시간이 길어진다. 또한, 메모리 테스트 결과인 결함 비트맵을 저장하기 위하여 큰 용량의 저장 장치가 필요하다.
- [0036] 그러나, 본 실시예에 의하면 메모리 테스트가 수행되는 동안에 리던던시 분석이 수행되어 메모리 테스트 시작부터 리던던시 분석 완료까지의 소요 시간을 줄일 수 있다는 장점이 제공되며, 모든 결함 정보를 저장할 필요가 없어 저장 장치의 용량을 감소시킬 수 있다는 장점도 제공된다.
- [0037] 도 4는 자동 검사 장치(ATE)와 리던던시 분석 장치(10)의 동작을 개요적으로 도시한 도면이다. 도 4를 참조하면, 자동 검사 장치(ATE)는 하나 이상의 검사 대상 메모리들을 검사할 수 있다. 도 4는 검사 대상 메모리 #0와 검사 대상 메모리 #1의 두 개의 메모리를 검사하는 예를 도시한다. 도 4로 도시된 예에서, 검사대상 메모리는 각각 두 개의 행 스레드와 두 개의 열 스레드를 가지는 것을 예시한다.
- [0038] 테스트가 시작되면 자동 테스트 장치(ATE)는 메모리 주소와 데이터로 구성된 테스트 패턴을 검사 대상 메모리로 전송하고, 검사 대상 메모리로부터 데이터를 수신한다. 검사 대상 메모리로부터 수신된 데이터는 자동 테스트 장치(ATE)가 검사 대상 메모리로 전송한 데이터와 비교되고, 자동 테스트 장치(ATE)는 결함을 감지한다.
- [0039] 자동 테스트 장치(ATE)가 검사 대상 메모리의 응답을 검사하는 과정에서 검사 대상 메모리 # 1에서 결함을 감지하면, 결함 위치의 행(row) 및 컬럼(column)을 나타내는 결함 주소 정보를 생성한다. 자동 테스트 장치(ATE)는 검사 대상 메모리들 중 결함이 있는 검사 대상 메모리와, 결함 개수를 나타내는 결함 개수 정보를 생성하고, 이들을 포함하는 결함 정보를 리던던시 분석 장치(10)로 전송한다.
- [0040] 도 4로 예시된 예에서, 자동 검사 장치(ATE)는 검사 대상 메모리 #0에서 결함을 발견하지 못하였으나, 검사 대상 메모리 #1에서 하나의 결함을 발견한 경우를 예시한다. 자동 검사 장치(ATE)는 메모리의 결함 개수 정보로 검사 대상 메모리 #0에는 0, 검사 대상 메모리 #1에는 1의 값을 설정한다. 또한, 자동 검사 장치(ATE)는 검사 대상 메모리 #1에서 검출된 결함의 주소인 Row 3, Col 6을 설정하여 리던던시 분석 장치(10)에 전송한다.
- [0041] 각 검사 대상 메모리에 대한 리던던시 분석은 자동 테스트 장치(ATE)가 제공한 결함 정보를 기초로 수행된다. 도 4로 도시된 예에서 검사 대상 메모리 하나에 대한 리던던시 분석은 한 블록에서 수행된다. 각 블록에는 복수개의 스레드(thread)가 할당될 수 있으며, 할당되는 스레드 수는 하나의 스레드에서 하나의 수리 케이스를 검사하므로 가능한 수리 케이스의 수와 같다.
- [0042] 리던던시 분석 장치(10)가 구동되어 초기화되면 리던던시 분석 상태를 저장할 공간이 미리 전역 메모리(미도시)에 할당된다. 리던던시 분석 장치(10)가 초기화되면 복구된(Repaired) 행 주소(Row)와 열 주소(Col)은 각각

-1, -1로 초기화된다. 검사 대상 메모리 #0에서 결함이 발견되지 않았으므로 블록 # 0에서는 리턴던시 분석이 수행되지 않는다. 블록 # 1에서 리턴던시 분석은 할당된 모든 스레드에서 실행되며, 리턴던시 분석 상태가 저장된다.

[0043] 리턴던시 분석 상태를 참조하면, 블록 #1에서 수행되는 스레드 # 0에서 결함은 행 스페어로 복구된다. 따라서 남아있는(Remain) 행 스페어 수(Row)가 1로 감소하고 복구된(Repaired) 행 주소(Row) 4를 포함하는 열 주소 4, -1이 저장된다. 마찬가지로, 스레드 # 5에서 결함은 열 스페어로 복구된다. 따라서 남아있는(Remain) 열 스페어 수(Col)가 1로 감소하고 복구된(Repaired) 열 주소(Col) 6을 포함하는 주소 -1, 6이 저장된다.

[0044] 리턴던시 분석 결과를 참조하면, 도시된 예에서 검사 대상 메모리 #0에는 결함이 발견되지 않아 블록 # 0에서 리턴던시 분석이 수행되지 않았다. 따라서, 스페어가 사용되지 않고, 블록 #0에서 수행된 리턴던시 분석 결과로 남은 스페어 수로 4가 저장된다. 그러나, 블록 # 1에서 수행되는 리턴던시 분석결과에 의하면 모든 스레드에서 하나의 스페어가 사용되므로, 블록 # 1에서 수행된 리턴던시 분석 결과로 남은 스페어 수로 3이 저장된다. 수리 성공에 1을 저장하여 모든 스레드가 결함 복구에 복구 성공을 나타낸다. 리턴던시 분석이 완료되어 리턴던시 분석 장치가 아이들 상태일 때 리턴던시 분석 결과를 자동 테스트 장치(ATE)로 전송한다.

[0045] 도 5는 결함 정보로부터 병렬 컴퓨팅 부(200)가 리턴던시 분석을 수행하는 과정을 개요적으로 도시한 도면이다. 도 5로 예시된 실시예에서, 리턴던시 분석은 스레드 #0(Thread #0)에서 스레드 #5(Thread #5)까지 총 6개의 스레드로 병렬적으로 수행된다. 다만, 리턴던시 분석을 수행하는 스레드의 개수는 간결하고 명확한 설명을 위한 것으로, 본 발명의 범위를 제한하는 것이 아니다.

[0046] 각 스레드 별로 스페어 영역의 조합에 상응하는 조합 코드가 설정된다. 일 실시예로, 검사 대상 메모리에는 네 개의 행 스페어와 네 개의 열 스페어가 형성된 것을 예시하며, 각 스레드 별로 할당된 조합 코드는 각 스페어의 할당 순서를 나타낸다. 도시된 실시예에서, 스레드 #0(Thread #0)는 RRRR ... C의 조합 코드가 부여되며, 이것은 메모리에서 최초로 발견된 결함 내지 네 번째로 발견된 결함은 행 스페어를 할당하며, 최후로 발견된 결함은 열 스페어로 할당하는 것을 나타낸다. 마찬가지로, 스레드 #3(Thread #3)는 CRRR...C의 조합코드가 부여되며, 이것은 메모리에서 최초로 발견된 결함은 열 스페어를 할당하고, 두 번째 내지 네 번째로 발견된 결함은 행 스페어로 할당하는 것을 나타낸다. 이와 같이 네 개의 행 스페어와 네 개의 열 스페어를 이용하여 리턴던시 할당을 수행할 수 있는 모든 경우 각각에 조합 코드가 부여되고, 이들 각각은 별개의 스레드에 의하여 리턴던시 분석이 수행된다.

[0047] 도 5로 예시된 예에서, 테스트 사이클 #1에서 자동 테스트 장치(ATE)로부터 메모리의 결함 정보와 조합 코드에 따라 리턴던시 할당이 수행된다. 예시를 위하여 메모리의 결함 위치는 메모리 어레이 내에 1로 표시되었다. 스레드 #0(Thread #0)의 조합 코드는 RRRR...C로, 최초 결함은 행 스페어로 할당된다. 스레드 #3(Thread #3)의 조합코드는 CRRR...C이므로, 최초 결함은 열 스페어로 할당된다. 결함을 스페어로 할당하는 리턴던시 할당 과정은 각 스레드들에서 병렬적으로 수행된다. 테스트 사이클 #1에서 발견된 결함에 대하여 할당된 스페어 영역의 개수(Repaired)는 모두 1로 설정된다.

[0048] 테스트 사이클 #2에서 검사 대상 메모리에 대하여 후속 검사가 수행되어 1과 2의 결함이 검색되었다. 자동 테스트 장치(ATE)는 결함 정보로 이들 결함을 리턴던시 분석 장치(10)에 제공한다. 리턴던시 분석 장치(10)에서 수행되는 각 스레드들은 제공된 결함 정보에 상응하는 결함에 대하여 설정된 조합코드로 리턴던시 할당을 수행한다. 테스트 사이클 #2에서 스레드 #1은 첫 번째와 두 번째 결함에 대하여 행 스페어들로 리턴던시 할당을 수행하며, 스레드 #3은 첫 번째와 두 번째 결함에 대하여 행 스페어와 열 스페어로 리턴던시 할당을 수행한다. 테스트 사이클 #2에서 리턴던시 할당을 수행하는데 필요한 행 스페어와 열 스페어의 개수는 모두 두 개로, 이를 필요한 스페어의 개수(Repaired)로 설정한다.

[0049] 테스트 사이클 #3에서 자동 테스트 장치(ATE)는 검사 대상 메모리에 대한 검사를 수행하였으나, 결함을 검출하지 못하였다. 이러한 경우에는 각 스레드는 동작하지 않을 수 있으며, 테스트 사이클 #1, #2에서 리턴던시 분석 결과(도 3 참조)를 자동 테스트 장치(ATE)로 전송할 수 있다.

[0050] 테스트 사이클 #4에서 자동 테스트 장치(ATE)에 의한 검사가 수행되어 결함 1, 2 및 3을 검출하고, 결함 정보를 리턴던시 분석 장치(10)에 제공한다. 리턴던시 분석 장치(10)에서 수행되는 각 스레드들은 제공된 결함 정보와, 설정된 조합 코드에 대하여 리턴던시 할당을 수행한다. 스레드 #1은 결함 1, 2, 3에 대하여 모두 행 스페어로 리턴던시 할당을 수행하여 총 세 개의 스페어 영역을 이용하여 리턴던시 할당을 수행한다. 그러나, 스레드 #2는 결함 1에 대하여는 행 스페어로 리턴던시 할당을 수행하나, 결함 2 및 3에 대하여는 하나의 열 스페어로 리턴던

시 할당을 수행할 수 있다. 따라서, 스투드 #2에 의하여 결함 1, 2 및 3에 대한 리턴던시 할당시 필요한 스페어 영역의 개수(Repaired)는 2로 설정된다.

[0051] 메모리 테스트 과정이 진행됨에 따라 검사 대상 메모리에서 발견된 결함에 대하여 행 스페어와 열 스페어들을 할당하여 리턴던시 할당을 수행하며, 리턴던시 할당시 사용되는 스페어들의 수를 갱신한다. 최후의 메모리 테스트 과정인 테스트 싸이클 #n 에서는 모든 결함을 해소하기 위한 리턴던시 할당시 필요한 스페어 영역의 개수(Repaired)와 가용한 스페어 영역의 개수를 비교하고, 각 스투드별로 메모리 수리 가능과 수리 불가능을 판단한다. 도시된 예에서 스투드 #2은 테스트 과정에서 발견된 결함을 수리할 수 있으며, 이 때, 메모리의 결함들은 RCRR...C의 스페어들로 리턴던시 할당되어 수리될 수 있다.

[0052] **실험예**

[0053] 본 실시예에 의한 리턴던시 분석 방법 및 리턴던시 분석 장치는 메모리 테스트와 함께 수행되어야한다. 메모리 테스트를 위하여 실제 메모리의 고장 분포와 유사한 Poly-Eggenberger 분포 모델을 사용하였다. 본 실시예의 성능을 측정하기 위하여 본 실시예 및 종래의 리턴던시 분석 방법에 대한 시뮬레이터와 1024 \* 1024 메모리를 모든 실험에 공통적으로 하였다. 결함 분포당 각 실험은 1,000 회 진행되었으며, 공정한 비교를 위해 평균 값을 계산하였다.

[0054] 종래 기술의 리턴던시 분석 방법에 대한 실험은 3.2GHz CPU를 사용하여 수행되었으며, 본 실시예를 구현하기 위하여 NVIDIA pascal 아키텍처 GPU가 사용되었으며, GPU를 프로그래밍을 위하여 NVIDIA CUDA 라이브러리를 사용하였다.

[0055] 본 실시예와 종래 기술을 리턴던시 분석 성능 및 하드웨어 오버 헤드 측면에서 평가하였으며, 본 실시예의 리턴던시 분석 성능을 평가하기 위해 대기 시간 및 정규화된(normalized) 수리 비율을 측정하였으며 종래 기술을 구현하는 데 필요한 하드웨어 오버 헤드와 비용은 종래 기술과 비교하여 추정되었다.

[0056] 다른 다섯 가지 리턴던시 분석 방법과 함께 본 실시예를 스페어 수에 따라 분석 시간과 수리율(repair rate)을 평가하였다. 메모리 크기는 1,024 행과 열로 고정되었고, 오류 수는 정규화 된 수리 비율을 평가하기 위해 최대 스페어 수보다 큰 21 개로 설정하였다.

[0057] 대부분의 경우 테스트와 리턴던시 분석은 동시에 수행되기 때문에 메모리 테트 후 리턴던시 분석 결과를 확인하는 데 필요한 시간만 리턴던시 분석 레이턴시로 간주된다. 그러나 테스트 종료시 오류가 발견되면 마지막 오류에 대한 분석 시간과 장치에서 호스트로 RA 결과를 복사하는 시간을 고려한다. 실험은 1,000 회 반복되며 분석 시간은 최종 테스트 반복 후 리턴던시 분석을 완료하는 데 걸린 평균 시간입니다.

[0058] 도 6은 예비 부품 수에 따른 RA의 분석 시간과 정규화 된 수리율을 도시한 도면이다. 본 실시예는 평가된 방법 중 가장 짧은 리턴던시 분석 대기 시간을 나타냈다. 스페어 수에 따라 수리 사례 수가 증가했기 때문에 본 실시예의 리턴던시 분석 지연 시간은 스페어 수에 비례하여 증가하였다. 스페어 수가 적을 때 종래 기술인 RM과 FAST는 정규화된 수리 율이 감소한 반면, 마찬가지로 종래 기술인 B & B, VERA , FGPM은 100 %의 정규화 된 수리 율을 유지하였다. 본 실시예는 100 %의 정규화 된 수리율을 유지하였다.

[0059] 10 개의 행 스페어와 10 개의 열 스페어가 있는 메모리에 대해 서로 다른 수의 오류가 있는 실험을 수행하였다. 도 7은 정규화된 수리율의 분석 시간을 도시한 도면이다. 종래 기술인 B & B, VERA, FPGM과 같은 리턴던시 분석 기술과 비교하여 본 실시예는 오류 수에 무관하게 가장 높은 리턴던시 분석 속도와 거의 일정한 대기 시간을 나타냈다. 예상대로 고장 수가 증가하더라도 본 실시예는 100.00 %의 정규화된 수리율을 유지하였으나, 종래 기술인 RM과 FAST는 정규화 된 수리율이 감소한 것을 확인할 수 있다.

[0060] 본 실시예는 메모리 테스터에서 결함이 감지되면 리턴던시 분석를 수행하므로 테스트 시퀀스가 완료된 후 추가 분석 시간이 필요하지 않다. 나아가, 본 실시예에 의하면 가능한 모든 중복할당 사례를 검사할 수 있으므로 종래 기술에 비하여 복구 속도가 향상된다. 나아가 도시된 것과 같이 높은 수리율을 제공한다는 장점이 제공된다.

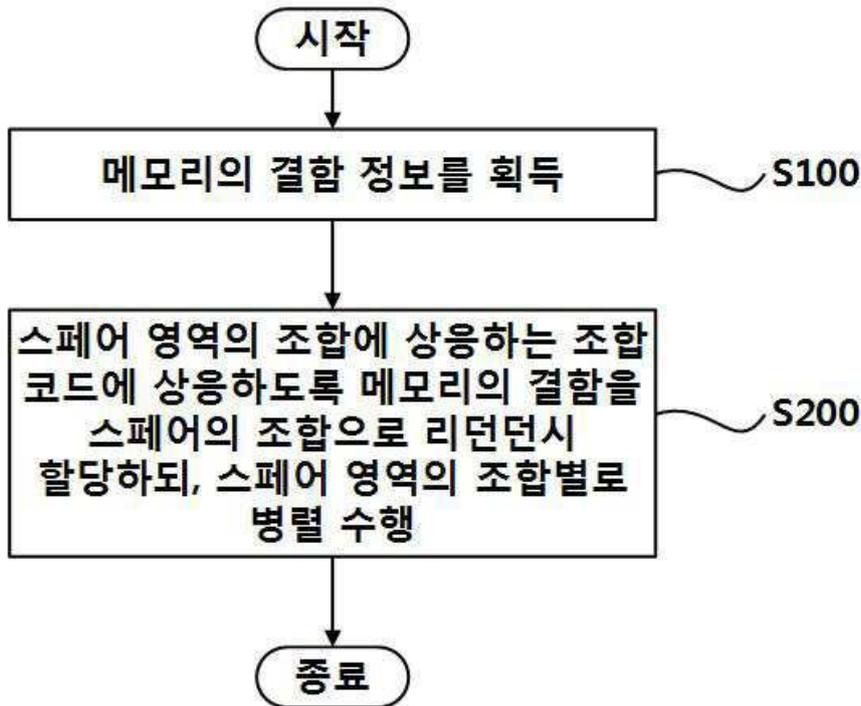
[0061] 본 발명에 대한 이해를 돕기 위하여 도면에 도시된 실시 예를 참고로 설명되었으나, 이는 실시를 위한 실시예로, 예시적인 것에 불과하며, 당해 분야에서 통상적 지식을 가진 자라면 이로부터 다양한 변형 및 균등한 타 실시 예가 가능하다는 점을 이해할 것이다. 따라서, 본 발명의 진정한 기술적 보호범위는 첨부된 특허청구범위에 의해 정해져야 할 것이다.

**부호의 설명**

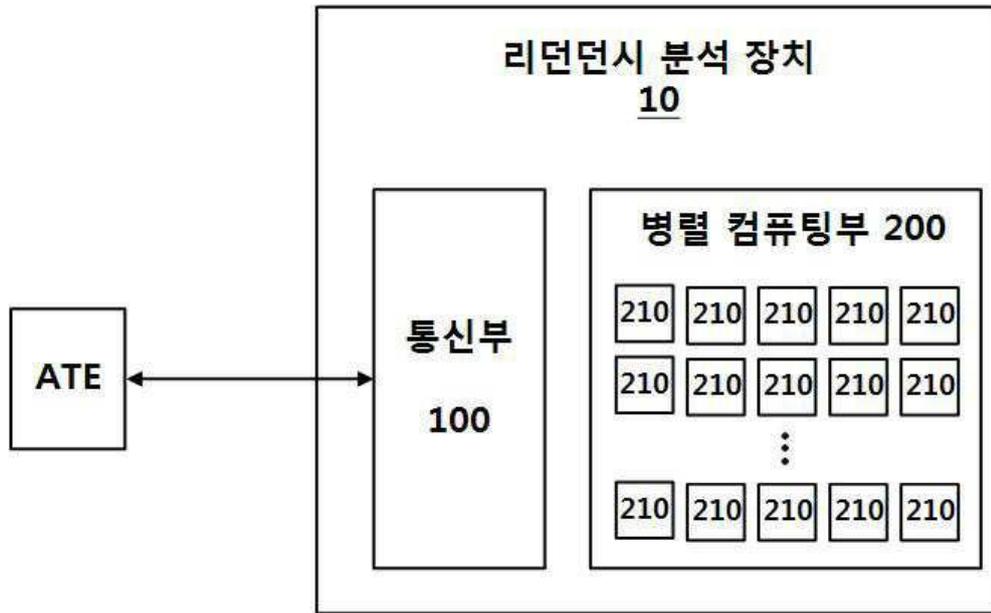
[0062] S100, S200: 본 실시예에 의한 리던던시 분석 방법의 개요적 단계  
10: 리던던시 분석 장치  
100: 통신부  
200: 병렬 컴퓨팅 부  
210: 코어

도면

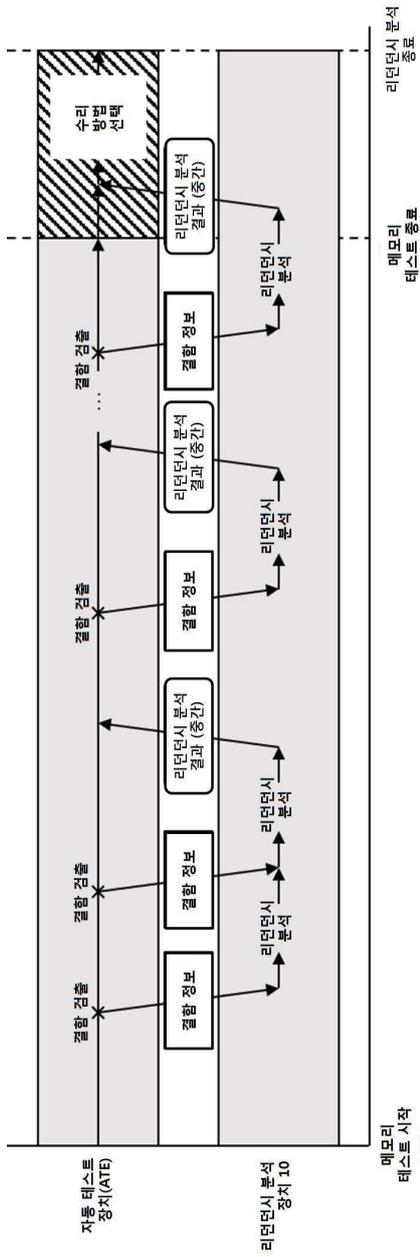
도면1



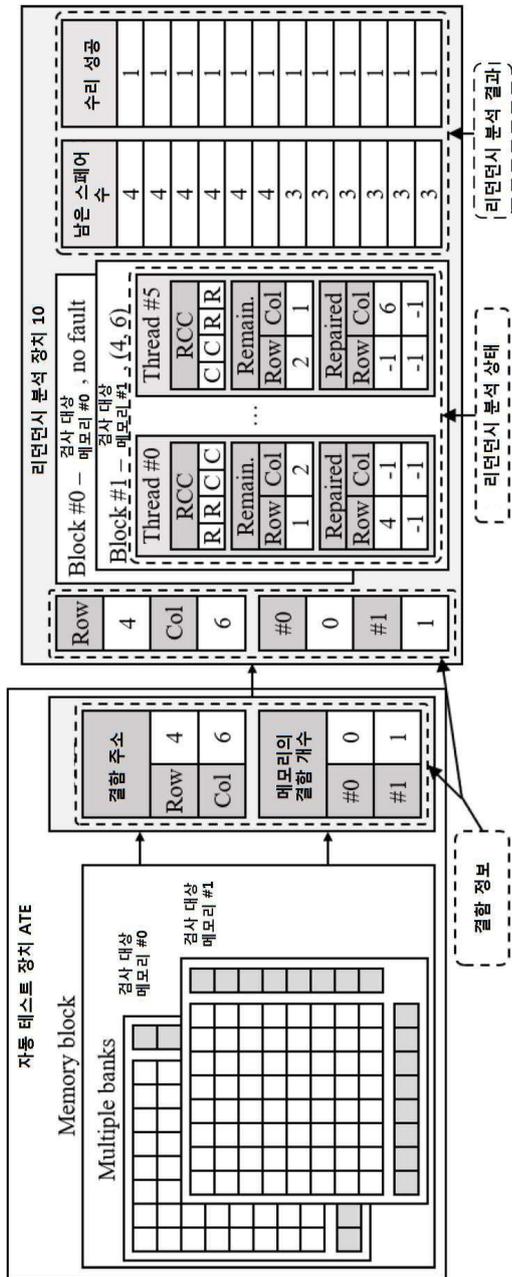
도면2



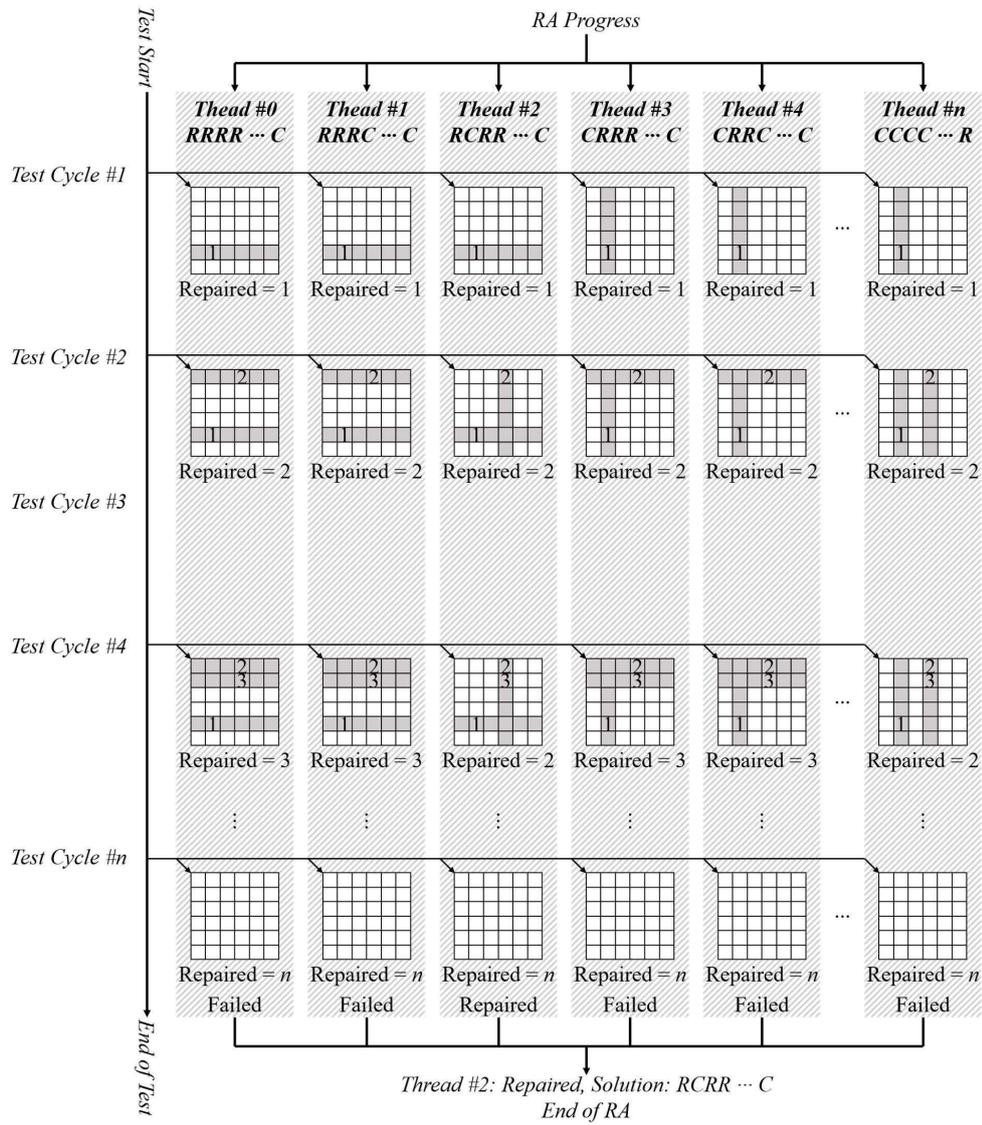
도면3



도면4



도면5



도면6

Number of Spares	RM [5]		B&B [6]		FAST [8]		VERA [9]		FGPM [10]		GRACE	
	RA Latency ( $\mu$ s)	Normalized Repair Rate (%)	RA Latency ( $\mu$ s)	Normalized Repair Rate (%)	RA Latency ( $\mu$ s)	Normalized Repair Rate (%)	RA Latency ( $\mu$ s)	Normalized Repair Rate (%)	RA Latency ( $\mu$ s)	Normalized Repair Rate (%)	RA Latency ( $\mu$ s)	Normalized Repair Rate (%)
R C												
7	$5.98 \times 10^0$	62.96	$2.70 \times 10^5$	100.00	$8.59 \times 10^2$	70.37	$1.89 \times 10^3$	100.00	$8.83 \times 10^2$	100.00	$3.76 \times 10^{-1}$	100.00
7	$6.10 \times 10^0$	83.05	$2.72 \times 10^5$	100.00	$7.31 \times 10^2$	89.83	$4.02 \times 10^3$	100.00	$1.76 \times 10^3$	100.00	$3.85 \times 10^{-1}$	100.00
8	$6.13 \times 10^0$	88.14	$2.82 \times 10^5$	100.00	$7.67 \times 10^2$	91.53	$4.11 \times 10^3$	100.00	$1.74 \times 10^3$	100.00	$3.83 \times 10^{-1}$	100.00
8	$6.29 \times 10^0$	95.40	$2.91 \times 10^5$	100.00	$4.38 \times 10^2$	98.85	$7.90 \times 10^3$	100.00	$2.75 \times 10^3$	100.00	$3.87 \times 10^{-1}$	100.00
8	$6.33 \times 10^0$	98.94	$3.09 \times 10^5$	100.00	$2.78 \times 10^2$	100.00	$1.30 \times 10^4$	100.00	$3.76 \times 10^3$	100.00	$4.00 \times 10^{-1}$	100.00
9	$6.33 \times 10^0$	98.94	$3.03 \times 10^5$	100.00	$2.69 \times 10^2$	100.00	$1.36 \times 10^4$	100.00	$3.99 \times 10^3$	100.00	$3.98 \times 10^{-1}$	100.00
9	$6.46 \times 10^0$	98.99	$3.16 \times 10^5$	100.00	$4.64 \times 10^1$	100.00	$1.85 \times 10^4$	100.00	$5.32 \times 10^3$	100.00	$4.03 \times 10^{-1}$	100.00

도면7

Number of Faults	RM [5]		B&B [6]		FAST [8]		VERA [9]		FGPM [10]		GRACE	
	RA Latency ( $\mu$ s)	Normalized Repair Rate (%)	RA Latency ( $\mu$ s)	Normalized Repair Rate (%)	RA Latency ( $\mu$ s)	Normalized Repair Rate (%)	RA Latency ( $\mu$ s)	Normalized Repair Rate (%)	RA Latency ( $\mu$ s)	Normalized Repair Rate (%)	RA Latency ( $\mu$ s)	Normalized Repair Rate (%)
21	$6.71 \times 10^0$	100.00	$3.56 \times 10^5$	100.00	$9.24 \times 10^0$	100.00	$3.34 \times 10^4$	100.00	$7.14 \times 10^3$	100.00	$4.30 \times 10^{-1}$	100.00
22	$7.10 \times 10^0$	100.00	$5.17 \times 10^5$	100.00	$1.67 \times 10^1$	100.00	$3.95 \times 10^4$	100.00	$7.63 \times 10^3$	100.00	$4.21 \times 10^{-1}$	100.00
23	$7.57 \times 10^0$	99.89	$8.55 \times 10^5$	100.00	$1.74 \times 10^1$	100.00	$4.24 \times 10^4$	100.00	$9.59 \times 10^3$	100.00	$3.93 \times 10^{-1}$	100.00
24	$7.91 \times 10^0$	96.00	$1.54 \times 10^6$	100.00	$1.68 \times 10^2$	98.23	$5.49 \times 10^4$	100.00	$1.12 \times 10^4$	100.00	$3.81 \times 10^{-1}$	100.00
25	$8.44 \times 10^0$	83.85	$2.38 \times 10^6$	100.00	$2.98 \times 10^2$	91.99	$5.94 \times 10^4$	100.00	$1.19 \times 10^4$	100.00	$4.14 \times 10^{-1}$	100.00
26	$8.90 \times 10^0$	69.17	$3.27 \times 10^6$	100.00	$6.93 \times 10^2$	84.33	$6.34 \times 10^4$	100.00	$1.38 \times 10^4$	100.00	$4.36 \times 10^{-1}$	100.00
27	$9.44 \times 10^0$	59.23	$6.59 \times 10^6$	100.00	$2.17 \times 10^3$	77.68	$6.70 \times 10^4$	100.00	$1.50 \times 10^4$	100.00	$3.92 \times 10^{-1}$	100.00
28	$9.82 \times 10^0$	55.63	$1.04 \times 10^7$	100.00	$4.89 \times 10^3$	76.88	$7.51 \times 10^4$	100.00	$1.78 \times 10^4$	100.00	$4.01 \times 10^{-1}$	100.00